ABOV SEMICONDUCTOR Co., Ltd.
8-BIT MICROCONTROLLERS

# MC96F7864

*User's Manual (Ver. 1.0)*

# REVISION HISTORY

**VERSION 0.0 (June 14, 2012)**

**VERSION 1.0 (January 9, 2013) This BooK**

Change '4.0/8.0mA (Typ/Max)' to "IDD1 @X-tal 12MHz, 5.5V" in DC Electrical characteristics.

Change '3.0/6.0mA (Typ/Max)' to "IDD1 @X-tal 10MHz, 3.3V" in DC Electrical characteristics.

Change '2.5/5.0mA (Typ/Max)' to "IDD1 @IRC 8MHz, 5.5V" in DC Electrical characteristics.

Change '1.7/3.4mA (Typ/Max)' to "IDD2 @X-tal 12MHz, 5.5V" in DC Electrical characteristics.

Change '1.0/2.0mA (Typ/Max)' to "IDD2 @X-tal 10MHz, 3.3V" in DC Electrical characteristics.

Change '1.0/2.0mA (Typ/Max)' to "IDD2 @IRC 8MHz, 5.5V" in DC Electrical characteristics.

Change '3.5uA/7.0uA (Typ/Max)' to "IDD4 @SX-tal 32.768KHz" in DC Electrical characteristics.

Change '1.024MHz/16.384MHz (Min/Max)' to "$f_{VCO}$" in Phase Locked-Loop characteristics.

Change 'R=6.8k/C1=820pF/C2=10nF' in Phase Locked-Loop characteristics.

Change '2.4V/VDD (Min/Max)' to "$V_{IH3}$" in DC Electrical characteristics.

Change '±1.5/±2.5/±3.5 (0°C~ +50°C/ -20°C ~ +85°C/-40°C ~ +85°C)' to "IRC Tolerance" in Internal RC Oscillator characteristics.

Change 'Typx0.93/Typx1.07(Min/Max)' to "VLC3" in LCD Voltage Characteristics.

Change 'Typ-0.2V/Typ+0.2V(Min/Max)' to "LCD Mid Bias Voltage" in LCD Voltage Characteristics.

Change '14/24uA(Typ/Max)' to "LVR/LVI Both Current" in LVR/LVI Characteristics.

Change '10/18uA(Typ/Max)' to "LVR/LVI One Current" in LVR/LVI Characteristics.

Add 'Recommanded Circuit and Layout With SMPS Power' in Electrical Characteristics.

Change ' ±6 LSB' to Integral Linear Error in ADC Characteristics.

Change ' ±5 LSB' to ZOE/FSE in ADC Characteristics.

Remove DAC/FADPCM Block.

Add a Package Type "MC96F7864L14 (64LQFP-1414)"

Version 1.0
Published by FAE Team

# Table of Contents

1. Overview ..... 11
1.1 Description ..... 11
1.2 Features ..... 12
1.3 Ordering Information ..... 14
1.4 Development Tools ..... 15
2. Block Diagram ..... 17
3. Pin Assignment ..... 18
4. Package Diagram ..... 22
5. Pin Description ..... 26
6. Port Structures ..... 32
6.1 General Purpose I/O Port ..... 32
6.2 External Interrupt I/O Port ..... 33
7. Electrical Characteristics ..... 34
7.1 Absolute Maximum Ratings ..... 34
7.2 Recommended Operating Conditions ..... 34
7.3 A/D Converter Characteristics ..... 35
7.4 Power-On Reset Characteristics ..... 35
7.5 Low Voltage Reset and Low Voltage Indicator Characteristics ..... 36
7.6 Phase Locked Loop Characteristics ..... 37
7.7 Internal RC Oscillator Characteristics ..... 37
7.8 Internal Watch-Dog Timer RC Oscillator Characteristics ..... 37
7.9 LCD Voltage Characteristics ..... 38
7.10 DC Characteristics ..... 39
7.11 AC Characteristics ..... 41
7.12 SPI Characteristics ..... 42
7.13 UART Characteristics ..... 43
7.14 I2C Characteristics ..... 44
7.15 Data Retention Voltage in Stop Mode ..... 45
7.16 Internal Flash Rom Characteristics ..... 46
7.17 Input/Output Capacitance ..... 46
7.18 Main Clock Oscillator Characteristics ..... 47
7.19 Sub Clock Oscillator Characteristics ..... 48
7.20 Main Oscillation Stabilization Characteristics ..... 49
7.21 Sub Oscillation Characteristics ..... 49
7.22 Operating Voltage Range ..... 50
7.23 Recommended Circuit and Layout ..... 51
7.24 Recommanded Circuit and Layout With SMPS Power ..... 52
7.25 Typical Characteristics ..... 53
8. Memory ..... 56
8.1 Program Memory ..... 56
8.2 Data Memory ..... 58
8.3 XRAM Memory ..... 60
8.4 SFR Map ..... 61
9. I/O Ports ..... 73
9.1 I/O Ports ..... 73
9.2 Port Register ..... 73
9.3 P0 Port ..... 76
9.4 P1 Port ..... 79
9.5 P2 Port ..... 82
9.6 P3 Port ..... 85

9.7 P4 Port ..... 88
9.8 P5 Port ..... 92
9.9 P6 Port ..... 95
9.10 P7 Port ..... 98
9.11 P8 Port ..... 101
10. Interrupt Controller ..... 103
10.1 Overview ..... 103
10.2 External Interrupt ..... 104
10.3 Block Diagram ..... 105
10.4 Interrupt Vector Table ..... 106
10.5 Interrupt Sequence ..... 107
10.6 Effective Timing after Controlling Interrupt Bit ..... 108
10.7 Multi Interrupt ..... 109
10.8 Interrupt Enable Accept Timing ..... 110
10.9 Interrupt Service Routine Address ..... 110
10.10 Saving/Restore General-Purpose Registers ..... 110
10.11 Interrupt Timing ..... 111
10.12 Interrupt Register Overview ..... 112
10.13 Interrupt Register Description ..... 113
11. Peripheral Hardware ..... 122
11.1 Clock Generator ..... 122
11.2 Basic Interval Timer ..... 127
11.3 Watch Dog Timer ..... 130
11.4 Watch Timer ..... 133
11.5 Timer 0/1/2 ..... 136
11.6 Timer 3/4/5/6 ..... 147
11.7 Timer 7/8 ..... 157
11.8 10-Bit PWM Generator ..... 186
11.9 Buzzer Driver ..... 198
11.10 SPI 2/3 ..... 200
11.11 UART2/3/4 ..... 206
11.12 USI0/1 (UART + SPI + I2C) ..... 220
11.13 12-Bit A/D Converter ..... 258
11.14 LCD Driver ..... 264
12. Power Down Operation ..... 278
12.1 Overview ..... 278
12.2 Peripheral Operation in IDLE/STOP Mode ..... 278
12.3 IDLE Mode ..... 279
12.4 STOP Mode ..... 280
12.5 Release Operation of STOP Mode ..... 281
13. RESET ..... 283
13.1 Overview ..... 283
13.2 Reset Source ..... 283
13.3 RESET Block Diagram ..... 283
13.4 RESET Noise Canceller ..... 284
13.5 Power on RESET ..... 284
13.6 External RESETB Input ..... 287
13.7 Brown Out Detector Processor ..... 288
13.8 LVI Block Diagram ..... 289
14. On-chip Debug System ..... 294
14.1 Overview ..... 294
14.2 Two-Pin External Interface ..... 295
15. Flash Memory ..... 300

15.1 Overview ..... 300
16. Configure Option ..... 311
16.1 Configure Option Control ..... 311
17. APPENDIX ..... 312

# List of Figures

Figure 1.1 OCD Debugger and Pin Description ..... 15
Figure 1.2 PGMplusUSB (Single Writer) ..... 16
Figure 1.3 StandAlone PGMplus (Single Writer) ..... 16
Figure 1.4 StandAlone Gang8 (for Mass Production) ..... 16
Figure 2.1 Block Diagram ..... 17
Figure 3.1 MC96F7864L 80LQFP-1212 Pin Assignment ..... 18
Figure 3.2 MC96F7864L14 80LQFP-1414 Pin Assignment ..... 19
Figure 3.3 MC96F7664L 64LQFP-1010 Pin Assignment ..... 20
Figure 3.4 MC96F7664L14 64LQFP-1414 Pin Assignment ..... 21
Figure 4.1 80-Pin LQFP-1212 Package ..... 22
Figure 4.2 80-Pin LQFP-1414 Package ..... 23
Figure 4.3 64-Pin LQFP-1010 Package ..... 24
Figure 4.4 64-Pin LQFP-1414 Package ..... 25
Figure 6.1 General Purpose I/O Port ..... 32
Figure 6.2 External Interrupt I/O Port ..... 33
Figure 7.1 Input Timing for RESETB ..... 41
Figure 7.2 Input Timing for External Interrupts ..... 41
Figure 7.3 Input Timing for EC0, EC1, EC2, EC3, EC4, EC5, EC6, EC7 ..... 41
Figure 7.4 SPI Timing ..... 42
Figure 7.5 Waveform for UART Timing Characteristics ..... 43
Figure 7.6 Timing Waveform for the UART Module ..... 43
Figure 7.7 I2C Timing ..... 44
Figure 7.8 Stop Mode Release Timing when Initiated by an Interrupt ..... 45
Figure 7.9 Stop Mode Release Timing when Initiated by RESETB ..... 45
Figure 7.10 Crystal/Ceramic Oscillator ..... 47
Figure 7.11 External Clock ..... 47
Figure 7.12 Crystal Oscillator ..... 48
Figure 7.13 External Clock ..... 48
Figure 7.14 Clock Timing Measurement at XIN ..... 49
Figure 7.15 Clock Timing Measurement at SXIN ..... 49
Figure 7.16 Operating Voltage Range ..... 50
Figure 7.17 Recommended Circuit and Layout ..... 51
Figure 7.18 Recommended Circuit and Layout with SMPS Power ..... 52
Figure 7.19 RUN (IDD1 ) Current ..... 53
Figure 7.20 IDLE (IDD2) Current ..... 53
Figure 7.21 SUB RUN (IDD3) Current ..... 54
Figure 7.22 SUB IDLE (IDD4) Current ..... 54
Figure 7.23 STOP (IDD5) Current ..... 55
Figure 8.1 Program Memory ..... 57
Figure 8.2 Data Memory Map ..... 58
Figure 8.3 Lower 128 Bytes RAM ..... 59

Figure 8.4 XDATA Memory Area ..... 60
Figure 10.1 External Interrupt Description ..... 104
Figure 10.2 Block Diagram of Interrupt ..... 105
Figure 10.3 Interrupt Vector Address Table ..... 107
Figure 10.4 Effective Timing of Interrupt Enable Register ..... 108
Figure 10.5 Effective Timing of Interrupt Flag Register ..... 108
Figure 10.6 Effective Timing of Interrupt ..... 109
Figure 10.7 Interrupt Response Timing Diagram ..... 110
Figure 10.8 Correspondence between Vector Table Address and the Entry Address of ISP ..... 110
Figure 10.9 Saving/Restore Process Diagram and Sample Source ..... 110
Figure 10.10 Timing Chart of Interrupt Acceptance and Interrupt Return Instruction ..... 111
Figure 11.1 Clock Generator Block Diagram ..... 122
Figure 11.2 Phase Locked-Loop Circuit Diagram ..... 123
Figure 11.3 Basic Interval Timer Block Diagram ..... 127
Figure 11.4 Watch Dog Timer Interrupt Timing Waveform ..... 130
Figure 11.5 Watch Dog Timer Block Diagram ..... 131
Figure 11.6 Watch Timer Block Diagram ..... 133
Figure 11.7 8-Bit Timer/Counter Mode for Timer 0/1/2 (Where n = 0, 1, and 2) ..... 137
Figure 11.8 8-Bit Timer/Counter 0/1/2 Example (Where n = 0, 1, and 2) ..... 137
Figure 11.9 8-Bit PWM Mode for Timer 0/1/2 (Where n = 0, 1, and 2) ..... 138
Figure 11.10 PWM Output Waveforms in PWM Mode for Timer 0/1/2 (Where n = 0, 1, and 2) ..... 139
Figure 11.11 8-Bit Capture Mode for Timer 0/1/2 (Where n = 0, 1, and 2) ..... 140
Figure 11.12 Input Capture Mode Operation for Timer 0/1/2 (Where n = 0, 1, and 2) ..... 141
Figure 11.13 Express Timer Overflow in Capture Mode (Where n = 0, 1, and 2) ..... 141
Figure 11.14 8-Bit Timer 0/1/2 Block Diagram (Where n = 0, 1, and 2) ..... 142
Figure 11.15 16-Bit Timer/Counter Mode for Timer 3/4/5/6 (where n = 3,4,5, and 6) ..... 148
Figure 11.16 16-Bit Timer/Counter 3/4/5/6 Example (where n = 3,4,5, and 6) ..... 148
Figure 11.17 16-Bit Capture Mode for Timer 3/4/5/6 (where n = 3,4,5, and 6) ..... 149
Figure 11.18 Input Capture Mode Operation for Timer 3/4/5/6 (where n = 3,4,5, and 6) ..... 150
Figure 11.19 Express Timer Overflow in Capture Mode (where n = 3,4,5, and 6) ..... 150
Figure 11.20 16-Bit PPG Mode for Timer 3/4/5/6 (where n = 3,4,5, and 6) ..... 151
Figure 11.21 16-Bit PPG Mode Timming chart for Timer 3/4/5/6 (where n = 3,4,5, and 6) ..... 152
Figure 11.22 16-Bit Timer 3/4/5/6 Block Diagram (where n = 3,4,5, and 6) ..... 153
Figure 11.23 8-Bit Timer/Counter Mode for Timer 7/8 ..... 158
Figure 11.24 16-Bit Timer/Counter Mode for Timer 7 ..... 159
Figure 11.25 8-Bit Capture Mode for Timer 7/8 ..... 161
Figure 11.26 16-Bit Capture Mode for Timer 7 ..... 162
Figure 11.27 10-Bit PWM Mode (Force 6-ch) ..... 164
Figure 11.28 10-Bit PWM Mode (Force All-ch) ..... 165
Figure 11.29 Example of PWM at 4 MHz ..... 166
Figure 11.30 Example of Changing the Period in Absolute Duty Cycle at 4 MHz ..... 166
Figure 11.31 Example of PWM Output Waveform ..... 167
Figure 11.32 Example of PWM waveform in Back-to-Back mode at 4 MHz ..... 167
Figure 11.33 Example of Phase Correction and Frequency correction of PWM ..... 168
Figure 11.34 Example of PWM External Synchronization with BLNK Input ..... 168
Figure 11.35 Example of Force Drive All Channel with A-ch ..... 169
Figure 11.36 Example of Force Drive 6-ch Mode ..... 170
Figure 11.37 Example of PWM Delay ..... 172
Figure 11.38 Two 8-Bit Timer 7/8 Block Diagram ..... 173

Figure 11.39 16-Bit Timer 7 Block Diagram ..... 174
Figure 11.40 10-Bit PWM Timer 8 Block Diagram ..... 174
Figure 11.41 A Timing Chart of 10-bit PWM One-Shot Mode Without Auto-Enable ..... 187
Figure 11.42 A Timing Chart of 10-bit PWM One-Shot Mode With Auto-Enable ..... 188
Figure 11.43 Timing Chart of 10-bit PWM Repeat Mode Without Delay ..... 189
Figure 11.44 Timing Chart of 10-bit PWM Repeat Mode With Delay ..... 189
Figure 11.45 Timing Chart of Noise Filter Function ..... 190
Figure 11.46 Timing Chart of Shot Stop ..... 190
Figure 11.47 Timing Chart of Emergency Stop ..... 191
Figure 11.48 10-Bit PWM generator Block Diagram ..... 192
Figure 11.49 Buzzer Driver Block Diagram ..... 198
Figure 11.50 SPI 2/3 Block Diagram (where n = 2 and 3) ..... 200
Figure 11.51 SPI 2/3 Transmit/Receive Timing Diagram at CPHA = 0 (Where n = 2 and 3) ..... 202
Figure 11.52 SPI 2/3 Transmit/Receive Timing Diagram at CPHA = 1 (Where n = 2 and 3) ..... 202
Figure 11.53 UART Block Diagram(where n = 2,3, and 4) ..... 207
Figure 11.54 Clock Generation Block Diagram (where n = 2,3, and 4) ..... 208
Figure 11.55 Frame Format ..... 209
Figure 11.56 Start Bit Sampling (where n = 2,3, and 4) ..... 213
Figure 11.57 Sampling of Data and Parity Bit (where n = 2,3, and 4) ..... 213
Figure 11.58 Stop Bit Sampling and Next Start Bit Sampling (where n = 2,3, and 4) ..... 214
Figure 11.59 USI0/1 UART Block Diagram (Where n = 0 and 1) ..... 222
Figure 11.60 Clock Generation Block Diagram (USIn, where n = 0 and 1) ..... 223
Figure 11.61 Synchronous Mode SCKn Timing (USIn , where n = 0 and 1) ..... 224
Figure 11.62 Frame Format (USI0/1) ..... 225
Figure 11.63 Asynchronous Start Bit Sampling (USIn, where n = 0 and 1) ..... 229
Figure 11.64 Asynchronous Sampling of Data and Parity Bit (USIn, where n = 0 and 1) ..... 229
Figure 11.65 Stop Bit Sampling and Next Start Bit Sampling (USIn, where n = 0 and 1) ..... 230
Figure 11.66 USI0/1 SPI Clock Formats when CPHAn=0 (where n = 0 and 1) ..... 232
Figure 11.67 USI0/1 SPI Clock Formats when CPHAn=1 (where n = 0 and 1) ..... 233
Figure 11.68 USI0/1 SPI Block Diagram (where n = 0 and 1) ..... 234
Figure 11.69 Bit Transfer on the I2C-Bus (USIn, where n = 0 and 1) ..... 235
Figure 11.70 START and STOP Condition (USIn, where n = 0 and 1) ..... 236
Figure 11.71 Data Transfer on the I2C-Bus (USIn, where n = 0 and 1) ..... 236
Figure 11.72 Acknowledge on the I2C-Bus (USIn, where n = 0 and 1) ..... 237
Figure 11.73 Clock Synchronization during Arbitration Procedure (USIn, where n = 0 and 1) ..... 238
Figure 11.74 Arbitration Procedure of Two Masters (USIn, where n = 0 and 1) ..... 238
Figure 11.75 Formats and States in the Master Transmitter Mode (USIn, where n = 0 and 1) ..... 240
Figure 11.76 Formats and States in the Master Receiver Mode (USIn, where n = 0 and 1) ..... 242
Figure 11.77 Formats and States in the Slave Transmitter Mode (USIn, where n = 0 and 1) ..... 244
Figure 11.78 Formats and States in the Slave Receiver Mode (USIn, where n = 0 and 1) ..... 246
Figure 11.79 USI0/1 I2C Block Diagram (where n = 0 and 1) ..... 247
Figure 11.80 12-bit ADC Block Diagram ..... 258
Figure 11.81 A/D Analog Input Pin with Capacitor ..... 259
Figure 11.82 A/D Power Pin with Capacitor ..... 259
Figure 11.83 ADC Operation for Align Bit ..... 260
Figure 11.84 A/D Converter Operation Flow ..... 261
Figure 11.85 LCD Circuit Block Diagram ..... 265
Figure 11.86 LCD Signal Waveforms (1/2Duty, 1/2Bias) ..... 266
Figure 11.87 LCD Signal Waveforms (1/3Duty, 1/3Bias) ..... 267

Figure 11.88 LCD Signal Waveforms (1/4Duty, 1/3Bias) ..... 268
Figure 11.89 LCD Signal Waveforms (1/8Duty, 1/4Bias) ..... 269
Figure 11.90 Internal Resistor Bias Connection ..... 270
Figure 11.91 External Resistor Bias Connection ..... 271
Figure 11.92 Resistor Bias Connection for External VLCD ..... 272
Figure 11.93 Capacitor Bias Connection ..... 273
Figure 11.94 LCD Circuit Block Diagram ..... 274
Figure 12.1 IDLE Mode Release Timing by External Interrupt ..... 279
Figure 12.2 STOP Mode Release Timing by External Interrupt ..... 280
Figure 12.3 STOP Mode Release Flow ..... 281
Figure 13.1 RESET Block Diagram ..... 283
Figure 13.2 Reset noise canceller timer diagram ..... 284
Figure 13.3 Fast VDD Rising Time ..... 284
Figure 13.4 Internal RESET Release Timing On Power-Up ..... 284
Figure 13.5 Configuration Timing when Power-on ..... 285
Figure 13.6 Boot Process WaveForm ..... 285
Figure 13.7 Timing Diagram after RESET ..... 287
Figure 13.8 Oscillator generating waveform example ..... 287
Figure 13.9 Block Diagram of BOD ..... 288
Figure 13.10 Internal Reset at the power fail situation ..... 288
Figure 13.11 Configuration timing when BOD RESET ..... 289
Figure 13.12 LVI Diagram ..... 289
Figure 14.1 Block Diagram of On-Chip Debug System ..... 294
Figure 14.2 10-bit Transmission Packet ..... 295
Figure 14.3 Data Transfer on the Twin Bus ..... 296
Figure 14.4 Bit Transfer on the Serial Bus ..... 296
Figure 14.5 Start and Stop Condition ..... 297
Figure 14.6 Acknowledge on the Serial Bus ..... 297
Figure 14.7 Clock Synchronization during Wait Procedure ..... 298
Figure 14.8 Connection of Transmission ..... 299
Figure 15.1 Flash Program ROM Structure ..... 301

# List of Tables

Table 1-1 Ordering Information of MC96F7864 ..... 14
Table 5-1 Normal Pin Description ..... 26
Table 7-1 Absolute Maximum Ratings ..... 34
Table 7-2 Recommended Operating Conditions ..... 34
Table 7-3 A/D Converter Characteristics ..... 35
Table 7-4 Power-on Reset Characteristics ..... 35
Table 7-5 LVR and LVI Characteristics ..... 36
Table 7-6 Phase Locked Loop Characteristics ..... 37
Table 7-7 Internal RC Oscillator Characteristics ..... 37
Table 7-8 Internal WDTRC Oscillator Characteristics ..... 37
Table 7-9 LCD Voltage Characteristics ..... 38
Table 7-10 DC Characteristics ..... 39
Table 7-11 AC Characteristics ..... 41
Table 7-12 SPI Characteristics ..... 42
Table 7-13 UART Characteristics ..... 43
Table 7-14 I2C Characteristics ..... 44
Table 7-15 Data Retention Voltage in Stop Mode ..... 45
Table 7-16 Internal Flash Rom Characteristics ..... 46
Table 7-17 Input/Output Capacitance ..... 46
Table 7-18 Main Clock Oscillator Characteristics ..... 47
Table 7-19 Sub Clock Oscillator Characteristics ..... 48
Table 7-20 Main Oscillation Stabilization Characteristics ..... 49
Table 7-21 Sub Oscillation Stabilization Characteristics ..... 49
Table 8-1 SFR Map Summary ..... 61
Table 8-2 Extended SFR Map Summary ..... 62
Table 8-3 SFR Map ..... 63
Table 8-4 Extended SFR Map ..... 67
Table 9-1 Port Register Map ..... 74
Table 10-1 Interrupt Group Priority Level ..... 103
Table 10-2 Interrupt Vector Address Table ..... 106
Table 10-3 Interrupt Register Map ..... 113
Table 11-1 Clock Generator Register Map ..... 123
Table 11-2 Basic Interval Timer Register Map ..... 128
Table 11-3 Watch Dog Timer Register Map ..... 131
Table 11-4 Watch Timer Register Map ..... 134
Table 11-5 Timer 0/1/2 Operating Modes ..... 136
Table 11-6 Timer 0/1/2 Register Map ..... 143
Table 11-7 Timer 3/4/5/6 Operating Modes ..... 147
Table 11-8 Timer 3/4/5/6 Register Map (where n = 3,4,5, and 6) ..... 153
Table 11-9 Timer 7/8 Operating Modes ..... 157
Table 11-10 PWM Frequency vs. Resolution at 8 MHz ..... 163
Table 11-11 PWM Channel Polarity ..... 163
Table 11-12 Timer 7/8 Register Map ..... 175

Table 11-13 10-bit PWM generator Operating Modes ..... 186
Table 11-14 10-Bit PWM Generator Register Map ..... 193
Table 11-15 Buzzer Frequency at 8 MHz ..... 198
Table 11-16 Buzzer Driver Register Map ..... 199
Table 11-17 SPI 2/3 Register Map ..... 203
Table 11-18 Equations for Calculating Baud Rate Register Setting ..... 208
Table 11-19 UART Register Map (where n = 2,3, and 4) ..... 214
Table 11-20 Examples of UARTnBD Settings for Commonly Used Oscillator Frequencies ..... 219
Table 11-21 Equations for Calculating USI0/1 Baud Rate Register Setting ..... 223
Table 11-22 CPOLn Functionality (where n = 0 and 1) ..... 231
Table 11-23 USI0/1 Register Map (where n = 0 and 1) ..... 248
Table 11-24 Examples of USI0BD and USI1BD Settings for Commonly Used Oscillator Frequencies 257
Table 11-25 ADC Register Map ..... 261
Table 11-26 LCD Register Map ..... 274
Table 11-27 LCD Frame Frequency ..... 276
Table 12-1 Peripheral Operation during Power Down Mode ..... 278
Table 12-2 Power Down Operation Register Map ..... 282
Table 13-1 Reset State ..... 283
Table 13-2 Boot Process Description ..... 286
Table 13-3 Reset Operation Register Map ..... 290
Table 15-1Flash Memory Register Map ..... 302

# MC96F7864

## CMOS SINGLE-CHIP 8-BIT MICROCONTROLLER WITH 12-BIT A/D CONVERTER

## 1. Overview

### 1.1 Description

The MC96F7864 is advanced CMOS 8-bit microcontroller with 64k bytes of FLASH. This is powerful microcontroller which provides a highly flexible and cost effective solution to many embedded control applications. This provides the following features : 64k bytes of FLASH, 256 bytes of IRAM, 3,072 bytes of XRAM , general purpose I/O, basic interval timer, watchdog timer, 8/16-bit timer/counter, 16-bit PPG output, 8-bit PWM output, 10-bit PWM output, watch timer, buzzer driving port, SPI, UART, I2C, 12-bit A/D converter, LCD driver, on-chip POR, LVR, LVI, on-chip oscillator and clock circuitry. The MC96F7864 also supports power saving modes to reduce power consumption.

| Device Name | FLASH | XRAM | IRAM | ADC | I/O PORT | Package |
|---|---|---|---|---|---|---|
| MC96F7864L | 64k bytes | 3,072 bytes | 256 bytes | 12 channel | 71 | 80 LQFP-1212 |
| MC96F7864L14 | | | | 12 channel | 71 | 80 LQFP-1414 |
| MC96F7664L | | | | 10 channel | 55 | 64 LQFP-1010 |
| MC96F7664L14 | | | | 10 channel | 55 | 64 LQFP-1414 |

## 1.2 Features

- **CPU**
  - 8 Bit CISC Core (8051 Compatible)
- **ROM (FLASH) Capacity**
  - 64k Bytes
  - Flash with self read/write capability
  - Parity bit check function for flash fail detection
  - On chip debug and In-system programming (ISP)
  - Endurance : 10,000 times (Sector 0~1019)
    100,000 times (Sector 1020~1023)
- **256 Bytes IRAM**
- **3,072 Bytes XRAM**
  - (40 Bytes including LCD display RAM)
- **General Purpose I/O (GPIO)**
  - Normal I/O : 21 Ports
    (P0, P1, P6[6:2])
  - LCD shared I/O : 50 Ports
    (P2, P3, P4, P5, P6[1:0], P7, P8)
- **Basic Interval Timer (BIT)**
  - 8Bit × 1ch
  - **Watch Dog Timer (WDT)**
  - 8Bit × 1ch
  - 5kHz internal RC oscillator
- **10Bit PWM Generator**
  - Emergency/Shot stop available
- **Timer/ Counter**
  - 8Bit × 3ch (T0/T1/T2), 16Bit × 4ch (T3/T4/T5/T6)
  - 8Bit × 2ch (T7/T8) or 16 Bit × 1ch (T7)
- **Programmable Pulse Generation**
  - 8Bit PWM (by T0/T1/T2)
  - Pulse generation (by T3/T4/T5/T6)
  - 6-ch 10Bit PWM for Motor (by T8)
- **Watch Timer (WT)**
  - 3.91mS/0.25S/0.5S/1S/1M interval at 32.768kHz
- **Buzzer**
  - 8Bit × 1ch
- **SPI**
  - 8Bit × 2ch
- **UART**
  - 8Bit × 3ch
- **USI (UART + SPI + I2C)**
  - 8Bit UART × 2ch, 8Bit SPI × 2ch, and I2C × 2ch
- **LCD Driver**
  - 36 Segments and 8 Common terminals
  - Internal or external resistor bias
  - Capacitor bias (Voltage Booster)
  - 16-step contrast control
  - 1/2, 1/3, 1/4, 1/5, 1/6, and 1/8 duty selectable
  - 1/2, 1/3, and 1/4 bias selectable
- **12 Bit A/D Converter**
  - 12 Input channels
- **Power On Reset**
  - Reset release level (1.4V)
- **Low Voltage Reset**
  - 14 level detect (1.60V/ 2.00V/ 2.10V/ 2.20V/ 2.32V/ 2.44V/ 2.59V/ 2.75V/ 2.93V/ 3.14V/ 3.38V/ 3.67V/ 4.00V/ 4.40V)
- **Low Voltage Indicator**
  - 13 level detect (2.00V/ 2.10V/ 2.20V/ 2.32V/ 2.44V/ 2.59V/ 2.75V/ 2.93V/ 3.14V/ 3.38V/ 3.67V/ 4.00V/ 4.40V)
  - External reference detect
- **Interrupt Sources**
  - External Interrupts
    (EINT0~9, EINT10~18) (19)
  - Timer(0/1/2/3/4/5/6/7/8) (13)
  - WDT (1)
  - BIT (1)
  - WT (1)
  - PWM (3)
  - SPI 2/3 (2)
  - UART 2/3/4 (6)
  - USI0/1 (6)
  - ADC (1)
- **Internal RC Oscillator**
  - Inernal RC frequency:
    16MHz ±1.5% ($T_A$= 0 ~ +50°C)
- **Power Down Mode**
  - STOP, IDLE mode
- **Operating Voltage and Frequency**
  - 1.8V ~ 5.5V (@32 ~ 38kHz with X-tal)
  - 1.8V ~ 5.5V (@0.4 ~ 4.2MHz with X-tal)
  - 2.7V ~ 5.5V (@0.4 ~ 12.0MHz with X-tal)
  - 1.8V ~ 5.5V (@0.5 ~ 8.0MHz with Internal RC)
  - 2.0V ~ 5.5V (@0.5 ~ 16.0MHz with Internal RC)
  - 2.0V ~ 5.5V (@1.0 ~ 16.0MHz with PLL)
  - Voltage dropout converter included for core

- **Minimum Instruction Execution Time**
  - 125nS (@ 16MHz main clock)
  - 61μS (@t 32.768kHz sub clock)
- **Operating Temperature:** – 40 ~ + 85℃
- **Oscillator Type**
  - 0.4-12MHz Crystal or Ceramic for main clock
  - 32.768kHz Crystal for sub clock
  - Phase locked loop (Max. 16.4MHz with sub clock)
- **Package Type**
  - 80 LQFP-1212
  - 80 LQFP-1414
  - 64 LQFP-1010
  - 64 LQFP-1414
  - Pb-free package

## 1.3 Ordering Information

**Table 1-1 Ordering Information of MC96F7864**

| Device name | ROM size | IRAM size | XRAM size | Package |
|---|---|---|---|---|
| MC96F7864L | 64k bytes FLASH | 256 bytes | 3,072 bytes | 80 LQFP-1212 |
| MC96F7864L14 | | | | 80 LQFP-1414 |
| MC96F7664L | | | | 64 LQFP-1010 |
| MC96F7664L14 | | | | 64 LQFP-1414 |

## 1.4 Development Tools

### 1.4.1 Compiler

We do not provide the compiler. Please contact the third parties.

The core of MC96F7864 is Mentor 8051. And, device ROM size is smaller than 64k bytes. Developer can use all kinds of third party's standard 8051 compiler.

### 1.4.2 OCD emulator and debugger

The OCD (On Chip Debug) emulator supports ABOV Semiconductor's 8051 series MCU emulation.

The OCD interface uses two-wire interfacing between PC and MCU which is attached to user's system. The OCD can read or change the value of MCU internal memory and I/O peripherals. And the OCD also controls MCU internal debugging logic, it means OCD controls emulation, step run, monitoring, etc.

The OCD Debugger program works on Microsoft-Windows NT, 2000, XP, Vista (32bit) operating system.

If you want to see more details, please refer to OCD debugger manual. You can download debugger S/W and manual from our web-site.

Connection:

- DSCL (MC96F7864 P14 port)
- DSDA (MC96F7864 P15 port)

OCD connector diagram: Connect OCD with user system

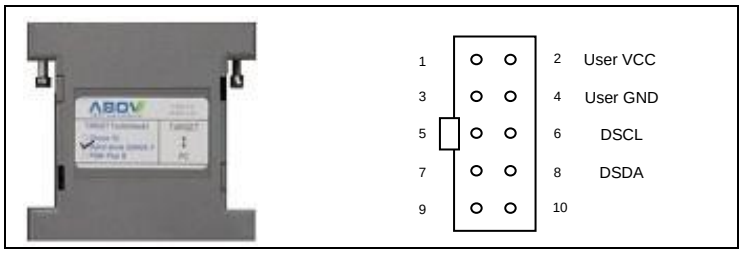



**Figure 1.1 OCD Debugger and Pin Description**

### 1.4.3 Programmer

Single programmer:

PGMplus USB: It programs MCU device directly.


**Figure 1.2 PGMplusUSB (Single Writer)**

StandAlone PGMplus: It programs MCU device directly.


**Figure 1.3 StandAlone PGMplus (Single Writer)**

OCD emulator: It can write code in MCU device too, because OCD debugging supports ISP (In System Programming).

It does not require additional H/W, except developer's target system.

Gang programmer:

It programs 8 MCU devices at once.

So, it is mainly used in mass production line.

Gang programmer is standalone type, it means it does not require host PC, after a program is downloaded from host PC to Gang programmer.


**Figure 1.4 StandAlone Gang8 (for Mass Production)**

## 2. Block Diagram

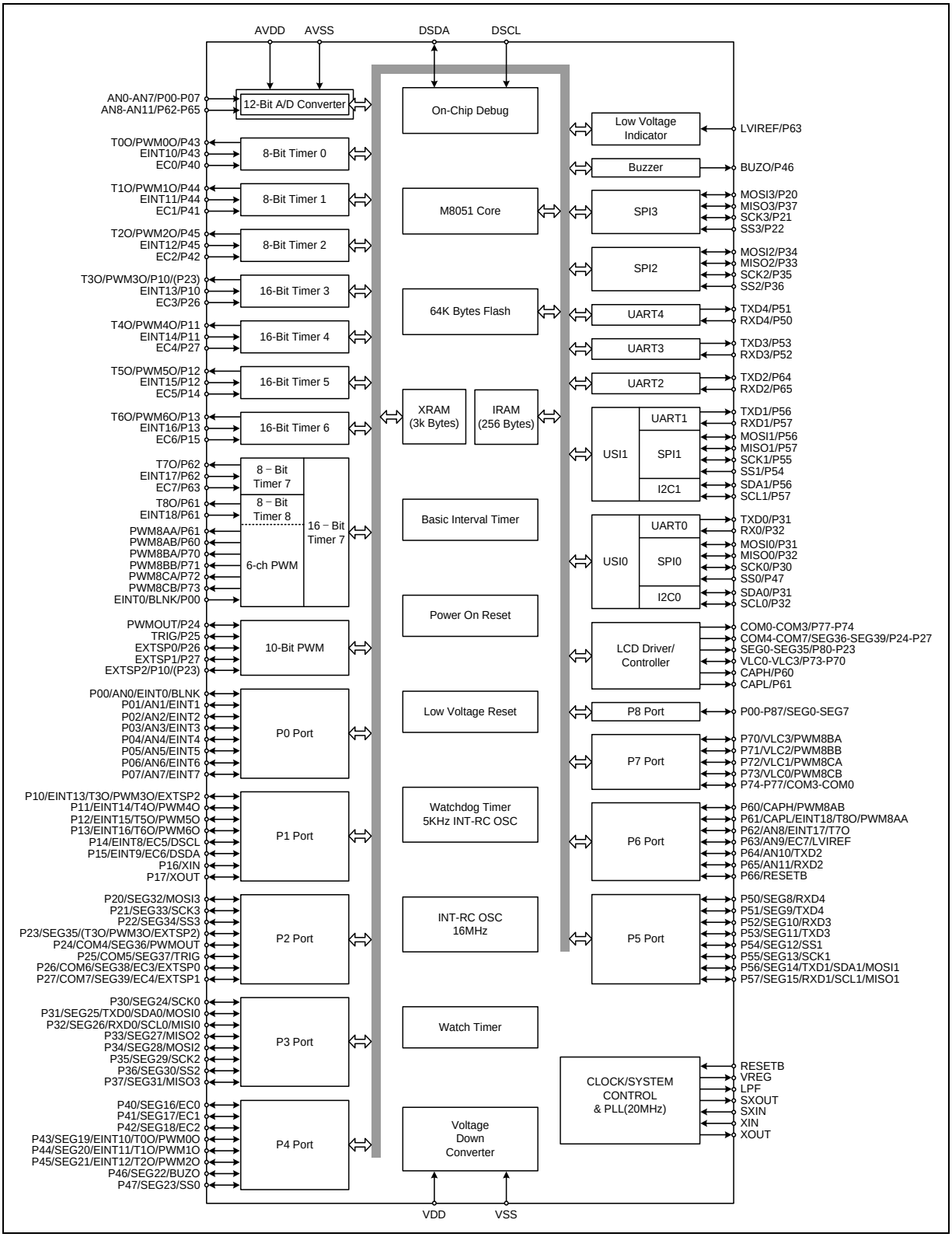



**Figure 2.1 Block Diagram**

NOTE) The P06-P07, P20-P23, P36-P37, and P8 are not in the 64-pin package.

## 3. Pin Assignment

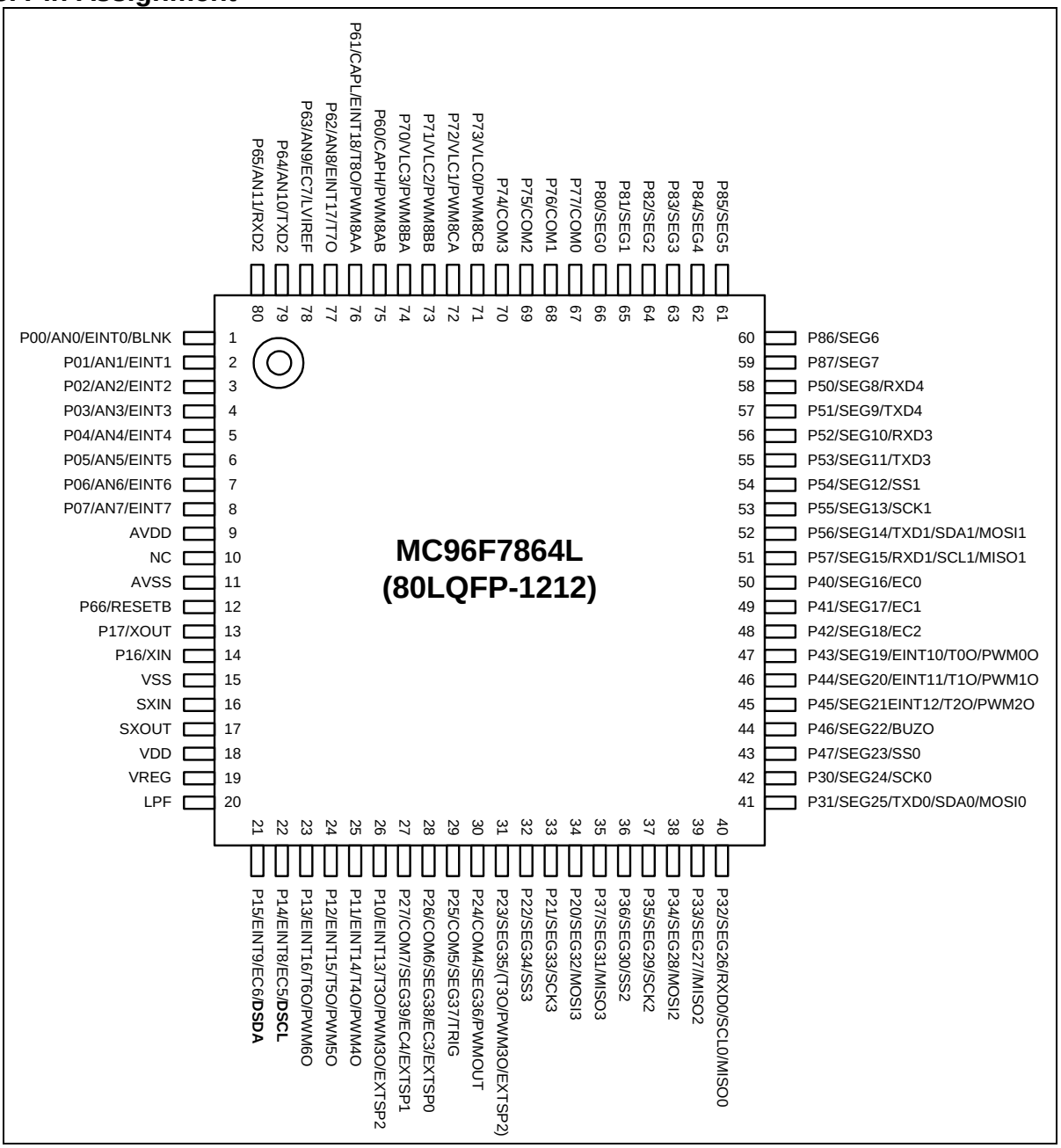



**Figure 3.1 MC96F7864L 80LQFP-1212 Pin Assignment**

NOTES) 1. On On-Chip Debugging, ISP uses P1[5:4] pin as DSDA, DSCL.
2. The pin in parentheses can be configured by software control.

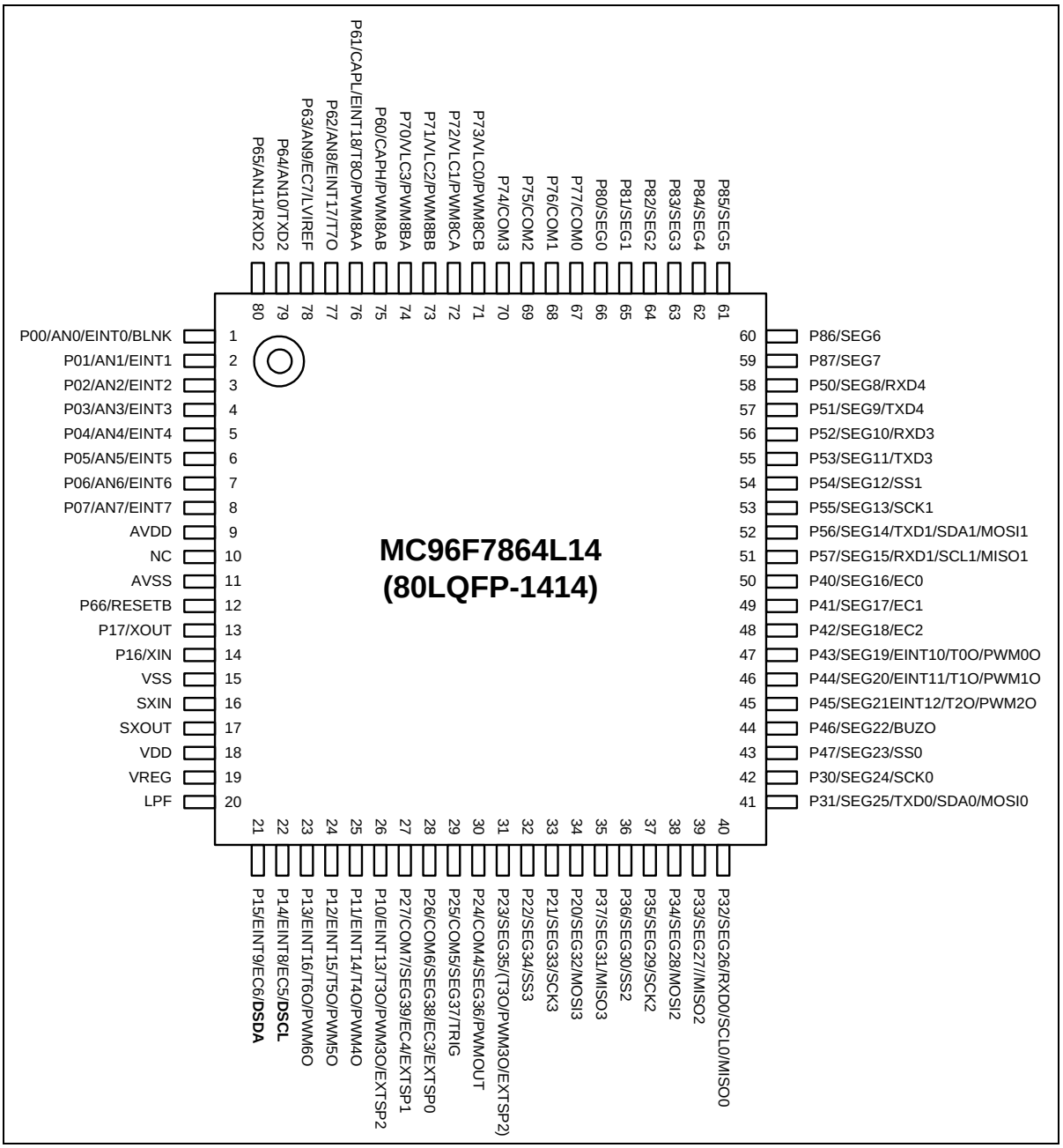



**Figure 3.2 MC96F7864L14 80LQFP-1414 Pin Assignment**

NOTES) 1. On On-Chip Debugging, ISP uses P1[5:4] pin as DSDA, DSCL.
2. The pin in parentheses can be configured by software control.

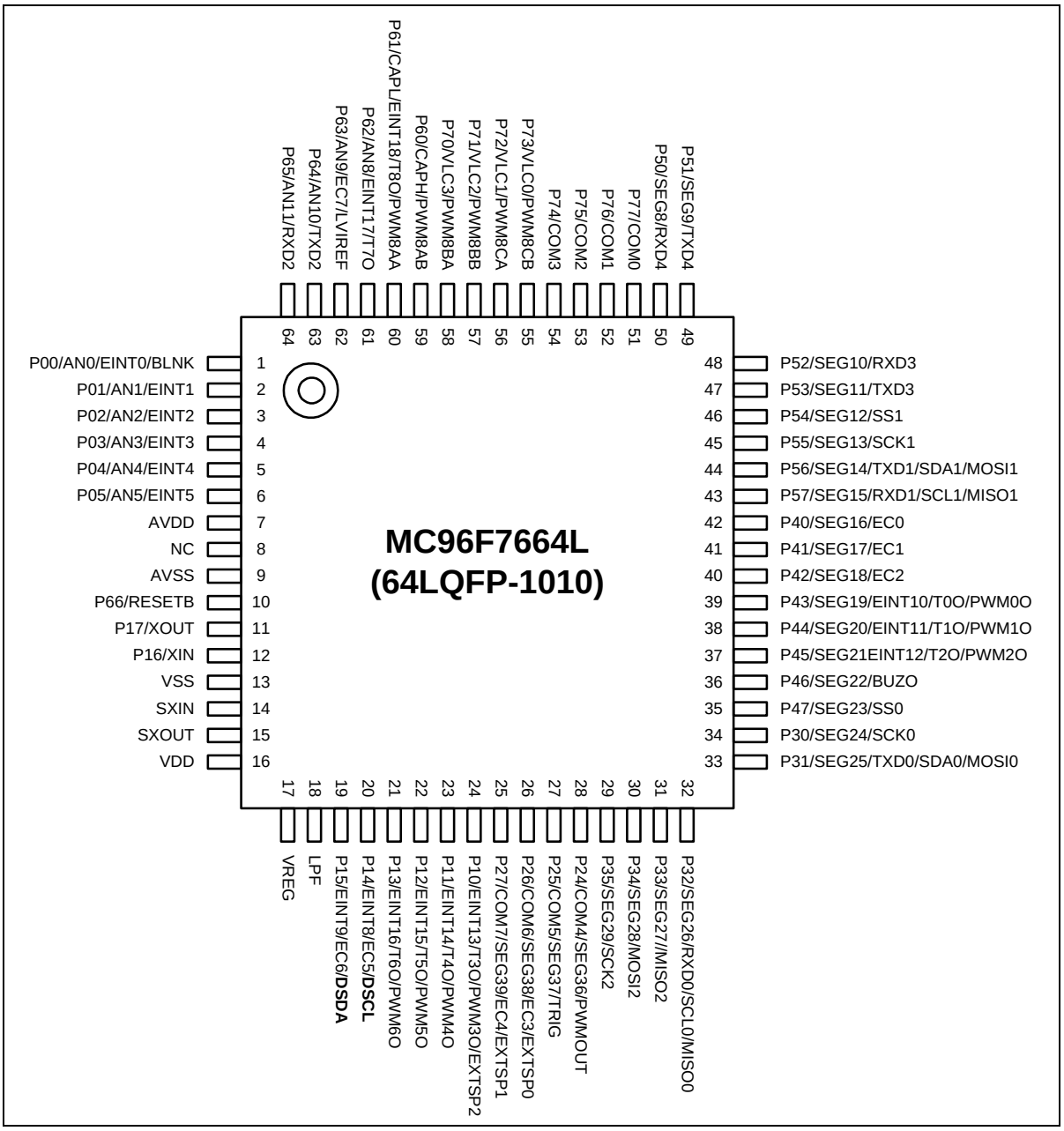



**Figure 3.3 MC96F7664L 64LQFP-1010 Pin Assignment**

NOTES) 1. On On-Chip Debugging, ISP uses P1[5:4] pin as DSDA, DSCL.
2. The P06-P07, P20-P23, P36-P37, and P8 pins should be selected as a push-pull output or an input with pull-up resistor by software control when the 64-pin package is used.

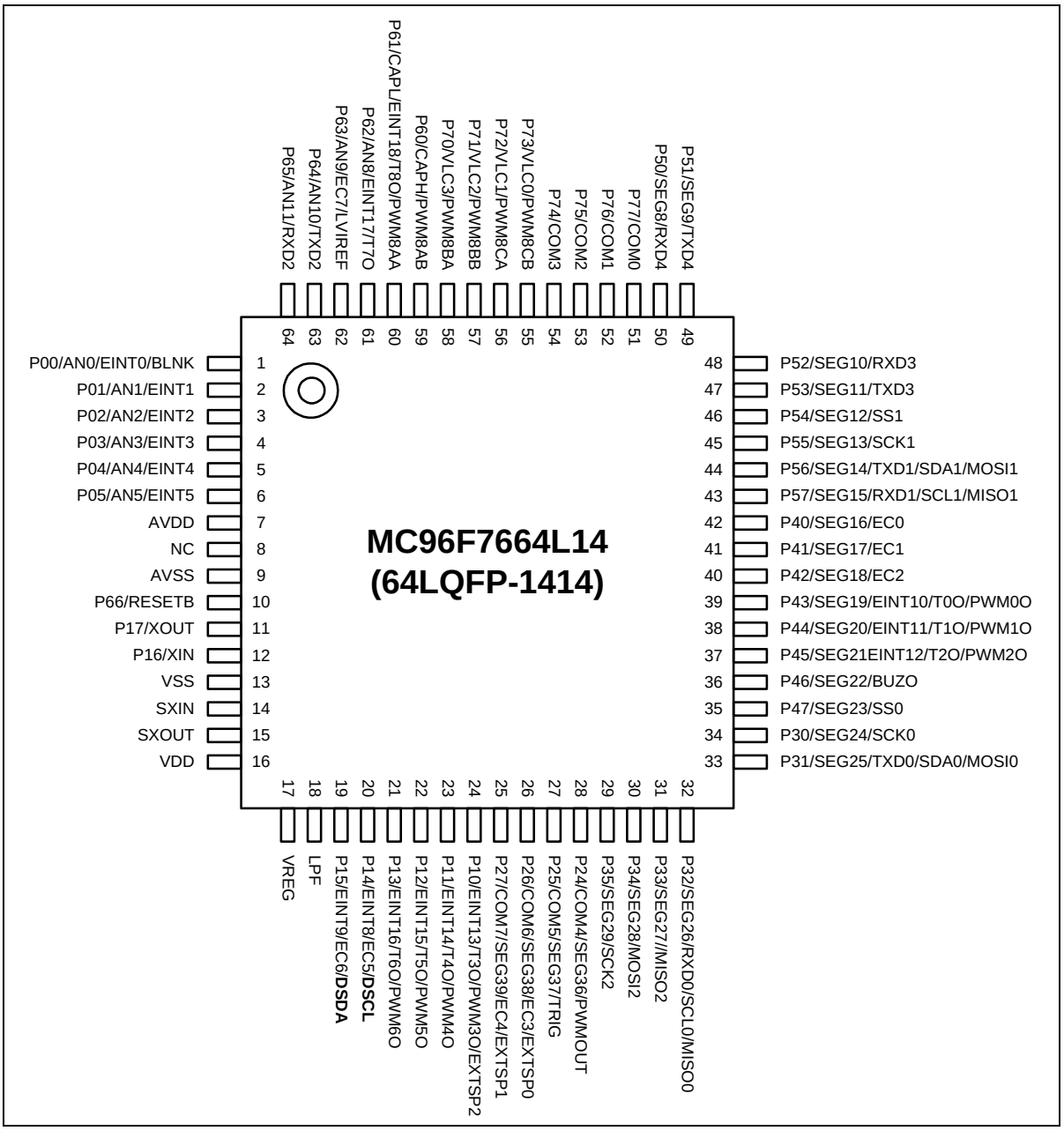
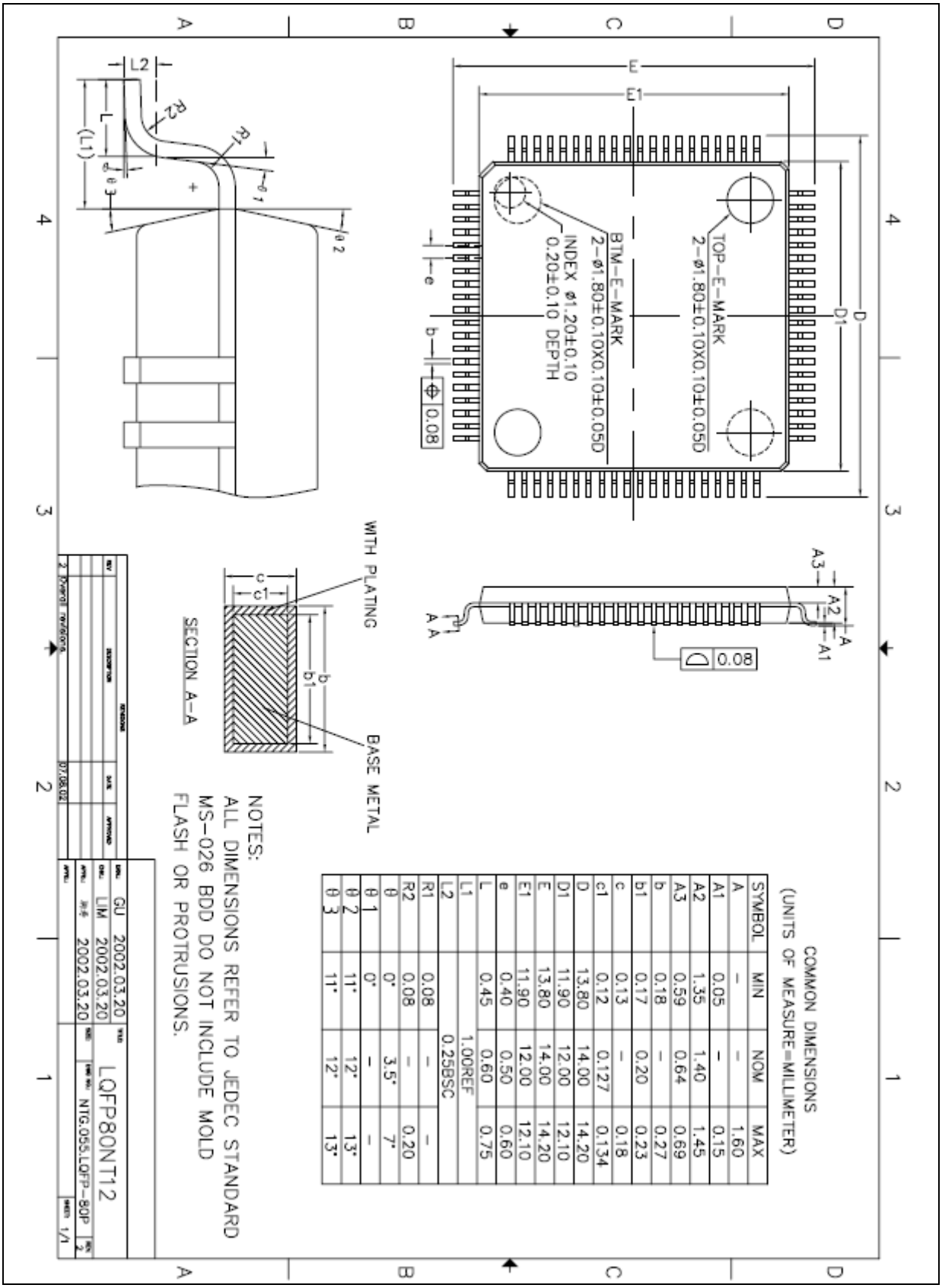



**Figure 3.4 MC96F7664L14 64LQFP-1414 Pin Assignment**

NOTES) 1. On On-Chip Debugging, ISP uses P1[5:4] pin as DSDA, DSCL.
2. The P06-P07, P20-P23, P36-P37, and P8 pins should be selected as a push-pull output or an input with pull-up resistor by software control when the 64-pin package is used.

## 4. Package Diagram




COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

| SYMBOL | MIN | NOM | MAX |
|---|---|---|---|
| A | – | – | 1.60 |
| A1 | 0.05 | – | 0.15 |
| A2 | 1.35 | 1.40 | 1.45 |
| A3 | 0.59 | 0.64 | 0.69 |
| b | 0.18 | – | 0.27 |
| b1 | 0.17 | 0.20 | 0.23 |
| c | 0.13 | – | 0.18 |
| c1 | 0.12 | 0.127 | 0.134 |
| D | 13.80 | 14.00 | 14.20 |
| D1 | 11.90 | 12.00 | 12.10 |
| E | 13.80 | 14.00 | 14.20 |
| E1 | 11.90 | 12.00 | 12.10 |
| e | 0.40 | 0.50 | 0.60 |
| L | 0.45 | 0.60 | 0.75 |
| L1 | | 1.00REF | |
| L2 | | 0.25BSC | |
| R1 | 0.08 | – | – |
| R2 | 0.08 | – | 0.20 |
| θ | 0° | 3.5° | 7° |
| θ 1 | 0° | – | – |
| θ 2 | 11° | 12° | 13° |
| θ 3 | 11° | 12° | 13° |

**Figure 4.1 80-Pin LQFP-1212 Package**

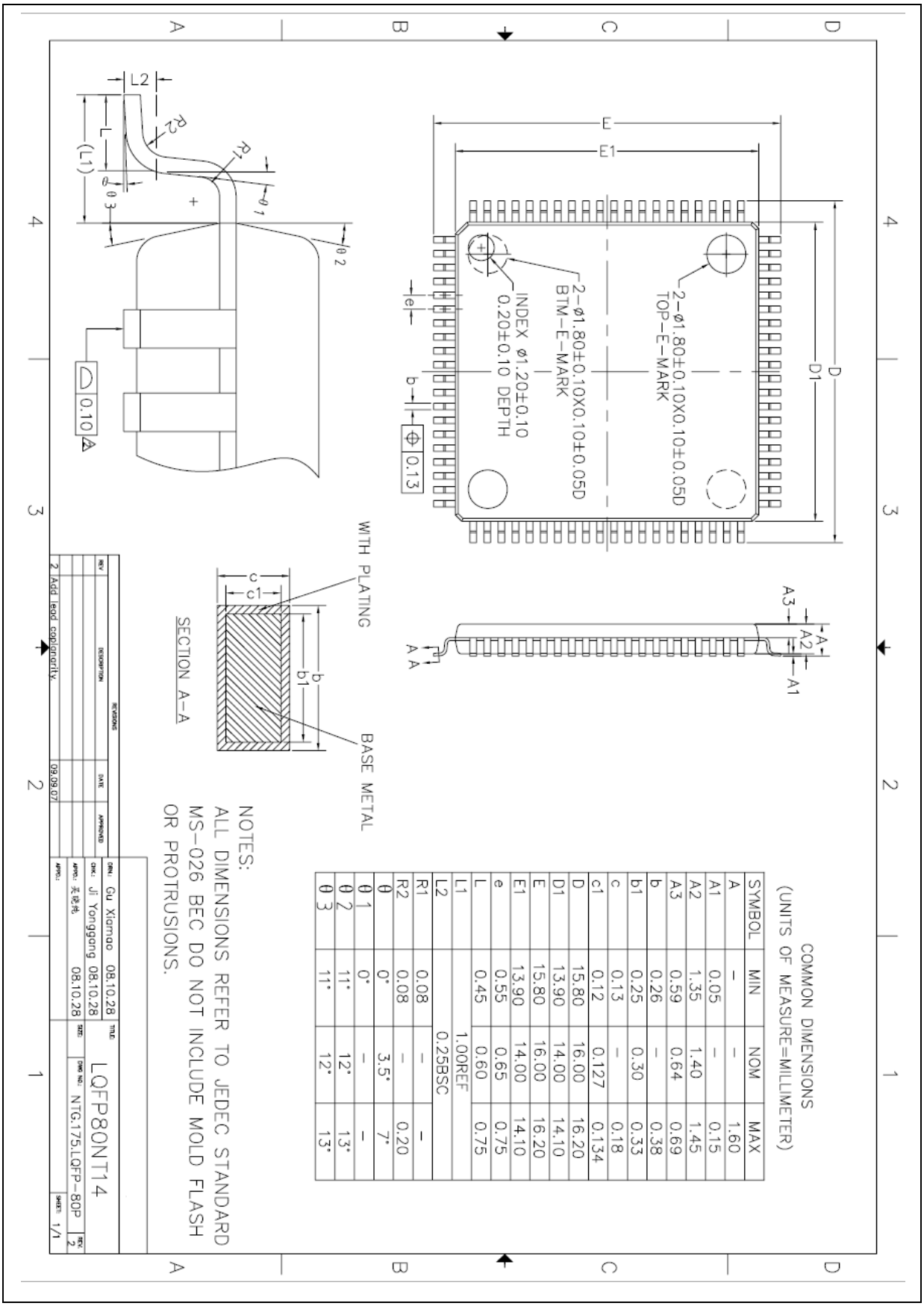



**Figure 4.2 80-Pin LQFP-1414 Package**

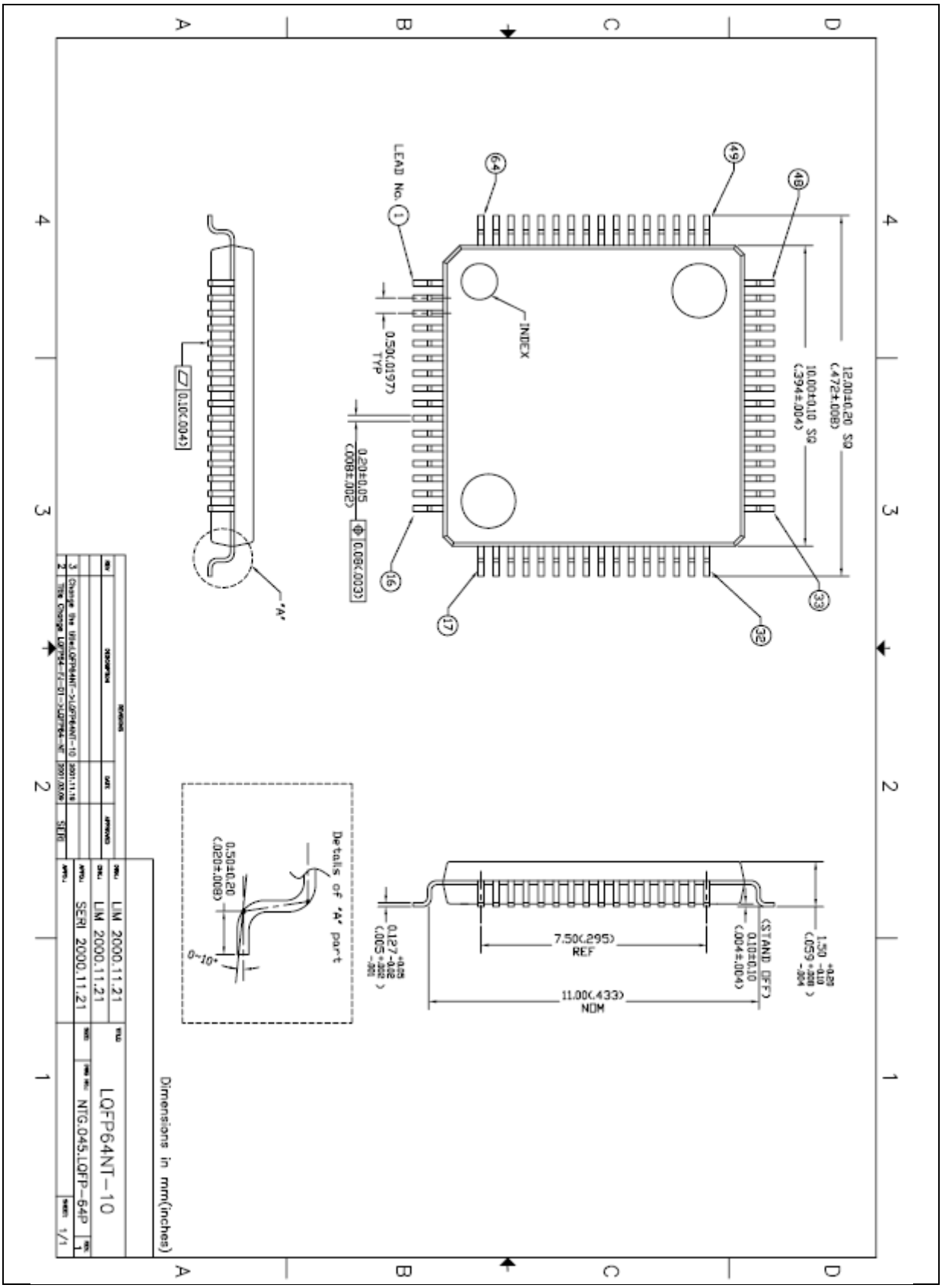



**Figure 4.3 64-Pin LQFP-1010 Package**

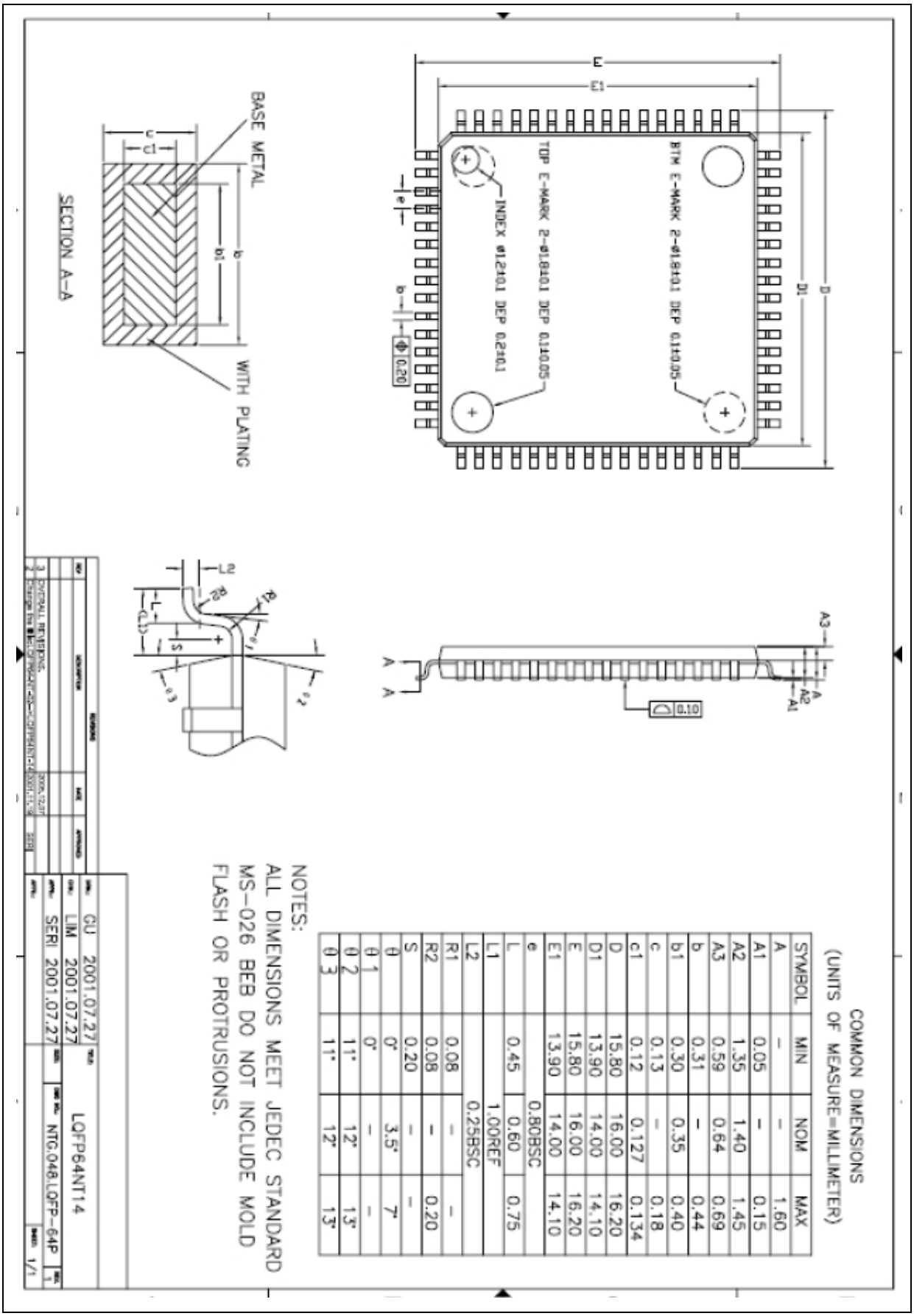



**Figure 4.4 64-Pin LQFP-1414 Package**

# 5. Pin Description

**Table 5-1 Normal Pin Description**

| PIN Name | I/O | Function | @RESET | Shared with |
|---|---|---|---|---|
| P00 | I/O | Port 0 is a bit-programmable I/O port which can be configured as a schmitt-trigger input, a push-pull output, or an open-drain output.<br>A pull-up resistor can be specified in 1-bit unit.<br>The P06-P07 are not in the 64-Pin package. | Input | AN0/EINT0/BLNK |
| P01 | | | | AN1/EINT1 |
| P02 | | | | AN2/EINT2 |
| P03 | | | | AN3/EINT3 |
| P04 | | | | AN4/EINT4 |
| P05 | | | | AN5/EINT5 |
| P06 | | | | AN6/EINT6 |
| P07 | | | | AN7/EINT7 |
| P10 | I/O | Port 1 is a bit-programmable I/O port which can be configured as a schmitt-trigger input, a push-pull output, or an open-drain output.<br>A pull-up resistor can be specified in 1-bit unit. | Input | EINT13/T3O/PWM3O/EXTSP2 |
| P11 | | | | EINT14/T4O/PWM4O |
| P12 | | | | EINT15/T5O/PWM5O |
| P13 | | | | EINT16/T6O/PWM6O |
| P14 | | | | EINT8/EC5/DSCL |
| P15 | | | | EINT9/EC6/DSDA |
| P16 | | | | XIN |
| P17 | | | | XOUT |
| P20 | I/O | Port 2 is a bit-programmable I/O port which can be configured as a schmitt-trigger input, a push-pull output, or an open-drain output.<br>A pull-up resistor can be specified in 1-bit unit.<br>The P20 – P23 are not in the 64-pin package. | Input | SEG32/MOSI3 |
| P21 | | | | SEG33/SCK3 |
| P22 | | | | SEG34/SS3 |
| P23 | | | | SEG35/(T3O/PWM3O/EXTSP2) |
| P24 | | | | COM4/SEG36/PWMOUT |
| P25 | | | | COM5/SEG37/TRIG |
| P26 | | | | COM6/SEG38/EC3/EXTSP0 |
| P27 | | | | COM7/SEG39/EC4/EXTSP1 |
| P30 | I/O | Port 3 is a bit-programmable I/O port which can be configured as a schmitt-trigger input, a push-pull output, or an open-drain output.<br>A pull-up resistor can be specified in 1-bit unit.<br>The P36-P37 are not in the 64-Pin package. | Input | SEG24/SCK0 |
| P31 | | | | SEG25/TXD0/SDA0/MOSI0 |
| P32 | | | | SEG26/RXD0/SCL0/MISO0 |
| P33 | | | | SEG27/MISO2 |
| P34 | | | | SEG28/MOSI2 |
| P35 | | | | SEG29/SCK2 |
| P36 | | | | SEG30/SS2 |
| P37 | | | | SEG31/MISO3 |
| P40 | I/O | Port 4 is a bit-programmable I/O port which can be configured as a schmitt-trigger input, a push-pull output, or an open-drain output.<br>A pull-up resistor can be specified in 1-bit unit. | Input | SEG16/EC0 |
| P41 | | | | SEG17/EC1 |
| P42 | | | | SEG18/EC2 |
| P43 | | | | SEG19/EINT10/T0O/PWM0O |
| P44 | | | | SEG20/EINT11/T1O/PWM1O |
| P45 | | | | SEG21/EINT12/T2O/PWM2O |
| P46 | | | | SEG22/BUZO |
| P47 | | | | SEG23/SS0 |

**Table 5-1 Normal Pin Description (Continued)**

| PIN Name | I/O | Function | @RESET | Shared with |
|---|---|---|---|---|
| P50 | I/O | Port 5 is a bit-programmable I/O port which can be configured as a schmitt-trigger input, a push-pull output, or an open-drain output.<br>A pull-up resistor can be specified in 1-bit unit. | Input | SEG8/RXD4 |
| P51 | | | | SEG9/TXD4 |
| P52 | | | | SEG10/RXD3 |
| P53 | | | | SEG11/TXD3 |
| P54 | | | | SEG12/SS1 |
| P55 | | | | SEG13/SCK1 |
| P56 | | | | SEG14/TXD1/SDA1/MOSI1 |
| P57 | | | | SEG15/RXD1/SCL1/MISO1 |
| P60 | I/O | Port 6 is a bit-programmable I/O port which can be configured as a schmitt-trigger input, a push-pull output, or an open-drain output.<br>A pull-up resistor can be specified in 1-bit unit. | Input | CAPH/PWM8AB |
| P61 | | | | CAPL/EINT18/T8O/PWM8AA |
| P62 | | | | AN8/EINT17/T7O |
| P63 | | | | AN9/EC7/LVIREF |
| P64 | | | | AN10/TXD2 |
| P65 | | | | AN11/RXD2 |
| P66 | | | | RESETB |
| P70 | I/O | Port 7 is a bit-programmable I/O port which can be configured as an input, a push-pull output, or an open-drain output.<br>A pull-up resistor can be specified in 1-bit unit. | Input | VLC3/PWM8BA |
| P71 | | | | VLC2/PWM8BB |
| P72 | | | | VLC1/PWM8CA |
| P73 | | | | VLC0/PWM8CB |
| P74 | | | | COM3 |
| P75 | | | | COM2 |
| P76 | | | | COM1 |
| P77 | | | | COM0 |
| P80 | I/O | Port 8 is a bit-programmable I/O port which can be configured as an input, a push-pull output, or an open-drain output.<br>A pull-up resistor can be specified in 1-bit unit.<br>The P8 is not in the 64-Pin package. | Input | SEG0 |
| P81 | | | | SEG1 |
| P82 | | | | SEG2 |
| P83 | | | | SEG3 |
| P84 | | | | SEG4 |
| P85 | | | | SEG5 |
| P86 | | | | SEG6 |
| P87 | | | | SEG7 |
| EINT0 | I/O | External interrupt inputs | Input | P00/AN0/BLNK |
| EINT1 | | | | P01/AN1 |
| EINT2 | | | | P02/AN2 |
| EINT3 | | | | P03/AN3 |
| EINT4 | | | | P04/AN4 |
| EINT5 | | | | P05/AN5 |
| EINT6 | | | | P06/AN6 |
| EINT7 | | | | P07/AN7 |
| EINT8 | | | | P14/EC5/DSCL |
| EINT9 | | | | P15/EC6/DSDA |
| EINT10 | I/O | External interrupt and Timer 0 capture input | Input | P43/SEG19/T0O/PWM0O |
| EINT11 | I/O | External interrupt and Timer 1 capture input | Input | P44/SEG20/T1O/PWM1O |
| EINT12 | I/O | External interrupt and Timer 2 capture input | Input | P45/SEG21/T2O/PWM2O |

**Table 5-1 Normal Pin Description (Continued)**

| PIN Name | I/O | Function | @RESET | Shared with |
|---|---|---|---|---|
| EINT13 | I/O | External interrupt and Timer 3 capture input | Input | P10/T3O/PWM3O/EXTSP2 |
| EINT14 | I/O | External interrupt and Timer 4 capture input | Input | P11/T4O/PWM4O |
| EINT15 | I/O | External interrupt and Timer 5 capture input | Input | P12/T5O/PWM5O |
| EINT16 | I/O | External interrupt and Timer 6 capture input | Input | P13/T6O/PWM6O |
| EINT17 | I/O | External interrupt and Timer 7 capture input | Input | P62/AN8/T7O |
| EINT18 | I/O | External interrupt and Timer 8 capture input | Input | P61/CAPL/T8O/PWM8AA |
| T0O | I/O | Timer 0 interval output | Input | P43/SEG19/EINT10/PWM0O |
| T1O | I/O | Timer 1 interval output | Input | P44/SEG20/EINT11/PWM1O |
| T2O | I/O | Timer 2 interval output | Input | P45/SEG21/EINT12/PWM2O |
| T3O | I/O | Timer 3 interval output | Input | P10/EINT13/PWM3O/EXTSP2 |
| T4O | I/O | Timer 4 interval output | Input | P11/EINT14/PWM4O |
| T5O | I/O | Timer 5 interval output | Input | P12/EINT15/PWM5O |
| T6O | I/O | Timer 6 interval output | Input | P13/EINT16/PWM6O |
| T7O | I/O | Timer 7 interval output | Input | P62/AN8/EINT17 |
| T8O | I/O | Timer 8 interval output | Input | P61/CAPL/EINT18/PWM8AA |
| PWM0O | I/O | Timer 0 PWM output | Input | P43/SEG19/EINT10/T0O |
| PWM1O | I/O | Timer 1 PWM output | Input | P44/SEG20/EINT11/T1O |
| PWM2O | I/O | Timer 2 PWM output | Input | P45/SEG21/EINT12/T2O |
| PWM3O | I/O | Timer 3 pulse output | Input | P10/EINT13/T3O/EXTSP2 |
| PWM4O | I/O | Timer 4 pulse output | Input | P11/EINT14/T4O |
| PWM5O | I/O | Timer 5 pulse output | Input | P12/EINT15/T5O |
| PWM6O | I/O | Timer 6 pulse output | Input | P13/EINT16/T6O |
| PWM8AA | I/O | Timer 8's 6-ch PWM outputs | Input | P61/CAPL/EINT18/T8O |
| PWM8AB | | | | P60/CAPH |
| PWM8BA | | | | P70/VLC3 |
| PWM8BB | | | | P71/VLC2 |
| PWM8CA | | | | P72/VLC1 |
| PWM8CB | | | | P73/VLC0 |
| BLNK | I/O | External sync signal input for 6-ch PWMs | Input | P00/AN0/EINT0 |
| EC0 | I/O | Timer 0 event count input | Input | P40/SEG16 |
| EC1 | I/O | Timer 1 event count input | Input | P41/SEG17 |
| EC2 | I/O | Timer 2 event count input | Input | P42/SEG18 |
| EC3 | I/O | Timer 3 event count input | Input | P26/COM6/SEG38/EXTSP0 |
| EC4 | I/O | Timer 4 event count input | Input | P27/COM7/SEG39/EXTSP1 |
| EC5 | I/O | Timer 5 event count input | Input | P14/EINT8/DSCL |
| EC6 | I/O | Timer 6 event count input | Input | P15/EINT9/DSDA |
| EC7 | I/O | Timer 7 event count input | Input | P63/AN9/LVIREF |
| SCK0 | I/O | Serial 0 clock input/output | Input | P30/SEG24 |
| SCK1 | I/O | Serial 1 clock input/output | Input | P55/SEG13 |
| SCK2 | I/O | Serial 2 clock input/output | Input | P35/SEG29 |
| SCK3 | I/O | Serial 3 clock input/output | Input | P21/SEG33 |

**Table 5-1 Normal Pin Description (Continued)**

| PIN Name | I/O | Function | @RESET | Shared with |
|---|---|---|---|---|
| MOSI0 | I/O | Serial 0 data input/output | Input | P31/SEG25/TXD0/SDA0 |
| MOSI1 | I/O | Serial 1 data input/output | Input | P56/SEG14/TXD1/SDA1 |
| MOSI2 | I/O | Serial 2 data input/output | Input | P34/SEG28 |
| MOSI3 | I/O | Serial 3 data input/output | Input | P20/SEG32 |
| MISO0 | I/O | Serial 0 data input/output | Input | P32/SEG26/RXD0/SCL0 |
| MISO1 | I/O | Serial 1 data input/output | Input | P57/SEG15/RXD1/SCL1 |
| MISO2 | I/O | Serial 2 data input/output | Input | P33/SEG27 |
| MISO3 | I/O | Serial 3 data input/output | Input | P37/SEG31 |
| SS0 | I/O | Slave 0 select input | Input | P47/SEG23 |
| SS1 | I/O | Slave 1 select input | Input | P54/SEG12 |
| SS2 | I/O | Slave 2 select input | Input | P36/SEG30 |
| SS3 | I/O | Slave 3 select input | Input | P22/SEG34 |
| TXD0 | I/O | UART 0 data output | Input | P31/SEG25/SDA0/MOSI0 |
| TXD1 | I/O | UART 1 data output | Input | P56/SEG14/SDA1/MOSI1 |
| TXD2 | I/O | UART 2 data output | Input | P64/AN10 |
| TXD3 | I/O | UART 3 data output | Input | P53/SEG11 |
| TXD4 | I/O | UART 4 data output | Input | P51/SEG9 |
| RXD0 | I/O | UART 0 data input | Input | P32/SEG26/SCL0/MISO0 |
| RXD1 | I/O | UART 1 data input | Input | P57/SEG15/SCL1/MISO1 |
| RXD2 | I/O | UART 2 data input | Input | P65/AN11 |
| RXD3 | I/O | UART 3 data input | Input | P52/SEG10 |
| RXD4 | I/O | UART 4 data input | Input | P50/SEG8 |
| SCL0 | I/O | I2C 0 clock input/output | Input | P32/SEG26/RXD0/MISO0 |
| SCL1 | I/O | I2C 1 clock input/output | Input | P57/SEG15/RXD1/MISO1 |
| SDA0 | I/O | I2C 0 data input/output | Input | P31/SEG25/TXD0/MOSI0 |
| SDA1 | I/O | I2C 1 data input/output | Input | P56/SEG14/TXD1/MOSI1 |
| BUZO | I/O | Buzzer signal output | Input | P46/SEG22 |
| AN0 | I/O | A/D converter analog input channels | Input | P00/EINT0/BLNK |
| AN1 | | | | P01/EINT1 |
| AN2 | | | | P02/EINT2 |
| AN3 | | | | P03/EINT3 |
| AN4 | | | | P04/EINT4 |
| AN5 | | | | P05/EINT5 |
| AN6 | | | | P06/EINT6 |
| AN7 | | | | P07/EINT7 |
| AN8 | | | | P62/EINT17/T7O |
| AN9 | | | | P63/EC7/LVIREF |
| AN10 | | | | P64/TXD2 |
| AN11 | | | | P65/RXD2 |
| LVIREF | I/O | Low voltage indicator reference voltage | Input | P63/AN9/EC7 |

**Table 5-1 Normal Pin Description (Continued)**

| PIN Name | I/O | Function | @RESET | Shared with |
|---|---|---|---|---|
| VLC0 | I/O | LCD bias voltage pins | Input | P73/PWM8CB |
| VLC1 | | | | P72/PWM8CA |
| VLC2 | | | | P71/PWM8BB |
| VLC3 | | | | P70/PWM8BA |
| CAPH | I/O | Capacitor terminals for voltage booster | Input | P60/PWM8AB |
| CAPL | | | | P61/EINT18/T8O/PWM8AA |
| COM0–COM3 | I/O | LCD common signal outputs | Input | P77–P74 |
| COM4 | | | | P24/SEG36/PWMOUT |
| COM5 | | | | P25/SEG37/TRIG |
| COM6 | | | | P26/SEG38/EC3/EXTSP0 |
| COM7 | | | | P27/SEG39/EC4/EXTSP1 |
| SEG0–SEG7 | I/O | LCD segment signal outputs | Input | P80–P87 |
| SEG8 | | | | P50/RXD4 |
| SEG9 | | | | P51/TXD4 |
| SEG10 | | | | P52/RXD3 |
| SEG11 | | | | P53/TXD3 |
| SEG12 | | | | P54/SS1 |
| SEG13 | | | | P55/SCK1 |
| SEG14 | | | | P56/TXD1/SDA1/MOSI1 |
| SEG15 | | | | P57/RXD1/SCL1/MISO1 |
| SEG16–SEG18 | | | | P40–P42/EC0–EC2 |
| SEG19 | | | | P43/EINT10/T0O/PWM0O |
| SEG20 | | | | P44/EINT11/T1O/PWM1O |
| SEG21 | | | | P45/EINT12/T2O/PWM2O |
| SEG22 | | | | P46/BUZO |
| SEG23 | | | | P47/SS0 |
| SEG24 | | | | P30/SCK0 |
| SEG25 | | | | P31/TXD0/SDA0/MOSI0 |
| SEG26 | | | | P32/RXD0/SCL0/MISO0 |
| SEG27 | | | | P33/MISO2 |
| SEG28 | | | | P34/MOSI2 |
| SEG29 | | | | P35/SCK2 |
| SEG30 | | | | P36/SS2 |
| SEG31 | | | | P37/MISO3 |
| SEG32 | | | | P20/MOSI3 |
| SEG33 | | | | P21/SCK3 |
| SEG34 | | | | P22/SS3 |
| SEG35 | | | | P23/(T3O/PWM3O/EXTSP2) |
| SEG36 | | | | P24/COM4/PWMOUT |
| SEG37 | | | | P25/COM5/TRIG |
| SEG38 | | | | P26/COM6/EC3/EXTSP0 |
| SEG39 | | | | P27/COM7/EC4/EXTSP1 |

**Table 5-1 Normal Pin Description (Continued)**

| PIN Name | I/O | Function | @RESET | Shared with |
|---|---|---|---|---|
| PWMOUT | I/O | 10-bit PWM output | Input | P24/COM4/SEG36 |
| TRIG | I/O | External trigger input for PWM generator | Input | P25/COM5/SEG37 |
| EXTSP0 | I/O | External shot/emergency stop inputs for PWM generator | Input | P26/COM6/SEG38/EC3 |
| EXTSP1 | | | | P27/COM7/SEG39/EC4 |
| EXTSP2 | | | | P10/EINT13/T3O/PWM3O |
| RESETB | I/O | System reset pin with a pull-up resistor when it is selected as the RESETB by CONFIGURE OPTION | Input | P66 |
| DSDA | I/O | On chip debugger data input/output (NOTE3,4) | Input | P15/EINT9/EC6 |
| DSCL | I/O | On chip debugger clock input (NOTE3,4) | Input | P14/EINT8/EC5 |
| XIN | I/O | Main oscillator pins | Input | P16 |
| XOUT | | | | P17 |
| SXIN | – | Sub oscillator pins | – | – |
| SXOUT | | | | – |
| LPF | – | Loop filter pump output for PLL | – | – |
| VREG | – | Regulator voltage output for sub clock 0.1uF capacitor needed | – | – |
| AVDD, AVSS | – | Analog power input pins | – | – |
| VDD, VSS | – | Digital Power input pins | – | – |

NOTES) 1. The P06-P07, P20-P23, P36-P37, and P8 are not in the 64-Pin package.
2. The P66/RESETB pin is configured as one of the P66 and the RESETB pin by the “CONFIGURE OPTION.”
3. If the P14/EINT8/EC5/DSCL and P15/EINT9/EC6/DSDA pins are connected to an emulator during the reset or power-on reset, the pins are automatically configured as the debugger pins.
4. The P14/EINT8/EC5/DSCL and P15/EINT9/EC6/DSDA pins are configured as inputs with internal pull-up resistors only during the reset or power-on reset.
5. The P17/XOUT and P16/XIN pins are configured as a function pin by software control.

