nuvoTon

ARM Cortex®-M4

32-bit Microcontroller

NuMicro™ NUC442/NUC472 Series Technical Reference Manual

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

TABLE OF CONTENTS

List of Figure

List of Tables

1 GENERAL DESCRIPTION

1.1 NuMicro™ NUC442/NUC472 General Description

The NuMicro™ NUC442 Connectivity series with embedded Cortex®-M4F core with DSP extensions and a Floating Point Unit runs up to 84 MHz with 256/512 Kbytes embedded flash memories and 64K-byte embedded SRAM. It is also equipped with plenty of peripheral devices, such as Timers, Watchdog Timers, RTC, PDMA, EBI, UART, Smart Card interface, SD HOST, SPI, I^2C, I^2S, PWM Timer, GPIO, LIN, CAN, PS/2, 12-bit ADC, analog comparator, operational amplifier, temperature sensor, Low Voltage Reset Controller and Brown-out Detector. The NUC442 also provides USB 2.0 full-speed Device/Host/OTG, USB 2.0 HS device and security functions such as tamper detection, symmetric cryptographic accelerator and secure Hash function accelerator.

The NuMicro™ NUC472 Advanced series with embedded Cortex®-M4F core with DSP extensions and a Floating Point Unit runs up to 84 MHz with 256/512 Kbytes embedded flash memories and 64K-byte embedded SRAM. It is also equipped with plenty of peripheral devices, such as Timers, Watchdog Timers, RTC, PDMA, EBI, UART, Smart Card interface, SD HOST, SPI, I^2C, I^2S, PWM Timer, GPIO, LIN, CAN, PS/2, 12-bit ADC, analog comparator, operational amplifier, temperature sensor, Low Voltage Reset Controller and Brown-out Detector. The NUC472 also provides Ethernet 10/100 MAC with MII and RMII interface, USB 2.0 full-speed Device/Host/OTG, USB 2.0 HS device and security functions such as tamper detection, symmetric cryptographic accelerator and secure Hash function accelerator.

Product Series	Ethernet	USB	CAN	SD Host	UART	SPI	I^2C	Smart Card Interface	Security	ADC
NUC442		•	•	•	•	•	•	•	•	•
NUC472	•	•	•	•	•	•	•	•	•	•

Table 1.1-1 Key Features Support Table

The NuMicro™ NUC442/NUC472 series is suitable for a wide range of applications such as:

- Industrial Automation
- PLCs
- Inverters
- Home Automation
- Security Alarm System
- Power Metering
- Portable Data Collector
- Portable RFID Reader
- System Supervisors
- USB Accessories
- Smart Card Reader
- Printer
- POS
- Motor Control

2 FEATURES

2.1 NuMicro™ NUC442 Features – Connectivity Series

- Core
  - ARM® Cortex®-M4 core running up to 84 MHz
  - Supports DSP extension
    - Supports hardware divider
  - Supports IEEE 754 compliant Floating-point Unit (FPU)
  - Supports Memory Protection Unit (MPU)
  - One 24-bit system timer
  - Supports low power sleep mode
    - Supports both WFI and WFE instructions
  - Single-cycle 32-bit hardware multiplier
  - Supports Nested Vectored Interrupt Controller (NVIC)
    - Supports programmable 256 level priorities for interrupts
  - Supports programmable maskable interrupts
- Build-in LDO for wide operating voltage ranged from 2.5 V to 5.5 V
- Flash Memory
  - 256/512 Kbytes Flash memory
  - Configurable program code/data allocation
  - ISP loader sizes 16 Kbytes
  - Supports 2-wired ICP update through SWD/ICE interface
  - Supports In-system program (ISP), In application program (IAP) update
  - 2 Kbytes page erase for flash
  - Supports fast parallel programming mode by external programmer
- SRAM
  - 64 Kbytes embedded SRAM
  - 24 Kbytes SRAM with hardware parity check
  - Supports byte-, half-word- and word-access parity check
  - Supports exception (NMI) generated once a parity check error occurs
  - Supports PDMA mode
- PDMA (Peripheral DMA)
  - Supports 16 independent configurable channels for automatic data transfer between memories and peripherals
  - Supports normal and Scatter-Gather Transfer modes
  - Supports 2 types of priorities modes: fixed-priority and round-robin modes
  - Supports byte-, half-word- and word-access
  - Auto increment the source and destination address
  - Supports 16-level FIFO
  - Supports bus abort status flag
  - Supports time out status flag
- Clock Control
  - Flexible selection for different applications
  - Built-in 22.1184 MHz high speed RC oscillator for system operation (variation < 2% at -40℃ ~ +105℃)
  - Built-in 10 kHz low speed RC oscillator for Watchdog Timer and Wake-up operation
  - Built-in 4~24 MHz high speed oscillator for external crystal input for precise timing operation
  - Built-in 32.768 kHz low speed oscillator for external crystal input for RTC function and low power system operation

- Supports one PLL, up to 84 MHz for high performance system operation, sourced from
  - Built-in 22.1184 MHz high speed RC oscillator
  - 4~24 MHz external high speed crystal oscillator
- Supports clock failure detection for system clock
- Supports exception (NMI) generated once a clock failure detected
- Flexible selection for different applications
- Supports clock out
- CPU clock source can be selected from USB PHY Embedded PLL

● GPIO

- Four I/O modes:
  - Quasi-bidirectional
  - Push-Pull output
  - Open-Drain output
  - Input only with high impendence
- TTL/Schmitt trigger input selectable
- I/O pin configured as interrupt source with edge/level trigger setting
- High driver and high sink IO mode support (To source 20mA and sink 15mA at 5V)
- Supports up to 114/101/77/45 GPIOs for LQFP144/128/100/64, respectively.

● Timer

- Supports 4 sets of 32-bit timers with 24-bit up-timer and one 8-bit pre-scale counter
- Independent clock source for each timer
- Provides one-shot, periodic, toggle and continuous counting operation modes
- Supports event counting function to count the event from external pin
- Supports input capture function to capture or reset counter value

● Watchdog Timer

- Supports multiple clock sources
- 8 selectable time out period from 1.6ms ~ 26.0sec (depending on clock source)
- Able to wake up from Power-down or Idle mode
- Interrupt or reset selectable on watchdog time-out

● Window Watchdog Timer

- Supports multiple clock sources
- Window set by 6-bit counter with 11-bit pre-scale
- Interrupt or reset selectable on time-out

● RTC

- Supports software compensation by setting frequency compensate register (FCR)
- Supports RTC counter (second, minute, hour) and calendar counter (day, month, year)
- Supports Alarm registers (second, minute, hour, day, month, year)
- Selectable 12-hour or 24-hour mode
- Automatic leap year recognition
- Supports periodic time tick interrupt with 8 period options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second
- Supports wake-up function
- Supports 96 bytes backup registers
- Programmable backup-register erase function
- Supports external power input pin (V_{BAT})
- Supports tamper detection function

● PWM

- Supports up to two 6-channel PWM outputs with 16-bit resolution
- Supports 8-bit presale and clock divider
- Supports period point, center point and edge point PWM Interrupt

- Supports One-shot or Auto-reload PWM counter operation mode
- Supports Edge-aligned or Center-aligned PWM counter type
- Supports 8-bit dead zone with maximum divided 8 pre-scale
- Supports brake function source from pin or comparator output
- Supports mask function for each PWM pin
- Supports independent, complementary, synchronized and group PWM output mode
- Supports trigger ADC start conversion at PWM counter period point, PWM counter center point, PWM output rising edge and PWM output falling edge
- Supports 12 Capture input channels with 16-bit resolution
- Supports rising or falling capture condition
- Supports capture interrupt

● EPWM (Enhanced PWM)

- Supports up to two EPWM
- Each EPWM has
  - Three independent 16-bit PWM duty control units with maximum 6 port pins
  - Group control bit: PWM2 and PWM4 are synchronized with PWM0
  - Supports Edge-aligned mode and Center-aligned mode
  - Programmable dead-time insertion between complementary paired PWMs
  - Each pin of from PWM0 to PWM5 has independent polarity setting control
  - Mask output control for Electrically Commutated Motor operation
  - Tri-state output at reset and brake state
  - Hardware brake protections
  - Two Interrupt Sources
  - PWM signals before polarity control stage are defined in view of positive logic. The PWM ports active high or active low are controlled by polarity control register.
  - High Source/Sink current

● Quadrature Encoder Interface (QEI)

- Supports up to two QEI controllers, QEI0 and QEI1
- Each QEI has:
  - Two QEI phase inputs, QEA and QEB; One Index input
  - One QEI control register (QEI_CTR) and one QEI Status Register (QEI_STS)
  - Four Quadrature encoder pulse counter operation modes

● Enhanced Input Capture Timer

- Supports up to two Input Capture Timer/Counter Units, Input Capture 0 and Input Capture 1
- Each unit has own interrupt vector
- 24-bit Input Capture up-counting timer/counter
- With noise filter in front end of input ports
- Edge detector with three options
  - Rising edge detection
  - Falling edge detection
  - Both edge detection
- Each input channel is supported with one capture counter hold register
- Captured event reset/reload capture counter option
- Supports compare-match function

● UART

- Supports up to six UART controllers
- Supports flow control (CTS and RTS)
- UART0 with 64-byte FIFO is for high speed
- UART1~5 with 16-byte FIFO for standard device
- Supports IrDA (SIR) and LIN function

- Supports RS-485 9-bit mode and direction control
- Programmable baud-rate generator up to 1/16 system clock
- Supports PDMA mode

● Smart Card Interface

- Supports up to six ISO-7816-3 ports
- Compliant to ISO-7816-3 T=0, T=1
- Separate receive / transmit 4 bytes entry FIFO for data payloads
- Programmable transmission clock frequency
- Programmable receiver buffer trigger level
- Programmable guard time selection (11 ETU ~ 267 ETU)
- A 24-bit and two 8 bit time out counter for Answer to Request (ATR) and waiting times processing
- Supports auto inverse convention function
- Supports transmitter and receiver error retry and error limit function
- Supports hardware activation/deactivation sequence process
- Supports hardware warm reset sequence process
- Supports hardware auto deactivation sequence when detect the card is removal
- Supports UART function

● SPI

- Up to four sets of SPI controllers
- Supports Master or Slave mode operation
- Supports 2-bit Transfer mode
- Supports Dual and Quad I/O Transfer mode
- Configurable bit length of a transfer word from 8 to 32-bit
- Provides separate 8-level depth transmit and receive FIFO buffers
- Supports MSB first or LSB first transfer sequence
- Supports byte reorder function
- Supports Byte or Word Suspend mode
- Supports PDMA transfer
- Supports 3-wire, no slave select signal, bi-direction interface
- Master up to 32 MHz, and Slave up to 16 MHz (chip working at 5V)

● I^2C

- Supports up to five sets of I^2C device
- Master/Slave mode
- Bidirectional data transfer between masters and slaves
- Multi-master bus
- Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
- Serial clock synchronization allows devices with different bit rates to communicate via one serial bus
- Serial clock synchronization can be used as a handshake mechanism to suspend and resume serial transfer
- Programmable clocks allowing versatile rate control
- Supports multiple address recognition (four slave address with mask option)
- Supports speed up to 1Mbps
- Supports PDMA mode
- Supports multi-address wake-up function

● I^2S

- Supports up to two I^2S interface
- Interface with external audio CODEC
- Supports Master and Slave mode
- Capable of handling 8-, 16-, 24- and 32-bit word sizes

- Supports mono and stereo audio data
- Supports I^2S and MSB justified data format
- Each provides two 8-word FIFO data buffers, one for transmitting and the other for receiving
- Generates interrupt requests when buffer levels cross a programmable boundary
- Each supports two PDMA requests, one for transmitting and the other for receiving

● CAN 2.0

- Supports up to two CAN controllers
- Supports CAN protocol version 2.0 part A and B
- Bit rates up to 1M bit/s
- Each supports 32 Message Objects
- Each Message Object has its own identifier mask
- Programmable FIFO mode (concatenation of Message Object)
- Supports interrupts
- Disabled Automatic Re-transmission mode for Time Triggered CAN applications
- Supports power-down wake-up function

● PS/2 Device Controller

- Host communication inhibit and request to send detection
- Reception frame error detection
- Programmable 1 to 16 bytes transmit buffer to reduce CPU intervention
- Double buffer for data reception
- S/W override bus

● USB 2.0 Controller

- Supports one set of USB 2.0 FS Device/Host/OTG or USB 2.0 HS Device
- Supports one set of USB 2.0 FS Host
- FS Host compatible with Open HCI 1.0 specification
- Compliant to USB specification version 2.0
- OTG supports USB OTG Supplement 1.3
- On-chip USB Transceiver
- Supports Control, Bulk In/Out, Interrupt and Isochronous transfers
- Auto suspend function when no bus signaling for 3 ms
- Provides 12 programmable endpoints and one dedicated control end point
- Supports 4095 bytes internal SRAM as USB buffer
- Provides remote wake-up capability
- On-chip 5V to 3.3V LDO for USB PHY
- On-chip PLL able to support 480 MHz clock
- Supports DMA master

● EBI

- Supports accessible space up to 256MB configured into 4 memory blocks (64MB/Memory Block), the actually external addressable space is dependent on package pin out
- Dedicated external chip select pin for each memory block
- Supports 8-/16-bit data width
- Supports byte write in 16-bit data width mode
- Supports PDMA mode
- Supports Address/Data Separated/Multiplexed Mode
- Supports Timing parameters individual adjustment for each memory block
- Supports “Timing Transparent Encrypt/Decrypt” for protecting data in each memory block (Individual Enable/Disable)

● Image Capture Interface

- CCIR601 & CCIR656 interfaces supported for connection to CMOS image sensor

- Resolution up to 3M pixel
- YUV422 and RGB565 color format supported for data-in from CMOS sensor
- YUV422, RGB565, RGB555 and Y-only color format supported for data storing to system memory
- Planar and packet data format supported for data storing to system memory
- Image cropping supported with the cropping window up to 4096x2048
- Image scaling-down supported
- Vertical and horizontal scaling-down for preview mode supported
- Scaling factor as N/M
- Two pairs of configurable 8-bit N and 8-bit M for vertical and horizontal scaling-down
- The value of N has to be equal to or less than M
- Frame rate control supported
- Combines two interlace fields to a single frame supported for data in from TV-decoder

● ADC

- Supports two operating modes: ADC mode and EADC (Enhance ADC mode with dual ADC Sampling)
- Selected as ADC mode
  - Supports single 12-bit ADC conversion
  - Analog input voltage range: 0~AV_{DD}
  - Up to 12 external single-ended analog input channels
  - Up to 6 differential analog input pairs
  - Supports single ADC interrupt
  - Supports easy control for power saving
  - External Vref pin can be used as input
  - Supports PDMA transfer
- Selected as EADC mode
  - Supports two 12-bit ADC simultaneous conversion
  - Analog input voltage range: 0~ AV_{DD}
  - Up to 16 external single-ended analog input channels
  - Each ADC can convert individually at normal operation
  - Four ADC interrupts with individual interrupt vector addresses
  - An A/D conversion source can be triggered by different events
  - Conversion results are held in 16 data registers with valid and overrun indicators
  - Sampling-oriented trigger setting and input setting for each sampling
  - Supports converting internal OP0, OP1 Amplifier output voltage
  - Supports converting internal band-gap voltage, internal temperature sensor output and analog ground

● Analog Comparator

- Supports up to three rail-to-rail analog comparators
- External input or internal Band-gap voltage selectable at negative node
- Interrupts generated when compare results change
- Supports power-down wake-up

● Operational Amplifier

- Supports up to two analog operational amplifiers
- Outputs can be used as the input of ADC

● Cryptographic Accelerator

- DES/TDES accelerator
  - Supports hardware DES (Data Encryption Standard)/TDES (Triple DES) accelerator
  - Supports 56, 112 and 168-bit keys
  - Supports ECB, CBC, CFB, OFB and CTR modes
  - Compliant with NIST 800 38A

- AES accelerator
  - Supports hardware AES (Advanced Encryption Standard) accelerator
  - Supports 128-, 192- and 256-bit keys
  - Supports ECB, CBC, CFB, OFB and CTR modes
  - Compliant with NIST 800 38A
- Secure Hash Function accelerator
  - Supports hardware SHA (Secure Hash) accelerator
  - Supports SHA-1 and SHA-224, -256
  - Compliant with FIPS 180-2

- Random Number Generator
  - Supports random bit generator
  - Supports a random number generator programmable 64, 128, 192 and 256 bits
- Cyclic Redundancy Calculation Unit
  - Supports four common polynomials CRC-CCITT, CRC-8, CRC-16, and CRC-32
  - Programmable initial value
  - Supports programmable order reverse setting for input data and CRC checksum
  - Supports programmable 1's complement setting for input data and CRC checksum
  - Supports 8/16/32-bit of data width
  - Interrupt generated once checksum error occurs
- SD Host Interface
  - Supports SD (Secure Digital) card and SD HOST interface
  - Compliant with SD Memory Card Specification Version 2.0
  - Supports 1 and 4-bit modes
  - Supports 25 MHz to achieve 100 Mbps at 3.3V operation
  - Supports DMA master
- Tamper Detection
  - Supports external tamper detection up to 2 input pins
  - Reset, NMI or Interrupt generated once tamper detected
- Debug
  - Supports Flash Patch and Breakpoint Unit (FPB)
    - Supports 8 hardware breakpoints
    - Supports 6 watchpoints
  - Supports the following debug ports
    - 2-pin Serial Wire Debug port (SWD)
    - Serial Wire Viewer (SWV)
- Supports 96-bit Unique ID (UID)
- Supports 128-bit Unique Customer ID (UCID)
- One built-in temperature sensor with 1℃ resolution
- Brown-out Detector
  - With 4 levels: 4.4 V/ 3.7 V/ 2.7 V/ 2.2 V
  - Supports Brown-out Interrupt and Reset option
- Low Voltage Reset
  - Threshold voltage levels: 2.0 V
- Operating Temperature: -40℃~105℃
- Packages
  - All Green package (RoHS)

- LQFP 144-pin/ 128-pin/ 100-pin/ 64-pin

2.2 NuMicro™ NUC472 Features - Advanced Series

- Core
  - ARM® Cortex®-M4 core running up to 84 MHz
  - Supports DSP extension
    - Supports hardware divider
  - Supports IEEE 754 compliant Floating-point Unit (FPU)
  - Supports Memory Protection Unit (MPU)
  - One 24-bit system timer
  - Supports low power sleep mode
    - Supports both WFI and WFE instructions
  - Single-cycle 32-bit hardware multiplier
  - Supports Nested Vectored Interrupt Controller (NVIC)
    - Supports programmable 256 level priorities for interrupts
  - Supports programmable maskable interrupts
- Build-in LDO for wide operating voltage ranged from 2.5 V to 5.5 V
- Flash Memory
  - 256K/512 Kbytes Flash memory
  - Configurable program code/data allocation
  - ISP loader sizes 16 Kbytes
  - Supports 2-wired ICP update through SWD/ICE interface
  - Supports In-system program (ISP), In application program (IAP) update
  - 2 Kbytes page erase for flash
  - Supports fast parallel programming mode by external programmer
- SRAM
  - 64 Kbytes embedded SRAM
  - 24 Kbytes SRAM with hardware parity check
  - Supports byte-, half-word- and word-access parity check
  - Supports exception (NMI) generated once a parity check error occurs
  - Supports PDMA mode
- PDMA (Peripheral DMA)
  - Supports 16 independent configurable channels for automatic data transfer between memories and peripherals
  - Supports normal and Scatter-Gather Transfer modes
  - Supports 2 types of priorities modes: fixed-priority and round-robin modes
  - Supports byte-, half-word- and word-access
  - Auto increment the source and destination address
  - Supports 16-level FIFO
  - Supports bus abort status flag
  - Supports time out status flag
- Clock Control
  - Flexible selection for different applications
  - Built-in 22.1184 MHz high speed RC oscillator for system operation (variation < 2% at -40℃ ~ +105℃)
  - Built-in 10 kHz low speed RC oscillator for Watchdog Timer and Wake-up operation
  - Built-in 4~24 MHz high speed oscillator for external crystal input for precise timing operation
  - Built-in 32.768 kHz low speed oscillator for external crystal input for RTC function and low power system operation
  - Supports one PLL, up to 84 MHz for high performance system operation, sourced from
    - Built-in 22.1184 MHz high speed RC oscillator

- 4~24 MHz external high speed crystal oscillator
- Supports clock failure detection for system clock
- Supports exception (NMI) generated once a clock failure detected
- Flexible selection for different applications
- Supports clock out
- CPU clock source can be selected from USB PHY Embedded PLL

- GPIO
  - Four I/O modes:
    - Quasi-bidirectional
    - Push-Pull output
    - Open-Drain output
    - Input only with high impendence
  - TTL/Schmitt trigger input selectable
  - I/O pin configured as interrupt source with edge/level trigger setting
  - High driver and high sink IO mode support (To source 20mA and sink 15mA at 5V)
  - Supports up to 144/114/101/77 GPIOs for LQFP176/144/128/100, respectively.
- Timer
  - Supports 4 sets of 32-bit timers with 24-bit up-timer and one 8-bit pre-scale counter
  - Independent clock source for each timer
  - Provides one-shot, periodic, toggle and continuous counting operation modes
  - Supports event counting function to count the event from external pin
  - Supports input capture function to capture or reset counter value
- Watchdog Timer
  - Supports multiple clock sources
  - 8 selectable time out period from 1.6ms ~ 26.0sec (depending on clock source)
  - Able to wake up from Power-down or Idle mode
  - Interrupt or reset selectable on watchdog time-out
- Window Watchdog Timer
  - Supports multiple clock sources
  - Window set by 6-bit counter with 11-bit pre-scale
  - Interrupt or reset selectable on time-out
- RTC
  - Supports software compensation by setting frequency compensate register (FCR)
  - Supports RTC counter (second, minute, hour) and calendar counter (day, month, year)
  - Supports Alarm registers (second, minute, hour, day, month, year)
  - Selectable 12-hour or 24-hour mode
  - Automatic leap year recognition
  - Supports periodic time tick interrupt with 8 period options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second
  - Supports wake-up function
  - Supports 96 bytes backup registers
  - Programmable backup-register erase function
  - Supports external power input pin (V_{BAT})
  - Supports tamper detection function
- PWM
  - Supports up to two 6-channel PWM outputs with 16-bit resolution
  - Supports 8-bit presale and clock divider
  - Supports period point, center point and edge point PWM Interrupt
  - Supports One-shot or Auto-reload PWM counter operation mode
  - Supports Edge-aligned or Center-aligned PWM counter type

- Supports 8-bit dead zone with maximum divided 8 pre-scale
- Supports brake function source from pin or comparator output
- Supports mask function for each PWM pin
- Supports independent, complementary, synchronized and group PWM output mode
- Supports trigger ADC start conversion at PWM counter period point, PWM counter center point, PWM output rising edge and PWM output falling edge
- Supports 12 Capture input channels with 16-bit resolution
- Supports rising or falling capture condition
- Supports capture interrupt

● EPWM (Enhanced PWM)

- Supports up to two EPWM
- Each EPWM has
  - Three independent 16-bit PWM duty control units with maximum 6 port pins.
  - Group control bit: PWM2 and PWM4 are synchronized with PWM0
  - Supports Edge-aligned mode and Center-aligned mode
  - Programmable dead-time insertion between complementary paired PWMs
  - Each pin of from PWM0 to PWM5 has independent polarity setting control
  - Mask output control for Electrically Commutated Motor operation
  - Tri-state output at reset and brake state
  - Hardware brake protections
  - Two Interrupt Sources
  - PWM signals before polarity control stage are defined in view of positive logic. The PWM ports active high or active low are controlled by polarity control register.
  - High Source/Sink current

● Quadrature Encoder Interface (QEI)

- Supports up to two QEI controllers, QEI0 and QEI1
- Each QEI has:
  - Two QEI phase inputs, QEA and QEB; One Index input
  - One QEI control register (QEI_CTR) and one QEI Status Register (QEI_STS)
  - Four Quadrature encoder pulse counter operation modes

● Enhanced Input Capture Timer

- Supports up to two Input Capture Timer/Counter Units, Input Capture 0 and Input Capture 1
- Each unit has own interrupt vector
- 24-bit Input Capture up-counting timer/counter
- With noise filter in front end of input ports
- Edge detector with three options
  - Rising edge detection
  - Falling edge detection
  - Both edge detection
- Each input channel is supported with one capture counter hold register
- Captured event reset/reload capture counter option
- Supports compare-match function

● UART

- Supports up to six UART controllers
- Supports flow control (CTS and RTS)
- UART0 with 64-byte FIFO is for high speed
- UART1~5 with 16-byte FIFO for standard device
- Supports IrDA (SIR) and LIN function
- Supports RS-485 9-bit mode and direction control
- Programmable baud-rate generator up to 1/16 system clock

- Supports PDMA mode

- Smart Card Interface
  - Supports up to six ISO-7816-3 ports
  - Compliant to ISO-7816-3 T=0, T=1
  - Separate receive / transmit 4 bytes entry FIFO for data payloads
  - Programmable transmission clock frequency
  - Programmable receiver buffer trigger level
  - Programmable guard time selection (11 ETU ~ 267 ETU)
  - A 24-bit and two 8 bit time out counter for Answer to Request (ATR) and waiting times processing
  - Supports auto inverse convention function
  - Supports transmitter and receiver error retry and error limit function
  - Supports hardware activation/deactivation sequence process
  - Supports hardware warm reset sequence process
  - Supports hardware auto deactivation sequence when detect the card is removal
  - Supports UART function
- SPI
  - Up to four sets of SPI controllers
  - Supports Master or Slave mode operation
  - Supports 2-bit Transfer mode
  - Supports Dual and Quad I/O Transfer mode
  - Configurable bit length of a transfer word from 8 to 32-bit
  - Provides separate 8-level depth transmit and receive FIFO buffers
  - Supports MSB first or LSB first transfer sequence
  - Supports byte reorder function
  - Supports Byte or Word Suspend mode
  - Supports PDMA transfer
  - Supports 3-wire, no slave select signal, bi-direction interface
  - Master up to 32 MHz, and Slave up to 16 MHz (chip working at 5V)
- I^2C
  - Supports up to five sets of I^2C device
  - Master/Slave mode
  - Bidirectional data transfer between masters and slaves
  - Multi-master bus
  - Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
  - Serial clock synchronization allows devices with different bit rates to communicate via one serial bus
  - Serial clock synchronization can be used as a handshake mechanism to suspend and resume serial transfer
  - Programmable clocks allowing versatile rate control
  - Supports multiple address recognition (four slave address with mask option)
  - Supports speed up to 1Mbps
  - Supports PDMA mode
  - Supports multi-address wake-up function
- I^2S
  - Supports up to two I^2S interface
  - Interface with external audio CODEC
  - Supports Master and Slave mode
  - Capable of handling 8-, 16-, 24- and 32-bit word sizes
  - Supports mono and stereo audio data
  - Supports I^2S and MSB justified data format

- Each provides two 8-word FIFO data buffers, one for transmitting and the other for receiving
- Generates interrupt requests when buffer levels cross a programmable boundary
- Each supports two PDMA requests, one for transmitting and the other for receiving

- CAN 2.0
  - Supports up to two CAN controllers
  - Supports CAN protocol version 2.0 part A and B
  - Bit rates up to 1M bit/s
  - Each supports 32 Message Objects
  - Each Message Object has its own identifier mask
  - Programmable FIFO mode (concatenation of Message Object)
  - Supports interrupts
  - Disabled Automatic Re-transmission mode for Time Triggered CAN applications
  - Supports power-down wake-up function
- PS/2 Device Controller
  - Host communication inhibit and request to send detection
  - Reception frame error detection
  - Programmable 1 to 16 bytes transmit buffer to reduce CPU intervention
  - Double buffer for data reception
  - S/W override bus
- USB 2.0 Controller
  - Supports one set of USB 2.0 FS Device/Host/OTG or USB 2.0 HS Device
  - Supports one set of USB 2.0 FS Host
  - FS Host compatible with Open HCI 1.0 specification
  - Compliant to USB specification version 2.0
  - OTG supports USB OTG Supplement 1.3
  - On-chip USB Transceiver
  - Supports Control, Bulk In/Out, Interrupt and Isochronous transfers
  - Auto suspend function when no bus signaling for 3 ms
  - Provides 12 programmable endpoints and one dedicated control end point
  - Supports 4095 bytes internal SRAM as USB buffer
  - Provides remote wake-up capability
  - On-chip 5V to 3.3V LDO for USB PHY
  - On-chip PLL able to support 480 MHz clock
  - Supports DMA master
- EBI
  - Supports accessible space up to 256MB configured into 4 memory blocks (64MB/Memory Block), the actually external addressable space is dependent on package pin out
  - Dedicated external chip select pin for each memory block
  - Supports 8-/16-bit data width
  - Supports byte write in 16-bit data width mode
  - Supports PDMA mode
  - Supports Address/Data Separated/Multiplexed Mode
  - Supports Timing parameters individual adjustment for each memory block
  - Supports “Timing Transparent Encrypt/Decrypt” for protecting data in each memory block (Individual Enable/Disable)
- Image Capture Interface
  - CCIR601 & CCIR656 interfaces supported for connection to CMOS image sensor
  - Resolution up to 3M pixel
  - YUV422 and RGB565 color format supported for data-in from CMOS sensor

- YUV422, RGB565, RGB555 and Y-only color format supported for data storing to system memory
- Planar and packet data format supported for data storing to system memory
- Image cropping supported with the cropping window up to 4096x2048
- Image scaling-down supported
- Vertical and horizontal scaling-down for preview mode supported
- Scaling factor as N/M
- Two pairs of configurable 8-bit N and 8-bit M for vertical and horizontal scaling-down
- The value of N has to be equal to or less than M
- Frame rate control supported
- Combines two interlace fields to a single frame supported for data in from TV-decoder

● ADC

- Supports two operating modes: ADC mode and EADC (Enhance ADC mode with dual ADC Sampling)
- Selected as ADC mode
  - Supports single 12-bit ADC conversion
  - Analog input voltage range: 0~AV_{DD}
  - Up to 12 external single-ended analog input channels
  - Up to 6 differential analog input pairs
  - Supports single ADC interrupt
  - Supports easy control for power saving
  - External Vref pin can be used as input
  - Supports PDMA transfer
- Selected as EADC mode
  - Supports two 12-bit ADC simultaneous conversion
  - Analog input voltage range: 0~ AV_{DD}
  - Up to 16 external single-ended analog input channels
  - Each ADC can convert individually at normal operation
  - Four ADC interrupts with individual interrupt vector addresses
  - An A/D conversion source can be triggered by different events
  - Conversion results are held in 16 data registers with valid and overrun indicators
  - Sampling-oriented trigger setting and input setting for each sampling
  - Supports converting internal OP0, OP1 Amplifier output voltage
  - Supports converting internal band-gap voltage, internal temperature sensor output and analog ground

● Analog Comparator

- Supports up to three rail-to-rail analog comparators
- External input or internal Band-gap voltage selectable at negative node
- Interrupts generated when compare results change
- Supports power-down wake-up

● Operational Amplifier

- Supports up to two analog operational amplifiers
- Outputs can be used as the input of ADC

● Cryptographic Accelerator

- DES/TDES accelerator
  - Supports hardware DES (Data Encryption Standard)/TDES (Triple DES) accelerator
  - Supports 56, 112 and 168-bit keys
  - Supports ECB, CBC, CFB, OFB and CTR modes
  - Compliant with NIST 800 38A
- AES accelerator
  - Supports hardware AES (Advanced Encryption Standard) accelerator

- Supports 128-, 192- and 256-bit keys
- Supports ECB, CBC, CFB, OFB and CTR modes
- Compliant with NIST 800 38A

- Secure Hash Function accelerator
  - Supports hardware SHA (Secure Hash) accelerator
  - Supports SHA-1 and SHA-224, -256
  - Compliant with FIPS 180-2

- Random Number Generator
  - Supports random bit generator
  - Supports a random number generator programmable 64, 128, 192 and 256 bits
- Cyclic Redundancy Calculation Unit
  - Supports four common polynomials CRC-CCITT, CRC-8, CRC-16, and CRC-32
  - Programmable initial value
  - Supports programmable order reverse setting for input data and CRC checksum
  - Supports programmable 1's complement setting for input data and CRC checksum
  - Supports 8/16/32-bit of data width
  - Interrupt generated once checksum error occurs
- Ethernet 10/100 MAC
  - Compliant with IEEE 802.3-2002
  - Supports MII and RMII interface
  - Supports IEEE 1588 v2
  - Supports TX buffer and RX buffer each with 256 bytes
  - Supports DMA Mode
- SD Host Interface
  - Supports SD (Secure Digital) card and SD HOST interface
  - Compliant with SD Memory Card Specification Version 2.0
  - Supports 1 and 4-bit modes
  - Supports 25 MHz to achieve 100 Mbps at 3.3V operation
  - Supports DMA master
- Tamper Detection
  - Supports external tamper detection up to 2 input pins
  - Reset, NMI or Interrupt generated once tamper detected
- Debug
  - Supports Flash Patch and Breakpoint Unit (FPB)
    - Supports 8 hardware breakpoints
    - Supports 6 watchpoints
  - Supports the following debug ports
    - 2-pin Serial Wire Debug port (SWD)
    - Serial Wire Viewer (SWV)
- Supports 96-bit Unique ID (UID)
- Supports 128-bit Unique Customer ID (UCID)
- One built-in temperature sensor with 1℃ resolution
- Brown-out Detector
  - With 4 levels: 4.4 V/ 3.7 V/ 2.7 V/ 2.2 V
  - Supports Brown-out Interrupt and Reset option
- Low Voltage Reset
  - Threshold voltage levels: 2.0 V

- Operating Temperature: -40℃~105℃
- Packages
  - All Green package (RoHS)
  - LQFP 176-pin/ 144-pin/ 128-pin/ 100-pin

3 ABBREVIATIONS

Acronym	Description
ACMP	Analog Comparator Controller
ADC	Analog-to-Digital Converter
AES	Advanced Encryption Standard
APB	Advanced Peripheral Bus
AHB	Advanced High-Performance Bus
BOD	Brown-out Detection
CAN	Controller Area Network
DAP	Debug Access Port
DES	Data Encryption Standard
EBI	External Bus Interface
EPWM	Enhanced Pulse Width Modulation
FIFO	First In, First Out
FMC	Flash Memory Controller
FPU	Floating-point Unit
GPIO	General-Purpose Input/Output
HCLK	The Clock of Advanced High-Performance Bus
HIRC	22.1184 MHz Internal High Speed RC Oscillator
HXT	4~24 MHz External High Speed Crystal Oscillator
IAP	In Application Programming
ICP	In Circuit Programming
ISP	In System Programming
LDO	Low Dropout Regulator
LIN	Local Interconnect Network
LIRC	10 kHz internal low speed RC oscillator (LIRC)
MPU	Memory Protection Unit
NVIC	Nested Vectored Interrupt Controller
PCLK	The Clock of Advanced Peripheral Bus
PDMA	Peripheral Direct Memory Access
PLL	Phase-Locked Loop
PWM	Pulse Width Modulation
QEI	Quadrature Encoder Interface
SD	Secure Digital

SPI	Serial Peripheral Interface
SPS	Samples per Second
TDES	Triple Data Encryption Standard
TMR	Timer Controller
UART	Universal Asynchronous Receiver/Transmitter
UCID	Unique Customer ID
USB	Universal Serial Bus
WDT	Watchdog Timer
WWDT	Window Watchdog Timer

Table 3.1-1 List of Abbreviations

4 PARTS INFORMATION LIST AND PIN CONFIGURATION

4.1 Selection Guide

4.1.1 NuMicro™ NUC442 Connectivity Series Selection Guide

Part Number	Flash (KB)	SRAM (KB)	ISP Loader ROM (KB)	I/O	Timer	Connectivity								Ethernet	PWM	Comp	OP	ADC (12-Bit)	RTC	ISO-7816-3*	ISP/ICP/IAP	Package
						UART	SPI	I^2C	USB	LIN	CAN	SC	I^2S									
NUC442JI8AE	512	64	16	114	4	6+6	4	5	v	6	2	6	2	--	16	3	2	x2, 16-ch	v	6	v	LQFP144
NUC442JG8AE	256	64	16	114	4	6+6	4	5	v	6	2	6	2	--	16	3	2	x2, 16-ch	v	6	v	LQFP144
NUC442KI8AE	512	64	16	100	4	6+6	4	5	v	6	2	6	2	--	16	3	2	x2, 16-ch	v	6	v	LQFP128
NUC442KG8AE	256	64	16	100	4	6+6	4	5	v	6	2	6	2	--	16	3	2	x2, 16-ch	v	6	v	LQFP128
NUC442VI8AE	512	64	16	77	4	6+5	4	5	v	6	2	5	2	--	16	3	--	x2, 16-ch	v	5	v	LQFP100
NUC442VG8AE	256	64	16	77	4	6+5	4	5	v	6	2	5	2	--	16	3	--	x2, 16-ch	v	5	v	LQFP100
NUC442RI8AE	512	64	16	45	4	4+3	3	2	v	4	2	3	1	--	8	2	--	x2, 8-ch	v	3	v	LQFP64
NUC442RG8AE	256	64	16	45	4	4+3	3	2	v	4	2	3	1	--	8	2	--	x2, 8-ch	v	3	v	LQFP64

*Marked in this table (6+6) means 6 UART + 6 ISO-7816 UART

*ISO-7816 UART supports full duplex mode

4.1.2 NuMicro™ NUC472 Advanced Series Selection Guide

Part Number	Flash (KB)	SRAM (KB)	ISP Loader ROM (KB)	I/O	Timer	Connectivity								Ethernet	PWM	Comp	OP	ADC (12-Bit)	RTC	ISO-7816-3*	ISP/ICP/IAP	Package
						UART	SPI	I^2C	USB	LIN	CAN	SC	I^2S									
NUC472HI8AE	512	64	16	144	4	6+6	4	5	v	6	2	6	6	v	16	3	2	x2, 16-ch	v	6	v	LQFP176
NUC472HG8AE	256	64	16	144	4	6+6	4	5	v	6	2	6	6	v	16	3	2	x2, 16-ch	v	6	v	LQFP176
NUC472JI8AE	512	64	16	114	4	6+6	4	5	v	6	2	6	6	v	16	3	2	x2, 16-ch	v	6	v	LQFP144

NUC472JG8AE	256	64	16	114	4	6+6	4	5	v	6	2	6	6	v	16	3	2	x2, 16-ch	v	6	v	LQFP144
NUC472KI8AE	512	64	16	101	4	6+6	4	5	v	6	2	6	5	v	16	3	2	x2, 16-ch	v	6	v	LQFP128
NUC472KG8AE	256	64	16	101	4	6+6	4	5	v	6	2	6	5	v	16	3	2	x2, 16-ch	v	6	v	LQFP128
NUC472VI8AE	512	64	16	77	4	6+5	4	5	v	6	2	5	3	v	16	3	--	x2, 16-ch	v	5	v	LQFP100
NUC472VG8AE	256	64	16	77	4	6+5	4	5	v	6	2	5	3	v	16	3	--	x2, 16-ch	v	5	v	LQFP100

*Marked in this table (6+6) means 6 UART + 6 ISO-7816 UART

*ISO-7816 UART supports full duplex mode

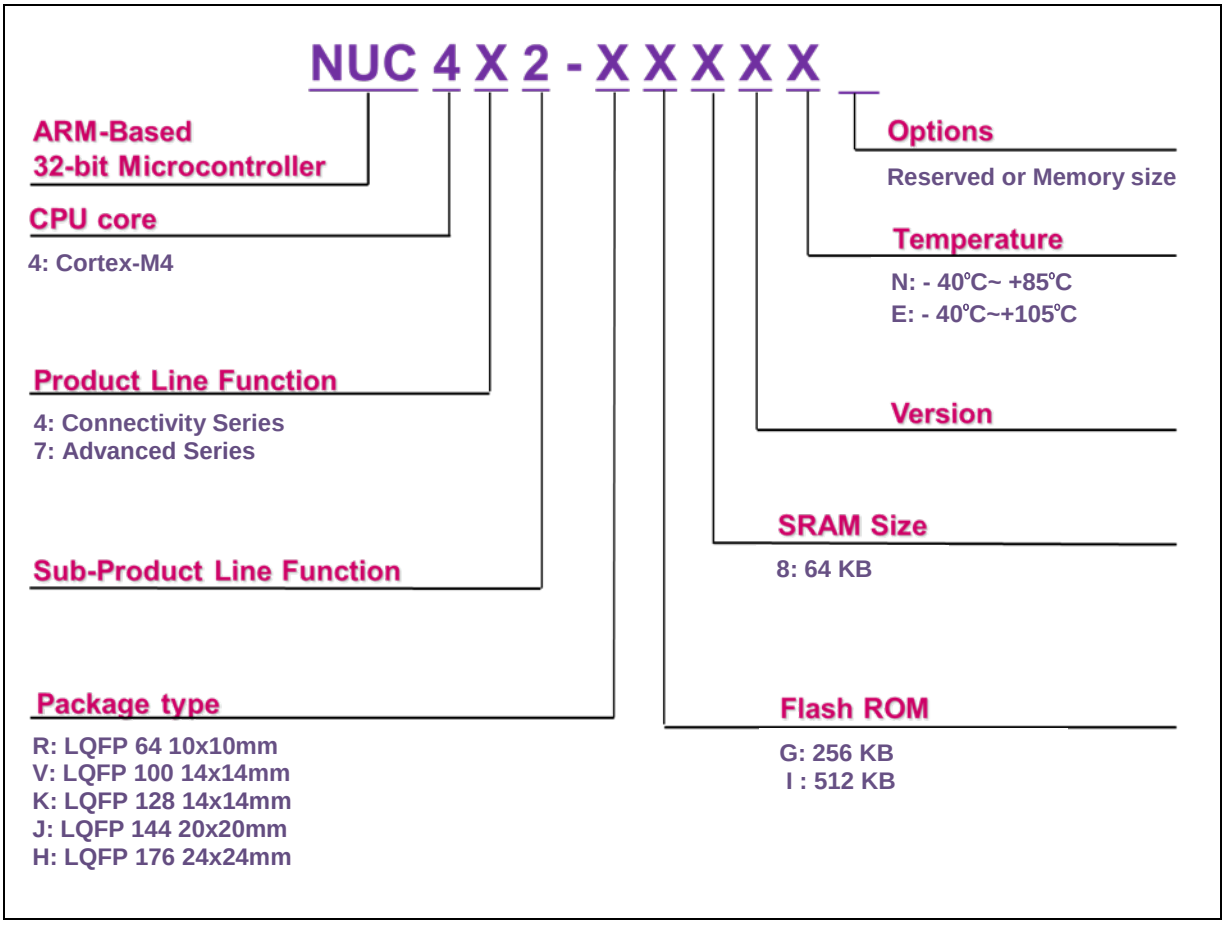


Figure 4.1-1 NuMicro™ NUC442/NUC472 Series Selection Code

4.2 Pin Configuration

4.2.1 NuMicro™ NUC442 Pin Diagrams

4.2.1.1 NuMicro™ NUC442Rxxxx LQFP 64-pin

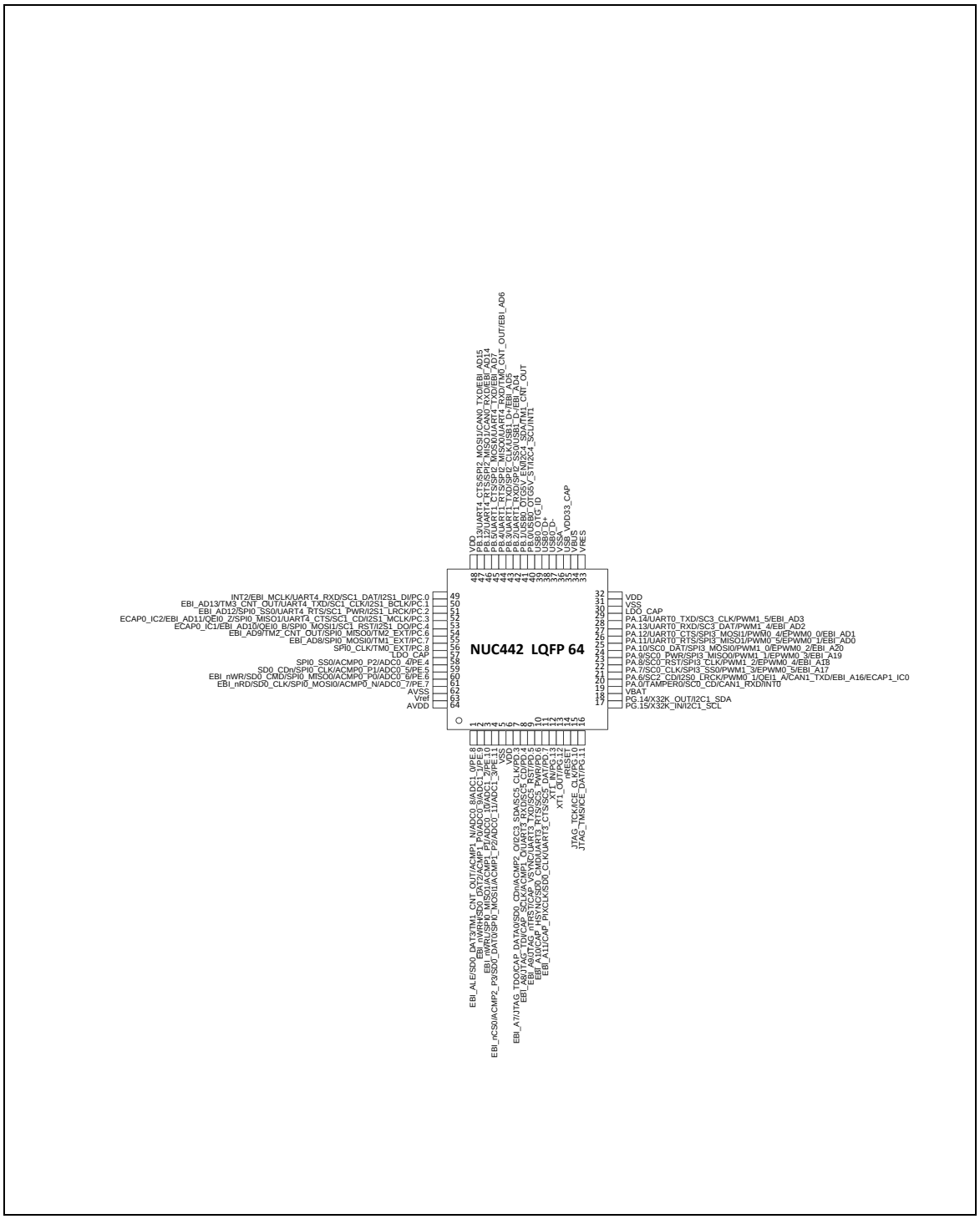


Figure 4.2-1 NuMicro™ NUC442Rxxxx LQFP 64-pin Diagram

4.2.1.2 NuMicro™ NUC442Vxxxx LQFP 100-pin

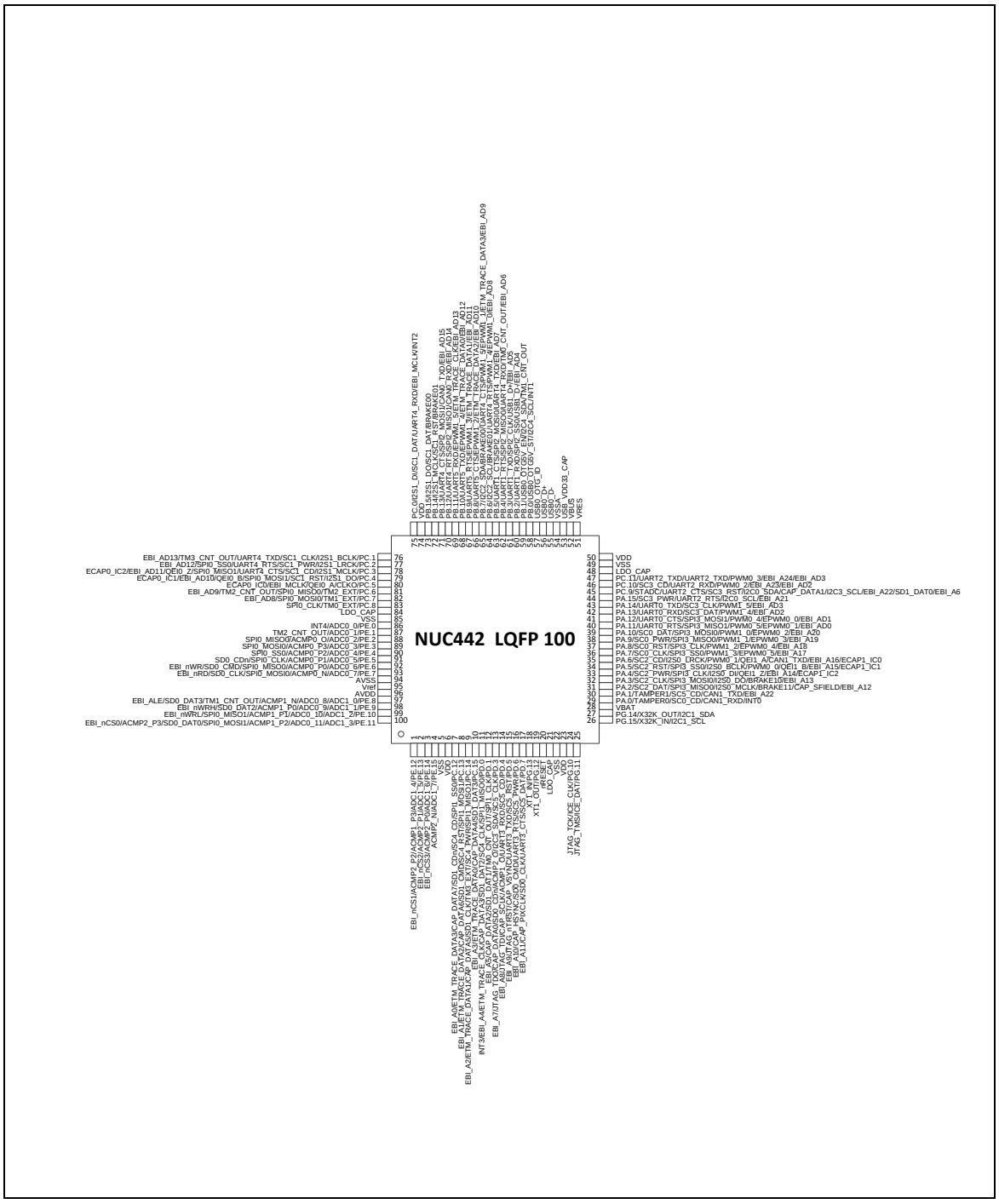


Figure 4.2-2 NuMicro™ NUC442Vxxxx LQFP 100-pin Diagram

4.2.1.3 NuMicro™ NUC442Kxxxx LQFP 128-pin

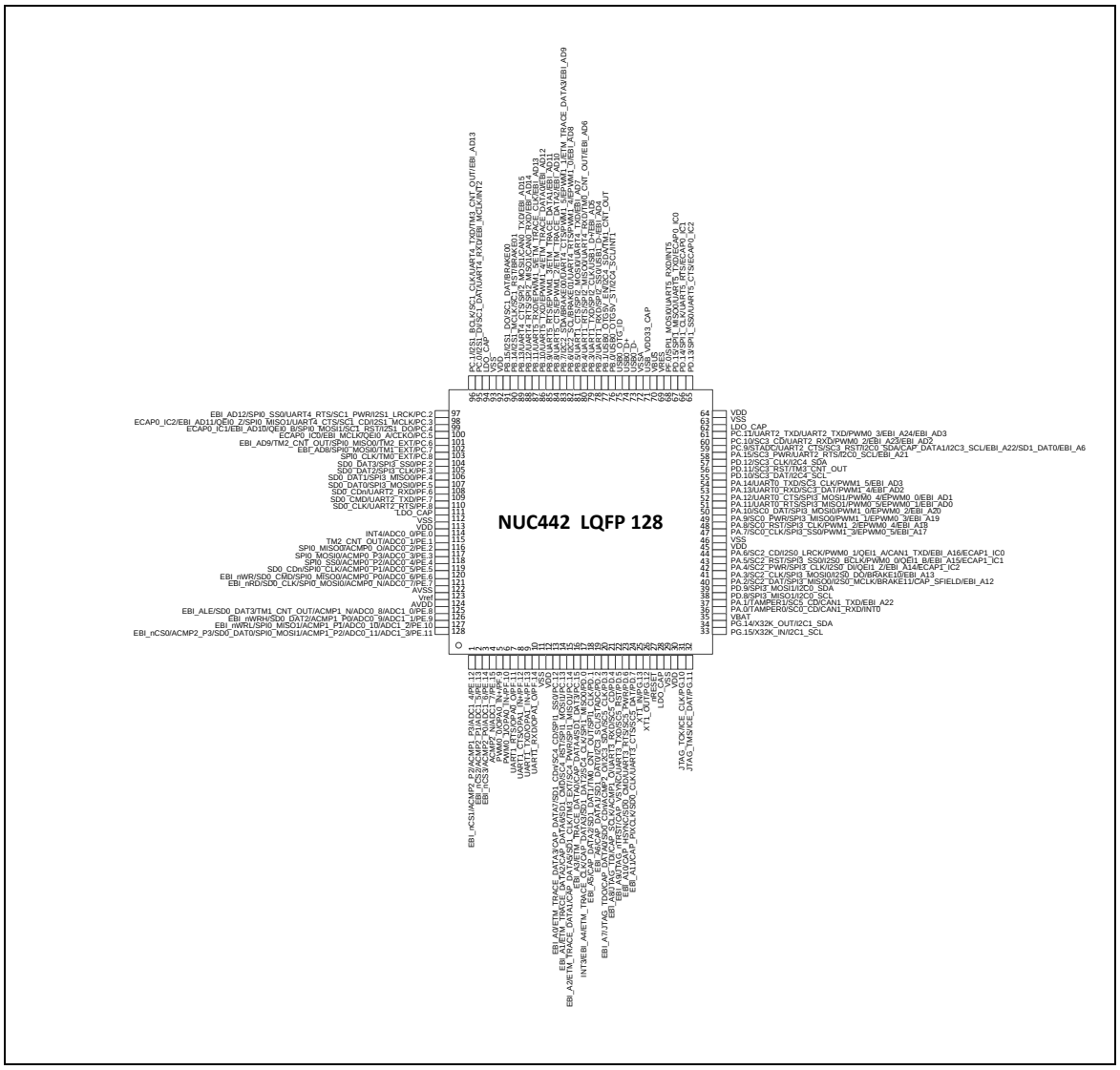


Figure 4.2-3 NuMicro™ NUC442Kxxxx LQFP 128-pin Diagram

4.2.1.4 NuMicro™ NUC442Jxxxx LQFP 144-pin

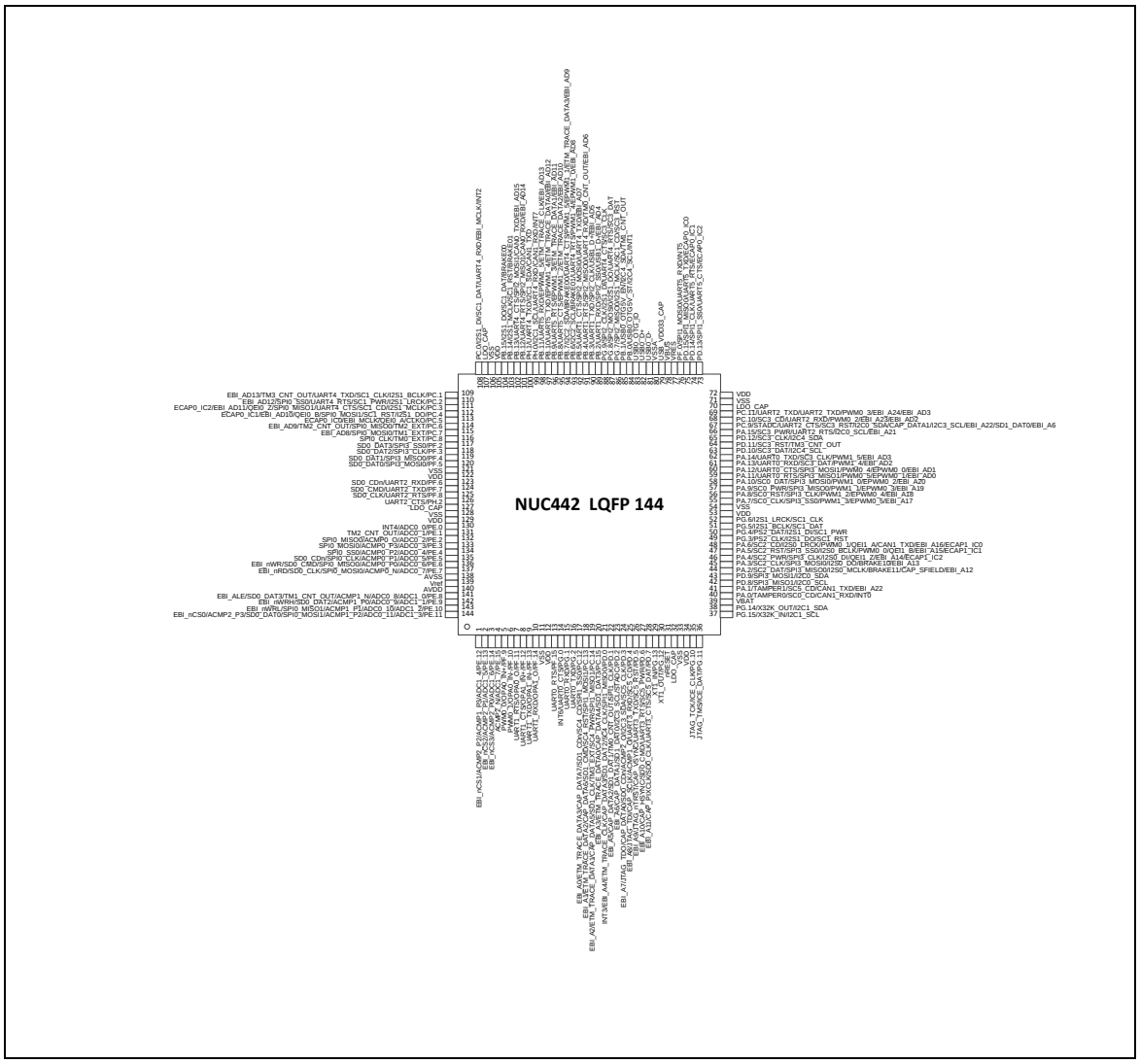


Figure 4.2-4 NuMicro™ NUC442Jxxxx LQFP 144-pin Diagram

4.2.2 NuMicro™ NUC472 Pin Diagrams

4.2.2.1 NuMicro™ NUC472Vxxxx LQFP 100-pin

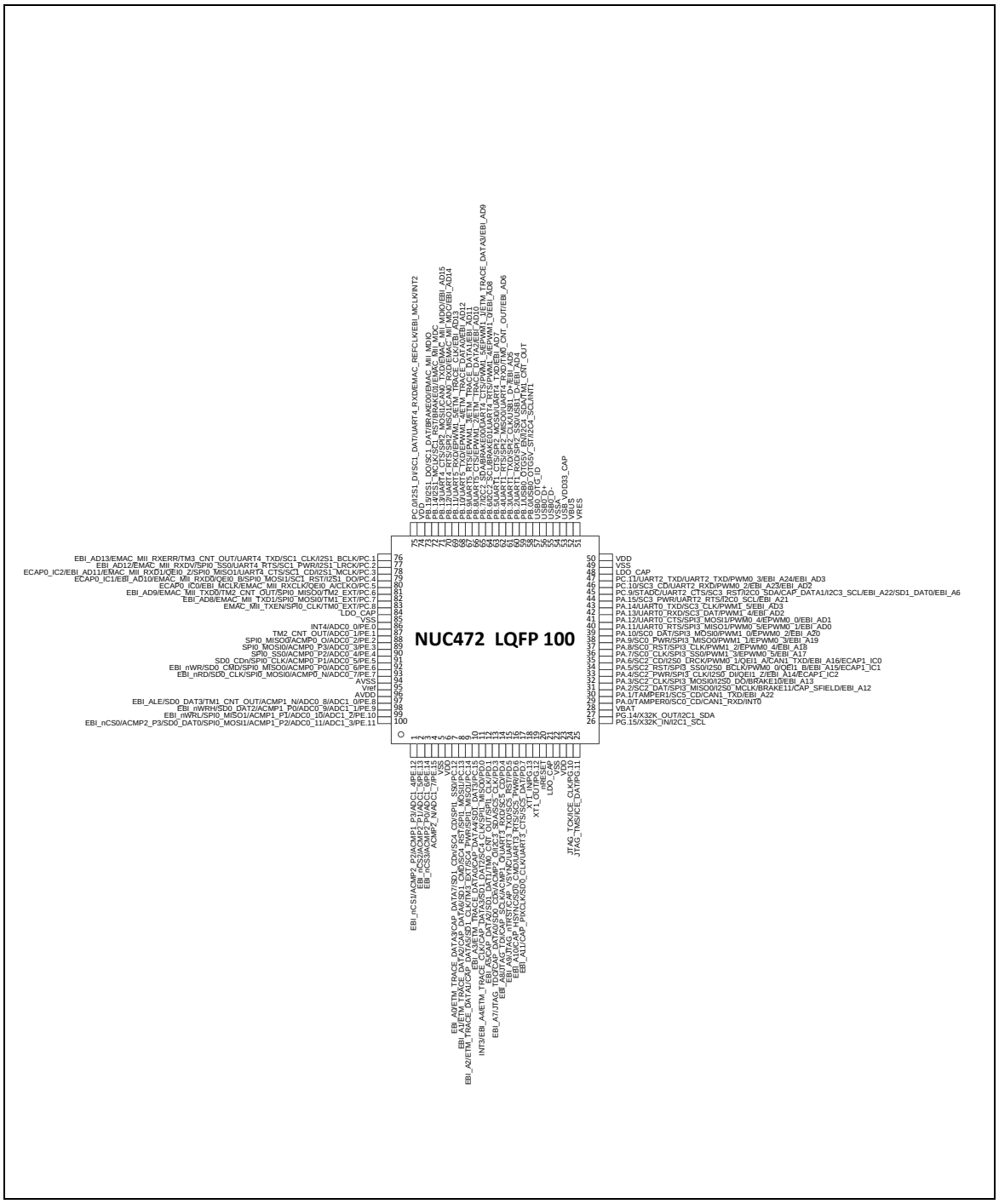


Figure 4.2-5 NuMicro™ NUC472Vxxxx LQFP 100-pin Diagram

4.2.2.2 NuMicro™ NUC472Kxxxx LQFP 128-pin

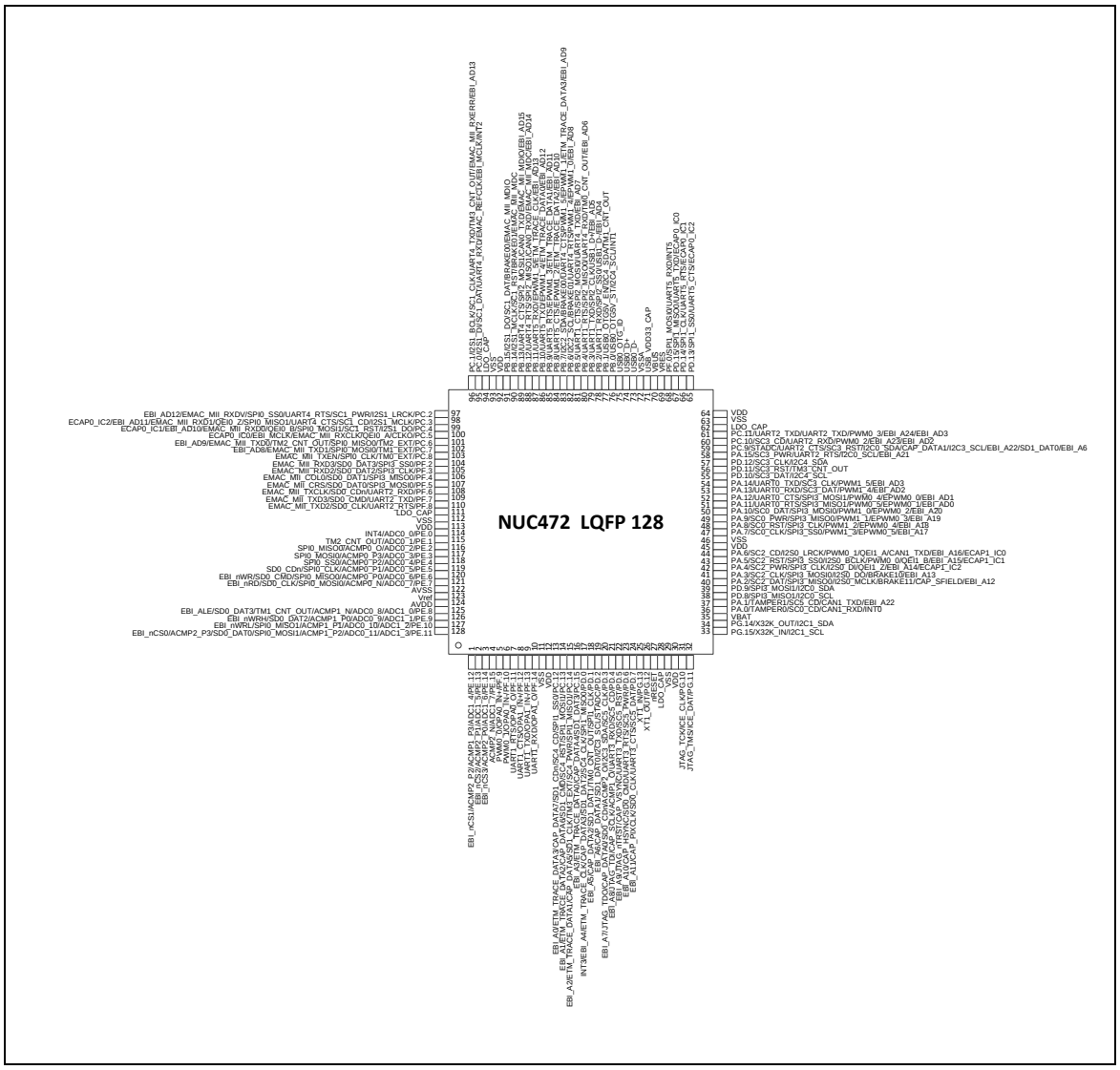


Figure 4.2-6 NuMicro™ NUC472Kxxxx LQFP 128-pin Diagram

4.2.2.3 NuMicro™ NUC472Jxxxx LQFP 144-pin

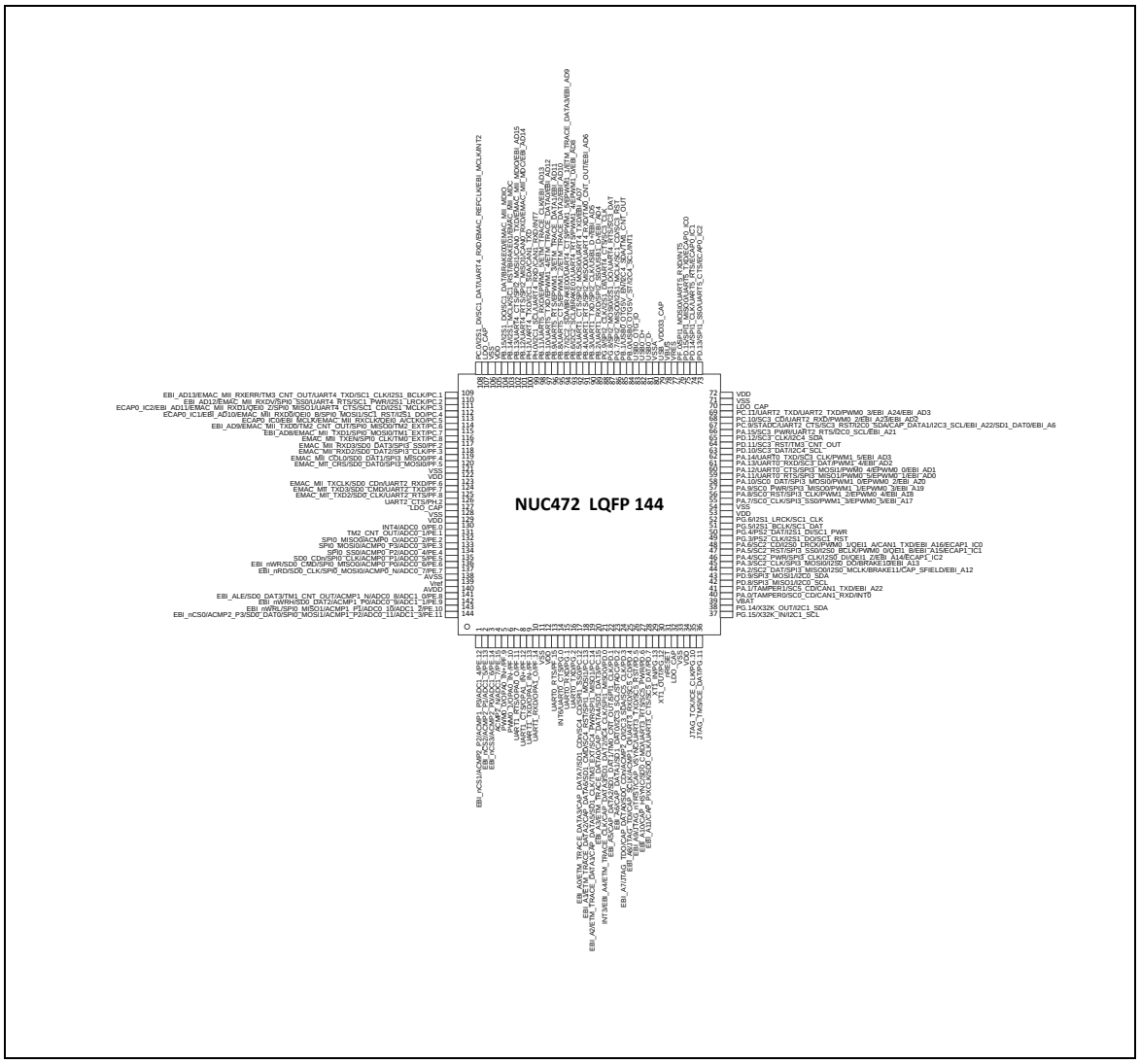


Figure 4.2-7 NuMicro™ NUC472Jxxxx LQFP 144-pin Diagram

4.2.2.4 NuMicro™ NUC472Hxxxx LQFP 176-pin

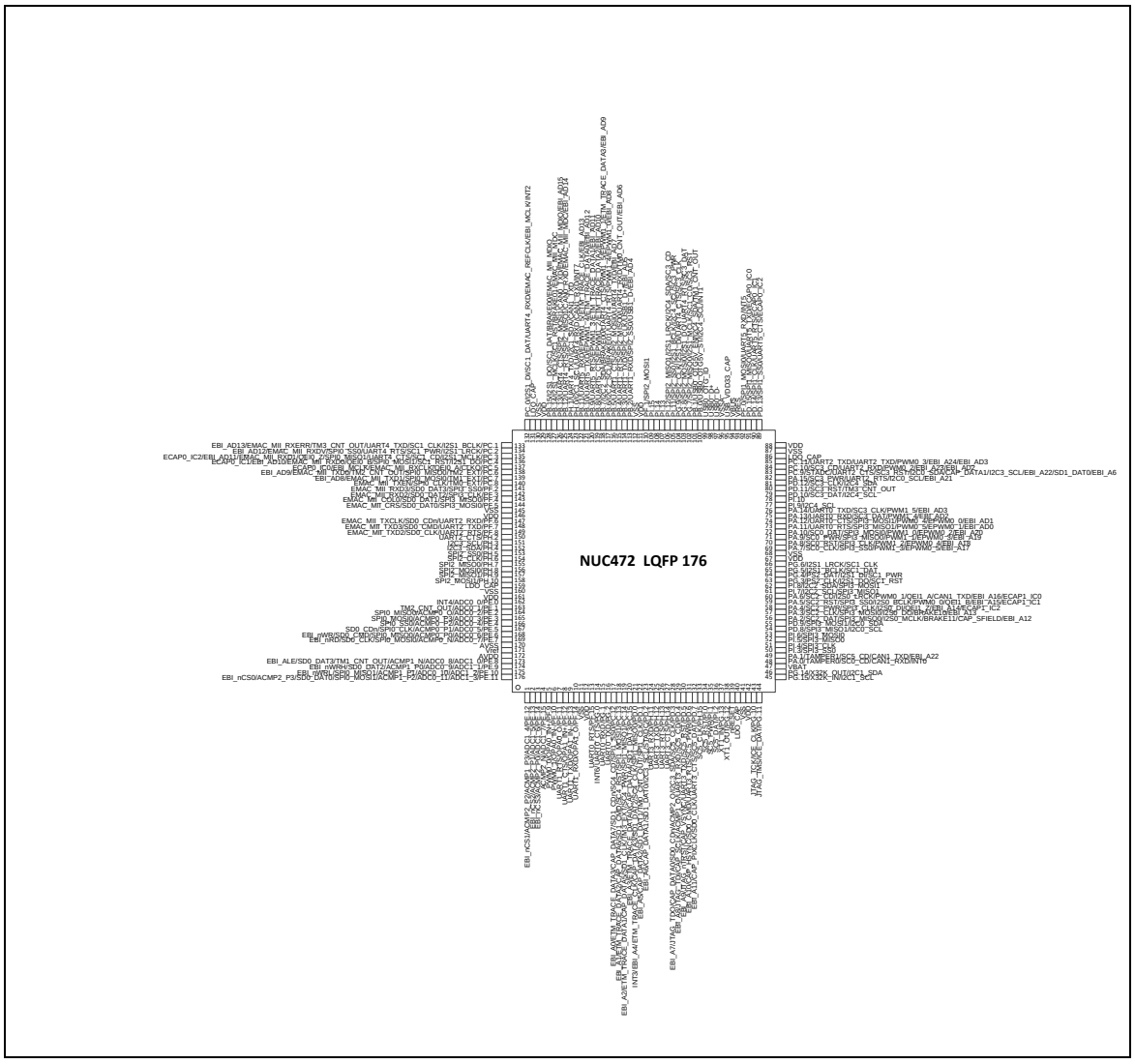


Figure 4.2-8 NuMicro™ NUC472Hxxxx LQFP 176-pin Diagram

4.3 Pin Description

4.3.1 NuMicro™ NUC442 Package LQFP 64-pin Description

MFP = Multi-function pin.

Pin No.	Pin Name	Type	MFP*	Description
1	PE.8	I/O	MFP0	General purpose digital I/O pin.
	ADC1_0	A	MPF1	ADC1 analog input.
	ADC0_8	A	MPF1	ADC0 analog input.
	ACMP1_N	A	MPF2	Analog comparator1 negative input pin.
	TM1_CNT_OUT	I/O	MPF3	Timer1 event counter input/toggle output.
	SD0_DAT3	I/O	MPF4	SD mode #0 data line bit 3.
	EBI_ALE	O	MPF7	EBI address latch enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
2	PE.9	I/O	MFP0	General purpose digital I/O pin.
	ADC1_1	A	MPF1	ADC1 analog input.
	ADC0_9	A	MPF1	ADC0 analog input.
	ACMP1_P0	A	MPF2	Analog comparator1 positive input pin.
	SD0_DAT2	I/O	MPF4	SD mode #0 data line bit 2.
	EBI_nWRH	O	MPF7	EBI write enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
3	PE.10	I/O	MFP0	General purpose digital I/O pin.
	ADC1_2	A	MPF1	ADC1 analog input.
	ADC0_10	A	MPF1	ADC0 analog input.
	ACMP1_P1	A	MPF2	Analog comparator1 positive input pin.
	SPI0_MISO1	I/O	MPF3	2nd SPI0 MISO (Master In, Slave Out) pin.
	SD0_DAT1	I/O	MPF4	SD mode #0 data line bit 1.
	EBI_nWRL	O	MPF7	EBI write enable output pin.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
4	PE.11	I/O	MFP0	General purpose digital I/O pin.
	ADC1_3	A	MPF1	ADC1 analog input.
	ADC0_11	A	MPF1	ADC0 analog input.
	ACMP1_P2	A	MPF2	Analog comparator1 positive input pin.
	SPI0_MOSI1	I/O	MPF3	2nd SPI0 MOSI (Master Out, Slave In) pin.
	SD0_DAT0	I/O	MPF4	SD mode #0 data line bit 0.
	ACMP2_P3	A	MPF5	Analog comparator2 positive input pin.
	EBI_nCS0	O	MPF7	EBI chip select 0 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
5	VSS	P	MFP0	Ground pin for digital circuit.
6	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
7	PD.3	I/O	MFP0	General purpose digital I/O pin.
	SC5_CLK	O	MPF1	SmartCard5 clock pin.
	I2C3_SDA	I/O	MPF2	I2C3 data input/output pin.
	ACMP2_O	O	MPF3	Analog ccomparator2 output .
	SD0_CDn	I	MPF4	SD mode #0 – card detect
	CAP_DATA0	I	MPF5	Image data input bus bit 7.
	EBI_A7	O	MPF7	EBI address bus bit7.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
8	PD.4	I/O	MFP0	General purpose digital I/O pin.
	SC5_CD	I	MPF1	SmartCard5 card detect pin.
	UART3_RXD	I	MPF2	Data receiver input pin for UART3.
	ACMP1_O	O	MPF3	Analog ccomparator1 output .
	CAP_SCLK	O	MPF5	Image capture interface sensor clock pin.

	EBI_A8	O	MPF7	EBI address bus bit8.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
9	PD.5	I/O	MFP0	General purpose digital I/O pin.
	SC5_RST	O	MPF1	SmartCard5 reset pin.
	UART3_TXD	O	MPF2	Data transmitter output pin for UART3.
	CAP_VSYNC	I	MPF5	Image capture interface VSYNC input pin.
	EBI_A9	O	MPF7	EBI address bus bit9.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
10	PD.6	I/O	MFP0	General purpose digital I/O pin.
	SC5_PWR	O	MPF1	SmartCard5 power pin.
	UART3_RTS	O	MPF2	Request to Send output pin for UART3.
	SD0_CMD	I/O	MPF4	SD mode #0 – command/response
	CAP_HSYNC	I	MPF5	Image capture interface HSYNC input pin.
	EBI_A10	O	MPF7	EBI address bus bit10.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
11	PD.7	I/O	MFP0	General purpose digital I/O pin.
	SC5_DAT	I/O	MPF1	SmartCard5 data pin.
	UART3_CTS	I	MPF2	Clear to Send input pin for UART3.
	SD0_CLK	O	MPF4	SD mode #0– clock.
	CAP_PIXCLK	I	MPF5	Image capture interface pix clock input pin.
	EBI_A11	O	MPF7	EBI address bus bit11.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
12	PG.13	I/O	MFP0	General purpose digital I/O pin.
	XT1_IN	I	MPF1	External 4~24 MHz (high-speed) crystal input pin.
13	PG.12	I/O	MFP0	General purpose digital I/O pin.

	XT1_OUT	O	MPF1	External 4~24 MHz (high-speed) crystal output pin.
14	nRESET	I	MFP0	External reset input: active LOW, with an internal pull-up. Set this pin low reset to initial state.
15	PG.10	I/O	MFP0	General purpose digital I/O pin.
	ICE_CLK	I	MPF1	Serial wired debugger clock pin
16	PG.11	I/O	MFP0	General purpose digital I/O pin.
	ICE_DAT	I/O	MPF1	Serial wired debugger data pin
17	PG.15	I/O	MFP0	General purpose digital I/O pin.
	X32K_IN	I	MPF1	External 32.768 kHz (low-speed) crystal input pin.
	I2C1_SCL	I/O	MPF3	I2C1 clock pin.
18	PG.14	I/O	MFP0	General purpose digital I/O pin.
	X32K_OUT	O	MPF1	External 32.768 kHz (low-speed) crystal output pin.
	I2C1_SDA	I/O	MPF3	I2C1 data input/output pin.
19	VBAT	P	MFP0	Battery power input pin.
20	PA.0	I/O	MFP0	General purpose digital I/O pin.
	TAMPER0	I/O	MPF1	Tamper detect pin 0.
	SC0_CD	I	MPF2	SmartCard0 card detect pin.
	CAN1_RXD	I	MPF3	CAN bus receiver1 input.
	INT0	I	MPF8	External interrupt0 input pin.
21	PA.6	I/O	MFP0	General purpose digital I/O pin.
	SC2_CD	I	MPF1	SmartCard2 card detect pin.
	I2S0_LRCK	O	MPF3	I2S0 left right channel clock.
	PWM0_1	I/O	MPF4	PWM0_1 output/capture input.
	QEI1_A	I	MPF5	Quadrature encoder phase A input of QEI Unit 1.
	CAN1_TXD	I	MPF6	CAN bus transmitter1 input.
	EBI_A16	O	MPF7	EBI address bus bit16.
	ECAP1_IC0	I	MPF8	Input 0 of enhanced capture unit 1.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
22	PA.7	I/O	MFP0	General purpose digital I/O pin.
	SC0_CLK	O	MPF2	SmartCard0 clock pin.
	SPI3_SS0	I/O	MPF3	General purpose digital I/O pin.
	PWM1_3	I/O	MPF4	PWM1_3 output/capture input.
	EPWM0_5	I/O	MPF5	PWM0_5 output/capture input.
	EBI_A17	O	MPF7	EBI address bus bit17.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
23	PA.8	I/O	MFP0	General purpose digital I/O pin.
	SC0_RST	O	MPF2	SmartCard0 reset pin.
	SPI3_CLK	O	MPF3	SPI3 serial clock pin.
	PWM1_2	I/O	MPF4	PWM1_2 output/capture input.
	EPWM0_4	I/O	MPF5	PWM0_4 output/capture input.
	EBI_A18	O	MPF7	EBI address bus bit18.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
24	PA.9	I/O	MFP0	General purpose digital I/O pin.
	SC0_PWR	O	MPF2	SmartCard0 power pin.
	SPI3_MISO0	I/O	MPF3	1st SPI3 MISO (Master In, Slave Out) pin.
	PWM1_1	I/O	MPF4	PWM1_1 output/capture input.
	EPWM0_3	I/O	MPF5	PWM0_3 output/capture input.
	EBI_A19	O	MPF7	EBI address bus bit19.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
25	PA.10	I/O	MFP0	General purpose digital I/O pin.
	SC0_DAT	I/O	MPF2	SmartCard0 data pin.
	SPI3_MOSI0	I/O	MPF3	1st SPI3 MOSI (Master Out, Slave In) pin.

	PWM1_0	I/O	MPF4	PWM1_0 output/capture input.
	EPWM0_2	I/O	MPF5	PWM0_2 output/capture input.
	EBI_A20	O	MPF7	EBI address bus bit20.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
26	PA.11	I/O	MFP0	General purpose digital I/O pin.
	UART0_RTS	O	MPF1	Request to Send output pin for UART0.
	SPI3_MISO1	I/O	MPF3	2nd SPI3 MISO (Master In, Slave Out) pin.
	PWM0_5	I/O	MPF4	PWM0_5 output/capture input.
	EPWM0_1	I/O	MPF5	PWM0_1 output/capture input.
	EBI_AD0	O	MPF7	EBI address/data bus bit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
27	PA.12	I/O	MFP0	General purpose digital I/O pin.
	UART0_CTS	I	MPF1	Clear to Send input pin for UART0.
	SPI3_MOSI1	I/O	MPF3	2nd SPI3 MOSI (Master Out, Slave In) pin.
	PWM0_4	I/O	MPF4	PWM0_4 output/capture input.
	EPWM0_0	I/O	MPF5	PWM0_0 output/capture input.
	EBI_AD1	O	MPF7	EBI address/data bus bit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
28	PA.13	I/O	MFP0	General purpose digital I/O pin.
	UART0_RXD	I	MPF1	Data receiver input pin for UART0.
	SC3_DAT	I/O	MPF3	SmartCard3 data pin.
	PWM1_4	I/O	MPF4	PWM1_4 output/capture input.
	EBI_AD2	O	MPF7	EBI address/data bus bit 2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
29	PA.14	I/O	MFP0	General purpose digital I/O pin.

	UART0_TXD	O	MPF1	Data transmitter output pin for UART0.
	SC3_CLK	O	MPF3	SmartCard3 clock pin.
	PWM1_5	I/O	MPF4	PWM1_5 output/capture input.
	EBI_AD3	O	MPF7	EBI address/data bus bit 3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
30	LDO_CAP	P	MFP0	LDO output pin.
31	VSS	P	MFP0	Ground pin for digital circuit.
32	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
33	VRES	A	MFP0	USB PHY VRES ground input pin. Add an 8.2K ohm resistor to VSSA.
34	VBUS	A	MFP0	USB PHY VBUS power input pin.
35	USB_VDD33_CAP	P	MFP0	Internal power regulator output 3.3V decoupling pin.
36	VSSA	P	MFP0	Ground pin for digital circuit. Add a Feritte Bead to digital ground VSS.
37	USB0_D-	A	MFP0	USB0 differential signal D+.
38	USB0_D+	A	MFP0	USB0 differential signal D+.
39	USB0_OTG_ID	I	MFP0	USB0OTG ID pin.
40	PB.0	I/O	MFP0	General purpose digital I/O pin.
	USB0_VBUS_ST	I	MPF1	USB0 external VBUS regulator status
	I2C4_SCL	I/O	MPF2	I2C4 clock pin.
	INT1	I	MPF8	External interrupt1 input pin.
41	PB.1	I/O	MFP0	General purpose digital I/O pin.
	USB0_VBUS_EN	O	MPF1	USB0 external VBUS regulator enable
	I2C4_SDA	I/O	MPF2	I2C4 data input/output pin.
	TM1_CNT_OUT	I/O	MPF3	Timer1 event counter input/toggle output.
42	PB.2	I/O	MFP0	General purpose digital I/O pin.
	UART1_RXD	I	MPF1	Data receiver input pin for UART1.

	SPI2_SS0	I/O	MPF2	General purpose digital I/O pin.
	USB1_D-	A	MPF3	USB1 differential signal D+.
	EBI_AD4	O	MPF7	EBI address/data bus bit 4.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
43	PB.3	I/O	MFP0	General purpose digital I/O pin.
	UART1_TXD	O	MPF1	Data transmitter output pin for UART1.
	SPI2_CLK	O	MPF2	SPI2 serial clock pin.
	USB1_D+	A	MPF3	USB1 differential signal D+.
	EBI_AD5	O	MPF7	EBI address/data bus bit 5.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
44	PB.4	I/O	MFP0	General purpose digital I/O pin.
	UART1_RTS	O	MPF1	Request to Send output pin for UART1.
	SPI2_MISO0	I/O	MPF2	1st SPI2 MISO (Master In, Slave Out) pin.
	UART4_RXD	I	MPF3	Data receiver input pin for UART4.
	TM0_CNT_OUT	I/O	MPF4	Timer0 event counter input/toggle output.
	EBI_AD6	O	MPF7	EBI address/data bus bit 6.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
45	PB.5	I/O	MFP0	General purpose digital I/O pin.
	UART1_CTS	I	MPF1	Clear to Send input pin for UART1.
	SPI2_MOSI0	I/O	MPF2	1st SPI2 MOSI (Master Out, Slave In) pin.
	UART4_TXD	O	MPF3	Data transmitter output pin for UART4.
	EBI_AD7	O	MPF7	EBI address/data bus bit 7.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
46	PB.12	I/O	MFP0	General purpose digital I/O pin.
	UART4_RTS	O	MPF1	Request to Send output pin for UART4.

	SPI2_MISO1	I/O	MPF2	2nd SPI2 MISO (Master In, Slave Out) pin.
	CAN0_RXD	I	MPF3	CAN bus receiver0 input.
	EBI_AD14	O	MPF7	EBI address/data bus bit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
47	PB.13	I/O	MFP0	General purpose digital I/O pin.
	UART4_CTS	I	MPF1	Clear to Send input pin for UART4.
	SPI2_MOSI1	I/O	MPF2	2nd SPI2 MOSI (Master Out, Slave In) pin.
	CAN0_TXD	I	MPF3	CAN bus transmitter0 input.
	EBI_AD15	O	MPF7	EBI address/data bus bit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
48	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
49	PC.0	I/O	MFP0	General purpose digital I/O pin.
	I2S1_DI	I	MPF1	I2S1 data input.
	SC1_DAT	I/O	MPF2	SmartCard1 data pin.
	UART4_RXD	I	MPF3	Data receiver input pin for UART4.
	EBI_MCLK	O	MPF7	EBI interface clock output pin.
	INT2	I	MPF8	External interrupt2 input pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
50	PC.1	I/O	MFP0	General purpose digital I/O pin.
	I2S1_BCLK	O	MPF1	I2S1 bit clock pin.
	SC1_CLK	O	MPF2	SmartCard1 clock pin.
	UART4_TXD	O	MPF3	Data transmitter output pin for UART4.
	TM3_CNT_OUT	I/O	MPF5	Timer3 event counter input/toggle output.
	EBI_AD13	O	MPF7	EBI address/data bus bit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

51	PC.2	I/O	MFP0	General purpose digital I/O pin.
	I2S1_LRCK	O	MPF1	I2S1 left right channel clock.
	SC1_PWR	O	MPF2	SmartCard1 power pin.
	UART4_RTS	O	MPF3	Request to Send output pin for UART4.
	SPI0_SS0	I/O	MPF4	General purpose digital I/O pin.
	EBI_AD12	O	MPF7	EBI address/data bus bit 12.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
52	PC.3	I/O	MFP0	General purpose digital I/O pin.
	I2S1_MCLK	O	MPF1	I2S1 master clock output pin.
	SC1_CD	I	MPF2	SmartCard1 card detect pin.
	UART4_CTS	I	MPF3	Clear to Send input pin for UART4.
	SPI0_MISO1	I/O	MPF4	2nd SPI0 MISO (Master In, Slave Out) pin.
	QEI0_Z	I	MPF5	Quadrature encoder phase Z input of QEI Unit 0.
	EBI_AD11	O	MPF7	EBI address/data bus bit 11.
	ECAP0_IC2	O	MPF8	Input 2 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
53	PC.4	I/O	MFP0	General purpose digital I/O pin.
	I2S1_DO	O	MPF1	I2S1 data output.
	SC1_RST	O	MPF2	SmartCard1 reset pin.
	SPI0_MOSI1	I/O	MPF4	2nd SPI0 MOSI (Master Out, Slave In) pin.
	QEI0_B	I	MPF5	Quadrature encoder phase B input of QEI Unit 0.
	EBI_AD10	O	MPF7	EBI address/data bus bit 10.
	ECAP0_IC1	O	MPF8	Input 1 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
54	PC.6	I/O	MFP0	General purpose digital I/O pin.
	TM2_EXT	I	MPF1	Timer2 external counter input

	SPI0_MISO0	I/O	MPF4	1st SPI0 MISO (Master In, Slave Out) pin.
	TM2_CNT_OUT	I/O	MPF5	Timer2 event counter input/toggle output.
	EBI_AD9	O	MPF7	EBI address/data bus bit 9.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
55	PC.7	I/O	MFP0	General purpose digital I/O pin.
	TM1_EXT	I	MPF1	Timer1 external counter input
	SPI0_MOSI0	I/O	MPF4	1st SPI0 MOSI (Master Out, Slave In) pin.
	EBI_AD8	O	MPF7	EBI address/data bus bit 8.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
56	PC.8	I/O	MFP0	General purpose digital I/O pin.
	TM0_EXT	I	MPF1	Timer0 external counter input
	SPI0_CLK	O	MPF4	SPI0 serial clock pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
57	LDO_CAP	P	MFP0	LDO output pin.
58	PE.4	I/O	MFP0	General purpose digital I/O pin.
	ADC0_4	A	MPF1	ADC0 analog input.
	ACMP0_P2	A	MPF2	Analog comparator0 positive input pin.
	SPI0_SS0	I/O	MPF3	General purpose digital I/O pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
59	PE.5	I/O	MFP0	General purpose digital I/O pin.
	ADC0_5	A	MPF1	ADC0 analog input.
	ACMP0_P1	A	MPF2	Analog comparator0 positive input pin.
	SPI0_CLK	O	MPF3	SPI0 serial clock pin.
	SD0_CDn	I	MPF4	SD mode #0 – card detect
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

60	PE.6	I/O	MFP0	General purpose digital I/O pin.
	ADC0_6	A	MPF1	ADC0 analog input.
	ACMP0_P0	A	MPF2	Analog comparator0 positive input pin.
	SPI0_MISO0	I/O	MPF3	1st SPI0 MISO (Master In, Slave Out) pin.
	SD0_CMD	I/O	MPF4	SD mode #0 – command/response
	EBI_nWR	O	MPF7	EBI write enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
61	PE.7	I/O	MFP0	General purpose digital I/O pin.
	ADC0_7	A	MPF1	ADC0 analog input.
	ACMP0_N	A	MPF2	Analog comparator0 negative input pin.
	SPI0_MOSI0	I/O	MPF3	1st SPI0 MOSI (Master Out, Slave In) pin.
	SD0_CLK	O	MPF4	SD mode #0 – clock.
	EBI_nRD	O	MPF7	EBI read enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
62	AVSS	P	MFP0	Ground pin for digital circuit.
63	Vref	A	MFP0	Voltage reference input for ADC.
64	AVDD	P	MFP0	Power supply for internal analog circuit.

Note: Pin Type I = Digital Input, O = Digital Output; A = Analog Pin; P = Power Pin;

4.3.2 NuMicro™ NUC442 Package LQFP 100-pin Description

MFP = Multi-function pin.

Pin No.	Pin Name	Type	MFP*	Description
1	PE.12	I/O	MFP0	General purpose digital I/O pin.
	ADC1_4	A	MPF1	ADC1 analog input.
	ACMP1_P3	A	MPF2	Analog comparator1 positive input pin.
	ACMP2_P2	A	MPF3	Analog comparator2 positive input pin.
	EBI_nCS1	O	MPF7	EBI chip select 1 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
2	PE.13	I/O	MFP0	General purpose digital I/O pin.
	ADC1_5	A	MPF1	ADC1 analog input.
	ACMP2_P1	A	MPF3	Analog comparator2 positive input pin.
	EBI_nCS2	O	MPF7	EBI chip select 2 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
3	PE.14	I/O	MFP0	General purpose digital I/O pin.
	ADC1_6	A	MPF1	ADC1 analog input.
	ACMP2_P0	A	MPF3	Analog comparator2 positive input pin.
	EBI_nCS3	O	MPF7	EBI chip select 3 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
4	PE.15	I/O	MFP0	General purpose digital I/O pin.
	ADC1_7	A	MPF1	ADC1 analog input.
	ACMP2_N	A	MPF3	Analog comparator2 negative input pin.
5	VSS	P	MFP0	Ground pin for digital circuit.
6	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
7	PC.12	I/O	MFP0	General purpose digital I/O pin.
	SPI1_SS0	I/O	MPF1	1st SPI1 slave select pin..

	SC4_CD	I	MPF2	SmartCard4 card detect pin.
	SD1_CDn	I	MPF4	SD mode #1 – card detect
	CAP_DATA7	I	MPF5	Image data input bus bit 7.
	EBI_A0	O	MPF7	EBI address bus bit0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
8	PC.13	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MOSI1	I/O	MPF1	2nd SPI1 MOSI (Master Out, Slave In) pin.
	SC4_RST	O	MPF2	SmartCard4 reset pin.
	SD1_CMD	I/O	MPF4	SD mode #1 – command/response
	CAP_DATA6	I	MPF5	Image data input bus bit 7.
	EBI_A1	O	MPF7	EBI address bus bit1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
9	PC.14	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MISO1	I/O	MPF1	2nd SPI1 MISO (Master In, Slave Out) pin.
	SC4_PWR	O	MPF2	SmartCard4 power pin.
	TM3_EXT	I	MPF3	Timer3 external counter input
	SD1_CLK	O	MPF4	SD mode #1 – clock.
	CAP_DATA5	I	MPF5	Image data input bus bit 7.
	EBI_A2	O	MPF7	EBI address bus bit2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
10	PC.15	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MOSI0	I/O	MPF1	1st SPI1 MOSI (Master Out, Slave In) pin.
	SC4_DAT	I/O	MPF2	SmartCard4 data pin.
	SD1_DAT3	I/O	MPF4	SD mode #1 data line bit 3.
	CAP_DATA4	I	MPF5	Image data input bus bit 7.
	EBI_A3	O	MPF7	EBI address bus bit3.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
11	PD.0	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MISO0	I/O	MPF1	1st SPI1 MISO (Master In, Slave Out) pin.
	SC4_CLK	O	MPF2	SmartCard4 clock pin.
	SD1_DAT2	I/O	MPF4	SD mode #1 data line bit 2.
	CAP_DATA3	I	MPF5	Image data input bus bit 7.
	EBI_A4	O	MPF7	EBI address bus bit4.
	INT3	I	MPF8	External interrupt3 input pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
12	PD.1	I/O	MFP0	General purpose digital I/O pin.
	SPI1_CLK	O	MPF1	SPI1 serial clock pin.
	TM0_CNT_OUT	I/O	MPF3	Timer0 event counter input/toggle output.
	SD1_DAT1	I/O	MPF4	SD mode #1 data line bit 1.
	CAP_DATA2	I	MPF5	Image data input bus bit 7.
	EBI_A5	O	MPF7	EBI address bus bit5.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
13	PD.3	I/O	MFP0	General purpose digital I/O pin.
	SC5_CLK	O	MPF1	SmartCard5 clock pin.
	I2C3_SDA	I/O	MPF2	I2C3 data input/output pin.
	ACMP2_O	O	MPF3	Analog ccomparator2 output.
	SD0_CDn	I	MPF4	SD mode #0 – card detect
	CAP_DATA0	I	MPF5	Image data input bus bit 7.
	EBI_A7	O	MPF7	EBI address bus bit7.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
14	PD.4	I/O	MFP0	General purpose digital I/O pin.

	SC5_CD	I	MPF1	SmartCard5 card detect pin.
	UART3_RXD	I	MPF2	Data receiver input pin for UART3.
	ACMP1_O	O	MPF3	Analog ccomparator1 output.
	CAP_SCLK	O	MPF5	Image capture interface sensor clock pin.
	EBI_A8	O	MPF7	EBI address bus bit8.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
15	PD.5	I/O	MFP0	General purpose digital I/O pin.
	SC5_RST	O	MPF1	SmartCard5 reset pin.
	UART3_TXD	O	MPF2	Data transmitter output pin for UART3.
	CAP_VSYNC	I	MPF5	Image capture interface VSYNC input pin.
	EBI_A9	O	MPF7	EBI address bus bit9.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
16	PD.6	I/O	MFP0	General purpose digital I/O pin.
	SC5_PWR	O	MPF1	SmartCard5 power pin.
	UART3_RTS	O	MPF2	Request to Send output pin for UART3.
	SD0_CMD	I/O	MPF4	SD mode #0 – command/response
	CAP_HSYNC	I	MPF5	Image capture interface HSYNC input pin.
	EBI_A10	O	MPF7	EBI address bus bit10.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
17	PD.7	I/O	MFP0	General purpose digital I/O pin.
	SC5_DAT	I/O	MPF1	SmartCard5 data pin.
	UART3_CTS	I	MPF2	Clear to Send input pin for UART3.
	SD0_CLK	O	MPF4	SD mode #0 – clock.
	CAP_PIXCLK	I	MPF5	Image capture interface pix clock input pin.
	EBI_A11	O	MPF7	EBI address bus bit11.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
18	PG.13	I/O	MFP0	General purpose digital I/O pin.
	XT1_IN	I	MPF1	External 4~24 MHz (high-speed) crystal input pin.
19	PG.12	I/O	MFP0	General purpose digital I/O pin.
	XT1_OUT	O	MPF1	External 4~24 MHz (high-speed) crystal output pin.
20	nRESET	I	MFP0	External reset input: active LOW, with an internal pull-up. Set this pin low reset to initial state.
21	LDO_CAP	P	MFP0	LDO output pin.
22	VSS	P	MFP0	Ground pin for digital circuit.
23	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
24	PG.10	I/O	MFP0	General purpose digital I/O pin.
	ICE_CLK	I	MPF1	Serial wired debugger clock pin
25	PG.11	I/O	MFP0	General purpose digital I/O pin.
	ICE_DAT	I/O	MPF1	Serial wired debugger data pin
26	PG.15	I/O	MFP0	General purpose digital I/O pin.
	X32K_IN	I	MPF1	External 32.768 kHz (low-speed) crystal input pin.
	I2C1_SCL	I/O	MPF3	I2C1 clock pin.
27	PG.14	I/O	MFP0	General purpose digital I/O pin.
	X32K_OUT	O	MPF1	External 32.768 kHz (low-speed) crystal output pin.
	I2C1_SDA	I/O	MPF3	I2C1 data input/output pin.
28	VBAT	P	MFP0	Battery power input pin.
29	PA.0	I/O	MFP0	General purpose digital I/O pin.
	TAMPER0	I/O	MPF1	Tamper detect pin 0.
	SC0_CD	I	MPF2	SmartCard0 card detect pin.
	CAN1_RXD	I	MPF3	CAN bus receiver1 input.
	INT0	I	MPF8	External interrupt0 input pin.

30	PA.1	I/O	MFP0	General purpose digital I/O pin.
	TAMPER1	I/O	MPF1	Tamper detect pin 1.
	SC5_CD	I	MPF2	SmartCard5 card detect pin.
	CAN1_TXD	I	MPF3	CAN bus transmitter1 input.
	EBI_A22	O	MPF7	EBI address bus bit22.
31	PA.2	I/O	MFP0	General purpose digital I/O pin.
	SC2_DAT	I/O	MPF1	SmartCard2 data pin.
	SPI3_MISO0	I/O	MPF2	1st SPI3 MISO (Master In, Slave Out) pin.
	I2S0_MCLK	O	MPF3	I2S0 master clock output pin.
	BRAKE11	I	MPF4	Brake input pin 1 of EPWM0.
	CAP_SFIELD	I	MPF5	Video input interface SFIELD input pin.
	EBI_A12	O	MPF7	EBI address bus bit12.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
32	PA.3	I/O	MFP0	General purpose digital I/O pin.
	SC2_CLK	O	MPF1	SmartCard2 clock pin.
	SPI3_MOSI0	I/O	MPF2	1st SPI3 MOSI (Master Out, Slave In) pin.
	I2S0_DO	O	MPF3	I2S0 data output.
	BRAKE10	I	MPF4	Brake input pin 0 of EPWM0.
	EBI_A13	O	MPF7	EBI address bus bit13.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
33	PA.4	I/O	MFP0	General purpose digital I/O pin.
	SC2_PWR	O	MPF1	SmartCard2 power pin.
	SPI3_CLK	O	MPF2	SPI3 serial clock pin.
	I2S0_DI	I	MPF3	I2S0 data input.
	QEI1_Z	I	MPF5	Quadrature encoder phase Z input of QEI Unit 1.
	EBI_A14	O	MPF7	EBI address bus bit14.

	ECAP1_IC2	I	MPF8	Input 2 of enhanced capture unit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
34	PA.5	I/O	MFP0	General purpose digital I/O pin.
	SC2_RST	O	MPF1	SmartCard2 reset pin.
	SPI3_SS0	I/O	MPF2	General purpose digital I/O pin.
	I2S0_BCLK	O	MPF3	I2S0 bit clock pin.
	PWM0_0	I/O	MPF4	PWM0_0 output/capture input.
	QEI1_B	I	MPF5	Quadrature encoder phase B input of QEI Unit 1.
	EBI_A15	O	MPF7	EBI address bus bit15.
	ECAP1_IC1	I	MPF8	Input 1 of enhanced capture unit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
35	PA.6	I/O	MFP0	General purpose digital I/O pin.
	SC2_CD	I	MPF1	SmartCard2 card detect pin.
	I2S0_LRCK	O	MPF3	I2S0 left right channel clock.
	PWM0_1	I/O	MPF4	PWM0_1 output/capture input.
	QEI1_A	I	MPF5	Quadrature encoder phase A input of QEI Unit 1.
	CAN1_TXD	I	MPF6	CAN bus transmitter1 input.
	EBI_A16	O	MPF7	EBI address bus bit16.
	ECAP1_IC0	I	MPF8	Input 0 of enhanced capture unit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
36	PA.7	I/O	MFP0	General purpose digital I/O pin.
	SC0_CLK	O	MPF2	SmartCard0 clock pin.
	SPI3_SS0	I/O	MPF3	General purpose digital I/O pin.
	PWM1_3	I/O	MPF4	PWM1_3 output/capture input.
	EPWM0_5	I/O	MPF5	PWM0_5 output/capture input.
	EBI_A17	O	MPF7	EBI address bus bit17.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
37	PA.8	I/O	MFP0	General purpose digital I/O pin.
	SC0_RST	O	MPF2	SmartCard0 reset pin.
	SPI3_CLK	O	MPF3	SPI3 serial clock pin.
	PWM1_2	I/O	MPF4	PWM1_2 output/capture input.
	EPWM0_4	I/O	MPF5	PWM0_4 output/capture input.
	EBI_A18	O	MPF7	EBI address bus bit18.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
38	PA.9	I/O	MFP0	General purpose digital I/O pin.
	SC0_PWR	O	MPF2	SmartCard0 power pin.
	SPI3_MISO0	I/O	MPF3	1st SPI3 MISO (Master In, Slave Out) pin.
	PWM1_1	I/O	MPF4	PWM1_1 output/capture input.
	EPWM0_3	I/O	MPF5	PWM0_3 output/capture input.
	EBI_A19	O	MPF7	EBI address bus bit19.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
39	PA.10	I/O	MFP0	General purpose digital I/O pin.
	SC0_DAT	I/O	MPF2	SmartCard0 data pin.
	SPI3_MOSI0	I/O	MPF3	1st SPI3 MOSI (Master Out, Slave In) pin.
	PWM1_0	I/O	MPF4	PWM1_0 output/capture input.
	EPWM0_2	I/O	MPF5	PWM0_2 output/capture input.
	EBI_A20	O	MPF7	EBI address bus bit20.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
40	PA.11	I/O	MFP0	General purpose digital I/O pin.
	UART0_RTS	O	MPF1	Request to Send output pin for UART0.
	SPI3_MISO1	I/O	MPF3	2nd SPI3 MISO (Master In, Slave Out) pin.

	PWM0_5	I/O	MPF4	PWM0_5 output/capture input.
	EPWM0_1	I/O	MPF5	PWM0_1 output/capture input.
	EBI_AD0	O	MPF7	EBI address/data bus bit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
41	PA.12	I/O	MFP0	General purpose digital I/O pin.
	UART0_CTS	I	MPF1	Clear to Send input pin for UART0.
	SPI3_MOSI1	I/O	MPF3	2nd SPI3 MOSI (Master Out, Slave In) pin.
	PWM0_4	I/O	MPF4	PWM0_4 output/capture input.
	EPWM0_0	I/O	MPF5	PWM0_0 output/capture input.
	EBI_AD1	O	MPF7	EBI address/data bus bit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
42	PA.13	I/O	MFP0	General purpose digital I/O pin.
	UART0_RXD	I	MPF1	Data receiver input pin for UART0.
	SC3_DAT	I/O	MPF3	SmartCard3 data pin.
	PWM1_4	I/O	MPF4	PWM1_4 output/capture input.
	EBI_AD2	O	MPF7	EBI address/data bus bit 2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
43	PA.14	I/O	MFP0	General purpose digital I/O pin.
	UART0_TXD	O	MPF1	Data transmitter output pin for UART0.
	SC3_CLK	O	MPF3	SmartCard3 clock pin.
	PWM1_5	I/O	MPF4	PWM1_5 output/capture input.
	EBI_AD3	O	MPF7	EBI address/data bus bit 3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
44	PA.15	I/O	MFP0	General purpose digital I/O pin.
	SC3_PWR	O	MPF1	SmartCard3 power pin.

	UART2_RTS	O	MPF2	Request to Send output pin for UART2.
	I2C0_SCL	I/O	MPF4	I2C0 clock pin.
	EBI_A21	O	MPF7	EBI address bus bit21.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
45	PC.9	I/O	MFP0	General purpose digital I/O pin.
	STADC	A	MPF1	ADC analog input.
	UART2_CTS	I	MPF2	Clear to Send input pin for UART2.
	SC3_RST	O	MPF3	SmartCard3 reset pin.
	I2C0_SDA	I/O	MPF4	I2C0 data input/output pin.
	CAP_DATA1	I	MPF5	Image data input bus bit 7.
	I2C3_SCL	I/O	MPF6	I2C3 clock pin.
	EBI_A22	O	MPF7	EBI address bus bit22.
	SD1_DAT0	I/O	MPF8	SD mode #1 data line bit 0.
	EBI_A6	O	MPF9	EBI address bus bit6.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
46	PC.10	I/O	MFP0	General purpose digital I/O pin.
	SC3_CD	I	MPF1	SmartCard3 card detect pin.
	UART2_RXD	I	MPF2	Data receiver input pin for UART2.
	PWM0_2	I/O	MPF4	PWM0_2 output/capture input.
	EBI_A23	O	MPF6	EBI address bus bit23.
	EBI_AD2	O	MPF7	EBI address/data bus bit 2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
47	PC.11	I/O	MFP0	General purpose digital I/O pin.
	UART2_TXD	O	MPF2	Data transmitter output pin for UART2.
	PWM0_3	I/O	MPF4	PWM0_3 output/capture input.
	EBI_A24	O	MPF6	EBI address bus bit24.

	EBI_AD3	O	MPF7	EBI address/data bus bit 3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
48	LDO_CAP	P	MFP0	LDO output pin.
49	VSS	P	MFP0	Ground pin for digital circuit.
50	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
51	VRES	A	MFP0	USB PHY VRES ground input pin. Add an 8.2K ohm resistor to VSSA.
52	VBUS	A	MFP0	USB PHY VBUS power input pin.
53	USB_VDD33_CAP	P	MFP0	Internal power regulator output 3.3V decoupling pin.
54	VSSA	P	MFP0	Ground pin for digital circuit. Add a Feritte Bead to digital ground VSS.
55	USB0_D-	A	MFP0	USB0 differential signal D+.
56	USB0_D+	A	MFP0	USB0 differential signal D+.
57	USB0_OTG_ID	I	MFP0	USB0OTG ID pin.
58	PB.0	I/O	MFP0	General purpose digital I/O pin.
	USB0_VBUS_ST	I	MPF1	USB0 external VBUS regulator status
	I2C4_SCL	I/O	MPF2	I2C4 clock pin.
	INT1	I	MPF8	External interrupt1 input pin.
59	PB.1	I/O	MFP0	General purpose digital I/O pin.
	USB0_VBUS_EN	O	MPF1	USB0 external VBUS regulator enable
	I2C4_SDA	I/O	MPF2	I2C4 data input/output pin.
	TM1_CNT_OUT	I/O	MPF3	Timer1 event counter input/toggle output.
60	PB.2	I/O	MFP0	General purpose digital I/O pin.
	UART1_RXD	I	MPF1	Data receiver input pin for UART1.
	SPI2_SS0	I/O	MPF2	General purpose digital I/O pin.
	USB1_D-	A	MPF3	USB1 differential signal D+.
	EBI_AD4	O	MPF7	EBI address/data bus bit 4.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
61	PB.3	I/O	MFP0	General purpose digital I/O pin.
	UART1_TXD	O	MPF1	Data transmitter output pin for UART1.
	SPI2_CLK	O	MPF2	SPI2 serial clock pin.
	USB1_D+	A	MPF3	USB1 differential signal D+.
	EBI_AD5	O	MPF7	EBI address/data bus bit 5.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
62	PB.4	I/O	MFP0	General purpose digital I/O pin.
	UART1_RTS	O	MPF1	Request to Send output pin for UART1.
	SPI2_MISO0	I/O	MPF2	1st SPI2 MISO (Master In, Slave Out) pin.
	UART4_RXD	I	MPF3	Data receiver input pin for UART4.
	TM0_CNT_OUT	I/O	MPF4	Timer0 event counter input/toggle output.
	EBI_AD6	O	MPF7	EBI address/data bus bit 6.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
63	PB.5	I/O	MFP0	General purpose digital I/O pin.
	UART1_CTS	I	MPF1	Clear to Send input pin for UART1.
	SPI2_MOSI0	I/O	MPF2	1st SPI2 MOSI (Master Out, Slave In) pin.
	UART4_TXD	O	MPF3	Data transmitter output pin for UART4.
	EBI_AD7	O	MPF7	EBI address/data bus bit 7.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
64	PB.6	I/O	MFP0	General purpose digital I/O pin.
	I2C2_SCL	I/O	MPF1	I2C2 clock pin.
	BRAKE01	I	MPF2	Brake input pin 1 of EPWMB.
	UART4_RTS	O	MPF3	Request to Send output pin for UART4.
	PWM1_4	I/O	MPF4	PWM1_4 output/capture input.

	EPWM1_0	I/O	MPF5	PWM1_0 output/capture input.
	EBI_AD8	O	MPF7	EBI address/data bus bit 8.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
65	PB.7	I/O	MFP0	General purpose digital I/O pin.
	I2C2_SDA	I/O	MPF1	I2C2 data input/output pin.
	BRAKE00	I	MPF2	Brake input pin 0 of EPWMB.
	UART4_CTS	I	MPF3	Clear to Send input pin for UART4.
	PWM1_5	I/O	MPF4	PWM1_5 output/capture input.
	EPWM1_1	I/O	MPF5	PWM1_1 output/capture input.
	EBI_AD9	O	MPF7	EBI address/data bus bit 9.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
66	PB.8	I/O	MFP0	General purpose digital I/O pin.
	UART5_CTS	I	MPF1	Clear to Send input pin for UART5.
	EPWM1_2	I/O	MPF5	PWM1_2 output/capture input.
	EBI_AD10	O	MPF7	EBI address/data bus bit 10.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
67	PB.9	I/O	MFP0	General purpose digital I/O pin.
	UART5_RTS	O	MPF1	Request to Send output pin for UART5.
	EPWM1_3	I/O	MPF5	PWM1_3 output/capture input.
	EBI_AD11	O	MPF7	EBI address/data bus bit 11.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
68	PB.10	I/O	MFP0	General purpose digital I/O pin.
	UART5_TXD	O	MPF1	Data transmitter output pin for UART5.
	EPWM1_4	I/O	MPF5	PWM1_4 output/capture input.
	EBI_AD12	O	MPF7	EBI address/data bus bit 12.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
69	PB.11	I/O	MFP0	General purpose digital I/O pin.
	UART5_RXD	I	MPF1	Data receiver input pin for UART5.
	EPWM1_5	I/O	MPF5	PWM1_5 output/capture input.
	EBI_AD13	O	MPF7	EBI address/data bus bit 13.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
70	PB.12	I/O	MFP0	General purpose digital I/O pin.
	UART4_RTS	O	MPF1	Request to Send output pin for UART4.
	SPI2_MISO1	I/O	MPF2	2nd SPI2 MISO (Master In, Slave Out) pin.
	CAN0_RXD	I	MPF3	CAN bus receiver0 input.
	EBI_AD14	O	MPF7	EBI address/data bus bit 14.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
71	PB.13	I/O	MFP0	General purpose digital I/O pin.
	UART4_CTS	I	MPF1	Clear to Send input pin for UART4.
	SPI2_MOSI1	I/O	MPF2	2nd SPI2 MOSI (Master Out, Slave In) pin.
	CAN0_TXD	I	MPF3	CAN bus transmitter0 input.
	EBI_AD15	O	MPF7	EBI address/data bus bit 15.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
72	PB.14	I/O	MFP0	General purpose digital I/O pin.
	I2S1_MCLK	O	MPF1	I2S1 master clock output pin.
	SC1_RST	O	MPF2	SmartCard1 reset pin.
	BRAKE01	I	MPF4	Brake input pin 1 of EPWMB.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
73	PB.15	I/O	MFP0	General purpose digital I/O pin.
	I2S1_DO	O	MPF1	I2S1 data output.

	SC1_DAT	I/O	MPF2	SmartCard1 data pin.
	BRAKE00	I	MPF4	Brake input pin 0 of EPWMB.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
74	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
75	PC.0	I/O	MFP0	General purpose digital I/O pin.
	I2S1_DI	I	MPF1	I2S1 data input.
	SC1_DAT	I/O	MPF2	SmartCard1 data pin.
	UART4_RXD	I	MPF3	Data receiver input pin for UART4.
	EBI_MCLK	O	MPF7	EBI interface clock output pin.
	INT2	I	MPF8	External interrupt2 input pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
76	PC.1	I/O	MFP0	General purpose digital I/O pin.
	I2S1_BCLK	O	MPF1	I2S1 bit clock pin.
	SC1_CLK	O	MPF2	SmartCard1 clock pin.
	UART4_TXD	O	MPF3	Data transmitter output pin for UART4.
	TM3_CNT_OUT	I/O	MPF5	Timer3 event counter input/toggle output.
	EBI_AD13	O	MPF7	EBI address/data bus bit 13.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
77	PC.2	I/O	MFP0	General purpose digital I/O pin.
	I2S1_LRCK	O	MPF1	I2S1 left right channel clock.
	SC1_PWR	O	MPF2	SmartCard1 power pin.
	UART4_RTS	O	MPF3	Request to Send output pin for UART4.
	SPI0_SS0	I/O	MPF4	General purpose digital I/O pin.
	EBI_AD12	O	MPF7	EBI address/data bus bit 12.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

78	PC.3	I/O	MFP0	General purpose digital I/O pin.
	I2S1_MCLK	O	MPF1	I2S1 master clock output pin.
	SC1_CD	I	MPF2	SmartCard1 card detect pin.
	UART4_CTS	I	MPF3	Clear to Send input pin for UART4.
	SPI0_MISO1	I/O	MPF4	2nd SPI0 MISO (Master In, Slave Out) pin.
	QEI0_Z	I	MPF5	Quadrature encoder phase Z input of QEI Unit 0.
	EBI_AD11	O	MPF7	EBI address/data bus bit 11.
	ECAP0_IC2	O	MPF8	Input 2 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
79	PC.4	I/O	MFP0	General purpose digital I/O pin.
	I2S1_DO	O	MPF1	I2S1 data output.
	SC1_RST	O	MPF2	SmartCard1 reset pin.
	SPI0_MOSI1	I/O	MPF4	2nd SPI0 MOSI (Master Out, Slave In) pin.
	QEI0_B	I	MPF5	Quadrature encoder phase B input of QEI Unit 0.
	EBI_AD10	O	MPF7	EBI address/data bus bit 10.
	ECAP0_IC1	O	MPF8	Input 1 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
80	PC.5	I/O	MFP0	General purpose digital I/O pin.
	CLKO	O	MFP1	Clock Output Pin.
	QEI0_A	I	MPF5	Quadrature encoder phase A input of QEI Unit 0.
	EBI_MCLK	O	MPF7	EBI interface clock output pin.
	ECAP0_IC0	O	MPF8	Input 0 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
81	PC.6	I/O	MFP0	General purpose digital I/O pin.
	TM2_EXT	I	MPF1	Timer2 external counter input
	SPI0_MISO0	I/O	MPF4	1st SPI0 MISO (Master In, Slave Out) pin.

	TM2_CNT_OUT	I/O	MPF5	Timer2 event counter input/toggle output.
	EBI_AD9	O	MPF7	EBI address/data bus bit 9.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
82	PC.7	I/O	MFP0	General purpose digital I/O pin.
	TM1_EXT	I	MPF1	Timer1 external counter input
	SPI0_MOSI0	I/O	MPF4	1st SPI0 MOSI (Master Out, Slave In) pin.
	EBI_AD8	O	MPF7	EBI address/data bus bit 8.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
83	PC.8	I/O	MFP0	General purpose digital I/O pin.
	TM0_EXT	I	MPF1	Timer0 external counter input
	SPI0_CLK	O	MPF4	SPI0 serial clock pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
84	LDO_CAP	P	MFP0	LDO output pin.
85	VSS	P	MFP0	Ground pin for digital circuit.
86	PE.0	I/O	MFP0	General purpose digital I/O pin.
	ADC0_0	A	MPF1	ADC0 analog input.
	INT4	I	MPF8	External interrupt4 input pin.
87	PE.1	I/O	MFP0	General purpose digital I/O pin.
	ADC0_1	A	MPF1	ADC0 analog input.
	TM2_CNT_OUT	I/O	MPF3	Timer2 event counter input/toggle output.
88	PE.2	I/O	MFP0	General purpose digital I/O pin.
	ADC0_2	A	MPF1	ADC0 analog input.
	ACMP0_O	O	MPF2	Analog ccomparator0 output .
	SPI0_MISO0	I/O	MPF3	1st SPI0 MISO (Master In, Slave Out) pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

89	PE.3	I/O	MFP0	General purpose digital I/O pin.
	ADC0_3	A	MPF1	ADC0 analog input.
	ACMP0_P3	A	MPF2	Analog comparator0 positive input pin.
	SPI0_MOSI0	I/O	MPF3	1st SPI0 MOSI (Master Out, Slave In) pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
90	PE.4	I/O	MFP0	General purpose digital I/O pin.
	ADC0_4	A	MPF1	ADC0 analog input.
	ACMP0_P2	A	MPF2	Analog comparator0 positive input pin.
	SPI0_SS0	I/O	MPF3	General purpose digital I/O pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
91	PE.5	I/O	MFP0	General purpose digital I/O pin.
	ADC0_5	A	MPF1	ADC0 analog input.
	ACMP0_P1	A	MPF2	Analog comparator0 positive input pin.
	SPI0_CLK	O	MPF3	SPI0 serial clock pin.
	SD0_CDn	I	MPF4	SD mode #0 – card detect
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
92	PE.6	I/O	MFP0	General purpose digital I/O pin.
	ADC0_6	A	MPF1	ADC0 analog input.
	ACMP0_P0	A	MPF2	Analog comparator0 positive input pin.
	SPI0_MISO0	I/O	MPF3	1st SPI0 MISO (Master In, Slave Out) pin.
	SD0_CMD	I/O	MPF4	SD mode #0 – command/response
	EBI_nWR	O	MPF7	EBI write enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
93	PE.7	I/O	MFP0	General purpose digital I/O pin.
	ADC0_7	A	MPF1	ADC0 analog input.

	ACMP0_N	A	MPF2	Analog comparator0 negative input pin.
	SPI0_MOSI0	I/O	MPF3	1st SPI0 MOSI (Master Out, Slave In) pin.
	SD0_CLK	O	MPF4	SD mode #0 – clock.
	EBI_nRD	O	MPF7	EBI read enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
94	AVSS	P	MFP0	Ground pin for digital circuit.
95	Vref	A	MFP0	Voltage reference input for ADC.
96	AVDD	P	MFP0	Power supply for internal analog circuit.
97	PE.8	I/O	MFP0	General purpose digital I/O pin.
	ADC1_0	A	MPF1	ADC1 analog input.
	ADC0_8	A	MPF1	ADC0 analog input.
	ACMP1_N	A	MPF2	Analog comparator1 negative input pin.
	TM1_CNT_OUT	I/O	MPF3	Timer1 event counter input/toggle output.
	SD0_DAT3	I/O	MPF4	SD mode #0 data line bit 3.
	EBI_ALE	O	MPF7	EBI address latch enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
98	PE.9	I/O	MFP0	General purpose digital I/O pin.
	ADC1_1	A	MPF1	ADC1 analog input.
	ADC0_9	A	MPF1	ADC0 analog input.
	ACMP1_P0	A	MPF2	Analog comparator1 positive input pin.
	SD0_DAT2	I/O	MPF4	SD mode #0 data line bit 2.
	EBI_nWRH	O	MPF7	EBI write enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
99	PE.10	I/O	MFP0	General purpose digital I/O pin.
	ADC1_2	A	MPF1	ADC1 analog input.
	ADC0_10	A	MPF1	ADC0 analog input.

	ACMP1_P1	A	MPF2	Analog comparator1 positive input pin.
	SPI0_MISO1	I/O	MPF3	2nd SPI0 MISO (Master In, Slave Out) pin.
	SD0_DAT1	I/O	MPF4	SD mode #0 data line bit 1.
	EBI_nWRL	O	MPF7	EBI write enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
100	PE.11	I/O	MFP0	General purpose digital I/O pin.
	ADC1_3	A	MPF1	ADC1 analog input.
	ADC0_11	A	MPF1	ADC0 analog input.
	ACMP1_P2	A	MPF2	Analog comparator1 positive input pin.
	SPI0_MOSI1	I/O	MPF3	2nd SPI0 MOSI (Master Out, Slave In) pin.
	SD0_DAT0	I/O	MPF4	SD mode #0 data line bit 0.
	ACMP2_P3	A	MPF5	Analog comparator2 positive input pin.
	EBI_nCS0	O	MPF7	EBI chip select 0 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

Note: Pin Type I = Digital Input, O = Digital Output; A = Analog Pin; P = Power Pin;

4.3.3 NuMicro™ NUC442 Package LQFP 128-pin Description

MFP = Multi-function pin.

Pin No.	Pin Name	Type	MFP*	Description
1	PE.12	I/O	MFP0	General purpose digital I/O pin.
	ADC1_4	A	MPF1	ADC1 analog input.
	ACMP1_P3	A	MPF2	Analog comparator1 positive input pin.
	ACMP2_P2	A	MPF3	Analog comparator2 positive input pin.
	EBI_nCS1	O	MPF7	EBI chip select 1 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
2	PE.13	I/O	MFP0	General purpose digital I/O pin.
	ADC1_5	A	MPF1	ADC1 analog input.
	ACMP2_P1	A	MPF3	Analog comparator2 positive input pin.
	EBI_nCS2	O	MPF7	EBI chip select 2 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
3	PE.14	I/O	MFP0	General purpose digital I/O pin.
	ADC1_6	A	MPF1	ADC1 analog input.
	ACMP2_P0	A	MPF3	Analog comparator2 positive input pin.
	EBI_nCS3	O	MPF7	EBI chip select 3 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
4	PE.15	I/O	MFP0	General purpose digital I/O pin.
	ADC1_7	A	MPF1	ADC1 analog input.
	ACMP2_N	A	MPF3	Analog comparator2 negative input pin.
5	PF.9	I/O	MFP0	General purpose digital I/O pin.
	OPA0_IN+	I/O	MPF1	General purpose digital I/O pin.
	PWM0_0	I/O	MPF4	PWM0_0 output/capture input.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

6	PF.10	I/O	MFP0	General purpose digital I/O pin.
	OPA0_IN-	I/O	MPF1	General purpose digital I/O pin.
	PWM0_1	I/O	MPF4	PWM0_1 output/capture input.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
7	PF.11	I/O	MFP0	General purpose digital I/O pin.
	OPA0_O	O	MPF1	Operational amplifier output pin
	UART1_RTS	O	MPF2	Request to Send output pin for UART1.
8	PF.12	I/O	MFP0	General purpose digital I/O pin.
	OPA1_IN+	I/O	MPF1	General purpose digital I/O pin.
	UART1_CTS	I	MPF2	Clear to Send input pin for UART1.
9	PF.13	I/O	MFP0	General purpose digital I/O pin.
	OPA1_IN-	I/O	MPF1	General purpose digital I/O pin.
	UART1_TXD	O	MPF2	Data transmitter output pin for UART1.
10	PF.14	I/O	MFP0	General purpose digital I/O pin.
	OPA1_O	O	MPF1	Operational amplifier output pin
	UART1_RXD	I	MPF2	Data receiver input pin for UART1.
11	VSS	P	MFP0	Ground pin for digital circuit.
12	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
13	PC.12	I/O	MFP0	General purpose digital I/O pin.
	SPI1_SS0	I/O	MPF1	1st SPI1 slave select pin..
	SC4_CD	I	MPF2	SmartCard4 card detect pin.
	SD1_CDn	I	MPF4	SD mode #1 – card detect
	CAP_DATA7	I	MPF5	Image data input bus bit 7.
	EBI_A0	O	MPF7	EBI address bus bit0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
14	PC.13	I/O	MFP0	General purpose digital I/O pin.

	SPI1_MOSI1	I/O	MPF1	2nd SPI1 MOSI (Master Out, Slave In) pin.
	SC4_RST	O	MPF2	SmartCard4 reset pin.
	SD1_CMD	I/O	MPF4	SD mode #1 – command/response
	CAP_DATA6	I	MPF5	Image data input bus bit 7.
	EBI_A1	O	MPF7	EBI address bus bit1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
15	PC.14	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MISO1	I/O	MPF1	2nd SPI1 MISO (Master In, Slave Out) pin.
	SC4_PWR	O	MPF2	SmartCard4 power pin.
	TM3_EXT	I	MPF3	Timer3 external counter input
	SD1_CLK	O	MPF4	SD mode #1– clock.
	CAP_DATA5	I	MPF5	Image data input bus bit 7.
	EBI_A2	O	MPF7	EBI address bus bit2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
16	PC.15	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MOSI0	I/O	MPF1	1st SPI1 MOSI (Master Out, Slave In) pin.
	SC4_DAT	I/O	MPF2	SmartCard4 data pin.
	SD1_DAT3	I/O	MPF4	SD mode #1 data line bit 3.
	CAP_DATA4	I	MPF5	Image data input bus bit 7.
	EBI_A3	O	MPF7	EBI address bus bit3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
17	PD.0	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MISO0	I/O	MPF1	1st SPI1 MISO (Master In, Slave Out) pin.
	SC4_CLK	O	MPF2	SmartCard4 clock pin.
	SD1_DAT2	I/O	MPF4	SD mode #1 data line bit 2;
	CAP_DATA3	I	MPF5	Image data input bus bit 7.

	EBI_A4	O	MPF7	EBI address bus bit4.
	INT3	I	MPF8	External interrupt3 input pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
18	PD.1	I/O	MFP0	General purpose digital I/O pin.
	SPI1_CLK	O	MPF1	SPI1 serial clock pin.
	TM0_CNT_OUT	I/O	MPF3	Timer0 event counter inpu/toggle output.
	SD1_DAT1	I/O	MPF4	SD mode #1 data line bit 1.
	CAP_DATA2	I	MPF5	Image data input bus bit 7.
	EBI_A5	O	MPF7	EBI address bus bit5.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
19	PD.2	I/O	MFP0	General purpose digital I/O pin.
	STADC	A	MPF1	ADC analog input.
	I2C3_SCL	I/O	MPF2	I2C3 clock pin.
	SD1_DAT0	I/O	MPF4	SD mode #1 data line bit 0.
	CAP_DATA1	I	MPF5	Image data input bus bit 7.
	EBI_A6	O	MPF7	EBI address bus bit6.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
20	PD.3	I/O	MFP0	General purpose digital I/O pin.
	SC5_CLK	O	MPF1	SmartCard5 clock pin.
	I2C3_SDA	I/O	MPF2	I2C3 data input/output pin.
	ACMP2_O	O	MPF3	Analog ccomparator2 output .
	SD0_CDn	I	MPF4	SD mode #0 – card detect
	CAP_DATA0	I	MPF5	Image data input bus bit 7.
	EBI_A7	O	MPF7	EBI address bus bit7.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

21	PD.4	I/O	MFP0	General purpose digital I/O pin.
	SC5_CD	I	MPF1	SmartCard5 card detect pin.
	UART3_RXD	I	MPF2	Data receiver input pin for UART3.
	ACMP1_O	O	MPF3	Analog ccomparator1 output .
	CAP_SCLK	O	MPF5	Image capture interface sensor clock pin.
	EBI_A8	O	MPF7	EBI address bus bit8.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
22	PD.5	I/O	MFP0	General purpose digital I/O pin.
	SC5_RST	O	MPF1	SmartCard5 reset pin.
	UART3_TXD	O	MPF2	Data transmitter output pin for UART3.
	CAP_VSYNC	I	MPF5	Image capture interface VSYNC input pin.
	EBI_A9	O	MPF7	EBI address bus bit9.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
23	PD.6	I/O	MFP0	General purpose digital I/O pin.
	SC5_PWR	O	MPF1	SmartCard5 power pin.
	UART3_RTS	O	MPF2	Request to Send output pin for UART3.
	SD0_CMD	I/O	MPF4	SD mode #0 – command/response
	CAP_HSYNC	I	MPF5	Image capture interface HSYNC input pin.
	EBI_A10	O	MPF7	EBI address bus bit10.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
24	PD.7	I/O	MFP0	General purpose digital I/O pin.
	SC5_DAT	I/O	MPF1	SmartCard5 data pin.
	UART3_CTS	I	MPF2	Clear to Send input pin for UART3.
	SD0_CLK	O	MPF4	SD mode #0 – clock.
	CAP_PIXCLK	I	MPF5	Image capture interface pix clock input pin.
	EBI_A11	O	MPF7	EBI address bus bit11.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
25	PG.13	I/O	MFP0	General purpose digital I/O pin.
	XT1_IN	I	MPF1	External 4~24 MHz (high-speed) crystal input pin.
26	PG.12	I/O	MFP0	General purpose digital I/O pin.
	XT1_OUT	O	MPF1	External 4~24 MHz (high-speed) crystal output pin.
27	nRESET	I	MFP0	External reset input: active LOW, with an internal pull-up. Set this pin low reset to initial state.
28	LDO_CAP	P	MFP0	LDO output pin.
29	VSS	P	MFP0	Ground pin for digital circuit.
30	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
31	PG.10	I/O	MFP0	General purpose digital I/O pin.
	ICE_CLK	I	MPF1	Serial wired debugger clock pin
32	PG.11	I/O	MFP0	General purpose digital I/O pin.
	ICE_DAT	I/O	MPF1	Serial wired debugger data pin
33	PG.15	I/O	MFP0	General purpose digital I/O pin.
	X32K_IN	I	MPF1	External 32.768 kHz (low-speed) crystal input pin.
	I2C1_SCL	I/O	MPF3	I2C1 clock pin.
34	PG.14	I/O	MFP0	General purpose digital I/O pin.
	X32K_OUT	O	MPF1	External 32.768 kHz (low-speed) crystal output pin.
	I2C1_SDA	I/O	MPF3	I2C1 data input/output pin.
35	VBAT	P	MFP0	Battery power input pin.
36	PA.0	I/O	MFP0	General purpose digital I/O pin.
	TAMPER0	I/O	MPF1	Tamper detect pin 0.
	SC0_CD	I	MPF2	SmartCard0 card detect pin.
	CAN1_RXD	I	MPF3	CAN bus receiver1 input.
	INT0	I	MPF8	External interrupt0 input pin.

37	PA.1	I/O	MFP0	General purpose digital I/O pin.
	TAMPER1	I/O	MPF1	Tamper detect pin 1.
	SC5_CD	I	MPF2	SmartCard5 card detect pin.
	CAN1_TXD	I	MPF3	CAN bus transmitter1 input.
	EBI_A22	O	MPF7	EBI address bus bit22.
38	PD.8	I/O	MFP0	General purpose digital I/O pin.
	SPI3_MISO1	I/O	MPF1	2nd SPI3 MISO (Master In, Slave Out) pin.
	I2C0_SCL	I/O	MPF2	I2C0 clock pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
39	PD.9	I/O	MFP0	General purpose digital I/O pin.
	SPI3_MOSI1	I/O	MPF1	2nd SPI3 MOSI (Master Out, Slave In) pin.
	I2C0_SDA	I/O	MPF2	I2C0 data input/output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
40	PA.2	I/O	MFP0	General purpose digital I/O pin.
	SC2_DAT	I/O	MPF1	SmartCard2 data pin.
	SPI3_MISO0	I/O	MPF2	1st SPI3 MISO (Master In, Slave Out) pin.
	I2S0_MCLK	O	MPF3	I2S0 master clock output pin.
	BRAKE11	I	MPF4	Brake input pin 1 of EPWM0_.
	CAP_SFIELD	I	MPF5	Video input interface SFIELD input pin.
	EBI_A12	O	MPF7	EBI address bus bit12.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
41	PA.3	I/O	MFP0	General purpose digital I/O pin.
	SC2_CLK	O	MPF1	SmartCard2 clock pin.
	SPI3_MOSI0	I/O	MPF2	1st SPI3 MOSI (Master Out, Slave In) pin.
	I2S0_DO	O	MPF3	I2S0 data output.
	BRAKE10	I	MPF4	Brake input pin 0 of EPWM0_.

	EBI_A13	O	MPF7	EBI address bus bit13.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
42	PA.4	I/O	MFP0	General purpose digital I/O pin.
	SC2_PWR	O	MPF1	SmartCard2 power pin.
	SPI3_CLK	O	MPF2	SPI3 serial clock pin.
	I2S0_DI	I	MPF3	I2S0 data input.
	QEI1_Z	I	MPF5	Quadrature encoder phase Z input of QEI Unit 1.
	EBI_A14	O	MPF7	EBI address bus bit14.
	ECAP1_IC2	I	MPF8	Input 2 of enhanced capture unit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
43	PA.5	I/O	MFP0	General purpose digital I/O pin.
	SC2_RST	O	MPF1	SmartCard2 reset pin.
	SPI3_SS0	I/O	MPF2	General purpose digital I/O pin.
	I2S0_BCLK	O	MPF3	I2S0 bit clock pin.
	PWM0_0	I/O	MPF4	PWM0_0 output/capture input.
	QEI1_B	I	MPF5	Quadrature encoder phase B input of QEI Unit 1.
	EBI_A15	O	MPF7	EBI address bus bit15.
	ECAP1_IC1	I	MPF8	Input 1 of enhanced capture unit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
44	PA.6	I/O	MFP0	General purpose digital I/O pin.
	SC2_CD	I	MPF1	SmartCard2 card detect pin.
	I2S0_LRCK	O	MPF3	I2S0 left right channel clock.
	PWM0_1	I/O	MPF4	PWM0_1 output/capture input.
	QEI1_A	I	MPF5	Quadrature encoder phase A input of QEI Unit 1.
	CAN1_TXD	I	MPF6	CAN bus transmitter1 input.
	EBI_A16	O	MPF7	EBI address bus bit16.

	ECAP1_IC0	I	MPF8	Input 0 of enhanced capture unit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
45	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
46	VSS	P	MFP0	Ground pin for digital circuit.
47	PA.7	I/O	MFP0	General purpose digital I/O pin.
	SC0_CLK	O	MPF2	SmartCard0 clock pin.
	SPI3_SS0	I/O	MPF3	General purpose digital I/O pin.
	PWM1_3	I/O	MPF4	PWM1_3 output/capture input.
	EPWM0_5	I/O	MPF5	PWM0_5 output/capture input.
	EBI_A17	O	MPF7	EBI address bus bit17.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
48	PA.8	I/O	MFP0	General purpose digital I/O pin.
	SC0_RST	O	MPF2	SmartCard0 reset pin.
	SPI3_CLK	O	MPF3	SPI3 serial clock pin.
	PWM1_2	I/O	MPF4	PWM1_2 output/capture input.
	EPWM0_4	I/O	MPF5	PWM0_4 output/capture input.
	EBI_A18	O	MPF7	EBI address bus bit18.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
49	PA.9	I/O	MFP0	General purpose digital I/O pin.
	SC0_PWR	O	MPF2	SmartCard0 power pin.
	SPI3_MISO0	I/O	MPF3	1st SPI3 MISO (Master In, Slave Out) pin.
	PWM1_1	I/O	MPF4	PWM1_1 output/capture input.
	EPWM0_3	I/O	MPF5	PWM0_3 output/capture input.
	EBI_A19	O	MPF7	EBI address bus bit19.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

50	PA.10	I/O	MFP0	General purpose digital I/O pin.
	SC0_DAT	I/O	MPF2	SmartCard0 data pin.
	SPI3_MOSI0	I/O	MPF3	1st SPI3 MOSI (Master Out, Slave In) pin.
	PWM1_0	I/O	MPF4	PWM1_0 output/capture input.
	EPWM0_2	I/O	MPF5	PWM0_2 output/capture input.
	EBI_A20	O	MPF7	EBI address bus bit20.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
51	PA.11	I/O	MFP0	General purpose digital I/O pin.
	UART0_RTS	O	MPF1	Request to Send output pin for UART0.
	SPI3_MISO1	I/O	MPF3	2nd SPI3 MISO (Master In, Slave Out) pin.
	PWM0_5	I/O	MPF4	PWM0_5 output/capture input.
	EPWM0_1	I/O	MPF5	PWM0_1 output/capture input.
	EBI_AD0	O	MPF7	EBI address/data bus bit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
52	PA.12	I/O	MFP0	General purpose digital I/O pin.
	UART0_CTS	I	MPF1	Clear to Send input pin for UART0.
	SPI3_MOSI1	I/O	MPF3	2nd SPI3 MOSI (Master Out, Slave In) pin.
	PWM0_4	I/O	MPF4	PWM0_4 output/capture input.
	EPWM0_0	I/O	MPF5	PWM0_0 output/capture input.
	EBI_AD1	O	MPF7	EBI address/data bus bit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
53	PA.13	I/O	MFP0	General purpose digital I/O pin.
	UART0_RXD	I	MPF1	Data receiver input pin for UART0.
	SC3_DAT	I/O	MPF3	SmartCard3 data pin.
	PWM1_4	I/O	MPF4	PWM1_4 output/capture input.
	EBI_AD2	O	MPF7	EBI address/data bus bit 2.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
54	PA.14	I/O	MFP0	General purpose digital I/O pin.
	UART0_TXD	O	MPF1	Data transmitter output pin for UART0.
	SC3_CLK	O	MPF3	SmartCard3 clock pin.
	PWM1_5	I/O	MPF4	PWM1_5 output/capture input.
	EBI_AD3	O	MPF7	EBI address/data bus bit 3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
55	PD.10	I/O	MFP0	General purpose digital I/O pin.
	SC3_DAT	I/O	MPF1	SmartCard3 data pin.
	I2C4_SCL	I/O	MPF2	I2C4 clock pin.
56	PD.11	I/O	MFP0	General purpose digital I/O pin.
	SC3_RST	O	MPF1	SmartCard3 reset pin.
	TM3_CNT_OUT	I/O	MPF3	Timer3 event counter input//toggle output.
57	PD.12	I/O	MFP0	General purpose digital I/O pin.
	SC3_CLK	O	MPF1	SmartCard3 clock pin.
	I2C4_SDA	I/O	MPF2	I2C4 data input/output pin.
58	PA.15	I/O	MFP0	General purpose digital I/O pin.
	SC3_PWR	O	MPF1	SmartCard3 power pin.
	UART2_RTS	O	MPF2	Request to Send output pin for UART2.
	I2C0_SCL	I/O	MPF4	I2C0 clock pin.
	EBI_A21	O	MPF7	EBI address bus bit21.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
59	PC.9	I/O	MFP0	General purpose digital I/O pin.
	STADC	A	MPF1	ADC analog input.
	UART2_CTS	I	MPF2	Clear to Send input pin for UART2.
	SC3_RST	O	MPF3	SmartCard3 reset pin.

	I2C0_SDA	I/O	MPF4	I2C0 data input/output pin.
	CAP_DATA1	I	MPF5	Image data input bus bit 7.
	I2C3_SCL	I/O	MPF6	I2C3 clock pin.
	EBI_A22	O	MPF7	EBI address bus bit22.
	SD1_DAT0	I/O	MPF8	SD mode #1 data line bit 0.
	EBI_A6	O	MPF9	EBI address bus bit6.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
60	PC.10	I/O	MFP0	General purpose digital I/O pin.
	SC3_CD	I	MPF1	SmartCard3 card detect pin.
	UART2_RXD	I	MPF2	Data receiver input pin for UART2.
	PWM0_2	I/O	MPF4	PWM0_2 output/capture input.
	EBI_A23	O	MPF6	EBI address bus bit23.
	EBI_AD2	O	MPF7	EBI address/data bus bit 2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
61	PC.11	I/O	MFP0	General purpose digital I/O pin.
	UART2_TXD	O	MPF2	Data transmitter output pin for UART2.
	PWM0_3	I/O	MPF4	PWM0_3 output/capture input.
	EBI_A24	O	MPF6	EBI address bus bit24.
	EBI_AD3	O	MPF7	EBI address/data bus bit 3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
62	LDO_CAP	P	MFP0	LDO output pin.
63	VSS	P	MFP0	Ground pin for digital circuit.
64	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
65	PD.13	I/O	MFP0	General purpose digital I/O pin.
	SPI1_SS0	I/O	MPF1	1st SPI1 slave select pin..

	UART5_CTS	I	MPF2	Clear to Send input pin for UART5.
	ECAP0_IC2	O	MPF3	Input 2 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
66	PD.14	I/O	MFP0	General purpose digital I/O pin.
	SPI1_CLK	O	MPF1	SPI1 serial clock pin.
	UART5_RTS	O	MPF2	Request to Send output pin for UART5.
	ECAP0_IC1	O	MPF3	Input 1 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
67	PD.15	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MISO0	I/O	MPF1	1st SPI1 MISO (Master In, Slave Out) pin.
	UART5_TXD	O	MPF2	Data transmitter output pin for UART5.
	ECAP0_IC0	O	MPF3	Input 0 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
68	PF.0	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MOSI0	I/O	MPF1	1st SPI1 MOSI (Master Out, Slave In) pin.
	UART5_RXD	I	MPF2	Data receiver input pin for UART5.
	INT5	I	MPF8	External interrupt5 input pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
69	VRES	A	MFP0	USB PHY VRES ground input pin. Add an 8.2K ohm resistor to VSSA.
70	VBUS	A	MFP0	USB PHY VBUS power input pin.
71	USB_VDD33_CAP	P	MFP0	Internal power regulator output 3.3V decoupling pin.
72	VSSA	P	MFP0	Ground pin for digital circuit. Add a Feritte Bead to digital ground VSS.
73	USB0_D-	A	MFP0	USB0 differential signal D+.
74	USB0_D+	A	MFP0	USB0 differential signal D+.

75	USB0_OTG_ID	I	MFP0	USB0OTG ID pin.
76	PB.0	I/O	MFP0	General purpose digital I/O pin.
	USB0_VBUS_ST	I	MPF1	USB0 external VBUS regulator status
	I2C4_SCL	I/O	MPF2	I2C4 clock pin.
	INT1	I	MPF8	External interrupt1 input pin.
77	PB.1	I/O	MFP0	General purpose digital I/O pin.
	USB0_VBUS_EN	O	MPF1	USB0 external VBUS regulator enable
	I2C4_SDA	I/O	MPF2	I2C4 data input/output pin.
	TM1_CNT_OUT	I/O	MPF3	Timer1 event counter input/toggle output.
78	PF.1	I/O	MFP0	General purpose digital I/O pin.
	SPI2_MOSI1	I/O	MPF1	2nd SPI2 MOSI (Master Out, Slave In) pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
79	PB.2	I/O	MFP0	General purpose digital I/O pin.
	UART1_RXD	I	MPF1	Data receiver input pin for UART1.
	SPI2_SS0	I/O	MPF2	General purpose digital I/O pin.
	USB1_D-	A	MPF3	USB1 differential signal D+.
	EBI_AD4	O	MPF7	EBI address/data bus bit 4.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
80	PB.3	I/O	MFP0	General purpose digital I/O pin.
	UART1_TXD	O	MPF1	Data transmitter output pin for UART1.
	SPI2_CLK	O	MPF2	SPI2 serial clock pin.
	USB1_D+	A	MPF3	USB1 differential signal D+.
	EBI_AD5	O	MPF7	EBI address/data bus bit 5.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
81	PB.4	I/O	MFP0	General purpose digital I/O pin.
	UART1_RTS	O	MPF1	Request to Send output pin for UART1.

	SPI2_MISO0	I/O	MPF2	1st SPI2 MISO (Master In, Slave Out) pin.
	UART4_RXD	I	MPF3	Data receiver input pin for UART4.
	TM0_CNT_OUT	I/O	MPF4	Timer0 event counter input/toggle output.
	EBI_AD6	O	MPF7	EBI address/data bus bit 6.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
82	PB.5	I/O	MFP0	General purpose digital I/O pin.
	UART1_CTS	I	MPF1	Clear to Send input pin for UART1.
	SPI2_MOSI0	I/O	MPF2	1st SPI2 MOSI (Master Out, Slave In) pin.
	UART4_TXD	O	MPF3	Data transmitter output pin for UART4.
	EBI_AD7	O	MPF7	EBI address/data bus bit 7.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
83	PB.6	I/O	MFP0	General purpose digital I/O pin.
	I2C2_SCL	I/O	MPF1	I2C2 clock pin.
	BRAKE01	I	MPF2	Brake input pin 1 of EPWMB.
	UART4_RTS	O	MPF3	Request to Send output pin for UART4.
	PWM1_4	I/O	MPF4	PWM1_4 output/capture input.
	EPWM1_0	I/O	MPF5	PWM1_0 output/capture input.
	EBI_AD8	O	MPF7	EBI address/data bus bit 8.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
84	PB.7	I/O	MFP0	General purpose digital I/O pin.
	I2C2_SDA	I/O	MPF1	I2C2 data input/output pin.
	BRAKE00	I	MPF2	Brake input pin 0 of EPWMB.
	UART4_CTS	I	MPF3	Clear to Send input pin for UART4.
	PWM1_5	I/O	MPF4	PWM1_5 output/capture input.
	EPWM1_1	I/O	MPF5	PWM1_1 output/capture input.
	EBI_AD9	O	MPF7	EBI address/data bus bit 9.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
85	PB.8	I/O	MFP0	General purpose digital I/O pin.
	UART5_CTS	I	MPF1	Clear to Send input pin for UART5.
	EPWM1_2	I/O	MPF5	PWM1_2 output/capture input.
	EBI_AD10	O	MPF7	EBI address/data bus bit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
86	PB.9	I/O	MFP0	General purpose digital I/O pin.
	UART5_RTS	O	MPF1	Request to Send output pin for UART5.
	EPWM1_3	I/O	MPF5	PWM1_3 output/capture input.
	EBI_AD11	O	MPF7	EBI address/data bus bit 11.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
87	PB.10	I/O	MFP0	General purpose digital I/O pin.
	UART5_TXD	O	MPF1	Data transmitter output pin for UART5.
	EPWM1_4	I/O	MPF5	PWM1_4 output/capture input.
	EBI_AD12	O	MPF7	EBI address/data bus bit 12.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
88	PB.11	I/O	MFP0	General purpose digital I/O pin.
	UART5_RXD	I	MPF1	Data receiver input pin for UART5.
	EPWM1_5	I/O	MPF5	PWM1_5 output/capture input.
	EBI_AD13	O	MPF7	EBI address/data bus bit 13.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
89	PB.12	I/O	MFP0	General purpose digital I/O pin.
	UART4_RTS	O	MPF1	Request to Send output pin for UART4.
	SPI2_MISO1	I/O	MPF2	2nd SPI2 MISO (Master In, Slave Out) pin.
	CAN0_RXD	I	MPF3	CAN bus receiver0 input.

	EBI_AD14	O	MPF7	EBI address/data bus bit 14.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
90	PB.13	I/O	MFP0	General purpose digital I/O pin.
	UART4_CTS	I	MPF1	Clear to Send input pin for UART4.
	SPI2_MOSI1	I/O	MPF2	2nd SPI2 MOSI (Master Out, Slave In) pin.
	CAN0_TXD	I	MPF3	CAN bus transmitter0 input.
	EBI_AD15	O	MPF7	EBI address/data bus bit 15.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
91	PB.14	I/O	MFP0	General purpose digital I/O pin.
	I2S1_MCLK	O	MPF1	I2S1 master clock output pin.
	SC1_RST	O	MPF2	SmartCard1 reset pin.
	BRAKE01	I	MPF4	Brake input pin 1 of EPWMB.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
92	PB.15	I/O	MFP0	General purpose digital I/O pin.
	I2S1_DO	O	MPF1	I2S1 data output.
	SC1_DAT	I/O	MPF2	SmartCard1 data pin.
	BRAKE00	I	MPF4	Brake input pin 0 of EPWMB.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
93	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
94	VSS	P	MFP0	Ground pin for digital circuit.
95	PC.0	I/O	MFP0	General purpose digital I/O pin.
	I2S1_DI	I	MPF1	I2S1 data input.
	SC1_DAT	I/O	MPF2	SmartCard1 data pin.
	UART4_RXD	I	MPF3	Data receiver input pin for UART4.
	EBI_MCLK	O	MPF7	EBI interface clock output pin.

	INT2	I	MPF8	External interrupt2 input pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
96	PC.1	I/O	MFP0	General purpose digital I/O pin.
	I2S1_BCLK	O	MPF1	I2S1 bit clock pin.
	SC1_CLK	O	MPF2	SmartCard1 clock pin.
	UART4_TXD	O	MPF3	Data transmitter output pin for UART4.
	TM3_CNT_OUT	I/O	MPF5	Timer3 event counter input/toggle output.
	EBI_AD13	O	MPF7	EBI address/data bus bit 13.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
97	PC.2	I/O	MFP0	General purpose digital I/O pin.
	I2S1_LRCK	O	MPF1	I2S1 left right channel clock.
	SC1_PWR	O	MPF2	SmartCard1 power pin.
	UART4_RTS	O	MPF3	Request to Send output pin for UART4.
	SPI0_SS0	I/O	MPF4	General purpose digital I/O pin.
	EBI_AD12	O	MPF7	EBI address/data bus bit 12.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
98	PC.3	I/O	MFP0	General purpose digital I/O pin.
	I2S1_MCLK	O	MPF1	I2S1 master clock output pin.
	SC1_CD	I	MPF2	SmartCard1 card detect pin.
	UART4_CTS	I	MPF3	Clear to Send input pin for UART4.
	SPI0_MISO1	I/O	MPF4	2nd SPI0 MISO (Master In, Slave Out) pin.
	QEI0_Z	I	MPF5	Quadrature encoder phase Z input of QEI Unit 0.
	EBI_AD11	O	MPF7	EBI address/data bus bit 11.
	ECAP0_IC2	O	MPF8	Input 2 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

99	PC.4	I/O	MFP0	General purpose digital I/O pin.
	I2S1_DO	O	MPF1	I2S1 data output.
	SC1_RST	O	MPF2	SmartCard1 reset pin.
	SPI0_MOSI1	I/O	MPF4	2nd SPI0 MOSI (Master Out, Slave In) pin.
	QEI0_B	I	MPF5	Quadrature encoder phase B input of QEI Unit 0.
	EBI_AD10	O	MPF7	EBI address/data bus bit 10.
	ECAP0_IC1	O	MPF8	Input 1 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
100	PC.5	I/O	MFP0	General purpose digital I/O pin.
	CLKO	O	MFP1	Clock Output Pin.
	QEI0_A	I	MPF5	Quadrature encoder phase A input of QEI Unit 0.
	EBI_MCLK	O	MPF7	EBI interface clock output pin.
	ECAP0_IC0	O	MPF8	Input 0 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
101	PC.6	I/O	MFP0	General purpose digital I/O pin.
	TM2_EXT	I	MPF1	Timer2 external counter input
	SPI0_MISO0	I/O	MPF4	1st SPI0 MISO (Master In, Slave Out) pin.
	TM2_CNT_OUT	I/O	MPF5	Timer2 event counter input/toggle output.
	EBI_AD9	O	MPF7	EBI address/data bus bit 9.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
102	PC.7	I/O	MFP0	General purpose digital I/O pin.
	TM1_EXT	I	MPF1	Timer1 external counter input
	SPI0_MOSI0	I/O	MPF4	1st SPI0 MOSI (Master Out, Slave In) pin.
	EBI_AD8	O	MPF7	EBI address/data bus bit 8.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

103	PC.8	I/O	MFP0	General purpose digital I/O pin.
	TM0_EXT	I	MPF1	Timer0 external counter input
	SPI0_CLK	O	MPF4	SPI0 serial clock pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
104	PF.2	I/O	MFP0	General purpose digital I/O pin.
	SPI3_SS0	I/O	MPF1	General purpose digital I/O pin.
	SD0_DAT3	I/O	MPF4	SD mode #0 data line bit 3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
105	PF.3	I/O	MFP0	General purpose digital I/O pin.
	SPI3_CLK	O	MPF1	SPI3 serial clock pin.
	SD0_DAT2	I/O	MPF4	SD mode #0 data line bit 2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
106	PF.4	I/O	MFP0	General purpose digital I/O pin.
	SPI3_MISO0	I/O	MPF1	1st SPI3 MISO (Master In, Slave Out) pin.
	SD0_DAT1	I/O	MPF4	SD mode #0 data line bit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
107	PF.5	I/O	MFP0	General purpose digital I/O pin.
	SPI3_MOSI0	I/O	MPF1	1st SPI3 MOSI (Master Out, Slave In) pin.
	SD0_DAT0	I/O	MPF4	SD mode #0 data line bit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
108	PF.6	I/O	MFP0	General purpose digital I/O pin.
	UART2_RXD	I	MPF1	Data receiver input pin for UART2.
	SD0_CDn	I	MPF4	SD mode #0 – card detect
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
109	PF.7	I/O	MFP0	General purpose digital I/O pin.

	UART2_TXD	O	MPF1	Data transmitter output pin for UART2.
	SD0_CMD	I/O	MPF4	SD mode #0 – command/response
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
110	PF.8	I/O	MFP0	General purpose digital I/O pin.
	UART2_RTS	O	MPF1	Request to Send output pin for UART2.
	SD0_CLK	O	MPF4	SD mode #0 – clock.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
111	LDO_CAP	P	MFP0	LDO output pin.
112	VSS	P	MFP0	Ground pin for digital circuit.
113	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
114	PE.0	I/O	MFP0	General purpose digital I/O pin.
	ADC0_0	A	MPF1	ADC0 analog input.
	INT4	I	MPF8	External interrupt4 input pin.
115	PE.1	I/O	MFP0	General purpose digital I/O pin.
	ADC0_1	A	MPF1	ADC0 analog input.
	TM2_CNT_OUT	I/O	MPF3	Timer2 event counter input/toggle output.
116	PE.2	I/O	MFP0	General purpose digital I/O pin.
	ADC0_2	A	MPF1	ADC0 analog input.
	ACMP0_O	O	MPF2	Analog ccomparator0 output .
	SPI0_MISO0	I/O	MPF3	1st SPI0 MISO (Master In, Slave Out) pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
117	PE.3	I/O	MFP0	General purpose digital I/O pin.
	ADC0_3	A	MPF1	ADC0 analog input.
	ACMP0_P3	A	MPF2	Analog comparator0 positive input pin.
	SPI0_MOSI0	I/O	MPF3	1st SPI0 MOSI (Master Out, Slave In) pin.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
118	PE.4	I/O	MFP0	General purpose digital I/O pin.
	ADC0_4	A	MPF1	ADC0 analog input.
	ACMP0_P2	A	MPF2	Analog comparator0 positive input pin.
	SPI0_SS0	I/O	MPF3	General purpose digital I/O pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
119	PE.5	I/O	MFP0	General purpose digital I/O pin.
	ADC0_5	A	MPF1	ADC0 analog input.
	ACMP0_P1	A	MPF2	Analog comparator0 positive input pin.
	SPI0_CLK	O	MPF3	SPI0 serial clock pin.
	SD0_CDn	I	MPF4	SD mode #0 – card detect
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
120	PE.6	I/O	MFP0	General purpose digital I/O pin.
	ADC0_6	A	MPF1	ADC0 analog input.
	ACMP0_P0	A	MPF2	Analog comparator0 positive input pin.
	SPI0_MISO0	I/O	MPF3	1st SPI0 MISO (Master In, Slave Out) pin.
	SD0_CMD	I/O	MPF4	SD mode #0 – command/response
	EBI_nWR	O	MPF7	EBI write enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
121	PE.7	I/O	MFP0	General purpose digital I/O pin.
	ADC0_7	A	MPF1	ADC0 analog input.
	ACMP0_N	A	MPF2	Analog comparator0 negative input pin.
	SPI0_MOSI0	I/O	MPF3	1st SPI0 MOSI (Master Out, Slave In) pin.
	SD0_CLK	O	MPF4	SD mode #0– clock.
	EBI_nRD	O	MPF7	EBI read enable output pin.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
122	AVSS	P	MFP0	Ground pin for digital circuit.
123	Vref	A	MFP0	Voltage reference input for ADC.
124	AVDD	P	MFP0	Power supply for internal analog circuit.
125	PE.8	I/O	MFP0	General purpose digital I/O pin.
	ADC1_0	A	MPF1	ADC1 analog input.
	ADC0_8	A	MPF1	ADC0 analog input.
	ACMP1_N	A	MPF2	Analog comparator1 negative input pin.
	TM1_CNT_OUT	I/O	MPF3	Timer1 event counter input/toggle output.
	SD0_DAT3	I/O	MPF4	SD mode #0 data line bit 3.
	EBI_ALE	O	MPF7	EBI address latch enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
126	PE.9	I/O	MFP0	General purpose digital I/O pin.
	ADC1_1	A	MPF1	ADC1 analog input.
	ADC0_9	A	MPF1	ADC0 analog input.
	ACMP1_P0	A	MPF2	Analog comparator1 positive input pin.
	SD0_DAT2	I/O	MPF4	SD mode #0 data line bit 2.
	EBI_nWRH	O	MPF7	EBI write enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
127	PE.10	I/O	MFP0	General purpose digital I/O pin.
	ADC1_2	A	MPF1	ADC1 analog input.
	ADC0_10	A	MPF1	ADC0 analog input.
	ACMP1_P1	A	MPF2	Analog comparator1 positive input pin.
	SPI0_MISO1	I/O	MPF3	2nd SPI0 MISO (Master In, Slave Out) pin.
	SD0_DAT1	I/O	MPF4	SD mode #0 data line bit 1.
	EBI_nWRL	O	MPF7	EBI write enable output pin.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
128	PE.11	I/O	MFP0	General purpose digital I/O pin.
	ADC1_3	A	MPF1	ADC1 analog input.
	ADC0_11	A	MPF1	ADC0 analog input.
	ACMP1_P2	A	MPF2	Analog comparator1 positive input pin.
	SPI0_MOSI1	I/O	MPF3	2nd SPI0 MOSI (Master Out, Slave In) pin.
	SD0_DAT0	I/O	MPF4	SD mode #0 data line bit 0.
	ACMP2_P3	A	MPF5	Analog comparator2 positive input pin.
	EBI_nCS0	O	MPF7	EBI chip select 0 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

Note: Pin Type I = Digital Input, O = Digital Output; A = Analog Pin; P = Power Pin;

4.3.4 NuMicro™ NUC442 Package LQFP 144-pin Description

MFP = Multi-function pin.

Pin No.	Pin Name	Type	MFP*	Description
1	PE.12	I/O	MFP0	General purpose digital I/O pin.
	ADC1_4	A	MPF1	ADC1 analog input.
	ACMP1_P3	A	MPF2	Analog comparator1 positive input pin.
	ACMP2_P2	A	MPF3	Analog comparator2 positive input pin.
	EBI_nCS1	O	MPF7	EBI chip select 1 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
2	PE.13	I/O	MFP0	General purpose digital I/O pin.
	ADC1_5	A	MPF1	ADC1 analog input.
	ACMP2_P1	A	MPF3	Analog comparator2 positive input pin.
	EBI_nCS2	O	MPF7	EBI chip select 2 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
3	PE.14	I/O	MFP0	General purpose digital I/O pin.
	ADC1_6	A	MPF1	ADC1 analog input.
	ACMP2_P0	A	MPF3	Analog comparator2 positive input pin.
	EBI_nCS3	O	MPF7	EBI chip select 3 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
4	PE.15	I/O	MFP0	General purpose digital I/O pin.
	ADC1_7	A	MPF1	ADC1 analog input.
	ACMP2_N	A	MPF3	Analog comparator2 negative input pin.
5	PF.9	I/O	MFP0	General purpose digital I/O pin.
	OPA0_IN+	I/O	MPF1	General purpose digital I/O pin.
	PWM0_0	I/O	MPF4	PWM0_0 output/capture input.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

6	PF.10	I/O	MFP0	General purpose digital I/O pin.
	OPA0_IN-	I/O	MPF1	General purpose digital I/O pin.
	PWM0_1	I/O	MPF4	PWM0_1 output/capture input.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
7	PF.11	I/O	MFP0	General purpose digital I/O pin.
	OPA0_O	O	MPF1	Operational amplifier output pin
	UART1_RTS	O	MPF2	Request to Send output pin for UART1.
8	PF.12	I/O	MFP0	General purpose digital I/O pin.
	OPA1_IN+	I/O	MPF1	General purpose digital I/O pin.
	UART1_CTS	I	MPF2	Clear to Send input pin for UART1.
9	PF.13	I/O	MFP0	General purpose digital I/O pin.
	OPA1_IN-	I/O	MPF1	General purpose digital I/O pin.
	UART1_TXD	O	MPF2	Data transmitter output pin for UART1.
10	PF.14	I/O	MFP0	General purpose digital I/O pin.
	OPA1_O	O	MPF1	Operational amplifier output pin
	UART1_RXD	I	MPF2	Data receiver input pin for UART1.
11	VSS	P	MFP0	Ground pin for digital circuit.
12	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
13	PF.15	I/O	MFP0	General purpose digital I/O pin.
	UART0_RTS	O	MPF1	Request to Send output pin for UART0.
14	PG.0	I/O	MFP0	General purpose digital I/O pin.
	UART0_CTS	I	MPF1	Clear to Send input pin for UART0.
	INT6	I	MPF8	External interrupt6 input pin.
15	PG.1	I/O	MFP0	General purpose digital I/O pin.
	UART0_RXD	I	MPF1	Data receiver input pin for UART0.
16	PG.2	I/O	MFP0	General purpose digital I/O pin.

	UART0_TXD	O	MPF1	Data transmitter output pin for UART0.
17	PC.12	I/O	MFP0	General purpose digital I/O pin.
	SPI1_SS0	I/O	MPF1	1st SPI1 slave select pin..
	SC4_CD	I	MPF2	SmartCard4 card detect pin.
	SD1_CDn	I	MPF4	SD mode #1 – card detect
	CAP_DATA7	I	MPF5	Image data input bus bit 7.
	EBI_A0	O	MPF7	EBI address bus bit0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
18	PC.13	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MOSI1	I/O	MPF1	2nd SPI1 MOSI (Master Out, Slave In) pin.
	SC4_RST	O	MPF2	SmartCard4 reset pin.
	SD1_CMD	I/O	MPF4	SD mode #1 – command/response
	CAP_DATA6	I	MPF5	Image data input bus bit 7.
	EBI_A1	O	MPF7	EBI address bus bit1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
19	PC.14	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MISO1	I/O	MPF1	2nd SPI1 MISO (Master In, Slave Out) pin.
	SC4_PWR	O	MPF2	SmartCard4 power pin.
	TM3_EXT	I	MPF3	Timer3 external counter input
	SD1_CLK	O	MPF4	SD mode #1– clock.
	CAP_DATA5	I	MPF5	Image data input bus bit 7.
	EBI_A2	O	MPF7	EBI address bus bit2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
20	PC.15	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MOSI0	I/O	MPF1	1st SPI1 MOSI (Master Out, Slave In) pin.
	SC4_DAT	I/O	MPF2	SmartCard4 data pin.

	SD1_DAT3	I/O	MPF4	SD mode #1 data line bit 3;
	CAP_DATA4	I	MPF5	Image data input bus bit 7.
	EBI_A3	O	MPF7	EBI address bus bit3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
21	PD.0	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MISO0	I/O	MPF1	1st SPI1 MISO (Master In, Slave Out) pin.
	SC4_CLK	O	MPF2	SmartCard4 clock pin.
	SD1_DAT2	I/O	MPF4	SD mode #1 data line bit 2;
	CAP_DATA3	I	MPF5	Image data input bus bit 7.
	EBI_A4	O	MPF7	EBI address bus bit4.
	INT3	I	MPF8	External interrupt3 input pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
22	PD.1	I/O	MFP0	General purpose digital I/O pin.
	SPI1_CLK	O	MPF1	SPI1 serial clock pin.
	TM0_CNT_OUT	I/O	MPF3	Timer0 event counter input/toggle output.
	SD1_DAT1	I/O	MPF4	SD mode #1 data line bit 1;
	CAP_DATA2	I	MPF5	Image data input bus bit 7.
	EBI_A5	O	MPF7	EBI address bus bit5.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
23	PD.2	I/O	MFP0	General purpose digital I/O pin.
	STADC	A	MPF1	ADC analog input.
	I2C3_SCL	I/O	MPF2	I2C3 clock pin.
	SD1_DAT0	I/O	MPF4	SD mode #1 data line bit 0.
	CAP_DATA1	I	MPF5	Image data input bus bit 7.
	EBI_A6	O	MPF7	EBI address bus bit6.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
24	PD.3	I/O	MFP0	General purpose digital I/O pin.
	SC5_CLK	O	MPF1	SmartCard5 clock pin.
	I2C3_SDA	I/O	MPF2	I2C3 data input/output pin.
	ACMP2_O	O	MPF3	Analog ccomparator2 output .
	SD0_CDn	I	MPF4	SD mode #0 – card detect
	CAP_DATA0	I	MPF5	Image data input bus bit 7.
	EBI_A7	O	MPF7	EBI address bus bit7.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
25	PD.4	I/O	MFP0	General purpose digital I/O pin.
	SC5_CD	I	MPF1	SmartCard5 card detect pin.
	UART3_RXD	I	MPF2	Data receiver input pin for UART3.
	ACMP1_O	O	MPF3	Analog ccomparator1 output.
	CAP_SCLK	O	MPF5	Image capture interface sensor clock pin.
	EBI_A8	O	MPF7	EBI address bus bit8.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
26	PD.5	I/O	MFP0	General purpose digital I/O pin.
	SC5_RST	O	MPF1	SmartCard5 reset pin.
	UART3_TXD	O	MPF2	Data transmitter output pin for UART3.
	CAP_VSYNC	I	MPF5	Image capture interface VSYNC input pin.
	EBI_A9	O	MPF7	EBI address bus bit9.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
27	PD.6	I/O	MFP0	General purpose digital I/O pin.
	SC5_PWR	O	MPF1	SmartCard5 power pin.
	UART3_RTS	O	MPF2	Request to Send output pin for UART3.

	SD0_CMD	I/O	MPF4	SD mode #0 – command/response
	CAP_HSYNC	I	MPF5	Image capture interface HSYNC input pin.
	EBI_A10	O	MPF7	EBI address bus bit10.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
28	PD.7	I/O	MFP0	General purpose digital I/O pin.
	SC5_DAT	I/O	MPF1	SmartCard5 data pin.
	UART3_CTS	I	MPF2	Clear to Send input pin for UART3.
	SD0_CLK	O	MPF4	SD mode #0 – clock.
	CAP_PIXCLK	I	MPF5	Image capture interface pix clock input pin.
	EBI_A11	O	MPF7	EBI address bus bit11.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
29	PG.13	I/O	MFP0	General purpose digital I/O pin.
	XT1_IN	I	MPF1	External 4~24 MHz (high-speed) crystal input pin.
30	PG.12	I/O	MFP0	General purpose digital I/O pin.
	XT1_OUT	O	MPF1	External 4~24 MHz (high-speed) crystal output pin.
31	nRESET	I	MFP0	External reset input: active LOW, with an internal pull-up. Set this pin low reset to initial state.
32	LDO_CAP	P	MFP0	LDO output pin.
33	VSS	P	MFP0	Ground pin for digital circuit.
34	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
35	PG.10	I/O	MFP0	General purpose digital I/O pin.
	ICE_CLK	I	MPF1	Serial wired debugger clock pin
36	PG.11	I/O	MFP0	General purpose digital I/O pin.
	ICE_DAT	I/O	MPF1	Serial wired debugger data pin
37	PG.15	I/O	MFP0	General purpose digital I/O pin.
	X32K_IN	I	MPF1	External 32.768 kHz (low-speed) crystal input pin.

	I2C1_SCL	I/O	MPF3	I2C1 clock pin.
38	PG.14	I/O	MFP0	General purpose digital I/O pin.
	X32K_OUT	O	MPF1	External 32.768 kHz (low-speed) crystal output pin.
	I2C1_SDA	I/O	MPF3	I2C1 data input/output pin.
39	VBAT	P	MFP0	Battery power input pin.
40	PA.0	I/O	MFP0	General purpose digital I/O pin.
	TAMPER0	I/O	MPF1	Tamper detect pin 0.
	SC0_CD	I	MPF2	SmartCard0 card detect pin.
	CAN1_RXD	I	MPF3	CAN bus receiver1 input.
	INT0	I	MPF8	External interrupt0 input pin.
41	PA.1	I/O	MFP0	General purpose digital I/O pin.
	TAMPER1	I/O	MPF1	Tamper detect pin 1.
	SC5_CD	I	MPF2	SmartCard5 card detect pin.
	CAN1_TXD	I	MPF3	CAN bus transmitter1 input.
	EBI_A22	O	MPF7	EBI address bus bit22.
42	PD.8	I/O	MFP0	General purpose digital I/O pin.
	SPI3_MISO1	I/O	MPF1	2nd SPI3 MISO (Master In, Slave Out) pin.
	I2C0_SCL	I/O	MPF2	I2C0 clock pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
43	PD.9	I/O	MFP0	General purpose digital I/O pin.
	SPI3_MOSI1	I/O	MPF1	2nd SPI3 MOSI (Master Out, Slave In) pin.
	I2C0_SDA	I/O	MPF2	I2C0 data input/output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
44	PA.2	I/O	MFP0	General purpose digital I/O pin.
	SC2_DAT	I/O	MPF1	SmartCard2 data pin.
	SPI3_MISO0	I/O	MPF2	1st SPI3 MISO (Master In, Slave Out) pin.

	I2S0_MCLK	O	MPF3	I2S0 master clock output pin.
	BRAKE11	I	MPF4	Brake input pin 1 of EPWMA.
	CAP_SFIELD	I	MPF5	Video input interface SFIELD input pin.
	EBI_A12	O	MPF7	EBI address bus bit12.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
45	PA.3	I/O	MFP0	General purpose digital I/O pin.
	SC2_CLK	O	MPF1	SmartCard2 clock pin.
	SPI3_MOSI0	I/O	MPF2	1st SPI3 MOSI (Master Out, Slave In) pin.
	I2S0_DO	O	MPF3	I2S0 data output.
	BRAKE10	I	MPF4	Brake input pin 0 of EPWMA.
	EBI_A13	O	MPF7	EBI address bus bit13.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
46	PA.4	I/O	MFP0	General purpose digital I/O pin.
	SC2_PWR	O	MPF1	SmartCard2 power pin.
	SPI3_CLK	O	MPF2	SPI3 serial clock pin.
	I2S0_DI	I	MPF3	I2S0 data input.
	QEI1_Z	I	MPF5	Quadrature encoder phase Z input of QEI Unit 1.
	EBI_A14	O	MPF7	EBI address bus bit14.
	ECAP1_IC2	I	MPF8	Input 2 of enhanced capture unit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
47	PA.5	I/O	MFP0	General purpose digital I/O pin.
	SC2_RST	O	MPF1	SmartCard2 reset pin.
	SPI3_SS0	I/O	MPF2	General purpose digital I/O pin.
	I2S0_BCLK	O	MPF3	I2S0 bit clock pin.
	PWM0_0	I/O	MPF4	PWM0_0 output/capture input.
	QEI1_B	I	MPF5	Quadrature encoder phase B input of QEI Unit 1.

	EBI_A15	O	MPF7	EBI address bus bit15.
	ECAP1_IC1	I	MPF8	Input 1 of enhanced capture unit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
48	PA.6	I/O	MFP0	General purpose digital I/O pin.
	SC2_CD	I	MPF1	SmartCard2 card detect pin.
	I2S0_LRCK	O	MPF3	I2S0 left right channel clock.
	PWM0_1	I/O	MPF4	PWM0_1 output/capture input.
	QEI1_A	I	MPF5	Quadrature encoder phase A input of QEI Unit 1.
	CAN1_TXD	I	MPF6	CAN bus transmitter1 input.
	EBI_A16	O	MPF7	EBI address bus bit16.
	ECAP1_IC0	I	MPF8	Input 0 of enhanced capture unit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
49	PG.3	I/O	MFP0	General purpose digital I/O pin.
	PS2_CLK	O	MPF1	PS2 clock pin.
	I2S1_DO	O	MPF2	I2S1 data output.
	SC1_RST	O	MPF3	SmartCard1 reset pin.
50	PG.4	I/O	MFP0	General purpose digital I/O pin.
	PS2_DAT	I/O	MPF1	PS2 data pin.
	I2S1_DI	I	MPF2	I2S1 data input.
	SC1_PWR	O	MPF3	SmartCard1 power pin.
51	PG.5	I/O	MFP0	General purpose digital I/O pin.
	I2S1_BCLK	O	MPF2	I2S1 bit clock pin.
	SC1_DAT	I/O	MPF3	SmartCard1 data pin.
52	PG.6	I/O	MFP0	General purpose digital I/O pin.
	I2S1_LRCK	O	MPF2	I2S1 left right channel clock.
	SC1_CLK	O	MPF3	SmartCard1 clock pin.

53	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
54	VSS	P	MFP0	Ground pin for digital circuit.
55	PA.7	I/O	MFP0	General purpose digital I/O pin.
	SC0_CLK	O	MPF2	SmartCard0 clock pin.
	SPI3_SS0	I/O	MPF3	General purpose digital I/O pin.
	PWM1_3	I/O	MPF4	PWM1_3 output/capture input.
	EPWM0_5	I/O	MPF5	PWM0_5 output/capture input.
	EBI_A17	O	MPF7	EBI address bus bit17.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
56	PA.8	I/O	MFP0	General purpose digital I/O pin.
	SC0_RST	O	MPF2	SmartCard0 reset pin.
	SPI3_CLK	O	MPF3	SPI3 serial clock pin.
	PWM1_2	I/O	MPF4	PWM1_2 output/capture input.
	EPWM0_4	I/O	MPF5	PWM0_4 output/capture input.
	EBI_A18	O	MPF7	EBI address bus bit18.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
57	PA.9	I/O	MFP0	General purpose digital I/O pin.
	SC0_PWR	O	MPF2	SmartCard0 power pin.
	SPI3_MISO0	I/O	MPF3	1st SPI3 MISO (Master In, Slave Out) pin.
	PWM1_1	I/O	MPF4	PWM1_1 output/capture input.
	EPWM0_3	I/O	MPF5	PWM0_3 output/capture input.
	EBI_A19	O	MPF7	EBI address bus bit19.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
58	PA.10	I/O	MFP0	General purpose digital I/O pin.
	SC0_DAT	I/O	MPF2	SmartCard0 data pin.

	SPI3_MOSI0	I/O	MPF3	1st SPI3 MOSI (Master Out, Slave In) pin.
	PWM1_0	I/O	MPF4	PWM1_0 output/capture input.
	EPWM0_2	I/O	MPF5	PWM0_2 output/capture input.
	EBI_A20	O	MPF7	EBI address bus bit20.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
59	PA.11	I/O	MFP0	General purpose digital I/O pin.
	UART0_RTS	O	MPF1	Request to Send output pin for UART0.
	SPI3_MISO1	I/O	MPF3	2nd SPI3 MISO (Master In, Slave Out) pin.
	PWM0_5	I/O	MPF4	PWM0_5 output/capture input.
	EPWM0_1	I/O	MPF5	PWM0_1 output/capture input.
	EBI_AD0	O	MPF7	EBI address/data bus bit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
60	PA.12	I/O	MFP0	General purpose digital I/O pin.
	UART0_CTS	I	MPF1	Clear to Send input pin for UART0.
	SPI3_MOSI1	I/O	MPF3	2nd SPI3 MOSI (Master Out, Slave In) pin.
	PWM0_4	I/O	MPF4	PWM0_4 output/capture input.
	EPWM0_0	I/O	MPF5	PWM0_0 output/capture input.
	EBI_AD1	O	MPF7	EBI address/data bus bit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
61	PA.13	I/O	MFP0	General purpose digital I/O pin.
	UART0_RXD	I	MPF1	Data receiver input pin for UART0.
	SC3_DAT	I/O	MPF3	SmartCard3 data pin.
	PWM1_4	I/O	MPF4	PWM1_4 output/capture input.
	EBI_AD2	O	MPF7	EBI address/data bus bit 2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

62	PA.14	I/O	MFP0	General purpose digital I/O pin.
	UART0_TXD	O	MPF1	Data transmitter output pin for UART0.
	SC3_CLK	O	MPF3	SmartCard3 clock pin.
	PWM1_5	I/O	MPF4	PWM1_5 output/capture input.
	EBI_AD3	O	MPF7	EBI address/data bus bit 3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
63	PD.10	I/O	MFP0	General purpose digital I/O pin.
	SC3_DAT	I/O	MPF1	SmartCard3 data pin.
	I2C4_SCL	I/O	MPF2	I2C4 clock pin.
64	PD.11	I/O	MFP0	General purpose digital I/O pin.
	SC3_RST	O	MPF1	SmartCard3 reset pin.
	TM3_CNT_OUT	I/O	MPF3	Timer3 event counter input/toggle output.
65	PD.12	I/O	MFP0	General purpose digital I/O pin.
	SC3_CLK	O	MPF1	SmartCard3 clock pin.
	I2C4_SDA	I/O	MPF2	I2C4 data input/output pin.
66	PA.15	I/O	MFP0	General purpose digital I/O pin.
	SC3_PWR	O	MPF1	SmartCard3 power pin.
	UART2_RTS	O	MPF2	Request to Send output pin for UART2.
	I2C0_SCL	I/O	MPF4	I2C0 clock pin.
	EBI_A21	O	MPF7	EBI address bus bit21.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
67	PC.9	I/O	MFP0	General purpose digital I/O pin.
	STADC	A	MPF1	ADC analog input.
	UART2_CTS	I	MPF2	Clear to Send input pin for UART2.
	SC3_RST	O	MPF3	SmartCard3 reset pin.
	I2C0_SDA	I/O	MPF4	I2C0 data input/output pin.

	CAP_DATA1	I	MPF5	Image data input bus bit 7.
	I2C3_SCL	I/O	MPF6	I2C3 clock pin.
	EBI_A22	O	MPF7	EBI address bus bit22.
	SD1_DAT0	I/O	MPF8	SD mode #1 data line bit 0.
	EBI_A6	O	MPF9	EBI address bus bit6.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
68	PC.10	I/O	MFP0	General purpose digital I/O pin.
	SC3_CD	I	MPF1	SmartCard3 card detect pin.
	UART2_RXD	I	MPF2	Data receiver input pin for UART2.
	PWM0_2	I/O	MPF4	PWM0_2 output/capture input.
	EBI_A23	O	MPF6	EBI address bus bit23.
	EBI_AD2	O	MPF7	EBI address/data bus bit 2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
69	PC.11	I/O	MFP0	General purpose digital I/O pin.
	UART2_TXD	O	MPF2	Data transmitter output pin for UART2.
	PWM0_3	I/O	MPF4	PWM0_3 output/capture input.
	EBI_A24	O	MPF6	EBI address bus bit24.
	EBI_AD3	O	MPF7	EBI address/data bus bit 3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
70	LDO_CAP	P	MFP0	LDO output pin.
71	VSS	P	MFP0	Ground pin for digital circuit.
72	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
73	PD.13	I/O	MFP0	General purpose digital I/O pin.
	SPI1_SS0	I/O	MPF1	1st SPI1 slave select pin.
	UART5_CTS	I	MPF2	Clear to Send input pin for UART5.

	ECAP0_IC2	O	MPF3	Input 2 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
74	PD.14	I/O	MFP0	General purpose digital I/O pin.
	SPI1_CLK	O	MPF1	SPI1 serial clock pin.
	UART5_RTS	O	MPF2	Request to Send output pin for UART5.
	ECAP0_IC1	O	MPF3	Input 1 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
75	PD.15	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MISO0	I/O	MPF1	1st SPI1 MISO (Master In, Slave Out) pin.
	UART5_TXD	O	MPF2	Data transmitter output pin for UART5.
	ECAP0_IC0	O	MPF3	Input 0 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
76	PF.0	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MOSI0	I/O	MPF1	1st SPI1 MOSI (Master Out, Slave In) pin.
	UART5_RXD	I	MPF2	Data receiver input pin for UART5.
	INT5	I	MPF8	External interrupt5 input pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
77	VRES	A	MFP0	USB PHY VRES ground input pin. Add an 8.2K ohm resistor to VSSA.
78	VBUS	A	MFP0	USB PHY VBUS power input pin.
79	USB_VDD33_CAP	P	MFP0	Internal power regulator output 3.3V decoupling pin.
80	VSSA	P	MFP0	Ground pin for digital circuit. Add a Feritte Bead to digital ground VSS.
81	USB0_D-	A	MFP0	USB0 differential signal D+.
82	USB0_D+	A	MFP0	USB0 differential signal D+.
83	USB0_OTG_ID	I	MFP0	USB0OTG ID pin.

84	PB.0	I/O	MFP0	General purpose digital I/O pin.
	USB0_VBUS_ST	I	MPF1	USB0 external VBUS regulator status
	I2C4_SCL	I/O	MPF2	I2C4 clock pin.
	INT1	I	MPF8	External interrupt1 input pin.
85	PB.1	I/O	MFP0	General purpose digital I/O pin.
	USB0_VBUS_EN	O	MPF1	USB0 external VBUS regulator enabled
	I2C4_SDA	I/O	MPF2	I2C4 data input/output pin.
	TM1_CNT_OUT	I/O	MPF3	Timer1 event counter input/toggle output.
86	PG.7	I/O	MFP0	General purpose digital I/O pin.
	SPI2_MISO0	I/O	MPF1	1st SPI2 MISO (Master In, Slave Out) pin.
	I2S1_MCLK	O	MPF2	I2S1 master clock output pin.
	SC1_CD	I	MPF3	SmartCard1 card detect pin.
	SC3_RST	O	MPF4	SmartCard3 reset pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
87	PG.8	I/O	MFP0	General purpose digital I/O pin.
	SPI2_MOSI0	I/O	MPF1	1st SPI2 MOSI (Master Out, Slave In) pin.
	I2S1_DO	O	MPF2	I2S1 data output.
	UART4_RTS	O	MPF3	Request to Send output pin for UART4.
	SC3_DAT	I/O	MPF4	SmartCard3 data pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
88	PG.9	I/O	MFP0	General purpose digital I/O pin.
	SPI2_CLK	O	MPF1	SPI2 serial clock pin.
	I2S1_DI	I	MPF2	I2S1 data input.
	UART4_CTS	I	MPF3	Clear to Send input pin for UART4.
	SC3_CLK	O	MPF4	SmartCard3 clock pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

89	PB.2	I/O	MFP0	General purpose digital I/O pin.
	UART1_RXD	I	MPF1	Data receiver input pin for UART1.
	SPI2_SS0	I/O	MPF2	General purpose digital I/O pin.
	USB1_D-	A	MPF3	USB1 differential signal D+.
	EBI_AD4	O	MPF7	EBI address/data bus bit 4.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
90	PB.3	I/O	MFP0	General purpose digital I/O pin.
	UART1_TXD	O	MPF1	Data transmitter output pin for UART1.
	SPI2_CLK	O	MPF2	SPI2 serial clock pin.
	USB1_D+	A	MPF3	USB1 differential signal D+.
	EBI_AD5	O	MPF7	EBI address/data bus bit 5.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
91	PB.4	I/O	MFP0	General purpose digital I/O pin.
	UART1_RTS	O	MPF1	Request to Send output pin for UART1.
	SPI2_MISO0	I/O	MPF2	1st SPI2 MISO (Master In, Slave Out) pin.
	UART4_RXD	I	MPF3	Data receiver input pin for UART4.
	TM0_CNT_OUT	I/O	MPF4	Timer0 event counter input/toggle output.
	EBI_AD6	O	MPF7	EBI address/data bus bit 6.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
92	PB.5	I/O	MFP0	General purpose digital I/O pin.
	UART1_CTS	I	MPF1	Clear to Send input pin for UART1.
	SPI2_MOSI0	I/O	MPF2	1st SPI2 MOSI (Master Out, Slave In) pin.
	UART4_TXD	O	MPF3	Data transmitter output pin for UART4.
	EBI_AD7	O	MPF7	EBI address/data bus bit 7.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

93	PB.6	I/O	MFP0	General purpose digital I/O pin.
	I2C2_SCL	I/O	MPF1	I2C2 clock pin.
	BRAKE01	I	MPF2	Brake input pin 1 of EPWMB.
	UART4_RTS	O	MPF3	Request to Send output pin for UART4.
	PWM1_4	I/O	MPF4	PWM1_4 output/capture input.
	EPWM1_0	I/O	MPF5	PWM1_0 output/capture input.
	EBI_AD8	O	MPF7	EBI address/data bus bit 8.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
94	PB.7	I/O	MFP0	General purpose digital I/O pin.
	I2C2_SDA	I/O	MPF1	I2C2 data input/output pin.
	BRAKE00	I	MPF2	Brake input pin 0 of EPWMB.
	UART4_CTS	I	MPF3	Clear to Send input pin for UART4.
	PWM1_5	I/O	MPF4	PWM1_5 output/capture input.
	EPWM1_1	I/O	MPF5	PWM1_1 output/capture input.
	EBI_AD9	O	MPF7	EBI address/data bus bit 9.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
95	PB.8	I/O	MFP0	General purpose digital I/O pin.
	UART5_CTS	I	MPF1	Clear to Send input pin for UART5.
	EPWM1_2	I/O	MPF5	PWM1_2 output/capture input.
	EBI_AD10	O	MPF7	EBI address/data bus bit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
96	PB.9	I/O	MFP0	General purpose digital I/O pin.
	UART5_RTS	O	MPF1	Request to Send output pin for UART5.
	EPWM1_3	I/O	MPF5	PWM1_3 output/capture input.
	EBI_AD11	O	MPF7	EBI address/data bus bit 1.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
97	PB.10	I/O	MFP0	General purpose digital I/O pin.
	UART5_TXD	O	MPF1	Data transmitter output pin for UART5.
	EPWM1_4	I/O	MPF5	PWM1_4 output/capture input.
	EBI_AD12	O	MPF7	EBI address/data bus bit 12.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
98	PB.11	I/O	MFP0	General purpose digital I/O pin.
	UART5_RXD	I	MPF1	Data receiver input pin for UART5.
	EPWM1_5	I/O	MPF5	PWM1_5 output/capture input.
	EBI_AD13	O	MPF7	EBI address/data bus bit 13.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
99	PH.0	I/O	MFP0	General purpose digital I/O pin.
	I2C1_SCL	I/O	MPF1	I2C1 clock pin.
	UART4_RXD	I	MPF2	Data receiver input pin for UART4.
	CAN1_RXD	I	MPF3	CAN bus receiver1 input.
	INT7	I	MPF8	External interrupt7 input pin.
100	PH.1	I/O	MFP0	General purpose digital I/O pin.
	UART4_TXD	O	MPF1	Data transmitter output pin for UART4.
	I2C1_SDA	I/O	MPF2	I2C1 data input/output pin.
	CAN1_TXD	I	MPF3	CAN bus transmitter1 input.
101	PB.12	I/O	MFP0	General purpose digital I/O pin.
	UART4_RTS	O	MPF1	Request to Send output pin for UART4.
	SPI2_MISO1	I/O	MPF2	2nd SPI2 MISO (Master In, Slave Out) pin.
	CAN0_RXD	I	MPF3	CAN bus receiver0 input.
	EBI_AD14	O	MPF7	EBI address/data bus bit 14.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
102	PB.13	I/O	MFP0	General purpose digital I/O pin.
	UART4_CTS	I	MPF1	Clear to Send input pin for UART4.
	SPI2_MOSI1	I/O	MPF2	2nd SPI2 MOSI (Master Out, Slave In) pin.
	CAN0_TXD	I	MPF3	CAN bus transmitter0 input.
	EBI_AD15	O	MPF7	EBI address/data bus bit 15.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
103	PB.14	I/O	MFP0	General purpose digital I/O pin.
	I2S1_MCLK	O	MPF1	I2S1 master clock output pin.
	SC1_RST	O	MPF2	SmartCard1 reset pin.
	BRAKE01	I	MPF4	Brake input pin 1 of EPWMB.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
104	PB.15	I/O	MFP0	General purpose digital I/O pin.
	I2S1_DO	O	MPF1	I2S1 data output.
	SC1_DAT	I/O	MPF2	SmartCard1 data pin.
	BRAKE00	I	MPF4	Brake input pin 0 of EPWMB.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
105	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
106	VSS	P	MFP0	Ground pin for digital circuit.
107	LDO_CAP	P	MFP0	LDO output pin.
108	PC.0	I/O	MFP0	General purpose digital I/O pin.
	I2S1_DI	I	MPF1	I2S1 data input.
	SC1_DAT	I/O	MPF2	SmartCard1 data pin.
	UART4_RXD	I	MPF3	Data receiver input pin for UART4.
	EBI_MCLK	O	MPF7	EBI interface clock output pin.

	INT2	I	MPF8	External interrupt2 input pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
109	PC.1	I/O	MFP0	General purpose digital I/O pin.
	I2S1_BCLK	O	MPF1	I2S1 bit clock pin.
	SC1_CLK	O	MPF2	SmartCard1 clock pin.
	UART4_TXD	O	MPF3	Data transmitter output pin for UART4.
	TM3_CNT_OUT	I/O	MPF5	Timer3 event counter input/toggle output.
	EBI_AD13	O	MPF7	EBI address/data bus bit 13.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
110	PC.2	I/O	MFP0	General purpose digital I/O pin.
	I2S1_LRCK	O	MPF1	I2S1 left right channel clock.
	SC1_PWR	O	MPF2	SmartCard1 power pin.
	UART4_RTS	O	MPF3	Request to Send output pin for UART4.
	SPI0_SS0	I/O	MPF4	General purpose digital I/O pin.
	EBI_AD12	O	MPF7	EBI address/data bus bit 12.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
111	PC.3	I/O	MFP0	General purpose digital I/O pin.
	I2S1_MCLK	O	MPF1	I2S1 master clock output pin.
	SC1_CD	I	MPF2	SmartCard1 card detect pin.
	UART4_CTS	I	MPF3	Clear to Send input pin for UART4.
	SPI0_MISO1	I/O	MPF4	2nd SPI0 MISO (Master In, Slave Out) pin.
	QEI0_Z	I	MPF5	Quadrature encoder phase Z input of QEI Unit 0.
	EBI_AD11	O	MPF7	EBI address/data bus bit 11.
	ECAP0_IC2	O	MPF8	Input 2 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

112	PC.4	I/O	MFP0	General purpose digital I/O pin.
	I2S1_DO	O	MPF1	I2S1 data output.
	SC1_RST	O	MPF2	SmartCard1 reset pin.
	SPI0_MOSI1	I/O	MPF4	2nd SPI0 MOSI (Master Out, Slave In) pin.
	QEI0_B	I	MPF5	Quadrature encoder phase B input of QEI Unit 0.
	EBI_AD10	O	MPF7	EBI address/data bus bit 10.
	ECAP0_IC1	O	MPF8	Input 1 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
113	PC.5	I/O	MFP0	General purpose digital I/O pin.
	CLKO	O	MFP1	Clock Output Pin.
	QEI0_A	I	MPF5	Quadrature encoder phase A input of QEI Unit 0.
	EBI_MCLK	O	MPF7	EBI interface clock output pin.
	ECAP0_IC0	O	MPF8	Input 0 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
114	PC.6	I/O	MFP0	General purpose digital I/O pin.
	TM2_EXT	I	MPF1	Timer2 external counter input
	SPI0_MISO0	I/O	MPF4	1st SPI0 MISO (Master In, Slave Out) pin.
	TM2_CNT_OUT	I/O	MPF5	Timer2 event counter input/toggle output.
	EBI_AD9	O	MPF7	EBI address/data bus bit 9.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
115	PC.7	I/O	MFP0	General purpose digital I/O pin.
	TM1_EXT	I	MPF1	Timer1 external counter input
	SPI0_MOSI0	I/O	MPF4	1st SPI0 MOSI (Master Out, Slave In) pin.
	EBI_AD8	O	MPF7	EBI address/data bus bit 8.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

116	PC.8	I/O	MFP0	General purpose digital I/O pin.
	TM0_EXT	I	MPF1	Timer0 external counter input
	SPI0_CLK	O	MPF4	SPI0 serial clock pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
117	PF.2	I/O	MFP0	General purpose digital I/O pin.
	SPI3_SS0	I/O	MPF1	General purpose digital I/O pin.
	SD0_DAT3	I/O	MPF4	SD mode #0 data line bit 3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
118	PF.3	I/O	MFP0	General purpose digital I/O pin.
	SPI3_CLK	O	MPF1	SPI3 serial clock pin.
	SD0_DAT2	I/O	MPF4	SD mode #0 data line bit 2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
119	PF.4	I/O	MFP0	General purpose digital I/O pin.
	SPI3_MISO0	I/O	MPF1	1st SPI3 MISO (Master In, Slave Out) pin.
	SD0_DAT1	I/O	MPF4	SD mode #0 data line bit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
120	PF.5	I/O	MFP0	General purpose digital I/O pin.
	SPI3_MOSI0	I/O	MPF1	1st SPI3 MOSI (Master Out, Slave In) pin.
	SD0_DAT0	I/O	MPF4	SD mode #0 data line bit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
121	VSS	P	MFP0	Ground pin for digital circuit.
122	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
123	PF.6	I/O	MFP0	General purpose digital I/O pin.
	UART2_RXD	I	MPF1	Data receiver input pin for UART2.
	SD0_CDn	I	MPF4	SD mode #0 – card detect

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
124	PF.7	I/O	MFP0	General purpose digital I/O pin.
	UART2_TXD	O	MPF1	Data transmitter output pin for UART2.
	SD0_CMD	I/O	MPF4	SD mode #0 – command/response
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
125	PF.8	I/O	MFP0	General purpose digital I/O pin.
	UART2_RTS	O	MPF1	Request to Send output pin for UART2.
	SD0_CLK	O	MPF4	SD mode #0– clock.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
126	PH.2	I/O	MFP0	General purpose digital I/O pin.
	UART2_CTS	I	MPF1	Clear to Send input pin for UART2.
127	LDO_CAP	P	MFP0	LDO output pin.
128	VSS	P	MFP0	Ground pin for digital circuit.
129	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
130	PE.0	I/O	MFP0	General purpose digital I/O pin.
	ADC0_0	A	MPF1	ADC0 analog input.
	INT4	I	MPF8	External interrupt4 input pin.
131	PE.1	I/O	MFP0	General purpose digital I/O pin.
	ADC0_1	A	MPF1	ADC0 analog input.
	TM2_CNT_OUT	I/O	MPF3	Timer2 event counter input/toggle output.
132	PE.2	I/O	MFP0	General purpose digital I/O pin.
	ADC0_2	A	MPF1	ADC0 analog input.
	ACMP0_O	O	MPF2	Analog ccomparator0 output .
	SPI0_MISO0	I/O	MPF3	1st SPI0 MISO (Master In, Slave Out) pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

133	PE.3	I/O	MFP0	General purpose digital I/O pin.
	ADC0_3	A	MPF1	ADC0 analog input.
	ACMP0_P3	A	MPF2	Analog comparator0 positive input pin.
	SPI0_MOSI0	I/O	MPF3	1st SPI0 MOSI (Master Out, Slave In) pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
134	PE.4	I/O	MFP0	General purpose digital I/O pin.
	ADC0_4	A	MPF1	ADC0 analog input.
	ACMP0_P2	A	MPF2	Analog comparator0 positive input pin.
	SPI0_SS0	I/O	MPF3	General purpose digital I/O pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
135	PE.5	I/O	MFP0	General purpose digital I/O pin.
	ADC0_5	A	MPF1	ADC0 analog input.
	ACMP0_P1	A	MPF2	Analog comparator0 positive input pin.
	SPI0_CLK	O	MPF3	SPI0 serial clock pin.
	SD0_CDn	I	MPF4	SD mode #0 – card detect
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
136	PE.6	I/O	MFP0	General purpose digital I/O pin.
	ADC0_6	A	MPF1	ADC0 analog input.
	ACMP0_P0	A	MPF2	Analog comparator0 positive input pin.
	SPI0_MISO0	I/O	MPF3	1st SPI0 MISO (Master In, Slave Out) pin.
	SD0_CMD	I/O	MPF4	SD mode #0 – command/response
	EBI_nWR	O	MPF7	EBI write enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
137	PE.7	I/O	MFP0	General purpose digital I/O pin.
	ADC0_7	A	MPF1	ADC0 analog input.

	ACMP0_N	A	MPF2	Analog comparator0 negative input pin.
	SPI0_MOSI0	I/O	MPF3	1st SPI0 MOSI (Master Out, Slave In) pin.
	SD0_CLK	O	MPF4	SD mode #0– clock.
	EBI_nRD	O	MPF7	EBI read enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
138	AVSS	P	MFP0	Ground pin for digital circuit.
139	Vref	A	MFP0	Voltage reference input for ADC.
140	AVDD	P	MFP0	Power supply for internal analog circuit.
141	PE.8	I/O	MFP0	General purpose digital I/O pin.
	ADC1_0	A	MPF1	ADC1 analog input.
	ADC0_8	A	MPF1	ADC0 analog input.
	ACMP1_N	A	MPF2	Analog comparator1 negative input pin.
	TM1_CNT_OUT	I/O	MPF3	Timer1 event counter input/toggle output.
	SD0_DAT3	I/O	MPF4	SD mode #0 data line bit 3.
	EBI_ALE	O	MPF7	EBI address latch enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
142	PE.9	I/O	MFP0	General purpose digital I/O pin.
	ADC1_1	A	MPF1	ADC1 analog input.
	ADC0_9	A	MPF1	ADC0 analog input.
	ACMP1_P0	A	MPF2	Analog comparator1 positive input pin.
	SD0_DAT2	I/O	MPF4	SD mode #0 data line bit 2.
	EBI_nWRH	O	MPF7	EBI write enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
143	PE.10	I/O	MFP0	General purpose digital I/O pin.
	ADC1_2	A	MPF1	ADC1 analog input.
	ADC0_10	A	MPF1	ADC0 analog input.

	ACMP1_P1	A	MPF2	Analog comparator1 positive input pin.
	SPI0_MISO1	I/O	MPF3	2nd SPI0 MISO (Master In, Slave Out) pin.
	SD0_DAT1	I/O	MPF4	SD mode #0 data line bit 1.
	EBI_nWRL	O	MPF7	EBI write enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
144	PE.11	I/O	MFP0	General purpose digital I/O pin.
	ADC1_3	A	MPF1	ADC1 analog input.
	ADC0_11	A	MPF1	ADC0 analog input.
	ACMP1_P2	A	MPF2	Analog comparator1 positive input pin.
	SPI0_MOSI1	I/O	MPF3	2nd SPI0 MOSI (Master Out, Slave In) pin.
	SD0_DAT0	I/O	MPF4	SD mode #0 data line bit 0.
	ACMP2_P3	A	MPF5	Analog comparator2 positive input pin.
	EBI_nCS0	O	MPF7	EBI chip select 0 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

Note: Pin Type I = Digital Input, O = Digital Output; A = Analog Pin; P = Power Pin

4.3.5 NuMicro™ NUC472 Package LQFP 100-pin Description

MFP = Multi-function pin.

Pin No.	Pin Name	Type	MFP*	Description
1	PE.12	I/O	MFP0	General purpose digital I/O pin.
	ADC1_4	A	MPF1	ADC1 analog input.
	ACMP1_P3	A	MPF2	Analog comparator1 positive input pin.
	ACMP2_P2	A	MPF3	Analog comparator2 positive input pin.
	EBI_nCS1	O	MPF7	EBI chip select 1 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
2	PE.13	I/O	MFP0	General purpose digital I/O pin.
	ADC1_5	A	MPF1	ADC1 analog input.
	ACMP2_P1	A	MPF3	Analog comparator2 positive input pin.
	EBI_nCS2	O	MPF7	EBI chip select 2 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
3	PE.14	I/O	MFP0	General purpose digital I/O pin.
	ADC1_6	A	MPF1	ADC1 analog input.
	ACMP2_P0	A	MPF3	Analog comparator2 positive input pin.
	EBI_nCS3	O	MPF7	EBI chip select 3 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
4	PE.15	I/O	MFP0	General purpose digital I/O pin.
	ADC1_7	A	MPF1	ADC1 analog input.
	ACMP2_N	A	MPF3	Analog comparator2 negative input pin.
5	VSS	P	MFP0	Ground pin for digital circuit.
6	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
7	PC.12	I/O	MFP0	General purpose digital I/O pin.
	SPI1_SS0	I/O	MPF1	1st SPI1 slave select pin.

	SC4_CD	I	MPF2	SmartCard4 card detect pin.
	SD1_CDn	I	MPF4	SD mode #1 – card detect
	CAP_DATA7	I	MPF5	Image data input bus bit 7.
	EBI_A0	O	MPF7	EBI address bus bit0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
8	PC.13	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MOSI1	I/O	MPF1	2nd SPI1 MOSI (Master Out, Slave In) pin.
	SC4_RST	O	MPF2	SmartCard4 reset pin.
	SD1_CMD	I/O	MPF4	SD mode #1 – command/response
	CAP_DATA6	I	MPF5	Image data input bus bit 7.
	EBI_A1	O	MPF7	EBI address bus bit1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
9	PC.14	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MISO1	I/O	MPF1	2nd SPI1 MISO (Master In, Slave Out) pin.
	SC4_PWR	O	MPF2	SmartCard4 power pin.
	TM3_EXT	I	MPF3	Timer3 external counter input
	SD1_CLK	O	MPF4	SD mode #1 – clock.
	CAP_DATA5	I	MPF5	Image data input bus bit 7.
	EBI_A2	O	MPF7	EBI address bus bit2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
10	PC.15	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MOSI0	I/O	MPF1	1st SPI1 MOSI (Master Out, Slave In) pin.
	SC4_DAT	I/O	MPF2	SmartCard4 data pin.
	SD1_DAT3	I/O	MPF4	SD mode #1 data line bit 3.
	CAP_DATA4	I	MPF5	Image data input bus bit 7.
	EBI_A3	O	MPF7	EBI address bus bit3.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
11	PD.0	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MISO0	I/O	MPF1	1st SPI1 MISO (Master In, Slave Out) pin.
	SC4_CLK	O	MPF2	SmartCard4 clock pin.
	SD1_DAT2	I/O	MPF4	SD mode #1 data line bit 2.
	CAP_DATA3	I	MPF5	Image data input bus bit 7.
	EBI_A4	O	MPF7	EBI address bus bit4.
	INT3	I	MPF8	External interrupt3 input pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
12	PD.1	I/O	MFP0	General purpose digital I/O pin.
	SPI1_CLK	O	MPF1	SPI1 serial clock pin.
	TM0_CNT_OUT	I/O	MPF3	Timer0 event counter input/toggle output.
	SD1_DAT1	I/O	MPF4	SD mode #1 data line bit 1;
	CAP_DATA2	I	MPF5	Image data input bus bit 7.
	EBI_A5	O	MPF7	EBI address bus bit5.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
13	PD.3	I/O	MFP0	General purpose digital I/O pin.
	SC5_CLK	O	MPF1	SmartCard5 clock pin.
	I2C3_SDA	I/O	MPF2	I2C3 data input/output pin.
	ACMP2_O	O	MPF3	Analog ccomparator2 output.
	SD0_CDn	I	MPF4	SD mode #0 – card detect
	CAP_DATA0	I	MPF5	Image data input bus bit 7.
	EBI_A7	O	MPF7	EBI address bus bit7.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
14	PD.4	I/O	MFP0	General purpose digital I/O pin.

	SC5_CD	I	MPF1	SmartCard5 card detect pin.
	UART3_RXD	I	MPF2	Data receiver input pin for UART3.
	ACMP1_O	O	MPF3	Analog ccomparator1 output.
	CAP_SCLK	O	MPF5	Image capture interface sensor clock pin.
	EBI_A8	O	MPF7	EBI address bus bit8.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
15	PD.5	I/O	MFP0	General purpose digital I/O pin.
	SC5_RST	O	MPF1	SmartCard5 reset pin.
	UART3_TXD	O	MPF2	Data transmitter output pin for UART3.
	CAP_VSYNC	I	MPF5	Image capture interface VSYNC input pin.
	EBI_A9	O	MPF7	EBI address bus bit9.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
16	PD.6	I/O	MFP0	General purpose digital I/O pin.
	SC5_PWR	O	MPF1	SmartCard5 power pin.
	UART3_RTS	O	MPF2	Request to Send output pin for UART3.
	SD0_CMD	I/O	MPF4	SD mode #0 – command/response
	CAP_HSYNC	I	MPF5	Image capture interface HSYNC input pin.
	EBI_A10	O	MPF7	EBI address bus bit10.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
17	PD.7	I/O	MFP0	General purpose digital I/O pin.
	SC5_DAT	I/O	MPF1	SmartCard5 data pin.
	UART3_CTS	I	MPF2	Clear to Send input pin for UART3.
	SD0_CLK	O	MPF4	SD mode #0 – clock.
	CAP_PIXCLK	I	MPF5	Image capture interface pix clock input pin.
	EBI_A11	O	MPF7	EBI address bus bit11.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
18	PG.13	I/O	MFP0	General purpose digital I/O pin.
	XT1_IN	I	MPF1	External 4~24 MHz (high-speed) crystal input pin.
19	PG.12	I/O	MFP0	General purpose digital I/O pin.
	XT1_OUT	O	MPF1	External 4~24 MHz (high-speed) crystal output pin.
20	nRESET	I	MFP0	External reset input: active LOW, with an internal pull-up. Set this pin low reset to initial state.
21	LDO_CAP	P	MFP0	LDO output pin.
22	VSS	P	MFP0	Ground pin for digital circuit.
23	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
24	PG.10	I/O	MFP0	General purpose digital I/O pin.
	ICE_CLK	I	MPF1	Serial wired debugger clock pin
25	PG.11	I/O	MFP0	General purpose digital I/O pin.
	ICE_DAT	I/O	MPF1	Serial wired debugger data pin
26	PG.15	I/O	MFP0	General purpose digital I/O pin.
	X32K_IN	I	MPF1	External 32.768 kHz (low-speed) crystal input pin.
	I2C1_SCL	I/O	MPF3	I2C1 clock pin.
27	PG.14	I/O	MFP0	General purpose digital I/O pin.
	X32K_OUT	O	MPF1	External 32.768 kHz (low-speed) crystal output pin.
	I2C1_SDA	I/O	MPF3	I2C1 data input/output pin.
28	VBAT	P	MFP0	Battery power input pin.
29	PA.0	I/O	MFP0	General purpose digital I/O pin.
	TAMPER0	I/O	MPF1	Tamper detect pin 0.
	SC0_CD	I	MPF2	SmartCard0 card detect pin.
	CAN1_RXD	I	MPF3	CAN bus receiver1 input.
	INT0	I	MPF8	External interrupt0 input pin.

30	PA.1	I/O	MFP0	General purpose digital I/O pin.
	TAMPER1	I/O	MPF1	Tamper detect pin 1.
	SC5_CD	I	MPF2	SmartCard5 card detect pin.
	CAN1_TXD	I	MPF3	CAN bus transmitter1 input.
	EBI_A22	O	MPF7	EBI address bus bit22.
31	PA.2	I/O	MFP0	General purpose digital I/O pin.
	SC2_DAT	I/O	MPF1	SmartCard2 data pin.
	SPI3_MISO0	I/O	MPF2	1st SPI3 MISO (Master In, Slave Out) pin.
	I2S0_MCLK	O	MPF3	I2S0 master clock output pin.
	BRAKE11	I	MPF4	Brake input pin 1 of EPWM0.
	CAP_SFIELD	I	MPF5	Video input interface SFIELD input pin.
	EBI_A12	O	MPF7	EBI address bus bit12.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
32	PA.3	I/O	MFP0	General purpose digital I/O pin.
	SC2_CLK	O	MPF1	SmartCard2 clock pin.
	SPI3_MOSI0	I/O	MPF2	1st SPI3 MOSI (Master Out, Slave In) pin.
	I2S0_DO	O	MPF3	I2S0 data output.
	BRAKE10	I	MPF4	Brake input pin 0 of EPWM0.
	EBI_A13	O	MPF7	EBI address bus bit13.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
33	PA.4	I/O	MFP0	General purpose digital I/O pin.
	SC2_PWR	O	MPF1	SmartCard2 power pin.
	SPI3_CLK	O	MPF2	SPI3 serial clock pin.
	I2S0_DI	I	MPF3	I2S0 data input.
	QEI1_Z	I	MPF5	Quadrature encoder phase Z input of QEI Unit 1.
	EBI_A14	O	MPF7	EBI address bus bit14.

	ECAP1_IC2	I	MPF8	Input 2 of enhanced capture unit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
34	PA.5	I/O	MFP0	General purpose digital I/O pin.
	SC2_RST	O	MPF1	SmartCard2 reset pin.
	SPI3_SS0	I/O	MPF2	General purpose digital I/O pin.
	I2S0_BCLK	O	MPF3	I2S0 bit clock pin.
	PWM0_0	I/O	MPF4	PWM0_0 output/capture input.
	QEI1_B	I	MPF5	Quadrature encoder phase B input of QEI Unit 1.
	EBI_A15	O	MPF7	EBI address bus bit15.
	ECAP1_IC1	I	MPF8	Input 1 of enhanced capture unit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
35	PA.6	I/O	MFP0	General purpose digital I/O pin.
	SC2_CD	I	MPF1	SmartCard2 card detect pin.
	I2S0_LRCK	O	MPF3	I2S0 left right channel clock.
	PWM0_1	I/O	MPF4	PWM0_1 output/capture input.
	QEI1_A	I	MPF5	Quadrature encoder phase A input of QEI Unit 1.
	CAN1_TXD	I	MPF6	CAN bus transmitter1 input.
	EBI_A16	O	MPF7	EBI address bus bit16.
	ECAP1_IC0	I	MPF8	Input 0 of enhanced capture unit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
36	PA.7	I/O	MFP0	General purpose digital I/O pin.
	SC0_CLK	O	MPF2	SmartCard0 clock pin.
	SPI3_SS0	I/O	MPF3	General purpose digital I/O pin.
	PWM1_3	I/O	MPF4	PWM1_3 output/capture input.
	EPWM0_5	I/O	MPF5	PWM0_5 output/capture input.
	EBI_A17	O	MPF7	EBI address bus bit17.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
37	PA.8	I/O	MFP0	General purpose digital I/O pin.
	SC0_RST	O	MPF2	SmartCard0 reset pin.
	SPI3_CLK	O	MPF3	SPI3 serial clock pin.
	PWM1_2	I/O	MPF4	PWM1_2 output/capture input.
	EPWM0_4	I/O	MPF5	PWM0_4 output/capture input.
	EBI_A18	O	MPF7	EBI address bus bit18.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
38	PA.9	I/O	MFP0	General purpose digital I/O pin.
	SC0_PWR	O	MPF2	SmartCard0 power pin.
	SPI3_MISO0	I/O	MPF3	1st SPI3 MISO (Master In, Slave Out) pin.
	PWM1_1	I/O	MPF4	PWM1_1 output/capture input.
	EPWM0_3	I/O	MPF5	PWM0_3 output/capture input.
	EBI_A19	O	MPF7	EBI address bus bit19.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
39	PA.10	I/O	MFP0	General purpose digital I/O pin.
	SC0_DAT	I/O	MPF2	SmartCard0 data pin.
	SPI3_MOSI0	I/O	MPF3	1st SPI3 MOSI (Master Out, Slave In) pin.
	PWM1_0	I/O	MPF4	PWM1_0 output/capture input.
	EPWM0_2	I/O	MPF5	PWM0_2 output/capture input.
	EBI_A20	O	MPF7	EBI address bus bit20.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
40	PA.11	I/O	MFP0	General purpose digital I/O pin.
	UART0_RTS	O	MPF1	Request to Send output pin for UART0.
	SPI3_MISO1	I/O	MPF3	2nd SPI3 MISO (Master In, Slave Out) pin.

	PWM0_5	I/O	MPF4	PWM0_5 output/capture input.
	EPWM0_1	I/O	MPF5	PWM0_1 output/capture input.
	EBI_AD0	O	MPF7	EBI address/data bus bit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
41	PA.12	I/O	MFP0	General purpose digital I/O pin.
	UART0_CTS	I	MPF1	Clear to Send input pin for UART0.
	SPI3_MOSI1	I/O	MPF3	2nd SPI3 MOSI (Master Out, Slave In) pin.
	PWM0_4	I/O	MPF4	PWM0_4 output/capture input.
	EPWM0_0	I/O	MPF5	PWM0_0 output/capture input.
	EBI_AD1	O	MPF7	EBI address/data bus bit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
42	PA.13	I/O	MFP0	General purpose digital I/O pin.
	UART0_RXD	I	MPF1	Data receiver input pin for UART0.
	SC3_DAT	I/O	MPF3	SmartCard3 data pin.
	PWM1_4	I/O	MPF4	PWM1_4 output/capture input.
	EBI_AD2	O	MPF7	EBI address/data bus bit 2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
43	PA.14	I/O	MFP0	General purpose digital I/O pin.
	UART0_TXD	O	MPF1	Data transmitter output pin for UART0.
	SC3_CLK	O	MPF3	SmartCard3 clock pin.
	PWM1_5	I/O	MPF4	PWM1_5 output/capture input.
	EBI_AD3	O	MPF7	EBI address/data bus bit 3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
44	PA.15	I/O	MFP0	General purpose digital I/O pin.
	SC3_PWR	O	MPF1	SmartCard3 power pin.

	UART2_RTS	O	MPF2	Request to Send output pin for UART2.
	I2C0_SCL	I/O	MPF4	I2C0 clock pin.
	EBI_A21	O	MPF7	EBI address bus bit21.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
45	PC.9	I/O	MFP0	General purpose digital I/O pin.
	STADC	A	MPF1	ADC analog input.
	UART2_CTS	I	MPF2	Clear to Send input pin for UART2.
	SC3_RST	O	MPF3	SmartCard3 reset pin.
	I2C0_SDA	I/O	MPF4	I2C0 data input/output pin.
	CAP_DATA1	I	MPF5	Image data input bus bit 7.
	I2C3_SCL	I/O	MPF6	I2C3 clock pin.
	EBI_A22	O	MPF7	EBI address bus bit22.
	SD1_DAT0	I/O	MPF8	SD mode #1 data line bit 0.
	EBI_A6	O	MPF9	EBI address bus bit6.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
46	PC.10	I/O	MFP0	General purpose digital I/O pin.
	SC3_CD	I	MPF1	SmartCard3 card detect pin.
	UART2_RXD	I	MPF2	Data receiver input pin for UART2.
	PWM0_2	I/O	MPF4	PWM0_2 output/capture input.
	EBI_A23	O	MPF6	EBI address bus bit23.
	EBI_AD2	O	MPF7	EBI address/data bus bit 2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
47	PC.11	I/O	MFP0	General purpose digital I/O pin.
	UART2_TXD	O	MPF2	Data transmitter output pin for UART2.
	PWM0_3	I/O	MPF4	PWM0_3 output/capture input.
	EBI_A24	O	MPF6	EBI address bus bit24.

	EBI_AD3	O	MPF7	EBI address/data bus bit 3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
48	LDO_CAP	P	MFP0	LDO output pin.
49	VSS	P	MFP0	Ground pin for digital circuit.
50	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
51	VRES	A	MFP0	USB PHY VRES ground input pin. Add an 8.2K ohm resistor to VSSA.
52	VBUS	A	MFP0	USB PHY VBUS power input pin.
53	USB_VDD33_CAP	P	MFP0	Internal power regulator output 3.3V decoupling pin.
54	VSSA	P	MFP0	Ground pin for digital circuit. Add a Feritte Bead to digital ground VSS.
55	USB0_D-	A	MFP0	USB0 differential signal D+.
56	USB0_D+	A	MFP0	USB0 differential signal D+.
57	USB0_OTG_ID	I	MFP0	USB0OTG ID pin.
58	PB.0	I/O	MFP0	General purpose digital I/O pin.
	USB0_VBUS_ST	I	MPF1	USB0 external VBUS regulator status
	I2C4_SCL	I/O	MPF2	I2C4 clock pin.
	INT1	I	MPF8	External interrupt1 input pin.
59	PB.1	I/O	MFP0	General purpose digital I/O pin.
	USB0_VBUS_EN	O	MPF1	USB0 external VBUS regulator enable
	I2C4_SDA	I/O	MPF2	I2C4 data input/output pin.
	TM1_CNT_OUT	I/O	MPF3	Timer1 event counter input/toggle output.
60	PB.2	I/O	MFP0	General purpose digital I/O pin.
	UART1_RXD	I	MPF1	Data receiver input pin for UART1.
	SPI2_SS0	I/O	MPF2	General purpose digital I/O pin.
	USB1_D-	A	MPF3	USB1 differential signal D+.
	EBI_AD4	O	MPF7	EBI address/data bus bit 4.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
61	PB.3	I/O	MFP0	General purpose digital I/O pin.
	UART1_TXD	O	MPF1	Data transmitter output pin for UART1.
	SPI2_CLK	O	MPF2	SPI2 serial clock pin.
	USB1_D+	A	MPF3	USB1 differential signal D+.
	EBI_AD5	O	MPF7	EBI address/data bus bit 5.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
62	PB.4	I/O	MFP0	General purpose digital I/O pin.
	UART1_RTS	O	MPF1	Request to Send output pin for UART1.
	SPI2_MISO0	I/O	MPF2	1st SPI2 MISO (Master In, Slave Out) pin.
	UART4_RXD	I	MPF3	Data receiver input pin for UART4.
	TM0_CNT_OUT	I/O	MPF4	Timer0 event counter input/toggle output.
	EBI_AD6	O	MPF7	EBI address/data bus bit 6.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
63	PB.5	I/O	MFP0	General purpose digital I/O pin.
	UART1_CTS	I	MPF1	Clear to Send input pin for UART1.
	SPI2_MOSI0	I/O	MPF2	1st SPI2 MOSI (Master Out, Slave In) pin.
	UART4_TXD	O	MPF3	Data transmitter output pin for UART4.
	EBI_AD7	O	MPF7	EBI address/data bus bit 7.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
64	PB.6	I/O	MFP0	General purpose digital I/O pin.
	I2C2_SCL	I/O	MPF1	I2C2 clock pin.
	BRAKE01	I	MPF2	Brake input pin 1 of EPWMB.
	UART4_RTS	O	MPF3	Request to Send output pin for UART4.
	PWM1_4	I/O	MPF4	PWM1_4 output/capture input.

	EPWM1_0	I/O	MPF5	PWM1_0 output/capture input.
	EBI_AD8	O	MPF7	EBI address/data bus bit 8.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
65	PB.7	I/O	MFP0	General purpose digital I/O pin.
	I2C2_SDA	I/O	MPF1	I2C2 data input/output pin.
	BRAKE00	I	MPF2	Brake input pin 0 of EPWMB.
	UART4_CTS	I	MPF3	Clear to Send input pin for UART4.
	PWM1_5	I/O	MPF4	PWM1_5 output/capture input.
	EPWM1_1	I/O	MPF5	PWM1_1 output/capture input.
	EBI_AD9	O	MPF7	EBI address/data bus bit 9.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
66	PB.8	I/O	MFP0	General purpose digital I/O pin.
	UART5_CTS	I	MPF1	Clear to Send input pin for UART5.
	EPWM1_2	I/O	MPF5	PWM1_2 output/capture input.
	EBI_AD10	O	MPF7	EBI address/data bus bit 10.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
67	PB.9	I/O	MFP0	General purpose digital I/O pin.
	UART5_RTS	O	MPF1	Request to Send output pin for UART5.
	EPWM1_3	I/O	MPF5	PWM1_3 output/capture input.
	EBI_AD11	O	MPF7	EBI address/data bus bit 11.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
68	PB.10	I/O	MFP0	General purpose digital I/O pin.
	UART5_TXD	O	MPF1	Data transmitter output pin for UART5.
	EPWM1_4	I/O	MPF5	PWM1_4 output/capture input.
	EBI_AD12	O	MPF7	EBI address/data bus bit 12.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
69	PB.11	I/O	MFP0	General purpose digital I/O pin.
	UART5_RXD	I	MPF1	Data receiver input pin for UART5.
	EPWM1_5	I/O	MPF5	PWM1_5 output/capture input.
	EBI_AD13	O	MPF7	EBI address/data bus bit 13.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
70	PB.12	I/O	MFP0	General purpose digital I/O pin.
	UART4_RTS	O	MPF1	Request to Send output pin for UART4.
	SPI2_MISO1	I/O	MPF2	2nd SPI2 MISO (Master In, Slave Out) pin.
	CAN0_RXD	I	MPF3	CAN bus receiver0 input.
	EMAC_MII_MDC	O	MPF6	MII/RMII Management Data Clock.
	EBI_AD14	O	MPF7	EBI address/data bus bit 14.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
71	PB.13	I/O	MFP0	General purpose digital I/O pin.
	UART4_CTS	I	MPF1	Clear to Send input pin for UART4.
	SPI2_MOSI1	I/O	MPF2	2nd SPI2 MOSI (Master Out, Slave In) pin.
	CAN0_TXD	I	MPF3	CAN bus transmitter0 input.
	EMAC_MII_MDIO	I/O	MPF6	MII/RMII Management Data I/O.
	EBI_AD15	O	MPF7	EBI address/data bus bit 15.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
72	PB.14	I/O	MFP0	General purpose digital I/O pin.
	I2S1_MCLK	O	MPF1	I2S1 master clock output pin.
	SC1_RST	O	MPF2	SmartCard1 reset pin.
	BRAKE01	I	MPF4	Brake input pin 1 of EPWMB.
	EMAC_MII_MDC	O	MPF6	MII/RMII Management Data Clock.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
73	PB.15	I/O	MFP0	General purpose digital I/O pin.
	I2S1_DO	O	MPF1	I2S1 data output.
	SC1_DAT	I/O	MPF2	SmartCard1 data pin.
	BRAKE00	I	MPF4	Brake input pin 0 of EPWMB.
	EMAC_MII_MDIO	I/O	MPF6	MII/RMII Management Data I/O.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
74	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
75	PC.0	I/O	MFP0	General purpose digital I/O pin.
	I2S1_DI	I	MPF1	I2S1 data input.
	SC1_DAT	I/O	MPF2	SmartCard1 data pin.
	UART4_RXD	I	MPF3	Data receiver input pin for UART4.
	EMAC_REFCLK	I	MPF6	EMAC RMII mode clock input
	EBI_MCLK	O	MPF7	EBI interface clock output pin.
	INT2	I	MPF8	External interrupt2 input pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
76	PC.1	I/O	MFP0	General purpose digital I/O pin.
	I2S1_BCLK	O	MPF1	I2S1 bit clock pin.
	SC1_CLK	O	MPF2	SmartCard1 clock pin.
	UART4_TXD	O	MPF3	Data transmitter output pin for UART4.
	TM3_CNT_OUT	I/O	MPF5	Timer3 event counter input/toggle output.
	EMAC_MII_RXERR	I	MPF6	MII/RMII Receive Data error.
	EBI_AD13	O	MPF7	EBI address/data bus bit 13.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
77	PC.2	I/O	MFP0	General purpose digital I/O pin.

	I2S1_LRCK	O	MPF1	I2S1 left right channel clock.
	SC1_PWR	O	MPF2	SmartCard1 power pin.
	UART4_RTS	O	MPF3	Request to Send output pin for UART4.
	SPI0_SS0	I/O	MPF4	General purpose digital I/O pin.
	EMAC_MII_RXDV	I	MPF6	MII Receive Data Valid / RMII CRS_DV Input.
	EBI_AD12	O	MPF7	EBI address/data bus bit 12.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
78	PC.3	I/O	MFP0	General purpose digital I/O pin.
	I2S1_MCLK	O	MPF1	I2S1 master clock output pin.
	SC1_CD	I	MPF2	SmartCard1 card detect pin.
	UART4_CTS	I	MPF3	Clear to Send input pin for UART4.
	SPI0_MISO1	I/O	MPF4	2nd SPI0 MISO (Master In, Slave Out) pin.
	QEI0_Z	I	MPF5	Quadrature encoder phase Z input of QEI Unit 0.
	EMAC_MII_RXD1	I	MPF6	MII/RMII Receive Data Bus Bit 1.
	EBI_AD11	O	MPF7	EBI address/data bus bit 11.
	ECAP0_IC2	O	MPF8	Input 2 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
79	PC.4	I/O	MFP0	General purpose digital I/O pin.
	I2S1_DO	O	MPF1	I2S1 data output.
	SC1_RST	O	MPF2	SmartCard1 reset pin.
	SPI0_MOSI1	I/O	MPF4	2nd SPI0 MOSI (Master Out, Slave In) pin.
	QEI0_B	I	MPF5	Quadrature encoder phase B input of QEI Unit 0.
	EMAC_MII_RXD0	I	MPF6	MII/RMII Receive Data Bus Bit 0.
	EBI_AD10	O	MPF7	EBI address/data bus bit 10.
	ECAP0_IC1	O	MPF8	Input 1 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

80	PC.5	I/O	MFP0	General purpose digital I/O pin.
	CLKO	O	MFP1	Clock Output Pin.
	QEI0_A	I	MPF5	Quadrature encoder phase A input of QEI Unit 0.
	EMAC_MII_RXCLK	I	MPF6	MII Receive Clock Input.
	EBI_MCLK	O	MPF7	EBI interface clock output pin.
	ECAP0_IC0	O	MPF8	Input 0 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
81	PC.6	I/O	MFP0	General purpose digital I/O pin.
	TM2_EXT	I	MPF1	Timer2 external counter input
	SPI0_MISO0	I/O	MPF4	1st SPI0 MISO (Master In, Slave Out) pin.
	TM2_CNT_OUT	I/O	MPF5	Timer2 event counter input/toggle output.
	EMAC_MII_TXD0	O	MPF6	MII/RMII Transmit Data Bus bit 0.
	EBI_AD9	O	MPF7	EBI address/data bus bit 9.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
82	PC.7	I/O	MFP0	General purpose digital I/O pin.
	TM1_EXT	I	MPF1	Timer1 external counter input
	SPI0_MOSI0	I/O	MPF4	1st SPI0 MOSI (Master Out, Slave In) pin.
	EMAC_MII_TXD1	O	MPF6	MII/RMII Transmit Data Bus bit 1.
	EBI_AD8	O	MPF7	EBI address/data bus bit 8.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
83	PC.8	I/O	MFP0	General purpose digital I/O pin.
	TM0_EXT	I	MPF1	Timer0 external counter input
	SPI0_CLK	O	MPF4	SPI0 serial clock pin.
	EMAC_MII_TXEN	O	MPF6	MII/RMII Transmit Enable.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

84	LDO_CAP	P	MFP0	LDO output pin.
85	VSS	P	MFP0	Ground pin for digital circuit.
86	PE.0	I/O	MFP0	General purpose digital I/O pin.
	ADC0_0	A	MPF1	ADC0 analog input.
	INT4	I	MPF8	External interrupt4 input pin.
87	PE.1	I/O	MFP0	General purpose digital I/O pin.
	ADC0_1	A	MPF1	ADC0 analog input.
	TM2_CNT_OUT	I/O	MPF3	Timer2 event counter input/toggle output.
88	PE.2	I/O	MFP0	General purpose digital I/O pin.
	ADC0_2	A	MPF1	ADC0 analog input.
	ACMP0_O	O	MPF2	Analog ccomparator0 output .
	SPI0_MISO0	I/O	MPF3	1st SPI0 MISO (Master In, Slave Out) pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
89	PE.3	I/O	MFP0	General purpose digital I/O pin.
	ADC0_3	A	MPF1	ADC0 analog input.
	ACMP0_P3	A	MPF2	Analog comparator0 positive input pin.
	SPI0_MOSI0	I/O	MPF3	1st SPI0 MOSI (Master Out, Slave In) pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
90	PE.4	I/O	MFP0	General purpose digital I/O pin.
	ADC0_4	A	MPF1	ADC0 analog input.
	ACMP0_P2	A	MPF2	Analog comparator0 positive input pin.
	SPI0_SS0	I/O	MPF3	General purpose digital I/O pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
91	PE.5	I/O	MFP0	General purpose digital I/O pin.
	ADC0_5	A	MPF1	ADC0 analog input.
	ACMP0_P1	A	MPF2	Analog comparator0 positive input pin.

	SPI0_CLK	O	MPF3	SPI0 serial clock pin.
	SD0_CDn	I	MPF4	SD mode #0 – card detect
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
92	PE.6	I/O	MFP0	General purpose digital I/O pin.
	ADC0_6	A	MPF1	ADC0 analog input.
	ACMP0_P0	A	MPF2	Analog comparator0 positive input pin.
	SPI0_MISO0	I/O	MPF3	1st SPI0 MISO (Master In, Slave Out) pin.
	SD0_CMD	I/O	MPF4	SD mode #0 – command/response
	EBI_nWR	O	MPF7	EBI write enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
93	PE.7	I/O	MFP0	General purpose digital I/O pin.
	ADC0_7	A	MPF1	ADC0 analog input.
	ACMP0_N	A	MPF2	Analog comparator0 negative input pin.
	SPI0_MOSI0	I/O	MPF3	1st SPI0 MOSI (Master Out, Slave In) pin.
	SD0_CLK	O	MPF4	SD mode #0 – clock.
	EBI_nRD	O	MPF7	EBI read enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
94	AVSS	P	MFP0	Ground pin for digital circuit.
95	Vref	A	MFP0	Voltage reference input for ADC.
96	AVDD	P	MFP0	Power supply for internal analog circuit.
97	PE.8	I/O	MFP0	General purpose digital I/O pin.
	ADC1_0	A	MPF1	ADC1 analog input.
	ADC0_8	A	MPF1	ADC0 analog input.
	ACMP1_N	A	MPF2	Analog comparator1 negative input pin.
	TM1_CNT_OUT	I/O	MPF3	Timer1 event counter input/toggle output.
	SD0_DAT3	I/O	MPF4	SD mode #0 data line bit 3.

	EBI_ALE	O	MPF7	EBI address latch enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
98	PE.9	I/O	MFP0	General purpose digital I/O pin.
	ADC1_1	A	MPF1	ADC1 analog input.
	ADC0_9	A	MPF1	ADC0 analog input.
	ACMP1_P0	A	MPF2	Analog comparator1 positive input pin.
	SD0_DAT2	I/O	MPF4	SD mode #0 data line bit 2.
	EBI_nWRH	O	MPF7	EBI write enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
99	PE.10	I/O	MFP0	General purpose digital I/O pin.
	ADC1_2	A	MPF1	ADC1 analog input.
	ADC0_10	A	MPF1	ADC0 analog input.
	ACMP1_P1	A	MPF2	Analog comparator1 positive input pin.
	SPI0_MISO1	I/O	MPF3	2nd SPI0 MISO (Master In, Slave Out) pin.
	SD0_DAT1	I/O	MPF4	SD mode #0 data line bit 1.
	EBI_nWRL	O	MPF7	EBI write enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
100	PE.11	I/O	MFP0	General purpose digital I/O pin.
	ADC1_3	A	MPF1	ADC1 analog input.
	ADC0_11	A	MPF1	ADC0 analog input.
	ACMP1_P2	A	MPF2	Analog comparator1 positive input pin.
	SPI0_MOSI1	I/O	MPF3	2nd SPI0 MOSI (Master Out, Slave In) pin.
	SD0_DAT0	I/O	MPF4	SD mode #0 data line bit 0.
	ACMP2_P3	A	MPF5	Analog comparator2 positive input pin.
	EBI_nCS0	O	MPF7	EBI chip select 0 enable output pin.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

Note: Pin Type I = Digital Input, O = Digital Output; A = Analog Pin; P = Power Pin

4.3.6 NuMicro™ NUC472 Package LQFP 128-pin Description

MFP = Multi-function pin.

Pin No.	Pin Name	Type	MFP*	Description
1	PE.12	I/O	MFP0	General purpose digital I/O pin.
	ADC1_4	A	MPF1	ADC1 analog input.
	ACMP1_P3	A	MPF2	Analog comparator1 positive input pin.
	ACMP2_P2	A	MPF3	Analog comparator2 positive input pin.
	EBI_nCS1	O	MPF7	EBI chip select 1 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
2	PE.13	I/O	MFP0	General purpose digital I/O pin.
	ADC1_5	A	MPF1	ADC1 analog input.
	ACMP2_P1	A	MPF3	Analog comparator2 positive input pin.
	EBI_nCS2	O	MPF7	EBI chip select 2 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
3	PE.14	I/O	MFP0	General purpose digital I/O pin.
	ADC1_6	A	MPF1	ADC1 analog input.
	ACMP2_P0	A	MPF3	Analog comparator2 positive input pin.
	EBI_nCS3	O	MPF7	EBI chip select 3 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
4	PE.15	I/O	MFP0	General purpose digital I/O pin.
	ADC1_7	A	MPF1	ADC1 analog input.
	ACMP2_N	A	MPF3	Analog comparator2 negative input pin.
5	PF.9	I/O	MFP0	General purpose digital I/O pin.
	OPA0_IN+	I/O	MPF1	General purpose digital I/O pin.
	PWM0_0	I/O	MPF4	PWM0_0 output/capture input.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

6	PF.10	I/O	MFP0	General purpose digital I/O pin.
	OPA0_IN-	I/O	MPF1	General purpose digital I/O pin.
	PWM0_1	I/O	MPF4	PWM0_1 output/capture input.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
7	PF.11	I/O	MFP0	General purpose digital I/O pin.
	OPA0_O	O	MPF1	Operational amplifier output pin
	UART1_RTS	O	MPF2	Request to Send output pin for UART1.
8	PF.12	I/O	MFP0	General purpose digital I/O pin.
	OPA1_IN+	I/O	MPF1	General purpose digital I/O pin.
	UART1_CTS	I	MPF2	Clear to Send input pin for UART1.
9	PF.13	I/O	MFP0	General purpose digital I/O pin.
	OPA1_IN-	I/O	MPF1	General purpose digital I/O pin.
	UART1_TXD	O	MPF2	Data transmitter output pin for UART1.
10	PF.14	I/O	MFP0	General purpose digital I/O pin.
	OPA1_O	O	MPF1	Operational amplifier output pin
	UART1_RXD	I	MPF2	Data receiver input pin for UART1.
11	VSS	P	MFP0	Ground pin for digital circuit.
12	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
13	PC.12	I/O	MFP0	General purpose digital I/O pin.
	SPI1_SS0	I/O	MPF1	1st SPI1 slave select pin.
	SC4_CD	I	MPF2	SmartCard4 card detect pin.
	SD1_CDn	I	MPF4	SD mode #1 – card detect
	CAP_DATA7	I	MPF5	Image data input bus bit 7.
	EBI_A0	O	MPF7	EBI address bus bit0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
14	PC.13	I/O	MFP0	General purpose digital I/O pin.

	SPI1_MOSI1	I/O	MPF1	2nd SPI1 MOSI (Master Out, Slave In) pin.
	SC4_RST	O	MPF2	SmartCard4 reset pin.
	SD1_CMD	I/O	MPF4	SD mode #1 – command/response
	CAP_DATA6	I	MPF5	Image data input bus bit 7.
	EBI_A1	O	MPF7	EBI address bus bit1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
15	PC.14	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MISO1	I/O	MPF1	2nd SPI1 MISO (Master In, Slave Out) pin.
	SC4_PWR	O	MPF2	SmartCard4 power pin.
	TM3_EXT	I	MPF3	Timer3 external counter input
	SD1_CLK	O	MPF4	SD mode #1– clock.
	CAP_DATA5	I	MPF5	Image data input bus bit 7.
	EBI_A2	O	MPF7	EBI address bus bit2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
16	PC.15	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MOSI0	I/O	MPF1	1st SPI1 MOSI (Master Out, Slave In) pin.
	SC4_DAT	I/O	MPF2	SmartCard4 data pin.
	SD1_DAT3	I/O	MPF4	SD mode #1 data line bit 3.
	CAP_DATA4	I	MPF5	Image data input bus bit 7.
	EBI_A3	O	MPF7	EBI address bus bit3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
17	PD.0	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MISO0	I/O	MPF1	1st SPI1 MISO (Master In, Slave Out) pin.
	SC4_CLK	O	MPF2	SmartCard4 clock pin.
	SD1_DAT2	I/O	MPF4	SD mode #1 data line bit 2;
	CAP_DATA3	I	MPF5	Image data input bus bit 7.

	EBI_A4	O	MPF7	EBI address bus bit4.
	INT3	I	MPF8	External interrupt3 input pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
18	PD.1	I/O	MFP0	General purpose digital I/O pin.
	SPI1_CLK	O	MPF1	SPI1 serial clock pin.
	TM0_CNT_OUT	I/O	MPF3	Timer0 event counter input/toggle output.
	SD1_DAT1	I/O	MPF4	SD mode #1 data line bit 1.
	CAP_DATA2	I	MPF5	Image data input bus bit 7.
	EBI_A5	O	MPF7	EBI address bus bit5.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
19	PD.2	I/O	MFP0	General purpose digital I/O pin.
	STADC	A	MPF1	ADC analog input.
	I2C3_SCL	I/O	MPF2	I2C3 clock pin.
	SD1_DAT0	I/O	MPF4	SD mode #1 data line bit 0.
	CAP_DATA1	I	MPF5	Image data input bus bit 7.
	EBI_A6	O	MPF7	EBI address bus bit6.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
20	PD.3	I/O	MFP0	General purpose digital I/O pin.
	SC5_CLK	O	MPF1	SmartCard5 clock pin.
	I2C3_SDA	I/O	MPF2	I2C3 data input/output pin.
	ACMP2_O	O	MPF3	Analog ccomparator2 output .
	SD0_CDn	I	MPF4	SD mode #0 – card detect
	CAP_DATA0	I	MPF5	Image data input bus bit 7.
	EBI_A7	O	MPF7	EBI address bus bit7.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

21	PD.4	I/O	MFP0	General purpose digital I/O pin.
	SC5_CD	I	MPF1	SmartCard5 card detect pin.
	UART3_RXD	I	MPF2	Data receiver input pin for UART3.
	ACMP1_O	O	MPF3	Analog ccomparator1 output .
	CAP_SCLK	O	MPF5	Image capture interface sensor clock pin.
	EBI_A8	O	MPF7	EBI address bus bit8.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
22	PD.5	I/O	MFP0	General purpose digital I/O pin.
	SC5_RST	O	MPF1	SmartCard5 reset pin.
	UART3_TXD	O	MPF2	Data transmitter output pin for UART3.
	CAP_VSYNC	I	MPF5	Image capture interface VSYNC input pin.
	EBI_A9	O	MPF7	EBI address bus bit9.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
23	PD.6	I/O	MFP0	General purpose digital I/O pin.
	SC5_PWR	O	MPF1	SmartCard5 power pin.
	UART3_RTS	O	MPF2	Request to Send output pin for UART3.
	SD0_CMD	I/O	MPF4	SD mode #0 – command/response
	CAP_HSYNC	I	MPF5	Image capture interface HSYNC input pin.
	EBI_A10	O	MPF7	EBI address bus bit10.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
24	PD.7	I/O	MFP0	General purpose digital I/O pin.
	SC5_DAT	I/O	MPF1	SmartCard5 data pin.
	UART3_CTS	I	MPF2	Clear to Send input pin for UART3.
	SD0_CLK	O	MPF4	SD mode #0 – clock.
	CAP_PIXCLK	I	MPF5	Image capture interface pix clock input pin.
	EBI_A11	O	MPF7	EBI address bus bit11.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
25	PG.13	I/O	MFP0	General purpose digital I/O pin.
	XT1_IN	I	MPF1	External 4~24 MHz (high-speed) crystal input pin.
26	PG.12	I/O	MFP0	General purpose digital I/O pin.
	XT1_OUT	O	MPF1	External 4~24 MHz (high-speed) crystal output pin.
27	nRESET	I	MFP0	External reset input: active LOW, with an internal pull-up. Set this pin low reset to initial state.
28	LDO_CAP	P	MFP0	LDO output pin.
29	VSS	P	MFP0	Ground pin for digital circuit.
30	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
31	PG.10	I/O	MFP0	General purpose digital I/O pin.
	ICE_CLK	I	MPF1	Serial wired debugger clock pin
32	PG.11	I/O	MFP0	General purpose digital I/O pin.
	ICE_DAT	I/O	MPF1	Serial wired debugger data pin
33	PG.15	I/O	MFP0	General purpose digital I/O pin.
	X32K_IN	I	MPF1	External 32.768 kHz (low-speed) crystal input pin.
	I2C1_SCL	I/O	MPF3	I2C1 clock pin.
34	PG.14	I/O	MFP0	General purpose digital I/O pin.
	X32K_OUT	O	MPF1	External 32.768 kHz (low-speed) crystal output pin.
	I2C1_SDA	I/O	MPF3	I2C1 data input/output pin.
35	VBAT	P	MFP0	Battery power input pin.
36	PA.0	I/O	MFP0	General purpose digital I/O pin.
	TAMPER0	I/O	MPF1	Tamper detect pin 0.
	SC0_CD	I	MPF2	SmartCard0 card detect pin.
	CAN1_RXD	I	MPF3	CAN bus receiver1 input.
	INT0	I	MPF8	External interrupt0 input pin.

37	PA.1	I/O	MFP0	General purpose digital I/O pin.
	TAMPER1	I/O	MPF1	Tamper detect pin 1.
	SC5_CD	I	MPF2	SmartCard5 card detect pin.
	CAN1_TXD	I	MPF3	CAN bus transmitter1 input.
	EBI_A22	O	MPF7	EBI address bus bit22.
38	PD.8	I/O	MFP0	General purpose digital I/O pin.
	SPI3_MISO1	I/O	MPF1	2nd SPI3 MISO (Master In, Slave Out) pin.
	I2C0_SCL	I/O	MPF2	I2C0 clock pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
39	PD.9	I/O	MFP0	General purpose digital I/O pin.
	SPI3_MOSI1	I/O	MPF1	2nd SPI3 MOSI (Master Out, Slave In) pin.
	I2C0_SDA	I/O	MPF2	I2C0 data input/output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
40	PA.2	I/O	MFP0	General purpose digital I/O pin.
	SC2_DAT	I/O	MPF1	SmartCard2 data pin.
	SPI3_MISO0	I/O	MPF2	1st SPI3 MISO (Master In, Slave Out) pin.
	I2S0_MCLK	O	MPF3	I2S0 master clock output pin.
	BRAKE11	I	MPF4	Brake input pin 1 of EPWM0_.
	CAP_SFIELD	I	MPF5	Video input interface SFIELD input pin.
	EBI_A12	O	MPF7	EBI address bus bit12.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
41	PA.3	I/O	MFP0	General purpose digital I/O pin.
	SC2_CLK	O	MPF1	SmartCard2 clock pin.
	SPI3_MOSI0	I/O	MPF2	1st SPI3 MOSI (Master Out, Slave In) pin.
	I2S0_DO	O	MPF3	I2S0 data output.
	BRAKE10	I	MPF4	Brake input pin 0 of EPWM0_.

	EBI_A13	O	MPF7	EBI address bus bit13.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
42	PA.4	I/O	MFP0	General purpose digital I/O pin.
	SC2_PWR	O	MPF1	SmartCard2 power pin.
	SPI3_CLK	O	MPF2	SPI3 serial clock pin.
	I2S0_DI	I	MPF3	I2S0 data input.
	QEI1_Z	I	MPF5	Quadrature encoder phase Z input of QEI Unit 1.
	EBI_A14	O	MPF7	EBI address bus bit14.
	ECAP1_IC2	I	MPF8	Input 2 of enhanced capture unit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
43	PA.5	I/O	MFP0	General purpose digital I/O pin.
	SC2_RST	O	MPF1	SmartCard2 reset pin.
	SPI3_SS0	I/O	MPF2	General purpose digital I/O pin.
	I2S0_BCLK	O	MPF3	I2S0 bit clock pin.
	PWM0_0	I/O	MPF4	PWM0_0 output/capture input.
	QEI1_B	I	MPF5	Quadrature encoder phase B input of QEI Unit 1.
	EBI_A15	O	MPF7	EBI address bus bit15.
	ECAP1_IC1	I	MPF8	Input 1 of enhanced capture unit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
44	PA.6	I/O	MFP0	General purpose digital I/O pin.
	SC2_CD	I	MPF1	SmartCard2 card detect pin.
	I2S0_LRCK	O	MPF3	I2S0 left right channel clock.
	PWM0_1	I/O	MPF4	PWM0_1 output/capture input.
	QEI1_A	I	MPF5	Quadrature encoder phase A input of QEI Unit 1.
	CAN1_TXD	I	MPF6	CAN bus transmitter1 input.
	EBI_A16	O	MPF7	EBI address bus bit16.

	ECAP1_IC0	I	MPF8	Input 0 of enhanced capture unit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
45	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
46	VSS	P	MFP0	Ground pin for digital circuit.
47	PA.7	I/O	MFP0	General purpose digital I/O pin.
	SC0_CLK	O	MPF2	SmartCard0 clock pin.
	SPI3_SS0	I/O	MPF3	General purpose digital I/O pin.
	PWM1_3	I/O	MPF4	PWM1_3 output/capture input.
	EPWM0_5	I/O	MPF5	PWM0_5 output/capture input.
	EBI_A17	O	MPF7	EBI address bus bit17.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
48	PA.8	I/O	MFP0	General purpose digital I/O pin.
	SC0_RST	O	MPF2	SmartCard0 reset pin.
	SPI3_CLK	O	MPF3	SPI3 serial clock pin.
	PWM1_2	I/O	MPF4	PWM1_2 output/capture input.
	EPWM0_4	I/O	MPF5	PWM0_4 output/capture input.
	EBI_A18	O	MPF7	EBI address bus bit18.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
49	PA.9	I/O	MFP0	General purpose digital I/O pin.
	SC0_PWR	O	MPF2	SmartCard0 power pin.
	SPI3_MISO0	I/O	MPF3	1st SPI3 MISO (Master In, Slave Out) pin.
	PWM1_1	I/O	MPF4	PWM1_1 output/capture input.
	EPWM0_3	I/O	MPF5	PWM0_3 output/capture input.
	EBI_A19	O	MPF7	EBI address bus bit19.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

50	PA.10	I/O	MFP0	General purpose digital I/O pin.
	SC0_DAT	I/O	MPF2	SmartCard0 data pin.
	SPI3_MOSI0	I/O	MPF3	1st SPI3 MOSI (Master Out, Slave In) pin.
	PWM1_0	I/O	MPF4	PWM1_0 output/capture input.
	EPWM0_2	I/O	MPF5	PWM0_2 output/capture input.
	EBI_A20	O	MPF7	EBI address bus bit20.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
51	PA.11	I/O	MFP0	General purpose digital I/O pin.
	UART0_RTS	O	MPF1	Request to Send output pin for UART0.
	SPI3_MISO1	I/O	MPF3	2nd SPI3 MISO (Master In, Slave Out) pin.
	PWM0_5	I/O	MPF4	PWM0_5 output/capture input.
	EPWM0_1	I/O	MPF5	PWM0_1 output/capture input.
	EBI_AD0	O	MPF7	EBI address/data bus bit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
52	PA.12	I/O	MFP0	General purpose digital I/O pin.
	UART0_CTS	I	MPF1	Clear to Send input pin for UART0.
	SPI3_MOSI1	I/O	MPF3	2nd SPI3 MOSI (Master Out, Slave In) pin.
	PWM0_4	I/O	MPF4	PWM0_4 output/capture input.
	EPWM0_0	I/O	MPF5	PWM0_0 output/capture input.
	EBI_AD1	O	MPF7	EBI address/data bus bit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
53	PA.13	I/O	MFP0	General purpose digital I/O pin.
	UART0_RXD	I	MPF1	Data receiver input pin for UART0.
	SC3_DAT	I/O	MPF3	SmartCard3 data pin.
	PWM1_4	I/O	MPF4	PWM1_4 output/capture input.
	EBI_AD2	O	MPF7	EBI address/data bus bit 2.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
54	PA.14	I/O	MFP0	General purpose digital I/O pin.
	UART0_TXD	O	MPF1	Data transmitter output pin for UART0.
	SC3_CLK	O	MPF3	SmartCard3 clock pin.
	PWM1_5	I/O	MPF4	PWM1_5 output/capture input.
	EBI_AD3	O	MPF7	EBI address/data bus bit 3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
55	PD.10	I/O	MFP0	General purpose digital I/O pin.
	SC3_DAT	I/O	MPF1	SmartCard3 data pin.
	I2C4_SCL	I/O	MPF2	I2C4 clock pin.
56	PD.11	I/O	MFP0	General purpose digital I/O pin.
	SC3_RST	O	MPF1	SmartCard3 reset pin.
	TM3_CNT_OUT	I/O	MPF3	Timer3 event counter input/toggle output.
57	PD.12	I/O	MFP0	General purpose digital I/O pin.
	SC3_CLK	O	MPF1	SmartCard3 clock pin.
	I2C4_SDA	I/O	MPF2	I2C4 data input/output pin.
58	PA.15	I/O	MFP0	General purpose digital I/O pin.
	SC3_PWR	O	MPF1	SmartCard3 power pin.
	UART2_RTS	O	MPF2	Request to Send output pin for UART2.
	I2C0_SCL	I/O	MPF4	I2C0 clock pin.
	EBI_A21	O	MPF7	EBI address bus bit21.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
59	PC.9	I/O	MFP0	General purpose digital I/O pin.
	STADC	A	MPF1	ADC analog input.
	UART2_CTS	I	MPF2	Clear to Send input pin for UART2.
	SC3_RST	O	MPF3	SmartCard3 reset pin.

	I2C0_SDA	I/O	MPF4	I2C0 data input/output pin.
	CAP_DATA1	I	MPF5	Image data input bus bit 7.
	I2C3_SCL	I/O	MPF6	I2C3 clock pin.
	EBI_A22	O	MPF7	EBI address bus bit22.
	SD1_DAT0	I/O	MPF8	SD mode #1 data line bit 0.
	EBI_A6	O	MPF9	EBI address bus bit6.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
60	PC.10	I/O	MFP0	General purpose digital I/O pin.
	SC3_CD	I	MPF1	SmartCard3 card detect pin.
	UART2_RXD	I	MPF2	Data receiver input pin for UART2.
	PWM0_2	I/O	MPF4	PWM0_2 output/capture input.
	EBI_A23	O	MPF6	EBI address bus bit23.
	EBI_AD2	O	MPF7	EBI address/data bus bit 2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
61	PC.11	I/O	MFP0	General purpose digital I/O pin.
	UART2_TXD	O	MPF2	Data transmitter output pin for UART2.
	PWM0_3	I/O	MPF4	PWM0_3 output/capture input.
	EBI_A24	O	MPF6	EBI address bus bit24.
	EBI_AD3	O	MPF7	EBI address/data bus bit 3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
62	LDO_CAP	P	MFP0	LDO output pin.
63	VSS	P	MFP0	Ground pin for digital circuit.
64	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
65	PD.13	I/O	MFP0	General purpose digital I/O pin.
	SPI1_SS0	I/O	MPF1	1st SPI1 slave select pin.

	UART5_CTS	I	MPF2	Clear to Send input pin for UART5.
	ECAP0_IC2	O	MPF3	Input 2 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
66	PD.14	I/O	MFP0	General purpose digital I/O pin.
	SPI1_CLK	O	MPF1	SPI1 serial clock pin.
	UART5_RTS	O	MPF2	Request to Send output pin for UART5.
	ECAP0_IC1	O	MPF3	Input 1 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
67	PD.15	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MISO0	I/O	MPF1	1st SPI1 MISO (Master In, Slave Out) pin.
	UART5_TXD	O	MPF2	Data transmitter output pin for UART5.
	ECAP0_IC0	O	MPF3	Input 0 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
68	PF.0	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MOSI0	I/O	MPF1	1st SPI1 MOSI (Master Out, Slave In) pin.
	UART5_RXD	I	MPF2	Data receiver input pin for UART5.
	INT5	I	MPF8	External interrupt5 input pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
69	VRES	A	MFP0	USB PHY VRES ground input pin. Add an 8.2K ohm resistor to VSSA.
70	VBUS	A	MFP0	USB PHY VBUS power input pin.
71	USB_VDD33_CAP	P	MFP0	Internal power regulator output 3.3V decoupling pin.
72	VSSA	P	MFP0	Ground pin for digital circuit. Add a Feritte Bead to digital ground VSS.
73	USB0_D-	A	MFP0	USB0 differential signal D+.
74	USB0_D+	A	MFP0	USB0 differential signal D+.

75	USB0_OTG_ID	I	MFP0	USB0OTG ID pin.
76	PB.0	I/O	MFP0	General purpose digital I/O pin.
	USB0_VBUS_ST	I	MPF1	USB0 external VBUS regulator status
	I2C4_SCL	I/O	MPF2	I2C4 clock pin.
	INT1	I	MPF8	External interrupt1 input pin.
77	PB.1	I/O	MFP0	General purpose digital I/O pin.
	USB0_VBUS_EN	O	MPF1	USB0 external VBUS regulator enable
	I2C4_SDA	I/O	MPF2	I2C4 data input/output pin.
	TM1_CNT_OUT	I/O	MPF3	Timer1 event counter input/toggle output.
78	PF.1	I/O	MFP0	General purpose digital I/O pin.
	SPI2_MOSI1	I/O	MPF1	2nd SPI2 MOSI (Master Out, Slave In) pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
79	PB.2	I/O	MFP0	General purpose digital I/O pin.
	UART1_RXD	I	MPF1	Data receiver input pin for UART1.
	SPI2_SS0	I/O	MPF2	General purpose digital I/O pin.
	USB1_D-	A	MPF3	USB1 differential signal D+.
	EBI_AD4	O	MPF7	EBI address/data bus bit 4.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
80	PB.3	I/O	MFP0	General purpose digital I/O pin.
	UART1_TXD	O	MPF1	Data transmitter output pin for UART1.
	SPI2_CLK	O	MPF2	SPI2 serial clock pin.
	USB1_D+	A	MPF3	USB1 differential signal D+.
	EBI_AD5	O	MPF7	EBI address/data bus bit 5.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
81	PB.4	I/O	MFP0	General purpose digital I/O pin.
	UART1_RTS	O	MPF1	Request to Send output pin for UART1.

	SPI2_MISO0	I/O	MPF2	1st SPI2 MISO (Master In, Slave Out) pin.
	UART4_RXD	I	MPF3	Data receiver input pin for UART4.
	TM0_CNT_OUT	I/O	MPF4	Timer0 event counter input/toggle output.
	EBI_AD6	O	MPF7	EBI address/data bus bit 6.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
82	PB.5	I/O	MFP0	General purpose digital I/O pin.
	UART1_CTS	I	MPF1	Clear to Send input pin for UART1.
	SPI2_MOSI0	I/O	MPF2	1st SPI2 MOSI (Master Out, Slave In) pin.
	UART4_TXD	O	MPF3	Data transmitter output pin for UART4.
	EBI_AD7	O	MPF7	EBI address/data bus bit 7.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
83	PB.6	I/O	MFP0	General purpose digital I/O pin.
	I2C2_SCL	I/O	MPF1	I2C2 clock pin.
	BRAKE01	I	MPF2	Brake input pin 1 of EPWMB.
	UART4_RTS	O	MPF3	Request to Send output pin for UART4.
	PWM1_4	I/O	MPF4	PWM1_4 output/capture input.
	EPWM1_0	I/O	MPF5	PWM1_0 output/capture input.
	EBI_AD8	O	MPF7	EBI address/data bus bit 8.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
84	PB.7	I/O	MFP0	General purpose digital I/O pin.
	I2C2_SDA	I/O	MPF1	I2C2 data input/output pin.
	BRAKE00	I	MPF2	Brake input pin 0 of EPWMB.
	UART4_CTS	I	MPF3	Clear to Send input pin for UART4.
	PWM1_5	I/O	MPF4	PWM1_5 output/capture input.
	EPWM1_1	I/O	MPF5	PWM1_1 output/capture input.
	EBI_AD9	O	MPF7	EBI address/data bus bit 9.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
85	PB.8	I/O	MFP0	General purpose digital I/O pin.
	UART5_CTS	I	MPF1	Clear to Send input pin for UART5.
	EPWM1_2	I/O	MPF5	PWM1_2 output/capture input.
	EBI_AD10	O	MPF7	EBI address/data bus bit 10.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
86	PB.9	I/O	MFP0	General purpose digital I/O pin.
	UART5_RTS	O	MPF1	Request to Send output pin for UART5.
	EPWM1_3	I/O	MPF5	PWM1_3 output/capture input.
	EBI_AD11	O	MPF7	EBI address/data bus bit 11.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
87	PB.10	I/O	MFP0	General purpose digital I/O pin.
	UART5_TXD	O	MPF1	Data transmitter output pin for UART5.
	EPWM1_4	I/O	MPF5	PWM1_4 output/capture input.
	EBI_AD12	O	MPF7	EBI address/data bus bit 12.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
88	PB.11	I/O	MFP0	General purpose digital I/O pin.
	UART5_RXD	I	MPF1	Data receiver input pin for UART5.
	EPWM1_5	I/O	MPF5	PWM1_5 output/capture input.
	EBI_AD13	O	MPF7	EBI address/data bus bit 13.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
89	PB.12	I/O	MFP0	General purpose digital I/O pin.
	UART4_RTS	O	MPF1	Request to Send output pin for UART4.
	SPI2_MISO1	I/O	MPF2	2nd SPI2 MISO (Master In, Slave Out) pin.
	CAN0_RXD	I	MPF3	CAN bus receiver0 input.

	EMAC_MII_MDC	O	MPF6	MII/RMII Management Data Clock.
	EBI_AD14	O	MPF7	EBI address/data bus bit 14.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
90	PB.13	I/O	MFP0	General purpose digital I/O pin.
	UART4_CTS	I	MPF1	Clear to Send input pin for UART4.
	SPI2_MOSI1	I/O	MPF2	2nd SPI2 MOSI (Master Out, Slave In) pin.
	CAN0_TXD	I	MPF3	CAN bus transmitter0 input.
	EMAC_MII_MDIO	I/O	MPF6	MII/RMII Management Data I/O.
	EBI_AD15	O	MPF7	EBI address/data bus bit 15.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
91	PB.14	I/O	MFP0	General purpose digital I/O pin.
	I2S1_MCLK	O	MPF1	I2S1 master clock output pin.
	SC1_RST	O	MPF2	SmartCard1 reset pin.
	BRAKE01	I	MPF4	Brake input pin 1 of EPWMB.
	EMAC_MII_MDC	O	MPF6	MII/RMII Management Data Clock.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
92	PB.15	I/O	MFP0	General purpose digital I/O pin.
	I2S1_DO	O	MPF1	I2S1 data output.
	SC1_DAT	I/O	MPF2	SmartCard1 data pin.
	BRAKE00	I	MPF4	Brake input pin 0 of EPWMB.
	EMAC_MII_MDIO	I/O	MPF6	MII/RMII Management Data I/O.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
93	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
94	VSS	P	MFP0	Ground pin for digital circuit.
95	PC.0	I/O	MFP0	General purpose digital I/O pin.

	I2S1_DI	I	MPF1	I2S1 data input.
	SC1_DAT	I/O	MPF2	SmartCard1 data pin.
	UART4_RXD	I	MPF3	Data receiver input pin for UART4.
	EMAC_REFCLK	I	MPF6	EMAC RMII mode clock input
	EBI_MCLK	O	MPF7	EBI interface clock output pin.
	INT2	I	MPF8	External interrupt2 input pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
96	PC.1	I/O	MFP0	General purpose digital I/O pin.
	I2S1_BCLK	O	MPF1	I2S1 bit clock pin.
	SC1_CLK	O	MPF2	SmartCard1 clock pin.
	UART4_TXD	O	MPF3	Data transmitter output pin for UART4.
	TM3_CNT_OUT	I/O	MPF5	Timer3 event counter input/toggle output.
	EMAC_MII_RXERR	I	MPF6	MII/RMII Receive Data error.
	EBI_AD13	O	MPF7	EBI address/data bus bit 13.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
97	PC.2	I/O	MFP0	General purpose digital I/O pin.
	I2S1_LRCK	O	MPF1	I2S1 left right channel clock.
	SC1_PWR	O	MPF2	SmartCard1 power pin.
	UART4_RTS	O	MPF3	Request to Send output pin for UART4.
	SPI0_SS0	I/O	MPF4	General purpose digital I/O pin.
	EMAC_MII_RXDV	I	MPF6	MII Receive Data Valid / RMII CRS_DV Input.
	EBI_AD12	O	MPF7	EBI address/data bus bit 12.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
98	PC.3	I/O	MFP0	General purpose digital I/O pin.
	I2S1_MCLK	O	MPF1	I2S1 master clock output pin.
	SC1_CD	I	MPF2	SmartCard1 card detect pin.

	UART4_CTS	I	MPF3	Clear to Send input pin for UART4.
	SPI0_MISO1	I/O	MPF4	2nd SPI0 MISO (Master In, Slave Out) pin.
	QEI0_Z	I	MPF5	Quadrature encoder phase Z input of QEI Unit 0.
	EMAC_MII_RXD1	I	MPF6	MII/RMII Receive Data Bus Bit 1.
	EBI_AD11	O	MPF7	EBI address/data bus bit 11.
	ECAP0_IC2	O	MPF8	Input 2 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
99	PC.4	I/O	MFP0	General purpose digital I/O pin.
	I2S1_DO	O	MPF1	I2S1 data output.
	SC1_RST	O	MPF2	SmartCard1 reset pin.
	SPI0_MOSI1	I/O	MPF4	2nd SPI0 MOSI (Master Out, Slave In) pin.
	QEI0_B	I	MPF5	Quadrature encoder phase B input of QEI Unit 0.
	EMAC_MII_RXD0	I	MPF6	MII/RMII Receive Data Bus Bit 0.
	EBI_AD10	O	MPF7	EBI address/data bus bit 10.
	ECAP0_IC1	O	MPF8	Input 1 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
100	PC.5	I/O	MFP0	General purpose digital I/O pin.
	CLKO	O	MFP1	Clock Output Pin.
	QEI0_A	I	MPF5	Quadrature encoder phase A input of QEI Unit 0.
	EMAC_MII_RXCLK	I	MPF6	MII Receive Clock Input.
	EBI_MCLK	O	MPF7	EBI interface clock output pin.
	ECAP0_IC0	O	MPF8	Input 0 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
101	PC.6	I/O	MFP0	General purpose digital I/O pin.
	TM2_EXT	I	MPF1	Timer2 external counter input
	SPI0_MISO0	I/O	MPF4	1st SPI0 MISO (Master In, Slave Out) pin.

	TM2_CNT_OUT	I/O	MPF5	Timer2 event counter input/toggle output.
	EMAC_MII_TXD0	O	MPF6	MII/RMII Transmit Data Bus bit 0.
	EBI_AD9	O	MPF7	EBI address/data bus bit 9.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
102	PC.7	I/O	MFP0	General purpose digital I/O pin.
	TM1_EXT	I	MPF1	Timer1 external counter input
	SPI0_MOSI0	I/O	MPF4	1st SPI0 MOSI (Master Out, Slave In) pin.
	EMAC_MII_TXD1	O	MPF6	MII/RMII Transmit Data Bus bit 1.
	EBI_AD8	O	MPF7	EBI address/data bus bit 8.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
103	PC.8	I/O	MFP0	General purpose digital I/O pin.
	TM0_EXT	I	MPF1	Timer0 external counter input
	SPI0_CLK	O	MPF4	SPI0 serial clock pin.
	EMAC_MII_TXEN	O	MPF6	MII/RMII Transmit Enable.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
104	PF.2	I/O	MFP0	General purpose digital I/O pin.
	SPI3_SS0	I/O	MPF1	General purpose digital I/O pin.
	SD0_DAT3	I/O	MPF4	SD mode #0 data line bit 3.
	EMAC_MII_RXD3	I	MPF6	MII Receive Data bus bit 3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
105	PF.3	I/O	MFP0	General purpose digital I/O pin.
	SPI3_CLK	O	MPF1	SPI3 serial clock pin.
	SD0_DAT2	I/O	MPF4	SD mode #0 data line bit 2.
	EMAC_MII_RXD2	I	MPF6	MII Receive Data Bus Bit 2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

106	PF.4	I/O	MFP0	General purpose digital I/O pin.
	SPI3_MISO0	I/O	MPF1	1st SPI3 MISO (Master In, Slave Out) pin.
	SD0_DAT1	I/O	MPF4	SD mode #0 data line bit 1.
	EMAC_MII_COL0	I	MPF6	MII Collision Detect Input Pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
107	PF.5	I/O	MFP0	General purpose digital I/O pin.
	SPI3_MOSI0	I/O	MPF1	1st SPI3 MOSI (Master Out, Slave In) pin.
	SD0_DAT0	I/O	MPF4	SD mode #0 data line bit 0.
	EMAC_MII_CRS	I	MPF6	MII Carrier Sense Input Pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
108	PF.6	I/O	MFP0	General purpose digital I/O pin.
	UART2_RXD	I	MPF1	Data receiver input pin for UART2.
	SD0_CDn	I	MPF4	SD mode #0 – card detect
	EMAC_MII_TXCLK	I	MPF6	MII Transmit Output Clock Pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
109	PF.7	I/O	MFP0	General purpose digital I/O pin.
	UART2_TXD	O	MPF1	Data transmitter output pin for UART2.
	SD0_CMD	I/O	MPF4	SD mode #0 – command/response
	EMAC_MII_TXD3	O	MPF6	MII Transmit Data Bus bit 3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
110	PF.8	I/O	MFP0	General purpose digital I/O pin.
	UART2_RTS	O	MPF1	Request to Send output pin for UART2.
	SD0_CLK	O	MPF4	SD mode #0 – clock.
	EMAC_MII_TXD2	O	MPF6	MII Transmit Data Bus bit 2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

111	LDO_CAP	P	MFP0	LDO output pin.
112	VSS	P	MFP0	Ground pin for digital circuit.
113	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
114	PE.0	I/O	MFP0	General purpose digital I/O pin.
	ADC0_0	A	MPF1	ADC0 analog input.
	INT4	I	MPF8	External interrupt4 input pin.
115	PE.1	I/O	MFP0	General purpose digital I/O pin.
	ADC0_1	A	MPF1	ADC0 analog input.
	TM2_CNT_OUT	I/O	MPF3	Timer2 event counter input/toggle output.
116	PE.2	I/O	MFP0	General purpose digital I/O pin.
	ADC0_2	A	MPF1	ADC0 analog input.
	ACMP0_O	O	MPF2	Analog ccomparator0 output .
	SPI0_MISO0	I/O	MPF3	1st SPI0 MISO (Master In, Slave Out) pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
117	PE.3	I/O	MFP0	General purpose digital I/O pin.
	ADC0_3	A	MPF1	ADC0 analog input.
	ACMP0_P3	A	MPF2	Analog comparator0 positive input pin.
	SPI0_MOSI0	I/O	MPF3	1st SPI0 MOSI (Master Out, Slave In) pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
118	PE.4	I/O	MFP0	General purpose digital I/O pin.
	ADC0_4	A	MPF1	ADC0 analog input.
	ACMP0_P2	A	MPF2	Analog comparator0 positive input pin.
	SPI0_SS0	I/O	MPF3	General purpose digital I/O pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
119	PE.5	I/O	MFP0	General purpose digital I/O pin.

	ADC0_5	A	MPF1	ADC0 analog input.
	ACMP0_P1	A	MPF2	Analog comparator0 positive input pin.
	SPI0_CLK	O	MPF3	SPI0 serial clock pin.
	SD0_CDn	I	MPF4	SD mode #0 – card detect
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
120	PE.6	I/O	MFP0	General purpose digital I/O pin.
	ADC0_6	A	MPF1	ADC0 analog input.
	ACMP0_P0	A	MPF2	Analog comparator0 positive input pin.
	SPI0_MISO0	I/O	MPF3	1st SPI0 MISO (Master In, Slave Out) pin.
	SD0_CMD	I/O	MPF4	SD mode #0 – command/response
	EBI_nWR	O	MPF7	EBI write enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
121	PE.7	I/O	MFP0	General purpose digital I/O pin.
	ADC0_7	A	MPF1	ADC0 analog input.
	ACMP0_N	A	MPF2	Analog comparator0 negative input pin.
	SPI0_MOSI0	I/O	MPF3	1st SPI0 MOSI (Master Out, Slave In) pin.
	SD0_CLK	O	MPF4	SD mode #0– clock.
	EBI_nRD	O	MPF7	EBI read enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
122	AVSS	P	MFP0	Ground pin for digital circuit.
123	Vref	A	MFP0	Voltage reference input for ADC.
124	AVDD	P	MFP0	Power supply for internal analog circuit.
125	PE.8	I/O	MFP0	General purpose digital I/O pin.
	ADC1_0	A	MPF1	ADC1 analog input.
	ADC0_8	A	MPF1	ADC0 analog input.
	ACMP1_N	A	MPF2	Analog comparator1 negative input pin.

	TM1_CNT_OUT	I/O	MPF3	Timer1 event counter input/toggle output.
	SD0_DAT3	I/O	MPF4	SD mode #0 data line bit 3.
	EBI_ALE	O	MPF7	EBI address latch enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
126	PE.9	I/O	MFP0	General purpose digital I/O pin.
	ADC1_1	A	MPF1	ADC1 analog input.
	ADC0_9	A	MPF1	ADC0 analog input.
	ACMP1_P0	A	MPF2	Analog comparator1 positive input pin.
	SD0_DAT2	I/O	MPF4	SD mode #0 data line bit 2.
	EBI_nWRH	O	MPF7	EBI write enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
127	PE.10	I/O	MFP0	General purpose digital I/O pin.
	ADC1_2	A	MPF1	ADC1 analog input.
	ADC0_10	A	MPF1	ADC0 analog input.
	ACMP1_P1	A	MPF2	Analog comparator1 positive input pin.
	SPI0_MISO1	I/O	MPF3	2nd SPI0 MISO (Master In, Slave Out) pin.
	SD0_DAT1	I/O	MPF4	SD mode #0 data line bit 1.
	EBI_nWRL	O	MPF7	EBI write enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
128	PE.11	I/O	MFP0	General purpose digital I/O pin.
	ADC1_3	A	MPF1	ADC1 analog input.
	ADC0_11	A	MPF1	ADC0 analog input.
	ACMP1_P2	A	MPF2	Analog comparator1 positive input pin.
	SPI0_MOSI1	I/O	MPF3	2nd SPI0 MOSI (Master Out, Slave In) pin.
	SD0_DAT0	I/O	MPF4	SD mode #0 data line bit 0.
	ACMP2_P3	A	MPF5	Analog comparator2 positive input pin.

	EBI_nCS0	O	MPF7	EBI chip select 0 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

Note: Pin Type I = Digital Input, O = Digital Output; A = Analog Pin; P = Power Pin

4.3.7 NuMicro™ NUC472 Package LQFP 144-pin Description

MFP = Multi-function pin.

Pin No.	Pin Name	Type	MFP*	Description
1	PE.12	I/O	MFP0	General purpose digital I/O pin.
	ADC1_4	A	MPF1	ADC1 analog input.
	ACMP1_P3	A	MPF2	Analog comparator1 positive input pin.
	ACMP2_P2	A	MPF3	Analog comparator2 positive input pin.
	EBI_nCS1	O	MPF7	EBI chip select 1 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
2	PE.13	I/O	MFP0	General purpose digital I/O pin.
	ADC1_5	A	MPF1	ADC1 analog input.
	ACMP2_P1	A	MPF3	Analog comparator2 positive input pin.
	EBI_nCS2	O	MPF7	EBI chip select 2 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
3	PE.14	I/O	MFP0	General purpose digital I/O pin.
	ADC1_6	A	MPF1	ADC1 analog input.
	ACMP2_P0	A	MPF3	Analog comparator2 positive input pin.
	EBI_nCS3	O	MPF7	EBI chip select 3 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
4	PE.15	I/O	MFP0	General purpose digital I/O pin.
	ADC1_7	A	MPF1	ADC1 analog input.
	ACMP2_N	A	MPF3	Analog comparator2 negative input pin.
5	PF.9	I/O	MFP0	General purpose digital I/O pin.
	OPA0_IN+	I/O	MPF1	General purpose digital I/O pin.
	PWM0_0	I/O	MPF4	PWM0_0 output/capture input.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

6	PF.10	I/O	MFP0	General purpose digital I/O pin.
	OPA0_IN-	I/O	MPF1	General purpose digital I/O pin.
	PWM0_1	I/O	MPF4	PWM0_1 output/capture input.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
7	PF.11	I/O	MFP0	General purpose digital I/O pin.
	OPA0_O	O	MPF1	Operational amplifier output pin
	UART1_RTS	O	MPF2	Request to Send output pin for UART1.
8	PF.12	I/O	MFP0	General purpose digital I/O pin.
	OPA1_IN+	I/O	MPF1	General purpose digital I/O pin.
	UART1_CTS	I	MPF2	Clear to Send input pin for UART1.
9	PF.13	I/O	MFP0	General purpose digital I/O pin.
	OPA1_IN-	I/O	MPF1	General purpose digital I/O pin.
	UART1_TXD	O	MPF2	Data transmitter output pin for UART1.
10	PF.14	I/O	MFP0	General purpose digital I/O pin.
	OPA1_O	O	MPF1	Operational amplifier output pin
	UART1_RXD	I	MPF2	Data receiver input pin for UART1.
11	VSS	P	MFP0	Ground pin for digital circuit.
12	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
13	PF.15	I/O	MFP0	General purpose digital I/O pin.
	UART0_RTS	O	MPF1	Request to Send output pin for UART0.
14	PG.0	I/O	MFP0	General purpose digital I/O pin.
	UART0_CTS	I	MPF1	Clear to Send input pin for UART0.
	INT6	I	MPF8	External interrupt6 input pin.
15	PG.1	I/O	MFP0	General purpose digital I/O pin.
	UART0_RXD	I	MPF1	Data receiver input pin for UART0.
16	PG.2	I/O	MFP0	General purpose digital I/O pin.

	UART0_TXD	O	MPF1	Data transmitter output pin for UART0.
17	PC.12	I/O	MFP0	General purpose digital I/O pin.
	SPI1_SS0	I/O	MPF1	1st SPI1 slave select pin.
	SC4_CD	I	MPF2	SmartCard4 card detect pin.
	SD1_CDn	I	MPF4	SD mode #1 – card detect
	CAP_DATA7	I	MPF5	Image data input bus bit 7.
	EBI_A0	O	MPF7	EBI address bus bit0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
18	PC.13	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MOSI1	I/O	MPF1	2nd SPI1 MOSI (Master Out, Slave In) pin.
	SC4_RST	O	MPF2	SmartCard4 reset pin.
	SD1_CMD	I/O	MPF4	SD mode #1 – command/response
	CAP_DATA6	I	MPF5	Image data input bus bit 7.
	EBI_A1	O	MPF7	EBI address bus bit1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
19	PC.14	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MISO1	I/O	MPF1	2nd SPI1 MISO (Master In, Slave Out) pin.
	SC4_PWR	O	MPF2	SmartCard4 power pin.
	TM3_EXT	I	MPF3	Timer3 external counter input
	SD1_CLK	O	MPF4	SD mode #1– clock.
	CAP_DATA5	I	MPF5	Image data input bus bit 7.
	EBI_A2	O	MPF7	EBI address bus bit2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
20	PC.15	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MOSI0	I/O	MPF1	1st SPI1 MOSI (Master Out, Slave In) pin.
	SC4_DAT	I/O	MPF2	SmartCard4 data pin.

	SD1_DAT3	I/O	MPF4	SD mode #1 data line bit 3;
	CAP_DATA4	I	MPF5	Image data input bus bit 7.
	EBI_A3	O	MPF7	EBI address bus bit3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
21	PD.0	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MISO0	I/O	MPF1	1st SPI1 MISO (Master In, Slave Out) pin.
	SC4_CLK	O	MPF2	SmartCard4 clock pin.
	SD1_DAT2	I/O	MPF4	SD mode #1 data line bit 2;
	CAP_DATA3	I	MPF5	Image data input bus bit 7.
	EBI_A4	O	MPF7	EBI address bus bit4.
	INT3	I	MPF8	External interrupt3 input pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
22	PD.1	I/O	MFP0	General purpose digital I/O pin.
	SPI1_CLK	O	MPF1	SPI1 serial clock pin.
	TM0_CNT_OUT	I/O	MPF3	Timer0 event counter input/toggle output.
	SD1_DAT1	I/O	MPF4	SD mode #1 data line bit 1;
	CAP_DATA2	I	MPF5	Image data input bus bit 7.
	EBI_A5	O	MPF7	EBI address bus bit5.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
23	PD.2	I/O	MFP0	General purpose digital I/O pin.
	STADC	A	MPF1	ADC analog input.
	I2C3_SCL	I/O	MPF2	I2C3 clock pin.
	SD1_DAT0	I/O	MPF4	SD mode #1 data line bit 0.
	CAP_DATA1	I	MPF5	Image data input bus bit 7.
	EBI_A6	O	MPF7	EBI address bus bit6.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
24	PD.3	I/O	MFP0	General purpose digital I/O pin.
	SC5_CLK	O	MPF1	SmartCard5 clock pin.
	I2C3_SDA	I/O	MPF2	I2C3 data input/output pin.
	ACMP2_O	O	MPF3	Analog ccomparator2 output .
	SD0_CDn	I	MPF4	SD mode #0 – card detect
	CAP_DATA0	I	MPF5	Image data input bus bit 7.
	EBI_A7	O	MPF7	EBI address bus bit7.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
25	PD.4	I/O	MFP0	General purpose digital I/O pin.
	SC5_CD	I	MPF1	SmartCard5 card detect pin.
	UART3_RXD	I	MPF2	Data receiver input pin for UART3.
	ACMP1_O	O	MPF3	Analog ccomparator1 output.
	CAP_SCLK	O	MPF5	Image capture interface sensor clock pin.
	EBI_A8	O	MPF7	EBI address bus bit8.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
26	PD.5	I/O	MFP0	General purpose digital I/O pin.
	SC5_RST	O	MPF1	SmartCard5 reset pin.
	UART3_TXD	O	MPF2	Data transmitter output pin for UART3.
	CAP_VSYNC	I	MPF5	Image capture interface VSYNC input pin.
	EBI_A9	O	MPF7	EBI address bus bit9.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
27	PD.6	I/O	MFP0	General purpose digital I/O pin.
	SC5_PWR	O	MPF1	SmartCard5 power pin.
	UART3_RTS	O	MPF2	Request to Send output pin for UART3.

	SD0_CMD	I/O	MPF4	SD mode #0 – command/response
	CAP_HSYNC	I	MPF5	Image capture interface HSYNC input pin.
	EBI_A10	O	MPF7	EBI address bus bit10.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
28	PD.7	I/O	MFP0	General purpose digital I/O pin.
	SC5_DAT	I/O	MPF1	SmartCard5 data pin.
	UART3_CTS	I	MPF2	Clear to Send input pin for UART3.
	SD0_CLK	O	MPF4	SD mode #0 – clock.
	CAP_PIXCLK	I	MPF5	Image capture interface pix clock input pin.
	EBI_A11	O	MPF7	EBI address bus bit11.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
29	PG.13	I/O	MFP0	General purpose digital I/O pin.
	XT1_IN	I	MPF1	External 4~24 MHz (high-speed) crystal input pin.
30	PG.12	I/O	MFP0	General purpose digital I/O pin.
	XT1_OUT	O	MPF1	External 4~24 MHz (high-speed) crystal output pin.
31	nRESET	I	MFP0	External reset input: active LOW, with an internal pull-up. Set this pin low reset to initial state.
32	LDO_CAP	P	MFP0	LDO output pin.
33	VSS	P	MFP0	Ground pin for digital circuit.
34	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
35	PG.10	I/O	MFP0	General purpose digital I/O pin.
	ICE_CLK	I	MPF1	Serial wired debugger clock pin
36	PG.11	I/O	MFP0	General purpose digital I/O pin.
	ICE_DAT	I/O	MPF1	Serial wired debugger data pin
37	PG.15	I/O	MFP0	General purpose digital I/O pin.

	X32K_IN	I	MPF1	External 32.768 kHz (low-speed) crystal input pin.
	I2C1_SCL	I/O	MPF3	I2C1 clock pin.
38	PG.14	I/O	MFP0	General purpose digital I/O pin.
	X32K_OUT	O	MPF1	External 32.768 kHz (low-speed) crystal output pin.
	I2C1_SDA	I/O	MPF3	I2C1 data input/output pin.
39	VBAT	P	MFP0	Battery power input pin.
40	PA.0	I/O	MFP0	General purpose digital I/O pin.
	TAMPER0	I/O	MPF1	Tamper detect pin 0.
	SC0_CD	I	MPF2	SmartCard0 card detect pin.
	CAN1_RXD	I	MPF3	CAN bus receiver1 input.
	INT0	I	MPF8	External interrupt0 input pin.
41	PA.1	I/O	MFP0	General purpose digital I/O pin.
	TAMPER1	I/O	MPF1	Tamper detect pin 1.
	SC5_CD	I	MPF2	SmartCard5 card detect pin.
	CAN1_TXD	I	MPF3	CAN bus transmitter1 input.
	EBI_A22	O	MPF7	EBI address bus bit22.
42	PD.8	I/O	MFP0	General purpose digital I/O pin.
	SPI3_MISO1	I/O	MPF1	2nd SPI3 MISO (Master In, Slave Out) pin.
	I2C0_SCL	I/O	MPF2	I2C0 clock pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
43	PD.9	I/O	MFP0	General purpose digital I/O pin.
	SPI3_MOSI1	I/O	MPF1	2nd SPI3 MOSI (Master Out, Slave In) pin.
	I2C0_SDA	I/O	MPF2	I2C0 data input/output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
44	PA.2	I/O	MFP0	General purpose digital I/O pin.
	SC2_DAT	I/O	MPF1	SmartCard2 data pin.

	SPI3_MISO0	I/O	MPF2	1st SPI3 MISO (Master In, Slave Out) pin.
	I2S0_MCLK	O	MPF3	I2S0 master clock output pin.
	BRAKE11	I	MPF4	Brake input pin 1 of EPWMA.
	CAP_SFIELD	I	MPF5	Video input interface SFIELD input pin.
	EBI_A12	O	MPF7	EBI address bus bit12.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
45	PA.3	I/O	MFP0	General purpose digital I/O pin.
	SC2_CLK	O	MPF1	SmartCard2 clock pin.
	SPI3_MOSI0	I/O	MPF2	1st SPI3 MOSI (Master Out, Slave In) pin.
	I2S0_DO	O	MPF3	I2S0 data output.
	BRAKE10	I	MPF4	Brake input pin 0 of EPWMA.
	EBI_A13	O	MPF7	EBI address bus bit13.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
46	PA.4	I/O	MFP0	General purpose digital I/O pin.
	SC2_PWR	O	MPF1	SmartCard2 power pin.
	SPI3_CLK	O	MPF2	SPI3 serial clock pin.
	I2S0_DI	I	MPF3	I2S0 data input.
	QEI1_Z	I	MPF5	Quadrature encoder phase Z input of QEI Unit 1.
	EBI_A14	O	MPF7	EBI address bus bit14.
	ECAP1_IC2	I	MPF8	Input 2 of enhanced capture unit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
47	PA.5	I/O	MFP0	General purpose digital I/O pin.
	SC2_RST	O	MPF1	SmartCard2 reset pin.
	SPI3_SS0	I/O	MPF2	General purpose digital I/O pin.
	I2S0_BCLK	O	MPF3	I2S0 bit clock pin.
	PWM0_0	I/O	MPF4	PWM0_0 output/capture input.

	QEI1_B	I	MPF5	Quadrature encoder phase B input of QEI Unit 1.
	EBI_A15	O	MPF7	EBI address bus bit15.
	ECAP1_IC1	I	MPF8	Input 1 of enhanced capture unit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
48	PA.6	I/O	MFP0	General purpose digital I/O pin.
	SC2_CD	I	MPF1	SmartCard2 card detect pin.
	I2S0_LRCK	O	MPF3	I2S0 left right channel clock.
	PWM0_1	I/O	MPF4	PWM0_1 output/capture input.
	QEI1_A	I	MPF5	Quadrature encoder phase A input of QEI Unit 1.
	CAN1_TXD	I	MPF6	CAN bus transmitter1 input.
	EBI_A16	O	MPF7	EBI address bus bit16.
	ECAP1_IC0	I	MPF8	Input 0 of enhanced capture unit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
49	PG.3	I/O	MFP0	General purpose digital I/O pin.
	PS2_CLK	O	MPF1	PS2 clock pin.
	I2S1_DO	O	MPF2	I2S1 data output.
	SC1_RST	O	MPF3	SmartCard1 reset pin.
50	PG.4	I/O	MFP0	General purpose digital I/O pin.
	PS2_DAT	I/O	MPF1	PS2 data pin.
	I2S1_DI	I	MPF2	I2S1 data input.
	SC1_PWR	O	MPF3	SmartCard1 power pin.
51	PG.5	I/O	MFP0	General purpose digital I/O pin.
	I2S1_BCLK	O	MPF2	I2S1 bit clock pin.
	SC1_DAT	I/O	MPF3	SmartCard1 data pin.
52	PG.6	I/O	MFP0	General purpose digital I/O pin.
	I2S1_LRCK	O	MPF2	I2S1 left right channel clock.

	SC1_CLK	O	MPF3	SmartCard1 clock pin.
53	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
54	VSS	P	MFP0	Ground pin for digital circuit.
55	PA.7	I/O	MFP0	General purpose digital I/O pin.
	SC0_CLK	O	MPF2	SmartCard0 clock pin.
	SPI3_SS0	I/O	MPF3	General purpose digital I/O pin.
	PWM1_3	I/O	MPF4	PWM1_3 output/capture input.
	EPWM0_5	I/O	MPF5	PWM0_5 output/capture input.
	EBI_A17	O	MPF7	EBI address bus bit17.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
56	PA.8	I/O	MFP0	General purpose digital I/O pin.
	SC0_RST	O	MPF2	SmartCard0 reset pin.
	SPI3_CLK	O	MPF3	SPI3 serial clock pin.
	PWM1_2	I/O	MPF4	PWM1_2 output/capture input.
	EPWM0_4	I/O	MPF5	PWM0_4 output/capture input.
	EBI_A18	O	MPF7	EBI address bus bit18.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
57	PA.9	I/O	MFP0	General purpose digital I/O pin.
	SC0_PWR	O	MPF2	SmartCard0 power pin.
	SPI3_MISO0	I/O	MPF3	1st SPI3 MISO (Master In, Slave Out) pin.
	PWM1_1	I/O	MPF4	PWM1_1 output/capture input.
	EPWM0_3	I/O	MPF5	PWM0_3 output/capture input.
	EBI_A19	O	MPF7	EBI address bus bit19.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
58	PA.10	I/O	MFP0	General purpose digital I/O pin.

	SC0_DAT	I/O	MPF2	SmartCard0 data pin.
	SPI3_MOSI0	I/O	MPF3	1st SPI3 MOSI (Master Out, Slave In) pin.
	PWM1_0	I/O	MPF4	PWM1_0 output/capture input.
	EPWM0_2	I/O	MPF5	PWM0_2 output/capture input.
	EBI_A20	O	MPF7	EBI address bus bit20.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
59	PA.11	I/O	MFP0	General purpose digital I/O pin.
	UART0_RTS	O	MPF1	Request to Send output pin for UART0.
	SPI3_MISO1	I/O	MPF3	2nd SPI3 MISO (Master In, Slave Out) pin.
	PWM0_5	I/O	MPF4	PWM0_5 output/capture input.
	EPWM0_1	I/O	MPF5	PWM0_1 output/capture input.
	EBI_AD0	O	MPF7	EBI address/data bus bit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
60	PA.12	I/O	MFP0	General purpose digital I/O pin.
	UART0_CTS	I	MPF1	Clear to Send input pin for UART0.
	SPI3_MOSI1	I/O	MPF3	2nd SPI3 MOSI (Master Out, Slave In) pin.
	PWM0_4	I/O	MPF4	PWM0_4 output/capture input.
	EPWM0_0	I/O	MPF5	PWM0_0 output/capture input.
	EBI_AD1	O	MPF7	EBI address/data bus bit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
61	PA.13	I/O	MFP0	General purpose digital I/O pin.
	UART0_RXD	I	MPF1	Data receiver input pin for UART0.
	SC3_DAT	I/O	MPF3	SmartCard3 data pin.
	PWM1_4	I/O	MPF4	PWM1_4 output/capture input.
	EBI_AD2	O	MPF7	EBI address/data bus bit 2.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
62	PA.14	I/O	MFP0	General purpose digital I/O pin.
	UART0_TXD	O	MPF1	Data transmitter output pin for UART0.
	SC3_CLK	O	MPF3	SmartCard3 clock pin.
	PWM1_5	I/O	MPF4	PWM1_5 output/capture input.
	EBI_AD3	O	MPF7	EBI address/data bus bit 3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
63	PD.10	I/O	MFP0	General purpose digital I/O pin.
	SC3_DAT	I/O	MPF1	SmartCard3 data pin.
	I2C4_SCL	I/O	MPF2	I2C4 clock pin.
64	PD.11	I/O	MFP0	General purpose digital I/O pin.
	SC3_RST	O	MPF1	SmartCard3 reset pin.
	TM3_CNT_OUT	I/O	MPF3	Timer3 event counter input/toggle output.
65	PD.12	I/O	MFP0	General purpose digital I/O pin.
	SC3_CLK	O	MPF1	SmartCard3 clock pin.
	I2C4_SDA	I/O	MPF2	I2C4 data input/output pin.
66	PA.15	I/O	MFP0	General purpose digital I/O pin.
	SC3_PWR	O	MPF1	SmartCard3 power pin.
	UART2_RTS	O	MPF2	Request to Send output pin for UART2.
	I2C0_SCL	I/O	MPF4	I2C0 clock pin.
	EBI_A21	O	MPF7	EBI address bus bit21.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
67	PC.9	I/O	MFP0	General purpose digital I/O pin.
	STADC	A	MPF1	ADC analog input.
	UART2_CTS	I	MPF2	Clear to Send input pin for UART2.
	SC3_RST	O	MPF3	SmartCard3 reset pin.

	I2C0_SDA	I/O	MPF4	I2C0 data input/output pin.
	CAP_DATA1	I	MPF5	Image data input bus bit 7.
	I2C3_SCL	I/O	MPF6	I2C3 clock pin.
	EBI_A22	O	MPF7	EBI address bus bit22.
	SD1_DAT0	I/O	MPF8	SD mode #1 data line bit 0.
	EBI_A6	O	MPF9	EBI address bus bit6.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
68	PC.10	I/O	MFP0	General purpose digital I/O pin.
	SC3_CD	I	MPF1	SmartCard3 card detect pin.
	UART2_RXD	I	MPF2	Data receiver input pin for UART2.
	PWM0_2	I/O	MPF4	PWM0_2 output/capture input.
	EBI_A23	O	MPF6	EBI address bus bit23.
	EBI_AD2	O	MPF7	EBI address/data bus bit 2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
69	PC.11	I/O	MFP0	General purpose digital I/O pin.
	UART2_TXD	O	MPF2	Data transmitter output pin for UART2.
	PWM0_3	I/O	MPF4	PWM0_3 output/capture input.
	EBI_A24	O	MPF6	EBI address bus bit24.
	EBI_AD3	O	MPF7	EBI address/data bus bit 3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
70	LDO_CAP	P	MFP0	LDO output pin.
71	VSS	P	MFP0	Ground pin for digital circuit.
72	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
73	PD.13	I/O	MFP0	General purpose digital I/O pin.
	SPI1_SS0	I/O	MPF1	1st SPI1 slave select pin.

	UART5_CTS	I	MPF2	Clear to Send input pin for UART5.
	ECAP0_IC2	O	MPF3	Input 2 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
74	PD.14	I/O	MFP0	General purpose digital I/O pin.
	SPI1_CLK	O	MPF1	SPI1 serial clock pin.
	UART5_RTS	O	MPF2	Request to Send output pin for UART5.
	ECAP0_IC1	O	MPF3	Input 1 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
75	PD.15	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MISO0	I/O	MPF1	1st SPI1 MISO (Master In, Slave Out) pin.
	UART5_TXD	O	MPF2	Data transmitter output pin for UART5.
	ECAP0_IC0	O	MPF3	Input 0 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
76	PF.0	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MOSI0	I/O	MPF1	1st SPI1 MOSI (Master Out, Slave In) pin.
	UART5_RXD	I	MPF2	Data receiver input pin for UART5.
	INT5	I	MPF8	External interrupt5 input pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
77	VRES	A	MFP0	USB PHY VRES ground input pin. Add an 8.2K ohm resistor to VSSA.
78	VBUS	A	MFP0	USB PHY VBUS power input pin.
79	USB_VDD33_CAP	P	MFP0	Internal power regulator output 3.3V decoupling pin.
80	VSSA	P	MFP0	Ground pin for digital circuit. Add a Feritte Bead to digital ground VSS.
81	USB0_D-	A	MFP0	USB0 differential signal D+.
82	USB0_D+	A	MFP0	USB0 differential signal D+.

83	USB0_OTG_ID	I	MFP0	USB0OTG ID pin.
84	PB.0	I/O	MFP0	General purpose digital I/O pin.
	USB0_VBUS_ST	I	MPF1	USB0 external VBUS regulator status
	I2C4_SCL	I/O	MPF2	I2C4 clock pin.
	INT1	I	MPF8	External interrupt1 input pin.
85	PB.1	I/O	MFP0	General purpose digital I/O pin.
	USB0_VBUS_EN	O	MPF1	USB0 external VBUS regulator enabled
	I2C4_SDA	I/O	MPF2	I2C4 data input/output pin.
	TM1_CNT_OUT	I/O	MPF3	Timer1 event counter input/toggle output.
86	PG.7	I/O	MFP0	General purpose digital I/O pin.
	SPI2_MISO0	I/O	MPF1	1st SPI2 MISO (Master In, Slave Out) pin.
	I2S1_MCLK	O	MPF2	I2S1 master clock output pin.
	SC1_CD	I	MPF3	SmartCard1 card detect pin.
	SC3_RST	O	MPF4	SmartCard3 reset pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
87	PG.8	I/O	MFP0	General purpose digital I/O pin.
	SPI2_MOSI0	I/O	MPF1	1st SPI2 MOSI (Master Out, Slave In) pin.
	I2S1_DO	O	MPF2	I2S1 data output.
	UART4_RTS	O	MPF3	Request to Send output pin for UART4.
	SC3_DAT	I/O	MPF4	SmartCard3 data pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
88	PG.9	I/O	MFP0	General purpose digital I/O pin.
	SPI2_CLK	O	MPF1	SPI2 serial clock pin.
	I2S1_DI	I	MPF2	I2S1 data input.
	UART4_CTS	I	MPF3	Clear to Send input pin for UART4.
	SC3_CLK	O	MPF4	SmartCard3 clock pin.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
89	PB.2	I/O	MFP0	General purpose digital I/O pin.
	UART1_RXD	I	MPF1	Data receiver input pin for UART1.
	SPI2_SS0	I/O	MPF2	General purpose digital I/O pin.
	USB1_D-	A	MPF3	USB1 differential signal D+.
	EBI_AD4	O	MPF7	EBI address/data bus bit 4.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
90	PB.3	I/O	MFP0	General purpose digital I/O pin.
	UART1_TXD	O	MPF1	Data transmitter output pin for UART1.
	SPI2_CLK	O	MPF2	SPI2 serial clock pin.
	USB1_D+	A	MPF3	USB1 differential signal D+.
	EBI_AD5	O	MPF7	EBI address/data bus bit 5.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
91	PB.4	I/O	MFP0	General purpose digital I/O pin.
	UART1_RTS	O	MPF1	Request to Send output pin for UART1.
	SPI2_MISO0	I/O	MPF2	1st SPI2 MISO (Master In, Slave Out) pin.
	UART4_RXD	I	MPF3	Data receiver input pin for UART4.
	TM0_CNT_OUT	I/O	MPF4	Timer0 event counter input/toggle output.
	EBI_AD6	O	MPF7	EBI address/data bus bit 6.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
92	PB.5	I/O	MFP0	General purpose digital I/O pin.
	UART1_CTS	I	MPF1	Clear to Send input pin for UART1.
	SPI2_MOSI0	I/O	MPF2	1st SPI2 MOSI (Master Out, Slave In) pin.
	UART4_TXD	O	MPF3	Data transmitter output pin for UART4.
	EBI_AD7	O	MPF7	EBI address/data bus bit 7.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
93	PB.6	I/O	MFP0	General purpose digital I/O pin.
	I2C2_SCL	I/O	MPF1	I2C2 clock pin.
	BRAKE01	I	MPF2	Brake input pin 1 of EPWMB.
	UART4_RTS	O	MPF3	Request to Send output pin for UART4.
	PWM1_4	I/O	MPF4	PWM1_4 output/capture input.
	EPWM1_0	I/O	MPF5	PWM1_0 output/capture input.
	EBI_AD8	O	MPF7	EBI address/data bus bit 8.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
94	PB.7	I/O	MFP0	General purpose digital I/O pin.
	I2C2_SDA	I/O	MPF1	I2C2 data input/output pin.
	BRAKE00	I	MPF2	Brake input pin 0 of EPWMB.
	UART4_CTS	I	MPF3	Clear to Send input pin for UART4.
	PWM1_5	I/O	MPF4	PWM1_5 output/capture input.
	EPWM1_1	I/O	MPF5	PWM1_1 output/capture input.
	EBI_AD9	O	MPF7	EBI address/data bus bit 9.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
95	PB.8	I/O	MFP0	General purpose digital I/O pin.
	UART5_CTS	I	MPF1	Clear to Send input pin for UART5.
	EPWM1_2	I/O	MPF5	PWM1_2 output/capture input.
	EBI_AD10	O	MPF7	EBI address/data bus bit 10.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
96	PB.9	I/O	MFP0	General purpose digital I/O pin.
	UART5_RTS	O	MPF1	Request to Send output pin for UART5.
	EPWM1_3	I/O	MPF5	PWM1_3 output/capture input.

	EBI_AD11	O	MPF7	EBI address/data bus bit 11.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
97	PB.10	I/O	MFP0	General purpose digital I/O pin.
	UART5_TXD	O	MPF1	Data transmitter output pin for UART5.
	EPWM1_4	I/O	MPF5	PWM1_4 output/capture input.
	EBI_AD12	O	MPF7	EBI address/data bus bit 12.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
98	PB.11	I/O	MFP0	General purpose digital I/O pin.
	UART5_RXD	I	MPF1	Data receiver input pin for UART5.
	EPWM1_5	I/O	MPF5	PWM1_5 output/capture input.
	EBI_AD13	O	MPF7	EBI address/data bus bit 13.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
99	PH.0	I/O	MFP0	General purpose digital I/O pin.
	I2C1_SCL	I/O	MPF1	I2C1 clock pin.
	UART4_RXD	I	MPF2	Data receiver input pin for UART4.
	CAN1_RXD	I	MPF3	CAN bus receiver1 input.
	INT7	I	MPF8	External interrupt7 input pin.
100	PH.1	I/O	MFP0	General purpose digital I/O pin.
	UART4_TXD	O	MPF1	Data transmitter output pin for UART4.
	I2C1_SDA	I/O	MPF2	I2C1 data input/output pin.
	CAN1_TXD	I	MPF3	CAN bus transmitter1 input.
101	PB.12	I/O	MFP0	General purpose digital I/O pin.
	UART4_RTS	O	MPF1	Request to Send output pin for UART4.
	SPI2_MISO1	I/O	MPF2	2nd SPI2 MISO (Master In, Slave Out) pin.
	CAN0_RXD	I	MPF3	CAN bus receiver0 input.
	EMAC_MII_MDC	O	MPF6	MII/RMII Management Data Clock.

	EBI_AD14	O	MPF7	EBI address/data bus bit 14.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
102	PB.13	I/O	MFP0	General purpose digital I/O pin.
	UART4_CTS	I	MPF1	Clear to Send input pin for UART4.
	SPI2_MOSI1	I/O	MPF2	2nd SPI2 MOSI (Master Out, Slave In) pin.
	CAN0_TXD	I	MPF3	CAN bus transmitter0 input.
	EMAC_MII_MDIO	I/O	MPF6	MII/RMII Management Data I/O.
	EBI_AD15	O	MPF7	EBI address/data bus bit 15.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
103	PB.14	I/O	MFP0	General purpose digital I/O pin.
	I2S1_MCLK	O	MPF1	I2S1 master clock output pin.
	SC1_RST	O	MPF2	SmartCard1 reset pin.
	BRAKE01	I	MPF4	Brake input pin 1 of EPWMB.
	EMAC_MII_MDC	O	MPF6	MII/RMII Management Data Clock.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
104	PB.15	I/O	MFP0	General purpose digital I/O pin.
	I2S1_DO	O	MPF1	I2S1 data output.
	SC1_DAT	I/O	MPF2	SmartCard1 data pin.
	BRAKE00	I	MPF4	Brake input pin 0 of EPWMB.
	EMAC_MII_MDIO	I/O	MPF6	MII/RMII Management Data I/O.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
105	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
106	VSS	P	MFP0	Ground pin for digital circuit.
107	LDO_CAP	P	MFP0	LDO output pin.
108	PC.0	I/O	MFP0	General purpose digital I/O pin.

	I2S1_DI	I	MPF1	I2S1 data input.
	SC1_DAT	I/O	MPF2	SmartCard1 data pin.
	UART4_RXD	I	MPF3	Data receiver input pin for UART4.
	EMAC_REFCLK	I	MPF6	EMAC RMII mode clock input
	EBI_MCLK	O	MPF7	EBI interface clock output pin.
	INT2	I	MPF8	External interrupt2 input pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
109	PC.1	I/O	MFP0	General purpose digital I/O pin.
	I2S1_BCLK	O	MPF1	I2S1 bit clock pin.
	SC1_CLK	O	MPF2	SmartCard1 clock pin.
	UART4_TXD	O	MPF3	Data transmitter output pin for UART4.
	TM3_CNT_OUT	I/O	MPF5	Timer3 event counter input/toggle output.
	EMAC_MII_RXERR	I	MPF6	MII/RMII Receive Data error.
	EBI_AD13	O	MPF7	EBI address/data bus bit 13.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
110	PC.2	I/O	MFP0	General purpose digital I/O pin.
	I2S1_LRCK	O	MPF1	I2S1 left right channel clock.
	SC1_PWR	O	MPF2	SmartCard1 power pin.
	UART4_RTS	O	MPF3	Request to Send output pin for UART4.
	SPI0_SS0	I/O	MPF4	General purpose digital I/O pin.
	EMAC_MII_RXDV	I	MPF6	MII Receive Data Valid / RMII CRS_DV Input.
	EBI_AD12	O	MPF7	EBI address/data bus bit 12.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
111	PC.3	I/O	MFP0	General purpose digital I/O pin.
	I2S1_MCLK	O	MPF1	I2S1 master clock output pin.
	SC1_CD	I	MPF2	SmartCard1 card detect pin.

	UART4_CTS	I	MPF3	Clear to Send input pin for UART4.
	SPI0_MISO1	I/O	MPF4	2nd SPI0 MISO (Master In, Slave Out) pin.
	QEI0_Z	I	MPF5	Quadrature encoder phase Z input of QEI Unit 0.
	EMAC_MII_RXD1	I	MPF6	MII/RMII Receive Data Bus Bit 1.
	EBI_AD11	O	MPF7	EBI address/data bus bit 11.
	ECAP0_IC2	O	MPF8	Input 2 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
112	PC.4	I/O	MFP0	General purpose digital I/O pin.
	I2S1_DO	O	MPF1	I2S1 data output.
	SC1_RST	O	MPF2	SmartCard1 reset pin.
	SPI0_MOSI1	I/O	MPF4	2nd SPI0 MOSI (Master Out, Slave In) pin.
	QEI0_B	I	MPF5	Quadrature encoder phase B input of QEI Unit 0.
	EMAC_MII_RXD0	I	MPF6	MII/RMII Receive Data Bus Bit 0.
	EBI_AD10	O	MPF7	EBI address/data bus bit 10.
	ECAP0_IC1	O	MPF8	Input 1 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
113	PC.5	I/O	MFP0	General purpose digital I/O pin.
	CLKO	O	MFP1	Clock Output Pin.
	QEI0_A	I	MPF5	Quadrature encoder phase A input of QEI Unit 0.
	EMAC_MII_RXCLK	I	MPF6	MII Receive Clock Input.
	EBI_MCLK	O	MPF7	EBI interface clock output pin.
	ECAP0_IC0	O	MPF8	Input 0 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
114	PC.6	I/O	MFP0	General purpose digital I/O pin.
	TM2_EXT	I	MPF1	Timer2 external counter input
	SPI0_MISO0	I/O	MPF4	1st SPI0 MISO (Master In, Slave Out) pin.