# 6. Port Structures

## 6.1 General Purpose I/O Port

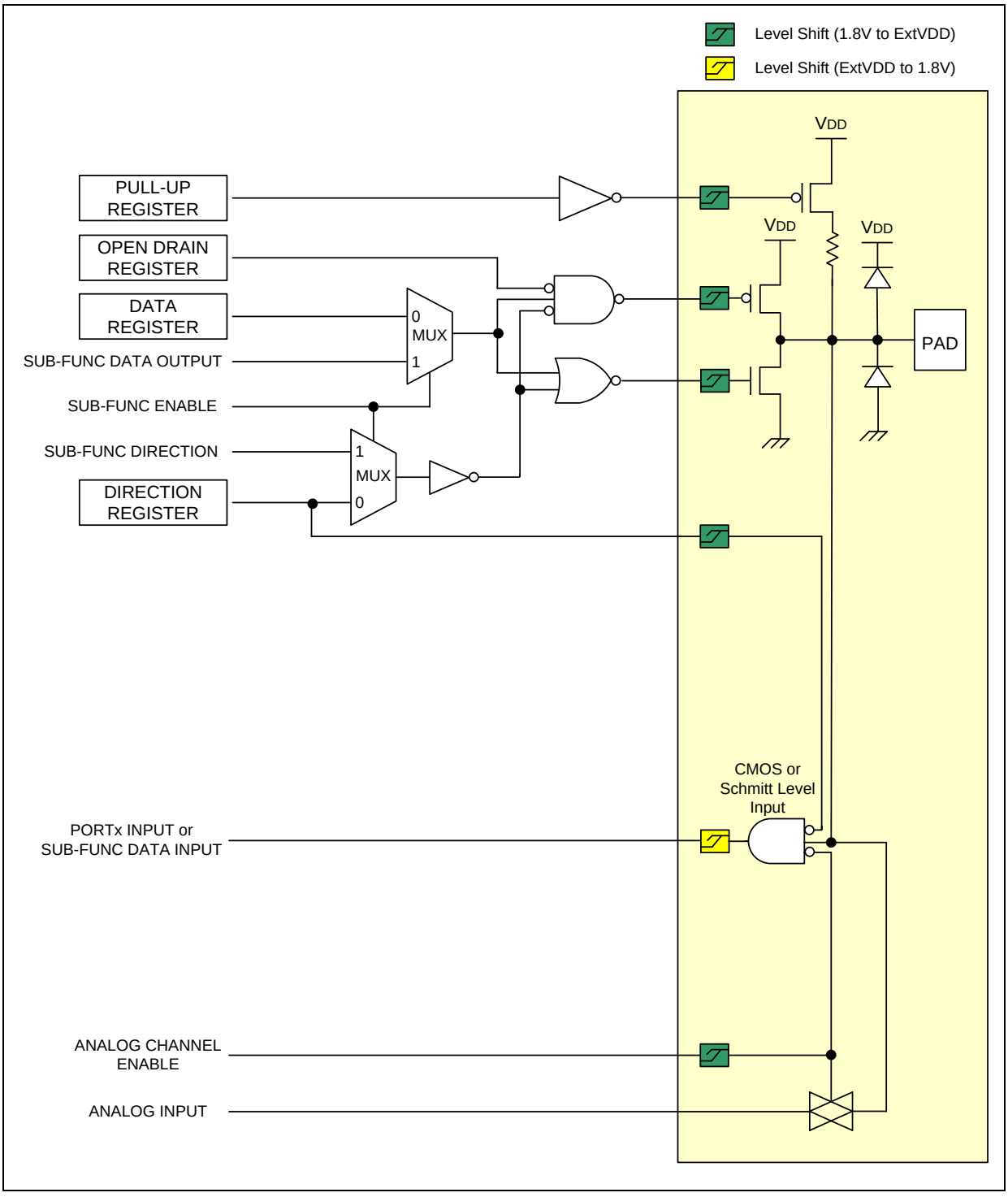



**Figure 6.1 General Purpose I/O Port**

## 6.2 External Interrupt I/O Port

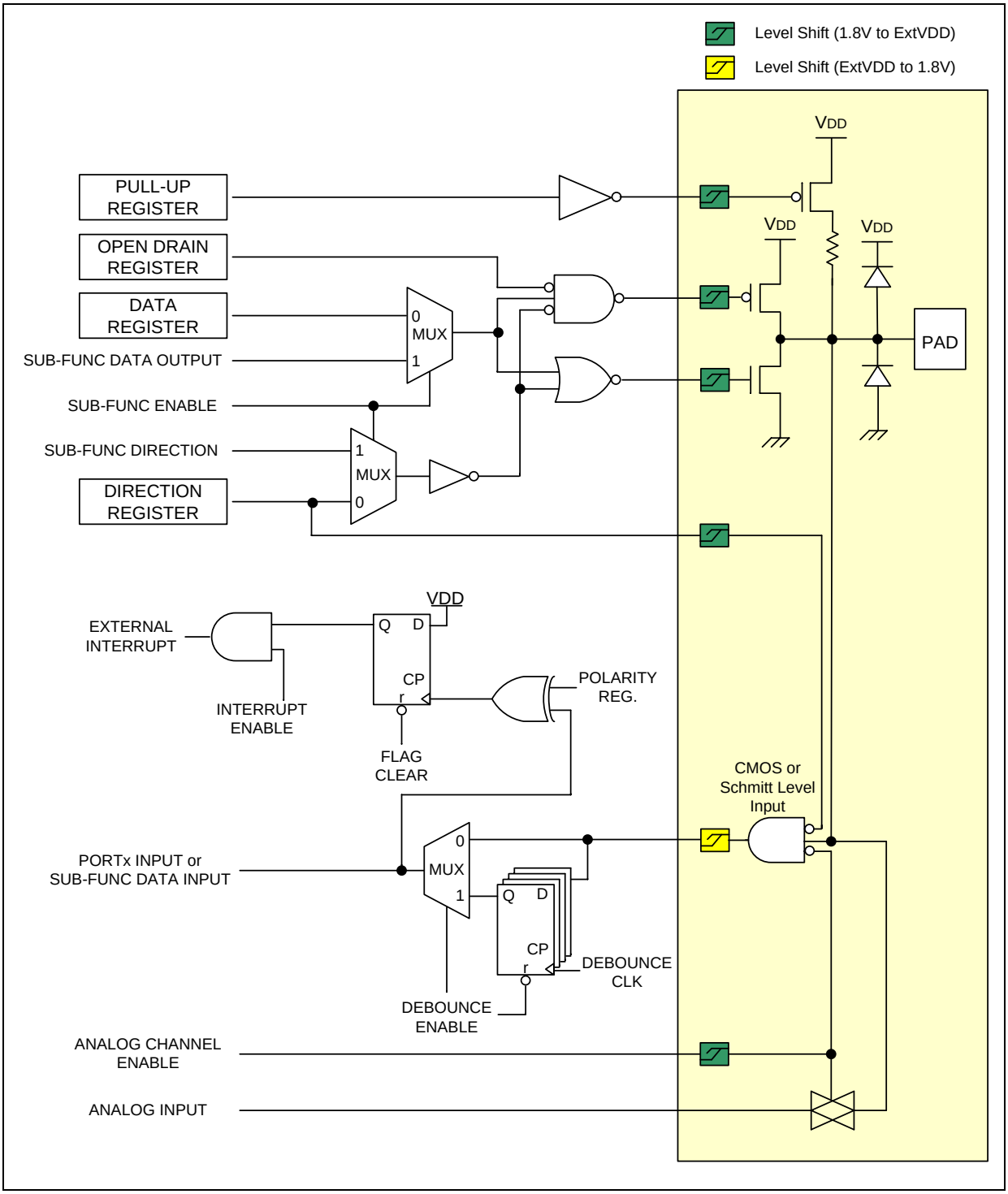



**Figure 6.2 External Interrupt I/O Port**

# 7. Electrical Characteristics

## 7.1 Absolute Maximum Ratings

**Table 7-1 Absolute Maximum Ratings**

| Parameter | Symbol | Rating | Unit | Note |
|---|---|---|---|---|
| Supply Voltage | VDD | -0.3 ~ +6.5 | V | – |
| Normal Voltage Pin | $V_I$ | -0.3 ~ VDD+0.3 | V | Voltage on any pin with respect to VSS |
| | $V_O$ | -0.3 ~ VDD+0.3 | V | |
| | $I_{OH}$ | -10 | mA | Maximum current output sourced by ($I_{OH}$ per I/O pin) |
| | $\sum I_{OH}$ | -80 | mA | Maximum current ($\sum I_{OH}$) |
| | $I_{OL}$ | 60 | mA | Maximum current sunk by ($I_{OL}$ per I/O pin) |
| | $\sum I_{OL}$ | 120 | mA | Maximum current ($\sum I_{OL}$) |
| Total Power Dissipation | $P_T$ | 600 | mW | – |
| Storage Temperature | $T_{STG}$ | -65 ~ +150 | °C | – |

NOTE) Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## 7.2 Recommended Operating Conditions

**Table 7-2 Recommended Operating Conditions**

($T_A$= -40°C ~ +85°C)

| Parameter | Symbol | Conditions | | MIN | TYP | MAX | Unit |
|---|---|---|---|---|---|---|---|
| Operating Voltage | VDD | $f_X$= 32 ~ 38kHz | SX-tal | 1.8 | – | 5.5 | V |
| | | $f_X$= 0.4 ~ 4.2MHz | X-tal | 1.8 | – | 5.5 | |
| | | $f_X$= 0.4 ~ 12.0MHz | | 2.7 | – | 5.5 | |
| | | $f_X$= 0.5 ~ 8.0MHz | Internal RC | 1.8 | – | 5.5 | |
| | | $f_X$= 0.5 ~ 16.0MHz | | 2.0 | – | 5.5 | |
| | | $f_X$= 1.0 ~ 16.0MHz | PLL | 2.0 | – | 5.5 | |
| Operating Temperature | $T_{OPR}$ | VDD= 1.8 ~ 5.5V | | -40 | – | 85 | °C |

## 7.3 A/D Converter Characteristics

**Table 7-3 A/D Converter Characteristics**

($T_A$= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, VDD=AVDD, VSS=AVSS=0V)

| Parameter | Symbol | Conditions | | MIN | TYP | MAX | Unit |
|---|---|---|---|---|---|---|---|
| Resolution | – | – | | – | 12 | – | bit |
| Integral Linear Error | ILE | AVDD= 2.7V – 5.5V<br>fx= 8MHz | | – | – | ±6 | LSB |
| Differential Linearity Error | DLE | | | – | – | ±1 | |
| Zero Offset Error | ZOE | | | – | – | ±5 | |
| Full Scale Error | FSE | | | – | – | ±5 | |
| Conversion Time | $t_{CON}$ | 12bit resolution, 8MHz | | 20 | – | – | µS |
| Analog Input Voltage | $V_{AN}$ | – | | AVSS | – | AVDD | V |
| Internal VDC Voltage | VDD18 | – | | – | 1.8 | – | V |
| Analog Input Leakage Current | $I_{AN}$ | AVDD= 5.12V | | – | – | 2 | µA |
| ADC Current | $I_{ADC}$ | Enable | AVDD= 5.12V | – | 1 | 2 | mA |
| | | Disable | | – | – | 0.1 | µA |

NOTES) 1. Zero offset error is the difference between 000000000000 and the converted output for zero input voltage (AVSS).

2. Full scale error is the difference between 111111111111 and the converted output for full-scale input voltage (AVDD).

## 7.4 Power-On Reset Characteristics

**Table 7-4 Power-on Reset Characteristics**

($T_A$= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, VSS= 0V)

| Parameter | Symbol | Conditions | MIN | TYP | MAX | Unit |
|---|---|---|---|---|---|---|
| RESET Release Level | $V_{POR}$ | – | – | 1.4 | – | V |
| VDD Voltage Rising Time | $t_R$ | – | 0.05 | – | 30 | V/mS |
| POR Current | $I_{POR}$ | – | – | 0.2 | – | µA |

## 7.5 Low Voltage Reset and Low Voltage Indicator Characteristics

**Table 7-5 LVR and LVI Characteristics**

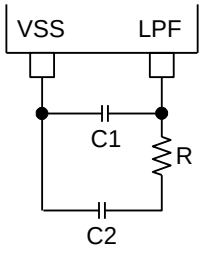
($T_A$= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, VSS= 0V)

| Parameter | Symbol | Conditions | | MIN | TYP | MAX | Unit |
|---|---|---|---|---|---|---|---|
| Detection Level | $V_{LVR}$<br>$V_{LVI}$ | The LVR can select all levels but LVI can select other levels except 1.60V. | | – | 1.60 | 1.79 | V |
| | | | | 1.85 | 2.00 | 2.15 | |
| | | | | 1.95 | 2.10 | 2.25 | |
| | | | | 2.05 | 2.20 | 2.35 | |
| | | | | 2.17 | 2.32 | 2.47 | |
| | | | | 2.29 | 2.44 | 2.59 | |
| | | | | 2.39 | 2.59 | 2.79 | |
| | | | | 2.55 | 2.75 | 2.95 | |
| | | | | 2.73 | 2.93 | 3.13 | |
| | | | | 2.94 | 3.14 | 3.34 | |
| | | | | 3.18 | 3.38 | 3.58 | |
| | | | | 3.37 | 3.67 | 3.97 | |
| | | | | 3.70 | 4.00 | 4.30 | |
| | | | | 4.10 | 4.40 | 4.70 | |
| Hysteresis | △V | – | | – | 50 | 150 | mV |
| Minimum Pulse Width | $t_{LW}$ | – | | 100 | – | – | µS |
| LVR and LVI Current | $I_{BL}$ | Enable (Both) | VDD= 3V RUN Mode | – | 14.0 | 24.0 | µA |
| | | Enable (One of two) | | – | 10.0 | 18.0 | |
| | | Disable (Both) | VDD= 3V | – | – | 0.1 | |

## 7.6 Phase Locked Loop Characteristics

**Table 7-6 Phase Locked Loop Characteristics**

($T_A$= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, VSS= 0V)

| Parameter | Symbol | Condition | MIN | TYP | MAX | Unit |
|---|---|---|---|---|---|---|
| Input Frequency Range | – |  | – | 32.768 | – | kHz |
| Output Frequency Range | $f_{VCO}$ | | 1.024 | – | 16.384 | MHz |
| Clock Duty Ratio | $T_{OD}$ | | 45 | 50 | 55 | % |
| Tolerance | – | | – | – | ±4 | % |
| Settling Time | $t_D$ | | – | 10 | 100 | mS |
| PLL Current | $I_{PLL}$ | Enable, $f_{VCO}$=16.384MHz | – | 0.5 | 1.0 | mA |
| | | Disable | – | – | 0.1 | uA |

NOTE) Where R= 6.8kΩ, C1=820pF, and C2= 10nF.

## 7.7 Internal RC Oscillator Characteristics

**Table 7-7 Internal RC Oscillator Characteristics**

($T_A$= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, VSS= 0V)

| Parameter | Symbol | Conditions | | MIN | TYP | MAX | Unit |
|---|---|---|---|---|---|---|---|
| Frequency | $f_{IRC}$ | $V_{DD}$ = 2.0V to 5.5 V | | – | 16 | – | MHz |
| Tolerance | – | $T_A$ = 0°C to +50°C | With 0.1uF Bypass Capacitor | – | – | ±1.5 | % |
| | | $T_A$ = -20°C to +85°C | | | | ±2.5 | |
| | | $T_A$ = -40°C to +85°C | | | | ±3.5 | |
| Clock Duty Ratio | TOD | – | | 40 | 50 | 60 | % |
| Stabilization Time | $T_{HFS}$ | – | | – | – | 100 | μS |
| IRC Current | $I_{IRC}$ | Enable | | – | 0.2 | – | mA |
| | | Disable | | – | – | 0.1 | μA |

Note: A 0.1uF Bypass capacitor should be connected to VDD and VSS. Refer to the “Recommend Circuit and Layout”.

## 7.8 Internal Watch-Dog Timer RC Oscillator Characteristics

**Table 7-8 Internal WDTRC Oscillator Characteristics**

($T_A$= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, VSS= 0V)

| Parameter | Symbol | Conditions | MIN | TYP | MAX | Unit |
|---|---|---|---|---|---|---|
| Frequency | $f_{WDTRC}$ | – | 2 | 5 | 10 | kHz |
| Stabilization Time | $t_{WDTS}$ | – | – | – | 1 | mS |
| WDTRC Current | $I_{WDTRC}$ | Enable | – | 1 | – | μA |
| | | Disable | – | – | 0.1 | |

## 7.9 LCD Voltage Characteristics

**Table 7-9 LCD Voltage Characteristics**

($T_A$= -40°C ~ +85°C, VDD= 2.0V ~ 5.5V, VSS= 0V)

| Parameter | Symbol | Conditions | | MIN | TYP | MAX | Unit |
|---|---|---|---|---|---|---|---|
| LCD Voltage | $V_{LC3}$ | Voltage booster enabled, 1/4 bias | LCDCCR=0000b | Typx0.93 | 0.75 | Typx1.07 | V |
| | | | LCDCCR=0001b | | 0.79 | | |
| | | | LCDCCR=0010b | | 0.83 | | |
| | | | LCDCCR=0011b | | 0.86 | | |
| | | | LCDCCR=0100b | | 0.90 | | |
| | | | LCDCCR=0101b | | 0.94 | | |
| | | | LCDCCR=0110b | | 0.98 | | |
| | | | LCDCCR=0111b | | 1.01 | | |
| | | | LCDCCR=1000b | | 1.05 | | |
| | | | LCDCCR=1001b | | 1.09 | | |
| | | | LCDCCR=1010b | | 1.13 | | |
| | | | LCDCCR=1011b | | 1.16 | | |
| | | | LCDCCR=1100b | | 1.20 | | |
| | | | LCDCCR=1101b | | 1.24 | | |
| | | | LCDCCR=1110b | | 1.28 | | |
| | | | LCDCCR=1111b | | 1.31 | | |
| LCD Mid Bias Voltage | $V_{LC0}$ | Voltage booster enabled, 1/4 bias, No panel load, VDD=3V | | Typx0.95 | 4xVLC3 | Typx1.05 | V |
| | $V_{LC1}$ | | | Typx0.95 | 3xVLC3 | Typx1.05 | |
| | $V_{LC2}$ | | | Typx0.95 | 2xVLC3 | Typx1.05 | |
| | $V_{LC1}$ | Voltage booster disabled, VDD=2.7V to 5.5V, 1/4 bias, LCD clock = 0Hz, VLC0=VDD | | Typ-0.2 | 0.75xVDD | Typ+0.2 | V |
| | $V_{LC2}$ | | | Typ-0.2 | 0.5xVDD | Typ+0.2 | |
| | $V_{LC3}$ | | | Typ-0.2 | 0.25xVDD | Typ+0.2 | |
| LCD Driver Output Impedance | $R_{LO}$ | VLCD=3V, ILOAD=±10uA | | – | 5 | 10 | kΩ |
| LCD Bias Dividing Resistor | $R_{LCD1}$ | Internal resistor mode, $T_A$ = 25°C | Low | 7.5 | 10 | 12.5 | kΩ |
| | $R_{LCD2}$ | | Mid | 35 | 50 | 65 | |
| | $R_{LCD3}$ | | High | 75 | 100 | 125 | |
| LCD Block Current | $I_{LCD}$ | Voltage booster mode, VDD=3V, VLCD=3.15V, 1/3Bias | | – | 3 | 6 | uA |

## 7.10 DC Characteristics

**Table 7-10 DC Characteristics**

($T_A$= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, VSS= 0V, $f_{XIN}$= 12MHz)

| Parameter | Symbol | Conditions | | MIN | TYP | MAX | Unit |
|---|---|---|---|---|---|---|---|
| Input High Voltage | $V_{IH1}$ | P0~P6, RESETB | | 0.8VDD | – | VDD | V |
| | $V_{IH2}$ | All input pins except $V_{IH1}$ | | 0.7VDD | – | VDD | V |
| | $V_{IH3}$ | P20, P33, P34, P37;<br>3V Interface mode | | 2.4 | – | VDD | V |
| Input Low Voltage | $V_{IL1}$ | P0~P6, RESETB | | – | – | 0.2VDD | V |
| | $V_{IL2}$ | All input pins except $V_{IL1}$ | | – | – | 0.3VDD | V |
| Output High Voltage | $V_{OH}$ | VDD= 4.5V, $I_{OH}$= -2mA,<br>All output ports; | | VDD-1.0 | – | – | V |
| Output Low Voltage | $V_{OL1}$ | VDD=4.5V, $I_{OL}$= 10mA;<br>All output ports except $V_{OL2}$ | | – | – | 1.0 | V |
| | $V_{OL2}$ | VDD= 4.5V, $I_{OL}$= 15mA ;<br>P10~P13, P23~P24, P4 | | – | – | 1.0 | V |
| Input High Leakage Current | $I_{IH}$ | All input ports | | – | – | 1 | µA |
| Input Low Leakage Current | $I_{IL}$ | All input ports | | -1 | – | – | µA |
| Pull-Up Resistor | $R_{PU1}$ | VI=0V,<br>$T_A$= 25°C<br>All Input ports | VDD=5.0V | 25 | 50 | 100 | kΩ |
| | | | VDD=3.0V | 50 | 100 | 200 | |
| | $R_{PU2}$ | VI=0V,<br>$T_A$= 25°C<br>RESETB | VDD=5.0V | 150 | 250 | 400 | kΩ |
| | | | VDD=3.0V | 300 | 500 | 700 | |
| OSC feedback resistor | $R_{X1}$ | XIN= VDD, XOUT= VSS<br>$T_A$= 25°C, VDD= 5V | | 600 | 1200 | 2000 | kΩ |
| | $R_{X2}$ | SXIN=VDD, SXOUT=VSS<br>$T_A$= 25 °C ,VDD=5V | | 2500 | 5000 | 10000 | |

**Table 7-10 DC Characteristics (Continued)**

($T_A$= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, VSS= 0V, $f_{XIN}$= 12MHz)

| Parameter | Symbol | Condition | | MIN | TYP | MAX | Unit |
|---|---|---|---|---|---|---|---|
| Supply Current | $I_{DD1}$ (RUN) | $f_{XIN}$= 12MHz, VDD= 5V±10% | | – | 4.0 | 8.0 | mA |
| | | $f_{XIN}$= 10MHz, VDD= 3V±10% | | – | 3.0 | 6.0 | |
| | | $f_{IRC}$= 8MHz, VDD= 5V±10% | | – | 2.5 | 5.0 | |
| | $I_{DD2}$ (IDLE) | $f_{XIN}$= 12MHz, VDD= 5V±10% | | – | 1.7 | 3.4 | mA |
| | | $f_{XIN}$= 10MHz, VDD= 3V±10% | | – | 1.0 | 2.0 | |
| | | $f_{IRC}$= 8MHz, VDD= 5V±10% | | – | 1.0 | 2.0 | |
| | $I_{DD3}$ | $f_{XIN}$= 32.768kHz<br>VDD= 3V±10%<br>$T_A$= 25°C<br>PSAVE=1 | Sub RUN | – | 60.0 | 90.0 | µA |
| | $I_{DD4}$ | | Sub IDLE | – | 3.5 | 7.0 | µA |
| | $I_{DD5}$ | STOP, VDD= 5V±10%, $T_A$= 25°C | | – | 0.5 | 3.0 | µA |

NOTES) 1. Where the $f_{XIN}$ is an external main oscillator, the $f_{SUB}$ is an external sub oscillator, the $f_{IRC}$ is an internal RC oscillator, the fPLL is the output frequency of the PLL (phase locked-loop), and the fx is the selected system clock.

2. All supply current items don't include the current of an internal Watch-dog timer RC (WDTRC) oscillator and a peripheral block.

3. All supply current items include the current of the power-on reset (POR) block.

## 7.11 AC Characteristics

**Table 7-11 AC Characteristics**

($T_A$= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V)

| Parameter | Symbol | Conditions | MIN | TYP | MAX | Unit |
|---|---|---|---|---|---|---|
| RESETB input low width | $t_{RST}$ | Input, VDD= 5V | 10 | – | – | μS |
| Interrupt input high, low width | $t_{IWH}$, $t_{IWL}$ | All interrupt, VDD= 5V | 200 | – | – | nS |
| External Counter Input High, Low Pulse Width | $t_{ECWH}$, $t_{ECWL}$ | ECn, VDD = 5 V (n= 0, 1, 2, 3, 4, 5, 6, 7) | 200 | – | – | |
| External Counter Transition Time | $t_{REC}$, $t_{FEC}$ | ECn, VDD = 5 V (n= 0, 1, 2, 3, 4, 5, 6, 7) | 20 | – | – | |

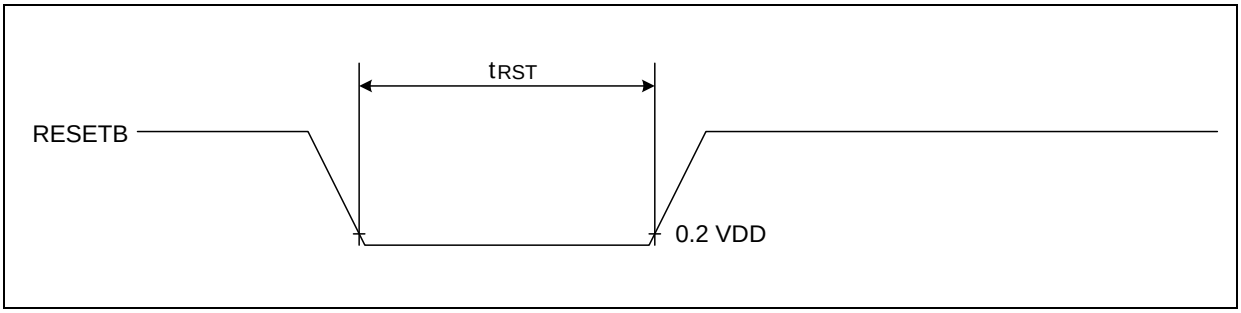



**Figure 7.1 Input Timing for RESETB**

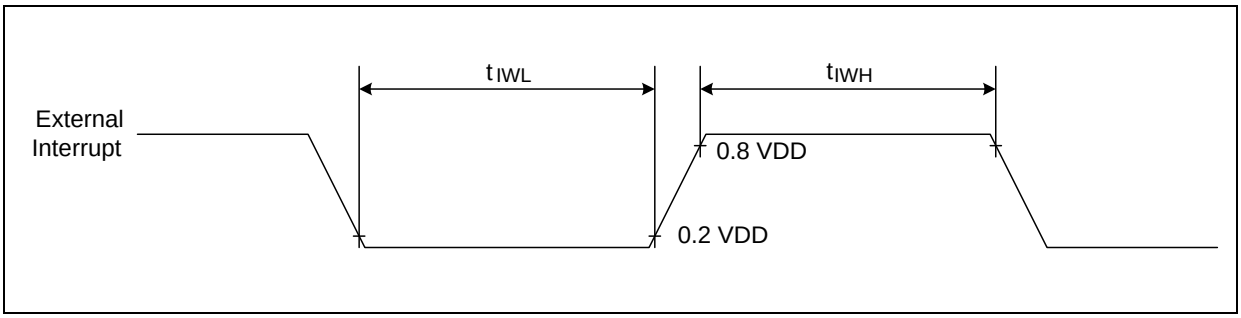



**Figure 7.2 Input Timing for External Interrupts**

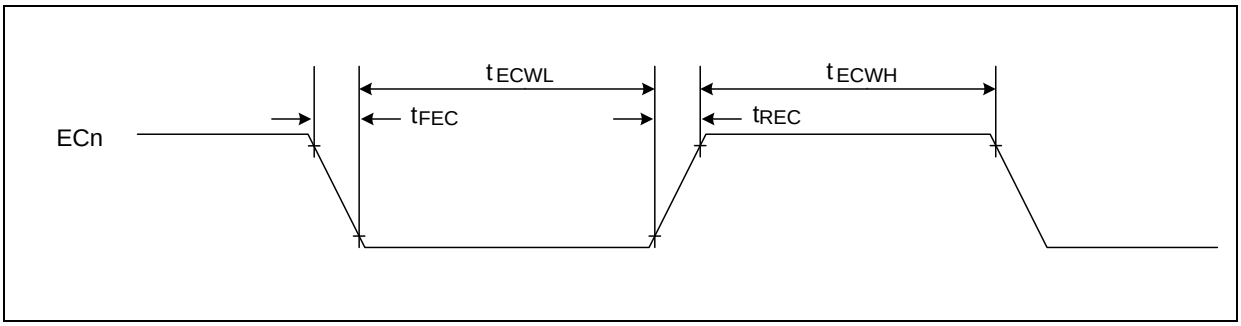



**Figure 7.3 Input Timing for EC0, EC1, EC2, EC3, EC4, EC5, EC6, EC7**

## 7.12 SPI Characteristics

**Table 7-12 SPI Characteristics**

($T_A$= -40°C – +85°C, VDD= 1.8V – 5.5V)

| Parameter | Symbol | Conditions | MIN | TYP | MAX | Unit |
|---|---|---|---|---|---|---|
| Output Clock Pulse Period | $t_{SCK}$ | Internal SCK source | 200 | – | – | nS |
| Input Clock Pulse Period | | External SCK source | 200 | – | – | |
| Output Clock High, Low Pulse Width | $t_{SCKH}$, $t_{SCKL}$ | Internal SCK source | 70 | – | – | |
| Input Clock High, Low Pulse Width | | External SCK source | 70 | – | – | |
| First Output Clock Delay Time | $t_{FOD}$ | Internal/External SCK source | 100 | – | – | |
| Output Clock Delay Time | $t_{DS}$ | – | – | – | 50 | |
| Input Setup Time | $t_{DIS}$ | – | 100 | – | – | |
| Input Hold Time | $t_{DIH}$ | – | 150 | – | – | |

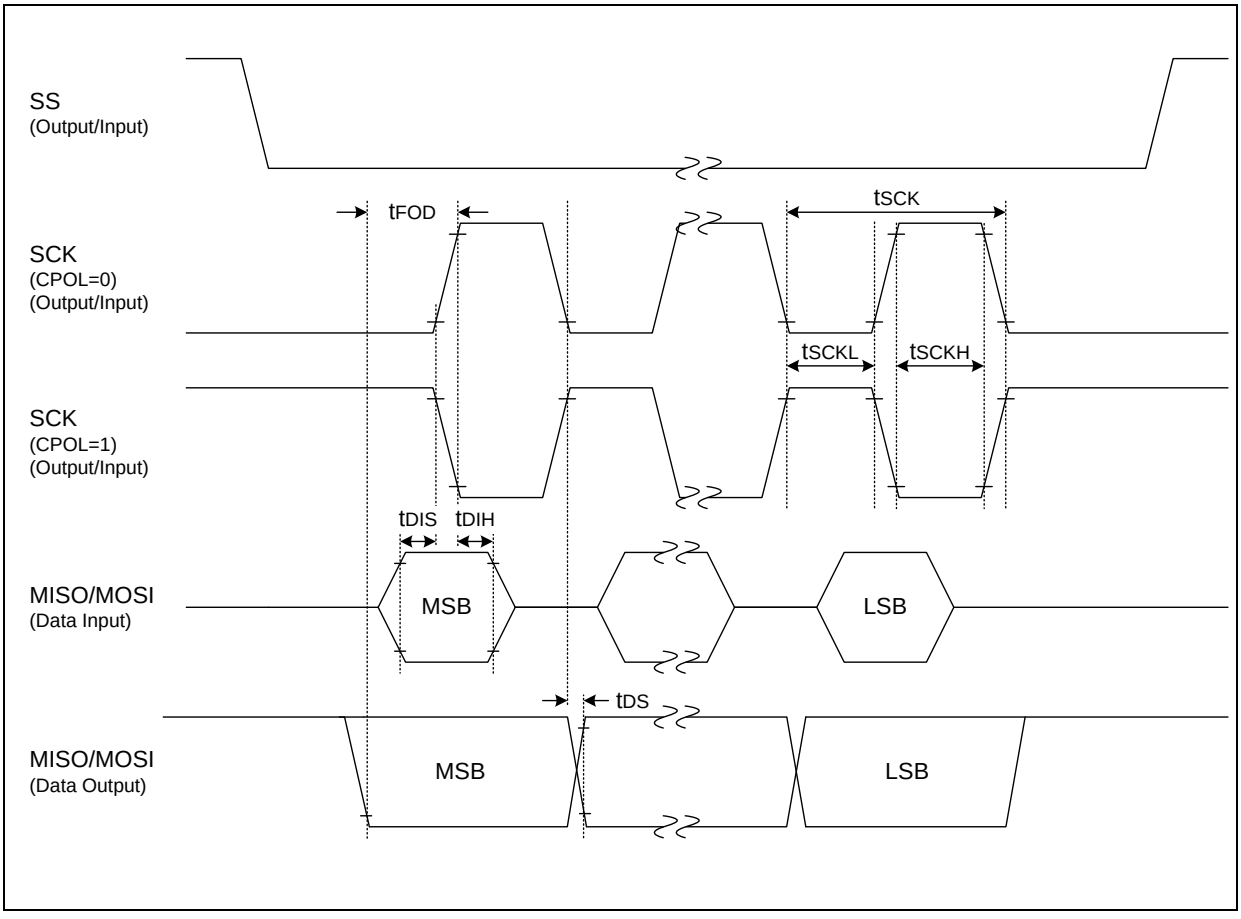



**Figure 7.4 SPI Timing**

## 7.13 UART Characteristics

**Table 7-13 UART Characteristics**

($T_A$= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, $f_{XIN}$=11.1MHz)

| Parameter | Symbol | MIN | TYP | MAX | Unit |
|---|---|---|---|---|---|
| Serial port clock cycle time | $t_{SCK}$ | 1250 | $t_{CPU}$ x 16 | 1650 | nS |
| Output data setup to clock rising edge | $t_{S1}$ | 590 | $t_{CPU}$ x 13 | – | nS |
| Clock rising edge to input data valid | $t_{S2}$ | – | – | 590 | nS |
| Output data hold after clock rising edge | $t_{H1}$ | $t_{CPU}$ - 50 | $t_{CPU}$ | – | nS |
| Input data hold after clock rising edge | $t_{H2}$ | 0 | – | – | nS |
| Serial port clock High, Low level width | $t_{HIGH}$, $t_{LOW}$ | 470 | $t_{CPU}$ x 8 | 970 | nS |

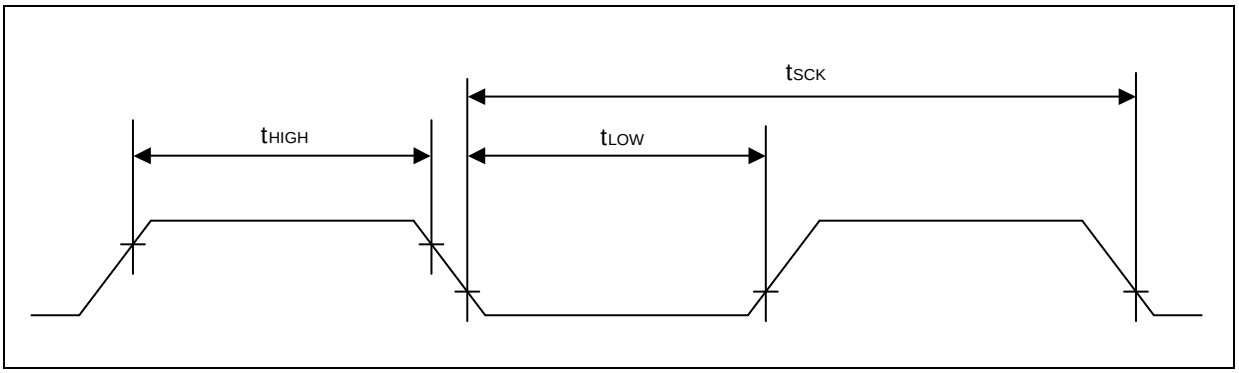



**Figure 7.5 Waveform for UART Timing Characteristics**

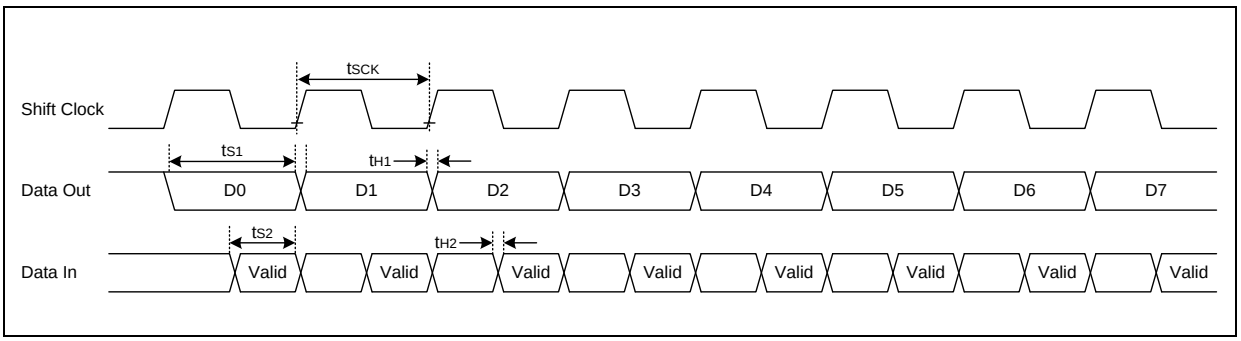



**Figure 7.6 Timing Waveform for the UART Module**

## 7.14 I2C Characteristics

**Table 7-14 I2C Characteristics**

($T_A$= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V)

| Parameter | Symbol | Standard Mode | | High-Speed Mode | | Unit |
|---|---|---|---|---|---|---|
| | | MIN | MAX | MIN | MAX | |
| Clock frequency | $t_{SCL}$ | 0 | 100 | 0 | 400 | kHz |
| Clock High Pulse Width | $t_{SCLH}$ | 4.0 | – | 0.6 | – | μS |
| Clock Low Pulse Width | $t_{SCLL}$ | 4.7 | – | 1.3 | – | |
| Bus Free Time | $t_{BF}$ | 4.7 | – | 1.3 | – | |
| Start Condition Setup Time | $t_{STSU}$ | 4.7 | – | 0.6 | – | |
| Start Condition Hold Time | $t_{STHD}$ | 4.0 | – | 0.6 | – | |
| Stop Condition Setup Time | $t_{SPSU}$ | 4.0 | – | 0.6 | – | |
| Stop Condition Hold Time | $t_{SPHD}$ | 4.0 | – | 0.6 | – | |
| Output Valid from Clock | $t_{VD}$ | 0 | – | 0 | – | |
| Data Input Hold Time | $t_{DIH}$ | 0 | – | 0 | 1.0 | |
| Data Input Setup Time | $t_{DIS}$ | 250 | – | 100 | – | nS |

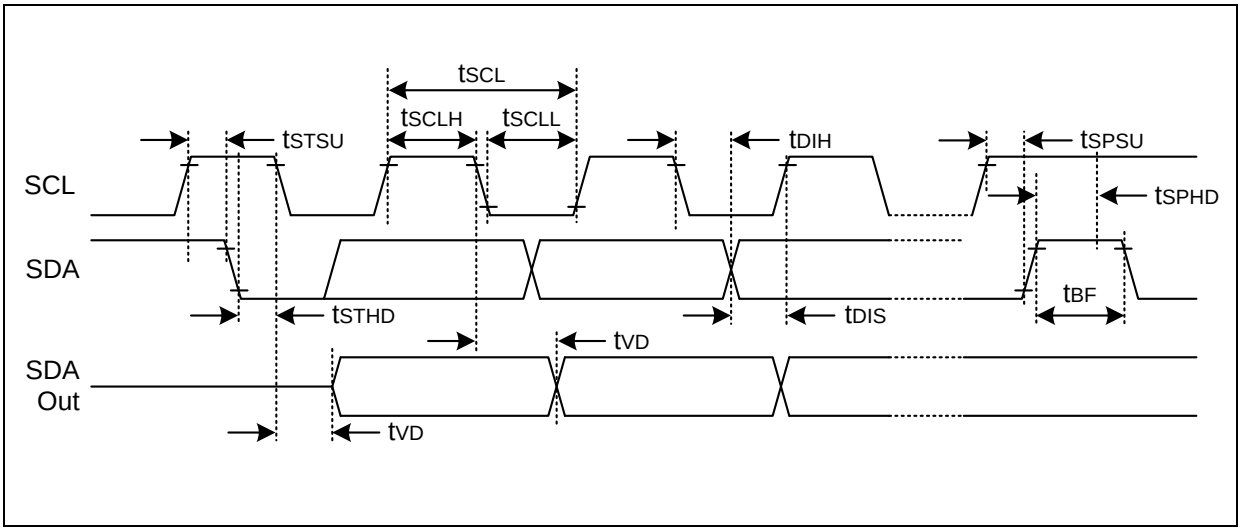


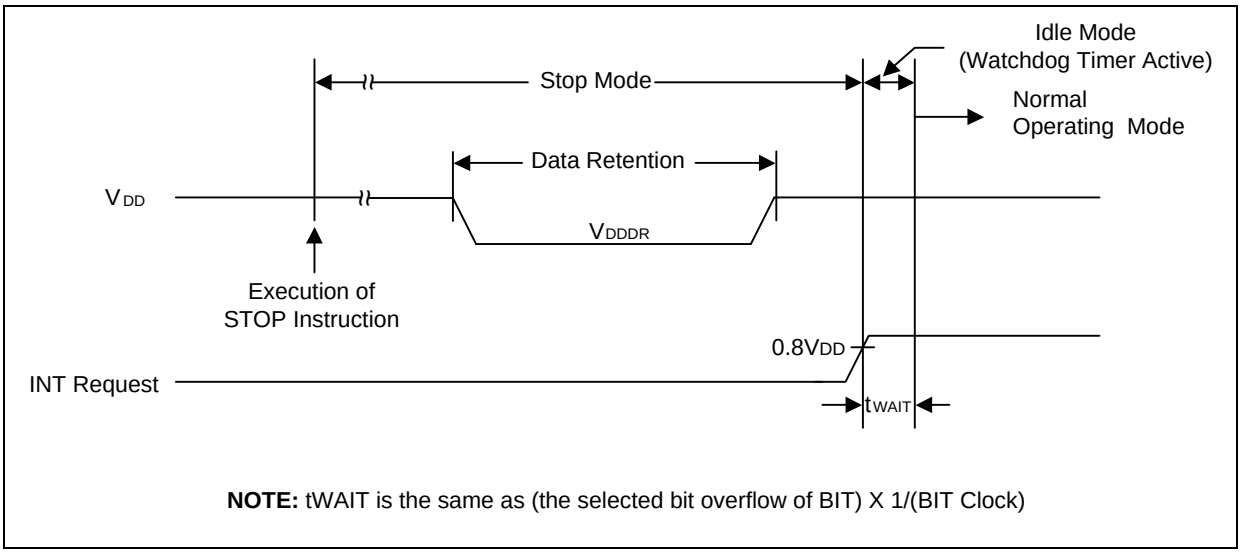

**Figure 7.7 I2C Timing**

## 7.15 Data Retention Voltage in Stop Mode

**Table 7-15 Data Retention Voltage in Stop Mode**

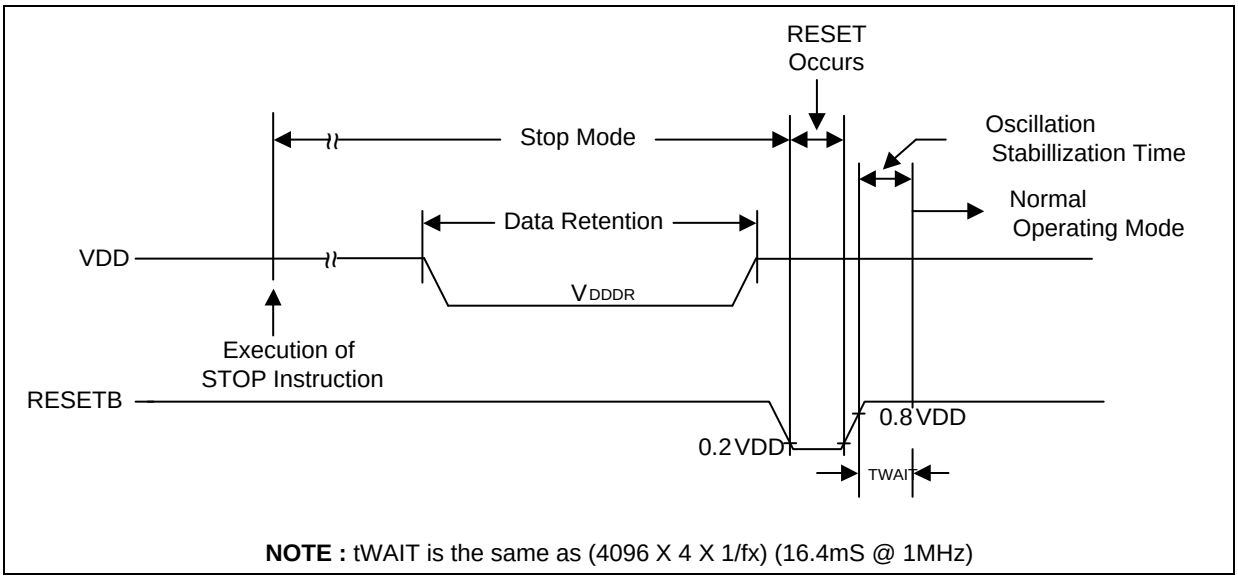
($T_A$= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V)

| Parameter | Symbol | Conditions | MIN | TYP | MAX | Unit |
|---|---|---|---|---|---|---|
| Data retention supply voltage | $V_{DDDR}$ | – | 1.8 | – | 5.5 | V |
| Data retention supply current | $I_{DDDR}$ | VDDR= 1.8V, ($T_A$= 25°C), Stop mode | – | – | 1 | μA |




**NOTE:** tWAIT is the same as (the selected bit overflow of BIT) X 1/(BIT Clock)

**Figure 7.8 Stop Mode Release Timing when Initiated by an Interrupt**




**NOTE :** tWAIT is the same as (4096 X 4 X 1/fx) (16.4mS @ 1MHz)

**Figure 7.9 Stop Mode Release Timing when Initiated by RESETB**

## 7.16 Internal Flash Rom Characteristics

**Table 7-16 Internal Flash Rom Characteristics**

($T_A$= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, VSS= 0V)

| Parameter | Symbol | Condition | MIN | TYP | MAX | Unit |
|---|---|---|---|---|---|---|
| Sector Write Time | $t_{FSW}$ | – | – | 2.5 | 2.7 | mS |
| Sector Erase Time | $t_{FSE}$ | – | – | 2.5 | 2.7 | |
| Hard-Lock Time | $t_{FHL}$ | – | – | 2.5 | 2.7 | |
| Page Buffer Reset Time | $t_{FBR}$ | – | – | – | 5 | μS |
| Flash Programming Frequency | $f_{PGM}$ | – | 0.4 | – | – | MHz |
| Endurance of Write/Erase (Sector 0~1019) | $NF_{WE}$ | – | – | – | 10,000 | Times |
| Endurance of Write/Erase (Sector 1020~1023) | | | | | 100,000 | Times |

NOTE) During a flash operation, SCLK[1:0] of SCCR must be set to “00” or “01” (INT-RC OSC or Main X-TAL for system clock).

## 7.17 Input/Output Capacitance

**Table 7-17 Input/Output Capacitance**

($T_A$= -40°C ~ +85°C, VDD= 0V)

| Parameter | Symbol | Condition | MIN | TYP | MAX | Unit |
|---|---|---|---|---|---|---|
| Input Capacitance | $C_{IN}$ | fx= 1MHz<br>Unmeasured pins are connected to VSS | – | – | 10 | pF |
| Output Capacitance | $C_{OUT}$ | | | | | |
| I/O Capacitance | $C_{IO}$ | | | | | |

## 7.18 Main Clock Oscillator Characteristics

**Table 7-18 Main Clock Oscillator Characteristics**

($T_A$= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V)

| Oscillator | Parameter | Condition | MIN | TYP | MAX | Unit |
|---|---|---|---|---|---|---|
| Crystal | Main oscillation frequency | 1.8V – 5.5V | 0.4 | – | 4.2 | MHz |
| | | 2.7V – 5.5V | 0.4 | – | 12.0 | |
| Ceramic Oscillator | Main oscillation frequency | 1.8V – 5.5V | 0.4 | – | 4.2 | MHz |
| | | 2.7V – 5.5V | 0.4 | – | 12.0 | |
| External Clock | XIN input frequency | 1.8V – 5.5V | 0.4 | – | 4.2 | MHz |
| | | 2.7V – 5.5V | 0.4 | – | 12.0 | |

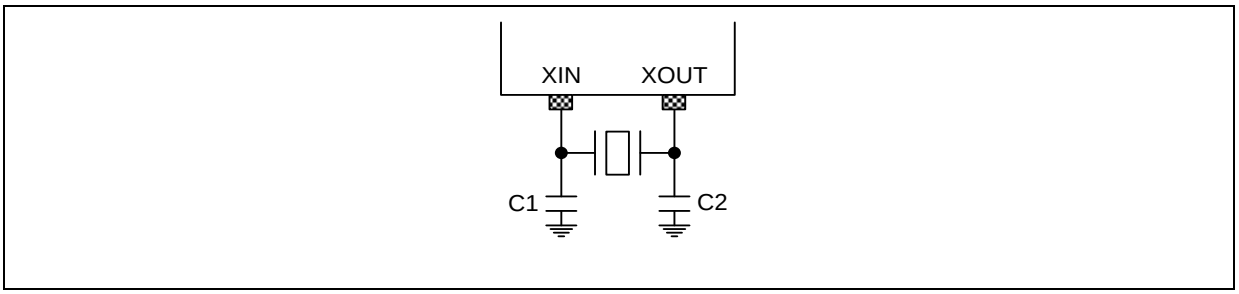



**Figure 7.10 Crystal/Ceramic Oscillator**

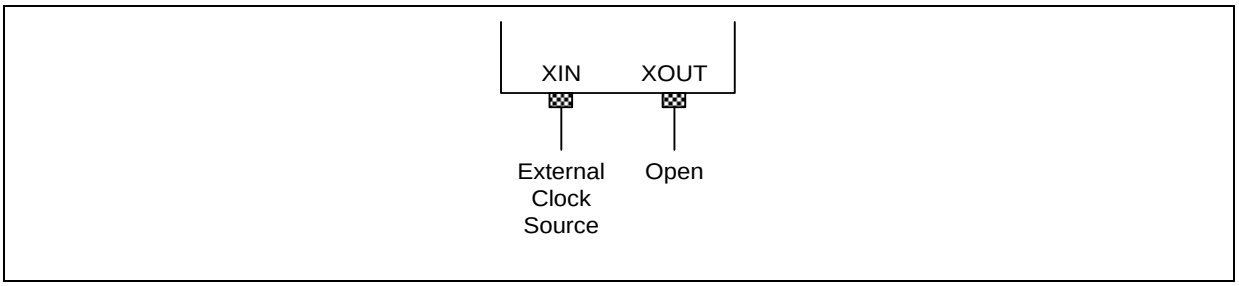



**Figure 7.11 External Clock**

## 7.19 Sub Clock Oscillator Characteristics

**Table 7-19 Sub Clock Oscillator Characteristics**

($T_A$= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V)

| Oscillator | Parameter | Condition | MIN | TYP | MAX | Unit |
|---|---|---|---|---|---|---|
| Crystal | Sub oscillation frequency | 1.8V – 5.5V | 32 | 32.768 | 38 | kHz |
| External Clock | SXIN input frequency | | 32 | – | 100 | kHz |

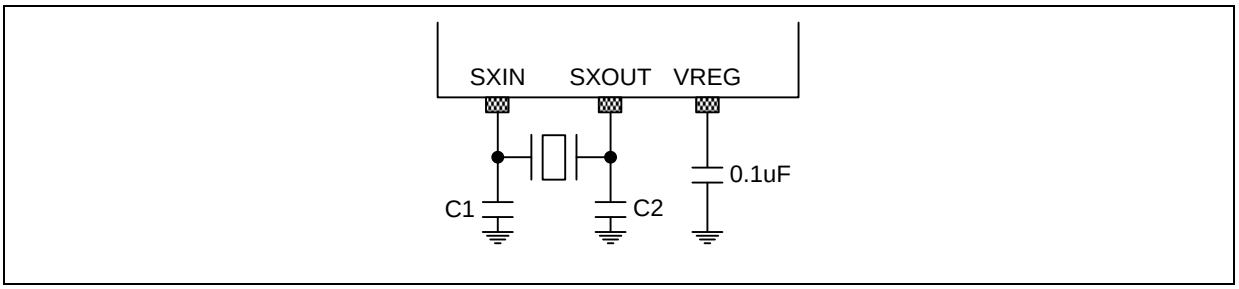



**Figure 7.12 Crystal Oscillator**

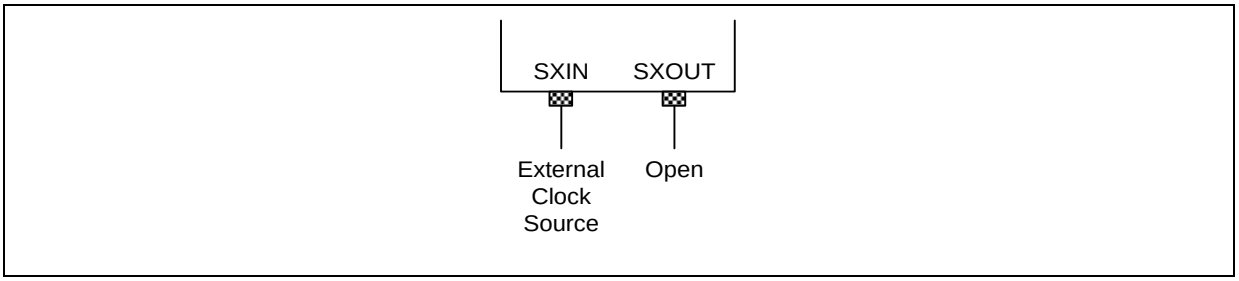



**Figure 7.13 External Clock**

## 7.20 Main Oscillation Stabilization Characteristics

**Table 7-20 Main Oscillation Stabilization Characteristics**

($T_A$= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V)

| Oscillator | Parameter | MIN | TYP | MAX | Unit |
|---|---|---|---|---|---|
| Crystal | fx > 1MHz<br>Oscillation stabilization occurs when VDD is equal to the minimum oscillator voltage range. | – | – | 60 | mS |
| Ceramic | | – | – | 10 | mS |
| External Clock | $f_{XIN}$ = 0.4 to 12MHz<br>XIN input high and low width ($t_{XH}$, $t_{XL}$) | 31 | – | 1250 | nS |

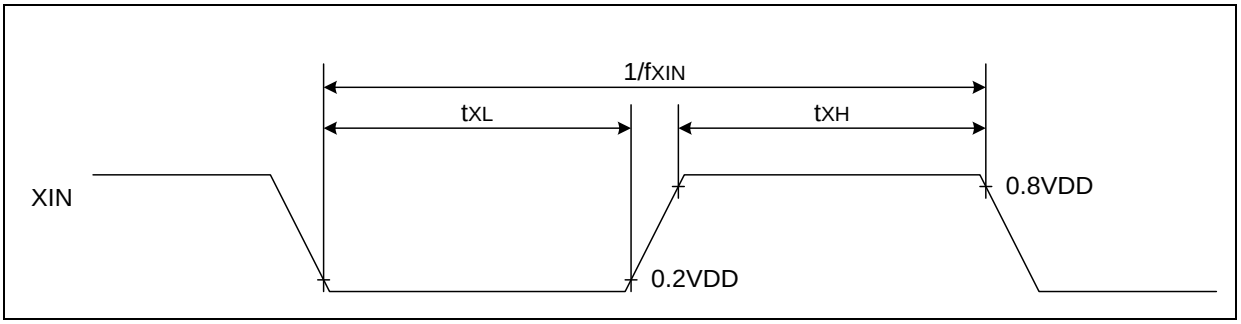



**Figure 7.14 Clock Timing Measurement at XIN**

## 7.21 Sub Oscillation Characteristics

**Table 7-21 Sub Oscillation Stabilization Characteristics**

($T_A$= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V)

| Oscillator | Parameter | MIN | TYP | MAX | Unit |
|---|---|---|---|---|---|
| Crystal | – | – | – | 10 | S |
| External Clock | SXIN input high and low width ($t_{XH}$, $t_{XL}$) | 5 | – | 15 | μS |

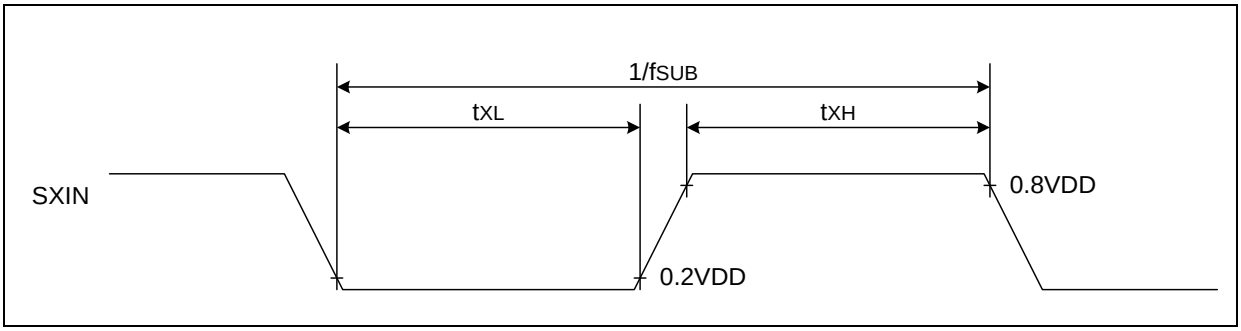



**Figure 7.15 Clock Timing Measurement at SXIN**

## 7.22 Operating Voltage Range

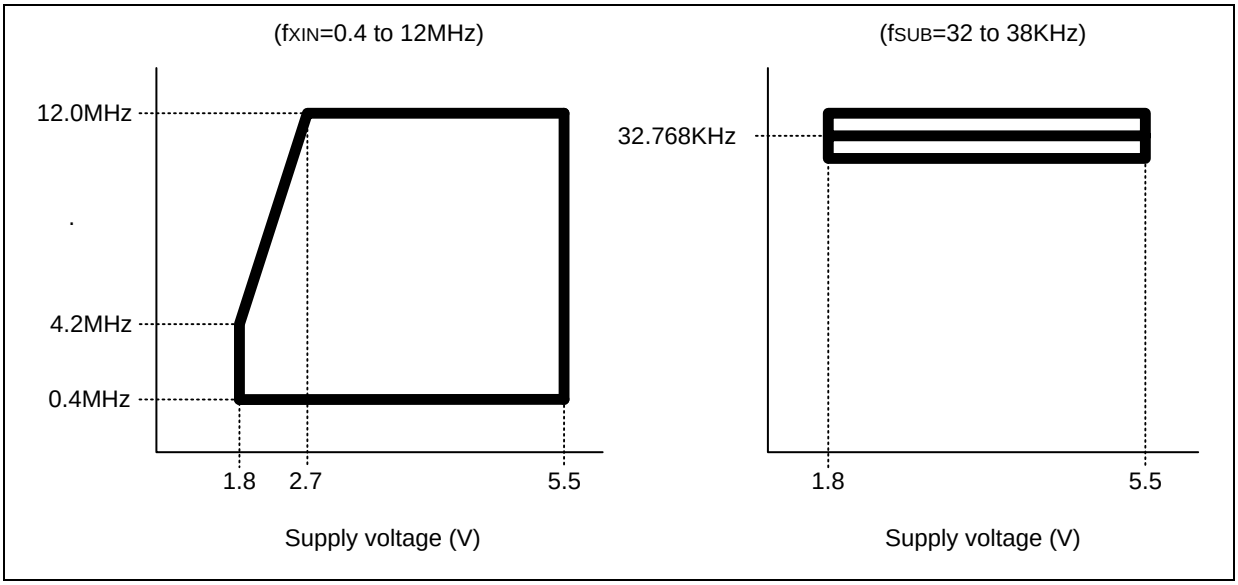



**Figure 7.16 Operating Voltage Range**

## 7.23 Recommended Circuit and Layout

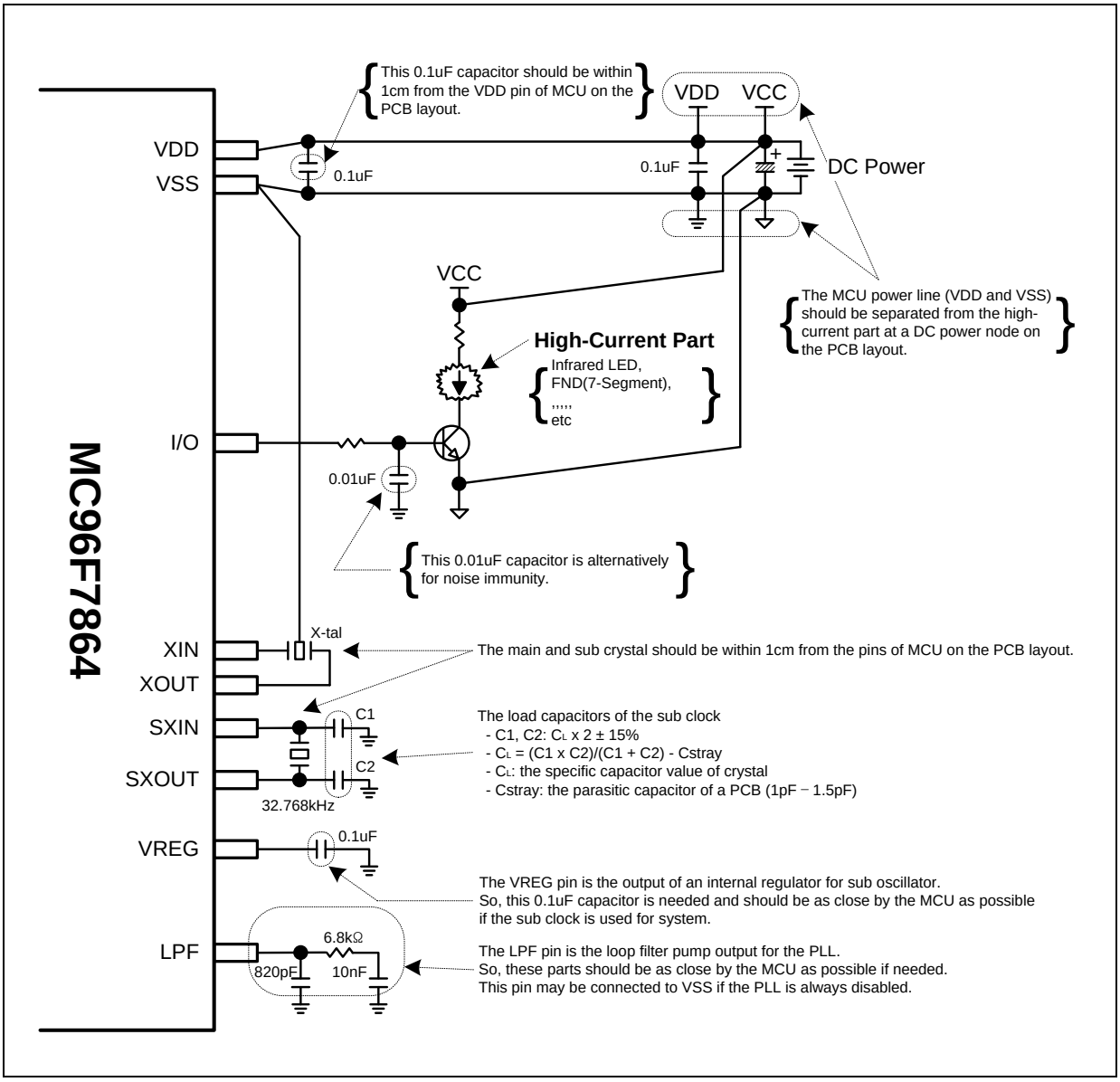
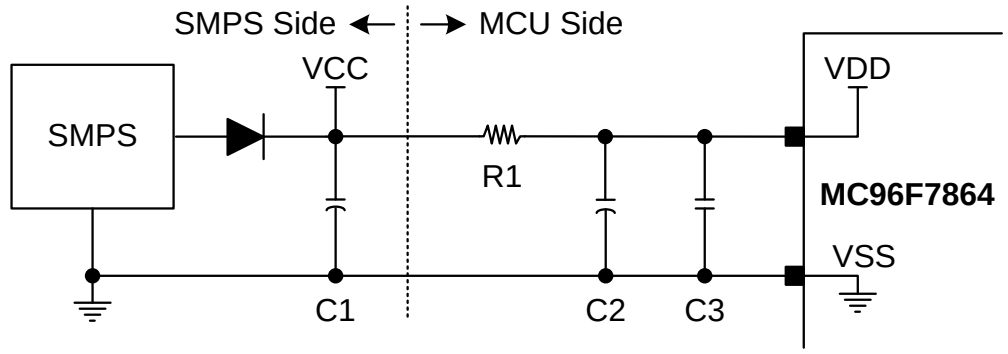



**Figure 7.17 Recommended Circuit and Layout**

## 7.24 Recommanded Circuit and Layout With SMPS Power




1. The C1 capacitor is to flatten out the voltage of the SMPS power, VCC.
   √ Recommended C1: 470uF/25V more.
2. The R1 and C2 are the RC filter for VDD and suppress the ripple of VCC.
   √ Recommended R1: 10Ω - 20Ω
   √ Recommended C2: 47uF/25V more
   √ The R1 and C2 should be as close by the C3 as possible.
3. The C3 capacitor is used for temperature compensation because an electrolytic capacitor becomes worse characteristics at low temperature.
   √ Recommended C3: ceramic capacitor 2.2uF more
   √ The C3 should be within 1cm from VDD pin of MCU on the PCB layout.
4. The above circuit is recommended to improve noise immunity (EFT, Surge, ESD, etc) when the SMPS supplies the VDD of MCU.

**Figure 7.18 Recommended Circuit and Layout with SMPS Power**

## 7.25 Typical Characteristics

These graphs and tables provided in this section are only for design guidance and are not tested or guaranteed. In graphs or tables some data are out of specified operating range (e.g. out of specified VDD range). This is only for information and devices are guaranteed to operate properly only within the specified range.

The data presented in this section is a statistical summary of data collected on units from different lots over a period of time. “Typical” represents the mean of the distribution while “max” or “min” represents (mean + 3σ) and (mean - 3σ) respectively where σ is standard deviation.

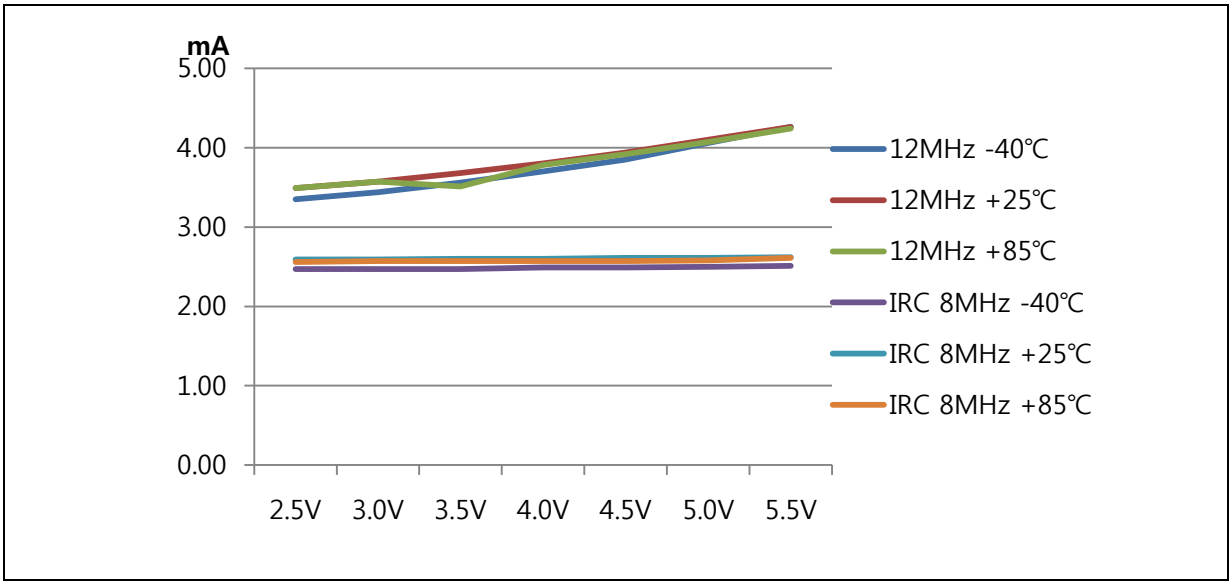



**Figure 7.19 RUN (IDD1 ) Current**

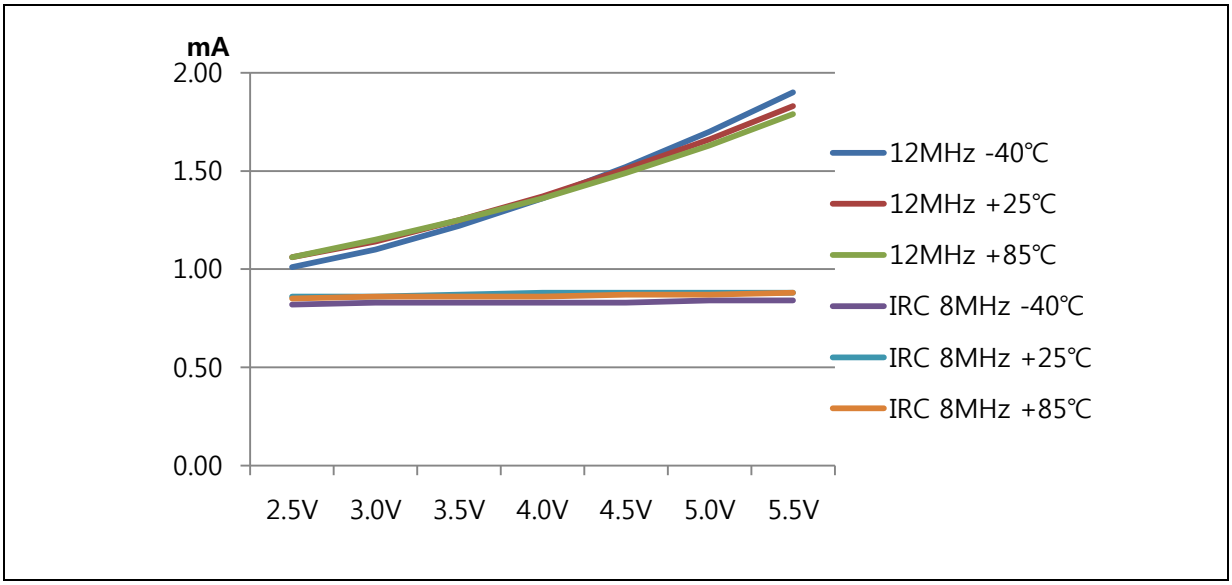



**Figure 7.20 IDLE (IDD2) Current**

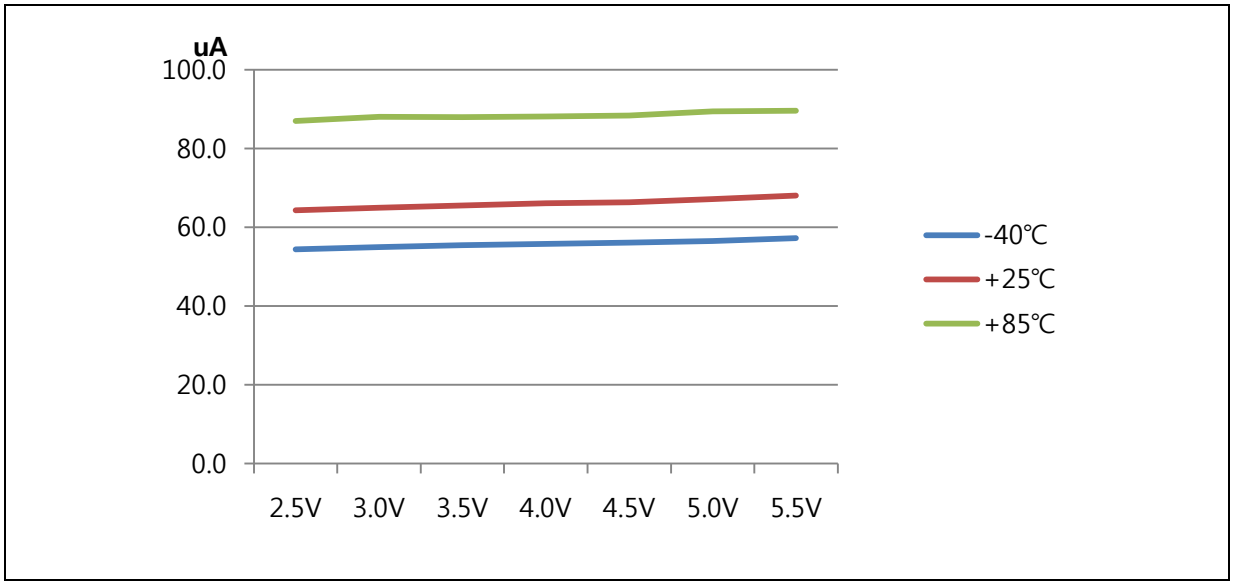



**Figure 7.21 SUB RUN (IDD3) Current**

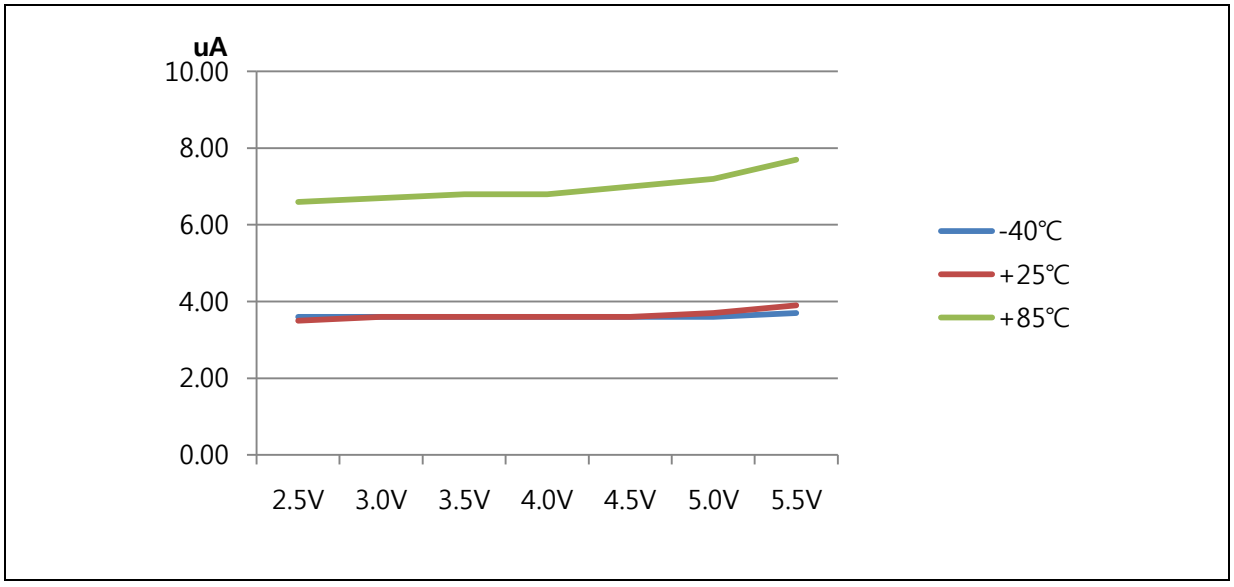



**Figure 7.22 SUB IDLE (IDD4) Current**

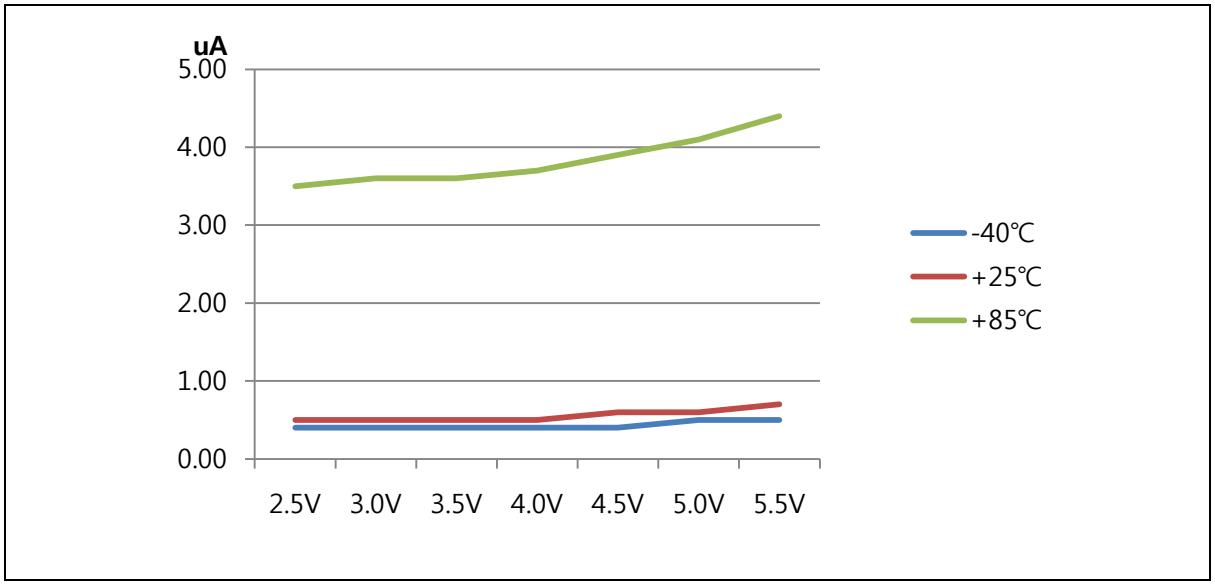



**Figure 7.23 STOP (IDD5) Current**

# 8. Memory

The MC96F7864 addresses two separate address memory stores: Program memory and Data memory. The logical separation of Program and Data memory allows Data memory to be accessed by 8-bit addresses, which makes the 8-bit CPU access the data memory more rapidly. Nevertheless, 16-bit Data memory addresses can also be generated through the DPTR register.

MC96F7864 provides on-chip 64k bytes of the ISP type flash program memory, which can be read and written to. Internal data memory (IRAM) is 256 bytes and it includes the stack area. External data memory (XRAM) is 3,072 bytes and it includes 40 bytes of LCD display RAM.

## 8.1 Program Memory

A 16-bit program counter is capable of addressing up to 64k bytes, this device has 64k bytes program memory space.

Figure 8-1 shows the map of the lower part of the program memory. After reset, the CPU begins execution from location 0000H. Each interrupt is assigned a fixed location in program memory. The interrupt causes the CPU to jump to that location, where it commences execution of the service routine. External interrupt 0-7, for example, is assigned to location 002BH. If external interrupt 0-7 is going to be used, its service routine must begin at location 002BH. If the interrupt is not going to be used, its service location is available as general purpose program memory. If an interrupt service routine is short enough (as is often the case in control applications), it can reside entirely within that 8 byte interval. Longer service routines can use a jump instruction to skip over subsequent interrupt locations, if other interrupts are in use.

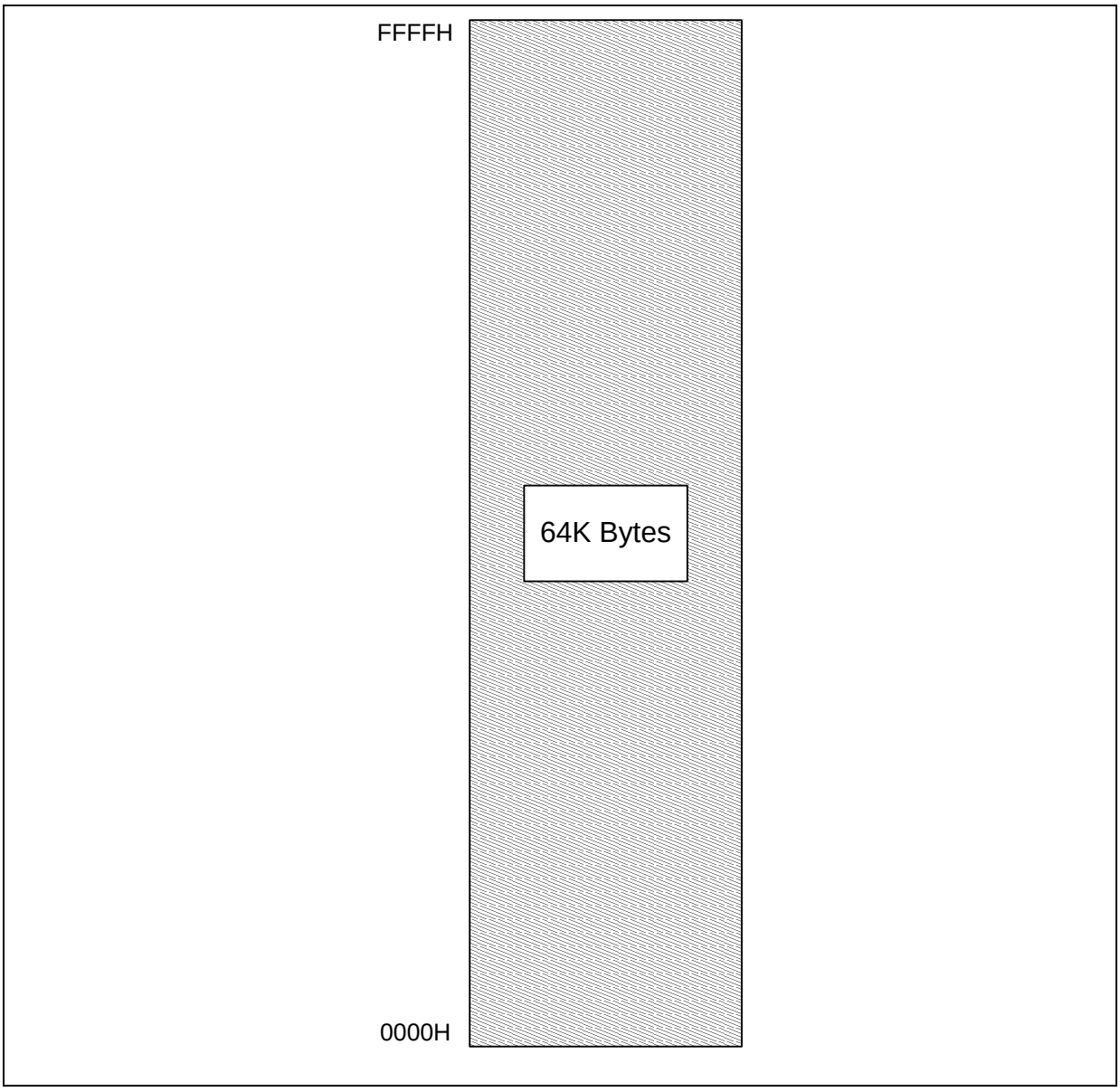



**Figure 8.1 Program Memory**

- 64k Bytes Including Interrupt Vector Region

## 8.2 Data Memory

Figure 8-2 shows the internal data memory space available.

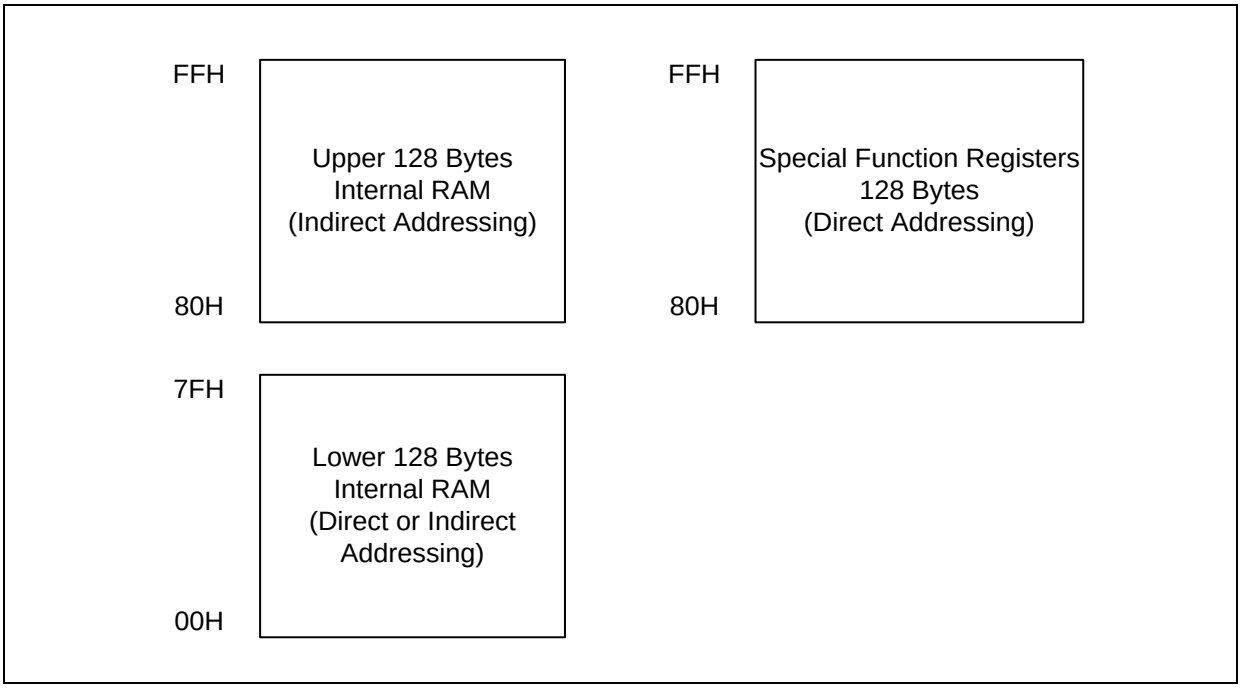



**Figure 8.2 Data Memory Map**

The internal data memory space is divided into three blocks, which are generally referred to as the lower 128 bytes, upper 128 bytes, and SFR space.

Internal data memory addresses are always one byte wide, which implies an address space of only 256 bytes. However, in fact the addressing modes for internal RAM can accommodate up to 384 bytes by using a simple trick. Direct addresses higher than 7FH access one memory space and indirect addresses higher than 7FH access a different memory space. Thus Figure 8-2 shows the upper 128 bytes and SFR space occupying the same block of addresses, 80H through FFH, although they are physically separate entities.

The lower 128 bytes of RAM are present in all 8051 devices as mapped in Figure 8-3. The lowest 32 bytes are grouped into 4 banks of 8 registers. Program instructions call out these registers as R0 through R7. Two bits in the Program Status Word select which register bank is in use. This allows more efficient use of code space, since register instructions are shorter than instructions that use direct addressing.

The next 16 bytes above the register banks form a block of bit-addressable memory space. The 8051 instruction set includes a wide selection of single-bit instructions, and the 128 bits in this area can be directly addressed by these instructions. The bit addresses in this area are 00H through 7FH.

All of the bytes in the lower 128 bytes can be accessed by either direct or indirect addressing. The upper 128 bytes RAM can only be accessed by indirect addressing. These spaces are used for data RAM and stack.

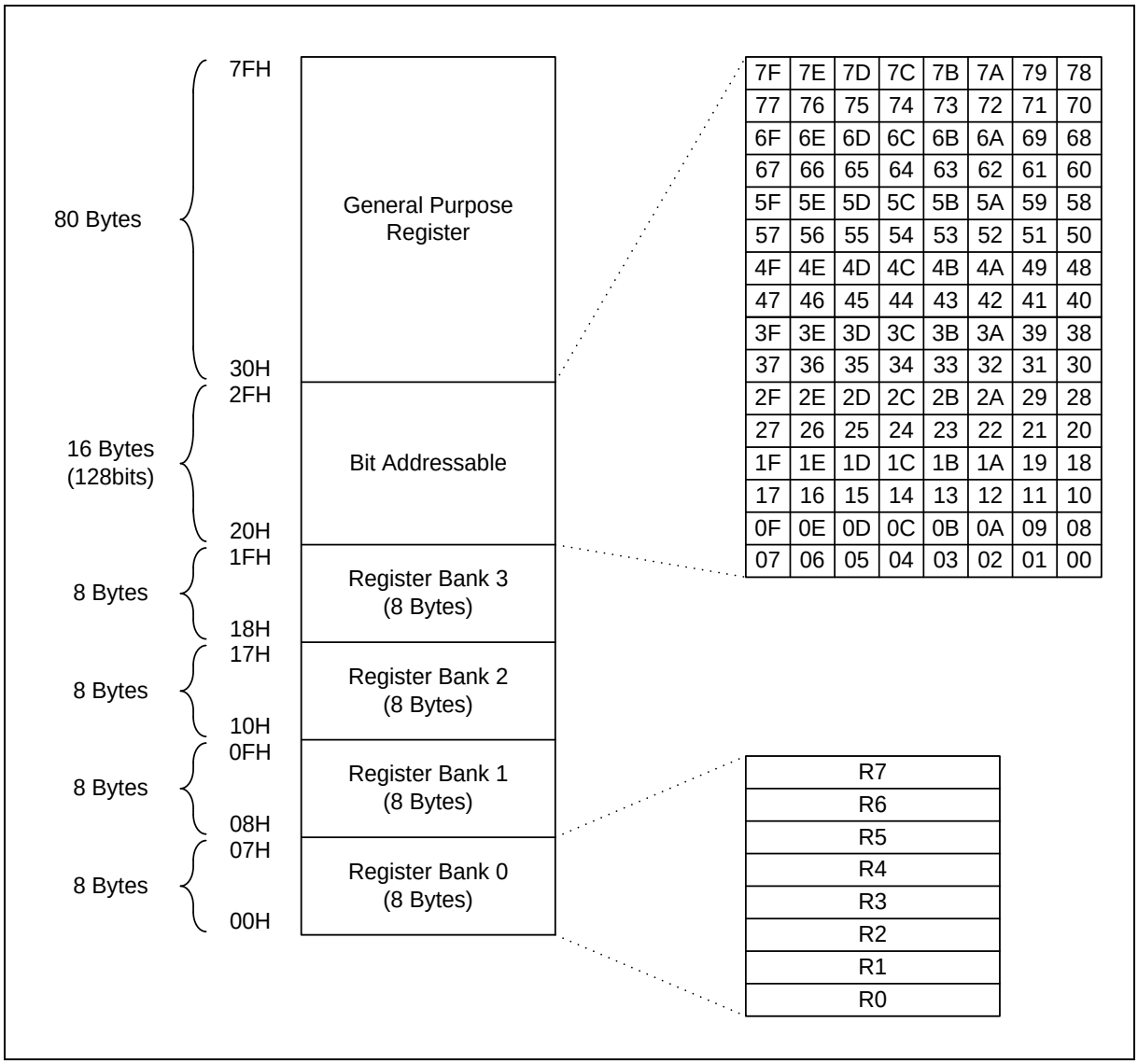



**Figure 8.3 Lower 128 Bytes RAM**

## 8.3 XRAM Memory

MC96F7864 has 3,072 bytes XRAM. This area has no relation with RAM/FLASH. It can be read and written to through SFR with 8-bit unit.

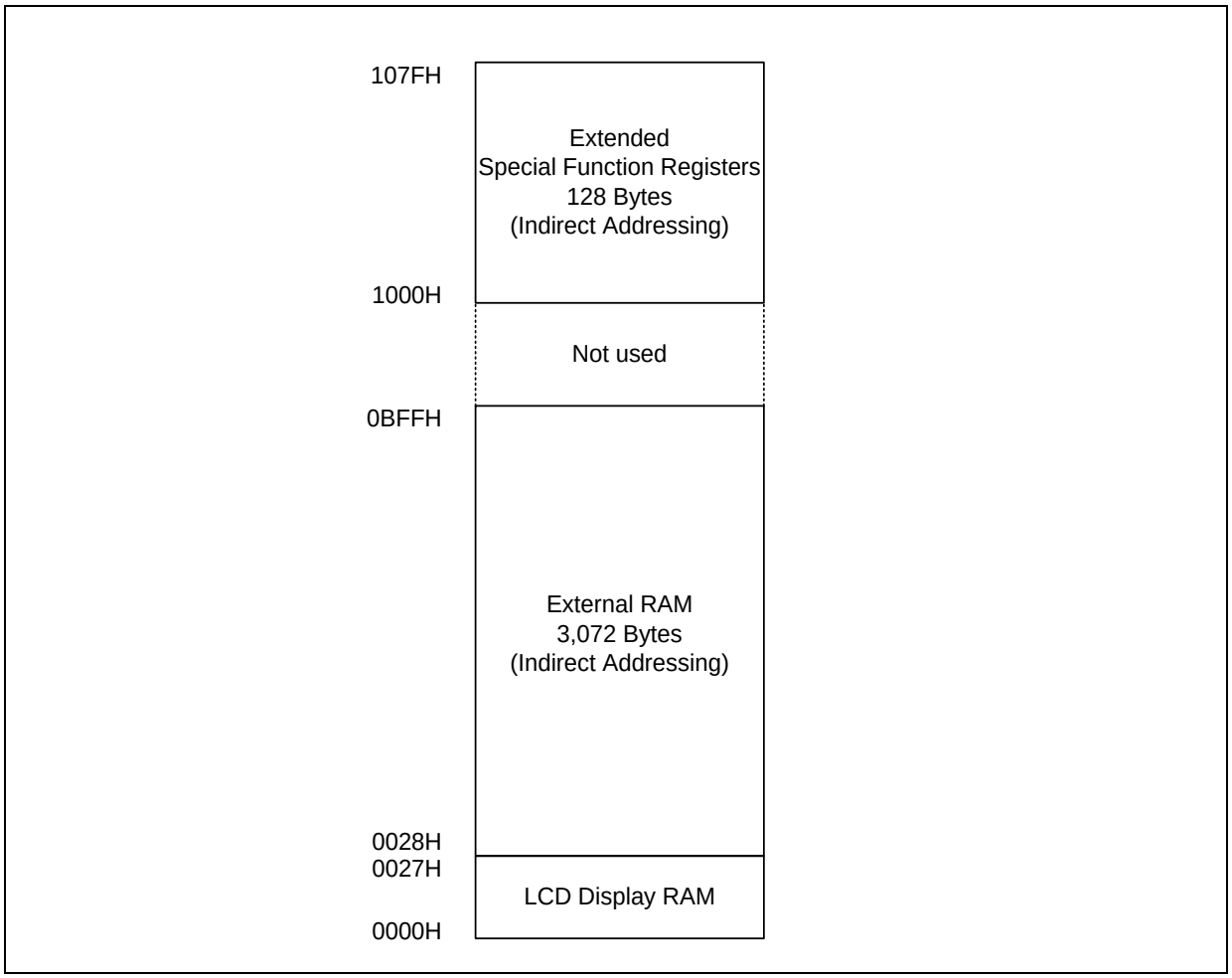



**Figure 8.4 XDATA Memory Area**

## 8.4 SFR Map

### 8.4.1 SFR Map Summary

**Table 8-1 SFR Map Summary**

- Reserved

M8051 compatible

| | 00H/8H(1) | 01H/9H | 02H/0AH | 03H/0BH | 04H/0CH | 05H/0DH | 06H/0EH | 07H/0FH |
|---|---|---|---|---|---|---|---|---|
| 0F8H | IP1 | – | FSADRH | FSADRM | FSADRL | FIDR | FMCR | – |
| 0F0H | B | LCDCCR | LCDCRL | LCDCRH | – | SPI3CR | SPI3DR | SPI3SR |
| 0E8H | RSTFR | P8IO | ADCCRL | ADCCRH | TINTCR | SPI2CR | SPI2DR | SPI2SR |
| 0E0H | ACC | P7IO | ADCDRL | ADCDRH | – | – | – | – |
| 0D8H | LVRCR | P6IO | T6CRL | T6CRH | T6ADRL | T6ADRH | T6BDRL | T6BDRH |
| 0D0H | PSW | P5IO | T5CRL | T5CRH | T5ADRL | T5ADRH | T5BDRL | T5BDRH |
| 0C8H | OSCCR | P4IO | T4CRL | T4CRH | T4ADRL | T4ADRH | T4BDRL | T4BDRH |
| 0C0H | P6 | P3IO | T3CRL | T3CRH | T3ADRL | T3ADRH | T3BDRL | T3BDRH |
| 0B8H | IP | P2IO | T1CR | T1CNT | T1DR/ T1CDR | T2CR | T2CNT | T2DR/ T2CDR |
| 0B0H | P5 | P1IO | T0CR | T0CNT | T0DR/ T0CDR | P0DB | P1DB | P46DB |
| 0A8H | IE | IE1 | IE2 | IE3 | TIFLAG | EIPOL1 | EIPOL2L | EIPOL2H |
| 0A0H | P4 | P0IO | EO | EIFLAG0 | EIFLAG1 | EIFLAG2 | EIPOL0L | EIPOL0H |
| 98H | P3 | P2PU | P3PU | P4PU | P5PU | P6PU | P7PU | P8PU |
| 90H | P2 | P7 | P8 | P0PU | P1PU | PLLCR | WTCR | BUZCR |
| 88H | P1 | WTDR/ WTCNT | SCCR | BITCR | BITCNT | WDTCR | WDTDR/ WDTCNT | BUZDR |
| 80H | P0 | SP | DPL | DPH | DPL1 | DPH1 | LVICR | PCON |

NOTE) These registers are bit-addressable.

**Table 8-2 Extended SFR Map Summary**

| - | Reserved |
|---|---|

| | 00H/8H[1] | 01H/9H | 02H/0AH | 03H/0BH | 04H/0CH | 05H/0DH | 06H/0EH | 07H/0FH |
|---|---|---|---|---|---|---|---|---|
| 1078H | PWMCRL | PWMCRH | PWMADRL | PWMADRH | PWMBDRL | PWMBDRH | PWMCNTL | PWMCNTH |
| 1070H | – | – | – | – | – | – | – | – |
| 1068H | – | – | – | – | – | – | – | – |
| 1060H | T8DLYA | T8DLYB | T8DLYC | T8DR | T8CAPR | T8CNT | PWMDLYDR | NFILDR |
| 1058H | T8PPRL | T8PPRH | T8ADRL | T8ADRH | T8BDRL | T8BDRH | T8CDRL | T8CDRH |
| 1050H | T7CR | T7CNT/ T7DR/ T7CAPR | T8CR | T8PCR1 | T8PCR2 | T8PCR3 | T8ISR | T8MSK |
| 1048H | UART4CR1 | UART4CR2 | UART4CR3 | UART4ST | UART4BD | UART4DR | – | – |
| 1040H | UART3CR1 | UART3CR2 | UART3CR3 | UART3ST | UART3BD | UART3DR | – | – |
| 1038H | UART2CR1 | UART2CR2 | UART2CR3 | UART2ST | UART2BD | UART2DR | PGINTCR | PGIFLAG |
| 1030H | USI1ST1 | USI1ST2 | USI1BD | USI1SDHR | USI1DR | USI1SCLR | USI1SCHR | USI1SAR |
| 1028H | USI1CR1 | USI1CR2 | USI1CR3 | USI1CR4 | – | – | – | – |
| 1020H | USI0ST1 | USI0ST2 | USI0BD | USI0SDHR | USI0DR | USI0SCLR | USI0SCHR | USI0SAR |
| 1018H | USI0CR1 | USI0CR2 | USI0CR3 | USI0CR4 | – | – | – | – |
| 1010H | P4FSRL | P4FSRH | P5FSRL | P5FSRH | P6FSRL | P6FSRH | P7FSRL | P7FSRH |
| 1008H | P8OD | P0FSR | P1FSR | P8FSR | P2FSRL | P2FSRH | P3FSRL | P3FSRH |
| 1000H | P0OD | P1OD | P2OD | P3OD | P4OD | P5OD | P6OD | P7OD |

### 8.4.2 SFR Map

**Table 8-3 SFR Map**

| Address | Function | Symbol | R/W | @Reset | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 80H | P0 Data Register | P0 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 81H | Stack Pointer | SP | R/W | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 82H | Data Pointer Register Low | DPL | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 83H | Data Pointer Register High | DPH | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 84H | Data Pointer Register Low 1 | DPL1 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 85H | Data Pointer Register High 1 | DPH1 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 86H | Low Voltage Indicator Control Register | LVICR | R/W | – | – | 0 | 0 | 0 | 0 | 0 | 0 |
| 87H | Power Control Register | PCON | R/W | 0 | – | – | – | 0 | 0 | 0 | 0 |
| 88H | P1 Data Register | P1 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 89H | Watch Timer Data Register | WTDR | W | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | Watch Timer Counter Register | WTCNT | R | – | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8AH | System and Clock Control Register | SCCR | R/W | – | 0 | – | – | – | – | 0 | 0 |
| 8BH | Basic Interval Timer Control Register | BITCR | R/W | 0 | 0 | 0 | – | 0 | 0 | 0 | 1 |
| 8CH | Basic Interval Timer Counter Register | BITCNT | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8DH | Watch Dog Timer Control Register | WDTCR | R/W | 0 | 0 | 0 | – | – | – | 0 | 0 |
| 8EH | Watch Dog Timer Data Register | WDTDR | W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | Watch Dog Timer Counter Register | WDTCNT | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8FH | BUZZER Data Register | BUZDR | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 90H | P2 Data Register | P2 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 91H | P7 Data Register | P7 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 92H | P8 Data Register | P8 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 93H | P0 Pull-up Resistor Selection Register | P0PU | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 94H | P1 Pull-up Resistor Selection Register | P1PU | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 95H | Phase Locked Loop Control Register | PLLCR | R/W | – | – | 0 | 0 | 0 | 0 | 0 | 0 |
| 96H | Watch Timer Control Register | WTCR | R/W | 0 | – | 0 | 0 | 0 | 0 | 0 | 0 |
| 97H | BUZZER Control Register | BUZCR | R/W | – | – | – | – | 0 | 0 | 0 | 0 |
| 98H | P3 Data Register | P3 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 99H | P2 Pull-up Resistor Selection Register | P2PU | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9AH | P3 Pull-up Resistor Selection Register | P3PU | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9BH | P4 Pull-up Resistor Selection Register | P4PU | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9CH | P5 Pull-up Resistor Selection Register | P5PU | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9DH | P6 Pull-up Resistor Selection Register | P6PU | R/W | – | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9EH | P7 Pull-up Resistor Selection Register | P7PU | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9FH | P8 Pull-up Resistor Selection Register | P8PU | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 8-3 SFR Map (Continued)**

| Address | Function | Symbol | R/W | @Reset | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| A0H | P4 Data Register | P4 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A1H | P0 Direction Register | P0IO | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A2H | Extended Operation Register | EO | R/W | – | – | – | 0 | – | 0 | 0 | 0 |
| A3H | External Interrupt Flag Register 0 | EIFLAG0 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A4H | External Interrupt Flag Register 1 | EIFLAG1 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A5H | External Interrupt Flag Register 2 | EIFLAG2 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A6H | External Interrupt Polarity 0 Low Register | EIPOL0L | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A7H | External Interrupt Polarity 0 High Register | EIPOL0H | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A8H | Interrupt Enable Register | IE | R/W | 0 | – | 0 | 0 | 0 | 0 | 0 | 0 |
| A9H | Interrupt Enable Register 1 | IE1 | R/W | – | – | 0 | 0 | 0 | 0 | 0 | 0 |
| AAH | Interrupt Enable Register 2 | IE2 | R/W | – | – | – | 0 | 0 | 0 | 0 | 0 |
| ABH | Interrupt Enable Register 3 | IE3 | R/W | – | – | 0 | 0 | 0 | 0 | 0 | 0 |
| ACH | Timer Interrupt Flag Register | TIFLAG | R/W | – | – | 0 | 0 | 0 | 0 | 0 | 0 |
| ADH | External Interrupt Polarity 1 Register | EIPOL1 | R/W | – | – | 0 | 0 | 0 | 0 | 0 | 0 |
| AEH | External Interrupt Polarity 2 Low Register | EIPOL2L | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| AFH | External Interrupt Polarity 2 High Register | EIPOL2H | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B0H | P5 Data Register | P5 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B1H | P1 Direction Register | P1IO | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B2H | Timer 0 Control Register | T0CR | R/W | 0 | – | 0 | 0 | 0 | 0 | 0 | 0 |
| B3H | Timer 0 Counter Register | T0CNT | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B4H | Timer 0 Data Register | T0DR | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | Timer 0 Capture Data Register | T0CDR | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B5H | P0 Debounce Enable Register | P0DB | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B6H | P1 Debounce Enable Register | P1DB | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B7H | P4/6 Debounce Enable Register | P46DB | R/W | – | – | – | 0 | 0 | 0 | 0 | 0 |
| B8H | Interrupt Priority Register | IP | R/W | – | – | 0 | 0 | 0 | 0 | 0 | 0 |
| B9H | P2 Direction Register | P2IO | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| BAH | Timer 1 Control Register | T1CR | R/W | 0 | – | 0 | 0 | 0 | 0 | 0 | 0 |
| BBH | Timer 1 Counter Register | T1CNT | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| BCH | Timer 1 Data Register | T1DR | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | Timer 1 Capture Data Register | T1CDR | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| BDH | Timer 2 Control Register | T2CR | R/W | 0 | – | 0 | 0 | 0 | 0 | 0 | 0 |
| BEH | Timer 2 Counter Register | T2CNT | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| BFH | Timer 2 Data Register | T2DR | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | Timer 2 Capture Data Register | T2CDR | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 8-3 SFR Map (Continued)**

| Address | Function | Symbol | R/W | @Reset | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| C0H | P6 Data Register | P6 | R/W | – | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| C1H | P3 Direction Register | P3IO | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| C2H | Timer 3 Control Low Register | T3CRL | R/W | 0 | 0 | 0 | – | – | 0 | 0 | 0 |
| C3H | Timer 3 Control High Register | T3CRH | R/W | 0 | 0 | 0 | 0 | – | – | – | 0 |
| C4H | Timer 3 A Data Low Register | T3ADRL | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| C5H | Timer 3 A Data High Register | T3ADRH | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| C6H | Timer 3 B Data Low Register | T3BDRL | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| C7H | Timer 3 B Data High Register | T3BDRH | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| C8H | Oscillator Control Register | OSCCR | R/W | – | – | 0 | 0 | 1 | 0 | 0 | 0 |
| C9H | P4 Direction Register | P4IO | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CAH | Timer 4 Control Low Register | T4CRL | R/W | 0 | 0 | 0 | – | – | 0 | 0 | 0 |
| CBH | Timer 4 Control High Register | T4CRH | R/W | 0 | 0 | 0 | 0 | – | – | – | 0 |
| CCH | Timer 4 A Data Low Register | T4ADRL | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| CDH | Timer 4 A Data High Register | T4ADRH | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| CEH | Timer 4 B Data Low Register | T4BDRL | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| CFH | Timer 4 B Data High Register | T4BDRH | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| D0H | Program Status Word Register | PSW | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| D1H | P5 Direction Register | P5IO | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| D2H | Timer 5 Control Low Register | T5CRL | R/W | 0 | 0 | 0 | – | – | 0 | 0 | 0 |
| D3H | Timer 5 Control High Register | T5CRH | R/W | 0 | 0 | 0 | 0 | – | – | – | 0 |
| D4H | Timer 5 A Data Low Register | T5ADRL | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| D5H | Timer 5 A Data High Register | T5ADRH | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| D6H | Timer 5 B Data Low Register | T5BDRL | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| D7H | Timer 5 B Data High Register | T5BDRH | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| D8H | Low Voltage Reset Control Register | LVRCR | R/W | 0 | – | – | 0 | 0 | 0 | 0 | 0 |
| D9H | P6 Direction Register | P6IO | R/W | – | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| DAH | Timer 6 Control Low Register | T6CRL | R/W | 0 | 0 | 0 | – | – | 0 | 0 | 0 |
| DBH | Timer 6 Control High Register | T6CRH | R/W | 0 | 0 | 0 | 0 | – | – | – | 0 |
| DCH | Timer 6 A Data Low Register | T6ADRL | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| DDH | Timer 6 A Data High Register | T6ADRH | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| DEH | Timer 6 B Data Low Register | T6BDRL | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| DFH | Timer 6 B Data High Register | T6BDRH | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Table 8-3 SFR Map (Continued)**

| Address | Function | Symbol | R/W | @Reset | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E0H | Accumulator A Register | ACC | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| E1H | P7 Direction Register | P7IO | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| E2H | A/D Converter Data Low Register | ADCDRL | R | x | x | x | x | x | x | x | x |
| E3H | A/D Converter Data High Register | ADCDRH | R | x | x | x | x | x | x | x | x |
| E4H | Reserved | – | – | – | | | | | | | |
| E5H | Reserved | – | – | – | | | | | | | |
| E6H | Reserved | – | – | – | | | | | | | |
| E7H | Reserved | – | – | – | | | | | | | |
| E8H | Reset Flag Register | RSTFR | R/W | 1 | x | 0 | 0 | x | – | – | 0 |
| E9H | P8 Direction Register | P8IO | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| EAH | A/D Converter Control Low Register | ADCCRL | R/W | 0 | 0 | – | 0 | 0 | 0 | 0 | 0 |
| EBH | A/D Converter Control High Register | ADCCRH | R/W | – | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ECH | Timer Interrupt Control Register | TINTCR | R/W | – | – | 0 | 0 | 0 | 0 | 0 | 0 |
| EDH | SPI2 Control Register | SPI2CR | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| EEH | SPI2 Data Register | SPI2DR | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| EFH | SPI2 Status Register | SPI2SR | R/W | 0 | 0 | 0 | – | 0 | 0 | – | – |
| F0H | B Register | B | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F1H | LCD Contrast Control Register | LCDCCR | R/W | – | – | – | – | 0 | 0 | 0 | 0 |
| F2H | LCD Driver Control Low Register | LCDCRL | R/W | – | – | – | 0 | 0 | 0 | 0 | 0 |
| F3H | LCD Driver Control High Register | LCDCRH | R/W | 0 | – | – | 0 | 0 | 0 | 0 | 0 |
| F4H | Reserved | – | – | – | | | | | | | |
| F5H | SPI3 Control Register | SPI3CR | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F6H | SPI3 Data Register | SPI3DR | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F7H | SPI3 Status Register | SPI3SR | R/W | 0 | 0 | 0 | – | 0 | 0 | – | – |
| F8H | Interrupt Priority Register 1 | IP1 | R/W | – | – | 0 | 0 | 0 | 0 | 0 | 0 |
| F9H | Reserved | – | – | – | | | | | | | |
| FAH | Flash Sector Address High Register | FSADRH | R/W | – | – | – | – | 0 | 0 | 0 | 0 |
| FBH | Flash Sector Address Middle Register | FSADRM | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| FCH | Flash Sector Address Low Register | FSADRL | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| FDH | Flash Identification Register | FIDR | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| FEH | Flash Mode Control Register | FMCR | R/W | 0 | – | – | – | – | 0 | 0 | 0 |
| FFH | Reserved | – | – | – | | | | | | | |

**Table 8-4 Extended SFR Map**

| Address | Function | Symbol | R/W | @Reset | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1000H | P0 Open-drain Selection Register | P0OD | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1001H | P1 Open-drain Selection Register | P1OD | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1002H | P2 Open-drain Selection Register | P2OD | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1003H | P3 Open-drain Selection Register | P3OD | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1004H | P4 Open-drain Selection Register | P4OD | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1005H | P5 Open-drain Selection Register | P5OD | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1006H | P6 Open-drain Selection Register | P6OD | R/W | – | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1007H | P7 Open-drain Selection Register | P7OD | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1008H | P8 Open-drain Selection Register | P8OD | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1009H | P0 Function Selection Register | P0FSR | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 100AH | P1 Function Selection Register | P1FSR | R/W | – | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 100BH | P8 Function Selection Register | P8FSR | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 100CH | P2 Function Selection Low Register | P2FSRL | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 100DH | P2 Function Selection High Register | P2FSRH | R/W | – | – | – | 0 | 0 | 0 | 0 | 0 |
| 100EH | P3 Function Selection Low Register | P3FSRL | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 100FH | P3 Function Selection High Register | P3FSRH | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1010H | P4 Function Selection Low Register | P4FSRL | R/W | – | – | – | 0 | 0 | 0 | 0 | 0 |
| 1011H | P4 Function Selection High Register | P4FSRH | R/W | – | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1012H | P5 Function Selection Low Register | P5FSRL | R/W | – | – | 0 | 0 | 0 | 0 | 0 | 0 |
| 1013H | P5 Function Selection High Register | P5FSRH | R/W | – | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1014H | P6 Function Selection Low Register | P6FSRL | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1015H | P6 Function Selection High Register | P6FSRH | R/W | – | – | – | – | – | 0 | 0 | 0 |
| 1016H | P7 Function Selection Low Register | P7FSRL | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1017H | P7 Function Selection High Register | P7FSRH | R/W | – | – | – | – | 0 | 0 | 0 | 0 |
| 1018H | USI0 Control Register 1 | USI0CR1 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1019H | USI0 Control Register 2 | USI0CR2 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 101AH | USI0 Control Register 3 | USI0CR3 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 101BH | USI0 Control Register 4 | USI0CR4 | R/W | 0 | – | 0 | 0 | 0 | 0 | 0 | 0 |
| 101CH | Reserved | – | – | – | | | | | | | |
| 101DH | Reserved | – | – | – | | | | | | | |
| 101EH | Reserved | – | – | – | | | | | | | |
| 101FH | Reserved | – | – | – | | | | | | | |

**Table 8-4 Extended SFR Map (Continued)**

| Address | Function | Symbol | R/W | @Reset | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1020H | USI0 Status Register 1 | USI0ST1 | R/W | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1021H | USI0 Status Register 2 | USI0ST2 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1022H | USI0 Baud Rate Generation Register | USI0BD | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1023H | USI0 SDA Hold Time Register | USI0SDHR | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1024H | USI0 Data Register | USI0DR | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1025H | USI0 SCL Low Period Register | USI0SCLR | R/W | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1026H | USI0 SCL High Period Register | USI0SCHR | R/W | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1027H | USI0 Slave Address Register | USI0SAR | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1028H | USI1 Control Register 1 | USI1CR1 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1029H | USI1 Control Register 2 | USI1CR2 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 102AH | USI1 Control Register 3 | USI1CR3 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 102BH | USI1 Control Register 4 | USI1CR4 | R/W | 0 | – | 0 | 0 | 0 | 0 | 0 | 0 |
| 102CH | Reserved | – | – | – | | | | | | | |
| 102DH | Reserved | – | – | – | | | | | | | |
| 102EH | Reserved | – | – | – | | | | | | | |
| 102FH | Reserved | – | – | – | | | | | | | |
| 1030H | USI1 Status Register 1 | USI1ST1 | R/W | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1031H | USI1 Status Register 2 | USI1ST2 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1032H | USI1 Baud Rate Generation Register | USI1BD | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1033H | USI1 SDA Hold Time Register | USI1SDHR | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1034H | USI1 Data Register | USI1DR | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1035H | USI1 SCL Low Period Register | USI1SCLR | R/W | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1036H | USI1 SCL High Period Register | USI1SCHR | R/W | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1037H | USI1 Slave Address Register | USI1SAR | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1038H | UART2 Control Register 1 | UART2CR1 | R/W | – | – | 0 | 0 | 0 | 0 | 0 | – |
| 1039H | UART2 Control Register 2 | UART2CR2 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 103AH | UART2 Control Register 3 | UART2CR3 | R/W | – | 0 | – | – | – | 0 | 0 | 0 |
| 103BH | UART2 Status Register | UART2ST | R/W | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 103CH | UART2 Baud Rate Generation Register | UART2BD | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 103DH | UART2 Data Register | UART2DR | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 103EH | PWM Generator Interrupt Control Register | PGINTCR | R/W | – | – | – | – | 0 | 0 | 0 | 0 |
| 103FH | PWM Generator Interrupt Flag Register | PGIFLAG | R/W | – | – | – | – | 0 | 0 | 0 | 0 |

**Table 8-4 Extended SFR Map (Continued)**

| Address | Function | Symbol | R/W | @Reset | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1040H | UART3 Control Register 1 | UART3CR1 | R/W | – | – | 0 | 0 | 0 | 0 | 0 | – |
| 1041H | UART3 Control Register 2 | UART3CR2 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1042H | UART3 Control Register 3 | UART3CR3 | R/W | – | 0 | – | – | – | 0 | 0 | 0 |
| 1043H | UART3 Status Register | UART3ST | R/W | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1044H | UART3 Baud Rate Generation Register | UART3BD | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1045H | UART3 Data Register | UART3DR | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1046H | Reserved | – | – | – | | | | | | | |
| 1047H | Reserved | – | – | – | | | | | | | |
| 1048H | UART4 Control Register 1 | UART4CR1 | R/W | – | – | 0 | 0 | 0 | 0 | 0 | – |
| 1049H | UART4 Control Register 2 | UART4CR2 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 104AH | UART4 Control Register 3 | UART4CR3 | R/W | – | 0 | – | – | – | 0 | 0 | 0 |
| 104BH | UART4 Status Register | UART4ST | R/W | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 104CH | UART4 Baud Rate Generation Register | UART4BD | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 104DH | UART4 Data Register | UART4DR | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 104EH | Reserved | – | – | – | | | | | | | |
| 104FH | Reserved | – | – | – | | | | | | | |
| 1050H | Timer 7 Control Register | T7CR | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1051H | Timer 7 Counter Register | T7CNT | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Timer 7 Data Register | T7DR | W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1051H | Timer 7 Capture Data Register | T7CAPR | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1052H | Timer 8 Control Register | T8CR | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1053H | Timer 8 PWM Control Register 1 | T8PCR1 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1054H | Timer 8 PWM Control Register 2 | T8PCR2 | R/W | 0 | – | 0 | 0 | 0 | 0 | 0 | 0 |
| 1055H | Timer 8 PWM Control Register 3 | T8PCR3 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1056H | Timer 8 Interrupt Status Register | T8ISR | R/W | 0 | 0 | 0 | 0 | 0 | – | – | – |
| 1057H | Timer 8 Interrupt Mask Register | T8MSK | R/W | 0 | 0 | 0 | 0 | 0 | – | – | – |
| 1058H | Timer 8 PWM Period Low Register | T8PPRL | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1059H | Timer 8 PWM Period High Register | T8PPRH | R/W | – | – | – | – | – | – | 0 | 0 |
| 105AH | Timer 8 PWM A Duty Low Register | T8ADRL | R/W | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 105BH | Timer 8 PWM A Duty High Register | T8ADRH | R/W | – | – | – | – | – | – | 0 | 0 |
| 105CH | Timer 8 PWM B Duty Low Register | T8BDRL | R/W | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 105DH | Timer 8 PWM B Duty High Register | T8BDRH | R/W | – | – | – | – | – | – | 0 | 0 |
| 105EH | Timer 8 PWM C Duty Low Register | T8CDRL | R/W | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 105FH | Timer 8 PWM C Duty High Register | T8CDRH | R/W | – | – | – | – | – | – | 0 | 0 |

**Table 8-4 Extended SFR Map (Continued)**

| Address | Function | Symbol | R/W | @Reset | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1060H | Timer 8 PWM A Delay Register | T8DLYA | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1061H | Timer 8 PWM B Delay Register | T8DLYB | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1062H | Timer 8 PWM C Delay Register | T8DLYC | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1063H | Timer 8 Data Register | T8DR | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1064H | Timer 8 Capture Data Register | T8CAPR | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1065H | Timer 8 Counter Register | T8CNT | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1066H | PWM Generator Delay Data Register | PWMDLYDR | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1067H | PWM Generator Noise Filter Register | NFILDR | R/W | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1068H | Reserved | – | – | – | | | | | | | |
| 1069H | Reserved | – | – | – | | | | | | | |
| 106AH | Reserved | – | – | – | | | | | | | |
| 106BH | Reserved | – | – | – | | | | | | | |
| 106CH | Reserved | – | – | – | | | | | | | |
| 106DH | Reserved | – | – | – | | | | | | | |
| 106EH | Reserved | – | – | – | | | | | | | |
| 106FH | Reserved | – | – | – | | | | | | | |
| 1070H | Reserved | – | – | – | | | | | | | |
| 1071H | Reserved | – | – | – | | | | | | | |
| 1072H | Reserved | – | – | – | | | | | | | |
| 1073H | Reserved | – | – | – | | | | | | | |
| 1074H | Reserved | – | – | – | | | | | | | |
| 1075H | Reserved | – | – | – | | | | | | | |
| 1076H | Reserved | – | – | – | | | | | | | |
| 1077H | Reserved | – | – | – | | | | | | | |
| 1078H | PWM Generator Control Low Register | PWMCRL | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1079H | PWM Generator Control High Register | PWMCRH | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 107AH | PWM Generator A Data Low Register | PWMADRL | R/W | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 107BH | PWM Generator A Data High Register | PWMADRH | R/W | – | – | – | – | – | – | 1 | 1 |
| 107CH | PWM Generator B Data Low Register | PWMBDRL | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 107DH | PWM Generator B Data High Register | PWMBDRH | R/W | – | – | – | – | – | – | 0 | 0 |
| 107EH | PWM Generator Counter Low Register | PWMCNTL | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 107FH | PWM Generator Counter High Register | PWMCNTH | R | – | – | – | – | – | – | 0 | 0 |

### 8.4.3 Compiler Compatible SFR

**ACC (Accumulator Register) : E0H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ACC | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

| | |
|---|---|
| **ACC** | Accumulator |

**B (B Register) : F0H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| B | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

| | |
|---|---|
| **B** | B Register |

**SP (Stack Pointer) : 81H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SP | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 07H

| | |
|---|---|
| **SP** | Stack Pointer |

**DPL (Data Pointer Register Low) : 82H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DPL | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

| | |
|---|---|
| **DPL** | Data Pointer Low |

**DPH (Data Pointer Register High) : 83H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DPH | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

| | |
|---|---|
| **DPH** | Data Pointer High |

**DPL1 (Data Pointer Register Low 1) : 84H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DPL1 | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**DPL1** Data Pointer Low 1

**DPH1 (Data Pointer Register High 1) : 85H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DPH1 | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**DPH1** Data Pointer High 1

**PSW (Program Status Word Register) : D0H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CY | AC | F0 | RS1 | RS0 | OV | F1 | P |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**CY** Carry Flag

**AC** Auxiliary Carry Flag

**F0** General Purpose User-Definable Flag

**RS1** Register Bank Select bit 1

**RS0** Register Bank Select bit 0

**OV** Overflow Flag

**F1** User-Definable Flag

**P** Parity Flag. Set/Cleared by hardware each instruction cycle to indicate an odd/even number of '1' bits in the accumulator

**EO (Extended Operation Register) : A2H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | TRAP_EN | – | DPSEL2 | DPSEL1 | DPSEL0 |
| – | – | – | R/W | – | R/W | R/W | R/W |

Initial value : 00H

**TRAP_EN** Select the Instruction **(Keep always '0')**.

0 Select MOVC @(DPTR++), A

1 Select Software TRAP Instruction

**DPSEL[2:0]** Select Banked Data Pointer Register

| DPSEL2 | DPSEL1 | SPSEL0 | Description |
|---|---|---|---|
| 0 | 0 | 0 | DPTR0 |
| 0 | 0 | 1 | DPTR1 |
| Reserved | | | |

# 9. I/O Ports

## 9.1 I/O Ports

The MC96F7864 has nine groups of I/O ports (P0 ~ P8). Each port can be easily configured by software as I/O pin, internal pull up and open-drain pin to meet various system configurations and design requirements. Also P0, P1, P4, and P6 include function that can generate interrupt according to change of state of the pin.

## 9.2 Port Register

### 9.2.1 Data Register (Px)

Data Register is a bidirectional I/O port. If ports are configured as output ports, data can be written to the corresponding bit of the Px. If ports are configured as input ports, the data can be read from the corresponding bit of the Px.

### 9.2.2 Direction Register (PxIO)

Each I/O pin can be independently used as an input or an output through the PxIO register. Bits cleared in this register will make the corresponding pin of Px to input mode. Set bits of this register will make the pin to output mode. Almost bits are cleared by a system reset, but some bits are set by a system reset.

### 9.2.3 Pull-up Resistor Selection Register (PxPU)

The on-chip pull-up resistor can be connected to I/O ports individually with a pull-up resistor selection register (PxPU). The pull-up register selection controls the pull-up resister enable/disable of each port. When the corresponding bit is 1, the pull-up resister of the pin is enabled. When 0, the pull-up resister is disabled. All bits are cleared by a system reset.

### 9.2.4 Open-drain Selection Register (PxOD)

There are internally open-drain selection registers (PxOD) for P0 ~ P8. The open-drain selection register controls the open-drain enable/disable of each port. Almost ports become push-pull by a system reset, but some ports become open-drain by a system reset.

### 9.2.5 Debounce Enable Register (PxDB)

P0[7:0], P1[5:0], P4[5:3], P61 and P62 support debounce function. Debounce clocks of each ports are fx/1, fx/4, fx/4096, and $f_{SUB}$.

### 9.2.6 Port Function Selection Register (PxFSR)

These registers define alternative functions of ports. Please remember that these registers should be set properly for alternative port function. A reset clears the PxFSR register to ‘00H’, which makes all pins to normal I/O ports.

### 9.2.7 Register Map

**Table 9-1 Port Register Map**

| Name | Address | Dir | Default | Description |
|---|---|---|---|---|
| P0 | 80H | R/W | 00H | P0 Data Register |
| P0IO | A1H | R/W | 00H | P0 Direction Register |
| P0OD | 1000H (ESFR) | R/W | 00H | P0 Open-drain Selection Register |
| P0PU | 93H | R/W | 00H | P0 Pull-up Resistor Selection Register |
| P0DB | B5H | R/W | 00H | P0 Debounce Enable Register |
| P0FSR | 1009H (ESFR) | R/W | 00H | P0 Function Selection Register |
| P1 | 88H | R/W | 00H | P1 Data Register |
| P1IO | B1H | R/W | 00H | P1 Direction Register |
| P1OD | 1001H (ESFR) | R/W | 00H | P1 Open-drain Selection Register |
| P1PU | 94H | R/W | 00H | P1 Pull-up Resistor Selection Register |
| P1DB | B6H | R/W | 00H | P1 Debounce Enable Register |
| P1FSR | 100AH (ESFR) | R/W | 00H | P1 Function Selection Register |
| P2 | 90H | R/W | 00H | P2 Data Register |
| P2IO | B9H | R/W | 00H | P2 Direction Register |
| P2OD | 1002H (ESFR) | R/W | 00H | P2 Open-drain Selection Register |
| P2PU | 99H | R/W | 00H | P2 Pull-up Resistor Selection Register |
| P2FSRH | 100DH (ESFR) | R/W | 00H | P2 Function Selection High Register |
| P2FSRL | 100CH (ESFR) | R/W | 00H | P2 Function Selection Low Register |
| P3 | 98H | R/W | 00H | P3 Data Register |
| P3IO | C1H | R/W | 00H | P3 Direction Register |
| P3OD | 1003H (ESFR) | R/W | 00H | P3 Open-drain Selection Register |
| P3PU | 9AH | R/W | 00H | P3 Pull-up Resistor Selection Register |
| P3FSRH | 100FH (ESFR) | R/W | 00H | P3 Function Selection High Register |
| P3FSRL | 100EH (ESFR) | R/W | 00H | P3 Function Selection Low Register |
| P4 | A0H | R/W | 00H | P4 Data Register |
| P4IO | C9H | R/W | 00H | P4 Direction Register |
| P4OD | 1004H (ESFR) | R/W | 00H | P4 Open-drain Selection Register |
| P4PU | 9BH | R/W | 00H | P4 Pull-up Resistor Selection Register |
| P46DB | B7H | R/W | 00H | P4/P6 Debounce Enable Register |
| P4FSRH | 1011H (ESFR) | R/W | 00H | P4 Function Selection High Register |
| P4FSRL | 1010H (ESFR) | R/W | 00H | P4 Function Selection Low Register |

**Table 9-1 Port Register Map (Continued)**

| Name | Address | Dir | Default | Description |
|---|---|---|---|---|
| P5 | B0H | R/W | 00H | P5 Data Register |
| P5IO | D1H | R/W | 00H | P5 Direction Register |
| P5OD | 1005H (ESFR) | R/W | 00H | P5 Open-drain Selection Register |
| P5PU | 9CH | R/W | 00H | P5 Pull-up Resistor Selection Register |
| P5FSRH | 1013H (ESFR) | R/W | 00H | P5 Function Selection High Register |
| P5FSRL | 1012H (ESFR) | R/W | 00H | P5 Function Selection Low Register |
| P6 | C0H | R/W | 00H | P6 Data Register |
| P6IO | D9H | R/W | 00H | P6 Direction Register |
| P6OD | 1006H (ESFR) | R/W | 00H | P6 Open-drain Selection Register |
| P6PU | 9DH | R/W | 00H | P6 Pull-up Resistor Selection Register |
| P6FSRH | 1015H (ESFR) | R/W | 00H | P6 Function Selection High Register |
| P6FSRL | 1014H (ESFR) | R/W | 00H | P6 Function Selection Low Register |
| P7 | 91H | R/W | 00H | P7 Data Register |
| P7IO | E1H | R/W | 00H | P7 Direction Register |
| P7OD | 1007H (ESFR) | R/W | 00H | P7 Open-drain Selection Register |
| P7PU | 9EH | R/W | 00H | P7 Pull-up Resistor Selection Register |
| P7FSRH | 1017H (ESFR) | R/W | 00H | P7 Function Selection High Register |
| P7FSRL | 1016H (ESFR) | R/W | 00H | P7 Function Selection Low Register |
| P8 | 92H | R/W | 00H | P8 Data Register |
| P8IO | E9H | R/W | 00H | P8 Direction Register |
| P8OD | 1008H (ESFR) | R/W | 00H | P8 Open-drain Selection Register |
| P8PU | 9FH | R/W | 00H | P8 Pull-up Resistor Selection Register |
| P8FSR | 100BH (ESFR) | R/W | 00H | P8 Function Selection Register |

## 9.3 P0 Port

### 9.3.1 P0 Port Description

P0 is 8-bit I/O port. P0 control registers consist of P0 data register (P0), P0 direction register (P0IO), debounce enable register (P0DB), P0 pull-up resistor selection register (P0PU) and P0 open-drain selection register (P0OD). Refer to the port function selection registers for the P0 function selection.

### 9.3.2 Register description for P0

**P0 (P0 Data Register) : 80H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P07 | P06 | P05 | P04 | P03 | P02 | P01 | P00 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P0[7:0]** I/O Data

**P0IO (P0 Direction Register) : A1H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P07IO | P06IO | P05IO | P04IO | P03IO | P02IO | P01IO | P00IO |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P0IO[7:0]** P0 Data I/O Direction.

0 Input

1 Output

NOTE) EINT0 ~ EINT7/BLNK function possible when input

**P0PU (P0 Pull-up Resistor Selection Register) : 93H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P07PU | P06PU | P05PU | P04PU | P03PU | P02PU | P01PU | P00PU |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P0PU[7:0]** Configure Pull-up Resistor of P0 Port

0 Disable

1 Enable

**P0OD (P0 Open-drain Selection Register) : 1000H(ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P07OD | P06OD | P05OD | P04OD | P03OD | P02OD | P01OD | P00OD |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P0OD[7:0]** Configure Open-drain of P0 Port

0 Push-pull output

1 Open-drain output

**P0DB (P0 Debounce Enable Register) : B5H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P07DB | P06DB | P05DB | P04DB | P03DB | P02DB | P01DB | P00DB |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P07DB** Configure Debounce of P07 Port
- 0 Disable
- 1 Enable

**P06DB** Configure Debounce of P06 Port
- 0 Disable
- 1 Enable

**P05DB** Configure Debounce of P05 Port
- 0 Disable
- 1 Enable

**P04DB** Configure Debounce of P04 Port
- 0 Disable
- 1 Enable

**P03DB** Configure Debounce of P03Port
- 0 Disable
- 1 Enable

**P02DB** Configure Debounce of P02Port
- 0 Disable
- 1 Enable

**P01DB** Configure Debounce of P01Port
- 0 Disable
- 1 Enable

**P00DB** Configure Debounce of P00Port
- 0 Disable
- 1 Enable

NOTES) 1. If a level is not detected on enabled pin three or four times in a row at the sampling clock, the signal is eliminated as noise.
2. A pulse level should be input for the duration of 3 clock or more to be actually detected as a valid edge.
3. The port debounce is automatically disabled at stop mode and recovered after stop mode release.
4. Refer to the port 1 debounce enable register (P1DB) for the debounce clock of port 0.

**P0FSR (Port 0 Function Selection Register) : 1009H(ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P0FSR7 | P0FSR6 | P0FSR5 | P0FSR4 | P0FSR3 | P0FSR2 | P0FSR1 | P0FSR0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P0FSR7** P07 Function select
- 0 I/O Port (EINT7 function possible when input)
- 1 AN7 Function

**P0FSR6** P06 Function Select
- 0 I/O Port (EINT6 function possible when input)
- 1 AN6 Function

**P0FSR5** P05 Function select
- 0 I/O Port (EINT5 function possible when input)
- 1 AN5 Function

**P0FSR4** P04 Function Select
- 0 I/O Port (EINT4 function possible when input)
- 1 AN4 Function

**P0FSR3** P03 Function select
- 0 I/O Port (EINT3 function possible when input)
- 1 AN3 Function

**P0FSR2** P02 Function Select
- 0 I/O Port (EINT2 function possible when input)
- 1 AN2 Function

**P0FSR1** P01 Function select
- 0 I/O Port (EINT1 function possible when input)
- 1 AN1 Function

**P0FSR0** P00 Function Select
- 0 I/O Port (EINT0/BLNK function possible when input)
- 1 AN0 Function

## 9.4 P1 Port

### 9.4.1 P1 Port Description

P1 is 8-bit I/O port. P1 control registers consist of P1 data register (P1), P1 direction register (P1IO), debounce enable register (P1DB), P1 pull-up resistor selection register (P1PU) and P1 open-drain selection register (P1OD) . Refer to the port function selection registers for the P1 function selection.

### 9.4.2 Register description for P1

**P1 (P1 Data Register) : 88H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P17 | P16 | P15 | P14 | P13 | P12 | P11 | P10 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P1[7:0]** I/O Data

**P1IO (P1 Direction Register) : B1H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P17IO | P16IO | P15IO | P14IO | P13IO | P12IO | P11IO | P10IO |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P1IO[7:0]** P1 Data I/O Direction

0 Input

1 Output

NOTE) EINT8/EINT9/EINT13–EINT15/EC5/EC6 function possible when input

**P1PU (P1 Pull-up Resistor Selection Register) : 94H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P17PU | P16PU | P15PU | P14PU | P13PU | P12PU | P11PU | P10PU |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P1PU[7:0]** Configure Pull-up Resistor of P1 Port

0 Disable

1 Enable

**P1OD (P1 Open-drain Selection Register) : 1001H(ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P17OD | P16OD | P15OD | P14OD | P13OD | P12OD | P11OD | P10OD |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P1OD[7:0]** Configure Open-drain of P1 Port

0 Push-pull output

1 Open-drain output

**P1DB (P1 Debounce Enable Register) : B6H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DBCLK1 | DBCLK0 | P15DB | P14DB | P13DB | P12DB | P11DB | P10DB |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**DBCLK[1:0]** Configure Debounce Clock of Port

| DBCLK1 | DBCLK0 | Description |
|---|---|---|
| 0 | 0 | fx/1 |
| 0 | 1 | fx/4 |
| 1 | 0 | fx/4096 |
| 1 | 1 | $f_{SUB}$(External sub OSC) |

**P15DB** Configure Debounce of P15 Port

| | |
|---|---|
| 0 | Disable |
| 1 | Enable |

**P14DB** Configure Debounce of P14 Port

| | |
|---|---|
| 0 | Disable |
| 1 | Enable |

**P13DB** Configure Debounce of P13 Port

| | |
|---|---|
| 0 | Disable |
| 1 | Enable |

**P12DB** Configure Debounce of P12 Port

| | |
|---|---|
| 0 | Disable |
| 1 | Enable |

**P11DB** Configure Debounce of P11Port

| | |
|---|---|
| 0 | Disable |
| 1 | Enable |

**P10DB** Configure Debounce of P10 Port

| | |
|---|---|
| 0 | Disable |
| 1 | Enable |

NOTES) 1. If a level is not detected on enabled pin three or four times in a row at the sampling clock, the signal is eliminated as noise.

2. A pulse level should be input for the duration of 3 clock or more to be actually detected as a valid edge.

3. The port debounce is automatically disabled at stop mode and recovered after stop mode release.

**P1FSR (Port 1 Function Selection Register) : 100AH(ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | P1FSR6 | P1FSR5 | P1FSR4 | P1FSR3 | P1FSR2 | P1FSR1 | P1FSR0 |
| – | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P1FSR6** P17 Function select

0 I/O Port

1 XOUT Function

**P1FSR5** P16 Function Select

0 I/O Port

1 XIN Function

**P1FSR4** P13 Function select

0 I/O Port (EINT16 function possible when input)

1 T6O/PWM6O Function

**P1FSR3** P12 Function Select

0 I/O Port (EINT15 function possible when input)

1 T5O/PWM5O Function

**P1FSR2** P11 Function select

0 I/O Port (EINT14 function possible when input)

1 T4O/PWM4O Function

**P1FSR[1:0]** P10 Function Select

| P1FSR1 | P1FSR0 | Description |
|---|---|---|
| 0 | 0 | I/O Port(EINT13 function possible when input) |
| 0 | 1 | T3O/PWM3O Function |
| 1 | 0 | EXTSP2 Function |
| 1 | 1 | Not used |

## 9.5 P2 Port

### 9.5.1 P2 Port Description

P2 is 8-bit I/O port. P2 control registers consist of P2 data register (P2), P2 direction register (P2IO), P2 pull-up resistor selection register (P2PU) and P2 open-drain selection register (P2OD). Refer to the port function selection registers for the P2 function selection.

### 9.5.2 Register description for P2

**P2 (P2 Data Register) : 90H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P27 | P26 | P25 | P24 | P23 | P22 | P21 | P20 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P2[7:0]** I/O Data

**P2IO (P2 Direction Register) : B9H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P27IO | P26IO | P25IO | P24IO | P23IO | P22IO | P21IO | P20IO |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P2IO[7:0]** P2 Data I/O Direction

0 Input

1 Output

NOTE) SS3/EC3–EC4/TRIG/EXTSP0/EXTSP1 function possible when input

**P2PU (P2 Pull-up Resistor Selection Register) : 99H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P27PU | P26PU | P25PU | P24PU | P23PU | P22PU | P21PU | P20PU |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P2PU[7:0]** Configure Pull-up Resistor of P2 Port

0 Disable

1 Enable

**P2OD (P2 Open-drain Selection Register) : 1002H(ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P27OD | P26OD | P25OD | P24OD | P23OD | P22OD | P21OD | P20OD |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P2OD[7:0]** Configure Open-drain of P2 Port

0 Push-pull output

1 Open-drain output

**P2FSRH (Port 2 Function Selection High Register) : 100DH(ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | P2FSRH4 | P2FSRH3 | P2FSRH2 | P2FSRH1 | P2FSRH0 |
| – | – | – | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P2FSRH4** P27 Function select

0 I/O Port (EXTSP1 function possible when input)

1 COM7/SEG39 Function

**P2FSRH3** P26 Function Select

0 I/O Port (EXTSP0 function possible when input)

1 COM6/SEG38 Function

**P2FSRH2** P25 Function select

0 I/O Port(TRIG function possible when input)

1 COM5/SEG37 Function

**P2FSRH[1:0]** P24 Function Select

| P2FSRH1 | P2FSRH0 | Description |
|---|---|---|
| 0 | 0 | I/O Port |
| 0 | 1 | COM4/SEG36 Function |
| 1 | 0 | PWMOUT Function |
| 1 | 1 | Not used |

NOTE) The P24-P27 is automatically configured as common or segment signal according to the duty in the LCDCRL register when the pin is selected as the sub-function for common/segment.

**P2FSRL (Port 2 Function Selection Low Register) : 100CH(ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SPI3_3V | P2FSRL6 | P2FSRL5 | P2FSRL4 | P2FSRL3 | P2FSRL2 | P2FSRL1 | P2FSRL0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**SPI3_3V** SPI3 Input Signal (MOSI3 or MISO3) 3V Interface Selection

0 Normal voltage interface mode for SPI3

1 3V interface input mode for SPI3 (When VDD >= 3V)

NOTE) When this bit is '1b', Serial input ( MOSI2 in Slave or MISO2 In Master) can receive the signal of 3V level.

**P2FSRL[6:5]** P23 Function Select

| P2FSRL6 | P2FSRL5 | Description |
|---|---|---|
| 0 | 0 | I/O Port |
| 0 | 1 | SEG35 Function |
| 1 | 0 | T3O/PWM3O Function |
| 1 | 1 | EXTSP2 Function |

**P2FSRL4** P22 Function Select

0 I/O Port (SS3 function possible when input)

1 SEG34 Function

**P2FSRL[3:2]** P21 Function Select

| P2FSRL3 | P2FSRL2 | Description |
|---|---|---|
| 0 | 0 | I/O Port |
| 0 | 1 | SEG33 Function |
| 1 | 0 | SCK3 Function |
| 1 | 1 | Not used |

**P2FSRL[1:0]** P20 Function Select

| P2FSRL1 | P1FSRL0 | Description |
|---|---|---|
| 0 | 0 | I/O Port |
| 0 | 1 | SEG32 Function |
| 1 | 0 | MOSI3 Function |
| 1 | 1 | Not used |

## 9.6 P3 Port

### 9.6.1 P3 Port Description

P3 is 8-bit I/O port. P3 control registers consist of P3 data register (P3), P3 direction register (P3IO), P3 pull-up resistor selection register (P3PU) and P3 open-drain selection register (P3OD). Refer to the port function selection registers for the P3 function selection.

### 9.6.2 Register description for P3

**P3 (P3 Data Register) : 98H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P37 | P36 | P35 | P34 | P33 | P32 | P31 | P30 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P3[7:0]** I/O Data

**P3IO (P3 Direction Register) : C1H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P37IO | P36IO | P35IO | P34IO | P33IO | P32IO | P31IO | P30IO |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P3IO[7:0]** P3 Data I/O Direction

0 Input

1 Output

NOTE) SS2 function possible when input

**P3PU (P3 Pull-up Resistor Selection Register) : 9AH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P37PU | P36PU | P35PU | P34PU | P33PU | P32PU | P31PU | P30PU |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P3PU[7:0]** Configure Pull-up Resistor of P3 Port

0 Disable

1 Enable

**P3OD (P3 Open-drain Selection Register) : 1003H(ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P37OD | P36OD | P35OD | P34OD | P33OD | P32OD | P31OD | P30OD |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P3OD[7:0]** Configure Open-drain of P3 Port

0 Push-pull output

1 Open-drain output

**P3FSRH (Port 3 Function Selection High Register) : 100FH(ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SPI2_3V | P3FSRH6 | P3FSRH5 | P3FSRH4 | P3FSRH3 | P3FSRH2 | P3FSRH1 | P3FSRH0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**SPI2_3V** SPI2 Input Signal (MOSI2 or MISO2) 3V Interface Selection

| | |
|---|---|
| 0 | Normal voltage interface mode for SPI2 |
| 1 | 3V interface input mode for SPI2 (When VDD >= 3V) |

**P3FSRH[6:5]** P37 Function Select

| P3FSRH6 | P3FSRH5 | Description |
|---|---|---|
| 0 | 0 | I/O Port |
| 0 | 1 | SEG31 Function |
| 1 | 0 | MISO3 Function |
| 1 | 1 | Not used |

**P3FSRH4** P36 Function Select

| | |
|---|---|
| 0 | I/O Port (SS2 function possible when input) |
| 1 | SEG30 Function |

**P3FSRH[3:2]** P35 Function Select

| P3FSRH3 | P3FSRH2 | Description |
|---|---|---|
| 0 | 0 | I/O Port |
| 0 | 1 | SEG29 Function |
| 1 | 0 | SCK2 Function |
| 1 | 1 | Not used |

**P3FSRH[1:0]** P34 Function Select

| P3FSRH1 | P3FSRH0 | Description |
|---|---|---|
| 0 | 0 | I/O Port |
| 0 | 1 | SEG28 Function |
| 1 | 0 | MOSI2 Function |
| 1 | 1 | Not used |

**P3FSRL (Port 3 Function Selection Low Register) : 100EH(ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P3FSRL7 | P3FSRL6 | P3FSRL5 | P3FSRL4 | P3FSRL3 | P3FSRL2 | P3FSRL1 | P3FSRL0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P3FSRL[7:6]** P33 Function Select

| P3FSRL7 | P3FSRL6 | Description |
|---|---|---|
| 0 | 0 | I/O Port |
| 0 | 1 | SEG27 Function |
| 1 | 0 | MISO2 Function |
| 1 | 1 | Not used |

**P3FSRL[5:4]** P32 Function Select

| P3FSRL5 | P3FSRL4 | Description |
|---|---|---|
| 0 | 0 | I/O Port |
| 0 | 1 | SEG26 Function |
| 1 | 0 | RXD0/SCL0/MISO0 Function |
| 1 | 1 | Not used |

**P3FSRL[3:2]** P31 Function Select

| P3FSRL3 | P3FSRL2 | Description |
|---|---|---|
| 0 | 0 | I/O Port |
| 0 | 1 | SEG25 Function |
| 1 | 0 | TXD0/SDA0/MOSI0 Function |
| 1 | 1 | Not used |

**P3FSRL[1:0]** P30 Function Select

| P3FSRL1 | P3FSRL0 | Description |
|---|---|---|
| 0 | 0 | I/O Port |
| 0 | 1 | SEG24 Function |
| 1 | 0 | SCK Function |
| 1 | 1 | Not used |

## 9.7 P4 Port

### 9.7.1 P4 Port Description

P4 is 8-bit I/O port. P4 control registers consist of P4 data register (P4), P4 direction register (P4IO), P4 pull-up resistor selection register (P4PU) and P4 open-drain selection register (P4OD). Refer to the port function selection registers for the P4 function selection.

### 9.7.2 Register description for P4

**P4 (P4 Data Register) : A0H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P47 | P46 | P45 | P44 | P43 | P42 | P41 | P40 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P4[7:0]** I/O Data

**P4IO (P4 Direction Register) : C9H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P47IO | P46IO | P45IO | P44IO | P43IO | P42IO | P41IO | P40IO |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P4IO[7:0]** P4 Data I/O Direction

0 Input

1 Output

NOTE) EINT10–EINT12/EC0–EC2/SS0 function possible when input

**P4PU (P4 Pull-up Resistor Selection Register) : 9BH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P47PU | P46PU | P45PU | P44PU | P43PU | P42PU | P41PU | P40PU |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P4PU[7:0]** Configure Pull-up Resistor of P4 Port

0 Disable

1 Enable

**P4OD (P4 Open-drain Selection Register) : 1004H(ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P47OD | P46OD | P45OD | P44OD | P43OD | P42OD | P41OD | P40OD |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P4OD[7:0]** Configure Open-drain of P4 Port

0 Push-pull output

1 Open-drain output

**P46DB (P4/P6 Debounce Enable Register) : B7H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | P62DB | P61DB | P45DB | P44DB | P43DB |
| – | – | – | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

| | | |
|---|---|---|
| **P62DB** | Configure Debounce of P62 Port | |
| | 0 | Disable |
| | 1 | Enable |
| **P61DB** | Configure Debounce of P61 Port | |
| | 0 | Disable |
| | 1 | Enable |
| **P45DB** | Configure Debounce of P45 Port | |
| | 0 | Disable |
| | 1 | Enable |
| **P44DB** | Configure Debounce of P44 Port | |
| | 0 | Disable |
| | 1 | Enable |
| **P43DB** | Configure Debounce of P43Port | |
| | 0 | Disable |
| | 1 | Enable |

NOTES) 1. If a level is not detected on enabled pin three or four times in a row at the sampling clock, the signal is eliminated as noise.

2. A pulse level should be input for the duration of 3 clock or more to be actually detected as a valid edge.
3. The port debounce is automatically disabled at stop mode and recovered after stop mode release.
4. Refer to the port 1 debounce enable register (P1DB) for the debounce clock of port 4 and 6.

**P4FSRH (Port 4 Function Selection High Register) : 1011H(ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | P4FSRH6 | P4FSRH5 | P4FSRH4 | P4FSRH3 | P4FSRH2 | P4FSRH1 | P4FSRH0 |
| – | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P4FSRH6** P47 Function Select

0 I/O Port (SS0 function possible when input)

1 SEG23 Function

**P4FSRH[5:4]** P46 Function Select

| P4FSRH5 | P4FSRH4 | Description |
|---|---|---|
| 0 | 0 | I/O Port |
| 0 | 1 | SEG22 Function |
| 1 | 0 | BUZO Function |
| 1 | 1 | Not used |

**P4FSRH[3:2]** P45 Function Select

| P4FSRH3 | P4FSRH2 | Description |
|---|---|---|
| 0 | 0 | I/O Port (EINT12 function possible when input) |
| 0 | 1 | SEG21 Function |
| 1 | 0 | T2O/PWM2O Function |
| 1 | 1 | Not used |

**P4FSRH[1:0]** P44 Function Select

| P4FSRH1 | P4FSRH0 | Description |
|---|---|---|
| 0 | 0 | I/O Port (EINT11 function possible when input) |
| 0 | 1 | SEG20 Function |
| 1 | 0 | T1O/PWM1O Function |
| 1 | 1 | Not used |

**P4FSRL (Port 4 Function Selection Low Register) : 1010H(ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | P4FSRL4 | P4FSRL3 | P4FSRL2 | P4FSRL1 | P4FSRL0 |
| – | – | – | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P4FSRL[4:3]** P43 Function Select

| P4FSRL4 | P4FSRL3 | Description |
|---|---|---|
| 0 | 0 | I/O Port (EINT10 function possible when input) |
| 0 | 1 | SEG19 Function |
| 1 | 0 | T0O/PWM0O Function |
| 1 | 1 | Not used |

**P4FSRL2** P42 Function Select

| | |
|---|---|
| 0 | I/O Port (EC2 function possible when input) |
| 1 | SEG18 Function |

**P4FSRL1** P41 Function Select

| | |
|---|---|
| 0 | I/O Port (EC1 function possible when input) |
| 1 | SEG17 Function |

**P4FSRL0** P40 Function Select

| | |
|---|---|
| 0 | I/O Port (EC0 function possible when input) |
| 1 | SEG16 Function |

## 9.8 P5 Port

### 9.8.1 P5 Port Description

P5 is 8-bit I/O port. P5 control registers consist of P5 data register (P5), P5 direction register (P5IO), P5 pull-up resistor selection register (P5PU) and P5 open-drain selection register (P5OD). Refer to the port function selection registers for the P5 function selection.

### 9.8.2 Register description for P5

**P5 (P5 Data Register) : B0H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P57 | P56 | P55 | P54 | P53 | P52 | P51 | P50 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P5[7:0]** I/O Data

**P5IO (P5 Direction Register) : D1H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P57IO | P56IO | P55IO | P54IO | P53IO | P52IO | P51IO | P50IO |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P5IO[7:0]** P5 Data I/O Direction

0 Input

1 Output

NOTE) RXD3/RXD4/SS1 function possible when input

**P5PU (P5 Pull-up Resistor Selection Register) : 9CH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P57PU | P56PU | P55PU | P54PU | P53PU | P52PU | P51PU | P50PU |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P5PU[7:0]** Configure Pull-up Resistor of P5 Port

0 Disable

1 Enable

**P5OD (P5 Open-drain Selection Register) : 1005H(ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P57OD | P56OD | P55OD | P54OD | P53OD | P52OD | P51OD | P50OD |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P5OD[7:0]** Configure Open-drain of P5 Port

0 Push-pull output

1 Open-drain output

**P5FSRH (Port 5 Function Selection High Register) : 1013H(ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | P5FSRH6 | P5FSRH5 | P5FSRH4 | P5FSRH3 | P5FSRH2 | P5FSRH1 | P5FSRH0 |
| – | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P5FSRH[6:5]** P57 Function Select

| P5FSRH6 | P5FSRH5 | Description |
|---|---|---|
| 0 | 0 | I/O Port |
| 0 | 1 | SEG15 Function |
| 1 | 0 | RXD1/SCL1/MISO1 Function |
| 1 | 1 | Not used |

**P5FSRH[4:3]** P56 Function Select

| P5FSRH4 | P5FSRH3 | Description |
|---|---|---|
| 0 | 0 | I/O Port |
| 0 | 1 | SEG14 Function |
| 1 | 0 | TXD1/SDA1/MOSI1 Function |
| 1 | 1 | Not used |

**P5FSRH[2:1]** P55 Function Select

| P5FSRH2 | P5FSRH1 | Description |
|---|---|---|
| 0 | 0 | I/O Port |
| 0 | 1 | SEG13 Function |
| 1 | 0 | SCK1 Function |
| 1 | 1 | Not used |

**P5FSRH0** P54 Function Select

| | |
|---|---|
| 0 | I/O Port (SS1 function possible when input) |
| 1 | SEG12 Function |

**P5FSRL (Port 5 Function Selection Low Register) : 1012H(ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | P5FSRL5 | P5FSRL4 | P5FSRL3 | P5FSRL2 | P5FSRL1 | P5FSRL0 |
| – | – | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P5FSRL[5:4]** P53 Function Select

| P5FSRL5 | P5FSRL4 | Description |
|---|---|---|
| 0 | 0 | I/O Port |
| 0 | 1 | SEG11 Function |
| 1 | 0 | TXD3 Function |
| 1 | 1 | Not used |

**P5FSRL3** P52 Function Select

| | |
|---|---|
| 0 | I/O Port (RXD3 function possible when input) |
| 1 | SEG10 Function |

**P5FSRL[2:1]** P51 Function Select

| P5FSRL2 | P5FSRL1 | Description |
|---|---|---|
| 0 | 0 | I/O Port |
| 0 | 1 | SEG9 Function |
| 1 | 0 | TXD4 Function |
| 1 | 1 | Not used |

**P5FSRL0** P50 Function Select

| | |
|---|---|
| 0 | I/O Port (RXD4 function possible when input) |
| 1 | SEG8 Function |

## 9.9 P6 Port

### 9.9.1 P6 Port Description

P6 is 7-bit I/O port. P6 control registers consist of P6 data register (P6), P6 direction register (P6IO), P6 pull-up resistor selection register (P6PU) and P6 open-drain selection register (P6OD). Refer to the port function selection registers for the P6 function selection.

### 9.9.2 Register description for P6

**P6 (P6 Data Register) : C0H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | P66 | P65 | P64 | P63 | P62 | P61 | P60 |
| – | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P6[6:0]** I/O Data

**P6IO (P6 Direction Register) : D9H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | P66IO | P65IO | P64IO | P63IO | P62IO | P61IO | P60IO |
| – | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P6IO[6:0]** P6 Data I/O Direction

0 Input

1 Output

NOTE) EINT17/EINT18/EC7/RXD2 function possible when input

**P6PU (P6 Pull-up Resistor Selection Register) : 9DH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | P66PU | P65PU | P64PU | P63PU | P62PU | P61PU | P60PU |
| – | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P6PU[6:0]** Configure Pull-up Resistor of P6 Port

0 Disable

1 Enable

**P6OD (P6 Open-drain Selection Register) : 1006H(ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | P66OD | P65OD | P64OD | P63OD | P62OD | P61OD | P60OD |
| – | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P6OD[6:0]** Configure Open-drain of P6 Port

0 Push-pull output

1 Open-drain output

**P6FSRH (Port 6 Function Selection High Register) : 1015H(ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | P6FSRH2 | P6FSRH1 | P6FSRH0 |
| – | – | – | – | – | R/W | R/W | R/W |

Initial value : 00H

**P6FSRH2** P65 Function Select

| | |
|---|---|
| 0 | I/OPort (RXD2 function possible when input) |
| 1 | AN11 Function |

**P6FSRH[1:0]** P64 Function Select

| P6FSRH1 | P6FSRH0 | Description |
|---|---|---|
| 0 | 0 | I/OPort |
| 0 | 1 | AN10 Function |
| 1 | 0 | TXD2 Function |
| 1 | 1 | Not used |

NOTE) Refer to the configure option for the P66/RESETB.

**P6FSRL (Port 6 Function Selection Low Register) : 1014H(ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P6FSRL7 | P6FSRL6 | P6FSRL5 | P6FSRL4 | P6FSRL3 | P6FSRL2 | P6FSRL1 | P6FSRL0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P6FSRL[7:6]** P63 Function Select

| P6FSRL7 | P6FSRL6 | Description |
|---|---|---|
| 0 | 0 | I/O Port (EC7 function possible when input) |
| 0 | 1 | AN9 Function |
| 1 | 0 | LVIREF Function |
| 1 | 1 | Not used |

**P6FSRL[5:4]** P62 Function Select

| P6FSRL5 | P6FSRL4 | Description |
|---|---|---|
| 0 | 0 | I/O Port (EINT17 function possible when input) |
| 0 | 1 | AN8 Function |
| 1 | 0 | T7O Function |
| 1 | 1 | Not used |

**P6FSRL[3:2]** P61 Function Select

| P6FSRL3 | P6FSRL2 | Description |
|---|---|---|
| 0 | 0 | I/O Port (EINT18 function possible when input) |
| 0 | 1 | CAPL Function |
| 1 | 0 | T8O/PWM8AA Function |
| 1 | 1 | Not used |

**P6FSRL[1:0]** P60 Function Select

| P6FSRL1 | P6FSRL0 | Description |
|---|---|---|
| 0 | 0 | I/O Port |
| 0 | 1 | CAPH Function |
| 1 | 0 | PWM8AB Function |
| 1 | 1 | Not used |

## 9.10 P7 Port

### 9.10.1 P7 Port Description

P7 is 8-bit I/O port. P7 control registers consist of P7 data register (P7), P7 direction register (P7IO), P7 pull-up resistor selection register (P7PU) and P7 open-drain selection register (P7OD). Refer to the port function selection registers for the P7 function selection.

### 9.10.2 Register description for P7

**P7 (P7 Data Register) : 91H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P77 | P76 | P75 | P74 | P73 | P72 | P71 | P70 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P7[7:0]** I/O Data

**P7IO (P7 Direction Register) : E1H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P77IO | P76IO | P75IO | P74IO | P73IO | P72IO | P71IO | P70IO |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P7IO[7:0]** P7 Data I/O Direction

0 Input

1 Output

**P7PU (P7 Pull-up Resistor Selection Register) : 9EH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P77PU | P76PU | P75PU | P74PU | P73PU | P72PU | P71PU | P70PU |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P7PU[7:0]** Configure Pull-up Resistor of P7 Port

0 Disable

1 Enable

**P7OD (P7 Open-drain Selection Register) : 1007H(ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P77OD | P76OD | P75OD | P74OD | P73OD | P72OD | P71OD | P70OD |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P7OD[7:0]** Configure Open-drain of P7 Port

0 Push-pull output

1 Open-drain output

**P7FSRH (Port 7 Function Selection High Register) : 1017H(ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | P7FSRH3 | P7FSRH2 | P7FSRH1 | P7FSRH0 |
| – | – | – | – | R/W | R/W | R/W | R/W |

Initial value : 00H

**P7FSRH3** P77 Function select
- 0 I/O Port
- 1 COM0 Function

**P7FSRH2** P76 Function Select
- 0 I/O Port
- 1 COM1 Function

**P7FSRH1** P75 Function select
- 0 I/O Port
- 1 COM2 Function

**P7FSRH0** P74 Function select
- 0 I/O Port
- 1 COM3 Function

**P7FSRL (Port 7 Function Selection Low Register) : 1016H(ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P7FSRL7 | P7FSRL6 | P7FSRL5 | P7FSRL4 | P7FSRL3 | P7FSRL2 | P7FSRL1 | P7FSRL0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P7FSRL[7:6]** P73 Function Select

| P7FSRL7 | P7FSRL6 | Description |
|---|---|---|
| 0 | 0 | I/O Port |
| 0 | 1 | VLC0 Function |
| 1 | 0 | PWM8CB Function |
| 1 | 1 | Not used |

**P7FSRL[5:4]** P72 Function Select

| P7FSRL5 | P7FSRL4 | Description |
|---|---|---|
| 0 | 0 | I/O Port |
| 0 | 1 | VLC1 Function |
| 1 | 0 | PWM8CA Function |
| 1 | 1 | Not used |

**P7FSRL[3:2]** P71 Function Select

| P7FSRL3 | P7FSRL2 | Description |
|---|---|---|
| 0 | 0 | I/O Port |
| 0 | 1 | VLC2 Function |
| 1 | 0 | PWM8BB Function |
| 1 | 1 | Not used |

**P7FSRL[1:0]** P70 Function Select

| P7FSRL1 | P7FSRL0 | Description |
|---|---|---|
| 0 | 0 | I/O Port |
| 0 | 1 | VLC3 Function |
| 1 | 0 | PWM8BA Function |
| 1 | 1 | Not used |

## 9.11 P8 Port

### 9.11.1 P8 Port Description

P8 is 8-bit I/O port. P8 control registers consist of P8 data register (P8), P8 direction register (P8IO), P8 pull-up resistor selection register (P8PU) and P8 open-drain selection register (P8OD). Refer to the port function selection registers for the P8 function selection.

### 9.11.2 Register description for P8

**P8 (P8 Data Register) : 92H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P87 | P86 | P85 | P84 | P83 | P82 | P81 | P80 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P8[7:0]** I/O Data

**P8IO (P8 Direction Register) : E9H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P87IO | P86IO | P85IO | P84IO | P83IO | P82IO | P81IO | P80IO |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P8IO[7:0]** P8 Data I/O Direction

0 Input

1 Output

**P8PU (P8 Pull-up Resistor Selection Register) : 9FH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P87PU | P86PU | P85PU | P84PU | P83PU | P82PU | P81PU | P80PU |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P8PU[7:0]** Configure Pull-up Resistor of P8 Port

0 Disable

1 Enable

**P8OD (P8 Open-drain Selection Register) : 1008H(ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P87OD | P86OD | P85OD | P84OD | P83OD | P82OD | P81OD | P80OD |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P8OD[7:0]** Configure Open-drain of P8 Port

0 Push-pull output

1 Open-drain output

**P8FSR (Port 8 Function Selection Register) : 100BH(ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| P8FSR7 | P8FSR6 | P8FSR5 | P8FSR4 | P8FSR3 | P8FSR2 | P8FSR1 | P8FSR0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**P8FSR7** P87 Function select
- 0 I/O Port
- 1 SEG7 Function

**P8FSR6** P86 Function Select
- 0 I/O Port
- 1 SEG6 Function

**P8FSR5** P85 Function select
- 0 I/O Port
- 1 SEG5 Function

**P8FSR4** P84 Function Select
- 0 I/O Port
- 1 SEG4 Function

**P8FSR3** P83 Function select
- 0 I/O Port
- 1 SEG3 Function

**P8FSR2** P82 Function Select
- 0 I/O Port
- 1 SEG2 Function

**P8FSR1** P81 Function select
- 0 I/O Port
- 1 SEG1 Function

**P8FSR0** P80 Function Select
- 0 I/O Port
- 1 SEG0 Function

# 10. Interrupt Controller

## 10.1 Overview

The MC96F7864 supports up to 24 interrupt sources. The interrupts have separate enable register bits associated with them, allowing software control. They can also have four levels of priority assigned to them. The non-maskable interrupt source is always enabled with a higher priority than any other interrupt source, and is not controllable by software. The interrupt controller has following features:

- Receive the request from 24 interrupt source
- 6 group priority
- 4 priority levels
- Multi Interrupt possibility
- If the requests of different priority levels are received simultaneously, the request of higher priority level is served first.
- Each interrupt source can be controlled by EA bit and each IEx bit
- Interrupt latency: 3~9 machine cycles in single interrupt system

The non-maskable interrupt is always enabled. The maskable interrupts are enabled through four pair of interrupt enable registers (IE, IE1, IE2, IE3). Each bit of IE, IE1, IE2, IE3 register individually enables/disables the corresponding interrupt source. Overall control is provided by bit 7 of IE (EA). When EA is set to ‘0’, all interrupts are disabled: when EA is set to ‘1’, interrupts are individually enabled or disabled through the other bits of the interrupt enable registers. The EA bit is always cleared to ‘0’ jumping to an interrupt service vector and set to ‘1’ executing the [RETI] instruction. The MC96F7864 supports a four-level priority scheme. Each maskable interrupt is individually assigned to one of four priority levels according to IP and IP1.

Default interrupt mode is level-trigger mode basically, but if needed, it is possible to change to edge-trigger mode. Table 10-1 shows the Interrupt Group Priority Level that is available for sharing interrupt priority. Priority of a group is set by two bits of interrupt priority registers (one bit from IP, another one from IP1). Interrupt service routine serves higher priority interrupt first. If two requests of different priority levels are received simultaneously, the request of higher priority level is served prior to the lower one.

**Table 10-1 Interrupt Group Priority Level**

| Interrupt Group | Highest → Lowest | | | | |
|---|---|---|---|---|---|
| 0 (Bit0) | Interrupt 0 | Interrupt 6 | Interrupt 12 | Interrupt 18 | Highest |
| 1 (Bit1) | Interrupt 1 | Interrupt 7 | Interrupt 13 | Interrupt 19 | |
| 2 (Bit2) | Interrupt 2 | Interrupt 8 | Interrupt 14 | Interrupt 20 | |
| 3 (Bit3) | Interrupt 3 | Interrupt 9 | Interrupt 15 | Interrupt 21 | |
| 4 (Bit4) | Interrupt 4 | Interrupt 10 | Interrupt 16 | Interrupt 22 | |
| 5 (Bit5) | Interrupt 5 | Interrupt 11 | Interrupt 17 | Interrupt 23 | Lowest |

## 10.2 External Interrupt

The external interrupt on INT5, INT16, and INT23 pins receive various interrupt request depending on the external interrupt polarity 0 high/low register (EIPOL0H/L), external interrupt polarity 1 register (EIPOL1), and external interrupt polarity 2 high/low register (EIPOL2H/L) as shown in Figure 10.1. Also each external interrupt source has enable/disable bits. The external interrupt flag 0 register (EIFLAG0), external interrupt flag 1 register (EIFLAG1), and external interrupt flag 2 register (EIFLAG2) provides the status of external interrupts.