	TM2_CNT_OUT	I/O	MPF5	Timer2 event counter input/toggle output.
	EMAC_MII_TXD0	O	MPF6	MII/RMII Transmit Data Bus bit 0.
	EBI_AD9	O	MPF7	EBI address/data bus bit 9.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
115	PC.7	I/O	MFP0	General purpose digital I/O pin.
	TM1_EXT	I	MPF1	Timer1 external counter input
	SPI0_MOSI0	I/O	MPF4	1st SPI0 MOSI (Master Out, Slave In) pin.
	EMAC_MII_TXD1	O	MPF6	MII/RMII Transmit Data Bus bit 1.
	EBI_AD8	O	MPF7	EBI address/data bus bit 8.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
116	PC.8	I/O	MFP0	General purpose digital I/O pin.
	TM0_EXT	I	MPF1	Timer0 external counter input
	SPI0_CLK	O	MPF4	SPI0 serial clock pin.
	EMAC_MII_TXEN	O	MPF6	MII/RMII Transmit Enable.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
117	PF.2	I/O	MFP0	General purpose digital I/O pin.
	SPI3_SS0	I/O	MPF1	General purpose digital I/O pin.
	SD0_DAT3	I/O	MPF4	SD mode #0 data line bit 3.
	EMAC_MII_RXD3	I	MPF6	MII Receive Data Bus Bit 3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
118	PF.3	I/O	MFP0	General purpose digital I/O pin.
	SPI3_CLK	O	MPF1	SPI3 serial clock pin.
	SD0_DAT2	I/O	MPF4	SD mode #0 data line bit 2.
	EMAC_MII_RXD2	I	MPF6	MII Receive Data Bus Bit 2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

119	PF.4	I/O	MFP0	General purpose digital I/O pin.
	SPI3_MISO0	I/O	MPF1	1st SPI3 MISO (Master In, Slave Out) pin.
	SD0_DAT1	I/O	MPF4	SD mode #0 data line bit 1.
	EMAC_MII_COL0	I	MPF6	MII Collision Detect Input Pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
120	PF.5	I/O	MFP0	General purpose digital I/O pin.
	SPI3_MOSI0	I/O	MPF1	1st SPI3 MOSI (Master Out, Slave In) pin.
	SD0_DAT0	I/O	MPF4	SD mode #0 data line bit 0.
	EMAC_MII_CRS	I	MPF6	MII Carrier Sense Input Pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
121	VSS	P	MFP0	Ground pin for digital circuit.
122	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
123	PF.6	I/O	MFP0	General purpose digital I/O pin.
	UART2_RXD	I	MPF1	Data receiver input pin for UART2.
	SD0_CDn	I	MPF4	SD mode #0 – card detect
	EMAC_MII_TXCLK	I	MPF6	MII Transmit Output Clock Pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
124	PF.7	I/O	MFP0	General purpose digital I/O pin.
	UART2_TXD	O	MPF1	Data transmitter output pin for UART2.
	SD0_CMD	I/O	MPF4	SD mode #0 – command/response
	EMAC_MII_TXD3	O	MPF6	MII Transmit Data Bus bit 3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
125	PF.8	I/O	MFP0	General purpose digital I/O pin.
	UART2_RTS	O	MPF1	Request to Send output pin for UART2.
	SD0_CLK	O	MPF4	SD mode #0– clock.

	EMAC_MII_TXD2	O	MPF6	MII Transmit Data Bus bit 2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
126	PH.2	I/O	MFP0	General purpose digital I/O pin.
	UART2_CTS	I	MPF1	Clear to Send input pin for UART2.
127	LDO_CAP	P	MFP0	LDO output pin.
128	VSS	P	MFP0	Ground pin for digital circuit.
129	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
130	PE.0	I/O	MFP0	General purpose digital I/O pin.
	ADC0_0	A	MPF1	ADC0 analog input.
	INT4	I	MPF8	External interrupt4 input pin.
131	PE.1	I/O	MFP0	General purpose digital I/O pin.
	ADC0_1	A	MPF1	ADC0 analog input.
	TM2_CNT_OUT	I/O	MPF3	Timer2 event counter input/toggle output.
132	PE.2	I/O	MFP0	General purpose digital I/O pin.
	ADC0_2	A	MPF1	ADC0 analog input.
	ACMP0_O	O	MPF2	Analog ccomparator0 output .
	SPI0_MISO0	I/O	MPF3	1st SPI0 MISO (Master In, Slave Out) pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
133	PE.3	I/O	MFP0	General purpose digital I/O pin.
	ADC0_3	A	MPF1	ADC0 analog input.
	ACMP0_P3	A	MPF2	Analog comparator0 positive input pin.
	SPI0_MOSI0	I/O	MPF3	1st SPI0 MOSI (Master Out, Slave In) pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
134	PE.4	I/O	MFP0	General purpose digital I/O pin.
	ADC0_4	A	MPF1	ADC0 analog input.

	ACMP0_P2	A	MPF2	Analog comparator0 positive input pin.
	SPI0_SS0	I/O	MPF3	General purpose digital I/O pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
135	PE.5	I/O	MFP0	General purpose digital I/O pin.
	ADC0_5	A	MPF1	ADC0 analog input.
	ACMP0_P1	A	MPF2	Analog comparator0 positive input pin.
	SPI0_CLK	O	MPF3	SPI0 serial clock pin.
	SD0_CDn	I	MPF4	SD mode #0 – card detect
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
136	PE.6	I/O	MFP0	General purpose digital I/O pin.
	ADC0_6	A	MPF1	ADC0 analog input.
	ACMP0_P0	A	MPF2	Analog comparator0 positive input pin.
	SPI0_MISO0	I/O	MPF3	1st SPI0 MISO (Master In, Slave Out) pin.
	SD0_CMD	I/O	MPF4	SD mode #0 – command/response
	EBI_nWR	O	MPF7	EBI write enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
137	PE.7	I/O	MFP0	General purpose digital I/O pin.
	ADC0_7	A	MPF1	ADC0 analog input.
	ACMP0_N	A	MPF2	Analog comparator0 negative input pin.
	SPI0_MOSI0	I/O	MPF3	1st SPI0 MOSI (Master Out, Slave In) pin.
	SD0_CLK	O	MPF4	SD mode #0– clock.
	EBI_nRD	O	MPF7	EBI read enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
138	AVSS	P	MFP0	Ground pin for digital circuit.
139	Vref	A	MFP0	Voltage reference input for ADC.

140	AVDD	P	MFP0	Power supply for internal analog circuit.
141	PE.8	I/O	MFP0	General purpose digital I/O pin.
	ADC1_0	A	MPF1	ADC1 analog input.
	ADC0_8	A	MPF1	ADC0 analog input.
	ACMP1_N	A	MPF2	Analog comparator1 negative input pin.
	TM1_CNT_OUT	I/O	MPF3	Timer1 event counter input/toggle output.
	SD0_DAT3	I/O	MPF4	SD mode #0 data line bit 3.
	EBI_ALE	O	MPF7	EBI address latch enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
142	PE.9	I/O	MFP0	General purpose digital I/O pin.
	ADC1_1	A	MPF1	ADC1 analog input.
	ADC0_9	A	MPF1	ADC0 analog input.
	ACMP1_P0	A	MPF2	Analog comparator1 positive input pin.
	SD0_DAT2	I/O	MPF4	SD mode #0 data line bit 2.
	EBI_nWRH	O	MPF7	EBI write enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
143	PE.10	I/O	MFP0	General purpose digital I/O pin.
	ADC1_2	A	MPF1	ADC1 analog input.
	ADC0_10	A	MPF1	ADC0 analog input.
	ACMP1_P1	A	MPF2	Analog comparator1 positive input pin.
	SPI0_MISO1	I/O	MPF3	2nd SPI0 MISO (Master In, Slave Out) pin.
	SD0_DAT1	I/O	MPF4	SD mode #0 data line bit 1.
	EBI_nWRL	O	MPF7	EBI write enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
144	PE.11	I/O	MFP0	General purpose digital I/O pin.
	ADC1_3	A	MPF1	ADC1 analog input.

	ADC0_11	A	MPF1	ADC0 analog input.
	ACMP1_P2	A	MPF2	Analog comparator1 positive input pin.
	SPI0_MOSI1	I/O	MPF3	2nd SPI0 MOSI (Master Out, Slave In) pin.
	SD0_DAT0	I/O	MPF4	SD mode #0 data line bit 0.
	ACMP2_P3	A	MPF5	Analog comparator2 positive input pin.
	EBI_nCS0	O	MPF7	EBI chip select 0 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

Note: Pin Type I = Digital Input, O = Digital Output; A = Analog Pin; P = Power Pin;

4.3.8 NuMicro™ NUC472 Package LQFP 176-pin Description

MFP = Multi-function pin.

Pin No.	Pin Name	Type	MFP*	Description
1	PE.12	I/O	MFP0	General purpose digital I/O pin.
	ADC1_4	A	MPF1	ADC1 analog input.
	ACMP1_P3	A	MPF2	Analog comparator1 positive input pin.
	ACMP2_P2	A	MPF3	Analog comparator2 positive input pin.
	EBI_nCS1	O	MPF7	EBI chip select 1 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
2	PE.13	I/O	MFP0	General purpose digital I/O pin.
	ADC1_5	A	MPF1	ADC1 analog input.
	ACMP2_P1	A	MPF3	Analog comparator2 positive input pin.
	EBI_nCS2	O	MPF7	EBI chip select 2 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
3	PE.14	I/O	MFP0	General purpose digital I/O pin.
	ADC1_6	A	MPF1	ADC1 analog input.
	ACMP2_P0	A	MPF3	Analog comparator2 positive input pin.
	EBI_nCS3	O	MPF7	EBI chip select 3 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
4	PE.15	I/O	MFP0	General purpose digital I/O pin.
	ADC1_7	A	MPF1	ADC1 analog input.
	ACMP2_N	A	MPF3	Analog comparator2 negative input pin.
5	PF.9	I/O	MFP0	General purpose digital I/O pin.
	OPA0_IN+	I/O	MPF1	General purpose digital I/O pin.
	PWM0_0	I/O	MPF4	PWM0_0 output/capture input.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

6	PF.10	I/O	MFP0	General purpose digital I/O pin.
	OPA0_IN-	I/O	MPF1	General purpose digital I/O pin.
	PWM0_1	I/O	MPF4	PWM0_1 output/capture input.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
7	PF.11	I/O	MFP0	General purpose digital I/O pin.
	OPA0_O	O	MPF1	Operational amplifier output pin
	UART1_RTS	O	MPF2	Request to Send output pin for UART1.
8	PF.12	I/O	MFP0	General purpose digital I/O pin.
	OPA1_IN+	I/O	MPF1	General purpose digital I/O pin.
	UART1_CTS	I	MPF2	Clear to Send input pin for UART1.
9	PF.13	I/O	MFP0	General purpose digital I/O pin.
	OPA1_IN-	I/O	MPF1	General purpose digital I/O pin.
	UART1_TXD	O	MPF2	Data transmitter output pin for UART1.
10	PF.14	I/O	MFP0	General purpose digital I/O pin.
	OPA1_O	O	MPF1	Operational amplifier output pin
	UART1_RXD	I	MPF2	Data receiver input pin for UART1.
11	VSS	P	MFP0	Ground pin for digital circuit.
12	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
13	PF.15	I/O	MFP0	General purpose digital I/O pin.
	UART0_RTS	O	MPF1	Request to Send output pin for UART0.
14	PG.0	I/O	MFP0	General purpose digital I/O pin.
	UART0_CTS	I	MPF1	Clear to Send input pin for UART0.
	INT6	I	MPF8	External interrupt6 input pin.
15	PG.1	I/O	MFP0	General purpose digital I/O pin.
	UART0_RXD	I	MPF1	Data receiver input pin for UART0.
16	PG.2	I/O	MFP0	General purpose digital I/O pin.

	UART0_TXD	O	MPF1	Data transmitter output pin for UART0.
17	PC.12	I/O	MFP0	General purpose digital I/O pin.
	SPI1_SS0	I/O	MPF1	1st SPI1 slave select pin.
	SC4_CD	I	MPF2	SmartCard4 card detect pin.
	SD1_CDn	I	MPF4	SD mode #1 – card detect
	CAP_DATA7	I	MPF5	Image data input bus bit 7.
	EBI_A0	O	MPF7	EBI address bus bit0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
18	PC.13	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MOSI1	I/O	MPF1	2nd SPI1 MOSI (Master Out, Slave In) pin.
	SC4_RST	O	MPF2	SmartCard4 reset pin.
	SD1_CMD	I/O	MPF4	SD mode #1 – command/response
	CAP_DATA6	I	MPF5	Image data input bus bit 7.
	EBI_A1	O	MPF7	EBI address bus bit1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
19	PC.14	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MISO1	I/O	MPF1	2nd SPI1 MISO (Master In, Slave Out) pin.
	SC4_PWR	O	MPF2	SmartCard4 power pin.
	TM3_EXT	I	MPF3	Timer3 external counter input
	SD1_CLK	O	MPF4	SD mode #1 – clock.
	CAP_DATA5	I	MPF5	Image data input bus bit 7.
	EBI_A2	O	MPF7	EBI address bus bit2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
20	PC.15	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MOSI0	I/O	MPF1	1st SPI1 MOSI (Master Out, Slave In) pin.
	SC4_DAT	I/O	MPF2	SmartCard4 data pin.

	SD1_DAT3	I/O	MPF4	SD mode #1 data line bit 3.
	CAP_DATA4	I	MPF5	Image data input bus bit 7.
	EBI_A3	O	MPF7	EBI address bus bit3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
21	PD.0	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MISO0	I/O	MPF1	1st SPI1 MISO (Master In, Slave Out) pin.
	SC4_CLK	O	MPF2	SmartCard4 clock pin.
	SD1_DAT2	I/O	MPF4	SD mode #1 data line bit 2.
	CAP_DATA3	I	MPF5	Image data input bus bit 7.
	EBI_A4	O	MPF7	EBI address bus bit4.
	INT3	I	MPF8	External interrupt3 input pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
22	PD.1	I/O	MFP0	General purpose digital I/O pin.
	SPI1_CLK	O	MPF1	SPI1 serial clock pin.
	TM0_CNT_OUT	I/O	MPF3	Timer0 event counter input/toggle output.
	SD1_DAT1	I/O	MPF4	SD mode #1 data line bit 1.
	CAP_DATA2	I	MPF5	Image data input bus bit 7.
	EBI_A5	O	MPF7	EBI address bus bit5.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
23	PD.2	I/O	MFP0	General purpose digital I/O pin.
	STADC	A	MPF1	ADC analog input.
	I2C3_SCL	I/O	MPF2	I2C3 clock pin.
	SD1_DAT0	I/O	MPF4	SD mode #1 data line bit 0.
	CAP_DATA1	I	MPF5	Image data input bus bit 7.
	EBI_A6	O	MPF7	EBI address bus bit6.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
24	PH.11	I/O	MFP0	General purpose digital I/O pin.
	UART3_RXD	I	MPF1	Data receiver input pin for UART3.
25	PH.12	I/O	MFP0	General purpose digital I/O pin.
	UART3_TXD	O	MPF1	Data transmitter output pin for UART3.
26	PH.13	I/O	MFP0	General purpose digital I/O pin.
	UART3_RTS	O	MPF1	Request to Send output pin for UART3.
27	PH.14	I/O	MFP0	General purpose digital I/O pin.
	UART3_CTS	I	MPF1	Clear to Send input pin for UART3.
28	PD.3	I/O	MFP0	General purpose digital I/O pin.
	SC5_CLK	O	MPF1	SmartCard5 clock pin.
	I2C3_SDA	I/O	MPF2	I2C3 data input/output pin.
	ACMP2_O	O	MPF3	Analog ccomparator2 output .
	SD0_CDn	I	MPF4	SD mode #0 – card detect
	CAP_DATA0	I	MPF5	Image data input bus bit 7.
	EBI_A7	O	MPF7	EBI address bus bit7.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
29	PD.4	I/O	MFP0	General purpose digital I/O pin.
	SC5_CD	I	MPF1	SmartCard5 card detect pin.
	UART3_RXD	I	MPF2	Data receiver input pin for UART3.
	ACMP1_O	O	MPF3	Analog ccomparator1 output.
	CAP_SCLK	O	MPF5	Image capture interface sensor clock pin.
	EBI_A8	O	MPF7	EBI address bus bit8.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
30	PD.5	I/O	MFP0	General purpose digital I/O pin.
	SC5_RST	O	MPF1	SmartCard5 reset pin.

	UART3_TXD	O	MPF2	Data transmitter output pin for UART3.
	CAP_VSYNC	I	MPF5	Image capture interface VSYNC input pin.
	EBI_A9	O	MPF7	EBI address bus bit9.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
31	PD.6	I/O	MFP0	General purpose digital I/O pin.
	SC5_PWR	O	MPF1	SmartCard5 power pin.
	UART3_RTS	O	MPF2	Request to Send output pin for UART3.
	SD0_CMD	I/O	MPF4	SD mode #0 – command/response
	CAP_HSYNC	I	MPF5	Image capture interface HSYNC input pin.
	EBI_A10	O	MPF7	EBI address bus bit10.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
32	PD.7	I/O	MFP0	General purpose digital I/O pin.
	SC5_DAT	I/O	MPF1	SmartCard5 data pin.
	UART3_CTS	I	MPF2	Clear to Send input pin for UART3.
	SD0_CLK	O	MPF4	SD mode #0 – clock.
	CAP_PIXCLK	I	MPF5	Image capture interface pix clock input pin.
	EBI_A11	O	MPF7	EBI address bus bit11.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
33	PH.15	I/O	MFP0	General purpose digital I/O pin.
	SC5_CLK	O	MPF2	SmartCard5 clock pin.
34	PI.0	I/O	MFP0	General purpose digital I/O pin.
	SC5_RST	O	MPF2	SmartCard5 reset pin.
35	PI.1	I/O	MFP0	General purpose digital I/O pin.
	SC5_PWR	O	MPF2	SmartCard5 power pin.
36	PI.2	I/O	MFP0	General purpose digital I/O pin.
	SC5_DAT	I/O	MPF1	SmartCard5 data pin.

37	PG.13	I/O	MFP0	General purpose digital I/O pin.
	XT1_IN	I	MPF1	External 4~24 MHz (high-speed) crystal input pin.
38	PG.12	I/O	MFP0	General purpose digital I/O pin.
	XT1_OUT	O	MPF1	External 4~24 MHz (high-speed) crystal output pin.
39	nRESET	I	MFP0	External reset input: active LOW, with an internal pull-up. Set this pin low reset to initial state.
40	LDO_CAP	P	MFP0	LDO output pin.
41	VSS	P	MFP0	Ground pin for digital circuit.
42	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
43	PG.10	I/O	MFP0	General purpose digital I/O pin.
	ICE_CLK	I	MPF1	Serial wired debugger clock pin
44	PG.11	I/O	MFP0	General purpose digital I/O pin.
	ICE_DAT	I/O	MPF1	Serial wired debugger data pin
45	PG.15	I/O	MFP0	General purpose digital I/O pin.
	X32K_IN	I	MPF1	External 32.768 kHz (low-speed) crystal input pin.
	I2C1_SCL	I/O	MPF3	I2C1 clock pin.
46	PG.14	I/O	MFP0	General purpose digital I/O pin.
	X32K_OUT	O	MPF1	External 32.768 kHz (low-speed) crystal output pin.
	I2C1_SDA	I/O	MPF3	I2C1 data input/output pin.
47	VBAT	P	MFP0	Battery power input pin.
48	PA.0	I/O	MFP0	General purpose digital I/O pin.
	TAMPER0	I/O	MPF1	Tamper detect pin 0.
	SC0_CD	I	MPF2	SmartCard0 card detect pin.
	CAN1_RXD	I	MPF3	CAN bus receiver1 input.
	INT0	I	MPF8	External interrupt0 input pin.
49	PA.1	I/O	MFP0	General purpose digital I/O pin.
	TAMPER1	I/O	MPF1	Tamper detect pin 1.

	SC5_CD	I	MPF2	SmartCard5 card detect pin.
	CAN1_TXD	I	MPF3	CAN bus transmitter1 input.
	EBI_A22	O	MPF7	EBI address bus bit22.
50	PI.3	I/O	MFP0	General purpose digital I/O pin.
	SPI3_SS0	I/O	MPF1	General purpose digital I/O pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
51	PI.4	I/O	MFP0	General purpose digital I/O pin.
	SPI3_CLK	O	MPF1	SPI3 serial clock pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
52	PI.5	I/O	MFP0	General purpose digital I/O pin.
	SPI3_MISO0	I/O	MPF1	1st SPI3 MISO (Master In, Slave Out) pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
53	PI.6	I/O	MFP0	General purpose digital I/O pin.
	SPI3_MOSI0	I/O	MPF1	1st SPI3 MOSI (Master Out, Slave In) pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
54	PD.8	I/O	MFP0	General purpose digital I/O pin.
	SPI3_MISO1	I/O	MPF1	2nd SPI3 MISO (Master In, Slave Out) pin.
	I2C0_SCL	I/O	MPF2	I2C0 clock pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
55	PD.9	I/O	MFP0	General purpose digital I/O pin.
	SPI3_MOSI1	I/O	MPF1	2nd SPI3 MOSI (Master Out, Slave In) pin.
	I2C0_SDA	I/O	MPF2	I2C0 data input/output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
56	PA.2	I/O	MFP0	General purpose digital I/O pin.
	SC2_DAT	I/O	MPF1	SmartCard2 data pin.

	SPI3_MISO0	I/O	MPF2	1st SPI3 MISO (Master In, Slave Out) pin.
	I2S0_MCLK	O	MPF3	I2S0 master clock output pin.
	BRAKE11	I	MPF4	Brake input pin 1 of EPWMA.
	CAP_SFIELD	I	MPF5	Video input interface SFIELD input pin.
	EBI_A12	O	MPF7	EBI address bus bit12.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
57	PA.3	I/O	MFP0	General purpose digital I/O pin.
	SC2_CLK	O	MPF1	SmartCard2 clock pin.
	SPI3_MOSI0	I/O	MPF2	1st SPI3 MOSI (Master Out, Slave In) pin.
	I2S0_DO	O	MPF3	I2S0 data output.
	BRAKE10	I	MPF4	Brake input pin 0 of EPWMA.
	EBI_A13	O	MPF7	EBI address bus bit13.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
58	PA.4	I/O	MFP0	General purpose digital I/O pin.
	SC2_PWR	O	MPF1	SmartCard2 power pin.
	SPI3_CLK	O	MPF2	SPI3 serial clock pin.
	I2S0_DI	I	MPF3	I2S0 data input.
	QEI1_Z	I	MPF5	Quadrature encoder phase Z input of QEI Unit 1.
	EBI_A14	O	MPF7	EBI address bus bit14.
	ECAP1_IC2	I	MPF8	Input 2 of enhanced capture unit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
59	PA.5	I/O	MFP0	General purpose digital I/O pin.
	SC2_RST	O	MPF1	SmartCard2 reset pin.
	SPI3_SS0	I/O	MPF2	General purpose digital I/O pin.
	I2S0_BCLK	O	MPF3	I2S0 bit clock pin.
	PWM0_0	I/O	MPF4	PWM0_0 output/capture input.

	QEI1_B	I	MPF5	Quadrature encoder phase B input of QEI Unit 1.
	EBI_A15	O	MPF7	EBI address bus bit15.
	ECAP1_IC1	I	MPF8	Input 1 of enhanced capture unit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
60	PA.6	I/O	MFP0	General purpose digital I/O pin.
	SC2_CD	I	MPF1	SmartCard2 card detect pin.
	I2S0_LRCK	O	MPF3	I2S0 left right channel clock.
	PWM0_1	I/O	MPF4	PWM0_1 output/capture input.
	QEI1_A	I	MPF5	Quadrature encoder phase A input of QEI Unit 1.
	CAN1_TXD	I	MPF6	CAN bus transmitter1 input.
	EBI_A16	O	MPF7	EBI address bus bit16.
	ECAP1_IC0	I	MPF8	Input 0 of enhanced capture unit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
61	PI.7	I/O	MFP0	General purpose digital I/O pin.
	I2C2_SCL	I/O	MPF1	I2C2 clock pin.
	SPI3_MISO1	I/O	MPF2	2nd SPI3 MISO (Master In, Slave Out) pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
62	PI.8	I/O	MFP0	General purpose digital I/O pin.
	I2C2_SDA	I/O	MPF1	I2C2 data input/output pin.
	SPI3_MOSI1	I/O	MPF2	2nd SPI3 MOSI (Master Out, Slave In) pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
63	PG.3	I/O	MFP0	General purpose digital I/O pin.
	PS2_CLK	O	MPF1	PS2 clock pin.
	I2S1_DO	O	MPF2	I2S1 data output.
	SC1_RST	O	MPF3	SmartCard1 reset pin.

64	PG.4	I/O	MFP0	General purpose digital I/O pin.
	PS2_DAT	I/O	MPF1	PS2 data pin.
	I2S1_DI	I	MPF2	I2S1 data input.
	SC1_PWR	O	MPF3	SmartCard1 power pin.
65	PG.5	I/O	MFP0	General purpose digital I/O pin.
	I2S1_BCLK	O	MPF2	I2S1 bit clock pin.
	SC1_DAT	I/O	MPF3	SmartCard1 data pin.
66	PG.6	I/O	MFP0	General purpose digital I/O pin.
	I2S1_LRCK	O	MPF2	I2S1 left right channel clock.
	SC1_CLK	O	MPF3	SmartCard1 clock pin.
67	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
68	VSS	P	MFP0	Ground pin for digital circuit.
69	PA.7	I/O	MFP0	General purpose digital I/O pin.
	SC0_CLK	O	MPF2	SmartCard0 clock pin.
	SPI3_SS0	I/O	MPF3	General purpose digital I/O pin.
	PWM1_3	I/O	MPF4	PWM1_3 output/capture input.
	EPWM0_5	I/O	MPF5	PWM0_5 output/capture input.
	EBI_A17	O	MPF7	EBI address bus bit17.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
70	PA.8	I/O	MFP0	General purpose digital I/O pin.
	SC0_RST	O	MPF2	SmartCard0 reset pin.
	SPI3_CLK	O	MPF3	SPI3 serial clock pin.
	PWM1_2	I/O	MPF4	PWM1_2 output/capture input.
	EPWM0_4	I/O	MPF5	PWM0_4 output/capture input.
	EBI_A18	O	MPF7	EBI address bus bit18.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

71	PA.9	I/O	MFP0	General purpose digital I/O pin.
	SC0_PWR	O	MPF2	SmartCard0 power pin.
	SPI3_MISO0	I/O	MPF3	1st SPI3 MISO (Master In, Slave Out) pin.
	PWM1_1	I/O	MPF4	PWM1_1 output/capture input.
	EPWM0_3	I/O	MPF5	PWM0_3 output/capture input.
	EBI_A19	O	MPF7	EBI address bus bit19.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
72	PA.10	I/O	MFP0	General purpose digital I/O pin.
	SC0_DAT	I/O	MPF2	SmartCard0 data pin.
	SPI3_MOSI0	I/O	MPF3	1st SPI3 MOSI (Master Out, Slave In) pin.
	PWM1_0	I/O	MPF4	PWM1_0 output/capture input.
	EPWM0_2	I/O	MPF5	PWM0_2 output/capture input.
	EBI_A20	O	MPF7	EBI address bus bit20.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
73	PA.11	I/O	MFP0	General purpose digital I/O pin.
	UART0_RTS	O	MPF1	Request to Send output pin for UART0.
	SPI3_MISO1	I/O	MPF3	2nd SPI3 MISO (Master In, Slave Out) pin.
	PWM0_5	I/O	MPF4	PWM0_5 output/capture input.
	EPWM0_1	I/O	MPF5	PWM0_1 output/capture input.
	EBI_AD0	O	MPF7	EBI address/data bus bit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
74	PA.12	I/O	MFP0	General purpose digital I/O pin.
	UART0_CTS	I	MPF1	Clear to Send input pin for UART0.
	SPI3_MOSI1	I/O	MPF3	2nd SPI3 MOSI (Master Out, Slave In) pin.
	PWM0_4	I/O	MPF4	PWM0_4 output/capture input.
	EPWM0_0	I/O	MPF5	PWM0_0 output/capture input.

	EBI_AD1	O	MPF7	EBI address/data bus bit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
75	PA.13	I/O	MFP0	General purpose digital I/O pin.
	UART0_RXD	I	MPF1	Data receiver input pin for UART0.
	SC3_DAT	I/O	MPF3	SmartCard3 data pin.
	PWM1_4	I/O	MPF4	PWM1_4 output/capture input.
	EBI_AD2	O	MPF7	EBI address/data bus bit 2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
76	PA.14	I/O	MFP0	General purpose digital I/O pin.
	UART0_TXD	O	MPF1	Data transmitter output pin for UART0.
	SC3_CLK	O	MPF3	SmartCard3 clock pin.
	PWM1_5	I/O	MPF4	PWM1_5 output/capture input.
	EBI_AD3	O	MPF7	EBI address/data bus bit 3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
77	PI.9	I/O	MFP0	General purpose digital I/O pin.
	I2C4_SCL	I/O	MPF4	I2C4 clock pin.
78	PI.10	I/O	MFP0	General purpose digital I/O pin.
79	PD.10	I/O	MFP0	General purpose digital I/O pin.
	SC3_DAT	I/O	MPF1	SmartCard3 data pin.
	I2C4_SCL	I/O	MPF2	I2C4 clock pin.
80	PD.11	I/O	MFP0	General purpose digital I/O pin.
	SC3_RST	O	MPF1	SmartCard3 reset pin.
	TM3_CNT_OUT	I/O	MPF3	Timer3 event counter input/toggle output.
81	PD.12	I/O	MFP0	General purpose digital I/O pin.
	SC3_CLK	O	MPF1	SmartCard3 clock pin.
	I2C4_SDA	I/O	MPF2	I2C4 data input/output pin.

82	PA.15	I/O	MFP0	General purpose digital I/O pin.
	SC3_PWR	O	MPF1	SmartCard3 power pin.
	UART2_RTS	O	MPF2	Request to Send output pin for UART2.
	I2C0_SCL	I/O	MPF4	I2C0 clock pin.
	EBI_A21	O	MPF7	EBI address bus bit21.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
83	PC.9	I/O	MFP0	General purpose digital I/O pin.
	STADC	A	MPF1	ADC analog input.
	UART2_CTS	I	MPF2	Clear to Send input pin for UART2.
	SC3_RST	O	MPF3	SmartCard3 reset pin.
	I2C0_SDA	I/O	MPF4	I2C0 data input/output pin.
	CAP_DATA1	I	MPF5	Image data input bus bit 7.
	I2C3_SCL	I/O	MPF6	I2C3 clock pin.
	EBI_A22	O	MPF7	EBI address bus bit22.
	SD1_DAT0	I/O	MPF8	SD mode #1 data line bit 0.
	EBI_A6	O	MPF9	EBI address bus bit6.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
84	PC.10	I/O	MFP0	General purpose digital I/O pin.
	SC3_CD	I	MPF1	SmartCard3 card detect pin.
	UART2_RXD	I	MPF2	Data receiver input pin for UART2.
	PWM0_2	I/O	MPF4	PWM0_2 output/capture input.
	EBI_A23	O	MPF6	EBI address bus bit23.
	EBI_AD2	O	MPF7	EBI address/data bus bit 2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
85	PC.11	I/O	MFP0	General purpose digital I/O pin.
	UART2_TXD	O	MPF2	Data transmitter output pin for UART2.

	PWM0_3	I/O	MPF4	PWM0_3 output/capture input.
	EBI_A24	O	MPF6	EBI address bus bit24.
	EBI_AD3	O	MPF7	EBI address/data bus bit 3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
86	LDO_CAP	P	MFP0	LDO output pin.
87	VSS	P	MFP0	Ground pin for digital circuit.
88	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
89	PD.13	I/O	MFP0	General purpose digital I/O pin.
	SPI1_SS0	I/O	MPF1	1st SPI1 slave select pin.
	UART5_CTS	I	MPF2	Clear to Send input pin for UART5.
	ECAP0_IC2	O	MPF3	Input 2 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
90	PD.14	I/O	MFP0	General purpose digital I/O pin.
	SPI1_CLK	O	MPF1	SPI1 serial clock pin.
	UART5_RTS	O	MPF2	Request to Send output pin for UART5.
	ECAP0_IC1	O	MPF3	Input 1 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
91	PD.15	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MISO0	I/O	MPF1	1st SPI1 MISO (Master In, Slave Out) pin.
	UART5_TXD	O	MPF2	Data transmitter output pin for UART5.
	ECAP0_IC0	O	MPF3	Input 0 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
92	PF.0	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MOSI0	I/O	MPF1	1st SPI1 MOSI (Master Out, Slave In) pin.
	UART5_RXD	I	MPF2	Data receiver input pin for UART5.

	INT5	I	MPF8	External interrupt5 input pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
93	VRES	A	MFP0	USB PHY VRES ground input pin. Add an 8.2K ohm resistor to VSSA.
94	VBUS	A	MFP0	USB PHY VBUS power input pin.
95	USB_VDD33_CAP	P	MFP0	Internal power regulator output 3.3V decoupling pin.
96	VSSA	P	MFP0	Ground pin for digital circuit. Add a Feritte Bead to digital ground VSS.
97	USB0_D-	A	MFP0	USB0 differential signal D+.
98	USB0_D+	A	MFP0	USB0 differential signal D+.
99	USB0_OTG_ID	I	MFP0	USB0OTG ID pin.
100	PB.0	I/O	MFP0	General purpose digital I/O pin.
	USB0_VBUS_ST	I	MPF1	USB0 external VBUS regulator status
	I2C4_SCL	I/O	MPF2	I2C4 clock pin.
	INT1	I	MPF8	External interrupt1 input pin.
101	PB.1	I/O	MFP0	General purpose digital I/O pin.
	USB0_VBUS_EN	O	MPF1	USB0 external VBUS regulator enabled
	I2C4_SDA	I/O	MPF2	I2C4 data input/output pin.
	TM1_CNT_OUT	I/O	MPF3	Timer1 event counter input/toggle output.
102	PG.7	I/O	MFP0	General purpose digital I/O pin.
	SPI2_MISO0	I/O	MPF1	1st SPI2 MISO (Master In, Slave Out) pin.
	I2S1_MCLK	O	MPF2	I2S1 master clock output pin.
	SC1_CD	I	MPF3	SmartCard1 card detect pin.
	SC3_RST	O	MPF4	SmartCard3 reset pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
103	PG.8	I/O	MFP0	General purpose digital I/O pin.
	SPI2_MOSI0	I/O	MPF1	1st SPI2 MOSI (Master Out, Slave In) pin.

	I2S1_DO	O	MPF2	I2S1 data output.
	UART4_RTS	O	MPF3	Request to Send output pin for UART4.
	SC3_DAT	I/O	MPF4	SmartCard3 data pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
104	PG.9	I/O	MFP0	General purpose digital I/O pin.
	SPI2_CLK	O	MPF1	SPI2 serial clock pin.
	I2S1_DI	I	MPF2	I2S1 data input.
	UART4_CTS	I	MPF3	Clear to Send input pin for UART4.
	SC3_CLK	O	MPF4	SmartCard3 clock pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
105	PI.11	I/O	MFP0	General purpose digital I/O pin.
	SPI2_SS0	I/O	MPF1	General purpose digital I/O pin.
	I2S1_BCLK	O	MPF2	I2S1 bit clock pin.
	I2C4_SCL	I/O	MPF3	I2C4 clock pin.
	SC3_PWR	O	MPF4	SmartCard3 power pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
106	PI.12	I/O	MFP0	General purpose digital I/O pin.
	SPI2_MISO1	I/O	MPF1	2nd SPI2 MISO (Master In, Slave Out) pin.
	I2S1_LRCK	O	MPF2	I2S1 left right channel clock.
	I2C4_SDA	I/O	MPF3	I2C4 data input/output pin.
	SC3_CD	I	MPF4	SmartCard3 card detect pin.
107	PI.13	I/O	MFP0	General purpose digital I/O pin.
108	PI.14	I/O	MFP0	General purpose digital I/O pin.
109	PI.15	I/O	MFP0	General purpose digital I/O pin.
110	PF.1	I/O	MFP0	General purpose digital I/O pin.
	SPI2_MOSI1	I/O	MPF1	2nd SPI2 MOSI (Master Out, Slave In) pin.

	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
111	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
112	VSS	P	MFP0	Ground pin for digital circuit.
113	PB.2	I/O	MFP0	General purpose digital I/O pin.
	UART1_RXD	I	MPF1	Data receiver input pin for UART1.
	SPI2_SS0	I/O	MPF2	General purpose digital I/O pin.
	USB1_D-	A	MPF3	USB1 differential signal D+.
	EBI_AD4	O	MPF7	EBI address/data bus bit 4.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
114	PB.3	I/O	MFP0	General purpose digital I/O pin.
	UART1_TXD	O	MPF1	Data transmitter output pin for UART1.
	SPI2_CLK	O	MPF2	SPI2 serial clock pin.
	USB1_D+	A	MPF3	USB1 differential signal D+.
	EBI_AD5	O	MPF7	EBI address/data bus bit 5.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
115	PB.4	I/O	MFP0	General purpose digital I/O pin.
	UART1_RTS	O	MPF1	Request to Send output pin for UART1.
	SPI2_MISO0	I/O	MPF2	1st SPI2 MISO (Master In, Slave Out) pin.
	UART4_RXD	I	MPF3	Data receiver input pin for UART4.
	TM0_CNT_OUT	I/O	MPF4	Timer0 event counter input/toggle output.
	EBI_AD6	O	MPF7	EBI address/data bus bit 6.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
116	PB.5	I/O	MFP0	General purpose digital I/O pin.
	UART1_CTS	I	MPF1	Clear to Send input pin for UART1.
	SPI2_MOSI0	I/O	MPF2	1st SPI2 MOSI (Master Out, Slave In) pin.

	UART4_TXD	O	MPF3	Data transmitter output pin for UART4.
	EBI_AD7	O	MPF7	EBI address/data bus bit 7.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
117	PB.6	I/O	MFP0	General purpose digital I/O pin.
	I2C2_SCL	I/O	MPF1	I2C2 clock pin.
	BRAKE01	I	MPF2	Brake input pin 1 of EPWMB.
	UART4_RTS	O	MPF3	Request to Send output pin for UART4.
	PWM1_4	I/O	MPF4	PWM1_4 output/capture input.
	EPWM1_0	I/O	MPF5	PWM1_0 output/capture input.
	EBI_AD8	O	MPF7	EBI address/data bus bit 8.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
118	PB.7	I/O	MFP0	General purpose digital I/O pin.
	I2C2_SDA	I/O	MPF1	I2C2 data input/output pin.
	BRAKE00	I	MPF2	Brake input pin 0 of EPWMB.
	UART4_CTS	I	MPF3	Clear to Send input pin for UART4.
	PWM1_5	I/O	MPF4	PWM1_5 output/capture input.
	EPWM1_1	I/O	MPF5	PWM1_1 output/capture input.
	EBI_AD9	O	MPF7	EBI address/data bus bit 9.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
119	PB.8	I/O	MFP0	General purpose digital I/O pin.
	UART5_CTS	I	MPF1	Clear to Send input pin for UART5.
	EPWM1_2	I/O	MPF5	PWM1_2 output/capture input.
	EBI_AD10	O	MPF7	EBI address/data bus bit 1.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
120	PB.9	I/O	MFP0	General purpose digital I/O pin.

	UART5_RTS	O	MPF1	Request to Send output pin for UART5.
	EPWM1_3	I/O	MPF5	PWM1_3 output/capture input.
	EBI_AD11	O	MPF7	EBI address/data bus bit 11.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
121	PB.10	I/O	MFP0	General purpose digital I/O pin.
	UART5_TXD	O	MPF1	Data transmitter output pin for UART5.
	EPWM1_4	I/O	MPF5	PWM1_4 output/capture input.
	EBI_AD12	O	MPF7	EBI address/data bus bit 12.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
122	PB.11	I/O	MFP0	General purpose digital I/O pin.
	UART5_RXD	I	MPF1	Data receiver input pin for UART5.
	EPWM1_5	I/O	MPF5	PWM1_5 output/capture input.
	EBI_AD13	O	MPF7	EBI address/data bus bit 13.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
123	PH.0	I/O	MFP0	General purpose digital I/O pin.
	I2C1_SCL	I/O	MPF1	I2C1 clock pin.
	UART4_RXD	I	MPF2	Data receiver input pin for UART4.
	CAN1_RXD	I	MPF3	CAN bus receiver1 input.
	INT7	I	MPF8	External interrupt7 input pin.
124	PH.1	I/O	MFP0	General purpose digital I/O pin.
	UART4_TXD	O	MPF1	Data transmitter output pin for UART4.
	I2C1_SDA	I/O	MPF2	I2C1 data input/output pin.
	CAN1_TXD	I	MPF3	CAN bus transmitter1 input.
125	PB.12	I/O	MFP0	General purpose digital I/O pin.
	UART4_RTS	O	MPF1	Request to Send output pin for UART4.
	SPI2_MISO1	I/O	MPF2	2nd SPI2 MISO (Master In, Slave Out) pin.

	CAN0_RXD	I	MPF3	CAN bus receiver0 input.
	EMAC_MII_MDC	O	MPF6	MII/RMII Management Data Clock.
	EBI_AD14	O	MPF7	EBI address/data bus bit 14.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
126	PB.13	I/O	MFP0	General purpose digital I/O pin.
	UART4_CTS	I	MPF1	Clear to Send input pin for UART4.
	SPI2_MOSI1	I/O	MPF2	2nd SPI2 MOSI (Master Out, Slave In) pin.
	CAN0_TXD	I	MPF3	CAN bus transmitter0 input.
	EMAC_MII_MDIO	I/O	MPF6	MII/RMII Management Data I/O.
	EBI_AD15	O	MPF7	EBI address/data bus bit 15.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
127	PB.14	I/O	MFP0	General purpose digital I/O pin.
	I2S1_MCLK	O	MPF1	I2S1 master clock output pin.
	SC1_RST	O	MPF2	SmartCard1 reset pin.
	BRAKE01	I	MPF4	Brake input pin 1 of EPWMB.
	EMAC_MII_MDC	O	MPF6	MII/RMII Management Data Clock.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
128	PB.15	I/O	MFP0	General purpose digital I/O pin.
	I2S1_DO	O	MPF1	I2S1 data output.
	SC1_DAT	I/O	MPF2	SmartCard1 data pin.
	BRAKE00	I	MPF4	Brake input pin 0 of EPWMB.
	EMAC_MII_MDIO	I/O	MPF6	MII/RMII Management Data I/O.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
129	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
130	VSS	P	MFP0	Ground pin for digital circuit.

131	LDO_CAP	P	MFP0	LDO output pin.
132	PC.0	I/O	MFP0	General purpose digital I/O pin.
	I2S1_DI	I	MPF1	I2S1 data input.
	SC1_DAT	I/O	MPF2	SmartCard1 data pin.
	UART4_RXD	I	MPF3	Data receiver input pin for UART4.
	EMAC_REFCLK	I	MPF6	EMAC RMII mode clock input
	EBI_MCLK	O	MPF7	EBI interface clock output pin.
	INT2	I	MPF8	External interrupt2 input pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
133	PC.1	I/O	MFP0	General purpose digital I/O pin.
	I2S1_BCLK	O	MPF1	I2S1 bit clock pin.
	SC1_CLK	O	MPF2	SmartCard1 clock pin.
	UART4_TXD	O	MPF3	Data transmitter output pin for UART4.
	TM3_CNT_OUT	I/O	MPF5	Timer3 event counter input/toggle output.
	EMAC_MII_RXERR	I	MPF6	MII/RMII Receive Data error.
	EBI_AD13	O	MPF7	EBI address/data bus bit 13.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
134	PC.2	I/O	MFP0	General purpose digital I/O pin.
	I2S1_LRCK	O	MPF1	I2S1 left right channel clock.
	SC1_PWR	O	MPF2	SmartCard1 power pin.
	UART4_RTS	O	MPF3	Request to Send output pin for UART4.
	SPI0_SS0	I/O	MPF4	General purpose digital I/O pin.
	EMAC_MII_RXDV	I	MPF6	MII Receive Data Valid / RMII CRS_DV Input.
	EBI_AD12	O	MPF7	EBI address/data bus bit 12.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
135	PC.3	I/O	MFP0	General purpose digital I/O pin.

	I2S1_MCLK	O	MPF1	I2S1 master clock output pin.
	SC1_CD	I	MPF2	SmartCard1 card detect pin.
	UART4_CTS	I	MPF3	Clear to Send input pin for UART4.
	SPI0_MISO1	I/O	MPF4	2nd SPI0 MISO (Master In, Slave Out) pin.
	QEI0_Z	I	MPF5	Quadrature encoder phase Z input of QEI Unit 0.
	EMAC_MII_RXD1	I	MPF6	MII/RMII Receive Data Bus Bit 1.
	EBI_AD11	O	MPF7	EBI address/data bus bit 11.
	ECAP0_IC2	O	MPF8	Input 2 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
136	PC.4	I/O	MFP0	General purpose digital I/O pin.
	I2S1_DO	O	MPF1	I2S1 data output.
	SC1_RST	O	MPF2	SmartCard1 reset pin.
	SPI0_MOSI1	I/O	MPF4	2nd SPI0 MOSI (Master Out, Slave In) pin.
	QEI0_B	I	MPF5	Quadrature encoder phase B input of QEI Unit 0.
	EMAC_MII_RXD0	I	MPF6	MII/RMII Receive Data Bus Bit 0.
	EBI_AD10	O	MPF7	EBI address/data bus bit 10.
	ECAP0_IC1	O	MPF8	Input 1 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
137	PC.5	I/O	MFP0	General purpose digital I/O pin.
	CLKO	O	MFP1	Clock Output Pin.
	QEI0_A	I	MPF5	Quadrature encoder phase A input of QEI Unit 0.
	EMAC_MII_RXCLK	I	MPF6	MII Receive Clock Input.
	EBI_MCLK	O	MPF7	EBI interface clock output pin.
	ECAP0_IC0	O	MPF8	Input 0 of enhanced capture unit 0.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
138	PC.6	I/O	MFP0	General purpose digital I/O pin.

	TM2_EXT	I	MPF1	Timer2 external counter input
	SPI0_MISO0	I/O	MPF4	1st SPI0 MISO (Master In, Slave Out) pin.
	TM2_CNT_OUT	I/O	MPF5	Timer2 event counter input/toggle output.
	EMAC_MII_TXD0	O	MPF6	MII/RMII Transmit Data Bus bit 0.
	EBI_AD9	O	MPF7	EBI address/data bus bit 9.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
139	PC.7	I/O	MFP0	General purpose digital I/O pin.
	TM1_EXT	I	MPF1	Timer1 external counter input
	SPI0_MOSI0	I/O	MPF4	1st SPI0 MOSI (Master Out, Slave In) pin.
	EMAC_MII_TXD1	O	MPF6	MII/RMII Transmit Data Bus bit 1.
	EBI_AD8	O	MPF7	EBI address/data bus bit 8.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
140	PC.8	I/O	MFP0	General purpose digital I/O pin.
	TM0_EXT	I	MPF1	Timer0 external counter input
	SPI0_CLK	O	MPF4	SPI0 serial clock pin.
	EMAC_MII_TXEN	O	MPF6	MII/RMII Transmit Enable.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
141	PF.2	I/O	MFP0	General purpose digital I/O pin.
	SPI3_SS0	I/O	MPF1	General purpose digital I/O pin.
	SD0_DAT3	I/O	MPF4	SD mode #0 data line bit 3.
	EMAC_MII_RXD3	I	MPF6	MII Receive Data Bus Bit 3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
142	PF.3	I/O	MFP0	General purpose digital I/O pin.
	SPI3_CLK	O	MPF1	SPI3 serial clock pin.
	SD0_DAT2	I/O	MPF4	SD mode #0 data line bit 2.

	EMAC_MII_RXD2	I	MPF6	MII Receive Data Bus Bit 2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
143	PF.4	I/O	MFP0	General purpose digital I/O pin.
	SPI3_MISO0	I/O	MPF1	1st SPI3 MISO (Master In, Slave Out) pin.
	SD0_DAT1	I/O	MPF4	SD mode #0 data line bit 1.
	EMAC_MII_COL0	I	MPF6	MII Collision Detect Input Pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
144	PF.5	I/O	MFP0	General purpose digital I/O pin.
	SPI3_MOSI0	I/O	MPF1	1st SPI3 MOSI (Master Out, Slave In) pin.
	SD0_DAT0	I/O	MPF4	SD mode #0 data line bit 0.
	EMAC_MII_CRS	I	MPF6	MII Carrier Sense Input Pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
145	VSS	P	MFP0	Ground pin for digital circuit.
146	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
147	PF.6	I/O	MFP0	General purpose digital I/O pin.
	UART2_RXD	I	MPF1	Data receiver input pin for UART2.
	SD0_CDn	I	MPF4	SD mode #0 – card detect
	EMAC_MII_TXCLK	I	MPF6	MII Transmit Output Clock Pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
148	PF.7	I/O	MFP0	General purpose digital I/O pin.
	UART2_TXD	O	MPF1	Data transmitter output pin for UART2.
	SD0_CMD	I/O	MPF4	SD mode #0 – command/response
	EMAC_MII_TXD3	O	MPF6	MII Transmit Data Bus bit 3.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
149	PF.8	I/O	MFP0	General purpose digital I/O pin.

	UART2_RTS	O	MPF1	Request to Send output pin for UART2.
	SD0_CLK	O	MPF4	SD mode #0 – clock.
	EMAC_MII_TXD2	O	MPF6	MII Transmit Data Bus bit 2.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
150	PH.2	I/O	MFP0	General purpose digital I/O pin.
	UART2_CTS	I	MPF1	Clear to Send input pin for UART2.
151	PH.3	I/O	MFP0	General purpose digital I/O pin.
	I2C3_SCL	I/O	MPF1	I2C3 clock pin.
152	PH.4	I/O	MFP0	General purpose digital I/O pin.
	I2C3_SDA	I/O	MPF1	I2C3 data input/output pin.
153	PH.5	I/O	MFP0	General purpose digital I/O pin.
	SPI2_SS0	I/O	MPF1	General purpose digital I/O pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
154	PH.6	I/O	MFP0	General purpose digital I/O pin.
	SPI2_CLK	O	MPF1	SPI2 serial clock pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
155	PH.7	I/O	MFP0	General purpose digital I/O pin.
	SPI2_MISO0	I/O	MPF1	1st SPI2 MISO (Master In, Slave Out) pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
156	PH.8	I/O	MFP0	General purpose digital I/O pin.
	SPI2_MOSI0	I/O	MPF1	1st SPI2 MOSI (Master Out, Slave In) pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
157	PH.9	I/O	MFP0	General purpose digital I/O pin.
	SPI2_MISO1	I/O	MPF1	2nd SPI2 MISO (Master In, Slave Out) pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

158	PH.10	I/O	MFP0	General purpose digital I/O pin.
	SPI2_MOSI1	I/O	MPF1	2nd SPI2 MOSI (Master Out, Slave In) pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
159	LDO_CAP	P	MFP0	LDO output pin.
160	VSS	P	MFP0	Ground pin for digital circuit.
161	VDD	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
162	PE.0	I/O	MFP0	General purpose digital I/O pin.
	ADC0_0	A	MPF1	ADC0 analog input.
	INT4	I	MPF8	External interrupt4 input pin.
163	PE.1	I/O	MFP0	General purpose digital I/O pin.
	ADC0_1	A	MPF1	ADC0 analog input.
	TM2_CNT_OUT	I/O	MPF3	Timer2 event counter input/toggle output.
164	PE.2	I/O	MFP0	General purpose digital I/O pin.
	ADC0_2	A	MPF1	ADC0 analog input.
	ACMP0_O	O	MPF2	Analog ccomparator0 output .
	SPI0_MISO0	I/O	MPF3	1st SPI0 MISO (Master In, Slave Out) pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
165	PE.3	I/O	MFP0	General purpose digital I/O pin.
	ADC0_3	A	MPF1	ADC0 analog input.
	ACMP0_P3	A	MPF2	Analog comparator0 positive input pin.
	SPI0_MOSI0	I/O	MPF3	1st SPI0 MOSI (Master Out, Slave In) pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
166	PE.4	I/O	MFP0	General purpose digital I/O pin.
	ADC0_4	A	MPF1	ADC0 analog input.
	ACMP0_P2	A	MPF2	Analog comparator0 positive input pin.

	SPI0_SS0	I/O	MPF3	General purpose digital I/O pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
167	PE.5	I/O	MFP0	General purpose digital I/O pin.
	ADC0_5	A	MPF1	ADC0 analog input.
	ACMP0_P1	A	MPF2	Analog comparator0 positive input pin.
	SPI0_CLK	O	MPF3	SPI0 serial clock pin.
	SD0_CDn	I	MPF4	SD mode #0 – card detect
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
168	PE.6	I/O	MFP0	General purpose digital I/O pin.
	ADC0_6	A	MPF1	ADC0 analog input.
	ACMP0_P0	A	MPF2	Analog comparator0 positive input pin.
	SPI0_MISO0	I/O	MPF3	1st SPI0 MISO (Master In, Slave Out) pin.
	SD0_CMD	I/O	MPF4	SD mode #0 – command/response
	EBI_nWR	O	MPF7	EBI write enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
169	PE.7	I/O	MFP0	General purpose digital I/O pin.
	ADC0_7	A	MPF1	ADC0 analog input.
	ACMP0_N	A	MPF2	Analog comparator0 negative input pin.
	SPI0_MOSI0	I/O	MPF3	1st SPI0 MOSI (Master Out, Slave In) pin.
	SD0_CLK	O	MPF4	SD mode #0 – clock.
	EBI_nRD	O	MPF7	EBI read enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
170	AVSS	P	MFP0	Ground pin for digital circuit.
171	Vref	A	MFP0	Voltage reference input for ADC.
172	AVDD	P	MFP0	Power supply for internal analog circuit.

173	PE.8	I/O	MFP0	General purpose digital I/O pin.
	ADC1_0	A	MPF1	ADC1 analog input.
	ADC0_8	A	MPF1	ADC0 analog input.
	ACMP1_N	A	MPF2	Analog comparator1 negative input pin.
	TM1_CNT_OUT	I/O	MPF3	Timer1 event counter input/toggle output.
	SD0_DAT3	I/O	MPF4	SD mode #0 data line bit 3.
	EBI_ALE	O	MPF7	EBI address latch enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
174	PE.9	I/O	MFP0	General purpose digital I/O pin.
	ADC1_1	A	MPF1	ADC1 analog input.
	ADC0_9	A	MPF1	ADC0 analog input.
	ACMP1_P0	A	MPF2	Analog comparator1 positive input pin.
	SD0_DAT2	I/O	MPF4	SD mode #0 data line bit 2.
	EBI_nWRH	O	MPF7	EBI write enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
175	PE.10	I/O	MFP0	General purpose digital I/O pin.
	ADC1_2	A	MPF1	ADC1 analog input.
	ADC0_10	A	MPF1	ADC0 analog input.
	ACMP1_P1	A	MPF2	Analog comparator1 positive input pin.
	SPI0_MISO1	I/O	MPF3	2nd SPI0 MISO (Master In, Slave Out) pin.
	SD0_DAT1	I/O	MPF4	SD mode #0 data line bit 1.
	EBI_nWRL	O	MPF7	EBI write enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.
176	PE.11	I/O	MFP0	General purpose digital I/O pin.
	ADC1_3	A	MPF1	ADC1 analog input.
	ADC0_11	A	MPF1	ADC0 analog input.

	ACMP1_P2	A	MPF2	Analog comparator1 positive input pin.
	SPI0_MOSI1	I/O	MPF3	2nd SPI0 MOSI (Master Out, Slave In) pin.
	SD0_DAT0	I/O	MPF4	SD mode #0 data line bit 0.
	ACMP2_P3	A	MPF5	Analog comparator2 positive input pin.
	EBI_nCS0	O	MPF7	EBI chip select 0 enable output pin.
	HS		Slew	This pad is embedded with “Slew Rate Control” capability.

Note: Pin Type I = Digital Input, O = Digital Output; A = Analog Pin; P = Power Pin;

4.3.9 Summary GPIO Multi-function Pin Description

MFP* = Multi-function pin. (Reference section 6.2.5)

MFP 0	MPF1	MPF2	MPF3	MPF4	MPF5	MPF6	MPF7	MPF8	MPF9	Other	Slew
PA.0	TAMPER0	SC0_CD	CAN1_RXD					INT0			
PA.1	TAMPER1	SC5_CD	CAN1_TXD				EBI_A22				
PA.2	SC2_DAT	SPI3_MISO0	I2S0_MCLK	BRAKE11	CAP_SFIELD		EBI_A12				HS
PA.3	SC2_CLK	SPI3_MOSI0	I2S0_DO	BRAKE10			EBI_A13				HS
PA.4	SC2_PWR	SPI3_CLK	I2S0_DI		QEI1_Z		EBI_A14	ECAP1_IC2			HS
PA.5	SC2_RST	SPI3_SS0	I2S0_BCLK	PWM0_0	QEI1_B		EBI_A15	ECAP1_IC1			HS
PA.6	SC2_CD		I2S0_LRCK	PWM0_1	QEI1_A	CAN1_TXD	EBI_A16	ECAP1_IC0			HS
PA.7		SC0_CLK	SPI3_SS0	PWM1_3	EPWM0_5		EBI_A17				HS
PA.8		SC0_RST	SPI3_CLK	PWM1_2	EPWM0_4		EBI_A18				HS
PA.9		SC0_PWR	SPI3_MISO0	PWM1_1	EPWM0_3		EBI_A19				HS
PA.10		SC0_DAT	SPI3_MOSI0	PWM1_0	EPWM0_2		EBI_A20				HS
PA.11	UART0_RTS		SPI3_MISO1	PWM0_5	EPWM0_1		EBI_AD0				HS
PA.12	UART0_CTS		SPI3_MOSI1	PWM0_4	EPWM0_0		EBI_AD1				HS
PA.13	UART0_RXD		SC3_DAT	PWM1_4			EBI_AD2				HS
PA.14	UART0_TXD		SC3_CLK	PWM1_5			EBI_AD3				HS
PA.15	SC3_PWR	UART2_RTS		I2C0_SCL			EBI_A21				HS
PB.0	USB0_VBUS_ST	I2C4_SCL						INT1			
PB.1	USB0_VBUS_EN	I2C4_SDA	TM1_CNT_OUT								
PB.2	UART1_RXD	SPI2_SS0	USB1_D-				EBI_AD4				HS
PB.3	UART1_TXD	SPI2_CLK	USB1_D+				EBI_AD5				HS
PB.4	UART1_RTS	SPI2_MISO0	UART4_RXD	TM0_CNT_OUT			EBI_AD6				HS
PB.5	UART1_CTS	SPI2_MOSI0	UART4_TXD				EBI_AD7				HS

PB.6	I2C2_SCL	BRAKE01	UART4_RTS	PWM1_4	EPWM1_0		EBI_AD8				HS
PB.7	I2C2_SDA	BRAKE00	UART4_CTS	PWM1_5	EPWM1_1		EBI_AD9				HS
PB.8	UART5_CTS				EPWM1_2		EBI_AD10				HS
PB.9	UART5_RTS				EPWM1_3		EBI_AD11				HS
PB.10	UART5_TXD				EPWM1_4		EBI_AD12				HS
PB.11	UART5_RXD				EPWM1_5		EBI_AD13				HS
PB.12	UART4_RTS	SPI2_MISO1	CAN0_RXD			EMAC_MII_MDC	EBI_AD14				HS
PB.13	UART4_CTS	SPI2_MOSI1	CAN0_TXD			EMAC_MII_MDIO	EBI_AD15				HS
PB.14	I2S1_MCLK	SC1_RST		BRAKE01		EMAC_MII_MDC					HS
PB.15	I2S1_DO	SC1_DAT		BRAKE00		EMAC_MII_MDIO					HS
PC.0	I2S1_DI	SC1_DAT	UART4_RXD			EMAC_REFCLK	EBI_MCLK	INT2			HS
PC.1	I2S1_BCLK	SC1_CLK	UART4_TXD		TM3_CNT_OUT	EMAC_MII_RXERR	EBI_AD13				HS
PC.2	I2S1_LRCK	SC1_PWR	UART4_RTS	SPI0_SS0		EMAC_MII_RXDV	EBI_AD12				HS
PC.3	I2S1_MCLK	SC1_CD	UART4_CTS	SPI0_MISO1	QEI0_Z	EMAC_MII_RXD1	EBI_AD11	ECAP0_IC2			HS
PC.4	I2S1_DO	SC1_RST		SPI0_MOSI1	QEI0_B	EMAC_MII_RXD0	EBI_AD10	ECAP0_IC1			HS
PC.5	CLKO				QEI0_A	EMAC_MII_RXCLK	EBI_MCLK	ECAP0_IC0			HS
PC.6	TM2_EXT			SPI0_MISO0	TM2_CNT_OUT	EMAC_MII_TXD0	EBI_AD9				HS
PC.7	TM1_EXT			SPI0_MOSI0		EMAC_MII_TXD1	EBI_AD8				HS
PC.8	TM0_EXT			SPI0_CLK		EMAC_MII_TXEN					HS
PC.9	STADC	UART2_CTS	SC3_RST	I2C0_SDA	CAP_DATA1	I2C3_SCL	EBI_A22	SD1_DAT0	EBI_A6		HS
PC.10	SC3_CD	UART2_RXD		PWM0_2		EBI_A23	EBI_AD2				HS
PC.11		UART2_TXD		PWM0_3		EBI_A24	EBI_AD3				HS
PC.12	SPI1_SS0	SC4_CD		SD1_CDn	CAP_DATA7		EBI_A0				HS

PC.13	SPI1_MOSI1	SC4_RST		SD1_CMD	CAP_DATA6		EBI_A1				HS
PC.14	SPI1_MISO1	SC4_PWR	TM3_EXT	SD1_CLK	CAP_DATA5		EBI_A2				HS
PC.15	SPI1_MOSI0	SC4_DAT		SD1_DAT3	CAP_DATA4		EBI_A3				HS
PD.0	SPI1_MISO0	SC4_CLK		SD1_DAT2	CAP_DATA3		EBI_A4	INT3			HS
PD.1	SPI1_CLK		TM0_CNT_OUT	SD1_DAT1	CAP_DATA2		EBI_A5				HS
PD.2	STADC	I2C3_SCL		SD1_DAT0	CAP_DATA1		EBI_A6				HS
PD.3	SC5_CLK	I2C3_SDA	ACMP2_O	SD0_CDn	CAP_DATA0		EBI_A7				HS
PD.4	SC5_CD	UART3_RXD	ACMP1_O		CAP_SCLK		EBI_A8				HS
PD.5	SC5_RST	UART3_TXD			CAP_VSYNC		EBI_A9				HS
PD.6	SC5_PWR	UART3_RTS		SD0_CMD	CAP_HSYNC		EBI_A10				HS
PD.7	SC5_DAT	UART3_CTS		SD0_CLK	CAP_PIXCLK		EBI_A11				HS
PD.8	SPI3_MISO1	I2C0_SCL									HS
PD.9	SPI3_MOSI1	I2C0_SDA									HS
PD.10	SC3_DAT	I2C4_SCL									
PD.11	SC3_RST		TM3_CNT_OUT								
PD.12	SC3_CLK	I2C4_SDA									
PD.13	SPI1_SS0	UART5_CTS	ECAP0_IC2								HS
PD.14	SPI1_CLK	UART5_RTS	ECAP0_IC1								HS
PD.15	SPI1_MISO0	UART5_TXD	ECAP0_IC0								HS
PE.0	ADC0_0							INT4			
PE.1	ADC0_1		TM2_CNT_OUT								
PE.2	ADC0_2	ACMP0_O	SPI0_MISO0								HS
PE.3	ADC0_3	ACMP0_P3	SPI0_MOSI0								HS
PE.4	ADC0_4	ACMP0_P2	SPI0_SS0								HS

PE.5	ADC0_5	ACMP0_P1	SPI0_CLK	SD0_CDn							HS
PE.6	ADC0_6	ACMP0_P0	SPI0_MISO0	SD0_CMD			EBI_nWR				HS
PE.7	ADC0_7	ACMP0_N	SPI0_MOSI0	SD0_CLK			EBI_nRD				HS
PE.8	ADC1_0/ ADC0_8	ACMP1_N	TM1_CNT_OUT	SD0_DAT3			EBI_ALE				HS
PE.9	ADC1_1/ADC0_9	ACMP1_P0		SD0_DAT2			EBI_nWRH				HS
PE.10	ADC1_2/ADC0_10	ACMP1_P1	SPI0_MISO1	SD0_DAT1			EBI_nWRL				HS
PE.11	ADC1_3/ADC0_11	ACMP1_P2	SPI0_MOSI1	SD0_DAT0	ACMP2_P3		EBI_nCS0				HS
PE.12	ADC1_4	ACMP1_P3	ACMP2_P2				EBI_nCS1				HS
PE.13	ADC1_5		ACMP2_P1				EBI_nCS2				HS
PE.14	ADC1_6		ACMP2_P0				EBI_nCS3				HS
PE.15	ADC1_7		ACMP2_N								
PF.0	SPI1_MOSI0	UART5_RXD						INT5			HS
PF.1	SPI2_MOSI1										HS
PF.2	SPI3_SS0			SD0_DAT3		EMAC_MII_RXD3					HS
PF.3	SPI3_CLK			SD0_DAT2		EMAC_MII_RXD2					HS
PF.4	SPI3_MISO0			SD0_DAT1		EMAC_MII_COL0					HS
PF.5	SPI3_MOSI0			SD0_DAT0		EMAC_MII_CRS					HS
PF.6	UART2_RXD			SD0_CDn		EMAC_MII_TXCLK					HS
PF.7	UART2_TXD			SD0_CMD		EMAC_MII_TXD3					HS
PF.8	UART2_RTS			SD0_CLK		EMAC_MII_TXD2					HS
PF.9	OPA0_IN+			PWM0_0							HS
PF.10	OPA0_IN-			PWM0_1							HS
PF.11	OPA0_O	UART1_RTS									

PF.12	OPA1_IN+	UART1_CTS									
PF.13	OPA1_IN-	UART1_TXD									
PF.14	OPA1_O	UART1_RXD									
PF.15	UART0_RTS										
PG.0	UART0_CTS							INT6			
PG.1	UART0_RXD										
PG.2	UART0_TXD										
PG.3	PS2_CLK	I2S1_DO	SC1_RST								
PG.4	PS2_DAT	I2S1_DI	SC1_PWR								
PG.5		I2S1_BCLK	SC1_DAT								
PG.6		I2S1_LRCK	SC1_CLK								
PG.7	SPI2_MISO0	I2S1_MCLK	SC1_CD	SC3_RST							HS
PG.8	SPI2_MOSI0	I2S1_DO	UART4_RTS	SC3_DAT							HS
PG.9	SPI2_CLK	I2S1_DI	UART4_CTS	SC3_CLK							HS
PG.10	ICE_CLK										
PG.11	ICE_DAT										
PG.12	XT1_OUT										
PG.13	XT1_IN										
PG.14	X32K_OUT		I2C1_SDA								
PG.15	X32K_IN		I2C1_SCL								
PH.0	I2C1_SCL	UART4_RXD	CAN1_RXD					INT7			
PH.1	UART4_TXD	I2C1_SDA	CAN1_TXD								
PH.2	UART2_CTS										
PH.3	I2C3_SCL										
PH.4	I2C3_SDA										
PH.5	SPI2_SS0										HS
PH.6	SPI2_CLK										HS

PH.7	SPI2_MISO0										HS
PH.8	SPI2_MOSI0										HS
PH.9	SPI2_MISO1										HS
PH.10	SPI2_MOSI1										HS
PH.11	UART3_RXD										
PH.12	UART3_TXD										
PH.13	UART3_RTS										
PH.14	UART3_CTS										
PH.15		SC5_CLK									
PI.0		SC5_RST									
PI.1		SC5_PWR									
PI.2	SC5_DAT										
PI.3	SPI3_SS0										HS
PI.4	SPI3_CLK										HS
PI.5	SPI3_MISO0										HS
PI.6	SPI3_MOSI0										HS
PI.7	I2C2_SCL	SPI3_MISO1									HS
PI.8	I2C2_SDA	SPI3_MOSI1									HS
PI.9				I2C4_SCL							
PI.10											
PI.11	SPI2_SS0	I2S1_BCLK	I2C4_SCL	SC3_PWR							HS
PI.12	SPI2_MISO1	I2S1_LRCK	I2C4_SDA	SC3_CD							
PI.13											
PI.14											
PI.15											

Note: PA.0* = TAMPER0 PA.1* = TAMPER1

4.3.10 Summary Function Pin Description

Group	Pin Name	GPIO	*MFP	Type	Description
ACMP0	ACMP0_N	PE.7	MPF2	A	Analog comparator0 negative input pin.
ACMP0	ACMP0_O	PE.2	MPF2	O	Analog ccomparator0 output .
ACMP0	ACMP0_P0	PE.6	MPF2	A	Analog comparator0 positive input pin.
ACMP0	ACMP0_P1	PE.5	MPF2	A	Analog comparator0 positive input pin.
ACMP0	ACMP0_P2	PE.4	MPF2	A	Analog comparator0 positive input pin.
ACMP0	ACMP0_P3	PE.3	MPF2	A	Analog comparator0 positive input pin.
ACMP1	ACMP1_N	PE.8	MPF2	A	Analog comparator1 negative input pin.
ACMP1	ACMP1_O	PD.4	MPF3	O	Analog ccomparator1 output .
ACMP1	ACMP1_P0	PE.9	MPF2	A	Analog comparator1 positive input pin.
ACMP1	ACMP1_P1	PE.10	MPF2	A	Analog comparator1 positive input pin.
ACMP1	ACMP1_P2	PE.11	MPF2	A	Analog comparator1 positive input pin.
ACMP1	ACMP1_P3	PE.12	MPF2	A	Analog comparator1 positive input pin.
ACMP2	ACMP2_N	PE.15	MPF3	A	Analog comparator2 negative input pin.
ACMP2	ACMP2_O	PD.3	MPF3	O	Analog ccomparator2 output .
ACMP2	ACMP2_P0	PE.14	MPF3	A	Analog comparator2 positive input pin.
ACMP2	ACMP2_P1	PE.13	MPF3	A	Analog comparator2 positive input pin.
ACMP2	ACMP2_P2	PE.12	MPF3	A	Analog comparator2 positive input pin.
ACMP2	ACMP2_P3	PE.11	MPF5	A	Analog comparator2 positive input pin.
ADC0	ADC0_0	PE.0	MPF1	A	ADC0 analog input.
ADC0	ADC0_1	PE.1	MPF1	A	ADC0 analog input.
ADC0	ADC0_10	PE.10	MPF1	A	ADC0 analog input.
ADC0	ADC0_11	PE.11	MPF1	A	ADC0 analog input.
ADC0	ADC0_2	PE.2	MPF1	A	ADC0 analog input.
ADC0	ADC0_3	PE.3	MPF1	A	ADC0 analog input.
ADC0	ADC0_4	PE.4	MPF1	A	ADC0 analog input.
ADC0	ADC0_5	PE.5	MPF1	A	ADC0 analog input.
ADC0	ADC0_6	PE.6	MPF1	A	ADC0 analog input.
ADC0	ADC0_7	PE.7	MPF1	A	ADC0 analog input.
ADC0	ADC0_8	PE.8	MPF1	A	ADC0 analog input.
ADC0	ADC0_9	PE.9	MPF1	A	ADC0 analog input.
ADC0	STADC	PD.2	MPF1	A	ADC analog input.
ADC1	ADC1_0	PE.8	MPF1	A	ADC1 analog input.

ADC1	ADC1_1	PE.9	MPF1	A	ADC1 analog input.
ADC1	ADC1_2	PE.10	MPF1	A	ADC1 analog input.
ADC1	ADC1_3	PE.11	MPF1	A	ADC1 analog input.
ADC1	ADC1_4	PE.12	MPF1	A	ADC1 analog input.
ADC1	ADC1_5	PE.13	MPF1	A	ADC1 analog input.
ADC1	ADC1_6	PE.14	MPF1	A	ADC1 analog input.
ADC1	ADC1_7	PE.15	MPF1	A	ADC1 analog input.
BPWMA	PWM0_0	PA.5	MPF4	I/O	PWM0_0 output/capture input.
BPWMA	PWM0_0	PF.9	MPF4	I/O	PWM0_0 output/capture input.
BPWMA	PWM0_1	PA.6	MPF4	I/O	PWM0_1 output/capture input.
BPWMA	PWM0_1	PF.10	MPF4	I/O	PWM0_1 output/capture input.
BPWMA	PWM0_2	PC.10	MPF4	I/O	PWM0_2 output/capture input.
BPWMA	PWM0_3	PC.11	MPF4	I/O	PWM0_3 output/capture input.
BPWMA	PWM0_4	PA.12	MPF4	I/O	PWM0_4 output/capture input.
BPWMA	PWM0_5	PA.11	MPF4	I/O	PWM0_5 output/capture input.
BPWMB	PWM1_0	PA.10	MPF4	I/O	PWM1_0 output/capture input.
BPWMB	PWM1_1	PA.9	MPF4	I/O	PWM1_1 output/capture input.
BPWMB	PWM1_2	PA.8	MPF4	I/O	PWM1_2 output/capture input.
BPWMB	PWM1_3	PA.7	MPF4	I/O	PWM1_3 output/capture input.
BPWMB	PWM1_4	PA.13	MPF4	I/O	PWM1_4 output/capture input.
BPWMB	PWM1_4	PB.6	MPF4	I/O	PWM1_4 output/capture input.
BPWMB	PWM1_5	PA.14	MPF4	I/O	PWM1_5 output/capture input.
BPWMB	PWM1_5	PB.7	MPF4	I/O	PWM1_5 output/capture input.
BRAKE	BRAKE00	PB.7	MPF2	I	Brake input pin 0 of EPWMB.
BRAKE	BRAKE00	PB.15	MPF4	I	Brake input pin 0 of EPWMB.
BRAKE	BRAKE01	PB.6	MPF2	I	Brake input pin 1 of EPWMB.
BRAKE	BRAKE01	PB.14	MPF4	I	Brake input pin 1 of EPWMB.
BRAKE	BRAKE10	PA.3	MPF4	I	Brake input pin 0 of EPWMA.
BRAKE	BRAKE11	PA.2	MPF4	I	Brake input pin 1 of EPWMA.
CAN0	CAN0_RXD	PB.12	MPF3	I	CAN bus receiver0 input.
CAN0	CAN0_TXD	PB.13	MPF3	I	CAN bus transmitter0 input.
CAN1	CAN1_RXD	PA.0	MPF3	I	CAN bus receiver1 input.
CAN1	CAN1_RXD	PH.0	MPF3	I	CAN bus receiver1 input.
CAN1	CAN1_TXD	PA.1	MPF3	I	CAN bus transmitter1 input.
CAN1	CAN1_TXD	PA.6	MPF6	I	CAN bus transmitter1 input.

CAN1	CAN1_TXD	PH.1	MPF3	I	CAN bus transmitter1 input.
CLKO	CLKO	PC.5	MPF1	O	Clock Output Pin.
EBI	EBI_A0	PC.12	MPF7	O	EBI address bus bit0.
EBI	EBI_A1	PC.13	MPF7	O	EBI address bus bit1.
EBI	EBI_A10	PD.6	MPF7	O	EBI address bus bit10.
EBI	EBI_A11	PD.7	MPF7	O	EBI address bus bit11.
EBI	EBI_A12	PA.2	MPF7	O	EBI address bus bit12.
EBI	EBI_A13	PA.3	MPF7	O	EBI address bus bit13.
EBI	EBI_A14	PA.4	MPF7	O	EBI address bus bit14.
EBI	EBI_A15	PA.5	MPF7	O	EBI address bus bit15.
EBI	EBI_A16	PA.6	MPF7	O	EBI address bus bit16.
EBI	EBI_A17	PA.7	MPF7	O	EBI address bus bit17.
EBI	EBI_A18	PA.8	MPF7	O	EBI address bus bit18.
EBI	EBI_A19	PA.9	MPF7	O	EBI address bus bit19.
EBI	EBI_A2	PC.14	MPF7	O	EBI address bus bit2.
EBI	EBI_A20	PA.10	MPF7	O	EBI address bus bit20.
EBI	EBI_A21	PA.15	MPF7	O	EBI address bus bit21.
EBI	EBI_A22	PC.9	MPF7	O	EBI address bus bit22.
EBI	EBI_A23	PC.10	MPF6	O	EBI address bus bit23.
EBI	EBI_A24	PC.11	MPF6	O	EBI address bus bit24.
EBI	EBI_A3	PC.15	MPF7	O	EBI address bus bit3.
EBI	EBI_A4	PD.0	MPF7	O	EBI address bus bit4.
EBI	EBI_A5	PD.1	MPF7	O	EBI address bus bit5.
EBI	EBI_A6	PD.2	MPF7	O	EBI address bus bit6.
EBI	EBI_A7	PD.3	MPF7	O	EBI address bus bit7.
EBI	EBI_A8	PD.4	MPF7	O	EBI address bus bit8.
EBI	EBI_A9	PD.5	MPF7	O	EBI address bus bit9.
EBI	EBI_AD0	PA.11	MPF7	O	EBI address/data bus bit 0.
EBI	EBI_AD1	PA.12	MPF7	O	EBI address/data bus bit 1.
EBI	EBI_AD10	PB.8	MPF7	O	EBI address/data bus bit 10.
EBI	EBI_AD10	PC.4	MPF7	O	EBI address/data bus bit 10.
EBI	EBI_AD11	PB.9	MPF7	O	EBI address/data bus bit 11.
EBI	EBI_AD11	PC.3	MPF7	O	EBI address/data bus bit 11.
EBI	EBI_AD12	PB.10	MPF7	O	EBI address/data bus bit 12.
EBI	EBI_AD12	PC.2	MPF7	O	EBI address/data bus bit 12.

EBI	EBI_AD13	PB.11	MPF7	O	EBI address/data bus bit 13.
EBI	EBI_AD13	PC.1	MPF7	O	EBI address/data bus bit 13.
EBI	EBI_AD14	PB.12	MPF7	O	EBI address/data bus bit 14.
EBI	EBI_AD15	PB.13	MPF7	O	EBI address/data bus bit 15.
EBI	EBI_AD2	PA.13	MPF7	O	EBI address/data bus bit 2.
EBI	EBI_AD3	PA.14	MPF7	O	EBI address/data bus bit 3.
EBI	EBI_AD4	PB.2	MPF7	O	EBI address/data bus bit 4.
EBI	EBI_AD5	PB.5	MPF7	O	EBI address/data bus bit 5.
EBI	EBI_AD6	PB.4	MPF7	O	EBI address/data bus bit 6.
EBI	EBI_AD7	PB.5	MPF7	O	EBI address/data bus bit 7.
EBI	EBI_AD8	PB.6	MPF7	O	EBI address/data bus bit 8.
EBI	EBI_AD8	PC.7	MPF7	O	EBI address/data bus bit 8.
EBI	EBI_AD9	PB.7	MPF7	O	EBI address/data bus bit 9.
EBI	EBI_AD9	PC.6	MPF7	O	EBI address/data bus bit 9.
EBI	EBI_ALE	PE.8	MPF7	O	EBI address latch enable output pin.
EBI	EBI_MCLK	PC.0	MPF7	O	EBI interface clock output pin.
EBI	EBI_nCS0	PE.11	MPF7	O	EBI chip select 0 enable output pin.
EBI	EBI_nCS1	PE.12	MPF7	O	EBI chip select 1 enable output pin.
EBI	EBI_nCS2	PE.13	MPF7	O	EBI chip select 2 enable output pin.
EBI	EBI_nCS3	PE.14	MPF7	O	EBI chip select 3 enable output pin.
EBI	EBI_nRD	PE.7	MPF7	O	EBI read enable output pin.
EBI	EBI_nWR	PE.6	MPF7	O	EBI write enable output pin.
EBI	EBI_nWRH	PE.9	MPF7	O	EBI write enable output pin.
EBI	EBI_nWRL	PE.10	MPF7	O	EBI write enable output pin.
EBI_AD	EBI_A16	PA.6	MPF7	O	EBI address bus bit16.
EBI_AD	EBI_A17	PA.7	MPF7	O	EBI address bus bit17.
EBI_AD	EBI_A18	PA.8	MPF7	O	EBI address bus bit18.
EBI_AD	EBI_A19	PA.9	MPF7	O	EBI address bus bit19.
EBI_AD	EBI_A20	PA.10	MPF7	O	EBI address bus bit20.
EBI_AD	EBI_A21	PA.15	MPF7	O	EBI address bus bit21.
EBI_AD	EBI_A22	PC.9	MPF7	O	EBI address bus bit22.
EBI_AD	EBI_AD0	PA.11	MPF7	O	EBI address/data bus bit 0.
EBI_AD	EBI_AD1	PA.12	MPF7	O	EBI address/data bus bit 1.
EBI_AD	EBI_AD10	PB.8	MPF7	O	EBI address/data bus bit 1.
EBI_AD	EBI_AD10	PC.4	MPF7	O	EBI address/data bus bit 1.

EBI_AD	EBI_AD11	PB.9	MPF7	O	EBI address/data bus bit 1.
EBI_AD	EBI_AD11	PC.3	MPF7	O	EBI address/data bus bit 1.
EBI_AD	EBI_AD12	PB.10	MPF7	O	EBI address/data bus bit 1.
EBI_AD	EBI_AD12	PC.2	MPF7	O	EBI address/data bus bit 1.
EBI_AD	EBI_AD13	PB.11	MPF7	O	EBI address/data bus bit 1.
EBI_AD	EBI_AD13	PC.1	MPF7	O	EBI address/data bus bit 1.
EBI_AD	EBI_AD14	PB.12	MPF7	O	EBI address/data bus bit 1.
EBI_AD	EBI_AD15	PB.13	MPF7	O	EBI address/data bus bit 1.
EBI_AD	EBI_AD2	PA.13	MPF7	O	EBI address/data bus bit 2.
EBI_AD	EBI_AD2	PC.10	MPF7	O	EBI address/data bus bit 2.
EBI_AD	EBI_AD3	PA.14	MPF7	O	EBI address/data bus bit 3.
EBI_AD	EBI_AD3	PC.11	MPF7	O	EBI address/data bus bit 3.
EBI_AD	EBI_AD4	PB.2	MPF7	O	EBI address/data bus bit 4.
EBI_AD	EBI_AD5	PB.3	MPF7	O	EBI address/data bus bit 5.
EBI_AD	EBI_AD6	PB.4	MPF7	O	EBI address/data bus bit 6.
EBI_AD	EBI_AD7	PB.5	MPF7	O	EBI address/data bus bit 7.
EBI_AD	EBI_AD8	PB.6	MPF7	O	EBI address/data bus bit 8.
EBI_AD	EBI_AD8	PC.7	MPF7	O	EBI address/data bus bit 8.
EBI_AD	EBI_AD9	PB.7	MPF7	O	EBI address/data bus bit 9.
EBI_AD	EBI_AD9	PC.6	MPF7	O	EBI address/data bus bit 9.
EBI_AD	EBI_ALE	PE.8	MPF7	O	EBI address latch enable output pin.
EBI_AD	EBI_MCLK	PC.5	MPF7	O	EBI interface clock output pin.
EBI_AD	EBI_nCS0	PE.11	MPF7	O	EBI chip select 0 enable output pin.
EBI_AD	EBI_nCS1	PE.12	MPF7	O	EBI chip select 1 enable output pin.
EBI_AD	EBI_nCS2	PE.13	MPF7	O	EBI chip select 2 enable output pin.
EBI_AD	EBI_nCS3	PE.14	MPF7	O	EBI chip select 3 enable output pin.
EBI_AD	EBI_nRD	PE.7	MPF7	O	EBI read enable output pin.
EBI_AD	EBI_nWR	PE.6	MPF7	O	EBI write enable output pin.
EBI_AD	EBI_nWRH	PE.9	MPF7	O	EBI write enable output pin.
EBI_AD	EBI_nWRL	PE.10	MPF7	O	EBI write enable output pin.
ECAP0	ECAP0_IC0	PC.5	MPF8	I	Input 0 of enhanced capture unit 0.
ECAP0	ECAP0_IC0	PD.15	MPF3	I	Input 0 of enhanced capture unit 0.
ECAP0	ECAP0_IC1	PC.4	MPF8	I	Input 1 of enhanced capture unit 0.
ECAP0	ECAP0_IC1	PD.14	MPF3	I	Input 1 of enhanced capture unit 0.
ECAP0	ECAP0_IC2	PC.3	MPF8	I	Input 2 of enhanced capture unit 0.

ECAP0	ECAP0_IC2	PD.13	MPF3	I	Input 2 of enhanced capture unit 0.
ECAP1	ECAP1_IC0	PA.6	MPF8	I	Input 0 of enhanced capture unit 1.
ECAP1	ECAP1_IC1	PA.5	MPF8	I	Input 1 of enhanced capture unit 1.
ECAP1	ECAP1_IC2	PA.4	MPF8	I	Input 2 of enhanced capture unit 1.
EMAC_MII	EMAC_MII_COL0	PF.4	MPF6	I	MII Collision Detect Input Pin.
EMAC_MII	EMAC_MII_CRS	PF.5	MPF6	I	MII Carrier Sense Input Pin.
EMAC_MII	EMAC_MII_MDC	PB.12	MPF6	O	MII/RMII Management Data Clock.
EMAC_MII	EMAC_MII_MDC	PB.14	MPF6	O	MII/RMII Management Data Clock.
EMAC_MII	EMAC_MII_MDIO	PB.13	MPF6	I/O	MII/RMII Management Data I/O.
EMAC_MII	EMAC_MII_MDIO	PB.15	MPF6	I/O	MII/RMII Management Data I/O.
EMAC_MII	EMAC_MII_RXCLK	PC.5	MPF6	I	MII Receive Clock Input.
EMAC_MII	EMAC_MII_RXD0	PC.4	MPF6	I	MII/RMII Receive Data Bus Bit 0.
EMAC_MII	EMAC_MII_RXD1	PC.3	MPF6	I	MII/RMII Receive Data Bus Bit 1.
EMAC_MII	EMAC_MII_RXD2	PF.3	MPF6	I	MII Receive Data Bus Bit 2.
EMAC_MII	EMAC_MII_RXD3	PF.2	MPF6	I	MII Receive Data Bus Bit 3.
EMAC_MII	EMAC_MII_RXDV	PC.2	MPF6	I	MII Receive Data Valid / RMII CRS_DV Input.
EMAC_MII	EMAC_MII_RXERR	PC.1	MPF6	I	MII/RMII Receive Data error.
EMAC_MII	EMAC_MII_TXCLK	PF.6	MPF6	I	MII Transmit Output Clock Pin.
EMAC_MII	EMAC_MII_TXD0	PC.6	MPF6	O	MII/RMII Transmit Data Bus bit 0.
EMAC_MII	EMAC_MII_TXD1	PC.7	MPF6	O	MII/RMII Transmit Data Bus bit 1.
EMAC_MII	EMAC_MII_TXD2	PF.8	MPF6	O	MII Transmit Data Bus bit 2.
EMAC_MII	EMAC_MII_TXD3	PF.7	MPF6	O	MII Transmit Data Bus bit 3.
EMAC_MII	EMAC_MII_TXEN	PC.8	MPF6	O	MII/RMII Transmit Enable.
EMAC_MII	EMAC_REFCLK	PC.0	MPF6	I	EMAC RMII mode clock input
EMAC_RMII	EMAC_MII_MDC	PB.12	MPF6	O	MII/RMII Management Data Clock.
EMAC_RMII	EMAC_MII_MDC	PB.14	MPF6	O	MII/RMII Management Data Clock.
EMAC_RMII	EMAC_MII_MDIO	PB.13	MPF6	I/O	MII/RMII Management Data I/O.
EMAC_RMII	EMAC_MII_MDIO	PB.15	MPF6	I/O	MII/RMII Management Data I/O.
EMAC_RMII	EMAC_MII_RXD0	PC.4	MPF6	I	MII/RMII Receive Data Bus Bit 0.
EMAC_RMII	EMAC_MII_RXD1	PC.3	MPF6	I	MII/RMII Receive Data Bus Bit 1.
EMAC_RMII	EMAC_MII_RXDV	PC.2	MPF6	I	MII Receive Data Valid / RMII CRS_DV Input.
EMAC_RMII	EMAC_MII_RXERR	PC.1	MPF6	I	MII/RMII Receive Data error.
EMAC_RMII	EMAC_MII_TXD0	PC.6	MPF6	O	MII/RMII Transmit Data Bus bit 0.
EMAC_RMII	EMAC_MII_TXD1	PC.7	MPF6	O	MII/RMII Transmit Data Bus bit 1.
EMAC_RMII	EMAC_MII_TXEN	PC.8	MPF6	O	MII/RMII Transmit Enable.

EMAC_RMII	EMAC_REFCLK	PC.0	MPF6	I	EMAC RMII mode clock input
EPWMA	EPWM0_0	PA.12	MPF5	I/O	PWM0_0 output/capture input.
EPWMA	EPWM0_1	PA.11	MPF5	I/O	PWM0_1 output/capture input.
EPWMA	EPWM0_2	PA.10	MPF5	I/O	PWM0_2 output/capture input.
EPWMA	EPWM0_3	PA.9	MPF5	I/O	PWM0_3 output/capture input.
EPWMA	EPWM0_4	PA.8	MPF5	I/O	PWM0_4 output/capture input.
EPWMA	EPWM0_5	PA.7	MPF5	I/O	PWM0_5 output/capture input.
EPWMB	EPWM1_0	PB.6	MPF5	I/O	PWM1_0 output/capture input.
EPWMB	EPWM1_1	PB.7	MPF5	I/O	PWM1_1 output/capture input.
EPWMB	EPWM1_2	PB.8	MPF5	I/O	PWM1_2 output/capture input.
EPWMB	EPWM1_3	PB.9	MPF5	I/O	PWM1_3 output/capture input.
EPWMB	EPWM1_4	PB.10	MPF5	I/O	PWM1_4 output/capture input.
EPWMB	EPWM1_5	PB.11	MPF5	I/O	PWM1_5 output/capture input.
I2C0	I2C0_SCL	PA.15	MPF4	I/O	I2C0 clock pin.
I2C0	I2C0_SCL	PD.8	MPF2	I/O	I2C0 clock pin.
I2C0	I2C0_SDA	PC.9	MPF4	I/O	I2C0 data input/output pin.
I2C0	I2C0_SDA	PD.9	MPF2	I/O	I2C0 data input/output pin.
I2C1	I2C1_SCL	PG.15	MPF3	I/O	I2C1 clock pin.
I2C1	I2C1_SCL	PH.0	MPF1	I/O	I2C1 clock pin.
I2C1	I2C1_SDA	PG.14	MPF3	I/O	I2C1 data input/output pin.
I2C1	I2C1_SDA	PH.1	MPF2	I/O	I2C1 data input/output pin.
I2C2	I2C2_SCL	PB.6	MPF1	I/O	I2C2 clock pin.
I2C2	I2C2_SCL	PI.7	MPF1	I/O	I2C2 clock pin.
I2C2	I2C2_SDA	PB.7	MPF1	I/O	I2C2 data input/output pin.
I2C2	I2C2_SDA	PI.8	MPF1	I/O	I2C2 data input/output pin.
I2C3	I2C3_SCL	PD.2	MPF2	I/O	I2C3 clock pin.
I2C3	I2C3_SCL	PH.3	MPF1	I/O	I2C3 clock pin.
I2C3	I2C3_SDA	PD.3	MPF2	I/O	I2C3 data input/output pin.
I2C3	I2C3_SDA	PH.4	MPF1	I/O	I2C3 data input/output pin.
I2C4	I2C4_SCL	PB.0	MPF2	I/O	I2C4 clock pin.
I2C4	I2C4_SCL	PD.10	MPF2	I/O	I2C4 clock pin.
I2C4	I2C4_SCL	PI.11	MPF3	I/O	I2C4 clock pin.
I2C4	I2C4_SDA	PB.1	MPF2	I/O	I2C4 data input/output pin.
I2C4	I2C4_SDA	PD.12	MPF2	I/O	I2C4 data input/output pin.
I2C4	I2C4_SDA	PI.12	MPF3	I/O	I2C4 data input/output pin.

I2S0	I2S0_BCLK	PA.5	MPF3	O	I2S0 bit clock pin.
I2S0	I2S0_DI	PA.4	MPF3	I	I2S0 data input.
I2S0	I2S0_DO	PA.3	MPF3	O	I2S0 data output.
I2S0	I2S0_LRCK	PA.6	MPF3	O	I2S0 left right channel clock.
I2S0	I2S0_MCLK	PA.2	MPF3	O	I2S0 master clock output pin.
I2S1	I2S1_BCLK	PC.1	MPF1	O	I2S1 bit clock pin.
I2S1	I2S1_BCLK	PG.5	MPF2	O	I2S1 bit clock pin.
I2S1	I2S1_BCLK	PI.11	MPF2	O	I2S1 bit clock pin.
I2S1	I2S1_DI	PC.0	MPF1	I	I2S1 data input.
I2S1	I2S1_DI	PG.4	MPF2	I	I2S1 data input.
I2S1	I2S1_DI	PG.9	MPF2	I	I2S1 data input.
I2S1	I2S1_DO	PB.15	MPF1	O	I2S1 data output.
I2S1	I2S1_DO	PC.4	MPF1	O	I2S1 data output.
I2S1	I2S1_DO	PG.3	MPF2	O	I2S1 data output.
I2S1	I2S1_DO	PG.8	MPF2	O	I2S1 data output.
I2S1	I2S1_LRCK	PC.2	MPF1	O	I2S1 left right channel clock.
I2S1	I2S1_LRCK	PG.6	MPF2	O	I2S1 left right channel clock.
I2S1	I2S1_LRCK	PI.12	MPF2	O	I2S1 left right channel clock.
I2S1	I2S1_MCLK	PB.14	MPF1	O	I2S1 master clock output pin.
I2S1	I2S1_MCLK	PC.3	MPF1	O	I2S1 master clock output pin.
I2S1	I2S1_MCLK	PG.7	MPF2	O	I2S1 master clock output pin.
ICE_SW	ICE_CLK	PG.10	MPF1	I	Serial wired debugger clock pin
ICE_SW	ICE_DAT	PG.11	MPF1	I/O	Serial wired debugger data pin
INT	INT0	PA.0	MPF8	I	External interrupt0 input pin.
INT	INT1	PB.0	MPF8	I	External interrupt1 input pin.
INT	INT2	PC.0	MPF8	I	External interrupt2 input pin.
INT	INT3	PD.0	MPF8	I	External interrupt3 input pin.
INT	INT4	PE.0	MPF8	I	External interrupt4 input pin.
INT	INT5	PF.0	MPF8	I	External interrupt5 input pin.
INT	INT6	PG.0	MPF8	I	External interrupt6 input pin.
INT	INT7	PH.0	MPF8	I	External interrupt7 input pin.
GPIO	nRESET	nRESET	MFP0	I	External reset input: active LOW, with an internal pull-up. Set this pin low reset to initial state.
GPIO	PA.0	PA.0	MFP0	I/O	General purpose digital I/O pin.
GPIO	PA.1	PA.1	MFP0	I/O	General purpose digital I/O pin.

GPIO	PA.10	PA.10	MFP0	I/O	General purpose digital I/O pin.
GPIO	PA.11	PA.11	MFP0	I/O	General purpose digital I/O pin.
GPIO	PA.12	PA.12	MFP0	I/O	General purpose digital I/O pin.
GPIO	PA.13	PA.13	MFP0	I/O	General purpose digital I/O pin.
GPIO	PA.14	PA.14	MFP0	I/O	General purpose digital I/O pin.
GPIO	PA.15	PA.15	MFP0	I/O	General purpose digital I/O pin.
GPIO	PA.2	PA.2	MFP0	I/O	General purpose digital I/O pin.
GPIO	PA.3	PA.3	MFP0	I/O	General purpose digital I/O pin.
GPIO	PA.4	PA.4	MFP0	I/O	General purpose digital I/O pin.
GPIO	PA.5	PA.5	MFP0	I/O	General purpose digital I/O pin.
GPIO	PA.6	PA.6	MFP0	I/O	General purpose digital I/O pin.
GPIO	PA.7	PA.7	MFP0	I/O	General purpose digital I/O pin.
GPIO	PA.8	PA.8	MFP0	I/O	General purpose digital I/O pin.
GPIO	PA.9	PA.9	MFP0	I/O	General purpose digital I/O pin.
GPIO	PB.0	PB.0	MFP0	I/O	General purpose digital I/O pin.
GPIO	PB.1	PB.1	MFP0	I/O	General purpose digital I/O pin.
GPIO	PB.10	PB.10	MFP0	I/O	General purpose digital I/O pin.
GPIO	PB.11	PB.11	MFP0	I/O	General purpose digital I/O pin.
GPIO	PB.12	PB.12	MFP0	I/O	General purpose digital I/O pin.
GPIO	PB.13	PB.13	MFP0	I/O	General purpose digital I/O pin.
GPIO	PB.14	PB.14	MFP0	I/O	General purpose digital I/O pin.
GPIO	PB.15	PB.15	MFP0	I/O	General purpose digital I/O pin.
GPIO	PB.3	PB.3	MFP0	I/O	General purpose digital I/O pin.
GPIO	PB.4	PB.4	MFP0	I/O	General purpose digital I/O pin.
GPIO	PB.5	PB.5	MFP0	I/O	General purpose digital I/O pin.
GPIO	PB.6	PB.6	MFP0	I/O	General purpose digital I/O pin.
GPIO	PB.7	PB.7	MFP0	I/O	General purpose digital I/O pin.
GPIO	PB.8	PB.8	MFP0	I/O	General purpose digital I/O pin.
GPIO	PB.9	PB.9	MFP0	I/O	General purpose digital I/O pin.
GPIO	PC.0	PC.0	MFP0	I/O	General purpose digital I/O pin.
GPIO	PC.1	PC.1	MFP0	I/O	General purpose digital I/O pin.
GPIO	PC.10	PC.10	MFP0	I/O	General purpose digital I/O pin.
GPIO	PC.11	PC.11	MFP0	I/O	General purpose digital I/O pin.
GPIO	PC.12	PC.12	MFP0	I/O	General purpose digital I/O pin.
GPIO	PC.13	PC.13	MFP0	I/O	General purpose digital I/O pin.

GPIO	PC.14	PC.14	MFP0	I/O	General purpose digital I/O pin.
GPIO	PC.15	PC.15	MFP0	I/O	General purpose digital I/O pin.
GPIO	PC.2	PC.2	MFP0	I/O	General purpose digital I/O pin.
GPIO	PC.3	PC.3	MFP0	I/O	General purpose digital I/O pin.
GPIO	PC.4	PC.4	MFP0	I/O	General purpose digital I/O pin.
GPIO	PC.5	PC.5	MFP0	I/O	General purpose digital I/O pin.
GPIO	PC.6	PC.6	MFP0	I/O	General purpose digital I/O pin.
GPIO	PC.7	PC.7	MFP0	I/O	General purpose digital I/O pin.
GPIO	PC.8	PC.8	MFP0	I/O	General purpose digital I/O pin.
GPIO	PC.9	PC.9	MFP0	I/O	General purpose digital I/O pin.
GPIO	PD.0	PD.0	MFP0	I/O	General purpose digital I/O pin.
GPIO	PD.1	PD.1	MFP0	I/O	General purpose digital I/O pin.
GPIO	PD.10	PD.10	MFP0	I/O	General purpose digital I/O pin.
GPIO	PD.11	PD.11	MFP0	I/O	General purpose digital I/O pin.
GPIO	PD.12	PD.12	MFP0	I/O	General purpose digital I/O pin.
GPIO	PD.13	PD.13	MFP0	I/O	General purpose digital I/O pin.
GPIO	PD.14	PD.14	MFP0	I/O	General purpose digital I/O pin.
GPIO	PD.15	PD.15	MFP0	I/O	General purpose digital I/O pin.
GPIO	PD.2	PD.2	MFP0	I/O	General purpose digital I/O pin.
GPIO	PD.3	PD.3	MFP0	I/O	General purpose digital I/O pin.
GPIO	PD.4	PD.4	MFP0	I/O	General purpose digital I/O pin.
GPIO	PD.5	PD.5	MFP0	I/O	General purpose digital I/O pin.
GPIO	PD.6	PD.6	MFP0	I/O	General purpose digital I/O pin.
GPIO	PD.7	PD.7	MFP0	I/O	General purpose digital I/O pin.
GPIO	PD.8	PD.8	MFP0	I/O	General purpose digital I/O pin.
GPIO	PD.9	PD.9	MFP0	I/O	General purpose digital I/O pin.
GPIO	PE.0	PE.0	MFP0	I/O	General purpose digital I/O pin.
GPIO	PE.1	PE.1	MFP0	I/O	General purpose digital I/O pin.
GPIO	PE.10	PE.10	MFP0	I/O	General purpose digital I/O pin.
GPIO	PE.11	PE.11	MFP0	I/O	General purpose digital I/O pin.
GPIO	PE.12	PE.12	MFP0	I/O	General purpose digital I/O pin.
GPIO	PE.13	PE.13	MFP0	I/O	General purpose digital I/O pin.
GPIO	PE.14	PE.14	MFP0	I/O	General purpose digital I/O pin.
GPIO	PE.15	PE.15	MFP0	I/O	General purpose digital I/O pin.
GPIO	PE.2	PE.2	MFP0	I/O	General purpose digital I/O pin.

GPIO	PE.3	PE.3	MFP0	I/O	General purpose digital I/O pin.
GPIO	PE.4	PE.4	MFP0	I/O	General purpose digital I/O pin.
GPIO	PE.5	PE.5	MFP0	I/O	General purpose digital I/O pin.
GPIO	PE.6	PE.6	MFP0	I/O	General purpose digital I/O pin.
GPIO	PE.7	PE.7	MFP0	I/O	General purpose digital I/O pin.
GPIO	PE.8	PE.8	MFP0	I/O	General purpose digital I/O pin.
GPIO	PE.9	PE.9	MFP0	I/O	General purpose digital I/O pin.
GPIO	PF.0	PF.0	MFP0	I/O	General purpose digital I/O pin.
GPIO	PF.1	PF.1	MFP0	I/O	General purpose digital I/O pin.
GPIO	PF.10	PF.10	MFP0	I/O	General purpose digital I/O pin.
GPIO	PF.11	PF.11	MFP0	I/O	General purpose digital I/O pin.
GPIO	PF.12	PF.12	MFP0	I/O	General purpose digital I/O pin.
GPIO	PF.13	PF.13	MFP0	I/O	General purpose digital I/O pin.
GPIO	PF.14	PF.14	MFP0	I/O	General purpose digital I/O pin.
GPIO	PF.15	PF.15	MFP0	I/O	General purpose digital I/O pin.
GPIO	PF.2	PF.2	MFP0	I/O	General purpose digital I/O pin.
GPIO	PF.3	PF.3	MFP0	I/O	General purpose digital I/O pin.
GPIO	PF.4	PF.4	MFP0	I/O	General purpose digital I/O pin.
GPIO	PF.5	PF.5	MFP0	I/O	General purpose digital I/O pin.
GPIO	PF.6	PF.6	MFP0	I/O	General purpose digital I/O pin.
GPIO	PF.7	PF.7	MFP0	I/O	General purpose digital I/O pin.
GPIO	PF.8	PF.8	MFP0	I/O	General purpose digital I/O pin.
GPIO	PF.9	PF.9	MFP0	I/O	General purpose digital I/O pin.
GPIO	PG.0	PG.0	MFP0	I/O	General purpose digital I/O pin.
GPIO	PG.1	PG.1	MFP0	I/O	General purpose digital I/O pin.
GPIO	PG.10	PG.10	MFP0	I/O	General purpose digital I/O pin.
GPIO	PG.11	PG.11	MFP0	I/O	General purpose digital I/O pin.
GPIO	PG.12	PG.12	MFP0	I/O	General purpose digital I/O pin.
GPIO	PG.13	PG.13	MFP0	I/O	General purpose digital I/O pin.
GPIO	PG.14	PG.14	MFP0	I/O	General purpose digital I/O pin.
GPIO	PG.15	PG.15	MFP0	I/O	General purpose digital I/O pin.
GPIO	PG.2	PG.2	MFP0	I/O	General purpose digital I/O pin.
GPIO	PG.3	PG.3	MFP0	I/O	General purpose digital I/O pin.
GPIO	PG.4	PG.4	MFP0	I/O	General purpose digital I/O pin.
GPIO	PG.5	PG.5	MFP0	I/O	General purpose digital I/O pin.

GPIO	PG.6	PG.6	MFP0	I/O	General purpose digital I/O pin.
GPIO	PG.7	PG.7	MFP0	I/O	General purpose digital I/O pin.
GPIO	PG.8	PG.8	MFP0	I/O	General purpose digital I/O pin.
GPIO	PG.9	PG.9	MFP0	I/O	General purpose digital I/O pin.
GPIO	PH.0	PH.0	MFP0	I/O	General purpose digital I/O pin.
GPIO	PH.1	PH.1	MFP0	I/O	General purpose digital I/O pin.
GPIO	PH.10	PH.10	MFP0	I/O	General purpose digital I/O pin.
GPIO	PH.11	PH.11	MFP0	I/O	General purpose digital I/O pin.
GPIO	PH.12	PH.12	MFP0	I/O	General purpose digital I/O pin.
GPIO	PH.13	PH.13	MFP0	I/O	General purpose digital I/O pin.
GPIO	PH.14	PH.14	MFP0	I/O	General purpose digital I/O pin.
GPIO	PH.15	PH.15	MFP0	I/O	General purpose digital I/O pin.
GPIO	PH.2	PH.2	MFP0	I/O	General purpose digital I/O pin.
GPIO	PH.3	PH.3	MFP0	I/O	General purpose digital I/O pin.
GPIO	PH.4	PH.4	MFP0	I/O	General purpose digital I/O pin.
GPIO	PH.5	PH.5	MFP0	I/O	General purpose digital I/O pin.
GPIO	PH.6	PH.6	MFP0	I/O	General purpose digital I/O pin.
GPIO	PH.7	PH.7	MFP0	I/O	General purpose digital I/O pin.
GPIO	PH.8	PH.8	MFP0	I/O	General purpose digital I/O pin.
GPIO	PH.9	PH.9	MFP0	I/O	General purpose digital I/O pin.
GPIO	PI.0	PI.0	MFP0	I/O	General purpose digital I/O pin.
GPIO	PI.1	PI.1	MFP0	I/O	General purpose digital I/O pin.
GPIO	PI.10	PI.10	MFP0	I/O	General purpose digital I/O pin.
GPIO	PI.11	PI.11	MFP0	I/O	General purpose digital I/O pin.
GPIO	PI.12	PI.12	MFP0	I/O	General purpose digital I/O pin.
GPIO	PI.13	PI.13	MFP0	I/O	General purpose digital I/O pin.
GPIO	PI.14	PI.14	MFP0	I/O	General purpose digital I/O pin.
GPIO	PI.15	PI.15	MFP0	I/O	General purpose digital I/O pin.
GPIO	PI.2	PI.2	MFP0	I/O	General purpose digital I/O pin.
GPIO	PI.3	PI.3	MFP0	I/O	General purpose digital I/O pin.
GPIO	PI.4	PI.4	MFP0	I/O	General purpose digital I/O pin.
GPIO	PI.5	PI.5	MFP0	I/O	General purpose digital I/O pin.
GPIO	PI.6	PI.6	MFP0	I/O	General purpose digital I/O pin.
GPIO	PI.7	PI.7	MFP0	I/O	General purpose digital I/O pin.
GPIO	PI.8	PI.8	MFP0	I/O	General purpose digital I/O pin.

GPIO	PI.9	PI.9	MFP0	I/O	General purpose digital I/O pin.
OTG_PHY	USB0_D-	USB0_D-		A	USB0 differential signal D+.
OTG_PHY	USB0_D+	USB0_D+		A	USB0 differential signal D+.
OTG_PHY	USB0_OTG_ID	USB0_OTG_ID		I	USB0OTG ID pin.
OTG_PHY	VBUS	VBUS		A	USB PHY VBUS power input pin.
ADC	Vref	Vref		A	Voltage reference input for ADC.
OTG_PHY	VRES	VRES		A	USB PHY VRES ground input pin. Add an 8.2K ohm resistor to VSSA.
OTG_PHY	VSSA	VSSA		P	Ground pin for digital circuit. Add a Feritte Bead to digital ground VSS.
LDO_CAP	LDO_CAP	LDO_CAP		P	LDO output pin.
LDO_CAP	USB_VDD33_CAP	USB_VDD33_CAP		P	Internal power regulator output 3.3V decoupling pin.
OPA0	OPA0_IN-	PF.10	MPF1	I/O	General purpose digital I/O pin.
OPA0	OPA0_IN+	PF.9	MPF1	I/O	General purpose digital I/O pin.
OPA0	OPA0_O	PF.11	MPF1	O	Operational amplifier output pin
OPA1	OPA1_IN-	PF.13	MPF1	I/O	General purpose digital I/O pin.
OPA1	OPA1_IN+	PF.12	MPF1	I/O	General purpose digital I/O pin.
OPA1	OPA1_O	PF.14	MPF1	O	Operational amplifier output pin
OTG	USB0_VBUS_EN	PB.1	MPF1	O	USB0 external VBUS regulator enabled
OTG	USB0_VBUS_ST	PB.0	MPF1	I	USB0 external VBUS regulator status
OTHER	CLKO	PC.5	MPF1	O	Clock Output Pin.
POWER	AVDD	AVDD		P	Power supply for internal analog circuit.
POWER	AVSS	AVSS		P	Ground pin for digital circuit.
POWER	VBAT	VBAT		P	Battery power input pin.
POWER	VBUS	VBUS		A	USB PHY VBUS power input pin.
POWER	VDD	VDD		P	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
POWER	VSS	VSS		P	Ground pin for digital circuit.
POWER	VSSA	VSSA		P	Ground pin for digital circuit. Add a Feritte Bead to digital ground VSS.
PS2D	PS2_CLK	PG.3	MPF1	O	PS2 clock pin.
PS2D	PS2_DAT	PG.4	MPF1	I/O	PS2 data pin.
QEI0	QEI0_A	PC.5	MPF5	I	Quadrature encoder phase A input of QEI Unit 0.
QEI0	QEI0_B	PC.4	MPF5	I	Quadrature encoder phase B input of QEI Unit 0.
QEI0	QEI0_Z	PC.3	MPF5	I	Quadrature encoder phase Z input of QEI Unit 0.

QEI1	QEI1_A	PA.6	MPF5	I	Quadrature encoder phase A input of QEI Unit 1.
QEI1	QEI1_B	PA.5	MPF5	I	Quadrature encoder phase B input of QEI Unit 1.
QEI1	QEI1_Z	PA.4	MPF5	I	Quadrature encoder phase Z input of QEI Unit 1.
SC0	SC0_CD	PA.0	MPF2	I	SmartCard0 card detect pin.
SC0	SC0_CLK	PA.7	MPF2	O	SmartCard0 clock pin.
SC0	SC0_DAT	PA.10	MPF2	I/O	SmartCard0 data pin.
SC0	SC0_PWR	PA.9	MPF2	O	SmartCard0 power pin.
SC0	SC0_RST	PA.8	MPF2	O	SmartCard0 reset pin.
SC1	SC1_CD	PC.3	MPF2	I	SmartCard1 card detect pin.
SC1	SC1_CD	PG.7	MPF3	I	SmartCard1 card detect pin.
SC1	SC1_CLK	PC.1	MPF2	O	SmartCard1 clock pin.
SC1	SC1_CLK	PG.6	MPF3	O	SmartCard1 clock pin.
SC1	SC1_DAT	PB.15	MPF2	I/O	SmartCard1 data pin.
SC1	SC1_DAT	PC.0	MPF2	I/O	SmartCard1 data pin.
SC1	SC1_DAT	PG.5	MPF3	I/O	SmartCard1 data pin.
SC1	SC1_PWR	PC.2	MPF2	O	SmartCard1 power pin.
SC1	SC1_PWR	PG.4	MPF3	O	SmartCard1 power pin.
SC1	SC1_RST	PB.14	MPF2	O	SmartCard1 reset pin.
SC1	SC1_RST	PC.4	MPF2	O	SmartCard1 reset pin.
SC1	SC1_RST	PG.3	MPF3	O	SmartCard1 reset pin.
SC2	SC2_CD	PA.6	MPF1	I	SmartCard2 card detect pin.
SC2	SC2_CLK	PA.3	MPF1	O	SmartCard2 clock pin.
SC2	SC2_DAT	PA.2	MPF1	I/O	SmartCard2 data pin.
SC2	SC2_PWR	PA.4	MPF1	O	SmartCard2 power pin.
SC2	SC2_RST	PA.5	MPF1	O	SmartCard2 reset pin.
SC3	SC3_CD	PC.10	MPF1	I	SmartCard3 card detect pin.
SC3	SC3_CD	PI.12	MPF4	I	SmartCard3 card detect pin.
SC3	SC3_CLK	PA.14	MPF3	O	SmartCard3 clock pin.
SC3	SC3_CLK	PD.12	MPF1	O	SmartCard3 clock pin.
SC3	SC3_CLK	PG.9	MPF4	O	SmartCard3 clock pin.
SC3	SC3_DAT	PA.13	MPF3	I/O	SmartCard3 data pin.
SC3	SC3_DAT	PD.10	MPF1	I/O	SmartCard3 data pin.
SC3	SC3_DAT	PG.8	MPF4	I/O	SmartCard3 data pin.
SC3	SC3_PWR	PA.15	MPF1	O	SmartCard3 power pin.
SC3	SC3_PWR	PI.11	MPF4	O	SmartCard3 power pin.

SC3	SC3_RST	PC.9	MPF3	O	SmartCard3 reset pin.
SC3	SC3_RST	PD.11	MPF1	O	SmartCard3 reset pin.
SC3	SC3_RST	PG.7	MPF4	O	SmartCard3 reset pin.
SC4	SC4_CD	PC.12	MPF2	I	SmartCard4 card detect pin.
SC4	SC4_CLK	PD.0	MPF2	O	SmartCard4 clock pin.
SC4	SC4_DAT	PC.15	MPF2	I/O	SmartCard4 data pin.
SC4	SC4_PWR	PC.14	MPF2	O	SmartCard4 power pin.
SC4	SC4_RST	PC.13	MPF2	O	SmartCard4 reset pin.
SC5	SC5_CD	PA.1	MPF2	I	SmartCard5 card detect pin.
SC5	SC5_CD	PD.4	MPF1	I	SmartCard5 card detect pin.
SC5	SC5_CLK	PD.3	MPF1	O	SmartCard5 clock pin.
SC5	SC5_CLK	PH.15	MPF2	O	SmartCard5 clock pin.
SC5	SC5_DAT	PD.7	MPF1	I/O	SmartCard5 data pin.
SC5	SC5_DAT	PI.2	MPF1	I/O	SmartCard5 data pin.
SC5	SC5_PWR	PD.6	MPF1	O	SmartCard5 power pin.
SC5	SC5_PWR	PI.1	MPF2	O	SmartCard5 power pin.
SC5	SC5_RST	PD.5	MPF1	O	SmartCard5 reset pin.
SC5	SC5_RST	PI.0	MPF2	O	SmartCard5 reset pin.
SDHOST0	SD0_CDn	PD.3	MPF4	I	SD mode #0 – card detect
SDHOST0	SD0_CDn	PE.5	MPF4	I	SD mode #0 – card detect
SDHOST0	SD0_CDn	PF.6	MPF4	I	SD mode #0 – card detect
SDHOST0	SD0_CLK	PD.7	MPF4	O	SD mode #0– clock;
SDHOST0	SD0_CLK	PE.7	MPF4	O	SD mode #0– clock;
SDHOST0	SD0_CLK	PF.8	MPF4	O	SD mode #0– clock;
SDHOST0	SD0_CMD	PD.6	MPF4	I/O	SD mode #0 – command/response
SDHOST0	SD0_CMD	PE.6	MPF4	I/O	SD mode #0 – command/response
SDHOST0	SD0_CMD	PF.7	MPF4	I/O	SD mode #0 – command/response
SDHOST0	SD0_DAT0	PE.11	MPF4	I/O	SD mode #0 data line bit 0;
SDHOST0	SD0_DAT0	PF.5	MPF4	I/O	SD mode #0 data line bit 0;
SDHOST0	SD0_DAT1	PE.10	MPF4	I/O	SD mode #0 data line bit 1;
SDHOST0	SD0_DAT1	PF.4	MPF4	I/O	SD mode #0 data line bit 1;
SDHOST0	SD0_DAT2	PE.9	MPF4	I/O	SD mode #0 data line bit 2;
SDHOST0	SD0_DAT2	PF.3	MPF4	I/O	SD mode #0 data line bit 2;
SDHOST0	SD0_DAT3	PE.8	MPF4	I/O	SD mode #0 data line bit 3;
SDHOST0	SD0_DAT3	PF.2	MPF4	I/O	SD mode #0 data line bit 3;

SDHOST1	SD1_CDn	PC.12	MPF4	I	SD mode #1 – card detect
SDHOST1	SD1_CLK	PC.14	MPF4	O	SD mode #1– clock;
SDHOST1	SD1_CMD	PC.13	MPF4	I/O	SD mode #1 – command/response
SDHOST1	SD1_DAT0	PD.2	MPF4	I/O	SD mode #1 data line bit 0;
SDHOST1	SD1_DAT1	PD.1	MPF4	I/O	SD mode #1 data line bit 1;
SDHOST1	SD1_DAT2	PD.0	MPF4	I/O	SD mode #1 data line bit 2;
SDHOST1	SD1_DAT3	PC.15	MPF4	I/O	SD mode #1 data line bit 3;
Slew	HS	PA.2	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PA.3	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PA.4	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PA.5	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PA.6	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PA.7	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PA.8	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PA.9	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PA.10	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PA.11	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PA.12	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PA.13	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PA.14	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PA.15	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PB.3	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PB.4	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PB.5	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PB.6	Slew		This pad is embedded with “Slew Rate Control” capability.

Slew	HS	PB.7	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PB.8	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PB.9	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PB.10	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PB.11	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PB.12	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PB.13	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PB.14	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PB.15	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PC.0	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PC.1	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PC.2	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PC.3	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PC.4	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PC.5	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PC.6	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PC.7	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PC.8	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PC.9	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PC.10	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PC.11	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PC.12	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PC.13	Slew		This pad is embedded with “Slew Rate Control” capability.

Slew	HS	PC.14	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PC.15	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PD.0	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PD.1	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PD.2	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PD.3	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PD.4	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PD.5	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PD.6	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PD.7	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PD.8	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PD.9	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PD.13	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PD.14	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PD.15	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PE.2	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PE.3	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PE.4	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PE.5	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PE.6	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PE.7	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PE.8	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PE.9	Slew		This pad is embedded with “Slew Rate Control” capability.

Slew	HS	PE.10	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PE.11	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PE.12	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PE.13	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PE.14	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PF.0	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PF.1	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PF.2	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PF.3	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PF.4	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PF.5	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PF.6	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PF.7	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PF.8	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PF.9	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PF.10	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PG.7	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PG.8	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PG.9	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PH.5	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PH.6	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PH.7	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PH.8	Slew		This pad is embedded with “Slew Rate Control” capability.

Slew	HS	PH.9	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PH.10	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PI.3	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PI.4	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PI.5	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PI.6	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PI.7	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PI.8	Slew		This pad is embedded with “Slew Rate Control” capability.
Slew	HS	PI.11	Slew		This pad is embedded with “Slew Rate Control” capability.
SPI0	SPI0_CLK	PC.8	MPF4	O	SPI0 serial clock pin.
SPI0	SPI0_CLK	PE.5	MPF3	O	SPI0 serial clock pin.
SPI0	SPI0_MISO0	PC.6	MPF4	I/O	1st SPI0 MISO (Master In, Slave Out) pin.
SPI0	SPI0_MISO0	PE.2	MPF3	I/O	1st SPI0 MISO (Master In, Slave Out) pin.
SPI0	SPI0_MISO0	PE.6	MPF3	I/O	1st SPI0 MISO (Master In, Slave Out) pin.
SPI0	SPI0_MISO1	PC.3	MPF4	I/O	2nd SPI0 MISO (Master In, Slave Out) pin.
SPI0	SPI0_MISO1	PE.10	MPF3	I/O	2nd SPI0 MISO (Master In, Slave Out) pin.
SPI0	SPI0_MOSI0	PC.7	MPF4	I/O	1st SPI0 MOSI (Master Out, Slave In) pin.
SPI0	SPI0_MOSI0	PE.3	MPF3	I/O	1st SPI0 MOSI (Master Out, Slave In) pin.
SPI0	SPI0_MOSI0	PE.7	MPF3	I/O	1st SPI0 MOSI (Master Out, Slave In) pin.
SPI0	SPI0_MOSI1	PC.4	MPF4	I/O	2nd SPI0 MOSI (Master Out, Slave In) pin.
SPI0	SPI0_MOSI1	PE.11	MPF3	I/O	2nd SPI0 MOSI (Master Out, Slave In) pin.
SPI0	SPI0_SS0	PC.2	MPF4	I/O	1st SPI0 slave select pin.
SPI0	SPI0_SS0	PE.4	MPF3	I/O	1st SPI0 slave select pin.
SPI1	SPI1_CLK	PD.1	MPF1	O	SPI1 serial clock pin.
SPI1	SPI1_CLK	PD.14	MPF1	O	SPI1 serial clock pin.
SPI1	SPI1_MISO0	PD.0	MPF1	I/O	1st SPI1 MISO (Master In, Slave Out) pin.
SPI1	SPI1_MISO0	PD.15	MPF1	I/O	1st SPI1 MISO (Master In, Slave Out) pin.
SPI1	SPI1_MISO1	PC.14	MPF1	I/O	2nd SPI1 MISO (Master In, Slave Out) pin.
SPI1	SPI1_MOSI0	PC.15	MPF1	I/O	1st SPI1 MOSI (Master Out, Slave In) pin.
SPI1	SPI1_MOSI0	PF.0	MPF1	I/O	1st SPI1 MOSI (Master Out, Slave In) pin.

SPI1	SPI1_MOSI1	PC.13	MPF1	I/O	2nd SPI1 MOSI (Master Out, Slave In) pin.
SPI1	SPI1_SS0	PC.12	MPF1	I/O	1st SPI1 slave select pin.
SPI1	SPI1_SS0	PD.13	MPF1	I/O	1st SPI1 slave select pin.
SPI2	SPI2_CLK	PB.3	MPF2	O	SPI2 serial clock pin.
SPI2	SPI2_CLK	PG.9	MPF1	O	SPI2 serial clock pin.
SPI2	SPI2_CLK	PH.6	MPF1	O	SPI2 serial clock pin.
SPI2	SPI2_MISO0	PB.4	MPF2	I/O	1st SPI2 MISO (Master In, Slave Out) pin.
SPI2	SPI2_MISO0	PG.7	MPF1	I/O	1st SPI2 MISO (Master In, Slave Out) pin.
SPI2	SPI2_MISO0	PH.7	MPF1	I/O	1st SPI2 MISO (Master In, Slave Out) pin.
SPI2	SPI2_MISO1	PB.12	MPF2	I/O	2nd SPI2 MISO (Master In, Slave Out) pin.
SPI2	SPI2_MISO1	PH.9	MPF1	I/O	2nd SPI2 MISO (Master In, Slave Out) pin.
SPI2	SPI2_MISO1	PI.12	MPF1	I/O	2nd SPI2 MISO (Master In, Slave Out) pin.
SPI2	SPI2_MOSI0	PB.5	MPF2	I/O	1st SPI2 MOSI (Master Out, Slave In) pin.
SPI2	SPI2_MOSI0	PG.8	MPF1	I/O	1st SPI2 MOSI (Master Out, Slave In) pin.
SPI2	SPI2_MOSI0	PH.8	MPF1	I/O	1st SPI2 MOSI (Master Out, Slave In) pin.
SPI2	SPI2_MOSI1	PB.13	MPF2	I/O	2nd SPI2 MOSI (Master Out, Slave In) pin.
SPI2	SPI2_MOSI1	PF.1	MPF1	I/O	2nd SPI2 MOSI (Master Out, Slave In) pin.
SPI2	SPI2_MOSI1	PH.10	MPF1	I/O	2nd SPI2 MOSI (Master Out, Slave In) pin.
SPI2	SPI2_SS0	PB.2	MPF2	I/O	1st SPI2 slave select pin.
SPI2	SPI2_SS0	PH.5	MPF1	I/O	1st SPI2 slave select pin.
SPI2	SPI2_SS0	PI.11	MPF1	I/O	1st SPI2 slave select pin.
SPI3	SPI3_CLK	PA.4	MPF2	O	SPI3 serial clock pin.
SPI3	SPI3_CLK	PA.8	MPF3	O	SPI3 serial clock pin.
SPI3	SPI3_CLK	PF.3	MPF1	O	SPI3 serial clock pin.
SPI3	SPI3_CLK	PI.4	MPF1	O	SPI3 serial clock pin.
SPI3	SPI3_MISO0	PA.2	MPF2	I/O	1st SPI3 MISO (Master In, Slave Out) pin.
SPI3	SPI3_MISO0	PA.9	MPF3	I/O	1st SPI3 MISO (Master In, Slave Out) pin.
SPI3	SPI3_MISO0	PF.4	MPF1	I/O	1st SPI3 MISO (Master In, Slave Out) pin.
SPI3	SPI3_MISO0	PI.5	MPF1	I/O	1st SPI3 MISO (Master In, Slave Out) pin.
SPI3	SPI3_MISO1	PA.11	MPF3	I/O	2nd SPI3 MISO (Master In, Slave Out) pin.
SPI3	SPI3_MISO1	PD.8	MPF1	I/O	2nd SPI3 MISO (Master In, Slave Out) pin.
SPI3	SPI3_MISO1	PI.7	MPF2	I/O	2nd SPI3 MISO (Master In, Slave Out) pin.
SPI3	SPI3_MOSI0	PA.3	MPF2	I/O	1st SPI3 MOSI (Master Out, Slave In) pin.
SPI3	SPI3_MOSI0	PA.10	MPF3	I/O	1st SPI3 MOSI (Master Out, Slave In) pin.
SPI3	SPI3_MOSI0	PF.5	MPF1	I/O	1st SPI3 MOSI (Master Out, Slave In) pin.

SPI3	SPI3_MOSI0	PI.6	MPF1	I/O	1st SPI3 MOSI (Master Out, Slave In) pin.
SPI3	SPI3_MOSI1	PA.12	MPF3	I/O	2nd SPI3 MOSI (Master Out, Slave In) pin.
SPI3	SPI3_MOSI1	PD.9	MPF1	I/O	2nd SPI3 MOSI (Master Out, Slave In) pin.
SPI3	SPI3_MOSI1	PI.8	MPF2	I/O	2nd SPI3 MOSI (Master Out, Slave In) pin.
SPI3	SPI3_SS0	PA.5	MPF2	I/O	1st SPI3 slave select pin.
SPI3	SPI3_SS0	PA.7	MPF3	I/O	1st SPI3 slave select pin.
SPI3	SPI3_SS0	PF.2	MPF1	I/O	1st SPI3 slave select pin.
SPI3	SPI3_SS0	PI.3	MPF1	I/O	1st SPI3 slave select pin.
T0EX	TM0_EXT	PC.8	MPF1	I	Timer0 external capture input
T1EX	TM1_EXT	PC.7	MPF1	I	Timer1 external capture input
T2EX	TM2_EXT	PC.6	MPF1	I	Timer2 external capture input
T3EX	TM3_EXT	PC.14	MPF3	I	Timer3 external capture input
TAMPER	TAMPER0	PA.0	MPF1	I/O	Tamper detect pin 0.
TAMPER	TAMPER1	PA.1	MPF1	I/O	Tamper detect pin 1.
Timer0	TM0_CNT_OUT	PB.4	MPF4	I/O	Timer0 event counter input/toggle output.
Timer0	TM0_CNT_OUT	PD.1	MPF3	I/O	Timer0 event counter input/toggle output.
Timer1	TM1_CNT_OUT	PB.1	MPF3	I/O	Timer1 event counter input/toggle output.
Timer1	TM1_CNT_OUT	PE.8	MPF3	I/O	Timer1 event counter input/toggle output.
Timer2	TM2_CNT_OUT	PC.6	MPF5	I/O	Timer2 event counter input/toggle output.
Timer2	TM2_CNT_OUT	PE.1	MPF3	I/O	Timer2 event counter input/toggle output.
Timer3	TM3_CNT_OUT	PC.1	MPF5	I/O	Timer3 event counter input/toggle output.
Timer3	TM3_CNT_OUT	PD.11	MPF3	I/O	Timer3 event counter input/toggle output.
UART0	UART0_CTS	PA.12	MPF1	I	Clear to Send input pin for UART0.
UART0	UART0_CTS	PG.0	MPF1	I	Clear to Send input pin for UART0.
UART0	UART0_RTS	PA.11	MPF1	O	Request to Send output pin for UART0.
UART0	UART0_RTS	PF.15	MPF1	O	Request to Send output pin for UART0.
UART0	UART0_RXD	PA.13	MPF1	I	Data receiver input pin for UART0.
UART0	UART0_RXD	PG.1	MPF1	I	Data receiver input pin for UART0.
UART0	UART0_TXD	PA.14	MPF1	O	Data transmitter output pin for UART0.
UART0	UART0_TXD	PG.2	MPF1	O	Data transmitter output pin for UART0.
UART1	UART1_CTS	PB.5	MPF1	I	Clear to Send input pin for UART1.
UART1	UART1_CTS	PF.12	MPF2	I	Clear to Send input pin for UART1.
UART1	UART1_RTS	PB.4	MPF1	O	Request to Send output pin for UART1.
UART1	UART1_RTS	PF.11	MPF2	O	Request to Send output pin for UART1.
UART1	UART1_RXD	PB.2	MPF1	I	Data receiver input pin for UART1.

UART1	UART1_RXD	PF.14	MPF2	I	Data receiver input pin for UART1.
UART1	UART1_TXD	PB.3	MPF1	O	Data transmitter output pin for UART1.
UART1	UART1_TXD	PF.13	MPF2	O	Data transmitter output pin for UART1.
UART2	UART2_CTS	PC.9	MPF2	I	Clear to Send input pin for UART2.
UART2	UART2_CTS	PH.2	MPF1	I	Clear to Send input pin for UART2.
UART2	UART2_RTS	PA.15	MPF2	O	Request to Send output pin for UART2.
UART2	UART2_RTS	PF.8	MPF1	O	Request to Send output pin for UART2.
UART2	UART2_RXD	PC.10	MPF2	I	Data receiver input pin for UART2.
UART2	UART2_RXD	PF.6	MPF1	I	Data receiver input pin for UART2.
UART2	UART2_TXD	PC.11	MPF1	O	Data transmitter output pin for UART2.
UART2	UART2_TXD	PF.7	MPF1	O	Data transmitter output pin for UART2.
UART3	UART3_CTS	PD.7	MPF2	I	Clear to Send input pin for UART3.
UART3	UART3_CTS	PH.14	MPF1	I	Clear to Send input pin for UART3.
UART3	UART3_RTS	PD.6	MPF2	O	Request to Send output pin for UART3.
UART3	UART3_RTS	PH.13	MPF1	O	Request to Send output pin for UART3.
UART3	UART3_RXD	PD.4	MPF2	I	Data receiver input pin for UART3.
UART3	UART3_RXD	PH.11	MPF1	I	Data receiver input pin for UART3.
UART3	UART3_TXD	PD.5	MPF2	O	Data transmitter output pin for UART3.
UART3	UART3_TXD	PH.12	MPF1	O	Data transmitter output pin for UART3.
UART4	UART4_CTS	PB.7	MPF3	I	Clear to Send input pin for UART4.
UART4	UART4_CTS	PB.13	MPF1	I	Clear to Send input pin for UART4.
UART4	UART4_CTS	PC.3	MPF3	I	Clear to Send input pin for UART4.
UART4	UART4_CTS	PG.9	MPF3	I	Clear to Send input pin for UART4.
UART4	UART4_RTS	PB.6	MPF3	O	Request to Send output pin for UART4.
UART4	UART4_RTS	PB.12	MPF1	O	Request to Send output pin for UART4.
UART4	UART4_RTS	PC.2	MPF3	O	Request to Send output pin for UART4.
UART4	UART4_RTS	PG.8	MPF3	O	Request to Send output pin for UART4.
UART4	UART4_RXD	PB.4	MPF3	I	Data receiver input pin for UART4.
UART4	UART4_RXD	PC.0	MPF3	I	Data receiver input pin for UART4.
UART4	UART4_RXD	PH.0	MPF2	I	Data receiver input pin for UART4.
UART4	UART4_TXD	PB.5	MPF3	O	Data transmitter output pin for UART4.
UART4	UART4_TXD	PC.1	MPF3	O	Data transmitter output pin for UART4.
UART4	UART4_TXD	PH.1	MPF1	O	Data transmitter output pin for UART4.
UART5	UART5_CTS	PB.8	MPF1	I	Clear to Send input pin for UART5.
UART5	UART5_CTS	PD.13	MPF2	I	Clear to Send input pin for UART5.

UART5	UART5_RTS	PB.9	MPF1	O	Request to Send output pin for UART5.
UART5	UART5_RTS	PD.14	MPF2	O	Request to Send output pin for UART5.
UART5	UART5_RXD	PB.11	MPF1	I	Data receiver input pin for UART5.
UART5	UART5_RXD	PF.0	MPF2	I	Data receiver input pin for UART5.
UART5	UART5_TXD	PB.10	MPF1	O	Data transmitter output pin for UART5.
UART5	UART5_TXD	PD.15	MPF2	O	Data transmitter output pin for UART5.
USBIOPHY	USB1_D-	PB.2	MPF3	A	USB1 differential signal D+.
USBIOPHY	USB1_D+	PB.3	MPF3	A	USB1 differential signal D+.
USBPHY	USB_VDD33_CAP	USB_VDD33_CAP		P	Internal power regulator output 3.3V decoupling pin.
USBPHY	USB0_D-	USB0_D-		A	USB0 differential signal D+.
USBPHY	USB0_D+	USB0_D+		A	USB0 differential signal D+.
USBPHY	USB0_OTG_ID	USB0_OTG_ID		I	USB0OTG ID pin.
USBPHY	VBUS	VBUS		A	USB PHY VBUS power input pin.
USBPHY	VRES	VRES		A	USB PHY VRES ground input pin. Add an 8.2K ohm resistor to VSSA.
USBPHY	VSSA	VSSA		A	Ground pin for digital circuit. Add a Feritte Bead to digital ground VSS.
CAP	CAP_DATA0	PD.3	MPF5	I	Image data input bus bit 7.
CAP	CAP_DATA1	PD.2	MPF5	I	Image data input bus bit 7.
CAP	CAP_DATA2	PD.1	MPF5	I	Image data input bus bit 7.
CAP	CAP_DATA3	PD.0	MPF5	I	Image data input bus bit 7.
CAP	CAP_DATA4	PC.15	MPF5	I	Image data input bus bit 7.
CAP	CAP_DATA5	PC.14	MPF5	I	Image data input bus bit 7.
CAP	CAP_DATA6	PC.13	MPF5	I	Image data input bus bit 7.
CAP	CAP_DATA7	PC.12	MPF5	I	Image data input bus bit 7.
CAP	CAP_HSYNC	PD.6	MPF5	I	Image capture interface HSYNC input pin.
CAP	CAP_PIXCLK	PD.7	MPF5	I	Image capture interface pix clock input pin.
CAP	CAP_SCLK	PD.4	MPF5	O	Image capture interface sensor clock pin.
CAP	CAP_SFIELD	PA.2	MPF5	I	Video input interface SFIELD input pin.
CAP	CAP_VSYNC	PD.5	MPF5	I	Image capture interface VSYNC input pin.
X32K	X32K_IN	PG.15	MPF1	I	External 32.768 kHz (low-speed) crystal input pin.
X32K	X32K_OUT	PG.14	MPF1	O	External 32.768 kHz (low-speed) crystal output pin.
XIN1	XT1_IN	PG.13	MPF1	I	External 4~24 MHz (high-speed) crystal input pin.
XIN1	XT1_OUT	PG.12	MPF1	O	External 4~24 MHz (high-speed) crystal output pin.

5 BLOCK DIAGRAM

5.1 NuMicro™ NUC442 Series Block Diagram

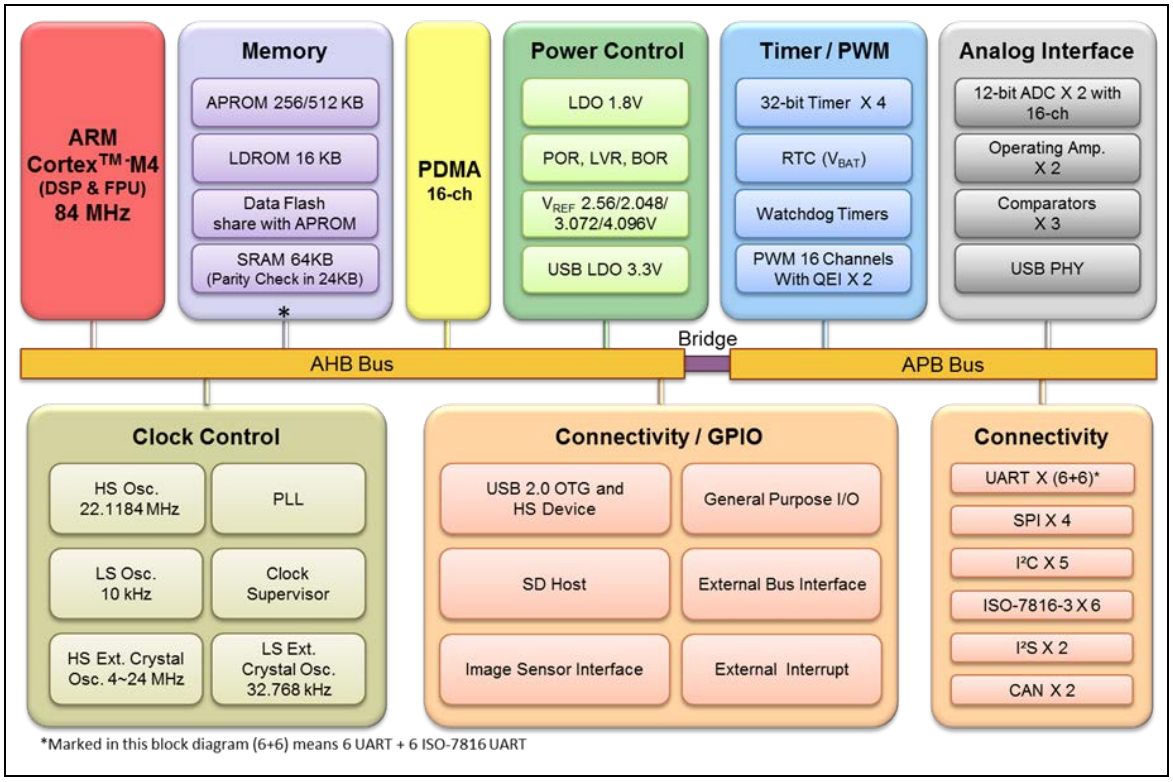


Figure 5.1-1 NuMicro™ NUC442 Series Block Diagram

5.2 NuMicro™ NUC472 Series Block Diagram

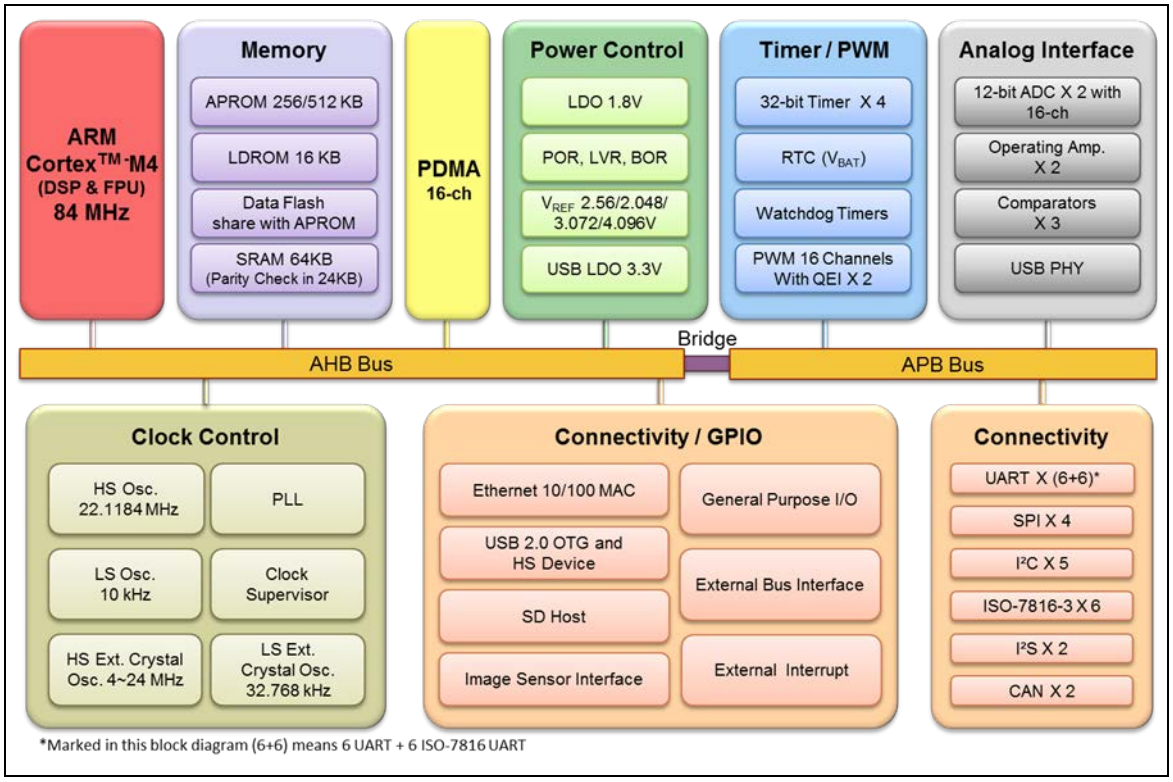


Figure 5.2-1 NuMicro™ NUC472 Series Block Diagram

6 FUNCTIONAL DESCRIPTION

6.1 ARM® Cortex®-M4 Core

The Cortex®-M4 processor, a configurable, multistage, 32-bit RISC processor, has three AMBA AHB-Lite interfaces for best parallel performance and includes an NVIC component. The processor with optional hardware debug functionality can execute Thumb code and is compatible with other Cortex®-M profile processors. The profile supports two modes -Thread mode and Handler mode. Handler mode is entered as a result of an exception. An exception return can only be issued in Handler mode. Thread mode is entered on Reset, and can be entered as a result of an exception return. The Cortex®-M4F is a processor with the same capability as the Cortex®-M4 processor and includes floating point arithmetic functionality. The NUC442/NUC472 series is embedded with Cortex®-M4F processor. Throughout this document, the name Cortex®-M4 refers to both Cortex®-M4 and Cortex®-M4F processors. The following figure shows the functional controller of the processor.

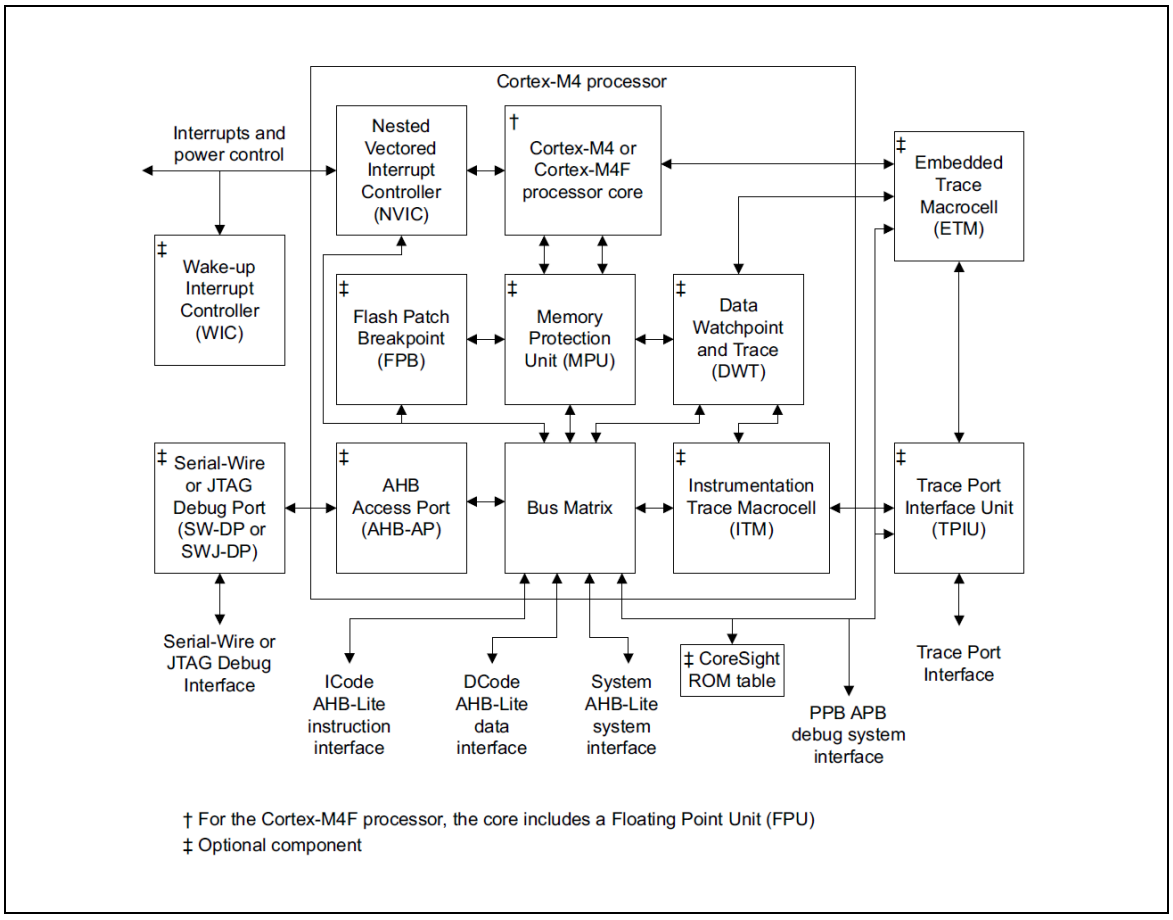


Figure 6.1-1 Cortex®-M4 Block Diagram

Cortex®-M4 processor features:

- A low gate count processor core, with low latency interrupt processing that has:
  - A subset of the Thumb instruction set, defined in the *ARMv7-M Architecture Reference Manual*

- Banked Stack Pointer (SP)
- Hardware integer divide instructions, SDIV and UDIV
- Handler and Thread modes
- Thumb and Debug states
- Support for interruptible-continued instructions LDM, STM, PUSH, and POP for low interrupt latency
- Automatic processor state saving and restoration for low latency *Interrupt Service Routine (ISR)* entry and exit
- Support for ARMv6 big-endian byte-invariant or little-endian accesses
- Support for ARMv6 unaligned accesses

- Floating Point Unit (FPU) in the Cortex®-M4F processor providing:
  - 32-bit instructions for single-precision (C float) data-processing operations
  - Combined Multiply and Accumulate instructions for increased precision (Fused MAC)
  - Hardware support for conversion, addition, subtraction, multiplication with optional accumulate, division, and square-root
  - Hardware support for denormals and all IEEE rounding modes
  - 32 dedicated 32-bit single precision registers, also addressable as 16 double-word registers
  - Decoupled three stage pipeline
- Nested Vectored Interrupt Controller (NVIC) closely integrated with the processor core to achieve low latency interrupt processing. Features include:
  - External interrupts. Configurable from 1 to 240 (the NUC442/NUC472 series configured with 97 interrupts)
  - Bits of priority, configurable from 3 to 8
  - Dynamic reprioritization of interrupts
  - Priority grouping which enables selection of preempting interrupt levels and non-preempting interrupt levels
  - Support for tril-chaining and late arrival of interrupts, which enables back-to-back interrupt processing without the overhead of state saving and restoration between interrupts.
  - Processor state automatically saved on interrupt entry, and restored on interrupt exit with on instruction overhead
  - Support for Wake-up Interrupt Controller (WIC) with Ultra-low Power Sleep mode
- Memory Protection Unit (MPU). An optional MPU for memory protection, including:
  - Eight memory regions
  - Sub Region Disable (SRD), enabling efficient use of memory regions
  - The ability to enable a background region that implements the default memory map attributes
- Low-cost debug solution that features:

- Debug access to all memory and registers in the system, including access to memory mapped devices, access to internal core registers when the core is halted, and access to debug control registers even while SYSRESETn is asserted.
- Serial Wire Debug Port(SW-DP) debug access
- Optional Flash Patch and Breakpoint (FPB) unit for implementing breakpoints and code patches
- Optional Data Watchpoint and Trace (DWT) unit for implementing watchpoints, data tracing, and system profiling
- Optional Instrumentation Trace Macrocell (ITM) for support of printf() style debugging
- Optional Trace Port Interface Unit (TPIU) for bridging to a Trace Port Analyzer (TPA), including Single Wire Output (SWO) mode

- Bus interfaces:
  - Three Advanced High-performance Bus-Lite (AHB-Lite) interfaces: ICode, Dcode, and System bus interfaces
  - Private Peripheral Bus (PPB) based on Advanced Peripheral Bus (APB) interface
  - Bit-band support that includes atomic bit-band write and read operations.
  - Memory access alignment
  - Write buffer for buffering of write data
  - Exclusive access transfers for multiprocessor systems

6.2 System Manager

6.2.1 Overview

System management includes the following sections:

- System Resets
- System Memory Map
- System management registers for Part Number ID, chip reset and on-chip controllers reset , multi-functional pin control
- System Timer (SysTick)
- Nested Vectored Interrupt Controller (NVIC)
- System Control registers

6.2.2 System Reset

The system reset can be issued by one of the below listed events. For these reset event flags can be read by SYS_RSTSTS register.

- Hardware Reset
  - Power-on Reset (POR)
  - Low level on the nRESEST Pin (nRST)
  - Watchdag Time-out Reset (WDT)
  - Low Voltage Reset (LVR)
  - Brown out Detector Reset BOD_RST)
- Software Reset
  - MCU Reset- SYSRESETREQ(AIRCR[2])
  - Cortes-M4 Core One-shot Reset – CPURST(IPRSTC[1])
  - Chip One-shot Reset – CHIPRST(IPRSTC[0])

Note: ISPCON.BS keeps the original value after MCU Reset and CPU Reset.

6.2.3 System Power Distribution

In this chip, power distribution is divided into three segments:

- Analog power from AVDD and AVSS provides the power for analog components operation.
- Digital power from VDD and VSS supplies the power to the internal regulator which provides a fixed 1.8 V power for digital operation and I/O pins.
- USB transceiver power from VBUS offers the power for operating the USB transceiver.
- RTC power from V_{BAT} supplies the power to internal regulator which provides a fixed 1.8V power for RTC operation and I/O pins .

The outputs of internal voltage regulators, LDO and VDD33, require an external capacitor which should be located close to the corresponding pin. Analog power (AVDD) should be the same voltage level of the digital power (VDD). The following figure shows the power distribution of the NuMicro™ NUC442/NUC472.

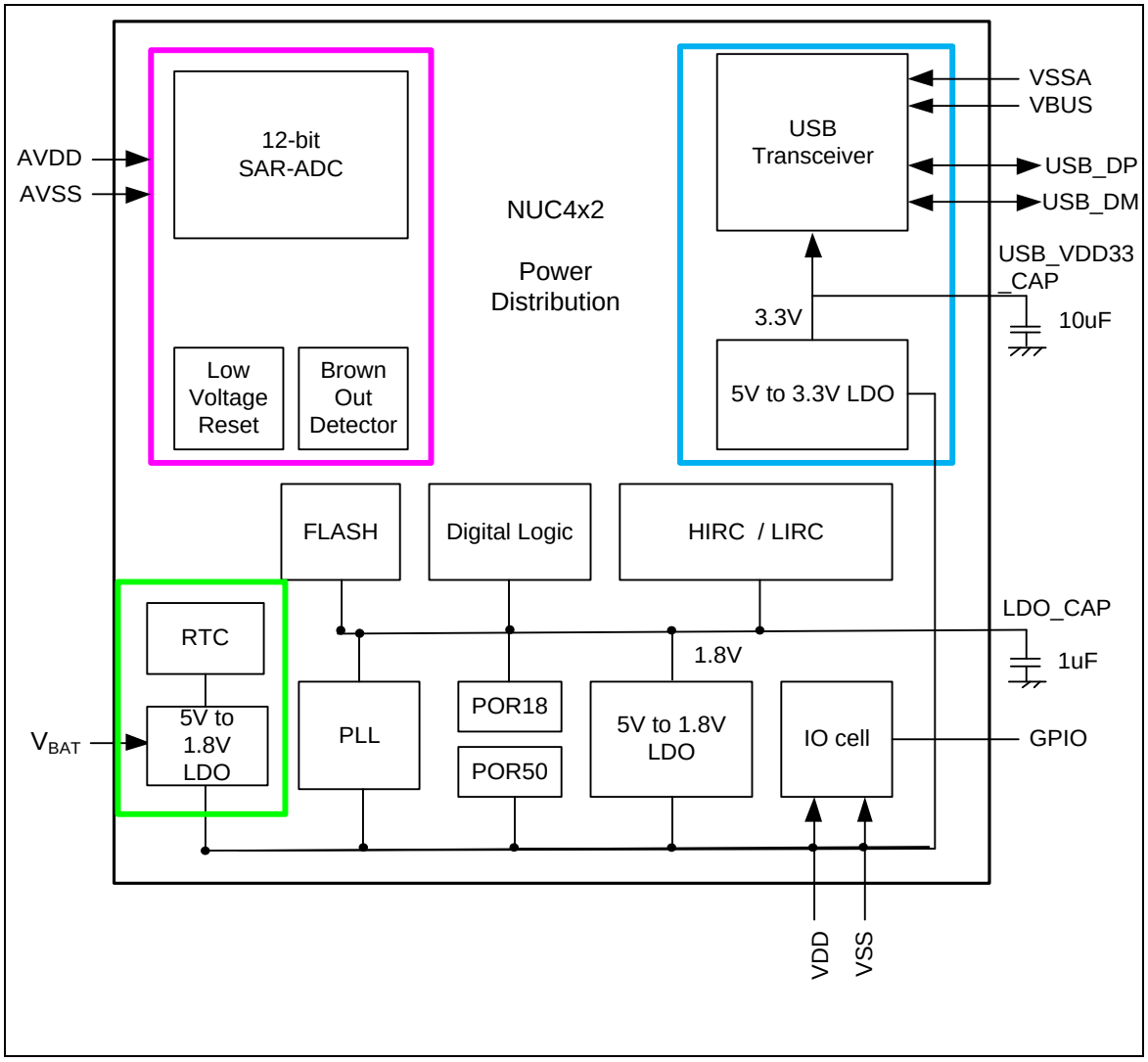


Figure 6.2-1 NuMicro™ NUC442/NUC472 Power Distribution Diagram

6.2.4 System Memory Map

The NUC442/NUC472 series provides 4G-byte addressing space. The memory locations assigned to each on-chip controllers are shown in the following table. The detailed register definition, memory space, and programming detailed will be described in the following sections for each on-chip peripherals. The NUC442/NUC472 series only supports little-endian data format.

Address Space	Token	Controllers
Flash and SRAM Memory Space		
0x0000_0000 – 0x0007_FFFF	FLASH_BA	FLASH Memory Space (512KB)
0x2000_0000 – 0x2000_FFFF	SRAM_BA	SRAM Memory Space (64KB)
0x6000_0000 – 0x6FFF_FFFF	EXTMEM_BA	External Memory Space (256MB)
Peripheral Controllers Space (0x4000_0000 – 0x400F_FFFF)		
0x4000_0000 – 0x4000_01FF	SYS_BA	System Control Registers
0x4000_0200 – 0x4000_02FF	CLK_BA	Clock Control Registers
0x4000_0300 – 0x4000_03FF	NMI_BA	Interrupt Multiplexer Control Registers
0x4000_4000 – 0x4000_4FFF	GPIO_BA	GPIO Control Registers
0x4000_8000 – 0x4000_8FFF	PDMA_BA	Peripheral DMA Control Registers
0x4000_9000 – 0x4000_9FFF	USBH_BA	USB Host Control Registers
0x4000_B000 – 0x4000_BFFF	EMAC_BA	Ethernet MAC Control Registers (NUC472 only)
0x4000_C000 – 0x4000_CFFF	FMC_BA	Flash Memory Control Registers
0x4000_D000 – 0x4000_DFFF	SDH_BA	SD HOST Control Registers
0x4001_0000 – 0x4001_0FFF	EBI_BA	External Bus Interface Control Registers
0x4001_9000 – 0x4001_9FFF	USBD_BA	USB device Control Registers
0x4003_0000 – 0x4003_0FFF	CAP_BA	Image Capture interface Registers
0x4003_1000 – 0x4003_1FFF	CRC_BA	CRC Generator Registers
0x5000_8000 – 0x5000_FFFF	CRYP_BA	Cryptographic Accelerator Registers
APB Controllers Space (0x4000_0000 ~ 0x400F_FFFF)		
0x4004_0000 – 0x4004_0FFF	WDT_BA	Watchdog Timer Control Registers
0x4004_1000 – 0x4004_1FFF	RTC_BA	Real Time Clock (RTC) Control Register
0x4004_3000 – 0x4004_3FFF	ADC_BA	Analog-Digital-Converter (ADC) Control Registers
0x4004_4000 – 0x4004_4FFF	EADC_BA	Enhance Analog-Digital-Converter (ADC) Control Registers
0x4004_5000 – 0x4004_5FFF	ACMP_BA	Analog Comparator Control Registers
0x4004_6000 – 0x4004_6FFF	OPA_BA	OP Amplifier Control Registers
0x4004_8000 – 0x4004_8FFF	I2S0_BA	I^2S0 Interface Control Registers
0x4004_9000 – 0x4004_9FFF	I2S1_BA	I^2S1 Interface Control Registers
0x4004_D000 – 0x4004_DFFF	OTG_BA	USB OTG

0x4005_0000 – 0x4005_0FFF	TMR01_BA	Timer0/Timer1 Control Registers
0x4005_1000 – 0x4005_1FFF	TMR23_BA	Timer2/Timer3 Control Registers
0x4005_8000 – 0x4005_8FFF	PWM0_BA	PWM0_0/1/2/3/4/5 Control Registers
0x4005_9000 – 0x4005_9FFF	PWM1_BA	PWM1_0/1/2/3/4/5 Control Registers
0x4005_C000 – 0x4005_CFFF	EPWM0_BA	Enhanced PWM0_0/1/2/3/4/5 Control Registers
0x4005_D000 – 0x4005_DFFF	EPWM1_BA	Enhanced PWM1_0/1/2/3/4/5 Control Registers
0x4006_0000 – 0x4006_0FFF	SPI0_BA	SPI0 with Master/Slave function Control Registers
0x4006_1000 – 0x4006_1FFF	SPI1_BA	SPI1 with Master/Slave function Control Registers
0x4006_2000 – 0x4006_2FFF	SPI2_BA	SPI2 with Master/Slave function Control Registers
0x4006_3000 – 0x4006_3FFF	SPI3_BA	SPI3 with Master/Slave function Control Registers
0x4007_0000 – 0x4007_0FFF	UART0_BA	UART0 Control Registers
0x4007_1000 – 0x4007_1FFF	UART1_BA	UART1 Control Registers
0x4007_2000 – 0x4007_2FFF	UART2_BA	UART2 Control Registers
0x4007_3000 – 0x4007_3FFF	UART3_BA	UART3 Control Registers
0x4007_4000 – 0x4007_4FFF	UART4_BA	UART4 Control Registers
0x4007_5000 – 0x4007_5FFF	UART5_BA	UART5 Control Registers
0x4008_0000 – 0x4008_0FFF	I2C0_BA	I^2C0 Interface Control Registers
0x4008_1000 – 0x4008_1FFF	I2C1_BA	I^2C1 Interface Control Registers
0x4008_2000 – 0x4008_2FFF	I2C2_BA	I^2C2 Interface Control Registers
0x4008_3000 – 0x4008_3FFF	I2C3_BA	I^2C3 Interface Control Registers
0x4008_4000 – 0x4008_4FFF	I2C4_BA	I^2C4 Interface Control Registers
0x4009_0000 – 0x4009_0FFF	SC0_BA	Smartcard0 Control Registers
0x4009_1000 – 0x4009_1FFF	SC1_BA	Smartcard1 Control Registers
0x4009_2000 – 0x4009_2FFF	SC2_BA	Smartcard2 Control Registers
0x4009_3000 – 0x4009_3FFF	SC3_BA	Smartcard3 Control Registers
0x4009_4000 – 0x4009_4FFF	SC4_BA	Smartcard4 Control Registers
0x4009_5000 – 0x4009_5FFF	SC5_BA	Smartcard5 Control Registers
0x400A_0000 – 0x400A_0FFF	CAN0_BA	CAN0 Bus Control Registers
0x400A_1000 – 0x400A_1FFF	CAN1_BA	CAN1 Bus Control Registers
0x400B_0000 – 0x400B_0FFF	QEI0_BA	Quadrature Encoder Interface 0 Control Registers
0x400B_1000 – 0x400B_1FFF	QEI1_BA	Quadrature Encoder Interface 1 Control Registers
0x400B_0000 – 0x400B_0FFF	ECAP0_BA	Capture Engine 0 Control Registers
0x400B_1000 – 0x400B_1FFF	ECAP1_BA	Capture Engine 1 Control Registers
0x400E_0000 – 0x400E_0FFF	PS2_BA	PS/2 interface Control Registers
System Controllers Space		

(0xE000_E000 ~ 0xE000_EFFF)		
0xE000_E010 – 0xE000_E0FF	SCS_BA	System Timer Control Registers
0xE000_E100 – 0xE000_ECFF	SCS_BA	External Interrupt Controller Control Registers
0xE000_ED00 – 0xE000_ED8F	SCS_BA	System Control Registers

Table 6.2-1 Address Space Assignments for On-Chip Controllers

6.2.5 System Control Registers

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
SYS Base Address: **SYS_BA = 0x4000_0000**				
SYS_PDID	SYS_BA+0x00	R	Part Device Identification Number Register	0x0014_0018[1]
SYS_RSTSTS	SYS_BA+0x04	R/W	System Reset Source Register	0x0000_014B
SYS_IPRST0	SYS_BA+0x08	R/W	Peripheral Controller Reset Control Register 1	0x0000_0000
SYS_IPRST1	SYS_BA+0x0C	R/W	Peripheral Controller Reset Control Register 2	0x0000_0000
SYS_IPRST2	SYS_BA+0x10	R/W	Peripheral Controller Reset Control Register 3	0x0000_0000
SYS_BODCTL	SYS_BA+0x18	R/W	Brown-out Detector Control Register	0x0000_008X
SYS_TEMPCTL	SYS_BA+0x1C	R/W	Temperature Sensor Control Register	0x0000_0000
SYS_VCID	SYS_BA+0x20	R	Hardware Version Control Register	0x0000_0000
SYS_PORCTL	SYS_BA+0x24	R/W	Power-On-Reset Controller Register	0x0000_0000
SYS_VREFCTL	SYS_BA+0x28	R/W	ADC VREF Control Register	0x0000_0000
SYS_USBPHY	SYS_BA+0x2C	R/W	USB PHY Control Register	0x0000_000X
SYS_GPA_MFPL	SYS_BA+0x30	R/W	Port A Low Byte Multi-function Control Register	0x0000_0000
SYS_GPA_MFPH	SYS_BA+0x34	R/W	Port A High Byte Multi-function Control Register	0x0000_0000
SYS_GPB_MFPL	SYS_BA+0x38	R/W	Port B Low Byte Multi-function Control Register	0x0000_0000
SYS_GPB_MFPH	SYS_BA+0x3C	R/W	Port B High Byte Multi-function Control Register	0x0000_0000
SYS_GPC_MFPL	SYS_BA+0x40	R/W	Port C Low Byte Multi-function Control Register	0x0000_0000
SYS_GPC_MFPH	SYS_BA+0x44	R/W	Port C High Byte Multi-function Control Register	0x0000_0000
SYS_GPD_MFPL	SYS_BA+0x48	R/W	Port D Low Byte Multi-function Control Register	0x0000_0000
SYS_GPD_MFPH	SYS_BA+0x4C	R/W	Port D High Byte Multi-function Control Register	0x0000_0000
SYS_GPE_MFPL	SYS_BA+0x50	R/W	Port E Low Byte Multi-function Control Register	0x0000_0000
SYS_GPE_MFPH	SYS_BA+0x54	R/W	Port E High Byte Multi-function Control Register	0x0000_0000
SYS_GPF_MFPL	SYS_BA+0x58	R/W	Port F Low Byte Multi-function Control Register	0x0000_0000
SYS_GPF_MFP	SYS_BA+0x5C	R/W	Port F High Byte Multi-function Control Register	0x0000_0000

H				
SYS_GPG_MFPL	SYS_BA+0x60	R/W	Port G Low Byte Multi-function Control Register	0x0000_0000
SYS_GPG_MFPH	SYS_BA+0x64	R/W	Port G High Byte Multi-function Control Register	0xXXXX_1100
SYS_GPH_MFPL	SYS_BA+0x68	R/W	Port H Low Byte Multi-function Control Register	0x0000_0000
SYS_GPH_MFPH	SYS_BA+0x6C	R/W	Port H High Byte Multi-function Control Register	0x0000_0000
SYS_GPI_MFPL	SYS_BA+0x70	R/W	Port I Low Byte Multi-function Control Register	0x0000_0000
SYS_GPI_MFPH	SYS_BA+0x74	R/W	Port I High Byte Multi-function Control Register	0x0000_0000
SYS_SRAM_INTCTL	SYS_BA+0xC0	R/W	SRAM Failed Interrupt Enable Control Register	0x0000_0000
SYS_SRAM_STATUS	SYS_BA+0xC4	R/W	SRAM Parity Check Error Flag	0x0000_0000
SYS_SRAM0_ERRADDR	SYS_BA+0xC8	R	SRAM Parity Check Error First Address1	0x0000_0000
SYS_SRAM1_ERRADDR	SYS_BA+0xCC	R	SRAM Parity Check Error First Address2	0x0000_0000
SYS_IRCTCTL	SYS_BA+0xF0	R/W	IRC Trim Control Register	0x0000_0000
SYS_IRCTIEN	SYS_BA+0xF4	R/W	IRC Trim Interrupt Enable Control Register	0x0000_0000
SYS_IRCTISTS	SYS_BA+0xF8	R/W	IRC Trim Interrupt Status Register	0x0000_0000
SYS_REGLCTL	SYS_BA+0x100	R/W	Register Write-Protection Control Register	0x0000_0000

Part Device ID Code Register (SYS_PDID)

Register	Offset	R/W	Description	Reset Value
SYS_PDID	SYS_BA+0x00	R	Part Device Identification Number Register	0x0014_0018[1]

[1] Every part number has a unique default reset value.

31	30	29	28	27	26	25	24
Part Number [31:24]							
23	22	21	20	19	18	17	16
Part Number [23:16]							
15	14	13	12	11	10	9	8
Part Number [15:8]							
7	6	5	4	3	2	1	0
Part Number [7:0]							

Bits	Description	
[31:0]	PDID	**Part Device Identification Number** This register reflects device part number code. S/W can read this register to identify which device is used.

Part Number	PID	Part Number	PID
NUC472HI8AE	0x00047201	NUC442JI8AE	0x00044201
NUC472HH8AE	0x00047202	NUC442JH8AE	0x00044202
NUC472HG8AE	0x00047203	NUC442JG8AE	0x00044203
NUC472JI8AE	0x00047204	NUC442KI8AE	0x00044204
NUC472JH8AE	0x00047205	NUC442KH8AE	0x00044205
NUC472JG8AE	0x00047206	NUC442KG8AE	0x00044206
NUC472KI8AE	0x00047207	NUC442VI8AE	0x00044207
NUC472KH8AE	0x00047208	NUC442VH8AE	0x00044208
NUC472KG8AE	0x00047209	NUC442VG8AE	0x00044209
NUC472VI8AE	0x00047210	NUC442RI8AE	0x00044210
NUC472VH8AE	0x00047211	NUC442RH8AE	0x00044211
NUC472VG8AE	0x00047212	NUC442RG8AE	0x00044212

System Reset Source Register (SYS_RSTSTS)

This register provides specific information for software to identify this chip's reset source from last operation.

Register	Offset	R/W	Description	Reset Value
SYS_RSTSTS	SYS_BA+0x04	R/W	System Reset Source Register	0x0000_014B

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
CPURF	Reserved	SYSRF	BODRF	LVRF	WDTRF	PINRF	PORF

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**CPURF**	**CPU Reset Flag** The CPU reset flag is set by hardware if software writes CPURST (SYS_IPRST0[1]) 1 to reset Cortex®-M4 Core and Flash Memory Controller (FMC). 0 = No reset from CPU. 1 = The Cortex®-M4 Core and FMC are reset by software setting CPURST to 1. **Note:** Write to clear this bit to 0.
[6]	**Reserved**	Reserved.
[5]	**SYSRF**	**System Reset Flag** The system reset flag is set by the "Reset Signal" from the Cortex®-M4 Core to indicate the previous reset source. 0 = No reset from Cortex®-M4. 1 = The Cortex®-M4 had issued the reset signal to reset the system by writing 1 to the bit SYSRESETREQ(AIRCR[2], Application Interrupt and Reset Control Register, address = 0xE000ED0C) in system control registers of Cortex®-M4 core. **Note:** Write 1 to clear this bit to 0.
[4]	**BODRF**	**BOD Reset Flag** The BOD reset flag is set by the "Reset Signal" from the Brown-Out Detector to indicate the previous reset source. 0 = No reset from BOD. 1 = The BOD had issued the reset signal to reset the system. **Note:** Write 1 to clear this bit to 0.

Bits	Description	
[3]	**LVRF**	**LVR Reset Flag** The LVR reset flag is set by the “Reset Signal” from the Low Voltage Reset Controller to indicate the previous reset source. 0 = No reset from LVR. 1 = LVR controller had issued the reset signal to reset the system. **Note:** Write 1 to clear this bit to 0.
[2]	**WDTRF**	**WDT Reset Flag** The WDT reset flag is set by the “Reset Signal” from the Watchdog Timer or Window Watchdog Timer to indicate the previous reset source. 0 = No reset from watchdog timer or window watchdog timer. 1 = The watchdog timer or window watchdog timer had issued the reset signal to reset the system. **Note1:** Write 1 to clear this bit to 0. **Note2**: Watchdog Timer register RSTF(WDT_CTL[2]) bit is set if the system has been reset by WDT time-out reset. Window Watchdog Timer register WWDTRF(WWDT_STATUS[1]) bit is set if the system has been reset by WWDT time-out reset.
[1]	**PINRF**	**nRESET Pin Reset Flag** The nRESET pin reset flag is set by the “Reset Signal” from the nRESET Pin to indicate the previous reset source. 0 = No reset from nRESET pin. 1 = Pin nRESET had issued the reset signal to reset the system. **Note:** Write 1 to clear this bit to 0.
[0]	**PORF**	**POR Reset Flag** The POR reset flag is set by the “Reset Signal” from the Power-on Reset (POR) Controller or bit CHIPRST (SYS_IPRST0[0]) to indicate the previous reset source. 0 = No reset from POR or CHIPRST. 1 = Power-on Reset (POR) or CHIPRST had issued the reset signal to reset the system. **Note:** Write 1 to clear this bit to 0.

Peripheral Reset Control Register1 (SYS_IPRST0)-AHB BUS

Register	Offset	R/W	Description	Reset Value
SYS_IPRST0	SYS_BA+0x08	R/W	Peripheral Controller Reset Control Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved			CRYPTORST	Reserved			CAPRST
7	6	5	4	3	2	1	0
CRCRST	SDHRST	EMACRST	USBHRST	EBIRST	PDMARST	CPURST	CHIPRST

Bits	Description	
[31:13]	**Reserved**	Reserved.
[12]	**CRYPTORST**	**CRYPTO Controller Reset (Write Protect)** Setting this bit to 1 will generate a reset signal to the CRYPTO controller. User needs to set this bit to 0 to release from the reset state. This bit is a write protected bit, It means programming this bit needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register SYS_REGLCTL at address GCR_BA+0x100 0 = CRYPTO controller normal operation. 1 = CRYPTO controller reset.
[11:9]	**Reserved**	Reserved.
[8]	**CAPRST**	**Image Capture Controller Reset (Write Protect)** Setting this bit to 1 will generate a reset signal to the CAP controller. User needs to set this bit to 0 to release from the reset state. This bit is a write protected bit, It means programming this bit needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Reference the register SYS_REGLCTL at address GCR_BA+0x100 0 = CAP controller normal operation. 1 = CAP controller reset.
[7]	**CRCRST**	**CRC Controller Reset (Write Protect)** Setting this bit to 1 will generate a reset signal to the CRC controller. User needs to set this bit to 0 to release from the reset state. This bit is a write protected bit, It means programming this bit needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Reference the register SYS_REGLCTL at address GCR_BA+0x100 0 = CRC controller normal operation. 1 = CRC controller reset.
[6]	**SDHRST**	**SD HOST Controller Reset (Write Protect)** Setting this bit to 1 will generate a reset signal to the SD HOST controller. User needs to set this bit to 0 to release from the reset state.

		This bit is a write protected bit, It means programming this bit needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register SYS_REGLCTL at address GCR_BA+0x100 0 = SD HOST controller normal operation. 1 = SD HOST controller reset.
[5]	**EMACRST**	**EMAC Controller Reset (Write Protect)** Setting this bit to 1 will generate a reset signal to the EMAC controller. User needs to set this bit to 0 to release from the reset state. This bit is the protected bit, It means programming this bit needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register SYS_REGLCTL at address GCR_BA+0x100 0 = EMAC controller normal operation. 1 = EMAC controller reset.
[4]	**USBHRST**	**USBH Controller Reset (Write Protect)** Setting this bit to 1 will generate a reset signal to the HSB HOST controller. User needs to set this bit to 0 to release from the reset state. This bit is a write protected bit, It means programming this bit needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register SYS_REGLCTL at address GCR_BA+0x100 0 = USBH controller normal operation. 1 = USBH controller reset.
[3]	**EBIRST**	**EBI Controller Reset (Write Protect)** Setting this bit to 1 will generate a reset signal to the EBI. User needs to set this bit to 0 to release from the reset state. This bit is a write protected bit, It means programming this bit needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register SYS_REGLCTL at address GCR_BA+0x100 0 = EBI controller normal operation. 1 = EBI controller reset.
[2]	**PDMARST**	**PDMA Controller Reset (Write Protect)** Setting this bit to 1 will generate a reset signal to the PDMA. User needs to set this bit to 0 to release from reset state. This bit is a write protected bit, It means programming this bit needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register SYS_REGLCTL at address GCR_BA+0x100. 0 = PDMA controller normal operation. 1 = PDMA controller reset.
[1]	**CPURST**	**Processor Core One-Shot Reset (Write Protect)** Setting this bit will only reset the processor core and Flash Memory Controller(FMC), and this bit will automatically return to 0 after the 2 clock cycles This bit is a write protected bit, It means programming this bit needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register SYS_REGLCTL at address GCR_BA+0x100 0 = Processor core normal operation. 1 = Processor core one-shot reset.
[0]	**CHIPRST**	**Chip One-Shot Reset (Write Protect)** Setting this bit will reset the whole chip, including Processor core and all peripherals, and this bit will automatically return to 0 after the 2 clock cycles. The CHIPRST is same as the POR reset, all the chip controllers is reset and the chip setting from flash are also reload. This bit is a write protected bit, which means programming this bit needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register

		SYS_REGLCTL at address GCR_BA+0x100 0 = Chip normal operation. 1 = Chip one shot reset.

Peripheral Reset Control Register2 (SYS_IPRST1) APB BUS

Setting these bits 1 will generate asynchronous reset signals to the corresponding IP controller. Users need to set these bits to 0 to release corresponding IP controller from reset state.

Register	Offset	R/W	Description	Reset Value
SYS_IPRST1	SYS_BA+0x0C	R/W	Peripheral Controller Reset Control Register 2	0x0000_0000

31	30	29	28	27	26	25	24
PS2RST	I2S1RST	I2SRST	EADCRST	USBDRST	OTGRST	CAN1RST	CAN0RST
23	22	21	20	19	18	17	16
Reserved		UART5RST	UART4RST	UART3RST	UART2RST	UART1RST	UART0RST
15	14	13	12	11	10	9	8
SPI3RST	SPI2RST	SPI1RST	SPI0RST	I2C3RST	I2C2RST	I2C1RST	I2C0RST
7	6	5	4	3	2	1	0
ACMPRST	Reserved	TMR3RST	TMR2RST	TMR1RST	TMR0RST	GPIORST	Reserved

Bits	Description	
[31]	PS2RST	**PS/2 Controller Reset** 0 = PS/2 controller normal operation. 1 = PS/2 controller reset.
[30]	I2S1RST	**I²S1 Controller Reset** 0 = I²S1 controller normal operation. 1 = I²S1 controller reset.
[29]	I2SRST	**I²S Controller Reset** 0 = I²S controller normal operation. 1 = I²S controller reset.
[28]	EADCRST	**ADC Controller Reset** 0 = ADC controller normal operation. 1 = ADC controller reset.
[27]	USBDRST	**USB Device Controller Reset** 0 = USB device controller normal operation. 1 = USB device controller reset.
[26]	Reserved	Reserved.
[25]	CAN1RST	**CAN1 Controller Reset** 0 = CAN1 controller normal operation. 1 = CAN1 controller reset.
[24]	CAN0RST	**CAN0 Controller Reset** 0 = CAN0 controller normal operation. 1 = CAN0 controller reset.

[23:22]	**Reserved**	Reserved.
[21]	**UART5RST**	**UART2 Controller Reset** 0 = UART5 controller normal operation. 1 = UART5 controller reset.
[20]	**UART4RST**	**UART4 Controller Reset** 0 = UART4 controller normal operation. 1 = UART4 controller reset.
[19]	**UART3RST**	**UART3 Controller Reset** 0 = UART3 controller normal operation. 1 = UART3 controller reset.
[18]	**UART2RST**	**UART2 Controller Reset** 0 = UART2 controller normal operation. 1 = UART2 controller reset.
[17]	**UART1RST**	**UART1 Controller Reset** 0 = UART1 controller normal operation. 1 = UART1 controller reset.
[16]	**UART0RST**	**UART0 Controller Reset** 0 = UART0 controller normal operation. 1 = UART0 controller reset.
[15]	**SPI3RST**	**SPI3 Controller Reset** 0 = SPI3 controller normal operation. 1 = SPI3 controller reset.
[14]	**SPI2RST**	**SPI2 Controller Reset** 0 = SPI2 controller normal operation. 1 = SPI2 controller reset.
[13]	**SPI1RST**	**SPI1 Controller Reset** 0 = SPI1 controller normal operation. 1 = SPI1 controller reset.
[12]	**SPI0RST**	**SPI0 Controller Reset** 0 = SPI0 controller normal operation. 1 = SPI0 controller reset.
[11:10]	**Reserved**	Reserved.
[9]	**I2C1RST**	**I^2C1 Controller Reset** 0 = I^2C1 controller normal operation. 1 = I^2C1 controller reset.
[8]	**I2C0RST**	**I^2C0 Controller Reset** 0 = I^2C0 controller normal operation. 1 = I^2C0 controller reset.
[7]	**ACMPRST**	**Analog Comparator Controller Reset** 0 = Analog Comparator controller normal operation. 1 = Analog Comparator controller reset.
[6]	**Reserved**	Reserved.

[5]	**TMR3RST**	**Timer3 Controller Reset** 0 = Timer3 controller normal operation. 1 = Timer3 controller reset.
[4]	**TMR2RST**	**Timer2 Controller Reset** 0 = Timer2 controller normal operation. 1 = Timer2 controller reset.
[3]	**TMR1RST**	**Timer1 Controller Reset** 0 = Timer1 controller normal operation. 1 = Timer1 controller reset.
[2]	**TMR0RST**	**Timer0 Controller Reset** 0 = Timer0 controller normal operation. 1 = Timer0 controller reset.
[1]	**GPIORST**	**GPIO Controller Reset** 0 = GPIO controller normal operation. 1 = GPIO controller reset.
[0]	**Reserved**	Reserved.

Peripheral Reset Control Register 3 (SYS_IPRST2) APB BUS

Setting these bits 1 will generate asynchronous reset signals to the corresponding IP controller. Users need to set these bits to 0 to release corresponding IP controller from reset state.

Register	Offset	R/W	Description	Reset Value
SYS_IPRST2	SYS_BA+0x10	R/W	Peripheral Controller Reset Control Register 3	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
QEI1RST	QEI0RST	Reserved				PWM1RST	PWM0RST
15	14	13	12	11	10	9	8
Reserved							I2C4RST
7	6	5	4	3	2	1	0
Reserved		SC5RST	SC4RST	SC3RST	SC2RST	SC1RST	SC0RST

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23]	**QEI1RST**	**QEI1 Controller Reset** 0 = QEI1 controller normal operation. 1 = QEI1 controller reset.
[22]	**QEI0RST**	**QEI0 Controller Reset** 0 = QEI0 controller normal operation. 1 = QEI0 controller reset.
[21:18]	**Reserved**	Reserved.
[17]	**PWM1RST**	**PWM1 Controller Reset** 0 = PWM1 controller normal operation. 1 = PWM1 controller reset.
[16]	**PWM0RST**	**PWM0 Controller Reset** 0 = PWM0 controller normal operation. 1 = PWM0 controller reset.
[15:9]	**Reserved**	Reserved.
[8]	**I2C4RST**	**I^2C4 Controller Reset** 0 = I^2C4 controller normal operation. 1 = I^2C4 controller reset.
[7:6]	**Reserved**	Reserved.
[5]	**SC5RST**	**SC5 Controller Reset** 0 = SC5 controller normal operation. 1 = SC5 controller reset.

[4]	**SC4RST**	**SC4 Controller Reset** 0 = SC4 controller normal operation. 1 = SC4 controller reset.
[3]	**SC3RST**	**SC3 Controller Reset** 0 = SC3 controller normal operation. 1 = SC3 controller reset.
[2]	**SC2RST**	**SC2 Controller Reset** 0 = SC2 controller normal operation. 1 = SC2 controller reset.
[1]	**SC1RST**	**SC1 Controller Reset** 0 = SC1 controller normal operation. 1 = SC1 controller reset.
[0]	**SC0RST**	**SC0 Controller Reset** 0 = SC0 controller normal operation. 1 = SC0 controller reset.

Brown-out Detector Control Register (SYS_BODCTL)

Partial of the SYS_BODCTL control registers values are initiated by the flash configuration and partial bits are write-protected bit. Programming write-protected bits needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register SYS_REGLCTL at address GCR_BA+0x100.

Register	Offset	R/W	Description	Reset Value
SYS_BODCTL	SYS_BA+0x18	R/W	Brown-out Detector Control Register	0x0000_008X

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
LVREN	**BODOUT**	**BODLPM**	**BODINTF**	**BODRSTEN**	**BODVL**		**BODEN**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**LVREN**	**Low Voltage Reset Enable Bit (Write Protect)** The LVR function reset the chip when the input power voltage is lower than LVR circuit setting. LVR function is enabled in default. 0 = Low Voltage Reset function Disabled. 1 = Low Voltage Reset function Enabled – After enabling the bit, the LVR function will be active with 100uS delay for LVR output stable (default). This bit is the protected bit, which means programming this needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register SYS_REGLCTL at address GCR_BA+0x100
[6]	**BODOUT**	**Brown-Out Detector Output Status** 0 = Brown-out Detector output status is 0. It means the detected voltage is higher than BODVL setting or BODEN is 0. 1 = Brown-out Detector output status is 1. It means the detected voltage is lower than BODVL setting. If the BODEN is 0, BOD function disabled , this bit always responds 0000
[5]	**BODLPM**	**Brown-Out Detector Low Power Mode (Write Protect)** The BOD consumes about 100uA in normal mode, the low power mode can reduce the current to about 1/10 but slow the BOD response. 0 = BOD operate in normal mode (default). 1 = BOD Low Power mode Enabled. This bit is the protected bit, which means programming this needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register SYS_REGLCTL at address GCR_BA+0x100.

Bits	Description	
[4]	**BODINTF**	**Brown-Out Detector Interrupt Flag** 0 = Brown-out Detector does not detect any voltage draft at VDD down through or up through the voltage of BODVL setting. 1 = When Brown-out Detector detects the VDD is dropped down through the voltage of BODVL setting or the VDD is raised up through the voltage of BODVL setting, this bit is set to 1 and the brown-out interrupt is requested if brown-out interrupt is enabled. **Note:** Write 1 to clear this bit to 0.
[3]	**BODRSTEN**	**Brown-Out Reset Enable Bit (Write Protect)** While the Brown-out Detector function is enabled (BODEN high) and BOD reset function is enabled (BODRSTEN high), BOD will assert a signal to reset chip when the detected voltage is lower than the threshold (BODOUT high). While the BOD function is enabled (BODEN high) and BOD interrupt function is enabled (BODRSTEN low), BOD will assert an interrupt if BODOUT is high. BOD interrupt will keep till to the BODEN set to 0. BOD interrupt can be blocked by disabling the NVIC BOD interrupt or disabling BOD function (set BODEN low). The default value is set by flash controller user configuration register config0 bit[20]. 0 = Brown-out “INTERRUPT” function Enabled. 1 = Brown-out “RESET” function Enabled. This bit is the protected bit, which means programming this needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register SYS_REGLCTL at address GCR_BA+0x100.
[2:1]	**BODVL**	**Brown-Out Detector Threshold Voltage Selection (Write Protect)** The default value is set by flash controller user configuration register config0 bit[22:21] Relationship between BODVL and Brown-out voltage listed below: 00 = 2.2V. 01 = 2.7V. 10 = 3.7V. 11 = 4.5V. This bit is the protected bit, which means programming this needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register SYS_REGLCTL at address GCR_BA+0x100.
[0]	**BODEN**	**Brown-Out Detector Enable Bit (Write Protect)** The default value is set by flash controller user configuration register config0 bit[23] 0 = Brown-out Detector function Disabled. 1 = Brown-out Detector function Enabled. This bit is the protected bit, which means programming this needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register SYS_REGLCTL at address GCR_BA+0x100.

Temperature Sensor Control Register (SYS_TEMPCTL)

Register	Offset	R/W	Description	Reset Value
SYS_TEMPCTL	SYS_BA+0x1C	R/W	Temperature Sensor Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							VTEMPEN

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**VTEMPEN**	**Temperature Sensor Enable Bit** This bit is used to enable/disable temperature sensor function. 0 = Temperature sensor function Disabled (default). 1 = Temperature sensor function Enabled. After this bit is set to 1, the value of temperature sensor output can be obtained from ADC conversion result. Please refer to ADC function chapter for details.

Version Control ID Register (SYS_VCID)

Register	Offset	R/W	Description	Reset Value
SYS_VCID	SYS_BA+0x20	R	Hardware Version Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VCID[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VCID[7:0]							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**VCID**	**Hardware Version Control (Ready Only)** These registers repress hardware version. These bits are the read protected bits. It means programming this needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register SYS_REGLCTL at address GCR_BA+0x100.

Power-on-Reset Control Register (SYS_PORCTL)

Register	Offset	R/W	Description	Reset Value
SYS_PORCTL	SYS_BA+0x24	R/W	Power-On-Reset Controller Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
POROFF[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
POROFF[7:0]							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**POROFF**	**Power-On-Reset Enable Bit (Write Protect)** When power on, the POR circuit generates a reset signal to reset the whole chip function, but noise on the power may cause the POR active again. User can disable internal POR circuit to avoid unpredictable noise to cause chip reset by writing 0x5AA5 to this field. The POR function will be active again when this field is set to another value or chip is reset by other reset source, including: /RESET, Watch dog, LVR reset, BOD reset, ICE reset command and the software-chip reset function This bit is the protected bit, which means programming this needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register SYS_REGLCTL at address GCR_BA+0x100.

VREF Control Register (SYS_VREFCTL)

Register	Offset	R/W	Description	Reset Value
SYS_VREFCTL	SYS_BA+0x28	R/W	ADC VREF Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved						PWMSYNCMODE	ADCMODESEL
7	6	5	4	3	2	1	0
Reserved			VREFCTL				

Bits	Description	
[31:10]	**Reserved**	Reserved.
[9]	**PWMSYNCMODE**	**PWM SYNC MODE (Write Protect)** 0 = PWM SYNC MODE Disabled; PWM engine clock can different with HCLK. 1 = PWM SYNC MODE Enabled; PWM engine clock is same as HCLK.
[8]	**ADCMODESEL**	**ADC IP Selection (Write Protect)** 0 = ADC mode. 1 = E ADC mode.
[7:5]	**Reserved**	Reserved.
[4:0]	**VREFCTL**	**Vref Control Bits (Write Protect)** 00011=Vref is internal 2.65V. 00111=Vref is internal 2.048V. 01011=Vref is internal 3.072V. 01111=Vref is internal 4.096V. 10000=Vref is from AVDD. Others=Reserved.

USBPHY Control Register (SYS_USBPHY)

Register	Offset	R/W	Description	Reset Value
SYS_USBPHY	SYS_BA+0x2C	R/W	USB PHY Control Register	0x0000_000X

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							LDO33EN
7	6	5	4	3	2	1	0
Reserved						USBROLE	

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**LDO33EN**	**LDO33 Enable Bit (Write Protect)** 0 = USB LDO33 Disabled. 1 = USB LDO33 Enabled.
[7:0]	**Reserved**	Reserved.
[1:0]	**USBROLE**	**USB Role Configuration (Write Protect)** USB role configuration can be from ROMMAP or software setting if software setting option, controlled by ROMMAP, is enabled. 00 = Standard USB device. 01 = Standard USB host. 10 = ID dependent device. 11 = On-The-Go device.

Multi-function GPIOA Low Byte Control Register (SYS_GPA_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPA_MFPL	SYS_BA+0x30	R/W	Port A Low Byte Multi-function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PA7MFP				PA6MFP			
23	22	21	20	19	18	17	16
PA5MFP				PA4MFP			
15	14	13	12	11	10	9	8
PA3MFP				PA2MFP			
7	6	5	4	3	2	1	0
PA1MFP				PA0MFP			

Bits	Description	
[31:28]	PA7MFP[3:0]	PA.7 Multi-function Pin Selection
[27:24]	PA6MFP[3:0]	PA.6 Multi-function Pin Selection
[23:20]	PA5MFP[3:0]	PA.5 Multi-function Pin Selection
[19:16]	PA4MFP[3:0]	PA.4 Multi-function Pin Selection
[15:12]	PA3MFP[3:0]	PA.3 Multi-function Pin Selection
[11:8]	PA2MFP[3:0]	PA.2 Multi-function Pin Selection
[7:4]	PA1MFP[3:0]	PA.1 Multi-function Pin Selection
[3:0]	PA0MFP[3:0]	PA.0 Multi-function Pin Selection

Multi-function GPIOA High Byte Control Register (SYS_GPA_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPA_MFPH	SYS_BA+0x34	R/W	Port A High Byte Multi-function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PA15MFP				PA14MFP			
23	22	21	20	19	18	17	16
PA13MFP				PA12MFP			
15	14	13	12	11	10	9	8
PA11MFP				PA10MFP			
7	6	5	4	3	2	1	0
PA9MFP				PA8MFP			

Bits	Description	
[31:28]	PA15MFP[3:0]	PA.15 Multi-function Pin Selection
[27:24]	PA14MFP[3:0]	PA.14 Multi-function Pin Selection
[23:20]	PA13MFP[3:0]	PA.13 Multi-function Pin Selection
[19:16]	PA12MFP[3:0]	PA.12 Multi-function Pin Selection
[15:12]	PA11MFP[3:0]	PA.11 Multi-function Pin Selection
[11:8]	PA10MFP[3:0]	PA.10 Multi-function Pin Selection
[7:4]	PA9MFP[3:0]	PA.9 Multi-function Pin Selection
[3:0]	PA8MFP[3:0]	PA.8 Multi-function Pin Selection

Multi-function GPIOB Low Byte Control Register (SYS_GPB_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPB_MFPL	SYS_BA+0x38	R/W	Port B Low Byte Multi-function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PB7MFP				PB6MFP			
23	22	21	20	19	18	17	16
PB5MFP				PB4MFP			
15	14	13	12	11	10	9	8
PB3MFP				PB2MFP			
7	6	5	4	3	2	1	0
PB1MFP				PB0MFP			

Bits	Description	
[31:28]	PB7MFP[3:0]	PB.7 Multi-function Pin Selection
[27:24]	PB6MFP[3:0]	PB.6 Multi-function Pin Selection
[23:20]	PB5MFP[3:0]	PB.5 Multi-function Pin Selection
[19:16]	PB4MFP[3:0]	PB.4 Multi-function Pin Selection
[15:12]	PB3MFP[3:0]	PB.3 Multi-function Pin Selection
[11:8]	PB2MFP[3:0]	PB.2 Multi-function Pin Selection
[7:4]	PB1MFP[3:0]	PB.1 Multi-function Pin Selection
[3:0]	PB0MFP[3:0]	PB.0 Multi-function Pin Selection

Multi-function GPIOB high byte Control Register (SYS_GPB_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPB_MFPH	SYS_BA+0x3C	R/W	Port B High Byte Multi-function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PB15MFP				PB14MFP			
23	22	21	20	19	18	17	16
PB13MFP				PB12MFP			
15	14	13	12	11	10	9	8
PB11MFP				PB10MFP			
7	6	5	4	3	2	1	0
PB9MFP				PB8MFP			

Bits	Description	
[31:28]	PB15MFP[3:0]	PB.15 Multi-function Pin Selection
[27:24]	PB14MFP[3:0]	PB.14 Multi-function Pin Selection
[23:20]	PB13MFP[3:0]	PB.13 Multi-function Pin Selection
[19:16]	PB12MFP[3:0]	PB.12 Multi-function Pin Selection
[15:12]	PB11MFP[3:0]	PB.11 Multi-function Pin Selection
[11:8]	PB10MFP[3:0]	PB.10 Multi-function Pin Selection
[7:4]	PB9MFP[3:0]	PB.9 Multi-function Pin Selection
[3:0]	PB8MFP[3:0]	PB.8 Multi-function Pin Selection

Multi-function GPIOC Low Byte Control Register (SYS_GPC_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPC_MFPL	SYS_BA+0x40	R/W	Port C Low Byte Multi-function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PC7MFP				PC6MFP			
23	22	21	20	19	18	17	16
PC5MFP				PC4MFP			
15	14	13	12	11	10	9	8
PC3MFP				PC2MFP			
7	6	5	4	3	2	1	0
PC1MFP				PC0MFP			

Bits	Description	
[31:28]	PC7MFP[3:0]	PC.7 Multi-function Pin Selection
[27:24]	PC6MFP[3:0]	PC.6 Multi-function Pin Selection
[23:20]	PC5MFP[3:0]	PC.5 Multi-function Pin Selection
[19:16]	PC4MFP[3:0]	PC.4 Multi-function Pin Selection
[15:12]	PC3MFP[3:0]	PC.3 Multi-function Pin Selection
[11:8]	PC2MFP[3:0]	PC.2 Multi-function Pin Selection
[7:4]	PC1MFP[3:0]	PC.1 Multi-function Pin Selection
[3:0]	PC0MFP[3:0]	PC.0 Multi-function Pin Selection

Multi-function GPIOC high byte Control Register (SYS_GPC_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPC_MFPH	SYS_BA+0x44	R/W	Port C High Byte Multi-function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PC15MFP				PC14MFP			
23	22	21	20	19	18	17	16
PC13MFP				PC12MFP			
15	14	13	12	11	10	9	8
PC11MFP				PC10MFP			
7	6	5	4	3	2	1	0
PC9MFP				PC8MFP			

Bits	Description	
[31:28]	PC15MFP[3:0]	PC.15 Multi-function Pin Selection
[27:24]	PC14MFP[3:0]	PC.14 Multi-function Pin Selection
[23:20]	PC13MFP[3:0]	PC.13 Multi-function Pin Selection
[19:16]	PC12MFP[3:0]	PC.12 Multi-function Pin Selection
[15:12]	PC11MFP[3:0]	PC.11 Multi-function Pin Selection
[11:8]	PC10MFP[3:0]	PC.10 Multi-function Pin Selection
[7:4]	PC9MFP[3:0]	PC.9 Multi-function Pin Selection
[3:0]	PC8MFP[3:0]	PC.8 Multi-function Pin Selection

Multi-function GPIOD Low Byte Control Register (SYS_GPD_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPD_MFPL	SYS_BA+0x48	R/W	Port D Low Byte Multi-function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PD7MFP				PD6MFP			
23	22	21	20	19	18	17	16
PD5MFP				PD4MFP			
15	14	13	12	11	10	9	8
PD3MFP				PD2MFP			
7	6	5	4	3	2	1	0
PD1MFP				PD0MFP			

Bits	Description	
[31:28]	PD7MFP[3:0]	PD.7 Multi-function Pin Selection
[27:24]	PD6MFP[3:0]	PD.6 Multi-function Pin Selection
[23:20]	PD5MFP[3:0]	PD.5 Multi-function Pin Selection
[19:16]	PD4MFP[3:0]	PD.4 Multi-function Pin Selection
[15:12]	PD3MFP[3:0]	PD.3 Multi-function Pin Selection
[11:8]	PD2MFP[3:0]	PD.2 Multi-function Pin Selection
[7:4]	PD1MFP[3:0]	PD.1 Multi-function Pin Selection
[3:0]	PD0MFP[3:0]	PD.0 Multi-function Pin Selection

Multi-function GPIOD high byte Control Register (SYS_GPD_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPD_MFPH	SYS_BA+0x4C	R/W	Port D High Byte Multi-function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PD15MFP				PD14MFP			
23	22	21	20	19	18	17	16
PD13MFP				PD12MFP			
15	14	13	12	11	10	9	8
PD11MFP				PD10MFP			
7	6	5	4	3	2	1	0
PD9MFP				PD8MFP			

Bits	Description	
[31:28]	PD15MFP[3:0]	PD.15 Multi-function Pin Selection
[27:24]	PD14MFP[3:0]	PD.14 Multi-function Pin Selection
[23:20]	PD13MFP[3:0]	PD.13 Multi-function Pin Selection
[19:16]	PD12MFP[3:0]	PD.12 Multi-function Pin Selection
[15:12]	PD11MFP[3:0]	PD.11 Multi-function Pin Selection
[11:8]	PD10MFP[3:0]	PD.10 Multi-function Pin Selection
[7:4]	PD9MFP[3:0]	PD.9 Multi-function Pin Selection
[3:0]	PD8MFP[3:0]	PD.8 Multi-function Pin Selection

Multi-function GPIOE Low Byte Control Register (SYS_GPE_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPE_MFPL	SYS_BA+0x50	R/W	Port E Low Byte Multi-function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PE7MFP				PE6MFP			
23	22	21	20	19	18	17	16
PE5MFP				PE4MFP			
15	14	13	12	11	10	9	8
PE3MFP				PE2MFP			
7	6	5	4	3	2	1	0
PE1MFP				PE0MFP			

Bits	Description	
[31:28]	PE7MFP[3:0]	PE.7 Multi-function Pin Selection
[27:24]	PE6MFP[3:0]	PE.6 Multi-function Pin Selection
[23:20]	PE5MFP[3:0]	PE.5 Multi-function Pin Selection
[19:16]	PE4MFP[3:0]	PE.4 Multi-function Pin Selection
[15:12]	PE3MFP[3:0]	PE.3 Multi-function Pin Selection
[11:8]	PE2MFP[3:0]	PE.2 Multi-function Pin Selection
[7:4]	PE1MFP[3:0]	PE.1 Multi-function Pin Selection
[3:0]	PE0MFP[3:0]	PE.0 Multi-function Pin Selection

Multi-function GPIOE high byte Control Register (SYS_GPE_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPE_MFPH	SYS_BA+0x54	R/W	Port E High Byte Multi-function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PE15MFP				PE14MFP			
23	22	21	20	19	18	17	16
PE13MFP				PE12MFP			
15	14	13	12	11	10	9	8
PE11MFP				PE10MFP			
7	6	5	4	3	2	1	0
PE9MFP				PE8MFP			

Bits	Description	
[31:28]	PE15MFP[3:0]	PE.15 Multi-function Pin Selection
[27:24]	PE14MFP[3:0]	PE.14 Multi-function Pin Selection
[23:20]	PE13MFP[3:0]	PE.13 Multi-function Pin Selection
[19:16]	PE12MFP[3:0]	PE.12 Multi-function Pin Selection
[15:12]	PE11MFP[3:0]	PE.11 Multi-function Pin Selection
[11:8]	PE10MFP[3:0]	PE.10 Multi-function Pin Selection
[7:4]	PE9MFP[3:0]	PE.9 Multi-function Pin Selection
[3:0]	PE8MFP[3:0]	PE.8 Multi-function Pin Selection

Multi-function GPIOF Low Byte Control Register (SYS_GPF_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPF_MFPL	SYS_BA+0x58	R/W	Port F Low Byte Multi-function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PF7MFP				PF6MFP			
23	22	21	20	19	18	17	16
PF5MFP				PF4MFP			
15	14	13	12	11	10	9	8
PF3MFP				PF2MFP			
7	6	5	4	3	2	1	0
PF1MFP				PF0MFP			

Bits	Description	
[31:28]	PF7MFP[3:0]	PF.7 Multi-function Pin Selection
[27:24]	PF6MFP[3:0]	PF.6 Multi-function Pin Selection
[23:20]	PF5MFP[3:0]	PF.5 Multi-function Pin Selection
[19:16]	PF4MFP[3:0]	PF.4 Multi-function Pin Selection
[15:12]	PF3MFP[3:0]	PF.3 Multi-function Pin Selection
[11:8]	PF2MFP[3:0]	PF.2 Multi-function Pin Selection
[7:4]	PF1MFP[3:0]	PF.1 Multi-function Pin Selection
[3:0]	PF0MFP[3:0]	PF.0 Multi-function Pin Selection

Multi-function GPIOF high byte Control Register (SYS_GPF_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPF_MFPH	SYS_BA+0x5C	R/W	Port F High Byte Multi-function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PF15MFP				PF14MFP			
23	22	21	20	19	18	17	16
PF13MFP				PF12MFP			
15	14	13	12	11	10	9	8
PF11MFP				PF10MFP			
7	6	5	4	3	2	1	0
PF9MFP				PF8MFP			

Bits	Description	
[31:28]	PF15MFP[3:0]	PF.15 Multi-function Pin Selection
[27:24]	PF14MFP[3:0]	PF.14 Multi-function Pin Selection
[23:20]	PF13MFP[3:0]	PF.13 Multi-function Pin Selection
[19:16]	PF12MFP[3:0]	PF.12 Multi-function Pin Selection
[15:12]	PF11MFP[3:0]	PF.11 Multi-function Pin Selection
[11:8]	PF10MFP[3:0]	PF.10 Multi-function Pin Selection
[7:4]	PF9MFP[3:0]	PF.9 Multi-function Pin Selection
[3:0]	PF8MFP[3:0]	PF.8 Multi-function Pin Selection

Multi-function GPIOG Low Byte Control Register (SYS_GPG_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPG_MFPL	SYS_BA+0x60	R/W	Port G Low Byte Multi-function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PG7MFP				PG6MFP			
23	22	21	20	19	18	17	16
PG5MFP				PG4MFP			
15	14	13	12	11	10	9	8
PG3MFP				PG2MFP			
7	6	5	4	3	2	1	0
PG1MFP				PG0MFP			

Bits	Description	
[31:28]	PG7MFP[3:0]	PG.7 Multi-function Pin Selection
[27:24]	PG6MFP[3:0]	PG.6 Multi-function Pin Selection
[23:20]	PG5MFP[3:0]	PG.5 Multi-function Pin Selection
[19:16]	PG4MFP[3:0]	PG.4 Multi-function Pin Selection
[15:12]	PG3MFP[3:0]	PG.3 Multi-function Pin Selection
[11:8]	PG2MFP[3:0]	PG.2 Multi-function Pin Selection
[7:4]	PG1MFP[3:0]	PG.1 Multi-function Pin Selection
[3:0]	PG0MFP[3:0]	PG.0 Multi-function Pin Selection

Multi-function GPIOG high byte Control Register (SYS_GPG_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPG_MFPH	SYS_BA+0x64	R/W	Port G High Byte Multi-function Control Register	0xXXXX_1100

31	30	29	28	27	26	25	24
PG15MFP				PG14MFP			
23	22	21	20	19	18	17	16
PG13MFP				PG12MFP			
15	14	13	12	11	10	9	8
PG11MFP				PG10MFP			
7	6	5	4	3	2	1	0
PG9MFP				PG8MFP			

Bits	Description	
[31:28]	PG15MFP[3:0]	PG.15 Multi-function Pin Selection
[27:24]	PG14MFP[3:0]	PG.14 Multi-function Pin Selection
[23:20]	PG13MFP[3:0]	PG.13 Multi-function Pin Selection
[19:16]	PG12MFP[3:0]	PG.12 Multi-function Pin Selection
[15:12]	PG11MFP[3:0]	PG.11 Multi-function Pin Selection
[11:8]	PG10MFP[3:0]	PG.10 Multi-function Pin Selection
[7:4]	PG9MFP[3:0]	PG.9 Multi-function Pin Selection
[3:0]	PG8MFP[3:0]	PG.8 Multi-function Pin Selection

Multi-function GPIOH Low Byte Control Register (SYS_GPH_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPH_MFPL	SYS_BA+0x68	R/W	Port H Low Byte Multi-function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PH7MFP				PH6MFP			
23	22	21	20	19	18	17	16
PH5MFP				PH4MFP			
15	14	13	12	11	10	9	8
PH3MFP				PH2MFP			
7	6	5	4	3	2	1	0
PH1MFP				PH0MFP			

Bits	Description	
[31:28]	PH7MFP[3:0]	PH.7 Multi-function Pin Selection
[27:24]	PH6MFP[3:0]	PH.6 Multi-function Pin Selection
[23:20]	PH5MFP[3:0]	PH.5 Multi-function Pin Selection
[19:16]	PH4MFP[3:0]	PH.4 Multi-function Pin Selection
[15:12]	PH3MFP[3:0]	PH.3 Multi-function Pin Selection
[11:8]	PH2MFP[3:0]	PH.2 Multi-function Pin Selection
[7:4]	PH1MFP[3:0]	PH.1 Multi-function Pin Selection
[3:0]	PH0MFP[3:0]	PH.0 Multi-function Pin Selection

Multi-function GPIOH high byte Control Register (SYS_GPH_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPH_MFPH	SYS_BA+0x6C	R/W	Port H High Byte Multi-function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PH15MFP				PH14MFP			
23	22	21	20	19	18	17	16
PH13MFP				PH12MFP			
15	14	13	12	11	10	9	8
PH11MFP				PH10MFP			
7	6	5	4	3	2	1	0
PH9MFP				PH8MFP			

Bits	Description	
[31:28]	PH15MFP[3:0]	PH.15 Multi-function Pin Selection
[27:24]	PH14MFP[3:0]	PH.14 Multi-function Pin Selection
[23:20]	PH13MFP[3:0]	PH.13 Multi-function Pin Selection
[19:16]	PH12MFP[3:0]	PH.12 Multi-function Pin Selection
[15:12]	PH11MFP[3:0]	PH.11 Multi-function Pin Selection
[11:8]	PH10MFP[3:0]	PH.10 Multi-function Pin Selection
[7:4]	PH9MFP[3:0]	PH.9 Multi-function Pin Selection
[3:0]	PH8MFP[3:0]	PH.8 Multi-function Pin Selection

Multi-function GPIOI Low Byte Control Register (SYS_GPI_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPI_MFPL	SYS_BA+0x70	R/W	Port I Low Byte Multi-function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PI7MFP				PI6MFP			
23	22	21	20	19	18	17	16
PI5MFP				PI4MFP			
15	14	13	12	11	10	9	8
PI3MFP				PI2MFP			
7	6	5	4	3	2	1	0
PI1MFP				PI0MFP			

Bits	Description	
[31:28]	PI7MFP[3:0]	PI.7 Multi-function Pin Selection
[27:24]	PI6MFP[3:0]	PI.6 Multi-function Pin Selection
[23:20]	PI5MFP[3:0]	PI.5 Multi-function Pin Selection
[19:16]	PI4MFP[3:0]	PI.4 Multi-function Pin Selection
[15:12]	PI3MFP[3:0]	PI.3 Multi-function Pin Selection
[11:8]	PI2MFP[3:0]	PI.2 Multi-function Pin Selection
[7:4]	PI1MFP[3:0]	PI.1 Multi-function Pin Selection
[3:0]	PI0MFP[3:0]	PI.0 Multi-function Pin Selection

Multi-function GPIOI high byte Control Register (SYS_GPI_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPI_MFPH	SYS_BA+0x74	R/W	Port I High Byte Multi-function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PI15MFP				PI14MFP			
23	22	21	20	19	18	17	16
PI13MFP				PI12MFP			
15	14	13	12	11	10	9	8
PI11MFP				PI10MFP			
7	6	5	4	3	2	1	0
PI9MFP				PI8MFP			

Bits	Description	
[31:28]	PI15MFP[3:0]	PI.15 Multi-function Pin Selection
[27:24]	PI14MFP[3:0]	PI.14 Multi-function Pin Selection
[23:20]	PI13MFP[3:0]	PI.13 Multi-function Pin Selection
[19:16]	PI12MFP[3:0]	PI.12 Multi-function Pin Selection
[15:12]	PI11MFP[3:0]	PI.11 Multi-function Pin Selection
[11:8]	PI10MFP[3:0]	PI.10 Multi-function Pin Selection
[7:4]	PI9MFP[3:0]	PI.9 Multi-function Pin Selection
[3:0]	PI8MFP[3:0]	PI.8 Multi-function Pin Selection

SRAM Failed Interrupt Enable Control Register (SYS_SRAM_INTCTL)

Register	Offset	R/W	Description	Reset Value
SYS_SRAM_INTCTL	SYS_BA+0xC0	R/W	SRAM Failed Interrupt Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							PERRIEN

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**PERRIEN**	**SRAM Parity Check Fail Interrupt Enable Bit** 0 = SRAMF INT Disabled. 1 = SRAMF INT Enabled when SRAM fail flag.

SRAM Parity Check Error Flag (SYS_SRAM_STATUS)

Register	Offset	R/W	Description	Reset Value
SYS_SRAM_STATUS	SYS_BA+0xC4	R/W	SRAM Parity Check Error Flag	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						PERRIF1	PERRIF0

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**PERRIF1**	**SRAM Parity Check Fail Flag** 0 = 2nd SRAM fail. 1 = 2nd SRAM Fail.
[0]	**PERRIF0**	**SRAM Parity Check Fail Flag** 0 = No first 1 SRAM fail. 1 = First SRAM Fail.

SRAM Parity Check Error First Address1 (SYS_SRAM0_ERRADDR)

Register	Offset	R/W	Description	Reset Value
SYS_SRAM0_ERRADDR	SYS_BA+0xC8	R	SRAM Parity Check Error First Address1	0x0000_0000

31	30	29	28	27	26	25	24
PERRADDR [31:24]							
23	22	21	20	19	18	17	16
PERRADDR [23:16]							
15	14	13	12	11	10	9	8
PERRADDR [15:8]							
7	6	5	4	3	2	1	0
PERRADDR [7:0]							

Bits	Description	
[31:0]	PERRADDR	**First SRAM Parity Check Fail Address** This register shows the first system SRAM parity error byte address.

SRAM Parity Check Error First Address2 (SYS_SRAM1_ERRADDR)

Register	Offset	R/W	Description	Reset Value
SYS_SRAM1_ERRADDR	SYS_BA+0xCC	R	SRAM Parity Check Error First Address2	0x0000_0000

31	30	29	28	27	26	25	24
PERRADDR [31:24]							
23	22	21	20	19	18	17	16
PERRADDR [23:16]							
15	14	13	12	11	10	9	8
PERRADDR [15:8]							
7	6	5	4	3	2	1	0
PERRADDR [7:0]							

Bits	Description	
[31:0]	PERRADDR	**Second SRAM Parity Check Fail Address** This register shows the second system SRAM parity error byte address.

HIRC Trim Control Register (SYS_IRCTCTL)

Register	Offset	R/W	Description	Reset Value
SYS_IRCTCTL	SYS_BA+0xF0	R/W	IRC Trim Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							CESTOPEN
7	6	5	4	3	2	1	0
RETRYCNT		LOOPSEL		Reserved		FREQSEL	

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**CESTOPEN**	**Clock Error Stop Enable Bit** 0 = The trim operation is keep going if clock is inaccuracy. 1 = The trim operation is stopped if clock is inaccuracy.
[7:6]	**RETRYCNT**	**Trim Value Update Limitation Count** This field defines that how many times the auto trim circuit will try to update the HIRC trim value before the frequency of HIRC locked. Once the HIRC locked, the internal trim value update counter will be reset. If the trim value update counter reached this limitation value and frequency of HIRC still doesn't lock, the auto trim operation will be disabled and FREQSEL will be cleared to 00. 00 = Trim retry count limitation is 64. 01 = Trim retry count limitation is 128. 10 = Trim retry count limitation is 256. 11 = Trim retry count limitation is 512.
[5:4]	**LOOPSEL**	**Trim Calculation Loop** This field defines that trim value calculation is based on how many 32.768 kHz clock. For example, if CALCLOOP is set as 00, auto trim circuit will calculate trim value based on the average frequency difference in 4 32.768 kHz clock. 00 = Trim value calculation is based on average difference in 4 32.768 kHz clock. 01 = Trim value calculation is based on average difference in 8 32.768 kHz clock. 10 = Trim value calculation is based on average difference in 16 32.768 kHz clock. 11 = Trim value calculation is based on average difference in 32 32.768 kHz clock.
[3:2]	**Reserved**	Reserved.
[1:0]	**FREQSEL**	**Trim Frequency Selection** This field indicates the target frequency of HIRC auto trim. If no any target frequency is selected (FREQSEL is 00), the HIRC auto trim function is disabled.

		During auto trim operation, if clock error detected with CESTOPEN is set to 1 or trim retry limitation count reached, this field will be cleared to 00 automatically. 00 = Disable HIRC auto trim function. 01 = Enable HIRC auto trim function and trim HIRC to 22.1184 MHz. 10 = Enable HIRC auto trim function and trim HIRC to 24 MHz. 11 = Reserved.

HIRC Trim Interrupt Enable Control Register (SYS_IRCTIEN)

Register	Offset	R/W	Description	Reset Value
SYS_IRCTIEN	SYS_BA+0xF4	R/W	IRC Trim Interrupt Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					CLKEIEN	TFAILIEN	Reserved

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**CLKEIEN**	**Clock Error Interrupt Enable Bit** This bit controls if CPU would get an interrupt while clock is inaccuracy during auto trim operation. If this bit is set to1, and CLKERRIF is set during auto trim operation, an interrupt will be triggered to notify the clock frequency is inaccuracy. 0 = Disable CLKERRIF status to trigger an interrupt to CPU. 1 = Enable CLKERRIF status to trigger an interrupt to CPU.
[1]	**TFAILIEN**	**Trim Failure Interrupt Enable** This bit controls if an interrupt will be triggered while HIRC trim value update limitation count reached and HIRC frequency still not locked on target frequency set by FREQSEL. If this bit is high and TFAILIF is set during auto trim operation, an interrupt will be triggered to notify that HIRC trim value update limitation count was reached. 0 = Disable TFAILIF status to trigger an interrupt to CPU. 1 = Enable TFAILIF status to trigger an interrupt to CPU.
[0]	**Reserved**	Reserved.

HIRC Trim Interrupt Status Register (SYS_IRCTISTS)

Register	Offset	R/W	Description	Reset Value
SYS_IRCTISTS	SYS_BA+0xF8	R/W	IRC Trim Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					CLKERRIF	TFAILIF	FREQLOCK

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**CLKERRIF**	**Clock Error Interrupt Status** When the frequency of external 32.768 kHz low-speed crystal or HIRC is shift larger to unreasonable value, this bit will be set and to be an indicate that clock frequency is inaccuracy Once this bit is set to 1, the auto trim operation stopped and FREQSEL will be cleared to 00 by hardware automatically if CESTOPEN is set to 1. If this bit is set and CLKEIEN is high, an interrupt will be triggered to notify the clock frequency is inaccuracy. Write 1 to clear this to 0. 0 = Clock frequency is accuracy. 1 = Clock frequency is inaccuracy.
[1]	**TFAILIF**	**Trim Failure Interrupt Status** This bit indicates that HIRC trim value update limitation count reached and the HIRC clock frequency still doesn't be locked. Once this bit is set, the auto trim operation stopped and FREQSEL will be cleared to 00 by hardware automatically. If this bit is set and TFAILIEN is high, an interrupt will be triggered to notify that HIRC trim value update limitation count was reached. Write 1 to clear this to 0. 0 = Trim value update limitation count does not reach. 1 = Trim value update limitation count reached and HIRC frequency still not locked.
[0]	**FREQLOCK**	**HIRC Frequency Lock Status** This bit indicates the HIRC frequency is locked. This is a status bit and doesn't trigger any interrupt.

Register Write-Protection Control Register (SYS_REGLCTL)

Some of the system control registers need to be protected to avoid inadvertent write and disturb the chip operation. These system control registers are protected after the power on reset till user to disable register protection. For user to program these protected registers, a register protection disable sequence needs to be followed by a special programming. The register protection disable sequence is writing the data “59h”, “16h” “88h” to the register SYS_REGLCTL address at 0x4000_0100 continuously. Any different data value, different sequence or any other write to other address during these three data writing will abort the whole sequence.

After the protection is disabled, user can check the protection disable bit at address 0x4000_0100 bit0, 1 is protection disable, and 0 is protection enable. Then user can update the target protected register value and then write any data to the address “0x4000_0100” to enable register protection.

This register is written to disable/enable register protection and read for the REGLCTL status.

Register	Offset	R/W	Description	Reset Value
SYS_REGLCTL	SYS_BA+0x100	R/W	Register Write-Protection Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
REGLCTL[7:1]							**REGLCTL[0]**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**REGLCTL**	**Register Write-Protection Code (Write Only)** Some registers have write-protection function. Writing these registers have to disable the protected function by writing the sequence value “59h”, “16h”, “88h” to this field. After this sequence is completed, the REGLCTL bit will be set to 1 and write-protection registers can be normal write.
[0]	**REGLCTL**	**Register Write-Protection Disable Index (Read Only)** 0 = Write-protection Enabled for writing protected registers. Any write to the protected register is ignored. 1 = Write-protection Disabled for writing protected registers. The Protected registers are: **SYS_IPRST0**: address 0x4000_0008 **SYS_BODCTL**: address 0x4000_0018 **SYS_PORCTL**: address 0x4000_0024 **PWRCON**: address 0x4000_0200 (bit[6] is not protected for power wake-up interrupt clear) **APBCLK bit[0]**: address 0x4000_0208 (bit[0] is watchdog clock enable) **CLKSEL0**: address 0x4000_0210 (for HCLK and CPU STCLK clock source select)

		CLKSEL1 bit[1:0]: address 0x4000_0214 (for watchdog clock source select) **NMI_SEL]**: address 0x4000_0300 (for NMI source select) **ISPCON**: address 0x4000_5000 (Flash ISP Control register) **ISPTRG**: address 0x4000_5010 (ISP Trigger Control register) **WTCR**: address 0x4004_0000 **FATCON**: address 0x4000_5018 **TAMPER**: address 0x400E_1000

6.2.6 System Timer (SysTick)

The Cortex®-M4 includes an integrated system timer, SysTick, which provides a simple, 24-bit clear-on-write, decrementing, wrap-on-zero counter with a flexible control mechanism. The counter can be used as a Real Time Operating System (RTOS) tick timer or as a simple counter.

When system timer is enabled, it will count down from the value in the SysTick Current Value Register (SYST_CVR) to zero, and reload (wrap) to the value in the SysTick Reload Value Register (SYST_RVR) on the next clock cycle, and then decrement on subsequent clocks. When the counter transitions to zero, the COUNTFLAG status bit is set. The COUNTFLAG bit clears on reads.

The SYST_CVR value is UNKNOWN on reset. Software should write to the register to clear it to zero before enabling the feature. This ensures the timer will count from the SYST_RVR value rather than an arbitrary value when it is enabled.

If the SYST_RVR is zero, the timer will be maintained with a current value of zero after it is reloaded with this value. This mechanism can be used to disable the feature independently from the timer enable bit.

For more detailed information, please refer to the "*ARM® Cortex®-M4 Technical Reference Manual*" and "*ARM® v6-M Architecture Reference Manual*".

6.2.6.1 System Timer Control Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
SCS Base Address: **SCS_BA = 0xE000_E000**				
SYST_CSR	SCS_BA+0x10	R/W	SysTick Control and Status Register	0x0000_0000
SYST_RVR	SCS_BA+0x14	R/W	SysTick Reload Value Register	0xXXXX_XXXX
SYST_CVR	SCS_BA+0x18	R/W	SysTick Current Value Register	0xXXXX_XXXX

6.2.6.2 System Timer Control Register Description

SysTick Control and Status (SYST_CSR)

Register	Offset	R/W	Description	Reset Value
SYST_CSR	SCS_BA+0x10	R/W	SysTick Control and Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							COUNTFLAG
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					CLKSRC	TICKINT	ENABLE

Bits	Description	
[31:17]	Reserved	Reserved.
[16]	COUNTFLAG	System Tick Counter Flag Returns 1 if timer counted to 0 since last time this register was read. COUNTFLAG is set by a count transition from 1 to 0. COUNTFLAG is cleared on read or by a write to the Current Value register.
[15:3]	Reserved	Reserved.
[2]	CLKSRC	System Tick Clock Source Selection 0 = Clock source is the (optional) external reference clock. 1 = Core clock used for SysTick.
[1]	TICKINT	System Tick Interrupt Enabled 0 = Counting down to 0 does not cause the SysTick exception to be pended. Software can use COUNTFLAG to determine if a count to zero has occurred. 1 = Counting down to 0 will cause the SysTick exception to be pended. Clearing the SysTick current value register by a register write in software will not cause SysTick to be pended.
[0]	ENABLE	System Tick Counter Enabled 0 = Counter Disabled. 1 = Counter will operate in a multi-shot manner.

SysTick Reload Value Register (SYST_RVR)

Register	Offset	R/W	Description	Reset Value
SYST_RVR	SCS_BA+0x14	R/W	SysTick Reload Value Register	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
RELOAD[23:16]							
15	14	13	12	11	10	9	8
RELOAD[15:8]							
7	6	5	4	3	2	1	0
RELOAD[7:0]							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**RELOAD**	**System Tick Reload Value** Value to load into the Current Value register when the counter reaches 0.

SysTick Current Value Register (SYST_CVR)

Register	Offset	R/W	Description	Reset Value
SYST_CVR	SCS_BA+0x18	R/W	SysTick Current Value Register	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
CURRENT [23:16]							
15	14	13	12	11	10	9	8
CURRENT [15:8]							
7	6	5	4	3	2	1	0
CURRENT[7:0]							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**CURRENT**	**System Tick Current Value** Current counter value. This is the value of the counter at the time it is sampled. The counter does not provide read-modify-write protection. The register is write-clear. A software write of any value will clear the register to 0. Unsupported bits RAZ (see SysTick Reload Value register).

6.2.7 Nested Vectored Interrupt Controller (NVIC)

The NVIC and the processor core interface are closely coupled to enable low latency interrupt processing and efficient processing of late arriving interrupts. The NVIC maintains knowledge of the stacked, or nested, interrupts to enable tail-chaining of interrupts. You can only fully access the NVIC from privileged mode, but you can cause interrupts to enter a pending state in user mode if you enable the Configuration and Control Register. Any other user mode access causes a bus fault. You can access all NVIC registers using byte, halfword, and word accesses unless otherwise stated. NVIC registers are located within the SCS (System Control Space). All NVIC registers and system debug registers are little-endian regardless of the endianness state of the processor.

The NVIC supports:

- An implementation-defined number of interrupts, in the range 1-240 interrupts.
- A programmable priority level of 0-16 for each interrupt; a higher level corresponds to a lower priority, so level 0 is the highest interrupt priority.
- Level and pulse detection of interrupt signals.
- Dynamic reprioritization of interrupts.
- Grouping of priority values into group priority and subpriority fields.
- Interrupt tail-chaining.
- An external Non Maskable Interrupt (NMI)
- WIC with Ultra-low Power Sleep mode support

The processor automatically stacks its state on exception entry and unstacks this state on exception exit, with no instruction overhead. This provides low latency exception handling.

6.2.7.1 Exception Model and System Interrupt Map

The following table lists the exception model supported by NUC4xx series. Software can set 16 levels of priority on some of these exceptions as well as on all interrupts. The highest user-configurable priority is denoted as “0x00” and the lowest priority is denoted as “0xF0” (The 4-LSB always 0). The default priority of all the user-configurable interrupts is “0x00”. Note that priority “0” is treated as the fourth priority on the system, after three system exceptions “Reset”, “NMI” and “Hard Fault”.

When any interrupts is accepted, the processor will automatically fetch the starting address of the interrupt service routine (ISR) from a vector table in memory. On system reset, the vector table is fixed at address 0x00000000. Privileged software can write to the VTOR to relocate the vector table start address to a different memory location, in the range 0x00000080 to 0x3FFFFF80,

The vector table contains the initialization value for the stack pointer on reset, and the entry point addresses for all exception handlers. The vector number on previous page defines the order of entries in the vector table associated with exception handler entry as illustrated in previous section.

Exception Type	Vector Number	Vector Address	Priority
Reset	1	0x00000004	-3
NMI	2	0x00000008	-2
Hard Fault	3	0x0000000C	-1
Memory Manager Fault	4	0x00000010	Configurable
Bus Fault	5	0x00000014	Configurable
Usage Fault	6	0x00000018	Configurable
Reserved	7 ~ 10		Reserved
SVCall	11	0x0000002C	Configurable
Debug Monitor	12	0x00000030	Configurable
Reserved	13		Reserved
PendSV	14	0x00000038	Configurable
SysTick	15	0x0000003C	Configurable
Interrupt (IRQ0 ~ IRQ)	16 ~ 144	0x00000000 + (Vector Number)*4	Configurable

Table 6.2-2 Exception Model

Vector Number	Interrupt Number (Bit In Interrupt Registers)	Interrupt Name	Interrupt Description
0 ~ 15	-	-	System exceptions
16	0	BOD_OUT	Brown-Out low voltage detected interrupt
17	1	IRC_INT	IRC TRIM interrupt
18	2	PWRWU_INT	Clock controller interrupt for chip wake-up from power down state
19	3	SRAMF	SRAM parity check fail
20	4	CLKF	Clock detection fail

21	5	-	Reserved
22	6	RTC_INT	Real time clock interrupt
23	7	TAMPER	TAMPER interrupt
24	8	EINT0	External signal interrupt from PA.0 pin
25	9	EINT1	External signal interrupt from PB.0 pin
26	10	EINT2	External signal interrupt from PC.0 pin
27	11	EINT3	External signal interrupt from PD.0 pin
28	12	EINT4	External signal interrupt from PE.0 pin
29	13	EINT5	External signal interrupt from PF.0 pin
30	14	EINT6	External signal interrupt from PG.0 pin
31	15	EINT7	External signal interrupt from PH.0 pin
32	16	GPA_INT	External signal interrupt from PI.0 pin
33	17	GPB_INT	External signal interrupt from PA[15:1]
34	18	GPC_INT	External signal interrupt from PB[15:1]
35	19	GPD_INT	External signal interrupt from PC[15:1]
36	20	GPE_INT	External signal interrupt from PD[15:1]
37	21	GPF_INT	External signal interrupt from PE[15:1]
38	22	GPG_INT	External signal interrupt from PF[15:1]
39	23	GPH_INT	External signal interrupt from PG[15:1]
40	24	GPI_INT	External signal interrupt from PH[15:1]
41	25	GPJ_INT	External signal interrupt from PI[15:1]
42 ~ 47	26 ~ 31	-	Reserved
48	32	TMR0_INT	Timer 0 interrupt
49	33	TMR1_INT	Timer 1 interrupt
50	34	TMR2_INT	Timer 2 interrupt
51	35	TMR3_INT	Timer 3 interrupt
52 ~ 55	36 ~ 39	-	Reserved
56	40	PDMA_INT	PDMA interrupt
57	41	-	Reserved
58	42	ADC_INT	ADC interrupt
59 ~ 61	43 ~ 45	-	Reserved
62	46	WDT_INT	Watchdog Timer interrupt
63	47	WWDG_INT	Window Watchdog Timer interrupt
64	48	EADC0	Enhanced ADC 0 interrupt
65	49	EADC1	Enhanced ADC 1 interrupt

66	50	EADC2	Enhanced ADC 2 interrupt
67	51	EADC3	Enhanced ADC 3 interrupt
68 ~ 71	52 ~ 55	-	Reserved
72	56	ACMP_INT	Analog Comparator-0 or Comaprator-1 interrupt
73 ~ 75	57 ~ 59	-	Reserved
76	60	OPA0_INT	Analog OP0 interrupt
77	61	OPA1_INT	Analog OP1 interrupt
78	62	ICAP0	Internal Capture 0 interrupt
79	63	ICAP1	Internal Capture 1 interrupt
80	64	PWM0_0_INT	Internal Capture 1 interruptPWM0_0 interrupt
81	65	PWM0_1_INT	PWM0_1 interrupt
82	66	PWM0_2_INT	PWM0_2 interrupt
83	67	PWM0_3_INT	PWM0_3 interrupt
84	68	PWM0_4_INT	PWM0_4 interrupt
85	69	PWM0_5_INT	PWM0_5 interrupt
86	70	PWMABRK	PWMA BRK interrupt
87	71	QEI0	QEI0 interrupt
88	72	PWM1_0_INT	PWM1_0 interrupt
89	73	PWM1_1_INT	PWM1_1 interrupt
90	74	PWM1_2_INT	PWM1_2 interrupt
91	75	PWM1_3_INT	PWM1_3 interrupt
92	76	PWM1_4_INT	PWM1_4 interrupt
93	77	PWM1_5_INT	PWM1_5 interrupt
94	78	PWMBBRK	PWMB BRK interrupt
95	79	QEI1	QEI1 interrupt
96	80	EPWMA_INT	EPWMA interrupt
97	81	EPWMABRK	EPWMA brake interrupt
98	82	EPWMB_INT	EPWMB interrupt
99	83	EPWMBBRK	EPWMB brake interrupt
100 ~ 103	84 ~ 87	-	Reserved
104	88	UDC_INT	USB device interrupt
105	89	UHC_INT	USB host interrupt
106	90	OTG_INT	USB OTG interrupt
107	91	-	Reserved
108	92	EMAC_TX	Ethernet MAC transmit interrupt

109	93	EMAC_RX	Ethernet MAC receive interrupt
110 ~ 111	94 ~ 95	-	Reserved
112	96	SPI0_INT	SPI0 interrupt
113	97	SPI1_INT	SPI1 interrupt
114	98	SPI2_INT	SPI2 interrupt
115	99	SPI3_INT	SPI3 interrupt
116 ~ 119	100 ~ 103	-	Reserved
120	104	UART0_INT	UART0 interrupt
121	105	UART1_INT	UART1 interrupt
122	106	UART2_INT	UART2 interrupt
123	107	UART3_INT	UART3 interrupt
124	108	UART4_INT	UART4 interrupt
125	109	UART5_INT	UART5 interrupt
126 ~ 127	110 ~ 111	-	Reserved
128	112	I2C0_INT	I2C0 interrupt
129	113	I2C1_INT	I2C1 interrupt
130	114	I2C2_INT	I2C2 interrupt
131	115	I2C3_INT	I2C3 interrupt
132	116	I2C4_INT	I2C4 interrupt
133 ~ 135	117 ~ 119	-	Reserved
136	120	SC0	Smartcard 0 interrupt
137	121	SC1	Smartcard 1 interrupt
138	122	SC2	Smartcard 2 interrupt
139	123	SC3	Smartcard 3 interrupt
140	124	SC4	Smartcard 4 interrupt
141	125	SC5	Smartcard 5 interrupt
142 ~ 143	126 ~ 127	-	Reserved
144	128	CAN0_INT	CAN0 interrupt
145	129	CAN1_INT	CAN1 interrupt
146 ~ 147	130 ~ 131	-	Reserved
148	132	I2S_INT	I2S interrupt
149	133	I2S1_INT	I2S1 interrupt
150 ~ 151	134 ~ 135	-	Reserved
152	136	SDHOST	SD host interrupt
153	137	-	Reserved

154	138	PS2_INT	PS/2 interrupt
155	139	CAP	Image capture interface interrupt
156	140	CRYPTO	Crypto interrupt

Table 6.2-3 Interrupt Number Table

6.2.7.2 Operation Description

NVIC interrupts can be enabled and disabled by writing to their corresponding Interrupt Set-Enable or Interrupt Clear-Enable register bit-field. The registers use a write-1-to-enable and write-1-to-clear policy, both registers reading back the current enabled state of the corresponding interrupts. When an interrupt is disabled, interrupt assertion will cause the interrupt to become Pending, however, the interrupt will not activate. If an interrupt is Active when it is disabled, it remains in its Active state until cleared by reset or an exception return. Clearing the enable bit prevents new activations of the associated interrupt.

NVIC interrupts can be pended/un-pended using a complementary pair of registers to those used to enable/disable the interrupts, named the Set-Pending Register and Clear-Pending Register respectively. The registers use a write-1-to-enable and write-1-to-clear policy, both registers reading back the current pended state of the corresponding interrupts. The Clear-Pending Register has no effect on the execution status of an Active interrupt.

NVIC interrupts are prioritized by updating an 8-bit field within a 32-bit register (each register supporting four interrupts).

The general registers associated with the NVIC are all accessible from a block of memory in the System Control Space and will be described in next section.

6.2.7.3 NVIC Control Registers

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
NVIC Base Address **NVIC_BA = 0xE000_E100**				
NVIC_ISERn **n=0,1..4**	NVIC_BA+0x4*n	R/W	IRQ0 ~ IRQ159 Set-Enable Control Register	0x0000_0000
NVIC_ICERn **n=0,1..4**	NVIC_BA+0x80 +0x4*n	R/W	IRQ0 ~ IRQ159 Clear-Enable Control Register	0x0000_0000
NVIC_ISPRn **n=0,1..4**	NVIC_BA+0x100 +0x4*n	R/W	IRQ0 ~ IRQ159 Set-Pending Control Register	0x0000_0000
NVIC_ICPRn **n=0,1..4**	NVIC_BA+0x180 +0x4*n	R/W	IRQ0 ~ IRQ159 Clear-Pending Control Register	0x0000_0000
NVIC_IABRn **n=0,1..4**	NVIC_BA+0x200 +0x4*n	R/W	IRQ0 ~ IRQ159 Active Bit Register	0x0000_0000
NVIC_IPRn **n=0,1..31**	NVIC_BA+0x300 +0x4*n	R/W	IRQ0 ~ IRQ159 Interrupt Priority Control Register	0x0000_0000
NVIC_STIR	NVIC_BA+0xE00	R/W	Software Trigger Interrupt Registers	0x0000_0000

IRQ0 ~ IRQ159 Set-Enable Control Register (NVIC_ISERn)

Register	Offset	R/W	Description	Reset Value
NVIC_ISERn **n=0,1..4**	NVIC_BA+0x4*n	R/W	IRQ0 ~ IRQ159 Set-Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
SETENA[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SETENA[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SETENA[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SETENA[7:0]							

Bits	Description	
[31:0]	**SETENA**	**Interrupt Set Enable Bit** The NVIC_ISER0-NVIC_ISER3 registers enable interrupts, and show which interrupts are enabled Write: 0 = No effect. 1 = Interrupt Enabled. Read: 0 = Interrupt Disabled. 1 = Interrupt Enabled.

IRQ0 ~ IRQ159 Clear-Enable Control Register (NVIC_ICERn)

Register	Offset	R/W	Description	Reset Value
NVIC_ICERn **n=0,1..4**	NVIC_BA+0x80 +0x4*n	R/W	IRQ0 ~ IRQ159 Clear-Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
CLRENA[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CLRENA[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CLRENA[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CLRENA[7:0]							

Bits	Description	
[31:0]	**CLRENA**	**Interrupt Clear Enable Control** The NVIC_ICER0-NVIC_ICER3 registers disable interrupts, and show which interrupts are enabled. Write: 0 = No effect. 1 = Interrupt Disabled. Read: 0 = Interrupt Disabled. 1 = Interrupt Enabled.

IRQ0 ~ IRQ159 Set-Pending Control Register (NVIC_ISPRn)

Register	Offset	R/W	Description	Reset Value
NVIC_ISPRn **n=0,1..4**	NVIC_BA+0x100 +0x4*n	R/W	IRQ0 ~ IRQ159 Set-Pending Control Register	0x0000_0000

31	30	29	28	27	26	25	24
SETPEND[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SETPEND[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SETPEND[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SETPEND[7:0]							

Bits	Description	
[31:0]	**SETPEND**	**Interrupt Set-Pending** The NVIC_ISPR0-NVIC_ISPR3 registers force interrupts into the pending state, and show which interrupts are pending Write: 0 = No effect. 1 = Changes interrupt state to pending. Read: 0 = Interrupt is not pending. 1 = Interrupt is pending.

IRQ0 ~ IRQ159 Clear-Pending Control Register (NVIC_ICPRn)

Register	Offset	R/W	Description	Reset Value
NVIC_ICPRn **n=0,1..4**	NVIC_BA+0x180 +0x4*n	R/W	IRQ0 ~ IRQ159 Clear-Pending Control Register	0x0000_0000

31	30	29	28	27	26	25	24
CLRPEND[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CLRPEND[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CLRPEND[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CLRPEND[7:0]							

Bits	Description	
[31:0]	**CLRPEND**	**Interrupt Clear-Pending** The NVIC_ICPR0-NCVIC_ICPR3 registers remove the pending state from interrupts, and show which interrupts are pending Write: 0 = No effect. 1 = Removes pending state an interrupt. Read: 0 = Interrupt is not pending. 1 = Interrupt is pending.

IRQ0 ~ IRQ159 Active Bit Register (NVIC_IABRn)

Register	Offset	R/W	Description	Reset Value
NVIC_IABRn **n=0,1..4**	NVIC_BA+0x200 +0x4*n	R/W	IRQ0 ~ IRQ159 Active Bit Register	0x0000_0000

31	30	29	28	27	26	25	24
ACTIVE[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ACTIVE[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ACTIVE[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ACTIVE[7:0]							

Bits	Description	
[31:0]	**ACTIVE**	**Interrupt Active Flags** The NVIC_IABR0-NVIC_IABR3 registers indicate which interrupts are active. 0 = interrupt not active. 1 = interrupt active.

IRQ0 ~ IRQ159 Interrupt Priority Register (NVIC_IPRn)

Register	Offset	R/W	Description	Reset Value
NVIC_IPRn **n=0,1..31**	NVIC_BA+0x300 +0x4*n	R/W	IRQ0 ~ IRQ159 Interrupt Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_4n[3]				**Reserved**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PRI_4n[2]				**Reserved**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PRI_4n[1]				**Reserved**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRI_4n[0]				**Reserved**			

Bits	Description	
[31:28]	**PRI_4n[3]**	**Priority Of IRQ_4n+3** "0" denotes the highest priority and "15" denotes the lowest priority
[27:24]	**Reserved**	Reserved.
[23:20]	**PRI_4n[2]**	**Priority Of IRQ_4n+2** "0" denotes the highest priority and "15" denotes the lowest priority
[19:16]	**Reserved**	Reserved.
[15:12]	**PRI_4n[1]**	**Priority Of IRQ_4n+1** "0" denotes the highest priority and "15" denotes the lowest priority
[11:8]	**Reserved**	Reserved.
[7:4]	**PRI_4n[0]**	**Priority Of IRQ_4n+0** "0" denotes the highest priority and "15" denotes the lowest priority
[3:0]	**Reserved**	Reserved.

Software Trigger Interrupt Register (NVIC_STIR)

Register	Offset	R/W	Description	Reset Value
NVIC_STIR	NVIC_BA+0xE00	R/W	Software Trigger Interrupt Registers	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							INTID [8]
7	6	5	4	3	2	1	0
INTID [7:0]							

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8:0]	**INTID**	**Interrupt ID** Write to the STIR To Generate An Interrupt from Software When the USERSETMPEND bit in the SCR is set to 1, unprivileged software can access the STIR Interrupt ID of the interrupt to trigger, in the range 0-63. For example, a value of 0x03 specifies interrupt IRQ3.

6.2.7.4 NMI Control Registers

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
NMI Base Address: **NMI_BA = 0x4000_0300**				
NMIEN	NMI_BA+0x00	R/W	NMI source interrupt Enable Control Register	0x0000_0000
NMISTS	NMI_BA+0x04	R	NMI source interrupt Status Register	0x0000_0000

NMI source Interrupt Enable Control Register (NMIEN)

Register	Offset	R/W	Description	Reset Value
NMIEN	NMI_BA+0x00	R/W	NMI source interrupt Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
EINT7	**EINT6**	**EINT5**	**EINT4**	**EINT3**	**EINT2**	**EINT1**	**EINT0**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TAMPER	**RTC**	**Reserved**	**CLKFAIL**	**SRAMFAIL**	**PWRWK**	**IRC**	**BOD**

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15]	**EINT7**	**External Interrupt 7 NMI Source Enable** 0 = External interrupt 7 NMI source Disabled. 1 = External interrupt 7 NMI source Enabled.
[14]	**EINT6**	**External Interrupt 6 NMI Source Enable** 0 = External interrupt 6 NMI source Disabled. 1 = External interrupt 6 NMI source Enabled.
[13]	**EINT5**	**External Interrupt 5 NMI Source Enable** 0 = External interrupt 5 NMI source Disabled. 1 = External interrupt 5 NMI source Enabled.
[12]	**EINT4**	**External Interrupt 4 NMI Source Enable** 0 = External interrupt 4 NMI source Disabled. 1 = External interrupt 4 NMI source Enabled.
[11]	**EINT3**	**External Interrupt 3 NMI Source Enable** 0 = External interrupt 3 NMI source Disabled. 1 = External interrupt 3 NMI source Enabled.
[10]	**EINT2**	**External Interrupt 2 NMI Source Enable** 0 = External interrupt 2 NMI source Disabled. 1 = External interrupt 2 NMI source Enabled.
[9]	**EINT1**	**External Interrupt 1 NMI Source Enable** 0 = External interrupt 1 NMI source Disabled. 1 = External interrupt 1 NMI source Enabled.

[8]	**EINT0**	**External Interrupt 0 NMI Source Enable** 0 = External interrupt 0 NMI source Disabled. 1 = External interrupt 0 NMI source Enabled.
[7]	**TAMPER**	**TAMPER_INT NMI Source Enable** 0 = Backup register tamper detected interrupt.NMI source Disabled. 1 = Backup register tamper detected interrupt.NMI source Enabled.
[6]	**RTC**	**RTC NMI Source Enable** 0 = RTC NMI source Disabled. 1 = RTC NMI source Enabled.
[5]	**Reserved**	Reserved.
[4]	**CLKFAIL**	**Clock Fail Detected NMI Source Enable** 0 = Clock fail detected interrupt NMI source Disabled. 1 = Clock fail detected interrupt NMI source Enabled.
[3]	**SRAMFAIL**	**SRAM ParityCheck Error NMI Source Enable** 0 = SRAM parity check error NMI source Disabled. 1 = SRAM parity check error NMI source Enabled.
[2]	**PWRWK**	**Power-down Mode Wake-up NMI Source Enable** 0 = Power-down mode wake-up NMI source Disabled. 1 = Power-down mode wake-up NMI source Enabled.
[1]	**IRC**	**IRC TRIM NMI Source Enable** 0 = IRC TRIM NMI source Disabled. 1 = IRC TRIM NMI source Enabled.
[0]	**BOD**	**BOD NMI Source Enable** 0 = BOD NMI source Disabled. 1 = BOD NMI source Enabled.

NMI source Interrupt Status Register (NMISTS)

Register	Offset	R/W	Description	Reset Value
NMISTS	NMI_BA+0x04	R	NMI source interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
EINT7	EINT6	EINT5	EINT4	EINT3	EINT2	EINT1	EINT0
7	6	5	4	3	2	1	0
TAMPER	RTC	Reserved	CLKFAIL	SRAMFAIL	PWRWK	IRC	BOD

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15]	**EINT7**	**External Interrupt 7 Interrupt Flag (Read Only)** 0 = External Interrupt 7 interrupt is deasserted. 1 = External Interrupt 7 interrupt is asserted.
[14]	**EINT6**	**External Interrupt 6 Interrupt Flag (Read Only)** 0 = External Interrupt 6 interrupt is deasserted. 1 = External Interrupt 6 interrupt is asserted.
[13]	**EINT5**	**External Interrupt 5 Interrupt Flag (Read Only)** 0 = External Interrupt 5 interrupt is deasserted. 1 = External Interrupt 5 interrupt is asserted.
[12]	**EINT4**	**External Interrupt 4 Interrupt Flag (Read Only)** 0 = External Interrupt 4 interrupt is deasserted. 1 = External Interrupt 4 interrupt is asserted.
[11]	**EINT3**	**External Interrupt 3 Interrupt Flag (Read Only)** 0 = External Interrupt 3 interrupt is deasserted. 1 = External Interrupt 3 interrupt is asserted.
[10]	**EINT2**	**External Interrupt 2 Interrupt Flag (Read Only)** 0 = External Interrupt 2 interrupt is deasserted. 1 = External Interrupt 2 interrupt is asserted.
[9]	**EINT1**	**External Interrupt 1 Interrupt Flag (Read Only)** 0 = External Interrupt 1 interrupt is deasserted. 1 = External Interrupt 1 interrupt is asserted.
[8]	**EINT0**	**External Interrupt 0 Interrupt Flag (Read Only)** 0 = External Interrupt 0 interrupt is deasserted. 1 = External Interrupt 0 interrupt is asserted.

[7]	**TAMPER**	**TAMPER_INT Interrupt Flag (Read Only)** 0 = Backup register tamper detected interrupt is deasserted. 1 = Backup register tamper detected interrupt is asserted.
[6]	**RTC**	**RTC Interrupt Flag (Read Only)** 0 = RTC interrupt is deasserted. 1 = RTC interrupt is asserted.
[5]	**Reserved**	Reserved.
[4]	**CLKFAIL**	**Clock Fail Detected Interrupt Flag (Read Only)** 0 = Clock fail detected interrupt is deasserted. 1 = Clock fail detected interrupt is asserted.
[3]	**SRAMFAIL**	**SRAM ParityCheck Error Interrupt Flag (Read Only)** 0 = SRAM parity check error interrupt is deasserted. 1 = SRAM parity check error interrupt is asserted.
[2]	**PWRWK**	**Power-down Mode Wake-up Interrupt Flag (Read Only)** 0 = Power-down mode wake-up interrupt is deasserted. 1 = Power-down mode wake-up interrupt is asserted.
[1]	**IRC**	**IRC TRIM Interrupt Flag (Read Only)** 0 = HIRC TRIM interrupt is deasserted. 1 = HIRC TRIM interrupt is asserted.
[0]	**BOD**	**BOD Interrupt Flag (Read Only)** 0 = BOD interrupt is deasserted. 1 = BOD interrupt is asserted.

6.2.8 System Control Register Map and Description

The Cortex®-M4 status and operation mode control are managed by System Control Registers. Including CPUID, Cortex®-M4 interrupt priority and power management can be controlled through these system control register

For more detailed information, please refer to the "*ARM® Cortex®-M4 Technical Reference Manual*" and "*ARM® v6-M Architecture Reference Manual*".

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
SCS Base Address: **SCS_BA = 0xE000_E000**				
ICSR	SCS_BA+0xD04	R/W	Interrupt Control and State Register	0x0000_0000
AIRCR	SCS_BA+0xD0C	R/W	Application Interrupt and Reset Control Register	0xFA05_0000
SCR	SCS_BA+0xD10	R/W	System Control Register	0x0000_0000
SHPR1	SCS_BA+0xD18	R/W	System Handler Priority Register 1	0x0000_0000
SHPR2	SCS_BA+0xD1C	R/W	System Handler Priority Register 2	0x0000_0000
SHPR3	SCS_BA+0xD20	R/W	System Handler Priority Register 3	0x0000_0000

Interrupt Control State Register (ICSR)

Register	Offset	R/W	Description	Reset Value
ICSR	SCS_BA+0xD04	R/W	Interrupt Control and State Register	0x0000_0000

31	30	29	28	27	26	25	24
NMIPENDSET	Reserved		PENDSVSET	PENDSVCLR	PENDSTSET	PENDSTCLR	Reserved
23	22	21	20	19	18	17	16
ISRPREEMPT	ISRPENDING	Reserved				VECTPENDING[5:4]	
15	14	13	12	11	10	9	8
VECTPENDING[3:0]				Reserved			
7	6	5	4	3	2	1	0
Reserved		VECTACTIVE[5:0]					

Bits	Description	
[31]	**NMIPENDSET**	**NMI Set-Pending Bit** Write: 0 = No effect. 1 = Changes NMI exception state to pending. Read: 0 = NMI exception is not pending. 1 = NMI exception is pending. **Note:** Because NMI is the highest-priority exception, normally the processor enters the NMI exception handler as soon as it detects a write of 1 to this bit. Entering the handler then clears this bit to 0. This means a read of this bit by the NMI exception handler returns 1 only if the NMI signal is reasserted while the processor is executing that handler.
[30:29]	**Reserved**	Reserved.
[28]	**PENDSVSET**	**PendSV Set-Pending Bit** Write: 0 = No effect. 1 = Changes PendSV exception state to pending. Read: 0 = PendSV exception is not pending. 1 = PendSV exception is pending. **Note:** Writing 1 to this bit is the only way to set the PendSV exception state to pending.
[27]	**PENDSVCLR**	**PendSV Clear-Pending Bit** Write: 0 = No effect. 1 = Removes the pending state from the PendSV exception. **Note:** This is a write only bit. To clear the PENDSV bit, you must “write 0 to PENDSVSET and write 1 to PENDSVCLR” at the same time.

[26]	**PENDSTSET**	**SysTick Exception Set-Pending Bit** Write: 0 = No effect. 1 = Changes SysTick exception state to pending. Read: 0 = SysTick exception is not pending. 1 = SysTick exception is pending.
[25]	**PENDSTCLR**	**SysTick Exception Clear-Pending Bit** Write: 0 = No effect. 1 = Removes the pending state from the SysTick exception. **Note:** This is a write only bit. To clear the PENDST bit, you must "write 0 to PENDSTSET and write 1 to PENDSTCLR" at the same time.
[24]	**Reserved**	Reserved.
[23]	**ISRPREEMPT**	**Interrupt Preempt Bit (Read only)** If set, a pending exception will be serviced on exit from the debug halt state.
[22]	**ISRPENDING**	**Interrupt Pending Flag, Excluding NMI and Faults (Read only)** 0 = Interrupt not pending. 1 = Interrupt pending.
[21:18]	**Reserved**	Reserved.
[17:12]	**VECTPENDING**	**Number of the Highest Pended Exception** Indicate the Exception Number of the Highest Priority Pending Enabled Exception 0 = no pending exceptions. Nonzero = the exception number of the highest priority pending enabled exception. The value indicated by this field includes the effect of the BASEPRI and FAULTMASK registers, but not any effect of the PRIMASK register.
[11]	**RETTOBASE**	**Preempted Active Exceptions indicator** Indicate whether There are Preempted Active Exceptions 0 = there are preempted active exceptions to execute. 1 = there are no active exceptions, or the currently-executing exception is the only active exception.
[10:6]	**Reserved**	Reserved.
[5:0]	**VECTACTIVE**	**Contains The Active Exception Number** 0 = Thread mode. Non-zero = The exception number of the currently active exception.

Application Interrupt and Reset Control Register (AIRCR)

Register	Offset	R/W	Description	Reset Value
AIRCR	SCS_BA+0xD0C	R/W	Application Interrupt and Reset Control Register	0xFA05_0000

31	30	29	28	27	26	25	24
VECTORKEY[15:8]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
VECTORKEY[7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ENDIANNESS	**Reserved**				**PRIGROUP**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**SYSRESETREQ**	**VECTCLKACTIVE**	**VECTRESET**

Bits	Description	
[31:16]	**VECTORKEY**	**Register Access Key** When writing this register, this field should be 0x05FA, otherwise the write action will be unpredictable. The VECTORKEY filed is used to prevent accidental write to this register from resetting the system or clearing of the exception status.
[15]	**ENDIANNESS**	**Data Endianness** 0 = Little-endian. 1 = Big-endian.
[14:11]	**Reserved**	Reserved.
[10:8]	**PRIGROUP**	**Interrupt Priority Grouping** This field determines the Split Of Group priority from subpriority,
[7:3]	**Reserved**	Reserved.
[2]	**SYSRESETREQ**	**System Reset Request** Writing This Bit to 1 Will Cause A Reset Signal To Be Asserted To The Chip And Indicate A Reset Is Requested This bit is write only and self-cleared as part of the reset sequence.
[1]	**VECTCLRACTIVE**	**Exception Active Status Clear Bit** Setting This Bit To 1 Will Clears All Active State Information For Fixed And Configurable Exceptions This bit is write only and can only be written when the core is halted. **Note:** It is the debugger's responsibility to re-initialize the stack.
[0]	**VECTRESET**	**Reserved for Debug Use** This bit is read as 0. You must write 0 to this bit, otherwise the behavior is Unpredictable

PRIGROUP	Binary Point	Group Priority Bits	Subpriority Bits	Number Of Group Priorities	Subpriorities
0b000	bxxxxxxx.y	[7:1]	[0]	128	2
0b001	bxxxxxx.yy	[7:2]	[1:0]	64	4
0b010	bxxxxx.yyy	[7:3]	[2:0]	32	8
0b011	bxxxx.yyyy	[7:4]	[3;0]	16	16
0b100	bxxx.yyyyy	[7:5]	[4:0]	8	32
0b101	bxx.yyyyyy	[7:6]	[5:0]	4	64
0b110	bx.yyyyyyy	[7]	[6:0]	2	128
0b111	b.yyyyyyyy	None	[7:0]	1	256

Table 6.2-4 Priority Grouping

System Control Register (SCR)

Register	Offset	R/W	Description	Reset Value
SCR	SCS_BA+0xD10	R/W	System Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			SEVONPEND	Reserved	SLEEPDEEP	SLEEPONEXIT	Reserved

Bits	Description	
[31:5]	**Reserved**	Reserved.
[4]	**SEVONPEND**	**Send Event On Pending** 0 = Only enabled interrupts or events can wake up the processor, while disabled interrupts are excluded. 1 = Enabled events and all interrupts, including disabled interrupts, can wake up the processor. When an event or interrupt enters pending state, the event signal wakes up the processor from WFE. If the processor is not waiting for an event, the event is registered and affects the next WFE. The processor also wakes up on execution of an SEV instruction or an external event.
[3]	**Reserved**	Reserved.
[2]	**SLEEPDEEP**	**Processor Deep Sleep and Sleep Mode Selection** Control Whether the Processor Uses Sleep Or Deep Sleep as its Low Power Mode. 0 = Sleep. 1 = Deep sleep.
[1]	**SLEEPONEXIT**	**Sleep-on-exit Enable Control** This bit indicate Sleep-On-Exit when Returning from Handler Mode to Thread Mode. 0 = Do not sleep when returning to Thread mode. 1 = Enters sleep, or deep sleep, on return from an ISR to Thread mode. Setting this bit to 1 enables an interrupt driven application to avoid returning to an empty main application.
[0]	**Reserved**	Reserved.

System Handler Priority Register 1 (SHPR1)

Register	Offset	R/W	Description	Reset Value
SHPR1	SCS_BA+0xD18	R/W	System Handler Priority Register 1	0x0000_0000

31	30	29	28	27	26	25	24
PRI_11		**Reserved**					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PRI_6							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PRI_5							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRI_4							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**PRI_6**	Priority of system handler 6, UsageFault
[15:8]	**PRI_5**	Priority of system handler 5, BusFault
[7:0]	**PRI_4**	Priority of system handler 4, MemManage

System Handler Priority Register 2 (SHPR2)

Register	Offset	R/W	Description	Reset Value
SHPR2	SCS_BA+0xD1C	R/W	System Handler Priority Register 2	0x0000_0000

31	30	29	28	27	26	25	24
PRI_11		**Reserved**					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:30]	**PRI_11**	**Priority Of System Handler 11 – SVCall** "0" denotes the highest priority and "3" denotes the lowest priority.
[29:0]	**Reserved**	Reserved.

System Handler Priority Register 3 (SHPR3)

Register	Offset	R/W	Description	Reset Value
SHPR3	SCS_BA+0xD20	R/W	System Handler Priority Register 3	0x0000_0000

31	30	29	28	27	26	25	24
PRI_15		**Reserved**					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PRI_14		**Reserved**					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:30]	**PRI_15**	**Priority Of System Handler 15 – SysTick** "0" denotes the highest priority and "3" denotes the lowest priority.
[29:24]	**Reserved**	Reserved.
[23:22]	**PRI_14**	**Priority Of System Handler 14 – PendSV** "0" denotes the highest priority and "3" denotes the lowest priority.
[21:0]	**Reserved**	Reserved.

6.3 Clock Controller

6.3.1 Overview

The clock controller generates clocks for the whole chip, including system clocks and all peripheral clocks. The clock controller also implements the power control function with the individually clock ON/OFF control, clock source selection and a clock divider. The chip will not enter Power-down mode until CPU sets the power-down enable bit (PWR_DOWN_EN) and Cortex®-M4 core executes the WFI/WFE instruction. After that, chip enters Power-down mode and wait for wake-up interrupt source triggered to leave Power-down mode. In Power-down mode, the clock controller turns off the 4~24 MHz external high-speed crystal (HXT) and 22.1184 MHz internal high-speed oscillator (HIRC) to reduce the overall system power consumption. The following figures show the clock generator and the overview of the clock source control.

The clock generator consists of 6 clock sources as listed below:

- 4~24 MHz external high speed crystal oscillator (HXT)
- Programmable PLL output clock frequency (PLL source can be selected from 4~24 MHz external high speed crystal (HXT) or 22.1184 MHz internal high speed oscillator (HIRC)) (PLL FOUT)
- 22.1184 MHz internal high speed RC oscillator (HIRC)
- 10 kHz internal low speed RC oscillator (LIRC)
- 32.768 kHz external low speed crystal oscillator (LXT)
- USB PHY's PLL output clock frequency (PLL2FOUT)

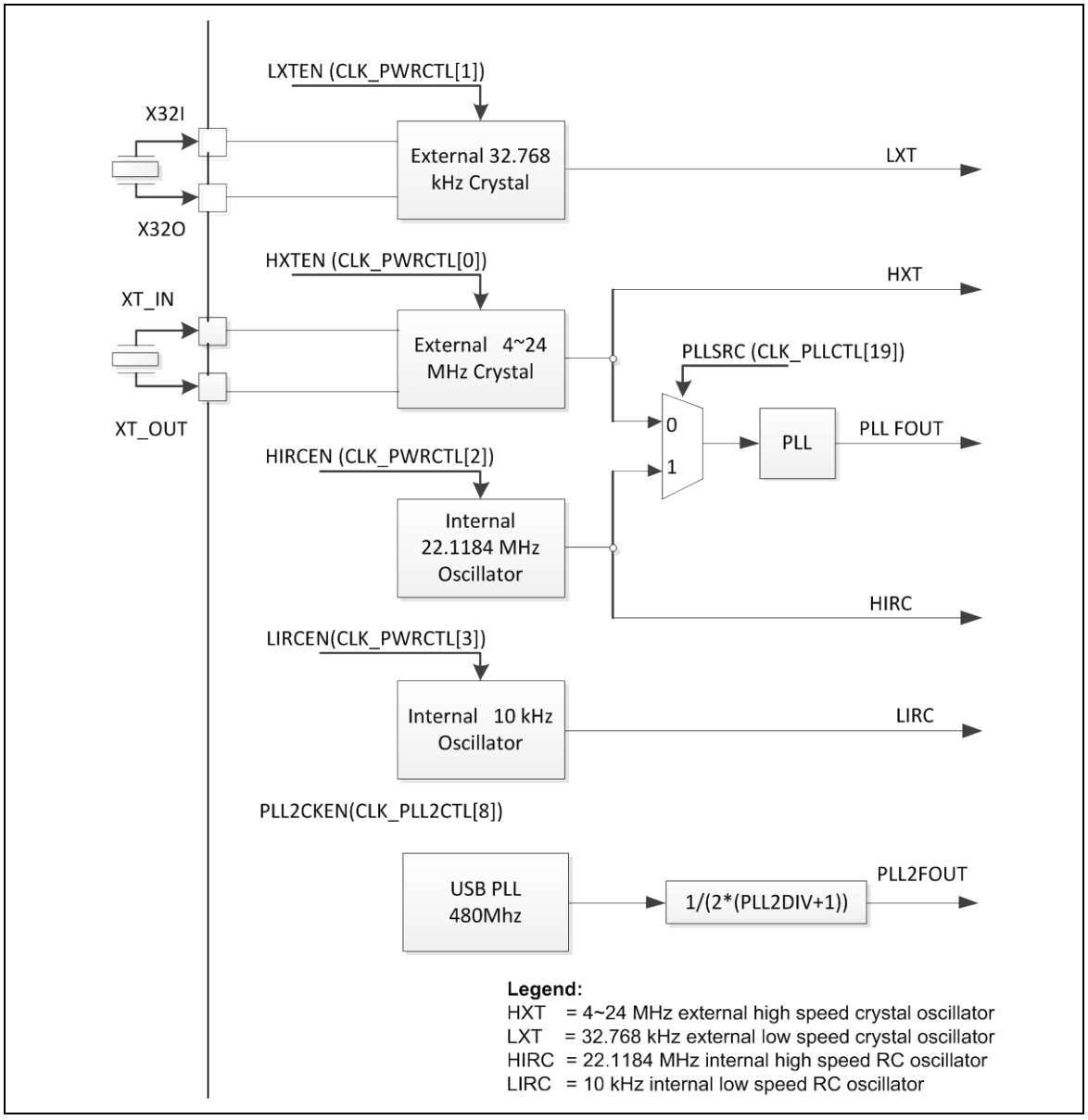


Figure 6.3-1 Clock Generator Block Diagram

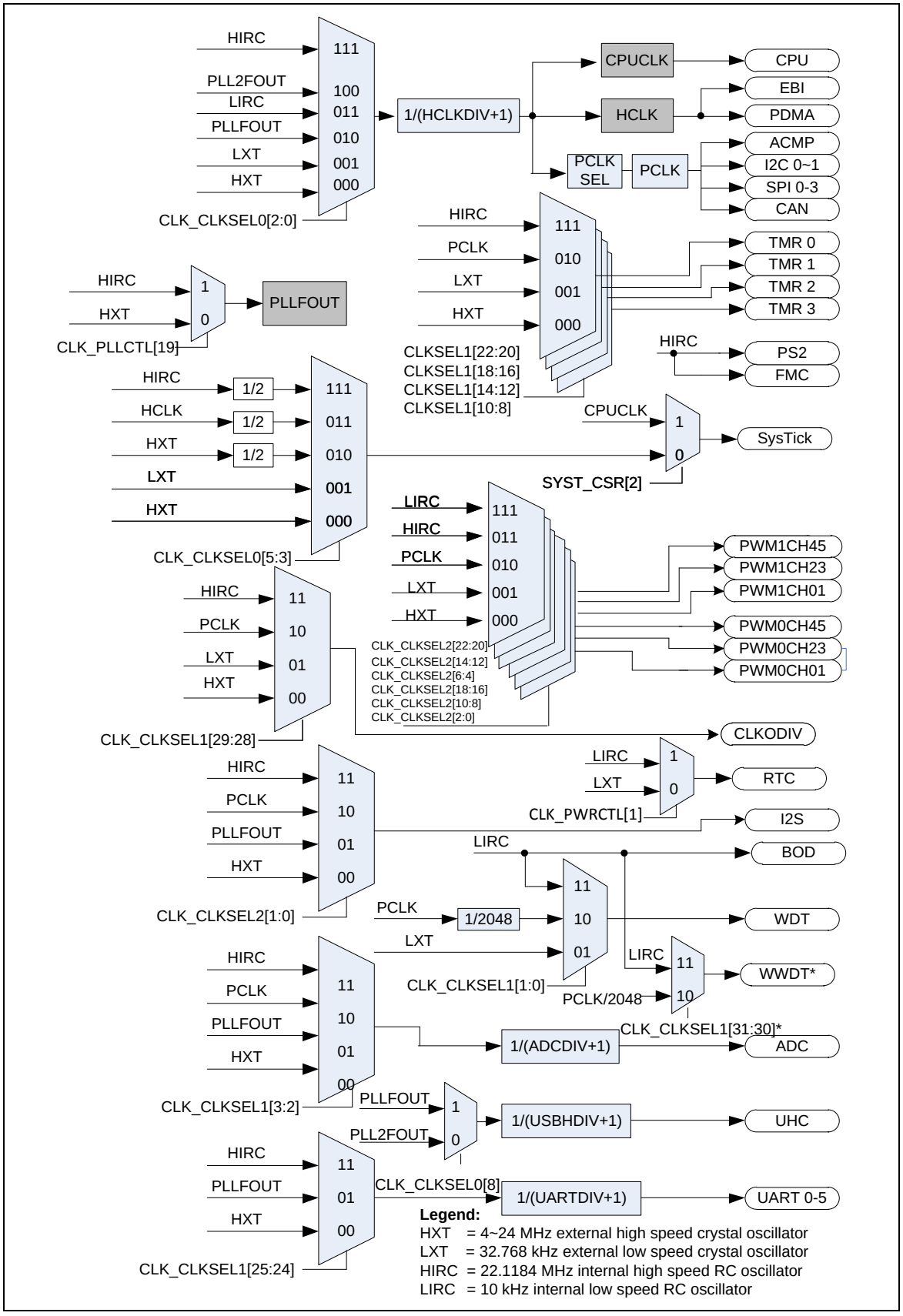

HIRC
PLL2FOUT
LIRC
PLLFOUT
LXT
HXT
111
100
011
010
001
000
CLK_CLKSEL0[2:0]
1/(HCLKDIV+1)
CPUCLK
HCLK
PCLK SEL
PCLK
CPU
EBI
PDMA
ACMP
I2C 0~1
SPI 0-3
CAN
HIRC
PCLK
LXT
HXT
111
010
001
000
CLKSEL1[22:20]
CLKSEL1[18:16]
CLKSEL1[14:12]
CLKSEL1[10:8]
TMR 0
TMR 1
TMR 2
TMR 3
HIRC
HXT
1
0
PLLFOUT
CLK_PLLCTL[19]
HIRC
PS2
FMC
HIRC
HCLK
HXT
LXT
HXT
1/2
1/2
1/2
111
011
010
001
000
CLK_CLKSEL0[5:3]
CPUCLK
1
0
SysTick
SYST_CSR[2]
LIRC
HIRC
PCLK
LXT
HXT
111
011
010
001
000
CLK_CLKSEL2[22:20]
CLK_CLKSEL2[14:12]
CLK_CLKSEL2[6:4]
CLK_CLKSEL2[18:16]
CLK_CLKSEL2[10:8]
CLK_CLKSEL2[2:0]
PWM1CH45
PWM1CH23
PWM1CH01
PWM0CH45
PWM0CH23
PWM0CH01
HIRC
PCLK
LXT
HXT
11
10
01
00
CLK_CLKSEL1[29:28]
CLKODIV
HIRC
PCLK
PLLFOUT
HXT
11
10
01
00
CLK_CLKSEL2[1:0]
LIRC
LXT
1
0
RTC
CLK_PWRCTL[1]
I2S
LIRC
BOD
PCLK
1/2048
LXT
11
10
01
CLK_CLKSEL1[1:0]
WDT
LIRC
11
10
PCLK/2048
WWDT*
HIRC
PCLK
PLLFOUT
HXT
11
10
01
00
CLK_CLKSEL1[3:2]
1/(ADCDIV+1)
CLK_CLKSEL1[31:30]*
ADC
PLLFOUT
PLL2FOUT
1
0
1/(USBHDIV+1)
UHC
HIRC
PLLFOUT
HXT
11
01
00
CLK_CLKSEL1[25:24]
CLK_CLKSEL0[8]
1/(UARTDIV+1)
UART 0-5
Legend:
HXT = 4~24 MHz external high speed crystal oscillator
LXT = 32.768 kHz external low speed crystal oscillator
HIRC = 22.1184 MHz internal high speed RC oscillator
LIRC = 10 kHz internal low speed RC oscillator

Figure 6.3-2 Clock Generator Global View Diagram

6.3.2 System Clock and SysTick Clock

The system clock has 5 clock sources, which were generated from clock generator block. The clock source switch depends on the register HCLKSEL (CLK_CLKSEL0 [2:0]). The block diagram is shown in the following figure.

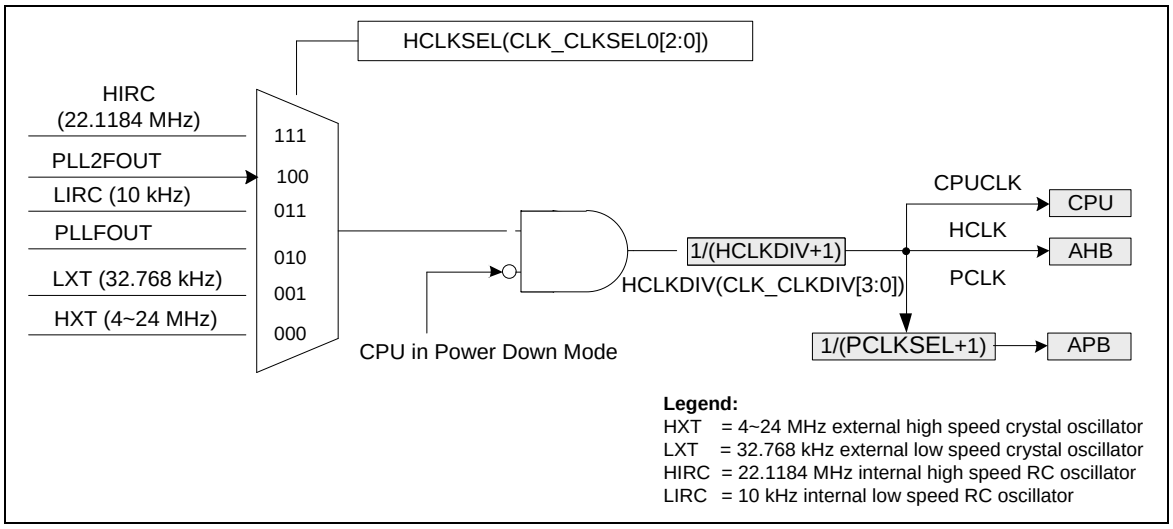


Figure 6.3-2 System Clock Block Diagram

6.3.3 Clock Monitor

The system clock has auto clock switch function to prevent system clock from being stopped. There are two clock detectors to monitor CPUCLK and HIRC and they have individual enable and interrupt control. The clock switch procedure is shown in the following figure. When any one detector is enabled, the LXT clock is enabled automatically. When the HIRC clock detector is enabled, the HIRC clock is enabled automatically.

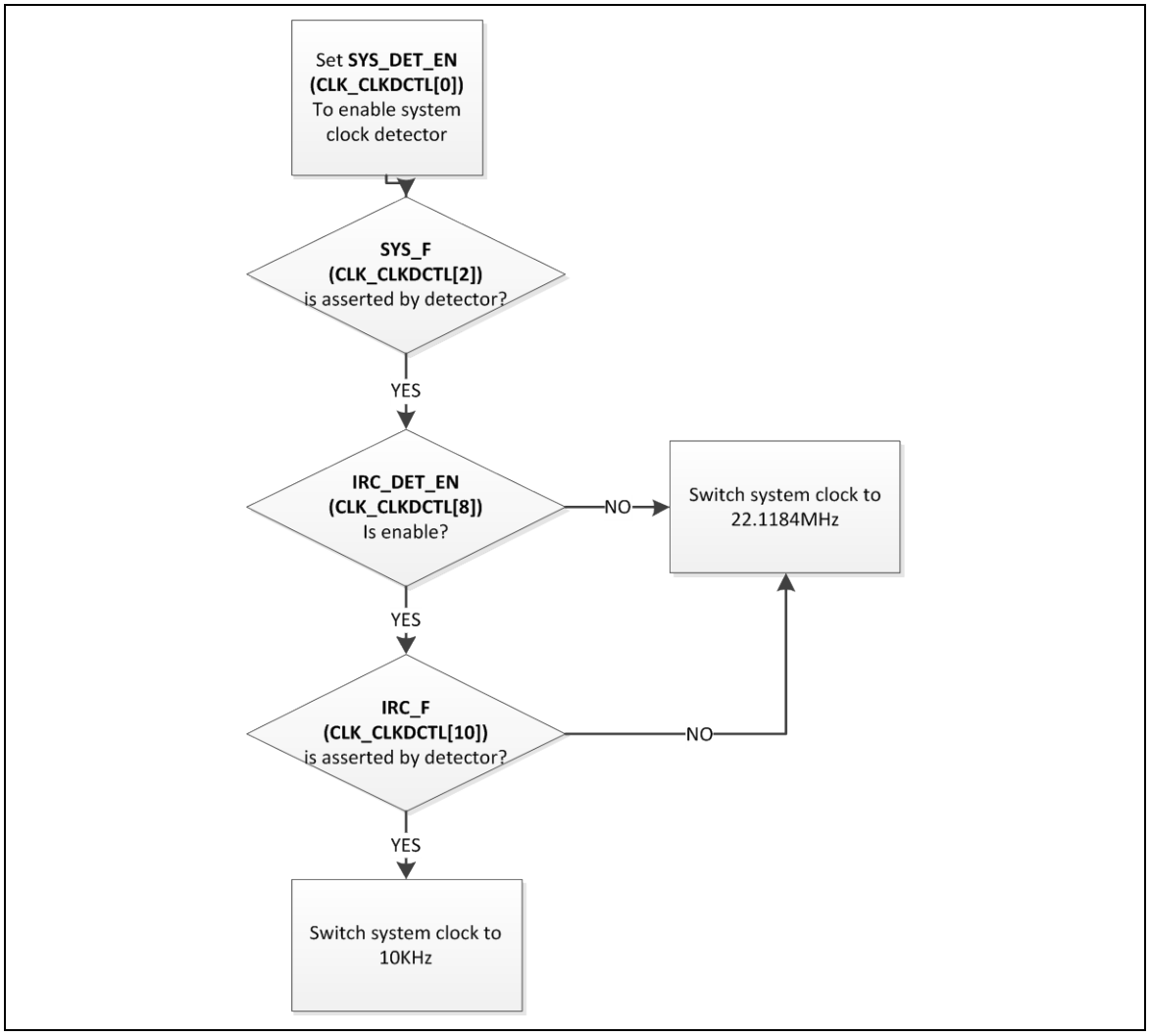


Figure 6.3-3 System Clock Switch Procedure

The clock source of SysTick in Cortex®-M4 core can use CPU clock or external clock (SYST_CSR[2]). If using external clock, the SysTick clock (STCLK) has 5 clock sources. The clock source switch depends on the setting of the register STCLKSEL (CLK_CLKSEL0[5:3]). The block diagram is shown in the following figure.

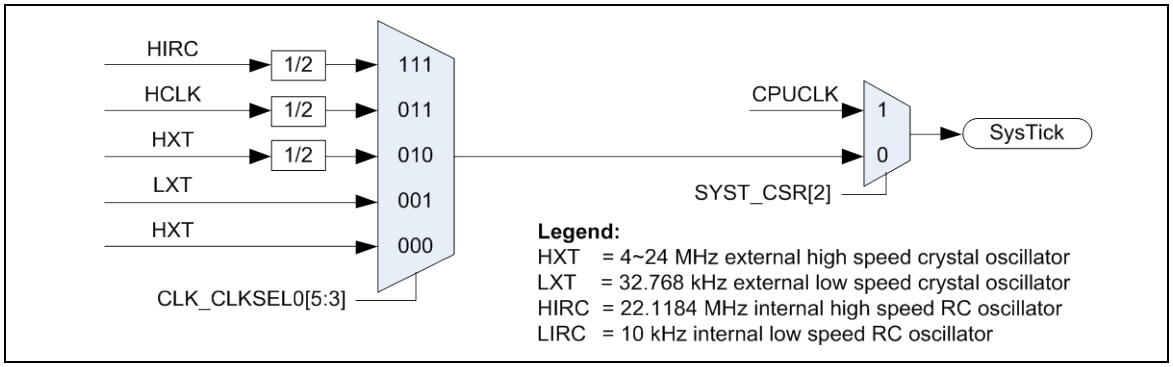


Figure 6.3-4 SysTick Clock Control Block Diagram

6.3.4 Peripherals Clock

The peripherals clock had different clock source switch setting, which depends on the different peripheral. Please refer the CLK_CLKSEL1 and CLK_CLKSEL2 register description in 5.3.7.

6.3.5 Power-down Mode Clock

When entering Power-down mode, system clocks, some clock sources, and some peripheral clocks are disabled. Some clock sources and peripherals clock are still active in Power-down mode.

For theses clocks, which still keep active, are listed below:

- Clock Generator
  - 10 kHz internal low-speed oscillator(LIRC) clock
  - 32.768 kHz external low-speed crystal (LXT)clock
- Peripherals Clock (When these IP adopt 32.768 kHz external or 10 kHz low-speed oscillator as clock source)

6.3.6 Frequency Divider Output

This device is equipped with a power-of-2 frequency divider which is composed by16 chained divide-by-2 shift registers. One of the 16 shift register outputs selected by a sixteen to one multiplexer is reflected to CLKO function pin. Therefore there are 16 options of power-of-2 divided clocks with the frequency from $F_{in}/2^1$ to $F_{in}/2^{16}$ where Fin is input clock frequency to the clock divider.

The output formula is $\mathbf{F_{out} = F_{in}/2^{(N+1)}}$, where F_{in} is the input clock frequency, F_{out} is the clock divider output frequency and N is the 4-bit value in FSEL (CLK_CLKOCTL[3:0]).

When writing 1 to FDIVEN (CLK_CLKOCTL[4]), the chained counter starts to count. When writing 0 to FDIVEN (CLK_CLKOCTL[4]), the chained counter continuously runs till divided clock reaches low state and stay in low state.

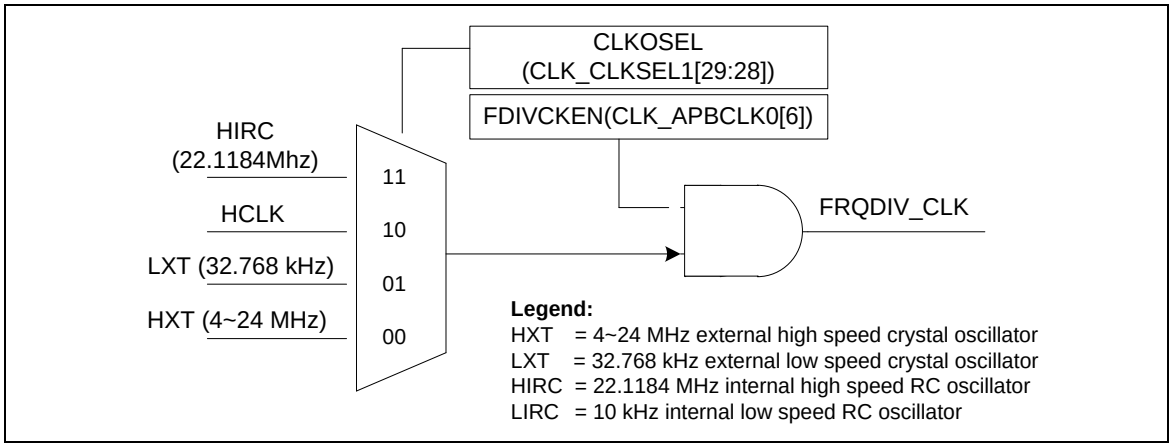


Figure 6.3-5 Clock Source of Frequency Divider

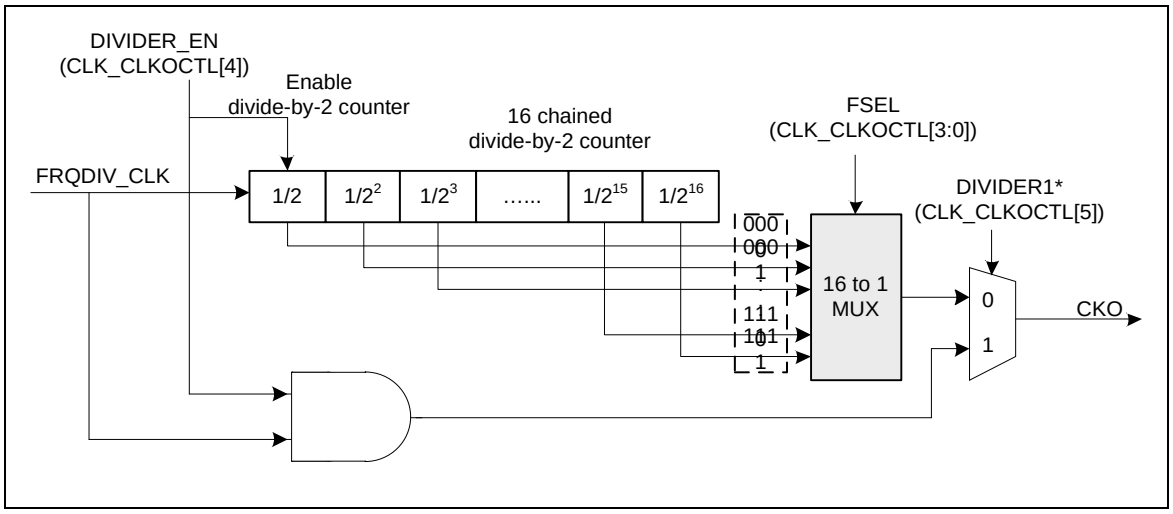


Figure 6.3-6 Block Diagram of Frequency Divider

6.3.7 Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
CLK Base Address: **CLK_BA = 0x4000_0200**				
CLK_PWRCTL	CLK_BA+0x00	R/W	System Power-down Control Register	0x0000_001X
CLK_AHBCLK	CLK_BA+0x04	R/W	AHB Devices Clock Enable Control Register	0x0000_0005
CLK_APBCLK0	CLK_BA+0x08	R/W	APB Devices Clock Enable Control Register 0	0x0000_0001
CLK_APBCLK1	CLK_BA+0x0C	R/W	APB Devices Clock Enable Control Register 1	0x0000_0000
CLK_CLKSEL0	CLK_BA+0x10	R/W	Clock Source Select Control Register 0	0x0033_053X
CLK_CLKSEL1	CLK_BA+0x14	R/W	Clock Source Select Control Register 1	0xB377_77FF
CLK_CLKSEL2	CLK_BA+0x18	R/W	Clock Source Select Control Register 2	0x0077_7777
CLK_CLKSEL3	CLK_BA+0x1C	R/W	Clock Source Select Control Register 3	0x000F_0FFF
CLK_CLKDIV0	CLK_BA+0x20	R/W	Clock Divider Number Register 0	0x0000_0000
CLK_CLKDIV1	CLK_BA+0x24	R/W	Clock Divider Number Register 1	0x0000_0000
CLK_CLKDIV2	CLK_BA+0x28	R/W	Clock Divider Number Register 2	0x0000_0000
CLK_CLKDIV3	CLK_BA+0x2C	R/W	Clock Divider Number Register 3	0x0000_0000
CLK_PLLCTL	CLK_BA+0x40	R/W	PLL Control Register	0x0005_8228
CLK_PLL2CTL	CLK_BA+0x44	R/W	PLL2 Control Register	0x0000_0000
CLK_STATUS	CLK_BA+0x50	R/W	Clock Status Monitor Register	0x0000_00XX
CLK_CLKOCTL	CLK_BA+0x60	R/W	Frequency Divider Control Register	0x0000_0000
CLK_CLKDCTL	CLK_BA+0x70	R/W	Clock Fail Detector Control Register	0x0000_0100

6.3.8 Register Description

Power-down Control Register (CLK_PWRCTL)

Except the BIT[6], all the other bits are protected, program these bits need to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register REGWRPROT at address GCR_BA+0x100.

Register	Offset	R/W	Description	Reset Value
CLK_PWRCTL	CLK_BA+0x00	R/W	System Power-down Control Register	0x0000_001X

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved						DBPDEN	PDWTCPU
7	6	5	4	3	2	1	0
PDEN	PDWKIF	PDWKIEN	PDWKDLY	LIRCEN	HIRCEN	LXTEN	HXTEN

Bits	Description	
[31:10]	**Reserved**	Reserved.
[9]	**DBPDEN**	**Chip Entering Power-Down Even ICE Connected** 0 = Chip enters power-down disabled in Debug mode. 1 = Chip enters power-down enabled in Debug mode.
[8]	**PDWTCPU**	**This Bit Control The Power-Down Entry Condition (Write Protect)** 0 = Chip enters Power-down mode when the PWR_DOWN_EN bit is set to 1. 1 = Chip enters Power-down mode when the both PDEN and PWR_DOWN_EN bits are set to 1 and CPU run WFI instruction.
[7]	**PDEN**	**System Power-Down Enable Bit (Write Protect)** When this bit is set to 1, Power-down mode is enabled and chip power-down behavior will depend on the PDEN bit. (a) If the PDEN is 0, then the chip enters Power-down mode immediately after the PWR_DOWN_EN bit set. (default) (b) if the PDEN is 1, then the chip keeps active till the CPU sleep mode is also active and then the chip enters Power-down mode When chip wakes up from Power-down mode, this bit is auto cleared. Users need to set this bit again for next power-down. In Power-down mode, HXT and the HIRC will be disabled in this mode, but the LXT and LIRC are not controlled by Power-down mode. In Power-down mode, the PLL and system clock are disabled, and ignored the clock source selection. The clocks of peripheral are not controlled by Power-down mode, if the peripheral clock source is from LXT or the LIRC. 0 = Chip operating normally or chip in idle mode by WFI command.

		1 = Chip enters Power-down mode instant or waits CPU sleep command WFI.
[6]	**PDWKIF**	**Power-Down Mode Wake-Up Interrupt Status** Set by “power-down wake-up event”, it indicates that resume from Power-down mode” The flag is set if the GPIO, USB, UART, WDT, CAN, ACMP, BOD, RTC or SDHOST wake-up occurred **Note1:** Write 1 to clear the bit to 0. **Note2:** This bit works only if PDWKIEN (CLK_PWRCTL[5]) set to 1.
[5]	**PDWKIEN**	**Power-Down Mode Wake-Up Interrupt Enable Bit (Write Protect)** 0 = Power-down Mode Wake-up Interrupt Disabled. 1 = Power-down Mode Wake-up Interrupt Enabled. **Note:** The interrupt will occur when both PDWKIF and PDWKIEN are high.
[4]	**PDWKDLY**	**Wake-Up Delay Counter Enable Bit (Write Protect)** When the chip wakes up from Power-down mode, the clock control will delay certain clock cycles to wait system clock stable. The delayed clock cycle is 4096 clock cycles when chip work at HXT, and 256 clock cycles when chip works at HIRC. 0 = Clock cycles delay Disabled. 1 = Clock cycles delay Enabled.
[3]	**LIRCEN**	**10 KHz Internal Low-Speed Oscillator(LIRC) Enable Bit (Write Protect)** 0 = LIRC Disabled. 1 = LIRC Enabled (default 1).
[2]	**HIRCEN**	**22.1184 MHz Internal High-Speed Oscillator Clock (HIRC) Enable Bit (Write Protect)** 0 = HIRC Disabled. 1 = HIRC Enabled.
[1]	**LXTEN**	**32.768 KHz External Low-Speed Crystal Clock(LXT) Enable Bit (Write Protect)** 0 = LXT Disabled. 1 = LXT (Normal operation) Enabled.
[0]	**HXTEN**	**4~24 MHz External High-Speed Crystal Clock(HXT) Enable Bit (Write Protect)** The bit default value is set by flash controller user configuration register config0 [26:24]. When the default clock source is from 4~24 MHz external high-speed crystal, this bit is set to 1 automatically 0 = HXT Disabled. 1 = HXT Enabled.

Register/Instruction Mode	PWR_DOWN_EN	PDEN	CPU Run WFI Instruction	Clock Disable
Normal operation	0	0	NO	All clocks are disabled by control register.
Idle mode **(CPU enters Sleep mode)**	0	0	YES	Only CPU clock is disabled.
Power-down mode	1	0	NO	Most clocks are disabled except 10 kHz/32.768 kHz; only RTC/WDT/Timer/PWM peripheral clock are still enabled.
Power-down mode **(CPU enters Deep Sleep mode)**	1	1	YES	Most clocks are disabled except 10 kHz/32.768 kHz, only RTC/WDT/Timer/PWM peripheral clock are still enabled.

Table 6.3-1 Power-down Mode Control Table

When the chip enters Power-down mode, user can wake up chip by some interrupt sources. User should enable the related interrupt sources and NVIC IRQ enable bits (NVIC_ISER) before set PWR_DOWN_EN bit in CLK_PWRCTL[7] to ensure chip can enter power-down and wake up successfully.

AHB Devices Clock Enable Control Register (CLK_AHBCLK)

These bits for this register are used to enable/disable clock for system clock, AHB bus devices clock.

Register	Offset	R/W	Description	Reset Value
CLK_AHBCLK	CLK_BA+0x04	R/W	AHB Devices Clock Enable Control Register	0x0000_0005

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved			CRYPTOCKEN	Reserved	USBDCKEN	SENCKEN	CAPCKEN
7	6	5	4	3	2	1	0
CRCCKEN	SDHCKEN	EMACCKEN	USBHCKEN	EBICKEN	ISPCKEN	PDMACKEN	Reserved

Bits	Description	
[31:13]	**Reserved**	Reserved.
[12]	**CRYPTOCKEN**	**Cryptographic Accelerator Clock Enable Bit** 0 = Cryptographic Accelerator clock Disabled. 1 = Cryptographic Accelerator clock Enabled.
[11]	**Reserved**	Reserved.
[10]	**USBDCKEN**	**USB 2.0 Device Clock Enable Bit** 0 = USB device controller's clock Disabled. 1 = USB device controller's clock Enabled.
[9]	**SENCKEN**	**Sensor Clock Enable Bit** 0 = Sensor clock Disabled. 1 = Sensor clock Enabled.
[8]	**CAPCKEN**	**Image Capture Interface Controller Clock Enable Bit** 0 = CAP controller's clock Disabled. 1 = CAP controller's clock Enabled.
[7]	**CRCCKEN**	**CRC Generator Controller Clock Enable Bit** 0 = CRC engine clock Disabled. 1 = CRC engine clock Enabled.
[6]	**SDHCKEN**	**SDHOST Controller Clock Enable Bit** 0 = SDHOST engine clock Disabled. 1 = SDHOST engine clock Enabled.
[5]	**EMACCKEN**	**Ethernet Controller Clock Enable Bit (NUC472 Only)** 0 = Ethernet Controller engine clock Disabled.

		1 = Ethernet Controller engine clock Enabled.
[4]	**USBHCKEN**	**USB HOST Controller Clock Enable Bit** 0 = USB HOST engine clock Disabled. 1 = USB HOST engine clock Enabled.
[3]	**EBICKEN**	**EBI Controller Clock Enable Bit** 0 = EBI engine clock Disabled. 1 = EBI engine clock Enabled.
[2]	**ISPCKEN**	**Flash ISP Controller Clock Enable Bit** 0 = Flash ISP engine clock Disabled. 1 = Flash ISP engine clock Enabled.
[1]	**PDMACKEN**	**PDMA Controller Clock Enable Bit** 0 = PDMA engine clock Disabled. 1 = PDMA engine clock Enabled.
[0]	**Reserved**	Reserved.

APB Devices Clock Enable Control Register (CLK_APBCLK0)

The bits of this register are used to enable/disable clock for peripheral controller clocks.

Register	Offset	R/W	Description	Reset Value
CLK_APBCLK0	CLK_BA+0x08	R/W	APB Devices Clock Enable Control Register 0	0x0000_0001

31	30	29	28	27	26	25	24
PS2CKEN	I2S1CKEN	I2S0CKEN	ADCCKEN	Reserved	OTGCKEN	CAN1CKEN	CAN0CKEN
23	22	21	20	19	18	17	16
Reserved		UART5CKEN	UART4CKEN	UART3CKEN	UART2CKEN	UART1CKEN	UART0CKEN
15	14	13	12	11	10	9	8
SPI3CKEN	SPI2CKEN	SPI1CKEN	SPI0CKEN	I2C3CKEN	I2C2CKEN	I2C1CKEN	I2C0CKEN
7	6	5	4	3	2	1	0
ACMPCKEN	CLKOCKEN	TMR3CKEN	TMR2CKEN	TMR1CKEN	TMR0CKEN	RTCCKEN	WDTCKEN

Bits	Description	
[31]	**PS2CKEN**	**PS/2 Clock Enable Bit** 0 = PS/2 clock Disabled. 1 = PS/2 clock Enabled.
[30]	**I2S1CKEN**	**I^{2}S1 Clock Enable Bit** 0 = I^{2}S1 Clock Disabled. 1 = I^{2}S1 Clock Enabled.
[29]	**I2S0CKEN**	**I^{2}S0 Clock Enable Bit** 0 = I^{2}S Clock Disabled. 1 = I^{2}S Clock Enabled.
[28]	**ADCCKEN**	**Analog-Digital-Converter (ADC) Clock Enable Bit** 0 = ADC clock Disabled. 1 = ADC clock Enabled.
[27]	**Reserved**	Reserved.
[26]	**OTGCKEN**	**USB 2.0 OTG Device Controller Clock Enable Bit** 0 = OTG clock Disabled. 1 = OTG clock Enabled.
[25]	**CAN1CKEN**	**CAN Bus Controller-1 Clock Enable Bit** 0 = CAN1 clock Disabled. 1 = CAN1 clock Enabled.
[24]	**CAN0CKEN**	**CAN Bus Controller-0 Clock Enable Bit** 0 = CAN0 clock Disabled. 1 = CAN0 clock Enabled.

[23:22]	**Reserved**	Reserved.
[21]	**UART5CKEN**	**UART5 Clock Enable Bit** 0 = UART5 clock Disabled. 1 = UART5 clock Enabled.
[20]	**UART4CKEN**	**UART4 Clock Enable Bit** 0 = UART4 clock Disabled. 1 = UART4 clock Enabled.
[19]	**UART3CKEN**	**UART3 Clock Enable Bit** 0 = UART3 clock Disabled. 1 = UART3 clock Enabled.
[18]	**UART2CKEN**	**UART2 Clock Enable Bit** 0 = UART2 clock Disabled. 1 = UART2 clock Enabled.
[17]	**UART1CKEN**	**UART1 Clock Enable Bit** 0 = UART1 clock Disabled. 1 = UART1 clock Enabled.
[16]	**UART0CKEN**	**UART0 Clock Enable Bit** 0 = UART0 clock Disabled. 1 = UART0 clock Enabled.
[15]	**SPI3CKEN**	**SPI3 Clock Enable Bit** 0 = SPI3 Clock Disabled. 1 = SPI3 Clock Enabled.
[14]	**SPI2CKEN**	**SPI2 Clock Enable Bit** 0 = SPI2 Clock Disabled. 1 = SPI2 Clock Enabled.
[13]	**SPI1CKEN**	**SPI1 Clock Enable Bit** 0 = SPI1 Clock Disabled. 1 = SPI1 Clock Enabled.
[12]	**SPI0CKEN**	**SPI0 Clock Enable Bit** 0 = SPI0 Clock Disabled. 1= SPI0 Clock Enabled.
[11]	**I2C3CKEN**	**I^2C3 Clock Enable Bit** 0 = I^2C3 Clock Disabled. 1 = I^2C3 Clock Enabled.
[10]	**I2C2CKEN**	**I^2C2 Clock Enable Bit** 0 = I^2C2 Clock Disabled. 1 = I^2C2 Clock Enabled.
[9]	**I2C1CKEN**	**I^2C1 Clock Enable Bit** 0 = I^2C1 Clock Disabled. 1 = I^2C1 Clock Enabled.
[8]	**I2C0CKEN**	**I^2C0 Clock Enable Bit** 0 = I^2C0 Clock Disabled.

		1 = I²C0 Clock Enabled.
[7]	**ACMPCKEN**	**Analog Comparator Clock Enable Bit** 0 = Analog Comparator Clock Disabled. 1 = Analog Comparator Clock Enabled.
[6]	**FDIVCKEN**	**Frequency Divider Output Clock Enable Bit** 0 = FDIV Clock Disabled. 1 = FDIV Clock Enabled.
[5]	**TMR3CKEN**	**Timer3 Clock Enable Bit** 0 = Timer3 Clock Disabled. 1 = Timer3 Clock Enabled.
[4]	**TMR2CKEN**	**Timer2 Clock Enable Bit** 0 = Timer2 Clock Disabled. 1 = Timer2 Clock Enabled.
[3]	**TMR1CKEN**	**Timer1 Clock Enable Bit** 0 = Timer1 Clock Disabled. 1 = Timer1 Clock Enabled.
[2]	**TMR0CKEN**	**Timer0 Clock Enable Bit** 0 = Timer0 Clock Disabled. 1 = Timer0 Clock Enabled.
[1]	**RTCCKEN**	**Real-Time-Clock APB Interface Clock Enable Bit** This bit is used to control the RTC APB clock only, The RTC engine clock source is from the 32.768 kHz external low-speed crystal. 0 = RTC Clock Disabled. 1 = RTC Clock Enabled.
[0]	**WDTCKEN**	**Watchdog Timer Clock Enable Bit (Write Protect)** This bit is the protected bit, which means programming this needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register REGWRPROT at address GCR_BA+0x100. 0 = Watchdog Timer Clock Disabled. 1 = Watchdog Timer Clock Enabled.

APB Devices Clock Enable Control Register 1 (CLK_APBCLK1)

The bits of this register are used to enable/disable clock for peripheral controller clocks.

Register	Offset	R/W	Description	Reset Value
CLK_APBCLK1	CLK_BA+0x0C	R/W	APB Devices Clock Enable Control Register 1	0x0000_0000

31	30	29	28	27	26	25	24
EADCCKEN	OPACKEN	EPWM1CKEN	EPWM0CKEN	ECAP1CKEN	ECAP0CKEN	Reserved	
23	22	21	20	19	18	17	16
QEI1CKEN	QEI0CKEN	PWM1CH45CKEN	PWM1CH23CKEN	PWM1CH01CKEN	PWM0CH45CKEN	PWM0CH23CKEN	PWM0CH01CKEN
15	14	13	12	11	10	9	8
Reserved							I2C4CKEN
7	6	5	4	3	2	1	0
Reserved		SC5CKEN	SC4CKEN	SC3CKEN	SC2CKEN	SC1CKEN	SC0CKEN

Bits	Description	
[31]	**EADCCKEN**	**Enhanced Analog-Digital-Converter (E ADC) Clock Enable Bit** 0 = EADC clock Disabled. 1 = EADC clock Enabled.
[30]	**OPACKEN**	**OP Amplifier (OPA) Clock Enable Bit** 0 = OPA clock Disabled. 1 = OPA clock Enabled.
[29]	**EPWM1CKEN**	**Enhanced PWM1 (EPWM) Clock Enable Bit** 0 = EPWM1 clock Disabled. 1 = EPWM1 clock Enabled.
[28]	**EPWM0CKEN**	**Enhanced PWM0 (EPWM) Clock Enable Bit** 0 = EPWM0 clock Disabled. 1 = EPWM0 clock Enabled.
[27]	**ECAP1CKEN**	**Enhanced CAP (ECAP1) Clock Enable Bit** 0 = ECAP1 clock Disabled. 1 = ECAP1 clock Enabled.
[26]	**ECAP0CKEN**	**Enhanced CAP (ECAP0) Clock Enable Bit** 0 = ECAP0 clock Disabled. 1 = ECAP0 clock Enabled.
[25:24]	**Reserved**	Reserved.
[23]	**QEI1CKEN**	**Quadrature Encoder Interface (QEI1) Clock Enable Bit** 0 = QEI1 clock Disabled. 1 = QEI1 clock Enabled.

[22]	**QEI0CKEN**	**Quadrature Encoder Interface (QEI0) Clock Enable Bit** 0 = QEI0 clock Disabled. 1 = QEI0 clock Enabled.
[21]	**PWM1CH45CKEN**	**PWM1_45 Clock Enable Bit** 0 = PWM1_45 Clock Disabled. 1 = PWM1_45 Clock Enabled.
[20]	**PWM1CH23CKEN**	**PWM1_23 Clock Enable Bit** 0 = PWM1_23 Clock Disabled. 1 = PWM1_23 Clock Enabled.
[19]	**PWM1CH01CKEN**	**PWM1_01 Clock Enable Bit** 0 = PWM1_01 Clock Disabled. 1 = PWM1_01 Clock Enabled.
[18]	**PWM0CH45CKEN**	**PWM0_45 Clock Enable Bit** 0 = PWM0_45 Clock Disabled. 1 = PWM0_45 Clock Enabled.
[17]	**PWM0CH23CKEN**	**PWM0_23 Clock Enable Bit** 0 = PWM0_23 Clock Disabled. 1 = PWM0_23 Clock Enabled.
[16]	**PWM0CH01CKEN**	**PWM0_01 Clock Enable Bit** 0 = PWM0_01 Clock Disabled. 1 = PWM0_01 Clock Enabled.
[15:9]	**Reserved**	Reserved.
[8]	**I2C4CKEN**	**I^2C4 Clock Enable Bit** 0 = I^2C4 Clock Disabled. 1 = I^2C4 Clock Enabled.
[7:6]	**Reserved**	Reserved.
[5]	**SC5CKEN**	**SC5 Clock Enable Bit** 0 = SC5 Clock Disabled. 1 = SC5 Clock Enabled.
[4]	**SC4CKEN**	**SC4 Clock Enable Bit** 0 = SC4 Clock Disabled. 1 = SC4 Clock Enabled.
[3]	**SC3CKEN**	**SC3 Clock Enable Bit** 0 = SC3 Clock Disabled. 1 = SC3 Clock Enabled.
[2]	**SC2CKEN**	**SC2 Clock Enable Bit** 0 = SC2 Clock Disabled. 1 = SC2 Clock Enabled.
[1]	**SC1CKEN**	**SC1 Clock Enable Bit** 0 = SC1 Clock Disabled. 1 = SC1 Clock Enabled.
[0]	**SC0CKEN**	**SC0 Clock Enable Bit**

		0 = SC0 Clock Disabled. 1 = SC0 Clock Enabled.

Clock Source Select Control Register 0 (CLK_CLKSEL0)

Register	Offset	R/W	Description	Reset Value
CLK_CLKSEL0	CLK_BA+0x10	R/W	Clock Source Select Control Register 0	0x0033_053X

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		SDHSEL		Reserved		ICAPSEL	
15	14	13	12	11	10	9	8
Reserved							USBHSEL
7	6	5	4	3	2	1	0
Reserved	PCLKSEL	STCLKSEL			HCLKSEL		

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21:20]	**SDHSEL**	**SDHOST Engine Clock Source Selection** These bits are protected bit. It means programming this bit needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register REGWRPROT at address GCR_BA+0x100. 00 = Clock source from HXT clock. 01 = Clock source from PLL clock. 10 = Clock source from HCLK. 11 = Clock source from HIRC clock.
[17:16]	**ICAPSEL**	**Image Capture Interface Clock Source Selection** These bits are protected bit. It means programming this bit needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register REGWRPROT at address GCR_BA+0x100. 00 = Clock source from HXT clock. 01 = Clock source from PLL clock. 10 = Clock source from HCLK. 11 = Clock source from HIRC clock.
[15:9]	**Reserved**	Reserved.
[8]	**USBHSEL**	**USB Host Clock Source Selection (Write Protect)** These bits are protected bit. It means programming this bit needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register REGWRPROT at address GCR_BA+0x100. 0 = Clock source from PLL2. 1 = Clock source from PLL.
[7]	**Reserved**	Reserved.
[6]	**PCLKSEL**	**PCLK Clock Source Selection (Write Protect)** These bits are protected bit. It means programming this bit needs to write “59h”, “16h”,

		"88h" to address 0x4000_0100 to disable register protection. Refer to the register REGWRPROT at address GCR_BA+0x100. 0 = Clock source from HCLK. 1 = Clock source from HCLK/2.
[5:3]	**STCLKSEL**	**Cortex® -M4 SysTick Clock Source Selection (Write Protect)** If SYST_CSR[2]=0, SysTick uses listed clock source below. These bits are protected bit. It means programming this bit needs to write "59h", "16h", "88h" to address 0x4000_0100 to disable register protection. Refer to the register REGWRPROT at address GCR_BA+0x100. 000 = Clock source from HXT clock. 001 = Clock source from LXT clock. 010 = Clock source from HXT clock/2. 011 = Clock source from HCLK/2. 111 = Clock source from HIRC clock/2.
[2:0]	**HCLKSEL**	**HCLK Clock Source Selection (Write Protect)** Before clock switching, the related clock sources (both pre-select and new-select) must be turned on 1. The 3-bit default value is reloaded from the value of CFOSC (Config0[26:24]) in user configuration register of Flash controller by any reset. Therefore the default value is either 000b or 111b. 2. These bits are protected bit, it means programming this bit needs to write "59h", "16h", "88h" to address 0x4000_0100 to disable register protection. Refer to the register REGWRPROT at address GCR_BA+0x100. 000 = Clock source from HXT clock. 001 = Clock source from LXT clock. 010 = Clock source from PLL clock. 011 = Clock source from LIRC clock. 100 = Clock source from PLL2 clock. 111 = Clock source from HIRC clock. Other = Reserved.

Clock Source Select Control Register 1 (CLK_CLKSEL1)

Before clock switching, the related clock sources (pre-selected and newly-selected) must be turned on.

Register	Offset	R/W	Description	Reset Value
CLK_CLKSEL1	CLK_BA+0x14	R/W	Clock Source Select Control Register 1	0xB377_77FF

31	30	29	28	27	26	25	24
WWDTSEL		CLKOSEL		Reserved		UARTSEL	
23	22	21	20	19	18	17	16
Reserved	TMR3SEL			Reserved	TMR2SEL		
15	14		12	11	10		
Reserved	TMR1SEL			Reserved	TMR0SEL		
7	6	5	4	3	2	1	0
SPI3SEL	SPI2SEL	SPI1SEL	SPI0SEL	EADCSEL		WDTSEL	

Bits	Description	
[31:30]	**WWDTSEL**	**Window Watchdog Timer Clock Source Selection** 00 = Reserved. 01 = Reserved. 10 = Clock source from **HCLK/**2048 clock. 11 = Clock source from LIRC clock.
[29:28]	**CLKOSEL**	**Clock Divider Clock Source Selection** 00 = Clock source from HXT clock. 01 = Clock source from LXT clock. 10 = Clock source from **HCLK.** 11 = Clock source from HIRC clock.
[27:26]	**Reserved**	Reserved.
[25:24]	**UARTSEL**	**UART Clock Source Selection** 00 = Clock source from HXT clock. 01 = Clock source from PLL clock. 10/11 = Clock source from HIRC clock.
[23]	**Reserved**	Reserved.
[22:20]	**TMR3SEL**	**TIMER3 Clock Source Selection** 000 = Clock source from HXT clock. 001 = Clock source from LXT clock. 010 = Clock source from PCLK. 011 = Clock source from external trigger. 101 = Clock source from LIRC clock. 111 = Clock source from HIRC clock.

		Others = reserved.
[19]	**Reserved**	Reserved.
[18:16]	**TMR2SEL**	**TIMER2 Clock Source Selection** 000 = Clock source from HXT clock. 001 = Clock source from LXT clock. 010 = Clock source from PCLK. 011 = Clock source from external trigger. 101 = Clock source from LIRC clock. 111 = Clock source from HIRC clock. Others = reserved.
[15]	**Reserved**	Reserved.
[14:12]	**TMR1SEL**	**TIMER1 Clock Source Selection** 000 = Clock source from HXT clock. 001 = Clock source from LXT clock. 010 = Clock source from PCLK. 011 = Clock source from external trigger. 101 = Clock source from LIRC clock. 111 = Clock source from HIRC clock. Others = reserved.
[11]	**Reserved**	Reserved.
[10:8]	**TMR0SEL**	**TIMER0 Clock Source Selection** 000 = Clock source from HXT clock. 001 = Clock source from LXT clock. 010 = Clock source from PCLK. 011 = Clock source from external trigger. 101 = Clock source from LIRC clock. 111 = Clock source from HIRC clock. Others = reserved.
[7]	**SPI3SEL**	**SPI3 Clock Source Selection** 0 = Clock source from PLL clock. 1 = Clock source from PCLK.
[6]	**SPI2SEL**	**SPI2 Clock Source Selection** 0 = Clock source from PLL clock. 1 = Clock source from PCLK.
[5]	**SPI1SEL**	**SPI1 Clock Source Selection** 0 = Clock source from PLL clock. 1 = Clock source from PCLK.
[4]	**SPI0SEL**	**SPI0 Clock Source Selection** 0 = Clock source from PLL clock. 1 = Clock source from PCLK.
[3:2]	**EADCSEL**	**ADC Clock Source Selection** 00 = Clock source from HXT clock. 01 = Clock source from PLL clock. 10 = Clock source from PCLK.

		11 = Clock source from HIRC clock.
[1:0]	**WDTSEL**	**Watchdog Timer Clock Source Selection (Write Protect)** These bits are protected bit,and programming this needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register REGWRPROT at address GCR_BA+0x100. 00 = Clock source from 4~24 MHz external high-speed crystal clock. 01 = Clock source from LXT clock. 10 = Clock source from HCLK/2048 clock. 11 = Clock source from LIRC clock.

Clock Source Select Control Register 2 (CLK_CLKSEL2)

Before clock switching, the related clock sources (pre-select and new-select) must be turned on.