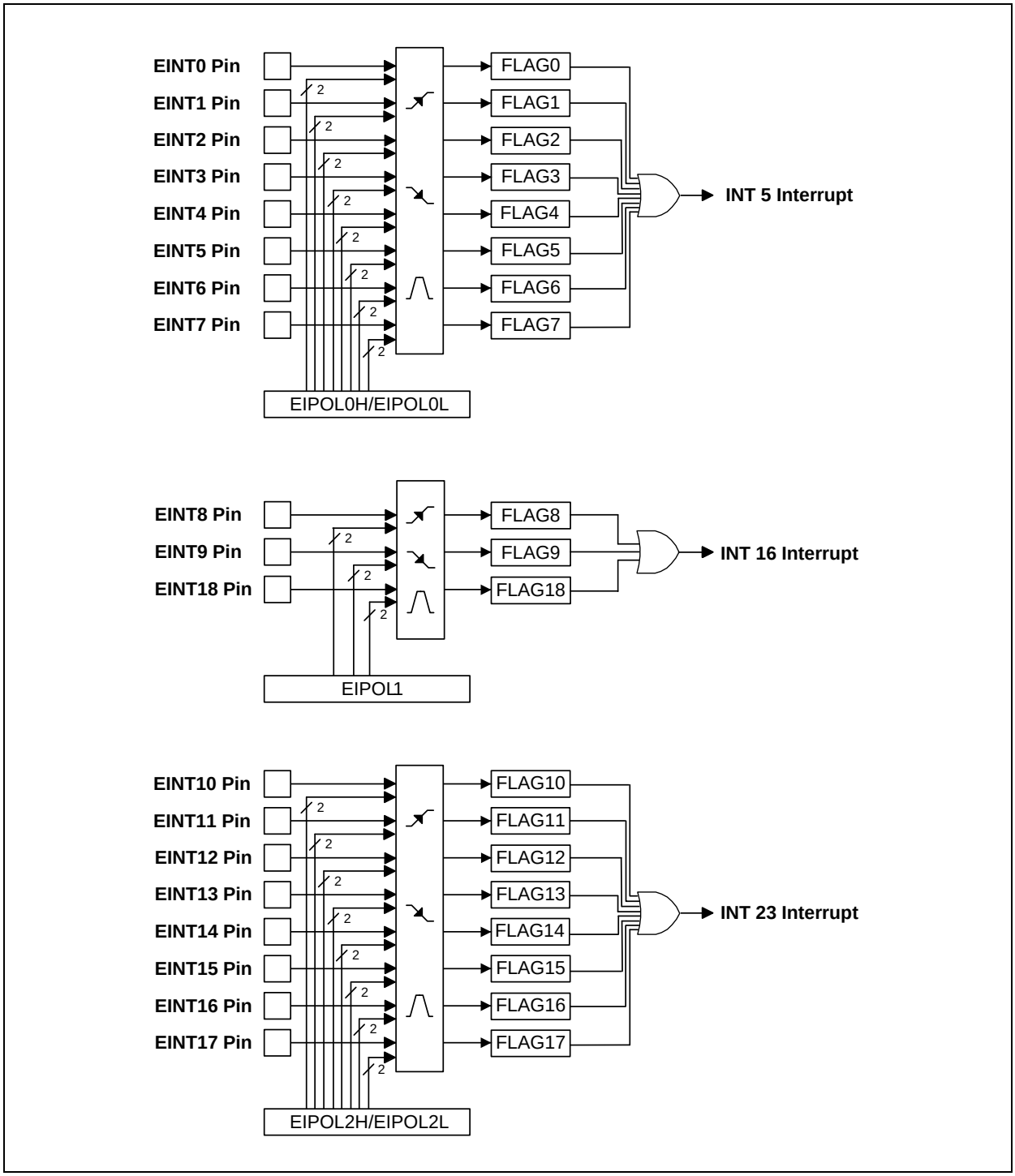



**Figure 10.1 External Interrupt Description**

## 10.3 Block Diagram

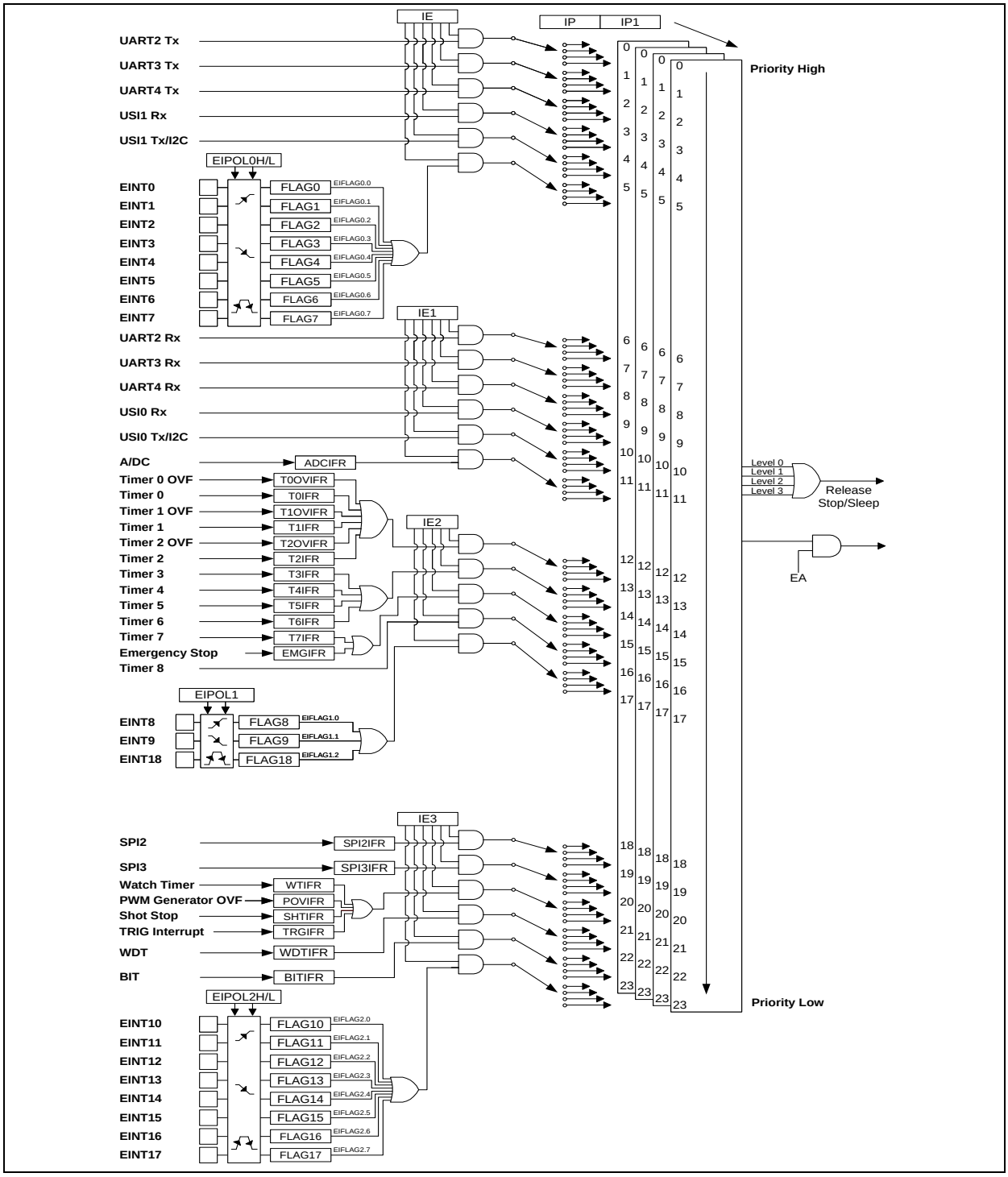



**Figure 10.2 Block Diagram of Interrupt**

NOTES) 1. The release signal for stop/idle mode may be generated by all interrupt sources which are enabled without reference to the priority level.

2. An interrupt request is delayed while data are written to IE, IE1, IE2, IE3, IP, IP1, and PCON register.

## 10.4 Interrupt Vector Table

The interrupt controller supports 24 interrupt sources as shown in the Table 10-2. When interrupt is served, long call instruction (LCALL) is executed and program counter jumps to the vector address. All interrupt requests have their own priority order.

**Table 10-2 Interrupt Vector Address Table**

| Interrupt Source | Symbol | Interrupt Enable Bit | Priority | Mask | Vector Address |
|---|---|---|---|---|---|
| Hardware RESET | RESETB | 0 0 | 0 | Non-Maskable | 0000H |
| UART2 Tx Interrupt | INT0 | IE.0 | 1 | Maskable | 0003H |
| UART3 Tx Interrupt | INT1 | IE.1 | 2 | Maskable | 000BH |
| UART4 Tx Interrupt | INT2 | IE.2 | 3 | Maskable | 0013H |
| USI1 Rx Interrupt | INT3 | IE.3 | 4 | Maskable | 001BH |
| USI1 Tx/I2C Interrupt | INT4 | IE.4 | 5 | Maskable | 0023H |
| External Interrupt 0 – 7 | INT5 | IE.5 | 6 | Maskable | 002BH |
| UART2 Rx Interrupt | INT6 | IE1.0 | 7 | Maskable | 0033H |
| UART3 Rx Interrupt | INT7 | IE1.1 | 8 | Maskable | 003BH |
| UART4 Rx Interrupt | INT8 | IE1.2 | 9 | Maskable | 0043H |
| USI0 Rx Interrupt | INT9 | IE1.3 | 10 | Maskable | 004BH |
| USI0 Tx/I2C Interrupt | INT10 | IE1.4 | 11 | Maskable | 0053H |
| ADC Interrupt | INT11 | IE1.5 | 12 | Maskable | 005BH |
| T0/1/2 OVF/Match Interrupt | INT12 | IE2.0 | 13 | Maskable | 0063H |
| T3/4/5/6 Match Interrupt | INT13 | IE2.1 | 14 | Maskable | 006BH |
| T7 Match Interrupt<br>Emergency Stop Interrupt | INT14 | IE2.2 | 15 | Maskable | 0073H |
| T8 Interrupt | INT15 | IE2.3 | 16 | Maskable | 007BH |
| External Interrupt 8/9/18 | INT16 | IE2.4 | 17 | Maskable | 0083H |
| - | INT17 | IE2.5 | 18 | Maskable | 008BH |
| SPI2 Interrupt | INT18 | IE3.0 | 19 | Maskable | 0093H |
| SPI3 Interrupt | INT19 | IE3.1 | 20 | Maskable | 009BH |
| WT Interrupt<br>PWM Generator Overflow Interrupt<br>Shot Stop Interrupt<br>TRIG Interrupt | INT20 | IE3.2 | 21 | Maskable | 00A3H |
| WDT Interrupt | INT21 | IE3.3 | 22 | Maskable | 00ABH |
| BIT Interrupt | INT22 | IE3.4 | 23 | Maskable | 00B3H |
| External Interrupt 10 – 17 | INT23 | IE3.5 | 24 | Maskable | 00BBH |

For maskable interrupt execution, EA bit must set ‘1’ and specific interrupt must be enabled by writing ‘1’ to associated bit in the IEx. If an interrupt request is received, the specific interrupt request flag is set to ‘1’. And it remains ‘1’ until CPU accepts interrupt. If the interrupt is served, the interrupt request flag will be cleared automatically.

## 10.5 Interrupt Sequence

An interrupt request is held until the interrupt is accepted or the interrupt latch is cleared to '0' by a reset or an instruction. Interrupt acceptance always generates at last cycle of the instruction. So instead of fetching the current instruction, CPU executes internally LCALL instruction and saves the PC at stack. For the interrupt service routine, the interrupt controller gives the address of LJMP instruction to CPU. Since the end of the execution of current instruction, it needs 3~9 machine cycles to go to the interrupt service routine. The interrupt service task is terminated by the interrupt return instruction [RETI]. Once an interrupt request is generated, the following process is performed.

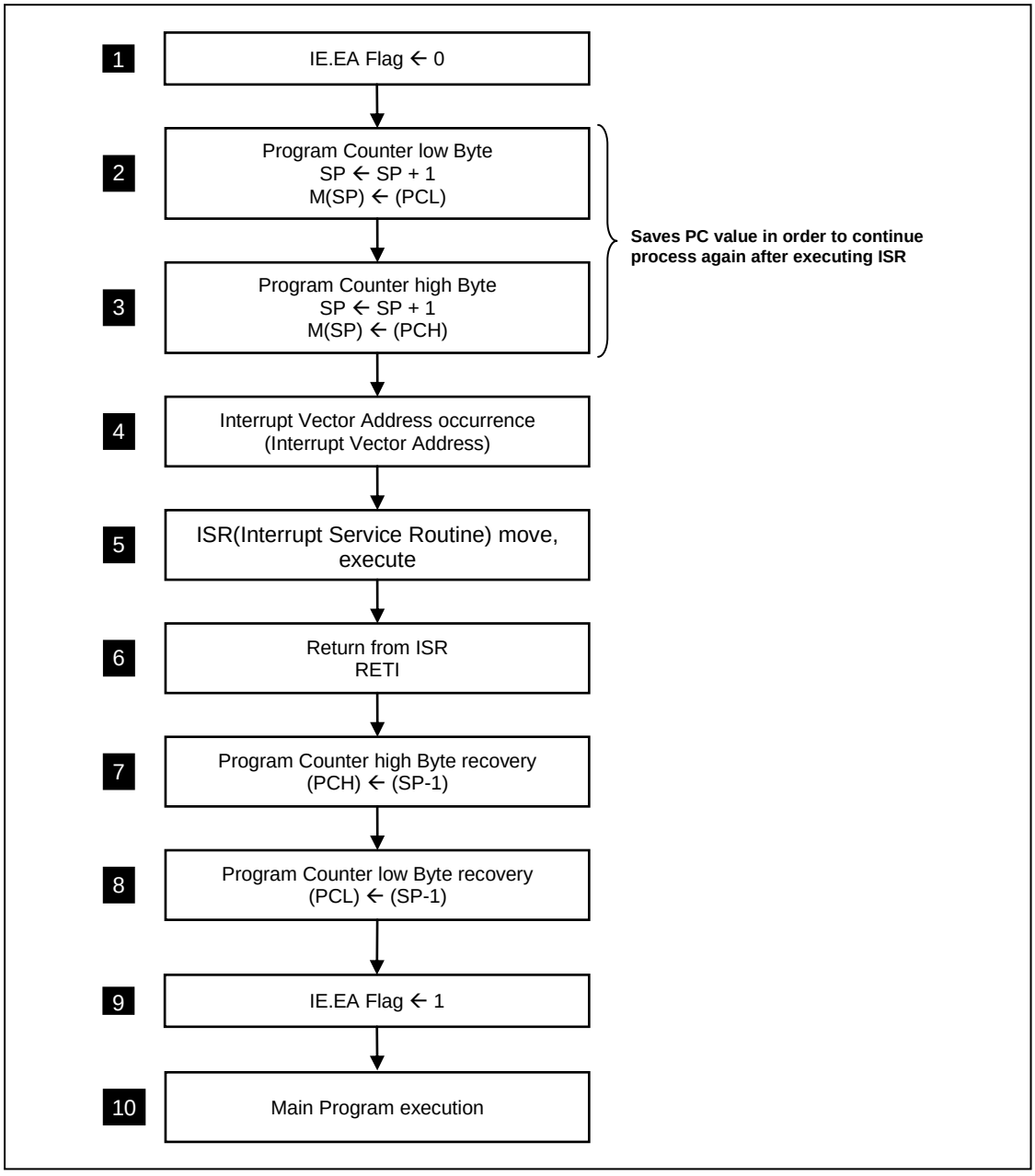



**Figure 10.3 Interrupt Vector Address Table**

## 10.6 Effective Timing after Controlling Interrupt Bit

Case a) Control Interrupt Enable Register (IE, IE1, IE2, IE3)

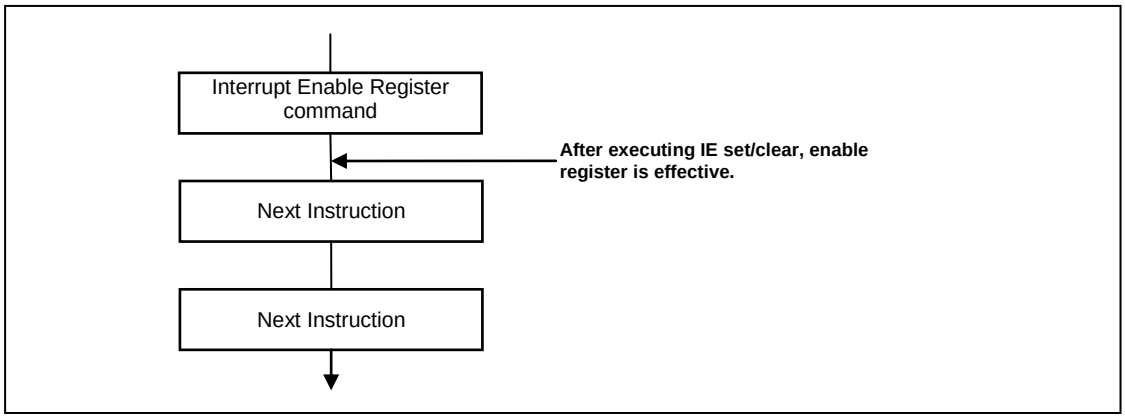



**Figure 10.4 Effective Timing of Interrupt Enable Register**

Case b) Interrupt flag Register

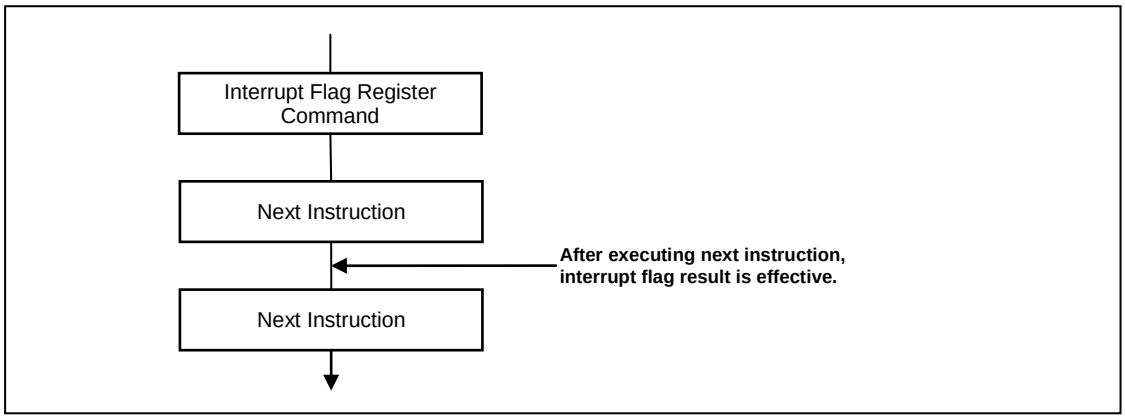



**Figure 10.5 Effective Timing of Interrupt Flag Register**

## 10.7 Multi Interrupt

If two requests of different priority levels are received simultaneously, the request of higher priority level is served first. If more than one interrupt request are received, the interrupt polling sequence determines which request is served first by hardware. However, for special features, multi-interrupt processing can be executed by software.

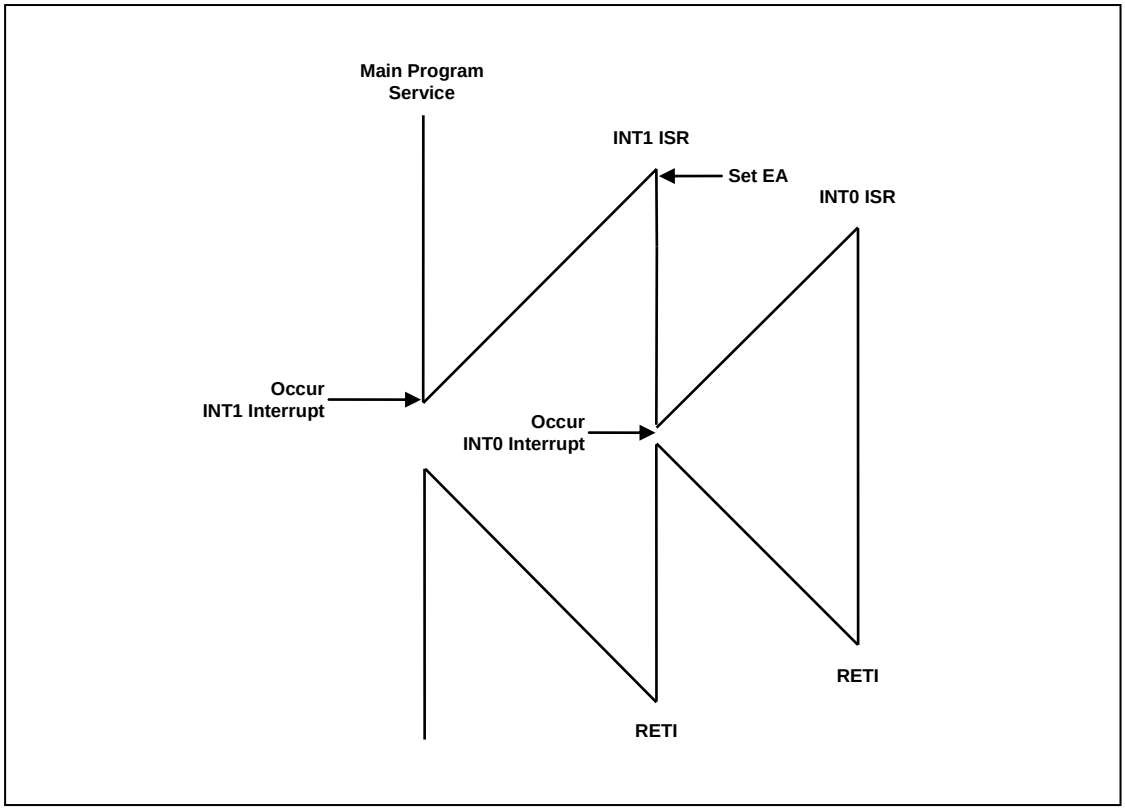



**Figure 10.6 Effective Timing of Interrupt**

Figure 10.6 shows an example of multi-interrupt processing. While INT1 is served, INT0 which has higher priority than INT1 is occurred. Then INT0 is served immediately and then the remain part of INT1 service routine is executed. If the priority level of INT0 is same or lower than INT1, INT0 will be served after the INT1 service has completed.

An interrupt service routine may be only interrupted by an interrupt of higher priority and, if two interrupts of different priority occur at the same time, the higher level interrupt will be served first. An interrupt cannot be interrupted by another interrupt of the same or a lower priority level. If two interrupts of the same priority level occur simultaneously, the service order for those interrupts is determined by the scan order.

## 10.8 Interrupt Enable Accept Timing

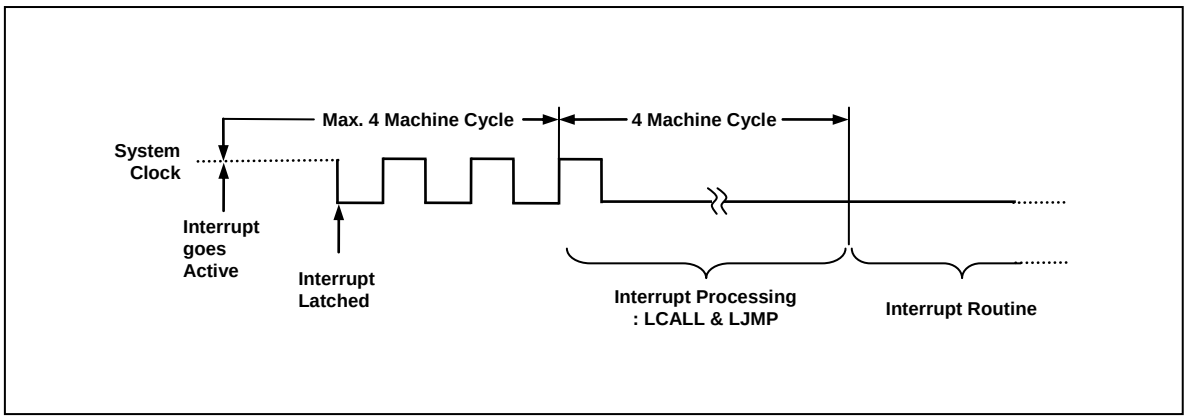



**Figure 10.7 Interrupt Response Timing Diagram**

## 10.9 Interrupt Service Routine Address

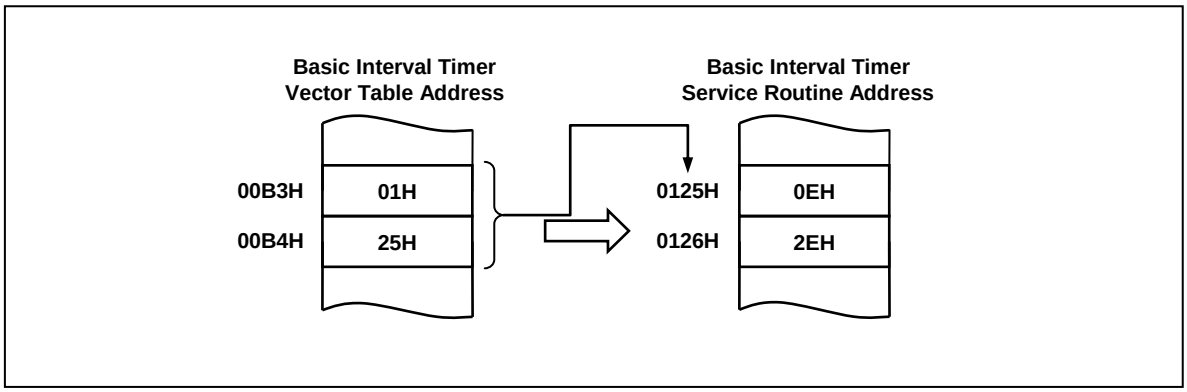



**Figure 10.8 Correspondence between Vector Table Address and the Entry Address of ISP**

## 10.10 Saving/Restore General-Purpose Registers

```
INTxx : PUSH   PSW
        PUSH   DPL
        PUSH   DPH
        PUSH   B
        PUSH   ACC
        :

Interrupt_Processing:
        :

        POP   ACC
        POP   B
        POP   DPH
        POP   DPL
        POP   PSW
        RETI
```

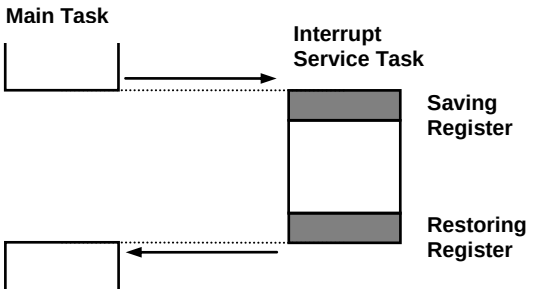



**Figure 10.9 Saving/Restore Process Diagram and Sample Source**

## 10.11 Interrupt Timing

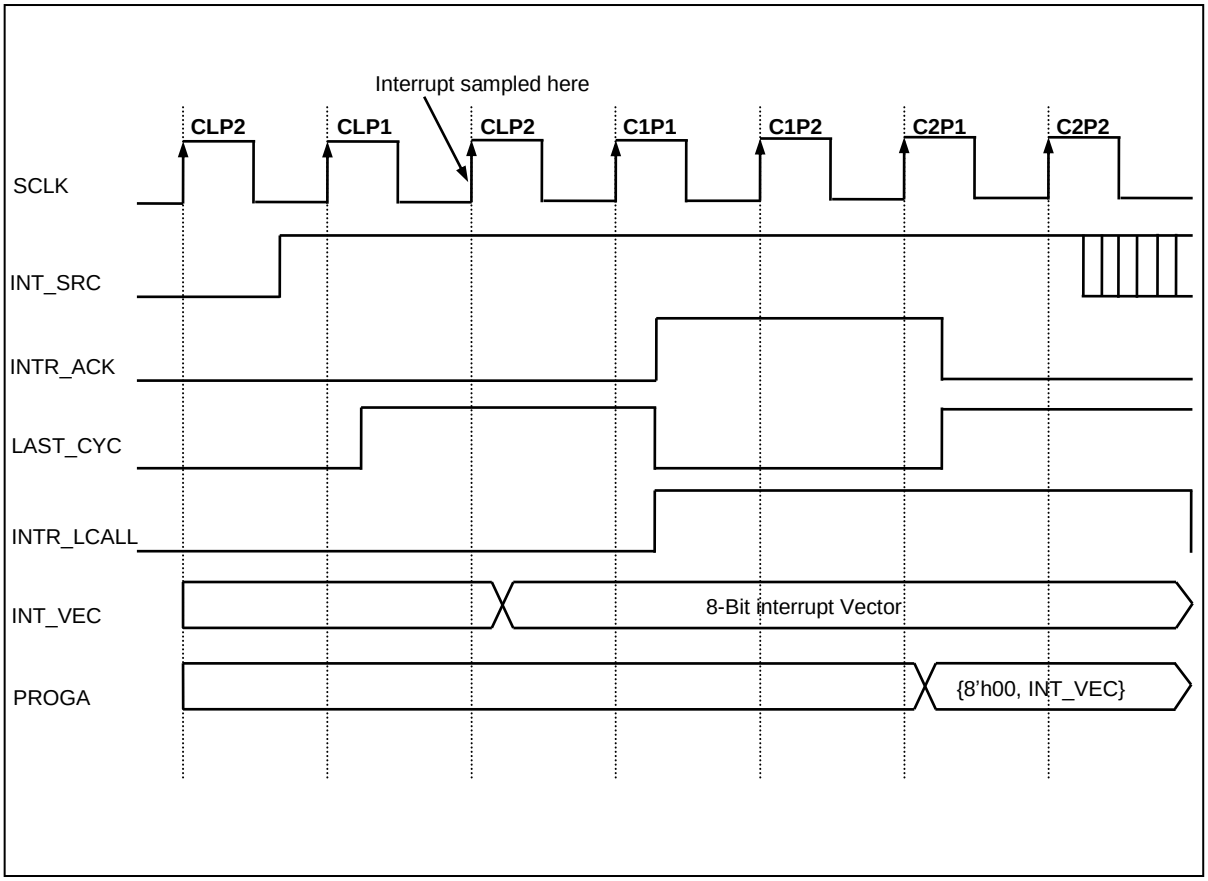



**Figure 10.10 Timing Chart of Interrupt Acceptance and Interrupt Return Instruction**

Interrupt sources are sampled at the last cycle of a command. If an interrupt source is detected the lower 8-bit of interrupt vector (INT_VEC) is decided. M8051W core makes interrupt acknowledge at the first cycle of a command, and executes long call to jump to interrupt service routine.

NOTE) command cycle CLPx: L=Last cycle, 1=1$^{st}$ cycle or 1$^{st}$ phase, 2=2$^{nd}$ cycle or 2$^{nd}$ phase

## 10.12 Interrupt Register Overview

### 10.12.1 Interrupt Enable Register (IE, IE1, IE2, IE3)

Interrupt enable register consists of global interrupt control bit (EA) and peripheral interrupt control bits. Total 24 peripherals are able to control interrupt.

### 10.12.2 Interrupt Priority Register (IP, IP1)

The 24 interrupts are divided into 6 groups which have each 4 interrupt sources. A group can be assigned 4 levels interrupt priority using interrupt priority register. Level 3 is the highest priority, while level 0 is the lowest priority. After a reset IP and IP1 are cleared to '00H'. If interrupts have the same priority level, lower number interrupt is served first.

### 10.12.3 External Interrupt Flag Register (EIFLAG0, EIFLAG1, EIFLAG2)

The external interrupt flag 0 register (EIFLAG0), external interrupt flag 1 register (EIFLAG1) and external interrupt flag 2 register (EIFLAG2) are set to '1' when the external interrupt generating condition is satisfied. The flag is cleared when the interrupt service routine is executed. Alternatively, the flag can be cleared by writing '0' to it.

### 10.12.4 External Interrupt Polarity Register (EIPOL0L/H, EIPOL1, EIPOL2L/H)

The external interrupt polarity 0 high/low register (EIPOL0H/L), external interrupt polarity 1 register (EIPOL1), external interrupt polarity 2 high/low register (EIPOL2H/L) determines which type of rising/falling/both edge interrupt. Initially, default value is no interrupt at any edge.

### 10.12.5 Register Map

**Table 10-3 Interrupt Register Map**

| Name | Address | Dir | Default | Description |
|---|---|---|---|---|
| IE | A8H | R/W | 00H | Interrupt Enable Register |
| IE1 | A9H | R/W | 00H | Interrupt Enable Register 1 |
| IE2 | AAH | R/W | 00H | Interrupt Enable Register 2 |
| IE3 | ABH | R/W | 00H | Interrupt Enable Register 3 |
| IP | B8H | R/W | 00H | Interrupt Priority Register |
| IP1 | F8H | R/W | 00H | Interrupt Priority Register 1 |
| EIFLAG0 | A3H | R/W | 00H | External Interrupt Flag 0 Register |
| EIPOL0L | A6H | R/W | 00H | External Interrupt Polarity 0 Low Register |
| EIPOL0H | A7H | R/W | 00H | External Interrupt Polarity 0 High Register |
| EIFLAG1 | A4H | R/W | 00H | External Interrupt Flag 1 Register |
| EIPOL1 | ADH | R/W | 00H | External Interrupt Polarity 1 Register |
| EIFLAG2 | A5H | R/W | 00H | External Interrupt Flag 2 Register |
| EIPOL2L | AEH | R/W | 00H | External Interrupt Polarity 2 Low Register |
| EIPOL2H | AFH | R/W | 00H | External Interrupt Polarity 2 High Register |

## 10.13 Interrupt Register Description

The interrupt register is used for controlling interrupt functions. Also it has external interrupt control registers. The interrupt register consists of interrupt enable register (IE), interrupt enable register 1 (IE1), interrupt enable register 2 (IE2) and interrupt enable register 3 (IE3). For external interrupt, it consists of external interrupt flag 0 register (EIFLAG0), external interrupt polarity 0 high/low register (EIPOL0H/L), external interrupt flag 1 register (EIFLAG1), external interrupt polarity 1 register (EIPOL1), external interrupt flag 2 register (EIFLAG2), and external interrupt polarity 2 high/low register (EIPOL2H/L).

### 10.13.1 Register Description for Interrupt

**IE (Interrupt Enable Register) : A8H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| EA | – | INT5E | INT4E | INT3E | INT2E | INT1E | INT0E |
| R/W | – | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**EA** Enable or Disable All Interrupt bits
- 0 All Interrupt disable
- 1 All Interrupt enable

**INT5E** Enable or Disable External Interrupt 0 ~ 7 (EINT0 ~ EINT7)
- 0 Disable
- 1 Enable

**INT4E** Enable or Disable USI1 Tx/I2C Interrupt
- 0 Disable
- 1 Enable

**INT3E** Enable or Disable USI1 Rx Interrupt
- 0 Disable
- 1 Enable

**INT2E** Enable or Disable UART4 Tx Interrupt
- 0 Disable
- 1 Enable

**INT1E** Enable or Disable UART3 Tx Interrupt
- 0 Disable
- 1 Enable

**INT0E** Enable or Disable UART2 Tx Interrupt
- 0 Disable
- 1 Enable

**IE1 (Interrupt Enable Register 1): A9H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | INT11E | INT10E | INT9E | INT8E | INT7E | INT6E |
| – | – | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value: 00H

**INT11E** Enable or Disable ADC Interrupt
- 0 Disable
- 1 Enable

**INT10E** Enable or Disable USI0 Tx/I2C Interrupt
- 0 Disable
- 1 Enable

**INT9E** Enable or Disable USI0 Rx Interrupt
- 0 Disable
- 1 Enable

**INT8E** Enable or Disable UART4 Rx Interrupt
- 0 Disable
- 1 Enable

**INT7E** Enable or Disable UART3 Rx Interrupt
- 0 Disable
- 1 Enable

**INT6E** Enable or Disable UART2 Rx Interrupt
- 0 Disable
- 1 Enable

**IE2 (Interrupt Enable Register 2) : AAH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | INT16E | INT15E | INT14E | INT13E | INT12E |
| – | – | – | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**INT16E** Enable or Disable External interrupt 8/9/18(EINT8/EINT9/EINT18)
0 Disable
1 Enable

**INT15E** Enable or Disable Timer 8 Interrupt
0 Disable
1 Enable

**INT14E** Enable or Disable Timer 7 Match/Emergency stop Interrupt
0 Disable
1 Enable

**INT13E** Enable or Disable Timer 3/4/5/6 Match nterrupt
0 Disable
1 Enable

**INT12E** Enable or Disable Timer 0/1/2 Overflow/Match Interrupt
0 Disable
1 Enable

**IE3 (Interrupt Enable Register 3) : ABH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | INT23E | INT22E | INT21E | INT20E | INT19E | INT18E |
| – | – | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**INT23E** Enable or Disable External Interrupt 10 ~ 17 (EINT10 ~ EINT17)

0 Disable

1 Enable

**INT22E** Enable or Disable BIT Interrupt

0 Disable

1 Enable

**INT21E** Enable or Disable WDT Interrupt

0 Disable

1 Enable

**INT20E** Enable or Disable WT, PWM generator overflow, Shot stop, and TRIG Interrupt

0 Disable

1 Enable

**INT19E** Enable or Disable SPI3 Interrupt

0 Disable

1 Enable

**INT18E** Enable or Disable SPI2 Interrupt

0 Disable

1 Enable

**IP (Interrupt Priority Register) : B8H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | IP5 | IP4 | IP3 | IP2 | IP1 | IP0 |
| – | – | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**IP1 (Interrupt Priority Register 1) : F8H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | IP15 | IP14 | IP13 | IP12 | IP11 | IP10 |
| – | – | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**IP[5:0], IP1[5:0]** Select Interrupt Group Priority

| IP1x | IPx | Description |
|---|---|---|
| 0 | 0 | level 0 (lowest) |
| 0 | 1 | level 1 |
| 1 | 0 | level 2 |
| 1 | 1 | level 3 (highest) |

**EIFLAG0 (External Interrupt Flag 0 Register) : A3H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FLAG7 | FLAG6 | FLAG5 | FLAG4 | FLAG3 | FLAG2 | FLAG1 | FLAG0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**EIFLAG0[7:0]** When an External Interrupt 0-7(EINT0-EINT7) is occurred, the flag becomes '1'. These flags are cleared only by writing a '0' to the corresponding bit. So, the FLAG0-FLAG7 flag should be cleared by software.

0 External Interrupt 0 ~ 7 not occurred

1 External Interrupt 0 ~ 7 occurred

**EIPOL0H (External Interrupt Polarity 0 High Register): A7H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| POL7 | | POL6 | | POL5 | | POL4 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value: 00H

**EIPOL0H[7:0]** External interrupt (EINT7, EINT6, EINT5, EINT4) polarity selection

| POLn[1:0] | | Description |
|---|---|---|
| 0 | 0 | No interrupt at any edge |
| 0 | 1 | Interrupt on rising edge |
| 1 | 0 | Interrupt on falling edge |
| 1 | 1 | Interrupt on both of rising and falling edge |

Where n = 4, 5, 6 and 7

**EIPOL0L (External Interrupt Polarity 0 Low Register): A6H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| POL3 | | POL2 | | POL1 | | POL0 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value: 00H

**EIPOL0L[7:0]** External interrupt (EINT3, EINT2, EINT1, EINT0) polarity selection

| POLn[1:0] | | Description |
|---|---|---|
| 0 | 0 | No interrupt at any edge |
| 0 | 1 | Interrupt on rising edge |
| 1 | 0 | Interrupt on falling edge |
| 1 | 1 | Interrupt on both of rising and falling edge |

Where n =0, 1, 2 and 3

**EIFLAG1 (External Interrupt Flag 1 Register) : A4H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| T7IFR | T6IFR | T5IFR | T4IFR | T3IFR | FLAG18 | FLAG9 | FLAG8 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**T7IFR** When T7 interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software.

0 T7 Interrupt no generation

1 T7 Interrupt generation

**T6IFR** When T6 interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software.

0 T6 Interrupt no generation

1 T6 Interrupt generation

**T5IFR** When T5 interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software.

0 T5 Interrupt no generation

1 T5 Interrupt generation

**T4IFR** When T4 interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software.

0 T4 Interrupt no generation

1 T4 Interrupt generation

**T3IFR** When T3 interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software.

0 T3 Interrupt no generation

1 T3 Interrupt generation

**EIFLAG1[2:0]** When an External Interrupt (EINT8, EINT9, EINT18) is occurred, the flag becomes '1'. These flags are cleared only by writing a '0' to the corresponding bit. So, the FLAG8, FLAG9, FLAG18 flag should be cleared by software.

0 External Interrupt 8/9/18 not occurred

1 External Interrupt 8/9/18 occurred

**EIPOL1 (External Interrupt Polarity 1 Register): ADH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | POL18 | | POL9 | | POL8 | |
| – | – | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value: 00H

**EIPOL1[7:0]** External interrupt (EINT18, EINT9, EINT8) polarity selection

| POLn[1:0] | | Description |
|---|---|---|
| 0 | 0 | No interrupt at any edge |
| 0 | 1 | Interrupt on rising edge |
| 1 | 0 | Interrupt on falling edge |
| 1 | 1 | Interrupt on both of rising and falling edge |

Where n =8, 9 and 18

**EIFLAG2 (External Interrupt Flag 2 Register) : A5H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FLAG17 | FLAG16 | FLAG15 | FLAG14 | FLAG13 | FLAG12 | FLAG11 | FLAG10 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**EIFLAG2[7:0]** When an External Interrupt 10-17(EINT10-EINT17) is occurred, the flag becomes '1'. These flags are cleared only by writing a '0' to the corresponding bit. So, the FLAG10-FLAG17 flag should be cleared by software.

0 External Interrupt 10 ~ 17 not occurred

1 External Interrupt 10 ~ 17 occurred

**EIPOL2H (External Interrupt Polarity 2 High Register): AFH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| POL17 | | POL16 | | POL15 | | POL14 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value: 00H

**EIPOL2H[7:0]** External interrupt (EINT17, EINT16, EINT15, EINT14) polarity selection

| POLn[1:0] | | Description |
|---|---|---|
| 0 | 0 | No interrupt at any edge |
| 0 | 1 | Interrupt on rising edge |
| 1 | 0 | Interrupt on falling edge |
| 1 | 1 | Interrupt on both of rising and falling edge |

Where n = 14, 15, 16 and 17

**EIPOL2L (External Interrupt Polarity 2 Low Register): AEH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| POL13 | | POL12 | | POL11 | | POL10 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value: 00H

**EIPOL2L[7:0]** External interrupt (EINT13, EINT12, EINT11, EINT10) polarity selection

| POLn[1:0] | | Description |
|---|---|---|
| 0 | 0 | No interrupt at any edge |
| 0 | 1 | Interrupt on rising edge |
| 1 | 0 | Interrupt on falling edge |
| 1 | 1 | Interrupt on both of rising and falling edge |

Where n =10, 11, 12 and 13

# 11. Peripheral Hardware

## 11.1 Clock Generator

### 11.1.1 Overview

As shown in Figure 11.1, the clock generator produces the basic clock pulses which provide the system clock to be supplied to the CPU and the peripheral hardware. It contains main/sub-frequency clock oscillator. The main/sub clock operation can be easily obtained by attaching a crystal between the XIN/SXIN and XOUT/SXOUT pin, respectively. The main/sub clock can be also obtained from the external oscillator. In this case, it is necessary to put the external clock signal into the XIN/SXIN pin and open the XOUT/SXOUT pin. The default system clock is 1MHz INT-RC Oscillator and the default division rate is sixteen. In order to stabilize system internally, it is used 1MHz INT-RC oscillator on POR.

- Calibrated Internal RC Oscillator (16 MHz )
  . INT-RC OSC/1 (16 MHz)
  . INT-RC OSC/2 (8 MHz)
  . INT-RC OSC/4 (4 MHz)
  . INT-RC OSC/8 (2 MHz)
  . INT-RC OSC/16 (1 MHz, Default system clock)
  . INT-RC OSC/32 (0.5 MHz)
- Main Crystal Oscillator (0.4~12 MHz)
- Sub Crystal Oscillator (32.768 kHz)
- Internal WDTRC Oscillator (5 kHz)

### 11.1.2 Block Diagram

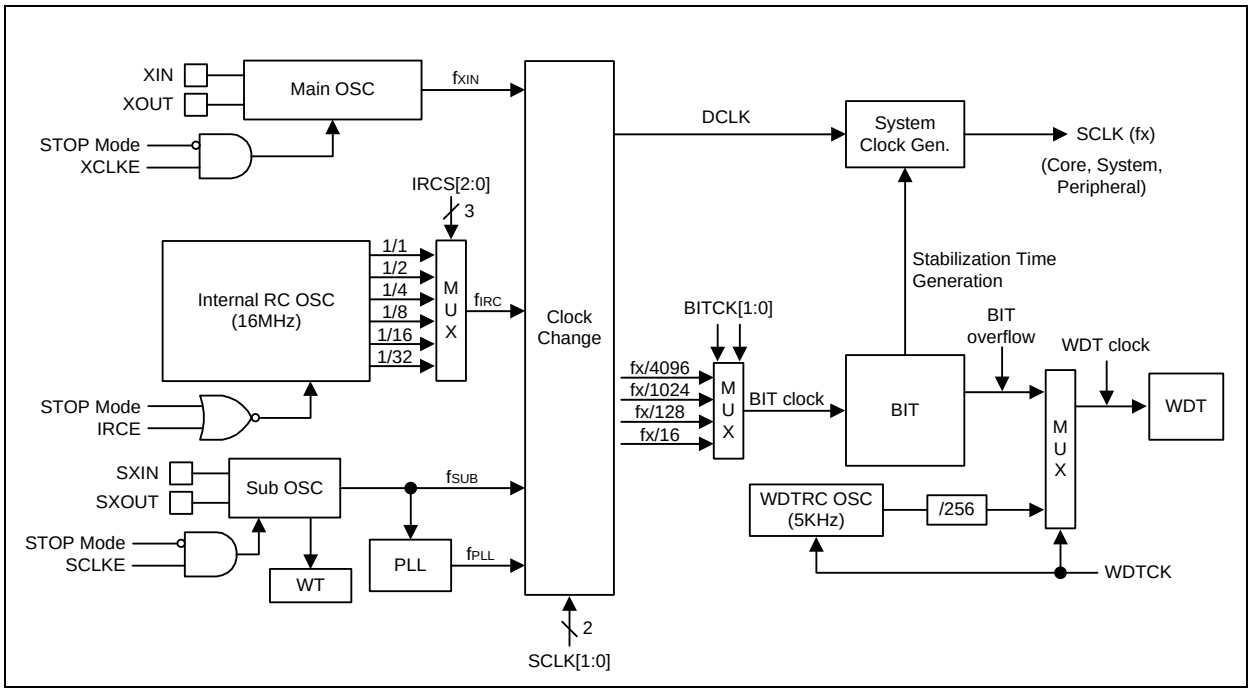



**Figure 11.1 Clock Generator Block Diagram**

### 11.1.3 Phase Locked-Loop Block Diagram

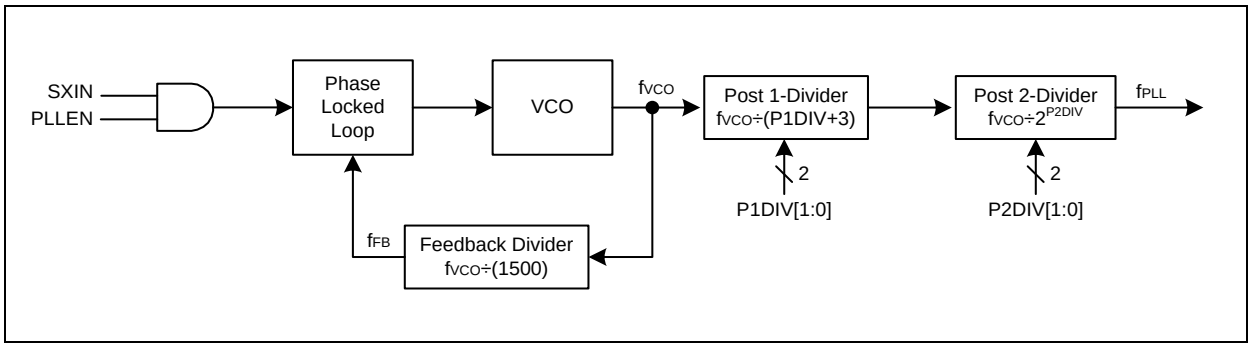



**Figure 11.2 Phase Locked-Loop Circuit Diagram**

### 11.1.4 Register Map

**Table 11-1 Clock Generator Register Map**

| Name | Address | Dir | Default | Description |
|---|---|---|---|---|
| SCCR | 8AH | R/W | 00H | System and Clock Control Register |
| OSCCR | C8H | R/W | 08H | Oscillator Control Register |
| PLLCR | 95H | R/W | 00H | Phase Locked-Loop Control Register |

### 11.1.5 Clock Generator Register Description

The clock generator register uses clock control for system operation. The clock generation consists of system and clock control register, oscillator control register, and phase locked-loop control register.

## 11.1.6 Register Description for Clock Generator

**SCCR (System and Clock Control Register) : 8AH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | PSAVE | – | – | – | – | SCLK1 | SCLK0 |
| – | R/W | – | – | – | – | R/W | R/W |

Initial value : 00H

**PSAVE** Power Save Mode Control Bit

0 Normal circuit for sub oscillator

1 Power saving circuit for sub oscillator

NOTES)

1. A capacitor (0.1μF) should be connected between VREG and VSS when the sub oscillator is used to power saving mode.
2. The PSAVE automatically cleared to '0' when the sub oscillator is stopped by SCLKE or CPU is entered into STOP mode in sub operating mode.
3. The delay is needed 500ms over from sub osc start to PSAVE change to "1".

**SCLK [1:0]** System Clock Selection Bit

| SCLK1 | SCLK0 | Description |
|---|---|---|
| 0 | 0 | INT RC OSC ($f_{IRC}$) for system clock |
| 0 | 1 | External Main OSC ($f_{XIN}$) for system clock |
| 1 | 0 | External Sub OSC ($f_{SUB}$) for system clock |
| 1 | 1 | Phase Locked-Loop ($f_{PLL}$) for system clock |

**OSCCR (Oscillator Control Register) : C8H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | IRCS2 | IRCS1 | IRCS0 | IRCE | XCLKE | SCLKE |
| – | – | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 08H

**IRCS[2:0]** Internal RC Oscillator Post-divider Selection

| IRCS2 | IRCS1 | IRCS0 | Description |
|---|---|---|---|
| 0 | 0 | 0 | INT-RC/32 (0.5MHz) |
| 0 | 0 | 1 | INT-RC/16 (1MHz) |
| 0 | 1 | 0 | INT-RC/8 (2MHz) |
| 0 | 1 | 1 | INT-RC/4 (4MHz) |
| 1 | 0 | 0 | INT-RC/2 (8MHz) |
| 1 | 0 | 1 | INT-RC/1 (16MHz) |
| Other values | | | Not used |

**IRCE** Control the Operation of the Internal RC Oscillator

| | |
|---|---|
| 0 | Enable operation of INT-RC OSC |
| 1 | Disable operation of INT-RC OSC |

**XCLKE** Control the Operation of the External Main Oscillator

| | |
|---|---|
| 0 | Disable operation of X-TAL |
| 1 | Enable operation of X-TAL |

**SCLKE** Control the Operation of the External Sub Oscillator

| | |
|---|---|
| 0 | Disable operation of SX-TAL |
| 1 | Enable operation of SX-TAL |

**PLLCR (Phase Locked-Loop Control Register) : 95H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | PLLSTA | P1DIV1 | P1DIV0 | P2DIV1 | P2DIV0 | PLLEN |
| – | – | R | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**PLLSTA** PLL Locked/Unlocked Status Bit

0 PLL currently in unlocked state

1 PLL currently in locked state

**P1DIV[1:0]** PLL Post 1-Divider Selection Bits (49.152MHz)

| P1DIV1 | P1DIV0 | Description |
|---|---|---|
| 0 | 0 | $f_{VCO}/3$ = 16.384MHz |
| 0 | 1 | $f_{VCO}/4$ = 12.888MHz |
| 1 | 0 | $f_{VCO}/5$ = 9.8304MHz |
| 1 | 1 | $f_{VCO}/6$ = 8.192MHz |

**P2DIV[1:0]** PLL Post 2-Divider Data Bits

| P2DIV1 | P2DIV0 | Description |
|---|---|---|
| 0 | 0 | $f_{PLL} = f_{VCO}/1$ |
| 0 | 1 | $f_{PLL} = f_{VCO}/2$ |
| 1 | 0 | $f_{PLL} = f_{VCO}/4$ |
| 1 | 1 | $f_{PLL} = f_{VCO}/8$ |

**PLLEN** PLL Enable/Disable Control Bit

0 PLL Disable

1 PLL Enable

## 11.2 Basic Interval Timer

### 11.2.1 Overview

The MC96F7864 has one 8-bit basic interval timer that is free-run and can't stop. Block diagram is shown in Figure 11.3. In addition, the basic interval timer generates the time base for watchdog timer counting. It also provides a basic interval timer interrupt (BITIFR).

The MC96F7864 has these basic interval timer (BIT) features:

- During Power On, BIT gives a stable clock generation time
- On exiting Stop mode, BIT gives a stable clock generation time
- As timer function, timer interrupt occurrence

### 11.2.2 Block Diagram

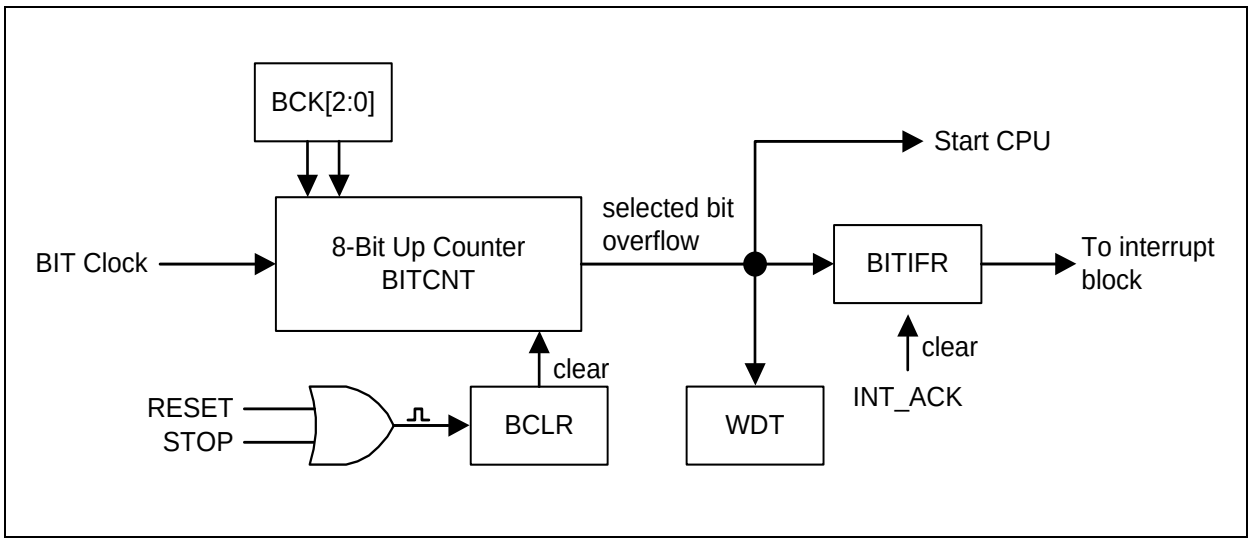



**Figure 11.3 Basic Interval Timer Block Diagram**

### 11.2.3 Register Map

**Table 11-2 Basic Interval Timer Register Map**

| Name | Address | Dir | Default | Description |
|---|---|---|---|---|
| BITCNT | 8CH | R | 00H | Basic Interval Timer Counter Register |
| BITCR | 8BH | R/W | 01H | Basic Interval Timer Control Register |

### 11.2.4 Basic Interval Timer Register Description

The basic interval timer register consists of basic interval timer counter register (BITCNT) and basic interval timer control register (BITCR). If BCLR bit is set to '1', BITCNT becomes '0' and then counts up. After 1 machine cycle, BCLR bit is cleared to '0' automatically.

### 11.2.5 Register Description for Basic Interval Timer

**BITCNT (Basic Interval Timer Counter Register) : 8CH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| BITCNT7 | BITCNT6 | BITCNT5 | BITCNT4 | BITCNT3 | BITCNT2 | BITCNT1 | BITCNT0 |
| R | R | R | R | R | R | R | R |

Initial value : 00H

**BITCNT[7:0]** BIT Counter

**BITCR (Basic Interval Timer Control Register) : 8BH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| BITIFR | BITCK1 | BITCK0 | – | BCLR | BCK2 | BCK1 | BCK0 |
| R/W | R/W | R/W | – | R/W | R/W | R/W | R/W |

Initial value : 01H

**BITIFR** When BIT Interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or auto clear by INT_ACK signal.

0 BIT interrupt no generation

1 BIT interrupt generation

**BITCK[1:0]** Select BIT clock source

| BITCK1 | BITCK0 | Description |
|---|---|---|
| 0 | 0 | fx/4096 |
| 0 | 1 | fx/1024 |
| 1 | 0 | fx/128 |
| 1 | 1 | fx/16 |

**BCLR** If this bit is written to '1', BIT Counter is cleared to '0'

0 Free Running

1 Clear Counter

**BCK[2:0]** Select BIT overflow period

| BCK2 | BCK1 | BCK0 | Description |
|---|---|---|---|
| 0 | 0 | 0 | Bit 0 overflow (BIT Clock * 2) |
| 0 | 0 | 1 | Bit 1 overflow (BIT Clock * 4) (default) |
| 0 | 1 | 0 | Bit 2 overflow (BIT Clock * 8) |
| 0 | 1 | 1 | Bit 3 overflow (BIT Clock * 16) |
| 1 | 0 | 0 | Bit 4 overflow (BIT Clock * 32) |
| 1 | 0 | 1 | Bit 5 overflow (BIT Clock * 64) |
| 1 | 1 | 0 | Bit 6 overflow (BIT Clock * 128) |
| 1 | 1 | 1 | Bit 7 overflow (BIT Clock * 256) |

## 11.3 Watch Dog Timer

### 11.3.1 Overview

The watchdog timer rapidly detects the CPU malfunction such as endless looping caused by noise or something like that, and resumes the CPU to the normal state. The watchdog timer signal for malfunction detection can be used as either a CPU reset or an interrupt request. When the watchdog timer is not being used for malfunction detection, it can be used as a timer to generate an interrupt at fixed intervals. It is possible to use free running 8-bit timer mode (WDTRSON='0') or watch dog timer mode (WDTRSON='1') as setting WDTCR[6] bit. If WDTCR[5] is written to '1', WDT counter value is cleared and counts up. After 1 machine cycle, this bit is cleared to '0' automatically. The watchdog timer consists of 8-bit binary counter and the watchdog timer data register. When the value of 8-bit binary counter is equal to the 8 bits of WDTCNT, the interrupt request flag is generated. This can be used as Watchdog timer interrupt or reset of CPU in accordance with the bit WDTRSON.

The input clock source of watch dog timer is the BIT overflow. The interval of watchdog timer interrupt is decided by BIT overflow period and WDTDR set value. The equation can be described as

WDT Interrupt Interval = (BIT Interrupt Interval) X (WDTDR Value+1)

### 11.3.2 WDT Interrupt Timing Waveform

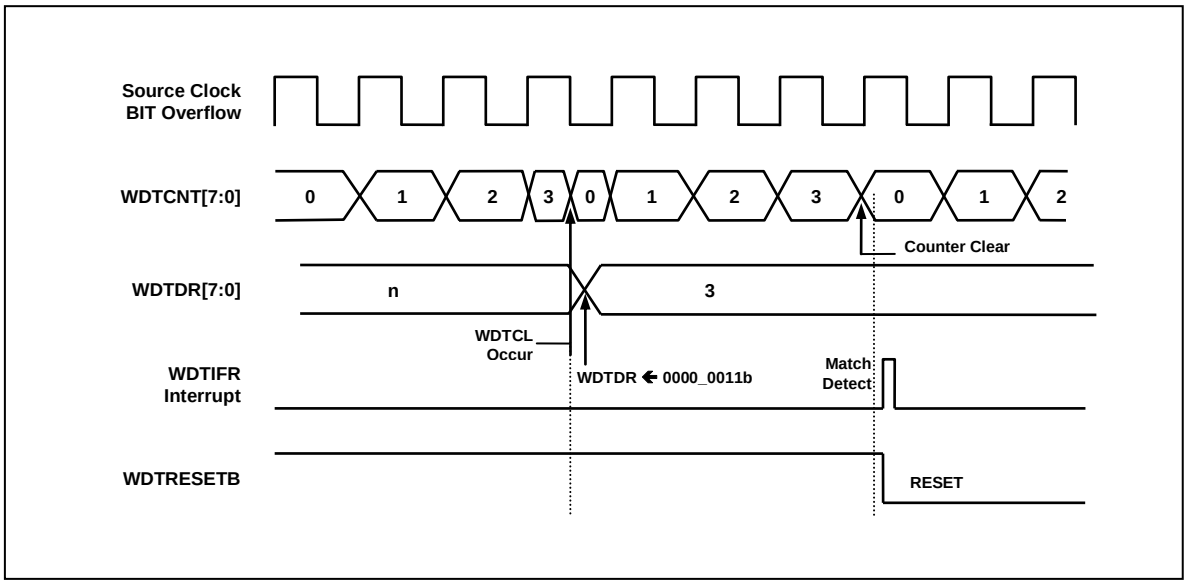



**Figure 11.4 Watch Dog Timer Interrupt Timing Waveform**

### 11.3.3 Block Diagram

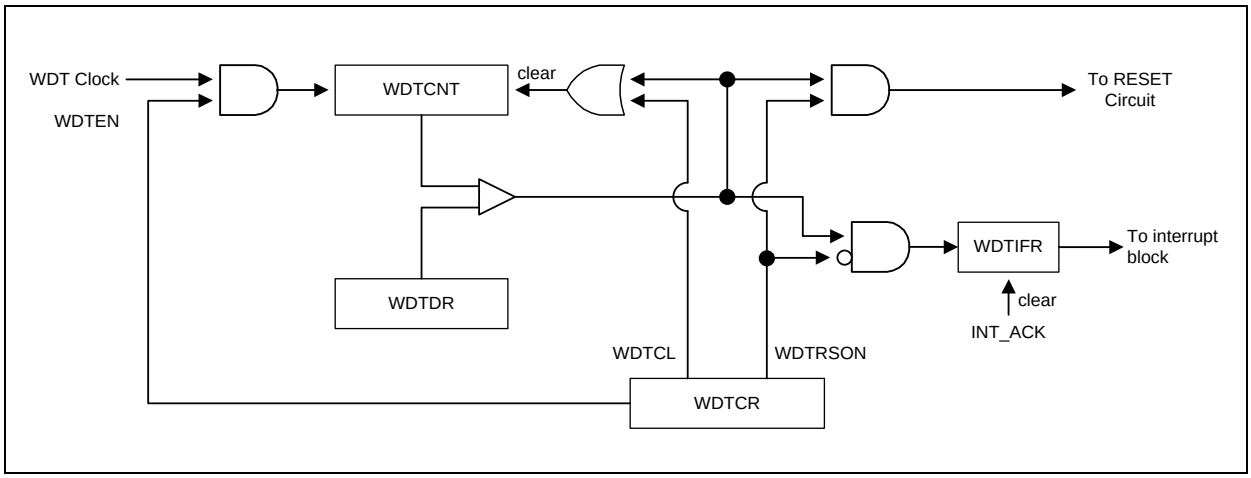



**Figure 11.5 Watch Dog Timer Block Diagram**

### 11.3.4 Register Map

**Table 11-3 Watch Dog Timer Register Map**

| Name | Address | Dir | Default | Description |
|---|---|---|---|---|
| WDTCNT | 8EH | R | 00H | Watch Dog Timer Counter Register |
| WDTDR | 8EH | W | FFH | Watch Dog Timer Data Register |
| WDTCR | 8DH | R/W | 00H | Watch Dog Timer Control Register |

### 11.3.5 Watch Dog Timer Register Description

The watch dog timer register consists of watch dog timer counter register (WDTCNT), watch dog timer data register (WDTDR) and watch dog timer control register (WDTCR).

### 11.3.6 Register Description for Watch Dog Timer

**WDTCNT (Watch Dog Timer Counter Register: Read Case) : 8EH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| WDTCNT7 | WDTCNT6 | WDTCNT5 | WDTCNT4 | WDTCNT3 | WDTCNT2 | WDTCNT1 | WDTCNT0 |
| R | R | R | R | R | R | R | R |

Initial value : 00H

**WDTCNT[7:0]** WDT Counter

**WDTDR (Watch Dog Timer Data Register: Write Case) : 8EH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| WDTDR7 | WDTDR6 | WDTDR5 | WDTDR4 | WDTDR3 | WDTDR2 | WDTDR1 | WDTDR0 |
| W | W | W | W | W | W | W | W |

Initial value : FFH

**WDTDR[7:0]** Set a period
WDT Interrupt Interval=(BIT Interrupt Interval) x(WDTDR Value+1)
NOTE) Do not write “0” in the WDTDR register.

**WDTCR (Watch Dog Timer Control Register) : 8DH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| WDTEN | WDTRSON | WDTCL | – | – | – | WDTCK | WDTIFR |
| R/W | R/W | R/W | – | – | – | R/W | R/W |

Initial value : 00H

**WDTEN** Control WDT Operation
0 Disable
1 Enable

**WDTRSON** Control WDT RESET Operation
0 Free Running 8-bit timer
1 Watch Dog Timer RESET ON

**WDTCL** Clear WDT Counter
0 Free Run
1 Clear WDT Counter (auto clear after 1 Cycle)

**WDTCK** Control WDT Clock Selection Bit
0 BIT overflow for WDT clock (WDTRC disable)
1 WDTRC for WDT clock (WDTRC enable)

**WDTIFR** When WDT Interrupt occurs, this bit becomes ‘1’. For clearing bit, write ‘0’ to this bit or auto clear by INT_ACK signal.
0 WDT Interrupt no generation
1 WDT Interrupt generation

## 11.4 Watch Timer

### 11.4.1 Overview

The watch timer has the function for RTC (Real Time Clock) operation. It is generally used for RTC design. The internal structure of the watch timer consists of the clock source select circuit, timer counter circuit, output select circuit, and watch timer control register. To operate the watch timer, determine the input clock source, output interval, and set WTEN to '1' in watch timer control register (WTCR). It is able to execute simultaneously or individually. To stop or reset WT, clear the WTEN bit in WTCR register. Even if CPU is STOP mode, sub clock is able to be so alive that WT can continue the operation. The watch timer counter circuits may be composed of 21-bit counter which contains low 14-bit with binary counter and high 7-bit counter in order to raise resolution. In WTDR, it can control WT clear and set interval value at write time, and it can read 7-bit WT counter value at read time.

The watch timer supplies the clock frequency for the LCD driver ($f_{LCD}$). Therefore, if the watch timer is disabled, the LCD driver controller does not operate.

### 11.4.2 Block Diagram

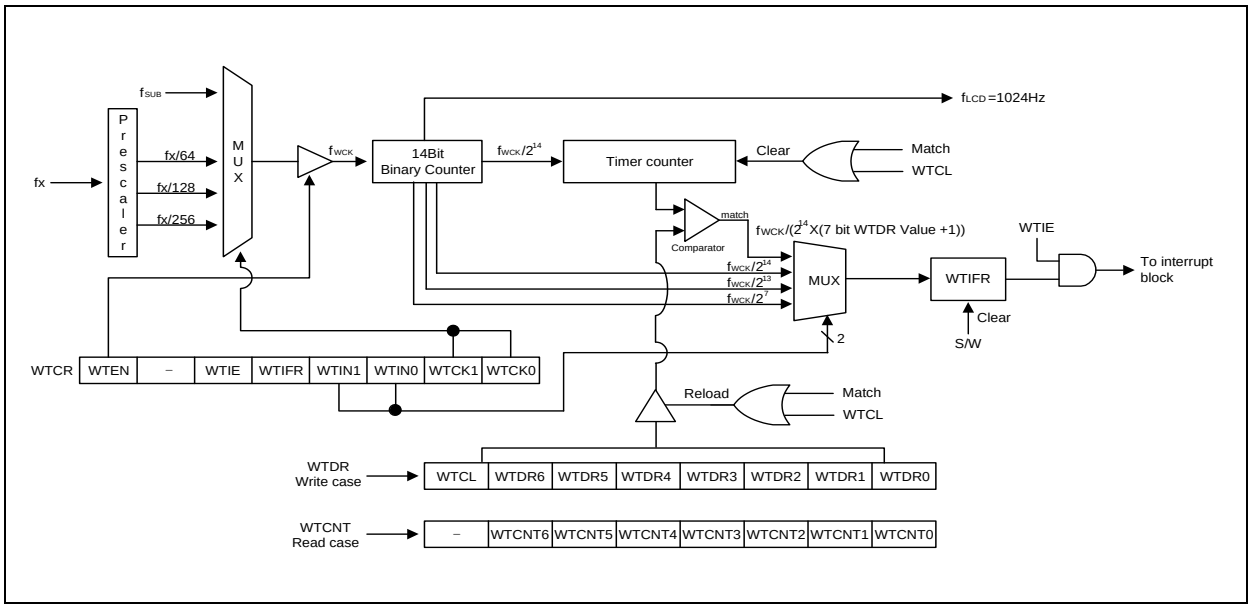



**Figure 11.6 Watch Timer Block Diagram**

### 11.4.3 Register Map

**Table 11-4 Watch Timer Register Map**

| Name | Address | Dir | Default | Description |
|---|---|---|---|---|
| WTCNT | 89H | R | 00H | Watch Timer Counter Register |
| WTDR | 89H | W | 7FH | Watch Timer Data Register |
| WTCR | 96H | R/W | 00H | Watch Timer Control Register |

### 11.4.4 Watch Timer Register Description

The watch timer register consists of watch timer counter register (WTCNT), watch timer data register (WTDR), and watch timer control register (WTCR). As WTCR is 7-bit writable/readable register, WTCR can control the clock source (WTCK[1:0]), interrupt interval (WTIN[1:0]), and function enable/disable (WTEN). Also there is WT interrupt flag bit (WTIFR).

### 11.4.5 Register Description for Watch Timer

**WTCNT (Watch Timer Counter Register: Read Case) : 89H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | WTCNT6 | WTCNT5 | WTCNT4 | WTCNT3 | WTCNT2 | WTCNT1 | WTCNT0 |
| – | R | R | R | R | R | R | R |

Initial value : 00H

**WTCNT[6:0]** WT Counter

**WTDR (Watch Timer Data Register: Write Case) : 89H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| WTCL | WTDR6 | WTDR5 | WTDR4 | WTDR3 | WTDR2 | WTDR1 | WTDR0 |
| R/W | W | W | W | W | W | W | W |

Initial value : 7FH

**WTCL** Clear WT Counter

0 Free Run

1 Clear WT Counter (auto clear after 1 Cycle)

**WTDR[6:0]** Set WT period

WT Interrupt Interval=fwck/(2^14 x(7bit WTDR Value+1))

NOTE) Do not write “0” in the WTDR register.

**WTCR (Watch Timer Control Register) : 96H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| WTEN | – | WTIE | WTIFR | WTIN1 | WTIN0 | WTCK1 | WTCK0 |
| R/W | – | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**WTEN** Control Watch Timer

0 Disable

1 Enable

**WTIE** Enable or Disable Watch Timer

0 Disable

1 Enable

**WTIFR** When WT Interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software.

0 WT Interrupt no generation

1 WT Interrupt generation

**WTIN[1:0]** Determine interrupt interval

| WTIN1 | WTIN0 | Description |
|---|---|---|
| 0 | 0 | $f_{WCK}$/2^7 |
| 0 | 1 | $f_{WCK}$/2^13 |
| 1 | 0 | $f_{WCK}$/2^14 |
| 1 | 1 | $f_{WCK}$/(2^14 x (7bit WTDR Value+1)) |

**WTCK[1:0]** Determine Source Clock

| WTCK1 | WTCK0 | Description |
|---|---|---|
| 0 | 0 | $f_{SUB}$ |
| 0 | 1 | $f_X$/256 |
| 1 | 0 | $f_X$/128 |
| 1 | 1 | $f_X$/64 |

NOTES) $f_X$ – System clock frequency (Where fx= 4.19MHz)

$f_{SUB}$ – Sub clock oscillator frequency (32.768kHz)

$f_{WCK}$ – Selected Watch timer clock

$f_{LCD}$ – LCD frequency (Where $f_X$= 4.19MHz, WTCK[1:0]='10'; $f_{LCD}$= 1024Hz)

## 11.5 Timer 0/1/2

### 11.5.1 Overview

The 8-bit timer 0/1/2 consists of multiplexer, timer 0/1/2 counter register, timer 0/1/2 data register, timer 0/1/2 capture data register, and timer 0/1/2 control register (TnCNT, TnDR, TnCDR, TnCR).

It has three operating modes:

- 8-bit timer/counter mode
- 8-bit PWM output mode
- 8-bit capture mode

The timer/counter 0/1/2 can be clocked by an internal or an external clock source (ECn). The clock source is selected by clock selection logic which is controlled by the clock selection bits (TnCK[2:0]).

- TIMER 0/1/2 clock source: $f_X$/2, 4, 8, 32, 128, 512, 2048 and ECn

In the capture mode, by EINT1n, the data is captured into input capture data register (TnCDR). In timer/counter mode, whenever counter value is equal to TnDR, TnO port toggles. Also the timer 0/1/2 outputs PWM waveform through PWMnO port in the PWM mode.

**Table 11-5 Timer 0/1/2 Operating Modes**

| TnEN | TnMS[1:0] | TnCK[2:0] | Timer n |
|---|---|---|---|
| 1 | 00 | XXX | 8 Bit Timer/Counter Mode |
| 1 | 01 | XXX | 8 Bit PWM Mode |
| 1 | 1X | XXX | 8 Bit Capture Mode |

### 11.5.2 8-Bit Timer/Counter Mode

The 8-bit timer/counter mode is selected by control register as shown in Figure 11.7.

The 8-bit timer have counter and data register. The counter register is increased by internal or external clock input. Timer 0/1/2 can use the input clock with one of 2, 4, 8, 32, 128, 512 and 2048 prescaler division rates (TnCK[2:0]). When the value of TnCNT and TnDR is identical in timer 0/1/2, a match signal is generated and the interrupt of Timer n occurs. TnCNT value is automatically cleared by match signal. It can be also cleared by software (TnCC).

The external clock (ECn) counts up the timer at the rising edge. If the ECn is selected as a clock source by TnCK[2:0], EC0/EC1/EC2 port should be set to the input port by P40IO/P41IO/P42IO bit.

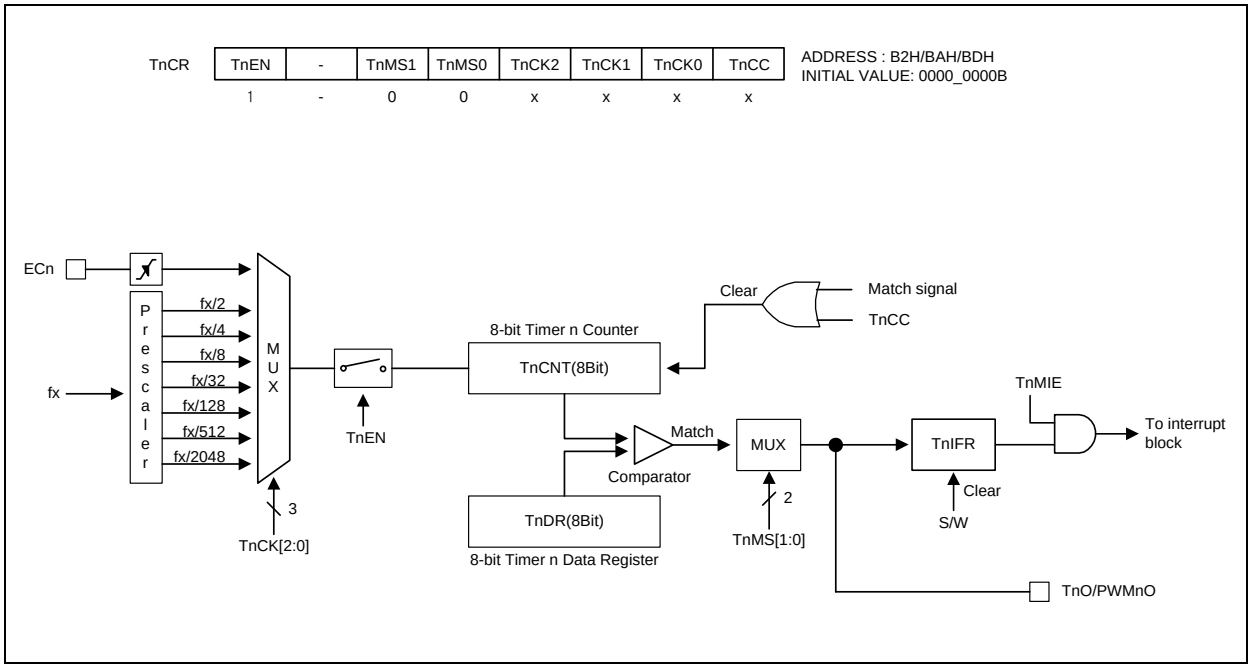



**Figure 11.7 8-Bit Timer/Counter Mode for Timer 0/1/2 (Where n = 0, 1, and 2)**

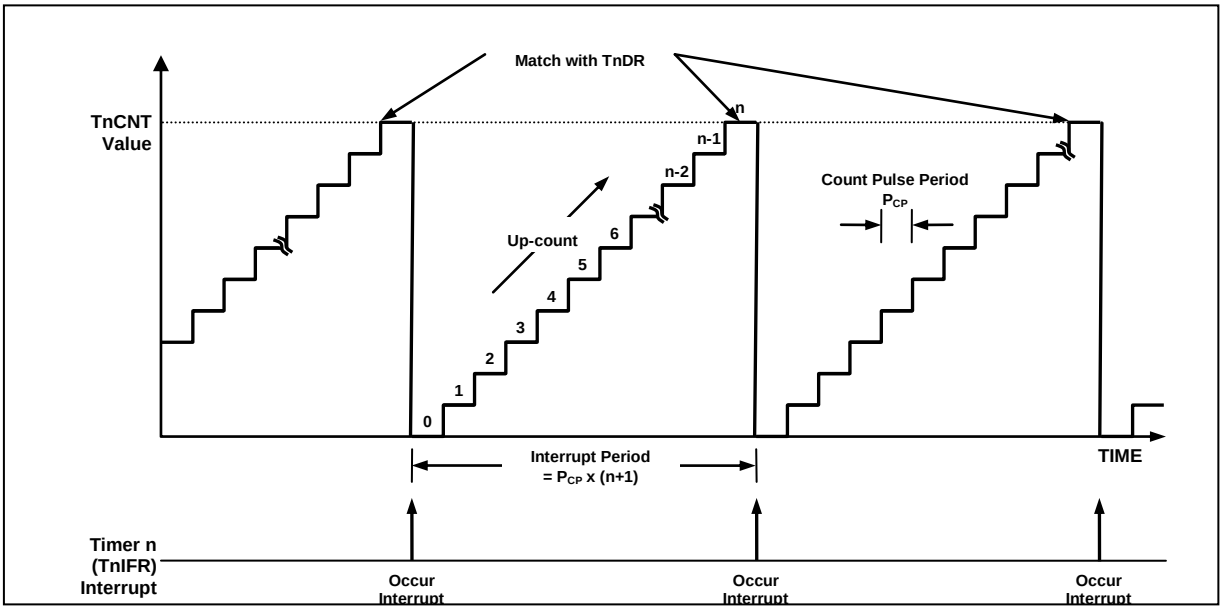



**Figure 11.8 8-Bit Timer/Counter 0/1/2 Example (Where n = 0, 1, and 2)**

### 11.5.3 8-Bit PWM Mode

The timer 0/1/2 has a high speed PWM (Pulse Width Modulation) function. In PWM mode, TnO/PWMnO pin outputs up to 8-bit resolution PWM output. This pin should be configured as a PWM output by setting the T0O/PWM0O, T1O/PWM1O and T2O/PWM2O function by P4FSRL[4:3], P4FSRH[1:0], and P4FSRH[3:2] bits. In the 8-bit timer/counter mode, a match signal is generated when the counter value is identical to the value of TnDR. When the value of TnCNT and TnDR is identical in timer n, a match signal is generated and the interrupt of timer 0/1/2 occurs. In PWM mode, the match signal does not clear the counter. Instead, it runs continuously, overflowing at “FFH”, and then continues incrementing from “00H”. The timer 0/1/2 overflow interrupt is generated whenever a counter overflow occurs. TnCNT value is cleared by software (TnCC) bit.

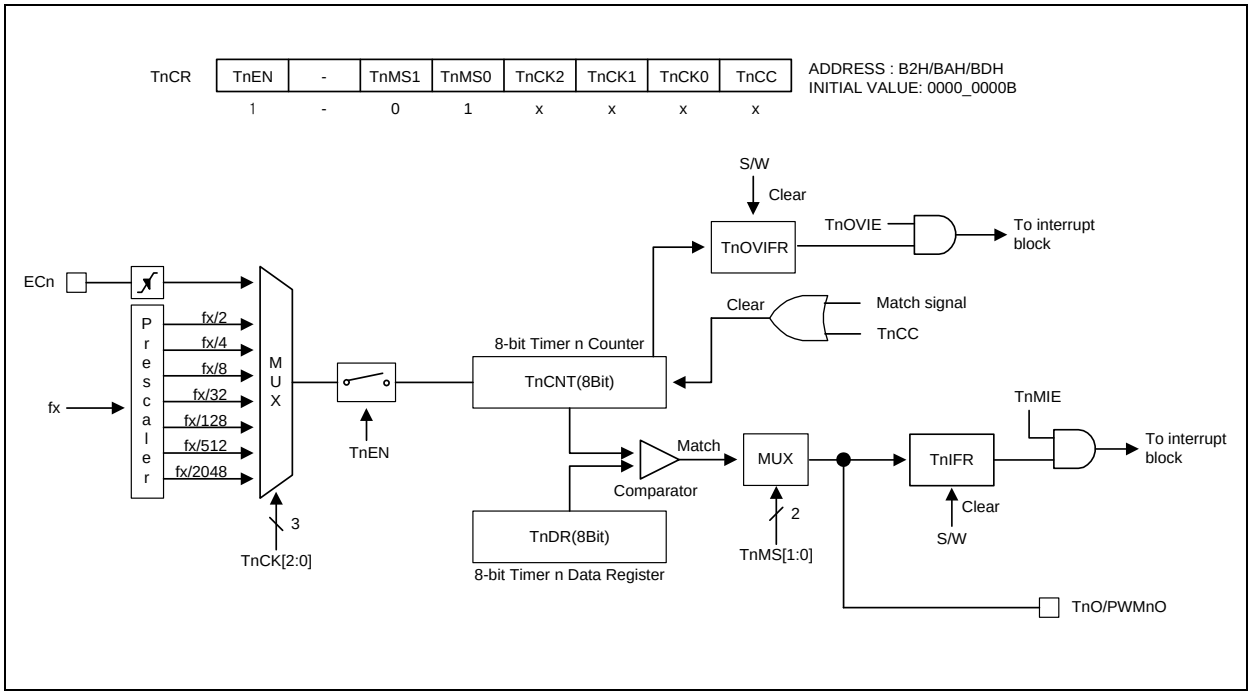



**Figure 11.9 8-Bit PWM Mode for Timer 0/1/2 (Where n = 0, 1, and 2)**

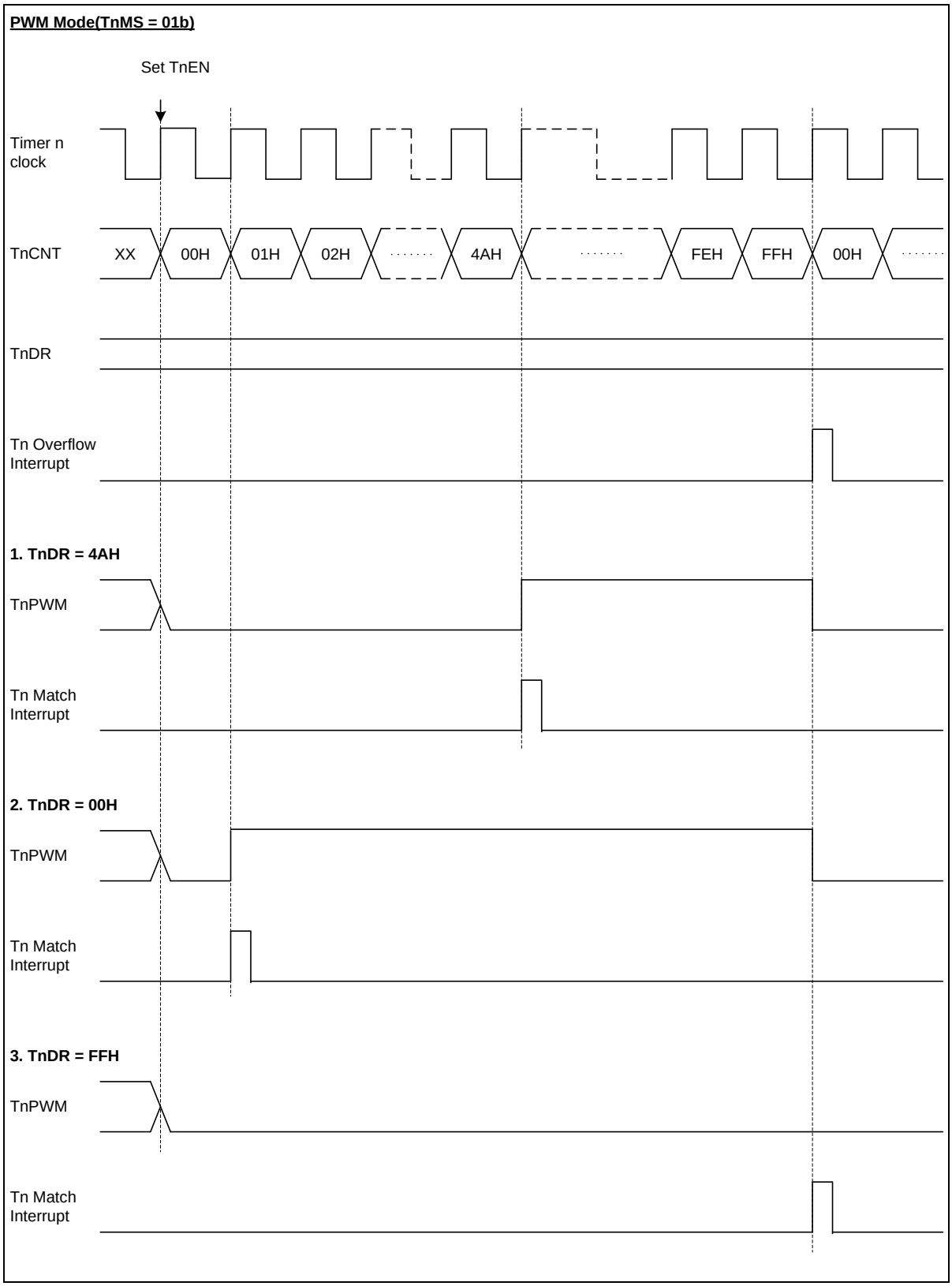



**Figure 11.10 PWM Output Waveforms in PWM Mode for Timer 0/1/2 (Where n = 0, 1, and 2)**

### 11.5.4 8-Bit Capture Mode

The timer 0/1/2 capture mode is set by TnMS[1:0] as '1x'. The clock source can use the internal/external clock. Basically, it has the same function as the 8-bit timer/counter mode and the interrupt occurs when TnCNT is equal to TnDR. TnCNT value is automatically cleared by match signal and it can be also cleared by software (TnCC).

This timer interrupt in capture mode is very useful when the pulse width of captured signal is wider than the maximum period of timer.

The capture result is loaded into TnCDR. In the timer n capture mode, timer n output (TnO) waveform is not available.

According to EIPOL2H/L registers setting, the external interrupt EINT1n function is chosen. Of course, the EINT1n pin must be set to an input port.

TnCDR and TnDR are in the same address. In the capture mode, reading operation reads TnCDR, not TnDR and writing operation will update TnDR.

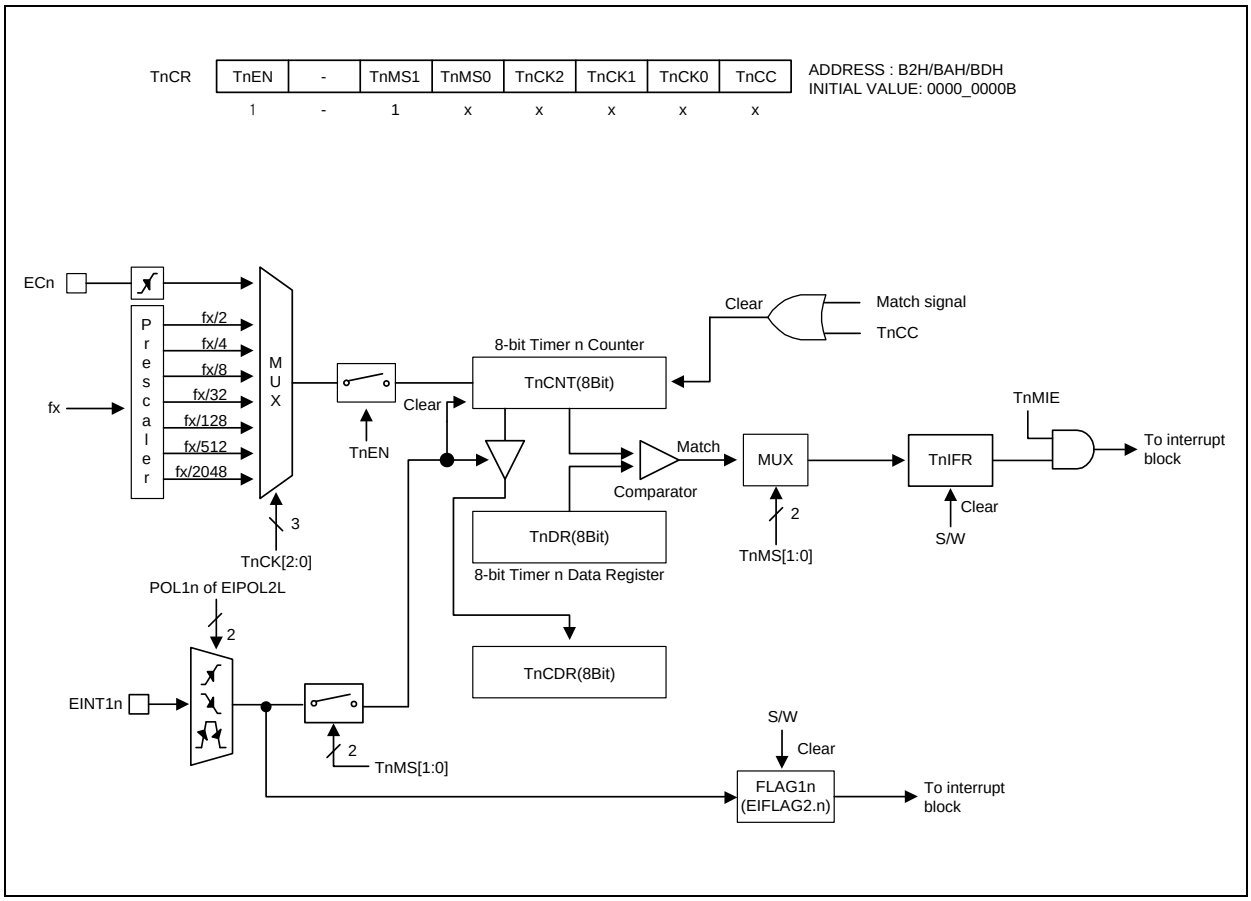



**Figure 11.11 8-Bit Capture Mode for Timer 0/1/2 (Where n = 0, 1, and 2)**

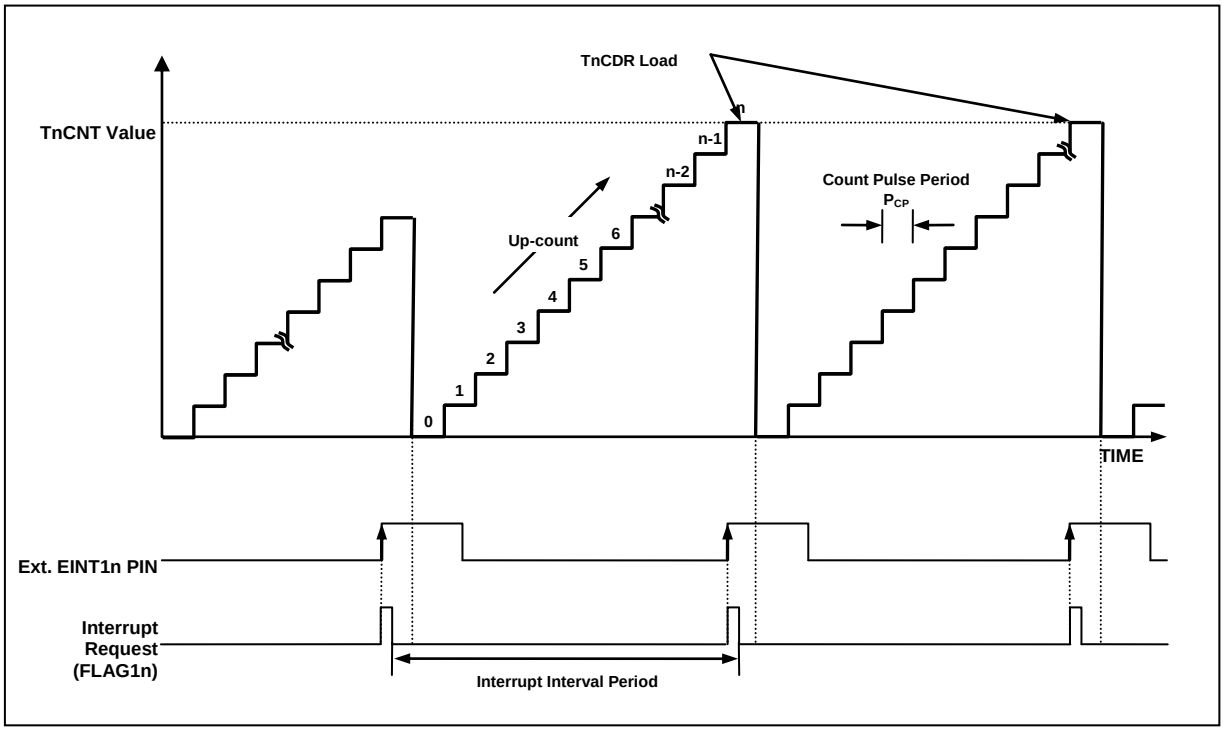



**Figure 11.12 Input Capture Mode Operation for Timer 0/1/2 (Where n = 0, 1, and 2)**

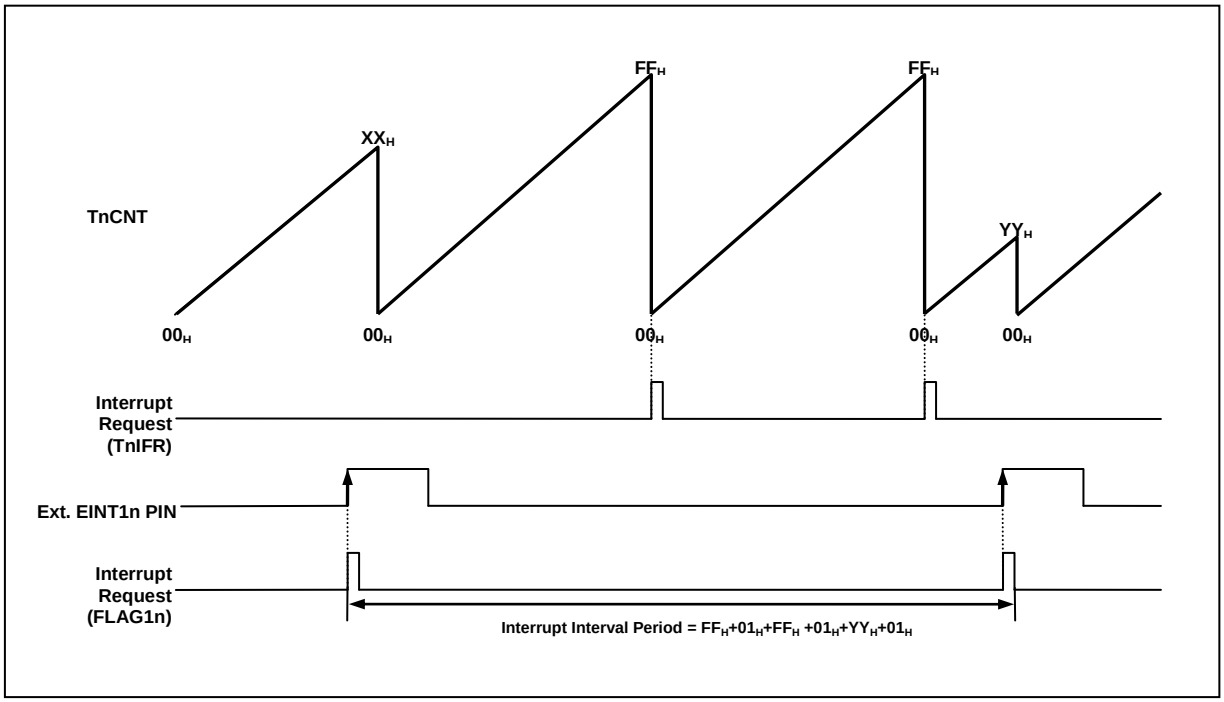



**Figure 11.13 Express Timer Overflow in Capture Mode (Where n = 0, 1, and 2)**

### 11.5.5 Block Diagram

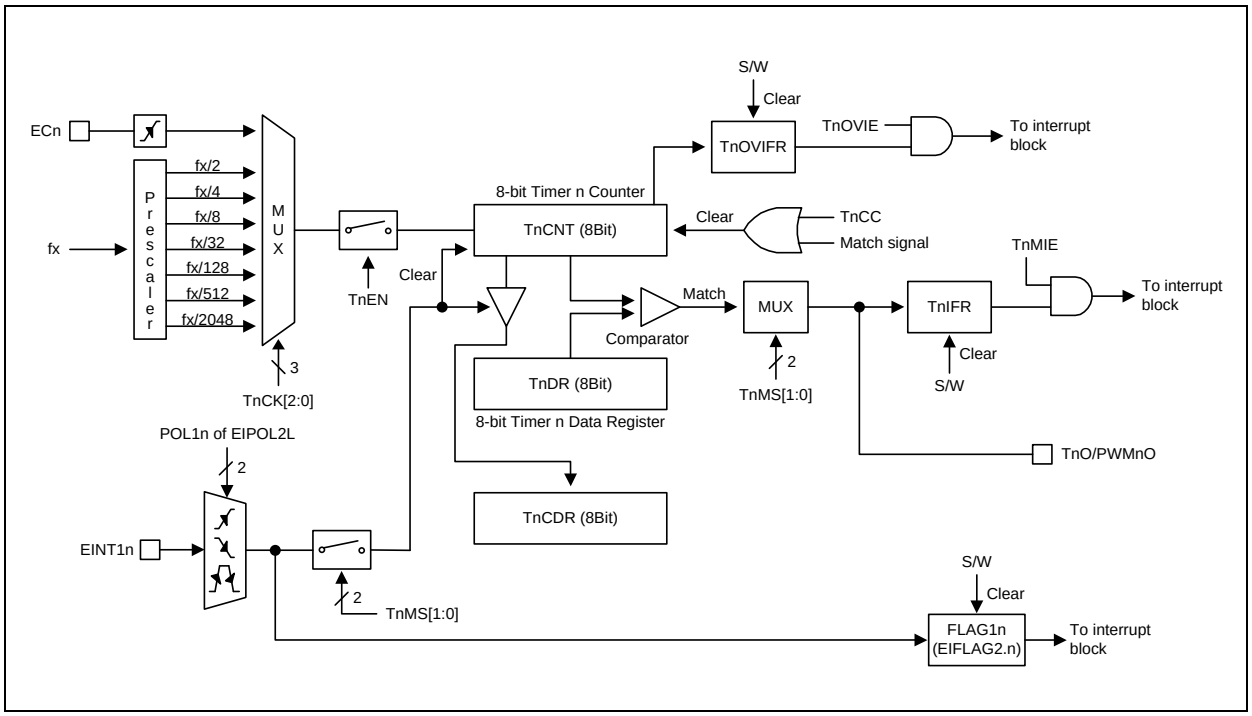



**Figure 11.14 8-Bit Timer 0/1/2 Block Diagram (Where n = 0, 1, and 2)**

### 11.5.6 Register Map

**Table 11-6 Timer 0/1/2 Register Map**

| Name | Address | Dir | Default | Description |
|---|---|---|---|---|
| TnCNT | B3H/BBH/BEH | R | 00H | Timer n Counter Register |
| TnDR | B4H/BCH/BFH | R/W | FFH | Timer n Data Register |
| TnCDR | B4H/BCH/BFH | R | 00H | Timer n Capture Data Register |
| TnCR | B2H/BAH/BDH | R/W | 00H | Timer n Control Register |
| TINTCR | ECH | R/W | 00H | Timer Interrupt Control Register |
| TIFLAG | ACH | R/W | 00H | Timer Interrupt Flag Register |

### 11.5.6.1 Timer/Counter 0/1/2 Register Description

The timer/counter 0/1/2 register consists of timer 0/1/2 counter register (TnCNT), timer 0/1/2 data register (TnDR), timer 0/1/2 capture data register (TnCDR), timer 0/1/2 control register (TnCR), timer interrupt control register(TINTCR), and timer interrupt flag register(TIFLAG).

### 11.5.6.2 Register Description for Timer/Counter 0/1/2

**TnCNT (Timer n Counter Register) : B3H/BBH/BEH, Where n = 0, 1, and 2**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TnCNT7 | TnCNT6 | TnCNT5 | TnCNT4 | TnCNT3 | TnCNT2 | TnCNT1 | TnCNT0 |
| R | R | R | R | R | R | R | R |

Initial value : 00H

**TnCNT[7:0]** Tn Counter

**TnDR (Timer n Data Register : Write only when it is capture mode) : B4H/BCH/BFH, Where n = 0, 1, and 2**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TnDR7 | TnDR6 | TnDR5 | TnDR4 | TnDR3 | TnDR2 | TnDR1 | TnDR0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : FFH

**TnDR[7:0]** Tn Data

**TnCDR (Timer n Capture Data Register : Capture mode only) : B4H/BCH/BFH, Where n = 0, 1, and 2**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TnCDR7 | TnCDR6 | TnCDR5 | TnCDR4 | TnCDR3 | TnCDR2 | TnCDR1 | TnCDR0 |
| R | R | R | R | R | R | R | R |

Initial value : 00H

**TnCDR[7:0]** Tn Capture Data

**TnCR (Timer n Control Register) : B2H/BAH/BDH, Where n = 0, 1, and 2**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TnEN | – | TnMS1 | TnMS0 | TnCK2 | TnCK1 | TnCK0 | TnCC |
| R/W | – | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**TnEN** Control Timer n

| | |
|---|---|
| 0 | Timer n disable |
| 1 | Timer n enable |

**TnMS[1:0]** Control Timer n Operation Mode

| TnMS1 | TnMS0 | Description |
|---|---|---|
| 0 | 0 | Timer/counter mode |
| 0 | 1 | PWM mode |
| 1 | x | Capture mode |

**TnCK[2:0]** Select Timer n clock source. fx is a system clock frequency

| TnCK2 | TnCK1 | TnCK0 | Description |
|---|---|---|---|
| 0 | 0 | 0 | fx/2 |
| 0 | 0 | 1 | fx/4 |
| 0 | 1 | 0 | fx/8 |
| 0 | 1 | 1 | fx/32 |
| 1 | 0 | 0 | fx/128 |
| 1 | 0 | 1 | fx/512 |
| 1 | 1 | 0 | fx/2048 |
| 1 | 1 | 1 | External Clock (ECn) |

**TnCC** Clear timer n Counter

| | |
|---|---|
| 0 | No effect |
| 1 | Clear the Timer n counter (When write, automatically cleared “0” after being cleared counter) |

NOTES) 1. Match Interrupt is generated in Capture mode.

2. Refer to the timer interrupt flag register (TIFLAG) for the T0/1/2 interrupt flags.

**TINTCR(Timer Interrupt Control Register) : ECH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | T2MIE | T1MIE | T0MIE | T2OVIE | T1OVIE | T0OVIE |
| – | – | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**T2MIE** Enable or Disable Timer 2 Match Interrupt
0 Disable
1 Enable

**T1MIE** Enable or Disable Timer 1 Match Interrupt
0 Disable
1 Enable

**T0MIE** Enable or Disable Timer 0 Match Interrupt
0 Disable
1 Enable

**T2OVIE** Enable or Disable Timer 2 Overflow Interrupt
0 Disable
1 Enable

**T1OVIE** Enable or Disable Timer 1 Overflow Interrupt
0 Disable
1 Enable

**T0OVIE** Enable or Disable Timer 0 Overflow Interrupt
0 Disable
1 Enable

**TIFLAG(Timer Interrupt Flag Register) : ACH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | T2OVIFR | T2IFR | T1OVIFR | T1IFR | T0OVIFR | T0IFR |
| – | – | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**T2OVIFR** When T2 overflow interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software.

0 Disable

1 Enable

**T1IFR** When T2 match interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software.

0 Disable

1 Enable

**T0OVIFR** When T1 overflow interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software.

0 Disable

1 Enable

**T2IFR** When T1 match interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software.

0 Disable

1 Enable

**T1OVIFR** When T0 overflow interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software.

0 Disable

1 Enable

**T0IFR** When T0 match interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software.

0 Disable

1 Enable

## 11.6 Timer 3/4/5/6

### 11.6.1.1 Overview

The 16-bit timer 3/4/5/6 consists of multiplexer, timer 3/4/5/6 A data register high/low, timer 3/4/5/6 B data register high/low, and timer 3/4/5/6 control register high/low (TnADRH, TnADRL, TnBDRH, TnBDRL, TnCRH, TnCRL).

It has four operating modes:

- 16-bit timer/counter mode
- 16-bit capture mode
- 16-bit PPG output mode (one-shot mode)
- 16-bit PPG output mode (repeat mode)

The timer/counter 3/4/5/6 can be clocked by an internal or an external clock source (ECn). The clock source is selected by clock selection logic which is controlled by the clock selection bits (TnCK[2:0]).

- TIMER 3/4/5/6 clock source: $f_X$/1, 2, 4, 8, 32, 128, 512 and ECn

In the capture mode, by EINT1n, the data is captured into input capture data register (TnBDRH/TnBDRL). Timer 3/4/5/6 outputs the comparision result between counter and data register through TnO port in timer/counter mode. Also Timer 3/4/5/6 outputs PWM wave form through PWMnO port in the PPG mode.

**Table 11-7 Timer 3/4/5/6 Operating Modes**

| TnEN | TnMS[1:0] | TnCK[2:0] | Timer n |
|---|---|---|---|
| 1 | 00 | XXX | 16 Bit Timer/Counter Mode |
| 1 | 01 | XXX | 16 Bit Capture Mode |
| 1 | 10 | XXX | 16 Bit PPG Mode (one-shot mode) |
| 1 | 11 | XXX | 16 Bit PPG Mode (repeat mode) |

## 11.6.2 16-Bit Timer/Counter Mode

The 16-bit timer/counter mode is selected by control register as shown in Figure 11.15.

The 16-bit timer have counter and data register. The counter register is increased by internal or external clock input. Timer 3/4/5/6 can use the input clock with one of 1, 2, 4, 8, 32, 128, and 512 prescaler division rates (TnCK[2:0]). When the value of TnCNTH, TnCNTL and the value of TnADRH, TnADRL are identical in Timer 3/4/5/6 respectively, a match signal is generated and the interrupt of Timer 3/4/5/6 occurs. The TnCNTH, TnCNTL value is automatically cleared by match signal. It can be also cleared by software (TnCC).

The external clock (ECn) counts up the timer at the rising edge. If the ECn is selected as a clock source by TnCK[2:0], EC3/EC4/EC5/EC6 port should be set to the input port by P26IO/P27IO/P14IO/P15IO bit.

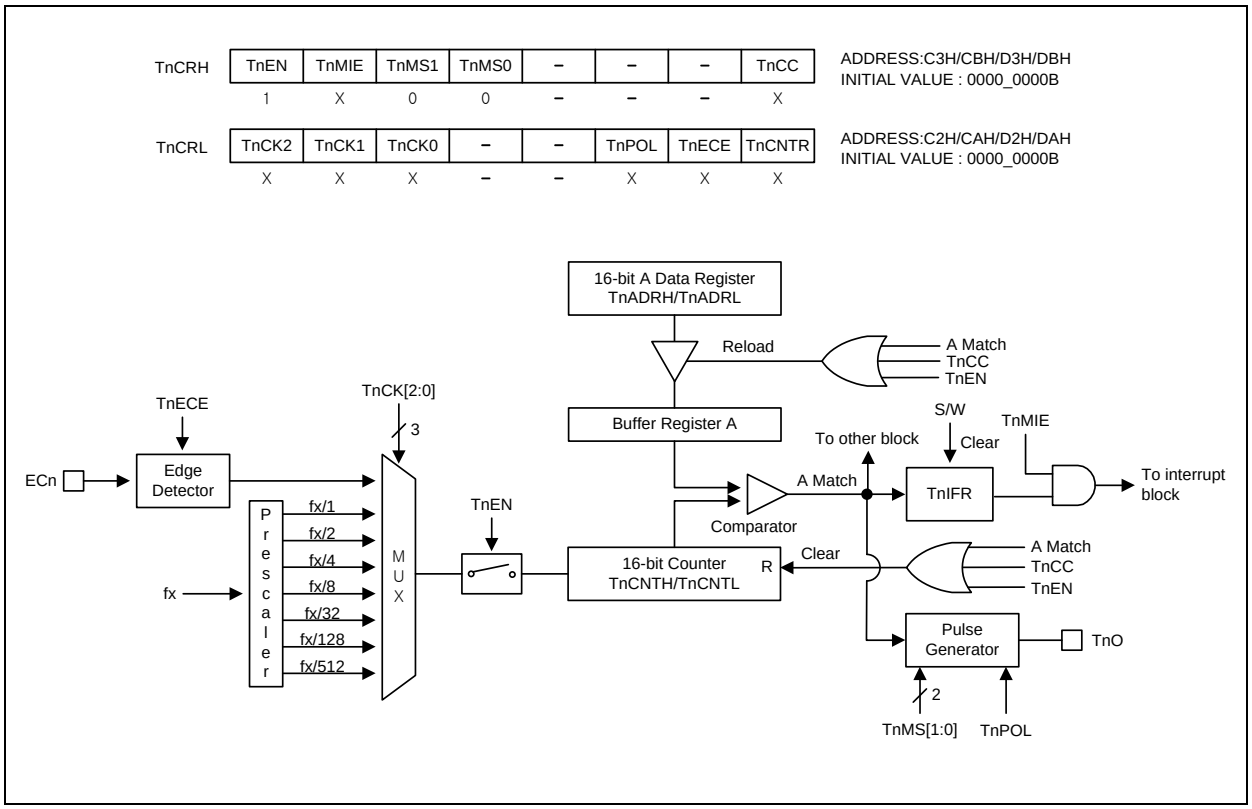



**Figure 11.15 16-Bit Timer/Counter Mode for Timer 3/4/5/6 (where n = 3,4,5, and 6)**

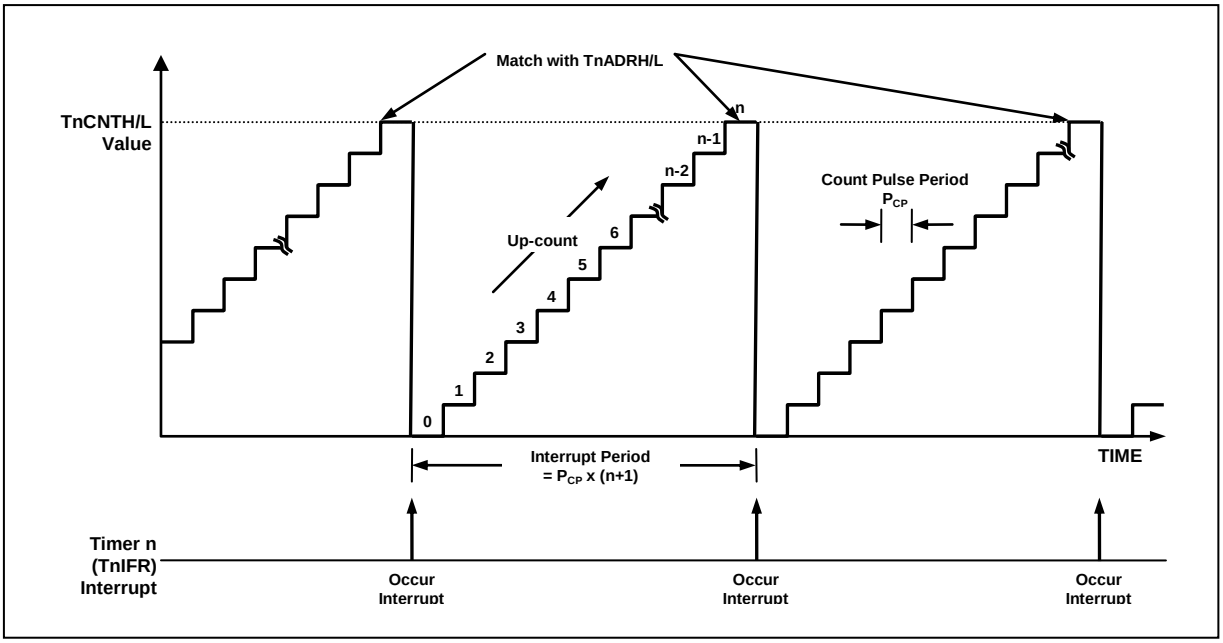



**Figure 11.16 16-Bit Timer/Counter 3/4/5/6 Example (where n = 3,4,5, and 6)**

### 11.6.3 16-Bit Capture Mode

The 16-bit Timer 3/4/5/6 capture mode is set by TnMS[1:0] as '01'. The clock source can use the internal/external clock. Basically, it has the same function as the 16-bit timer/counter mode and the interrupt occurs when TnCNTH/TnCNTL is equal to TnADRH/TnADRL. The TnCNTH, TnCNTL values are automatically cleared by match signal. It can be also cleared by software (TnCC).

This timer interrupt in capture mode is very useful when the pulse width of captured signal is wider than the maximum period of timer.

The capture result is loaded into TnBDRH/TnBDRL.

According to EIPOL2L/H registers setting, the external interrupt EINT1n function is chosen. Of course, the EINT1n pin must be set as an input port.

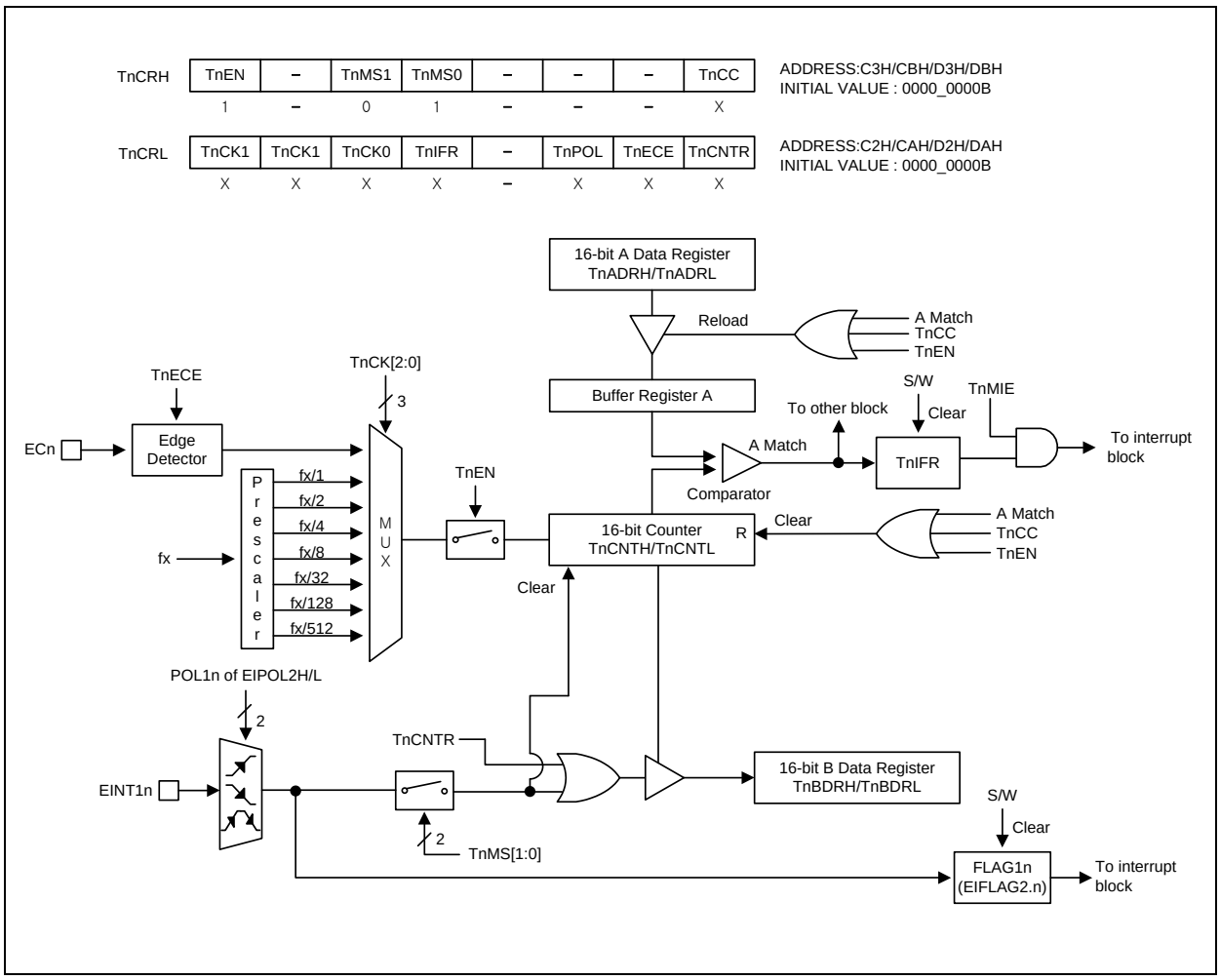



**Figure 11.17 16-Bit Capture Mode for Timer 3/4/5/6 (where n = 3,4,5, and 6)**

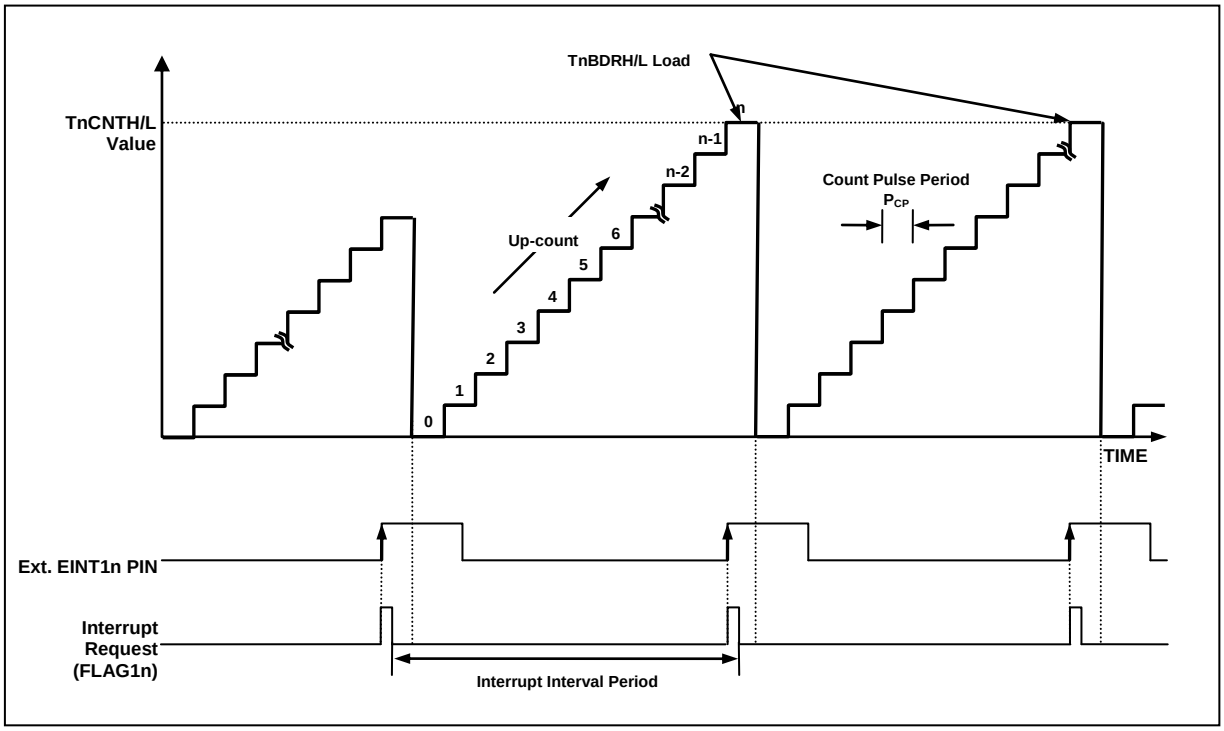



**Figure 11.18 Input Capture Mode Operation for Timer 3/4/5/6 (where n = 3,4,5, and 6)**

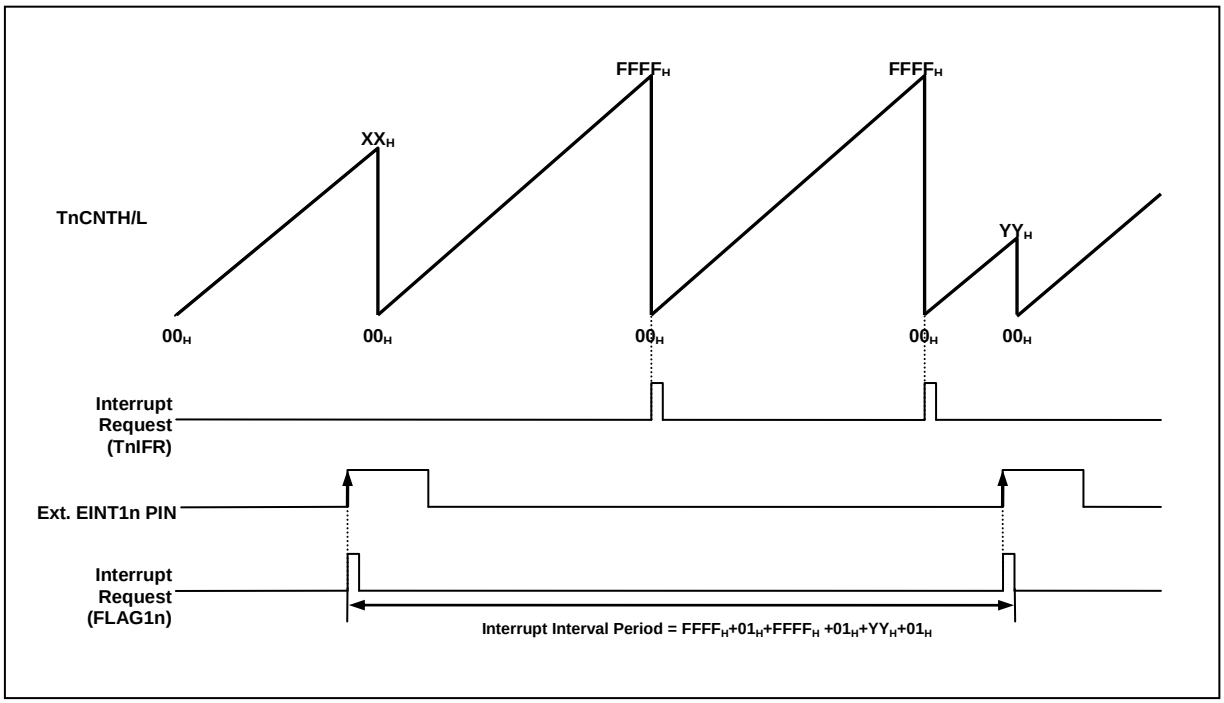
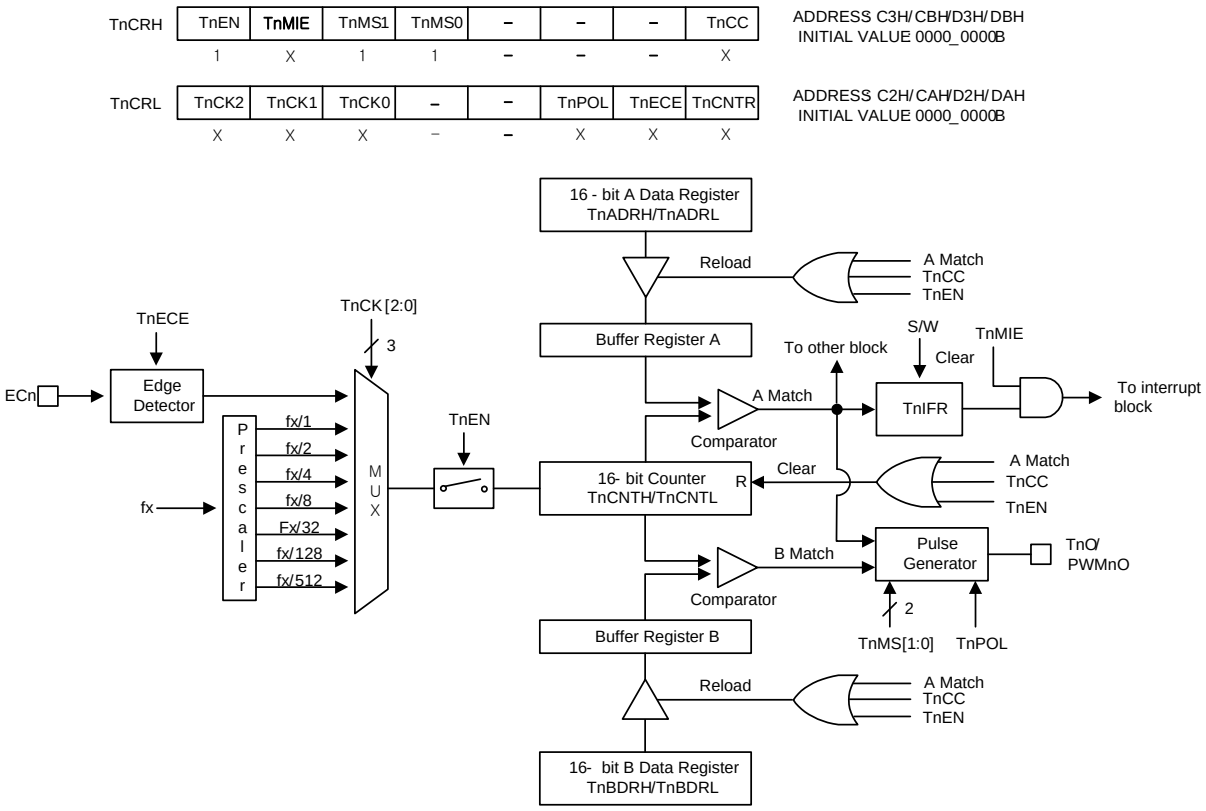



**Figure 11.19 Express Timer Overflow in Capture Mode (where n = 3,4,5, and 6)**

### 11.6.4 16-Bit PPG Mode

The Timer 3/4/5/6 has a PPG (Programmable Pulse Generation) function. In PPG mode, T3O/PWM3O, T4O/PWM4O, T5O/PWM5O, and T6O/PWM6O pin outputs up to 16-bit resolution PWM output. This pin should be configured as a PWM output by setting P1FSR[1:0] to '01', P1FSR[2] to '1', P1FSR[3] to '1', and P1FSR[4] to '1'. The period of the PWM output is determined by the TnADRH/TnADRL. And the duty of the PWM output is determined by the TnBDRH/TnBDRL.




NOTE) The TnEN is automatically cleared to logic "0" after one pulse is generated at a PPG one-shot mode.

**Figure 11.20 16-Bit PPG Mode for Timer 3/4/5/6 (where n = 3,4,5, and 6)**

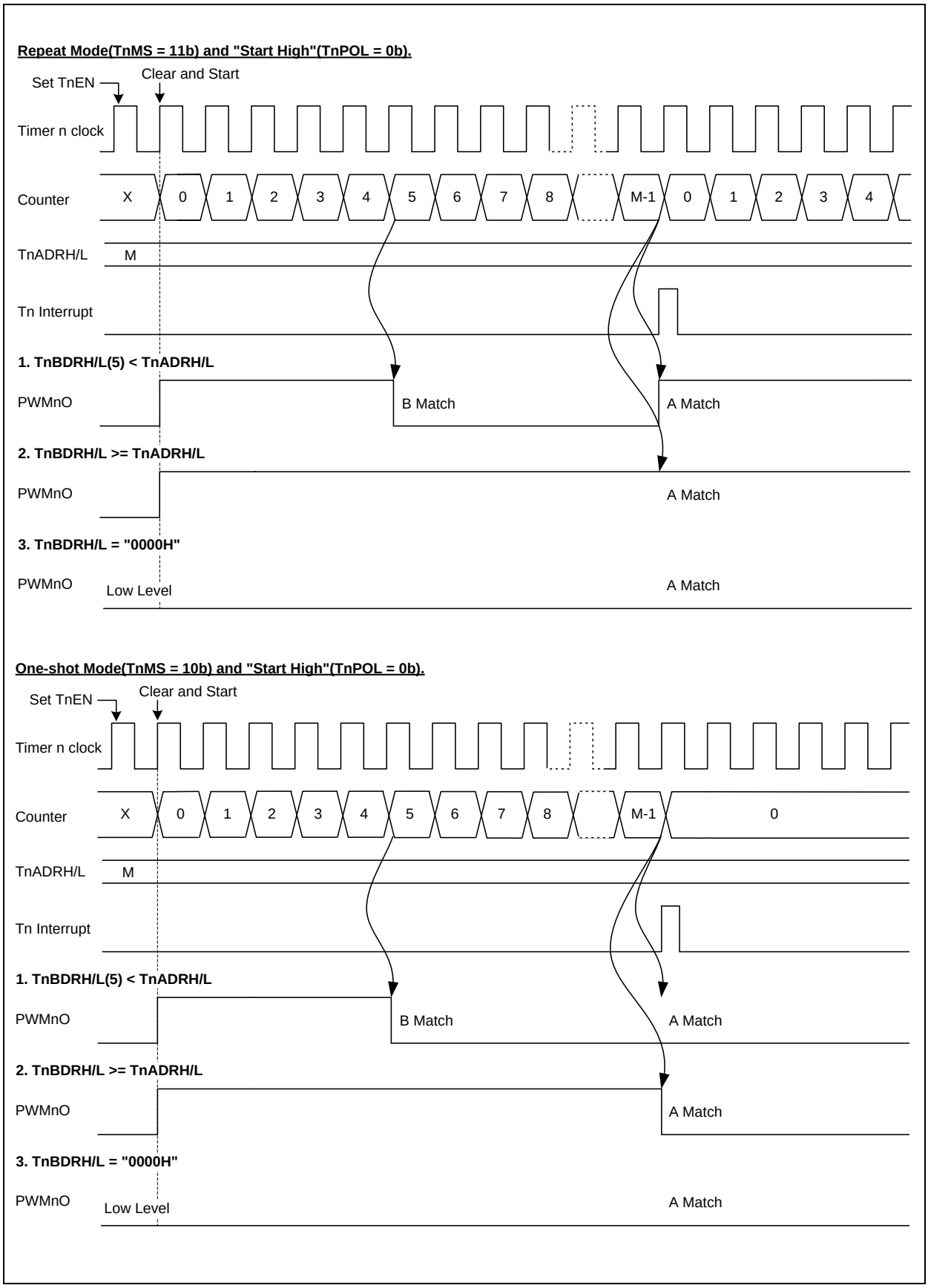



**Figure 11.21 16-Bit PPG Mode Timming chart for Timer 3/4/5/6 (where n = 3,4,5, and 6)**

## 11.6.5 Block Diagram

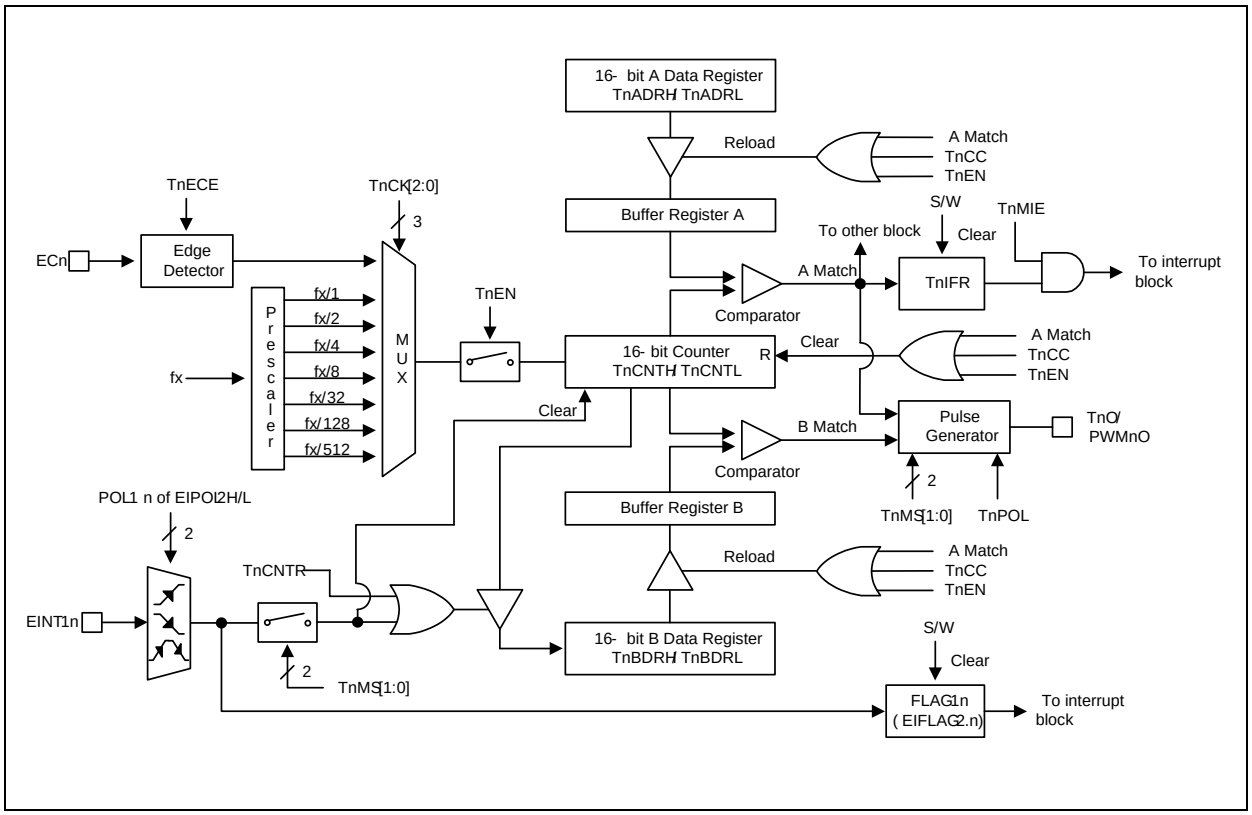



**Figure 11.22 16-Bit Timer 3/4/5/6 Block Diagram (where n = 3,4,5, and 6)**

## 11.6.6 Register Map

**Table 11-8 Timer 3/4/5/6 Register Map (where n = 3,4,5, and 6)**

| Name | Address | Dir | Default | Description |
|---|---|---|---|---|
| TnADRH | C5H/CDH/D5H/DDH | R/W | FFH | Timer n A Data High Register |
| TnADRL | C4H/CCH/D4H/DCH | R/W | FFH | Timer n A Data Low Register |
| TnBDRH | C7H/CFH/D7H/DFH | R/W | FFH | Timer n B Data High Register |
| TnBDRL | C6H/CEH/D6H/DEH | R/W | FFH | Timer n B Data Low Register |
| TnCRH | C3H/CBH/D3H/DBH | R/W | 00H | Timer n Control High Register |
| TnCRL | C2H/CAH/D2H/DAH | R/W | 00H | Timer n Control Low Register |

### 11.6.6.1 Timer/Counter 3/4/5/6 Register Description

The timer/counter 3/4/5/6 register consists of Timer 3/4/5/6 A data high register (TnADRH), Timer 3/4/5/6 A data low register (TnADRL), Timer 3/4/5/6 B data high register (TnBDRH), Timer 3/4/5/6 B data low register (TnBDRL), Timer 3/4/5/6 control High register (TnCRH) and Timer 3/4/5/6 control low register (TnCRL).

### 11.6.6.2 Register Description for Timer/Counter 3/4/5/6

**TnADRH (Timer n A data High Register) : C5H/CDH/D5H/DDH (where n = 3,4,5, and 6)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TnADRH7 | TnADRH6 | TnADRH5 | TnADRH4 | TnADRH3 | TnADRH2 | TnADRH1 | TnADRH0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : FFH

**TnADRH[7:0]** Tn A Data High Byte

**TnADRL (Timer n A Data Low Register) : C4H/CCH/D4H/DCH (where n = 3,4,5, and 6)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TnADRL7 | TnADRL6 | TnADRL5 | TnADRL4 | TnADRL3 | TnADRL2 | TnADRL1 | TnADRL0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : FFH

**TnADRL[7:0]** Tn A Data Low Byte

NOTE) Do not write “0000H” in the TnADRH/TnADRL register when PPG mode

**TnBDRH (Timer n B Data High Register) : C7H/CFH/D7H/DFH (where n = 3,4,5, and 6)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TnBDRH7 | TnBDRH6 | TnBDRH5 | TnBDRH4 | TnBDRH3 | TnBDRH2 | TnBDRH1 | TnBDRH0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : FFH

**TnBDRH[7:0]** Tn B Data High Byte

**TnBDRL (Timer n B Data Low Register) : C6H/CEH/D6H/DEH (where n = 3,4,5, and 6)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TnBDRL7 | TnBDRL6 | TnBDRL5 | TnBDRL4 | TnBDRL3 | TnBDRL2 | TnBDRL1 | TnBDRL0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : FFH

**TnBDRL[7:0]** Tn B Data Low Byte

**TnCRH (Timer n Control High Register) : C3H/CBH/D3H/DBH (where n = 3,4,5, and 6)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TnEN | TnMIE | TnMS1 | TnMS0 | – | – | – | TnCC |
| R/W | R/W | R/W | R/W | – | – | – | R/W |

Initial value : 00H

**TnEN** Control Timer n

0 Timer n disable

1 Timer n enable (Counter clear and start)

**TnMIE** Enable or Disable Timer n Match Interrupt

0 Disable

1 Enable

**TnMS[1:0]** Control Timer n Operation Mode

| TnMS1 | TnMS0 | Description |
|---|---|---|
| 0 | 0 | Timer/counter mode (TnO: toggle at A match) |
| 0 | 1 | Capture mode (The A match interrupt can occur) |
| 1 | 0 | PPG one-shot mode (PWMnO) |
| 1 | 1 | PPG repeat mode (PWMnO) |

**TnCC** Clear Timer n Counter

0 No effect

1 Clear the Timer n counter (When write, automatically cleared “0” after being cleared counter)

NOTES) Refer to the external interrupt flag 1 register (EIFLAG1) for the T6/T5/T4/T3 interrupt flags.

**TnCRL (Timer n Control Low Register) : C2H/CAH/D2H/DAH (where n = 3,4,5, and 6)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TnCK2 | TnCK1 | TnCK0 | – | – | TnPOL | TnECE | TnCNTR |
| R/W | R/W | R/W | – | – | R/W | R/W | R/W |

Initial value : 00H

**TnCK[2:0]** Select Timer n clock source. fx is main system clock frequency

| TnCK2 | TnCK1 | TnCK0 | Description |
|---|---|---|---|
| 0 | 0 | 0 | fx/512 |
| 0 | 0 | 1 | fx/128 |
| 0 | 1 | 0 | fx/32 |
| 0 | 1 | 1 | fx/8 |
| 1 | 0 | 0 | fx/4 |
| 1 | 0 | 1 | fx/2 |
| 1 | 1 | 0 | fx/1 |
| 1 | 1 | 1 | External clock (ECn) |

**TnPOL** TnO/PWMnO Polarity Selection

0 Start High (TnO/PWMnO is low level at disable)

1 Start Low (TnO/PWMnO is high level at disable)

**TnECE** Timer n External Clock Edge Selection

0 External clock falling edge

1 External clock rising edge

**TnCNTR** Timer n Counter Read Control

0 No effect

1 Load the counter value to the B data register (When write, automatically cleared “0” after being loaded)

## 11.7 Timer 7/8

### 11.7.1 Overview

Timer 7 and timer 8 can be used either two 8-bit timer/counter or one 16-bit timer/counter with combine them. Each 8-bit timer/event counter module has multiplexer, comparator, 8-bit timer data register, 8-bit counter register, control register and capture data register (T7CNT, T7DR, T7CAPR, T7CR, T8CNT, T8DR, T8CAPR, T8CR). For PWM, it has PWM register (T8PPRL. T8PPRH, T8ADRL, T8ADRH, T8BDRL, T8BDRH, T8CDRL, T8CDRH, T8DLYA, T8DLYB, T8DLYC).

It has five operating modes:

- 8-bit timer/counter mode
- 8-bit capture mode
- 16-bit timer/counter mode
- 16-bit capture mode
- 10-bit PWM mode

The timer/counter 7 and 8 can be clocked by an internal or an external clock source (EC7). The clock source is selected by clock selection logic which is controlled by the clock selection bits (T7CK[2:0], T8CK[3:0]). Also the timer/counter 8 can use more clock sources than timer/counter 7.

- TIMER 7 clock source: $f_X$/2, 4, 8, 32, 128, 512, 2048 and EC7
- TIMER 8 clock source: $f_X$/1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384 and T7 clock

In the capture mode, by EINT17/EINT18, the data is captured into input capture data register (T7CAPR, T8CAPR).

In 8-bit timer/counter 7/8 mode, whenever counter value is equal to T7DR/T8DR, T7O/T8O port toggles. Also In 16-bit timer/counter 7 mode,

The timer 7 outputs the comparison result between counter and data register through T7O port. The PWM wave form to PWMAA, PWMAB, PWMBA, PWMBB, PWMCA, PWMCB Port (6-channel) in the PWM mode.

**Table 11-9 Timer 7/8 Operating Modes**

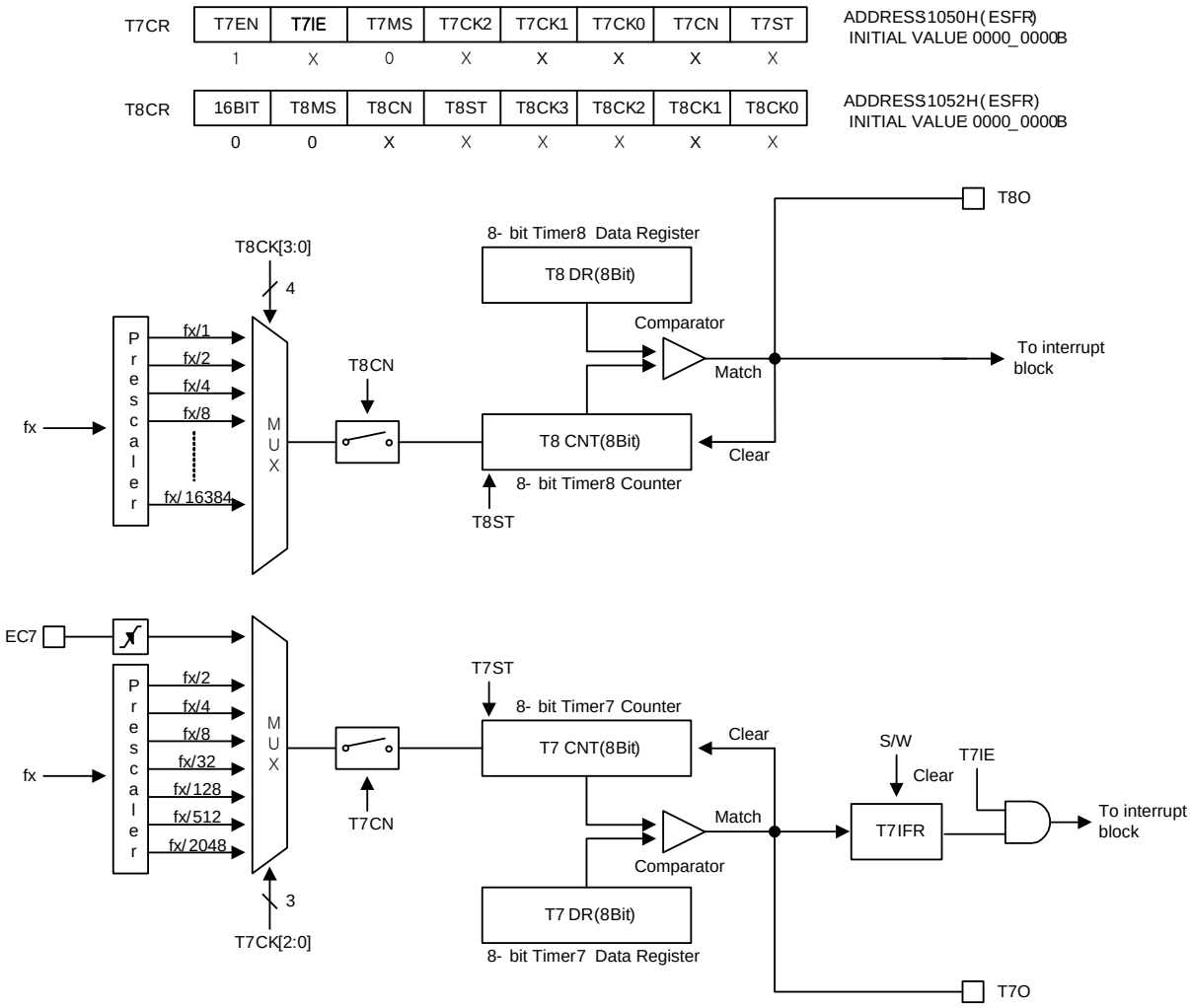
| 16BIT | T7MS | T8MS | PWM8E | T7CK[2:0] | T7CK[3:0] | Timer 7 | Timer 8 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | XXX | XXXX | 8 Bit Timer/Counter Mode | 8 Bit Timer/Counter Mode |
| 0 | 1 | 1 | 0 | XXX | XXXX | 8 Bit Capture Mode | 8 Bit Capture Mode |
| 1 | 0 | 0 | 0 | XXX | XXXX | 16 Bit Tmer/Counter Mode | |
| 1 | 1 | 1 | 0 | XXX | XXXX | 16 Bit Capture Mode | |
| 0 | X | X | 1 | XXX | XXXX | 10 Bit PWM Mode | |

### 11.7.2 8-Bit Timer/Counter 7/8 Mode

The 8-bit timer/counter mode is selected by control register as shown in Figure 11.23.

The two 8-bit timers have each counter and data register. The counter register is increased by internal or external clock input. Timer 7 can use the input clock with one of 2, 4, 8, 32, 128, 512, 2048 and EC7 prescaler division rates (T7CK[2:0]). Timer 8 can use the input clock with one of 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384 and timer 7 clock prescaler division rates (T8CK[3:0]). When the value of T7CNT, T8CNT and T7DR, T8DR are respectively identical in Timer 7/8, the interrupt Timer 7/8 occurs.

The external clock (EC7) counts up the timer at the rising edge. If the EC7 is selected as a clock source by T7CK[2:0], EC7 port should be set to the input port by P63IO bit. Timer 8 can't use the external EC7 clock.




NOTE) Do not set to "1111b" in the T8CK[3:0], when two 8-bit timer 7/8 modes.

**Figure 11.23 8-Bit Timer/Counter Mode for Timer 7/8**

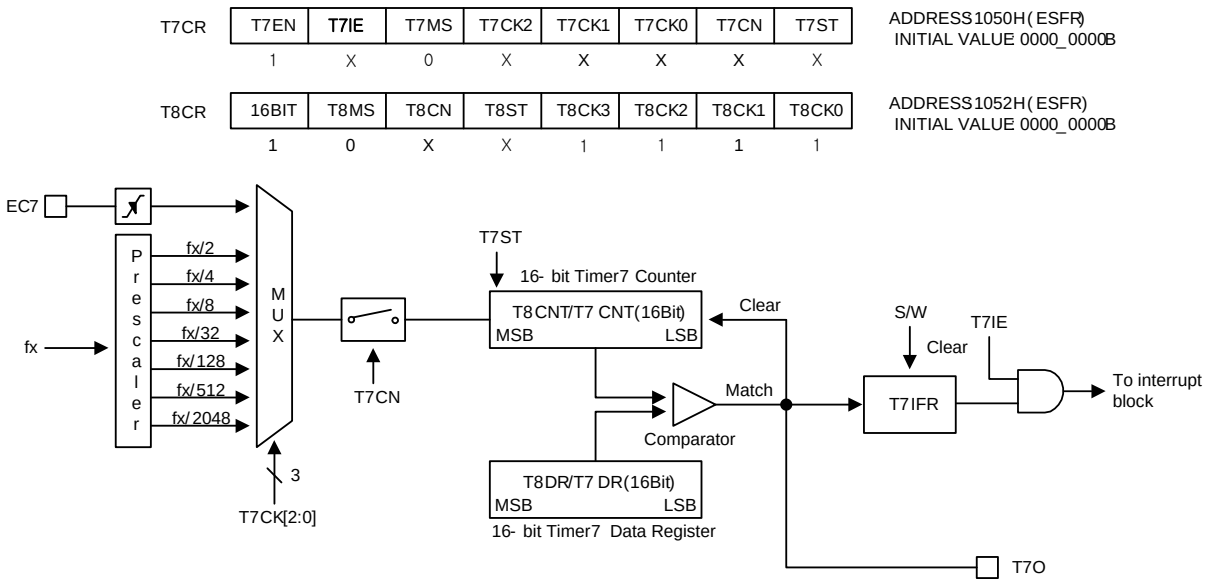
### 11.7.3 16-Bit Timer/Counter 7 Mode

The 16-bit timer/counter mode is selected by control register as shown in Figure 11.24.

The 16-bit timer have counter and data register. The counter register is increased by internal or external clock input. Timer 7 can use the input clock with one of 2, 4, 8, 32, 128, 512 and 2048 prescaler division rates (T7CK[2:0]).

A 16-bit timer/counter register T7CNT, T8CNT are incremented from 0000H to FFFFH until it matches T7DR, T8DR and then cleared to 0000H. The match signal output generates the Timer 7 Interrupt (No timer 8 interrupt). The clock source is selected from T7CK[2:0] and 16BIT bit must be set to '1'. Timer 7 is LSB 8-bit, the timer 8 is MSB 8-bit.

The external clock (EC7) counts up the timer at the rising edge. f the EC7 is selected as a clock source by T7CK[2:0], EC7 port should be set to the input port by P63IO bit.




NOTE) The T8CR.7 bit (16BIT) should be set to '1' and the T8CK[3:0] should be set to "1111b".

**Figure 11.24 16-Bit Timer/Counter Mode for Timer 7**

### 11.7.4 8-Bit Timer 7/8 Capture Mode

The 8-bit Capture 7 and 8 mode is selected by control register as shown in Figure 11.25.

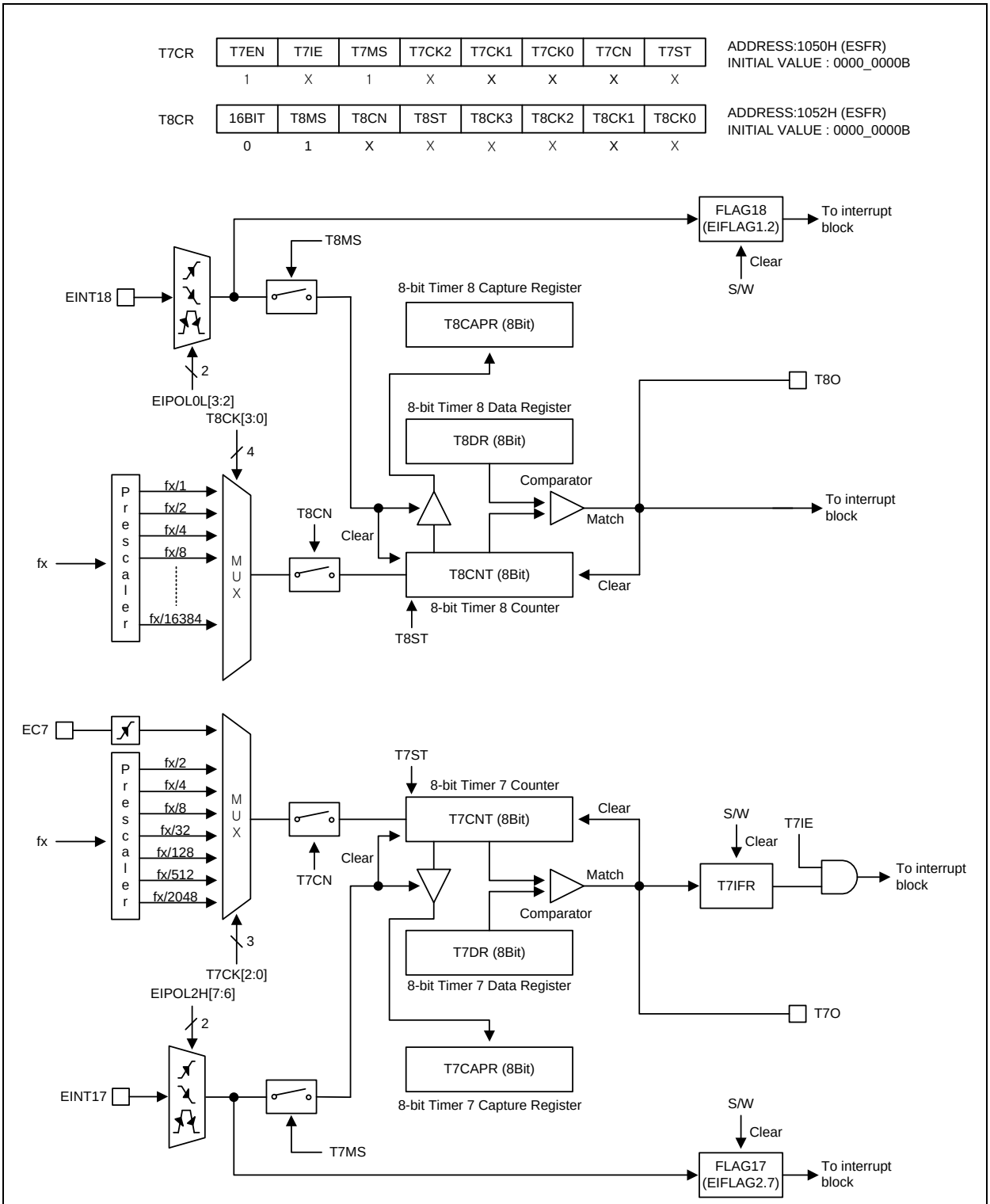
The timer 7/8 capture mode is set by T7MS, T8MS as '1'. The clock source can use the internal/external clock. Basically, it has the same function as the 8-bit timer/counter mode and the interrupt occurs when T7CNT, T8CNT is equal to T7DR, T8DR. The T7CNT, T8CNT value is automatically cleared by match signal.

This timer interrupt in capture mode is very useful when the pulse width of captured signal is wider than the maximum period of timer.

The capture result is loaded into T7CAPR, T8CAPR. In the timer 7/8 capture mode, timer 7/8 output (T7O, T8O) waveform is not available.

According to the EIPOL2H/EIPOL1 register setting, the external interrupt EINT17 and EINT18 function is chose. Of course, the EINT17 and EINT18 pins must be set to an input port.

The T7CAPR and T7DR are in the same address. In the capture mode, reading operation reads T7CAPR, not T7DR and writing operation will update T7DR. The T8CAPR has the same function.



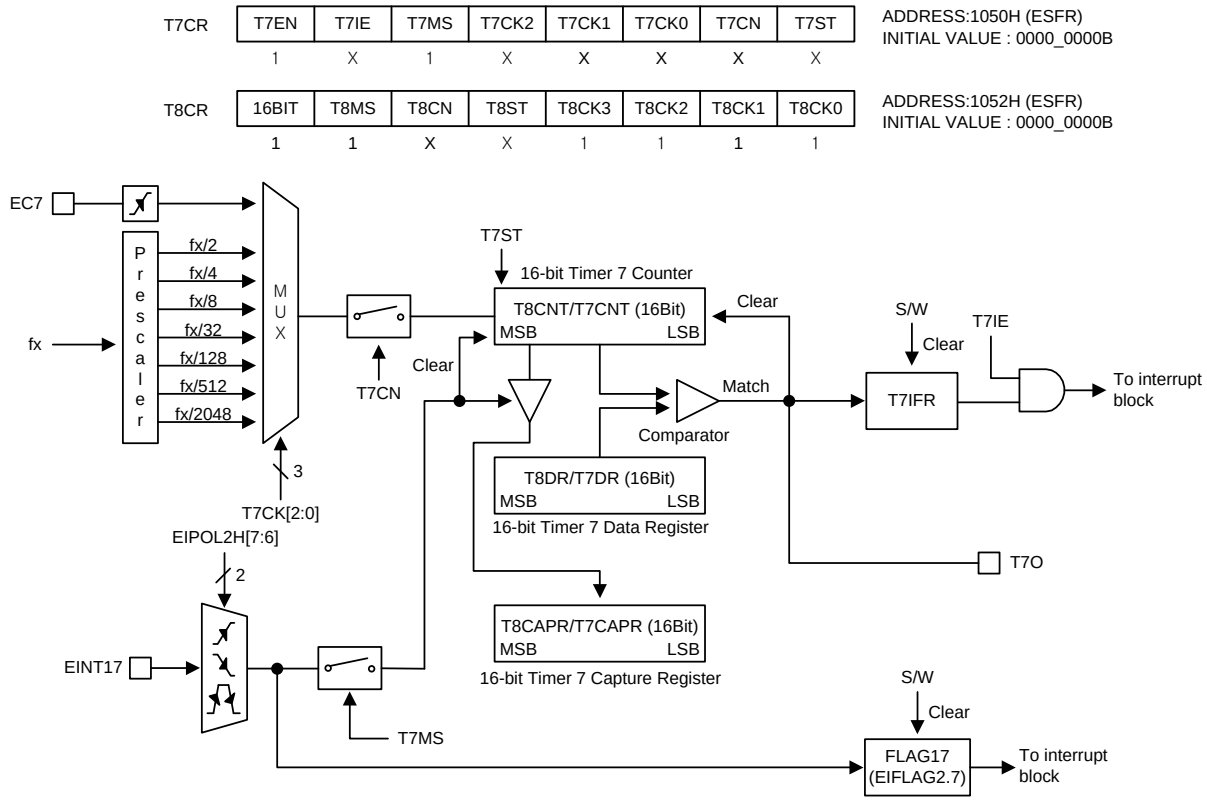

NOTE) Do not set to “1111b” in the T7CK[3:0], when two 8-bit timer 7/8 modes.

**Figure 11.25 8-Bit Capture Mode for Timer 7/8**

### 11.7.5 16-Bit Timer 7 Capture Mode

The 16-bit Capture mode is selected by control register as shown in Figure 11.26.

The 16-bit capture mode is the same operation as 8-bit capture mode, except that the timer register uses 16 bits. The 16-bit timer 7 capture mode is set by T7MS, T8MS as '1'. The clock source is selected from T7CK[2:0] and 16BIT bit must be set to '1'. Timer 7 is LSB 8-bit, the timer 8 is MSB 8-bit.




NOTE) The T8CR.7 bit (16BIT) should be set to '1' and the T8CK[3:0] should be set to "1111b".

**Figure 11.26 16-Bit Capture Mode for Timer 7**

### 11.7.6 10-Bit Timer 8 PWM Mode

The timer 8 has a high speed PWM (Pulse Width Modulation) function. In PWM mode, the 6-channel pins output up to 10-bit resolution PWM output. This pin should be configured as a PWM output by set PWM8E to '1'. When the value of 2bit +T8CNT and T8PPRH/L are identical in timer 8, a period match signal is generated and the interrupt of timer 8 occurs. In 10-bit PWM mode, A, B, C, bottom(underflow) match signal are generated when the 10-bit counter value are identical to the value of T8xDRH/L. The period of the PWM output is determined by the T8PPRH/L (PWM period register), T8xDRH/L (each channel PWM duty register).

PWM Period = [T8PPRH/T8PPRL ] X Source Clock

PWM Duty(A-ch) = [ T8ADRH/T8ADRL ] X Source Clock

**Table 11-10 PWM Frequency vs. Resolution at 8 MHz**

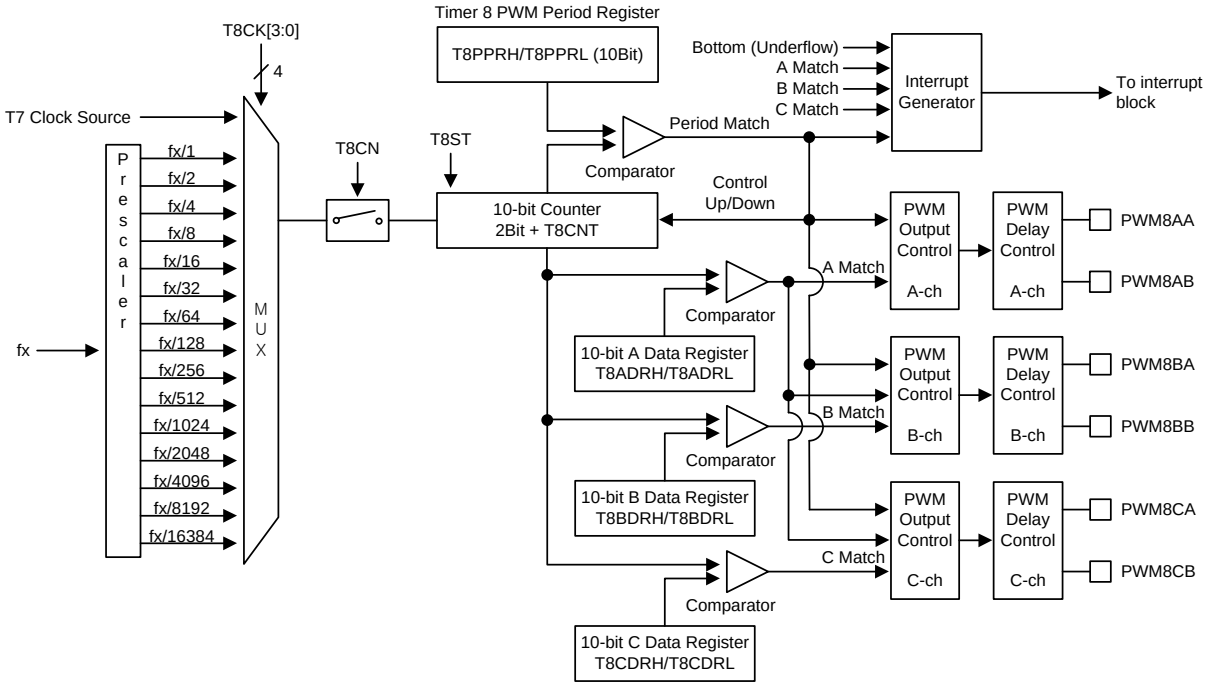
| Resolution | Frequency | | |
|---|---|---|---|
| | T8CK[3:0]=0001 (250ns) | T8CK[3:0]=0010 (500ns) | T8CK[3:0]=0100 (2us) |
| 10 Bit | 3.9KHz | 1.95KHz | 0.49KHz |
| 9 Bit | 7.8KHz | 3.9KHz | 0.98KHz |
| 8 Bit | 15.6KHz | 7.8KHz | 1.95KHz |
| 7 Bit | 31.2KHz | 15.6KHz | 3.91KHz |

The POLxA bit of T8PCR3 register decides the polarity of duty cycle. If the duty value is set same to the period value, the PWM output is determined by the bit POLxA (1: High, 0: Low). And if the duty value is set to "00H", the PWM output is determined by the bit POLxA (1: Low, 0: High).

**Table 11-11 PWM Channel Polarity**

| PHLT:PxxOE | POLxA | POLBO | POLxB | PWM8xA Pin Output | PWM8xB Pin Output |
|---|---|---|---|---|---|
| 0x, x0, 00 | 0 | 0 | 0 | Low-level | Low-level |
| | | | 1 | Low-level | High-level |
| | | 1 | x | Low-level | Low-level |
| | 1 | 0 | 0 | High-level | High-level |
| | | | 1 | High-level | Low-level |
| | | 1 | x | High-level | High-level |
| 11 | 0 | x | 0 | Positive-phase | Positive-Phase |
| | | | 1 | Positive-phase | Negative-Phase |
| | 1 | x | 0 | Negative-Phase | Negative-Phase |
| | | | 1 | Negative-Phase | Positive-phase |

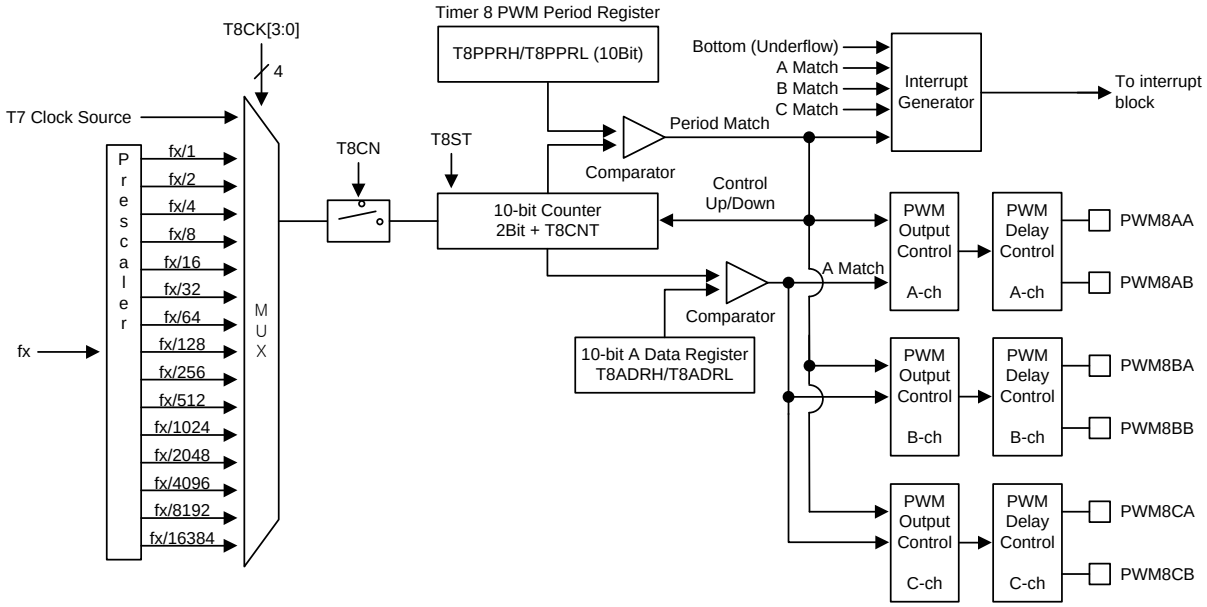
| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| T8CR | 16BIT | T8MS | T8CN | T8ST | T8CK3 | T8CK2 | T8CK1 | T8CK0 | ADDRESS:1052H (ESFR)<br>INITIAL VALUE : 0000_0000B |
| | 0 | X | X | X | X | X | X | X | |
| T8PCR1 | PWM8E | ESYNC | BMOD | PHLT | UPDT | UALL | NOPS1 | NOPS0 | ADDRESS:1053H (ESFR)<br>INITIAL VALUE : 0000_0000B |
| | 1 | X | X | X | X | X | X | X | |
| T8PCR2 | FORCA | – | PAAOE | PABOE | PBAOE | PBBOE | PCAOE | PCBOE | ADDRESS:1054H (ESFR)<br>INITIAL VALUE : 0000_0000B |
| | 0 | – | X | X | X | X | X | X | |
| T8PCR3 | HZCLR | POLBO | POLAA | POLAB | POLBA | POLBB | POLCA | POLCB | ADDRESS:1055H (ESFR)<br>INITIAL VALUE : 0000_0000B |
| | X | X | X | X | X | X | X | X | |




NOTE) Do not set to “1111b” in the T8CK[3:0], when two 8-bit timer 7/8 modes.

**Figure 11.27 10-Bit PWM Mode (Force 6-ch)**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| T8CR | 16BIT | T8MS | T8CN | T8ST | T8CK3 | T8CK2 | T8CK1 | T8CK0 | ADDRESS:1052H (ESFR)<br>INITIAL VALUE : 0000_0000B |
| | 0 | X | X | X | X | X | X | X | |
| T8PCR1 | PWM8E | ESYNC | BMOD | PHLT | UPDT | UALL | NOPS1 | NOPS0 | ADDRESS:1053H (ESFR)<br>INITIAL VALUE : 0000_0000B |
| | 1 | X | X | X | X | X | X | X | |
| T8PCR2 | FORCA | – | PAAOE | PABOE | PBAOE | PBBOE | PCAOE | PCBOE | ADDRESS:1054H (ESFR)<br>INITIAL VALUE : 0000_0000B |
| | 1 | – | X | X | X | X | X | X | |
| T8PCR3 | HZCLR | POLBO | POLAA | POLAB | POLBA | POLBB | POLCA | POLCB | ADDRESS:1055H (ESFR)<br>INITIAL VALUE : 0000_0000B |
| | X | X | X | X | X | X | X | X | |




NOTE) Do not set to “1111b” in the T8CK[3:0], when two 8-bit timer 7/8 modes.

**Figure 11.28 10-Bit PWM Mode (Force All-ch)**

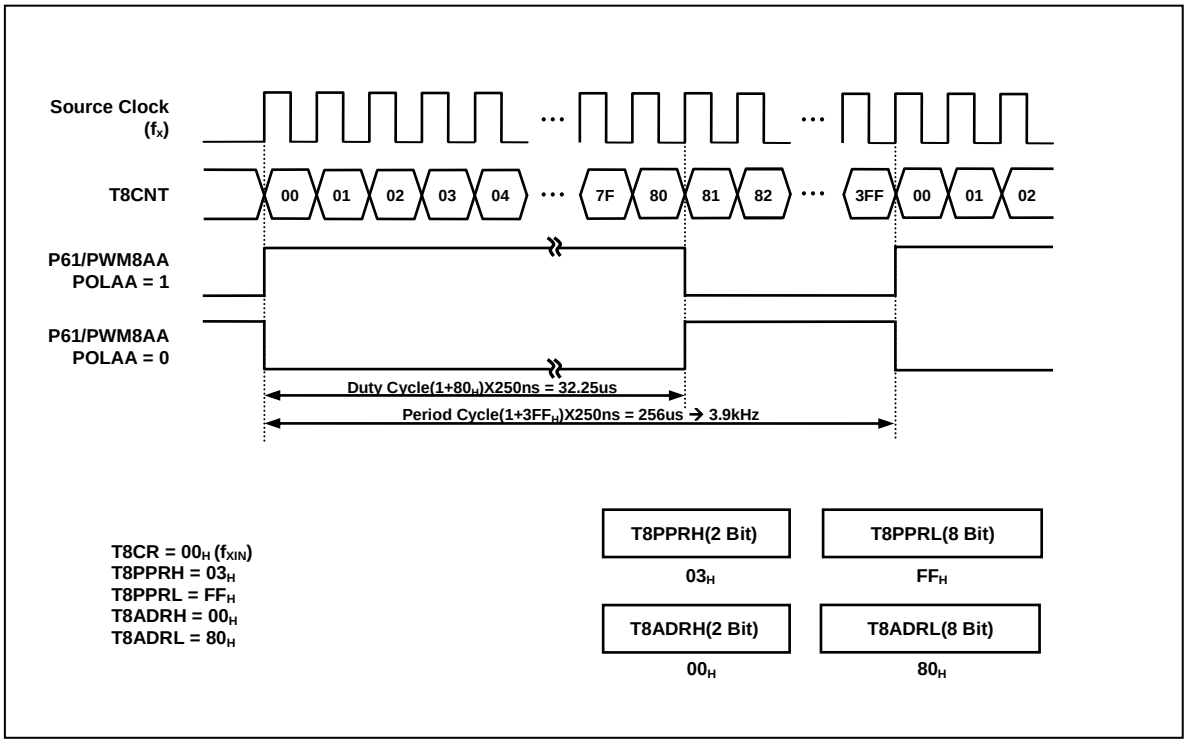



**Figure 11.29 Example of PWM at 4 MHz**

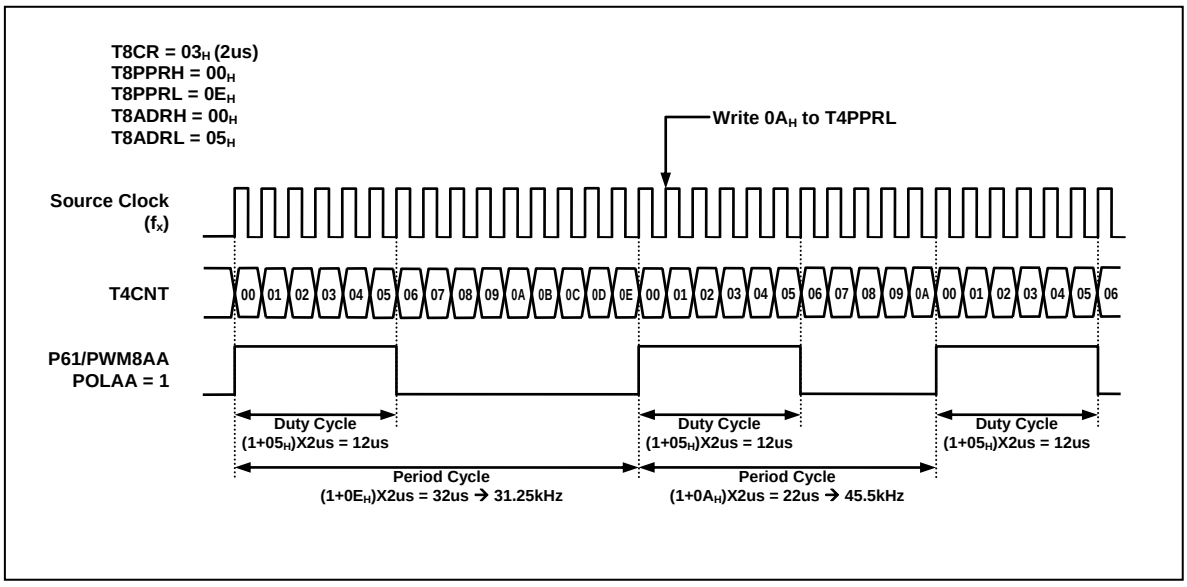



**Figure 11.30 Example of Changing the Period in Absolute Duty Cycle at 4 MHz**

### Update period & duty register value at once

The period and duty of PWM comes to move from temporary registers to T8PPRH/L (PWM Period Register) and T8ADRH/L,T8BDRH/L,T8CDRH/L (PWM Duty Register) when always period match occurs. If you want that the period and duty is immediately changed, the UPDT bit in the T8PCR1 register must set to '1'. It should be noted that it needs the 3 cycle of timer clock for data transfer in the internal clock synchronization circuit. So the update data is written before 3 cycle of timer clock to get the right output waveform.