Register	Offset	R/W	Description	Reset Value
CLK_CLKSEL2	CLK_BA+0x18	R/W	Clock Source Select Control Register 2	0x0077_7777

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved	PWM1CH45SEL			Reserved	PWM1CH23SEL		
15	14	13	12	11	10	9	8
Reserved	PWM1CH01SEL			Reserved	PWM0CH45SEL		
7	6	5	4	3	2	1	0
Reserved	PPWM0CH23SEL			Reserved	PWM0CH01SEL		

Bits	Description	
[31:23]	**Reserved**	Reserved.
[22:20]	**PWM1CH45SEL**	**PWM1_4 And PWM1_5 Clock Source Selection** PWM1_4 and PWM1_5 used the same Engine clock source; both of them use the same prescaler. The Engine clock source of PWM1_4 and PWM1_5 is defined by PWM1CH45SEL[2:0] 000 = Clock source from HXT clock. 001 = Clock source from LXT clock. 010 = Clock source from PCLK. 011 = Clock source from HIRC clock. 100 = Clock source from LIRC clock. Other = Reserved.
[19]	**Reserved**	Reserved.
[18:16]	**PWM1CH23SEL**	**PWM1_2 And PWM1_3 Clock Source Selection** PWM1_2 and PWM1_3 uses the same Engine clock source, both of them use the same prescaler. The Engine clock source of PWM1_2 and PWM1_3 is defined by PWM1CH23SEL[2:0] 000 = Clock source from HXT clock. 001 = Clock source from LXT clock. 010 = Clock source from PCLK. 011 = Clock source from HIRC clock. 100= Clock source from LIRC clock. Other = Reserved.
[15]	**Reserved**	Reserved.
[14:12]	**PWM1CH01SEL**	**PWM1_0 And PWM1_1 Clock Source Selection** PWM1_0 and PWM1_1 uses the same Engine clock source, both of them use the same

		prescaler. The Engine clock source of PWM1_0 and PWM1_1 is defined by PWM1CH01SEL[2:0] 000 = Clock source from HXT clock. 001 = Clock source from LXT clock. 010 = Clock source from PCLK. 011 = Clock source from HIRC clock. 100 = Clock source from LIRC clock. Other = Reserved.
[11]	**Reserved**	Reserved.
[10:8]	**PWM0CH45SEL**	**PWM0_4 And PWM0_5 Clock Source Selection** PWM0_4 and PWM0_5 used the same Engine clock source; both of them use the same prescaler. The Engine clock source of PWM0_4 and PWM0_5 is defined by PWM0CH45SEL[2:0] 000 = Clock source from HXT clock. 001 = Clock source from LXT clock. 010 = Clock source from PCLK. 011 = Clock source from HIRC clock. 100 = Clock source from LIRC clock. Other = Reserved.
[7]	**Reserved**	Reserved.
[6:4]	**PPWM0CH23SEL**	**PWM0_2 And PWM0_3 Clock Source Selection** PWM0_2 and PWM0_3 uses the same Engine clock source, both of them use the same prescaler. The Engine clock source of PWM0_2 and PWM0_3 is defined by PPWM0CH23SEL[2:0] 000 = Clock source from HXT clock. 001 = Clock source from LXT clock. 010 = Clock source from PCLK. 011 = Clock source from HIRC clock. 100 = Clock source from LIRC clock. Other = Reserved.
[3]	**Reserved**	Reserved.
[2:0]	**PWM0CH01SEL**	**PWM0_0 And PWM0_1 Clock Source Selection** PWM0_0 and PWM0_1 uses the same Engine clock source, both of them use the same prescaler. The Engine clock source of PWM0_0 and PWM0_1 is defined by PWM0CH01SEL[2:0] 000 = Clock source from HXT clock. 001 = Clock source from LXT clock. 010 = Clock source from PCLK. 011 = Clock source from HIRC clock. 100 = Clock source from LIRC clock. Other = Reserved.

Clock Source Select Control Register 3 (CLK_CLKSEL3)

Before clock switching, the related clock sources (pre-select and new-select) must be turned on.

Register	Offset	R/W	Description	Reset Value
CLK_CLKSEL3	CLK_BA+0x1C	R/W	Clock Source Select Control Register 3	0x000F_0FFF

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				I2S1SEL		I2S0SEL	
15	14	13	12	11	10	9	8
Reserved				SC5SEL		SC4SEL	
7	6	5	4	3	2	1	0
SC3SEL		SC2SEL		SC1SEL		SC0SEL	

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19:18]	**I2S1SEL**	**I^2S1 Clock Source Selection** 00 = Clock source from HXT clock. 01 = Clock source from PLL clock. 10 = Clock source from PCLK. 11 = Clock source from HIRC clock.
[17:16]	**I2S0SEL**	**I^2S0 Clock Source Selection** 00 = Clock source from HXT clock. 01 = Clock source from PLL clock. 10 = Clock source from PCLK. 11 = Clock source from HIRC clock.
[15:12]	**Reserved**	Reserved.
[11:10]	**SC5SEL**	**SC5 Clock Source Selection** 00 = Clock source from HXT clock. 01 = Clock source from PLL clock. 10 = PCLK. 11 = Clock source from HIRC clock.
[9:8]	**SC4SEL**	**SC4 Clock Source Selection** 00 = Clock source from HXT clock. 01 = Clock source from PLL clock. 10 = PCLK. 11 = Clock source from HIRC clock.
[7:6]	**SC3SEL**	**SC3 Clock Source Selection**

		00 = Clock source from HXT clock. 01 = Clock source from PLL clock. 10 = PCLK. 11 = Clock source from HIRC clock.
[5:4]	**SC2SEL**	**SC2 Clock Source Selection** 00 = Clock source from HXT clock. 01 = Clock source from PLL clock. 10 = PCLK. 11 = Clock source from HIRC clock.
[3:2]	**SC1SEL**	**SC1 Clock Source Selection** 00 = Clock source from HXT clock. 01 = Clock source from PLL clock. 10 = PCLK. 11 = Clock source from HIRC clock.
[1:0]	**SC0SEL**	**SC0 Clock Source Selection** 00 = Clock source from HXT clock. 01 = Clock source from PLL clock. 10 = PCLK. 11 = Clock source from HIRC clock.

Clock Divider Register 0 (CLK_CLKDIV0)

Register	Offset	R/W	Description	Reset Value
CLK_CLKDIV0	CLK_BA+0x20	R/W	Clock Divider Number Register 0	0x0000_0000

31	30	29	28	27	26	25	24
SDHDIV							
23	22	21	20	19	18	17	16
ADCDIV							
15	14	13	12	11	10	9	8
Reserved				UARTDIV			
7	6	5	4	3	2	1	0
USBHDIV				HCLKDIV			

Bits	Description	
[31:24]	**SDHDIV**	**SDHOST Clock Divide Number From SDHOST Clock Source** SDHOST clock frequency = (SDHOST clock source frequency) / (SDHDIV + 1).
[23:16]	**ADCDIV**	**ADC Clock Divide Number From ADC Clock Source** ADC clock frequency = (ADC clock source frequency) / (ADCDIV + 1).
[15:12]	**Reserved**	Reserved.
[11:8]	**UARTDIV**	**UART Clock Divide Number From UART Clock Source** UART clock frequency = (UART clock source frequency) / (UARTDIV + 1).
[7:4]	**USBHDIV**	**USB Host Clock Divide Number From PLL Clock** USB Host clock frequency = (PLL frequency) / (USBHDIV + 1).
[3:0]	**HCLKDIV**	**HCLK Clock Divide Number From HCLK Clock Source** HCLK clock frequency = (HCLK clock source frequency) / (HCLKDIV + 1).

Clock Divider Register 1 (CLK_CLKDIV1)

Register	Offset	R/W	Description	Reset Value
CLK_CLKDIV1	CLK_BA+0x24	R/W	Clock Divider Number Register 1	0x0000_0000

31	30	29	28	27	26	25	24
SC3DIV							
23	22	21	20	19	18	17	16
SC2DIV							
15	14	13	12	11	10	9	8
SC1DIV							
7	6	5	4	3	2	1	0
SC0DIV							

Bits	Description	
[31:24]	**SC3DIV**	**SC3 Clock Divide Number From SC3 Clock Source** SC3 clock frequency = (SC3 clock source frequency) / (SC3DIV+ 1).
[23:16]	**SC2DIV**	**SC2 Clock Divide Number From SC2 Clock Source** SC2 clock frequency = (SC2 clock source frequency) / (SC2DIV + 1).
[15:8]	**SC1DIV**	**SC1 Clock Divide Number From SC1 Clock Source** SC1 clock frequency = (SC1 clock source frequency) / (SC1DIV + 1).
[7:0]	**SC0DIV**	**SC0 Clock Divide Number From SC0 Clock Source** SC0 clock frequency = (SC0 clock source frequency) / (SC0DIV+ 1).

Clock Divider Register 2 (CLK_CLKDIV2)

Register	Offset	R/W	Description	Reset Value
CLK_CLKDIV2	CLK_BA+0x28	R/W	Clock Divider Number Register 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
SC5DIV							
7	6	5	4	3	2	1	0
SC4DIV							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:8]	**SC5DIV**	**SC5 Clock Divide Number From SC5 Clock Source** SC5 clock frequency = (SC5 clock source frequency) / (SC5DIV + 1).
[7:0]	**SC4DIV**	**SC4 Clock Divide Number From SC4 Clock Source** SC4 clock frequency = (SC4 clock source frequency) / (SC4DIV + 1).

Clock Divider Register 3 (CLK_CLKDIV3)

Register	Offset	R/W	Description	Reset Value
CLK_CLKDIV3	CLK_BA+0x2C	R/W	Clock Divider Number Register 3	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
EMACDIV							
15	14	13	12	11	10	9	8
VSENSEDIV							
7	6	5	4	3	2	1	0
CAPDIV							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**EMACDIV**	**Ethernet Clock Divide Number Form HCLK (NUC472 Only)** EMAC MDCLK clock frequency = (HCLK) / (EMACDIV + 1).
[15:8]	**VSENSEDIV**	**Video Pixel Clock Divide Number From ICAP Clock Source** Video pixel clock frequency = (ICAP clock source frequency) / (VSENSEDIV + 1).
[7:0]	**CAPDIV**	**Image Capture Seneor Clock Divide Number From ICAP Clock Source** Image sensor clock frequency = (ICAP clock source frequency) / (ICAPDIV + 1).

PLL Control Register (CLK_PLLCTL)

The PLL reference clock input is from the HXT clock input or from the HIRC. These registers are used to control the PLL output frequency and PLL operation mode.

Programming these bits needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register REGWRPROT at address GCR_BA+0x100.

Register	Offset	R/W	Description	Reset Value
CLK_PLLCTL	CLK_BA+0x40	R/W	PLL Control Register	0x0005_8228

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved			PLLREMAP	PLLSRC	OE	BP	PD
15	14	13	12	11	10	9	8
OUTDV		INDIV					FBDIV
7	6	5	4	3	2	1	0
FBDIV							

Bits	Description	
[31:21]	**Reserved**	Reserved.
[20]	**PLLREMAP**	**PLL Remap Enable Bit** 0 = PLL remap enable. 1 = PLL remap disable.
[19]	**PLLSRC**	**PLL Source Clock Selection** 0 = PLL source clock from HXT. 1 = PLL source clock from HIRC.
[18]	**OE**	**PLL OE (FOUT Enable) Pin Control** 0 = PLL FOUT Enabled. 1 = PLL FOUT is fixed low.
[17]	**BP**	**PLL Bypass Control** 0 = PLL is in normal mode (default). 1 = PLL clock output is same as clock input (XTALin).
[16]	**PD**	**Power-Down Mode** If set the PWR_DOWN_EN bit to 1 in CLK_PWRCTL register, the PLL will enter Power-down mode, too. 0 = PLL is in normal mode. 1 = PLL is in Power-down mode (default).
[15:14]	**OUTDV**	**PLL Output Divider Control Pins** Refer to the formulas below the table.

[13:9]	**INDIV**	**PLL Input Divider Control Pins** Refer to the formulas below the table.
[8:0]	**FBDIV**	**PLL Feedback Divider Control Pins** Refer to the formulas below the table.

Output Clock Frequency Setting is:

$$\mathrm{FOU} = \mathrm{FIN} \times \frac{NF}{NR} \times \frac{1}{NO}$$

And with following constrains:

1. $3.2\ \mathrm{MHz} < \mathrm{FIN} < 150\ \mathrm{MHz}$
2. $800\ \mathrm{kHz} < \frac{\mathrm{FIN}}{2 \times \mathrm{NR}} < 8\ \mathrm{MHz}$
3. $100\ \mathrm{MHz} < \mathrm{FCO} = \mathrm{FIN} \times \frac{\mathrm{NF}}{\mathrm{NR}} < 200\ \mathrm{MHz}$, where $120\ \mathrm{MHz} < \mathrm{FCO}$ is preferred.

Symbol	Description
FOUT	Output Clock Frequency
FIN	Input (Reference) Clock Frequency
NR	Input Divider (INDIV + 2)
NF	Feedback Divider (FBDIV + 2)
NO	OUTDV = “00” : NO = 1 OUTDV = “01” : NO = 2 OUTDV = “10” : NO = 2 OUTDV = “11” : NO = 4

Default Frequency Setting

The default value: 0x8228
FIN = 12 MHz
NR = (1+2) = 3
NF = (40+2) = 42
NO = 2
FOUT = 12/2 x 42 x 1/3 = 84 MHz

The value: 0xC22E
FIN = 12 MHz
NR = (1+2) = 3
NF = (46+2) = 48
NO = 4
FOUT = 12/4 x 48 x 1/3 = 48 MHz

PLL2 Control Register (CLK_PLL2CTL)

The PLL2 reference clock input is from the HXT clock input.

The PLL2 is from USB_PHY and enables fixed 20X. The registers are used to control the PLL output frequency and PLL operation mode.

Programming these bits needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register REGWRPROT at address GCR_BA+0x100.

Register	Offset	R/W	Description	Reset Value
CLK_PLL2CTL	CLK_BA+0x44	R/W	PLL2 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
	Reserved						PLL2CKEN
7	6	5	4	3	2	1	0
PLL2DIV							

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**PLL2CKEN**	**USB OHY 480 MHz Enable Bit** This bit enables USB PHY PLL (480 MHz), and user needs to care extend 12 MHz source. 0 = USB PHY PLL (480 MHz) Disabled. 1 = USB PHY PLL (480 MHz) Enabled.
[7:0]	**PLL2DIV**	**PLL2 Divider Control** PLL2 clock frequency = (480 MHz) / 2 / (PLL2DIV + 1). **Note:** Max. PLL frequency 240 MHz when XTL12M.

Clock Status Register (CLK_STATUS)

These bits of this register are used to monitor if the chip clock source is stable or not, and whether the clock switch is failed.

Register	Offset	R/W	Description	Reset Value
CLK_STATUS	CLK_BA+0x50	R/W	Clock Status Monitor Register	0x0000_00XX

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
CLKSFAIL	Reserved	PLL2STB	HIRCSTB	LIRCSTB	PLLSTB	LXTSTB	HXTSTB

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**CLKSFAIL**	**Clock Switching Fail Flag** 0 = Clock switching success. 1 = Clock switching failure. **Note1:** This bit is updated when software switches system clock source. If switch target clock is stable, this bit will be set to 0. If switch target clock is not stable, this bit will be set to 1. **Note2:** Write 1 to clear the bit to 0.
[6]	**Reserved**	Reserved.
[5]	**PLL2STB**	**Internal PLL2 Clock Source Stable Flag** 0 = Internal PLL2 clock is not stable or disabled. 1 = Internal PLL2 clock is stable. **Note:** This bit is read only.
[4]	**HIRCSTB**	**22.1184 MHz Internal High-Speed Oscillator Clock (HIRC) Clock Source Stable Flag** 0 = HIRC clock is not stable or disabled. 1 = HIRC clock is stable. **Note:** This bit is read only.
[3]	**LIRCSTB**	**10 KHz Internal Low-Speed Oscillator Clock (LIRC)Source Stable Flag** 0 = LIRC clock is not stable or disabled. 1 = LIRC clock is stable. **Note:** This bit is read only.
[2]	**PLLSTB**	**Internal PLL Clock Source Stable Flag** 0 = Internal PLL clock is not stable or disabled.

		1 = Internal PLL clock is stable. **Note:** This bit is read only.
[1]	**LXTSTB**	**32.768 KHz External Low-Speed Crystal Clock(LXT) Source Stable Flag** 0 = LXT clock is not stable or disabled. 1 = LXT clock is stabled. **Note:** This is read only.
[0]	**HXTSTB**	**4~24 MHz External High-Speed Crystal Clock(HXT) Source Stable Flag** 0 = HXT clock is not stable or disabled. 1 = HXT clock is stable. **Note:** This bit is read only.

Frequency Divider Control Register (CLK_CLKOCTL)

Register	Offset	R/W	Description	Reset Value
CLK_CLKOCTL	CLK_BA+0x60	R/W	Frequency Divider Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		DIV1EN	CLKOEN	FSEL			

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**DIV1EN**	**Frequency Divider 1 Enable Bit** 0 = Divider output frequency is dependent on FSEL value when FDIVEN is enabled. 1 = Divider output frequency is input clock frequency.
[4]	**CLKOEN**	**Clock Output Enable Bit** 0 = Clock output disabled. 1 = Clock output enabled.
[3:0]	**FSEL**	**Divider Output Frequency Selection** The formula of output frequency is: $F_{out} = F_{in}/2^{(N+1)}$. F_{in} is the input clock frequency. F_{out} is the frequency of divider output clock. N is the 4-bit value of FSEL[3:0].

Clock Stop Detector Control Register (CLK_CLKDCTL)

Register	Offset	R/W	Description	Reset Value
CLK_CLKDCTL	CLK_BA+0x70	R/W	Clock Fail Detector Control Register	0x0000_0100

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					IRCFIF	IRCFIEN	IRCDEN
7	6	5	4	3	2	1	0
Reserved					SYSFIF	SYSFIEN	SYSFDEN

Bits	Description	
[31:11]	**Reserved**	Reserved.
[10]	**IRCFIF**	**Internal RC Clock Fail Flag** 0 = IRC clock normal. 1 = IRC abnormal (write “1” to clear) .
[9]	**IRCFIEN**	**Internal RC Clock Detector Interrupt Enable Bit** 0 = IRC clock fail interrupt disabled. 1 = IRC clock fail interrupt enabled.
[8]	**IRCDEN**	**Internal RC Clock Detector Enable Bit** 0 = IRC clock fail interrupt disabled. 1 = IRC clock fail interrupt enabled.
[7:3]	**Reserved**	Reserved.
[2]	**SYSFIF**	**System Clock Detect Fail Flag** 0 = System clock normal. 1 = System clock abnormal (write “ 1” to clear).
[1]	**SYSFIEN**	**System Clock Detector Interrupt Enable Bit** 0 = system clock fail interrupt disabled. 1 = system clock fail interrupt enabled.
[0]	**SYSFDEN**	**System Clock Detector Enable Bit** 0 = system clock fail interrupt disabled. 1 = system clock fail interrupt enabled.

6.4 Analog Comparator Controller (ACMP)

6.4.1 Overview

The NUC442/NUC472 contains three comparators which can be used in a number of different configurations. The comparator output is a logical one when positive input is greater than negative input; otherwise, the output is zero. Each comparator can be configured to cause an interrupt when the comparator output value changes. The block diagram is shown below.

6.4.2 Features

- Analog input voltage range: 0 ~ AV_{DD}
- Supports hysteresis function
- Selectable input sources of positive input and negative input
- One ACMP interrupt vector for all comparators

6.4.3 Block Diagram

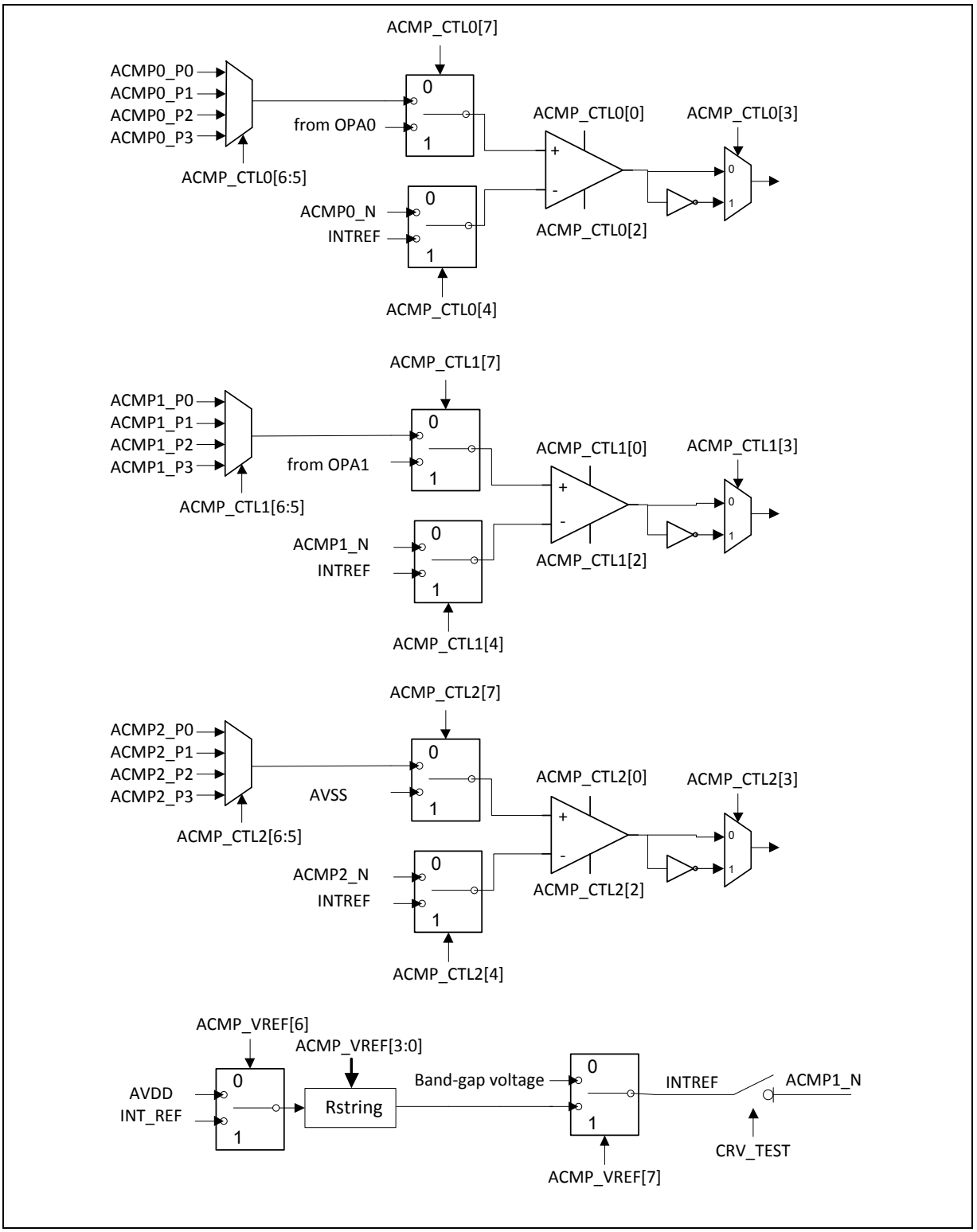


Figure 6.4-1 Analog Comparator Block Diagram

6.4.4 Functional Description

6.4.4.1 Interrupt Sources

The output of comparators are sampled by PCLK and reflected at ACMPO0 (ACMP_STATUS[3]), ACMPO1(ACMP_STATUS [4]) and ACMPO2(ACMP_STATUS [5]). If ACMPIE (ACMP_CTLx[1]) is set to 1, the comparator interrupt will be enabled. As the output state of comparator is changed, the comparator interrupt will be asserted and the corresponding flag, ACMPIF0(ACMP_STATUS[0]), ACMPIF1(ACMP_STATUS[1]) or ACMPIF2(ACMP_STATUS[2]), will be set. Software can clear the flag to 0 by writing 1 to it.

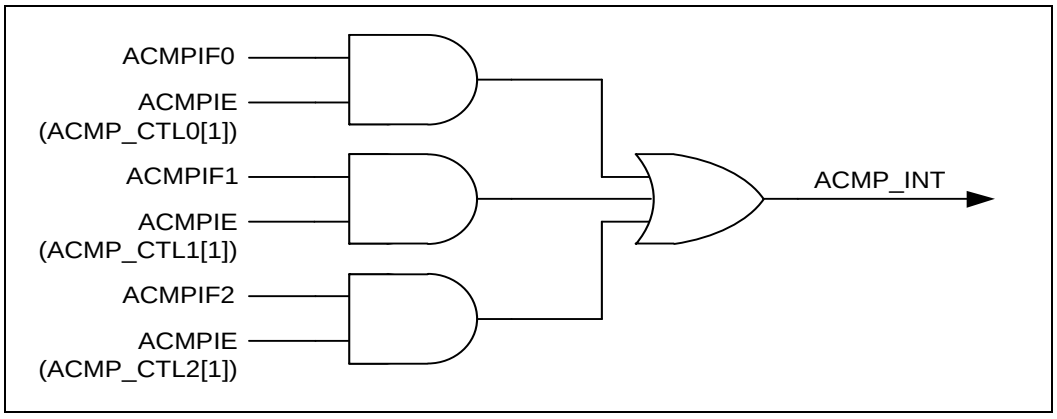


Figure 6.4-2 Comparator Controller Interrupt Sources

6.4.4.2 Hysteresis Function

The analog comparator provides the hysteresis function to make the comparator output transition more stable. If comparator output is 0, it will not be changed to 1 until the positive input voltage exceeds the negative input voltage by a high threshold voltage. Similarly, if comparator output is 1, it will not be changed to 0 until the positive input voltage drops below the negative input voltage by a low threshold voltage.

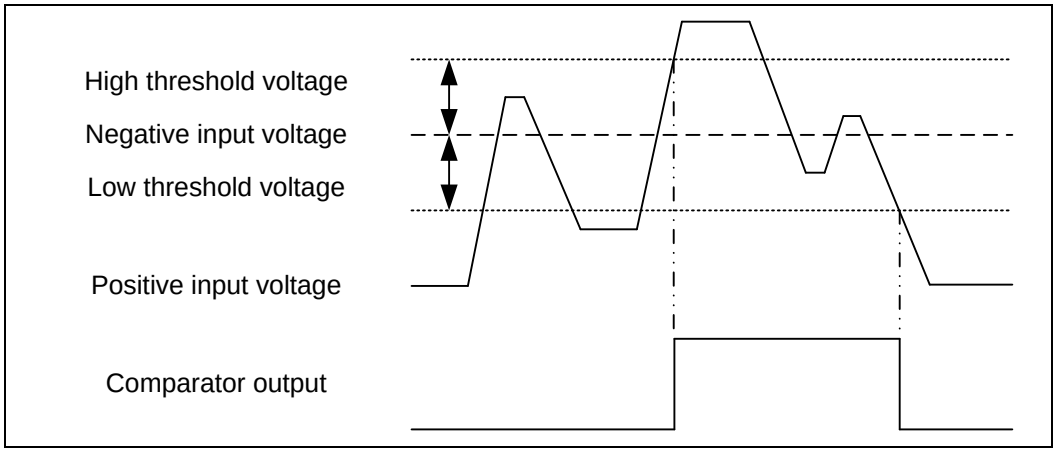


Figure 6.4-3 Comparator Hysteresis Function

6.4.5 Comparator Reference Voltage (CRV)

6.4.5.1 Introduction

The comparator reference voltage (CRV) module is responsible for generating reference voltage for comparators. The CRV module consists of resisters ladder and analog switch. User can set the CRV output voltage by setting the CRVCTL[3:0] (ACMP_VREF[3:0]) and select the reference voltage to comparator by setting NEGSEL (ACMP_CTLx[4]) and IREFSEL (ACMP_VREF[7]).

6.4.5.2 Features:

- User selectable references voltage by setting CRVCTL[3:0] registers
- Automatic disable resisters ladder for reducing power consumption when setting IREFSEL =0 (selecting band-gap voltage output)

The block diagram of the CRV module is shown below:

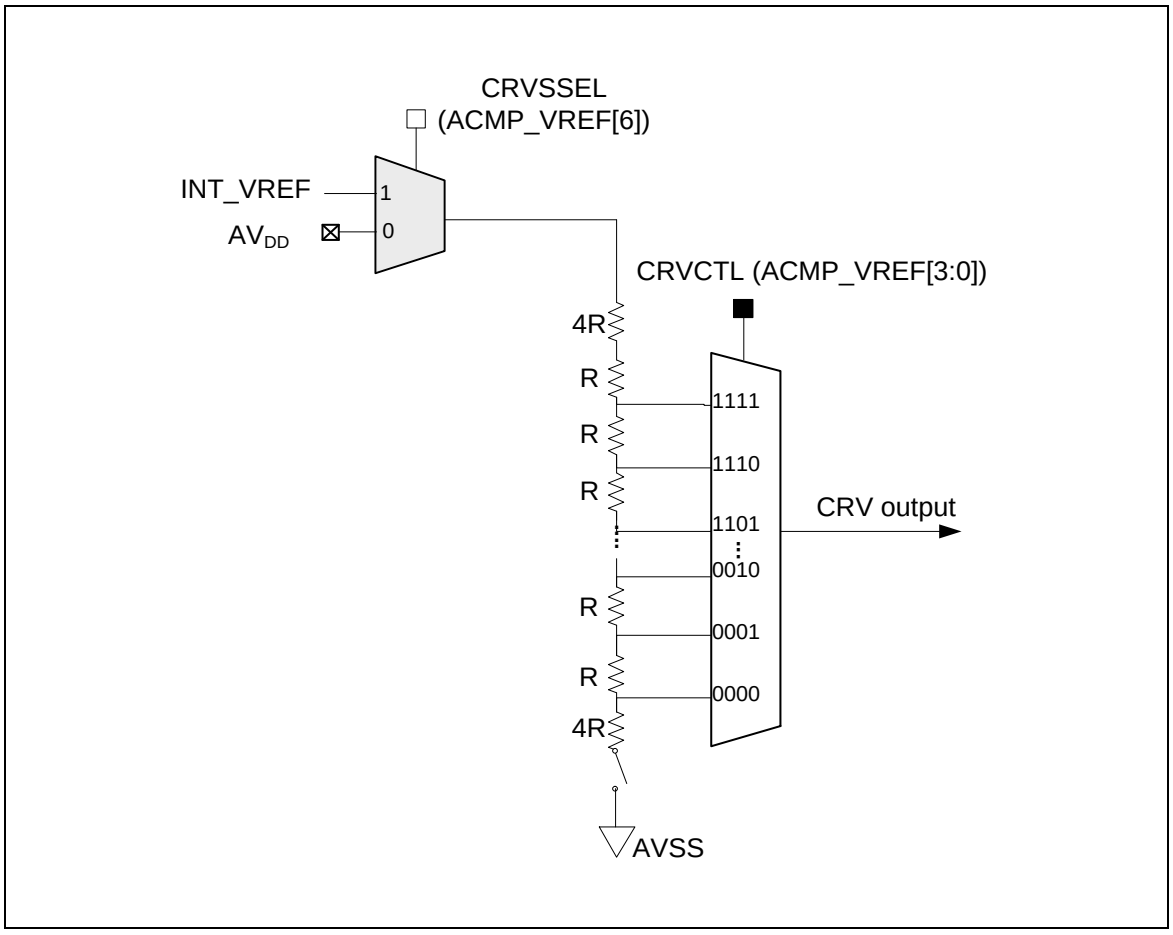


Figure 6.4-4 Comparator Reference Voltage Block Diagram

6.4.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
ACMP Base Address: **ACMP_BA = 0x4004_5000**				
ACMP_CTL0	ACMP_BA+0x00	R/W	Analog Comparator 0 Control Register	0x0000_0000
ACMP_CTL1	ACMP_BA+0x04	R/W	Analog Comparator 1 Control Register	0x0000_0000
ACMP_CTL2	ACMP_BA+0x08	R/W	Analog Comparator 2 Control Register	0x0000_0000
ACMP_STATUS	ACMP_BA+0x0C	R/W	Analog Comparator Status Register	0x0000_0000
ACMP_VREF	ACMP_BA+0x10	R/W	Analog Comparator Reference Voltage Control Register	0x0000_0000

6.4.7 Register Description

Analog Comparator 0 Control Register (ACMP_CR0)

Register	Offset	R/W	Description	Reset Value
ACMP_CTL0	ACMP_BA+0x00	R/W	Analog Comparator 0 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
POSSEL			**NEGSEL**	**ACMPOINV**	**HYSEN**	**ACMPIE**	**ACMPEN**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:5]	**POSSEL**	**Comparator 0 Positive Input Selection** 000= Input from ACMP0_P0. 001= Input from ACMP0_P1. 010= Input from ACMP0_P2. 011= Input from ACMP0_P3. 100= Input from OPA0. The other options are reserved.
[4]	**NEGSEL**	**Comparator 0 Negative Input Selection** 0 = The source of comparator 0 negative input is from ACMP0_N pin. 1 = The internal comparator reference voltage (Band-gap voltage or CRV) is selected as the source of comparator 0 negative input.
[3]	**ACMPOINV**	**Comparator 0 Output Inverse** 0 = Comparator 0 output inverse Disabled. 1 = Comparator 0 output inverse Enabled.
[2]	**HYSEN**	**Comparator 0 Hysteresis Enable Bit** 0 = Comparator 0 hysteresis Disabled (Default). 1 = Comparator 0 hysteresis Enabled (typical range is 20 mV).
[1]	**ACMPIE**	**Comparator 0 Interrupt Enable Bit** 0 = Comparator 0 interrupt Disabled. 1 = Comparator 0 interrupt Enabled.
[0]	**ACMPEN**	**Comparator 0 Enable Bit** 0 = Comparator 0 Disabled. 1 = Comparator 0 Enabled.

		Note: The comparator output needs to wait 2 us stable time after ACMPEN is set.

Analog Comparator 1 Control Register (ACMP_CR1)

Register	Offset	R/W	Description	Reset Value
ACMP_CTL1	ACMP_BA+0x04	R/W	Analog Comparator 1 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
POSSEL			**NEGSEL**	**ACMPOINV**	**HYSEN**	**ACMPIE**	**ACMPEN**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:5]	**POSSEL**	**Comparator 1 Positive Input Selection** 000= Input from ACMP1_P0. 001= Input from ACMP1_P1. 010= Input from ACMP1_P2. 011= Input from ACMP1_P3. 100= Input from OPA1. The other options are reserved.
[4]	**NEGSEL**	**Comparator 1 Negative Input Selection** 0 = The source of comparator 1 negative input is from ACMP1_N pin. 1 = The internal comparator reference voltage (Band-gap voltage or CRV) is selected as the source of comparator 1 negative input.
[3]	**ACMPOINV**	**Comparator 1 Output Inverse Control** 0 = Comparator 1 output inverse Disabled. 1 = Comparator 1 output inverse Enabled.
[2]	**HYSEN**	**Comparator 1 Hysteresis Enable Bit** 0 = Comparator 1 hysteresis Disabled (Default). 1 = Comparator 1 hysteresis Enabled (typical range is 20 mV).
[1]	**ACMPIE**	**Comparator 1 Interrupt Enable Bit** 0 = Comparator 1 interrupt Disabled. 1 = Comparator 1 interrupt Enabled.
[0]	**ACMPEN**	**Comparator 1 Enable Bit** 0 = Comparator 1 Disabled. 1 = Comparator 1 Enabled.

Bits	Description	
		The comparator output needs to wait 2 us stable time after ACMPEN is set.

Analog Comparator 2 Control Register (ACMP_CR2)

Register	Offset	R/W	Description	Reset Value
ACMP_CTL2	ACMP_BA+0x08	R/W	Analog Comparator 2 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
POSSEL			NEGSEL	ACMPOINV	HYSEN	ACMPIE	ACMPEN

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:5]	**POSSEL**	**Comparator 2 Positive Input Selection** 000= Input from ACMP2_P0. 001= Input from ACMP2_P1. 010= Input from ACMP2_P2. 011= Input from ACMP2_P3. The other options are reserved.
[4]	**NEGSEL**	**Comparator 2 Negative Input Selection** 0 = The source of comparator 2 negative input is from ACMP2_N pin. 1 = The internal comparator reference voltage (Band-gap voltage or CRV) is selected as the source of comparator 2 negative input.
[3]	**ACMPOINV**	**Comparator 2 Output Inverse Control** 0 = Comparator 2 output inverse Disabled. 1 = Comparator 2 output inverse Enabled.
[2]	**HYSEN**	**Comparator 2 Hysteresis Enable Bit** 0 = Comparator 2 hysteresis Disabled (Default). 1 = Comparator 2 hysteresis Enabled (typical range is 20 mV).
[1]	**ACMPIE**	**Comparator 2 Interrupt Enable Bit** 0 = Comparator 2 interrupt Disabled. 1 = Comparator 2 interrupt Enabled.
[0]	**ACMPEN**	**Comparator 2 Enable Bit** 0 = Comparator 2 Disabled. 1 = Comparator 2 Enabled. The comparator output needs to wait 2 us stable time after ACMPEN is set.

Analog Comparator Status Register (ACMP_STATUS)

Register	Offset	R/W	Description	Reset Value
ACMP_STATUS	ACMP_BA+0x0C	R/W	Analog Comparator Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		ACMPO2	ACMPO1	ACMPO0	ACMPIF2	ACMPIF1	ACMPIF0

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**ACMPO2**	**Comparator 2 Output** Synchronized to the APB clock to allow reading by software. Cleared when the comparator 2 is disabled (ACMP_CTL2[0] = 0).
[4]	**ACMPO1**	**Comparator 1 Output** Synchronized to the APB clock to allow reading by software. Cleared when the comparator 1 is disabled (ACMP_CTL1[0] = 0).
[3]	**ACMPO0**	**Comparator 0 Output** Synchronized to the APB clock to allow reading by software. Cleared when the comparator 0 is disabled (ACMP_CTL0[0] = 0).
[2]	**ACMPIF2**	**Comparator 2 Flag** This bit is set by hardware whenever the comparator 2 output changes state. This will cause an interrupt if ACMP_CTL2[1] is set to 1. Write 1 to clear this bit to 0.
[1]	**ACMPIF1**	**Comparator 1 Flag** This bit is set by hardware whenever the comparator 1 output changes state. This will cause an interrupt if ACMP_CTL1[1] is set to 1. Write 1 to clear this bit to 0.
[0]	**ACMPIF0**	**Comparator 0 Flag** This bit is set by hardware whenever the comparator 0 output changes state. This will cause an interrupt if ACMP_CTL0[1] is set to 1. Write 1 to clear this bit to 0.

ACMP Reference Voltage Control Register (ACMP_VREF)

Register	Offset	R/W	Description	Reset Value
ACMP_VREF	ACMP_BA+0x10	R/W	Analog Comparator Reference Voltage Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IREFSEL	**CRVSSEL**	**Reserved**		**CRVCTL**			

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**IREFSEL**	**Internal Reference Selection** 0 = Band-gap voltage is selected as internal reference. 1 = CRV is selected as internal reference.
[6]	**CRVSSEL**	**CRV Source Voltage Selection** 0 = VDDA is selected as CRV source voltage. 1 = Internal reference voltage is selected as CRV source voltage.
[5:4]	**Reserved**	Reserved.
[3:0]	**CRVCTL [3:0]**	**Comparator Reference Voltage Setting** CRV = CRV source voltage * (1/6+VREF[3:0]/24).

6.5 Analog-to-Digital Converter (ADC)

6.5.1 Overview

The NUC442/NUC472 contains one 12-bit successive approximation analog-to-digital converters (SAR A/D converter) with 12 external input channels. The A/D converter supports three operation modes —Single Mode, Single-cycle Scan Mode and Continuous Scan Mode. The A/D converters can be started by software, external pin (STADC) or PWM trigger.

6.5.2 Features

- Analog input voltage range: 0~Analog Supply Voltage from AV_{DD}
- 12-bit resolution and 10-bit accuracy is guaranteed
- Up to 12 single-end analog input channels or 6 differential analog input channels.
- Supports conversion rate 400 kSPS while V_{REF} is between 2.5V~5.5V and up to 800 kSPS while V_{REF} is between 4.5V~5.5V in single-end mode.
- Supports conversion rate 800 kSPS while V_{REF} is between 2.5V~5.5V and up to 1 MSPS while V_{REF} is between 3.0V~5.5V in differential mode.
- Three operation modes
  - Single Mode: A/D conversion is performed one time on a specified channel
  - Single-cycle Scan Mode: A/D conversion is performed one cycle on all specified channels with the sequence from the smallest numbered channel to the largest numbered channel
  - Continuous Scan Mode: A/D converter continuously performs Single-cycle scan mode until software stops A/D conversion
- An A/D conversion can be started by
  - Software write 1 to A/D conversion start bit (ADST)
  - External pin (STADC)
  - PWM trigger with optional start delay period
- Conversion results are held in data registers for each channel with valid and overrun indicators
- Conversion result can be compared with specified value and user can select whether to generate an interrupt when conversion result is larger than, smaller than or equal to the compare register setting
- Supports internal source: internal band-gap voltage, and internal temperature sensor output voltage
- Supports PDMA transfer

6.5.3 Block Diagram

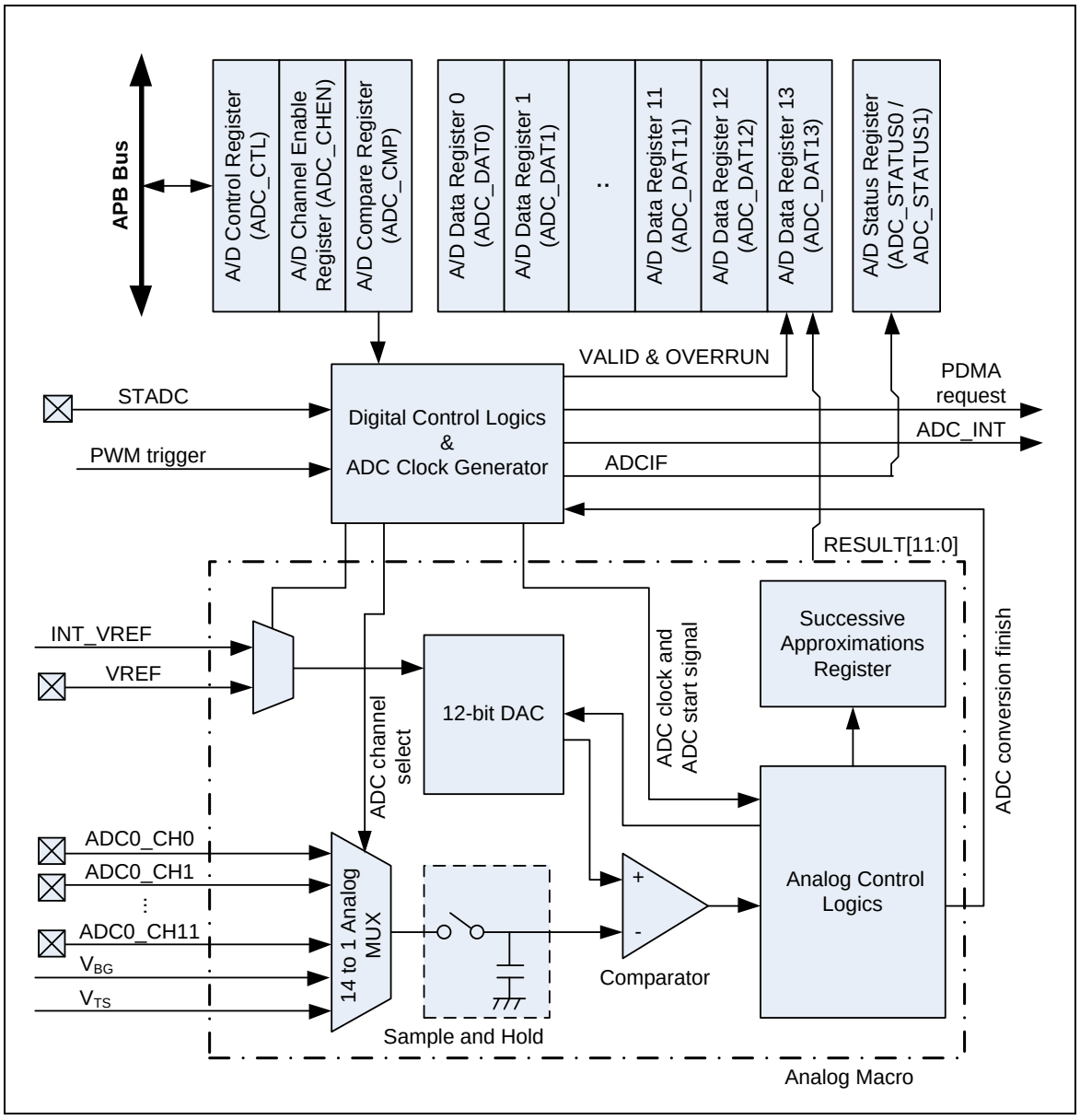


Figure 6.5-1 ADC Controller Block Diagram

6.5.4 Functional Description

The A/D converter operates by successive approximation with 12-bit resolution. The ADC has three operation modes: Single Mode, Single-cycle Scan Mode and Continuous Scan Mode. When changing the operation mode or analog input channel, to prevent incorrect operation, software must clear SWTRG (ADC_CTL[11]) register.

6.5.4.1 ADC Clock Generator

The maximum sampling rate is up to 800 kSPS. The ADC engine has four clock sources selected by 2-bit EADCSEL (CLK_CLKDIV1[3:2]), the ADC clock frequency is divided by an 8-bit prescaler with the formula:

The ADC clock frequency = (ADC clock source frequency) / (ADCDIV+1);

Where the 8-bit ADCDIV is located in register CLK_CLKDIV0[23:16].

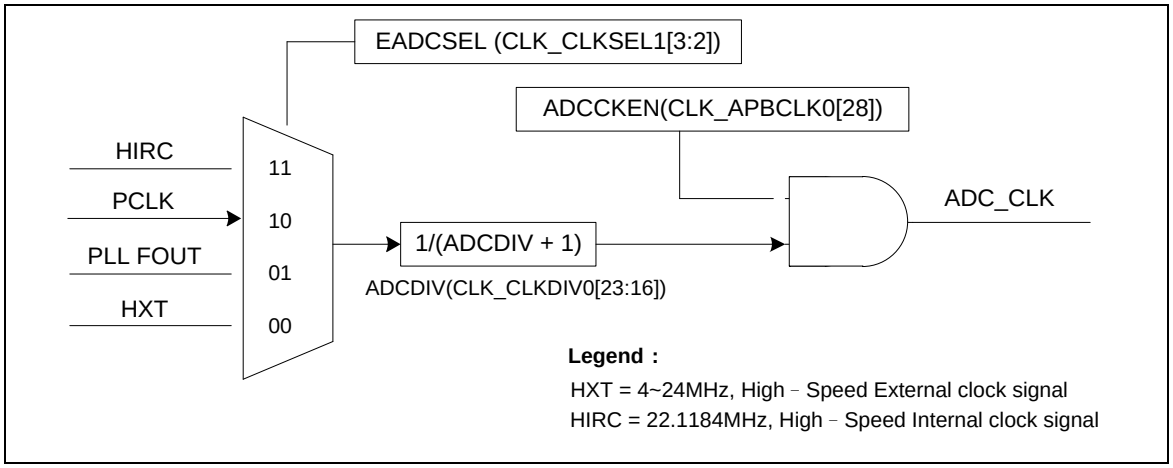


Figure 6.5-2 ADC Clock Control

6.5.4.2 Single Mode

In Single Mode, A/D conversion is performed only once on the specified single channel. The operations are as follows:

1. A/D conversion will be started when the SWTRG (ADC_CTL[11]) is set to 1 by software or external hardware trigger input.
2. When A/D conversion is finished, the result is stored in the A/D data register corresponding to the channel.
3. The ADIF (ADC_STATUS0[0]) bit will be set to 1. If the ADCIEN (ADC_CTL[1]) bit is set to 1, the ADC interrupt will be asserted.
4. The SWTRG (ADC_CTL[11]) bit remains 1 during A/D conversion. When A/D conversion ends, the SWTRG (ADC_CTL[11]) bit is automatically cleared to 0 and the A/D converter enters idle state.

Note: If software enables more than one channel in Single mode, the channel with the smallest number will be selected and the other enabled channels will be ignored.

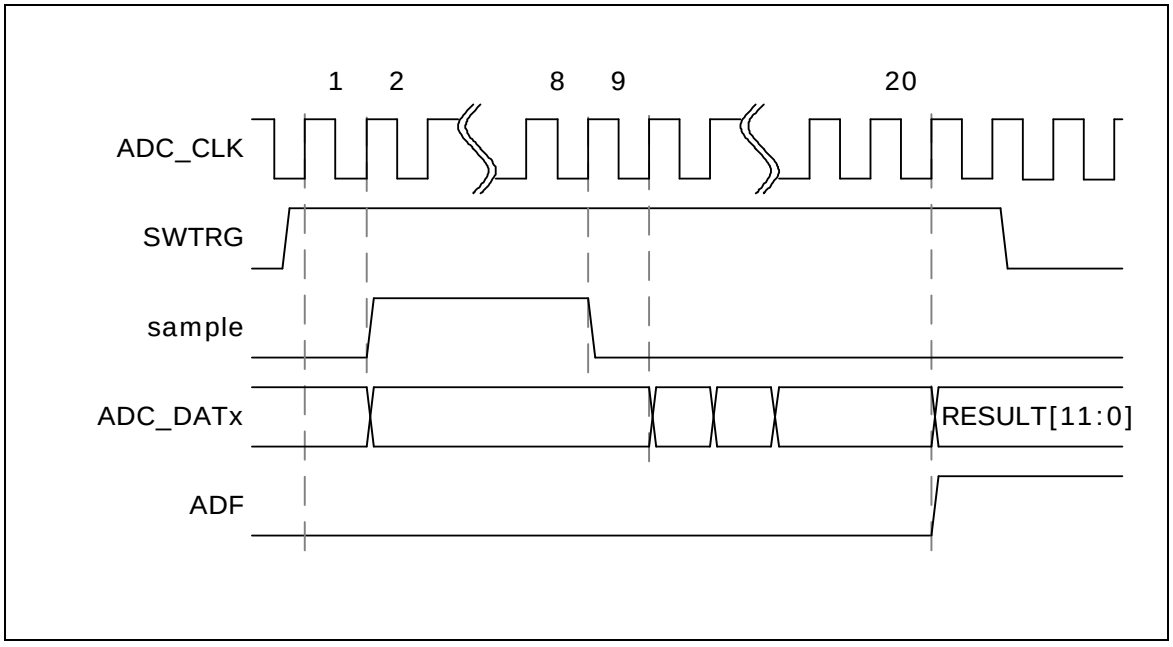


Figure 6.5-3 Single Mode Conversion Timing Diagram

6.5.4.3 Single-Cycle Scan Mode

In Single-cycle Scan Mode, A/D conversion will sample and convert the specified channels once in the sequence from the smallest number enabled channel to the largest number enabled channel.

1. When the SWTRG (ADC_CTL[11]) bit is set to 1 by software or external hardware trigger input, A/D conversion starts on the channel with the smallest number.
2. When A/D conversion for each enabled channel is completed, the result is sequentially transferred to the A/D data register corresponding to each channel.
3. When the conversions of all the enabled channels are completed, the ADIF (ADC_STATUS0[0]) bit is set to 1. If the ADC interrupt function is enabled, the ADC interrupt occurs.
4. After A/D conversion ends, the SWTRG (ADC_CTL[11]) bit is automatically cleared to 0 and the A/D converter enters idle state. If SWTRG (ADC_CTL[11]) bit is cleared to 0 before all enabled ADC channels conversion done, ADC controller will finish current conversion and load conversion result to the current A/D data register.

An example timing diagram for single-cycle scan on enabled channels (0, 2, 3 and 7) is shown in the following figure.

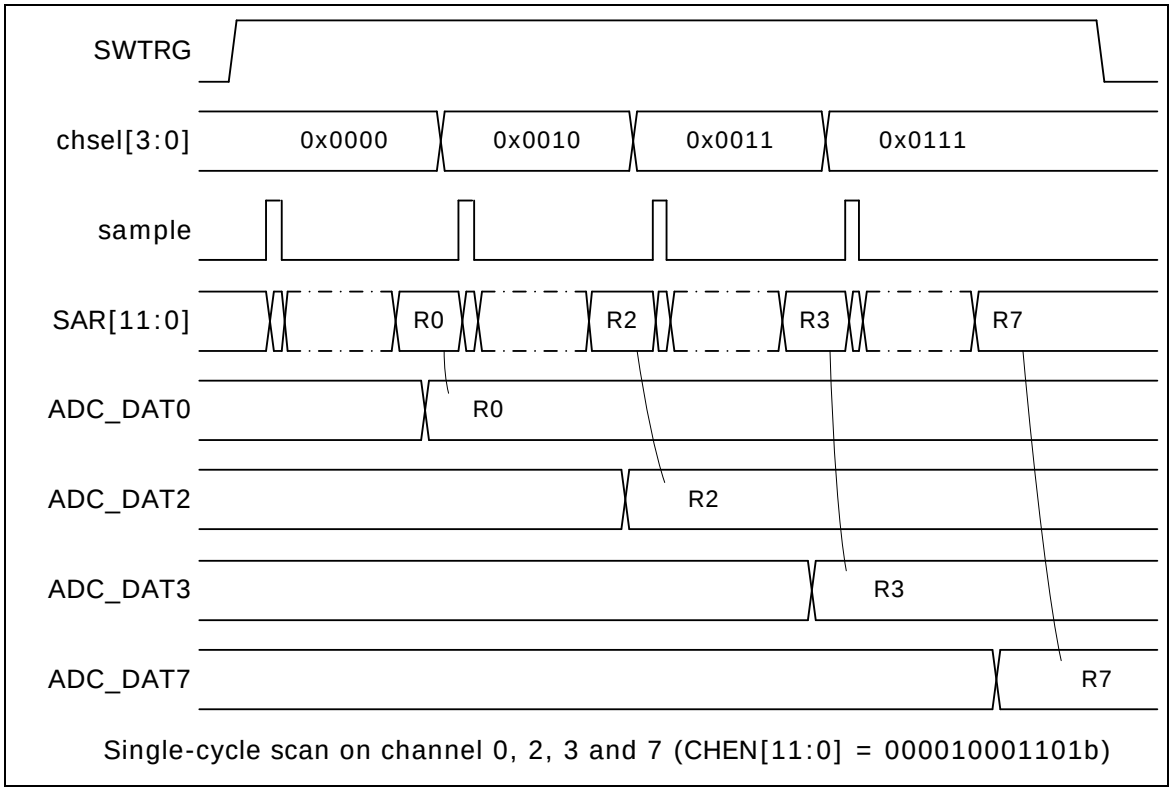


Figure 6.5-4 Single-Cycle Scan on Enabled Channels Timing Diagram

6.5.4.4 Continuous Scan Mode

In continuous scan mode, A/D conversion is performed sequentially on the specified channels that enabled by CHEN (ADCHER[11:0]) bits. The operations are as follows:

1. When the SWTRG (ADC_CTL[11]) bit is set to 1 by software or external trigger input, A/D conversion starts on the channel with the smallest number.
2. When A/D conversion for each enabled channel is completed, the result of each enabled channel is stored in the A/D data register corresponding to each enabled channel.
3. When A/D converter completes the conversions of all enabled channels sequentially, the ADIF (ADC_STATUS0[0]) bit will be set to 1. If the ADC interrupt function is enabled, the ADC interrupt occurs. The conversion of the enabled channel with the smallest number will start again if software has not cleared the SWTRG (ADC_CTL[11]) bit.
4. As long as the SWTRG (ADC_CTL[11]) bit remains at 1, the step 2 ~ 3 will be repeated. When SWTRG (ADC_CTL[11]) bit is cleared to 0, ADC controller will finish current conversion and load conversion result to the current A/D data register.

An example timing diagram for continuous scan on enabled channels (0, 2, 3 and 7) is shown below:

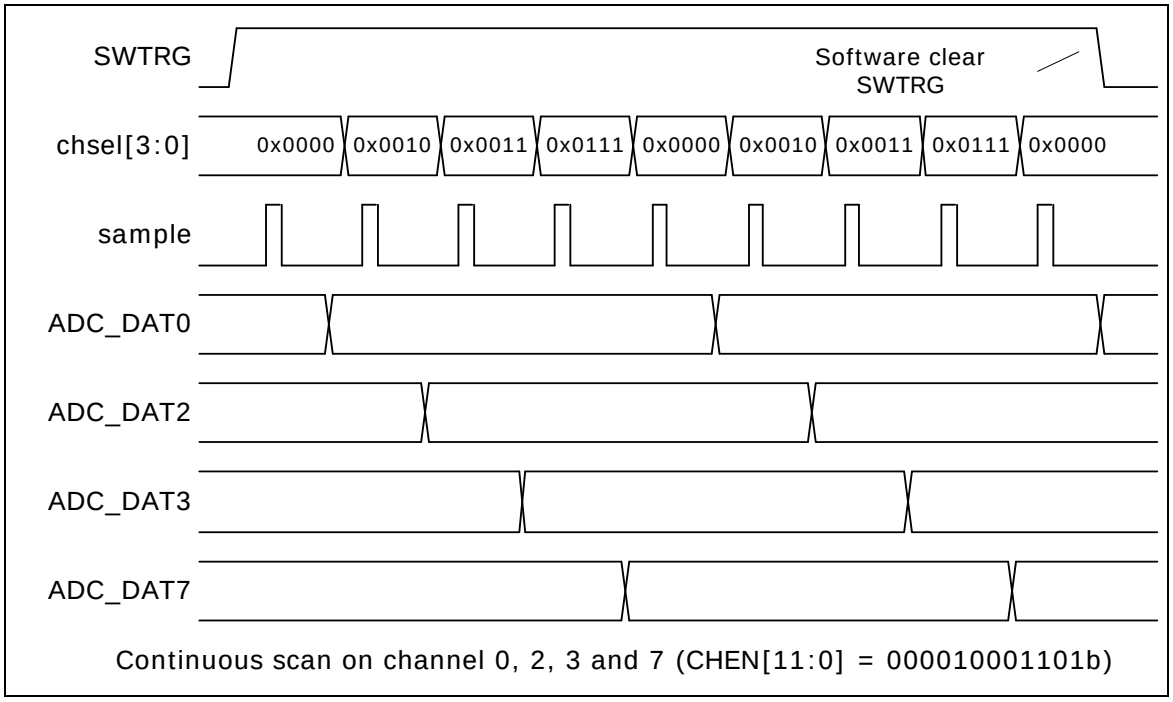


Figure 6.5-5 Continuous Scan on Enabled Channels Timing Diagram

6.5.4.5 External hardware trigger

ADC has two external hardware trigger sources: external pin (STADC) trigger and PWM trigger.

6.5.4.6 External pin (STADC) trigger

In Single-cycle Scan Mode, A/D conversion can be triggered by external pin (STADC). When the HWTRGCOND (ADC_CTL[8]) bit is set to high to enable ADC external hardware trigger function, setting the HWTRGSEL (ADC_CTL[5:4]) bits to 00b is to select external hardware trigger input source from the external pin (STADC). Software can set HWTRGCOND (ADC_CTL[7:6]) bits to select external pin (STADC) trigger condition is falling/rising edge or low/high level. If level trigger condition is selected, the external pin (STADC) must be kept at defined state at least 8 system clocks. The SWTRG (ADC_CTL[11]) bit will be set to 1 at the 9th system clock and start to conversion. Conversion is continuous if external trigger input is kept at active state in level trigger mode. It is stopped only when trigger condition disappears. If edge trigger condition is selected, the high and low state must be kept at least 4 system clocks. Pulse that is shorter than this specification will be ignored.

6.5.4.7 PWM trigger

In Single-cycle Scan Mode, A/D conversion can be triggered by PWM request. When the HWTRGCOND (ADC_CTL[8]) bit is set to high to enable ADC external hardware trigger function, setting the HWTRGSEL (ADC_CTL[5:4]) bits to 11b is to select external hardware trigger input source from PWM trigger. When PWM trigger is enabled, setting PWMTRGDLY (ADC_CTL[23:16]) bits can insert a delay time between PWM trigger condition and ADC start conversion.

6.5.4.8 Conversion Result Monitor by Compare Function

ADC controller provides two sets of compare register ADCMPR0 and ADCMPR1, to monitor maximum two specified channels conversion results from A/D conversion controller (refer to the following figure). Software can select which channel to be monitored by set CMPCH (ADC_CMPx[6:3]) and CMPCOND (ADC_CMPx[2]) bit is used to check conversion result is less

than specified value or greater than (equal to) value specified in CMPDAT (ADC_CMPx[27:16]). When the conversion of the channel specified by CMPCH (ADC_CMPx[6:3]) is completed, the comparing action will be triggered one time automatically. When the compare result meets the setting, compare match counter will increase 1, otherwise, the compare match counter will be clear to 0. When counter value reach the setting of CMPMCNT (ADC_CMPx[11:8]) + 1, then ADCMPF0/1 (ADC_STATUS0[2:1]) bit will be set to 1, if ADCMPIE (ADC_CMPx[1]) bit is set then an ADC_INT interrupt request is generated. Software can use it to monitor the external analog input pin voltage transition in scan mode without imposing a load on software. Detailed logics diagram is shown below:

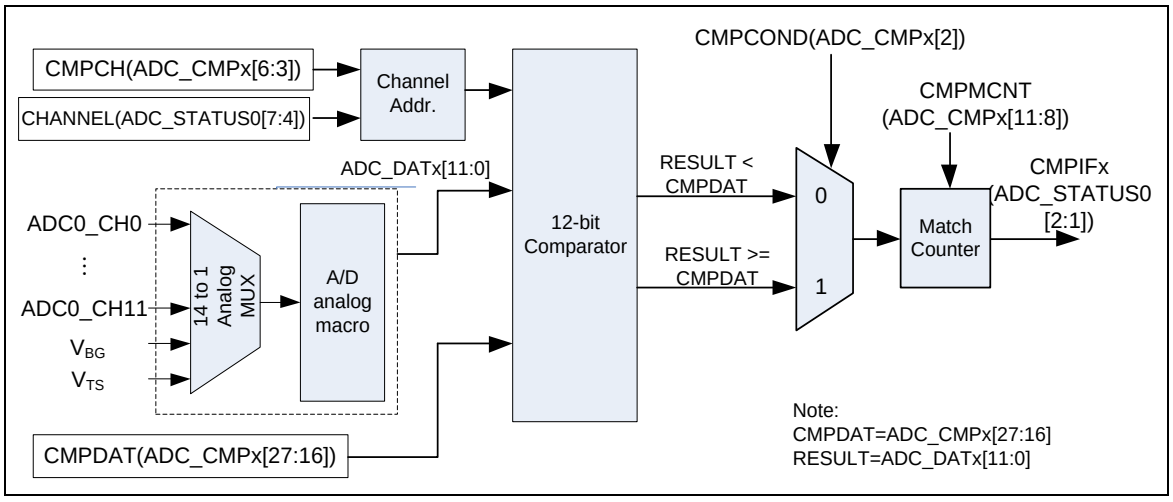


Figure 6.5-6 A/D Conversion Result Monitor Block Diagram

6.5.4.9 Interrupt Sources

There are three interrupt sources of ADC interrupt. When an ADC operation mode finishes its conversion, the A/D conversion end flag, ADIF (ADC_STATUS0[0]), will be set to 1. The ADCMPF0 (ADC_STATUS0[1]) and ADCMPF1 (ADC_STATUS0[2]) are the compare flags of compare function. When the conversion result meets the settings of ADCMPR0/1, the corresponding flag will be set to 1. When one of the flags, ADIF (ADC_STATUS0[0]), ADCMPF0 (ADC_STATUS0[1]) and ADCMPF1(ADC_STATUS0[2]), is set to 1 and the corresponding interrupt enable bit, ADCIEN (ADC_CTL[1]) and ADCMPIE (ADC_CMPx[1]), is set to 1, the ADC interrupt will be asserted. Software can clear the flag to revoke the interrupt request.

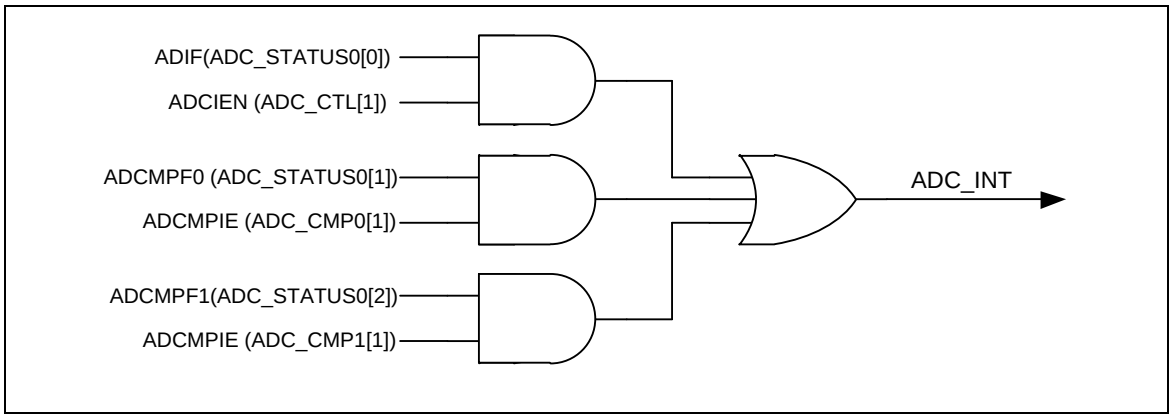


Figure 6.5-7 A/D Controller Interrupt

6.5.4.10 Peripheral DMA Request

A/D controller supports PDMA transfer for A/D conversion result of channel 0 to channel 11. When A/D conversion is finished, the conversion result will be loaded into ADC_DAT register and VALID (ADC_DATx[17]) bit will be set to 1. If the PDMAEN (ADC_CTL[9]) bit is set, ADC controller will generate a request to PDMA. User can use PDMA to transfer the conversion results to a user-specified memory space without CPU's intervention. The source address of PDMA operation is fixed at ADC_CURDAT, no matter what channels was selected. When PDMA is transferring the conversion result, ADC will continue converting the next selected channel if the operation mode of ADC is single scan mode or continuous scan mode. User can monitor current PDMA transfer data through reading ADC_CURDAT register. If ADC completes the conversion of a selected channel and the last conversion result of the same channel has not been transferred by PDMA, OV (ADC_DATx[16]) bit of the corresponding channel will be set and the last ADC conversion result will be overwrite by the new ADC conversion result. PDMA will transfer the latest data of selected channels to the user-specified destination address.

6.5.5 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
ADC Base Address: **ADC_BA = 0x4004_3000**				
ADC_DAT0	ADC_BA+0x00	R	ADC Data Register 0	0x0000_0000
ADC_DAT1	ADC_BA+0x04	R	ADC Data Register 1	0x0000_0000
ADC_DAT2	ADC_BA+0x08	R	ADC Data Register 2	0x0000_0000
ADC_DAT3	ADC_BA+0x0C	R	ADC Data Register 3	0x0000_0000
ADC_DAT4	ADC_BA+0x10	R	ADC Data Register 4	0x0000_0000
ADC_DAT5	ADC_BA+0x14	R	ADC Data Register 5	0x0000_0000
ADC_DAT6	ADC_BA+0x18	R	ADC Data Register 6	0x0000_0000
ADC_DAT7	ADC_BA+0x1C	R	ADC Data Register 7	0x0000_0000
ADC_DAT8	ADC_BA+0x20	R	ADC Data Register 8	0x0000_0000
ADC_DAT9	ADC_BA+0x24	R	ADC Data Register 9	0x0000_0000
ADC_DAT10	ADC_BA+0x28	R	ADC Data Register 10	0x0000_0000
ADC_DAT11	ADC_BA+0x2C	R	ADC Data Register 11	0x0000_0000
ADC_DAT12	ADC_BA+0x30	R	ADC Data Register 12 (for Band-gap Voltage)	0x0000_0000
ADC_DAT13	ADC_BA+0x34	R	ADC Data Register 13 (for Temperature Sensor)	0x0000_0000
ADC_CTL	ADC_BA+0x40	R/W	ADC Control Register	0x0000_0000
ADC_CHEN	ADC_BA+0x44	R/W	ADC Channel Enable Control Register	0x0000_0000
ADC_CMP0	ADC_BA+0x48	R/W	ADC Compare Register 0	0x0000_0000
ADC_CMP1	ADC_BA+0x4C	R/W	ADC Compare Register 1	0x0000_0000
ADC_STATUS0	ADC_BA+0x50	R/W	ADC Status Register 0	0x0000_0000
ADC_STATUS1	ADC_BA+0x54	R	ADC Status Register 1	0x0000_0000
ADC_CURDAT	ADC_BA+0x60	R	ADC PDMA Current Transfer Data Register	0x0000_0000

6.5.6 Register Description

ADC Data Registers (ADC_DAT0 ~ ADC_DAT 13)

Register	Offset	R/W	Description	Reset Value
ADC_DAT0	ADC_BA+0x00	R	ADC Data Register 0	0x0000_0000
ADC_DAT1	ADC_BA+0x04	R	ADC Data Register 1	0x0000_0000
ADC_DAT2	ADC_BA+0x08	R	ADC Data Register 2	0x0000_0000
ADC_DAT3	ADC_BA+0x0C	R	ADC Data Register 3	0x0000_0000
ADC_DAT4	ADC_BA+0x10	R	ADC Data Register 4	0x0000_0000
ADC_DAT5	ADC_BA+0x14	R	ADC Data Register 5	0x0000_0000
ADC_DAT6	ADC_BA+0x18	R	ADC Data Register 6	0x0000_0000
ADC_DAT7	ADC_BA+0x1C	R	ADC Data Register 7	0x0000_0000
ADC_DAT8	ADC_BA+0x20	R	ADC Data Register 8	0x0000_0000
ADC_DAT9	ADC_BA+0x24	R	ADC Data Register 9	0x0000_0000
ADC_DAT10	ADC_BA+0x28	R	ADC Data Register 10	0x0000_0000
ADC_DAT11	ADC_BA+0x2C	R	ADC Data Register 11	0x0000_0000
ADC_DAT12	ADC_BA+0x30	R	ADC Data Register 12 (for Band-gap Voltage)	0x0000_0000
ADC_DAT13	ADC_BA+0x34	R	ADC Data Register 13 (for Temperature Sensor)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						**VALID**	**OV**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESULT[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESULT[7:0]							

Bits	Description	
[31:18]	**Reserved**	Reserved.

[17]	**VALID**	**Valid Flag (Read Only)** 0 = Data in RESULT (ADC_DATx[15:0]) bits is not valid. 1 = Data in RESULT (ADC_DATx[15:0]) bits is valid. This bit is set to 1 when corresponding channel analog input conversion is completed and cleared by hardware after ADC_DAT register is read.
[16]	**OV**	**Overrun Flag (Read Only)** 0 = Data in RESULT (ADC_DATx[15:0]) is recent conversion result. 1 = Data in RESULT (ADC_DATx[15:0]) is overwrite. If converted data in RESULT[15:0] has not been read before new conversion result is loaded to this register, OV is set to 1 and previous conversion result is gone. It is cleared by hardware after ADC_DAT register is read.
[15:0]	**RESULT**	**A/D Conversion Result** This field contains conversion result of ADC. When DMOF (ADC_CTL[31]) bit is set to 0, 12-bit ADC conversion result with unsigned format will be filled in RESULT[11:0] and zero will be filled in RESULT[15:12]. When DMOF (ADC_CTL[31]) bit set to 1, 12-bit ADC conversion result with 2'complement format will be filled in RESULT[11:0] and signed bits to will be filled in RESULT[15:12].

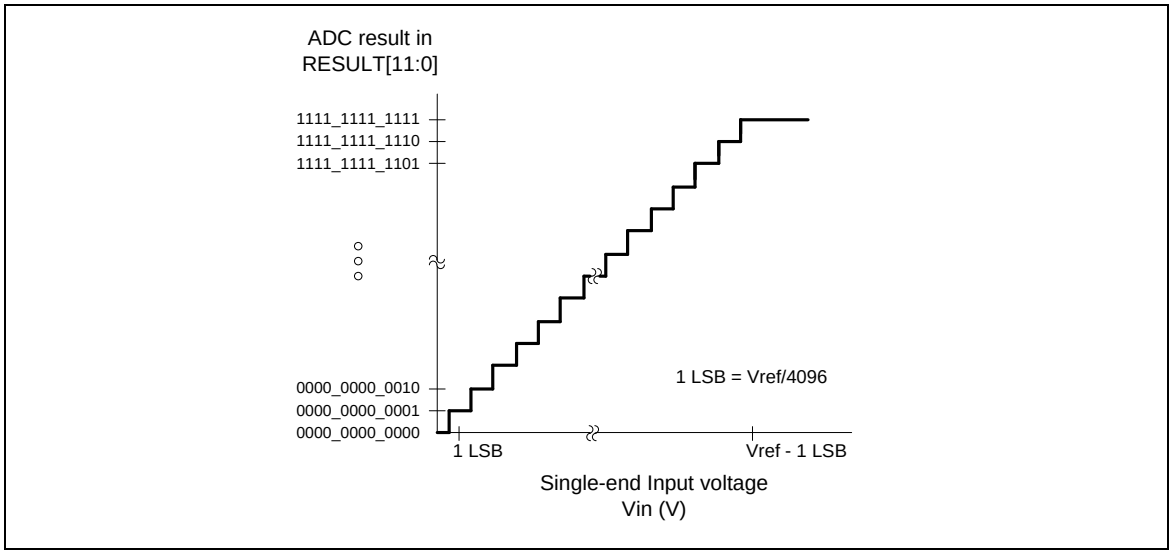


Figure 6.5-8 Conversion Result Mapping Diagram of Single-end Input

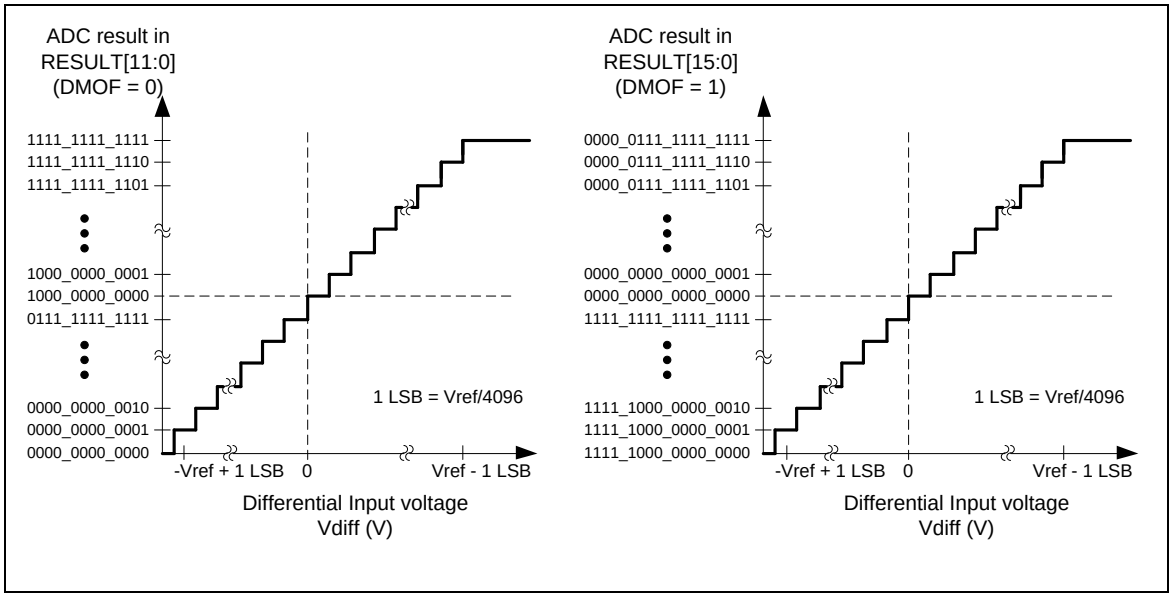


Figure 6.5-9 Conversion Result Mapping Diagram of Differential Input

ADC Control Register (ADC_CTL)

Register	Offset	R/W	Description	Reset Value
ADC_CTL	ADC_BA+0x40	R/W	ADC Control Register	0x0000_0000

31	30	29	28	27	26	25	24
DMOF	Reserved						
23	22	21	20	19	18	17	16
PWMTRGDLY							
15	14	13	12	11	10	9	8
Reserved				SWTRG	DIFFEN	PDMAEN	HWTRGEN
7	6	5	4	3	2	1	0
HWTRGCOND		HWTRGSEL		OPMODE		ADCIEN	ADCEN

Bits	Description	
[31]	**DMOF**	**ADC Differential Input Mode Output Format** 0 = A/D conversion result will be filled in RESULT (ADC_DATx[15:0]) registers with unsigned format. 1 = A/D conversion result will be filled in RESULT (ADC_DATx[15:0]) registers with 2'complement format.
[30:24]	**Reserved**	Reserved.
[23:16]	**PWMTRGDLY**	**PWM Trigger Delay Time** Setting this field will delay ADC start conversion time after PWM trigger comes. PWM trigger delay time is 4 * system clock * PWMTRGDLY (ADC_CTL[23:16])
[15:12]	**Reserved**	Reserved.
[11]	**SWTRG**	**A/D Conversion Start** 0 = Conversion stopped and A/D converter enter idle state. 1 = Conversion start. The SWTRG (ADC_CTL[11]) bit can be set to 1 from two sources: software and hardware trigger. The SWTRG (ADC_CTL[11]) bit will be cleared to 0 by hardware automatically at the ends of single mode and single-cycle scan mode. In continuous scan mode, A/D conversion is continuously performed until software write 0 to this bit or chip reset.

[10]	**DIFFEN**	**Differential Input Mode Enable Bit** 0 = Single-end analog input mode. 1 = Differential analog input mode. The A/D analog input ADC0_CH0/ADC0_CH1 consists of a differential pair. So as ADC0_CH2/ADC0_CH3, ADC0_CH4/ADC0_CH5, ADC0_CH6/ADC0_CH7, ADC0_CH8/ADC0_CH9 and ADC0_CH10/ADC0_CH11. The even channel defines as plus analog input voltage (V_{plus}) and the odd channel defines as minus analog input voltage (V_{minus}). Differential input voltage (V_{diff}) = V_{plus} - V_{minus}, where V_{plus} is the analog input; V_{minus} is the inverted analog input. In differential input mode, only the even number of the two corresponding channels needs to be enabled in ADCHER (ADC_CHEN[11:0]). The conversion result will be placed to the corresponding data register of the enabled channel.
[9]	**PDMAEN**	**PDMA Transfer Enable Bit** 0 = PDMA data transfer Disabled. 1 = PDMA data transfer in ADC_DATx Enabled. When A/D conversion is completed, the converted data is loaded into ADC_DATx, software can enable this bit to generate a PDMA data transfer request. When PDMAEN (ADC_CTL[9]) is set to 1, software must set ADCIEN (ADC_CTL[1]) bit to 0 to disable interrupt.
[8]	**HWTRGEN**	**External Hardware Trigger Enable Bit** Enable or disable hardware triggering of A/D conversion. The hardware trigger source include external pin (STADC) or PWM trigger which is controlled by HWTRGSEL (ADC_CTL[5:4]) register. 0 = Disabled. 1 = Enabled. ADC hardware trigger function is only supported in single-cycle scan mode.
[7:6]	**HWTRGCOND**	**External Pin Trigger Conditions** These two bits decide external pin (STADC) trigger event. The signal must be kept at stable state at least 8 system clocks for level trigger and 4 system clocks at high and low state for edge trigger. 00 = Low level. 01 = High level. 10 = Falling edge. 11 = Rising edge.
[5:4]	**HWTRGSEL**	**External Hardware Trigger Source** 00 = A/D conversion is started by external pin (STADC). 01 = Reserved. 10 = Reserved. 11 = PWM0 or PWM1 trigger condition is matched. Software should disable HWTRGCOND (ADC_CTL[8]) and SWTRG (ADC_CTL[11]) before changing HWTRGSEL (ADC_CTL[5:4]). In hardware trigger mode, the SWTRG (ADC_CTL[11]) bit is set by hardware trigger source.

[3:2]	**OPMODE**	**ADC Operation Mode** 00 = Single conversion. 01 = Reserved. 10 = Single-cycle scan. 11 = Continuous scan. When changing the operation mode, software should disable SWTRG (ADC_CTL[11]) bit firstly.
[1]	**ADCIEN**	**ADC Interrupt Enable Bit** 0 = ADC interrupt function Disabled. 1 = ADC interrupt function Enabled. A/D conversion end interrupt request is generated if ADCIEN (ADC_CTL[1]) bit is set to 1.
[0]	**ADCEN**	**ADC Enable Bit** 0 = ADC analog circuit Disabled. 1 = ADC analog circuit Enabled. Before disabling ADC clock, this bit should be cleared to 0 by software.

ADC Channel Enable Control Register (ADC_CHEN)

Register	Offset	R/W	Description	Reset Value
ADC_CHEN	ADC_BA+0x44	R/W	ADC Channel Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						**ADTSEN**	**ADBGEN**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**CHEN[11:8]**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CHEN[7:0]							

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17]	**ADTSEN**	**Internal Temperature Sensor Selection** 0 = Internal temperature sensor is not selected to be the analog input source of ADC. 1 = Internal temperature sensor is selected to be the analog input source of ADC. ADC can only work at Single mode when software selects the temperature sensor voltage as the analog input source of ADC.
[16]	**ADBGEN**	**Internal Band-Gap Selection** 0 = Internal band-gap is not selected to be the analog input source of ADC. 1 = Internal band-gap is selected to be the analog input source of ADC. ADC can only work at Single mode when software selects the band-gap voltage as the analog input source of ADC
[15:12]	**Reserved**	Reserved.
[11:0]	**CHEN**	**Analog Input Channel Enable Bit** Set CHEN (ADC_CHEN[11:0]) to enable the corresponding analog input channel (ADC0_CH1 ~ ADC0_CH11). If DIFFEN bit is set to 1, only the even number channels need to be enabled 0 = ADC input channel Disabled. 1 = ADC input channel Enabled.

ADC Compare Register 0/1 (ADC_CMP0/1)

Register	Offset	R/W	Description	Reset Value
ADC_CMP0	ADC_BA+0x48	R/W	ADC Compare Register 0	0x0000_0000
ADC_CMP1	ADC_BA+0x4C	R/W	ADC Compare Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				**CMPDAT[11:8]**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CMPDAT[7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**CMPMCNT**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**CMPCH**				**CMPCOND**	**ADCMPIE**	**ADCMPEN**

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:16]	**CMPDAT**	**Compared Data** When DMOF (ADC_CTL[31]) bit is set to 0, ADC comparator compares CMPDAT (ADC_CTL[27:16]) with conversion result with unsigned format. CMPDAT (ADC_CTL[27:16]) should be filled in unsigned format. When DMOF (ADC_CTL[31]) bit is set to 1, ADC comparator compares CMPDAT (ADC_CTL[27:16]) with conversion result with 2'complement format. CMPDAT (ADC_CTL[27:16]) should be filled in 2'complement format.
[15:12]	**Reserved**	Reserved.
[11:8]	**CMPMCNT**	**Compare Match Count** When the specified ADC channel analog conversion result matches the compare condition defined by CMPCOND (ADC_CMPx[2]), the internal match counter will increase 1. When the internal counter reaches the value to CMPMCNT (ADC_CMPx[11:8]) + 1, the ADCMPFx (ADC_STATUS0[2:1]) bit will be set.
[7]	**Reserved**	Reserved.

[6:3]	**CMPCH**	**Compare Channel Selection** 0000 = Channel 0 conversion result is selected to be compared. 0001 = Channel 1 conversion result is selected to be compared. 0010 = Channel 2 conversion result is selected to be compared. 0011 = Channel 3 conversion result is selected to be compared. 0100 = Channel 4 conversion result is selected to be compared. 0101 = Channel 5 conversion result is selected to be compared. 0110 = Channel 6 conversion result is selected to be compared. 0111 = Channel 7 conversion result is selected to be compared. 1000 = Channel 8 conversion result is selected to be compared. 1001 = Channel 9 conversion result is selected to be compared. 1010 = Channel 10 conversion result is selected to be compared. 1011 = Channel 11 conversion result is selected to be compared. 1100 = band-gap voltage result is selected to be compared. 1101 = temperature sensor conversion result is selected to be compared. Others = reserved.
[2]	**CMPCOND**	**Compare Condition** 0 = Set the compare condition as that when a 12-bit A/D conversion result is less than the 12-bit CMPDAT (ADC_CMPx[27:16]), the internal match counter will increase one. 1 = Set the compare condition as that when a 12-bit A/D conversion result is greater or equal to the 12-bit CMPDAT (ADC_CMPx[27:16]), the internal match counter will increase one. **Note:** When the internal counter reaches the value to CMPMCNT (ADC_CMPx[11:8]) + 1, the ADCMPFx (ADC_STATUS0[2:1]) bit will be set.
[1]	**ADCMPIE**	**Compare Interrupt Enable Bit** 0 = Compare function interrupt Disabled. 1 = Compare function interrupt Enabled. If the compare function is enabled and the compare condition matches the setting of CMPCOND (ADC_CMPx[2]) and CMPMCNT(ADC_CMPx[11:8]), ADCMPFx (ADC_STATUS0[2:1]) bit will be asserted, in the meanwhile, if ADCMPIE (ADC_CMPx[1])is set to 1, a compare interrupt request is generated.
[0]	**ADCMPEN**	**Compare Enable Bit** 0 = Compare function Disabled. 1 = Compare function Enabled. Set this bit to 1 to enable ADC controller to compare CMPDAT (ADC_CMPx[27:16]) with the conversion result of the channel specified by CMPCH (ADC_CMPx[6:3]) when the conversion data of the specified channel is loaded into ADC_DATx register.

ADC Status Register 0 (ADC_STATUS0)

Register	Offset	R/W	Description	Reset Value
ADC_STATUS0	ADC_BA+0x50	R/W	ADC Status Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
CHANNEL				BUSY	ADCMPF1	ADCMPF0	ADIF

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:4]	**CHANNEL**	**Current Conversion Channel (Read Only)** This field reflects the current conversion channel when BUSY (ADC_STATUS0[3]) = 1. When BUSY (ADC_STATUS0[3]) = 0, it shows the number of the next converted channel.
[3]	**BUSY**	**BUSY/IDLE (Read Only)** 0 = ADC is in idle state. 1 = ADC is doing conversion. This bit is mirror of as SWTRG (ADC_CTL[11]) bit.
[2]	**ADCMPF1**	**Compare Flag** When the selected channel A/D conversion result meets setting condition in ADCMPR1 then this bit is set to 1. And it is cleared by writing 1 to self. 0 = Conversion result in ADC_DATx does not meet ADCMPR1 setting. 1 = Conversion result in ADC_DATx meets ADCMPR1 setting.
[1]	**ADCMPF0**	**Compare Flag** When the selected channel A/D conversion result meets setting condition in ADCMPR0 then this bit is set to 1. And it is cleared by writing 1 to self. 0 = Conversion result in ADC_DATx does not meet ADCMPR0 setting. 1 = Conversion result in ADC_DATx meets ADCMPR0 setting.
[0]	**ADIF**	**ADC Interrupt Flag** A status flag that indicates the end of A/D conversion. ADIF (ADC_STATUS0[0]) is set to 1 at these two conditions: 1. When A/D conversion ends in Single mode 2. When A/D conversion ends on all specified channels in Scan mode **Note:** This flag can be cleared by writing 1 to it.

ADC Status Register 1 (ADC_STATUS1)

Register	Offset	R/W	Description	Reset Value
ADC_STATUS1	ADC_BA+0x54	R	ADC Status Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		OV[13:8]					
23	22	21	20	19	18	17	16
OV[7:0]							
15	14	13	12	11	10	9	8
Reserved		VALID[13:8]					
7	6	5	4	3	2	1	0
VALID[7:0]							

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:16]	**OV**	**Overrun Flag (Read Only)** It is a mirror to OV (ADC_DATx[16]) bit.
[15:14]	**Reserved**	Reserved.
[13:0]	**VALID**	**Data Valid Flag (Read Only)** It is a mirror of VALID (ADC_DATx[17]) bit.

ADC PDMA Current Transfer Data Register (ADC_CURDAT)

Register	Offset	R/W	Description	Reset Value
ADC_CURDAT	ADC_BA+0x60	R	ADC PDMA Current Transfer Data Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						CURDAT[17:16]	
15	14	13	12	11	10	9	8
CURDAT[15:8]							
7	6	5	4	3	2	1	0
CURDAT[7:0]							

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17:0]	**CURDAT**	**ADC PDMA Current Transfer Data Biit (Read Only)** When PDMA transferring, read this register can monitor current PDMA transfer data.

6.6 12-bit Analog-to-Digital Converter (Enhanced ADC)

6.6.1 Overview

The NUC442/NUC472 series contains two 12-bit successive approximation analog-to-digital converters (SAR A/D converter) with 16 external input channels and 5 internal channels. The two A/D converters ADC0 and ADC1 can be sampled with simultaneous or single sampling mode. The A/D converters can be started by software, PWM triggers, timer0~3 overflow pulse triggers, ADINT0, ADINT1 interrupt EOC pulse trigger and external pin (STADC) input signal.

6.6.2 Features

- Enhanced ADC mode with dual ADC
- Analog input voltage range: 0~Analog Supply Voltage from AV_{DD}.
- 12-bit resolution and 10-bit accuracy is guaranteed.
- Two SAR ADC converters, including ADC0 and ADC1
- Up to 16 single-end analog external input channels
- Up to 5 internal channels; ADC0 supports four internal channels, including temperature sensor, band-gap voltage, analog ground and OP amplifier 0; ADC1 supports only OP amplifier 1
- Four ADC interrupts with individual interrupt vectors

- Supports conversion rate 400 kSPS while VREF is between 2.5V~5.5V and up to 800 kSPS while VREF is between 4.5V~5.5V in single-end mode.
- Double buffer for channel 0~3 of each ADC0 and ADC1
- Two operating modes:
  - Single sampling mode: two ADC converters run at normal operation.
  - Simultaneous sampling mode: Allow two ADC converters can be sampled simultaneously.
- An A/D conversion can be started by:
  - Software write 1 to A/D start conversion bit (SWTRGx, x = 0~15)
  - External pin (STADC)
  - Timer0~3 overflow pulse triggers
  - ADINT0 and ADINT1 interrupt EOC pulse triggers
  - PWM triggers
- Conversion results are held in 16 data registers with valid and overrun indicators.
- Each of SAMPLE00~SAMPLE07 ADC control logic modules configurable for ADC0 converter channel ADC0_CH0~ADC0_CH7 and trigger source.
- Each of SAMPLE10~SAMPLE17 ADC control logic modules configurable for ADC1 converter channel ADC1_CH0~ADC1_CH7 and trigger source.
- ADC0 channel 8, 9, 10, 11 input sources as band-gap voltage, temperature sensor, analog ground and OP amplify 0.
- ADC1 channel 8 as OP amplify 1.

Note: if user configures bit 8 of VREFCR register to 1, the NUC442/NUC472 ADC becomes Enhanced ADC mode with dual ADC, if user configure bit 8 of VREFCR register to 0, the NUC442/NUC472 ADC become basic ADC mode with single ADC

6.6.3 Block Diagram

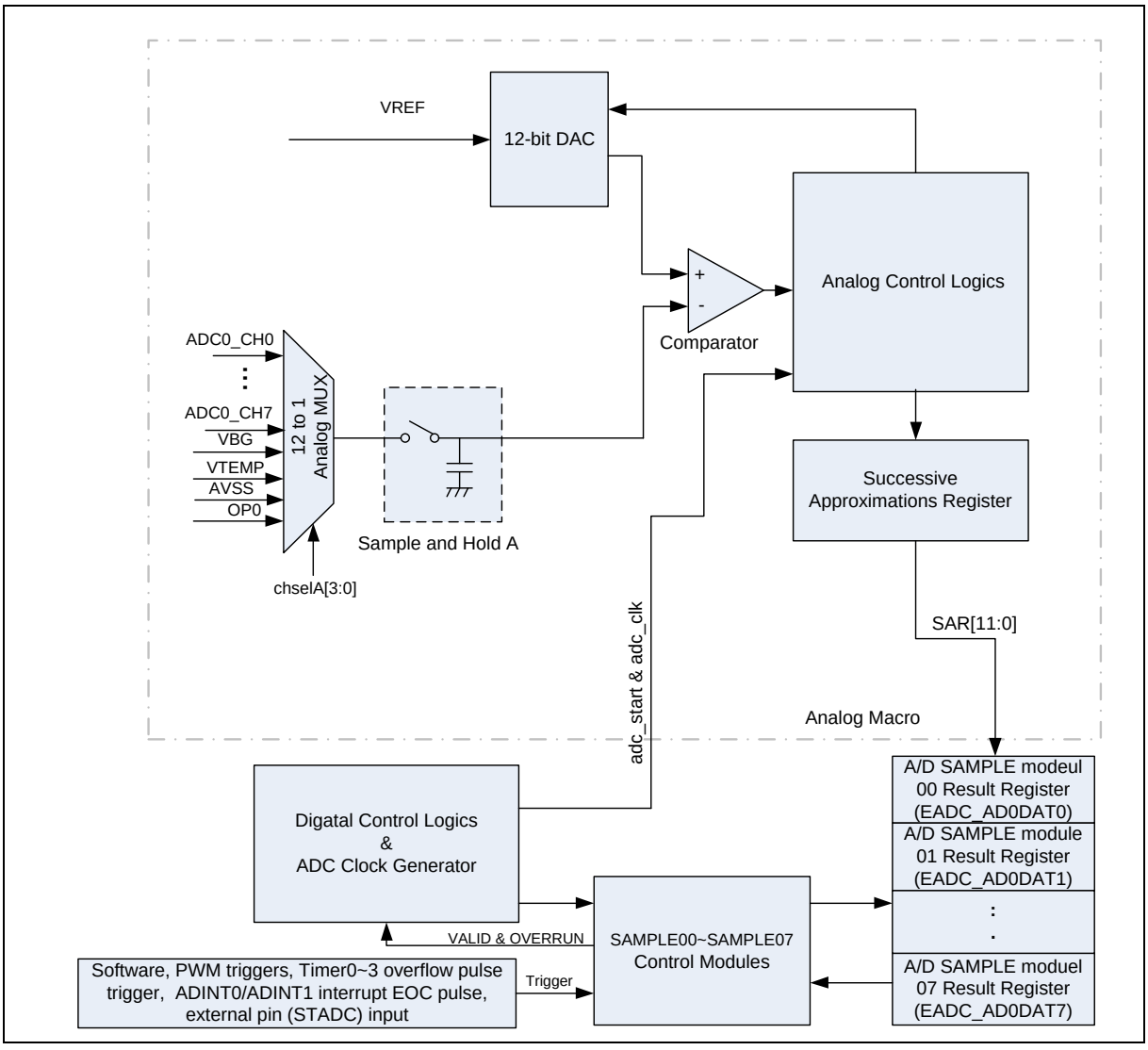


Figure 6.6-1 ADC0 Converter Block Diagram

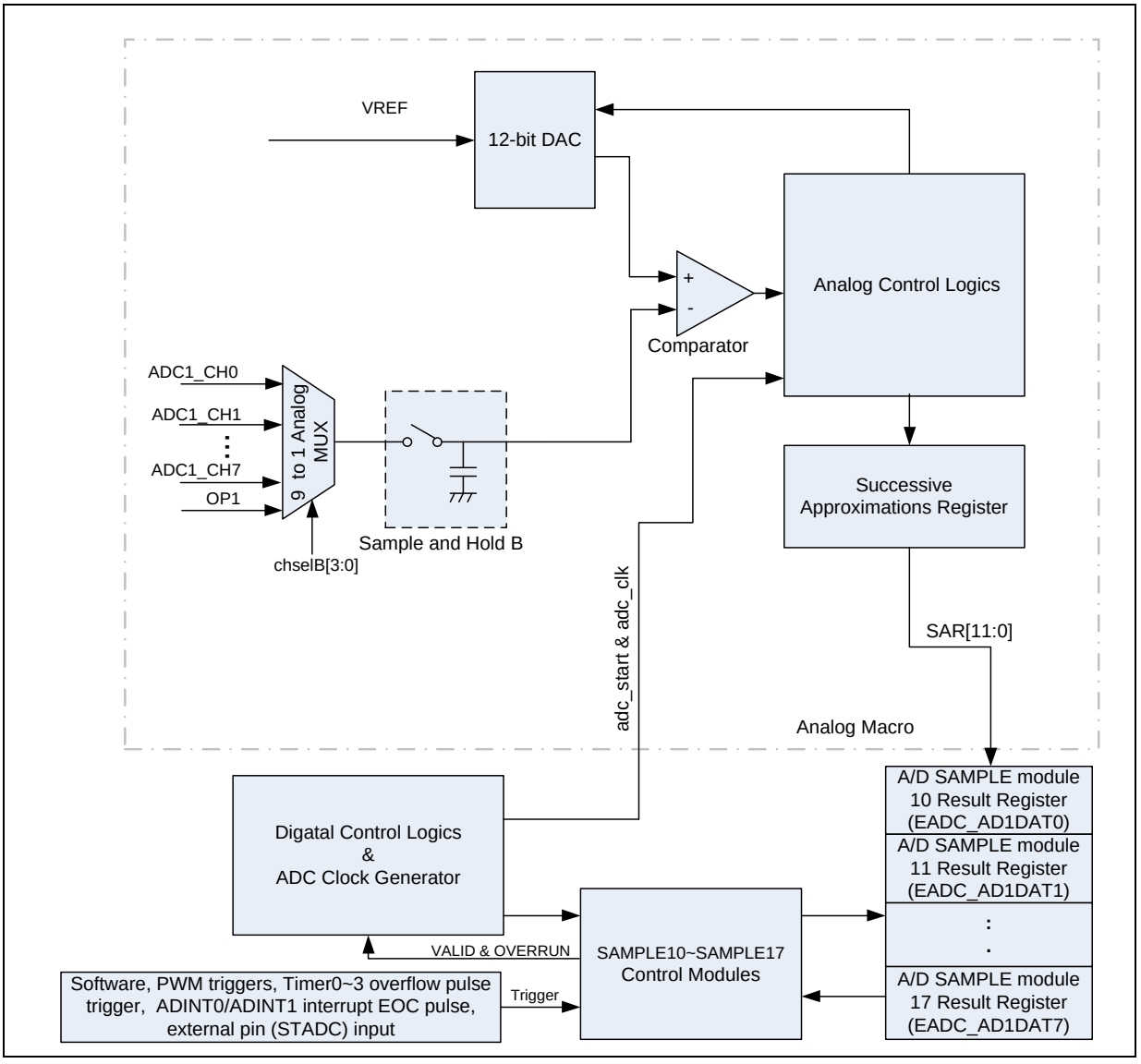


Figure 6.6-2 ADC1 Converter Block Diagram

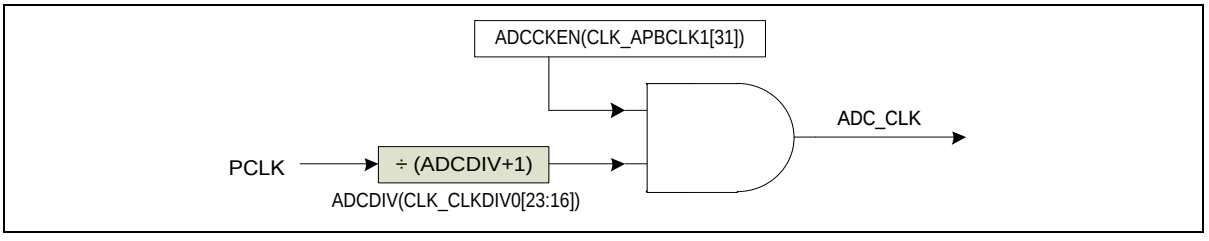


Figure 6.6-3 ADC Clock Control

6.6.4 Operation Procedure

There are two ADC converters (ADC0 and ADC1), each ADC converter consists of eight SAMPLE

control modules, and the A/D converter operates by successive approximation with 12-bit resolution.

The A/D operation is based on SAMPLE00~SAMPLE07 (ADC0 converter) and SAMPLE10~SAMPLE17 (ADC1 converter) control logic modules, each of them has its configuration to decide which trigger source to starts the conversion, which channel to convert. Different SAMPLE modules can be configured for the same channel, trigger source. It provides user a flexible means to get the over-sampling results.

The ADC conversion trigger sources are as below list:

- Software write 1 to A/D start conversion bit (SWTRGx, x = 0~15)
- External pin (STADC)
- Timer0~3 overflow pulse triggers
- ADINT0, ADINT1 ADC interrupt EOC pulse triggers
- PWM triggers

The ADINT0 or ADINT1 interrupt pulses are generated whenever the specific SAMPLE module A/D EOC (End of conversion) pulse is generated. ADINT0 or ADINT1 interrupt pulse triggers can be fed back to trigger another A/D conversion, and is useful if a continuous scan conversion is needed.

6.6.4.1 ADC Clock Generator

The maximum sampling rate is up to 800 kHz and the conversion time is less 1.25μs. It needs 20 ADC clocks to complete an A/D conversion. The ADC engine clock source is from PCLK clock, the ADC clock frequency is divided by an 8-bit pre-scalar with the formula:

The ADC clock frequency = (PCLK) / (ADCDIV+1);
where the 8-bit ADCDIV is located in register CLKDIV0[23:16].

In generally, software can set ADCDIV to get 16 MHz or slightly less.

6.6.4.2 ADC Single Sampling Mode

When a ADC conversion is performed on the SAMPLE module x specified single channel, the operations are as follows:

1. A/D conversion is started when the SWTRGx (EADC_SWTRG [15:0]) bit is set to 1 by software or other trigger inputs.
2. When A/D conversion is finished, the 12-bit result is stored in the ADC data register EADC_ADnDATx corresponding to the SAMPLE module x.
3. On completion of conversion, the ADIFn (EADC_STATUS1 [3:0]) bit is set to 1 and ADC interrupt (ADINTn) is requested if the ADIEn (EADC_CTL [5:2]) bit is set to 1.
4. The SWTRGx (EADC_SWTRG [15:0]) bit remains 1 during A/D conversion. When A/D conversion ends, the SWTRGx (EADC_SWTRG [15:0]) bit is automatically cleared to 0 and the A/D converter will do another pending conversion.

Note: If software or other trigger enables more than one channel in single or simultaneous sampling mode, the SAMPLE module specified channel with highest priority is converted and other enabled channels will be pended.

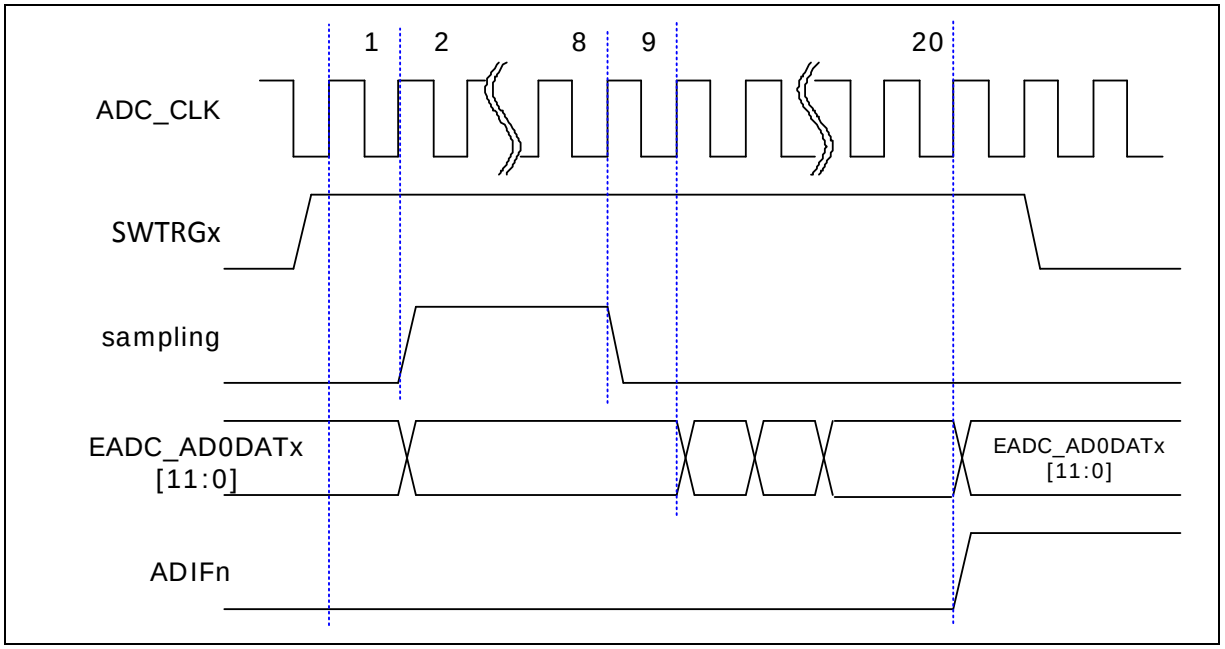


Figure 6.6-4 Single Sampling Mode Conversion Timing Diagram

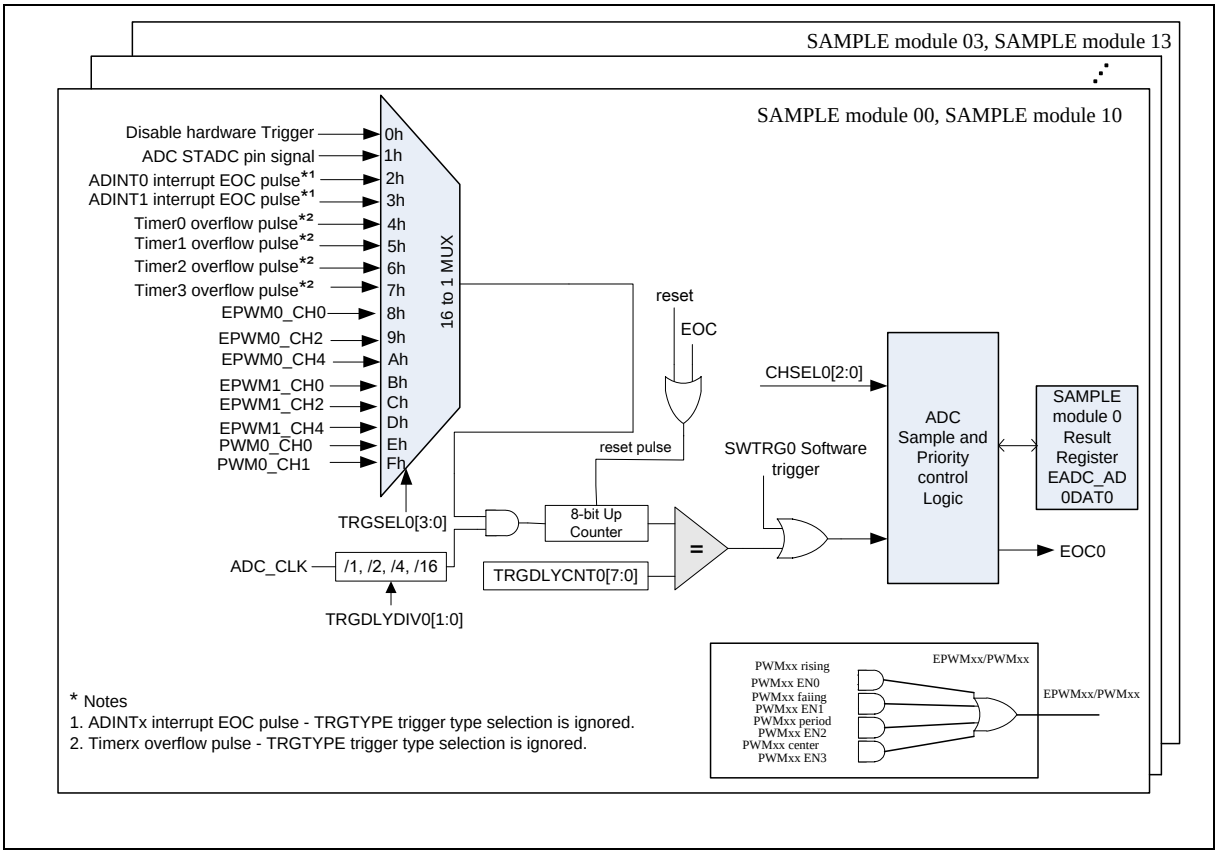


Figure 6.6-5 SAMPLE00~SAMPLE03 and SAMPLE10~SAMPLE13 Control Block Diagram

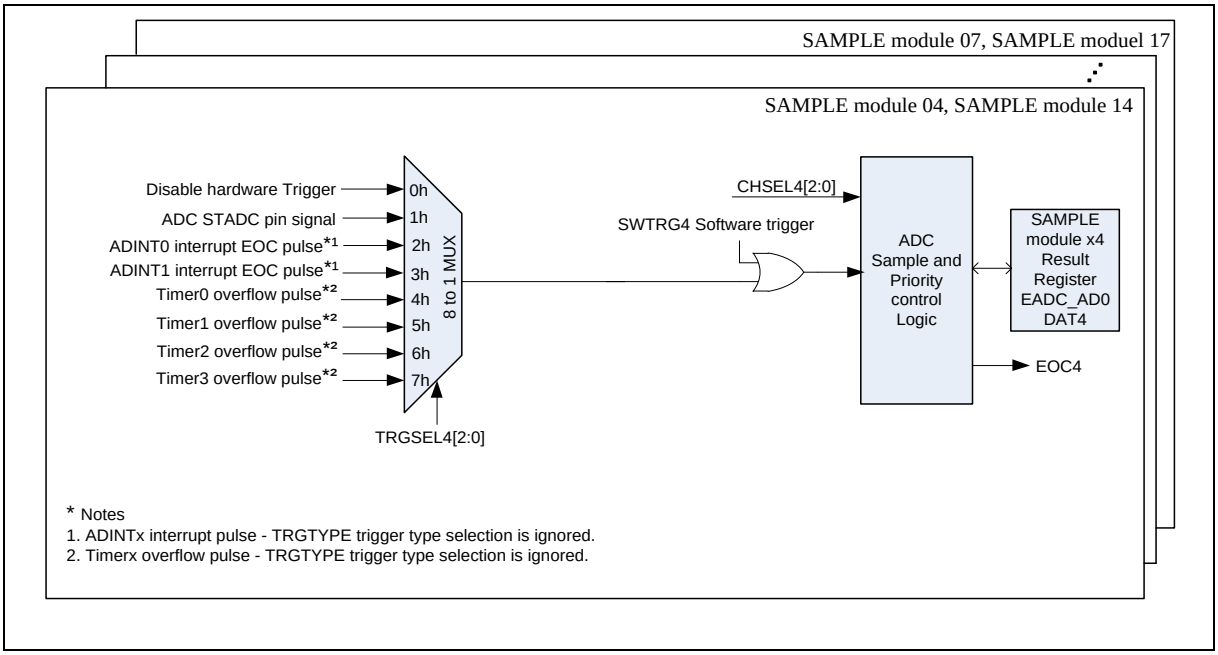


Figure 6.6-6 SAMPLE04~SAMPLE07 and SAMPLE14~SAMPLE17 Control Block Diagram

6.6.4.3 ADC Simultaneous Sampling Mode

The NUC442/NUC472 has two ADCs that allow two different ADC channels to be simultaneously sampled. The priority rules are same as the single sampling mode.

The same numbered of SAMPLE0 and the SAMPLE1 are coupled together. For example, SAMPLE00 and SAMPLE10 are coupled when simultaneous sampling mode enable bit SIMUSEL0 (ADSMSELR [0]) bit = 1.

The ADC simultaneous sampling mode conversion operations are as below:

1. Only SAMPLE0 trigger can start a pair of conversions.
2. SAMPLE0 assign an A-channel and SAMPLE1 also assign a B-channel in simultaneous sampling mode. The SAMPLE1 specified trigger will be ignored.
3. The A-channel conversion result is stored in specific SAMPLE0 EADC_AD0DAT0 register, and the conversion result of the B-channel is placed in the same numbered SAMPLE1 EADC_AD0DAT1 register.

For example:

If SIMUSEL2 (ADSMSELR [2]) bit = 1 and SAMPLE02 is configured to sample ADC0_CH4, it defines the SAMPLE02 and same numbered SAMPLE12 are coupled at simultaneous sampling conversion mode, if SAMPLE12 is configured to sample ADC1_CH3, the pair of ADC conversion channels are ADC0_CH4 and ADC1_CH3.

After a channel pair (ADC0_CH4, ADC1_CH3) of ADC conversion completes, the results of those two ADC conversions will be placed in registers EADC_AD0DAT2 and EADC_AD1DAT2.

6.6.4.4 ADC Conversion Priority

There are two priority groups converter for determining the conversion order when multiple SAMPLE modules trigger flags are set at the same time. SAMPLE00~SAMPLE07 priority group is for ADC0

converter, SAMPLE10~SAMPLE17 priority group is for ADC1 converter.

SAMPLE module with lower number has higher priority than the higher number SAMPLE module, if two SAMPLE modules are triggered at the same time, the SAMPLE module with lower number will start to convert ADC first.

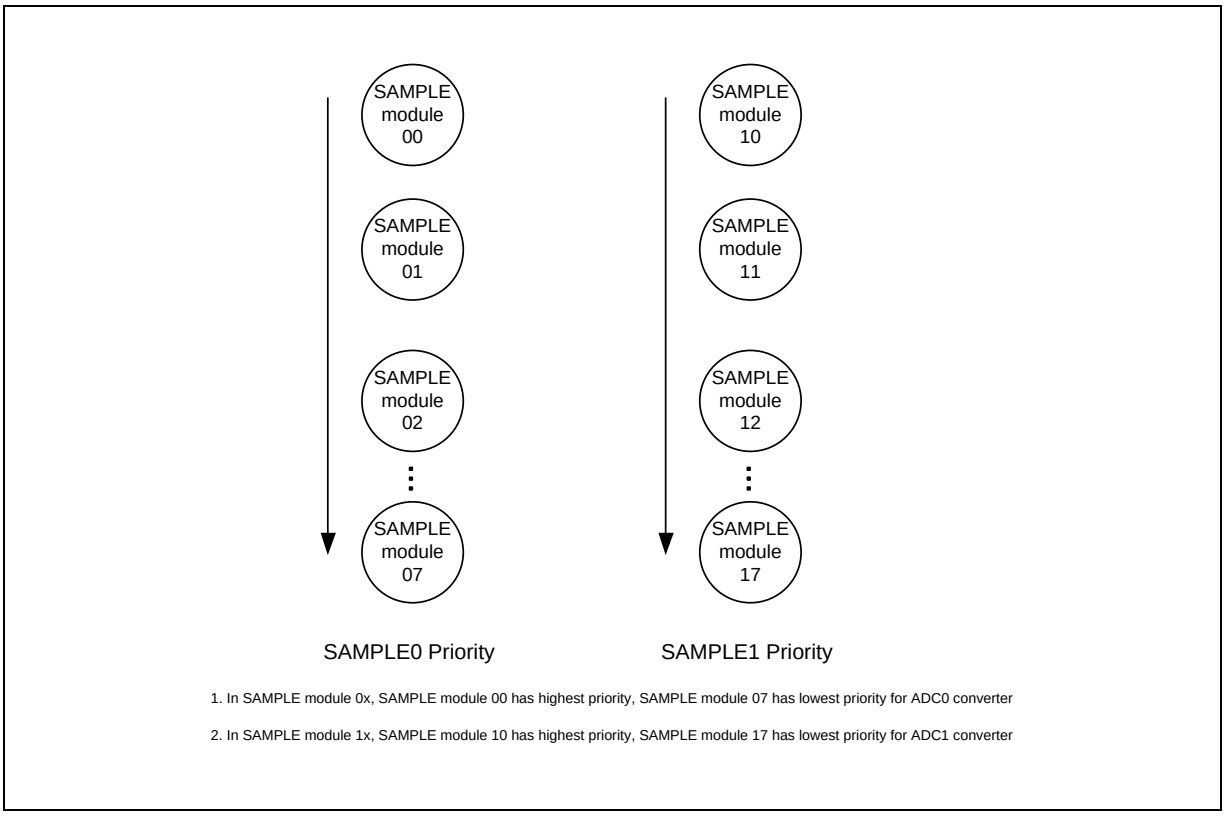
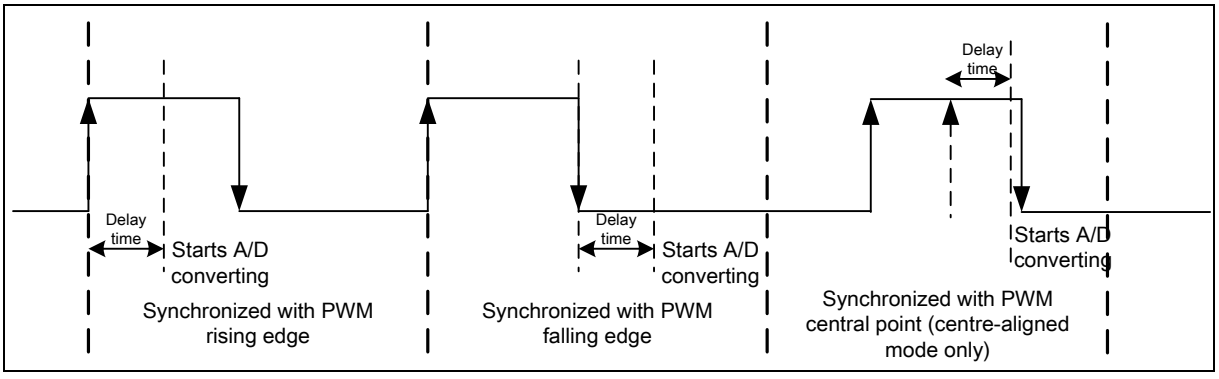


Figure 6.6-7 SAMPLE Module Conversion Priority Arbitrator Diagram

6.6.4.5 ADC Start Synchronous with PWM

Besides software start, ADCINT0, 1 interrupt pulse, and external pin (STADC) to start ADC conversion, this device has new feature to allow PWM channels to trigger the ADC start. User may configure PWM trigger types: rising, falling PWM edge or center point of PWM (center-aligned mode only) to trigger ADC start. The device also allow user to configure the amount of delay prior to ADC start after hardware detected the PWM edge. The following figure shows the programmable delay time for PWM-triggered ADC start conversion.



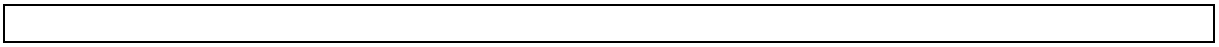


Figure 6.6-8 PWM-triggered ADC Start Conversion

6.6.4.6 ADC SAMPLE End of Conversion Interrupt Operation

There are 4 ADC interrupts ADINT0~3, and each of these interrupts has its own interrupt vector address and can be configured to select a specific SAMPLE module EOC pulse (SAMPLE00~SAMPLE07, SAMPLE10~SAMPLE17 End of conversion pulses) as its interrupt trigger source.

The ADINT0, ADINT1 interrupt pulses are generated whenever the specific SAMPLE A/D EOC (End of conversion) pulse is generated. It also can be the SAMPLE module conversion trigger sources, and user can use it to do the ADC continuous scan conversion.

Example for “Continuous scan”:

Step1. If ADC SAMPLE02’s EOC pulse is enabled to be the ADINT0 interrupt trigger, AD0SPIE2 (EADC_INTSRC0 [2]) bit is set to 1, and ADINT0 is selected as SAMPLE00, SAMPLE11, SAMPLE02 hardware conversion trigger.

Step2. Set software trigger SWTRG[2] bit to 1 to start a SAMPLE02 ADC conversion, after the conversion completes, it generates an EOC pulse signal and ADINT0 interrupt pulse at end of SAMPLE02 ADC conversion, ADINT0 interrupt pulse will trigger the SAMPLE00, SAMPLE11, SAMPLE02 to start the ADC conversions.

Step3. ADINT0 interrupt pulse repeats to trigger SAMPLE00, SAMPLE11, SAMPLE02 ADC conversions automatically.

Step4. Clear TRGSEL (EADC_AD0SPCTL2) bits to 0 to disable SAMPLE02 module’s ADINT0 interrupt pulse hardware trigger, if needs to stop the continuous scan.

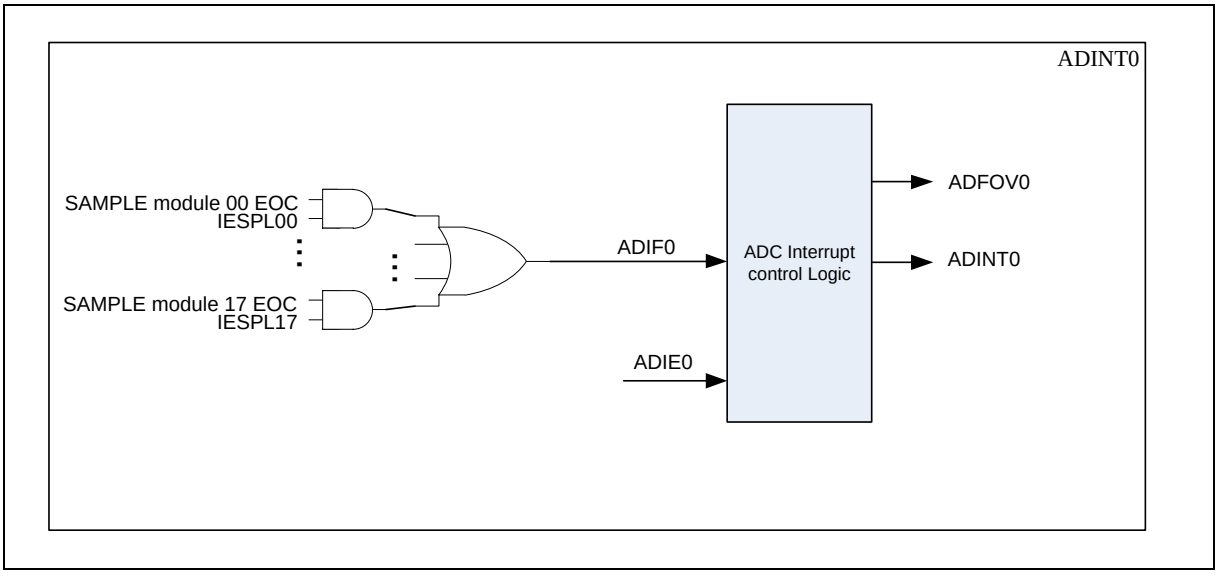


Figure 6.6-9 SAMPLE module A/D EOC Signal for ADINT0 Interrupt

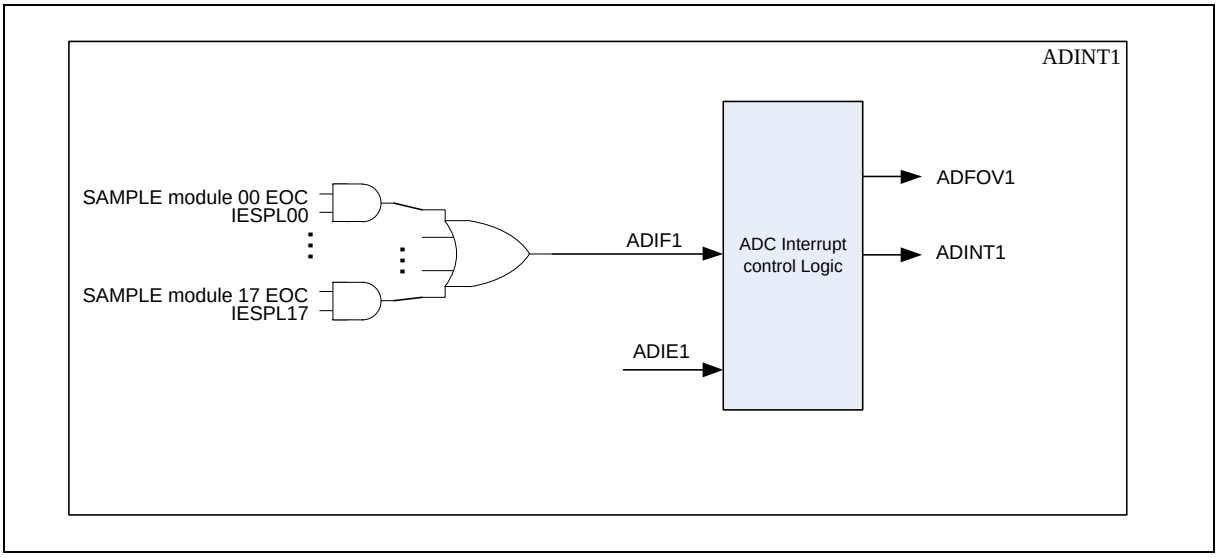


Figure 6.6-10 SAMPLE module A/D EOC Signal for ADINT1 Interrupt

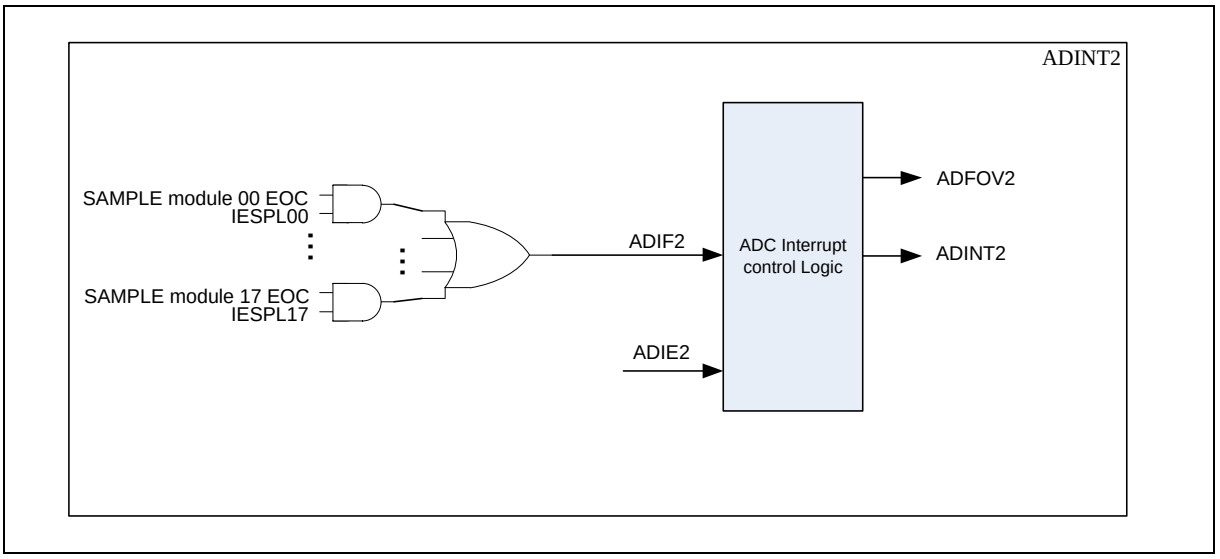


Figure 6.6-11 SAMPLE module A/D EOC Signal for ADINT2 Interrupt

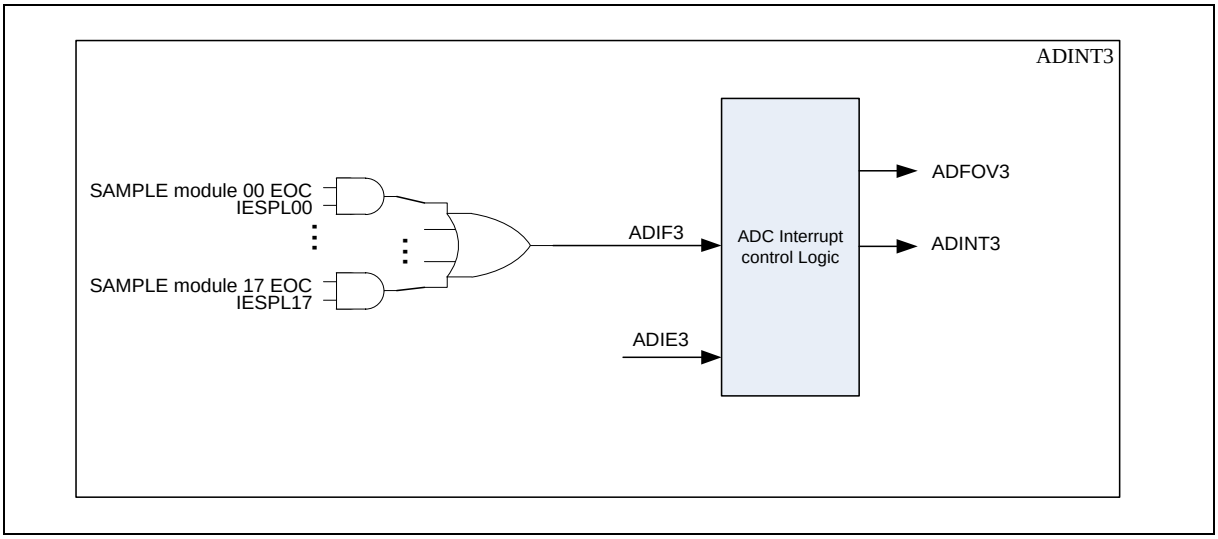


Figure 6.6-12 SAMPLE module A/D EOC Signal for ADINT3 Interrupt

6.6.4.7 Input Sampling and A/D Conversion Time

The A/D converter sample the analog input when A/D conversion start delay time (Td) has passed after SWTRGx (EADC_SWTRG [15:0]) bit is set to 1, then start conversion. Due to ADC clock is generated by PCLK divided by (N+1), the maximum delay time from APB write to A/D start sampling analog input time is 2N PCLKs. The start delay time is shown as below:

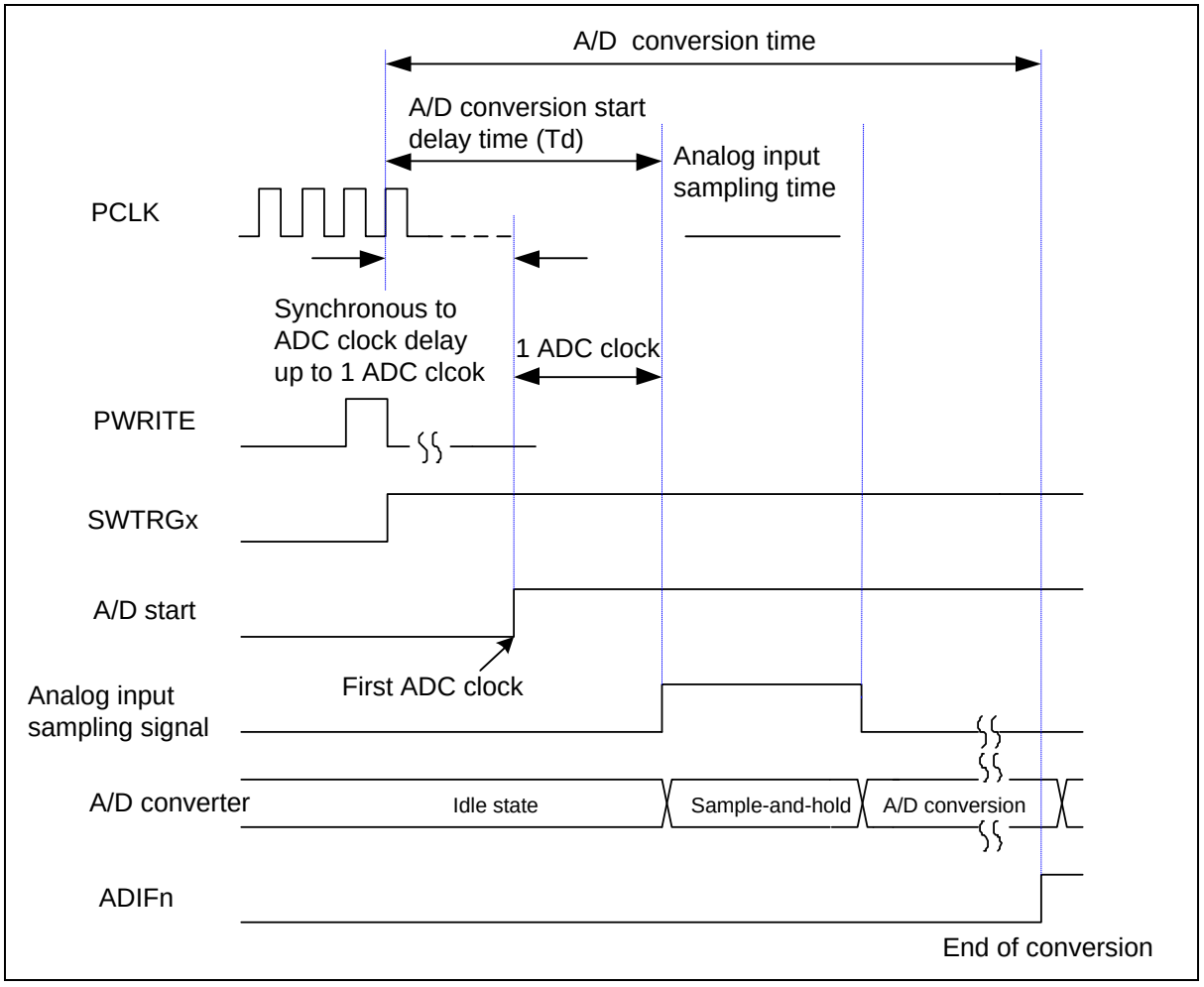


Figure 6.6-13 Conversion Start Delay Timing Diagram

A/D conversion can be triggered by external pin (STADC) request. Setting the TRGSEL (EADC_AD0SPCTLx[7:4]) bits to 01h is to select external trigger input from the external pin (STADC). Software can set EXTFEN (EADC_AD0SPCTLx[21]) and EXTREN (EADC_AD0SPCTLx[20]) to enable pin STADC trigger condition is falling or rising edge. An 8-bit sampling counter is used to deglitch. If edge trigger condition is selected, the high and low state must be kept at least 4 HCLKs. Pulse that is shorter than this specification will be ignored.

6.6.4.8 A/D Extend Sampling Time

When A/D operation at high ADC clock rate, the sampling time of analog input voltage may not enough if the analog channel has heavy loading to cause fully charge time is longer. SW can set A/D extend sampling time by writing EXTSMPTx(EADC_EXTSMPT [7:0]/ EADC_EXTSMPT [23:16]) register. The A/D extend sampling time is present between A/D controller judge which channel to be converting and A/D start to conversion. The range of extend sampling time is from 0 ~255 ADC clock. The extend sampling time is shown below:

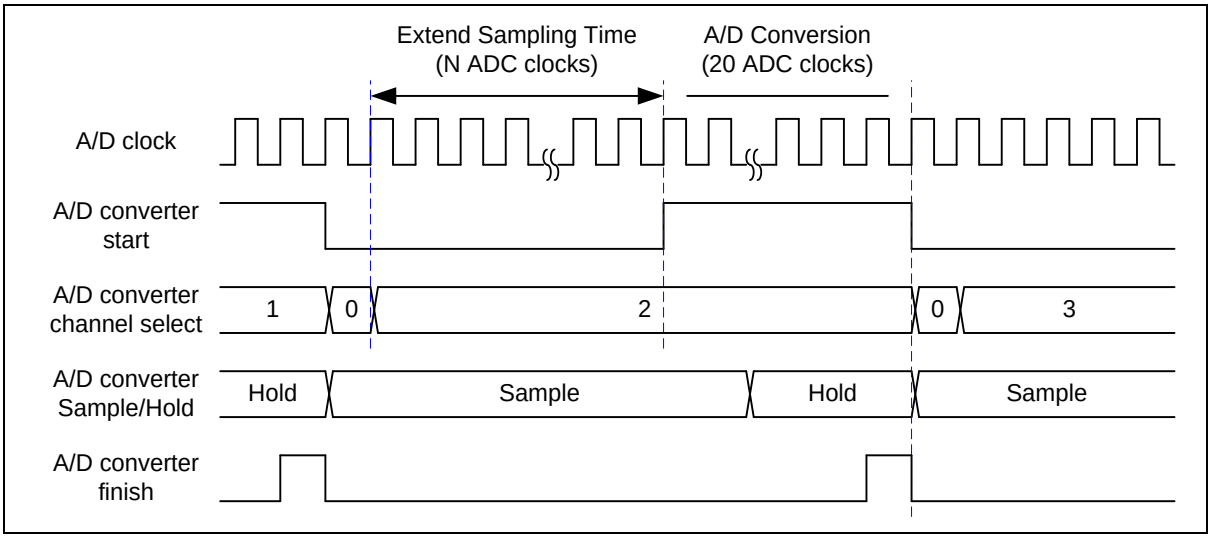


Figure 6.6-14 A/D Extend Sampling Timing Diagram

6.6.4.9 Conversion Result Monitor by Compare Mode

The NUC442/NUC472 controller provides two sets of compare register EADC_CMP0 and 1 to monitor maximum two specified SAMPLE00~SAMPLE03, SAMPLE10~SAMPLE13 conversion results from A/D conversion module, as shown in the following figure. Software can select which SAMPLE module result to be monitored by set CMPSPL(EADC_CMP[5:3]) and CMPCOND (EADC_CMPx[2]) bit is used to check conversion result is less than specify value or greater than (equal to) value specified in CMPDAT (EADC_CMPx[27:16]). When the conversion of the SAMPLE module specified by CMPSPL (EADC_CMPx[5:3]) is completed, the comparing action will be triggered one time automatically. When the compare result meets the setting, compare match counter will increase 1, when counter value reach the setting of CMPMATCNT (EADC_CMPx[11:8]) + 1 then ADCMPFx (EADC_STATUS1[7:6]) bit will be set to 1, if ADCMPIE (EADC_CMPx[1]) bit is set then an ADINT3 interrupt request is generated. Software can use it to monitor the external analog input pin voltage transition. Detailed logics diagram is shown below:

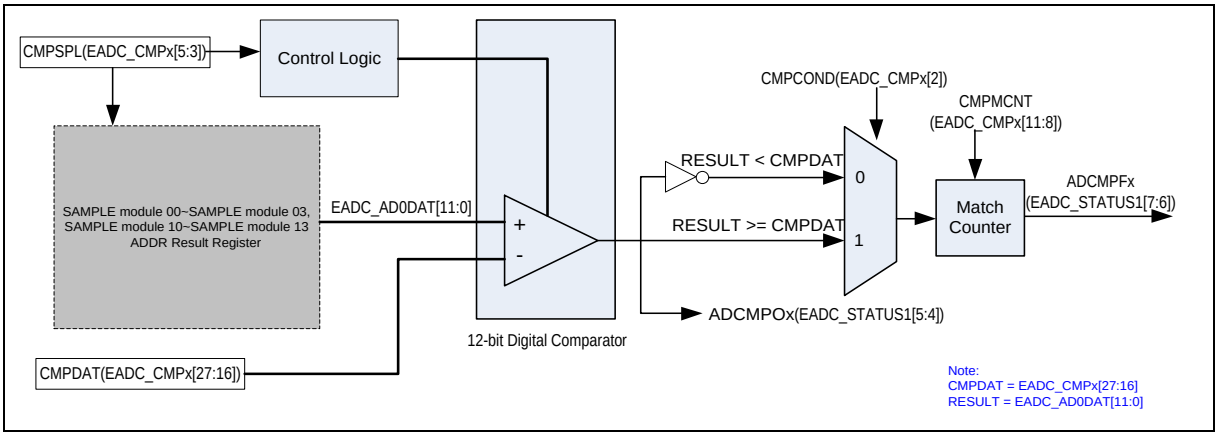


Figure 6.6-15 A/D Conversion Result Monitor Logics Diagram

6.6.4.10 Interrupt Sources

The A/D converter generates a conversion end ADIFn (EADC_STATUS1 [3:0]) bit register upon the end of specific SAMPLE module A/D conversion. If ADIEn (EADC_CTL [5:2]) bit is set then conversion end interrupt request ADINTn is generated.

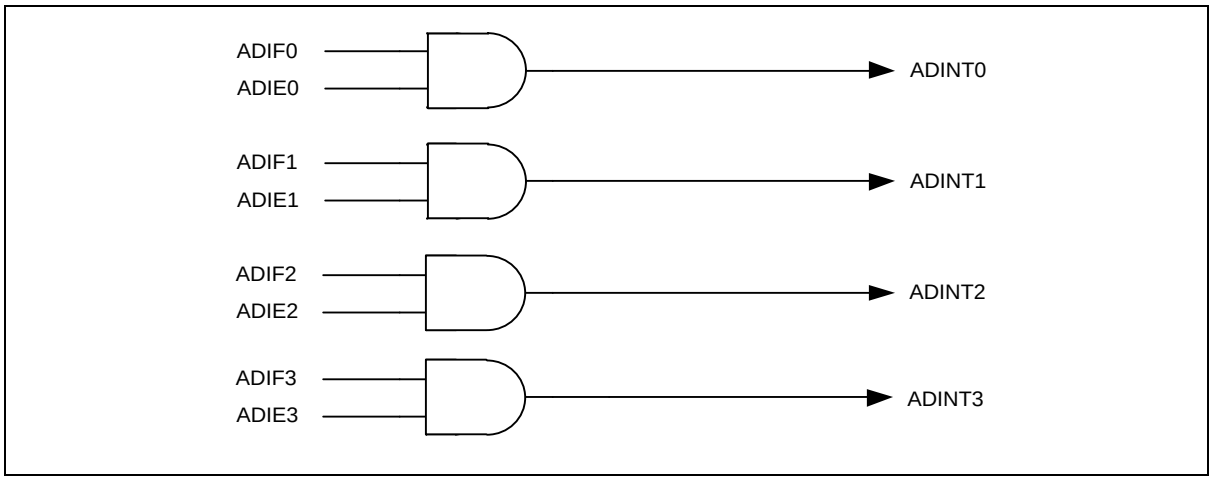


Figure 6.6-16 A/D Controller Interrupts

6.6.5 Register Map

R: read only, **W**: write only, **R/W**: both read and write, **C**: Only value 0 can be written

Register	Offset	R/W	Description	Reset Value
EADC Base Address: **EADC_BA = 0x4004_4000**				
EADC_AD0DAT0	EADC_BA+0x00	R	A/D Data Register 0 for SAMPLE00	0x0000_0000
EADC_AD0DAT1	EADC_BA+0x04	R	A/D Data Register 1 for SAMPLE01	0x0000_0000
EADC_AD0DAT2	EADC_BA+0x08	R	A/D Data Register 2 for SAMPLE02	0x0000_0000
EADC_AD0DAT3	EADC_BA+0x0C	R	A/D Data Register 3 for SAMPLE03	0x0000_0000
EADC_AD0DAT4	EADC_BA+0x10	R	A/D Data Register 4 for SAMPLE04	0x0000_0000
EADC_AD0DAT5	EADC_BA+0x14	R	A/D Data Register 5 for SAMPLE05	0x0000_0000
EADC_AD0DAT6	EADC_BA+0x18	R	A/D Data Register 6 for SAMPLE06	0x0000_0000
EADC_AD0DAT7	EADC_BA+0x1C	R	A/D Data Register 7 for SAMPLE07	0x0000_0000
EADC_AD1DAT0	EADC_BA+0x20	R	A/D Data Register 8 for SAMPLE10	0x0000_0000
EADC_AD1DAT1	EADC_BA+0x24	R	A/D Data Register 9 for SAMPLE11	0x0000_0000
EADC_AD1DAT2	EADC_BA+0x28	R	A/D Data Register 10 for SAMPLE12	0x0000_0000
EADC_AD1DAT3	EADC_BA+0x2C	R	A/D Data Register 11 for SAMPLE13	0x0000_0000
EADC_AD1DAT4	EADC_BA+0x30	R	A/D Data Register 12 for SAMPLE14	0x0000_0000
EADC_AD1DAT5	EADC_BA+0x34	R	A/D Data Register 13 for SAMPLE15	0x0000_0000
EADC_AD1DAT6	EADC_BA+0x38	R	A/D Data Register 14 for SAMPLE16	0x0000_0000
EADC_AD1DAT7	EADC_BA+0x3C	R	A/D Data Register 15 for SAMPLE17	0x0000_0000
EADC_CTL	EADC_BA+0x40	R/W	A/D Control Register	0x0000_0000
EADC_SWTRG	EADC_BA+0x48	W	A/D SAMPLE module Software Start Register	0x0000_0000
EADC_PENDSTS	EADC_BA+0x4C	R	A/D Start of Conversion Pending Flag Register	0x0000_0000
EADC_ADIFOV	EADC_BA+0x50	R/W	A/D ADINT3~0 Interrupt Flag Overrun Register	0x0000_0000
EADC_OVSTS	EADC_BA+0x54	R/W	A/D SAMPLE module Start of Conversion Overrun Flag Register	0x0000_0000
EADC_AD0SPCTL0	EADC_BA+0x58	R/W	A/D SAMPLE00 Control Register	0x0000_0000
EADC_AD0SPCTL1	EADC_BA+0x5C	R/W	A/D SAMPLE01 Control Register	0x0000_0000
EADC_AD0SPCTL2	EADC_BA+0x60	R/W	A/D SAMPLE02 Control Register	0x0000_0000
EADC_AD0SPCTL3	EADC_BA+0x64	R/W	A/D SAMPLE03 Control Register	0x0000_0000

Register	Offset	R/W	Description	Reset Value
EADC Base Address: **EADC_BA = 0x4004_4000**				
EADC_AD0SPCTL4	EADC_BA+0x68	R/W	A/D SAMPLE04 Control Register	0x0000_0000
EADC_AD0SPCTL5	EADC_BA+0x6C	R/W	A/D SAMPLE05 Control Register	0x0000_0000
EADC_AD0SPCTL6	EADC_BA+0x70	R/W	A/D SAMPLE06 Control Register	0x0000_0000
EADC_AD0SPCTL7	EADC_BA+0x74	R/W	A/D SAMPLE07 Control Register	0x0000_0000
EADC_AD1SPCTL0	EADC_BA+0x78	R/W	A/D SAMPLE10 Control Register	0x0000_0000
EADC_AD1SPCTL1	EADC_BA+0x7C	R/W	A/D SAMPLE11 Control Register	0x0000_0000
EADC_AD1SPCTL2	EADC_BA+0x80	R/W	A/D SAMPLE12 Control Register	0x0000_0000
EADC_AD1SPCTL3	EADC_BA+0x84	R/W	A/D SAMPLE13 Control Register	0x0000_0000
EADC_AD1SPCTL4	EADC_BA+0x88	R/W	A/D SAMPLE14 Control Register	0x0000_0000
EADC_AD1SPCTL5	EADC_BA+0x8C	R/W	A/D SAMPLE15 Control Register	0x0000_0000
EADC_AD1SPCTL6	EADC_BA+0x90	R/W	A/D SAMPLE16 Control Register	0x0000_0000
EADC_AD1SPCTL7	EADC_BA+0x94	R/W	A/D SAMPLE17 Control Register	0x0000_0000
EADC_SIMUSEL	EADC_BA+0xA4	R/W	A/D SAMPLE module Simultaneous Sampling Mode Select Register	0x0000_0000
EADC_CMP0	EADC_BA+0xA8	R/W	A/D Result Compare Register 0	0x0000_0000
EADC_CMP1	EADC_BA+0xAC	R/W	A/D Result Compare Register 1	0x0000_0000
EADC_STATUS0	EADC_BA+0xB0	R	A/D Status Register 0	0x0000_0000
EADC_STATUS1	EADC_BA+0xB4	R/W	A/D Status Register 1	0x0000_0000
EADC_EXTSMPT	EADC_BA+0xB8	R/W	A/D Timing Control Register	0x0000_0000
EADC_AD0DDAT0	EADC_BA+0x100	R	A/D double Data Register 0 for SAMPLE00	0x0000_0000
EADC_AD0DDAT1	EADC_BA+0x104	R	A/D double Data Register 1 for SAMPLE01	0x0000_0000
EADC_AD0DDAT2	EADC_BA+0x108	R	A/D double Data Register 2 for SAMPLE02	0x0000_0000
EADC_AD0DDAT3	EADC_BA+0x10C	R	A/D double Data Register 3 for SAMPLE03	0x0000_0000
EADC_AD1DDAT0	EADC_BA+0x120	R	A/D double Data Register 0 for SAMPLE10	0x0000_0000
EADC_AD1DDAT1	EADC_BA+0x124	R	A/D double Data Register 1 for SAMPLE11	0x0000_0000
EADC_AD1DDAT2	EADC_BA+0x128	R	A/D double Data Register 2 for SAMPLE12	0x0000_0000
EADC_AD1DDAT3	EADC_BA+0x12C	R	A/D double Data Register 3 for SAMPLE13	0x0000_0000
EADC_DBMEN	EADC_BA+0x130	R/W	A/D Double Buffer Mode select	0x0000_0000

Register	Offset	R/W	Description	Reset Value
EADC Base Address: **EADC_BA = 0x4004_4000**				
EADC_INTSRC0	EADC_BA+0x134	R/W	A/D Interrupt 0 Source Enable Control Register	0x0000_0000
EADC_INTSRC1	EADC_BA+0x138	R/W	A/D Interrupt 1 Source Enable Control Register	0x0000_0000
EADC_INTSRC2	EADC_BA+0x13C	R/W	A/D Interrupt 2 Source Enable Control Register	0x0000_0000
EADC_INTSRC3	EADC_BA+0x140	R/W	A/D Interrupt 3 Source Enable Control Register	0x0000_0000
EADC_AD0TRGEN0	EADC_BA+0x144	R/W	A/D trigger condition for SAMPLE00	0x0000_0000
EADC_AD0TRGEN1	EADC_BA+0x148	R/W	A/D trigger condition for SAMPLE01	0x0000_0000
EADC_AD0TRGEN2	EADC_BA+0x14C	R/W	A/D trigger condition for SAMPLE02	0x0000_0000
EADC_AD0TRGEN3	EADC_BA+0x150	R/W	A/D trigger condition for SAMPLE03	0x0000_0000
EADC_AD1TRGEN0	EADC_BA+0x154	R/W	A/D trigger condition for SAMPLE10	0x0000_0000
EADC_AD1TRGEN1	EADC_BA+0x158	R/W	A/D trigger condition for SAMPLE11	0x0000_0000
EADC_AD1TRGEN2	EADC_BA+0x15C	R/W	A/D trigger condition for SAMPLE12	0x0000_0000
EADC_AD1TRGEN3	EADC_BA+0x160	R/W	A/D trigger condition for SAMPLE13	0x0000_0000

6.6.6 Register Description

A/D Data Registers (EADC_AD0DAT0~EADC_AD0DAT7, EADC_AD1DAT0~EADC_AD1DAT7)

Register	Offset	R/W	Description	Reset Value
EADC_AD0DAT0	EADC_BA+0x00	R	A/D Data Register 0 for SAMPLE00	0x0000_0000
EADC_AD0DAT1	EADC_BA+0x04	R	A/D Data Register 1 for SAMPLE01	0x0000_0000
EADC_AD0DAT2	EADC_BA+0x08	R	A/D Data Register 2 for SAMPLE02	0x0000_0000
EADC_AD0DAT3	EADC_BA+0x0C	R	A/D Data Register 3 for SAMPLE03	0x0000_0000
EADC_AD0DAT4	EADC_BA+0x10	R	A/D Data Register 4 for SAMPLE04	0x0000_0000
EADC_AD0DAT5	EADC_BA+0x14	R	A/D Data Register 5 for SAMPLE05	0x0000_0000
EADC_AD0DAT6	EADC_BA+0x18	R	A/D Data Register 6 for SAMPLE06	0x0000_0000
EADC_AD0DAT7	EADC_BA+0x1C	R	A/D Data Register 7 for SAMPLE07	0x0000_0000
EADC_AD1DAT0	EADC_BA+0x20	R	A/D Data Register 8 for SAMPLE10	0x0000_0000
EADC_AD1DAT1	EADC_BA+0x24	R	A/D Data Register 9 for SAMPLE11	0x0000_0000
EADC_AD1DAT2	EADC_BA+0x28	R	A/D Data Register 10 for SAMPLE12	0x0000_0000
EADC_AD1DAT3	EADC_BA+0x2C	R	A/D Data Register 11 for SAMPLE13	0x0000_0000
EADC_AD1DAT4	EADC_BA+0x30	R	A/D Data Register 12 for SAMPLE14	0x0000_0000
EADC_AD1DAT5	EADC_BA+0x34	R	A/D Data Register 13 for SAMPLE15	0x0000_0000
EADC_AD1DAT6	EADC_BA+0x38	R	A/D Data Register 14 for SAMPLE16	0x0000_0000
EADC_AD1DAT7	EADC_BA+0x3C	R	A/D Data Register 15 for SAMPLE17	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						VALID	OV
15	14	13	12	11	10	9	8
Reserved				RESULT [11:8]			

7	6	5	4	3	2	1	0
RESULT[7:0]							

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17]	**VALID**	**Valid Flag** 0 = Data in RESULT (EADC_ADnDATx[11:0]) is not valid. 1 = Data in RESULT (EADC_ADnDATx[11:0]) is valid. This bit is set to 1 when corresponding SAMPLE module channel analog input conversion is completed and cleared by hardware after EADC_ADnDATx register is read. **Note:** n = 0~1 for ADC unit 0~1. **Note:** x = 0~7 for Sample 0~7.
[16]	**OV**	**Overrun Flag** 0 = Data in RESULT (EADC_ADnDATx[11:0]) is recent conversion result. 1 = Data in RESULT (EADC_ADnDATx[11:0]) is overwrite. If converted data in RESULT (EADC_ADnDATx[11:0]) has not been read before new conversion result is loaded to this register, OV (EADC_ADnDATx[16]) is set to 1. It is cleared by hardware after EADC_ADnDATx register is read. **Note:** n = 0~1 for ADC unit 0~1. **Note:** x = 0~7 for Sample 0~7.
[15:12]	**Reserved**	Reserved.
[11:0]	**RESULT**	**A/D Conversion Result** This field contains 12 bits conversion result.

A/D Control Register (EADC_CTL)

Register	Offset	R/W	Description	Reset Value
EADC_CTL	EADC_BA+0x40	R/W	A/D Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**ADCIEN3**	**ADCIEN2**	**ADCIEN1**	**ADCIEN0**	**ADCRST**	**ADCEN**

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**ADCIEN3**	**Specific SAMPLE MODULE A/D ADINT3 Interrupt Enable Bit** 0 = Specific SAMPLE MODULE A/D ADINT3 interrupt function Disabled. 1 = Specific SAMPLE MODULE A/D ADINT3 interrupt function Enabled. The A/D converter generates a conversion end ADIF3 (EADC_STATUS1 [3]) flag upon the end of specific SAMPLE MODULE A/D conversion. If ADCIEN3 (EADC_CTL [5]) bit is set then conversion end interrupt request ADINT3 is generated.
[4]	**ADCIEN2**	**Specific SAMPLE MODULE A/D ADINT2 Interrupt Enable Bit** 0 = Specific SAMPLE MODULE A/D ADINT2 interrupt function Disabled. 1 = Specific SAMPLE MODULE A/D ADINT2 interrupt function Enabled. The A/D converter generates a conversion end ADIF2 (EADC_STATUS1 [2]) flag upon the end of specific SAMPLE MODULE A/D conversion. If ADCIEN2 (EADC_CTL [4]) bit is set then conversion end interrupt request ADINT2 is generated.
[3]	**ADCIEN1**	**Specific SAMPLE MODULE A/D ADINT1 Interrupt Enable Bit** 0 = Specific SAMPLE MODULE A/D ADINT1 interrupt function Disabled. 1 = Specific SAMPLE MODULE A/D ADINT1 interrupt function Enabled. The A/D converter generates a conversion end ADIF0 (EADC_STATUS1 [1]) flag upon the end of specific SAMPLE MODULE A/D conversion. If ADCIEN1 EADC_CTL [3]) bit is set then conversion end interrupt request ADINT1 is generated.
[2]	**ADCIEN0**	**Specific SAMPLE MODULE A/D ADINT0 Interrupt Enable Bit** 0 = Specific SAMPLE MODULE A/D ADINT0 interrupt function Disabled. 1 = Specific SAMPLE MODULE A/D ADINT0 interrupt function Enabled. The A/D converter generates a conversion end ADIF0 (EADC_STATUS1 [0]) flag upon the end of specific SAMPLE MODULE A/D conversion. If ADCIEN0 (EADC_CTL [2]) bit is set then conversion end interrupt request ADINT0 is generated.

Bits	Description	
[1]	**ADCRST**	**ADC0, ADC1 A/D Converter Control Circuits Reset** 0 = No effect. 1 = Cause ADC control circuits reset to initial state, but not change the ADC registers value. The ADCRST (EADC_CTL [1]) bit remains 1 during ADC reset, when ADC reset end, the ADCRST (EADC_CTL [1]) bit is automatically cleared to 0.
[0]	**ADCEN**	**A/D Converter Enable Bit** 0 = Disabled. 1 = Enabled. Before starting A/D conversion function, this bit should be set to 1. Clear it to 0 to disable A/D converter analog circuit power consumption.

A/D SAMPLE MODULE Software Start Register (EADC_SWTRG)

Register	Offset	R/W	Description	Reset Value
EADC_SWTRG	EADC_BA+0x48	W	A/D SAMPLE MODULE Software Start Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SWTRG[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SWTRG[7:0]							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:8]	**SWTRG[15:8]**	**A/D SAMPLE17~SAMPLE10 Software Force To Start ADC Conversion** 0 = No effect. 1 = Start an ADC conversion when the priority is given to SAMPLE1x. **Note:** x = 0~7.
[7:0]	**SWTRG[7:0]**	**A/D SAMPLE07~SAMPLE00 Software Force To Start ADC Conversion** 0 = No effect. 1 = Start an ADC conversion when the priority is given to SAMPLE0x. **Note:** x = 0~7.

A/D SAMPLE MODULE Start of Conversion Pending Flag Register (EADC_PENDSTS)

Register	Offset	R/W	Description	Reset Value
EADC_PENDSTS	EADC_BA+0x4C	R	A/D Start of Conversion Pending Flag Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
STPF[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
STPF[7:0]							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:8]	**STPF[15:8]**	**A/D SAMPLE17~SAMPLE10 Start Of Conversion Pending Flag** 0 = There is no pending conversion for SAMPLE1x. 1 = SAMPLE1x ADC start of conversion is pending. **Note:** This bit remains 1 during pending state, when the respective ADC conversion is started, the STPF bit is automatically cleared to 0. **Note:** x = 0~7.
[7:0]	**STPF[7:0]**	**A/D SAMPLE07~SAMPLE00 Start Of Conversion Pending Flag** 0 = There is no pending conversion for SAMPLE0x. 1 = SAMPLE0x ADC start of conversion is pending. **Note:** This bit remains 1 during pending state, when the respective ADC conversion is started, the STPF bit is automatically cleared to 0. **Note:** x = 0~7.

A/D Interrupt Flag Overrun Register (EADC_ADIFOV)

Register	Offset	R/W	Description	Reset Value
EADC_ADIFOV	EADC_BA+0x50	R/W	A/D ADINT3~0 Interrupt Flag Overrun Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**ADFOV[3:0]**			

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3]	**ADFOV3**	**A/D ADINT3 Interrupt Flag Overrun** 0 = ADINT3 interrupt flag is not overwritten to 1. 1 = ADINT3 interrupt flag is overwritten to 1. **Note:** This bit is cleared by writing 1 to it.
[2]	**ADFOV2**	**A/D ADINT2 Interrupt Flag Overrun** 0 = ADINT2 interrupt flag is not overwritten to 1. 1 = ADINT2 interrupt flag is overwritten to 1. **Note:** This bit is cleared by writing 1 to it.
[1]	**ADFOV1**	**A/D ADINT1 Interrupt Flag Overrun** 0 = ADINT1 interrupt flag is not overwritten to 1. 1 = ADINT1 interrupt flag is overwritten to 1. **Note:** This bit is cleared by writing 1 to it.
[0]	**ADFOV0**	**A/D ADINT0 Interrupt Flag Overrun** 0 = ADINT0 interrupt flag is not overwritten to 1. 1 = ADINT0 interrupt flag is overwritten to 1. **Note:** This bit is cleared by writing 1 to it.

A/D SAMPLE MODULE Overrun Flag Register (EADC_OVSTS)

Register	Offset	R/W	Description	Reset Value
EADC_OVSTS	EADC_BA+0x54	R/W	A/D SAMPLE MODULE Start of Conversion Overrun Flag Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SPOVF[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SPOVF[7:0]							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:8]	**SPOVF[15:8]**	**A/D SAMPLE17~SAMPLE10 Start Of Conversion Overrun Flag** 0 = No SAMPLE1x event overrun. 1 = Indicates a new SAMPLE1x event is generated while an old one event is pending. If there is a new trigger event comes when the SAMPLE MODULE is pending for the last trigger event, the overrun is happened and the SPOVF bit will be set as 1. **Note:** This bit is cleared by writing 1 to it. **Note:** x = 0~7.
[7:0]	**SPOVF[7:0]**	**A/D SAMPLE07~SAMPLE00 Start Of Conversion Overrun Flag** 0 = No SAMPLE0x event overrun. 1 = Indicates a new SAMPLE0x event is generated while an old one event is pending. If there is a new trigger event comes when the SAMPLE MODULE is pending for the last trigger event, the overrun is happened and the SPOVF bit will be set as 1. **Note:** This bit is cleared by writing 1 to it. **Note:** x = 0~7.

A/D SAMPLEn0~SAMPLEn3 Control Registers (EADC_AD0SPCTL0 ~ EADC_AD0SPCTL3, EADC_AD1SPCTL0 ~ EADC_AD1SPCTL3)

Register	Offset	R/W	Description	Reset Value
EADC_AD0SPCTL0	EADC_BA+0x58	R/W	A/D SAMPLE00 Control Register	0x0000_0000
EADC_AD0SPCTL1	EADC_BA+0x5C	R/W	A/D SAMPLE01 Control Register	0x0000_0000
EADC_AD0SPCTL2	EADC_BA+0x60	R/W	A/D SAMPLE02 Control Register	0x0000_0000
EADC_AD0SPCTL3	EADC_BA+0x64	R/W	A/D SAMPLE03 Control Register	0x0000_0000
EADC_AD1SPCTL0	EADC_BA+0x78	R/W	A/D SAMPLE10 Control Register	0x0000_0000
EADC_AD1SPCTL1	EADC_BA+0x7C	R/W	A/D SAMPLE11 Control Register	0x0000_0000
EADC_AD1SPCTL2	EADC_BA+0x80	R/W	A/D SAMPLE12 Control Register	0x0000_0000
EADC_AD1SPCTL3	EADC_BA+0x84	R/W	A/D SAMPLE13 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		EXTFEN	EXTREN	Reserved		TRGDLYDIV[1:0]	
15	14	13	12	11	10	9	8
TRGDLYCNT[7:0]							
7	6	5	4	3	2	1	0
TRGSEL[3:0]				CHSEL[3:0]			

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21]	**EXTFEN**	**A/D External Pin Falling Edge Trigger Enable Bit** 0 = A/D external pin falling edge trigger Disabled. 1 = A/D external pin falling edge trigger Enabled.
[20]	**EXTREN**	**A/D External Pin Rising Edge Trigger Enable Bit** 0 = A/D external pin rising edge trigger Disabled. 1 = A/D external pin rising edge trigger Enabled.
[19:18]	**Reserved**	Reserved.

Bits	Description	
[17:16]	**TRGDLYDIV[1:0]**	**A/D SAMPLE MODULE Start Of Conversion Trigger Delay Clock Divider Selection** Trigger delay clock frequency: 00 = ADC_CLK/1. 01 = ADC_CLK/2. 10 = ADC_CLK/4. 11 = ADC_CLK/16.
[15:8]	**TRGDLYCNT[7:0]**	**A/D SAMPLE MODULE Start Of Conversion Trigger Delay Time** Trigger delay time = (TRGDLYCNT + 4) x Trigger delay clock period.
[7:4]	**TRGSEL**	**A/D SAMPLE MODULE Start Of Conversion Trigger Source Selection** 0000 = Disable hardware trigger. 0001 = External pin (STADC) trigger. 0010 = ADC ADINT0 interrupt EOC pulse trigger. 0011 = ADC ADINT1 interrupt EOC pulse trigger. 0100 = Timer0 overflow pulse trigger. 0101 = Timer1 overflow pulse trigger. 0110 = Timer2 overflow pulse trigger. 0111 = Timer3 overflow pulse trigger. 1000 = EPWM0_CH0 trigger. 1001 = EPWM0_CH2 trigger. 1010 = EPWM0_CH4 trigger. 1011 = EPWM1_CH0 trigger. 1100 = EPWM1_CH2 trigger. 1101 = EPWM1_CH4 trigger. 1110 = PWM0_CH0 trigger. 1111 = PWM0_CH1 trigger.
[3:0]	**CHSEL**	**A/D SAMPLE MODULE 0,1 Channel Selection** 0000 = ADCn_CH0. 0001 = ADCn_CH1. 0010 = ADCn_CH2. 0011 = ADCn_CH3. 0100 = ADCn_CH4. 0101 = ADCn_CH5. 0110 = ADCn_CH6. 0111 = ADCn_CH7. For SAMPLE MODULE0 1000 = VBG. 1001 = VTEMP. 1010 = AVSS. 1011 = OP0. For SAMPLE MODULE1 1000= OP1.

A/D SAMPLEn4~7 Control Registers (EADC_AD0SPCTL4 ~ EADC_AD0SPCTL7, EADC_AD1SPCTL4 ~ EADC_AD1SPCTL7)

Register	Offset	R/W	Description	Reset Value
EADC_AD0SPCTL4	EADC_BA+0x68	R/W	A/D SAMPLE04 Control Register	0x0000_0000
EADC_AD0SPCTL5	EADC_BA+0x6C	R/W	A/D SAMPLE05 Control Register	0x0000_0000
EADC_AD0SPCTL6	EADC_BA+0x70	R/W	A/D SAMPLE06 Control Register	0x0000_0000
EADC_AD0SPCTL7	EADC_BA+0x74	R/W	A/D SAMPLE07 Control Register	0x0000_0000
EADC_AD1SPCTL4	EADC_BA+0x88	R/W	A/D SAMPLE14 Control Register	0x0000_0000
EADC_AD1SPCTL5	EADC_BA+0x8C	R/W	A/D SAMPLE15 Control Register	0x0000_0000
EADC_AD1SPCTL6	EADC_BA+0x90	R/W	A/D SAMPLE16 Control Register	0x0000_0000
EADC_AD1SPCTL7	EADC_BA+0x94	R/W	A/D SAMPLE17 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		EXTFEN	EXTREN	Reserved			
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	TRGSEL[2:0]			CHSEL[3:0]			

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21]	**EXTFEN**	**A/D External Pin Falling Edge Trigger Enable Bit** 0 = A/D external pin falling edge trigger Disabled. 1 = A/D external pin falling edge trigger Enabled.
[20]	**EXTREN**	**A/D External Pin Rising Edge Trigger Enable Bit** 0 = A/D external pin rising edge trigger Disabled. 1 = A/D external pin rising edge trigger Enabled.
[19:7]	**Reserved**	Reserved.

Bits	Description	
[6:4]	**TRGSEL**	**A/D SAMPLE MODULE Start Of Conversion Trigger Source Selection** 000 = Disable hardware trigger. 001 = External pin (STADC) trigger. 010 = ADC ADINT0 interrupt EOC pulse trigger. 011 = ADC ADINT1 interrupt EOC pulse trigger. 100 = Timer0 overflow pulse trigger. 101 = Timer1 overflow pulse trigger. 110 = Timer2 overflow pulse trigger. 111 = Timer3 overflow pulse trigger.
[3:0]	**CHSEL**	**A/D SAMPLE MODULE 0,1 Channel Selection** 0000 = ADCn_CH0. 0001 = ADCn_CH1. 0010 = ADCn_CH2. 0011 = ADCn_CH3. 0100 = ADCn_CH4. 0101 = ADCn_CH5. 0110 = ADCn_CH6. 0111 = ADCn_CH7. For SAMPLE MODULE 0 1000 = VBG. 1001 = VTEMP. 1010 = AVSS. 1011 = OP0. For SAMPLE MODULE 1 1000= OP1.

A/D Simultaneous Sampling Mode Select Register (EADC_SIMUSEL)

Register	Offset	R/W	Description	Reset Value
EADC_SIMUSEL	EADC_BA+0xA4	R/W	A/D SAMPLE MODULE Simultaneous Sampling Mode Select Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SIMUSEL7	**SIMUSEL6**	**SIMUSEL5**	**SIMUSEL4**	**SIMUSEL3**	**SIMUSEL2**	**SIMUSEL1**	**SIMUSEL0**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**SIMUSEL7**	**A/D SAMPLE07, SAMPLE17 Simultaneous Sampling Mode Selection** 0 = SAMPLE07, SAMPLE17 are in single sampling mode, both SAMPLE07 and SAMPLE17's 3 bits of CHSEL define the ADC channels to be converted. 1 = SAMPLE07, SAMPLE17 are in simultaneous sampling mode, Only SAMPLE07 can trigger the both ADC conversions of SAMPLE07 and SAMPLE17, SAMPLE17 trigger select TRGSEL is ignored. If SAMPLE07's CHSEL = 1, SAMPLE17's CHSEL = 3, the pair of channels are ADC0_CH1, ADC1_CH3, they will do the ADC conversion at the same time to reach the simulataneous sampling goal.
[6]	**SIMUSEL6**	**A/D SAMPLE06, SAMPLE16 Simultaneous Sampling Mode Selection** 0 = SAMPLE06, SAMPLE16 are in single sampling mode, both SAMPLE06 and SAMPLE16's 3 bits of CHSEL define the ADC channels to be converted. 1 = SAMPLE06, SAMPLE16 are in simultaneous sampling mode, Only SAMPLE06 can trigger the both ADC conversions of SAMPLE06 and SAMPLE16, SAMPLE16 trigger select TRGSEL is ignored. If SAMPLE06's CHSEL = 1, and SAMPLE16's CHSEL = 3, the pair of channels are ADC0_CH1, ADC1_CH3, they will do the ADC conversion at the same time to reach the simulataneous sampling goal.
[5]	**SIMUSEL5**	**A/D SAMPLE05, SAMPLE15 Simultaneous Sampling Mode Selection** 0 = SAMPLE05, SAMPLE15 are in single sampling mode, both SAMPLE05 and SAMPLE15's 3 bits of CHSEL define the ADC channels to be converted. 1 = SAMPLE05, SAMPLE15 are in simultaneous sampling mode, Only SAMPLE05 can trigger the both ADC conversions of SAMPLE05 and SAMPLE15, SAMPLE15 trigger select TRGSEL is ignored. if SAMPLE05's CHSEL = 1, and SAMPLE15's CHSEL = 3, the pair of channels are ADC0_CH1, ADC1_CH3, they will do the ADC conversion at the same time to reach the simulataneous sampling goal.

Bits	Description	
[4]	**SIMUSEL4**	**A/D SAMPLE04, SAMPLE14 Simultaneous Sampling Mode Selection** 0 = SAMPLE04, SAMPLE14 are in single sampling mode, both SAMPLE04 and SAMPLE14's 3 bits of CHSEL define the ADC channels to be converted. 1 = SAMPLE04, SAMPLE14 are in simultaneous sampling mode, Only SAMPLE04 can trigger the both ADC conversions of SAMPLE04 and SAMPLE14, SAMPLE14 trigger select TRGSEL is ignored. If SAMPLE04's CHSEL = 1, and SAMPLE14's CHSEL = 3, the pair of channels are ADC0_CH1, ADC1_CH3, they will do the ADC conversion at the same time to reach the simulataneous sampling goal.
[3]	**SIMUSEL3**	**A/D SAMPLE03, SAMPLE13 Simultaneous Sampling Mode Selection** 0 = SAMPLE03, SAMPLE13 are in single sampling mode, both SAMPLE03 and SAMPLE13's 3 bits of CHSEL define the ADC channels to be converted. 1 = SAMPLE03, SAMPLE13 are in simultaneous sampling mode, Only SAMPLE03 can trigger the both ADC conversions of SAMPLE03 and SAMPLE13, SAMPLE13 trigger select TRGSEL is ignored. If SAMPLE03's CHSEL = 1, and SAMPLE13's CHSEL = 3, the pair of channels are ADC0_CH1, ADC1_CH3, they will do the ADC conversion at the same time to reach the simulataneous sampling goal.
[2]	**SIMUSEL2**	**A/D SAMPLE02, SAMPLE12 Simultaneous Sampling Mode Selection** 0 = SAMPLE02, SAMPLE12 are in single sampling mode, both SAMPLE02 and SAMPLE12's 3 bits of CHSEL define the ADC channels to be converted. 1 = SAMPLE02, SAMPLE12 are in simultaneous sampling mode, Only SAMPLE02 can trigger the both ADC conversions of SAMPLE02 and SAMPLE12, SAMPLE12 trigger select TRGSEL is ignored. If SAMPLE02's CHSEL = 1, and SAMPLE12's CHSEL = 3, the pair of channels are ADC0_CH1, ADC1_CH3, they will do the ADC conversion at the same time to reach the simulataneous sampling goal.
[1]	**SIMUSEL1**	**A/D SAMPLE01, SAMPLE11 Simultaneous Sampling Mode Selection** 0 = SAMPLE01, SAMPLE11 are in single sampling mode, both SAMPLE01 and SAMPLE11's 3 bits of CHSEL define the ADC channels to be converted. 1 = SAMPLE01, SAMPLE11 are in simultaneous sampling mode, Only SAMPLE01 can trigger the both ADC conversions of SAMPLE01 and SAMPLE11, SAMPLE11 trigger select TRGSEL is ignored. If SAMPLE01's CHSEL = 1, and SAMPLE11's CHSEL = 3, the pair of channels are ADC0_CH1, ADC1_CH3, they will do the ADC conversion at the same time to reach the simulataneous sampling goal.
[0]	**SIMUSEL0**	**A/D SAMPLE00, SAMPLE10 Simultaneous Sampling Mode Selection** 0 = SAMPLE00, SAMPLE10 are in single sampling mode, both SAMPLE00 and SAMPLE10's 3 bits of CHSEL define the ADC channels to be converted. 1 = SAMPLE00, SAMPLE10 are in simultaneous sampling mode, Only SAMPLE00 can trigger the both ADC conversions of SAMPLE00 and SAMPLE10, SAMPLE10 trigger select TRGSEL is ignored. If SAMPLE00's CHSEL = 1, and SAMPLE10's CHSEL = 3, the pair of channels are ADC0_CH1, ADC1_CH3, they will do the ADC conversion at the same time to reach the simulataneous sampling goal.

A/D Result Compare Register 0/1 (EADC_CMP0/1)

Register	Offset	R/W	Description	Reset Value
EADC_CMP0	EADC_BA+0xA8	R/W	A/D Result Compare Register 0	0x0000_0000
EADC_CMP1	EADC_BA+0xAC	R/W	A/D Result Compare Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				CMPDAT[11:8]			
23	22	21	20	19	18	17	16
CMPDAT[7:0]							
15	14	13	12	11	10	9	8
Reserved				CMPMCNT			
7	6	5	4	3	2	1	0
Reserved		CMPSPL			CMPCOND	ADCMPIE	ADCMPEN

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:16]	**CMPDAT**	**Compared Data** The 12 bits data is used to compare with conversion result of specified SAMPLE. Software can use it to monitor the external analog input pin voltage transition without imposing a load on software.
[15:12]	**Reserved**	Reserved.
[11:8]	**CMPMCNT**	**Compare Match Count** When the specified A/D SAMPLE MODULE analog conversion result matches the compare condition defined by CMPCOND (EADC_CMPx[2]), the internal match counter will increase 1. When the internal counter reaches the value to CMPMCNT (EADC_CMPx[11:8] + 1, the ADCMPF (EADC_STATUS1 [7:6]) bit will be set.
[7:6]	**Reserved**	Reserved.
[5:3]	**CMPSPL**	**Compare SAMPLE MODULE Selection** 000 = SAMPLE00 conversion result EADC_AD0DAT0 is selected to be compared. 001 = SAMPLE01 conversion result EADC_AD0DAT1 is selected to be compared. 010 = SAMPLE02 conversion result EADC_AD0DAT2 is selected to be compared. 011 = SAMPLE03 conversion result EADC_AD0DAT3 is selected to be compared. 100 = SAMPLE10 conversion result EADC_AD1DAT0 is selected to be compared. 101 = SAMPLE11 conversion result EADC_AD1DAT1 is selected to be compared. 110 = SAMPLE12 conversion result EADC_AD1DAT2 is selected to be compared. 111 = SAMPLE13 conversion result EADC_AD1DAT3 is selected to be compared.

Bits	Description	
[2]	**CMPCOND**	**Compare Condition** 0= Set the compare condition as that when a 12-bit A/D conversion result is less than the 12-bit CMPDAT (EADC_CMPx[27:16]), the internal match counter will increase one. 1= Set the compare condition as that when a 12-bit A/D conversion result is greater or equal to the 12-bit CMPDAT (EADC_CMPx[27:16]), the internal match counter will increase one. **Note:** When the internal counter reaches the value to CMPMCNT (EADC_CMPx[11:8]) + 1, the CMPF bit will be set.
[1]	**ADCMPIE**	**A/D Result Compare Interrupt Enable Bit** 0 = Compare function interrupt Disabled. 1 = Compare function interrupt Enabled. If the compare function is enabled and the compare condition matches the setting of CMPCOND (EADC_CMPx[2]) and CMPMCNT (EADC_CMPx[11:8]), ADCMPF (EADC_STATUS1 [7:6]) bit will be asserted, in the meanwhile, if ADCMPIE (EADC_CMPx[1]) is set to 1, a compare interrupt request is generated.
[0]	**ADCMPEN**	**A/D Result Compare Enable Bit** 0 = Compare Disabled. 1 = Compare Enabled. Set this bit to 1 to enable compare CMPDAT (EADC_CMPx[27:16]) with specified SAMPLE MODULE conversion result when converted data is loaded into ADDR register.

A/D Status Register 0 (EADC_STATUS0)

Register	Offset	R/W	Description	Reset Value
EADC_STATUS0	EADC_BA+0xB0	R	A/D Status Register 0	0x0000_0000

31	30	29	28	27	26	25	24
OV[15:8]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
OV[7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VALID[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VALID[7:0]							

Bits	Description	
[31:24]	**OV[15:8]**	**ADDR17~ADDR10 Overrun Flag (Read Only)** It is a mirror to OV bit in SAMPLE 1 A/D result data register EADC_AD0DAT1x. **Note:** x = 0~7.
[23:16]	**OV[7:0]**	**ADDR07~ ADDR00 Overrun Flag (Read Only)** It is a mirror to OV bit in SAMPLE0 A/D result data register EADC_AD0DAT0x. **Note:** x = 0~7.
[15:8]	**VALID[15:8]**	**ADDR17~ ADDR10 Data Valid Flag (Read Only)** It is a mirror of VALID bit in SAMPLE1 A/D result data register EADC_AD0DAT1x. **Note:** x = 0~7.
[7:0]	**VALID[7:0]**	**ADDR07~ ADDR00 Data Valid Flag (Read Only)** It is a mirror of VALID bit in SAMPLE0 A/D result data register EADC_AD0DAT0x. **Note:** x = 0~7.

A/D Status Register 1 (EADC_STATUS1)

Register	Offset	R/W	Description	Reset Value
EADC_STATUS1	EADC_BA+0xB4	R/W	A/D Status Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				AOV	AVALID	STOVF	ADOVIF
23	22	21	20	19	18	17	16
CHANNEL1				Reserved			BUSY1
15	14	13	12	11	10	9	8
CHANNEL0				Reserved			BUSY0
7	6	5	4	3	2	1	0
ADCMPF1	ADCMPF0	ADCMPO1	ADCMPO0	ADIF3	ADIF2	ADIF1	ADIF0

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27]	**AOV**	**For All SAMPLE MODULE A/D Result Data Register Overrun Flags Check** 0 = None of SAMPLE MODULE data register overrun flag OVx (EADC_ADnDATx[16]) is set to 1. 1 = Any one of SAMPLE MODULE data register overrun flag OVx (EADC_ADnDATx[16]) is set to 1. **Note:** This bit will keep 1 when any OVx (EADC_ADnDATx[16]) Flag is equal to 1.
[26]	**AVALID**	**For All SAMPLE MODULE A/D Result Data Register ADDR Data Valid Flag Check** 0 = None of SAMPLE MODULE data register valid flag VALIDx (EADC_ADnDATx[17]) is set to 1. 1 = Any one of SAMPLE MODULE data register valid flag VALIDx (EADC_ADnDATx[17]) is set to 1. **Note:** This bit will keep 1 when any VALIDx (EADC_ADnDATx[17]) Flag is equal to 1.
[25]	**STOVF**	**For All A/D SAMPLE MODULE Start Of Conversion Overrun Flags Check** 0 = None of SAMPLE MODULE event overrun flag SPOVFx (ADSPOVFR [15:0]) is set to 1. 1 = Any one of SAMPLE MODULE event overrun flag SPOVFx (ADSPOVFR [15:0]) is set to 1. **Note:** This bit will keep 1 when any SPOVFx (ADSPOVFR [15:0]) Flag is equal to 1.
[24]	**ADOVIF**	**All A/D Interrupt Flag Overrun Bits Check** 0 = None of ADINT interrupt flag ADFOVx (ADIFOVR [15:0]) is overwritten to 1. 1 = Any one of ADINT interrupt flag ADFOVx (ADIFOVR [15:0]) is overwritten to 1. **Note:** This bit will keep 1 when any ADFOVx (ADIFOVR [15:0]) Flag is equal to 1.

Bits	Description	
[23:20]	**CHANNEL1[3:0]**	**Current Conversion Channel (Read Only)** This filed reflects ADC1 current conversion channel when BUSY1 (EADC_STATUS1 [16]) = 1. When BUSY1 (EADC_STATUS1 [16]) = 0, it shows the last converted channel. 0000 = ADC1_CH0. 0001 = ADC1_CH1. 0010 = ADC1_CH2. 0011 = ADC1_CH3. 0100 = ADC1_CH4. 0101 = ADC1_CH5. 0110 = ADC1_CH6. 0111 = ADC1_CH7. 1000 = OPA1_O. Other = reversed.
[19:17]	**Reserved**	Reserved.
[16]	**BUSY1**	**Busy/Idle** 0 = A/D converter 1 (ADC1) is in idle state. 1 = A/D converter 1 (ADC1) is doing conversion.
[15:12]	**CHANNEL0[3:0]**	**Current Conversion Channel (Read Only)** This filed reflects ADC0 current conversion channel when BUSY0 (EADC_STATUS1 [8]) = 1. When BUSY0 (EADC_STATUS1 [8]) = 0, it shows the last converted channel. 0000 = ADC0_CH0. 0001 = ADC0_CH1. 0010 = ADC0_CH2. 0011 = ADC0_CH3. 0100 = ADC0_CH4. 0100 = ADC0_CH5. 0110 = ADC0_CH6. 0111 = ADC0_CH7. 1000 = VBG. 1001 = VTEMP. 1010 = AVSS. 1011 = OPA0_O. Other = reserved.
[11:9]	**Reserved**	Reserved.
[8]	**BUSY0**	**Busy/Idle (Read Only)** 0 = A/D converter 0 (ADC0) is in idle state. 1 = A/D converter 0 (ADC0) is doing conversion.
[7]	**ADCMPF1**	**ADC Compare 1 Flag** When the specific SAMPLE MODULE A/D conversion result meets setting condition in EADC_CMP1 then this bit is set to 1. 0 = Conversion result in ADDR does not meet EADC_CMP1 setting. 1 = Conversion result in ADDR meets EADC_CMP1 setting. **Note:** This bit is cleared by writing 1 to it.

Bits	Description	
[6]	**ADCMPF0**	**ADC Compare 0 Flag** When the specific SAMPLE MODULE A/D conversion result meets setting condition in EADC_CMP0 then this bit is set to 1. 0 = Conversion result in ADDR does not meet EADC_CMP0 setting. 1 = Conversion result in ADDR meets EADC_CMP0 setting. **Note:** This bit is cleared by writing 1 to it.
[5]	**ADCMPO1**	**ADC Compare 1 Output Status** The 12 bits compare1 data CMPDAT (EADC_CMP1 [27:16]) is used to compare with conversion result of specified SAMPLE MODULE. Software can use it to monitor the external analog input pin voltage status. 0 = Conversion result in ADDR less than CMPDAT EADC_CMP1 [27:16]) setting. 1 = Conversion result in ADDR great than or equal CMPDAT (EADC_CMP1 [27:16]) setting.
[4]	**ADCMPO0**	**ADC Compare 0 Output Status** The 12 bits compare0 data CMPDAT EADC_CMP0 [27:16]) is used to compare with conversion result of specified SAMPLE MODULE. Software can use it to monitor the external analog input pin voltage status. 0 = Conversion result in ADDR less than CMPDAT (EADC_CMP0 [27:16]) setting. 1 = Conversion result in ADDR great than or equal CMPDAT (EADC_CMP0 [27:16]) setting.
[3]	**ADIF3**	**A/D ADINT3 Interrupt Flag** 0 = No ADINT3 interrupt pulse received. 1 = ADINT3 interrupt pulse has been received. **Note1:** This bit is cleared by writing 1 to it. **Note2:**This bit indicates whether an A/D conversion of specific SAMPLE MODULE has been completed
[2]	**ADIF2**	**A/D ADINT2 Interrupt Flag** 0 = no ADINT2 interrupt pulse received. 1 = ADINT2 interrupt pulse has been received. **Note1:** This bit is cleared by writing 1 to it. **Note2:**This bit indicates whether an A/D conversion of specific SAMPLE MODULE has been completed
[1]	**ADIF1**	**A/D ADINT1 Interrupt Flag** 0 = No ADINT1 interrupt pulse received. 1 = ADINT1 interrupt pulse has been received. **Note1:** This bit is cleared by writing 1 to it. **Note2:**This bit indicates whether an A/D conversion of specific SAMPLE MODULE has been completed
[0]	**ADIF0**	**A/D ADINT0 Interrupt Flag** 0 = No ADINT0 interrupt pulse received. 1 = ADINT0 interrupt pulse has been received. **Note1:** This bit is cleared by writing 1 to it. **Note2:**This bit indicates whether an A/D conversion of specific SAMPLE MODULE has been completed

A/D Timing Control Register (EADC_EXTSMPT)

Register	Offset	R/W	Description	Reset Value
EADC_EXTSMPT	EADC_BA+0xB8	R/W	A/D Timing Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
EXTSMPT1							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
EXTSMPT0							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**EXTSMPT1**	**ADC1 Extend Sampling Time** When A/D converting at high conversion rate, the sampling time of analog input voltage may not enough if input channel loading is heavy, SW can extend A/D sampling time after trigger source is coming to get enough sampling time. The range of start delay time is from 0~255 ADC clock.
[15:8]	**Reserved**	Reserved.
[7:0]	**EXTSMPT0**	**ADC0 Extend Sampling Time** When A/D converting at high conversion rate, the sampling time of analog input voltage may not enough if input channel loading is heavy, SW can extend A/D sampling time after trigger source is coming to get enough sampling time. The range of start delay time is from 0~255 ADC clock.

A/D Double Data Registers for A/D Data Registers (EADC_AD0DAT0~EADC_AD0DAT3, EADC_AD1DAT0~EADC_AD1DAT3)

Register	Offset	R/W	Description	Reset Value
EADC_AD0DDAT0	EADC_BA+0x100	R	A/D double Data Register 0 for SAMPLE00	0x0000_0000
EADC_AD0DDAT1	EADC_BA+0x104	R	A/D double Data Register 1 for SAMPLE01	0x0000_0000
EADC_AD0DDAT2	EADC_BA+0x108	R	A/D double Data Register 2 for SAMPLE02	0x0000_0000
EADC_AD0DDAT3	EADC_BA+0x10C	R	A/D double Data Register 3 for SAMPLE03	0x0000_0000
EADC_AD1DDAT0	EADC_BA+0x120	R	A/D double Data Register 0 for SAMPLE10	0x0000_0000
EADC_AD1DDAT1	EADC_BA+0x124	R	A/D double Data Register 1 for SAMPLE11	0x0000_0000
EADC_AD1DDAT2	EADC_BA+0x128	R	A/D double Data Register 2 for SAMPLE12	0x0000_0000
EADC_AD1DDAT3	EADC_BA+0x12C	R	A/D double Data Register 3 for SAMPLE13	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							VALID
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				RESULT [11:8]			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESULT[7:0]							

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**VALID**	**Valid Flag** 0 = Double buffer data in RESULT (EADC_ADnDATx[11:0]) bits is not valid. 1 = Double buffer data in RESULT (EADC_ADnDATx[11:0]) bits is valid. This bit is set to 1 when corresponding SAMPLE MODULE channel analog input conversion is completed and cleared by hardware after EADC_ADnDATx register is read.
[15:12]	**Reserved**	Reserved.
[11:0]	**RESULT**	**A/D Conversion Result** This field contains 12 bits conversion result.

A/D Double Buffer Mode select (EADC_DBMEN)

Register	Offset	R/W	Description	Reset Value
EADC_DBMEN	EADC_BA+0x130	R/W	A/D Double Buffer Mode select	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**AD1DBM3**	**AD1DBM2**	**AD1DBM1**	**AD1DBM0**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**AD0DBM3**	**AD0DBM2**	**AD0DBM1**	**AD0DBM0**

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11]	**AD1DBM3**	**Double Buffer Mode For SAMPLE13** 0 = SAMPLE13 has one sample result register. (default) 1 =SAMPLE13 has two sample result registers.
[10]	**AD1DBM2**	**Double Buffer Mode For SAMPLE12** 0 = SAMPLE12 has one sample result register. (default).. 1 =SAMPLE12 has two sample result registers.
[9]	**AD1DBM1**	**Double Buffer Mode For SAMPLE11** 0 = SAMPLE11 has one sample result register. (default). 1 =SAMPLE11 has two sample result registers.
[8]	**AD1DBM0**	**Double Buffer Mode For SAMPLE10** 0 = SAMPLE10 has one sample result register. (default) 1 =SAMPLE10 has two sample result registers.
[7:4]	**Reserved**	Reserved.
[3]	**AD0DBM3**	**Double Buffer Mode For SAMPLE03** 0 = SAMPLE03 has one sample result register. (default). 1 =SAMPLE03 has two sample result registers.
[2]	**AD0DBM2**	**Double Buffer Mode For SAMPLE02** 0 = SAMPLE02 has one sample result register. (default).. 1 =SAMPLE02 has two sample result registers.
[1]	**AD0DBM1**	**Double Buffer Mode For SAMPLE01** 0 = SAMPLE01 has one sample result register. (default). 1 = SAMPLE01 has two sample result registers.

Bits	Description	
[0]	**AD0DBM0**	**Double Buffer Mode For SAMPLE00** 0 = SAMPLE00 has one sample result register. (default). 1 =SAMPLE00 has two sample result registers.

ADC interrupt Source Enable Control Register (EADC_INTSRC0 ~ EADC_INTSRC3)

Register	Offset	R/W	Description	Reset Value
EADC_INTSRC0	EADC_BA+0x134	R/W	A/D Interrupt 0 Source Enable Control Register	0x0000_0000
EADC_INTSRC1	EADC_BA+0x138	R/W	A/D Interrupt 1 Source Enable Control Register	0x0000_0000
EADC_INTSRC2	EADC_BA+0x13C	R/W	A/D Interrupt 2 Source Enable Control Register	0x0000_0000
EADC_INTSRC3	EADC_BA+0x140	R/W	A/D Interrupt 3 Source Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
AD1SPIE7	**AD1SPIE6**	**AD1SPIE5**	**AD1SPIE4**	**AD1SPIE3**	**AD1SPIE2**	**AD1SPIE1**	**AD1SPIE0**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AD0SPIE7	**AD0SPIE6**	**AD0SPIE5**	**AD0SPIE4**	**AD0SPIE3**	**AD0SPIE2**	**AD0SPIE1**	**AD0SPIE0**

Bits	Description	
[15]	**AD1SPIE7**	**SAMPLE17 Interrupt Mask Enable Bit** 0 = SAMPLE17 interrupt mask Disabled. 1 = SAMPLE17 interrupt mask Enabled.
[14]	**AD1SPIE6**	**SAMPLE16 Interrupt Mask Enable Bit** 0 = SAMPLE16 interrupt mask Disabled. 1 = SAMPLE16 interrupt mask Enabled.
[13]	**AD1SPIE5**	**SAMPLE15 Interrupt Mask Enable Bit** 0 = SAMPLE15 interrupt mask Disabled. 1 = SAMPLE15 interrupt mask Enabled.
[12]	**AD1SPIE4**	**SAMPLE14 Interrupt Mask Enable Bit** 0 = SAMPLE14 interrupt mask Disabled. 1 = SAMPLE14 interrupt mask Enabled.
[11]	**AD1SPIE3**	**SAMPLE13 Interrupt Mask Enable Bit** 0 = SAMPLE13 interrupt mask Disabled. 1 = SAMPLE13 interrupt mask Enabled.
[10]	**AD1SPIE2**	**SAMPLE12 Interrupt Mask Enable Bit** 0 = SAMPLE12 interrupt mask Disabled. 1 = SAMPLE12 interrupt mask Enabled.

Bits	Description	
[9]	**AD1SPIE1**	**SAMPLE11 Interrupt Mask Enable Bit** 0 = SAMPLE11 interrupt mask Disabled. 1 = SAMPLE11 interrupt mask Enabled.
[8]	**AD1SPIE0**	**SAMPLE10 Interrupt Mask Enable Bit** 0 = SAMPLE10 interrupt mask Disabled. 1 = SAMPLE10 interrupt mask Enabled.
[7]	**AD0SPIE7**	**SAMPLE07 Interrupt Mask Enable Bit** 0 = SAMPLE07 interrupt mask Disabled. 1 = SAMPLE07 interrupt mask Enabled.
[6]	**AD0SPIE6**	**SAMPLE06 Interrupt Mask Enable Bit** 0 = SAMPLE06 interrupt mask Disabled. 1 = SAMPLE06 interrupt mask Enabled.
[5]	**AD0SPIE5**	**SAMPLE05 Interrupt Mask Enable Bit** 0 = SAMPLE05 interrupt mask Disabled. 1 = SAMPLE05 interrupt mask Enabled.
[4]	**AD0SPIE4**	**SAMPLE04 Interrupt Mask Enable Bit** 0 = SAMPLE04 interrupt mask Disabled. 1 = SAMPLE04 interrupt mask Enabled.
[3]	**AD0SPIE3**	**SAMPLE03 Interrupt Mask Enable Bit** 0 = SAMPLE03 interrupt mask Disabled. 1 = SAMPLE03 interrupt mask Enabled.
[2]	**AD0SPIE2**	**SAMPLE02 Interrupt Mask Enable Bit** 0 = SAMPLE02 interrupt mask Disabled. 1 = SAMPLE02 interrupt mask Enabled.
[1]	**AD0SPIE1**	**SAMPLE01 Interrupt Mask Enable Bit** 0 = SAMPLE01 interrupt mask Disabled. 1 = SAMPLE01 interrupt mask Enabled.
[0]	**AD0SPIE0**	**SAMPLE00 Interrupt Mask Enable Bit** 0 = SAMPLE00 interrupt mask Disabled. 1 = SAMPLE00 interrupt mask Enabled.

SAMPLE MODULE Trigger Condition Enable Control Register (EADC_AD0TRGEN0 ~ EADC_AD0TRGEN3, EADC_AD1TRGEN0 ~ EADC_AD1TRGEN3)

Register	Offset	R/W	Description	Reset Value
EADC_AD0TRGEN0	EADC_BA+0x144	R/W	A/D trigger condition for SAMPLE00	0x0000_0000
EADC_AD0TRGEN1	EADC_BA+0x148	R/W	A/D trigger condition for SAMPLE01	0x0000_0000
EADC_AD0TRGEN2	EADC_BA+0x14C	R/W	A/D trigger condition for SAMPLE02	0x0000_0000
EADC_AD0TRGEN3	EADC_BA+0x150	R/W	A/D trigger condition for SAMPLE03	0x0000_0000
EADC_AD1TRGEN0	EADC_BA+0x154	R/W	A/D trigger condition for SAMPLE10	0x0000_0000
EADC_AD1TRGEN1	EADC_BA+0x158	R/W	A/D trigger condition for SAMPLE11	0x0000_0000
EADC_AD1TRGEN2	EADC_BA+0x15C	R/W	A/D trigger condition for SAMPLE12	0x0000_0000
EADC_AD1TRGEN3	EADC_BA+0x160	R/W	A/D trigger condition for SAMPLE13	0x0000_0000

31	30	29	28	27	26	25	24
PWM01CEN	PWM01PEN	PWM01FEN	PWM01REN	PWM00CEN	PWM00PEN	PWM00FEN	PWM00REN
23	22	21	20	19	18	17	16
EPWM14CEN	EPWM14PEN	EPWM14FEN	EPWM14REN	EPWM12CEN	EPWM12PEN	EPWM12FEN	EPWM12REN
15	14	13	12	11	10	9	8
EPWM10CEN	EPWM10PEN	EPWM10FEN	EPWM10REN	EPWM04CEN	EPWM04PEN	EPWM04FEN	EPWM04REN
7	6	5	4	3	2	1	0
EPWM02CEN	EPWM02PEN	EPWM02FEN	EPWM02REN	EPWM00CEN	EPWM00PEN	EPWM00FEN	EPWM00REN

Bits	Description	
[31]	**PWM01CEN**	**PWM0_CH1 Center Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[30]	**PWM01PEN**	**PWM0_CH1 Period Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[29]	**PWM01FEN**	**PWM0_CH1 Falling Edge Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[28]	**PWM01REN**	**PWM0_CH1 Rising Edge Trigger Enable Bit** 0 = Disabled. 1 = Enabled.

Bits	Description	
[27]	**PWM00CEN**	**PWM0_CH0 Center Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[26]	**PWM00PEN**	**PWM0_CH0 Period Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[25]	**PWM00FEN**	**PWM0_CH0 Falling Edge Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[24]	**PWM00REN**	**PWM0_CH0 Rising Edge Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[23]	**EPWM14CEN**	**EPWM1_CH4 Center Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[22]	**EPWM14PEN**	**EPWM1_CH4 Period Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[21]	**EPWM14FEN**	**EPWM1_CH4 Falling Edge Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[20]	**EPWM14REN**	**EPWM1_CH4 Rising Edge Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[19]	**EPWM12CEN**	**EPWM1_CH2 Center Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[18]	**EPWM12PEN**	**EPWM1_CH2 Period Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[17]	**EPWM120FEN**	**EPWM1_CH2 Falling Edge Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[16]	**EPWM12REN**	**EPWM1_CH2 Rising Edge Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[15]	**EPWM10CEN**	**EPWM1_CH0 Center Trigger Enable Bit** 0 = Disabled. 1 = Enabled.

Bits	Description	
[14]	**EPWM10PEN**	**EPWM1_CH0 Period Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[13]	**EPWM10FEN**	**EPWM1_CH0 Falling Edge Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[12]	**EPWM10REN**	**EPWM1_CH0 Rising Edge Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[11]	**EPWM04CEN**	**EPWM0_CH4 Center Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[10]	**EPWM04PEN**	**EPWM0_CH4 Period Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[9]	**EPWM04FEN**	**EPWM0_CH4 Falling Rdge Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[8]	**EPWM04REN**	**EPWM0_CH4 Rising Edge Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[7]	**EPWM02CEN**	**EPWM0_CH2 Center Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[6]	**EPWM02PEN**	**EPWM0_CH2 Period Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[5]	**EPWM02FEN**	**EPWM0_CH2 Falling Edge Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[4]	**EPWM02REN**	**EPWM0_CH2 Rising Edge Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[3]	**EPWM00CEN**	**EPWM0_CH0 Center Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[2]	**EPWM00PEN**	**EPWM0_CH0 Period Trigger Enable Bit** 0 = Disabled. 1 = Enabled.

Bits	Description	
[1]	**EPWM00FEN**	**EPWM0_CH0 Falling Edge Trigger Enable Bit** 0 = Disabled. 1 = Enabled.
[0]	**EPWM00REN**	**EPWM0_CH0 Rising Edge Trigger Enable Bit** 0 = Disabled. 1 = Enabled.

6.7 Controller Area Network (CAN)

6.7.1 Overview

The C_CAN consists of the CAN Core, Message RAM, Message Handler, Control Registers and Module Interface (Refer to *Figure 6.7-1*). The CAN Core performs communication according to the CAN protocol version 2.0 part A and B. The bit rate can be programmed to values up to 1MBit/s. For the connection to the physical layer, additional transceiver hardware is required.

For communication on a CAN network, individual Message Objects are configured. The Message Objects and Identifier Masks for acceptance filtering of received messages are stored in the Message RAM. All functions concerning the handling of messages are implemented in the Message Handler. These functions include acceptance filtering, the transfer of messages between the CAN Core and the Message RAM, and the handling of transmission requests as well as the generation of the module interrupt.

The register set of the C_CAN can be accessed directly by the software through the module interface. These registers are used to control/configure the CAN Core and the Message Handler and to access the Message RAM.

6.7.2 Features

- Supports CAN protocol version 2.0 part A and B
- Bit rates up to 1 MBit/s
- 32 Message Objects
- Each Message Object has its own identifier mask
- Programmable FIFO mode (concatenation of Message Objects)
- Maskable interrupt
- Disabled Automatic Re-transmission mode for Time Triggered CAN applications
- Programmable loop-back mode for self-test operation
- 16-bit module interfaces to the AMBA APB bus
- Supports wake-up function

6.7.3 Block Diagram

The C_CAN interfaces with the AMBA APB bus. The following figure shows the block diagram of the C_CAN.

- **CAN Core**

CAN Protocol Controller and Rx/Tx Shift Register for serial/parallel conversion of messages.

- **Message RAM**

Stores Message Objects and Identifier Masks

- **Registers**

All registers used to control and to configure the C_CAN.

- **Message Handler**

State Machine that controls the data transfer between the Rx/Tx Shift Register of the CAN Core and the Message RAM as well as the generation of interrupts as programmed in the Control and Configuration Registers.

- **Module Interface**

C_CAN interfaces to the AMBA APB 16-bit bus from ARM.

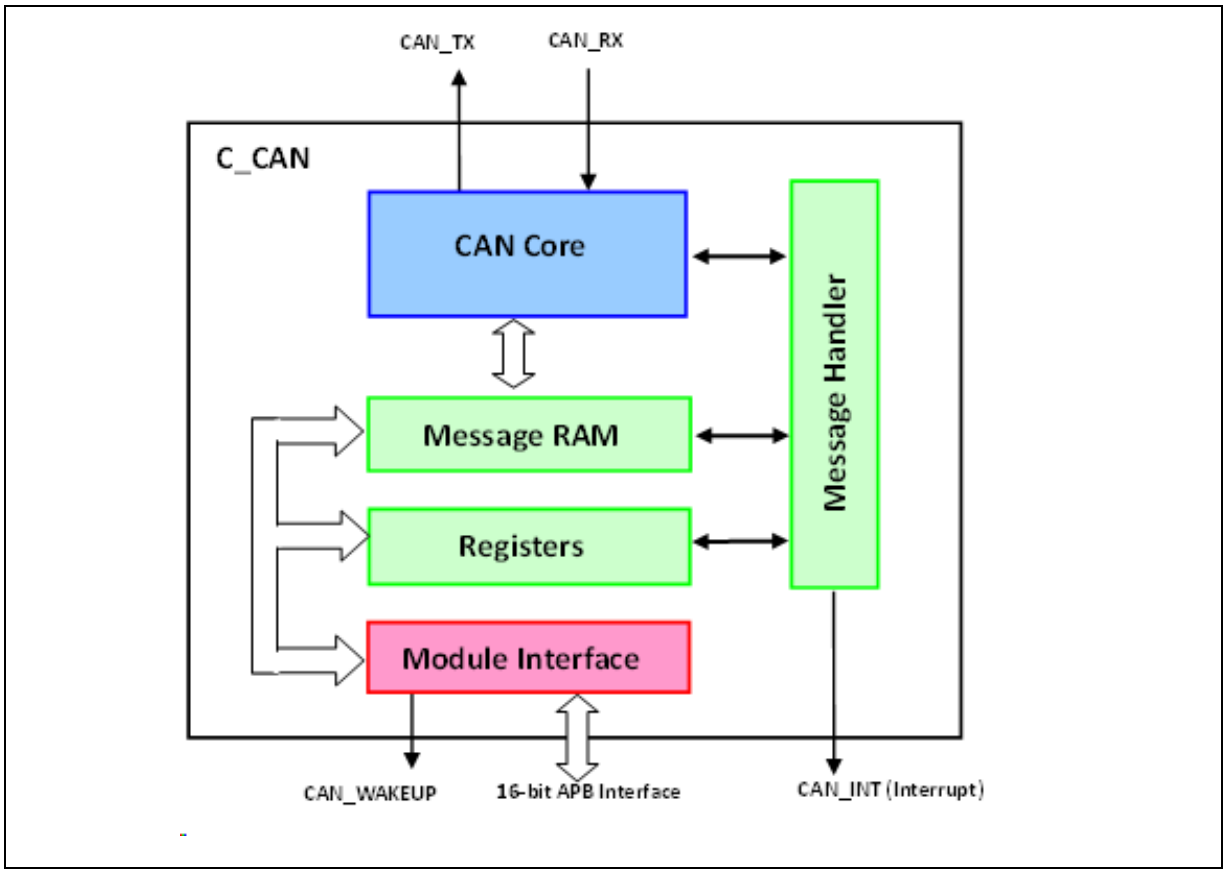


Figure 6.7-1 CAN Peripheral Block Diagram

6.7.4 Functional Description

6.7.4.1 Software Initialization

The software initialization is started by setting the **Init** bit in the CAN Control Register, either by a software or a hardware reset, or by going to *Bus_Off* state.

While the **Init** bit is set, all messages transfer to and from the CAN bus are stopped and the status of the **CAN_TX** output pin is recessive (HIGH). The Error Management Logic (EML) counters are unchanged. Setting the **Init** bit does not change any configuration register.

To initialize the CAN Controller, software has to set up the Bit Timing Register and each Message Object. If a Message Object is not required, the corresponding **MsgVal** bit should be cleared. Otherwise, the entire Message Object has to be initialized.

Access to the Bit Timing Register and to the Baud Rate Prescaler Extension Register for configuring bit timing is enabled when both the **Init** and **CCE** bits in the CAN Control Register are set.

Resetting the Init bit (by software only) finishes the software initialization. Later, the Bit Stream Processor (BSP) (see Section 5.13.6.10: Configuring the Bit Timing) synchronizes itself to the data transfer on the CAN bus by waiting for the occurrence of a sequence of 11 consecutive recessive bits (☐ Bus Idle) before it can take part in bus activities and start the message transfer.

The initialization of the Message Objects is independent of Init and can be done on the fly, but the Message Objects should all be configured to particular identifiers or set to not valid before the BSP starts the message transfer.

To change the configuration of a Message Object during normal operation, the software has to start by resetting the corresponding **MsgVal** bit. When the configuration is completed, **MsgVal** is set again.

6.7.4.2 CAN Message Transfer

Once the C_CAN is initialized and Init bit is reset to zero, the C_CAN Core synchronizes itself to the CAN bus and starts the message transfer.

Received messages are stored in their appropriate Message Objects if they pass the Message Handler's acceptance filtering. The whole message including all arbitration bits, DLC and eight data bytes are stored in the Message Object. If the Identifier Mask is used, the arbitration bits which are masked to "don't care" may be overwritten in the Message Object.

Software can read or write each message any time through the Interface Registers and the Message Handler guarantees data consistency in case of concurrent accesses.

Messages to be transmitted are updated by the application software. If a permanent Message Object (arbitration and control bits are set during configuration) exists for the message, only the data bytes are updated and the **TxRqst** bit with **NewDat** bit are set to start the transmission. If several transmit messages are assigned to the same Message Object (when the number of Message Objects is not sufficient), the whole Message Object has to be configured before the transmission of this message is requested.

The transmission of any number of Message Objects may be requested at the same time. Message objects are transmitted subsequently according to their internal priority. Messages may be updated or set to not valid any time, even when their requested transmission is still pending. The old data will be discarded when a message is updated before its pending transmission has started.

Depending on the configuration of the Message Object, the transmission of a message may be requested autonomously by the reception of a remote frame with a matching identifier.

6.7.4.3 *Disabled Automatic Retransmission*

In accordance with the CAN Specification (see ISO11898, 6.3.3 Recovery Management), the C_CAN provides means for automatic retransmission of frames that have lost arbitration or have been disturbed by errors during transmission. The frame transmission service will not be confirmed to the user before the transmission is successfully completed. This means that, by default, automatic retransmission is enabled. It can be disabled to enable the C_CAN to work within a Time Triggered CAN (TTCAN, see ISO11898-1) environment.

The Disabled Automatic Retransmission mode is enabled by setting the Disable Automatic Retransmission (**DAR**) bit in the CAN Control Register to one. In this operation mode, the programmer has to consider the different behavior of bits TxRqst and NewDat in the Control Registers of the Message Buffers:

- When a transmission starts, bit **TxRqst** of the respective Message Buffer is cleared, while bit **NewDat** remains set.
- When the transmission completed successfully, bit **NewDat** is cleared.
- When a transmission fails (lost arbitration or error), bit **NewDat** remains set.
- To restart the transmission, the software should set the bit **TxRqst** again.

6.7.5 Test Mode

Test Mode is entered by setting the Test bit in the CAN Control Register. In Test Mode, bits Tx1, Tx0, LBack, Silent and Basic in the Test Register are writeable. Bit Rx monitors the state of the CAN_RX pin and therefore is only readable. All Test Register functions are disabled when the Test bit is cleared.

6.7.5.1 *Silent Mode*

The CAN Core can be set in Silent Mode by programming the **Silent** bit in the Test Register to one. In Silent Mode, the C_CAN is able to receive valid data frames and valid remote frames, but it sends only recessive bits on the CAN bus and it cannot start a transmission. If the CAN Core is required to send a dominant bit (ACK bit, Error Frames), the bit is rerouted internally so that the CAN Core monitors this dominant bit, although the CAN bus may remain in recessive state. The Silent Mode can be used to analysis the traffic on a CAN bus without affecting it by the transmission of *dominant* bits. The following figure shows the connection of signals **CAN_TX** and **CAN_RX** to the CAN Core in Silent Mode.

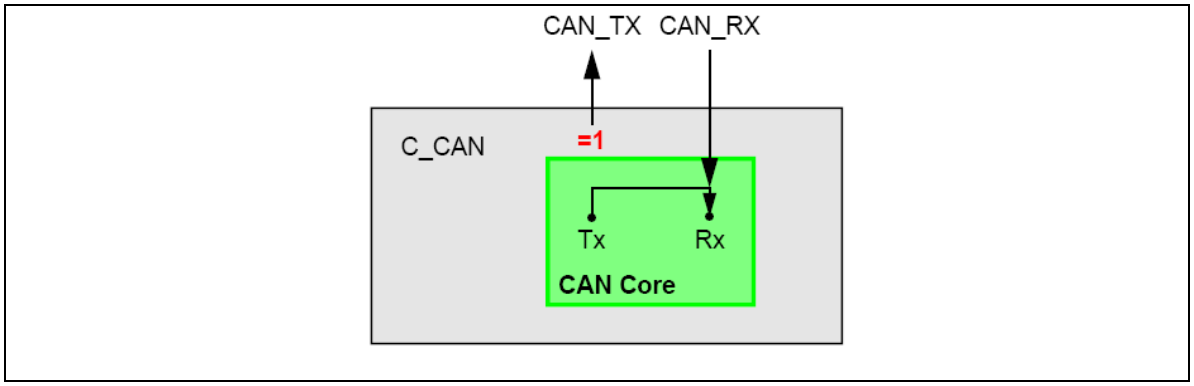


Figure 6.7-2 CAN Core in Silent Mode

6.7.5.2 Loop Back Mode

The CAN Core can be set in Loop Back Mode by programming the Test Register bit **LBack** to one. In Loop Back Mode, the CAN Core treats its own transmitted messages as received messages and stores them in a Receive Buffer (if they pass acceptance filtering).Figure 5-78 shows the connection of signals, **CAN_TX** and **CAN_RX**, to the CAN Core in Loop Back Mode.

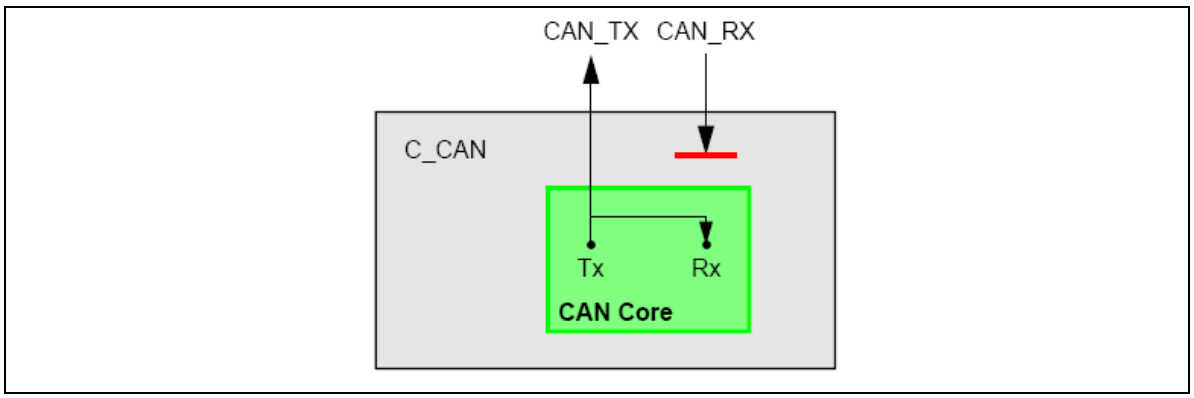


Figure 6.7-3 CAN Core in Loop Back Mode

This mode is provided for self-test functions. To be independent from external stimulation, the CAN Core ignores acknowledge errors (recessive bit sampled in the acknowledge slot of a data/ remote frame) in Loop Back Mode. In this mode, the CAN Core performs an internal feedback from its Tx output to its Rx input. The actual value of the **CAN_RX** input pin is disregarded by the CAN Core. The transmitted messages can be monitored on the **CAN_TX** pin.

6.7.5.3 Loop Back Combined with Silent Mode

It is also possible to combine Loop Back Mode and Silent Mode by programming bits **LBack** and **Silent** to one at the same time. This mode can be used for a “Hot Selftest”, which means that C_CAN can be tested without affecting a running CAN system connected to the **CAN_TX** and **CAN_RX** pins. In this mode, the **CAN_RX** pin is disconnected from the CAN Core and the **CAN_TX** pin is held recessive. The following figure shows the connection of signals **CAN_TX** and **CAN_RX** to the CAN Core in case of the combination of Loop Back Mode with Silent Mode.

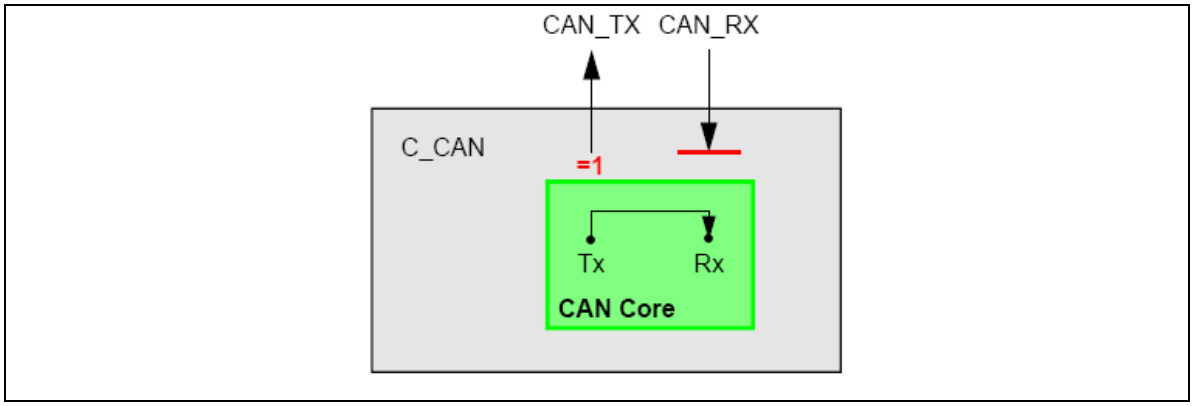


Figure 6.7-4 CAN Core in Loop Back Mode Combined with Silent Mode

6.7.5.4 Basic Mode

The CAN Core can be set in Basic Mode by programming the Test Register bit **Basic** to one. In this mode, the C_CAN runs without the Message RAM.

The IF1 Registers are used as Transmit Buffer. The transmission of the contents of the IF1 Registers is requested by writing the **Busy** bit of the IF1 Command Request Register to one. The IF1 Registers are locked while the **Busy** bit is set. The Busy bit indicates that the transmission is pending.

As soon the CAN bus is idle, the IF1 Registers are loaded into the shift register of the CAN Core and the transmission is started. When the transmission has been completed, the **Busy** bit is reset and the locked IF1 Registers are released.

A pending transmission can be aborted at any time by resetting the **Busy** bit in the IF1 Command Request Register while the IF1 Registers are locked. If the software has reset the **Busy** bit, a possible retransmission in case of lost arbitration or in case of an error is disabled.

The IF2 Registers are used as a Receive Buffer. After the reception of a message the contents of the shift register is stored into the IF2 Registers, without any acceptance filtering.

Additionally, the actual contents of the shift register can be monitored during the message transfer. Each time a read Message Object is initiated by writing the **Busy** bit of the IF2 Command Request Register to one, the contents of the shift register are stored into the IF2 Registers.

In Basic Mode, the evaluation of all Message Object related control and status bits and the control bits of the IFn Command Mask Registers are turned off. The message number of the Command request registers is not evaluated. The **NewDat** and **MsgLst** bits in the IF2 Message Control Register retain their function, **DLC3-0** indicates the received DLC, and the other control bits are read as '0'.

6.7.5.5 Software Control of CAN_TX Pin

Four output functions are available for the CAN transmit pin, **CAN_TX**. In addition to its default function (serial data output), the CAN transmit pin can drive the CAN Sample Point signal to monitor CAN_Core's bit timing and it can drive constant dominant or recessive values. The latter two functions, combined with the readable CAN receive pin **CAN_RX**, can be used to check the physical layer of the CAN bus.

The output mode for the **CAN_TX** pin is selected by programming the **Tx1** and **Tx0** bits of the CAN Test Register.

The three test functions of the **CAN_TX** pin interfere with all CAN protocol functions. **CAN_TX** must be left in its default function when CAN message transfer or any of the test modes (Loop Back Mode, Silent Mode, or Basic Mode) are selected.

6.7.6 CAN Communications

6.7.6.1 Managing Message Objects

The configuration of the Message Objects in the Message RAM (with the exception of the bits **MsgVal**, **NewDat**, **IntPnd**, and **TxRqst**) will not be affected by resetting the chip. All the Message Objects must be initialized by the application software or they must be "not valid" (**MsgVal** = '0') and the bit timing must be configured before the application software clears the Init bit in ter.

The configuration of a Message Object is done by programming Mask, Arbitration, Control and Data fields of one of the two interface registers to the desired values. By writing to the corresponding IFn Command Request Register, the IFn Message Buffer Registers are loaded into the addressed Message Object in the Message RAM.

When the Init bit in the CAN Control Register is cleared, the CAN Protocol Controller state machine of the CAN_Core and the state machine of the Message Handler control the internal

data flow of the C_CAN. Received messages that pass the acceptance filtering are stored into the Message RAM, messages with pending transmission request are loaded into the CAN_Core's Shift Register and are transmitted through the CAN bus.

The application software reads received messages and updates messages to be transmitted through the IFn Interface Registers. Depending on the configuration, the application software is interrupted on certain CAN message and CAN error events.

6.7.6.2 Message Handler State Machine

The Message Handler controls the data transfer between the Rx/Tx Shift Register of the CAN Core, the Message RAM and the IFn Registers.

The Message Handler FSM controls the following functions:

- Data Transfer from IFn Registers to the Message RAM
- Data Transfer from Message RAM to the IFn Registers
- Data Transfer from Shift Register to the Message RAM
- Data Transfer from Message RAM to Shift Register
- Data Transfer from Shift Register to the Acceptance Filtering unit
- Scanning of Message RAM for a matching Message Object
- Handling of TxRqst flags
- Handling of interrupts.

6.7.6.3 Data Transfer from/to Message RAM

When the application software initiates a data transfer between the IFn Registers and Message RAM, the Message Handler sets the Busy bit in the respective Command Request Register (CAN_IFn_CRR) to '1'. After the transfer has completed, the Busy bit is again cleared (see the following figure).

The respective Command Mask Register specifies whether a complete Message Object or only parts of it will be transferred. Due to the structure of the Message RAM, it is not possible to write single bits/bytes of one Message Object. It is always necessary to write a complete Message Object into the Message RAM. Therefore, the data transfer from the IFn Registers to the Message RAM requires a read-modify-write cycle. First, those parts of the Message Object that are not to be changed are read from the Message RAM and then the complete contents of the Message Buffer Registers are written into the Message Object.

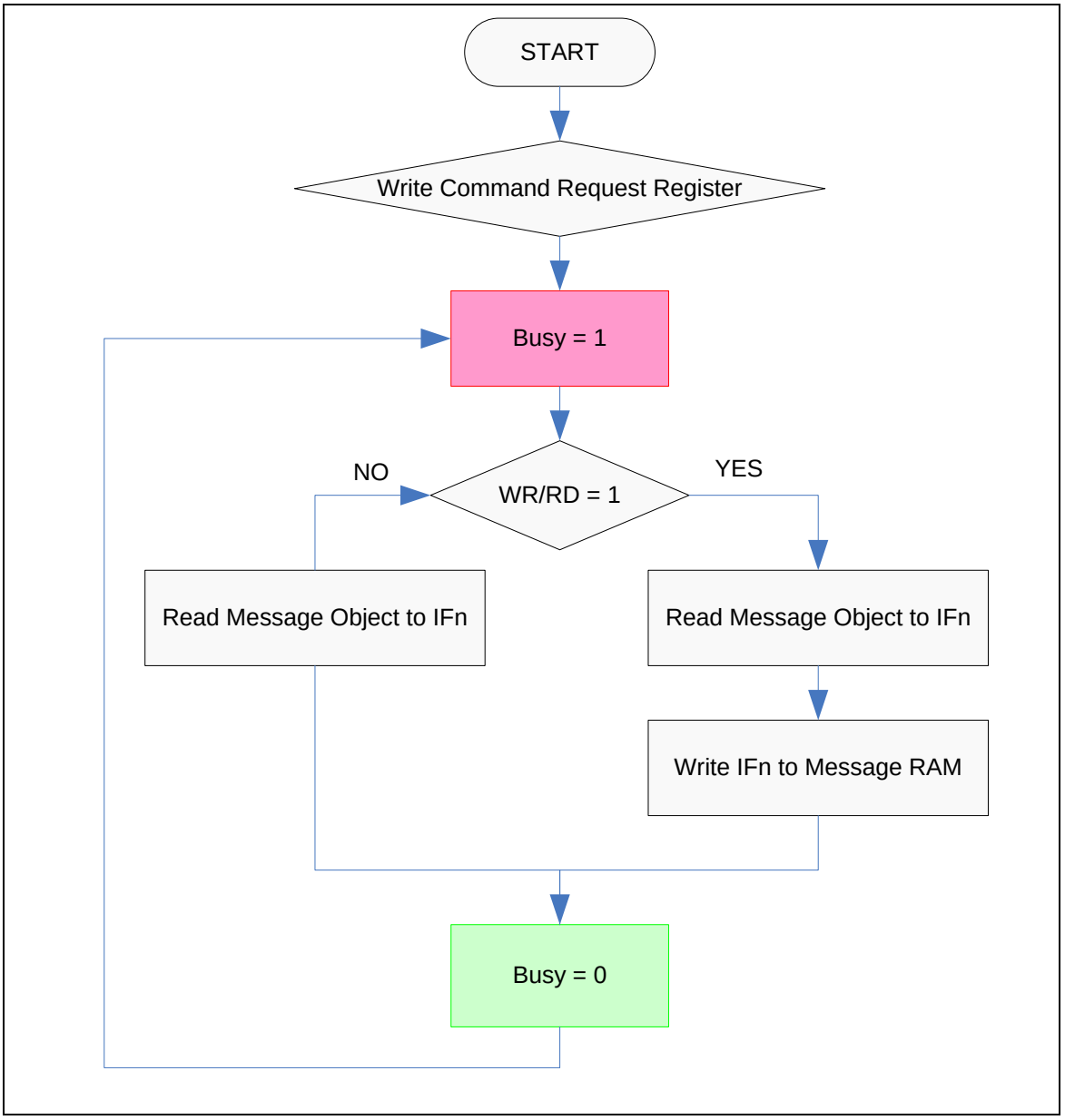


Figure 6.7-5 Data transfer between IF*n* Registers and Message

After a partial write of a Message Object, the Message Buffer Registers that are not selected in the Command Mask Register will set the actual contents of the selected Message Object.

After a partial read of a Message Object, the Message Buffer Registers that are not selected in the Command Mask Register will be left unchanged.

6.7.6.4 Message Transmission

If the shift register of the CAN Core cell is ready for loading and if there is no data transfer between the IFn Registers and Message RAM, the **MsgVal** bits in the Message Valid Register and **TxRqst** bits in the Transmission Request Register are evaluated. The valid Message Object with the highest priority pending transmission request is loaded into the shift register by the Message Handler and the transmission is started. The **NewDat** bit of the Message Object is reset.

After a successful transmission and also if no new data was written to the Message Object (**NewDat** = '0') since the start of the transmission, the **TxRqst** bit of the Message Control register (CAN_IFn_MCR) will be reset. If TxIE bit of the Message Control register (CAN_IFn_MCR) is set, **IntPnd** bit of the Interrupt Identifier register will be set after a successful transmission. If the C_CAN has lost the arbitration or if an error occurred during the transmission, the message will be retransmitted as soon as the CAN bus is free again. Meanwhile, if the transmission of a message with higher priority has been requested, the messages will be transmitted in the order of their priority.

6.7.6.5 Acceptance Filtering of Received Messages

When the arbitration and control field (Identifier + IDE + RTR + DLC) of an incoming message is completely shifted into the Rx/Tx Shift Register of the CAN Core, the Message Handler FSM starts the scanning of the Message RAM for a matching valid Message Object.

To scan the Message RAM for a matching Message Object, the Acceptance Filtering unit is loaded with the arbitration bits from the CAN Core shift register. The arbitration and mask fields (including **MsgVal**, **UMask**, **NewDat**, and **EoB**) of Message Object 1 are then loaded into the Acceptance Filtering unit and compared with the arbitration field from the shift register. This is repeated with each following Message Object until a matching Message Object is found or until the end of the Message RAM is reached.

If a match occurs, the scan is stopped and the Message Handler FSM proceeds depending on the type of frame (Data Frame or Remote Frame) received.

Reception of Data Frame

The Message Handler FSM stores the message from the CAN Core shift register into the respective Message Object in the Message RAM. Not only the data bytes, but all arbitration bits and the Data Length Code are stored into the corresponding Message Object. This is done to keep the data bytes connected with the identifier even if arbitration mask registers are used.

The **NewDat** bit is set to indicate that new data (not yet seen by the software) has been received. The application software should reset **NewDat** bit when the Message Object has been read. If at the time of reception, the **NewDat** bit was already set, **MsgLst** is set to indicate that the previous data (supposedly not seen by the software) is lost. If the **RxIE** bit is set, the **IntPnd** bit is set, causing the Interrupt Register to point to this Message Object.

The **TxRqst** bit of this Message Object is reset to prevent the transmission of a Remote Frame, while the requested Data Frame has just been received.

Reception of Remote Frame

When a Remote Frame is received, three different configurations of the matching Message Object have to be considered:

1) Dir = '1' (direction = transmit), **RmtEn** = '1', **UMask** = '1' or'0'

   At the reception of a matching Remote Frame, the TxRqst bit of this Message Object is set. The rest of the Message Object remains unchanged.

2) Dir = '1' (direction = transmit), **RmtEn** = '0', **UMask** ='0'

   At the reception of a matching Remote Frame, the TxRqst bit of this Message Object remains unchanged; the Remote Frame is ignored.

3) Dir = '1' (direction = transmit), **RmtEn** = '0', **UMask** ='1'

At the reception of a matching Remote Frame, the **TxRqst** bit of this Message Object is reset. The arbitration and control field (Identifier + IDE + RTR + DLC) from the shift register is stored in the Message Object of the Message RAM and the **NewDat** bit of this Message Object is set. The data field of the Message Object remains unchanged; the Remote Frame is treated similar to a received Data Frame.

6.7.6.6 Receive/Transmit Priority

The receive/transmit priority for the Message Objects is attached to the message number. Message Object 1 has the highest priority, while Message Object 32 has the lowest priority. If more than one transmission request is pending, they are serviced due to the priority of the corresponding Message Object

6.7.6.7 Configuring a Transmit Object

The following table shows how a Transmit Object should be initialized.

Ms	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	1	0	0	0	appl.	0	appl.	0

Table 6.7-1 Initialization of a Transmit Object

Note: appl. = application software.

The Arbitration Register values (**ID28-0** and **Xtd** bit) are provided by the application. They define the identifier and type of the outgoing message. If an 11-bit Identifier (“Standard Frame”) is used, it is programmed to ID28 - ID18. The ID17 - ID0 can then be disregarded.

If the **TxIE** bit is set, the **IntPnd** bit will be set after a successful transmission of the Message Object.

If the **RmtEn** bit is set, a matching received Remote Frame will cause the **TxRqst** bit to be set; the Remote Frame will autonomously be answered by a Data Frame.

The Data Register values (**DLC3-0**, **Data0-7**) are provided by the application, **TxRqst** and **RmtEn** may not be set before the data is valid.

The Mask Registers (**Msk28-0**, **UMask**, **MXtd**, and **MDir** bits) may be used (**UMask**=’1’) to allow groups of Remote Frames with similar identifiers to set the **TxRqst** bit. The Dir bit should not be masked.

6.7.6.8 Updating a Transmit Object

The software may update the data bytes of a Transmit Object any time through the IFn Interface registers, neither **MsgVal** nor **TxRqst** have to be reset before the update.

Even if only a part of the data bytes are to be updated, all four bytes of the corresponding IFn Data A Register or IFn Data B Register have to be valid before the contents of that register are transferred to the Message Object. Either the application software has to write all four bytes into the IFn Data Register or the Message Object is transferred to the IFn Data Register before the software writes the new data bytes.

When only the (eight) data bytes are updated, first 0x0087 is written to the Command Mask Register and then the number of the Message Object is written to the Command Request Register, concurrently updating the data bytes and setting **TxRqst**.

To prevent the reset of **TxRqst** at the end of a transmission that may already be in progress while the data is updated, **NewDat** has to be set together with **TxRqst**.

When **NewDat** is set together with **TxRqst**, **NewDat** will be reset as soon as the new transmission has started.

6.7.6.9 Configuring a Receive Object

The following table shows how a Receive Object should be initialized.

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	0	0	0	appl.	0	0	0	0

Table 6.7-2 Initialization of a Receive Object

The Arbitration Registers values **(ID28-0** and **Xtd** bit) are provided by the application. They define the identifier and type of accepted received messages. If an 11-bit Identifier (“Standard Frame”) is used, it is programmed to ID28 - ID18. Then ID17 - ID0 can be disregarded. When a Data Frame with an 11-bit Identifier is received, ID17 - ID0 will be set to ‘0’.

If the **RxIE** bit is set, the **IntPnd** bit will be set when a received Data Frame is accepted and stored in the Message Object.

The Data Length Code (DLC3-0) is provided by the application. When the Message Handler stores a Data Frame in the Message Object, it will store the received Data Length Code and eight data bytes. If the Data Length Code is less than 8, the remaining bytes of the Message Object will be overwritten by unspecified values.

The Mask Registers (Msk28-0, **UMask**, **MXtd**, and **MDir** bits) may be used (**UMask**=’1’) to allow groups of Data Frames with similar identifiers to be accepted. The Dir bit should not be masked in typical applications.

6.7.6.10 Handling Received Messages

The application software may read a received message any time through the IFn Interface registers. The data consistency is guaranteed by the Message Handler state machine.

Typically, the software will write first 0x007F to the Command Mask Register and then the number of the Message Object to the Command Request Register. This combination will transfer the whole received message from the Message RAM into the Message Buffer Register. Additionally, the bits **NewDat** and **IntPnd** are cleared in the Message RAM (not in the Message Buffer).

If the Message Object uses masks for acceptance filtering, the arbitration bits shows which of the matching messages have been received.

The actual value of **NewDat** shows whether a new message has been received since the last time this Message Object was read. The actual value of **MsgLst** shows whether more than one message has been received since the last time this Message Object was read. **MsgLst** will not be

automatically reset.

By means of a Remote Frame, the software may request another CAN node to provide new data for a receive object. Setting the **TxRqst** bit of a receive object will cause the transmission of a Remote Frame with the receive object's identifier. This Remote Frame triggers the other CAN node to start the transmission of the matching Data Frame. If the matching Data Frame is received before the Remote Frame could be transmitted, the **TxRqst** bit is automatically reset.

6.7.6.11 Configuring a FIFO Buffer

With the exception of the EoB bit, the configuration of Receive Objects belonging to a FIFO Buffer is the same as the configuration of a (single) Receive Object, see Section 5.13.6.5: Configuring a Receive Object.

To concatenate two or more Message Objects into a FIFO Buffer, the identifiers and masks (if used) of these Message Objects have to be programmed to matching values. Due to the implicit priority of the Message Objects, the Message Object with the lowest number will be the first Message Object of the FIFO Buffer. The **EoB** bit of all Message Objects of a FIFO Buffer except the last have to be programmed to zero. The **EoB** bits of the last Message Object of a FIFO Buffer is set to one, configuring it as the End of the Block.

6.7.6.12 Receiving Messages with FIFO Buffers

Received messages with identifiers matching to a FIFO Buffer are stored into a Message Object of this FIFO Buffer starting with the Message Object with the lowest message number.

When a message is stored into a Message Object of a FIFO Buffer, the **NewDat** bit of this Message Object is set. By setting **NewDat** while **EoB** is zero, the Message Object is locked for further write access by the Message Handler until the application software has written the NewDat bit back to zero.

Messages are stored into a FIFO Buffer until the last Message Object of this FIFO Buffer is reached. If none of the preceding Message Objects is released by writing **NewDat** to zero, all further messages for this FIFO Buffer will be written into the last Message Object of the FIFO Buffer and therefore overwrite previous messages.

6.7.6.13 Reading from a FIFO Buffer

When the application software transfers the contents of a Message Object to the IFn Message Buffer register by writing its number to the IFn Command Request Register, the corresponding Command Mask Register should be programmed in such a way that bits **NewDat** and **IntPnd** are reset to zero (**TxRqst/NewDat** = '1' and **ClrIntPnd** = '1'). The values of these bits in the Message Control Register always reflect the status before resetting the bits.

To assure the correct function of a FIFO Buffer, the application software should read the Message Objects starting at the FIFO Object with the lowest message number.

The following figure shows how a set of Message Objects which are concatenated to a FIFO Buffer can be handled by the application software.

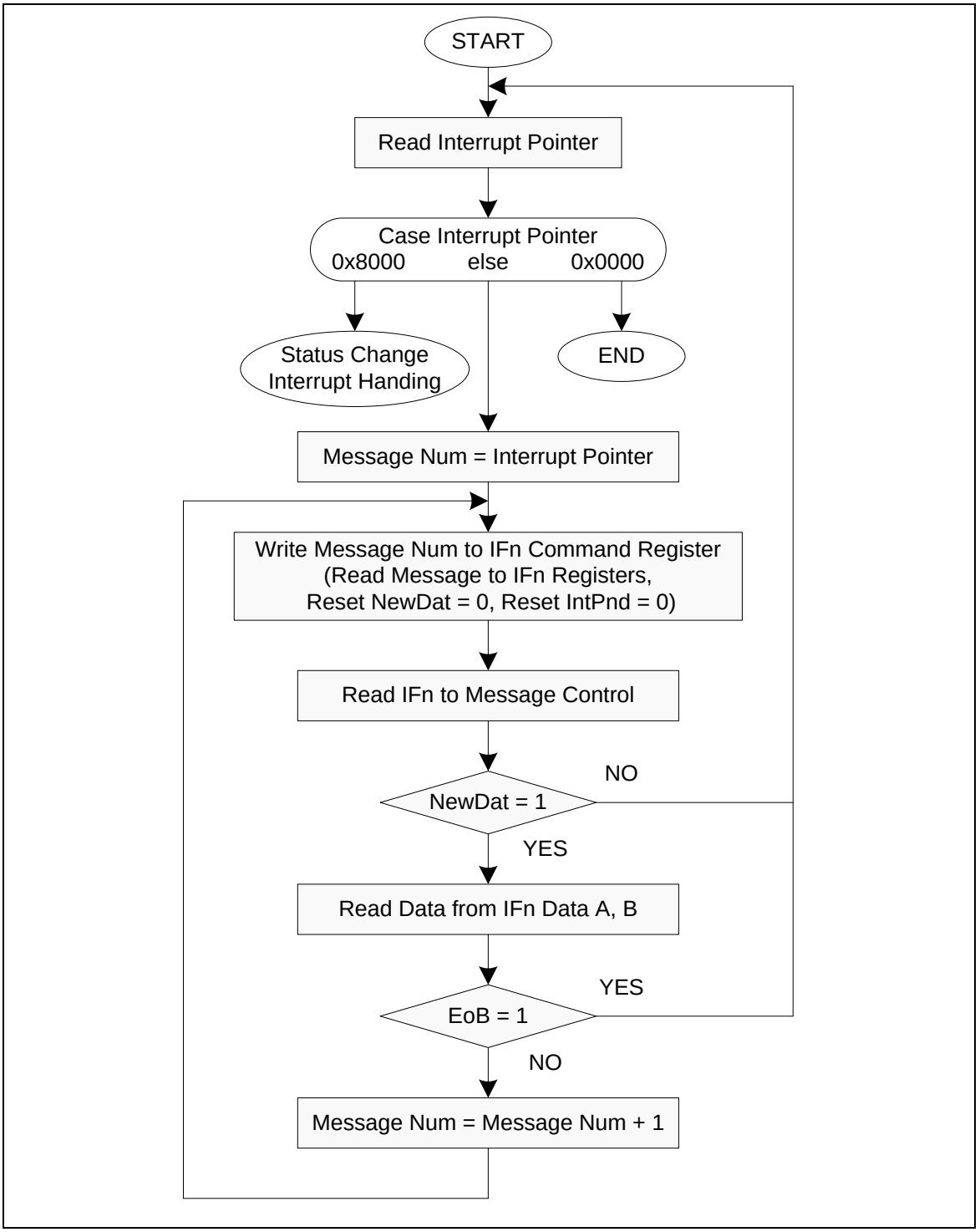


Figure 6.7-6 Application Software Handling of a FIFO Buffer

6.7.6.14 Handling Interrupts

If several interrupts are pending, the CAN Interrupt Register will point to the pending interrupt with the highest priority, disregarding their chronological order. An interrupt remains pending until the application software has cleared it.

The Status Interrupt has the highest priority. Among the message interrupts, interrupt priority of the Message Object decreases with increasing message number.

A message interrupt is cleared by clearing the **IntPnd** bit of the Message Object. The Status Interrupt is cleared by reading the Status Register.

The interrupt identifier, **IntId**, in the Interrupt Register, indicates the cause of the interrupt. When no interrupt is pending, the register will hold the value zero. If the value of the Interrupt Register is different from zero, then there is an interrupt pending and, if IE is set, the CAN_INT interrupt signal is active. The interrupt remains active until the Interrupt Register is back to value zero (the cause of the interrupt is reset) or until IE is reset.

The value 0x8000 indicates that an interrupt is pending because the CAN Core has updated (not necessarily changed) the Status Register (Error Interrupt or Status Interrupt). This interrupt has the highest priority. The application software can update (reset) the status bits **RxOk**, **TxOk** and LEC, but a write access of the software to the Status Register can never generate or reset an interrupt.

All other values indicate that the source of the interrupt is one of the Message Objects. **IntId** points to the pending message interrupt with the highest interrupt priority.

The application software controls whether a change of the Status Register may cause an interrupt (bits EIE and SIE in the CAN Control Register) and whether the interrupt line becomes active when the Interrupt Register is different from zero (bit IE in the CAN Control Register). The Interrupt Register will be updated even when IE is reset.

The application software has two possibilities to follow the source of a message interrupt. First, it can follow the **IntId** in the Interrupt Register and second it can poll the Interrupt Pending Register.

An interrupt service routine that is reading the message that is the source of the interrupt may read the message and reset the Message Object's **IntPnd** at the same time (bit **ClrIntPnd** in the Command Mask Register). When **IntPnd** is cleared, the Interrupt Register will point to the next Message Object with a pending interrupt.

6.7.6.15 Configuring the Bit Timing

Even if minor errors in the configuration of the CAN bit timing do not result in immediate failure, the performance of a CAN network can be reduced significantly.

In many cases, the CAN bit synchronization will amend a faulty configuration of the CAN bit timing to such a degree that only occasionally an error frame is generated. However, in the case of arbitration, when two or more CAN nodes simultaneously try to transmit a frame, a misplaced sample point may cause one of the transmitters to become error passive.

The analysis of such sporadic errors requires a detailed knowledge of the CAN bit synchronization inside a CAN node and interaction of the CAN nodes on the CAN bus.

6.7.6.16 Bit Time and Bit Rate

CAN supports bit rates in the range of lower than 1 Kbit/s up to 1000 Kbit/s. Each member of the CAN network has its own clock generator, usually a quartz oscillator. The timing parameter of the bit time (i.e. the reciprocal of the bit rate) can be configured individually for each CAN

node, creating a common bit rate even though the oscillator periods of the CAN nodes (f_{osc}) may be different.

The frequencies of these oscillators are not absolutely stable, small variations are caused by changes in temperature or voltage and by deteriorating components. As long as the variations remain inside a specific oscillator tolerance range (df), the CAN nodes are able to compensate for the different bit rates by re-synchronizing to the bit stream.

According to the CAN specification, the bit time is divided into four segments (see the following figure). The Synchronization Segment, the Propagation Time Segment, the Phase Buffer Segment 1 and the Phase Buffer Segment 2. Each segment consists of a specific, programmable number of time quanta (see Table 6.7-3). The length of the time quantum (t_q), which is the basic time unit of the bit time, is defined by the CAN controller's APB clock f_{APB} and the BRP bit of the Bit Timing Register (CAN_BTR): t_q = BRP / f_{APB}.

The Synchronization Segment, Sync_Seg, is that part of the bit time where edges of the CAN bus level are expected to occur. The distance between an edge that occurs outside of Sync_Seg, and the Sync_Seg is called the phase error of that edge. The Propagation Time Segment, Prop_Seg, is intended to compensate for the physical delay times within the CAN network. The Phase Buffer Segments Phase_Seg1 and Phase_Seg2 surround the Sample Point. The (Re-)Synchronization Jump Width (SJW) defines how far a re-synchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.

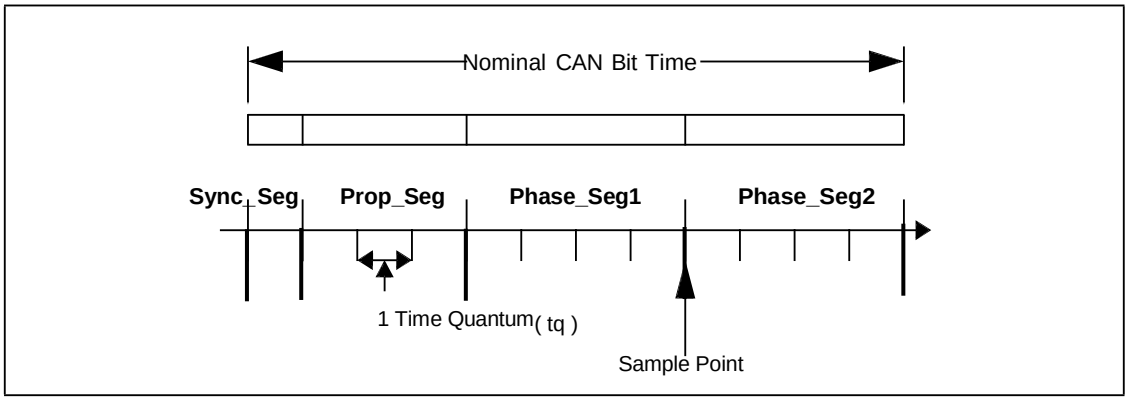


Figure 6.7-7 Bit Timing

Parameter	Range	Remark
BRP	[1 .. 32]	Defines the length of the time quantum t_q
Sync_Seg	1 t_q	Fixed length, synchronization of bus input to APB clock
Prop_Seg	[1.. 8] t_q	Compensates for the physical delay times
Phase_Seg1	[1..8] t_q	Which may be lengthened temporarily by synchronization
Phase_Seg2	[1.. 8] t_q	Which may be shortened temporarily by synchronization
SJW	[1 .. 4] t_q	Which may not be longer than either Phase Buffer Segment

This table describes the minimum programmable ranges required by the CAN protocol

Table 6.7-3 CAN Bit Time Parameters

A given bit rate may be met by different bit time configurations, but for the proper function of the CAN network the physical delay times and the oscillator's tolerance range have to be considered.

6.7.6.17 Propagation Time Segment

This part of the bit time is used to compensate physical delay times within the network. These delay times consist of the signal propagation time on the bus and the internal delay time of the CAN nodes.

Any CAN node synchronized to the bit stream on the CAN bus will be out of phase with the transmitter of that bit stream, caused by the signal propagation time between the two nodes. The CAN protocol's non-destructive bitwise arbitration and the dominant acknowledge bit provided by receivers of CAN messages requires that a CAN node transmitting a bit stream must also be able to receive dominant bits transmitted by other CAN nodes that are synchronized to that bit stream. The example in the following figure shows the phase shift and propagation times between two CAN nodes.

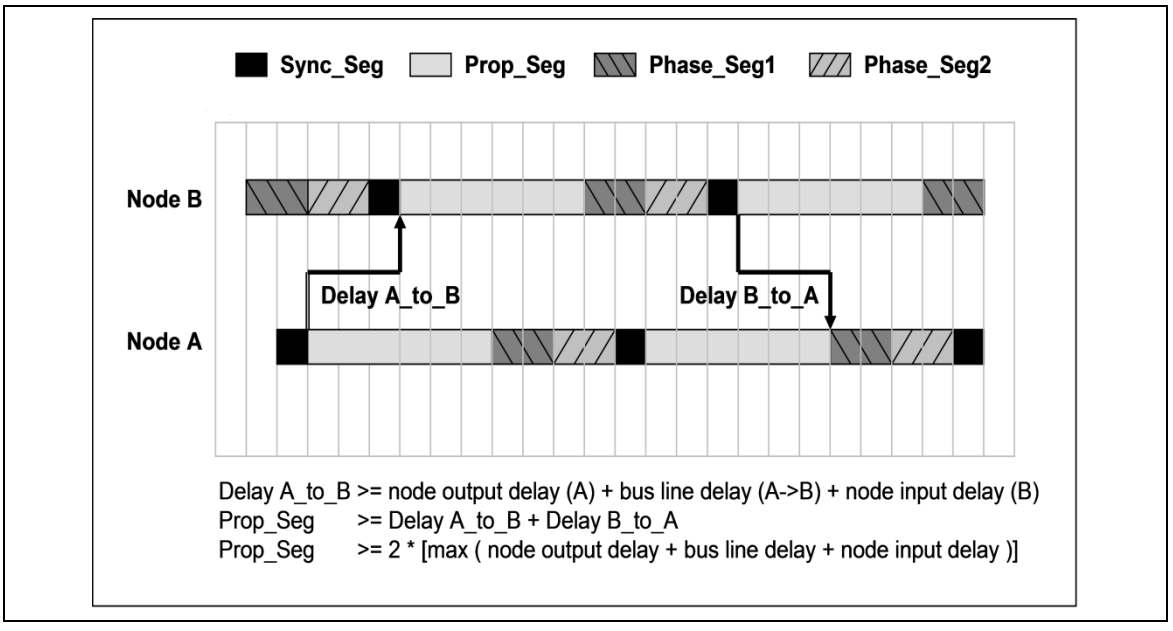


Figure 6.7-8 Propagation Time Segment

In this example, both nodes A and B are transmitters, performing an arbitration for the CAN bus. Node A has sent its Start of Frame bit less than one bit time earlier than node B, therefore node B has synchronized itself to the received edge from recessive to dominant. Since node B has received this edge delay (A_to_B) after it has been transmitted, B's bit timing segments are shifted with respect to A. Node B sends an identifier with higher priority and so it will win the arbitration at a specific identifier bit when it transmits a dominant bit while node A transmits a recessive bit. The dominant bit transmitted by node B will arrive at node A after the delay

(B_to_A).

Due to oscillator tolerances, the actual position of node A's Sample Point can be anywhere inside the nominal range of node A's Phase Buffer Segments, so the bit transmitted by node B must arrive at node A before the start of Phase_Seg1. This condition defines the length of Prop_Seg.

If the edge from recessive to dominant transmitted by node B arrives at node A after the start of Phase_Seg1, it can happen that node A samples a recessive bit instead of a dominant bit, resulting in a bit error and the destruction of the current frame by an error flag.

The error occurs only when two nodes arbitrate for the CAN bus that have oscillators of opposite ends of the tolerance range and that are separated by a long bus line. This is an example of a minor error in the bit timing configuration (Prop_Seg to short) that causes sporadic bus errors.

Some CAN implementations provide an optional 3 Sample Mode but the C_CAN does not. In this mode, the CAN bus input signal passes a digital low-pass filter, using three samples and a majority logic to determine the valid bit value. This results in an additional input delay of 1 t_q, requiring a longer Prop_Seg.

6.7.6.18 Phase Buffer Segments and Synchronization

The Phase Buffer Segments (Phase_Seg1 and Phase_Seg2) and the Synchronization Jump Width (SJW) are used to compensate for the oscillator tolerance. The Phase Buffer Segments may be lengthened or shortened by synchronization.

Synchronizations occur on edges from recessive to dominant, their purpose is to control the distance between edges and Sample Points.

Edges are detected by sampling the actual bus level in each time quantum and comparing it with the bus level at the previous Sample Point. A synchronization may be done only if a recessive bit was sampled at the previous Sample Point and if the bus level at the actual time quantum is dominant.

An edge is synchronous if it occurs inside of Sync_Seg, otherwise the distance between edge and the end of Sync_Seg is the edge phase error, measured in time quanta. If the edge occurs before Sync_Seg, the phase error is negative, else it is positive.

Two types of synchronization exist, Hard Synchronization and Re-synchronization.

A Hard Synchronization is done once at the start of a frame and inside a frame only when Re-synchronizations occur.

- **Hard Synchronization**

After a hard synchronization, the bit time is restarted with the end of Sync_Seg, regardless of the edge phase error. Thus hard synchronization forces the edge, which has caused the hard synchronization to lie within the synchronization segment of the restarted bit time.

- **Bit Re-synchronization**

Re-synchronization leads to a shortening or lengthening of the bit time such that the position of the sample point is shifted with regard to the edge.

When the phase error of the edge which causes Re-synchronization is positive, Phase_Seg1 is

lengthened. If the magnitude of the phase error is less than SJW, Phase_Seg1 is lengthened by the magnitude of the phase error, else it is lengthened by SJW.

When the phase error of the edge, which causes Re-synchronization is negative, Phase_Seg2 is shortened. If the magnitude of the phase error is less than SJW, Phase_Seg2 is shortened by the magnitude of the phase error, else it is shortened by SJW.

When the magnitude of the phase error of the edge is less than or equal to the programmed value of SJW, the results of Hard Synchronization and Re-synchronization are the same. If the magnitude of the phase error is larger than SJW, the Re-synchronization cannot compensate the phase error completely, an error (phase error - SJW) remains.

Only one synchronization may be done between two Sample Points. The Synchronizations maintain a minimum distance between edges and Sample Points, giving the bus level time to stabilize and filtering out spikes that are shorter than (Prop_Seg + Phase_Seg1).

Apart from noise spikes, most synchronizations are caused by arbitration. All nodes synchronize “hard” on the edge transmitted by the “leading” transceiver that started transmitting first, but due to propagation delay times, they cannot become ideally synchronized. The “leading” transmitter does not necessarily win the arbitration, therefore the receivers have to synchronize themselves to different transmitters that subsequently “take the lead” and that are differently synchronized to the previously “leading” transmitter. The same happens at the acknowledge field, where the transmitter and some of the receivers will have to synchronize to that receiver that “takes the lead” in the transmission of the dominant acknowledge bit.

Synchronizations after the end of the arbitration will be caused by oscillator tolerance, when the differences in the oscillator’s clock periods of transmitter and receivers sum up during the time between synchronizations (at most ten bits). These summarized differences may not be longer than the SJW, limiting the oscillator’s tolerance range.

The examples in the following figure show how the Phase Buffer Segments are used to compensate for phase errors. There are three drawings of each two consecutive bit timings. The upper drawing shows the synchronization on a “late” edge, the lower drawing shows the synchronization on an “early” edge, and the middle drawing is the reference without synchronization.

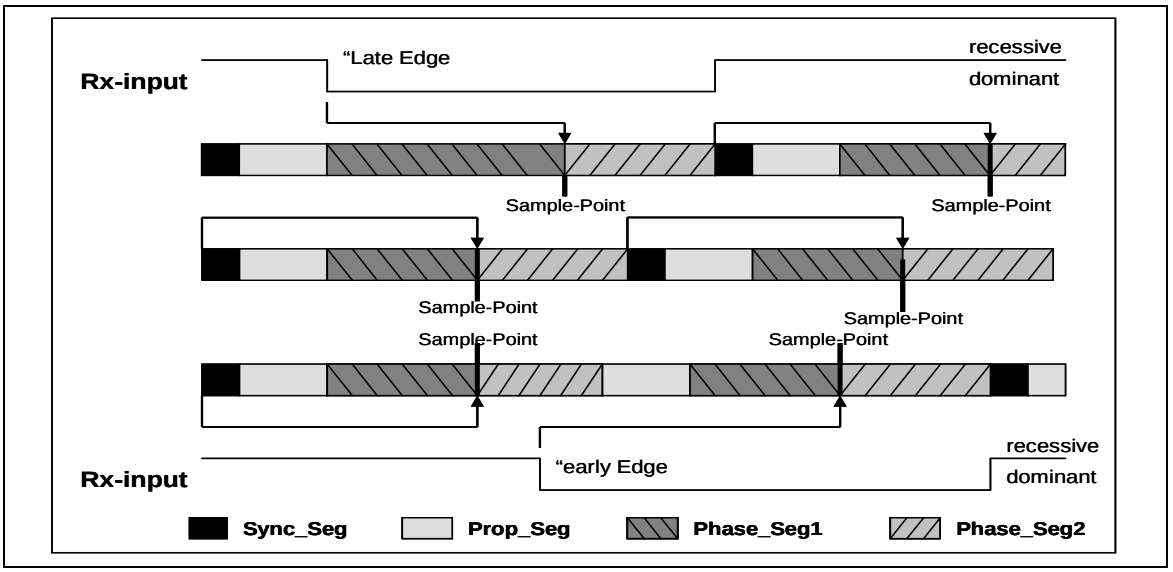


Figure 6.7-9 Synchronization on “late” and “early” Edges

In the first example an edge from recessive to dominant occurs at the end of Prop_Seg. The edge is “late” since it occurs after the Sync_Seg. Reacting to the “late” edge, Phase_Seg1 is lengthened so that the distance from the edge to the Sample Point is the same as it would have been from the Sync_Seg to the Sample Point if no edge had occurred. The phase error of this “late” edge is less than SJW, so it is fully compensated and the edge from dominant to recessive at the end of the bit, which is one nominal bit time long, occurs in the Sync_Seg.

In the second example an edge from recessive to dominant occurs during Phase_Seg2. The edge is “early” since it occurs before a Sync_Seg. Reacting to the “early” edge, Phase_Seg2 is shortened and Sync_Seg is omitted, so that the distance from the edge to the Sample Point is the same as it would have been from an Sync_Seg to the Sample Point if no edge had occurred. As in the previous example, the magnitude of this “early” edge’s phase error is less than SJW, so it is fully compensated.

The Phase Buffer Segments are lengthened or shortened temporarily only; at the next bit time, the segments return to their nominal programmed values.

In these examples, the bit timing is seen from the point of view of the CAN implementation’s state machine, where the bit time starts and ends at the Sample Points. The state machine omits Sync_Seg when synchronising on an “early” edge because it cannot subsequently redefine that time quantum of Phase_Seg2 where the edge occurs to be the Sync_Seg.

The examples in the following figure show how short dominant noise spikes are filtered by synchronisations. In both examples the spike starts at the end of Prop_Seg and has the length of (Prop_Seg + Phase_Seg1).

In the first example, the Synchronization Jump Width is greater than or equal to the phase error of the spike’s edge from recessive to dominant. Therefore the Sample Point is shifted after the end of the spike; a recessive bus level is sampled.

In the second example, SJW is shorter than the phase error, so the Sample Point cannot be shifted far enough; the dominant spike is sampled as actual bus level.

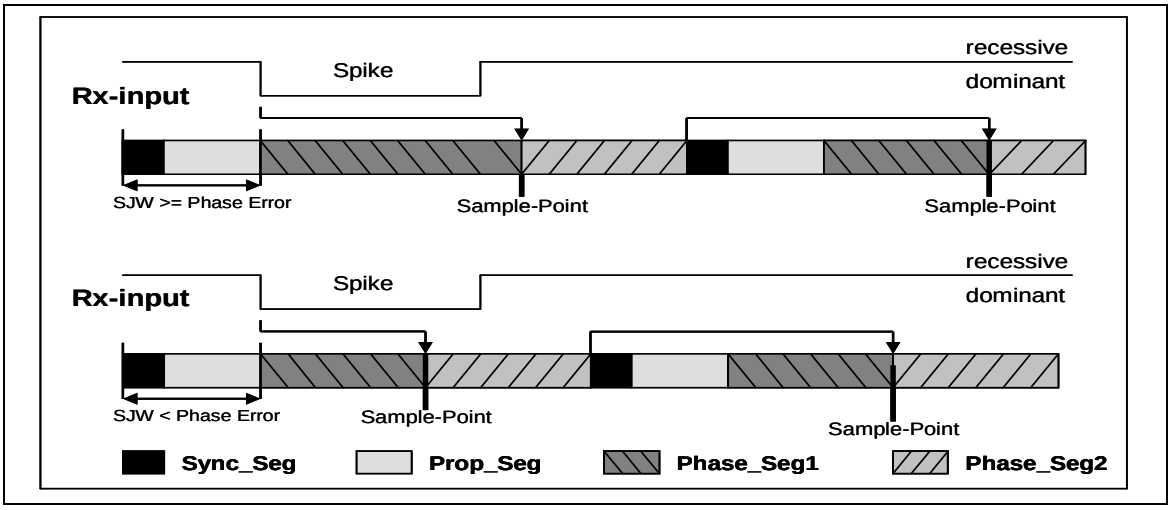


Figure 6.7-10 Filtering of Short Dominant Spikes

6.7.6.19 Oscillator Tolerance Range

The oscillator tolerance range was increased when the CAN protocol was developed from version 1.1 to version 1.2 (version 1.0 was never implemented in silicon). The option to

synchronize on edges from dominant to recessive became obsolete, only edges from recessive to dominant are considered for synchronization. The protocol update to version 2.0 (A and B) had no influence on the oscillator tolerance.

The tolerance range df for an oscillator frequency f_{osc} around the nominal frequency f_{nom} is:

□ 1 – df□ □ f_{nom} □ f_{osc} □ □ 1 + df□ □ f_{nom}

It depends on the proportions of Phase_Seg1, Phase_Seg2, SJW, and the bit time. The maximum tolerance df is the defined by two conditions (both shall be met):

$$\text{I: } df <= \frac{\min(\text{Phase\_Seg1}, \text{Phase\_Seg2})}{2 * (13 * \text{bit\_time} - \text{Phase\_Seg2})}$$

$$\text{II: } df <= \frac{\text{SJW}}{20 * \text{bit\_time}}$$

Note: These conditions base on the APB cock = f osc.

It has to be considered that SJW may not be larger than the smaller of the Phase Buffer Segments and that the Propagation Time Segment limits that part of the bit time that may be used for the Phase Buffer Segments.

The combination Prop_Seg = 1 and Phase_Seg1 = Phase_Seg2 = SJW = 4 allows the largest possible oscillator tolerance of 1.58%. This combination with a Propagation Time Segment of only 10% of the bit time is not suitable for short bit times; it can be used for bit rates of up to 125 Kbit/s (bit time = 8 □s) with a bus length of 40 m.

6.7.6.20 Configuring the CAN Protocol Controller

In most CAN implementations and also in the C_CAN, the bit timing configuration is programmed in two register bytes. The sum of Prop_Seg and Phase_Seg1 (as TSEG1) is combined with Phase_Seg2 (as TSEG2) in one byte, SJW and BRP are combined in the other byte.

In these bit timing registers, the four components TSEG1, TSEG2, SJW, and BRP have to be programmed to a numerical value that is one less than its functional value. Therefore, instead of values in the range of [1..n], values in the range of [0..n-1] are programmed. That way, e.g. SJW (functional range of [1..4]) is represented by only two bits.

Therefore the length of the bit time is (programmed values) [TSEG1 + TSEG2 + 3] t_q or (functional values) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] t_q.

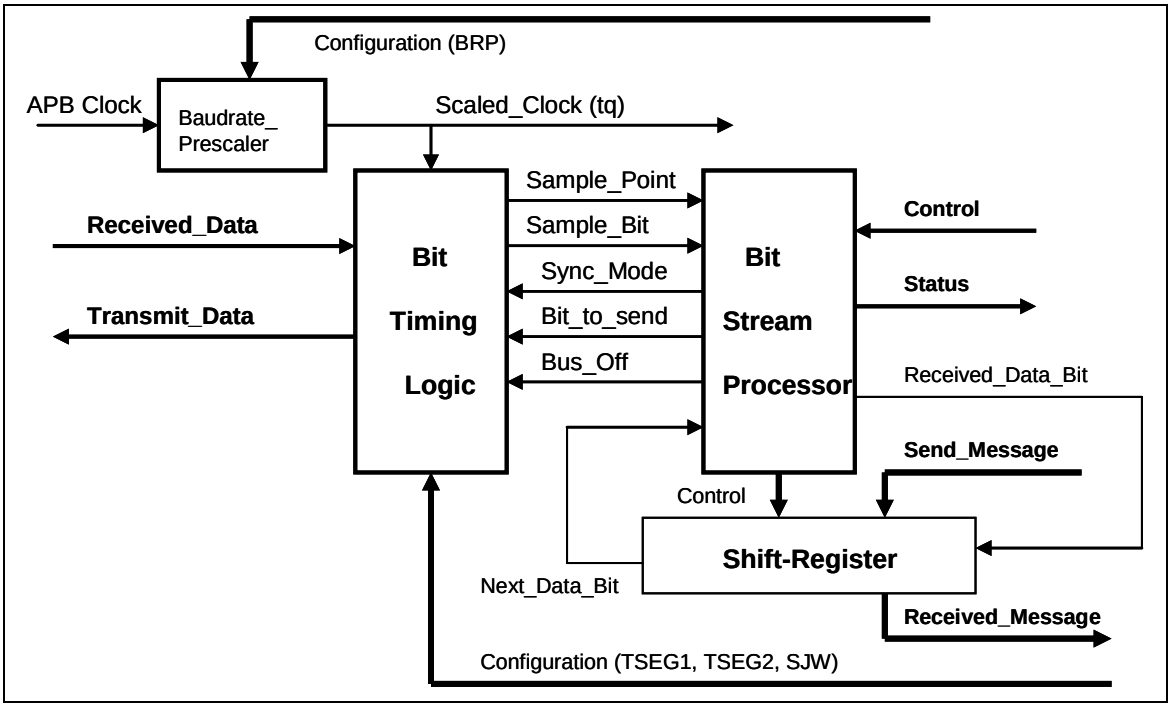


Figure 6.7-11 Structure of the CAN Core's CAN Protocol Controller

The data in the bit timing registers is the configuration input of the CAN protocol controller. The Baud Rate Prescaler (configured by BRP) defines the length of the time quantum, the basic time unit of the bit time; the Bit Timing Logic (configured by TSEG1, TSEG2, and SJW) defines the number of time quanta in the bit time.

The processing of the bit time, the calculation of the position of the Sample Point, and occasional synchronizations are controlled by the BTL (Bit Timing Logic) state machine, which is evaluated once each time quantum. The rest of the CAN protocol controller, the BSP (Bit Stream Processor) state machine is evaluated once each bit time, at the Sample Point.

The Shift Register sends the messages serially and parallelizes received messages. It's loading and shifting is controlled by the BSP.

The BSP translates messages into frames and vice versa. It generates and discards the enclosing fixed format bits, inserts and extracts stuff bits, calculates and checks the CRC code, performs the error management, and decides which type of synchronization is to be used. It is evaluated at the Sample Point and processes the sampled bus input bit. The time that is needed to calculate the next bit to be sent after the Sample point (e.g. data bit, CRC bit, stuff bit, error flag, or idle) is called the Information Processing Time (IPT).

The IPT is application specific but may not be longer than 2 t_q; the IPT for the C_CAN is 0 t_q. Its length is the lower limit of the programmed length of Phase_Seg2. In case of a synchronization, Phase_Seg2 may be shortened to a value less than IPT, which does not affect bus timing.

6.7.6.21 Calculating Bit Timing Parameters

Usually, the calculation of the bit timing configuration starts with a desired bit rate or bit time. The resulting bit time (1/bit rate) must be an integer multiple of the APB clock period.

The bit time may consist of 4 to 25 time quanta, the length of the time quantum t_q is defined by

the Baud Rate Prescaler with t_q = (Baud Rate Prescaler)/f_{apb_clk}. Several combinations may lead to the desired bit time, allowing iterations of the following steps.

First part of the bit time to be defined is the Prop_Seg. Its length depends on the delay times measured in the APB clock. A maximum bus length as well as a maximum node delay has to be defined for expandible CAN bus systems. The resulting time for Prop_Seg is converted into time quanta (rounded up to the nearest integer multiple of t_q).

The Sync_Seg is 1 t_q long (fixed), leaving (bit time – Prop_Seg – 1) t_q for the two Phase Buffer Segments. If the number of remaining t_q is even, the Phase Buffer Segments have the same length, Phase_Seg2 □ Phase_Seg1, else Phase_Seg2 □ Phase_Seg1 + 1.

The minimum nominal length of Phase_Seg2 has to be regarded as well. Phase_Seg2 may not be shorter than the IPT of the CAN controller, which, depending on the actual implementation, is in the range of [0..2] t_q.

The length of the Synchronization Jump Width is set to its maximum value, which is the minimum of 4 and Phase_Seg1.

The oscillator tolerance range necessary for the resulting configuration is calculated by the formulas given in Section 5.13.6.10.4: Oscillator Tolerance Range

If more than one configuration is possible, that configuration allowing the highest oscillator tolerance range should be chosen.

CAN nodes with different system clocks require different configurations to come to the same bit rate. The calculation of the propagation time in the CAN network, based on the nodes with the longest delay times, is done once for the whole network.

The oscillator tolerance range of the CAN systems is limited by that node with the lowest tolerance range.

The calculation may show that bus length or bit rate have to be decreased or that the stability of the oscillator frequency has to be increased in order to find a protocol compliant configuration of the CAN bit timing. The resulting configuration is written into the Bit Timing Register: (Phase_Seg2-1) & (Phase_Seg1+Prop_Seg-1) & (SynchronisationJumpWidth-1)&(Prescaler-1)

Example for Bit Timing at High Baud Rate

In this example, the frequency of APB_CLK is 10 MHz, BRP is 0, the bit rate is 1 MBit/s.

T_q	100	ns	= t_{APB_CLK}
delay of bus driver	50	ns	
delay of receiver circuit	30	ns	
delay of bus line (40m)	220	ns	
t_{Prop}	600	ns	= 6 • t_q
t_{SJW}	100	ns	= 1 • t_q
t_{TSeg1}	700	ns	= t_{Prop} + t_{SJW}
t_{TSeg2}	200	ns	= Information Processing Time + 1 • t_q
tSync-Seg	100	ns	= 1 • tq
bit time	1000	ns	= $t_{Sync\text{-}Seg}$ + t_{TSeg1} + t_{TSeg2}
tolerance for APB_CLK	0.39	%	$= \frac{Min(PB1, PB2)}{2 \times 13 \times (bit\ \ time - PB2))}$ $= \frac{0.1\mu s}{2x(13x(1\mu s - 0.2\mu s))}$

In this example, the concatenated bit time parameters are $(2\text{-}1)_3$&$(7\text{-}1)_4$&$(1\text{-}1)_2$&$(1\text{-}1)_6$, and the Bit Timing Register is programmed to= 0x1600.

Example for Bit Timing at Low Baud Rate

In this example, the frequency of APB_CLK is 2 MHz, BRP is 1, the bit rate is 100 Kbit/s.

t_q	1	□s = 2 •t_{APB_CLK}
delay of bus driver	200	ns
delay of receiver circuit	80	ns
delay of bus line (40m)	220	ns
t_{Prop}	1	□s = 1 •t_q
t_{SJW}	4	□s = 4 • t_q
t_{TSeg1}	5	□s = t_{Prop} + t_{SJW}
t_{TSeg2}	4	□s = Information Processing Time + 3 • t_q
$t_{Sync-Seg}$	1	□s = 1 • t_q
bit time	10	□s = $t_{Sync-Seg}$ + t_{TSeg1} + t_{TSeg2}
tolerance for APB_CLK	1.58	$\% = \frac{Min(PB1, PB2)}{2 \times 13 \times (bit\ \ time - PB2))}$ $= \frac{4us}{2 \times (13 \times (10us - 4us))}$

In this example, the concatenated bit time parameters are $(4-1)_3$&$(5-1)_4$&$(4-1)_2$&$(2-1)_6$, and the Bit Timing Register is programmed to= 0x34C1.

6.7.7 Register Description

The C_CAN allocates an address space of 256 bytes. The registers are organized as 16-bit registers.

The two sets of interface registers (IF1 and IF2) control the software access to the Message RAM. They buffer the data to be transferred to and from the RAM, avoiding conflicts between software accesses and message reception/transmission.

6.7.8 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
CAN Base Address: **CANx_BA = 0x400A_0000 + (0x1000 * x)** **x = 0,1**				
CAN_CON	CANx_BA+0x00	R/W	Control Register	0x0000_0001
CAN_STATUS	CANx_BA+0x04	R/W	Status Register	0x0000_0000
CAN_ERR	CANx_BA+0x08	R	Error Counter Register	0x0000_0000
CAN_BTIME	CANx_BA+0x0C	R/W	Bit Timing Register	0x0000_2301
CAN_IIDR	CANx_BA+0x10	R	Interrupt Identifier Register	0x0000_0000
CAN_TEST	CANx_BA+0x14	R/W	Test Register	0x0000_00x0
CAN_BRPE	CANx_BA+0x18	R/W	Baud Rate Prescaler Extension Register	0x0000_0000
CAN_IF1_CREQ	CANx_BA+0x20	R/W	IF1 Command Request Register	0x0000_0001
CAN_IF1_CMASK	CANx_BA+0x24	R/W	IF1 Command Mask Register	0x0000_0000
CAN_IF1_MASK1	CANx_BA+0x28	R/W	IF1 Mask 1 Register	0x0000_FFFF
CAN_IF1_MASK2	CANx_BA+0x2C	R/W	IF1 Mask 2 Register	0x0000_FFFF
CAN_IF1_ARB1	CANx_BA+0x30	R/W	IF1 Arbitration 1 Register	0x0000_0000
CAN_IF1_ARB2	CANx_BA+0x34	R/W	IF1 Arbitration 2 Register	0x0000_0000
CAN_IF1_MCON	CANx_BA+0x38	R/W	IF1 Message Control Register	0x0000_0000
CAN_IF1_DAT_A1	CANx_BA+0x3C	R/W	IF1 Data A1 Register	0x0000_0000
CAN_IF1_DAT_A2	CANx_BA+0x40	R/W	IF1 Data A2 Register	0x0000_0000
CAN_IF1_DAT_B1	CANx_BA+0x44	R/W	IF1 Data B1 Register	0x0000_0000
CAN_IF1_DAT_B2	CANx_BA+0x48	R/W	IF1 Data B2 Register	0x0000_0000
CAN_IF2_CREQ	CANx_BA+0x80	R/W	IF2 Command Request Register	0x0000_0001

CAN_IF2_CMASK	CANx_BA+0x84	R/W	IF2 Command Mask Register	0x0000_0000
CAN_IF2_MASK1	CANx_BA+0x88	R/W	IF2 Mask 1 Register	0x0000_FFFF
CAN_IF2_MASK2	CANx_BA+0x8C	R/W	IF2 Mask 2 Register	0x0000_FFFF
CAN_IF2_ARB1	CANx_BA+0x90	R/W	IF2 Arbitration 1 Register	0x0000_0000
CAN_IF2_ARB2	CANx_BA+0x94	R/W	IF2 Arbitration 2 Register	0x0000_0000
CAN_IF2_MCON	CANx_BA+0x98	R/W	IF2 Message Control Register	0x0000_0000
CAN_IF2_DAT_A1	CANx_BA+0x9C	R/W	IF2 Data A1 Register	0x0000_0000
CAN_IF2_DAT_A2	CANx_BA+0xA0	R/W	IF2 Data A2 Register	0x0000_0000
CAN_IF2_DAT_B1	CANx_BA+0xA4	R/W	IF2 Data B1 Register	0x0000_0000
CAN_IF2_DAT_B2	CANx_BA+0xA8	R/W	IF2 Data B2 Register	0x0000_0000
CAN_TXREQ1	CANx_BA+0x100	R	Transmission Request Register 1	0x0000_0000
CAN_TXREQ2	CANx_BA+0x104	R	Transmission Request Register 2	0x0000_0000
CAN_NDAT1	CANx_BA+0x120	R	New Data Register 1	0x0000_0000
CAN_NDAT2	CANx_BA+0x124	R	New Data Register 2	0x0000_0000
CAN_IPND1	CANx_BA+0x140	R	Interrupt Pending Register 1	0x0000_0000
CAN_IPND2	CANx_BA+0x144	R	Interrupt Pending Register 2	0x0000_0000
CAN_MVLD1	CANx_BA+0x160	R	Message Valid Register 1	0x0000_0000
CAN_MVLD2	CANx_BA+0x164	R	Message Valid Register 2	0x0000_0000
CAN_WU_EN	CANx_BA+0x168	R/W	Wake-up Enable Control Register	0x0000_0000
CAN_WU_STATUS	CANx_BA+0x16C	R/W	Wake-up Status Register	0x0000_0000

Note: 0x00 & 0br0000000, where r signifies the actual value of the CAN_RX

6.7.9 CAN Interface Reset State

After the hardware reset, the C_CAN registers hold the reset values which are given in the register description in *6.7.8 Register Map*.

Additionally the *busoff* state is reset and the output CAN_TX is set to recessive (HIGH). The value 0x0001 (Init = '1') in the CAN Control Register enables the software initialization. The C_CAN does not influence the CAN bus until the application software resets the Init bit to '0'.

The data stored in the Message RAM is not affected by a hardware reset. After powered on, the contents of the Message RAM are undefined.

CAN Register Map for Each Bit Function

Addr Offset	Register Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00h	CAN_CON	Reserved								Test	CCE	DAR	Res	EIE	SIE	IE	Init
04h	CAN_STATUS	Reserved								BOff	EWarn	EPass	RxOk	TxOk	LEC		
08h	CAN_ERR	RP	REC6-0							TEC7-0							
0Ch	CAN_BTIME	Res	TSeg2			TSeg1				SJW		BRP					
10h	CAN_IIDR	IntId15-8								IntId7-0							
14h	CAN_TEST	Reserved								Rx	Tx1	Tx0	LBack	Silent	Basic	Reserved	
18h	CAN_BRPE	Reserved												BRPE			
20h	CAN_IF1_CREQ	Busy	Reserved									Message Number					
24h	CAN_IF1_CMASK	Reserved								WR/RD	Mask	Arb	Control	ClrIntPnd	TxRqst/	Data A	Data B
28h	CAN_IF1_MASK1	Msk15-0															
2Ch	CAN_IF1_MASK2	MXtd	MDir	Res	Msk28-16												
30h	CAN_IF1_ARB1	ID15-0															
34h	CAN_IF1_ARB2	MsgVal	Xtd	Dir	ID28-16												

Addr Offset	Register Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
38h	CAN_IF1_MCON	NewDat	MsgLst	IntPnd	UMask	TxIE	RxIE	RmtEn	TxRqst	EoB	Reserved			DLC3-0			
3Ch	CAN_IF1_DAT_A1	Data(1)								Data(0)							
40h	CAN_IF1_DAT_A2	Data(3)								Data(2)							
44h	CAN_IF1_DAT_B1	Data(5)								Data(4)							
48h	CAN_IF1_DAT_B2	Data(7)								Data(6)							
80h	CAN_IF2_CREQ	Busy	Reserved									Message Number					
84h	CAN_IF2_CMASK	Reserved								WR/RD	Mask	Arb	Control	ClrIntPnd	TxRqst/	Data A	Data B
88h	CAN_IF2_MASK1	Msk15-0															
8Ch	CAN_IF2_MASK2	MXtd	MDir	Res.	Msk28-16												
90h	CAN_IF2_ARB1	ID15-0															
94h	CAN_IF2_ARB2	MsgVal	Xtd	Dir	ID28-16												
98h	CAN_IF2_MCON	NewDat	MsgLst	IntPnd	UMask	TxIE	RxIE	RmtEn	TxRqst	EoB	Reserved			DLC3-0			
9Ch	CAN_IF2_DAT_A1	Data(1)								Data(0)							

Addr Offset	Register Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A0h	CAN_IF2_DAT_A2	Data(3)								Data(2)							
A4h	CAN_IF2_DAT_B1	Data(5)								Data(4)							
A8h	CAN_IF2_DAT_B2	Data(7)								Data(6)							
100h	CAN_TXREQ1	TxRqst16-1															
104h	CAN_TXREQ2	TxRqst32-17															
120h	CAN_NDAT1	NewDat16-1															
124h	CAN_NDAT2	NewDat32-17															
140h	CAN_IPND1	IntPnd16-1															
144h	CAN_IPND2	IntPnd32-17															
160h	CAN_MVLD1	MsgVal16-1															
164h	CAN_MVLD2	MsgVal32-17															
168h	CAN_WU_EN	Reserved															WAKUP_EN
16Ch	CAN_WU_STATUS	Reserved															WAKUP_STS
170h	CAN_RAM_CEN	Reserved															RAM_CEN
Others	Reserved	Reserved															

Table 6.7-4 CAN Register Map for Each Bit Function

Note: Reserved bits are read as 0' except for IFn Mask 2 Register where they are read as '1'.

Res. = Reserved

CAN Control Register (CAN_CON)

Register	Offset	R/W	Description	Reset Value
CAN_CON	CANx_BA+0x00	R/W	Control Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Test	CCE	DAR	Reserved	EIE	SIE	IE	Init

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**Test**	**Test Mode Enable Bit** 0 = Normal Operation. 1 = Test Mode.
[6]	**CCE**	**Configuration Change Enable Bit** 0 = No write access to the Bit Timing Register. 1 = Write access to the Bit Timing Register (CAN_BTIME) allowed. (while Init bit (CAN_CON[0]) = 1).
[5]	**DAR**	**Automatic Re-Transmission Disable Bit** 0 = Automatic Retransmission of disturbed messages enabled. 1 = Automatic Retransmission disabled.
[4]	**Reserved**	Reserved.
[3]	**EIE**	**Error Interrupt Enable Bit** 0 = Disabled - No Error Status Interrupt will be generated. 1 = Enabled - A change in the bits BOff (CAN_STATUS[7]) or EWarn (CAN_STATUS[6]) in the Status Register will generate an interrupt.
[2]	**SIE**	**Status Change Interrupt Enable Bit** 0 = Disabled - No Status Change Interrupt will be generated. 1 = Enabled - An interrupt will be generated when a message transfer is successfully completed or a CAN bus error is detected.
[1]	**IE**	**Module Interrupt Enable Bit** 0 = Disabled. 1 = Enabled.
[0]	**Init**	**Init Initialization** 0 = Normal Operation.

		1 = Initialization is started.

Note: The busoff recovery sequence (see CAN Specification Rev. 2.0) cannot be shortened by setting or resetting the Init bit. If the device goes in the busoff state, it will set Init of its own accord, stopping all bus activities. Once Init has been cleared by the CPU, the device will then wait for 129 occurrences of Bus Idle (129 * 11 consecutive recessive bits) before resuming normal operations. At the end of the busoff recovery sequence, the Error Management Counters will be reset.

During the waiting time after resetting Init, each time a sequence of 11 recessive bits has been monitored, a Bit0Error code is written to the Status Register, enabling the CPU to readily check up whether the CAN bus is stuck at dominant or continuously disturbed and to monitor the proceeding of the busoff recovery sequence.

CAN Status Register (CAN_STATUS)

Register	Offset	R/W	Description	Reset Value
CAN_STATUS	CANx_BA+0x04	R/W	Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
BOFF	EWarn	EPass	RxOK	TxOK	LEC		

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**BOff**	**Bus-Off Status (Read Only)** 0 = The CAN module is not in bus-off state. 1 = The CAN module is in bus-off state.
[6]	**EWarn**	**Error Warning Status (Read Only)** 0 = Both error counters are below the error warning limit of 96. 1 = At least one of the error counters in the EML has reached the error warning limit of 96.
[5]	**EPass**	**Error Passive (Read Only)** 0 = The CAN Core is error active. 1 = The CAN Core is in the error passive state as defined in the CAN Specification.
[4]	**RxOK**	**Received A Message Successfully** 0 = No message has been successfully received since this bit was last reset by the CPU. This bit is never reset by the CAN Core. 1 = A message has been successfully received since this bit was last reset by the CPU (independent of the result of acceptance filtering).
[3]	**TxOK**	**Transmitted A Message Successfully** 0 = Since this bit was reset by the CPU, no message has been successfully transmitted. This bit is never reset by the CAN Core. 1 = Since this bit was last reset by the CPU, a message has been successfully (error free and acknowledged by at least one other node) transmitted.
[2:0]	**LEC**	**Last Error Code (Type Of The Last Error To Occur On The CAN Bus)** The LEC field holds a code, which indicates the type of the last error to occur on the CAN bus. This field will be cleared to '0' when a message has been transferred (reception or transmission) without error. The unused code '7' may be written by the CPU to check for updates. The following table describes the error code.

Error Code	Meanings
0	No Error
1	Stuff Error: More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed.
2	Form Error: A fixed format part of a received frame has the wrong format.
3	AckError: The message this CAN Core transmitted was not acknowledged by another node.
4	Bit1Error: During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (bit of logical value '1'), but the monitored bus value was dominant.
5	Bit0Error: During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), though the device wanted to send a dominant level (data or identifier bit logical value '0'), but the monitored Bus value was recessive. During busoff recovery, this status is set each time a sequence of 11 recessive bits has been monitored. This enables the CPU to monitor the proceedings of the busoff recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed).
6	CRCError: The CRC check sum was incorrect in the message received, the CRC received for an incoming message does not match with the calculated CRC for the received data.
7	Unused: When the LEC shows the value '7', no CAN bus event was detected since the CPU wrote this value to the LEC.

Table 6.7-5 Error Code

Status Interrupts

A Status Interrupt is generated by bits **BOff** and **EWarn** (Error Interrupt) or by **RxOk**, **TxOk**, and **LEC** (Status Change Interrupt) assumed that the corresponding enable bits in the CAN Control Register are set. A change of bit **EPass** or a write to **RxOk**, **TxOk**, or **LEC** will never generate a Status Interrupt.

Reading the Status Register will clear the Status Interrupt value (8000h) in the Interrupt Register, if it is pending.

CAN Error Counter Register (CAN_ERR)

Register	Offset	R/W	Description	Reset Value
CAN_ERR	CANx_BA+0x08	R	Error Counter Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
RP	REC[6:0]						
7	6	5	4	3	2	1	0
TEC[7:0]							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15]	**RP**	**Receive Error Passive** 0 = The Receive Error Counter is below the error passive level. 1 = The Receive Error Counter has reached the error passive level as defined in the CAN Specification.
[14:8]	**REC**	**Receive Error Counter** Actual state of the Receive Error Counter. Values between 0 and 127.
[7:0]	**TEC**	**Transmit Error Counter** Actual state of the Transmit Error Counter. Values between 0 and 255.

Bit Timing Register (CAN_BTIME)

Register	Offset	R/W	Description	Reset Value
CAN_BTIME	CANx_BA+0x0C	R/W	Bit Timing Register	0x0000_2301

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	TSeg2			TSeg1			
7	6	5	4	3	2	1	0
SJW		BRP					

Bits	Description	
[31:15]	**Reserved**	Reserved.
[14:12]	**TSeg2**	**Time Segment After Sample Point** 0x0-0x7: Valid values for TSeg2 are [0 … 7]. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.
[11:8]	**TSeg1**	**Time Segment Before The Sample Point Minus Sync_Seg** 0x01-0x0F: valid values for TSeg1 are [1 … 15]. The actual interpretation by the hardware of this value is such that one more than the value programmed is used.
[7:6]	**SJW**	**(Re)Synchronization Jump Width** 0x0-0x3: Valid programmed values are [0 … 3]. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.
[5:0]	**BRP**	**Baud Rate Prescaler** 0x01-0x3F: The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Baud Rate Prescaler are [0 … 63]. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

Note: With a module clock APB_CLK of 8 MHz, the reset value of 0x2301 configures the C_CAN for a bit rate of 500 Kbit/s. The registers are only writable if bits CCE and Init in the CAN Control Register are set.

Interrupt Identify Register (CAN_IIDR)

Register	Offset	R/W	Description	Reset Value
CAN_IIDR	CANx_BA+0x10	R	Interrupt Identifier Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
IntId[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IntId[7:0]							

Bits	Description	
[15:0]	**IntId**	**Interrupt Identifier (Indicates The Source Of The Interrupt)** If several interrupts are pending, the CAN Interrupt Register will point to the pending interrupt with the highest priority, disregarding their chronological order. An interrupt remains pending until the application software has cleared it. If **IntId** is different from 0x0000 and IE is set, the IRQ interrupt signal to the EIC is active. The interrupt remains active until **IntId** is back to value 0x0000 (the cause of the interrupt is reset) or until **IE** is reset. The Status Interrupt has the highest priority. Among the message interrupts, the Message Object' s interrupt priority decreases with increasing message number. A message interrupt is cleared by clearing the Message Object's **IntPnd** bit. The Status Interrupt is cleared by reading the Status Register.

IntId Value	Meanings
0x0000	No Interrupt is Pending
0x0001-0x0020	Number of Message Object which caused the interrupt.
0x0021-0x7FFF	Unused
0x8000	Status Interrupt
0x8001-0xFFFF	Unused

Table 6.7-6 Source of Interrupts

Test Register (CAN_TEST)

Register	Offset	R/W	Description	Reset Value
CAN_TEST	CANx_BA+0x14	R/W	Test Register	0x0000_00x0

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Rx	Tx[1:0]		LBack	Silent	Basic	Res	

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**Rx**	**Monitors The Actual Value Of CAN_RX Pin (Read Only) *(1)** 0 = The CAN bus is dominant (CAN_RX = ‘0’). 1 = The CAN bus is recessive (CAN_RX = ‘1’).
[6:5]	**Tx[1:0]**	**Tx[1:0]: Control Of CAN_TX Pin** 00 = Reset value, CAN_TX pin is controlled by the CAN Core. 01 = Sample Point can be monitored at CAN_TX pin. 10 = CAN_TX pin drives a dominant (‘0’) value. 11 = CAN_TX pin drives a recessive (‘1’) value.
[4]	**LBack**	**Loop Back Mode Enable Bit** 0 = Loop Back Mode is disabled. 1 = Loop Back Mode is enabled.
[3]	**Silent**	**Silent Mode** 0 = Normal operation. 1 = The module is in Silent Mode.
[2]	**Basic**	**Basic Mode** 0 = Basic Mode disabled. 1= IF1 Registers used as Tx Buffer, IF2 Registers used as Rx Buffer.
[1:0]	**Res**	**Reserved** There are reserved bits. These bits are always read as ‘0’ and must always be written with ‘0’.

Reset value: 0000 0000 R000 0000 b (R:current value of RX pin)

Write access to the Test Register is enabled by setting the Test bit in the CAN Control Register. The different test functions may be combined, but **Tx[1-0]** “00” disturbs message transfer.

Baud Rate Prescaler Extension REGISTER (CAN_BRPE)

Register	Offset	R/W	Description	Reset Value
CAN_BRPE	CANx_BA+0x18	R/W	Baud Rate Prescaler Extension Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**BRPE**			

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3:0]	**BRPE**	**BRPE: Baud Rate Prescaler Extension** 0x00-0x0F: By programming **BRPE**, the Baud Rate Prescaler can be extended to values up to 1023. The actual interpretation by the hardware is that one more than the value programmed by **BRPE** (MSBs) and **BTIME** (LSBs) is used.

Message Interface Register Sets

There are two sets of Interface Registers, which are used to control the CPU access to the Message RAM. The Interface Registers avoid conflict between the CPU accesses to the Message RAM and CAN message reception and transmission by buffering the data to be transferred. A complete Message Object or parts of the Message Object may be transferred between the Message RAM and the IF*n* Message Buffer registers in one single transfer.

The function of the two interface register sets is identical except for the Basic test mode. They can be used the way one set of registers is used for data transfer to the Message RAM while the other set of registers is used for the data transfer from the Message RAM, allowing both processes to be interrupted by each other. The following table provides an overview of the two Interface Register sets.

Each set of Interface Registers consists of Message Buffer Registers controlled by their own Command Registers. The Command Mask Register specifies the direction of the data transfer and which parts of a Message Object will be transferred. The Command Request Register is used to select a Message Object in the Message RAM as target or source for the transfer and to start the action specified in the Command Mask Register.

Address	IF1 Register Set	Address	IF2 Register Set
CANx_BA+0x20	IF1 Command Request	CANx_BA+0x80	IF2 Command Request
CANx_BA+0x24	IF1 Command Mask	CANx_BA+0x84	IF2 Command Mask
CANx_BA+0x28	IF1 Mask 1	CANx_BA+0x88	IF2 Mask 1
CANx_BA+0x2C	IF1 Mask 2	CANx_BA+0x8C	IF2 Mask 2
CANx_BA+0x30	IF1 Arbitration 1	CANx_BA+0x90	IF2 Arbitration 1
CANx_BA+0x34	IF1 Arbitration 2	CANx_BA+0x94	IF2 Arbitration 2
CANx_BA+0x38	IF1 Message Control	CANx_BA+0x98	IF2 Message Control
CANx_BA+0x3C	IF1 Data A 1	CANx_BA+0x9C	IF2 Data A 1
CANx_BA+0x40	IF1 Data A 2	CANx_BA+0xA0	IF2 Data A 2
CANx_BA+0x44	IF1 Data B 1	CANx_BA+0xA4	IF2 Data B 1
CANx_BA+0x48	IF1 Data B 2	CANx_BA+0xA8	IF2 Data B 2

Table 6.7-7 IF1 and IF2 Message Interface Register

IFn Command Request Register (CAN_IFn_CREQ)

Register	Offset	R/W	Description	Reset Value
CAN_IF1_CREQ	CANx_BA+0x20	R/W	IF1 Command Request Register	0x0000_0001
CAN_IF2_CREQ	CANx_BA+0x80	R/W	IF2 Command Request Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Busy	Res						
7	6	5	4	3	2	1	0
Res		Message Number					

Bits	Description	
[15]	**Busy**	**Busy Flag** 0 = Read/write action has finished. 1 = Writing to the IFn Command Request Register is in progress. This bit can only be read by the software.
[14:6]	**Reserved**	Reserved.
[5:0]	**Message Number**	**Message Number** 0x01-0x20: Valid Message Number, the Message Object in the Message RAM is selected for data transfer. 0x00: Not a valid Message Number, interpreted as 0x20. 0x21-0x3F: Not a valid Message Number, interpreted as 0x01-0x1F.

A message transfer is started as soon as the application software has written the message number to the Command Request Register. With this write operation, the Busy bit is automatically set to notify the CPU that a transfer is in progress. After a waiting time of 3 to 6 APB_CLK periods, the transfer between the Interface Register and the Message RAM is completed. The Busy bit is cleared.

Note: When a Message Number that is not valid is written into the Command Request Register, the Message Number will be transformed into a valid value and that Message Object will be transferred.

IFn Command Mask Register (CAN_IFn_CMASK)

The control bits of the IF*n* Command Mask Register specify the transfer direction and select which of the IF*n* Message Buffer Registers are source or target of the data transfer.

Register	Offset	R/W	Description	Reset Value
CAN_IF1_CMASK	CANx_BA+0x24	R/W	IF1 Command Mask Register	0x0000_0000
CAN_IF2_CMASK	CANx_BA+0x84	R/W	IF2 Command Mask Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
WR/RD	Mask	Arb	Control	ClrIntPnd	TxRqst/ NewDat	DAT_A	DAT_B

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**WR/RD**	**Write / Read Mode** 0 = Read: Transfer data from the Message Object addressed by the Command Request Register into the selected Message Buffer Registers. 1 = Write: Transfer data from the selected Message Buffer Registers to the Message Object addressed by the Command Request Register.
[6]	**Mask**	**Access Mask Bits** Write Operation: 0 = Mask bits unchanged. 1 = Transfer Identifier Mask + MDir + MXtd to Message Object. Read Operation: 0 = Mask bits unchanged. 1 = Transfer Identifier Mask + MDir + MXtd to IFn Message Buffer Register.
[5]	**Arb**	**Access Arbitration Bits** Write Operation: 0 = Arbitration bits unchanged. 1 = Transfer Identifier + Dir (CAN_IFn_ARB2[13]) + Xtd (CAN_IFn_ARB2[14]) + MsgVal (CAN_IFn_APB2[15]) to Message Object. Read Operation: 0 = Arbitration bits unchanged. 1 = Transfer Identifier + Dir + Xtd + MsgVal to IFn Message Buffer Register.
[4]	**Control**	**Control Access Control Bits** Write Operation:

		0 = Control Bits unchanged. 1 = Transfer Control Bits to Message Object. Read Operation: 0 = Control Bits unchanged. 1 = Transfer Control Bits to IFn Message Buffer Register.
[3]	**ClrIntPnd**	**Clear Interrupt Pending Bit** Write Operation: When writing to a Message Object, this bit is ignored. Read Operation: 0 = IntPnd bit (CAN_IFn_MCON[13]) remains unchanged. 1 = Clear IntPnd bit in the Message Object.
[2]	**TxRqst/NewDat**	**Access Transmission Request Bit When Write Operation** 0 = TxRqst bit unchanged. 1 = Set TxRqst bit. **Note:** If a transmission is requested by programming bit TxRqst/NewDat in the IFn Command Mask Register, bit TxRqst in the IFn Message Control Register will be ignored. Access New Data Bit when Read Operation. 0 = NewDat bit remains unchanged. 1 = Clear NewDat bit in the Message Object. **Note:** A read access to a Message Object can be combined with the reset of the control bits IntPnd and NewDat. The values of these bits transferred to the IFn Message Control Register always reflect the status before resetting these bits.
[1]	**DAT_A**	**Access Data Bytes [3:0]** Write Operation: 0 = Data Bytes [3:0] unchanged. 1 = Transfer Data Bytes [3:0] to Message Object. Read Operation: 0 = Data Bytes [3:0] unchanged. 1 = Transfer Data Bytes [3:0] to IFn Message Buffer Register.
[0]	**DAT_B**	**Access Data Bytes [7:4]** Write Operation: 0 = Data Bytes [7:4] unchanged. 1 = Transfer Data Bytes [7:4] to Message Object. Read Operation: 0 = Data Bytes [7:4] unchanged. 1 = Transfer Data Bytes [7:4] to IFn Message Buffer Register.

IFn Mask 1 Register (CAN_IFn_MASK1)

Register	Offset	R/W	Description	Reset Value
CAN_IF1_MASK1	CANx_BA+0x28	R/W	IF1 Mask 1 Register	0x0000_FFFF
CAN_IF2_MASK1	CANx_BA+0x88	R/W	IF2 Mask 1 Register	0x0000_FFFF

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Msk[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Msk[7:0]							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**Msk[15:0]**	**Identifier Mask 15-0** 0 = The corresponding bit in the identifier of the message object cannot inhibit the match in the acceptance filtering. 1 = The corresponding identifier bit is used for acceptance filtering.

IFn Mask 2 Register (CAN_IFn_MASK2)

Register	Offset	R/W	Description	Reset Value
CAN_IF1_MASK2	CANx_BA+0x2C	R/W	IF1 Mask 2 Register	0x0000_FFFF
CAN_IF2_MASK2	CANx_BA+0x8C	R/W	IF2 Mask 2 Register	0x0000_FFFF

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
MXtd	**MDir**	**Reserved**	**Msk[28:24]**				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Msk[23:16]							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15]	**MXtd**	**Mask Extended Identifier** 0 = The extended identifier bit (IDE) has no effect on the acceptance filtering. 1 = The extended identifier bit (IDE) is used for acceptance filtering. **Note:** When 11-bit (“standard”) Identifiers are used for a Message Object, the identifiers of received Data Frames are written into bits ID28 to ID18 (CAN_IFn_ARB2[12:2]). For acceptance filtering, only these bits together with mask bits Msk28 to Msk18 (CAN_IFn_MASK2[12:2]) are considered.
[14]	**MDir**	**Mask Message Direction** 0 = The message direction bit (Dir (CAN_IFn_ARB2[13])) has no effect on the acceptance filtering. 1 = The message direction bit (Dir) is used for acceptance filtering.
[13]	**Reserved**	Reserved.
[12:0]	**Msk[28:16]**	**Identifier Mask 28-16** 0 = The corresponding bit in the identifier of the message object cannot inhibit the match in the acceptance filtering. 1 = The corresponding identifier bit is used for acceptance filtering.

IFn Arbitration 1 Register (CAN_IFn_ARB1)

Register	Offset	R/W	Description	Reset Value
CAN_IF1_ARB1	CANx_BA+0x30	R/W	IF1 Arbitration 1 Register	0x0000_0000
CAN_IF2_ARB1	CANx_BA+0x90	R/W	IF2 Arbitration 1 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ID[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ID[7:0]							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**ID[15:0]**	**Message Identifier 15-0** ID28 - ID0, 29-bit Identifier (“Extended Frame”). ID28 - ID18, 11-bit Identifier (“Standard Frame”)

IFn Arbitration 2 Register (CAN_IFn_ARB2)

Register	Offset	R/W	Description	Reset Value
CAN_IF1_ARB2	CANx_BA+0x34	R/W	IF1 Arbitration 2 Register	0x0000_0000
CAN_IF2_ARB2	CANx_BA+0x94	R/W	IF2 Arbitration 2 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
MsgVal	**Xtd**	**Dir**	**ID[28:24]**				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ID[23:16]							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15]	**MsgVal**	**Message Valid** 0 = The Message Object is ignored by the Message Handler. 1 = The Message Object is configured and should be considered by the Message Handler. **Note:** The application software must reset the MsgVal bit of all unused Messages Objects during the initialization before it resets bit Init (CAN_CON[0]). This bit must also be reset before the identifier Id28-0 (CAN_IFn_ARB1/2), the control bits Xtd (CAN_IFn_ARB2[14]), Dir (CAN_IFn_APB2[13]), or the Data Length Code DLC3-0 (CAN_IFn_MCON[3:0]) are modified, or if the Messages Object is no longer required.
[14]	**Xtd**	**Extended Identifier** 0 = The 11-bit (“standard”) Identifier will be used for this Message Object. 1 = The 29-bit (“extended”) Identifier will be used for this Message Object.
[13]	**Dir**	**Message Direction** 0 = Direction is receive. On TxRqst, a Remote Frame with the identifier of this Message Object is transmitted. On reception of a Data Frame with matching identifier, that message is stored in this Message Object. 1 = Direction is transmit. On TxRqst, the respective Message Object is transmitted as a Data Frame. On reception of a Remote Frame with matching identifier, the TxRqst bit (CAN_IFn_CMASK[2]) of this Message Object is set (if RmtEn (CAN_IFn_MCON[9]) = one).
[12:0]	**ID[28:16]**	**Message Identifier 28-16** ID28 - ID0, 29-bit Identifier (“Extended Frame”). ID28 - ID18, 11-bit Identifier (“Standard Frame”)

IFn Message Control Register (CAN_IFn_MCON)

Register	Offset	R/W	Description	Reset Value
CAN_IF1_MCON	CANx_BA+0x38	R/W	IF1 Message Control Register	0x0000_0000
CAN_IF2_MCON	CANx_BA+0x98	R/W	IF2 Message Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
NewDat	MsgLst	IntPnd	UMask	TxIE	RxIE	RmtEn	TxRqst
7	6	5	4	3	2	1	0
EoB	Reserved			DLC[3:0]			

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15]	**NewDat**	**New Data** 0 = No new data has been written into the data portion of this Message Object by the Message Handler since last time this flag was cleared by the application software. 1 = The Message Handler or the application software has written new data into the data portion of this Message Object.
[14]	**MsgLst**	Message Lost (only valid for Message Objects with direction = receive). 0 = No message lost since last time this bit was reset by the CPU. 1 = The Message Handler stored a new message into this object when NewDat was still set, the CPU has lost a message.
[13]	**IntPnd**	**Interrupt Pending** 0 = This message object is not the source of an interrupt. 1 = This message object is the source of an interrupt. The Interrupt Identifier in the Interrupt Register will point to this message object if there is no other interrupt source with higher priority.
[12]	**UMask**	**Use Acceptance Mask** 0 = Mask ignored. 1 = Use Mask (Msk28-0, MXtd, and MDir) for acceptance filtering. **Note:** If the UMask bit is set to one, the Message Object's mask bits have to be programmed during initialization of the Message Object before MsgVal bit (CAN_IFn_APB2[15]) is set to one.
[11]	**TxIE**	**Transmit Interrupt Enable Bit** 0 = IntPnd (CAN_IFn_MCON[13]) will be left unchanged after the successful transmission of a frame. 1 = IntPnd will be set after a successful transmission of a frame.
[10]	**RxIE**	**Receive Interrupt Enable Bit** 0 = IntPnd (CAN_IFn_MCON[13]) will be left unchanged after a successful reception of a frame.

		1 = IntPnd will be set after a successful reception of a frame.
[9]	**RmtEn**	**Remote Enable Bit** 0 = At the reception of a Remote Frame, TxRqst (CAN_IFn_MCON[8]) is left unchanged. 1 = At the reception of a Remote Frame, TxRqst is set.
[8]	**TxRqst**	**Transmit Request** 0 = This Message Object is not waiting for transmission. 1 = The transmission of this Message Object is requested and is not yet done.
[7]	**EoB**	**End Of Buffer** 0 = Message Object belongs to a FIFO Buffer and is not the last Message Object of that FIFO Buffer. 1 = Single Message Object or last Message Object of a FIFO Buffer. **Note:** This bit is used to concatenate two or more Message Objects (up to 32) to build a FIFO Buffer. For single Message Objects (not belonging to a FIFO Buffer), this bit must always be set to one.
[6:4]	**Reserved**	Reserved.
[3:0]	**DLC**	**Data Length Code** 0-8: Data Frame has 0-8 data bytes. 9-15: Data Frame has 8 data bytes **Note:** The Data Length Code of a Message Object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the Message Handler stores a data frame, it will write the DLC to the value given by the received message. Data 0: 1st data byte of a CAN Data Frame Data 1: 2nd data byte of a CAN Data Frame Data 2: 3rd data byte of a CAN Data Frame Data 3: 4th data byte of a CAN Data Frame Data 4: 5th data byte of a CAN Data Frame Data 5: 6th data byte of a CAN Data Frame Data 6: 7th data byte of a CAN Data Frame Data 7 : 8th data byte of a CAN Data Frame **Note:** The Data 0 Byte is the first data byte shifted into the shift register of the CAN Core during a reception while the Data 7 byte is the last. When the Message Handler stores a Data Frame, it will write all the eight data bytes into a Message Object. I f the Data Length Code is less than 8, the remaining bytes of the Message Object will be overwritten by unspecified values.

IFn Data A1 Register (CAN_IFn_DAT_A1)

Register	Offset	R/W	Description	Reset Value
CAN_IF1_DAT_A1	CANx_BA+0x3C	R/W	IF1 Data A1 Register	0x0000_0000
CAN_IF2_DAT_A1	CANx_BA+0x9C	R/W	IF2 Data A1 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Data1							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Data0							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:8]	**Data1**	**Data Byte 1** 2nd data byte of a CAN Data Frame
[7:0]	**Data0**	**Data Byte 0** 1st data byte of a CAN Data Frame

IFn Data A2 Register (CAN_IFn_DAT_A2)

Register	Offset	R/W	Description	Reset Value
CAN_IF1_DAT_A2	CANx_BA+0x40	R/W	IF1 Data A2 Register	0x0000_0000
CAN_IF2_DAT_A2	CANx_BA+0xA0	R/W	IF2 Data A2 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Data3							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Data2							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:8]	**Data3**	**Data Byte 3** 4th data byte of CAN Data Frame
[7:0]	**Data2**	**Data Byte 2** 3rd data byte of CAN Data Frame

IFn Data B1 Register (CAN_IFn_DAT_B1)

Register	Offset	R/W	Description	Reset Value
CAN_IF1_DAT_B1	CANx_BA+0x44	R/W	IF1 Data B1 Register	0x0000_0000
CAN_IF2_DAT_B1	CANx_BA+0xA4	R/W	IF2 Data B1 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Data5							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Data4							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:8]	**Data5**	**Data Byte 5** 6th data byte of CAN Data Frame
[7:0]	**Data4**	**Data Byte 4** 5th data byte of CAN Data Frame

IFn Data B2 Register (CAN_IFn_DAT_B2)

Register	Offset	R/W	Description	Reset Value
CAN_IF1_DAT_B2	CANx_BA+0x48	R/W	IF1 Data B2 Register	0x0000_0000
CAN_IF2_DAT_B2	CANx_BA+0xA8	R/W	IF2 Data B2 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Data7							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Data6							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:8]	**Data7**	**Data Byte 7** 8th data byte of CAN Data Frame.
[7:0]	**Data6**	**Data Byte 6** 7th data byte of CAN Data Frame.

In a CAN Data Frame, Data0 is the first, Data7 is the last byte to be transmitted or received. In CAN's serial bit stream, the MSB of each byte will be transmitted first.

Message Object in the Message Memory

There are 32 Message Objects in the Message RAM. To avoid conflicts between application software access to the Message RAM and CAN message reception and transmission, the CPU cannot directly access the Message Objects, these accesses are handled through the IF*n* Interface Registers. The following table provides an overview of the structures of a Message Object.

Message Object												
UMask	Msk [28:0]	MXtd	MDir	EoB	NewDat		MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
MsgVal	ID [28:0]	Xtd	Dir	DLC [3:0]	Data0	Data1	Data2	Data3	Data4	Data5	Data6	Data7

Table 6.7-8 Structure of a Message Object in the Message Memory

The Arbitration Registers **ID28-0**, **Xtd**, and **Dir** are used to define the identifier and type of outgoing messages and are used (together with the mask registers **Msk28-0**, **MXtd**, and **MDir**) for acceptance filtering of incoming messages. A received message is stored in the valid Message Object with matching identifier and Direction = *receive* (Data Frame) or Direction = *transmit* (Remote Frame). Extended frames can be stored only in Message Objects with **Xtd** = one, standard frames in Message Objects with **Xtd** = zero. If a received message (Data Frame or Remote Frame) matches with more than one valid Message Object, it is stored into that with the lowest message number.

Message Handler Registers

All Message Handler registers are read only. Their contents (**TxRqst**, **NewDat**, **IntPnd**, and **MsgVal** bits of each Message Object and the Interrupt Identifier) are status information provided by the Message Handler FSM.

Transmission Request Register 1 (CAN_TXREQ1)

These registers hold the **TxRqst** bits of the 32 Message Objects. By reading the **TxRqst** bits, the software can check which Message Object in a Transmission Request is pending. The **TxRqst** bit of a specific Message Object can be set/reset by the application software through the IFn Message Interface Registers or by the Message Handler after reception of a Remote Frame or after a successful transmission.

Register	Offset	R/W	Description	Reset Value
CAN_TXREQ1	CANx_BA+0x100	R	Transmission Request Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TxRqst 16-9							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TxRqst 8-1							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**TxRqst 16-1**	**Transmission Request Bits 16-1 (Of All Message Objects)** 0 = This Message Object is not waiting for transmission. 1 = The transmission of this Message Object is requested and is not yet done. These bits are read only.

Transmission Request Register 2 (CAN_TXREQ2)

Register	Offset	R/W	Description	Reset Value
CAN_TXREQ2	CANx_BA+0x104	R	Transmission Request Register 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
TxRqst32-25							
7	6	5	4	3	2	1	0
TxRqst24-17							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**TxRqst 32-17**	**Transmission Request Bits 32-17 (Of All Message Objects)** 0 = This Message Object is not waiting for transmission. 1 = The transmission of this Message Object is requested and is not yet done. These bits are read only.

New Data Register 1 (CAN_NDAT1)

These registers hold the **NewDat** bits of the 32 Message Objects. By reading out the **NewDat** bits, the software can check for which Message Object the data portion was updated. The **NewDat** bit of a specific Message Object can be set/reset by the software through the IFn Message Interface Registers or by the Message Handler after reception of a Data Frame or after a successful transmission.

Register	Offset	R/W	Description	Reset Value
CAN_NDAT1	CANx_BA+0x120	R	New Data Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
NewData16-9							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NewData 8-1							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**NewData16-1**	**New Data Bits 16-1 (Of All Message Objects)** 0 = No new data has been written into the data portion of this Message Object by the Message Handler since the last time this flag was cleared by the application software. 1 = The Message Handler or the application software has written new data into the data portion of this Message Object.

New Data Register 2 (CAN_NDAT2)

Register	Offset	R/W	Description	Reset Value
CAN_NDAT2	CANx_BA+0x124	R	New Data Register 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
NewData 32-25							
7	6	5	4	3	2	1	0
NewData 24-17							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**NewData 32-17**	**New Data Bits 32-17 (Of All Message Objects)** 0 = No new data has been written into the data portion of this Message Object by the Message Handler since the last time this flag was cleared by the application software. 1 = The Message Handler or the application software has written new data into the data portion of this Message Object.

Interrupt Pending Register 1 (CAN_IPND1)

These registers contain the **IntPnd** bits of the 32 Message Objects. By reading the **IntPnd** bits, the software can check for which Message Object an interrupt is pending. The **IntPnd** bit of a specific Message Object can be set/reset by the application software through the IFn Message Interface Registers or by the Message Handler after reception or after a successful transmission of a frame. This will also affect the value of **IntId** in the Interrupt Register.

Register	Offset	R/W	Description	Reset Value
CAN_IPND1	CANx_BA+0x140	R	Interrupt Pending Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
IntPnd16-9							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IntPnd 8-1							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**IntPnd16-1**	**Interrupt Pending Bits 16-1 (Of All Message Objects)** 0 = This message object is not the source of an interrupt. 1 = This message object is the source of an interrupt.

Interrupt Pending Register 2 (CAN_IPND2)

Register	Offset	R/W	Description	Reset Value
CAN_IPND2	CANx_BA+0x144	R	Interrupt Pending Register 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
IntPnd 32-25							
7	6	5	4	3	2	1	0
IntPnd 24-17							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**IntPnd 32-17**	**Interrupt Pending Bits 32-17(Of All Message Objects)** 0 = This message object is not the source of an interrupt. 1 = This message object is the source of an interrupt.

Message Valid Register 1 (CAN_MVLD1)

These registers hold the **MsgVal** bits of the 32 Message Objects. By reading the **MsgVal** bits, the application software can check which Message Object is valid. The **MsgVal** bit of a specific Message Object can be set/reset by the application software via the IFn Message Interface Registers.

Register	Offset	R/W	Description	Reset Value
CAN_MVLD1	CANx_BA+0x160	R	Message Valid Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
MsgVal 16- 9							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MsgVal 8-1							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**MsgVal 16-1**	**Message Valid Bits 16-1 (Of All Message Objects) (Read Only)** 0 = This Message Object is ignored by the Message Handler. 1 = This Message Object is configured and should be considered by the Message Handler. Ex. CAN_MVLD1[0] means Message object No.1 is valid or not. If CAN_MVLD1[0] is set, message object No.1 is configured.

Message Valid Register 2 (CAN_MVLD2)

Register	Offset	R/W	Description	Reset Value
CAN_MVLD2	CANx_BA+0x164	R	Message Valid Register 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
MsgVal 32-25							
7	6	5	4	3	2	1	0
MsgVal 24-17							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**MsgVal 32-17**	**Message Valid Bits 32-17 (Of All Message Objects) (Read Only)** 0 = This Message Object is ignored by the Message Handler. 1 = This Message Object is configured and should be considered by the Message Handler. Ex.CAN_MVLD2[15] means Message object No.32 is valid or not. If CAN_MVLD2[15] is set, message object No.32 is configured.

Wake-up Enable Control Register (CAN_WU_EN)

Register	Offset	R/W	Description	Reset Value
CAN_WU_EN	CANx_BA+0x168	R/W	Wake-up Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							WAKUP_EN

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**WAKUP_EN**	**Wake-Up Enable Bit** 0 = The wake-up function Disabled. 1 = The wake-up function Enabled. **Note:** User can wake-up system when there is a falling edge in the CAN_Rx pin..

Wake-up Status Register (CAN_WU_STATUS)

Register	Offset	R/W	Description	Reset Value
CAN_WU_STATUS	CANx_BA+0x16C	R/W	Wake-up Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							WAKUP_STS

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**WAKUP_STS**	**Wake-Up Status** 0 = No wake-up event occurred. 1 = Wake-up event occurred. **Note:** This bit can be cleared by writing '0'.

6.8 CRC Controller

6.8.1 Overview

The Cyclic Redundancy Check (CRC) generator can perform CRC calculation with programmable polynomial settings.

6.8.2 Features

- Supports four common polynomials CRC-CCITT, CRC-8, CRC-16, and CRC-32
  - CRC-CCITT: $X^{16} + X^{12} + X^5 + 1$
  - CRC-8: $X^8 + X^2 + X + 1$
  - CRC-16: $X^{16} + X^{15} + X^2 + 1$
  - CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- Programmable seed value
- Supports programmable order reverse setting for input data and CRC checksum
- Supports programmable 1's complement setting for input data and CRC checksum
- Supports 8/16/32-bit of data width
  - 8-bit write mode: 1-AHB clock cycle operation
  - 16-bit write mode: 2-AHB clock cycle operation
  - 32-bit write mode: 4-AHB clock cycle operation
- Supports using PDMA to write data to perform CRC operation

6.8.3 Block Diagram

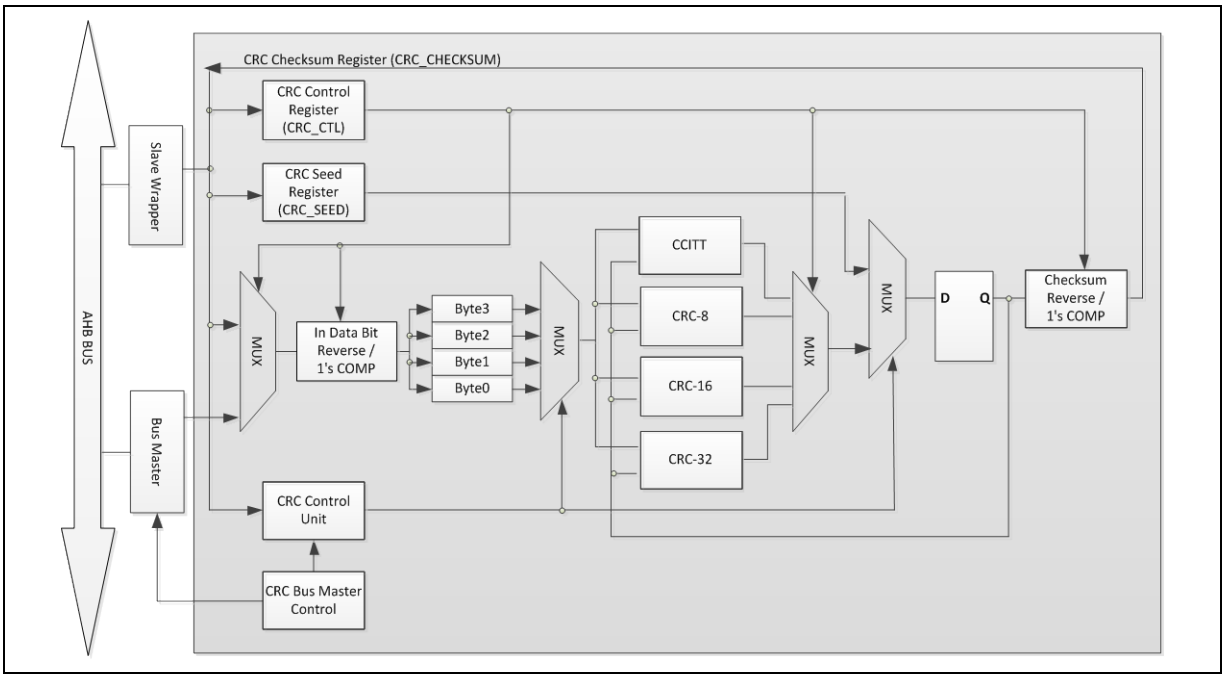


Figure 6.8-1 CRC Generator Block Diagram

6.8.4 Basic Configuration

The CRC peripheral clock is enabled in CRCCKEN (CLK_AHBCLK[7]). After CRC is setting, user can start to perform CRC calculate by control CRC's registers.

6.8.5 Functional Description

CRC generator can perform CRC calculation with programmable polynomial settings. The operation polynomial includes CRC-CCITT, CRC-8, CRC-16 and CRC-32; User can choose the CRC operation polynomial mode by setting CRCMODE[1:0] (CRC_CTL[31:30] CRC Polynomial Mode).

The CRC generator supports CPU mode only. The following is a program sequence example.

1. Enable CRC generator by setting CRCEN (CRC_CTL[0] CRC Channel Enable Control).
2. Initial setting for CRC calculation.
   - Configure 1's complement for CRC checksum by setting CHKSFMT (CRC_CTL[27] Checksum Complement).
   - Configure bit order reverse for CRC checksum by setting CHKSREV (CRC_CTL[25] Checksum Bit Order Reverse).
   - Configure 1's complement for CRC write data by setting DATFMT (CRC_CTL[26] Write Data Complement).
   - Configure bit order reverse for CRC write data by setting DATREV (CRC_CTL[24] Write Data Bit Order Reverse).
3. Perform CRC reset to load the initial seed value to CRC circuit by setting CRCRST (CRC_CTL[1] CRC Engine Reset).
4. Write data to CRC_DAT register to calculate CRC checksum.
5. Get the CRC checksum result by reading CRC_CHECKSUM register.

6.8.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
CRC Base Address: **CRC_BA = 0x4003_1000**				
CRC_CTL	CRC_BA+0x00	R/W	CRC Control Register	0x2000_0000
CRC_DAT	CRC_BA+0x04	R/W	CRC Write Data Register	0x0000_0000
CRC_SEED	CRC_BA+0x08	R/W	CRC Seed Register	0xFFFF_FFFF
CRC_CHECKSUM	CRC_BA+0x0C	R	CRC Checksum Register	0x0000_0000

6.8.7 Register Description

CRC Control Register (CRC_CTL)

Register	Offset	R/W	Description	Reset Value
CRC_CTL	CRC_BA+0x00	R/W	CRC Control Register	0x2000_0000

31	30	29	28	27	26	25	24
CRCMODE		DATLEN		CHKSFMT	DATFMT	CHKSREV	DATREV
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						CRCRST	CRCEN

Bits	Description	
[31:30]	**CRCMODE**	**CRC Polynomial Mode Selection** 00 = CRC-CCITT Polynomial mode. 01 = CRC-8 Polynomial mode. 10 = CRC-16 Polynomial mode. 11 = CRC-32 Polynomial mode.
[29:28]	**DATLEN**	**CPU Write Data Length** This field indicates the write data length. 00 = Data length is 8-bit mode. 01 = Data length is 16-bit mode. 1x = Data length is 32-bit mode. **Note:** When the data length is 8-bit mode, the valid data is DATA [7:0]; if the data length is 16-bit mode, the valid data is DATA [15:0].
[27]	**CHKSFMT**	**Checksum Complement** 0 = No bit order reverse for CRC checksum. 1 = 1's complement for CRC checksum.
[26]	**DATFMT**	**Write Data Complement** 0 = No bit order reversed for CRC write data in. 1 = 1's complement for CRC write data in.
[25]	**CHKSREV**	**Checksum Reverse** 0 = No bit order reverse for CRC checksum. 1 = Bit order reverse for CRC checksum. **Note:** If the checksum data is 0XDD7B0F2E, the bit order reversed for CRC checksum is 0x74F0DEBB.

[24]	**DATREV**	**Write Data Order Reverse** 0 = No bit order reversed for CRC write data in. 1 = Bit order reversed for CRC write data in (per byte). **Note:** If the write data is 0xAABBCCDD, the bit order reverse for CRC write data in is 0x55DD33BB
[23:2]	**Reserved**	Reserved.
[1]	**CRCRST**	**CRC Engine Reset** 0 = No effect. 1 = Reset the internal CRC state machine and internal buffer. The contents of control register will not be cleared. This bit will automatically be cleared after few clock cycles. **Note:** Setting this bit will reload the initial seed value.
[0]	**CRCEN**	**CRC Channel Enable Bit** 0 = CRC function Disabled. 1 = CRC function Enabled.

CRC Write Data Register (CRC_DAT)

Register	Offset	R/W	Description	Reset Value
CRC_DAT	CRC_BA+0x04	R/W	CRC Write Data Register	0x0000_0000

31	30	29	28	27	26	25	24
DATA [31:24]							
23	22	21	20	19	18	17	16
DATA [23:16]							
15	14	13	12	11	10	9	8
DATA [15:8]							
7	6	5	4	3	2	1	0
DATA [7:0]							

Bits	Description	
[31:0]	DATA	**CRC Write Data Bits** Software can write data to this field to perform CRC operation, or uses PDMA function to get the data from memory **Note1:** The CRC_CTL [DATFMT] and CRC_CTL [DATREV] bit setting will affect this field; for example, if DATREV = 1, if the write data in DATA register is 0xAABBCCDD, the read data from DATA register will be 0x55DD33BB.

CRC Seed Register (CRC_SEED)

Register	Offset	R/W	Description	Reset Value
CRC_SEED	CRC_BA+0x08	R/W	CRC Seed Register	0xFFFF_FFFF

31	30	29	28	27	26	25	24
SEED [31:24]							
23	22	21	20	19	18	17	16
SEED [23:16]							
15	14	13	12	11	10	9	8
SEED [15:8]							
7	6	5	4	3	2	1	0
SEED [7:0]							

Bits	Description	
[31:0]	SEED	**CRC Seed Bits** This field indicates the CRC seed value.

CRC Checksum Register (CRC_CHECKSUM)

Register	Offset	R/W	Description	Reset Value
CRC_CHECKSUM	CRC_BA+0x0C	R	CRC Checksum Register	0x0000_0000

31	30	29	28	27	26	25	24
CHECKSUM [31:24]							
23	22	21	20	19	18	17	16
CHECKSUM [23:16]							
15	14	13	12	11	10	9	8
CHECKSUM [15:8]							
7	6	5	4	3	2	1	0
CHECKSUM [7:0]							

Bits	Description	
[31:0]	CHECKSUM	**CRC Checksum Bits** This field indicates the CRC checksum.

6.9 Cryptographic Accelerator

6.9.1 Overview

The Crypto (Cryptographic Accelerator) includes a secure pseudo random number generator (PRNG) core and supports AES, DES/TDES, and SHA algorithms.

The PRNG core supports 64 bits, 128 bits, 192 bits, and 256 bits random number generation.

The AES accelerator is an implementation fully compliant with the AES (Advance Encryption Standard) encryption and decryption algorithm. The AES accelerator supports ECB, CBC, CFB, OFB, CTR, CBC-CS1, CBC-CS2, and CBC-CS3 mode.

The DES/TDES accelerator is an implementation fully compliant with the DES and Triple DES encryption/decryption algorithm. The DES/TDES accelerator supports ECB, CBC, CFB, OFB, and CTR mode.

The SHA accelerator is an implementation fully compliant with the SHA-160, SHA-224 and SHA-256.

6.9.2 Features

- PRNG
  - Supports 64 bits, 128 bits , 192 bits, and 256 bits random number generation
- AES
  - Supports FIPS NIST 197
  - Supports SP800-38A and addendum
  - Supports 128, 192, and 256 bits key
  - Supports both encryption and decryption
  - Supports ECB, CBC, CFB, OFB , CTR, CBC-CS1, CBC-CS2, and CBC-CS3 mode
  - Supports key expander
- DES
  - Supports FIPS 46-3
  - Supports both encryption and decryption
  - Supports ECB, CBC, CFB, OFB, and CTR mode
- TDES
  - Supports FIPS NIST 800-67
  - Implemented according to the X9.52 standard
  - Supports two keys or three keys mode
  - Supports both encryption and decryption
  - Supports ECB, CBC, CFB, OFB, and CTR mode
- SHA
  - Supports FIPS NIST 180, 180-2
  - Supports SHA-160, SHA-224, and SHA-256

6.9.3 Block Diagram

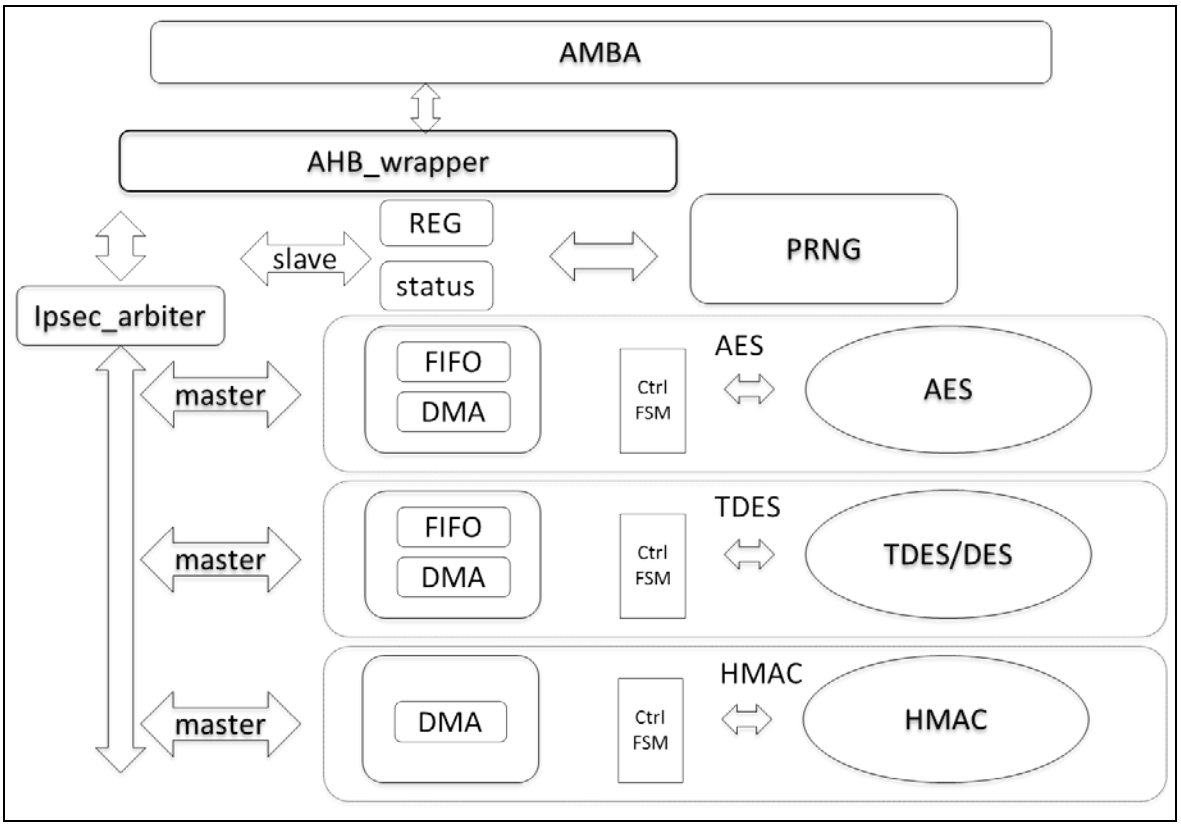


Figure 6.9-1 Cryptographic Accelerator Block Diagram

6.9.4 Functional Description

The cryptographic accelerator includes a secure pseudo random number generator (PRNG) core and supports AES, DES/TDES, SHA algorithms. The accelerator can be used in different data security applications, such as secure communications that need cryptographic protection and integrity.

1. The PRNG core supports 64 bits, 128 bits, 192 bits, and 256 bits random number generation configured by KEYSZ.
2. The AES accelerator is a fully compliant implementation of the AES (Advance Encryption Standard) encryption and decryption algorithm. The AES accelerator supports ECB, CBC, CFB, OFB, CTR, CBC-CS1, CBC-CS2, and CBC-CS3 mode. The AES accelerator provides the DMA function to reduce the CPU intervention, and supports three burst lengths, sixteen-words, eight-words, and four-words.
3. The DES/TDES accelerator is a fully compliant implementation of the DES and Triple DES encryption/decryption algorithm. The DES/TDES accelerator supports ECB, CBC, CFB, OFB, and CTR mode. The DES/TDES accelerator also supports the DMA function to reduce the CPU intervention. Only two burst lengths, four words and eight words, are supported.
4. The SHA accelerator is a fully compliant implementation of the SHA-160, SHA-224, and SHA-256. The SHA accelerator also supports the DMA function to reduce the CPU intervention. It supports three burst lengths, sixteen-words, eight-words, and four-words.

Software can control the data flow by enabling the CRPT_INTEN, and monitor the accelerator status by checking the CRPT_INTSTS.

The cryptographic accelerator supports the following features to enhance the performance.

1. **DMA mode:** Once DMA source address register, destination address register, and byte count register are configurated by CPU, moving data from and to accelerator is done by DMA logic totally. This mode can off-load the loading from the CPU. The cryptographic accelerator embeds four hardware DMA channels for AES engine, four hardware DMA channels for DES/TDES engine, and one hardware DMA channel for SHA engine.
2. **DMA Cascade mode:** In the case that the data SRAM resource is tight, or another peripheral is scheduled to switch, the data source or sink needs an update, while the setting for the accelerator operation is planned to be kept. In this mode, software can update DMA source address register, destination address register, and byte count register during a cascade operation, without finishing the accelerator operation.
3. **Non-DMA mode:** In the case that the input data is small in size, DMA mode is not preferred. This mode can reduce the processing time for the accelerator, since no DMA related register needs a configuration, and no latency in DMA logic is introduced. Input data was feeding to cryptographic engine via writing to data input register.
4. **Channel Expansion mode:** In this mode, several virtual channels in one of four DMA channels are feasible in AES or DES/TDES mode. The total channel number can exceed the limit of four DMA channels. The intermediate data from feedback registers (CRPT_AES_FDBCKx, CRPT_TDES_FDBCKH, and CRPT_TDES_FDBCKL) should be stored temporarily in data SRAM. And switch to another configuration setting of accelerator operation that includes operational mode, encryption/decryption, key, key size, IV, and other parameters. Once switching back, the intermediate data from feedback registers should be written to initial vectors (CRPT_AESn_IVx, CRPT_TDESx_IVH, and CRPT_TDESx_IVL) for the accelerator to continue the operation with the original configuration setting. Note that, in ECB mode, there is no need to move the intermediate data from feedback registers to IV.

6.9.4.1 PRNG (Pseudo Random Number Generator)

The PRNG block diagram is depicted below. The core supports 64 bits, 128 bits, 192 bits, and 256 bits random number generation configured by KEYSZ.

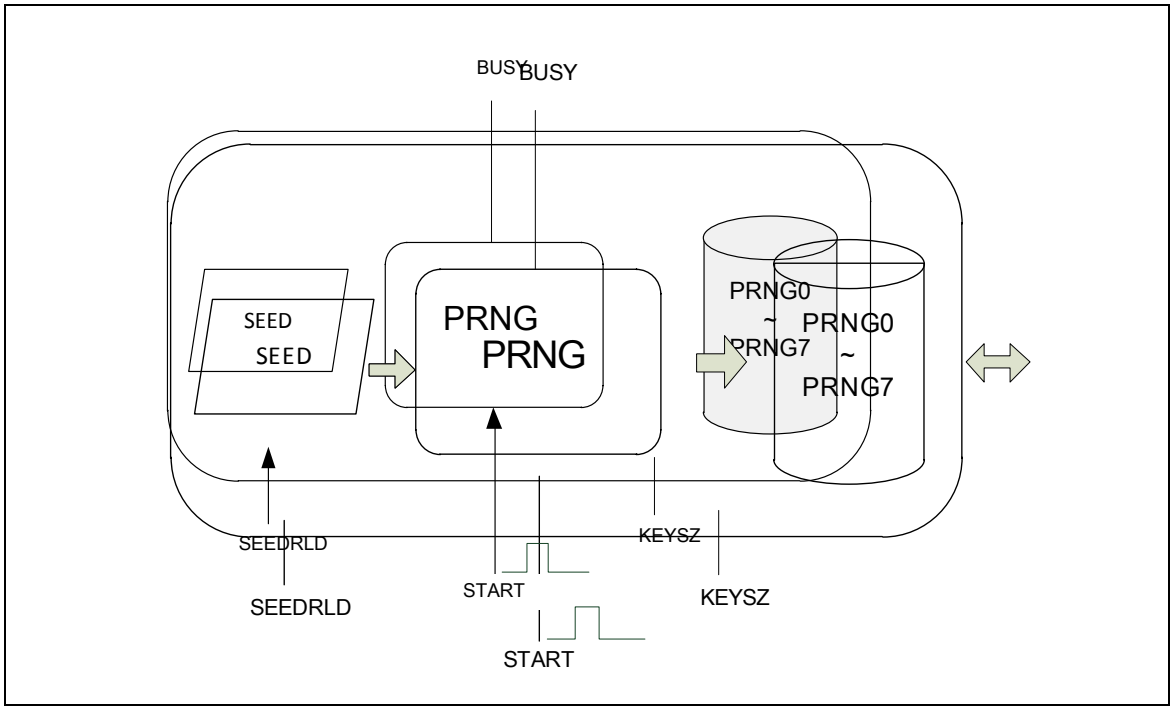


Figure 6.9-2 PRNG Function Diagram

The programming steps to get the pseudo random number are depicted below.

1. Check the BUSY(CRPT_PRNG_CTL[8]) until it comes to 0.
2. Initialize PRNG parameters. Configure KEYSZ(CRPT_PRNG_CTL[3:2]), and write a random seed to PRNG _SEED. Note that CRPT_PRNG_SEED should be initialized since it's not initialized as the chip powers up.
3. Configure PRNG control register CRPT_PRNG_CTL.
4. Software checks BUSY(CRPT_PRNG_CTL[8]) until it comes to 0, or waits for the PRNG done interrupt (must enable the corresponding interrupt enable register). Then software can read the output random numbers from CRPT_PRNG_KEY0 ~ CRPT_PRNG_KEY7.

6.9.4.2 AES (Advanced Encryption Standard)

Electronic Codebook Mode:

The Electronic Codebook (ECB) mode is a confidentiality mode that features the assignment of a fixed ciphertext block to each plaintext block, for a given key. It's analogous to the assignment of code words in a codebook.

In ECB encryption, each block of the plaintext is applied to the forward cipher function $CIPH_k$ directly and independently. The resulting sequence of output blocks is the ciphertext. In ECB dencryption, each block of the ciphertext is applied to the inverse cipher function $CIPH^{-1}_k$ directly and independently. The resulting sequence of output blocks is the plaintext.

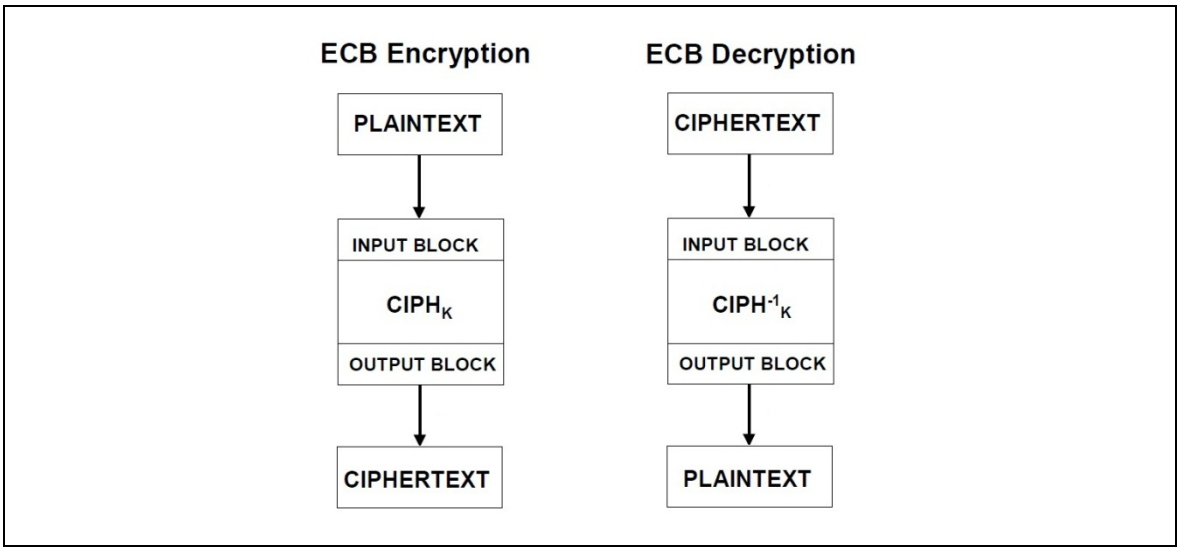


Figure 6.9-3 Electronic Codebook Mode

In ECB mode, any given plaintext block always gets encrypted to the same ciphertext block under a given key. If this property is undesirable in a particular application, the ECB mode should not be used.

Cipher Block Chaining Mode:

The Cipher Block Chaining (CBC) mode is a confidentiality mode whose encryption process features the combining chaining of the plaintext blocks with the previous ciphertext blocks. The CBC mode requires an initialization vector (IV) to combine with the first plaintext block. The IV does not need to be secret, but it must be unpredictable.

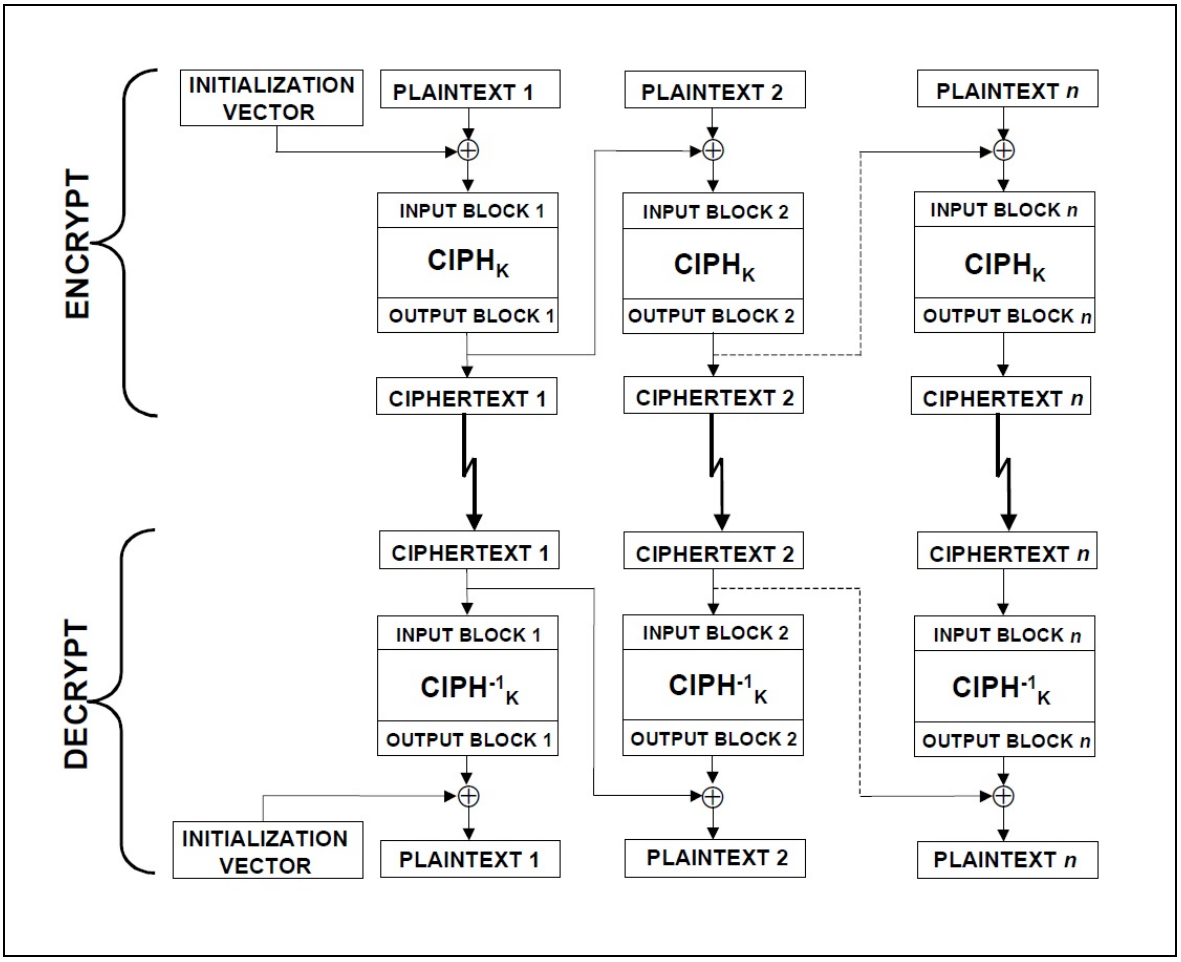


Figure 6.9-4 Cipher Block Chaining Mode

Cipher Feedback Mode (CFB):

The Cipher Feedback (CFB) mode is a confidentiality mode that features the feedback of successive ciphertext segments into the input blocks of the forward cipher to generate output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa. The CFB mode requires an IV as the initial input block. The IV need not be secret, but it must be unpredictable.

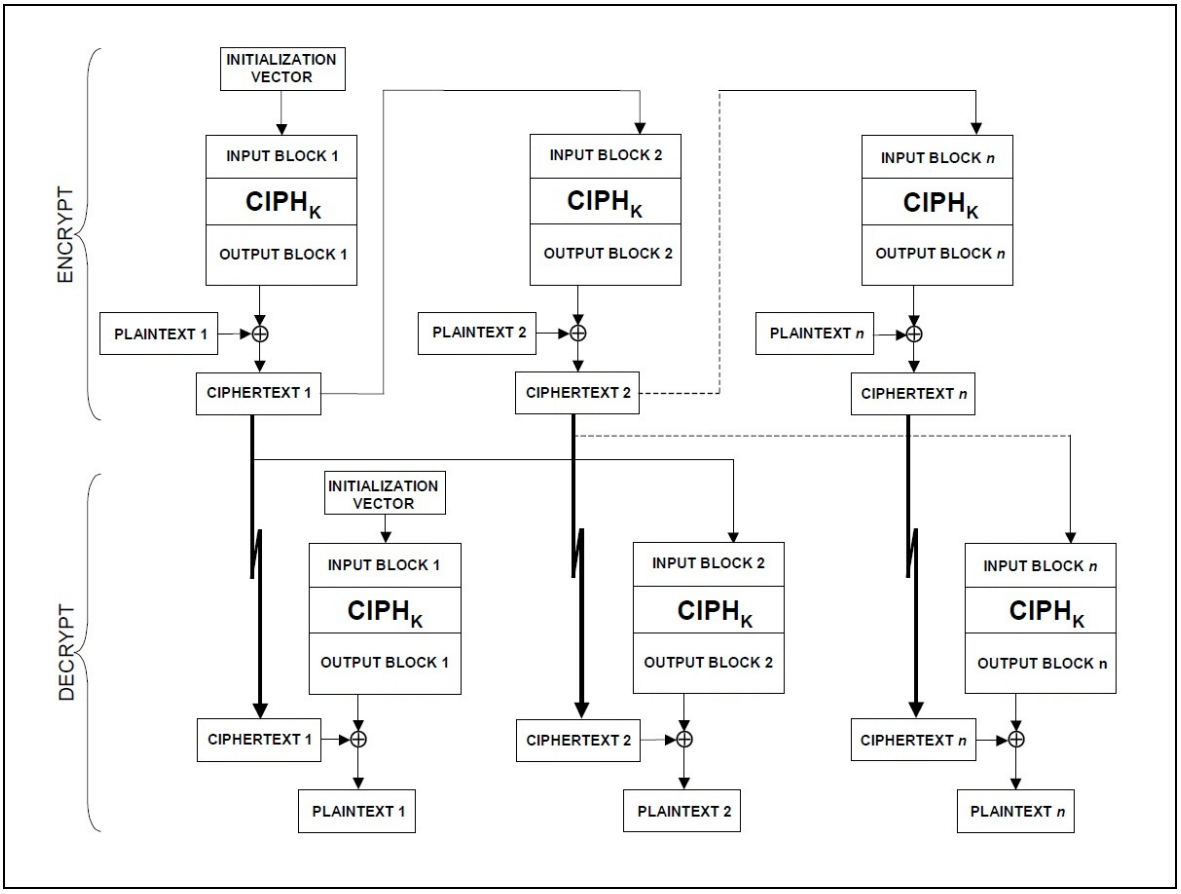


Figure 6.9-5 Cipher Feedback Mode

Output Feedback Mode:

The Output Feedback (OFB) mode is a confidentiality mode that features the iteration of the forward cipher on an IV to generate a sequence of output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa. The OFB mode requires that the IV is a nonce, i.e., the IV must be unique for each execution of the mode under the given key.

The OFB mode requires a unique IV for every message that is ever encrypted under the given key. If, contrary to this requirement, the same IV is used for the encryption of more than one message, then the confidentiality of those messages may be compromised. Confidentiality may be similarly be compromised if any of the input blocks to the forward cipher function for the encryption of a message is designated as the IV for the encryption of another message under the given key.

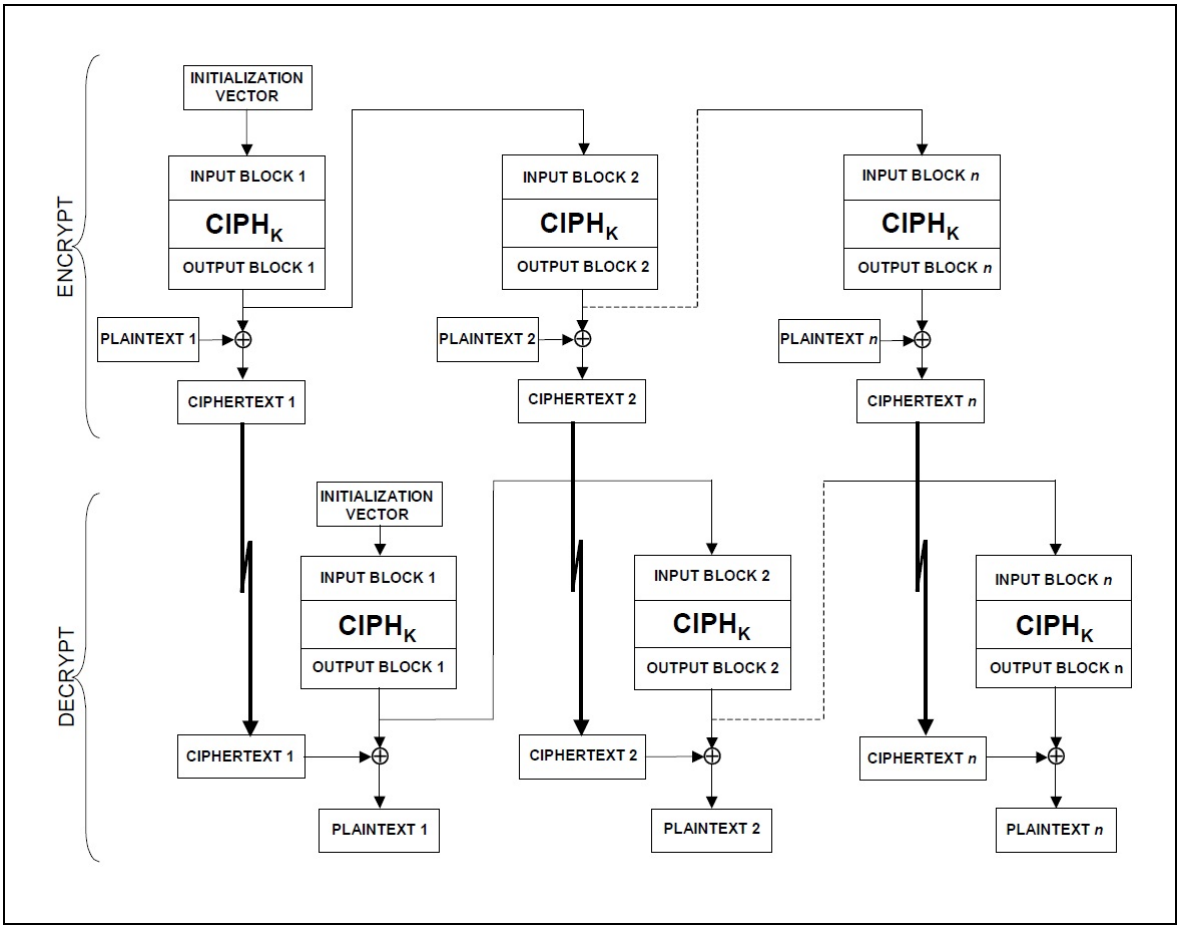


Figure 6.9-6 Output Feedback Mode

Counter Mode (CTR):

The Counter (CTR) mode is a confidentiality mode that features the application of the forward cipher to a set of input blocks, called counters, to produce a sequence of output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa. The sequence of counters must have the property that each block in the sequence is different from every other block. This condition is not restricted to a single message: across all of the messages that are encrypted under the given key, all of the counters must be distinct.

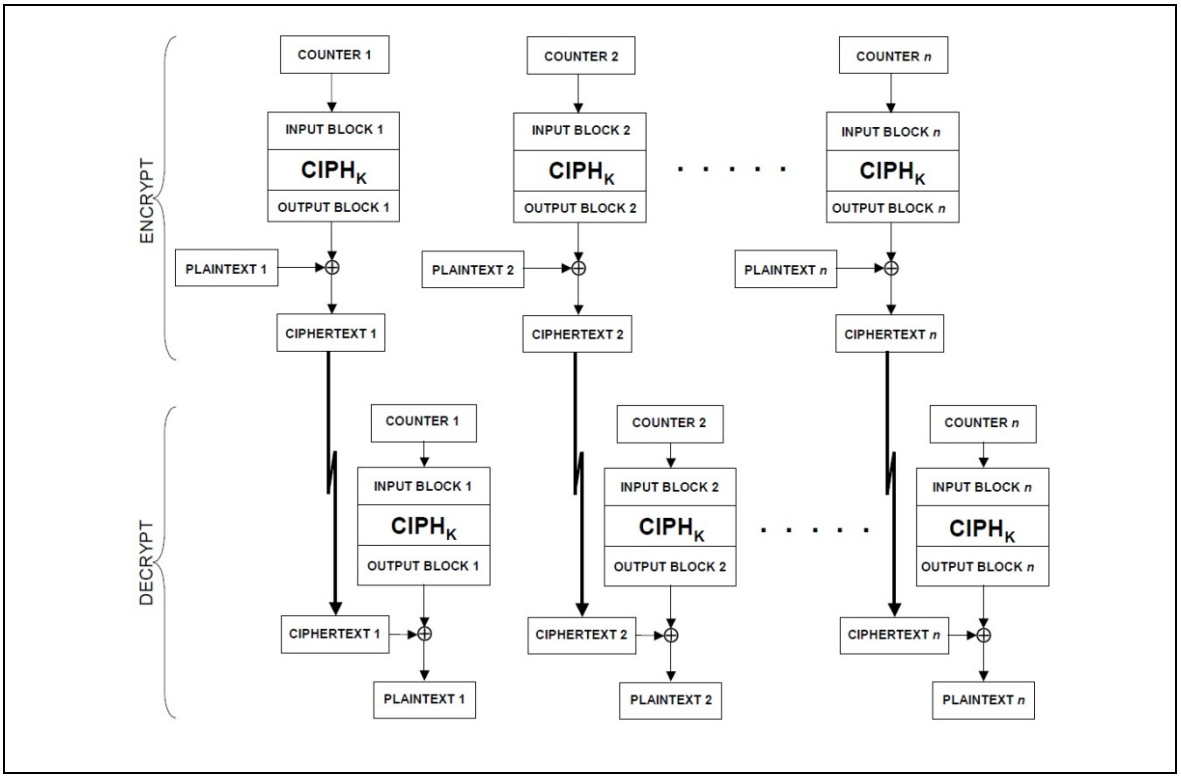


Figure 6.9-7 Counter Mode

CBC Ciphertext-Stealing 1 Mode (CBC-CS1):

The figure below illustrates the CBC-CS1-Encrypt algorithm for the case that P_n^* is a partial block. The cryptographic accelerator would append P_n^* with '0' to form a complete block P_n.

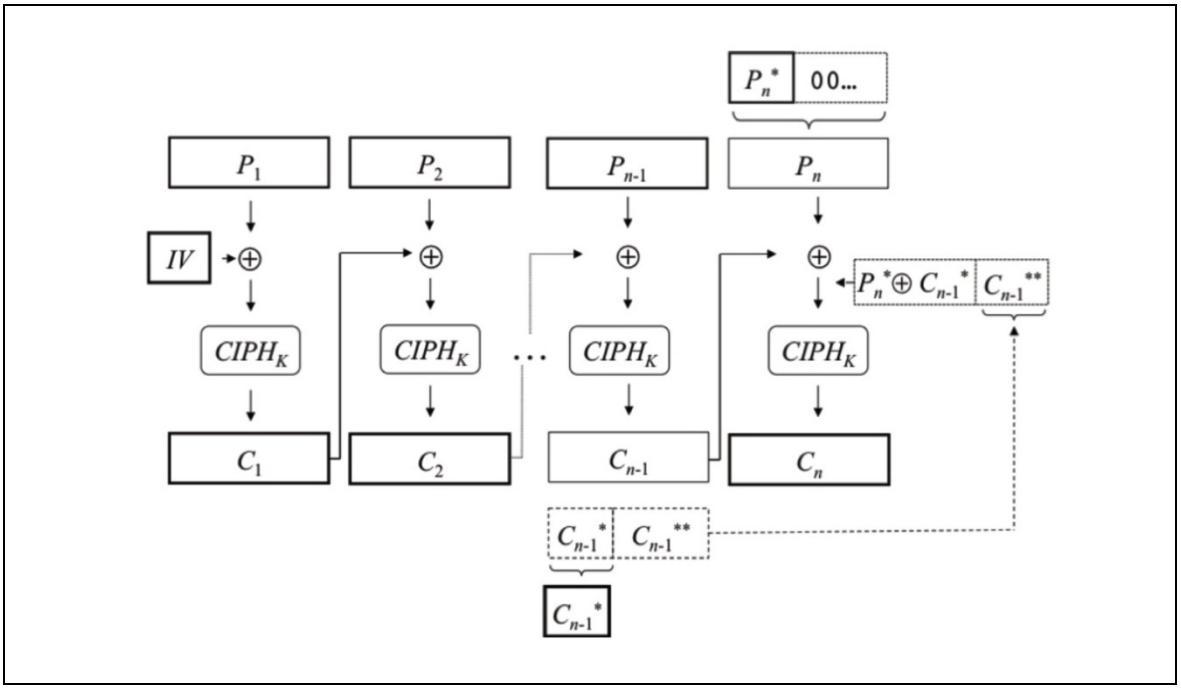


Figure 6.9-8 CBC-CS1 Encryption

The figure below illustrates the CBC-CS1-Decrypt algorithm for the case that C_{n-1}^* is a partial block.

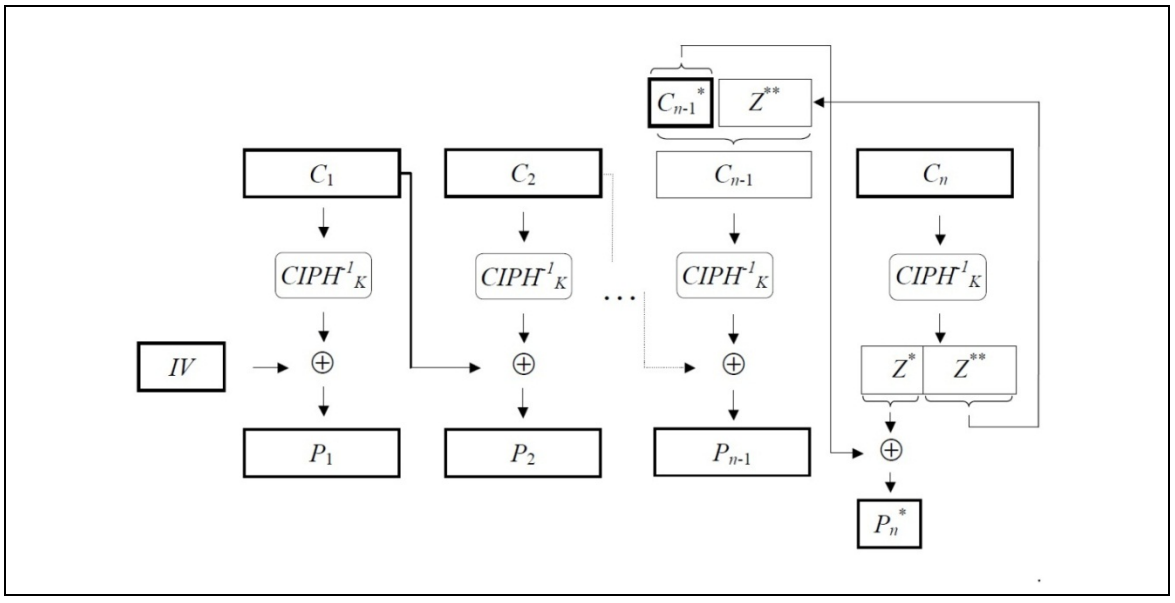


Figure 6.9-9 CBC-CS1 Decryption

CBC Ciphertext-Stealing 2 Mode (CBC-CS2):

When P_n^* is a partial block, then CBC-CS2-Encrypt and CBC-CS1-Encrypt differ only in the ordering of C_{n-1}^* and C_n..

CBC Ciphertext-Stealing 3 Mode (CBC-CS3):

C_{n-1}^* and C_n are unconditionally swapped, i.e., even when C_{n-1}^* is a complete block; therefore, CBC-CS3 is not strictly an extension of CBC mode. In the other case, i.e., when C_{n-1}^* is a nonempty partial block, CBC-CS3-Encrypt is equivalent to CBC-CS2-Encrypt.

Refer to the following programming steps for how to program the AES related registers.

AES DMA mode programming flow:

1. Write 1 to AESIEN(CRPT_INTEN [0]) to enable AES interrupt.
2. Select one from four DMA channels.
3. Program AES key to registers CRPT_AESn_KEY0 ~ CRPT_ AESn_KEY7. (where n is the selected channel number)
4. Program initial vectors to registers CRPT_AESn_IV0 ~ CRPT_AESn_IV3.
5. Program DMA source address to register CRPT_AESn_SADR.
6. Program DMA destination address to register CRPT_AESn_DADR.
7. Program DMA byte count to register CRPT_AESn_CNT.
8. Configure AES control register CRPT_AES_CTL for channel selection, encryption/decryption, operational mode, DMA mode, key size, and DMA input/output swap.
9. Write input data to DMA source address with selected DMA byte count.
10. Write 1 to START(CRPT_AES_CTL[0]) to start AES encryption/decryption.
11. Waits for the AES interrupt flag AESIF(CRPT_INTSTS[0]) be set.
12. Read output data from DMA destination address with selected DMA byte count.
13. Repeat step 9 to step 12 until all data processed.

AES Non-DMA mode programming flow:

1. Write 1 to AESIEN(CRPT_INTEN[0]) to enable AES interrupt.
2. Program AES key to register CRPT_AESn_KEY0 ~ CRPT_ AESn_KEY7. (where n is the selected channel number)
3. Program initial vectors to register CRPT_AESn_IV0 ~ CRPT_AESn_IV3.
4. Configure AES control register (CRPT_AES_CTL) for channel select, encryption/decryption, operational mode, and key size.
5. Write 1 to START(CRPT_AES_CTL[0]) to start AES encryption/decryption.
6. Polling INBUFFULL(CRPT_AES_STS[9]) and OUTBUFEMPTY(CRPT_AES_STS[16]). If INBUFFULL(CRPT_AES_STS[9]) is 0, write 32 bits input data to CRPT_AES_DATIN. If OUTBUFEMPTY(CRPT_AES_STS[16]) is 0, read 32 bits data from CRPT_AES_DATOUT.
7. Repeat step 6 until 128 bits data (16 bytes) are written to and read from AES engine.
8. Write 1 to DMALAST(CRPT_AES_CTL[5]).
9. Repeat steps 6 to step 8 until all data processed.

6.9.4.3 DES/TDES (Data Encryption Standard / Triple DES)

FIPS 46-3 specifies two cryptographic algorithms, the Data Encryption Standard(DES) and the Triple Data Encryption Algorithm (TDEA). The cryptographic accelerator supports FIPS 46-3, both encryption and decryption, and ECB, CBC, CFB, OFB and CTR modes.

TDES DMA mode programming flow:

1. Write 1 to TDESIEN(CRPT_INTEN[8]) to enable TDES interrupt.

2. Check the TDES engine is in idle state, i.e., BUSY(CRPT_TDES_STS[0]) is 0.
3. Program TDES key to registers CRPT_TDESn_KEY1H, CRPT_TDESn_KEY1L, CRPT_TDESn_KEY2H, CRPT_TDESn_KEY2L, CRPT_TDESn_KEY3H, and CRPT_TDESn_KEY3L. (where n is the selected channel number)
4. Program initial vector to registers CRPT_TDESn_IVH and CRPT_TDESn_IVL.
5. Program DMA source address to register CRPT_TDESn_SADR.
6. Program DMA destination address to register CRPT_TDESn_DADR.
7. Program DMA byte count to register CRPT_TDESn_CNT.
8. Configure TDES control register CRPT_TDES_CTL for channel selection, encryption/decryption, operational mode, DMA mode, TDES keys, and DMA input/output swap.
9. Write input data to DMA source address with selected DMA byte count.
10. Write 1 to START(CRPT_TDES_CTL[0]) to start TDES encryption/decryption.
11. Waits for the TDES interrupt flag TDESIF(CRPT_INTSTS[8]) be set.
12. Read output data from DMA destination address with selected DMA byte count.
13. Repeat step 9 to step 12 until all data processed.

TDES Non-DMA mode programming flow:

1. Write 1 to TDESIEN(CRPT_INTEN[8]) to enable TDES interrupt.
2. Check the TDES engine is in idle state, i.e., BUSY(CRPT_TDES_STS[0]) is 0.
3. Program TDES key to registers CRPT_TDESn_KEY1H, CRPT_TDESn_KEY1L, CRPT_TDESn_KEY2H, CRPT_TDESn_KEY2L, CRPT_TDESn_KEY3H, and CRPT_TDESn_KEY3L. (where n is the selected channel number)
4. Program initial vector to registers CRPT_TDESn_IVH and CRPT_TDESn_IVL.
5. Configure TDES control register CRPT_TDES_CTL for channel selection, encryption/decryption, operational mode, and TDES keys.
6. Write 1 to START(CRPT_TDES_CTL[0]) to start TDES encryption/decryption.
7. Polling INBUFFULL(CRPT_TDES_STS[9]) and OUTBUFEMPTY(CRPT_TDES_STS[16]). If INBUFFULL(CRPT_TDES_STS[9]) is 0, write 32 bits input data to CRPT_TDES_DATIN. If OUTBUFEMPTY(CRPT_TDES_STS[16]) is 0, read 32 bits data from CRPT_TDES_DATOUT.
8. Repeat step 7 until 64 bits data (8 bytes) are written to and read from TDES engine.
9. Write 1 to DMALAST(CRPT_TDES_CTL[5]).
10. Repeat steps 7 to step 9 until all data processed.

6.9.4.4 SHA (Secure Hash Algorithm)

The Secure Hash Algorithm is a family of cryptographic hash functions published by the National Institute of Standards and Technology (NIST) as a U.S. Federal Information Processing Standard (FIPS).

	Output Size (Bits)	Internal State Size (Bits)	Block Size (Bits)	Rounds
SHA-160	160	160	512	80
SHA-224	224	256	512	64
SHA-256	256	256	512	64

Table 6.9-1 Comparison of SHA Functions

User can refer to the following steps to understand how to program the SHA related registers.

SHA DMA mode programming flow:

1. Write 1 to SHAIEN(CRPT_INTEN[24]) to enable SHA interrupt.
2. Configure SHA control register CRPT_SHA_CTL for SHA engine input/output data swap, DMA mode, and SHA operation mode.
3. Program DMA source address to register CRPT_SHA_SADDR.
4. Program DMA byte count to register CRPT_SHA_DMACNT.
5. Write input data to DMA source address with selected DMA byte count.
6. Write 1 to START(CRPT_SHA_CTL[0]) to start SHA encryption.
7. Waits for the SHA interrupt flag SHAIF(CRPT_INTSTS[24]) be set.
8. Read output digest (SHA160: CRPT_SHA_DGST0 ~ CRPT_SHA_DGST4, SHA224: CRPT_SHA_DGST0 ~ CRPT_SHA_DGST6, SHA256: CRPT_SHA_DGST0 ~ CRPT_SHA_DGST7).

SHA Non-DMA mode programming flow:

1. Configure SHA control register CRPT_SHA_CTL for SHA engine input/output data swap and SHA operation mode.
2. If it's the last input word, set DMALAST(CRPT_SHA_CTL[5]).
3. Write 1 to START(CRPT_SHA_CTL[0]) to start SHA encryption.
4. Waits for the SHA data input request DATINREQ(CRPT_SHA_STS[16]) be set.
5. Write one word of input data to CRPT_SHA_DATIN.
6. Repeat step 2 to 5 until all inut words are written into SHA engine.
7. Waits for the BUSY (CRPT_SHA_STS[0]) be cleared.
8. Read output digest (SHA160: CRPT_SHA_DGST0 ~ CRPT_SHA_DGST4, SHA224: CRPT_SHA_DGST0 ~ CRPT_SHA_DGST6, SHA256: CRPT_SHA_DGST0 ~ CRPT_SHA_DGST7).

6.9.5 Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
CRYPTO Base Address:				
CRYP_BA = 0x5008_0000				
CRPT_INTEN	CRYP_BA+0x000	R/W	Crypto Interrupt Enable Control Register	0x0000_0000
CRPT_INTSTS	CRYP_BA+0x004	R/W	Crypto Interrupt Flag	0x0000_0000
CRPT_PRNG_CTL	CRYP_BA+0x008	R/W	PRNG Control Register	0x0000_0000
CRPT_PRNG_SEED	CRYP_BA+0x00C	W	Seed for PRNG	Undefined
CRPT_PRNG_KEY0	CRYP_BA+0x010	R	PRNG Generated Key0	Undefined
CRPT_PRNG_KEY1	CRYP_BA+0x014	R	PRNG Generated Key1	Undefined
CRPT_PRNG_KEY2	CRYP_BA+0x018	R	PRNG Generated Key2	Undefined
CRPT_PRNG_KEY3	CRYP_BA+0x01C	R	PRNG Generated Key3	Undefined
CRPT_PRNG_KEY4	CRYP_BA+0x020	R	PRNG Generated Key4	Undefined
CRPT_PRNG_KEY5	CRYP_BA+0x024	R	PRNG Generated Key5	Undefined
CRPT_PRNG_KEY6	CRYP_BA+0x028	R	PRNG Generated Key6	Undefined
CRPT_PRNG_KEY7	CRYP_BA+0x02C	R	PRNG Generated Key7	Undefined
CRPT_AES_FDBCK0	CRYP_BA+0x050	R	AES Engine Output Feedback Data after Cryptographic Operation	0x0000_0000
CRPT_AES_FDBCK1	CRYP_BA+0x054	R	AES Engine Output Feedback Data after Cryptographic Operation	0x0000_0000
CRPT_AES_FDBCK2	CRYP_BA+0x058	R	AES Engine Output Feedback Data after Cryptographic Operation	0x0000_0000
CRPT_AES_FDBCK3	CRYP_BA+0x05C	R	AES Engine Output Feedback Data after Cryptographic Operation	0x0000_0000
CRPT_TDES_FDBCKH	CRYP_BA+0x060	R	TDES/DES Engine Output Feedback High Word Data after Cryptographic Operation	0x0000_0000
CRPT_TDES_FDBCKL	CRYP_BA+0x064	R	TDES/DES Engine Output Feedback Low Word Data after Cryptographic Operation	0x0000_0000
CRPT_AES_CTL	CRYP_BA+0x100	R/W	AES Control Register	0x0000_0000
CRPT_AES_STS	CRYP_BA+0x104	R	AES Engine Flag	0x0001_0100
CRPT_AES_DATIN	CRYP_BA+0x108	R/W	AES Engine Data Input Port Register	0x0000_0000
CRPT_AES_DATOUT	CRYP_BA+0x10C	R	AES Engine Data Output Port Register	0x0000_0000
CRPT_AES0_KEY0	CRYP_BA+0x110	R/W	AES Key Word 0 Register for Channel 0	0x0000_0000
CRPT_AES0_KEY1	CRYP_BA+0x114	R/W	AES Key Word 1 Register for Channel 0	0x0000_0000
CRPT_AES0_KEY2	CRYP_BA+0x118	R/W	AES Key Word 2 Register for Channel 0	0x0000_0000
CRPT_AES0_KEY3	CRYP_BA+0x11C	R/W	AES Key Word 3 Register for Channel 0	0x0000_0000

CRPT_AES0_KEY4	CRYP_BA+0x120	R/W	AES Key Word 4 Register for Channel 0	0x0000_0000
CRPT_AES0_KEY5	CRYP_BA+0x124	R/W	AES Key Word 5 Register for Channel 0	0x0000_0000
CRPT_AES0_KEY6	CRYP_BA+0x128	R/W	AES Key Word 6 Register for Channel 0	0x0000_0000
CRPT_AES0_KEY7	CRYP_BA+0x12C	R/W	AES Key Word 7 Register for Channel 0	0x0000_0000
CRPT_AES0_IV0	CRYP_BA+0x130	R/W	AES Initial Vector Word 0 Register for Channel 0	0x0000_0000
CRPT_AES0_IV1	CRYP_BA+0x134	R/W	AES Initial Vector Word 1 Register for Channel 0	0x0000_0000
CRPT_AES0_IV2	CRYP_BA+0x138	R/W	AES Initial Vector Word 2 Register for Channel 0	0x0000_0000
CRPT_AES0_IV3	CRYP_BA+0x13C	R/W	AES Initial Vector Word 3 Register for Channel 0	0x0000_0000
CRPT_AES0_SADDR	CRYP_BA+0x140	R/W	AES DMA Source Address Register for Channel 0	0x0000_0000
CRPT_AES0_DADDR	CRYP_BA+0x144	R/W	AES DMA Destination Address Register for Channel 0	0x0000_0000
CRPT_AES0_CNT	CRYP_BA+0x148	R/W	AES Byte Count Register for Channel 0	0x0000_0000
CRPT_AES1_KEY0	CRYP_BA+0x14C	R/W	AES Key Word 0 Register for Channel 1	0x0000_0000
CRPT_AES1_KEY1	CRYP_BA+0x150	R/W	AES Key Word 1 Register for Channel 1	0x0000_0000
CRPT_AES1_KEY2	CRYP_BA+0x154	R/W	AES Key Word 2 Register for Channel 1	0x0000_0000
CRPT_AES1_KEY3	CRYP_BA+0x158	R/W	AES Key Word 3 Register for Channel 1	0x0000_0000
CRPT_AES1_KEY4	CRYP_BA+0x15C	R/W	AES Key Word 4 Register for Channel 1	0x0000_0000
CRPT_AES1_KEY5	CRYP_BA+0x160	R/W	AES Key Word 5 Register for Channel 1	0x0000_0000
CRPT_AES1_KEY6	CRYP_BA+0x164	R/W	AES Key Word 6 Register for Channel 1	0x0000_0000
CRPT_AES1_KEY7	CRYP_BA+0x168	R/W	AES Key Word 7 Register for Channel 1	0x0000_0000
CRPT_AES1_IV0	CRYP_BA+0x16C	R/W	AES Initial Vector Word 0 Register for Channel 1	0x0000_0000
CRPT_AES1_IV1	CRYP_BA+0x170	R/W	AES Initial Vector Word 1 Register for Channel 1	0x0000_0000
CRPT_AES1_IV2	CRYP_BA+0x174	R/W	AES Initial Vector Word 2 Register for Channel 1	0x0000_0000
CRPT_AES1_IV3	CRYP_BA+0x178	R/W	AES Initial Vector Word 3 Register for Channel 1	0x0000_0000
CRPT_AES1_SADDR	CRYP_BA+0x17C	R/W	AES DMA Source Address Register for Channel 1	0x0000_0000
CRPT_AES1_DADDR	CRYP_BA+0x180	R/W	AES DMA Destination Address Register for Channel 1	0x0000_0000
CRPT_AES1_CNT	CRYP_BA+0x184	R/W	AES Byte Count Register for Channel 1	0x0000_0000
CRPT_AES2_KEY0	CRYP_BA+0x188	R/W	AES Key Word 0 Register for Channel 2	0x0000_0000
CRPT_AES2_KEY1	CRYP_BA+0x18C	R/W	AES Key Word 1 Register for Channel 2	0x0000_0000
CRPT_AES2_KEY2	CRYP_BA+0x190	R/W	AES Key Word 2 Register for Channel 2	0x0000_0000
CRPT_AES2_KEY3	CRYP_BA+0x194	R/W	AES Key Word 3 Register for Channel 2	0x0000_0000
CRPT_AES2_KEY4	CRYP_BA+0x198	R/W	AES Key Word 4 Register for Channel 2	0x0000_0000
CRPT_AES2_KEY5	CRYP_BA+0x19C	R/W	AES Key Word 5 Register for Channel 2	0x0000_0000
CRPT_AES2_KEY6	CRYP_BA+0x1A0	R/W	AES Key Word 6 Register for Channel 2	0x0000_0000
CRPT_AES2_KEY7	CRYP_BA+0x1A4	R/W	AES Key Word 7 Register for Channel 2	0x0000_0000

CRPT_AES2_IV0	CRYP_BA+0x1A8	R/W	AES Initial Vector Word 0 Register for Channel 2	0x0000_0000
CRPT_AES2_IV1	CRYP_BA+0x1AC	R/W	AES Initial Vector Word 1 Register for Channel 2	0x0000_0000
CRPT_AES2_IV2	CRYP_BA+0x1B0	R/W	AES Initial Vector Word 2 Register for Channel 2	0x0000_0000
CRPT_AES2_IV3	CRYP_BA+0x1B4	R/W	AES Initial Vector Word 3 Register for Channel 2	0x0000_0000
CRPT_AES2_SADDR	CRYP_BA+0x1B8	R/W	AES DMA Source Address Register for Channel 2	0x0000_0000
CRPT_AES2_DADDR	CRYP_BA+0x1BC	R/W	AES DMA Destination Address Register for Channel 2	0x0000_0000
CRPT_AES2_CNT	CRYP_BA+0x1C0	R/W	AES Byte Count Register for Channel 2	0x0000_0000
CRPT_AES3_KEY0	CRYP_BA+0x1C4	R/W	AES Key Word 0 Register for Channel 3	0x0000_0000
CRPT_AES3_KEY1	CRYP_BA+0x1C8	R/W	AES Key Word 1 Register for Channel 3	0x0000_0000
CRPT_AES3_KEY2	CRYP_BA+0x1CC	R/W	AES Key Word 2 Register for Channel 3	0x0000_0000
CRPT_AES3_KEY3	CRYP_BA+0x1D0	R/W	AES Key Word 3 Register for Channel 3	0x0000_0000
CRPT_AES3_KEY4	CRYP_BA+0x1D4	R/W	AES Key Word 4 Register for Channel 3	0x0000_0000
CRPT_AES3_KEY5	CRYP_BA+0x1D8	R/W	AES Key Word 5 Register for Channel 3	0x0000_0000
CRPT_AES3_KEY6	CRYP_BA+0x1DC	R/W	AES Key Word 6 Register for Channel 3	0x0000_0000
CRPT_AES3_KEY7	CRYP_BA+0x1E0	R/W	AES Key Word 7 Register for Channel 3	0x0000_0000
CRPT_AES3_IV0	CRYP_BA+0x1E4	R/W	AES Initial Vector Word 0 Register for Channel 3	0x0000_0000
CRPT_AES3_IV1	CRYP_BA+0x1E8	R/W	AES Initial Vector Word 1 Register for Channel 3	0x0000_0000
CRPT_AES3_IV2	CRYP_BA+0x1EC	R/W	AES Initial Vector Word 2 Register for Channel 3	0x0000_0000
CRPT_AES3_IV3	CRYP_BA+0x1F0	R/W	AES Initial Vector Word 3 Register for Channel 3	0x0000_0000
CRPT_AES3_SADDR	CRYP_BA+0x1F4	R/W	AES DMA Source Address Register for Channel 3	0x0000_0000
CRPT_AES3_DADDR	CRYP_BA+0x1F8	R/W	AES DMA Destination Address Register for Channel 3	0x0000_0000
CRPT_AES3_CNT	CRYP_BA+0x1FC	R/W	AES Byte Count Register for Channel 3	0x0000_0000
CRPT_TDES_CTL	CRYP_BA+0x200	R/W	TDES/DES Control Register	0x0000_0000
CRPT_TDES_STS	CRYP_BA+0x204	R	TDES/DES Engine Flag	0x0001_0100
CRPT_TDES0_KEY1H	CRYP_BA+0x208	R/W	TDES/DES Key 1 High Word Register for Channel 0	0x0000_0000
CRPT_TDES0_KEY1L	CRYP_BA+0x20C	R/W	TDES/DES Key 1 Low Word Register for Channel 0	0x0000_0000
CRPT_TDES0_KEY2H	CRYP_BA+0x210	R/W	TDES Key 2 High Word Register for Channel 0	0x0000_0000
CRPT_TDES0_KEY2L	CRYP_BA+0x214	R/W	TDES Key 2 Low Word Register for Channel 0	0x0000_0000
CRPT_TDES0_KEY3H	CRYP_BA+0x218	R/W	TDES Key 3 High Word Register for Channel 0	0x0000_0000
CRPT_TDES0_KEY3L	CRYP_BA+0x21C	R/W	TDES Key 3 Low Word Register for Channel 0	0x0000_0000
CRPT_TDES0_IVH	CRYP_BA+0x220	R/W	TDES/DES Initial Vector High Word Register for Channel 0	0x0000_0000
CRPT_TDES0_IVL	CRYP_BA+0x224	R/W	TDES/DES Initial Vector Low Word Register for Channel 0	0x0000_0000

CRPT_TDES0_SADDR	CRYP_BA+0x228	R/W	TDES/DES DMA Source Address Register for Channel 0	0x0000_0000
CRPT_TDES0_DADDR	CRYP_BA+0x22C	R/W	TDES/DES DMA Destination Address Register for Channel 0	0x0000_0000
CRPT_TDES0_CNT	CRYP_BA+0x230	R/W	TDES/DES Byte Count Register for Channel 0	0x0000_0000
CRPT_TDES_DATIN	CRYP_BA+0x234	R/W	TDES/DES Engine Input data Word Register	0x0000_0000
CRPT_TDES_DATOUT	CRYP_BA+0x238	R	TDES/DES Engine Output data Word Register	0x0000_0000
CRPT_TDES1_KEY1H	CRYP_BA+0x248	R/W	TDES/DES Key 1 High Word Register for Channel 1	0x0000_0000
CRPT_TDES1_KEY1L	CRYP_BA+0x24C	R/W	TDES/DES Key 1 Low Word Register for Channel 1	0x0000_0000
CRPT_TDES1_KEY2H	CRYP_BA+0x250	R/W	TDES Key 2 High Word Register for Channel 1	0x0000_0000
CRPT_TDES1_KEY2L	CRYP_BA+0x254	R/W	TDES Key 2 Low Word Register for Channel 1	0x0000_0000
CRPT_TDES1_KEY3H	CRYP_BA+0x258	R/W	TDES Key 3 High Word Register for Channel 1	0x0000_0000
CRPT_TDES1_KEY3L	CRYP_BA+0x25C	R/W	TDES Key 3 Low Word Register for Channel 1	0x0000_0000
CRPT_TDES1_IVH	CRYP_BA+0x260	R/W	TDES/DES Initial Vector High Word Register for Channel 1	0x0000_0000
CRPT_TDES1_IVL	CRYP_BA+0x264	R/W	TDES/DES Initial Vector Low Word Register for Channel 1	0x0000_0000
CRPT_TDES1_SADDR	CRYP_BA+0x268	R/W	TDES/DES DMA Source Address Register for Channel 1	0x0000_0000
CRPT_TDES1_DADDR	CRYP_BA+0x26C	R/W	TDES/DES DMA Destination Address Register for Channel 1	0x0000_0000
CRPT_TDES1_CNT	CRYP_BA+0x270	R/W	TDES/DES Byte Count Register for Channel 1	0x0000_0000
CRPT_TDES2_KEY1H	CRYP_BA+0x288	R/W	TDES/DES Key 1 High Word Register for Channel 2	0x0000_0000
CRPT_TDES2_KEY1L	CRYP_BA+0x28C	R/W	TDES/DES Key 1 Low Word Register for Channel 2	0x0000_0000
CRPT_TDES2_KEY2H	CRYP_BA+0x290	R/W	TDES Key 2 High Word Register for Channel 2	0x0000_0000
CRPT_TDES2_KEY2L	CRYP_BA+0x294	R/W	TDES Key 2 Low Word Register for Channel 2	0x0000_0000
CRPT_TDES2_KEY3H	CRYP_BA+0x298	R/W	TDES Key 3 High Word Register for Channel 2	0x0000_0000
CRPT_TDES2_KEY3L	CRYP_BA+0x29C	R/W	TDES Key 3 Low Word Register for Channel 2	0x0000_0000
CRPT_TDES2_IVH	CRYP_BA+0x2A0	R/W	TDES/DES Initial Vector High Word Register for Channel 2	0x0000_0000
CRPT_TDES2_IVL	CRYP_BA+0x2A4	R/W	TDES/DES Initial Vector Low Word Register for Channel 2	0x0000_0000
CRPT_TDES2_SADDR	CRYP_BA+0x2A8	R/W	TDES/DES DMA Source Address Register for Channel 2	0x0000_0000
CRPT_TDES2_DADDR	CRYP_BA+0x2AC	R/W	TDES/DES DMA Destination Address Register for Channel 2	0x0000_0000
CRPT_TDES2_CNT	CRYP_BA+0x2B0	R/W	TDES/DES Byte Count Register for Channel 2	0x0000_0000
CRPT_TDES3_KEY1H	CRYP_BA+0x2C8	R/W	TDES/DES Key 1 High Word Register for Channel 3	0x0000_0000
CRPT_TDES3_KEY1L	CRYP_BA+0x2CC	R/W	TDES/DES Key 1 Low Word Register for Channel 3	0x0000_0000
CRPT_TDES3_KEY2H	CRYP_BA+0x2D0	R/W	TDES Key 2 High Word Register for Channel 3	0x0000_0000

CRPT_TDES3_KEY2L	CRYP_BA+0x2D4	R/W	TDES Key 2 Low Word Register for Channel 3	0x0000_0000
CRPT_TDES3_KEY3H	CRYP_BA+0x2D8	R/W	TDES Key 3 High Word Register for Channel 3	0x0000_0000
CRPT_TDES3_KEY3L	CRYP_BA+0x2DC	R/W	TDES Key 3 Low Word Register for Channel 3	0x0000_0000
CRPT_TDES3_IVH	CRYP_BA+0x2E0	R/W	TDES/DES Initial Vector High Word Register for Channel 3	0x0000_0000
CRPT_TDES3_IVL	CRYP_BA+0x2E4	R/W	TDES/DES Initial Vector Low Word Register for Channel 3	0x0000_0000
CRPT_TDES3_SADDR	CRYP_BA+0x2E8	R/W	TDES/DES DMA Source Address Register for Channel 3	0x0000_0000
CRPT_TDES3_DADDR	CRYP_BA+0x2EC	R/W	TDES/DES DMA Destination Address Register for Channel 3	0x0000_0000
CRPT_TDES3_CNT	CRYP_BA+0x2F0	R/W	TDES/DES Byte Count Register for Channel 3	0x0000_0000
CRPT_SHA_CTL	CRYP_BA+0x300	R/W	SHA Control Register	0x0000_0000
CRPT_SHA_STS	CRYP_BA+0x304	R	SHA Status Flag	0x0000_0000
CRPT_SHA_DGST0	CRYP_BA+0x308	R	SHA Digest Message 0	0x0000_0000
CRPT_SHA_DGST1	CRYP_BA+0x30C	R	SHA Digest Message 1	0x0000_0000
CRPT_SHA_DGST2	CRYP_BA+0x310	R	SHA Digest Message 2	0x0000_0000
CRPT_SHA_DGST3	CRYP_BA+0x314	R	SHA Digest Message 3	0x0000_0000
CRPT_SHA_DGST4	CRYP_BA+0x318	R	SHA Digest Message 4	0x0000_0000
CRPT_SHA_DGST5	CRYP_BA+0x31C	R	SHA Digest Message 5	0x0000_0000
CRPT_SHA_DGST6	CRYP_BA+0x320	R	SHA Digest Message 6	0x0000_0000
CRPT_SHA_DGST7	CRYP_BA+0x324	R	SHA Digest Message 7	0x0000_0000
CRPT_SHA_KEYCNT	CRYP_BA+0x348	R/W	SHA Key Byte Count Register	0x0000_0000
CRPT_SHA_SADDR	CRYP_BA+0x34C	R/W	SHA DMA Source Address Register	0x0000_0000
CRPT_SHA_DMACNT	CRYP_BA+0x350	R/W	SHA Byte Count Register	0x0000_0000
CRPT_SHA_DATIN	CRYP_BA+0x354	R/W	SHA Engine Non-DMA Mode Data Input Port Register	0x0000_0000

6.9.6 Register Description

6.9.6.1 Crypto Register

CRYPTO Interrupt Enable Control Register (CRPT_INTEN)

Register	Offset	R/W	Description	Reset Value
CRPT_INTEN	CRYP_BA+0x000	R/W	Crypto Interrupt Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						**SHAERRIEN**	**SHAIEN**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**PRNGIEN**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**TDESERRIEN**	**TDESIEN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**AESERRIEN**	**AESIEN**

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25]	**SHAERRIEN**	**SHA Error Interrupt Enable Bit** 0 = SHA error interrupt flag Disabled. 1 = SHA error interrupt flag Enabled.
[24]	**SHAIEN**	**SHA Interrupt Enable Bit** 0 = SHA interrupt Disabled. 1 = SHA interrupt Enabled. In DMA mode, an interrupt will be triggered when amount of data set in SHA_DMA_CNT is fed into the SHA engine. In Non-DMA mode, an interrupt will be triggered when the SHA engine finishes the operation.
[23:17]	**Reserved**	Reserved.
[16]	**PRNGIEN**	**PRNG Interrupt Enable Bit** 0 = PRNG interrupt Disabled. 1 = PRNG interrupt Enabled.
[15:10]	**Reserved**	Reserved.
[9]	**TDESERRIEN**	**TDES/DES Error Flag Enable Bit** 0 = TDES/DES error interrupt flag Disabled. 1 = TDES/DES error interrupt flag Enabled.

[8]	**TDESIEN**	**TDES/DES Interrupt Enable Bit** 0 = TDES/DES interrupt Disabled. 1 = TDES/DES interrupt Enabled. In DMA mode, an interrupt will be triggered when amount of data set in TDES_DMA_CNT is fed into the TDES engine. In Non-DMA mode, an interrupt will be triggered when the TDES engine finishes the operation.
[7:2]	**Reserved**	Reserved.
[1]	**AESERRIEN**	**AES Error Flag Enable Bit** 0 = AES error interrupt flag Disabled. 1 = AES error interrupt flag Enabled.
[0]	**AESIEN**	**AES Interrupt Enable Bit** 0 = AES interrupt Disabled. 1 = AES interrupt Enabled. In DMA mode, an interrupt will be triggered when amount of data set in AES_DMA_CNT is fed into the AES engine. In Non-DMA mode, an interrupt will be triggered when the AES engine finishes the operation.

CRYPTO Interrupt Flag Register (CRPT_INTSTS)

Register	Offset	R/W	Description	Reset Value
CRPT_INTSTS	CRYP_BA+0x004	R/W	Crypto Interrupt Flag	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						SHAERRIF	SHAIF
23	22	21	20	19	18	17	16
Reserved							PRNGIF
15	14	13	12	11	10	9	8
Reserved						TDESERRIF	TDESIF
7	6	5	4	3	2	1	0
Reserved						AESERRIF	AESIF

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25]	**SHAERRIF**	**SHA Error Flag** This register includes operating and setting error. The detail flag is shown in SHA _FLAG register. This bit is cleared by writing 1, and it has no effect by writing 0. 0 = No SHA error. 1 = SHA error interrupt.
[24]	**SHAIF**	**SHA Finish Interrupt Flag** This bit is cleared by writing 1, and it has no effect by writing 0. 0 = No SHA interrupt. 1 = SHA operation done interrupt.
[23:17]	**Reserved**	Reserved.
[16]	**PRNGIF**	**PRNG Finish Interrupt Flag** This bit is cleared by writing 1, and it has no effect by writing 0. 0 = No PRNG interrupt. 1 = PRNG key generation done interrupt.
[15:10]	**Reserved**	Reserved.
[9]	**TDESERRIF**	**TDES/DES Error Flag** This bit includes the operating and setting error. The detailed flag is shown in the TDES _FLAG register. This includes operating and setting error. This bit is cleared by writing 1, and it has no effect by writing 0. 0 = No TDES/DES error. 1 = TDES/DES encryption/decryption error interrupt.

[8]	**TDESIF**	**TDES/DES Finish Interrupt Flag** This bit is cleared by writing 1, and it has no effect by writing 0. 0 = No TDES/DES interrupt. 1 = TDES/DES encryption/decryption done interrupt.
[7:2]	**Reserved**	Reserved.
[1]	**AESERRIF**	**AES Error Flag** This bit is cleared by writing 1, and it has no effect by writing 0. 0 = No AES error. 1 = AES encryption/decryption done interrupt.
[0]	**AESIF**	**AES Finish Interrupt Flag** This bit is cleared by writing 1, and it has no effect by writing 0. 0 = No AES interrupt. 1 = AES encryption/decryption done interrupt.

6.9.6.2 PRNG Register

PRNG Control Register (CRPT_PRNG_CTL)

Register	Offset	R/W	Description	Reset Value
CRPT_PRNG_CTL	CRYP_BA+0x008	R/W	PRNG Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**BUSY**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**KEYSZ**		**SEEDRLD**	**START**

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**BUSY**	**PRNG Busy (Read Only)** 0 = PRNG engine is idle. 1 = Indicate that the PRNG engine is generating CRPT_PRNG_KEYx.
[7:4]	**Reserved**	Reserved.
[3:2]	**KEYSZ**	**PRNG Generate Key Size** 00 = 64 bits. 01 = 128 bits. 10 = 192 bits. 11 = 256 bits.
[1]	**SEEDRLD**	**Reload New Seed For PRNG Engine** 0 = Generating key based on the current seed. 1 = Reload new seed.
[0]	**START**	**Start PRNG Engine** 0 = Stop PRNG engine. 1 = Generate new key and store the new key to register CRPT_PRNG_KEYx , which will be cleared when the new key is generated.

PRNG Seed Register (CRPT_PRNG_SEED)

Register	Offset	R/W	Description	Reset Value
CRPT_PRNG_SEED	CRYP_BA+0x00C	W	Seed for PRNG	Undefined

31	30	29	28	27	26	25	24
SEED							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SEED							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SEED							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SEED							

Bits	Description	
[31:0]	**SEED**	**Seed For PRNG (Write Only)** The bits store the seed for PRNG engine.

PRNG Key x Register (CRPT_PRNG_KEYx)

Register	Offset	R/W	Description	Reset Value
CRPT_PRNG_KEY0	CRYP_BA+0x010	R	PRNG Generated Key0	Undefined
CRPT_PRNG_KEY1	CRYP_BA+0x014	R	PRNG Generated Key1	Undefined
CRPT_PRNG_KEY2	CRYP_BA+0x018	R	PRNG Generated Key2	Undefined
CRPT_PRNG_KEY3	CRYP_BA+0x01C	R	PRNG Generated Key3	Undefined
CRPT_PRNG_KEY4	CRYP_BA+0x020	R	PRNG Generated Key4	Undefined
CRPT_PRNG_KEY5	CRYP_BA+0x024	R	PRNG Generated Key5	Undefined
CRPT_PRNG_KEY6	CRYP_BA+0x028	R	PRNG Generated Key6	Undefined
CRPT_PRNG_KEY7	CRYP_BA+0x02C	R	PRNG Generated Key7	Undefined

31	30	29	28	27	26	25	24
KEYx							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
KEYx							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
KEYx							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
KEYx							

Bits	Description	
[31:0]	**KEYx**	**Store PRNG Generated Key (Read Only)** The bits store the key that is generated by PRNG.

6.9.6.3 AES Register

AES Control Register (CRPT_AES_CTL)

Register	Offset	R/W	Description	Reset Value
CRPT_AES_CTL	CRYP_BA+0x100	R/W	AES Control Register	0x0000_0000

31	30	29	28	27	26	25	24
KEYPRT	KEYUNPRT					CHANNEL	
23	22	21	20	19	18	17	16
INSWAP	OUTSWAP	Reserved					ENCRPT
15	14	13	12	11	10	9	8
OPMODE							
7	6	5	4	3	2	1	0
DMAEN	DMACSCAD	DMALAST	Reserved	KEYSZ		STOP	START

Bits	Description	
[31]	**KEYPRT**	**Protect Key** Read as a flag to reflect KEYPRT. 0 = No effect. 1 = Protect the content of the AES key from reading. The return value for reading CRPT_AESn_KEYx is not the content of the registers CRPT_AESn_KEYx. Once it is set, it can be cleared by asserting KEYUNPRT. And the key content would be cleared as well.
[30:26]	**KEYUNPRT**	**Unprotect Key** Writing 0 to CRPT_AES_CTL [31] and “10110” to CRPT_AES_CTL [30:26] is to unprotect the AES key. The KEYUNPRT can be read and written. When it is written as the AES engine is operating, BUSY flag is 1, there would be no effect on KEYUNPRT.
[25:24]	**CHANNEL**	**AES Engine Working Channel** 00 = Current control register setting is for channel 0. 01 = Current control register setting is for channel 1. 10 = Current control register setting is for channel 2. 11 = Current control register setting is for channel 3.
[23]	**INSWAP**	**AES Engine Input Data Swap** 0 = Keep the original order. 1 = The order that CPU feeds data to the accelerator will be changed from {byte3, byte2, byte1, byte0} to {byte0, byte1, byte2, byte3}.
[22]	**OUTSWAP**	**AES Engine Output Data Swap** 0 = Keep the original order. 1 = The order that CPU outputs data from the accelerator will be changed from {byte3, byte2, byte1, byte0} to {byte0, byte1, byte2, byte3}.

[16]	**ENCRPT**	**AES Encryption/Decryption** 0 = AES engine executes decryption operation. 1 = AES engine executes encryption operation.
[15:8]	**OPMODE**	**AES Engine Operation Modes** 0x00 = ECB (Electronic Codebook Mode) 0x01 = CBC (Cipher Block Chaining Mode). 0x02 = CFB (Cipher Feedback Mode). 0x03 = OFB (Output Feedback Mode). 0x04 = CTR (Counter Mode). 0x10 = CBC-CS1 (CBC Ciphertext-Stealing 1 Mode). 0x11 = CBC-CS2 (CBC Ciphertext-Stealing 2 Mode). 0x12 = CBC-CS3 (CBC Ciphertext-Stealing 3 Mode).
[7]	**DMAEN**	**AES Engine DMA Enable Bit** 0 = AES DMA engine Disabled. The AES engine operates in Non-DMA mode, and gets data from the port CRPT_AES_DATIN. 1 = AES DMA engine Enabled. The AES engine operates in DMA mode, and data movement from/to the engine is done by DMA logic.
[6]	**DMACSCAD**	**AES Engine DMA With Cascade Mode** 0 = DMA cascade function Disabled. 1 = In DMA cascade mode, software can update DMA source address register, destination address register, and byte count register during a cascade operation, without finishing the accelerator operation.
[5]	**DMALAST**	**AES Last Block** In DMA mode, this bit must be set as beginning the last DMA cascade round. In Non-DMA mode, this bit must be set when feeding in the last block of data in ECB, CBC, CTR, OFB, and CFB mode, and feeding in the (last-1) block of data at CBC-CS1, CBC-CS2, and CBC-CS3 mode. This bit is always 0 when it's read back. Must be written again once START is triggered.
[4]	**Reserved**	Reserved.
[3:2]	**KEYSZ**	**AES Key Size** This bit defines three different key size for AES operation. 2'b00 = 128 bits key. 2'b01 = 192 bits key. 2'b10 = 256 bits key. 2'b11 = Reserved. If the AES accelerator is operating and the corresponding flag BUSY is 1, updating this register has no effect.
[1]	**STOP**	**AES Engine Stop** 0 = No effect. 1 = Stop AES engine. **Note:** This bit is always 0 when it's read back.

[0]	**START**	**AES Engine Start** 0 = No effect. 1 = Start AES engine. BUSY flag will be set. **Note:** This bit is always 0 when it's read back.

AES Status Flag Register (CRPT_AES_STS)

Register	Offset	R/W	Description	Reset Value
CRPT_AES_STS	CRYP_BA+0x104	R	AES Engine Flag	0x0001_0100

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved			BUSERR	Reserved	OUTBUFERR	OUTBUFFULL	OUTBUFEMPTY
15	14	13	12	11	10	9	8
Reserved			CNTERR	Reserved	INBUFERR	INBUFFULL	INBUFEMPTY
7	6	5	4	3	2	1	0
Reserved							BUSY

Bits	Description	
[31:21]	**Reserved**	Reserved.
[20]	**BUSERR**	**AES DMA Access Bus Error Flag** 0 = No error. 1 = Bus error will stop DMA operation and AES engine.
[19]	**Reserved**	Reserved.
[18]	**OUTBUFERR**	**AES Out Buffer Error Flag** 0 = No error. 1 = Error happens during getting the result from AES engine.
[17]	**OUTBUFFULL**	**AES Out Buffer Full Flag** 0 = AES output buffer is not full. 1 = AES output buffer is full, and software needs to get data from AES_DATA_OUT. Otherwise, the AES engine will be pending since the output buffer is full.
[16]	**OUTBUFEMPTY**	**AES Out Buffer Empty** 0 = AES output buffer is not empty. There are some valid data kept in output buffer. 1 = AES output buffer is empty. Software cannot get data from AES_DATA_OUT. Otherwise, the flag OUTBUFERR will be set to 1 since the output buffer is empty.
[15:13]	**Reserved**	Reserved.
[12]	**CNTERR**	**AES_CNT Setting Error** 0 = No error in AES_CNT setting. 1 = AES_CNT is not a multiply of 16 in ECB, CBC, CFB, OFB, and CTR mode.
[11]	**Reserved**	Reserved.

[10]	**INBUFERR**	**AES Input Buffer Error Flag** 0 = No error. 1 = Error happens during feeding data to the AES engine.
[9]	**INBUFFULL**	**AES Input Buffer Full Flag** 0 = AES input buffer is not full. Software can feed the data into the AES engine. 1 = AES input buffer is full. Software cannot feed data to the AES engine. Otherwise, the flag INBUFERR will be set to 1.
[8]	**INBUFEMPTY**	**AES Input Buffer Empty** 0 = There are some data in input buffer waiting for the AES engine to process. 1 = AES input buffer is empty. Software needs to feed data to the AES engine. Otherwise, the AES engine will be pending to wait for input data.
[7:1]	**Reserved**	Reserved.
[0]	**BUSY**	**AES Engine Busy** 0 = The AES engine is idle or finished. 1 = The AES engine is under processing.

AES Data Input Port Register (CRPT_AES_DATIN)

Register	Offset	R/W	Description	Reset Value
CRPT_AES_DATIN	CRYP_BA+0x108	R/W	AES Engine Data Input Port Register	0x0000_0000

31	30	29	28	27	26	25	24
DATIN							
23	22	21	20	19	18	17	16
DATIN							
15	14	13	12	11	10	9	8
DATIN							
7	6	5	4	3	2	1	0
DATIN							

Bits	Description	
[31:0]	DATIN	**AES Engine Input Port** CPU feeds data to AES engine through this port by checking CRPT_AES_STS. Feed data as INBUFFULL is 0.

AES Data Output Port Register (CRPT_AES_DATOUT)

Register	Offset	R/W	Description	Reset Value
CRPT_AES_DATOUT	CRYP_BA+0x10C	R	AES Engine Data Output Port Register	0x0000_0000

31	30	29	28	27	26	25	24
DATOUT							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DATOUT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DATOUT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATOUT							

Bits	Description	
[31:0]	**DATOUT**	**AES Engine Output Port** CPU gets results from the AES engine through this port by checking CRPT_AES_STS. Get data as OUTBUFEMPTY is 0.

AES Key Word x Register (CRPT_AES0_KEYx, CRPT_AES1_KEYx, CRPT_AES2_KEYx, CRPT_AES3_KEYx)

Register	Offset	R/W	Description	Reset Value
CRPT_AES0_KEY0	CRYP_BA+0x110	R/W	AES Key Word 0 Register for Channel 0	0x0000_0000
CRPT_AES0_KEY1	CRYP_BA+0x114	R/W	AES Key Word 1 Register for Channel 0	0x0000_0000
CRPT_AES0_KEY2	CRYP_BA+0x118	R/W	AES Key Word 2 Register for Channel 0	0x0000_0000
CRPT_AES0_KEY3	CRYP_BA+0x11C	R/W	AES Key Word 3 Register for Channel 0	0x0000_0000
CRPT_AES0_KEY4	CRYP_BA+0x120	R/W	AES Key Word 4 Register for Channel 0	0x0000_0000
CRPT_AES0_KEY5	CRYP_BA+0x124	R/W	AES Key Word 5 Register for Channel 0	0x0000_0000
CRPT_AES0_KEY6	CRYP_BA+0x128	R/W	AES Key Word 6 Register for Channel 0	0x0000_0000
CRPT_AES0_KEY7	CRYP_BA+0x12C	R/W	AES Key Word 7 Register for Channel 0	0x0000_0000
CRPT_AES1_KEY0	CRYP_BA+0x14C	R/W	AES Key Word 0 Register for Channel 1	0x0000_0000
CRPT_AES1_KEY1	CRYP_BA+0x150	R/W	AES Key Word 1 Register for Channel 1	0x0000_0000
CRPT_AES1_KEY2	CRYP_BA+0x154	R/W	AES Key Word 2 Register for Channel 1	0x0000_0000
CRPT_AES1_KEY3	CRYP_BA+0x158	R/W	AES Key Word 3 Register for Channel 1	0x0000_0000
CRPT_AES1_KEY4	CRYP_BA+0x15C	R/W	AES Key Word 4 Register for Channel 1	0x0000_0000
CRPT_AES1_KEY5	CRYP_BA+0x160	R/W	AES Key Word 5 Register for Channel 1	0x0000_0000
CRPT_AES1_KEY6	CRYP_BA+0x164	R/W	AES Key Word 6 Register for Channel 1	0x0000_0000
CRPT_AES1_KEY7	CRYP_BA+0x168	R/W	AES Key Word 7 Register for Channel 1	0x0000_0000
CRPT_AES2_KEY0	CRYP_BA+0x188	R/W	AES Key Word 0 Register for Channel 2	0x0000_0000
CRPT_AES2_KEY1	CRYP_BA+0x18C	R/W	AES Key Word 1 Register for Channel 2	0x0000_0000
CRPT_AES2_KEY2	CRYP_BA+0x190	R/W	AES Key Word 2 Register for Channel 2	0x0000_0000
CRPT_AES2_KEY3	CRYP_BA+0x194	R/W	AES Key Word 3 Register for Channel 2	0x0000_0000
CRPT_AES2_KEY4	CRYP_BA+0x198	R/W	AES Key Word 4 Register for Channel 2	0x0000_0000

CRPT_AES2_KEY5	CRYP_BA+0x19C	R/W	AES Key Word 5 Register for Channel 2	0x0000_0000
CRPT_AES2_KEY6	CRYP_BA+0x1A0	R/W	AES Key Word 6 Register for Channel 2	0x0000_0000
CRPT_AES2_KEY7	CRYP_BA+0x1A4	R/W	AES Key Word 7 Register for Channel 2	0x0000_0000
CRPT_AES3_KEY0	CRYP_BA+0x1C4	R/W	AES Key Word 0 Register for Channel 3	0x0000_0000
CRPT_AES3_KEY1	CRYP_BA+0x1C8	R/W	AES Key Word 1 Register for Channel 3	0x0000_0000
CRPT_AES3_KEY2	CRYP_BA+0x1CC	R/W	AES Key Word 2 Register for Channel 3	0x0000_0000
CRPT_AES3_KEY3	CRYP_BA+0x1D0	R/W	AES Key Word 3 Register for Channel 3	0x0000_0000
CRPT_AES3_KEY4	CRYP_BA+0x1D4	R/W	AES Key Word 4 Register for Channel 3	0x0000_0000
CRPT_AES3_KEY5	CRYP_BA+0x1D8	R/W	AES Key Word 5 Register for Channel 3	0x0000_0000
CRPT_AES3_KEY6	CRYP_BA+0x1DC	R/W	AES Key Word 6 Register for Channel 3	0x0000_0000
CRPT_AES3_KEY7	CRYP_BA+0x1E0	R/W	AES Key Word 7 Register for Channel 3	0x0000_0000

31	30	29	28	27	26	25	24
KEY							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
KEY							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
KEY							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
KEY							

Bits	Description

[31:0]	**KEY**	**CRPT_AESn_KEYx** The KEY keeps the security key for AES operation. n = 0, 1..3. x = 0, 1..7. The security key for AES accelerator can be 128, 192, or 256 bits and four, six, or eight 32-bit registers are to store each security key. {CRPT_AESn_KEY3, CRPT_AESn_KEY2, CRPT_AESn_KEY1, CRPT_AESn_KEY0} stores the 128-bit security key for AES operation. {CRPT_AESn_KEY5, CRPT_AESn_KEY4, CRPT_AESn_KEY3, CRPT_AESn_KEY2, CRPT_AESn_KEY1, CRPT_AESn_KEY0} stores the 192-bit security key for AES operation. {CRPT_AESn_KEY7, CRPT_AESn_KEY6, CRPT_AESn_KEY5, CRPT_AESn_KEY4, CRPT_AESn_KEY3, CRPT_AESn_KEY2, CRPT_AESn_KEY1, CRPT_AESn_KEY0} stores the 256-bit security key for AES operation.

AES Initial Vector Word x Register (CRPT_AES0_IVx, CRPT_AES1_IVx, CRPT_AES2_IVx, CRPT_AES3_IVx)

Register	Offset	R/W	Description	Reset Value
CRPT_AES0_IV0	CRYP_BA+0x130	R/W	AES Initial Vector Word 0 Register for Channel 0	0x0000_0000
CRPT_AES0_IV1	CRYP_BA+0x134	R/W	AES Initial Vector Word 1 Register for Channel 0	0x0000_0000
CRPT_AES0_IV2	CRYP_BA+0x138	R/W	AES Initial Vector Word 2 Register for Channel 0	0x0000_0000
CRPT_AES0_IV3	CRYP_BA+0x13C	R/W	AES Initial Vector Word 3 Register for Channel 0	0x0000_0000
CRPT_AES1_IV0	CRYP_BA+0x16C	R/W	AES Initial Vector Word 0 Register for Channel 1	0x0000_0000
CRPT_AES1_IV1	CRYP_BA+0x170	R/W	AES Initial Vector Word 1 Register for Channel 1	0x0000_0000
CRPT_AES1_IV2	CRYP_BA+0x174	R/W	AES Initial Vector Word 2 Register for Channel 1	0x0000_0000
CRPT_AES1_IV3	CRYP_BA+0x178	R/W	AES Initial Vector Word 3 Register for Channel 1	0x0000_0000
CRPT_AES2_IV0	CRYP_BA+0x1A8	R/W	AES Initial Vector Word 0 Register for Channel 2	0x0000_0000
CRPT_AES2_IV1	CRYP_BA+0x1AC	R/W	AES Initial Vector Word 1 Register for Channel 2	0x0000_0000
CRPT_AES2_IV2	CRYP_BA+0x1B0	R/W	AES Initial Vector Word 2 Register for Channel 2	0x0000_0000
CRPT_AES2_IV3	CRYP_BA+0x1B4	R/W	AES Initial Vector Word 3 Register for Channel 2	0x0000_0000
CRPT_AES3_IV0	CRYP_BA+0x1E4	R/W	AES Initial Vector Word 0 Register for Channel 3	0x0000_0000
CRPT_AES3_IV1	CRYP_BA+0x1E8	R/W	AES Initial Vector Word 1 Register for Channel 3	0x0000_0000
CRPT_AES3_IV2	CRYP_BA+0x1EC	R/W	AES Initial Vector Word 2 Register for Channel 3	0x0000_0000
CRPT_AES3_IV3	CRYP_BA+0x1F0	R/W	AES Initial Vector Word 3 Register for Channel 3	0x0000_0000

31	30	29	28	27	26	25	24
IV							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
IV							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
IV							

7	6	5	4	3	2	1	0
IV							

Bits	Description	
[31:0]	**IV**	**CRPT_AESn_IVx** n = 0, 1..3. x = 0, 1..3. Four initial vectors (CRPT_AESn_IV0, CRPT_AESn_IV1, CRPT_AESn_IV2, and CRPT_AESn_IV3) are for AES operating in CBC, CFB, and OFB mode. Four registers (CRPT_AESn_IV0, CRPT_AESn_IV1, CRPT_AESn_IV2, and CRPT_AESn_IV3) act as Nonce counter when the AES engine is operating in CTR mode.

AES DMA Source Address Register (CRPT_AES0_SADDR, CRPT_AES1_SADDR, CRPT_AES2_SADDR, CRPT_AES3_SADDR)

Register	Offset	R/W	Description	Reset Value
CRPT_AES0_SADDR	CRYP_BA+0x140	R/W	AES DMA Source Address Register for Channel 0	0x0000_0000
CRPT_AES1_SADDR	CRYP_BA+0x17C	R/W	AES DMA Source Address Register for Channel 1	0x0000_0000
CRPT_AES2_SADDR	CRYP_BA+0x1B8	R/W	AES DMA Source Address Register for Channel 2	0x0000_0000
CRPT_AES3_SADDR	CRYP_BA+0x1F4	R/W	AES DMA Source Address Register for Channel 3	0x0000_0000

31	30	29	28	27	26	25	24
SADDR[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SADDR[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SADDR[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SADDR[7:0]							

Bits	Description	
[31:0]	**SADDR**	**AES DMA Source Address** The AES accelerator supports DMA function to transfer the plain text between system memory and embedded FIFO. The SADDR keeps the source address of the data buffer where the source text is stored. Based on the source address, the AES accelerator can read the plain text from system memory and do AES operation. The start of source address should be located at word boundary. In other words, bit 1 and 0 of SADDR are ignored. SADDR can be read and written. Writing to SADDR while the AES accelerator is operating doesn't affect the current AES operation. But the value of SADDR will be updated later on. Consequently, software can prepare the DMA source address for the next AES operation. In DMA mode, software can update the next CRPT_AESn_SADDR before triggering START. The value of CRPT_AESn_SADDR and CRPT_AESn_DADDR can be the same.

AES DMA Destination Address Register (CRPT_AES0_DADDR, CRPT_AES1_DADDR, CRPT_AES2_DADDR, CRPT_AES3_DADDR)

Register	Offset	R/W	Description	Reset Value
CRPT_AES0_DADDR	CRYP_BA+0x144	R/W	AES DMA Destination Address Register for Channel 0	0x0000_0000
CRPT_AES1_DADDR	CRYP_BA+0x180	R/W	AES DMA Destination Address Register for Channel 1	0x0000_0000
CRPT_AES2_DADDR	CRYP_BA+0x1BC	R/W	AES DMA Destination Address Register for Channel 2	0x0000_0000
CRPT_AES3_DADDR	CRYP_BA+0x1F8	R/W	AES DMA Destination Address Register for Channel 3	0x0000_0000

31	30	29	28	27	26	25	24
DADDR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DADDR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DADDR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DADDR							

Bits	Description	
[31:0]	**DADDR**	**AES DMA Destination Address** The AES accelerator supports DMA function to transfer the cipher text between system memory and embedded FIFO. The DADDR keeps the destination address of the data buffer where the engine output's text will be stored. Based on the destination address, the AES accelerator can write the cipher text back to system memory after the AES operation is finished. The start of destination address should be located at word boundary. In other words, bit 1 and 0 of DADDR are ignored. DADDR can be read and written. Writing to DADDR while the AES accelerator is operating doesn't affect the current AES operation. But the value of DADDR will be updated later on. Consequently, software can prepare the destination address for the next AES operation. In DMA mode, software can update the next CRPT_AESn_DADDR before triggering START. The value of CRPT_AESn_SADDR and CRPT_AESn_DADDR can be the same.

AES Byte Count Register (CRPT_AES0_CNT, CRPT_AES1_CNT, CRPT_AES2_CNT, CRPT_AES3_CNT)

Register	Offset	R/W	Description	Reset Value
CRPT_AES0_CNT	CRYP_BA+0x148	R/W	AES Byte Count Register for Channel 0	0x0000_0000
CRPT_AES1_CNT	CRYP_BA+0x184	R/W	AES Byte Count Register for Channel 1	0x0000_0000
CRPT_AES2_CNT	CRYP_BA+0x1C0	R/W	AES Byte Count Register for Channel 2	0x0000_0000
CRPT_AES3_CNT	CRYP_BA+0x1FC	R/W	AES Byte Count Register for Channel 3	0x0000_0000

31	30	29	28	27	26	25	24
CNT							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CNT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNT							

Bits	Description	
[31:0]	**CNT**	**AES Byte Count** The CRPT_AESn_CNT keeps the byte count of source text that is for the AES engine operating in DMA mode. The CRPT_AESn_CNT is 32-bit and the maximum of byte count is 4G bytes. CRPT_AESn_CNT can be read and written. Writing to CRPT_AESn_CNT while the AES accelerator is operating doesn't affect the current AES operation. But the value of CRPT_AESn_CNT will be updated later on. Consequently, software can prepare the byte count of data for the next AES operation. According to CBC-CS1, CBC-CS2, and CBC-CS3 standard, the count of operation data must be at least one block. Operations that are less than one block will output unexpected result. In Non-DMA ECB, CBC, CFB, OFB, and CTR mode, CRPT_AESn_CNT must be set as byte count for the last block of data before feeding in the last block of data. In Non-DMA CBC-CS1, CBC-CS2, and CBC-CS3 mode, CRPT_AESn_CNT must be set as byte count for the last two blocks of data before feeding in the last two blocks of data.

AES Feedback x Register (CRPT_AES_FDBCKx)

Register	Offset	R/W	Description	Reset Value
CRPT_AES_FDBCK0	CRYP_BA+0x050	R	AES Engine Output Feedback Data after Cryptographic Operation	0x0000_0000
CRPT_AES_FDBCK1	CRYP_BA+0x054	R	AES Engine Output Feedback Data after Cryptographic Operation	0x0000_0000
CRPT_AES_FDBCK2	CRYP_BA+0x058	R	AES Engine Output Feedback Data after Cryptographic Operation	0x0000_0000
CRPT_AES_FDBCK3	CRYP_BA+0x05C	R	AES Engine Output Feedback Data after Cryptographic Operation	0x0000_0000

31	30	29	28	27	26	25	24
FDBCK							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
FDBCK							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FDBCK							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FDBCK							

Bits	Description	
[31:0]	**FDBCK**	**AES Feedback Information** The feedback value is 128 bits in size. The AES engine uses the data from CRPT_AES_FDBCKx as the data inputted to CRPT_AESn_IVx for the next block in DMA cascade mode. The AES engine outputs feedback information for IV in the next block's operation. Software can use this feedback information to implement more than four DMA channels. Software can store that feedback value temporarily. After switching back, fill the stored feedback value to this register in the same channel operation, and then continue the operation with the original setting.

6.9.6.4 TDES/DES Register

TDES/DES Control Register (CRPT_TDES_CTL)

Register	Offset	R/W	Description	Reset Value
CRPT_TDES_CTL	CRYP_BA+0x200	R/W	TDES/DES Control Register	0x0000_0000

31	30	29	28	27	26	25	24
KEYPRT	**KEYUNPRT**					**CHANNEL**	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
INSWAP	**OUTSWAP**	**BLKSWAP**	**Reserved**				**ENCRPT**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**OPMODE**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAEN	**DMACSCAD**	**DMALAST**	**Reserved**	**3KEYS**	**TMODE**	**STOP**	**START**

Bits	Description	
[31]	**KEYPRT**	**Protect Key** Read as a flag to reflect KEYPRT. 0 = No effect. 1 = This bit is to protect the content of TDES key from reading. The return value for reading CRPT_ TDESn_KEYxH/L is not the content in the registers CRPT_ TDESn_KEYxH/L. Once it is set, it can be cleared by asserting KEYUNPRT. The key content would be cleared as well.
[30:26]	**KEYUNPRT**	**Unprotect Key** Writing 0 to CRPT_TDES_CTL [31] and “10110” to CRPT_TDES_CTL [30:26] is to unprotect TDES key. The KEYUNPRT can be read and written. When it is written as the TDES engine is operating, BUSY flag is 1, there would be no effect on KEYUNPRT.
[25:24]	**CHANNEL**	**TDES/DES Engine Working Channel** 00 = Current control register setting is for channel 0. 01 = Current control register setting is for channel 1. 10 = Current control register setting is for channel 2. 11 = Current control register setting is for channel 3.
[23]	**INSWAP**	**TDES/DES Engine Input Data Swap** 0 = Keep the original order. 1 = The order that CPU feeds data to the accelerator will be changed from {byte3, byte2, byte1, byte0} to {byte0, byte1, byte2, byte3}.
[22]	**OUTSWAP**	**TDES/DES Engine Output Data Swap** 0 = Keep the original order. 1 = The order that CPU outputs data from the accelerator will be changed from {byte3, byte2, byte1, byte0} to {byte0, byte1, byte2, byte3}.

[21]	**BLKSWAP**	**TDES/DES Engine Block Double Word Endian Swap** 0 = Keep the original order, e.g. **{WORD_H, WORD_L}**. 1 = When this bit is set to 1, the TDES engine would exchange high and low word in the sequence **{WORD_L, WORD_H}.**
[20:17]	**Reserved**	Reserved.
[16]	**ENCRPT**	**TDES/DES Encryption/Decryption** 0 = TDES engine executes decryption operation. 1 = TDES engine executes encryption operation.
[15:11]	**Reserved**	Reserved.
[10:8]	**OPMODE**	**TDES/DES Engine Operation Mode** 0x00 = ECB (Electronic Codebook Mode). 0x01 = CBC (Cipher Block Chaining Mode). 0x02 = CFB (Cipher Feedback Mode). 0x03 = OFB (Output Feedback Mode). 0x04 = CTR (Counter Mode). Others = CTR (Counter Mode).
[7]	**DMAEN**	**TDES/DES Engine DMA Enable Bit** 0 = TDES_DMA engine Disabled. TDES engine operates in Non-DMA mode, and get data from the port CRPT_TDES_DATIN. 1 = TDES_DMA engine Enabled. TDES engine operates in DMA mode, and data movement from/to the engine is done by DMA logic.
[6]	**DMACSCAD**	**TDES/DES Engine DMA With Cascade Mode** 0 = DMA cascade function Disabled. 1 = In DMA Cascade mode, software can update DMA source address register, destination address register, and byte count register during a cascade operation, without finishing the accelerator operation.
[5]	**DMALAST**	**TDES/DES Engine Start For The Last Block** In DMA mode, this bit must be set as beginning the last DMA cascade round. In Non-DMA mode, this bit must be set as feeding in last block of data.
[4]	**Reserved**	Reserved.
[3]	**3KEYS**	**TDES/DES Key Number** 0 = Select KEY1 and KEY2 in TDES/DES engine. 1 = Triple keys in TDES/DES engine Enabled.
[2]	**TMODE**	**TDES/DES Engine Operating Mode** 0 = Set DES mode for TDES/DES engine. 1 = Set Triple DES mode for TDES/DES engine.
[1]	**STOP**	**TDES/DES Engine Stop** 0 = No effect. 1 = Stop TDES/DES engine. **Note:** The bit is always 0 when it's read back.

[0]	**START**	**TDES/DES Engine Start** 0 = No effect. 1 = Start TDES/DES engine. The flag BUSY would be set. **Note:** The bit is always 0 when it's read back.

TDES/DES Status Flag Register (CRPT_TDES_STS)

Register	Offset	R/W	Description	Reset Value
CRPT_TDES_STS	CRYP_BA+0x204	R	TDES/DES Engine Flag	0x0001_0100

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved			BUSERR	Reserved	OUTBUFERR	OUTBUFFULL	OUTBUFEMPTY
15	14	13	12	11	10	9	8
Reserved					INBUFERR	INBUFFULL	INBUFEMPTY
7	6	5	4	3	2	1	0
Reserved							BUSY

Bits	Description	
[31:21]	**Reserved**	Reserved.
[20]	**BUSERR**	**TDES/DES DMA Access Bus Error Flag** 0 = No error. 1 = Bus error will stop DMA operation and TDES/DES engine.
[19]	**Reserved**	Reserved.
[18]	**OUTBUFERR**	**TDES/DES Out Buffer Error Flag** 0 = No error. 1 = Error happens during getting test result from TDES/DES engine.
[17]	**OUTBUFFULL**	**TDES/DES Output Buffer Full Flag** 0 = TDES/DES output buffer is not full. 1 = TDES/DES output buffer is full, and software needs to get data from TDES_DATA_OUT. Otherwise, the TDES/DES engine will be pending since output buffer is full.
[16]	**OUTBUFEMPTY**	**TDES/DES Output Buffer Empty Flag** 0 = TDES/DES output buffer is not empty. There are some valid data kept in output buffer. 1 = TDES/DES output buffer is empty, Software cannot get data from TDES_DATA_OUT. Otherwise the flag OUTBUFERR will be set to 1, since output buffer is empty.
[15:11]	**Reserved**	Reserved.
[10]	**INBUFERR**	**TDES/DES In Buffer Error Flag** 0 = No error. 1 = Error happens during feeding data to the TDES/DES engine.

[9]	**INBUFFULL**	**TDES/DES In Buffer Full Flag** 0 = TDES/DES input buffer is not full. Software can feed the data into the TDES/DES engine. 1 = TDES input buffer is full. Software cannot feed data to the TDES/DES engine. Otherwise, the flag INBUFERR will be set to 1.
[8]	**INBUFEMPTY**	**TDES/DES In Buffer Empty** 0 = There are some data in input buffer waiting for the TDES/DES engine to process. 1 = TDES/DES input buffer is empty. Software needs to feed data to the TDES/DES engine. Otherwise, the TDES/DES engine will be pending to wait for input data.
[7:1]	**Reserved**	Reserved.
[0]	**BUSY**	**TDES/DES Engine Busy** 0 = TDES/DES engine is idle or finished. 1 = TDES/DES engine is under processing.

TDES/DES Key 1, 2, 3 High/Low Word Register (TDES_KEY1H/L, TDES_KEY2H/L, TDES_KEY3H/L)

Register	Offset	R/W	Description	Reset Value
CRPT_TDES0_KEY1H	CRYP_BA+0x208	R/W	TDES/DES Key 1 High Word Register for Channel 0	0x0000_0000
CRPT_TDES0_KEY1L	CRYP_BA+0x20C	R/W	TDES/DES Key 1 Low Word Register for Channel 0	0x0000_0000
CRPT_TDES0_KEY2H	CRYP_BA+0x210	R/W	TDES Key 2 High Word Register for Channel 0	0x0000_0000
CRPT_TDES0_KEY2L	CRYP_BA+0x214	R/W	TDES Key 2 Low Word Register for Channel 0	0x0000_0000
CRPT_TDES0_KEY3H	CRYP_BA+0x218	R/W	TDES Key 3 High Word Register for Channel 0	0x0000_0000
CRPT_TDES0_KEY3L	CRYP_BA+0x21C	R/W	TDES Key 3 Low Word Register for Channel 0	0x0000_0000
CRPT_TDES1_KEY1H	CRYP_BA+0x248	R/W	TDES/DES Key 1 High Word Register for Channel 1	0x0000_0000
CRPT_TDES1_KEY1L	CRYP_BA+0x24C	R/W	TDES/DES Key 1 Low Word Register for Channel 1	0x0000_0000
CRPT_TDES1_KEY2H	CRYP_BA+0x250	R/W	TDES Key 2 High Word Register for Channel 1	0x0000_0000
CRPT_TDES1_KEY2L	CRYP_BA+0x254	R/W	TDES Key 2 Low Word Register for Channel 1	0x0000_0000
CRPT_TDES1_KEY3H	CRYP_BA+0x258	R/W	TDES Key 3 High Word Register for Channel 1	0x0000_0000
CRPT_TDES1_KEY3L	CRYP_BA+0x25C	R/W	TDES Key 3 Low Word Register for Channel 1	0x0000_0000
CRPT_TDES2_KEY1H	CRYP_BA+0x288	R/W	TDES/DES Key 1 High Word Register for Channel 2	0x0000_0000
CRPT_TDES2_KEY1L	CRYP_BA+0x28C	R/W	TDES/DES Key 1 Low Word Register for Channel 2	0x0000_0000
CRPT_TDES2_KEY2H	CRYP_BA+0x290	R/W	TDES Key 2 High Word Register for Channel 2	0x0000_0000
CRPT_TDES2_KEY2L	CRYP_BA+0x294	R/W	TDES Key 2 Low Word Register for Channel 2	0x0000_0000
CRPT_TDES2_KEY3H	CRYP_BA+0x298	R/W	TDES Key 3 High Word Register for Channel 2	0x0000_0000
CRPT_TDES2_KEY3L	CRYP_BA+0x29C	R/W	TDES Key 3 Low Word Register for Channel 2	0x0000_0000
CRPT_TDES3_KEY1H	CRYP_BA+0x2C8	R/W	TDES/DES Key 1 High Word Register for Channel 3	0x0000_0000
CRPT_TDES3_KEY1L	CRYP_BA+0x2CC	R/W	TDES/DES Key 1 Low Word Register for Channel 3	0x0000_0000
CRPT_TDES3_KEY2H	CRYP_BA+0x2D0	R/W	TDES Key 2 High Word Register for Channel 3	0x0000_0000

CRPT_TDES3_KEY2L	CRYP_BA+0x2D4	R/W	TDES Key 2 Low Word Register for Channel 3	0x0000_0000
CRPT_TDES3_KEY3H	CRYP_BA+0x2D8	R/W	TDES Key 3 High Word Register for Channel 3	0x0000_0000
CRPT_TDES3_KEY3L	CRYP_BA+0x2DC	R/W	TDES Key 3 Low Word Register for Channel 3	0x0000_0000

31	30	29	28	27	26	25	24
KEYH/KEYL							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
KEYH/KEYL							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
KEYH/KEYL							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
KEYH/KEYL							

Bits	Description	
[31:0]	**KEYH/KEYL**	**TDES/DES Key X High/Low Word** The key registers for TDES/DES algorithm calculation The security key for the TDES/DES accelerator is 64 bits. Thus, it needs two 32-bit registers to store a security key. The register CRPT_TDESn_KEYxH is used to keep the bit [63:32] of security key for the TDES/DES operation, while the register CRPT_TDESn_KEYxL is used to keep the bit [31:0].

TDES/DES IV High/Low Word Register (CRPT_TDES0_IVH/L, CRPT_TDES1_IVH/L, CRPT_TDES2_IVH/L, CRPT_TDES3_IVH/L)

Register	Offset	R/W	Description	Reset Value
CRPT_TDES0_IVH	CRYP_BA+0x220	R/W	TDES/DES Initial Vector High Word Register for Channel 0	0x0000_0000
CRPT_TDES0_IVL	CRYP_BA+0x224	R/W	TDES/DES Initial Vector Low Word Register for Channel 0	0x0000_0000
CRPT_TDES1_IVH	CRYP_BA+0x260	R/W	TDES/DES Initial Vector High Word Register for Channel 1	0x0000_0000
CRPT_TDES1_IVL	CRYP_BA+0x264	R/W	TDES/DES Initial Vector Low Word Register for Channel 1	0x0000_0000
CRPT_TDES2_IVH	CRYP_BA+0x2A0	R/W	TDES/DES Initial Vector High Word Register for Channel 2	0x0000_0000
CRPT_TDES2_IVL	CRYP_BA+0x2A4	R/W	TDES/DES Initial Vector Low Word Register for Channel 2	0x0000_0000
CRPT_TDES3_IVH	CRYP_BA+0x2E0	R/W	TDES/DES Initial Vector High Word Register for Channel 3	0x0000_0000
CRPT_TDES3_IVL	CRYP_BA+0x2E4	R/W	TDES/DES Initial Vector Low Word Register for Channel 3	0x0000_0000

31	30	29	28	27	26	25	24
IVH/IVL							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
IVH/IVL							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
IVH/IVL							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IVH/IVL							

Bits	Description	
[31:0]	**IVH/IVL**	**TDES/DES Initial Vector High/Low Word** Initial vector (IV) is for TDES/DES engine in CBC, CFB, and OFB mode. IV is Nonce counter for TDES/DES engine in CTR mode.

TDES/DES DMA Source Address Register (CRPT_TDES0_SADDR, CRPT_TDES1_SADDR, CRPT_TDES2_SADDR, CRPT_TDES3_SADDR)

Register	Offset	R/W	Description	Reset Value
CRPT_TDES0_SADDR	CRYP_BA+0x228	R/W	TDES/DES DMA Source Address Register for Channel 0	0x0000_0000
CRPT_TDES1_SADDR	CRYP_BA+0x268	R/W	TDES/DES DMA Source Address Register for Channel 1	0x0000_0000
CRPT_TDES2_SADDR	CRYP_BA+0x2A8	R/W	TDES/DES DMA Source Address Register for Channel 2	0x0000_0000
CRPT_TDES3_SADDR	CRYP_BA+0x2E8	R/W	TDES/DES DMA Source Address Register for Channel 3	0x0000_0000

31	30	29	28	27	26	25	24
SADDR[31:24]							
23	22	21	20	19	18	17	16
SADDR[23:16]							
15	14	13	12	11	10	9	8
SADDR[15:8]							
7	6	5	4	3	2	1	0
SADDR[7:0]							

Bits	Description	
[31:0]	**SADDR**	**TDES/DES DMA Source Address** The TDES/DES accelerator supports DMA function to transfer the plain text between system memory and embedded FIFO. The CRPT_TDESn_SADDR keeps the source address of the data buffer where the source text is stored. Based on the source address, the TDES/DES accelerator can read the plain text from system memory and do TDES/DES operation. The start of source address should be located at word boundary. In other words, bit 1 and 0 of CRPT_TDESn_SADDR are ignored. TDES_SADR can be read and written. Writing to TDES_SADR while the TDES/DES accelerator is operating doesn't affect the current TDES/DES operation. But the value of CRPT_TDESn_SADDR will be updated later on. Consequently, software can prepare the DMA source address for the next TDES/DES operation. In DMA mode, software can update the next CRPT_TDESn_SADDR before triggering START. CRPT_TDESn_SADDR and CRPT_TDESn_DADDR can be the same in the value.

TDES/DES DMA Destination Address Register (CRPT_TDES0_DADDR, CRPT_TDES1_DADDR, CRPT_TDES2_DADDR, CRPT_TDES3_DADDR)

Register	Offset	R/W	Description	Reset Value
CRPT_TDES0_DADDR	CRYP_BA+0x22C	R/W	TDES/DES DMA Destination Address Register for Channel 0	0x0000_0000
CRPT_TDES1_DADDR	CRYP_BA+0x26C	R/W	TDES/DES DMA Destination Address Register for Channel 1	0x0000_0000
CRPT_TDES2_DADDR	CRYP_BA+0x2AC	R/W	TDES/DES DMA Destination Address Register for Channel 2	0x0000_0000
CRPT_TDES3_DADDR	CRYP_BA+0x2EC	R/W	TDES/DES DMA Destination Address Register for Channel 3	0x0000_0000

31	30	29	28	27	26	25	24
DADDR[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DADDR[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DADDR[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DADDR[7:0]							

Bits	Description	
[31:0]	**DADDR**	**TDES/DES DMA Destination Address** The TDES/DES accelerator supports DMA function to transfer the cipher text between system memory and embedded FIFO. The CRPT_TDESn_DADDR keeps the destination address of the data buffer where the engine output's text will be stored. Based on the destination address, the TDES/DES accelerator can write the cipher text back to system memory after the TDES/DES operation is finished. The start of destination address should be located at word boundary. In other words, bit 1 and 0 of CRPT_TDESn_DADDR are ignored. TDES_DADR can be read and written. Writing to TDES_DADR while the TDES/DES accelerator is operating doesn't affect the current TDES/DES operation. But the value of CRPT_TDESn_DADDR will be updated later on. Consequently, software can prepare the destination address for the next TDES/DES operation. In DMA mode, software can update the next CRPT_TDESn_DADDR before triggering START. CRPT_TDESn_SADDR and CRPT_TDESn_DADDR can be the same in the value.

TDES/DES Block Count Register (CRPT_TDES0_CNT, CRPT_TDES1_CNT, CRPT_TDES2_CNT, CRPT_TDES3_CNT)

Register	Offset	R/W	Description	Reset Value
CRPT_TDES0_CNT	CRYP_BA+0x230	R/W	TDES/DES Byte Count Register for Channel 0	0x0000_0000
CRPT_TDES1_CNT	CRYP_BA+0x270	R/W	TDES/DES Byte Count Register for Channel 1	0x0000_0000
CRPT_TDES2_CNT	CRYP_BA+0x2B0	R/W	TDES/DES Byte Count Register for Channel 2	0x0000_0000
CRPT_TDES3_CNT	CRYP_BA+0x2F0	R/W	TDES/DES Byte Count Register for Channel 3	0x0000_0000

31	30	29	28	27	26	25	24
CNT[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CNT[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CNT[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNT[7:0]							

Bits	Description	
[31:0]	**CNT**	**TDES/DES Byte Count** The CRPT_TDESn_CNT keeps the byte count of source text that is for the TDES/DES engine operating in DMA mode. The CRPT_TDESn_CNT is 32-bit and the maximum of byte count is 4G bytes. CRPT_TDESn_CNT can be read and written. Writing to CRPT_TDESn_CNT while the TDES/DES accelerator is operating doesn't affect the current TDES/DES operation. But the value of CRPT_TDESn_CNT will be updated later on. Consequently, software can prepare the byte count of data for the next TDES /DES operation. In Non-DMA ECB, CBC, CFB, OFB, and CTR mode, TDES_CNT must be set as byte count for the last block of data before feeding in the last block of data.

TDES/DES Data Input Port Register (CRPT_TDES_DATIN)

Register	Offset	R/W	Description	Reset Value
CRPT_TDES_DATIN	CRYP_BA+0x234	R/W	TDES/DES Engine Input data Word Register	0x0000_0000

31	30	29	28	27	26	25	24
DATIN[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DATIN[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DATIN[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATIN[7:0]							

Bits	Description	
[31:0]	**DATIN**	**TDES/DES Engine Input Port** CPU feeds data to TDES/DES engine through this port by checking CRPT_TDES_STS. Feed data as INBUFFULL is 0.

TDES/DES Data Output Port Register (CRPT_TDES_DATOUT)

Register	Offset	R/W	Description	Reset Value
CRPT_TDES_DATOUT	CRYP_BA+0x238	R	TDES/DES Engine Output data Word Register	0x0000_0000

31	30	29	28	27	26	25	24
DATOUT[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DATOUT[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DATOUT[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATOUT[7:0]							

Bits	Description	
[31:0]	**DATOUT**	**TDES/DES Engine Output Port** CPU gets result from the TDES/DES engine through this port by checking CRPT_TDES_STS. Get data as OUTBUFEMPTY is 0.

TDES/DES Feedback x Register (CRPT_TDES_FEEDBACKx)

Register	Offset	R/W	Description	Reset Value
CRPT_TDES_FDBCKH	CRYP_BA+0x060	R	TDES/DES Engine Output Feedback High Word Data after Cryptographic Operation	0x0000_0000
CRPT_TDES_FDBCKL	CRYP_BA+0x064	R	TDES/DES Engine Output Feedback Low Word Data after Cryptographic Operation	0x0000_0000

31	30	29	28	27	26	25	24
FDBCK							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
FDBCK							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FDBCK							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FDBCK							

Bits	Description	
[31:0]	**FDBCK**	**TDES/DES Feedback** The feedback value is 64 bits in size. The TDES/DES engine uses the data from {CRPT_TDES_FDBCKH, CRPT_TDES_FDBCKL} as the data inputted to {CRPT_TDESn_IVH, CRPT_TDESn_IVL} for the next block in DMA cascade mode. The feedback register is for CBC, CFB, and OFB mode. TDES/DES engine outputs feedback information for IV in the next block's operation. Software can use this feedback information to implement more than four DMA channels. Software can store that feedback value temporarily. After switching back, fill the stored feedback value to this register in the same channel operation. Then can continue the operation with the original setting.

6.9.6.5 SHA Register

SHA Control Register (CRPT_SHA_CTL)

Register	Offset	R/W	Description	Reset Value
CRPT_SHA_CTL	CRYP_BA+0x300	R/W	SHA Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
INSWAP	**OUTSWAP**	**Reserved**					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**OPMODE**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAEN	**Reserved**	**DMALAST**	**Reserved**			**STOP**	**START**

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23]	**INSWAP**	**SHA Engine Input Data Swap** 0 = Keep the original order. 1 = The order that CPU feeds data to the accelerator will be changed from {byte3, byte2, byte1, byte0} to {byte0, byte1, byte2, byte3}.
[22]	**OUTSWAP**	**SHA Engine Output Data Swap** 0 = Keep the original order. 1 = The order that CPU feeds data to the accelerator will be changed from {byte3, byte2, byte1, byte0} to {byte0, byte1, byte2, byte3}.
[21:11]	**Reserved**	Reserved.
[10:8]	**OPMODE**	**SHA Engine Operation Modes** 000 = SHA160. 100 = SHA256. 101 = SHA224. **Note:** These bits can be read and written, but writing to them wouldn't take effect as BUSY is 1.
[7]	**DMAEN**	**SHA Engine DMA Enable Bit** 0 = SHA_DMA engine Disabled. The SHA engine operates in Non-DMA mode, and gets data from the port CRPT_SHA_DATIN. 1 = SHA_DMA engine Enabled. The SHA engine operates in DMA mode, and data movement from/to the engine is done by DMA logic.
[6]	**Reserved**	Reserved.

[5]	**DMALAST**	**SHA Last Block** In DMA mode, this bit must be set as beginning the last DMA cascade round. In Non-DMA mode, this bit must be set as feeding in last byte of data.
[4:2]	**Reserved**	Reserved.
[1]	**STOP**	**SHA Engine Stop** 0 = No effect. 1 = Stop SHA engine. **Note:** This bit is always 0 when it's read back.
[0]	**START**	**SHA Engine Start** 0 = No effect. 1 = Start SHA engine. BUSY flag will be set. **Note:** This bit is always 0 when it's read back.

SHA Flag Register (CRPT_SHA_STS)

Register	Offset	R/W	Description	Reset Value
CRPT_SHA_STS	CRYP_BA+0x304	R	SHA Status Flag	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							DATINREQ
15	14	13	12	11	10	9	8
Reserved							DMAERR
7	6	5	4	3	2	1	0
Reserved						DMABUSY	BUSY

Bits	Description	
[31:16]	**Reserved**	Reserved.
[16]	**DATINREQ**	**SHA Non-DMA Mode Data Input Request** 0 = No effect. 1 = Request SHA Non-DMA mode data input.
[15:9]	**Reserved**	Reserved.
[8]	**DMAERR**	**SHA Engine DMA Error Flag** 0 = Show the SHA engine access normal. 1 = Show the SHA engine access error.
[7:2]	**Reserved**	Reserved.
[1]	**DMABUSY**	**SHA Engine DMA Busy Flag** 0 = SHA DMA engine is idle or finished. 1 = SHA DMA engine is busy.
[0]	**BUSY**	**SHA Engine Busy** 0 = SHA engine is idle or finished. 1 = SHA engine is busy.

SHA Outputs Digest Word Register (CRPT_SHA_DGSTx)

Register	Offset	R/W	Description	Reset Value
CRPT_SHA_DGST0	CRYP_BA+0x308	R	SHA Digest Message 0	0x0000_0000
CRPT_SHA_DGST1	CRYP_BA+0x30C	R	SHA Digest Message 1	0x0000_0000
CRPT_SHA_DGST2	CRYP_BA+0x310	R	SHA Digest Message 2	0x0000_0000
CRPT_SHA_DGST3	CRYP_BA+0x314	R	SHA Digest Message 3	0x0000_0000
CRPT_SHA_DGST4	CRYP_BA+0x318	R	SHA Digest Message 4	0x0000_0000
CRPT_SHA_DGST5	CRYP_BA+0x31C	R	SHA Digest Message 5	0x0000_0000
CRPT_SHA_DGST6	CRYP_BA+0x320	R	SHA Digest Message 6	0x0000_0000
CRPT_SHA_DGST7	CRYP_BA+0x324	R	SHA Digest Message 7	0x0000_0000

31	30	29	28	27	26	25	24
DGST[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DGST[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DGST[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DGST[7:0]							

Bits	Description	
[31:0]	**DGST**	**SHA Digest Message Word** For SHA-160, the digest is stored in CRPT_SHA_DGST0 ~ CRPT_SHA_DGST4. For SHA-224, the digest is stored in CRPT_SHA_DGST0 ~ CRPT_SHA_DGST6. For SHA-256, the digest is stored in CRPT_SHA_DGST0 ~ CRPT_SHA_DGST7.

SHA Key Byte Count Register (CRPT_SHA_KEYCNT)

Register	Offset	R/W	Description	Reset Value
CRPT_SHA_KEYCNT	CRYP_BA+0x348	R/W	SHA Key Byte Count Register	0x0000_0000

31	30	29	28	27	26	25	24
KEYCNT							
23	22	21	20	19	18	17	16
KEYCNT							
15	14	13	12	11	10	9	8
KEYCNT							
7	6	5	4	3	2	1	0
KEYCNT							

Bits	Description	
[31:0]	KEYCNT	**SHA Key Byte Count** The CRPT_SHA_KEYCNT keeps the byte count of key that SHA engine operates. The register is 32-bit and the maximum byte count is 4G bytes. It can be read and written. Writing to the register CRPT_SHA_KEYCNT as the SHA accelerator operating doesn't affect the current SHA operation. But the value of CRPT_SHA_KEYCNT will be updated later on. Consequently, software can prepare the key count for the next SHA operation.

SHA DMA Source Address Register (CRPT_SHA_SADDR)

Register	Offset	R/W	Description	Reset Value
CRPT_SHA_SADDR	CRYP_BA+0x34C	R/W	SHA DMA Source Address Register	0x0000_0000

31	30	29	28	27	26	25	24
SADDR[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SADDR[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SADDR[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SADDR[7:0]							

Bits	Description	
[31:0]	**SADDR**	**SHA DMA Source Address** The SHA accelerator supports DMA function to transfer the plain text between system memory and embedded FIFO. The CRPT_SHA_SADDR keeps the source address of the data buffer where the source text is stored. Based on the source address, the SHA accelerator can read the plain text from system memory and do SHA operation. The start of source address should be located at word boundary. In other words, bit 1 and 0 of CRPT_SHA_SADDR are ignored. CRPT_SHA_SADDR can be read and written. Writing to CRPT_SHA_SADDR while the SHA accelerator is operating doesn't affect the current SHA operation. But the value of CRPT_SHA_SADDR will be updated later on. Consequently, software can prepare the DMA source address for the next SHA operation. In DMA mode, software can update the next TDES_SADR before triggering START. CRPT_SHA_SADDR and CRPT_SHA_DADDR can be the same in the value.

SHA Byte Count Register (CRPT_SHA_DMACNT)

Register	Offset	R/W	Description	Reset Value
CRPT_SHA_DMACNT	CRYP_BA+0x350	R/W	SHA Byte Count Register	0x0000_0000

31	30	29	28	27	26	25	24
DMACNT							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMACNT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMACNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMACNT							

Bits	Description	
[31:0]	**DMACNT**	**SHA Operation Byte Count** The CRPT_SHA_DMACNT keeps the byte count of source text that is for the SHA engine operating in DMA mode. The CRPT_SHA_DMACNT is 32-bit and the maximum of byte count is 4G bytes. CRPT_SHA_DMACNT can be read and written. Writing to CRPT_SHA_DMACNT while the SHA accelerator is operating doesn't affect the current SHA operation. But the value of CRPT_SHA_DMACNT will be updated later on. Consequently, software can prepare the byte count of data for the next SHA operation. In Non-DMA mode, CRPT_SHA_DMACNT must be set as the byte count of the last block before feeding in the last block of data.

SHA Data Input Port Register (CRPT_SHA_DATIN)

Register	Offset	R/W	Description	Reset Value
CRPT_SHA_DATIN	CRYP_BA+0x354	R/W	SHA Engine Non-DMA Mode Data Input Port Register	0x0000_0000

31	30	29	28	27	26	25	24
DATIN							
23	22	21	20	19	18	17	16
DATIN							
15	14	13	12	11	10	9	8
DATIN							
7	6	5	4	3	2	1	0
DATIN							

Bits	Description	
[31:0]	**DATIN**	**SHA Engine Input Port** CPU feeds data to SHA engine through this port by checking CRPT_SHA_STS. Feed data as DATINREQ is 1.

6.10 PDMA Controller (PDMA)

6.10.1 Overview

The direct memory access (PDMA) controller is used to provide high-speed data transfer. The PDMA controller can module transfers data from one address to another without CPU intervention. This has the benefit of reducing the workload of CPU and keeps CPU resources free for other applications. The PDMA controllers have a total of 16 channels and each channel can perform transfer between memory and peripherals or between memory and memory. The PDMA supports time-out function for each channel.

6.10.2 Features

- Supports 16 independently configurable channels.
- Selectable 2 level of priority (fixed priority or round-robin priority).
- Data size of 8, 16, and 32 bits.
- Source and destination address increment size by byte, half-word, word or no increment.
- Supports software or peripheral request, and the request type can be single or burst.
- Supports Scatter-Gather mode to perform sophisticated transfer through the use of the descriptor link list table.
- Supports time-out function for each channel.

6.10.3 Block Diagram

The block diagram about PDMA clock control is shown as follows.

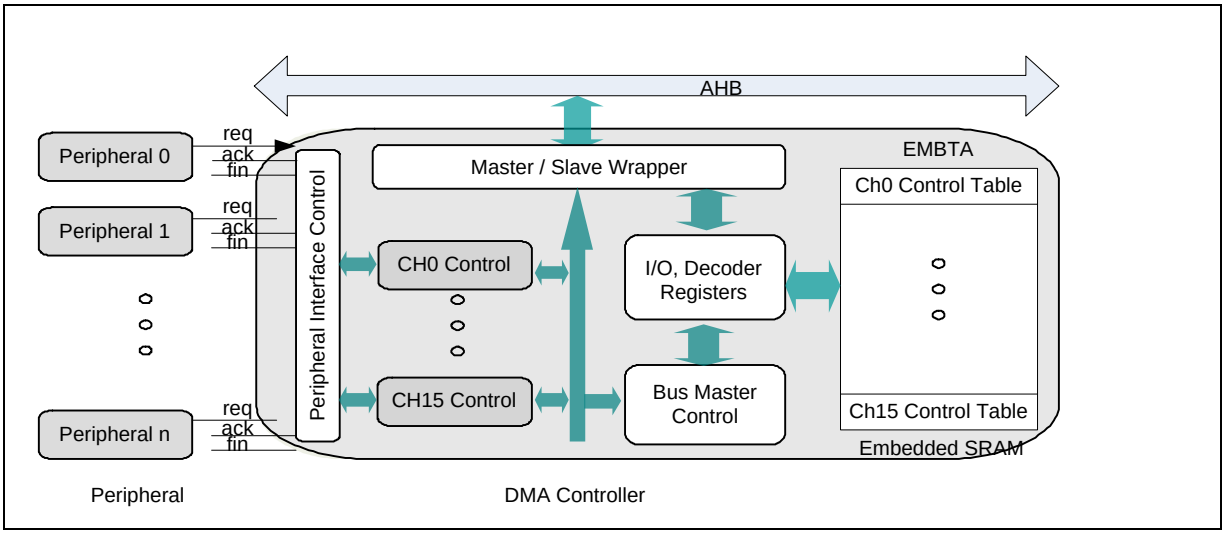


Figure 6.10-1 PDMA Controller Block Diagram

6.10.4 Functional Description

The direct memory access (PDMA) controller module transfers data from one address to another

without CPU intervention. The PDMA controller supports a single request type or burst request type and the request source can be from software request or peripheral request. A single request means that user or peripheral is ready to transfer one item (every item needs one request), and the burst request means that user or peripheral is ready to transfer multiple items (multiple items only need one request). The PDMA controller is embedded with an SRAM to store a set of channel control structures (embedded description table). The description table may have one or more entries for each PDMA channel, and thus the PDMA controller also supports sophisticated transfer mode through the use of more entries (Scatter-gather Operation mode). The following table shows the layout in SRAM of embedded descriptor table.

PDMA_EMBADR Offset	Channel
0x0000	PDMA Embedded Descriptor Table 0 (DSCT0)
0x0010	PDMA Embedded Descriptor Table 1 (DSCT1)
0x0020	PDMA Embedded Descriptor Table 2 (DSCT2)
0x0030	PDMA Embedded Descriptor Table 3 (DSCT3)
0x0040	PDMA Embedded Descriptor Table 4 (DSCT4)
0x0050	PDMA Embedded Descriptor Table 5 (DSCT5)
0x0060	PDMA Embedded Descriptor Table 6 (DSCT6)
0x0070	PDMA Embedded Descriptor Table 7 (DSCT7)
0x0080	PDMA Embedded Descriptor Table 8 (DSCT8)
0x0090	PDMA Embedded Descriptor Table 9 (DSCT9)
0x00A0	PDMA Embedded Descriptor Table 10 (DSCT10)
0x00B0	PDMA Embedded Descriptor Table 11 (DSCT11)
0x00C0	PDMA Embedded Descriptor Table 12 (DSCT12)
0x00D0	PDMA Embedded Descriptor Table 13 (DSCT13)
0x00E0	PDMA Embedded Descriptor Table 14 (DSCT14)
0x00F0	PDMA Embedded Descriptor Table 15 (DSCT15)

Table 6.10-1 Memory Map of Embedded Descriptor Table

Embedded Description Table Data Structure

The embedded description table data structure contains many transfer information including the transfer source address, transfer destination address, transfer count, burst size, request type and operation mode. The PDMA controller also supports sophisticated transfer mode through the use of the description link list table, and can perform more complex transfer such as wrapper-around mode or table link list (Scatter-Gather). The following figure shows the diagram of embedded description table data structure.

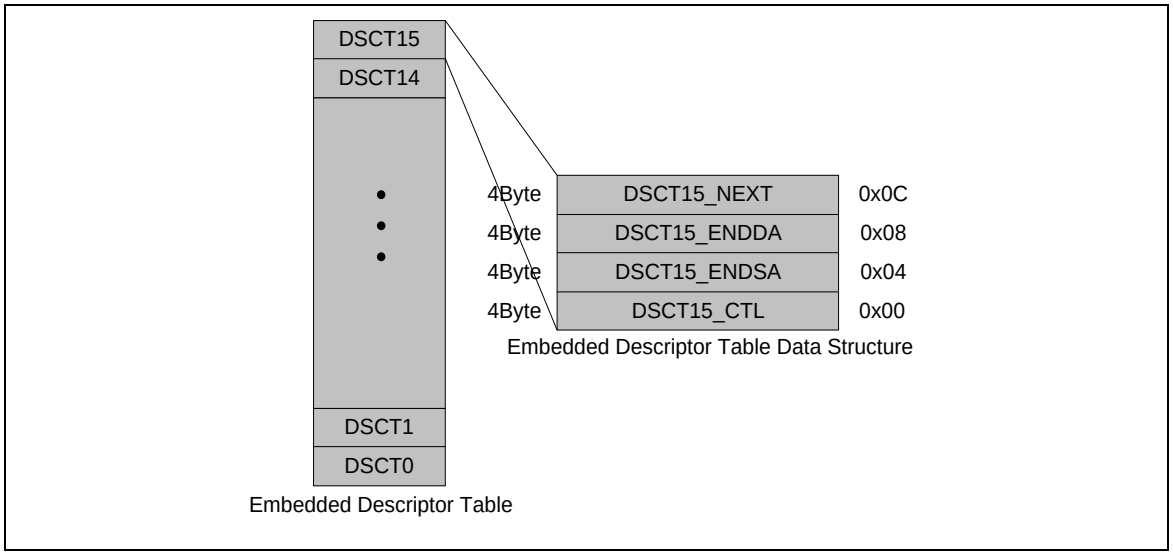


Figure 6.10-2 Embedded Description Table Data Structure

PDMA Operation Mode

The PDMA controller supports three operation modes including Stop mode, Basic mode, and Scatter-Gather mode.

Stop Mode

Stop mode is not a valid operation mode and can be used for software to indicate this description table channel is in use. When configuring the embedded description table, user must check if the operation mode is in stop mode. If the description table is not in stop mode, software re-configure channel setting may have some operation error. When the PDMA controller finishes this description table task, the operation mode will be updated to Stop mode. If software configures the operation mode to Stop mode, the PDMA controller will not perform any transfer and then clear this operation request. Finishing this task will generate an interrupt to CPU if each PDMA interrupt bit is enabled (PDMA_INTEN).

Basic Mode

Basic mode is used to perform one description table transfer mode. This mode can be used to transfer data between memory and memory or peripheral and memory. User must fill the transfer count in DSCTx_CTL [29:16] register and select transfer width, destination address increment size, source address increment size, burst size and request type, then the PDMA will perform PDMA operation. Finishing this task will generate an interrupt to CPU if each PDMA interrupt bit is enabled (PDMA_INTEN).

Scatter-Gather Mode

Scatter-Gather mode is a complex mode and can perform sophisticated transfer through the use of the description link list table. Through operation mode user can perform peripheral wrapper-around, multiple PDMA task or can be used for when data needs to be transfer to or from varied locations in system memory instead of a set of contiguous locations.

In Scatter-Gather mode, the embedded table is just used for jumping to the next table. The first embedded task will not perform any operation transfer. Finishing each task will generate an

interrupt to CPU if each PDMA interrupt bit is enabled (PDMA_INTEN) and DSCTx_CTL [7] bit is “0” (when finishing task and DSCTx_CTL [7] bit is “0”, each PDMA_TDSTS flag will be asserted and if this bit is “1” PDMA_TDSTS will not be active).

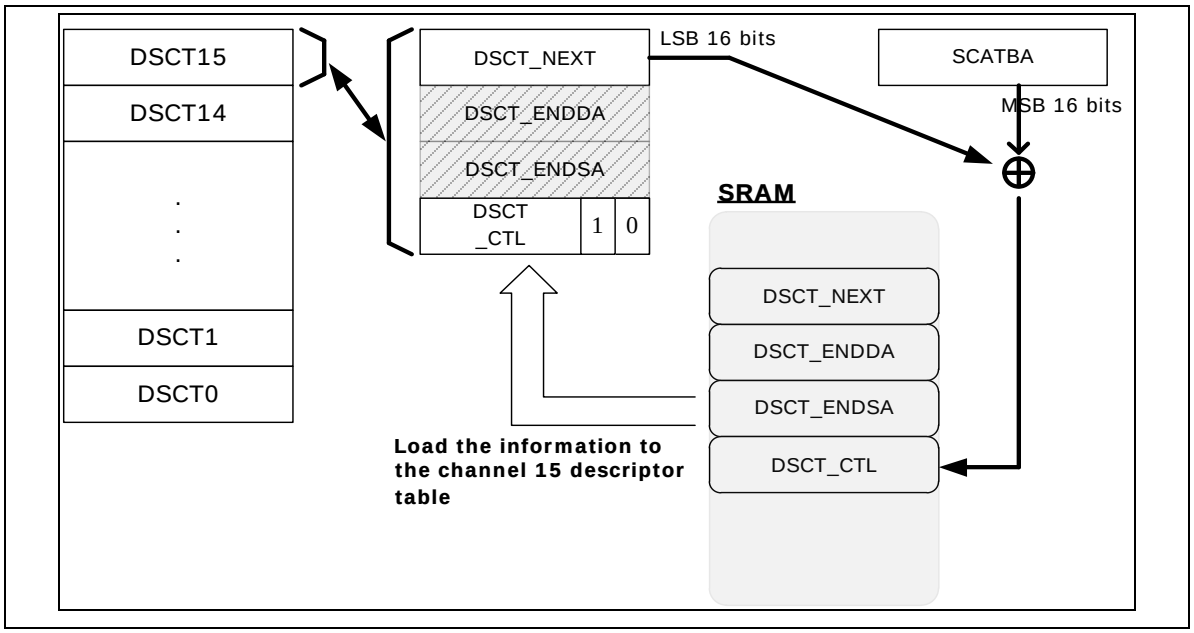


Figure 6.10-3 Embedded Description Table Data Structure

The figure above shows the block diagram of Scatter-Gather mode. If channel 15 has been triggered, and the operation mode is in Scatter-Gather (it means that DSCTx_CTL[1:0] are equal to 2’b10.), the hardware will load the real PDMA information task from the address generated by adding the DSCTx_NEXT and PDMA_SCATBA. When loading the information is finished, it will start by this information automatically. However, if the next PDMA information is also in the Scatter-Gather mode, the hardware will catch the next PDMA information block when the current task is finished. The Scatter-Gather stops until the PDMA operation mode is Basic mode or Stop mode.

Channel Priority

The PDMA controller supports two level channel priorities including fixed priority and round-robin priority. The fixed priority channel has higher priority than the normal priority channel. If multiple channels are set for fixed priority or round-robin priority, the higher channel will have higher priority (i.e. channel 15 has the highest priority).

Request Type

The PDMA controller supports two request types: the request type contains single request type mode and burst request type mode configure by setting DSCTx_CTL [2] register.

When operated in single request type, each transfer item needs one request signal until the transfer count DSCTx_CTL [29:16] decrease to 0. In this mode, the burst size (BURSIZE) is not useful to control the transfer size. The burst size will be fixed as one.

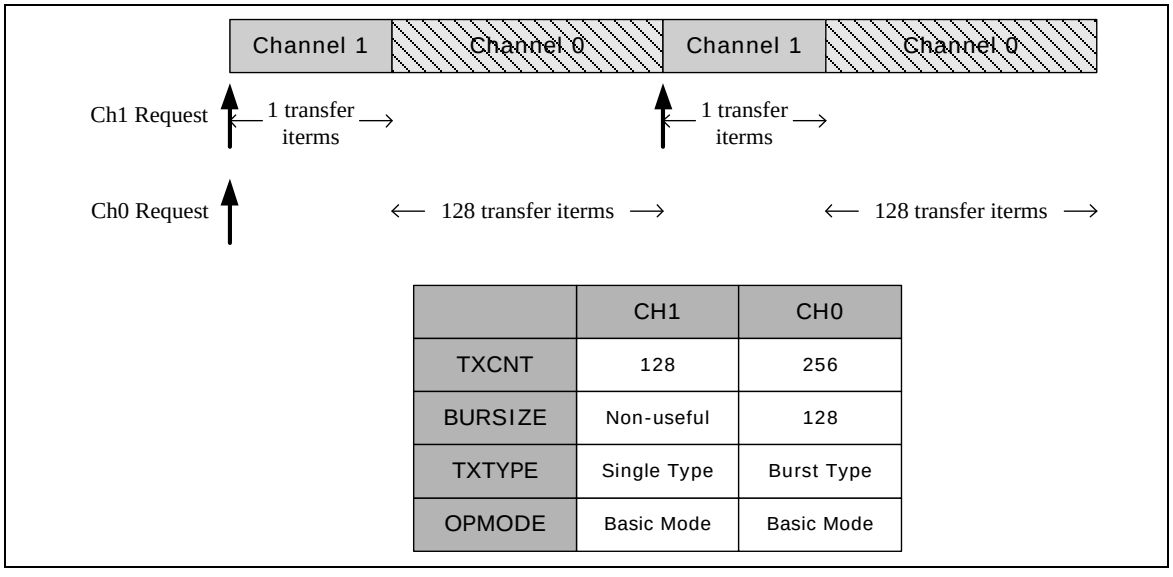
For the burst request type, multiple transfer items (BURSIZE) need only one request signal until the transfer count DSCTx_CTL [29:16] decrease to 0. This mode can used for peripheral PDMA to do burst request. But user must confirm if each peripheral supports burst request or not.

Channel Number	Priority Setting (PDMA_FPIOSET)	Arbitration Priority In Descending Order
15	Fixed Priority (Enable)	Highest
14	Fixed Priority (Enable)	---
---	---	---
0	Fixed Priority (Enable)	---
15	Round-Robin Priority (Disable)	---
14	Round-Robin Priority (Disable)	---
---	---	---
0	Round-Robin Priority (Disable)	Lowest

Table 6.10-2 Channel Priority Table

The following figure shows an example about single request type and burst request type. In this example, channel 1 uses single request type. Channel 0 uses burst request type and BURSIZE = 128.

(1) Channel 0 and channel 1 get the trigger signal at the same time.
(2) Because channel 1 has higher priority than channel 0 (default), the PDMA controller will load the channel 1 descriptor table and executed. But channel 1 is single request type, so PDMA controller will only access one transfer item.
(3) Then, PDMA controller turns to the channel 0 and loads channel 0 descriptor table. The channel 0 is burst request type and the burst size selected to 128. Therefore, PDMA controller will access 128 transfer items.
(4) After channel 0 finishes 128 transfer items, the PDMA controller will turn to channel 1 and wait the channel 1 trigger signal. If one trigger signal happens, PDMA controller also access one transfer item and turn to channel 0.



	CH1	CH0
TXCNT	128	256
BURSIZE	Non-useful	128
TXTYPE	Single Type	Burst Type
OPMODE	Basic Mode	Basic Mode

Figure 6.10-4 Embedded Description Table Data Structure

Channel time-out

When the PDMA channel is enabled and the channel has been selected to the peripheral, the channel's corresponding time-out counter will start count down, where counter is based on 10KHz clock. If time-out counter counts to zero, the PDMA controller will generate interrupt signal. By setting PDMA Time-out Period Counter Register to control time-out counter reload value (Channel 0 and Channel 1 by setting PDMA_TOC0_1, Channel 2 and Channel 3 by setting PDMA_TOC2_3 and so on).

In the general case, time-out counter reloads value is 0xffffffff, the PDMA time-out counter reloads counter period from PDMA Time-out Period Counter Register only when channel getting request or channel has been selected.

It is important to disable the PDMA channels' enable before enter power-down mode, because time-out counter still counting based on 10KHz clock. Without disable the PDMA channel's enable before enter power-down mode will generate unpredictable interrupt signal.

6.10.5 Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
PDMA Base Address: **PDMA_BA = 0x4000_8000**				
PDMA_DSCT0_CTL	PDMA_BA + 0x000	R/W	Descriptor Table Control Register of PDMA Channel 0	0xXXXX_XXXX
PDMA_DSCT0_ENDSA	PDMA_BA + 0x004	R/W	End Source Address Register of PDMA Channel 0	0xXXXX_XXXX
PDMA_DSCT0_ENDDA	PDMA_BA + 0x008	R/W	End Destination Address Register of PDMA Channel 0	0xXXXX_XXXX
PDMA_DSCT0_NEXT	PDMA_BA + 0x00C	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 0	0xXXXX_XXXX
PDMA_DSCT1_CTL	PDMA_BA + 0x010	R/W	Descriptor Table Control Register of PDMA Channel 1	0xXXXX_XXXX
PDMA_DSCT1_ENDSA	PDMA_BA + 0x014	R/W	End Source Address Register of PDMA Channel 1	0xXXXX_XXXX
PDMA_DSCT1_ENDDA	PDMA_BA + 0x018	R/W	End Destination Address Register of PDMA Channel 1	0xXXXX_XXXX
PDMA_DSCT1_NEXT	PDMA_BA + 0x01C	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 1	0xXXXX_XXXX
PDMA_DSCT2_CTL	PDMA_BA + 0x020	R/W	Descriptor Table Control Register of PDMA Channel 2	0xXXXX_XXXX
PDMA_DSCT2_ENDSA	PDMA_BA + 0x024	R/W	End Source Address Register of PDMA Channel 2	0xXXXX_XXXX
PDMA_DSCT2_ENDDA	PDMA_BA + 0x028	R/W	End Destination Address Register of PDMA Channel 2	0xXXXX_XXXX
PDMA_DSCT2_NEXT	PDMA_BA + 0x02C	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 2	0xXXXX_XXXX
PDMA_DSCT3_CTL	PDMA_BA + 0x030	R/W	Descriptor Table Control Register of PDMA Channel 3	0xXXXX_XXXX
PDMA_DSCT3_ENDSA	PDMA_BA + 0x034	R/W	End Source Address Register of PDMA Channel 3	0xXXXX_XXXX
PDMA_DSCT3_ENDDA	PDMA_BA + 0x038	R/W	End Destination Address Register of PDMA Channel 3	0xXXXX_XXXX
PDMA_DSCT3_NEXT	PDMA_BA + 0x03C	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 3	0xXXXX_XXXX
PDMA_DSCT4_CTL	PDMA_BA + 0x040	R/W	Descriptor Table Control Register of PDMA Channel 4	0xXXXX_XXXX
PDMA_DSCT4_ENDSA	PDMA_BA + 0x044	R/W	End Source Address Register of PDMA Channel 4	0xXXXX_XXXX
PDMA_DSCT4_ENDDA	PDMA_BA + 0x048	R/W	End Destination Address Register of PDMA Channel 4	0xXXXX_XXXX

PDMA_DSCT4_NEXT	PDMA_BA + 0x04C	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 4	0xXXXX_XXXX
PDMA_DSCT5_CTL	PDMA_BA + 0x050	R/W	Descriptor Table Control Register of PDMA Channel 5	0xXXXX_XXXX
PDMA_DSCT5_ENDSA	PDMA_BA + 0x054	R/W	End Source Address Register of PDMA Channel 5	0xXXXX_XXXX
PDMA_DSCT5_ENDDA	PDMA_BA + 0x058	R/W	End Destination Address Register of PDMA Channel 5	0xXXXX_XXXX
PDMA_DSCT5_NEXT	PDMA_BA + 0x05C	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 5	0xXXXX_XXXX
PDMA_DSCT6_CTL	PDMA_BA + 0x060	R/W	Descriptor Table Control Register of PDMA Channel 6	0xXXXX_XXXX
PDMA_DSCT6_ENDSA	PDMA_BA + 0x064	R/W	End Source Address Register of PDMA Channel 6	0xXXXX_XXXX
PDMA_DSCT6_ENDDA	PDMA_BA + 0x068	R/W	End Destination Address Register of PDMA Channel 6	0xXXXX_XXXX
PDMA_DSCT6_NEXT	PDMA_BA + 0x06C	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 6	0xXXXX_XXXX
PDMA_DSCT7_CTL	PDMA_BA + 0x070	R/W	Descriptor Table Control Register of PDMA Channel 7	0xXXXX_XXXX
PDMA_DSCT7_ENDSA	PDMA_BA + 0x074	R/W	End Source Address Register of PDMA Channel 7	0xXXXX_XXXX
PDMA_DSCT7_ENDDA	PDMA_BA + 0x078	R/W	End Destination Address Register of PDMA Channel 7	0xXXXX_XXXX
PDMA_DSCT7_NEXT	PDMA_BA + 0x07C	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 7	0xXXXX_XXXX
PDMA_DSCT8_CTL	PDMA_BA + 0x080	R/W	Descriptor Table Control Register of PDMA Channel 8	0xXXXX_XXXX
PDMA_DSCT8_ENDSA	PDMA_BA + 0x084	R/W	End Source Address Register of PDMA Channel 8	0xXXXX_XXXX
PDMA_DSCT8_ENDDA	PDMA_BA + 0x088	R/W	End Destination Address Register of PDMA Channel 8	0xXXXX_XXXX
PDMA_DSCT8_NEXT	PDMA_BA + 0x08C	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 8	0xXXXX_XXXX
PDMA_DSCT9_CTL	PDMA_BA + 0x090	R/W	Descriptor Table Control Register of PDMA Channel 9	0xXXXX_XXXX
PDMA_DSCT9_ENDSA	PDMA_BA + 0x094	R/W	End Source Address Register of PDMA Channel 9	0xXXXX_XXXX
PDMA_DSCT9_ENDDA	PDMA_BA + 0x098	R/W	End Destination Address Register of PDMA Channel 9	0xXXXX_XXXX
PDMA_DSCT9_NEXT	PDMA_BA + 0x09C	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 9	0xXXXX_XXXX
PDMA_DSCT10_CTL	PDMA_BA + 0x0A0	R/W	Descriptor Table Control Register of PDMA Channel 10	0xXXXX_XXXX
PDMA_DSCT10_ENDSA	PDMA_BA + 0x0A4	R/W	End Source Address Register of PDMA Channel 10	0xXXXX_XXXX

PDMA_DSCT10_ENDDA	PDMA_BA + 0x0A8	R/W	End Destination Address Register of PDMA Channel 10	0xXXXX_XXXX
PDMA_DSCT10_NEXT	PDMA_BA + 0x0AC	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 10	0xXXXX_XXXX
PDMA_DSCT11_CTL	PDMA_BA + 0x0B0	R/W	Descriptor Table Control Register of PDMA Channel 11	0xXXXX_XXXX
PDMA_DSCT11_ENDSA	PDMA_BA + 0x0B4	R/W	End Source Address Register of PDMA Channel 11	0xXXXX_XXXX
PDMA_DSCT11_ENDDA	PDMA_BA + 0x0B8	R/W	End Destination Address Register of PDMA Channel 11	0xXXXX_XXXX
PDMA_DSCT11_NEXT	PDMA_BA + 0x0BC	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 11	0xXXXX_XXXX
PDMA_DSCT12_CTL	PDMA_BA + 0x0C0	R/W	Descriptor Table Control Register of PDMA Channel 12	0xXXXX_XXXX
PDMA_DSCT12_ENDSA	PDMA_BA + 0x0C4	R/W	End Source Address Register of PDMA Channel 12	0xXXXX_XXXX
PDMA_DSCT12_ENDDA	PDMA_BA + 0x0C8	R/W	End Destination Address Register of PDMA Channel 12	0xXXXX_XXXX
PDMA_DSCT12_NEXT	PDMA_BA + 0x0CC	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 12	0xXXXX_XXXX
PDMA_DSCT13_CTL	PDMA_BA + 0x0D0	R/W	Descriptor Table Control Register of PDMA Channel 13	0xXXXX_XXXX
PDMA_DSCT13_ENDSA	PDMA_BA + 0x0D4	R/W	End Source Address Register of PDMA Channel 13	0xXXXX_XXXX
PDMA_DSCT13_ENDDA	PDMA_BA + 0x0D8	R/W	End Destination Address Register of PDMA Channel 13	0xXXXX_XXXX
PDMA_DSCT13_NEXT	PDMA_BA + 0x0DC	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 13	0xXXXX_XXXX
PDMA_DSCT14_CTL	PDMA_BA + 0x0E0	R/W	Descriptor Table Control Register of PDMA Channel 14	0xXXXX_XXXX
PDMA_DSCT14_ENDSA	PDMA_BA + 0x0E4	R/W	End Source Address Register of PDMA Channel 14	0xXXXX_XXXX
PDMA_DSCT14_ENDDA	PDMA_BA + 0x0E8	R/W	End Destination Address Register of PDMA Channel 14	0xXXXX_XXXX
PDMA_DSCT14_NEXT	PDMA_BA + 0x0EC	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 14	0xXXXX_XXXX
PDMA_DSCT15_CTL	PDMA_BA + 0x0F0	R/W	Descriptor Table Control Register of PDMA Channel 15	0xXXXX_XXXX
PDMA_DSCT15_ENDSA	PDMA_BA + 0x0F4	R/W	End Source Address Register of PDMA Channel 15	0xXXXX_XXXX
PDMA_DSCT15_ENDDA	PDMA_BA + 0x0F8	R/W	End Destination Address Register of PDMA Channel 15	0xXXXX_XXXX
PDMA_DSCT15_NEXT	PDMA_BA + 0x0FC	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 15	0xXXXX_XXXX
PDMA_CURSCAT0	PDMA_BA + 0x100	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 0	0x0000_0000

PDMA_CURSCAT1	PDMA_BA + 0x104	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 1	0x0000_0000
PDMA_CURSCAT2	PDMA_BA + 0x108	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 2	0x0000_0000
PDMA_CURSCAT3	PDMA_BA + 0x10C	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 3	0x0000_0000
PDMA_CURSCAT4	PDMA_BA + 0x110	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 4	0x0000_0000
PDMA_CURSCAT5	PDMA_BA + 0x114	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 5	0x0000_0000
PDMA_CURSCAT6	PDMA_BA + 0x118	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 6	0x0000_0000
PDMA_CURSCAT7	PDMA_BA + 0x11C	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 7	0x0000_0000
PDMA_CURSCAT8	PDMA_BA + 0x120	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 8	0x0000_0000
PDMA_CURSCAT9	PDMA_BA + 0x124	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 9	0x0000_0000
PDMA_CURSCAT10	PDMA_BA + 0x128	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 10	0x0000_0000
PDMA_CURSCAT11	PDMA_BA + 0x12C	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 11	0x0000_0000
PDMA_CURSCAT12	PDMA_BA + 0x130	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 12	0x0000_0000
PDMA_CURSCAT13	PDMA_BA + 0x134	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 13	0x0000_0000
PDMA_CURSCAT14	PDMA_BA + 0x138	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 14	0x0000_0000
PDMA_CURSCAT15	PDMA_BA + 0x13C	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 15	0x0000_0000
PDMA_CHCTL	PDMA_BA + 0x400	R/W	PDMA Channel Control Register	0x0000_0000
PDMA_STOP	PDMA_BA + 0x404	W	PDMA Stop Transfer Register	0x0000_0000
PDMA_SWREQ	PDMA_BA + 0x408	W	PDMA Software Request Register	0x0000_0000
PDMA_TRGSTS	PDMA_BA + 0x40C	R	PDMA Request Active Flag Register	0x0000_0000
PDMA_PRISET	PDMA_BA + 0x410	R/W	PDMA Fixed Priority Setting Register	0x0000_0000
PDMA_PRICLR	PDMA_BA + 0x414	W	PDMA Fixed Priority Clear Register	0x0000_0000
PDMA_INTEN	PDMA_BA + 0x418	R/W	PDMA Interrupt Enable Control Register	0x0000_0000
PDMA_INTSTS	PDMA_BA + 0x41C	R/W	PDMA Interrupt Status Register	0x0000_0000
PDMA_ABTSTS	PDMA_BA + 0x420	R/W	PDMA Read/Write Target Abort Flag Register	0x0000_0000
PDMA_TDSTS	PDMA_BA + 0x424	R/W	PDMA Transfer Done Flag Register	0x0000_0000
PDMA_SCATSTS	PDMA_BA + 0x428	R/W	PDMA Scatter-Gather Transfer Done Flag Register	0x0000_0000
PDMA_TACTSTS	PDMA_BA + 0x42C	R	PDMA Transfer on Active Flag Register	0x0000_0000

PDMA_SCATBA	PDMA_BA + 0x43C	R/W	PDMA Scatter-Gather Descriptor Table Base Address Register	0x2000_0000
PDMA_TOC0_1	PDMA_BA + 0x440	R/W	PDMA Time-out Period Counter Ch1 and Ch0 Register	0xFFFF_FFFF
PDMA_TOC2_3	PDMA_BA + 0x444	R/W	PDMA Time-out Period Counter Ch3 and Ch2 Register	0xFFFF_FFFF
PDMA_TOC4_5	PDMA_BA + 0x448	R/W	PDMA Time-out Period Counter Ch5 and Ch4 Register	0xFFFF_FFFF
PDMA_TOC6_7	PDMA_BA + 0x44C	R/W	PDMA Time-out Period Counter Ch7 and Ch6 Register	0xFFFF_FFFF
PDMA_TOC8_9	PDMA_BA + 0x450	R/W	PDMA Time-out Period Counter Ch9 and Ch8 Register	0xFFFF_FFFF
PDMA_TOC10_11	PDMA_BA + 0x454	R/W	PDMA Time-out Period Counter Ch11 and Ch10 Register	0xFFFF_FFFF
PDMA_TOC12_13	PDMA_BA + 0x458	R/W	PDMA Time-out Period Counter Ch13 and Ch12 Register	0xFFFF_FFFF
PDMA_TOC14_15	PDMA_BA + 0x45C	R/W	PDMA Time-out Period Counter Ch15 and Ch14 Register	0xFFFF_FFFF
PDMA_REQSEL0_3	PDMA_BA + 0x480	R/W	PDMA Source Module Select Register 0	0x1F1F_1F1F
PDMA_REQSEL4_7	PDMA_BA + 0x484	R/W	PDMA Source Module Select Register 1	0x1F1F_1F1F
PDMA_REQSEL8_11	PDMA_BA + 0x488	R/W	PDMA Source Module Select Register 2	0x1F1F_1F1F
PDMA_REQSEL12_15	PDMA_BA + 0x48C	R/W	PDMA Source Module Select Register 3	0x1F1F_1F1F

6.10.6 Register Description

PDMA Channel Control Register (PDMA_CHCTL)

Register	Offset	R/W	Description	Reset Value
PDMA_CHCTL	PDMA_BA + 0x400	R/W	PDMA Channel Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
CHEN15	CHEN14	CHEN13	CHEN12	CHEN11	CHEN10	CHEN9	CHEN8
7	6	5	4	3	2	1	0
CHEN7	CHEN6	CHEN5	CHEN4	CHEN3	CHEN2	CHEN1	CHEN0

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**CHEN[x]**	**PDMA Channel Enable Control Bit[X]** Set this bit to 1 to enable PDMA[x] operation. 0 = PDMA channel [x] Disabled. 1 = PDMA channel [x] Enabled. **Note1:** If software stops each PDMA transfer by setting PDMA_STOP register, this bit will be cleared automatically after finishing current transfer. **Note2:** Software reset (writing 0xFFFF_FFF to PDMA_STOP register) will clear this bit. **Note3:** If each channel is not set as enabled, each channel cannot be active.

Note: The x in the descriptor table represents the PDMA channel.

PDMA Stop Transfer Register (PDMA_STOP)

Register	Offset	R/W	Description	Reset Value
PDMA_STOP	PDMA_BA + 0x404	W	PDMA Stop Transfer Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
STOP15	**STOP14**	**STOP13**	**STOP12**	**STOP11**	**STOP10**	**STOP9**	**STOP8**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
STOP7	**STOP6**	**STOP5**	**STOP4**	**STOP3**	**STOP2**	**STOP1**	**STOP0**

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**STOP[x]**	**PDMA Stop Transfer Bit [X]** User can stop the PDMA transfer by software reset (writing all '1' to PDMA_STOP register) or by PDMA_STOP register. The difference between software reset and PDMA_STOP register is when software set software reset, the operation will be stopped imminently that include the on-going transfer and the channel enable bit and request active flag will be cleared to '0'. When software set PDMA_STOP bit, the operation will finish the on-going transfer channel and then clear the channel enable bit and request active flag. Software can poll channel enable bit to know if the on-going transfer is finished. 0 = No effect. 1 = Stop PDMA transfer[x]. **Note1:** This field is Write-Only; **Note2:** Setting all PDMA_STOP bit to "1" will generate software reset to reset internal state machine (the embedded table will not be reset).

Note: The x in the descriptor table represents the PDMA channel.

PDMA Software Request Register (PDMA_SWREQ)

Register	Offset	R/W	Description	Reset Value
PDMA_SWREQ	PDMA_BA + 0x408	W	PDMA Software Request Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SWREQ15	**SWREQ14**	**SWREQ13**	**SWREQ12**	**SWREQ11**	**SWREQ10**	**SWREQ9**	**SWREQ8**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SWREQ7	**SWREQ6**	**SWREQ5**	**SWREQ4**	**SWREQ3**	**SWREQ2**	**SWREQ1**	**SWREQ0**

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11:0]	**SWREQ[x]**	**PDMA Software Request Bit [X]** Set this bit to 1 to generate a software request to PDMA [x]. 0 = No effect. 1 = Generate a software request. **Note1:** This field is Write-Only. Software can indicate which channel is on active by reading PDMA_TRGSTS register. Active flag may be triggered by software request or peripheral request. **Note2:** If user does not enable each PDMA channel, the software request will be ignored.

Note: The x in the descriptor table represents the PDMA channel.

PDMA Request Active Flag Register (PDMA_TRGSTS)

Register	Offset	R/W	Description	Reset Value
PDMA_TRGSTS	PDMA_BA + 0x40C	R	PDMA Request Active Flag Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
REQSTS15	**REQSTS14**	**REQSTS13**	**REQSTS12**	**REQSTS11**	**REQSTS10**	**REQSTS9**	**REQSTS8**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
REQSTS7	**REQSTS6**	**REQSTS5**	**REQSTS4**	**REQSTS3**	**REQSTS2**	**REQSTS1**	**REQSTS0**

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**REQSTS[x]**	**PDMA Request Active Flag [X]** This flag indicates whether channel[x] have a request or not. 0 = Have no requests. 1 = Have a request. **Note1:** The request may come from software request (SWREQ) or peripheral request. **Note2:** When PDMA finishes channel transfer, this bit will be cleared automatically **Note3:** Software reset (setting PDMA_STOP to 0xFFFF_FFFF) will clear this bit.

Note: The x in the descriptor table represents the PDMA channel.

PDMA High Priority Setting Register (PDMA_PRISET)

Register	Offset	R/W	Description	Reset Value
PDMA_PRISET	PDMA_BA + 0x410	R/W	PDMA Fixed Priority Setting Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FPRISET15	**FPRISET14**	**FPRISET13**	**FPRISET12**	**FPRISET11**	**FPRISET10**	**FPRISET9**	**FPRISET8**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FPRISET7	**FPRISET6**	**FPRISET5**	**FPRISET4**	**FPRISET3**	**FPRISET2**	**FPRISET1**	**FPRISET0**

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**FPRISET[x]**	**PDMA Fixed Priority Setting Bit[X]** Set this bit to 1 to enable fix priority level. 0 = No effect. 1 = Set PDMA channel [x] be fixed priority channel. The PDMA channel priority is shown in the following table.

Note: The x in the descriptor table represents the PDMA channel.

PDMA Fix Priority Clear Register (PDMA_PRICLR)

Register	Offset	R/W	Description	Reset Value
PDMA_PRICLR	PDMA_BA + 0x414	W	PDMA Fixed Priority Clear Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
FPRICLR15	FPRICLR14	FPRICLR13	FPRICLR12	FPRICLR11	FPRICLR10	FPRICLR9	FPRICLR8
7	6	5	4	3	2	1	0
FPRICLR7	FPRICLR6	FPRICLR5	FPRICLR4	FPRICLR3	FPRICLR2	FPRICLR1	FPRICLR0

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**FPRICLR[x]**	**PDMA Fix Priority Clear Bit** Set this bit to 1 to clear fixed priority level. 0 = No effect. 1 = Set PDMA channel [x] to be round-robin priority channel. **Note:** This field is Write-Only, and software can indicate the channel priority by reading PDMA_PRISET register.

Note: The x in the descriptor table represents the PDMA channel.

PDMA Interrupt Enable Register (PDMA_INTEN)

Register	Offset	R/W	Description	Reset Value
PDMA_INTEN	PDMA_BA + 0x418	R/W	PDMA Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
INTEN15	**INTEN14**	**INTEN13**	**INTEN12**	**INTEN11**	**INTEN10**	**INTEN9**	**INTEN8**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INTEN7	**INTEN6**	**INTEN5**	**INTEN4**	**INTEN3**	**INTEN2**	**INTEN1**	**INTEN0**

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**INTEN[x]**	**PDMA Interrupt Enable** This field is used for enabling PDMA channel[x] interrupt. 0 = PDMA channel [x] interrupt Disabled. 1 = PDMA channel [x] interrupt Enabled.

Note: The x in the descriptor table represents the PDMA channel.

PDMA Interrupt Status Register (PDMA_INTSTS)

Register	Offset	R/W	Description	Reset Value
PDMA_INTSTS	PDMA_BA + 0x41C	R/W	PDMA Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
REQTOF15	REQTOF 14	REQTOF 13	REQTOF12	REQTOF11	REQTOF10	REQTOF9	REQTOF8
15	14	13	12	11	10	9	8
REQTOF7	REQTOF6	REQTOF5	REQTOF4	REQTOF3	REQTOF2	REQTOF1	REQTOF0
7	6	5	4	3	2	1	0
Reserved					TEIF	TDIF	ABTIF

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:8]	**REQTOFx**	**Time-Out Status Flag For Each Channel** This flag indicates that PDMA has waited peripheral request for a period defined by TIMECNTx 0 = No time-out flag. 1 = Time-out flag. **Note:** This field is Read only, but software can write 1 to clear it.
[7:3]	**Reserved**	Reserved.
[2]	**TEIF**	**Table Empty Interrupt Status Flag** This bit indicates that PDMA has finished each table transmission and the operation is Stop mode. Software can read PDMA_SCATSTS register to indicate which channel finished transfer. 0 = Not finished yet. 1 = PDMA channel has finished and the operation is Stop mode. **Note:** This field is Read only.
[1]	**TDIF**	**Transfer Done Interrupt Status Flag** This bit indicates that PDMA has finished transmission; Software can read PDMA_TDSTS register to indicate which channel finished transfer. 0 = Not finished yet. 1 = PDMA channel has finished transmission. **Note:** This field is Read only.
[0]	**ABTIF**	**PDMA Read/Write Target Abort Interrupt Status Flag** This bit indicates that PDMA has target abort error; Software can read PDMA_ABTSTS register to find which channel has target abort error. 0 = No bus ERROR response received. 1 = Bus ERROR response received. **Note:** This field is read only.

PDMA Target Abort Flag Register (PDMA_ABTSTS)

Register	Offset	R/W	Description	Reset Value
PDMA_ABTSTS	PDMA_BA + 0x420	R/W	PDMA Target Abort Flag Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
ABTIF15	ABTIF14	ABTIF13	ABTIF12	ABTIF11	ABTIF10	ABTIF9	ABTIF8
7	6	5	4	3	2	1	0
ABTIF7	ABTIF6	ABTIF5	ABTIF4	ABTIF3	ABTIF2	ABTIF1	ABTIF0

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**ABTIF**	**PDMA Target Abort Status Flag** This bit indicates which PDMA has target abort error; 0 = No bus ERROR response received. 1 = Bus ERROR response received. **Note:** This field is read only, but software can write 1 to clear it.

Note: The x in the descriptor table represents the PDMA channel.

PDMA Transfer Done Flag Register (PDMA_TDSTS)

Register	Offset	R/W	Description	Reset Value
PDMA_TDSTS	PDMA_BA + 0x424	R/W	PDMA Transfer Done Flag Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
TDIF15	TDIF14	TDIF13	TDIF12	TDIF11	TDIF10	TDIF9	TDIF8
7	6	5	4	3	2	1	0
TDIF7	TDIF6	TDIF5	TDIF4	TDIF3	TDIF2	TDIF1	TDIF0

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**TDIF[x]**	**Transfer Done Flag** This bit indicates which PDMA channel has finished transmission. 0 = Not finished yet. 1 = PDMA channel has finished transmission. **Note:** This field is read only, but software can write 1 to clear.

Note: The x in the descriptor table represents the PDMA channel.

PDMA Scatter-Gather Transfer Done Flag Register (PDMA_SCATSTS)

Register	Offset	R/W	Description	Reset Value
PDMA_SCATSTS	PDMA_BA + 0x428	R/W	PDMA Scatter-Gather Transfer Done Flag Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
TEMPTYF15	TEMPTYF14	TEMPTYF13	TEMPTYF12	TEMPTYF11	TEMPTYF10	TEMPTYF9	TEMPTYF8
7	6	5	4	3	2	1	0
TEMPTYF7	TEMPTYF6	TEMPTYF5	TEMPTYF4	TEMPTYF3	TEMPTYF2	TEMPTYF1	TEMPTYF0

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**TEMPTYF[x]**	**Table Empty Flag** This bit indicates which PDMA channel table has finished transmission and the operation mode is Stop mode; 0 = Not finished or not in Stop mode. 1 = PDMA channel has finished transmission and the operation is Stop mode. **Note:** This field is read only, but software can write 1 to clear.

Note: The x in the descriptor table represents the PDMA channel.

PDMA Transfer on Active Flag Register (PDMA_TACTSTS)

Register	Offset	R/W	Description	Reset Value
PDMA_TACTSTS	PDMA_BA + 0x42C	R	PDMA Transfer on Active Flag Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
TXACTF15	TXACTF14	TXACTF13	TXACTF12	TXACTF11	TXACTF10	TXACTF9	TXACTF8
7	6	5	4	3	2	1	0
TXACTF7	TXACTF6	TXACTF5	TXACTF4	TXACTF3	TXACTF2	TXACTF1	TXACTF0

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**TXACTF[x]**	**Transfer On Active Flag** This bit indicates which PDMA channel is on active. 0 = PDMA channel is not finished. 1 = PDMA channel is on active.

Note: The x in the descriptor table represents the PDMA channel.

PDMA Scatter-Gather Descriptor Table Base Address Register (PDMA_SCATBA)

Register	Offset	R/W	Description	Reset Value
PDMA_SCATBA	PDMA_BA + 0x43C	R/W	PDMA Scatter-Gather Descriptor Table Base Address Register	0x2000_0000

31	30	29	28	27	26	25	24
SCATBA [15:8]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SCATBA [7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:16]	**SCATBA**	**PDMA Scatter-Gather Descriptor Table Base Address** In Scatter-Gather mode, this is the base address for calculating the next link - list address. The next link address equation is Next Link Address = { SCATBA[15:0], DSCTx_NEXT[15:2], 2'b00}. **Note:** Only useful in Scatter-Gather mode.
[15:0]	**Reserved**	Reserved.

PDMA Time-out Period Counter Register (PDMA_TOC0_1)

Register	Offset	R/W	Description	Reset Value
PDMA_TOC0_1	PDMA_BA + 0x440	R/W	PDMA Time-out Period Counter Ch1 and Ch0 Register	0xFFFF_FFFF

31	30	29	28	27	26	25	24
TOC1[15:8]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TOC1[7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TOC0[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TOC0[7:0]							

Bits	Description	
[31:16]	**TOC1**	**Time-Out Period Counter For Channel 1** This controls the period of time-out function for channel 1. The calculation unit is based on 10 kHz clock.
[15:0]	**TOC0**	**Time-Out Period Counter For Channel 0** This controls the period of time-out function for channel 0. The calculation unit is based on 10 kHz clock.

PDMA Time-out Period Counter Register (PDMA_TOC2_3)

Register	Offset	R/W	Description	Reset Value
PDMA_TOC2_3	PDMA_BA + 0x444	R/W	PDMA Time-out Period Counter Ch3 and Ch2 Register	0xFFFF_FFFF

31	30	29	28	27	26	25	24
TOC3[15:8]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TOC3[7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TOC2[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TOC2[7:0]							

Bits	Description	
[31:16]	**TOC3**	**Time-Out Period Counter For Channel 3** This controls the period of time-out function for channel 3. The calculation unit is based on 10 kHz clock.
[15:0]	**TOC2**	**Time-Out Period Counter For Channel 2** This controls the period of time-out function for channel 2. The calculation unit is based on 10 kHz clock.

PDMA Time-out Period Counter Register (PDMA_TOC4_5)

Register	Offset	R/W	Description	Reset Value
PDMA_TOC4_5	PDMA_BA + 0x448	R/W	PDMA Time-out Period Counter Ch5 and Ch4 Register	0xFFFF_FFFF

31	30	29	28	27	26	25	24
TOC5[15:8]							
23	22	21	20	19	18	17	16
TOC5[7:0]							
15	14	13	12	11	10	9	8
TOC4[15:8]							
7	6	5	4	3	2	1	0
TOC4[7:0]							

Bits	Description	
[31:16]	TOC5	**Time-Out Period Counter For Channel 5** This controls the period of time-out function for channel 5. The calculation unit is based on 10 kHz clock.
[15:0]	TOC4	**Time-Out Period Counter For Channel 4** This controls the period of time-out function for channel 4. The calculation unit is based on 10 kHz clock.

PDMA Time-out Period Counter Register (PDMA_TOC6_7)

Register	Offset	R/W	Description	Reset Value
PDMA_TOC6_7	PDMA_BA + 0x44C	R/W	PDMA Time-out Period Counter Ch7 and Ch6 Register	0xFFFF_FFFF

31	30	29	28	27	26	25	24
TOC7[15:8]							
23	22	21	20	19	18	17	16
TOC7[7:0]							
15	14	13	12	11	10	9	8
TOC6[15:8]							
7	6	5	4	3	2	1	0
TOC6[7:0]							

Bits	Description	
[31:16]	TOC7	**Time-Out Period Counter For Channel 7** This controls the period of time-out function for channel 7. The calculation unit is based on 10 kHz clock.
[15:0]	TOC6	**Time-Out Period Counter For Channel 6** This controls the period of time-out function for channel 6. The calculation unit is based on 10 kHz clock.

PDMA Time-out Period Counter Register (PDMA_TOC8_9)

Register	Offset	R/W	Description	Reset Value
PDMA_TOC8_9	PDMA_BA + 0x450	R/W	PDMA Time-out Period Counter Ch9 and Ch8 Register	0xFFFF_FFFF

31	30	29	28	27	26	25	24
TOC9[15:8]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TOC9[7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TOC8[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TOC8[7:0]							

Bits	Description	
[31:16]	**TOC9**	**Time-Out Period Counter For Channel 9** This controls the period of time-out function for channel 9. The calculation unit is based on 10 kHz clock.
[15:0]	**TOC8**	**Time-Out Period Counter For Channel 8** This controls the period of time-out function for channel 8. The calculation unit is based on 10 kHz clock.

PDMA Time-out Period Counter Register (PDMA_TOC10_11)

Register	Offset	R/W	Description	Reset Value
PDMA_TOC10_11	PDMA_BA + 0x454	R/W	PDMA Time-out Period Counter Ch11 and Ch10 Register	0xFFFF_FFFF

31	30	29	28	27	26	25	24
TOC11[15:8]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TOC11[7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TOC10[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TOC10[7:0]							

Bits	Description	
[31:16]	**TOC11**	**Time-Out Period Counter For Channel 11** This controls the period of time-out function for channel 11. The calculation unit is based on 10 kHz clock.
[15:0]	**TOC10**	**Time-Out Period Counter For Channel 10** This controls the period of time-out function for channel 10. The calculation unit is based on 10 kHz clock.

PDMA Time-out Period Counter Register (PDMA_TOC12_13)

Register	Offset	R/W	Description	Reset Value
PDMA_TOC12_13	PDMA_BA + 0x458	R/W	PDMA Time-out Period Counter Ch13 and Ch12 Register	0xFFFF_FFFF

31	30	29	28	27	26	25	24
TOC13[15:8]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TOC13[7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TOC12[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TOC12[7:0]							

Bits	Description	
[31:16]	**TOC13**	**Time-Out Period Counter For Channel 13** This controls the period of time-out function for channel 13. The calculation unit is based on 10 kHz clock.
[15:0]	**TOC12**	**Time-Out Period Counter For Channel 12** This controls the period of time-out function for channel 12. The calculation unit is based on 10 kHz clock.

PDMA Time-out Period Counter Register (PDMA_TOC14_15)

Register	Offset	R/W	Description	Reset Value
PDMA_TOC14_15	PDMA_BA + 0x45C	R/W	PDMA Time-out Period Counter Ch15 and Ch14 Register	0xFFFF_FFFF

31	30	29	28	27	26	25	24
TOC15[15:8]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TOC15[7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TOC14[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TOC14[7:0]							

Bits	Description	
[31:16]	**TOC15**	**Time-Out Period Counter For Channel 15** This control the period of time-out function for channel 15. The calculation unit is based on 10 kHz clock.
[15:0]	**TOC14**	**Time-Out Period Counter For Channel 14** This control the period of time-out function for channel 14. The calculation unit is based on 10 kHz clock.

PDMA Source Module Select Register 0 (PDMA_REQSEL0_3)

Register	Offset	R/W	Description	Reset Value
PDMA_REQSEL0_3	PDMA_BA + 0x480	R/W	PDMA Source Module Select Register Ch0 ~ Ch3	0x1F1F_1F1F

31	30	29	28	27	26	25	24
Reserved			REQSRC3				
23	22	21	20	19	18	17	16
Reserved			REQSRC2				
15	14	13	12	11	10	9	8
Reserved			REQSRC1				
7	6	5	4	3	2	1	0
Reserved			REQSRC0				

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28:24]	**REQSRC3 [4:0]**	**Channel 3 Selection** This filed defines which peripheral is connected to PDMA channel 3. Software can configure the peripheral setting by REQSRC3. The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.
[23:21]	**Reserved**	Reserved.
[20:16]	**REQSRC2 [4:0]**	**Channel 2 Selection** This filed defines which peripheral is connected to PDMA channel 2. Software can configure the peripheral setting by REQSRC2. The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.
[15:13]	**Reserved**	Reserved.
[12:8]	**REQSRC1 [4:0]**	**Channel 1 Selection** This filed defines which peripheral is connected to PDMA channel 1. Software can configure the peripheral setting by REQSRC1. The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.
[7:5]	**Reserved**	Reserved.
[4:0]	**REQSRC0 [4:0]**	**Channel 0 Selection** This filed defines which peripheral is connected to PDMA channel 0. Software can configure the peripheral by setting REQSRC0. 00000 = Connect to SPI0_TX. 00001 = Connect to SPI1_TX. 00010 = Connect to SPI2_TX. 00011 = Connect to SPI3_TX. 00100 = Connect to UART0_TX. 00101 = Connect to UART1_TX. 00110 = Connect to UART2_TX.

Bits	Description	
		00111 = Connect to UART3_TX. 01000 = Connect to UART4_TX. 01001 = Connect to UART5_TX. 01010 = Reserved. 01011 = Connect to I2S_TX. 01100 = Connect to I2S1_TX. 01101 = Connect to SPI0_RX. 01110 = Connect to SPI1_RX. 01111 = Connect to SPI2_RX. 10000 = Connect to SPI3_RX. 10001 = Connect to UART0_RX. 10010 = Connect to UART1_RX. 10011 = Connect to UART2_RX. 10100 = Connect to UART3_RX. 10101 = Connect to UART4_RX. 10110 = Connect to UART5_RX. 10111 = Reserved. 11000 = Connect to ADC. 11001 = Connect to I2S_RX. 11010 = Connect to I2S1_RX. Other = Reserved.

PDMA Source Module Select Register 1 (PDMA_REQSEL4_7)

Register	Offset	R/W	Description	Reset Value
PDMA_REQSEL4_7	PDMA_BA + 0x484	R/W	PDMA Source Module Select Register Ch4 ~ Ch7	0x1F1F_1F1F

31	30	29	28	27	26	25	24
Reserved			REQSRC7				
23	22	21	20	19	18	17	16
Reserved			REQSRC6				
15	14	13	12	11	10	9	8
Reserved			REQSRC5				
7	6	5	4	3	2	1	0
Reserved			REQSRC4				

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28:24]	**REQSRC7 [4:0]**	**Channel 7 Selection** This filed defines which peripheral is connected to PDMA channel 7. Software can configure the peripheral setting by REQSRC7. The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.
[23:21]	**Reserved**	Reserved.
[20:16]	**REQSRC6 [4:0]**	**Channel 6 Selection** This filed defines which peripheral is connected to PDMA channel 6. Software can configure the peripheral setting by REQSRC6. The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.
[15:13]	**Reserved**	Reserved.
[12:8]	**REQSRC5 [4:0]**	**Channel 1 Selection** This filed defines which peripheral is connected to PDMA channel 5. Software can configure the peripheral setting by REQSRC5. The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.
[7:5]	**Reserved**	Reserved.
[4:0]	**REQSRC4 [4:0]**	**Channel 0 Selection** This filed defines which peripheral is connected to PDMA channel 4. Software can configure the peripheral setting by REQSRC4. The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.

PDMA Source Module Select Register 2 (PDMA_REQSEL8_11)

Register	Offset	R/W	Description	Reset Value
PDMA_REQSEL8_11	PDMA_BA + 0x488	R/W	PDMA Source Module Select Register Ch8 ~ Ch11	0x1F1F_1F1F

31	30	29	28	27	26	25	24
Reserved			REQSRC11				
23	22	21	20	19	18	17	16
Reserved			REQSRC10				
15	14	13	12	11	10	9	8
Reserved			REQSRC9				
7	6	5	4	3	2	1	0
Reserved			REQSRC8				

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28:24]	**REQSRC11 [4:0]**	**Channel 11 Selection** This filed defines which peripheral is connected to PDMA channel 11. Software can configure the peripheral setting by REQSRC11. The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.
[23:21]	**Reserved**	Reserved.
[20:16]	**REQSRC10 [4:0]**	**Channel 10 Selection** This filed defines which peripheral is connected to PDMA channel 10. Software can configure the peripheral setting by REQSRC10. The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.
[15:13]	**Reserved**	Reserved.
[12:8]	**REQSRC9 [4:0]**	**Channel 9 Selection** This filed defines which peripheral is connected to PDMA channel 9. Software can configure the peripheral setting by REQSRC9. The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.
[7:5]	**Reserved**	Reserved.
[4:0]	**REQSRC8 [4:0]**	**Channel 8 Selection** This filed defines which peripheral is connected to PDMA channel 8. Software can configure the peripheral setting by REQSRC8. The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.

PDMA Source Module Select Register 3 (PDMA_REQSEL12_15)

Register	Offset	R/W	Description	Reset Value
PDMA_REQSEL12_15	PDMA_BA + 0x48C	R/W	PDMA Source Module Select Register Ch12 ~ Ch15	0x1F1F_1F1F

31	30	29	28	27	26	25	24
Reserved			**REQSRC15**				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			**REQSRC14**				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			**REQSRC13**				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**REQSRC12**				

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28:24]	**REQSRC15 [4:0]**	**Channel 15 Selection** This filed defines which peripheral is connected to PDMA channel 15. Software can configure the peripheral setting by REQSRC15. The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.
[23:21]	**Reserved**	Reserved.
[20:16]	**REQSRC14 [4:0]**	**Channel 14 Selection** This filed defines which peripheral is connected to PDMA channel 14. Software can configure the peripheral setting by REQSRC14. The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.
[15:13]	**Reserved**	Reserved.
[12:8]	**REQSRC13 [4:0]**	**Channel 13 Selection** This filed defines which peripheral is connected to PDMA channel 13. Software can configure the peripheral setting by REQSRC13. The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.
[7:5]	**Reserved**	Reserved.
[4:0]	**REQSRC12 [4:0]**	**Channel 12 Selection** This filed defines which peripheral is connected to PDMA channel 12. Software can configure the peripheral setting by REQSRC12. The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.

Descriptor Table Control Register (PDMA_DSCTn_CTL) (n = 0~15)

Register	Offset	R/W	Description	Reset Value
PDMA_DSCT0_CTL	PDMA_BA + 0x000	R/W	Descriptor Table Control Register of PDMA Channel 0	0xXXXX_XXXX
PDMA_DSCT1_CTL	PDMA_BA + 0x010	R/W	Descriptor Table Control Register of PDMA Channel 1	0xXXXX_XXXX
PDMA_DSCT2_CTL	PDMA_BA + 0x020	R/W	Descriptor Table Control Register of PDMA Channel 2	0xXXXX_XXXX
PDMA_DSCT3_CTL	PDMA_BA + 0x030	R/W	Descriptor Table Control Register of PDMA Channel 3	0xXXXX_XXXX
PDMA_DSCT4_CTL	PDMA_BA + 0x040	R/W	Descriptor Table Control Register of PDMA Channel 4	0xXXXX_XXXX
PDMA_DSCT5_CTL	PDMA_BA + 0x050	R/W	Descriptor Table Control Register of PDMA Channel 5	0xXXXX_XXXX
PDMA_DSCT6_CTL	PDMA_BA + 0x060	R/W	Descriptor Table Control Register of PDMA Channel 6	0xXXXX_XXXX
PDMA_DSCT7_CTL	PDMA_BA + 0x070	R/W	Descriptor Table Control Register of PDMA Channel 7	0xXXXX_XXXX
PDMA_DSCT8_CTL	PDMA_BA + 0x080	R/W	Descriptor Table Control Register of PDMA Channel 8	0xXXXX_XXXX
PDMA_DSCT9_CTL	PDMA_BA + 0x090	R/W	Descriptor Table Control Register of PDMA Channel 9	0xXXXX_XXXX
PDMA_DSCT10_CTL	PDMA_BA + 0x0A0	R/W	Descriptor Table Control Register of PDMA Channel 10	0xXXXX_XXXX
PDMA_DSCT11_CTL	PDMA_BA + 0x0B0	R/W	Descriptor Table Control Register of PDMA Channel 11	0xXXXX_XXXX
PDMA_DSCT12_CTL	PDMA_BA + 0x0C0	R/W	Descriptor Table Control Register of PDMA Channel 12	0xXXXX_XXXX
PDMA_DSCT13_CTL	PDMA_BA + 0x0D0	R/W	Descriptor Table Control Register of PDMA Channel 13	0xXXXX_XXXX
PDMA_DSCT14_CTL	PDMA_BA + 0x0E0	R/W	Descriptor Table Control Register of PDMA Channel 14	0xXXXX_XXXX
PDMA_DSCT15_CTL	PDMA_BA + 0x0F0	R/W	Descriptor Table Control Register of PDMA Channel 15	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved		TXCNT					
23	22	21	20	19	18	17	16
TXCNT							
15	14	13	12	11	10	9	8
Reserved		TXWIDTH		DAINC		SAINC	
7	6	5	4	3	2	1	0

TBINTDIS	BURSIZE	Reserved	TXTYPE	OPMODE

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:16]	**TXCNT[13:0]**	**Transfer Count** The TXCNT represents the required number of PDMA transfer, the real transfer count is (TXCNT + 1); The maximum transfer count is 16384, every transfer may be byte, half-word or word that is dependent on TXWIDTH field. **Note:** When PDMA finish each transfer item, this field will be decrease imminently
[13:12]	**TXWIDTH[1:0]**	**Transfer Width Selection** This field is used for transfer width. 00 = 8 bits for every transfer item. 01 = 16 bits for every transfer item. 10 = 32 bits for every transfer item. 11 = Reserved. **Note:** The PDMA transfer source address (DSCTx_ENDSA) and PDMA transfer destination address (DSCTx_ENDDA) should be alignment under the TXWIDTH selection
[11:10]	**DAINC [1:0]**	**Destination Address Increment** This field is used to set the destination address increment size 11 = No Increment (Fixed Address.). Other = Increment and size is depended on TXWIDTH selection.
[9:8]	**SAINC[1:0]**	**Source Address Increment** This field is used to set the source address increment size 11 = No Increment (Fixed Address.). Other = Increment and size is depended on TXWIDTH selection.
[7]	**TBINTDIS**	**Table Interrupt Disable Bit** This field can be used to decide whether to enable table interrupt or not. When with transfer done flag, this bit is only used for scatter-gather mode. If the TBINTDIS bit is enabled when PDMA finishes this task, there will no interrupt generated. However, with the table empty flag, this bit is also useful. If it is set to '1", the TEMPTYF will not be set when this situation has happened. 0 = Table interrupt Enabled. 1 = Table interrupt Disabled.
[6:4]	**BURSIZE [2:0]**	**Burst Size** This field is used for peripheral to determine the burst size or used for determine the re-arbitration size. But if in Single Request Type, this field is not useful and only 1 transfer item been transmitted for each transfer 000 = 128 transfers. 001 = 64 transfers. 010 = 32 transfers. 011 = 16 transfers. 100 = 8 transfers. 101 = 4 transfers. 110 = 2 transfers. 111 = 1 transfers.

Bits	Description	
[3]	**Reserved**	Reserved.
[2]	**TXTYPE**	**Request Type** 0 = Burst request type. 1 = Single request type.
[1:0]	**OPMODE**	**PDMA Operation Mode Selection** 00 = Stop Mode. Channel is stopped or this table is complete, when PDMA finish channel table task, OPMODE will be cleared to stop mode automatically. 01 = Basic Mode. The descriptor table only has one task. When this task is finished, the PDMA_INTSTS[x] will be asserted. 10 = Scatter-Gather Mode. When operating in this mode, user must give the next descriptor table address in DSCTx_NEXT register; PDMA will ignore this task, and then load the next task to execute. **Note:** Before filling transfer task in the descriptor table, user must check if the descriptor table is complete.

Note: The x in the descriptor table represents the PDMA channel.

End Source Address Register (PDMA_DSCn_ENDSA) (n = 0~15)

Register	Offset	R/W	Description	Reset Value
PDMA_DSCT0_ENDSA	PDMA_BA + 0x004	R/W	End Source Address Register of PDMA Channel 0	0xXXXX_XXXX
PDMA_DSCT1_ENDSA	PDMA_BA + 0x014	R/W	End Source Address Register of PDMA Channel 1	0xXXXX_XXXX
PDMA_DSCT2_ENDSA	PDMA_BA + 0x024	R/W	End Source Address Register of PDMA Channel 2	0xXXXX_XXXX
PDMA_DSCT3_ENDSA	PDMA_BA + 0x034	R/W	End Source Address Register of PDMA Channel 3	0xXXXX_XXXX
PDMA_DSCT4_ENDSA	PDMA_BA + 0x044	R/W	End Source Address Register of PDMA Channel 4	0xXXXX_XXXX
PDMA_DSCT5_ENDSA	PDMA_BA + 0x054	R/W	End Source Address Register of PDMA Channel 5	0xXXXX_XXXX
PDMA_DSCT6_ENDSA	PDMA_BA + 0x064	R/W	End Source Address Register of PDMA Channel 6	0xXXXX_XXXX
PDMA_DSCT7_ENDSA	PDMA_BA + 0x074	R/W	End Source Address Register of PDMA Channel 7	0xXXXX_XXXX
PDMA_DSCT8_ENDSA	PDMA_BA + 0x084	R/W	End Source Address Register of PDMA Channel 8	0xXXXX_XXXX
PDMA_DSCT9_ENDSA	PDMA_BA + 0x094	R/W	End Source Address Register of PDMA Channel 9	0xXXXX_XXXX
PDMA_DSCT10_ENDSA	PDMA_BA + 0x0A4	R/W	End Source Address Register of PDMA Channel 10	0xXXXX_XXXX
PDMA_DSCT11_ENDSA	PDMA_BA + 0x0B4	R/W	End Source Address Register of PDMA Channel 11	0xXXXX_XXXX
PDMA_DSCT12_ENDSA	PDMA_BA + 0x0C4	R/W	End Source Address Register of PDMA Channel 12	0xXXXX_XXXX
PDMA_DSCT13_ENDSA	PDMA_BA + 0x0D4	R/W	End Source Address Register of PDMA Channel 13	0xXXXX_XXXX
PDMA_DSCT14_ENDSA	PDMA_BA + 0x0E4	R/W	End Source Address Register of PDMA Channel 14	0xXXXX_XXXX
PDMA_DSCT15_ENDSA	PDMA_BA + 0x0F4	R/W	End Source Address Register of PDMA Channel 15	0xXXXX_XXXX

31	30	29	28	27	26	25	24
ENDSA							
23	22	21	20	19	18	17	16
ENDSA							
15	14	13	12	11	10	9	8
ENDSA							
7	6	5	4	3	2	1	0

ENDSA

Bits	Description	
[31:0]	**ENDSA**	**PDMA Transfer Ending Source Address Bits** This field indicates a 32-bit ending source address of PDMA. **Note:** If the source start address is 0x2000_0000, the transfer count is 0x100 and the source address increment is word, this field must be filled 0x2000_0400. The equation is “0x2000_0400 = 0x2000_0000 + 0x100*4(word)”.

Note: The x in the descriptor table represents the PDMA channel.

End Destination Address Register (PDMA_DSCTn_ENDDA) (n = 0~15)

Register	Offset	R/W	Description	Reset Value
PDMA_DSCT0_ENDDA	PDMA_BA + 0x008	R/W	End Destination Address Register of PDMA Channel 0	0xXXXX_XXXX
PDMA_DSCT1_ENDDA	PDMA_BA + 0x018	R/W	End Destination Address Register of PDMA Channel 1	0xXXXX_XXXX
PDMA_DSCT2_ENDDA	PDMA_BA + 0x028	R/W	End Destination Address Register of PDMA Channel 2	0xXXXX_XXXX
PDMA_DSCT3_ENDDA	PDMA_BA + 0x038	R/W	End Destination Address Register of PDMA Channel 3	0xXXXX_XXXX
PDMA_DSCT4_ENDDA	PDMA_BA + 0x048	R/W	End Destination Address Register of PDMA Channel 4	0xXXXX_XXXX
PDMA_DSCT5_ENDDA	PDMA_BA + 0x058	R/W	End Destination Address Register of PDMA Channel 5	0xXXXX_XXXX
PDMA_DSCT6_ENDDA	PDMA_BA + 0x068	R/W	End Destination Address Register of PDMA Channel 6	0xXXXX_XXXX
PDMA_DSCT7_ENDDA	PDMA_BA + 0x078	R/W	End Destination Address Register of PDMA Channel 7	0xXXXX_XXXX
PDMA_DSCT8_ENDDA	PDMA_BA + 0x088	R/W	End Destination Address Register of PDMA Channel 8	0xXXXX_XXXX
PDMA_DSCT9_ENDDA	PDMA_BA + 0x098	R/W	End Destination Address Register of PDMA Channel 9	0xXXXX_XXXX
PDMA_DSCT10_ENDDA	PDMA_BA + 0x0A8	R/W	End Destination Address Register of PDMA Channel 10	0xXXXX_XXXX
PDMA_DSCT11_ENDDA	PDMA_BA + 0x0B8	R/W	End Destination Address Register of PDMA Channel 11	0xXXXX_XXXX
PDMA_DSCT12_ENDDA	PDMA_BA + 0x0C8	R/W	End Destination Address Register of PDMA Channel 12	0xXXXX_XXXX
PDMA_DSCT13_ENDDA	PDMA_BA + 0x0D8	R/W	End Destination Address Register of PDMA Channel 13	0xXXXX_XXXX
PDMA_DSCT14_ENDDA	PDMA_BA + 0x0E8	R/W	End Destination Address Register of PDMA Channel 14	0xXXXX_XXXX
PDMA_DSCT15_ENDDA	PDMA_BA + 0x0F8	R/W	End Destination Address Register of PDMA Channel 15	0xXXXX_XXXX

31	30	29	28	27	26	25	24
ENDDA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ENDDA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ENDDA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

ENDDA

Bits	Description	
[31:0]	**ENDDA [31:0]**	**PDMA Transfer Ending Destination Address Bits** This field indicates a 32-bit ending destination address of PDMA. **Note:** If the destination start address is 0x2000_0000, the transfer count is 0x100 and the destination address increment is word, this field must be filled 0x2000_0400. The equation is “0x2000_0400 = 0x2000_0000 + 0x100*4(word)”.

Note: The x in the descriptor table represents the PDMA channel.

Scatter-Gather Descriptor Table Offset Register (PDMA_DSCTn_NEXT) (n = 0~15)

Register	Offset	R/W	Description	Reset Value
PDMA_DSCT0_NEXT	PDMA_BA + 0x00C	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 0	0xXXXX_XXXX
PDMA_DSCT1_NEXT	PDMA_BA + 0x01C	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 1	0xXXXX_XXXX
PDMA_DSCT2_NEXT	PDMA_BA + 0x02C	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 2	0xXXXX_XXXX
PDMA_DSCT3_NEXT	PDMA_BA + 0x03C	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 3	0xXXXX_XXXX
PDMA_DSCT4_NEXT	PDMA_BA + 0x04C	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 4	0xXXXX_XXXX
PDMA_DSCT5_NEXT	PDMA_BA + 0x05C	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 5	0xXXXX_XXXX
PDMA_DSCT6_NEXT	PDMA_BA + 0x06C	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 6	0xXXXX_XXXX
PDMA_DSCT7_NEXT	PDMA_BA + 0x07C	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 7	0xXXXX_XXXX
PDMA_DSCT8_NEXT	PDMA_BA + 0x08C	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 8	0xXXXX_XXXX
PDMA_DSCT9_NEXT	PDMA_BA + 0x09C	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 9	0xXXXX_XXXX
PDMA_DSCT10_NEXT	PDMA_BA + 0x0AC	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 10	0xXXXX_XXXX
PDMA_DSCT11_NEXT	PDMA_BA + 0x0BC	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 11	0xXXXX_XXXX
PDMA_DSCT12_NEXT	PDMA_BA + 0x0CC	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 12	0xXXXX_XXXX
PDMA_DSCT13_NEXT	PDMA_BA + 0x0DC	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 13	0xXXXX_XXXX
PDMA_DSCT14_NEXT	PDMA_BA + 0x0EC	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 14	0xXXXX_XXXX
PDMA_DSCT15_NEXT	PDMA_BA + 0x0FC	R/W	Scatter-Gather Descriptor Table Offset Address of PDMA Channel 15	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
NEXT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

NEXT	Reserved

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:2]	**NEXT**	**PDMA Next Description Table Offset Address Bits** This field indicates the offset of next descriptor table address in system memory. **Note1:** The next descriptor table address must be word boundary. **Note2:** The system memory based address is 0x2000_0000 (PDMA_SCATBA), if the next descriptor table is 0x2000_0100, that this field must fill 0x0100. **Note3:** Before filled transfer task in the description table, user must check if the descriptor table is complete.
[1:0]	**Reserved**	Reserved.

Note: The x in the descriptor table represents the PDMA channel.

Current Scatter-Gather Descriptor Table Address of PDMA Channel[n] (n = 0~15)

Register	Offset	R/W	Description	Reset Value
PDMA_CURSCAT0	PDMA_BA + 0x100	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 0	0x0000_0000
PDMA_CURSCAT1	PDMA_BA + 0x104	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 1	0x0000_0000
PDMA_CURSCAT2	PDMA_BA + 0x108	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 2	0x0000_0000
PDMA_CURSCAT3	PDMA_BA + 0x10C	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 3	0x0000_0000
PDMA_CURSCAT4	PDMA_BA + 0x110	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 4	0x0000_0000
PDMA_CURSCAT5	PDMA_BA + 0x114	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 5	0x0000_0000
PDMA_CURSCAT6	PDMA_BA + 0x118	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 6	0x0000_0000
PDMA_CURSCAT7	PDMA_BA + 0x11C	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 7	0x0000_0000
PDMA_CURSCAT8	PDMA_BA + 0x120	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 8	0x0000_0000
PDMA_CURSCAT9	PDMA_BA + 0x124	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 9	0x0000_0000
PDMA_CURSCAT10	PDMA_BA + 0x128	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 10	0x0000_0000
PDMA_CURSCAT11	PDMA_BA + 0x12C	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 11	0x0000_0000
PDMA_CURSCAT12	PDMA_BA + 0x130	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 12	0x0000_0000
PDMA_CURSCAT13	PDMA_BA + 0x134	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 13	0x0000_0000
PDMA_CURSCAT14	PDMA_BA + 0x138	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 14	0x0000_0000
PDMA_CURSCAT15	PDMA_BA + 0x13C	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 15	0x0000_0000

31	30	29	28	27	26	25	24
CURADDR							
23	22	21	20	19	18	17	16
CURADDR							
15	14	13	12	11	10	9	8
CURADDR							
7	6	5	4	3	2	1	0

CURADDR

Bits	Description	
[31:0]	**CURADDR [31:0]**	**PDMA External Current Descriptor Address Bits** This field indicates a 32-bit current external descriptor address of PDMA. **Note:** This field is read only and only used for Scatter-Gather mode to indicate the current external descriptor address.

6.11 External Bus Interface (EBI)

6.11.1 Overview

The NUC442/NUC472 series is equipped with an external bus interface (EBI) for external device use. To minimize the connections between external device and this chip, EBI supports address bus and data bus multiplex mode. Also, the address latch enable (ALE) signal supported differentiate the address and data cycle.

In consideration of pin resource, address and Data separate mode can improve the EBI performance and save the address latch.

6.11.2 Features

External Bus Interface has the following functions:

- Four chip selects (nCS[0]~nCS[3])
- External devices with max. 32M-byte (8-bit data width)/64M-byte (16-bit data width) addressable space supported for each chip select (nCS[x])
- Variable external bus base clock (MCLK)
- 8-bit or 16-bit data width are supported for each chip select (nCS[x])
- Variable data access time (tACC), address latch enable time (tALE) and address hold time (tAHD) supported for each chip select (nCS[x])
- Address bus and data bus multiplex mode supported to save the address pins
- Address bus and data bus separate mode supported to have better performance
- Configurable idle cycle supported for different access condition: Write command finish (W2X), Read-to-Read (R2R) and Read-to-Write(R2W)

6.11.3 Block Diagram

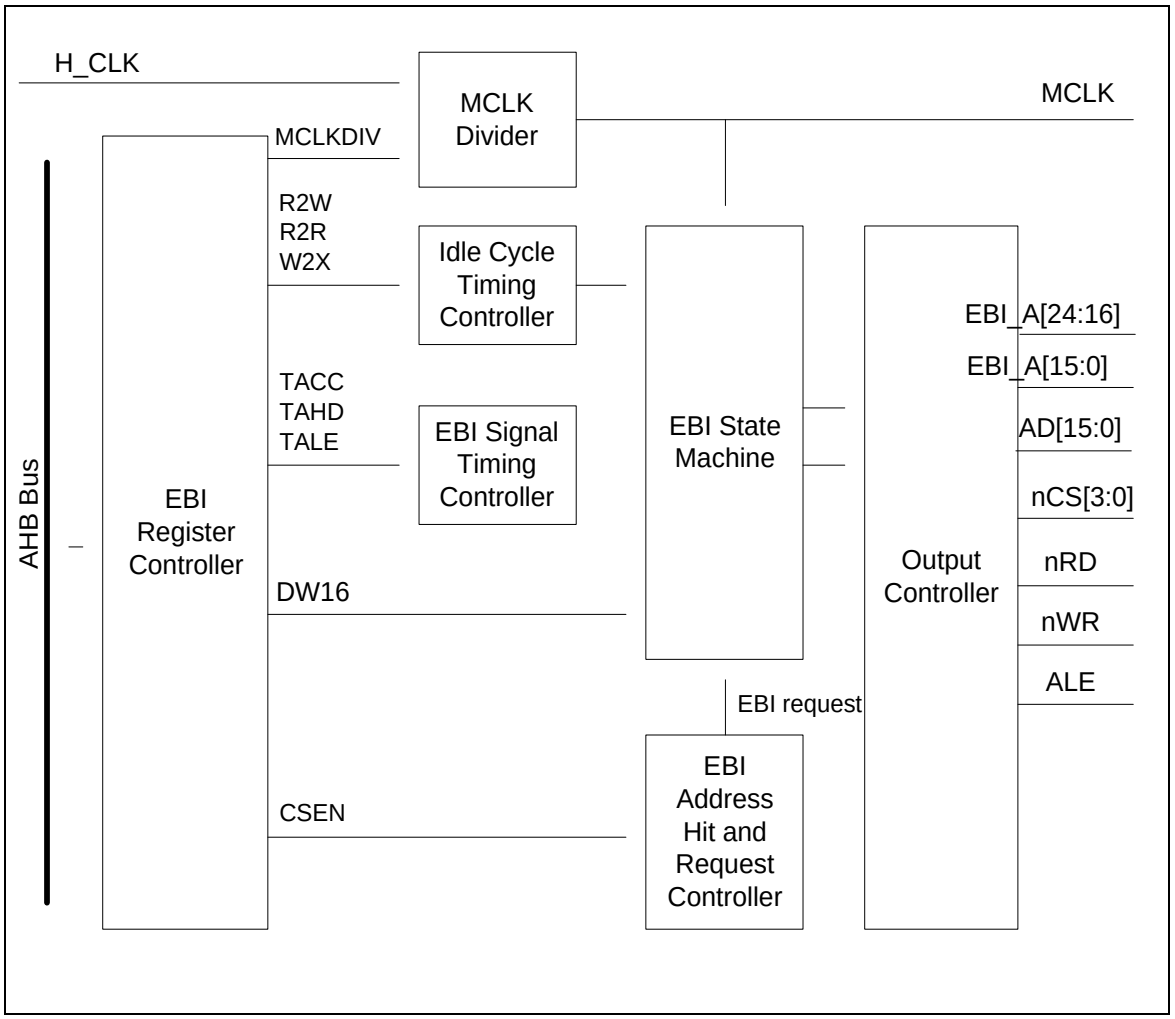


Figure 6.11-1 EBI Block Diagram

6.11.4 Functional Description

6.11.4.1 EBI Area and Address Hit

EBI mapping address is located at 0x6000_0000 ~ 0x6FFF_FFFF and the total memory space is 256Mbyte. When system request address hit EBI's memory space, the corresponding EBI chip select signal is assert and EBI state machine operates.

Chip Select	Address Mapping
nCS[0]	0x6000_0000 ~ 0x63FF_FFFF
nCS[1]	0x6400_0000 ~ 0x67FF_FFFF
nCS[2]	0x6800_0000 ~ 0x6BFF_FFFF
nCS[3]	0x6C00_0000 ~ 0x6FFF_FFFF

To map the whole EBI memory space, it requires 25-bit address for 16-bit device. For package that output less than 25-bit address, EBI will map device to mirror space. Ex: For package with 23-bit EBI address, EBI will mapped external device (for Bank0/nCS[0]) to 0x6000_0000 ~ 0x60FF_FFFF, 0x6100_0000 ~ 0x61FF_FFFF, 0x6200_0000 ~ 0x62FF_FFFF and 0x6300_0000 ~ 0x63FF_FFFF simultaneously.

For an 8-bit device (64Mbyte memory space for Bank0/nCS[0]), EBI mapped this 32Mbyte device to 0x6000_0000 ~ 0x61FF_FFFF and 0x6200_0000 ~ 0x63FF_FFFF simultaneously.

6.11.4.2 EBI Data Width Connection - Address Bus and Data Bus Multiplex Mode

EBI supports device whose address bus and data bus are multiplexed. For the external device with separated address and data bus, the connection to device needs additional logic to latch the address. In this case, pin ALE is connected to the latch device to latch the address value. Pin AD is the input of the latch device, and the output of the latch device is connected to the address of external device. For 16-bit device, the AD [15:0] shared by address and 16-bit data, A [24:16] is dedicated for address and could be connected to 8-bit device directly. For 8-bit device, only AD [7:0] shared by address and 8-bit data, A [24:8] is dedicated for address and could be connected to 8-bit device directly.

For 8-bit data width, chip system address bit [24:0] is used as the device's address [24:0] and the system address [26:25] will be decoded to be nCS[3:0]. For 16-bit data width, chip system address bit [25:1] is used as the device's address [24:0] and chip system address bit [0] is useless. The system address [27:26] will be decoded to be nCS[3:0].

EBI Bit Width	System Address (AHBADR)	EBI Address
8-bit	AHBADR[24:0]	EBI_A[24:0]
16-bit	AHBADR[25:1]	EBI_A[24:0]

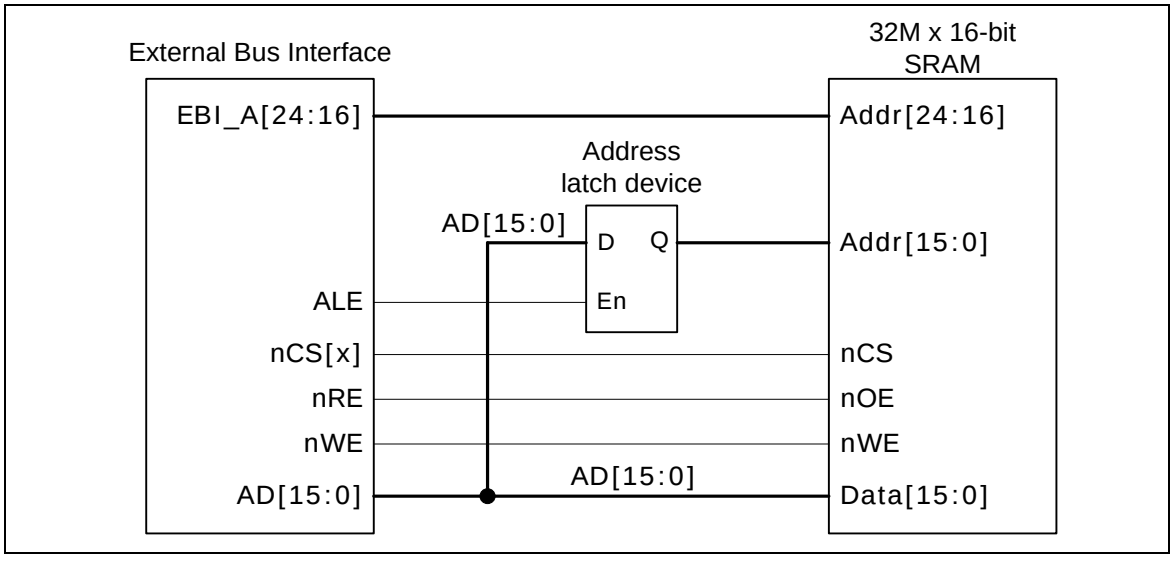


Figure 6.11-2 Connection of 16-bit EBI Data Width with 16-bit Device

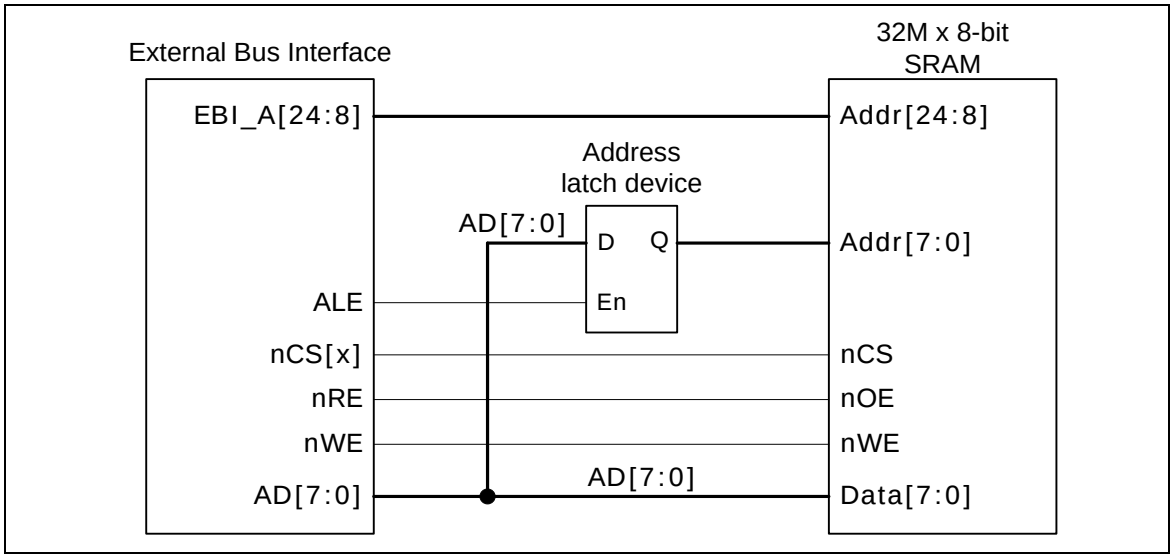


Figure 6.11-3 Connection of 8-bit EBI Data Width with 8-bit Device

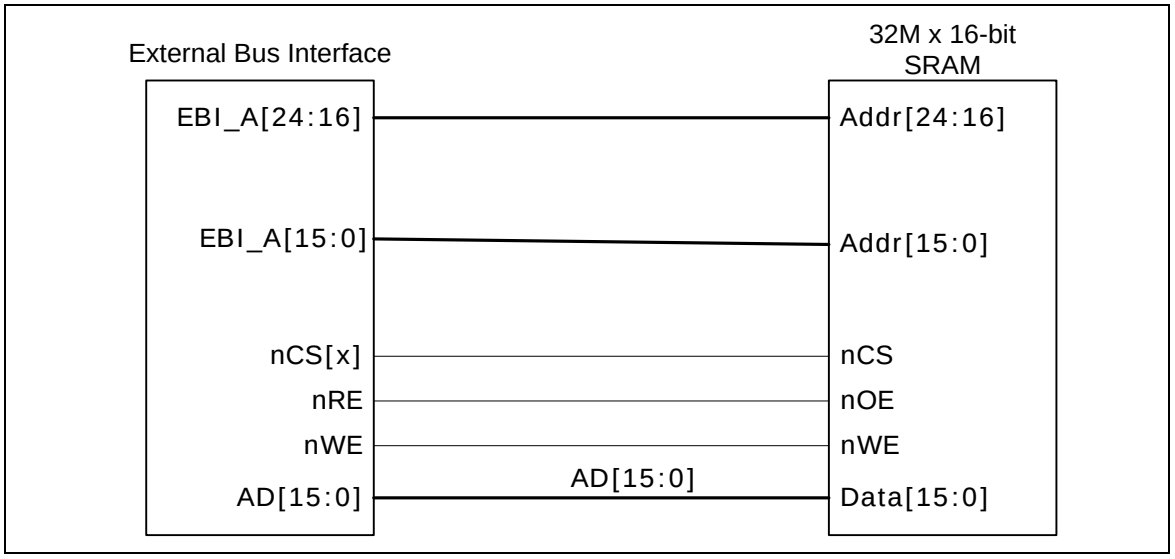


Figure 6.11-4 Connection of 16-bit EBI Data Width with 16-bit Device in Address/Data Separating Mode

When the system access data width is larger than the EBI data width, EBI controller will complete a system access command by operating EBI access more than once. For example, if system requests a 32-bit data through EBI device, EBI controller will operate accessing four times when setting EBI data width with 8-bit.

6.11.4.3 EBI Operating Control

MCLK Control

In the chip, all EBI signals will be synchronized by MCLK when EBI is operating. When chip connects to the external device with slower operating frequency, the MCLK can divide most to HCLK/32 by setting MCLKDIV(EBI_CTL[10:8]) of register EBI_CTL. Therefore, chip can suitable

for a wide frequency range of EBI device. If MCLK is set to HCLK/1, EBI signals are synchronized by positive edge of MCLK, else by negative edge of MCLK.

Operation and Access Timing Control

In the start of access, chip select (nCS[3:0]) asserts to low and wait one MCLK for address setup time (tASU) for address stable. Then ALE asserts to high after address is stable and keeps for a period of time (tALE) for address latch. After latch address, ALE asserts to low and wait one MCLK for latch hold time (tLHD) and another one MCLK cycle (tA2D) that is inserted behind address hold time to be the bus turn-around time for address change to data. Then nRD asserts to low when read access or nWR asserts to low when write access. Then nRD or nWR asserts to high after keeps access time (tACC) for reading output stable or writing finish. After that, EBI signals keep for data access hold time (tAHD) and chip select asserts to high, address is released by current access control.

EBI controller provides a flexible timing control for different external device. In EBI timing control, tASU, tLHD and tA2D are fixed to 1 MCLK cycle, tAHD can modulate to 1~8 MCLK cycles by setting TAHD of register EBI_TCTL, tACC can modulate to 1~32 MCLK cycles by setting TACC of register EBI_TCTL, and tALE can modulate to 1~8 MCLK cycles by setting tALE of register EBI_TCTL.

For each chip select, the EBI provides individual register for timing control.

Parameter	Value	Unit	Description
tASU	1	MCLK	Address Latch Setup Time.
tALE	1 ~ 8	MCLK	ALE High Period. Controlled by TALE(EBI_TCTL[2:0]) of EBI_TCTL.
tLHD	1	MCLK	Address Latch Hold Time.
tA2D	1	MCLK	Address To Data Delay (Bus Turn-Around Time).
tACC	1 ~ 32	MCLK	Data Access Time. Controlled by TACC(EBI_TCTL[7:3]) of EBI_TCTL.
tAHD	1 ~ 8	MCLK	Data Access Hold Time. Controlled by TAHD(EBI_TCTL[10:8]) of EBI_TCTL.
IDLE	0 ~ 15	MCLK	Idle Cycle. Controlled by R2R(EBI_TCTL[27:24]), R2W(EBI_TCTL[19:16]) and W2X(EBI_TCTL[15:12]) of EBI_TCTL.

Table 6.11-1 Timing Control Parameter Settings

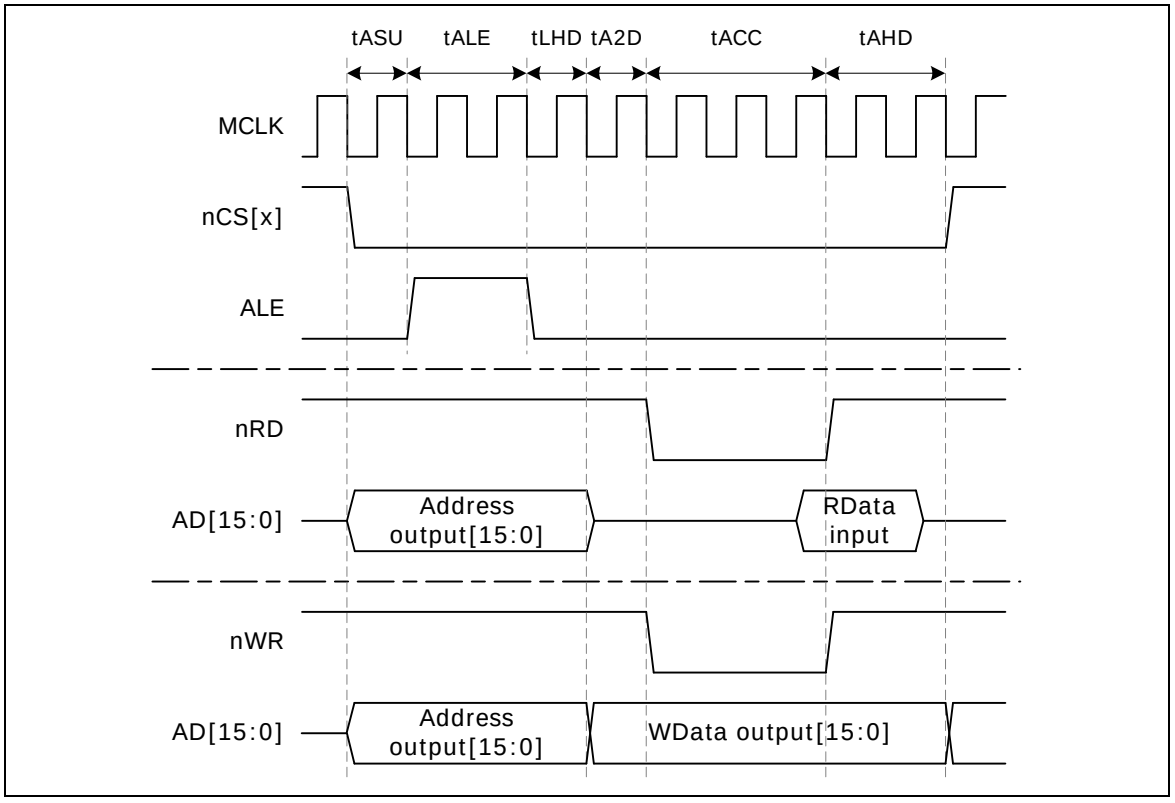


Figure 6.11-5 Timing Control Waveform for 16-bit Data Width

Figure 6.11-5 shows an example of setting 16-bit data width. In this example, AD bus is used for being address [15:0] and data [15:0]. When ALE assert to high, AD is address output. After address is latched, ALE asserts to low and the AD bus change to high impedance to wait device output data in read access operation, or it is used for being write data output.

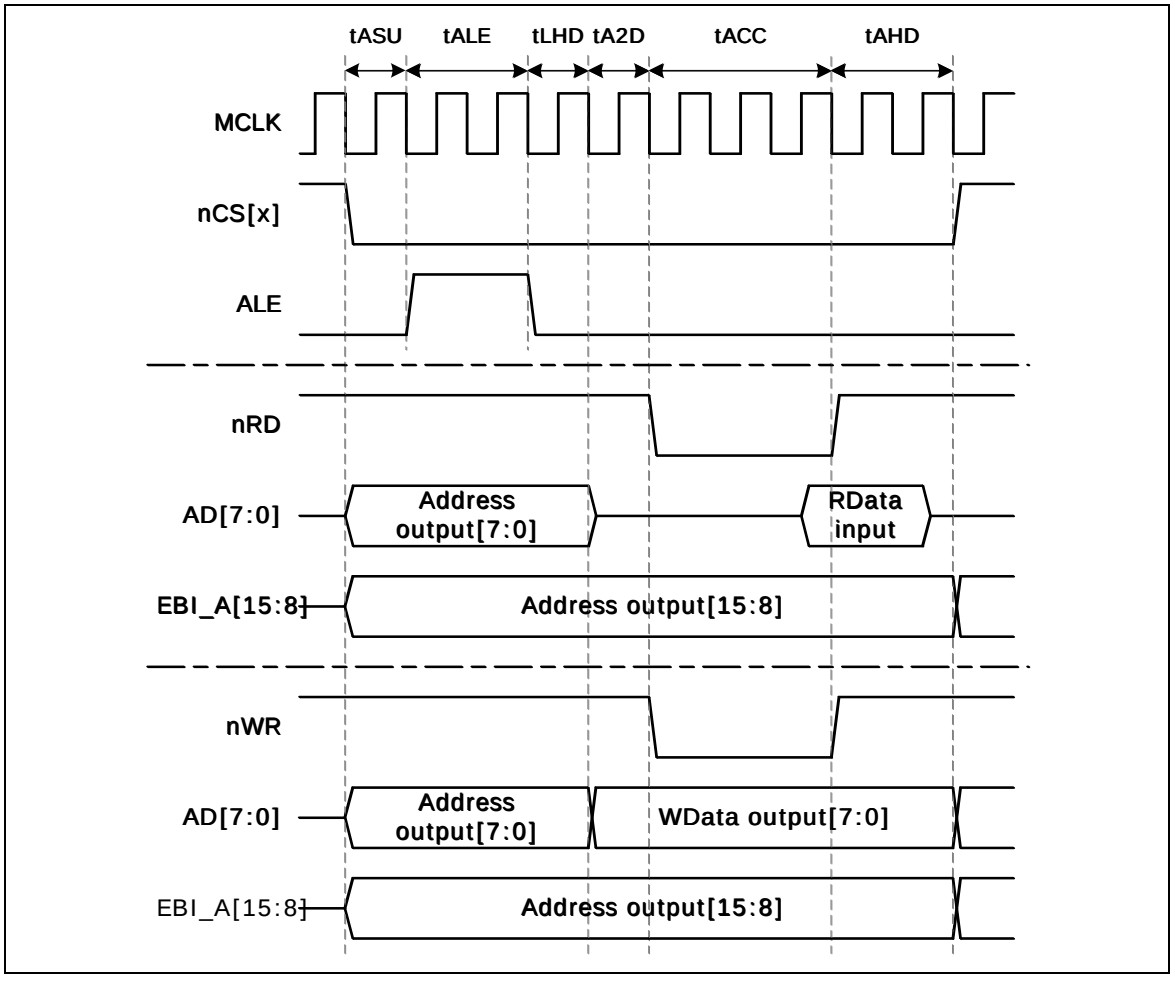


Figure 6.11-6 Timing Control Waveform for 8-bit Data Width

Figure 6.11-6 shows an example of setting 8-bit data width. The difference between 8-bit and 16-bit data width is AD [15:8]. In 8-bit data width setting, AD [15:8] always be Address [15:8] output so that external latch need only 8-bit width.

Insert Idle Cycle

When EBI accessing continuously, there may occur bus conflict if the device access time is much slow with system operating. EBI controller supply additional idle cycle to solve this problem. During idle cycle, all control signals of EBI are inactive. The following figure shows idle cycle:

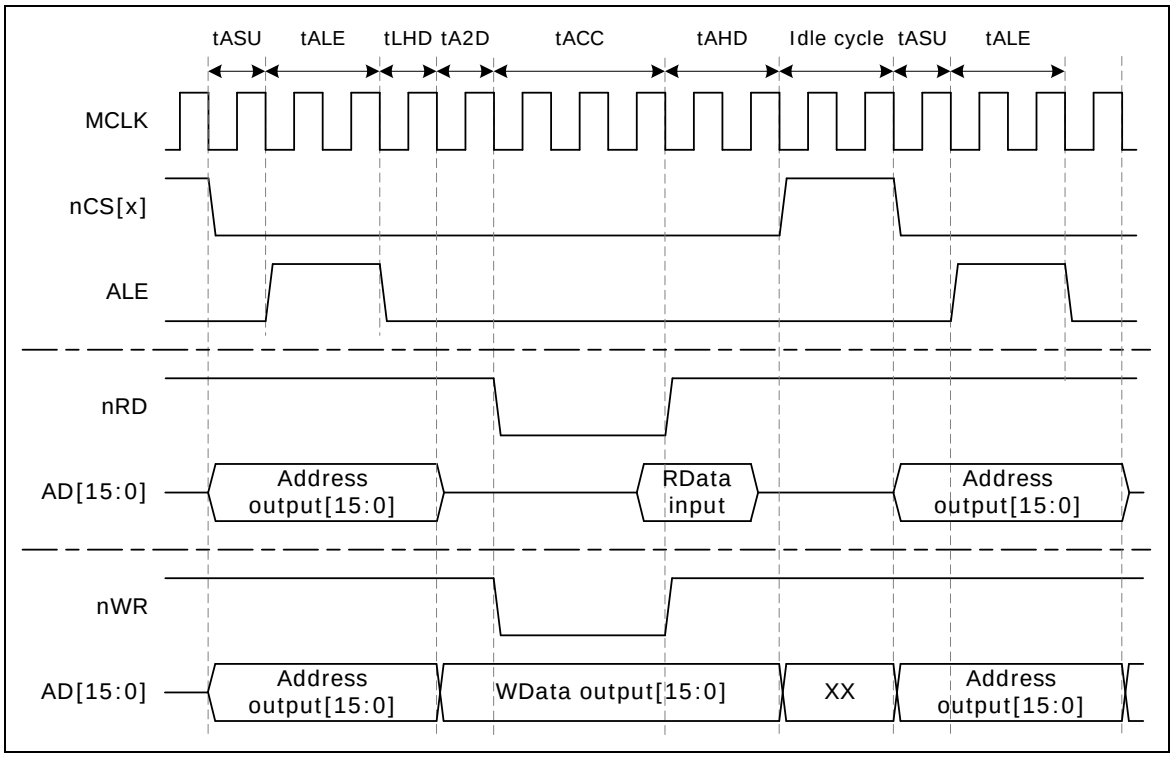


Figure 6.11-7 Timing Control Waveform for Insert Idle Cycle

There are three kind of conditions that EBI can insert idle cycle by timing control:

1. After write access
2. After read access and before next read access
3. After read access and before next write access

By setting W2X(EBI_TCTL[15:12]), R2R(EBI_TCTL[27:24]) and R2W(EBI_TCTL[19:16]) of register EBI_TCTL, the time of idle cycle can be specified from 1~15 MCLK.

Address & Data Separate Mode

For pin resource is not issue, address and data separate mode can better the EBI performance and save the address latch. The following figure shows address and data separate mode:

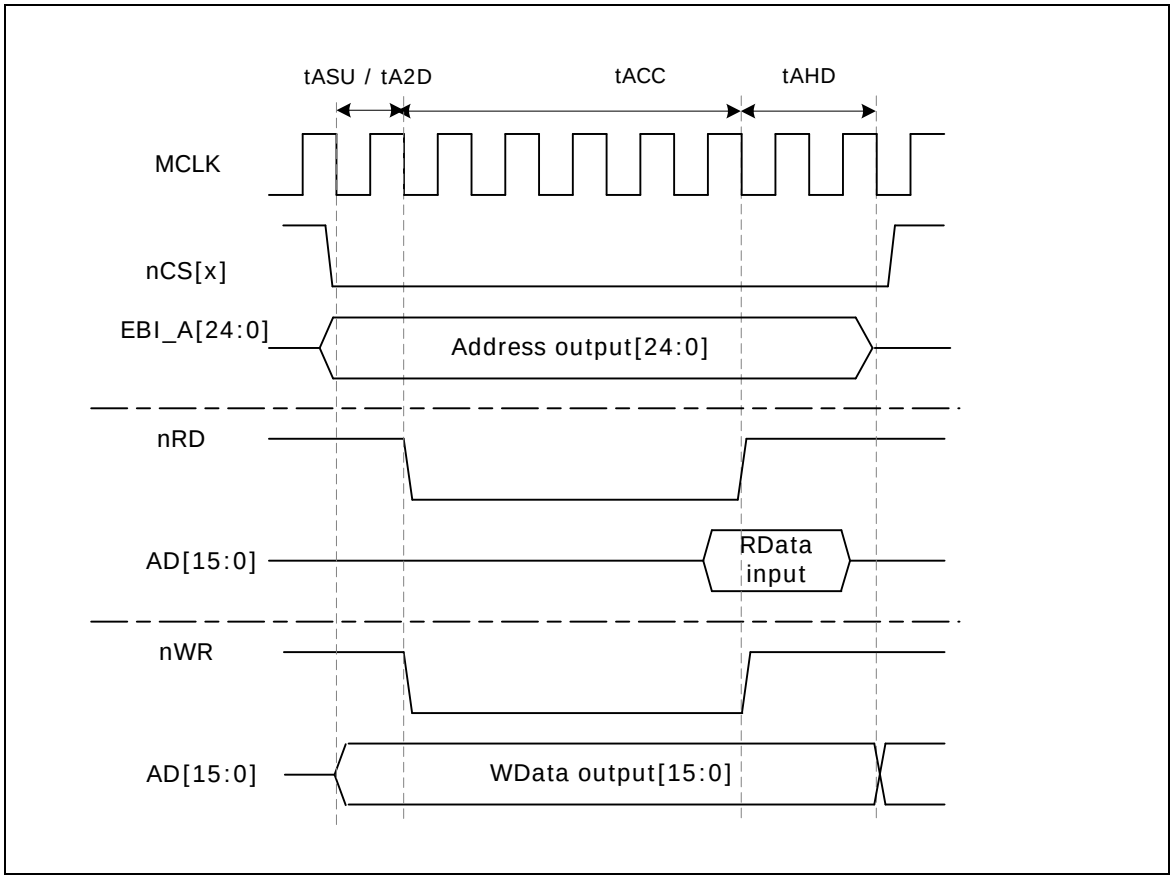


Figure 6.11-8 Timing Control Waveform for Address & Data Separate Mode (16-bit Data Width)

6.11.4.4 EBI Crypto Function

EBI supports the transparent crypto function to protect data storing in external device. The crypto engine contains linear and non-linear operations. User can enable/disable this function for each individual bank selection (EBI_nCS [3:0]) by setting CRYPTOEN(EBI_CTL [27:24]). Enabling this function will increase data latency about 5 ns. The crypto function is convenient for pure data type accessing sink like SRAM operations and not suits for register type accessing sink like flash.

For example, the user can firstly set the crypto 128 bit-width key (EBI_KEY0, EBI_KEY1, EBI_KEY2 and EBI_KEY3) and turn on the crypto function (EBI_CTL[24], menas nCS[0]). Then the following read/write access to nCS[0] will be processed by the transparent Decrypt/Encrypt function. On the above case, the other accesses to nCS[3:1] will be treated as original operation without crypto function. By user's application, each bank can be individually enabled/disabled crypto functionality.

6.11.5 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
EBI Base Address: **EBI_BA = 0x4001_0000**				
EBI_CTL	EBI_BA+0x00	R/W	External Bus Interface General Control Register	0x0000_0000
EBI_TCTL0	EBI_BA+0x04	R/W	External Bus Interface Bank0 Timing Control Register	0x0000_0000
EBI_TCTL1	EBI_BA+0x08	R/W	External Bus Interface Bank1 Timing Control Register	0x0000_0000
EBI_TCTL2	EBI_BA+0x0c	R/W	External Bus Interface Bank2 Timing Control Register	0x0000_0000
EBI_TCTL3	EBI_BA+0x10	R/W	External Bus Interface Bank3 Timing Control Register	0x0000_0000
EBI_KEY0	EBI_BA+0x14	R/W	External Bus Interface Crypto Key Word 0	0x0000_0000
EBI_KEY1	EBI_BA+0x18	R/W	External Bus Interface Crypto Key Word 1	0x0000_0000
EBI_KEY2	EBI_BA+0x1c	R/W	External Bus Interface Crypto Key Word 2	0x0000_0000
EBI_KEY3	EBI_BA+0x20	R/W	External Bus Interface Crypto Key Word 3	0x0000_0000

6.11.6 Register Description

External Bus Interface Control Register (EBI_CTL)

Register	Offset	R/W	Description	Reset Value
EBI_CTL	EBI_BA+0x00	R/W	External Bus Interface General Control Register	0x0000_0000

31	30	29	28	27	26	25	24
CSPOLINV				**CRYPTOEN**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reversed							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reversed					**MCLKDIV**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reversed							

Bits	Description	
[31:28]	**CSPOLINV**	**Reverse Chip Select** The original design Chip Select is active low nCS. ”Chip Select Active High” can be specified by customers—Bit[28+n] is for nCS[n], where n=0~3. 0 = nCS (chip select active low). 1 = CS (chip select active high).
[27:24]	**CRYPTOEN**	**Encrypt/Decrypt Function Enable Bits (For 4 Individual Chip Select)** 0 = Encrypt/Decrypt function Disabled. 1 = Encrypt/Decrypt function Enabled.
[23:11]	**Reserved**	Reserved.
[10:8]	**MCLKDIV**	**External Output Clock Divider** The frequency of EBI output clock is controlled by MCLKDIV as below: 000 = HCLK/1. 001 = HCLK/2. 010 = HCLK/4. 011 = HCLK/8. 100 = HCLK/16. 101 = HCLK/32. 11x = Default. **Note:** Default value of output clock is HCLK/1
[7:0]	**Reserved**	Reserved.

External Bus Interface Timing Control Register0 (EBI_TCTL0)

Register	Offset	R/W	Description	Reset Value
EBI_TCTL0	EBI_BA+0x04	R/W	External Bus Interface Bank0 Timing Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	SEPEN	DW16	CSEN	R2R			
23	22	21	20	19	18	17	16
Reserved				R2W			
15	14	13	12	11	10	9	8
W2X				Reversed	TAHD		
7	6	5	4	3	2	1	0
TACC					TALE		

Bits	Description	
[31]	**Reserved**	Reserved.
[30]	**SEPEN**	**EBI Bank0 Address/Data Bus Separating Enable Bit** 0 = Address/Data Bus Separating Disabled. 1 = Address/Data Bus Separating Enabled.
[29]	**DW16**	**EBI Bank0 Data Width 16-Bit** This bit defines if the data bus is 8-bit or 16-bit. 0 = EBI data width is 8-bit. 1 = EBI data width is 16-bit.
[28]	**CSEN**	**EBI Bank0 Enable Bit** This bit is the functional enable bit for EBI. 0 = EBI function Disabled. 1 = EBI function Enabled.
[27:24]	**R2R**	**Bank0 Idle State Cycle Between Read-Read** When read action is finish and next action is going to read, idle state is inserted and nCS[0] return to high if R2R is not zero. Idle state cycle = (R2R*MCLK). 0 = reserved.
[23:20]	**Reserved**	Reserved.
[19:16]	**R2W**	**Bank0 Idle State Cycle Between Read-Write** When read action is finish and next action is going to write, idle state is inserted and nCS[0] return to high if R2W is not zero. Idle state cycle = (R2W*MCLK). 0 = reserved.
[15:12]	**W2X**	**Bank0 Idle State Cycle After Write** When write action is finish, idle state is inserted and nCS[0] return to high if W2X is not zero.

		Idle state cycle = (W2X*MCLK). 0 = reserved.
[11]	**Reserved**	Reserved.
[10:8]	**TAHD**	**EBI Bank0 Data Access Hold Time** TAHD define data access hold time (tAHD). tAHD = (TAHD +1) * MCLK.
[7:3]	**TACC**	**EBI Bank0 Data Access Time** TACC define data access time (tACC). tACC = (TACC +1) * MCLK.
[2:0]	**TALE**	**Bank0 Expand Time Of ALE** The ALE width (tALE) to latch the address can be controlled by TALE. tALE = (TALE+1)*MCLK.

External Bus Interface Timing Control Register1 (EBI_TCTL1)

Register	Offset	R/W	Description	Reset Value
EBI_TCTL1	EBI_BA+0x08	R/W	External Bus Interface Bank1 Timing Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	SEPEN	DW16	CSEN	R2R			
23	22	21	20	19	18	17	16
Reserved				R2W			
15	14	13	12	11	10	9	8
W2X				Reversed	TAHD		
7	6	5	4	3	2	1	0
TACC					TALE		

Bits	Description	
[31]	**Reserved**	Reserved.
[30]	**SEPEN**	**EBI Bank1 Address/Data Bus Separating Enable Bit** 0 = Address/Data Bus Separating Disabled. 1 = Address/Data Bus Separating Enabled.
[29]	**DW16**	**EBI Bank1 Data Width 16-Bit** This bit defines if the data bus is 8-bit or 16-bit. 0 = EBI data width is 8-bit. 1 = EBI data width is 16-bit.
[28]	**CSEN**	**EBI Bank1 Enable Bit** This bit is the functional enable bit for EBI. 0 = EBI function Disabled. 1 = EBI function Enabled.
[27:24]	**R2R**	**Bank1 Idle State Cycle Between Read-Read** When read action is finish and next action is going to read, idle state is inserted and nCS[1] return to high if R2R is not zero. Idle state cycle = (R2R*MCLK). 0 = reserved.
[23:20]	**Reserved**	Reserved.
[19:16]	**R2W**	**Bank1 Idle State Cycle Between Read-Write** When read action is finish and next action is going to write, idle state is inserted and nCS[1] return to high if R2W is not zero. Idle state cycle = (R2W*MCLK). 0 = reserved.
[15:12]	**W2X**	**Bank1 Idle State Cycle After Write** When write action is finish, idle state is inserted and nCS[1] return to high if W2X is not zero.

		Idle state cycle = (W2X*MCLK). 0 =reserved.
[11]	**Reserved**	Reserved.
[10:8]	**TAHD**	**EBI Bank1 Data Access Hold Time** TAHD define data access hold time (tAHD). tAHD = (TAHD +1) * MCLK.
[7:3]	**TACC**	**EBI Bank1 Data Access Time** TACC define data access time (tACC). tACC = (TACC +1) * MCLK.
[2:0]	**TALE**	**Bank1 Expand Time Of ALE** The ALE width (tALE) to latch the address can be controlled by TALE. tALE = (TALE+1)*MCLK.

External Bus Interface Timing Control Register2 (EBI_TCTL2)

Register	Offset	R/W	Description	Reset Value
EBI_TCTL2	EBI_BA+0x0c	R/W	External Bus Interface Bank2 Timing Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	SEPEN	DW16	CSEN	R2R			
23	22	21	20	19	18	17	16
Reserved				R2W			
15	14	13	12	11	10	9	8
W2X				Reversed	TAHD		
7	6	5	4	3	2	1	0
TACC					TALE		

Bits	Description	
[31]	**Reserved**	Reserved.
[30]	**SEPEN**	**EBI Bank2 Address/Data Bus Separating Enable Bit** 0 = Address/Data Bus Separating Disabled. 1 = Address/Data Bus Separating Enabled.
[29]	**DW16**	**EBI Bank2 Data Width 16-Bit** This bit defines if the data bus is 8-bit or 16-bit. 0 = EBI data width is 8-bit. 1 = EBI data width is 16-bit.
[28]	**CSEN**	**EBI Bank2 Enable Bit** This bit is the functional enable bit for EBI. 0 = EBI function Disabled. 1 = EBI function Enabled.
[27:24]	**R2R**	**Bank2 Idle State Cycle Between Read-Read** When read action is finish and next action is going to read, idle state is inserted and nCS[2] return to high if R2R is not 0. Idle state cycle = (R2R*MCLK). 0 = Reserved.
[23:20]	**Reserved**	Reserved.
[19:16]	**R2W**	**Bank2 Idle State Cycle Between Read-Write** When read action is finish and next action is going to write, idle state is inserted and nCS[2] return to high if R2W is not 0. Idle state cycle = (R2W*MCLK). 0 = Reserved.
[15:12]	**W2X**	**Bank2 Idle State Cycle After Write** When write action is finish, idle state is inserted and nCS[2] return to high if W2X is not zero.

		Idle state cycle = (W2X*MCLK). 0 = reserved.
[11]	**Reserved**	Reserved.
[10:8]	**TAHD**	**EBI Bank2 Data Access Hold Time** TAHD define data access hold time (tAHD). tAHD = (TAHD +1) * MCLK.
[7:3]	**TACC**	**EBI Bank2 Data Access Time** TACC define data access time (tACC). tACC = (TACC +1) * MCLK.
[2:0]	**TALE**	**Bank2 Expand Time Of ALE** The ALE width (tALE) to latch the address can be controlled by TALE. tALE = (TALE+1)*MCLK.

External Bus Interface Timing Control Register3 (EBI_TCTL3)

Register	Offset	R/W	Description	Reset Value
EBI_TCTL3	EBI_BA+0x10	R/W	External Bus Interface Bank3 Timing Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	SEPEN	DW16	CSEN	R2R			
23	22	21	20	19	18	17	16
Reserved				R2W			
15	14	13	12	11	10	9	8
W2X				Reversed	TAHD		
7	6	5	4	3	2	1	0
TACC					TALE		

Bits	Description	
[31]	**Reserved**	Reserved.
[30]	**SEPEN**	**EBI Bank3 Address/Data Bus Separating Enable Bit** 0 = Address/Data Bus Separating Disabled. 1 = Address/Data Bus Separating Enabled.
[29]	**DW16**	**EBI Bank3 Data Width 16-Bit** This bit defines if the data bus is 8-bit or 16-bit. 0 = EBI data width is 8-bit. 1 = EBI data width is 16-bit.
[28]	**CSEN**	**EBI Bank3 Enable Bit** This bit is the functional enable bit for EBI. 0 = EBI function Disabled. 1 = EBI function Enabled.
[27:24]	**R2R**	**Bank3 Idle State Cycle Between Read-Read** When read action is finish and next action is going to read, idle state is inserted and nCS[3] return to high if R2R is not zero. Idle state cycle = (R2R*MCLK). 0 : reserved.
[23:20]	**Reserved**	Reserved.
[19:16]	**R2W**	**Bank3 Idle State Cycle Between Read-Write** When read action is finish and next action is going to write, idle state is inserted and nCS[3] return to high if R2W is not zero. Idle state cycle = (R2W*MCLK). 0 : reserved.
[15:12]	**W2X**	**Bank3 Idle State Cycle After Write** When write action is finish, idle state is inserted and nCS[3] return to high if W2X is not zero.

		Idle state cycle = (W2X*MCLK). 0 : reserved.
[11]	**Reserved**	Reserved.
[10:8]	**TAHD**	**EBI Bank3 Data Access Hold Time** TAHD define data access hold time (tAHD). tAHD = (TAHD +1) * MCLK.
[7:3]	**TACC**	**EBI Bank3 Data Access Time** TACC define data access time (tACC). tACC = (TACC +1) * MCLK.
[2:0]	**TALE**	**Bank3 Expand Time Of ALE** The ALE width (tALE) to latch the address can be controlled by TALE. tALE = (TALE+1)*MCLK.

External Bus Interface Crypto KEY Word0 (EBI_KEY0)

Register	Offset	R/W	Description	Reset Value
EBI_KEY0	EBI_BA+0x14	R/W	External Bus Interface Crypto Key Word 0	0x0000_0000

31	30	29	28	27	26	25	24
KEY[31:24]							
23	22	21	20	19	18	17	16
KEY[23:16]							
15	14	13	12	11	10	9	8
KEY[15:8]							
7	6	5	4	3	2	1	0
KEY[7:0]							

Bits	Description	
[31:0]	KEY	Crypto Key Word 0 (key[31:0]).

External Bus Interface Crypto KEY Word1 (EBI_KEY1)

Register	Offset	R/W	Description	Reset Value
EBI_KEY1	EBI_BA+0x18	R/W	External Bus Interface Crypto Key Word 1	0x0000_0000

31	30	29	28	27	26	25	24
KEY[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
KEY[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
KEY[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
KEY[7:0]							

Bits	Description	
[31:0]	**KEY**	Crypto Key Word 1 (key[63:32]).

External Bus Interface Crypto KEY Word2 (EBI_KEY2)

Register	Offset	R/W	Description	Reset Value
EBI_KEY2	EBI_BA+0x1c	R/W	External Bus Interface Crypto Key Word 2	0x0000_0000

31	30	29	28	27	26	25	24
KEY[31:24]							
23	22	21	20	19	18	17	16
KEY[23:16]							
15	14	13	12	11	10	9	8
KEY[15:8]							
7	6	5	4	3	2	1	0
KEY[7:0]							

Bits	Description	
[31:0]	KEY	Crypto Key Word 2 (key[95:64]).

External Bus Interface Crypto KEY Word3 (EBI_KEY3)

Register	Offset	R/W	Description	Reset Value
EBI_KEY3	EBI_BA+0x20	R/W	External Bus Interface Crypto Key Word 3	0x0000_0000

31	30	29	28	27	26	25	24
KEY[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
KEY[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
KEY[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
KEY[7:0]							

Bits	Description	
[31:0]	**KEY**	Crypto Key Word 3 (key[127:96]).

6.12 Ethernet MAC Controller (EMAC) (NUC472 Only)

6.12.1 Overview

This chip provides an Ethernet MAC Controller (EMAC) for Network application.

The Ethernet MAC controller consists of IEEE 802.3/Ethernet protocol engine with internal CAM function for recognizing Ethernet MAC addresses, Transmit-FIFO, Receive-FIFO, TX/RX state machine controller, time stamping engine for IEEE 1588, Magic Packet parsing engine and status controller.

The EMAC supports both the MII and RMII (Reduced MII) interface to connect with external Ethernet PHY.

6.12.2 Features

- Supports IEEE Std. 802.3 CSMA/CD protocol
- Supports Ethernet frame time stamping for IEEE Std. 1588 – 2002 protocol
- Supports both half and full duplex for 10 Mbps or 100 Mbps operation
- Supports both MII and RMII interface
- Supports MII Management function to control external Ethernet PHY
- Supports pause and remote pause function for flow control
- Supports long frame (more than 1518 bytes) and short frame (less than 64 bytes) reception
- Supports 16 entries CAM function for Ethernet MAC address recognition
- Supports Magic Packet recognition to wake system up from power-down mode
- Supports 256 bytes transmit FIFO and 256 bytes receive FIFO
- Supports DMA function

6.12.3 Block Diagram

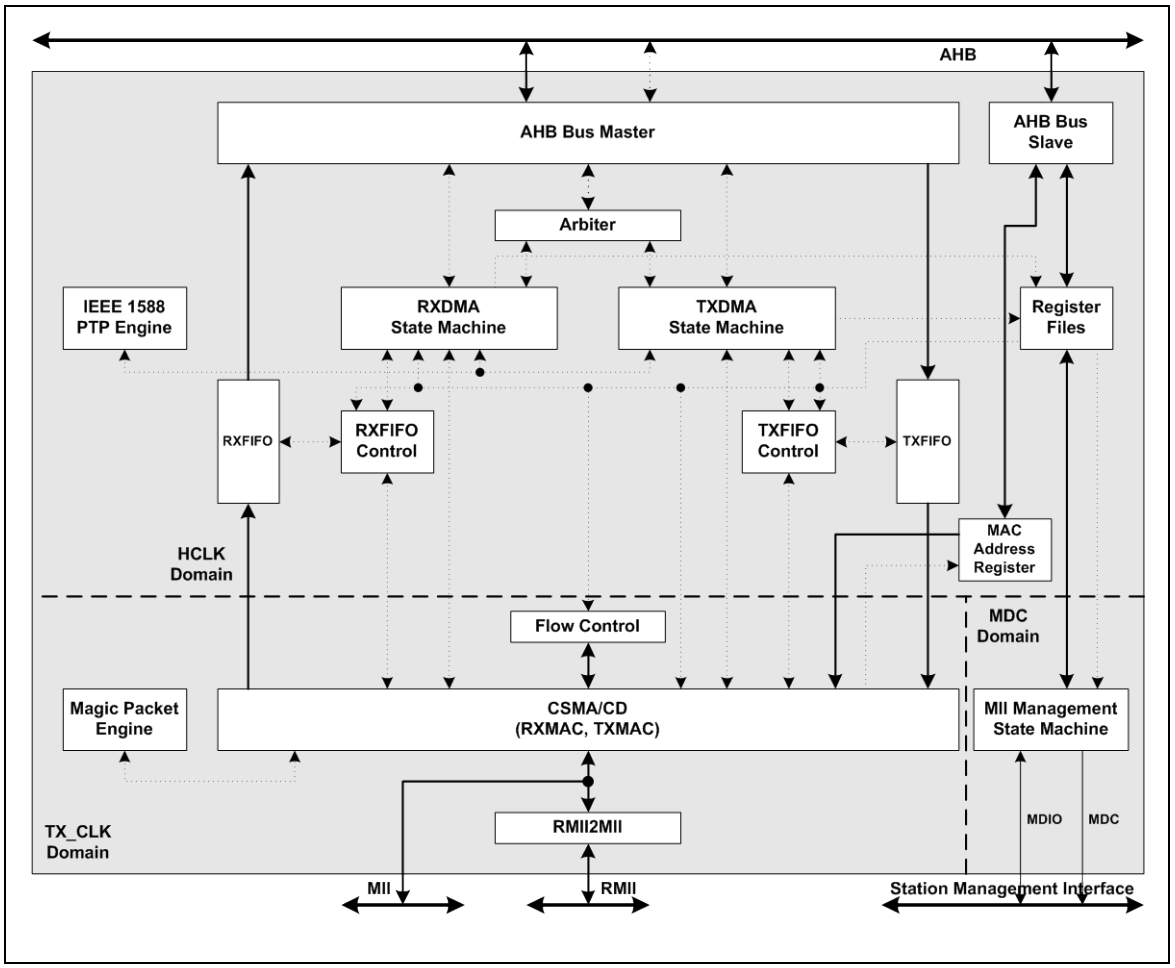


Figure 6.12-1 Ethernet MAC Controller Block Diagram

6.12.4 Functional Description

6.12.4.1 Arbiter

In the EMAC, there are two different bus requests, RXREQ and TXREQ respectively. Arbiter does the arbitration between the RXREQ and TXREQ, and then decides which one can request the AHB bus. The arbitration results are shown below:

RXREQ	TXREQ	Granted
0	0	Neither TXDMA nor RXDMA granted.
0	1	TXDMA granted.
1	0	RXDMA granted.
1	1	If TXFIFO valid data byte count is less than RXFIFO free space byte count, TXDMA granted.
1	1	If RXFIFO free space byte count is less than or equal to TXFIFO valid space byte count, RXDAM granted.

Table 6.12-1 Arbiter Arbitration Results

6.12.4.2 TXDMA State Machine

The TXDMA state machine transfers data from the system memory to the internal 256 bytes transmit FIFO through the AHB master. Then, the TXDMA state machine will request the transmit MAC to send the data out. During the transmission process, the TXDMA will fetch the transmit descriptor first. Through the buffer address field of the transmit descriptor, the TXDMA fetch the frame data from the system and store it into the internal 256 bytes transmit FIFO. Then, the transmit MAC will read frame data from the transmit FIFO and send the frame out. After the finish of the frame transmission, the TXDMA updates the transmit status of current frame and write the transmit descriptor back to the system memory to indicate the frame transmission has finished.

6.12.4.3 RXDMA State Machine

The RXDMA state machine transfers data from the internal 256 bytes receiving FIFO to the system memory through AHB master. During the receiving process, the RXDMA will fetch the received descriptor first. Through the buffer address field of the received descriptor, the RXDMA will know memory space which is allocated to store the incoming frame. After the received MAC indicates there is a new incoming frame, the RXDMA starts to transfer the frame data from the internal received FIFO to the system memory. After the receiving process has finished, the RXDMA will update the receiving status of current frame and write the received descriptor back to system memory to indicate a new incoming frame is in the system memory.

6.12.4.4 Flow Control

This block implements the flow control function while EMAC operates in the full duplex mode. The flow control function is defined in the IEEE 802.3 Std. chapter 31. The type of flow control frame defined in the IEEE 802.3 Std. is only the PAUSE frame at the moment. The control frame transmission and reception is programmable through the control registers.

To receive a control frame, software must set the bit ACP (Accept Control Packet) of register EMAC_CTL (MAC Command Register). While a PAUSE frame is received, the flow control

function will pause the transmission process after the current transmitting frame has been transmitted out.

To transmit a control frame out, software must program the destination MAC address of control frame into the register pair {EMAC_CAM13M, EMAC_CAM13L}, source MAC address into the register pair {EMAC_CAM14M, EMAC_CAM14L}, and configure LENGTH, OPCODE and OPERAND of control frame into the register pair {EMAC_CAM15MSB, EMAC_CAM15LSB}, and then set the bit SDPZ (EMAC_CTL[16]). The bit SDPZ (EMAC_CTL[16]) will be cleared while the control frame has been transmitted out.

6.12.4.5 MII Management State Machine

The MII management function of EMAC is compliant to IEEE 802.3 Std. Through the MII management interface, software can access the control and status registers of the external PHY chip. Two programmable registers EMAC_MIIMDAT (MAC MII Management Data Register) and EMAC_MIIMCTL (MAC MII Management Data Control and Address Register) are for MII management function. Set the bit BUSY (EMAC_MIIMCTL[17]) will trigger the MII management state machine. After the MII management cycle is finished, the BUSY bit will be cleared automatically.

6.12.4.6 Media Access Control (MAC)

The function of Ethernet MAC fully meets the requirements defined by the IEEE802.3u specification. The following paragraphs describe the frame structure and the operation of the transmission and receiving.

The transmission data frame sent from the transmit DMA will be encapsulated by the MAC before transmitting onto the MII bus. The sent data will be assembled with the preamble, the start frame delimiter (SFD), the frame check sequence and the padding for enforcing those less than 64 bytes to meet the minimum size frame and CRC sequence. The outgoing frame format will be as follows.

110101010 --- 10101010	10101011	d0	d1	d2	--	dn	Padding	CRC31	CRC30	---	CRC0

Figure 6.12-2 Ethernet Frame Format

As mentioned by the above format, the preamble is a consecutive 7-byte long with the pattern “10101010” and the SFD is a one byte 10101011 data. The padding data will be all 0 value if the sent data frame is less than 64 bytes. The padding disable function specified in the bit P of the transmit descriptor is used to control if the MAC needs to pad data at the end of frame data or not when the transmitted data frame is less than 64 bytes. The padding data will not be appended if the padding disable bit is set to be high. The bits CRC0 ... CRC31 are the 32 bits cyclic redundancy check (CRC) sequence. The CRC encoding is defined by the following polynomial specified by the IEEE802.3. This 32 bits CRC appending function will be disabled if the Inhibit CRC of the transmission descriptor is set to high.

The MAC also performs many other transmission functions specified by the IEEE802.3, including the inter-frame spacing function, collision detection, collision enforcement, collision back off and retransmission. The collision back-off timer is a function of the integer slot time, 512 bit times. The number of slot times to delay between the current transmission attempt to the next attempt is determined by a uniformly distributed random integer algorithm specified by the IEEE802.3. The MAC performs the receive functions specified by the IEEE 802.3 including the address recognition function, the frame check sequence validation, the frame disassembly, framing and

collision filtering.

6.12.4.7 Time Stamping Engine for IEEE 1588

The EMAC supports a time stamping engine for IEEE Std. 1588. In this time stamping engine, a 64-bit counter implemented to generate the reference timing, the registers EMAC_TSSEC and ETSLSR.

In frame transmission, if TSEN (EMAC_TSCTL[0]) and TTSEN of TXDES 0 (TXDMA Descriptor Word 0) are both high, EMAC would store the 64-bit reference timing value to TXDES 1 (TXDMA Descriptor Word 1) and TXDES 2 (TXDMA Descriptor Word 2) when frame transmission completed.

In frame reception, if TSEN (EMAC_TSCTL[0]) is high, EMAC would store the 64-bit reference timing value to RXDES 1 (RXDMA Descriptor Word 1) and RXDES 3 (RXDMA Descriptor Word 3) when the frame reception finished.

The figure shown below describes how the 64-bit counter works to generate the reference timing.

The 64-bit counter formed by two 32-bit counters, the EMAC_TSSEC and EMAC_TSSUBSEC, a updated using the EMAC's input reference clock, the HCLK. Two difference methods, controlled by TSMODE (EMAC_TSCTL[3]), implemented to increase 32-bit EMAC_TSSUBSEC counter by value configured in register EMAC_TSINC. When TSMODE (EMAC_TSCTL[3]) is low, TSLSR counter increased in every clock. When TSMODE (EMAC_TSCTL[3]) is high, TSLSR counter increased only when accumulator is overflow.

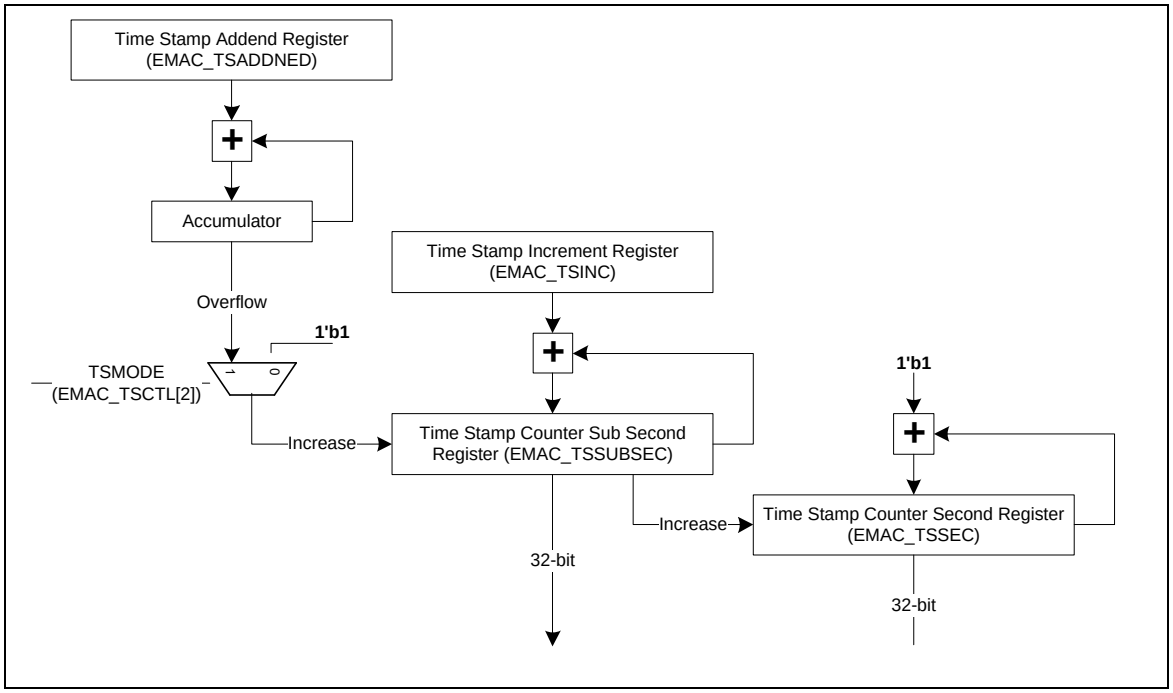


Figure 6.12-3 64-bit Reference Timing Counter

6.12.4.8 Magic Packet Parsing Engine

The EMAC supports a Magic packet parsing engine for recognizing Magic packet. The Magic packet is a broadcast frame which payload includes 6 bytes of 0xFF, followed by 16 repetitions of

48-bit MAC address defined by registers EMAC_CAM0M and EMAC_CAM0L.

The WOLEN (EMAC_CTL[6]) controls if the Magic packet parsing engine enabled. If WOLEN (EMAC_CTL[6]) is high, EMAC will set bit WOLIF (EMAC_INTSTS[15]) high to indicate Magic packet received. At the same time, EMAC generates an event to wake system up from power-down mode. If WOLIEN of register MIEN is high, EMAC generates an RX interrupt to CPU simultaneously.

6.12.5 DMA Descriptors Data Structure

A link-list data structure named as descriptor is used to keep the control, status and data information of each frame. Through the descriptor, CPU and EMAC exchange the information for frame reception and transmission.

Two different descriptors defined in EMAC. One named as RXDMA descriptor for frame reception and the other named as TXDMA descriptor for frame transmission. Each RXDMA or TXDMA descriptor consists of four words. The descriptor keeps the much control, status information and the details of descriptor are described below.

6.12.5.1 RXDMA Descriptor Data Structure

The RXDMA descriptor consists of four 32-bit words. The data structure of RXDMA descriptor shown in figure below.

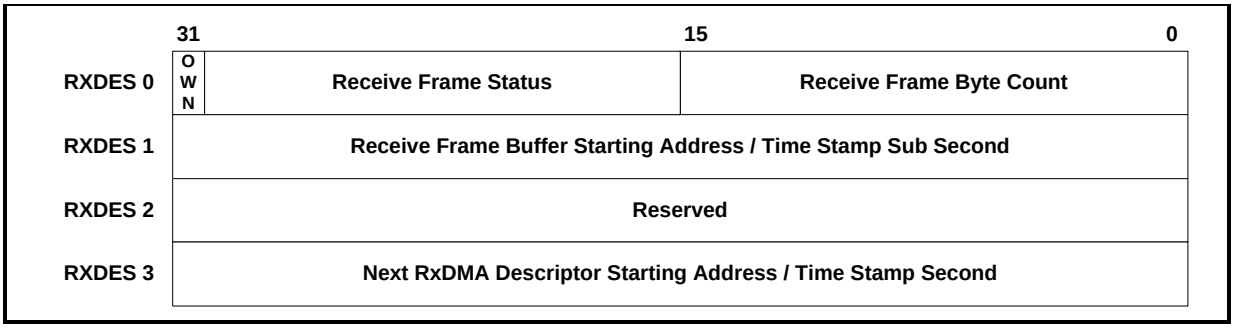


Figure 6.12-4 RXDMA Descriptor Data Structure

RXDES 0: RXDMA Descriptor Word 0

The RXDMA descriptor word 0 contains a descriptor ownership indicator, receive frame status, and receive frame byte count. The detail description of RXDES 0 is shown below.

31	30	29	28	27	26	25	24
Owner	Reserved						
23	22	21	20	19	18	17	16
RTSAS	RPIF	ALIEIF	RXGDIF	LPIF	Reserved	CRCEIF	RXIF
15	14	13	12	11	10	9	8
RBC							
7	6	5	4	3	2	1	0
RBC							

Bits	Field	Description
[31]	**Owner**	**Ownership** The ownership field defines which one, the CPU or EMAC, is the owner of each RX descriptor. Only the owner has right to modify the RX descriptor and the others can read the RX descriptor only. If the O=1'b1 indicates the EMAC RXDMA is the owner of RX descriptor and the RX descriptor is available for frame reception. After the frame reception completed, EMAC RXDMA modified ownership field to 1'b0. If the O=1'b0 indicates the CPU is the owner of RX descriptor. After the CPU completed the frame processing, it modified the ownership field to 1'b1 and released the RX descriptor to EMAC RXDMA. 0 = The owner is CPU. 1 = The owner is EMAC.
[30:24]	**Reserved**	Reserved.
[23]	**RTSAS**	**RX Time Stamp Active Status** This bit is to indicate the time stamping circuit stamped this incoming frame successfully. When this bit set high, RX Descriptor Word 1 and RX Descriptor Word 3 keep the time stamp value recorded when this incoming frame is received completely. 0 = RX Descriptor Word 1 and RX Descriptor Word 3 does not keep the time stamp value. 1 = RX Descriptor Word 1 and RX Descriptor Word 3 keep the time stamp value.
[22]	**RPIF**	**Runt Packet** The RPIF indicates the frame stored in the data buffer pointed by RX descriptor is a short frame (frame length is less than 64 bytes). 0 = The frame is not a short frame. 1 = The frame is a short frame.
[21]	**ALIEIF**	**Alignment Error** The ALIEIF indicates the frame stored in the data buffer pointed by RX descriptor is not a multiple of byte. 0 = The frame is a multiple of byte. 1 = The frame is not a multiple of byte.

[20]	**RXGDIF**	**Frame Reception Complete** The RXGDIF indicates the frame reception has completed and stored in the data buffer pointed by RX descriptor. 0 = The frame reception does not complete yet. 1 = The frame reception completed.
[19]	**LPIF**	**Long Packet Interrupt Flag** The LPIF indicates the frame stored in the data buffer pointed by RX descriptor is a long frame (frame length is greater than 1518 bytes). 0 = The frame is not a long frame. 1 = The frame is a long frame.
[18]	**Reserved**	Reserved.
[17]	**CRCEIF**	**CRC Error** The CRCEIF indicates the frame stored in the data buffer pointed by RX descriptor incurred CRC error. 0 = The frame does not incur CRC error. 1 = The frame incurred CRC error.
[16]	**RXIF**	**Receive Interrupt** The RXIF indicates the frame stored in the data buffer pointed by RX descriptor caused an interrupt condition. 0 = The frame does not cause an interrupt. 1 = The frame caused an interrupt.
[15:0]	**RBC**	**Receive Byte Count** The RBC indicates the byte count of the frame stored in the data buffer pointed by RX descriptor. The four bytes CRC field is also included in the receive byte count. But if the STRIPCRC (EMAC_CTL[5]) is enabled, the four bytes CRC field will be excluded from the receive byte count.

RXDES 1: RXDMA Descriptor Word 1

The RXDMA descriptor word 1 contains the received frame buffer starting address or time stamp least significant 32-bit value. The detail description of RXDES 1 is shown below.

31	30	29	28	27	26	25	24
RXBSA/TSSUBSEC							
23	22	21	20	19	18	17	16
RXBSA/TSSUBSEC							
15	14	13	12	11	10	9	8
RXBSA/TSSUBSEC							
7	6	5	4	3	2	1	0
RXBSA/TSSUBSEC							

Bits	Field	Description
[31:0]	**RXBSA**	**Receive Buffer Starting Address** The RXBSA is the buffer starting address to store the received packet.

Bits	Field	Description
[31:0]	**TSSUBSEC**	**Time Stamp Sub-Second** If TSEN (EMAC_TSCTL[0]) enabled, Ethernet MAC controller would store time stamp least signification 32-bit value, register ETSLSR, into this field when it writes back RX Descriptor to system memory.

RXDES 2: RXDMA Descriptor Word 2

The RXDMA descriptor word 2 currently is reserved.

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							

Bits	Field	Description
[31:0]	**Reserved**	Reserved.

RXDES 3: RXDMA Descriptor Word 3

The RXDMA descriptor word 3 contains the next RXDMA descriptor starting address or time stamp most significant 32-bit value. The detail description of RXDES 3 is shown below.

31	30	29	28	27	26	25	24
NRXDSA/TSSEC							
23	22	21	20	19	18	17	16
NRXDSA/TSSEC							
15	14	13	12	11	10	9	8
NRXDSA/TSSEC							
7	6	5	4	3	2	1	0
NRXDSA/TSSEC							

Bits	Field	Description
[31:0]	**NRXDSA**	**Next RX Descriptor Starting Address** NRXDSA is the starting address of the next RX descriptor. When Ethernet MAC controller fetches the next RX descriptor, it ignored the bits [1:0] of NRXDSA.

Bits	Field	Description
[31:0]	**TSSEC**	**Time Stamp Second** If TSEN (EMAC_TSCTL[0]) enabled, Ethernet MAC controller would store time stamp most signification 32-bit value, register ETSMSR, into this field when it writes back RX Descriptor to system memory.

6.12.5.2 TXDMA Descriptor Data Structure

The TXDMA descriptor consists of four 32-bit words. The data structure of TXDMA descriptor shown in figure below.

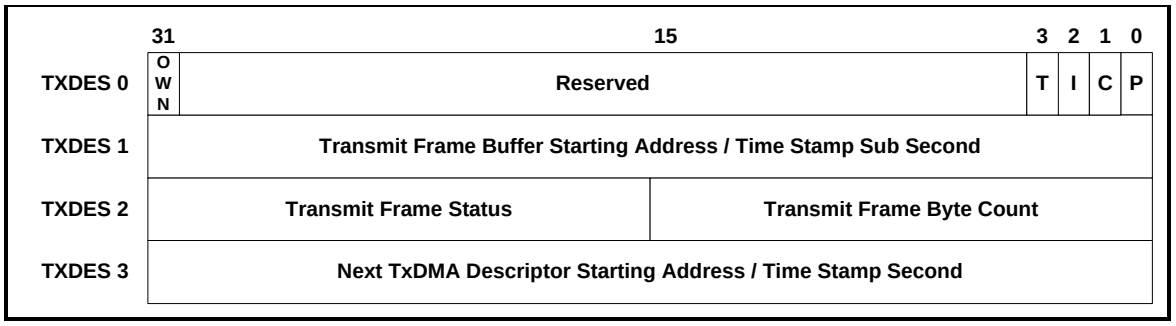


Figure 6.12-5 TXDMA Descriptor Data Structure

TXDES 0: TXDMA Descriptor Word 0

The TXDMA descriptor word 0 contains a descriptor ownership indicator. In addition, it also contains control bits for transmit frame padding, CRC append, interrupt enable and time stamping control. The detail description of TXDES 0 is shown below.

31	30	29	28	27	26	25	24
Owner	Reserved						
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				TTSEN	INTEN	CRCAPP	PADEN

Bits	Field	Description
[31]	**Owner**	**Ownership** The ownership field defines which one, the CPU or EMAC, is the owner of each TX descriptor. Only the owner has right to modify the TX descriptor and the other can read the TX descriptor only. If the O=1'b1 indicates the EMAC TXDMA is the owner of TX descriptor and the TX descriptor is available for frame transmission. After the frame transmission completed, EMAC TXDMA modify ownership field to 1'b0 and return the ownership of TX descriptor to CPU. If the O=1'b0 indicates the CPU is the owner of TX descriptor. After the CPU prepares new frame to wait transmission, it modifies the ownership field to 1'b1 and releases the TX descriptor to EMAC TXDMA. 0 = The owner is CPU. 1 = The owner is EMAC.
[30:4]	**Reserved**	Reserved.
[3]	**TTSEN**	**TX Time Stamp Enable Bit** When this bit set high and IEEE 1588 PTP function is also enabled, the embedded time stamping circuit would stamp this frame when SFD of frame is transmitted out on MII/RMII. 0 = IEEE 1588 time stamp function Disabled for this frame. 1 = IEEE 1588 time stamp function Enabled for this frame.
[2]	**INTEN**	**Transmit Interrupt Enable Bit** The INTEN controls the interrupt trigger circuit after the frame transmission completed. If the INTEN enabled, the EMAC will trigger interrupt after frame transmission completed. Otherwise, the interrupt doesn't be triggered. 0 = Frame transmission interrupt masked. 1 = Frame transmission interrupt Enabled.

[1]	**CRCAPP**	**CRC Append** The CRCAPP control the CRC append during frame transmission. If CRCAPP is enabled, the 4-bytes CRC checksum will be appended to frame at the end of frame transmission. 0 = 4-bytes CRC appending Disabled. 1 = 4-bytes CRC appending Enabled.
[0]	**PADEN**	**Padding Enable Bit** The PADEN control the PAD bits appending while the length of transmission frame is less than 60 bytes. If PADEN is enabled, EMAC does the padding automatically. 0 = PAD bits appending Disabled. 1 = PAD bits appending Enabled.

TXDES 1: TXDMA Descriptor Word 1

The TXDMA descriptor word 1 contains the transmit frame buffer starting address or time stamp least significant 32-bit value. The detail description of TXDES 1 is shown below.

31	30	29	28	27	26	25	24
TXBSA/TSSUBSEC							
23	22	21	20	19	18	17	16
TXBSA/TSSUBSEC							
15	14	13	12	11	10	9	8
TXBSA/TSSUBSEC							
7	6	5	4	3	2	1	0
TXBSA/TSSUBSEC							

Bits	Field	Description
[31:2]	**TXBSA**	**Transmit Buffer Starting Address** The TXBSA is the starting address of buffer where transmit packet data stored.

Bits	Field	Description
[31:0]	**TSSUBSEC**	**Time Stamp Sub-Second** If TSEN (EMAC_TSCTL[0]) and TTSEN of TX Descriptor word 0 both enabled, Ethernet MAC controller would store time stamp least signification 32-bit value, register EMAC_TSSUBSEC, into this field when it writes back TX Descriptor to system memory.

TXDES 2: TXDMA Descriptor Word 2

The TXDMA descriptor word 2 contains transmit frame status, and transmit frame byte count. The detail description of TXDES 2 is shown below.

31	30	29	28	27	26	25	24
COLCNT				TTSAS	SQE	TXPAUSED	TXHALT
23	22	21	20	19	18	17	16
LCIF	TXABTIF	NCSIF	EXDEFIF	TXCPIF	Reserved	DEF	TXIF
15	14	13	12	11	10	9	8
TBC							
7	6	5	4	3	2	1	0
TBC							

Bits	Field	Description
[31:28]	**COLCNT**	**Collision Count** The COLCNT indicates the how many collisions found consecutively during a packet transmission. If the packet incurred 16 consecutive collisions during transmission, the COLCNT is 0x0 and bit TXABTIF is set high.
[27]	**TTSAS**	**TX Time Stamp Active Status** This bit is to indicate the time stamping circuit stamped this frame successfully. When this bit set high, TX Descriptor Word 1 and TX Descriptor Word 3 keep the time stamp value recorded when SFD of frame is transmitted out on MII/RMII. 0 = TX Descriptor Word 1 and TX Descriptor Word 3 does not keep the time stamp value. 1 = TX Descriptor Word 1 and TX Descriptor Word 3 keep the time stamp value.
[26]	**SQE**	**SQE Error** The SQE indicates the SQE error found at end of packet transmission on 10Mbps half-duplex mode. The SQE error check will only be done while both bit SQECHKEN (EMAC_CTL[17]) is enabled and EMAC is operating on 10Mbps half-duplex mode. 0 = No SQE error found at end of packet transmission. 1 = SQE error found at end of packet transmission.
[25]	**TXPAUSED**	**Transmission Paused** The TXPAUSED indicates the next normal packet transmission process will be paused temporally because EMAC received a PAUSE control frame, or software sets the bit SDPZ (EMAC_MCM[16]) and enables EMAC to transmit a PAUSE control frame out. 0 = Next normal packet transmission process continue normally. 1 = Next normal packet transmission process paused.
[24]	**TXHALT**	**Transmission Halted** The TXHALT indicates the next normal packet transmission process will be halted because the bit TXON (EMAC_CTL[8]) is disabled by software. 0 = Next normal packet transmission process continue normally. 1 = Next normal packet transmission process halted.

[23]	**LCIF**	**Late Collision** The LCIF indicates the collision found in the outside of 64 bytes collision window. This means after the 64 bytes of a frame has been transmitted out to the network, the collision still found. The late collision check will only be done while EMAC is operating on half-duplex mode. 0 = No collision found in the outside of 64 bytes collision window. 1 = Collision found in the outside of 64 bytes collision window.
[22]	**TXABTIF**	**Transmission Abort** The TXABTIF indicates the packet incurred 16 consecutive collisions during transmission, and then the transmission process for this packet is aborted. The transmission abort is only available while EMAC is operating on half-duplex mode. 0 = Packet does not incur 16 consecutive collisions during transmission. 1 = Packet incurred 16 consecutive collisions during transmission.
[21]	**NCSIF**	**No Carrier Sense** The NCSIF indicates the MII I/F signal CRS does not active at the start of or during the packet transmission. The NCSIF is only available while EMAC is operating on half-duplex mode. 0 = CRS signal does not active at the start of or during the packet transmission. 1 = CRS signal actives correctly.
[20]	**EXDEFIF**	**Defer Exceed** The EXDEFIF indicates the frame waiting for transmission has deferred over 0.32768ms on 100Mbps mode, or 3.2768ms on 10Mbps mode. The deferral exceed check will only be done while bit NODEF (EMAC_CTL[9]) is disabled, and EMAC is operating on half-duplex mode. 0 = Frame waiting for transmission did not defer over 0.32768ms (100Mbps) or 3.2768ms (10Mbps). 1 = Frame waiting for transmission deferred over 0.32768ms (100Mbps) or 3.2768ms (10Mbps).
[19]	**TXCPIF**	**Transmission Complete** The TXCPIF indicates the packet transmission has completed correctly. 0 = The packet transmission does not complete. 1 = The packet transmission completed.
[18]	**Reserved**	Reserved.
[17]	**DEF**	**Transmission Deferred** The DEF indicates the packet transmission has deferred once. The DEF is only available while EMAC is operating on half-duplex mode. 0 = Packet transmission does not defer. 1 = Packet transmission deferred once.
[16]	**TXIF**	**Transmit Interrupt** The TXIF indicates the packet transmission would trigger an interrupt condition. 0 = The packet transmission would not trigger an interrupt. 1 = The packet transmission would trigger an interrupt.
[15:0]	**TBC**	**Transmit Byte Count** The TBC indicates the byte count of the frame stored in the data buffer pointed by TX descriptor for transmission.

TXDES 3: TXDMA Descriptor Word 3

The TXDMA descriptor word 3 contains the next TXDMA descriptor starting address or time stamp most significant 32-bit value. The detail description of TXDES 3 is shown below.

31	30	29	28	27	26	25	24
NTXDSA/TSSEC							
23	22	21	20	19	18	17	16
NTXDSA/TSSEC							
15	14	13	12	11	10	9	8
NTXDSA/TSSEC							
7	6	5	4	3	2	1	0
NTXDSA/TSSEC							

Bits	Field	Description
[31:0]	**NTXDSA**	**Next TX Descriptor Starting Address** NTXDSA is the starting address of the next TX descriptor. When Ethernet MAC controller fetches the next TX descriptor, it ignored the bits [1:0] of NTXDSA.

Bits	Field	Description
[31:0]	**TSSEC**	**Time Stamp Second** If TSEN (EMAC_TSCTL[0]) and TTSEN of TX Descriptor word 0 are both enabled, Ethernet MAC controller would store time stamp most signification 32-bit value, register EMAC_TSSEC, into this field when it writes back TX Descriptor to system memory.

6.12.6 Register and Memory Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
EMAC Base Address: **EMAC_BA = 0x4000_B000**				
EMAC_CAMCTL	EMAC_BA+0x000	R/W	CAM Comparison Control Register	0x0000_0000
ECAM_CAMEN	EMAC_BA+0x004	R/W	CAM Enable Register	0x0000_0000
EMAC_CAM0M	EMAC_BA+0x008	R/W	CAM0 Most Significant Word Register	0x0000_0000
EMAC_CAM0L	EMAC_BA+0x00C	R/W	CAM0 Least Significant Word Register	0x0000_0000
EMAC_CAM1M	EMAC_BA+0x010	R/W	CAM1 Most Significant Word Register	0x0000_0000
EMAC_CAM1L	EMAC_BA+0x014	R/W	CAM1 Least Significant Word Register	0x0000_0000
EMAC_CAM2M	EMAC_BA+0x018	R/W	CAM2 Most Significant Word Register	0x0000_0000
EMAC_CAM2L	EMAC_BA+0x01C	R/W	CAM2 Least Significant Word Register	0x0000_0000
EMAC_CAM3M	EMAC_BA+0x020	R/W	CAM3 Most Significant Word Register	0x0000_0000
EMAC_CAM3L	EMAC_BA+0x024	R/W	CAM3 Least Significant Word Register	0x0000_0000
EMAC_CAM4M	EMAC_BA+0x028	R/W	CAM4 Most Significant Word Register	0x0000_0000
EMAC_CAM4L	EMAC_BA+0x02C	R/W	CAM4 Least Significant Word Register	0x0000_0000
EMAC_CAM5M	EMAC_BA+0x030	R/W	CAM5 Most Significant Word Register	0x0000_0000
EMAC_CAM5L	EMAC_BA+0x034	R/W	CAM5 Least Significant Word Register	0x0000_0000
EMAC_CAM6M	EMAC_BA+0x038	R/W	CAM6 Most Significant Word Register	0x0000_0000
EMAC_CAM6L	EMAC_BA+0x03C	R/W	CAM6 Least Significant Word Register	0x0000_0000
EMAC_CAM7M	EMAC_BA+0x040	R/W	CAM7 Most Significant Word Register	0x0000_0000
EMAC_CAM7L	EMAC_BA+0x044	R/W	CAM7 Least Significant Word Register	0x0000_0000
EMAC_CAM8M	EMAC_BA+0x048	R/W	CAM8 Most Significant Word Register	0x0000_0000
EMAC_CAM8L	EMAC_BA+0x04C	R/W	CAM8 Least Significant Word Register	0x0000_0000
EMAC_CAM9M	EMAC_BA+0x050	R/W	CAM9 Most Significant Word Register	0x0000_0000
EMAC_CAM9L	EMAC_BA+0x054	R/W	CAM9 Least Significant Word Register	0x0000_0000
EMAC_CAM10M	EMAC_BA+0x058	R/W	CAM10 Most Significant Word Register	0x0000_0000
EMAC_CAM10L	EMAC_BA+0x05C	R/W	CAM10 Least Significant Word Register	0x0000_0000
EMAC_CAM11M	EMAC_BA+0x060	R/W	CAM11 Most Significant Word Register	0x0000_0000
EMAC_CAM11L	EMAC_BA+0x064	R/W	CAM11 Least Significant Word Register	0x0000_0000

EMAC_CAM12M	EMAC_BA+0x068	R/W	CAM12 Most Significant Word Register	0x0000_0000
EMAC_CAM12L	EMAC_BA+0x06C	R/W	CAM12 Least Significant Word Register	0x0000_0000
EMAC_CAM13M	EMAC_BA+0x070	R/W	CAM13 Most Significant Word Register	0x0000_0000
EMAC_CAM13L	EMAC_BA+0x074	R/W	CAM13 Least Significant Word Register	0x0000_0000
EMAC_CAM14M	EMAC_BA+0x078	R/W	CAM14 Most Significant Word Register	0x0000_0000
EMAC_CAM14L	EMAC_BA+0x07C	R/W	CAM14 Least Significant Word Register	0x0000_0000
EMAC_CAM15MSB	EMAC_BA+0x080	R/W	CAM15 Most Significant Word Register	0x0000_0000
EMAC_CAM15LSB	EMAC_BA+0x084	R/W	CAM15 Least Significant Word Register	0x0000_0000
EMAC_TXDSA	EMAC_BA+0x088	R/W	Transmit Descriptor Link List Start Address Register	0xFFFF_FFFC
EMAC_RXDSA	EMAC_BA+0x08C	R/W	Receive Descriptor Link List Start Address Register	0xFFFF_FFFC
EMAC_CTL	EMAC_BA+0x090	R/W	MAC Control Register	0x0040_0000
EMAC_MIIMDAT	EMAC_BA+0x094	R/W	MII Management Data Register	0x0000_0000
EMAC_MIIMCTL	EMAC_BA+0x098	R/W	MII Management Control and Address Register	0x0090_0000
EMAC_FIFOCTL	EMAC_BA+0x09C	R/W	FIFO Threshold Control Register	0x0000_0101
EMAC_TXST	EMAC_BA+0x0A0	W	Transmit Start Demand Register	Undefined
EMAC_RXST	EMAC_BA+0x0A4	W	Receive Start Demand Register	Undefined
EMAC_MRFL	EMAC_BA+0x0A8	R/W	Maximum Receive Frame Control Register	0x0000_0800
EMAC_INTEN	EMAC_BA+0x0AC	R/W	MAC Interrupt Enable Register	0x0000_0000
EMAC_INTSTS	EMAC_BA+0x0B0	R/W	MAC Interrupt Status Register	0x0000_0000
EMAC_GENSTS	EMAC_BA+0x0B4	R/W	MAC General Status Register	0x0000_0000
EMAC_MPCNT	EMAC_BA+0x0B8	R/W	Missed Packet Count Register	0x0000_7FFF
EMAC_RPCNT	EMAC_BA+0x0BC	R	MAC Receive Pause Count Register	0x0000_0000
EMAC_FRSTS	EMAC_BA+0x0C8	R/W	DMA Receive Frame Status Register	0x0000_0000
EMAC_CTXDSA	EMAC_BA+0x0CC	R	Current Transmit Descriptor Start Address Register	0x0000_0000
EMAC_CTXBSA	EMAC_BA+0x0D0	R	Current Transmit Buffer Start Address Register	0x0000_0000
EMAC_CRXDSA	EMAC_BA+0x0D4	R	Current Receive Descriptor Start Address Register	0x0000_0000
EMAC_CRXBSA	EMAC_BA+0x0D8	R	Current Receive Buffer Start Address Register	0x0000_0000
EMAC_TSCTL	EMAC_BA+0x100	R/W	Time Stamp Control Register	0x0000_0000
EMAC_TSSEC	EMAC_BA+0x110	R	Time Stamp Counter Second Register	0x0000_0000

EMAC_TSSUBSEC	EMAC_BA+0x114	R	Time Stamp Counter Sub Second Register	0x0000_0000
EMAC_TSINC	EMAC_BA+0x118	R/W	Time Stamp Increment Register	0x0000_0000
EMAC_TSADDEND	EMAC_BA+0x11C	R/W	Time Stamp Addend Register	0x0000_0000
EMAC_UPDSEC	EMAC_BA+0x120	R/W	Time Stamp Update Second Register	0x0000_0000
EMAC_UPDSUBSEC	EMAC_BA+0x124	R/W	Time Stamp Update Sub Second Register	0x0000_0000
EMAC_ALMSEC	EMAC_BA+0x128	R/W	Time Stamp Alarm Second Register	0x0000_0000
EMAC_ALMSUBSEC	EMAC_BA+0x12C	R/W	Time Stamp Alarm Sub Second Register	0x0000_0000

6.12.7 Register Description

CAM Command Register (EMAC_CAMCTL)

The EMAC supports CAM function for destination MAC address recognition. The EMAC_CAMCTL control the CAM comparison function, and unicast, multicast, and broadcast packet reception.

Register	Offset	R/W	Description	Reset Value
EMAC_CAMCTL	EMAC_BA+0x000	R/W	CAM Comparison Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			CMPEN	COMPEN	ABP	AMP	AUP

Bits	Description	
[31:5]	**Reserved**	Reserved.
[4]	**CMPEN**	**CAM Compare Enable** The CMPEN controls the enable of CAM comparison function for destination MAC address recognition. If software wants to receive a packet with specific destination MAC address, configures the MAC address into CAM 12~0, then enables that CAM entry and set CMPEN to 1. 0 = CAM comparison function for destination MAC address recognition disabled. 1 = CAM comparison function for destination MAC address recognition enabled.
[3]	**COMPEN**	**Complement CAM Comparison Enable** The COMPEN controls the complement of the CAM comparison result. If the CMPEN and COMPEN are both enabled, the incoming packet with specific destination MAC address configured in CAM entry will be dropped. And the incoming packet with destination MAC address does not configured in any CAM entry will be received. 0 = The CAM comparison result does not complement. 1 = The CAM comparison result complemented.
[2]	**ABP**	**Accept Broadcast Packet** The ABP controls the broadcast packet reception. If ABP is enabled, EMAC receives all incoming packet its destination MAC address is a broadcast address. 0 = EMAC receives packet depends on the CAM comparison result. 1 = EMAC receives all broadcast packets.

[1]	**AMP**	**Accept Multicast Packet** The AMP controls the multicast packet reception. If AMP is enabled, EMAC receives all incoming packet its destination MAC address is a multicast address. 0 = EMAC receives packet depends on the CAM comparison result. 1 = EMAC receives all multicast packets.
[0]	**AUP**	**Accept Unicast Packet** The AUP controls the unicast packet reception. If AUP is enabled, EMAC receives all incoming packet its destination MAC address is a unicast address. 0 = EMAC receives packet depends on the CAM comparison result. 1 = EMAC receives all unicast packets.

CAMCMR Setting and Comparison Results

The following table is the address recognition result in different CAMCMR configuration. The column Result shows the incoming packet type that can pass the address recognition in specific CAM configuration. The C, U, M and B represents the:

C: Indicates the destination MAC address of incoming packet has been configured in CAM entry.

U: Indicates the incoming packet is a unicast packet.

M: Indicates the incoming packet is a multicast packet.

B: Indicates the incoming packet is a broadcast packet.

CMPEN	CCAM	AUP	AMP	ABP	Result			
0	0	0	0	0	No Packet			
0	0	0	0	1	B			
0	0	0	1	0	M			
0	0	0	1	1	M	B		
0	0	1	0	0	C	U		
0	0	1	0	1	C	U	B	
0	0	1	1	0	C	U	M	
0	0	1	1	1	C	U	M	B
0	1	0	0	0	C	U	M	B
0	1	0	0	1	C	U	M	B
0	1	0	1	0	C	U	M	B
0	1	0	1	1	C	U	M	B
0	1	1	0	0	C	U	M	B
0	1	1	0	1	C	U	M	B
0	1	1	1	0	C	U	M	B
0	1	1	1	1	C	U	M	B
1	0	0	0	0	C			
1	0	0	0	1	C	B		

1	0	0	1	0	C	M		
1	0	0	1	1	C	M	B	
1	0	1	0	0	C	U		
1	0	1	0	1	C	U	B	
1	0	1	1	0	C	U	M	
1	0	1	1	1	C	U	M	B
1	1	0	0	0	U	M	B	
1	1	0	0	1	U	M	B	
1	1	0	1	0	U	M	B	
1	1	0	1	1	U	M	B	
1	1	1	0	0	C	U	M	B
1	1	1	0	1	C	U	M	B
1	1	1	1	0	C	U	M	B
1	1	1	1	1	C	U	M	B

Table 6.12-2 Different CAMCMR Setting and Type of Received Packet

CAM Enable Register (ECAM_CAMEN)

The ECAM_CAMEN controls the validation of each CAM entry. Each CAM entry must be enabled first before it participates in the destination MAC address recognition.

Register	Offset	R/W	Description	Reset Value
ECAM_CAMEN	EMAC_BA+0x004	R/W	CAM Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CAM15EN	**CAM14EN**	**CAM13EN**	**CAM12EN**	**CAM11EN**	**CAM10EN**	**CAM9EN**	**CAM8EN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CAM7EN	**CAM6EN**	**CAM5EN**	**CAM4EN**	**CAM3EN**	**CAM2EN**	**CAM1EN**	**CAM0EN**

Bits	Description	
[31:16]	**Reserved**	Reserved.
[x]	**CAMxEN**	**CAM Entry X Enable Bit** The CAMxEN controls the validation of CAM entry x. The CAM entry 13, 14 and 15 are for PAUSE control frame transmission. If software wants to transmit a PAUSE control frame out to network, the enable bits of these three CAM entries all must be enabled first. 0 = CAM entry x Disabled. 1 = CAM entry x Enabled.

CAM Entry Register (EMAC_CAMxMSB, x = 0, 1, 2..14)

The EMAC is equipped with 16 CAM entries. In these 16 CAM entries, 13 entries (entry 0~12) are to keep destination MAC address for packet recognition, and the other 3 entries (entry 13~15) are for PAUSE control frame transmission. Each CAM entry consists of 6 bytes. Consequently, 2 register are us for each CAM entry.

For packet recognition, a register pair {EMAC_CAMxMSB, ECAMxL} represents a CAM entry and can keep a destination MAC address. The corresponding CAM enable bit CAMxEN (ECAM_CAMEN[x]) is also needed be enabled. The x can be the 0 to 12.

The register pairs {EMAC_CAM13M, EMAC_CAM13L}, {EMAC_CAM14M, EMAC_CAM14L} and {EMAC_CAM15MSB, EMAC_CAM15LSB} are used for flow control function.

Register	Offset	R/W	Description	Reset Value
EMAC_CAM0M	EMAC_BA+0x008	R/W	CAM0 Most Significant Word Register	0x0000_0000
EMAC_CAM1M	EMAC_BA+0x010	R/W	CAM1 Most Significant Word Register	0x0000_0000
EMAC_CAM2M	EMAC_BA+0x018	R/W	CAM2 Most Significant Word Register	0x0000_0000
EMAC_CAM3M	EMAC_BA+0x020	R/W	CAM3 Most Significant Word Register	0x0000_0000
EMAC_CAM4M	EMAC_BA+0x028	R/W	CAM4 Most Significant Word Register	0x0000_0000
EMAC_CAM5M	EMAC_BA+0x030	R/W	CAM5 Most Significant Word Register	0x0000_0000
EMAC_CAM6M	EMAC_BA+0x038	R/W	CAM6 Most Significant Word Register	0x0000_0000
EMAC_CAM7M	EMAC_BA+0x040	R/W	CAM7 Most Significant Word Register	0x0000_0000
EMAC_CAM8M	EMAC_BA+0x048	R/W	CAM8 Most Significant Word Register	0x0000_0000
EMAC_CAM9M	EMAC_BA+0x050	R/W	CAM9 Most Significant Word Register	0x0000_0000
EMAC_CAM10M	EMAC_BA+0x058	R/W	CAM10 Most Significant Word Register	0x0000_0000
EMAC_CAM11M	EMAC_BA+0x060	R/W	CAM11 Most Significant Word Register	0x0000_0000
EMAC_CAM12M	EMAC_BA+0x068	R/W	CAM12 Most Significant Word Register	0x0000_0000
EMAC_CAM13M	EMAC_BA+0x070	R/W	CAM13 Most Significant Word Register	0x0000_0000
EMAC_CAM14M	EMAC_BA+0x078	R/W	CAM14 Most Significant Word Register	0x0000_0000

31	30	29	28	27	26	25	24
MACADDR5							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
MACADDR4							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
MACADDR3							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

MACADDR2

Bits	Description	
[31:24]	**MACADDR5**	**MAC Address Byte 5** The CAMxM keeps the bit 47~16 of MAC address. The x can be the 0~14. The register pair {EMAC_CAMxM, EMAC_CAMxL} represents a CAM entry and keeps a MAC address. For example, if the MAC address 00-50-BA-33-BA-44 kept in CAM entry 1, the register EMAC_CAM1M is 0x0050_BA33 and EMAC_CAM1L is 0xBA44_0000.
[23:16]	**MACADDR4**	**MAC Address Byte 4**
[15:8]	**MACADDR3**	**MAC Address Byte 3**
[7:0]	**MACADDR2**	**MAC Address Byte 2**

CAM Entry Register (EMAC_CAMxLSB; x = 0, 1, 2..14)

Register	Offset	R/W	Description	Reset Value
EMAC_CAM0L	EMAC_BA+0x00C	R/W	CAM0 Least Significant Word Register	0x0000_0000
EMAC_CAM1L	EMAC_BA+0x014	R/W	CAM1 Least Significant Word Register	0x0000_0000
EMAC_CAM2L	EMAC_BA+0x01C	R/W	CAM2 Least Significant Word Register	0x0000_0000
EMAC_CAM3L	EMAC_BA+0x024	R/W	CAM3 Least Significant Word Register	0x0000_0000
EMAC_CAM4L	EMAC_BA+0x02C	R/W	CAM4 Least Significant Word Register	0x0000_0000
EMAC_CAM5L	EMAC_BA+0x034	R/W	CAM5 Least Significant Word Register	0x0000_0000
EMAC_CAM6L	EMAC_BA+0x03C	R/W	CAM6 Least Significant Word Register	0x0000_0000
EMAC_CAM7L	EMAC_BA+0x044	R/W	CAM7 Least Significant Word Register	0x0000_0000
EMAC_CAM8L	EMAC_BA+0x04C	R/W	CAM8 Least Significant Word Register	0x0000_0000
EMAC_CAM9L	EMAC_BA+0x054	R/W	CAM9 Least Significant Word Register	0x0000_0000
EMAC_CAM10L	EMAC_BA+0x05C	R/W	CAM10 Least Significant Word Register	0x0000_0000
EMAC_CAM11L	EMAC_BA+0x064	R/W	CAM11 Least Significant Word Register	0x0000_0000
EMAC_CAM12L	EMAC_BA+0x06C	R/W	CAM12 Least Significant Word Register	0x0000_0000
EMAC_CAM13L	EMAC_BA+0x074	R/W	CAM13 Least Significant Word Register	0x0000_0000
EMAC_CAM14L	EMAC_BA+0x07C	R/W	CAM14 Least Significant Word Register	0x0000_0000

31	30	29	28	27	26	25	24
MACADDR1							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
MACADDR0							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:24]	**MACADDR1**	**MAC Address Byte 1** The CAMxL keeps the bit 15~0 of MAC address. The x can be the 0~14. The register pair {EMAC_CAMxM, EMAC_CAMxL} represents a CAM entry and keeps a MAC address. For example, if the MAC address 00-50-BA-33-BA-44 kept in CAM entry 1, the register

		EMAC_CAM1M is 0x0050_BA33 and EMAC_CAM1L is 0xBA44_0000.
[23:16]	**MACADDR0**	**MAC Address Byte 0**
[15:0]	**Rserved**	Reserved

CAM Entry Register (EMAC_CAM15MSB)

Register	Offset	R/W	Description	Reset Value
EMAC_CAM15MSB	EMAC_BA+0x080	R/W	CAM15 Most Significant Word Register	0x0000_0000

31	30	29	28	27	26	25	24
LENGTH							
23	22	21	20	19	18	17	16
LENGTH							
15	14	13	12	11	10	9	8
OPCODE							
7	6	5	4	3	2	1	0
OPCODE							

Bits	Description	
[31:16]	LENGTH	**LENGTH Field Of PAUSE Control Frame** In the PAUSE control frame, a LENGTH field defined and is 16"h8808.
[15:0]	OPCODE	**OP Code Field Of PAUSE Control Frame** In the PAUSE control frame, an op code field defined and is 16"h0001.