**Phase correction & Frequency correction**

On operating PWM, it is possible that it is changed the phase and the frequency by using BMOD bit (back-to-back mode) in T8PCR1 register. (Figure 11.31, Figure 11.32, Figure 11.33 referred)

In the back-to-back mode, the counter of PWM repeats up/down count. In fact, the effective duty and period becomes twofold of the register set values. (Figure 11.32, Figure 11.32 referred)

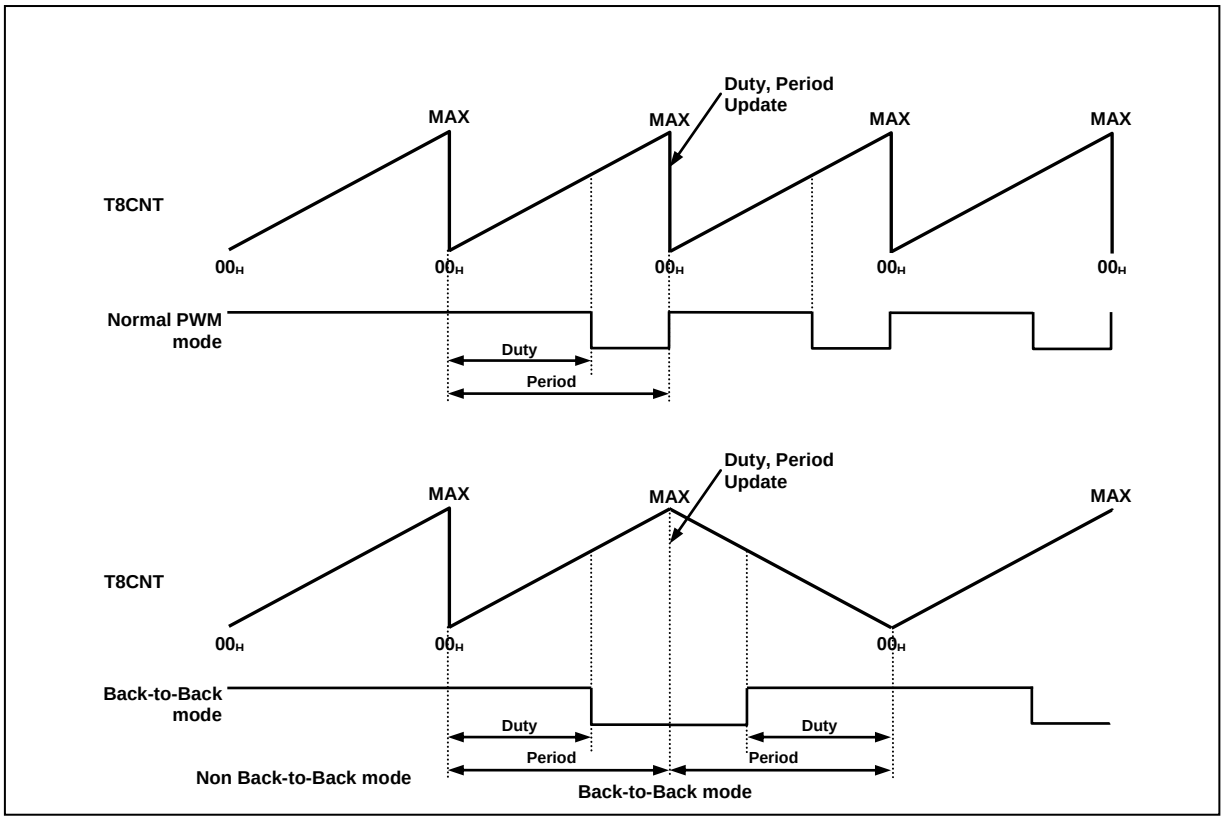



**Figure 11.31 Example of PWM Output Waveform**

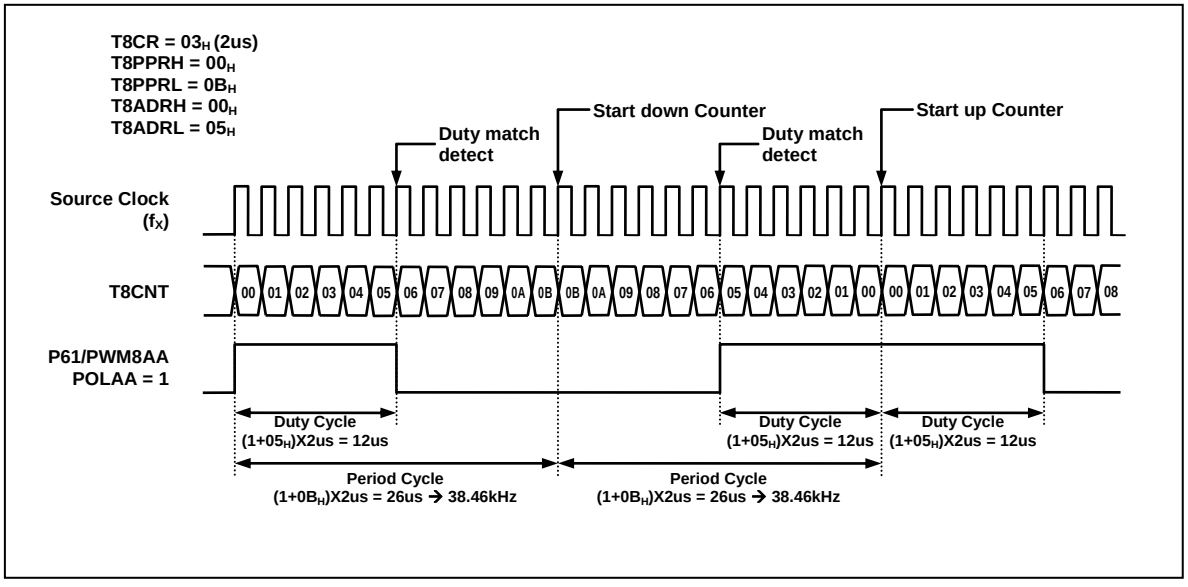



**Figure 11.32 Example of PWM waveform in Back-to-Back mode at 4 MHz**

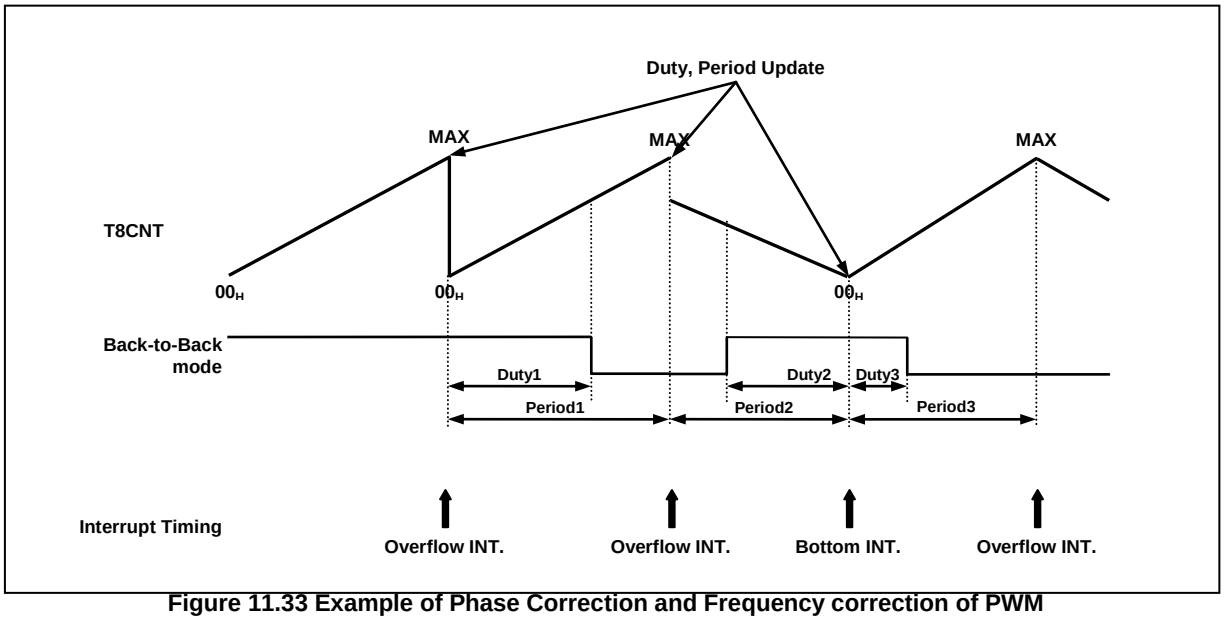



**Figure 11.33 Example of Phase Correction and Frequency correction of PWM**

### External Sync

If using ESYNC bit of T8PCR1 register, it is possible to synchronize the output of PWM from external signal.

If ESYNC bit sets to '1', the external signal moves to PWM module through the BLNK pin. If BLNK signal is low, immediately PWM output becomes a reset value, and internal counter becomes reset. If BLNK signal returns to '1', the counter is started again and PWM output is normally generated. (Figure 11.34 referred)

### PWM Halt

If using PHLT bit of T8PCR1 register, it is possible to stop PWM operation by the software. During PHLT bit being '1', PWM output becomes a reset value, and internal counter becomes reset as 0. Without changing PWM setting, temporarily it is able to stop PWM. In case of T8CNT, when stopping counter, PWM output pin remains before states. But if PHLT bit sets to '1', PWM output pin has reset value.

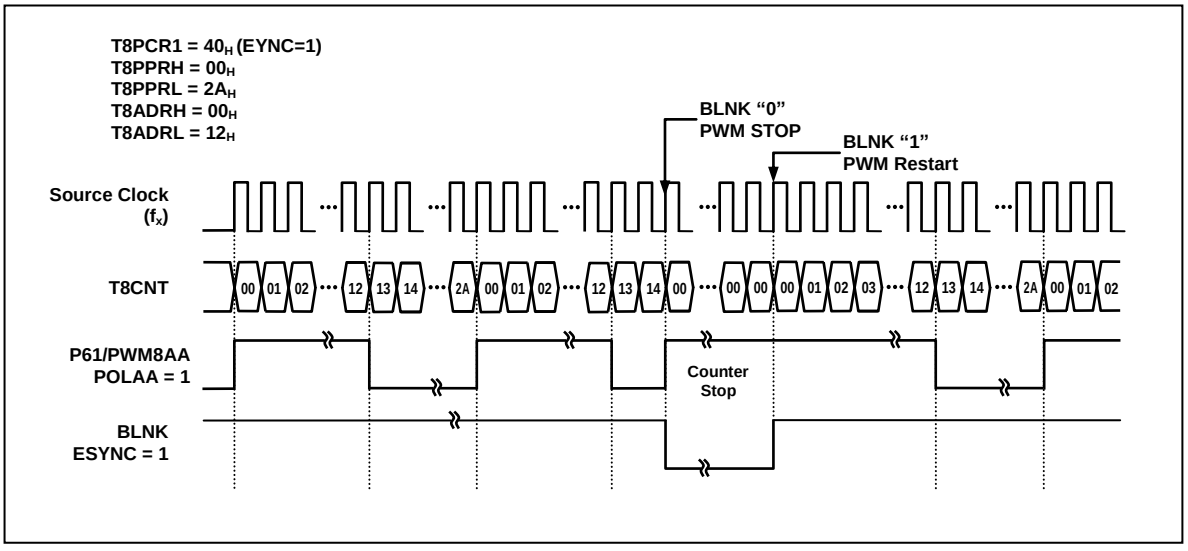



**Figure 11.34 Example of PWM External Synchronization with BLNK Input**

**FORCE Drive ALL Channel with A-ch mode**

If FORCA bit sets to '1', it is possible to enable or disable all PWM output pins through PWM outputs which occur from A-ch duty counter. It is noted that the inversion outputs of A, B, C channel have the same A-ch output waveform. According to POLAA/BB/CC, it is able to control the inversion of outputs.

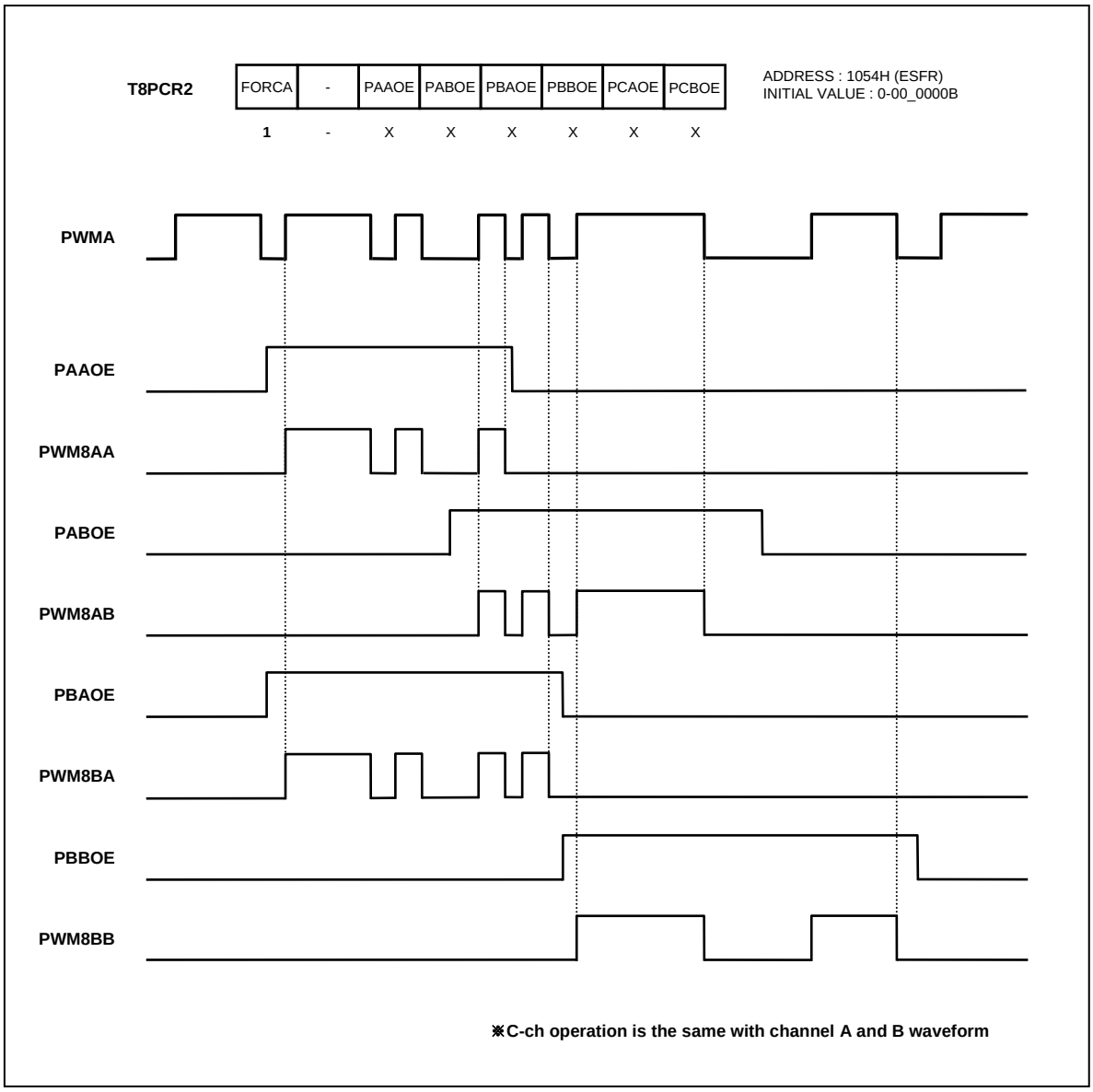



**Figure 11.35 Example of Force Drive All Channel with A-ch**

**FORCE 6-Ch Drive**

If FORCA bit sets to ‘0’, it is possible to enable or disable PWM output pin and inversion output pin generated through the duty counter of each channel. The inversion output is the reverse phase of the PWM output. A AA/AB output of the A-channel duty register, a BA/BB output of the B-channel duty register, a CA/CB output of the C-channel duty register are controlled respectively. If the UALL bit is set to ‘1’, it is updated B/C channel duty at the same time, when it is written by a A-channel duty register.

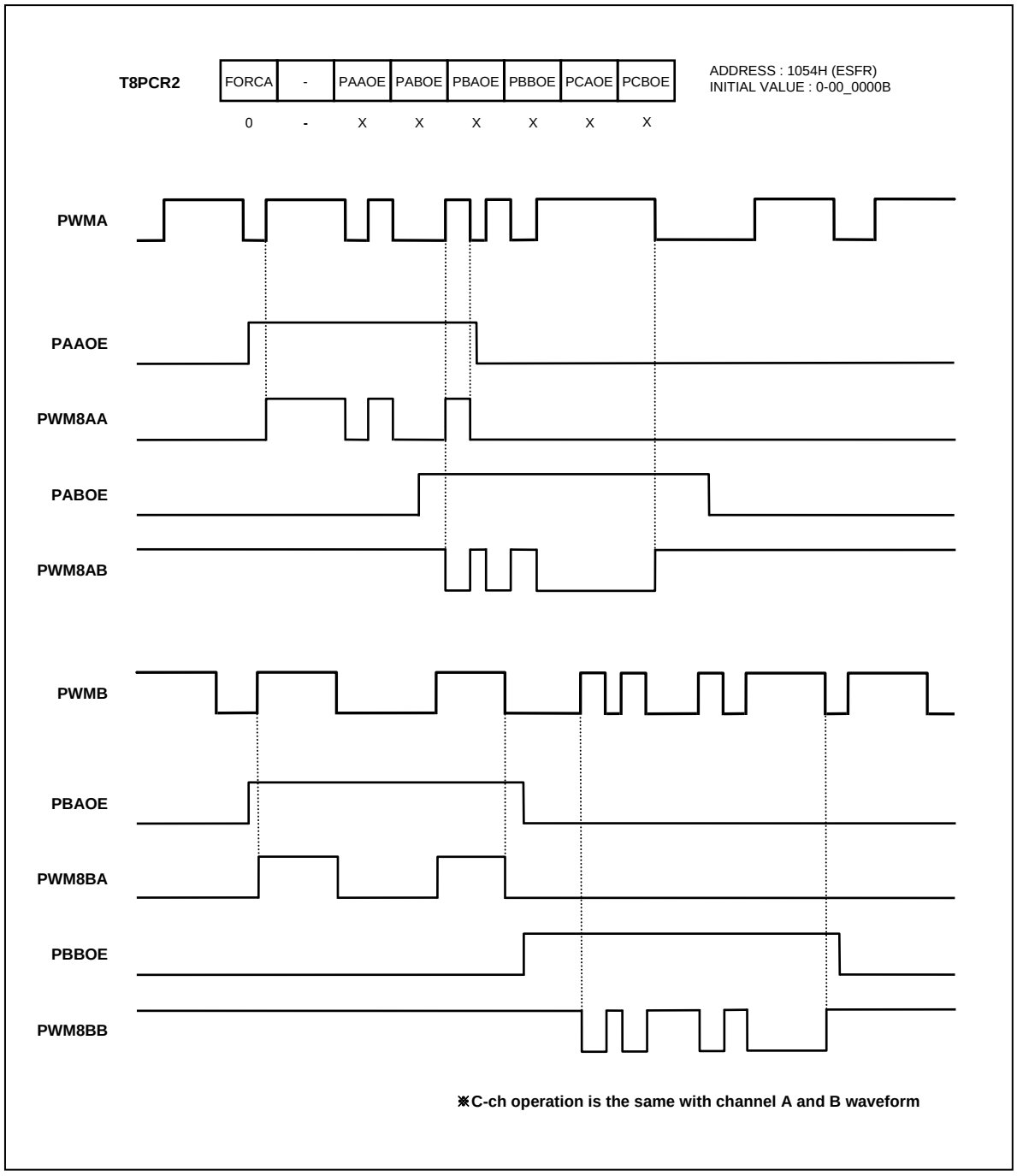



**Figure 11.36 Example of Force Drive 6-ch Mode**

**PWM output Delay**

If using the T8DLYA, T8DLYB, T8DLYC register, it can delay PWM output based on the rising edge. At that time, it does not change the falling edge, so the duty is reduced as the time delay. In POLAA/BA/CA setting to '0', the delay is applied to the falling edge. In POLAA/BA/CA setting to '1', the delay is applied to the rising edge. It can produce a pair of Non-overlapping clock. The each channel is able to have 4-bit delay. As it can select the clock up to 1/8 divided clock using NOPS[1:0] the delay of its maximum 128 timer clock cycle is produced.

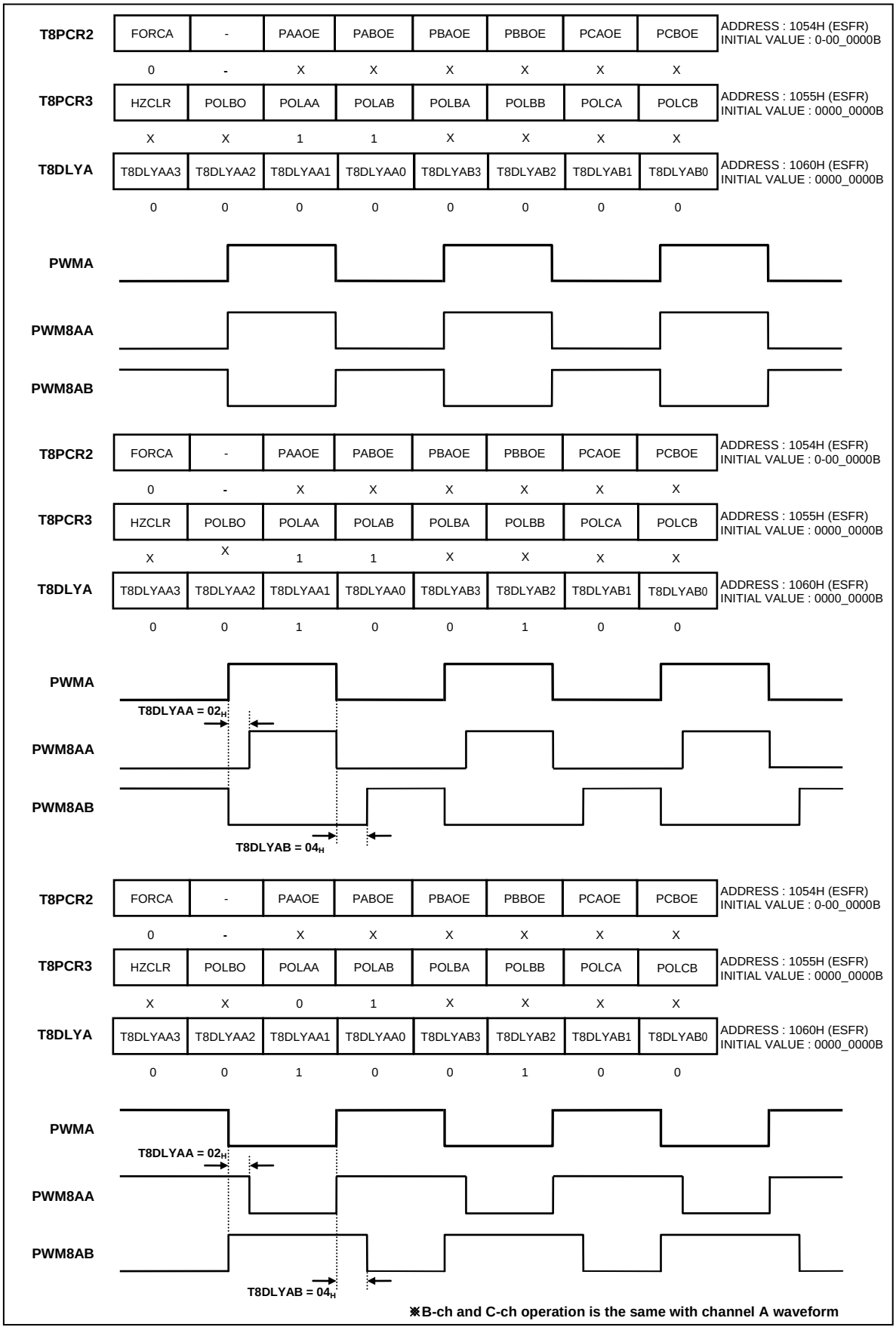
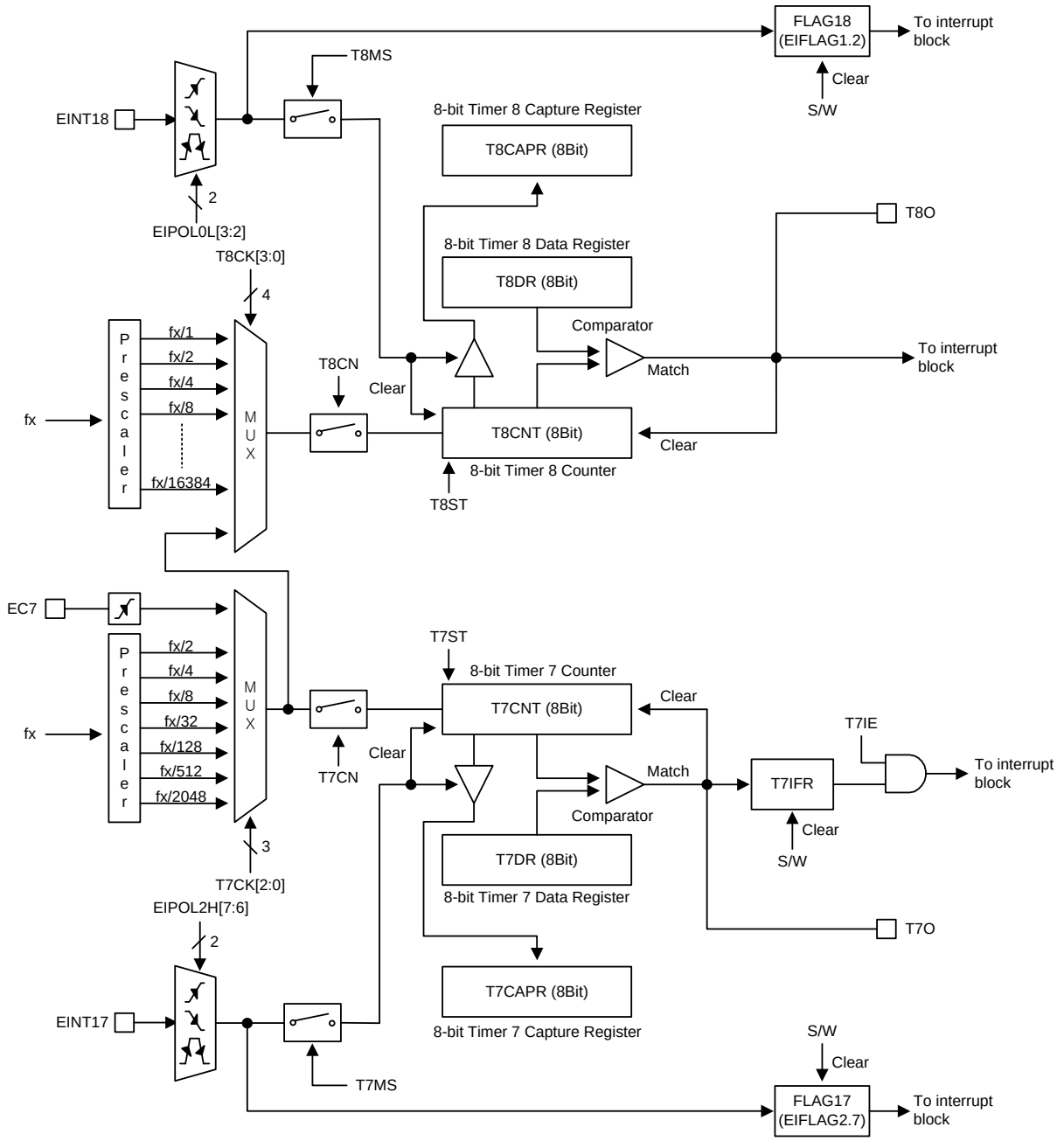


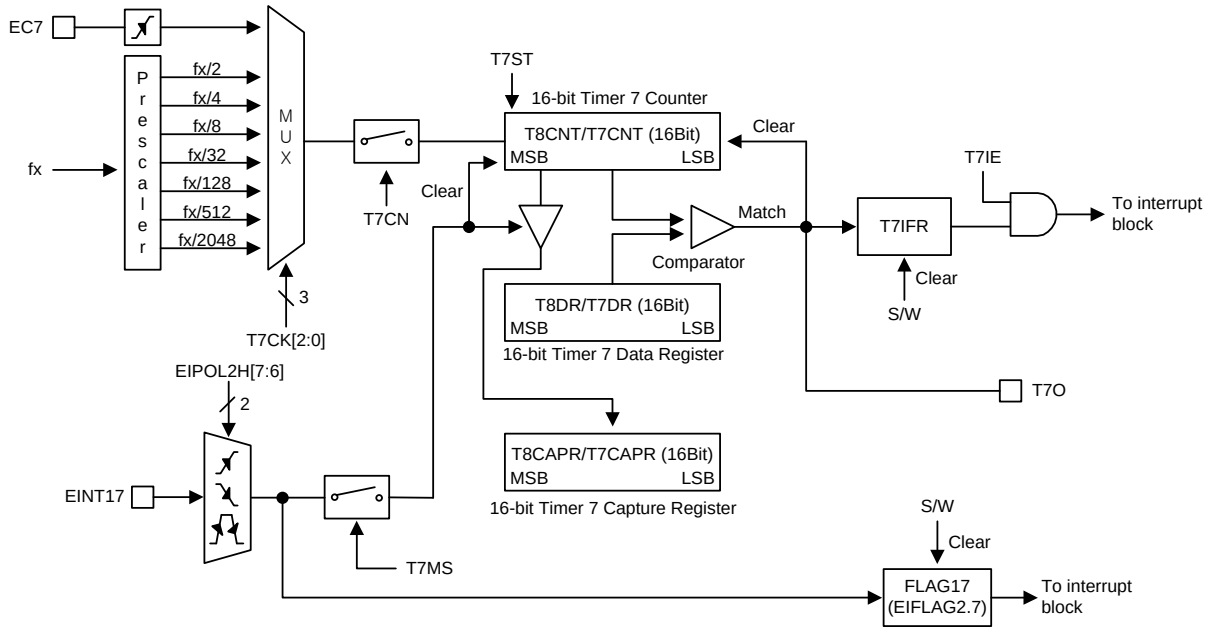

**Figure 11.37 Example of PWM Delay**

### 11.7.7 Block Diagram



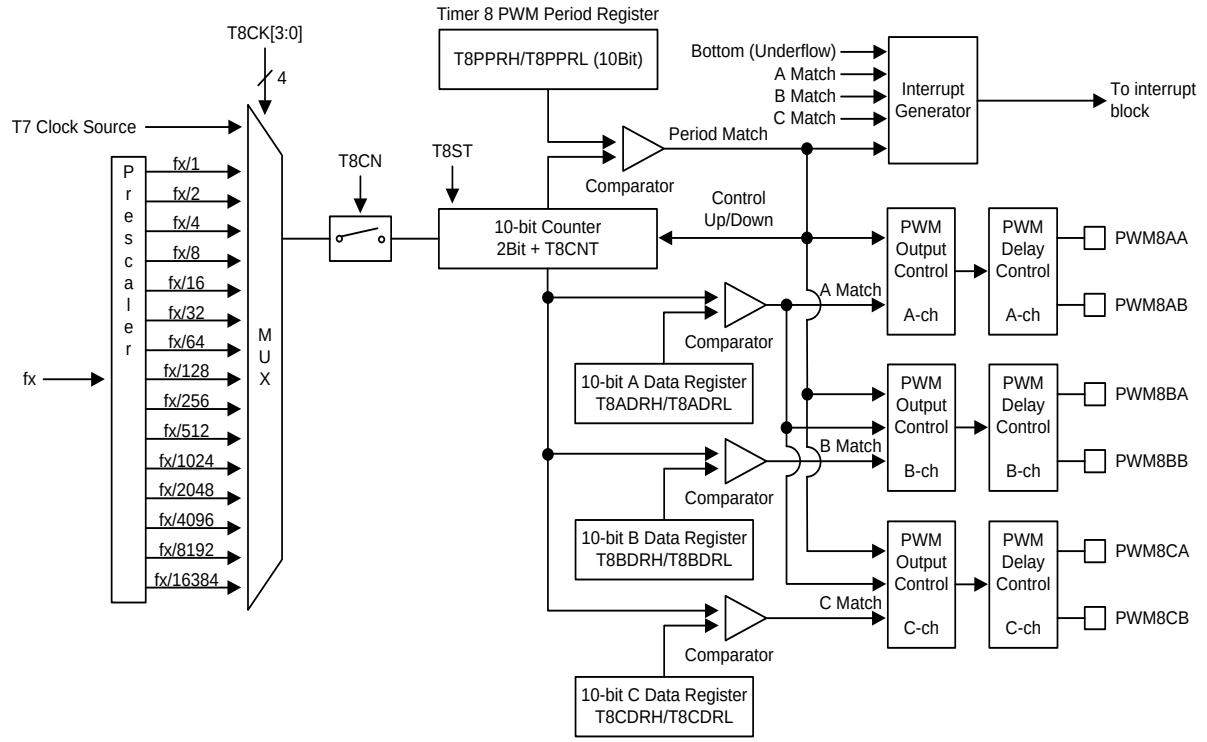

NOTE) Do not set to “1111b” in the T8CK[3:0], when two 8-bit timer 7/8 modes.

**Figure 11.38 Two 8-Bit Timer 7/8 Block Diagram**




NOTE) The T8CR.7 bit (16BIT) should be set to '1' and the T8CK[3:0] should be set to "1111b".

**Figure 11.39 16-Bit Timer 7 Block Diagram**




NOTE) Do not set to "1111b" in the T8CK[3:0], when two 8-bit timer 7/8 modes.

**Figure 11.40 10-Bit PWM Timer 8 Block Diagram**

### 11.7.8 Register Map

**Table 11-12 Timer 7/8 Register Map**

| Name | Address | Dir | Default | Description |
|---|---|---|---|---|
| T7CNT | 1051H (ESFR) | R | 00H | Timer 7 Counter Register |
| T7DR | 1051H (ESFR) | W | FFH | Timer 7 Data Register |
| T7CAPR | 1051H (ESFR) | R | 00H | Timer 7 Capture Data Register |
| T7CR | 1050H (ESFR) | R/W | 00H | Timer 7 Control Register |
| T8PPRH | 1059H (ESFR) | R/W | 00H | Timer 8 PWM Period High Register |
| T8PPRL | 1058H (ESFR) | R/W | FFH | Timer 8 PWM Period Low Register |
| T8ADRH | 105BH (ESFR) | R/W | 00H | Timer 8 PWM A Duty High Register |
| T8ADRL | 105AH (ESFR) | R/W | 7FH | Timer 8 PWM A Duty Low Register |
| T8BDRH | 105DH (ESFR) | R/W | 00H | Timer 8 PWM B Duty High Register |
| T8BDRL | 105CH (ESFR) | R/W | 7FH | Timer 8 PWM B Duty Low Register |
| T8CDRH | 105FH (ESFR) | R/W | 00H | Timer 8 PWM C Duty High Register |
| T8CDRL | 105EH (ESFR) | R/W | 7FH | Timer 8 PWM C Duty Low Register |
| T8DLYA | 1060H (ESFR) | R/W | 00H | Timer 8 PWM A Delay Register |
| T8DLYB | 1061H (ESFR) | R/W | 00H | Timer 8 PWM B Delay Register |
| T8DLYC | 1062H (ESFR) | R/W | 00H | Timer 8 PWM C Delay Register |
| T8DR | 1063H (ESFR) | R/W | FFH | Timer 8 Data Register |
| T8CAPR | 1064H (ESFR) | R | 00H | Timer 8 Capture Data Register |
| T8CNT | 1065H (ESFR) | R | 00H | Timer 8 Counter Register |
| T8CR | 1052H (ESFR) | R/W | 00H | Timer 8 Control Register |
| T8PCR1 | 1053H (ESFR) | R/W | 00H | Timer 8 PWM Control Register 1 |
| T8PCR2 | 1054H (ESFR) | R/W | 00H | Timer 8 PWM Control Register 2 |
| T8PCR3 | 1055H (ESFR) | R/W | 00H | Timer 8 PWM Control Register 3 |
| T8ISR | 1056H (ESFR) | R/W | 00H | Timer 8 Interrupt Status Register |
| T8MSK | 1057H (ESFR) | R/W | 00H | Timer 8 Interrupt Mask Register |

### 11.7.8.1 Timer/Counter 7 Register Description

The timer/counter 7 register consists of timer 7 counter register (T7CNT), timer 7 data register (T7DR), timer 7 capture data register (T7CAPR) and timer 7 control register (T7CR).

### 11.7.8.2 Register Description for Timer/Counter 7

**T7CNT (Timer 7 Counter Register: Read Case, Timer mode only) : 1051H (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| T7CNT7 | T7CNT6 | T7CNT5 | T7CNT4 | T7CNT3 | T7CNT2 | T7CNT1 | T7CNT0 |
| R | R | R | R | R | R | R | R |

Initial value : 00H

**T7CNT[7:0]** T7 Counter

**T7DR (Timer 7 Data Register: Write Case) : 1051H (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| T7DR7 | T7DR6 | T7DR5 | T7DR4 | T7DR3 | T7DR2 | T7DR1 | T7DR0 |
| W | W | W | W | W | W | W | W |

Initial value : FFH

**T7DR[7:0]** T7 Data

**T7CAPR (Timer 7 Capture Data Register: Read Case, Capture mode only) : 1051H (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| T7CAPR7 | T7CAPR6 | T7CAPR5 | T7CAPR4 | T7CAPR3 | T7CAPR2 | T7CAPR1 | T7CAPR0 |
| R | R | R | R | R | R | R | R |

Initial value : 00H

**T7CAPR[7:0]** T7 Capture Data

**T7CR (Timer 7 Control Register) : 1050H (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| T7EN | T7IE | T7MS | T7CK2 | T7CK1 | T7CK0 | T7CN | T7ST |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**T7EN** Control Timer 7
- 0 Timer 7 disable
- 1 Timer 7 enable

**T7IE** Enable or Disable Timer 7 Interrupt
- 0 Disable
- 1 Enable

**T7MS** Control Timer 7 Operation Mode
- 0 Timer/counter mode (T7O: toggle at match)
- 1 Capture mode (the match interrupt can occur)

**T7CK[2:0]** Select Timer 7 clock source. fx is main system clock frequency

| T7CK2 | T7CK1 | T7CK0 | Description |
|---|---|---|---|
| 0 | 0 | 0 | fx/2 |
| 0 | 0 | 1 | fx/4 |
| 0 | 1 | 0 | fx/8 |
| 0 | 1 | 1 | fx/32 |
| 1 | 0 | 0 | fx/128 |
| 1 | 0 | 1 | fx/512 |
| 1 | 1 | 0 | fx/2048 |
| 1 | 1 | 1 | External Clock (EC7) |

**T7CN** Control Timer 7 Count Pause/Continue
- 0 Temporary count stop
- 1 Continue count

**T7ST** Control Timer 7 Start/Stop
- 0 Counter stop
- 1 Clear counter and start

NOTE) Refer to the external interrupt flag 1 register (EIFLAG1) to the T7 interrupt flag.

### 11.7.8.3 Timer/Counter 8 Register Description

The timer/counter 8 register consists of timer 8 PWM period high/low register (T8PPRH/L), timer 8 PWM A duty high/low register (T8ADRH/L), timer 8 PWM B duty high/low register (T8BDRH/L), timer 8 PWM C duty high/low register (T8CDRH/L), timer 8 PWM A delay register (T8DLYA), timer 8 PWM B delay register (T8DLYB), timer 8 PWM C delay register (T8DLYC), timer 8 data register (T8DR), timer 8 capture data register (T8CAPR), timer 8 counter register (T8CNT), timer 8 control register (T8CR), timer 8 PWM control register 1 (T8PCR1), timer 8 PWM control register 2 (T8PCR2), timer 8 PWM control register 3 (T8PCR3), timer 8 interrupt status register (T8ISR) and timer 8 interrupt mask register (T8MSK).

### 11.7.8.4 Register Description for Timer/Counter 8

**T8PPRH (Timer 8 PWM Period High Register : 6-ch PWM mode only) : 1059H (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | T8PPRH1 | T8PPRH0 |
| – | – | – | – | – | – | R/W | R/W |

Initial value : 00H

**T8PPRH[1:0]** T8 PWM Period Data High Byte

**T8PPRL (Timer 8 PWM Period Low Register : 6-ch PWM mode only) : 1058H (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| T8PPRL7 | T8PPRL6 | T8PPRL5 | T8PPRL4 | T8PPRL3 | T8PPRL2 | T8PPRL1 | T8PPRL0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : FFH

**T8PPRL[7:0]** T8 PWM Period Data Low Byte

**T8ADRH (Timer 8 PWM A Duty High Register : 6-ch PWM mode only) : 105BH (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | T8ADRH1 | T8ADRH0 |
| – | – | – | – | – | – | R/W | R/W |

Initial value : 00H

**T8ADRH[1:0]** T8 PWM A Duty Data High Byte

**T8ADRL (Timer 8 PWM A Duty Low Register : 6-ch PWM mode only) : 105AH (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| T8ADRL7 | T8ADRL6 | T8ADRL5 | T8ADRL4 | T8ADRL3 | T8ADRL2 | T8ADRL1 | T8ADRL0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 7FH

**T8ADRL[7:0]** T8 PWM A Duty Data Low Byte

**T8BDRH (Timer 8 PWM B Duty High Register : 6-ch PWM mode only) : 105DH (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | T8BDRH1 | T8BDRH0 |
| – | – | – | – | – | – | R/W | R/W |

Initial value : 00H

**T8BDRH[1:0]** T8 PWM B Duty Data High Byte

**T8BDRL (Timer 8 PWM B Duty Low Register : 6-ch PWM mode only) : 105CH (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| T8BDRL7 | T8BDRL6 | T8BDRL5 | T8BDRL4 | T8BDRL3 | T8BDRL2 | T8BDRL1 | T8BDRL0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 7FH

**T8BDRL[7:0]** T8 PWM B Duty Data Low Byte

**T8CDRH (Timer 8 PWM C Duty High Register : 6-ch PWM mode only) : 105FH (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | T8CDRH1 | T8CDRH0 |
| – | – | – | – | – | – | R/W | R/W |

Initial value : 00H

**T4CDRH[1:0]** T4 PWM C Duty Data High Byte

**T8CDRL (Timer 8 PWM C Duty Low Register : 6-ch PWM mode only) : 105EH (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| T8CDRL7 | T8CDRL6 | T8CDRL5 | T8CDRL4 | T8CDRL3 | T8CDRL2 | T8CDRL1 | T8CDRL0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 7FH

**T8CDRL[7:0]** T8 PWM C Duty Data Low Byte

**T8DLYA (Timer 8 PWM A Delay Register : 6-ch PWM mode only) : 1060H (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| T8DLYAA3 | T8DLYAA2 | T8DLYAA1 | T8DLYAA0 | T8DLYAB3 | T8DLYAB2 | T8DLYAB1 | T8DLYAB0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**T8DLYAA[3:0]** PWM8AA Delay Data (Rising edge only)
**T8DLYAB[3:0]** PWM8AB Delay Data (Rising edge only)

**T8DLYB (Timer 8 PWM B Delay Register : 6-ch PWM mode only) : 1061H (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| T8DLYBA3 | T8DLYBA2 | T8DLYBA1 | T8DLYBA0 | T4DLYBB3 | T8DLYBB2 | T8DLYBB1 | T8DLYBB0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**T8DLYBA[3:0]** PWM8BA Delay Data (Rising edge only)
**T8DLYBB[3:0]** PWM8BB Delay Data (Rising edge only)

**T8DLYC (Timer 8 PWM C Delay Register : 6-ch PWM mode only) : 1062H (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| T8DLYCA3 | T8DLYCA2 | T8DLYCA1 | T8DLYCA0 | T8DLYCB3 | T8DLYCB2 | T8DLYCB1 | T8DLYCB0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**T8DLYCA[3:0]** PWM8CA Delay Data (Rising edge only)

**T8DLYCB[3:0]** PWM8CB Delay Data (Rising edge only)

**T8DR (Timer 8 Data Register: Timer and Capture mode only) : 1063H (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| T8DR7 | T8DR6 | T8DR5 | T8DR4 | T4DR3 | T8DR2 | T8DR1 | T8DR0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : FFH

**T8DR[7:0]** T8 Data

**T8CAPR (Timer 8 Capture Data Register: Read Case, Capture mode only) : 1064H (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| T8CAPR7 | T8CAPR6 | T8CAPR5 | T8CAPR4 | T8CAPR3 | T8CAPR2 | T8CAPR1 | T8CAPR0 |
| R | R | R | R | R | R | R | R |

Initial value : 00H

**T8CAPR[7:0]** T8 Capture Data

**T8CNT (Timer 8 Counter Register: Read Case, Timer mode only) : 1065H (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| T8CNT7 | T8CNT6 | T8CNT5 | T8CNT4 | T8CNT3 | T8CNT2 | T8CNT1 | T8CNT0 |
| R | R | R | R | R | R | R | R |

Initial value : 00H

**T8CNT[7:0]** T8 Counter

**T8CR (Timer 8 Control Register) : 1052H (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 16BIT | T8MS | T8CN | T8ST | T8CK3 | T8CK2 | T8CK1 | T8CK0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**16BIT** Select Two 8-bit or 16-bit Mode for Timer 7/8

0 Two 8-bit Timer 7/8

1 16-bit Timer 7

**T8MS** Control Timer 8 Operation Mode

0 Timer/counter mode (T8O: toggle at match)

1 Capture mode (the match interrupt can occur)

**T8CN** Control Timer 8 Count Pause/Continue

0 Temporary count stop

1 Continue count

**T8ST** Control Timer 8 Start/Stop

0 Counter stop

1 Clear counter and start

**T8CK[3:0]** Select Timer 8 clock source. fx is main system clock frequency

| T8CK3 | T8CK2 | T8CK1 | T8CK0 | Description |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | fx/1 |
| 0 | 0 | 0 | 1 | fx/2 |
| 0 | 0 | 1 | 0 | fx/4 |
| 0 | 0 | 1 | 1 | fx/8 |
| 0 | 1 | 0 | 0 | fx/16 |
| 0 | 1 | 0 | 1 | fx/32 |
| 0 | 1 | 1 | 0 | fx/64 |
| 0 | 1 | 1 | 1 | fx/128 |
| 1 | 0 | 0 | 0 | fx/256 |
| 1 | 0 | 0 | 1 | fx/512 |
| 1 | 0 | 1 | 0 | fx/1024 |
| 1 | 0 | 1 | 1 | fx/2048 |
| 1 | 1 | 0 | 0 | fx/4096 |
| 1 | 1 | 0 | 1 | fx/8192 |
| 1 | 1 | 1 | 0 | fx/16384 |
| 1 | 1 | 1 | 1 | Timer 7 clock (only 16-Bit Timer 7) |

NOTE) Do not set to “1111b” in the T8CK[3:0] when two 8-bit timer 7/8 modes.

**T8PCR1 (Timer 8 PWM Control Register 1) : 1053H (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PWM8E | ESYNC | BMOD | PHLT | UPDT | UALL | NOPS1 | NOPS0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**PWM8E** Control Timer 8 Mode

0 Select timer/counter or capture mode of Timer 8

1 Select 10-bit PWM mode of Timer 8

**ESYNC** Select the Operation of External Sync with the BLNK pin

0 Disable external sync operation

1 Enable external sync operation
(The all PWM8xA/PWM8xB pins are high-impedance outputs on rising edge of the BLNK input pin. Where x= A, B and C)

**BMOD** Control Back-to-Back Mode Operation

0 Disable back-to-back mode (up count only)

1 Enable back-to-back mode (up/down count only)

**PHLT** Control Timer 8 PWM Operation

0 Run 10-bit PWM

1 Stop 10-bit PWM (counter hold and output disable)

**UPDT** Select the Update Timer of T8PPR/T8ADR/T8BDR/T8CDR

0 Update at period match of T8CNT and T8PPR

1 Update at any time when written

**UALL** Control Update All Duty Registers (T8ADR/T8BDR/T8CDR)

0 Write a duty register separately

1 Wrtie all duty registers via Timer 8 PWM A dury register (T8ADR)

**NOPS[1:0]** Select on-Overlap Prescaler

| NOPS1 | NOPS0 | Description |
|---|---|---|
| 0 | 0 | $f_{PWM}$/1 |
| 0 | 1 | $f_{PWM}$ /2 |
| 1 | 0 | $f_{PWM}$ /4 |
| 1 | 1 | $f_{PWM}$ /8 |

NOTE) Where the $f_{PWM}$ is the clock frequency of the Timer 8 PWM.

**T8PCR2 (Timer 8 PWM Control Register 2) : 1054H (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FORCA | – | PAAOE | PABOE | PBAOE | PBBOE | PCAOE | PCBOE |
| R/W | – | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**FORCA** Control The PWM outputs Mode

0 6-channel mode
(The PWM8xA/PWM8xB pins are output according to the T8xDR registers, respectively. Where x = A, B and C)

1 Force A-channel mode
(The all PWM8xA/PWM8xB pins are output according to the only T8ADR registers. Where x = A, B and C)

**PAAOE** Select Channel PWM8AA Operation

0 Disable PWM8AA output

1 Enable PWM8AA output

**PABOE** Select Channel PWM8AB Operation

0 Disable PWM8AB output

1 Enable PWM8AB output

**PBAOE** Select Channel PWM8BA Operation

0 Disable PWM8BA output

1 Enable PWM8BA output

**PBBOE** Select Channel PWM8BB Operation

0 Disable PWM8BB output

1 Enable PWM8BB output

**PCAOE** Select Channel PWM8CA Operation

0 Disable PWM8CA output

1 Enable PWM8CA output

**PCBOE** Select Channel PWM8CB Operation

0 Disable PWM8CB output

1 Enable PWM8CB output

**T8PCR3 (Timer 8 PWM Control Register 3) : 1055H (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| HZCLR | POLBO | POLAA | POLAB | POLBA | POLBB | POLCA | POLCB |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**HZCLR** High-Impedance Output Clear Bit

0 No effect

1 Clear high-impedance output

(The PWM8xA/PWM8xB pins are back to output and this bit is automatically cleared to logic '0'. where x = A, B and C)

**POLBO** Configure PWM8AB/PWM8BB/PWM8CB Channel Polarity When these pins are disabled

0 These pins are output according to the polarity setting when disable (POLAB/POLBB/POLCB bits)

1 These pins are same level as the PWM8xA pins regardless of the polarity setting when disable (POLAB/POLBB/POLCB bits, where x = A, B and C)

**POLAA** Configure PWM8AA Channel Polarity

0 Start at high level (This pin is low level when disable)

1 Start at low level (This pin is high level when disable)

**POLAB** Configure PWM8AB Channel Polarity

0 Non-inversion signal of PWM8AA pin

1 Inversion signal of PWM8AA pin

**POLBA** Configure PWM8AA Channel Polarity

0 Start at high level (This pin is low level when disable)

1 Start at low level (This pin is high level when disable)

**POLBB** Configure PWM8BB Channel Polarity

0 Non-inversion signal of PWM8BA pin

1 Inversion signal of PWM8BA pin

**POLCA** Configure PWM8CA Channel Polarity

0 Start at high level (This pin is low level when disable)

1 Start at low level (This pin is high level when disable)

**POLCB** Configure PWM8CB Channel Polarity

0 Non-inversion signal of PWM8CA pin

1 Inversion signal of PWM8CA pin

**T8ISR (Timer 8 Interrupt Status Register) : 1056H (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| IOVR | IBTM | ICMA | ICMB | ICMC | – | – | – |
| R/W | R/W | R/W | R/W | R/W | – | – | – |

Initial value : 00H

**IOVR** Timer 8 Compare Match or Timer 8 Overflow Interrupt Status, Write '0' to this bit for clear

0 Compare match or Overflow no occurrence

1 Compare match or Overflow occurrence

**IBTM** Timer 8 Bottom Interrupt Status, Write ‘0’ to this bit for clear
(In the Back-to-Back mode)

0 Bottom no occurrence

1 Bottom occurrence

**ICMA** Timer 8 PWM A-ch Match Interrupt Staus, Write ‘0’ to this bit for clear

0 PWM A-ch match no occurrence

1 PWM A-ch match occurrence

**ICMB** Timer 8 PWM B-ch Match Interrupt Status, Write ‘0’ to this bit for clear

0 PWM B-ch match no occurrence

1 PWM B-ch match occurrence

**ICMC** Timer 8 PWM C-ch Match Interrupt Status, Write ‘0’ to this bit for clear

0 PWM C-ch match no occurrence

1 PWM C-ch match occurrence

**T8MSK (Timer 8 Interrupt Mask Register) : 1057H (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| OVRMSK | BTMMSK | CMAMSK | CMBMSK | CMCMSK | – | – | – |
| R/W | R/W | R/W | R/W | R/W | – | – | – |

Initial value : 00H

**OVRMSK** Control Timer 8 compare match or Overflow Interrupt

0 Disble compare match or overflow interrupt

1 Enable compare match or overflow interrupt

**BTMMSK** Control Timer 8 Bottom Interrupt

0 Disble bottom interrupt

1 Enable bottom interrupt

**CMAMSK** Control Timer 8 PWM A-ch Match Interrupt

0 Disble PWM A-ch match interrupt

1 Enable PWM A-ch match interrupt

**CMBMSK** Control Timer 8 PWM B-ch Match Interrupt

0 Disble PWM B-ch match interrupt

1 Enable PWM B-ch match interrupt

**CMCMSK** Control Timer 8 PWM C-ch Match Interrupt

0 Disble PWM C-ch match interrupt

1 Enable PWM C-ch match interrupt

## 11.8 10-Bit PWM Generator

### 11.8.1.1 Overview

The 10-bit PWM generator consists of multiplexer, PWM A data high/low register, PWM B data high/low register, pulse generator, delay controller, shot stop controller, emergency stop controller, and PWM control high/low register (PWMADRH/L, PWMBDRH/L, PWMDLYDR, NFILDR, PWMCRH/L, PWMCNTH/L). This 10-bit PWM generator can be used for a IH cooker application.

### 11.8.2 Function Description

It has three operating modes:

- 10-bit PWM one-shot mode without auto-enable
- 10-bit PWM one-shot mode with auto-enable
- 10-bit PWM repeat mode

The 10-bit PWM generator can select a divided system clock from prescaler output. The clock source is selected by clock selection logic which is controlled by the clock selection bits (PWMCK[2:0]).

- 10-bit PWM clock source: $f_X$/1, 2, 4, 8, 16, 32, 64, and 128

The 10-bit PWM generator has four external falling edge trigger pins. The one pin is TRIG. The other three pins are EXTSP0, EXTSP1, and EXTSP2. The falling edge signal of the TRIG pin will clear the 10-bit PWM generator counter. It will restart one PWM cycle immediately or after some delay time. The delay time can be enabled or disabled by TRIGRS[1:0] bits. The delay time is programmable with the 10-bit PWM generator delay data register (PWMDLYDR). The EXTSP0, EXTSP1, and EXTSP2 pins can be selected for shot stop or emergency stop signal by ESPnS[1:0] bits. Where n = 0, 1, and 2. The falling edge signal of the pin which is selected for the shot stop holds the PWMOUT pin to low level by PWMPOL bit set to “0b” or to high level by PWMPOL bit set to “1b” for the current PWM cycle. After that, The 10-bit PWM generator will restart with the value of the PWM B data register (PWMBDRH/L) and the value of the PWM A data register (PWMADRH/L) is changed with the value of the PWM B data register. The emergency stop controller STOP the 10-bit PWM generator by the falling egde signal of the selected pin. The 10-bit PWM generator has a noise filter controller to remove a mis-trigger signal of the TRIG pin. The noise filter controller dons’t work with the noise filter data register (NFILDR) set to “00H”. If the noise filter data register isn’t “00H”, the noise filter can eliminate an invalid signal of the TRIG pin and the filtering time is programmable by the register.

**Table 11-13 10-bit PWM generator Operating Modes**

| PWMEN | P2FSRH[1:0] | PWMMD[1:0] | PWMCK[2:0] | 10-bit PWM generator |
|---|---|---|---|---|
| 1 | 10 | 00 | XXX | One-shot mode without auto-enable |
| 1 | 10 | 01 | XXX | One-shot mode with auto-enable |
| 1 | 10 | 1x | XXX | Repeat mode |

### 11.8.3 10-Bit PWM One-Shot Mode Without Auto-Enable

The 10-bit PWM one-shot mode without auto-enable is selected by PWMMD[1:0] bits set to “00b”. A 10-bit counter register is increased by internal clock input on operation. When the PWMCNTH/PWMCNTL is identical to the PWMADRH/PWMADRL, a match signal is generated, the PWMOUT pin is inverted, and the counting is continued to “3FFH”. If a valid falling edge signal comes from the TRIG pin during the counting, the counter will be cleared to “000H” and the 10-bit PWM generator will be restarted. But if there is no a valid falling edge signal of the TRIG pin, an overflow will be generated finally and the 10-bit PWM generator finishes operation with automatically clearing the PWMEN bit.

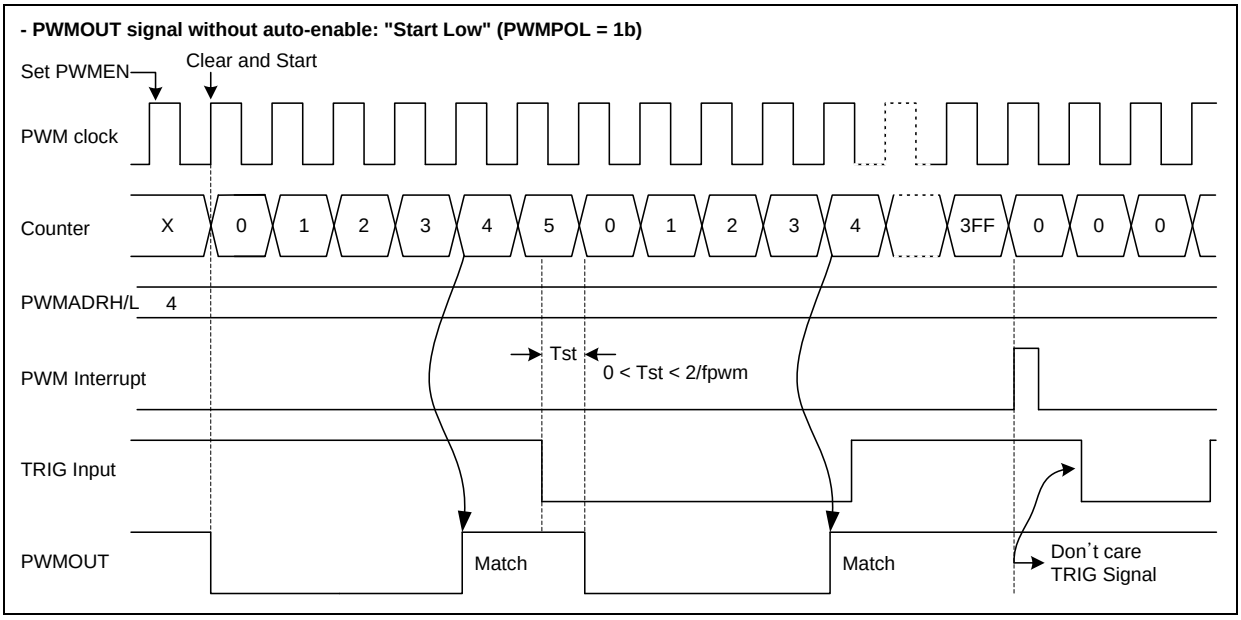



**Figure 11.41 A Timing Chart of 10-bit PWM One-Shot Mode Without Auto-Enable**

### 11.8.4 10-Bit PWM One-Shot Mode With Auto-Enable

The 10-bit PWM one-shot mode with auto-enable is selected by PWMMD[1:0] bits set to “01b”. The function of the 10-bit PWM one-shot mode with auto-enable is the same as the one of the 10-bit PWM one-shot mode without auto-enable, but the 10-bit PWM generator in the 10-bit PWM one-shot mode with auto-enable is automatically enabled as PWMEN bit set to ‘1’ by a valid falling edge of the TRIG pin even if the operation stop. Of course, The EMGIFR bit should be cleared to ‘0’ before a valid falling edge of the TRIG pin.

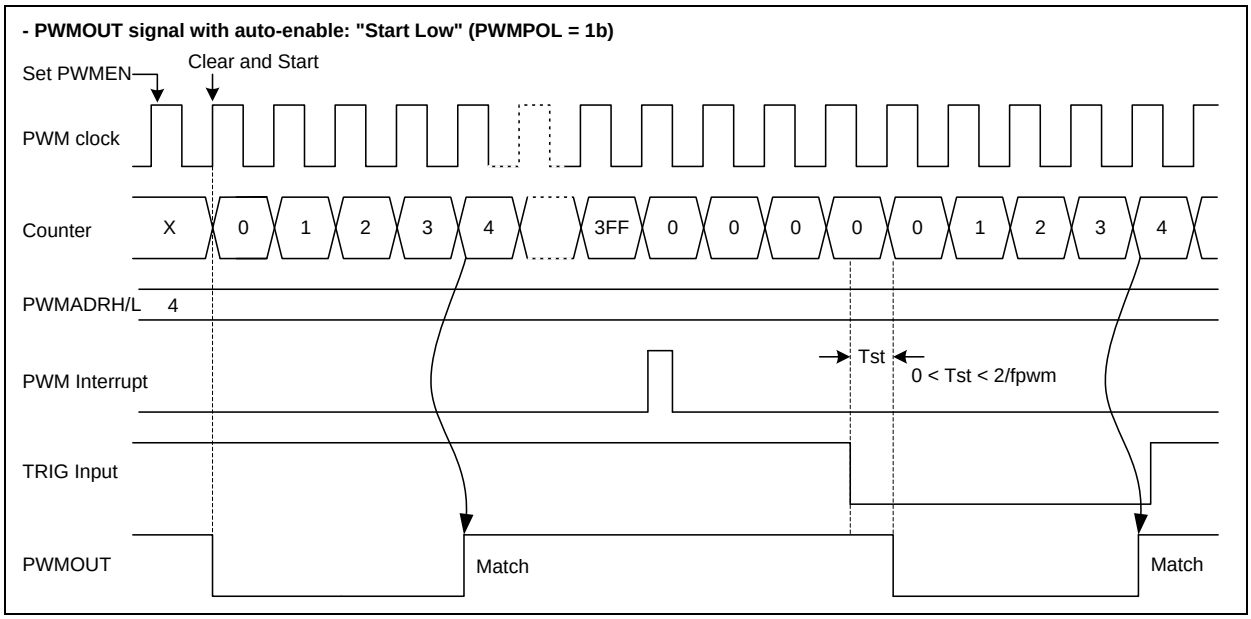



**Figure 11.42 A Timing Chart of 10-bit PWM One-Shot Mode With Auto-Enable**

### 11.8.5 10-Bit PWM Repeat Mode

The 10-bit PWM repeat mode is selected by PWMMD[1:0] bits set to “1xb”. A 10-bit counter register is increased by internal clock input on operation. When the PWMCNTH/PWMCNTL is identical to the PWMADRH/PWMADRL, a match signal is generated, the PWMOUT pin is inverted, and the counting is continued to “3FFH”. If the counter reaches “3FFH” , the counter will be cleared to “000H” and the 10-bit PWM generator will be restarted.

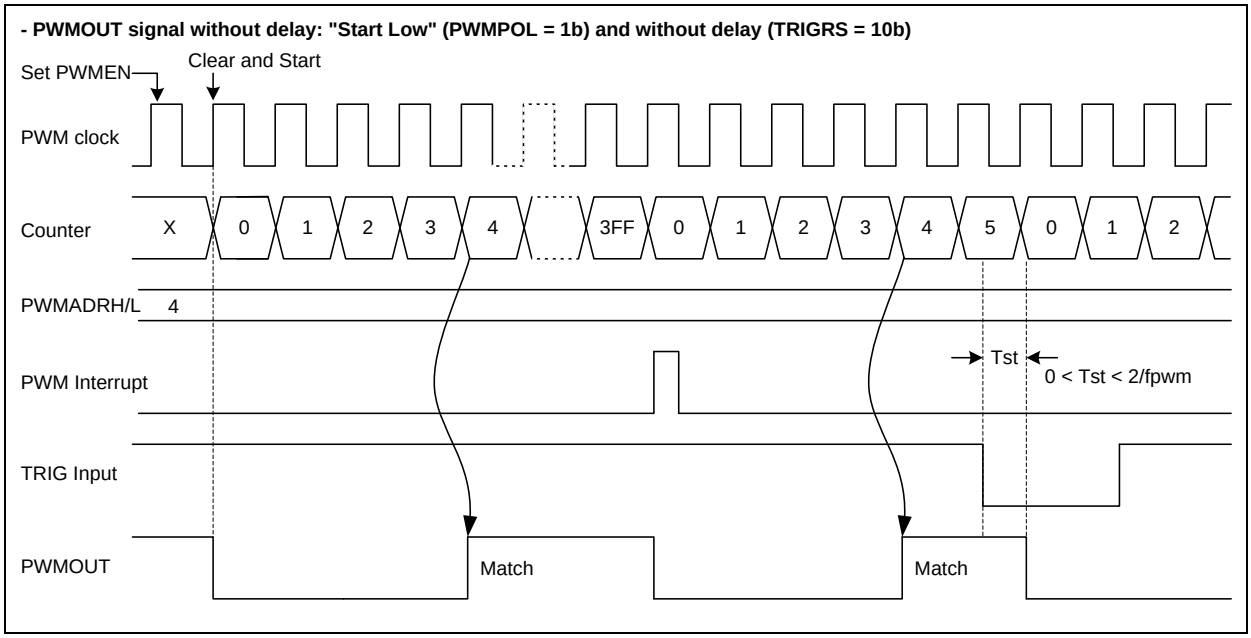



**Figure 11.43 Timing Chart of 10-bit PWM Repeat Mode Without Delay**

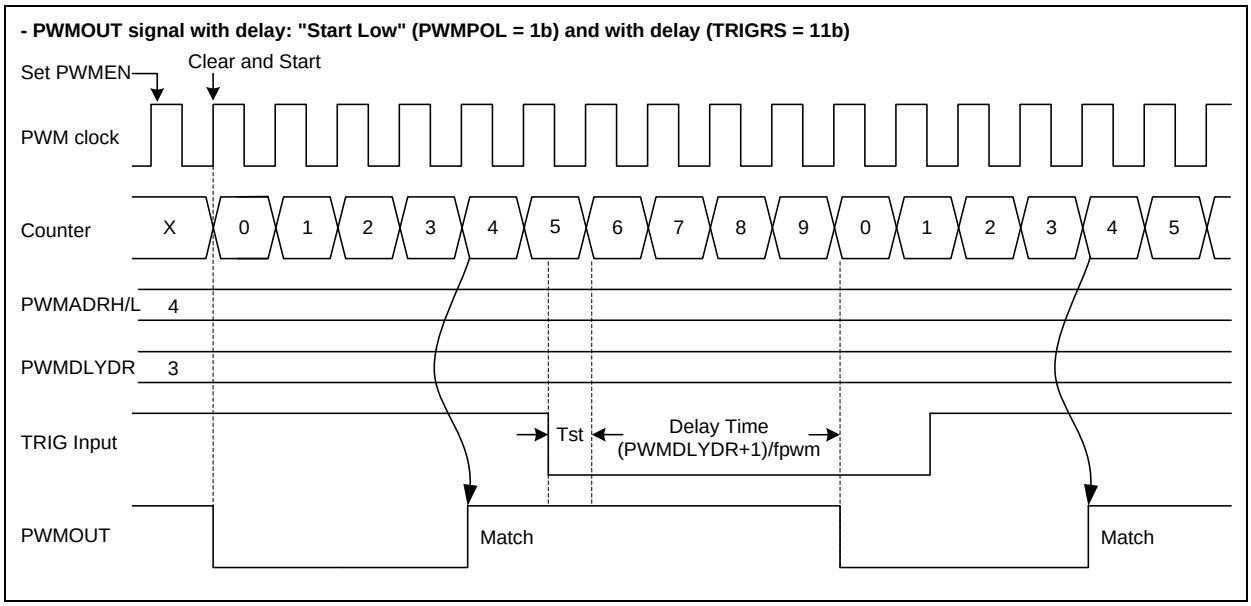



**Figure 11.44 Timing Chart of 10-bit PWM Repeat Mode With Delay**

### 11.8.6 Timing Chart of the Valid Falling Edge by an External Pin

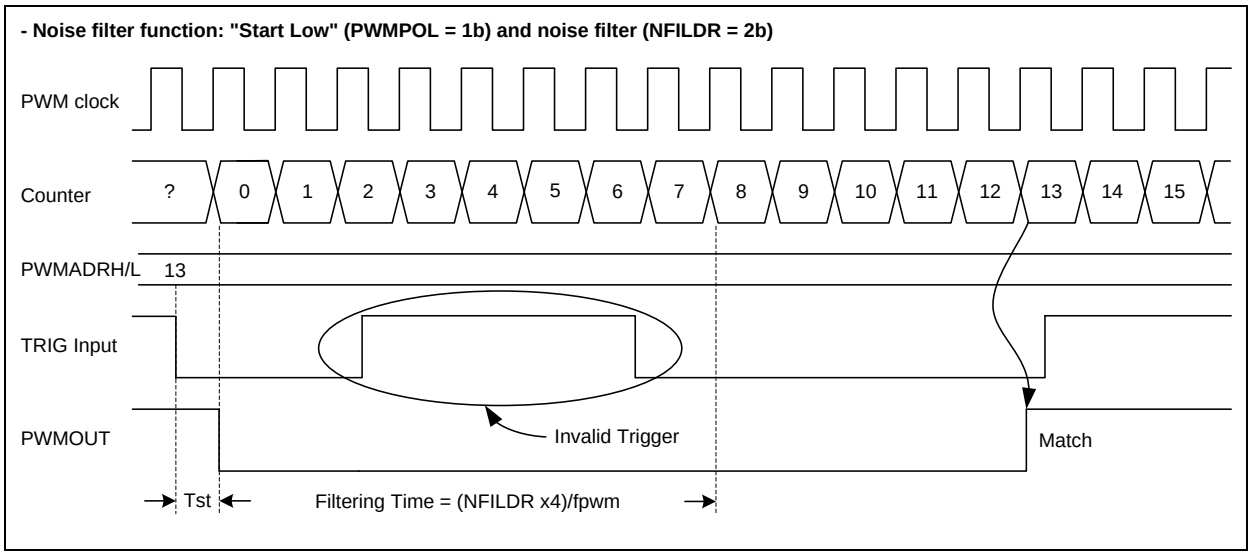



**Figure 11.45 Timing Chart of Noise Filter Function**

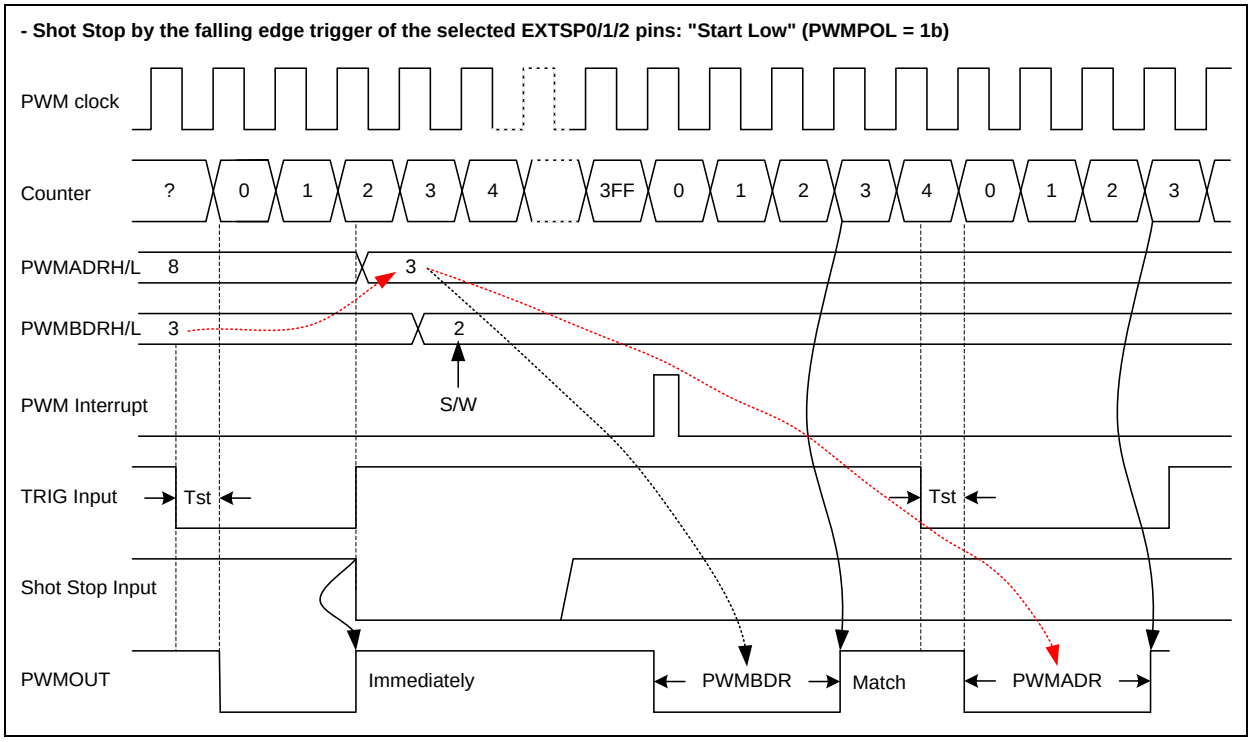



**Figure 11.46 Timing Chart of Shot Stop**

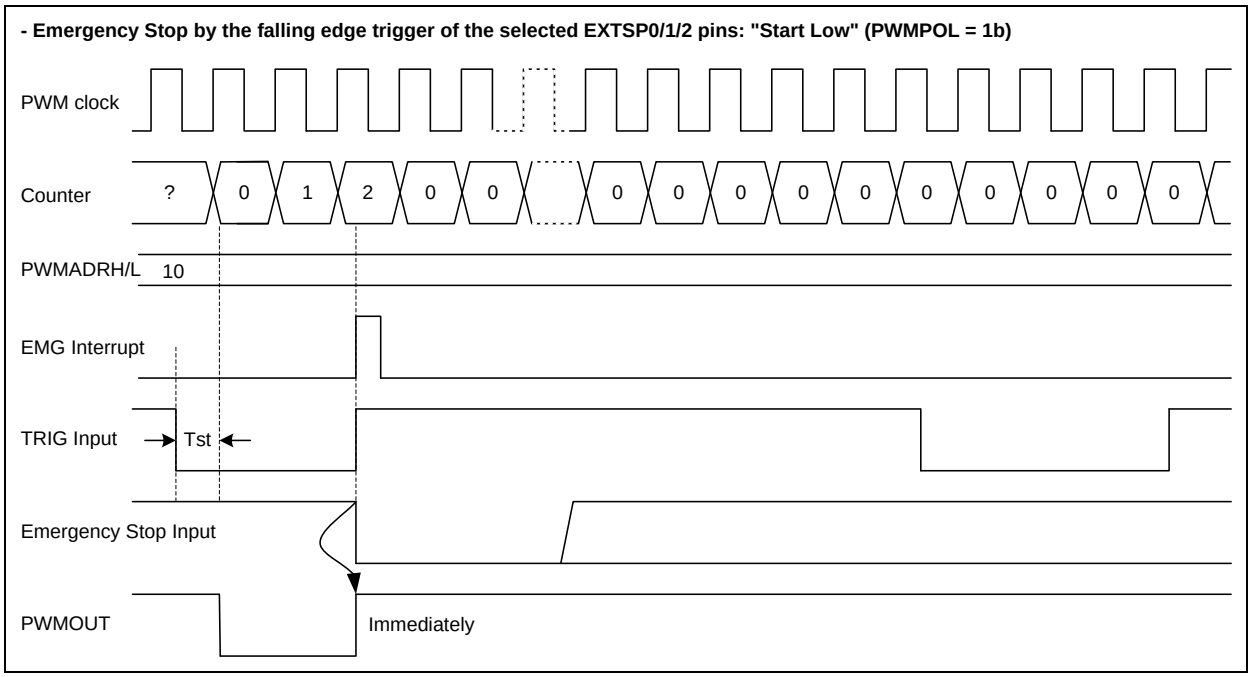
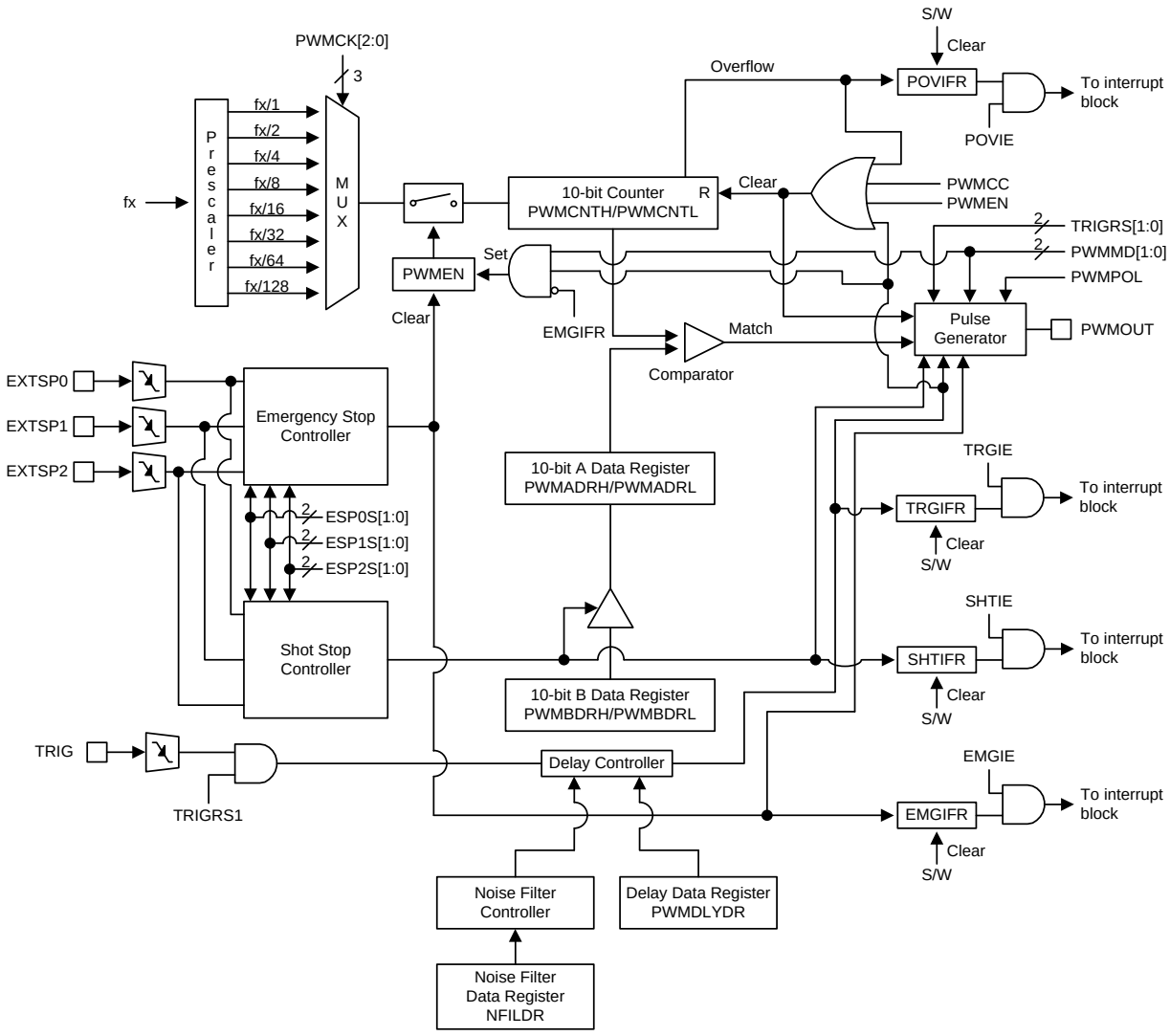



**Figure 11.47 Timing Chart of Emergency Stop**

### 11.8.7 Block Diagram




NOTES)

1. The PWMEN bit is automatically cleared to “0b” after a overflow of the 10-bit counter occurs at a PWM one-shot mode.
2. The PWMEN bit is set to “1b” by a falling edge trigger of the external TRIG pin when it is a PWM one-shot mode with auto-enable and the EMGIFR bit is set to “0b”.
3. The noise filter controller has a timer to realize an invalid trigger signal. The timer will be cleared and restarted every the new cycle of the PWM generator.
4. The noise filter controller doesn’t work if the noise filter data register (NFILDR) is “00H”.
5. If a shot stop occurs by a falling edge trigger of the selected pin, the PWMOUT pin goes to low/high level according to PWMPOL = “0b”/”1b” during the remaining current PWM cycle, respectively. After that, the next PWM cycle starts with the PWMBDR.

**Figure 11.48 10-Bit PWM generator Block Diagram**

### 11.8.8 Register Map

**Table 11-14 10-Bit PWM Generator Register Map**

| Name | Address | Dir | Default | Description |
|---|---|---|---|---|
| PWMCRH | 1079H (ESFR) | R/W | 00H | PWM Generator Control High Register |
| PWMCRL | 1078H (ESFR) | R/W | 00H | PWM Generator Control Low Register |
| PWMADRH | 107BH (ESFR) | R/W | 03H | PWM Generator A Data High Register |
| PWMADRL | 107AH (ESFR) | R/W | FFH | PWM Generator A Data Low Register |
| PWMBDRH | 107DH (ESFR) | R/W | 00H | PWM Generator B Data High Register |
| PWMBDRL | 107CH (ESFR) | R/W | 00H | PWM Generator B Data Low Register |
| PWMCNTH | 107FH (ESFR) | R | 00H | PWM Generator Counter High Register |
| PWMCNTL | 107EH (ESFR) | R | 00H | PWM Generator Counter Low Register |
| PWMDLYDR | 1066H (ESFR) | R/W | 00H | PWM Generator Delay Data Register |
| NFILDR | 1067H (ESFR) | R/W | 0FH | PWM Generator Noise Filter Data Register |
| PGINTCR | 103EH (ESFR) | R/W | 00H | PWM Generator Interrupt Control Register |
| PGIFLAG | 103FH (ESFR) | R/W | 00H | PWM Generator Interrupt Flag Register |

### 11.8.8.1 10-Bit PWM Generator Register Description

The 10-bit PWM generator register consists of PWM generator A data high register (PWMADRH), PWM generator A data low register (PWMADRL), PWM generator B data high register (PWMBDRH), PWM generator B data low register (PWMBDRL), PWM generator control high register (PWMCRH), PWM generator control low register (PWMCRL), PWM generator delay data register (PWMDLYDR), PWM generator noise filter data register (NFILDR), PWM generator interrupt control register (PGINTCR), and PWM generator interrupt flag register (PGIFLAG).

### 11.8.8.2 Register Description for 10-Bit PWM Generator

**PWMADRH (PWM Generator A Data High Register) : 107BH (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | PWMADR9 | PWMADR8 |
| – | – | – | – | – | – | R/W | R/W |

Initial value : 03H

**PWMADR[9:8]** PWM Generator A Data High Byte

**PWMADRL (PWM Generator A Data Low Register) : 107AH (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PWMADR7 | PWMADR6 | PWMADR5 | PWMADR4 | PWMADR3 | PWMADR2 | PWMADR1 | PWMADR0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : FFH

**PWMADR[7:0]** PWM Generator A Data Low Byte

**PWMBDRH (PWM Generator B Data High Register) : 107DH (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | PWMBDR9 | PWMBDR8 |
| – | – | – | – | – | – | R/W | R/W |

Initial value : 00H

**PWMBDR[9:8]** PWM Generator B Data High Byte

**PWMBDRL (PWM Generator B Data Low Register) : 107CH (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PWMBDR7 | PWMBDR6 | PWMBDR5 | PWMBDR4 | PWMBDR3 | PWMBDR2 | PWMBDR1 | PWMBDR0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**PWMBDR[7:0]** PWM Generator B Data Low Byte

**PWMCNTH (PWM Generator Counter High Register) : 107FH (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | PWMCNT9 | PWMCNT8 |
| – | – | – | – | – | – | R | R |

Initial value : 00H

**PWMCNT[9:8]** PWM Generator Counter High Byte

**PWMCNTL (PWM Generator Counter Low Register) : 107EH (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PWMCNT7 | PWMCNT6 | PWMCNT5 | PWMCNT4 | PWMCNT3 | PWMCNT2 | PWMCNT1 | PWMCNT0 |
| R | R | R | R | R | R | R | R |

Initial value : 00H

**PWMCNT[7:0]** PWM Generator Counter Low Byte

**PWMDLYDR (PWM Generator Delay Data Register) : 1066H (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PWMDLY7 | PWMDLY6 | PWMDLY5 | PWMDLY4 | PWMDLY3 | PWMDLY2 | PWMDLY1 | PWMDLY0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**PWMDLY[7:0]** PWM Generator Delay Data

**NFILDR (PWM Generator Noise Filter Data Register) : 1067H (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| NFILDR7 | NFILDR6 | NFILDR5 | NFILDR4 | NFILDR3 | NFILDR2 | NFILDR1 | NFILDR0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 0FH

**NFILDR[7:0]** PWM Generator Noise Filter Data

**PWMCRH (PWM Generator Control High Register) : 1079H (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ESP2S1 | ESP2S0 | ESP1S1 | ESP1S0 | ESP0S1 | ESP0S0 | PWMCC | PWMEN |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**ESPnS[1:0]** Select the external EXTSPn pin function with falling edge trigger

| ESPnS1 | ESPnS0 | Description |
|---|---|---|
| 0 | 0 | Disable the EXTSPn pin function |
| 0 | 1 | Enable shot stop. The PWMOUT pin is held for the current PWM cycle and restarts with PWMBDR by the external EXTSPn pin falling edge trigger. If a shot stop occurs, the PWMOUT pin is held for the current PWM cycle to low level when PWMPOL bit is '0' and high level when PWMPOL bit is '1'. |
| 1 | 0 | Enable emergency stop. The PWMOUT pin can be stopped by the external EXTSPn pin falling edge trigger or software control with PWMEN bit. If an emergency stop occurs, the PWMEN bit is immediately cleared to '0'. |
| 1 | 1 | Not available |

NOTE) Where n = 0, 1, and 2.