CAM Entry Register (EMAC_CAM15LSB)

Register	Offset	R/W	Description	Reset Value
EMAC_CAM15LSB	EMAC_BA+0x084	R/W	CAM15 Least Significant Word Register	0x0000_0000

31	30	29	28	27	26	25	24
OPERAND							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							

Bits	Description	
[31:24]	OPERAND	**Pause Parameter** In the PAUSE control frame, an OPERAND field defined and controls how much time the destination Ethernet MAC Controller paused. The unit of the OPERAND is a slot time, the 512 bits time.
[23:0]	Reserved	Reserved.

Transmit Descriptor Link List Start Address Register (EMAC_TXDSA)

The TX descriptor defined in EMAC is a link-list data structure. The EMAC_TXDSA keeps the starting address of this link-list. In other words, the EMAC_TXDSA keeps the starting address of the 1st TX descriptor. EMAC_TXDSA must be configured by software before the bit TXON (EMAC_CTL[8]) is enabled.

Register	Offset	R/W	Description	Reset Value
EMAC_TXDSA	EMAC_BA+0x088	R/W	Transmit Descriptor Link List Start Address Register	0xFFFF_FFFC

31	30	29	28	27	26	25	24
TXDSA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TXDSA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TXDSA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TXDSA							

Bits	Description	
[31:0]	**TXDSA**	**Transmit Descriptor Link-List Start Address** The TXDSA keeps the start address of transmit descriptor link-list. If the software enables the bit TXON (EMAC_CTL[8]), the content of TXDSA will be loaded into the current transmit descriptor start address register (EMAC_CTXDSA). The TXDSA does not be updated by EMAC. During the operation, EMAC will ignore the bits [1:0] of TXDSA. This means that TX descriptors must locate at word boundary memory address.

Receive Descriptor Link List Start Address Register (EMAC_RXDSA)

The RX descriptor defined in EMAC is a link-list data structure. The EMAC_RXDSA keeps the starting address of this link-list. In other words, the EMAC_RXDSA keeps the starting address of the 1st RX descriptor. EMAC_RXDSA must be configured by software before the bit RXON (EMAC_CTL[0]) is enabled.

Register	Offset	R/W	Description	Reset Value
EMAC_RXDSA	EMAC_BA+0x08C	R/W	Receive Descriptor Link List Start Address Register	0xFFFF_FFFC

31	30	29	28	27	26	25	24
RXDSA							
23	22	21	20	19	18	17	16
RXDSA							
15	14	13	12	11	10	9	8
RXDSA							
7	6	5	4	3	2	1	0
RXDSA							

Bits	Description	
[31:0]	**RXDSA**	**Receive Descriptor Link-List Start Address** The RXDSA keeps the start address of receive descriptor link-list. If the S/W enables the bit RXON (EMAC_CTL[0]), the content of RXDSA will be loaded into the current receive descriptor start address register (EMAC_CRXDSA). The RXDSA does not be updated by EMAC. During the operation, EMAC will ignore the bits [1:0] of RXDSA. This means that RX descriptors must locate at word boundary memory address.

MAC Control Register (EMAC_CTL)

The EMAC_CTL provides the control information for EMAC. Some command settings affect both frame transmission and reception, such as bit FUDUP (EMAC_CTL[18]), the full/half duplex mode selection, or bit OPMODE (EMAC_CTL[20]), the 100/10M bps mode selection. Some command settings control frame transmission and reception separately, likes bit TXON (EMAC_CTL[8]) and RXON (EMAC_CTL[0]).

Register	Offset	R/W	Description	Reset Value
EMAC_CTL	EMAC_BA+0x090	R/W	MAC Control Register	0x0040_0000

31	30	29	28	27	26	25	24
Reserved							RST
23	22	21	20	19	18	17	16
Reserved	RMIIEN	Reserved	OPMODE	RMIIRXCTL	FUDUP	SQECHKEN	SDPZ
15	14	13	12	11	10	9	8
Reserved						NODEF	TXON
7	6	5	4	3	2	1	0
Reserved	WOLEN	STRIPCRC	AEP	ACP	ARP	ALP	RXON

Bits	Description	
[31:25]	**Reserved**	Reserved.
[24]	**RST**	**Software Reset** The RST implements a reset function to make the EMAC return default state. The RST is a self-clear bit. This means after the software reset finished, the RST will be cleared automatically. Enable RST can also reset all control and status registers, exclusive of the control bits RMIIEN (EMAC_CTL[22]), and OPMODE (EMAC_CTL[20]). The EMAC re-initial is necessary after the software reset completed. 0 = Software reset completed. 1 = Software reset Enabled.
[23]	**Reserved**	Reserved.
[22]	**RMIIEN**	**RMII Mode Enable** This bit controls if Ethernet MAC controller connected with off-chip Ethernet PHY by MII interface or RMII interface. The RST (EMAC_CTL[24]) would not affect RMIIEN value. 0 = Ethernet MAC controller MII mode Enabled. 1 = Ethernet MAC controller RMII mode Enabled.
[21]	**Reserved**	Reserved.
[20]	**OPMODE**	**Operation Mode Selection** The OPMODE defines that if the EMAC is operating on 10M or 100M bps mode. The RST (EMAC_CTL[24]) would not affect OPMODE value. 0 = EMAC operates in 10Mbps mode. 1 = EMAC operates in 100Mbps mode.

[19]	**RMIIRXCTL**	**RMII RX Control** The RMIIRXCTL control the receive data sample in RMII mode. It's necessary to set this bit high when RMIIEN (EMAC_CTL[[22]) is high. 0 = RMII RX control disabled. 1 = RMII RX control enabled.
[18]	**FUDUP**	**Full Duplex Mode Selection** The FUDUP controls that if EMAC is operating on full or half duplex mode. 0 = EMAC operates in half duplex mode. 1 = EMAC operates in full duplex mode.
[17]	**SQECHKEN**	**SQE Checking Enable** The SQECHKEN controls the enable of SQE checking. The SQE checking is only available while EMAC is operating on 10M bps and half duplex mode. In other words, the SQECHKEN cannot affect EMAC operation, if the EMAC is operating on 100M bps or full duplex mode. 0 = SQE checking Disabled while EMAC is operating in 10Mbps and Half Duplex mode. 1 = SQE checking Enabled while EMAC is operating in 10Mbps and Half Duplex mode.
[16]	**SDPZ**	**Send PAUSE Frame** The SDPZ controls the PAUSE control frame transmission. If S/W wants to send a PAUSE control frame out, the CAM entry 13, 14 and 15 must be configured first and the corresponding CAM enable bit of CAMEN register also must be set. Then, set SDPZ to 1 enables the PAUSE control frame transmission. The SDPZ is a self-clear bit. This means after the PAUSE control frame transmission has completed, the SDPZ will be cleared automatically. It is recommended that only enabling SNDPAUSE while EMAC is operating in Full Duplex mode. 0 = PAUSE control frame transmission completed. 1 = PAUSE control frame transmission Enabled.
[15:10]	**Reserved**	Reserved.
[9]	**NODEF**	**No Deferral** The NODEF controls the enable of deferral exceed counter. If NODEF is set to high, the deferral exceed counter is disabled. The NODEF is only useful while EMAC is operating on half duplex mode. 0 = The deferral exceed counter Enabled. 1 = The deferral exceed counter Disabled.
[8]	**TXON**	**Frame Transmission ON** The TXON controls the normal packet transmission of EMAC. If the TXON is set to high, the EMAC starts the packet transmission process, including the TX descriptor fetching, packet transmission and TX descriptor modification. It is must to finish EMAC initial sequence before enable TXON. Otherwise, the EMAC operation is undefined. If the TXON is disabled during EMAC is transmitting a packet out, the EMAC stops the packet transmission process after the current packet transmission finished. 0 = Packet transmission process stopped. 1 = Packet transmission process started.
[7]	**Reserved**	Reserved.

[6]	**WOLEN**	**Wake On LAN Enable** The WOLEN high enables the functionality that Ethernet MAC controller checked if the incoming packet is Magic Packet and wakeup system from Power-down mode. If incoming packet was a Magic Packet and the system was in Power-down, the Ethernet MAC controller would generate a wakeup event to wake system up from Power-down mode. 0 = Wake-up by Magic Packet function Disabled. 1 = Wake-up by Magic Packet function Enabled.
[5]	**STRIPCRC**	**Strip CRC Checksum** The STRIPCRC controls if the length of incoming packet is calculated with 4 bytes CRC checksum. If the STRIPCRC is set to high, 4 bytes CRC checksum is excluded from length calculation of incoming packet. 0 = The 4 bytes CRC checksum is included in packet length calculation. 1 = The 4 bytes CRC checksum is excluded in packet length calculation.
[4]	**AEP**	**Accept CRC Error Packet** The AEP controls the EMAC accepts or drops the CRC error packet. If the AEP is set to high, the incoming packet with CRC error will be received by EMAC as a good packet. 0 = Ethernet MAC controller dropped the CRC error packet. 1 = Ethernet MAC controller received the CRC error packet.
[3]	**ACP**	**Accept Control Packet** The ACP controls the control frame reception. If the ACP is set to high, the EMAC will accept the control frame. Otherwise, the control frame will be dropped. It is recommended that S/W only enable ACP while EMAC is operating on full duplex mode. 0 = Ethernet MAC controller dropped the control frame. 1 = Ethernet MAC controller received the control frame.
[2]	**ARP**	**Accept Runt Packet** The ARP controls the runt packet, which length is less than 64 bytes, reception. If the ARP is set to high, the EMAC will accept the runt packet. Otherwise, the runt packet will be dropped. 0 = Ethernet MAC controller dropped the runt packet. 1 = Ethernet MAC controller received the runt packet.
[1]	**ALP**	**Accept Long Packet** The ALP controls the long packet, which packet length is greater than 1518 bytes, reception. If the ALP is set to high, the EMAC will accept the long packet. Otherwise, the long packet will be dropped. 0 = Ethernet MAC controller dropped the long packet. 1 = Ethernet MAC controller received the long packet.
[0]	**RXON**	**Frame Reception ON** The RXON controls the normal packet reception of EMAC. If the RXON is set to high, the EMAC starts the packet reception process, including the RX descriptor fetching, packet reception and RX descriptor modification. It is necessary to finish EMAC initial sequence before enable RXON. Otherwise, the EMAC operation is undefined. If the RXON is disabled during EMAC is receiving an incoming packet, the EMAC stops the packet reception process after the current packet reception finished. 0 = Packet reception process stopped. 1 = Packet reception process started.

MII Management Data Register (EMAC_MIIMDAT)

The EMAC provides MII management function to access the control and status registers of the external PHY. The EMAC_MIIMDAT register is used to store the data that will be written into the registers of external PHY for write command or the data that is read from the registers of external PHY for read command.

Register	Offset	R/W	Description	Reset Value
EMAC_MIIMDAT	EMAC_BA+0x094	R/W	MII Management Data Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DATA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATA							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**DATA**	**MII Management Data** The DATA is the 16 bits data that will be written into the registers of external PHY for MII Management write command or the data from the registers of external PHY for MII Management read command.

MII Management Control and Address Register (EMAC_MIIMCTL)

The EMAC provides MII management function to access the control and status registers of the external PHY. The EMAC_MIIMCTL register is used to keep the MII management command information, like the register address, external PHY address, MDC clocking rate, read/write etc.

Register	Offset	R/W	Description	Reset Value
EMAC_MIIMCTL	EMAC_BA+0x098	R/W	MII Management Control and Address Register	0x0090_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				**MDCON**	**PREAMSP**	**BUSY**	**WRITE**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			**PHYADDR**				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**PHYREG**				

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19]	**MDCON**	**MDC Clock ON** The MDC controls the MDC clock generation. If the MDCON is set to high, the MDC clock is turned on. 0 = MDC clock off. 1 = MDC clock on.
[18]	**PREAMSP**	**Preamble Suppress** The PREAMSP controls the preamble field generation of MII management frame. If the PREAMSP is set to high, the preamble field generation of MII management frame is skipped. 0 = Preamble field generation of MII management frame not skipped. 1 = Preamble field generation of MII management frame skipped.
[17]	**BUSY**	**Busy Bit** The BUSY controls the enable of the MII management frame generation. If S/W wants to access registers of external PHY, it set BUSY to high and EMAC generates the MII management frame to external PHY through MII Management I/F. The BUSY is a self-clear bit. This means the BUSY will be cleared automatically after the MII management command finished. 0 = MII management command generation finished. 1 = MII management command generation Enabled.
[16]	**WRITE**	**Write Command** The Write defines the MII management command is a read or write. 0 = MII management command is a read command. 1 = MII management command is a write command.

[15:13]	**Reserved**	Reserved.
[12:8]	**PHYADDR**	**PHY Address** The PHYADDR keeps the address to differentiate which external PHY is the target of the MII management command.
[7:5]	**Reserved**	Reserved.
[4:0]	**PHYREG**	**PHY Register Address** The PHYREG keeps the address to indicate which register of external PHY is the target of the MII management command.

MII Management Function Frame Format

In IEEE Std. 802.3 clause 22.2.4, the MII management function is defined. The MII management function is used for the purpose of controlling the PHY and gathering status from the PHY. The MII management frame format is shown as follow.

	Management Frame Fields							
	PRE	**ST**	**OP**	**PHYAD**	**REGAD**	**TA**	**DATA**	**IDLE**
READ	**1…1**	**01**	**10**	**AAAAA**	**RRRRR**	**Z0**	**DDDDDDDDDDDDDDDD**	**Z**
WRITE	**1…1**	**01**	**01**	**AAAAA**	**RRRRR**	**10**	**DDDDDDDDDDDDDDDD**	**Z**

Figure 6.12-6 MII Management Frame Format

MII Management Function Configure Sequence

Read	**Write**
1. Set appropriate EMAC_MDCCR.	1. Write data to EMAC_MIIMDAT register
2. Set PHYADDR and PHYREG.	2. Set appropriate EMAC_MDCCR.
3. Set Write to 1"b0	3. Set PHYADDR and PHYREG.
4. Set bit BUSY (EMAC_MIIMCTL[17]) to 1"b1 to send a MII management frame out.	4. Set Write to 1"b1
5. Wait BUSY (EMAC_MIIMCTL[17]) to become 1"b0.	5. Set bit BUSY (EMAC_MIIMCTL[17]) to 1"b1 to send a MII management frame out.
6. Read data from EMAC_MIIMDAT register.	6. Wait BUSY (EMAC_MIIMCTL[17]) to become 1"b0.
7. Finish the read command.	7. Finish the write command.

Table 6.12-3 MII Management Function Configure Sequence

FIFO Threshold Control Register (EMAC_FIFOCTL)

The EMAC_FIFOCTL defines the high and low threshold of internal FIFOs, including TXFIFO and RXFIFO. The threshold of internal FIFOs is related to EMAC request generation and when the frame transmission starts. The EMAC_FIFOCTL also defines the burst length of AHB bus cycle for system memory access.

Register	Offset	R/W	Description	Reset Value
EMAC_FIFOCTL	EMAC_BA+0x09C	R/W	FIFO Threshold Control Register	0x0000_0101

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		BURSTLEN		Reserved			
15	14	13	12	11	10	9	8
Reserved						TXFIFOTH	
7	6	5	4	3	2	1	0
Reserved						RXFIFOTH	

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21:20]	**BURSTLEN**	**DMA Burst Length** This defines the burst length of AHB bus cycle while EMAC accesses system memory. 00 = 16 words. 01 = 16 words. 10 = 8 words. 11 = 4 words.
[19:10]	**Reserved**	Reserved.
[9:8]	**TXFIFOTH**	**TXFIFO Low Threshold** The TXFIFOTH controls when TXDMA requests internal arbiter for data transfer between system memory and TXFIFO. The TXFIFOTH defines not only the low threshold of TXFIFO, but also the high threshold. The high threshold is the twice of low threshold always. During the packet transmission, if the TXFIFO reaches the high threshold, the TXDMA stops generate request to transfer frame data from system memory to TXFIFO. If the frame data in TXFIFO is less than low threshold, TXDMA starts to transfer frame data from system memory to TXFIFO. The TXFIFOTH also defines when the TXMAC starts to transmit frame out to network. The TXMAC starts to transmit the frame out while the TXFIFO first time reaches the high threshold during the transmission of the frame. If the frame data length is less than TXFIFO high threshold, the TXMAC starts to transmit the frame out after the frame data are all inside the TXFIFO. 00 = Undefined. 01 = TXFIFO low threshold is 64B and high threshold is 128B. 10 = TXFIFO low threshold is 80B and high threshold is 160B. 11 = TXFIFO low threshold is 96B and high threshold is 192B.

[7:2]	**Reserved**	Reserved.
[1:0]	**RXFIFOTH**	**RXFIFO Low Threshold** The RXFIFOTH controls when RXDMA requests internal arbiter for data transfer between RXFIFO and system memory. The RXFIFOTH defines not only the high threshold of RXFIFO, but also the low threshold. The low threshold is the half of high threshold always. During the packet reception, if the RXFIFO reaches the high threshold, the RXDMA starts to transfer frame data from RXFIFO to system memory. If the frame data in RXFIFO is less than low threshold, RXDMA stops to transfer the frame data to system memory. 00 = Depend on the burst length setting. If the burst length is 8 words, high threshold is 8 words, too. 01 = RXFIFO high threshold is 64B and low threshold is 32B. 10 = RXFIFO high threshold is 128B and low threshold is 64B. 11 = RXFIFO high threshold is 192B and low threshold is 96B.

Transmit Start Demand Register (EMAC_TXST)

S/W issues a write command to EMAC_TXST register to make TXDMA to leave Halt state and continue the frame transmission.

Register	Offset	R/W	Description	Reset Value
EMAC_TXST	EMAC_BA+0x0A0	W	Transmit Start Demand Register	Undefined

31	30	29	28	27	26	25	24
TXST							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TXST							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TXST							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TXST							

Bits	Description	
[31:0]	**TXST**	**Transmit Start Demand** If the TX descriptor is not available for use of TXDMA after the TXON (EMAC_CTL[8]) is enabled, the FSM (Finite State Machine) of TXDMA enters the Halt state and the frame transmission is halted. After the S/W has prepared the new TX descriptor for frame transmission, it must issue a write command to EMAC_TXST register to make TXDMA to leave Halt state and continue the frame transmission. The EMAC_TXST is a write only register and read from this register is undefined. The write to EMAC_TXST register takes effect only when TXDMA stayed at Halt state.

Receive Start Demand Register (EMAC_RXST)

S/W issues a write command to EMAC_RXST register to make RXDMA to leave Halt state and continue the frame reception.

Register	Offset	R/W	Description	Reset Value
EMAC_RXST	EMAC_BA+0x0A4	W	Receive Start Demand Register	Undefined

31	30	29	28	27	26	25	24
RXST							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RXST							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RXST							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RXST							

Bits	Description	
[31:0]	**RXST**	**Receive Start Demand** If the RX descriptor is not available for use of RXDMA after the RXON (EMAC_CTL[0]) is enabled, the FSM (Finite State Machine) of RXDMA enters the Halt state and the frame reception is halted. After the S/W has prepared the new RX descriptor for frame reception, it must issue a write command to EMAC_RXST register to make RXDMA to leave Halt state and continue the frame reception. The EMAC_RXST is a write only register and read from this register is undefined. The write to EMAC_RXST register take effect only when RXDMA stayed at Halt state.

Maximum Receive Frame Control Register (EMAC_MRFL)

The EMAC_MRFL defines the maximum frame length for a received frame that can be stored in the system memory. It is recommend that only use this register while S/W wants to receive a frame which length is greater than 1518 bytes.

Register	Offset	R/W	Description	Reset Value
EMAC_MRFL	EMAC_BA+0x0A8	R/W	Maximum Receive Frame Control Register	0x0000_0800

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
MRFL							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MRFL							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**MRFL**	**Maximum Receive Frame Length** The MRFL defines the maximum frame length for received frame. If the frame length of received frame is greater than MRFL, and bit MFLEIEN (EMAC_INTEN[8]) is also enabled, the bit MFLEIF (EMAC_INTSTS[8]) is set and the RX interrupt is triggered. It is recommended that only use MRFL to qualify the length of received frame while S/W wants to receive a frame which length is greater than 1518 bytes.

MAC Interrupt Enable Register (EMAC_INTEN)

The EMAC_INTEN controls the enable of EMAC interrupt status to generate interrupt. Two interrupts, RXIF for frame reception and TXIF for frame transmission, are generated from EMAC to CPU.

Register	Offset	R/W	Description	Reset Value
EMAC_INTEN	EMAC_BA+0x0AC	R/W	MAC Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved			TSALMIEN	Reserved			TXBEIEN
23	22	21	20	19	18	17	16
TDUIEN	LCIEN	TXABTIEN	NCSIEN	EXDEFIEN	TXCPIEN	TXUDIEN	TXIEN
15	14	13	12	11	10	9	8
WOLIEN	CFRIEN	Reserved		RXBEIEN	RDUIEN	DENIEN	MFLEIEN
7	6	5	4	3	2	1	0
MPCOVIEN	RPIEN	ALIEIEN	RXGDIEN	LPIEN	RXOVIEN	CRCEIEN	RXIEN

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28]	**TSALMIEN**	**Time Stamp Alarm Interrupt Enable Bit** The TSALMIEN controls the TSALMIF (EMAC_INTSTS[28]) interrupt generation. If TSALMIF (EMAC_INTSTS[28]) is set, and both TSALMIEN and TXIEN (EMAC_INTEN[16]) enabled, the EMAC generates the TX interrupt to CPU. If TSALMIEN or TXIEN (EMAC_INTEN[16]) disabled, no TX interrupt generated to CPU even the TXTSALMIF (EMAC_INTEN[28]) is set. 0 = TXTSALMIF (EMAC_INTSTS[28]) trigger TX interrupt Disabled. 1 = TXTSALMIF (EMAC_INTSTS[28]) trigger TX interrupt Enabled.
[27:25]	**Reserved**	Reserved.
[24]	**TXBEIEN**	**Transmit Bus Error Interrupt Enable Bit** The TXBEIEN controls the TXBEIF (EMAC_INTSTS[24]) interrupt generation. If TXBEIF (EMAC_INTSTS[24]) is set, and both TXBEIEN and TXIEN (EMAC_INTEN[16]) are enabled, the EMAC generates the TX interrupt to CPU. If TXBEIEN or TXIEN (EMAC_INTEN[16]) is disabled, no TX interrupt is generated to CPU even the TXBEIF (EMAC_INTSTS[24]) is set. 0 = TXBEIF (EMAC_INTSTS[24]) trigger TX interrupt Disabled. 1 = TXBEIF (EMAC_INTSTS[24]) trigger TX interrupt Enabled.
[23]	**TDUIEN**	**Transmit Descriptor Unavailable Interrupt Enable Bit** The TDUIEN controls the TDUIF (EMAC_INTSTS[23]) interrupt generation. If TDUIF (EMAC_INTSTS[23]) is set, and both TDUIEN and TXIEN (EMAC_INTEN[16]) are enabled, the EMAC generates the TX interrupt to CPU. If TDUIEN or TXIEN (EMAC_INTEN[16]) is disabled, no TX interrupt is generated to CPU even the TDUIF (EMAC_INTSTS[23]) is set. 0 = TDUIF (EMAC_INTSTS[23]) trigger TX interrupt Disabled. 1 = TDUIF (EMAC_INTSTS[23]) trigger TX interrupt Enabled.

[22]	**LCIEN**	**Late Collision Interrupt Enable Bit** The LCIEN controls the LCIF (EMAC_INTSTS[22]) interrupt generation. If LCIF (EMAC_INTSTS[22]) is set, and both LCIEN and TXIEN (EMAC_INTEN[16]) are enabled, the EMAC generates the TX interrupt to CPU. If LCIEN or TXIEN (EMAC_INTEN[16]) is disabled, no TX interrupt is generated to CPU even the LCIF (EMAC_INTSTS[22]) is set. 0 = LCIF (EMAC_INTSTS[22]) trigger TX interrupt Disabled. 1 = LCIF (EMAC_INTSTS[22]) trigger TX interrupt Enabled.
[21]	**TXABTIEN**	**Transmit Abort Interrupt Enable Bit** The TXABTIEN controls the TXABTIF (EMAC_INTSTS[21]) interrupt generation. If TXABTIF (EMAC_INTSTS[21]) is set, and both TXABTIEN and TXIEN (EMAC_INTEN[16]) are enabled, the EMAC generates the TX interrupt to CPU. If TXABTIEN or TXIEN (EMAC_INTEN[16]) is disabled, no TX interrupt is generated to CPU even the TXABTIF (EMAC_INTSTS[21]) is set. 0 = TXABTIF (EMAC_INTSTS[21]) trigger TX interrupt Disabled. 1 = TXABTIF (EMAC_INTSTS[21]) trigger TX interrupt Enabled.
[20]	**NCSIEN**	**No Carrier Sense Interrupt Enable Bit** The NCSIEN controls the NCSIF (EMAC_INTSTS[20]) interrupt generation. If NCSIF (EMAC_INTSTS[20]) is set, and both NCSIEN and TXIEN (EMAC_INTEN[16]) are enabled, the EMAC generates the TX interrupt to CPU. If NCSIEN or TXIEN (EMAC_INTEN[16]) is disabled, no TX interrupt is generated to CPU even the NCSIF (EMAC_INTSTS[20]) is set. 0 = NCSIF (EMAC_INTSTS[20]) trigger TX interrupt Disabled. 1 = NCSIF (EMAC_INTSTS[20]) trigger TX interrupt Enabled.
[19]	**EXDEFIEN**	**Defer Exceed Interrupt Enable Bit** The EXDEFIEN controls the EXDEFIF (EMAC_INTSTS[19]) interrupt generation. If EXDEFIF (EMAC_INTSTS[19]) is set, and both EXDEFIEN and TXIEN (EMAC_INTEN[16]) are enabled, the EMAC generates the TX interrupt to CPU. If EXDEFIEN or TXIEN (EMAC_INTEN[16]) is disabled, no TX interrupt is generated to CPU even the EXDEFIF (EMAC_INTSTS[19]) is set. 0 = EXDEFIF (EMAC_INTSTS[19]) trigger TX interrupt Disabled. 1 = EXDEFIF (EMAC_INTSTS[19]) trigger TX interrupt Enabled.
[18]	**TXCPIEN**	**Transmit Completion Interrupt Enable Bit** The TXCPIEN controls the TXCPIF (EMAC_INTSTS[18]) interrupt generation. If TXCPIF (EMAC_INTSTS[18]) is set, and both TXCPIEN and TXIEN (EMAC_INTEN[16]) are enabled, the EMAC generates the TX interrupt to CPU. If TXCPIEN or TXIEN (EMAC_INTEN[16]) is disabled, no TX interrupt is generated to CPU even the TXCPIF (EMAC_INTSTS[18]) is set. 0 = TXCPIF (EMAC_INTSTS[18]) trigger TX interrupt Disabled. 1 = TXCPIF (EMAC_INTSTS[18]) trigger TX interrupt Enabled.
[17]	**TXUDIEN**	**Transmit FIFO Underflow Interrupt Enable Bit** The TXUDIEN controls the TXUDIF (EMAC_INTSTS[17]) interrupt generation. If TXUDIF (EMAC_INTSTS[17]) is set, and both TXUDIEN and TXIEN (EMAC_INTEN[16]) are enabled, the EMAC generates the TX interrupt to CPU. If TXUDIEN or TXIEN (EMAC_INTEN[16]) is disabled, no TX interrupt is generated to CPU even the TXUDIF (EMAC_INTSTS[17]) is set. 0 = TXUDIF (EMAC_INTSTS[17]) TX interrupt Disabled. 1 = TXUDIF (EMAC_INTSTS[17]) TX interrupt Enabled.

[16]	**TXIEN**	**Transmit Interrupt Enable Bit** The TXIEN controls the TX interrupt generation. If TXIEN is enabled and TXIF (EMAC_INTSTS[16]) is high, EMAC generates the TX interrupt to CPU. If TXIEN is disabled, no TX interrupt is generated to CPU even any status bit of EMAC_INTSTS[24:17] set and the corresponding bit of EMAC_INTEN is enabled. In other words, if S/W wants to receive TX interrupt from EMAC, this bit must be enabled. And, if S/W doesn't want to receive any TX interrupt from EMAC, disables this bit. 0 = TXIF (EMAC_INTSTS[16]) is masked and TX interrupt generation Disabled. 1 = TXIF (EMAC_INTSTS[16]) is not masked and TX interrupt generation Enabled.
[15]	**WOLIEN**	**Wake On LAN Interrupt Enable** The WOLIEN controls the WOLIF (EMAC_INTSTS[15]) interrupt generation. If WOLIF (EMAC_INTSTS[15]) is set, and both WOLIEN and RXIEN (EMAC_INTEN[0]) are enabled, the EMAC generates the RX interrupt to CPU. If WOLIEN or RXIEN (EMAC_INTEN[0]) is disabled, no RX interrupt is generated to CPU even the WOLIF (EMAC_INTSTS[15]) is set. 0 = WOLIF (EMAC_INTSTS[15]) trigger RX interrupt Disabled. 1 = WOLIF (EMAC_INTSTS[15]) trigger RX interrupt Enabled.
[14]	**CFRIEN**	**Control Frame Receive Interrupt Enable Bit** The CFRIEN controls the CFRIF (EMAC_INTSTS[14]) interrupt generation. If CFRIF (EMAC_INTSTS[14]) is set, and both CFRIEN and RXIEN (EMAC_INTEN[0]) are enabled, the EMAC generates the RX interrupt to CPU. If CFRIEN or RXIEN (EMAC_INTEN[0]) is disabled, no RX interrupt is generated to CPU even the CFRIF (EMAC_INTSTS[14]) register is set. 0 = CFRIF (EMAC_INTSTS[14]) trigger RX interrupt Disabled. 1 = CFRIF (EMAC_INTSTS[14]) trigger RX interrupt Enabled.
[13:12]	**Reserved**	Reserved.
[11]	**RXBEIEN**	**Receive Bus Error Interrupt Enable Bit** The RXBEIEN controls the RXBEIF (EMAC_INTSTS[11]) interrupt generation. If RXBEIF (EMAC_INTSTS[11]) is set, and both RXBEIEN and RXIEN (EMAC_INTEN[0]) are enabled, the EMAC generates the RX interrupt to CPU. If RXBEIEN or RXIEN (EMAC_INTEN[0]) is disabled, no RX interrupt is generated to CPU even the RXBEIF (EMAC_INTSTS[11]) is set. 0 = RXBEIF (EMAC_INTSTS[11]) trigger RX interrupt Disabled. 1 = RXBEIF (EMAC_INTSTS[11]) trigger RX interrupt Enabled.
[10]	**RDUIEN**	**Receive Descriptor Unavailable Interrupt Enable Bit** The RDUIEN controls the RDUIF (EMAC_INTSTS[10]) interrupt generation. If RDUIF (EMAC_INTSTS[10]) is set, and both RDUIEN and RXIEN (EMAC_INTEN[0]) are enabled, the EMAC generates the RX interrupt to CPU. If RDUIEN or RXIEN (EMAC_INTEN[0]) is disabled, no RX interrupt is generated to CPU even the RDUIF (EMAC_MIOSTA[10]) register is set. 0 = RDUIF (EMAC_INTSTS[10]) trigger RX interrupt Disabled. 1 = RDUIF (EMAC_INTSTS[10]) trigger RX interrupt Enabled.
[9]	**DENIEN**	**DMA Early Notification Interrupt Enable Bit** The DENIEN controls the DENIF (EMAC_INTSTS[9]) interrupt generation. If DENIF (EMAC_INTSTS[9]) is set, and both DENIEN and RXIEN (EMAC_INTEN[0]) are enabled, the EMAC generates the RX interrupt to CPU. If DENIEN or RXIEN (EMAC_INTEN[0]) is disabled, no RX interrupt is generated to CPU even the DENIF (EMAC_INTSTS[9]) is set. 0 = TDENIF (EMAC_INTSTS[9]) trigger RX interrupt Disabled. 1 = TDENIF (EMAC_INTSTS[9]) trigger RX interrupt Enabled.

[8]	**MFLEIEN**	**Maximum Frame Length Exceed Interrupt Enable** The MFLEIEN controls the MFLEIF (EMAC_INTSTS[8]) interrupt generation. If MFLEIF (EMAC_INTSTS[8]) is set, and both MFLEIEN and RXIEN (EMAC_INTEN[0]) are enabled, the EMAC generates the RX interrupt to CPU. If MFLEIEN or RXIEN (EMAC_INTEN[0]) is disabled, no RX interrupt is generated to CPU even the MFLEIF (EMAC_INTSTS[8]) is set. 0 = MFLEIF (EMAC_INTSTS[8]) trigger RX interrupt Disabled. 1 = MFLEIF (EMAC_INTSTS[8]) trigger RX interrupt Enabled.
[7]	**MPCOVIEN**	**Miss Packet Counter Overrun Interrupt Enable** The MPCOVIEN controls the MPCOVIF (EMAC_INTSTS[7]) interrupt generation. If MPCOVIF (EMAC_INTSTS[7]) is set, and both MPCOVIEN and RXIEN (EMAC_INTEN[0]) are enabled, the EMAC generates the RX interrupt to CPU. If MPCOVIEN or RXIEN (EMAC_INTEN[0]) is disabled, no RX interrupt is generated to CPU even the MPCOVIF (EMAC_INTSTS[7]) is set. 0 = MPCOVIF (EMAC_INTSTS[7]) trigger RX interrupt Disabled. 1 = MPCOVIF (EMAC_INTSTS[7]) trigger RX interrupt Enabled.
[6]	**RPIEN**	**Runt Packet Interrupt Enable Bit** The RPIEN controls the RPIF (EMAC_INTSTS[6]) interrupt generation. If RPIF (EMAC_INTSTS[6]) is set, and both RPIEN and RXIEN (EMAC_INTEN[0]) are enabled, the EMAC generates the RX interrupt to CPU. If RPIEN or RXIEN (EMAC_INTEN[0]) is disabled, no RX interrupt is generated to CPU even the RPIF (EMAC_INTSTS[6]) is set. 0 = RPIF (EMAC_INTSTS[6]) trigger RX interrupt Disabled. 1 = RPIF (EMAC_INTSTS[6]) trigger RX interrupt Enabled.
[5]	**ALIEIEN**	**Alignment Error Interrupt Enable Bit** The ALIEIEN controls the ALIEIF (EMAC_INTSTS[5]) interrupt generation. If ALIEIF (EMAC_INTSTS[5]) is set, and both ALIEIEN and RXIEN (EMAC_INTEN[0]) are enabled, the EMAC generates the RX interrupt to CPU. If ALIEIEN or RXIEN (EMAC_INTEN[0]) is disabled, no RX interrupt is generated to CPU even the ALIEIF (EMAC_INTSTS[5]) is set. 0 = ALIEIF (EMAC_INTSTS[5]) trigger RX interrupt Disabled. 1 = ALIEIF (EMAC_INTSTS[5]) trigger RX interrupt Enabled.
[4]	**RXGDIEN**	**Receive Good Interrupt Enable Bit** The RXGDIEN controls the RXGDIF (EMAC_INTSTS[4]) interrupt generation. If RXGDIF (EMAC_INTSTS[4]) is set, and both RXGDIEN and RXIEN (EMAC_INTEN[0]) are enabled, the EMAC generates the RX interrupt to CPU. If RXGDIEN or RXIEN (EMAC_INTEN[0]) is disabled, no RX interrupt is generated to CPU even the RXGDIF (EMAC_INTSTS[4]) is set. 0 = RXGDIF (EMAC_INTSTS[4]) trigger RX interrupt Disabled. 1 = RXGDIF (EMAC_INTSTS[4]) trigger RX interrupt Enabled.
[3]	**LPIEN**	**Long Packet Interrupt Enable** The LPIEN controls the LPIF (EMAC_INTSTS[3]) interrupt generation. If LPIF (EMAC_INTSTS[3]) is set, and both LPIEN and RXIEN (EMAC_INTEN[0]) are enabled, the EMAC generates the RX interrupt to CPU. If LPIEN or RXIEN (EMAC_INTEN[0]) is disabled, no RX interrupt is generated to CPU even the LPIF (EMAC_INTSTS[3]) is set. 0 = LPIF (EMAC_INTSTS[3]) trigger RX interrupt Disabled. 1 = LPIF (EMAC_INTSTS[3]) trigger RX interrupt Enabled.

[2]	**RXOVIEN**	**Receive FIFO Overflow Interrupt Enable Bit** The RXOVIEN controls the RXOVIF (EMAC_INTSTS[2]) interrupt generation. If RXOVIF (EMAC_INTSTS[2]) is set, and both RXOVIEN and RXIEN (EMAC_INTEN[0]) are enabled, the EMAC generates the RX interrupt to CPU. If RXOVIEN or RXIEN (EMAC_INTEN[0]) is disabled, no RX interrupt is generated to CPU even the RXOVIF (EMAC_INTSTS[2]) is set. 0 = RXOVIF (EMAC_INTSTS[2]) trigger RX interrupt Disabled. 1 = RXOVIF (EMAC_INTSTS[2]) trigger RX interrupt Enabled.
[1]	**CRCEIEN**	**CRC Error Interrupt Enable Bit** The CRCEIEN controls the CRCEIF (EMAC_INTSTS[1]) interrupt generation. If CRCEIF (EMAC_INTSTS[1]) is set, and both CRCEIEN and RXIEN (EMAC_INTEN[0]) are enabled, the EMAC generates the RX interrupt to CPU. If CRCEIEN or RXIEN (EMAC_INTEN[0]) is disabled, no RX interrupt is generated to CPU even the CRCEIF (EMAC_INTSTS[1]) is set. 0 = CRCEIF (EMAC_INTSTS[1]) trigger RX interrupt Disabled. 1 = CRCEIF (EMAC_INTSTS[1]) trigger RX interrupt Enabled.
[0]	**RXIEN**	**Receive Interrupt Enable Bit** The RXIEN controls the RX interrupt generation. If RXIEN is enabled and RXIF (EMAC_INTSTS[0]) is high, EMAC generates the RX interrupt to CPU. If RXIEN is disabled, no RX interrupt is generated to CPU even any status bit EMAC_INTSTS[15:1] is set and the corresponding bit of EMAC_INTEN is enabled. In other words, if S/W wants to receive RX interrupt from EMAC, this bit must be enabled. And, if S/W doesn't want to receive any RX interrupt from EMAC, disables this bit. 0 = RXIF (EMAC_INTSTS[0]) is masked and RX interrupt generation Disabled. 1 = RXIF (EMAC_INTSTS[0]) is not masked and RX interrupt generation Enabled.

MAC Interrupt Status Register (EMAC_INTSTS)

The EMAC_INTSTS keeps much EMAC statuses, such as frame transmission, reception status and internal FIFO status. The statuses kept in EMAC_INTSTS will trigger the reception or transmission interrupt. The EMAC_INTSTS is a write clear register and write 1 to corresponding bit clears the status and also clears the interrupt.

Register	Offset	R/W	Description	Reset Value
EMAC_INTSTS	EMAC_BA+0x0B0	R/W	MAC Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved			TSALMIF	Reserved			TXBEIF
23	22	21	20	19	18	17	16
TDUIF	LCIF	TXABTIF	NCSIF	EXDEFIF	TXCPIF	TXUDIF	TXIF
15	14	13	12	11	10	9	8
WOLIF	CFRIF	Reserved		RXBEIF	RDUIF	DENIF	MFLEIF
7	6	5	4	3	2	1	0
MPCOVIF	RPIF	ALIEIF	RXGDIF	LPIF	RXOVIF	CRCEIF	RXIF

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28]	**TSALMIF**	**Time Stamp Alarm Interrupt** The TSALMIF high indicates the EMAC_TSSEC register value equals to EMAC_ALMSEC register and EMAC_TSSUBSEC register value equals to register EMAC_TSMLSR. If TSALMIF is high and TSALMIEN (EMAC_INTEN[28]) enabled, the TXIF will be high. Write 1 to this bit clears the TSALMIF status. 0 = EMAC_TSSEC did not equal EMAC_ALMSEC or EMAC_TSSUBSEC did not equal EMAC_ALMSUBSEC. 1 = EMAC_TSSEC equals EMAC_ALMSEC and EMAC_TSSUBSEC equals EMAC_ALMSUBSEC.
[27:25]	**Reserved**	Reserved.
[24]	**TXBEIF**	**Transmit Bus Error Interrupt** The TXBEIF high indicates the memory controller replies ERROR response while EMAC access system memory through TXDMA during packet transmission process. Reset EMAC is recommended while TXBEIF status is high. If the TXBEIF is high and TXBEIEN (EMAC_INTEN[24]) is enabled, the TXIF will be high. Write 1 to this bit clears the TXBEIF status. 0 = No ERROR response is received. 1 = ERROR response is received.

[23]	**TDUIF**	**Transmit Descriptor Unavailable Interrupt** The TDUIF high indicates that there is no available TX descriptor for packet transmission and TXDMA will stay at Halt state. Once, the TXDMA enters the Halt state, S/W must issues a write command to TSDR register to make TXDMA leave Halt state while new TX descriptor is available. If the TDUIF is high and TDUIEN (EMAC_INTEN[23]) is enabled, the TXIF will be high. Write 1 to this bit clears the TDUIF status. 0 = TX descriptor is available. 1 = TX descriptor is unavailable.
[22]	**LCIF**	**Late Collision Interrupt** The LCIF high indicates the collision occurred in the outside of 64 bytes collision window. This means after the 64 bytes of a frame has been transmitted out to the network, the collision still occurred. The late collision check will only be done while EMAC is operating on half-duplex mode. If the LCIF is high and LCIEN (EMAC_INTEN[22]) is enabled, the TXIF will be high. Write 1 to this bit clears the LCIF status. 0 = No collision occurred in the outside of 64 bytes collision window. 1 = Collision occurred in the outside of 64 bytes collision window.
[21]	**TXABTIF**	**Transmit Abort Interrupt** The TXABTIF high indicates the packet incurred 16 consecutive collisions during transmission, and then the transmission process for this packet is aborted. The transmission abort is only available while EMAC is operating on half-duplex mode. If the TXABTIF is high and TXABTIEN (EMAC_INTEN[21]) is enabled, the TXIF will be high. Write 1 to this bit clears the TXABTIF status. 0 = Packet does not incur 16 consecutive collisions during transmission. 1 = Packet incurred 16 consecutive collisions during transmission.
[20]	**NCSIF**	**No Carrier Sense Interrupt** The NCSIF high indicates the MII I/F signal CRS does not active at the start of or during the packet transmission. The NCSIF is only available while EMAC is operating on half-duplex mode. If the NCSIF is high and NCSIEN (EMAC_INTEN[20]) is enabled, the TXIF will be high. Write 1 to this bit clears the NCSIF status. 0 = CRS signal actives correctly. 1 = CRS signal does not active at the start of or during the packet transmission.
[19]	**EXDEFIF**	**Defer Exceed Interrupt** The EXDEFIF high indicates the frame waiting for transmission has deferred over 0.32768ms on 100Mbps mode, or 3.2768ms on 10Mbps mode. The deferral exceed check will only be done while bit NODEF of MCMDR is disabled, and EMAC is operating on half-duplex mode. If the EXDEFIF is high and EXDEFIEN (EMAC_INTEN[19]) is enabled, the TXIF will be high. Write 1 to this bit clears the EXDEFIF status. 0 = Frame waiting for transmission has not deferred over 0.32768ms (100Mbps) or 3.2768ms (10Mbps). 1 = Frame waiting for transmission has deferred over 0.32768ms (100Mbps) or 3.2768ms (10Mbps).
[18]	**TXCPIF**	**Transmit Completion Interrupt** The TXCPIF indicates the packet transmission has completed correctly. If the TXCPIF is high and TXCPIEN (EMAC_INTEN[18]) is enabled, the TXIF will be high. Write 1 to this bit clears the TXCPIF status. 0 = The packet transmission not completed. 1 = The packet transmission has completed.

[17]	**TXUDIF**	**Transmit FIFO Underflow Interrupt** The TXUDIF high indicates the TXFIFO underflow occurred during packet transmission. While the TXFIFO underflow occurred, the EMAC will retransmit the packet automatically without S/W intervention. If the TXFIFO underflow occurred often, it is recommended that modify TXFIFO threshold control, the TXFIFOTH of FFTCR register, to higher level. If the TXUDIF is high and TXUDIEN (EMAC_INTEN[17]) is enabled, the TXIF will be high. Write 1 to this bit clears the TXUDIF status. 0 = No TXFIFO underflow occurred during packet transmission. 1 = TXFIFO underflow occurred during packet transmission.
[16]	**TXIF**	**Transmit Interrupt** The TXIF indicates the TX interrupt status. If TXIF high and its corresponding enable bit, TXIEN (EMAC_INTEN[16]), is also high indicates the EMAC generates TX interrupt to CPU. If TXIF is high but TXIEN (EMAC_INTEN[16]) is disabled, no TX interrupt is generated. The TXIF is logic OR result of bit logic AND result of EMAC_INTSTS[28:17] and EMAC_INTEN[28:17]. In other words, if any bit of EMAC_INTSTS[28:17] is high and its corresponding enable bit in EMAC_INTEN[28:17] is also enabled, the TXIF will be high. Because the TXIF is a logic OR result, clears EMAC_INTSTS[28:17] makes TXIF be cleared, too. 0 = No status bit in EMAC_INTSTS[28:17] is set or no enable bit in EMAC_INTEN[28:17] is enabled. 1 = At least one status in EMAC_INTSTS[28:17] is set and its corresponding enable bit in EMAC_INTEN[28:17] is enabled, too.
[15]	**WOLIF**	**Wake On LAN Interrupt Flag** The WOLIF high indicates EMAC receives a Magic Packet. The CFRIF only available while system is in power down mode and WOLEN is set high. If the WOLIF is high and WOLIEN (EMAC_INTEN[15]) is enabled, the RXIF will be high. Write 1 to this bit clears the WOLIF status. 0 = The EMAC does not receive the Magic Packet. 1 = The EMAC receives a Magic Packet.
[14]	**CFRIF**	**Control Frame Receive Interrupt** The CFRIF high indicates EMAC receives a flow control frame. The CFRIF only available while EMAC is operating on full duplex mode. If the CFRIF is high and CFRIEN (EMAC_INTEN[14]) is enabled, the RXIF will be high. Write 1 to this bit clears the CFRIF status. 0 = The EMAC does not receive the flow control frame. 1 = The EMAC receives a flow control frame.
[13:12]	**Reserved**	Reserved.
[11]	**RXBEIF**	**Receive Bus Error Interrupt** The RXBEIF high indicates the memory controller replies ERROR response while EMAC access system memory through RXDMA during packet reception process. Reset EMAC is recommended while RXBEIF status is high. If the RXBEIF is high and RXBEIEN (EMAC_INTEN[11]) is enabled, the RXIF will be high. Write 1 to this bit clears the RXBEIF status. 0 = No ERROR response is received. 1 = ERROR response is received.

[10]	**RDUIF**	**Receive Descriptor Unavailable Interrupt** The RDUIF high indicates that there is no available RX descriptor for packet reception and RXDMA will stay at Halt state. Once, the RXDMA enters the Halt state, S/W must issues a write command to RSDR register to make RXDMA leave Halt state while new RX descriptor is available. If the RDUIF is high and RDUIEN (EMAC_INTEN[10]) is enabled, the RXIF will be high. Write 1 to this bit clears the RDUIF status. 0 = RX descriptor is available. 1 = RX descriptor is unavailable.
[9]	**DENIF**	**DMA Early Notification Interrupt** The DENIF high indicates the EMAC has received the LENGTH field of the incoming packet. If the DENIF is high and DENIENI (EMAC_INTEN[9]) is enabled, the RXIF will be high. Write 1 to this bit clears the DENIF status. 0 = The LENGTH field of incoming packet has not received yet. 1 = The LENGTH field of incoming packet has received.
[8]	**MFLEIF**	**Maximum Frame Length Exceed Interrupt Flag** The MFLEIF high indicates the length of the incoming packet has exceeded the length limitation configured in DMARFC register and the incoming packet is dropped. If the MFLEIF is high and MFLEIEN (EMAC_INTEN[8]) is enabled, the RXIF will be high. Write 1 to this bit clears the MFLEIF status. 0 = The length of the incoming packet does not exceed the length limitation configured in DMARFC. 1 = The length of the incoming packet has exceeded the length limitation configured in DMARFC.
[7]	**MPCOVIF**	**Missed Packet Counter Overrun Interrupt Flag** The MPCOVIF high indicates the MPCNT, Missed Packet Count, has overflow. If the MPCOVIF is high and MPCOVIEN (EMAC_INTEN[7]) is enabled, the RXIF will be high. Write 1 to this bit clears the MPCOVIF status. 0 = The MPCNT has not rolled over yet. 1 = The MPCNT has rolled over yet.
[6]	**RPIF**	**Runt Packet Interrupt** The RPIF high indicates the length of the incoming packet is less than 64 bytes and the packet is dropped. If the ARP (EMAC_CTL[2]) is set, the short packet is regarded as a good packet and RPIF will not be set. If the RPIF is high and RPIEN (EMAC_INTEN[6]) is enabled, the RXIF will be high. Write 1 to this bit clears the RPIF status. 0 = The incoming frame is not a short frame or S/W wants to receive a short frame. 1 = The incoming frame is a short frame and dropped.
[5]	**ALIEIF**	**Alignment Error Interrupt** The ALIEIF high indicates the length of the incoming frame is not a multiple of byte. If the ALIEIF is high and ALIEIEN (EMAC_INTEN[5]) is enabled, the RXIF will be high. Write 1 to this bit clears the ALIEIF status. 0 = The frame length is a multiple of byte. 1 = The frame length is not a multiple of byte.
[4]	**RXGDIF**	**Receive Good Interrupt** The RXGDIF high indicates the frame reception has completed. If the RXGDIF is high and RXGDIEN (EAMC_MIEN[4]) is enabled, the RXIF will be high. Write 1 to this bit clears the RXGDIF status. 0 = The frame reception has not complete yet. 1 = The frame reception has completed.

[3]	**LPIF**	**Long Packet Interrupt Flag** The LPIF high indicates the length of the incoming packet is greater than 1518 bytes and the incoming packet is dropped. If the ALP (EMAC_CTL[1]) is set, the long packet will be regarded as a good packet and LPIF will not be set. If the LPIF is high and LPIEN (EMAC_INTEN[3]) is enabled, the RXIF will be high. Write 1 to this bit clears the LPIF status. 0 = The incoming frame is not a long frame or S/W wants to receive a long frame. 1 = The incoming frame is a long frame and dropped.
[2]	**RXOVIF**	**Receive FIFO Overflow Interrupt** The RXOVIF high indicates the RXFIFO overflow occurred during packet reception. While the RXFIFO overflow occurred, the EMAC drops the current receiving packer. If the RXFIFO overflow occurred often, it is recommended that modify RXFIFO threshold control, the RXFIFOTH of FFTCR register, to higher level. If the RXOVIF is high and RXOVIEN (EMAC_INTEN[2]) is enabled, the RXIF will be high. Write 1 to this bit clears the RXOVIF status. 0 = No RXFIFO overflow occurred during packet reception. 1 = RXFIFO overflow occurred during packet reception.
[1]	**CRCEIF**	**CRC Error Interrupt** The CRCEIF high indicates the incoming packet incurred the CRC error and the packet is dropped. If the AEP (EMAC_CTL[4]) is set, the CRC error packet will be regarded as a good packet and CRCEIF will not be set. If the CRCEIF is high and CRCEIEN (EMAC_INTEN[1]) is enabled, the RXIF will be high. Write 1 to this bit clears the CRCEIF status. 0 = The frame does not incur CRC error. 1 = The frame incurred CRC error.
[0]	**RXIF**	**Receive Interrupt** The RXIF indicates the RX interrupt status. If RXIF high and its corresponding enable bit, RXIEN (EMAC_INTEN[0]), is also high indicates the EMAC generates RX interrupt to CPU. If RXIF is high but RXIEN (EMAC_INTEN[0]) is disabled, no RX interrupt is generated. The RXIF is logic OR result of bit logic AND result of EMAC_INTSTS[15:1] and EMAC_INTEN[15:1]. In other words, if any bit of EMAC_INTSTS[15:1] is high and its corresponding enable bit in EMAC_INTEN[15:1] is also enabled, the RXIF will be high. Because the RXIF is a logic OR result, clears EMAC_INTSTS[15:1] makes RXIF be cleared, too. 0 = No status bit in EMAC_INTSTS[15:1] is set or no enable bit in EMAC_INTEN[15:1] is enabled. 1 = At least one status in EMAC_INTSTS[15:1] is set and its corresponding enable bit in EMAC_INTEN[15:1] is enabled, too.

MAC General Status Register (EMAC_GENSTS)

The EMAC_GENSTS also keeps the statuses of EMAC. But the statuses in the EMAC_GENSTS will not trigger any interrupt. The EMAC_GENSTS is a write clear register and write 1 to corresponding bit clears the status.

Register	Offset	R/W	Description	Reset Value
EMAC_GENSTS	EMAC_BA+0x0B4	R/W	MAC General Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved			RPSTS	TXHALT	SQE	TXPAUSED	DEF
7	6	5	4	3	2	1	0
COLCNT				Reserved	RXFFULL	RXHALT	CFR

Bits	Description	
[31:13]	**Reserved**	Reserved.
[12]	**RPSTS**	**Remote Pause Status** The RPSTS indicates that remote pause counter down counting actives. After Ethernet MAC controller sent PAUSE frame out successfully, it starts the remote pause counter down counting. When this bit high, it's predictable that remote Ethernet MAC controller wouldn't start the packet transmission until the down counting done. 0 = Remote pause counter down counting done. 1 = Remote pause counter down counting actives.
[11]	**TXHALT**	**Transmission Halted** The TXHALT high indicates the next normal packet transmission process will be halted because the bit TXON (EMAC_CTL[8]) is disabled be S/W. 0 = Next normal packet transmission process will go on. 1 = Next normal packet transmission process will be halted.
[10]	**SQE**	**Signal Quality Error** The SQE high indicates the SQE error found at end of packet transmission on 10Mbps half-duplex mode. The SQE error check will only be done while both bit SQECHKEN (EMAC_CTL[17]) is enabled and EMAC is operating on 10Mbps half-duplex mode. 0 = No SQE error found at end of packet transmission. 1 = SQE error found at end of packet transmission.
[9]	**TXPAUSED**	**Transmission Paused** The TXPAUSED high indicates the next normal packet transmission process will be paused temporally because EMAC received a PAUSE control frame. 0 = Next normal packet transmission process will go on. 1 = Next normal packet transmission process will be paused.

[8]	**DEF**	**Deferred Transmission** The DEF high indicates the packet transmission has deferred once. The DEF is only available while EMAC is operating on half-duplex mode. 0 = Packet transmission does not defer. 1 = Packet transmission has deferred once.
[7:4]	**COLCNT**	**Collision Count** The COLCNT indicates that how many collisions occurred consecutively during a packet transmission. If the packet incurred 16 consecutive collisions during transmission, the COLCNT will be 4"h0 and bit TXABTIF will be set to 1.
[3]	**Reserved**	Reserved.
[2]	**RXFFULL**	**RXFIFO Full** The RXFFULL indicates the RXFIFO is full due to four 64-byte packets are kept in RXFIFO and the following incoming packet will be dropped. 0 = The RXFIFO is not full. 1 = The RXFIFO is full and the following incoming packet will be dropped.
[1]	**RXHALT**	**Receive Halted** The RXHALT high indicates the next normal packet reception process will be halted because the bit RXON of MCMDR is disabled be S/W. 0 = Next normal packet reception process will go on. 1 = Next normal packet reception process will be halted.
[0]	**CFR**	**Control Frame Received** The CFRIF high indicates EMAC receives a flow control frame. The CFRIF only available while EMAC is operating on full duplex mode. 0 = The EMAC does not receive the flow control frame. 1 = The EMAC receives a flow control frame.

Missed Packet Count Register (EMAC_MPCNT)

The EMAC_MPCNT keeps the number of packets that were dropped due to various types of receive errors. The EMAC_MPCNT is a read clear register. In addition, S/W also can write an initial value to EMAC_MPCNT and the missed packet counter will start counting from that initial value. If the missed packet counter is overflow, the MPCOVIF (EMAC_INTSTS[7]) will be set.

Register	Offset	R/W	Description	Reset Value
EMAC_MPCNT	EMAC_BA+0x0B8	R/W	Missed Packet Count Register	0x0000_7FFF

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
MPCNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MPCNT							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**MPCNT**	**Miss Packet Count** The MPCNT indicates the number of packets that were dropped due to various types of receive errors. The following type of receiving error makes missed packet counter increase: 1. Incoming packet is incurred RXFIFO overflow. 2. Incoming packet is dropped due to RXON is disabled. 3. Incoming packet is incurred CRC error.

MAC Receive Pause Count Register (EMAC_RPCNT)

The EMAC supports the PAUSE control frame reception and recognition. If EMAC received a PAUSE control frame, the OPERAND field of the PAUSE control frame will be extracted and stored in the EMAC_RPCNT register. The EMAC_RPCNT register will keep the same while TX of EMAC is pausing due to the PAUSE control frame is received. The EMAC_RPCNT is read only and write to this register has no effect.

Register	Offset	R/W	Description	Reset Value
EMAC_RPCNT	EMAC_BA+0x0BC	R	MAC Receive Pause Count Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RPCNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RPCNT							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**RPCNT**	**MAC Receive Pause Count** The RPCNT keeps the OPERAND field of the PAUSE control frame. It indicates how many slot time (512 bit time) the TX of EMAC will be paused.

DMA Receive Frame Status Register (EMAC_FRSTS)

The EMAC_FRSTS is used to keep the LENGTH field of each incoming Ethernet packet.

Register	Offset	R/W	Description	Reset Value
EMAC_FRSTS	EMAC_BA+0x0C8	R/W	DMA Receive Frame Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RXFLT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RXFLT							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**RXFLT**	**Receive Frame LENGTH** The RXFLT keeps the LENGTH field of each incoming Ethernet packet. If the bit DENIEN (EMAC_INTEN[9]) is enabled and the LENGTH field of incoming packet has received, the bit DENIF (EMAC_INTSTS[9]) will be set and trigger interrupt. And, the content of LENGTH field will be stored in RXFLT.

Current Transmit Descriptor Start Address Register (EMAC_CTXDSA)

Register	Offset	R/W	Description	Reset Value
EMAC_CTXDSA	EMAC_BA+0x0CC	R	Current Transmit Descriptor Start Address Register	0x0000_0000

31	30	29	28	27	26	25	24
CTXDSA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CTXDSA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CTXDSA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CTXDSA							

Bits	Description	
[31:0]	**CTXDSA**	**Current Transmit Descriptor Start Address** The CTXDSA keeps the start address of TX descriptor that is used by TXDMA currently. The CTXDSA is read only and write to this register has no effect.

Current Transmit Buffer Start Address Register (EMAC_CTXBSA)

Register	Offset	R/W	Description	Reset Value
EMAC_CTXBSA	EMAC_BA+0x0D0	R	Current Transmit Buffer Start Address Register	0x0000_0000

31	30	29	28	27	26	25	24
CTXBSA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CTXBSA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CTXBSA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CTXBSA							

Bits	Description	
[31:0]	**CTXBSA**	**Current Transmit Buffer Start Address** The CTXDSA keeps the start address of TX frame buffer that is used by TXDMA currently. The CTXBSA is read only and write to this register has no effect.

Current Receive Descriptor Start Address Register (EMAC_CRXDSA)

Register	Offset	R/W	Description	Reset Value
EMAC_CRXDSA	EMAC_BA+0x0D4	R	Current Receive Descriptor Start Address Register	0x0000_0000

31	30	29	28	27	26	25	24
CRXDSA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CRXDSA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CRXDSA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CRXDSA							

Bits	Description	
[31:0]	**CRXDSA**	**Current Receive Descriptor Start Address** The CRXDSA keeps the start address of RX descriptor that is used by RXDMA currently. The CRXDSA is read only and write to this register has no effect.

Current Receive Buffer Start Address Register (EMAC_CRXBSA)

Register	Offset	R/W	Description	Reset Value
EMAC_CRXBSA	EMAC_BA+0x0D8	R	Current Receive Buffer Start Address Register	0x0000_0000

31	30	29	28	27	26	25	24
CRXBSA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CRXBSA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CRXBSA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CRXBSA							

Bits	Description	
[31:0]	**CRXBSA**	**Current Receive Buffer Start Address** The CRXBSA keeps the start address of RX frame buffer that is used by RXDMA currently. The CRXBSA is read only and write to this register has no effect.

Time Stamp Control Register (EMAC_TSCTL)

Register	Offset	R/W	Description	Reset Value
EMAC_TSCTL	EMAC_BA+0x100	R/W	Time Stamp Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**TSALMEN**	**Reserved**	**TSUPDATE**	**TSMODE**	**TSIEN**	**TSEN**

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**TSALMEN**	**Time Stamp Alarm Enable Bit** Set this bit high enable Ethernet MAC controller to set TSALMIF (EMAC_INTSTS[28]) high when EMAC_TSSEC equals to EMAC_ALMSEC and EMAC_TSSUBSEC equals to EMAC_ALMSUBSEC. 0 = Alarm disabled when EMAC_TSSEC equals to EMAC_ALMSEC and EMAC_TSSUBSEC equals to EMAC_ALMSUBSEC. 1 = Alarm enabled when EMAC_TSSEC equals to EMAC_ALMSEC and EMAC_TSSUBSEC equals to EMAC_ALMSUBSEC.
[4]	**Reserved**	Reserved.
[3]	**TSUPDATE**	**Time Stamp Counter Time Update Enable Bit** Set this bit high enables Ethernet MAC controller to add value of register EMAC_UPDSEC and EMAC_UPDSUBSEC to PTP time stamp counter. After the add operation finished, Ethernet MAC controller clear this bit to low automatically. 0 = No action. 1 = EMAC_UPDSEC updated to EMAC_TSSEC and EMAC_UPDSUBSEC updated to EMAC_TSSUBSEC.
[2]	**TSMODE**	**Time Stamp Fine Update Enable Bit** This bit chooses the time stamp counter update mode. 0 = Time stamp counter is in coarse update mode. 1 = Time stamp counter is in fine update mode.

[1]	**TSIEN**	**Time Stamp Counter Initialization Enable Bit** Set this bit high enables Ethernet MAC controller to load value of register EMAC_UPDSEC and EMAC_UPDSUBSEC to PTP time stampe counter. After the load operation finished, Ethernet MAC controller clear this bit to low automatically. 0 = Time stamp counter initialization done. 1 = Time stamp counter initialization Enabled.
[0]	**TSEN**	**Time Stamp Function Enable Bit** This bit controls if the IEEE 1588 PTP time stamp function is enabled or not. Set this bit high to enable IEEE 1588 PTP time stamp function while set this bit low to disable IEEE 1588 PTP time stamp function. 0 = I EEE 1588 PTP time stamp function Disabled. 1 = IEEE 1588 PTP time stamp function Enabled.

Time Stamp Counter Second Register (EMAC_TSSEC)

Register	Offset	R/W	Description	Reset Value
EMAC_TSSEC	EMAC_BA+0x110	R	Time Stamp Counter Second Register	0x0000_0000

31	30	29	28	27	26	25	24
SEC							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SEC							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SEC							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SEC							

Bits	Description	
[31:0]	**SEC**	**Time Stamp Counter Second** This register reflects the bit [63:32] value of 64-bit reference timing counter. This 32-bit value is used as the second part of time stamp when TSEN (EMAC_TSCTL[0]) is high.

Time Stamp Counter Sub Second Register (EMAC_TSSUBSEC)

Register	Offset	R/W	Description	Reset Value
EMAC_TSSUBSEC	EMAC_BA+0x114	R	Time Stamp Counter Sub Second Register	0x0000_0000

31	30	29	28	27	26	25	24
SUBSEC							
23	22	21	20	19	18	17	16
SUBSEC							
15	14	13	12	11	10	9	8
SUBSEC							
7	6	5	4	3	2	1	0
SUBSEC							

Bits	Description	
[31:0]	SUBSEC	**Time Stamp Counter Sub-Second** This register reflects the bit [31:0] value of 64-bit reference timing counter. This 32-bit value is used as the sub-second part of time stamp when TSEN (EMAC_TSCTL[0]) is high.

Time Stamp Increment Register (EMAC_TSINC)

Register	Offset	R/W	Description	Reset Value
EMAC_TSINC	EMAC_BA+0x118	R/W	Time Stamp Increment Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNTINC							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**CNTINC**	**Time Stamp Counter Increment** Time stamp counter increment value. If TSEN (EMAC_TSCTL[0]) is high, EMAC adds EMAC_TSSUBSEC with this 8-bit value every time when it wants to increase the EMAC_TSSUBSEC value.

Time Stamp Addend Register (EMAC_TSADDEND)

Register	Offset	R/W	Description	Reset Value
EMAC_TSADDEND	EMAC_BA+0x11C	R/W	Time Stamp Addend Register	0x0000_0000

31	30	29	28	27	26	25	24
ADDEND							
23	22	21	20	19	18	17	16
ADDEND							
15	14	13	12	11	10	9	8
ADDEND							
7	6	5	4	3	2	1	0
ADDEND							

Bits	Description	
[31:0]	ADDEND	**Time Stamp Counter Addend** This register keeps a 32-bit value for accumulator to enable increment of EMAC_TSSUBSEC. If TSEN (EMAC_TSCTL[0]) and TSMODE (EMAC_TSCTL[2]) are both high, EMAC increases accumulator with this 32-bit value in each HCLK. Once the accumulator is overflow, it generates a enable to increase EMAC_TSSUBSEC with an 8-bit value kept in register EMAC_TSINC.

Time Stamp Update Second Register (EMAC_UPDSEC)

Register	Offset	R/W	Description	Reset Value
EMAC_UPDSEC	EMAC_BA+0x120	R/W	Time Stamp Update Second Register	0x0000_0000

31	30	29	28	27	26	25	24
SEC							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SEC							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SEC							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SEC							

Bits	Description	
[31:0]	**SEC**	**Time Stamp Counter Second Update** When TSIEN (EMAC_TSCTL[1]) is high. EMAC loads this 32-bit value to EMAC_TSSEC directly. When TSUPDATE (EMAC_TSCTL[3]) is high, EMAC increases EMAC_TSSEC with this 32-bit value.

Time Stamp Update Sub Second Register (EMAC_UPDSUBSEC)

Register	Offset	R/W	Description	Reset Value
EMAC_UPDSUBSEC	EMAC_BA+0x124	R/W	Time Stamp Update Sub Second Register	0x0000_0000

31	30	29	28	27	26	25	24
SUBSEC							
23	22	21	20	19	18	17	16
SUBSEC							
15	14	13	12	11	10	9	8
SUBSEC							
7	6	5	4	3	2	1	0
SUBSEC							

Bits	Description	
[31:0]	SUBSEC	**Time Stamp Counter Sub-Second Update** When TSIEN (EMAC_TSCTL[1]) is high. EMAC loads this 32-bit value to EMAC_TSSUBSEC directly. When TSUPDATE (EMAC_TSCTL[3]) is high, EMAC increases EMAC_TSSUBSEC with this 32-bit value.

Time Stamp Alarm Second Register (EMAC_ALMSEC)

Register	Offset	R/W	Description	Reset Value
EMAC_ALMSEC	EMAC_BA+0x128	R/W	Time Stamp Alarm Second Register	0x0000_0000

31	30	29	28	27	26	25	24
SEC							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SEC							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SEC							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SEC							

Bits	Description	
[31:0]	**SEC**	**Time Stamp Counter Second Alarm** Time stamp counter second part alarm value. This value is only useful when ALMEN (EMAC_TSCTL[5]) high. If ALMEN (EMAC_TSCTL[5]) is high, EMAC_TSSEC equals to EMAC_ALMSEC and EMAC_TSSUBSEC equals to EMAC_ALMSUBSEC, Ethernet MAC controller set TSALMIF (EMAC_INTSTS[28]) high.

Time Stamp Alarm Sub Second Register (EMAC_ALMSUBSEC)

Register	Offset	R/W	Description	Reset Value
EMAC_ALMSUBSEC	EMAC_BA+0x12C	R/W	Time Stamp Alarm Sub Second Register	0x0000_0000

31	30	29	28	27	26	25	24
SUBSEC							
23	22	21	20	19	18	17	16
SUBSEC							
15	14	13	12	11	10	9	8
SUBSEC							
7	6	5	4	3	2	1	0
SUBSEC							

Bits	Description	
[31:0]	SUBSEC	**Time Stamp Counter Sub-Second Alarm** Time stamp counter sub-second part alarm value. This value is only useful when ALMEN (EMAC_TSCTL[5]) high. If ALMEN (EMAC_TSCTL[5]) is high, EMAC_TSSEC equals to EMAC_ALMSEC and EMAC_TSSUBSEC equals to EMAC_ALMSUBSEC, Ethernet MAC controller set TSALMIF (EMAC_INTSTS[28]) high.

6.13 Flash Memory Controller (FMC)

6.13.1 Overview

The NUC442/NUC472 is equipped with 256/512 Kbytes on-chip embedded flash for application program memory (APROM) and data flash that can be updated through ISP procedure. In-System-Programming (ISP) and In-Application-Programming (IAP) enables user to update chip embedded flash when chip is soldered on PCB. After chip is powered on, Cortex®-M4 CPU fetches code from APROM or LDROM decided by boot select (CBS) in Config0. By the way, the NUC442/NUC472 also provides Data Flash for user to store some application dependent data before chip is powered off.

The NUC442/NUC472 supports another flexible feature: configurable data flash size. The data flash size is decided by data flash enable (DFEN) in Config0 and data flash base address (DFBADR) in Config1. When DFEN is set to 1, the data flash size is zero. When DFEN is set to 0, the APROM and data flash share 256/512 Kbytes continuous address and the start address of data flash is defined by (DFBADR) in Config1.

6.13.2 Features

- Runs up to 84 MHz with zero wait state for continuous address read access
- 256/512 Kbytes application program memory (APROM) and data flash
- 16 Kbytes in system programming (ISP) loader program memory (LDROM)
- Configurable Data flash size with 2 Kbytes page erase unit
- Flash write protect size with 16 Kbytes per block unit
- User Configuration memory with CRC checking
- In System Program (ISP) /In Application Program (IAP) to update on chip Flash

6.13.3 Block Diagram

The flash memory controller consists of AHB slave interface, ISP control logic and flash macro interface timing control logic. The block diagram of flash memory controller is shown as follows.

6.13.4 Flash Memory Organization

The flash memory consists of application program memory (APROM), Data Flash, ISP loader program memory (LDROM), user configuration and user hidden block. User configuration block provides four words to control system logic, such as flash security lock, boot select, brown-out voltage level, Data Flash base address and configuration checksum. It works like a fuse for power on setting. It is loaded from flash memory to its corresponding control registers during chip powered on. User can set these bits according to different application requests. The Data Flash start address and its size can be defined by user depending on the application.

Block Name	Size	Start Address	End Address
APROM	(256-2*N) KB ((512-2*N) KB	0x0000_0000	0x0003_FFFF (256KB, if DFEN=1) 0x0007_FFFF (512KB, if DFEN=1) DFBADR-1 (if DFEN=0)
Data Flash	2*N KB (if DFEN=0)	DFBADR (if DFEN=0)	0x0003_FFFF (256KB) 0x0007_FFFF (512KB)
LDROM	16 KB	0x0010_0000	0x0010_3FFF
User Configuration	4 words	0x0030_0000	0x0030_000C

Note: N is the page number of configured data flash. One page size is 2048 bytes, , N >= 0

Table 6.13-1 Memory Address Map

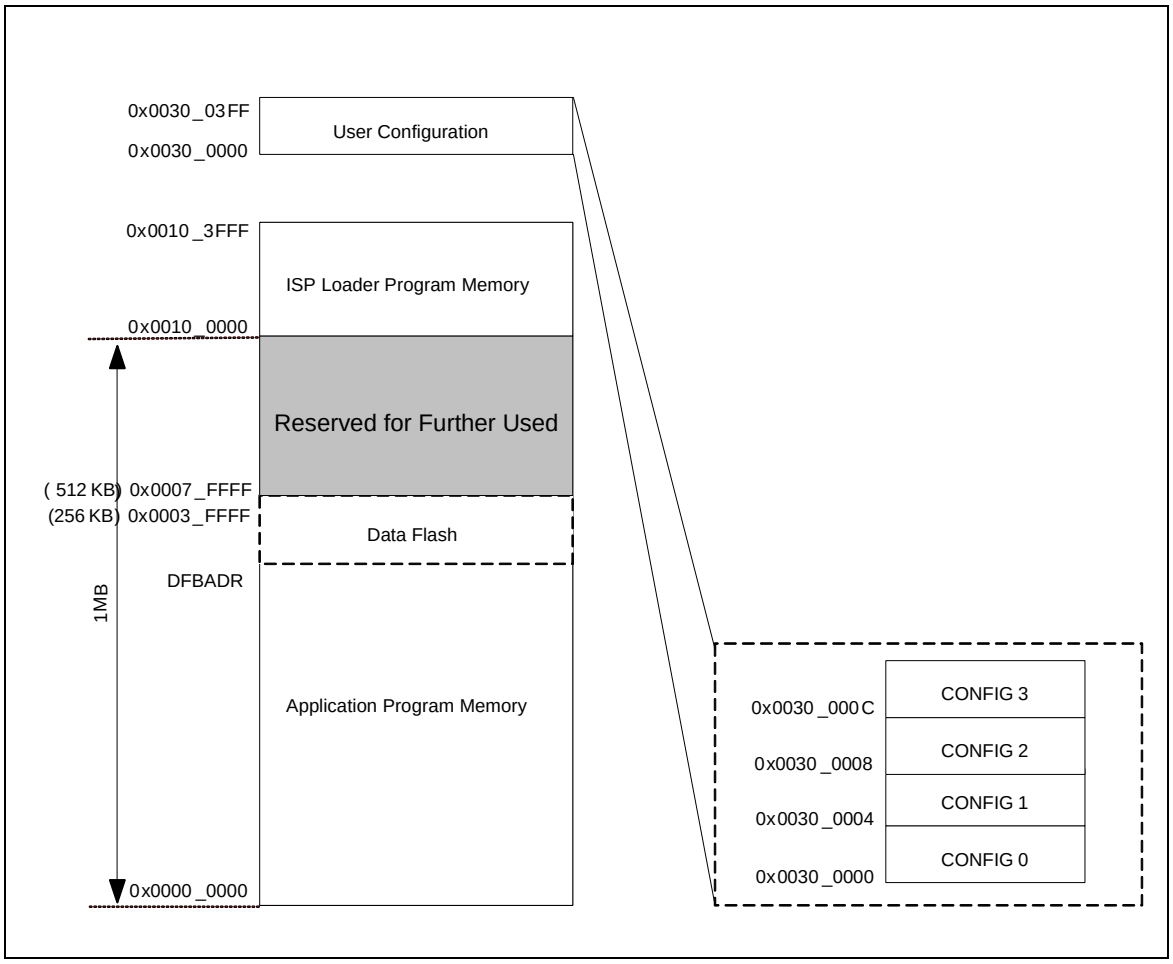


Figure 6.13-1 Flash Memory Organization

6.13.5 Boot Selection

The NUC442/NUC472 provides In System Programming (ISP) feature for user to update application program memory (APROM) when chip is soldered on PCB. A dedicated 16 KB ISP loader program memory (LDROM) is used to store ISP firmware. User can select that CPU fetches code from APROM or LDROM by boot select (CBS) in Config0.

CBS[1:0]	Boot Selection
00	**LDROM with IAP function** Chip booting from LDROM, program executing range including LDROM and most of APROM (all except first 2048 bytes as the first 2048 bytes is mapped from LDROM). LDROM address is mapping to 0x0010_0000 ~ 0x0010_3FFF, and also the first 2048 bytes of LDROM is mapping to the address 0x0000_0000 ~ 0x0000_07FF. The address 0x0000_0000 ~ 0x0000_07FF can be re-mapped to any other page within executing range through ISP command. Both APROM and LDROM are programmable in this mode no matter the code is currently running on LDROM or APROM. Data Flash is meaningless in this mode, because any area of APROM and LDROM can just be used as the Data Flash. DFBADR is not functioned in this mode.
01	**LDROM without IAP function** Chip booting from LDROM, program executing range only including LDROM; APROM cannot be access by program directly, except by through ISP. LDROM is write-protected in this mode.
10	**APROM with IAP function** Chip booting from APROM, program executing range including LDROM and APROM LDROM address is mapping to 0x0010_0000~0x0010_3FFF The address 0x0000_0000 ~ 0x0000_07FF can be re-mapped to any other page within executing range though ISP command. Both APROM and LDROM are programmable in this mode no matter the code is currently running on LDROM or APROM. Data Flash is meaningless in this mode, because any area of APROM and LDROM can just be used as the Data Flash. DFBADR is not functioned in this mode.
11	**APROM without IAP function** Chip booting from APROM and program executing range only including APROM. LDROM cannot be access by program directly, except by through ISP. APROM is write-protected in this mode.

Table 6.13-2 NUC442/NUC472 Boot Selection

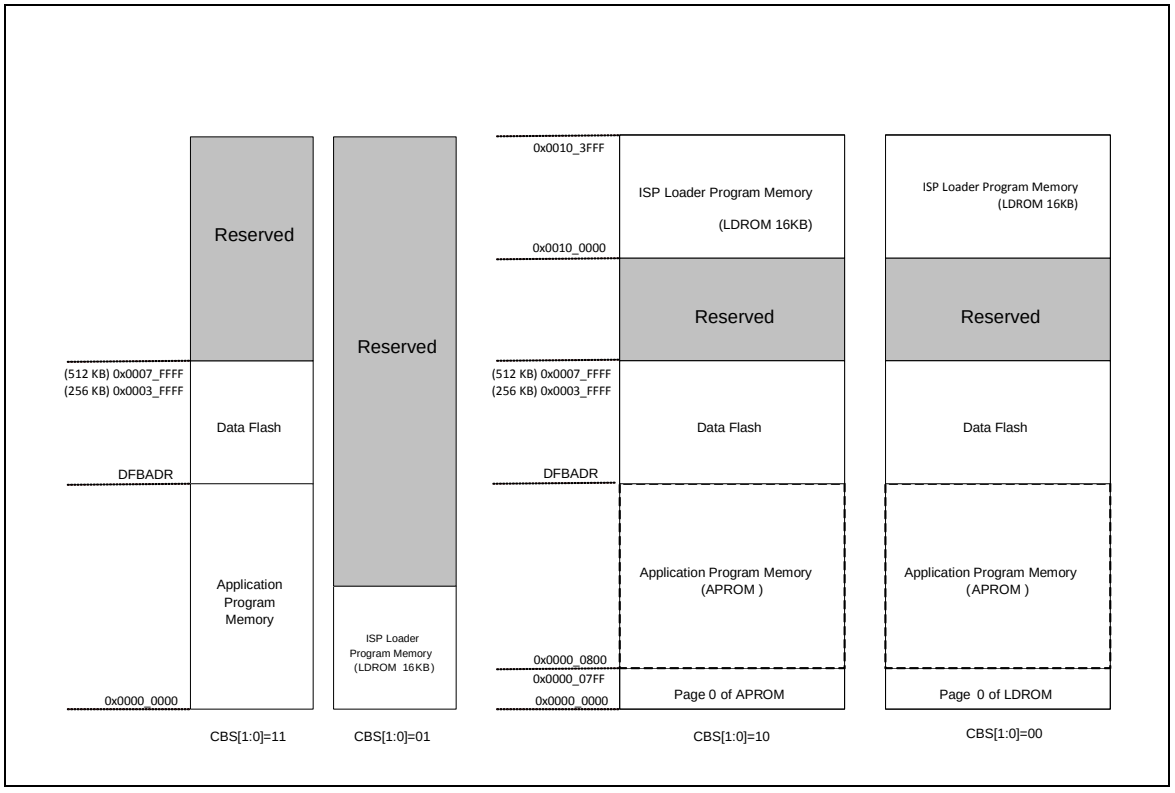


Figure 6.13-2 NUC442/NUC472 Boot Selection

6.13.6 In Application Programming

The NUC442/NUC472 series provides a new In-application-programming (IAP) function for user to switch the code executing between APROM and LDROM without a reset. User can enable the IAP function by re-booting chip and setting the chip boot selection bits in Config0 (CBS[1:0]) as 10b or 00b.

In the case that NUC442/NUC472 boots from APROM with the IAP function enabled (CBS[1:0] = 10b), the executable range of code includes all of APROM and LDROM. The address space of APROM is kept as the original size but the address space of the 16 KB LDROM is mapped to 0x0010_0000~ 0x0010_3FFF.

In the case that NUC442/NUC472 boots from LDROM with the IAP function enabled (CBS[1:0] = 00b), the executable range of code includes all of LDROM and almost all of APROM except for its first page. User cannot access the first page of APROM because the first page of executable code range becomes the mirror of the first page of LDROM as set by default. Meanwhile, the address space of 16 KB LDROM is mapped to 0x0010_0000~0x0010_3FFF.

Please refer to the following figure for the address map while IAP is activating.

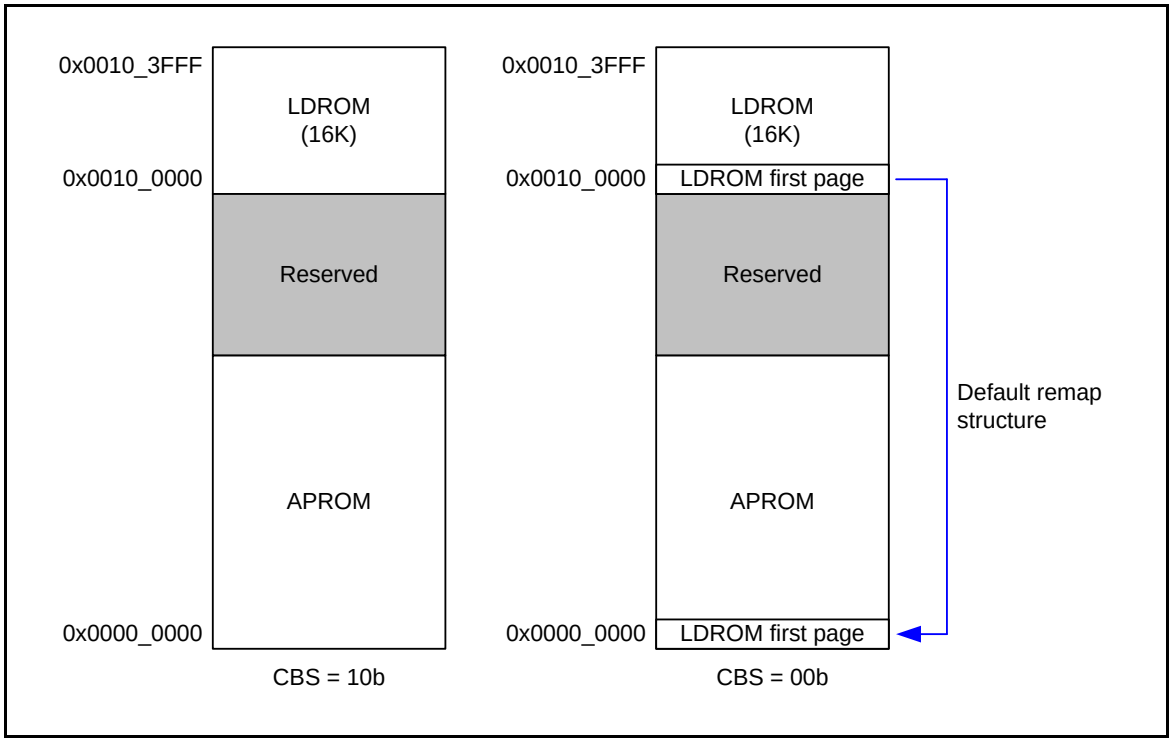


Figure 6.13-3 Executable Range of Code with IAP Function Enabled

When chip boots with the IAP function enabled, any other page within the executable range of code can be mirrored to the first page of executable code (0x0000_0000~0x0000_07FF) any time. User can change the remap address of the first executing page by filling the target remap address to FMC_ISPADDR and then go through ISP procedure with the Vector Page Re-map command. After changing the remap address, user can check if the change is successful by reading the VECMAP(FMC_ISPSTS[20:9]).

6.13.7 Data Flash

The NUC442/NUC472 provides data flash for user to store data, which is read or written through ISP procedure. The size of each erase unit is 2 Kbytes. When a word will be changed, all 512 words need to be copied to another page or SRAM in advance.

The application program memory (APROM) and the data flash are shared with APROM with variable size defined by user. If Data Flash enable (DFEN) bit in Config0 is 1, there is no Data Flash and all 256/512 Kbytes size is used for APROM. If Data Flash enable (DFEN) bit in Config0 is 0, the Fata Flash share with APROM and its base address is defined by Data Flash base address (DFBADR) bits in Config1. Under this setting, the application program memory size is ((256/512)-2*N) KB and data flash size is 2*N KB.

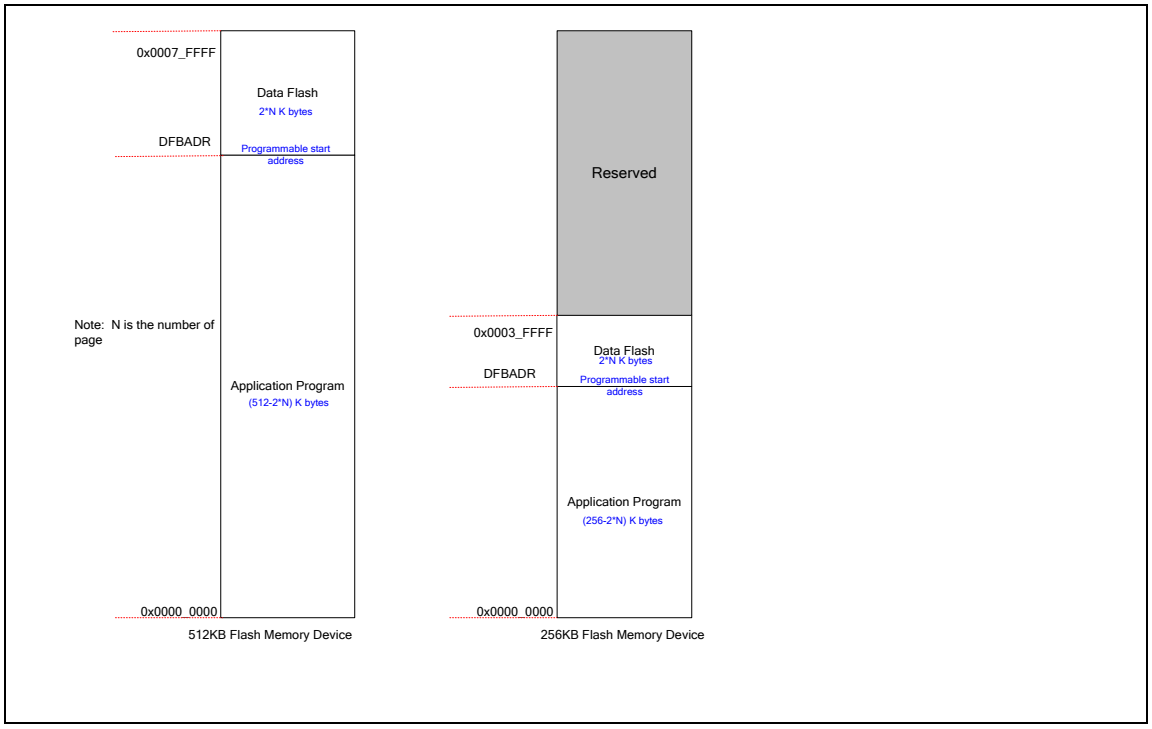


Figure 6.13-4 Flash Memory Structure

6.13.8 User Configuration

Config0 (Address = 0x0030_0000)

31	30	29	28	27	26	25	24
CWDTEN	CWDTPDEN	Reserved	CKF	CFGXT1	CFOSC		
23	22	21	20	19	18	17	16
CBODEN	CBOV1	CBOV0	CBORST	Reserved			
15	14	13	12	11	10	9	8
RMII	CFG32K	Reserved		LDWPEN	CIOINI	Reserved	
7	6	5	4	3	2	1	0
CBS		Reserved				LOCK	DFEN

Bits	Field	Description
[31]	**CWDTEN**	**Watchdog hardware Enable Bit** 0 = Window Watchdog Timer Enabled when chip is powered on. 1 = Window Watchdog Timer Disabled when chip is powered on.
[30]	**CWDTPDEN**	**Watchdog Clock Power-down Enable Bit** 0 = Watchdog Timer clock kept enabled when chip enters Power-down. 1 = Watchdog Timer clock is controlled by LIRCEN (CLK_PWRCTL[3]) when chip enters Power-down. **Note:** This bit only works if CWDTEN is set to 0
[29]	**Reserved**	Reserved.
[28]	**CKF**	**XT1 Clock Filter Enable Bit** 0 = XT1 clock filter Disabled. 1 = XT1 clock filter Enabled.
[27]	**CFGXT1**	**GPG[13:12] Multi-Function Selection** 0 = GPG[13:12] work as GPIO. 1 = GPG[13:12] work as external 4~24 MHz crystal pin,. GPG[13]: XOUT, GPG[12]: XIN
[26:24]	**CFOSC**	**CPU Clock Source Selection After Reset** The value of CFOSC will be loaded to HCLKSEL (CLK_CLKSEL0[2:0]) in system register after any reset occurs. 000 = External 4~24 MHz high-speed crystal clock. 111 = Internal RC 22.1184 MHz high-speed oscillator clock Others = Reserved.
[23]	**CBODEN**	**Brown-out Detector Enable Bit** 0= Brown-out detect Enabled after powered on. 1= Brown-out detect Disabled after powered on.

[22:21]	**CBOV1-0**	**Brown-out Voltage Selection** 00 = 2.2V 01 = 2.7V 10 = 3.7V 11 = 4.5V
[20]	**CBORST**	**Brown-out Reset Enable Bit** 0 = Brown-out reset Enabled after powered on. 1 = Brown-out reset Disabled after powered on.
[19:16]	**Reserved**	Reserved.
[15]	**RMII**	**EMAC Interface Selection** 0 = MII mode Enabled. 1 = RMII mode Enabled.
[14]	**CFG32K**	**GPG[15:14] Multi-Function Selection** 0 = GPG[15:14] work as GPIO. 1 = GPG[15:14] work as external 32 kHz crystal pin,. GPG[15]: X32_O, GPG[14]: X32_I
[13:12]	**Reserved**	Reserved.
[11]	**LDWPEN**	**LDROM Write-Protected Enable Bit** 0 = LDROM (16KB) write-protect Enabled,. 1 = LDROM (16KB) write-protect Disabled.
[10]	**CIOINI**	**I/O Initial State Selection** 0 = All GPIO set as Quasi-bidirectional mode after chip powered on. 1 = All GPIO set as input tri-state mode after powered on.
[9:8]	**Reserved**	Reserved.
[7:6]	**CBS**	**Chip Boot Selection** 00 = LDROM with IAP function. 01 = LDROM without IAP function. 10 = APROM with IAP function. 11 = APROM without IAP function. User can set CBS[0] = 0 to support IAP function. When CBS[0] = 0, the LDROM is mapping to address 0x100000 and APROM is mapping to address 0x0. User could access them by their address without boot switching. In other words, if IAP function is supported, the code in LDROM and APROM can be called by each other. **Note1:** BS bit of ISPCON is only can be used to control boot switching when CBS[0] = 1. **Note2:** VECMAP is only can be used to remap page 0 of APROM or LDROM to 0x0~0x7FF when CBS[0] = 0.
[5:2]	**Reserved**	Reserved.
[1]	**LOCK**	**Lock Mode** 0 = Lock Mode. When flash data is locked, only device ID, Config0 and Config1 can be read by writer and ICP through serial debug interface. Others data is locked as 0xFFFFFFFF. ISP can read data anywhere 1 = Normal Mode. Flash data is not locked.
[0]	**DFEN**	**Data Flash Enable Bit** 0 = Data flash Enabled. 1 = Data flash Disabled.

Config1 (Address = 0x0030_0004)

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				DFBADR.19	DFBADR.18	DFBADR.17	DFBADR.16
15	14	13	12	11	10	9	8
DFBADR.15	DFBADR.14	DFBADR.13	DFBADR.12	DFBADR.11	DFBADR.10	DFBADR.9	DFBADR.8
7	6	5	4	3	2	1	0
DFBADR.7	DFBADR.6	DFBADR.5	DFBADR.4	DFBADR.3	DFBADR.2	DFBADR.1	DFBADR.0

Bits	Field	Description
[31:20]	**Reserved**	Reserved.
[19:0]	**DFBADR**	**Data Flash Base Address (This register works only when DFEN set to 0)** If DFEN is set to 0, the data flash base address is defined by user. Since on-chip flash erase unit is 2 Kbytes, it is mandatory to keep bit 10~0 as 0.

Config2 (Address = 0x0030_0008)

31	30	29	28	27	26	25	24
BWP31	BWP30	BWP29	BWP28	BWP27	BWP26	BWP25	BWP24
23	22	21	20	19	18	17	16
BWP23	BWP22	BWP21	BWP20	BWP19	BWP18	BWP17	BWP16
15	14	13	12	11	10	9	8
BWP15	BWP14	BWP13	BWP12	BWP11	BWP10	BWP9	BWP8
7	6	5	4	3	2	1	0
BWP7	BWP6	BWP5	BWP4	BWP3	BWP2	BWP1	BWP0

Bits	Field	Description
[31:0]	BWP	**Flash Block Write Protect** If BWP.N bit is set to 0, ISP cannot program and erase the APROM flash relative region. (N=0~31).

Bit	Flash Protect Region	Bit	Flash Protect Region
BWP0	0x00_0000 ~ 0x00_3FFF	BWP16	0x04_0000 ~ 0x04_3FFF
BWP1	0x00_4000 ~ 0x00_7FFF	BWP17	0x04_4000 ~ 0x04_7FFF
BWP2	0x00_8000 ~ 0x00_BFFF	BWP18	0x04_8000 ~ 0x04_BFFF
BWP3	0x00_C000 ~ 0x00_FFFF	BWP19	0x04_C000 ~ 0x04_FFFF
BWP4	0x01_0000 ~ 0x01_3FFF	BWP20	0x05_0000 ~ 0x05_3FFF
BWP5	0x01_4000 ~ 0x01_7FFF	BWP21	0x05_4000 ~ 0x05_7FFF
BWP6	0x01_8000 ~ 0x01_BFFF	BWP22	0x05_8000 ~ 0x05_BFFF
BWP7	0x01_C000 ~ 0x01_FFFF	BWP23	0x05_C000 ~ 0x05_FFFF
BWP8	0x02_0000 ~ 0x02_3FFF	BWP24	0x06_0000 ~ 0x06_3FFF
BWP9	0x02_4000 ~ 0x02_7FFF	BWP25	0x06_4000 ~ 0x06_7FFF
BWP10	0x02_8000 ~ 0x02_BFFF	BWP26	0x06_8000 ~ 0x06_BFFF
BWP11	0x02_C000 ~ 0x02_FFFF	BWP27	0x06_C000 ~ 0x06_FFFF
BWP12	0x03_0000 ~ 0x03_3FFF	BWP28	0x07_0000 ~ 0x07_3FFF
BWP13	0x03_4000 ~ 0x03_7FFF	BWP29	0x07_4000 ~ 0x07_7FFF
BWP14	0x03_8000 ~ 0x03_BFFF	BWP30	0x07_8000 ~ 0x07_BFFF
BWP15	0x03_C000 ~ 0x03_FFFF	BWP31	0x07_C000 ~ 0x07_FFFF

Config3 (Address = 0x0030_000C)

31	30	29	28	27	26	25	24
CFGCHECKSUM[31:24]							
23	22	21	20	19	18	17	16
CFGCHECKSUM[23:16]							
15	14	13	12	11	10	9	8
CFGCHECKSUM[15:8]							
7	6	5	4	3	2	1	0
CFGCHECKSUM[7:0]							

Bits	Field	Description
[31:0]	**CFGCHECKSUM**	**User Configuration CRC Checksum** If Config0~Config2 are not 0xFFFF_FFFF, Config3 needs to program the correct CRC checksum value; otherwise, if the CRC checksum value is not correct, this chip will enter lock protect mode automatically. The Cyclic Redundancy Check (CRC) polynomial is CRC-8: $X^8 + X^2 + X + 1$ for each 8 bits with seed 0xFF Example : if Config0 = 0xF8FF_FFF...E Config1 = 0x0001_F00...0 Config2 = 0xFFFF_FFF...0 Then CFGCHECKSUM = 0x E5E6_E7B5.

6.13.9 In System Program (ISP)

The application program memory and data flash are supported both in hardware programming and in system programming (ISP). Hardware programming mode uses gang-writers to reduce programming costs and time to market while the products enter into the mass production state. However, if the product is just under development or the end product needs firmware updating in the hand of an end user, the hardware programming mode will make repeated programming difficult and inconvenient. ISP method makes it easy and possible.The NUC442/NUC472 supports ISP mode allowing a device to be reprogrammed under software control. Furthermore, the capability to update the application firmware makes wide range of applications possible.

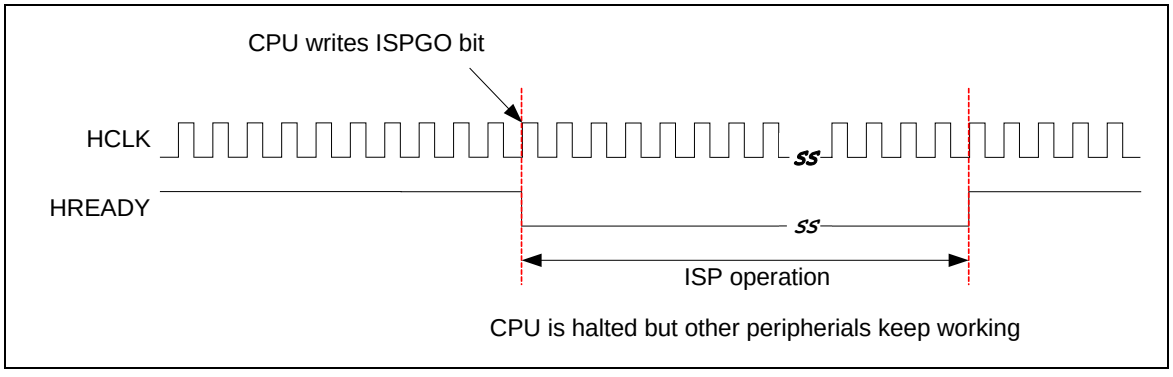
ISP is performed without removing the microcontroller from the system. Various interfaces enable LDROM firmware to get new program code easily. The most common method to perform ISP is via UART along with the firmware in LDROM. General speaking, PC transfers the new APROM code through serial port. Then LDROM firmware receives it and re-programs into APROM through ISP commands. The ISP firmware and PC application program for the NUC442/NUC472 series enables user to easily perform ISP through Nuvoton ISP tool.

6.13.10 ISP Procedure

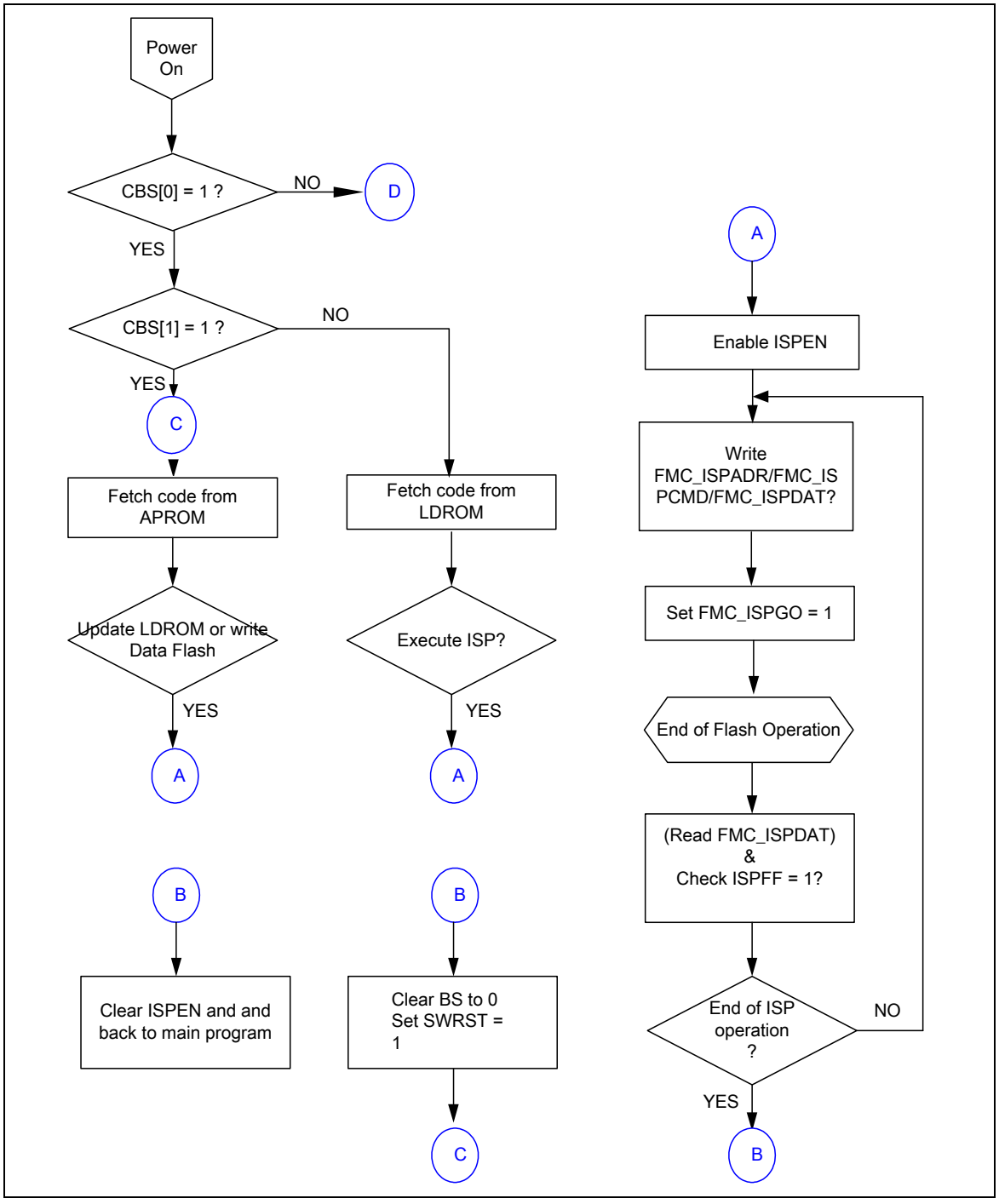
The NUC442/NUC472 supports booting from APROM or LDROM initially defined by user configuration bits (CBS). If user wants to update application program in APROM, he can write BS=1 and starts software reset to make chip boot from LDROM. The first step to start ISP function is write ISPEN bit to 1. Software is required to write SYS_REGLCTL register in Global Control Register (GCR) with 0x59, 0x16 and 0x88 before writing FMC_ISPCTL register. This procedure is used to protect flash memory from destroying owning to unintended write during power on/off duration.

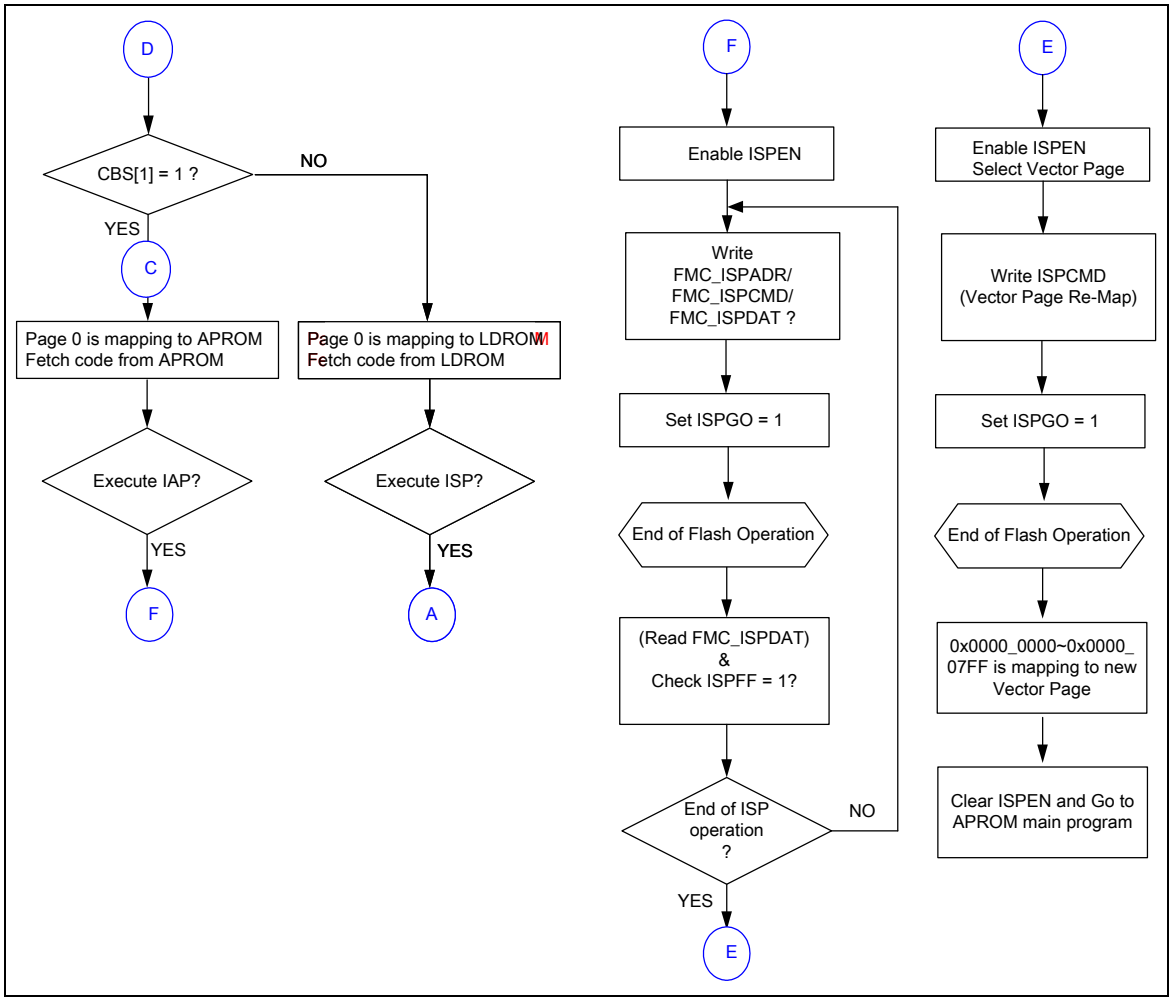
Several error conditions are checked after software writes ISPGO bit. If error condition occurs, ISP operation is not been started and ISP fail flag will be set instead of. ISPFF flag is cleared by s/w, it will not be over written in next ISP operation. The next ISP procedure can be started even ISPFF bit keeps at 1. It is recommended that software to check ISPFF bit and clear it after each ISP operation if it is set to 1.

When ISPGO bit is set, CPU will wait for ISP operation finish, during this period; peripheral still keeps working as usual. If any interrupt request occur, CPU will not service it till ISP operation finish. When ISP operation is finished, the ISPGO bit will be cleared by hardware automatically. User can know if ISP operation is finished by checking this bit.



Note: The NuMicro™ NUC442/NUC472 series allows user to update CONFIG value by ISP.


Power On
CBS[0] = 1 ?
NO
D
YES
CBS[1] = 1 ?
NO
YES
C
Fetch code from APROM
Update LDROM or write Data Flash
YES
A
B
Clear ISPEN and and back to main program
Fetch code from LDROM
Execute ISP?
YES
A
B
Clear BS to 0
Set SWRST = 1
C
A
Enable ISPEN
Write FMC_ISPADR/FMC_ISPCMD/FMC_ISPDAT?
Set FMC_ISPGO = 1
End of Flash Operation
(Read FMC_ISPDAT) & Check ISPFF = 1?
End of ISP operation ?
NO
YES
B


D
CBS[1] = 1 ?
NO
YES
C
Page 0 is mapping to APROM
Fetch code from APROM
Page 0 is mapping to LDROM
Fetch code from LDROM
Execute IAP?
Execute ISP?
YES
YES
F
A
F
Enable ISPEN
Write
FMC_ISPADR/
FMC_ISPCMD/
FMC_ISPDAT ?
Set ISPGO = 1
End of Flash Operation
(Read FMC_ISPDAT)
&
Check ISPFF = 1?
End of ISP
operation
?
NO
YES
E
E
Enable ISPEN
Select Vector Page
Write ISPCMD
(Vector Page Re-Map)
Set ISPGO = 1
End of Flash Operation
0x0000_0000~0x0000_
07FF is mapping to new
Vector Page
Clear ISPEN and Go to
APROM main program

ISP Mode	ISPCMD			ISPADR			ISPDAT
	CMD	CMD	CMD[3:0]	A21	A20	A[19:0]	D[31:0]
FLASH Page Erase	1	0	0010	0	A20	Address in A[19:0]	0xFFFF_FFFF
FLASH Program	1	0	0001	0	A20	Address in A[19:0]	Data in D[31:0]
FLASH Read	0	0	0000	0	A20	Address in A[19:0]	Data out D[31:0]
CONFIG Page Erase	1	0	0010	1	1	Address in A[19:0]	0xFFFF_FFFF
CONFIG Program	1	0	0001	1	1	Address in A[19:0]	Data in D[31:0]
CONFIG Read	0	0	0000	1	1	Address in A[19:0]	Data out D[31:0]
Vector Page Re-Map	1	0	1110	0	A20	Address in A[19:0]	x
Read Company ID	0	0	1011	0	0	Address in A[19:0]	Data out D[31:0] = 0x0000_00DA
Read Device ID	0	0	11000	0	0	Address in A[19:0]	Data out D[31:0] = 0x0000_xxxx
Read Unique ID	0	0	0100	0	0	Address in A[19:0] = 0x0_0000 0x0_0004 0x0_0008	Data out
Read Unique CID	0	0	0100	0	0	Address in A[19:0] = 0x0_0010 0x0_0014 0x0_0018 0x0_001C	Data out

Table 6.13-3 ISP Mode Command

6.13.11 Flash Control Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
FMC Base Address: **FMC_BA = 0x4000_C000**				
FMC_ISPCTL	FMC_BA+0x00	R/W	ISP Control Register	0x0000_0000
FMC_ISPADDR	FMC_BA+0x04	R/W	ISP Address Register	0x0000_0000
FMC_ISPDAT	FMC_BA+0x08	R/W	ISP Data Register	0x0000_0000
FMC_ISPCMD	FMC_BA+0x0C	R/W	ISP Command Register	0x0000_0000
FMC_ISPTRG	FMC_BA+0x10	R/W	ISP Trigger Register	0x0000_0000
FMC_DFBA	FMC_BA+0x14	R	Data Flash Base Address	0xXXXX_XXXX
FMC_FTCTL	FMC_BA+0x18	R/W	Flash Access Time Control Register	0x0000_0000
FMC_ISPSTS	FMC_BA+0x40	R	ISP Status Register	0x0000_0000
FMC_FBWP	FMC_BA+0x44	R/W	Flash Block Write Protect Control Register	0xXXXX_XXXX
FMC_MPDAT0	FMC_BA+0x80	R/W	ISP Multi-Word Program Data0 Register	0x0000_0000
FMC_MPDAT1	FMC_BA+0x84	R/W	ISP Multi-Word Program Data1 Register	0x0000_0000
FMC_MPDAT2	FMC_BA+0x88	R/W	ISP Multi-Word Program Data2 Register	0x0000_0000
FMC_MPDAT3	FMC_BA+0x8C	R/W	ISP Multi-Word Program Data3 Register	0x0000_0000
FMC_MPSTS	FMC_BA+0xC0	R	ISP Multi-Word Program Status Register	0x0000_0000
FMC_MPADDR	FMC_BA+0xC4	R	ISP Multi-Word Program Address Status Register	0x0000_0000

6.13.12 Flash Control Register Description

ISP Control Register (FMC_ISPCTL)

Register	Offset	R/W	Description	Reset Value
FMC_ISPCTL	FMC_BA+0x00	R/W	ISP Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	ISPFF	LDUEN	CFGUEN	APUEN	Reserved	BS	ISPEN

Bits	Description	
[31:7]	**Reserved**	Reserved.
[6]	**ISPFF**	**ISP Fail Flag (Write Protect)** This bit is set by hardware when a triggered ISP meets any of the following conditions: (1) APROM writes to itself if APUEN is set to 0. (2) LDROM writes to itself if LDUEN is set to 0. (3) CONFIG is erased/programmed if CFGUEN is set to 0. (4) Destination address is illegal, such as over an available range. **Note:** This bit needs to be cleared by writing 1 to it.
[5]	**LDUEN**	**LDROM Update Enable Bit (Write Protect)** LDROM update enable bit. 0 = LDROM cannot be updated. 1 = LDROM can be updated.
[4]	**CFGUEN**	**Config-Bits Update By ISP Enable Bit (Write Protect)** 0 = ISP Disabled to update config-bits. 1 = ISP Enabled to update config-bits.
[3]	**APUEN**	**APROM Update Enable Bit (Write Protect)** 0 = APROM cannot be updated when the chip runs in APROM. 1 = APROM can be updated when the chip runs in APROM.
[2]	**Reserved**	Reserved.

[1]	**BS**	**Boot Select (Write Protect)** Set/clear this bit to select next booting from LDROM/APROM, respectively. This bit also functions as chip booting status flag, which can be used to check where chip booted from. This bit is initiated with the inversed value of CBS in Config0 after any reset is happened except CPU reset (RSTS_CPU is 1) or system reset (RSTS_SYS) is happened 0 = Boot from APROM. 1 = Boot from LDROM.
[0]	**ISPEN**	**ISP Enable Bit (Write Protect)** ISP function enable bit. Set this bit to enable ISP function. 0 = ISP function Disabled. 1 = ISP function Enabled.

ISP Address (FMC_ISPADDR)

Register	Offset	R/W	Description	Reset Value
FMC_ISPADDR	FMC_BA+0x04	R/W	ISP Address Register	0x0000_0000

31	30	29	28	27	26	25	24
ISPADDR[31:24]							
23	22	21	20	19	18	17	16
ISPADDR[23:16]							
15	14	13	12	11	10	9	8
ISPADDR[15:8]							
7	6	5	4	3	2	1	0
ISPADDR[7:0]							

Bits	Description	
[31:0]	ISPADDR	**ISP Address** The NUC442/NUC472 series is equipped with an embedded flash and supports word program only. ISPADDR[1:0] must be kept 00b for ISP operation.

ISP Data Register (FMC_ISPDAT)

Register	Offset	R/W	Description	Reset Value
FMC_ISPDAT	FMC_BA+0x08	R/W	ISP Data Register	0x0000_0000

31	30	29	28	27	26	25	24
ISPDAT[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ISPDAT [23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ISPDAT [15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ISPDAT [7:0]							

Bits	Description	
[31:0]	**ISPDAT**	**ISP Data** Write data to this register before ISP program operation. Read data from this register after ISP read operation.

ISP Command (FMC_ISPCMD)

Register	Offset	R/W	Description	Reset Value
FMC_ISPCMD	FMC_BA+0x0C	R/W	ISP Command Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**CMD**					

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5:0]	**CMD**	**ISP Command** Please check the table below for ISP commands.

ISP command table:

Operation Mode	CMD
32b Read	0x00
32b Program	0x21
64b Read	0x40
64b Program	0x61
Multi-Word Program	0x27
Page Erase	0x22
Vector Page Re-Map	0x2E
Read UID	0x04
Read UCID	0x04
Read CID	0x0B
Read DID	0x0C

ISP Trigger Register (FMC_ISPTRG)

Register	Offset	R/W	Description	Reset Value
FMC_ISPTRG	FMC_BA+0x10	R/W	ISP Trigger Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**ISPGO**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**ISPGO**	**ISP Start Trigger** Write 1 to start ISP operation and this bit will be cleared to 0 by hardware automatically when ISP operation is finished. 0 = ISP operation is finished. 1 = ISP is progressed. This bit is the protected bit, It means programming this bit needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register SYS_REGLCTL at address GCR_BA+0x100

Data Flash Base Address Register (FMC_DFBA)

Register	Offset	R/W	Description	Reset Value
FMC_DFBA	FMC_BA+0x14	R	Data Flash Base Address	0xXXXX_XXXX

31	30	29	28	27	26	25	24
DFBA[31:23]							
23	22	21	20	19	18	17	16
DFBA[23:16]							
15	14	13	12	11	10	9	8
DFBA[15:8]							
7	6	5	4	3	2	1	0
DFBA[7:0]							

Bits	Description	
[31:0]	DFBA	**Data Flash Base Address** This register indicates data flash start address. It is a read only register. The data flash is shared with APROM and data flash size is defined by user configuration and the content of this register is loaded from Config1.

Flash Access Time Control Register (FMC_FTCTL)

Register	Offset	R/W	Description	Reset Value
FMC_FTCTL	FMC_BA+0x18	R/W	Flash Access Time Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	FOM			Reserved			

Bits	Description	
[31:8]	**Reserved**	Reserved.
[6:4]	**FOM**	**Frequency Optimization Mode (Write Protect)** When chip operation frequency is lower, chip can work more efficiently by setting FOM bits **FOM[2:0]** / **Optimized Frequency (OF)** 001 / 0 MHz < OF ≤ 24 MHz 010 / 24 MHz < OF ≤ 48 MHz 011 / 48 MHz < OF ≤ 72 MHz others / Reserved
[3:0]	**Reserved**	Reserved.

ISP Status Register (FMC_ISPSTS)

Register	Offset	R/W	Description	Reset Value
FMC_ISPSTS	FMC_BA+0x40	R	ISP Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					CFGCRCF	Reserved	
23	22	21	20	19	18	17	16
Reserved			VECMAP[11:7]				
15	14	13	12	11	10	9	8
VECMAP[6:0]							Reserved
7	6	5	4	3	2	1	0
Reserved	ISPFF	Reserved			CBS		ISPBUSY

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26]	**CFGCRCF**	**User-Configuration CRC Check Flag (Read Only)** This bit is set by hardware when detecting CONFIG CRC checksum is error 0 = CONFIG CRC checksum is OK. 1 = CONFIG CRC checksum error and force chip into LOCK mode.
[25:21]	**Reserved**	Reserved.
[20:9]	**VECMAP**	**Vector Page Mapping Address (Read Only)** The current flash address space 0x0000_0000~0x0000_07FF is mapping to address {VECMAP[11:2], 11'h000} ~ {VECMAP[11:2], 11'h7FF} VECMAP[1:0] is needed to set 0.
[8:7]	**Reserved**	Reserved.
[6]	**ISPFF**	**ISP Fail Flag (Read Only)** This bit is set by hardware when a triggered ISP meets any of the following conditions: (1) APROM writes to itself if APUEN is set to 0. (2) LDROM writes to itself if LDUEN is set to 0. (3) CONFIG is erased/programmed if CFGUEN is set to 0. (4) Destination address is illegal, such as over an available range.
[5:3]	**Reserved**	Reserved.
[2:1]	**CBS**	**Chip Boot Selection Mode** This CBS field is just a copy of User-Configuration Config0 CBS[7:6].
[0]	**ISPBUSY**	**ISP Busy Flag** 0 = ISP operation is finished. 1 = ISP is progressed.

Flash Block Write Protect Control Register (FMC_FBWP)

Register	Offset	R/W	Description	Reset Value
FMC_FBWP	FMC_BA+0x44	R/W	Flash Block Write Protect Control Register	0xXXXX_XXXX

31	30	29	28	27	26	25	24
BWP31	BWP30	BWP29	BWP28	BWP27	BWP26	BWP25	BWP24
23	22	21	20	19	18	17	16
BWP23	BWP22	BWP21	BWP20	BWP19	BWP18	BWP17	BWP16
15	14	13	12	11	10	9	8
BWP15	BWP14	BWP13	BWP12	BWP11	BWP10	BWP9	BWP8
7	6	5	4	3	2	1	0
BWP7	BWP6	BWP5	BWP4	BWP3	BWP2	BWP1	BWP0

Bits	Description	
[31:0]	BWP	**Flash Block Write Protect Control** If BWP.N bit is set to 0, the APROM memory relative region cannot program and erase by ISP. (N=0~31).

Bit	Block Protect Region	Bit	Block Protect Region
BWP0	0x00_0000 ~ 0x00_3FFF	BWP16	0x04_0000 ~ 0x04_3FFF
BWP1	0x00_4000 ~ 0x00_7FFF	BWP17	0x04_4000 ~ 0x04_7FFF
BWP2	0x00_8000 ~ 0x00_BFFF	BWP18	0x04_8000 ~ 0x04_BFFF
BWP3	0x00_C000 ~ 0x00_FFFF	BWP19	0x04_C000 ~ 0x04_FFFF
BWP4	0x01_0000 ~ 0x01_3FFF	BWP20	0x05_0000 ~ 0x05_3FFF
BWP5	0x01_4000 ~ 0x01_7FFF	BWP21	0x05_4000 ~ 0x05_7FFF
BWP6	0x01_8000 ~ 0x01_BFFF	BWP22	0x05_8000 ~ 0x05_BFFF
BWP7	0x01_C000 ~ 0x01_FFFF	BWP23	0x05_C000 ~ 0x05_FFFF
BWP8	0x02_0000 ~ 0x02_3FFF	BWP24	0x06_0000 ~ 0x06_3FFF
BWP9	0x02_4000 ~ 0x02_7FFF	BWP25	0x06_4000 ~ 0x06_7FFF
BWP10	0x02_8000 ~ 0x02_BFFF	BWP26	0x06_8000 ~ 0x06_BFFF
BWP11	0x02_C000 ~ 0x02_FFFF	BWP27	0x06_C000 ~ 0x06_FFFF
BWP12	0x03_0000 ~ 0x03_3FFF	BWP28	0x07_0000 ~ 0x07_3FFF
BWP13	0x03_4000 ~ 0x03_7FFF	BWP29	0x07_4000 ~ 0x07_7FFF
BWP14	0x03_8000 ~ 0x03_BFFF	BWP30	0x07_8000 ~ 0x07_BFFF
BWP15	0x03_C000 ~ 0x03_FFFF	BWP31	0x07_C000 ~ 0x07_FFFF

This register is loaded from Config2 when chip is power on. It is read only, except the correct Super Key is matched.

		This register is also a protected bit which means programming this bit needs to write “59h”, “16h”, “88h” to address GCR_BA+0x100 to disable register protection. Refer to the register SYS_REGLCTL at address GCR_BA+0x100

ISP Multi-Word Program Data 0 Register (FMC_MPDAT0)

Register	Offset	R/W	Description	Reset Value
FMC_MPDAT0	FMC_BA+0x80	R/W	ISP Multi-Word Program Data 0 Register	0x0000_0000

31	30	29	28	27	26	25	24
ISPDAT0 [31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ISPDAT0 [23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ISPDAT0 [15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ISPDAT0 [7:0]							

Bits	Description	
[31:0]	**ISPDAT0**	**ISP Data 0** This register is the first 32-bit data for 32b/64b/multi-word program, and it is also the mirror of FMC_ISPDAT register, both registers keep the same data.

ISP Multi-Word Program Data 1 Register (FMC_MPDAT1)

Register	Offset	R/W	Description	Reset Value
FMC_MPDAT1	FMC_BA+0x84	R/W	ISP Multi-Word Program Data 1 Register	0x0000_0000

31	30	29	28	27	26	25	24
ISPDAT1 [31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ISPDAT1 [23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ISPDAT1 [15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ISPDAT1 [7:0]							

Bits	Description	
[31:0]	**ISPDAT1**	**ISP Data 1** This register is the second 32-bit data for 32b/64b/multi-word program.

ISP Multi-Word Program Data 2 Register (FMC_MPDAT2)

Register	Offset	R/W	Description	Reset Value
FMC_MPDAT2	FMC_BA+0x88	R/W	ISP Multi-Word Program Data 2 Register	0x0000_0000

31	30	29	28	27	26	25	24
ISPDAT2 [31:24]							
23	22	21	20	19	18	17	16
ISPDAT2 [23:16]							
15	14	13	12	11	10	9	8
ISPDAT2 [15:8]							
7	6	5	4	3	2	1	0
ISPDAT2 [7:0]							

Bits	Description	
[31:0]	ISPDAT2	ISP Data 2 This register is the third 32-bit data for 32b/64b/multi-word program.

ISP Multi-Word Program Data 3 Register (FMC_MPDAT3)

Register	Offset	R/W	Description	Reset Value
FMC_MPDAT3	FMC_BA+0x8C	R/W	ISP Multi-Word Program Data 3 Register	0x0000_0000

31	30	29	28	27	26	25	24
ISPDAT3 [31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ISPDAT3 [23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ISPDAT3 [15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ISPDAT3 [7:0]							

Bits	Description	
[31:0]	**ISPDAT3**	**ISP Data 3** This register is the fourth 32-bit data for 32b/64b/multi-word program.

ISP Multi-Word Program Status Register (FMC_MPSTS)

Register	Offset	R/W	Description	Reset Value
FMC_MPSTS	FMC_BA+0xC0	R	ISP Multi-Word Program Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
D3	D2	D1	D0	Reserved	ISPFF	Reserved	MPBUSY

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**D3**	**ISP DATA 3 Flag (Read Only)** This bit is set when FMC_MPDAT3 is written and auto-clear to 0 when the FMC_MPDAT3 is programmed to flash complete. 0 = FMC_MPDAT3 register is empty, or program to flash complete. 1 = FMC_MPDAT3 register has been written, and not programmed to flash yet.
[6]	**D2**	**ISP DATA 2 Flag (Read Only)** This bit is set when FMC_MPDAT2 is written and auto-clear to 0 when the FMC_MPDAT2 is programmed to flash complete. 0 = FMC_MPDAT2 register is empty, or program to flash complete. 1 = FMC_MPDAT2 register has been written, and not programmed to flash yet.
[5]	**D1**	**ISP DATA 1 Flag (Read Only)** This bit is set when FMC_MPDAT1 is written and auto-clear to 0 when the FMC_MPDAT1 is programmed to flash complete. 0 = FMC_MPDAT1 register is empty, or program to flash complete. 1 = FMC_MPDAT1 register has been written, and not programmed to flash yet.
[4]	**D0**	**ISP DATA 0 Flag (Read Only)** This bit is set when FMC_MPDAT0 is written and auto-clear to 0 when the FMC_MPDAT0 is programmed to flash complete. 0 = FMC_MPDAT0 register is empty, or program to flash complete. 1 = FMC_ISPDAT0 register has been written, and not programmed to flash yet.
[2]	**ISPFF**	**ISP Fail Flag (Read Only)** This bit is set when ISP Multi-Word Program operation failed.
[1]	**Reserved**	Reserved.

[0]	**MPBUSY**	**ISP Multi-Word Program Busy Flag (Read Only)** 0 = ISP Multi-Word Program operation is aborted or finished. 1 = ISP Multi-Word Program operation is progressed.

ISP Multi-Word Program Address Status (FMC_MPADDR)

Register	Offset	R/W	Description	Reset Value
FMC_MPADDR	FMC_BA+0xC4	R/W	ISP Multi-Word Program Address Status Register	0x0000_0000

31	30	29	28	27	26	25	24
MPADDR[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
MPADDR[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
MPADDR[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MPADDR[7:0]							

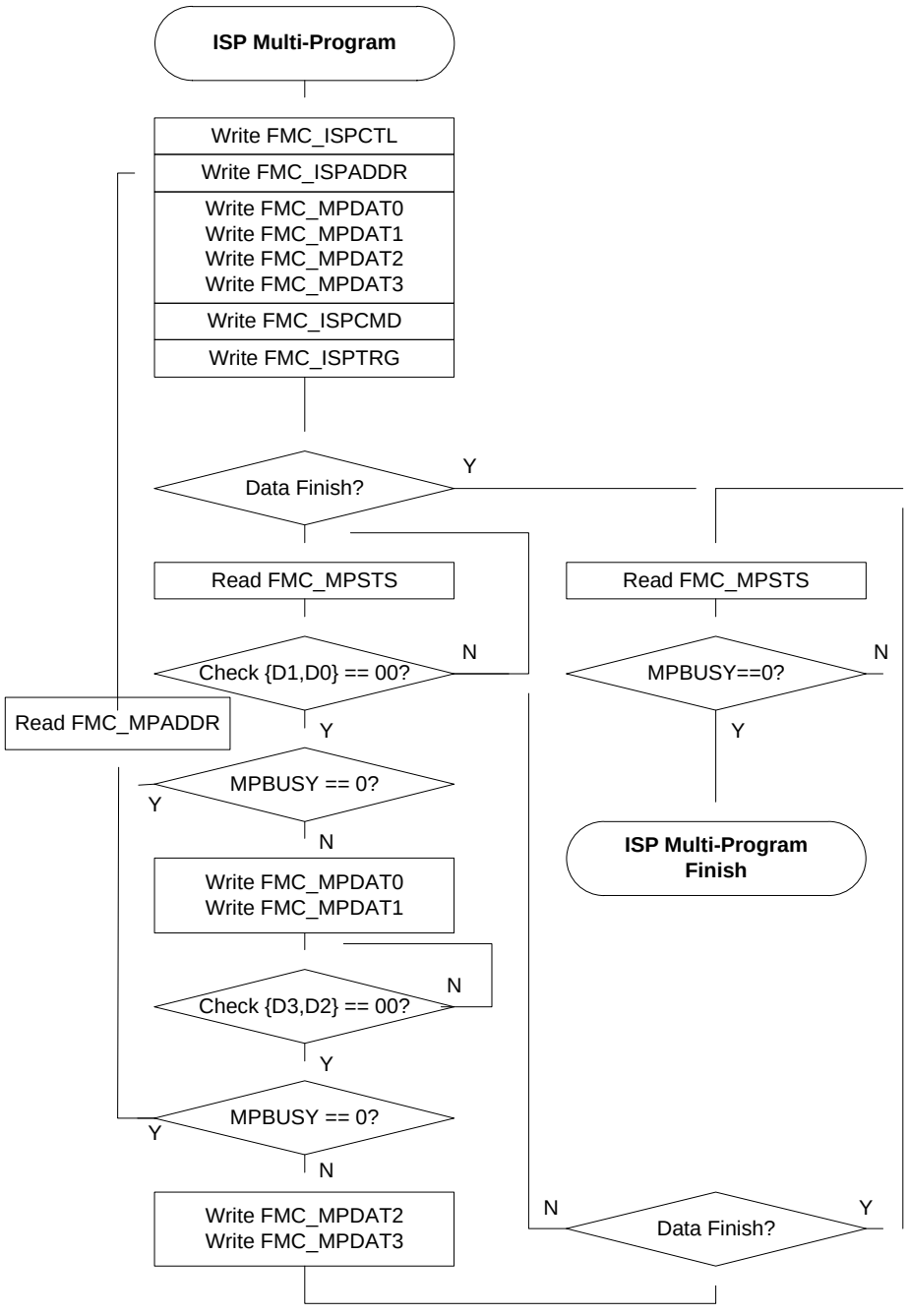
Bits	Description	
[31:0]	**MPADDR**	**ISP Multi-Word Program Address Status** MPADDR is the address of ISP Multi-Word Program operation when MPBUSY flag is 1. MPADDR will keep the final address when Multi-Word Program is aborted or finished.

Multi-Program Flow and Notice

1. CPU has to run this ISP operation by through internal SRAM.
2. CPU should monitor the Buffer status (FMC_MPSTS) to update the programming data in time.

Otherwise, FMC will exit ISP operation.

3. If CPU cannot update data in time, CPU needs restart Multi-Word Program procedure to continue, FMC_MPADDR provides the last program address information.
4. The max. programming length is 512 bytes
5. The min. programming length is 16 bytes (4 words)
6. The starting ISP address in FMC_ISPADDR register has to be 16-byte align, that means, bit[3:0] should be 0, and the address is limited on APROM and LDROM.



Software View

6.14 General Purpose I/O (GPIO)

6.14.1 Overview

The NUC442/NUC472 series has up to 114/144 General Purpose I/O pins shared with other function pins depending on the chip configuration. These 114/144 pins are arranged in 8/9 ports named GPIOA, GPIOB, GPIOC, GPIOD, GPIOE, GPIOF, GPIOG, GPIOH and GPIOI (NUC472 only). GPIO has 16 pins on echo port. Each one of the 144 pins is independent and has the corresponding register bits to control the pin mode function and data.

The I/O type of each of I/O pins can be configured by software individually as input, output, open-drain or quasi-bidirectional mode. After reset, the I/O type of all pins stay in quasi-bidirectional mode and port data register Px_DOUT[15:0] resets to 0x0000_FFFF. Each I/O pin has a very weak individual pull-up resistor which is about 110 KΩ~300 KΩ for V_{DD} is from 5.0 V to 2.5 V.

6.14.2 Features

- Four I/O modes:
  - Quasi-bidirectional mode
  - Push-Pull Output mode
  - Open-Drain Output mode
  - Input only with high impendence mode
- TTL/Schmitt trigger input selectable
- I/O pin can be configured as interrupt source with edge/level setting
- Supports High Driver and High Sink IO mode

6.14.3 Functional Description

6.14.3.1 Input Mode

Set Px_MODE (MODEn[1:0]) to 00b and the GPIOx port [n] pin is in Input mode and the I/O pin is in tri-state (high impedance) without output drive capability. The Px_PIN value reflects the status of the corresponding port pins.

6.14.3.2 Output Mode

Set Px_MODE (MODEn[1:0]) to 01b when the GPIOx port [n] pin is in Output mode and the I/O pin supports digital output function with source/sink current capability. The bit value in the corresponding bit [n] of Px_DOUT is driven on the pin.

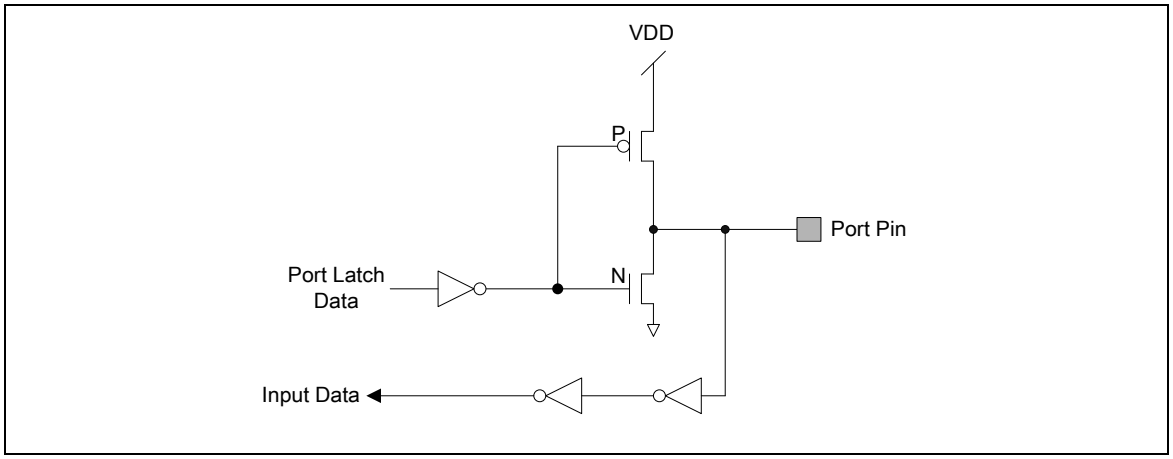


Figure 6.14-1 Push-Pull Output

6.14.3.3 Open-Drain Mode

Set Px_MODE (MODEn[1:0]) to 10b the GPIOx port [n] pin is in Open-Drain mode and the digital output function of I/O pin supports only sink current capability, an additional pull-up register is needed for driving high state. If the bit value in the corresponding bit [n] of Px_DOUT is 0, the pin drive a “low” output on the pin. If the bit value in the corresponding bit [n] of Px_DOUT is 1, the pin output drives high that is controlled by external pull high resistor.

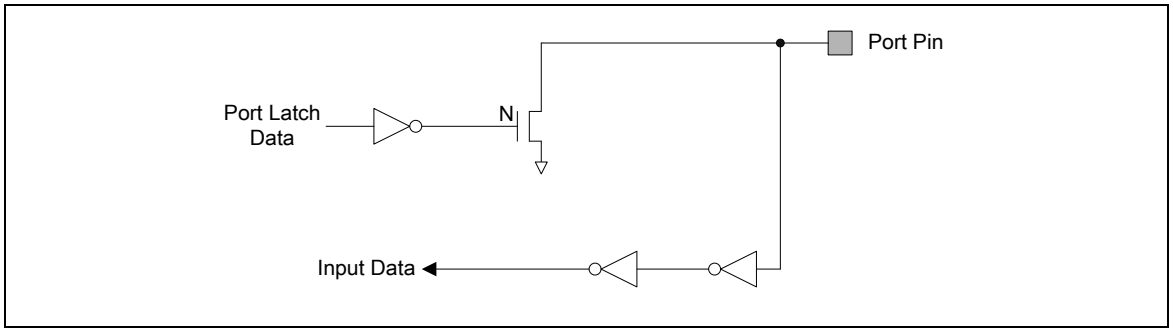


Figure 6.14-2 Open-Drain Output

6.14.3.4 Quasi-bidirectional Mode

Set Px_MODE (MODEn[1:0]) to 11b the GPIOx port [n] pin is in Quasi-bidirectional mode and the I/O pin supports digital output and input function at the same time but the source current is only up to hundreds uA. Before the digital input function is performed the corresponding bit in Px_DOUT must be set to 1. The quasi-bidirectional output is common on the 80C51 and most of its derivatives. If the bit value in the corresponding bit [n] of Px_DOUT is 0, the pin drive a “low” output on the pin. If the bit value in the corresponding bit [n] of Px_DOUT is 1, the pin will check the pin value. If pin value is high, no action takes. If pin state is low, the pin will drive strong high with 2 clock cycles on the pin and then disable the strong output drive and then the pin status is control by internal pull-up resistor. Note that the source current capability in quasi-bidirectional mode is only about 200 uA to 30 uA for VDD is form 5.0 V to 2.5 V.

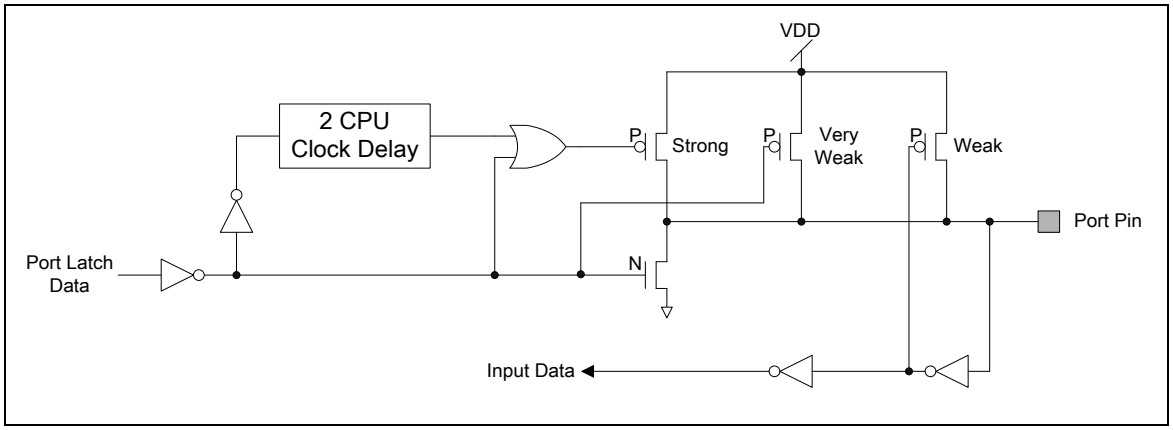


Figure 6.14-3 Quasi-bidirectional I/O Mode

6.14.4 Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
GPIO Base Address: **GPIO_BA = 0x4000_4000**				
PA_MODE	GPIO_BA+0x000	R/W	PA I/O Mode Control	0xFFFF_FFFF
PA_DINOFF	GPIO_BA+0x004	R/W	PA Digital Input Path Disable Control	0x0000_0000
PA_DOUT	GPIO_BA+0x008	R/W	PA Data Output Value	0x0000_FFFF
PA_DATMSK	GPIO_BA+0x00C	R/W	PA Data Output Write Mask	0x0000_0000
PA_PIN	GPIO_BA+0x010	R	PA Pin Value	0x0000_XXXX
PA_DBEN	GPIO_BA+0x014	R/W	PA De-Bounce Enable Control	0x0000_0000
PA_INTTYPE	GPIO_BA+0x018	R/W	PA Interrupt Trigger Type Register	0x0000_0000
PA_INTEN	GPIO_BA+0x01C	R/W	PA Interrupt Enable	0x0000_0000
PA_INTSRC	GPIO_BA+0x020	R/W	PA Interrupt Source Flag	0x0000_XXXX
PA_SMTEN	GPIO_BA+0x024	R/W	PA Input Schmitt Trigger Enable	0x0000_0000
PA_SLEWCTL	GPIO_BA+0x028	R/W	PA High Slew Rate Control	0x0000_0000
PB_MODE	GPIO_BA+0x040	R/W	PB I/O Mode Control	0xFFFF_FFFF
PB_DINOFF	GPIO_BA+0x044	R/W	PB Digital Input Path Disable Control	0x0000_0000
PB_DOUT	GPIO_BA+0x048	R/W	PB Data Output Value	0x0000_FFFF
PB_DATMSK	GPIO_BA+0x04C	R/W	PB Data Output Write Mask	0x0000_0000
PB_PIN	GPIO_BA+0x050	R	PB Pin Value	0x0000_XXXX
PB_DBEN	GPIO_BA+0x054	R/W	PB De-Bounce Enable Control	0x0000_0000
PB_INTTYPE	GPIO_BA+0x058	R/W	PB Interrupt Trigger Type Register	0x0000_0000
PB_INTEN	GPIO_BA+0x05C	R/W	PB Interrupt Enable	0x0000_0000
PB_INTSRC	GPIO_BA+0x060	R/W	PB Interrupt Source Flag	0x0000_XXXX
PB_SMTEN	GPIO_BA+0x064	R/W	PB Input Schmitt Trigger Enable	0x0000_0000
PB_SLEWCTL	GPIO_BA+0x068	R/W	PB High Slew Rate Control	0x0000_0000
PC_MODE	GPIO_BA+0x080	R/W	PC I/O Mode Control	0xFFFF_FFFF
PC_DINOFF	GPIO_BA+0x084	R/W	PC Digital Input Path Disable Control	0x0000_0000
PC_DOUT	GPIO_BA+0x088	R/W	PC Data Output Value	0x0000_FFFF

Register	Offset	R/W	Description	Reset Value
PC_DATMSK	GPIO_BA+0x08C	R/W	PC Data Output Write Mask	0x0000_0000
PC_PIN	GPIO_BA+0x090	R	PC Pin Value	0x0000_XXXX
PC_DBEN	GPIO_BA+0x094	R/W	PC De-Bounce Enable Control	0x0000_0000
PC_INTTYPE	GPIO_BA+0x098	R/W	PC Interrupt Trigger Type Register	0x0000_0000
PC_INTEN	GPIO_BA+0x09C	R/W	PC Interrupt Enable	0x0000_0000
PC_INTSRC	GPIO_BA+0x0A0	R/W	PC Interrupt Source Flag	0x0000_XXXX
PC_SMTEN	GPIO_BA+0x0A4	R/W	PC Input Schmitt Trigger Enable	0x0000_0000
PC_SLEWCTL	GPIO_BA+0x0A8	R/W	PC High Slew Rate Control	0x0000_0000
PD_MODE	GPIO_BA+0x0C0	R/W	PD I/O Mode Control	0xFFFF_FFFF
PD_DINOFF	GPIO_BA+0x0C4	R/W	PD Digital Input Path Disable Control	0x0000_0000
PD_DOUT	GPIO_BA+0x0C8	R/W	PD Data Output Value	0x0000_FFFF
PD_DATMSK	GPIO_BA+0x0CC	R/W	PD Data Output Write Mask	0x0000_0000
PD_PIN	GPIO_BA+0x0D0	R	PD Pin Value	0x0000_XXXX
PD_DBEN	GPIO_BA+0x0D4	R/W	PD De-Bounce Enable Control	0x0000_0000
PD_INTTYPE	GPIO_BA+0x0D8	R/W	PD Interrupt Trigger Type Register	0x0000_0000
PD_INTEN	GPIO_BA+0x0DC	R/W	PD Interrupt Enable	0x0000_0000
PD_INTSRC	GPIO_BA+0x0E0	R/W	PD Interrupt Source Flag	0x0000_XXXX
PD_SMTEN	GPIO_BA+0x0E4	R/W	PD Input Schmitt Trigger Enable	0x0000_0000
PD_SLEWCTL	GPIO_BA+0x0E8	R/W	PD High Slew Rate Control	0x0000_0000
PE_MODE	GPIO_BA+0x100	R/W	PE I/O Mode Control	0xFFFF_FFFF
PE_DINOFF	GPIO_BA+0x104	R/W	PE Digital Input Path Disable Control	0x0000_0000
PE_DOUT	GPIO_BA+0x108	R/W	PE Data Output Value	0x0000_FFFF
PE_DATMSK	GPIO_BA+0x10C	R/W	PE Data Output Write Mask	0x0000_0000
PE_PIN	GPIO_BA+0x110	R	PE Pin Value	0x0000_XXXX
PE_DBEN	GPIO_BA+0x114	R/W	PE De-Bounce Enable Control	0x0000_0000
PE_INTTYPE	GPIO_BA+0x118	R/W	PE Interrupt Trigger Type Register	0x0000_0000
PE_INTEN	GPIO_BA+0x11C	R/W	PE Interrupt Enable	0x0000_0000
PE_INTSRC	GPIO_BA+0x120	R/W	PE Interrupt Source Flag	0x0000_XXXX

Register	Offset	R/W	Description	Reset Value
PE_SMTEN	GPIO_BA+0x124	R/W	PE Input Schmitt Trigger Enable	0x0000_0000
PE_SLEWCTL	GPIO_BA+0x128	R/W	PE High Slew Rate Control	0x0000_0000
PF_MODE	GPIO_BA+0x140	R/W	PF I/O Mode Control	0xFFFF_FFFF
PF_DINOFF	GPIO_BA+0x144	R/W	PF Digital Input Path Disable Control	0x0000_0000
PF_DOUT	GPIO_BA+0x148	R/W	PF Data Output Value	0x0000_FFFF
PF_DATMSK	GPIO_BA+0x14C	R/W	PF Data Output Write Mask	0x0000_0000
PF_PIN	GPIO_BA+0x150	R	PF Pin Value	0x0000_XXXX
PF_DBEN	GPIO_BA+0x154	R/W	PF De-Bounce Enable Control	0x0000_0000
PF_INTTYPE	GPIO_BA+0x158	R/W	PF Interrupt Trigger Type Register	0x0000_0000
PF_INTEN	GPIO_BA+0x15C	R/W	PF Interrupt Enable	0x0000_0000
PF_INTSRC	GPIO_BA+0x160	R/W	PF Interrupt Source Flag	0x0000_XXXX
PF_SMTEN	GPIO_BA+0x164	R/W	PF Input Schmitt Trigger Enable	0x0000_0000
PF_SLEWCTL	GPIO_BA+0x168	R/W	PF High Slew Rate Control	0x0000_0000
PG_MODE	GPIO_BA+0x180	R/W	PG I/O Mode Control	0xFFFF_FFFF
PG_DINOFF	GPIO_BA+0x184	R/W	PG Digital Input Path Disable Control	0x0000_0000
PG_DOUT	GPIO_BA+0x188	R/W	PG Data Output Value	0x0000_FFFF
PG_DATMSK	GPIO_BA+0x18C	R/W	PG Data Output Write Mask	0x0000_0000
PG_PIN	GPIO_BA+0x190	R	PG Pin Value	0x0000_XXXX
PG_DBEN	GPIO_BA+0x194	R/W	PG De-Bounce Enable Control	0x0000_0000
PG_INTTYPE	GPIO_BA+0x198	R/W	PG Interrupt Trigger Type Register	0x0000_0000
PG_INTEN	GPIO_BA+0x19C	R/W	PG Interrupt Enable	0x0000_0000
PG_INTSRC	GPIO_BA+0x1A0	R/W	PG Interrupt Source Flag	0x0000_XXXX
PG_SMTEN	GPIO_BA+0x1A4	R/W	PG Input Schmitt Trigger Enable	0x0000_0000
PG_SLEWCTL	GPIO_BA+0x1A8	R/W	PG High Slew Rate Control	0x0000_0000
PH_MODE	GPIO_BA+0x1C0	R/W	PH I/O Mode Control	0xFFFF_FFFF
PH_DINOFF	GPIO_BA+0x1C4	R/W	PH Digital Input Path Disable Control	0x0000_0000
PH_DOUT	GPIO_BA+0x1C8	R/W	PH Data Output Value	0x0000_FFFF
PH_DATMSK	GPIO_BA+0x1CC	R/W	PH Data Output Write Mask	0x0000_0000
PH_PIN	GPIO_BA+0x1D0	R	PH Pin Value	0x0000_XXXX

Register	Offset	R/W	Description	Reset Value
PH_DBEN	GPIO_BA+0x1D4	R/W	PH De-Bounce Enable Control	0x0000_0000
PH_INTTYPE	GPIO_BA+0x1D8	R/W	PH Interrupt Trigger Type Register	0x0000_0000
PH_INTEN	GPIO_BA+0x1DC	R/W	PH Interrupt Enable	0x0000_0000
PH_INTSRC	GPIO_BA+0x1E0	R/W	PH Interrupt Source Flag	0x0000_XXXX
PH_SMTEN	GPIO_BA+0x1E4	R/W	PH Input Schmitt Trigger Enable	0x0000_0000
PH_SLEWCTL	GPIO_BA+0x1E8	R/W	PH High Slew Rate Control	0x0000_0000
PI_MODE	GPIO_BA+0x200	R/W	PI I/O Mode Control	0xFFFF_FFFF
PI_DINOFF	GPIO_BA+0x204	R/W	PI Digital Input Path Disable Control	0x0000_0000
PI_DOUT	GPIO_BA+0x208	R/W	PI Data Output Value	0x0000_FFFF
PI_DATMSK	GPIO_BA+0x20C	R/W	PI Data Output Write Mask	0x0000_0000
PI_PIN	GPIO_BA+0x210	R	PI Pin Value	0x0000_XXXX
PI_DBEN	GPIO_BA+0x214	R/W	PI De-Bounce Enable Control	0x0000_0000
PI_INTTYPE	GPIO_BA+0x218	R/W	PI Interrupt Trigger Type Register	0x0000_0000
PI_INTEN	GPIO_BA+0x21C	R/W	PI Interrupt Enable	0x0000_0000
PI_INTSRC	GPIO_BA+0x220	R/W	PI Interrupt Source Flag	0x0000_XXXX
PI_SMTEN	GPIO_BA+0x224	R/W	PI Input Schmitt Trigger Enable	0x0000_0000
PI_SLEWCTL	GPIO_BA+0x228	R/W	PI High Slew Rate Control	0x0000_0000
GPIO_DBCTL	GPIO_BA+0x440	R/W	Interrupt De-bounce Control	0x0000_0020
PA0_PDIO	GPIO_BA+0x800	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA1_PDIO	GPIO_BA+0x804	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA2_PDIO	GPIO_BA+0x808	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA3_PDIO	GPIO_BA+0x80C	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA4_PDIO	GPIO_BA+0x810	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA5_PDIO	GPIO_BA+0x814	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA6_PDIO	GPIO_BA+0x818	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA7_PDIO	GPIO_BA+0x81C	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA8_PDIO	GPIO_BA+0x820	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA9_PDIO	GPIO_BA+0x824	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA10_PDIO	GPIO_BA+0x828	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001

Register	Offset	R/W	Description	Reset Value
PA11_PDIO	GPIO_BA+0x82C	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA12_PDIO	GPIO_BA+0x830	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA13_PDIO	GPIO_BA+0x834	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA14_PDIO	GPIO_BA+0x838	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA15_PDIO	GPIO_BA+0x83C	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PB0_PDIO	GPIO_BA+0x840	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB1_PDIO	GPIO_BA+0x844	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB2_PDIO	GPIO_BA+0x848	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB3_PDIO	GPIO_BA+0x84C	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB4_PDIO	GPIO_BA+0x850	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB5_PDIO	GPIO_BA+0x854	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB6_PDIO	GPIO_BA+0x858	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB7_PDIO	GPIO_BA+0x85C	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB8_PDIO	GPIO_BA+0x860	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB9_PDIO	GPIO_BA+0x864	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB10_PDIO	GPIO_BA+0x868	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB11_PDIO	GPIO_BA+0x86C	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB12_PDIO	GPIO_BA+0x870	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB13_PDIO	GPIO_BA+0x874	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB14_PDIO	GPIO_BA+0x878	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB15_PDIO	GPIO_BA+0x87C	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PC0_PDIO	GPIO_BA+0x880	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC1_PDIO	GPIO_BA+0x884	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC2_PDIO	GPIO_BA+0x888	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC3_PDIO	GPIO_BA+0x88C	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC4_PDIO	GPIO_BA+0x890	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC5_PDIO	GPIO_BA+0x894	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC6_PDIO	GPIO_BA+0x898	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC7_PDIO	GPIO_BA+0x89C	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001

Register	Offset	R/W	Description	Reset Value
PC8_PDIO	GPIO_BA+0x8A0	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC9_PDIO	GPIO_BA+0x8A4	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC10_PDIO	GPIO_BA+0x8A8	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC11_PDIO	GPIO_BA+0x8AC	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC12_PDIO	GPIO_BA+0x8B0	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC13_PDIO	GPIO_BA+0x8B4	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC14_PDIO	GPIO_BA+0x8B8	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC15_PDIO	GPIO_BA+0x8BC	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PD0_PDIO	GPIO_BA+0x8C0	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD1_PDIO	GPIO_BA+0x8C4	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD2_PDIO	GPIO_BA+0x8C8	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD3_PDIO	GPIO_BA+0x8CC	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD4_PDIO	GPIO_BA+0x8D0	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD5_PDIO	GPIO_BA+0x8D4	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD6_PDIO	GPIO_BA+0x8D8	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD7_PDIO	GPIO_BA+0x8DC	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD8_PDIO	GPIO_BA+0x8E0	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD9_PDIO	GPIO_BA+0x8E4	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD10_PDIO	GPIO_BA+0x8E8	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD11_PDIO	GPIO_BA+0x8EC	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD12_PDIO	GPIO_BA+0x8F0	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD13_PDIO	GPIO_BA+0x8F4	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD14_PDIO	GPIO_BA+0x8F8	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD15_PDIO	GPIO_BA+0x8FC	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PE0_PDIO	GPIO_BA+0x900	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE1_PDIO	GPIO_BA+0x904	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE2_PDIO	GPIO_BA+0x908	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE3_PDIO	GPIO_BA+0x90C	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001

Register	Offset	R/W	Description	Reset Value
PE4_PDIO	GPIO_BA+0x910	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE5_PDIO	GPIO_BA+0x914	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE6_PDIO	GPIO_BA+0x918	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE7_PDIO	GPIO_BA+0x91C	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE8_PDIO	GPIO_BA+0x920	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE9_PDIO	GPIO_BA+0x924	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE10_PDIO	GPIO_BA+0x928	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE11_PDIO	GPIO_BA+0x92C	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE12_PDIO	GPIO_BA+0x930	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE13_PDIO	GPIO_BA+0x934	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE14_PDIO	GPIO_BA+0x938	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE15_PDIO	GPIO_BA+0x93C	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PF0_PDIO	GPIO_BA+0x940	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF1_PDIO	GPIO_BA+0x944	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF2_PDIO	GPIO_BA+0x948	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF3_PDIO	GPIO_BA+0x94C	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF4_PDIO	GPIO_BA+0x950	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF5_PDIO	GPIO_BA+0x954	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF6_PDIO	GPIO_BA+0x958	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF7_PDIO	GPIO_BA+0x95C	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF8_PDIO	GPIO_BA+0x960	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF9_PDIO	GPIO_BA+0x964	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF10_PDIO	GPIO_BA+0x968	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF11_PDIO	GPIO_BA+0x96C	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF12_PDIO	GPIO_BA+0x970	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF13_PDIO	GPIO_BA+0x974	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF14_PDIO	GPIO_BA+0x978	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF15_PDIO	GPIO_BA+0x97C	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PG0_PDIO	GPIO_BA+0x980	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001

Register	Offset	R/W	Description	Reset Value
PG1_PDIO	GPIO_BA+0x984	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG2_PDIO	GPIO_BA+0x988	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG3_PDIO	GPIO_BA+0x98C	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG4_PDIO	GPIO_BA+0x990	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG5_PDIO	GPIO_BA+0x994	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG6_PDIO	GPIO_BA+0x998	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG7_PDIO	GPIO_BA+0x99C	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG8_PDIO	GPIO_BA+0x9A0	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG9_PDIO	GPIO_BA+0x9A4	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG10_PDIO	GPIO_BA+0x9A8	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG11_PDIO	GPIO_BA+0x9AC	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG12_PDIO	GPIO_BA+0x9B0	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG13_PDIO	GPIO_BA+0x9B4	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG14_PDIO	GPIO_BA+0x9B8	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG15_PDIO	GPIO_BA+0x9BC	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PH0_PDIO	GPIO_BA+0x9C0	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH1_PDIO	GPIO_BA+0x9C4	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH2_PDIO	GPIO_BA+0x9C8	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH3_PDIO	GPIO_BA+0x9CC	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH4_PDIO	GPIO_BA+0x9D0	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH5_PDIO	GPIO_BA+0x9D4	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH6_PDIO	GPIO_BA+0x9D8	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH7_PDIO	GPIO_BA+0x9DC	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH8_PDIO	GPIO_BA+0x9E0	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH9_PDIO	GPIO_BA+0x9E4	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH10_PDIO	GPIO_BA+0x9E8	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH11_PDIO	GPIO_BA+0x9EC	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH12_PDIO	GPIO_BA+0x9F0	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001

Register	Offset	R/W	Description	Reset Value
PH13_PDIO	GPIO_BA+0x9F4	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH14_PDIO	GPIO_BA+0x9F8	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH15_PDIO	GPIO_BA+0x9FC	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PI0_PDIO	GPIO_BA+0xA00	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI1_PDIO	GPIO_BA+0xA04	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI2_PDIO	GPIO_BA+0xA08	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI3_PDIO	GPIO_BA+0xA0C	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI4_PDIO	GPIO_BA+0xA10	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI5_PDIO	GPIO_BA+0xA14	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI6_PDIO	GPIO_BA+0xA18	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI7_PDIO	GPIO_BA+0xA1C	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI8_PDIO	GPIO_BA+0xA20	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI9_PDIO	GPIO_BA+0xA24	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI10_PDIO	GPIO_BA+0xA28	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI11_PDIO	GPIO_BA+0xA2C	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI12_PDIO	GPIO_BA+0xA30	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI13_PDIO	GPIO_BA+0xA34	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI14_PDIO	GPIO_BA+0xA38	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI15_PDIO	GPIO_BA+0xA3C	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001

6.14.5 Register Description

GPIO Port n I/O Mode Control

Register	Offset	R/W	Description	Reset Value
PA_MODE	GPIO_BA+0x000	R/W	PA I/O Mode Control	0xFFFF_FFFF
PB_MODE	GPIO_BA+0x040	R/W	PB I/O Mode Control	0xFFFF_FFFF
PC_MODE	GPIO_BA+0x080	R/W	PC I/O Mode Control	0xFFFF_FFFF
PD_MODE	GPIO_BA+0x0C0	R/W	PD I/O Mode Control	0xFFFF_FFFF
PE_MODE	GPIO_BA+0x100	R/W	PE I/O Mode Control	0xFFFF_FFFF
PF_MODE	GPIO_BA+0x140	R/W	PF I/O Mode Control	0xFFFF_FFFF
PG_MODE	GPIO_BA+0x180	R/W	PG I/O Mode Control	0xFFFF_FFFF
PH_MODE	GPIO_BA+0x1C0	R/W	PH I/O Mode Control	0xFFFF_FFFF
PI_MODE	GPIO_BA+0x200	R/W	PI I/O Mode Control	0xFFFF_FFFF

31	30	29	28	27	26	25	24
MODE15		MODE14		MODE13		MODE12	
23	22	21	20	19	18	17	16
MODE11		MODE10		MODE9		MODE8	
15	14	13	12	11	10	9	8
MODE7		MODE6		MODE5		MODE4	
7	6	5	4	3	2	1	0
MODE3		MODE2		MODE1		MODE0	

Bits	Description	
[2m+1:2m] m=0,1..15	**MODEm**	**Port N Bit M I/O Mode Control** Determine the I/O mode of port n bit m. 00 = INPUT only mode. 01 = OUTPUT mode. 10 = Open-drain mode. 11 = Quasi-bidirectional mode. Reset value: 0xFFFF_FFFF when (cfg_io =1'b0). 0x0000_00000 when (cfg_io =1'b1).

GPIO Port n Digital Input Path Disable

Register	Offset	R/W	Description	Reset Value
PA_DINOFF	GPIO_BA+0x004	R/W	PA Digital Input Path Disable Control	0x0000_0000
PB_DINOFF	GPIO_BA+0x044	R/W	PB Digital Input Path Disable Control	0x0000_0000
PC_DINOFF	GPIO_BA+0x084	R/W	PC Digital Input Path Disable Control	0x0000_0000
PD_DINOFF	GPIO_BA+0x0C4	R/W	PD Digital Input Path Disable Control	0x0000_0000
PE_DINOFF	GPIO_BA+0x104	R/W	PE Digital Input Path Disable Control	0x0000_0000
PF_DINOFF	GPIO_BA+0x144	R/W	PF Digital Input Path Disable Control	0x0000_0000
PG_DINOFF	GPIO_BA+0x184	R/W	PG Digital Input Path Disable Control	0x0000_0000
PH_DINOFF	GPIO_BA+0x1C4	R/W	PH Digital Input Path Disable Control	0x0000_0000
PI_DINOFF	GPIO_BA+0x204	R/W	PI Digital Input Path Disable Control	0x0000_0000

31	30	29	28	27	26	25	24
DINOFF[15:8]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DINOFF[7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[m+16] m=0,1..15	**DINOFF[m]**	**Port N Bit M Off Digital Input Path** Each of these bits is used to turn off the digital input path of port n bit m pin. If input is analog signal, users can turn off digital input path to avoid input current leakage. 0 = Digital input path Enabled. 1 = Digital input path Disabled (Digital input is tied to low).
[15:0]	**Reserved**	Reserved.

GPIO Port n Data Output

Register	Offset	R/W	Description	Reset Value
PA_DOUT	GPIO_BA+0x008	R/W	PA Data Output Value	0x0000_FFFF
PB_DOUT	GPIO_BA+0x048	R/W	PB Data Output Value	0x0000_FFFF
PC_DOUT	GPIO_BA+0x088	R/W	PC Data Output Value	0x0000_FFFF
PD_DOUT	GPIO_BA+0x0C8	R/W	PD Data Output Value	0x0000_FFFF
PE_DOUT	GPIO_BA+0x108	R/W	PE Data Output Value	0x0000_FFFF
PF_DOUT	GPIO_BA+0x148	R/W	PF Data Output Value	0x0000_FFFF
PG_DOUT	GPIO_BA+0x188	R/W	PG Data Output Value	0x0000_FFFF
PH_DOUT	GPIO_BA+0x1C8	R/W	PH Data Output Value	0x0000_FFFF
PI_DOUT	GPIO_BA+0x208	R/W	PI Data Output Value	0x0000_FFFF

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DOUT[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DOUT[7:0]							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[m] m=0,1..15	**DOUT[m]**	**Port N Bit M Output** Each of these bits control the status of port n bit m when this pin is configures as output, open-drain, or Quasi-bidirectional mode. 0 = Drive port n bit m high low. 1 = Drive port n bit m high level.

GPIO Port n Data Output Write Mask

Register	Offset	R/W	Description	Reset Value
PA_DATMSK	GPIO_BA+0x00C	R/W	PA Data Output Write Mask	0x0000_0000
PB_DATMSK	GPIO_BA+0x04C	R/W	PB Data Output Write Mask	0x0000_0000
PC_DATMSK	GPIO_BA+0x08C	R/W	PC Data Output Write Mask	0x0000_0000
PD_DATMSK	GPIO_BA+0x0CC	R/W	PD Data Output Write Mask	0x0000_0000
PE_DATMSK	GPIO_BA+0x10C	R/W	PE Data Output Write Mask	0x0000_0000
PF_DATMSK	GPIO_BA+0x14C	R/W	PF Data Output Write Mask	0x0000_0000
PG_DATMSK	GPIO_BA+0x18C	R/W	PG Data Output Write Mask	0x0000_0000
PH_DATMSK	GPIO_BA+0x1CC	R/W	PH Data Output Write Mask	0x0000_0000
PI_DATMSK	GPIO_BA+0x20C	R/W	PI Data Output Write Mask	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DATMSK[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATMSK[7:0]							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[m] m=0,1..15	**DATMSK[m]**	**Port N Bit M Data Output Write Mask** These bits are used to protect the corresponding register of Pn_DOUT[m]. When set the DATMSK [m] to 1, the writing to Pn_DOUT[m] bit is ignored. The write to port pin latch is masked. 0 = Pn_DOUT[m] bit writing is valid. 1 = Pn_DOUT[m] bit writing is ignored.

GPIO Port n Pin Value

Register	Offset	R/W	Description	Reset Value
PA_PIN	GPIO_BA+0x010	R	PA Pin Value	0x0000_XXXX
PB_PIN	GPIO_BA+0x050	R	PB Pin Value	0x0000_XXXX
PC_PIN	GPIO_BA+0x090	R	PC Pin Value	0x0000_XXXX
PD_PIN	GPIO_BA+0x0D0	R	PD Pin Value	0x0000_XXXX
PE_PIN	GPIO_BA+0x110	R	PE Pin Value	0x0000_XXXX
PF_PIN	GPIO_BA+0x150	R	PF Pin Value	0x0000_XXXX
PG_PIN	GPIO_BA+0x190	R	PG Pin Value	0x0000_XXXX
PH_PIN	GPIO_BA+0x1D0	R	PH Pin Value	0x0000_XXXX
PI_PIN	GPIO_BA+0x210	R	PI Pin Value	0x0000_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PIN[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PIN[7:0]							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[m] m=0,1..15	**PIN[m]**	**Port N Bit M Pin Value** Each bit of the register reflects the actual status of the respective port pin. If bit is 1, it indicates the corresponding pin status is high, else the pin status is low.

GPIO Port n De-bounce Enable

Register	Offset	R/W	Description	Reset Value
PA_DBEN	GPIO_BA+0x014	R/W	PA De-Bounce Enable Control	0x0000_0000
PB_DBEN	GPIO_BA+0x054	R/W	PB De-Bounce Enable Control	0x0000_0000
PC_DBEN	GPIO_BA+0x094	R/W	PC De-Bounce Enable Control	0x0000_0000
PD_DBEN	GPIO_BA+0x0D4	R/W	PD De-Bounce Enable Control	0x0000_0000
PE_DBEN	GPIO_BA+0x114	R/W	PE De-Bounce Enable Control	0x0000_0000
PF_DBEN	GPIO_BA+0x154	R/W	PF De-Bounce Enable Control	0x0000_0000
PG_DBEN	GPIO_BA+0x194	R/W	PG De-Bounce Enable Control	0x0000_0000
PH_DBEN	GPIO_BA+0x1D4	R/W	PH De-Bounce Enable Control	0x0000_0000
PI_DBEN	GPIO_BA+0x214	R/W	PI De-Bounce Enable Control	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DBEN[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DBEN[7:0]							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[m] m=0,1..15	**DBEN[m]**	**Port N Bit M Input De-Bounce Enable** DBEN[m] is used to enable the de-bounce function for each corresponding bit. DBEN[m] is valid for “edge-triggered” interrupt only and is ignored for “level triggered” interrupt. If the input signal pulse width can’t be sampled by continuous two de-bounce sample cycle The input signal transition is seen as the signal bounce and will not trigger the interrupt. The de-bounce clock is controlled by GPIO_DBCTL register. 0 = Port n bit m input de-bounce Disabled. 1 = Port n bit m input de-bounce Enabled.

GPIO Port n Interrupt Trigger Type

Register	Offset	R/W	Description	Reset Value
PA_INTTYPE	GPIO_BA+0x018	R/W	PA Interrupt Trigger Type Register	0x0000_0000
PB_INTTYPE	GPIO_BA+0x058	R/W	PB Interrupt Trigger Type Register	0x0000_0000
PC_INTTYPE	GPIO_BA+0x098	R/W	PC Interrupt Trigger Type Register	0x0000_0000
PD_INTTYPE	GPIO_BA+0x0D8	R/W	PD Interrupt Trigger Type Register	0x0000_0000
PE_INTTYPE	GPIO_BA+0x118	R/W	PE Interrupt Trigger Type Register	0x0000_0000
PF_INTTYPE	GPIO_BA+0x158	R/W	PF Interrupt Trigger Type Register	0x0000_0000
PG_INTTYPE	GPIO_BA+0x198	R/W	PG Interrupt Trigger Type Register	0x0000_0000
PH_INTTYPE	GPIO_BA+0x1D8	R/W	PH Interrupt Trigger Type Register	0x0000_0000
PI_INTTYPE	GPIO_BA+0x218	R/W	PI Interrupt Trigger Type Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TYPE[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TYPE[7:0]							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[m] m=0,1..15	**TYPE[m]**	**Port N Bit M Edge Or Level Triggered Interrupt Control** TYPE[m] decides the pin interrupt triggered by level or edge. If the interrupt is by edge trigger, the trigger source can be controlled by de-bounce. If the interrupt is by level trigger, the input source is sampled by one HCLK clock and generates the interrupt. 0 = Edge triggered interrupt. 1 = Level triggered interrupt. **Note1:** If pin is set as the level trigger interrupt, only one level can be set on the registers Pn_INTEN. If set both the level to trigger interrupt, the setting is ignored and no interrupt will occur. **Note2:** The de-bounce function is valid for edge triggered interrupt. If the interrupt mode

		is level triggered, the de-bounce enable bit is ignored.

GPIO Port n Interrupt Enable

Register	Offset	R/W	Description	Reset Value
PA_INTEN	GPIO_BA+0x01C	R/W	PA Interrupt Enable	0x0000_0000
PB_INTEN	GPIO_BA+0x05C	R/W	PB Interrupt Enable	0x0000_0000
PC_INTEN	GPIO_BA+0x09C	R/W	PC Interrupt Enable	0x0000_0000
PD_INTEN	GPIO_BA+0x0DC	R/W	PD Interrupt Enable	0x0000_0000
PE_INTEN	GPIO_BA+0x11C	R/W	PE Interrupt Enable	0x0000_0000
PF_INTEN	GPIO_BA+0x15C	R/W	PF Interrupt Enable	0x0000_0000
PG_INTEN	GPIO_BA+0x19C	R/W	PG Interrupt Enable	0x0000_0000
PH_INTEN	GPIO_BA+0x1DC	R/W	PH Interrupt Enable	0x0000_0000
PI_INTEN	GPIO_BA+0x21C	R/W	PI Interrupt Enable	0x0000_0000

31	30	29	28	27	26	25	24
RHIEN[15:8]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RHIEN[7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FLIEN[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FLIEN[7:0]							

Bits	Description	
[m+16] m=0,1..15	**RHIEN[m]**	**Port N Bit M Interrupt Enable For Rising Edge Or High Level Input** RHIEN[m] enables the interrupt for each of the corresponding input of Port n. Setting this bit to 1 also enables the pin wake-up function. 0 = Port n bit m high-level or rising edge interrupt Disabled. 1 = Port n bit m high-level or rising edge interrupt Enabled.
[m] m=0,1..15	**FLIEN[m]**	**Port N Bit M Interrupt Enable For Falling Edge Or Low Level Input** FLIEN[n] enables the interrupt for each of the corresponding input of Port n. Setting this bit to 1 also enables the pin wake-up function. 0 = Port n bit m low-level or falling edge interrupt Disabled. 1 = Port n bit m low-level or falling edge interrupt Enabled.

GPIO Port n Interrupt Source Flag

Register	Offset	R/W	Description	Reset Value
PA_INTSRC	GPIO_BA+0x020	R/W	PA Interrupt Source Flag	0x0000_XXXX
PB_INTSRC	GPIO_BA+0x060	R/W	PB Interrupt Source Flag	0x0000_XXXX
PC_INTSRC	GPIO_BA+0x0A0	R/W	PC Interrupt Source Flag	0x0000_XXXX
PD_INTSRC	GPIO_BA+0x0E0	R/W	PD Interrupt Source Flag	0x0000_XXXX
PE_INTSRC	GPIO_BA+0x120	R/W	PE Interrupt Source Flag	0x0000_XXXX
PF_INTSRC	GPIO_BA+0x160	R/W	PF Interrupt Source Flag	0x0000_XXXX
PG_INTSRC	GPIO_BA+0x1A0	R/W	PG Interrupt Source Flag	0x0000_XXXX
PH_INTSRC	GPIO_BA+0x1E0	R/W	PH Interrupt Source Flag	0x0000_XXXX
PI_INTSRC	GPIO_BA+0x220	R/W	PI Interrupt Source Flag	0x0000_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
INTSRC[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INTSRC[7:0]							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[m] m=0,1..15	**INTSRC [m]**	**Port N Bit M Interrupt Trigger Source Indicator** Read: 0 = No interrupt at Port n. 1 = Port n bit m generate an interrupt. Write: 0= No effect. 1= Clear the correspond pending interrupt.

GPIO Port n Input Schmitt Trigger Enable

Register	Offset	R/W	Description	Reset Value
PA_SMTEN	GPIO_BA+0x024	R/W	PA Input Schmitt Trigger Enable	0x0000_0000
PB_SMTEN	GPIO_BA+0x064	R/W	PB Input Schmitt Trigger Enable	0x0000_0000
PC_SMTEN	GPIO_BA+0x0A4	R/W	PC Input Schmitt Trigger Enable	0x0000_0000
PD_SMTEN	GPIO_BA+0x0E4	R/W	PD Input Schmitt Trigger Enable	0x0000_0000
PE_SMTEN	GPIO_BA+0x124	R/W	PE Input Schmitt Trigger Enable	0x0000_0000
PF_SMTEN	GPIO_BA+0x164	R/W	PF Input Schmitt Trigger Enable	0x0000_0000
PG_SMTEN	GPIO_BA+0x1A4	R/W	PG Input Schmitt Trigger Enable	0x0000_0000
PH_SMTEN	GPIO_BA+0x1E4	R/W	PH Input Schmitt Trigger Enable	0x0000_0000
PI_SMTEN	GPIO_BA+0x224	R/W	PI Input Schmitt Trigger Enable	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SMTEN[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SMTEN[7:0]							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[m] m=0,1..15	**SMTEN[m]**	**Port N Bit M Input Schmitt Trigger Enable Bit** 0 = P I/O input Schmitt Trigger function Disabled. 1 = P I/O input Schmitt Trigger function Enabled.

GPIO Port n High Slew Rate Control

Register	Offset	R/W	Description	Reset Value
PA_SLEWCTL	GPIO_BA+0x028	R/W	PA High Slew Rate Control	0x0000_0000
PB_SLEWCTL	GPIO_BA+0x068	R/W	PB High Slew Rate Control	0x0000_0000
PC_SLEWCTL	GPIO_BA+0x0A8	R/W	PC High Slew Rate Control	0x0000_0000
PD_SLEWCTL	GPIO_BA+0x0E8	R/W	PD High Slew Rate Control	0x0000_0000
PE_SLEWCTL	GPIO_BA+0x128	R/W	PE High Slew Rate Control	0x0000_0000
PF_SLEWCTL	GPIO_BA+0x168	R/W	PF High Slew Rate Control	0x0000_0000
PG_SLEWCTL	GPIO_BA+0x1A8	R/W	PG High Slew Rate Control	0x0000_0000
PH_SLEWCTL	GPIO_BA+0x1E8	R/W	PH High Slew Rate Control	0x0000_0000
PI_SLEWCTL	GPIO_BA+0x228	R/W	PI High Slew Rate Control	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
HSREN[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
HSREN[7:0]							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[m] m=0,1..15	**HSREN[m]**	**Port N Bit M High Slew Rate Control** 0 = P I/O output with basic slew rate. 1 = P I/O output with higher slew rate.

Note: This bit is effective only with HSREN Pin.

Interrupt De-bounce Cycle Control (GPIO_DBCTL)

Register	Offset	R/W	Description	Reset Value
GPIO_DBCTL	GPIO_BA+0x440	R/W	Interrupt De-bounce Control	0x0000_0020

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		ICLKON	DBCLKSRC	DBCLKSEL			

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**ICLKON**	**Interrupt Clock On Mode** Setting this bit to 0 will disable the interrupt generate circuit clock if the pin[n] interrupt is disabled. 0 = Disable the clock if the all port interrupts are disabled. 1 = Interrupt generated circuit clock always Enabled.
[4]	**DBCLKSRC**	**De-Bounce Counter Clock Source Selection** 0 = De-bounce counter clock source is the HCLK. 1 = De-bounce counter clock source is the internal 10 kHz clock.
[3:0]	**DBCLKSEL**	**De-Bounce Sampling Cycle Selection** 0000 = Sample interrupt input once per 1 clocks. 0001 = Sample interrupt input once per 2 clocks. 0010 = Sample interrupt input once per 4 clocks. 0011 = Sample interrupt input once per 8 clocks. 0100 = Sample interrupt input once per 16 clocks. 0101 = Sample interrupt input once per 32 clocks. 0110 = Sample interrupt input once per 64 clocks. 0111 = Sample interrupt input once per 128 clocks. 1000 = Sample interrupt input once per 256 clocks. 1001 = Sample interrupt input once per 2*256 clocks. 1010 = Sample interrupt input once per 4*256 clocks. 1011 = Sample interrupt input once per 8*256 clocks. 1100 = Sample interrupt input once per 16*256 clocks. 1101 = Sample interrupt input once per 32*256 clocks. 1110 = Sample interrupt input once per 64*256 clocks. 1111 = Sample interrupt input once per 128*256 clocks.

GPIO Port n Bit m I/O value

Register	Offset	R/W	Description	Reset Value
PA0_PDIO	GPIO_BA+0x800	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA1_PDIO	GPIO_BA+0x804	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA2_PDIO	GPIO_BA+0x808	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA3_PDIO	GPIO_BA+0x80C	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA4_PDIO	GPIO_BA+0x810	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA5_PDIO	GPIO_BA+0x814	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA6_PDIO	GPIO_BA+0x818	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA7_PDIO	GPIO_BA+0x81C	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA8_PDIO	GPIO_BA+0x820	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA9_PDIO	GPIO_BA+0x824	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA10_PDIO	GPIO_BA+0x828	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA11_PDIO	GPIO_BA+0x82C	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA12_PDIO	GPIO_BA+0x830	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA13_PDIO	GPIO_BA+0x834	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA14_PDIO	GPIO_BA+0x838	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PA15_PDIO	GPIO_BA+0x83C	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PB0_PDIO	GPIO_BA+0x840	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB1_PDIO	GPIO_BA+0x844	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB2_PDIO	GPIO_BA+0x848	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB3_PDIO	GPIO_BA+0x84C	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB4_PDIO	GPIO_BA+0x850	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB5_PDIO	GPIO_BA+0x85	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001

	4			
PB6_PDIO	GPIO_BA+0x858	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB7_PDIO	GPIO_BA+0x85C	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB8_PDIO	GPIO_BA+0x860	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB9_PDIO	GPIO_BA+0x864	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB10_PDIO	GPIO_BA+0x868	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB11_PDIO	GPIO_BA+0x86C	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB12_PDIO	GPIO_BA+0x870	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB13_PDIO	GPIO_BA+0x874	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB14_PDIO	GPIO_BA+0x878	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PB15_PDIO	GPIO_BA+0x87C	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PC0_PDIO	GPIO_BA+0x880	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC1_PDIO	GPIO_BA+0x884	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC2_PDIO	GPIO_BA+0x888	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC3_PDIO	GPIO_BA+0x88C	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC4_PDIO	GPIO_BA+0x890	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC5_PDIO	GPIO_BA+0x894	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC6_PDIO	GPIO_BA+0x898	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC7_PDIO	GPIO_BA+0x89C	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC8_PDIO	GPIO_BA+0x8A0	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC9_PDIO	GPIO_BA+0x8A4	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC10_PDIO	GPIO_BA+0x8A8	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC11_PDIO	GPIO_BA+0x8AC	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001

PC12_PDIO	GPIO_BA+0x8B0	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC13_PDIO	GPIO_BA+0x8B4	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC14_PDIO	GPIO_BA+0x8B8	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PC15_PDIO	GPIO_BA+0x8BC	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PD0_PDIO	GPIO_BA+0x8C0	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD1_PDIO	GPIO_BA+0x8C4	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD2_PDIO	GPIO_BA+0x8C8	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD3_PDIO	GPIO_BA+0x8CC	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD4_PDIO	GPIO_BA+0x8D0	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD5_PDIO	GPIO_BA+0x8D4	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD6_PDIO	GPIO_BA+0x8D8	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD7_PDIO	GPIO_BA+0x8DC	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD8_PDIO	GPIO_BA+0x8E0	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD9_PDIO	GPIO_BA+0x8E4	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD10_PDIO	GPIO_BA+0x8E8	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD11_PDIO	GPIO_BA+0x8EC	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD12_PDIO	GPIO_BA+0x8F0	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD13_PDIO	GPIO_BA+0x8F4	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD14_PDIO	GPIO_BA+0x8F8	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PD15_PDIO	GPIO_BA+0x8FC	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PE0_PDIO	GPIO_BA+0x900	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE1_PDIO	GPIO_BA+0x904	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE2_PDIO	GPIO_BA+0x908	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001

PE3_PDIO	GPIO_BA+0x90C	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE4_PDIO	GPIO_BA+0x910	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE5_PDIO	GPIO_BA+0x914	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE6_PDIO	GPIO_BA+0x918	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE7_PDIO	GPIO_BA+0x91C	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE8_PDIO	GPIO_BA+0x920	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE9_PDIO	GPIO_BA+0x924	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE10_PDIO	GPIO_BA+0x928	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE11_PDIO	GPIO_BA+0x92C	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE12_PDIO	GPIO_BA+0x930	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE13_PDIO	GPIO_BA+0x934	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE14_PDIO	GPIO_BA+0x938	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PE15_PDIO	GPIO_BA+0x93C	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001
PF0_PDIO	GPIO_BA+0x940	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF1_PDIO	GPIO_BA+0x944	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF2_PDIO	GPIO_BA+0x948	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF3_PDIO	GPIO_BA+0x94C	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF4_PDIO	GPIO_BA+0x950	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF5_PDIO	GPIO_BA+0x954	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF6_PDIO	GPIO_BA+0x958	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF7_PDIO	GPIO_BA+0x95C	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF8_PDIO	GPIO_BA+0x960	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF9_PDIO	GPIO_BA+0x964	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001

PF10_PDIO	GPIO_BA+0x968	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF11_PDIO	GPIO_BA+0x96C	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF12_PDIO	GPIO_BA+0x970	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF13_PDIO	GPIO_BA+0x974	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF14_PDIO	GPIO_BA+0x978	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PF15_PDIO	GPIO_BA+0x97C	R/W	GPIO PF.n Pin Data Input/Output	0x0000_0001
PG0_PDIO	GPIO_BA+0x980	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG1_PDIO	GPIO_BA+0x984	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG2_PDIO	GPIO_BA+0x988	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG3_PDIO	GPIO_BA+0x98C	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG4_PDIO	GPIO_BA+0x990	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG5_PDIO	GPIO_BA+0x994	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG6_PDIO	GPIO_BA+0x998	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG7_PDIO	GPIO_BA+0x99C	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG8_PDIO	GPIO_BA+0x9A0	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG9_PDIO	GPIO_BA+0x9A4	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG10_PDIO	GPIO_BA+0x9A8	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG11_PDIO	GPIO_BA+0x9AC	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG12_PDIO	GPIO_BA+0x9B0	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG13_PDIO	GPIO_BA+0x9B4	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG14_PDIO	GPIO_BA+0x9B8	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PG15_PDIO	GPIO_BA+0x9BC	R/W	GPIO PG.n Pin Data Input/Output	0x0000_0001
PH0_PDIO	GPIO_BA+0x9C0	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001

PH1_PDIO	GPIO_BA+0x9C4	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH2_PDIO	GPIO_BA+0x9C8	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH3_PDIO	GPIO_BA+0x9CC	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH4_PDIO	GPIO_BA+0x9D0	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH5_PDIO	GPIO_BA+0x9D4	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH6_PDIO	GPIO_BA+0x9D8	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH7_PDIO	GPIO_BA+0x9DC	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH8_PDIO	GPIO_BA+0x9E0	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH9_PDIO	GPIO_BA+0x9E4	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH10_PDIO	GPIO_BA+0x9E8	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH11_PDIO	GPIO_BA+0x9EC	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH12_PDIO	GPIO_BA+0x9F0	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH13_PDIO	GPIO_BA+0x9F4	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH14_PDIO	GPIO_BA+0x9F8	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PH15_PDIO	GPIO_BA+0x9FC	R/W	GPIO PH.n Pin Data Input/Output	0x0000_0001
PI0_PDIO	GPIO_BA+0xA00	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI1_PDIO	GPIO_BA+0xA04	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI2_PDIO	GPIO_BA+0xA08	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI3_PDIO	GPIO_BA+0xA0C	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI4_PDIO	GPIO_BA+0xA10	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI5_PDIO	GPIO_BA+0xA14	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI6_PDIO	GPIO_BA+0xA18	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI7_PDIO	GPIO_BA+0xA1C	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001

PI8_PDIO	GPIO_BA+0xA20	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI9_PDIO	GPIO_BA+0xA24	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI10_PDIO	GPIO_BA+0xA28	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI11_PDIO	GPIO_BA+0xA2C	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI12_PDIO	GPIO_BA+0xA30	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI13_PDIO	GPIO_BA+0xA34	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI14_PDIO	GPIO_BA+0xA38	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001
PI15_PDIO	GPIO_BA+0xA3C	R/W	GPIO PI.n Pin Data Input/Output	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**PDIO**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**PDIO**	**Port N Bit M (PDIO) Value** Write: 0 = Clear PDIO port latch to output low. 1 = Set PDIO port latch to output high. Read: 0 = Port pin of PDIO is a low level. 1 = Port pin of PDIO is a high level. For example, a writing of PA0 reflects the value of bit PA_DOUT[0], a reading returns the value of PA_PIN[0].

6.15 I²C Serial Interface Controller (Master/Slave)

6.15.1 Overview

I²C is a two-wire, bi-directional serial bus that provides a simple and efficient method of data exchange between devices. The I²C standard is a true multi-master bus including collision detection and arbitration that prevents data corruption if two or more masters attempt to control the bus simultaneously.

Data is transferred between a Master and a Slave synchronously to SCL on the SDA line on a byte-by-byte basis. Each data byte is 8-bit long. There is one SCL clock pulse for each data bit with the MSB being transmitted first. An acknowledge bit follows each transferred byte. Each bit is sampled during the high period of SCL; therefore, the SDA line may be changed only during the low period of SCL and must be held stable during the high period of SCL. A transition on the SDA line while SCL is high is interpreted as a command (START or STOP). Please refer to the following figure for more details about I²C Bus Timing.

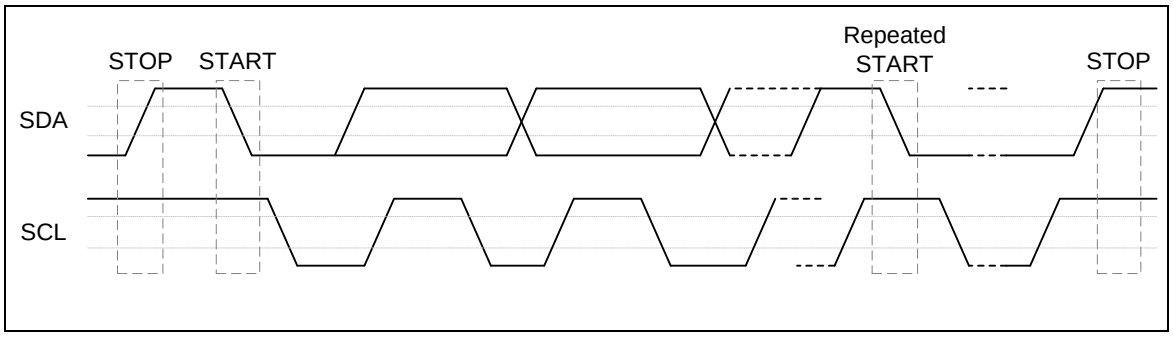


Figure 6.15-1 I²C Bus Timing

The device on-chip I²C logic provides the serial interface that meets the I²C bus standard mode specification. The I²C port handles byte transfers autonomously. To enable this port, the bit I2CEN (I2C_CTL[6]) should be set to '1'. The I²C H/W interfaces to the I²C bus via two pins: SDA and SCL. Pull up resistor is needed for I²C operation as these are open drain pins. When the I/O pins are used as I²C port, user must set the pins function to I²C in advance.

6.15.2 Features

The I²C bus uses two wires (SDA and SCL) to transfer information between devices connected to the bus. The main features of the bus are:

- Master/Slave mode
- Bidirectional data transfer between masters and slaves
- Multi-master bus
- Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
- Serial clock synchronization allows devices with different bit rates to communicate via one serial bus
- Serial clock synchronization can be used as a handshake mechanism to stretch and un-stretch serial transfer
- Built-in a 14-bit time-out counter will request the I²C interrupt if the I²C bus hangs up and time-out counter overflows.
- Programmable divider allowing for versatile rate control
- Supports 7-bit addressing mode
- Supports multiple address recognition (four slave addresses with mask option)
- Supports address match wakeup function

6.15.3 Functional Description

6.15.3.1 I²C Protocol

Normally, a standard communication consists of four parts:

1) START or Repeated START signal generation
2) Slave address and R/W bit transfer
3) Data transfer
4) STOP signal generation

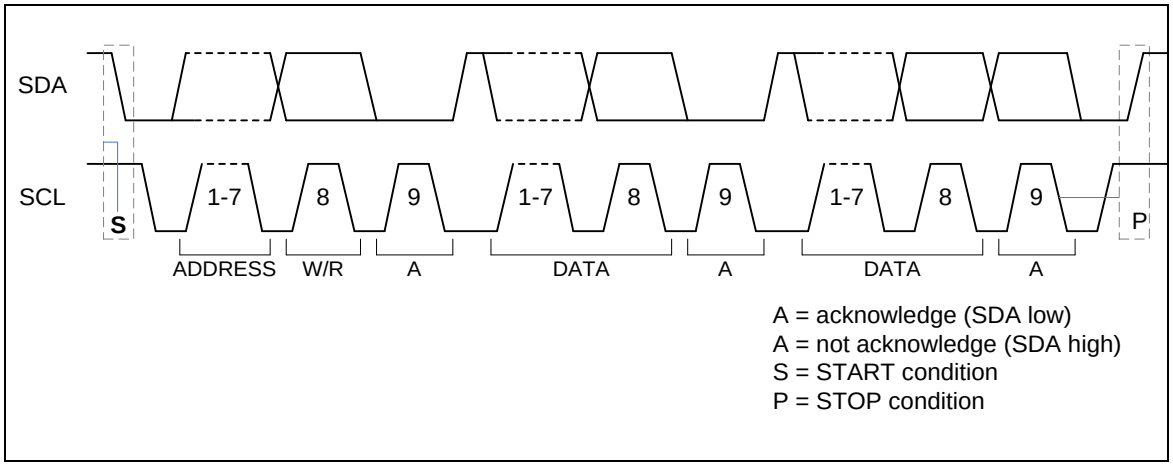


Figure 6.15-2 I²C Protocol

6.15.3.2 Data transfer on the I²C bus

The following figure shows a master transmits data to slave. A master addresses a slave with a 7-bit address and 1-bit write index to denote master wants to transmit data to slave. The master keep transmitting data after slave returns acknowledge to master.

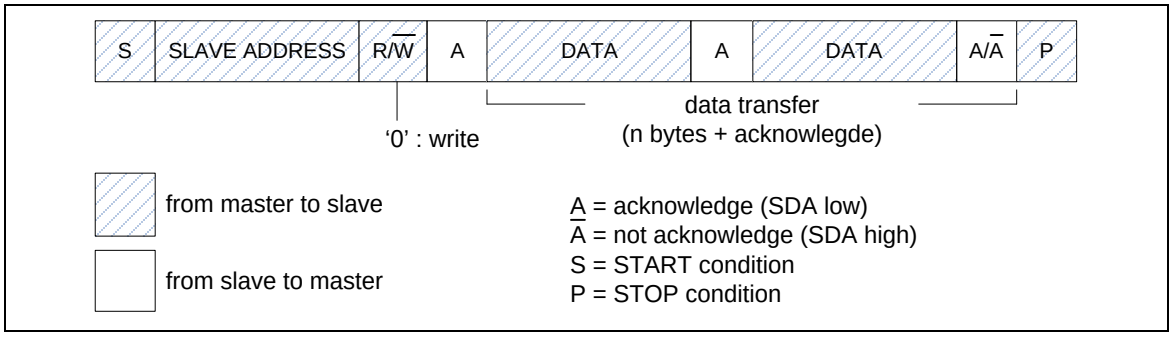


Figure 6.15-3 Master Transmits Data to Slave

The following figure shows a master read data from slave. A master addresses a slave with a 7-bit address and 1-bit read index to denote master wants to read data from slave. The slave will start transmitting data after slave returns acknowledge to master.

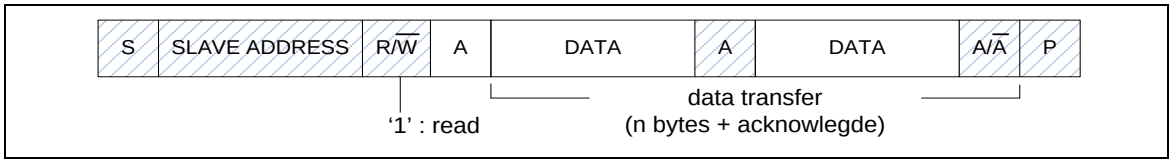


Figure 6.15-4 Master Reads Data from Slave

6.15.3.3 START or Repeated START signal and STOP signal

When the bus is free/idle, meaning no master device is engaging the bus (both SCL and SDA lines are high), a master can initiate a transfer by sending a START signal. A START signal, usually referred to as the S-bit, is defined as a HIGH to LOW transition on the SDA line while SCL is HIGH. The START signal denotes the beginning of a new data transfer.

A Repeated START (Sr) is no STOP signal between two START signals. The master uses this method to communicate with another slave or the same slave in a different transfer direction (e.g. from writing to a device to reading from a device) without releasing the bus.

The master can terminate the communication by generating a STOP signal. A STOP signal, usually referred to as the P-bit, is defined as a LOW to HIGH transition on the SDA line while SCL is HIGH.

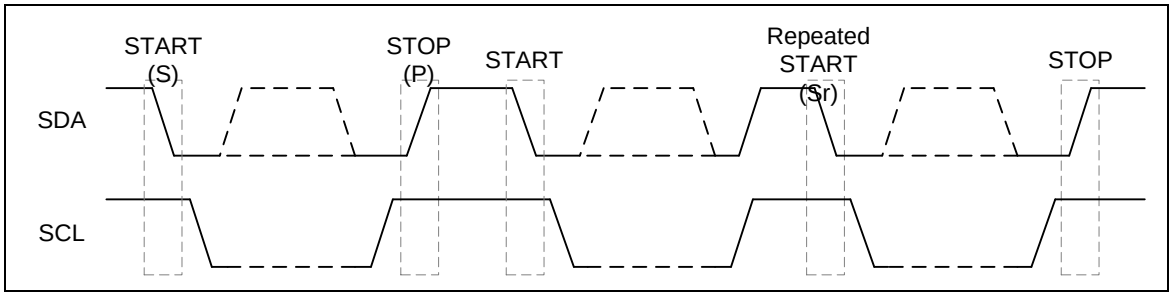


Figure 6.15-5 START and STOP Condition

6.15.3.4 Slave Address Transfer

The first byte of data transferred by the master immediately after the START signal is the slave address. This is a 7-bit calling address followed by a RW bit. The RW bit signals the slave the data transfer direction. No two slaves in the system can have the same address. Only the slave with an address that matches the one transmitted by the master will respond by returning an acknowledge bit by pulling the SDA low at the 9th SCL clock cycle.

6.15.3.5 Data Transfer

Once successful slave addressing has been achieved, the data transfer can proceed on a byte-by-byte basis in the direction specified by the RW bit sent by the master. Each transferred byte is followed by an acknowledge bit on the 9th SCL clock cycle. If the slave signals a Not Acknowledge (NACK), the master can generate a STOP signal to abort the data transfer or generate a Repeated START signal and start a new transfer cycle.

If the master, as the receiving device, does Not Acknowledge (NACK) the slave, the slave releases the SDA line for the master to generate a STOP or Repeated START signal.

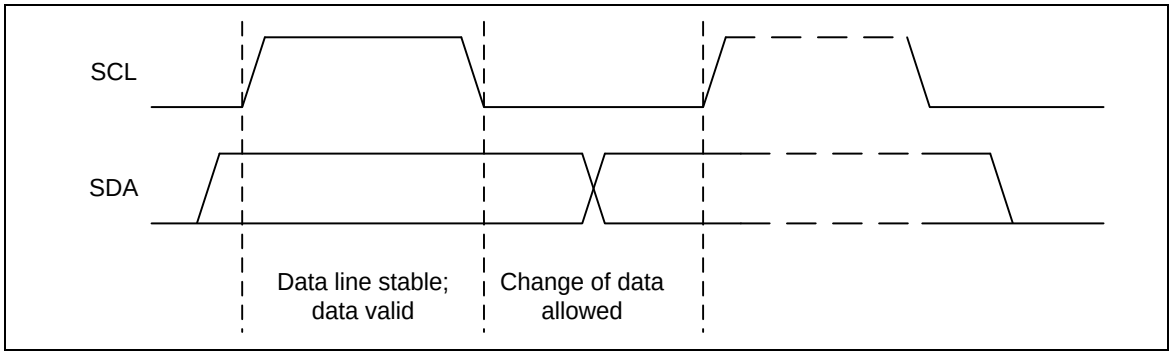


Figure 6.15-6 Bit Transfer on I²C Bus

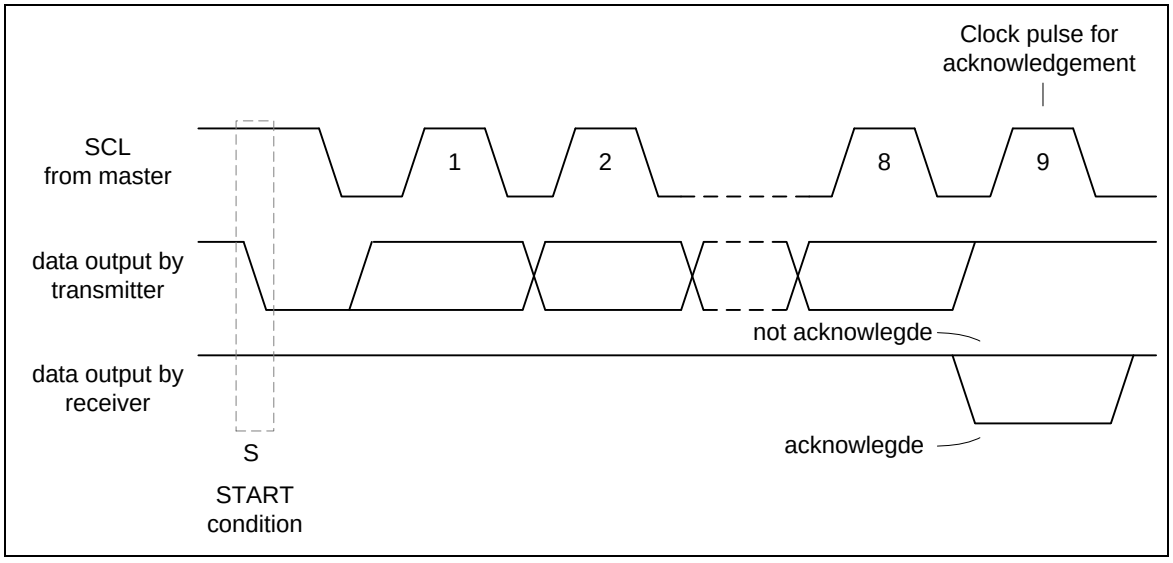


Figure 6.15-7 Acknowledge on I²C Bus

6.15.4 Operation Modes

The on-chip I²C ports support three operation modes, Master, Slave, and General Call Mode.

In a given application, I²C port may operate as a master or as a slave. In Slave mode, the I²C port hardware looks for its own slave address and the general call address. If one of these addresses is detected, and if the slave is willing to receive or transmit data from/to master(by setting the AA bit), acknowledge pulse will be transmitted out on the 9th clock, hence an interrupt is requested on both master and slave devices if interrupt is enabled. When the microcontroller wishes to become the bus master, hardware waits until the bus is free before entering Master mode so that a possible slave action is not be interrupted. If bus arbitration is lost in Master mode, I²C port switches to Slave mode immediately and can detect its own slave address in the same serial transfer.

To control the I²C bus transfer in each mode, user needs to set I2C_CTL, I2C_DAT registers according to current status code of I2C_STATUS register. In other words, for each I²C bus action, user needs to check current status by I2C_STATUS register, and then set I2C_CTL, I2C_DAT registers to take bus action. Finally, check the response status by I2CSTATUS.

The bits, STA, STO and AA in I2C_CTL register are used to control the next state of the I²C hardware after SI flag of I2C_CTL [3] register is cleared. Upon completion of the new action, a new status code will be updated in I2C_STATUS register and the SI flag of I2C_CTL register will be set. If the I²C interrupt control bit EI (I2C_CTL [7]) is set, appropriate action or software branch of the new status code can be performed in the Interrupt service routine.

The following figure shows the current I²C status code is 0x08, and then set I2C_DATA=SLA+W and (STA,STO,SI,AA) = (0,0,1,x) to send the address to I²C bus. If a slave on the bus matches the address and response ACK, the I2C_STATUS will be updated by status code 0x18.

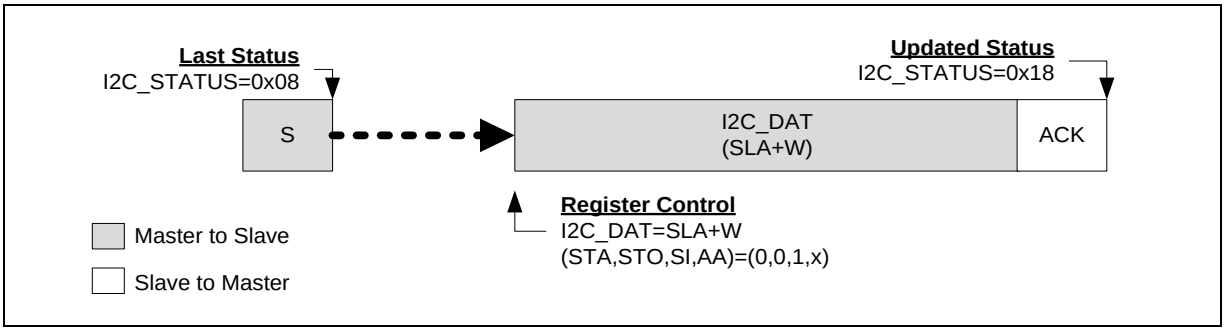


Figure 6.15-8 Control I²C Bus according to Current I²C Status

6.15.4.1 Master Mode

In below figures, all possible protocols for I²C master are shown. User needs to follow proper path of the flow to implement required I²C protocol.

In other words, user can send a START signal to bus and I²C will be in Master Transmitter mode (Figure 6.15-9) or Master receiver mode (Figure 6.15-10) after START signal has been sent successfully and new status code would be 0x08. Followed by START signal, user can send slave address, read/write bit, data and Repeat START, STOP to perform I²C protocol.

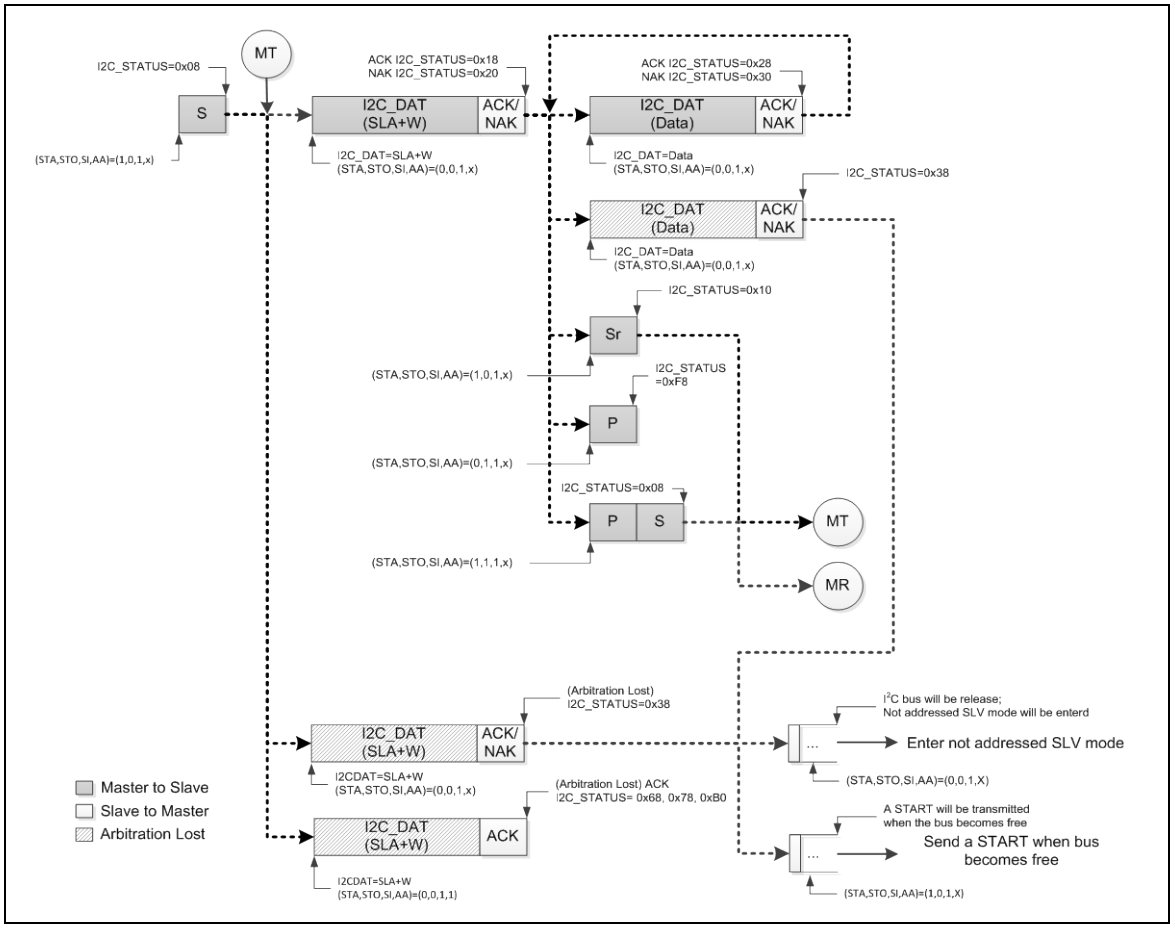


Figure 6.15-9 Master Transmitter Mode Control Flow

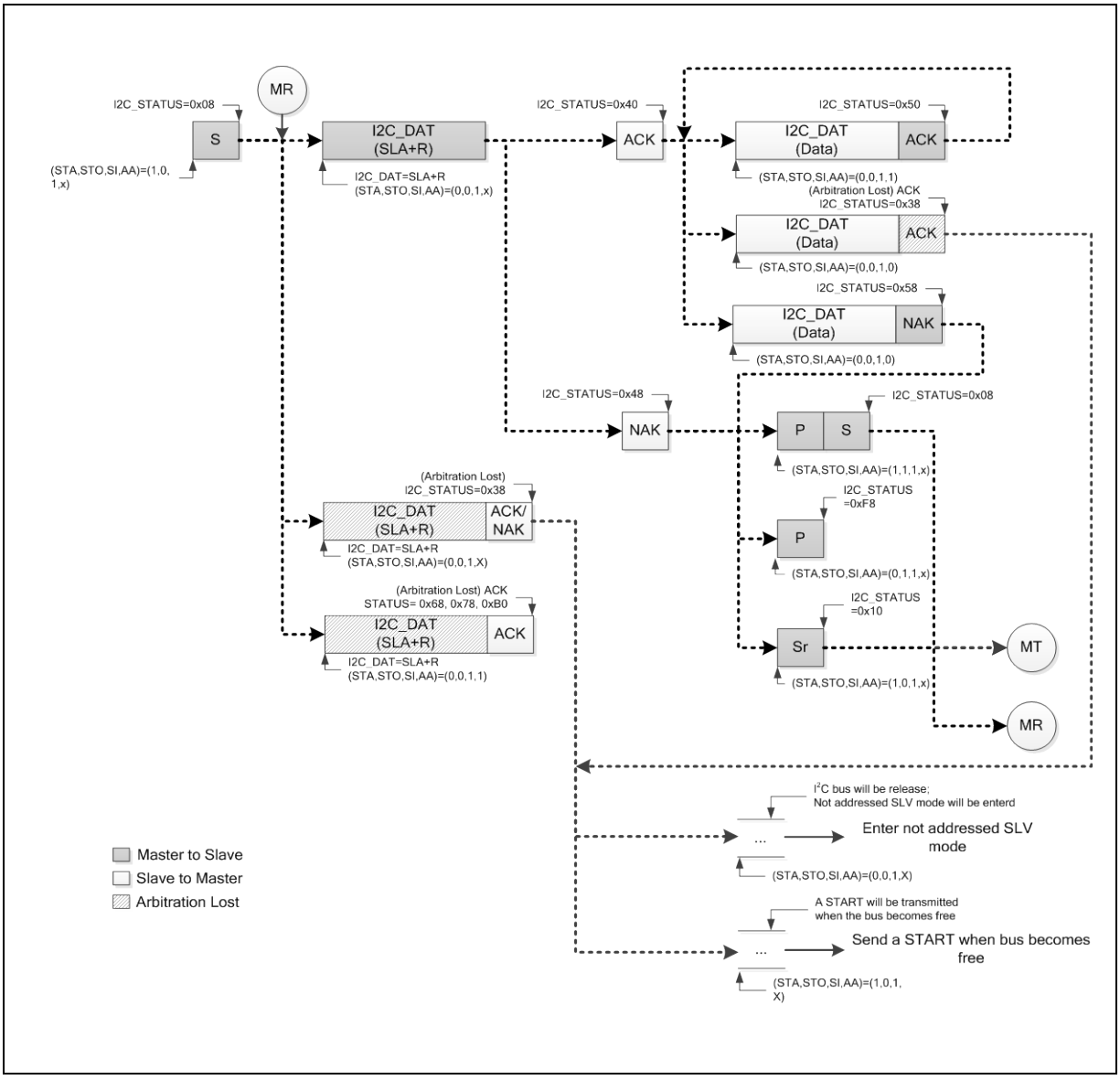


Figure 6.15-10 Master Receiver Mode Control Flow

If the I²C is in Master mode and gets arbitration lost, the status code will be 0x38. In status 0x38, user may set (STA, STO, SI, AA) = (1, 0, 1, X) to send START to re-start Master operation when bus become free. Otherwise, user may set (STA, STO, SI, AA) = (0, 0, 1, X) to release I²C bus and enter not addressed Slave mode.

6.15.4.2 Slave Mode

When reset default, I²C is not addressed and will not recognize the address on I²C bus. User can set slave address by I2CADDRx and set (STA, STO, SI, AA) = (0, 0, 1, 1) to let I²C recognize the address sent by master. The following figure shows all the possible flow for I²C in Slave mode.

If bus arbitration is lost in Master mode, I²C port switches to Slave mode immediately and can detect its own slave address in the same serial transfer. If the detected address is SLA+W

(Master want to write data to Slave) after arbitration lost, the status code is 0x68. If the detected address is SLA+R (Master want to read data from Slave) after arbitration lost, the status code is 0xB0.

Note: During I²C communication, the SCL clock will be released when writing '1' to clear SI (I2C_CTL[3]) flag in Slave mode.

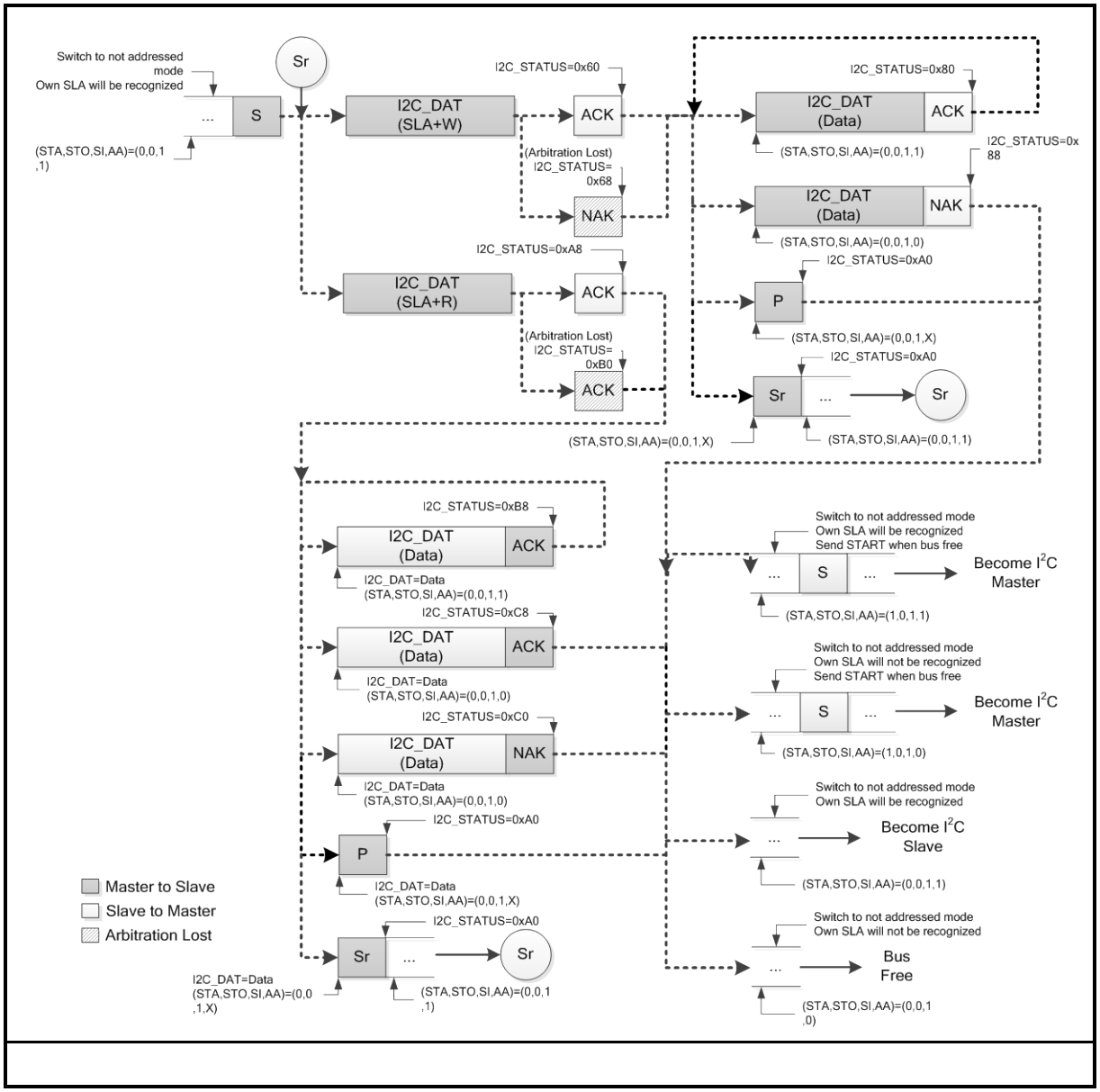


Figure 6.15-11 Slave Mode Control Flow

If I²C is still receiving data in addressed Slave mode but got a STOP or Repeat START, the status code will be 0xA0. User could follow the action for status code 0x88 as shown in the above figure when getting 0xA0 status.

If I²C is still transmitting data in addressed Slave mode but got a STOP or Repeat START, the status code will be 0xA0. User could follow the action for status code 0xC8 as shown in the above figure when getting 0xA0 status.

Note: After slave gets status of 0x88, 0xC8, 0xC0 and 0xA0, slave can switch to not address mode and own SLA will not be recognized. If entering this status, slave will not receive any I²C signal or address from master. At this status, I²C should be reset to leave this status.

6.15.4.3 General Call (GC) Mode

If the GC bit (I2C_ADDRn [0]) is set, the I²C port hardware will respond to General Call address (00H). User can clear GC bit to disable general call function. When the GC bit is set and the I²C is in Slave mode, it can receive the general call address by 0x00 after master send general call address to I²C bus, then it will follow status of GC mode.

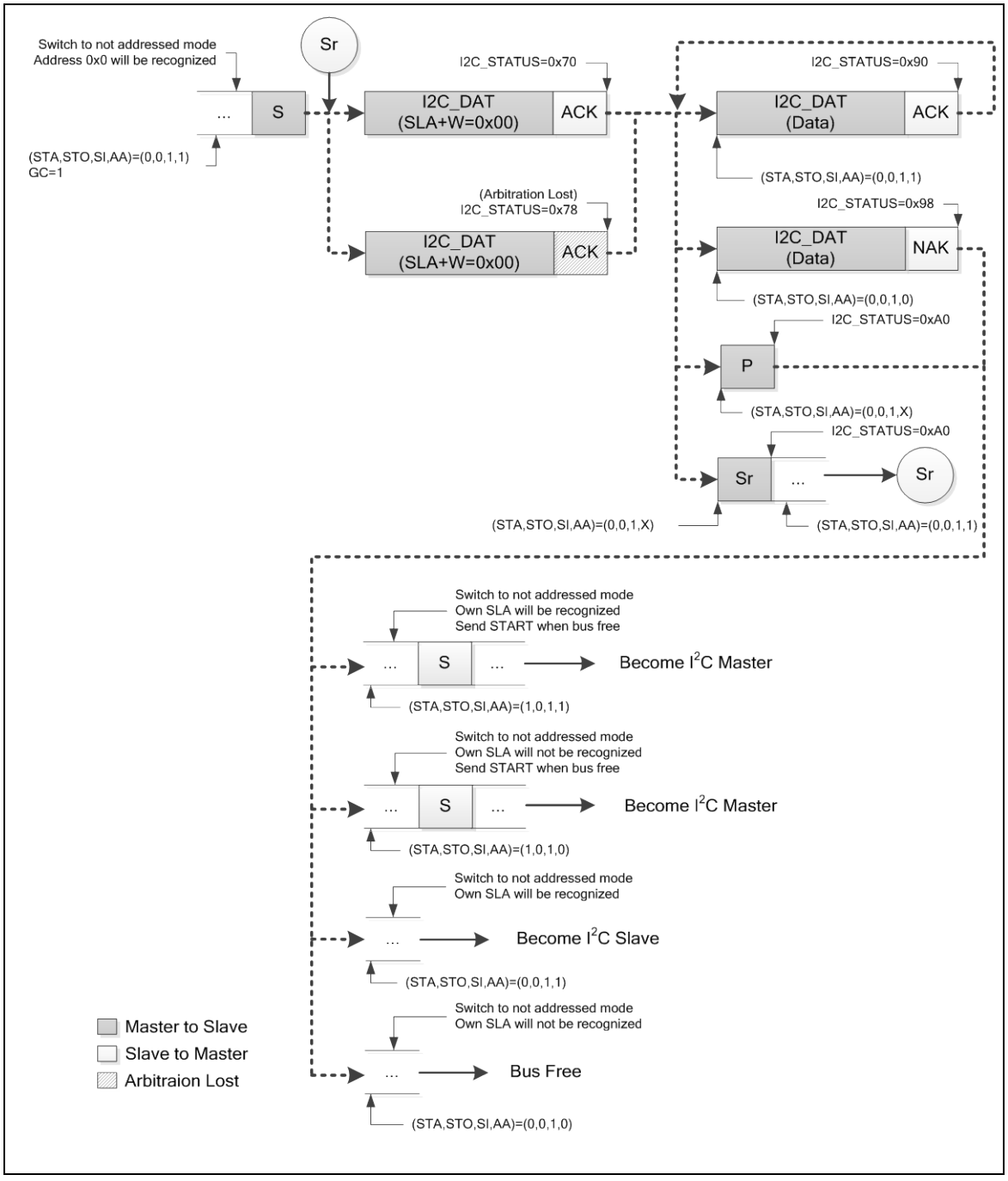


Figure 6.15-12 GC Mode

If I²C is still receiving data in GC mode but got a STOP or Repeat START, the status code will be 0xA0. User could follow the action for status code 0x98 in above figure when getting 0xA0 status.

Note: After slave gets status of 0x98 and 0xA0, slave can switch to not address mode and own SLA will not be recognized. If entering this status, slave will not receive any I²C signal or address from master. At this time, I²C controller should be reset to leave this status.

6.15.4.4 Multi-Master

In some applications, there are two or more masters on the same I²C bus to access slaves, and the masters may transmit data simultaneously. The I²C supports multi-master by including collision detection and arbitration to prevent data corruption.

- When I2C_STATUS = 0x38, an “Arbitration Lost” is received. Arbitration lost event maybe occur during the send START bit, data bits or STOP bit. User could set (STA, STO, SI, AA) = (1, 0, 1, X) to send START again when bus free, or set (STA, STO, SI, AA) = (0, 0, 1, X) to send STOP to back to not addressed Slave mode.
- When I2C_STATUS = 0x00, a “Bus Error” is received. To recover I²C bus from a bus error, STO should be set and SI should be cleared, and then STO is cleared to release bus.
  - Set (STA, STO, SI, AA) = (0, 1, 1, X) to stop current transfer
  - Set (STA, STO, SI, AA) = (0, 0, 1, X) to release bus

6.15.4.5 Example for Random Read on EEPROM

The following steps are used to configure the I²C related registers when using I²C to read data from EEPROM.

1. Set the multi-function pin in the “SYS_GPD_MFP” register as SCL (PD.8) and SDA (PD.9) pins.
2. Enable I²C APB clock, I2C0CKEN=1 in the “CLK_APBCLK0[8]” register.
3. Set I2C0_RST=1 to reset I²C controller then set I²C controller to normal operation, I2C0RST=0 in the “SYS_IPRST1[8]” register.
4. Set I2CEN (I2C_CTL[6])=1 to enable I²C controller in the “I2C_CTL” register.
5. Give I²C clock a divided register value for I²C clock rate in the “I2C_CLKDIV”.
6. Set SETENA=0x00040000 in the “NVIC_ISER” register to set I²C IRQ.
7. Set INTEN (I2C_CTL[7])=1 to enable I²C Interrupt in the “I2C_CTL” register.
8. Set I²C address registers which are “I2C_ADDR0~I2C_ADDR3”.

Random read operation is one of the methods of access EEPROM. The method allows the master to access any address of EEPROM space. The following figure shows the EEPROM random read operation.

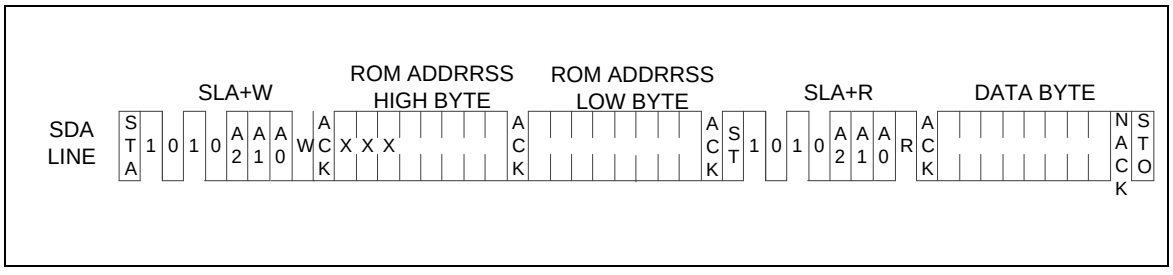


Figure 6.15-13 EEPROM Random Read

The following figure shows how to use I²C controller to implement the protocol of EEPROM random read.

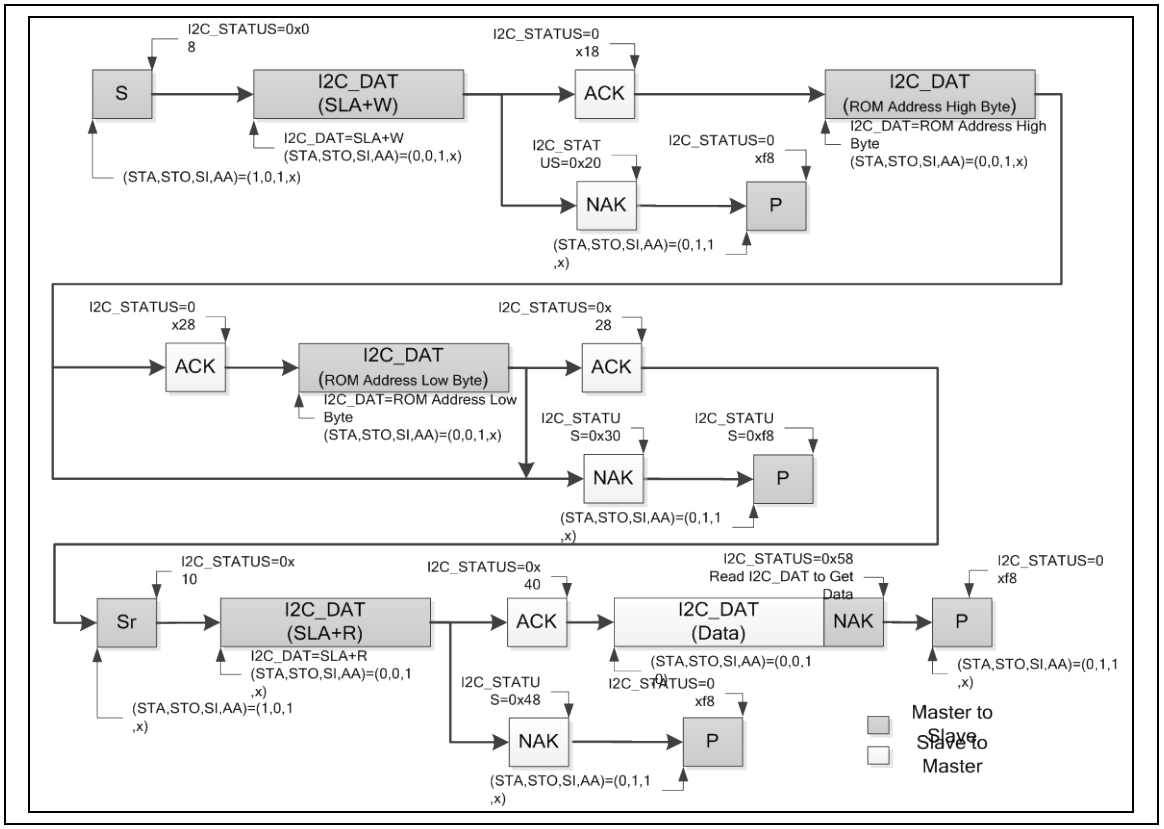


Figure 6.15-14 Protocol of EEPROM Random Read

The I²C controller sends START to bus to be a master. Then it sends a SLA+W (Slave address + Write bit) to EERPOM followed by two bytes data address to set the EEPROM address to read. Finally, a Repeat START followed by SLA+R is sent to read the data from EEPROM.

6.15.5 Protocol Registers

The CPU interfaces to the I²C port through the following thirteen special function registers: I2C_CTL (control register), I2C_STATUS (status register), I2C_DAT (data register), I2C_ADDRn (address registers, n=0~3), I2C_ADDRMSKn (address mask registers, n=0~3), I2C_CLKDIV (clock rate register) and I2C_TOCTL (Time-out counter register). All bit 31~ bit 8 of these I²C special function registers are reserved. These bits do not have any functions and are all zero if read back.

When I²C port is enabled by setting I2CEN (I2C_CTL [6]) to high, the internal states will be controlled by I2C_CTL and I²C logic hardware. Once a new status code is generated and stored in I2C_STATUS, the I²C Interrupt Flag bit SI (I2C_CTL [3]) will be set automatically. If the Enable Interrupt bit INTEN (I2C_CTL [7]) is set high at this time, the I²C interrupt will be generated. The bit field I2C_STATUS[7:3] stores the internal state code, the lowest 3 bits of I2C_STATUS are always zero and the content keeps stable until SI is cleared by software. The base address is 0x4008_0000 ~ 0x4008_4000.

6.15.5.1 Address Registers (I2CADDR)

The I²C port is equipped with four slave address registers I2C_ADDRn (n=0~3). The contents of the register are irrelevant when I²C is in Master mode. In the slave mode, the bit field I2C_ADDRn[7:1] must be loaded with the chip's own slave address. The I²C hardware will react if the contents of I2C_ADDRn are matched with the received slave address.

The I²C ports support the "General Call" function. If the GC bit (I2C_ADDRn [0]) is set the I²C port hardware will respond to General Call address (00H). Clear GC bit to disable general call function.

When GC bit is set and the I²C is in Slave mode, it can receive the general call address by 00H after Master send general call address to I²C bus, then it will follow status of GC mode.

I²C bus controllers support multiple address recognition with four address mask registers I2C_ADDRMSKn (n=0~3). When the bit in the address mask register is set to one, it means the received corresponding address bit is don't-care. If the bit is set to zero, that means the received corresponding register bit should be exact the same as address register.

6.15.5.2 Data Register (I2C_DAT)

This register contains a byte of serial data to be transmitted or a byte which just has been received. The CPU can read from or write to this 8-bit (DAT [7:0]) directly while it is not in the process of shifting a byte. When I²C is in a defined state and the serial interrupt flag (SI) is set. Data in DAT [7:0] remains stable as long as SI bit is set. While data is being shifted out, data on the bus is simultaneously being shifted in; DAT [7:0] always contains the last data byte present on the bus. Thus, in the event of arbitration lost, the transition from master transmitter to slave receiver is made with the correct data in DAT [7:0].

DAT [7:0] and the acknowledge bit form a 9-bit shift register, the acknowledge bit is controlled by the I²C hardware and cannot be accessed by the CPU. Serial data is shifted through the acknowledge bit into DAT [7:0] on the rising edges of serial clock pulses on the SCL line. When a byte has been shifted into DAT [7:0], the serial data is available in DAT [7:0], and the acknowledge bit (ACK or NACK) is returned by the control logic during the ninth clock pulse. Serial data is shifted out from DAT [7:0] on the falling edges of SCL clock pulses, and is shifted into DAT [7:0] on the rising edges of SCL clock pulses.

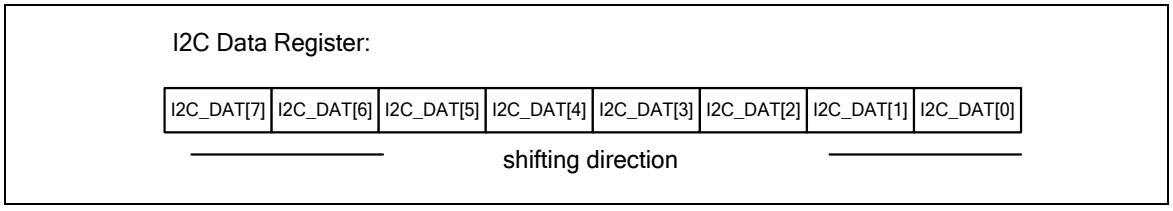


Figure 6.15-15 I²C Data Shifting Direction

6.15.5.3 Control Register (I2C_CTL)

The CPU can read from and write to this 8-bit field of I2C_CTL [7:0] directly. Two bits are affected by hardware: the SI bit is set when the I²C hardware requests a serial interrupt, and the STO bit is cleared when a STOP condition is present on the bus. The STO bit is also cleared when I2CEN = 0.

INTEN Enable Interrupt.

I2CEN Set to enable I²C serial function controller. When I2CEN=1 the I²C serial function enables. The Multi-function pin function of SDA and SCL must be set to I²C function.

STA I²C START Control Bit. Setting STA to logic 1 to enter Master mode, the I²C hardware sends a START or repeat START condition to bus when the bus is free.

STO I²C STOP Control Bit. In Master mode, setting STO to transmit a STOP condition to bus then I²C hardware will check the bus condition if a STOP condition is detected this flag will be cleared by hardware automatically. In a slave mode, setting STO resets I²C hardware to the defined “not addressed” slave mode. This means it is NO LONGER in the slave receiver mode to receive data from the master transmit device.

SI I²C Interrupt Flag. When a new I²C state is present in the I2C_STATUS register, the SI flag is set by hardware, and if bit INTEN (I2C_CTL [7]) is set, the I²C interrupt is requested. SI must be cleared by software. Clear SI by writing 1 to this bit. All states are listed in I2C_STATUS Register section.

AA Assert Acknowledge Control Bit. When AA=1 prior to address or data received, an acknowledged (low level to SDA) will be returned during the acknowledge clock pulse on the SCL line when 1.) A slave is acknowledging the address sent from master, 2.) The receiver devices are acknowledging the data sent by transmitter. When AA=0 prior to address or data received, a Not acknowledged (high level to SDA) will be returned during the acknowledge clock pulse on the SCL line.

6.15.5.4 Status Register (I2C_STATUS)

I2C_STATUS [7:0] is an 8-bit read only register. The three least significant bits are always 0. The bit field I2C_STATUS [7:3] contain the status code. There are 26 possible status codes, All states are listed in section 5.6.6. When I2C_STATUS [7:0] contains F8H, no serial interrupt is requested. All other I2C_STATUS [7:3] values correspond to defined I²C states. When each of these states is entered, a status interrupt is requested (SI = 1). A valid status code is present in I2C_STATUS[7:3] one cycle after SI is set by hardware and is still present one cycle after SI has been reset by software.

In addition, state 00H stands for a Bus Error. A Bus Error occurs when a START or STOP condition is present at an illegal position in the format frame. Examples of illegal positions are during the serial transfer of an address byte, a data byte or an acknowledge bit. To recover I²C from bus error, STO should be set and SI should be clear to enter not addressed slave mode. Then clear STO to release bus and to wait new communication. I²C bus cannot recognize stop condition during this action when bus error occurs.

Master Mode		**Slave Mode**	
STATUS	**Description**	**STATUS**	**Description**
0x08	Start	0xA0	Slave Transmit Repeat Start or Stop
0x10	Master Repeat Start	0xA8	Slave Transmit Address ACK
0x18	Master Transmit Address ACK	0xB0	Slave Transmit Arbitration Lost
0x20	Master Transmit Address NACK	0xB8	Slave Transmit Data ACK
0x28	Master Transmit Data ACK	0xC0	Slave Transmit Data NACK
0x30	Master Transmit Data NACK	0xC8	Slave Transmit Last Data ACK
0x38	Master Arbitration Lost	0x60	Slave Receive Address ACK
0x40	Master Receive Address ACK	0x68	Slave Receive Arbitration Lost

0x48	Master Receive Address NACK	0x80	Slave Receive Data ACK
0x50	Master Receive Data ACK	0x88	Slave Receive Data NACK
0x58	Master Receive Data NACK	0x70	GC mode Address ACK
0x00	Bus error	0x78	GC mode Arbitration Lost
-	-	0x90	GC mode Data ACK
-	-	0x98	GC mode Data NACK
0xF8	Bus Released **Note:** Status "0xF8" exists in both master/slave modes, and it won't raise interrupt.		

Table 6.15-1 I²C Status Code Description Table

6.15.5.5 I2C Clock Baud Rate Bits (I2C_CLKDIV)

The data baud rate of I²C is determines by I2C_CLKDIV [7:0] register when I²C is in Master mode. It is not important when I²C is in a slave mode. In the slave modes, I²C will automatically synchronize with any clock frequency from master I²C device.

The data baud rate of I²C setting is Data Baud Rate of I²C = (system clock) / (4x (I2C_CLKDIV [7:0] +1)). If system clock = 16 MHz, the I2C_CLKDIV [7:0] = 40 (28H), so data baud rate of I²C = 16 MHz/ (4x (40 +1)) = 97.5 Kbits/sec.

6.15.5.6 I²C Time-out Counter Register (I2C_TOCTL)

There is a 14-bit time-out counter which can be used to deal with the I²C bus hang-up. If the time-out counter is enabled, the counter starts up counting until it overflows (TOIF (I2C_TOCTL[0])=1) and generates I²C interrupt to CPU or stops counting by clearing TOCEN (I2C_TOCTL[2]) to 0. When the time-out counter is enabled, setting flag SI (I2C_CTL[3]) to high will reset counter and re-start up counting after SI is cleared. If I²C bus hangs up, it causes the I2C_STATUS and flag SI are not updated for a period, the 14-bit time-out counter may overflow and acknowledge CPU the I²C interrupt. Refer to the following figure for the 14-bit time-out counter. User may write 1 to clear TOIF to 0.

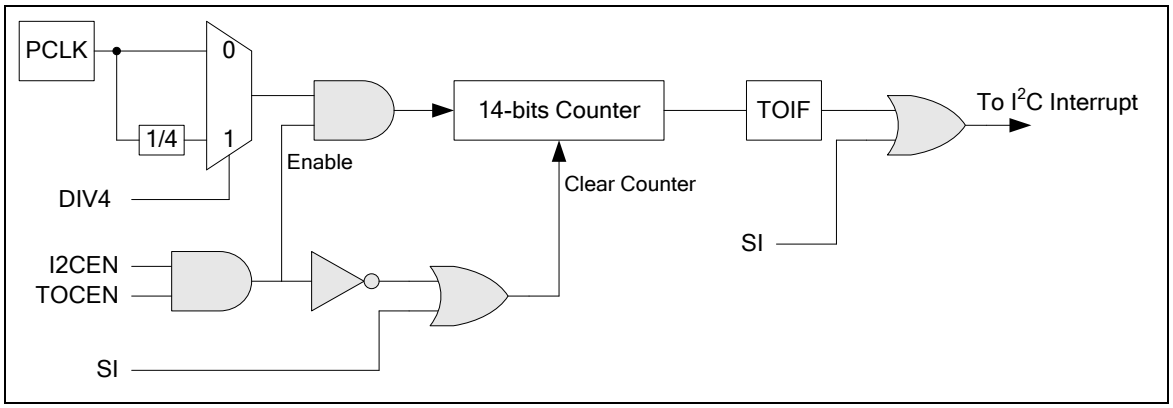


Figure 6.15-16 I²C Time-out Count Block Diagram

*6.15.5.7 The I²C wake-up control Register (**I2C_WKCTL**)*

When entering sleep mode, other I²C master can wake up our chip by addressing our I²C device, user must configure the related setting before entering sleep mode.

WKUPEN enables I²C wake-up function

*6.15.5.8 The I²C wake-up status Register (**I2C_WKSTS**)*

When system is woken up by other I²C master device, WKIF is set to indicate this event

WKIF Wake-up interrupt flag

6.15.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
I2C Base Address: **I2Cx_BA = 0x4008_0000 + x * 0x1000** **x=0,1..4**				
I2C_CTL	I2Cx_BA+0x00	R/W	I^2C Control Register	0x0000_0000
I2C_ADDR0	I2Cx_BA+0x04	R/W	I^2C Slave Address Register0	0x0000_0000
I2C_DAT	I2Cx_BA+0x08	R/W	I^2C Data Register	0x0000_0000
I2C_STATUS	I2Cx_BA+0x0C	R	I^2C Status Register	0x0000_00F8
I2C_CLKDIV	I2Cx_BA+0x10	R/W	I^2C Clock Divided Register	0x0000_0000
I2C_TOCTL	I2Cx_BA+0x14	R/W	I^2C Time-out Control Register	0x0000_0000
I2C_ADDR1	I2Cx_BA+0x18	R/W	I^2C Slave Address Register1	0x0000_0000
I2C_ADDR2	I2Cx_BA+0x1C	R/W	I^2C Slave Address Register2	0x0000_0000
I2C_ADDR3	I2Cx_BA+0x20	R/W	I^2C Slave Address Register3	0x0000_0000
I2C_ADDRMSK0	I2Cx_BA+0x24	R/W	I^2C Slave Address Mask Register0	0x0000_0000
I2C_ADDRMSK1	I2Cx_BA+0x28	R/W	I^2C Slave Address Mask Register1	0x0000_0000
I2C_ADDRMSK2	I2Cx_BA+0x2C	R/W	I^2C Slave Address Mask Register2	0x0000_0000
I2C_ADDRMSK3	I2Cx_BA+0x30	R/W	I^2C Slave Address Mask Register3	0x0000_0000
I2C_WKCTL	I2Cx_BA+0x3C	R/W	I^2C Wake-up Control Register	0x0000_0000
I2C_WKSTS	I2Cx_BA+0x40	R/W	I^2C Wake-up Status Register	0x0000_0000

6.15.7 Register Description

I²C Control Register (I2C_CTL)

Register	Offset	R/W	Description	Reset Value
I2C_CTL	I2Cx_BA+0x00	R/W	I²C Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
INTEN	I2CEN	STA	STO	SI	AA	Reserved	

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**INTEN**	**I²C Interrupt Enable Bit** 0 = I²C interrupt Disabled. 1 = I²C interrupt Enabled.
[6]	**I2CEN**	**I²C Controller Enable Bit** 0 = Disabled. 1 = Enabled. Set to enable I²C serial function controller. When I2CEN=1 the I²C serial function enables. The multi-function pin function of SDA and SCL must set to I²C function first.
[5]	**STA**	**I²C START Control** Setting STA to logic 1 to enter Master mode, the I²C hardware sends a START or repeat START condition to bus when the bus is free.
[4]	**STO**	**I²C STOP Control** In Master mode, setting STO to transmit a STOP condition to bus then I²C hardware will check the bus condition if a STOP condition is detected this bit will be cleared by hardware automatically. In a slave mode, setting STO resets I²C hardware to the defined “not addressed” slave mode. This means it is NO LONGER in the slave receiver mode to receive data from the master transmit device.
[3]	**SI**	**I²C Interrupt Flag** When a new I²C state is present in the I2C_STATUS register, the SI flag is set by hardware, and if bit INTEN (I2C_CTL [7]) is set, the I²C interrupt is requested. SI must be cleared by software. Clear SI by writing 1 to this bit.
[2]	**AA**	**Assert Acknowledge Control** When AA =1 prior to address or data received, an acknowledged (low level to SDA) will be returned during the acknowledge clock pulse on the SCL line when 1.) A slave is acknowledging the address sent from master, 2.) The receiver devices are acknowledging the data sent by transmitter. When AA=0 prior to address or data received, a Not

		acknowledged (high level to SDA) will be returned during the acknowledge clock pulse on the SCL line.
[1:0]	**Reserved**	Reserved.

I²C Data Register (I2C_DAT)

Register	Offset	R/W	Description	Reset Value
I2C_DAT	I2Cx_BA+0x08	R/W	I²C Data Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DAT							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**DAT**	**I²C Data Bits** Bit [7:0] is located with the 8-bit transferred data of I²C serial port.

I²C Status Register (I2C_STATUS)

Register	Offset	R/W	Description	Reset Value
I2C_STATUS	I2Cx_BA+0x0C	R	I²C Status Register	0x0000_00F8

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
STATUS[7:3]					0	0	0

Bits	Description	
[31:8]	Reserved	Reserved.
[7:0]	STATUS	**I²C Status Bits** The status register of I²C: The three least significant bits are always 0. The five most significant bits contain the status code. Refer to section 6.15.5.4 for detail description.

I²C Clock Divided Register (I2C_CLKDIV)

Register	Offset	R/W	Description	Reset Value
I2C_CLKDIV	I2Cx_BA+0x10	R/W	I²C Clock Divided Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
DIVIDER							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**DIVIDER**	**I²C Clock Divided Bits** The I²C clock rate bits: Data Baud Rate of I²C = (system clock) / (4x (DIVIDER+1)). **Note:** The minimum value of DIVIDER is 4.

I²C Time-out Counter Register (I2C_TOCTL)

Register	Offset	R/W	Description	Reset Value
I2C_TOCTL	I2Cx_BA+0x14	R/W	I²C Time-out Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					TOCEN	TOCDIV4	TOIF

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**TOCEN**	**Time-Out Counter Enable Bit** 0 = Disabled. 1 = Enabled. When Enabled, the 14-bit time-out counter will start counting when SI (I2C_CTL[3]) is clear. Setting flag SI to high will reset counter and re-start up counting after SI is cleared.
[1]	**TOCDIV4**	**Time-Out Counter Input Clock Divided By 4** 0 = Disabled. 1 = Enabled. When Enabled, The time-out period is extend 4 times.
[0]	**TOIF**	**Time-Out Flag** This bit is set by H/W when I²C time-out happened and it can interrupt CPU if I²C interrupt enable bit (INTEN (I2C_CTL[7])) is set to 1. **Note:** Write 1 to clear this bit.

I²C Slave Address Register (I2CADDRx)

Register	Offset	R/W	Description	Reset Value
I2CADDR0	I2Cx_BA+0x04	R/W	I²C Slave Address Register0	0x0000_0000
I2C_ADDR1	I2Cx_BA+0x18	R/W	I²C Slave Address Register1	0x0000_0000
I2C_ADDR2	I2Cx_BA+0x1C	R/W	I²C Slave Address Register2	0x0000_0000
I2C_ADDR3	I2Cx_BA+0x20	R/W	I²C Slave Address Register3	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDR							**GC**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:1]	**ADDR**	**I²C Address Bits** The content of this register is irrelevant when I²C is in Master mode. In the slave mode, the seven most significant bits must be loaded with the chip's own address. The I²C hardware will react if either of the address is matched.
[0]	**GC**	**General Call Function** 0 = General Call Function Disabled. 1 = General Call Function Enabled.

I²C Slave Address Mask Register (I2CADMx)

Register	Offset	R/W	Description	Reset Value
I2C_ADDRMSK0	I2Cx_BA+0x24	R/W	I²C Slave Address Mask Register0	0x0000_0000
I2C_ADDRMSK1	I2Cx_BA+0x28	R/W	I²C Slave Address Mask Register1	0x0000_0000
I2C_ADDRMSK2	I2Cx_BA+0x2C	R/W	I²C Slave Address Mask Register2	0x0000_0000
I2C_ADDRMSK3	I2Cx_BA+0x30	R/W	I²C Slave Address Mask Register3	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDRMSK							**Reserved**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:1]	**ADDRMSK**	**I²C Address Mask Bits** 0 = Mask Disabled (the received corresponding register bit should be exact the same as address register.). 1 = Mask Enabled (the received corresponding address bit is don't care.). I²C bus controllers support multiple address recognition with four address mask register. When the bit in the address mask register is set to one, it means the received corresponding address bit is don't-care. If the bit is set to zero, that means the received corresponding register bit should be exact the same as address register.
[0]	**Reserved**	Reserved.

I²C Wake-up Control Register (I2C_WKCTL)

Register	Offset	R/W	Description	Reset Value
I2C_WKCTL	I2Cx_BA+0x3C	R/W	I²C Wake-up Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**WKEN**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**WKEN**	**I²C Wake-Up Enable Bit** 0 = I²C wake-up function Disabled. 1 = I²C wake-up function Enabled.

I²C Wake-up Status Register (I2C_WKSTS)

Register	Offset	R/W	Description	Reset Value
I2C_WKSTS	I2Cx_BA+0x40	R/W	I²C Wake-up Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**WKIF**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**WKIF**	**I²C Wake-Up Flag** 0 = No wake up occurred. 1 = Wake up from Power-down mode. **Note:** Software can write 1 to clear this bit.

6.16 I²S Controller (I²S)

6.16.1 Overview

The I²S controller consists of IIS protocol to interface with external audio CODEC. Two 8 word deep FIFO for read path and write path respectively and is capable of handling 8/16/24/32 bits word sizes. DMA controller handles the data movement between FIFO and memory.

6.16.2 Features

- Operates as either Master or Slave
- Capable of handling 8, 16, 24 and 32 bits word sizes
- Supports Mono and stereo audio data
- Supports I²S and MSB justified data format
- Provides two 8 word FIFO data buffers, one for transmitting and the other for receiving
- Generates interrupt requests when buffer levels cross a programmable boundary
- Two DMA requests, one for transmitting and the other for receiving

6.16.3 Block Diagram

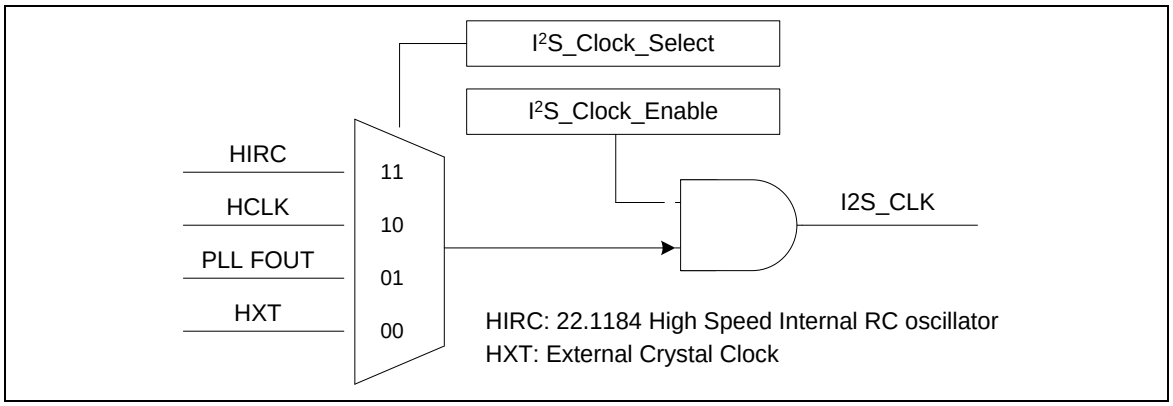


Figure 6.16-1 I²S Clock Control Diagram

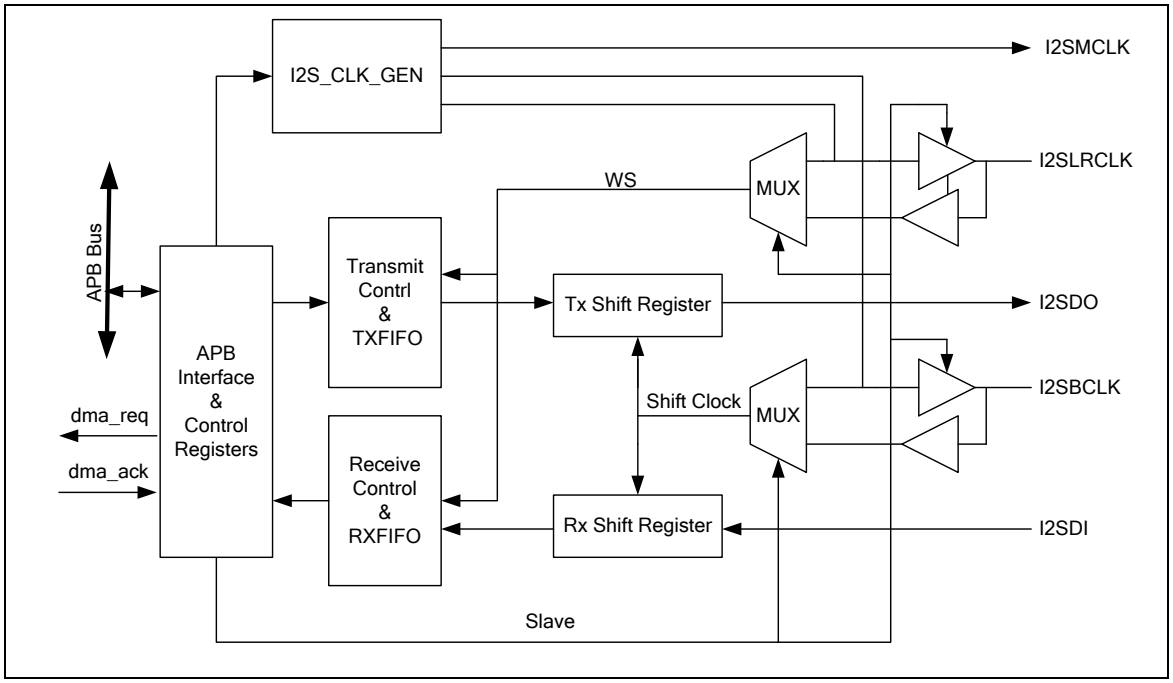


Figure 6.16-2 I²S Controller Block Diagram

6.16.4 Timing Diagram Description

6.16.4.1 I²S Operation

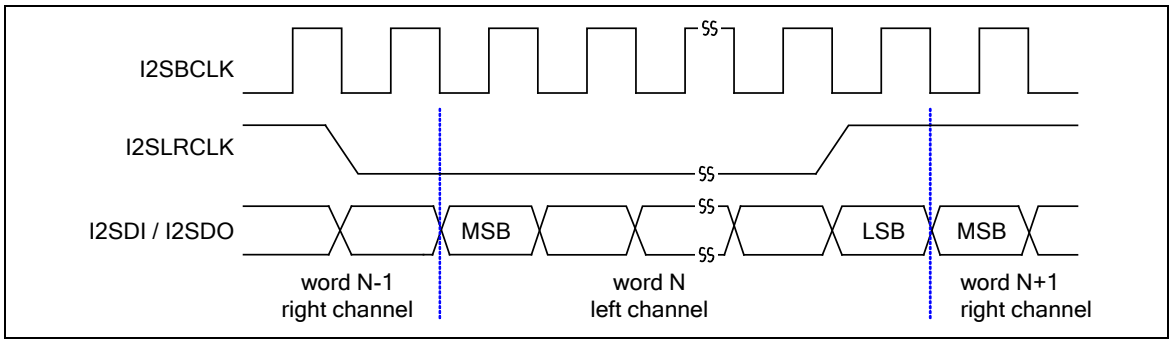


Figure 6.16-3 I²S Bus Timing Diagram (PCM = 0, Format = 0)

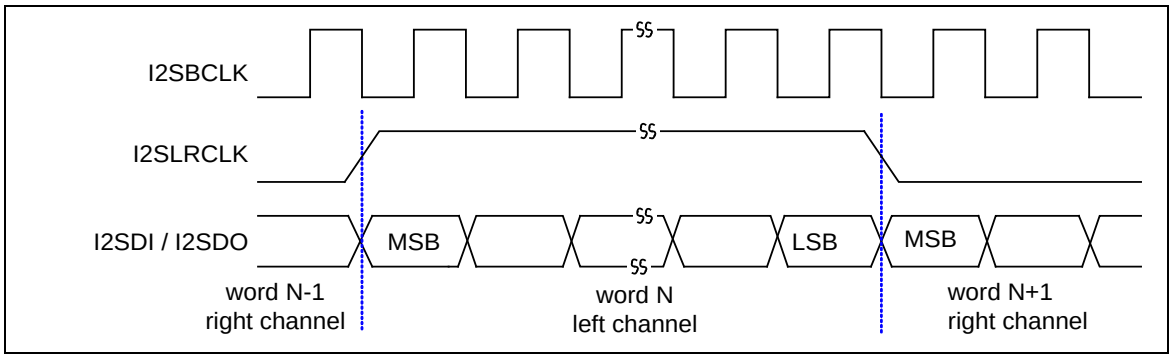


Figure 6.16-4 MSB Justified Timing Diagram (PCM = 0, Format = 1)

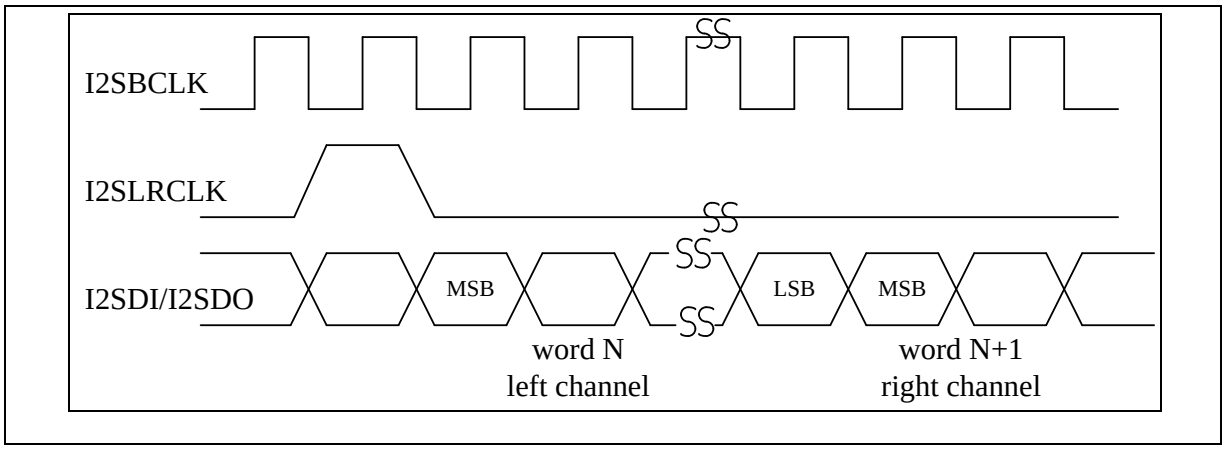


Figure 6.16-5 PCM A Audio Timing Diagram (PCM = 1, Format = 0)

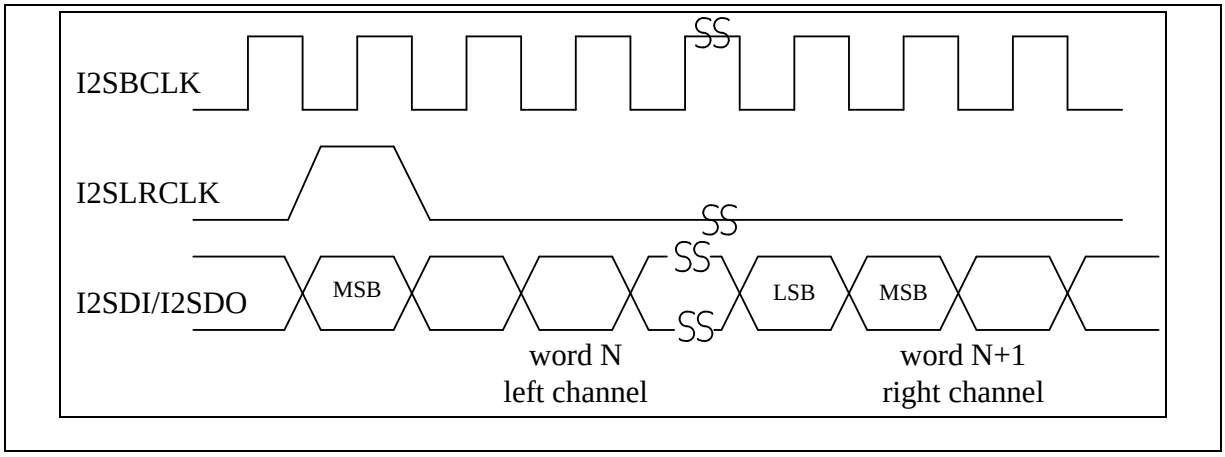


Figure 6.16-6 PCM B Audio Timing Diagram (PCM = 1, Format = 1)

6.16.4.2 FIFO operation

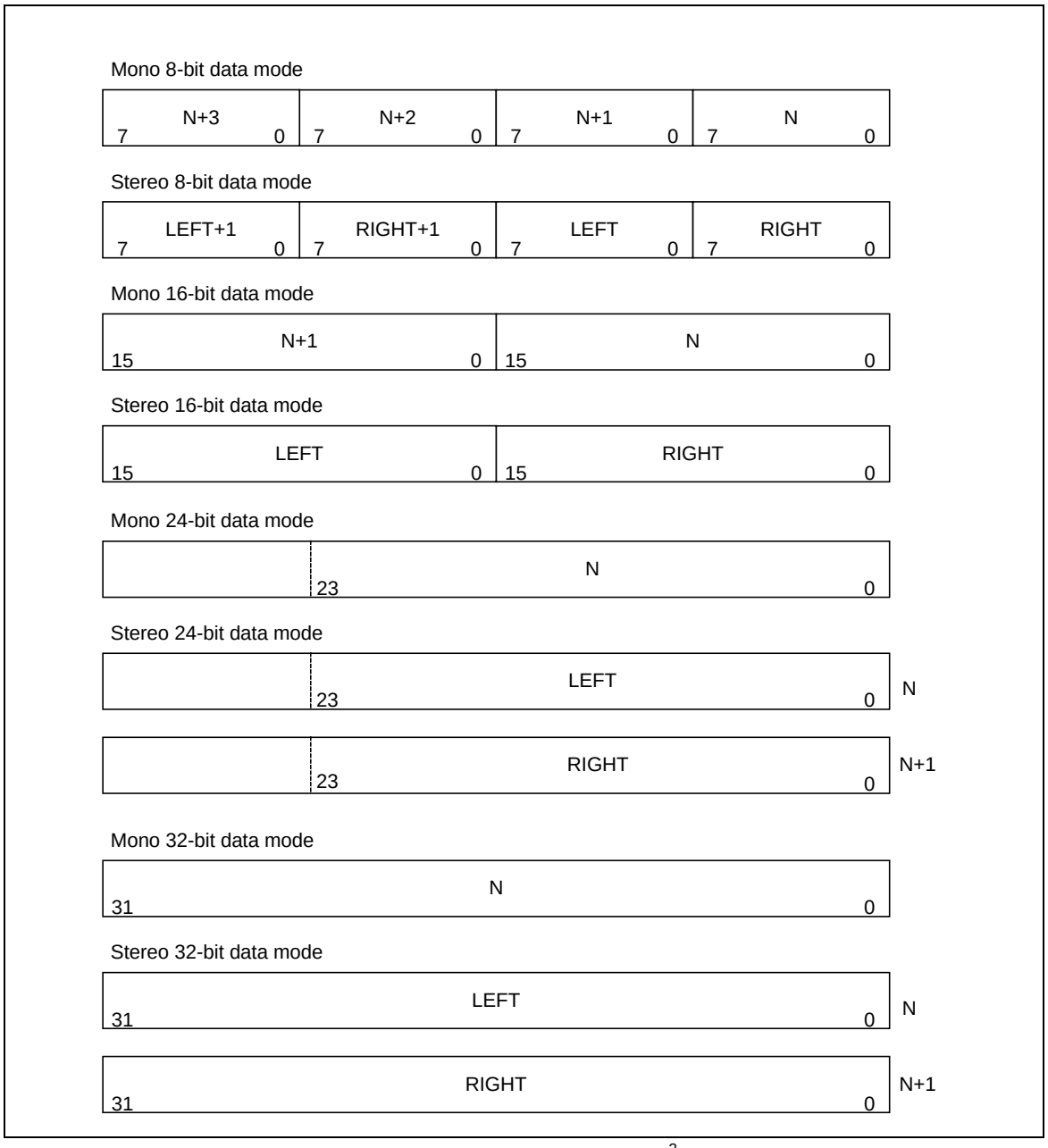


Figure 6.16-7 FIFO Contents for Various I^2S Modes

6.16.5 Functional Description

6.16.5.1 Zero Crossing

When playing the audio by I²S function, the output data comes from the memory by PDMA or by CPU. However, it may result some pop noise if the playing gain level is changed by user at any time. Because, the output data is not zero, and the output data cross the gain change will generate a sharp pop noise. Therefore, the zero_crossing flag will help to reduce this situation. If enable the zero crossing function, hardware will detect the next transfer data is zero or sign change. If the next data is zero or sign change, zero_crossing flag will be set to high, and the output data will be mute automatically, until the flag is cleared by software.

6.16.5.2 PDMA Mode

The I²S function can use PDMA function to access the data. When in transfer mode and enable PDMA function, if TX FIFO is not full, the I²S will generate the request signal and get a data from memory by PDMA IP automatically, until the TX FIFO is full. However, when in receive mode and enable PDMA function, if the TX FIFO is not empty, the I²S will generate the request signal and move a receive data to memory by PDMA automatically, until the RX FIFO is empty. So, using PDMA function will save the CPU loading to service other function.

6.16.5.3 Master/Slave Interface

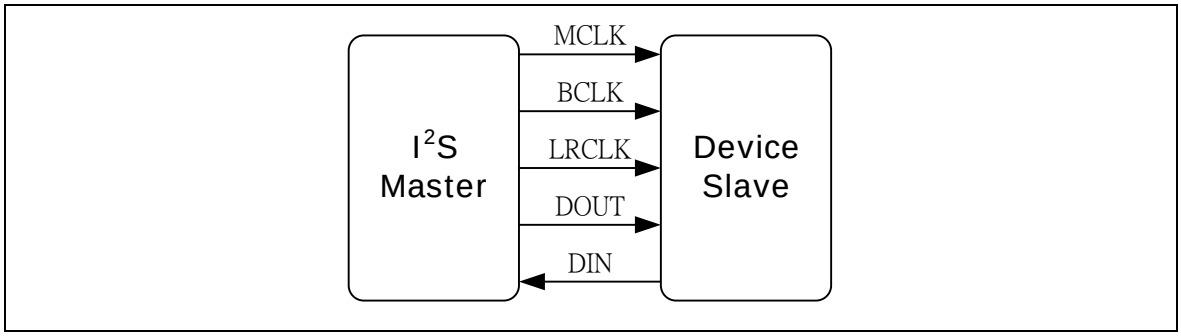


Figure 6.16-8 Master mode Interface Block Diagram

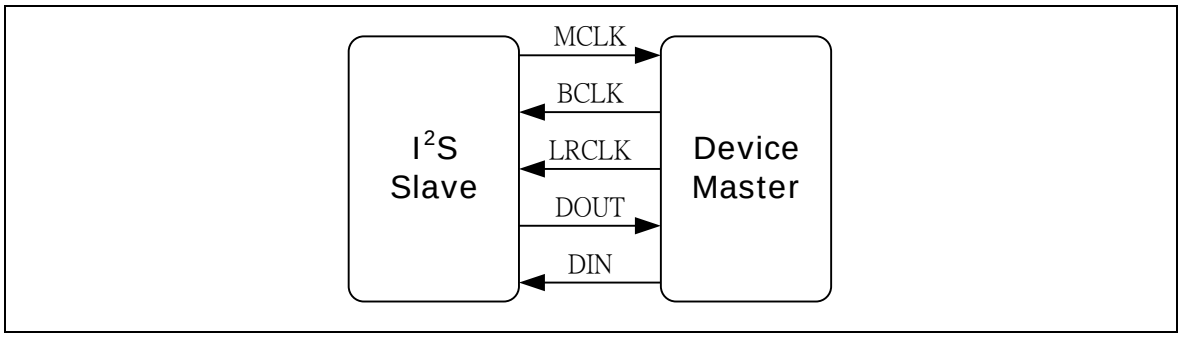


Figure 6.16-9 Slave mode Interface Block Diagram

In Master mode, the MCLK, BCLK, LRCLK is output to device slave. And if in slave mode, the MCLK is output to device master, and BCLK or LRCLK is input from device master.

6.16.6 Register Map

R: Read only, **W**: Write only, **R/W**: Both read and write

Register	Offset	R/W	Description	Reset Value
I2S Base Address **I2S_BA = 0x4004_8000**				
I2S_CTL	I2S_BA+0x00	R/W	I^2S Control Register	0x0000_0000
I2S_CLKDIV	I2S_BA+0x04	R/W	I^2S Clock Divider Register	0x0000_0000
I2S_IEN	I2S_BA+0x08	R/W	I^2S Interrupt Enable Register	0x0000_0000
I2S_STATUS	I2S_BA+0x0C	R/W	I^2S Status Register	0x0014_1000
I2S_TX	I2S_BA+0x10	W	I^2S Transmit FIFO Register	0x0000_0000
I2S_RX	I2S_BA+0x14	R	I^2S Receive FIFO Register	0x0000_0000

6.16.7 Register Description

I²S Control Register (I2S_CTL)

Register	Offset	R/W	Description	Reset Value
I2S_CTL	I2S_BA+0x00	R/W	I²S Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							PCMEN
23	22	21	20	19	18	17	16
RXLCH	Reserved	RXPDMAEN	TXPDMAEN	RXCLR	TXCLR	LZCEN	RZCEN
15	14	13	12	11	10	9	8
MCLKEN	RXTH			TXTH			SLAVE
7	6	5	4	3	2	1	0
FORMAT	MONO	WDWIDTH		MUTE	RXEN	TXEN	I2SEN

Bits	Description	
[31:25]	**Reserved**	Reserved.
[24]	**PCMEN**	**PCM Interface Enable Bit** 0 = I²S Interface. 1 = PCM Interface.
[23]	**RXLCH**	**Receive Left Channel Enable Bit** 0 = Receives right channel data when monaural format is selected. 1 = Receives left channel data when monaural format is selected. **Note:** When monaural format is selected (MONO = 1), I²S will receive right channel data if RXLCH is set to 0, and receive left channel data if RXLCH is set to 1.
[22]	**Reserved**	Reserved.
[21]	**RXPDMAEN**	**Receive DMA Enable Bit** 0 = RX DMA Disabled. 1 = RX DMA Enabled. **Note:** When RX DMA is enabled, I²S requests DMA to transfer data from receive FIFO to SRAM if FIFO is not empty.
[20]	**TXPDMAEN**	**Transmit DMA Enable Bit** 0 = TX DMA Disabled. 1 = TX DMA Enabled. **Note:** When TX DMA is enables, I²S request DMA to transfer data from SRAM to transmit FIFO if FIFO is not full.

[19]	**RXCLR**	**Clear Receive FIFO** 0 = No Effect. 1 = Clear RX FIFO. **Note1:** Write 1 to clear receive FIFO, internal pointer is reset to FIFO start point, and RXCNT (I2S_STATUS[27:24]) returns 0 and receive FIFO becomes empty. **Note2:** This bit is cleared by hardware automatically, read it return zero.
[18]	**TXCLR**	**Clear Transmit FIFO** 0 = No Effect. 1 = Clear TX FIFO. **Note1:** Write 1 to clear transmit FIFO, internal pointer is reset to FIFO start point, and TXCNT(I2S_STATUS[31:28]) returns 0 and transmit FIFO becomes empty but data in transmit FIFO is not changed. **Note2:** This bit is clear by hardware automatically, read it return zero.
[17]	**LZCEN**	**Left Channel Zero-Cross Detect Enable Bit** 0 = Left channel zero-cross detect Disabled. 1 = Left channel zero-cross detect Enabled. **Note1:** If this bit is set to 1, when left channel data sign bit change or next shift data bits are all zero then LZCIF(I2S_STATUS[23]) flag is set to 1. **Note2:** If LZCIF Flag is set to 1, the left channel will be mute.
[16]	**RZCEN**	**Right Channel Zero-Cross Detection Enable Bit** 0 = Right channel zero-cross detect Disabled. 1 = Right channel zero-cross detect Enabled. **Note1:** If this bit is set to 1, when right channel data sign bit change or next shift data bits are all zero then RZCIF(I2S_STATUS[22]) flag is set to 1. **Note2:** If RZCIF Flag is set to 1, the right channel will be mute.
[15]	**MCLKEN**	**Master Clock Enable Bit** 0 = Master clock Disabled. 1 = Master clock Enabled. **Note:** If the external crystal clock in NuMicro™ NUC442/NUC472 series is frequency 2*N*256fs, software can program MCLKDIV(I2S_CLKDIV[5:0]) to get 256fs clock to audio codec chip.
[14:12]	**RXTH**	**Receive FIFO Threshold Level** 000 = 1 word data in receive FIFO. 001 = 2 word data in receive FIFO. 010 = 3 word data in receive FIFO. 011 = 4 word data in receive FIFO. 100 = 5 word data in receive FIFO. 101 = 6 word data in receive FIFO. 110 = 7 word data in receive FIFO. 111 = 8 word data in receive FIFO. **Note:** When received data word(s) in buffer is equal to or higher than threshold level then RXTHIF flag is set.

[11:9]	**TXTH**	**Transmit FIFO Threshold Level** 000 = 0 word data in transmit FIFO. 001 = 1 word data in transmit FIFO. 010 = 2 words data in transmit FIFO. 011 = 3 words data in transmit FIFO. 100 = 4 words data in transmit FIFO. 101 = 5 words data in transmit FIFO. 110 = 6 words data in transmit FIFO. 111 = 7 words data in transmit FIFO. **Note:** If remain data word(s) in transmit FIFO is the same or less than threshold level then TXTHIF flag is set.
[8]	**SLAVE**	**Slave Mode Enable Bit** 0 = Master mode. 1 = Slave mode. **Note:** I^2S can operate as master or slave. For Master mode, I2S_BCLK and I2S_LRCLK pins are output mode and send bit clock from NuMicro™ NUC442/NUC472 series to Audio CODEC chip. In Slave mode, I2S_BCLK and I2S_LRCLK pins are input mode and I2S_BCLK and I2S_LRCLK signals are received from outer Audio CODEC chip.
[7]	**FORMAT**	**Data Format Selection** If PCM=0,. 0 = I2S data format. 1 = MSB justified data format. If PCM=1,. 0 = PCM mode A. 1 = PCM mode B.
[6]	**MONO**	**Monaural Data Control** 0 = Data is stereo format. 1 = Data is monaural format. **Note:** when chip records data, only right channel data will be saved if monaural format is select.
[5:4]	**WDWIDTH**	**Word Width** 00 = data is 8-bit. 01 = data is 16-bit. 10 = data is 24-bit. 11 = data is 32-bit.
[3]	**MUTE**	**Transmit Mute Enable Bit** 0 = Transmit data is shifted from buffer. 1 = Transmit zero data.
[2]	**RXEN**	**Receive Enable Bit** 0 = Data receiving Disabled. 1 = Data receiving Enabled.
[1]	**TXEN**	**Transmit Enable Bit** 0 = Data transmission Disabled. 1 = Data transmission Enabled.

[0]	**I2SEN**	**I²S Controller Enable Bit** 0 = Disabled. 1 = Enabled.

I²S Clock Divider (I2S_CLKDIV)

Register	Offset	R/W	Description	Reset Value
I2S_CLKDIV	I2S_BA+0x04	R/W	I²S Clock Divider Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**BCLKDIV**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
BCLKDIV							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**MCLKDIV**					

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16:8]	**BCLKDIV**	**Bit Clock Divider** If I²S operates in Master mode, bit clock is provided by the NuMicro™ NUC442/NUC472 series. Software can program these bits to generate sampling rate clock frequency. F_BCLK= F_I2SCLK / (2*(BCLKDIV + 1)). **Note:** F_BCLK is the frequency of BCLK and F_I2SCLK is the frequency of I2S_CLK
[7:6]	**Reserved**	Reserved.
[5:0]	**MCLKDIV**	**Master Clock Divider** If chip external crystal frequency is (2xMCLKDIV)*256fs then software can program these bits to generate 256fs clock frequency to audio codec chip. If MCLKDIV is set to 0, MCLK is the same as external clock input. For example, sampling rate is 24 kHz and chip external crystal clock is 12.288 MHz, set MCLKDIV = 1. F_MCLK = F_I2SCLK/(2x(MCLKDIV)) (When MCLKDIV is >= 1). F_MCLK = F_I2SCLK (When MCLKDIV is set to 0). **Note:** F_MCLK is the frequency of MCLK, and F_i2SCLK is the frequency of the I2S_CLK

I²S Interrupt Enable Register (I2S_IEN)

Register	Offset	R/W	Description	Reset Value
I2S_IEN	I2S_BA+0x08	R/W	I²S Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved			LZCIEN	RZCIEN	TXTHIEN	TXOVIEN	TXUDIEN
7	6	5	4	3	2	1	0
Reserved					RXTHIEN	RXOVIEN	RXUDIEN

Bits	Description	
[31:13]	**Reserved**	Reserved.
[12]	**LZCIEN**	**Left Channel Zero-Cross Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled. **Note:** Interrupt occurs if this bit is set to 1 and left channel zero-cross
[11]	**RZCIEN**	**Right Channel Zero-Cross Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled. **Note:** Interrupt occurs if this bit is set to 1 and right channel zero-cross
[10]	**TXTHIEN**	**Transmit FIFO Threshold Level Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled. **Note:** Interrupt occurs if this bit is set to 1 and data words in transmit FIFO is less than TXTH(I2S_CTL[11:9]).
[9]	**TXOVIEN**	**Transmit FIFO Overflow Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled. **Note:** Interrupt occurs if this bit is set to 1 and TXOVIF(I2S_STATUS[17]) flag is set to 1
[8]	**TXUDIEN**	**Transmit FIFO Underflow Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled. **Note:** Interrupt occur if this bit is set to 1 and TXUDIF(I2S_STATUS[16]) flag is set to 1.
[7:3]	**Reserved**	Reserved.

[2]	**RXTHIEN**	**Receive FIFO Threshold Level Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled. **Note:** When data word in receive FIFO is equal or higher than RXTH(I2S_CTL[14:12]) and the RXTHIF bit is set to 1. If RXTHIEN bit is enabled, interrupt occur.
[1]	**RXOVIEN**	**Receive FIFO Overflow Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled. **Note:** Interrupt occurs if this bit is set to 1 and RXOVIF(I2S_STATUS[9]) flag is set to 1
[0]	**RXUDIEN**	**Receive FIFO Underflow Interrupt E Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled. **Note:** If software reads receive FIFO when it is empty then RXUDIF(I2S_STATUS[8]) flag is set to 1.

I²S Status Register (I2S_STATUS)

Register	Offset	R/W	Description	Reset Value
I2S_STATUS	I2S_BA+0x0C	R/W	I²S Status Register	0x0014_1000

31	30	29	28	27	26	25	24
TXCNT				RXCNT			
23	22	21	20	19	18	17	16
LZCIF	RZCIF	TXBUSY	TXEMPTY	TXFULL	TXTHIF	TXOVIF	TXUDIF
15	14	13	12	11	10	9	8
Reserved			RXEMPTY	RXFULL	RXTHIF	RXOVIF	RXUDIF
7	6	5	4	3	2	1	0
Reserved				RIGHT	TXIF	RXIF	I2SIF

Bits	Description	
[31:28]	**TXCNT**	**Transmit FIFO Level (Read Only)** These bits indicate word number in transmit FIFO 0000 = No data. 0001 = 1 word in transmit FIFO. …. 1000 = 8 words in transmit FIFO.
[27:24]	**RXCNT**	**Receive FIFO Level (Read Only)** These bits indicate word number in receive FIFO 0000 = No data. 0001 = 1 word in receive FIFO. …. 1000 = 8 words in receive FIFO.
[23]	**LZCIF**	**Left Channel Zero-Cross Flag** It indicates left channel next sample data sign bit is changed or all data bits are zero. 0 = No zero-cross. 1 = Left channel zero-cross is detected. **Note:** Write 1 to clear this bit to 0.
[22]	**RZCIF**	**Right Channel Zero-Cross Flag** It indicates right channel next sample data sign bit is changed or all data bits are zero. 0 = No zero-cross. 1 = Right channel zero-cross is detected. **Note:** Write 1 to clear this bit to 0.

[21]	**TXBUSY**	**Transmit Busy (Read Only)** 0 = Transmit shift buffer is empty. 1 = Transmit shift buffer is busy. **Note:** This bit is cleared to 0 when all data in transmit FIFO and shift buffer is shifted out. And set to 1 when 1st data is load to shift buffer.
[20]	**TXEMPTY**	**Transmit FIFO Empty (Read Only)** This bit reflect data word number in transmit FIFO is zero 0 = Not empty. 1 = Empty.
[19]	**TXFULL**	**Transmit FIFO Full (Read Only)** This bit reflect data word number in transmit FIFO is 8 0 = Not full. 1 = Full.
[18]	**TXTHIF**	**Transmit FIFO Threshold Flag (Read Only)** 0 = Data word(s) in FIFO is higher than threshold level. 1 = Data word(s) in FIFO is equal or lower than threshold level. **Note:** When data word(s) in transmit FIFO is equal or lower than threshold value set in TXTH(I2S_CTL[11:9]) the TXTHIF bit becomes to 1. It keeps at 1 till TXCNT is higher than TXTH after software write TXFIFO register.
[17]	**TXOVIF**	**Transmit FIFO Overflow Flag** 0 = No overflow. 1 = Overflow. **Note1:** Write data to transmit FIFO when it is full and this bit set to 1 **Note2:** Write 1 to clear this bit to 0.
[16]	**TXUDIF**	**Transmit FIFO Underflow Flag** 0 = No underflow. 1 = Underflow. **Note1:** When transmit FIFO is empty and shift logic hardware read data from data FIFO causes this set to 1. **Note2:** Write 1 to clear this bit to 0.
[15:13]	**Reserved**	Reserved.
[12]	**RXEMPTY**	**Receive FIFO Empty (Read Only)** 0 = Not empty. 1 = Empty. **Note:** This bit reflects data words number in receive FIFO is zero
[11]	**RXFULL**	**Receive FIFO Full (Read Only)** 0 = Not full. 1 = Full. **Note:** This bit reflects data words number in receive FIFO is 8.
[10]	**RXTHIF**	**Receive FIFO Threshold Flag (Read Only)** 0 = Data word(s) in FIFO is lower than threshold level. 1 = Data word(s) in FIFO is equal or higher than threshold level. **Note:** When data word(s) in receive FIFO is equal or higher than threshold value set in RXTH(I2S_CTL[14:12]) the RXTHIF bit becomes to 1. It keeps at 1 till RXCNT less than RXTH after software read RXFIFO register.

[9]	**RXOVIF**	**Receive FIFO Overflow Flag** 0 = No overflow occur. 1 = Overflow occur. **Note1:** When receive FIFO is full and receive hardware attempt write to data into receive FIFO then this bit is set to 1, data in 1st buffer is overwrote. **Note2:** Write 1 to clear this bit to 0.
[8]	**RXUDIF**	**Receive FIFO Underflow Flag** 0 = No underflow occur. 1 = Underflow occur. **Note1:** When receive FIFO is empty, and software reads the receive FIFO again. This bit will be set to 1, and it indicates underflow situation occurs. **Note2:** Write 1 to clear this bit to zero
[7:4]	**Reserved**	Reserved.
[3]	**RIGHT**	**Right Channel (Read Only)** 0 = Left channel. 1 = Right channel. **Note:** This bit indicate current transmit data is belong to right channel
[2]	**TXIF**	**I^2S Transmit Interrupt (Read Only)** 0 = No transmit interrupt. 1 = Transmit interrupt.
[1]	**RXIF**	**I^2S Receive Interrupt (Read Only)** 0 = No receive interrupt. 1 = Receive interrupt.
[0]	**I2SIF**	**I^2S Interrupt Flag (Read Only)** 0 = No I^2S interrupt. 1 = I^2S interrupt. **Note:** It is wire-OR of TXIF and RXIF bits.

I²S Transmit FIFO (I2S_TX)

Register	Offset	R/W	Description	Reset Value
I2S_TX	I2S_BA+0x10	W	I²S Transmit FIFO Register	0x0000_0000

31	30	29	28	27	26	25	24
TX							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TX							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TX							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TX							

Bits	Description	
[31:0]	**TX**	**Transmit FIFO Bits** I²S contains 8 words (8x32 bit) data buffer for data transmit. Write data to this register to prepare data for transmit. The remaining word number is indicated by TXCNT(I2S_STATUS[31:28]).

I²S Receive FIFO (I2S_RX)

Register	Offset	R/W	Description	Reset Value
I2S_RX	I2S_BA+0x14	R	I²S Receive FIFO Register	0x0000_0000

31	30	29	28	27	26	25	24
RX							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RX							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RX							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RX							

Bits	Description	
[31:0]	**RX**	**Receive FIFO Bits** I²S contains 8 words (8x32 bit) data buffer for data receive. Read this register to get data in FIFO. The remaining data word number is indicated by RXCNT(I2S_STATUS[27:24]).

6.17 Image Capture Interface (ICAP)

6.17.1 Overview

The Image Capture Interface is designed to capture image data from a sensor. After capturing or fetching image data, it will process the image data, and then FIFO outputs them into a frame buffer.

6.17.2 Block Diagram

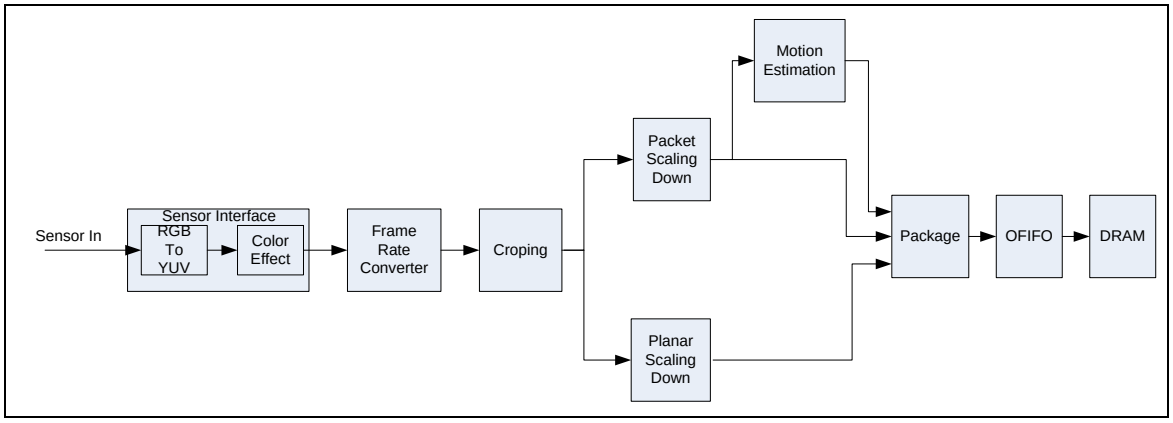


Figure 6.17-1 Image Capture Interface Block Diagram

6.17.3 Features

- 8-bit RGB565 sensor
- 8-bit YUV422 sensor
- Supports CCIR601 YCbCr color range scale to full YUV color range
- Supports 4 packaging format for packet data output: YUYV, Y only, RGB565, RGB555
- Supports YUV422 planar data output
- Supports the CROP function to crop input image to the required size for digital application.
- Supports the down scaling function to scale input image to the required size for digital application.
- Supports frame rate control
- Supports field detection and even/odd field skip mechanism
- Supports packet output dual buffer control through hardware buffer controller
- Supports negative/sepia/posterization color effect
- Supports two independent capture interfaces

6.17.4 Functional Description

6.17.4.1 Image Capture Flow Chart

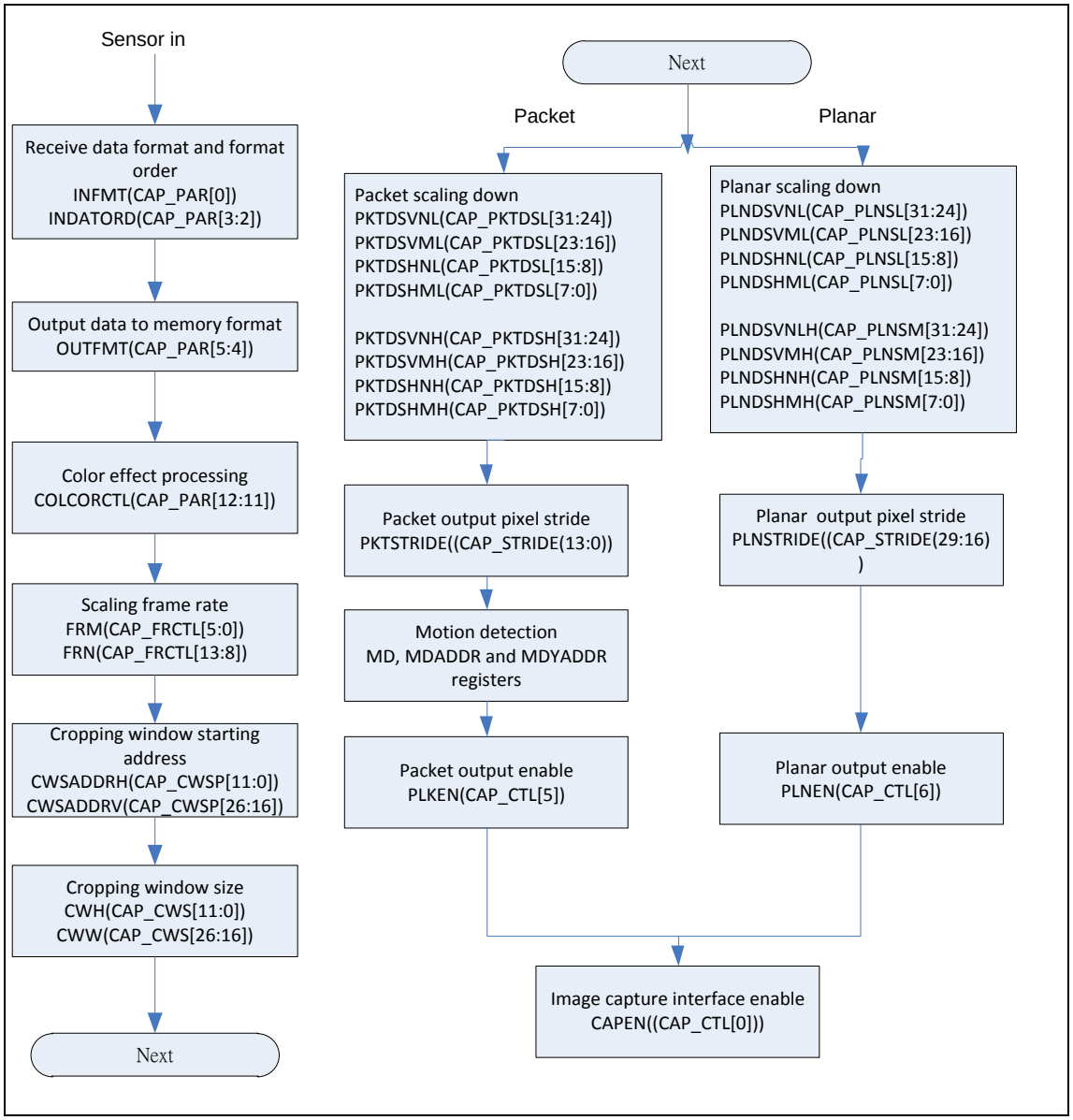


Figure 6.17-2 Image Capture Flow Chart

6.17.4.2 Cropping Feature

The capture interface can select a window from the received image. The size of the window is specified by the number of pixel clocks (horizontal dimension) and the number of lines (vertical dimension). The start (left upper corner) coordinates can be specified by the CAP_CWSP register. The size (vertical dimension in number of lines and horizontal dimension in number of pixel clocks) can be specified by the CAP_CWS register.

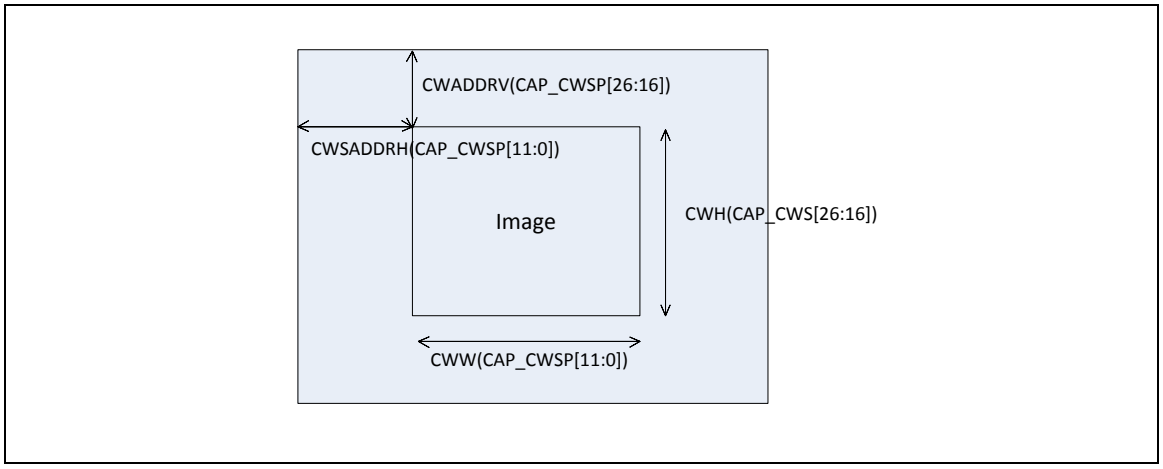


Figure 6.17-3 Image Start and Size of the Window after Cropping Block

6.17.4.3 One Shutter Mode (Single Frame)

In this mode, a single frame is captured. After the SHUTTER (CAP_CTL[16]) bit is set, the Image Capture interface automatically disables the capture interface after a frame is captured.

6.17.4.4 Motion Detection

The feature is used to detect object movement. MDSM (CAP_MD[9]) and MDBS (CAP_MD[8]) are determined using the method of storage. If the difference between the center of BASEADDR(CAP_PKTBA0[31:0]) block and the center of MDYADDR(CAP_MDYADDR[31:0]) block is greater than MDTHR (CAP_MD[20:16]) the first bit will change to 1, and other bits will be different, as shown in the following figure.

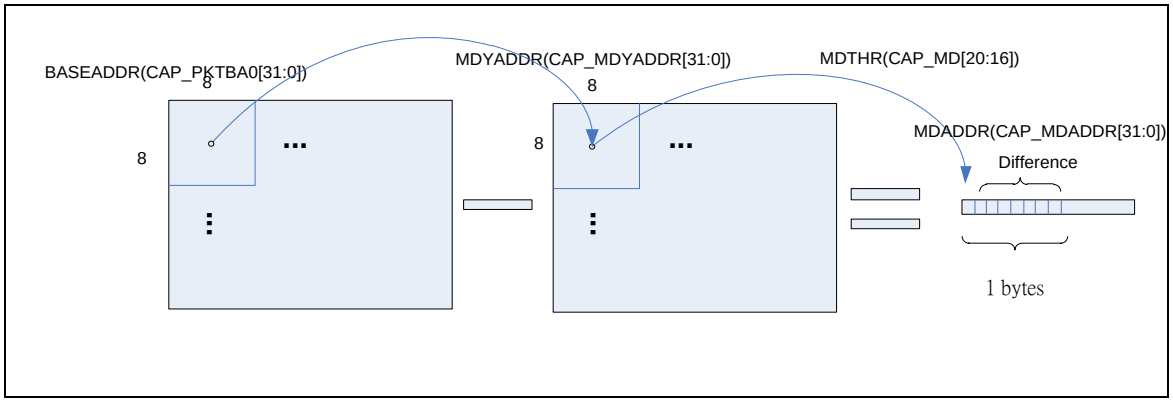


Figure 6.17-4 MDSM is set to 0 and MDBS is set to 1

If the difference between the center of BASEADDR(CAP_PKTBA0[31:0]) block and the center of CAP_MDYADDR block is greater than MDTHR (CAP_MD[20:16]) the first bit is set to 1, and if the next difference between the center of PACBA0 block and the center of MDYADDR(CAP_MDYADDR[31:0]) block is greater than MDTHR (CAP_MD[20:16]) the second bit will change to 1, as shown in the following figure.

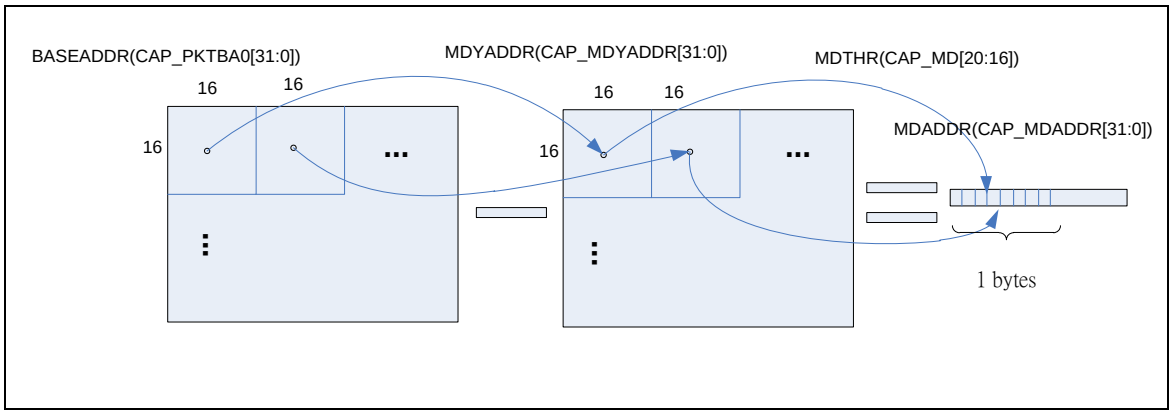


Figure 6.17-5 MDSM is set to 1 and MDBS is set to 0

6.17.5 Register Map

R: read only, **W**: write only, **R / W**: both read and write, **C**: Only value 0 can be written

Register	Offset	R/W	Description	Reset Value
CAP Base Address: **CAP_BA = 0x4003_0000**				
CAP_CTL	CAP_BA+0x00	R/W	Image Capture Interface Control Register	0x0000_0040
CAP_PAR	CAP_BA+0x04	R/W	Image Capture Interface Parameter Register	0x0000_0000
CAP_INT	CAP_BA+0x08	R/W	Image Capture Interface Interrupt Register	0x0000_0000
CAP_POSTERIZE	CAP_BA+0x0C	R/W	YUV Component Posterizing Factor Register	0x0000_0000
CAP_MD	CAP_BA+0x10	R/W	Motion Detection Register	0x0000_0000
CAP_MDADDR	CAP_BA+0x14	R/W	Motion Detection Output Address Register	0x0000_0000
CAP_MDYADDR	CAP_BA+0x18	R/W	Motion Detection Temp Y Output Address Register	0x0000_0000
CAP_SEPIA	CAP_BA+0x1C	R/W	Sepia Effect Control Register	0x0000_0000
CAP_CWSP	CAP_BA+0x20	R/W	Cropping Window Starting Address Register	0x0000_0000
CAP_CWS	CAP_BA+0x24	R/W	Cropping Window Size Register	0x0000_0000
CAP_PKTSL	CAP_BA+0x28	R/W	Packet Scaling Vertical/Horizontal Factor Register (LSB)	0x0000_0000
CAP_PLNSL	CAP_BA+0x2C	R/W	Planar Scaling Vertical/Horizontal Factor Register (LSB)	0x0000_0000
CAP_FRCTL	CAP_BA+0x30	R/W	Scaling Frame Rate Factor Register	0x0000_0000
CAP_STRIDE	CAP_BA+0x34	R/W	Frame Output Pixel Stride Width Register	0x0000_0000
CAP_FIFOTH	CAP_BA+0x3C	R/W	FIFO Threshold Register	0x070D_0507
CAP_CMPADDR	CAP_BA+0x40	R/W	Compare Memory Base Address Register	0xFFFF_FFFC
CAP_PKTSM	CAP_BA+0x48	R/W	Packet Scaling Vertical/Horizontal Factor Register (MSB)	0x0000_0000
CAP_PLNSM	CAP_BA+0x4C	R/W	Planar Scaling Vertical/Horizontal Factor Register (MSB)	0x0000_0000
CAP_CURADDRP	CAP_BA+0x50	R	Current Packet System Memory Address Register	0x0000_0000
CAP_CURADDRY	CAP_BA+0x54	R	Current Planar Y System Memory Address Register	0x0000_0000
CAP_CURADDRU	CAP_BA+0x58	R	Current Planar U System Memory Address Register	0x0000_0000
CAP_CURVADDR	CAP_BA+0x5C	R	Current Planar V System Memory Address Register	0x0000_0000
CAP_PKTBA0	CAP_BA+0x60	R/W	System Memory Packet Base Address 0 Register	0x0000_0000
CAP_PKTBA1	CAP_BA+0x64	R/W	System Memory Packet Base Address 1 Register	0x0000_0000
CAP_YBA	CAP_BA+0x80	R/W	System Memory Planar Y Base Address Register	0x0000_0000
CAP_UBA	CAP_BA+0x84	R/W	System Memory Planar U Base Address Register	0x0000_0000
CAP_VBA	CAP_BA+0x88	R/W	System Memory Planar V Base Address Register	0x0000_0000

6.17.6 Register Description

Image Capture Interface Control Register (CAP_CTL)

Register	Offset	R/W	Description	Reset Value
CAP_CTL	CAP_BA+0x00	R/W	Image Capture Interface Control Register	0x0000_0040

31	30	29	28	27	26	25	24
Reserved							VPRST
23	22	21	20	19	18	17	16
Reserved			UPDATE	Reserved			SHUTTER
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	PKTEN	PLNEN	Reserved	ADDRSW	Reserved	Reserved	CAPEN

Bits	Description	
[31:25]	**Reserved**	Reserved.
[24]	**VPRST**	**Capture Interface Reset** 0 = Capture interface reset Disabled. 1 = Capture interface reset Enabled.
[23:21]	**Reserved**	Reserved.
[20]	**UPDATE**	**Update Register At New Frame** 0 = Update register at new frame Disabled. 1 = Update register at new frame Enabled (Auto clear to 0 when register updated).
[19:16]	**Reserved**	Reserved.
[16]	**SHUTTER**	**Image Capture Interface Automatically Disable The Capture Inteface After A Frame Had Been Captured** 0 = Shutter Disabled. 1 = Shutter Enabled.
[15:7]	**Reserved**	Reserved.
[6]	**PKTEN**	**Packet Output Enable** 0 = Packet output Disabled. 1 = Packet output Enabled.
[5]	**PLNEN**	**Planar Output Enable** 0 = Planar output Disabled. 1 = Planar output Enabled.
[4]	**Reserved**	Reserved.
[3]	**ADDRSW**	**Packet Buffer Address Switch**

		0 = Packet buffer address switch Disabled. 1 = Packet buffer address switch Enabled.
[2:1]	**Reserved**	Reserved.
[0]	**CAPEN**	**Image Capture Interface Enable** 0 = Image Capture Interface Disabled. 1 = Image Capture Interface Enabled.

Image Capture Interface Parameter Register (CAP_PAR)

Register	Offset	R/W	Description	Reset Value
CAP_PAR	CAP_BA+0x04	R/W	Image Capture Interface Parameter Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved					FBB	Reserved	
15	14	13	12	11	10	9	8
Reserved			COLORCTL		VSP	HSP	PCLKP
7	6	5	4	3	2	1	0
PLNFMT	RANGE	OUTFMT		INDATORD		SENTYPE	INFMT

Bits	Description	
[31:19]	**Reserved**	Reserved.
[18]	**FBB**	**Field By Blank** Hardware will tag field0 or field1 by vertical blanking instead of FIELD flag in ccir-656 mode. 0 = Field by blank Disabled. 1 = Field by blank Enabled.
[17:13]	**Reserved**	Reserved.
[12:11]	**COLORCTL**	**Special COLORCTL Processing** 00 = Normal Color. 01 = Sepia effect, corresponding U,V component value is set at register CAP_SEPIA. 10 = Negative picture. 11 = Posterize image, the Y, U, V components posterizing factor are set at register CAP_POSTERIZE.
[10]	**VSP**	**Sensor Vsync Polarity** 0 = Sync Low. 1 = Sync High.
[9]	**HSP**	**Sensor Hsync Polarity** 0 = Sync Low. 1 = Sync High.
[8]	**PCLKP**	**Sensor Pixel Clock Polarity** 0 = Input video data and signals are latched by falling edge of Pixel Clock. 1 = Input video data and signals are latched by rising edge of Pixel Clock.
[7]	**PLNFMT**	**Planar Output YUV Format** 0 = YUV422. 1 = YUV420.

<table>
<tr><td>[6]</td><td>RANGE</td><td>Scale Input YUV CCIR601 Color Range To Full Range
0 = default.
1 = Scale to full range.</td></tr>
<tr><td>[5:4]</td><td>OUTFMT</td><td>Image Data Format Output To System Memory
00 = YCbCr422.
01 = Only output Y.
10 = RGB555.
11 = RGB565.</td></tr>
<tr><td>[3:2]</td><td>INDATORD</td><td>Sensor Input Data Order
If INFMT = 0 (YCbCr),.
Byte 0 1 2 3
00 = Y0 U0 Y1 V0.
01 = Y0 V0 Y1 U0.
10 = U0 Y0 V0 Y1.
11 = V0 Y0 U0 Y1.
If INFMT = 1 (RGB565),.
00
<table>
<tr><td>Byte</td><td colspan="2">Bit [7:0]</td></tr>
<tr><td>0</td><td>R[4:0]</td><td>G[5:3]</td></tr>
<tr><td>1</td><td>G[2:0]</td><td>B[4:0]</td></tr>
</table>
01
<table>
<tr><td>Byte</td><td colspan="2">Bit [7:0]</td></tr>
<tr><td>0</td><td>B[4:0]</td><td>G[5:3]</td></tr>
<tr><td>1</td><td>G[2:0]</td><td>R[4:0]</td></tr>
</table>
10
<table>
<tr><td>Byte</td><td colspan="2">Bit [7:0]</td></tr>
<tr><td>0</td><td>G[2:0]</td><td>B[4:0]</td></tr>
<tr><td>1</td><td>R[4:0]</td><td>G[5:3]</td></tr>
</table>
11
<table>
<tr><td>Byte</td><td colspan="2">Bit [7:0]</td></tr>
<tr><td>0</td><td>G[2:0]</td><td>R[4:0]</td></tr>
<tr><td>1</td><td>B[4:0]</td><td>G[5:3]</td></tr>
</table></td></tr>
<tr><td>[1]</td><td>SENTYPE</td><td>Sensor Input Type
0 = CCIR601.
1 = CCIR656, VSync & Hsync embedded in the data signal.</td></tr>
<tr><td>[0]</td><td>INFMT</td><td>Sensor Input Data Format
0 = YCbCr422.
1 = RGB565.</td></tr>
</table>

Image Capture Interface Interrupt Register (CAP_INT)

Register	Offset	R/W	Description	Reset Value
CAP_INT	CAP_BA+0x08	R/W	Image Capture Interface Interrupt Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved			MDIEN	ADDRMIEN	Reserved	MEIEN	VIEN
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			MDINTF	ADDRMINTF	Reserved	MEINTF	VINTF

Bits	Description	
[31:21]	**Reserved**	Reserved.
[20]	**MDIEN**	**Motion Detection Output Finish Interrupt Enable** 0 = CAP_MD finish interrupt Disabled. 1 = CAP_MD finish interrupt Enabled.
[19]	**ADDRMIEN**	**Address Match Interrupt Enable** 0 = Address match interrupt Disabled. 1 = Address match interrupt Enabled.
[18]	**Reserved**	Reserved.
[17]	**MEIEN**	**System Memory Error Interrupt Enable** 0 = System memory error interrupt Disabled. 1 = System memory error interrupt Enabled.
[16]	**VIEN**	**Video Frame End Interrupt Enable** 0 = Video frame end interrupt Disabled. 1 = Video frame end interrupt Enabled.
[15:5]	**Reserved**	Reserved.
[4]	**MDINTF**	**Motion Detection Output Finish Interrupt** If this bit shows 1, Motion Detection Output Finish Interrupt occurred. Write 1 to clear it.
[3]	**ADDRMINTF**	**Memory Address Match Interrupt** If this bit shows 1, Memory Address Match Interrupt occurred. Write 1 to clear it.
[2]	**Reserved**	Reserved.
[1]	**MEINTF**	**Bus Master Transfer Error Interrupt** If this bit shows 1, Transfer Error occurred. Write 1 to clear it.

[0]	**VINTF**	**Video Frame End Interrupt** If this bit shows 1, receiving a frame completed. Write 1 to clear it.

YUV Component Posterizing Factor Register (CAP_POSTERIZE)

Register	Offset	R/W	Description	Reset Value
CAP_POSTERIZE	CAP_BA+0x0C	R/W	YUV Component Posterizing Factor Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
YCOMP							
15	14	13	12	11	10	9	8
UCOMP							
7	6	5	4	3	2	1	0
VCOMP							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**YCOMP**	**Y Component Posterizing Factor** Final_Y_Out = Original_Y[7:0] & Y_Posterizing_Factor.
[15:8]	**UCOMP**	**U Component Posterizing Factor** Final_U_Out = Original_U[7:0] & U_Posterizing_Factor.
[7:0]	**VCOMP**	**V Component Posterizing Factor** Final_V_Out = Original_V[7:0] & V_Posterizing_Factor.