**PWMCC** Clear PWM Generator Counter

| | |
|---|---|
| 0 | No effect |
| 1 | Clear the PWM generator counter (When write, automatically cleared "0" after being cleared counter) |

**PWMEN** Control PWM Generator

| | |
|---|---|
| 0 | PWM generator disable |
| 1 | PWM generator enable (Counter clear and start) |

NOTE) This bit shouldn't set to "1b" when EMGIFR = "1b".

**PWMCRL (PWM Generator Control Low Register) : 1078H (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PWMCK2 | PWMCK1 | PWMCK0 | PWMPOL | PWMMD1 | PWMMD0 | TRIGRS1 | TRIGRS0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**PWMCK[2:0]** Select PWM generator clock source. fx is system clock frequency.

| PWMCK2 | PWMCK1 | PWMCK0 | Description |
|---|---|---|---|
| 0 | 0 | 0 | fx/128 |
| 0 | 0 | 1 | fx/64 |
| 0 | 1 | 0 | fx/32 |
| 0 | 1 | 1 | fx/16 |
| 1 | 0 | 0 | fx/8 |
| 1 | 0 | 1 | fx/4 |
| 1 | 1 | 0 | fx/2 |
| 1 | 1 | 1 | fx/1 |

**PWMPOL** PWM Output Polarity Selection

| | |
|---|---|
| 0 | Start High (PWMOUT is low level at disable) |
| 1 | Start Low (PWMOUT is high level at disalbe) |

**PWMMD[1:0]** PWM Generator Operation Mode

| PWMMD1 | PWMMD0 | Description |
|---|---|---|
| 0 | 0 | PWM one-shot mode without auto-enable |
| 0 | 1 | PWM one-shot mode with auto-enable (The PWMEN bit is set by the external TRIG pin valid falling edge trigger) |
| 1 | x | PWM repeat mode |

**TRIGRS[1:0]** Select the external TRIG pin function with falling edge trigger

| TRIGRS1 | TRIGRS0 | Description |
|---|---|---|
| 0 | x | Disable the TRIG pin function |
| 1 | 0 | Enable the TRIG pin function without delay |
| 1 | 1 | Enable the TRIG pin function with delay |

**PGINTCR (PWM Generator Interrupt Control Register) : 103EH (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | POVIE | EMGIE | SHTIE | TRGIE |
| – | – | – | – | R/W | R/W | R/W | R/W |

Initial value : 00H

**POVIE** Enable or Disable PWM Generator Overflow Interrupt

0 Disable

1 Enable

**EMGIE** Enable or Disable Emergency Stop Interrupt

0 Disable

1 Enable

**SHTIE** Enable or Disable Shot Stop Interrupt

0 Disable

1 Enable

**TRGIE** Enable or Disable TRIG Interrupt

0 Disable

1 Enable

**PGIFLAG (PWM Generator Interrupt Flag Register) : 103FH (ESFR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | POVIFR | EMGIFR | SHTIFR | TRGIFR |
| – | – | – | – | R/W | R/W | R/W | R/W |

Initial value : 00H

**POVIFR** When PWM generator overflow interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software.

0 PWM generator overflow interrupt no generation

1 PWM generator overflow interrupt generation

**EMGIFR** When emergency stop interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software.

0 Emergency stop interrupt no generation

1 Emergency stop interrupt generation

**SHTIFR** When shot stop interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software.

0 Shot stop interrupt no generation

1 Shot stop interrupt generation

**TRGIFR** When TRIG interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software.

0 TRIG interrupt no generation

1 TRIG interrupt generation

## 11.9 Buzzer Driver

### 11.9.1 Overview

The Buzzer consists of 8 bit counter, buzzer data register (BUZDR), and buzzer control register (BUZCR). The Square Wave (61.035Hz~125.0 kHz @8MHz) is outputted through P46/SEG22/BUZO pin. The buzzer data register (BUZDR) controls the bsuzzer frequency (look at the following expression). In buzzer control register (BUZCR), BUCK[2:0] selects source clock divided by prescaler.

$$f_{BUZ}(Hz) = \frac{\text{Oscillator Frequency}}{2 \times \text{PrescalerRatio} \times (\text{BUZDR} + 1)}$$

**Table 11-15 Buzzer Frequency at 8 MHz**

| BUZDR[7:0] | Buzzer Frequency (kHz) | | | |
|---|---|---|---|---|
| | BUZCR[3:1]=000 | BUZCR[3:1]=001 | BUZCR[3:1]=010 | BUZCR[3:1]=011 |
| 0000_0000 | 125kHz | 62.5kHz | 31.25kHz | 15.625kHz |
| 0000_0001 | 62.5kHz | 31.25kHz | 15.625kHz | 7.812kHz |
| … | … | … | … | … |
| 1111_1101 | 492.126Hz | 246.063Hz | 123.031Hz | 61.515Hz |
| 1111_1110 | 490.196Hz | 245.098Hz | 122.549Hz | 61.274Hz |
| 1111_1111 | 488.281Hz | 244.141Hz | 122.07Hz | 61.035Hz |

### 11.9.2 Block Diagram

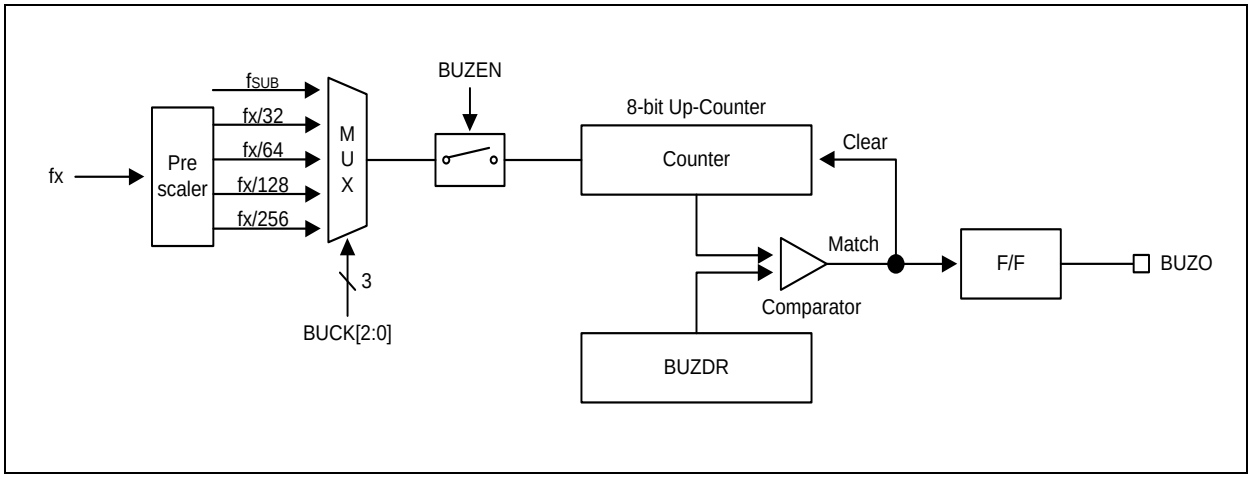



**Figure 11.49 Buzzer Driver Block Diagram**

### 11.9.3 Register Map

**Table 11-16 Buzzer Driver Register Map**

| Name | Address | Dir | Default | Description |
|---|---|---|---|---|
| BUZDR | 8FH | R/W | FFH | Buzzer Data Register |
| BUZCR | 97H | R/W | 00H | Buzzer Control Register |

### 11.9.4 Buzzer Driver Register Description

Buzzer driver consists of buzzer data register (BUZDR) and buzzer control register (BUZCR).

### 11.9.5 Register Description for Buzzer Driver

**BUZDR (Buzzer Data Register) : 8FH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| BUZDR7 | BUZDR6 | BUZDR5 | BUZDR4 | BUZDR3 | BUZDR2 | BUZDR1 | BUZDR0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : FFH

**BUZDR[7:0]** This bits control the Buzzer frequency
Its resolution is 00H ~ FFH

**BUZCR (Buzzer Control Register) : 97H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | BUCK2 | BUCK1 | BUCK0 | BUZEN |
| – | – | – | – | R/W | R/W | R/W | R/W |

Initial value : 00H

**BUCK[2:0]** Buzzer Driver Source Clock Selection

| BUCK2 | BUCK1 | BUCK0 | Description |
|---|---|---|---|
| 0 | 0 | 0 | fx/32 |
| 0 | 0 | 1 | fx/64 |
| 0 | 1 | 0 | fx/128 |
| 0 | 1 | 1 | fx/256 |
| 1 | x | x | $f_{SUB}$ (External Sub OSC) |

**BUZEN** Buzzer Driver Operation Control

| | |
|---|---|
| 0 | Buzzer Driver disable |
| 1 | Buzzer Driver enable |

NOTE) fx is the system clock frequency

## 11.10 SPI 2/3

### 11.10.1 Overview

There is serial peripheral interface (SPI 2/3) one channel in MC96F7864. The SPI 2/3 allows synchronous serial data transfer between the external serial devices. It can do Full-duplex communication by 4-wire (MOSI2/3, MISO2/3, SCK2/3, SS2/3), support master/slave mode, can select serial clock (SCK2/3) polarity, phase and whether LSB first data transfer or MSB first data transfer.

### 11.10.2 Block Diagram

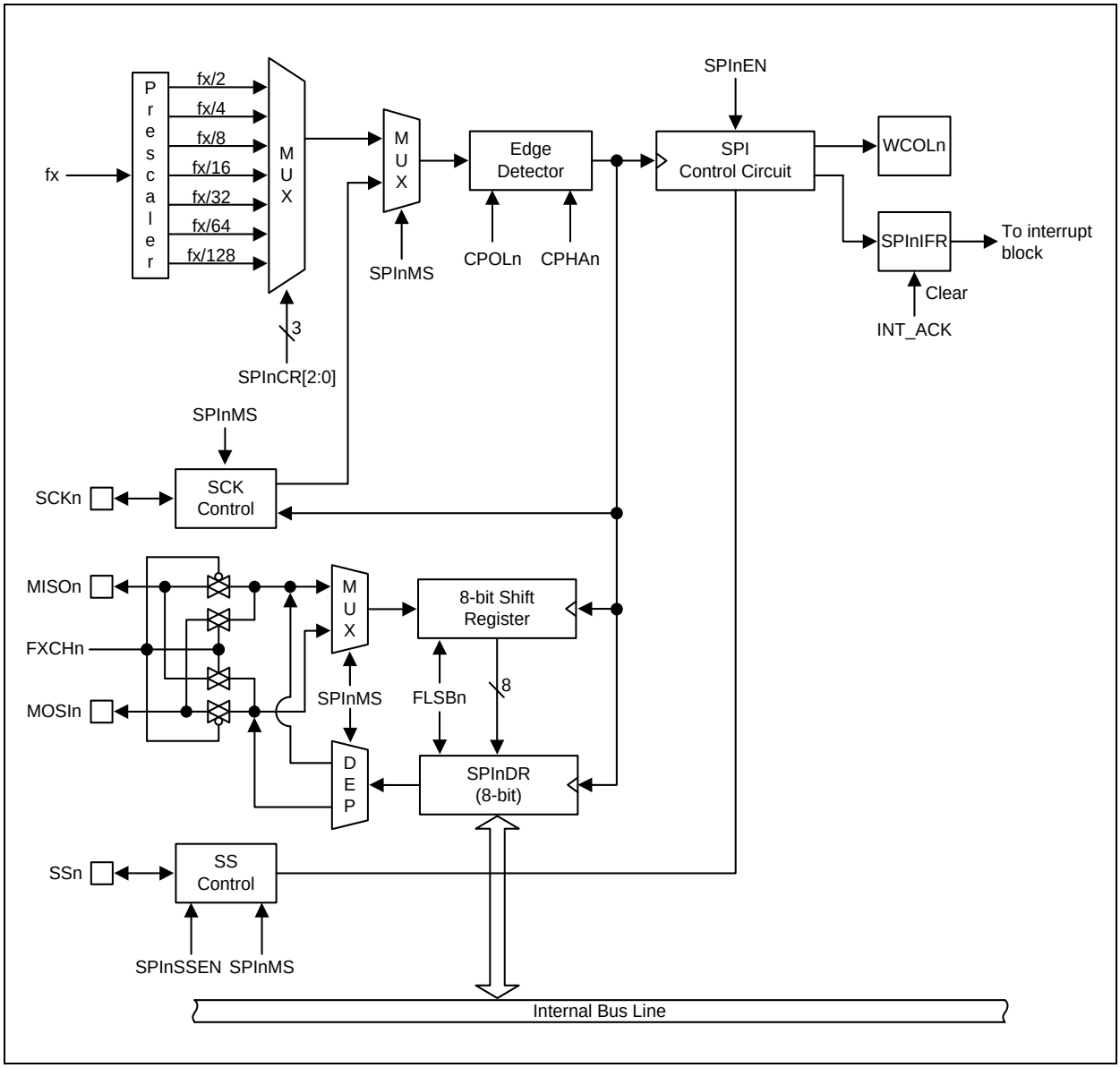



**Figure 11.50 SPI 2/3 Block Diagram (where n = 2 and 3)**

### 11.10.3 Data Transmit / Receive Operation

User can use SPI 2/3 for serial data communication by following step

1. Select SPI 2/3 operation mode(master/slave, polarity, phase) by control register SPInCR.
2. When the SPI 2/3 is configured as a Master, it selects a Slave by SS2/3 signal (active low).

   When the SPI 2/3 is configured as a Slave, it is selected by SS2/3 signal incoming from Master
3. When the user writes a byte to the data register SPInDR, SPI 2/3 will start an operation.
4. In this time, if the SPI 2/3 is configured as a Master, serial clock will come out of SCK2/3 pin. And Master shifts the eight bits into the Slave (transmit), Slave shifts the eight bits into the Master at the same time (receive). If the SPI 2/3 is configured as a Slave, serial clock will come into SCK2/3 pin. And Slave shifts the eight bits into the Master (transmit), Master shifts the eight bits into the Slave at the same time (receive).
5. When transmit/receive is done, SPInIFR bit will be set. If the SPI 2/3 interrupt is enabled, an interrupt is requested. And SPInIFR bit is cleared by hardware when executing the corresponding interrupt. If SPI 2/3 interrupt is disable, SPInIFR bit is cleared when user read the status register SPInSR, and then access (read/write) the data register SPInDR.

### 11.10.4 SS2/3 pin function

1. When the SPI 2/3 is configured as a Slave, the SS2/3 pin is always input. If LOW signal come into SS2/3 pin, the SPI 2/3 logic is active. And if ‘HIGH’ signal come into SS2/3 pin, the SPI 2/3 logic is stop. In this time, SPI 2/3 logic will be reset, and invalidated any received data.
2. When the SPI 2/3 is configured as a Master, the user can select the direction of the SS2/3 pin by port direction register (P36IO/P22IO). If the SS2/3 pin is configured as an output, user can use general P36IO/P22IO output mode. If the SS2/3 pin is configured as an input, ‘HIGH’ signal must come into SS2/3 pin to guarantee Master operation. If ‘LOW’ signal come into SS2/3 pin, the SPI 2/3 logic interprets this as another master selecting the SPI 2/3 as a slave and starting to send data to it. To avoid bus contention, MSB bit of SPICR will be cleared and the SPI 2/3 becomes a Slave and then, SPInIFR bit of SPInSR will be set, and if the SPI 2/3 interrupt is enabled, an interrupt is requested.

NOTES)

- When the SS2/3 pin is configured as an output at Master mode, SS2/3 pin’s output value is defined by user’s software (P36IO/P22IO). Before SPInCR setting, the direction of SS2/3 pin must be defined
- If you don’t need to use SS2/3 pin, clear the SPInSSEN bit of SPInSR. So, you can use disabled pin by P36IO/P22IO freely. In this case, SS2/3 signal is driven by ‘HIGH’ or ‘LOW’ internally. In other words, master is ‘HIGH’, salve is ‘LOW’
- When SS2/3 pin is configured as input, if ‘HIGH’ signal come into SS2/3 pin, SS_HIGH flag bit will be set. And you can clear it by writing ‘0’.

### 11.10.5 SPI 2 Timing Diagram

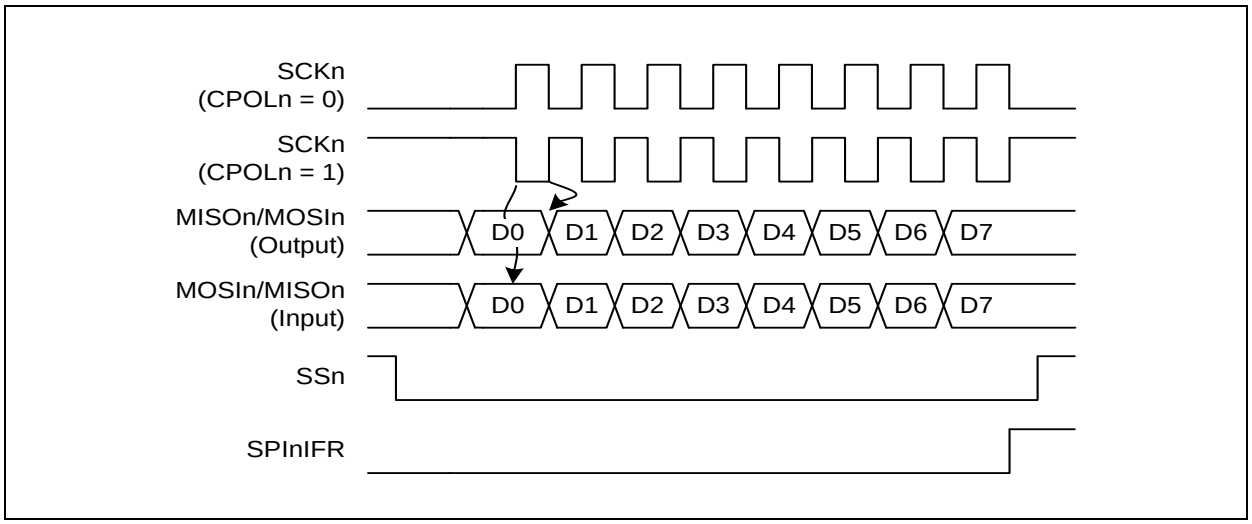



**Figure 11.51 SPI 2/3 Transmit/Receive Timing Diagram at CPHA = 0 (Where n = 2 and 3)**

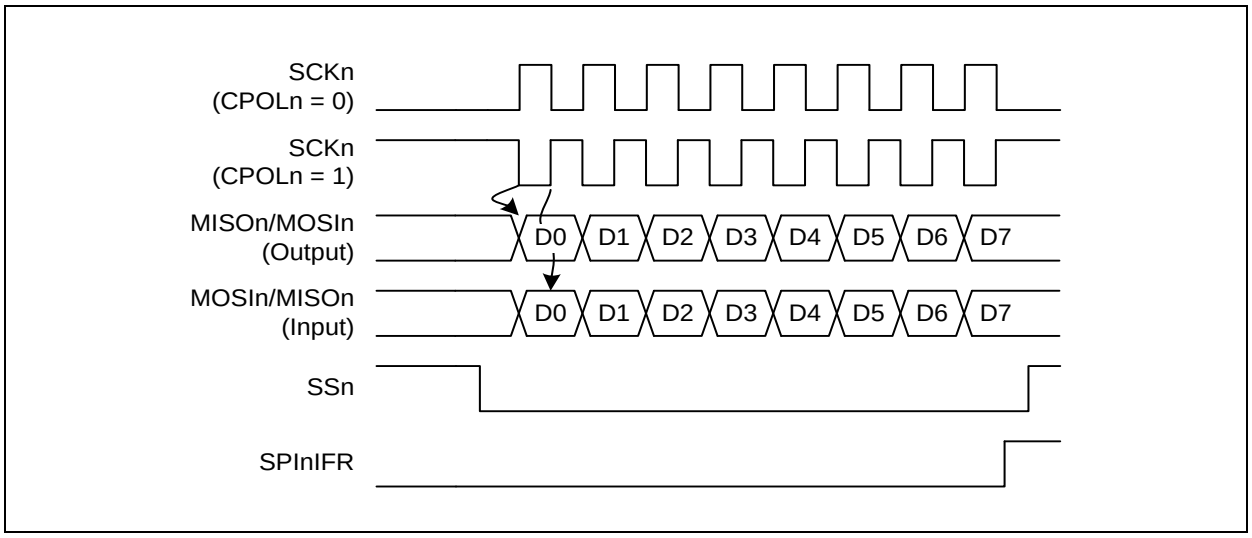



**Figure 11.52 SPI 2/3 Transmit/Receive Timing Diagram at CPHA = 1 (Where n = 2 and 3)**

### 11.10.6 Register Map

**Table 11-17 SPI 2/3 Register Map**

| Name | Address | Dir | Default | Description |
|---|---|---|---|---|
| SPInSR | EFH/F7H | R/W | 00H | SPI n Status Register |
| SPInDR | EEH/F6H | R/W | 00H | SPI n Data Register |
| SPInCR | EDH/F5H | R/W | 00H | SPI n Control Register |

**(where n = 2 and 3)**

### 11.10.7 SPI 2/3 Register Description

The SPI 2/3 register consists of SPI 2/3 control register (SPInCR), SPI 2/3 status register (SPInSR) and SPI 2/3 data register (SPInDR)

### 11.10.8 Register Description for SPI 2/3

**SPInDR (SPI 2/3 Data Register) : EEH/F6H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SPInDR7 | SPInDR6 | SPInDR5 | SPInDR4 | SPInDR3 | SPInDR2 | SPInDR1 | SPInDR0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**SPInDR [7:0]** SPI 2/3 Data
When it is written a byte to this data register, the SPI 2/3 will start an operation.

**SPInSR (SPI 2/3 Status Register) : EFH/F7H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SPIInFR | WCOLn | SS_HIGHn | – | FXCHn | SPInSSEN | – | – |
| R/W | R | R/W | – | R/W | R/W | – | – |

Initial value : 00H

**SPIIFRn** When SPI 2/3 Interrupt occurs, this bit becomes '1'. IF SPI 2/3 interrupt is enable, this bit is auto cleared by INT_ACK signal. And if SPI 2/3 Interrupt is disable, this bit is cleared when the status register SPnISR is read, and then access (read/write) the data register SPInDR

0 SPI 2/3 Interrupt no generation

1 SPI 2/3 Interrupt generation

**WCOLn** This bit is set if any data are written to the data register SPInDR during transfer. This bit is cleared when the status register SPInSR is read, and then access (read/write) the data register SPInDR

0 No collision

1 Collision

**SS_HIGHn** When the SS2/3 pin is configured as input, if "HIGH" signal comes into the pin, this flag bit will be set.

0 Cleared when '0' is written

1 No effect when '1' is written

**FXCHn** SPI 2/3 port function exchange control bit.

0 No effect

1 Exchange MOSIn and MISOn function

**SPInSSEN** This bit controls the SS2/3 pin operation

0 Disable

1 Enable (The corresponding pin should be a normal input)

**SPInCR (SPI 2/3 Control Register) : EDH/F5H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SPInEN | FLSBn | SPInMS | CPOLn | CPHAn | SPInDSCR | SPInSCR1 | SPInSCR0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**SPInEN** This bit controls the SPI 2/3 operation

0 Disable SPI 2/3 operation

1 Enable SPI 2/3 operation

**FLSBn** This bit selects the data transmission sequence

0 MSB first

1 LSB first

**SPInMS** This bit selects whether Master or Slave mode

0 Slave mode

1 Master mode

**CPOLn** This two bits control the serial clock (SCK2/3) mode.

**CPHAn** Clock polarity(CPOLn) bit determine SCK2/3's value at idle mode.

Clcok phase (CPHAn) bit determine if data are sampled on the leading or trailing edge of SCK2/3.

| CPOLn | CPHAn | Leading edge | Trailing edge |
|---|---|---|---|
| 0 | 0 | Sample (Rising) | Setup (Falling) |
| 0 | 1 | Setup (Rising) | Sample (Falling) |
| 1 | 0 | Sample (Falling) | Setup (Rising) |
| 1 | 1 | Setup (Falling) | Sample (Rising) |

**SPInDSCR SPInSCR [1:0]** These three bits select the SCK2/3 rate of the device configured as a master. When DSCR bit is written one, SCK2/3 will be doubled in master mode.

| SPInDSCR | SPInSCR 1 | SPInSCR 0 | SCKn frequency |
|---|---|---|---|
| 0 | 0 | 0 | fx/4 |
| 0 | 0 | 1 | fx/16 |
| 0 | 1 | 0 | fx/64 |
| 0 | 1 | 1 | fx/128 |
| 1 | 0 | 0 | fx/2 |
| 1 | 0 | 1 | fx/8 |
| 1 | 1 | 0 | fx/32 |
| 1 | 1 | 1 | fx/64 |

## 11.11 UART2/3/4

### 11.11.1 Overview

The universal asynchronous serial receiver and transmitter (UART2/3/4) is a highly flexible serial communication device. The main features are listed below.

- Full Duplex Operation (Independent Serial Receive and Transmit Registers)
- Baud Rate Generator
- Supports Serial Frames with 5, 6, 7, 8, or 9 Data Bits and 1 or 2 Stop Bits
- Odd or Even Parity Generation and Parity Check Supported by Hardware
- Data OverRun Detection
- Framing Error Detection
- Three Separate Interrupts on TX Complete, TX Data Register Empty and RX Complete

UART2/3/4 has baud rate generator, transmitter and receiver. The baud rate generator for asynchronous operation. The Transmitter consists of a single write buffer, a serial shift register, parity generator and control logic for handling different serial frame formats. The write buffer allows continuous transfer of data without any delay between frames. The receiver is the most complex part of the UART2/3/4 module due to its clock and data recovery units. The recovery unit is used for asynchronous data reception. In addition to the recovery unit, the receiver includes a parity checker, a shift register, a two-level receive FIFOn (UARTnDR) and control logic. The receiver supports the same frame formats as the transmitter and can detect frame error, data overrun and parity errors.

### 11.11.2 Block Diagram

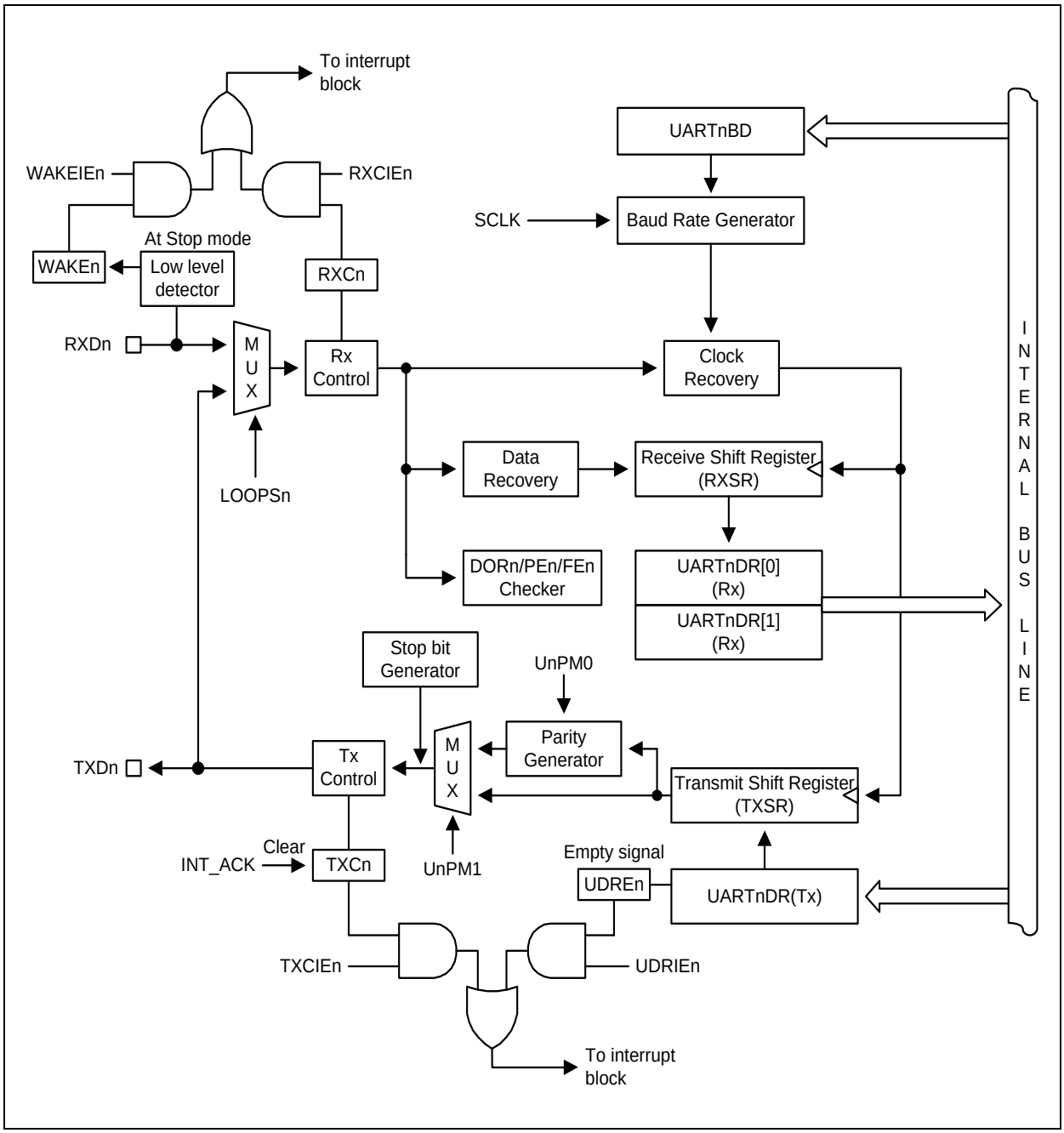



**Figure 11.53 UART Block Diagram(where n = 2,3, and 4)**

### 11.11.3 Clock Generation

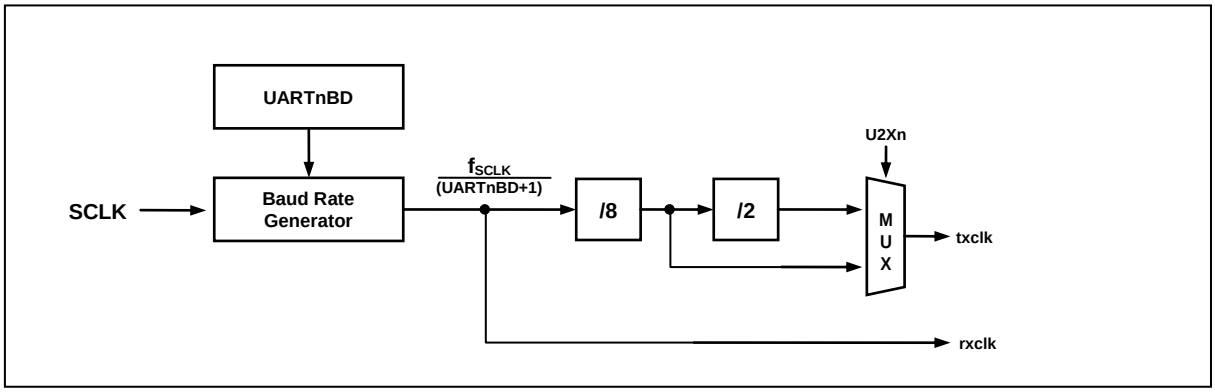



**Figure 11.54 Clock Generation Block Diagram (where n = 2,3, and 4)**

The clock generation logic generates the base clock for the transmitter and receiver.

Following table shows equations for calculating the baud rate (in bps).

**Table 11-18 Equations for Calculating Baud Rate Register Setting**

| Operating Mode | Equation for Calculating Baud Rate |
|---|---|
| Normal Mode(U2Xn=0) | $\text{Baud Rate} = \frac{\text{fx}}{16(\text{UARTnBD} + 1)}$ |
| Double Speed Mode(U2Xn=1) | $\text{Baud Rate} = \frac{\text{fx}}{8(\text{UARTnBD} + 1)}$ |

(where n = 2,3, and 4)

### 11.11.4 Data format

A serial frame is defined to be one character of data bits with synchronization bits (start and stop bits), and optionally a parity bit for error detection.

The UART2/3/4 supports all 30 combinations of the following as valid frame formats.

- 1 start bit
- 5, 6, 7, 8 or 9 data bits
- no, even or odd parity bit
- 1 or 2 stop bits

A frame starts with the start bit followed by the least significant data bit (LSB). Then the next data bits, up to nine, are succeeding, ending with the most significant bit (MSB). If parity function is enabled, the parity bit is inserted between the last data bit and the stop bit. A high-to-low transition on data pin is considered as start bit. When a complete frame is transmitted, it can be directly followed by a new frame, or the communication line can be set to an idle state. The idle means high state of data pin. The following figure shows the possible combinations of the frame formats. Bits inside brackets are optional.

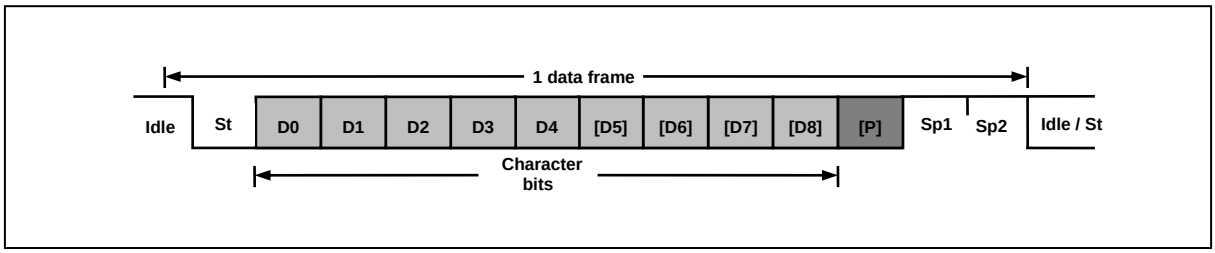



**Figure 11.55 Frame Format**

1 data frame consists of the following bits

- Idle No communication on communication line (TxD/RxD)
- St Start bit (Low)
- Dn Data bits (0~8)
- Parity bit ---------- Even parity, Odd parity, No parity
- Stop bit(s) ---------- 1 bit or 2 bits

The frame format used by the UART2/3/4 is set by the UnSIZE[2:0], UnPM[1:0] and USBSn bits in UARTnCR1 and UARTnCR3 register. The Transmitter and Receiver use the same setting.

### 11.11.5 Parity bit

The parity bit is calculated by doing an exclusive-OR of all the data bits. If odd parity is used, the result of the exclusive-or is inverted. The parity bit is located between the MSB and first stop bit of a serial frame.

$P_{even} = D_{n-1} \wedge \ldots \wedge D_3 \wedge D_2 \wedge D_1 \wedge D_0 \wedge 0$

$P_{odd} = D_{n-1} \wedge \ldots \wedge D_3 \wedge D_2 \wedge D_1 \wedge D_0 \wedge 1$

$P_{even}$ : Parity bit using even parity

$P_{odd}$ : Parity bit using odd parity

$D_n$ : Data bit n of the character

### 11.11.6 UART2/3/4 Transmitter

The UART2/3/4 transmitter is enabled by setting the TXEn bit in UARTnCR2 register. When the Transmitter is enabled, the TXDn pin should be set to TXDn function for the serial output pin of UART2/3/4 by the P6FSRH[1:0]/P5FSRL[5:4]/P5FSRL[2:1]. The baud-rate, operation mode and frame format must be setup once before doing any transmission.

#### 11.11.6.1 Sending Tx data

A data transmission is initiated by loading the transmit buffer (UARTnDR register I/O location) with the data to be transmitted. The data written in transmit buffer is moved to the shift register when the shift register is ready to send a new frame. The shift register is loaded with the new data if it is in idle state or immediately after the last stop bit of the previous frame is transmitted. When the shift register is loaded with new data, it will transfer one complete frame according to the settings of control registers. If the 9-bit characters are used, the ninth bit must be written to the UnTX8 bit in UARTnCR3 register before it is loaded to the transmit buffer (UARTnDR register).

#### 11.11.6.2 Transmitter flag and interrupt

The UART2/3/4 transmitter has 2 flags which indicate its state. One is UART2/3/4 data register empty flag (UDREn) and the other is transmit complete flag (TXCn). Both flags can be interrupt sources.

UDREn flag indicates whether the transmit buffer is ready to receive new data. This bit is set when the transmit buffer is empty and cleared when the transmit buffer contains data to be transmitted but has not yet been moved into the shift register. And also this flag can be cleared by writing ‘0’ to this bit position. Writing ‘1’ to this bit position is prevented.

When the data register empty interrupt enable (UDRIEn) bit in UARTnCR2 register is set and the global interrupt is enabled, UART2/3/4 data register empty interrupt is generated while UDREn flag is set.

The transmit complete (TXCn) flag bit is set when the entire frame in the transmit shift register has been shifted out and there is no more data in the transmit buffer. The TXCn flag is automatically cleared when the transmit complete interrupt service routine is executed, or it can be cleared by writing ‘0’ to TXCn bit in UARTnST register.

When the transmit complete interrupt enable (TXCIEn) bit in UARTnCR2 register is set and the global interrupt is enabled, UART2/3/4 transmit complete interrupt is generated while TXCn flag is set.

### 11.11.6.3 Parity Generator

The parity generator calculates the parity bit for the serial frame data to be sent. When parity bit is enabled (UnPM[1]=1), the transmitter control logic inserts the parity bit between the MSB and the first stop bit of the frame to be sent.

### 11.11.6.4 Disabling Transmitter

Disabling the transmitter by clearing the TXEn bit will not become effective until ongoing transmission is completed. When the Transmitter is disabled, the TXDn pin can be used as a normal general purpose I/O (GPIO).

### 11.11.7 UART Receiver

The UART2/3/4 receiver is enabled by setting the RXEn bit in the UARTnCR2 register. When the receiver is enabled, the RXDn pin should be set to the input port for the serial input pin of UART2/3/4 by P65IO/ P52IO/ P50IO bit. The baud-rate, mode of operation and frame format must be set before serial reception.

### 11.11.7.1 Receiving Rx data

The receiver starts data reception when it detects a valid start bit (LOW) on RXDn pin. Each bit after start bit is sampled at pre-defined baud-rate (asynchronous) and shifted into the receive shift register until the first stop bit of a frame is received. Even if there's $2^{nd}$ stop bit in the frame, the $2^{nd}$ stop bit is ignored by the receiver. That is, receiving the first stop bit means that a complete serial frame is present in the receiver shift register and contents of the shift register are to be moved into the receive buffer. The receive buffer is read by reading the UARTnDR register.

If 9-bit characters are used (UnSIZE[2:0] = "111"), the ninth bit is stored in the RX8n bit position in the UARTnCR3 register. The $9^{th}$ bit must be read from the RX8n bit before reading the low 8 bits from the UARTnDR register. Likewise, the error flags FEn, DORn, PEn must be read before reading the data from UARTnDR register. It's because the error flags are stored in the same FIFO position of the receive buffer.

### 11.11.7.2 Receiver Flag and Interrupt

The UART2/3/4 receiver has one flag that indicates the receiver state.

The receive complete (RXCn) flag indicates whether there are unread data in the receive buffer. This flag is set when there are unread data in the receive buffer and cleared when the receive buffer is empty. If the receiver is disabled (RXEn=0), the receiver buffer is flushed and the RXCn flag is cleared.

When the receive complete interrupt enable (RXCIEn) bit in the UARTnCR2 register is set and global interrupt is enabled, the UART2/3/4 receiver complete interrupt is generated while RXCn flag is set.

The UART2/3/4 receiver has three error flags which are frame error (FEn), data overrun (DORn) and parity error (PEn). These error flags can be read from the UARTnST register. As received data are stored in the 2-level receive buffer, these error flags are also stored in the same position of receive buffer. So, before reading received data from UARTnDR register, read the UARTnST register first which contains error flags.

The frame error (FEn) flag indicates the state of the first stop bit. The FEn flag is ‘0’ when the stop bit was correctly detected as ‘1’, and the FE flag is ‘1’ when the stop bit was incorrect, i.e. detected as ‘0’. This flag can be used for detecting out-of-sync conditions between data frames.

The data overrun (DORn) flag indicates data loss due to a receive buffer full condition. DORn occurs when the receive buffer is full, and another new data is present in the receive shift register which are to be stored into the receive buffer. After the DORn flag is set, all the incoming data are lost. To prevent data loss or clear this flag, read the receive buffer.

The parity error (PEn) flag indicates that the frame in the receive buffer had a parity error when received. If parity check function is not enabled (UnPM[1]=0), the PEn bit is always read ‘0’.

### 11.11.7.3 Parity Checker

If parity bit is enabled (UnPM[1]=1), the Parity Checker calculates the parity of the data bits in incoming frame and compares the result with the parity bit from the received serial frame.

### 11.11.7.4 Disabling Receiver

In contrast to transmitter, disabling the Receiver by clearing RXEn bit makes the Receiver inactive immediately. When the receiver is disabled, the receiver flushes the receive buffer, the remaining data in the buffer is all reset, and the RXDn pin can be used as a normal general purpose I/O (GPIO).

### 11.11.7.5 Asynchronous Data Reception

To receive asynchronous data frame, the UART includes a clock and data recovery unit. The clock recovery logic is used for synchronizing the internally generated baud-rate clock to the incoming asynchronous serial frame on the RXDn pin.

The data recovery logic samples and low pass filters the incoming bits, and this removes the noise of RXDn pin.

The next figure illustrates the sampling process of the start bit of an incoming frame. The sampling rate is 16 times of the baud-rate in normal mode and 8 times the baud-rate for double speed mode (U2Xn=1). The horizontal arrows show the synchronization variation due to the asynchronous sampling process. Note that larger time variation is shown when using the double speed mode.

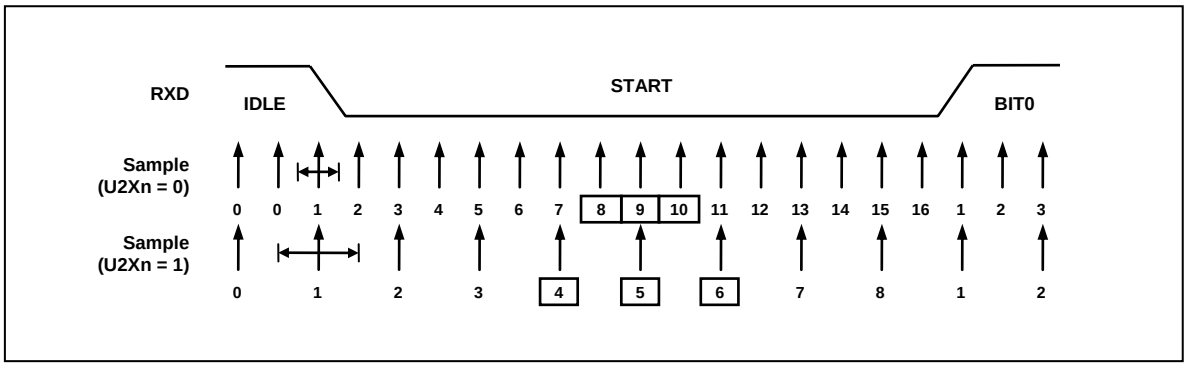



**Figure 11.56 Start Bit Sampling (where n = 2,3, and 4)**

When the receiver is enabled (RXEn=1), the clock recovery logic tries to find a high-to-low transition on the RXDn line, the start bit condition. After detecting high to low transition on RXDn line, the clock recovery logic uses samples 8,9, and 10 for normal mode to decide if a valid start bit is received. If more than 2 samples have logical low level, it is considered that a valid start bit is detected and the internally generated clock is synchronized to the incoming data frame. And the data recovery can begin. The synchronization process is repeated for each start bit.

As described above, when the receiver clock is synchronized to the start bit, the data recovery can begin. Data recovery process is almost similar to the clock recovery process. The data recovery logic samples 16 times for each incoming bits for normal mode and 8 times for double speed mode, and uses sample 8, 9, and 10 to decide data value. If more than 2 samples have low levels, the received bit is considered to a logic ‘0’ and if more than 2 samples have high levels, the received bit is considered to a logic ‘1’. The data recovery process is then repeated until a complete frame is received including the first stop bit. The decided bit value is stored in the receive shift register in order. Note that the Receiver only uses the first stop bit of a frame. Internally, after receiving the first stop bit, the Receiver is in idle state and waiting to find start bit.

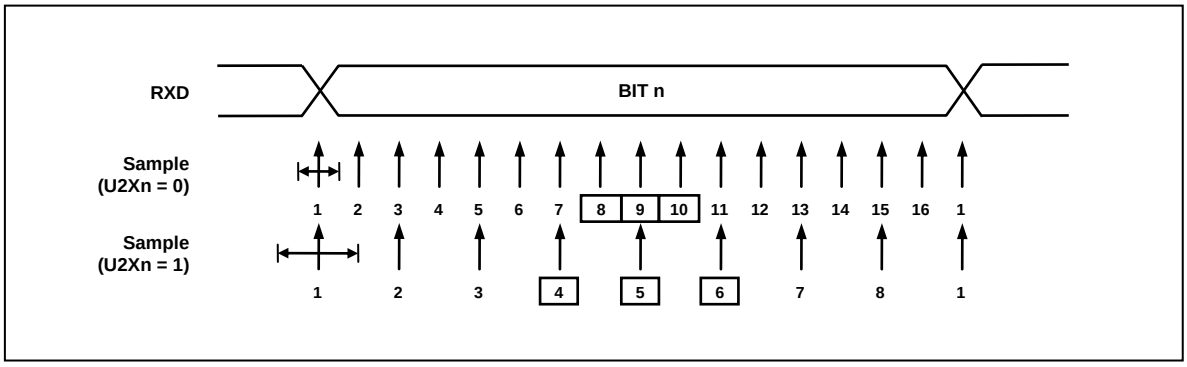



**Figure 11.57 Sampling of Data and Parity Bit (where n = 2,3, and 4)**

The process for detecting stop bit is like clock and data recovery process. That is, if 2 or more samples of 3 center values have high level, correct stop bit is detected, else a frame error (FEn) flag is set. After deciding whether the first stop bit is valid or not, the Receiver goes to idle state and monitors the RXD line to check a valid high to low transition is detected (start bit detection).

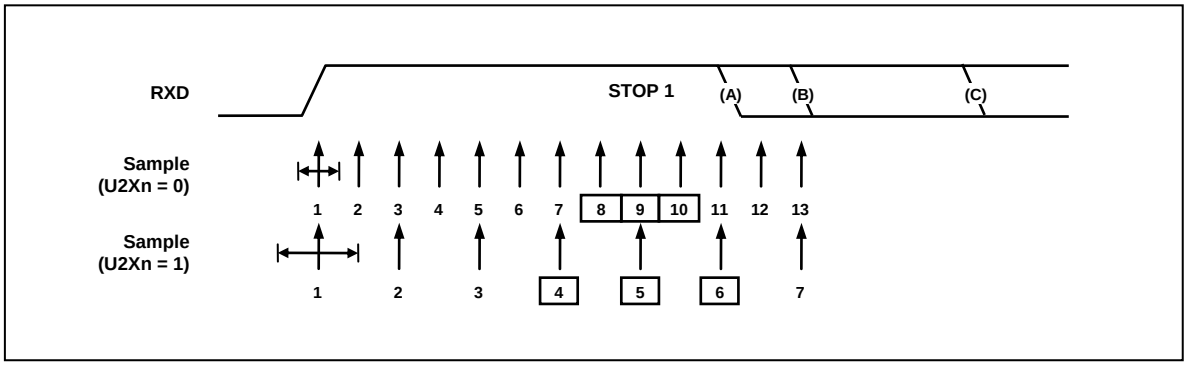



**Figure 11.58 Stop Bit Sampling and Next Start Bit Sampling (where n = 2,3, and 4)**

## 11.11.8 Register Map

**Table 11-19 UART Register Map (where n = 2,3, and 4)**

| Name | Address | Dir | Default | Description |
|---|---|---|---|---|
| UARTnBD | 103CH/1044H/104CH (ESFR) | R/W | FFH | UARTn Baud Rate Generation Register |
| UARTnDR | 103DH/1045H/104DH (ESFR) | R/W | 00H | UARTn Data Register |
| UARTnCR1 | 1038H/1040H/1048H (ESFR) | R/W | 00H | UARTn Control Register 1 |
| UARTnCR2 | 1039H/1041H/1049H (ESFR) | R/W | 00H | UARTn Control Register 2 |
| UARTnCR3 | 103AH/1042H/104AH (ESFR) | R/W | 00H | UARTn Control Register 3 |
| UARTnST | 103BH/1043H/104BH (ESFR) | R/W | 80H | UARTn Status Register |

## 11.11.9 UART Register Description

UART2/3/4 module consists of UART2/3/4 baud rate generation register (UARTnBD), UART2/3/4 data register (UARTnDR), UART2/3/4 control register 1 (UARTnCR1), UART2/3/4 control register 2 (UARTnCR2) ,UART2/3/4 control register 3 (UARTnCR3), and UART2/3/4 status register (UARTnST).

## 11.11.10 Register Description for UART2/3/4

**UARTnBD (UARTn Baud Rate Generation Register) : 103CH/1044H/104CH (ESFR), Where n = 2, 3, and 4**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| UARTnBD7 | UARTnBD6 | UARTnBD5 | UARTnBD4 | UARTnBD3 | UARTnBD2 | UARTnBD1 | UARTnBD0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : FFH

**UARTnBD [7:0]** The value in this register is used to generate internal baud rate. To prevent malfunction, do not write '0'.

**UARTnDR (UARTn Data Register) : 103DH/1045H/104DH (ESFR), Where n = 2, 3, and 4**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| UARTnDR7 | UARTnDR6 | UARTnDR5 | UARTnDR4 | UARTnDR3 | UARTnDR2 | UARTnDR1 | UARTnDR0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**UARTnDR [7:0]** The UARTn Transmit Buffer and Receive Buffer share the same I/O address with this DATA register. The Transmit Data Buffer is the destination for data written to the UARTnDR register. Reading the UARTnDR register returns the contents of the Receive Buffer.

Write this register only when the UDREn flag is set.

**UARTnCR1 (UARTn Control Register 1) : 1038H/1040H/1048H (ESFR), Where n = 2, 3, and 4**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| - | - | UnPM1 | UnPM0 | UnSIZE2 | UnSIZE1 | UnSIZE0 | - |
| - | - | R/W | R/W | R/W | R/W | R/W | - |

Initial value : 00H

**UnPM[1:0]** Selects Parity Generation and Check methods

| UnPM1 | UnPM0 | Parity |
|---|---|---|
| 0 | 0 | No Parity |
| 0 | 1 | Reserved |
| 1 | 0 | Even Parity |
| 1 | 1 | Odd Parity |

**UnSIZE[2:0]** Selects the Length of Data Bits in Frame

| UnSIZE2 | UnSIZE1 | UnSIZE0 | Data Length |
|---|---|---|---|
| 0 | 0 | 0 | 5 bit |
| 0 | 0 | 1 | 6 bit |
| 0 | 1 | 0 | 7 bit |
| 0 | 1 | 1 | 8 bit |
| 1 | 1 | 1 | 9 bit |
| Other values | | | Reserved |

**UARTnCR2 (UARTn Control Register 2) : 1039H/1041H/1049H (ESFR), Where n = 2, 3, and 4**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| UDRIEn | TXCIEn | RXCIEn | WAKEIEn | TXEn | RXEn | UARTnEN | U2Xn |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**UDRIEn** Interrupt enable bit for UARTn Data Register Empty

0 Interrupt from UDREn is inhibited (use polling)

1 When UDREn is set, request an interrupt

**TXCIEn** Interrupt enable bit for Transmit Complete

0 Interrupt from TXCn is inhibited (use polling)

1 When TXCn is set, request an interrupt

**RXCIEn** Interrupt enable bit for Receive Complete

0 Interrupt from RXCn is inhibited (use polling)

1 When RXCn is set, request an interrupt

**WAKEIEn** Interrupt enable bit for Wake in STOP mode. When device is in stop mode, if RXDn goes to LOW level an interrupt can be requested to wake-up system. At that time the UDRIEn bit and UARTnST register value should be set to ‘0b’ and “00H”, respectively.

0 Interrupt from Wake is inhibited

1 When WAKEn is set, request an interrupt

**TXEn** Enables the transmitter unit

0 Transmitter is disabled

1 Transmitter is enabled

**RXEn** Enables the receiver unit

0 Receiver is disabled

1 Receiver is enabled

**UARTnEN** Activate UARTn module by supplying clock. When one of TXEn and RXEn values is “1”, the UARTnEN bit always set to “1”.

0 UARTn is disabled (clock is halted)

1 UARTn is enabled

**U2Xn** This bit selects receiver sampling rate.

0 Normal operation

1 Double Speed operation

**UARTnCR3 (UARTn Control Register 3) : 103AH/1042H/104AH (ESFR), Where n = 2, 3, and 4**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| - | LOOPSn | - | - | - | USBSn | UnTX8 | UnRX8 |
| - | R/W | - | - | - | R/W | R/W | R |

Initial value : 00H

**LOOPSn** Controls the Loop Back Mode of UARTn, for test mode

0 Normal operation

1 Loop Back mode

**USBSn** Selects the length of stop bit.

0 1 Stop Bit

1 2 Stop Bit

**UnTX8** The ninth bit of data frame in UARTn. Write this bit first before loading the UARTnDR register

0 MSB ($9^{th}$ bit) to be transmitted is '0'

1 MSB ($9^{th}$ bit) to be transmitted is '1'

**UnRX8** The ninth bit of data frame in UARTn. Read this bit first before reading the receive buffer

0 MSB ($9^{th}$ bit) received is '0'

1 MSB ($9^{th}$ bit) received is '1'

**UARTnST (UARTn Status Register) : 103BH/1043H/104BH (ESFR), Where n = 2, 3, and 4**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| UDREn | TXCn | RXCn | WAKEn | SOFTRSTn | DORn | FEn | PEn |
| R/W | R/W | R | R/W | R/W | R | R/W | R/W |

Initial value : 80H

**UDREn** The UDREn flag indicates if the transmit buffer (UARTnDR) is ready to receive new data. If UDREn is '1', the buffer is empty and ready to be written. This flag can generate a UDREn interrupt.

0 Transmit buffer is not empty.

1 Transmit buffer is empty.

**TXCn** This flag is set when the entire frame in the transmit shift register has been shifted out and there is no new data currently present in the transmit buffer. This flag is automatically cleared when the interrupt service routine of a TXCn interrupt is executed. This flag can generate a TXCn interrupt.

0 Transmission is ongoing.

1 Transmit buffer is empty and the data in transmit shift register are shifted out completely.

**RXCn** This flag is set when there are unread data in the receive buffer and cleared when all the data in the receive buffer are read. The RXCn flag can be used to generate a RXCn interrupt.

0 There is no data unread in the receive buffer

1 There are more than 1 data in the receive buffer

**WAKEn** This flag is set when the RXDn pin is detected low while the CPU is in stop mode. This flag can be used to generate a WAKEn interrupt. This bit should be cleared by program software.

0 No WAKEn interrupt is generated.

1 WAKEn interrupt is generated.

**SOFTRSTn** This is an internal reset and only has effect on UARTn. Writing '1' to this bit initializes the internal logic of UARTn and this bit is automatically cleared.

0 No operation

1 Reset UARTn

**DORn** This bit is set if a Data OverRun occurs. While this bit is set, the incoming data frame is ignored. This flag is valid until the receive buffer is read.

0 No Data OverRun

1 Data OverRun detected

**FEn** This bit is set if the first stop bit of next character in the receive buffer is detected as '0'. This bit is valid until the receive buffer is read.

0 No Frame Error

1 Frame Error detected

**PEn** This bit is set if the next character in the receive buffer has a Parity Error to be received while Parity Checking is enabled. This bit is valid until the receive buffer is read.

0 No Parity Error

1 Parity Error detected

## 11.11.11 Baud Rate setting (example)

**Table 11-20 Examples of UARTnBD Settings for Commonly Used Oscillator Frequencies**

| Baud Rate | fx=1.00MHz | | fx=1.8432MHz | | fx=2.00MHz | |
|---|---|---|---|---|---|---|
| | UARTnBD | ERROR | UARTnBD | ERROR | UARTnBD | ERROR |
| 2400 | 25 | 0.2% | 47 | 0.0% | 51 | 0.2% |
| 4800 | 12 | 0.2% | 23 | 0.0% | 25 | 0.2% |
| 9600 | 6 | -7.0% | 11 | 0.0% | 12 | 0.2% |
| 14.4k | 3 | 8.5% | 7 | 0.0% | 8 | -3.5% |
| 19.2k | 2 | 8.5% | 5 | 0.0% | 6 | -7.0% |
| 28.8k | 1 | 8.5% | 3 | 0.0% | 3 | 8.5% |
| 38.4k | 1 | -18.6% | 2 | 0.0% | 2 | 8.5% |
| 57.6k | - | - | 1 | -25.0% | 1 | 8.5% |
| 76.8k | - | - | 1 | 0.0% | 1 | -18.6% |
| 115.2k | - | - | - | - | - | - |
| 230.4k | - | - | - | - | - | - |

(continued)

| Baud Rate | fx=3.6864MHz | | fx=4.00MHz | | fx=7.3728MHz | |
|---|---|---|---|---|---|---|
| | UARTnBD | ERROR | UARTnBD | ERROR | UARTnBD | ERROR |
| 2400 | 95 | 0.0% | 103 | 0.2% | 191 | 0.0% |
| 4800 | 47 | 0.0% | 51 | 0.2% | 95 | 0.0% |
| 9600 | 23 | 0.0% | 25 | 0.2% | 47 | 0.0% |
| 14.4k | 15 | 0.0% | 16 | 2.1% | 31 | 0.0% |
| 19.2k | 11 | 0.0% | 12 | 0.2% | 23 | 0.0% |
| 28.8k | 7 | 0.0% | 8 | -3.5% | 15 | 0.0% |
| 38.4k | 5 | 0.0% | 6 | -7.0% | 11 | 0.0% |
| 57.6k | 3 | 0.0% | 3 | 8.5% | 7 | 0.0% |
| 76.8k | 2 | 0.0% | 2 | 8.5% | 5 | 0.0% |
| 115.2k | 1 | 0.0% | 1 | 8.5% | 3 | 0.0% |
| 230.4k | - | - | - | - | 1 | 0.0% |
| 250k | - | - | - | - | 1 | -7.8% |
| 0.5M | - | - | - | - | - | - |

(continued)

| Baud Rate | fx=8.00MHz | | fx=11.0592MHz | |
|---|---|---|---|---|
| | UARTnBD | ERROR | UARTnBD | ERROR |
| 2400 | 207 | 0.2% | - | - |
| 4800 | 103 | 0.2% | 143 | 0.0% |
| 9600 | 51 | 0.2% | 71 | 0.0% |
| 14.4k | 34 | -0.8% | 47 | 0.0% |
| 19.2k | 25 | 0.2% | 35 | 0.0% |
| 28.8k | 16 | 2.1% | 23 | 0.0% |
| 38.4k | 12 | 0.2% | 17 | 0.0% |
| 57.6k | 8 | -3.5% | 11 | 0.0% |
| 76.8k | 6 | -7.0% | 8 | 0.0% |
| 115.2k | 3 | 8.5% | 5 | 0.0% |
| 230.4k | 1 | 8.5% | 2 | 0.0% |
| 250k | 1 | 0.0% | 2 | -7.8% |
| 0.5M | - | - | - | - |
| 1M | - | - | - | - |

Where n = 2,3, and 4

## 11.12 USI0/1 (UART + SPI + I2C)

### 11.12.1 Overview

The USI0/1 consists of USI0/1 control register1/2/3/4, USI0/1 status register 1/2, USI0/1 baud-rate generation register, USI0/1 data register, USI0/1 SDA hold time register, USI0/1 SCL high period register, USI0/1 SCL low period register, and USI0/1 slave address register (USInCR1, USInCR2, USInCR3, USInCR4, USInST1, USInST2, USInBD, USInDR, USInSDHR, USInSCHR, USInSCLR, USInSAR).

The operation mode is selected by the operation mode of USI0/1 selection bits (USInMS[1:0]).

It has four operating modes:

- Asynchronous mode (UART)
- Synchronous mode
- SPI mode
- I2C mode

### 11.12.2 USI0/1 UART Mode

The universal synchronous and asynchronous serial receiver and transmitter (UART) is a highly flexible serial communication device. The main features are listed below.

- Full Duplex Operation (Independent Serial Receive and Transmit Registers)
- Asynchronous or Synchronous Operation
- Baud Rate Generator
- Supports Serial Frames with 5,6,7,8, or 9 Data Bits and 1 or 2 Stop Bits
- Odd or Even Parity Generation and Parity Check Supported by Hardware
- Data OverRun Detection
- Framing Error Detection
- Three Separate Interrupts on TX Complete, TX Data Register Empty and RX Complete
- Double Speed Asynchronous communication mode

USI0/1 has three main parts of clock generator, Transmitter and receiver. The clock generation logic consists of synchronization logic for external clock inut used by synchronous or SPI slave operation, and the baud rate generator for asynchronous or master (synchronous or SPI) operation.

The Transmitter consists of a single write buffer, a serial shift register, parity generator and control logic for handling different serial frame formats. The write buffer allows continuous transfer of data without any delay between frames. The receiver is the most complex part of the UART module due to its clock and data recovery units. The recovery unit is used for asynchronous data reception. In addition to the recovery unit, the receiver includes a parity checker, a shift register, a two-level receive FIFO (USInDR) and control logic. The receiver supports the same frame formats as the transmitter and can detect frame error, data overrun and parity errors.

### 11.12.3 USI0/1 UART Block Diagram

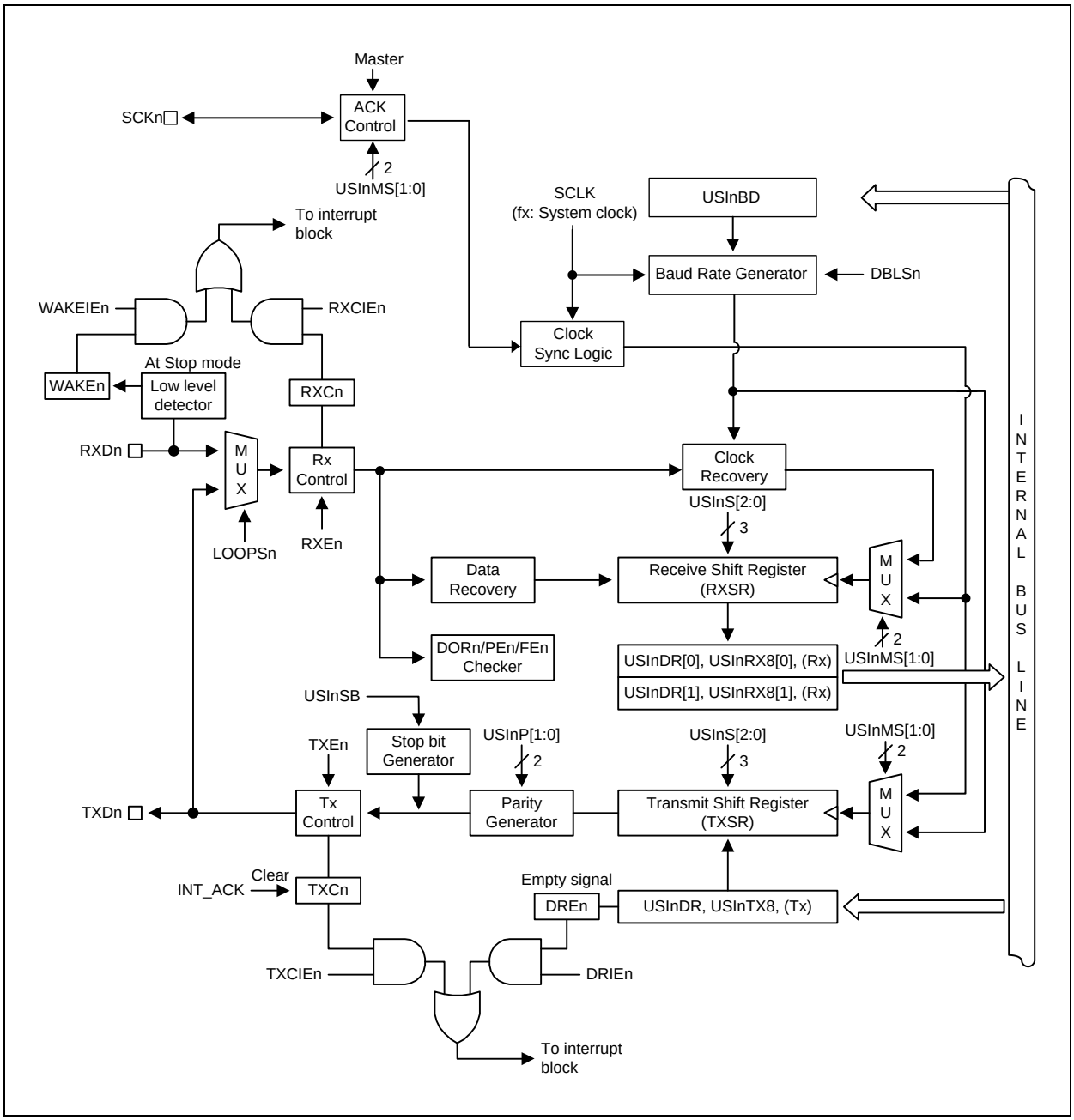



**Figure 11.59 USI0/1 UART Block Diagram (Where n = 0 and 1)**

### 11.12.4 USI0/1 Clock Generation

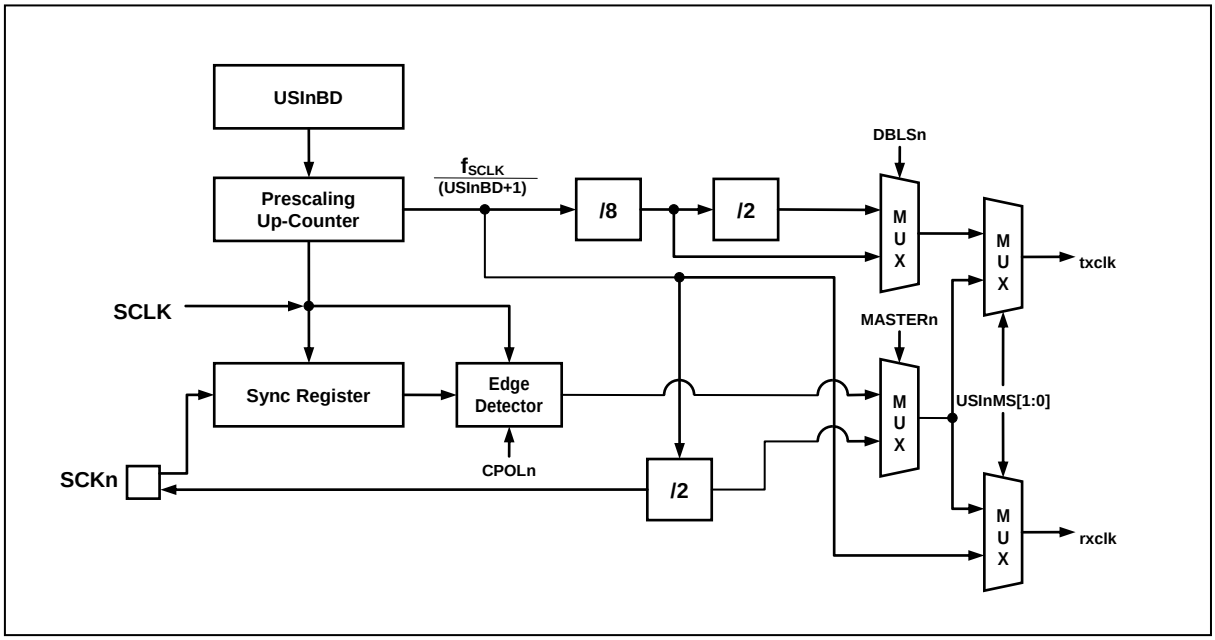



**Figure 11.60 Clock Generation Block Diagram (USIn, where n = 0 and 1)**

The clock generation logic generates the base clock for the transmitter and receiver. The USI0/1 supports four modes of clock operation and those are normal asynchronous, double speed asynchronous, master synchronous and slave synchronous mode. The clock generation scheme for master SPI and slave SPI mode is the same as master synchronous and slave synchronous operation mode. The USInMS[1:0] bits in USInCR1 register selects asynchronous or synchronous operation. Asynchronous double speed mode is controlled by the DBLSn bit in the USInCR2 register. The MASTERn bit in USInCR3 register controls whether the clock source is internal (master mode, output pin) or external (slave mode, input pin). The SCKn pin is active only when the USI0/1 operates in synchronous or SPI mode.

Following table shows the equations for calculating the baud rate (in bps).

**Table 11-21 Equations for Calculating USI0/1 Baud Rate Register Setting**

| Operating Mode | Equation for Calculating Baud Rate |
|---|---|
| Asynchronous Normal Mode (DBLSn=0) | $\text{Baud Rate} = \frac{\text{fx}}{16(\text{USInBD} + 1)}$ |
| Asynchronous Double Speed Mode (DBLSn=1) | $\text{Baud Rate} = \frac{\text{fx}}{8(\text{USInBD} + 1)}$ |
| Synchronous or SPI Master Mode | $\text{Baud Rate} = \frac{\text{fx}}{2(\text{USInBD} + 1)}$ |

### 11.12.5 USI0/1 External Clock (SCKn)

External clocking is used in the synchronous mode of operation.

External clock input from the SCKn pin is sampled by a synchronization logic to remove meta-stability. The output from the synchronization logic must be passed through an edge detector before it is used by the transmitter and receiver. This process introduces two CPU clock period delay. The maximum frequency of the external SCKn pin is limited up-to 1MHz.

### 11.12.6 USI0 Synchronous mode operation

When synchronous or SPI mode is used, the SCKn pin will be used as either clock input (slave) or clock output (master). Data sampling and transmitter is issued on the different edge of SCKn clock each other. For example, if data input on RXDn (MISOn in SPI mode) pin is sampled on the rising edge of SCKn clock, data output on TXDn (MOSIn in SPI mode) pin is altered on the falling edge.

The CPOLn bit in USInCR1 register selects which SCKn clock edge is used for data sampling and which is used for data change. As shown in the figure below, when CPOLn is zero, the data will be changed at rising SCKn edge and sampled at falling SCKn edge.

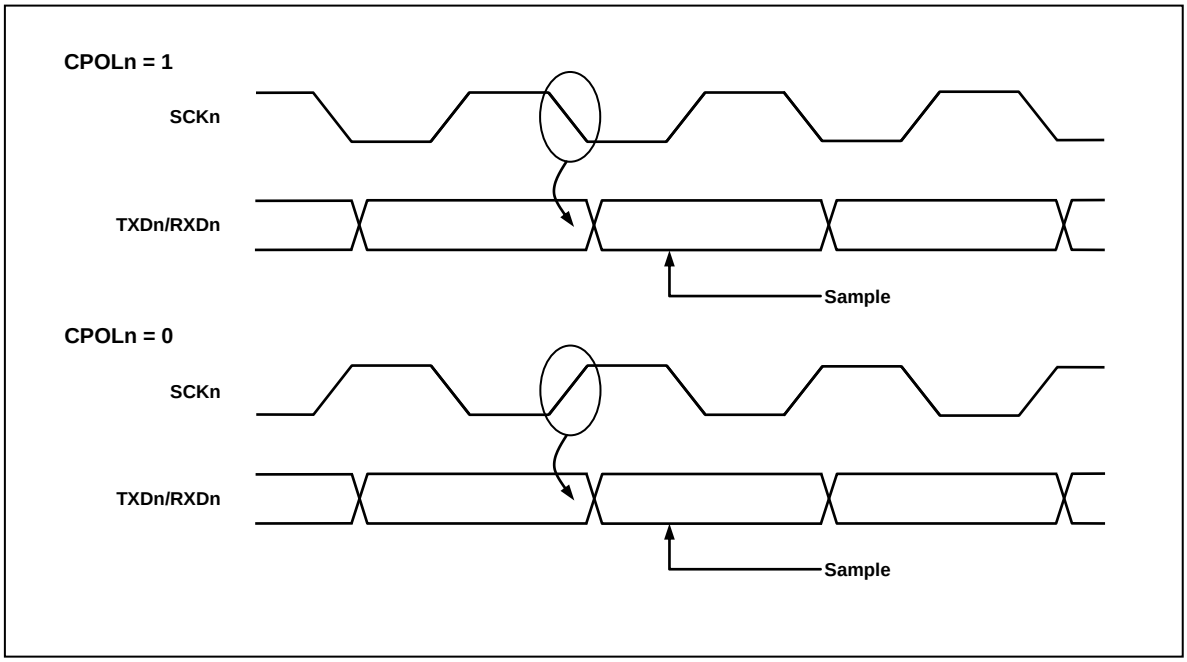



**Figure 11.61 Synchronous Mode SCKn Timing (USIn , where n = 0 and 1)**

### 11.12.7 USI0/1 UART Data format

A serial frame is defined to be one character of data bits with synchronization bits (start and stop bits), and optionally a parity bit for error detection.

The UART supports all 30 combinations of the following as valid frame formats.

- 1 start bit
- 5, 6, 7, 8 or 9 data bits
- no, even or odd parity bit
- 1 or 2 stop bits

A frame starts with the start bit followed by the least significant data bit (LSB). Then the next data bits, up to nine, are succeeding, ending with the most significant bit (MSB). If parity function is enabled, the parity bit is inserted between the last data bit and the stop bit. A high-to-low transition on data pin is considered as start bit. When a complete frame is transmitted, it can be directly followed by a new frame, or the communication line can be set to an idle state. The idle means high state of data pin. The following figure shows the possible combinations of the frame formats. Bits inside brackets are optional.

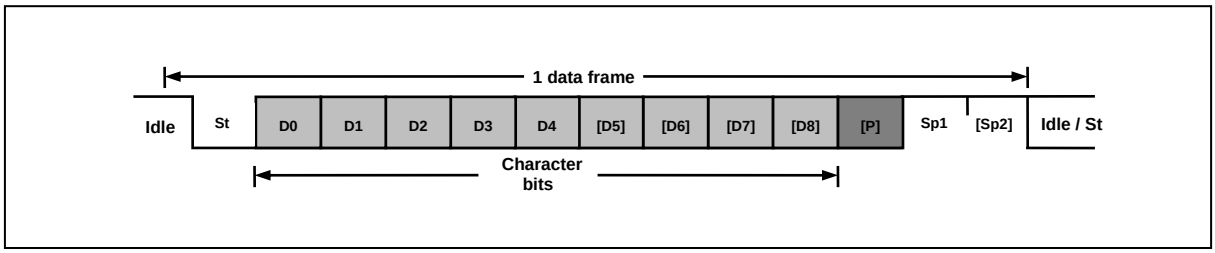



**Figure 11.62 Frame Format (USI0/1)**

1 data frame consists of the following bits

- Idle No communication on communication line (TXDn/RXDn)
- St Start bit (Low)
- Dn Data bits (0~8)
- Parity bit ---------- Even parity, Odd parity, No parity
- Stop bit(s) ---------- 1 bit or 2 bits

The frame format used by the UART is set by the USInS[2:0], USInPM[1:0] bits in USInCR1 register and USInSB bit in USInCR3 register. The Transmitter and Receiver use the same setting.

### 11.12.8 USI0/1 UART Parity bit

The parity bit is calculated by doing an exclusive-OR of all the data bits. If odd parity is used, the result of the exclusive-O is inverted. The parity bit is located between the MSB and first stop bit of a serial frame.

$P_{even} = D_{n-1} \wedge \ldots \wedge D_3 \wedge D_2 \wedge D_1 \wedge D_0 \wedge 0$

$P_{odd} = D_{n-1} \wedge \ldots \wedge D_3 \wedge D_2 \wedge D_1 \wedge D_0 \wedge 1$

$P_{even}$ : Parity bit using even parity

$P_{odd}$ : Parity bit using odd parity

$D_n$ : Data bit n of the character

### 11.12.9 USI0/1 UART Transmitter

The UART transmitter is enabled by setting the TXEn bit in USInCR2 register. When the Transmitter is enabled, the TXDn pin should be set to TXDn function for the serial output pin of UART by the P3FSRL[3:2]/P5FSRH[4:3]. The baud-rate, operation mode and frame format must be setup once before doing any transmission. In synchronous operation mode, the SCKn pin is used as transmission clock, so it should be selected to do SCKn function by P3FSRL[1:0]/P5FSRH[2:1] .

#### 11.12.9.1 USI0/1 UART Sending Tx data

A data transmission is initiated by loading the transmit buffer (USInDR register I/O location) with the data to be transmitted. The data written in transmit buffer is moved to the shift register when the shift register is ready to send a new frame. The shift register is loaded with the new data if it is in idle state or immediately after the last stop bit of the previous frame is transmitted. When the shift register is loaded with new data, it will transfer one complete frame according to the settings of control registers. If the 9-bit characters are used in asynchronous or synchronous operation mode, the ninth bit must be written to the USInTX8 bit in USInCR3 register before it is loaded to the transmit buffer (USInDR register).

#### 11.12.9.2 USI0/1 UART Transmitter flag and interrupt

The UART transmitter has 2 flags which indicate its state. One is UART data register empty flag (DREn) and the other is transmit complete flag (TXCn). Both flags can be interrupt sources.

DREn flag indicates whether the transmit buffer is ready to receive new data. This bit is set when the transmit buffer is empty and cleared when the transmit buffer contains data to be transmitted but has not yet been moved into the shift register. And also this flag can be cleared by writing ‘0’ to this bit position. Writing ‘1’ to this bit position is prevented.

When the data register empty interrupt enable (DRIEn) bit in USInCR2 register is set and the global interrupt is enabled, USInST1 status register empty interrupt is generated while DREn flag is set.

The transmit complete (TXCn) flag bit is set when the entire frame in the transmit shift register has been shifted out and there is no more data in the transmit buffer. The TXCn flag is automatically cleared when the transmit complete interrupt service routine is executed, or it can be cleared by writing ‘0’ to TXCn bit in USInST1 register.

When the transmit complete interrupt enable (TXCIEn) bit in USInCR2 register is set and the global interrupt is enabled, UART transmit complete interrupt is generated while TXCn flag is set.

### 11.12.9.3 USI0/1 UART Parity Generator

The parity generator calculates the parity bit for the serial frame data to be sent. When parity bit is enabled (USInPM1=1), the transmitter control logic inserts the parity bit between the MSB and the first stop bit of the frame to be sent.

### 11.12.9.4 USI0/1 UART Disabling Transmitter

Disabling the transmitter by clearing the TXEn bit will not become effective until ongoing transmission is completed. When the Transmitter is disabled, the TXDn pin can be used as a normal general purpose I/O (GPIO).

### 11.12.10 USI0/1 UART Receiver

The UART receiver is enabled by setting the RXEn bit in the USInCR2 register. When the receiver is enabled, the RXDn pin should be set to RXDn function for the serial input pin of UART by P3FSRL[5:4]/P5FSRH[6:5]. The baud-rate, mode of operation and frame format must be set before serial reception. In synchronous or SPI operation mode the SCKn pin is used as transfer clock, so it should be selected to do SCKn function by P3FSRL[1:0]/P5FSRH[2:1]. In SPI operation mode the SSn input pin in slave mode or can be configured as SSn output pin in master mode. This can be done by setting USInSSEN bit in USnCR3 register.

### 11.12.10.1 USI0/1 UART Receiving Rx data

When UART is in synchronous or asynchronous operation mode, the receiver starts data reception when it detects a valid start bit (LOW) on RXDn pin. Each bit after start bit is sampled at pre-defined baud-rate (asynchronous) or sampling edge of SCKn (synchronous), and shifted into the receive shift register until the first stop bit of a frame is received. Even if there's $2^{nd}$ stop bit in the frame, the $2^{nd}$ stop bit is ignored by the receiver. That is, receiving the first stop bit means that a complete serial frame is present in the receiver shift register and contents of the shift register are to be moved into the receive buffer. The receive buffer is read by reading the USInDR register.

If 9-bit characters are used (USInS[2:0] = "111"), the ninth bit is stored in the USInRX8 bit position in the USInCR3 register. The $9^{th}$ bit must be read from the USInRX8 bit before reading the low 8 bits from the USInDR register. Likewise, the error flags FEn, DORn, PEn must be read before reading the data from USInDR register. It's because the error flags are stored in the same FIFO position of the receive buffer.

### 11.12.10.2 USI0/1 UART Receiver Flag and Interrupt

The UART receiver has one flag that indicates the receiver state.

The receive complete (RXCn) flag indicates whether there are unread data in the receive buffer. This flag is set when there are unread data in the receive buffer and cleared when the receive buffer is empty. If the receiver is disabled (RXEn=0), the receiver buffer is flushed and the RXCn flag is cleared.

When the receive complete interrupt enable (RXCIEn) bit in the USInCR2 register is set and global interrupt is enabled, the UART receiver complete interrupt is generated while RXCn flag is set.

The UART receiver has three error flags which are frame error (FEn), data overrun (DORn) and parity error (PEn). These error flags can be read from the USInST1 register. As received data are stored in the 2-level receive buffer, these error flags are also stored in the same position of receive buffer. So, before reading received data from USInDR register, read the USInST1 register first which contains error flags.

The frame error (FEn) flag indicates the state of the first stop bit. The FEn flag is '0' when the stop bit was correctly detected as "1", and the FEn flag is "1" when the stop bit was incorrect, i.e. detected as "0". This flag can be used for detecting out-of-sync conditions between data frames.

The data overrun (DORn) flag indicates data loss due to a receive buffer full condition. DORn occurs when the receive buffer is full, and another new data is present in the receive shift register which are to be stored into the receive buffer. After the DORn flag is set, all the incoming data are lost. To prevent data loss or clear this flag, read the receive buffer.

The parity error (PEn) flag indicates that the frame in the receive buffer had a parity error when received. If parity check function is not enabled (USInPM1=0), the PEn bit is always read "0".

### 11.12.10.3 USI0/1 UART Parity Checker

If parity bit is enabled (USInPM1=1), the Parity Checker calculates the parity of the data bits in incoming frame and compares the result with the parity bit from the received serial frame.

### 11.12.10.4 USI0/1 UART Disabling Receiver

In contrast to transmitter, disabling the Receiver by clearing RXEn bit makes the Receiver inactive immediately. When the receiver is disabled, the receiver flushes the receive buffer, the remaining data in the buffer is all reset, and the RXDn pin can be used as a normal general purpose I/O (GPIO).

### 11.12.10.5 USI0/1 Asynchronous Data Reception

To receive asynchronous data frame, the UART includes a clock and data recovery unit. The clock recovery logic is used for synchronizing the internally generated baud-rate clock to the incoming asynchronous serial frame on the RXDn pin.

The data recovery logic samples and low pass filters the incoming bits, and this removes the noise of RXDn pin.

The next figure illustrates the sampling process of the start bit of an incoming frame. The sampling rate is 16 times of the baud-rate in normal mode and 8 times the baud-rate for double speed mode (DBLSn=1). The horizontal arrows show the synchronization variation due to the asynchronous sampling process. Note that larger time variation is shown when using the double speed mode.

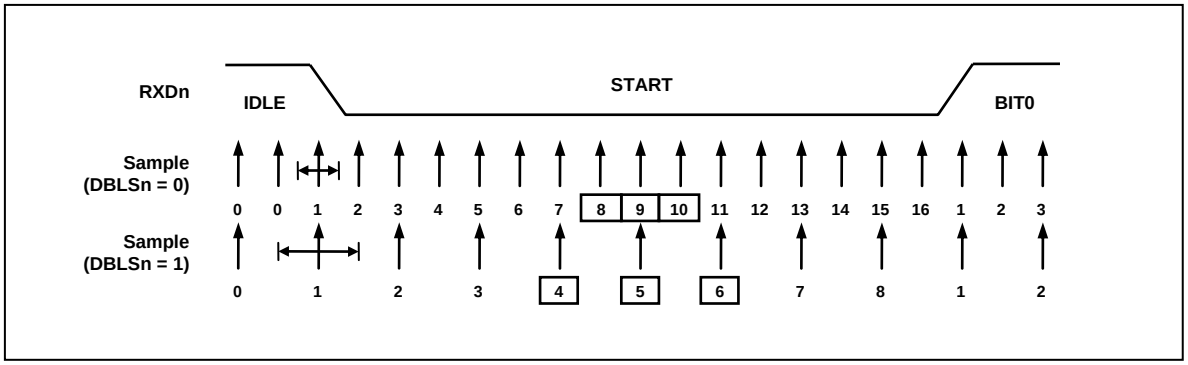



**Figure 11.63 Asynchronous Start Bit Sampling (USIn, where n = 0 and 1)**

When the receiver is enabled (RXEn=1), the clock recovery logic tries to find a high-to-low transition on the RXDn line, the start bit condition. After detecting high to low transition on RXDn line, the clock recovery logic uses samples 8, 9 and 10 for normal mode to decide if a valid start bit is received. If more than 2 samples have logical low level, it is considered that a valid start bit is detected and the internally generated clock is synchronized to the incoming data frame. And the data recovery can begin. The synchronization process is repeated for each start bit.