Motion Detection Register (CAP_MD)

Register	Offset	R/W	Description	Reset Value
CAP_MD	CAP_BA+0x10	R/W	Motion Detection Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved			MDTHR				
15	14	13	12	11	10	9	8
Reserved				MDDF		MDSM	MDBS
7	6	5	4	3	2	1	0
Reserved							MDEN

Bits	Description	
[31:21]	**Reserved**	Reserved.
[20:16]	**MDTHR**	**Motion Detection Differential Threshold**
[15:12]	**Reserved**	Reserved.
[11:10]	**MDDF**	**Motion Detection Detect Frequency** 00 = Each frame. 01 = Every 2 frame. 10 = Every 3 frame. 11 = Every 4 frame.
[9]	**MDSM**	**Motion Detection Save Mode** 0 = 1 bit DIFF + 7 bit Y Differential. 1 = 1 bit DIFF only.
[8]	**MDBS**	**Motion Detection Block Size** 0 = 16x16. 1 = 8x8.
[7:1]	**Reserved**	Reserved.
[0]	**MDEN**	**Motion Detection Enable** 0 = CAP_MD Disabled. 1 = CAP_MD Enabled.

Motion Detection Output Address Register (CAP_MDADDR)

Register	Offset	R/W	Description	Reset Value
CAP_MDADDR	CAP_BA+0x14	R/W	Motion Detection Output Address Register	0x0000_0000

31	30	29	28	27	26	25	24
MDADDR							
23	22	21	20	19	18	17	16
MDADDR							
15	14	13	12	11	10	9	8
MDADDR							
7	6	5	4	3	2	1	0
MDADDR							

Bits	Description	
[31:0]	MDADDR	Motion Detection Output Address Register (Word Alignment)

Motion Detection Temp Y Output Address Register (CAP_MDYADDR)

Register	Offset	R/W	Description	Reset Value
CAP_MDYADDR	CAP_BA+0x18	R/W	Motion Detection Temp Y Output Address Register	0x0000_0000

31	30	29	28	27	26	25	24
MDYADDR							
23	22	21	20	19	18	17	16
MDYADDR							
15	14	13	12	11	10	9	8
MDYADDR							
7	6	5	4	3	2	1	0
MDYADDR							

Bits	Description	
[31:0]	MDYADDR	Motion Detection Temp Y Output Address Register (Word Alignment)

Sepia Effect Control Register (CAP_SEPIA)

Register	Offset	R/W	Description	Reset Value
CAP_SEPIA	CAP_BA+0x1C	R/W	Sepia Effect Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
UCOMP							
7	6	5	4	3	2	1	0
VCOMP							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:8]	**UCOMP**	Define the constant U component while “Sepia” color effect is turned on.
[7:0]	**VCOMP**	Define the constant V component while “Sepia” color effect is turned on.

Cropping Window Starting Address Register (CAP_CWSP)

Register	Offset	R/W	Description	Reset Value
CAP_CWSP	CAP_BA+0x20	R/W	Cropping Window Starting Address Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					**CWSADDRV**		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CWSADDRV							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**CWSADDRH**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CWSADDRH							

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26:16]	**CWSADDRV**	**Cropping Window Vertical Starting Address**
[15:12]	**Reserved**	Reserved.
[11:0]	**CWSADDRH**	**Cropping Window Horizontal Starting Address**

Cropping Window Size Register (CAP_CWS)

Register	Offset	R/W	Description	Reset Value
CAP_CWS	CAP_BA+0x24	R/W	Cropping Window Size Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					**CWH**		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CWH							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**CWW**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CWW							

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26:16]	**CWH**	**Cropping Window Height**
[15:12]	**Reserved**	Reserved.
[11:0]	**CWW**	**Cropping Window Width**

Packet Scaling Vertical/Horizontal Factor Register (LSB) (CAP_PKTSL)

Register	Offset	R/W	Description	Reset Value
CAP_PKTSL	CAP_BA+0x28	R/W	Packet Scaling Vertical/Horizontal Factor Register (LSB)	0x0000_0000

31	30	29	28	27	26	25	24
PKTSVNL							
23	22	21	20	19	18	17	16
PKTSVML							
15	14	13	12	11	10	9	8
PKTSHNL							
7	6	5	4	3	2	1	0
PKTSHML							

Bits	Description	
[31:24]	**PKTSVNL**	**Packet Scaling Vertical Factor N (Lower 8-Bit)** Specify the lower 8-bit of numerator part (N) of the vertical scaling factor. The lower 8-bit will be cascaded with higher 8-bit (PKDSVNH) to form a 16-bit numerator of vertical factor.
[23:16]	**PKTSVML**	**Packet Scaling Vertical Factor M (Lower 8-Bit)** Specify the lower 8-bit of denominator part (M) of the vertical scaling factor. The lower 8-bit will be cascaded with higher 8-bit (PKDSVMH) to form a 16-bit denominator (M) of vertical factor. The output image width will be equal to the image height * N/M. **Note:** The value of N must be equal to or less than M.
[15:8]	**PKTSHNL**	**Packet Scaling Horizontal Factor N (Lower 8-Bit)** Specify the lower 8-bit of numerator part (N) of the horizontal scaling factor. The lower 8-bit will be cascaded with higher 8-bit (PKDSHNH) to form a 16-bit numerator of horizontal factor.
[7:0]	**PKTSHML**	**Packet Scaling Horizontal Factor M (Lower 8-Bit)** Specifies the lower 8-bit of denominator part (M) of the horizontal scaling factor. The lower 8-bit will be cascaded with higher 8-bit (PKDSHMH) to form a 16-bit denominator (M) of vertical factor. The output image width will be equal to the image width * N/M. **Note:** The value of N must be equal to or less than M.

Planar Scaling Vertical/Horizontal Factor Register (LSB) (CAP_PLNSL)

Register	Offset	R/W	Description	Reset Value
CAP_PLNSL	CAP_BA+0x2C	R/W	Planar Scaling Vertical/Horizontal Factor Register (LSB)	0x0000_0000

31	30	29	28	27	26	25	24
PLNSVNL							
23	22	21	20	19	18	17	16
PLNSVML							
15	14	13	12	11	10	9	8
PLNSHNL							
7	6	5	4	3	2	1	0
PLNSHML							

Bits	Description	
[31:24]	PLNSVNL	**Planar Scaling Vertical Factor N (Lower 8-Bit)** Specify the lower 8-bit of numerator part (N) of the vertical scaling factor. The lower 8-bit will be cascaded with higher 8-bit (PNDSVNH) to form a 16-bit numerator of vertical factor.
[23:16]	PLNSVML	**Planar Scaling Vertical Factor M (Lower 8-Bit)** Specify the lower 8-bit of denominator part (M) of the vertical scaling factor. The lower 8-bit will be cascaded with higher 8-bit (PNDSVMH) to form a 16-bit denominator (M) of vertical factor. The output image width will be equal to the image height * N/M. **Note:** The value of N must be equal to or less than M.
[15:8]	PLNSHNL	**Planar Scaling Horizontal Factor N (Lower 8-Bit)** Specify the lower 8-bit of numerator part (N) of the horizontal scaling factor. The lower 8-bit will be cascaded with higher 8-bit (PNDSHNH) to form a 16-bit numerator of horizontal factor.
[7:0]	PLNSHML	**Planar Scaling Horizontal Factor M (Lower 8-Bit)** Specify the lower 8-bit of denominator part (M) of the horizontal scaling factor. The lower 8-bit will be cascaded with higher 8-bit (PNDSHMH) to form a 16-bit denominator (M) of vertical factor. The output image width will be equal to the image width * N/M. **Note:** The value of N must be equal to or less than M.

Packet Scaling Vertical/Horizontal Factor Register (MSB) (CAP_PKTSM)

Register	Offset	R/W	Description	Reset Value
CAP_PKTSM	CAP_BA+0x48	R/W	Packet Scaling Vertical/Horizontal Factor Register (MSB)	0x0000_0000

31	30	29	28	27	26	25	24
PKTSVNH							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PKTSVMH							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PKTSHNH							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PKTSHMH							

Bits	Description	
[31:24]	**PKTSVNH**	**Packet Scaling Vertical Factor N (Higher 8-Bit)** Specify the higher 8-bit of numerator part (N) of the vertical scaling factor. Please refer to the register "CAP_PKTSL" to check the cooperation between these two registers.
[23:16]	**PKTSVMH**	**Packet Scaling Vertical Factor M (Higher 8-Bit)** Specify the lower 8-bit of denominator part (M) of the vertical scaling factor. Please refer to the register "CAP_PKTSL" to check the cooperation between these two registers.
[15:8]	**PKTSHNH**	**Packet Scaling Horizontal Factor N (Higher 8-Bit)** Specify the lower 8-bit of numerator part (N) of the horizontal scaling factor. Please refer to the register "CAP_PKTSL" for the detailed operation.
[7:0]	**PKTSHMH**	**Packet Scaling Horizontal Factor M (Higher 8-Bit)** Specify the lower 8-bit of denominator part (M) of the horizontal scaling factor. Please refer to the register "CAP_PKTSL" for the detailed operation.

Planar Scaling Vertical/Horizontal Factor Register (MSB) (CAP_PLNSM)

Register	Offset	R/W	Description	Reset Value
CAP_PLNSM	CAP_BA+0x4C	R/W	Planar Scaling Vertical/Horizontal Factor Register (MSB)	0x0000_0000

31	30	29	28	27	26	25	24
PLNSVNH							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PLNSVMH							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PLNSHNH							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PLNSHMH							

Bits	Description	
[31:24]	**PLNSVNH**	**Planar Scaling Vertical Factor N (Higher 8-Bit)** Specifies the higher 8-bit of numerator part (N) of the vertical scaling factor. For detailed programming, please refer to the register "CAP_PLNSL".
[23:16]	**PLNSVMH**	**Planar Scaling Vertical Factor M (Higher 8-Bit)** Specifies the lower 8-bit of denominator part (M) of the vertical scaling factor. For detailed programming, please refer to the register "CAP_PLNSL".
[15:8]	**PLNSHNH**	**Planar Scaling Horizontal Factor N (Higher 8-Bit)** Specifies the higher 8-bit of numerator part (N) of the horizontal scaling factor. For detailed programming, please refer to the register "CAP_PLNSL".
[7:0]	**PLNSHMH**	**Planar Scaling Horizontal Factor M (Higher 8-Bit)** Specifies the higher 8-bit of denominator part (M) of the horizontal scaling factor For detailed programming, please refer to the register "CAP_PLNSL".

Scaling Frame Rate Factor Register (CAP_FRCTL)

Register	Offset	R/W	Description	Reset Value
CAP_FRCTL	CAP_BA+0x30	R/W	Scaling Frame Rate Factor Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	28	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		FRN					
7	6	5	4	3	2	1	0
Reserved		FRM					

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13:8]	**FRN**	**Scaling Frame Rate Factor N** Specify the denominator part (N) of the frame rate scaling factor.
[7:6]	**Reserved**	Reserved.
[5:0]	**FRM**	**Scaling Frame Rate Factor M** Specify the denominator part (M) of the frame rate scaling factor. The output image frame rate will be equal to input image frame rate * (N/M). **Note:** The value of N must be equal to or less than M.

Output Frame Pixel Stride Width (CAP_STRIDE)

Register	Offset	R/W	Description	Reset Value
CAP_STRIDE	CAP_BA+0x34	R/W	Frame Output Pixel Stride Width Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		PLNSTRIDE					
23	22	21	20	19	28	17	16
PLNSTRIDE							
15	14	13	12	11	10	9	8
Reserved		PKTSTRIDE					
7	6	5	4	3	2	1	0
PKTSTRIDE							

Bits	Description	
[31:28]	**Reserved**	Reserved.
[29:16]	**PLNSTRIDE**	**Planar Frame Output Pixel Stride Width** The output pixel stride size of planar pipe.
[15:14]	**Reserved**	Reserved.
[13:0]	**PKTSTRIDE**	**Packet Frame Output Pixel Stride Width** The output pixel stride size of packet pipe.

FIFO Threshold Register (CAP_FIFOTH)

Register	Offset	R/W	Description	Reset Value
CAP_FIFOTH	CAP_BA+0x3C	R/W	FIFO Threshold Register	0x070D_0507

31	30	29	28	27	26	25	24
OVF	Reserved		PKTFTH				
23	22	21	20	19	18	17	16
Reserved			PLNYFTH				
15	14	13	12	11	10	9	8
Reserved				PLNUFTH			
7	6	5	4	3	2	1	0
Reserved				PLNVFTH			

Bits	Description	
[31]	**OVF**	**FIFO Overflow Flag**
[30:29]	**Reserved**	Reserved.
[28:24]	**PKTFTH**	**Packet FIFO Threshold**
[23:21]	**Reserved**	Reserved.
[20:16]	**PLNYFTH**	**Planar Y FIFO Threshold**
[15:12]	**Reserved**	Reserved.
[11:8]	**PLNUFTH**	**Planar U FIFO Threshold**
[7:4]	**Reserved**	Reserved.
[3:0]	**PLNVFTH**	**Planar V FIFO Threshold**

Compare Memory Address Register (CAP_CMPADDR)

Register	Offset	R/W	Description	Reset Value
CAP_CMPADDR	CAP_BA+0x40	R/W	Compare Memory Base Address Register	0xFFFF_FFFC

31	30	29	28	27	26	25	24
CMPADDR							
23	22	21	20	19	18	17	16
CMPADDR							
15	14	13	12	11	10	9	8
CMPADDR							
7	6	5	4	3	2	1	0
CMPADDR							

Bits	Description	
[31:0]	CMPADDR	**Compare Memory Base Address** Word aligns address; ignore the bits [1:0].

Current Packet System Memory Address Register (CAP_CURADDRP)

Register	Offset	R/W	Description	Reset Value
CAP_CURADDRP	CAP_BA+0x50	R	Current Packet System Memory Address Register	0x0000_0000

31	30	29	28	27	26	25	24
CURADDR							
23	22	21	20	19	18	17	16
CURADDR							
15	14	13	12	11	10	9	8
CURADDR							
7	6	5	4	3	2	1	0
CURADDR							

Bits	Description	
[31:0]	CURADDR	Current Packet Output Memory Address

Current Planar Y System Memory Address Register (CAP_CURADDRY)

Register	Offset	R/W	Description	Reset Value
CAP_CURADDRY	CAP_BA+0x54	R	Current Planar Y System Memory Address Register	0x0000_0000

31	30	29	28	27	26	25	24
CURADDR							
23	22	21	20	19	18	17	16
CURADDR							
15	14	13	12	11	10	9	8
CURADDR							
7	6	5	4	3	2	1	0
CURADDR							

Bits	Description	
[31:0]	CURADDR	Current Planar Y Output Memory Address

Current Planar U System Memory Address Register (CAP_CURADDRU)

Register	Offset	R/W	Description	Reset Value
CAP_CURADDRU	CAP_BA+0x58	R	Current Planar U System Memory Address Register	0x0000_0000

31	30	29	28	27	26	25	24
CURADDR							
23	22	21	20	19	18	17	16
CURADDR							
15	14	13	12	11	10	9	8
CURADDR							
7	6	5	4	3	2	1	0
CURADDR							

Bits	Description	
[31:0]	CURADDR	Current Planar U Output Memory Address

Current Planar V System Memory Address Register (CAP_CURVADDR)

Register	Offset	R/W	Description	Reset Value
CAP_CURVADDR	CAP_BA+0x5C	R	Current Planar V System Memory Address Register	0x0000_0000

31	30	29	28	27	26	25	24
CURADDR							
23	22	21	20	19	18	17	16
CURADDR							
15	14	13	12	11	10	9	8
CURADDR							
7	6	5	4	3	2	1	0
CURADDR							

Bits	Description	
[31:0]	CURADDR	Current Planar V Output Memory Address

System Memory Packet Base Address 0 Register (CAP_PKTBA0)

Register	Offset	R/W	Description	Reset Value
CAP_PKTBA0	CAP_BA+0x60	R/W	System Memory Packet Base Address 0 Register	0x0000_0000

31	30	29	28	27	26	25	24
BASEADDR							
23	22	21	20	19	18	17	16
BASEADDR							
15	14	13	12	11	10	9	8
BASEADDR							
7	6	5	4	3	2	1	0
BASEADDR							

Bits	Description	
[31:0]	BASEADDR	**System Memory Packet Base Address 0** Word aligns address; ignore the bits [1:0].

System Memory Packet Base Address 1 Register (CAP_PKTBA1)

Register	Offset	R/W	Description	Reset Value
CAP_PKTBA1	CAP_BA+0x64	R/W	System Memory Packet Base Address 1 Register	0x0000_0000

31	30	29	28	27	26	25	24
BASEADDR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
BASEADDR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
BASEADDR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BASEADDR							

Bits	Description	
[31:0]	**BASEADDR**	**System Memory Packet Base Address 1** Word aligns address; ignore the bits [1:0].

System Memory Planar Y Base Address Register (CAP_YBA)

Register	Offset	R/W	Description	Reset Value
CAP_YBA	CAP_BA+0x80	R/W	System Memory Planar Y Base Address Register	0x0000_0000

31	30	29	28	27	26	25	24
BASEADDR							
23	22	21	20	19	18	17	16
BASEADDR							
15	14	13	12	11	10	9	8
BASEADDR							
7	6	5	4	3	2	1	0
BASEADDR							

Bits	Description	
[31:0]	BASEADDR	**System Memory Planar Y Base Address** Word aligns address; ignore the bits [1:0].

System Memory Planar U Base Address Register (CAP_UBA)

Register	Offset	R/W	Description	Reset Value
CAP_UBA	CAP_BA+0x84	R/W	System Memory Planar U Base Address Register	0x0000_0000

31	30	29	28	27	26	25	24
BASEADDR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
BASEADDR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
BASEADDR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BASEADDR							

Bits	Description	
[31:0]	**BASEADDR**	**System Memory Planar U Base Address** Word aligns address; ignore the bits [1:0].

System Memory Planar V Base Address Register (CAP_VBA)

Register	Offset	R/W	Description	Reset Value
CAP_VBA	CAP_BA+0x88	R/W	System Memory Planar V Base Address Register	0x0000_0000

31	30	29	28	27	26	25	24
BASEADDR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
BASEADDR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
BASEADDR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BASEADDR							

Bits	Description	
[31:0]	**BASEADDR**	**System Memory Planar V Base Address** Word aligns address; ignore the bits [1:0].

6.18 Enhanced Input Capture Timer

6.18.1 Overview

This device provides up to two units of Input Capture Timer/Counter which capture function can detect the digital edge changed signal at channel inputs. Each unit has three input capture channels. The timer/counter is equipped with up counting, reload and compare-match capabilities.

6.18.2 Features

- Up to two Input Capture Timer/Counter Units, Input Capture 0 and Input Capture 1.
- Each unit has own interrupt vector.
- 24-bit Input Capture up-counting timer/counter
- With noise filter in front end of input ports.
- Edge detector with three options:
  - Rising edge detection
  - Falling edge detection
  - Both edge detection
- Each input channel is supported one capture counter hold register.
- Captured event reset/reload capture counter option.
- Supports compare-match function.

6.18.3 Input Capture Timer/Counter Architecture

Each of the input capture timer/counter unit supports 3 input channels with three programmable input signal sources. The port pins IC0 to IC2 can be fed to the inputs of capture unit, besides, the QEI controller input signals (CHA, CHB and CHX), analog comparator outputs (CPO), OPA digital output (OPDOx), and ADC compare output (ADCMPOx) also can be internally routed to the capture inputs by software configuration. The following figures illustrate the architecture of Input Capture.

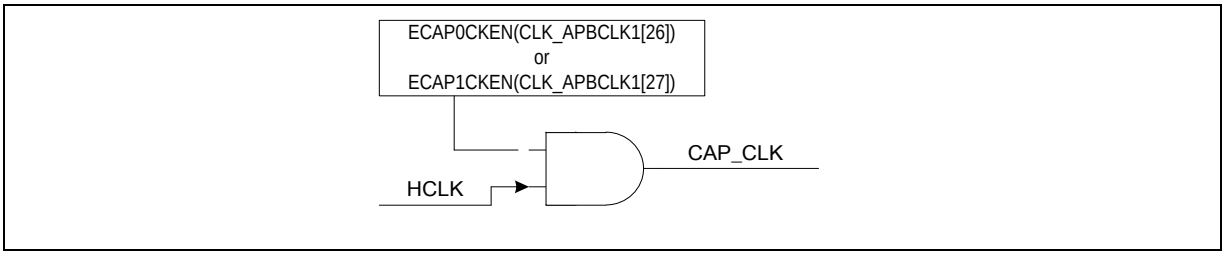


Figure 6.18-1 Input Capture Timer/Counter Clock Source Control

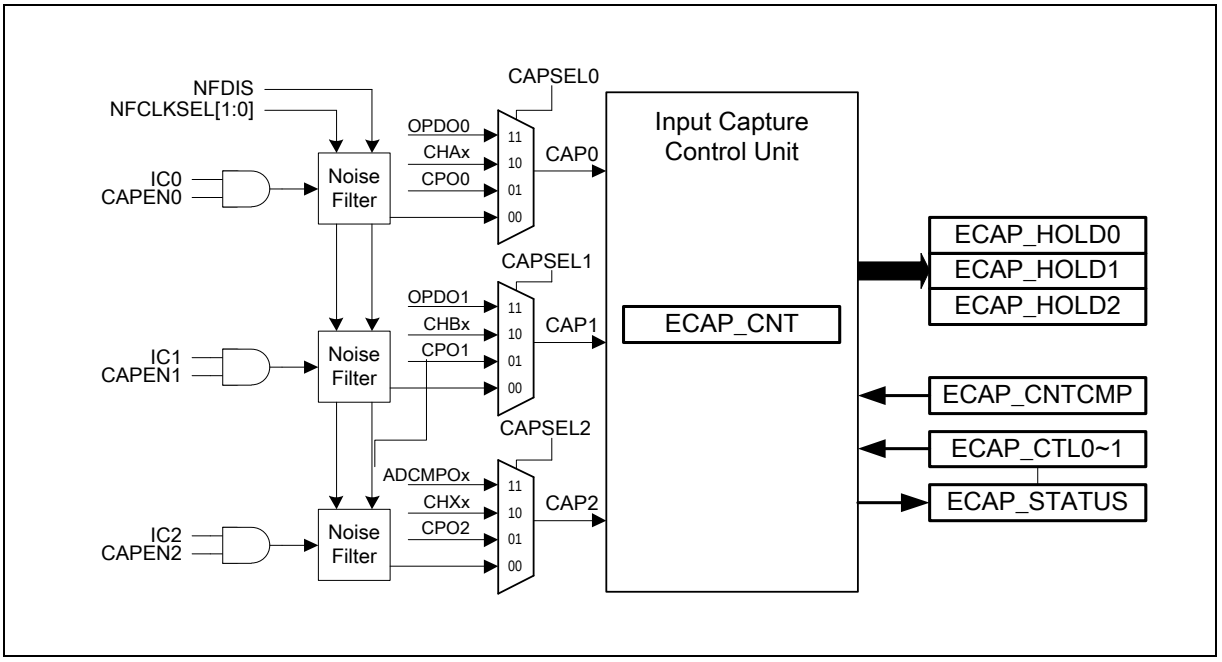


Figure 6.18-2 Input Capture Timer/Counter Architecture

6.18.4 Input Noise Filter

The architecture of input noise filter is similar to QEI controller with four sampling rate options. Refer to Figure 6.18-3 for details.

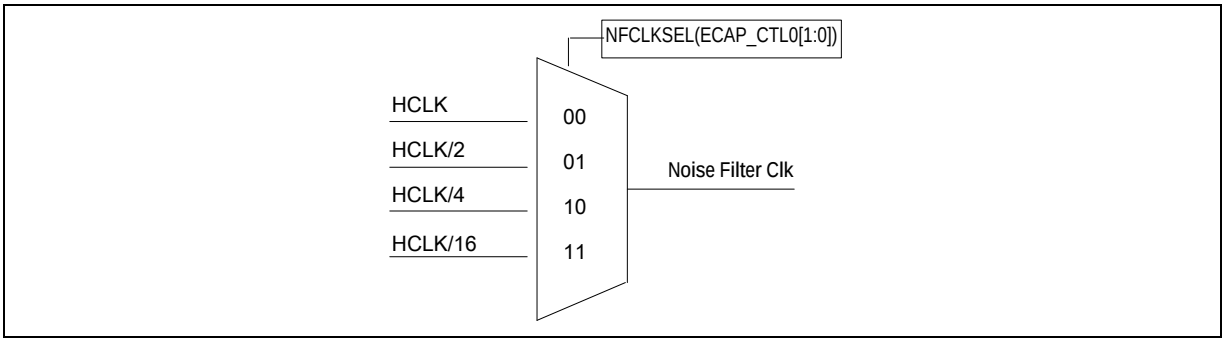


Figure 6.18-3 Noise Filter Sampling Clock Selection

6.18.5 Operation of Input Capture Timer/Counter

The capture modules are function to detect and measure pulse width and period of a square wave. Input channel 0 to 2 have their own edge detector but share with one capture timer/counter i.e. ECAP_CNT. The trigger option is programmable through EDGESELx (ECAP_CTL1[0]) register. It supports positive edge, negative edge and both edge triggers. Each capture module consists of an enable control bit, CAPEN0 to CAPEN2. The capture counter (ECAP_CNT) serves as a 24-bit up counter. It supports reload and compared modes. The Input Capture Timer/Counter Enable bit (CAPENx(ECAP_CTL0[16])) must be set to enable Input Capture Timer/Counter functions. More details are described in next sections.

6.18.5.1 Capture Function

Each time the capture input trigger is validated, the content of the free running 24 bits capture counter ECAP_CNT will be captured/transferred into the capture hold registers, ECAP_HOLD0~2, depending on which channel trigger. This action also causes the CAPF flag bits in ECAP_STATUS to be set, which will also generate an interrupt (if enabled by CAPIENx (ECAP_CTL0[16]). The CAPF0~2 flags are logical “OR” to the interrupt module. Flag is set by hardware and cleared by software. Software will have to resolve on the priority of the interrupt flags.

Setting the CPTCLR (ECAP_CTL0[26]), will allow hardware to reset capture counter (ECAP_CNT) automatically after the value of ECAP_CNT has been captured. Priority is given to reset counter after capturing the counter value in the capture register.

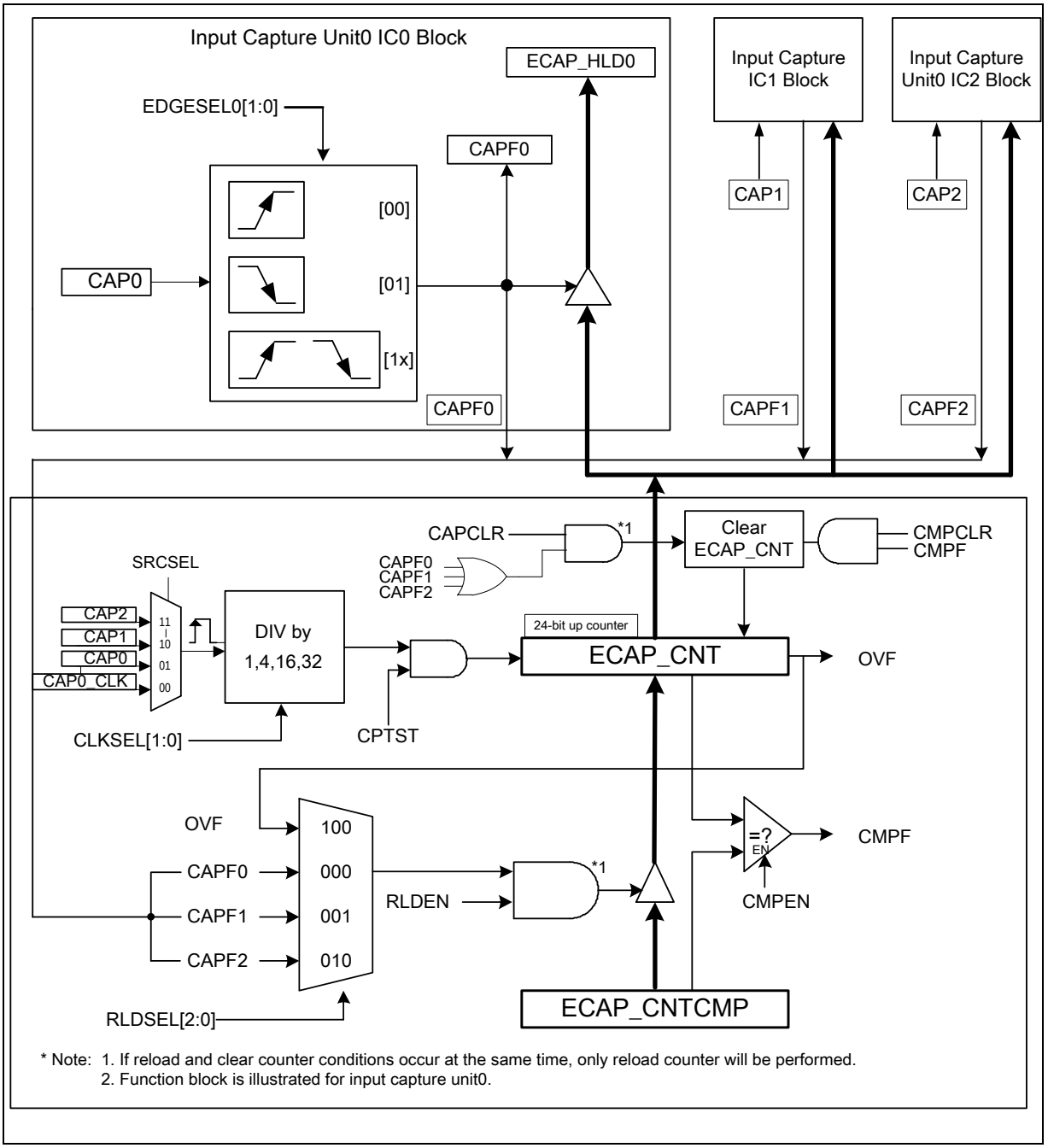


Figure 6.18-4 Input Capture Timer/Counter Functions Block

6.18.5.2 Compare Mode

The compare function is enabled by setting the CMPEN (ECAP_CTL0[28]) bit to 1. ECAP_CNTCMP will serves as a compare register. As ECAP_CNT counting up, upon matching with ECAP_CNTCMP value, CMPF (ECAP_STATUS[4]) will be set, which will generate an interrupt request if capture compare interrupt enable bit, CMPIEN (ECAP_CTL0[21]), is set. And then the timer reload from 0 and starts counting again.

Setting the CMPCLR (ECAP_CTL0[25]), will allow hardware to reset capture counter automatically after a match has occurred.

6.18.5.3 Reload Mode

Input Capture Timer/Counter can be also be configured for reload mode. The reload function is enabled by setting the RLDEN (ECAP_CTL0[27]) to 1. In this mode, ECAP_CNTCMP serves as a reload register. When ECAP_CNT overflows, a reload is generated that causes the contents of the ECAP_CNTCMP register to be reloaded into the ECAP_CNT register, if RLDEN is set. However, if RLDEN = 0, ECAP_CNT will be reload with 0, and count up again.

Alternatively, other reload source is also possible by the capture inputs by configuring the RLDSEL (ECAP_CTL1[10:8]). This action also sets the CAPFx flag bits in ECAP_STATUS register.

6.18.6 Input Capture Timer/Counter Interrupt Architecture

There are five interrupt sources for one input capture unit, each one has an interrupt flag and enable control bit, which can trigger Input Capture Timer/Counter Interrupt. Note that all the interrupt flags are set by hardware and must be cleared by software.

The following figure demonstrates the architecture of Input Capture Timer/Counter interrupts.

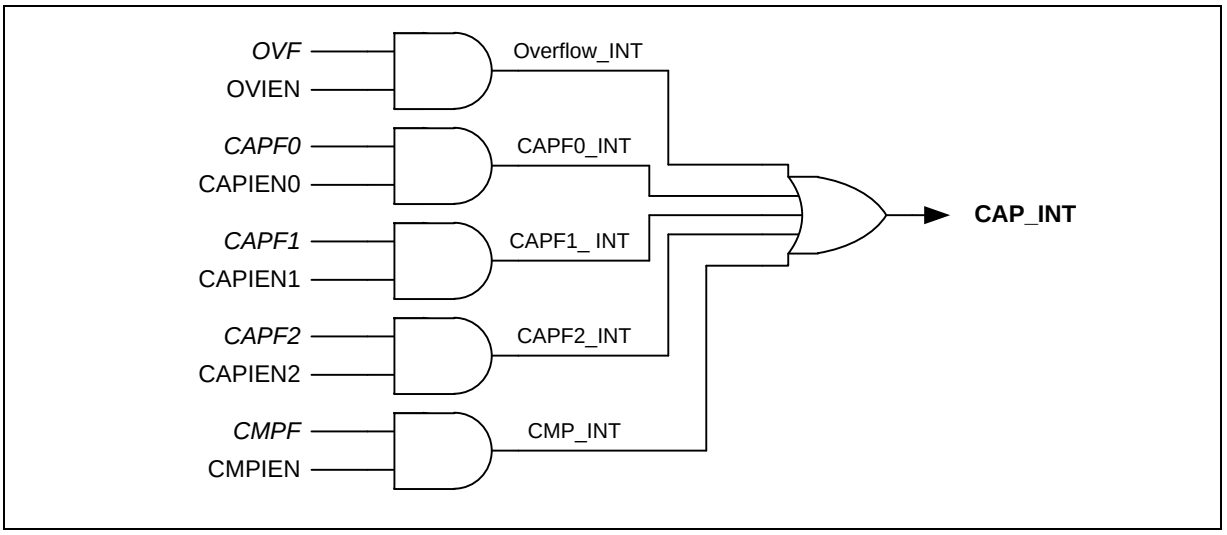


Figure 6.18-5 Input Capture Timer/Counter Interrupt Architecture Diagram

6.18.7 Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
ECAP Base Address: **ECAPn_BA = 0x400B_4000+ n *0x1000** **n=0,1**				
ECAP_CNT	ECAPn_BA+0x00	R/W	Input Capture Counter (24-bit up counter)	0x0000_0000
ECAP_HOLD0	ECAPn_BA+0x04	R/W	Input Capture Counter Hold Register 0	0x0000_0000
ECAP_HOLD1	ECAPn_BA+0x08	R/W	Input Capture Counter Hold Register 1	0x0000_0000
ECAP_HOLD2	ECAPn_BA+0x0C	R/W	Input Capture Counter Hold Register 2	0x0000_0000
ECAP_CNTCMP	ECAPn_BA+0x10	R/W	Input Capture Counter Compare Register	0x0000_0000
ECAP_CTL0	ECAPn_BA+0x14	R/W	Input Capture Control Register 0	0x0000_0000
ECAP_CTL1	ECAPn_BA+0x18	R/W	Input Capture Control Register 1	0x0000_0000
ECAP_STATUS	ECAPn_BA+0x1C	R/W	Input Capture Status Register	0x0000_0000

6.18.8 Register Description

Input Capture Counter (ECAP_CNT)

Register	Offset	R/W	Description	Reset Value
ECAP_CNT	ECAPn_BA+0x00	R/W	Input Capture Counter (24-bit up counter)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
VAL[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VAL[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VAL[7:0]							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**VAL**	**Input Capture Timer/Counter** The input Capture Timer/Counter is a 24-bit up-counting counter. The clock source for the counter is from the clock divider output which the CAP_CLK is software optionally divided by 1,4,16 or 32.

Input Capture Counter Hold Register (ECAP_HOLD0~2)

Register	Offset	R/W	Description	Reset Value
ECAP_HOLD0	ECAPn_BA+0x04	R/W	Input Capture Counter Hold Register 0	0x0000_0000
ECAP_HOLD1	ECAPn_BA+0x08	R/W	Input Capture Counter Hold Register 1	0x0000_0000
ECAP_HOLD2	ECAPn_BA+0x0C	R/W	Input Capture Counter Hold Register 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
VAL[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VAL[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VAL[7:0]							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**VAL**	**Input Capture Counter Hold Register** When an active input capture channel detects a valid edge signal change, the ECAP_CNT value is latched into the corresponding holding register. Each input channel has itself holding register named by ECAP_HOLDx where x is from 0 to 2 to indicate inputs from IC0 to IC2, respectively.

Input Capture Counter Compare Register (ECAP_CNTCMP)

Register	Offset	R/W	Description	Reset Value
ECAP_CNTCMP	ECAPn_BA+0x10	R/W	Input Capture Counter Compare Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
VAL[23:16]							
15	14	13	12	11	10	9	8
VAL[15:8]							
7	6	5	4	3	2	1	0
VAL[7:0]							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**VAL**	**Input Capture Counter Compare Register** If the compare function is enabled (CMPEN(ECAP_CTL0[28]) = 1), the compare register is loaded with the value that the compare function compares the capture counter (ECAP_CNT) with. If the reload control is enabled (RLDEN = 1), an overflow event or capture events will trigger the hardware to reload ECAP_CNTCMP into ECAP_CNT.

Input Capture Timer/Counter Control Register (ECAP_CTL0)

Register	Offset	R/W	Description	Reset Value
ECAP_CTL0	ECAPn_BA+0x14	R/W	Input Capture Control Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		CAPEN	CMPEN	RLDEN	CPTCLR	CMPCLR	CNTEN
23	22	21	20	19	18	17	16
Reserved		CMPIEN	OVIEN	Reserved	CAPIEN2	CAPIEN1	CAPIEN0
15	14	13	12	11	10	9	8
Reserved		CAPSEL2		CAPSEL1		CAPSEL0	
7	6	5	4	3	2	1	0
Reserved	CAPEN2	CAPEN1	CAPEN0	CAPNF_DIS	Reserved	NFDIS	

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29]	**CAPEN**	**Input Capture Timer/Counter Enable Bit** 0 = Input Capture function Disabled. 1 = Input Capture function Enabled.
[28]	**CMPEN**	**Compare Function Enable Bit** The compare function in input capture timer/counter is to compare the dynamic counting ECAP_CNT with the compare register ECAP_CNTCMP, if ECAP_CNT value reaches ECAP_CNTCMP, the flag CMPF will be set. 0 = The compare function Disabled. 1 = The compare function Enabled.
[27]	**RLDEN**	**Reload Function Enable Bit** Setting this bit to enable the reload function. If the reload control is enabled, an overflow event (OVF) or capture events (CAPFx) will trigger the hardware to reload ECAP_CNTCMP into ECAP_CNT. 0 = The reload function Disabled. 1 = The reload function Enabled.
[26]	**CPTCLR**	**Input Capture Counter Cleared By Capture Events Control** If this bit is set to 1, the capture counter (ECAP_CNT) will be cleared to zero when any one of capture events (CAPF0~3) occurs. 0 = Capture events (CAPF0~3) can clear capture counter (ECAP_CNT) Disabled. 1 = Capture events (CAPF0~3) can clear capture counter (ECAP_CNT) Enabled.
[25]	**CMPCLR**	**Input Capture Counter Cleared By Compare-Match Control** If this bit is set to 1, the capture counter (ECAP_CNT) will be cleared to 0 when the compare-match event (CAMCMPF = 1) occurs. 0 = Compare-match event (CAMCMPF) can clear capture counter (ECAP_CNT) Disabled. 1 = Compare-match event (CAMCMPF) can clear capture counter (ECAP_CNT) Enabled.

Bits	Description	
[24]	**CNTEN**	**Input Capture Counter Start** Setting this bit to 1, the capture counter (ECAP_CNT) starts up-counting synchronously with capture clock input (CAP_CLK). 0 = ECAP_CNT stop counting. 1 = ECAP_CNT starts up-counting.
[23:22]	**Reserved**	Reserved.
[21]	**CMPIEN**	**CMPF Trigger Input Capture Interrupt Enable Bit** 0 = The flag CMPF can trigger Input Capture interrupt Disabled. 1 = The flag CMPF can trigger Input Capture interrupt Enabled.
[20]	**OVIEN**	**OVF Trigger Input Capture Interrupt Enable Bit** 0 = The flag OVUNF can trigger Input Capture interrupt Disabled. 1 = The flag OVUNF can trigger Input Capture interrupt Enabled.
[19]	**Reserved**	Reserved.
[18]	**CAPIEN2**	**Input Capture Channel 2 Interrupt Enable Bit** 0 = The flag CAPF2 can trigger Input Capture interrupt Disabled. 1 = The flag CAPF2 can trigger Input Capture interrupt Enabled.
[17]	**CAPIEN1**	**Input Capture Channel 1 Interrupt Enable Bit** 0 = The flag CAPF1 can trigger Input Capture interrupt Disabled. 1 = The flag CAPF1 can trigger Input Capture interrupt Enabled.
[16]	**CAPIEN0**	**Input Capture Channel 0 Interrupt Enable Bit** 0 = The flag CAPF0 can trigger Input Capture interrupt Disabled. 1 = The flag CAPF0 can trigger Input Capture interrupt Enabled.
[15:14]	**Reserved**	Reserved.
[13:12]	**CAPSEL2**	**CAP2 Input Source Selection** 00 = CAP2 input is from port pin IC2. 01 = CAP2 input is from signal CPO2 (Analog comparator 2 output). 10 = CAP2 input is from signal CHX of QEI controller unit x. 11 = CAP2 input is from signal ADCMPOx (ADC compare output x). **Note:** Input capture unit n matches QEI or comparator unit x, where x = 0~1.
[11:10]	**CAPSEL1**	**CAP1 Input Source Selection** 00 = CAP1 input is from port pin IC1. 01 = CAP1 input is from signal CPO1 (Analog comparator 1 output). 10 = CAP1 input is from signal CHB of QEI controller unit x. 11 = CAP1 input is from signal OPDO1 (OP1 digital output). **Note:** Input capture unit n matches QEI or comparator unit x, where x = 0~1.
[9:8]	**CAPSEL0**	**CAP0 Input Source Selection** 00 = CAP0 input is from port pin IC0. 01 = CAP0 input is from signal CPO0 (Analog comparator 0 output). 10 = CAP0 input is from signal CHA of QEI controller unit x. 11 = CAP0 input is from signal OPDO0 (OP0 digital output). **Note:** Input capture unit n matches QEI or comparator unit x, where x = 0~1.
[7]	**Reserved**	Reserved.

Bits	Description	
[6]	**CAPEN2**	**Port Pin IC2 Input To Input Capture Unit Enable Bit** 0 = IC2 input to Input Capture Unit Disabled. 1 = IC2 input to Input Capture Unit Enabled.
[5]	**CAPEN1**	**Port Pin IC1 Input To Input Capture Unit Enable Bit** 0 = IC1 input to Input Capture Unit Disabled. 1 = IC1 input to Input Capture Unit Enabled.
[4]	**CAPEN0**	**Port Pin IC0 Input To Input Capture Unit Enable Bit** 0 = IC0 input to Input Capture Unit Disabled. 1 = IC0 input to Input Capture Unit Enabled.
[3]	**CAPNF_DIS**	**Input Capture Noise Filter Disable Bit** 0 = Noise filter of Input Capture Enabled. 1 = Noise filter of Input Capture Disabled.
[2]	**Reserved**	Reserved.
[1:0]	**NFDIS[1:0]**	**Noise Filter Clock Pre-Divide Selection** To determine the sampling frequency of the Noise Filter clock 00 = CAP_CLK. 01 = CAP_CLK/2. 10 = CAP_CLK/4. 11 = CAP_CLK/16.

Input Capture Timer/Counter Control Register (ECAP_CTL1)

Register	Offset	R/W	Description	Reset Value
ECAP_CTL1	ECAPn_BA+0x18	R/W	Input Capture Control Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						SRCSEL	
15	14	13	12	11	10	9	8
Reserved	CLKSEL			Reserved	RLDSEL		
7	6	5	4	3	2	1	0
Reserved		EDGESEL2		EDGESEL1		EDGESEL0	

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17:16]	**SRCSEL**	**Capture Timer/Counter Clock Source Selection** Select the capture timer/counter clock source. 00 = CAP_CLK (default). 01 = CAP0. 10 = CAP1. 11 = CAP2.
[15]	**Reserved**	Reserved.
[14:12]	**CLKSEL**	**Capture Timer Clock Divide Selection** The capture timer clock has a pre-divider with four divided options controlled by CLKSEL[1:0]. 000 = CAP_CLK/1. 001 = CAP_CLK/4. 010 = CAP_CLK/16. 011 = CAP_CLK/32. 100 = CAP_CLK/64. 101 = CAP_CLK/96. 110 = CAP_CLK/112. 111 = CAP_CLK/128.
[11]	**Reserved**	Reserved.

Bits	Description	
[10:8]	**RLDSEL[2:0]**	**ECAP_CNT Reload Trigger Source Selection** If the reload function is enabled RLDEN (ECAP_CTL0[27]) = 1, when a reload trigger event comes, the ECAP_CNT is reloaded with ECAP_CNTCMP. RLDSEL[2:0] determines the ECAP_CNT reload trigger source 000 = CAPF0. 001 = CAPF1. 010 = CAPF2. 100 = OVF. Others = Reserved.
[7:6]	**Reserved**	Reserved.
[5:4]	**EDGESEL2 [1:0]**	**Channel 2 Captured Edge Selection** Input capture can detect falling edge change or rising edge change only, or one of both edge changes. 00 = Detect rising edge. 01 = Detect falling edge. 1x = Detect either rising or falling edge.
[3:2]	**EDGESEL1[1:0]**	**Channel 1 Captured Edge Selection** Input capture can detect falling edge change only, rising edge change only or one of both edge change 00 = Detect rising edge. 01 = Detect falling edge. 1x = Detect either rising or falling edge.
[1:0]	**EDGESEL0[1:0]**	**Channel 0 Captured Edge Selection** Input capture can detect falling edge change only, rising edge change only or one of both edge change 00 = Detect rising edge. 01 = Detect falling edge. 1x = Detect either rising or falling edge.

Input Capture Timer/Counter Status Register (ECAP_STATUS)

Register	Offset	R/W	Description	Reset Value
ECAP_STATUS	ECAPn_BA+0x1C	R/W	Input Capture Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		OVF	CMPF	Reserved	CAPF2	CAPF1	CAPF0

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**OVF**	**Input Capture Counter Overflow Flag** Flag is set by hardware when input capture up counter (ECAP_CNT) overflows from 0x00FF_FFFF to zero. 0 = No overflow occurs in ECAP_CNT. 1 = ECAP_CNT overflows. **Note:** This bit is only cleared by writing 1 to it.
[4]	**CMPF**	**Input Capture Compare-Match Flag** If the input capture compare function is enabled, the flag is set by hardware while capture counter (ECAP_CNT) up counts and reach to the ECAP_CNTCMP value. 0 = ECAP_CNT does not match with ECAP_CNTCMP value. 1 = ECAP_CNT counts to the same as ECAP_CNTCMP value. **Note:** This bit is only cleared by writing 1 to it.
[3]	**Reserved**	Reserved.
[2]	**CAPF2**	**Input Capture Channel 2 Captured Flag** When the input capture channel 2 detects a valid edge change at CAP2 input, it will set flag CAPF2 to high. 0 = No valid edge change is detected at CAP2 input. 1 = A valid edge change is detected at CAP2 input. **Note:** This bit is only cleared by writing 1 to it.
[1]	**CAPF1**	**Input Capture Channel 1 Captured Flag** When the input capture channel 1 detects a valid edge change at CAP1 input, it will set flag CAPF1 to high. 0 = No valid edge change is detected at CAP1 input. 1 = A valid edge change is detected at CAP1 input. **Note:** This bit is only cleared by writing 1 to it.

Bits	Description	
[0]	**CAPF0**	**Input Capture Channel 0 Captured Flag** When the input capture channel 0 detects a valid edge change at CAP0 input, it will set flag CAPF0 to high. 0 = No valid edge change is detected at CAP0 input. 1 = A valid edge change is detected at CAP0 input. **Note:** This bit is only cleared by writing 1 to it.

6.19 OP Amplifier

6.19.1 Overview

This device integrated two operational amplifiers. It can be enabled through OPENx (OPA_CTL[1:0]) bits. User can measure the outputs of the OP amplifier as the OP amplifier output to the integrated EADC channel EADC0_11 and EADC1_8, where digital results can be taken.

6.19.2 Features

- Analog input voltage range: 0~Vdd.
- Two analog OP amplifiers
- Software enabled to connect OP amplifier 0,1 outputs to A/D converter channel AINA[B], AINB[8] respectively
- Schmitt trigger buffer outputs of OP amplifier0, 1 can be as one of the comparator interrupt sources.
- OP amplifier 0 output can be an optional input source of integrated comparator 0 positive input
- OP amplifier 1 output can be an optional input source of integrated comparator 1 positive input

6.19.3 Block Diagram

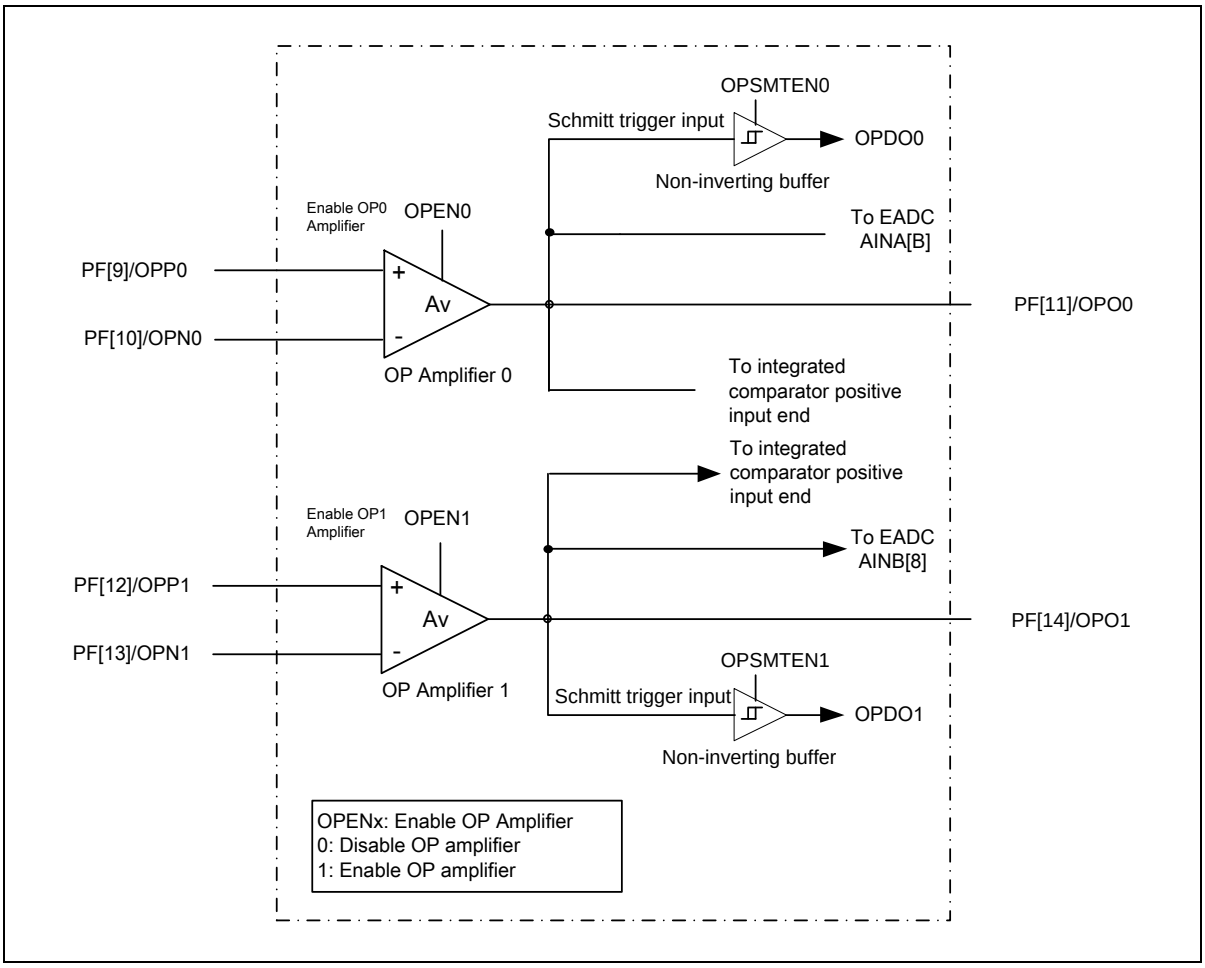


Figure 6.19-1 OP Amplifier Block Diagram

6.19.4 Interrupt Sources

OPDF0 (OPA_STATUS[4]) and OPDF1 (OPA_STATUS[5]) interrupt flag are set respectively by hardware whenever the OP0, 1 amplifier Schmitt trigger non-inverting buffer output change states. The flag bit is cleared by writing 1 to it. Schmitt trigger buffer outputs of OP amplifier0, 1 can be as one of the comparator interrupt sources.

6.19.5 Register Map

R: read only, **W**: write only, **R/W**: both read and write, **C**: Only value 0 can be written

Register	Offset	R/W	Description	Reset Value
OPS Base Address: **OPA_BA = 0x4004_6000**				
OPA_CTL	OPA_BA+0x00	R/W	OP Amplifier Control Register	0x0000_0000
OPA_STATUS	OPA_BA+0x04	R/W	OP Amplifier Status Register	0x0000_0000

6.19.6 Register Description

OPA Control Register (OPA_CTL)

Register	Offset	R/W	Description	Reset Value
OPA_CTL	OPA_BA+0x00	R/W	OP Amplifier Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**OPAIE1**	**OPAIE0**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**Reserved**	**OPSMTEN1**	**OPSMTEN0**	**Reserved**	**Reserved**	**OPEN1**	**OPEN0**

Bits	Description	
[31:10]	**Reserved**	Reserved.
[9]	**OPAIE1**	**OP Amplifier 1 Schmitt Trigger Digital Output Interrupt Enable Bit** 0 = OP Amplifier 1 digital output interrupt function Disabled. 1 = OP Amplifier 1 digital output interrupt function Enabled. OPDF1 interrupt flag is set by hardware whenever the OP amplifier 1 Schmitt trigger non-inverting buffer digital output changes state, in the meanwhile, if OPAIE1 is set to 1, a comparator interrupt request is generated.
[8]	**OPAIE0**	**OP Amplifier 0 Schmitt Trigger Digital Output Interrupt Enable Bit** 0 = OP Amplifier 0 digital output interrupt function Disabled. 1 = OP Amplifier 0 digital output interrupt function Enabled. The OPDF0 interrupt flag is set by hardware whenever the OP amplifier 0 Schmitt trigger non-inverting buffer digital output changes state, in the meanwhile, if OPAIE0 is set to 1, a comparator interrupt request is generated.
[7:6]	**Reserved**	Reserved.
[5]	**OPSMTEN1**	**OP Amplifier 1 Schmitt Trigger Non-Inverting Buffer Enable Bit** 0 = Disabled. 1 = Enabled.
[4]	**OPSMTEN0**	**OP Amplifier 0 Schmitt Trigger Non-Inverting Buffer Enable Bit** 0 = Disabled. 1 = Enabled.
[3:2]	**Reserved**	Reserved.

Bits	Description	
[1]	**OPEN1**	**OP Amplifier 1 Enable Bit** 0 = Disabled. 1 = Enabled. **Note:** OP Amplifier 1 output needs wait stable 20μs after OPEN1 is first set.
[0]	**OPEN0**	**OP Amplifier 0 Enable Bit** 0 = Disabled. 1 = Enabled. **Note:** OP Amplifier 0 output needs wait stable 20μs after OPEN0 is first set.

OPA Status Register (OPA_STATUS)

Register	Offset	R/W	Description	Reset Value
OPA_STATUS	OPA_BA+0x04	R/W	OP Amplifier Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		OPDF1	OPDF0	Reserved		OPDO1	OPDO0

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**OPDF1**	**OP Amplifier 1 Schmitt Trigger Digital Output Interrupt Flag** OPDF1 interrupt flag is set by hardware whenever the OP amplifier 1 Schmitt trigger non-inverting buffer digital output changes state. **This bit is cleared by writing 1 to it.**
[4]	**OPDF0**	**OP Amplifier 0 Schmitt Trigger Digital Output Interrupt Flag** OPDF0 interrupt flag is set by hardware whenever the OP amplifier 0 Schmitt trigger non-inverting buffer digital output changes state. **This bit is cleared by writing 1 to it.**
[3:2]	**Reserved**	Reserved.
[1]	**OPDO1**	**OP Amplifier 1 Digital Output** Synchronized to the APB clock to allow reading by software. Cleared when the Schmitt trigger buffer is disabled (OPSMTEN1 = 0).
[0]	**OPDO0**	**OP Amplifier 0 Digital Output** Synchronized to the APB clock to allow reading by software. Cleared when the Schmitt trigger buffer is disabled (OPSMTEN0 = 0).

6.20 PS/2 Device Controller (PS2D)

6.20.1 Overview

The PS/2 device controller provides basic timing control for PS/2 communication. All communication between the device and the host is managed through the CLK and DATA pins. Unlike PS/2 keyboard or mouse device controller, the received/transmit code needs to be translated as meaningful code by firmware. The device controller generates the CLK signal after receiving a request to send, but host has ultimate control over communication. DATA sent from the host to the device is read on the rising edge and DATA sent from device to the host is change after rising edge. A 16 bytes FIFO is used to reduce CPU intervention. Software can select 1 to 16 bytes for a continuous transmission.

6.20.2 Features

- Host communication inhibit and request to send detection
- Reception frame error detection
- Programmable 1 to 16 bytes transmit buffer to reduce CPU intervention
- Double buffer for data reception
- Software override bus

6.20.3 Block Diagram

The PS/2 device controller consists of APB interface and timing control logic for DATA and CLK lines.

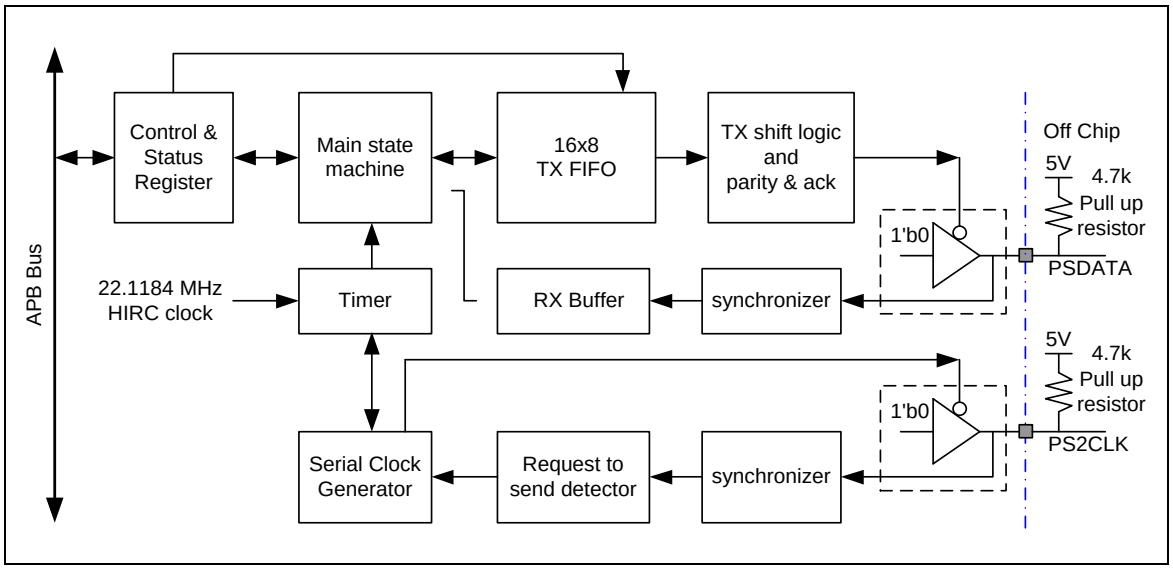


Figure 6.20-1 PS/2 Device Block Diagram

6.20.4 Functional Description

6.20.4.1 Communication

The PS/2 device implements a bidirectional synchronous serial protocol. The bus is "Idle" when both lines are high (open-collector). This is the only state where the device is allowed start to transmit DATA. The host has ultimate control over the bus and may inhibit communication at any time by pulling the CLK line low.

The CLK signal is generated by PS/2 device. If the host wants to send DATA, it must first inhibit communication from the device by pulling CLK low. The host then pulls DATA low and releases CLK. This is the "Request-to-Send" state and signals the device to start generating CLK pulses.

DATA	CLK	Bus State
High	High	Idle
High	Low	Communication Inhibit
Low	High	Host Request to Send

All data is transmitted one byte at a time and each byte is sent in a frame consisting of 11 or 12 bits. These bits are:

- 1 start bit. This is always 0
- 8 DATA bits, least significant bit first
- 1 parity bit (odd parity)
- 1 stop bit. This is always 1
- 1 acknowledge bit (host-to-device communication only)

The parity bit is set if there is an even number of 1's in the data bits and cleared to 0 if there is an odd number of 1's in the data bits. The number of 1's in the data bits plus the parity bit always adds up to an odd number set to 1. This is used for error detection. The device must check this bit and if incorrect it should respond as if it had received an invalid command.

The host may inhibit communication at any time by pulling the CLK line low for at least 100 microseconds. If a transmission is inhibited before the 11th clock pulse, the device must abort the current transmission and prepare to retransmit the current data when host releases Clock. In order to reserve enough time for software to decode host command, the transmit logic is blocked by RXIF(PS2_INTSTS[0]) bit, the software must clear RXIF(PS2_INTSTS[0]) bit to start retransmit. Software can write CLRFIFO(PS2_CTL[8]) to 1 to reset FIFO pointer if need.

Device-to-Host

The device uses a serial protocol with 11-bit frames. These bits are:

- 1 start bit. This is always 0
- 8 DATA bits, least significant bit first
- 1 parity bit (odd parity)
- 1 stop bit. This is always 1

The device writes a bit on the DATA line when CLK is high, and it is read by the host when CLK is low, which is illustrated in the following figure.

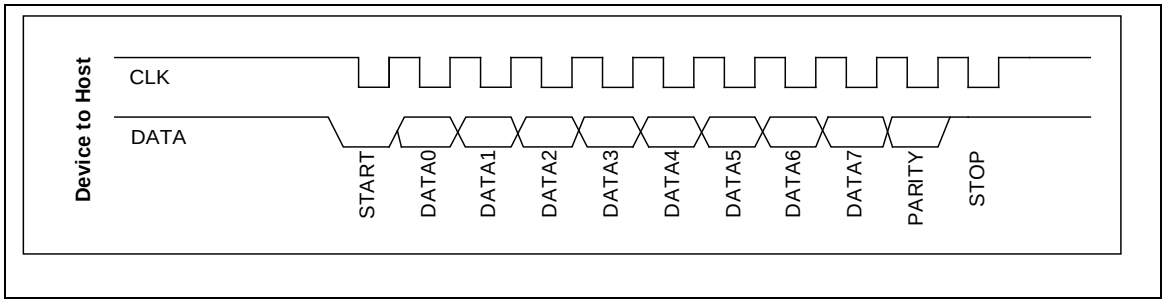


Figure 6.20-2 Data Format of Device-to-Host

Host-to-Device:

The PS/2 device always generates the CLK signal. If the host wants to send DATA, it must first put the CLK and DATA lines in a "Request-to-send" state as follows:

- Inhibit communication by pulling CLK low for at least 100 microseconds
- Apply "Request-to-send" by pulling DATA low, and then release CLK

The device should check for the state at intervals not to exceed 10 milliseconds. When the device detects this state, it will begin generating CLK signals and CLK in eight DATA bits and one stop bit. The host changes the DATA line only when the CLK line is low, and DATA is read by the device when CLK is high.

After the stop bit is received, the device will acknowledge the received byte by bringing the DATA line low and generating one last CLK pulse. If the host does not release the DATA line after the 11th CLK pulse, the device will continue to generate CLK pulses until the DATA line is released.

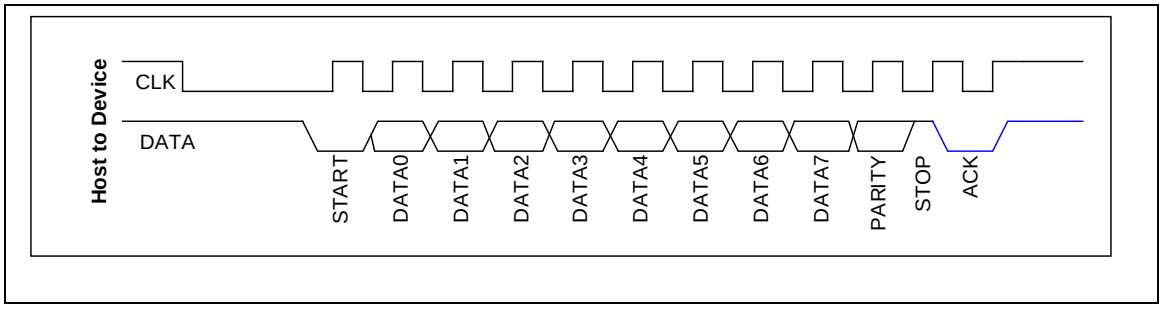


Figure 6.20-3 Data Format of Host-to-Device

The detailed host and the device DATA and CLK timing for communication is shown in Figure 6.20-4.

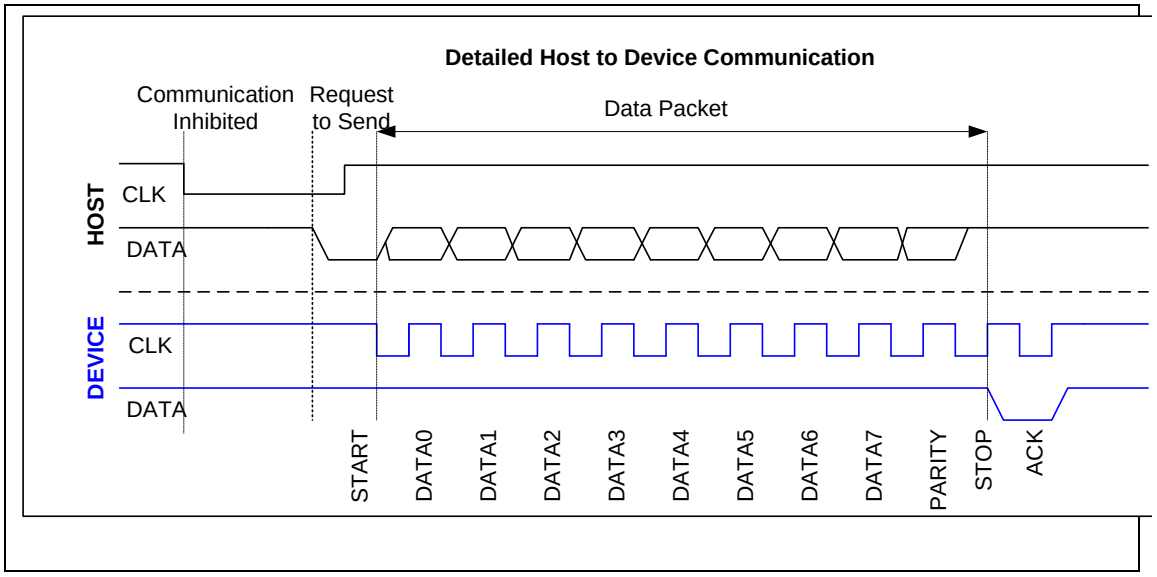


Figure 6.20-4 PS/2 Bit Data Format

6.20.4.2 PS/2 Bus Timing Specification

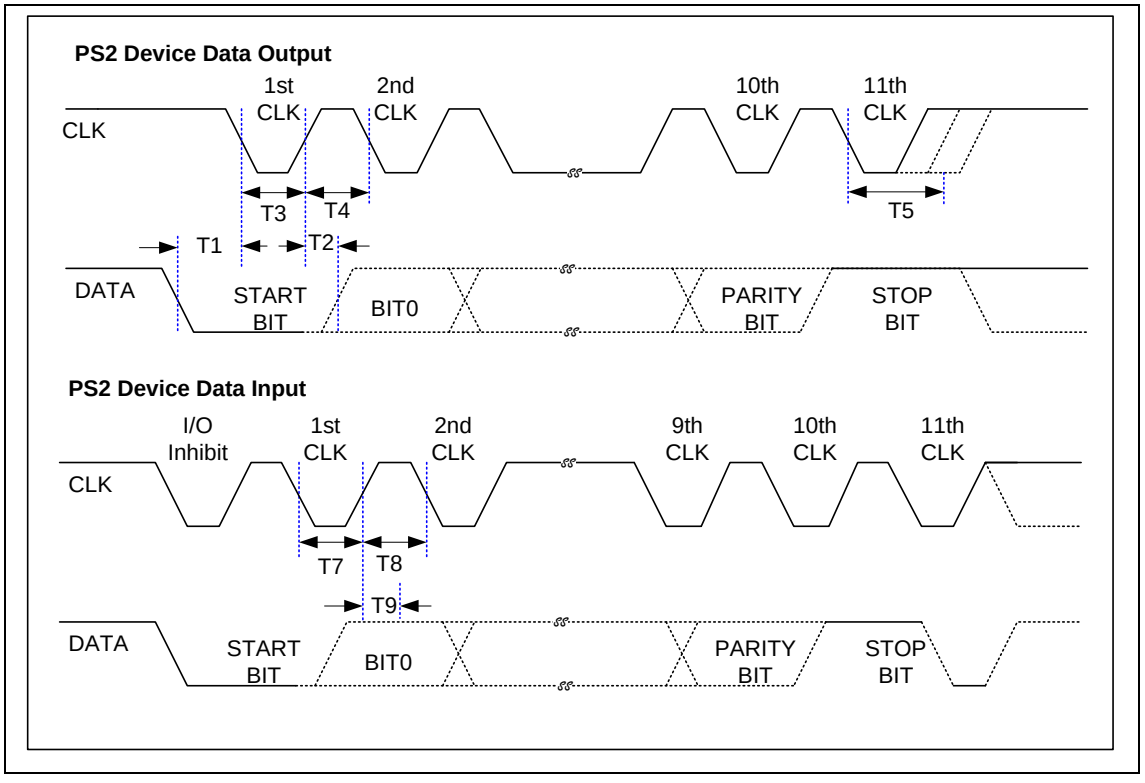


Figure 6.20-5 PS/2 Bus Timing

Symbol	Timing Parameter	Min	Max
T1	DATA transition to the falling edge of CLK	5us	25us
T2	Rising edge of CLK to DATA transition	5us	T4-5us
T3	Duration of CLK inactive	30us	50us
T4	Duration of CLK active	30us	50us
T5	Time to auxiliary device inhibit after 11th clock to ensure auxiliary device does not start another transmission	>0	50us
T7	Duration of CLK inactive	30us	50us
T8	Duration of CLK active	30us	50us
T9	Time from inactive to active CLK transition, used for time auxiliary device sample DATA	5us	25us

6.20.4.3 TX FIFO Operation

Writing PS2_TXDAT0 register starts device to host communication. Software is required to define TXFDEPTH(PS2_CTL[6:3]) before writing transmission data to TX FIFO. 1st START bit is sent to PS/2 bus 100us after software writes TX FIFO, if there is more than 4 bytes data need to be sent, Software can write residual data to PS2_TXDAT1-3 before 4th byte transmit complete. A time delay 100us is added between two consecutive bytes.

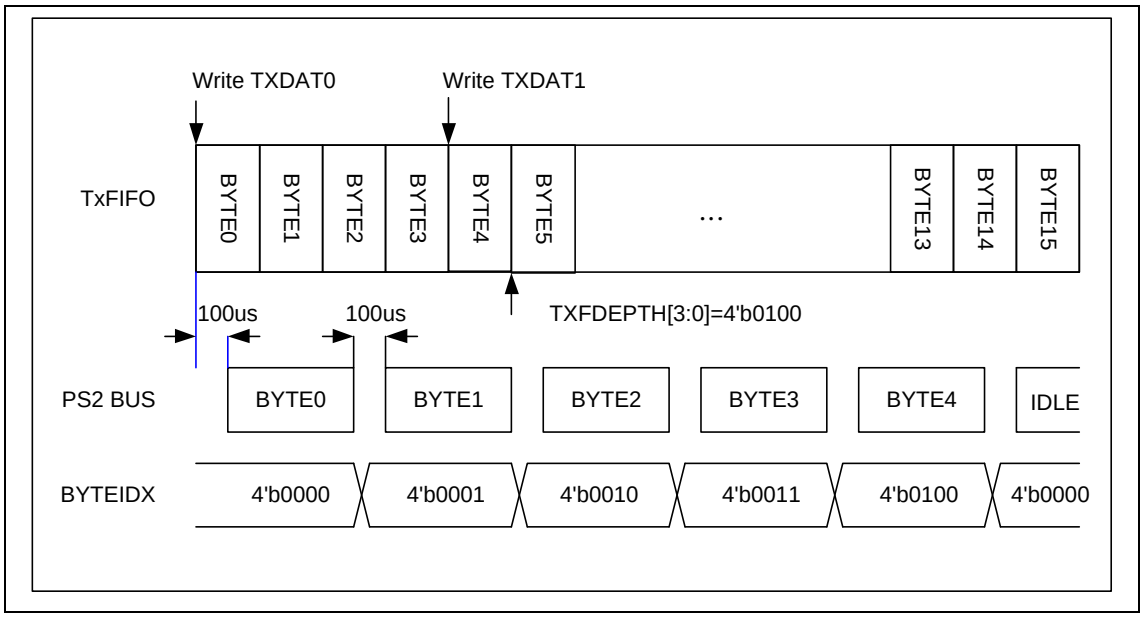


Figure 6.20-6 PS/2 Data Format

6.20.5 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
PS2 Base Address: **PS2_BA=0x400E_0000**				
PS2_CTL	PS2_BA+0x00	R/W	PS/2 Control Register	0x0000_0000
PS2_TXDAT0	PS2_BA+0x04	R/W	PS/2 Transmit DATA Register 0	0x0000_0000
PS2_TXDAT1	PS2_BA+0x08	R/W	PS/2 Transmit DATA Register 1	0x0000_0000
PS2_TXDAT2	PS2_BA+0x0C	R/W	PS/2 Transmit DATA Register 2	0x0000_0000
PS2_TXDAT3	PS2_BA+0x10	R/W	PS/2 Transmit DATA Register 3	0x0000_0000
PS2_RXDAT	PS2_BA+0x14	R	PS/2 Receive DATA Register	0x0000_0000
PS2_STATUS	PS2_BA+0x18	R/W	PS/2 Status Register	0x0000_0083
PS2_INTSTS	PS2_BA+0x1C	R/W	PS/2 Interrupt Status Register	0x0000_0000

6.20.6 Register Description

PS/2 Control Register (PS2_CTL)

Register	Offset	R/W	Description	Reset Value
PS2_CTL	PS2_BA+0x00	R/W	PS/2 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**FPS2DAT**	**FPS2CLK**	**OVERRIDE**	**CLRFIFO**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ACK	**TXFDEPTH**				**RXIEN**	**TXIEN**	**PS2EN**

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11]	**FPS2DAT**	**Force DATSTAT Line** It forces DATSTAT high or low regardless of the internal state of the device controller if OVERRIDE is set to high. 0 = Force DATSTAT low. 1 = Force DATSTAT high.
[10]	**FPS2CLK**	**Force CLKSTAT Line** It forces CLKSTAT line high or low regardless of the internal state of the device controller if OVERRIDE is set to high. 0 = Force CLKSTAT line low. 1 = Force CLKSTAT line high.
[9]	**OVERRIDE**	**Software Override PS/2 CLK/DATA Pin State** 0 = CLKSTAT and DATSTAT pins are controlled by internal state machine. 1 = CLKSTAT and DATSTAT pins are controlled by software.
[8]	**CLRFIFO**	**Clear TX FIFO** Write 1 to this bit to terminate device to host transmission. The TXEMPTY(PS2_STATUS[7]) bit will be set to 1 and pointer BYTEIDEX(PS2_STATUS[11:8]) is reset to 0 regardless there is residue data in buffer or not. The buffer content is not been cleared. 0 = Not active. 1 = Clear FIFO.
[7]	**ACK**	**Acknowledge Enable Bit** 0 = Always sends acknowledge to host at 12th clock for host to device communication. 1 = If parity error or stop bit is not received correctly, acknowledge bit will not be sent to host at 12th clock.

[6:3]	**TXFDEPTH**	**Transmit Data FIFO Depth** There is 16-byte buffer for data transmit. Software can define the FIFO depth from 1 to 16 bytes depending on the application. 0 = 1 byte. 1 = 2 bytes. … 14 = 15 bytes. 15 = 16 bytes.
[2]	**RXIEN**	**Receive Interrupt Enable Bit** 0 = Data receive complete interrupt Disabled. 1 = Data receive complete interrupt Enabled.
[1]	**TXIEN**	**Transmit Interrupt Enable Bit** 0 = Data transmit complete interrupt Disabled. 1 = Data transmit complete interrupt Enabled.
[0]	**PS2EN**	**PS/2 Device Enable Bit** Enable PS/2 device controller. 0 = Disabled. 1 = Enabled.

PS/2 TX DATA Register 0-3 (PS2_TXDAT0-3)

Register	Offset	R/W	Description	Reset Value
PS2_TXDAT0	PS2_BA+0x04	R/W	PS/2 Transmit DATA Register 0	0x0000_0000
PS2_TXDAT1	PS2_BA+0x08	R/W	PS/2 Transmit DATA Register 1	0x0000_0000
PS2_TXDAT2	PS2_BA+0x0C	R/W	PS/2 Transmit DATA Register 2	0x0000_0000
PS2_TXDAT3	PS2_BA+0x10	R/W	PS/2 Transmit DATA Register 3	0x0000_0000

31	30	29	28	27	26	25	24
DAT[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DAT[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DAT[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DAT[7:0]							

Bits	Description	
[31:0]	**DAT**	**Transmit Data** Write data to this register starts device to host communication if bus is in IDLE state. Software must enable PS2EN(PS2_CTL[0]) before writing data to TX buffer.

PS/2 Receiver DATA Register (PS2_RXDAT)

Register	Offset	R/W	Description	Reset Value
PS2_RXDAT	PS2_BA+0x14	R	PS/2 Receive DATA Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DAT[7:0]							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**DAT**	**Received Data** For host to device communication, after acknowledge bit is sent, the received data is copied from receive shift register to PS2_RXDAT register. CPU must read this register before next byte reception complete; otherwise, the data will be overwritten and RXOV(PS2_STATUS[6]) bit will be set to 1.

PS/2 Status Register (PS2_STATUS)

Register	Offset	R/W	Description	Reset Value
PS2_STATUS	PS2_BA+0x18	R/W	PS/2 Status Register	0x0000_0083

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				BYTEIDX			
7	6	5	4	3	2	1	0
TXEMPTY	RXOV	TXBUSY	RXBUSY	RXPARITY	FRAMEERR	DATSTAT	CLKSTAT

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11:8]	**BYTEIDX**	**Byte Index** It indicates which data byte in transmit data shift register. When all data in FIFO is transmitted and it will be cleared to 0. **Note:** This bit is read only.
[7]	**TXEMPTY**	**TX FIFO Empty** When software writes any data to PS2_TXDAT0-3 the TXEMPTY bit is cleared to 0 immediately if PS2EN is enabled. When transmitted data byte number is equal to FIFODEPTH then TXEMPTY bit is set to 1. 0 = There is data to be transmitted. 1 = FIFO is empty. **Note:** This bit is read only.
[6]	**RXOV**	**RX Buffer Overwrite** 0 = No overwrite. 1 = Data in PS2_RXDAT register is overwritten by new received data.

BYTEIDX	DATA Transmit	BYTEIDX	DATA Transmit
0000	TXDATA0[7:0]	1000	TXDATA2[7:0]
0001	TXDATA0[15:8]	1001	TXDATA2[15:8]
0010	TXDATA0[23:16]	1010	TXDATA2[23:16]
0011	TXDATA0[31:24]	1011	TXDATA2[31:24]
0100	TXDATA1[7:0]	1100	TXDATA3[7:0]
0101	TXDATA1[15:8]	1101	TXDATA3[15:8]
0110	TXDATA1[23:16]	1110	TXDATA3[23:16]
0111	TXDATA1[31:24]	1111	TXDATA3[31:24]

		Note: Write 1 to clear this bit.
[5]	**TXBUSY**	**Transmit Busy** This bit indicates that the PS/2 device is currently sending data. 0 = Idle. 1 = Currently sending data. **Note:** This bit is read only.
[4]	**RXBUSY**	**Receive Busy** This bit indicates that the PS/2 device is currently receiving data. 0 = Idle. 1 = Currently receiving data. **Note:** This bit is read only.
[3]	**RXPARITY**	**Received Parity** This bit reflects the parity bit for the last received data byte (odd parity). **Note:** This bit is read only.
[2]	**FRAMEERR**	**Frame Error** For host to device communication, if STOP bit (logic 1) is not received it is a frame error. If frame error occurs, DATA line may keep at low state after 12th clock. At this moment, software overrides CLKSTAT to send clock till DATSTAT release to high state. After that, device sends a “Resend” command to host. 0 = No frame error. 1 = Frame error occurred . **Note:** Write 1 to clear this bit.
[1]	**DATSTAT**	**DATA Pin State** This bit reflects the status of the DATSTAT line after synchronizing and sampling.
[0]	**CLKSTAT**	**CLK Pin State** This bit reflects the status of the CLKSTAT line after synchronizing.

PS/2 Interrupt Identification Register (PS2_INTSTS)

Register	Offset	R/W	Description	Reset Value
PS2_INTSTS	PS2_BA+0x1C	R/W	PS/2 Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**TXIF**	**RXIF**

Bits	Description	
[31:3]	**Reserved**	Reserved.
[1]	**TXIF**	**Transmit Interrupt** This bit is set to 1 after STOP bit is transmitted. Interrupt occurs if TXIEN(PS2_CTL[1]) bit is set to 1. 0 = No interrupt. 1 = Transmit interrupt occurred. **Note:** Write 1 to clear this bit to 0.
[0]	**RXIF**	**Receive Interrupt** This bit is set to 1 when acknowledge bit is sent for Host to device communication. Interrupt occurs if RXIEN(PS2_CTL[2]) bit is set to 1. 0 = No interrupt. 1 = Receive interrupt occurred. **Note:** Write 1 to clear this bit to 0.

6.21 PWM Generator and Capture Timer (PWM)

6.21.1 Overview

The NUC442/NUC472 has two PWM generators – PWM0 and PWM1. PWM0 supports 6 channels PWM output or 6 channels input capture, and these two functions share the same pins (PWM0_CH0/ PWM0_CH1/PWM0_CH2/PWM0_CH3/PWM0_CH4/PWM0_CH5). PWM1 supports 6 channels PWM output or 6 channels input capture, and these two functions share the same pins (PWM1_CH0/PWM1_CH1/PWM1_CH2/ PWM1_CH3/PWM1_CH4/PWM1_CH5).

The PWM generator has 16-bit PWM counter and comparator, and the PWM counter supports edge-aligned or center-aligned operating types. The PWM generator supports two standard PWM output modes: Independent output mode and Complementary output mode with 8-bit Dead-zone generator. In addition, PWM generator supports two special output mode: Synchronize mode and Group mode. It also has 8-bit prescaler and clock divider with 5 divided frequencies (1, 1/2, 1/4, 1/8, 1/16) to support wide range clock frequency of PWM counter. For PWM output control unit, it supports polarity output function, independent pin mask function and brake function. PWM generator can send ADC start trigger signal if one of the following conditions happened: PWM counter period point, PWM counter center point, PWM output rising edge and PWM output falling edge.

The PWM generator also supports input capture function. It supports latch PWM counter value to corresponding register when input channel has a rising transition, falling transition or both transition is happened.

6.21.2 Features

6.21.2.1 PWM function features

- Supports 12 PWM output channels with 16-bit resolution
- Supports 8-bit prescaler and clock divider
- Supports period point, center point and edge point PWM Interrupt
- Supports One-shot or Auto-reload PWM counter operation mode
- Supports Edge-aligned or Center-aligned PWM counter type
- Supports 8-bit dead zone with maximum divided 16 prescaler
- Supports brake function source from pin or comparator output
- Supports mask function for each PWM pin
- Supports independent, complementary, synchronized and group PWM output mode
- Supports trigger ADC start conversion at PWM counter period point, PWM counter center point, PWM output rising edge and PWM output falling edge

6.21.2.2 Capture Function Features

- Supports 12 Capture input channels with 16-bit resolution
- Supports rising or falling capture condition
- Supports capture interrupt

6.21.3 Block Diagram

The following six figures illustrate the architecture of PWM in pair (e.g. PWM-Timer 0/1 are in one pair, PWM-Timer 2/3 are in one pair and PWM-Timer 4/5 are in another one).

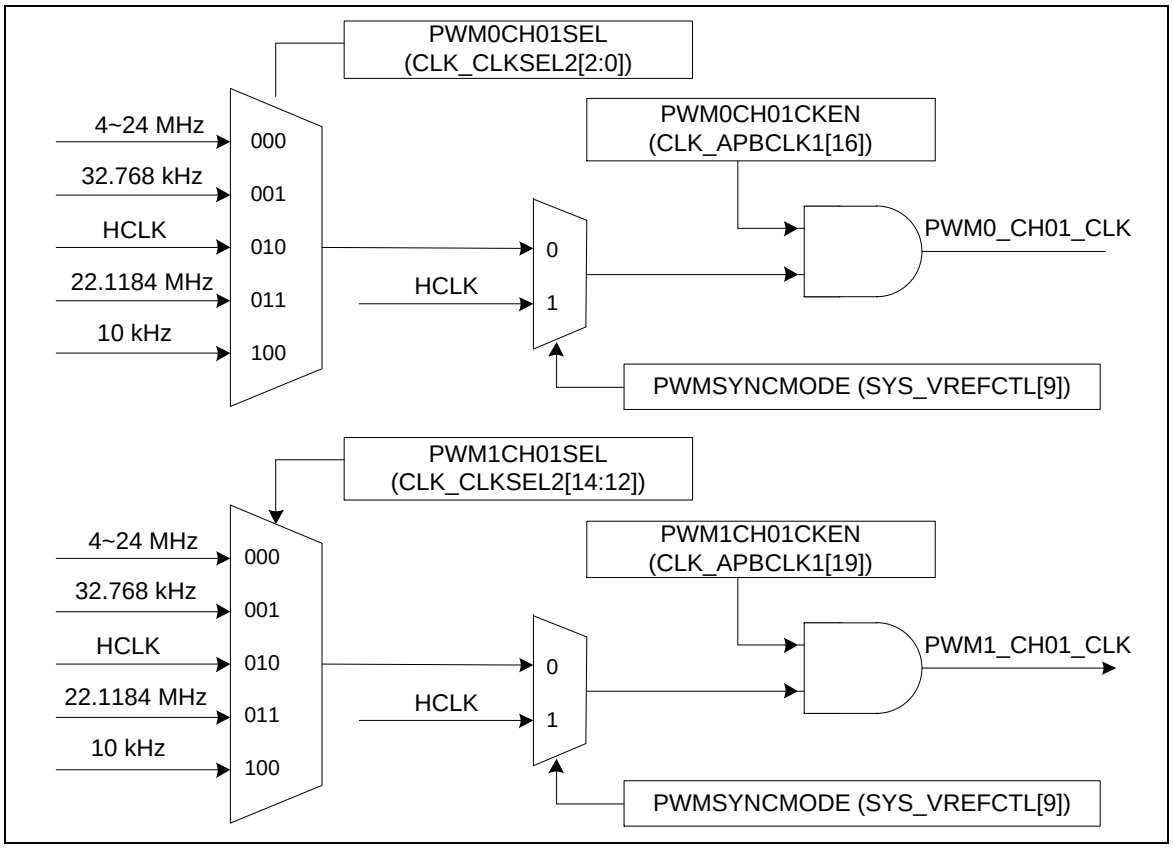


Figure 6.21-1 PWM Generator 0 Clock Source Control

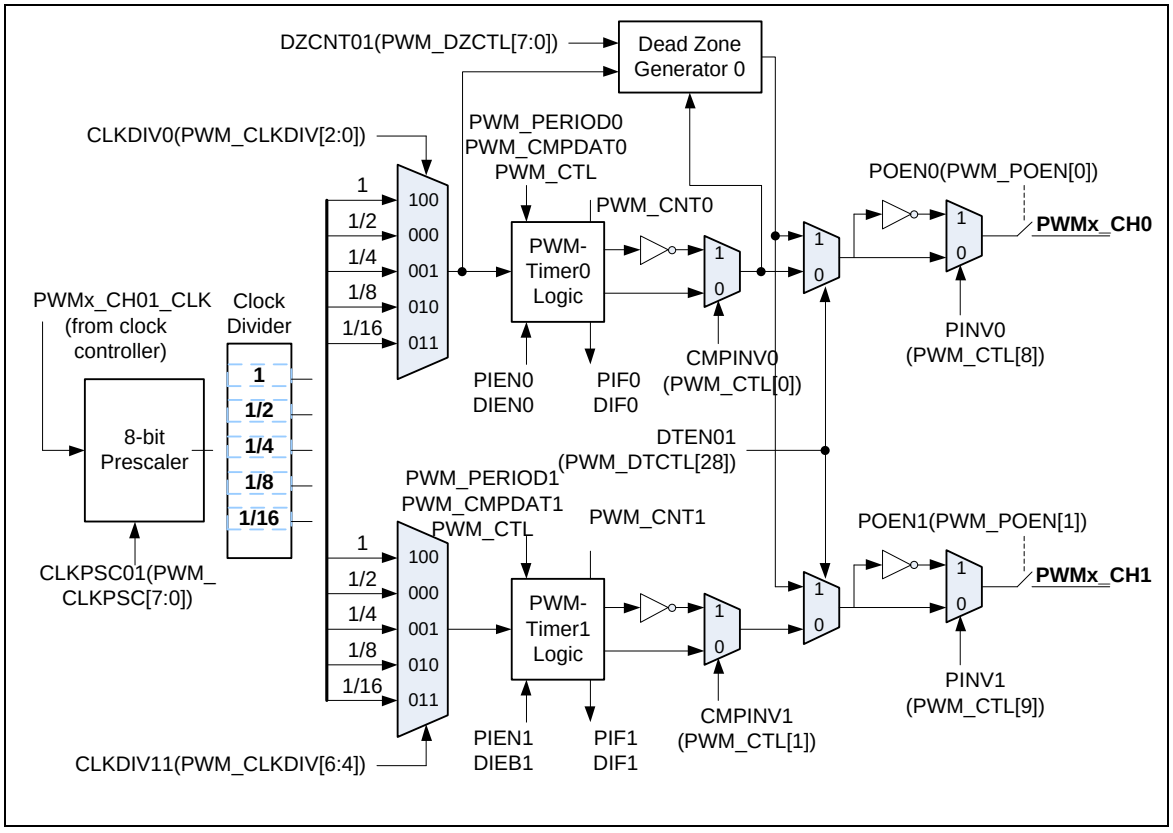


Figure 6.21-2 PWM Generator 0 Architecture Diagram

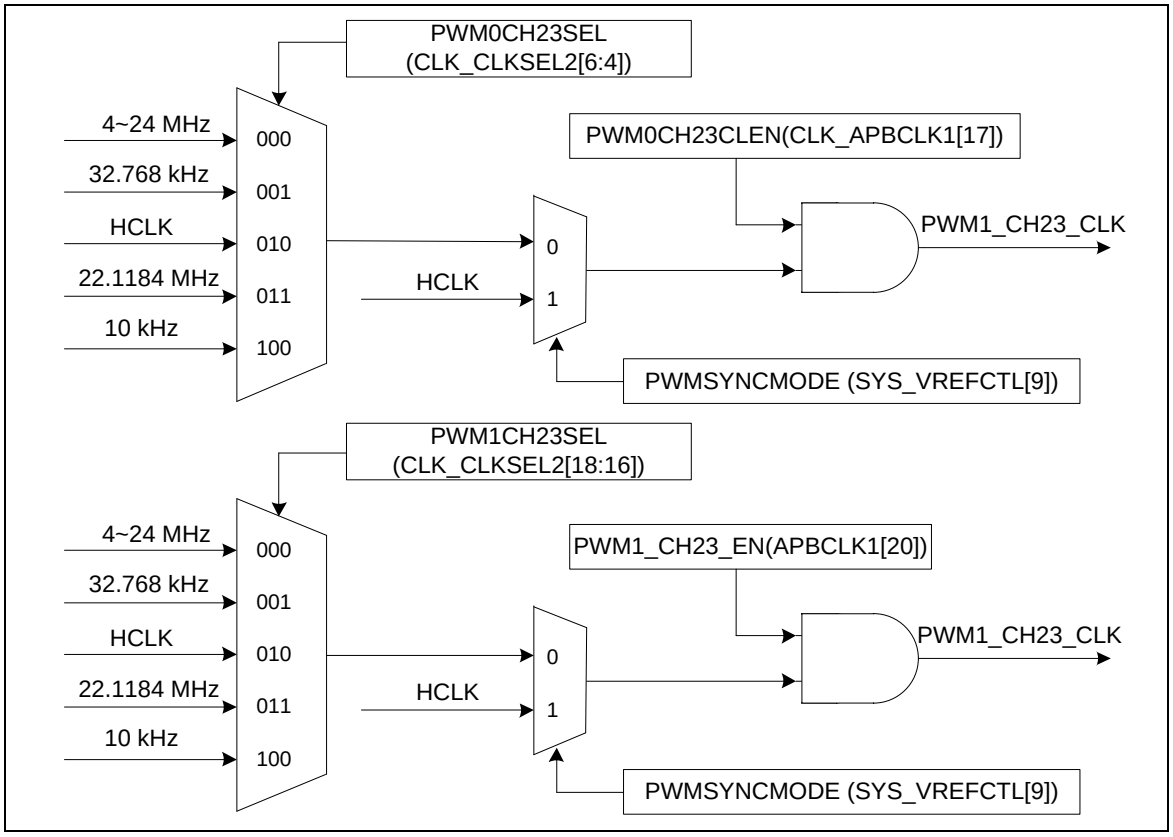


Figure 6.21-3 PWM Generator 2 Clock Source Control

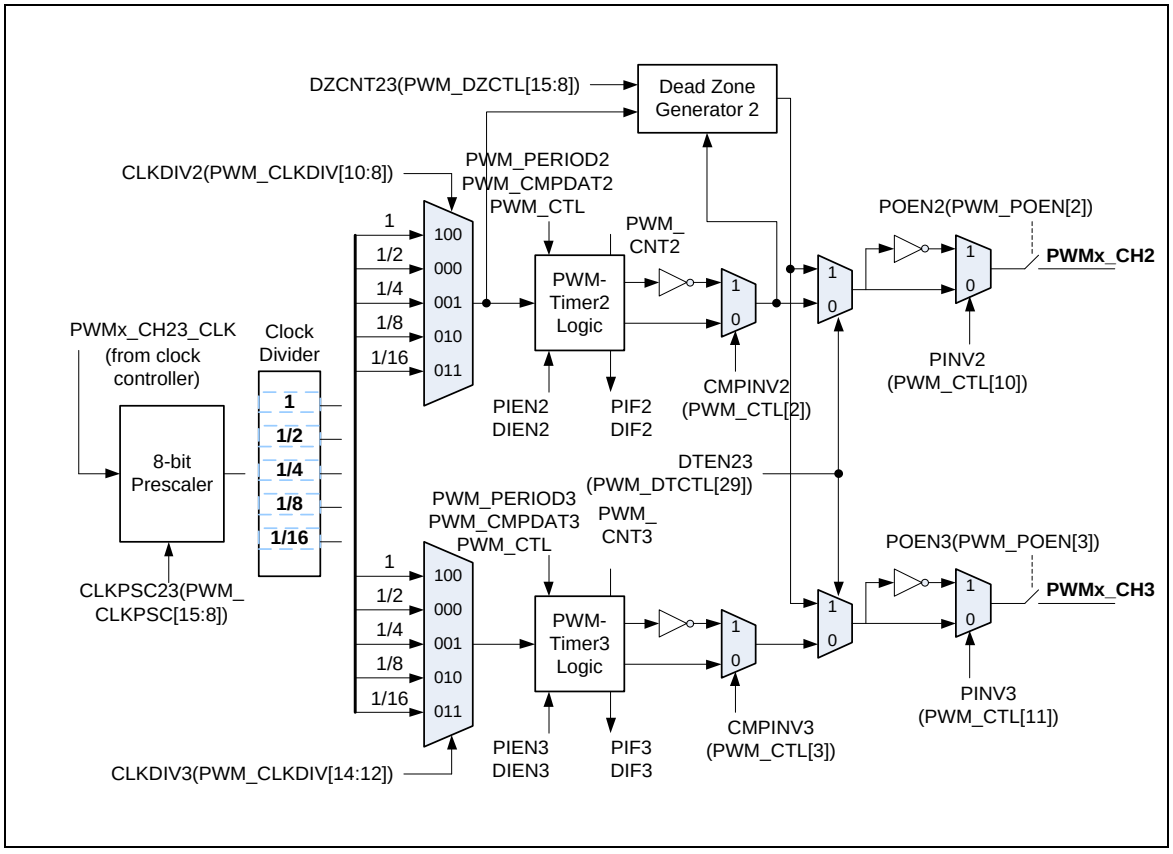


Figure 6.21-4 PWM Generator 2 Architecture Diagram

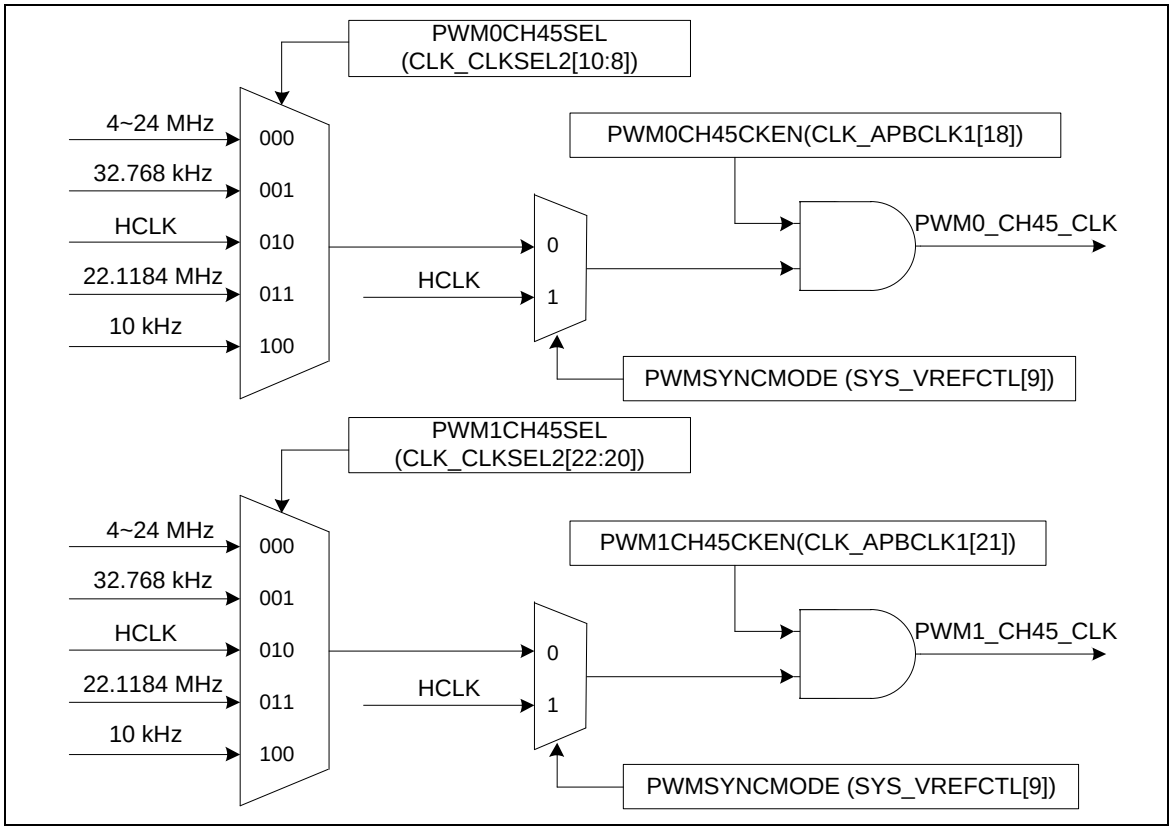


Figure 6.21-5 PWM Generator 4 Clock Source Control

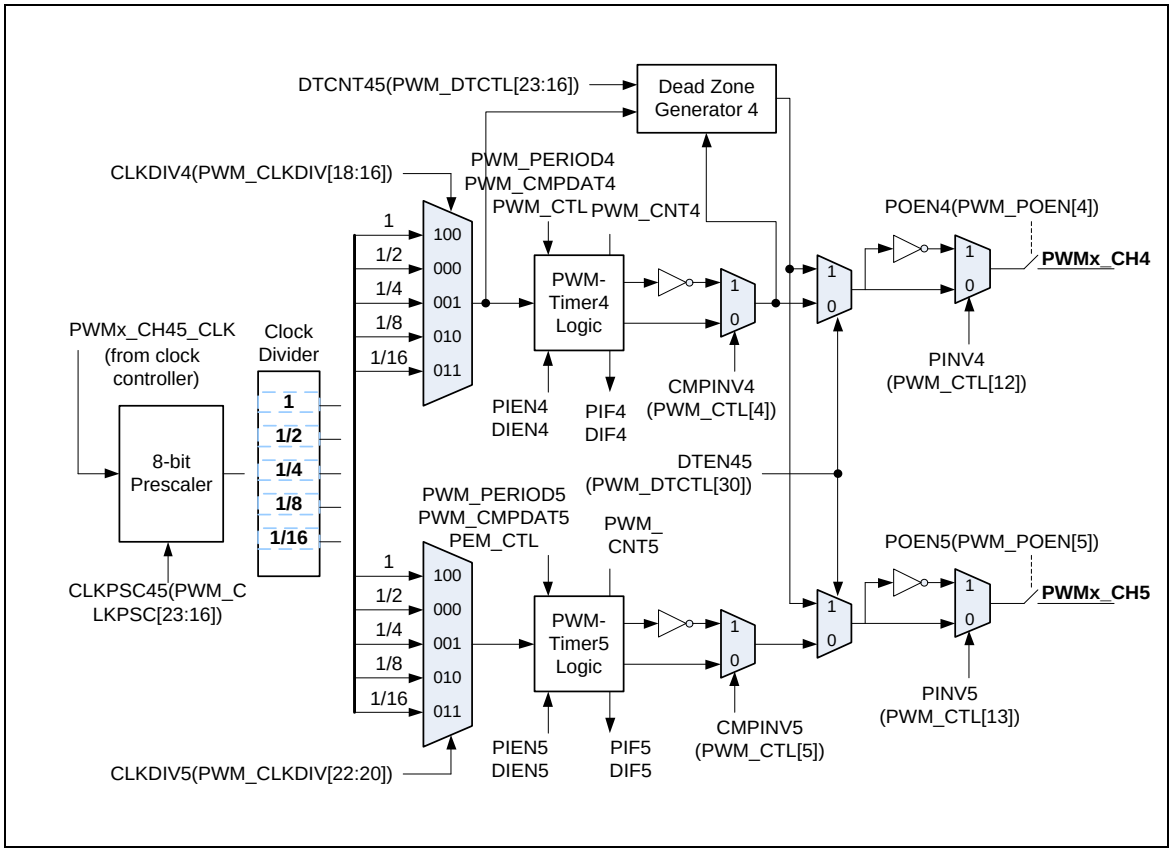


Figure 6.21-6 PWM Generator 4 Architecture Diagram

6.21.4 Functional Description

6.21.4.1 PWM Counter Type

PWM counter supports 2 types operation: Edge-aligned type and Center-aligned type.

6.21.4.2 Edge-aligned type (down-counter)

In Edge-aligned type, the 16 bits PWM counter is a down counter and starts down-counting from PWM_PERIODn to zero to finish a PWM period. The value of PWM counter will be compared with PWM_CMPDATn at the PWM comparator unit. The PWM comparator unit will output low when the value of PWM counter is larger than PWM_CMPDATn and output high when the value of PWM counter is equal or smaller than PWM_CMPDATn. Base on this operating mechanism, the period and duty of PWM can be controlled by PWM_PERIODn and PWM_CMPDATn. The PWM follows the formula below and the legend of PWM waveform of Edge-aligned type is shown in the following figure.

- PWM clock frequency = PWMx_CHy_CLK/[(prescale+1)*(clock divider)]; where x could be 0 or 1 and y could be 0, 2 or 4 depends on selected PWM channel.
- Duty ratio = (PWM_CMPDATn + 1)/(PWM_PERIODn + 1)
- PWM_CMPDATn >= PWM_PERIODn: PWM output is always high
- PWM_CMPDATn < PWM_PERIODn: PWM low width= (PWM_PERIODn - PWM_CMPDATn) unit[1]; PWM high width = (PWM_CMPDATn + 1) unit

- PWM_CMPDATn = 0: PWM low width = (PWM_PERIODn) unit; PWM high width = 1 unit

Note: [1] Unit = one PWM clock cycle.

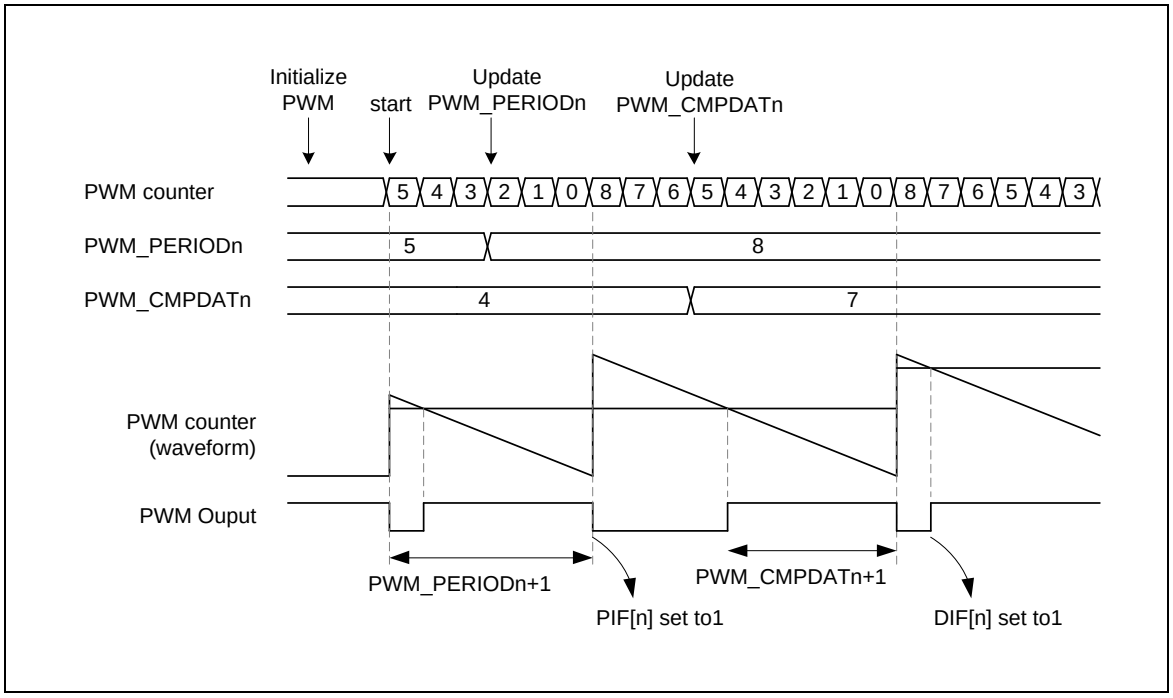


Figure 6.21-7 PWM waveform of Edge-aligned type

6.21.4.3 Center-aligned PWM (Up/down-counter)

In Center-aligned type, the 16 bits PWM counter is an up/down counter and start counting-up from 0 to the value of (PWM_PERIODn+1) and then start counting down to zero to finish a PWM period. The value of PWM counter will be compared with PWM_CMPDATn at the PWM comparator unit. The PWM comparator unit will output low when the value of PWM counter is larger than PWM_CMPDATn and output high when the value of PWM counter is equal or smaller than PWM_CMPDATn. Base on this operating mechanism, the period and duty of PWM can be controlled by PWM_PERIODn and PWM_CMPDATn. The PWM follows the formula below and the legend of PWM waveform of Edge-aligned type is shown in the following figure

- PWM clock frequency = PWMx_CHy_CLK/[(prescale+1)*(clock divider)]; where x could be 0 or 1 and y could be 0, 2 or 4 depends on selected PWM channel.
- Duty ratio = [(2 x PWM_CMPDATn) + 1]/[2 x (PWM_PERIODn + 1)]
- PWM_CMPDATn > PWM_PERIODn: PWM output is always high
- PWM_CMPDATn <= PWM_PERIODn: PWM low width= 2 x (PWM_PERIODn - PWM_CMPDATn) + 1 unit[1]; PWM high width = (2 x PWM_CMPDATn) + 1 unit
- PWM_CMPDATn = 0: PWM low width = (2 x PWM_PERIODn) + 1 unit; PWM high width = 1 unit

Note: [1] Unit = one PWM clock period.

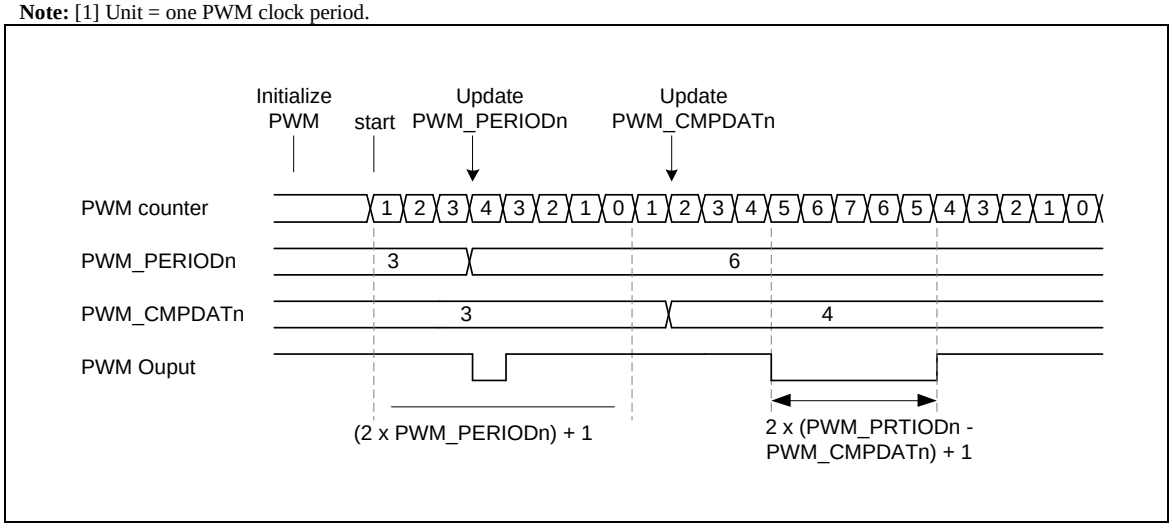


Figure 6.21-8 Center-Aligned Mode Output Waveform

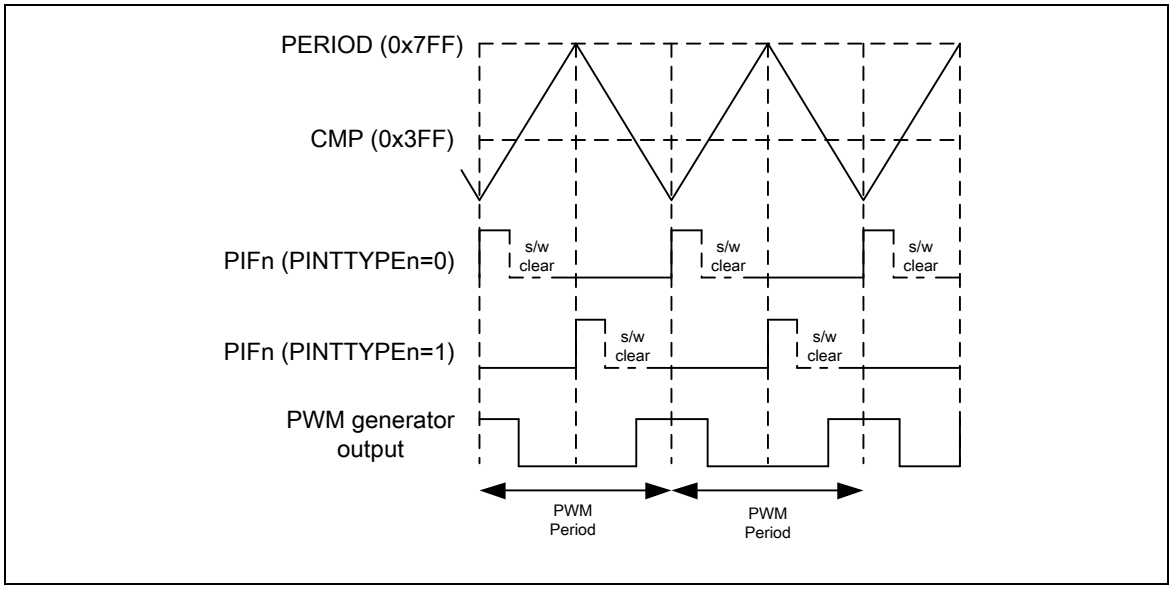


Figure 6.21-9 PWM Center Aligned Interrupt Generate Timing Waveform

6.21.4.4 PWM Counter Operation mode

The PWM counter supports two operation modes: One-shot mode and Auto-reload mode. PWM counter will operate in One-shot mode if CNTMODE (PWM_CTL[16+n]) bit is set to 0, and operate in Auto-reload mode if CNTMODE (PWM_CTL[16+n]) bit is set to 1. It is recommend that configure PWM counter operation mode before setting PWM_PERIODn, PWM_CMPDATn and enable PWM counter running (set CHnEN bit as 1) because the content of PWM_PERIODn and PWM_CMPDATn will be cleared to zero when PWM counter operation mode is changed.

In One-shot mode, PWM_CMPDATn and PWM_PERIODn should be written first and then set CHnEN bit to 1 to enable PWM counter start running. After PWM counter down count from PWM_PERIODn to zero at Edge-aligned type or up count from zero to PWM_PERIODn+1 and then down count to zero at Center-aligned type, PWM_PERIODn and PWM_CMPDATn will be cleared to zero by hardware and PWM counter will be held. Software need to write new PWM_CMPDATn and PWM_PERIODn value to set next one-shot period and duty. When re-start next one-shot operation, the PWM_CMPDATn should be written first because PWM counter will auto re-start counting when PWM_PERIODn is written a non-zero value.

In Auto-reload mode, PWM_CMPDATn and PWM_PERIODn should be written first and then set CHnEN bit to 1 to enable PWM counter start running. The value of PWM_PERIODn will reload to PWM counter when it down count reaches zero. If PWM_PERIODn is set to zero, PWM counter will be held.

6.21.4.5 PWM Double Buffering

The PWM has double buffering function for PWM period and duty. When software changes PWM period (by writing PWM_PERIODn) or PWM duty (by writing PWM_CMPDATn), the period and duty of PWM will reload new value at the start of next period without affecting current timer operation. The double buffing timing waveform shows below:

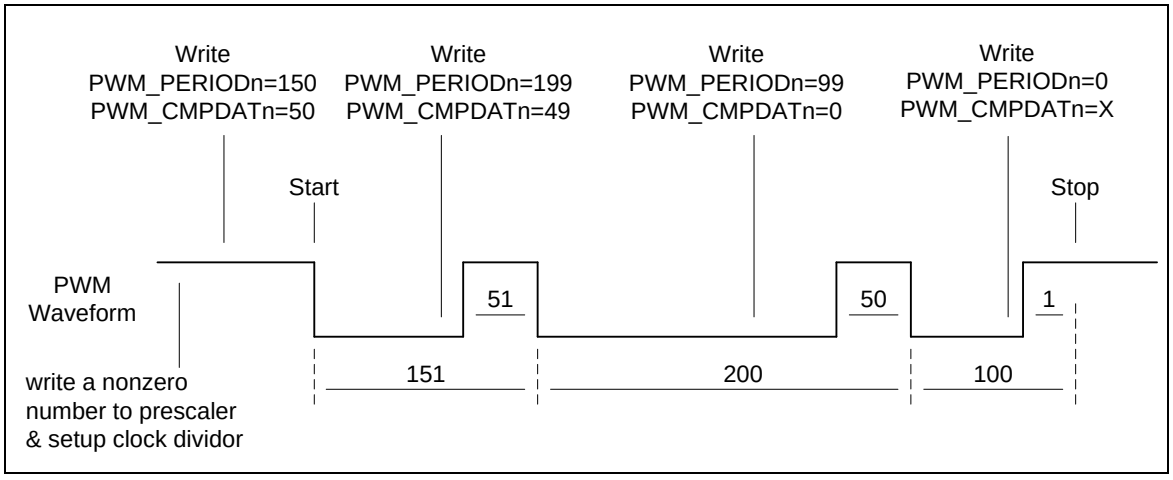


Figure 6.21-10 PWM Double Buffering Timing Waveform

6.21.4.6 PWM Output Control Unit

The PWM output control unit include Dead-Zone Generator, output mode control, mask control, brake control, polarity control.

6.21.4.6.1 Dead-Zone Generator

The Dead-zone generator inserts an “off” period called “dead zone” between the turnings off of one pin to the turning on of the complementary pin of the paired pins. This is to prevent damage to the power switching devices that connect to the PWM output pins. The complementary output pair mode has an 8-bit down counter used to produce the dead zone insertion. The complementary outputs are delayed until the timer counts down to zero.

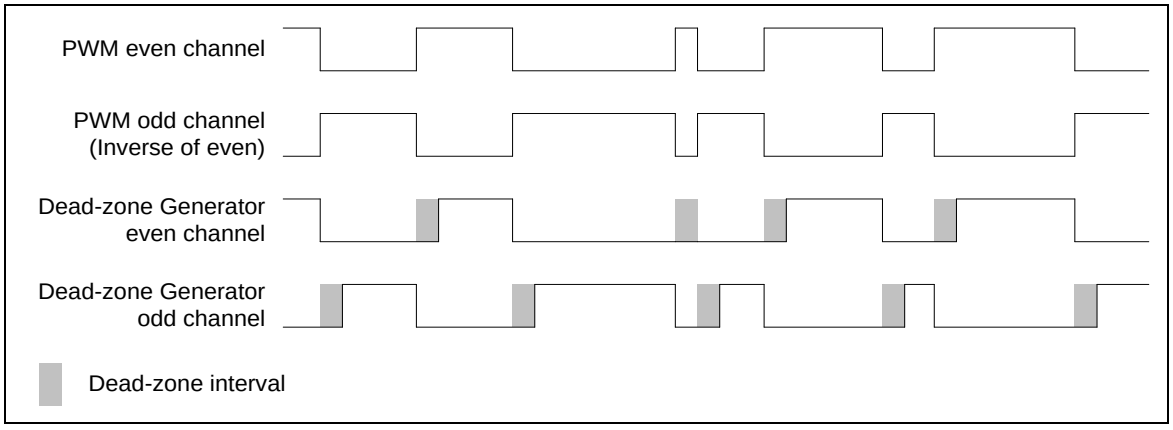


Figure 6.21-11 PWM Paired-output with Dead-zone Generation Operation

6.21.4.6.2 Output Mode Control

The PWM supports four output modes: Independent mode which may be applied to DC motor system, Complementary mode with dead-zone insertion which may be used in the application of AC induction motor and permanent magnet synchronous motor, Synchronous mode that makes both pins of each pair are in phase. Besides, the Group mode, forces the PWMx_CH0,

PWMx_CH2 and PWMx_CH4 synchronous with PWMx_CH0 generator, may simplify updating duty control in DC and BLDC motor applications.

Independent mode

Independent mode is enabled when OUTMODE (PWM_CTL[6]) = 0.

By default, the PWM is operating in independent mode, with six PWM channels outputs: PWMx_CH0, PWMx_CH1, PWMx_CH2, PWMx_CH3, PWMx_CH4 and PWMx_CH5. Each channel is running off its own period and duty.

Complementary mode

Complementary mode is enabled when OUTMODE (PWM_CTL[6]) = 1.

In this module there are three duty-cycle generators utilized for complementary mode, with total of three PWM output pair pins in this module. The total six PWM outputs are grouped into output pairs of even and odd numbered outputs. In complimentary modes, the internal odd PWM signal PGx must always be the complement of the corresponding even PWM signal. For example, PG1 will be the complement of PG0. PG3 will be the complement of PG2 and PG5 will be the complement of PG4. The time base for the PWM module is provided by its own 16-bit timer.

Group mode

Group mode is enabled when GROUPEN (PWM_CTL[7]) = 1.

This control allows all even PWM channels output to be duty controllable by PWM0 duty register.

Both (PWM_MUX2, PWM_MUX3) and (PWM_MUX4, PWM_MUX5) pairs will follow (PWM_MUX0, PWM_MUX1), which imply;

PWM_MUX4 = PWM_MUX2 = PWM_MUX0 and PWM_MUX5 = PWM_MUX3 = PWM_MUX1; And

PWM_MUX5 = PWM_MUX3 = PWM_MUX1= invert (PWM_MUX0) if Complementary mode is enabled OUTMODE (PWM_CTL[6]) = 1

Synchronous mode

Synchronous mode is enabled when SYNCEN (PWM_CTL[15]) = 1.

In the synchronization mode the PWM pair signals from PWM Generator are in-phase.

It means

PWM_MUX1=PWM_MUX0, PWM_MUX3=PWM_MUX2 and PWM_MUX5=PWM_MUX4.

For application, please do not use Group and Synchronous mode simultaneously because the Synchronous mode will be inactive.

6.21.4.6.3 PWM Mask Control

Each of the PWM channel output value can be manually overridden with the settings in the PWM Mask Enable Control Register (PWM_MSKEN) and the PWM Masked Data Register (PWM_MSK). With these settings, the PWM channel outputs can be assigned to specified logic states independent of the duty cycle comparison units. The PWM mask bits are useful when controlling various types of Electrically Commutated Motor (ECM) like a BLDC motor. The

PWM_MSKEN register contains six bits, MSKENn(PWM_MSKEN[5:0]). If the MASKENn is set to active-high, the PWM channel n output will be overridden. The PWM_MSK register contains six bits, MSKDATn(PWM_MSK[5:0]). The bit value of the MSKDATn determines the state value of the PWM channel n output when the channel is overridden. Figure 6.21-12 Illustration of Mask Control Waveformshows an example of how PWM mask control can be used for the override feature.

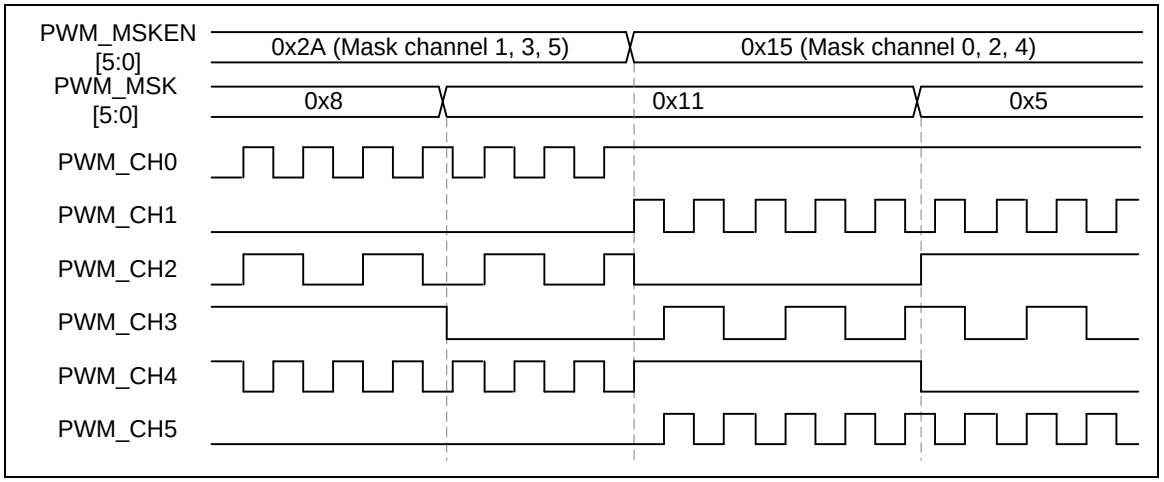


Figure 6.21-12 Illustration of Mask Control Waveform

6.21.4.6.4 Brake Control

This device supports two brake detectors, BK0 and BK1, and each of them has 4 brake signals, one external brake pin (BKPx0 for BK0 and BKPx1 for BK1), and three analog comparator outputs. Both external brake pins have each 4-degree digital filter that is user controllable through BKxFILT.1-0 bits (x=0 and 1 for BK0 and BK1). The Brake function is controlled by the contents of the SFR PWMCON register.

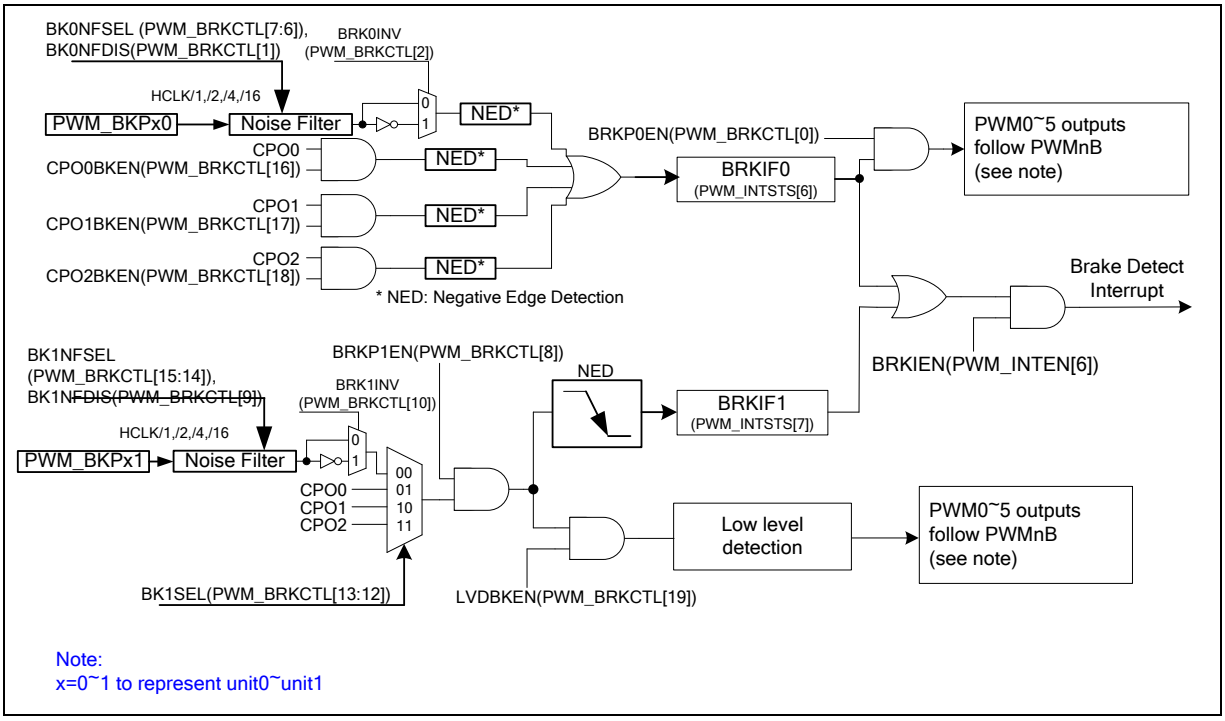


Figure 6.21-13 PWM Brake Function

6.21.4.6.5 Polarity Control

Each PWM port, from PWMx_CH0 to PWMx_CH5, has an independent polarity control to configure the polarity of the active state of the PWM output. By default, the PWM output is active high. This implies the PWM OFF state is low and ON state is high. This definition is variable through setting the PWM Output Polarity Inverse Enable PINV (PWM_CTL[13:8]), for each individual PWM channel.

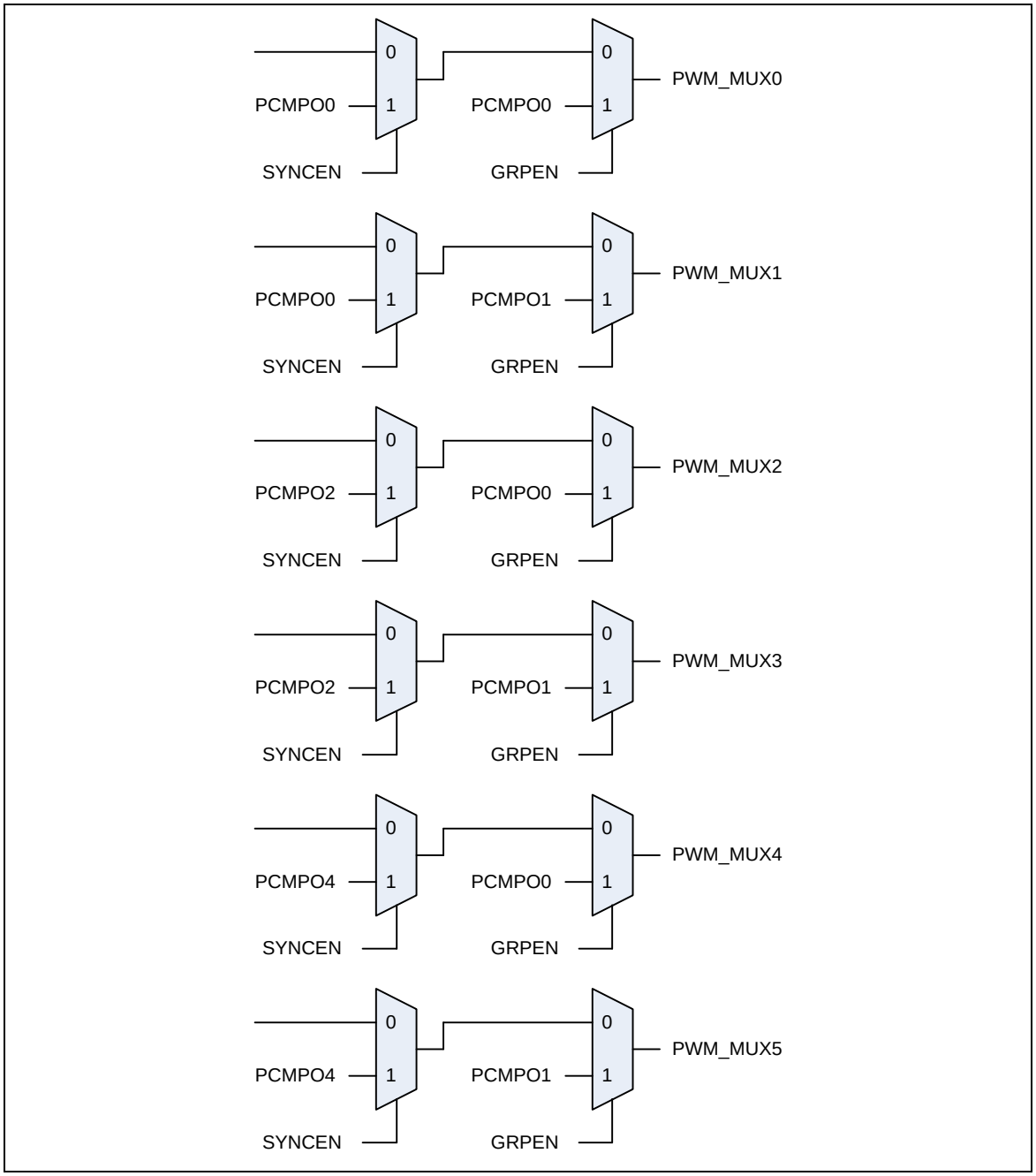


Figure 6.21-14 PWM Output Multiplex for Group Mode and Synchronous Mode

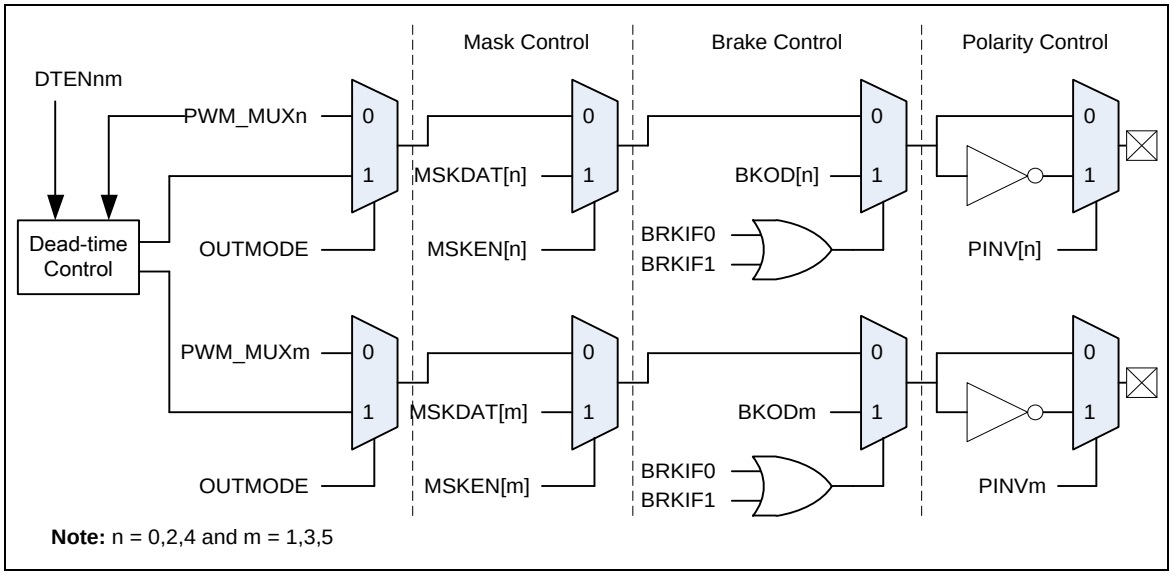


Figure 6.21-15 PWM Multiplex for Mask Control, Brake Control and Polarity Control

6.21.4.7 Capture Operation

The channel of capture input and PWM output share one pin and PWM counter. The capture function always latches PWM counter to RCAPDATn when input channel has a rising transition and latches PWM counter to FCAPDATn when input channel has a falling transition. Capture interrupt is programmable by setting CRLIE[n] (Rising latch Interrupt enable) and CFLIE[n] (Falling latch Interrupt enable) to decide the condition of interrupt occur. Whenever the capture controller issues a capture interrupt, the corresponding PWM counter will be reloaded with PWM_PERIODn at this moment. Note that the corresponding GPIO pins must be configured as capture function (enable CAPINEN[n]) for the corresponding capture channel.

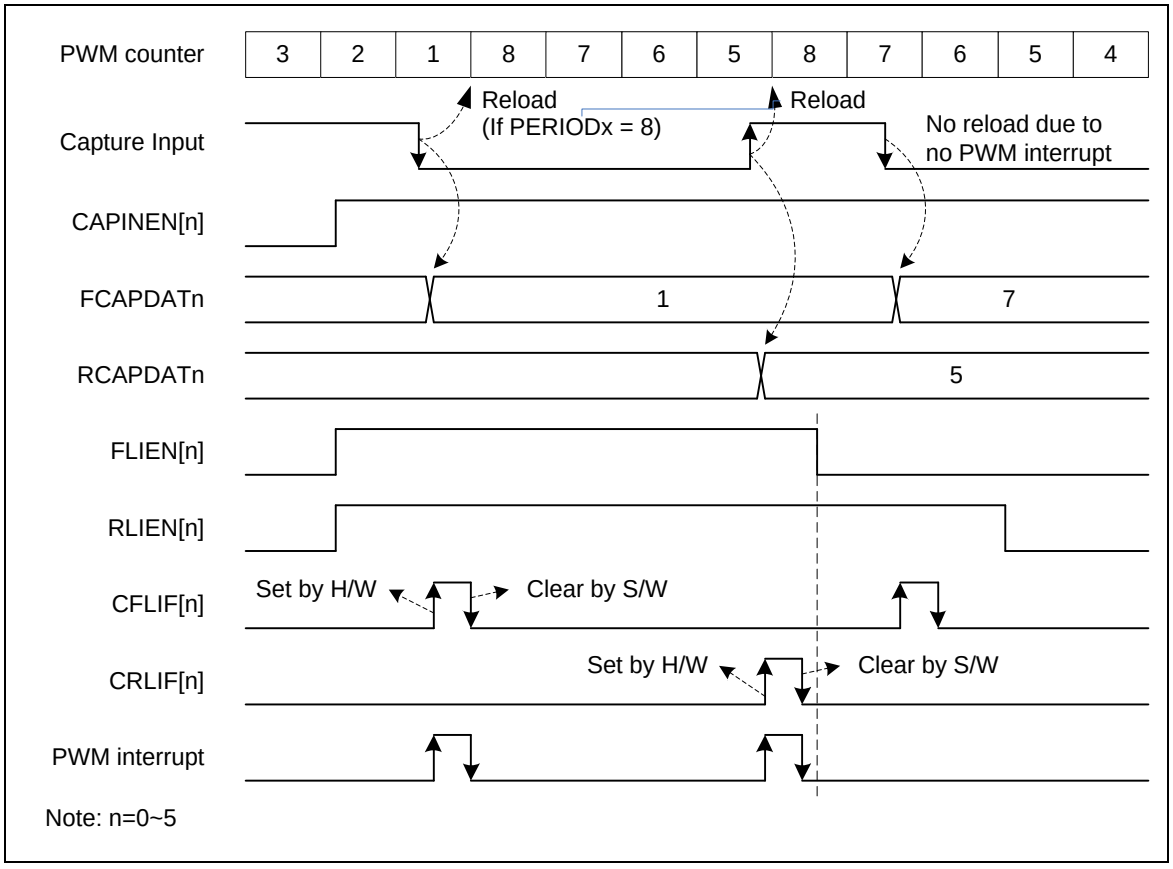


Figure 6.21-16 Capture Operation Timing

In this case, the PERIOD is 8:

1. The PWM counter will be reloaded with PWM_PERIODn when a capture interrupt flag (CAPIFx) is set.
2. The channel low pulse width is (PWM_PERIODn + 1 - PWM_RCAPDATn)
3. The channel high pulse width is (PWM_PERIODn + 1 - PWM_FCAPDATn)

6.21.4.8 PWM-Timer Interrupt Architecture

The NUC442/NUC472 supports 2 independent interrupt vectors for PWM, one is for PWM0 and the other is for PWM1. Each PWM interrupt (PWM_INT) can source from PWM0_INT, PWM1_INT, PWM2_INT, PWM3_INT, PWM4_INT, PWM5_INT and BRAKE_INT. BRAKE_INT is set to 1 when BRKIF0 or BRKIF1 is triggered if brake interrupt enable bit (BRKIEN) is set 1.PWMn_INT is set to 1 when four condition is match: PWMIFn is triggered if PWM interrupt enable bit (PWMIEn) is set to 1, PWMDIFn is triggered if PWM duty interrupt enable bit (PWMDIEn) is set to 1, CRLIFn is triggered if rising capture interrupt enable bit (CRLIEn) is set to 1, and CFLIFn is triggered if falling capture interrupt enable bit (CFLIEn) is set to 1. The following figure demonstrates the architecture of PWM interrupts.

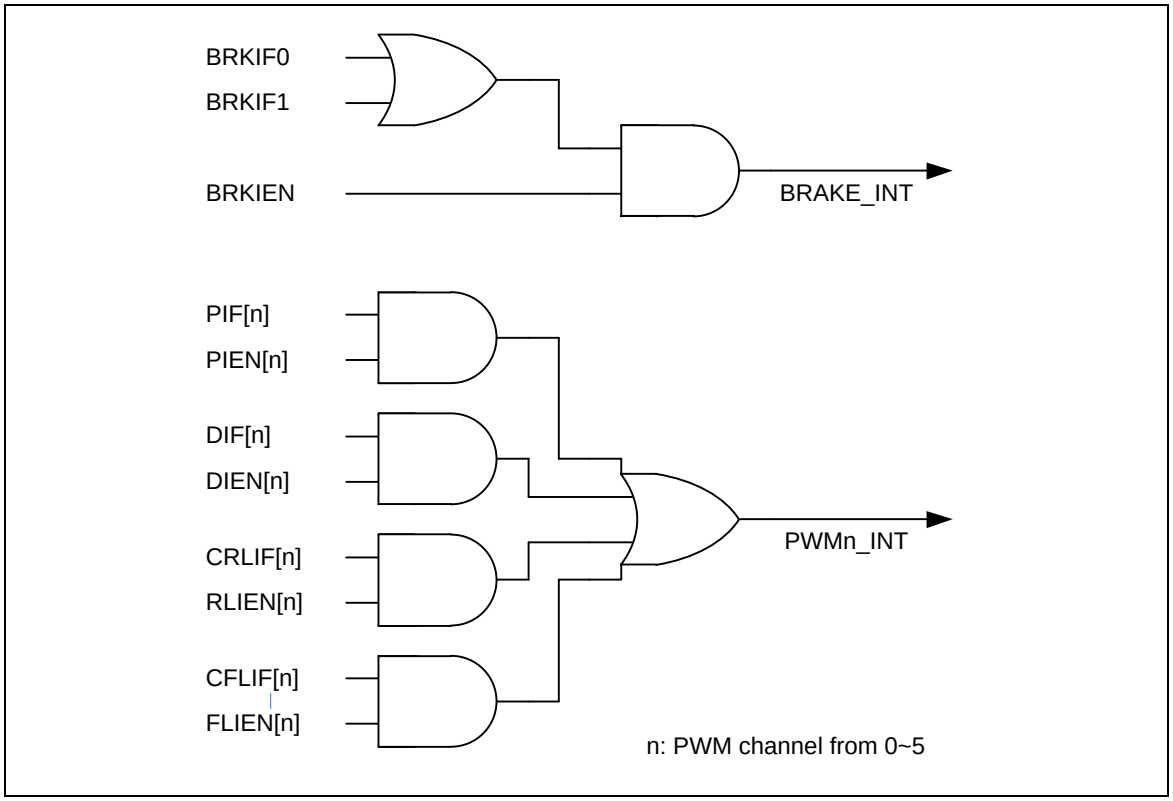


Figure 6.21-17 PWM Interrupt Architecture Diagram

6.21.4.9 PWM Start Procedure

The following procedure is recommended for starting a PWM drive.

1. Set clock source divider select register (PWM_CSDR)
2. Set prescaler (PWM_CLKPSC)
3. Set PWM counter auto-reload/one-shot operation mode, PWM comparator output inverter on/off, PWM output polarity, PWM counter Edge-aligned/Center-aligned type, PWM interrupt type (PWM_CTL)
4. Set Dead-zone generator on/off (PWM_DTCTL)
5. Set comparator register (PWM_CMPDATn) for setting PWM duty.
6. Set PWM counter register (PWM_PERIODn) for setting PWM period.
7. Set interrupt Enable Control Register (PWM_INTEN) (option)
8. Set corresponding GPIO pins as PWM function for the corresponding PWM channel.
9. Enable PWM timer start running (Set CNTEN (PWM_CNTEN[n]) = 1)

6.21.4.10 PWM Re-Start Procedure in One-shot mode

After PWM waveform is generated once in PWM One-shot mode, PWM counter will be stopped automatically. The following procedure is recommended for re-starting PWM at One-shot mode.

1. Set comparator register (PWM_CMPDATn) for setting PWM duty.

2. Set PWM counter register (PWM_PERIODn) for setting PWM period. After setup PWM_PERIODn, PWM wave will be generated.

6.21.4.11 PWM Stop Procedure

Method 1:

Set PWM counter register (PWM_PERIODn) as 0, and monitor PWM_CNTn (current value of 16-bit down-counter). When PWM_CNTn reaches to 0, disable PWM-Timer CNTEN (PWM_CNTEN[n]). ***(Recommended)***

Method 2:

Set 16-bit down counter (PWM_PERIODn) as 0. When interrupt request happened, disable PWM-Timer CNTEN (PWM_CNTEN[n]). ***(Recommended)***

Method 3:

Disable PWM-Timer directly (CNTEN (PWM_CNTEN[n]). ***(Not recommended)***

The reason why method 3 is not recommended is that disable (CNTEN (PWM_CNTEN[n]) will immediately stop PWM output signal and lead to change the duty of the PWM output, this may cause damage to the control circuit of motor

6.21.4.12 Capture Start Procedure

9. Set clock source divider select register (PWM_CLKDIV)
10. Set prescaler (PWM_CLKPSC)
11. Set capture input inverter on/off and capture function enable (PWM_CAPCTL)
12. Set PWM counter as Auto-reload mode, Edge-aligned type (PWM_CTL)
13. Set PWM period register (PWM_PERIODn)
14. Enable capture input path (PWM_CAPCTL)
15. Enable PWM timer start running (Set (CNTEN (PWM_CNTEN[n]) = 1)

6.21.5 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
PWM Base Address: **PWMx_BA = 0x4005_8000 + 0x1000 * x** **x=0,1**				
PWM_CLKPSC	PWMx_BA+0x00	R/W	PWM Clock Prescale Register	0x0000_0000
PWM_CLKDIV	PWMx_BA+0x04	R/W	PWM Clock Divide Register	0x0000_0000
PWM_CTL	PWMx_BA+0x08	R/W	PWM Control Register	0x0000_0000
PWM_CNTEN	PWMx_BA+0x0C	R/W	PWM Counter Enable Control Register	0x0000_0000
PWM_PERIOD 0	PWMx_BA+0x10	R/W	PWM Period Register 0	0x0000_0000
PWM_PERIOD 1	PWMx_BA+0x14	R/W	PWM Period Register 1	0x0000_0000
PWM_PERIOD 2	PWMx_BA+0x18	R/W	PWM Period Register 2	0x0000_0000
PWM_PERIOD 3	PWMx_BA+0x1C	R/W	PWM Period Register 3	0x0000_0000
PWM_PERIOD 4	PWMx_BA+0x20	R/W	PWM Period Register 4	0x0000_0000
PWM_PERIOD 5	PWMx_BA+0x24	R/W	PWM Period Register 5	0x0000_0000
PWM_CMPDAT 0	PWMx_BA+0x28	R/W	PWM Comparator Register 0	0x0000_0000
PWM_CMPDAT 1	PWMx_BA+0x2C	R/W	PWM Comparator Register 1	0x0000_0000
PWM_CMPDAT 2	PWMx_BA+0x30	R/W	PWM Comparator Register 2	0x0000_0000
PWM_CMPDAT 3	PWMx_BA+0x34	R/W	PWM Comparator Register 3	0x0000_0000
PWM_CMPDAT 4	PWMx_BA+0x38	R/W	PWM Comparator Register 4	0x0000_0000
PWM_CMPDAT 5	PWMx_BA+0x3C	R/W	PWM Comparator Register 5	0x0000_0000
PWM_CNT0	PWMx_BA+0x40	R	PWM Data Register 0	0x0000_0000
PWM_CNT1	PWMx_BA+0x44	R	PWM Data Register 1	0x0000_0000
PWM_CNT2	PWMx_BA+0x48	R	PWM Data Register 2	0x0000_0000
PWM_CNT3	PWMx_BA+0x4C	R	PWM Data Register 3	0x0000_0000
PWM_CNT4	PWMx_BA+0x50	R	PWM Data Register 4	0x0000_0000

PWM_CNT5	PWMx_BA+0x54	R	PWM Data Register 5	0x0000_0000
PWM_MSKEN	PWMx_BA+0x58	R/W	PWM Mask Control Register	0x0000_0000
PWM_MSK	PWMx_BA+0x5C	R/W	PWM Mask Data Register	0x0000_0000
PWM_DTCTL	PWMx_BA+0x60	R/W	PWM Dead-zone Control Register	0x0000_0000
PWM_TRGADCTL	PWMx_BA+0x64	R/W	PWM Trigger Control Register	0x0000_0000
PWM_TRGADCSTS	PWMx_BA+0x68	R/W	PWM Trigger ADC Status Register	0x0000_0000
PWM_BRKCTL	PWMx_BA+0x6C	R/W	PWM Brake Control Register	0x0000_0000
PWM_INTCTL	PWMx_BA+0x70	R/W	PWM Interrupt Control Register	0x0000_0000
PWM_INTEN	PWMx_BA+0x74	R/W	PWM Interrupt Enable Control Register	0x0000_0000
PWM_INTSTS	PWMx_BA+0x78	R/W	PWM Interrupt Flag Register	0x0000_0000
PWM_POEN	PWMx_BA+0x7C	R/W	PWM Output Enable Control Register	0x0000_0000
PWM_CAPCTL	PWMx_BA+0x80	R/W	PWM Capture Control Register	0x0000_0000
PWM_CAPINEN	PWMx_BA+0x84	R/W	PWM Capture Input Enable Control Register	0x0000_0000
PWM_CAPSTS	PWMx_BA+0x88	R	PWM Capture Status Register	0x0000_0000
PWM_RCAPDAT0	PWMx_BA+0x90	R	PWM Capture Rising Latch Register 0	0x0000_0000
PWM_FCAPDAT0	PWMx_BA+0x94	R	PWM Capture Falling Latch Register 0	0x0000_0000
PWM_RCAPDAT1	PWMx_BA+0x98	R	PWM Capture Rising Latch Register 1	0x0000_0000
PWM_FCAPDAT1	PWMx_BA+0x9C	R	PWM Capture Falling Latch Register 1	0x0000_0000
PWM_RCAPDAT2	PWMx_BA+0xA0	R	PWM Capture Rising Latch Register 2	0x0000_0000
PWM_FCAPDAT2	PWMx_BA+0xA4	R	PWM Capture Falling Latch Register 2	0x0000_0000
PWM_RCAPDAT3	PWMx_BA+0xA8	R	PWM Capture Rising Latch Register 3	0x0000_0000
PWM_FCAPDAT3	PWMx_BA+0xAC	R	PWM Capture Falling Latch Register 3	0x0000_0000
PWM_RCAPDAT4	PWMx_BA+0xB0	R	PWM Capture Rising Latch Register 4	0x0000_0000
PWM_FCAPDAT4	PWMx_BA+0xB4	R	PWM Capture Falling Latch Register 4	0x0000_0000
PWM_RCAPDAT5	PWMx_BA+0xB8	R	PWM Capture Rising Latch Register 5	0x0000_0000
PWM_FCAPDA	PWMx_BA+0xBC	R	PWM Capture Falling Latch Register 5	0x0000_0000

T5				
PWM_SBS0	PWMx_BA+0xE0	R	PWM0 Synchronous Busy Status Register	0x0000_0000
PWM_SBS1	PWMx_BA+0xE4	R	PWM1 Synchronous Busy Status Register	0x0000_0000
PWM_SBS2	PWMx_BA+0xE8	R	PWM2 Synchronous Busy Status Register	0x0000_0000
PWM_SBS3	PWMx_BA+0xEC	R	PWM3 Synchronous Busy Status Register	0x0000_0000
PWM_SBS4	PWMx_BA+0xF0	R	PWM4 Synchronous Busy Status Register	0x0000_0000
PWM_SBS5	PWMx_BA+0xF4	R	PWM5 Synchronous Busy Status Register	0x0000_0000

6.21.6 Register Description

PWM Clock Prescale Register (PWM_CLKPSC)

Register	Offset	R/W	Description	Reset Value
PWM_CLKPSC	PWMx_BA+0x00	R/W	PWM Clock Prescale Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CLKPSC45							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CLKPSC23							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CLKPSC01							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**CLKPSC45**	**PWM Counter Base-Clock Prescale For PWM Pair Of Channel 4 And Channel 5** The base-clock of PWM counter is decided by clock pre-scalar and clock divider. Each PWM pair share one PWM counter base-clock prescaler. The base-clock of PWM counter is divided by (CLKPSC45 + 1). If the value of CLKPSC45 is zero, the base-clock prescaler will stop output clock and corresponding PWM counter will also stop.
[15:8]	**CLKPSC23**	**PWM Counter Base-Clock Prescale For PWM Pair Of Channel 2 And Channel 3** The base-clock of PWM counter is decided by clock pre-scalar and clock divider. Each PWM pair share one PWM counter base-clock prescaler. The base-clock of PWM counter is divided by (CLKPSC23 + 1). If the value of CLKPSC23 is zero, the base-clock prescaler will stop output clock and corresponding PWM counter will also stop.
[7:0]	**CLKPSC01**	**PWM Counter Base-Clock Prescale For PWM Pair Of Channel 0 And Channel 1** The base-clock of PWM counter is decided by clock pre-scalar and clock divider. Each PWM pair share one PWM counter base-clock prescaler. The base-clock of PWM counter is divided by (CLKPSC01 + 1). If the value of CLKPSC01 is zero, the base-clock prescaler will stop output clock and corresponding PWM counter will also stop.

PWM Clock Divide Register (PWM_CLKDIV)

Register	Offset	R/W	Description	Reset Value
PWM_CLKDIV	PWMx_BA+0x04	R/W	PWM Clock Divide Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved	CLKDIV5			Reserved	CLKDIV4		
15	14	13	12	11	10	9	8
Reserved	CLKDIV3			Reserved	CLKDIV2		
7	6	5	4	3	2	1	0
Reserved	CLKDIV1			Reserved	CLKDIV0		

Bits	Description	
[31:23]	**Reserved**	Reserved.
[22:20]	**CLKDIV5**	**PWM Counter Base-Clock Divide For PWMx_CH5** The base-clock of PWM counter is decided by clock pre-scalar and clock divider. Each PWM counter has independent clock divider control register and the divided value is listed in the table below: 000 = 2. 001 = 4. 010 = 8. 011 = 16. 100 = 1.
[19]	**Reserved**	Reserved.
[18:16]	**CLKDIV4**	**PWM Counter Base-Clock Divide For PWMx_CH4** (Table is the same as CLKDIV5)
[15]	**Reserved**	Reserved.
[14:12]	**CLKDIV3**	**PWM Counter Base-Clock Divide For PWMx_CH3** (Table is the same as CLKDIV5)
[11]	**Reserved**	Reserved.
[10:8]	**CLKDIV2**	**PWM Counter Base-Clock Divide For PWMx_CH2** (Table is the same as CLKDIV5)
[7]	**Reserved**	Reserved.
[6:4]	**CLKDIV1**	**PWM Counter Base-Clock Divide For PWMx_CH1** (Table is the same as CLKDIV5)
[3]	**Reserved**	Reserved.

[2:0]	**CLKDIV0**	**PWM Counter Base-Clock Divide For PWMx_CH0** (Table is the same as CLKDIV5)

PWM Control Register (PWM_CTL)

Register	Offset	R/W	Description	Reset Value
PWM_CTL	PWMx_BA+0x08	R/W	PWM Control Register	0x0000_0000

31	30	29	28	27	26	25	24
DBGTRIOFF	Reserved	CNTTYPE					
23	22	21	20	19	18	17	16
Reserved		CNTMODE					
15	14	13	12	11	10	9	8
SYNCEN	Reserved	PINV					
7	6	5	4	3	2	1	0
GROUPEN	OUTMODE	CMPINV					

Bits	Description	
[31]	**DBGTRIOFF**	**ICE Debug Mode Acknowledge Disable Bit (Write Protect)** 0 = ICE debug mode acknowledgement effects PWM output. PWM pin will be forced as tri-state while ICE debug mode acknowledged. 1 = ICE debug mode acknowledgement disabled. **Note:** The PWM pin will keep output no matter ICE debug mode acknowledged or not.
[30]	**Reserved**	Reserved.
[29:24]	**CNTTYPE**	**PWM Counter Operation Aligned Type** 0 = PWM counter operating as Edge-aligned type. 1 = PWM counter operating as Center-aligned type. **Note:** Each bit control corresponding PWM channel
[23:22]	**Reserved**	Reserved.
[21:16]	**CNTMODE**	**PWM Counter Operation Mode** 0 = PWM counter working as One-shot mode. 1 = PWM counter working as Auto-reload mode. **Note:** Each bit control corresponding PWM channel **Note:** If there is a transition at this bit, it will cause PWM_PERIODn and PWM_CMPDATn be cleared.
[15]	**SYNCEN**	**Synchronous Mode Enable Bit** 0 = The signals timing of each PWM channel are independent. 1 = Unify the signals timing of PWM_CH0 and PWM_CH1 in the same phase which is controlled by PWM0 and so as another two PWM pair. **Note:** If Group and Synchronous mode are enabled simultaneously, the Synchronous mode will be inactive.
[14]	**Reserved**	Reserved.
[13:8]	**PINV**	**PWM Output Polar Inverse Enable Bit** The register controls polarity state of PWM output

		0 = PWM output polar inverse Disabled. 1 = PWM output polar inverse Enabled. **Note:** Each bit controls the corresponding PWM channel.
[7]	**GROUPEN**	**Group Mode Enable Bit** 0 = The signals timing of each PWM channel are independent. 1 = Unify the signals timing of PWM_CH0, PWM_CH2 and PWM_CH4 in the same phase which is controlled by PWM_CH0 and unify the signals timing of PWM_CH1, PWM_CH3 and PWM_CH5 in the same phase which is controlled by PWM_CH1.
[6]	**OUTMODE**	**PWM Output Mode** The register controls the output mode of PWM 0 = PWM output at independent mode. 1 = PWM output at complementary mode.
[5:0]	**CMPINV**	**PWM Comparator Output Inverter Enable Bit** When CMPINV is set to high, the PWM comparator output signals will be inversed, 0 = Comparator output inverter Disabled. 1 = Comparator output inverter Enabled. **Note:** Each bit control corresponding PWM channel

PWM Counter Enable Control Register (PWM_CNTEN)

Register	Offset	R/W	Description	Reset Value
PWM_CNTEN	PWMx_BA+0x0C	R/W	PWM Counter Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**CNTEN**					

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5:0]	**CNTEN**	**PWM Counter Enable Bit** 0 = PWM Counter Stop Running. 1 = PWM Counter Start Running. **Note:** Each bit controls the corresponding PWM channel.

PWM Period Register 5-0 (PWM_PERIOD5-0)

Register	Offset	R/W	Description	Reset Value
PWM_PERIOD0	PWMx_BA+0x10	R/W	PWM Period Register 0	0x0000_0000
PWM_PERIOD1	PWMx_BA+0x14	R/W	PWM Period Register 1	0x0000_0000
PWM_PERIOD2	PWMx_BA+0x18	R/W	PWM Period Register 2	0x0000_0000
PWM_PERIOD3	PWMx_BA+0x1C	R/W	PWM Period Register 3	0x0000_0000
PWM_PERIOD4	PWMx_BA+0x20	R/W	PWM Period Register 4	0x0000_0000
PWM_PERIOD5	PWMx_BA+0x24	R/W	PWM Period Register 5	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
PERIOD[15:8]							
7	6	5	4	3	2	1	0
PERIOD[7:0]							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**PERIOD**	**PWM Period Register** PERIOD determines the PWM period. PWM frequency = PWMxy_CLK/[(prescale+1)*(clock divider)*(PERIOD+1)]; where xy, could be 01, 23 or 45, depends on selected PWM channel. For Edge-aligned mode: ● Duty ratio = (CMP+1)/(PERIOD+1). ● CMP >= PERIOD: PWM output is always high. ● CMP < PERIOD: PWM low width = (PERIOD-CMP) unit; PWM high width = (CMP+1) unit. ● CMP = 0: PWM low width = (PERIOD) unit; PWM high width = 1 unit. For Center-aligned mode: ● Duty ratio = [(2 x CMP) + 1]/[2 x (PERIOD+1)]. ● CMP > PERIOD: PWM output is always high. ● CMP <= PERIOD: PWM low width = 2 x (PERIOD-CMP) + 1 unit; PWM high width = (2 x CMP) + 1 unit.

		● CMP = 0: PWM low width = 2 x PERIOD + 1 unit; PWM high width = 1 unit. (Unit = one PWM clock cycle). **Note1:** Any write to PERIOD will take effect in next PWM cycle. **Note2:** When PWM operating at center-aligned type, PERIOD value should be set between 0x0000 to 0xFFFE. If PERIOD equal to 0xFFFF, the PWM will work unpredictable. **Note3:** When PERIOD value is set to 0, PWM output is always high.

PWM Comparator Register 5-0 (PWM_CMPDAT5-0)

Register	Offset	R/W	Description	Reset Value
PWM_CMPDAT0	PWMx_BA+0x28	R/W	PWM Comparator Register 0	0x0000_0000
PWM_CMPDAT1	PWMx_BA+0x2C	R/W	PWM Comparator Register 1	0x0000_0000
PWM_CMPDAT2	PWMx_BA+0x30	R/W	PWM Comparator Register 2	0x0000_0000
PWM_CMPDAT3	PWMx_BA+0x34	R/W	PWM Comparator Register 3	0x0000_0000
PWM_CMPDAT4	PWMx_BA+0x38	R/W	PWM Comparator Register 4	0x0000_0000
PWM_CMPDAT5	PWMx_BA+0x3C	R/W	PWM Comparator Register 5	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CMP[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CMP[7:0]							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**CMP**	**PWM Compare Register** CMP determines the PWM duty. PWM frequency = PWMxy_CLK/[(prescale+1)*(clock divider)*(PERIOD+1)]; where xy, could be 01, 23 or, 45, depends on selected PWM channel. For Edge-aligned mode: ● Duty ratio = (CMP+1)/(PERIOD+1). ● CMP >= PERIOD: PWM output is always high. ● CMP < PERIOD: PWM low width = (PERIOD-CMP) unit; PWM high width = (CMP+1) unit. ● CMP = 0: PWM low width = (PERIOD) unit; PWM high width = 1 unit. For Center-aligned mode: ● Duty ratio = [(2 x CMP) + 1]/[2 x (PERIOD+1)]. ● CMP > PERIOD: PWM output is always high. ● CMP <= PERIOD: PWM low width = 2 x (PERIOD-CMP) + 1 unit; PWM high width = (2 x CMP) + 1 unit. ● CMP = 0: PWM low width = 2 x PERIOD + 1 unit; PWM high width = 1 unit.

		(Unit = one PWM clock cycle). **Note:** Any write to CMP will take effect in next PWM cycle.

PWM Data Register 5-0 (PWM_CNT5-0)

Register	Offset	R/W	Description	Reset Value
PWM_CNT0	PWMx_BA+0x40	R	PWM Data Register 0	0x0000_0000
PWM_CNT1	PWMx_BA+0x44	R	PWM Data Register 1	0x0000_0000
PWM_CNT2	PWMx_BA+0x48	R	PWM Data Register 2	0x0000_0000
PWM_CNT3	PWMx_BA+0x4C	R	PWM Data Register 3	0x0000_0000
PWM_CNT4	PWMx_BA+0x50	R	PWM Data Register 4	0x0000_0000
PWM_CNT5	PWMx_BA+0x54	R	PWM Data Register 5	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CNT[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNT[7:0]							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**CNT**	**PWM Data Register** User can monitor CNT to know the current value in 16-bit down counter. **Note:** It is recommended that read this register when PWM engine clock is source from system clock, otherwise a transition value of PWM counter may be read.

PWM Mask Control Register (PWM_MSKEN)

Register	Offset	R/W	Description	Reset Value
PWM_MSKEN	PWMx_BA+0x58	R/W	PWM Mask Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**MSKEN**					

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5:0]	**MSKEN**	**PWM Mask Enable Bits** The PWM output signal will be masked when this bit is enabled. The corresponding PWMn channel will be output with MSKDAT data. 0 = PWM output signal is non-masked. 1 = PWM output signal is masked and output with MSKDAT data. **Note:** Each bit controls the corresponding PWM channel.

PWM Mask Data Register (PWM_MSK)

Register	Offset	R/W	Description	Reset Value
PWM_MSK	PWMx_BA+0x5C	R/W	PWM Mask Data Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**MSKDAT**					

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5:0]	**MSKDAT**	**PWM Mask Data Bit:** This data bit control the state of PWMn output pin, if corresponding mask function is enabled. 0 = Output logic low to PWMn. 1 = Output logic high to PWMn. **Note:** Each bit controls the corresponding PWM channel.

PWM Dead-zone Control Register (PWM_DTCTL)

Register	Offset	R/W	Description	Reset Value
PWM_DTCTL	PWMx_BA+0x60	R/W	PWM Dead-zone Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	DTEN45	DTEN23	DTEN01	Reserved		DTDIV	
23	22	21	20	19	18	17	16
DTCNT45							
15	14	13	12	11	10	9	8
DTCNT23							
7	6	5	4	3	2	1	0
DTCNT01							

Bits	Description	
[31]	**Reserved**	Reserved.
[30]	**DTEN45**	**Dead-Zone Enable Control for PWM Pair Of Channel 4 and Channel 5** Dead-zone insertion is only active when this pair of complementary PWM is enabled. If Dead-zone insertion is inactive, the outputs of pin pair are complementary without any delay. 0 = Dead-zone insertion Disabled. 1 = Dead-zone insertion Enabled.
[29]	**DTEN23**	**Dead-Zone Enable Control for PWM Pair Of Channel 2 and Channel 3** Dead-zone insertion is only active when this pair of complementary PWM is enabled. If Dead-zone insertion is inactive, the outputs of pin pair are complementary without any delay. 0 = Dead-zone insertion Disabled. 1 = Dead-zone insertion Enabled.
[28]	**DTEN01**	**Dead-Zone Enable Control for PWM Pair Of Channel 0 and Channel 1** Dead-zone insertion is only active when this pair of complementary PWM is enabled. If Dead-zone insertion is inactive, the outputs of pin pair are complementary without any delay. 0 = Dead-zone insertion Disabled. 1 = Dead-zone insertion Enabled.
[27:26]	**Reserved**	Reserved.
[25:24]	**DTDIV**	**Dead-Zone Generator Divider** 00 = Dead-zone clock equal to PWM base clock divide 1. 01 = Dead-zone clock equal to PWM base clock divide 2. 10 = Dead-zone clock equal to PWM base clock divide 4. 11 = Dead-zone clock equal to PWM base clock divide 8.
[23:16]	**DTCNT45**	**Dead-Zone Interval For PWM Pair Of Channel 4 And Channel 5** These 8-bit determine the Dead-zone length.

		The unit time of Dead-zone length is received from corresponding PWM_CLKDIV.
[15:8]	**DTCNT23**	**Dead-Zone Interval For PWM Pair Of Channel 2 And Channel 3** These 8-bit determine the Dead-zone length. The unit time of Dead-zone length is received from corresponding PWM_CLKDIV.
[7:0]	**DTCNT01**	**Dead-Zone Interval For PWM Pair Of Channel 0 And Channel 1** These 8-bit determine the Dead-zone length. The unit time of Dead-zone length is received from corresponding PWM_CLKDIV.

Note: This register can only be updated when PWM stop running

PWM Trigger Control Register (PWM_TRGADCTL)

Register	Offset	R/W	Description	Reset Value
PWM_TRGADCTL	PWMx_BA+0x64	R/W	PWM Trigger Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		RTRGEN					
23	22	21	20	19	18	17	16
Reserved		FTRGEN					
15	14	13	12	11	10	9	8
Reserved		CTRGEN					
7	6	5	4	3	2	1	0
Reserved		PTRGEN					

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:24]	**RTRGEN**	**PWM Rising Edge Point Trigger Enable Bits** 0 = PWM rising edge point trigger ADC function Disabled. 1 = PWM rising edge point trigger ADC function Enabled. PWM can trigger ADC to start conversion when PWM output pin rising edge is detected if this bit is set to1. **Note:** Each bit controls the corresponding PWM channel.
[23:22]	**Reserved**	Reserved.
[21:16]	**FTRGEN**	**PWM Falling Edge Point Trigger Enable Bits** 0 = PWM falling edge point trigger ADC function Disabled. 1 = PWM falling edge point trigger ADC function Enabled. PWM can trigger ADC to start conversion when PWM output pin falling edge is detected if this bit is set to1. **Note:** Each bit controls the corresponding PWM channel.
[15:14]	**Reserved**	Reserved.
[13:8]	**CTRGEN**	**PWM Center Point Trigger Enable Bits** 0 = PWM center point trigger ADC function Disabled. 1 = PWM center point trigger ADC function Enabled. PWM can trigger ADC to start conversion when PWM counter up count to (PERIODn+1) if this bit is set to1. **Note1:** This bit should keep at 0 when PWM counter operating in Edge-aligned type. **Note2:** Each bit controls the corresponding PWM channel.
[7:6]	**Reserved**	Reserved.
[5:0]	**PTRGEN**	**PWM Period Point Trigger Enable Bits**

		0 = PWM period point trigger ADC function Disabled. 1 = PWM period point trigger ADC function Enabled. PWM can trigger ADC to start conversion when PWM counter down count to zero if this bit is set to1. **Note:** Each bit controls the corresponding PWM channel.

PWM Trigger Indicator Register (PWM_TRGADCSTS)

Register	Offset	R/W	Description	Reset Value
PWM_TRGADCSTS	PWMx_BA+0x68	R/W	PWM Trigger ADC Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		RTRGF					
23	22	21	20	19	18	17	16
Reserved		FTRGF					
15	14	13	12	11	10	9	8
Reserved		CTRGF					
7	6	5	4	3	2	1	0
Reserved		PTRGF					

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:24]	**RTRGF**	**PWM Rising Edge Point Trigger Indicator** This bit is set to 1 by hardware when PWM output pin rising edge is detected if corresponding RETRGEN bit is 1. After this bit is set to 1, ADC will start conversion if ADC triggered source is selected by PWM. **Note1:** Write 1 to clear this bit. **Note2:** Each bit controls the corresponding PWM channel.
[23:22]	**Reserved**	Reserved.
[21:16]	**FTRGF**	**PWM Falling Edge Point Trigger Indicator** This bit is set to 1 by hardware when PWM output pin falling edge is detected if corresponding FETRGEN bit is 1. After this bit is set to 1, ADC will start conversion if ADC triggered source is selected by PWM. **Note1:** Write 1 to clear this bit. **Note2:** Each bit controls the corresponding PWM channel.
[15:14]	**Reserved**	Reserved.
[13:8]	**CTRGF**	**PWM Center Point Trigger Flag** This bit is set to 1 by hardware when PWM counter up counts to (PERIODn+1) if the corresponding CTRGEN bit is 1. After this bit is set to 1, ADC will start conversion if ADC triggered source is selected by PWM. **Note1:** Write 1 to clear this bit. **Note2:** Each bit controls the corresponding PWM channel.
[7:6]	**Reserved**	Reserved.
[5:0]	**PTRGF**	**PWM Period Point Trigger Flag** This bit is set to 1 by hardware when PWM counter down count to zero if corresponding PTRGEN bit is 1. After this bit is set to 1, ADC will start conversion if ADC triggered source is selected by PWM. **Note1:** Write 1 to clear this bit.

		Note2: Each bit controls the corresponding PWM channel.

PWM Brake Control Register (PWM_BRKCTL)

Register	Offset	R/W	Description	Reset Value
PWM_BRKCTL	PWMx_BA+0x6C	R/W	PWM Brake Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		BKOD					
23	22	21	20	19	18	17	16
Reserved				LVDBKEN	CPO2BKEN	CPO1BKEN	CPO0BKEN
15	14	13	12	11	10	9	8
BRK1NFSEL		BK1SEL		Reserved	BRK1INV	BRK1NFDIS	BRKP1EN
7	6	5	4	3	2	1	0
BRK0NFSEL		Reserved		Reserved	BRK0INV	BRK0NFDIS	BRKP0EN

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:24]	**BKOD**	**PWM Brake Output Data Register** 0 = PWM output low when fault brake conditions asserted. 1 = PWM output high when fault brake conditions asserted. **Note:** Each bit controls the corresponding PWM channel.
[23:20]	**Reserved**	Reserved.
[19]	**LVDBKEN**	**Low-Level Detection Trigger PWM Brake Function 1 Enable Bit** 0 = Brake Function 1 triggered by Low-level detection Disabled. 1 = Brake Function 1 triggered by Low-level detection Enabled.
[18]	**CPO2BKEN**	**CPO2 Digital Output As Brake 0 Source Enable Bit** 0 = CPO2 as one brake source in Brake 0 Disabled. 1 = CPO2 as one brake source in Brake 0 Enabled.
[17]	**CPO1BKEN**	**CPO1 Digital Output As Brake 0 Source Enable Bit** 0 = CPO1 as one brake source in Brake 0 Disabled. 1 = CPO1 as one brake source in Brake 0 Enabled.
[16]	**CPO0BKEN**	**CPO0 Digital Output As Brake0 Source Enable Bit** 0 = CPO0 as one brake source in Brake 0 Disabled. 1 = CPO0 as one brake source in Brake 0 Enabled.
[15:14]	**BRK1NFSEL**	**Brake 1 (BKPx1 Pin) Edge Detector Filter Clock Selection** 00 = Filter clock = HCLK. 01 = Filter clock = HCLK/2. 10 = Filter clock = HCLK/4. 11 = Filter clock = HCLK/16.
[13:12]	**BK1SEL**	**Brake Function 1 Source Selection**

		00 = From external pin BKP1. 01 = From analog comparator 0 output (CPO0). 10 = From analog comparator 1 output (CPO1). 11 = Reserved.
[11]	**Reserved**	Reserved.
[10]	**BRK1INV**	**Inverse BKP1 State** 0 = The state of pin BKPx1 is passed to the negative edge detector. 1 = The inversed state of pin BKPx1 is passed to the negative edge detector.
[9]	**BRK1NFDIS**	**PWM Brake 1 Noise Filter Disable Bit** 0 = Noise filter of PWM Brake 1 Enabled. 1 = Noise filter of PWM Brake 1 Disabled.
[8]	**BRKP1EN**	**Brake1 Function Enable Bit** 0 = Brake1 function Disabled. 1 = Brake1 function Enabled.
[7:6]	**BRK0NFSEL**	**Brake 0 (BKPx0 Pin) Edge Detector Filter Clock Selection** 00 = Filter clock = HCLK. 01 = Filter clock = HCLK/2. 10 = Filter clock = HCLK/4. 11 = Filter clock = HCLK/16.
[5:3]	**Reserved**	Reserved.
[2]	**BRK0INV**	**Inverse BKP0 State** 0 = The state of pin BKPx0 is passed to the negative edge detector. 1 = The inversed state of pin BKPx0 is passed to the negative edge detector.
[1]	**BRK0NFDIS**	**PWM Brake 0 Noise Filter Disable Bit** 0 = Noise filter of PWM Brake 0 Enabled. 1 = Noise filter of PWM Brake 0 Disabled.
[0]	**BRKP0EN**	**Brake0 Function Enable Bit** 0 = Brake0 detect function Disabled. 1 = Brake0 detect function Enabled.

PWM Interrupt Control Register (PWM_INTCTL)

Register	Offset	R/W	Description	Reset Value
PWM_INTCTL	PWMx_BA+0x70	R/W	PWM Interrupt Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		DINTTYPE					
7	6	5	4	3	2	1	0
Reserved		PINTTYPE					

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13:8]	**DINTTYPE**	**PWM Duty Interrupt Type Selection** 0 = DIF[n] will be set if PWM counter matches PWM_CMPDATn register during down counting. 1 = DIF[n] will be set if PWM counter matches PWM_CMPDATn register during up counting. **Note1:** This bit should keep at 0 when PWM counter operating in Edge-aligned type. **Note2:** Each bit controls the corresponding PWM channel.
[7:6]	**Reserved**	Reserved.
[5:0]	**PINTTYPE**	**PWM Period Interrupt Type Selection** 0 = PIF[n] will be set if PWM counter underflow. 1 = PIF[n] will be set if PWM counter matches PWM_PERIODn register. **Note1:** This bit should keep at 0 when PWM counter operating in Edge-aligned type. **Note2:** Each bit controls the corresponding PWM channel.

PWM Interrupt Enable Control Register (PWM_INTEN)

Register	Offset	R/W	Description	Reset Value
PWM_INTEN	PWMx_BA+0x74	R/W	PWM Interrupt Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		FLIEN					
23	22	21	20	19	18	17	16
Reserved		RLIEN					
15	14	13	12	11	10	9	8
Reserved		DIEN					
7	6	5	4	3	2	1	0
Reserved	BRKIEN	PIEN					

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:24]	**FLIEN**	**Falling Latch Interrupt Enable Bit** 0 = Falling latch interrupt Disabled. 1 = Falling latch interrupt Enabled. **Note:** Each bit controls the corresponding PWM channel.
[23:22]	**Reserved**	Reserved.
[21:16]	**RLIEN**	**Rising Latch Interrupt Enable Bits** 0 = Rising latch interrupt Disabled. 1 = Rising latch interrupt Enabled. **Note:** Each bit controls the corresponding PWM channel.
[15:14]	**Reserved**	Reserved.
[13:8]	**DIEN**	**PWM Duty Interrupt Enable Bits** 0 = Duty interrupt Disabled. 1 = Duty interrupt Enabled. **Note:** Each bit controls the corresponding PWM channel.
[7]	**Reserved**	Reserved.
[6]	**BRKIEN**	**Brake0 and Brak1 Interrupt Enable Bit** 0 = Disabling flags BFK0 and BFK1 to trigger PWM interrupt. 1 = Enabling flags BRKIF0 and BRKIF1 can trigger PWM interrupt.
[5:0]	**PIEN**	**PWM Period Interrupt Enable Bits** 0 = Period interrupt Disabled. 1 = Period interrupt Enabled. **Note:** Each bit controls the corresponding PWM channel.

PWM Interrupt Flag Register (PWM_INTSTS)

Register	Offset	R/W	Description	Reset Value
PWM_INTSTS	PWMx_BA+0x78	R/W	PWM Interrupt Flag Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		CFLIF					
23	22	21	20	19	18	17	16
BRKSTS1	BRKSTS0	CRLIF					
15	14	13	12	11	10	9	8
Reserved	BRKLK0	DIF					
7	6	5	4	3	2	1	0
BRKIF1	BRKIF0	PIF					

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:24]	**CFLIF**	**Capture Falling Latch Interrupt Flag** 0 = No capture falling latch condition happened. 1 = Capture falling latch condition happened, this flag will be set to high. **Note:** This bit must be cleared by writing 1 to it.
[23]	**BRKSTS1**	**Brake 1 Status (Read Only)** 0 = PWM had been out of Brake 1 state. 1 = PWM is in Brake 1 state.
[22]	**BRKSTS0**	**Brake 0 Status (Read Only)** 0 = PWM had been out of Brake 0 state. 1 = PWM is in Brake 0 state.
[21:16]	**CRLIF**	**Capture Rising Latch Interrupt Flag** 0 = No capture rising latch condition happened. 1 = Capture rising latch condition happened, this flag will be set to high. **Note:** This bit must be cleared by writing 1 to it.
[15]	**Reserved**	Reserved.
[14]	**BRKLK0**	**PWM Brake0 Locked** 0 = Brake 0 state is released. 1 = When PWM Brake detects a falling signal at BKP0, this flag will be set to high to indicate the Brake0 state is locked. **Note:** This bit must be cleared by writing 1 to it.
[13:8]	**DIF**	**PWM Duty Interrupt Flag** Flag is set by hardware when channel 0 PWM counter down count and reaches CMP0. Software can clear this bit by writing 1 to it. **Note:** If CMP is equal to PERIOD, this flag is not working in edge-aligned type selection.

[7]	**BRKIF1**	**PWM Brake1 Flag** 0 = PWM Brake 1 is able to poll falling signal at BKP1 and has not recognized any one. 1 = When PWM Brake 1 detects a falling signal at pin BKP1, this flag will be set to high. **Note:** This bit must be cleared by writing 1 to it.
[6]	**BRKIF0**	**PWM Brake0 Flag** 0 = PWM Brake 0 is able to poll falling signal at BKP0 and has not recognized any one. 1 = When PWM Brake 0 detects a falling signal at BKP0, this flag will be set to high. **Note:** This bit must be cleared by writing 1 to it.
[5:0]	**PIF**	**PWM Period Interrupt Flag** This bit is set by hardware when PWM counter reaches the requirement condition of interrupt (depending on **PINTTYPE (PWM_INTCTL[n])**). Software can write 1 to clear this bit to 0.

PWM Output Enable Control Register (PWM_POEN)

Register	Offset	R/W	Description	Reset Value
PWM_POEN	PWMx_BA+0x7C	R/W	PWM Output Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		POEN					

Bits	Description	
[31:6]	Reserved	Reserved.
[5:0]	POEN	**PWM Pin Output Enable Bit** 0 = PWM pin at tri-state. 1 = PWM pin in output mode. **Note:** Each bit controls the corresponding PWM channel.

PWM Capture Control Register (PWM_CAPCTL)

Register	Offset	R/W	Description	Reset Value
PWM_CAPCTL	PWMx_BA+0x80	R/W	PWM Capture Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		FCRLDEN					
23	22	21	20	19	18	17	16
Reserved		RCRLDEN					
15	14	13	12	11	10	9	8
Reserved		CAPINV					
7	6	5	4	3	2	1	0
Reserved		CAPEN					

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:24]	**FCRLDEN**	**Falling Latch Reload Enable Bits** 0 = Falling latch reload counter Disabled. 1 = Falling latch reload counter Enabled.
[23:22]	**Reserved**	Reserved.
[21:16]	**RCRLDEN**	**Rising Latch Reload Enable Bits** 0 = Rising latch reload counter Enabled. 1 = Rising latch reload counter Enabled.
[15:14]	**Reserved**	Reserved.
[13:8]	**CAPINV**	**Capture Inverter Enable Bits** 0 = Capture source inverter Disabled. 1 = Capture source inverter Enabled. Reverse the input signal from GPIO **Note:** Each bit controls the corresponding PWM channel.
[7:6]	**Reserved**	Reserved.
[5:0]	**CAPEN**	**Capture Function Enable Bits** 0 = Capture function Disabled. RCAPDAT and FCAPDAT will not be updated. 1 = Capture function Enabled. Capture latched the PWM counter value and saved to RCAPDAT (Rising latch) and FCAPDAT (Falling latch). **Note:** Each bit controls the corresponding PWM channel.

PWM Capture Input Enable Control Register (PWM_CAPINEN)

Register	Offset	R/W	Description	Reset Value
PWM_CAPINEN	PWMx_BA+0x84	R/W	PWM Capture Input Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**CAPINEN**					

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5:0]	**CAPINEN**	**Capture Input Enable Bits** 0 = PWM Channel capture input path Disabled. The input of PWM channel capture function is always regarded as 0. 1 = PWM Channel capture input path Enabled. The input of PWM channel capture function comes from correlative multifunction pin. **Note:** Each bit controls the corresponding PWM channel.

PWM Capture Status Register (PWM_CAPSTS)

Register	Offset	R/W	Description	Reset Value
PWM_CAPSTS	PWMx_BA+0x88	R	PWM Capture Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		FLIFOV					
7	6	5	4	3	2	1	0
Reserved		CRIFOV					

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13:8]	**FLIFOV**	**Falling Latch Interrupt Flag Overrun Status** This flag indicates if falling latch happened when the corresponding CFLIF is 1 **Note:** This bit will be cleared automatically when user clear corresponding CFLIF.
[7:6]	**Reserved**	Reserved.
[5:0]	**CRIFOV**	**Rising Latch Interrupt Flag Overrun Status** This flag indicates if rising latch happened when the corresponding CRLIF is 1 **Note:** This bit will be cleared automatically when user clear corresponding CRLIF.

PWM Capture Rising Latch Register5-0 (PWM_RCAPDAT5-0)

Register	Offset	R/W	Description	Reset Value
PWM_RCAPDAT0	PWMx_BA+0x90	R	PWM Capture Rising Latch Register 0	0x0000_0000
PWM_RCAPDAT1	PWMx_BA+0x98	R	PWM Capture Rising Latch Register 1	0x0000_0000
PWM_RCAPDAT2	PWMx_BA+0xA0	R	PWM Capture Rising Latch Register 2	0x0000_0000
PWM_RCAPDAT3	PWMx_BA+0xA8	R	PWM Capture Rising Latch Register 3	0x0000_0000
PWM_RCAPDAT4	PWMx_BA+0xB0	R	PWM Capture Rising Latch Register 4	0x0000_0000
PWM_RCAPDAT5	PWMx_BA+0xB8	R	PWM Capture Rising Latch Register 5	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RCAPDAT[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RCAPDAT[7:0]							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**RCAPDAT**	**Capture Rising Latch Register** Latch the PWM counter when Channel 0/1/2/3/4/5 has rising transition.

PWM Capture Falling Latch Register5-0 (PWM_FCAPDAT5-0)

Register	Offset	R/W	Description	Reset Value
PWM_FCAPDAT0	PWMx_BA+0x94	R	PWM Capture Falling Latch Register 0	0x0000_0000
PWM_FCAPDAT1	PWMx_BA+0x9C	R	PWM Capture Falling Latch Register 1	0x0000_0000
PWM_FCAPDAT2	PWMx_BA+0xA4	R	PWM Capture Falling Latch Register 2	0x0000_0000
PWM_FCAPDAT3	PWMx_BA+0xAC	R	PWM Capture Falling Latch Register 3	0x0000_0000
PWM_FCAPDAT4	PWMx_BA+0xB4	R	PWM Capture Falling Latch Register 4	0x0000_0000
PWM_FCAPDAT5	PWMx_BA+0xBC	R	PWM Capture Falling Latch Register 5	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FCAPDAT[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FCAPDAT[7:0]							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**FCAPDAT**	**Capture Falling Latch Register** Latch the PWM counter when Channel 0/1/2/3/4/5 has Falling transition.

PWM0 Synchronous Busy Status Register (PWM_SBS0)

Register	Offset	R/W	Description	Reset Value
PWM_SBS0	PWMx_BA+0xE0	R	PWM0 Synchronous Busy Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**SYNCBUSY**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**SYNCBUSY**	**PWM Synchronous Busy** When software writes PWM_PERIOD0/PWM_CMPDAT0/PWM_CLKPSC or switch PWM0 counter operation mode CNTMOD (PWM_CTL[16]), PWM will have a busy time to update these values completely because PWM clock may be different from system clock domain. Software needs to check this busy status before writes PWM_PERIOD0/PWM_CMPDAT0/ PWM_CLKPSC or switch PWM0 counter operation mode to make sure previous setting has been update completely. This bit will be set when software writes PWM_PERIOD0/PWM_CMPDAT0/PWM_CLKPSC or switch PWM0 operation mode CNTMOD (PWM_CTL[16]) and will be cleared by hardware automatically when PWM update these value completely.

PWM1 Synchronous Busy Status Register (PWM_SBS1)

Register	Offset	R/W	Description	Reset Value
PWM_SBS1	PWMx_BA+0xE4	R	PWM1 Synchronous Busy Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**SYNCBUSY**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**SYNCBUSY**	**PWM Synchronous Busy** When software writes PWM_PERIOD1/PWM_CMPDAT1/PWM_CLKPSC or switch PWM1 counter operation mode CNTMOD (PWM_CTL [17]), PWM will have a busy time to update these values completely because PWM clock may be different from system clock domain. Software needs to check this busy status before writes PWM_PERIOD1/PWM_CMPDAT1/ PWM_CLKPSC or switch PWM1 counter operation mode to make sure previous setting has been update completely. This bit will be set when software writes PWM_PERIOD1/PWM_CMPDAT1/PWM_CLKPSC or switch PWM1 operation mode CNTMOD (PWM_CTL [17]) and will be cleared by hardware automatically when PWM update these value completely.

PWM2 Synchronous Busy Status Register (PWM_SBS2)

Register	Offset	R/W	Description	Reset Value
PWM_SBS2	PWMx_BA+0xE8	R	PWM2 Synchronous Busy Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**SYNCBUSY**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**SYNCBUSY**	**PWM Synchronous Busy** When software writes PWM_PERIOD2/PWM_CMPDAT2/PWM_CLKPSC or switch PWM2 counter operation mode CNTMOD (PWM_CTL [18]), PWM will have a busy time to update these values completely because PWM clock may be different from system clock domain. Software needs to check this busy status before writes PWM_PERIOD2/PWM_CMPDAT2/ PWM_CLKPSC or switch PWM2 counter operation mode to make sure previous setting has been update completely. This bit will be set when software writes PWM_PERIOD2/PWM_CMPDAT2/PWM_CLKPSC or switch PWM2 operation mode CNTMOD (PWM_CTL [18]) and will be cleared by hardware automatically when PWM update these value completely.

PWM3 Synchronous Busy Status Register (PWM_SBS3)

Register	Offset	R/W	Description	Reset Value
PWM_SBS3	PWMx_BA+0xEC	R	PWM3 Synchronous Busy Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**SYNCBUSY**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**SYNCBUSY**	**PWM Synchronous Busy** When software writes PWM_PERIOD3/PWM_CMPDAT3/PWM_CLKPSC or switch PWM3 counter operation mode CNTMOD (PWM_CTL [19]), PWM will have a busy time to update these values completely because PWM clock may be different from system clock domain. Software needs to check this busy status before writes PWM_PERIOD3/PWM_CMPDAT3/ PWM_CLKPSC or switch PWM3 counter operation mode to make sure previous setting has been update completely. This bit will be set when software writes PWM_PERIOD3/PWM_CMPDAT3/PWM_CLKPSC or switch PWM3 operation mode CNTMOD (PWM_CTL [19]) and will be cleared by hardware automatically when PWM update these value completely.

PWM4 Synchronous Busy Status Register (PWM_SBS4)

Register	Offset	R/W	Description	Reset Value
PWM_SBS4	PWMx_BA+0xF0	R	PWM4 Synchronous Busy Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**SYNCBUSY**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**SYNCBUSY**	**PWM Synchronous Busy** When software writes PWM_PERIOD4/PWM_CMPDAT4/PWM_CLKPSC or switch PWM4 counter operation mode CNTMOD (PWM_CTL [20]), PWM will have a busy time to update these values completely because PWM clock may be different from system clock domain. Software needs to check this busy status before writes PWM_PERIOD4/PWM_CMPDAT4/ PWM_CLKPSC or switch PWM4 counter operation mode to make sure previous setting has been update completely. This bit will be set when software writes PWM_PERIOD4/PWM_CMPDAT4/PWM_CLKPSC or switch PWM4 operation mode CNTMOD (PWM_CTL [20]) and will be cleared by hardware automatically when PWM update these value completely.

PWM5 Synchronous Busy Status Register (PWM_SBS5)

Register	Offset	R/W	Description	Reset Value
PWM_SBS5	PWMx_BA+0xF4	R	PWM5 Synchronous Busy Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**SYNCBUSY**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**SYNCBUSY**	**PWM Synchronous Busy** When software writes PWM_PERIOD5/PWM_CMPDAT5/PWM_CLKPSC or switch PWM5 counter operation mode CNTMOD (PWM_CTL [21]), PWM will have a busy time to update these values completely because PWM clock may be different from system clock domain. Software needs to check this busy status before writes PWM_PERIOD5/PWM_CMPDAT5/ PWM_CLKPSC or switch PWM5 counter operation mode to make sure previous setting has been update completely. This bit will be set when software writes PWM_PERIOD5/PWM_CMPDAT5/PWM_CLKPSC or switch PWM5 operation mode CNTMOD (PWM_CTL [21]) and will be cleared by hardware automatically when PWM update these value completely.

6.22 Enhanced PWM Generator (EPWM)

6.22.1 Overview

This device is built in two PWM units with the same architecture which function is specially designed for driving motor control applications. Using the PWM, input capture module and QEI controller with proper control flow by software can easily drive the 3-phase Brushless DC motor, 3-phase AC induction motor and DC motor.

6.22.2 Features

Each unit supports the features below:

- Three independent 16-bit PWM duty control units with maximum 6 port pins:
  - 3 independent PWM output: EPWM0_CH0, EPWM0_CH2 and EPWM0_CH4 for Unit 0 EPWM1_CH0, EPWM1_CH2 and EPWM1_CH4 for Unit 1
    - 3 complementary PWM pairs, with each pin in a pair mutually complement to each other and capable of programmable dead-time insertion: (EPWMx_CH0, EPWMx_CH1), (EPWM_CMPDAT2, EPWMx_CH3) and (EPWMx_CH4, EPWMx_CH5) where x=0~1.
  - 3 synchronous PWM pairs, with each pin in a pair in-phase: (EPWMx_CH0, EPWMx_CH1), (EPWMx_CH2, EPWMx_CH3) and (EPWMx_CH4, EPWMx_CH5) where x=0~1
- Group control bits: EPWMx_CH2 and EPWMx_CH4 are synchronized with EPWMx_CH0
- Supports Edge aligned mode and Center aligned mode
- Programmable dead-time insertion between complementary paired PWMs
- Each pin of EPWMx_CH0 to EPWMx_CH5 has independent polarity setting control
- Mask output control for Electrically Commutated Motor operation
- Tri-state output at reset and brake state
- Hardware brake protection
- Two Interrupt Sources:
  - Interrupt is synchronously requested at PWM frequency when up/down counter comparison matched (edge and center aligned modes) or underflow (center aligned mode).
  - Interrupt is requested when external brake pins asserted
- PWM signals before polarity control stage are defined in the view of positive logic. The PWM ports is active high or active low are controlled by polarity control register.
- High Source/Sink current.

After CPU reset the internal output of the each PWM channels depends on the polarity setting. The interval between successive outputs is controlled by a 16–bit up/down counter which uses a software selectable clock source with configurable internal clock pre-scalar as its input. The PWM counter clock has the frequency as the clock source F_{PWM} = EPWMx_CLK/Pre-scalar; Here the EPWMx_CLK synchronized with CPU clock HCLK.

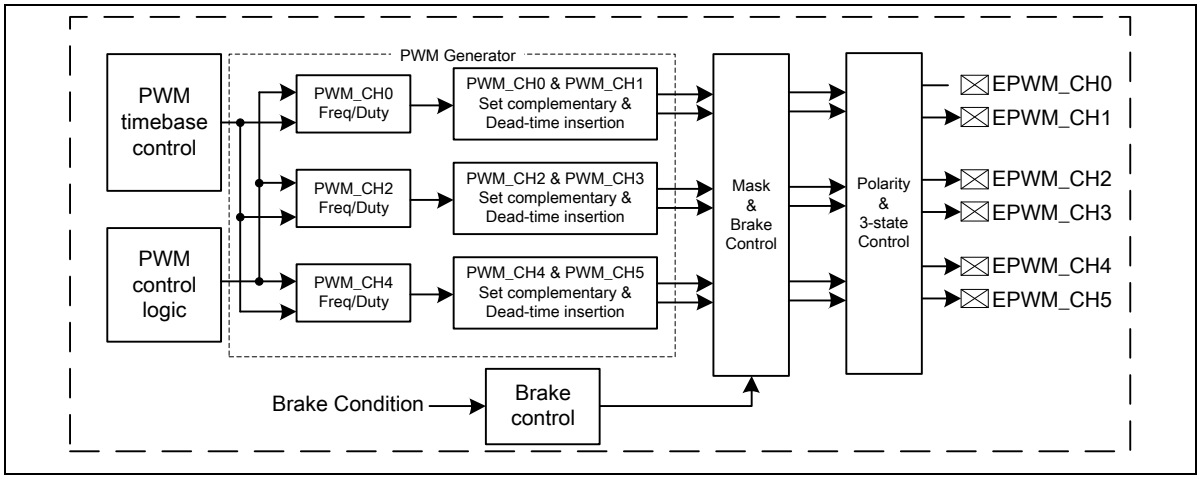


Figure 6.22-1 PWM Block Diagram

6.22.3 PWM Operation

The following diagrams show the PWM clock source control and PWM time-base generator.

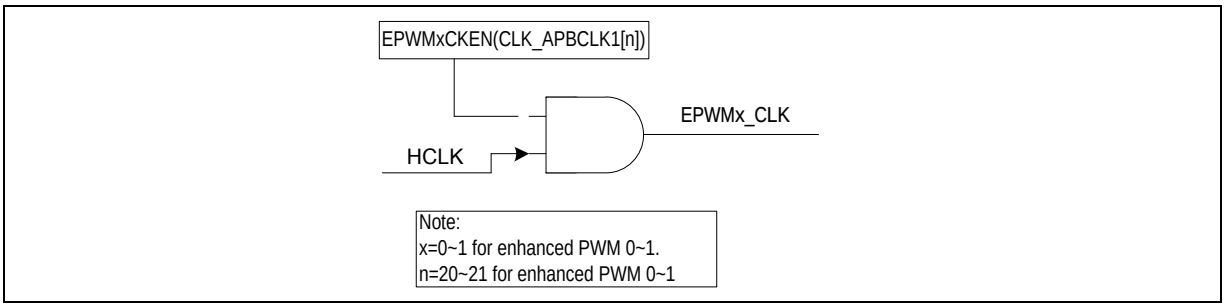


Figure 6.22-2 PWM Clock Source Control

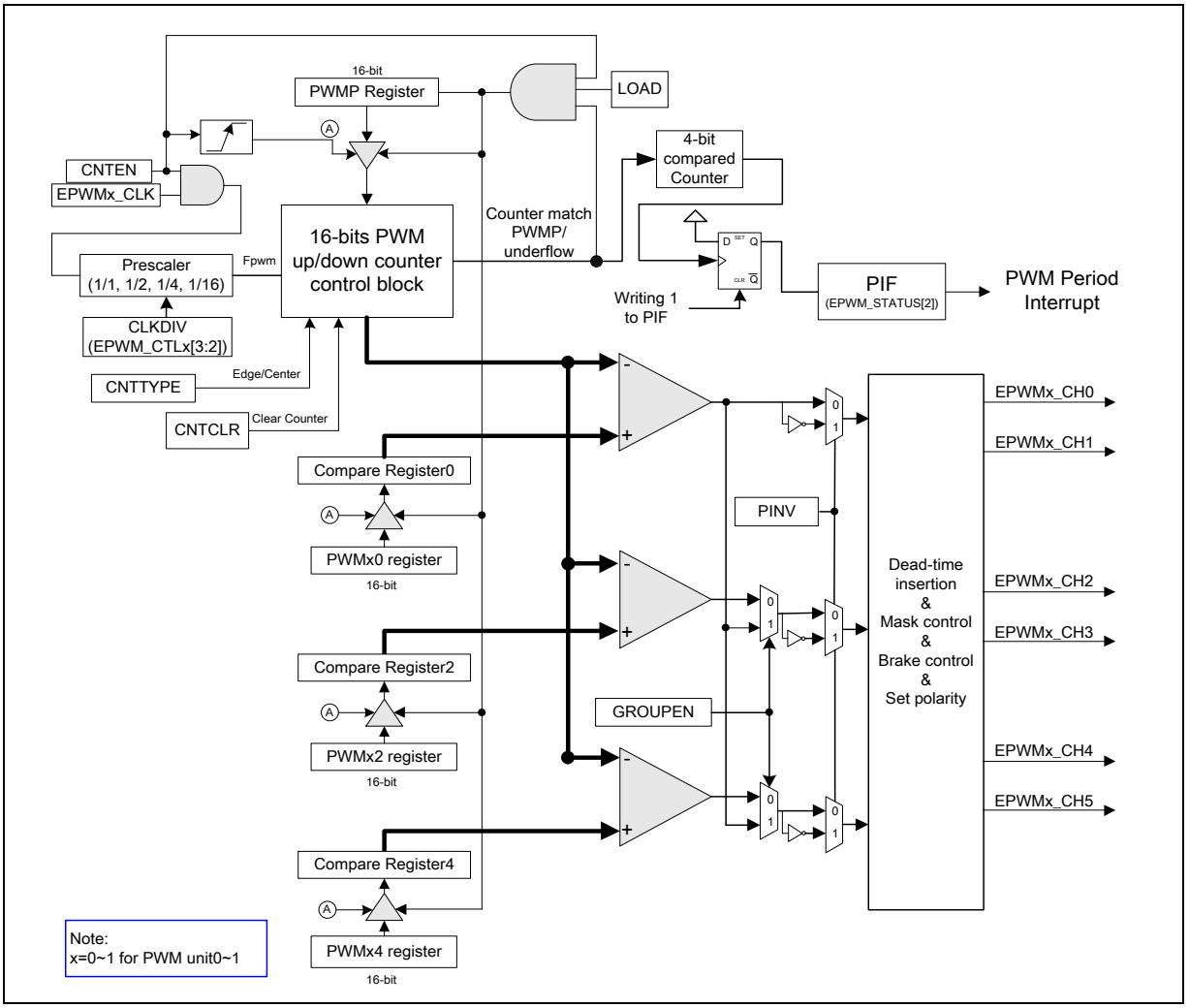


Figure 6.22-3 PWM Time-base Generator

The overall functioning of the PWM module is controlled by the contents of the EPWM_CTL, EPWM_PERIOD and EPWM_CMPDATn registers. The operation of most of the control bits is straightforward. CNTEN (EPWM_CTL[7]) allows the PWM to be either in the run or idle state. The transfer of the data from the EPWM_PERIOD register to 16-bit PWM period will occur on the rising edge of CNTEN or during CNTEN with Load (EPWM_CTL[6]) and PWM counter match/underflow occurs. The transfer of the data from the EPWM_CMPDATn registers to the compare registers is controlled by the Load (EPWM_CTL[6]) (with condition that CNTEN (EPWM_CTL[7])=1 and match/underflow occurs). [x=0-1].

Notes:

- A compare value greater than the counter reloaded value resulted in the PWM output being permanently high. In addition there are two special cases. **If compare register is set to 0x0000, the PWMn output will stay at low, and if compare register is set to 0xFFFF, the PWMn output will stuck at high until there is a change in the compare register.** [n = 0-3].
- During ICP, ISP or ICE mode, PWM pins will be tri-stated. PWM operation will stop and module reset. When exit from ICP, ISP or ICE mode, the PWM pins will follow the control register settings.
- In ICE mode, the condition which causes CPU stops or pauses running will force the PWMn output pins in tri-stated, when CPU runs the PWMn pins will follow the control register settings.

When software writing the EPWM_PERIOD register, the value is written into a holding register. The

transfer of data from this holding register, into the register which contains the actual reload value, occurs when the following conditions are met; Load = 1, CNTEN = 1 and PWM match/underflow. The width of each PWM output pulse is determined by the value in the appropriate compare register. Each PWM registers of **EPWM_PERIODx, EPWM_CMPDAT0, EPWM_CMPDAT2 and EPWM_CMPDAT4**, in the format of 16-bit width, decides the PWM period and each channel's duty cycle. If the PINV (Inverse PWM Comparator Output) is set to high the PWM comparator output signals will be inversed, therefore the PWM Duty (in percentage) is changed to (1-Duty) before PINV is set to high and PWM Duty registers, EPWM_CMPDAT0/2/4 represent Duty-off time.

Please take note that duty registers EPWM_CMPDAT0/2/4 and period registers EPWM_PERIODx are double-buffered registers used to set the duty cycle and counting period for the PWM time base respectively. For the 1st buffer it is accessible by user while the 2nd buffer holds the actual compare value used in the present period. Load (EPWM_CTLx0[6]) must be set to 1 to enable the value to be loaded in to the 2nd buffer register when counter underflow/match.

6.22.3.1 PWM Operation Mode

This device supports 2 operation modes: Edge-aligned and Center-aligned mode.

The following equations show the formula for period and duty for each pwm operation mode:

Edge aligned:

Period = (PERIOD + 1) * EPWMx_CLK period * pre-scalar

Duty = (Duty + 1) * EPWMx_CLK period * pre-scalar

Center aligned:

Period = (PERIOD * 2) * EPWMx_CLK period * pre-scalar

Duty = (Duty * 2 + 1) * EPWMx_CLK period * pre-scalar

Note: "CMP" refers to EPWM_CMPDAT0/2/4/6 register value.

Edge aligned PWM (up-counter)

In Edge-aligned PWM Output mode, the 16 bits PWM counter will starts counting from 0 to match with the value of the duty cycle PWM0 (old), when this happen it will toggle the PWM0 generator output to low. The counter will continue counting to match with the value of the period register EPWM_PERIOD (old), at this moment, it toggles the PWM0 generator output to high and new PWM0 (new) and EPWM_PERIOD (new) are updated with Load=1 and request the PWM interrupt if PWM interrupt is enabled(EIE1.6=1).

The following figures depict the Edge-aligned PWM timing and operation flow.

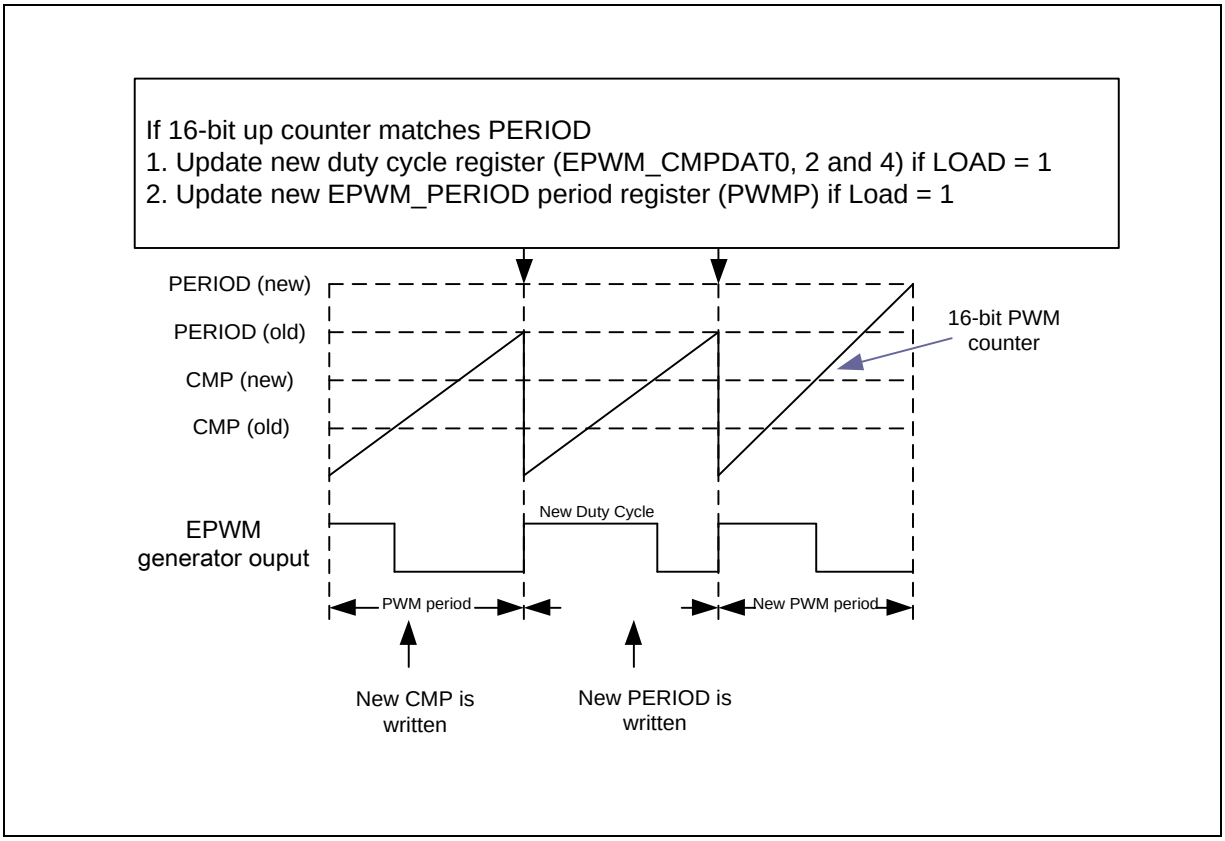


Figure 6.22-4 Edge-Aligned PWM

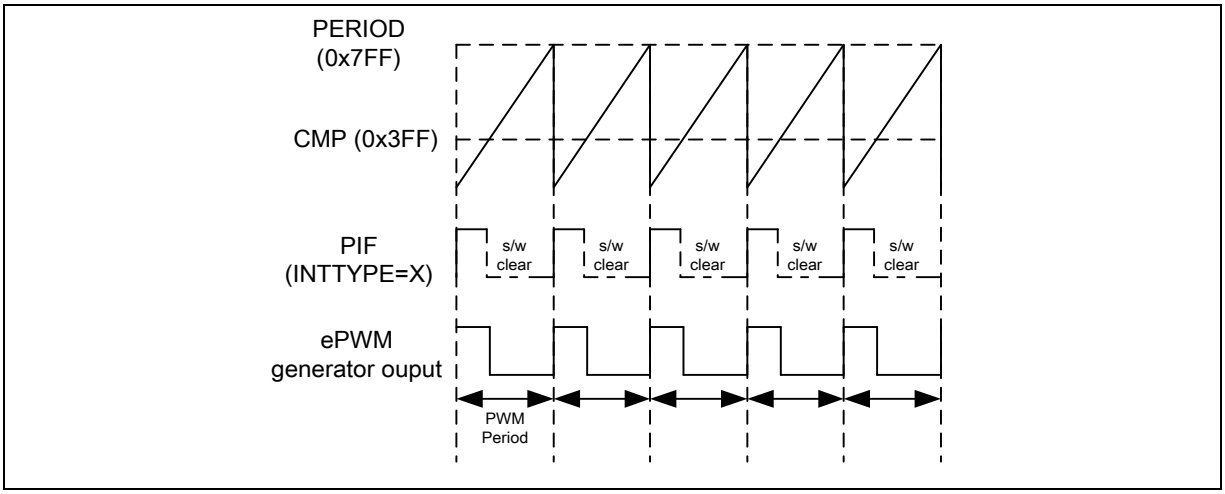


Figure 6.22-5 PWM0 Edge Aligned Waveform Output

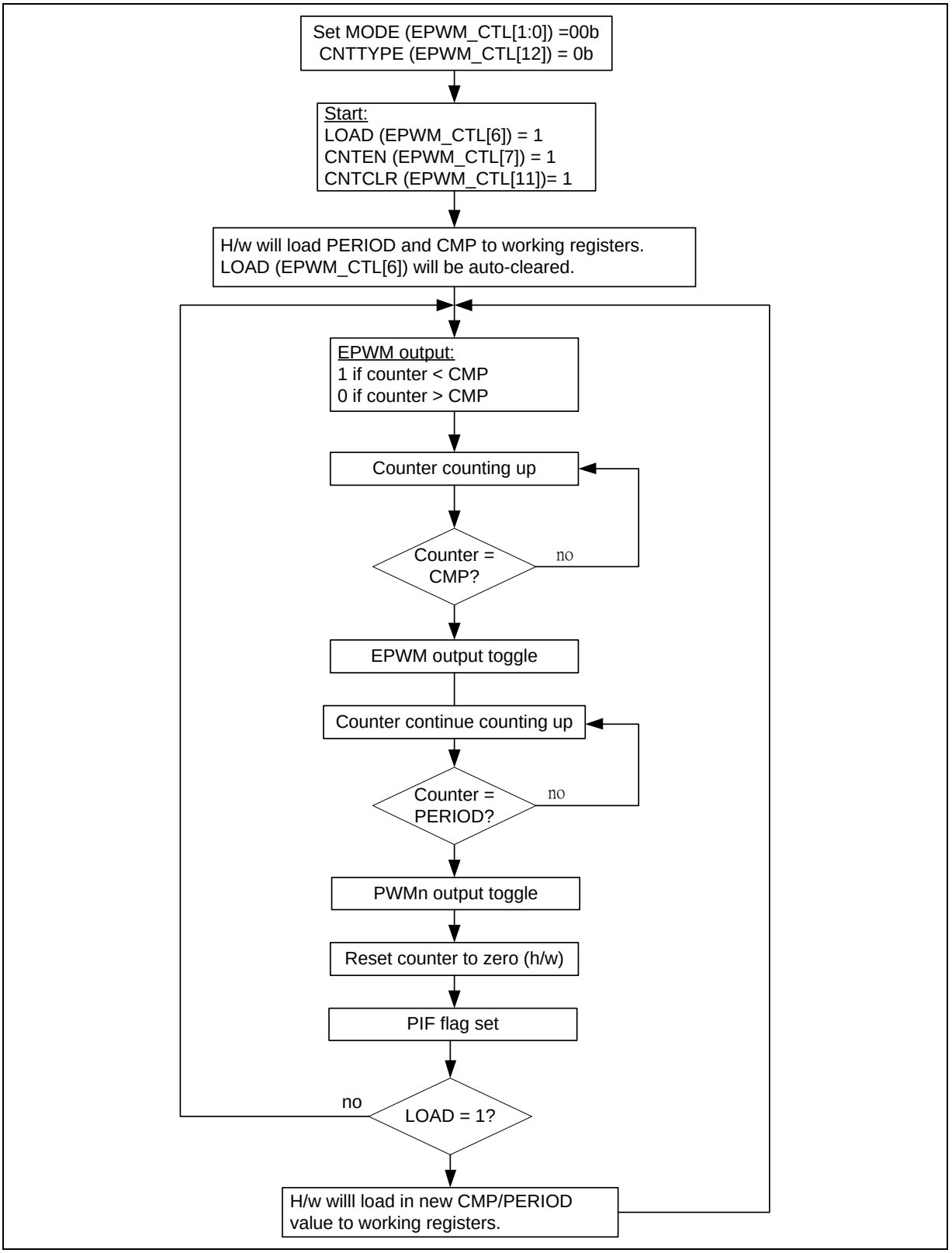


Figure 6.22-6 Edge-Aligned Flow Diagram

Center-Aligned PWM (up/down counter)

The Center-aligned PWM signals are produced by the module when the PWM time base is configured in an Up/Down Counting mode. The PWM counter will start counting-up from 0 to match the value of PWM_CMPDAT0 (old), this will cause the toggling of the PWM_CH0 generator output to low. The counter will continue counting to match with the PERIOD (old). Upon reaching this states counter is configured automatically to down counting, when PWM counter matches the PWM_CMPDAT0 (old) value again the PWM_CH0 generator output toggles to high. Once the PWM counter underflows it will update the PWM period register EPWM_PERIOD (new) and compare register EPWM_CMPDAT0(new) with Load = 1.

In Center-aligned mode, the PWM interrupt is requested at down-counter underflow if INTTYPE (EPWM_CTL[8]) =0, i.e. at start (end) of each PWM cycle or at up-counter matching with EPWM_PERIOD if INTTYPE (EPWM_CTL[8]) =1, i.e. at center point of PWM cycle.

The following figures depict the Center-aligned PWM timing and operation flow.

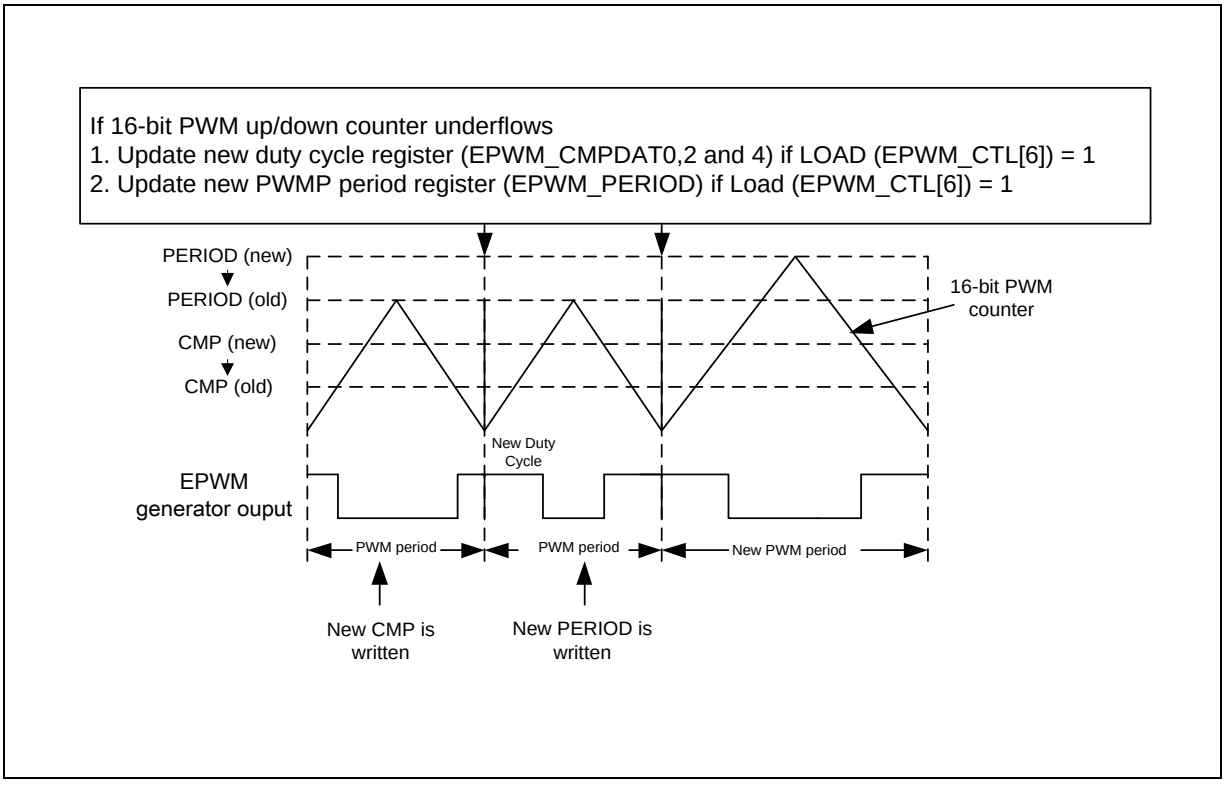


Figure 6.22-7 Center-Aligned Mode

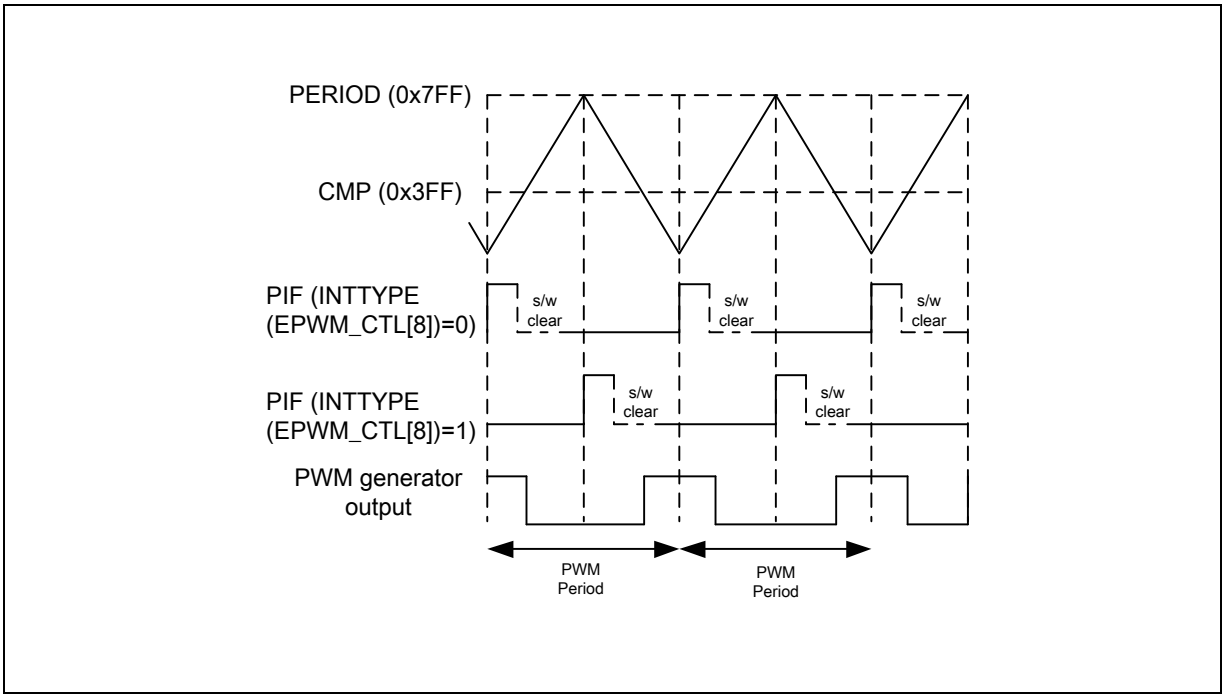


Figure 6.22-8 Example PWM0 Center-Aligned Waveform Output

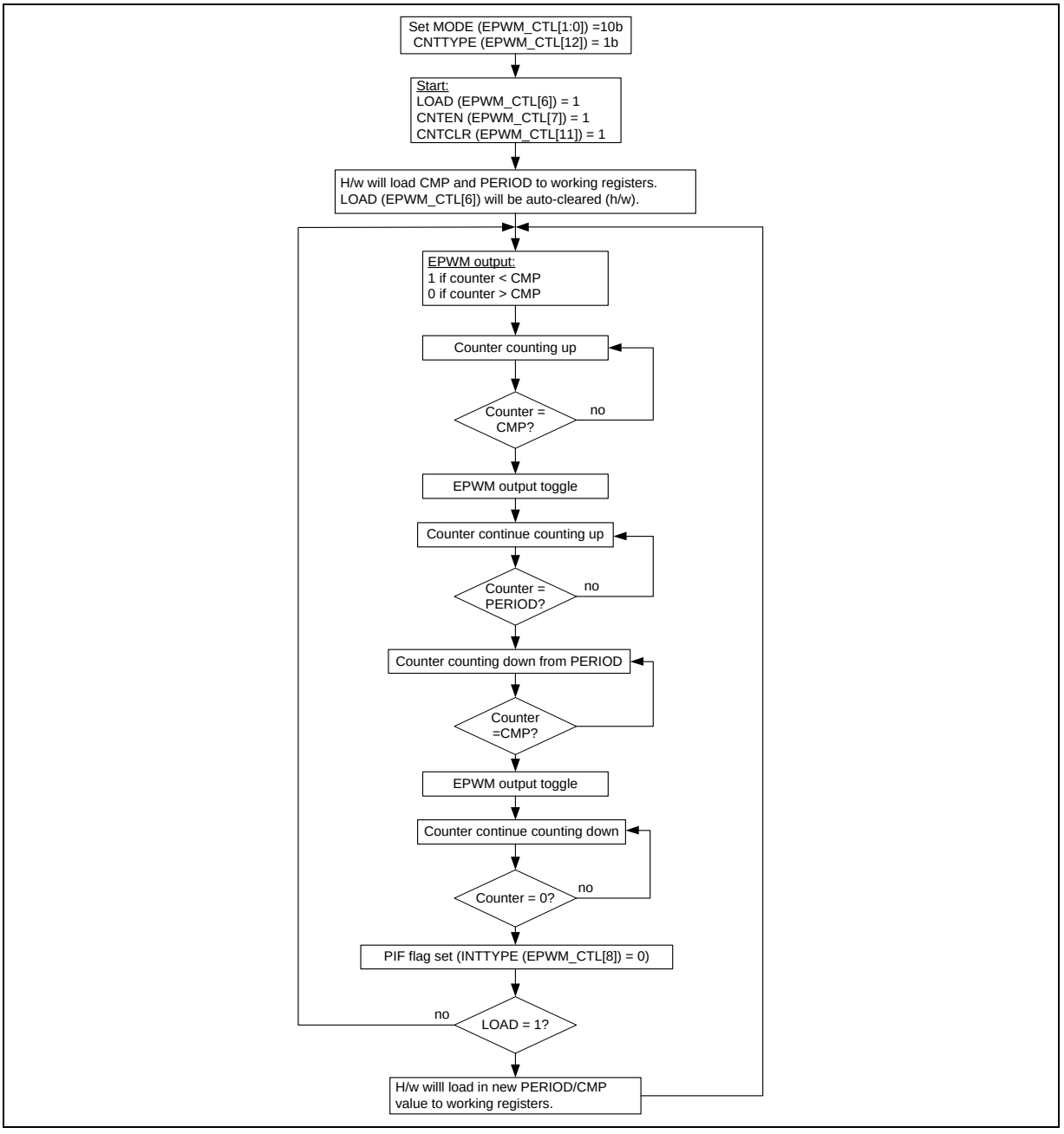
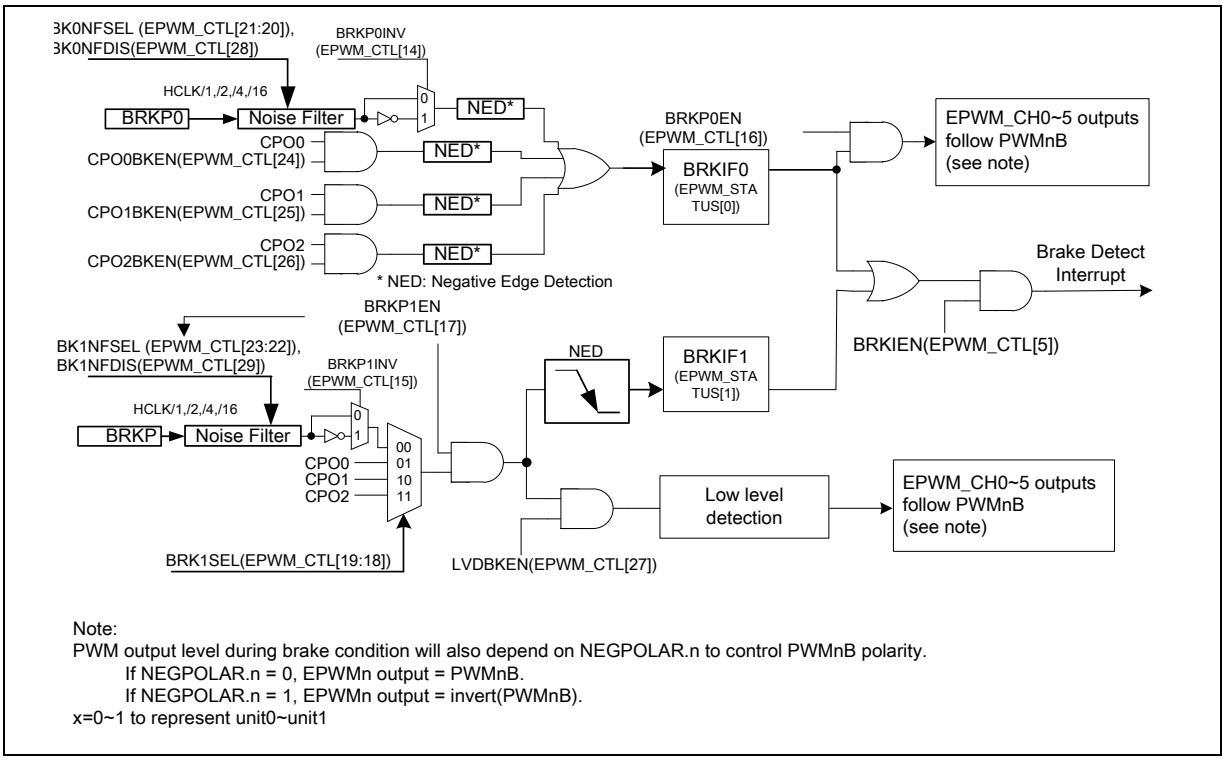


Figure 6.22-9 Center-aligned Flow Diagram (INTTYPE (EPWM_CTL[8]) = 0)

6.22.4 PWM Brake

This device supported 2 brake detectors, BK0 and BK1, each of them has 4 brake signals, one external brake pin (BKPx0 for BK0 and BKPx1 for BK1), analog comparator output and two OP amplifier digital outputs. Both external brake pins have each a 4-degree digital filter that is user controllable through BKxFILT.1-0 bits (x=0 and 1 for BK0 and BK1). The Brake function is controlled

by the contents of the SFR EPWM_CTL register



Note:
PWM output level during brake condition will also depend on NEGPOLAR.n to control PWMnB polarity.
If NEGPOLAR.n = 0, EPWMn output = PWMnB.
If NEGPOLAR.n = 1, EPWMn output = invert(PWMnB).
x=0~1 to represent unit0~unit1

Figure 6.22-10 PWM Brake Function

The following table summarizes the effect of each brake pins; Figure 6.22-11 and Figure 6.22-12 illustrate the brake signals vs. PWM operation.

Brake Source	Brake Trigger Condition	Actions
BKP0 (I/O pin); CPO0 (Comparator0 output); CPO1 (Comparator1 output); CPO2 (Comparator2 output);	A falling edge at brake 0 edge detector.	1. BRKIF0 and BRK0LOCK flags set by the falling edge at brake 0 edge detector; but cleared by software. 2. EPWM_CH0~5 outputs follow BRKOUTn bits. NEGPOLAR bits are also able to control the polarity of BRKOUTn. If NEGPOLAR.n=0, PWMn pin output = BRKOUTn. If NEGPOLAR.n=1, PWMn pin output = invert(BRKOUTn). 3. EPWM_CH0~5 will keep in BRKOUTn state before BRK0LOCK flag is cleared by software. 4. CNTEN bit remain asserted to keep PWM generators running. 5. If user clears BRK0LOCK flag, the brake state will be released on next PWM cycle/period. 6. If user clear CNTEN to 0 before BRK0LOCK flag is cleared, EPWM_CH0~5 will remain in BRKOUTn state.
BKP1 (I/O pin); CPO0 (Comparator0 output); CPO1 (Comparator1 output); CPO2	A falling edge at brake 1 edge detector; Low level state	1. BRKIF1 flag is set by the falling edge at brake 1 edge detector; but cleared by software. 2. If LVDBKEN=1 and Brake 1 signal detected as low level, EPWM_CH0~5 pin outputs follow BRKOUTn bits. Otherwise, EPWM_CH0~5 will continue follow PWM generators' output. In both situations, NEGPOLAR bits are also able to control port polarity. 3. CNTEN bit remain asserted to keep PWM generators running. 4. EPWM_CH0~5 resume if BKP1 pin state returns to high state.

(Comparator2 output)		EPWM_CH0~5 will resume on start of next PWM cycle/period.

Note: User must set brake enable bits, BRKP0EN and BRKP1EN, in order for the above to be effective.

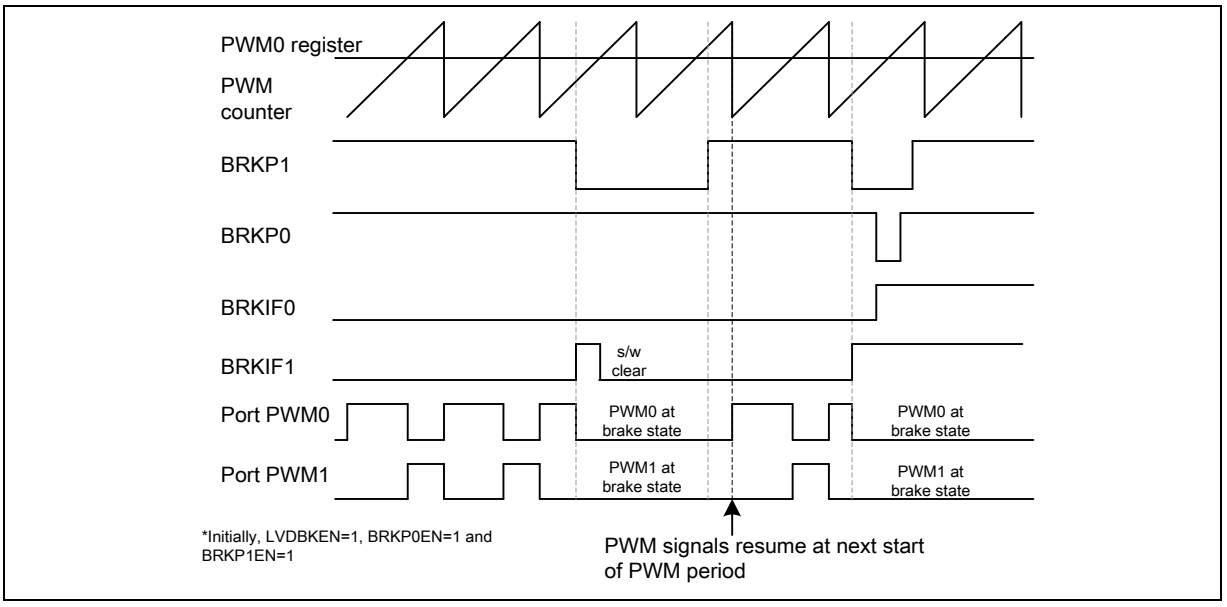


Figure 6.22-11 PWM Brake Condition (Edge-aligned Mode)

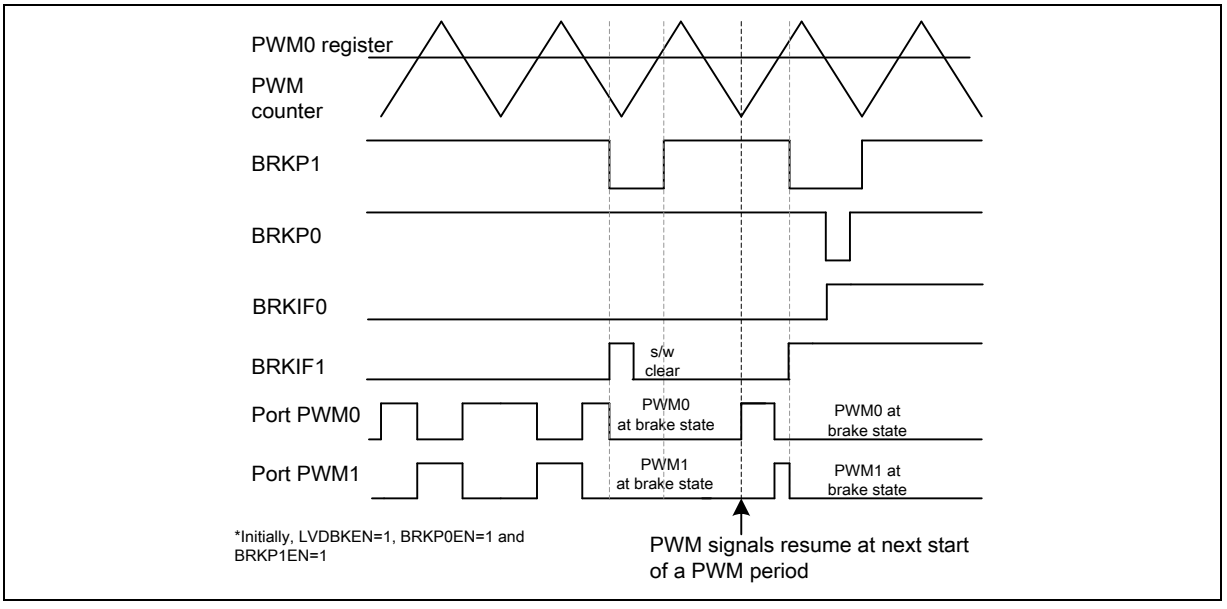


Figure 6.22-12 PWM Brake Condition (Centre-aligned Mode)

Since the both brake conditions being asserted will automatically cause BKFn flag will be set, the user program can poll these brake flag bits or enable PWM's brake interrupt to determine which condition causes a brake to occur.

6.22.5 PWM Port Output Driving Control

There are two enhanced PWM units each unit has six output pins in this device. The PWM port outputs are P0.0~P0.5 and P1.0~P1.5 for unit 0 and unit 1, respectively.

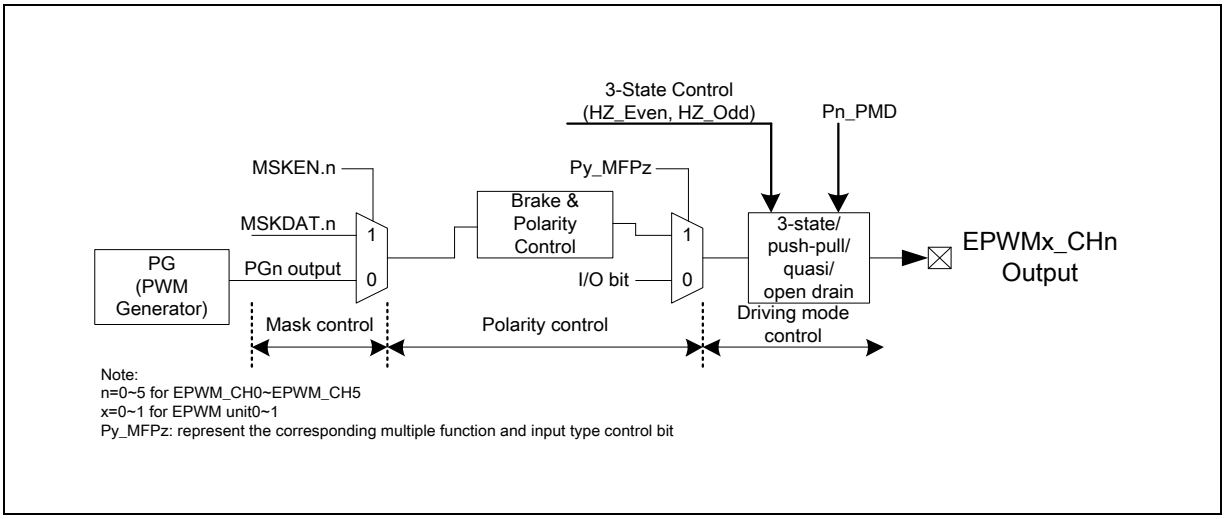


Figure 6.22-13 PWM Output Driving Control

The driving type of PWM output ports can be initialized as Tri-state type or other type dependent with Pn_MODE register setting after any reset. As shown in the above diagram, PWM output structures are controllable through config-bit, register bits PWMPOEN[EVENOUTEN0/1, ODDOUTEN0/1] and Pn_MODE mode registers.

EVENOUTEN/ODDOUTEN (In PWMPOEN Register)	Even/Odd PWM Outputs Drive Mode
0	Depending on Pn_MODE
1	Driving mode = Hi-Z.

Note: Register bits for EVENOUTEN and ODDOUTEN are latched from config0 during all reset.

6.22.6 PWM modes

This powerful PWM unit supports Independent mode which may be applied to DC and BLDC motor system, Complementary mode with dead-time insertion which may be used in the application of AC induction motor and synchronous motor, Synchronous mode that makes both pins of each pair are in phase. Besides, the Group mode, forces the EPWM_CMPDAT0, EPWM_CMPDAT2 and EPWM_CMPDAT4 synchronous with EPWM_CMPDAT0 generator, may simplify updating duty control in DC and BLDC motor applications.

6.22.6.1 Independent mode

Independent mode is enabled when MODE(EPWM_CTL0[1:0]) = 00b.

On default, the PWM is operating in independent mode, with three PWM even channels outputs:

EPWM_CH0, EPWM_CH2 and EPWM_CH4. Each channel is running off its own duty-cycle generator module. The states of PWM1, PWM3 and PWM5 are reset to zero by default if PWM Mask output function is not enabled (MSKEN=0x00).

6.22.6.2 Complementary mode

Complementary mode is enabled when MODE(EPWM_CTL0[1:0]) = 01b.

In this module there are three duty-cycle generators utilized for complementary mode, with total of three PWM output pair pins in this module. The total six PWM outputs are grouped into output pairs of even and odd numbered outputs. In complimentary modes, the internal odd PWM signal PGx must always be the complement of the corresponding even PWM signal. For example, PG1 will be the complement of PG0. PG3 will be the complement of PG2 and PG5 will be the complement of PG4. The time base for the PWM module is provided by its own 16-bit timer, which also incorporates selectable pre-scalar options.

6.22.6.3 Dead-Time Insertion

The dead time generator inserts an “off” period called “dead time” between the turnings off of one pin to the turning on of the complementary pin of the paired pins. This is to prevent damage to the power switching devices that will be connected to the PWM output pins. The complementary output pair mode has an **11-bit down counter** used to produce the dead time insertion. The complementary outputs are delayed until the timer counts down to zero.

The dead-time can be calculated from the following formula:

Dead-time = EPWMx_CLK period * (DTCNT[10:0]+1).

The timing diagram below indicates the dead time insertion for one pair of PWM signals.

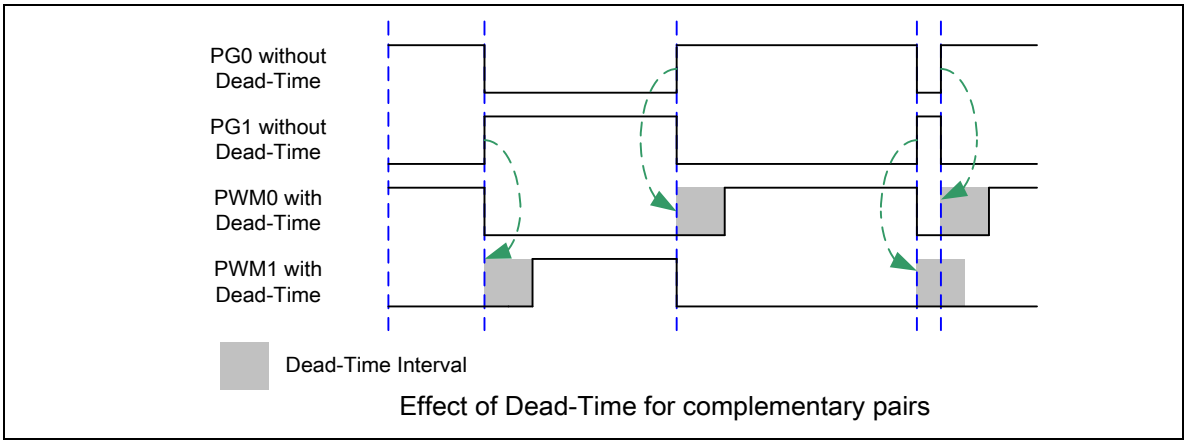


Figure 6.22-14 Dead-Time Insertion

The PWM Dead-time Control Register, EPWM_DTCTL, **have protection** in writing. In Power inverter application, a dead time insertion avoids the upper and lower switches of the half bridge from being active at the same time. Hence the dead time control is crucial to proper operation of a system. Some amount of time must be provided between turning off of one PWM output in a complementary pair and turning on the other transistor as the power output devices cannot switch instantaneously.

6.22.6.4 Synchronous mode

Synchronous mode is enabled when MODE[1:0] = 10b.

In the synchronization mode the PWM pair signals from PWM Generator are in-phase.

PG1=PG0, PG3=PG2 and PG5=PG4.

6.22.6.5 Group mode

Group mode is enabled when GROUPEN (EPWM_CTLx0[13]) = 1.

This device support Group Mode control. This control allows all even PWM channels output to be duty controllable by PWM0 duty register.

If GROUPEN = 1, both (PG2, PG3) and (PG4, PG5) pairs will follow (PG0, PG1), which imply;

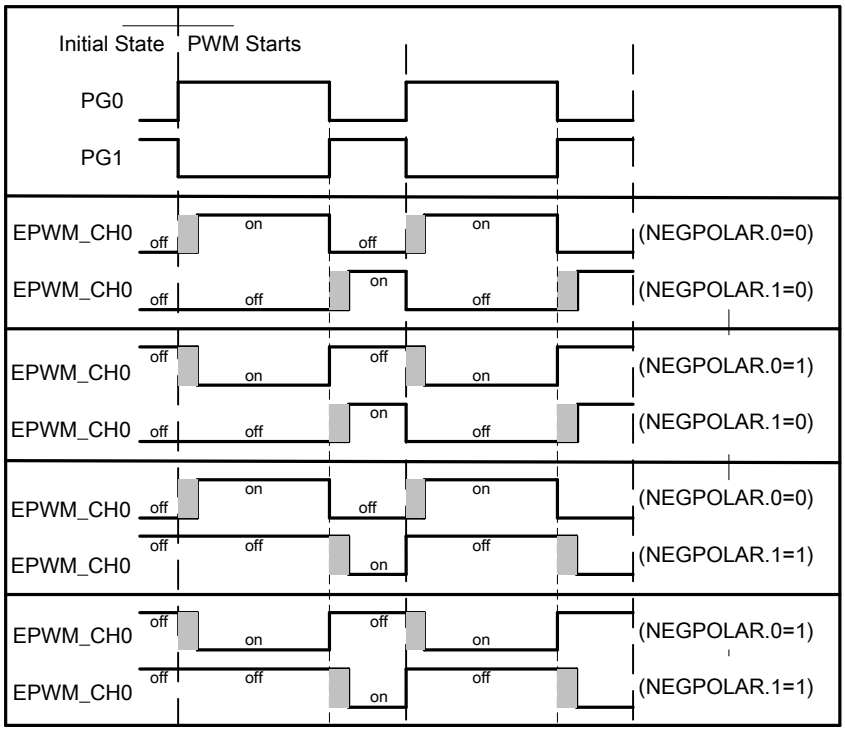
PG4 = PG2 = PG0;

PG5 = PG3 = PG1 = invert (PG0) if Complementary mode is enabled (PMOD.1-0=01b)

6.22.7 Polarity Control

Each PWM port of EPWM_CH0 to EPWM_CH5 has independent polarity control to configure the polarity of active state of PWM output. At default, the PWM output is active high. This implies the PWM OFF state is low and ON state is high. This is controllable through PWM Negative Polarity Control Register, EPWM_NPCTL, on each individual PWM channel.

The following diagram shows the initial state before PWM starts with different polarity settings.

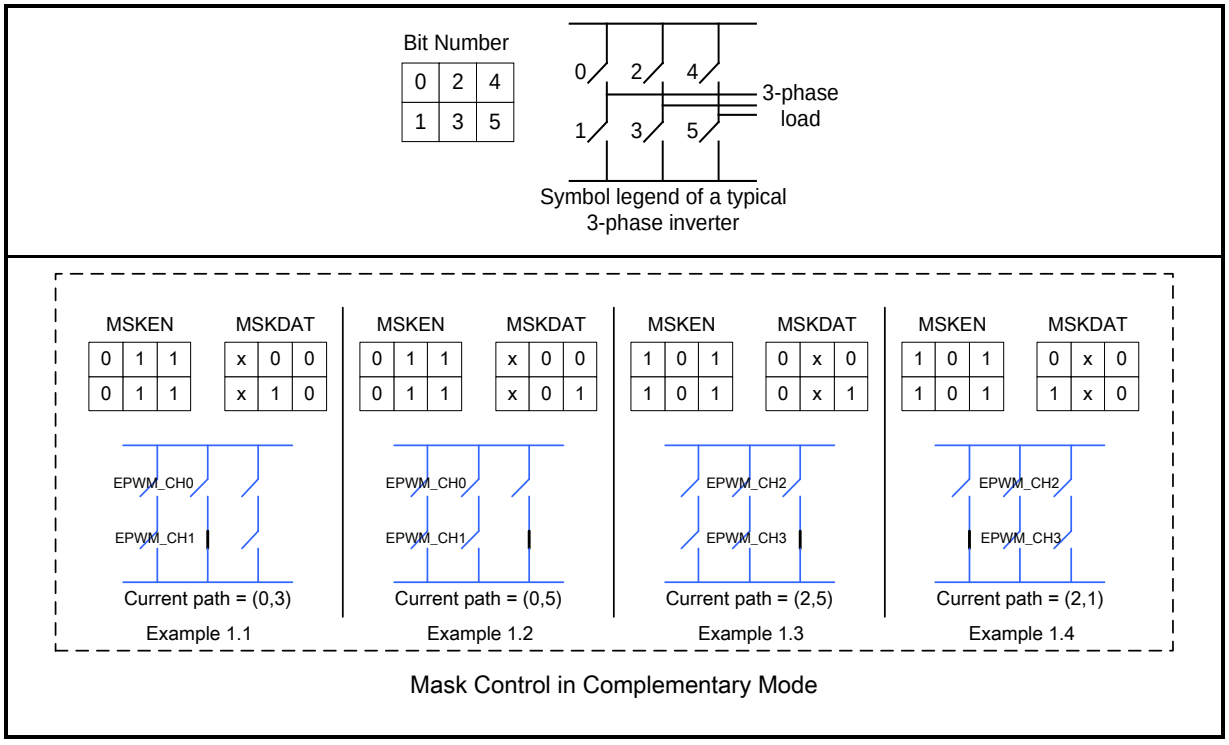


PWM output with dead-time insertion and polarity control

Figure 6.22-15 Initial State and Polarity Control with Rising Edge Dead Time Insertion

6.22.8 PWM Mask Output Function

Each of the PWM output channels can be manually overridden by using the appropriate bits in the PWM Mask Enable Control Register (EPWM_MSKEN) and PWM Masked Data Register (EPWM_MSK) to drive the PWM I/O pins to specified logic states independent of the duty cycle comparison units. The PWM mask bits are useful when controlling various types of Electrically Commutated Motor (ECM) like a BLDC motor. The EPWM_MSKEN register contains six bits, MSKEN(EPWM_MSKEN[5:0]) determine which PWM I/O pins will be overridden. On reset MSKEN(EPWM_MSKEN[5:0]) is 00H. The EPWM_MSK register contains six bits, MSKDAT(EPWM_MSK[5:0]) determine the state of the PWM I/O pins when a particular output is masked via the MSKDAT(EPWM_MSK[5:0]) bits. On reset MSKDAT(EPWM_MSK[5:0]) is 00H. The MSKEN(EPWM_MSKEN[5:0]) bits are active-high. When the MSKEN(EPWM_MSKEN[5:0]) bits are set, the corresponding MSKDAT(EPWM_MSK[5:0]) will have effect on the PWM channel. When one of the MSKEN(EPWM_MSKEN[5:0]) is sets, the output on the corresponding PWM I/O pin will be determined by the state of the MSKDAT(EPWM_MSK[5:0]) and polarity control bit.



Mask Control in Complementary Mode

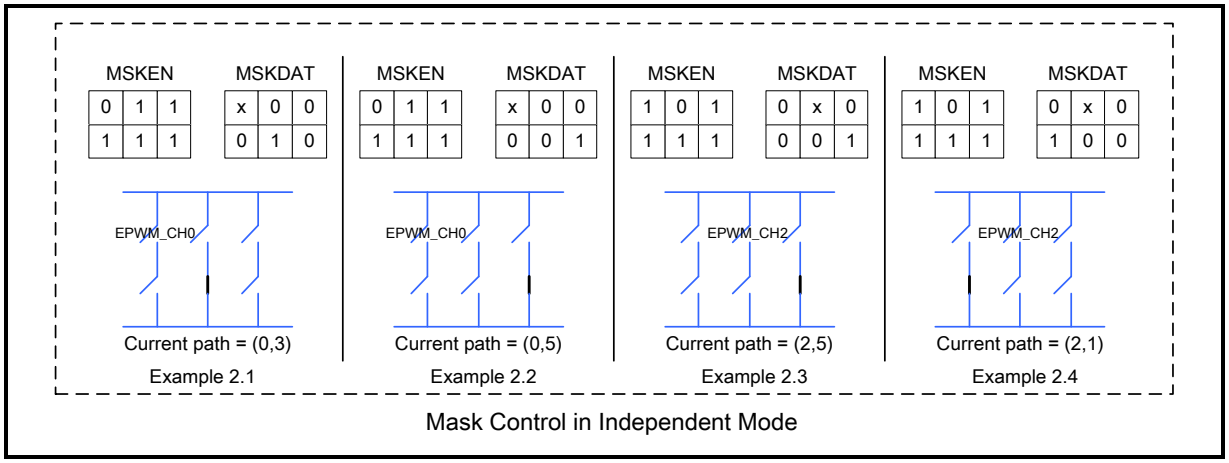


Figure 6.22-16 Illustration of Mask Control

The figure above shows example of how PWM mask control can be used for the override feature.

In example 1.1; MSKEN[5:0] = 11 1100b, MSKDAT[5:0] = 0010xxb (complementary mode)

PWM channels 2-5 are masked from PWM frequency/duty generators.

PWM channels 2-5 outputs are determined by state of MSKDAT bits.

PWM channels 0 and 1 follow PWM generator.

Switch 0 (On/Off) : Control by PWM_CH0 (PWM_CH0 frequency/duty generator).

Switch 1 (On/Off) : Control by PWM_CH1 (inverted of PWM_CH0, complementary mode).

Switch 2 (Off) : MSKDAT[2] = 0.

Switch 3 (On) : MSKDAT[3] = 1.

Switch 4 (Off) : MSKDAT[4] = 0.

Switch 5 (Off) : MSKDAT[5] = 0.

In example 1.3; MSKEN[5:0] = 11 0011b, MSKDAT[5:0] = 10xx00b (complementary mode)

PWM channels 0, 1, 4 and 5 are masked from PWM frequency/duty generators.

PWM channels 0, 1, 4 and 5 outputs are determined by state of MSKDAT bits.

PWM channels 2 and 3 follow PWM generator.

Switch 0 (Off) : MSKDAT[0] = 0.

Switch 1 (Off) : MSKDAT[1] = 0.

Switch 2 (On/Off) : Control by PWM_CH2 (PWM_CH2 frequency/duty generator).

Switch 3 (On/Off) : Control by PWM_CH3 (inverted of PWM_CH2, complementary mode).

Switch 4 (Off) : MSKDAT[4] = 0.

Switch 5 (On) : MSKDAT[5] = 1.

In example 2.1; MSKEN[5:0] = 11 1110b, MSKDAT[5:0] = 00100xb (independent mode)

PWM channels 1-5 are masked from PWM frequency/duty generators.

PWM channels 1-5 outputs are determined by state of MSKDAT bits.

PWM channel 0 follow PWM generator.

Switch 0 (On/Off) : Control by PWM_CH0 (PWM_CH0 frequency/duty generator).

Switch 1 (Off) : MSKDAT[1] = 0.

Switch 2 (Off) : MSKDAT[2] = 0.

Switch 3 (On) : MSKDAT[3] = 1.

Switch 4 (Off) : MSKDAT[4] = 0.

Switch 5 (Off) : MSKDAT[5] = 0.

In example 2.3; MSKEN[5:0] = 11 1011b, MSKDAT[5:0] = 100x00b (independent mode)

PWM channels 0,1,3,4 and 5 are masked from PWM frequency/duty generators.

PWM channels 0,1,3,4 and 5 outputs are determined by state of MSKDAT bits.

PWM channel 2 follow PWM generator.

Switch 0 (Off) : MSKDAT[0] = 0.

Switch 1 (Off) : MSKDAT[1] = 0.

Switch 2 (On/Off) : Control by PWM_CH2 (PWM_CH2 frequency/duty generator).

Switch 3 (Off) : MSKDAT[3] = 0.

Switch 4 (Off) : MSKDAT[4] = 0.

Switch 5 (On) : MSKDAT[5] = 1.

6.22.9 Asymmetric PWM Output

This device supports asymmetric PWM output. When asymmetric function is enabled, each PWM channel has an alternate compare register (EPWM_ASYMCMP0/2/4) which is reloaded into the PWM compare register after first time compare matched. The following legends example the PWM_CH0 output with its architecture and output waveform. The active duty width in asymmetric PWM output type, for example PWM_CH0 of unit0, is (PWM00 + ASPWM00 +1).

The following are the constraints in using Asymmetric PWM function:

1. PWM compare register must less than (ASYMCMP + 1) if ASYMMODE0(EPWM_ASYMCTL[8:9]) = 01b.

2. PWM compare register must greater than (ASYMCMP + 1) if ASYMMODE0(EPWM_ASYMCTL[8:9]) = 10b.

3. That ASYMCMP equals PERIOD or ASYMCMP equals 00H is not allowed

4. That EPWM_CMPDAT0 equals PERIOD or EPWM_CMPDAT0 equals 00H is not allowed.

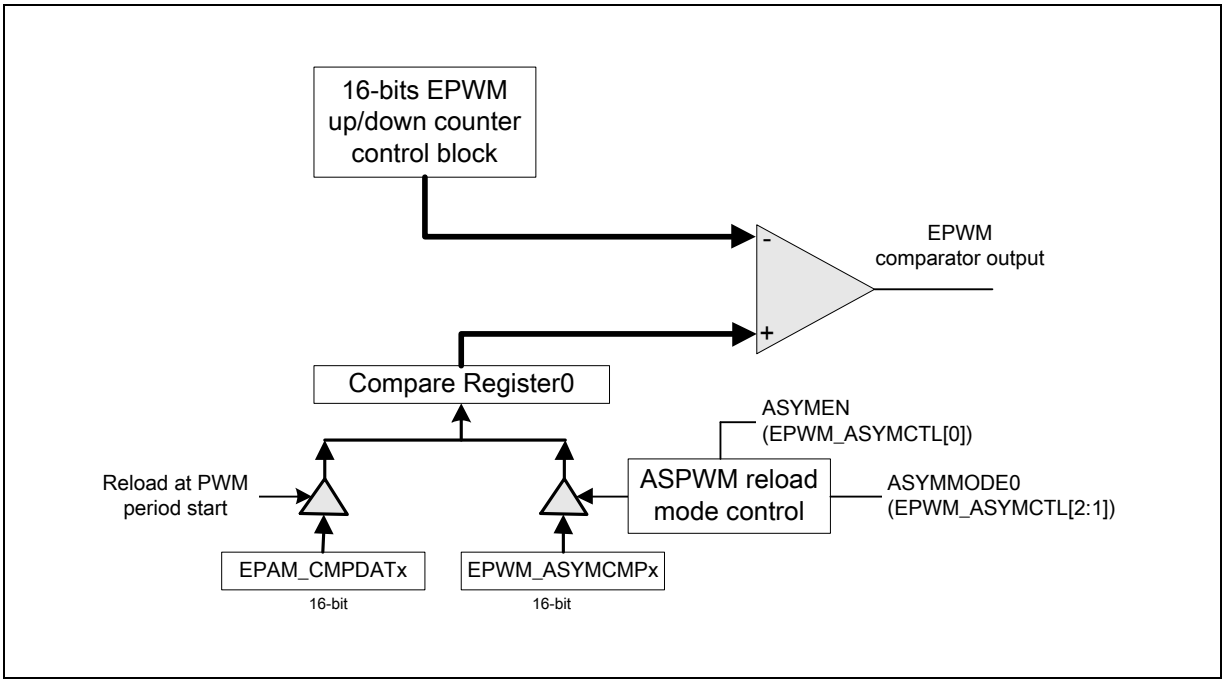


Figure 6.22-17 Asymmetric PWM Architecture

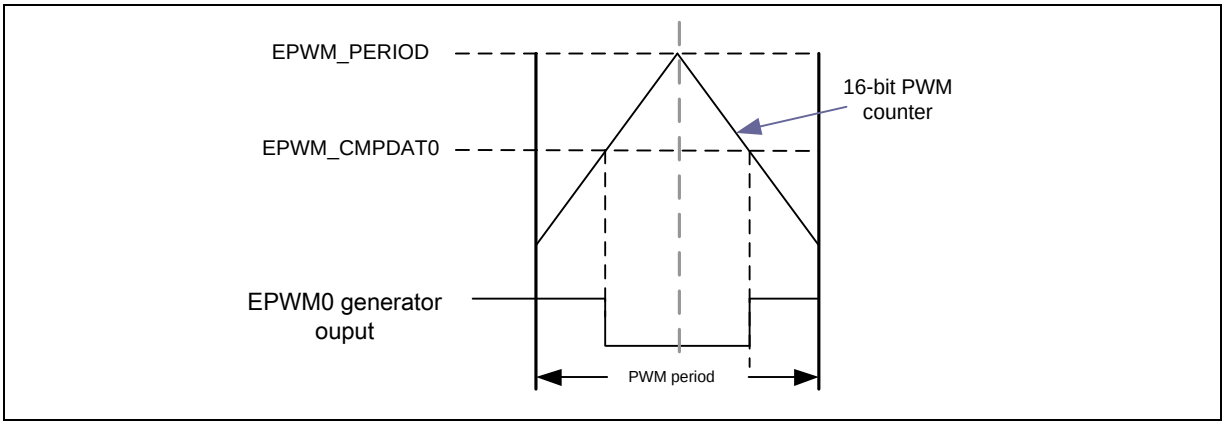


Figure 6.22-18 Symmetric PWM Output in Centre Aligned Mode

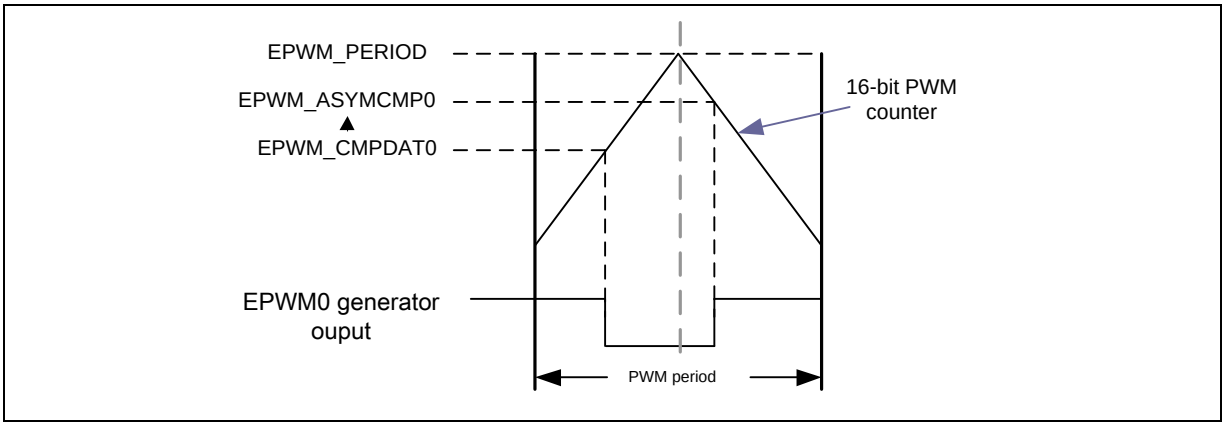


Figure 6.22-19 Asymmetric PWM Output for ASPRLDM=00b

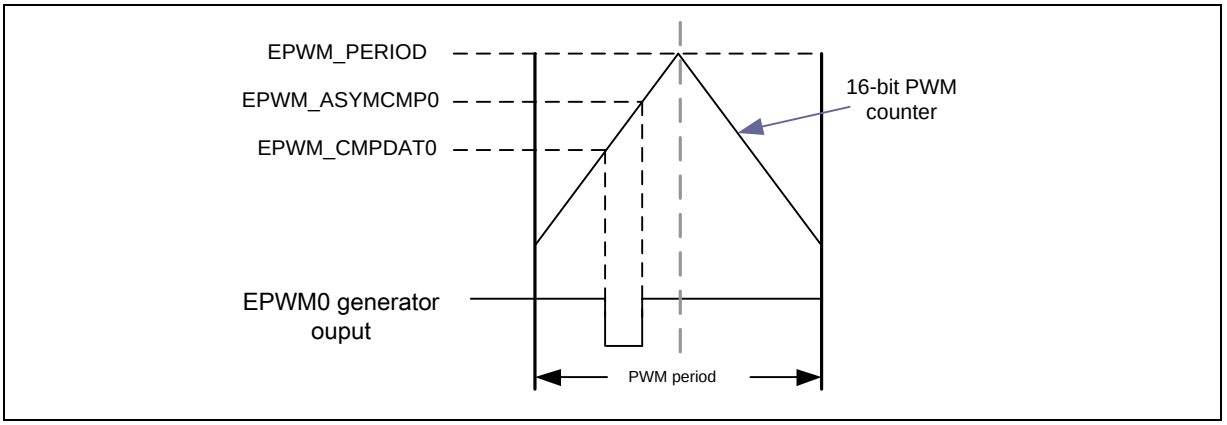


Figure 6.22-20 Asymmetric PWM Output for ASPRLDM=01b

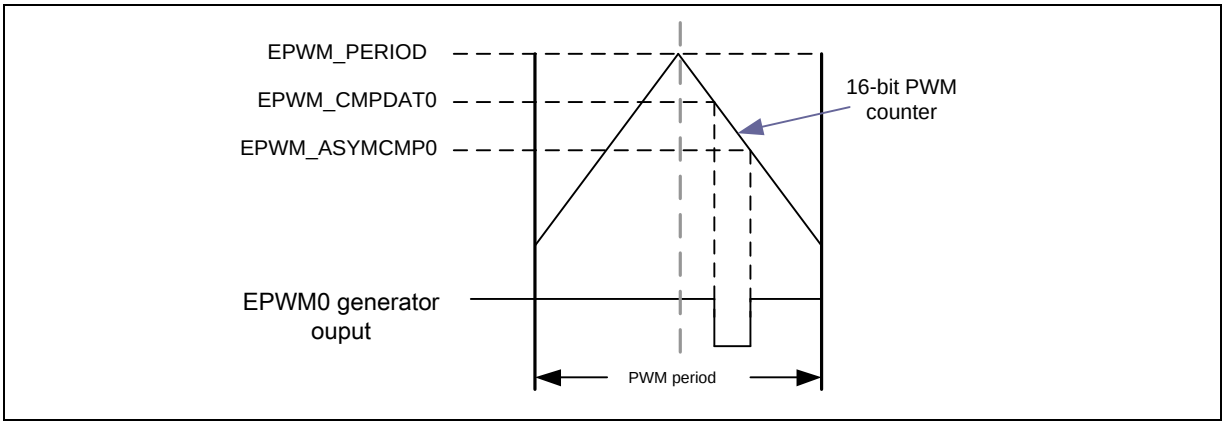


Figure 6.22-21 Asymmetric PWM Output for ASPRLDM=10b

6.22.10 Interrupt Architecture of Enhanced PWM

There are six interrupt sources for each PWM unit, which are PWM period flag (PIF), Brake0 flag (BRKIF0), Brake1 flag (BRKIF1), PWM_CH0 edge flag, PWM_CH2 edge flag and PWM_CH4 edge flag. The bit BRKIEN (EPWM_CTL[5]) control the brake interrupt enable; the bit PWMIEN (EPWM_CTL[4]) controls the PWM periodic interrupt enable; the bit EDGEIEN0 (EPWM_EINTCTL[0]) controls the PWM0 edge interrupt enable; the bit EDGEIEN2 (EPWM_EINTCTL[1]) controls the enable of PWM2 edge interrupt; the bit EDGEIEN4 (EPWM_EINTCTL[2]) controls the enable of PWM4 edge interrupt. **Note that all the interrupt flags are set by hardware and must be cleared by software writing 1 to flags.**

Figure 6.22-22demonstrates the architecture of enhanced PWM interrupts.

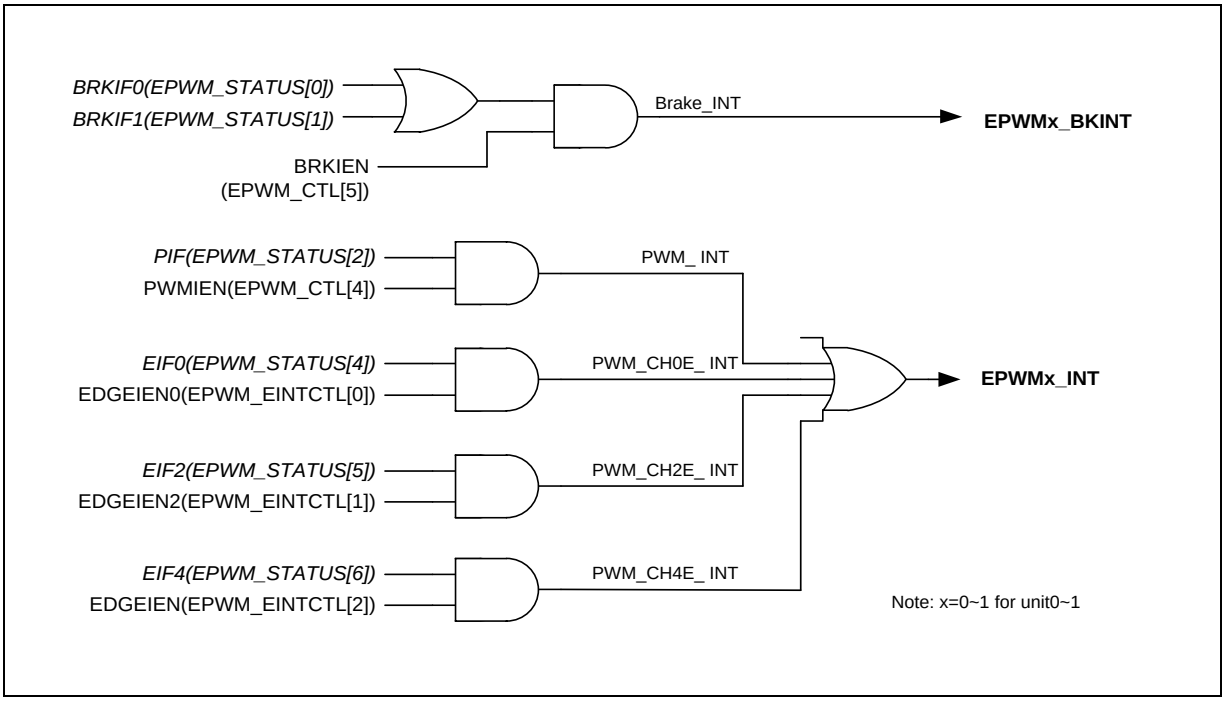


Figure 6.22-22 Enhanced PWM Interrupt Architecture

6.22.11 Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
EPWM Base Address: **EPWMn_BA = 0x4005_C000 + n * 0x1000** **n = 0,1**				
EPWM_CTL	EPWMn_BA+0x00	R/W	PWM Control Register	0x0000_0000
EPWM_STATUS	EPWMn_BA+0x04	R/W	PWM Status Register	0x0000_0000
EPWM_PERIOD	EPWMn_BA+0x08	R/W	PWM Period Register	0x0000_0000
EPWM_CMPDAT0	EPWMn_BA+0x0C	R/W	EPWM_CMPDAT0 Duty Register	0x0000_0000
EPWM_CMPDAT2	EPWMn_BA+0x10	R/W	EPWM_CMPDAT2 Duty Register	0x0000_0000
EPWM_CMPDAT4	EPWMn_BA+0x14	R/W	EPWM_CMPDAT4 Duty Register	0x0000_0000
EPWM_MSKEN	EPWMn_BA+0x18	R/W	PWM Mask Mode Enable Control Register	0x0000_0000
EPWM_MSK	EPWMn_BA+0x1C	R/W	PWM Mask Mode Data Register	0x0000_0000
EPWM_ASYMCMP0	EPWMn_BA+0x20	R/W	Asymmetric EPWM_CMPDAT0 Duty Register	0x0000_0000
EPWM_ASYMCMP2	EPWMn_BA+0x24	R/W	Asymmetric EPWM_CMPDAT2 Duty Register	0x0000_0000
EPWM_ASYMCMP4	EPWMn_BA+0x28	R/W	Asymmetric EPWM_CMPDAT4 Duty Register	0x0000_0000
EPWM_DTCTL	EPWMn_BA+0x2C	R/W	PWM Dead-time Control Register	0x0000_0000
EPWM_BRKOUT	EPWMn_BA+0x30	R/W	PWM Brake Output	0x0000_0000
EPWM_NPCTL	EPWMn_BA+0x34	R/W	PWM Negative Polarity Control	0x0000_0000
EPWM_ASYMCTL	EPWMn_BA+0x38	R/W	Asymmetric PWM Control Register	0x0000_0000
EPWM_PERIODCNT	EPWMn_BA+0x3C	R/W	PIF Compared Counter	0x0000_0000
EPWM_EINTCTL	EPWMn_BA+0x40	R/W	PWM Edge Interrupt Control Register	0x0000_0000
EPWM_OUTEN0	EPWMn_BA+0x44	R/W	PWM Output Enable Control Register	0x0000_0000

6.22.12 Register Description

PWM Period Register (EPWM_PERIOD)

Register	Offset	R/W	Description	Reset Value
EPWM_PERIOD	EPWMn_BA+0x08	R/W	PWM Period Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PERIOD							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PERIOD							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**PERIOD**	**PWM Period Register** **Edge aligned**: Period = (PERIOD(EPWM_PERIOD[0:15]) + 1) * EPWMx_CLK period * pre-scalar. Duty = (CMP(EPWM_CMPDATn[0:15]) +1)* EPWMx_CLK period * pre-scalar. **Centre aligned**: Period = (PERIOD(EPWM_PERIOD[0:15]) * 2) * EPWMx_CLK period * pre-scalar. Duty = (CMP(EPWM_CMPDATn[0:15]) *2 + 1) * EPWMx_CLK period * pre-scalar.

PWM Duty Register (EPWM_CMPDAT0/2/4)

Register	Offset	R/W	Description	Reset Value
EPWM_CMPDAT0	EPWMn_BA+0x0C	R/W	EPWM_CMPDAT0 Duty Register	0x0000_0000
EPWM_CMPDAT2	EPWMn_BA+0x10	R/W	EPWM_CMPDAT2 Duty Register	0x0000_0000
EPWM_CMPDAT4	EPWMn_BA+0x14	R/W	EPWM_CMPDAT4 Duty Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CMP							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CMP							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**CMP**	**PWM Duty Register** **Edge aligned**: Period = (PERIOD(EPWM_PERIOD[0:15]) + 1) * EPWMx_CLK period * pre-scalar. Duty = (CMP(EPWM_CMPDATn[0:15]) +1)* EPWMx_CLK period * pre-scalar. **Centre aligned**: Period = (PERIOD(EPWM_PERIOD[0:15]) * 2) * EPWMx_CLK period * pre-scalar. Duty = (CMP(EPWM_CMPDATn[0:15]) *2 + 1) * EPWMx_CLK period * pre-scalar.

PWM Mask Enable Control Register (EPWM_MSKEN)

Register	Offset	R/W	Description	Reset Value
EPWM_MSKEN	EPWMn_BA+0x18	R/W	PWM Mask Mode Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**MSKEN**					

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5:0]	**MSKEN**	**PWM Mask Enable Bits** The PWM generator signal will be masked when this bit is enabled. The corresponding EPWM_CHn channel will be output with EPWM_MSK[n] data. 0 = PWM generator signal is output to next stage. 1 = PWM generator signal is masked and EPWM_MSK[n] is output to next stage. **Note:** n = 0~5.

PWM Mask Data Register (EPWM_MSK)

Register	Offset	R/W	Description	Reset Value
EPWM_MSK	EPWMn_BA+0x1C	R/W	PWM Mask Mode Data Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**MSKDAT**					

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5:0]	**MSKDAT**	**PWM Mask Data Bit** This data bit control the state of EPWM_CHn output pin, if the corresponding EPWM_MSKEN[n] = 1. 0 = Output logic low to EPWM_CHn. 1 = Output logic high to EPWM_CHn. **Note:** n = 0~5.

PWM Control Register (EPWM_CTL)

Register	Offset	R/W	Description	Reset Value
EPWM_CTL	EPWMn_BA+0x00	R/W	PWM Control Register	0x0000_0000

31	30	29	28	27	26	25	24
CTRLD	Reserved	BRK1NFDIS	BRK0NFDIS	LVDBKEN	CPO2BKEN	CPO1BKEN	CPO0BKEN
23	22	21	20	19	18	17	16
BRK1NFSEL		BRK0NFSEL		BRK1SEL		BRKP1EN	BRKP0EN
15	14	13	12	11	10	9	8
BRKP1INV	BRKP0INV	GROUPEN	CNTTYPE	CNTCLR	Reserved	PINV	INTTYPE
7	6	5	4	3	2	1	0
CNTEN	LOAD	BRKIEN	PWMIEN	CLKDIV		MODE	

Bits	Description	
[31]	**CTRLD**	**Center Reload Mode Enable Bit** 0 = EPWM reload compare register at the period point of PWM counter. 1 = EPWM reload compare register at the center point of PWM counter. This bit only work when EPWM operation at center aligned mode.
[30]	**Reserved**	Reserved.
[29]	**BRK1NFDIS**	**PWM Brake 1 Noise Filter Disable Bit** 0 = Noise filter of PWM Brake 1 Enabled. 1 = Noise filter of PWM Brake 1 Disabled.
[28]	**BRK0NFDIS**	**PWM Brake 0 Noise Filter Disable Bit** 0 = Noise filter of PWM Brake 0 Enabled. 1 = Noise filter of PWM Brake 0 Disabled.
[27]	**LVDBKEN**	**Low-Level Detection Trigger PWM Brake Function 1 Enable Bit** 0 = Brake Function 1 triggered by Low-level detection Disabled. 1 = Brake Function 1 triggered by Low-level detection Enabled.
[26]	**CPO2BKEN**	**CPO2 Digital Output As Brake 0 Source Enable Bit** 0 = CPO2 as one brake source in Brake 0 Disabled. 1 = CPO2 as one brake source in Brake 0 Enabled.
[25]	**CPO1BKEN**	**CPO1 Digital Output As Brake 0 Source Enable Bit** 0 = CPO1 as one brake source in Brake 0 Disabled. 1 = CPO1 as one brake source in Brake 0 Enabled.
[24]	**CPO0BKEN**	**CPO0 Digital Output As Brake0 Source Enable Bit** 0 = CPO0 as one brake source in Brake 0 Disabled. 1 = CPO0 as one brake source in Brake 0 Enabled.

Bits	Description	
[23:22]	**BRK1NFSEL**	**Brake 1 (BKPx1 Pin) Edge Detector Filter Clock Selection** 00 = Filter clock = HCLK. 01 =Filter clock = HCLK/2. 10 = Filter clock = HCLK/4. 11 = Filter clock = HCLK/16.
[21:20]	**BRK0NFSEL**	**Brake 0 (BKPx0 Pin) Edge Detector Filter Clock Selection** 00 = Filter clock = HCLK. 01 = Filter clock = HCLK/2. 10 = Filter clock = HCLK/4. 11 = Filter clock = HCLK/16.
[19:18]	**BRK1SEL**	**Brake Function 1 Source Selection** 00 = From external pin BKPx1 (x=0~1 for unit0~1). 01 = From analog comparator 0 output (CPO0). 10 = From analog comparator 1 output (CPO1). 11 = From analog comparator 2 output (CPO2).
[17]	**BRKP1EN**	**BKPx1 Pin Trigger Brake Function Enable Bit** 0 = PWMx Brake Function 1 Disabled. 1 = PWMx Brake Function 1 Enabled. **Note:** x=0~1 for PWM unit0~1.
[16]	**BRKP0EN**	**BKPx0 Pin Trigger Brake Function0 Enable Bit** 0 = PWMx Brake Function 0 Disabled. 1 = PWMx Brake Function 0 Enabled. **Note:** x=0~1 for PWM unit0~1.
[15]	**BRKP1INV**	**Inverse BKP1 State** 0 = The state of pin BKPx1 is passed to the negative edge detector. 1 = The inversed state of pin BKPx1 is passed to the negative edge detector.
[14]	**BRKP0INV**	**Inverse BKP0 State** 0 = The state of pin BKPx0 is passed to the negative edge detector. 1 = The inversed state of pin BKPx0 is passed to the negative edge detector.
[13]	**GROUPEN**	**Group Bit** 0 = The signals timing of EPWM_CMPDAT0, EPWM_CMPDAT2 and EPWM_CMPDAT4 are independent. 1 = Unify the signals timing of EPWM_CMPDAT0, EPWM_CMPDAT2 and EPWM_CMPDAT4 in the same phase which is controlled by EPWM_CMPDAT0.
[12]	**CNTTYPE**	**PWM Aligned Type Selection** 0 = Edge-aligned type. 1 = Centre-aligned type.
[11]	**CNTCLR**	**Clear PWM Counter Control** 1 = Clear 16-bit PWM counter to 000H. **Note:** It is automatically cleared by hardware.
[10]	**Reserved**	Reserved.

Bits	Description	
[9]	**PINV**	**Inverse PWM Comparator Output** When PINV is set to high the PWM comparator output signals will be inversed, therefore the PWM Duty (in percentage) is changed to (1-Duty) before PINV is set to high. 0 = Not inverse PWM comparator output. 1 = Inverse PWM comparator output.
[8]	**INTTYPE**	**PWM Interrupt Type Selection** 0 = PIF will be set if PWM counter underflow. 1 = PIF will be set if PWM counter matches EPWM_PERIOD register. **Note:** This bit is effective when PWM in central align mode only.
[7]	**CNTEN**	**Start CNTEN Control** 0 = The PWM stops running. 1 = The PWM counter starts running.
[6]	**LOAD**	**Re-Load PWM Period Registers (EPWM_PERIOD) And PWM Compare Registers (EPWM_CMPDAT0~4) Control** 0 = No action if written with 0. The value of PWM period register (EPWM_PERIOD) and PWM compare registers (EPWM_CMPDAT0~EPWM_CMPDAT4) are not loaded to PWM counter and Comparator registers. 1 = Hardware will update the value of PWM period register (EPWM_PERIOD) and PWM compare registers (EPWM_CMPDAT0~EPWM_CMPDAT4) to PWM Counter and Comparator register at the time of PWM Counter matches PERIOD in edge and central aligned modes or at the time of PWM Counter down counts with underflow in central aligned mode. **Note:** This bit is software write, hardware clear and always read zero.
[5]	**BRKIEN**	**Brake0 And Brak1 Interrupt Enable Bit** 0 = Disabling flags BRKIF0 and BRKIF1 to trigger PWM interrupt. 1 = Enabling flags BRKIF0 and BRKIF1 can trigger PWM interrupt.
[4]	**PWMIEN**	**PWM Interrupt Enable Bit** 0 = Disabling flag PIF to trigger PWM interrupt. 1 = Enabling flag PIF can trigger PWM interrupt.
[3:2]	**CLKDIV**	**PWM Clock Pre-Divider Selection** 00 = PWM clock = EPWM_CLK. 01 = PWM clock = EPWM_CLK/2. 10 = PWM clock = EPWM_CLK/4. 11 = PWM clock = EPWM_CLK/16.
[1:0]	**MODE**	**PWM Mode Selection** 00 = Independent mode. 01 = Pair/Complementary mode. 10 = Synchronized mode. 11 = Reserved.

PWM Status Register (EPWM_STATUS)

Register	Offset	R/W	Description	Reset Value
EPWM_STATUS	EPWMn_BA+0x04	R/W	PWM Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						BRK1STS	BRK0STS
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							BRK0LOCK
7	6	5	4	3	2	1	0
Reserved	EIF4	EIF2	EIF0	Reserved	PIF	BRKIF1	BRKIF0

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25]	**BRK1STS**	**Brake 1 Status (Read Only)** 0 = PWM had been out of Brake 1 state. 1 = PWM is in Brake 1 state.
[24]	**BRK0STS**	**Brake 0 Status (Read Only)** 0 = PWM had been out of Brake 0 state. 1 = PWM is in Brake 0 state.
[23:9]	**Reserved**	Reserved.
[8]	**BRK0LOCK**	**PWM Brake0 Locked** 0 = Brake 0 state is released. 1 = When PWM Brake detects a falling signal at BKP0, this flag will be set to high to indicate the Brake0 state is locked. **Note:** This bit must be cleared by writing 1 to it.
[7]	**Reserved**	Reserved.
[6]	**EIF4**	**PWMx_CH4 Edge Flag** 0 = The PWMx_CH4 doesn't toggle.. 1 = Hardware will set this flag to high at the time of PWMx_CH4 rising or falling. If EDGEIEN4(EPWM_EINTCTL[10]) = 0, this bit is set when PWMx_CH4 falling is detected. If EDGEIEN4(EPWM_EINTCTL[10]) = 1, this bit is set when PWMx_CH4 rising is detected.. **Note:** This bit must be cleared by writing 1 to it.
[5]	**EIF2**	**PWMx_CH2 Edge Flag** 0 = The PWMx_CH2 doesn't toggle. 1 = Hardware will set this flag to high at the time of PWMx_CH2 rising or falling. If EDGEIEN2(EPWM_EINTCTL[9]) = 0, this bit is set when PWMx_CH2 falling is detected. If EDGEIEN2(EPWM_EINTCTL[9])= 1, this bit is set when PWMx_CH2 rising is detected.. **Note:** This bit must be cleared by writing 1 to it.

[4]	**EIF0**	**PWMx_CH0 Edge Flag** 0 = The PWMx_CH0 doesn't toggle. 1 = Hardware will set this flag to high at the time of PWMx_CH0 rising or falling. If EDGEIEN0(EPWM_EINTCTL[8]) = 0, this bit is set when PWMx_CH0 falling is detected. If EDGEIEN0(EPWM_EINTCTL[8]) = 1, this bit is set when PWMx_CH0 rising is detected.. **Note:** This bit must be cleared by writing 1 to it.
[3]	**Reserved**	Reserved.
[2]	**PIF**	**PWM Period Flag** 0 = PWM Counter has not up counted to the value of PERIOD or down counted with underflow.. 1 = Hardware will set this flag to high at the time of PWM Counter matches PERIOD in edge and Centre aligned modes or at the time of PWM Counter down counts with underflow in Centre aligned mode. **Note:** This bit must be cleared by writing 1 to it.
[1]	**BRKIF1**	**PWM Brake1 Flag** 0 = PWM Brake 1 is able to poll falling signal at BKP1 and has not recognized any one. 1 = When PWM Brake 1 detects a falling signal at pin BKP1, this flag will be set to high. **Note:** This bit must be cleared by writing 1 to it.
[0]	**BRKIF0**	**PWM Brake0 Flag** 0 = PWM Brake 0 is able to poll falling signal at BKP0 and has not recognized any one. 1 = When PWM Brake 0 detects a falling signal at BKP0, this flag will be set to high. **Note:** This bit must be cleared by writing 1 to it.

Asymmetric PWM Control Register (EPWM_ASYMCTL)

Register	Offset	R/W	Description	Reset Value
EPWM_ASYMCTL	EPWMn_BA+0x38	R/W	Asymmetric PWM Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						**ASYMMODE4**	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						**ASYMMODE2**	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**ASYMMODE0**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**ASYMEN**

Bits	Description	
[31:26]	**Reserved**	Reserved.

Bits	Description	
[25:24]	**ASYMMODE4**	**Asymmetric PWMx_CH4 Reload Mode Setting** 00 = 1. PWM compare register 4 is reload CMP (EPWM_CMPDAT4[15:0]) at PWM cycle start. 2. PWM module compares the PWM counter (PWMCNT) with PWM compare register 4 in up-counting cycle 3. PWM compare output toggles at compare match, in the meanwhile, PWM compare register 4 is reloaded with CMP (EPWM_ASYMCMP4[15:0]). 4. PWM module compares the PWM counter (PWMCNT) with PWM compare register 4 in down-counting cycle. 01 = 1. PWM compare register 4 is reload CMP (EPWM_CMPDAT4[15:0]) at PWM cycle start. 2. PWM module compares the PWM counter (PWMCNT) with PWM compare register 4 in up-counting cycle 3. PWM compare output toggles at compare match, in the meanwhile, PWM compare register 0 is reloaded with CMP (EPWM_ASYMCMP4[15:0]). 4. PWM module compares the PWM counter (PWMCNT) with PWM compare register 4 in up-counting cycle. 5. EPWM_CMPDAT4 must be less than EPWM_ASYMCMP4. 10 = 1. PWM compare register 4 is reload CMP (EPWM_CMPDAT4[15:0]) at PWM cycle start. 2. PWM module compares the PWM counter (PWMCNT) with PWM compare register 4 in down-counting cycle 3. PWM compare output toggles at compare match, in the meanwhile, PWM compare register 4 is reloaded with CMP (EPWM_ASYMCMP4[15:0]). 4. PWM module compares the PWM counter (PWMCNT) with PWM compare register 4 in down-counting cycle. 5. EPWM_CMPDAT4 must be greater than EPWM_ASYMCMP4. 11= Reserved. **Note1:** x=0~1 for PWM unit 0~1. **Note2:** This bit field is available only when ASYMEN=1. **Note3:** The setting will be effected at the condition of LOAD=1 and from the start of next PWM cycle.
[23:18]	**Reserved**	Reserved.

Bits	Description	
[17:16]	**ASYMMODE2**	**Asymmetric PWMx_CH2 Reload Mode Setting** 00 = 1. PWM compare register 2 is reload CMP (EPWM_CMPDAT2[15:0]) at PWM cycle start. 2. PWM module compares the PWM counter (PWMCNT) with PWM compare register 2 in up-counting cycle 3. PWM compare output toggles at compare match, in the meanwhile, PWM compare register 2 is reloaded with CMP (EPWM_ASYMCMP2[15:0]). 4. PWM module compares the PWM counter (PWMCNT) with PWM compare register 2 in down-counting cycle. 01 = 1. PWM compare register 2 is reload CMP (EPWM_CMPDAT2[15:0]) at PWM cycle start. 2. PWM module compares the PWM counter (PWMCNT) with PWM compare register 2 in up-counting cycle 3. PWM compare output toggles at compare match, in the meanwhile, PWM compare register2 is reloaded with CMP (EPWM_ASYMCMP2[15:0]). 4. PWM module compares the PWM counter (PWMCNT) with PWM compare register 2 in up-counting cycle. 5. EPWM_CMPDAT2 must be less than EPWM_ASYMCMP2 10 = 1. PWM compare register 2 is reload PWM_Duty (PWMx2[15:0]) at PWM cycle start. 2. PWM module compares the PWM counter (PWMCNT) with PWM compare register 2 in down-counting cycle 3. PWM compare output toggles at compare match, in the meanwhile, PWM compare register 2 is reloaded with CMP (EPWM_ASYMCMP2[15:0]). 4. PWM module compares the PWM counter (PWMCNT) with PWM compare register 2 in down-counting cycle. 5. EPWM_CMPDAT2 must be greater than EPWM_ASYMCMP2. 11 = Reserved. **Note1:** x=0~1 for PWM unit 0~1. **Note2:** This bit field is available only when ASYMEN=1. **Note3:** The setting will be effected at the condition of LOAD=1 and from the start of next PWM cycle.
[15:10]	**Reserved**	Reserved.

Bits	Description	
[9:8]	**ASYMMODE0**	**Asymmetric PWMx_CH0 Reload Mode Setting** 00 = 1. PWM compare register 0 is reload CMP (EPWM_CMPDAT0[15:0]) at PWM cycle start. 2. PWM module compares the PWM counter (PWMCNT) with PWM compare register 0 in up-counting cycle 3. PWM compare output toggles at compare match, in the meanwhile, PWM compare register 0 is reloaded with CMP (EPWM_ASYMCMP0[15:0]). 4. PWM module compares the PWM counter (PWMCNT) with PWM compare register 0 in down-counting cycle. 01 = 1. PWM compare register 0 is reload CMP (EPWM_CMPDAT0[15:0]) at PWM cycle start. 2. PWM module compares the PWM counter (PWMCNT) with PWM compare register 0 in up-counting cycle 3. PWM compare output toggles at compare match, in the meanwhile, PWM compare register 0 is reloaded with CMP (EPWM_ASYMCMP0[15:0]). 4. PWM module compares the PWM counter (PWMCNT) with PWM compare register 0 in up-counting cycle. 5. EPWM_CMPDAT0 must be less than EPWM_ASYMCMP0 10 = 1. PWM compare register 0 is reload CMP (EPWM_CMPDAT0[15:0]) at PWM cycle start. 2. PWM module compares the PWM counter (PWMCNT) with PWM compare register 0 in down-counting cycle 3. PWM compare output toggles at compare match, in the meanwhile, PWM compare register 0 is reloaded with CMP (EPWM_ASYMCMP2[15:0]). 4. PWM module compares the PWM counter (PWMCNT) with PWM compare register 0 in down-counting cycle. 5. EPWM_CMPDAT0 must be greater than EPWM_ASYMCMP0. 11 = Reserved. **Note1:** x=0~1 for PWM unit 0~1. **Note2:** This bit field is available only when ASYMEN=1. **Note3:** The setting will be effected at the condition of LOAD=1 and from the start of next PWM cycle.
[7:1]	**Reserved**	Reserved.
[0]	**ASYMEN**	**Asymmetric PWM Enable Bit** 0 = Asymmetric PWM function Disabled. 1 = Asymmetric PWM function Enabled. **Note:** This control bit is only valid when PWM module is set in Centre-aligned mode.

Asymmetric PWM Register (EPWM_ASYMCMP0/2/4)

Register	Offset	R/W	Description	Reset Value
EPWM_ASYMCMP0	EPWMn_BA+0x20	R/W	Asymmetric EPWM_CMPDAT0 Duty Register	0x0000_0000
EPWM_ASYMCMP2	EPWMn_BA+0x24	R/W	Asymmetric EPWM_CMPDAT2 Duty Register	0x0000_0000
EPWM_ASYMCMP4	EPWMn_BA+0x28	R/W	Asymmetric EPWM_CMPDAT4 Duty Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CMP							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CMP							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**CMP**	**Asymmetric PWM Duty Register** When the asymmetric PWM function is enabled, this 16-bit field determines the second time compared value after PWM counter has matched with EPWM_CMPDATx in the first half PWM cycle.

PWM Dead-time Control Register (EPWM_DTCTL)

Register	Offset	R/W	Description	Reset Value
EPWM_DTCTL	EPWMn_BA+0x2C	R/W	PWM Dead-time Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved					**DTEN4**	**DTEN2**	**DTEN0**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**DTCNT**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DTCNT							

Bits	Description	
[31:19]	**Reserved**	Reserved.
[18]	**DTEN4**	**Dead-Time Insertion Enable Control for PWMx Pair (PWM_CH4, PWM_CH5)** Dead-time insertion is only active when this pair of complementary PWM is enabled. If dead-time insertion is inactive, the outputs of pin pair are complementary without any delay. 0 = Dead-time insertion Disabled on the pin pair (PWM_CH4, PWM_CH5). 1 = Dead-time insertion Enabled on the pin pair (PWM_CH4, PWM_CH5). **Note:** x=0~1 for PWM unit0~1.
[17]	**DTEN2**	**Dead-Time Insertion Enable Control for PWMx Pair (PWM_CH2, PWM_CH3)** Dead-time insertion is only active when this pair of complementary PWM is enabled. If dead-time insertion is inactive, the outputs of pin pair are complementary without any delay. 0 = Dead-time insertion Disabled on the pin pair (PWM_CH2, PWM_CH3). 1 = Dead-time insertion Enabled on the pin pair (PWM_CH2, PWM_CH3). **Note:** x=0~1 for PWM unit0~1.
[16]	**DTEN0**	**Dead-Time Insertion Enable Control for PWMx Pair (PWM_CH0, PWM_CH1)** Dead-time insertion is only active when this pair of complementary PWM is enabled. If dead-time insertion is inactive, the outputs of pin pair are complementary without any delay. 0 = Dead-time insertion Disabled on the pin pair (PWM_CH0, PWM_CH1). 1 = Dead-time insertion Enabled on the pin pair (PWM_CH0, PWM_CH1). **Note:** x=0~1 for PWM unit0~1.
[15:11]	**Reserved**	Reserved.
[10:0]	**DTCNT**	**Dead-Time Counter** The dead-time can be calculated from the following formula: Dead-time = EPWM_CLK period * (DTCNT.[10:0]+1).

PWM Brake Output (EPWM_BRKOUT)

Register	Offset	R/W	Description	Reset Value
EPWM_BRKOUT	EPWMn_BA+0x30	R/W	PWM Brake Output	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**BRKOUT**					

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5:0]	**BRKOUT**	**PWM Brake Output** When PWM Brake is asserted, the PWM_CH0~5 output state before polarity control will follow bit0~5 setting, respectively. 0 = The PWM_CHn output before polarity control is low when Brake is asserted. 1 = The PWM_CHn output before polarity control is high when Brake is asserted. **Note:** n = 0~5.

PWM Negative Polarity Control (EPWM_NPCTL)

Register	Offset	R/W	Description	Reset Value
EPWM_NPCTL	EPWMn_BA+0x34	R/W	PWM Negative Polarity Control	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**NEGPOLAR**					

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5:0]	**NEGPOLAR**	**PWM Negative Polarity Control** The register bit controls polarity/active state of real PWM output. 0 = PWM_CHn output is active high. 1 = PWM_CHn output is active low. **Note:** n = 0~5.

PIF Compared Counter (EPWM_PERIODCNT)

Register	Offset	R/W	Description	Reset Value
EPWM_PERIODCNT	EPWMn_BA+0x3C	R/W	PIF Compared Counter	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				PERIODCNT			

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3:0]	**PERIODCNT**	**PIF Compared Counter** The register sets the count number which defines how many times of PWM period occurs to set bit PIF to request the PWM period interrupt. PIF will be set in every (1 + PERIODCNT[3:0]) times of PWM period or center point defined by INTTYPE when EPWM_CTL [8] occurred.

PWM Edge Interrupt Control Register (EPWM_EINTCTL)

Register	Offset	R/W	Description	Reset Value
EPWM_EINTCTL	EPWMn_BA+0x40	R/W	PWM Edge Interrupt Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**EINTTYPE4**	**EINTTYPE2**	**EINTTYPE0**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**EDGEIEN4**	**EDGEIEN2**	**EDGEIEN0**

Bits	Description	
[31:11]	**Reserved**	Reserved.
[10]	**EINTTYPE4**	**PWMx4 Edge Interrupt Type** 0 = EIF4 will be set if falling edge is detected at PWMx4. 1 = EIF4 will be set if rising edge is detected at PWMx4.
[9]	**EINTTYPE2**	**PWMx2 Edge Interrupt Type** 0 = EIF2 will be set if falling edge is detected at PWMx2. 1 = EIF2 will be set if rising edge is detected at PWMx2.
[8]	**EINTTYPE0**	**PWMx0 Edge Interrupt Type** 0 = EIF0 will be set if falling edge is detected at PWMx0. 1 = EIF0 will be set if rising edge is detected at PWMx0.
[7:3]	**Reserved**	Reserved.
[2]	**EDGEIEN4**	**PWMx4 Edge Interrupt Enable Bit** 0 = Disable flag EIF4 to trigger PWM interrupt. 1 = Enabling flag EIF4 can trigger PWM interrupt.
[1]	**EDGEIEN2**	**PWMx2 Edge Interrupt Enable Bit** 0 = Disabling flag EIF2 can trigger PWM interrupt. 1 = Enabling flag EIF2 can trigger PWM interrupt.
[0]	**EDGEIEN0**	**PWMx0 Edge Interrupt Enable Bit** 0 = Disabling flag EIF0 to trigger PWM interrupt. 1 = Enabling flag EIF0 can trigger PWM interrupt.

PWM Output Enable Control Register (EPWM_OUTEN0)

Register	Offset	R/W	Description	Reset Value
EPWM_OUTEN0	EPWMn_BA+0x44	R/W	PWM Output Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**ODDOUTEN**	**EVENOUTEN**

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**ODDOUTEN**	**PWM Odd Ports Output Enable Bit** 0 = PWM odd ports output Disabled (PWM even ports at tri-state). 1 = PWM odd ports output Enabled.
[0]	**EVENOUTEN**	**PWM Even Ports Output Enable Bit** 0 = PWM even ports output Disabled (PWM even ports at tri-state). 1 = PWM even ports output Enabled.

6.23 Quadrature Encoder Interface (QEI)

6.23.1 Overview

There are two QEI controllers in this device. The Quadrature Encoder Interface (QEI) decodes speed of rotation and motion sensor information. It can be used in any application that uses a quadrature encoder for feedback.

6.23.2 Features

- Up to two QEI controllers, QEI0 and QEI1.
- Two QEI phase inputs, QEA and QEB; One Index input.
- One QEI control register (QEI_CTL) and one QEI Status Register (QEI_STATUS)
- Four Quadrature encoder pulse counter operation modes:
  - Mode0: x4 free-counting mode
  - Mode1: x2 free-counting mode
  - Mode2: x4 compare-counting mode
  - Mode3: x2 compare-counting mode
- Encoder Pulse Width measurement mode

6.23.3 QEI Architecture

The QEI controller inputs, QEA and QEB, accept the outputs from a quadrature encoded source, such as incremental optical shaft encoder. Two channels, A and B, nominally 90 degrees out of phase, are required. A quadrature encoder usually provides an index signal (to pin IDX) which can be used to indicate an absolute position. There is a noise filter and polarity control for each signal before QEI control unit.

The following figures illustrate the architecture of Quadrature Encoder Interface Controller.

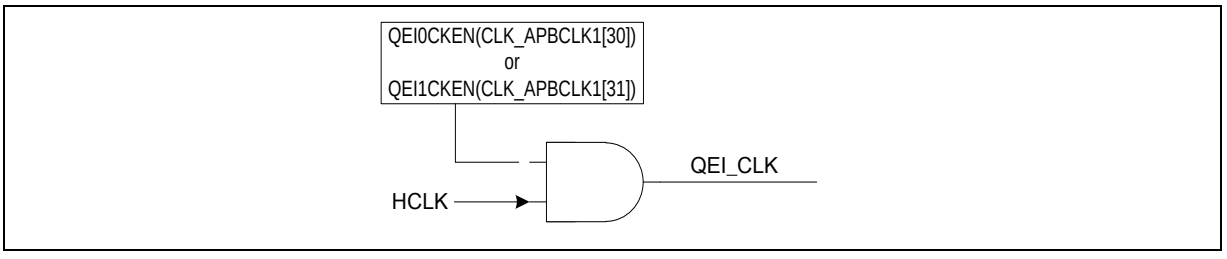


Figure 6.23-1 QEI Clock Source Control

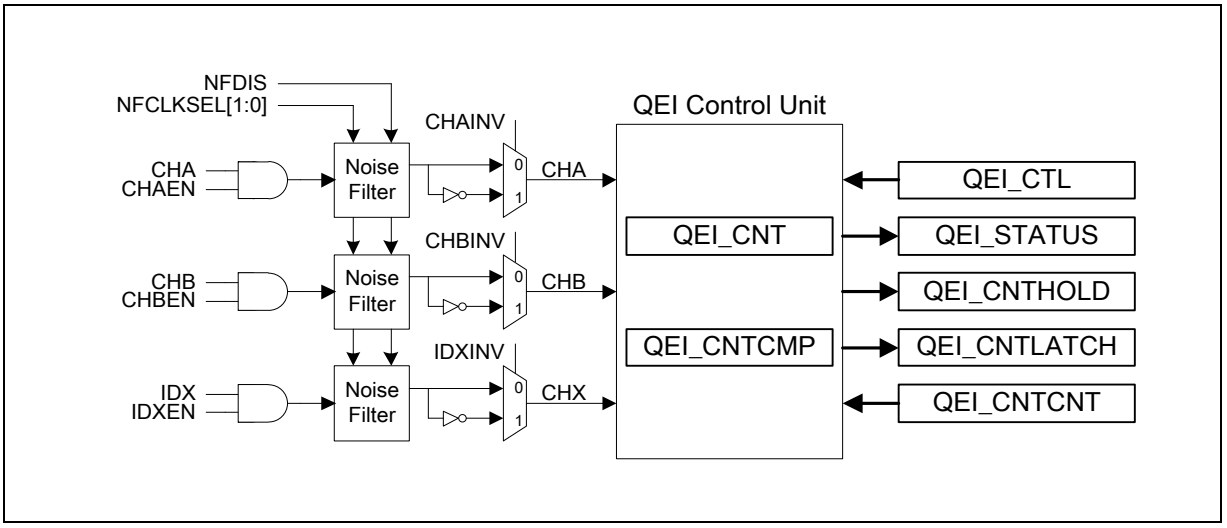


Figure 6.23-2 QEI Block Diagram

The QEI control logic detects the relation of phase lead/lag between the filtered signals CHA and CHB and CHX to produce direction indication bit (DIRF) and clock (QCLK) to control pulse counter. The comparator/reload logic compares the pulse counter and maximum count and control the function of reloading pulse counter in compare-counting mode. In Free-counting mode the pulse counter (QEI_CNT) will count until the 0xFFFF_FFFF value; while in Compare-counting mode the pulse counter will counts until the QEI_CNTMAX value and the pulse counter will be reset to zero to restart the next cyclic counting.

6.23.4 Input Noise Filter

Each pin of QEI inputs is equipped a noise filter which can filter the unwanted noise from. The QEA, QEB and IDX noise filters can be disabled through bits NFDIS. If enabled, the capture logic required to sample 4 consecutive same capture input value in order to recognize an edge as a capture event. A possible implementation of digital noise filter is as follow; the interval between pulses requirement for input capture is 4 QEI_CLK clocks width. Any pulse width less than or equal to 3 QEI_CLK clocks will not have any trigger. CHA, CHB and CHX are the outputs of QEA, QEB and IDX respectively after going through noise filter and polarity control. Refer to the following figures. If the noise filter is disabled the input signals QEA, QEB and IDX are passed to the internal signals CHA, CHB and CHX respectively without any delay.

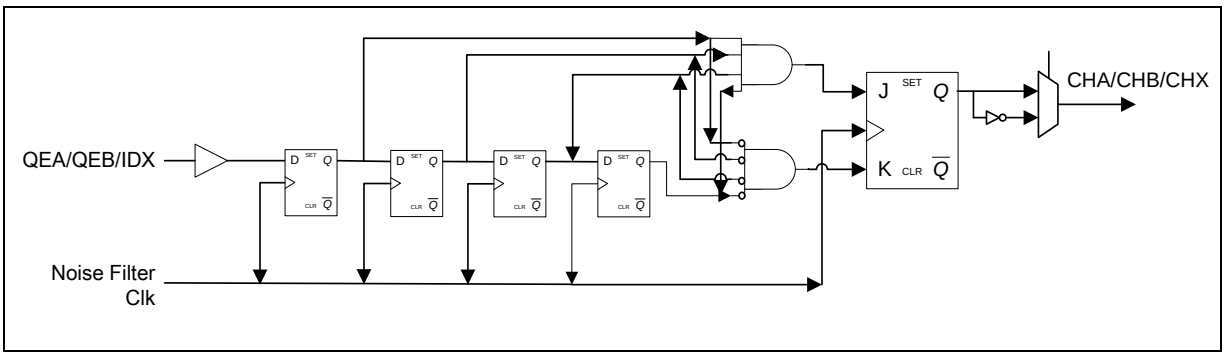


Figure 6.23-3 Noise Filter

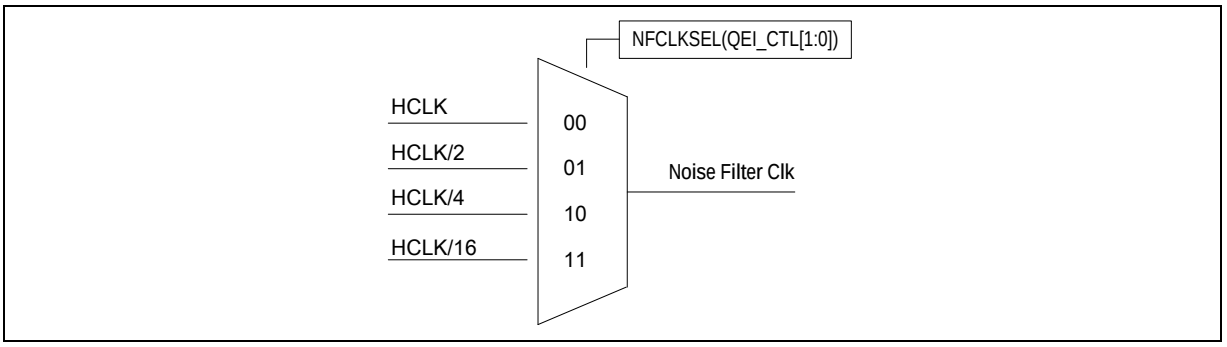


Figure 6.23-4 Noise Filter Sampling Clock Selection

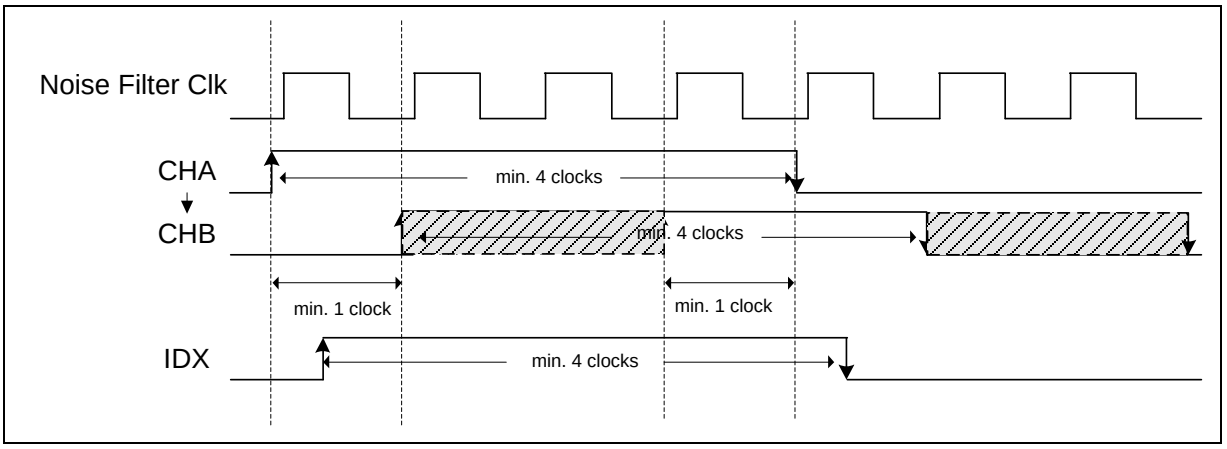


Figure 6.23-5 QEA/QEB/IDX Timing Requirement through Noise Filter

6.23.5 Operation of Quadrature Encoder Interface

There are four Quadrature encoder pulse counter operation modes

- Mode0: x4 free-counting mode
- Mode1: x2 free-counting mode
- Mode2: x4 compare-counting mode
- Mode3: x2 compare-counting mode

6.23.5.1 Free-counting mode

The quadrature encoder pulse counter (QEI_CNT) up or down counts according direction indication bit DIRF (QEI_STATUS [8]). When overflow or underflow occurs, it sets flag OVUNF (QEI_STATUS[2]). Refer to Figure 6.23-6 and Figure 6.23-7 for detailed timing.

6.23.5.2 Compare-counting mode

Pulse counter up or down counts according to direction indication bit DIRF (QEI_STATUS[8]). On up counting, flag OVUNF (QEI_STATUS[2]) will be asserted when QEI_CNT overflows from QEI_CNTMAX to zero on the next CHA edge for x2 counting mode, and on CHA/CHB edge for x4

counting mode. On down counting, flag OVUNF (QEI_STATUS[2]) will be asserted when QEI_CNT underflows from zero to QEI_CNTMAX on the next CHA edge for x2 counting mode, and on CHA/CHB edge for x4 counting mode. This mode provides the position of a rotor to user. If a quadrature encoder output 1024 pulses to CHA per round, user can write QEI_CNTMAX and QEI_CNTCMP with 4095 in x4 mode or 2047 in x2 mode and reset QEI_CNT at initial before compare-counting mode is active. When the QEI_CNT overflows from QEI_CNTCMP, here QEI_CNTCMP should be preset the same value as QEI_CNTMAX, it means rotor runs one round on next CHA/CHB edge. Refer to Figure 6.23-6 and Figure 6.23-7 for detailed timing.

6.23.5.3 X4/X2 counting modes

In x4 counting mode, the pulse counter increases or decreases one on every CHA and CHB edge based on the phase relationship of CHA and CHB signals.

QEI x4 Counting mode provides for a finer resolution of the rotor position, since the counter increments or decrements more frequently for each QEA/QEB input pulse pair than in QEI x2 mode. This mode is selected by setting the QEI Counting Mode Selection bits MODE(QEI_CTL[9:8]) to 00b or 01b. In this mode, the QEI logic detects every edge on every QEA and QEB input edges.

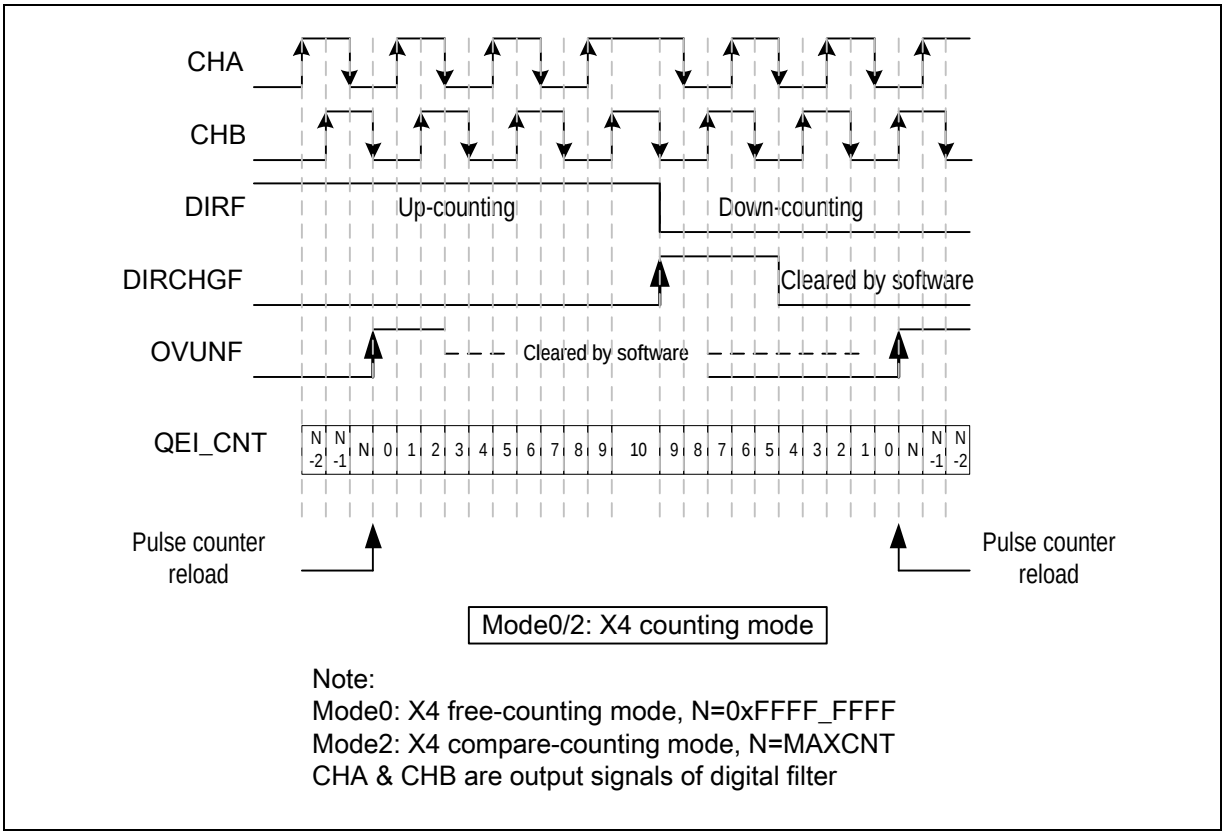


Figure 6.23-6 X4 Counting Mode

In x2 counting mode, the pulse counter increases or decreases one on every CHA edge based on the phase relationship of CHA and CHB signals.

QEI x2 Counting mode is selected by setting the QEI Counting Mode Selection bits MODE(QEI_CTL[9:8]) to 01b or 11b. In this mode, the QEI logic detects every edge on the QEA input only. Every rising and falling edge on the QEA signal clocks the pulse counter.

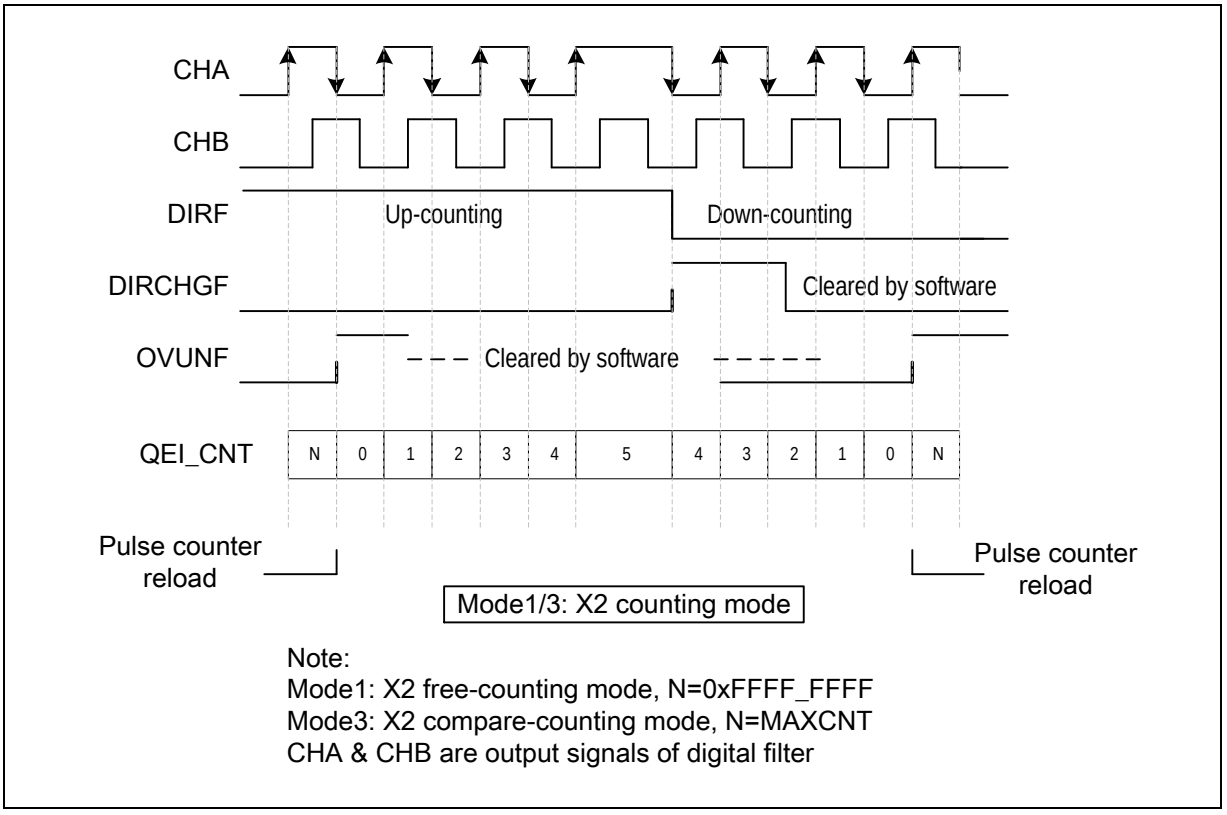


Figure 6.23-7 X2 Counting Mode

6.23.5.4 Direction of Count

If CHA lead CHB, the pulse counter is increased by 1. If CHA lags CHB, the pulse counter is decreased by 1. The QEI control logic generates a signal that sets the DIRF (QEI_STATUS[8]); this in turn determines the direction of the count. When CHA leads CHB, DIRF is set as 1, and the position counter increments on every active edge. When CHA lags CHB, DIRF is cleared, and the position counter decrements on every active edge. Refer to below table.

Current Signal Detected	Previous Signal Detected				DIRF (Counting Direction)
	Rising		Falling		
	CHA	CHB	CHA	CHB	
CHA rising				✓	1 (Increment)
		✓			0 (Decrement)
			✓		Toggle (direction change)
CHA falling				✓	0 (Decrement)

		✓			1 (Increment)
	✓				Toggle (direction change)
CHB rising	✓				1 (Increment)
			✓		0 (Decrement)
				✓	Toggle (direction change)
CHB falling			✓		1 (Increment)
	✓				0 (Decrement)
		✓			Toggle (direction change)

Table 6.23-1 Direction of Count

6.23.5.5 Up-Counting

Under the forward direction the DIRF bit is 1 when up-counting. Software needs to clear the OVUNF (QEI_STATUS[2]). For the free-counting mode the QEI_CNT counter will count until it matches 0xFFFF_FFFF and next edges on the forward direction will set the bit OVUNF (QEI_STATUS[2]) high and reset QEI_CNT to zero. For compare-counting mode the QEI_CNT counter counts until the QEI_CNTMAX value and next edges on the forward direction will set the bit OVUNF (QEI_STATUS[2]) high and reset QEI_CNT to zero. Changes of direction trigger a down-count and QEI_CNT decreasing in counter value. For X2 mode, only CHA edge will set OVUNF (QEI_STATUS[2]) while for X4 mode both CHA and CHB edges will set OVUNF(QEI_STATUS[2]).

6.23.5.6 Down-Counting

A change of direction will cause the counter to down count for X2/X4 counting mode. It is indicated with the DIRF bit as 0 and DIRCHGF (QEI_STATUS[3]) flag is set to 1. At this stage the QEI_CNT will start to down-count. In free-counting mode the pulse counter will reload with 0xFFFF_FFFF when it down counts to zero and sets OVUNF (QEI_STATUS[2]) to high in the next edge. The pulse counter will reload with QEI_CNTMAX when it down counts to zero in compare-counting mode and sets OVUNF (QEI_STATUS[2]) to high in the next edge. For X2 mode, only CHA edge will set OVUNF (QEI_STATUS[2]) while for X4 mode both CHA and CHB edges will set OVUNF(QEI_STATUS[2]).

6.23.6 Compare Function

The compare function in QEI controller is to compare the dynamic counting QEI_CNT with the compare register QEI_CNTCMP. When QEI_CNT up or down counts and reaches QEI_CNTCMP, the flag CMPF (QEI_STATUS[1]) will be set. Set bit CMPEN (QEI_CTL[28]) to one to enable the compare function otherwise disable it.

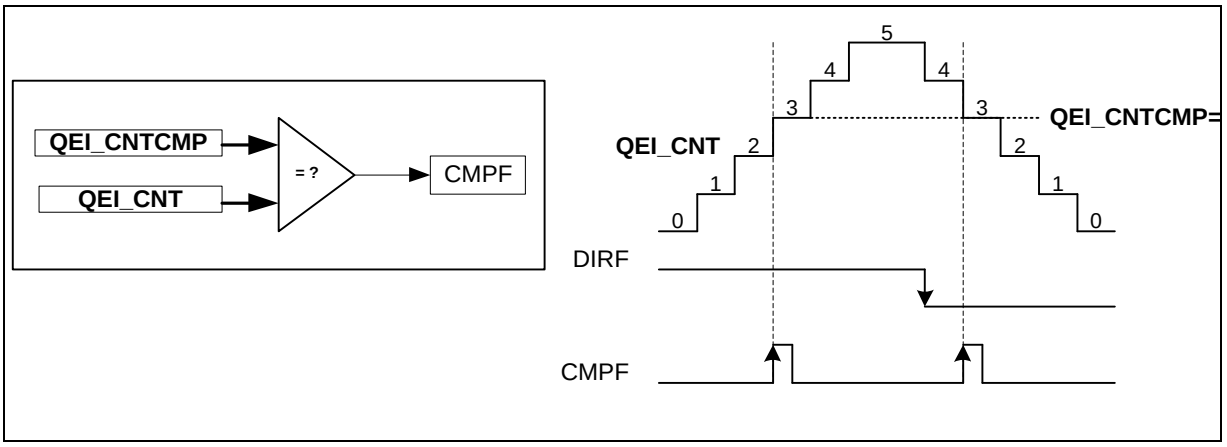


Figure 6.23-8 Compare Operation

6.23.7 Reload Counter by Pin IDX

The QEI_CNT counter can be reset to zero or reload with the content of QEI_CNTMAX by the signal CHX (the filtered and polarity-set output of pin IDX) trigger. When the IDX Reload bit **IDXRLDEN** (QEI_CTL[27]) is set, a rising edge of CHX causes QEI controller to reset the QEI_CNT to zero if the counter is in up-counting; if the counter is in down-counting the rising edge of CHX causes the QEI controller reload the QEI_CNT with the content of QEI_CNTMAX. Refer to the following figure for details.

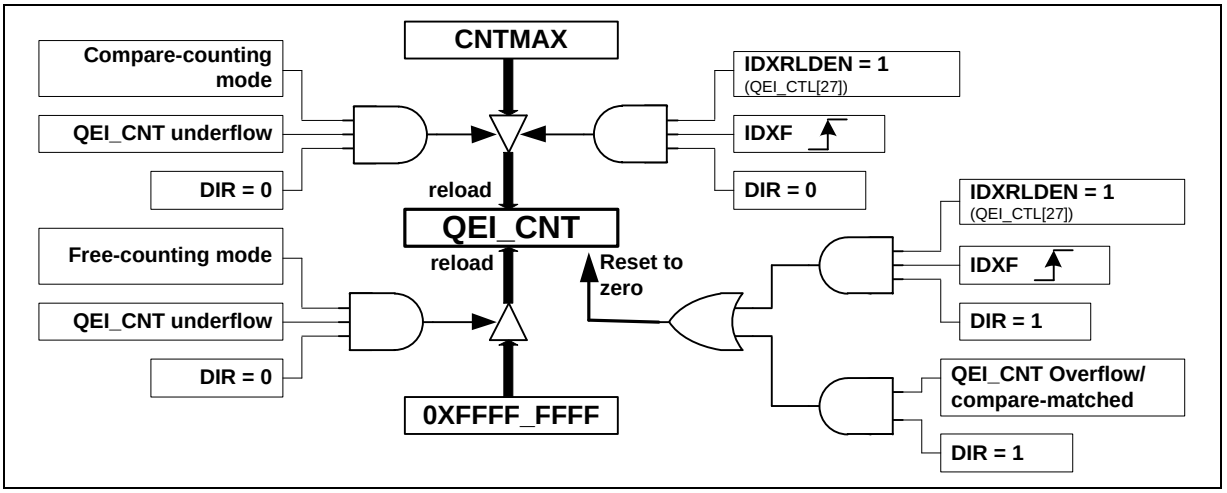


Figure 6.23-9 QEI_CNT Reload/Reset Control

6.23.8 Capture QEI Counter

If the bit HOLDCNT (QEI_CTL[24]) is set, the QEI_CNT content will be captured into QEI Counter Hold Register (QEI_CNTHOLD), the data will be held until the next HOLDCNT (QEI_CTL[24]) trigger comes. The bit HOLDCNT can be set by writing 1 to it or the rising edge of timers interrupt flags TIF (TIMERx_INTSTS[0]).

Note: The bit HOLDCNT is automatically cleared by hardware after QEI_CNTHOLD captures the content of QEI counter.

If the bit IDXLATEN (QEI_CTL[25]) is set, the QEI_CNT content will be latched into QEI Counter Index Latch Register (QEI_CNTLATCH) at every rising edge of CHX signal.

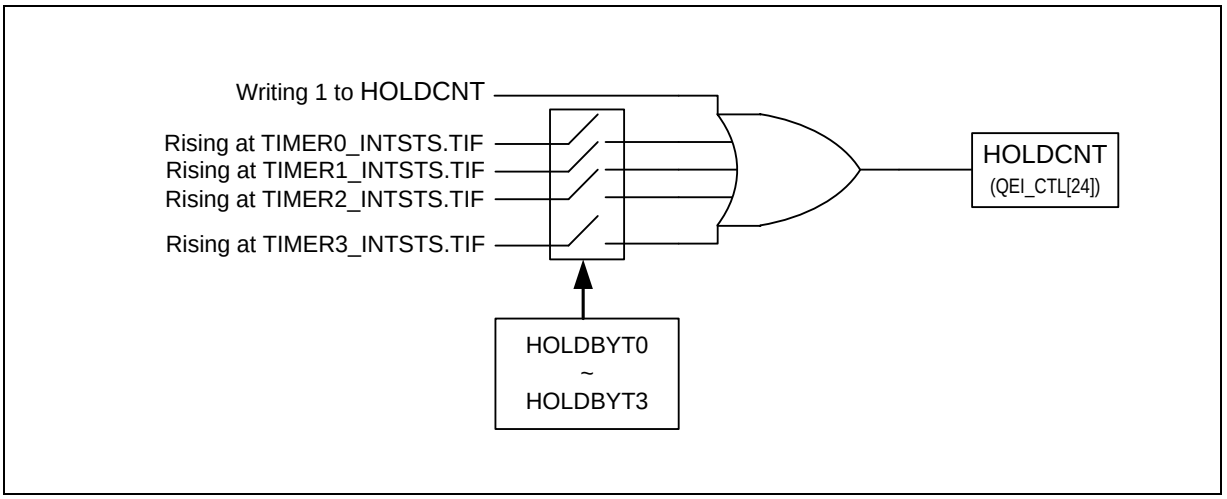


Figure 6.23-10 Trigger Control of Capturing QEI Counter

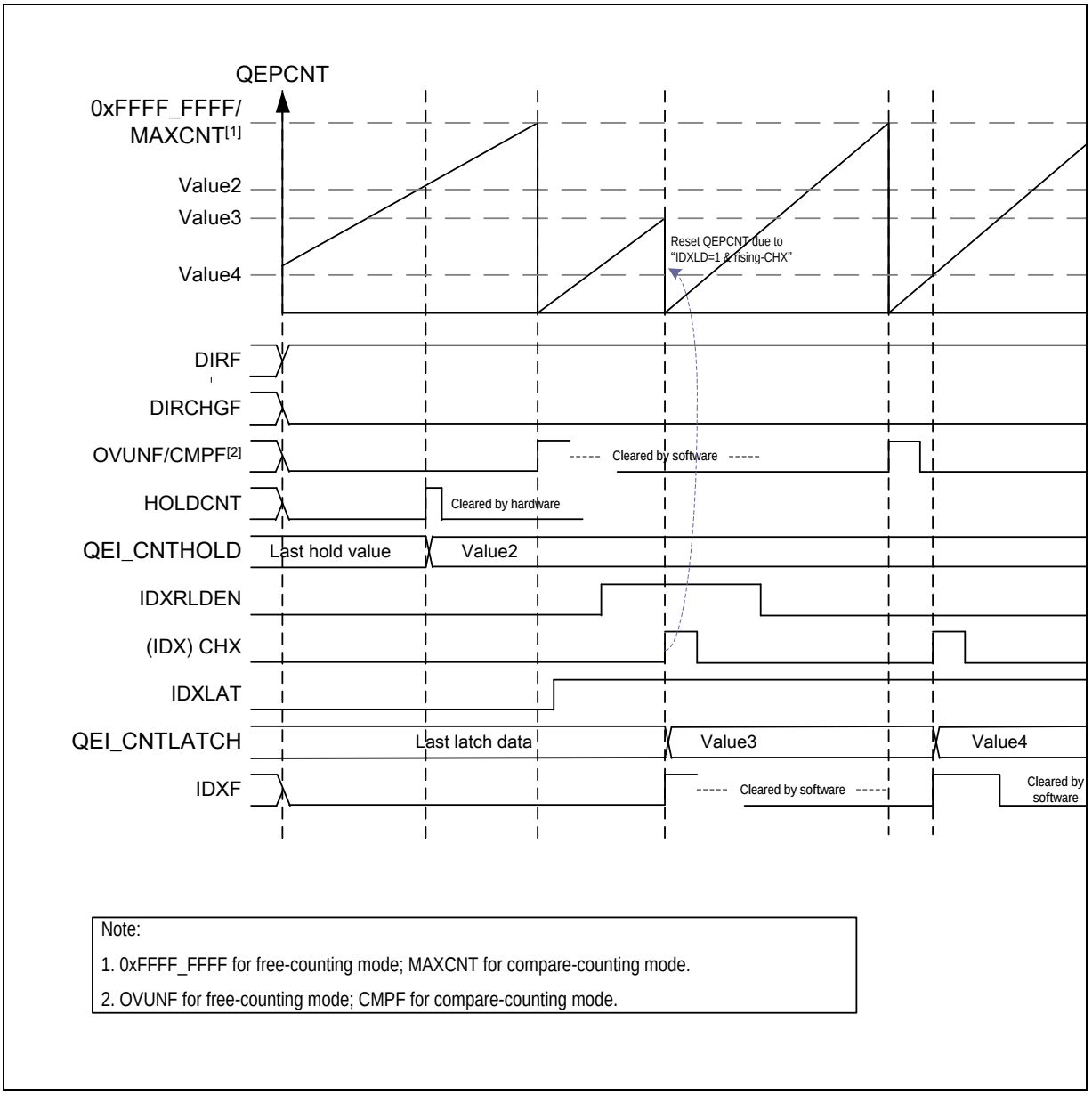


Figure 6.23-11 Capture and Latch QEI Counter

6.23.9 QEI Interrupt Architecture

There are four interrupt sources, each one of them has an interrupt flag and enable control bit, can trigger QEI Interrupt. When QEI counter is up-counting and QEI_CNT overflows or down-counting and underflows, the Overflow/Underflow flag OVUNF(QEI_STATUS[2]) will be set by hardware and it will trigger QEI Interrupt request if bit OVUNIEN (QEI_CTL[16]) is high. When QEI controller detects the encoder rotation change, it toggles the direction indication bit DIRF (QEI_STATUS[8]) and the flag DIRCHGF (QEI_STATUS[3]) will be set by hardware that requests the QEI interrupt if bit DIRIEN (QEI_CTL[17]) is set. When the QEI counter counting value is equal to the value of QEI Counter Compare Register (QEI_CNTCMP), the flag CMPF (QEI_STATUS[1]) will be set by hardware and the QEI Interrupt will be requested if bit CMPIEN (QEI_CTL[18]) is high. When QEI controller detects a

rising edge at signal CHX (the filtered and polarity-set output of pin IDX), the flag IDXF (QEI_STATUS[0]) will set by hardware and the QEI interrupt will be requested if bit IDXIEN (QEI_CTL[19]) is set. **Note that the four flags, OVUNF, DIRCHGF, CMPF and IDXF are set by hardware and must be cleared by software.** The following figure demonstrates the architecture of Quadrature Encoder Interface Controller interrupts.

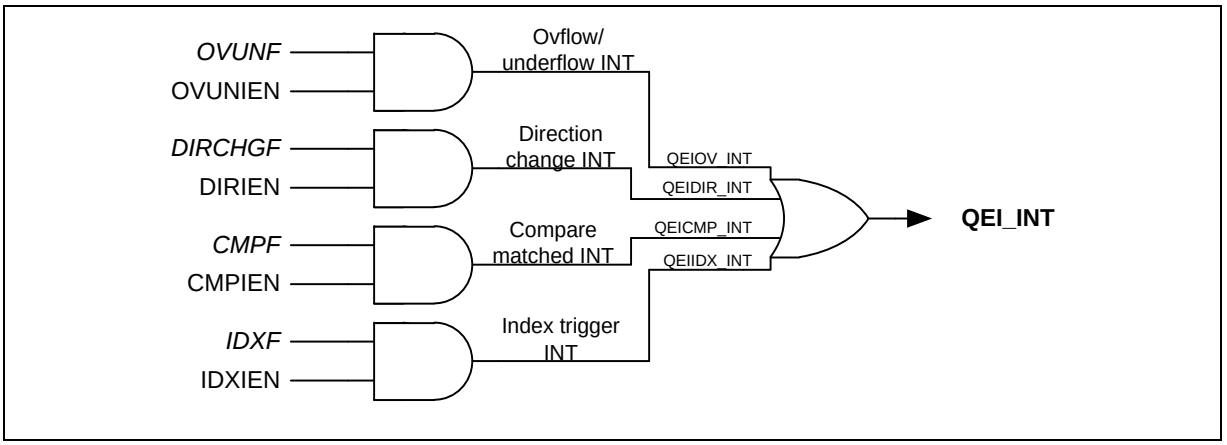


Figure 6.23-12 Quadrature Encoder Interface Interrupt Architecture Diagram

6.23.10 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
QEI Base Address: **QEIn_BA = 0x400B_0000 + n * 0x1000** **n=0,1**				
QEI_CNT	QEIn_BA+0x00	R/W	QEI Pulse Counter	0x0000_0000
QEI_CNTHOLD	QEIn_BA+0x04	R/W	QEI Pulse Counter Hold Register	0x0000_0000
QEI_CNTLATCH	QEIn_BA+0x08	R/W	QEI Pulse Counter Index Latch Register	0x0000_0000
QEI_CNTCMP	QEIn_BA+0x0C	R/W	QEI Pulse Counter Compare Register	0x0000_0000
QEI_CNTMAX	QEIn_BA+0x14	R/W	QEI Pre-set Maximum Count Register	0x0000_0000
QEI_CTL	QEIn_BA+0x18	R/W	QEI Controller Control Register	0x0000_0000
QEI_STATUS	QEIn_BA+0x2C	R/W	QEI Controller Status Register	0x0000_0000

6.23.11 Register Description

QEI Pulse Counter Register (QEI_CNT)

Register	Offset	R/W	Description	Reset Value
QEI_CNT	QEIn_BA+0x00	R/W	QEI Pulse Counter Register	0x0000_0000

31	30	29	28	27	26	25	24
VAL[31:24]							
23	22	21	20	19	18	17	16
VAL[23:16]							
15	14	13	12	11	10	9	8
VAL[15:8]							
7	6	5	4	3	2	1	0
VAL[7:0]							

Bits	Description	
[31:0]	VAL	**Quadrature Encoder Pulse Counter** A 32-bit up/down counter. When an effective phase pulse is detected, this counter is increased by one if the bit DIRF (QEI_STATUS[8]) is one or decreased by one if the bit DIRF is zero. This register performs an integrator which count value is proportional to the encoder position. The pulse counter may be initialized to a predetermined value by one of three events occurs: 1. Software is written if QEIEN (QEI_CTL[29]) = 0. 2. Compare-match event if QEIEN=1 and QEI is in compare-counting mode. 3. Index signal change if QEIEN=1 and IDXRLDEN (QEI_CTL[27])=1.

QEI Pulse Counter Hold Register (QEI_CNTHOLD)

Register	Offset	R/W	Description	Reset Value
QEI_CNTHOLD	QEIn_BA+0x04	R/W	QEI Pulse Counter Hold Register	0x0000_0000

31	30	29	28	27	26	25	24
VAL[31:24]							
23	22	21	20	19	18	17	16
VAL[23:16]							
15	14	13	12	11	10	9	8
VAL[15:8]							
7	6	5	4	3	2	1	0
VAL[7:0]							

Bits	Description	
[31:0]	VAL	**Quadrature Encoder Pulse Counter Hold Register** When bit HOLDCNT (QEIx_CTL[24]) goes from low to high, the QEI_CNT value is copied into QEI_CNTHOLD register.

QEI Pulse Counter Index Latch Register (QEI_CNTLATCH)

Register	Offset	R/W	Description	Reset Value
QEI_CNTLATCH	QEIn_BA+0x08	R/W	QEI Pulse Counter Index Latch Register	0x0000_0000

31	30	29	28	27	26	25	24
VAL[31:24]							
23	22	21	20	19	18	17	16
VAL[23:16]							
15	14	13	12	11	10	9	8
VAL[15:8]							
7	6	5	4	3	2	1	0
VAL[7:0]							

Bits	Description	
[31:0]	VAL	**Quadrature Encoder Pulse Counter Index Latch** When the IDXF (QEI_STATUS[0]) bit is set, the QEI_CNT value is copied into QEI_CNTLATCH register.

QEI Pulse Counter Compare-Match Register (QEI_CNTCMP)

Register	Offset	R/W	Description	Reset Value
QEI_CNTCMP	QEIn_BA+0x0C	R/W	QEI Pulse Counter Compare Register	0x0000_0000

31	30	29	28	27	26	25	24
VAL[31:24]							
23	22	21	20	19	18	17	16
VAL[23:16]							
15	14	13	12	11	10	9	8
VAL[15:8]							
7	6	5	4	3	2	1	0
VAL[7:0]							

Bits	Description	
[31:0]	VAL	**Quadrature Encoder Pulse Counter Compare** if the QEI controller is in the compare-counting mode CMPEN (QEI_CTL[28]) =1, when the value of QEI_CNT matches the value of VAL the bit CMPF will be set. This register is software writable.

QEI Preset Maximum Count Register (QEI_CNTMAX)

Register	Offset	R/W	Description	Reset Value
QEI_CNTMAX	QEIn_BA+0x14	R/W	QEI Pre-set Maximum Count Register	0x0000_0000

31	30	29	28	27	26	25	24
VAL[31:24]							
23	22	21	20	19	18	17	16
VAL[23:16]							
15	14	13	12	11	10	9	8
VAL[15:8]							
7	6	5	4	3	2	1	0
VAL[7:0]							

Bits	Description	
[31:0]	VAL	**Quadrature Encoder Preset Maximum Count** This register value determined by user stores the maximum value which may be the number of the quadrature encoder pulses in a revolution for the QEI controller compare-counting mode.

Quadrature Encoder Interface Control Register (QEI_CTL)

Register	Offset	R/W	Description	Reset Value
QEI_CTL	QEIn_BA+0x18	R/W	QEI Controller Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		QEIEN	CMPEN	IDXRLDEN	Reserved	IDXLATEN	HOLDCNT
23	22	21	20	19	18	17	16
HOLDTMR3	HOLDTMR2	HOLDTMR1	HOLDTMR0	IDXIEN	CMPIEN	DIRIEN	OVUNIEN
15	14	13	12	11	10	9	8
Reserved	IDXINV	CHBINV	CHAINV	Reserved		MODE	
7	6	5	4	3	2	1	0
Reserved	IDXEN	CHBEN	CHAEN	NFDIS	Reserved	NFCLKSEL	

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29]	**QEIEN**	**Quadrature Encoder Interface Controller Enable Bit** 0 = QEI controller function Disabled. 1 = QEI controller function Enabled.
[28]	**CMPEN**	**The Compare Function Enable Bit** The compare function in QEI controller is to compare the dynamic counting QEI_CNT with the compare register QEI_CNTCMP, if QEI_CNT value reaches QEI_CNTCMP, the flag CMPF will be set. 0 = Compare function Disabled. 1 = Compare function Enabled.
[27]	**IDXRLDEN**	**Index Trigger QEI_CNT Reload Enable Bit** When this bit is high and a rising edge comes on signal CHX, the QEI_CNT will be reset to zero if the counter is in up-counting type (DIRF = 1); while the QEI_CNT will be reloaded with QEI_CNTMAX content if the counter is in down-counting type (DIRF = 0). 0 = Reload function Disabled. 1 = QEI_CNT re-initialized by Index signal Enabled.
[26]	**Reserved**	Reserved.
[25]	**IDXLATEN**	**Index Latch QEI_CNT Enable Bit** If this bit is set to high, the QEI_CNT content will be latched into QEI_CNTLATCH at every rising on signal CHX. 0 = The index signal latch QEI counter function Disabled. 1 = The index signal latch QEI counter function Enabled.

Bits	Description	
[24]	**HOLDCNT**	**Hold QEI_CNT Control** When this bit is set from low to high, the QEI_CNT value is copied into QEI_CNTHOLD. This bit may be set by writing 1 to it or Timer0~Timer3 interrupt flag TIF (TIMERx_INTSTS[0]). 0 = No operation. 1 = QEI_CNT content is captured and stored in QEI_CNTHOLD. **Note:** This bit is automatically cleared after QEI_CNTHOLD holds QEI_CNT value.
[23]	**HOLDTMR3**	**Hold QEI_CNT By Timer 3** 0 = TIF (TIMER3_INTSTS[0]) has no effect on HOLDCNT. 1 = A rising edge of bit TIF(TIMER3_INTSTS[0]) in timer 3 sets HOLDCNT to 1.
[22]	**HOLDTMR2**	**Hold QEI_CNT By Timer 2** 0 = TIF(TIMER2_INTSTS[0]) has no effect on HOLDCNT. 1 = A rising edge of bit TIF(TIMER2_INTSTS[0]) in timer 2 sets HOLDCNT to 1.
[21]	**HOLDTMR1**	**Hold QEI_CNT By Timer 1** 0 = TIF(TIMER1_INTSTS[0]) has no effect on HOLDCNT. 1 = A rising edge of bit TIF (TIMER1_INTSTS[0]) in timer 1 sets HOLDCNT to 1.
[20]	**HOLDTMR0**	**Hold QEI_CNT By Timer 0** 0 = TIF (TIMER0_INTSTS[0]) has no effect on HOLDCNT. 1 = A rising edge of bit TIF(TIMER0_INTSTS[0]) in timer 0 sets HOLDCNT to 1.
[19]	**IDXIEN**	**IDXF Trigger QEI Interrupt Enable Bit** 0 = The IDXF can trigger QEI interrupt Disabled. 1 = The IDXF can trigger QEI interrupt Enabled.
[18]	**CMPIEN**	**CMPF Trigger QEI Interrupt Enable Bit** 0 = CMPF can trigger QEI controller interrupt Disabled. 1 = CMPF can trigger QEI controller interrupt Enabled.
[17]	**DIRIEN**	**DIRCHGF Trigger QEI Interrupt Enable Bit** 0 = DIRCHGF can trigger QEI controller interrupt Disabled. 1 = DIRCHGF can trigger QEI controller interrupt Enabled.
[16]	**OVUNIEN**	**OVUNF Trigger QEI Interrupt Enable Bit** 0 = OVUNF can trigger QEI controller interrupt Disabled. 1 = OVUNF can trigger QEI controller interrupt Enabled.
[15]	**Reserved**	Reserved.
[14]	**IDXINV**	**Inverse IDX Input Polarity** 0 = Not inverse IDX input polarity. 1 = IDX input polarity is inversed to QEI controller.
[13]	**CHBINV**	**Inverse QEB Input Polarity** 0 = Not inverse QEB input polarity. 1 = QEB input polarity is inversed to QEI controller.
[12]	**CHAINV**	**Inverse QEA Input Polarity** 0 = Not inverse QEA input polarity. 1 = QEA input polarity is inversed to QEI controller.
[11:10]	**Reserved**	Reserved.

Bits	Description	
[9:8]	**MODE[1:0]**	**QEI Counting Mode Selection** There are four quadrature encoder pulse counter operation modes. 00 = X4 Free-counting Mode. 01 = X2 Free-counting Mode. 10 = X4 Compare-counting Mode. 11 = X2 Compare-counting Mode.
[7]	**Reserved**	Reserved.
[6]	**IDXEN**	**IDX Input To QEI Controller Enable Bit** 0 = IDX input to QEI Controller Disabled. 1 = IDX input to QEI Controller Enabled.
[5]	**CHBEN**	**QEB Input To QEI Controller Enable Bit** 0 = QEB input to QEI Controller Disabled. 1 = QEB input to QEI Controller Enabled.
[4]	**CHAEN**	**QEA Input To QEI Controller Enable Bit** 0 = QEA input to QEI Controller Disabled. 1 = QEA input to QEI Controller Enabled.
[3]	**NFDIS**	**QEI Controller Input Noise Filter Disable Bit** 0 = The noise filter of QEI controller Enabled. 1 = The noise filter of QEI controller Disabled.
[2]	**Reserved**	Reserved.
[1:0]	**NFCLKSEL[1:0]**	**Noise Filter Clock Pre-Divide Selection** To determine the sampling frequency of the Noise Filter clock . 00 = QEI_CLK. 01 = QEI_CLK/2. 10 = QEI_CLK/4. 11 = QEI_CLK/16.

Quadrature Encoder Interface Status Register (QEI_STATUS)

Register	Offset	R/W	Description	Reset Value
QEI_STATUS	QEIn_BA+0x2C	R/W	QEI Controller Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							DIRF
7	6	5	4	3	2	1	0
Reserved				DIRCHGF	OVUNF	CMPF	IDXF

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**DIRF**	**QEI Counter Counting Direction Indication** 0 = QEI Counter is in down-counting. 1 = QEI Counter is in up-counting. **Note:** This bit is set/reset by hardware according to the phase detection between CHA and CHB.
[7:4]	**Reserved**	Reserved.
[3]	**DIRCHGF**	**Direction Change Flag** Flag is set by hardware while QEI counter counting direction is changed. Software can clear this bit by writing 1 to it. 0 = No change in QEI counter counting direction. 1 = QEI counter counting direction is changed. **Note:** This bit is only cleared by writing 1 to it.
[2]	**OVUNF**	**QEI Counter Overflow Or Underflow Flag** Flag is set by hardware while QEI_CNT overflows from 0xFFFF_FFFF to zero in free-counting mode or from the QEI_CNTMAX value to zero in compare-counting mode. Similarly, the flag is set wile QEI counter underflows from zero to 0xFFFF_FFFF or QEI_CNTMAX. 0 = No overflow or underflow occurs in QEI counter. 1 = QEI counter occurs counting overflow or underflow. **Note:** This bit is only cleared by writing 1 to it.
[1]	**CMPF**	**Compare-Match Flag** If the QEI compare function is enabled, the flag is set by hardware while QEI counter up or down counts and reach to the QEI_CNTCMP value. 0 = QEI counter does not match with QEI_CNTCMP value. 1 = QEI counter counts to the same as QEI_CNTCMP value. **Note:** This bit is only cleared by writing 1 to it.

Bits	Description	
[0]	**IDXF**	**IDX Detected Flag** When the QEI controller detects a rising edge on signal CHX it will set flag IDXF to high. 0 = No rising edge detected on signal CHX. 1 = A rising edge occurs on signal CHX. **Note:** This bit is only cleared by writing 1 to it.

6.24 Real Time Clock (RTC)

6.24.1 Overview

The Real Time Clock (RTC) controller provides the real time clock and calendar information. The clock source of RTC controller is from an external 32.768 kHz low-speed crystal which connected at pins X32I and X32O (refer to pin Description) or from an external 32.768 kHz low-speed oscillator output fed at pin X32I. The RTC controller provides the real time clock (hour, minute, second) in RTC_TIME (RTC Time Loading Register) as well as calendar information (year, month, day) in RTC_CAL (RTC Calendar Loading Register). It also offers RTC alarm function that user can preset the alarm time in RTC_TALM (RTC Time Alarm Register) and alarm calendar in RTC_CALM (RTC Calendar Alarm Register). The data format of RTC time and calendar message are all expressed in BCD (Binary Coded Decimal) format.

The RTC controller supports periodic RTC Time Tick and Alarm Match interrupts. The periodic RTC Time Tick interrupt has 8 period interval options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second which are selected by RTC_TICK (RTC_TICK[2:0] Time Tick Register). When real time and calendar message in RTC_TIME and RTC_CAL are equal to alarm time and calendar settings in RTC_TALM and RTC_CALM, the ALMIF (RTC_INTSTS [0] RTC Alarm Interrupt Flag) is set to 1 and the RTC alarm interrupt signal is generated if the ALMIEN (RTC_INTEN [0] Alarm Interrupt Enable) is enabled.

Both RTC Time Tick and Alarm Match interrupt signal can cause chip to wake-up from Idle or Power-down mode if the corresponding interrupt enable bit (ALMIEN or TICKIEN) is set to 1 before chip enters Idle or Power-down mode.

6.24.2 Features

- Supports real time counter in RTC_TIME (hour, minute, second) and calendar counter in RTC_CAL (year, month, day) for RTC time and calendar check
- Supports alarm time (hour, minute, second) and calendar (year, month, day) settings in RTC_TALM and RTC_CALM
- Selectable 12-hour or 24-hour time scale in RTC_CLKFMT register
- Supports Leap Year indication in RTC_LEAPYEAR register
- Supports Day of the Week counter in RTC_WEEKDAY register
- Frequency of RTC clock source compensated by the RTC_FREQADJ register
- All time and calendar message expressed in BCD format
- Supports periodic RTC Time Tick interrupt with 8 period interval options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second
- Supports RTC Time Tick and Alarm Match interrupt
- Supports chip wake-up from Idle or Power-down mode while a RTC interrupt signal is generated
- Supports 96 bytes spare registers to store user's important information
- Supports a tamper detect function to detect the transition of tamper detect pin

6.24.3 Block Diagram

The block diagram of Real Time Clock is shown below.

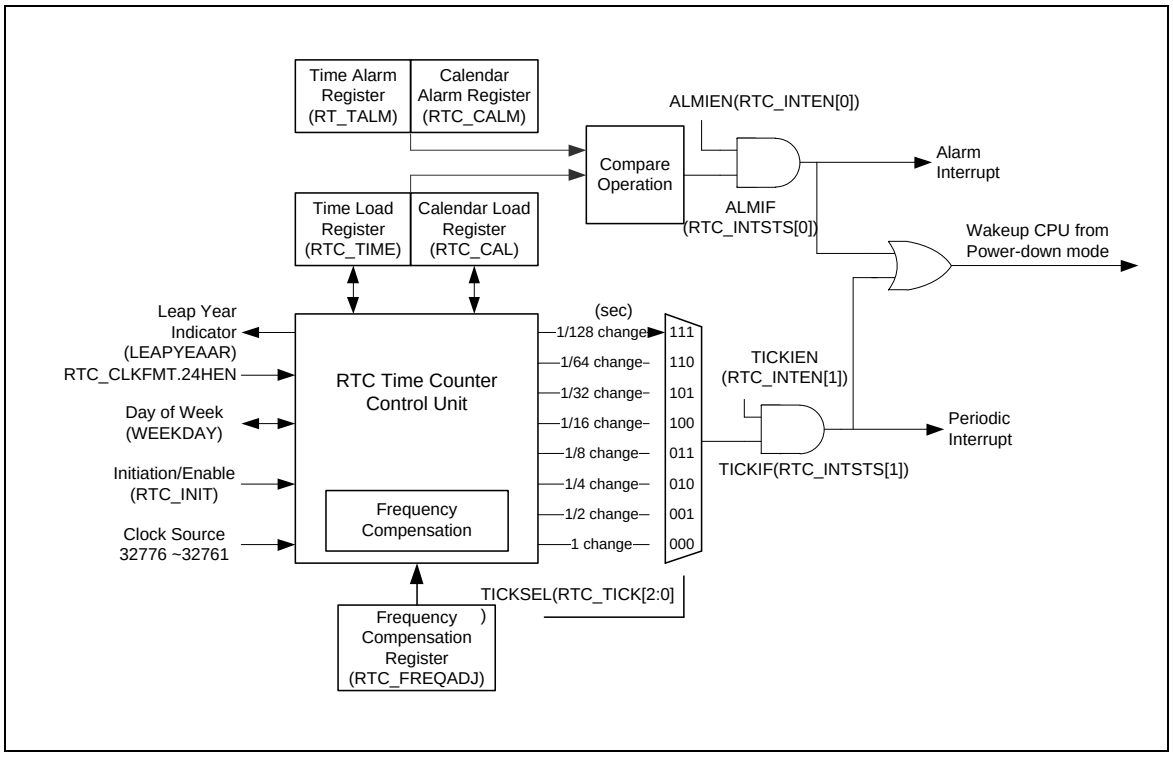


Figure 6.24-1 RTC Block Diagram

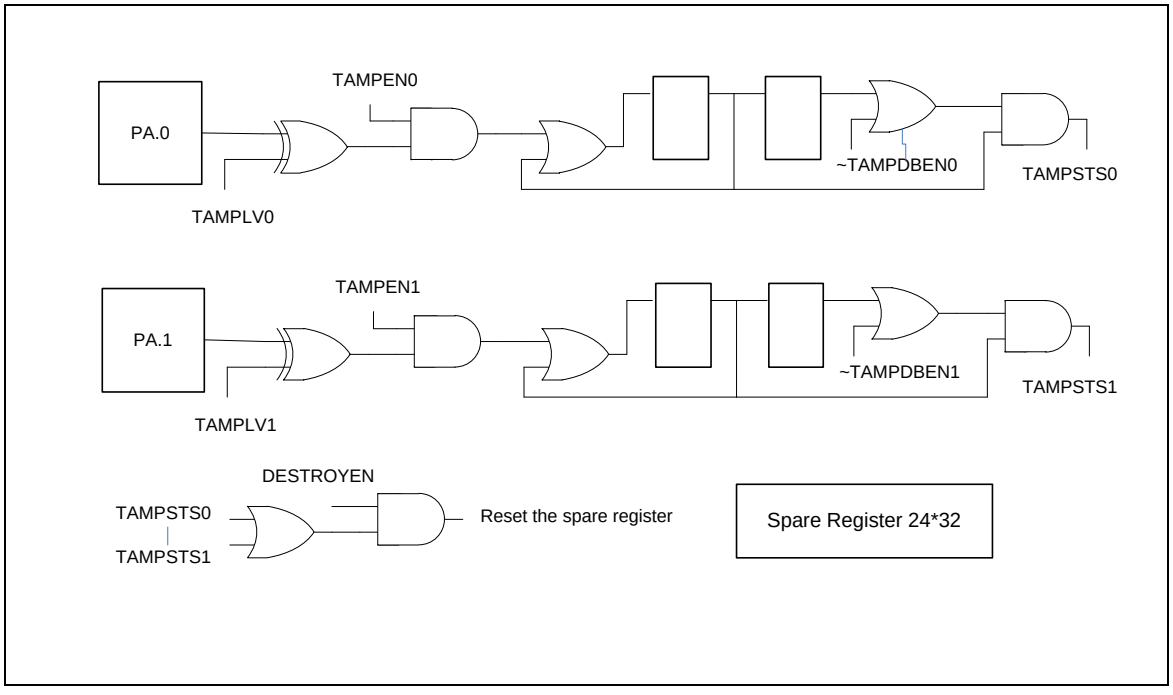


Figure 6.24-2 Tamper Detector and Spare Register

6.24.4 Functional Description

6.24.4.1 RTC Initiation

When a RTC block is powered on, RTC is at reset state. User has to write a number 0xa5eb1357 to RTC_INIT (RTC_INIT [31:0] RTC Initiation) register to make RTC leaving reset state. Once the RTC_INIT is written as 0xa5eb1357, the RTC will be in normal active state permanently. User can read Active (RTC_INIT[0] RTC Active Status) bit status to check the RTC is at normal active state or reset state.

6.24.4.2 Access to RTC register

Due to clock frequency difference between RTC clock and system clock, when user write new data to any one of the RTC registers, the data will not be updated until 2 RTC clocks later (about 60us).

In addition, user must be aware that RTC controller does not check whether loaded data is out of bounds or not in RTC_TIME, RTC_CAL, RTC_TALM and RTC_CALM registers. RTC does not check rationality between RTC_WEEKDAY and RTC_CAL either.

6.24.4.3 RTC Read/Write Enable

RTC_RWEN [RTC_RWEN [15:0] RTC Register Access Enable Password] is used to unlock RCT registers read/write protect function. If RTC_RWEN [15:0] is written to 0xA965, user can read RWENF (RTC_RWEN [16] RTC Register Access Enable Flag) bit status to check the RTC registers are read/write access or locked. Once RWENF bit enabled, RTC Access Enable function will keep effect at least 1024 RTC clocks (about 30ms) and RWENF bit will be cleared automatically after 1024 RTC clocks.

6.24.4.4 Frequency Compensation

The RTC source clock may not precise to exactly 32768 Hz and the RTC_FREQADJ register (Frequency Compensation Register) allows software to make digital compensation to the RTC source clock only if the frequency of RTC source clock is in the range from 32761 Hz to 32776 Hz.

Following are the compensation examples for the real RTC source clock is higher or lower than 32768 Hz.

Example 1: (RTC Source Clock > 32768 Hz)
RTC Source Clock Measured: 32773.65 Hz (> 32768 Hz)
Integer Part: 32773 => 0x8005
INTEGER (RTC_FREQADJ [11:8] Integer Part) = 0x05 – 0x01 + 0x08 = 0x0c
Fraction Part: 0.65
FRACTION (RTC_FREQADJ [5:0] Fraction Part) = 0.65 X 64 = 41.6 = 0x2A
RTC_FREQADJ Register Should Be As 0xC2A
Example 2: (RTC source clock ≦ 32768 Hz)
RTC source clock measured: 32765.27 Hz (≦ 32768 Hz)
Integer part: 32765 => 0x7FFD
INTEGER (RTC_FREQADJ [11:8] Integer Part) = 0x0D – 0x01 – 0x08 = 0x04
Fraction part: 0.27
FRACTION (RTC_FREQADJ [5:0] Fraction Part) = 0.27 x 64 = 17.28 = 0x11
RTC_FREQADJ register should be as 0x411

6.24.4.5 Time and Calendar counter

RTC_TIME and RTC_CAL are used to load the real time and calendar. RTC_TALM and RTC_CALM are used for setup alarm time and calendar.

6.24.4.6 12/24 hour Time Scale Selection

The 12/24 hour time scale selection depends on RTC_CLKFMT bit (RTC_CLKFMT [0] 24-Hour / 12-Hour Time Scale Selection).

6.24.4.7 Day of the Week counter

The RTC controller provides day of week in WEEKDAY(RTC_WEEKDAY[2:0] Day of the Week Register). The value is defined from 0 to 6 to represent Sunday to Saturday respectively.

6.24.4.8 Periodic Time Tick Interrupt

The Periodic Time Tick interrupt has 8 period interval options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second that are selected by RTC_TICK (RTC_TICK[2:0] Time Tick Register). When Periodic Time Tick interrupt is enabled by setting TICKIEN (RTC_INTEN [1] Time Tick Interrupt Enable) to 1, the Periodic Time Tick interrupt is requested periodically in the period selected by RTC_TICK[2:0] settings.

6.24.4.9 Alarm Interrupt

When the real time and calendar message in RTC_TIME and RTC_CAL are equal to alarm time and calendar settings in RTC_TALM and RTC_CALM, the ALMIF (RTC_INTSTS [0] RTC Alarm Interrupt Flag) is set to 1 and the RTC alarm interrupt signal is generated if the ALMIEN (RTC_INTEN[0] Alarm Interrupt Enable) is enabled.

6.24.4.10 Alarm Registers

1. All data in RTC_TALM, RTC_CALM, RTC_TIME and RTC_CAL registers are all expressed in BCD format.
2. Programmer has to make sure that the loaded values are reasonable. For example, Load RTC_CAL as 201a (year), 13 (month), 00 (day), or RTC_CAL does not match with RTC_WEEKDAY, etc.
3. Registers value after powered on or reset:

Register	Reset State
RTC_RWEN	0
RTC_CAL	05/1/1 (year/month/day)
RTC_TIME	00:00:00 (hour : minute : second)
RTC_CALM	00/00/00 (year/month/day)
RTC_TALM	00:00:00 (hour : minute : second)
RTC_CLKFMT	1 (24-hour mode)
RTC_WEEKDAY	6 (Saturday)
RTC_INTEN	0
RTC_INTSTS	0
RTC_LEAPYEAR	0
RTC_TICK	0

In RTC_CAL and RTC_CALM, only 2 BCD digits are used to express “year”. The 2 BCD digits of xy means 20xy, rather than 19xy or 21xy.

6.24.4.11 Spare registers and tamper detect pin

The RTC equips 96 bytes spare registers to store user's important information. And also equips a tamper detect function to detect the transition of tamper detect pin. Once the transition defined in register RTC_TAMPCTL is detected in tamper detect pin, the 96 bytes spare registers will be cleared by RTC automatically.

As these 96 bytes spare registers are located in RTC (32.768 kHz) clock domain (it's asynchronous with system clock domain), a synchronization latency is necessary when writing data to these 96 bytes spare registers. Once CPU writes one of 24 spare registers (RTC_SPR0 ~ RTC_SPR23), it's necessary to poll bit SPRRWRDY (RTC_SPRCTL[7]) to check if data is written into registers. CPU could only access (read or write) the spare registers again once SPRRWRDY is high. Any access (read or write) to spare registers while SPRRWRDY low is undefined.

6.24.5 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
RTC Base Address: **RTC_BA = 0x4004_1000**				
RTC_INIT	RTC_BA+0x00	R/W	RTC Initiation Register	0x0000_0000
RTC_RWEN	RTC_BA+0x04	R/W	RTC Access Enable Register	0x0000_0000
RTC_FREQADJ	RTC_BA+0x08	R/W	RTC Frequency Compensation Register	0x0000_0700
RTC_TIME	RTC_BA+0x0C	R/W	Time Loading Register	0x0000_0000
RTC_CAL	RTC_BA+0x10	R/W	Calendar Loading Register	0x0005_0101
RTC_CLKFMT	RTC_BA+0x14	R/W	Time Scale Selection Register	0x0000_0001
RTC_WEEKDAY	RTC_BA+0x18	R/W	Day of the Week Register	0x0000_0006
RTC_TALM	RTC_BA+0x1C	R/W	Time Alarm Register	0x0000_0000
RTC_CALM	RTC_BA+0x20	R/W	Calendar Alarm Register	0x0000_0000
RTC_LEAPYEAR	RTC_BA+0x24	R	Leap Year Indication Register	0x0000_0000
RTC_INTEN	RTC_BA+0x28	R/W	RTC Interrupt Enable Register	0x0000_0000
RTC_INTSTS	RTC_BA+0x2C	R/W	RTC Interrupt Indicator Register	0x0000_0000
RTC_TICK	RTC_BA+0x30	R/W	RTC Time Tick Register	0x0000_0000
RTC_SPRCTL	RTC_BA+0x3C	R/W	RTC Spare Functional Control Register	0x0000_0080
RTC_SPR0	RTC_BA+0x40	R/W	RTC Spare Register 0	0x0000_0000
RTC_SPR1	RTC_BA+0x44	R/W	RTC Spare Register 1	0x0000_0000
RTC_SPR2	RTC_BA+0x48	R/W	RTC Spare Register 2	0x0000_0000
RTC_SPR3	RTC_BA+0x4C	R/W	RTC Spare Register 3	0x0000_0000
RTC_SPR4	RTC_BA+0x50	R/W	RTC Spare Register 4	0x0000_0000
RTC_SPR5	RTC_BA+0x54	R/W	RTC Spare Register 5	0x0000_0000
RTC_SPR6	RTC_BA+0x58	R/W	RTC Spare Register 6	0x0000_0000
RTC_SPR7	RTC_BA+0x5C	R/W	RTC Spare Register 7	0x0000_0000
RTC_SPR8	RTC_BA+0x60	R/W	RTC Spare Register 8	0x0000_0000
RTC_SPR9	RTC_BA+0x64	R/W	RTC Spare Register 9	0x0000_0000
RTC_SPR10	RTC_BA+0x68	R/W	RTC Spare Register 10	0x0000_0000

RTC_SPR11	RTC_BA+0x6C	R/W	RTC Spare Register 11	0x0000_0000
RTC_SPR12	RTC_BA+0x70	R/W	RTC Spare Register 12	0x0000_0000
RTC_SPR13	RTC_BA+0x74	R/W	RTC Spare Register 13	0x0000_0000
RTC_SPR14	RTC_BA+0x78	R/W	RTC Spare Register 14	0x0000_0000
RTC_SPR15	RTC_BA+0x7C	R/W	RTC Spare Register 15	0x0000_0000
RTC_SPR16	RTC_BA+0x80	R/W	RTC Spare Register 16	0x0000_0000
RTC_SPR17	RTC_BA+0x84	R/W	RTC Spare Register 17	0x0000_0000
RTC_SPR18	RTC_BA+0x88	R/W	RTC Spare Register 18	0x0000_0000
RTC_SPR19	RTC_BA+0x8C	R/W	RTC Spare Register 19	0x0000_0000
RTC_SPR20	RTC_BA+0x90	R/W	RTC Spare Register 20	0x0000_0000
RTC_SPR21	RTC_BA+0x94	R/W	RTC Spare Register 21	0x0000_0000
RTC_SPR22	RTC_BA+0x98	R/W	RTC Spare Register 22	0x0000_0000
RTC_SPR23	RTC_BA+0x9C	R/W	RTC Spare Register 23	0x0000_0000
RTC_TAMPCTL	RTC_BA+0x110	R/W	Tamper Control Register	0x0000_0000
RTC_TAMPSTS	RTC_BA+0x114	R/W	Tamper Status Register	0x0000_0000
RTC_TAMP0PCTL	RTC_BA+0x124	R/W	TAMPER0 Pin I/O Mode Control	0x0000_001F
RTC_TAMP1PCTL	RTC_BA+0x128	R/W	TAMPER1 Pin I/O Mode Control	0x0000_001F
RTC_LXTIPCTL	RTC_BA+0x12C	R/W	32K Input Pin I/O Mode Control	0x0000_001F
RTC_LXTOPCTL	RTC_BA+0x130	R/W	32K Output Pin I/O Mode Control	0x0000_001F
RTC_TAMSK	RTC_BA+0x140	R/W	Time Alarm MASK Register	0x0000_0000
RTC_CAMSK	RTC_BA+0x144	R/W	Calendar Alarm MASK Register	0x0000_0000

6.24.6 Register Description

RTC Initiation Register (RTC_INIT)

Register	Offset	R/W	Description	Reset Value
RTC_INIT	RTC_BA+0x00	R/W	RTC Initiation Register	0x0000_0000

31	30	29	28	27	26	25	24
INIT							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
INIT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
INIT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INIT							**INIT[0]/Active**

Bits	Description	
[31:1]	**INIT[31:1]**	**RTC Initiation** When RTC block is powered on, RTC is at reset state. User has to write a number (0xa5eb1357) to INIT to make RTC leaving reset state. Once the INIT is written as 0xa5eb1357, the RTC will be in un-reset state permanently. The INIT is a write-only field and read value will be always “0”.
[0]	**INIT[0]/Active**	**RTC Active Status (Read Only)** 0 = RTC is at reset state. 1 = RTC is at normal active state.

RTC Access Enable Control Register (RTC_RWEN)

Register	Offset	R/W	Description	Reset Value
RTC_RWEN	RTC_BA+0x04	R/W	RTC Access Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							RWENF
15	14	13	12	11	10	9	8
RWENF							
7	6	5	4	3	2	1	0
RWENF							

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**RWENF**	**RTC Register Access Enable Flag (Read Only)** 0 = RTC register read/write Disabled. 1 = RTC register read/write Enabled. This bit will be set after RTC_RWEN[15:0] register is load a 0xA965, and be cleared automatically after 1024 RTC clock.

Register / RTC_RWEN.ENF	1	0
RTC_INIT	R/W	R/W
RTC_RWEN	R/W	R/W
RTC_FREQADJ	R/W	-
RTC_TIME	R/W	R
RTC_CAL	R/W	R
RTC_CLKFMT	R/W	R/W
RTC_WEEKDAY	R/W	R
RTC_TALM	R/W	-
RTC_CALM	R/W	-
RTC_LEAPYEAR	R	R
RTC_INTEN	R/W	R/W
RTC_INTSTS	R/W	R/W
RTC_TICK	R/W	-
RTC_SPRCTL	R/W	-

		RTC_TAMPCTL	R/W	-
		RTC_TAMPSTS	R/W	-
		RTC_LXTGAIN	R/W	-
		SDI0_PADCTL	R/W	-
		SDI1_PADCTL	R/W	-
		RTC_LXTIPCTL	R/W	-
		RTC_LXTOPCTL	R/W	-
		RTC_TAMSK	R/W	-
		RTC_CAMSK	R/W	-
		RTC_SPR0	R/W	-
		RTC_SPR1	R/W	-
		RTC_SPR2	R/W	-
		RTC_SPR3	R/W	-
		RTC_SPR4	R/W	-
		RTC_SPR5	R/W	-
		RTC_SPR6	R/W	-
		RTC_SPR7	R/W	-
		RTC_SPR8	R/W	-
		RTC_SPR9	R/W	-
		RTC_SPR10	R/W	-
		RTC_SPR11	R/W	-
		RTC_SPR12	R/W	-
		RTC_SPR13	R/W	-
		RTC_SPR14	R/W	-
		RTC_SPR15	R/W	-
		RTC_SPR16	R/W	-
		RTC_SPR17	R/W	-
		RTC_SPR18	R/W	-
		RTC_SPR19	R/W	-
		RTC_SPR20	R/W	-
		RTC_SPR21	R/W	-
		RTC_SPR22	R/W	-
		RTC_SPR23	R/W	-
[15:0]	**RWEN**	**RTC Register Access Enable Password (Write Only)** Writing 0xA965 to this register will enable RTC access and keep 1024 RTC clock.		

RTC Frequency Compensation Register (RTC_FREQADJ)

Register	Offset	R/W	Description	Reset Value
RTC_FREQADJ	RTC_BA+0x08	R/W	RTC Frequency Compensation Register	0x0000_0700

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				INTEGER			
7	6	5	4	3	2	1	0
Reserved		FRACTION					

Bits	Description	
[31:12]	Reserved	Reserved.
[11:8]	INTEGER	**Integer Part** Integer part of Detected Value / RTC_FREQADJ[11:8] / Integer part of Detected Value / RTC_FREQADJ[11:8] 32776 / 1111 / 32768 / 0111 32775 / 1110 / 32767 / 0110 32774 / 1101 / 32766 / 0101 32773 / 1100 / 32765 / 0100 32772 / 1011 / 32764 / 0011 32771 / 1010 / 32763 / 0010 32770 / 1001 / 32762 / 0001 32769 / 1000 / 32761 / 0000
[5:0]	FRACTION	**Fraction Part** Formula = (fraction part of detected value) x 60. **Note:** Digit in RTC_FREQADJ must be expressed as hexadecimal number.

Note: This register can be read back after the RTC register access enable bit RWENF (RTC_RWEN[16]) is active.

RTC Time Loading Register (RTC_TIME)

Register	Offset	R/W	Description	Reset Value
RTC_TIME	RTC_BA+0x0C	R/W	Time Loading Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		TENHR		HR			
15	14	13	12	11	10	9	8
Reserved	TENMIN			MIN			
7	6	5	4	3	2	1	0
Reserved	TENSEC			SEC			

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21:20]	**TENHR**	10-Hour Time Digit (0~2)
[19:16]	**HR**	1-Hour Time Digit (0~9)
[15]	**Reserved**	Reserved.
[14:12]	**TENMIN**	10-Min Time Digit (0~5)
[11:8]	**MIN**	1-Min Time Digit (0~9)
[7]	**Reserved**	Reserved.
[6:4]	**TENSEC**	10-Sec Time Digit (0~5)
[3:0]	**SEC**	1-Sec Time Digit (0~9)

Note:

1. RTC_TIME is a BCD digit counter and RTC will not check loaded data.
2. The reasonable value range is listed in the parenthesis.

RTC Calendar Loading Register (RTC_CAL)

Register	Offset	R/W	Description	Reset Value
RTC_CAL	RTC_BA+0x10	R/W	Calendar Loading Register	0x0005_0101

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
TENYEAR				YEAR			
15	14	13	12	11	10	9	8
Reserved			TENMON	MON			
7	6	5	4	3	2	1	0
Reserved		TENDAY		DAY			

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:20]	**TENYEAR**	10-Year Calendar Digit (0~9)
[19:16]	**YEAR**	1-Year Calendar Digit (0~9)
[15:13]	**Reserved**	Reserved.
[12]	**TENMON**	10-Month Calendar Digit (0~1)
[11:8]	**MON**	1-Month Calendar Digit (0~9)
[7:6]	**Reserved**	Reserved.
[5:4]	**TENDAY**	10-Day Calendar Digit (0~3)
[3:0]	**DAY**	1-Day Calendar Digit (0~9)

Note:

1. RTC_CAL is a BCD digit counter and RTC will not check loaded data.
2. The reasonable value range is listed in the parenthesis.

RTC Time Scale Selection Register (RTC_CLKFMT)

Register	Offset	R/W	Description	Reset Value
RTC_CLKFMT	RTC_BA+0x14	R/W	Time Scale Selection Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							24HEN

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**24HEN**	**24-Hour / 12-Hour Time Scale Selection** Indicates that RTC_TIME and RTC_TALM are in 24-hour time scale or 12-hour time scale 0 = 12-hour time scale with AM and PM indication selected. 1 = 24-hour time scale selected.

24-hour time scale	12-hour time scale	24-hour time scale	12-hour time scale (PM time + 20)
00	12(AM12)	12	32(PM12)
01	01 (AM01)	13	21 (PM01)
02	02(AM02)	14	22(PM02)
03	03(AM03)	15	23(PM03)
04	04 (AM04)	16	24 (PM04)
05	05(AM05)	17	25(PM05)
06	06(AM06)	18	26(PM06)
07	07(AM07)	19	27(PM07)
08	08(AM08)	20	28(PM08)
09	09(AM09)	21	29(PM09)
10	10 (AM10)	22	30 (PM10)
11	11 (AM11)	23	31 (PM11)

RTC Day of the Week Register (RTC_WEEKDAY)

Register	Offset	R/W	Description	Reset Value
RTC_WEEKDAY	RTC_BA+0x18	R/W	Day of the Week Register	0x0000_0006

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					WEEKDAY		

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**WEEKDAY**	**Day Of The Week Bits** 0 = Sunday 1 = Monday 2 = Tuesday 3 = Wednesday 4 = Thursday 5 = Friday 6 = Saturday

RTC Time Alarm Register (RTC_TALM)

Register	Offset	R/W	Description	Reset Value
RTC_TALM	RTC_BA+0x1C	R/W	Time Alarm Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		TENHR		HR			
15	14	13	12	11	10	9	8
Reserved	TENMIN			MIN			
7	6	5	4	3	2	1	0
Reserved	TENSEC			SEC			

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21:20]	**TENHR**	10-Hour Time Digit of Alarm Setting (0~2)
[19:16]	**HR**	1-Hour Time Digit of Alarm Setting (0~9)
[15]	**Reserved**	Reserved.
[14:12]	**TENMIN**	10-Min Time Digit of Alarm Setting (0~5)
[11:8]	**MIN**	1-Min Time Digit of Alarm Setting (0~9)
[7]	**Reserved**	Reserved.
[6:4]	**TENSEC**	10-Sec Time Digit of Alarm Setting (0~5)
[3:0]	**SEC**	1-Sec Time Digit of Alarm Setting (0~9)

Note:

1. RTC_TALM is a BCD digit counter and RTC will not check loaded data.
2. The reasonable value range is listed in the parenthesis.
3. This register can be read back after the RTC register access enable bit RWENF (RTC_RWEN[16]) is active.

RTC Calendar Alarm Register (RTC_CALM)

Register	Offset	R/W	Description	Reset Value
RTC_CALM	RTC_BA+0x20	R/W	Calendar Alarm Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
TENYEAR				YEAR			
15	14	13	12	11	10	9	8
Reserved			TENMON	MON			
7	6	5	4	3	2	1	0
Reserved		TENDAY		DAY			

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:20]	**TENYEAR**	10-Year Calendar Digit of Alarm Setting (0~9)
[19:16]	**YEAR**	1-Year Calendar Digit of Alarm Setting (0~9)
[15:13]	**Reserved**	Reserved.
[12]	**TENMON**	10-Month Calendar Digit of Alarm Setting (0~1)
[11:8]	**MON**	1-Month Calendar Digit of Alarm Setting (0~9)
[7:6]	**Reserved**	Reserved.
[5:4]	**TENDAY**	10-Day Calendar Digit of Alarm Setting (0~3)
[3:0]	**DAY**	1-Day Calendar Digit of Alarm Setting (0~9)

Note:

1. RTC_CALM is a BCD digit counter and RTC will not check loaded data.
2. The reasonable value range is listed in the parenthesis.
3. This register can be read back after the RTC register access enable bit RWENF (RTC_RWEN[16]) is active.

RTC Leap Year Indication Register (RTC_LEAPYEAR)

Register	Offset	R/W	Description	Reset Value
RTC_LEAPYEAR	RTC_BA+0x24	R	Leap Year Indication Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**LEAPYEAR**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**LEAPYEAR**	**Leap Year Indicator (Read Only)** 0 = This year is not a leap year. 1 = This year is leap year.

RTC Interrupt Enable Control Register (RTC_INTEN)

Register	Offset	R/W	Description	Reset Value
RTC_INTEN	RTC_BA+0x28	R/W	RTC Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						TICKIEN	ALMIEN

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**TICKIEN**	**Time Tick Interrupt Enable Bit** 0 = RTC Time Tick Interrupt Disabled. 1 = RTC Time Tick Interrupt Enabled.
[0]	**ALMIEN**	**Alarm Interrupt Enable Bit** 0 = RTC Alarm Interrupt Disabled. 1 = RTC Alarm Interrupt Enabled.

RTC Interrupt Indication Register (RTC_INTSTS)

Register	Offset	R/W	Description	Reset Value
RTC_INTSTS	RTC_BA+0x2C	R/W	RTC Interrupt Indicator Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						TICKIF	ALMIF

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**TICKIF**	**RTC Time Tick Interrupt Flag** When RTC Time Tick happened, this bit will be set to 1 and an interrupt will be generated if RTC Tick Interrupt enabled (TICKIEN (RTC_INTEN[1])) is set to 1. Chip will also be waken up if RTC Tick Interrupt is enabled and this bit is set to 1 when chip is running at Power-down mode. **Note:** This bit can be cleared by writing 1 to it.
[0]	**ALMIF**	**RTC Alarm Interrupt Flag** When RTC real time counters RTC_TIME and RTC_CAL reach the alarm setting time registers RTC_TALM and RTC_CALM, this bit will be set to 1 and an interrupt will be generated if RTC Alarm Interrupt enabled (ALMIEN (RTC_INTEN(0)) is set to 1. Chip will also be waken up if RTC Alarm Interrupt is enabled and this bit is set to 1 when chip is running at Power-down mode. **Note:** This bit can be cleared by writing 1 to it.

RTC Time Tick Register (RTC_TICK)

Register	Offset	R/W	Description	Reset Value
RTC_TICK	RTC_BA+0x30	R/W	RTC Time Tick Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					TICKSEL[2:0]		

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**TICKSEL**	**Time Tick Bits** The RTC time tick period for Periodic Time Tick Interrupt request. **Note:** These bits can be read back after the RTC register access enable bit RWENF (RTC_RWEN[16]) is active.

TICKSEL[2:0]	Time tick (second)
0	1
1	1/2
2	1/4
3	1/8
4	1/16
5	1/32
6	1/64
7	1/128

RTC Spare Functional Control Register (RTC_SPRCTL)

Register	Offset	R/W	Description	Reset Value
RTC_SPRCTL	RTC_BA+0x3C	R/W	RTC Spare Functional Control Register	0x0000_0080

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
SPRRWRDY	Reserved				SPRRWEN	Reserved	

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**SPRRWRDY**	**SPR Register Ready** This bit indicates if the registers RTC_SPRCTL, RTC_SPR0 ~ RTC_SPR23 are ready to be accessed. After CPU writing registers RTC_SPRCTL, RTC_SPR0 ~ RTC_SPR23, polling this bit to check if these registers are updated done is necessary. This bit is read only and any write to it won't take any effect. 0 = RTC_SPRCTL, RTC_SPR0 ~ RTC_SPR23 updating is in progress. 1 = RTC_SPRCTL, RTC_SPR0 ~ RTC_SPR23 are updated done and ready to be accessed.
[6:3]	**Reserved**	Reserved.
[2]	**SPRRWEN**	**SPR Register Enable Bit** This bit controls the spare register to be enabled or not. 0 = Spare register Disabled and RTC_SPR0 ~ RTC_SPR23 cannot be accessed. 1 = Spare register Enabled and RTC_SPR0 ~ RTC_SPR23 can be accessed.
[1:0]	**Reserved**	Reserved.

RTC Spare Register (SPRx)

Register	Offset	R/W	Description	Reset Value
RTC_SPR0	RTC_BA+0x40	R/W	RTC Spare Register 0	0x0000_0000
RTC_SPR1	RTC_BA+0x44	R/W	RTC Spare Register 1	0x0000_0000
RTC_SPR2	RTC_BA+0x48	R/W	RTC Spare Register 2	0x0000_0000
RTC_SPR3	RTC_BA+0x4C	R/W	RTC Spare Register 3	0x0000_0000
RTC_SPR4	RTC_BA+0x50	R/W	RTC Spare Register 4	0x0000_0000
RTC_SPR5	RTC_BA+0x54	R/W	RTC Spare Register 5	0x0000_0000
RTC_SPR6	RTC_BA+0x58	R/W	RTC Spare Register 6	0x0000_0000
RTC_SPR7	RTC_BA+0x5C	R/W	RTC Spare Register 7	0x0000_0000
RTC_SPR8	RTC_BA+0x60	R/W	RTC Spare Register 8	0x0000_0000
RTC_SPR9	RTC_BA+0x64	R/W	RTC Spare Register 9	0x0000_0000
RTC_SPR10	RTC_BA+0x68	R/W	RTC Spare Register 10	0x0000_0000
RTC_SPR11	RTC_BA+0x6C	R/W	RTC Spare Register 11	0x0000_0000
RTC_SPR12	RTC_BA+0x70	R/W	RTC Spare Register 12	0x0000_0000
RTC_SPR13	RTC_BA+0x74	R/W	RTC Spare Register 13	0x0000_0000
RTC_SPR14	RTC_BA+0x78	R/W	RTC Spare Register 14	0x0000_0000
RTC_SPR15	RTC_BA+0x7C	R/W	RTC Spare Register 15	0x0000_0000
RTC_SPR16	RTC_BA+0x80	R/W	RTC Spare Register 16	0x0000_0000
RTC_SPR17	RTC_BA+0x84	R/W	RTC Spare Register 17	0x0000_0000
RTC_SPR18	RTC_BA+0x88	R/W	RTC Spare Register 18	0x0000_0000
RTC_SPR19	RTC_BA+0x8C	R/W	RTC Spare Register 19	0x0000_0000
RTC_SPR20	RTC_BA+0x90	R/W	RTC Spare Register 20	0x0000_0000
RTC_SPR21	RTC_BA+0x94	R/W	RTC Spare Register 21	0x0000_0000
RTC_SPR22	RTC_BA+0x98	R/W	RTC Spare Register 22	0x0000_0000
RTC_SPR23	RTC_BA+0x9C	R/W	RTC Spare Register 23	0x0000_0000

31	30	29	28	27	26	25	24
SPARE[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SPARE[23:16]							

15	14	13	12	11	10	9	8
SPARE[15:8]							
7	6	5	4	3	2	1	0
SPARE[7:0]							

Bits	Description	
[31:0]	**SPARE**	**SPARE Bits** This field is used to store back-up information defined by software. This field will be cleared by hardware automatically once a snooper pin event is detected. Before storing back-up information in to SPARE register, software should write 0xA965 to RTC_RWEN to make sure register read/write enabled.

Note: RTC_SPR23[31] is the control bit to achieve the control signal which would be parking at the V_{BAT} control signal while the system re-power up period. When system initialization is completed, then software can write that control bit to switch the control signal back to core power area setting. So, those GPIO pad can be fully controlled by application in the whole power-down & re-power up period.

RTC TAMPER Control Register (RTC_TAMPCTL)

Register	Offset	R/W	Description	Reset Value
RTC_TAMPCTL	RTC_BA+0x110	R/W	Tamper Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
TAMPLV1	TAMPLV0	TAMPDBEN1	TAMPDBEN0	TAMPEN1	TAMPEN0	DESTROYEN	TIEN

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**TAMPLV1**	**Tamper1 Level** 0 = Low. 1 = High.
[6]	**TAMPLV0**	**Tamper0 Level** 0 = Low. 1 = High.
[5]	**TAMPDBEN1**	**Tamper1 De-Bounce Enable Bit** 0 = Disabled. 1 = Enabled.
[4]	**TAMPDBEN0**	**Tamper0 De-Bounce Enable Bit** 0 = Disabled. 1 = Enabled.
[3]	**TAMPEN1**	**Tamper1 Detect Enable Bit** 0 = Disabled. 1 = Enabled.
[2]	**TAMPEN0**	**Tamper0 Detect Enable Bit** 0 = Disabled. 1 = Enabled.
[1]	**DESTROYEN**	**Destroy Spare Register Enable Bit** 0 = Disabled. 1 = Enabled.
[0]	**TIEN**	**Tamper Interrupt Enable Bit**

		0 = Tamper interrupt Disabled. 1 = Tamper interrupt Enabled.

RTC TAMPER Status Control Register (RTC_TAMPSTS)

Register	Offset	R/W	Description	Reset Value
RTC_TAMPSTS	RTC_BA+0x114	R/W	Tamper Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						TAMPSTS1	TAMPSTS0

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**TAMPSTS1**	**Tamper1 Sense Flag** 0 = No invasion. 1 = Tamper1 detect invasion. **Note:** Write 1 to clear it
[0]	**TAMPSTS0**	**Tamper0 Sense Flag** 0 = No invasion. 1 = Tamper0 detect invasion. **Note:** Write 1 to clear it

RTC IOPAD Control Register (RTC_TAMP0PCTL)

Register	Offset	R/W	Description	Reset Value
RTC_TAMP0PCTL	RTC_BA+0x124	R/W	TAMPER0 Pin I/O Mode Control Register	0x0000_001F
RTC_TAMP1PCTL	RTC_BA+0x128	R/W	TAMPER1 Pin I/O Mode Control Register	0x0000_001F
RTC_LXTIPCTL	RTC_BA+0x12C	R/W	32K Input Pin I/O Mode Control Register	0x0000_001F
RTC_LXTOPCTL	RTC_BA+0x130	R/W	32K Output Pin I/O Mode Control Register	0x0000_001F

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**DINOFF**	**TYPE**	**TRIEN**	**OUTEN**	**OUTLV**

Bits	Description	
[31:4]	**Reserved**	Reserved.
[4]	**DINOFF**	**Off Digital** 0 = Off digital Disabled. 1 = Off digital Enabled.
[3]	**TYPE**	**Type** 0 = Input Schmitt Trigger function Disabled. 1 = Input Schmitt Trigger function Enabled.
[2]	**TRIEN**	**Tri-State** 0 = Tri-state Disabled. 1 = Tri-state Enabled.
[1]	**OUTEN**	**Output Enable Bit** 0 = Output Enabled. 1 = Output Disabled.
[0]	**OUTLV**	**Output Level** 0 = Low. 1 = High.

RTC Time Alarm Register MASK (RTC_TAMSK)

Register	Offset	R/W	Description	Reset Value
RTC_TAMSK	RTC_BA+0x140	R/W	Time Alarm MASK Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		MTENHR	MHR	MTENMIN	MMIN	MTENSEC	MSEC

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**MTENHR**	Mask 10-Hour Time Digit of Alarm Setting (0~2)
[4]	**MHR**	Mask 1-Hour Time Digit of Alarm Setting (0~9)
[3]	**MTENMIN**	Mask 10-Min Time Digit of Alarm Setting (0~5)
[2]	**MMIN**	Mask 1-Min Time Digit of Alarm Setting (0~9)
[1]	**MTENSEC**	Mask 10-Sec Time Digit of Alarm Setting (0~5)
[0]	**MSEC**	Mask 1-Sec Time Digit of Alarm Setting (0~9)

Note:

1. RTC_TALM is a BCD digit counter and RTC will not check loaded data.
2. The reasonable value range is listed in the parenthesis.
3. This register can be read back after the RTC register access enable bit RWENF (RTC_RWEN[16]) is active.

RTC Calendar Alarm Mask Register (RTC_CAMSK)

Register	Offset	R/W	Description	Reset Value
RTC_CAMSK	RTC_BA+0x144	R/W	Calendar Alarm Mask Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		MTENYEAR	MYEAR	MTENMON	MMON	MTENDAY	MDAY

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**MTENYEAR**	Mask 10-Year Calendar Digit of Alarm Setting (0~9)
[4]	**MYEAR**	Mask 1-Year Calendar Digit of Alarm Setting (0~9)
[3]	**MTENMON**	Mask 10-Month Calendar Digit of Alarm Setting (0~1)
[2]	**MMON**	Mask 1-Month Calendar Digit of Alarm Setting (0~9)
[1]	**MTENDAY**	Mask 10-Day Calendar Digit of Alarm Setting (0~3)
[0]	**MDAY**	Mask 1-Day Calendar Digit of Alarm Setting (0~9)

Note:

1. RTC_CALM is a BCD digit counter and RTC will not check loaded data.
2. The reasonable value range is listed in the parenthesis.
3. This register can be read back after the RTC register access enable bit RWENF (RTC_RWEN[16]) is active.

6.25 Smart Card Host Interface (SC)

6.25.1 Overview

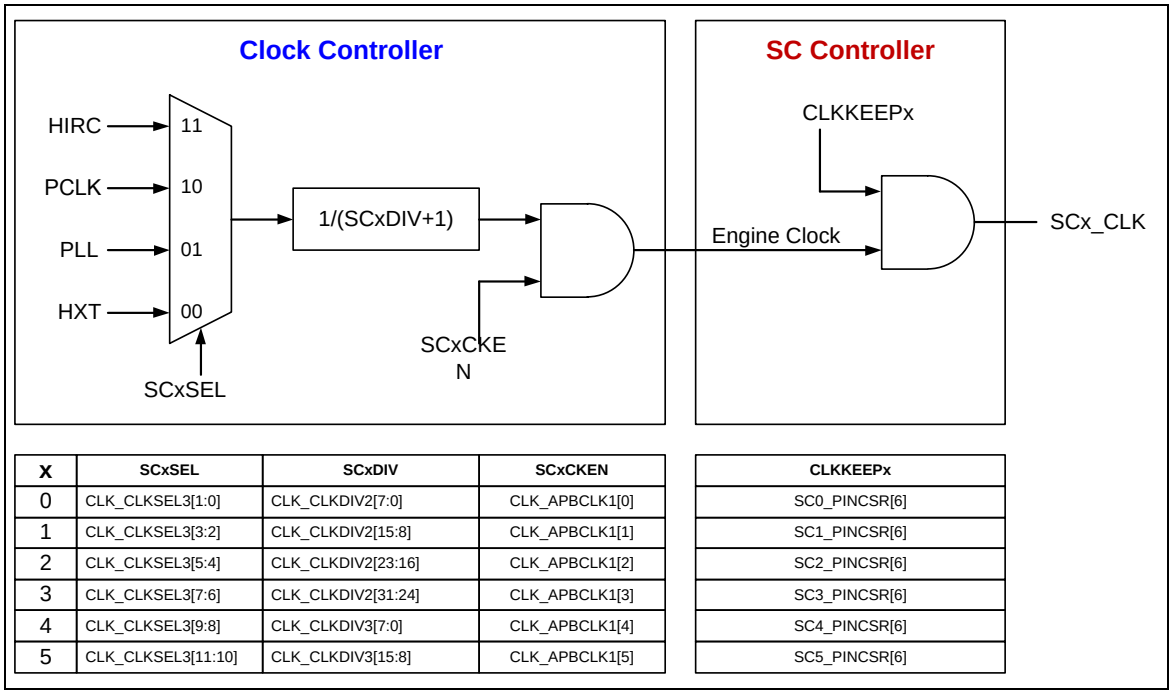
The Smart Card Interface controller (SC controller) is based on ISO/IEC 7816-3 standard and fully compliant with PC/SC Specifications. It also provides status of card insertion/removal.

6.25.2 Features

- ISO-7816-3 T = 0, T = 1 compliant.
- EMV2000 compliant
- Up to six ISO-7816-3 ports
- Separates receive/transmit 4 byte entry FIFO for data payloads.
- Programmable transmission clock frequency.
- Programmable receiver buffer trigger level.
- Programmable guard time selection (11 ETU ~ 267 ETU).
- A 24-bit and two 8 bit counters for Answer to Request (ATR) and waiting times processing.
- Supports auto inverse convention function.
- Supports transmitter and receiver error retry and error number limitation function.
- Supports hardware activation sequence process.
- Supports hardware warm reset sequence process.
- Supports hardware deactivation sequence process.
- Supports hardware auto deactivation sequence when detected the card removal.
- Supports UART mode
  - Full duplex, asynchronous communications.
  - Separates receiving / transmitting 4 bytes entry FIFO for data payloads.
  - Supports programmable baud rate generator for each channel.
  - Supports programmable receiver buffer trigger level.
  - Programmable transmitting data delay time between the last stop bit leaving the TX-FIFO and the de-assertion by setting SC_EGT register.
  - Programmable even, odd or no parity bit generation and detection.
  - Programmable stop bit, 1 or 2 stop bit generation

6.25.3 Block Diagram

The SC clock control and block diagram are shown as follows. The SC controller is completely asynchronous design with to clock domains, PCLK and engine clock, note that the PCLK should be higher or equal than the frequency of engine clock.



x	SCxSEL	SCxDIV	SCxCKEN	CLKKEEPx
0	CLK_CLKSEL3[1:0]	CLK_CLKDIV2[7:0]	CLK_APBCLK1[0]	SC0_PINCSR[6]
1	CLK_CLKSEL3[3:2]	CLK_CLKDIV2[15:8]	CLK_APBCLK1[1]	SC1_PINCSR[6]
2	CLK_CLKSEL3[5:4]	CLK_CLKDIV2[23:16]	CLK_APBCLK1[2]	SC2_PINCSR[6]
3	CLK_CLKSEL3[7:6]	CLK_CLKDIV2[31:24]	CLK_APBCLK1[3]	SC3_PINCSR[6]
4	CLK_CLKSEL3[9:8]	CLK_CLKDIV3[7:0]	CLK_APBCLK1[4]	SC4_PINCSR[6]
5	CLK_CLKSEL3[11:10]	CLK_CLKDIV3[15:8]	CLK_APBCLK1[5]	SC5_PINCSR[6]

Figure 6.25-1 SC Clock Control Diagram (4-bit Prescale Counter in Clock Controller)

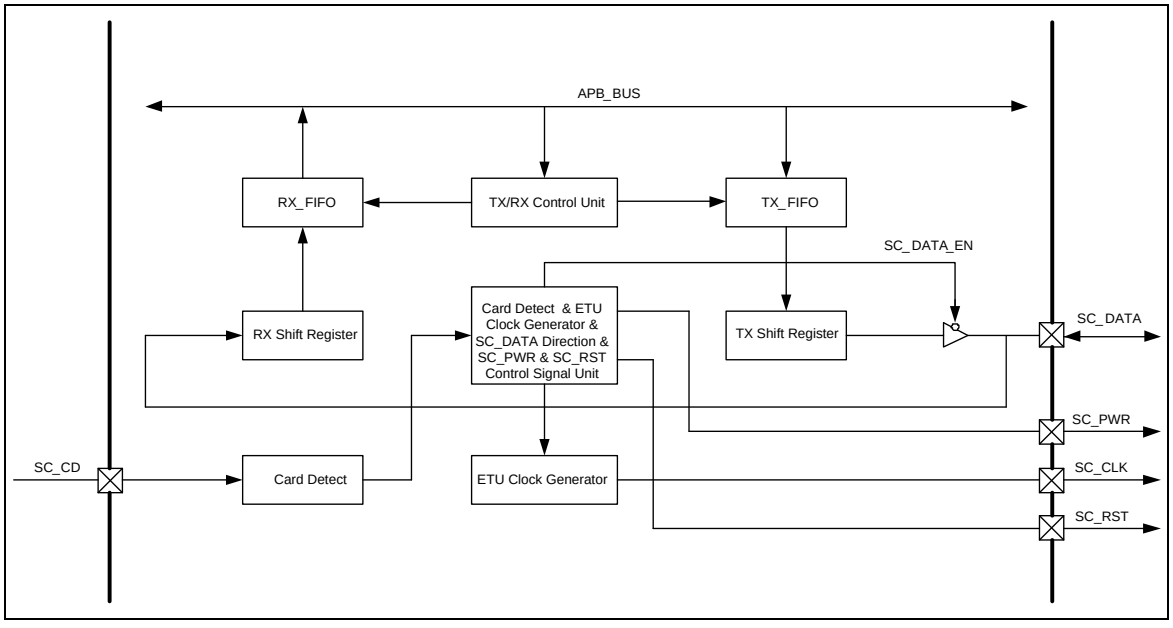


Figure 6.25-2 SC Controller Block Diagram

6.25.4 Functional description

Basically, the smart card interface acts as a half-duplex asynchronous communication port and its data format is composed of ten consecutive bits which is show as follows.

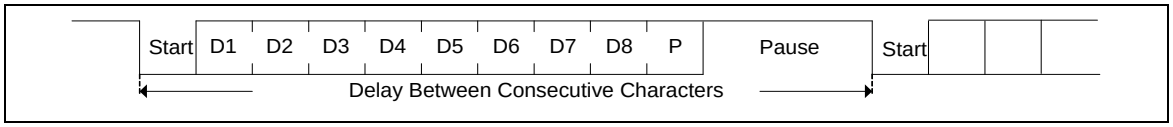


Figure 6.25-3 SC Data Character

6.25.4.1 Activation, Warm Reset and Deactivation Sequence

Activation

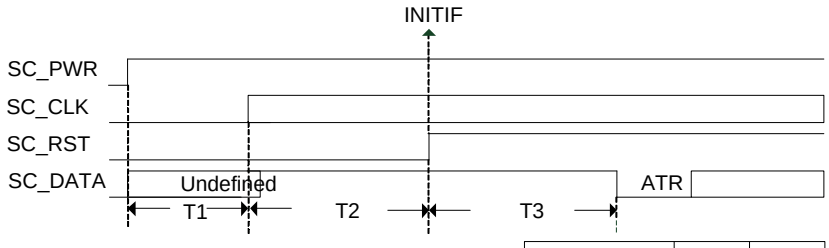
The Smart Card Interface controller supports hardware activation, warm reset and deactivation sequence. The activation sequence is show as Follows.

- Set SCRST to low
- Set SC_PWR at high level and SC_DAT at high level (reception mode) .
- Enable SC_CLK clock
- De-assert SCRST to high

The activation sequence can be controlled by software or hardware. If software wants to control it, software can control SC_PINCTL and SC_TMRx register to process the activation sequence or setting ACTEN(SC_ALTCTL[3]) register, and then the interface will perform the hardware activation sequence.

Following is activation control sequence in hardware activation mode:

- Set activation timing by setting INITSEL(SC_ALTCTL[9:8]).
- TMR0 can be selected when TMRSEL(SC_CTL[14:13]) is 01, 10 or 11.
- Set operation mode OPMODE(SC_TMRCTL0[27:24]) to 0011 and give an Answer to Request value by setting CNT (SC_TMRCTL0[23:0]) register.
- When hardware de-asserts SCRST to high, hardware will generator an interrupt INITIF to CPU at the same time INITIEN(SC_INTEN[8]) = '1'
- If the TMR0 decreases the counter to "1" (start from SCRST) and the card does not response ATR before that time, hardware will generate interrupt TMR0IF (SC_INTSTS[3]) to CPU.



Time	Comment
T1	SC_PWR to SC_CLK Start
T2	SC_CLK Start to SC_RST Assert
T3	SC_CLK Start to ATR Appear

INIT_SEL (SC_ ALTCT[9:8])	T1	T2
00	85	489
01	133	537
10	165	569
11	165	42060

Unit: SC Clock

Note : The values are measured by chip I/O pin and the real value will depend on system design

Figure 6.25-4 SC Activation Sequence

Warm Reset

The warm reset sequence is showed as follows.

- Set SC_RST to low by programming RSTSTS (SC_PINCTL[18]) to '0'.
- Set SC_DAT to high by programming DATSTS (SC_PINCTL[16]) to '1'.
- Set SC_RST to high by programming RSTSTS (SC_PINCTL[18]) to '1'.

The warm reset sequence can be controlled by software or hardware. If software wants to control it, software can control SC_PINCTL and SC_TMRx register to process the warm reset sequence or set WARSTEN(SC_ALTCTL[4] register, and then the interface will perform the hardware warm reset sequence.

Following is warm reset control sequence in hardware warm reset mode

- Set warm reset timing by setting INITSEL(SC_ALTCTL[9:8]).
- Select TMR0 by setting TMRSEL(SC_CTL[14:13]) register (TMRSEL can be 01, 10, or 11).
- Set operation mode OPMODE(SC_TMRCTL0[27:24]) to 011 and give an Answer to Request value by setting CNT(SC_TMRCTL0[23:0]) register.
- Set CNTEN0(SC_ALTCTL[5]) and WARSTEN(SC_ALTCTL[4]) to start counting.
- When hardware de-asserts SCRST to high, hardware will generate an interrupt INTIF(SC_INTSTS[8]) to CPU at the same time (INITIEN(SC_INTEN[8] = '1')
- If the TMR0 decrease the counter to "1" (start from SC_RST) and the card does not response ATR before that time, hardware will generate interrupt TMR0IF (SC_INTSTS[3]) to CPU.

Time	Comment
T4	SC_RST to SC_DATA Reception Mode
T5	SC_DATA Reception Mode to SC_RST Assert
T6	SC_CLK Start to ATR Appear

INIT_SEL (SC_ALTCTL[9:8])	T4	T5
00	81	483
01	129	531
10	161	563
11	161	42106

Unit: SC Clock

Note: This value is measured by chip IO pin and the real value will depend on system design

Figure 6.25-5 SC Warm Reset Sequence

Deactivation

The deactivation sequence is shown as follows:

- Set SCRST to low.
- Stop SC_CLK.
- Set SC_DATA to state low.
- Deactivated SC_PWR.

The deactivation sequence can be controlled by software or hardware. If software wants to control

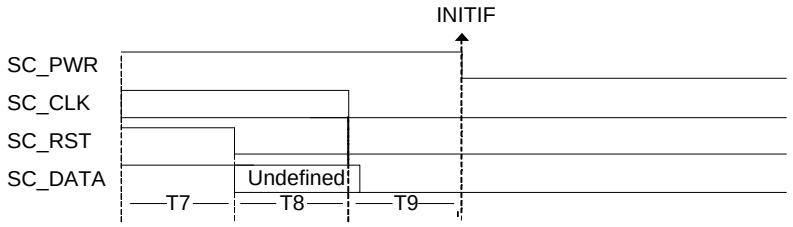
it, software can control SC_PINCTL and SC_TMRCTL0 register to process the deactivation sequence or set DACTEN(SC_ALTCTL[2]) register, and then the interface will perform the hardware deactivation sequence.

The SC controller also supports auto deactivation sequence when the card removal detection is set ADACEN(SC_ALTCTL[11])

Following is deactivation control sequence in hardware deactivation mode:

Set deactivation timing by setting INITSEL(SC_ALTCTL[9:8]). Set DACTEN(SC_ALTCTL[2]) to start counting by SC_ALTCTL register.

When hardware de-asserts SC_PWR to low, the controller will generate an interrupt to CPU at the same time (if INITIEN(SC_IER[8] = 1)



Time	Comment
T7	Deactivation Trigger to SC_RST Low
T8	SMC_RST Low to Stop SC_CLK
T9	Stop SC_CLK to Stop SC_PWR

INITSEL (SC_ALTCTL[9:8])	T7	T8	T9
00	97	83	87
01	145	131	135
10	177	163	167
11	177	163	167

Unit: SC Clock

Figure 6.25-6 SC Deactivation Sequence

6.25.4.2 Initial Character TS

According to 7816-3, the initial character TS of answer to According to 7816-3 T=0 mode description, request (ATR) has two possible patterns (as shown in the following figure). If the TS pattern is 1100_0000, it is inverse convention. When decoded by inverse convention, the conveyed byte is equal to '3F'. If the TS pattern is 1101_1100, it is direct convention. When decoded by direct convention, the conveyed byte is equal to '3B'.Software can set AUTOCEN(SC_CTL[3]) and then the operating convention will be decided by hardware. Software can also set the CONSEL(SC_CTL[5:4]) register (set to 00 or 11) to change the operating convention after SC received TS of answer to request (ATR).

If software enables auto convention function by setting AUTOCEN(SC_CTL[3])register, the setting step must be done before Answer to Request state and the first data must be 0x3B or 0x3F. After hardware received first data and stored it at buffer, the hardware will decided the convention and change the CONSEL(SC_CTL[5:4]) register automatically. If the first data is neither 0x3B nor 0x3F, the hardware will generate an interrupt ACERRIF (if ACERRIEN (SC_INTEN [10])= "1") to CPU.

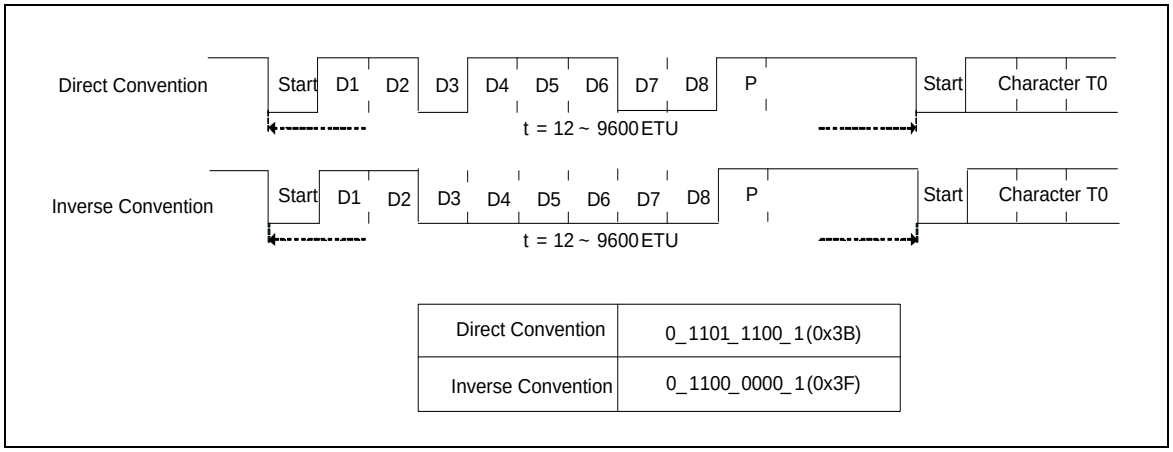


Figure 6.25-7 Initial Character TS

6.25.4.3 Error Signal and Character Repetition

According to ISO7816-3 T=0 mode description, as shown in following, if the receiver receives a wrong parity bit, it will pull the SC_DAT to low by 1.5 bit period to inform the transmitter parity error. Then the transmitter will retransmit the character. The SC interface controller supports hardware error detection function in receiver and supports hardware re-transmit function in transmitter. Software can enable re-transmit function by setting TXRTYEN(SC_CTL[23]). Software can also define the retry (re-transmit) number limitation in TXRTY(SC_CTL[22:20]) register. The re-transmit number is up to TXRTY +1 and if the re-transmit number is equal to TXRTY +1, TXOVERR flag will be set by hardware and if TERRIEN (SC_INTEN [2], SC controller will generate a transfer error interrupt to CPU. Software can also define the received retry number limitation in RXRTY(SC_CTL[18:16]) register. The receiver retry number is up to RXRTY +1, if the number of received errors by receiver is equal to RXRTY +1, receiver will receive this error data to buffer and RXOVERR flag will be set by hardware and if TERRIEN(SC_INTEN[2]), SC controller will generate a transfer error interrupt to CPU.

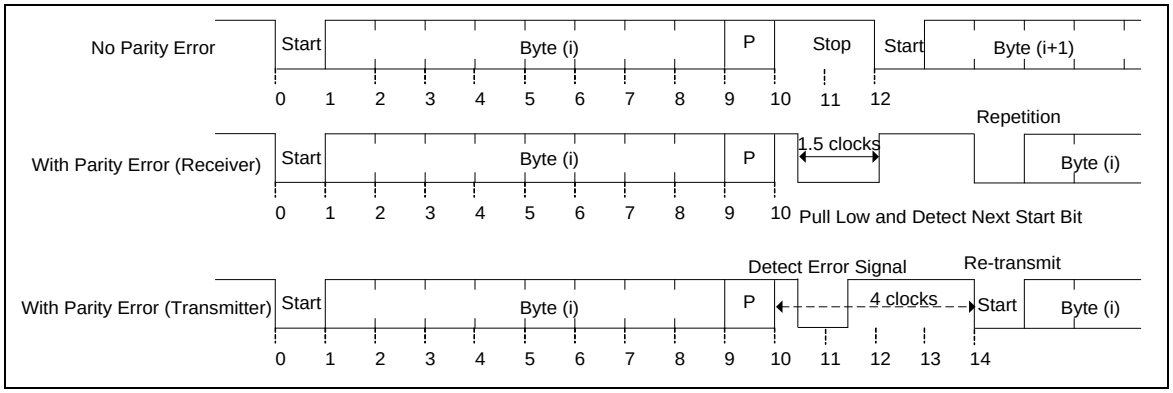


Figure 6.25-8 SC Error Signal

6.25.4.4 Internal Time-out Counter

The smart card interface includes a 24-bit time-out counter and two 8 bit time-out counters. These counters help the controller in processing different real-time interval (ATR, WWT, BWT, etc.). Each counter can be set to start counting once the trigger enable bit has been written or a START bit has been detected.

The following is the programming flow:

Enable counter by setting TMRSEL (SC_CTL[14:13]). Select operation mode OPMODE (SC_TMRCTLx[27:24]) and give a count value CNT(SC_TMRCTLx[23:0]). Set CNTEN0 (SC_ALTCTL [5], CNTEN1 (SC_ALTCTL [6] or CNTEN2 (SC_ALTCTL [7]) is to start counting.

The SC_TMRCTL0, SC_TMRCTL1 and SC_TMRCTL2 timer operation mode are listed below table

Note: Only SC_TMRCTL0 supports mode 0011.

OPMODE	Operation Description	
0000	The down counter started when TMRx_SEN (SC_ALTCTL[7:5]) enabled and ended when counter time-out. The time-out value will be CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0])+1	
	Start	Start counting when TMRx_SEN (SC_ALTCTL[7:5]) enabled
	End	When the down counter equals to 0, hardware will set TMRx_IS(SC_INTSTS[5:3]) and clear TMRx_SEN (SC_ALTCTL[7:5]) automatically.
0001	The down counter started when the first START bit (reception or transmission) detected and ended when counter time-out. The time-out value will be CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0])+1.	
	Start	Start counting when the first START bit (reception or transmission) detected after TMRx_SEN (SC_ALTCTL[7:5]) set to 1.
	End	When the down counter equals to 0, hardware will set TMRx_IS(SC_INTSTS[5:3]) and clear TMRx_SEN (SC_ALTCTL[7:5]) automatically.
0010	The down counter started when the first START bit (reception) detected and ended when counter time-out . The time-out value will be CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0])+1.	
	Start	Start counting when the first START bit (reception) detected bit after TMRx_SEN (SC_ALTCTL[7:5]) set to 1.
	End	When the down counter equals to 0, hardware will set TMRx_IS(SC_INTSTS[5:3]) and clear TMRx_SEN (SC_ALTCTL[7:5]) automatically.
0011	The down counter is only used for hardware activation, warm reset sequence to measure ATR timing. The timing starts when SC_RST de-assertion and ends when ATR response received or time-out. If the counter decreases to 0 before ATR response received, hardware will generate an interrupt to CPU. The time-out value will be CNT (SC_TMRCTL0[23:0])+1.	
	Start	Start counting when SC_RST de-assertion after CNTEN0 (SC_ALTCTL[5]) set to 1. It is used for hardware activation, warm reset mode.
	End	When the down counter equals to 0 before ATR response received, hardware will set TMR0IF and clear CNTEN0 (SC_ALTCTL[5]) automatically. When ATR received and down counter does not equal to 0, hardware will clear CNTEN0 (SC_ALTCTL[5]) automatically.
0100	Same as 0000, but when the down counter equals to 0, hardware will set TMRx_IS(SC_INTSTS[5:3]) and counter will re-load the CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) value and re-count until software clears TMRx_SEN (SC_ALTCTL[7:5]). When TMRx_ATV (SC_ALTCTL[15:13]) =1, software can change CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) value at any time. When the down counter equals to 0, counter will reload the new value of CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) and re-count. The time-out value will be CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0])+1.	
0101	Same as 0001, but when the down counter equals to 0, hardware will set TMRx_IS(SC_INTSTS[5:3]) and	

	counter will re-load the CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) value. When the next START bit is detected, counter will re-count until software clears TMRx_ATV (SC_ALTCTL[15:13]). When TMRx_ATV (SC_ALTCTL[15:13]) =1 software can change CNT (SC_TMRCTL0[23:0], SC_TMRCTL0[7:0], SC_TMRCTL0[7:0]) value at any time. When the down counter equal to 0, it will reload the new value of CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) and re-counting. The time-out value will be CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0])+1.	
0110	Same as 0010, but when the down counter equals to 0, it will set TMRx_IS(SC_INTSTS[5:3]) and counter will re-load the CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) value. When the next START bit is detected, counter will re-count until software clears TMRx_SEN (SC_ALTCTL[7:5]). When TMRx_ATV (SC_ALTCTL[15:13]) =1, software can change CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) value at any time. When the down counter equals to 0, counter will reload the new value of CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) and re-count. The time-out value will be CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0])+1.	
0111	The down counter started when the first START bit (reception or transmission) detected and ended when software clears TMRx_SEN (SC_ALTCTL[7:5]) bit. If next START bit detected, counter will reload the new value of CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) and re-counting. If the counter decreases to 0 before the next START bit detected, hardware will generate an interrupt to CPU. The time-out value will be CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0])+1.	
	Start	Start counting when the first START bit detected after TMRx_SEN (SC_ALTCTL[7:5]) set to 1.
	End	Stop counting after TMRx_SEN (SC_ALTCTL[7:5]) set to 0.
1000	The up counter starts when TMRx_SEN (SC_ALTCTL[7:5]) enabled and ends when TMRx_SEN (SC_ALTCTL[7:5]) disabled. This count value will be stored in CNTx(SC_TMRDAT0 [23:0], SC_TMRDAT1_2[7:0], SC_TMRDAT1_2[15:8). In this mode, hardware cannot generate any interrupt to CPU. The real count value will be CNTx(SC_TMRDAT0 [23:0], SC_TMRDAT1_2[7:0], SC_TMRDAT1_2[15:8]) +1.	
	Start	Start counting after TMRx_SEN (SC_ALTCTL[7:5]) set to 1, and the start count value is 0 (hardware will ignore CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) value).
	End	Stop counting after TMRx_SEN (SC_ALTCTL[7:5]) set to 0 and the value stored to CNTx(SC_TMRDAT0 [23:0], SC_TMRDAT1_2[7:0], SC_TMRDAT1_2[15:8) register.
1111	Down counter starts when software set TMRx_SEN (SC_ALTCTL[7:5]) bit or any START bit been detected and ends when software clears TMRx_SEN (SC_ALTCTL[7:5]) bit. If next START bit detected, counter will reload the new value of CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) and re-counting. If the counter decreases to “0” before the next START bit be detected, hardware will generate an interrupt to CPU. The time-out value will be CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0])+1.	
	Start	Start count when the TMRx_SEN (SC_ALTCTL[7:5]) set to “1” or any START bit (TMRx_SEN (SC_ALTCTL[7:5]) must be set) be detected
	End	Stop count after TMRx_SEN (SC_ALTCTL[7:5]) set to “0”.

6.25.4.5 UART Mode

When the UARTEN(SC_UARTCTL[0]) bit set, the Smart Card Interface controller can also be used as base UART function. The following is the program example for UART mode.

Program example:

1. Software can entry UART mode by setting UARTEN(SC_UARTCTL[0]) bit.
2. Do software reset by setting RXRST(SC_ALTCTL[1]) and TXRST(SC_ALTCTL[0])

SC_ALTCTL[TXRST] bit to ensure that all state machine return idle state.

3. Filled “0” to CONSEL(SC_CTL[5:4]) and AUTOCEN (SC_CTL[3]) field. (In UART mode, those fields must be “0”)
4. Select the UART baud rate by setting ETURDIV (SC_ETUCTL[11:0]) fields.
5. Select the data format include data length (by setting WLS(SC_UARTCTL[5:4]), parity format (by setting OPE(SC_UARTCTL[7]) and PBOFF(SC_UARTCTL[6])) and stop bit length (by setting NSB(SC_CTL[15] or EGT(SC_EGT[7:0])).
6. Select the receiver buffer trigger level by setting RXTRGLV(SC_CTL[7:6]) field and select the receiver buffer time-out value by setting RFTMR (SC_RXTOUT[8:0] field.
7. Write SC_DAT (TX) register or read SC_DAT (RX) register can perform UART function.

6.25.5 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
SC Base Address: **SCx_BA = 0x4009_0000 + (0x1000 * x)** **x = 0,1..5**				
SC_DAT	SCx_BA+0x00	R/W	SC Receive and Transmit Buffer Register	Undefined
SC_CTL	SCx_BA+0x04	R/W	SC Control Register	0x0000_0000
SC_ALTCTL	SCx_BA+0x08	R/W	SC Alternate Control Register	0x0000_0000
SC_EGT	SCx_BA+0x0C	R/W	SC Extend Guard Time Register	0x0000_0000
SC_RXTOUT	SCx_BA+0x10	R/W	SC Receive Buffer Time-out Register	0x0000_0000
SC_ETUCTL	SCx_BA+0x14	R/W	SC ETU Control Register	0x0000_0173
SC_INTEN	SCx_BA+0x18	R/W	SC Interrupt Enable Control Register	0x0000_0000
SC_INTSTS	SCx_BA+0x1C	R/W	SC Interrupt Status Register	0x0000_0002
SC_STATUS	SCx_BA+0x20	R/W	SC Status Register	0x0000_0202
SC_PINCTL	SCx_BA+0x24	R/W	SC Pin Control State Register	0x0000_00x0
SC_TMRCTL0	SCx_BA+0x28	R/W	SC Internal Timer Control Register 0	0x0000_0000
SC_TMRCTL1	SCx_BA+0x2C	R/W	SC Internal Timer Control Register 1	0x0000_0000
SC_TMRCTL2	SCx_BA+0x30	R/W	SC Internal Timer Control Register 2	0x0000_0000
SC_UARTCTL	SCx_BA + 0x34	R/W	SC UART Mode Control Register	0x0000_0000
SC_TMRDAT0	SCx_BA+0x38	R	SC Timer 0 Current Data Register	0x0000_07FF
SC_TMRDAT1_2	SCx_BA+0x3C	R	SC Timer 1 and 2 Current Data Register	0x0000_7F7F

6.25.6 Register Description

SC Receive and Transmit Buffer Register (SC_DAT)

Register	Offset	R/W	Description	Reset Value
SC_DAT	SCx_BA+0x00	R/W	SC Receive and Transmit Buffer Register	Undefined

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DAT							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**DAT**	**Receiving/ Transmit Buffer** Write Operation: By writing data to DAT, the SC will send out an 8-bit data. **Note:** If SCEN(SC_CTL[0]) is not enabled, DAT cannot be programmed. Read Operation: By reading DAT, the SC will return an 8-bit received data.

SC Control Register (SC_CTL)

Register	Offset	R/W	Description	Reset Value
SC_CTL	SCx_BA+0x04	R/W	SC Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	SYNC	Reserved			CDLV	CDDBSEL	
23	22	21	20	19	18	17	16
TXRTYEN	TXRTY			RXRTYEN	RXRTY		
15	14	13	12	11	10	9	8
NSB	TMRSEL		BGT				
7	6	5	4	3	2	1	0
RXTRGLV		CONSEL		AUTOCEN	TXOFF	RXOFF	SCEN

Bits	Description	
[31]	**Reserved**	Reserved.
[30]	**SYNC**	**SYNC Flag Indicator** Due to synchronization, software should check this bit before writing a new value to RXRTY and TXRTY. 0 = synchronizing is completion, user can write new data to SC_PINCTL register. 1 = Last value is synchronizing. **Note:** This bit is read only.
[29:27]	**Reserved**	Reserved.
[26]	**CDLV**	**Card Detect Level** 0 = When hardware detects the card detect pin from high to low, it indicates a card is detected. 1 = When hardware detects the card detect pin from low to high, it indicates a card is detected.  **Note:** Software must select card detect level before Smart Card engine enabled.
[25:24]	**CDDBSEL**	**Card Detect De-Bounce Selection** This field indicates the card detect de-bounce selection. 00 = De-bounce sample card insert once per 384 (128 * 3) engine clocks and de-bounce sample card removal once per 128 engine clocks. 01 = De-bounce sample card insert once per 192 (64 * 3) engine clocks and de-bounce sample card removal once per 64 engine clocks. 10 = De-bounce sample card insert once per 96 (32 * 3) engine clocks and de-bounce sample card removal once per 32 engine clocks. 11 = De-bounce sample card insert once per 48 (16 * 3) engine clocks and de-bounce

		sample card removal once per 16 engine clocks.
[23]	**TXRTYEN**	**TX Error Retry Enable Bit** This bit enables transmitter retry function when parity error has occurred. 0 = TX error retry function Disabled. 1 = TX error retry function Enabled.
[22:20]	**TXRTY**	**TX Error Retry Count Number** This field indicates the maximum number of transmitter retries that are allowed when parity error has occurred. **Note1:** The real retry number is TXRTY + 1, so 8 is the maximum retry number**.** **Note2:** This field cannot be changed when TXRTYEN enabled. The change flow is to disable TX_ETRTRY_EN first and then fill in new retry value.
[19]	**RXRTYEN**	**RX Error Retry Enable Bit** This bit enables receiver retry function when parity error has occurred. 0 = RX error retry function Disabled. 1 = RX error retry function Enabled. **Note:** Software must fill in the RXRTY value before enabling this bit.
[18:16]	**RXRTY**	**RX Error Retry Count Number** This field indicates the maximum number of receiver retries that are allowed when parity error has occurred **Note1:** The real retry number is RXRTY + 1, so 8 is the maximum retry number. **Note2:** This field cannot be changed when RXRTYEN enabled. The change flow is to disable RX_ETRTRY_EN first and then fill in new retry value.
[15]	**NSB**	**Stop Bit Length** This field indicates the length of stop bit. 0 = The stop bit length is 2 ETU. 1= The stop bit length is 1 ETU. **Note:** The default stop bit length is 2. SMC and UART adopts NSB to program the stop bit length
[14:13]	**TMRSEL**	**Timer Selection** 00 = All internal timer function Disabled. 01 = Internal 24 bit timer Enabled. Software can configure it by setting SC_TMRCTL0 [23:0]. SC_TMRCTL1 and SC_TMRCTL2 will be ignored in this mode. 10 = internal 24 bit timer and 8 bit internal timer Enabled. Software can configure the 24 bit timer by setting SC_TMRCTL0 [23:0] and configure the 8 bit timer by setting SC_TMRCTL1[7:0]. SC_TMRCTL2 will be ignored in this mode. 11 = Internal 24 bit timer and two 8 bit timers Enabled. Software can configure them by setting SC_TMRCTL0 [23:0], SC_TMRCTL1 [7:0] and SC_TMRCTL2 [7:0].
[12:8]	**BGT**	**Block Guard Time (BGT)** Block guard time means the minimum bit length between the leading edges of two consecutive characters between different transfer directions. This field indicates the counter for the bit length of block guard time. According to ISO7816-3, in T = 0 mode, software must fill 15 (real block guard time = 16.5) to this field; in T = 1 mode, software must fill 21 (real block guard time = 22.5) to it. **Note:** The real block guard time is BGT + 1.

[7:6]	**RXTRGLV**	**Rx Buffer Trigger Level** When the number of bytes in the receiving buffer equals the RXTRGLV, the RDA_IF will be set (if IER [RDAIENN] is enabled, an interrupt will be generated). 00 = INTR_RDA Trigger Level with 01 Bytes. 01 = INTR_RDA Trigger Level with 02 Bytes. 10 = INTR_RDA Trigger Level with 03 Bytes. 11 = Reserved.
[5:4]	**CONSEL**	**Convention Selection** 00 = Direct convention. 01 = Reserved. 10 = Reserved. 11 = Inverse convention. **Note:** If AUTOCEN(SC_CTL[3]) enabled, this fields are ignored.
[3]	**AUTOCEN**	**Auto Convention Enable Bit** 0 = Auto-convention Disabled. 1 = Auto-convention Enabled. When hardware receives TS in answer to reset state and the TS is direct convention, CONSEL(SC_CTL[5:4]) will be set to 00 automatically, otherwise if the TS is inverse convention, and CONSEL (SC_CTL[5:4]) will be set to 11. If software enables auto convention function, the setting step must be done before Answer to Reset state and the first data must be 3B or 3F. After hardware received first data and stored it at buffer, hardware will decided the convention and change the CONSEL (SC_CTL[5:4]) bits automatically. If the first data is not 3B or 3F, hardware will generate an interrupt ACERRIF (if ACERRIEN (SC_INTEN[10]) = 1 to CPU.
[2]	**TXOFF**	**TX Transition Disable Bit** 0 = The transceiver Enabled. 1 = The transceiver Disabled.
[1]	**RXOFF**	**RX Transition Disable Bit** 0 = The receiver Enabled. 1 = The receiver Disabled. **Note:** If AUTOCEN is enabled, this field must be ignored.
[0]	**SCEN**	**SC Engine Enable Bit** Set this bit to 1 to enable SC operation. If this bit is cleared, SC will force all transition to IDLE state.

SC Alternate Control Register (SC_ALTCTL)

Register	Offset	R/W	Description	Reset Value
SC_ALTCTL	SCx_BA+0x08	R/W	SC Alternate Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
ACTSTS2	ACTSTS1	ACTSTS0	RXBGTEN	ADACEN	Reserved	INITSEL	
7	6	5	4	3	2	1	0
CNTEN2	CNTEN1	CNTEN0	WARSTEN	ACTEN	DACTEN	RXRST	TXRST

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15]	**ACTSTS2**	**Internal Timer2 Active State (Read Only)** This bit indicates the timer counter status of timer2. 0 = Timer2 is not active. 1 = Timer2 is active.
[14]	**ACTSTS1**	**Internal Timer1 Active State (Read Only)** This bit indicates the timer counter status of timer1. 0 = Timer1 is not active. 1 = Timer1 is active.
[13]	**ACTSTS0**	**Internal Timer0 Active State (Read Only)** This bit indicates the timer counter status of timer0. 0 = Timer0 is not active. 1 = Timer0 is active.
[12]	**RXBGTEN**	**Receiver Block Guard Time Function Enable Bit** 0 = Receiver block guard time function Disabled. 1 = Receiver block guard time function Enabled.
[11]	**ADACEN**	**Auto Deactivation When Card Removal** 0 = Auto deactivation Disabled when hardware detected the card removal. 1 = Auto deactivation Enabled when hardware detected the card removal. **Note:** When the card is removed, hardware will stop any process and then do deactivation sequence (if this bit is set). If this process completes, hardware will generate an interrupt INITIF to CPU.
[11:10]	**Reserved**	Reserved.
[9:8]	**INITSEL**	**Initial Timing Selection** This fields indicates the timing of hardware initial state (activation or warm-reset or deactivation).

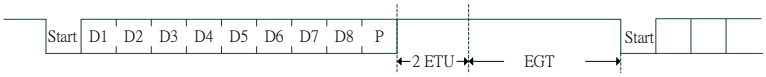
		Unit: SC clock Activation: refer to SC Activation Sequence in Figure 5.19-4. Warm-reset: refer to Warm-Reset Sequence in Figure 5.19-5 Deactivation: refer to Deactivation Sequence in Figure 5.19-6
[7]	**CNTEN2**	**Internal Timer2 Start Enable Bit** This bit enables Timer 2 to start counting. Software can fill 0 to stop it and set 1 to reload and count. 0 = Stops counting. 1 = Start counting. **Note1:** This field is used for internal 8 bit timer when TMRSEL(SC_CTL[14:13]) = 11. Don't filled CNTEN2 when TMRSEL(SC_CTL[14:13]) = 00 or TMRSEL(SC_CTL[14:13]) = 01 or TMRSEL(SC_CTL[14:13]) = 10. **Note2:** If the operation mode is not in auto-reload mode (SC_TMRCTL2[26] = 0), this bit will be auto-cleared by hardware. **Note3:** This field will be cleared by TXRST(SC_ALTCTL[0]) and RXRST(SC_ALTCTL[1]). So don't fill this bit, TXRST(SC_ALTCTL[0]), and RXRST(SC_ALTCTL[1]) at the same time. **Note4:** If SCEN(SC_CTL[0]) is not enabled, this filed cannot be programmed.
[6]	**CNTEN1**	**Internal Timer1 Start Enable Bit** This bit enables Timer 1 to start counting. Software can fill 0 to stop it and set 1 to reload and count. 0 = Stops counting. 1 = Start counting. **Note1:** This field is used for internal 8 bit timer when TMRSEL(SC_CTL[14:13]) = 10 or TMRSEL(SC_CTL[14:13]) = 11. Don't filled CNTEN1 when SC_CTL([TMRSEL] = 00 or SC_CTL[TMRSEL] = 01. **Note2:** If the operation mode is not in auto-reload mode (SC_TMRCTL1[26] = 0), this bit will be auto-cleared by hardware. **Note3:** This field will be cleared by TXRST(SC_ALTCTL[0]) and RXRST(SC_ALTCTL[1]), so don't fill this bit, TXRST(SC_ALTCTL[0]), and RXRST(SC_ALTCTL[1]) at the same time. **Note4:** If SCEN(SC_CTL[0]) is not enabled, this filed cannot be programmed.
[5]	**CNTEN0**	**Internal Timer0 Start Enable Bit** This bit enables Timer 0 to start counting. Software can fill 0 to stop it and set 1 to reload and count. 0 = Stops counting. 1 = Start counting. **Note1:** This field is used for internal 24 bit timer when TMRSEL (SC_CTL[14:13]) = 01. **Note2:** If the operation mode is not in auto-reload mode (SC_TMRCTL0[26] = 0), this bit will be auto-cleared by hardware. **Note3:** This field will be cleared by TXRST(SC_ALTCTL[0]) and RXRST(SC_ALTCTL[1]). So don't fill this bit, TXRST and RXRST at the same time. **Note4:** If SCEN(SC_CTL[0]) is not enabled, this filed cannot be programmed.
[4]	**WARSTEN**	**Warm Reset Sequence Generator Enable Bit** This bit enables SC controller to initiate the card by warm reset sequence 0 = No effect. 1 = Warm reset sequence generator Enabled. **Note1:** When the warm reset sequence completed, this bit will be cleared automatically and the INITIF(SC_INTSTS[8]) will be set to 1. **Note2:** This field will be cleared by TXRST(SC_ALTCTL[0]) and RXRST(SC_ALTCTL[1]), so don't fill this bit, TXRST, and RXRST at the same time.

		Note3: If SCEN(SC_CTL[0]) is not enabled, this filed cannot be programmed.
[3]	**ACTEN**	**Activation Sequence Generator Enable Bit** This bit enables SC controller to initiate the card by activation sequence 0 = No effect. 1 = Activation sequence generator Enabled. **Note1:** When the activation sequence completed, this bit will be cleared automatically and the INITIF(SC_INTSTS[8]) will be set to 1. **Note2:** This field will be cleared by TXRST(SC_ALTCTL[0]) and RXRST(SC_ALTCTL[1]), so don't fill this bit, TXRST(SC_ALTCTL[0]), and RXRST(SC_ALTCTL[1]) at the same time. **Note3:** If SCEN(SC_CTL[0]) is not enabled, this filed cannot be programmed.
[2]	**DACTEN**	**Deactivation Sequence Generator Enable Bit** This bit enables SC controller to initiate the card by deactivation sequence 0 = No effect. 1 = Deactivation sequence generator Enabled. **Note1:** When the deactivation sequence completed, this bit will be cleared automatically and the INITIF(SC_INTSTS[8]) will be set to 1. **Note2:** This field will be cleared by TXRST(SC_ALTCTL[0]) and RXRST(SC_ALTCTL[1]). So don't fill this bit, TXRST, and RXRST at the same time. **Note3:** If SC_CTL [SCEN] not enabled, this filed cannot be programmed.
[1]	**RXRST**	**Rx Software Reset** When RXRST is set, all the bytes in the receiver buffer and Rx internal state machine will be cleared. 0 = No effect. 1 = Reset the Rx internal state machine and pointers. **Note:** This bit will be auto cleared after reset is complete.
[0]	**TXRST**	**TX Software Reset** When TXRST is set, all the bytes in the transmit buffer and TX internal state machine will be cleared. 0 = No effect. 1 = Reset the TX internal state machine and pointers. **Note:** This bit will be auto cleared after reset is complete.

SC Extend Guard Time Register (SC_EGT)

Register	Offset	R/W	Description	Reset Value
SC_EGT	SCx_BA+0x0C	R/W	SC Extend Guard Time Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
EGT							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**EGT**	**Extended Guard Time** This field indicates the extended guard timer value.  **Note:** The counter is ETU base and the real extended guard time is EGT.

SC Receiver buffer Time-out Register (SC_RXTOUT)

Register	Offset	R/W	Description	Reset Value
SC_RXTOUT	SCx_BA+0x10	R/W	SC Receive Buffer Time-out Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							RFTM
7	6	5	4	3	2	1	0
RFTM							

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8:0]	**RFTM**	**SC Receiver Buffer Time-Out (ETU Base)** The time-out counter resets and starts counting whenever the RX buffer received a new data word. Once the counter decrease to 1 and no new data is received or CPU does not read data by reading SC_DAT buffer, a receiver time-out interrupt INT_RTMR will be generated(if RXTOIF(SC_INTEN[9]) = 1). **Note1:** The counter unit is ETU based and the interval of time-out is RFTM + 0.5 **Note2:** Fill all 0 to this field indicates to disable this function.

SC Clock Divider Control Register (SC_ETUCTL)

Register	Offset	R/W	Description	Reset Value
SC_ETUCTL	SCx_BA+0x14	R/W	SC ETU Control Register	0x0000_0173

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
CMPEN	Reserved			ETURDIV			
7	6	5	4	3	2	1	0
ETURDIV							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15]	**CMPEN**	**Compensation Mode Enable Bit** This bit enables clock compensation function. When this bit enabled, hardware will alternate between n clock cycles and n-1 clock cycles, where n is the value to be written into the ETURDIV. 0 = Compensation function Disabled. 1 = Compensation function Enabled.
[14:12]	**Reserved**	Reserved.
[11:0]	**ETURDIV**	**ETU Rate Divider** The field indicates the clock rate divider. The real ETU is ETURDIV + 1. **Note:** Software can configure this field, but this field must be greater than 0x004.

SC Interrupt Control Register (SC_INTEN)

Register	Offset	R/W	Description	Reset Value
SC_INTEN	SCx_BA+0x18	R/W	SC Interrupt Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					ACERRIEN	RXTOIF	INITIEN
7	6	5	4	3	2	1	0
CDIEN	BGTIEN	TMR2IEN	TMR1IEN	TMR0IEN	TERRIEN	TXBEIEN	RDAIEN

Bits	Description	
[31:11]	**Reserved**	Reserved.
[10]	**ACERRIEN**	**Auto Convention Error Interrupt Enable Bit** This field is used for auto-convention error interrupt enable. 0 = Auto-convention error interrupt Disabled. 1 = Auto-convention error interrupt Enabled.
[9]	**RXTOIF**	**Receiver Buffer Time-Out Interrupt Enable Bit** This field is used for receiver buffer time-out interrupt enable. 0 = Receiver buffer time-out interrupt Disabled. 1 = Receiver buffer time-out interrupt Enabled.
[8]	**INITIEN**	**Initial End Interrupt Enable Bit** This field is used for activation (ACTEN(SC_ALTCTL[3] = 1)), deactivation ((DACTEN SC_ALTCTL[2]) = 1) and warm reset (WARSTEN (SC_ALTCTL [4])) sequence interrupt enable. 0 = Initial end interrupt Disabled. 1 = Initial end interrupt Enabled.
[7]	**CDIEN**	**Card Detect Interrupt Enable Bit** This field is used for card detect interrupt enable. The card detect status is CINSERT(SC_STATUS[12]) 0 = Card detect interrupt Disabled. 1 = Card detect interrupt Enabled.
[6]	**BGTIEN**	**Block Guard Time Interrupt Enable Bit** This field is used for block guard time interrupt enable. 0 = Block guard time Disabled. 1 = Block guard time Enabled.
[5]	**TMR2IEN**	**Timer2 Interrupt Enable Bit** This field is used for TMR2 interrupt enable.

		0 = Timer2 interrupt Disabled. 1 = Timer2 interrupt Enabled.
[4]	**TMR1IEN**	**Timer1 Interrupt Enable Bit** This field is used to enable the TMR1 interrupt. 0 = Timer1 interrupt Disabled. 1 = Timer1 interrupt Enabled.
[3]	**TMR0IEN**	**Timer0 Interrupt Enable Bit** This field is used to enable TMR0 interrupt enable. 0 = Timer0 interrupt Disabled. 1 = Timer0 interrupt Enabled.
[2]	**TERRIEN**	**Transfer Error Interrupt Enable Bit** This field is used for transfer error interrupt enable. The transfer error states is at SC_STATUS register which includes receiver break error BEF(SC_STATUS[6]), frame error FEF(SC_STATUS[5]), parity error PEF(SC_STATUS[4]), receiver buffer overflow error RXOV(SC_STATUS[0]), transmit buffer overflow error TXOVER(SC_STATUS[8]), receiver retry over limit error RXOVERR(SC_STATUS[22] and transmitter retry over limit error TXOVERR(SC_STATUS[30]. 0 = Transfer error interrupt Disabled. 1 = Transfer error interrupt Enabled.
[1]	**TBEIEN**	**Transmit Buffer Empty Interrupt Enable Bit** This field is used for transmit buffer empty interrupt enable. 0 = Transmit buffer empty interrupt Disabled. 1 = Transmit buffer empty interrupt Enabled.
[0]	**RDAIEN**	**Receive Data Reach Interrupt Enable Bit** This field is used for received data reaching trigger level RXTRGLV (SC_CTL[7:6]) interrupt enable. 0 = Receive data reach trigger level interrupt Disabled. 1 = Receive data reach trigger level interrupt Enabled.

SC Interrupt Status Register (SC_INTSTS)

Register	Offset	R/W	Description	Reset Value
SC_INTSTS	SCx_BA+0x1C	R/W	SC Interrupt Status Register	0x0000_0002

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					ACERRIF	RBTOIF	INITIF
7	6	5	4	3	2	1	0
CDIF	BGTIF	TMR2IF	TMR1IF	TMR0IF	TERRIF	TBEIF	RDAIF

Bits	Description	
[31:11]	**Reserved**	Reserved.
[10]	**ACERRIF**	**Auto Convention Error Interrupt Status Flag (Read Only)** This field indicates auto convention sequence error. If the received TS at ATR state is neither 3B nor 3F, this bit will be set. **Note:** This bit is read only, but it can be cleared by writing 1 to it.
[9]	**RBTOIF**	**Receiver Buffer Time-Out Interrupt Status Flag (Read Only)** This field is used for receiver buffer time-out interrupt status flag. **Note:** This field is the status flag of receiver buffer time-out state. If software wants to clear this bit, software must read all receiver buffer remaining data by reading SC_DAT buffer,
[8]	**INITIF**	**Initial End Interrupt Status Flag (Read Only)** This field is used for activation (ACTEN(SC_ALTCTL[3])), deactivation (DACTEN (SC_ALTCTL[2])) and warm reset (WARSTEN (SC_ALTCTL[4])) sequence interrupt status flag. **Note:** This bit is read only, but it can be cleared by writing 1 to it.
[7]	**CDIF**	**Card Detect Interrupt Status Flag (Read Only)** This field is used for card detect interrupt status flag. The card detect status is CINSERT (SC_STATUS[12]) and CREMOVE(SC_STATUS[11]). **Note:** This field is the status flag of CINSERT(SC_STATUS[12]) or CREMOVE(SC_STATUS[11])]. So if software wants to clear this bit, software must write 1 to this field.
[6]	**BGTIF**	**Block Guard Time Interrupt Status Flag (Read Only)** This field is used for block guard time interrupt status flag. **Note1:** This bit is valid when RXBGTEN (SC_ALTCTL[12]) is enabled. **Note2:** This bit is read only, but it can be cleared by writing “1” to it.
[5]	**TMR2IF**	**Timer2 Interrupt Status Flag (Read Only)** This field is used for TMR2 interrupt status flag.

		Note: This bit is read only, but it can be cleared by writing 1 to it.
[4]	**TMR1IF**	**Timer1 Interrupt Status Flag (Read Only)** This field is used for TMR1 interrupt status flag. **Note:** This bit is read only, but it can be cleared by writing 1 to it.
[3]	**TMR0IF**	**Timer0 Interrupt Status Flag (Read Only)** This field is used for TMR0 interrupt status flag. **Note:** This bit is read only, but it can be cleared by writing 1 to it.
[2]	**TERRIF**	**Transfer Error Interrupt Status Flag (Read Only)** This field is used for transfer error interrupt status flag. The transfer error states is at SC_STATUS register which includes receiver break error BEF(SC_STATUS[6]), frame error FEF(SC_STATUS[5], parity error PEF(SC_STATUS[4] and receiver buffer overflow error RXOV(SC_STATUS[0]), transmit buffer overflow error TXOVER(SC_STATUS[8]), receiver retry over limit error RXOVERR(SC_STATUS[22] and transmitter retry over limit error TXOVERR(SC_SC[30]). **Note:** This field is the status flag of BEF(SC_STATUS[6]), FEF(SC_STATUS[5], PEF(SC_STATUS[4], RXOV(SC_STATUS[0]), TXOVER(SC_STATUS[8]), RXOVERR(SC_STATUS[22] or TXOVERR(SC_SC[30]). So, if software wants to clear this bit, software must write 1 to each field.
[1]	**TBEIF**	**Transmit Buffer Empty Interrupt Status Flag (Read Only)** This field is used for transmit buffer empty interrupt status flag. **Note:** This field is the status flag of transmit buffer empty state. If software wants to clear this bit, software must write data to SC_DAT buffer and then this bit will be cleared automatically.
[0]	**RDAIF**	**Receive Data Reach Interrupt Status Flag (Read Only)** This field is used for received data reaching trigger level RXTRGLV (SC_CTL[7:6]) interrupt status flag. **Note:** This field is the status flag of received data reaching RXTRGLV (SC_CTL[7:6]). If software reads data from SC_DAT and receiver buffer data byte number is less than RXTRGLV (SC_CTL[7:6]), this bit will be cleared automatically.

SC Transfer Status Register (SC_STATUS)

Register	Offset	R/W	Description	Reset Value
SC_STATUS	SCx_BA+0x20	R/W	SC Status Register	0x0000_0202

31	30	29	28	27	26	25	24
TXACT	TXOVERR	TXRERR	Reserved		TXPOINT		
23	22	21	20	19	18	17	16
RXACT	RXOVERR	RXRERR	Reserved		RXPOINT		
15	14	13	12	11	10	9	8
Reserved		CDPINSTS	CINSERT	CREMOVE	TXFULL	TXEMPTY	TXOV
7	6	5	4	3	2	1	0
Reserved	BEF	FEF	PEF	Reserved	RXFULL	RXEMPTY	RXOV

Bits	Description	
[31]	**TXACT**	**Transmit In Active Status Flag (Read Only)** 0 = This bit is cleared automatically when TX transfer is finished or the last byte transmission has completed. 1 = This bit is set by hardware when TX transfer is in active and the STOP bit of the last byte has been transmitted.
[30]	**TXOVERR**	**Transmitter Over Retry Error (Read Only)** This bit is set by hardware when transmitter re-transmits over retry number limitation. **Note:** This bit is read only, but it can be cleared by writing 1 to it.
[29]	**TXRERR**	**Transmitter Retry Error (Read Only)** This bit is set by hardware when transmitter re-transmits. **Note1:** This bit is read only, but it can be cleared by writing 1 to it. **Note2** This bit is a flag and cannot generate any interrupt to CPU.
[28:26]	**Reserved**	Reserved.
[25:24]	**TXPOINT**	**Transmit Buffer Pointer Status Flag (Read Only)** This field indicates the TX buffer pointer status flag. When CPU writes data into SC_DAT, TXPOINT increases one. When one byte of TX Buffer is transferred to transmitter shift register, TXPOINT decreases one.
[23]	**RXACT**	**Receiver In Active Status Flag (Read Only)** This bit is set by hardware when RX transfer is in active. This bit is cleared automatically when RX transfer is finished.
[22]	**RXOVERR**	**Receiver Over Retry Error (Read Only)** This bit is set by hardware when RX transfer error retry over retry number limit. **Note1:** This bit is read only, but it can be cleared by writing 1 to it. **Note2:** If CPU enables receiver retries function by setting RXRTYEN (SC_CTL[19]), the PEF(SC_STATUS[4]) flag will be ignored (hardware will not set PEF(SC_STATUS[4])).
[21]	**RXRERR**	**Receiver Retry Error (Read Only)**

		This bit is set by hardware when RX has any error and retries transfer. **Note1:** This bit is read only, but it can be cleared by writing 1 to it. **Note2** This bit is a flag and cannot generate any interrupt to CPU. **Note3:** If CPU enables receiver retry function by setting RXRTYEN (SC_CTL[19]) , the PEF(SC_STATUS[4]) flag will be ignored (hardware will not set PEF(SC_STATUS[4])).
[20:18]	**Reserved**	Reserved.
[17:16]	**RXPOINT**	**Receiver Buffer Pointer Status Flag (Read Only)** This field indicates the RX buffer pointer status flag. When SC receives one byte from external device, RXPOINT(SC_STATUS[17:16]) increases one. When one byte of RX buffer is read by CPU, RXPOINT(SC_STATUS[17:16]) decreases one.
[15:14]	**Reserved**	Reserved.
[13]	**CDPINSTS**	**Card Detect Status Of SC_CD Pin Status (Read Only)** This bit is the pin status flag of SC_CD 0 = The SC_CD pin state at low. 1 = The SC_CD pin state at high.
[12]	**CINSERT**	**Card Detect Insert Status Of SC_CD Pin (Read Only)** This bit is set whenever card has been inserted. 0 = No effect.1 = Card insert. **Note1:** This bit is read only, but it can be cleared by writing “1” to it. **Note2:** The card detect engine will start after SCEN (SC_CTL[0]) set.
[11]	**CREMOVE**	**Card Detect Removal Status Of SC_CD Pin (Read Only)** This bit is set whenever card has been removal. 0 = No effect. 1 = Card removed. **Note1:** This bit is read only, but it can be cleared by writing “1” to it. **Note2:** Card detect engine will start after SCEN (SC_CTL[0])set.
[10]	**TXFULL**	**Transmit Buffer Full Status Flag (Read Only)** This bit indicates TX buffer full or not. This bit is set when TX pointer is equal to 4, otherwise is cleared by hardware.
[9]	**TXEMPTY**	**Transmit Buffer Empty Status Flag (Read Only)** This bit indicates TX buffer empty or not. When the last byte of TX buffer has been transferred to Transmitter Shift Register, hardware sets this bit high. It will be cleared when writing data into SC_DAT (TX buffer not empty).
[8]	**TXOV**	**TX Overflow Error Interrupt Status Flag (Read Only)** If TX buffer is full, an additional write to SC_DAT will cause this bit be set to “1” by hardware. **Note:** This bit is read only, but it can be cleared by writing 1 to it.
[7]	**Reserved**	Reserved.
[6]	**BEF**	**Receiver Break Error Status Flag (Read Only)** This bit is set to a logic 1 whenever the received data input (RX) held in the “spacing state” (logic 0) is longer than a full word transmission time (that is, the total time of “start bit” + data bits + parity + stop bits). . **Note1:** This bit is read only, but it can be cleared by writing 1 to it. **Note2:** If CPU sets receiver retries function by setting RXRTYEN(SC_CTL[19]) , hardware will not set this flag.

[5]	**FEF**	**Receiver Frame Error Status Flag (Read Only)** This bit is set to logic 1 whenever the received character does not have a valid “stop bit” (that is, the stop bit following the last data bit or parity bit is detected as a logic 0). **Note1:** This bit is read only, but it can be cleared by writing 1 to it. **Note2:** If CPU sets receiver retries function by setting RXRTYEN(SC_CTL[19]) , hardware will not set this flag.
[4]	**PEF**	**Receiver Parity Error Status Flag (Read Only)** This bit is set to logic 1 whenever the received character does not have a valid “parity bit”. **Note1:** This bit is read only, but it can be cleared by writing 1 to it. **Note2:** If CPU sets receiver retries function by setting RXRTYEN(SC_CTL[19]) , hardware will not set this flag.
[3]	**Reserved**	Reserved.
[2]	**RXFULL**	**Receiver Buffer Full Status Flag (Read Only)** This bit indicates RX buffer full or not. This bit is set when RX pointer is equal to 4, otherwise it is cleared by hardware.
[1]	**RXEMPTY**	**Receiver Buffer Empty Status Flag(Read Only)** This bit indicates RX buffer empty or not. When the last byte of Rx buffer has been read by CPU, hardware sets this bit high. It will be cleared when SC receives any new data.
[0]	**RXOV**	**RX Overflow Error Status Flag (Read Only)** This bit is set when RX buffer overflow. If the number of received bytes is greater than Rx Buffer size (4 bytes), this bit will be set. **Note:** This bit is read only, but it can be cleared by writing 1 to it.

SC PIN Control State Register (SC_PINCTL)

Register	Offset	R/W	Description	Reset Value
SC_PINCTL	SCx_BA+0x24	R/W	SC Pin Control State Register	0x0000_00x0

31	30	29	28	27	26	25	24
Reserved	SYNC	Reserved					
23	22	21	20	19	18	17	16
Reserved					RSTSTS	PWRSTS	DATSTS
15	14	13	12	11	10	9	8
Reserved				PWRINV	Reserved	SCDOOUT	Reserved
7	6	5	4	3	2	1	0
Reserved	CLKKEEP	Reserved				SCRST	PWREN

Bits	Description	
[31]	**Reserved**	Reserved.
[30]	**SYNC**	**SYNC Flag Indicator** Due to synchronization, software should check this bit when writing a new value to SC_PINCTL register. 0 = Synchronizing is completion, user can write new data to SC_PINCTL register. 1 = Last value is synchronizing. **Note:** This bit is read only.
[29:12]	**Reserved**	Reserved.
[18]	**RSTSTS**	**SC_RST Pin Signals** This bit is the pin status of SC_RST 0 = SC_RST pin is low. 1 = SC_RST pin is high. **Note:** When SC is operated at activation, warm reset or deactivation mode, this bit will be changed automatically. This bit is not allowed to program when SC is operated at these modes.
[17]	**PWRSTS**	**SC_PWR Pin Signal** This bit is the pin status of SC_PWR 0 = SC_PWR pin to low. 1 = SC_PWR pin to high. **Note:** When SC is operated at activation, warm reset or deactivation mode, this bit will be changed automatically. This bit is not allowed to program when SC is operated at these modes.
[16]	**DATSTS**	**This bit is the pin status of SC_DAT** 0 = The SC_DAT pin is low. 1 = The SC_DAT pin is high.

[11]	**PWRINV**	**SC_POW Pin Inverse** This bit is used for inverse the SC_POW pin. There are four kinds of combination for SC_POW pin setting by PWRINV and PWREN(SC_PINCTL[0]). PWRINV is bit 1 and PWREN is bit 0 for SC_POW_Pin as high or low voltage selection. PWRINV is 0 and PWREN is 0, than SC_POW Pin output 0. PWRINV is 0 and PWREN is 1, than SC_POW Pin output 1. PWRINV is 1 and PWREN is 0, than SC_POW Pin output 1. PWRINV is 1 and PWREN is 1, than SC_POW Pin output 0. **Note:** Software must select PWRINV before Smart Card is enabled by SCEN (SC_CTL[0]).
[10]	**Reserved**	Reserved.
[9]	**SCDOOUT**	**SC Data Output Pin** This bit is the pin status of SCDOSTS but user can drive SCDOSTS pin to high or low by setting this bit. 0 = Drive SCDOSTS pin to low. 1 = Drive SCDOSTS pin to high. **Note:** When SC is at activation, warm reset or deactivation mode, this bit will be changed automatically. So don't fill this field when SC is in these modes.
[8]	**Reserved**	Reserved.
[6]	**CLKKEEP**	**SC Clock Enable Bit** 0 = SC clock generation Disabled. 1 = SC clock always keeps free running. **Note:** When operating in activation, warm reset or deactivation mode, this bit will be changed automatically. So don't fill this field when operating in these modes.
[5:2]	**Reserved**	Reserved.
[1]	**SCRST**	**SCRST Pin Signal** This bit is the pin status of SCRST but user can drive SCRST pin to high or low by setting this bit. Write this field to drive SCRST pin. 0 = Drive SCRST pin to low. 1 = Drive SCRST pin to high. Read this field to get SCRST pin status. 0 = SCRST pin status is low. 1 = SCRST pin status is high. **Note:** When operating at activation, warm reset or deactivation mode, this bit will be changed automatically. So don't fill this field when operating in these modes.
[0]	**PWREN**	**SC_PWREN Pin Signal** Software can set PWREN and PWRINV to decide SC_PWR pin is in high or low level. Write this field to drive SC_PWR pin Refer PWRINV description for programming SC_PWR pin voltage level. Read this field to get SC_PWR pin status. 0 = SC_PWR pin status is low. 1 = SC_PWR pin status is high. **Note:** When operating at activation, warm reset or deactivation mode, this bit will be changed automatically. So don't fill this field when operating in these modes.

SC Timer Control Register 0 (SC_TMRCTL0)

Register	Offset	R/W	Description	Reset Value
SC_TMRCTL0	SCx_BA+0x28	R/W	SC Internal Timer Control Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				OPMODE			
23	22	21	20	19	18	17	16
CNT							
15	14	13	12	11	10	9	8
CNT							
7	6	5	4	3	2	1	0
CNT							

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:24]	**OPMODE**	**Timer 0 Operation Mode Selection** This field indicates the internal 24-bit timer operation selection. Refer to 6.25.4.4 for programming Timer0.
[23:0]	**CNT**	**Timer 0 Counter Value (ETU Base)** This field indicates the internal timer operation values.

SC Timer Control Register 1 (SC_TMRCTL1)

Register	Offset	R/W	Description	Reset Value
SC_TMRCTL1	SCx_BA+0x2C	R/W	SC Internal Timer Control Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				OPMODE			
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
CNT							

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:24]	**OPMODE**	**Timer 1 Operation Mode Selection** This field indicates the internal 8-bit timer operation selection. Refer to 6.25.4.4 for programming Timer1.
[23:8]	**Reserved**	Reserved.
[7:0]	**CNT**	**Timer 1 Counter Value (ETU Base)** This field indicates the internal timer operation values.

SC Timer Control Register 2 (SC_TMRCTL2)

Register	Offset	R/W	Description	Reset Value
SC_TMRCTL2	SCx_BA+0x30	R/W	SC Internal Timer Control Register 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				OPMODE			
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
CNT							

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:24]	**OPMODE**	**Timer 2 Operation Mode Selection** This field indicates the internal 8-bit timer operation selection Refer to 6.25.4.4 for programming Timer2
[23:8]	**Reserved**	Reserved.
[7:0]	**CNT**	**Timer 2 Counter Value (ETU Base)** This field indicates the internal timer operation values.

SC UART Mode Control Register (SC_UACTL)

Register	Offset	R/W	Description	Reset Value
SC_UARTCTL	SCx_BA + 0x34	R/W	SC UART Mode Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
OPE	PBOFF	WLS	Reserved				UARTEN

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**OPE**	**Odd Parity Enable Bit** 0 = Even number of logic 1's are transmitted or check the data word and parity bits in receiving mode. 1 = Odd number of logic 1's are transmitted or check the data word and parity bits in receiving mode. **Note:** This bit has effect only when PBOFF bit is '0'.
[6]	**PBOFF**	**Parity Bit Disable Bit** 0 = Parity bit is generated or checked between the "last data word bit" and "stop bit" of the serial data. 1 = Parity bit is not generated (transmitting data) or checked (receiving data) during transfer. **Note:** In smart card mode, this field must be '0' (default setting is with parity bit)
[5:4]	**WLS [1:0]**	**Data Length** 00 = Character Data Length is 8 bits. 01 = Character Data Length is 7 bits. 10 = Character Data length is 6 bits. 11 = Character Data Length is 5 bits. **Note:** In smart card mode, this WLS must be '00'
[3:1]	**Reserved**	Reserved.
[0]	**UARTEN**	**UART Mode Enable Bit** 0 = Smart Card mode. 1 = UART mode. **Note1:** When operating in UART mode, user must set CONSEL (SC_CTL[5:4]) = 00 and AUTOCEN(SC_CTL[3]) = 0. **Note2:** When operating in Smart Card mode, user must set SC_UARTCTL [7:0] = 00. **Note3:** When UART is enabled, hardware will generate a reset to reset FIFO and internal

Bits	Description	
		state machine.

SC Timer 0 Current Data Register (SC_TMRDAT0)

Register	Offset	R/W	Description	Reset Value
SC_TMRDAT0	SCx_BA+0x38	R	SC Timer 0 Current Data Register	0x0000_07FF

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
CNT0							
15	14	13	12	11	10	9	8
CNT0							
7	6	5	4	3	2	1	0
CNT0							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**CNT0**	**Timer0 Current Counter Value (Read Only)** This field indicates the current count values of timer0.

SC Timer 1 and 2 Current Data Register (SC_TMRDAT1_2)

Register	Offset	R/W	Description	Reset Value
SC_TMRDAT1_2	SCx_BA+0x3C	R	SC Timer 1 and 2 Current Data Register	0x0000_7F7F

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
CNT2							
7	6	5	4	3	2	1	0
CNT1							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:8]	**CNT2**	**Timer2 Current Counter Value (Read Only)** This field indicates the current count values of timer2.
[7:0]	**CNT1**	**Timer1 Current Counter Value (Read Only)** This field indicates the current count values of timer1.

6.26 Secure Digital Host Controller

6.26.1 Overview

The Secure Digital Host Controller (SD Host) has DMAC unit and SD unit. The DMAC unit provides a DMA (Direct Memory Access) function for SD to exchange data between system memory and shared buffer (128 bytes), and the SD unit controls the interface of SD/SDHC. The SDHOST controller can support SD/SDHC and cooperated with DMAC to provide a fast data transfer between system memory and cards.

6.26.2 Features

- AMBA AHB master/slave interface compatible, for data transfer and register read/write.
- Supports single DMA channel.
- Supports hardware Scatter-Gather function.
- Using single 128 Bytes shared buffer for data exchange between system memory and cards.
- Synchronous design for DMA with single clock domain, AHB bus clock (HCLK).
- Interface with DMAC for register read/write and data transfer.
- Supports SD/SDHC card.
- Completely asynchronous design for Secure Digital with two clock domains, HCLK and Engine clock, note that frequency of HCLK should be higher than the frequency of peripheral clock.

6.26.3 Block Diagram and Card Pad Assignment

The block diagram and Card Pad Assignment of SDHOST Controller is shown as follows.

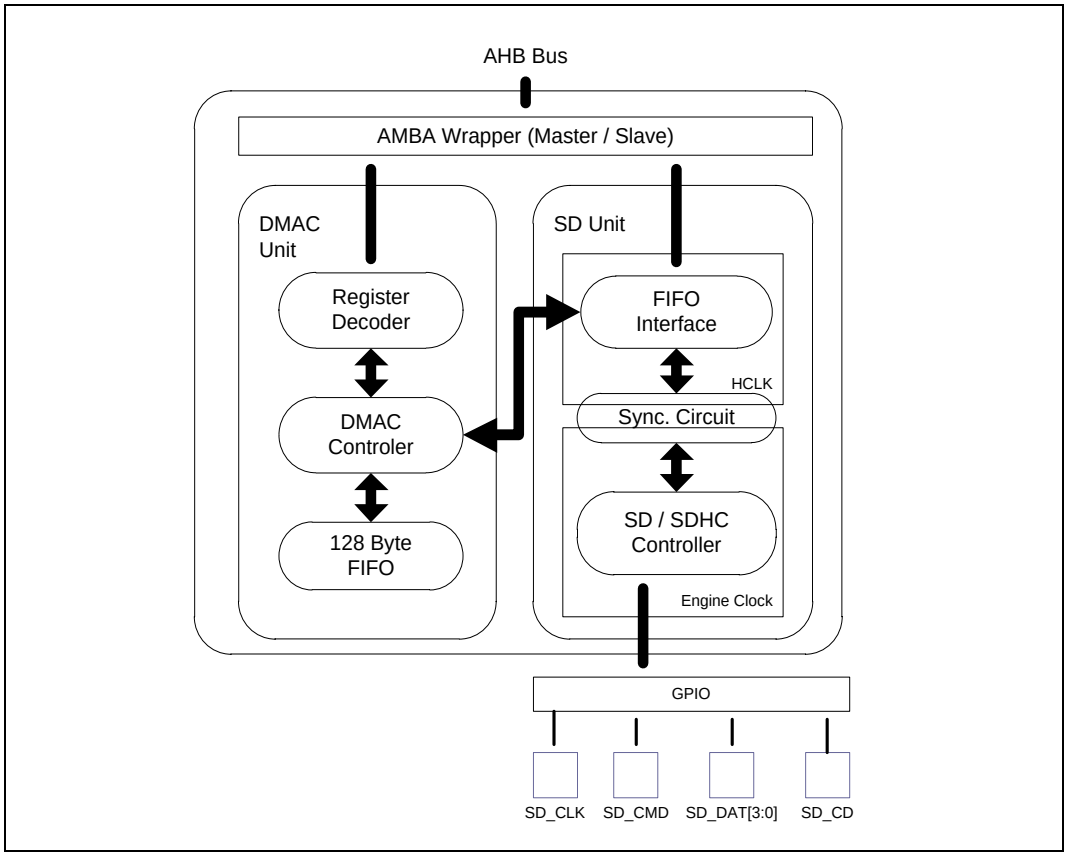


Figure 6.26-1 SD Host Controller Block Diagram

NAME	Description
SD_DAT0	SD Data (bit 0)
SD_DAT1	SD Data (bit 1)
SD_DAT2	SD Data (bit 2)
SD_DAT3	SD Data (bit 3)
SD_CMD	SD Command / Response
SD_CLK	SD Clock pin
SD_CD	Card Detect (Source can be GPIO or DAT3 (in SDIER)

Table 6.26-1 SD/SDHC Pad Assignment

6.26.4 SD Host DMA Controller

The SD host DMA controller provides a DMA (Direct Memory Access) function for SD host controller to exchange data between system memory (SRAM) and shared buffer (128 bytes). Arbitration of DMA

request between SD host is done by DMA's bus master. Software just simply fills in the starting address and enables DMA, and then let DMA to handle the data transfer automatically.

There is a 128 bytes shared buffer inside DMA, it can provide multi-block transfers for SD host. Software can access these shared buffers directly when SD host is not in busy.

6.26.4.1 Programming Flow

Here is a simple example programming flow without DMA Scatter-Gather enable.

1. Set DMAEN (SDH_DMACTL[0]) to enable DMA.
2. Fill corresponding starting address in SDH_DMASA for SD host.
3. Trigger SD host to start DMA transfer.
4. Wait for transfer finished.

Here is a simple example programming flow with DMA Scatter-Gather enable.

1. Set DMAEN (SDH_DMACTL[0]) to enable DMA (SDH_DMACTL[0]) and SGEN (SDH_DMACTL[3]) to enable Scatter-Gather function.
2. Fill corresponding starting address of Physical Address Descriptor (PAD) table in SDH_DMASA for SD host.
3. When bit-0 of SDH_DMASA is 1, the PAD will fetch in out of order, otherwise, it's fetched in order from PAD. The first time of writing bit-0 with 1 or not is not available for this function. The bits will be available in PAD table.
4. Trigger SD host to start DMA transfer.
5. Wait for transfer finished.

6.26.5 SD Host Functional Description

SD host provides an interface for SD/SDHC card access. This SD controller provides 2 SD ports – port0 and port1. Each port can provide 1-bit/4-bit data bus mode for SD.

SD controller uses an independent clock source named SDCLK as engine clock. SDCLK can be completely asynchronous with system clock HCLK, software can change SD clock arbitrary. Note that HCLK should be faster than SDCLK.

This SD controller can generate all types of 48-bit command to SD card and retrieve all types of response from SD card. After response in, the content of response will be stored at SDRSP0 and SDRSP1. SD controller will calculate CRC7 and check its correctness for response. If CRC7 is error, CDRIF (SDH_INTSTS[1]) will be set and CRC7 (SDH_INTSTS[2]) will be '0'. For response R1b, software should notice that after response in, SD card will put busy signal on data line DAT0; software should check this status with clock polling until it became high. For response R3, CRC7 is invalid; but SD controller will still calculate CRC7 and get an error result, software should ignore this error and clear CDRIF flag (SDH_INTSTS[1]).

This SD controller is composed of two state machines – command/response part and data part. For command/response part, the trigger bits are COEN, RIEN, R2_EN, CLK74OE and CLK8_OE in SDH_CTL register. If software enables all of these bits, the execution priority will be CLK74OE > COEN > RIEN/R2_EN > CLK8_OE, note that RIEN and R2_EN can't be triggered at the same time. For data part, there are DIEN and DOEN for choose. Software can only trigger one of them at one time. If DIEN is triggered, SD controller waits start bit from data line DAT0 immediately, and then get specified amount data from SD card. After data-in, SD controller will check CRC16 correctness; if it is error, CDRIF (SDH_INTSTS[1]) will be set and CRC16 (SDH_INTSTS[3]) will be '0'. If DOEN is triggered, SD controller will wait response in finished, and then send specified amount data to SD card. After data-out, SD controller will get CRC status from SD card and check its correctness; it should be '010', otherwise CDRIF (SDH_INTSTS[1]) will be set and CRCSTAT (SDH_INTSTS[6:4]) will be the value it received.

If R2_EN is triggered, SD controller will receive response R2 (136 bits) from SD card, CRC7 and end bit will be dropped. The receiving data will be placed at DMA's buffer, starting from address offset 0x0.

This SD controller also provides multiple block transfer function (change BLKLEN to change the block length). Software can use this function to accelerate data transfer throughput. If CRC7, CRC16 or CRC status is error, SD controller will stop transfer and set CDRIF (SDH_INTSTS[1]), software should do engine reset when this situation occurred.

There is a hardware time-out mechanism for response in and data in inside SD engine. Software can specify a 24-bit time-out value at TOUT, and then SD controller will decide when to time-out according to this value.

6.26.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
SDH Base address: **SDH_BA = 0x4000_D000**				
SDH_FB_n **n=0,1..31**	SDH_BA+0x000 + 0x4 * n	R/W	Shared Buffer (FIFO)	0x0000_0000
SDH_DMACTL	SDH_BA+0x400	R/W	DMA Control and Status Register	0x0000_0000
SDH_DMASA	SDH_BA+0x408	R/W	DMA Transfer Starting Address Register	0x0000_0000
SDH_DMABCNT	SDH_BA+0x40C	R	DMA Transfer Byte Count Register	0x0000_0000
SDH_DMAINTEN	SDH_BA+0x410	R/W	DMA Interrupt Enable Control Register	0x0000_0001
SDH_DMAINTSTS	SDH_BA+0x414	R/W	DMA Interrupt Status Register	0x0000_0000
SDH_GCTL	SDH_BA+0x800	R/W	Global Control and Status Register	0x0000_0000
SDH_GINTEN	SDH_BA+0x804	R/W	Global Interrupt Control Register	0x0000_0001
SDH_GINTSTS	SDH_BA+0x808	R/W	Global Interrupt Status Register	0x0000_0000
SDH_CTL	SDH_BA+0x820	R/W	SD Control and Status Register	0x0101_0000
SDH_CMDARG	SDH_BA+0x824	R/W	SD Command Argument Register	0x0000_0000
SDH_INTEN	SDH_BA+0x828	R/W	SD Interrupt Control Register	0x0000_0A00
SDH_INTSTS	SDH_BA+0x82C	R/W	SD Interrupt Status Register	0x000X_008C
SDH_RESP0	SDH_BA+0x830	R	SD Receiving Response Token Register 0	0x0000_0000
SDH_RESP1	SDH_BA+0x834	R	SD Receiving Response Token Register 1	0x0000_0000
SDH_BLEN	SDH_BA+0x838	R/W	SD Block Length Register	0x0000_01FF
SDH_TOUT	SDH_BA+0x83C	R/W	SD Response/Data-in Time-out Register	0x0000_0000

6.26.7 Register Description

DMA Control and Status Register (SDH_DMACTL)

Register	Offset	R/W	Description	Reset Value
SDH_DMACTL	SDH_BA+0x400	R/W	DMA Control and Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved						DMABUSY	Reserved
7	6	5	4	3	2	1	0
Reserved				SGEN	Reserve	DMARST	DMAEN

Bits	Description	
[31:10]	**Reserved**	Reserved.
[9]	**DMABUSY**	**DMA Transfer Is In Progress** This bit indicates if SD Host is granted and doing DMA transfer or not. 0 = DMA transfer is not in progress. 1 = DMA transfer is in progress.
[8:4]	**Reserved**	Reserved.
[3]	**SGEN**	**Scatter-Gather Function Enable Bit** 0 = Scatter-gather function Disabled (DMA will treat the starting address in DMASAR as starting pointer of a single block memory). 1 = Scatter-gather function Enabled (DMA will treat the starting address in DMASAR as a starting address of Physical Address Descriptor (PAD) table. The format of these Pads' will be described later).
[2]	**Reserved**	Reserved.
[1]	**DMARST**	**Software Engine Reset** 0 = No effect. 1 = Reset internal state machine and pointers. The contents of control register will not be cleared. This bit will auto be cleared after few clock cycles. **Note:** The software reset DMA related registers.
[0]	**DMAEN**	**DMA Engine Enable Bit** 0 = DMA Disabled. 1 = DMA Enabled. If this bit is cleared, DMA will ignore all requests from SD host and force bus master into IDLE state. **Note:** If target abort is occurred, DMAEN will be cleared.

DMA Transfer Starting Address Register (SDH_DMASA)

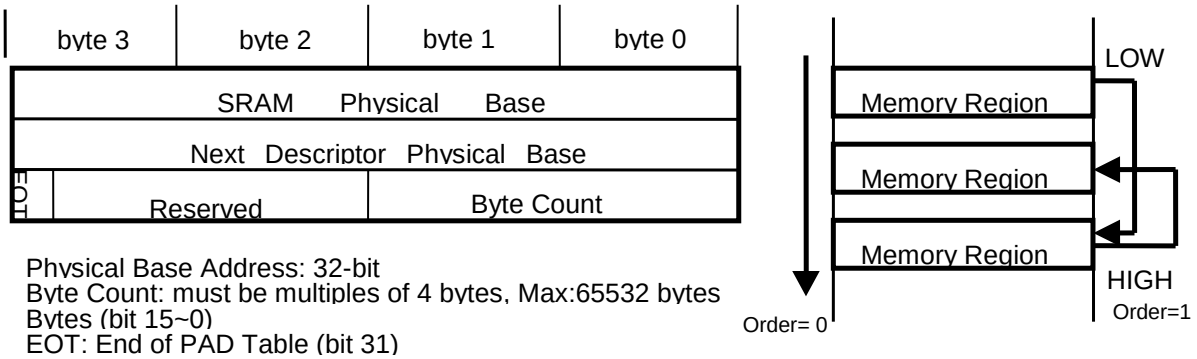
Register	Offset	R/W	Description	Reset Value
SDH_DMASA	SDH_BA+0x408	R/W	DMA Transfer Starting Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMASA[31:24]							
23	22	21	20	19	18	17	16
DMASA[23:16]							
15	14	13	12	11	10	9	8
DMASA[15:8]							
7	6	5	4	3	2	1	0
DMASA[7:0]							ORDER

Bits	Description	
[31:1]	**DMASA**	**DMA Transfer Starting Address** This field pads 0 as least significant bit indicates a 32-bit starting address of system memory (SRAM) for DMA to retrieve or fill in data. If DMA is not in normal mode, this field will be interpreted as a starting address of Physical Address Descriptor (PAD) table.
[0]	**ORDER**	**Determined To The PAD Table Fetching Is In Order Or Out Of Order** 0 = PAD table is fetched in order. 1 = PAD table is fetched out of order. **Note:** the bit0 is valid in scatter-gather mode when SGEN = 1.

Note: Starting address of the SRAM must be word aligned, for example, 0x0000_0000, 0x0000_0004…

The format of PAD table must like below. Note that the total byte count of all Pads must be equal to the byte count filled in SD host. EOT should be set to 1 in the last descriptor.



Physical Base Address: 32-bit
Byte Count: must be multiples of 4 bytes, Max:65532 bytes
Bytes (bit 15~0)
EOT: End of PAD Table (bit 31)

DMA Transfer Byte Count Register (SDH_DMABCNT)

Register	Offset	R/W	Description	Reset Value
SDH_DMABCNT	SDH_BA+0x40C	R	DMA Transfer Byte Count Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						BCNT[25:24]	
23	22	21	20	19	18	17	16
BCNT[23:16]							
15	14	13	12	11	10	9	8
BCNT[15:8]							
7	6	5	4	3	2	1	0
BCNT[7:0]							

Bits	Description	
[31:26]	Reserved	Reserved.
[25:0]	BCNT	**DMA Transfer Byte Count (Read Only)** This field indicates the remained byte count of DMA transfer. The value of this field is valid only when DMA is busy; otherwise, it is 0.

DMA Interrupt Enable Control Register (SDH_DMAINTEN)

Register	Offset	R/W	Description	Reset Value
SDH_DMAINTEN	SDH_BA+0x410	R/W	DMA Interrupt Enable Control Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						WEOTIEN	ABORTIEN

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**WEOTIEN**	**Wrong EOT Encountered Interrupt Enable Bit** 0 = Interrupt generation Disabled when wrong EOT is encountered. 1 = Interrupt generation Enabled when wrong EOT is encountered.
[0]	**ABORTIEN**	**DMA Read/Write Target Abort Interrupt Enable Bit** 0 = Target abort interrupt generation Disabled during DMA transfer. 1 = Target abort interrupt generation Enabled during DMA transfer.

DMA Interrupt Status Register (SDHDMACISR)

Register	Offset	R/W	Description	Reset Value
SDH_DMAINTSTS	SDH_BA+0x414	R/W	DMA Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						WEOTIF	ABORTIF

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**WEOTIF**	**Wrong EOT Encountered Interrupt Flag** When DMA Scatter-Gather function is enabled, and EOT of the descriptor is encountered before DMA transfer finished (that means the total sector count of all PAD is less than the sector count of SD host), this bit will be set. 0 = No EOT encountered before DMA transfer finished. 1 = EOT encountered before DMA transfer finished. **Note:** This bit is read only, but can be cleared by writing '1' to it.
[0]	**ABORTIF**	**DMA Read/Write Target Abort Interrupt Flag** 0 = No bus ERROR response received. 1 = Bus ERROR response received. **Note:** This bit is read only, but can be cleared by writing '1' to it.

Note: When DMA's bus master received ERROR response, it means that target abort is happened. DMA will stop transfer and respond this event and then go to IDLE state. When target abort occurred or WEOTIF is set, software must reset DMA and SD host, and then transfer those data again.

Global Control and Status Register (SDH_GCTL)

Register	Offset	R/W	Description	Reset Value
SDH_GCTL	SDH_BA+0x800	R/W	Global Control and Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						SDEN	GCTLRST

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**SDEN**	**Secure Digital Functionality Enable Bit** 0 = SD functionality disabled. 1 = SD functionality enabled.
[0]	**GCTLRST**	**Software Engine Reset** 0 = No effect. 1 = Reset SD host. The contents of control register will not be cleared. This bit will auto cleared after reset complete.

Global Interrupt Control Register (SDH_GINTEN)

Register	Offset	R/W	Description	Reset Value
SDH_GINTEN	SDH_BA+0x804	R/W	Global Interrupt Control Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							DTAIEN

Bits	Description	
[31:1]	Reserved	Reserved.
[0]	DTAIEN	**DMA READ/WRITE Target Abort Interrupt Enable Bit** 0 = DMA READ/WRITE target abort interrupt generation disabled. 1 = DMA READ/WRITE target abort interrupt generation enabled.

Global Interrupt Status Register (SDH_GINTSTS)

Register	Offset	R/W	Description	Reset Value
SDH_GINTSTS	SDH_BA+0x808	R/W	Global Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							DTAIF

Bits	Description	
[31:1]	Reserved	Reserved.
[0]	DTAIF	**DMA READ/WRITE Target Abort Interrupt Flag (Read Only)** This bit indicates DMA received an ERROR response from internal AHB bus during DMA read/write operation. When Target Abort is occurred, please reset all engine. 0 = No bus ERROR response received. 1 = Bus ERROR response received. **Note:** This bit is read only, but can be cleared by writing '1' to it.

SD Control and Status Register (SDH_CTL)

Register	Offset	R/W	Description	Reset Value
SDH_CTL	SDH_BA+0x820	R/W	SD Control and Status Register	0x0101_0000

31	30	29	28	27	26	25	24
CLKKEEP1	SDPORT		Reserved	SDNWR			
23	22	21	20	19	18	17	16
BLKCNT							
15	14	13	12	11	10	9	8
DBW	CTLRST	CMDCODE					
7	6	5	4	3	2	1	0
CLKKEEP0	CLK8OEN	CLK74OEN	R2EN	DOEN	DIEN	RIEN	COEN

Bits	Description	
[31]	**CLKKEEP1**	**SD Clock Enable Control for Port 1** 0 = SD clock generation Disabled. 1 = SD clock always keeps free running.
[30:29]	**SDPORT**	**SD Port Selection** 00 = Port 0 selected. 01 = Port 1 selected. Other = Reserved.
[28]	**Reserved**	Reserved.
[27:24]	**SDNWR**	**NWR Parameter For Block Write Operation** This value indicates the NWR parameter for data block write operation in SD clock counts. The actual clock cycle will be SDNWR+1.
[23:16]	**BLKCNT**	**Block Counts To Be Transferred Or Received** This field contains the block counts for data-in and data-out transfer. For READ_MULTIPLE_BLOCK and WRITE_MULTIPLE_BLOCK command, software can use this function to accelerate data transfer and improve performance. Don't fill 0x0 to this field. **Note:** For READ_MULTIPLE_BLOCK and WRITE_MULTIPLE_BLOCK command, the actual total length is BLKCNT * (BLKLEN +1).
[15]	**DBW**	**SD Data Bus Width (For 1-Bit / 4-Bit Selection)** 0 = Data bus width is 1-bit. 1 = Data bus width is 4-bit.
[14]	**CTLRST**	**Software Engine Reset** 0 = No effect. 1 = Reset the internal state machine and counters. The contents of control register will not be cleared (but RIEN, DIEN, DOEN and R2_EN will be cleared). This bit will be auto cleared after few clock cycles.

[13:8]	**CMDCODE**	**SD Command Code** This register contains the SD command code (0x00 – 0x3F).
[7]	**CLKKEEP0**	**SD Clock Enable Control for Port 0** 0 = SD clock generation Disabled. 1 = SD clock always keeps free running.
[6]	**CLK8OEN**	**Generating 8 Clock Cycles Output Enable Bit** 0 = No effect. (Please use DMARST (SDH_CTL [0]) to clear this bit.) 1 = Enabled, SD host will output 8 clock cycles. **Note:** When operation is finished, this bit will be cleared automatically, so don't write 0 to this bit (the controller will be abnormal).
[5]	**CLK74OEN**	**Initial 74 Clock Cycles Output Enable Bit** 0 = No effect. (Please use DMARST (SDH_CTL [0]) to clear this bit.) 1 = Enabled, SD host will output 74 clock cycles to SD card. **Note:** When operation is finished, this bit will be cleared automatically, so don't write 0 to this bit (the controller will be abnormal).
[4]	**R2EN**	**Response R2 Input Enable Bit** 0 = No effect. (Please use DMARST (SDH_CTL [0]) to clear this bit.) 1 = Enabled, SD host will wait to receive a response R2 from SD card and store the response data into DMC's flash buffer (exclude CRC7). **Note:** When operation is finished, this bit will be cleared automatically, so don't write 0 to this bit (the controller will be abnormal).
[3]	**DOEN**	**Data Output Enable Bit** 0 = No effect. (Please use DMARST (SDH_CTL [0]) to clear this bit.) 1 = Enabled, SD host will transfer block data and the CRC16 value to SD card. **Note:** When operation is finished, this bit will be cleared automatically, so don't write 0 to this bit (the controller will be abnormal).
[2]	**DIEN**	**Data Input Enable Bit** 0 = No effect. (Please use DMARST (SDH_CTL [0]) to clear this bit.) 1 = Enabled, SD host will wait to receive block data and the CRC16 value from SD card. **Note:** When operation is finished, this bit will be cleared automatically, so don't write 0 to this bit (the controller will be abnormal).
[1]	**RIEN**	**Response Input Enable Bit** 0 = No effect. (Please use DMARST (SDH_CTL [0]) to clear this bit.) 1 = Enabled, SD host will wait to receive a response from SD card. **Note:** When operation is finished, this bit will be cleared automatically, so don't write 0 to this bit (the controller will be abnormal).
[0]	**COEN**	**Command Output Enable Bit** 0 = No effect. (Please use DMARST (SDH_CTL [0]) to clear this bit.) 1 = Enabled, SD host will output a command to SD card. **Note:** When operation is finished, this bit will be cleared automatically, so don't write 0 to this bit (the controller will be abnormal).

SD Command Argument Register (SDH_CMDARG)

Register	Offset	R/W	Description	Reset Value
SDH_CMDARG	SDH_BA+0x824	R/W	SD Command Argument Register	0x0000_0000

31	30	29	28	27	26	25	24
ARGUMENT							
23	22	21	20	19	18	17	16
ARGUMENT							
15	14	13	12	11	10	9	8
ARGUMENT							
7	6	5	4	3	2	1	0
ARGUMENT							

Bits	Description	
[31:0]	ARGUMENT	**SD Command Argument** This register contains a 32-bit value specifies the argument of SD command from host controller to SD card. Before trigger COEN (SDH_CTL [0]), software should fill argument in this field.

SD Interrupt Control Register (SDH_INTEN)

Register	Offset	R/W	Description	Reset Value
SDH_INTEN	SDH_BA+0x828	R/W	SD Interrupt Control Register	0x0000_0A00

31	30	29	28	27	26	25	24
CDSRC1	CDSRC0	Reserved					
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	WKIEN	DITOIEN	RTOIEN	Reserved		CDIEN1	CDIEN0
7	6	5	4	3	2	1	0
Reserved						CRCIEN	BLKDIEN

Bits	Description	
[31]	**CDSRC1**	**SD1 Card Detect Source Selection** 0 = From SD1 card's DAT3 pin. Host need clock to got data on pin DAT3. Please make sure CLKKEEP1 (SDH_CTL[31]) is 1 in order to generate free running clock for DAT3 pin. 1 = From GPIO pin.
[30]	**CDSRC0**	**SD0 Card Detect Source Selection** 0 = From SD0 card's DAT3 pin. Host need clock to got data on pin DAT3. Please make sure CLKKEEP0 (SDH_CTL[7]) is 1 in order to generate free running clock for DAT3 pin. 1 = From GPIO pin.
[29:15]	**Reserved**	Reserved.
[14]	**WKIEN**	**Wake-Up Signal Generating Enable Bit** Enable/Disable wake-up signal generating of SD host when current using SD card issues an interrupt (wake-up) via DAT [1] to host. 0 = Disabled. 1 = Enabled.
[13]	**DITOIEN**	**Data Input Time-Out Interrupt Enable Bit** Enable/Disable interrupts generation of SD controller when data input time-out. Time-out value is specified at TOUT register. 0 = Disabled. 1 = Enabled.
[12]	**RTOIEN**	**Response Time-Out Interrupt Enable Bit** Enable/Disable interrupts generation of SD controller when receiving response or R2 time-out. Time-out value is specified at TOUT register. 0 = Disabled. 1 = Enabled.

[11:10]	**Reserved**	Reserved.
[9]	**CDIEN1**	**SD1 Card Detection Interrupt Enable Bit** Enable/Disable interrupts generation of SD controller when card 1 is inserted or removed. 0 = Disable. 1 = Enabled.
[8]	**CDIEN0**	**SD0 Card Detection Interrupt Enable Bit** Enable/Disable interrupts generation of SD controller when card 0 is inserted or removed. 0 = Disable. 1 = Enabled.
[7:2]	**Reserved**	Reserved.
[1]	**CRCIEN**	**CRC7, CRC16 And CRC Status Error Interrupt Enable Bit** 0 = SD host will not generate interrupt when CRC7, CRC16 and CRC status is error. 1 = SD host will generate interrupt when CRC7, CRC16 and CRC status is error.
[0]	**BLKDIEN**	**Block Transfer Done Interrupt Enable Bit** 0 = SD host will not generate interrupt when data-in (out) transfer done. 1 = SD host will generate interrupt when data-in (out) transfer done.

SD Interrupt Status Register (SDH_INTSTS)

Register	Offset	R/W	Description	Reset Value
SDH_INTSTS	SDH_BA+0x82C	R/W	SD Interrupt Status Register	0x000X_008C

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved					DAT1STS	CDSTS1	CDSTS0
15	14	13	12	11	10	9	8
Reserved		DITOIF	RTOIF	Reserved		CDIF1	CDIF0
7	6	5	4	3	2	1	0
DAT0STS	CRCSTS			CRC16	CRC7	CRCIF	BLKDIF

Bits	Description	
[31:19]	**Reserved**	Reserved.
[18]	**DAT1STS**	**DAT1 Pin Status Of SD Port (Read Only)** This bit indicates the DAT1 pin status of SD port.
[17]	**CDSTS1**	**Card Detect Status Of SD1 (Read Only)** This bit indicates the card detect pin status of SD1, and is used for card detection. When there is a card inserted in or removed from SD1, software should check this bit to confirm if there is really a card insertion or removal. If CDSRC1 (SDH_INTEN[31]) = 0, to select DAT3 for card detection:. 0 = Card removed. 1 = Card inserted. If CDSRC1 (SDH_INTEN[31]) = 1, to select GPIO for card detection:. 0 = Card inserted. 1 = Card removed.
[16]	**CDSTS0**	**Card Detect Status Of SD0 (Read Only)** This bit indicates the card detect pin status of SD0, and is used for card detection. When there is a card inserted in or removed from SD0, software should check this bit to confirm if there is really a card insertion or removal. If CDSRC0 (SDH_INTEN[30]) = 0, to select DAT3 for card detection:. 0 = Card removed. 1 = Card inserted. If CDSRC0 (SDH_INTEN[30]) = 1, to select GPIO for card detection:. 0 = Card inserted. 1 = Card removed.
[15:14]	**Reserved**	Reserved.

[13]	**DITOIF**	**Data Input Time-Out Interrupt Flag (Read Only)** This bit indicates that SD host counts to time-out value when receiving data (waiting start bit). 0 = Not time-out. 1 = Data input time-out. **Note:** This bit is read only, but can be cleared by writing '1' to it.
[12]	**RTOIF**	**Response Time-Out Interrupt Flag (Read Only)** This bit indicates that SD host counts to time-out value when receiving response or R2 (waiting start bit). 0 = Not time-out. 1 = Response time-out. **Note:** This bit is read only, but can be cleared by writing '1' to it.
[11:10]	**Reserved**	Reserved.
[9]	**CDIF1**	**SD1 Card Detection Interrupt Flag (Read Only)** This bit indicates that SD card 1 is inserted or removed. Only when CDIEN1 (SDH_INTEN[9]) is set to 1, this bit is active. 0 = No card is inserted or removed. 1 = There is a card inserted in or removed from SD1. **Note:** This bit is read only, but can be cleared by writing '1' to it.
[8]	**CDIF0**	**SD0 Card Detection Interrupt Flag (Read Only)** This bit indicates that SD card 0 is inserted or removed. Only when CDIEN0 (SDH_INTEN[8]) is set to 1, this bit is active. 0 = No card is inserted or removed. 1 = There is a card inserted in or removed from SD0. **Note:** This bit is read only, but can be cleared by writing '1' to it.
[7]	**DAT0STS**	**DAT0 Pin Status Of Current Selected SD Port (Read Only)** This bit is the DAT0 pin status of current selected SD port.
[6:4]	**CRCSTS**	**CRC Status Value Of Data-Out Transfer (Read Only)** SD host will record CRC status of data-out transfer. Software could use this value to identify what type of error is during data-out transfer. 010 = Positive CRC status. 101 = Negative CRC status. 111 = SD card programming error occurs.
[3]	**CRC16**	**CRC16 Check Status Of Data-In Transfer (Read Only)** SD host will check CRC16 correctness after data-in transfer. 0 = Fault. 1 = OK.
[2]	**CRC7**	**CRC7 Check Status (Read Only)** SD host will check CRC7 correctness during each response in. If that response does not contain CRC7 information (ex. R3), then software should turn off CRCIEN (SDH_INTEN[1]) and ignore this bit. 0 = Fault. 1 = OK.

[1]	**CRCIF**	**CRC7, CRC16 And CRC Status Error Interrupt Flag (Read Only)** This bit indicates that SD host has occurred CRC error during response in, data-in or data-out (CRC status error) transfer. When CRC error is occurred, software should reset SD engine. Some response (ex. R3) doesn't have CRC7 information with it; SD host will still calculate CRC7, get CRC error and set this flag. In this condition, software should ignore CRC error and clears this bit manually. 0 = No CRC error is occurred. 1 = CRC error is occurred. **Note:** This bit is read only, but can be cleared by writing '1' to it.
[0]	**BLKDIF**	**Block Transfer Done Interrupt Flag (Read Only)** This bit indicates that SD host has finished all data-in or data-out block transfer. If there is a CRC16 error or incorrect CRC status during multiple block data transfer, the transfer will be broken and this bit will also be set. 0 = Not finished yet. 1 = Done. **Note:** This bit is read only, but can be cleared by writing '1' to it.

SD Receiving Response Token Register 0 (SDH_RESP0)

Register	Offset	R/W	Description	Reset Value
SDH_RESP0	SDH_BA+0x830	R	SD Receiving Response Token Register 0	0x0000_0000

31	30	29	28	27	26	25	24
RESPTK0							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESPTK0							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESPTK0							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESPTK0							

Bits	Description	
[31:0]	**RESPTK0**	**SD Receiving Response Token 0** SD host controller will receive a response token for getting a reply from SD card when RIEN (SDH_CTL[1]) is set. This field contains response bit 47-16 of the response token.

SD Receiving Response Token Register 1 (SDH_RESP1)

Register	Offset	R/W	Description	Reset Value
SDH_RESP1	SDH_BA+0x834	R	SD Receiving Response Token Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
RESPTK1							

Bits	Description	
[7:0]	RESPTK1	**SD Receiving Response Token 1** SD host controller will receive a response token for getting a reply from SD card when RIEN (SDH_CTL[1]) is set. This register contains the bit 15-8 of the response token.

SD Block Length Register (SDH_BLEN)

Register	Offset	R/W	Description	Reset Value
SDH_BLEN	SDH_BA+0x838	R/W	SD Block Length Register	0x0000_01FF

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					BLKLEN		
7	6	5	4	3	2	1	0
BLKLEN							

Bits	Description	
[10:0]	**BLKLEN**	**SD BLOCK LENGTH In Byte Unit** An 11-bit value specifies the SD transfer byte count of a block. The actual byte count is equal to BLKLEN+1. **Note:** The default SD block length is 512 bytes

SD Response/Data-in Time-out Register (SDH_TOUT)

Register	Offset	R/W	Description	Reset Value
SDH_TOUT	SDH_BA+0x83C	R/W	SD Response/Data-in Time-out Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
TOUT							
15	14	13	12	11	10	9	8
TOUT							
7	6	5	4	3	2	1	0
TOUT							

Bits	Description	
[23:0]	TOUT	**SD Response/Data-In Time-Out Value** A 24-bit value specifies the time-out counts of response and data input. SD host controller will wait start bit of response or data-in until this value reached. The time period depends on SD engine clock frequency. Do not write a small number into this field, or you may never get response or data due to time-out. **Note:** Filling 0x0 into this field will disable hardware time-out function.

6.27 Serial Peripheral Interface (SPI)

6.27.1 Overview

The Serial Peripheral Interface (SPI) is a synchronous serial data communication protocol that operates in full duplex mode. Devices communicate in Master/Slave mode with the 4-wire bi-direction interface. The NUC442/NUC472 series contains up to four sets of SPI controllers performing a serial-to-parallel conversion on data received from a peripheral device, and a parallel-to-serial conversion on data transmitted to a peripheral device. Each set of SPI controller can be configured as a master or a slave device.

The SPI controller supports 2-bit Transfer mode to connect 2 off-chip slave devices at the same time. This controller also supports the PDMA function to access the data buffer and also supports Dual and Quad I/O Transfer mode.

6.27.2 Features

- Up to four sets of SPI controllers
- Supports Master or Slave mode operation
- Supports 2-bit Transfer mode
- Supports Dual and Quad I/O Transfer mode
- Configurable bit length of a transfer word from 8 to 32-bit
- Provides separate 8-level depth transmit and receive FIFO buffers
- Supports MSB first or LSB first transfer sequence
- Two slave select lines in Master mode
- Supports byte reorder function
- Supports Byte or Word Suspend mode
- Supports PDMA transfer
- Supports 3-wire, no slave select signal, bi-direction interface

6.27.3 Block Diagram

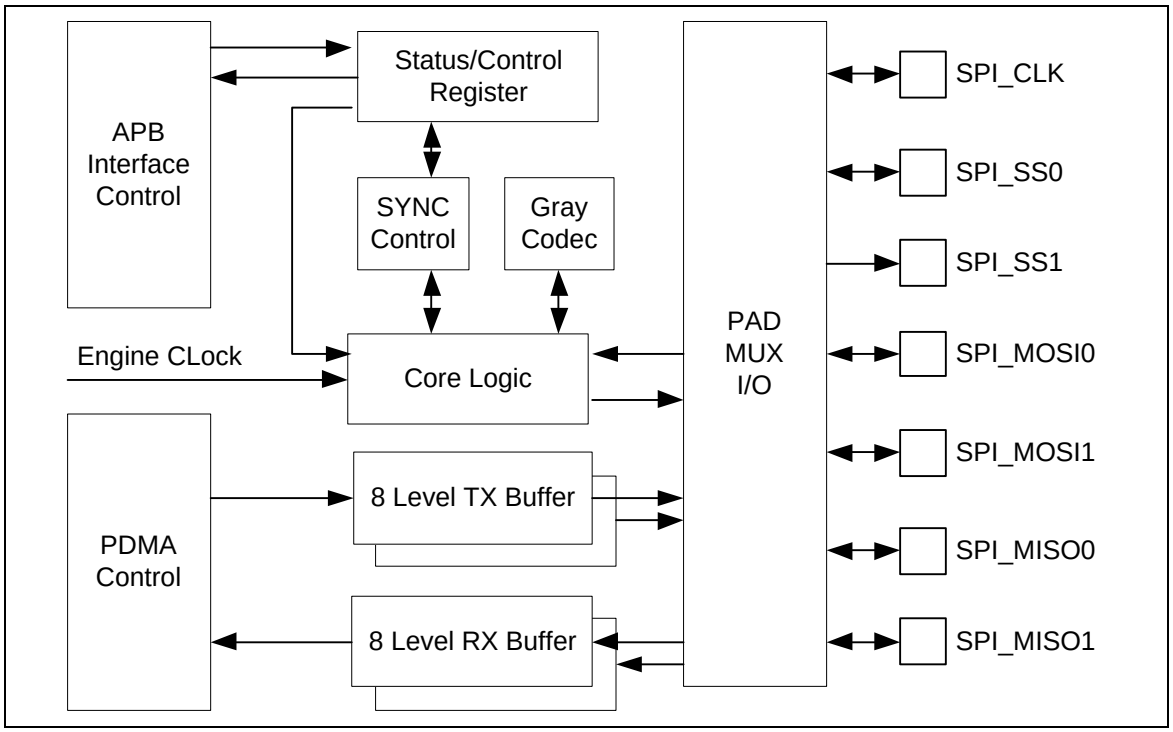


Figure 6.27-1 SPI Block Diagram

6.27.4 Functional Description

Peripheral Clock and SPI Bus Clock

The SPI controller needs the peripheral clock to drive the SPI logic unit to perform the data transfer. The peripheral clock rate is determined by the SPI_CLKDIV register. Each bit of the CLK_CLKSEL1[7:4] register determines the clock source of the peripheral clock. The clock source can be HCLK or PLL output clock. The DIVIDER setting of SPI_CLKDIV register determines the divisor of the clock rate calculation.

In Master mode, the output frequency of the SPI bus clock output pin is equal to the peripheral clock rate. In general, the SPI bus clock is denoted as SPI clock. In Slave mode, the SPI bus clock is provided by an off-chip master device. The peripheral clock rate of slave device must be faster than the SPI bus clock rate of the master device connected together. The frequency of SPI peripheral clock cannot be faster than the APB clock rate regardless of Master or Slave mode. (If the clock sources of peripheral clock and APB clock are different, the frequency of SPI peripheral clock shall be slower than the APB clock rate regardless of Master or Slave mode.)

Master/Slave Mode

This SPI controller can be set as Master or Slave mode by setting the SLAVE bit (SPI_CTL[18]) to communicate with the off-chip SPI slave or master device. The application block diagrams in master and Slave mode are shown below.

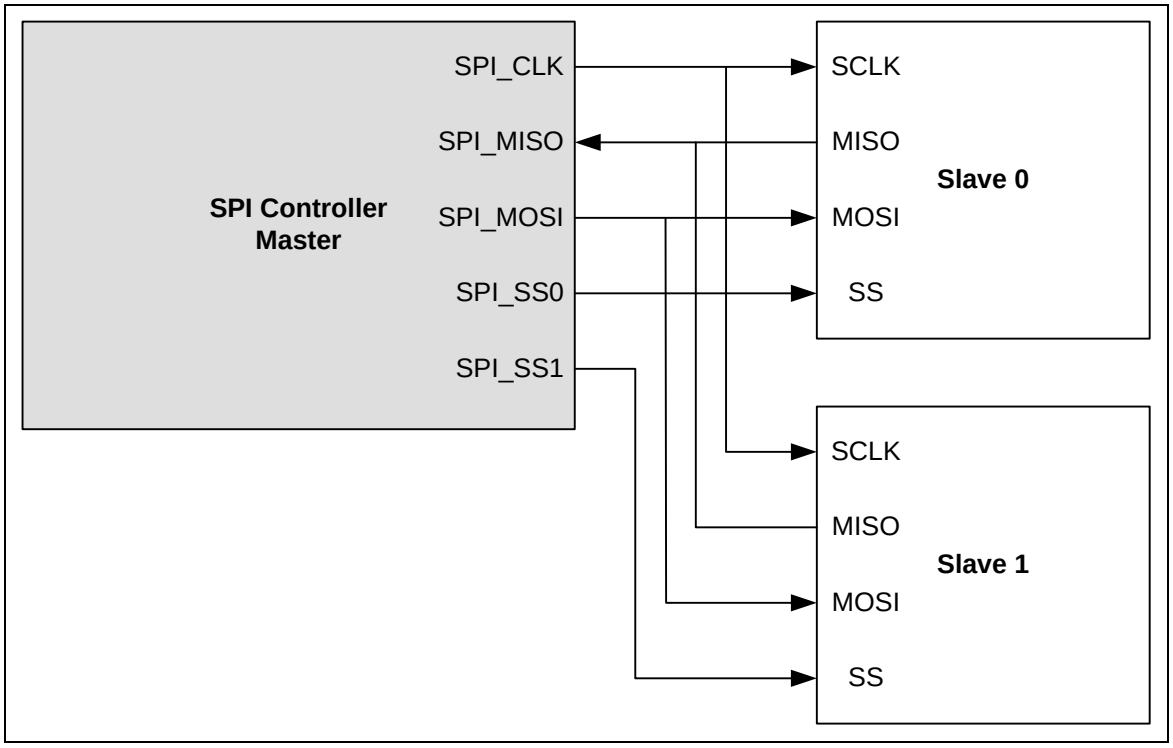


Figure 6.27-2 SPI Master Mode Application Block Diagram

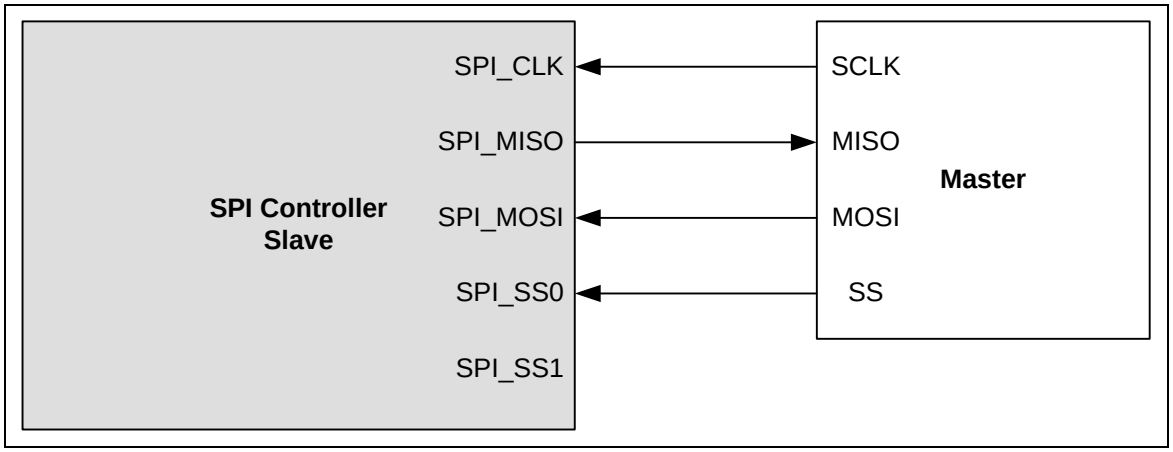


Figure 6.27-3 SPI Slave Mode Application Block Diagram

Slave Selection

In Master mode, the SPI controller can drive up to two off-chip slave devices through the slave select output pins SPI_SS0 and SPI_SS1. In Slave mode, the off-chip master device drives the slave select signal from the SPI_SS0 input port to this SPI controller. In Master/Slave mode, the active state of slave select signal can be programmed to low or high active in SSACTPOL bit (SPI_SSCTL[2]). The selection of slave select conditions depends on what type of peripheral slave/master device is connected.

Automatic Slave Selection

In Master mode, if the bit AUTOSS (SPI_SSCTL[3]) is set, the slave select signals will be generated automatically and output to the SPI_SS0 and SPI_SS1 pins according to whether SS[0] (SPI_SSCTL[0]) and SS[1] (SPI_SSCTL[1]) are enabled or not. This means that the slave select signals, which are selected in SPI_SSCTL[1:0], will be asserted by the SPI controller when the SPI data transfer is started by writing the transfer data into the FIFO and will be de-asserted after one transaction is finished. If the AUTOSS bit is cleared, the slave select output signals will be asserted/de-asserted by manual setting/clearing the related bits of SPI_SSCTL[1:0]. The active state of the slave select output signals is specified in SSACTPOL bit (SPI_SSCTL[2]).

In Master mode, if the value of SUSPITV[3:0] is less than 3 and the AUTOSS is set as 1, the slave select signal will be kept in active state between two successive transactions.

In Slave mode, to recognize the inactive state of the slave select signal, the inactive period of the slave select signal must be larger than or equal to 3 module clock periods between two successive transactions.

Clock Polarity

The CLKPOL bit (SPI_CTL[3]) defines the SPI clock idle state. If CLKPOL = 1, the output SPI clock is idle at high state; if CLKPOL = 0, it is idle at low state.

Transmit/Receive Bit Length

The bit length of a transaction word is defined in DWIDTH bit field (SPI_CTL[12:8]) and can be configured up to 32-bit length in a transaction word for transmitting and receiving.

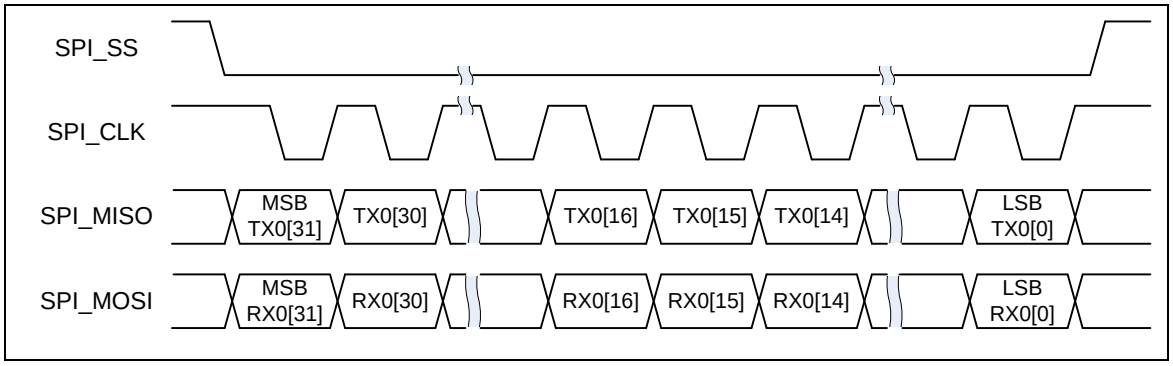


Figure 6.27-4 32-bit in One Transaction

LSB/MSB First

The LSB bit (SPI_CTL[13]) defines the bit transfer sequence in a transaction. If the LSB bit is set to 1, the transfer sequence is LSB first. The bit 0 will be transferred firstly. If the LSB bit is cleared to 0, the transfer sequence is MSB first.

Transmit Edge

The TXNEG bit (SPI_CTL[2]) defines the data transmitted out either on negative edge or on positive edge of SPI clock.

Receive Edge

The RXNEG bit (SPI_CTL[1]) defines the data received either on negative edge or on positive edge of SPI clock.

Note: The settings of TXNEG and RXNEG are mutual exclusive. In other words, do not transmit and receive data at the same clock edge.

Word Suspend

The four bit fields of SUSPITV (SPI_CTL[7:4]) provide a configurable suspend interval, 0.5 ~ 15.5 SPI clock periods, between two successive transaction words in Master mode. The definition of the suspend interval is the interval between the last clock edge of the preceding transaction word and the first clock edge of the following transaction word. The default value of SUSPITV is 0x3 (3.5 SPI clock cycles).

Byte Reorder

When the transfer is set as MSB first (LSB = 0) and the REORDER bit is set to 1, the data stored in the TX buffer and RX buffer will be rearranged in the order as [BYTE0, BYTE1, BYTE2, BYTE3] in 32-bit Transfer mode (DWIDTH = 0). The sequence of transmitted/received data will be BYTE0, BYTE1, BYTE2, and then BYTE3. If the DWIDTH is set as 24-bit transfer mode, the data in TX buffer and RX buffer will be rearranged as [unknown byte, BYTE0, BYTE1, BYTE2]. The SPI controller will transmit/receive data with the sequence of BYTE0, BYTE1 and then BYTE2. Each byte will be transmitted/received with MSB first. The rule of 16-bit mode is the same as above. Byte reorder function is only available when DWIDTH is configured as 16, 24, and 32 bits.

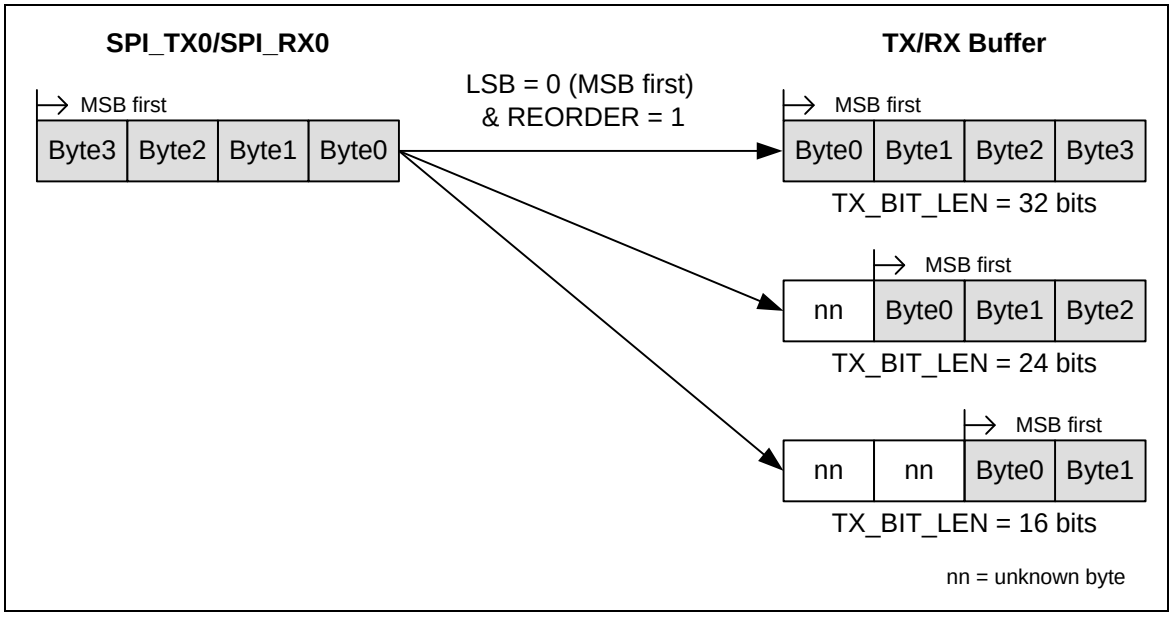


Figure 6.27-5 Byte Reorder Function

Byte Suspend

In Master mode, if REORDER, SPI_CTL[19], is set to 1, a suspend interval of 0.5 ~ 15.5 SPI clock periods will be inserted by hardware between two successive bytes in a transaction word. Both settings of byte suspend interval and word suspend interval are configured in SUSPITV (SPI_CTL[7:4]).

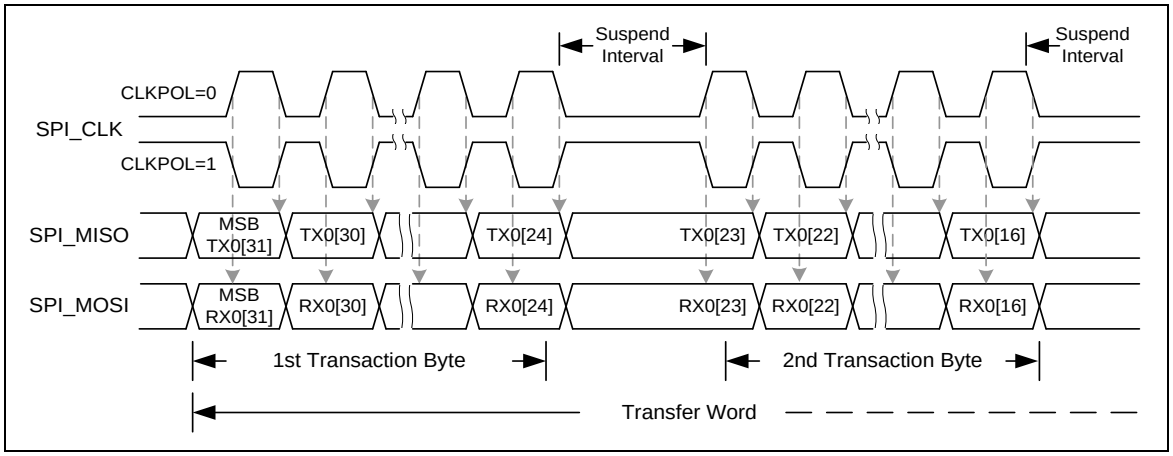


Figure 6.27-6 Timing Waveform for Byte Suspend

3-Wire Mode

When the SLV3WIRE bit is set by software to enable the Slave 3-wire mode, the SPI controller can work with no slave select signal in Slave mode. The SLV3WIRE bit only takes effect in Slave mode. Only three pins, SPI_CLK, SPI_MISO, and SPI_MOSI, are required to communicate with a SPI master. The SPISS pin can be configured as a GPIO. When the SLV3WIRE bit is set to 1, the SPI slave will be ready to transmit/receive data after the SPIEN bit is set to 1.

2-bit Mode

The SPI controller also supports 2-bit Transfer mode when setting the TWOBIT bit (SPI_CTL[16]) to 1. In 2-bit mode, the SPI controller performs full duplex data transfer. In other words, the 2-bit serial data can be transmitted and received simultaneously.

For example, in Master mode, the first data stored in the SPI_TX register will be transmitted through the SPI_MOSI0 and the second data stored in the SPI_TX register will be transmitted through the SPI_MOSI1 pin respectively. In the meanwhile, the first read in SPI_RX will store the data received from SPI_MISO0 pin and the second read in SPI_RX will store the data received from SPI_MISO1 pin.

In Slave mode, the two data stored in the SPI_TX will be transmitted through the SPI_MISO0 and SPI_MISO1 pin respectively. In the meanwhile, the SPI_RX will store the data received from the SPI_MOSI0 and SPI_MOSI1 pin, same as the Master mode.

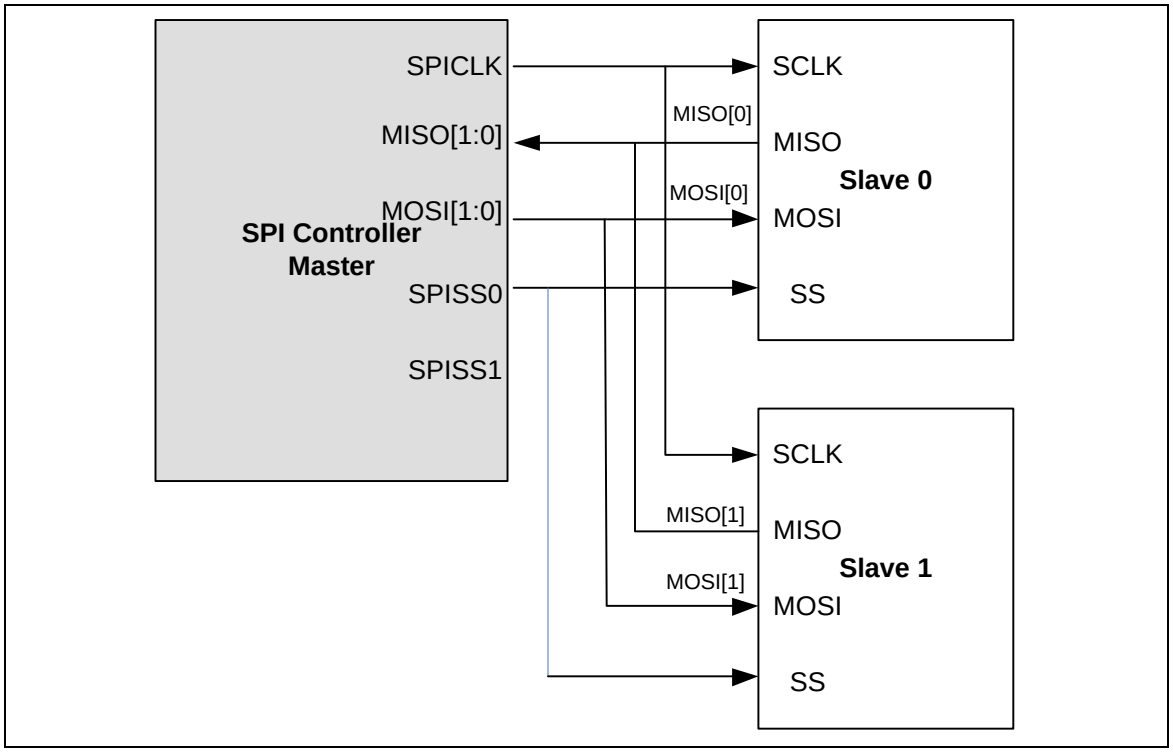


Figure 6.27-7 2-bit Mode System Architecture

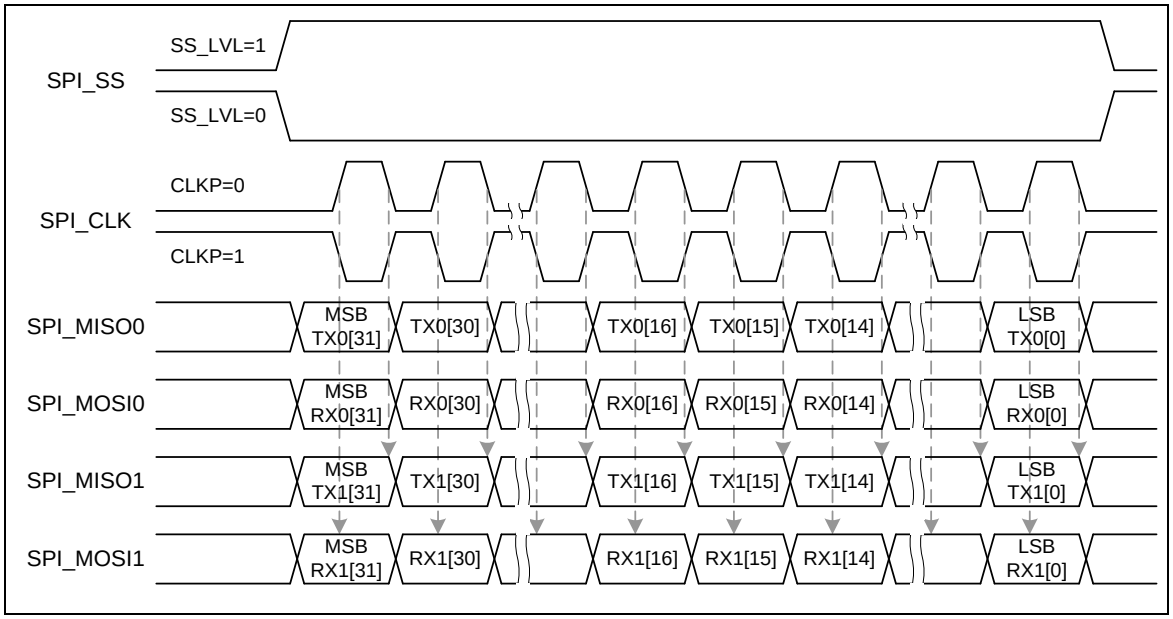


Figure 6.27-8 2-bit Mode (Slave Mode)

Dual and Quad I/O Mode

The SPI controller also supports dual and quad I/O transfer when setting the DUALIOEN bit or the QUADIOEN bit (SPI_CTL[21], SPI_CTL[22]) to 1. Many general SPI flashes support Dual/ Quad I/O transfer. The QDIODIR bit (SPI_CTL[20]) is used to define the direction of the transfer data. When the QDIODIR bit is set to 1, the controller will send the data to external device. When the QDIODIR bit is set to 0, the controller will read the data from the external device. This function supports 8, 16, 24, and 32 bits of length.

The Dual/Quad I/O mode is not supported when the Slave 3-wire mode or the byte reorder function is enabled.

For Dual I/O mode, if both the DUALIOEN and QDIODIR bits are set as 1, the SPI_MOSI0 is the even bit data output and the SPI_MISO0 will be set as the odd bit data output. If the DUALIOEN is set as 1 and QDIODIR is set as 0, both the SPI_MISO0 and SPI_MOSI0 will be set as data input ports.

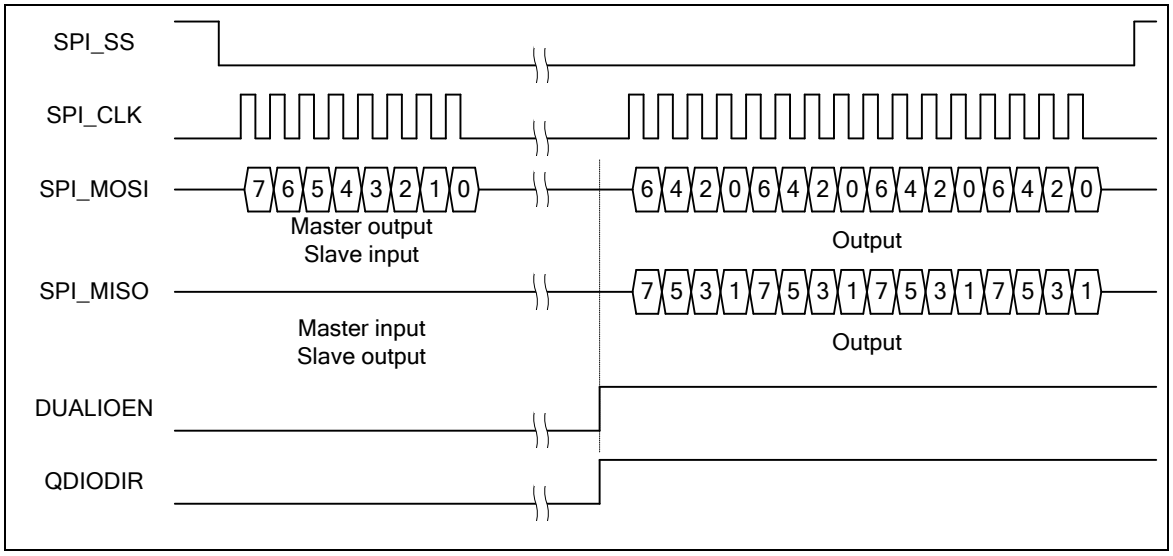


Figure 6.27-9 Bit Sequence of Dual Output Mode

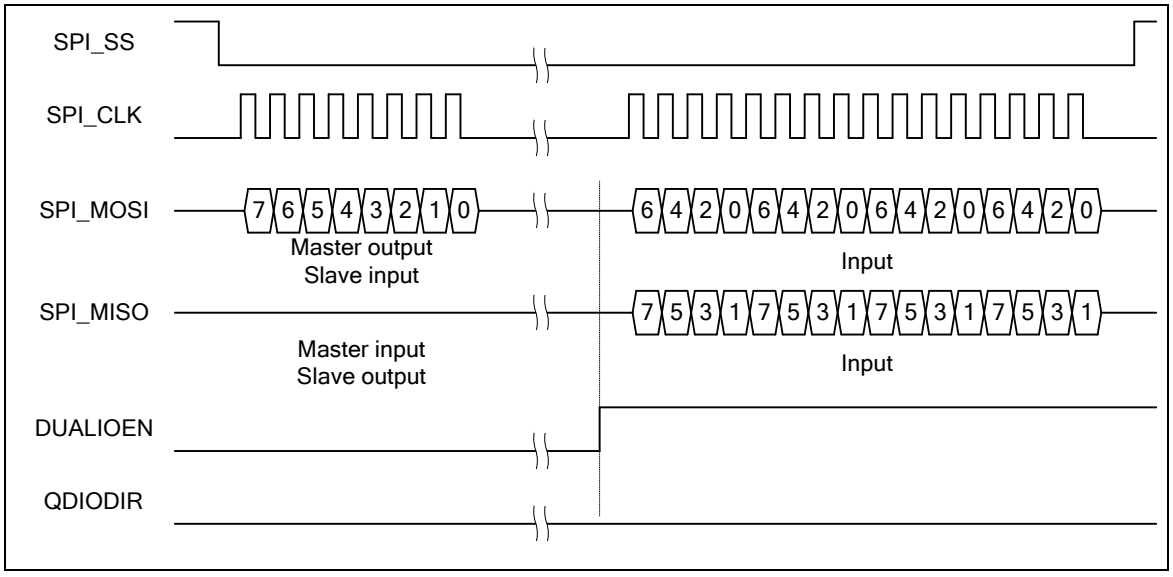


Figure 6.27-10 Bit Sequence of Dual Input Mode

For Quad I/O mode, if both the QUADIOEN and QDIODIR bits are set as 1, the SPI_MOSI0 and SPI_MOSI1 are the even bit data output and the SPI_MISO0 and SPI_MISO1 will be set as the odd bit data output. If the QUADIOEN is set as 1 and QDIODIR is set as 0, both the SPI_MISO0, SPI_MISO1, SPI_MOSI0 and SPI_MOSI1 will be set as data input ports.

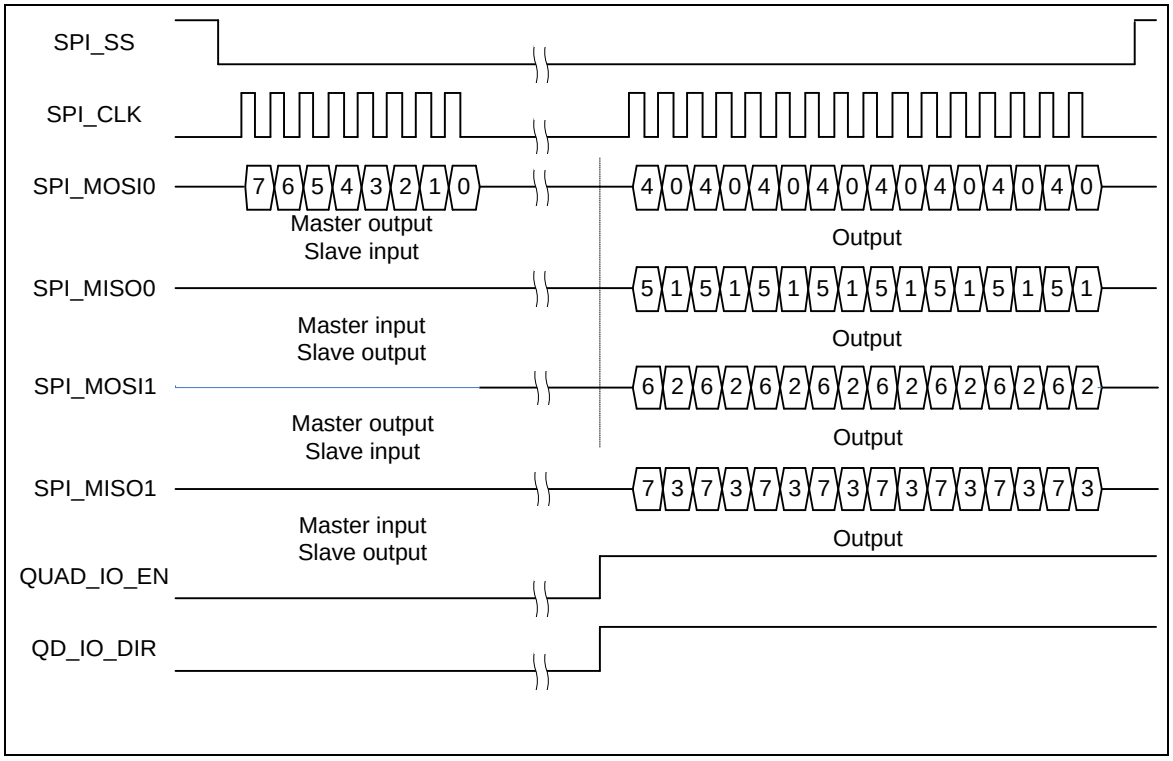


Figure 6.27-11 Bit Sequence of Quad Output Mode

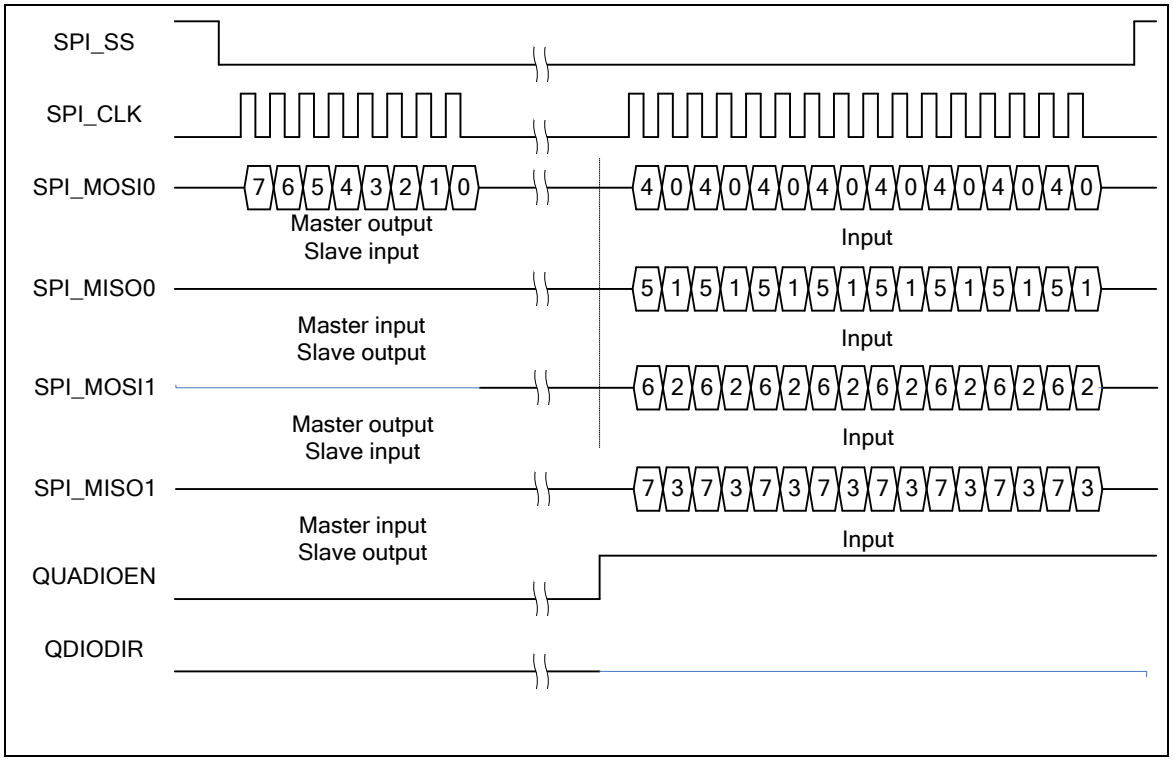


Figure 6.27-12 Bit Sequence of Quad Input Mode

8-Level FIFO Buffer

The SPI controllers equip with eight 32-bit wide transmit and receive FIFO buffers.

The transmit FIFO buffer is an 8-level depth, 32-bit wide, first-in, first-out register buffer. 8 data can be written to the transmit FIFO buffer in advance through software by writing the SPI_TX register. The data stored in the transmit FIFO buffer will be read and sent out by the transmission control logic. If the 8-level transmit FIFO buffer is full, the TXFULL (SPI_STATUS[17]) bit will be set to 1. When the SPI transmission logic unit draws out the last datum of the transmit FIFO buffer, so that the 8-level transmit FIFO buffer is empty, the TXEMPTY (SPI_STATUS[16]) bit will be set to 1. Notice that the TXEMPTY (SPI_STATUS[16]) flag is set to 1 while the last transaction is still in progress. In Master mode, both the BUSY bit (SPI_STATUS[0]) and TXEMPTY (SPI_STATUS[16]) bit should be checked by software to make sure whether the SPI is in idle or not.

The received FIFO buffer is also an 8-level depth, 32-bit wide, first-in, first-out register buffer. The receive control logic will store the received data to this buffer. The FIFO buffer data can be read from SPI_RX register by software. There are FIFO related status bits, like RXEMPTY (SPI_STATUS[8]) and RXFULL (SPI_STATUS[9]), to indicate the current status of FIFO buffer.

The transmitting and receiving threshold can be set through software by setting the TXTH (SPI_FIFOCTL[30:24]) and RXTH (SPI_FIFOCTL[26:24]) settings. When the count of valid data stored in transmit FIFO buffer is less than or equal to TXTH setting, the TXTHIF (SPI_STATUS[18]) bit will be set to 1. When the count of valid data stored in receive FIFO buffer is larger than RXTH (SPI_FIFOCTL[26:24]) setting, the RXTHIF (SPI_STATUS[10]) bit will be set to 1.

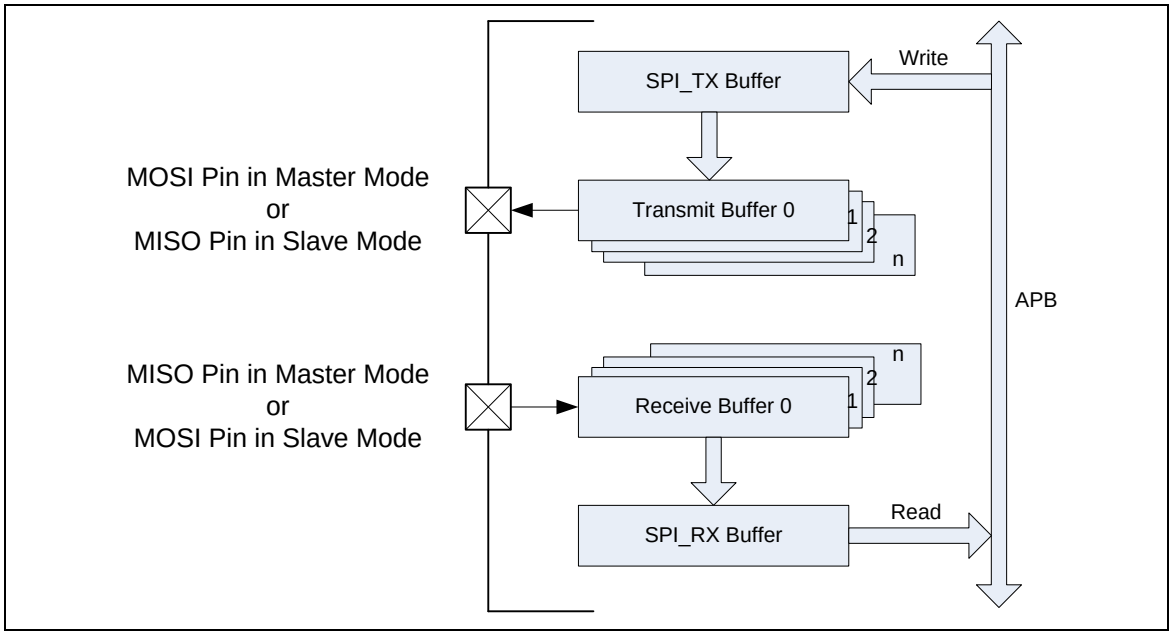


Figure 6.27-13 FIFO Mode Block Diagram

In Master mode, the first datum is written to the SPI_TX register, the TXEMPTY (SPI_STATUS[16]) flag will be cleared to 0. The transmission immediately starts as long as the

transmit FIFO buffer is not empty. User can write the next data into SPI_TX register immediately. The SPI controller will insert a suspend interval between two successive transactions and the period of suspend interval is decided by the setting of SUSPITV (SPI_CTL [7:4]). User can write data into SPI_TX register as long as the TXFULL (SPI_STATUS[17]) flag is 0.

The subsequent transactions will be triggered automatically if the transmitted data are updated in time. If the SPI_TX register does not be updated after all data transfer are done, the transfer will stop.

In Master mode, during receiving operation, the serial data are received from SPI_MISO0/1 pin and stored to receive FIFO buffer. The RXEMPTY (SPI_STATUS[8]) flag will be cleared to 0 while the receive FIFO buffer contains unread data. The received data can be read by software from SPI_RX register as long as the RXEMPTY (SPI_STATUS[8]) flag is 0. If the receive FIFO buffer contains 8 unread data, the RXFULL flag will be set to 1. The SPI controller will stop receiving data until the SPI_RX register is read by software.

In Slave mode, during transmission operation, when data is written to the SPI_TX register by software, the data will be loaded into transmit FIFO buffer and the TXEMPTY (SPI_STATUS[16]) flag will be set to 0. The transmission will start when the slave device receives clock signal from master. Data can be written to SPI_TX register as long as the TXFULL (SPI_STATUS[17]) flag is 0. After all data have been drawn out by the SPI transmission logic unit and the SPI_TX register is not updated by software, the TXEMPTY (SPI_STATUS[16]) flag will be set to 1.

If there is no any data is written to the SPI_TX register, the under-run event, TXUFIF (SPI_STATUS[19]) will active when the slave select active and the serial clock input this controller. Under the previous condition, the Slave modeerror 1, SLVURIF (SPI_STATUS[7]) will be set to 1 when SS (Slave Select) goes to inactive state. (Reference to **Interrupt** section)

In Slave mode, during receiving operation, the serial data is received from SPI_MOSI0/1 pin and stored to SPI_RX register. The reception mechanism is similar to Master mode reception operation. If the receive FIFO buffer contains 8 unread data, the RXFULL (SPI_STATUS[9]) flag will be set to 1 and the RXOVIF (SPI_STATUS[11]) will be set 1 if there is more serial data is received from SPI_MOSI and follow-up data will be dropped. If the receive bit counter mismatch with the DWIDTH when the slave select line goes to inactive state, the Slave mode error 0, SLVBEIF (SPI_STATUS[6]) will be set to 1. (Reference to **Interrupt** section)

When the Slave select is active and the value of SLVTOCNT (SPI_SSCTL[31:16]) is not 0, the Slave time-out counter in the SPI controller logic will start after the serial clock input. This counter will be cleared after one transaction done or the SLVTOCNT (SPI_SSCTL[31:16]) is set to 0. If the value of the time-out counter greater or equal to the value of SLVTOCNT (SPI_SSCTL[31:16]) before one transaction done, the slave time-out event occurs and the SLVTOIF (SPI_STATUS[5]) will be set to 1.

A receive time-out function is built-in in this controller. When the receive FIFO is not empty and no read operation in receive FIFO over 64 SPI clock period in Master mode or over 576 SPI module clock period in Slave mode, the receive time-out occurs and the SLVTOIF (SPI_STATUS[5]) be set to 1. When the receive FIFO is read by user, the time-out status will be cleared automatically.

Interrupt

- SPI unit transfer interrupt

As the SPI controller finishes a unit transfer, the unit transfer interrupt flag UNITIF (SPI_STATUS[1]) will be set to 1. The unit transfer interrupt event will generate an interrupt to CPU if the unit transfer interrupt enable bit UNITIEN (SPI_CTL[17]) is set. The unit transfer interrupt flag can be cleared only by writing 1 to it.

- SPI slave select interrupt

In slave mode, there are slave select active and in-active interrupt flag, SSACTIF

(SPI_STATUS[2]) and SSINAIF (SPI_STATUS[3]) will be set to 1 when the SPIEN (SPI_CTL[0]) and SLAVE (SPI_CTL[18]) bits were set to 1 and slave senses the slave select signal active or inactive. The SPI controller will issue an interrupt if the SSINAIEN (SPI_SSCTL[13]) or SSACTIEN (SPI_SSCTL[12]) are set to 1.

- Slave Time-out interrupt

In Slave mode, there is slave time-out function for user to know that there is serial clock input but one transaction is not finished over the period of SLVTOCNT (SPI_SSCTL[31:16]) basing on module clock.

When the Slave select is active and the value of SLVTOCNT (SPI_SSCTL[31:16]) is not 0, the Slave time-out counter in the SPI controller logic will start after the serial clock input. This counter will be clear after one transaction done or the SLVTOCNT (SPI_SSCTL[31:16]) is set to 0. If the value of the time-out counter greater or equal than the value of SLVTOCNT (SPI_SSCTL[31:16]) before one transaction done, the slave time-out event occurs and the SLVTOIF (SPI_STATUS[5]) will be set to 1. The SPI controller will issue an interrupt if the SLVTOIEN (SPI_SSCTL[5]) is set to 1.

- Slave Error 0 interrupt

In Slave mode, if the transmit/ receive bit count mismatch with the DWIDTH (SPI_CTL[12:8]) when the slave select line goes to inactive state, the Slave mode error 0, SLVBEIF (SPI_STATUS[6]) will be set to 1. The SPI controller will issue an interrupt if the SLVBEIEN (SPI_SSCTL[8]) is set to 1.

Note: 1. In Slave transmit mode, if there is bit length transmit error (bit count mismatch), the user shall set the TXRST (SPI_FIFOCTL[1]) bit and write the transmit datum again to restart the next transaction.

2. If the slave select active but there is no any serial clock input, the SLVBEIF (SPI_STATUS[6]) also active when the slave select goes to inactive state.

- Slave Error 1 interrupt

In Slave mode, if there is no any data is written to the SPI_TX register, the under-run event, TXUFIF (SPI_STATUS[19]) will be active when the slave select is active and the serial clock input this controller. The SPI controller will issue an interrupt if the TXUFIEN (SPI_FIFOCTL[7]) is set to 1.

Under the previous condition, the Slave mode error 1, SLVURIF (SPI_STATUS[7]) will be set to 1 when SS goes to inactive state and transmit under-run occurs. The SPI controller will issue an interrupt if the SLVURIEN (SPI_SSCTL[9]) is set to 1.

Note: In Slave 3-wire mode, the slave select bus active all the time so that the user shall polling the TXUFIF (SPI_STATUS[19]) bit to know if there is transmit under-run event or not.

- Receive Overrun interrupt

In Slave mode, if the receive FIFO buffer contains 8 unread data, the RXFULL (SPI_STATUS[9]) flag will be set to 1 and the RXOVIF (SPI_STATUS[11]) will be set 1 if there is more serial data is received from SPI_MOSI and the RXOVIF (SPI_STATUS[11]) will be set to 1 and follow-up data will be dropped. The SPI controller will issue an interrupt if the RXOVIEN (SPI_FIFOCTL[5]) is set to 1.

- Receive FIFO time-out interrupt

In FIFO mode, there is a time-out function to inform user. If there is a received data in the FIFO and it is not read by software over 64 SPI module clock periods in Master mode or over 576 SPI module clock periods in Slave mode, it will send a time-out interrupt to the system if the time-out interrupt enable bit, RXTOIEN (SPI_FIFOCTL[4]) is set to 1.

- Transmit FIFO interrupt

In FIFO mode, if the valid data count of the transmit FIFO buffer is less than or equal to the

setting value of TXTH (SPI_FIFOCTL[30:28]), the transmit FIFO interrupt flag will be set to 1. The SPI controller will generate a transmit FIFO interrupt to the system if the transmit FIFO interrupt enable bit (SPI_FIFOCTL[3]) is set to 1.

- Receive FIFO interrupt

In FIFO mode, if the valid data count of the receive FIFO buffer is larger than the setting value of RXTH (SPI_FIFOCTL[26:24]), the receive FIFO interrupt flag will be set to 1. The SPI controller will generate a receive FIFO interrupt to the system if the receive FIFO interrupt enable bit RXTHIEN (SPI_FIFOCTL[2]) is set to 1.

Programming Note

In the SPIEN (SPI_CTL[0]) bit description, it has mentioned that "All configurations should be set before writing 1 to this SPIEN (SPI_CTL[0]) bit." To change the configuration including the registers of SPI_CTL, SPI_CLKDIV, SPI_SSCTL and SPI_FIFOCTL, the user shall clear the SPIEN (SPI_CTL[0]) bit and then confirm the SPIENSTS bit (SPI_STATUS[15]) is '0'.

6.27.5 Timing Diagram

The active state of slave select signal can be defined by setting the SSACTPOL (SPI_SSCTL[2]). The SPI clock which is in idle state can be configured as high or low state by setting the CLKPOL (SPI_CTL[3]). It also provides the bit length of a transaction word in DWIDTH (SPI_CTL[12:8]), and transmitting/receiving data from MSB or LSB first in LSB (SPI_CTL[13]). User can also select which edge of SPI clock to transmit/receive data in TXNEG/RXNEG (SPI_CTL[2:1]). Four SPI timing diagrams of master/slave operations and the related settings are shown below.

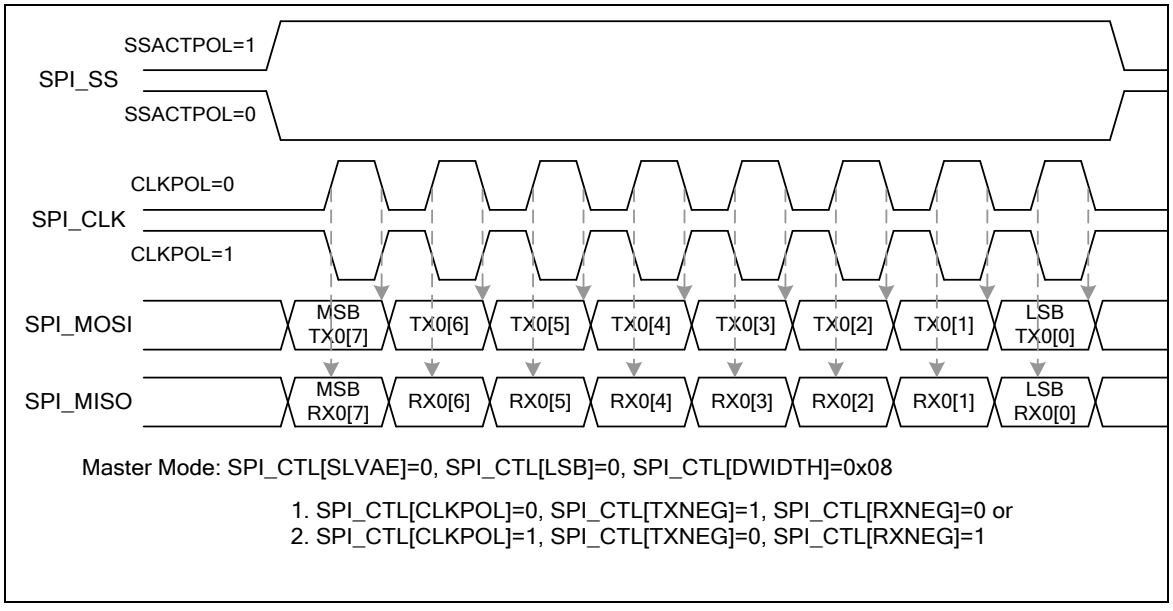


Figure 6.27-14 SPI Timing in Master Mode

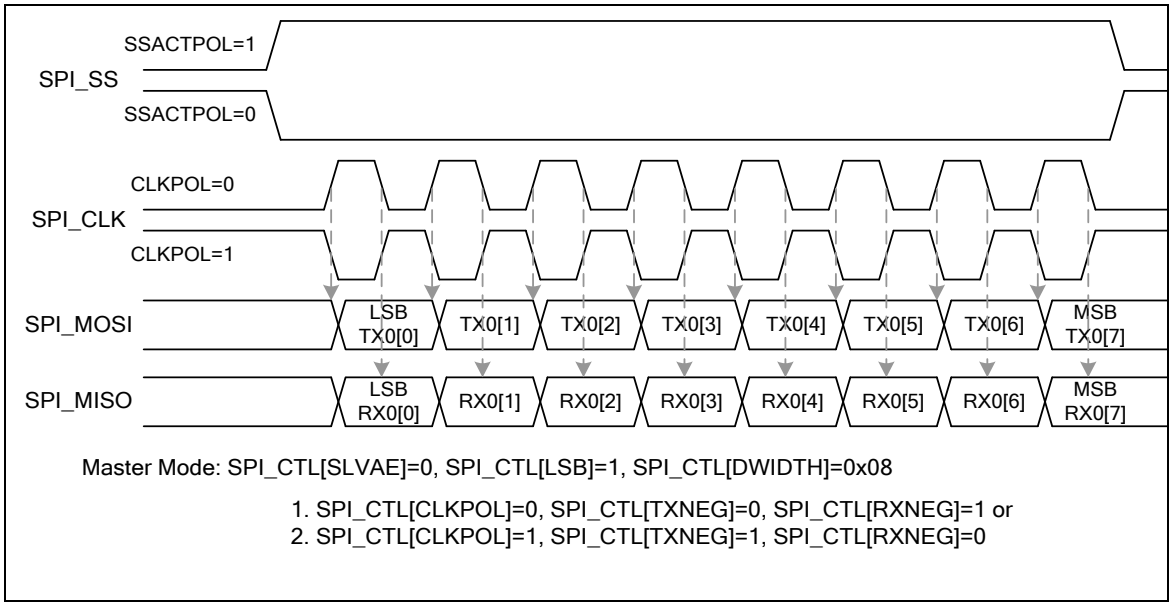


Figure 6.27-15 SPI Timing in Master Mode (Alternate Phase of SPICLK)

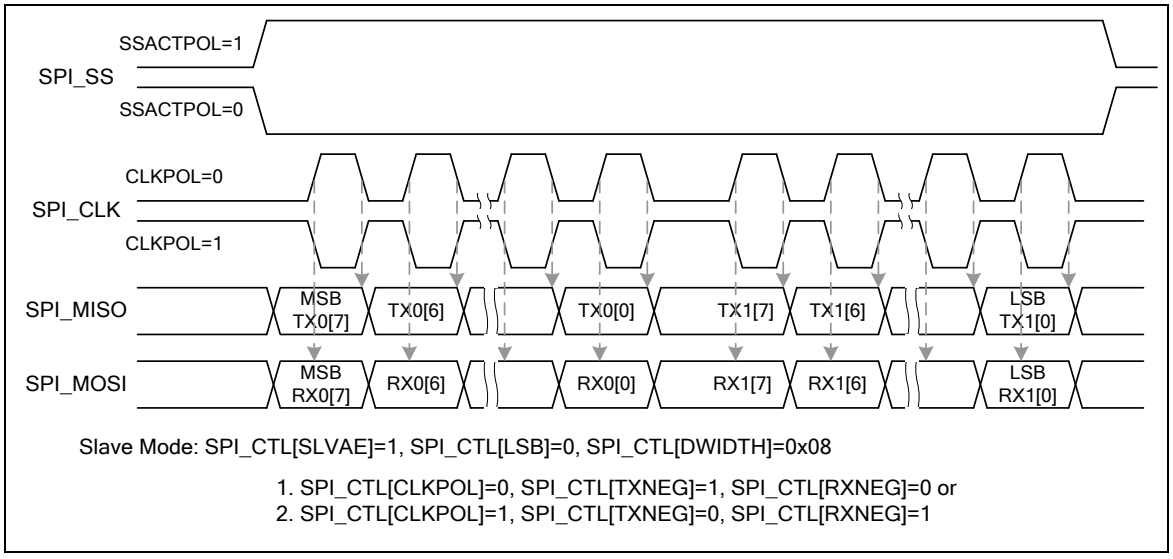


Figure 6.27-16 SPI Timing in Slave Mode

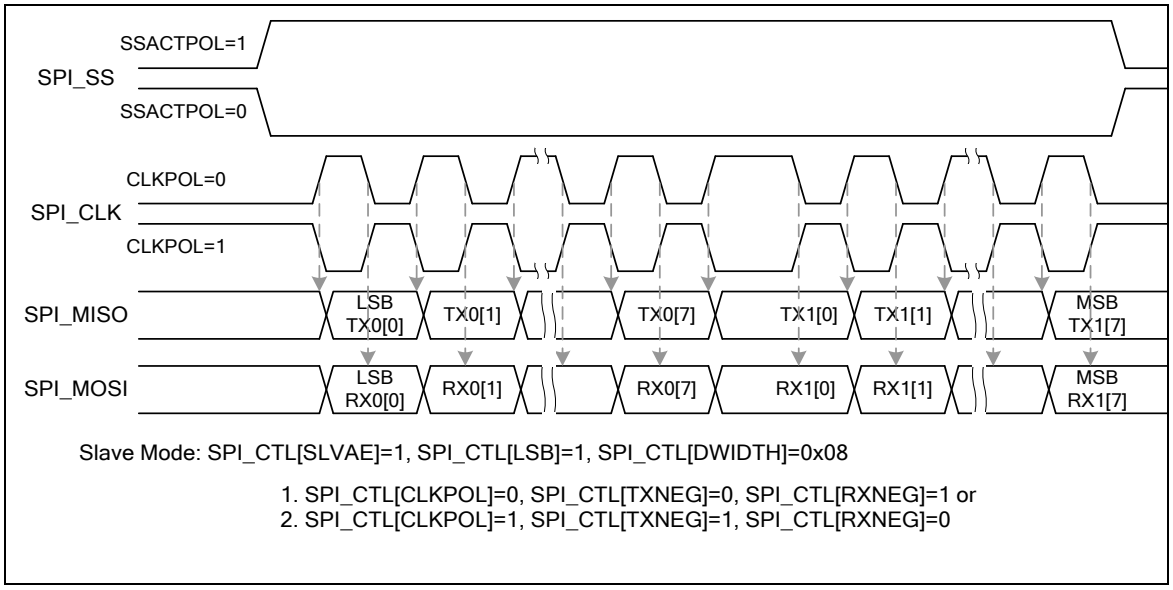


Figure 6.27-17 SPI Timing in Slave Mode (Alternate Phase of SPICLK)

6.27.6 Programming Flows

Example 1: The SPI controller is set as a master to access an off-chip slave device with the following specifications:

- Data bit is latched on positive edge of SPI clock.
- Data bit is driven on negative edge of SPI clock.
- Data is transferred from MSB first.
- SPICLK is idle at low state.
- Only one byte of data to be transmitted/received in a transaction.
- Uses the first SPI slave select pin to connect with an off-chip slave device. The slave select signal is active low.

The operation flow is as follows:

1) Set the DIVIDER (SPI_CLKDIV [7:0]) register to determine the output frequency of SPI clock.
2) Write the SPI_SSCTL register a proper value for the related settings of Master mode:
    1. Disable the Automatic Slave Select bit AUTOSS(SPI_SSCTL[3] = 0).
    2. Select low level trigger output of slave select signal in the Slave Select Active Level bit SSACTPOL (SPI_SSCTL[2] = 0).
    3. Select slave select signal to be output active at the I/O pin by setting the Slave Select Register bits SS[0] (SPI_SSCTL[0]) to active the off-chip slave device.
3) Write the related settings into the SPI_CTL register to control the SPI master actions.
    1. Set this SPI controller as master device. (SLAVE (SPI_CTL[18] = 0)).
    2. Force the SPI clock idle state at low. (CLKPOL (SPI_CTL[3] = 0)).

3. Select data transmitted at negative edge of SPI clock. (TXNEG (SPI_CTL[2] = 1)).
4. Select data latched at positive edge of SPI clock. (RXNEG (SPI_CTL[1] = 0)).
5. Set the bit length of word transfer as 8-bit. (DWIDTH (SPI_CTL[12:8] = 0x08)).
6. Set MSB transfer first. (MSB (SPI_CTL[13] = 0)).

4) Set the SPIEN (SPI_CTL [0] = 1) to start the data transfer with the SPI interface.
5) If this SPI master attempts to transmit (write) one byte data to the off-chip slave device, write the byte data that will be transmitted into the SPI_TX register.
6) Waiting for SPI interrupt (if the Interrupt Enable IE bit is set) or just polling the SPIEN bit till it is cleared to 0 by hardware automatically.
7) Read out the received one byte data from SPI_RX[7:0].
8) Go to 5) to continue another data transfer or set SS[0] to 0 to inactivate the off-chip slave device.

Example 2: The SPI controller is set as a slave device and connects with an off-chip master device. The off-chip master device communicates with the on-chip SPI slave controller through the SPI interface with the following specifications:

- Data bit is latched on positive edge of SPI clock.
- Data bit is driven on negative edge of SPI clock.
- Data is transferred from LSB first.
- SPICLK is idle at high state.
- Only one byte of data to be transmitted/received in a transaction.
- Slave select signal is high level trigger.

The operation flow is as follows:

1) Write the SPI_SSCTL register a proper value for the related settings of Slave mode.

   Select high level for the input of slave select signal by setting the Slave Select Active Level bit SSACTPOL (SPI_SSCTL[2] = 1).

2) Write the related settings into the SPI_CTL register to control this SPI slave actions
   1. Set the SPI controller as slave device. (SLAVE (SPI_CTL[18] = 1)).
   2. Select the SPI clock idle state at high. (CLKPOL (SPI_CTL[3] = 1)).
   3. Select data transmitted at negative edge of SPI clock. (TXNEG (SPI_CTL[2] = 1)).
   4. Select data latched at positive edge of SPI clock. (RXNEG (SPI_CTL[1] = 0)).
   5. Set the bit length of word transfer as 8-bit. (DWIDTH (SPI_CTL[12:8] = 0x08)).
   6. Set LSB transfer first in LSB bit (SPI_CTL[13] = 1).
3) Set the SPIEN (SPI_CTL[0] = 1) to wait for the slave select trigger input and SPI clock input from the off-chip master device to start the data transfer at the SPI interface.

4) If this SPI slave attempts to transmit (be read) one byte data to the off-chip master device, write the byte data that will be transmitted into the SPI_TX register.

5) If this SPI slave just only attempts to receive (be written) one byte data from the off-chip master device and does not care what data will be transmitted, the SPI_TX register does not need to be updated by software.

6) Waiting for SPI interrupt (if UNITIEN (SPI_CTL[17]) = 1.

7) Read out the received one byte data from SPI_RX[7:0].

8) Go to 4) to continue another data transfer or stop data transfer

6.27.7 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
SPI Base Address: **SPIx_BA = 0x4006_0000 + 0x1000 * x** **x=0,1..3**				
SPI_CTL	SPIx_BA+0x00	R/W	SPI Control Register	0x0000_0034
SPI_CLKDIV	SPIx_BA+0x04	R/W	SPI Clock Divider Register	0x0000_0000
SPI_SSCTL	SPIx_BA+0x08	R/W	SPI Slave Select Control Register	0x0000_0000
SPI_PDMACTL	SPIx_BA+0x0C	R/W	SPI PDMA Control Register	0x0000_0000
SPI_FIFOCTL	SPIx_BA+0x10	R/W	SPI FIFO Control Register	0x4400_0000
SPI_STATUS	SPIx_BA+0x14	R/W	SPI Status Register	0x0005_0110
SPI_TX	SPIx_BA+0x20	W	SPI Data Transmit Register	0x0000_0000
SPI_RX	SPIx_BA+0x30	R	SPI Data Receive Register	0x0000_0000

Note: x = 0, 1,2,3

6.27.8 Register Description

SPI Control and Status Register (SPI_CTL)

Register	Offset	R/W	Description	Reset Value
SPI_CTL	SPIx_BA+0x00	R/W	SPI Control Register	0x0000_0034

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved	QUADIOEN	DUALIOEN	QDIODIR	REORDER	SLAVE	UNITIEN	TWOBIT
15	14	13	12	11	10	9	8
Reserved		LSB	DWIDTH				
7	6	5	4	3	2	1	0
SUSPITV				CLKPOL	TXNEG	RXNEG	SPIEN

Bits	Description	
[31:23]	**Reserved**	Reserved.
[22]	**QUADIOEN**	**Quad I/O Mode Enable Bit** 0 = Quad I/O mode Disabled. 1 = Quad I/O mode Enabled.
[21]	**DUALIOEN**	**Dual I/O Mode Enable Bit** 0 = Dual I/O mode Disabled. 1 = Dual I/O mode Enabled.
[20]	**QDIODIR**	**Quad Or Dual I/O Mode Direction Control** 0 = Quad or Dual Input mode. 1 = Quad or Dual Output mode.
[19]	**REORDER**	**Byte Reorder Function Enable Bit** 0 = Byte reorder function Disabled. 1 = Byte reorder function Enabled. **Note:** 1. Byte reorder function is only available if DWIDTH is defined as 16, 24, and 32 bits. 2. The byte reorder function is not supported when the Quad or Dual I/O mode is enabled. 3. A byte suspend interval will be inserted among each byte. The period of the byte suspend interval depends on the setting of SUSPITV.
[18]	**SLAVE**	**Slave Mode Enable Bit** 0 = Master mode. 1 = Slave mode.
[17]	**UNITIEN**	**Unit Transfer Interrupt Enable Bit**

		0 = SPI unit transfer interrupt Disabled. 1 = SPI unit transfer interrupt Enabled.
[16]	**TWOBIT**	**2-Bit Mode Enable Bit** 0 = 2-bit mode Disabled. 1 = 2-bit mode Enabled. **Note:** When 2-bit mode is enabled, the first serial transmitted bit data is from the first FIFO buffer data, and the 2nd serial transmitted bit data is from the second FIFO buffer data. As the same as transmitted function, the first received bit data is stored into the first FIFO buffer and the 2nd received bit data is stored into the second FIFO buffer at the same time.
[15:14]	**Reserved**	Reserved.
[13]	**LSB**	**Send LSB First** 0 = MSB first. 1 = LSB first. **Note:** 1. The LSB, bit 0 of the SPI TX register, is sent first to the SPI data output pin, and the first bit received from the SPI data input pin will be put in the LSB position of the RX register (bit 0 of SPI_RX). 2. The MSB, which bit of transmit/receive register depends on the setting of DWIDTH, is transmitted/received first.
[12:8]	**DWIDTH**	**Data Transmit Bit Width** This field specifies how many bits can be transmitted / received in one transaction. The minimum bit length is 8 bits and can up to 32 bits. DWIDTH = 0x08 … 8 bits. DWIDTH = 0x09 … 9 bits. …… DWIDTH = 0x1F … 31 bits. DWIDTH = 0x00 … 32 bits.
[7:4]	**SUSPITV**	**Suspend Interval (Master Only)** SUSPITV = 0x0 … 0.5 SPICLK clock cycle. SUSPITV = 0x1 … 1.5 SPICLK clock cycle. …… SUSPITV = 0xE … 14.5 SPICLK clock cycle. SUSPITV = 0xF … 15.5 SPICLK clock cycle. **Note:** The four bits provide configurable suspend interval between two successive transmit/receive transaction in a transfer. The definition of the suspend interval is the interval between the last clock edge of the preceding transaction word and the first clock edge of the following transaction word. The default value is 0x3. The period of the suspend interval is obtained according to the following equation. (SUSPITV[3:0] + 0.5) * period of SPICLK clock cycle
[3]	**CLKPOL**	**Clock Polarity** 0 = SPICLK is idle low. 1 = SPICLK is idle high.
[2]	**TXNEG**	**Transmit On Negative Edge** 0 = Transmitted data output signal is changed on the rising edge of SPICLK. 1 = Transmitted data output signal is changed on the falling edge of SPICLK.
[1]	**RXNEG**	**Receive On Negative Edge** 0 = Received data input signal is latched on the rising edge of SPICLK. 1 = Received data input signal is latched on the falling edge of SPICLK.

[0]	**SPIEN**	**SPI Transfer Control Enable Bit** 0 = Transfer control Disabled. 1 = Transfer control Enabled. **Note:** 1. In Master mode, the transfer will start when there is data in the FIFO buffer after this is set to 1. In Slave mode, this device is ready to receive data when this bit is set to 1. 2. All configurations should be set before writing 1 to this SPIEN bit. (eg: TXNEG, RXNEG, DWIDTH, LSB, CLKPOL, and so on).

SPI Divider Register (SPI_CLKDIV)

Register	Offset	R/W	Description	Reset Value
SPI_CLKDIV	SPIx_BA+0x04	R/W	SPI Clock Divider Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
DIVIDER[7:0]							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**DIVIDER**	**Clock Divider Register** The value in this field is the frequency divider for generating the peripheral clock, f_{spi_eclk}, and the SPI bus clock of SPI master. The frequency is obtained according to the following equation. $f_{spi\_peripheral} = \frac{f_{spi\_clock\_src}}{(DIVIDER+1)}$ Note : 1. $f_{spi\_clock\_src}$ is the peripheral clock source, which is defined in the clock control, CLK_SEL1 register. 3. $f_{spi\_peripheral}$ is the peripheral clock which is used to drive the SPI logic unit.

SPI Slave Select Register (SPI_SSCTL)

Register	Offset	R/W	Description	Reset Value
SPI_SSCTL	SPIx_BA+0x08	R/W	SPI Slave Select Control Register	0x0000_0000

31	30	29	28	27	26	25	24
SLVTOCNT[15:8]							
23	22	21	20	19	18	17	16
SLVTOCNT[7:0]							
15	14	13	12	11	10	9	8
Reserved		SSINAIEN	SSACTIEN	Reserved		SLVURIEN	SLVBEIEN
7	6	5	4	3	2	1	0
Reserved	SLVTORST	SLVTOIEN	SLV3WIRE	AUTOSS	SSACTPOL	SS	

Bits	Description	
[31:16]	**SLVTOCNT**	**Slave Mode Time-Out Period** 0 = Slave time out function disabled. Others = Slave time out period. **Note:** In Slave mode, these bits indicate the time-out period when there is bus clock input during slave select active. The clock source of the time-out counter is Slave peripheral clock. If the value is 0, it indicates the slave mode time-out function is disabled.
[15:14]	**Reserved**	Reserved.
[13]	**SSINAIEN**	**Slave Select Inactive Interrupt Enable Bit** 0 = Slave select inactive interrupt Disabled. 1 = Slave select inactive interrupt Enabled.
[12]	**SSACTIEN**	**Slave Select Active Interrupt Enable Bit** 0 = Slave select active interrupt Disabled. 1 = Slave select active interrupt Enabled.
[11:10]	**Reserved**	Reserved.
[9]	**SLVURIEN**	**Slave Mode Error 1 Interrupt Enable Bit** 0 = Slave mode error 1 interrupt Disabled. 1 = Slave mode error 1 interrupt Enabled.
[8]	**SLVBEIEN**	**Slave Mode Error 0 Interrupt Enable Bit** 0 = Slave mode error 0 interrupt Disabled. 1 = Slave mode error 0 interrupt Enabled.
[7]	**Reserved**	Reserved.
[6]	**SLVTORST**	**Slave Mode Time-Out FIFO Clear** 0 = Time out FIFO clear Disabled. 1 = Time out FIFO clear Enabled. **Note:** Both the FIFO clear function, TXRST and RXRST, active automatically when there is

		slave mode time-out event.
[5]	**SLVTOIEN**	**Slave Mode Time-Out Interrupt Enable Bit** 0 = Slave mode time-out interrupt Disabled. 1 = Slave mode time-out interrupt Enabled.
[4]	**SLV3WIRE**	**Slave 3-Wire Mode Enable Bit** 0 = 4-wire bi-direction interface. 1 = 3-wire bi-direction interface. **Note:** This is used to ignore the slave select signal in Slave mode. The SPI controller can work with 3-wire interface including SPI_CLK, SPI_MISO, and SPI_MOSI.
[3]	**AUTOSS**	**Automatic Slave Select Function Enable Bit (Master Only)** 0 = Automatic slave select function Disabled. 1 = Automatic slave select function Enabled. **Note1:** If this bit is cleared, slave select signals will be asserted/de-asserted by setting /clearing the corresponding bits of SPI_SSCTL[1:0]. **Note2:** If this bit is set, SPI_SS0/1 signals will be generated automatically. It means that device/slave select signal, which is set in SPI_SSCTL[1:0], will be asserted by the SPI controller when transmit/receive is started, and will be de-asserted after each transmit/receive is finished.
[2]	**SSACTPOL**	**Slave Select Active Level** 0 = The slave select signal SPI_SS0/1 is active on low-level. 1 = The slave select signal SPI_SS0/1 is active on high-level. **Note:** This bit defines the active status of slave select signal (SPI_SS0/1).
[1:0]	**SS**	**Slave Select Control (Master Only)** If AUTOSS bit is cleared to 0, 0 = Set the SPI_SS line to inactive state. 1 = Set the proper SPI_SS line to active state. If AUTOSS bit is set to 1, 0 = Keep the SPI_SS line at inactive state. 1 = Select the SPI_SS line to be automatically driven to active state for the duration of transmission/reception, and will be driven to inactive state for the rest of the time. The active state of SPI_SS is specified in SSACTPOL bit. **Note:** SPI_SS is defined as the slave select input in Slave mode.

SPI PDMA Control Register (SPI_PDMACTL)

Register	Offset	R/W	Description	Reset Value
SPI_PDMACTL	SPIx_BA+0x0C	R/W	SPI PDMA Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					PDMARST	RXPDMAEN	TXPDMAEN

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**PDMARST**	**PDMA Reset** 0 = No effect. 1 = Reset the PDMA control logic of the SPI controller. This bit will be cleared to 0 automatically.
[1]	**RXPDMAEN**	**Receive PDMA Enable Bit** 0 = Receive PDMA Disabled. 1 = Receive PDMA Enabled. **Note:** Setting this bit to 1 will start the receive PDMA process. The SPI controller will issue request to PDMA controller automatically when the SPI receive buffer is not empty. This bit will be cleared to 0 by hardware automatically after PDMA transfer is done.
[0]	**TXPDMAEN**	**Transmit DMA Enable Bit** 0 = Transmit PDMA Disabled. 1 = Transmit PDMA Enabled. **Note:** Setting this bit to 1 will start the transmit PDMA process. SPI controller will issue request to PDMA controller automatically. Hardware will clear this bit to 0 automatically after PDMA transfer done.

SPI FIFO Control Register (SPI_FIFOCTL)

Register	Offset	R/W	Description	Reset Value
SPI_FIFOCTL	SPIx_BA+0x10	R/W	SPI FIFO Control Register	0x4400_0000

31	30	29	28	27	26	25	24
Reserved	TXTH			Reserved	RXTH		
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
TXUFIEN	TXUFPOL	RXOVIEN	RXTOIEN	TXTHIEN	RXTHIEN	TXRST	RXRST

Bits	Description	
[31]	**Reserved**	Reserved.
[30:28]	**TXTH**	**Transmit FIFO Threshold** If the valid data count of the transmit FIFO buffer is less than or equal to the TXTH setting, the TXTHIF bit will be set to 1, else the TXTHIF bit will be cleared to 0.
[27]	**Reserved**	Reserved.
[26:24]	**RXTH**	**Receive FIFO Threshold** If the valid data count of the receive FIFO buffer is larger than the RXTH setting, the RXTHIF bit will be set to 1, else the RXTHIF bit will be cleared to 0.
[23:8]	**Reserved**	Reserved.
[7]	**TXUFIEN**	**Transmit Under Run Interrupt Enable Bit (Slave Only)** 0 = Transmit FIFO under-run interrupt Disabled. 1 = Transmit FIFO under-run interrupt Enabled.
[6]	**TXUFPOL**	**Transmit Under-Run Data Out (Slave Only)** 0 = The SPI data bus is keep low if there is transmit under-run event. 1 = The SPI data bus is keep high if there is transmit under-run event. **Note1:** The under run event is actived after the bus clock input and the hardware synchronous, so that the first 1~3 bit (depending on the relation between system clock and the peripheral clock) data out will be the last transaction data. **Note2:** If the frequency of system clock approach to peripheral clock, they may need 3-bit time to report the transmit under-run event.
[5]	**RXOVIEN**	**Receive FIFO Overrun Interrupt Enable Bit** 0 = Receive FIFO overrun interrupt Disabled. 1 = Receive FIFO overrun interrupt Enabled.
[4]	**RXTOIEN**	**Slave Receive Time-Out Interrupt Enable Bit (Slave Only)** 0 = Receive time-out interrupt Disabled. 1 = Receive time-out interrupt Enabled.

[3]	**TXTHIEN**	**Transmit FIFO Threshold Interrupt Enable Bit** 0 = TX FIFO threshold interrupt Disabled. 1 = TX FIFO threshold interrupt Enabled.
[2]	**RXTHIEN**	**Receive FIFO Threshold Interrupt Enable Bit** 0 = RX FIFO threshold interrupt Disabled. 1 = RX FIFO threshold interrupt Enabled.
[1]	**TXRST**	**Clear Transmit FIFO Buffer** 0 = No effect. 1 = Clear transmit FIFO buffer. **Note1:** If there is slave receive time-out event, the TXRST will be set 1 when the SLVTORST, SPI_SSCTL[6], is enabled. **Note2:** The TXFULL bit will be cleared to 0 and the TXEMPTY bit will be set to 1. This bit will be cleared to 0 by hardware about 3 system clocks + 3 peripheral clock after it is set to 1.
[0]	**RXRST**	**Clear Receive FIFO Buffer** 0 = No effect. 1 = Clear receive FIFO buffer. **Note1:** If there is slave receive time-out event, the RXRST will be set 1 when the SLVTORST, SPI_SSCTL[6], is enabled. **Note2:** The RXFULL bit will be cleared to 0 and the RXEMPTY bit will be set to 1. This bit will be cleared to 0 by hardware about 3 system clocks + 3 peripheral clock after it is set to 1.

SPI Status Register (SPI_STATUS)

Register	Offset	R/W	Description	Reset Value
SPI_STATUS	SPIx_BA+0x14	R/W	SPI Status Register	0x0005_0110

31	30	29	28	27	26	25	24
TXCNT				RXCNT			
23	22	21	20	19	18	17	16
TXRXRST	Reserved			TXUFIF	TXTHIF	TXFULL	TXEMPTY
15	14	13	12	11	10	9	8
SPIENSTS	Reserved		RXTOIF	RXOVIF	RXTHIF	RXFULL	RXEMPTY
7	6	5	4	3	2	1	0
SLVURIF	SLVBEIF	SLVTOIF	SSLINE	SSINAIF	SSACTIF	UNITIF	BUSY

Bits	Description	
[31:28]	**TXCNT**	**Transmit FIFO Data Count (Read Only)** This bit field indicates the valid data count of transmit FIFO buffer.
[27:24]	**RXCNT**	**Receive FIFO Data Count (Read Only)** This bit field indicates the valid data count of receive FIFO buffer.
[23]	**TXRXRST**	**FIFO CLR Status (Read Only)** 0 = Done the FIFO buffer clear function of TXRST or RXRST. 1 = Doing the FIFO buffer clear function of TXRST or RXRST. **Note:** Both the TXRST, RXRST, need 3 system clock + 3 peripheral clock , the status of this bit support the user to monitor the clear function is doing or done.
[22:20]	**Reserved**	Reserved.
[19]	**TXUFIF**	**Transmit FIFO Under-Run Interrupt Status** 0 =No under-run interrupt event. 1 = Under-run interrupt occurred. **Note:** When the transmit FIFO buffer is empty and there is no datum written into the FIFO buffer, if there is more bus clock input , the output data depends on the setting of TXUFPOL and this bit will be set to 1 and this bit will be cleared by writing 1 to it.
[18]	**TXTHIF**	**Transmit FIFO Threshold Interrupt Status (Read Only)** 0 = The valid data count within the transmit FIFO buffer is larger than the setting value of TXTH. 1 = The valid data count within the transmit FIFO buffer is less than or equal to the setting value of TXTH. **Note:** If TXTHIEN = 1 and TXTHIF = 1, the SPI controller will generate a SPI interrupt request.
[17]	**TXFULL**	**Transmit FIFO Buffer Full Indicator (Read Only)** 0 = Transmit FIFO buffer is not full. 1 = Transmit FIFO buffer is full.
[16]	**TXEMPTY**	**Transmit FIFO Buffer Empty Indicator (Read Only)**

		0 = Transmit FIFO buffer is not empty. 1 = Transmit FIFO buffer is empty.
[15]	**SPIENSTS**	**SPI Enable Bit Status (Read Only)** 0 = Indicates the transmit control bit is disabled. 1 = Indicates the transfer control bit is active. **Note:** The clock source of SPI controller logic is peripheral clock, it is asynchronous with the system clock. In order to make sure the function is disabled in SPI controller logic, this bit indicates the real status of SPIEN in SPI controller logic for user.
[14:13]	**Reserved**	Reserved.
[12]	**RXTOIF**	**Receive Time-Out Interrupt Status** 0 = No receive FIFO time-out event. 1 = FIFO time-out event occurred. **Note:** Receive FIFO buffer is not empty and no read operation on receive FIFO buffer over 64 SPI clock period in Master mode or over 576 peripheral clock period in Slave mode. When the received FIFO buffer is read by software, the time-out status will be cleared automatically. This bit will be cleared by writing 1 to it.
[11]	**RXOVIF**	**Receive FIFO Overrun Status** 0 = No FIFO over-run event. 1 = FIFO over-run event occurred. **Note:** When the receive FIFO buffer is full, the follow-up data will be dropped and this bit will be set to 1. This bit will be cleared by writing 1 to it.
[10]	**RXTHIF**	**Receive FIFO Threshold Interrupt Status (Read Only)** 0 = The valid data count within the Rx FIFO buffer is smaller than or equal to the setting value of RXTH. 1 = The valid data count within the receive FIFO buffer is larger than the setting value of RXTH. **Note:** If RX_INTEN = 1 and RX_INTSTS = 1, the SPI controller will generate a SPI interrupt request.
[9]	**RXFULL**	**Receive FIFO Buffer Empty Indicator (Read Only)** 0 = Receive FIFO buffer is not empty. 1 = Receive FIFO buffer is empty.
[8]	**RXEMPTY**	**Receive FIFO Buffer Empty Indicator (Read Only)** 0 = Receive FIFO buffer is not empty. 1 = Receive FIFO buffer is empty.
[7]	**SLVURIF**	**Slave Mode Error 1 Interrupt Status** 0 = No Slave mode error 1 event. 1 = Slave mode error 1 occurs. **Note:** In Slave mode, transmit under-run occurs when the slave select line goes to inactive state. This bit will be cleared by writing 1 to it.
[6]	**SLVBEIF**	**Slave Mode Error 0 Interrupt Status** 0 = No Slave mode error 0 event. 1 = Slave mode error 0 occurs. **Note1:** If the slave select active but there is no any bus clock input, the SLVBEIF also active when the slave select goes to inactive state. This bit will be cleared by writing 1 to it. **Note2:** In Slave mode, there is bit counter mismatch with DWIDTH when the slave select line goes to inactive state.
[5]	**SLVTOIF**	**Slave Time-Out Interrupt Status** 0 = Slave time-out is not active.

		1 = Slave time-out is active. **Note1:** If the DWIDTH is set 16, one transaction is equal 16 bits bus clock period. This bit will be cleared by writing 1 to it. **Note2:** When the Slave Select is active and the value of SLVTOCNT is not 0 and the bus clock input, the slave time-out counter in SPI controller logic will be start. When the value of time-out counter greater or equal than the value of SLVTOCNT, SPI_SSCTL[31:16], during before one transaction done, the slave time-out interrupt event will active.
[4]	**SSLINE**	**Slave Select Line Bus Status (Read Only)** 0 = Indicates the slave select line bus status is 0. 1 = Indicates the slave select line bus status is 1. **Note:** If SSACTPOL, SPI_SSCTL[2], is set 0, and the SSLINE is 1, the SPI slave select is in inactive status.
[3]	**SSINAIF**	**Slave Select Inactive Interrupt Status** 0 = Slave select inactive interrupt is clear or not occur. 1 = Slave select inactive interrupt event occurred. **Note:** This bit will be cleared by writing 1 to it.
[2]	**SSACTIF**	**Slave Select Active Interrupt Status** 0 = Slave select active interrupt is clear or not occur. 1 = Slave select active interrupt event occurred. **Note:** This bit will be cleared by writing 1 to it.
[1]	**UNITIF**	**Unit Transfer Interrupt Status** 0 = No transaction has been finished since this bit was cleared to 0. 1 = SPI controller has finished one unit transfer. **Note:** This bit will be cleared by writing 1 to it.
[0]	**BUSY**	**Busy Status (Read Only)** 0 = SPI controller is in Idle state. 1 = SPI controller is in busy state. The following listing are the bus busy conditions: ● SPIEN = 1 and the TXEMPTY = 0. ● For SPI Master, the TXEMPTY = 1 but the current transaction is not finished yet. ● For SPI Slave receive mode, the SPIEN = 1 and there is serial clock input into the SPI core logic when slave select is active. ● For SPI Slave transmit mode, the SPIEN = 1 and the transmit buffer is not empty in SPI core logic even if the slave select is inactive.

SPI Data Transmit Register (SPI_TX)

Register	Offset	R/W	Description	Reset Value
SPI_TX	SPIx_BA+0x20	W	SPI Data Transmit Register	0x0000_0000

31	30	29	28	27	26	25	24
TX[31:24]							
23	22	21	20	19	18	17	16
TX[23:16]							
15	14	13	12	11	10	9	8
TX[15:8]							
7	6	5	4	3	2	1	0
TX[7:0]							

Bits	Description	
[31:0]	TX	**Data Transmit Bits** The data transmit registers pass through the transmitted data into the 8-level transmit FIFO buffer. The number of valid bits depends on the setting of transmit bit width field of the SPI_CTL register. In Master mode, the serial data in SPI bus output need 5 module clock cycle when the data transmit registers pass through the transmitted data into the 8-level transmit FIFO buffer. For example, if DWIDTH is set to 0x08, the bits TX[7:0] will be transmitted. If DWIDTH is set to 0x00, the SPI controller will perform a 32-bit transfer.

SPI Data Receive Register (SPI_RX)

Register	Offset	R/W	Description	Reset Value
SPI_RX	SPIx_BA+0x30	R	SPI Data Receive Register	0x0000_0000

31	30	29	28	27	26	25	24
RX[31:24]							
23	22	21	20	19	18	17	16
RX[23:16]							
15	14	13	12	11	10	9	8
RX[15:8]							
7	6	5	4	3	2	1	0
RX[7:0]							

Bits	Description	
[31:0]	RX	**Data Receive Bits** There is 8-level FIFO buffer in this controller. The data receive register holds the earliest datum received from SPI data input pin. If the RXEMPTY bit, SPI_STATUS[8], is not set to 1, the receive FIFO buffer can be accessed through software by reading this register. This is a read only register.

6.28 Timer Controller (TIMER)

6.28.1 Overview

The Timer Controller includes four 32-bit timers, TIMER0 ~ TIMER3, allowing user to easily implement a timer control for applications. The timer can perform functions, such as frequency measurement, delay timing, clock generation, and event counting by external input pins, and interval measurement by external capture pins.

6.28.2 Features

- Four sets of 32-bit timers with 24-bit up counter and one 8-bit prescale counter
- Independent clock source for each timer
- Provides one-shot, periodic, toggle and continuous counting operation modes
- Time-out period = (Period of timer clock input) * (8-bit prescale counter + 1) * CMPDAT (TIMERx_CMP[23:0])
- Maximum counting cycle time = (1 / T MHz) * (2^8) * (2^{24}), T is the period of timer clock
- 24-bit up counter value is readable through TIMERx_CNT (Timer Data Register)
- Supports event counting function to count the event from external pin (TM0~TM3)
- Supports external capture pin (TM0_EXT~TM3_EXT) for interval measurement
- Supports external capture pin (TM0_EXT~TM3_EXT) to reset 24-bit up counter
- Supports chip wake-up from Idle/Power-down mode if a timer interrupt signal is generated

6.28.3 Block Diagram

The Timer Controller block diagram and clock control are shown as follows.

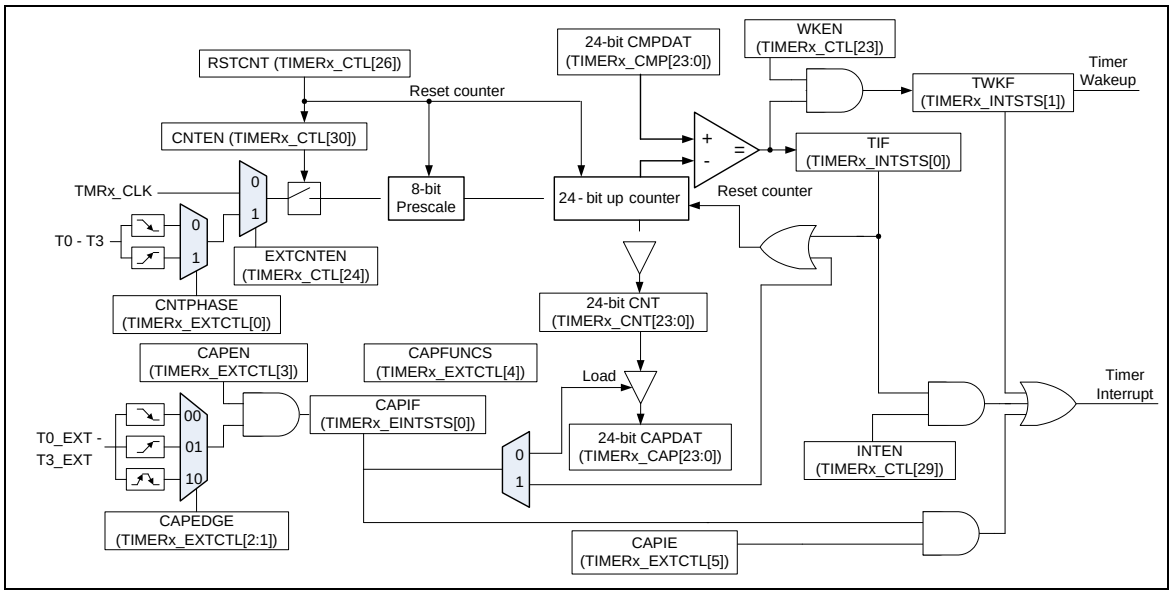


Figure 6.28-1 Timer Controller Block Diagram

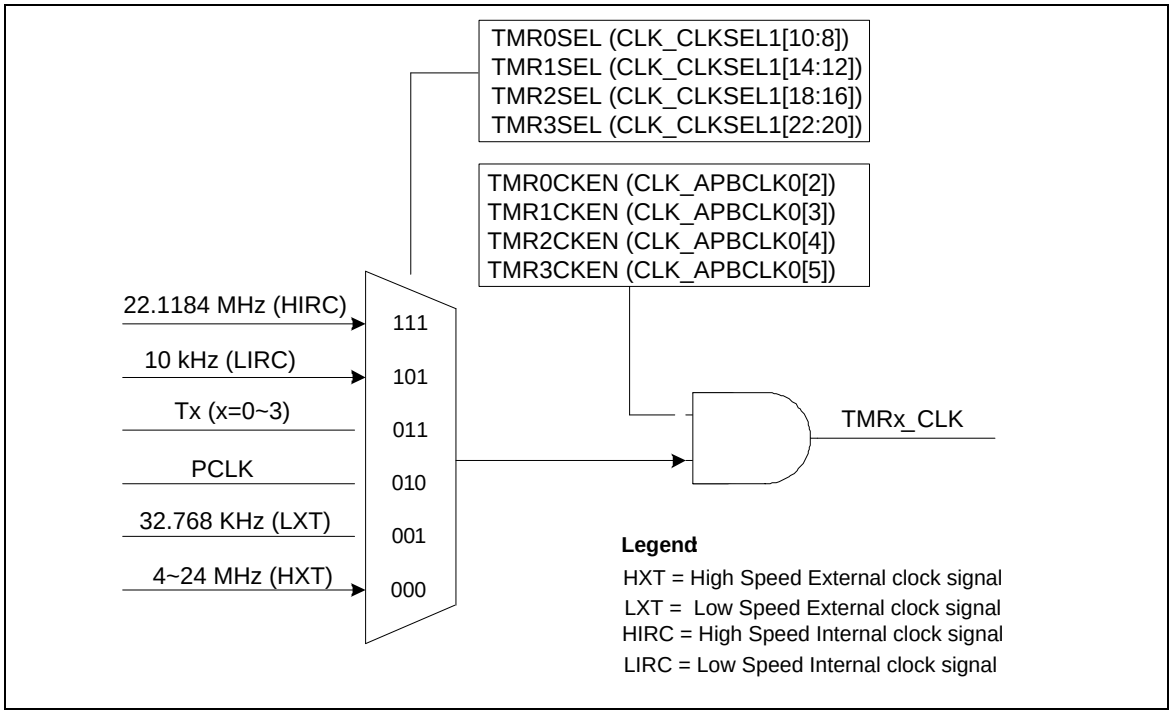


Figure 6.28-2 Clock Source of Timer Controller

6.28.4 Basic Configuration

The peripheral clock source of Tiimer0 ~ Timer3 can be enabled in CLK_APBCLK0[5:2] and selected as different frequency in CLK_CLKSEL1[10:8] for Timer0, CLK_CLKSEL1[14:12] for Timer1, CLK_CLKSEL1[18:16] for Timer2 and CLK_CLKSEL1[22:20] for Timer3.

6.28.5 Functional Description

Timer controller provides One-shot, Period, Toggle and Continuous Counting operation modes. The event counting function is also provided to count the events/counts from external pin and external pin capture function for interval measurement or reset timer counter. Each operating function mode is shown as follows.

6.28.5.1 Timer Interrupt Flag

Timer controller supports two interrupt flags; one is TIF flag and its set while timer counter value (TIMERx_CNT) matches the timer compared value (TIMERx_CMP), the other is CAPIF(TIMERxEINTSTS[0]) flag and its set when the transition on the TMx_EXT pin associated CAPEDGE(TIMERx_EXTCTL[2:1]) setting.

6.28.5.2 One–shot Mode

If timer controller is configured at one-shot mode (TIMERx_CTL[28:27] is 00) and CNTEN (TIMERx_CTL[30]) bit is set, the timer counter starts up counting. Once the TIMERx_CNT value reaches TIMERx_CMP value, the TIF flag will be set to 1, TIMERx_CNT value and CNTEN bit is cleared by timer controller then timer counting operation stops. In the meantime, if the INTEN (TIMERx_CTL[29]) bit is enabled, the timer interrupt signal is generated and sent to NVIC to inform CPU also.

In this operating mode, once the TIMERx_CNT value reaches TIMERx_CMP value, TIF (TIMERx_INTSTS[0]) will set to 1, timer counting operation stops and the timer counter value (TIMERx_CNT value) goes back to counting initial value then CNTEN (TIMERx_CTL[30]) is cleared to 0 by timer controller automatically. That is to say, timer operates timer counting and compares with TIMERx_CMP value function only one time after programming the timer compare register (TIMERx_CMP) value and CNTEN (TIMERx_CTL[30]) is set to 1. Accordingly, this operating mode is called “one-shot mode”.

6.28.5.3 Periodic Mode

If the timer is operated in period mode (TIMERx_CTL[28:27] is 01) and CNTEN (TIMERx_CTL[30]) is set to 1, the timer counter starts up counting. Once the TIMERx_CNT value reaches CMPDAT value, the TIF (TIMERx_INTSTS[0]) will set to 1. If INTEN (TIMERx_CTL[29]) is set to 1, and TIF (TIMERx_INTSTS[0]) is 1 then the interrupt signal is generated and sent to NVIC to inform CPU for indicating that the timer counting overflow happens. If INTEN (TIMERx_CTL[29]) is set to 0, no interrupt signal is generated.

In this operating mode, once theTIMERx_CNT value reaches TIMERx_CMP value, TIF (TIMERx_INTSTS[0]) will set to 1, the timer counter value (TIMERx_CNT value) goes back to counting initial value and CNTEN (TIMERx_CTL[30] timer enable bit) is kept at 1 (counting enable continuously) and timer counter operates up counting again. If TIF (TIMERx_INTSTS[0]) is cleared by software, once the TIMERx_CNT value reaches TIMERx_CMP value again, TIF (TIMERx_INTSTS[0]) will set to 1 also. That is to say, timer operates timer counting and compares with TIMERx_CMP value function periodically. The timer counting operation does not

stop until the CNTEN (TIMERx_CTL[30]) is set to 0. The interrupt signal is also generated periodically. So, this operating mode is called “periodic mode”.

6.28.5.4 Toggle-Output Mode

If the timer is operated in toggle-out mode (TIMERx_CTL[28:27] is 10) and CNTEN (TIMERx_CTL[30]) is set to 1, the timer counter starts up counting. Once the TIMERx_CNT value reaches TIMERx_CMP value, the TIF (TIMERx_INTSTS[0]) will set to 1. If INTEN (TIMERx_CTL[29]) is set to 1, and TIF (TIMERx_INTSTS[0]) is 1 then the interrupt signal is generated and sent to NVIC to inform CPU for indicating that the timer counting overflow happens. If INTEN (TIMERx_CTL[29]) is set to 0, no interrupt signal is generated.

In this operating mode, once the timer TIMERx_CNT value reaches TIMERx_CMP value, TIF (TIMERx_INTSTS[0]) will set to 1, toggle-out signal (TM0~TM3 pin) is set to 1, the TIMERx_CNT value goes back to counting initial value and CNTEN (TIMERx_CTL[30]) is kept at 1 and timer counter operates up counting again. If TIF (TIMERx_INTSTS[0]) is cleared by software, once the TIMERx_CNT value reaches TIMERx_CMP value again, TIF (TIMERx_INTSTS[0]) will set to 1 also and toggle-out signal (TM0~TM3 pin) is set to 0. The timer counting operation does not stop until the CNTEN (TIMERx_CTL[30]) is set to 0. Thus, the toggle-out signal (TM0~TM3 pin) is changing back and forth with 50% duty cycle. So, this operating mode is called toggle-out mode.

6.28.5.5 Continuous Counting Mode

If the timer is operated in continuous counting mode (TIMERx_CTL[28:27] is 11) and CNTEN (TIMERx_CTL[30]) is set to 1, the timer counter starts up counting. Once the TIMERx_CNT value reaches TIMERx_CMP value, the TIF (TIMERx_INTSTS[0]) will set to 1. If INTEN (TIMERx_CTL[29]) is set to 1, and TIF (TIMERx_INTSTS[0]) is 1 then the interrupt signal is generated and sent to NVIC to inform CPU for indicating that the timer counting overflow happens. If INTEN (TIMERx_CTL[29]) is set to 0, no interrupt signal is generated.

In this operating mode, once the TIMERx_CNT value reaches TIMERx_CMP value, TIF (TIMERx_INTSTS[0]) will set to 1 and CNTEN (TIMERx_CTL[30]) is kept at 1 and timer counter continuous counting without reload the timer counter value (TIMERx_CNT value) to counting initial value. User can change different timer compare register (TIMERx_CMP) value immediately without disabling timer counter and restarting timer counter counting.

For example, the TIMERx_CMP value is set as 80, first. Once the TIMERx_CNT value reaches to 80, TIF will set to 1 and CNTEN (TIMERx_CTL[30]) is kept at 1 and TIMERx_CNT value will not goes back to 0, it continues to count 81, 82, 83,⋯ to 2^{24} -1, 0, 1, 2, 3, ⋯ to 2^{24} -1 again and again. Next, if user programs TIMERx_CMP value as 200 and the TIF (TIMERx_INTSTS[0]) is cleared to 0, then TIF (TIMERx_INTSTS[0]) will set to 1 again when TIMERx_CNT value reaches to 200. At last, user programs TIMERx_CMP value as 500 and clears TIF (TIMERx_INTSTS[0]) to 0, then TIF (TIMERx_INTSTS[0]) will set to 1 again when TIMERx_CNT value reaches to 500. In this mode, the TIMERx_CNT value is keeping up counting always even if TIF (TIMERx_INTSTS[0]) is 1. So, this operation mode is called “continuous counting mode”.

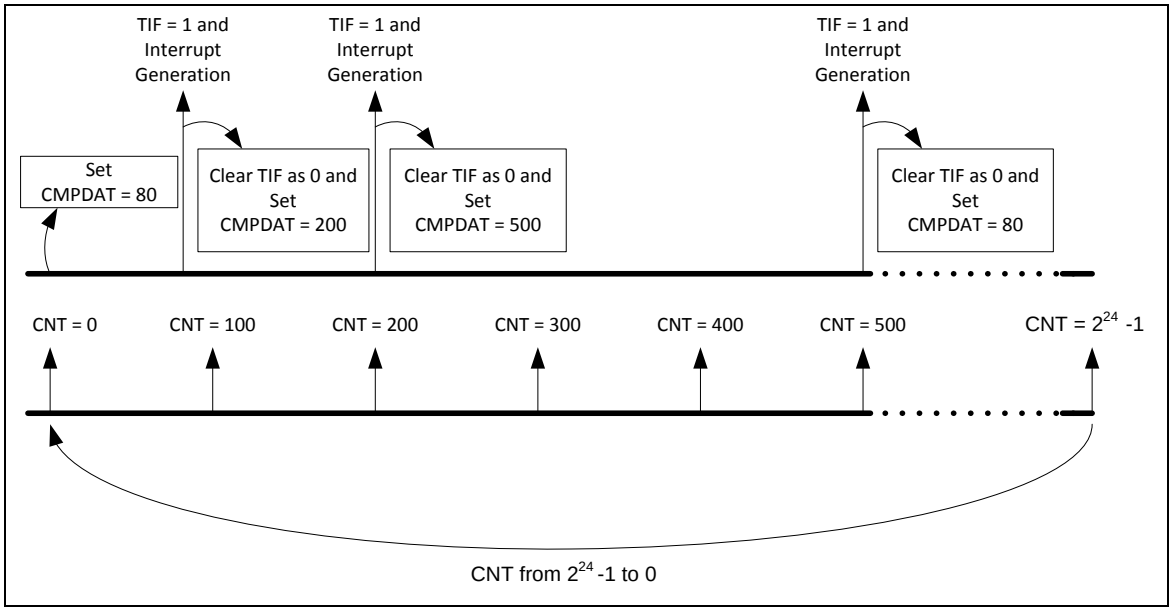


Figure 6.28-3 Continuous Counting Mode

6.28.5.6 Event Counting Mode

Timer controller also provides an application which can count the input event from TMx (x= 0~3) pin and the number of event will reflect to TIMERx_CNT value. It is also called as event counting function. In this function, EXTCNTEN (TIMERx_CTL[24]) bit should be set and the timer peripheral clock source should be set as PCLK.

Software can enable or disable TMx pin de-bounce circuit by CNTDBEN (TIMERx_EXTCTL[7]) bit. The input event frequency should be less than 1/3 PCLK if TMx (x= 0~3) pin de-bounce disabled or less than 1/8 PCLK if TMx (x= 0~3) pin de-bounce enabled to assure the returned TIMERx_CNT value is incorrect, and software can also select edge detection phase of TMx pin by CNTPHASE (TIMERx_EXTCTL[0]) bit.

In event counting mode, the timer counting operation mode can be selected as one-shot, periodic and continuous counting mode to counts the TIMERx_CNT value by input event from TMx (x= 0~3) pin.

6.28.5.7 Event Capture Mode

The event capture function is used to capture TIMERx_CNT value to TIMERx_CAP value while edge transition detected on TMx_EXT (x= 0~3) pin. In this mode, CAPFUNCS (TIMERx_EXTCTL[4]) bit should be as 0 for select TMx_EXT transition is using as the event capture function and the timer peripheral clock source should be set as PCLK.

Software can enable or disable TMx_EXT pin de-bounce circuit by CAPDBEN (TIMERx_EXTCTL[6]) bit. The transition frequency of TMx_EXT pin should be less than 1/3 PCLK if TMx_EXT pin de-bounce disabled or less than 1/8 PCLK if TMx_EXT pin de-bounce enabled to assure the capture function can be work normally, and software can also select edge transition detection of TMx_EXT pin by CAPEDGE (TIMERx_EXTCTL[2:1]) bits.

In event capture mode, software does not consider what timer counting operation mode is selected, the capture event occurred only if edge transition on TMx_EXT pin is detected.

6.28.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
TIMER Base Address: **TIMER_BA01 = 0x4005_0000** **TIMER_BA23 = 0x4005_1000**				
TIMER0_CTL	TIMER_BA01+0x00	R/W	Timer0 Control and Status Register	0x0000_0005
TIMER0_CMP	TIMER_BA01+0x04	R/W	Timer0 Compare Register	0x0000_0000
TIMER0_INTSTS	TIMER_BA01+0x08	R/W	Timer0 Interrupt Status Register	0x0000_0000
TIMER0_CNT	TIMER_BA01+0x0C	R	Timer0 Data Register	0x0000_0000
TIMER0_CAP	TIMER_BA01+0x10	R	Timer0 Capture Data Register	0x0000_0000
TIMER0_EXTCTL	TIMER_BA01+0x14	R/W	Timer0 External Control Register	0x0000_0000
TIMER0_EINTSTS	TIMER_BA01+0x18	R/W	Timer0 External Interrupt Status Register	0x0000_0000
TIMER1_CTL	TIMER_BA01+0x20	R/W	Timer1 Control and Status Register	0x0000_0005
TIMER1_CMP	TIMER_BA01+0x24	R/W	Timer1 Compare Register	0x0000_0000
TIMER1_INTSTS	TIMER_BA01+0x28	R/W	Timer1 Interrupt Status Register	0x0000_0000
TIMER1_CNT	TIMER_BA01+0x2C	R	Timer1 Data Register	0x0000_0000
TIMER1_CAP	TIMER_BA01+0x30	R	Timer1 Capture Data Register	0x0000_0000
TIMER1_EXTCTL	TIMER_BA01+0x34	R/W	Timer1 External Control Register	0x0000_0000
TIMER1_EINTSTS	TIMER_BA01+0x38	R/W	Timer1 External Interrupt Status Register	0x0000_0000
TIMER2_CTL	TIMER_BA23+0x00	R/W	Timer2 Control and Status Register	0x0000_0005
TIMER2_CMP	TIMER_BA23+0x04	R/W	Timer2 Compare Register	0x0000_0000
TIMER2_INTSTS	TIMER_BA23+0x08	R/W	Timer2 Interrupt Status Register	0x0000_0000
TIMER2_CNT	TIMER_BA23+0x0C	R	Timer2 Data Register	0x0000_0000
TIMER2_CAP	TIMER_BA23+0x10	R	Timer2 Capture Data Register	0x0000_0000

TIMER2_EXTCTL	TIMER_BA23+0x14	R/W	Timer2 External Control Register	0x0000_0000
TIMER2_EINTSTS	TIMER_BA23+0x18	R/W	Timer2 External Interrupt Status Register	0x0000_0000
TIMER3_CTL	TIMER_BA23+0x20	R/W	Timer3 Control and Status Register	0x0000_0005
TIMER3_CMP	TIMER_BA23+0x24	R/W	Timer3 Compare Register	0x0000_0000
TIMER3_INTSTS	TIMER_BA23+0x28	R/W	Timer3 Interrupt Status Register	0x0000_0000
TIMER3_CNT	TIMER_BA23+0x2C	R	Timer3 Data Register	0x0000_0000
TIMER3_CAP	TIMER_BA23+0x30	R	Timer3 Capture Data Register	0x0000_0000
TIMER3_EXTCTL	TIMER_BA23+0x34	R/W	Timer3 External Control Register	0x0000_0000
TIMER3_EINTSTS	TIMER_BA23+0x38	R/W	Timer3 External Interrupt Status Register	0x0000_0000

6.28.7 Register Description

Timer Control Register (TIMERx_CTL)

Register	Offset	R/W	Description	Reset Value
TIMER0_CTL	TIMER_BA01+0x00	R/W	Timer0 Control and Status Register	0x0000_0005
TIMER1_CTL	TIMER_BA01+0x20	R/W	Timer1 Control and Status Register	0x0000_0005
TIMER2_CTL	TIMER_BA23+0x00	R/W	Timer2 Control and Status Register	0x0000_0005
TIMER3_CTL	TIMER_BA23+0x20	R/W	Timer3 Control and Status Register	0x0000_0005

31	30	29	28	27	26	25	24
ICEDEBUG	**CNTEN**	**INTEN**	**OPMODE[1:0]**		**RSTCNT**	**ACTSTS**	**EXTCNTEN**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
WKEN	**TOGDIS2**	**TOGDIS1**	**Reserved**				**CNTDATEN**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PSC[7:0]							

Bits	Description	
[31]	**ICEDEBUG**	**ICE Debug Mode Acknowledge Disable Bit (Write Protect)** 0 = ICE debug mode acknowledgement effects TIMER counting. TIMER counter will be held while CPU is held by ICE. 1 = ICE debug mode acknowledgement Disabled. TIMER counter will keep going no matter CPU is held by ICE or not.
[30]	**CNTEN**	**Timer Enable Bit** 0 = Stops/Suspends counting. 1 = Starts counting. **Note1:** In stop status, and then set CNTEN to 1 will enable the 24-bit up counter to keep counting from the last stop counting value. **Note2:** This bit is auto-cleared by hardware in one-shot mode (TIMERx_CTL[28:27] = 00) when the timer interrupt flag TIF (TIMERx_INTSTS[0]) is generated.
[29]	**INTEN**	**Interrupt Enable Bit** 0 = Timer Interrupt Disabled. 1 = Timer Interrupt Enabled. If this bit is enabled, when the timer interrupt flag TIF is set to 1, the timer interrupt signal is generated and inform to CPU.

[28:27]	**OPMODE**	**Timer Operation Mode** 00 = The Timer controller is operated in One-shot mode. 01 = The Timer controller is operated in Periodic mode. 10 = The Timer controller is operated in Toggle-output mode. 11 = The Timer controller is operated in Continuous Counting mode.
[26]	**RSTCNT**	**Timer Reset Bit** Setting this bit will reset the 24-bit up counter value (TIMERx_CNT) and also force CNTEN (TIMERx_CTL[30]) to 0 if ACTSTS (TIMERx_CTL[25]) is 1. 0 = No effect. 1 = Reset 8-bit PSC counter, 24-bit up counter value and CNTEN bit.
[25]	**ACTSTS**	**Timer Active Status Bit (Read Only)** This bit indicates the 24-bit up counter status. 0 = 24-bit up counter is not active. 1 = 24-bit up counter is active.
[24]	**EXTCNTEN**	**Counter Mode Enable Bit** This bit is for external counting pin function enabled. When timer is used as an event counter, this bit should be set to 1 and select PCLK as timer clock source. 0 = External counter mode Disabled. 1 = External counter mode Enabled.
[23]	**WKEN**	**Wake-Up Enable** If this bit is set to 1, while timer interrupt flag TIF (TIMERx_INTSTS[0]) is 1 and INTEN (TIMERx_CTL[29]) is enabled, the timer interrupt signal will generate a wake-up trigger event to CPU. 0 = Wake-up trigger event Disabled if timer interrupt signal generated. 1 = Wake-up trigger event Enabled if timer interrupt signal generated.
[22]	**TOGDIS2**	**Toggle Output 2 Disable** Setting this bit will disable the Toggle output pins group 2. 0 = Toggle output pins group 2 Enabled. 1 = Toggle output pins group 2 Disabled. **Note1:** If both TOUT1 (group 1 pins) and TOUT2 (group 2 pins) function are enabled, toggle output signal is generated only from TOUT1 pins. **Note2:** The group2 pins are PD1, PE8, PE1, and PD11.
[21]	**TOGDIS1**	**Toggle Output 1 Disable** Setting this bit will disable the Toggle output pins group 1. 0 = Toggle output pins group 1 Enabled. 1 = Toggle output pins group 1 Disabled. **Note:** The group1 pins are PB4, PB1, PC6, and PC1.
[20:17]	**Reserved**	Reserved.
[16]	**CNTDATEN**	**Data Load Enable** When this bit is set, timer counter value (TIMERx_CNT) will be updated continuously to monitor internal 24-bit up counter value as the counter is counting. 0 = Timer Data Register update Disabled.

		1 = Timer Data Register update Enabled while timer counter is active.
[15:8]	**Reserved**	Reserved.
[7:0]	**PSC**	**PSC Counter** Timer input clock source is divided by (PSC+1) before it is fed to the timer up counter. If this field is 0 (PSC = 0), then there is no scaling.

Timer Compare Register (TIMERx_CMP)

Register	Offset	R/W	Description	Reset Value
TIMER0_CMP	TIMER_BA01+0x04	R/W	Timer0 Compare Register	0x0000_0000
TIMER1_CMP	TIMER_BA01+0x24	R/W	Timer1 Compare Register	0x0000_0000
TIMER2_CMP	TIMER_BA23+0x04	R/W	Timer2 Compare Register	0x0000_0000
TIMER3_CMP	TIMER_BA23+0x24	R/W	Timer3 Compare Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CMPDAT [23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CMPDAT [15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CMPDAT [7:0]							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**CMPDAT**	**Timer Compared Value** CMPDAT is a 24-bit compared value register. When the internal 24-bit up counter value is equal to CMPDAT value, the TIF (TIMERx_INTSTS[0] timer interrupt flag) will set to 1. Time-out period = (Period of timer clock input) * (8-bit PSC + 1) * (24-bit CMPDAT). **Note1:** Never write 0x0 or 0x1 in CMPDAT field, or the timer will run into unknown state. **Note2:** When timer is operating at continuous counting mode, the 24-bit up counter will keep counting continuously even if software writes a new value into CMPDAT field. But if timer is operating at other modes, the 24-bit up counter will restart counting and using newest CMPDAT value to be the timer compared value if software writes a new value into CMPDAT field.

Timer Interrupt Status Register (TIMERx_INTSTS)

Register	Offset	R/W	Description	Reset Value
TIMER0_INTSTS	TIMER_BA01+0x08	R/W	Timer0 Interrupt Status Register	0x0000_0000
TIMER1_INTSTS	TIMER_BA01+0x28	R/W	Timer1 Interrupt Status Register	0x0000_0000
TIMER2_INTSTS	TIMER_BA23+0x08	R/W	Timer2 Interrupt Status Register	0x0000_0000
TIMER3_INTSTS	TIMER_BA23+0x28	R/W	Timer3 Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**TWKF**	**TIF**

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**TWKF**	**Timer Wake-Up Flag** This bit indicates the interrupt wake-up flag status of timer. 0 = Timer does not cause CPU wake-up. 1 = CPU wake-up from Idle or power-down mode if timer time-out interrupt signal generated. **Note:** This bit is cleared by writing 1 to it.
[0]	**TIF**	**Timer Interrupt Flag** This bit indicates the interrupt flag status of Timer while TIMERx_CNT value reaches to CMPDAT value. 0 = No effect. 1 = TIMERx_CNT value matches the CMPDAT value. **Note:** This bit is cleared by writing 1 to it.

Timer Data Register (TIMERx_CNT)

Register	Offset	R/W	Description	Reset Value
TIMER0_CNT	TIMER_BA01+0x0C	R	Timer0 Data Register	0x0000_0000
TIMER1_CNT	TIMER_BA01+0x2C	R	Timer1 Data Register	0x0000_0000
TIMER2_CNT	TIMER_BA23+0x0C	R	Timer2 Data Register	0x0000_0000
TIMER3_CNT	TIMER_BA23+0x2C	R	Timer3 Data Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CNT[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CNT[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNT[7:0]							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**CNT**	**Timer Data Register** 1. EXTCNTEN (TIMERx_CTL[24]) = 0 : CNT is 24- bit counter value. User can read TIMERx_CNT for getting current 24- bit counter value if TIMERx_CTL[24] is set to 0 2. EXTCNTEN (TIMERx_CTL[24]) = 1 : CNT is 24- bit event counter value. User can read CNT for getting current 24- bit event counter value if TIMERx_CTL[24] is 1

Timer Capture Data Register (TIMERx_CAP)

Register	Offset	R/W	Description	Reset Value
TIMER0_CAP	TIMER_BA01+0x10	R	Timer0 Capture Data Register	0x0000_0000
TIMER1_CAP	TIMER_BA01+0x30	R	Timer1 Capture Data Register	0x0000_0000
TIMER2_CAP	TIMER_BA23+0x10	R	Timer2 Capture Data Register	0x0000_0000
TIMER3_CAP	TIMER_BA23+0x30	R	Timer3 Capture Data Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CAPDAT[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CAPDAT[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CAPDAT[7:0]							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**CAPDAT**	**Timer Capture Data Register** When CAPEN (TIMERx_EXTCTL[3]) bit is set, CAPFUNCS (TIMERx_EXTCTL[4]) bit is 0, and a transition on TMx_EXT pin matched the CAPEDGE (TIMERx_EXTCTL[2:1]) setting, CAPIF (TIMERx_EINTSTS[0]) will set to 1 and the current timer counter value (TIMERx_CNT value) will be auto-loaded into this CAPDAT field.

Timer External Control Register (TIMERx_EXTCTL)

Register	Offset	R/W	Description	Reset Value
TIMER0_EXTCTL	TIMER_BA01+0x14	R/W	Timer0 External Control Register	0x0000_0000
TIMER1_EXTCTL	TIMER_BA01+0x34	R/W	Timer1 External Control Register	0x0000_0000
TIMER2_EXTCTL	TIMER_BA23+0x14	R/W	Timer2 External Control Register	0x0000_0000
TIMER3_EXTCTL	TIMER_BA23+0x34	R/W	Timer3 External Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNTDBEN	**CAPDBEN**	**CAPIEN**	**CAPFUNCS**	**CAPEN**	**CAPEDGE**		**CNTPHASE**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**CNTDBEN**	**Timer Counter Pin De-Bounce Enable** 0 = TMx (x= 0~3) pin de-bounce Disabled. 1 = TMx (x= 0~3) pin de-bounce Enabled. If this bit is enabled, the edge detection of TMx pin is detected with de-bounce circuit.
[6]	**CAPDBEN**	**Timer External Capture Pin De-Bounce Enable** 0 = TMx_EXT (x= 0~3) pin de-bounce Disabled. 1 = TMx_EXT (x= 0~3) pin de-bounce Enabled. If this bit is enabled, the edge detection of TMx_EXT pin is detected with de-bounce circuit.
[5]	**CAPIEN**	**Timer External Interrupt Enable** 0 = TMx_EXT (x= 0~3) pin detection Interrupt Disabled. 1 = TMx_EXT (x= 0~3) pin detection Interrupt Enabled. CAPIEN is used to enable timer external interrupt. If CAPIEN enabled, timer will rise an interrupt when CAPIF = 1. For example, while CAPIEN = 1, CAPEN = 1, and CAPEDGE = 00, a 1 to 0 transition on the TEX pin will cause the CAPIF(TIMERx_EINTSTS[0]) interrupt flag to be set then the interrupt signal is generated and sent to NVIC to inform CPU.
[4]	**CAPFUNCS**	**Timer External Reset Counter / Capture Mode Select**

		0 = Transition on TMx_EXT (x= 0~3) pin is using to save the 24-bit timer counter value. (TIMERx_CNT value) to timer capture value (TIMERx_CAP value) if CAPIF (TIMERx_EINTSTS[0]) is set to 1 1 = Transition on TMx_EXT (x= 0~3) pin is using to reset the 24-bit timer counter value.
[3]	**CAPEN**	**Timer External Pin Enable** This bit enables the CAPFUNCS (TIMERx_EXTCTL[4]) function on the TMx_EXT pin. 0 = CAPFUNCS function of TMx_EXT (x= 0~3) pin will be ignored. 1 = CAPFUNCS function of TMx_EXT (x= 0~3) pin is active.
[2:1]	**CAPEDGE**	**Timer External Pin Edge Detect** 00 = A 1 to 0 transition on TMx_EXT (x= 0~3) pin will be detected. 01 = A 0 to 1 transition on TMx_EXT (x= 0~3) pin will be detected. 10 = Either 1 to 0 or 0 to 1 transition on TMx_EXT (x= 0~3) pin will be detected. 11 = Reserved.
[0]	**CNTPHASE**	**Timer External Count Phase** This bit indicates the detection phase of external counting pin. 0 = A falling edge of external counting pin will be counted. 1 = A rising edge of external counting pin will be counted.

Timer External Interrupt Status Register (TIMERx_INTSTS)

Register	Offset	R/W	Description	Reset Value
TIMER0_EINTSTS	TIMER_BA01+0x18	R/W	Timer0 External Interrupt Status Register	0x0000_0000
TIMER1_EINTSTS	TIMER_BA01+0x38	R/W	Timer1 External Interrupt Status Register	0x0000_0000
TIMER2_EINTSTS	TIMER_BA23+0x18	R/W	Timer2 External Interrupt Status Register	0x0000_0000
TIMER3_EINTSTS	TIMER_BA23+0x38	R/W	Timer3 External Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**CAPIF**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**CAPIF**	**Timer External Interrupt Flag** This bit indicates the timer external interrupt flag status. When CAPEN (TIMERx_EXTCTL[3]) bit is set, CAPFUNCS (TIMERx_EXTCTL[4]) bit is 0, and a transition on TMx_EXT (x= 0~3) pin matched the CAPEDGE (TIMERx_EXTCTL[2:1]) setting, this bit will set to 1 by hardware. 0 = TMx_EXT (x= 0~3) pin interrupt did not occur. 1 = TMx_EXT (x= 0~3) pin interrupt occurred. **Note:** This bit is cleared by writing 1 to it.

6.29 Watchdog Timer (WDT)

6.29.1 Overview

The Watchdog Timer is used to perform a system reset when system runs into an unknown state. This prevents system from hanging for an infinite period of time. Besides, this Watchdog Timer supports the function to wake-up system from Idle/Power-down mode.

6.29.2 Features

- 18-bit free running up counter for Watchdog Timer time-out interval.
- Selectable time-out interval (2^4 ~ 2^{18}) and the time-out interval is 104 ms ~ 26.3168 s if WDT_CLK = 10 kHz.
- System kept in reset state for a period of (1 / WDT_CLK) * 63
- Supports selectable Watchdog Timer reset delay period, including (1024+2)、(128+2) 、(16+2) or (1+2) WDT_CLK reset delay period.
- Supports force Watchdog Timer enabled after chip powered on or reset while CWDTEN (Config0[31] watchdog enable) bit is set to 0.
- Supports Watchdog Timer time-out wake-up function when WDT clock source is selected as 10 kHz low-speed oscillator.

6.29.3 Block Diagram

The Watchdog Timer clock control and block diagram are shown as follows.

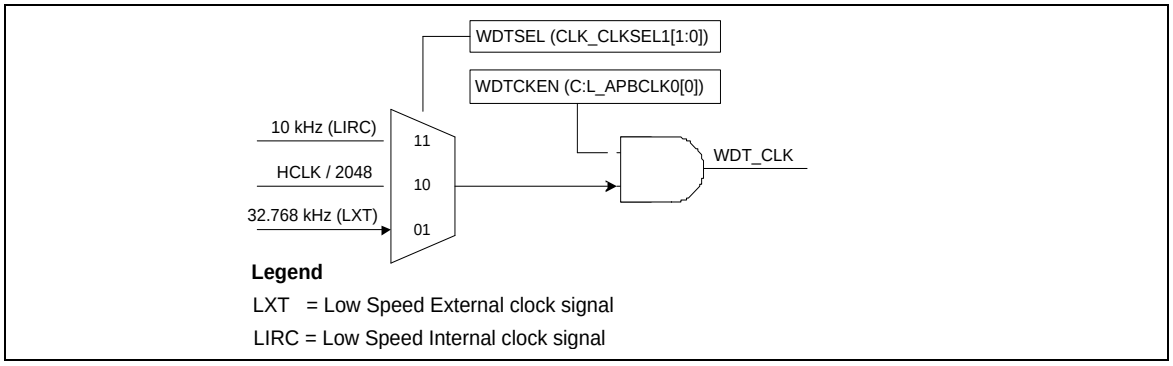
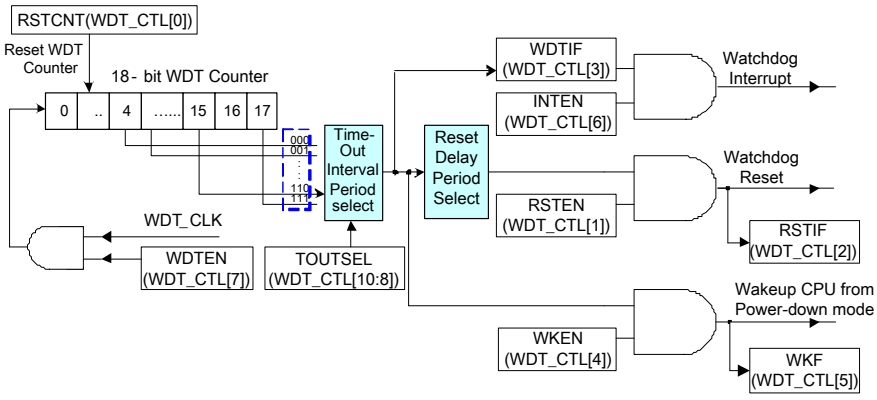


Figure 6.29-1 Watchdog Timer Clock Control



Note1: WDT resets CPU and lasts 63 WDT_CLK.

Note2: If user intends to use WDT to wake-up Idle/Power-down mode, it is recommended that CPU clock source is set the same as WDT clock source before CPU enters Power-Down mode.

Note3: The WDT reset delay period can be selected as 3/18/130/1026 WDT_CLK.

Figure 6.29-2 Watchdog Timer Block Diagram

6.29.4 Basic Configuration

The WDT peripheral clock is enabled in CLK_APBCLK[0] and clock source can be selected in CLK_CLKSEL1[1:0].
Or user can setting CONFIG[31] 0 to force Watchdog Timer enabled and active in 10 kHz after chip powered on or reset.

6.29.5 Functional Description

The Watchdog Timer (WDT) includes an 18-bit free running up counter with programmable time-out intervals. The following table shows the WDT time-out interval period selection and the following figure shows the WDT time-out interval and reset period timing.

WDT Time-out Interrupt

Setting WDTEN(WDT_CLT[7]) bit to 1 will enable the WDT function and the WDT counter to start counting up. There are eight time-out interval period can be selected by setting TOUTSEL(WDT_CTL[10:8]). When the WDT up counter reaches the TOUTSEL settings, WDT time-out interrupt will occur then IF flag will be set to 1 immediately.

WDT Reset Delay Period and Reset System

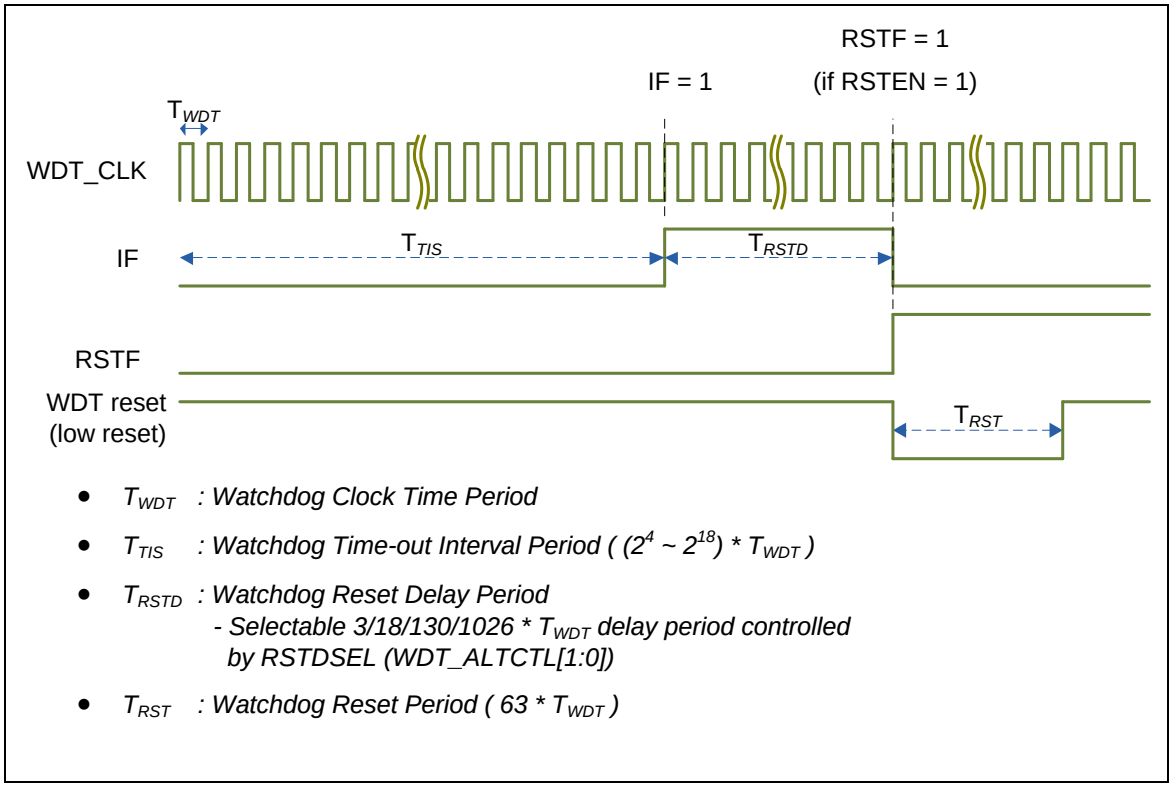
There is a specified T_{RSTD} delay period follows the IF flag is setting to 1. User must enabled RSTCNT(WDT_CTL[0]) bit to reset the 18-bit WDT up counter value to avoid generate WDT time-out reset signal before the T_{RSTD} delay period expires. If the WDT up counter value has not been cleared after the specific T_{RSTD} delay period expires, the WDT control will set RST(WDT_CTL[2])F flag to 1 if RSTEN(WDT_CTL[1]) bit is enabled, then chip enters to reset state immediately. Refer to Figure 6.29-3, the T_{RST} reset period will keep last 63 WDT clocks then chip restart executing program from reset vector (0x0000_0000). The RSTF flag will keep 1 after WDT time-out reset the chip, user can check RSTF flag by software to recognize the system has been reset by WDT time-out reset or not.

WDT Wake-up

If WDT clock source is selected to 10 kHz, system can be waken-up from Power-down mode while WDT time-out interrupt signal is generated and WKEN(WDT_CTL[4]) bit enabled. In the meanwhile, the WKF(WDT_CTL[5]) flag will set to 1 automatically, user can check WKF flag by software to recognize the system has been waken-up by WDT time-out interrupt or not.

TOUTSEL	Time-Out Interval Selection T_{TIS}	Interrupt Period T_{INT}	Reset Delay Period T_{RSTD}
000	$2^4 * T_{WDT}$	$1024 * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$
001	$2^6 * T_{WDT}$	$1024 * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$
010	$2^8 * T_{WDT}$	$1024 * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$
011	$2^{10} * T_{WDT}$	$1024 * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$
100	$2^{12} * T_{WDT}$	$1024 * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$
101	$2^{14} * T_{WDT}$	$1024 * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$
110	$2^{16} * T_{WDT}$	$1024 * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$
111	$2^{18} * T_{WDT}$	$1024 * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$

Table 6.29-1 Watchdog Timer Interval Selection



- T_{WDT} : *Watchdog Clock Time Period*
- T_{TIS} : *Watchdog Time-out Interval Period ((2^4 ~ 2^{18}) * T_{WDT})*
- T_{RSTD} : *Watchdog Reset Delay Period*
  *- Selectable 3/18/130/1026 * T_{WDT} delay period controlled by RSTDSEL (WDT_ALTCTL[1:0])*
- T_{RST} : *Watchdog Reset Period (63 * T_{WDT})*

Figure 6.29-3 Timing of Interrupt and Reset Signal

6.29.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
WDT Base Address: **WDT_BA = 0x4004_0000**				
WDT_CTL	WDT_BA+0x00	R/W	Watchdog Timer Control Register	0x0000_0700
WDT_ALTCTL	WDT_BA+0x04	R/W	Watchdog Timer Alternative Control Register	0x0000_0000

6.29.7 Register Description

Watchdog Timer Control Register (WDT_CTL)

Register	Offset	R/W	Description	Reset Value
WDT_CTL	WDT_BA+0x00	R/W	Watchdog Timer Control Register	0x0000_0700

Note: All bits written in this register are write-protected. Programming it needs to write “59h”, “16h”, “88h” to address 0x4000_0100 to disable register protection. Refer to the register SYS_REGLCTL at address SYS_BA+0x100.

31	30	29	28	27	26	25	24
ICEDEBUG	**Reserved**						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**TOUTSEL**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WDTEN	**INTEN**	**WKF**	**WKEN**	**IF**	**RSTF**	**RSTEN**	**RSTCNT**

Bits	Description	
[31]	**ICEDEBUG**	**ICE Debug Mode Acknowledge Disable Bit (Write Protect)** 0 = ICE debug mode acknowledgement affects Watchdog Timer counting. Watchdog Timer counter will be held while CPU is held by ICE. 1 = ICE debug mode acknowledgement Disabled. **Note:** Watchdog Timer counter will keep going no matter CPU is held by ICE or not.
[30:11]	**Reserved**	Reserved.
[10:8]	**TOUTSEL**	**Watchdog Timer Time-Out Interval Selection (Write Protect)** These three bits select the time-out interval period for the Watchdog Timer. 000 = 2^4 *T_{WDT}. 001 = 2^6 *T_{WDT}. 010 = 2^8 *T_{WDT}. 011 = 2^{10} *T_{WDT}. 100 = 2^{12} *T_{WDT}. 101 = 2^{14} *T_{WDT}. 110 = 2^{16} *T_{WDT}. 111 = 2^{18} *T_{WDT}.
[7]	**WDTEN**	**Watchdog Timer Enable Bit (Write Protect)** 0 = Watchdog Timer Disabled (This action will reset the internal counter). 1 = Watchdog Timer Enabled. **Note:** If CWDTEN (Config0[31] watchdog enable) bit is set to 0, this bit is forced as 1 and software cannot change this bit to 0.
[6]	**INTEN**	**Watchdog Timer Interrupt Enable Bit (Write Protect)** If this bit is enabled, the WDT time-out interrupt signal is generated and inform to CPU.

		0 = Watchdog Timer interrupt Disabled. 1 = Watchdog Timer interrupt Enabled.
[5]	**WKF**	**Watchdog Timer Wake-Up Flag** This bit indicates the interrupt wake-up flag status of WDT 0 = Watchdog Timer does not cause chip wake-up. 1 = Chip wake-up from Idle or Power-down mode if WDT time-out interrupt signal generated. **Note:** This bit is cleared by writing 1 to it.
[4]	**WKEN**	**Watchdog Timer Wake-Up Function Enable Bit (Write Protect)** If this bit is set to 1, while WDT interrupt flag IF(WDT_CTL[3]) is generated to 1 and INTEN (WDT_CTL[6] WDT interrupt enable) is enabled, the WDT time-out interrupt signal will generate a wake-up trigger event to chip. 0 = Wake-up trigger event Disabled if WDT time-out interrupt signal generated. 1 = Wake-up trigger event Enabled if WDT time-out interrupt signal generated. **Note:** Chip can be woken-up by WDT time-out interrupt signal generated only if WDT clock source is selected to 10 kHz oscillator.
[3]	**IF**	**Watchdog Timer Interrupt Flag** This bit will set to 1 while WDT counter value reaches the selected WDT time-out interval 0 = Watchdog Timer time-out interrupt did not occur. 1 = Watchdog Timer time-out interrupt occurred. **Note:** This bit is cleared by writing 1 to it.
[2]	**RSTF**	**Watchdog Timer Reset Flag** This bit indicates the system has been reset by WDT time-out reset or not. 0 = Watchdog Timer time-out reset did not occur. 1 = Watchdog Timer time-out reset occurred. **Note:** This bit is cleared by writing 1 to it.
[1]	**RSTEN**	**Watchdog Timer Reset Enable Bit (Write Protect)** Setting this bit will enable the Watchdog Timer time-out reset function If the WDT counter value has not been cleared after the specific WDT reset delay period expires. 0 = Watchdog Timer time-out reset function Disabled. 1 = Watchdog Timer time-out reset function Enabled.
[0]	**RSTCNT**	**Clear Watchdog Timer (Write Protect)** 0 = No effect. 1 = Reset the internal 18-bit WDT counter. **Note:** This bit will be automatically cleared by hardware.

Watchdog Timer Alternative Control Register (WDT_ALTCTL)

Register	Offset	R/W	Description	Reset Value
WDT_ALTCTL	WDT_BA+0x04	R/W	Watchdog Timer Alternative Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**RSTDSEL**	

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1:0]	**RSTDSEL**	**Watchdog Timer Reset Delay Selection (Write Protect)** When WDT time-out happened, software has a time named WDT reset delay period to clear WDT counter to prevent WDT time-out reset happened. Software can select a suitable value of WDT reset delay period for different WDT time-out period. 00 = Watchdog Timer reset delay period is (1024+2) * WDT_CLK. 01 = Watchdog Timer reset delay period is (128+2) * WDT_CLK. 10 = Watchdog Timer reset delay period is (16+2) * WDT_CLK. 11 = Watchdog Timer reset delay period is (1+2) * WDT_CLK. **Note:** This register will be reset to 0 if WDT time-out reset happened

6.30 Window Watchdog Timer (WWDT)

6.30.1 Overview

The Window Watchdog Timer is used to perform a system reset within a specified window period to prevent software run to uncontrollable status by any unpredictable condition.

6.30.2 Features

- 6-bit down counter (WWDT_CNT[5:0]) and 6-bit compare value (WWDT_CTL[21:16]) to make the window period flexible
- Selectable maximum 11-bit WWDT clock prescale (WWDT_CTL[11:8]) to make WWDT time-out interval variable

6.30.3 Block Diagram

The Window Watchdog Timer block diagram is shown as follows.

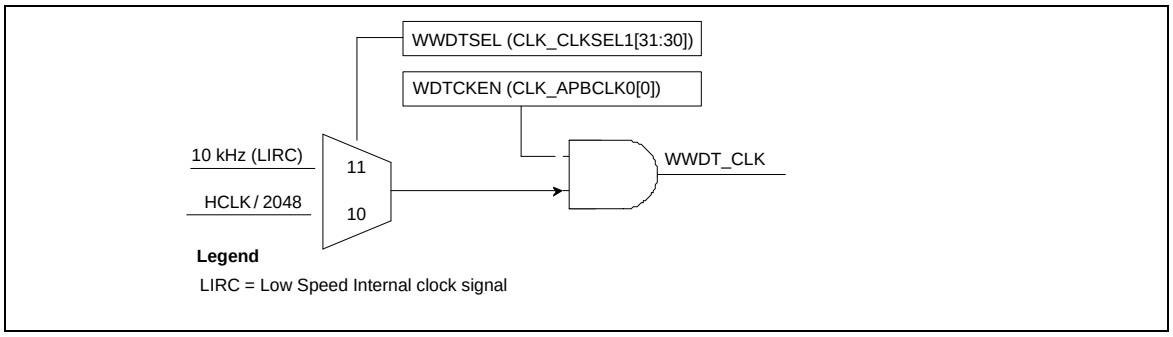


Figure 6.30-1 Window Watchdog Timer Clock Control

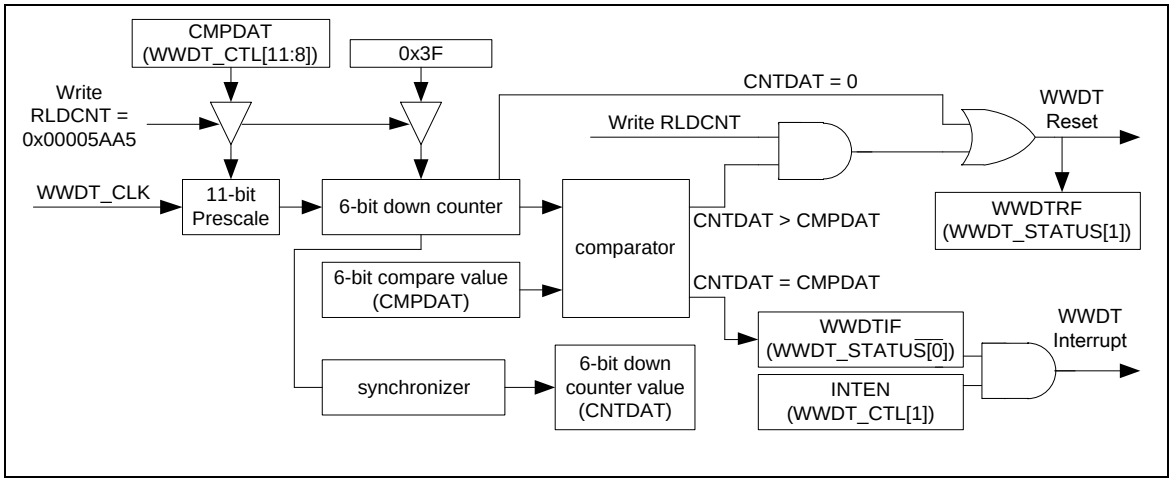


Figure 6.30-2 Window Watchdog Timer Block Diagram

6.30.4 Basic Configuration

The WWDT peripheral clock is enabled in CLK_APBCLK[0] and clock source can be selected in CLK_CLKSEL1[31:30].

6.30.5 Functional Description

The Window Watchdog Timer includes a 6-bit down counter with programmable prescale value to define different time-out intervals. The clock source of 6-bit Window Watchdog Timer is based on system clock divide 2048 (HCLK/2048) or internal 10 kHz oscillator with a programmable maximum 11-bit prescale value. Also, the programmable 11-bit prescale value is controlled by PSCSEL (WWDT_CTL[11:8] WWDT prescale period select) and the correlate of PSCSEL and prescale value are listed in the following table.

PSCSEL	Prescaler Value	Time-Out Period	Time-Out Interval (WWDT_CLK= 10 KHz)
0000	1	1 * 64 * T_{WWDT}	6.4 ms
0001	2	2 * 64 * T_{WWDT}	12.8 ms
0010	4	4 * 64 * T_{WWDT}	25.6 ms
0011	8	8 * 64 * T_{WWDT}	51.2 ms
0100	16	16 * 64 * T_{WWDT}	102.4 ms
0101	32	32 * 64 * T_{WWDT}	204.8 ms
0110	64	64 * 64 * T_{WWDT}	409.6 ms
0111	128	128 * 64 * T_{WWDT}	819.2 ms
1000	192	192 * 64 * T_{WWDT}	1.2288 s
1001	256	256 * 64 * T_{WWDT}	1.6384 s
1010	384	384 * 64 * T_{WWDT}	2.4576 s
1011	512	512 * 64 * T_{WWDT}	3.2768 s
1100	768	768 * 64 * T_{WWDT}	4.9152 s
1101	1024	1024 * 64 * T_{WWDT}	6.5536 s
1110	1536	1536 * 64 * T_{WWDT}	9.8304 s
1111	2048	2048 * 64 * T_{WWDT}	13.1072 s

Table 6.30-1 Window Watchdog Prescaler Value Selection

The Window Watchdog Timer can be enabled only once by software setting WWDTEN (WWDT_CTL[0) bit to 1 after chip power on or reset and the WWDT down counter will start counting from 0x3F and cannot be stopped by software unless chip has been reset again.

During down counting by the WWDT counter, the WWDTIF (WWDT_STATUS[0]) is set to 1 if the WWDT counter value is equal to CMPDAT (WWDT_CTL[21:16]) value; if INTEN (WWDT_CTL[1]) is also set to 1 by software, the WWDT time-out interrupt signal is generated also while WWDTIF is set to 1 by hardware.

The WWDT time-out reset signal is generated when the WWDT counter value reaches 0. Before WWDT counter down counting to 0, software can write **0x00005AA5** to WWDT_RLDCNT register to reload WWDT internal counter value to 0x3F to prevent WWDT time-out reset from happening when the current WWDT counter value (WWDT_CNT value) is equal to or less than CMPDAT value. If the current WWDT counter value (WWDT_CNT value) is greater than CMPDAT value and software writes **0x00005AA5** to the WWDT_RLDCNT register, WWDT reset signal will be generated to cause chip reset. Figure 6.30-3 shows the reset and reload behavior of WWDT.

To prevent program unexpectedly disable Window Watchdog Timer counter, the control register WWDT_CTL can only be written once after chip is powered on or reset. Software cannot disable Window Watchdog Timer counter counting, change time-out prescale period or change window compare value while WWDTEN bit has been enabled by software unless chip is reset.

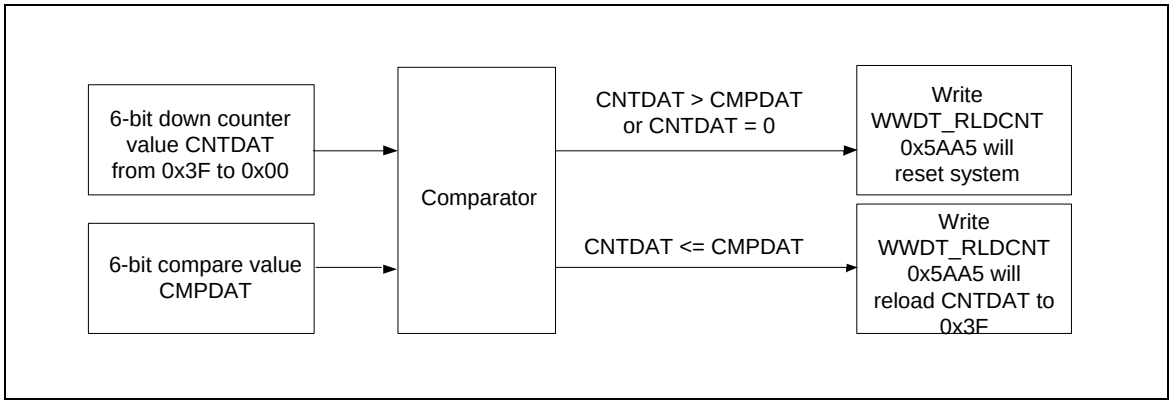


Figure 6.30-3 Window Watchdog Timer Reset and Reload Behavior

When user writes **0x00005AA5** to WWDT_RLDCNT register to reload WWDT counter value to 0x3F, it needs 3 WWDT clocks to sync reload command to actually perform reload action. This means if user set PSCSEL (WWDT_CTL[11:8]t) to 0000, the prescale value should be as 1, and the CMPDAT (WWDT_CTL[21:16]) value must be larger than 2; otherwise, writing WWDT_RLDCNT to reload WWDT counter value to 0x3F is unavailable while WWDTIF(WWDT_STATUS[0]) is generated and WWDT reset system event always happened. The following table shows the limitation of CMPDAT.

PSCSEL	Prescale Value	Valid CMPDAT Value
0000	1	0x3 ~ 0x3F
0001	2	0x2 ~ 0x3F
Others	Others	0x0 ~ 0x3F

Table 6.30-2 CMPDAT Setting Limitation

6.30.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
WWDT Base Address: **WWDT_BA = 0x4004_0100**				
WWDT_RLDCNT	WWDT_BA+0x00	W	Window Watchdog Timer Reload Counter Register	0x0000_0000
WWDT_CTL	WWDT_BA+0x04	R/W	Window Watchdog Timer Control Register	0x003F_0800
WWDT_STATUS	WWDT_BA+0x08	R/W	Window Watchdog Timer Status Register	0x0000_0000
WWDT_CNT	WWDT_BA+0x0C	R	Window Watchdog Timer Counter Value Register	0x0000_003F

6.30.7 Register Description

Window Watchdog Timer Reload Counter Register (WWDT_RLDCNT)

Register	Offset	R/W	Description	Reset Value
WWDT_RLDCNT	WWDT_BA+0x00	W	Window Watchdog Timer Reload Counter Register	0x0000_0000

31	30	29	28	27	26	25	24
RLDCNT[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RLDCNT[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RLDCNT[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RLDCNT[7:0]							

Bits	Description	
[31:0]	**RLDCNT**	**WWDT Reload Counter Bit** Writing 0x00005AA5 to this register will reload the Window Watchdog Timer counter value to 0x3F. **Note:** Software can only write RLDCNT to reload WWDT counter value when current WWDT counter value between 0 and CMPDAT. If software writes RLDCNT when current WWDT counter value is larger than CMPDAT, WWDT reset signal will generate immediately.

Window Watchdog Timer Control Register (WWDT_CTL)

Register	Offset	R/W	Description	Reset Value
WWDT_CTL	WWDT_BA+0x04	R/W	Window Watchdog Timer Control Register	0x003F_0800

Note: This register can be write only one time after chip is powered on or reset.

31	30	29	28	27	26	25	24
ICEDEBUG	Reserved						
23	22	21	20	19	18	17	16
Reserved		CMPDAT					
15	14	13	12	11	10	9	8
Reserved				PSCSEL			
7	6	5	4	3	2	1	0
Reserved						INTEN	WWDTEN

Bits	Description	
[31]	**ICEDEBUG**	**ICE Debug Mode Acknowledge Disable Bit** 0 = ICE debug mode acknowledgement effects WWDT counting. WWDT down counter will be held while CPU is held by ICE. 1 = ICE debug mode acknowledgement Disabled. WWDT down counter will keep going no matter CPU is held by ICE or not.
[30:22]	**Reserved**	Reserved.
[21:16]	**CMPDAT**	**WWDT Window Compare Bits** Set this register to adjust the valid reload window. **Note:** Software can only write RLDCNT to reload WWDT counter value when current WWDT counter value between 0 and CMPDAT. If Software writes RLDCNT when current WWDT counter value larger than CMPDAT, WWDT reset signal will generate immediately.
[15:12]	**Reserved**	Reserved.
[11:8]	**PSCSEL**	**WWDT Counter Prescale Period Selection** 0000 = Pre-scale is 1; Max time-out period is 1 * 64 * T_{WWDT}. 0001 = Pre-scale is 2; Max time-out period is 2 * 64 * T_{WWDT}. 0010 = Pre-scale is 4; Max time-out period is 4 * 64 * T_{WWDT}. 0011 = Pre-scale is 8; Max time-out period is 8 * 64 * T_{WWDT}. 0100 = Pre-scale is 16; Max time-out period is 16 * 64 * T_{WWDT}. 0101 = Pre-scale is 32; Max time-out period is 32 * 64 * T_{WWDT}. 0110 = Pre-scale is 64; Max time-out period is 64 * 64 * T_{WWDT}. 0111 = Pre-scale is 128; Max time-out period is 128 * 64 * T_{WWDT}. 1000 = Pre-scale is 192; Max time-out period is 192 * 64 * T_{WWDT}. 1001 = Pre-scale is 256; Max time-out period is 256 * 64 * T_{WWDT}. 1010 = Pre-scale is 384; Max time-out period is 384 * 64 * T_{WWDT}. 1011 = Pre-scale is 512; Max time-out period is 512 * 64 * T_{WWDT}.

		1100 = Pre-scale is 768; Max time-out period is 768 * 64 * T_{WWDT}. 1101 = Pre-scale is 1024; Max time-out period is 1024 * 64 * T_{WWDT}. 1110 = Pre-scale is 1536; Max time-out period is 1536 * 64 * T_{WWDT}. 1111 = Pre-scale is 2048; Max time-out period is 2048 * 64 * T_{WWDT}.
[7:2]	**Reserved**	Reserved.
[1]	**INTEN**	**WWDT Interrupt Enable Bit** If this bit is enabled, the WWDT counter compare match interrupt signal is generated and inform to CPU. 0 = WWDT counter compare match interrupt Disabled. 1 = WWDT counter compare match interrupt Enabled.
[0]	**WWDTEN**	**WWDT Enable Bit** Set this bit to enable Window Watchdog Timer counter counting. 0 = Window Watchdog Timer counter is stopped. 1 = Window Watchdog Timer counter is starting counting.

Window Watchdog Timer Status Register (WWDT_STATUS)

Register	Offset	R/W	Description	Reset Value
WWDT_STATUS	WWDT_BA+0x08	R/W	Window Watchdog Timer Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**WWDTRF**	**WWDTIF**

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**WWDTRF**	**WWDT Timer-Out Reset Flag** This bit indicates the system has been reset by WWDT time-out reset or not. 0 = WWDT time-out reset did not occur. 1 = WWDT time-out reset occurred. **Note:** This bit is cleared by writing 1 to it.
[0]	**WWDTIF**	**WWDT Compare Match Interrupt Flag** This bit indicates the interrupt flag status of WWDT while WWDT counter value matches CMPDAT value. 0 = No effect. 1 = WWDT counter value matches CMPDAT value. **Note:** This bit is cleared by writing 1 to it.

Window Watchdog Timer Counter Value Register (WWDT_CNT)

Register	Offset	R/W	Description	Reset Value
WWDT_CNT	WWDT_BA+0x0C	R	Window Watchdog Timer Counter Value Register	0x0000_003F

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**CNTDAT**					

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5:0]	**CNTDAT**	**WWDT Counter Value** This register reflects the current WWDT counter value and is read only.

6.31 UART Interface Controller (UART)

The NUC442/NUC472 provides up to six channels of Universal Asynchronous Receiver/Transmitters (UART). UART0 supports High-speed UART and UART1~5 perform Normal Speed UART, besides, all the UART channels support flow control function.

6.31.1 Overview

The Universal Asynchronous Receiver/Transmitter (UART) performs a serial-to-parallel conversion on data received from the peripheral, and a parallel-to-serial conversion on data transmitted from the CPU. The UART controller also supports IrDA SIR Function, LIN master/slave mode function and RS-485 mode functions. Each UART channel supports seven types of interrupts including transmitter FIFO empty interrupt (INT_THRE), receiver threshold level reaching interrupt (INT_RDA), line status interrupt (parity error or framing error or break interrupt) (INT_RLS), receiver buffer time-out interrupt (INT_TOUT), MODEM/Wake-up status interrupt (INT_MODEM), Buffer error interrupt (INT_BUF_ERR) and LIN interrupt (INT_LIN).

The UART0 is built-in with a 64-byte transmitter FIFO (TX_FIFO) and a 64-byte receiver FIFO (RX_FIFO) that reduces the number of interrupts presented to the CPU and the UART1~2 are equipped 16-byte transmitter FIFO (TX_FIFO) and 16-byte receiver FIFO (RX_FIFO). The CPU can read the status of the UART at any time during the operation. The reported status information includes the type and condition of the transfer operations being performed by the UART, as well as 4 error conditions (parity error, framing error, break interrupt and buffer error) probably occur while receiving data. The UART includes a programmable baud rate generator that is capable of dividing clock input by divisors to produce the serial clock that transmitter and receiver need.

All of the controllers support auto-flow control function that uses two low-level signals, /CTS (clear-to-send) and /RTS (request-to-send), to control the flow of data transfer between the UART and external devices (ex: Modem). When auto-flow is enabled, the UART is not allowed to receive data until the UART asserts /RTS to external device. When the number of bytes in the RX FIFO equals the value of RTSTRGLV (UART_FIFO [19:16]), the /RTS is de-asserted. The UART sends data out when UART controller detects /CTS is asserted from external device. If a valid asserted /CTS is not detected the UART controller will not send data out.

The UART controllers also provides Serial IrDA (SIR, Serial Infrared) function (User must set IrDA_EN (UART_FUNCSEL[1]) to enable IrDA function). The SIR specification defines a short-range infrared asynchronous serial transmission mode with one start bit, 8 data bits, and 1 stop bit. The maximum data rate is 115.2 Kbps (half duplex). The IrDA SIR block contains an IrDA SIR Protocol encoder/decoder. The IrDA SIR protocol is half-duplex only. So it cannot transmit and receive data at the same time. The IrDA SIR physical layer specifies a minimum 10ms transfer delay between transmission and reception. This delay feature must be implemented by software.

The alternate function of UART controllers is LIN (Local Interconnect Network) function. The LIN mode is selected by setting the LIN_EN bit (UART_FUNCSEL[0]). In LIN mode, one start bit and 8-bit data format with 1-bit stop bit are required in accordance with the LIN standard.

For the NUC442/NUC472 series, another alternate function of UART controllers is RS-485 9-bit mode function, and direction control provided by RTS pin to implement the function by software. The RS-485 mode is selected by setting the UART_FUNCSEL register to select RS-485 function. The RS-485 driver control is implemented using the RTS control signal from an asynchronous serial port to enable the RS-485 driver. In RS-485 mode, many characteristics of the RX and TX are the same as UART.

6.31.2 Features

- Full duplex, asynchronous communications

- Separate receive / transmit 64/16 bytes (UART0/UART1~5) entry FIFO for data payloads
- Supports hardware auto flow control/flow control function (CTS, RTS) and programmable RTS flow control trigger level
- Programmable receiver buffer trigger level
- Supports programmable baud-rate generator for each channel individually
- Supports CTS wake-up function
- Supports 8-bit receiver buffer time-out detection function
- UART0~5 can be served by the DMA controller
- Programmable transmitting data delay time between the last stop and the next start bit by setting DLY (UART_TOUT [15:8]) register
- Supports break error, frame error, parity error and receive / transmit buffer overflow detect function
- Fully programmable serial-interface characteristics
  - Programmable number of data bit, 5-, 6-, 7-, 8-bit character
  - Programmable parity bit, even, odd, no parity or stick parity bit generation and detection
  - Programmable stop bit, 1, 1.5, or 2 stop bit generation
- Supports IrDA SIR function mode
  - Support for 3-/16-bit duration
- Supports LIN function mode
  - Supports LIN master/slave mode
  - Supports programmable break generation function for transmitter
  - Supports break detect function for receiver
  - Supports master identifier field parity generation and slave identifier field parity check function
  - Supports LIN slave header reception function
  - Supports LIN slave automatic resynchronization function
  - Supports LIN slave header error detect function
- Supports RS-485 function mode
  - Supports RS-485 9-bit mode
  - Supports hardware or software direct enable control provided by RTS pin

6.31.3 Block Diagram

The UART clock control and block diagrams are shown below.

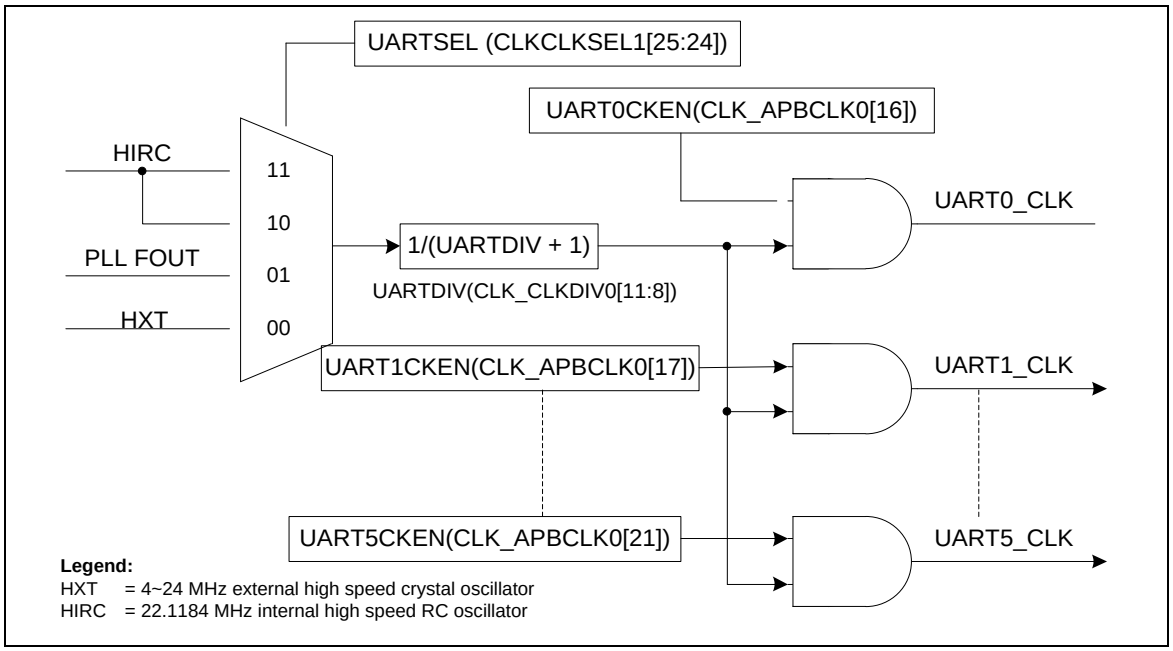


Figure 6.31-1 UART Clock Control Diagram

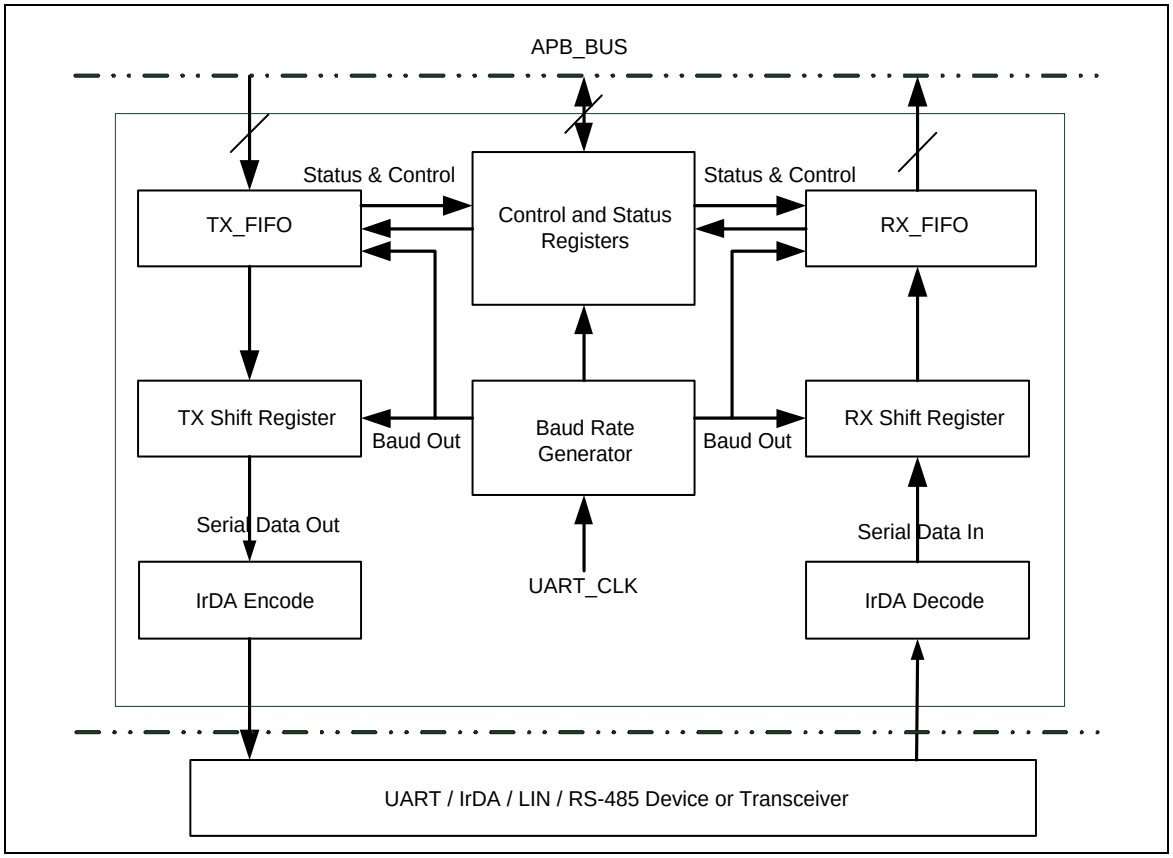


Figure 6.31-2 UART Block Diagram

TX_FIFO
The transmitter is buffered with a 64/16 byte FIFO to reduce the number of interrupts presented to the CPU.

RX_FIFO
The receiver is buffered with a 64/16 byte FIFO (plus three error bits per byte) to reduce the number of interrupts presented to the CPU.

TX shift Register
This block is the shifting the transmitting data out serially control block.

RX shift Register
This block is the shifting the receiving data in serially control block.

Baud Rate Generator
Divide the external clock by the divisor to get the desired baud rate clock. Refer to baud rate equation.

IrDA Encode
This block is IrDA encode control block.

IrDA Decode
This block is IrDA decode control block.

Control and Status Register
This field is register set including the FIFO control registers (UART_FIFO), FIFO status registers (UART_FIFOSTS), and line control register (UART_LINE) for transmitter and receiver. The time-

out control register (UART_TOUT) identifies the condition of time-out interrupt. This register set also includes the interrupt enable register (UART_INTEN) and interrupt status register (UART_INTSTS) to enable or disable the responding interrupt and to identify the occurrence of the responding interrupt. There are seven types of interrupts, transmitter FIFO empty interrupt(INT_THRE), receiver threshold level reaching interrupt (INT_RDA), line status interrupt (parity error or framing error or break interrupt) (INT_RLS), time-out interrupt (INT_TOUT), MODEM/Wake-up status interrupt (INT_MODEM), Buffer error interrupt (INT_BUF_ERR) and LIN receiver break field detected interrupt (INT_LIN_RX_BREAK).

The following diagram demonstrates the auto-flow control block diagram.

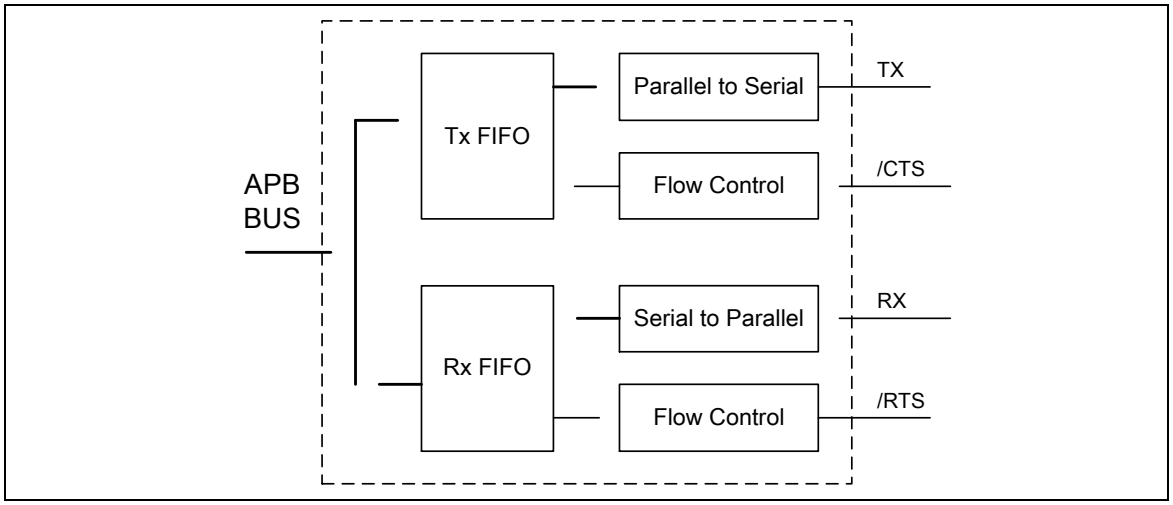


Figure 6.31-3 Auto Flow Control Block Diagram

6.31.4 IrDA Mode

The UART supports IrDA SIR (Serial Infrared) Transmit Encoder and Receive Decoder, and IrDA mode is selected by setting the **IrDA_EN** bit in **UART_FUNCSEL** register.

In IrDA mode, the BAUDM1 (UART_BAUD[29]) must be disabled.

Baud Rate = Clock / (16 * BRD), where BRD is Baud Rate Divider in UART_BAUD register.

The following diagram demonstrates the IrDA control block diagram.

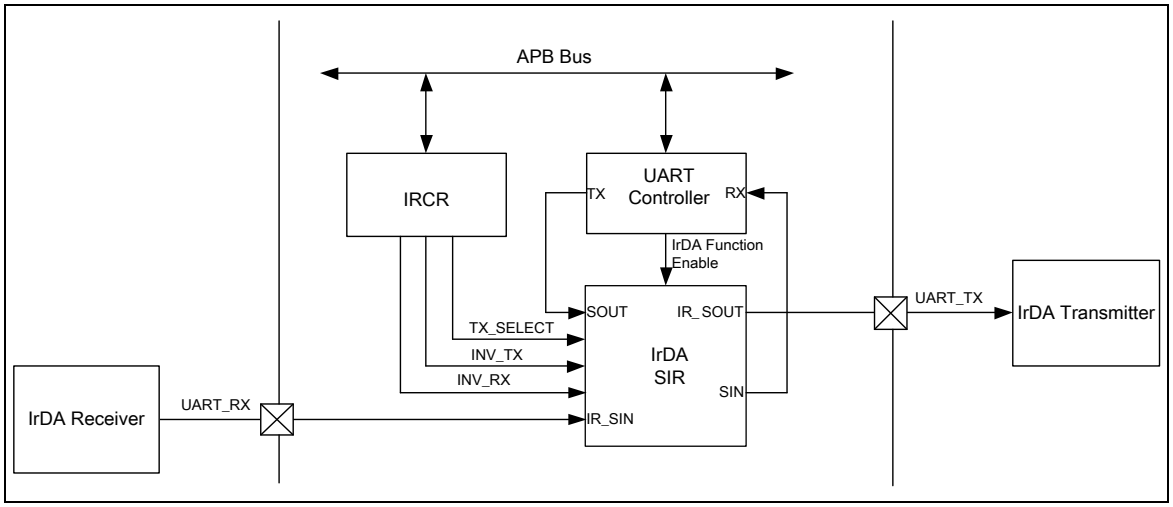


Figure 6.31-4 IrDA Block Diagram

6.31.4.1 IrDA SIR Transmit Encoder

The IrDA SIR Transmit Encoder modulate Non-Return-to Zero (NRZ) transmit bit stream output from UART. The IrDA SIR physical layer specifies use of Return-to-Zero, Inverted (RZI) modulation scheme which represent logic 0 as an infra light pulse. The modulated output pulse stream is transmitted to an external output driver and infrared Light Emitting Diode.

The transmitted pulse width is specified as 3/16 period of baud rate.

6.31.4.2 IrDA SIR Receive Decoder

The IrDA SIR Receive Decoder demodulates the return-to-zero bit stream from the input detector and outputs the NRZ serial bits stream to the UART received data input. The decoder input is normally high in the idle state. (Because of this, IRCR bit 6 should be set as 1 by default)

A start bit is detected when the decoder input is LOW

6.31.4.3 IrDA SIR Operation

The IrDA SIR encoder/decoder provides functionality which converts between UART data stream and half duplex serial SIR interface. The following diagram shows the IrDA encoder/decoder waveform.

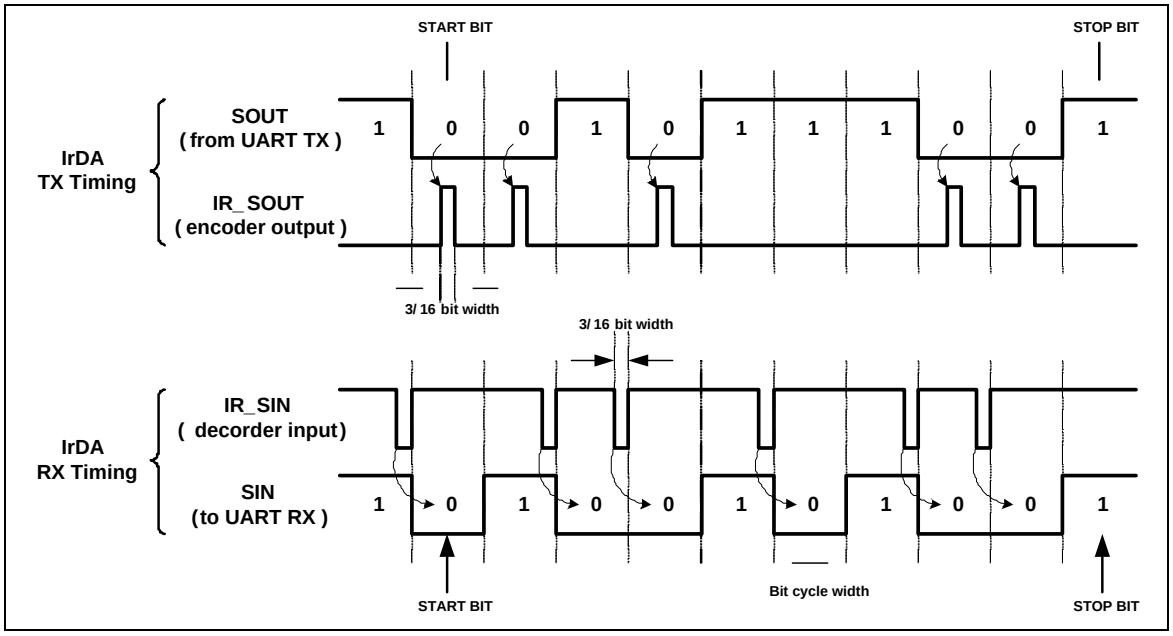


Figure 6.31-5 IrDA TX/RX Timing Diagram

6.31.5 LIN (Local Interconnection Network) Mode

The UART supports LIN function, and LIN mode is selected by setting the UART_FUNCSEL register. The UART support LIN break/delimiter generation and break/delimiter detection in LIN master mode, support header detection and automatic resynchronization in LIN slave mode.

6.31.5.1 Structure of LIN Frame

According to the LIN protocol, all information is transmitted packed as frames; a frame consist (provided by the master task) a header and a response (provided by a slave task). That is any communication on the LIN bus is started by the master sending a header, followed by the response. The header (provided by the master task) consists of a break field and sync field followed by a frame identifier (frame ID). The frame identifier uniquely defines the purpose of the frame. The slave task appointed for providing the response associated with the frame ID and the response consists of a data field and a checksum field. The following diagram is the structure of LIN function mode.

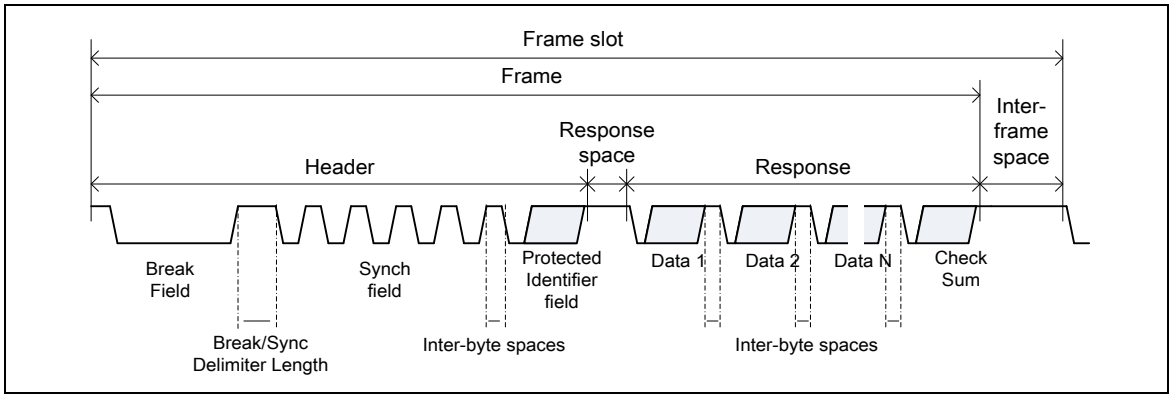


Figure 6.31-6 LIN Frame Structure

6.31.5.2 LIN Master Mode

The UART controller supports LIN master mode by setting the UART_FUNCSEL register. In LIN mode, each byte field is initialed by a start bit with value zero (dominant), followed by 8 data bits (WLS (UART_LINE[1:0]) = 2'b11) and no parity bit, LSB is first and ended by 1 stop bit (NSB (UART_LINE[2]) = 1) with value one (recessive) in accordance with the LIN standard. In LIN master mode, software may need some initial process, and the initialization process flow of LIN master is shown as follows:

1. Select the desired baud-rate by setting the UART_BAUD register.
2. Select LIN function mode by setting UART_FUNCSEL register.
3. Configure the data length to 8 bits by setting (WLS bits (UART_LINE[1:0]) = 2'b11) and disable parity check by clearing PBE bit (UART_LINE[3]) and configure the stop bit to 1 by clearing NSB bit (UART_LINE[2]).

A complete header consists of a break field and sync field followed by a frame identifier (frame ID). The UART controller can be selected header sending by three header selected mode. The header selected mode can be "break field" or "break field and sync field" or "break field, sync field and frame ID field" by setting HSEL (UART_LINCTL[23:22]). If the header selected is "break field", software must handle the following sequence to sending a complete header to bus by filled sync data (0x55) and frame ID data to UART_DAT register. If the header selected is "break field and sync field", software must handle the sequence to sending a complete header to bus by filled frame ID data to UART_DAT register, and if the header selected is "break field, sync field and frame ID field", hardware will control the header sending sequence automatically but software must filled frame ID data to PID (UART_LINCTL[31:24]) register. When operating in header selected is "break field, sync field and frame ID field" mode, the frame ID parity bit can be calculated by software or hardware depending on the IDPEN bit (UART_LINCTL[9]) setting.

While transmitting LIN data, software can monitor the LIN bus transfer state by hardware or software. User can enable hardware monitoring by setting BITERREN bit (UART_LINCTL[12]), and when operating in LIN transmitter state, if the input pin (SIN) state is not equal to the output pin (SOUT) state that the hardware wills generator an interrupt to CPU. User also can monitor the LIN bus transfer state by check the read back data in UART_DAT register. The following sequence is a program sequence example:

Procedure without software error monitoring in master mode:

1. Choose the hardware transmission header field by setting HSEL bits (UART_LINCTL[23:22]).
2. Request header transmission by setting the SENDH bit (UART_LINCTL[8])
3. Wait for the TXEMPTYF flag (UART_FIFOSTS[28]).

Note1: The break + delimiter default setting is 13 dominant bits and 1 delimiter bit, software can change it by setting BRKFL bits (UART_LINCTL[19:16]) and BSL bits (UART_LINCTL[21:20]), to change the dominant bits.

Note2: The break/sync delimiter length default setting is 1 bit time and the inter-byte space is 1 bit

time, software can change it by setting BSL bits (UART_LINCTL[21:20]) and DLY bit (UART_TOUT[15:8]).

Note3: If the header includes “break field, sync field and frame ID field”, software must fill frame ID in PID bits (UART_LINCTL[31:24]) before trigger header transmission (setting the SENDH bit (UART_LINCTL[8])). The frame ID parity can be generated by software or hardware depends on IDPEN (UART_LINCTL [9]). If the parity generated by software (IDPEN (UART_LINCTL[9]) = 0), software must fill 8 bit data (include 2 bit parity) in this field, and if the parity generated by hardware (IDPEN (UART_LINCTL[9]) = 1), software fill ID0~ID5, hardware will calculi P0 and P1.

Procedure with software error monitoring in Master mode:

1. Choose if the hardware transmission header field only includes “break field” by setting HSEL bits (UART_LINCTL[23:22]) = 0x00.
2. Enable break detection function by setting BRKDETEN bit (UART_LINCTL[10]).
3. Request break + delimiter transmission by setting the SENDH bit (UART_LINCTL[8]).
4. Wait for the BRKDETF flag (UART_LINSTS[8]).
5. Request sync field transmission by writing 0x55 into UART_DAT register.
6. Wait for the RDAIF flag (UART_INTSTS[0]) and read back the UART_DAT register.
7. Request header frame ID transmission by writing the protected identifier value in the UART_DAT register.
8. Wait for the RDAIF flag (UART_INTSTS[0]) register and read back the UART_DAT register.

- LIN break and delimiter detection

When software enable the break detection function by setting BRKDETEN bit (UART_LINCTL[10]), the break detection circuit is activated. The break detection circuit is totally independent from the UART receiver.

When the break detection function is enabled, the circuit looks at the input SIN pin for a start signal. If detect great than 11 bits are detected as 0, and are followed by a delimiter character, the BRKDETF flag (UART_LINSTS[8]) register at the end of break field. If the LINIEN bit (UART_INTEN[8]) = 1, an interrupt will be generated. The behavior of the break detection and break flag is shown in the following figure.

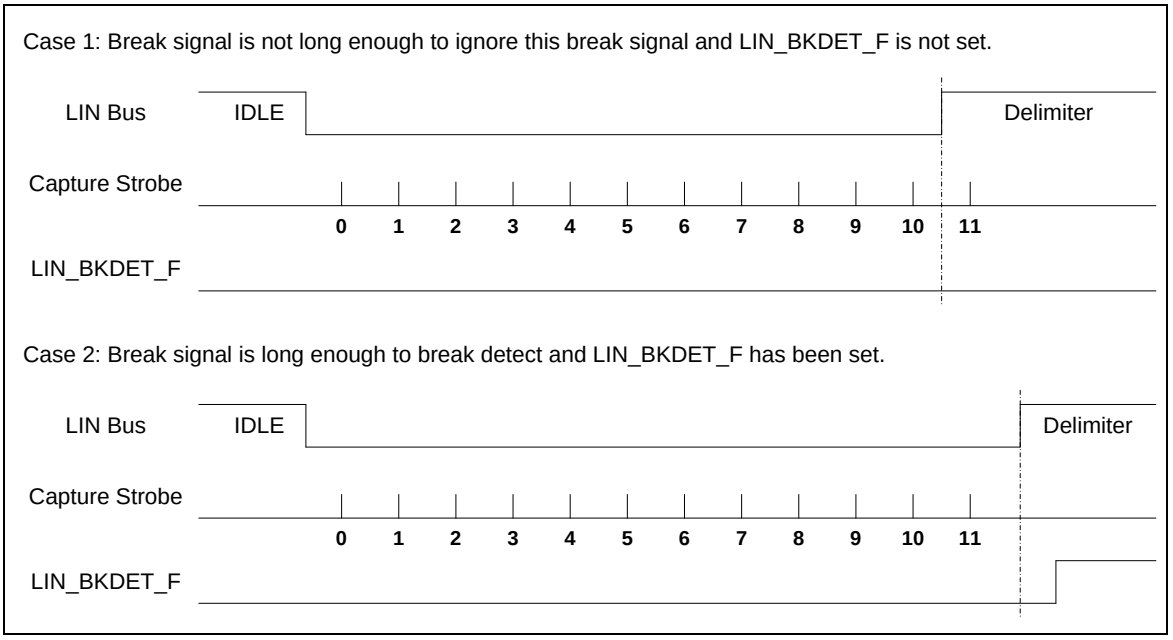


Figure 6.31-7 Break Detection in LIN Mode

The break detection circuit is totally independent from the UART receiver, but if UART controller is sending “break field”, at the same time the break detection circuit detected “break field”, the received data will not be dropped and the frame error will not set. However if a “break field” is detected while controller is not sending “break field”, the data will be stored in UART_DAT and the frame error will be set. The behavior of the break detection and frame error flag is shown in following Figure 6.31-8 Relationship between Break Detection and Frame Error Detection in Figure 6.31-8.

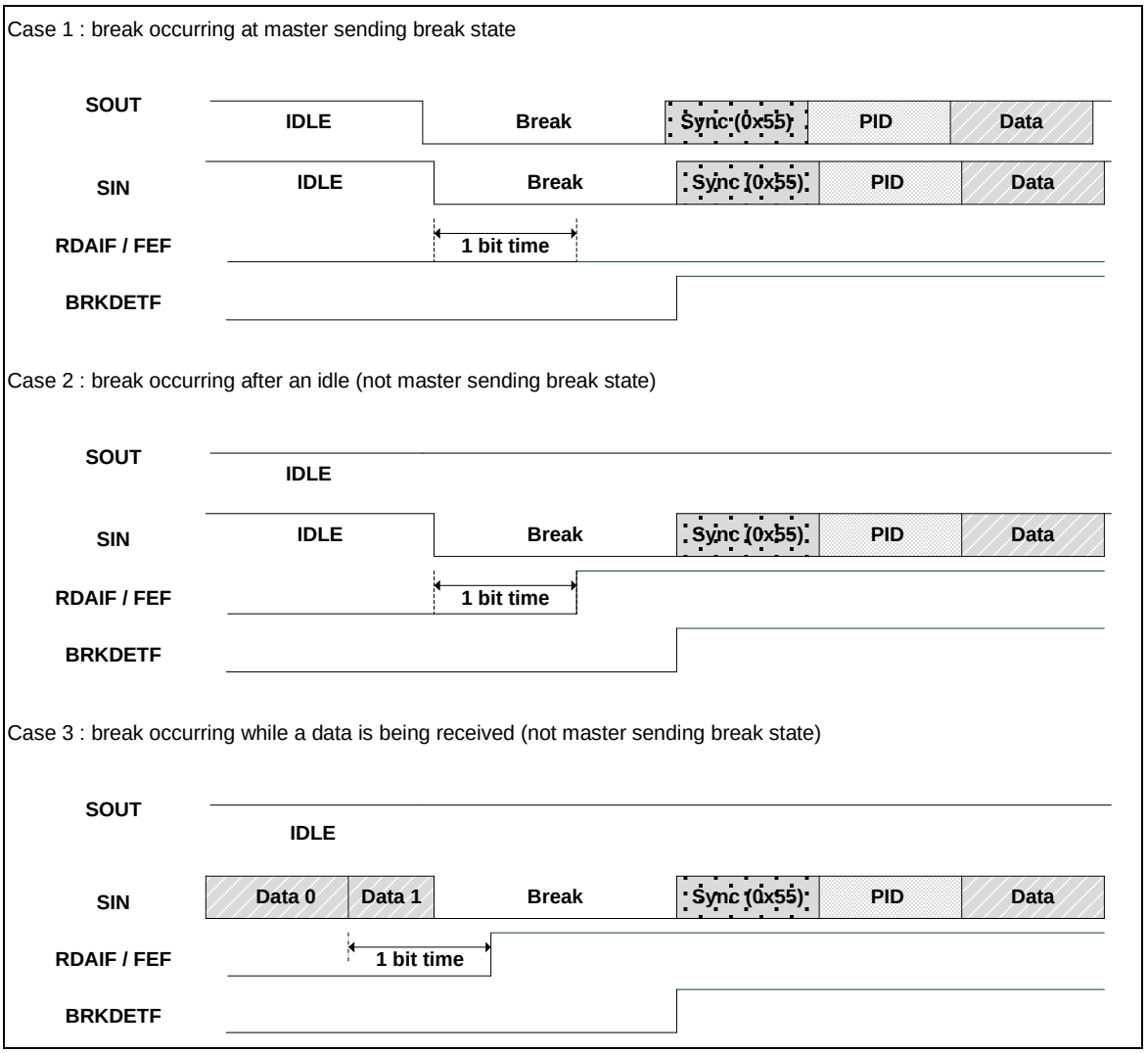


Figure 6.31-8 Relationship between Break Detection and Frame Error Detection

- LIN response transmission

  The LIN master can transmit response (master is the publisher of the response) and receive response (master is the subscriber of the response). When the master is the publisher of the response, the master send response by writing to UART_DAT register, and if the master is the subscriber of the response, the master will received n data bytes by other slave node.

6.31.5.3 LIN Slave Mode

The UART controller support LIN slave mode by setting the SLVEN bit (UART_LINCTL[0]). In LIN mode, each byte field is initialed by a start bit with value zero (dominant), followed by 8 data bits (WLS (UART_LINE[1:0]) = 2'b11) and no parity bit, LSB is first and ended by 1 stop bit (NSB (UART_LINE[2]) = 1) with value one (recessive) in accordance with the LIN standard. The initialization process flow of LIN slave mode is shown as follows:

1. Select the desired baud-rate by setting the UART_BAUD register.

2. Select LIN function mode by setting UART_FUNCSEL register.

3. Configure the data length to 8 bits by setting WLS bits (UART_LINE[1:0]) = 2'b11 and disable parity check by clearing PBE bit and configure the stop bit to 1 by clearing NSB bit (UART_LINE[2]).

4. Enable LIN slave mode by setting the SLVEN bit (UART_LINCTL[0]).

- LIN header reception

According to the LIN protocol, a slave node must wait for a valid header which coming from the master node. Then application has to take following action (depending on the master header frame ID value).

- Receive the response.
- Transmit the response.
- Ignore the response and wait for next header.

In LIN slave mode, user can enable slave header detection function by setting SLVHDEN bit (UART_LINCTL[1]) to detect complete frame header (receive "break field", "sync field" and "frame ID field"). When a LIN header is received, the SLVHDETF flag (UART_LINSTS[0]) will be set (If the LINIEN bit (UART_INTEN[8]) = 1, an interrupt will be generated). User can enable frame ID parity check function by setting IDPEN (UART_LINCTL[9]), and if the frame ID parity of received frame header is incorrect, the LIN_IDPERR_F flag (UART_LINSTS[2]) (If the LINIEN bit (UART_INTEN[8]) = 1, an interrupt will be generated) will be set together with the SLVHDETF flag (UART_LINSTS[0]). User also can put LIN in mute mode by setting MUTE bit (UART_LINCTL[4]). This mode allows detection of headers only (break + sync + frame ID) and prevents the reception of any other characters. In order to avoid bit rate tolerance, the controller support automatic resynchronization function to avoid clock deviation error, user can enable this feature by setting SLVAREN bit (UART_LINCTL[2]).

- LIN response transmission

The LIN slave can transmit response (slave is the publisher of the response) and receive response (slave is the subscriber of the response). When the slave is the publisher of the response, the slave send response by writing UART_DAT register, and if the slave is the subscriber of the response, the slave received n data bytes by other slave node.

Note: During LIN data transmission, software can monitor the LIN bus transfer state by hardware or software. User can enable hardware monitoring by setting BRKL bit (UART_LINCTL[12]), and when operating in LIN transmitter state, if the input pin (SIN) state is not equal to the output pin (SOUT) state that the hardware wills generator an interrupt to CPU. User also can monitor the LIN bus transfer state by check the read back data in UART_DAT register. The following sequence is a program sequence example:

- LIN header time-out error

The LIN slave controller contains a header time-out counter. If the entire header is not received within the maximum time limit of 57 bit times, the header error flag (SLVHEF (UART_LINSTS[1])) will be set. The time-out counter is enabled at each break detect edge and stopped in the following conditions.

- A LIN frame ID field has been received.
- The header error flag assert.
- Software write 1 to SLVSYNCF bit (UART_LINSTS[3]) to re-search new frame header.

- Mute mode and LIN wake-up

  In mute mode, it allows detection of headers only and prevents the reception of any other characters. User can enable mute mode by setting MUTE bit (UART_LINCTL[4]) and wake-up condition can be selected by HSEL (UART_LINCTL[23:22]).

Note: It is recommended to put LIN in mute mode by setting MUTE bit (UART_LINCTL[4]) after checksum transmission.

The LIN controller leaves mute mode in following conditions.

- While HSEL (UART_LINCTL[23:22]) is “break field”, when detect a valid LIN break + delimiter, the controller will enable the receiver (left mute mode) and the following data (sync data and frame ID data) will be stored in RX-FIFO.
- While HSEL (UART_LINCTL[23:22]) is “break field and sync field”, when detect a valid LIN break + delimiter and valid sync field without frame error, the controller will enable the receiver (left mute mode) and the following data (ID data) will be stored in RX-FIFO.
- While HSEL (UART_LINCTL[23:22]) is “break field, sync field and ID field”, when detect a valid LIN break + delimiter and valid sync field without frame error and the following ID data (without frame error) match PID (UART_LINCTL[31:24]), the controller will enable the receiver (left mute mode) and the following data will be stored in RX-FIFO.

Slave mode without automatic resynchronization

User can disable automatic resynchronization function to fix the communication baud rate. When operating in without automatic resynchronization mode, software needs some initial process, and the initialization process flow of without automatic resynchronization mode is shown as follows:

1. Select the desired baud-rate by setting the UART_BAUD register.
2. Select LIN function mode by setting UART_FUNCSEL register.
3. Enable LIN slave mode by setting the SLVEN bit (UART_LINCTL[0]).

Slave mode with automatic resynchronization

User can enable automatic resynchronization function by setting SLVAREN bit (UART_LINCTL[2]). In automatic resynchronization mode, the controller will adjust the baud rate generator after each sync field reception. The other program sequence is similar to Slave mode without automatic resynchronization section.

When automatic resynchronization function is enabled, after each LIN break field, the time duration between five falling edges is sampled on engine clock and the result of this measurement is stored in an internal 13-bit register and the UART_BAUD register value will automatically updated at the end of the fifth falling edge. If the measure timer (13bit) overflow before five falling edges, then the header error flag (SLVHEF (UART_LINSTS[1])) will be set.

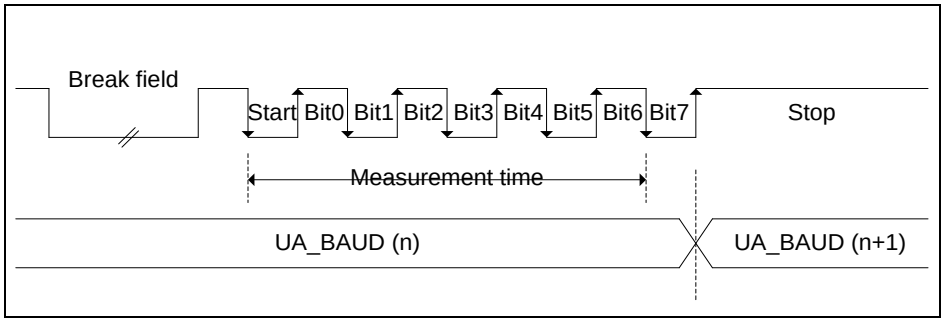


Figure 6.31-9 LIN Sync Field Measurement

When operating in automatic resynchronization mode, software must select the desired baud rate by setting the UART_BAUD register and hardware will store it at internal TEMP_REG register, after each LIN break field, the time duration between five falling edges is sampled on engine clock and the result of this measurement is stored in an internal 13-bit register (BAUD_LIN) and the result will be updated to UART_BAUD register automatically.

In order to guarantee the transmission baud rate, the baud rate generator must reload the initial value before each new break reception. The initial value is programmed by the application during initialization (TEMP_REG). User can setting SLVDUEN bit (UART_LINCTL[3]) to enable auto reload initial baud rate value function. If the SLVDUEN bit (UART_LINCTL[3]) is set, when received the next character, hardware will auto reload the initial value to UART_BAUD, and when the UART_BAUD be updated, the SLVDUEN bit (UART_LINCTL[3]) will be cleared automatically. The behavior of LIN updated method as shown in the following figure.

Note1: It is recommended to set the SLVDUEN bit (UART_LINCTL[3]) before every checksum reception.

Note2: When header error been detected, user must writing 1 to SLVSYNCF bit (UART_LINSTS[3]) to re-search new frame header. When user writing 1 to it, hardware will reload the initial baud-rate (TEMP_REG) and re-search new frame header.

Note3: When operating in automatic resynchronization mode, the baud rate setting must be mode2 (BAUDM1 (UART_BAUD[29])) and (BAUDM0 (UART_BAUD[28])) must be 1.

Case1: UART_BAUD read/write operation when SLVDUEN = 0

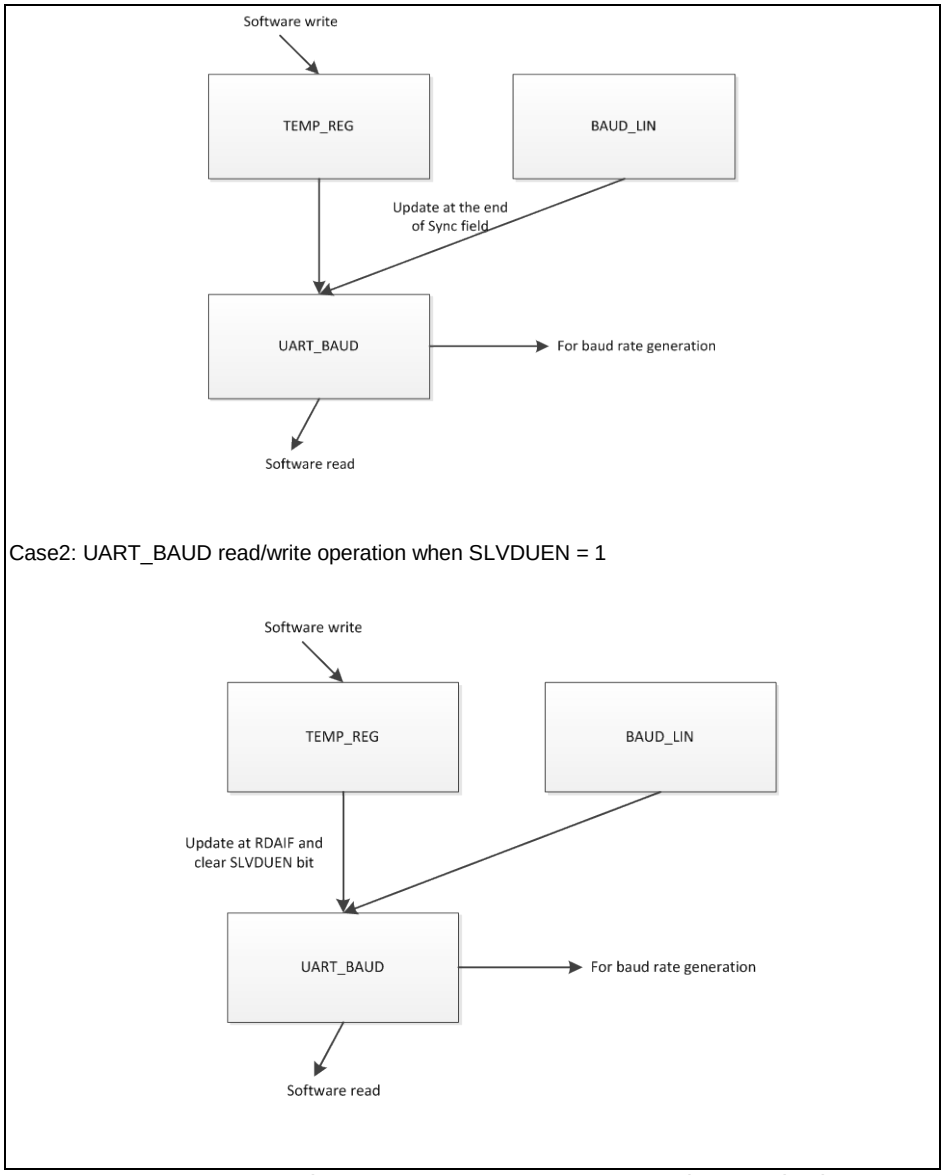


Figure 6.31-10 UART_BAUD Update Method

- Deviation error on the sync field

  When operating in automatic resynchronization mode, the controller will check the deviation error on the sync field. The deviation error is checked by comparing the current baud rate with the received sync field. Two checks are performed in parallel.

Check1: Based on measurement between the first falling edge and the last falling edge of the sync field.

- If the difference great than 15%, the header error flag (SLVHEF (UART_LINSTS[1])) will be set.
- If the difference between 15% and 14%, the header error flag (SLVHEF (UART_LINSTS[1])) may either set or not (depending on the data dephasing).

Check2: Based on measurement of time between each falling edge of the sync field.

- If the difference great than 19%, the header error flag (SLVHEF (UART_LINSTS[1])) will be set.
- If the difference between 19% and 15%, the header error flag (SLVHEF (UART_LINSTS[1])) may either set or not (depending on the data dephasing).

Note: The deviation check is based on the current baud-rate clock. Therefore, in order to guarantee correct deviation checking, the baud-rate must reload the nominal value before each new break reception by setting SLVDUEN bit (UART_LINCTL[3]) (It is recommend setting the SLVDUEN bit (UART_LINCTL[3]) before every checksum reception).

- LIN header error detection

  In LIN slave function mode, when user enables header detection function by setting SLVHDEN bit (UART_LINCTL[1]), the hardware will handle the header detect flow. If the header has error, the LIN header error flag (SLVHEF (UART_LINSTS[1])) will be set and an interrupt is generated if the LINIEN bit (UART_INTEN[8]) is set. When header error been detect, user must to reset the detect circuit to re-search new frame header by writing 1 to SLVSYNCF bit(UART_LINSTS[3]) to re-search new frame header.

  The LIN header error flag (SLVHEF (UART_LINSTS[1])) is set if one of the following conditions occurs:

  - Break Delimiter is too short (less 0.5 bit time).
  - Frame error in sync field or Identifier field.
  - The sync field data is not 0x55 (without automatic resynchronization mode).
  - The sync field deviation error (with automatic resynchronization mode).
  - The sync field measure time-out (with automatic resynchronization mode).
  - LIN header reception time-out.

6.31.6 RS-485 Function Mode

The UART supports **RS-485 9-bit mode function**. The RS-485 mode is selected by setting the UART_FUNCSEL register to select RS-485 function. The RS-485 driver control is implemented using the RTS control signal from an asynchronous serial port to enable the RS-485 driver. In RS-485 mode, many characteristics of the RX and TX are same as UART.

In RS-485 mode, the controller can configuration of it as an RS-485 addressable slave and the RS-485 master transmitter will identify an address character by setting the parity (9^{th} bit) to 1. For data characters, the parity is set to 0. Software can use UART_LINE register to control the 9-th bit (When the PBE (UART_LINE[3]), EPE (UART_LINE[4]) and SPE (UART_LINE[5]) are set, the 9-th bit is transmitted 0 and when PBE (UART_LINE[3]) and SPE (UART_LINE[5]) are set and EPE (UART_LINE[4]) is cleared, the 9-th bit is transmitted 1). The Controller support three operation mode that is RS-485 Normal Multidrop Operation Mode (NMM), RS-485 Auto Address Detection Operation Mode (AAD) and RS-485 Auto Direction Control Operation Mode (AUD), software can choose any operation mode by programming UA_RS-485_CSR register, and software can driving the transfer delay time between the last stop bit leaving the TX-FIFO and the de-assertion of by setting DLY bits (UART_TOUT [15:8]).

RS-485 Normal Multidrop Operation Mode (NMM)

In RS-485 Normal Multidrop operation mode, in first, software must decided the data which before the address byte be detected will be stored in RX-FIFO or not. If software want to ignore any data before address byte detected, the flow is set RXOFF bit (UART_FCR[8]) then enable RS485NMM bit (UART_ALTCTL[8]) and the receiver will ignore any data until an address byte is detected (bit9 =1) and the address byte data will be stored in the RX-FIFO. If software wants to receive any data before address byte detected, the flow is disable RXOFF bit (UART_FCR[8]) then enable RS485NMM bit (UART_ALTCTL[8]) and the receiver will received any data. If an address byte is detected (bit9 =1), it will generator an interrupt to CPU and software can decide whether enable or disable receiver to accept the following data byte by setting RXOFF bit (UART_FCR[8]). If the receiver is be enabled, all received byte data will be accepted and stored in the RX-FIFO, and if the receiver is disabled, all received byte data will be ignore until the next address byte be detected. If software disable receiver by setting RXOFF bit (UART_FCR[8]), when a next address byte be detected, the controller will clear the RXOFF bit (UART_FCR[8]) and the address byte data will be stored in the RX-FIFO.

RS-485 Auto Address Detection Operation Mode (AAD)

In RS-485 Auto Address Detection Operation Mode, the receiver will ignore any data until an address byte is detected (bit9 =1) and the address byte data match the ADDRMV (UART_ALTCTL[31:24]) value. The address byte data will be stored in the RX-FIFO. The all received byte data will be accepted and stored in the RX-FIFO until and address byte data not match the ADDRMV (UART_ALTCTL[31:24]) value.

RS-485 Auto Direction Mode (AUD)

Another option function of RS-485 controllers is **RS-485 auto direction control function**. The RS-485 driver control is implemented using the RTS control signal from an asynchronous serial port to enable the RS-485 driver. The RTS line is connected to the RS-485 driver enable such that setting the RTS line to high (logic 1) enables the RS-485 driver. Setting the RTS line to low (logic 0) puts the driver into the tri-state condition. User can setting RTSACTLV bit (UART_MODEM[9]) to change the RTS driving level.

Program Sequence example:

1. Program FUNCSEL in UART_FUNCSEL to select RS-485 function.
2. Program the RXOFF bit (UART_FIFO[8]) to determine enable or disable RS-485 receiver
3. Program the RS-485_NMM or RS-485_AAD mode.
4. If the RS-485_AAD mode is selected, the ADDRMV (UART_ALTCTL[31:24]) is programmed for auto address match value.
5. Determine auto direction control by programming RS-485_AUD.

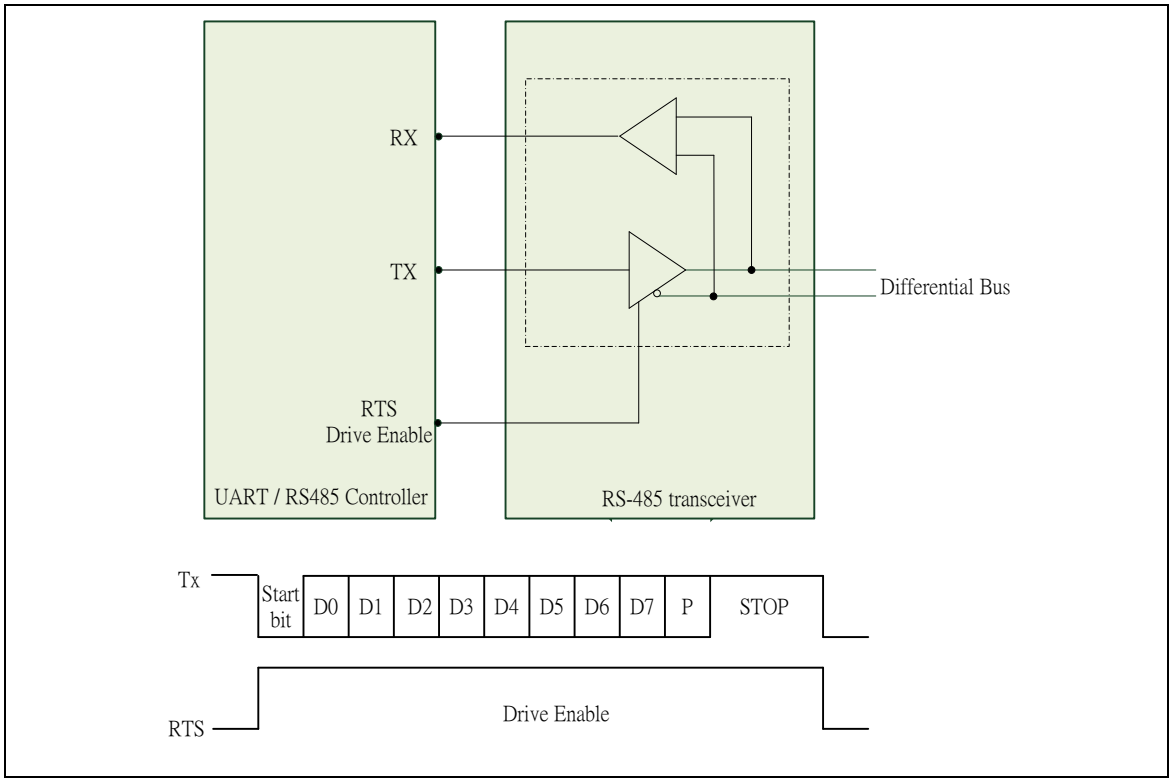


Figure 6.31-11 RS-485 Frame Structure

6.31.7 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
UART Base Address: **UARTx_BA = 0x4007_0000 + 0x1000 * x** **x=0,1..5**				
UART_DAT	UARTx_BA+0x00	R/W	UARTx Receive / Transmit Buffer Register	Undefined
UART_INTEN	UARTx_BA+0x04	R/W	UARTx Interrupt Enable Register	0x0000_0000
UART_FIFO	UARTx_BA+0x08	R/W	UARTx FIFO Control Register	0x0000_0000
UART_LINE	UARTx_BA+0x0C	R/W	UARTx Line Control Register	0x0000_0000
UART_MODEM	UARTx_BA+0x10	R/W	UARTx Modem Control Register	0x0000_0000
UART_MODEMSTS	UARTx_BA+0x14	R/W	UARTx Modem Status Register	0x0000_0000
UART_FIFOSTS	UARTx_BA+0x18	R/W	UARTx FIFO Status Register	0x1040_4000
UART_INTSTS	UARTx_BA+0x1C	R/W	UARTx Interrupt Status Register	0x0000_0002
UART_TOUT	UARTx_BA+0x20	R/W	UARTx Time-out Register	0x0000_0000
UART_BAUD	UARTx_BA+0x24	R/W	UARTx Baud Rate Divisor Register	0x0F00_0000
UART_IRDA	UARTx_BA+0x28	R/W	UARTx IrDA Control Register	0x0000_0040
UART_ALTCTL	UARTx_BA+0x2C	R/W	UARTx Alternate Control/Status Register	0x0000_000C
UART_FUNCSEL	UARTx_BA+0x30	R/W	UARTx Function Select Register	0x0000_0000
UART_LINCTL	UARTx_BA+0x34	R/W	UARTx LIN Control Register	0x000C_0000
UART_LINSTS	UARTx_BA+0x38	R/W	UARTx LIN Status Register	0x0000_0000
UART_LINDEBUG	UARTx_BA+0x3C	R/W	UARTx LIN Debug Register	0x0000_0000
UART_SCCTL	UARTx_BA+0x40	R/W	UARTx SC Control Register	0x0000_0000
UART_SCSTATUS	UARTx_BA+0x44	R/W	UARTx SC Flag Status Register	0x0000_0000

Note: Where the x of “UARTx_BA” is 0~5.

6.31.8 Register Description

UARTx Receive / Transmit Buffer Register (UART_DAT)

Register	Offset	R/W	Description	Reset Value
UART_DAT	UARTx_BA+0x00	R/W	UARTx Receive / Transmit Buffer Register	Undefined

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DAT							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**DAT**	**Receiving/Transmit Buffer** Write Operation: By writing one byte to this register, the data byte will be stored in transmitter FIFO. The UART Controller will send out the data stored in transmitter FIFO top location through the UART_TXD. Read Operation: By reading this register, the UART will return an 8-bit data received from receiving FIFO

Interrupt Enable Register (UART_INTEN)

Register	Offset	R/W	Description	Reset Value
UART_INTEN	UARTx_BA+0x04	R/W	UARTx Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
RXPDMAEN	TXPDMAEN	ATOCTSEN	ATORTSEN	TOCNTEN	Reserved		LINIEN
7	6	5	4	3	2	1	0
Reserved	WKCTSIEN	BUFERRIEN	RXTOIEN	MODEMIEN	RLSIEN	THREIEN	RDAIEN

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15]	**RXPDMAEN**	**RX DMA Enable Bit** This bit can enable or disable RX DMA service. 0 = RX DMA Disabled. 1 = RX DMA Enabled.
[14]	**TXPDMAEN**	**TX DMA Enable Bit** This bit can enable or disable TX DMA service. 0 = TX DMA Disabled. 1 = TX DMA Enabled.
[13]	**ATOCTSEN**	**CTS Auto Flow Control Enable Bit** 0 = CTS auto flow control Disabled. 1 = CTS auto flow control Enabled. When CTS auto-flow is enabled, the UART will send data to external device when CTS input assert (UART will not send data to device until CTS is asserted).
[12]	**ATORTSEN**	**RTS Auto Flow Control Enable Bit** 0 = RTS auto flow control Disabled. 1 = RTS auto flow control Enabled. When RTS auto-flow is enabled, if the number of bytes in the RX FIFO equals the RTSTRGLV (UART_FIFO[19:16]), the UART will de-assert RTS signal.
[11]	**TOCNTEN**	**Time-Out Counter Enable Bit** 0 = Time-out counter Disabled. 1 = Time-out counter Enabled.
[10:9]	**Reserved**	Reserved.
[8]	**LINIEN**	**LIN RX Break Field Detected Interrupt Enable Bit**

		0 = Lin bus RX break filed interrupt Disabled. 1 = Lin bus RX break filed interrupt Enabled. **Note:** This field is used for LIN function mode.
[7]	**Reserved**	Reserved.
[6]	**WKCTSIEN**	**UART Wake-Up Function Enable Bit** 0 = UART wake-up function Disabled. 1 = UART wake-up function Enabled when the chip is in Power-down mode, an external CTS change will wake up chip from Power-down mode.
[5]	**BUFERRIEN**	**Buffer Error Interrupt Enable Bit** 0 = INT_BUF_ERR Disabled. 1 = INT_BUF_ERR Enabled.
[4]	**RXTOIEN**	**RX Time-Out Interrupt Enable Bit** 0 = NT_TOUT Disabled. 1 = INT_TOUT Enabled.
[3]	**MODEMIEN**	**Modem Status Interrupt Enable Bit** 0 = INT_MODEM Disabled. 1 = INT_MODEM Enabled.
[2]	**RLSIEN**	**Receive Line Status Interrupt Enable Bit** 0 = INT_RLS Disabled. 1 = INT_RLS Enabled.
[1]	**THREIEN**	**Transmit Holding Register Empty Interrupt Enable Bit** 0 = INT_THRE Disabled. 1 = INT_THRE Enabled.
[0]	**RDAIEN**	**Receive Data Available Interrupt Enable Bit** 0 = INT_RDA Disabled. 1 = INT_RDA Enabled.

FIFO Control Register (UART_FIFO)

Register	Offset	R/W	Description	Reset Value
UART_FIFO	UARTx_BA+0x08	R/W	UARTx FIFO Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				RTSTRGLV			
15	14	13	12	11	10	9	8
Reserved							RXOFF
7	6	5	4	3	2	1	0
RFITL				Reserved	TXRST	RXRST	Reserved

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19:16]	**RTSTRGLV**	**RTS Trigger Level For Auto-Flow Control Use** RTSTRGLV / Trigger Level (Bytes) 0000 / 01 0001 / 04 0010 / 08 0011 / 14 0100 / 30/14 (High-speed/Normal Speed) 0101 / 46/14 (High-speed/Normal Speed) 0110 / 62/14 (High-speed/Normal Speed) others / 62/14 (High-speed/Normal Speed) **Note:** This field is used for auto RTS flow control.
[15:9]	**Reserved**	Reserved.
[8]	**RXOFF**	**Receiver Disable** The receiver is disabled or not. 0 = Receiver Enabled. 1 = Receiver Disabled. **Note:** This field is used for RS-485 Normal Multi-drop mode. It should be programmed before RS-485_NMM (UART_ALTCTL [8]) is programmed.
[7:4]	**RFITL**	**RX FIFO Interrupt (INT_RDA) Trigger Level** When the number of bytes in the receive FIFO equals the RFITL, the RDAIF will be set (if RDAIEN (UART_INTEN [0]) is enabled, an interrupt will generated). RFITL / INTR_RDA Trigger Level (Bytes)

		0000 01 0001 04 0010 08 0011 14 0100 30/14 (High-speed/Normal Speed) 0101 46/14 (High-speed/Normal Speed) 0110 62/14 (High-speed/Normal Speed) others 62/14 (High-speed/Normal Speed)
[3]	**Reserved**	Reserved.
[2]	**TXRST**	**TX Field Software Reset** When TX_RST is set, all the byte in the transmit FIFO and TX internal state machine are cleared. 0 = No effect. 1 = Reset the TX internal state machine and pointers. **Note:** This bit will auto clear needs at least 3 UART engine clock cycles.
[1]	**RXRST**	**RX Field Software Reset** When RX_RST is set, all the byte in the receiver FIFO and RX internal state machine are cleared. 0 = No effect. 1 = Reset the RX internal state machine and pointers. **Note:** This bit will be automatically cleared for at least 3 UART engine clock cycles.
[0]	**Reserved**	Reserved.

Line Control Register (UART_LINE)

Register	Offset	R/W	Description	Reset Value
UART_LINE	UARTx_BA+0x0C	R/W	UARTx Line Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	BCB	SPE	EPE	PBE	NSB	WLS	

Bits	Description	
[31:7]	**Reserved**	Reserved.
[6]	**BCB**	**Break Control** When this bit is set to logic 1, the serial data output (TX) is forced to the Spacing State (logic 0). This bit acts only on TX and has no effect on the transmitter logic.
[5]	**SPE**	**Stick Parity Enable Bit** 0 = Stick parity Disabled. 1 = If bit 3 and 4 are logic 1, the parity bit is transmitted and checked as logic 0. If bit 3 is 1 and bit 4 is 0 then the parity bit is transmitted and checked as 1.
[4]	**EPE**	**Even Parity Enable Bit** 0 = Odd number of logic 1's is transmitted and checked in each word. 1 = Even number of logic 1's is transmitted and checked in each word. This bit is effective only when bit 3 (parity bit enable) is set.
[3]	**PBE**	**Parity Bit Enable Bit** 0 = No parity bit. 1 = Parity bit is generated on each outgoing character and is checked on each incoming data.
[2]	**NSB**	**Number Of "STOP Bit"** 0= One " STOP bit" is generated in the transmitted data. 1= One and a half " STOP bit" is generated in the transmitted data when 5-bit word length is selected. Two "STOP bit" is generated when 6-, 7- and 8-bit word length is selected.
[1:0]	**WLS**	**Word Length Selection** **WLS[1:0]** / **Character Length** 00 / 5-bit 01 / 6-bit

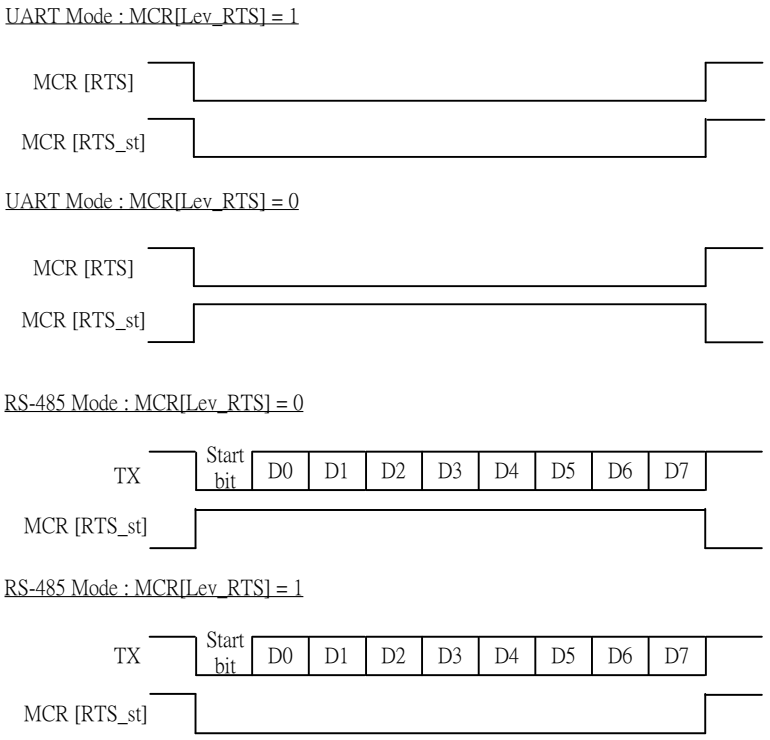
		10	7-bit	
		11	8-bit	

MODEM Control Register (UART_MODEM)

Register	Offset	R/W	Description	Reset Value
UART_MODEM	UARTx_BA+0x10	R/W	UARTx Modem Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**RTSSTS**	**Reserved**			**RTSACTLV**	**Reserved**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**RTS**	**Reserved**

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13]	**RTSSTS**	**RTS Pin State (Read Only)** This bit is the output pin status of RTS.
[12:10]	**Reserved**	Reserved.

[9]	**RTSACTLV**	**RTS Trigger Level** This bit can change the RTS trigger level. 0= Low level triggered. 1= High level triggered. 
[8:2]	**Reserved**	Reserved.
[1]	**RTS**	**RTS (Request-To-Send) Signal** 0 = Drive RTS pin to logic 1 (If the **RTSACTLV** set to low level triggered). 1 = Drive RTS pin to logic 0 (If the **RTSACTLV** set to low level triggered). 0 = Drive RTS pin to logic 0 (If the **RTSACTLV** set to high level triggered). 1 = Drive RTS pin to logic 1 (If the **RTSACTLV** set to high level triggered).
[0]	**Reserved**	Reserved.

Modem Status Register (UART_MODEMSTS)

Register	Offset	R/W	Description	Reset Value
UART_MODEMSTS	UARTx_BA+0x14	R/W	UARTx Modem Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							CTSACTLV
7	6	5	4	3	2	1	0
Reserved			CTSSTS	Reserved			CTSDETF

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**CTSACTLV**	**CTS Trigger Level** This bit can change the CTS trigger level. 0= Low level triggered. 1= High level triggered.
[7:5]	**Reserved**	Reserved.
[4]	**CTSSTS**	**CTS Pin Status (Read Only)** This bit is the pin status of CTS.
[3:1]	**Reserved**	Reserved.
[0]	**CTSDETF**	**Detect CTS State Change Flag (Read Only)** This bit is set whenever CTS input has change state, and it will generate Modem interrupt to CPU when MODEMIEN (UART_INTEN[3]) is set to 1. Software can write 1 to clear this bit to 0

FIFO Status Register (UART_FIFOSTS)

Register	Offset	R/W	Description	Reset Value
UART_FIFOSTS	UARTx_BA+0x18	R/W	UARTx FIFO Status Register	0x1040_4000

31	30	29	28	27	26	25	24
Reserved			TXEMPTYF	Reserved			TXOVIF
23	22	21	20	19	18	17	16
TXFULL	TXEMPTY	TXPTR					
15	14	13	12	11	10	9	8
RXFULL	RXEMPTY	RXPTR					
7	6	5	4	3	2	1	0
Reserved	BIF	FEF	PEF	ADDRDETF	SCERR	Reserved	RXOVIF

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28]	**TXEMPTYF**	**Transmitter Empty Flag (Read Only)** Bit is set by hardware when TX FIFO (UART_DAT) is empty and the STOP bit of the last byte has been transmitted. Bit is cleared automatically when TX FIFO is not empty or the last byte transmission has not completed.
[27:25]	**Reserved**	Reserved.
[24]	**TXOVIF**	**TX Overflow Error Interrupt Flag (Read Only)** If TX FIFO (UART_DAT) is full, an additional write to UART_DAT will cause this bit to logic 1. **Note:** This bit is read only, but it can be cleared by writing '1' to it.
[23]	**TXFULL**	**Transmitter FIFO Full (Read Only)** This bit indicates TX FIFO full or not. This bit is set when TXPTR is equal to 64/16(UART0/UART1~5), otherwise is cleared by hardware.
[22]	**TXEMPTY**	**Transmitter FIFO Empty (Read Only)** This bit indicates TX FIFO empty or not. When the last byte of TX FIFO has been transferred to Transmitter Shift Register, hardware sets this bit high. It will be cleared when writing data into DAT (TX FIFO not empty).
[21:16]	**TXPTR**	**TX FIFO Pointer (Read Only)** This field indicates the TX FIFO Buffer Pointer. When CPU writes one byte into UART_DAT, TXPTR increases one. When one byte of TX FIFO is transferred to Transmitter Shift Register, TXPTR decreases one.
[15]	**RXFULL**	**Receiver FIFO Full (Read Only)** This bit initiates RX FIFO full or not. This bit is set when RXPTR is equal to 64/16(UART0/UART1~5), otherwise is cleared by

		hardware.
[14]	**RXEMPTY**	**Receiver FIFO Empty (Read Only)** This bit initiate RX FIFO empty or not. When the last byte of RX FIFO has been read by CPU, hardware sets this bit high. It will be cleared when UART receives any new data.
[13:8]	**RXPTR**	**RX FIFO Pointer (Read Only)** This field indicates the RX FIFO Buffer Pointer. When UART receives one byte from external device, RXPTR increases one. When one byte of RX FIFO is read by CPU, RXPTR decreases one.
[7]	**Reserved**	Reserved.
[6]	**BIF**	**Break Interrupt Flag (Read Only)** This bit is set to a logic 1 whenever the received data input(RX) is held in the “spacing state” (logic 0) for longer than a full word transmission time (that is, the total time of “start bit” + data bits + parity + stop bits) and is reset whenever the CPU writes 1 to this bit. **Note:** This bit is read only, but it can be cleared by writing ‘1’ to it.
[5]	**FEF**	**Framing Error Flag (Read Only)** This bit is set to logic 1 whenever the received character does not have a valid “stop bit” (that is, the stop bit following the last data bit or parity bit is detected as a logic 0), and is reset whenever the CPU writes 1 to this bit. **Note:** This bit is read only, but it can be cleared by writing ‘1’ to it.
[4]	**PEF**	**Parity Error Flag (Read Only)** This bit is set to logic 1 whenever the received character does not have a valid “parity bit”, and is reset whenever the CPU writes 1 to this bit. **Note:** This bit is read only, but it can be cleared by writing ‘1’ to it.
[3]	**ADDRDETF**	**RS-485 Address Byte Detection Flag (Read Only)** This bit is set to logic 1 and set RS-485_ADD_EN (UART_ALTCTL[15]) whenever in RS-485 mode the receiver detect any address byte received address byte character (bit9 = ‘1’) bit”, and it is reset whenever the CPU writes 1 to this bit. **Note1:** This field is used for RS-485 function mode. **Note2:** This bit is read only, but it can be cleared by writing ‘1’ to it.
[2]	**SCERR**	**Smart Card Over Error Retry Flag** It is set to 1 when transmitter re-transmits over the retry number (TXRTY (UART_SCCTL[6:4])) or the receiver transfer error retry over retry number (RXRTY (UART_SCCTL[2:0])) 0 = No any transmitter re-transmits over or receiver transfer error retry over. 1 = one of the transmitter re-transmits over active or receiver transfer error retry over active. **Note1:** This field is used for SC function mode. **Note2:** This bit is read only, but it can be cleared by writing ‘1’ to it.
[1]	**Reserved**	Reserved.
[0]	**RXOVIF**	**RX Overflow Error IF (Read Only)** This bit is set when RX FIFO overflow. If the number of bytes of received data is greater than RX_FIFO (UART_DAT) size, 64/16 bytes of UART0/UART1, this bit will be set. **Note:** This bit is read only, but it can be cleared by writing ‘1’ to it.

Interrupt Status Control Register (UART_INTSTS)

Register	Offset	R/W	Description	Reset Value
UART_INTSTS	UARTx_BA+0x1C	R/W	UARTx Interrupt Status Register	0x0000_0002

31	30	29	28	27	26	25	24
Reserved		HWBUFEINT	HWTOINT	HWMODINT	HWRLSINT	Reserved	
23	22	21	20	19	18	17	16
Reserved		HWBUFEIF	HWTOIF	HWMODIF	HWRLSIF	Reserved	
15	14	13	12	11	10	9	8
LININT	Reserved	BUFERRINT	RXTOINT	MODEMINT	RLSINT	THREINT	RDAINT
7	6	5	4	3	2	1	0
LINIF	Reserved	BUFERRIF	RXTOIF	MODENIF	RLSIF	THREIF	RDAIF

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29]	**HWBUFEINT**	**In DMA Mode, Buffer Error Interrupt Indicator (Read Only)** This bit is set if BUFERRIEN and HWBFERIF are both set to 1. 0 = No buffer error interrupt is generated in DMA mode. 1 = The buffer error interrupt is generated in DMA mode.
[28]	**HWTOINT**	**In DMA Mode, Time-Out Interrupt Indicator (Read Only)** This bit is set if RXTOIEN and HWTOIF are both set to 1. 0 = No Tout interrupt is generated in DMA mode. 1 = Tout interrupt is generated in DMA mode.
[27]	**HWMODINT**	**In DMA Mode, MODEM Status Interrupt Indicator (Read Only)** This bit is set if MODEMIEN and HWMODIF are both set to 1. 0 = No Modem interrupt is generated in DMA mode. 1 = Modem interrupt is generated in DMA mode.
[26]	**HWRLSINT**	**In DMA Mode, Receive Line Status Interrupt Indicator (Read Only)** This bit is set if RLSIEN and HWRLSIF are both set to 1. 0 = No RLS interrupt is generated in DMA mode. 1 = RLS interrupt is generated in DMA mode.
[25:22]	**Reserved**	Reserved.
[21]	**HWBUFEIF**	**In DMA Mode, Buffer Error Interrupt Flag (Read Only)** This bit is set when the TX or RX FIFO overflows (TXOVIF or RXOVIF is set). When BERRIF is set, the transfer maybe is not correct. If BUFERRIEN bit (UART_INTEN [5]) is enabled, the buffer error interrupt will be generated. **Note:** This bit is cleared when both TXOVIF and RXOVIF are cleared.
[20]	**HWTOIF**	**In DMA Mode, Time-Out Interrupt Flag (Read Only)** This bit is set when the RX FIFO is not empty and no activities occurred in the RX FIFO and the time-out counter equal to TOIC. If TOCNTEN (UART_INTEN[11]) is enabled, the

		Time-out interrupt will be generated. **Note:** This bit is read only and user can read UART_DAT (RX is in active) to clear it.
[19]	**HWMODIF**	**In DMA Mode, MODEM Interrupt Flag (Read Only)** This bit is set when the CTS pin has state change (CTSDETF = 1). If MODEMIEN (UART_INTEN[3]) is enabled, the Modem interrupt will be generated. **Note:** This bit is read only and reset to 0 when bit CTSDETF is cleared by a write 1 on CTSDETF.
[18]	**HWRLSIF**	**In DMA Mode, Receive Line Status Flag (Read Only)** This bit is set when the RX receive data have parity error, framing error or break error (at least one of 3 bits, BIF, FEF and PEF, is set). If RLSIEN bit (UART_INTEN[2]) is enabled, the RLS interrupt will be generated. **Note1:** In RS-485 function mode, this field includes “receiver detect any address byte received address byte character (bit9 = ‘1’) bit". **Note2:** In SC function mode, this field includes “error retry over flag ". **Note3:** This bit is read only and reset to 0 when all bits of BIF, FEF and PEF are cleared.
[17:16]	**Reserved**	Reserved.
[15]	**LININT**	**LIN Bus Interrupt Indicator (Read Only)** This bit is set if LINIEN and LINIF are both set to 1. 0 = No LIN RX Break interrupt is generated. 1 = LIN RX Break interrupt is generated.
[14]	**Reserved**	Reserved.
[13]	**BUFERRINT**	**Buffer Error Interrupt Indicator (Read Only)** This bit is set if BUFERRIEN and BERRIF are both set to 1. 0 = No buffer error interrupt is generated. 1 = The buffer error interrupt is generated.
[12]	**RXTOINT**	**Time-Out Interrupt Indicator (Read Only)** This bit is set if TOCNTEN and RXTOIF are both set to 1. 0 = No Tout interrupt is generated. 1 = Tout interrupt is generated.
[11]	**MODEMINT**	**MODEM Status Interrupt Indicator (Read Only)** This bit is set if MODEMIEN and MODENIF are both set to 1. 0 = No Modem interrupt is generated. 1 = Modem interrupt is generated.
[10]	**RLSINT**	**Receive Line Status Interrupt Indicator (Read Only)** This bit is set if RLSIEN and RLSIF are both set to 1. 0 = No RLS interrupt is generated. 1 = RLS interrupt is generated.
[9]	**THREINT**	**Transmit Holding Register Empty Interrupt Indicator (Read Only)** This bit is set if THREIEN and THREIF are both set to 1. 0 = No THRE interrupt is generated. 1 = THRE interrupt is generated.
[8]	**RDAINT**	**Receive Data Available Interrupt Indicator (Read Only)** This bit is set if RDAIEN and RDAIF are both set to 1. 0 = No RDA interrupt is generated. 1 = RDA interrupt is generated.

[7]	**LINIF**	**LIN Bus Flag (Read Only)** This bit is set when LIN slave header detect (SLVHDETF=1), LIN break detect (BRKDETF=1), bit error detect (BITEF=1), LIN slave ID parity error (SLVIDPEF) or LIN slave header error detect (SLVHEF) If LINIEN bit (UART_INTEN[8]) is enabled the LIN interrupt will be generated. **Note:** This bit is cleared when both SLVHDETF and LIN_BRDER_F and BITEF and LINS_IDPENR_F and SLVHEF are cleared
[6]	**Reserved**	Reserved.
[5]	**BUFERRIF**	**Buffer Error Interrupt Flag (Read Only)** This bit is set when the TX or RX FIFO overflows (TXOVIF or RXOVIF is set). When BERRIF is set, the transfer maybe is not correct. If BUFERRIEN bit (UART_INTEN[5]) is enabled, the buffer error interrupt will be generated. **Note:** This bit is cleared when both TXOVIF and RXOVIF are cleared.
[4]	**RXTOIF**	**Time-Out Interrupt Flag (Read Only)** This bit is set when the RX FIFO is not empty and no activities occurred in the RX FIFO and the time-out counter equal to TOIC. If TOCNTEN bit (UART_INTEN[11]) is enabled, the Time-out interrupt will be generated. **Note:** This bit is read only and user can read UART_DAT (RX is in active) to clear it.
[3]	**MODENIF**	**MODEM Interrupt Flag (Read Only)** This bit is set when the CTS pin has state change (CTSDETF=1). If MODEMIEN bit (UART_INTEN[3]) is enabled, the Modem interrupt will be generated. **Note:** This bit is read only and reset to 0 when bit CTSDETF is cleared by a write 1 on CTSDETF.
[2]	**RLSIF**	**Receive Line Interrupt Flag (Read Only)** This bit is set when the RX receive data have parity error, framing error or break error (at least one of 3 bits, BIF, FEF and PEF, is set). If RLSIEN bit (UART_INTEN[2]) is enabled, the RLS interrupt will be generated. **Note1:** In RS-485 function mode, this field include “receiver detect any address byte received address byte character (bit9 = ‘1’) bit". **Note2:** In SC function mode, this field includes “error retry over flag ". **Note3:** This bit is read only and reset to 0 when all bits of BIF, FEF and PEF are cleared.
[1]	**THREIF**	**Transmit Holding Register Empty Interrupt Flag (Read Only)** This bit is set when the last data of TX FIFO is transferred to Transmitter Shift Register. If THREIEN bit (UART_INTEN[1]) is enabled, the THRE interrupt will be generated. **Note:** This bit is read only and it will be cleared when writing data into DAT (TX FIFO not empty).
[0]	**RDAIF**	**Receive Data Available Interrupt Flag (Read Only)** When the number of bytes in the RX FIFO equals the RFITL then the RDAIF will be set. If RDAIEN bit (UART_INTEN[0]) is enabled, the RDA interrupt will be generated. **Note:** This bit is read only and it will be cleared when the number of unread bytes of RX FIFO drops below the threshold level (RFITL).

UART Interrupt Source	Interrupt Enable Bit	Interrupt Indicator To Interrupt Controller	Interrupt Flag	Flag Cleared By
Buffer Error Interrupt INT_BUF_ERR	BUFERRIEN	HWBEINT	**HWBUFEIF** = (TXOVIF or RXOVIF)	Write ‘1’ to TXOVIF/ RXOVIF
RX Time-out Interrupt INT_TOUT	RXTOIEN	HWTOINT	**HWTOIF**	Read UART_DAT
Modem Status Interrupt	MODEMIEN	HWMODINT	**HWMODIF** =	Write ‘1’ to CTSDETF

INT_MODEM			(CTSDETF)	
Receive Line Status Interrupt INT_RLS	RLSIEN	HWRLSINT	**HWRLSIF** = (BIF or FEF or PEF or RS-485_ADD_DETFor SCERR)	Write '1' to BIF/FEF/PEF/ RS-485_ADD_DETF/ TXOVERR/ RXOVERR
Transmit Holding Register Empty Interrupt INT_THRE	THREIEN	HW_THRE_INT	**HW_THRE_IF**	Write UART_DAT
Receive Data Available Interrupt INT_RDA	RDAIEN	HW_RDA_INT	**HW_RDA_IF**	Read UART_DAT

Table 6.31-1 UART Interrupt Sources and Flag List in DMA Mode

UART Interrupt Source	**Interrupt Enable Bit**	**Interrupt Indicator To Interrupt Controller**	**Interrupt Flag**	**Flag Cleared By**
LIN interrupt	LIN_IEN	LINIF	**LINIF**	Write '1' to SLVHDETF/BRKDETF/BITEF/SLVIDPEF/SLVHEF
Buffer Error Interrupt INT_BUF_ERR	BUFERRIEN	BERRINT	**BERRIF** = (TXOVIF or RXOVIF)	Write '1' to TXOVIF/ RXOVIF
RX Time-out Interrupt INT_TOUT	RXTOIEN	RXTOINT	**RXTOIF**	Read UART_DAT
Modem Status Interrupt INT_MODEM	MODEMIEN	MODEMINT	**MODENIF** = (CTSDETF)	Write '1' to CTSDETF
Receive Line Status Interrupt INT_RLS	RLSIEN	RLSINT	**RLSIF** = (BIF or FEF or PEF or RS-485_ADD_DETFor SCERR)	Write '1' to BIF/FEF/PEF/ RS-485_ADD_DETF/ TXOVERR/ RXOVERR
Transmit Holding Register Empty Interrupt INT_THRE	THREIEN	THERINT	**THREIF**	Write UART_DAT
Receive Data Available Interrupt INT_RDA	RDAIEN	RDAINT	**RDAIF**	Read UART_DAT

Table 6.31-2 UART Interrupt Sources and Flag List in Software Mode

Time-out Register (UART_TOUT)

Register	Offset	R/W	Description	Reset Value
UART_TOUT	UARTx_BA+0x20	R/W	UARTx Time-out Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
DLY							
7	6	5	4	3	2	1	0
TOIC							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:8]	**DLY**	**TX Delay Time Value** This field is use to programming the transfer delay time between the last stop bit and next start bit.  **Note:** The counter clock is baud rate clock
[7:0]	**TOIC**	**Time-Out Interrupt Comparator** The time-out counter resets and starts counting (the counting clock = baud rate clock) whenever the RX FIFO receives a new data word. Once the content of time-out counter (TOUT_CNT) is equal to that of time-out interrupt comparator (TOIC), a receiver time-out interrupt (INT_TOUT) is generated if RXTOIEN (UART_INTEN[4]). A new incoming data word or RX FIFO empty clears INT_TOUT. In order to avoid receiver time-out interrupt generation immediately during one character is being received, TOIC value should be set between 40 and 255. So, for example, if TOIC is set with 40, the time-out interrupt is generated after four characters are not received when 1 stop bit and no parity check is set for UART transfer.

Baud Rate Divider Register (UART_BAUD)

Register	Offset	R/W	Description	Reset Value
UART_BAUD	UARTx_BA+0x24	R/W	UARTx Baud Rate Divisor Register	0x0F00_0000

31	30	29	28	27	26	25	24
Reserved		BAUDM1	BAUDM0	EDIVM1			
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
BRD							
7	6	5	4	3	2	1	0
BRD							

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29]	**BAUDM1**	**Divider X Enable Bit** The BRD = Baud Rate Divider, and the baud rate equation is Baud Rate = Clock / [M * (BRD + 2)]; The default value of M is 16. 0 = Divider X Disabled (the equation of M = 16). 1 = Divider X Enabled (the equation of M = X+1, but EDIVM1 (UART_BAUD[27:24]) must >= 8). Refer to the table below for more information. **Note:** In IrDA mode, this bit must disable.
[28]	**BAUDM0**	**Divider X Equal To 1** 0 = Divider M = X (the equation of M = X+1, but EDIVM1 (UART_BAUD[27:24]) must >= 8). 1 = Divider M = 1 (the equation of M = 1, but BRD (UART_BAUD[15:0]) must >= 3). Refer to the table below for more information.
[27:24]	**EDIVM1**	**Divider X** The baud rate divider M = X+1.
[23:16]	**Reserved**	Reserved.
[15:0]	**BRD**	**Baud Rate Divider** The field indicated the baud rate divider

The baud rate equation is Baud Rate = UART_CLK / M * [BRD + 2], where M and BRD are defined in Baud Rate Divider Register (UART_BAUD). The following tables list the equations in the various conditions and the UART baud rate setting.

Mode	BAUDM1	BAUDM0	DIVIDER X	BRD	Baud Rate Equation
0	Disable	0	Don't Care	A	UART_CLK / [16 * (A+2)]
1	Enable	0	B	A	UART_CLK / [(B+1) * (A+2)] , B must >= 8

2	Enable	1	Don't care	A	UART_CLK / (A+2), A must >=9

Table 6.31-3 Baud Rate Equation Table

System Clock = Internal 22.1184 MHz High-Speed Oscillator						
Baud Rate	**Mode0**		**Mode1**		**Mode2**	
	Parameter	**Register**	**Parameter**	**Register**	**Parameter**	**Register**
921600	x	x	A=0,B=11	0x2B00_0000	A=22	0x3000_0016
460800	A=1	0x0000_0001	A=1,B=15 A=2,B=11	0x2F00_0001 0x2B00_0002	A=46	0x3000_002E
230400	A=4	0x0000_0004	A=4,B=15 A=6,B=11	0x2F00_0004 0x2B00_0006	A=94	0x3000_005E
115200	A=10	0x0000_000A	A=10,B=15 A=14,B=11	0x2F00_000A 0x2B00_000E	A=190	0x3000_00BE
57600	A=22	0x0000_0016	A=22,B=15 A=30,B=11	0x2F00_0016 0x2B00_001E	A=382	0x3000_017E
38400	A=34	0x0000_0022	A=62,B=8 A=46,B=11 A=34,B=15	0x2800_003E 0x2B00_002E 0x2F00_0022	A=574	0x3000_023E
19200	A=70	0x0000_0046	A=126,B=8 A=94,B=11 A=70,B=15	0x2800_007E 0x2B00_005E 0x2F00_0046	A=1150	0x3000_047E
9600	A=142	0x0000_008E	A=254,B=8 A=190,B=11 A=142,B=15	0x2800_00FE 0x2B00_00BE 0x2F00_008E	A=2302	0x3000_08FE
4800	A=286	0x0000_011E	A=510,B=8 A=382,B=11 A=286,B=15	0x2800_01FE 0x2B00_017E 0x2F00_011E	A=4606	0x3000_11FE

Table 6.31-4 Baud Rate Equation Table

IrDA Control Register (IRDA)

Register	Offset	R/W	Description	Reset Value
UART_IRDA	UARTx_BA+0x28	R/W	UARTx IrDA Control Register	0x0000_0040

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FIXPULSE	**RXINV**	**TXINV**	**Reserved**			**TXEN**	**Reserved**

Bits	Description	
[31:7]	**Reserved**	Reserved.
[7]	**FIXPULSE**	Pulse width of TX is fixed 1.6us.
[6]	**RXINV**	**IrDA Inverse Receive Input Signal** 0 = None inverse receiving input signal. 1 = Inverse receiving input signal. (Default)
[5]	**TXINV**	**IrDA Inverse Transmitting Output Signal** 0 = None inverse transmitting signal. (Default) 1 = Inverse transmitting output signal.
[4:2]	**Reserved**	Reserved.
[1]	**TXEN**	**IrDA Receiver/Transmitter Selection Enable Bit** 0 = IrDA Transmitter Disabled and Receiver Enabled. (Default) 1 = IrDA Transmitter Enabled and Receiver Disabled.
[0]	**Reserved**	Reserved.

Note: In IrDA mode, the UART_BAUD [BAUDM1] register must be disabled (the baud equation must be Clock / 16 * (BRD).

UART Alternate Control/Status Register (UART_ALTCTL)

Register	Offset	R/W	Description	Reset Value
UART_ALTCTL	UARTx_BA+0x2C	R/W	UARTx Alternate Control/Status Register	0x0000_000C

31	30	29	28	27	26	25	24
ADDRMV							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
ADDRDEN	Reserved				RS485AUD	RS485AAD	RS485NMM
7	6	5	4	3	2	1	0
LINTXEN	LINRXEN	Reserved		BRKFL			

Bits	Description	
[31:24]	**ADDRMV**	**Address Match Value** This field contains the RS-485 address match values. **Note:** This field is used for RS-485 auto address detection mode.
[23:16]	**Reserved**	Reserved.
[15]	**ADDRDEN**	**RS-485 Address Detection Enable Bit** This bit is use to enable RS-485 address detection mode. 0 = address detection mode Disabled. 1 = Address detection mode Enabled. **Note:** This field is used for RS-485 any operation mode.
[14:11]	**Reserved**	Reserved.
[10]	**RS485AUD**	**RS-485 Auto Direction Mode (AUD)** 0 = RS-485 Auto Direction Operation (AUO) mode Disabled. 1 = RS-485 Auto Direction Operation (AUO) mode Enabled. **Note:** It can be active with RS-485_AAD or RS-485_NMM operation mode.
[9]	**RS485AAD**	**RS-485 Auto Address Detection Operation Mode (AAD)** 0 = RS-485 Auto Address Detection (AAD) Operation mode Disabled. 1 = RS-485 Auto Address Detection (AAD) Operation mode Enabled. **Note:** It can't be active with RS-485_NMM operation mode.
[8]	**RS485NMM**	**RS-485 Normal Multi-Drop Operation Mode (NMM)** 0 = RS-485 Normal Multi-drop Operation Mode (NMM) Disabled. 1 = RS-485 Normal Multi-drop Operation Mode (NMM) Enabled. **Note:** It can't be active with RS-485_AAD operation mode.
[7]	**LINTXEN**	**LIN TX Break Mode Enable Bit** The LIN TX header can be "break field" or "break and sync field" or "break, sync and frame ID field" depending on the setting HSEL register.

		0 = Send LIN TX header Disabled. 1 = Send LIN TX header Enabled. **Note:** When transmitter header field (it may be “break” or “break + sync” or “break + sync + frame ID” selected by HSEL field) transfer operation finished, this bit will be cleared automatically.
[6]	**LINRXEN**	**LIN RX Enable Bit** 0 = LIN RX mode Disabled. 1 = LIN RX mode Enabled.
[5:4]	**Reserved**	Reserved.
[3:0]	**LIN_BKFL**	**LIN Break Field Length** This field indicates a 4-bit LIN TX break field count. **Note1:** This break field length is BRKFL + 1. **Note2:** According to LIN spec, the reset value is 0xC (break field length = 13).

UART Function Select Register (UART_FUNCSEL)

Register	Offset	R/W	Description	Reset Value
UART_FUNCSEL	UARTx_BA+0x30	R/W	UARTx Function Select Register	0x0000_0000

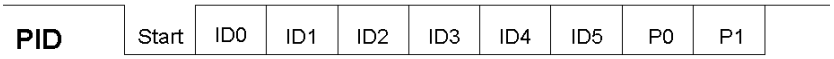
31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					FUNCSEL		

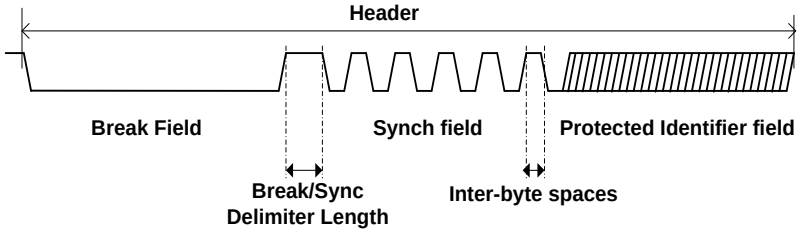
Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**FUNCSEL**	**Function Select Enable Bit** 000 = UART function. 001 = LIN function Enabled. 010 = IrDA function Enabled. 011 = RS-485 function Enabled. 100 = Smart-Card function Enabled.

UART LIN Control Register (UART_LINCTL)

Register	Offset	R/W	Description	Reset Value
UART_LINCTL	UARTx_BA+0x34	R/W	UARTx LIN Control Register	0x000C_0000

31	30	29	28	27	26	25	24
PID							
23	22	21	20	19	18	17	16
HSEL		BSL		BRKFL			
15	14	13	12	11	10	9	8
Reserved			BITERREN	RXOFF	BRKDETEN	IDPEN	SENDH
7	6	5	4	3	2	1	0
Reserved			MUTE	SLVDUEN	SLVAREN	SLVHDEN	SLVEN

Bits	Description	
[31:24]	**PID**	**This Field Contains The LIN Frame ID Value In LIN Function Mode, The Frame ID Parity Can Be Generated By Software Or Hardware Depending On UART_LINCTL [IDPEN]** If the parity generated by hardware (IDPEN (UART_LINCTL[9]) = 1), user fill ID0~ID5, hardware will calculi P0 and P1, otherwise user must filled frame ID and parity in this field.  P0 = ID0 xor ID1 xor ID2 xor ID4 P1 = ~(ID1 xor ID3 xor ID4 xor ID5) **Note1:** User can fill any 8-bit value to this field and the bit 24 indicates ID0 (LSB first) **Note2:** This field can be used for LIN Master mode or Slave mode.
[23:22]	**HSEL**	**LIN Header Selection** 00 = LIN header includes “break field”. 01 = LIN header includes “break field” and “sync field”. 10 = LIN header includes “break field”, “sync field” and “frame ID field”. 11 = LIN header includes “break field”, “sync field” and “frame ID field”, but this mode only supports Receiver mode, not support transmitter mode. This mode difference with mode “10”; in this mode, the receiver will receive ID field (not check the PID (UART_LINCTL[31:24]) register) and when received ID field the SLVHDETF will be asserted (if SLVHDEN (UART_LINCTL[1]) be set). **Note:** This bit is used to master mode for LIN to sending header field (SENDH = 1) or used to slave to indicates wake-up condition from mute mode (MUTE).
[21:20]	**BSL**	**LIN Break/Sync Delimiter Length** 00 = LIN break/sync delimiter length is 1 bit time. 10 = The LIN break/sync delimiter length is 2 bit time. 10 = The LIN break/sync delimiter length is 3 bit time. 11 = The LIN break/sync delimiter length is 4 bit time.

		 **Note:** This bit used for LIN master to send header field.
[19:16]	**BRKFL**	**LIN Break Field Length** This field indicates a 4-bit LIN TX break field count. **Note1:** These registers are shadow registers of LIN_BKFL (UART_ALTCTL[3:0]), User can read/write it by setting LIN_BKFL (UART_ALTCTL[3:0]) or LIN_BKFL (UART_LINCTL[19:16]). **Note2:** This break field length is BRKFL + 1. **Note3:** According to LIN spec, the reset value is 0XC (break field length = 13).
[15:13]	**Reserved**	Reserved.
[12]	**BITERREN**	**Bit Error Detect Enable Bit** 0 = Bit error detection function Disabled. 1 = Bit error detection Enabled. **Note:** In LIN function mode, when occur bit error, hardware will generate an interrupt to CPU (INT_LIN) and the BITEF (UART_LINSTS[9]) flag will be asserted.
[11]	**RXOFF**	If the receiver is be enabled (RXOFF = 0), all received byte data will be accepted and stored in the RX-FIFO, and if the receiver is disabled (RXOFF = 1), all received byte data will be ignore. 0 = Bit error detection function Disabled. 1 = Bit error detection Enabled. **Note:** This bit is only valid when operating in LIN function mode (UART_FUNCSEL = 2'b01).
[10]	**BRKDETEN**	**LIN Break Detection Enable Bit** When detect great than 11/10 bits are detected as 0, and are followed by a delimiter character, the BRKDETF flag (UART_LINSTS[8]) at the end of break field. If the LINIEN bit (UART_INTEN[8]) = 1, an interrupt will be generated. 0 = LIN break detection Disabled. 1 = LIN break detection Enabled.
[9]	**IDPEN**	**LIN ID Parity Enable Bit** 0 = LIN frame ID parity Disabled. 1 = LIN frame ID parity Enabled. **Note1:** This bit can be used for LIN master to sending header field (SENDH = 1 and HSEL = 2'b10) or be used for enable LIN slave received frame ID parity checked. **Note2:** This bit is only used when operation header transmitter is in HSEL = 2'b10.
[8]	**SENDH**	**LIN TX Send Header Enable Bit** The LIN TX header can be "break field" or "break and sync field" or "break, sync and frame ID field" depending on the setting HSEL register. 0 = Send LIN TX header Disabled. 1 = Send LIN TX header Enabled. **Note:** When transmitter header field (it may be "break" or "break + sync" or "break + sync + frame ID" selected by HSEL field) transfer operation finished, this bit will be cleared automatically.

[7:5]	**Reserved**	Reserved.
[4]	**MUTE**	**LIN Mute Mode Enable Bit** 0 = LIN mute mode. Disabled 1 = LIN mute mode Enabled. **Note:** The wake-up condition from mute mode and each control and interactions of this field are explained in 6.31.5.3.
[3]	**SLVDUEN**	**LIN Slave Divider Update Method Enable Bit** 0 = UART_BAUD is updated as soon as UART_BAUD is writing by software (if no automatic resynchronization update occurs at the same time). 1 = UART_BAUD is updated at the next received character. User must set the bit before checksum reception. **Note1:** This bit only valid in LIN slave mode (SLVEN = 1). **Note2:** This bit is used for LIN slave automatic resynchronization mode (for non-automatic resynchronization mode, this bit should be kept cleared). **Note3:** The control and interactions of this field are explained in 6.31.5.3.
[2]	**SLVAREN**	**LIN Slave Automatic Resynchronization Mode Enable Bit** 0 = LIN automatic resynchronization Disabled. 1 = LIN automatic resynchronization Enabled. **Note1:** This bit only valid in LIN slave mode (SLVEN = 1). **Note2:** When operating in Automatic Resynchronization mode, the baud rate setting must be mode2 (BAUDM1 (UART_BAUD[29]) and BAUDM0 (UART_BAUD[28]) must be 1). **Note3:** The control and interactions of this field are explained in 6.31.5.3.
[1]	**SLVHDEN**	**LIN Slave Header Detection Enable Bit** 0 = LIN slave header detection Disabled. 1 = LIN slave header detection Enabled. **Note1:** This bit only valid in LIN slave mode (SLVEN = 1). **Note2:** In LIN function mode, when header field (break + sync + frame ID) is detected, hardware will generate an interrupt to CPU (INT_LIN) and the SLVHDETF flag (UART_LINSTS[0]) will be asserted.
[0]	**SLVEN**	**LIN Slave Mode Enable Bit** 0 = LIN slave mode Disabled. 1 = LIN slave mode Enabled.

LIN Status Register (UART_LINSTS)

Register	Offset	R/W	Description	Reset Value
UART_LINSTS	UARTx_BA+0x38	R/W	UARTx LIN Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved						BITEF	BRKDETF
7	6	5	4	3	2	1	0
Reserved				SLVSYNCF	SLVIDPEF	SLVHEF	SLVHDETF

Bits	Description	
[31:10]	**Reserved**	Reserved.
[9]	**BITEF**	**Bit Error Detect Status Flag (Read Only)** At TX transfer state, hardware will monitoring the bus state, if the input pin (SIN) state not equals to the output pin (SOUT) state, BITEF will be set. When occur bit error, hardware will generate an interrupt to CPU (INT_LIN). **Note1:** This bit is read only, but can be cleared by writing 1 to it. **Note2:** This bit is only valid when enable bit error detection function (BRKL (UART_LINCTL[12]) == 1).
[8]	**BRKDETF**	**LIN Break Detection Flag (Read Only)** This bit is set by hardware when a break is detected and be cleared by writing 1 to it. 0 = LIN break not detected. 1 = LIN break detected. **Note1:** This bit is read only, but can be cleared by writing 1 to it. **Note2:** This bit is only valid when enable LIN break detection function (BRKDETEN (UART_LINCTL[10]))
[7:4]	**Reserved**	Reserved.
[3]	**SLVSYNCF**	**LIN Slave Sync Field** This bit indicates that the LIN sync field is being analyzed. When the receiver header have some error been detect, user must to reset the internal circuit to re-search new frame header by writing 1 to this bit. 0 = The current character is not at LIN sync state. 1 = The current character is at LIN sync state. **Note1:** This bit only valid in LIN Slave mode (SLVEN = 1). **Note2:** This bit is read only, but can be cleared by writing 1 to it. **Note3:** When user writing 1 to it, hardware will reload the initial baud-rate and re-search new frame header, the control and interactions of this field are explained in 6.31.5.3.
[2]	**SLVIDPEF**	**LIN Slave ID Parity Error Flag (Read Only)**

		This bit is set by hardware when receipted frame ID parity is not correct. 0 = no active. 1 = Receipted frame ID parity is not correct. **Note1:** This bit is read only, but can be cleared by writing 1 to it. **Note2:** This bit is only valid in LIN slave mode (SLVEN (UART_LINCTL[0]) = 1) and LIN frame ID parity check function (IDPEN (UART_LINCTL[9])) is enabled.
[1]	**SLVHEF**	**LIN Slave Header Error Flag (Read Only)** This bit is set by hardware when a LIN header error is detected in LIN slave mode and be cleared by writing 1 to it. The header include “break delimiter is too short”, “frame error in sync field or Identifier field”, “sync field data is not 0x55 without automatic resynchronization mode”, “sync field deviation error with automatic resynchronization mode”, “sync field measure time-out with automatic resynchronization mode” and “LIN header reception time-out”. 0 = LIN header error not detected. 1 = LIN header error detected. **Note1:** This bit is read only, but can be cleared by writing 1 to it. **Note2:** This bit is only valid in LIN slave mode (SLVEN (UART_LINCTL[0]) = 1) and LIN slave header detection function (SLVHDEN (UART_LINCTL[1])) is enabled.
[0]	**SLVHDETF**	**LIN Slave Header Detection Flag (Read Only)** This bit is set by hardware when a LIN header is detected in LIN slave mode and be cleared by writing 1 to it. 0 = LIN header not detected. 1 = LIN header detected (break + sync + frame ID). **Note1:** This bit is read only, but can be cleared by writing 1 to it. **Note2:** This bit is only valid in LIN slave mode (SLVEN (UART_LINCTL[0]) = 1) and LIN slave header detection function (SLVHDEN (UART_LINCTL[1])) is enabled. **Note3:** When the ID parity check (IDPEN (UART_LINCTL[9]) = 1) is enabled, if hardware detect complete harder (“break + sync + frame ID”), the LINS_HEDT_F (UART_LINCTL[1]) will be set no matter the frame ID is corrected or not.

LIN Debug Register (UART_LINDEBUG)

Register	Offset	R/W	Description	Reset Value
UART_LINDEBUG	UARTx_BA+0x3C	R/W	UARTx LIN Debug Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				SYNCERRF	FRAMEERRF	TOF	DEVERRF

Bits	Description	
[31:3]	**Reserved**	Reserved.
[3]	**SYNCERRF**	**LIN Header Sync Data Error (Read Only)** This bit indicates the header error cause by the LIN received sync data is not 0x55.
[2]	**FRAMEERRF**	**LIN Header Frame Error Flag (Read Only)** This bit indicates the header error cause by break delimiter is too short or frame error in sync field or Identifier field.
[1]	**TOF**	**LIN Header Time-Out (Read Only)** This bit indicates the header error cause by the LIN header reception time-out.
[0]	**DEVERRF**	**LIN Header Deviation Error (Read Only)** This bit indicates the header error cause by the sync field deviation error or sync field measure time-out with automatic resynchronization mode.

6.32 USB 2.0 Device Controller

6.32.1 Overview

The USB device controller interfaces the AHB bus and the UTMI bus. The USB controller contains both the AHB master interface and AHB slave interface. CPU programs the USB controller registers through the AHB slave interface. For IN or OUT transfer, the USB device controller needs to write data to memory or read data from memory through the AHB master interface. The USB device controller is complaint with USB 2.0 specification and it contains 12 configurable endpoints in addition to control endpoint. These endpoints could be configured to BULK, INTERRUPT or ISO. The USB device controller has a built-in DMA to relieve the load of CPU.

6.32.2 Features

- USB Specification reversion 2.0 compliant
- Supports 12 configurable endpoints in addition to Control Endpoint
- Each of the endpoints can be Isochronous, Bulk or Interrupt and either IN or OUT direction
- Three different operation modes of an in-endpoint – Auto Validation mode, Manual Validation mode, Fly mode
- Supports DMA operation
- 4092 Bytes Configurable RAM used as endpoint buffer
- Supports Endpoint Maximum Packet Size up to 1024 bytes

6.32.3 Block Diagram

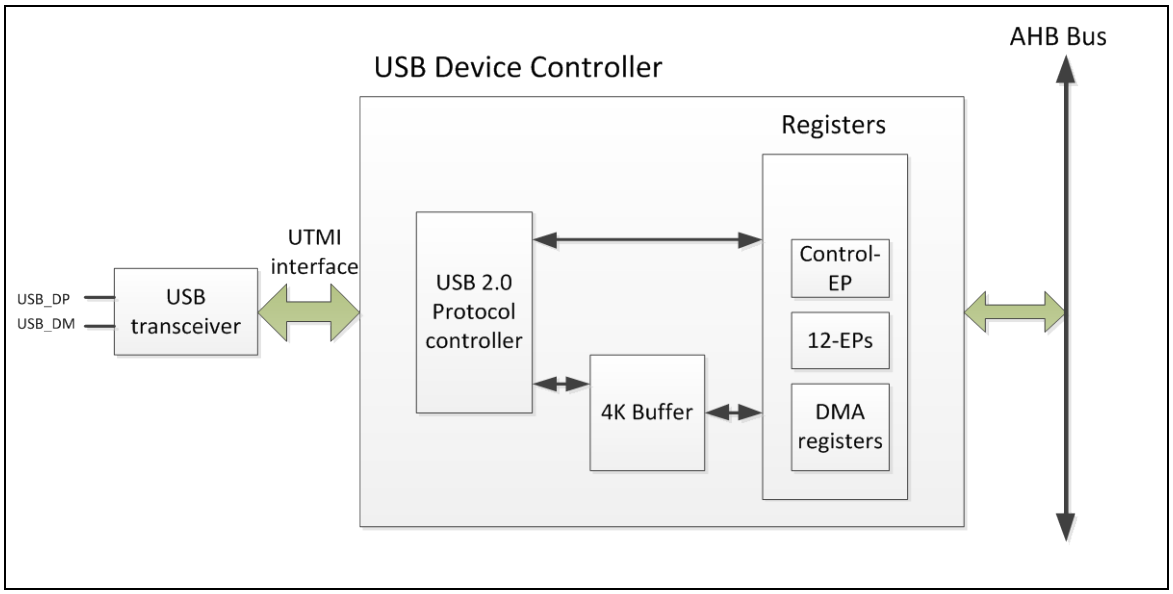


Figure 6.32-1 USB Device Controller Block Diagram

6.32.4 Functional Description

6.32.4.1 Operation of different In-transfer modes

The data for any in-transfer is written into the internal buffer when in turn is sent to the host on receipt of an in-token. There are three different modes by which the data sent to the host is validated by CPU.

- Auto-Validation Mode
- Manual-Validation Mode
- Fly Mode

6.32.4.2 Auto-Validation Mode

If an endpoint is selected to be operating in auto-validation mode, the endpoint responds only with data payload to be equal to EPMPS register. The endpoint controller wait until the amount of data is equal to EPMPS value and then validates the data. If CPU needs to send a short-packet at the end of a transfer, the SHORTTXEN bit of USBD_EPxRSPCTL[6] should be set. When this bit set, any remaining data in the buffer is validated and is sent to the host, for the forthcoming in-token.

This mode requires least intervention of CPU, as most of the work is done by the USB device controller. The mode can be selected, when the data payload sent to host is always equal to MPS size.

SHORTTXEN	Data Availability In Buffer	Data Sent/NAK Sent
0	< Max. Packet Size	NAK sent
0	>= Max. Packet Size	Data payload of max. packet size
1	< Max. Packet Size	Available data of < max. packet size

1	>= Max. Packet Size	Data payload of max. packet size sent

6.32.4.3 Manual-Validation Mode

If the endpoint is selected to be operating in manual-validation mode, the endpoint responds only when the data in the buffer is validated by CPU every time. The CPU has to write data into the buffer and then write the count of the data into EPxTXCNT register. Once the data is validating by writing a count into the EPxTXCNT register, the data is sent to the host on receipt of an in-token.

This mode requires intervention of CPU for each transfer. But this would be useful, if the data-count to be sent each time is not fixed, and it is being decided by CPU.

EPxTXCNT Written	Data Availability In Buffer	Data Sent/NAK Sent
NO	-	NAK
YES	EPxTXCNT	Data payload of EPxTXCNT sent

6.32.4.4 Fly Mode

The fly mode of operation is simplest mode of operation, where there is no validation procedure. The buffer is being filled by CPU. If an in-token is send from the host, the data in the buffer is automatically validated and sent to the host. If the data in the buffer spans more than one packet of maximum packet size, the controller automatically packs to equal to MPS and send it to the host.

This mode requires the least intervention by CPU. This mode is best suited for isochronous data transfer, where the speed of data transfer is more important than the packet size.

Data Availability In Buffer	Data Sent
< Max. Packet Size	Data available sent
>= Max. Packet Size	

6.32.4.5 Scatter-Gather function

When enabling the scatter gather DMA function, setting SGEN high and USBD_DMACNT set 8 bytes, DMA will be enabled to fetch the descriptor which describes the real memory address and length. The descriptor will be an 8-byte format, like the following:

	Format			
	[31]	[30]	[29:0]	
Word0	MEM_ADDR[31:0]			
Word1	EOT	RD	Reserved	Count[19:0]

MEM_ADDR: It specifies the memory address (AHB address).

EOT: end of transfer. When this bit is set to high, it means this is the last descriptor.

RD: “1” means read from memory into buffer. “0” means read from buffer into memory.

6.32.5 Registers Map

Register	Offset	R/W	Description	Reset Value
USBD Base Address: **USBD_BA = 0x4001_9000**				
USBD_GINTSTS	USBD_BA+0x000	R	Interrupt Status Low Register	0x0000_0000
USBD_GINTEN	USBD_BA+0x008	R/W	Interrupt Enable Low Register	0x0000_0001
USBD_BUSINTSTS	USBD_BA+0x010	R/W	USB Bus Interrupt Status Register	0x0000_0000
USBD_BUSINTEN	USBD_BA+0x014	R/W	USB Bus Interrupt Enable Register	0x0000_0040
USBD_OPER	USBD_BA+0x018	R/W	USB Operational Register	0x0000_0002
USBD_FRAMECNT	USBD_BA+0x01C	R	USB Frame Count Register	0x0000_0000
USBD_FADDR	USBD_BA+0x020	R/W	USB Function Address Register	0x0000_0000
USBD_TEST	USBD_BA+0x024	R/W	USB Test Mode Register	0x0000_0000
USBD_CEPDAT	USBD_BA+0x028	R/W	Control-Endpoint Data Buffer	0x0000_0000
USBD_CEPCTL	USBD_BA+0x02C	R/W	Control-Endpoint Control and Status	0x0000_0000
USBD_CEPINTEN	USBD_BA+0x030	R/W	Control-Endpoint Interrupt Enable	0x0000_0000
USBD_CEPINTSTS	USBD_BA+0x034	R/W	Control-Endpoint Interrupt Status	0x0000_1800
USBD_CEPTXCNT	USBD_BA+0x038	R/W	Control-Endpoint In-transfer Data Count	0x0000_0000
USBD_CEPRXCNT	USBD_BA+0x03C	R	Control-Endpoint Out-transfer Data Count	0x0000_0000
USBD_CEPDATCNT	USBD_BA+0x040	R	Control-Endpoint data count	0x0000_0000
USBD_SETUP1_0	USBD_BA+0x044	R	Setup1 & Setup0 bytes	0x0000_0000
USBD_SETUP3_2	USBD_BA+0x048	R	Setup3 & Setup2 Bytes	0x0000_0000
USBD_SETUP5_4	USBD_BA+0x04C	R	Setup5 & Setup4 Bytes	0x0000_0000
USBD_SETUP7_6	USBD_BA+0x050	R	Setup7 & Setup6 Bytes	0x0000_0000
USBD_CEPBUFSTART	USBD_BA+0x054	R/W	Control Endpoint RAM Start Address Register	0x0000_0000
USBD_CEPBUFEND	USBD_BA+0x058	R/W	Control Endpoint RAM End Address Register	0x0000_0000
USBD_DMACTL	USBD_BA+0x05C	R/W	DMA Control Status Register	0x0000_0000
USBD_DMACNT	USBD_BA+0x060	R/W	DMA Count Register	0x0000_0000
USBD_EPADAT	USBD_BA+0x064	R/W	Endpoint A Data Register	0x0000_0000
USBD_EPAINTSTS	USBD_BA+0x068	R/W	Endpoint A Interrupt Status Register	0x0000_0003
USBD_EPAINTEN	USBD_BA+0x06C	R/W	Endpoint A Interrupt Enable Register	0x0000_0000
USBD_EPADATCNT	USBD_BA+0x070	R	Endpoint A Data Available Count Register	0x0000_0000
USBD_EPARSPCTL	USBD_BA+0x074	R/W	Endpoint A Response Control Register	0x0000_0000

USBD_EPAMPS	USBD_BA+0x078	R/W	Endpoint A Maximum Packet Size Register	0x0000_0000
USBD_EPATXCNT	USBD_BA+0x07C	R/W	Endpoint A Transfer Count Register	0x0000_0000
USBD_EPACFG	USBD_BA+0x080	R/W	Endpoint A Configuration Register	0x0000_0012
USBD_EPABUFSTART	USBD_BA+0x084	R/W	Endpoint A RAM Start Address Register	0x0000_0000
USBD_EPABUFEND	USBD_BA+0x088	R/W	Endpoint A RAM End Address Register	0x0000_0000
USBD_EPBDAT	USBD_BA+0x08C	R/W	Endpoint B Data Register	0x0000_0000
USBD_EPBINTSTS	USBD_BA+0x090	R/W	Endpoint B Interrupt Status Register	0x0000_0003
USBD_EPBINTEN	USBD_BA+0x094	R/W	Endpoint B Interrupt Enable Register	0x0000_0000
USBD_EPBDATCNT	USBD_BA+0x098	R	Endpoint B Data Available Count Register	0x0000_0000
USBD_EPBRSPCTL	USBD_BA+0x09C	R/W	Endpoint B Response Control Register	0x0000_0000
USBD_EPBMPS	USBD_BA+0x0A0	R/W	Endpoint B Maximum Packet Size Register	0x0000_0000
USBD_EPBTXCNT	USBD_BA+0x0A4	R/W	Endpoint B Transfer Count Register	0x0000_0000
USBD_EPBCFG	USBD_BA+0x0A8	R/W	Endpoint B Configuration Register	0x0000_0022
USBD_EPBBUFSTART	USBD_BA+0x0AC	R/W	Endpoint B RAM Start Address Register	0x0000_0000
USBD_EPBBUFEND	USBD_BA+0x0B0	R/W	Endpoint B RAM End Address Register	0x0000_0000
USBD_EPCDAT	USBD_BA+0x0B4	R/W	Endpoint C Data Register	0x0000_0000
USBD_EPCINTSTS	USBD_BA+0x0B8	R/W	Endpoint C Interrupt Status Register	0x0000_0003
USBD_EPCINTEN	USBD_BA+0x0BC	R/W	Endpoint C Interrupt Enable Register	0x0000_0000
USBD_EPCDATCNT	USBD_BA+0x0C0	R	Endpoint C Data Available Count Register	0x0000_0000
USBD_EPCRSPCTL	USBD_BA+0x0C4	R/W	Endpoint C Response Control Register	0x0000_0000
USBD_EPCMPS	USBD_BA+0x0C8	R/W	Endpoint C Maximum Packet Size Register	0x0000_0000
USBD_EPCTXCNT	USBD_BA+0x0CC	R/W	Endpoint C Transfer Count Register	0x0000_0000
USBD_EPCCFG	USBD_BA+0x0D0	R/W	Endpoint C Configuration Register	0x0000_0032
USBD_EPCBUFSTART	USBD_BA+0x0D4	R/W	Endpoint C RAM Start Address Register	0x0000_0000
USBD_EPCBUFEND	USBD_BA+0x0D8	R/W	Endpoint C RAM End Address Register	0x0000_0000
USBD_EPDDAT	USBD_BA+0x0DC	R/W	Endpoint D Data Register	0x0000_0000
USBD_EPDINTSTS	USBD_BA+0x0E0	R/W	Endpoint D Interrupt Status Register	0x0000_0003
USBD_EPDINTEN	USBD_BA+0x0E4	R/W	Endpoint D Interrupt Enable Register	0x0000_0000
USBD_EPDDATCNT	USBD_BA+0x0E8	R	Endpoint D Data Available Count Register	0x0000_0000
USBD_EPDRSPCTL	USBD_BA+0x0EC	R/W	Endpoint D Response Control Register	0x0000_0000
USBD_EPDMPS	USBD_BA+0x0F0	R/W	Endpoint D Maximum Packet Size Register	0x0000_0000

USBD_EPDTXCNT	USBD_BA+0x0F4	R/W	Endpoint D Transfer Count Register	0x0000_0000
USBD_EPDCFG	USBD_BA+0x0F8	R/W	Endpoint D Configuration Register	0x0000_0042
USBD_EPDBUFSTART	USBD_BA+0x0FC	R/W	Endpoint D RAM Start Address Register	0x0000_0000
USBD_EPDBUFEND	USBD_BA+0x100	R/W	Endpoint D RAM End Address Register	0x0000_0000
USBD_EPEDAT	USBD_BA+0x104	R/W	Endpoint E Data Register	0x0000_0000
USBD_EPEINTSTS	USBD_BA+0x108	R/W	Endpoint E Interrupt Status Register	0x0000_0003
USBD_EPEINTEN	USBD_BA+0x10C	R/W	Endpoint E Interrupt Enable Register	0x0000_0000
USBD_EPEDATCNT	USBD_BA+0x110	R	Endpoint E Data Available Count Register	0x0000_0000
USBD_EPERSPCTL	USBD_BA+0x114	R/W	Endpoint E Response Control Register	0x0000_0000
USBD_EPEMPS	USBD_BA+0x118	R/W	Endpoint E Maximum Packet Size Register	0x0000_0000
USBD_EPETXCNT	USBD_BA+0x11C	R/W	Endpoint E Transfer Count Register	0x0000_0000
USBD_EPECFG	USBD_BA+0x120	R/W	Endpoint E Configuration Register	0x0000_0052
USBD_EPEBUFSTART	USBD_BA+0x124	R/W	Endpoint E RAM Start Address Register	0x0000_0000
USBD_EPEBUFEND	USBD_BA+0x128	R/W	Endpoint E RAM End Address Register	0x0000_0000
USBD_EPFDAT	USBD_BA+0x12C	R/W	Endpoint F Data Register	0x0000_0000
USBD_EPFINTSTS	USBD_BA+0x130	R/W	Endpoint F Interrupt Status Register	0x0000_0003
USBD_EPFINTEN	USBD_BA+0x134	R/W	Endpoint F Interrupt Enable Register	0x0000_0000
USBD_EPFDATCNT	USBD_BA+0x138	R	Endpoint F Data Available Count Register	0x0000_0000
USBD_EPFRSPCTL	USBD_BA+0x13C	R/W	Endpoint F Response Control Register	0x0000_0000
USBD_EPFMPS	USBD_BA+0x140	R/W	Endpoint F Maximum Packet Size Register	0x0000_0000
USBD_EPFTXCNT	USBD_BA+0x144	R/W	Endpoint F Transfer Count Register	0x0000_0000
USBD_EPFCFG	USBD_BA+0x148	R/W	Endpoint F Configuration Register	0x0000_0062
USBD_EPFBUFSTART	USBD_BA+0x14C	R/W	Endpoint F RAM Start Address Register	0x0000_0000
USBD_EPFBUFEND	USBD_BA+0x150	R/W	Endpoint F RAM End Address Register	0x0000_0000
USBD_EPGDAT	USBD_BA+0x154	R/W	Endpoint G Data Register	0x0000_0000
USBD_EPGINTSTS	USBD_BA+0x158	R/W	Endpoint G Interrupt Status Register	0x0000_0003
USBD_EPGINTEN	USBD_BA+0x15C	R/W	Endpoint G Interrupt Enable Register	0x0000_0000
USBD_EPGDATCNT	USBD_BA+0x160	R	Endpoint G Data Available Count Register	0x0000_0000
USBD_EPGRSPCTL	USBD_BA+0x164	R/W	Endpoint G Response Control Register	0x0000_0000
USBD_EPGMPS	USBD_BA+0x168	R/W	Endpoint G Maximum Packet Size Register	0x0000_0000
USBD_EPGTXCNT	USBD_BA+0x16C	R/W	Endpoint G Transfer Count Register	0x0000_0000

USBD_EPGCFG	USBD_BA+0x170	R/W	Endpoint G Configuration Register	0x0000_0072
USBD_EPGBUFSTART	USBD_BA+0x174	R/W	Endpoint G RAM Start Address Register	0x0000_0000
USBD_EPGBUFEND	USBD_BA+0x178	R/W	Endpoint G RAM End Address Register	0x0000_0000
USBD_EPHDAT	USBD_BA+0x17C	R/W	Endpoint H Data Register	0x0000_0000
USBD_EPHINTSTS	USBD_BA+0x180	R/W	Endpoint H Interrupt Status Register	0x0000_0003
USBD_EPHINTEN	USBD_BA+0x184	R/W	Endpoint H Interrupt Enable Register	0x0000_0000
USBD_EPHDATCNT	USBD_BA+0x188	R	Endpoint H Data Available Count Register	0x0000_0000
USBD_EPHRSPCTL	USBD_BA+0x18C	R/W	Endpoint H Response Control Register	0x0000_0000
USBD_EPHMPS	USBD_BA+0x190	R/W	Endpoint H Maximum Packet Size Register	0x0000_0000
USBD_EPHTXCNT	USBD_BA+0x194	R/W	Endpoint H Transfer Count Register	0x0000_0000
USBD_EPHCFG	USBD_BA+0x198	R/W	Endpoint H Configuration Register	0x0000_0082
USBD_EPHBUFSTART	USBD_BA+0x19C	R/W	Endpoint H RAM Start Address Register	0x0000_0000
USBD_EPHBUFEND	USBD_BA+0x1A0	R/W	Endpoint H RAM End Address Register	0x0000_0000
USBD_EPIDAT	USBD_BA+0x1A4	R/W	Endpoint I Data Register	0x0000_0000
USBD_EPIINTSTS	USBD_BA+0x1A8	R/W	Endpoint I Interrupt Status Register	0x0000_0003
USBD_EPIINTEN	USBD_BA+0x1AC	R/W	Endpoint I Interrupt Enable Register	0x0000_0000
USBD_EPIDATCNT	USBD_BA+0x1B0	R	Endpoint I Data Available Count Register	0x0000_0000
USBD_EPIRSPCTL	USBD_BA+0x1B4	R/W	Endpoint I Response Control Register	0x0000_0000
USBD_EPIMPS	USBD_BA+0x1B8	R/W	Endpoint I Maximum Packet Size Register	0x0000_0000
USBD_EPITXCNT	USBD_BA+0x1BC	R/W	Endpoint I Transfer Count Register	0x0000_0000
USBD_EPICFG	USBD_BA+0x1C0	R/W	Endpoint I Configuration Register	0x0000_0092
USBD_EPIBUFSTART	USBD_BA+0x1C4	R/W	Endpoint I RAM Start Address Register	0x0000_0000
USBD_EPIBUFEND	USBD_BA+0x1C8	R/W	Endpoint I RAM End Address Register	0x0000_0000
USBD_EPJDAT	USBD_BA+0x1CC	R/W	Endpoint J Data Register	0x0000_0000
USBD_EPJINTSTS	USBD_BA+0x1D0	R/W	Endpoint J Interrupt Status Register	0x0000_0003
USBD_EPJINTEN	USBD_BA+0x1D4	R/W	Endpoint J Interrupt Enable Register	0x0000_0000
USBD_EPJDATCNT	USBD_BA+0x1D8	R	Endpoint J Data Available Count Register	0x0000_0000
USBD_EPJRSPCTL	USBD_BA+0x1DC	R/W	Endpoint J Response Control Register	0x0000_0000
USBD_EPJMPS	USBD_BA+0x1E0	R/W	Endpoint J Maximum Packet Size Register	0x0000_0000
USBD_EPJTXCNT	USBD_BA+0x1E4	R/W	Endpoint J Transfer Count Register	0x0000_0000
USBD_EPJCFG	USBD_BA+0x1E8	R/W	Endpoint J Configuration Register	0x0000_00A2

USBD_EPJBUFSTART	USBD_BA+0x1EC	R/W	Endpoint J RAM Start Address Register	0x0000_0000
USBD_EPJBUFEND	USBD_BA+0x1F0	R/W	Endpoint J RAM End Address Register	0x0000_0000
USBD_EPKDAT	USBD_BA+0x1F4	R/W	Endpoint K Data Register	0x0000_0000
USBD_EPKINTSTS	USBD_BA+0x1F8	R/W	Endpoint K Interrupt Status Register	0x0000_0003
USBD_EPKINTEN	USBD_BA+0x1FC	R/W	Endpoint K Interrupt Enable Register	0x0000_0000
USBD_EPKDATCNT	USBD_BA+0x200	R	Endpoint K Data Available Count Register	0x0000_0000
USBD_EPKRSPCTL	USBD_BA+0x204	R/W	Endpoint K Response Control Register	0x0000_0000
USBD_EPKMPS	USBD_BA+0x208	R/W	Endpoint K Maximum Packet Size Register	0x0000_0000
USBD_EPKTXCNT	USBD_BA+0x20C	R/W	Endpoint K Transfer Count Register	0x0000_0000
USBD_EPKCFG	USBD_BA+0x210	R/W	Endpoint K Configuration Register	0x0000_00B2
USBD_EPKBUFSTART	USBD_BA+0x214	R/W	Endpoint K RAM Start Address Register	0x0000_0000
USBD_EPKBUFEND	USBD_BA+0x218	R/W	Endpoint K RAM End Address Register	0x0000_0000
USBD_EPLDAT	USBD_BA+0x21C	R/W	Endpoint L Data Register	0x0000_0000
USBD_EPLINTSTS	USBD_BA+0x220	R/W	Endpoint L Interrupt Status Register	0x0000_0003
USBD_EPLINTEN	USBD_BA+0x224	R/W	Endpoint L Interrupt Enable Register	0x0000_0000
USBD_EPLDATCNT	USBD_BA+0x228	R	Endpoint L Data Available Count Register	0x0000_0000
USBD_EPLRSPCTL	USBD_BA+0x22C	R/W	Endpoint L Response Control Register	0x0000_0000
USBD_EPLMPS	USBD_BA+0x230	R/W	Endpoint L Maximum Packet Size Register	0x0000_0000
USBD_EPLTXCNT	USBD_BA+0x234	R/W	Endpoint L Transfer Count Register	0x0000_0000
USBD_EPLCFG	USBD_BA+0x238	R/W	Endpoint L Configuration Register	0x0000_00C2
USBD_EPLBUFSTART	USBD_BA+0x23C	R/W	Endpoint L RAM Start Address Register	0x0000_0000
USBD_EPLBUFEND	USBD_BA+0x240	R/W	Endpoint L RAM End Address Register	0x0000_0000
USBD_DMAADDR	USBD_BA+0x700	R/W	AHB DMA Address Register	0x0000_0000
USBD_PHYCTL	USBD_BA+0x704	R/W	USB PHY Control Register	0x0000_0420

6.32.6 Register Description

Global Interrupt Status Register (USBD_GINTSTS)

Register	Offset	R/W	Description	Reset Value
USBD_GINTSTS	USBD_BA+0x000	R	Global Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		EPLIF	EPKIF	EPJIF	EPIIF	EPHIF	EPGIF
7	6	5	4	3	2	1	0
EPFIF	EPEIF	EPDIF	EPCIF	EPBIF	EPAIF	CEPIF	USBIF

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13]	**EPLIF**	**Endpoints L Interrupt** When set, the corresponding Endpoint L's interrupt status register should be read to determine the cause of the interrupt. 0 = No interrupt event occurred. 1 = The related interrupt event is occurred.
[12]	**EPKIF**	**Endpoints K Interrupt** When set, the corresponding Endpoint K's interrupt status register should be read to determine the cause of the interrupt. 0 = No interrupt event occurred. 1 = The related interrupt event is occurred.
[11]	**EPJIF**	**Endpoints J Interrupt** When set, the corresponding Endpoint J's interrupt status register should be read to determine the cause of the interrupt. 0 = No interrupt event occurred. 1 = The related interrupt event is occurred.
[10]	**EPIIF**	**Endpoints I Interrupt** When set, the corresponding Endpoint I's interrupt status register should be read to determine the cause of the interrupt. 0 = No interrupt event occurred. 1 = The related interrupt event is occurred.

[9]	**EPHIF**	**Endpoints H Interrupt** When set, the corresponding Endpoint H's interrupt status register should be read to determine the cause of the interrupt. 0 = No interrupt event occurred. 1 = The related interrupt event is occurred.
[8]	**EPGIF**	**Endpoints G Interrupt** When set, the corresponding Endpoint G's interrupt status register should be read to determine the cause of the interrupt. 0 = No interrupt event occurred. 1 = The related interrupt event is occurred.
[7]	**EPFIF**	**Endpoints F Interrupt** When set, the corresponding Endpoint F's interrupt status register should be read to determine the cause of the interrupt. 0 = No interrupt event occurred. 1 = The related interrupt event is occurred.
[6]	**EPEIF**	**Endpoints E Interrupt** When set, the corresponding Endpoint E's interrupt status register should be read to determine the cause of the interrupt. 0 = No interrupt event occurred. 1 = The related interrupt event is occurred.
[5]	**EPDIF**	**Endpoints D Interrupt** When set, the corresponding Endpoint D's interrupt status register should be read to determine the cause of the interrupt. 0 = No interrupt event occurred. 1 = The related interrupt event is occurred.
[4]	**EPCIF**	**Endpoints C Interrupt** When set, the corresponding Endpoint C's interrupt status register should be read to determine the cause of the interrupt. 0 = No interrupt event occurred. 1 = The related interrupt event is occurred.
[3]	**EPBIF**	**Endpoints B Interrupt** When set, the corresponding Endpoint B's interrupt status register should be read to determine the cause of the interrupt. 0 = No interrupt event occurred. 1 = The related interrupt event is occurred.
[2]	**EPAIF**	**Endpoints A Interrupt** When set, the corresponding Endpoint A's interrupt status register should be read to determine the cause of the interrupt. 0 = No interrupt event occurred. 1 = The related interrupt event is occurred.
[1]	**CEPIF**	**Control Endpoint Interrupt** This bit conveys the interrupt status for control endpoint. When set, Control-ep's interrupt status register should be read to determine the cause of the interrupt. 0 = No interrupt event occurred. 1 = The related interrupt event is occurred.

[0]	**USBIF**	**USB Interrupt** This bit conveys the interrupt status for USB specific events endpoint. When set, USB interrupt status register should be read to determine the cause of the interrupt. 0 = No interrupt event occurred. 1 = The related interrupt event is occurred.

Global Interrupt Enable Register (USBD_GINTEN)

Register	Offset	R/W	Description	Reset Value
USBD_GINTEN	USBD_BA+0x008	R/W	Global Interrupt Enable Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		EPLIEN	EPKIEN	EPJIEN	EPIIEN	EPHIEN	EPGIENNN
7	6	5	4	3	2	1	0
EPFIEN	EPEIEN	EPDIEN	EPCIEN	EPBIEN	EPAIEN	CEPIEN	USBIEN

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13]	**EPLIEN**	**Interrupt Enable Control for Endpoint L** When set, this bit enables a local interrupt to be generated when an interrupt is pending for the endpoint L 0 = The related interrupt Disabled. 1 = The related interrupt Enabled.
[12]	**EPKIEN**	**Interrupt Enable Control for Endpoint K** When set, this bit enables a local interrupt to be generated when an interrupt is pending for the endpoint K 0 = The related interrupt Disabled. 1 = The related interrupt Enabled.
[11]	**EPJIEN**	**Interrupt Enable Control for Endpoint J** When set, this bit enables a local interrupt to be generated when an interrupt is pending for the endpoint J 0 = The related interrupt Disabled. 1 = The related interrupt Enabled.
[10]	**EPIIEN**	**Interrupt Enable Control for Endpoint I** When set, this bit enables a local interrupt to be generated when an interrupt is pending for the endpoint I 0 = The related interrupt Disabled. 1 = The related interrupt Enabled.
[9]	**EPHIEN**	**Interrupt Enable Control for Endpoint H** When set, this bit enables a local interrupt to be generated when an interrupt is pending for the endpoint H 0 = The related interrupt Disabled. 1 = The related interrupt Enabled.

[8]	**EPGIEN**	**Interrupt Enable Control for Endpoint G** When set, this bit enables a local interrupt to be generated when an interrupt is pending for the endpoint G 0 = The related interrupt Disabled. 1 = The related interrupt Enabled.
[7]	**EPFIEN**	**Interrupt Enable Control for Endpoint F** When set, this bit enables a local interrupt to be generated when an interrupt is pending for the endpoint F 0 = The related interrupt Disabled. 1 = The related interrupt Enabled.
[6]	**EPEIEN**	**Interrupt Enable Control for Endpoint E** When set, this bit enables a local interrupt to be generated when an interrupt is pending for the endpoint E 0 = The related interrupt Disabled. 1 = The related interrupt Enabled.
[5]	**EPDIEN**	**Interrupt Enable Control for Endpoint D** When set, this bit enables a local interrupt to be generated when an interrupt is pending for the endpoint D 0 = The related interrupt Disabled. 1 = The related interrupt Enabled.
[4]	**EPCIEN**	**Interrupt Enable Control for Endpoint C** When set, this bit enables a local interrupt to be generated when an interrupt is pending for the endpoint C 0 = The related interrupt Disabled. 1 = The related interrupt Enabled.
[3]	**EPBIEN**	**Interrupt Enable Control for Endpoint B** When set, this bit enables a local interrupt to be generated when an interrupt is pending for the endpoint B 0 = The related interrupt Disabled. 1 = The related interrupt Enabled.
[2]	**EPAIEN**	**Interrupt Enable Control for Endpoint A** When set, this bit enables a local interrupt to be generated when an interrupt is pending for the endpoint A. 0 = The related interrupt Disabled. 1 = The related interrupt Enabled.
[1]	**CEPIEN**	**Control Endpoint Interrupt Enable Bit** When set, this bit enables a local interrupt to be generated when an interrupt is pending for the control endpoint. 0 = The related interrupt Disabled. 1 = The related interrupt Enabled.
[0]	**USBIEN**	**USB Interrupt Enable Bit** When set, this bit enables a local interrupt to be generated when a USB event occurs on the bus. 0 = The related interrupt Disabled. 1 = The related interrupt Enabled.

USB Bus Interrupt Status Register (USBD_BUSINTSTS)

Register	Offset	R/W	Description	Reset Value
USBD_BUSINTSTS	USBD_BA+0x010	R/W	USB Bus Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							VBUSDETIF
7	6	5	4	3	2	1	0
Reserved	PHYCLKVLDIF	DMADONEIF	HISPDIF	SUSPENDIF	RESUMEIF	RSTIF	SOFIF

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**VBUSDETIF**	**VBUS Detection Interrupt Status** 0 = No VBUS is plug-in. 1 = VBUS is plug-in. **Note:** Write 1 to clear this bit to 0.
[7]	**Reserved**	Reserved.
[6]	**PHYCLKVLDIF**	**Usable Clock Interrupt** 0 = Usable clock is not available. 1 = Usable clock is available from the transceiver. **Note:** Write 1 to clear this bit to 0.
[5]	**DMADONEIF**	**DMA Completion Interrupt** 0 = No DMA transfer over. 1 = DMA transfer is over. **Note:** Write 1 to clear this bit to 0.
[4]	**HISPDIF**	**High-Speed Settle** 0 = No valid high-speed reset protocol is detected. 1 = Valid high-speed reset protocol is over and the device has settled in high-speed. **Note:** Write 1 to clear this bit to 0.
[3]	**SUSPENDIF**	**Suspend Request** This bit is set as default and it has to be cleared by writing '1' before the USB reset. This bit is also set when a USB Suspend request is detected from the host. 0 = No USB Suspend request is detected from the host. 1= USB Suspend request is detected from the host. **Note:** Write 1 to clear this bit to 0.

[2]	**RESUMEIF**	**Resume** When set, this bit indicates that a device resume has occurred. 0 = No device resume has occurred. 1 = Device resume has occurred. **Note:** Write 1 to clear this bit to 0.
[1]	**RSTIF**	**Reset Status** When set, this bit indicates that either the USB root port reset is end. 0 = No USB root port reset is end. 1 = USB root port reset is end. **Note:** Write 1 to clear this bit to 0.
[0]	**SOFIF**	**SOF Receive Control** This bit indicates when a start-of-frame packet has been received. 0 = No start-of-frame packet has been received. 1 = Start-of-frame packet has been received. **Note:** Write 1 to clear this bit to 0.

USB Bus Interrupt Enable Register (USBD_BUSINTEN)

Register	Offset	R/W	Description	Reset Value
USBD_BUSINTEN	USBD_BA+0x014	R/W	USB Bus Interrupt Enable Register	0x0000_0040

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							VBUSDETIEN
7	6	5	4	3	2	1	0
Reserved	PHYCLKVLDIEN	DMADONEIEN	HISPDIEN	SUSPENDIEN	RESUMEIEN	RSTIEN	SOFIEN

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**VBUSDETIEN**	**VBUS Detection Interrupt Enable Bit** This bit enables the VBUS floating detection interrupt. 0 = VBUS floating detection interrupt Disabled. 1 = VBUS floating detection interrupt Enabled.
[7]	**Reserved**	Reserved.
[6]	**PHYCLKVLDIEN**	**Usable Clock Interrupt** This bit enables the usable clock interrupt. 0 = Usable clock interrupt Disabled. 1 = Usable clock interrupt Enabled.
[5]	**DMADONEIEN**	**DMA Completion Interrupt** This bit enables the DMA completion interrupt 0 = DMA completion interrupt Disabled. 1 = DMA completion interrupt Enabled.
[4]	**HISPDIEN**	**High-Speed Settle** This bit enables the high-speed settle interrupt. 0 = High-speed settle interrupt Disabled. 1 = High-speed settle interrupt Enabled.
[3]	**SUSPENDIEN**	**Suspend Request** This bit enables the Suspend interrupt. 0 = Suspend interrupt Disabled. 1 = Suspend interrupt Enabled.

[2]	**RESUMEIEN**	**Resume** This bit enables the Resume interrupt. 0 = Resume interrupt Disabled. 1 = Resume interrupt Enabled.
[1]	**RSTIEN**	**Reset Status** This bit enables the USB-Reset interrupt. 0 = USB-Reset interrupt Disabled. 1 = USB-Reset interrupt Enabled.
[0]	**SOFIEN**	**SOF Interrupt** This bit enables the SOF interrupt. 0 = SOF interrupt Disabled. 1 = SOF interrupt Enabled.

USB Operational Register (USBD_OPER)

Register	Offset	R/W	Description	Reset Value
USBD_OPER	USBD_BA+0x018	R/W	USB Operational Register	0x0000_0002

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**CURSPD**	**HISPDEN**	**RESUMEEN**

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**CURSPD**	**USB Current Speed** 0 = The device has settled in Full Speed. 1 = The USB device controller has settled in High-speed.
[1]	**HISPDEN**	**USB High-Speed** 0 = The USB device controller to suppress the chirp-sequence during reset protocol, thereby allowing the USB device controller to settle in full-speed, even though it is connected to a USB2.0 Host. 1 = The USB device controller to initiate a chirp-sequence during reset protocol.
[0]	**RESUMEEN**	**Generate Resume** 0 = No Resume sequence to be initiated to the host. 1 = A Resume sequence to be initiated to the host if device remote wakeup is enabled. This bit is self-clearing.

USB Frame Count Register (USBD_FRAMECNT)

Register	Offset	R/W	Description	Reset Value
USBD_FRAMECNT	USBD_BA+0x01C	R	USB Frame Count Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**FRAMECNT**					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FRAMECNT					**MFRAMECNT**		

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13:3]	**FRAMECNT**	**Frame Counter** This field contains the frame count from the most recent start-of-frame packet.
[2:0]	**MFRAMECNT**	**Micro-Frame Counter** This field contains the micro-frame number for the frame number in the frame counter field.

USB Function Address Register (USBD_FADDR)

Register	Offset	R/W	Description	Reset Value
USBD_FADDR	USBD_BA+0x020	R/W	USB Function Address Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**FADDR**						

Bits	Description	
[31:7]	**Reserved**	Reserved.
[6:0]	**FADDR**	**USB Function Address** This field contains the current USB address of the device. This field is cleared when a root port reset is detected.

USB Test Mode Register (USBD_TEST)

Register	Offset	R/W	Description	Reset Value
USBD_TEST	USBD_BA+0x024	R/W	USB Test Mode Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**TESTMODE**		

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**TESTMODE**	**Test Mode Selection** 000 = Normal Operation. 001 = Test_J. 010 = Test_K. 011 = Test_SE0_NAK. 100 = Test_Packet. 101 = Test_Force_Enable. 110 = Reserved. 111 = Reserved. **Note:** This field is cleared when root port reset is detected.

Control-ep Data Buffer (USBD_CEPDAT)

Register	Offset	R/W	Description	Reset Value
USBD_CEPDAT	USBD_BA+0x028	R/W	Control-Endpoint Data Buffer	0x0000_0000

31	30	29	28	27	26	25	24
DAT							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DAT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DAT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DAT							

Bits	Description	
[31:0]	**DAT**	**Control-Endpoint Data Buffer** Control endpoint data buffer for the buffer transaction (read or write). **Note:** Only word or byte access are supported.

Control-Endpoint Control Register (USBD_CEPCTL)

Register	Offset	R/W	Description	Reset Value
USBD_CEPCTL	USBD_BA+0x02C	R/W	Control-Endpoint Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**FLUSH**	**ZEROLEN**	**STALLEN**	**NAKCLR**

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3]	**FLUSH**	**CEP-FLUSH Bit** 0 = No the packet buffer and its corresponding USBD_CEPDATCNT register to be cleared. 1 = The packet buffer and its corresponding USBD_CEPDATCNT register to be cleared. This bit is self-cleaning.
[2]	**ZEROLEN**	**Zero Packet Length** This bit is valid for Auto Validation mode only. 0 = No zero length packet to the host during Data stage to an IN token. 1 = USB device controller can send a zero length packet to the host during Data stage to an IN token. This bit gets cleared once the zero length data packet is sent. So, the local CPU need not write again to clear this bit.
[1]	**STALLEN**	**Stall Enable Bit** When this stall bit is set, the control endpoint sends a stall handshake in response to any in or out token thereafter. This is typically used for response to invalid/unsupported requests. When this bit is being set the NAK clear bit has to be cleared at the same time since the NAK clear bit has highest priority than STALL. It is automatically cleared on receipt of a next setup-token. So, the local CPU need not write again to clear this bit. 0 = No sends a stall handshake in response to any in or out token thereafter. 1 = The control endpoint sends a stall handshake in response to any in or out token thereafter. **Note:** Only when CPU writes data[1:0] is 2'b10 or 2'b00, this bit can be updated.

[0]	**NAKCLR**	**No Acknowledge Control** This bit plays a crucial role in any control transfer. 0 = The bit is being cleared by the local CPU by writing zero, the USB device controller will be responding with NAKs for the subsequent status phase. This mechanism holds the host from moving to the next request, until the local CPU is also ready to process the next request. 1 = This bit is set to one by the USB device controller, whenever a setup token is received. The local CPU can take its own time to finish off any house-keeping work based on the request and then clear this bit. **Note:** Only when CPU writes data[1:0] is 2'b10 or 2'b00, this bit can be updated.

Control Endpoint Interrupt Enable(USBD_CEPINTEN)

Register	Offset	R/W	Description	Reset Value
USBD_CEPINTEN	USBD_BA+0x030	R/W	Control-Endpoint Interrupt Enable	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved			BUFEMPTYIEN	BUFFULLIEN	STSDONEIEN	ERRIEN	STALLIEN
7	6	5	4	3	2	1	0
NAKIEN	RXPKIEN	TXPKIEN	PINGIEN	INTKIEN	OUTTKIEN	SETUPPKIEN	SETUPTKIEN

Bits	Description	
[31:13]	**Reserved**	Reserved.
[12]	**BUFEMPTYIEN**	**Buffer Empty Interrupt** 0 = The buffer empty interrupt in Control Endpoint Disabled. 1= The buffer empty interrupt in Control Endpoint Enabled.
[11]	**BUFFULLIEN**	**Buffer Full Interrupt** 0 = The buffer full interrupt in Control Endpoint Disabled. 1 = The buffer full interrupt in Control Endpoint Enabled.
[10]	**STSDONEIEN**	**Status Completion Interrupt** 0 = The Status Completion interrupt in Control Endpoint Disabled. 1 = The Status Completion interrupt in Control Endpoint Enabled.
[9]	**ERRIEN**	**USB Error Interrupt** 0 = The USB Error interrupt in Control Endpoint Disabled. 1 = The USB Error interrupt in Control Endpoint Enabled.
[8]	**STALLIEN**	**STALL Sent Interrupt** 0 = The STALL sent interrupt in Control Endpoint Disabled. 1 = The STALL sent interrupt in Control Endpoint Enabled.
[7]	**NAKIEN**	**NAK Sent Interrupt** 0 = The NAK sent interrupt in Control Endpoint Disabled. 1 = The NAK sent interrupt in Control Endpoint Enabled.
[6]	**RXPKIEN**	**Data Packet Received Interrupt** 0 = The data received interrupt in Control Endpoint Disabled. 1 = The data received interrupt in Control Endpoint Enabled.

[5]	**TXPKIEN**	**Data Packet Transmitted Interrupt** 0 = The data packet transmitted interrupt in Control Endpoint Disabled. 1 = The data packet transmitted interrupt in Control Endpoint Enabled.
[4]	**PINGIEN**	**Ping Token Interrupt** 0 = The ping token interrupt in Control Endpoint Disabled. 1 = The ping token interrupt Control Endpoint Enabled.
[3]	**INTKIEN**	**In Token Interrupt** 0 = The IN token interrupt in Control Endpoint Disabled. 1 = The IN token interrupt in Control Endpoint Enabled.
[2]	**OUTTKIEN**	**Out Token Interrupt** 0 = The OUT token interrupt in Control Endpoint Disabled. 1 = The OUT token interrupt in Control Endpoint Enabled.
[1]	**SETUPPKIEN**	**Setup Packet Interrupt** 0 = The SETUP packet interrupt in Control Endpoint Disabled. 1 = The SETUP packet interrupt in Control Endpoint Enabled.
[0]	**SETUPTKIEN**	**Setup Token Interrupt Enable Bit** 0 = The SETUP token interrupt in Control Endpoint Disabled. 1 = The SETUP token interrupt in Control Endpoint Enabled.

Control-Endpoint Interrupt Status (USBD_CEPINTSTS)

Register	Offset	R/W	Description	Reset Value
USBD_CEPINTSTS	USBD_BA+0x034	R/W	Control-Endpoint Interrupt Status	0x0000_1800

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved			BUFEMPTYIF	BUFFULLIF	STSDONEIF	ERRIF	STALLIF
7	6	5	4	3	2	1	0
NAKIF	RXPKIF	TXPKIF	PINGIF	INTKIF	OUTTKIF	SETUPPKIF	SETUPTKIF

Bits	Description	
[31:13]	**Reserved**	Reserved.
[12]	**BUFEMPTYIF**	**Buffer Empty Interrupt** 0 = The control-endpoint buffer is not empty. 1 = The control-endpoint buffer is empty. **Note:** Write 1 to clear this bit to 0.
[11]	**BUFFULLIF**	**Buffer Full Interrupt** 0 = The control-endpoint buffer is not full. 1 = The control-endpoint buffer is full. **Note:** Write 1 to clear this bit to 0.
[10]	**STSDONEIF**	**Status Completion Interrupt** 0 = Not a USB transaction has completed successfully. 1 = The status stage of a USB transaction has completed successfully. **Note:** Write 1 to clear this bit to 0.
[9]	**ERRIF**	**USB Error Interrupt** 0 = No error had occurred during the transaction. 1 = An error had occurred during the transaction. **Note:** Write 1 to clear this bit to 0.
[8]	**STALLIF**	**STALL Sent Interrupt** 0 = Not a stall-token is sent in response to an IN/OUT token. 1 = A stall-token is sent in response to an IN/OUT token. **Note:** Write 1 to clear this bit to 0.

[7]	**NAKIF**	**NAK Sent Interrupt** 0 = Not a NAK-token is sent in response to an IN/OUT token. 1 = A NAK-token is sent in response to an IN/OUT token. **Note:** Write 1 to clear this bit to 0.
[6]	**RXPKIF**	**Data Packet Received Interrupt** 0 = Not a data packet is successfully received from the host for an OUT-token and an ACK is sent to the host. 1 = A data packet is successfully received from the host for an OUT-token and an ACK is sent to the host. **Note:** Write 1 to clear this bit to 0.
[5]	**TXPKIF**	**Data Packet Transmitted Interrupt** 0 = Not a data packet is successfully transmitted to the host in response to an IN-token and an ACK-token is received for the same. 1 = A data packet is successfully transmitted to the host in response to an IN-token and an ACK-token is received for the same. **Note:** Write 1 to clear this bit to 0.
[4]	**PINGIF**	**Ping Token Interrupt** 0 = The control-endpoint does not received a ping token from the host. 1 = The control-endpoint receives a ping token from the host. **Note:** Write 1 to clear this bit to 0.
[3]	**INTKIF**	**In Token Interrupt** 0 = The control-endpoint does not received an IN token from the host. 1 = The control-endpoint receives an IN token from the host. **Note:** Write 1 to clear this bit to 0.
[2]	**OUTTKIF**	**Out Token Interrupt** 0 = The control-endpoint does not received an OUT token from the host. 1 = The control-endpoint receives an OUT token from the host. **Note:** Write 1 to clear this bit to 0.
[1]	**SETUPPKIF**	**Setup Packet Interrupt** This bit must be cleared (by writing 1) before the next setup packet can be received. If the bit is not cleared, then the successive setup packets will be overwritten in the setup packet buffer. 0 = Not a Setup packet has been received from the host. 1 = A Setup packet has been received from the host. **Note:** Write 1 to clear this bit to 0.
[0]	**SETUPTKIF**	**Setup Token Interrupt** 0 = Not a Setup token is received. 1 = A Setup token is received. Writing 1 clears this status bit **Note:** Write 1 to clear this bit to 0.

Control-Endpoint In-transfer Data Count (USBD_CEPTXCNT)

Register	Offset	R/W	Description	Reset Value
USBD_CEPTXCNT	USBD_BA+0x038	R/W	Control-Endpoint In-transfer Data Count	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TXCNT							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**TXCNT**	**In-Transfer Data Count** There is no mode selection for the control endpoint (but it operates like manual mode).The local-CPU has to fill the control-endpoint buffer with the data to be sent for an in-token and to write the count of bytes in this register. When zero is written into this field, a zero length packet is sent to the host. When the count written in the register is more than the MPS, the data sent will be of only MPS.

Control-Endpoint Out-transfer Data Count (USBD_CEPRXCNT)

Register	Offset	R/W	Description	Reset Value
USBD_CEPRXCNT	USBD_BA+0x03C	R	Control-Endpoint Out-transfer Data Count	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RXCNT							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**RXCNT**	**Out-Transfer Data Count** The USB device controller maintains the count of the data received in case of an out transfer, during the control transfer.

Control- Endpoint data count (USBD_CEPDATCNT)

Register	Offset	R/W	Description	Reset Value
USBD_CEPDATCNT	USBD_BA+0x040	R	Control-Endpoint data count	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DATCNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATCNT							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**DATCNT**	**Control-Endpoint Data Count** The USB device controller maintains the count of the data of control-endpoint.

Setup1 & Setup0 bytes (USBD_SETUP1_0)

Register	Offset	R/W	Description	Reset Value
USBD_SETUP1_0	USBD_BA+0x044	R	Setup1 & Setup0 bytes	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
SETUP1							
7	6	5	4	3	2	1	0
SETUP0							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:8]	**SETUP1**	**Setup Byte 1[15:8]** This register provides byte 1 of the last setup packet received. For a Standard Device Request, the following bRequest Code information is returned. 00000000 = Get Status. 00000001 = Clear Feature. 00000010 = Reserved. 00000011 = Set Feature. 00000100 = Reserved. 00000101 = Set Address. 00000110 = Get Descriptor. 00000111 = Set Descriptor. 00001000 = Get Configuration. 00001001 = Set Configuration. 00001010 = Get Interface. 00001011 = Set Interface. 00001100 = Synch Frame.

[7:0]	**SETUP0**	**Setup Byte 0[7:0]** This register provides byte 0 of the last setup packet received. For a Standard Device Request, the following bmRequestType information is returned. Bit 7(Direction): 0: Host to device 1: Device to host Bit 6-5 (Type): 00: Standard 01: Class 10: Vendor 11: Reserved Bit 4-0 (Recipient) 00000**:** Device 00001: Interface 00010: Endpoint 00011: Other Others: Reserved

Setup3 & Setup2 Bytes (USBD_SETUP3_2)

Register	Offset	R/W	Description	Reset Value
USBD_SETUP3_2	USBD_BA+0x048	R	Setup3 & Setup2 Bytes	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
SETUP3							
7	6	5	4	3	2	1	0
SETUP2							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:8]	**SETUP3**	**Setup Byte 3 [15:8]** This register provides byte 3 of the last setup packet received. For a Standard Device Request, the most significant byte of the wValue field is returned.
[7:0]	**SETUP2**	**Setup Byte 2 [7:0]** This register provides byte 2 of the last setup packet received. For a Standard Device Request, the least significant byte of the wValue field is returned.

Setup5 & Setup4 Bytes (USBD_SETUP5_4)

Register	Offset	R/W	Description	Reset Value
USBD_SETUP5_4	USBD_BA+0x04C	R	Setup5 & Setup4 Bytes	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SETUP5							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SETUP4							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:8]	**SETUP5**	**Setup Byte 5[15:8]** This register provides byte 5 of the last setup packet received. For a Standard Device Request, the most significant byte of the wIndex field is returned.
[7:0]	**SETUP4**	**Setup Byte 4[7:0]** This register provides byte 4 of the last setup packet received. For a Standard Device Request, the least significant byte of the wIndex is returned.

Setup7 & Setup6 bytes (USBD_SETUP7_6)

Register	Offset	R/W	Description	Reset Value
USBD_SETUP7_6	USBD_BA+0x050	R	Setup7 & Setup6 Bytes	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SETUP7							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SETUP6							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:8]	**SETUP7**	**Setup Byte 7[15:8]** This register provides byte 7 of the last setup packet received. For a Standard Device Request, the most significant byte of the wLength field is returned.
[7:0]	**SETUP6**	**Setup Byte 6[7:0]** This register provides byte 6 of the last setup packet received. For a Standard Device Request, the least significant byte of the wLength field is returned.

Control Endpoint RAM Start Address Register (USBD_CEPBUFSTART)

Register	Offset	R/W	Description	Reset Value
USBD_CEPBUFSTART	USBD_BA+0x054	R/W	Control Endpoint RAM Start Address Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**SADDR**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SADDR							

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11:0]	**SADDR**	**Control-Endpoint Start Address** This is the start-address of the RAM space allocated for the control-endpoint.

Control Endpoint RAM End Address Register (USBD_CEPBUFEND)

Register	Offset	R/W	Description	Reset Value
USBD_CEPBUFEND	USBD_BA+0x058	R/W	Control Endpoint RAM End Address Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**EADDR**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EADDR							

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11:0]	**EADDR**	**Control-Endpoint End Address** This is the end-address of the RAM space allocated for the control-endpoint.

DMA Control Status Register (USBD_DMACTL)

Register	Offset	R/W	Description	Reset Value
USBD_DMACTL	USBD_BA+0x05C	R/W	DMA Control Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
DMARST	SGEN	DMAEN	DMARD	EPNUM			

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**DMARST**	**Reset DMA State Machine** 0 = No reset the DMA state machine. 1 = Reset the DMA state machine.
[6]	**SGEN**	**Scatter Gather Function Enable Bit** 0 = Scatter gather function Disabled. 1 = Scatter gather function Enabled.
[5]	**DMAEN**	**DMA Enable Bit** 0 = DMA function Disabled. 1 = DMA function Enabled.
[4]	**DMARD**	**DMA Operation** 0 = The operation is a DMA write (read from USB buffer). DMA will check endpoint data available count (USBD_EPxDATCNT) according to EPNM setting before to perform DMA write operation. 1 = The operation is a DMA read (write to USB buffer).
[3:0]	**EPNUM**	**DMA Endpoint Address Bits** Used to define the Endpoint Address

DMA Count Register (USBD_DMACNT)

Register	Offset	R/W	Description	Reset Value
USBD_DMACNT	USBD_BA+0x060	R/W	DMA Count Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				**DMACNT**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMACNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMACNT							

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19:0]	**DMACNT**	**DMA Transfer Count** The transfer count of the DMA operation to be performed is written to this register.

Endpoint A~L Data Register (USBD_EPADAT~ USBD_EPLDAT)

Register	Offset	R/W	Description	Reset Value
USBD_EPADAT	USBD_BA+0x064	R/W	Endpoint A Data Register	0x0000_0000
USBD_EPBDAT	USBD_BA+0x08C	R/W	Endpoint B Data Register	0x0000_0000
USBD_EPCDAT	USBD_BA+0x0B4	R/W	Endpoint C Data Register	0x0000_0000
USBD_EPDDAT	USBD_BA+0x0DC	R/W	Endpoint D Data Register	0x0000_0000
USBD_EPEDAT	USBD_BA+0x104	R/W	Endpoint E Data Register	0x0000_0000
USBD_EPFDAT	USBD_BA+0x12C	R/W	Endpoint F Data Register	0x0000_0000
USBD_EPGDAT	USBD_BA+0x154	R/W	Endpoint G Data Register	0x0000_0000
USBD_EPHDAT	USBD_BA+0x17C	R/W	Endpoint H Data Register	0x0000_0000
USBD_EPIDAT	USBD_BA+0x1A4	R/W	Endpoint I Data Register	0x0000_0000
USBD_EPJDAT	USBD_BA+0x1CC	R/W	Endpoint J Data Register	0x0000_0000
USBD_EPKDAT	USBD_BA+0x1F4	R/W	Endpoint K Data Register	0x0000_0000
USBD_EPLDAT	USBD_BA+0x21C	R/W	Endpoint L Data Register	0x0000_0000

31	30	29	28	27	26	25	24
EPDAT							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
EPDAT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
EPDAT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EPDAT							

Bits	Description	
[31:0]	**EPDAT**	**Endpoint A~L Data Register** Endpoint A~L data buffer for the buffer transaction (read or write). **Note:** Only word or byte access are supported.

Endpoint A~L Interrupt Status Register (USBD_EPAINTSTS~ USBD_EPLINTSTS)

Register	Offset	R/W	Description	Reset Value
USBD_EPAINTSTS	USBD_BA+0x068	R/W	Endpoint A Interrupt Status Register	0x0000_0003
USBD_EPBINTSTS	USBD_BA+0x090	R/W	Endpoint B Interrupt Status Register	0x0000_0003
USBD_EPCINTSTS	USBD_BA+0x0B8	R/W	Endpoint C Interrupt Status Register	0x0000_0003
USBD_EPDINTSTS	USBD_BA+0x0E0	R/W	Endpoint D Interrupt Status Register	0x0000_0003
USBD_EPEINTSTS	USBD_BA+0x108	R/W	Endpoint E Interrupt Status Register	0x0000_0003
USBD_EPFINTSTS	USBD_BA+0x130	R/W	Endpoint F Interrupt Status Register	0x0000_0003
USBD_EPGINTSTS	USBD_BA+0x158	R/W	Endpoint G Interrupt Status Register	0x0000_0003
USBD_EPHINTSTS	USBD_BA+0x180	R/W	Endpoint H Interrupt Status Register	0x0000_0003
USBD_EPIINTSTS	USBD_BA+0x1A8	R/W	Endpoint I Interrupt Status Register	0x0000_0003
USBD_EPJINTSTS	USBD_BA+0x1D0	R/W	Endpoint J Interrupt Status Register	0x0000_0003
USBD_EPKINTSTS	USBD_BA+0x1F8	R/W	Endpoint K Interrupt Status Register	0x0000_0003
USBD_EPLINTSTS	USBD_BA+0x220	R/W	Endpoint L Interrupt Status Register	0x0000_0003

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			**SHORTRXIF**	**ERRIF**	**NYETIF**	**STALLIF**	**NAKIF**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PINGIF	**INTKIF**	**OUTTKIF**	**RXPKIF**	**TXPKIF**	**SHORTTXIF**	**BUFEMPTYIF**	**BUFFULLIF**

Bits	Description	
[31:13]	**Reserved**	Reserved.
[12]	**SHORTRXIF**	**Bulk Out Short Packet Received** 0 = No bulk out short packet is received. 1 = Received bulk out short packet (including zero length packet). **Note:** Write 1 to clear this bit to 0.
[11]	**ERRIF**	**ERR Sent** 0 = No any error in the transaction. 1 = There occurs any error in the transaction. **Note:** Write 1 to clear this bit to 0.

[10]	**NYETIF**	**NYET Sent** 0 = The space available in the RAM is sufficient to accommodate the next on coming data packet. 1 = The space available in the RAM is not sufficient to accommodate the next on coming data packet. **Note:** Write 1 to clear this bit to 0.
[9]	**STALLIF**	**USB STALL Sent** 0 = The last USB packet could be accepted or provided because the endpoint was stalled, and was acknowledged with a STALL. 1 = The last USB packet could not be accepted or provided because the endpoint was stalled, and was acknowledged with a STALL. **Note:** Write 1 to clear this bit to 0.
[8]	**NAKIF**	**USB NAK Sent** 0 = The last USB IN packet could be provided, and was acknowledged with an ACK. 1 = The last USB IN packet could not be provided, and was acknowledged with a NAK. **Note:** Write 1 to clear this bit to 0.
[7]	**PINGIF**	**PING Token Interrupt** 0 = A Data PING token has not been received from the host. 1 = A Data PING token has been received from the host. **Note:** Write 1 to clear this bit to 0.
[6]	**INTKIF**	**Data IN Token Interrupt** 0 = Not Data IN token has been received from the host. 1 = A Data IN token has been received from the host. **Note:** Write 1 to clear this bit to 0.
[5]	**OUTTKIF**	**Data OUT Token Interrupt** 0 = A Data OUT token has not been received from the host. 1 = A Data OUT token has been received from the host. This bit also set by PING token (in high-speed only). **Note:** Write 1 to clear this bit to 0.
[4]	**RXPKIF**	**Data Packet Received Interrupt** 0 = No data packet is received from the host by the endpoint. 1 = A data packet is received from the host by the endpoint. **Note:** Write 1 to clear this bit to 0.
[3]	**TXPKIF**	**Data Packet Transmitted Interrupt** 0 = Not a data packet is transmitted from the endpoint to the host. 1 = A data packet is transmitted from the endpoint to the host. **Note:** Write 1 to clear this bit to 0.
[2]	**SHORTTXIF**	**Short Packet Transferred Interrupt** 0 = The length of the last packet was not less than the Maximum Packet Size (EPMPS). 1 = The length of the last packet was less than the Maximum Packet Size (EPMPS). **Note:** Write 1 to clear this bit to 0.

[1]	**BUFEMPTYIF**	**Buffer Empty** For an IN endpoint, a buffer is available to the local side for writing up to FIFO full of bytes. 0 = The endpoint buffer is not empty. 1 = The endpoint buffer is empty. For an OUT endpoint: 0 = The currently selected buffer has not a count of 0. 1 = The currently selected buffer has a count of 0, or no buffer is available on the local side (nothing to read). **Note:** This bit is read-only.
[0]	**BUFFULLIF**	**Buffer Full** For an IN endpoint, the currently selected buffer is full, or no buffer is available to the local side for writing (no space to write). For an OUT endpoint, there is a buffer available on the local side, and there are FIFO full of bytes available to be read (entire packet is available for reading). 0 = The endpoint packet buffer is not full. 1 = The endpoint packet buffer is full. **Note:** This bit is read-only.

Endpoint A~L Interrupt Enable Control Register (USBD_EPAINTEN~ USBD_EPLINTEN)

Register	Offset	R/W	Description	Reset Value
USBD_EPAINTEN	USBD_BA+0x06C	R/W	Endpoint A Interrupt Enable Register	0x0000_0000
USBD_EPBINTEN	USBD_BA+0x094	R/W	Endpoint B Interrupt Enable Register	0x0000_0000
USBD_EPCINTEN	USBD_BA+0x0BC	R/W	Endpoint C Interrupt Enable Register	0x0000_0000
USBD_EPDINTEN	USBD_BA+0x0E4	R/W	Endpoint D Interrupt Enable Register	0x0000_0000
USBD_EPEINTEN	USBD_BA+0x10C	R/W	Endpoint E Interrupt Enable Register	0x0000_0000
USBD_EPFINTEN	USBD_BA+0x134	R/W	Endpoint F Interrupt Enable Register	0x0000_0000
USBD_EPGINTEN	USBD_BA+0x15C	R/W	Endpoint G Interrupt Enable Register	0x0000_0000
USBD_EPHINTEN	USBD_BA+0x184	R/W	Endpoint H Interrupt Enable Register	0x0000_0000
USBD_EPIINTEN	USBD_BA+0x1AC	R/W	Endpoint I Interrupt Enable Register	0x0000_0000
USBD_EPJINTEN	USBD_BA+0x1D4	R/W	Endpoint J Interrupt Enable Register	0x0000_0000
USBD_EPKINTEN	USBD_BA+0x1FC	R/W	Endpoint K Interrupt Enable Register	0x0000_0000
USBD_EPLINTEN	USBD_BA+0x224	R/W	Endpoint L Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			**SHORTRXIEN**	**ERRIEN**	**NYETIEN**	**STALLIEN**	**NAKIEN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PINGIEN	**INTKIEN**	**OUTTKIEN**	**RXPKIEN**	**TXPKIEN**	**SHORTTXIEN**	**BUFEMPTYIEN**	**BUFFULLIEN**

Bits	Description	
[31:13]	**Reserved**	Reserved.
[12]	**SHORTRXIEN**	**Bulk Out Short Packet Interrupt Enable Bit** When set, this bit enables a local interrupt to be set whenever bulk out short packet occurs on the bus for this endpoint. 0 = Bulk out interrupt Disabled. 1 = Bulk out interrupt Enabled.

[11]	**ERRIEN**	**ERR Interrupt Enable Bit** When set, this bit enables a local interrupt to be set whenever ERR condition occurs on the bus for this endpoint. 0 = Error event interrupt Disabled. 1 = Error event interrupt Enabled.
[10]	**NYETIEN**	**NYET Interrupt Enable Bit** When set, this bit enables a local interrupt to be set whenever NYET condition occurs on the bus for this endpoint. 0 = NYET condition interrupt Disabled. 1 = NYET condition interrupt Enabled.
[9]	**STALLIEN**	**USB STALL Sent Interrupt Enable Bit** When set, this bit enables a local interrupt to be set when a stall token is sent to the host. 0 = STALL token interrupt Disabled. 1 = STALL token interrupt Enabled.
[8]	**NAKIEN**	**USB NAK Sent Interrupt Enable Bit** When set, this bit enables a local interrupt to be set when a NAK token is sent to the host. 0 = NAK token interrupt Disabled. 1 = NAK token interrupt Enabled.
[7]	**PINGIEN**	**PING Token Interrupt Enable Bit** When set, this bit enables a local interrupt to be set when a PING token has been received from the host. 0 = PING token interrupt Disabled. 1 = PING token interrupt Enabled.
[6]	**INTKIEN**	**Data IN Token Interrupt Enable Bit** When set, this bit enables a local interrupt to be set when a Data IN token has been received from the host. 0 = Data IN token interrupt Disabled. 1 = Data IN token interrupt Enabled.
[5]	**OUTTKIEN**	**Data OUT Token Interrupt Enable Bit** When set, this bit enables a local interrupt to be set when a Data OUT token has been received from the host. 0 = Data OUT token interrupt Disabled. 1 = Data OUT token interrupt Enabled.
[4]	**RXPKIEN**	**Data Packet Received Interrupt Enable Bit** When set, this bit enables a local interrupt to be set when a data packet has been transmitted to the host. 0 = Data packet has been transmitted to the host interrupt Disabled. 1 = Data packet has been transmitted to the host interrupt Enabled.
[3]	**TXPKIEN**	**Data Packet Transmitted Interrupt Enable Bit** When set, this bit enables a local interrupt to be set when a data packet has been received from the host. 0 = Data packet has been received from the host interrupt Disabled. 1 = Data packet has been received from the host interrupt Enabled.

[2]	**SHORTTXIEN**	**Short Packet Transferred Interrupt Enable Bit** When set, this bit enables a local interrupt to be set when a short data packet has been transferred to/from the host. 0 = Short data packet interrupt Disabled. 1 = Short data packet interrupt Enabled.
[1]	**BUFEMPTYIEN**	**Buffer Empty Interrupt** When set, this bit enables a local interrupt to be set when a buffer empty condition is detected on the bus. 0 = Buffer empty interrupt Disabled. 1 = Buffer empty interrupt Enabled.
[0]	**BUFFULLIEN**	**Buffer Full Interrupt** When set, this bit enables a local interrupt to be set when a buffer full condition is detected on the bus. 0 = Buffer full interrupt Disabled. 1 = Buffer full interrupt Enabled.

Endpoint A~L Data Available Count Register (USBD_EPADATCNT~ USBD_EPLDATCNT)

Register	Offset	R/W	Description	Reset Value
USBD_EPADATCNT	USBD_BA+0x070	R	Endpoint A Data Available Count Register	0x0000_0000
USBD_EPBDATCNT	USBD_BA+0x098	R	Endpoint B Data Available Count Register	0x0000_0000
USBD_EPCDATCNT	USBD_BA+0x0C0	R	Endpoint C Data Available Count Register	0x0000_0000
USBD_EPDDATCNT	USBD_BA+0x0E8	R	Endpoint D Data Available Count Register	0x0000_0000
USBD_EPEDATCNT	USBD_BA+0x110	R	Endpoint E Data Available Count Register	0x0000_0000
USBD_EPFDATCNT	USBD_BA+0x138	R	Endpoint F Data Available Count Register	0x0000_0000
USBD_EPGDATCNT	USBD_BA+0x160	R	Endpoint G Data Available Count Register	0x0000_0000
USBD_EPHDATCNT	USBD_BA+0x188	R	Endpoint H Data Available Count Register	0x0000_0000
USBD_EPIDATCNT	USBD_BA+0x1B0	R	Endpoint I Data Available Count Register	0x0000_0000
USBD_EPJDATCNT	USBD_BA+0x1D8	R	Endpoint J Data Available Count Register	0x0000_0000
USBD_EPKDATCNT	USBD_BA+0x200	R	Endpoint K Data Available Count Register	0x0000_0000
USBD_EPLDATCNT	USBD_BA+0x228	R	Endpoint L Data Available Count Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	**DMALOOP**						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMALOOP							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DATCNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATCNT							

Bits	Description	
[31]	**Reserved**	Reserved.
[30:16]	**DMALOOP**	**DMA Loop** This register is the remaining DMA loop to complete. Each loop means 32-byte transfer.

[15:0]	**DATCNT**	**Data Count** For an IN endpoint (EPDIR(USBD_EPxCFG[3] is high.), this register returns the number of valid bytes in the IN endpoint packet buffer. For an OUT endpoint (EPDIR(USBD_EPxCFG[3] is low.), this register returns the number of received valid bytes in the Host OUT transfer.

Endpoint A~L Response Control Register (USBD_EPARSPCTL~ USBD_EPLRSPCTL)

Register	Offset	R/W	Description	Reset Value
USBD_EPARSPCTL	USBD_BA+0x074	R/W	Endpoint A Response Control Register	0x0000_0000
USBD_EPBRSPCTL	USBD_BA+0x09C	R/W	Endpoint B Response Control Register	0x0000_0000
USBD_EPCRSPCTL	USBD_BA+0x0C4	R/W	Endpoint C Response Control Register	0x0000_0000
USBD_EPDRSPCTL	USBD_BA+0x0EC	R/W	Endpoint D Response Control Register	0x0000_0000
USBD_EPERSPCTL	USBD_BA+0x114	R/W	Endpoint E Response Control Register	0x0000_0000
USBD_EPFRSPCTL	USBD_BA+0x13C	R/W	Endpoint F Response Control Register	0x0000_0000
USBD_EPGRSPCTL	USBD_BA+0x164	R/W	Endpoint G Response Control Register	0x0000_0000
USBD_EPHRSPCTL	USBD_BA+0x18C	R/W	Endpoint H Response Control Register	0x0000_0000
USBD_EPIRSPCTL	USBD_BA+0x1B4	R/W	Endpoint I Response Control Register	0x0000_0000
USBD_EPJRSPCTL	USBD_BA+0x1DC	R/W	Endpoint J Response Control Register	0x0000_0000
USBD_EPKRSPCTL	USBD_BA+0x204	R/W	Endpoint K Response Control Register	0x0000_0000
USBD_EPLRSPCTL	USBD_BA+0x22C	R/W	Endpoint L Response Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
DISBUF	SHORTTXEN	ZEROLEN	HALT	TOGGLE	MODE		FLUSH

Bits	Description	
[31:8]	Reserved	Reserved.

[7]	**DISBUF**	**Buffer Disable Bit** This bit is used to receive unknown size OUT short packet. The received packet size is reference USBD_EPxDATCNT register. 0 = Buffer Not Disabled when Bulk-OUT short packet is received. 1 = Buffer Disabled when Bulk-OUT short packet is received.
[6]	**SHORTTXEN**	**Short Packet Transfer Enable** This bit is applicable only in case of Auto-Validate Method. This bit is set to validate any remaining data in the buffer which is not equal to the MPS of the endpoint, and happens to be the last transfer. This bit gets cleared once the data packet is sent. 0 = Not validate any remaining data in the buffer which is not equal to the MPS of the endpoint. 1 = Validate any remaining data in the buffer which is not equal to the MPS of the endpoint.
[5]	**ZEROLEN**	**Zero Length** This bit is used to send a zero-length packet response to an IN-token. When this bit is set, a zero packet is sent to the host on reception of an IN-token. This bit gets cleared once the zero length data packet is sent. 0 = A zero packet is not sent to the host on reception of an IN-token. 1 = A zero packet is sent to the host on reception of an IN-token.
[4]	**HALT**	**Endpoint Halt** This bit is used to send a STALL handshake as response to the token from the host. When an Endpoint Set Feature (ep_halt) is detected by the local CPU, it must write a '1' to this bit. 0 = Not send a STALL handshake as response to the token from the host. 1 = Send a STALL handshake as response to the token from the host.
[3]	**TOGGLE**	**Endpoint Toggle** This bit is used to clear the endpoint data toggle bit. Reading this bit returns the current state of the endpoint data toggle bit. The local CPU may use this bit to initialize the end-point's toggle in case of reception of a Set Interface request or a Clear Feature (ep_halt) request from the host. Only when toggle bit is "1", this bit can be written into the inversed write data bit[3]. 0 = Not clear the endpoint data toggle bit. 1 = Clear the endpoint data toggle bit.
[2:1]	**MODE**	**Mode Control** The two bits decide the operation mode of the in-endpoint. 00: Auto-Validate Mode 01: Manual-Validate Mode 10: Fly Mode 11: Reserved These bits are not valid for an out-endpoint. The auto validate mode will be activated when the reserved mode is selected.
[0]	**FLUSH**	**Buffer Flush** Writing 1 to this bit causes the packet buffer to be flushed and the corresponding EP_AVAIL register to be cleared. This bit is self-clearing. This bit should always be written after an configuration event. 0 = The packet buffer is not flushed. 1 = The packet buffer is flushed by user.

Endpoint A~L Maximum Packet Size Register (USBD_EPAMPS~ USBD_EPLMPS)

Register	Offset	R/W	Description	Reset Value
USBD_EPAMPS	USBD_BA+0x078	R/W	Endpoint A Maximum Packet Size Register	0x0000_0000
USBD_EPBMPS	USBD_BA+0x0A0	R/W	Endpoint B Maximum Packet Size Register	0x0000_0000
USBD_EPCMPS	USBD_BA+0x0C8	R/W	Endpoint C Maximum Packet Size Register	0x0000_0000
USBD_EPDMPS	USBD_BA+0x0F0	R/W	Endpoint D Maximum Packet Size Register	0x0000_0000
USBD_EPEMPS	USBD_BA+0x118	R/W	Endpoint E Maximum Packet Size Register	0x0000_0000
USBD_EPFMPS	USBD_BA+0x140	R/W	Endpoint F Maximum Packet Size Register	0x0000_0000
USBD_EPGMPS	USBD_BA+0x168	R/W	Endpoint G Maximum Packet Size Register	0x0000_0000
USBD_EPHMPS	USBD_BA+0x190	R/W	Endpoint H Maximum Packet Size Register	0x0000_0000
USBD_EPIMPS	USBD_BA+0x1B8	R/W	Endpoint I Maximum Packet Size Register	0x0000_0000
USBD_EPJMPS	USBD_BA+0x1E0	R/W	Endpoint J Maximum Packet Size Register	0x0000_0000
USBD_EPKMPS	USBD_BA+0x208	R/W	Endpoint K Maximum Packet Size Register	0x0000_0000
USBD_EPLMPS	USBD_BA+0x230	R/W	Endpoint L Maximum Packet Size Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**EPMPS**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EPMPS							

Bits	Description	
[31:11]	**Reserved**	Reserved.
[10:0]	**EPMPS**	**Endpoint Maximum Packet Size** This field determines the Maximum Packet Size of the Endpoint.

Endpoint A~L Transfer Count Register (USBD_EPATXCNT~ USBD_EPLTXCNT)

Register	Offset	R/W	Description	Reset Value
USBD_EPATXCNT	USBD_BA+0x07C	R/W	Endpoint A Transfer Count Register	0x0000_0000
USBD_EPBTXCNT	USBD_BA+0x0A4	R/W	Endpoint B Transfer Count Register	0x0000_0000
USBD_EPCTXCNT	USBD_BA+0x0CC	R/W	Endpoint C Transfer Count Register	0x0000_0000
USBD_EPDTXCNT	USBD_BA+0x0F4	R/W	Endpoint D Transfer Count Register	0x0000_0000
USBD_EPETXCNT	USBD_BA+0x11C	R/W	Endpoint E Transfer Count Register	0x0000_0000
USBD_EPFTXCNT	USBD_BA+0x144	R/W	Endpoint F Transfer Count Register	0x0000_0000
USBD_EPGTXCNT	USBD_BA+0x16C	R/W	Endpoint G Transfer Count Register	0x0000_0000
USBD_EPHTXCNT	USBD_BA+0x194	R/W	Endpoint H Transfer Count Register	0x0000_0000
USBD_EPITXCNT	USBD_BA+0x1BC	R/W	Endpoint I Transfer Count Register	0x0000_0000
USBD_EPJTXCNT	USBD_BA+0x1E4	R/W	Endpoint J Transfer Count Register	0x0000_0000
USBD_EPKTXCNT	USBD_BA+0x20C	R/W	Endpoint K Transfer Count Register	0x0000_0000
USBD_EPLTXCNT	USBD_BA+0x234	R/W	Endpoint L Transfer Count Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
					TXCNT		
7	6	5	4	3	2	1	0
TXCNT							

Bits	Description	
[31:11]	**Reserved**	Reserved.

[10:0]	**TXCNT**	**Endpoint Transfer Count** For IN endpoints, this field determines the total number of bytes to be sent to the host in case of manual validation method. For OUT endpoints, this field has no effect.

Endpoint A~L Configuration Register (USBD_EPACFG~ USBD_EPLCFG)

Register	Offset	R/W	Description	Reset Value
USBD_EPACFG	USBD_BA+0x080	R/W	Endpoint A Configuration Register	0x0000_0012
USBD_EPBCFG	USBD_BA+0x0A8	R/W	Endpoint B Configuration Register	0x0000_0022
USBD_EPCCFG	USBD_BA+0x0D0	R/W	Endpoint C Configuration Register	0x0000_0032
USBD_EPDCFG	USBD_BA+0x0F8	R/W	Endpoint D Configuration Register	0x0000_0042
USBD_EPECFG	USBD_BA+0x120	R/W	Endpoint E Configuration Register	0x0000_0052
USBD_EPFCFG	USBD_BA+0x148	R/W	Endpoint F Configuration Register	0x0000_0062
USBD_EPGCFG	USBD_BA+0x170	R/W	Endpoint G Configuration Register	0x0000_0072
USBD_EPHCFG	USBD_BA+0x198	R/W	Endpoint H Configuration Register	0x0000_0082
USBD_EPICFG	USBD_BA+0x1C0	R/W	Endpoint I Configuration Register	0x0000_0092
USBD_EPJCFG	USBD_BA+0x1E8	R/W	Endpoint J Configuration Register	0x0000_00A2
USBD_EPKCFG	USBD_BA+0x210	R/W	Endpoint K Configuration Register	0x0000_00B2
USBD_EPLCFG	USBD_BA+0x238	R/W	Endpoint L Configuration Register	0x0000_00C2

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**Reserved**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EPNUM				**EPDIR**	**EPTYPE**		**EPEN**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:4]	**EPNUM**	**Endpoint Number** This field selects the number of the endpoint. Valid numbers 1 to 15. **Note**: Do not support two endpoints have same endpoint number.

[3]	**EPDIR**	**Endpoint Direction** 0 = out-endpoint (Host OUT to Device). 1 = in-endpoint (Host IN to Device). **Note:** A maximum of one OUT and IN endpoint is allowed for each endpoint number.
[2:1]	**EPTYPE**	**Endpoint Type** This field selects the type of this endpoint. Endpoint 0 is forced to a Control type. 00 = Reserved. 01 = Bulk. 10 = Interrupt. 11 = Isochronous.
[0]	**EPEN**	**Endpoint Valid** When set, this bit enables this endpoint. This bit has no effect on Endpoint 0, which is always enabled. 0 = The endpoint Disabled. 1 = The endpoint Enabled.

Endpoint A~L RAM Start Address Register (USBD_EPABUFSTART~ USBD_EPLBUFSTART)

Register	Offset	R/W	Description	Reset Value
USBD_EPABUFSTART	USBD_BA+0x084	R/W	Endpoint A RAM Start Address Register	0x0000_0000
USBD_EPBBUFSTART	USBD_BA+0x0AC	R/W	Endpoint B RAM Start Address Register	0x0000_0000
USBD_EPCBUFSTART	USBD_BA+0x0D4	R/W	Endpoint C RAM Start Address Register	0x0000_0000
USBD_EPDBUFSTART	USBD_BA+0x0FC	R/W	Endpoint D RAM Start Address Register	0x0000_0000
USBD_EPEBUFSTART	USBD_BA+0x124	R/W	Endpoint E RAM Start Address Register	0x0000_0000
USBD_EPFBUFSTART	USBD_BA+0x14C	R/W	Endpoint F RAM Start Address Register	0x0000_0000
USBD_EPGBUFSTART	USBD_BA+0x174	R/W	Endpoint G RAM Start Address Register	0x0000_0000
USBD_EPHBUFSTART	USBD_BA+0x19C	R/W	Endpoint H RAM Start Address Register	0x0000_0000
USBD_EPIBUFSTART	USBD_BA+0x1C4	R/W	Endpoint I RAM Start Address Register	0x0000_0000
USBD_EPJBUFSTART	USBD_BA+0x1EC	R/W	Endpoint J RAM Start Address Register	0x0000_0000
USBD_EPKBUFSTART	USBD_BA+0x214	R/W	Endpoint K RAM Start Address Register	0x0000_0000
USBD_EPLBUFSTART	USBD_BA+0x23C	R/W	Endpoint L RAM Start Address Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**SADDR**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SADDR							

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11:0]	**SADDR**	**Endpoint Start Address** This is the start-address of the RAM space allocated for the endpoint A~L.

Endpoint A~L RAM End Address Register (USBD_EPABUFEND~ USBD_EPLBUFEND)

Register	Offset	R/W	Description	Reset Value
USBD_EPABUFEND	USBD_BA+0x088	R/W	Endpoint A RAM End Address Register	0x0000_0000
USBD_EPBBUFEND	USBD_BA+0x0B0	R/W	Endpoint B RAM End Address Register	0x0000_0000
USBD_EPCBUFEND	USBD_BA+0x0D8	R/W	Endpoint C RAM End Address Register	0x0000_0000
USBD_EPDBUFEND	USBD_BA+0x100	R/W	Endpoint D RAM End Address Register	0x0000_0000
USBD_EPEBUFEND	USBD_BA+0x128	R/W	Endpoint E RAM End Address Register	0x0000_0000
USBD_EPFBUFEND	USBD_BA+0x150	R/W	Endpoint F RAM End Address Register	0x0000_0000
USBD_EPGBUFEND	USBD_BA+0x178	R/W	Endpoint G RAM End Address Register	0x0000_0000
USBD_EPHBUFEND	USBD_BA+0x1A0	R/W	Endpoint H RAM End Address Register	0x0000_0000
USBD_EPIBUFEND	USBD_BA+0x1C8	R/W	Endpoint I RAM End Address Register	0x0000_0000
USBD_EPJBUFEND	USBD_BA+0x1F0	R/W	Endpoint J RAM End Address Register	0x0000_0000
USBD_EPKBUFEND	USBD_BA+0x218	R/W	Endpoint K RAM End Address Register	0x0000_0000
USBD_EPLBUFEND	USBD_BA+0x240	R/W	Endpoint L RAM End Address Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**EADDR**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EADDR							

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11:0]	**EADDR**	**Endpoint End Address** This is the end-address of the RAM space allocated for the endpoint A~L.

AHB Address Register (USBD_DMAADDR)

Register	Offset	R/W	Description	Reset Value
USBD_DMAADDR	USBD_BA+0x700	R/W	AHB DMA Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMAADDR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMAADDR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMAADDR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAADDR							

Bits	Description	
[31:0]	**DMAADDR**	**DMAADDR** The register specifies the address from which the DMA has to read / write. The address must WORD (32-bit) aligned.

USB PHY Control Register (USBD_PHYCTL)

Register	Offset	R/W	Description	Reset Value
USBD_PHYCTL	USBD_BA+0x704	R/W	USB PHY Control Register	0x0000_0420

31	30	29	28	27	26	25	24
VBUSDET	**Reserved**						**WKEN**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**PHYEN**	**DPPUEN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31]	**VBUSDET**	**VBUS Status** 0 = The VBUS is not detected yet. 1 = The VBUS is detected.
[30:25]	**Reserved**	Reserved.
[24]	**WKEN**	**Wake-Up Enable Bit** 0 = The wake-up function Disabled. 1 = The wake-up function Enabled.
[23:10]	**Reserved**	Reserved.
[9]	**PHYEN**	**PHY Suspend Enable Bit** 0 = The USB PHY is suspend. 1 = The USB PHY is not suspend.
[8]	**DPPUEN**	**DP Pull-Up** 0 = Pull-up resistor on D+ Disabled. 1 = Pull-up resistor on D+ Enabled.
[7:0]	**Reserved**	Reserved.

6.33 USB 1.1 Host Controller (USBH)

6.33.1 Overview

This chip is equipped with a USB 1.1 Host Controller (USBH) that supports Open Host Controller Interface (OpenHCI, OHCI) Specification, a register-level description of a host controller, to manage the devices and data transfer of Universal Serial Bus (USB).

The USBH supports an integrated Root Hub with a USB port, a DMA for real-time data transfer between system memory and USB bus, port power control and port over current detection.

The USBH is responsible for detecting the connect and disconnect of USB devices, managing data transfer, collecting status and activity of USB bus, providing power control and detecting over current of attached USB devices.

6.33.2 Features

- Supports Universal Serial Bus (USB) Specification Revision 1.1.
- Supports Open Host Controller Interface (OpenHCI) Specification Revision 1.0.
- Supports both full-speed (12Mbps) and low-speed (1.5Mbps) USB devices.
- Supports Control, Bulk, Interrupt and Isochronous transfers.
- Supports an integrated Root Hub.
- Supports two USB host ports:
  - USB Host port 1 is shared with USB device (OTG function).
  - USB Host port 2 is an independent host port. The port 2 host function can work even port1 functioned used as an USB device.
- Supports port power control and port over current detection.
- Supports DMA for real-time data transfer.

6.33.3 Block Diagram

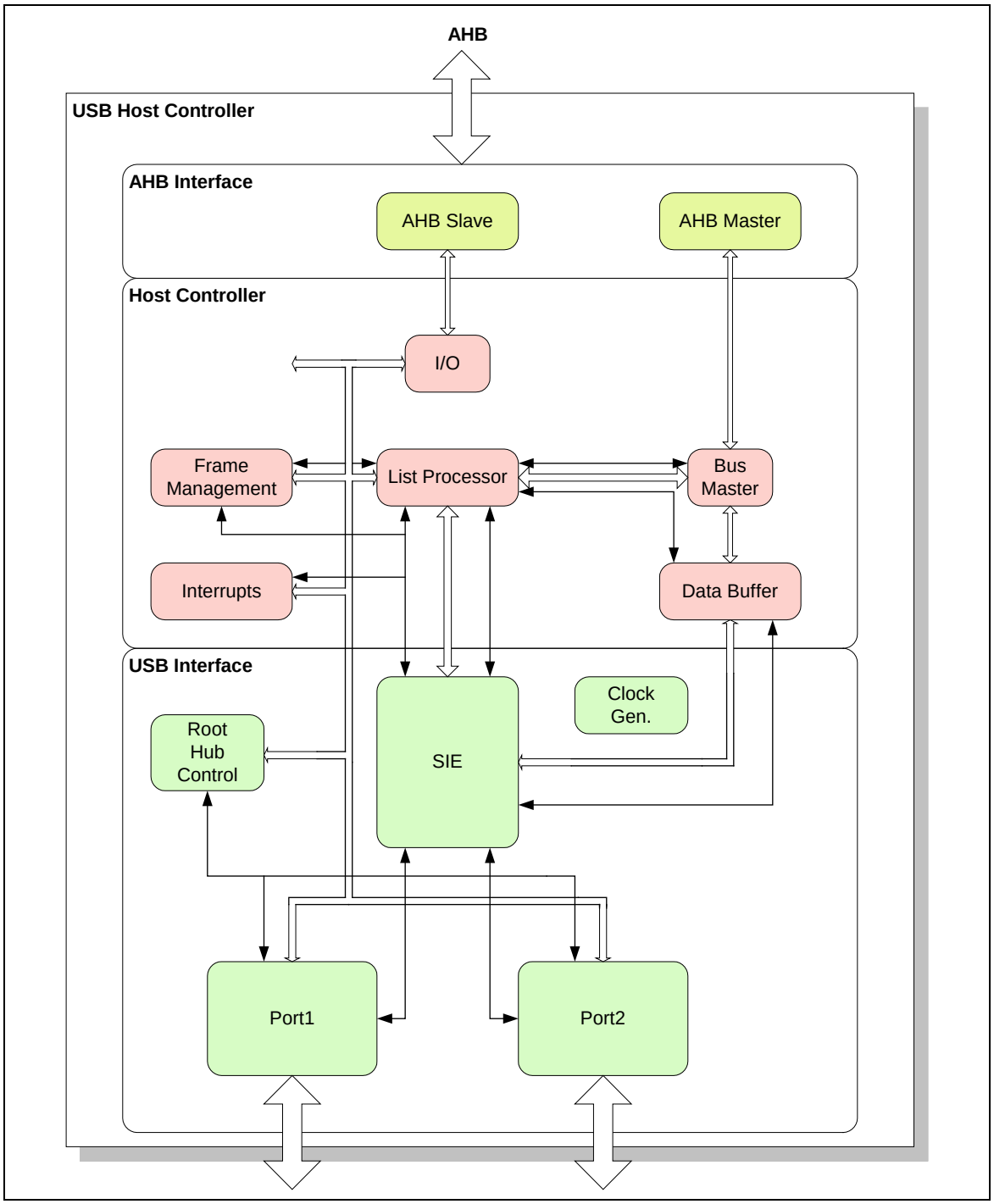


Figure 6.33-1 USB 1.1 Host Controller Block Diagram

6.33.4 Basic Configuration

The USBH clock source is derived from PLL. User has to set the PLL related configurations

before USB host controller is enabled. Set the USBHCKEN (CLK_AHBCLK[4]) bit to enable USBH clock and 4-bit pre-scaler USBDIV (CLK_CLKDIV0[7:4]) to generate the proper USBH clock rate. The proper USBH clock rate is 48 MHz.

6.33.5 Functional Description

6.33.5.1 AHB Interface

The OpenHCI Host Controller is connected to the system by the AHB bus. The design requires both master and slave bus operations. As a master, the Host Controller is responsible for running cycles on the AHB bus to access EDs and TDs as well as transferring data between memory and the local data buffer. As a slave, the Host Controller monitors the cycles on the AHB bus and determines when to respond to these cycles. Configuration and non-real-time control access to the Host Controller operational registers are through the AHB bus slave interface.

6.33.5.2 Host Controller

The host controller includes 5 functional blocks, including List Processing, Frame Management, Interrupt Processing, Host Controller Bus Master and Data Buffer.

The List Processor manages the data structures from the Host Controller Driver and coordinates all activity within the Host Controller.

The Frame Management is responsible for managing the frame specific tasks required by the USB specification and the OpenHCI specification. These tasks are:

1) Management of the OpenHCI frame specific Operational Registers
2) Operation of the Largest Data Packet Counter.
3) Performing frame qualifications on USB Transaction requests to the SIE.
4) Generate SOF token requests to the SIE.

Interrupts are the communication method for HC-initiated communication with the Host Controller Driver. There are several events that may trigger an interrupt from the Host Controller. Each specific event sets a specific bit in the HcInterruptStatus register.

The Host Controller Bus Master is the central block in the data path. The Host Controller Bus Master coordinates all access to the AHB Interface. There are two sources of bus mastering within Host Controller: the List Processor and the Data Buffer Engine.

The Data Buffer serves as the data interface between the Bus Master and the SIE. It is a combination of a 64-byte latched based bi-directional asynchronous FIFO and a single Dword AHB Holding Register.

6.33.5.3 USB Interface

The USB interface includes the integrated Root Hub with two external ports, Port 1 and Port 2 as well as the Serial Interface Engine (SIE) and USB clock generator. The interface combines responsibility for executing bus transactions requested by the HC as well as the hub and port management specified by USB.

The SIE is responsible for managing all transactions to the USB. It controls the bus protocol, packet generation/extraction, data parallel-to-serial conversion, CRC coding, bit stuffing, and NRZI encoding. All transactions on the USB are requested from the List Processor and Frame Manager.

The Root Hub is a collection of ports that are individually controlled and a hub that maintains control/status over functions common to all ports.

6.33.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
USBH Base Address: **USBH_BA = 0x4000_9000**				
HcRevision	USBH_BA+0x000	R	Host Controller Revision Register	0x0000_0110
HcControl	USBH_BA+0x004	R/W	Host Controller Control Register	0x0000_0000
HcCommandStatus	USBH_BA+0x008	R/W	Host Controller Command Status Register	0x0000_0000
HcInterruptStatus	USBH_BA+0x00C	R/W	Host Controller Interrupt Status Register	0x0000_0000
HcInterruptEnable	USBH_BA+0x010	R/W	Host Controller Interrupt Enable Control Register	0x0000_0000
HcInterruptDisable	USBH_BA+0x014	R/W	Host Controller Interrupt Disable Control Register	0x0000_0000
HcHCCA	USBH_BA+0x018	R/W	Host Controller Communication Area Register	0x0000_0000
HcPeriodCurrentED	USBH_BA+0x01C	R/W	Host Controller Period Current ED Register	0x0000_0000
HcControlHeadED	USBH_BA+0x020	R/W	Host Controller Control Head ED Register	0x0000_0000
HcControlCurrentED	USBH_BA+0x024	R/W	Host Controller Control Current ED Register	0x0000_0000
HcBulkHeadED	USBH_BA+0x028	R/W	Host Controller Bulk Head ED Register	0x0000_0000
HcBulkCurrentED	USBH_BA+0x02C	R/W	Host Controller Bulk Current ED Register	0x0000_0000
HcDoneHead	USBH_BA+0x030	R/W	Host Controller Done Head Register	0x0000_0000
HcFmInterval	USBH_BA+0x034	R/W	Host Controller Frame Interval Register	0x0000_2EDF
HcFmRemaining	USBH_BA+0x038	R	Host Controller Frame Remaining Register	0x0000_0000
HcFmNumber	USBH_BA+0x03C	R	Host Controller Frame Number Register	0x0000_0000
HcPeriodicStart	USBH_BA+0x040	R/W	Host Controller Periodic Start Register	0x0000_0000
HcLSThreshold	USBH_BA+0x044	R/W	Host Controller Low-speed Threshold Register	0x0000_0628
HcRhDescriptorA	USBH_BA+0x048	R/W	Host Controller Root Hub Descriptor A Register	0x0000_0902
HcRhDescriptorB	USBH_BA+0x04C	R/W	Host Controller Root Hub Descriptor B Register	0x0000_0000
HcRhStatus	USBH_BA+0x050	R/W	Host Controller Root Hub Status Register	0x0000_0000
HcRhPortStatus1	USBH_BA+0x054	R/W	Host Controller Root Hub Port Status [1]	0x0000_0000
HcRhPortStatus2	USBH_BA+0x058	R/W	Host Controller Root Hub Port Status [2]	0x0000_0000
HcPhyControl	USBH_BA+0x200	R/W	Host Controller PHY Control Register	0x0000_0000

HcMiscControl	USBH_BA+0x204	R/W	Host Controller Miscellaneous Control Register	0x0000_0000

6.33.7 Register Description

Host Controller Revision Register (HcRevision)

Register	Offset	R/W	Description	Reset Value
HcRevision	USBH_BA+0x000	R	Host Controller Revision Register	0x0000_0110

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
REV							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**REV**	**Revision** Indicates the Open HCI Specification revision number implemented by the Hardware. Host Controller supports 1.1 specification. (X.Y = XYh).

Host Controller Control Register (HcControl)

Register	Offset	R/W	Description	Reset Value
HcControl	USBH_BA+0x004	R/W	Host Controller Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
HCFS		**BLE**	**CLE**	**IE**	**PLE**	**CBSR**	

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**Reserved**	Reserved.
[7:6]	**HCFS**	**Host Controller Functional State** This field sets the Host Controller state. The Controller may force a state change from USBSUSPEND to USBRESUME after detecting resume signaling from a downstream port. States are: 00 = USBSUSPEND. 01 = USBOPERATIONAL. 10 = USBRESUME. 11 = USBRESET.
[5]	**BLE**	**Bulk List Enable Bit** 0 = Processing of the Bulk list after next SOF (Start-Of-Frame) Disabled. 1 = Processing of the Bulk list in the next frame Enabled.
[4]	**CLE**	**Control List Enable Bit** 0 = Processing of the Control list after next SOF (Start-Of-Frame) Disabled. 1 = Processing of the Control list in the next frame Enabled.
[3]	**IE**	**Isochronous Enable Bit** Both ISOEn and PLE (HcControl[2]) high enables Host Controller to process the Isochronous list. Either ISOEn or PLE (HcControl[2]) is low disables Host Controller to process the Isochronous list. 0 = Processing of the Isochronous list after next SOF (Start-Of-Frame) Disabled. 1 = Processing of the Isochronous list in the next frame Enabled, if the PLE (HcControl[2]) is high, too.

[2]	**PLE**	**Periodic List Enable Bit** When set, this bit enables processing of the Periodic (interrupt and isochronous) list. The Host Controller checks this bit prior to attempting any periodic transfers in a frame. 0 = Processing of the Periodic (Interrupt and Isochronous) list after next SOF (Start-Of-Frame) Disabled. 1 = Processing of the Periodic (Interrupt and Isochronous) list in the next frame Enabled. **Note:** To enable the processing of the Isochronous list, user has to set both PLE and IE (HcControl[3]) high.
[1:0]	**CBSR**	**Control Bulk Service Ratio** This specifies the service ratio between Control and Bulk EDs. Before processing any of the non-periodic lists, HC must compare the ratio specified with its internal count on how many nonempty Control EDs have been processed, in determining whether to continue serving another Control ED or switching to Bulk EDs. The internal count will be retained when crossing the frame boundary. In case of reset, HCD is responsible for restoring this Value. 00 = Number of Control EDs over Bulk EDs served is 1:1. 01 = Number of Control EDs over Bulk EDs served is 2:1. 10 = Number of Control EDs over Bulk EDs served is 3:1. 11 = Number of Control EDs over Bulk EDs served is 4:1.

Host Controller Command Status Register (HcCommandStatus)

Register	Offset	R/W	Description	Reset Value
HcCommandStatus	USBH_BA+0x008	R/W	Host Controller Command Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						SOC	
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					BLF	CLF	HCR

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17:16]	**SOC**	**Scheduling Overrun Count** These bits are incremented on each scheduling overrun error. It is initialized to 00b and wraps around at 11b. This will be incremented when a scheduling overrun is detected even if SO (HcInterruptStatus[0]) has already been set.
[15:3]	**Reserved**	Reserved.
[2]	**BLF**	**Bulk List Filled** Set high to indicate there is an active TD on the Bulk list. This bit may be set by either software or the Host Controller and cleared by the Host Controller each time it begins processing the head of the Bulk list. 0 = No active TD found or Host Controller begins to process the head of the Bulk list. 1 = An active TD added or found on the Bulk list.
[1]	**CLF**	**Control List Filled** Set high to indicate there is an active TD on the Control List. It may be set by either software or the Host Controller and cleared by the Host Controller each time it begins processing the head of the Control List. 0 = No active TD found or Host Controller begins to process the head of the Control list. 1 = An active TD added or found on the Control list.
[0]	**HCR**	**Host Controller Reset** This bit is set to initiate the software reset of Host Controller. This bit is cleared by the Host Controller, upon completed of the reset operation. This bit, when set, didn't reset the Root Hub and no subsequent reset signaling be asserted to its downstream ports. 0 = Host Controller is not in software reset state. 1 = Host Controller is in software reset state.

Host Controller Interrupt Status Register (HcInterruptStatus)

Register	Offset	R/W	Description	Reset Value
HcInterruptStatus	USBH_BA+0x00C	R/W	Host Controller Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	RHSC	FNO	Reserved	RD	SF	WDH	SO

Bits	Description	
[31:7]	**Reserved**	Reserved.
[6]	**RHSC**	**Root Hub Status Change** This bit is set when the content of HcRhStatus or the content of HcRhPortStatus1 register has changed. 0 = The content of HcRhStatus and the content of HcRhPortStatus1 register didn't change. 1 = The content of HcRhStatus or the content of HcRhPortStatus1 register has changed.
[5]	**FNO**	**Frame Number Overflow** This bit is set when bit 15 of Frame Number changes from 1 to 0 or from 0 to 1. 0 = The bit 15 of Frame Number didn't change. 1 = The bit 15 of Frame Number changes from 1 to 0 or from 0 to 1.
[4]	**Reserved**	Reserved
[3]	**RD**	**Resume Detected** Set when Host Controller detects resume signaling on a downstream port. 0 = No resume signaling detected on a downstream port. 1 = Resume signaling detected on a downstream port.
[2]	**SF**	**Start Of Frame** Set when the Frame Management functional block signals a 'Start of Frame' event. Host Control generates a SOF token at the same time. 0 =.Not the start of a frame. 1 =.Indicate the start of a frame and Host Controller generates a SOF token.
[1]	**WDH**	**Write Back Done Head** Set after the Host Controller has written HcDoneHead to HccaDoneHead. Further updates of the HccaDoneHead will not occur until this bit has been cleared. 0 =.Host Controller didn't update HccaDoneHead. 1 =.Host Controller has written HcDoneHead to HccaDoneHead.

[0]	**SO**	**Scheduling Overrun** Set when the List Processor determines a Schedule Overrun has occurred. 0 = Schedule Overrun didn't occur. 1 = Schedule Overrun has occurred.

Host Controller Interrupt Enable Control Register (HcInterruptEnable)

Register	Offset	R/W	Description	Reset Value
HcInterruptEnable	USBH_BA+0x010	R/W	Host Controller Interrupt Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MIE	Reserved						
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	RHSC	FNO	Reserved	RD	SF	WDH	SO

Bits	Description	
[31]	**MIE**	**Master Interrupt Enable Bit** This bit is a global interrupt enable. A write of '1' allows interrupts to be enabled via the specific enable bits listed above. Write Operation: 0 = No effect. 1 = Interrupt generation due to RHSC (HcInterruptStatus[6]), FNO (HcInterruptStatus[5]), RD (HcInterruptStatus[3]), SF (HcInterruptStatus[2]), WDH (HcInterruptStatus[1]) or SO (HcInterruptStatus[0]) Enabled if the corresponding bit in HcInterruptEnable is high. Read Operation: 0 = Interrupt generation due to RHSC (HcInterruptStatus[6]), FNO (HcInterruptStatus[5]), RD (HcInterruptStatus[3]), SF (HcInterruptStatus[2]), WDH (HcInterruptStatus[1]) or SO (HcInterruptStatus[0]) Disabled even if the corresponding bit in HcInterruptEnable is high. 1 = Interrupt generation due to RHSC (HcInterruptStatus[6]), FNO (HcInterruptStatus[5]), RD (HcInterruptStatus[3]), SF (HcInterruptStatus[2]), WDH (HcInterruptStatus[1]) or SO (HcInterruptStatus[0]) Enabled if the corresponding bit in HcInterruptEnable is high.
[30:7]	**Reserved**	Reserved.
[6]	**RHSC**	**Root Hub Status Change Interrupt Enable Bit** Write Operation: 0 = No effect. 1 = Interrupt generation due to RHSC (HcInterruptStatus[6]) Enabled. Read Operation: 0 = Interrupt generation due to RHSC (HcInterruptStatus[6]) Disabled. 1 = Interrupt generation due to RHSC (HcInterruptStatus[6]) Enabled.

[5]	**FNO**	**Frame Number Overflow Interrupt Enable Bit** Write Operation: 0 = No effect. 1 = Interrupt generation due to FNO (HcInterruptStatus[5]) Enabled. Read Operation: 0 = Interrupt generation due to FNO (HcInterruptStatus[5]) Disabled. 1 = Interrupt generation due to FNO (HcInterruptStatus[5]) Enabled.
[4]	**Reserved**	Reserved.
[3]	**RD**	**Resume Detected Interrupt Enable Bit** Write Operation: 0 = No effect. 1 = Interrupt generation due to RD (HcInterruptStatus[3]) Enabled. Read Operation: 0 = Interrupt generation due to RD (HcInterruptStatus[3]) Disabled. 1 = Interrupt generation due to RD (HcInterruptStatus[3]) Enabled.
[2]	**SF**	**Start Of Frame Interrupt Enable Bit** Write Operation: 0 = No effect. 1 = Interrupt generation due to SF (HcInterruptStatus[2]) Enabled. Read Operation: 0 = Interrupt generation due to SF (HcInterruptStatus[2]) Disabled. 1 = Interrupt generation due to SF (HcInterruptStatus[2]) Enabled.
[1]	**WDH**	**Write Back Done Head Interrupt Enable Bit** Write Operation: 0 = No effect. 1 = Interrupt generation due to WDH (HcInterruptStatus[1]) Enabled. Read Operation: 0 = Interrupt generation due to WDH (HcInterruptStatus[1]) Disabled. 1 = Interrupt generation due to WDH (HcInterruptStatus[1]) Enabled.
[0]	**SO**	**Scheduling Overrun Interrupt Enable Bit** Write Operation: 0 = No effect. 1 = Interrupt generation due to SO (HcInterruptStatus[0]) Enabled. Read Operation: 0 = Interrupt generation due to SO (HcInterruptStatus[0]) Disabled. 1 = Interrupt generation due to SO (HcInterruptStatus[0]) Enabled.

Host Controller Interrupt Disable Control Register (HcInterruptDisable)

Register	Offset	R/W	Description	Reset Value
HcInterruptDisable	USBH_BA+0x014	R/W	Host Controller Interrupt Disable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MIE	**Reserved**						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**RHSC**	**FNO**	**Reserved**	**RD**	**SF**	**WDH**	**SO**

Bits	Description	
[31]	**MIE**	**Master Interrupt Disable Bit** Global interrupt disable. Writing '1' to disable all interrupts. Write Operation: 0 = No effect. 1 = Interrupt generation due to RHSC (HcInterruptStatus[6]), FNO (HcInterruptStatus[5]), RD (HcInterruptStatus[3]), SF (HcInterruptStatus[2]), WDH (HcInterruptStatus[1]) or SO (HcInterruptStatus[0]) Disabled if the corresponding bit in HcInterruptEnable is high. Read Operation: 0 = Interrupt generation due to RHSC (HcInterruptStatus[6]), FNO (HcInterruptStatus[5]), RD (HcInterruptStatus[3]), SF (HcInterruptStatus[2]), WDH (HcInterruptStatus[1]) or SO (HcInterruptStatus[0]) Disabled even if the corresponding bit in HcInterruptEnable is high. 1 = Interrupt generation due to RHSC (HcInterruptStatus[6]), FNO (HcInterruptStatus[5]), RD (HcInterruptStatus[3]), SF (HcInterruptStatus[2]), WDH (HcInterruptStatus[1]) or SO (HcInterruptStatus[0]) Enabled if the corresponding bit in HcInterruptEnable is high.
[30:7]	**Reserved**	Reserved.
[6]	**RHSC**	**Root Hub Status Change Disable Bit** Write Operation: 0 = No effect. 1 = Interrupt generation due to RHSC (HcInterruptStatus[6]) Disabled. Read Operation: 0 = Interrupt generation due to RHSC (HcInterruptStatus[6]) Disabled. 1 = Interrupt generation due to RHSC (HcInterruptStatus[6]) Enabled.

[5]	**FNO**	**Frame Number Overflow Disable Bit** Write Operation: 0 = No effect. 1 = Interrupt generation due to FNO (HcInterruptStatus[5]) Disabled. Read Operation: 0 = Interrupt generation due to FNO (HcInterruptStatus[5]) Disabled. 1 = Interrupt generation due to FNO (HcInterruptStatus[5]) Enabled.
[4]	**Reserved**	Reserved.
[3]	**RD**	**Resume Detected Disable Bit** Write Operation: 0 = No effect. 1 = Interrupt generation due to RD (HcInterruptStatus[3]) Disabled. Read Operation: 0 = Interrupt generation due to RD (HcInterruptStatus[3]) Disabled. 1 = Interrupt generation due to RD (HcInterruptStatus[3]) Enabled.
[2]	**SF**	**Start Of Frame Disable Bit** Write Operation: 0 = No effect. 1 = Interrupt generation due to SF (HcInterruptStatus[2]) Disabled. Read Operation: 0 = Interrupt generation due to SF (HcInterruptStatus[2]) Disabled. 1 = Interrupt generation due to SF (HcInterruptStatus[2]) Enabled.
[1]	**WDH**	**Write Back Done Head Disable Bit** Write Operation: 0 = No effect. 1 = Interrupt generation due to WDH (HcInterruptStatus[1]) Disabled. Read Operation: 0 = Interrupt generation due to WDH (HcInterruptStatus[1]) Disabled. 1 = Interrupt generation due to WDH (HcInterruptStatus[1]) Enabled.
[0]	**SO**	**Scheduling Overrun Disable Bit** Write Operation: 0 = No effect. 1 = Interrupt generation due to SO (HcInterruptStatus[0]) Disabled. Read Operation: 0 = Interrupt generation due to SO (HcInterruptStatus[0]) Disabled. 1 = Interrupt generation due to SO (HcInterruptStatus[0]) Enabled.

Host Controller Communication Area Register (HcHCCA)

Register	Offset	R/W	Description	Reset Value
HcHCCA	USBH_BA+0x018	R/W	Host Controller Communication Area Register	0x0000_0000

31	30	29	28	27	26	25	24
HCCA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
HCCA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
HCCA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:8]	**HCCA**	**Host Controller Communication Area** Pointer to indicate base address of the Host Controller Communication Area (HCCA).
[7:0]	**Reserved**	Reserved.

Host Controller Period Current ED Register (HcPeriodCurrentED)

Register	Offset	R/W	Description	Reset Value
HcPeriodCurrentED	USBH_BA+0x01C	R/W	Host Controller Period Current ED Register	0x0000_0000

31	30	29	28	27	26	25	24
PCED							
23	22	21	20	19	18	17	16
PCED							
15	14	13	12	11	10	9	8
PCED							
7	6	5	4	3	2	1	0
PCED				Reserved			

Bits	Description	
[31:4]	**PCED**	**Periodic Current ED** Pointer to indicate physical address of the current Isochronous or Interrupt Endpoint Descriptor.
[3:0]	**Reserved**	Reserved.

Host Controller Control Head ED Register (HcControlHeadED)

Register	Offset	R/W	Description	Reset Value
HcControlHeadED	USBH_BA+0x020	R/W	Host Controller Control Head ED Register	0x0000_0000

31	30	29	28	27	26	25	24
CHED							
23	22	21	20	19	18	17	16
CHED							
15	14	13	12	11	10	9	8
CHED							
7	6	5	4	3	2	1	0
CHED				Reserved			

Bits	Description	
[31:4]	**CHED**	**Control Head ED** Pointer to indicate physical address of the first Endpoint Descriptor of the Control list.
[3:0]	**Reserved**	Reserved.

Host Controller Control Current ED Register (HcControlCurrentED)

Register	Offset	R/W	Description	Reset Value
HcControlCurrentED	USBH_BA+0x024	R/W	Host Controller Control Current ED Register	0x0000_0000

31	30	29	28	27	26	25	24
CCED							
23	22	21	20	19	18	17	16
CCED							
15	14	13	12	11	10	9	8
CCED							
7	6	5	4	3	2	1	0
CCED				Reserved			

Bits	Description	
[31:4]	**CCED**	**Control Current Head ED** Pointer to indicate the physical address of the current Endpoint Descriptor of the Control list.
[3:0]	**Reserved**	Reserved.

Host Controller Bulk Head ED Register (HcBulkHeadED)

Register	Offset	R/W	Description	Reset Value
HcBulkHeadED	USBH_BA+0x028	R/W	Host Controller Bulk Head ED Register	0x0000_0000

31	30	29	28	27	26	25	24
BHED							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
BHED							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
BHED							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BHED				**Reserved**			

Bits	Description	
[31:4]	**BHED**	**Bulk Head ED** Pointer to indicate the physical address of the first Endpoint Descriptor of the Bulk list.
[3:0]	**Reserved**	Reserved.

Host Controller Bulk Current Head ED Register (HcBulkCurrentED)

Register	Offset	R/W	Description	Reset Value
HcBulkCurrentED	USBH_BA+0x02C	R/W	Host Controller Bulk Current ED Register	0x0000_0000

31	30	29	28	27	26	25	24
BCED							
23	22	21	20	19	18	17	16
BCED							
15	14	13	12	11	10	9	8
BCED							
7	6	5	4	3	2	1	0
BCED				Reserved			

Bits	Description	
[31:4]	**BCED**	**Bulk Current Head ED** Pointer to indicate the physical address of the current endpoint of the Bulk list.
[3:0]	**Reserved**	Reserved.

Host Controller Done Head Register (HcDoneHead)

Register	Offset	R/W	Description	Reset Value
HcDoneHead	USBH_BA+0x030	R/W	Host Controller Done Head Register	0x0000_0000

31	30	29	28	27	26	25	24
DH							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DH							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DH							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DH				**Reserved**			

Bits	Description	
[31:4]	**DH**	**Done Head** Pointer to indicate the physical address of the last completed Transfer Descriptor that was added to the Done queue.
[3:0]	**Reserved**	Reserved.

Host Controller Frame Interval Register (HcFmInterval)

Register	Offset	R/W	Description	Reset Value
HcFmInterval	USBH_BA+0x034	R/W	Host Controller Frame Interval Register	0x0000_2EDF

31	30	29	28	27	26	25	24
FIT	**FSMPS**						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
FSMPS							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**FI**					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FI							

Bits	Description	
[31]	**FIT**	**Frame Interval Toggle** This bit is toggled by Host Controller Driver when it loads a new value into FI (HcFmInterval[13:0]). 0 = Host Controller Driver didn't load new value into FI (HcFmInterval[13:0]). 1 = Host Controller Driver loads a new value into FI (HcFmInterval[13:0]).
[30:16]	**FSMPS**	**FS Largest Data Packet** This field specifies a value that is loaded into the Largest Data Packet Counter at the beginning of each frame.
[15:14]	**Reserved**	Reserved.
[13:0]	**FI**	**Frame Interval** This field specifies the length of a frame as (bit times - 1). For 12,000 bit times in a frame, a value of 11,999 is stored here.

Host Controller Frame Remaining Register (HcFmRemaining)

Register	Offset	R/W	Description	Reset Value
HcFmRemaining	USBH_BA+0x038	R	Host Controller Frame Remaining Register	0x0000_0000

31	30	29	28	27	26	25	24
FRT	Reserved						
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		FR					
7	6	5	4	3	2	1	0
FR							

Bits	Description	
[31]	FRT	**Frame Remaining Toggle** This bit is loaded from the FIT (HcFmInterval[31]) whenever FR (HcFmRemaining[13:0]) reaches 0
[30:14]	Reserved	Reserved.
[13:0]	FR	**Frame Remaining** When the Host Controller is in the USBOPERATIONAL state, this 14-bit field decrements each 12 MHz clock period. When the count reaches 0, (end of frame) the counter reloads with Frame Interval. In addition, the counter loads when the Host Controller transitions into USBOPERATIONAL.

Host Controller Frame Number Register (HcFmNumber)

Register	Offset	R/W	Description	Reset Value
HcFmNumber	USBH_BA+0x03C	R	Host Controller Frame Number Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FN							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**FN**	**Frame Number** This 16-bit incrementing counter field is incremented coincident with the loading of FR (HcFmRemaining[13:0]). The count rolls over from 'FFFFh' to '0h.'

Host Controller Periodic Start Register (HcPeriodicStart)

Register	Offset	R/W	Description	Reset Value
HcPeriodicStart	USBH_BA+0x040	R/W	Host Controller Periodic Start Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**PS**					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PS							

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13:0]	**PS**	**Periodic Start** This field contains a value used by the List Processor to determine where in a frame the Periodic List processing must begin.

Host Controller Low-speed Threshold Register (HcLSThreshold)

Register	Offset	R/W	Description	Reset Value
HcLSThreshold	USBH_BA+0x044	R/W	Host Controller Low-speed Threshold Register	0x0000_0628

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**LST**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
LST							

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11:0]	**LST**	**Low-Speed Threshold** This field contains a value which is compared to the FR (HcFmRemaining[13:0]) field prior to initiating a Low-speed transaction. The transaction is started only if FR (HcFmRemaining[13:0]) >= this field. The value is calculated by Host Controller Driver with the consideration of transmission and setup overhead.

Host Controller Root Hub Descriptor A Register (HcRhDescriptorA)

Register	Offset	R/W	Description	Reset Value
HcRhDescriptorA	USBH_BA+0x048	R/W	Host Controller Root Hub Descriptor A Register	0x0000_0902

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved			NOCP	OCPM	DT	NPS	PSM
7	6	5	4	3	2	1	0
NDP							

Bits	Description	
[31:13]	**Reserved**	Reserved.
[12]	**NOCP**	**No Overcurrent Protection** This bit describes how the over current status for the Root Hub ports reported. 0 = Over current status is reported. 1 = Over current status is not reported.
[11]	**OCPM**	**Overcurrent Protection Mode** This bit describes how the over current status for the Root Hub ports reported. This bit is only valid when NOCP (HcRhDescriptorA[12]) is cleared. 0 = Global Over current. 1 = Individual Over current.
[10:9]	**Reserved**	Reserved.
[8]	**PSM**	**Power Switching Mode** This bit is used to specify how the power switching of the Root Hub ports is controlled. 0 = Global Switching. 1 = Individual Switching.
[7:0]	**NDP**	**Number Downstream Ports** Root Hub supports two downstream ports. It's 2 in this Root Hub.

Host Controller Root Hub Descriptor B Register (HcRhDescriptorB)

Register	Offset	R/W	Description	Reset Value
HcRhDescriptorB	USBH_BA+0x04C	R/W	Host Controller Root Hub Descriptor B Register	0x0000_0000

31	30	29	28	27	26	25	24
PPCM							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PPCM							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:16]	**PPCM**	**Port Power Control Mask** Global power switching. This field is only valid if PowerSwitchingMode is set (individual port switching). When set, the port only responds to individual port power switching commands (Set/ClearPortPower). When cleared, the port only responds to global power switching commands (Set/ClearGlobalPower). 0 = Port power controlled by global power switching. 1 = Port power controlled by port power switching. Note: PPCM[15:3] and PPCM[0] are reserved.
[15:0]	**Reserved**	Reserved.

Host Controller Root Hub Status Register (HcRhStatus)

Register	Offset	R/W	Description	Reset Value
HcRhStatus	USBH_BA+0x050	R/W	Host Controller Root Hub Status Register	0x0000_0000

31	30	29	28	27	26	25	24
CRWE	**Reserved**						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						**OCIC**	**LPSC**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DRWE	**Reserved**						
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**OCI**	**LPS**

Bits	Description	
[31]	**CRWE**	**Clear Remote Wake-Up Enable Bit** This bit is use to clear DRWE (HcRhStatus[15]). This bit always read as zero. Write Operation: 0 = No effect. 1 = Clear DRWE (HcRhStatus[15]).
[31:18]	**Reserved**	Reserved.
[17]	**OCIC**	**Over Current Indicator Change** This bit is set by hardware when a change has occurred in OCI (HcRhStatus[1]). Write 1 to clear this bit to zero. 0 = OCI (HcRhStatus[1]) didn't change. 1 = OCI (HcRhStatus[1]) change.
[16]	**LPSC**	**SetGlobalPower** In global power mode (PSM (HcRhDescriptorA[8]) = 0), this bit is written to one to enable power to all ports. This bit always read as zero. Write Operation: 0 = No effect. 1 = Set global power.

[15]	**DRWE**	**Device Remote Wakeup Enable Bit** This bit controls if port's Connect Status Change as a remote wake-up event. Write Operation: 0 = No effect. 1 = Connect Status Change as a remote wake-up event Enabled. Read Operation: 0 = Connect Status Change as a remote wake-up event Disabled. 1 = Connect Status Change as a remote wake-up event Enabled.
[14:2]	**Reserved**	Reserved.
[1]	**OCI**	**Overcurrent Indicator** This bit reflects the state of the over current status pin. This field is only valid if NOCP (HcRhDesA[12]) and OCPM (HcRhDesA[11]) are cleared. 0 = No over current condition. 1 = Over current condition.
[0]	**LPS**	**Clear Global Power** In global power mode (PSM (HcRhDescriptorA[8]) = 0), this bit is written to one to clear all ports' power. This bit always read as zero. Write Operation: 0 = No effect. 1 = Clear global power.

Host Controller Root Hub Port Status (HcRhPrt [1: 2])

Register	Offset	R/W	Description	Reset Value
HcRhPortStatus1	USBH_BA+0x054	R/W	Host Controller Root Hub Port Status [1]	0x0000_0000
HcRhPortStatus2	USBH_BA+0x058	R/W	Host Controller Root Hub Port Status [2]	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			**PRSC**	**OCIC**	**PSSC**	**PESC**	**CSC**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**LSDA**	**PPS**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**PRS**	**POCI**	**PSS**	**PES**	**CCS**

Bits	Description	
[31:21]	**Reserved**	Reserved.
[20]	**PRSC**	**Port Reset Status Change** This bit indicates that the port reset signal has completed. Write 1 to clear this bit to zero. 0 = Port reset is not complete. 1 = Port reset is complete.
[19]	**OCIC**	**Port Over Current Indicator Change** This bit is set when POCI (HcRhPortStatus1[3]) changes. Write 1 to clear this bit to zero. 0 = POCI (HcRhPortStatus1[3]) didn't change. 1 = POCI (HcRhPortStatus1[3]) changes.
[18]	**PSSC**	**Port Suspend Status Change** This bit indicates the completion of the selective resume sequence for the port. Write 1 to clear this bit to zero. 0 = Port resume is not completed. 1 = Port resume completed.
[17]	**PESC**	**Port Enable Status Change** This bit indicates that the port has been disabled (PES (HcRhPortStatus1[1]) cleared) due to a hardware event. Write 1 to clear this bit to zero. 0 = PES (HcRhPortStatus1[1]) didn't change. 1 = PES (HcRhPortStatus1[1]) changed.

[16]	**CSC**	**Connect Status Change** This bit indicates connect or disconnect event has been detected (CCS (HcRhPortStatus1[0]) changed). Write 1 to clear this bit to zero. 0 = No connect/disconnect event (CCS (HcRhPortStatus1[0]) didn't change). 1 = Hardware detection of connect/disconnect event (CCS (HcRhPortStatus1[0]) changed).
[15:10]	**Reserved**	Reserved.
[9]	**LSDA**	**Low Speed Device Attached (Read) Or Clear Port Power (Write)** This bit defines the speed (and bud idle) of the attached device. It is only valid when CCS (HcRhPortStatus1[0]) is set. This bit is also used to clear port power. Write Operation: 0 = No effect. 1 = Clear PPS (HcRhPortStatus1[8]). Read Operation: 0 = Full Speed device. 1 = Low-speed device.
[8]	**PPS**	**Port Power Status** This bit reflects the power state of the port regardless of the power switching mode. Write Operation: 0 = No effect. 1 = Port Power Enabled. Read Operation: 0 = Port power is Diabled. 1 = Port power is Enabled.
[7:5]	**Reserved**	Reserved.
[4]	**PRS**	**Port Reset Status** This bit reflects the reset state of the port. Write Operation: 0 = No effect. 1 = Set port reset. Read Operation 0 = Port reset signal is not active. 1 = Port reset signal is active.
[3]	**POCI**	**Port Over Current Indicator (Read) Or Clear Port Suspend (Write)** This bit reflects the state of the over current status pin dedicated to this port. This field is only valid if NOCP (HcRhDescriptorA[12]) is cleared and OCPM (HcRhDescriptorA[11]) is set. This bit is also used to initiate the selective result sequence for the port. Write Operation: 0 = No effect. 1 = Clear port suspend. Read Operation: 0 = No over current condition. 1 = Over current condition.

[2]	**PSS**	**Port Suspend Status** This bit indicates the port is suspended Write Operation: 0 = No effect. 1 = Set port suspend. Read Operation: 0 = Port is not suspended. 1 = Port is selectively suspended.
[1]	**PES**	**Port Enable Status** Write Operation: 0 = No effect. 1 = Set port enable. Read Operation: 0 = Port Disabled. 1 = Port Enabled.
[0]	**CCS**	**CurrentConnectStatus (Read) Or ClearPortEnable Bit (Write)** Write Operation: 0 = No effect. 1 = Clear port enable. Read Operation: 0 = No device connected. 1 = Device connected.

Host Controller PHY Control Register (HcPhyControl)

Register	Offset	R/W	Description	Reset Value
HcPhyControl	USBH_BA+0x200	R/W	Host Controller PHY Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				**STBYEN**	**Reserved**		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27]	**STBYEN**	**USB Transceiver Standby Enable Bit** This bit controls if USB transceiver could enter the standby mode to reduce power consumption. 0 = The USB transceiver would never enter the standby mode. 1 = The USB transceiver will enter standby mode while port is in power off state (port power is inactive).
[26:0]	**Reserved**	Reserved.

Host Controller Miscellaneous Control Register (HcMiscControl)

Register	Offset	R/W	Description	Reset Value
HcMiscControl	USBH_BA+0x204	R/W	Host Controller Miscellaneous Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						**DPRT2**	**DPRT1**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**SIEPD**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**PCAL**	**OCAL**	**Reserved**	**ABORT**	**DBR16**

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17]	**DPRT2**	**Port 2 Disable Bit** This bit controls if the connection between USB host controller and transceiver of port 2 is disabled. If the connection is disabled, the USB host controller will not recognize any event of USB bus. Set this bit high, the transceiver of port 2 will also be forced into the standby mode no matter what USB host controller operation is. 0 = The connection between USB host controller and transceiver of port 2 is enabled. 1 = The connection between USB host controller and transceiver of port 2 is disabled and the transceiver of port 2 will also be forced into the standby mode.
[16]	**DPRT1**	**Port 1 Disable Bit** This bit controls if the connection between USB host controller and transceiver of port 1 is disabled. If the connection is disabled, the USB host controller will not recognize any event of USB bus. Set this bit high, the transceiver of port 1 will also be forced into the standby mode no matter what USB host controller operation is. 0 = The connection between USB host controller and transceiver of port 1 is enabled. 1 = The connection between USB host controller and transceiver of port 1 is disabled and the transceiver of port 1 will also be forced into the standby mode.
[15:9]	**Reserved**	Reserved.
[8]	**SIEPD**	**SIE Pipeline Disable Bit** When set, waits for all USB bus activity to complete prior to returning completion status to the List Processor. This is a failsafe mechanism to avoid potential problems with the clk_dr transition between 1.5 MHz and 12 MHz.
[7:5]	**Reserved**	Reserved.

[4]	**PCAL**	**Port Power Control Active Low** This bit controls the polarity of port power control to external power IC. 0 = Port power control is high active. 1 = Port power control is low active.
[3]	**OCAL**	**Overcurrent Active Low** This bit controls the polarity of overcurrent flag from external power IC. 0 = Overcurrent flag is high active. 1 = Overcurrent flag is low active.
[2]	**Reserved**	Reserved.
[1]	**ABORT**	**AHB Bus ERROR Response** This bit indicates there is an ERROR response received in AHB bus. 0 = No ERROR response received. 1 = ERROR response received.
[0]	**DBR16**	**Data Buffer Region 16** When set, the size of the data buffer region is 16 bytes. Otherwise, the size is 32 bytes.

6.34 USB OTG Controller

6.34.1 Overview

The USB OTG controller is used to interface USB OTG PHY and USB controller, either USB host controller or USB device controller. The USB OTG controller supports the HNP and SRP protocols defined in the On-The-Go and Embedded Host Supplement to the USB 2.0 Revision 1.3 Specification. Combining USB host controller, USB device controller and USB OTG controller can act as Host-only, Device-only, ID-dependent or OTG Device through setting. Host-only mode can support both full-speed and low-speed. Device-only mode only supports full-speed. ID-dependent mode and OTG Device mode supporting speed is dependent on current role.

6.34.2 Features

- Built-in OTG PHY to support protocols defined in On-The-GO Supplement Rev 1.3 Specification, Including:
  - HNP: Host Negotiation Protocol
  - SRP: Session Request Protocol
- Configurable to operate as:
  - Host-only
  - Device-only
  - ID dependent device
  - OTG device: A-device or B-device, depending on the ID pin status.

6.34.3 Block Diagram

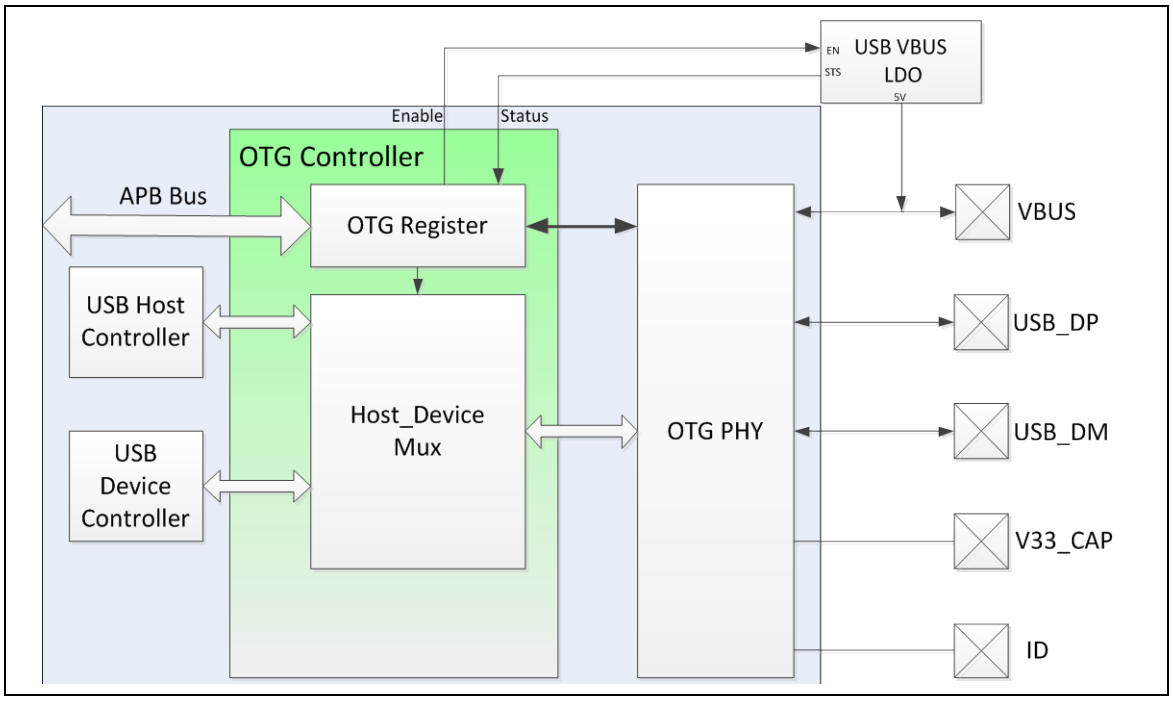


Figure 6.34-1 USB OTG Controller Block Diagram

6.34.4 Functional Description

The USB OTG controller is used to interface USB OTG PHY and USB controller, either USB host controller or USB device controller. Combining USB host controller, USB device controller and USB OTG controller can act as Host-only, Device-only or OTG Device through setting In Host-only, Device-only or ID dependent mode, OTG controller acts simply as a multiplexer. In OTG Device mode, the OTG controller will handle OTG protocols, including HNP, and SRP, depending on the role. If the ID pin is false, as OTG A-device, OTG controller will handle HNP protocol. If the ID pin is true, as OTG B-device, OTG controller will handle SRP protocol.

Host-only mode

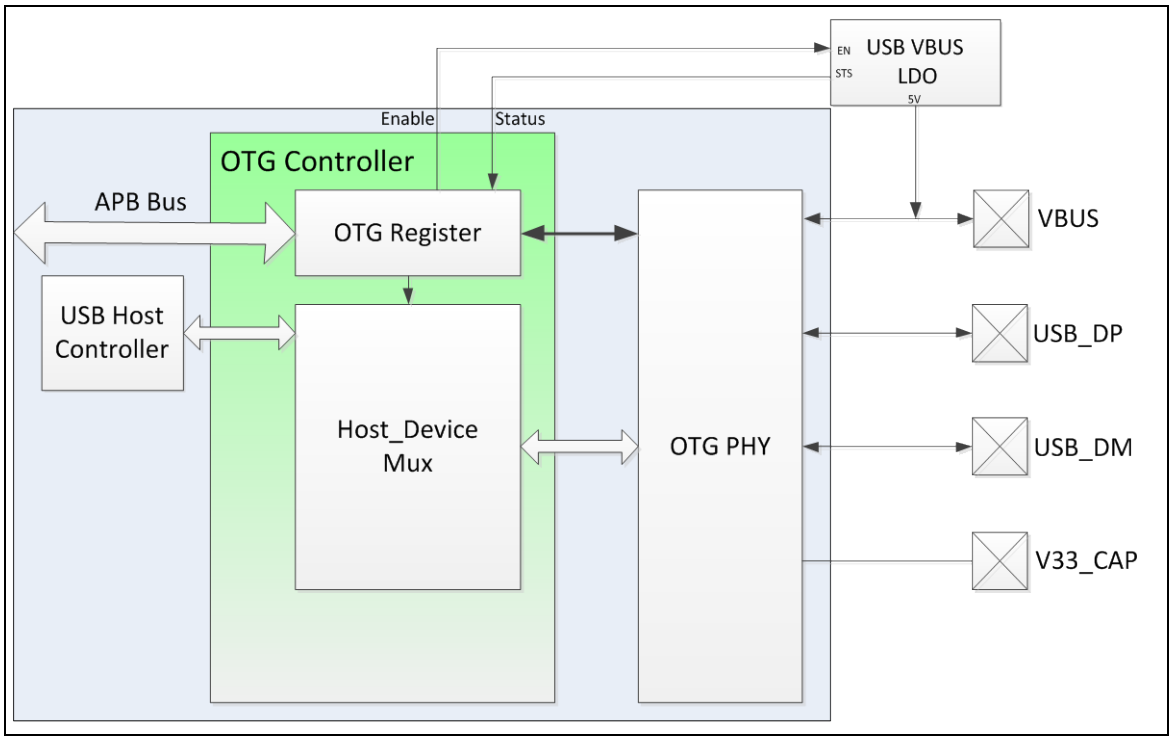


Figure 6.34-2 HOST-Only Mode

Device-only mode

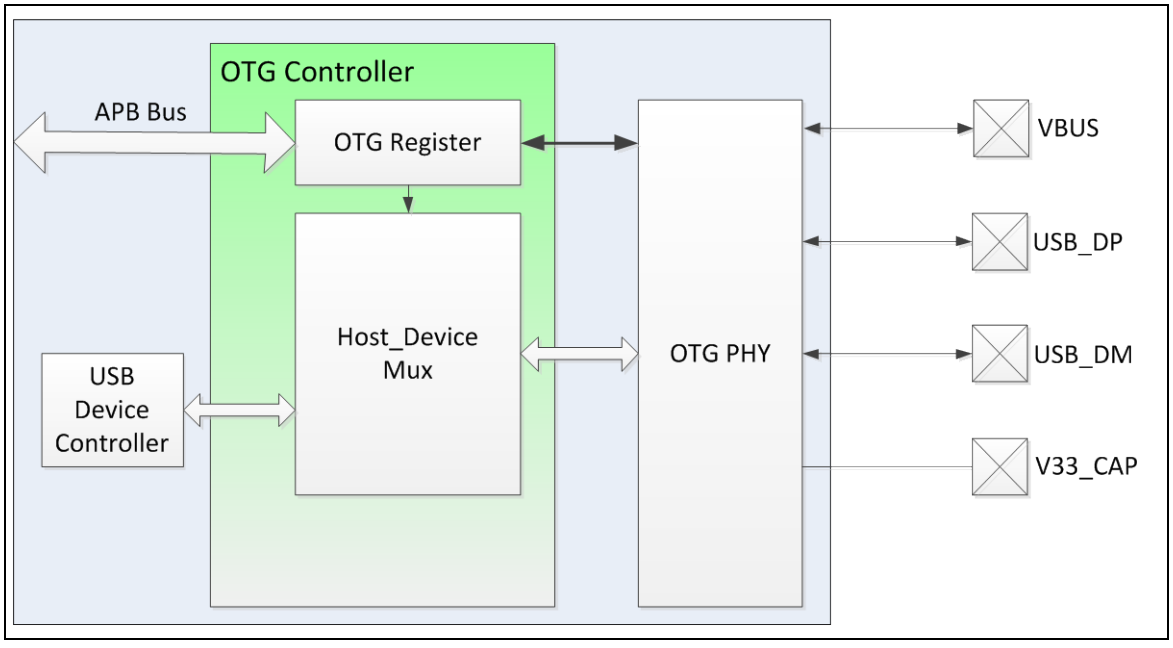


Figure 6.34-3 Device-Only Mode

ID-dependent (ID low)

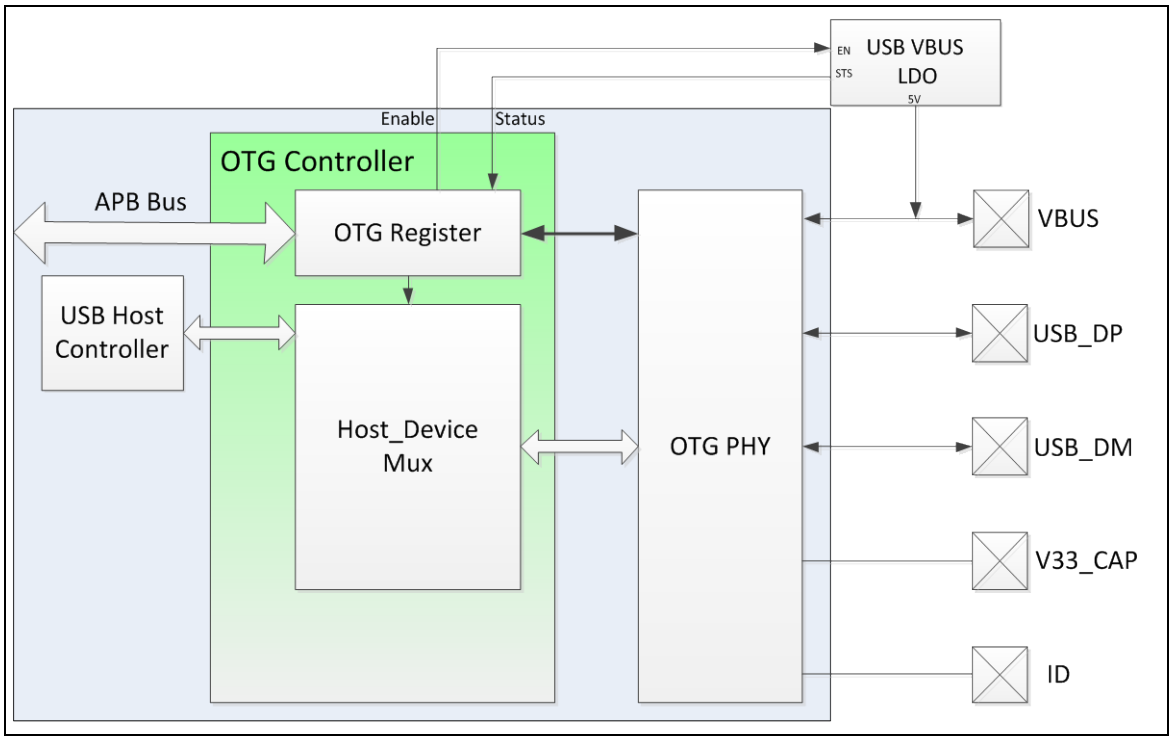


Figure 6.34-4 ID-Dependent (ID Low)

ID-dependent (ID high)

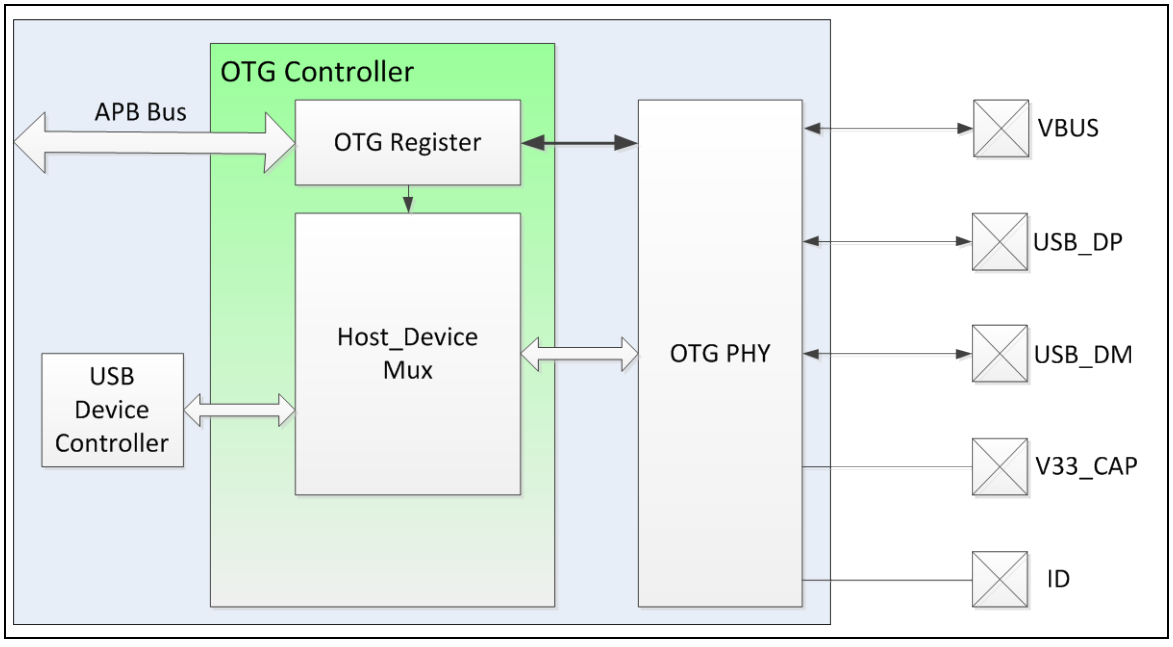


Figure 6.34-5 ID-Dependent (ID High)

OTG device

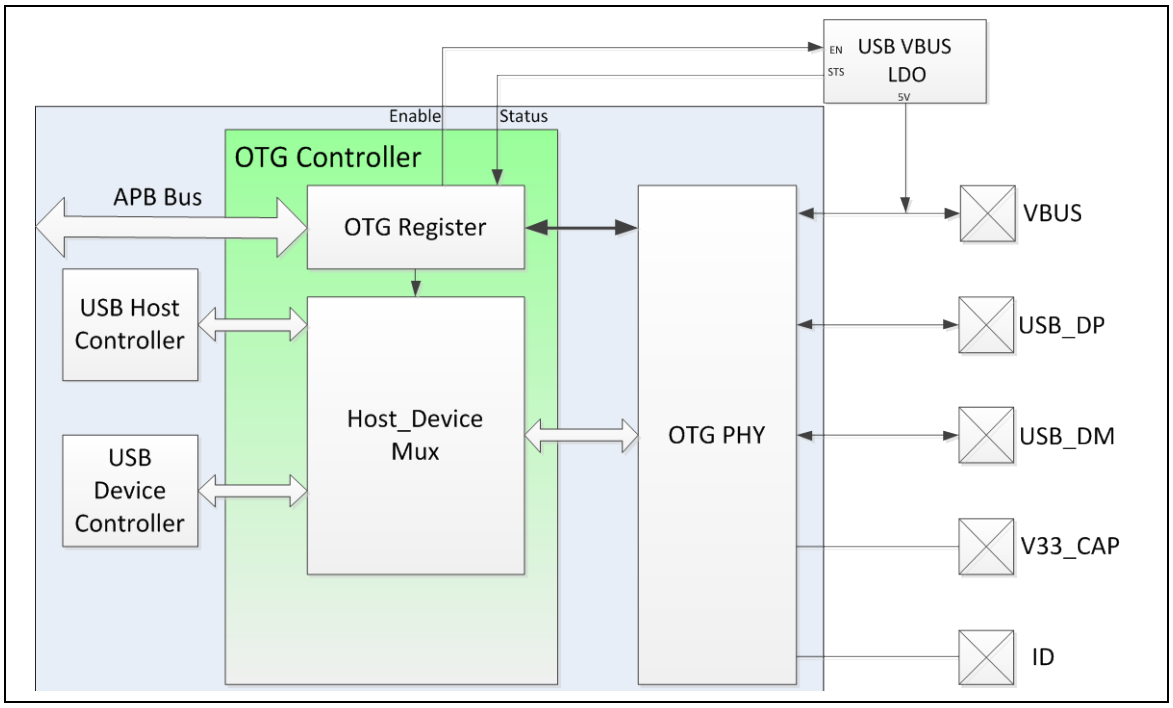


Figure 6.34-6 OTG Device

6.34.5 Register and Memory Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
OTG Base Address: **OTG_BA = 0x4004_D000**				
OTG_CTL	OTG_BA+0x00	R/W	OTG Control Register	0x0000_0000
OTG_PHYCTL	OTG_BA+0x04	R/W	OTG PHY Control Register	0x0000_0080
OTG_INTEN	OTG_BA+0x08	R/W	OTG Interrupt Enable Register	0x0000_0000
OTG_INTSTS	OTG_BA+0x0C	R/W	OTG Interrupt Status Register	0x0000_0000
OTG_STATUS	OTG_BA+0x10	R	Functional Status Register	0x0000_0006

6.34.6 Register Description

OTG Control Register (OTG_CTL)

Register	Offset	R/W	Description	Reset Value
OTG_CTL	OTG_BA+0x00	R/W	OTG Control Register	0x0000_0000

+	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							WKEN
7	6	5	4	3	2	1	0
PDEVCKON	Reserved	Reserved	OTGEN	Reserved	HNPREQEN	BUSREQ	VBUSDROP

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**WKEN**	**OTG Wake-Up Enable Bit** 0= OTG ID pin status change wake-up Disabled. 1 = OTG ID pin status change wake-up Enabled.
[7]	**PDEVCKON**	**Force OTG PHY Output Clock To USB Device** If software configures OTG controller as OTG device and OTG device as A-device, OTG controller will output OTG PHY clock (30 MHz) to USB device only when OTG device as A-peripheral. If software needs to configure USB device before role change (from A-Host to A-Peripheral), software can set this bit high to output OTG PHY clock to USB device. 0= USB device clock is available only when OTG device as a peripheral. 1 = Force output OTG PHY clock to USB device.
[6]	**Reserved**	Reserved.
[5]	**Reserved**	Reserved.
[4]	**OTGEN**	**OTG Function** Enable Bit If USB is configured as OTG device, this bit must set high. 0= OTG function Disabled. 1 = OTG function Enabled.
[3]	**Reserved**	Reserved.
[2]	**HNPREQEN**	**OTG B-Device HNP Enable/Request** Set this bit to TRUE after the OTG A-device successfully sends a SetFeature(b_hnp_enable) command to the OTG B-device This bit will be cleared automatically when a bus reset or SESS_VLD goes from TRUE to FALSE.
[1]	**BUSREQ**	**OTG A-Device Bus Request** If user application of an OTG A-device wants to do data transfers via USB bus, set this bit to high Otherwise if user application won't use the bus any more, set this bit low. This bit

Bits	Description	
		will be automatically cleared if VBUSDROP bit is set to TRUE.
[0]	**VBUSDROP**	**Drop The VUSB Bus** If user application running on this OTG A-device wants to conserve power consumption, set this bit to high When set this bit to TRUE, BUSREQ shall be cleared as well. 0 = Did Not drop the VBUS and keep going on USB data transfers. 1 = Drop the VBUS to conserve power consumption.

OTG PHY Control Register (OTG_PHYCTL)

Register	Offset	R/W	Description	Reset Value
OTG_PHYCTL	OTG_BA+0x04	R/W	OTG PHY Control Register	0x0000_0080

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved						OTGPHYEN	PHYCLK
7	6	5	4	3	2	1	0
IDDETEN	VBENPOL	VBSTSPOL	Reserved		DMPDEN	DPPDEN	SWPDEN

Bits	Description	
[31:10]	**Reserved**	Reserved.
[9]	**OTGPHYEN**	**OTG PHY Enable Control when Device Configured as OTG-Device** When device is configured as OTG-device, hardware will not enable OTG PHY automatically. Software can set OTG_EN to enable OTG PHY. 0 = OTG PHY Disabled. 1 = OTG PHY Enabled.
[8]	**PHYCLK**	**PHY Input Clock Selection** 0 = PHY input clock is12 MHz. 1 = PHY input clock is 24 MHz.
[7]	**IDDETEN**	**ID Detection Enable** 0 = Sampling on ID pin Enabled. 1 = Sampling on ID pin Disabled.
[6]	**VBENPOL**	**Off-Chip USB VBUS Power Enable Polarity** The OTG controller will enable off-chip USB VBUS LDO to provide VBUS power when need. The polarity of enabling off-chip BSU VBUS LDO (high active or low active) depends on the selected component. This bit provides the inverse option of off-chip USB VBUS LDO enable. 0 = The polarity of enabling off-chip USB VBUS LDO from the OTG controller not inversed. 1 = The polarity of enabling off-chip USB VBUS LDO from the OTG controller inversed.
[5]	**VBSTSPOL**	**Off-Chip USB VBUS Power Status Polarity** The polarity of off-chip USB VBUS LDO valid depends on the selected component. This bit provides the inversed option of off-chip USB VBUS LDO valid. 0 = The polarity of off-chip USB VBUS LDO valid not inversed. 1 = The polarity of off-chip USB VBUS LDO valid inversed.
[4:3]	**Reserved**	Reserved.
[2]	**DMPDEN**	**D- Pull-Down Enable Control Set SWPDEN to TRUE before using this function**

Bits	Description	
		0 = 15 kΩ resistor pull-down on D- pin Disabled. 1 = 15 kΩ resistor pull-down on D- pin Enabled.
[1]	**DPPDEN**	**D+ Pull-Down Enable Control Set SWPDEN to TRUE before using this function** 0 = 15 kΩ resistor pull-down on D+ pin Disabled. 1 = 15 kΩ resistor pull-down on D+ pin Enabled.
[0]	**SWPDEN**	**Software Control Pull-Down On Data Lines Enable Bit** 0 = Pull-down resistors on data lines is controlled by OTG control logic. 1 = Pull-down resistors on data lines is controlled by software. **Note:** Software must set this bit high before controlling DPPDEN and DMPDEN.

OTG Interrupt Enable Register (OTG_INTEN)

Register	Offset	R/W	Description	Reset Value
OTG_INTEN	OTG_BA+0x08	R/W	OTG Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		SRPDETIEN	Reserved	SECHGIEN	VBCHGIEN	AVLDCHGIEN	BVLDCHGIEN
7	6	5	4	3	2	1	0
HOSTIEN	PDEVIEN	IDCHGIEN	GOIDLEIEN	HNPFIEN	SRPFIEN	VBEIEN	ROLECHGIEN

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13]	**SRPDETIEN**	**SRP Detected Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[12]	**Reserved**	Reserved.
[11]	**SECHGIEN**	**SESSEND Status Changed Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[10]	**VBCHGIEN**	**VBVALID Status Changed Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[9]	**AVLDCHGIEN**	**A-Device Session Valid Status Change (From High To Low Or From Low To High) Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[8]	**BVLDCHGIEN**	**B-Device Session Valid Status Change (From High To Low Or From Low To High) Interrupt Enable Bit** 0 =Interrupt Disabled. 1 = Interrupt Enabled.
[7]	**HOSTIEN**	**Act As Host Interrupt Enable Bit** 0= This device as a host interrupt Disabled. 1 = This device as a host interrupt Enabled.
[6]	**PDEVIEN**	**Act As Peripheral Interrupt Enable Bit** 0 = This device as a peripheral interrupt Disabled. 1 = This device as a peripheral interrupt Enabled.

Bits	Description	
[5]	**IDCHGIEN**	**IDSTS Changed Interrupt Enable** 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[4]	**GOIDLEIEN**	**OTG Device Goes IDLE State Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled. **Note:** Going to idle state means going to a_idle or b_idle state. Please refer to A-device state diagram and B-device state diagram in OTG spec.
[3]	**HNPFIEN**	**HNP Fail Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[2]	**SRPFIEN**	**SRP Fail Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[1]	**VBEIEN**	**VBUS Error Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled. **Note:** VBUS error means going to a_vbus_err state. Please refer to A-device state diagram in OTG spec.
[0]	**ROLECHGIEN**	**Role(Host Or Peripheral) Changed Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled.

OTG Interrupt Status Register (OTG_INTSTS)

Register	Offset	R/W	Description	Reset Value
OTG_INTSTS	OTG_BA+0x0C	R/W	OTG Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		SRPDETIF	Reserved	SECHGIF	VBCHGIF	AVLDCHGIF	BVLDCHGIF
7	6	5	4	3	2	1	0
HOSTIF	PDEVIF	IDCHGIF	GOIDLEIF	HNPFIF	SRPFIF	VBEIF	ROLECHGIF

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13]	**SRPDETIF**	**SRP Detected Interrupt Status** 0 = SRP not detected. 1 = SRP detected. **Note:** Write 1 to clear this status.
[12]	**Reserved**	Reserved.
[11]	**SECHGIF**	**SESSEND State Change Interrupt Status** 0 = Session end not toggled. 1 = SESSEND from high to low or from low to high. **Note:** Write 1 to clear this flag.
[10]	**VBCHGIF**	**VBVALID State Change Interrupt Status** 0 = VBUS_VLD not toggled. 1 = VBUS_VLD from high to low or from low to high. **Note:** Write 1 to clear this flag.
[9]	**AVLDCHGIF**	**A-Device Session Valid State Change Interrupt Status** 0 = AVLD not toggled. 1 = AVLD from high to low or low to high. **Note:** Write 1 to clear this flag.
[8]	**BVLDCHGIF**	**B-Device Session Valid State Change Interrupt Status** 0 = BVLD not toggled. 1 = BVLD from high to low or low to high. **Note:** Write 1 to clear this status.
[7]	**HOSTIF**	**Act As Host Interrupt Status** 0= This device does not act as a host. 1 = This device acts as a host.

Bits	Description	
		Note: Write 1 to clear this flag.
[6]	**PDEVIF**	**Act As Peripheral Interrupt Status** 0= This device does not act as a peripheral. 1 = This device acts as a peripheral. **Note:** Write 1 to clear this flag.
[5]	**IDCHGIF**	**ID State Change Interrupt Status** 0 = IDSTS not toggled. 1 = IDSTS from high to low or from low to high. **Note1:** BUSREQ (OTG_CTL[1]) will be cleared when IDDIG is high. **Note2:** Write 1 to clear this flag.
[4]	**GOIDLEIF**	**OTG Device Goes IDLE Interrupt Status** Flag is set if the OTG device transfers from non-idle state to idle state. The OTG device will be neither a host nor a peripheral. 0 = OTG device does not go back to idle state(a_idle or b_idle). 1 = OTG device go back to idle state(a_idle or b_idle). **Note:** Write 1 to clear this flag.
[3]	**HNPFIF**	**HNP Fail Interrupt Status** When A-device has granted B-device to be host and USB bus in SE0 state, this bit will be set in specified interval (b_ase0_brst_tmr, defined in OTG spec. specification), A-device does not signal connect signal. **Note:** Write 1 to clear this flag.
[2]	**SRPFIF**	**SRP Fail Interrupt Status** After initiating SRP, an OTG B-device will wait at least TB_SRP_FAIL min, defined in OTG specification, for the OTG A-device respond This flag is set when the OTG B-device didn't get the response from the remote A-device to turn VBUS on and generate a bus reset. **Note:** Write 1 to clear this flag.
[1]	**VBEIF**	**VBUS Error Interrupt Status** This flag will be set in one of two conditions ● One case is that voltage on VBUS cannot reach a minimum valid threshold 4.4V within a maximum time of 100ms after OTG A device starting to drive. ● The other case is that the supplied VBUS drops below a minimum valid threshold due to the overcurrent condition. **Note:** Write 1 to clear this flag and recover from the VBUS error state.
[0]	**ROLECHGIF**	**OTG Role Change Interrupt Status** This flag is set when the role of an OTG device changed from a host to a peripheral, or changed from a peripheral to a host **Note:** Write 1 to clear this flag.

OTG Function Status Register (OTG_STATUS)

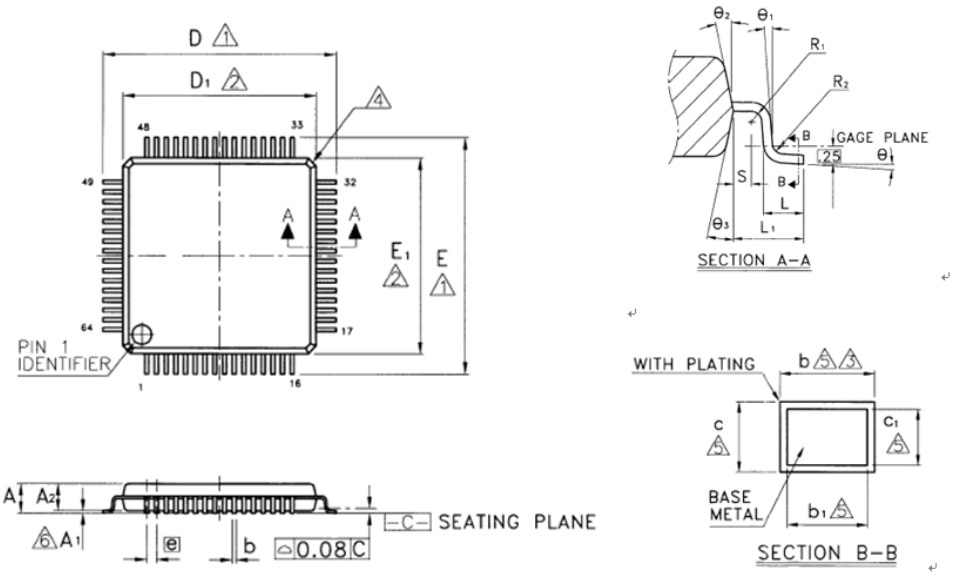
Register	Offset	R/W	Description	Reset Value
OTG_STATUS	OTG_BA+0x10	R	Functional Status Register	0x0000_0006

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		VBUSVLD	AVLD	BVLD	SESSEND	IDSTS	OVERCUR

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**VBUSVLD**	**VBUS Valid Status** 0 = VBUS < 4.4V. 1 = VBUS > 4.75V.
[4]	**AVLD**	**A-Device Session Valid Status** 0 = VBUS < 0.8V. 1 = VBUS > 2V.
[3]	**BVLD**	**B-Device Session Valid Status** 0 = VBUS < 0.8V. 1 = VBUS > 4V.
[2]	**SESSEND**	**Session End Status** 0 = VBUS > 0.8V. 1 = VBUS < 0.2V.
[1]	**IDSTS**	**ID Pin State Of Mini-B/Micro-Plug** 0 = Mini-A/Micro-A plug is attached. 1 = Mini-B/Micro-B plug is attached.
[0]	**OVERCUR**	**Overcurrent Condition** The voltage on VBUS cannot reach a minimum VBUS valid threshold, 4.4V minimum, within a maximum time of 100ms after OTG A device starting to drive 0 = OTG A-device drives VBUS successfully. 1 = Overcurrent condition occurred.

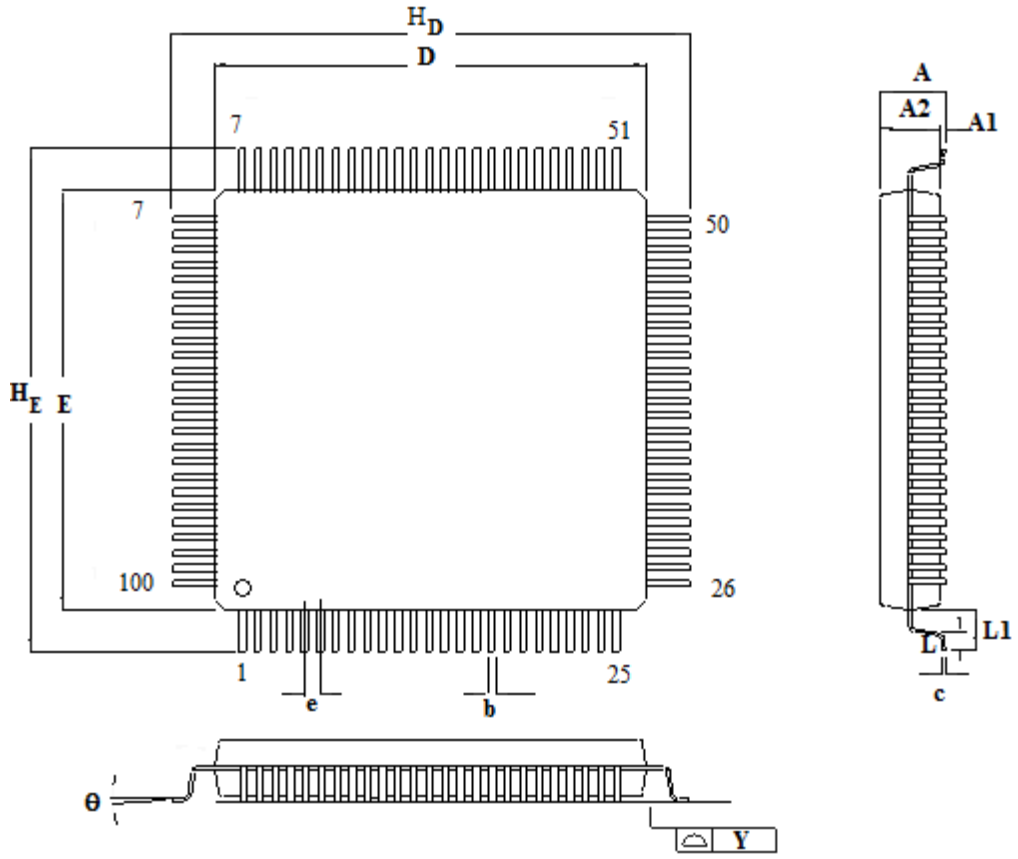
7 PACKAGE DIMENSIONS

7.1 LQFP 64L (10x10x1.4 mm footprint 2.0 mm)



Symbol	Dimension in mm			Dimension in inch		
	Min	Nom	Max	Min	Nom	Max
A	—	—	1.60	—	—	0.063
A1	0.05	—	0.15	0.002	—	0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
b	0.17	0.22	0.27	0.007	0.009	0.011
b1	0.17	0.20	0.23	0.007	0.008	0.009
c	0.09	—	0.20	0.004	—	0.008
c1	0.09	—	0.16	0.004	—	0.006
D	12.00 BSC			0.472 BSC		
D1	10.00 BSC			0.394 BSC		
E	12.00 BSC			0.472 BSC		
E1	10.00 BSC			0.394 BSC		
e	0.50 BSC			0.020 BSC		
L	0.45	0.60	0.75	0.018	0.024	0.030
L1	1.00 REF			0.039 REF		
R1	0.08	—	—	0.003	—	—
R2	0.08	—	0.20	0.003	—	0.008
S	0.20	—	—	0.008	—	—
θ	0°	3.5°	7°	0°	3.5°	7°
θ1	0°	—	—	0°	—	—
θ2	12°TYP			12°TYP		
θ3	12°TYP			12°TYP		

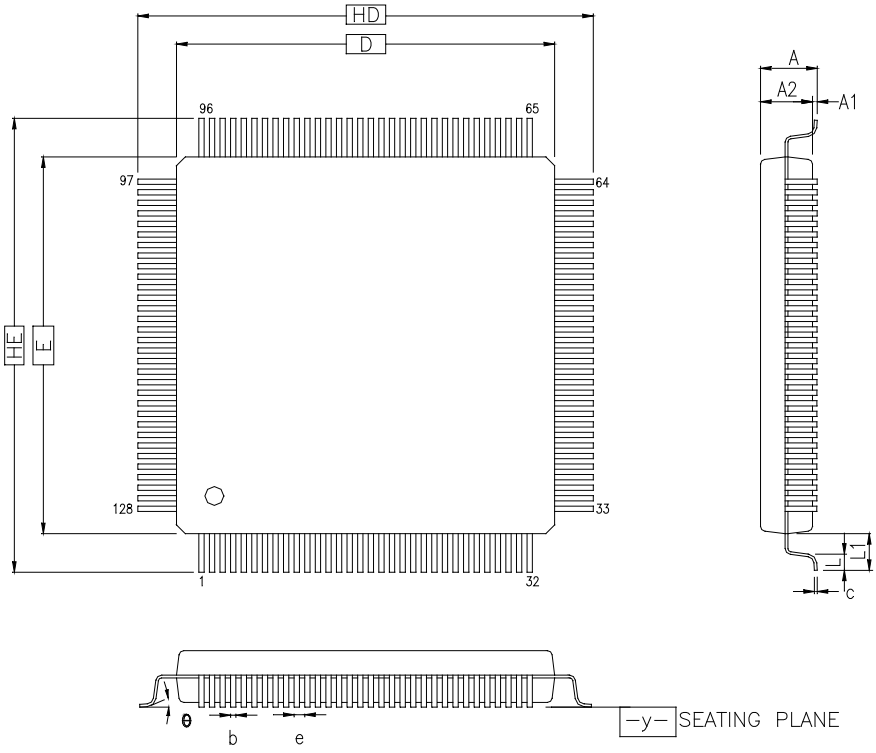
7.2 LQFP 100L (14x14x1.4 mm footprint 2.0 mm)



Controlling Dimension : Millimeters

SYMBOLS	MIN.	NOM.	MAX.
A	-	-	1.60
A1	0.05	-	-
A2	1.35	1.40	1.45
b	0.17	0.22	0.27
c	0.10	0.15	0.20
D	13.90	14.00	14.10
E	13.90	14.00	14.10
e	-	0.50	-
HD	15.80	16.00	16.20
HE	15.80	16.00	16.20
L	0.45	0.60	0.75
L1	-	1	-
y	-	-	0.10
θ	0°	-	7°

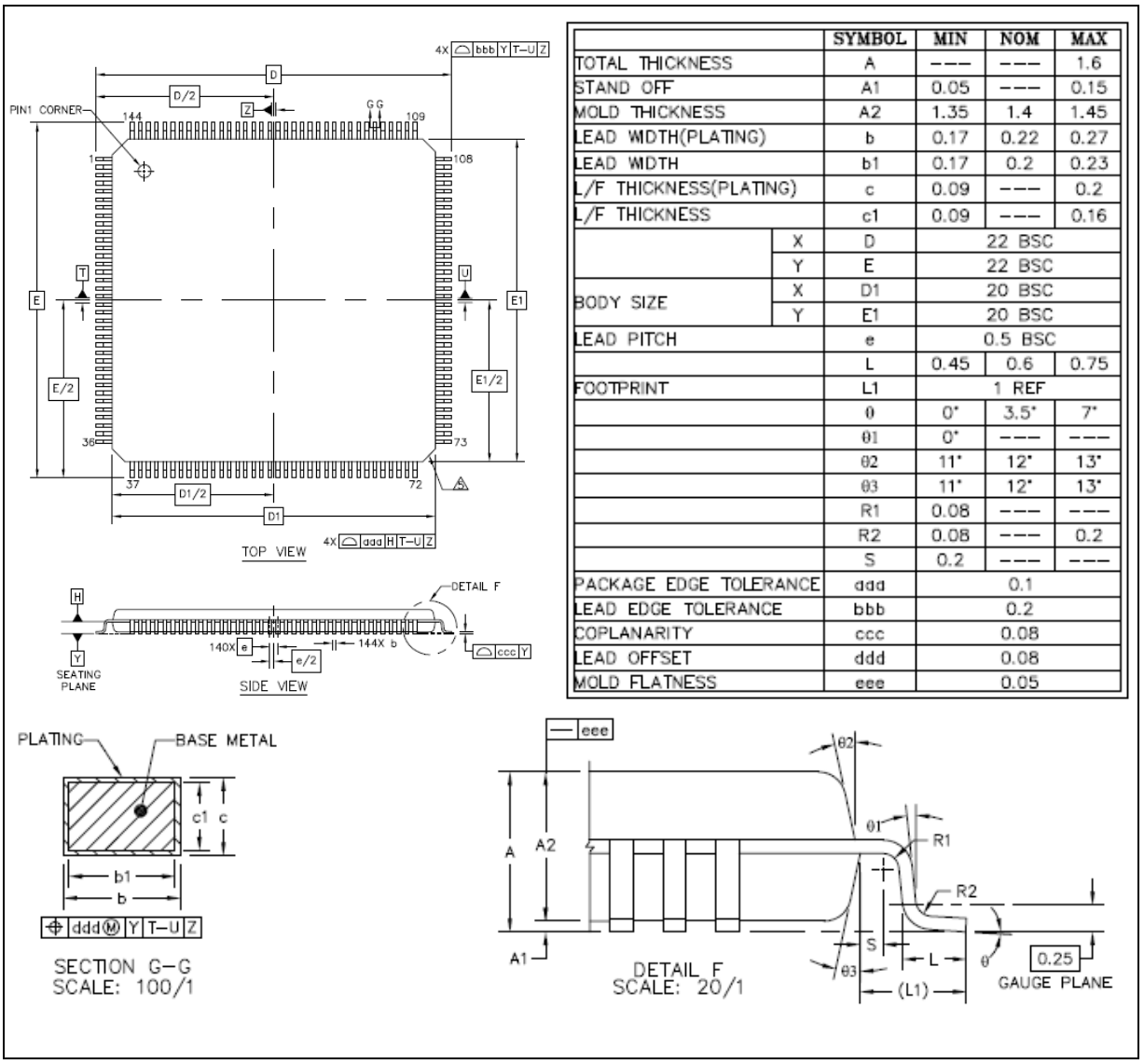
7.3 LQFP 128L (14x14x1.4 mm footprint 2.0 mm)



COTROL DIMENSIONS ARE IN MILLIMETERS.

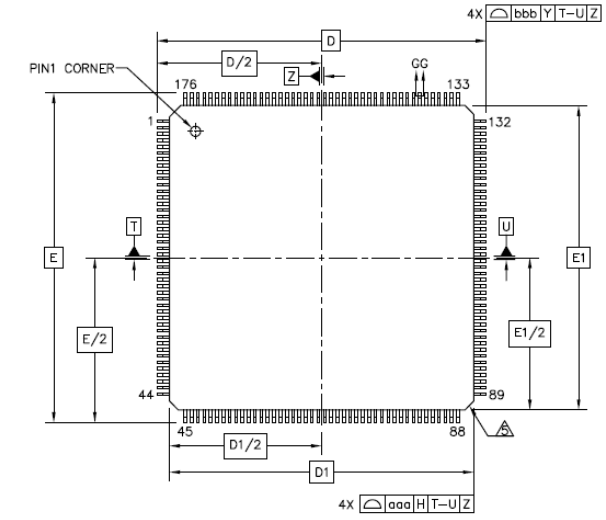
SYMBOL	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	—	—	1.60	—	—	0.063
A1	0.05	—	0.15	0.002	—	0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
HD	16.00 BSC.			0.630 BSC.		
D	14.00 BSC.			0.551 BSC.		
HE	16.00 BSC.			0.630 BSC.		
E	14.00 BSC.			0.551 BSC.		
b	0.13	0.16	0.23	0.005	0.006	0.009
e	0.40 BSC.			0.016 BSC.		
θ	0°	3.5°	7°	0°	3.5°	7°
c	0.09	—	0.20	0.004	—	0.008
L	0.45	0.60	0.75	0.018	0.024	0.030
L_1	1.00 REF			0.039 REF		
y	–	–	0.1	–	–	0.004

7.4 LQFP 144L (20x20x1.4 mm footprint 2.0 mm)

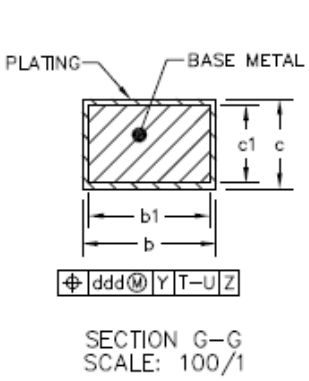


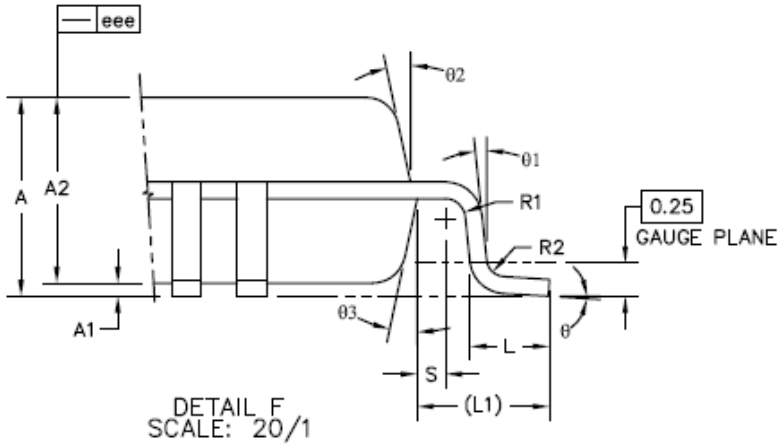
		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	---	---	1.6
STAND OFF		A1	0.05	---	0.15
MOLD THICKNESS		A2	1.35	1.4	1.45
LEAD WIDTH(PLATING)		b	0.17	0.22	0.27
LEAD WIDTH		b1	0.17	0.2	0.23
L/F THICKNESS(PLATING)		c	0.09	---	0.2
L/F THICKNESS		c1	0.09	---	0.16
BODY SIZE	X	D	22 BSC		
	Y	E	22 BSC		
	X	D1	20 BSC		
	Y	E1	20 BSC		
LEAD PITCH		e	0.5 BSC		
		L	0.45	0.6	0.75
FOOTPRINT		L1	1 REF		
		θ	0°	3.5°	7°
		θ1	0°	---	---
		θ2	11°	12°	13°
		θ3	11°	12°	13°
		R1	0.08	---	---
		R2	0.08	---	0.2
		S	0.2	---	---
PACKAGE EDGE TOLERANCE		aaa	0.1		
LEAD EDGE TOLERANCE		bbb	0.2		
COPLANARITY		ccc	0.08		
LEAD OFFSET		ddd	0.08		
MOLD FLATNESS		eee	0.05		

7.5 LQFP 176L (24x24x1.4 mm footprint 2.0 mm)



		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	---	---	1.6
STAND OFF		A1	0.05	---	0.15
MOLD THICKNESS		A2	1.35	1.4	1.45
LEAD WIDTH(PLATING)		b	0.17	0.22	0.27
LEAD WIDTH		b1	0.17	0.2	0.23
L/F THICKNESS(PLATING)		c	0.09	---	0.2
L/F THICKNESS		c1	0.09	---	0.16
BODY SIZE	X	D	26 BSC		
	Y	E	26 BSC		
	X	D1	24 BSC		
	Y	E1	24 BSC		
LEAD PITCH		e	0.5 BSC		
		L	0.45	0.6	0.75
FOOTPRINT		L1	1 REF		
		θ	0°	3.5°	7°
		θ1	0°	---	---
		θ2	11°	12°	13°
		θ3	11°	12°	13°
		R1	0.08	---	---
		R2	0.08	---	0.2
		S	0.2 REF.		
PACKAGE EDGE TOLERANCE		aaa	0.1		
LEAD EDGE TOLERANCE		bbb	0.2		
COPLANARITY		ccc	0.08		
LEAD OFFSET		ddd	0.07		
MOLD FLATNESS		eee	0.05		





8 REVISION HISTORY

Revision	Date	Description
1.01	Nov. 1, 2013	Preliminary version
1.02	Nov.15, 2013	Editorial changes.
1.03	Jan. 29, 2014	Updated the Clock Generator Global View Diagram. Updated the LQFP 64L package dimension.
1.04	Apr. 16, 2014	Modified the pin description table.
1.05	May 23, 2014	Renamed registers and bit fields.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.