As described above, when the receiver clock is synchronized to the start bit, the data recovery can begin. Data recovery process is almost similar to the clock recovery process. The data recovery logic samples 16 times for each incoming bits for normal mode and 8 times for double speed mode, and uses sample 8, 9 and 10 to decide data value. If more than 2 samples have low levels, the received bit is considered to a logic ‘0’ and if more than 2 samples have high levels, the received bit is considered to a logic ‘1’. The data recovery process is then repeated until a complete frame is received including the first stop bit. The decided bit value is stored in the receive shift register in order. Note that the Receiver only uses the first stop bit of a frame. Internally, after receiving the first stop bit, the Receiver is in idle state and waiting to find start bit.

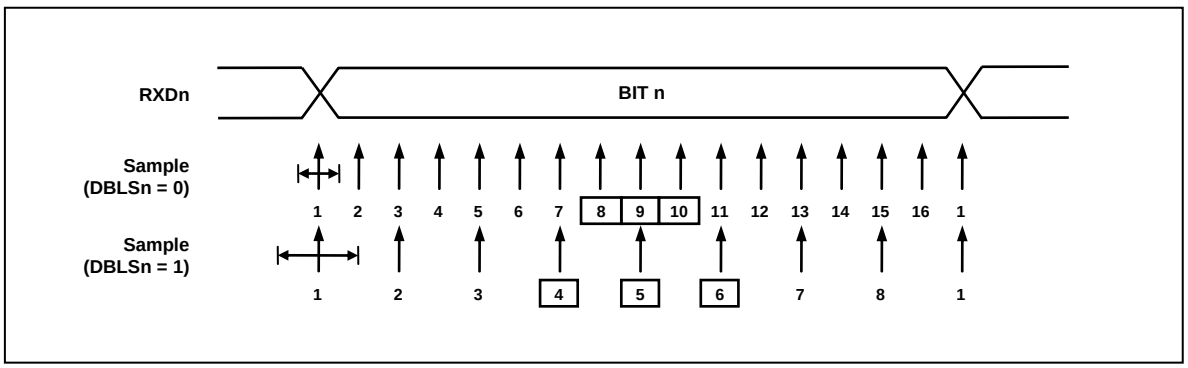



**Figure 11.64 Asynchronous Sampling of Data and Parity Bit (USIn, where n = 0 and 1)**

The process for detecting stop bit is like clock and data recovery process. That is, if 2 or more samples of 3 center values have high level, correct stop bit is detected, else a frame error (FEn) flag is set. After deciding whether the first stop bit is valid or not, the Receiver goes to idle state and monitors the RXDn line to check a valid high to low transition is detected (start bit detection).

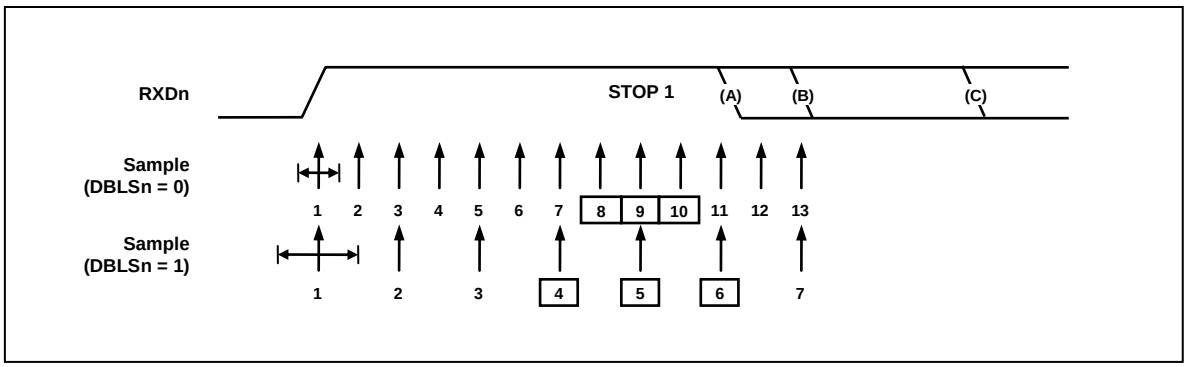



**Figure 11.65 Stop Bit Sampling and Next Start Bit Sampling (USIn, where n = 0 and 1)**

### 11.12.11 USI0/1 SPI Mode

The USI0/1 can be set to operate in industrial standard SPI compliant mode. The SPI mode has the following features.

- Full Duplex, Three-wire synchronous data transfer
- Mater and Slave Operation
- Supports all four SPI0 modes of operation (mode 0, 1, 2, and 3)
- Selectable LSB first or MSB first data transfer
- Double buffered transmit and receive
- Programmable transmit bit rate

When SPI mode is enabled (USInMS[1:0]="11"), the slave select (SSn) pin becomes active LOW input in slave mode operation, or can be output in master mode operation if USInSSEN bit is set to '0'.

Note that during SPI mode of operation, the pin RXDn is renamed as MISOn and TXDn is renamed as MOSIn for compatibility to other SPI devices.

### 11.12.12 USI0/1 SPI Clock Formats and Timing

To accommodate a wide variety if synchronus serial peripherals from different manufacturers, the USI0/1 has a clock polarity bit (CPOLn) and a clock phase control bit (CPHAn) to select one of four clock formats for data transfers. CPOLn selectively insert an inverter in series with the clock. CPHAn chooses between two different clock phase relationships between the clock and data. Note that CPHAn and CPOLn bits in USInCR1 register have different meanings according to the USInMS[1:0] bits which decides the operating mode of USI0/1.

Table below shows four combinations of CPOLn and CPHAn for SPI mode 0, 1, 2, and 3.

**Table 11-22 CPOLn Functionality (where n = 0 and 1)**

| SPI Mode | CPOLn | CPHAn | Leading Edge | Trailing Edge |
|---|---|---|---|---|
| 0 | 0 | 0 | Sample (Rising) | Setup (Falling) |
| 1 | 0 | 1 | Setup (Rising) | Sample (Falling) |
| 2 | 1 | 0 | Sample (Falling) | Setup (Rising) |
| 3 | 1 | 1 | Setup (Falling) | Sample (Rising) |

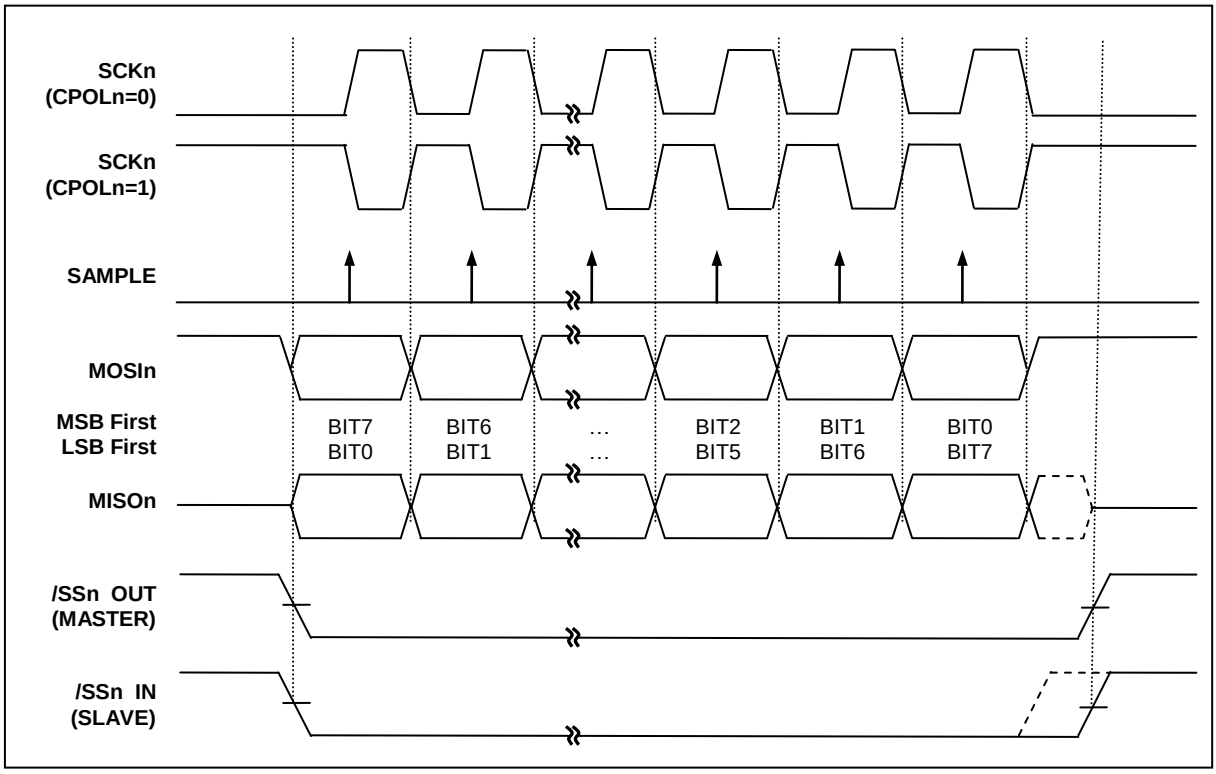



**Figure 11.66 USI0/1 SPI Clock Formats when CPHAn=0 (where n = 0 and 1)**

When CPHAn=0, the slave begins to drive its MISOn output with the first data bit value when SSn goes to active low. The first SCKn edge causes both the master and the slave to sample the data bit value on their MISOn and MOSIn inputs, respectively. At the second SCKn edge, the USI0/1 shifts the second data bit value out to the MOSIn and MISOn outputs of the master and slave, respectively. Unlike the case of CPHAn=1, when CPHAn=0, the slave's SSn input must go to its inactive high level between transfers. This is because the slave can prepare the first data bit when it detects falling edge of SSn input.

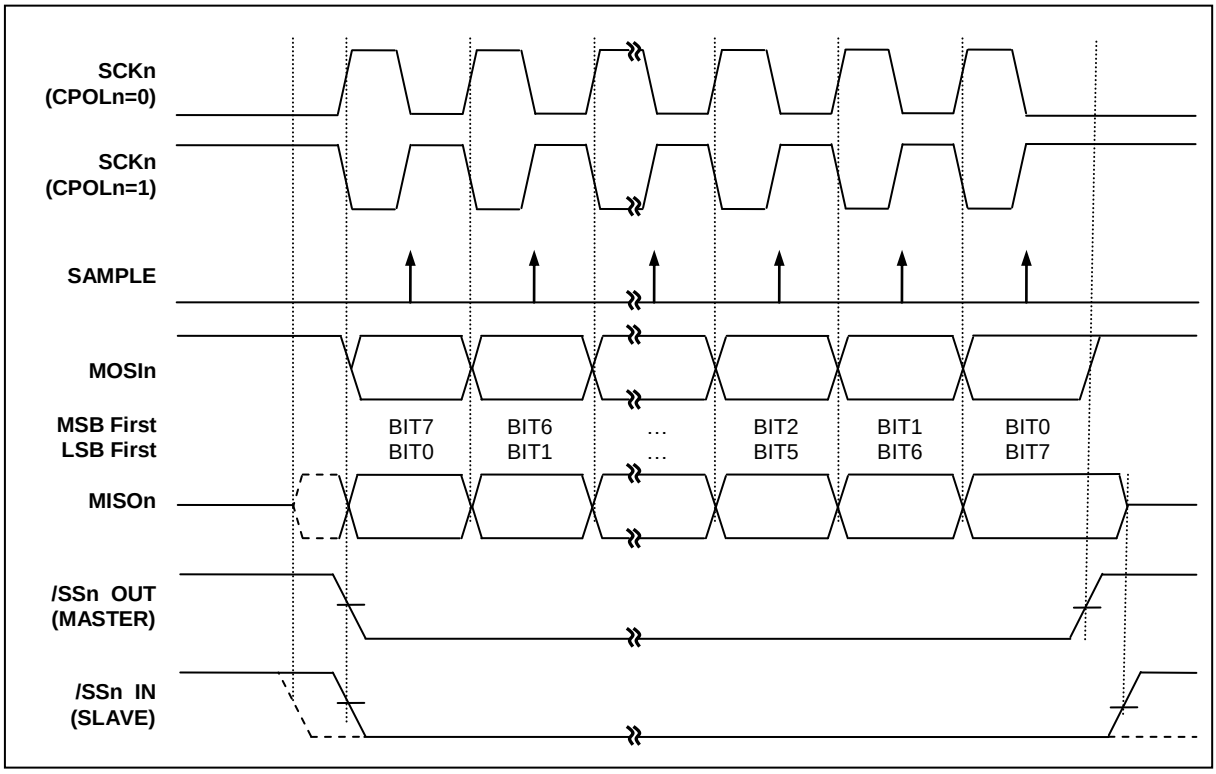



**Figure 11.67 USI0/1 SPI Clock Formats when CPHAn=1 (where n = 0 and 1)**

When CPHAn=1, the slave begins to drive its MISOn output when SSn goes active low, but the data is not defined until the first SCKn edge. The first SCKn edge shifts the first bit of data from the shifter onto the MOSIn output of the master and the MISOn output of the slave. The next SCKn edge causes both the master and slave to sample the data bit value on their MISOn and MOSIn inputs, respectively. At the third SCKn edge, the USI0/1 shifts the second data bit value out to the MOSIn and MISOn output of the master and slave respectively. When CPHAn=1, the slave's SSn input is not required to go to its inactive high level between transfers.

Because the SPI logic reuses the USI0/1 resources, SPI mode of operation is similar to that of synchronous or asynchronous operation. An SPI transfer is initiated by checking for the USI0/1 Data Register Empty flag (DREn=1) and then writing a byte of data to the USInDR Register. In master mode of operation, even if transmission is not enabled (TXEn=0), writing data to the USInDR register is necessary because the clock SCKn is generated from transmitter block.

### 11.12.13 USI0/1 SPI Block Diagram

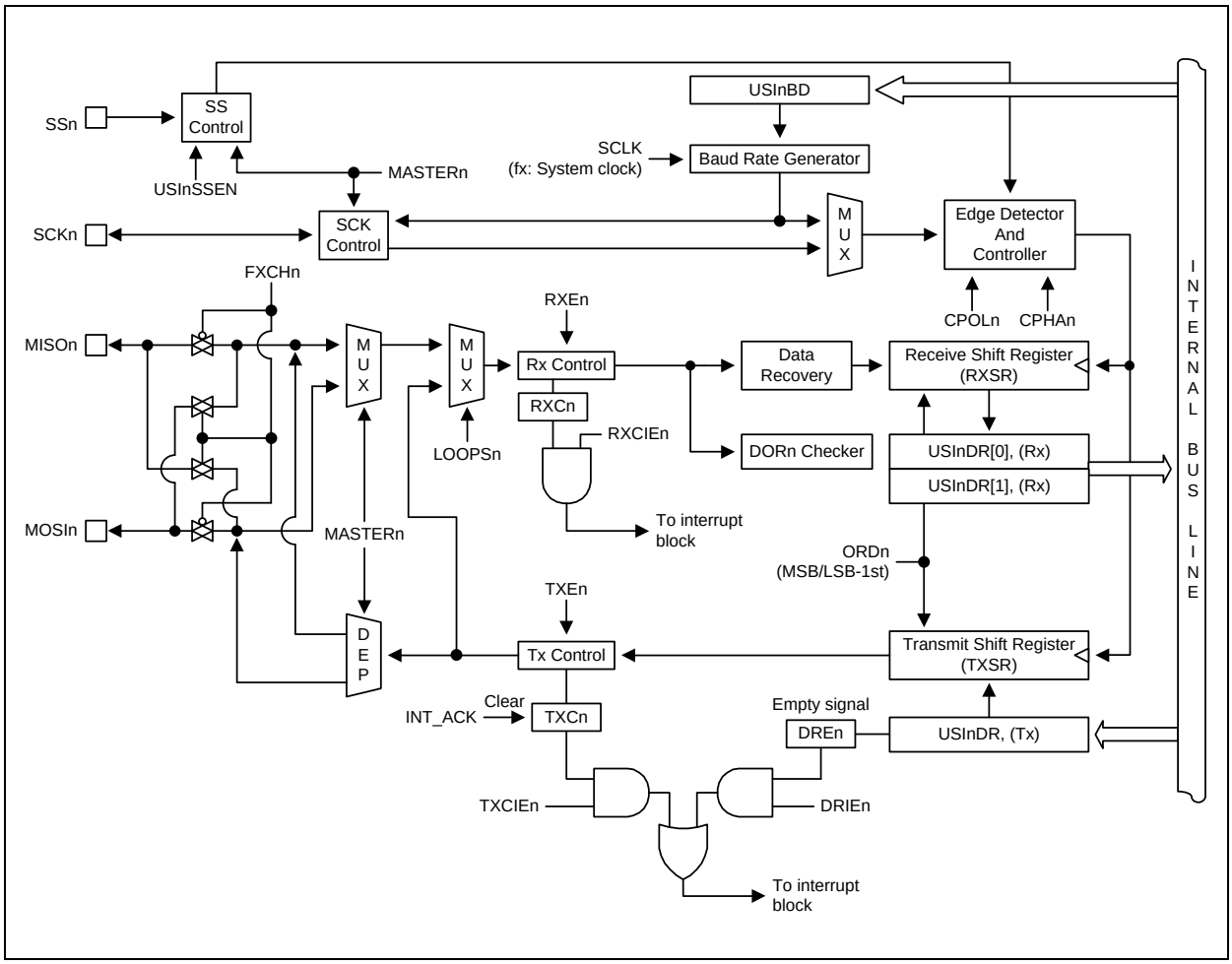



**Figure 11.68 USI0/1 SPI Block Diagram (where n = 0 and 1)**

### 11.12.14 USI0/1 I2C Mode

The USI0/1 can be set to operate in industrial standard serial communicatin protocols mode. The I2C mode uses 2 bus lines serial data line (SDAn) and serial clock line (SCLn) to exchange data. Because both SDAn and SCLn lines are open-drain output, each line needs pull-up resistor. The features are as shown below.

- Compatible with I2C bus standard
- Multi-master operation
- Up to 400kHz data transfer read speed
- 7 bit address
- Both master and slave operation
- Bus busy detection

### 11.12.15 USI0/1 I2C Bit Transfer

The data on the SDAn line must be stable during HIGH period of the clock, SCLn. The HIGH or LOW state of the data line can only change when the clock signal on the SCLn line is LOW. The exceptions are START(S), repeated START(Sr) and STOP(P) condition where data line changes when clock line is high.

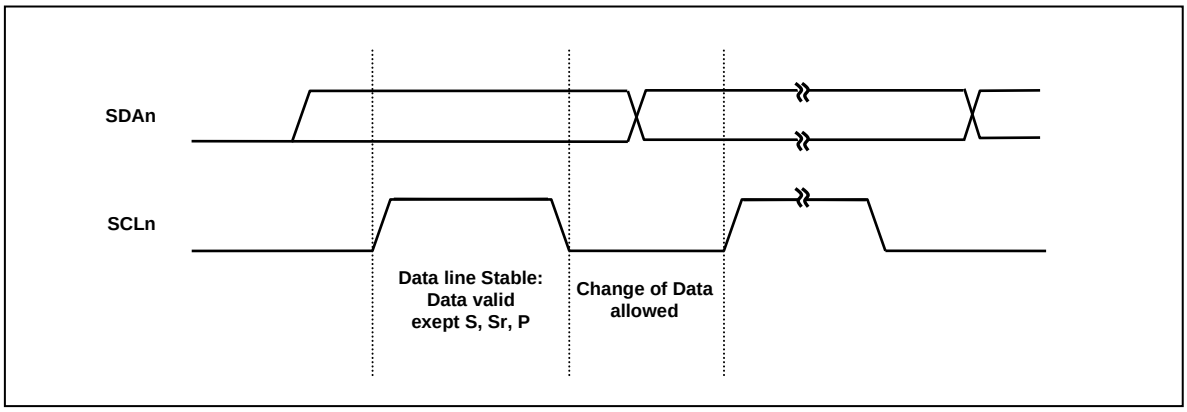



**Figure 11.69 Bit Transfer on the I2C-Bus (USIn, where n = 0 and 1)**

## 11.12.16 USI0/1 I2C Start / Repeated Start / Stop

One master can issue a START (S) condition to notice other devices connected to the SCLn, SDAn lines that it will use the bus. A STOP (P) condition is generated by the master to release the bus lines so that other devices can use it.

A high to low transition on the SDAn line while SCLn is high defines a START (S) condition.
A low to high transition on the SDAn line while SCLn is high defines a STOP (P) condition.

START and STOP conditions are always generated by the master. The bus is considered to be busy after START condition. The bus is considered to be free again after STOP condition, ie, the bus is busy between START and STOP condition. If a repeated START condition (Sr) is generated instead of STOP condition, the bus stays busy. So, the START and repeated START conditions are functionally identical.

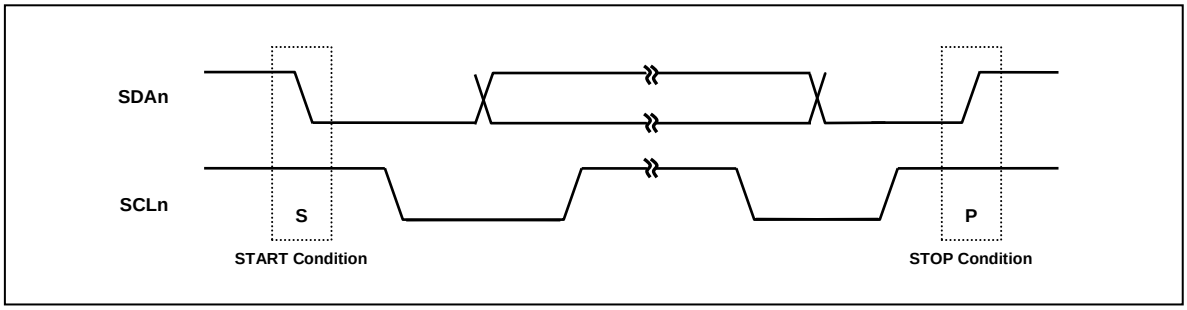



**Figure 11.70 START and STOP Condition (USIn, where n = 0 and 1)**

## 11.12.17 USI0/1 I2C Data Transfer

Every byte put on the SDAn line must be 8-bits long. The number of bytes that can be transmitted per transfer is unlimited. Each byte has to be followed by an acknowledge bit. Data is transferred with the most significant bit (MSB) first. If a slave can't receive or transmit another complete byte of data until it has performed some other function, it can hold the clock line SCLn LOW to force the master into a wait state. Data transfer then continues when the slave is ready for another byte of data and releases clock line SCLn.

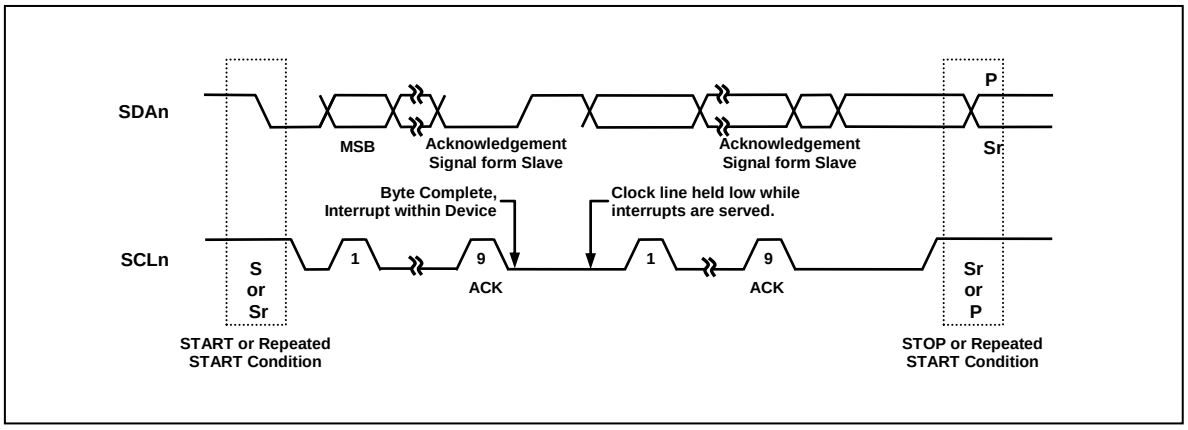



**Figure 11.71 Data Transfer on the I2C-Bus (USIn, where n = 0 and 1)**

### 11.12.18 USI0/1 I2C Acknowledge

The acknowledge related clock pulse is generated by the master. The transmitter releases the SDAn line (HIGH) during the acknowledge clock pulse. The receiver must pull down the SDAn line during the acknowledge clock pulse so that it remains stable LOW during the HIGH period of this clock pulse. When a slave is addressed by a master (Address Packet), and if it is unable to receive or transmit because it's performing some real time function, the data line must be left HIGH by the slave. And also, when a slave addressed by a master is unable to receive more data bits, the slave receiver must release the SDAn line (Data Packet). The master can then generate either a STOP condition to abort the transfer, or a repeated START condition to start a new transfer.

If a master receiver is involved in a transfer, it must signal the end of data to the slave transmitter by not generating an acknowledge on the last byte that was clocked out of the slave. The slave transmitter must release the data line to allow the master to generate a STOP or repeated START condition.

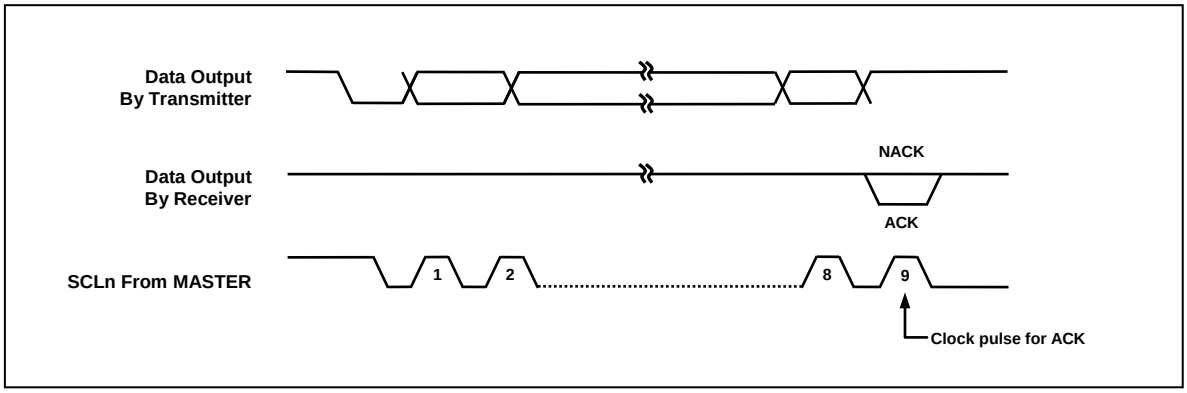



**Figure 11.72 Acknowledge on the I2C-Bus (USIn, where n = 0 and 1)**

### 11.12.19 USI0/1 I2C Synchronization / Arbitration

Clock synchronization is performed using the wired-AND connection of I2C interfaces to the SCLn line. This means that a HIGH to LOW transition on the SCLn line will cause the devices concerned to start counting off their LOW period and it will hold the SCLn line in that state until the clock HIGH state is reached. However the LOW to HIGH transition of this clock may not change the state of the SCLn line if another clock is still within its LOW period. In this way, a synchronized SCLn clock is generated with its LOW period determined by the device with the longest clock LOW period, and its HIGH period determined by the one with the shortest clock HIGH period.

A master may start a transfer only if the bus is free. Two or more masters may generate a START condition. Arbitration takes place on the SDAn line, while the SCLn line is at the HIGH level, in such a way that the master which transmits a HIGH level, while another master is transmitting a LOW level will switch off its DATA output state because the level on the bus doesn't correspond to its own level. Arbitration continues for many bits until a winning master gets the ownership of I2C bus. Its first stage is comparison of the address bits.

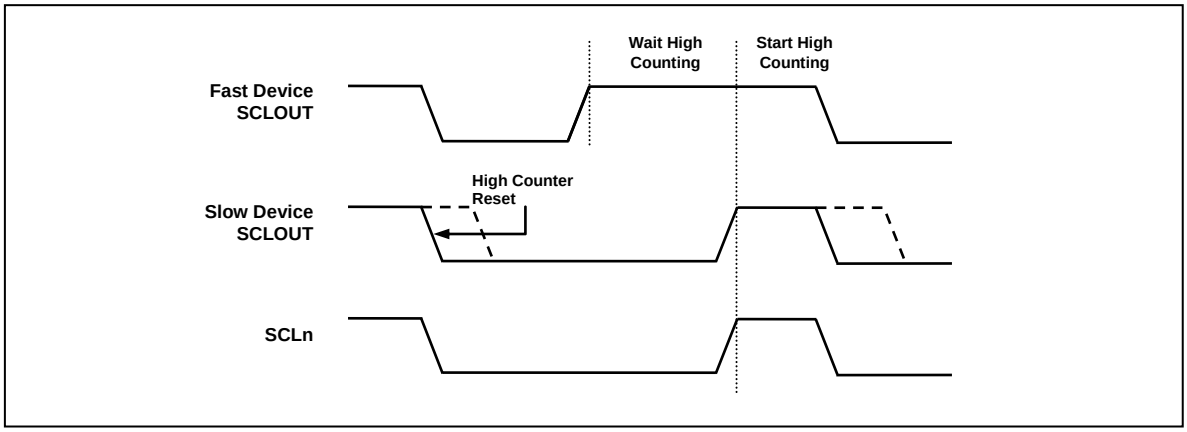


**Figure 11.73 Clock Synchronization during Arbitration Procedure (USIn, where n = 0 and 1)**

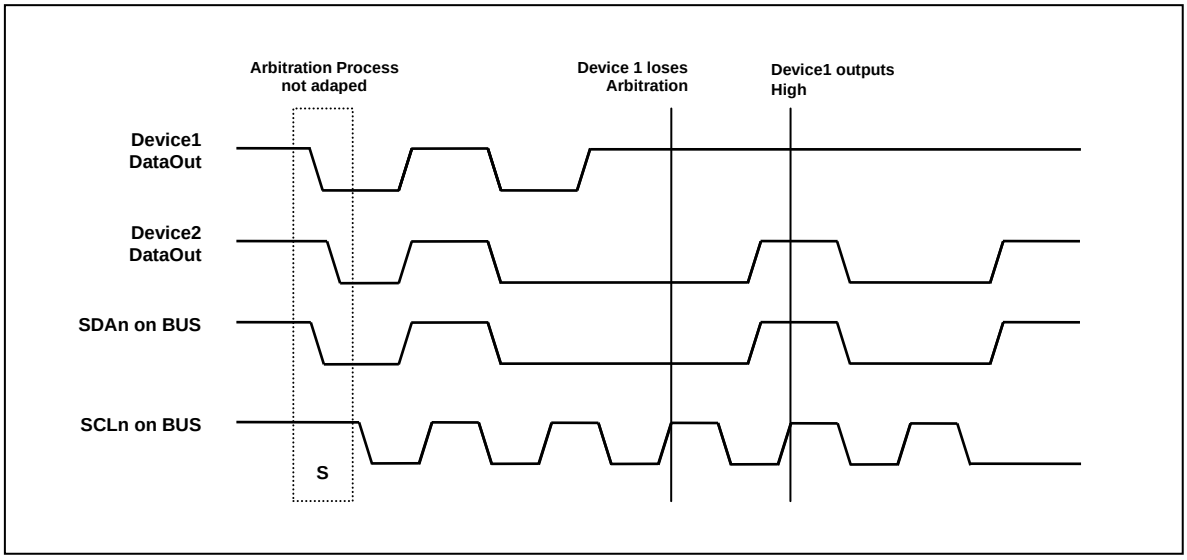



**Figure 11.74 Arbitration Procedure of Two Masters (USIn, where n = 0 and 1)**

## 11.12.20 USI0/1 I2C Operation

The I2C is byte-oriented and interrupt based. Interrupts are issued after all bus events except for a transmission of a START condition. Because the I2C is interrupt based, the application software is free to carry on other operations during a I2C byte transfer.

Note that when a I2C interrupt is generated, IICnIFR flag in USInCR4 register is set, it is cleared by writing an any value to USInST2. When I2C interrupt occurs, the SCLn line is hold LOW until writing any value to USInST2. When the IICnIFR flag is set, the USInST2 contains a value indicating the current state of the I2C bus. According to the value in USInST2, software can decide what to do next.

I2C can operate in 4 modes by configuring master/slave, transmitter/receiver. The operating mode is configured by a winning master. A more detailed explanation follows below.

### 11.12.20.1 USI0/1 I2C Master Transmitter

To operate I2C in master transmitter, follow the recommended steps below.

1. Enable I2C by setting USInMS[1:0] bits in USInCR1 and USInEN bit in USInCR2. This provides main clock to the peripheral.
2. Load SLAn+W into the USInDR where SLAn is address of slave device and W is transfer direction from the viewpoint of the master. For master transmitter, W is '0'. Note that USInDR is used for both address and data.
3. Configure baud rate by writing desired value to both USInSCLR and USInSCHR for the Low and High period of SCLn line.
4. Configure the USInSDHR to decide when SDAn changes value from falling edge of SCLn. If SDAn should change in the middle of SCLn LOW period, load half the value of USInSCLR to the USInSDHR.
5. Set the STARTCn bit in USInCR4. This transmits a START condition. And also configure how to handle interrupt and ACK signal. When the STARTCn bit is set, 8-bit data in USInDR is transmitted out according to the baud-rate.
6. This is ACK signal processing stage for address packet transmitted by master. When 7-bit address and 1-bit transfer direction is transmitted to target slave device, the master can know whether the slave acknowledged or not in the 9$^{th}$ high period of SCLn. If the master gains bus mastership, I2C generates GCALLn interrupt regardless of the reception of ACK from the slave device. When I2C loses bus mastership during arbitration process, the MLOSTn bit in USInST2 is set, and I2C waits in idle state or can be operate as an addressed slave. To operate as a slave when the MLOSTn bit in USInST2 is set, the ACKnEN bit in USInCR4 must be set and the received 7-bit address must equal to the USInSLA[6:0] bits in USInSAR. In this case I2C operates as a slave transmitter or a slave receiver (go to appropriate section). In this stage, I2C holds the SCLn LOW. This is because to decide whether I2C continues serial transfer or STOP communication. The following steps continue assuming that I2C does not lose mastership during first data transfer.

   I2C (Master) can choose one of the following cases regardless of the reception of ACK signal from slave.

   1) Master receives ACK signal from slave, so continues data transfer because slave can receive more data from master. In this case, load data to transmit to USInDR.
   2) Master STOP data transfer even if it receives ACK signal from slave. In this case, set the STOPCn bit in USInCR4.
   3) Master transmits repeated START condition with not checking ACK signal. In this case, load SLAn+R/W into the USInDR and set STARTCn bit in USInCR4.

   After doing one of the actions above, write any arbitrary to USInST2 to release SCLn line. In case of 1), move to step 7. In case of 2), move to step 9 to handle STOP interrupt. In case of 3), move to step 6 after transmitting the data in USInDR and if transfer direction bit is '1' go to master receiver section.
7. 1-Byte of data is being transmitted. During data transfer, bus arbitration continues.
8. This is ACK signal processing stage for data packet transmitted by master. I2C holds the SCLn LOW. When I2C loses bus mastership while transmitting data arbitrating other masters, the MLOSTn bit in USInST2 is set. If then, I2C waits in idle state. When the data in USInDR is transmitted completely, I2C generates TENDn interrupt.

   I2C can choose one of the following cases regardless of the reception of ACK signal from slave.

   1) Master receives ACK signal from slave, so continues data transfer because slave can receive more data from master. In this case, load data to transmit to USInDR.
   2) Master STOP data transfer even if it receives ACK signal from slave. In this case, set the STOPCn bit in USInCR4.
   3) Master transmits repeated START condition with not checking ACK signal. In this case, load SLAn+R/W into the USInDR and set the STARTCn bit in USInCR4.

   After doing one of the actions above, write any arbitrary to USInST2 to release SCLn line. In case of 1), move to step 7. In case of 2), move to step 9 to handle STOP interrupt. In case of 3), move to step 6 after transmitting the data in USInDR, and if transfer direction bit is '1' go to master receiver section.
9. This is the final step for master transmitter function of I2C, handling STOP interrupt. The STOP bit indicates that data transfer between master and slave is over. To clear USInST2, write any value to USInST2. After this, I2C enters idle state.

The next figure depicts above process for master transmitter operation of I2C.

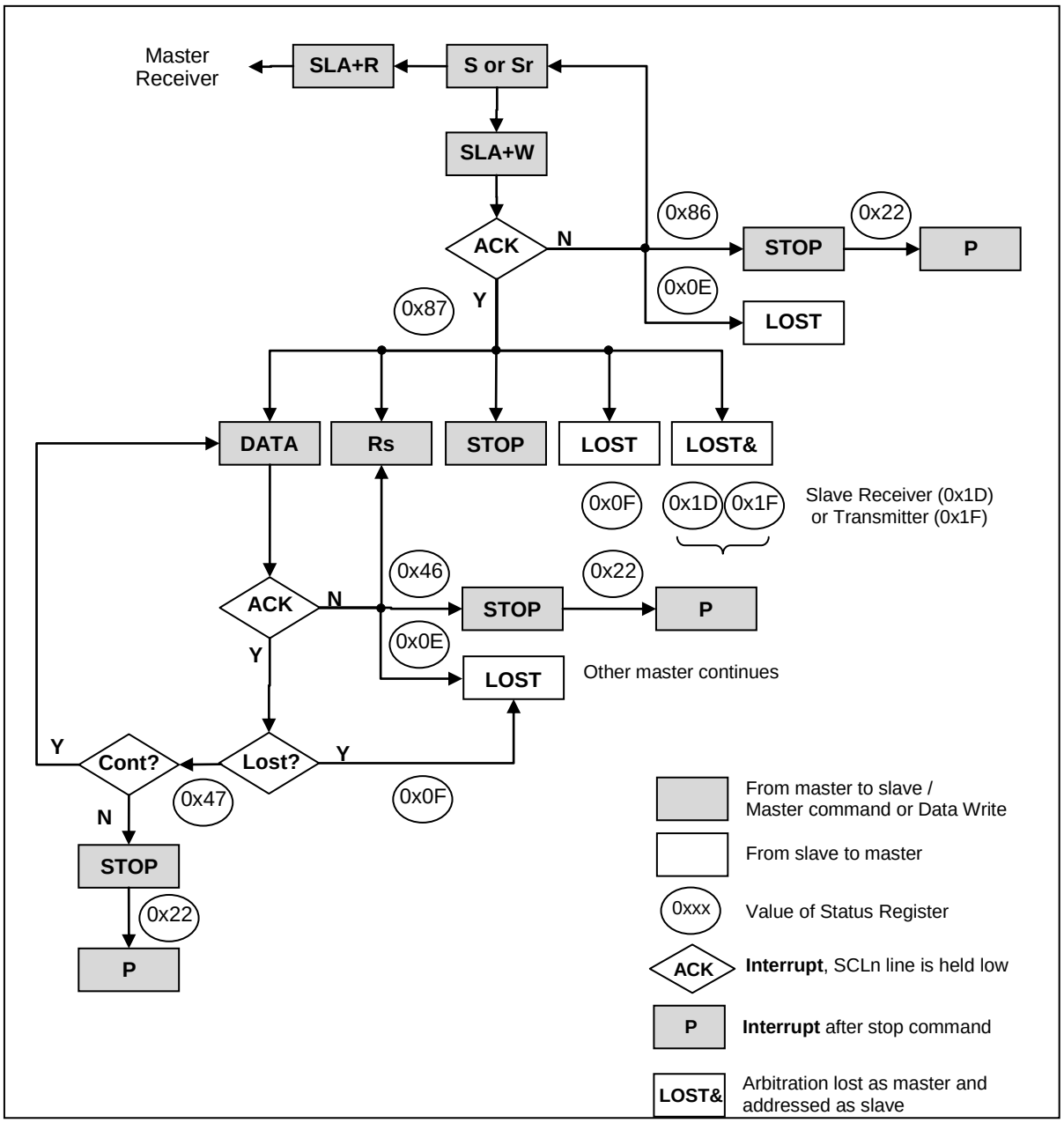



**Figure 11.75 Formats and States in the Master Transmitter Mode (USIn, where n = 0 and 1)**

### 11.12.20.2 USI0/1 I2C Master Receiver

To operate I2C in master receiver, follow the recommended steps below.

1. Enable I2C by setting USInMS[1:0] bits in USInCR1 and USInEN bit in USInCR2. This provides main clock to the peripheral.
2. Load SLAn+R into the USInDR where SLA is address of slave device and R is transfer direction from the viewpoint of the master. For master receiver, R is '1'. Note that USInDR is used for both address and data.
3. Configure baud rate by writing desired value to both USInSCLR and USInSCHR for the Low and High period of SCLn line.
4. Configure the USInSDHR to decide when SDAn changes value from falling edge of SCLn. If SDAn should change in the middle of SCLn LOW period, load half the value of USInSCLR to the USInSDHR.
5. Set the STARTCn bit in USInCR4. This transmits a START condition. And also configure how to handle interrupt and ACK signal. When the STARTCn bit is set, 8-bit data in USInDR is transmitted out according to the baud-rate.
6. This is ACK signal processing stage for address packet transmitted by master. When 7-bit address and 1-bit transfer direction is transmitted to target slave device, the master can know whether the slave acknowledged or not in the 9$^{th}$ high period of SCLn. If the master gains bus mastership, I2C generates GCALL interrupt regardless of the reception of ACK from the slave device. When I2C loses bus mastership during arbitration process, the MLOSTn bit in USInST2 is set, and I2C waits in idle state or can be operate as an addressed slave. To operate as a slave when the MLOSTn bit in USInST2 is set, the ACKnEN bit in USInCR4 must be set and the received 7-bit address must equal to the USInSLA[6:0] bits in USInSAR. In this case I2C operates as a slave transmitter or a slave receiver (go to appropriate section). In this stage, I2C holds the SCLn LOW. This is because to decide whether I2C continues serial transfer or STOP communication. The following steps continue assuming that I2C does not lose mastership during first data transfer.

   I2C (Master) can choose one of the following cases according to the reception of ACK signal from slave.

   1) Master receives ACK signal from slave, so continues data transfer because slave can prepare and transmit more data to master. Configure ACKnEN bit in USInCR4 to decide whether I2C ACKnowledges the next data to be received or not.
   2) Master STOP data transfer because it receives no ACK signal from slave. In this case, set the STOPCn bit in USInCR4.
   3) Master transmits repeated START condition due to no ACK signal from slave. In this case, load SLAn+R/W into the USInDR and set STARTCn bit in USInCR4.

   After doing one of the actions above, write arbitrary value to USInST2 to release SCLn line. In case of 1), move to step 7. In case of 2), move to step 9 to handle STOP interrupt. In case of 3), move to step 6 after transmitting the data in USInDR and if transfer direction bit is '0' go to master transmitter section.

7. 1-Byte of data is being received.
8. This is ACK signal processing stage for data packet transmitted by slave. I2C holds the SCLn LOW. When 1-Byte of data is received completely, I2C generates TENDn interrupt.

   I2C can choose one of the following cases according to the RXACKn flag in USInST2.

   1) Master continues receiving data from slave. To do this, set ACKnEN bit in USInCR4 to ACKnowledge the next data to be received.
   2) Master wants to terminate data transfer when it receives next data by not generating ACK signal. This can be done by clearing ACKnEN bit in USInCR4.
   3) Because no ACK signal is detected, master terminates data transfer. In this case, set the STOPCn bit in USInCR4.
   4) No ACK signal is detected, and master transmits repeated START condition. In this case, load SLAn+R/W into the USInDR and set the STARTCn bit in USInCR4.

   After doing one of the actions above, write arbitrary value to USInST2 to release SCLn line. In case of 1) and 2), move to step 7. In case of 3), move to step 9 to handle STOP interrupt. In case of 4), move to step 6 after transmitting the data in USInDR, and if transfer direction bit is '0' go to master transmitter section.

9. This is the final step for master receiver function of I2C, handling STOP interrupt. The STOP bit indicates that data transfer between master and slave is over. To clear USInST2, write any value to USInST2. After this, I2C enters idle state.

The processes described above for master receiver operation of I2C can be depicted as the following figure.

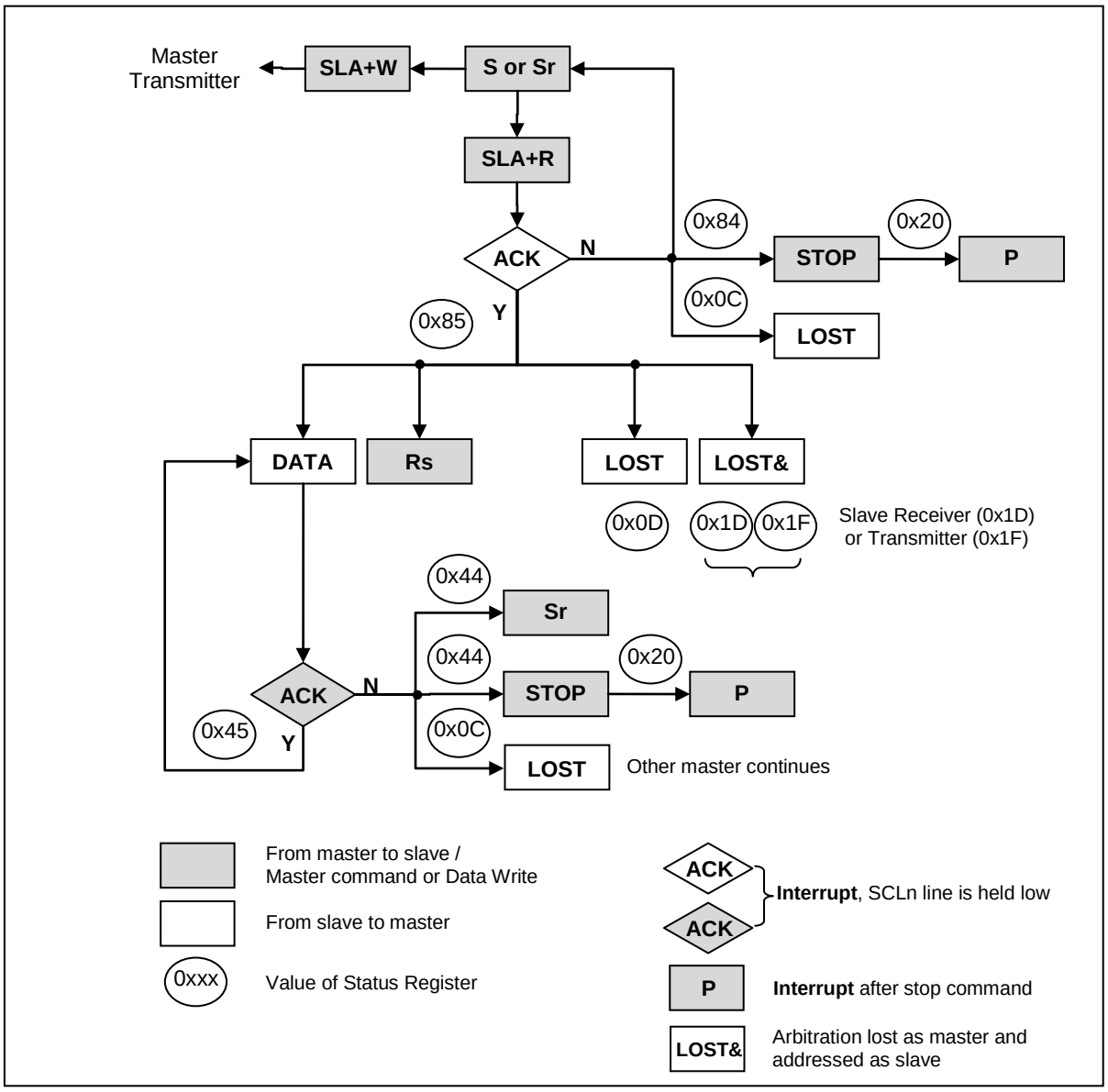



**Figure 11.76 Formats and States in the Master Receiver Mode (USIn, where n = 0 and 1)**

### 11.12.20.3 USI0/1 I2C Slave Transmitter

To operate I2C in slave transmitter, follow the recommended steps below.

1. If the main operating clock (SCLK) of the system is slower than that of SCLn, load value 0x00 into USInSDHR to make SDAn change within one system clock period from the falling edge of SCLn. Note that the hold time of SDAn is calculated by SDAH x period of SCLK where SDAH is multiple of number of SCLK coming from USInSDHR. When the hold time of SDAn is longer than the period of SCLK, I2C (slave) cannot transmit serial data properly.
2. Enable I2C by setting USInMS[1:0] bits in USInCR1 , IICnIE bit in USInCR4 and USInEN bit in USInCR2. This provides main clock to the peripheral.
3. When a START condition is detected, I2C receives one byte of data and compares it with USInSLA[6:0] bits in USInSAR. If the GCALLn bit in USInSAR is enabled, I2C compares the received data with value 0x00, the general call address.
4. If the received address does not equal to USInSLA[6:0] bits in USInSAR, I2C enters idle state ie, waits for another START condition. Else if the address equals to USInSLA[6:0] bits and the ACKnEN bit is enabled, I2C generates SSELn interrupt and the SCLn line is held LOW. Note that even if the address equals to USInSLA[6:0] bits, when the ACKnEN bit is disabled, I2C enters idle state. When SSELn interrupt occurs, load transmit data to USInDR and write arbitrary value to USInST2 to release SCLn line.
5. 1-Byte of data is being transmitted.
6. In this step, I2C generates TENDn interrupt and holds the SCLn line LOW regardless of the reception of ACK signal from master. Slave can select one of the following cases.

   1) No ACK signal is detected and I2C waits STOP or repeated START condition.
   2) ACK signal from master is detected. Load data to transmit into USInDR.

   After doing one of the actions above, write arbitrary value to USInST2 to release SCLn line. In case of 1) move to step 7 to terminate communication. In case of 2) move to step 5. In either case, a repeated START condition can be detected. For that case, move step 4.
7. This is the final step for slave transmitter function of I2C, handling STOP interrupt. The STOPCn bit indicates that data transfer between master and slave is over. To clear USInST2, write any value to USInST2. After this, I2C enters idle state.

The next figure shows flow chart for handling slave transmitter function of I2C.

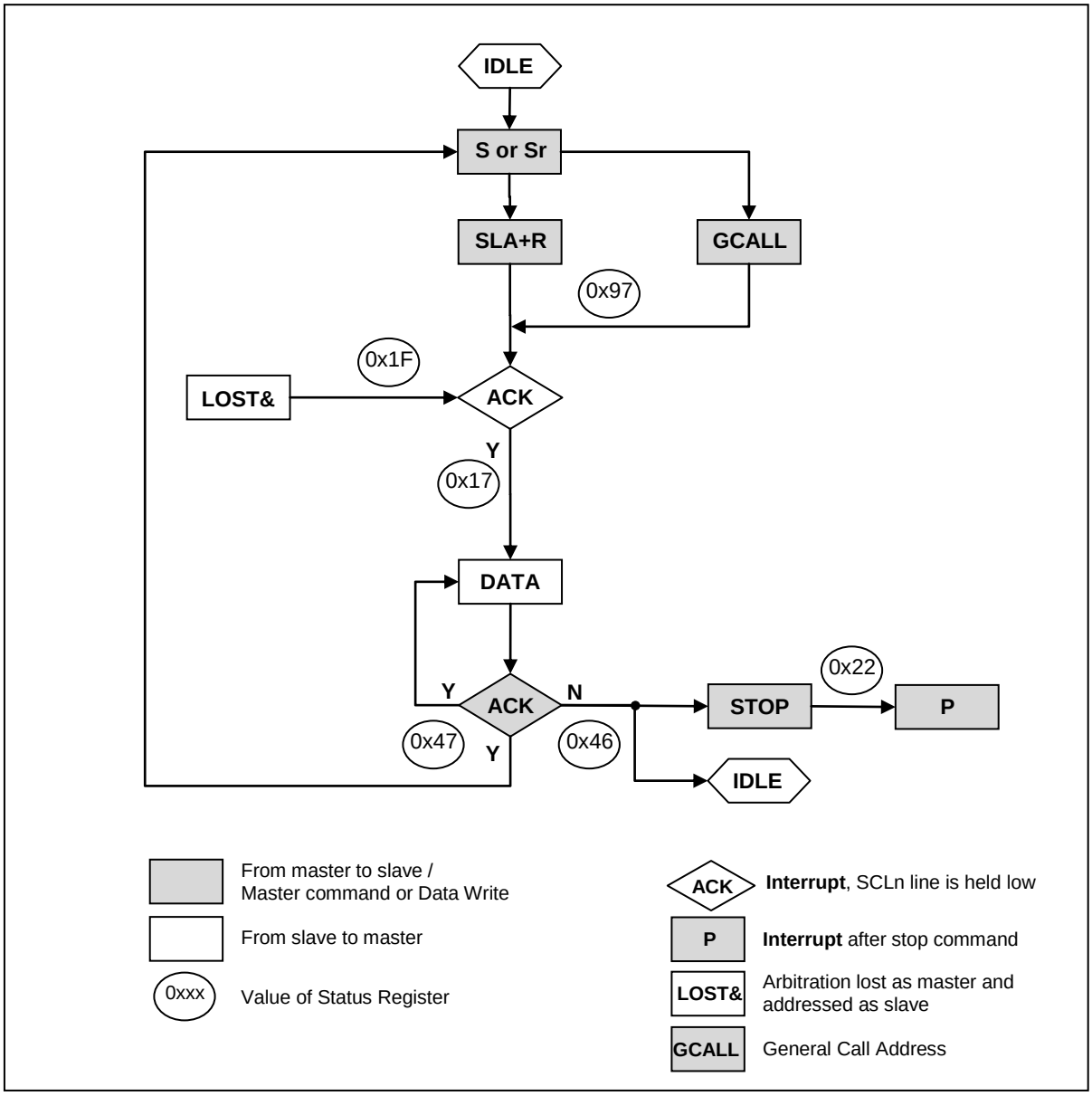



**Figure 11.77 Formats and States in the Slave Transmitter Mode (USIn, where n = 0 and 1)**

### 11.12.20.4 USI0/1 I2C Slave Receiver

To operate I2C in slave receiver, follow the recommended steps below.

1. If the main operating clock (SCLK) of the system is slower than that of SCLn, load value 0x00 into USInSDHR to make SDAn change within one system clock period from the falling edge of SCLn. Note that the hold time of SDAn is calculated by SDAH x period of SCLK where SDAH is multiple of number of SCLK coming from USInSDHR. When the hold time of SDAn is longer than the period of SCLK, I2C (slave) cannot transmit serial data properly.
2. Enable I2C by setting USInMS[1:0] bits in USInCR1, IICnIE bit in USInCR4 and USInEN bit in USInCR2. This provides main clock to the peripheral.
3. When a START condition is detected, I2C receives one byte of data and compares it with USInSLA[6:0] bits in USInSAR. If the GCALLn bit in USInSAR is enabled, I2C compares the received data with value 0x00, the general call address.
4. If the received address does not equal to SLAn bits in USInSAR, I2C enters idle state ie, waits for another START condition. Else if the address equals to SLAn bits and the ACKnEN bit is enabled, I2C generates SSELn interrupt and the SCLn line is held LOW. Note that even if the address equals to SLAn bits, when the ACKnEN bit is disabled, I2C enters idle state. When SSELn interrupt occurs and I2C is ready to receive data, write arbitrary value to USInST2 to release SCLn line.
5. 1-Byte of data is being received.
6. In this step, I2C generates TENDn interrupt and holds the SCLn line LOW regardless of the reception of ACK signal from master. Slave can select one of the following cases.

   1) No ACK signal is detected (ACKnEN=0) and I2C waits STOP or repeated START condition.
   2) ACK signal is detected (ACKnEN=1) and I2C can continue to receive data from master.

   After doing one of the actions above, write arbitrary value to USInST2 to release SCLn line. In case of 1) move to step 7 to terminate communication. In case of 2) move to step 5. In either case, a repeated START condition can be detected. For that case, move step 4.
7. This is the final step for slave receiver function of I2C, handling STOP interrupt. The STOPCn bit indicates that data transfer between master and slave is over. To clear USInST2, write any value to USInST2. After this, I2C enters idle state.

The process can be depicted as following figure when I2C operates in slave receiver mode.

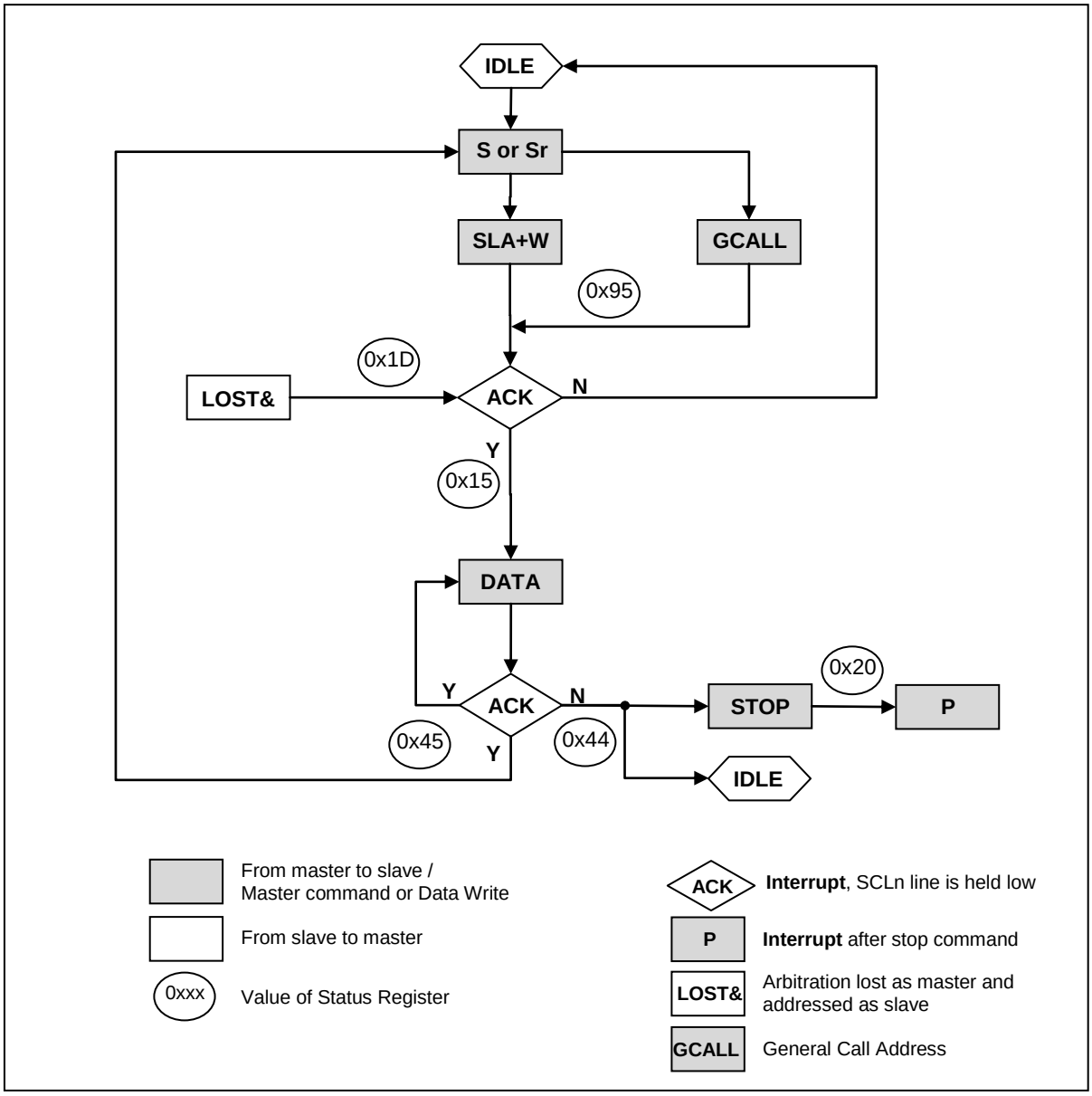
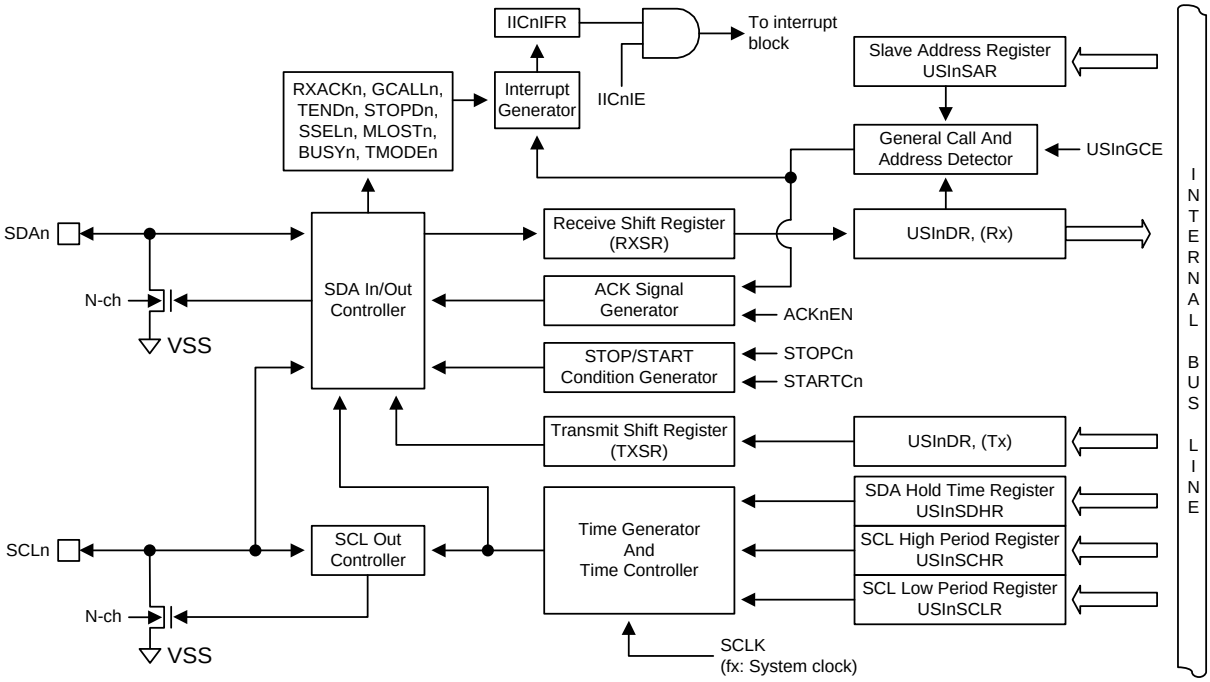



**Figure 11.78 Formats and States in the Slave Receiver Mode (USIn, where n = 0 and 1)**

### 11.12.21 USI0/1 I2C Block Diagram




NOTE) When the USI0/1 block is an I2C mode and the corresponding port is an sub-function for SCLn/SDAn pin, The SCLn/SDAn pins are automatically set to the N-channel open-drain outputs and the input latch is read in the case of reading the pins. The corresponding pull-up resistor is determined by the control register.

**Figure 11.79 USI0/1 I2C Block Diagram (where n = 0 and 1)**

## 11.12.22 Register Map

**Table 11-23 USI0/1 Register Map (where n = 0 and 1)**

| Name | Address | Dir | Default | Description |
|---|---|---|---|---|
| USInBD | 1022H/1032H (ESFR) | R/W | FFH | USIn Baud Rate Generation Register |
| USInDR | 1024H/1034H (ESFR) | R/W | 00H | USIn Data Register |
| USInSDHR | 1023H/1033H (ESFR) | R/W | 01H | USIn SDA Hold Time Register |
| USInSCHR | 1026H/1036H (ESFR) | R/W | 3FH | USIn SCL High Period Register |
| USInSCLR | 1025H/1035H (ESFR) | R/W | 3FH | USIn SCL Low Period Register |
| USInSAR | 1027H/1037H (ESFR) | R/W | 00H | USIn Slave Address Register |
| USInCR1 | 1018H/1028H (ESFR) | R/W | 00H | USIn Control Register 1 |
| USInCR2 | 1019H/1029H (ESFR) | R/W | 00H | USIn Control Register 2 |
| USInCR3 | 101AH/102AH (ESFR) | R/W | 00H | USIn Control Register 3 |
| USInCR4 | 101BH/102BH (ESFR) | R/W | 00H | USIn Control Register 4 |
| USInST1 | 1020H/1030H (ESFR) | R/W | 80H | USIn Status Register 1 |
| USInST2 | 1021H/1031H (ESFR) | R/W | 00H | USIn Status Register 2 |

## 11.12.23 USI0/1 Register Description

USI0/1 module consists of USI0/1 baud rate generation register (USInBD), USI0/1 data register (USInDR), USI0/1 SDA hold time register (USInSDHR), USI0/1 SCL high period register (USInSCHR), USI0/1 SCL low period Register (USInSCLR), USI0/1 slave address register (USInSAR), USI0/1 control register 1/2/3/4 (USInCR1/2/3/4), USI0/1 status register 1/2 (USInST1/2).

## 11.12.24 Register Description for USI0/1

**USInBD (USI0/1 Baud- Rate Generation Register: For UART and SPI mode) : 1022H/1032H (ESFR), n = 0, 1**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| USInBD7 | USInBD6 | USInBD5 | USInBD4 | USInBD3 | USInBD2 | USInBD1 | USInBD0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : FFH

**USInBD[7:0]** The value in this register is used to generate internal baud rate in asynchronous mode or to generate SCKn clock in SPI mode. To prevent malfunction, do not write '0' in asynchronous mode and do not write '0' or '1' in SPI mode.

NOTE) In common with USInSAR register, USInBD register is used for slave address register when the USI0/1 I2C mode.

**USInDR (USI0/1 Data Register: For UART, SPI, and I2C mode) : 1024H/1034H (ESFR), n = 0, 1**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| USInDR7 | USInDR6 | USInDR5 | USInDR4 | USInDR3 | USInDR2 | USInDR1 | USInDR0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**USInDR[7:0]** The USIn transmit buffer and receive buffer share the same I/O address with this DATA register. The transmit data buffer is the destination for data written to the USInDR register. Reading the USInDR register returns the contents of the receive buffer.

Write to this register only when the DREn flag is set. In SPI master mode, the SCK clock is generated when data are written to this register.

**USInSDHR (USI0/1 SDA Hold Time Register: For I2C mode) : 1023H/1033H (ESFR), n = 0, 1**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| USInSDHR7 | USInSDHR6 | USInSDHR5 | USInSDHR4 | USInSDHR3 | USInSDHR2 | USInSDHR1 | USInSDHR0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 01H

**USInSDHR[7:0]** The register is used to control SDAn output timing from the falling edge of SCLn in I2C mode.

NOTE) that SDA is changed after $t_{SCLK}$ X USInSDHR. In master mode, load half the value of USInSCLR to this register to make SDA change in the middle of SCL.

In slave mode, configure this register regarding the frequency of SCL from master.

The SDA is changed after $t_{SCLK}$ X (USInSDHR+2) in master mode. So, to insure operation in slave mode, the value $t_{SCLK}$ X (USInSDHR + 1) must be smaller than the period of SCL.

**USInSCHR (USI0/1 SCL High Period Register: For I2C mode) : 1026H/1036H (ESFR), n = 0, 1**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| USInSCHR7 | USInSCHR6 | USInSCHR5 | USInSCHR4 | USInSCHR3 | USInSCHR2 | USInSCHR1 | USInSCHR0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 3FH

**USInSCHR[7:0]** This register defines the high period of SCLn when it operates in I2C master mode.

The base clock is SCLK, the system clock, and the period is calculated by the formula: $t_{SCLK}$ X (4 X USInSCHR + 2) where $t_{SCLK}$ is the period of SCLK.

**So, the operating frequency of I2C master mode is calculated by the following equation.**

$$f_{I2C} = \frac{1}{t_{SCLK} \text{ X } (4 \text{ X } (\text{USInSCLR} + \text{USInSCHR}) + 4)}$$

**USInSCLR (USI0/1 SCL Low Period Register: For I2C mode) : 1025H/1035H (ESFR), n = 0, 1**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| USInSCLR7 | USInSCLR6 | USInSCLR5 | USInSCLR4 | USInSCLR3 | USInSCLR2 | USInSCLR1 | USInSCLR0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 3FH

**USInSCLR[7:0]** This register defines the high period of SCL when it operates in I2C master mode.

The base clock is SCLK, the system clock, and the period is calculated by the formula: $t_{SCLK}$ X (4 X USInSCLR +2) where $t_{SCLK}$ is the period of SCLK.

**USInSAR (USI0/1 Slave Address Register: For I2C mode) : 1027H/1037H (ESFR), n = 0, 1**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| USInSLA6 | USInSLA5 | USInSLA4 | USInSLA3 | USInSLA2 | USInSLA1 | USInSLA0 | USInGCE |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**USInSLA[6:0]** These bits configure the slave address of I2C when it operaties in I2C slave mode.

**USInGCE** This bit decides whether I2C allows general call address or not in I2C slave mode.

0 Ignore general call address

1 Allow general call address

**USInCR1 (USI0/1 Control Register 1: For UART, SPI, and I2C mode) : 1018H/1028H (ESFR), n = 0, 1**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| USInMS1 | USInMS0 | USInPM1 | USInPM0 | USInS2 | USInS1<br>ORDn | USInS0<br>CPHAn | CPOLn |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**USInMS[1:0]** Selects operation mode of USIn

| USInMS1 | USInMS0 | Operation mode |
|---|---|---|
| 0 | 0 | Asynchronous Mode (UART) |
| 0 | 1 | Synchronous Mode (UART) |
| 1 | 0 | I2C mode |
| 1 | 1 | SPI mode |

**USInPM[1:0]** Selects parity generation and check methods (only UART mode)

| USInPM1 | USInPM0 | Parity |
|---|---|---|
| 0 | 0 | No Parity |
| 0 | 1 | Reserved |
| 1 | 0 | Even Parity |
| 1 | 1 | Odd Parity |

**USInS[2:0]** When in asynchronous or synchronous mode of operation, selects the length of data bits in frame

| USInS2 | USInS1 | USInS0 | Data Length |
|---|---|---|---|
| 0 | 0 | 0 | 5 bit |
| 0 | 0 | 1 | 6 bit |
| 0 | 1 | 0 | 7 bit |
| 0 | 1 | 1 | 8 bit |
| 1 | 0 | 0 | Reserved |
| 1 | 0 | 1 | Reserved |
| 1 | 1 | 0 | Reserved |
| 1 | 1 | 1 | 9 bit |

**ORDn** This bit in the same bit position with USInS1. The MSB of the data byte is transmitted first when set to '1' and the LSB when set to '0' (onl SPI mode)

| | |
|---|---|
| 0 | LSB-first |
| 1 | MSB-first |

**CPOLn** This bit determines the clock polarity of ACK in synchronous or SPI mode.

| | |
|---|---|
| 0 | TXD change@Rising Edge, RXD change@Falling Edge |
| 1 | TXD change@Falling Edge, RXD change@Rising Edge |

**CPHAn** This bit is in the same bit position with USInS0. This bit determines if data are sampled on the leading or trailing edge of SCK (only SPI mode).

| CPOLn | CPHAn | Leading edge | Trailing edge |
|---|---|---|---|
| 0 | 0 | Sample (Rising) | Setup (Falling) |
| 0 | 1 | Setup (Rising) | Sample (Falling) |
| 1 | 0 | Sample (Falling) | Setup (Rising) |
| 1 | 1 | Setup (Falling) | Sample (Rising) |

**USInCR2 (USI0/1 Control Register 2: For UART, SPI, and I2C mode) : 1019H/1029H (ESFR), n = 0, 1**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DRIEn | TXCIEn | RXCIEn | WAKEIEn | TXEn | RXEn | USInEN | DBLSn |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**DRIEn** Interrupt enable bit for data register empty (only UART and SPI mode).

0 Interrupt from DREn is inhibited (use polling)

1 When DREn is set, request an interrupt

**TXCIEn** Interrupt enable bit for transmit complete (only UART and SPI mode).

0 Interrupt from TXCn is inhibited (use polling)

1 When TXCn is set, request an interrupt

**RXCIEn** Interrupt enable bit for receive complete (only UART and SPI mode).

0 Interrupt from RXCn is inhibited (use polling)

1 When RXCn is set, request an interrupt

**WAKEIEn** Interrupt enable bit for asynchronous wake in STOP mode. When device is in stop mode, if RXDn goes to low level an interrupt can be requested to wake-up system. (only UART mode). At that time the DRIEn bit and USInST1 register value should be set to ‘0b’ and “00H”, respectively.

0 Interrupt from Wake is inhibited

1 When WAKEn is set, request an interrupt

**TXEn** Enables the transmitter unit (only UART and SPI mode).

0 Transmitter is disabled

1 Transmitter is enabled

**RXEn** Enables the receiver unit (only UART and SPI mode).

0 Receiver is disabled

1 Receiver is enabled

**USInEN** Activate USIn function block by supplying.

0 USIn is disabled

1 USIn is enabled

**DBLSn** This bit selects receiver sampling rate (only UART).

0 Normal asynchronous operation

1 Double Speed asynchronous operation

**USInCR3 (USI0/1 Control Register 3: For UART, SPI, and I2C mode) : 101AH/102AH (ESFR), n = 0, 1**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| MASTERn | LOOPSn | DISSCKn | USInSSEN | FXCHn | USInSB | USInTX8 | USInRX8 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R |

Initial value : 00H

**MASTERn** Selects master or slave in SPI and synchronous mode operation and controls the direction of SCKn pin

0 Slave mode operation (External clock for SCK).

1 Master mode operation(Internal clock for SCK).

**LOOPSn** Controls the loop back mode of USIn for test mode (only UART and SPI mode)

0 Normal operation

1 Loop Back mode

**DISSCKn** In synchronous mode of operation, selects the waveform of SCKn output

0 ACK is free-running while UART is enabled in synchronous master mode

1 ACK is active while any frame is on transferring

**USInSSEN** This bit controls the SSn pin operation (only SPI mode)

0 Disable

1 Enable (The SSn pin should be a normal input)

**FXCHn** SPI port function exchange control bit (only SPI mode)

0 No effect

1 Exchange MOSIn and MISOn function

**USInSB** Selects the length of stop bit in asynchronous or synchronous mode of operation.

0 1 Stop Bit

1 2 Stop Bit

**USInTX8** The ninth bit of data frame in asynchronous or synchronous mode of operation. Write this bit first before loading the USInDR register

0 MSB ($9^{th}$ bit) to be transmitted is ‘0’

1 MSB ($9^{th}$ bit) to be transmitted is ‘1’

**USInRX8** The ninth bit of data frame in asynchronous or synchronous mode of operation. Read this bit first before reading the receive buffer (only UART mode).

0 MSB ($9^{th}$ bit) received is ‘0’

1 MSB ($9^{th}$ bit) received is ‘1’

**USInCR4 (USI0/1 Control Register 4: For I2C mode) : 101BH/102BH (ESFR), n = 0, 1**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| IICnIFR | – | TXDLYENBn | IICnIE | ACKnEN | IMASTERn | STOPCn | STARTCn |
| R | – | R/W | R/W | R/W | R | R/W | R/W |

Initial value : 00H

**IICnIFR** This is an interrupt flag bit for I2C mode. When an interrupt occurs, this bit becomes '1'. This bit is cleared when write any values in th USInST2.

0 I2C interrupt no generation

1 I2C interrupt generation

**TXDLYENBn** USInSDHR register control bit

0 Enable USInSDHR register

1 Disable USInSDHR register

**IICnIE** Interrupt Enable bit for I2C mode

0 Interrupt from I2C is inhibited (use polling)

1 Enable interrupt for I2C

**ACKnEN** Controls ACK signal Generation at ninth SCL period.

0 No ACK signal is generated (SDA =1)

1 ACK signal is generated (SDA =0)

NOTES) ACK signal is output (SDA =0) for the following 3 cases.

1. When received address packet equals to USInSLA bits in USInSAR.

2. When received address packet equals to value 0x00 with GCALLn enabled.

3. When I2C operates as a receiver (master or slave)

**IMASTERn** Represent operating mode of I2C

0 I2C is in slave mode

1 I2C is in master mode

**STOPCn** When I2C is master, STOP condition generation

0 No effect

1 STOP condition is to be generated

**STARTCn** When I2C is master, START condition generation

0 No effect

1 START or repeated START condition is to be generated

**USInST1 (USI0/1 Status Register 1: For UART and SPI mode) : 1020H/1030H (ESFR), n = 0, 1**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DREn | TXCn | RXCn | WAKEn | USInRST | DORn | FEn | PEn |
| R/W | R/W | R | R/W | R/W | R | R/W | R/W |

Initial value : 80H

**DREn** The DREn flag indicates if the transmit buffer (USInDR) is ready to receive new data. If DREn is '1', the buffer is empty and ready to be written. This flag can generate a DREn interrupt.

0 Transmit buffer is not empty.

1 Transmit buffer is empty.

**TXCn** This flag is set when the entire frame in the transmit shift register has been shifted out and there is no new data currently present in the transmit buffer. This flag is automatically cleared when the interrupt service routine of a TXCn interrupt is executed. This flag can generate a TXCn interrupt. This bit is automatically cleared.

0 Transmission is ongoing.

1 Transmit buffer is empty and the data in transmit shift register are shifted out completely.

**RXCn** This flag is set when there are unread data in the receive buffer and cleared when all the data in the receive buffer are read. The RXCn flag can be used to generate a RXCn interrupt.

0 There is no data unread in the receive buffer

1 There are more than 1 data in the receive buffer

**WAKEn** This flag is set when the RXDn pin is detected low while the CPU is in STOP mode. This flag can be used to generate a WAKEn interrupt. This bit is set only when in asynchronous mode of operation. This bit should be cleared by program software. (only UART mode)

0 No WAKE interrupt is generated.

1 WAKE interrupt is generated

**USInRST** This is an internal reset and only has effect on USIn. Writing '1' to this bit initializes the internal logic of USIn and this bit is automatically cleared to '0'.

0 No operation

1 Reset USIn

**DORn** This bit is set if a Data OverRun occurs. While this bit is set, the incoming data frame is ignored. This flag is valid until the receive buffer is read.

0 No Data OverRun

1 Data OverRun detected

**FEn** This bit is set if the first stop bit of next character in the receive buffer is detected as '0'. This bit is valid until the receive buffer is read. (only UART mode)

0 No Frame Error

1 Frame Error detected

**PEn** This bit is set if the next character in the receive buffer has a Parity Error to be received while Parity Checking is enabled. This bit is valid until the receive buffer is read. (only UART mode)

0 No Parity Error

1 Parity Error detected

**USInST2 (USI0/1 Status Register 2: For I2C mode) : 1021H/1031H (ESFR), n = 0, 1**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| GCALLn | TENDn | STOPDn | SSELn | MLOSTn | BUSYn | TMODEn | RXACKn |
| R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**GCALLn**(NOTE) This bit has different meaning depending on whether I2C is master or slave. When I2C is a master, this bit represents whether it received AACK (address ACK) from slave.

0 No AACK is received (Master mode)

1 AACK is received (Master mode)

When I2C is a slave, this bit is used to indicated general call.

0 General call address is not detected (Slave mode)

1 General call address is detected (Slave mode)

**TENDn**(NOTE) This bit is set when 1-byte of data is transferred completely

0 1 byte of data is not completely transferred

1 1 byte of data is completely transferred

**STOPDn**(NOTE) This bit is set when a STOP condition is detected.

0 No STOP condition is detected

1 STOP condition is detected

**SSELn**(NOTE) This bit is set when I2C is addressed by other master.

0 I2C is not selected as a slave

1 I2C is addressed by other master and acts as a slave

**MLOSTn**(NOTE) This bit represents the result of bus arbitration in master mode.

0 I2C maintains bus mastership

1 I2C maintains bus mastership during arbitration process

**BUSYn** This bit reflects bus status.

0 I2C bus is idle, so a master can issue a START condition

1 I2C bus is busy

**TMODEn** This bit is used to indicate whether I2C is transmitter or receiver.

0 I2C is a receiver

1 I2C is a transmitter

**RXACKn** This bit shows the state of ACK signal

0 No ACK is received

1 ACK is received at ninth SCL period

NOTE) These bits can be source of interrupt.

When an I2C interrupt occurs except for STOP mode, the SCL line is hold LOW. To release SCL, write arbitrary value to USInST2. When USInST2 is written, the TENDn, STOPDn, SSELn, MLOSTn, and RXACKn bits are cleared.

## 11.12.25 Baud Rate setting (example)

**Table 11-24 Examples of USI0BD and USI1BD Settings for Commonly Used Oscillator Frequencies**

| Baud Rate | fx=1.00MHz | | fx=1.8432MHz | | fx=2.00MHz | |
|---|---|---|---|---|---|---|
| | USI0BD/USI1BD | ERROR | USI0BD/USI1BD | ERROR | USI0BD/USI1BD | ERROR |
| 2400 | 25 | 0.2% | 47 | 0.0% | 51 | 0.2% |
| 4800 | 12 | 0.2% | 23 | 0.0% | 25 | 0.2% |
| 9600 | 6 | -7.0% | 11 | 0.0% | 12 | 0.2% |
| 14.4k | 3 | 8.5% | 7 | 0.0% | 8 | -3.5% |
| 19.2k | 2 | 8.5% | 5 | 0.0% | 6 | -7.0% |
| 28.8k | 1 | 8.5% | 3 | 0.0% | 3 | 8.5% |
| 38.4k | 1 | -18.6% | 2 | 0.0% | 2 | 8.5% |
| 57.6k | - | - | 1 | -25.0% | 1 | 8.5% |
| 76.8k | - | - | 1 | 0.0% | 1 | -18.6% |
| 115.2k | - | - | - | - | - | - |
| 230.4k | - | - | - | - | - | - |

(continued)

| Baud Rate | fx=3.6864MHz | | fx=4.00MHz | | fx=7.3728MHz | |
|---|---|---|---|---|---|---|
| | USI0BD/USI1BD | ERROR | USI0BD/USI1BD | ERROR | USI0BD/USI1BD | ERROR |
| 2400 | 95 | 0.0% | 103 | 0.2% | 191 | 0.0% |
| 4800 | 47 | 0.0% | 51 | 0.2% | 95 | 0.0% |
| 9600 | 23 | 0.0% | 25 | 0.2% | 47 | 0.0% |
| 14.4k | 15 | 0.0% | 16 | 2.1% | 31 | 0.0% |
| 19.2k | 11 | 0.0% | 12 | 0.2% | 23 | 0.0% |
| 28.8k | 7 | 0.0% | 8 | -3.5% | 15 | 0.0% |
| 38.4k | 5 | 0.0% | 6 | -7.0% | 11 | 0.0% |
| 57.6k | 3 | 0.0% | 3 | 8.5% | 7 | 0.0% |
| 76.8k | 2 | 0.0% | 2 | 8.5% | 5 | 0.0% |
| 115.2k | 1 | 0.0% | 1 | 8.5% | 3 | 0.0% |
| 230.4k | - | - | - | - | 1 | 0.0% |
| 250k | - | - | - | - | 1 | -7.8% |
| 0.5M | - | - | - | - | - | - |

(continued)

| Baud Rate | fx=8.00MHz | | fx=11.0592MHz | |
|---|---|---|---|---|
| | USI0BD/USI1BD | ERROR | USI0BD/USI1BD | ERROR |
| 2400 | 207 | 0.2% | - | - |
| 4800 | 103 | 0.2% | 143 | 0.0% |
| 9600 | 51 | 0.2% | 71 | 0.0% |
| 14.4k | 34 | -0.8% | 47 | 0.0% |
| 19.2k | 25 | 0.2% | 35 | 0.0% |
| 28.8k | 16 | 2.1% | 23 | 0.0% |
| 38.4k | 12 | 0.2% | 17 | 0.0% |
| 57.6k | 8 | -3.5% | 11 | 0.0% |
| 76.8k | 6 | -7.0% | 8 | 0.0% |
| 115.2k | 3 | 8.5% | 5 | 0.0% |
| 230.4k | 1 | 8.5% | 2 | 0.0% |
| 250k | 1 | 0.0% | 2 | -7.8% |
| 0.5M | - | - | - | - |
| 1M | - | - | - | - |

## 11.13 12-Bit A/D Converter

### 11.13.1 Overview

The analog-to-digital converter (A/D) allows conversion of an analog input signal to corresponding 12-bit digital value. The A/D module has twelve analog inputs. The output of the multiplexer is the input into the converter which generates the result through successive approximation. The A/D module has four registers which are the A/D converter control high register (ADCCRH), A/D converter control low register (ADCCRL), A/D converter data high register (ADCDRH), and A/D converter data low register (ADCDRL). The channels to be converted are selected by setting ADSEL[3:0]. To execute A/D conversion, TRIG[2:0] bits should be set to 'xxx'. The register ADCDRH and ADCDRL contains the results of the A/D conversion. When the conversion is completed, the result is loaded into the ADCDRH and ADCDRL, the A/D conversion status bit AFLAG is set to '1', and the A/D interrupt is set. During A/D conversion, AFLAG bit is read as '0'.

### 11.13.2 Conversion Timing

The A/D conversion process requires 4 steps (4 clock edges) to convert each bit and 10 clocks to set up A/D conversion. Therefore, total of 58 clocks are required to complete a 12-bit conversion: When fxx/8 is selected for conversion clock with a 12MHz fxx clock frequency, one clock cycle is 0.66 μs. Each bit conversion requires 4 clocks, the conversion rate is calculated as follows:

4 clocks/bit × 12 bits + set-up time = 58 clocks,

58 clock × 0.66 μs = 38.28 μs at 1.5 MHz (12 MHz/8)

NOTE) The A/D converter needs at least 20 μs for conversion time. So you must set the conversion time more than 20 μs.

### 11.13.3 Block Diagram

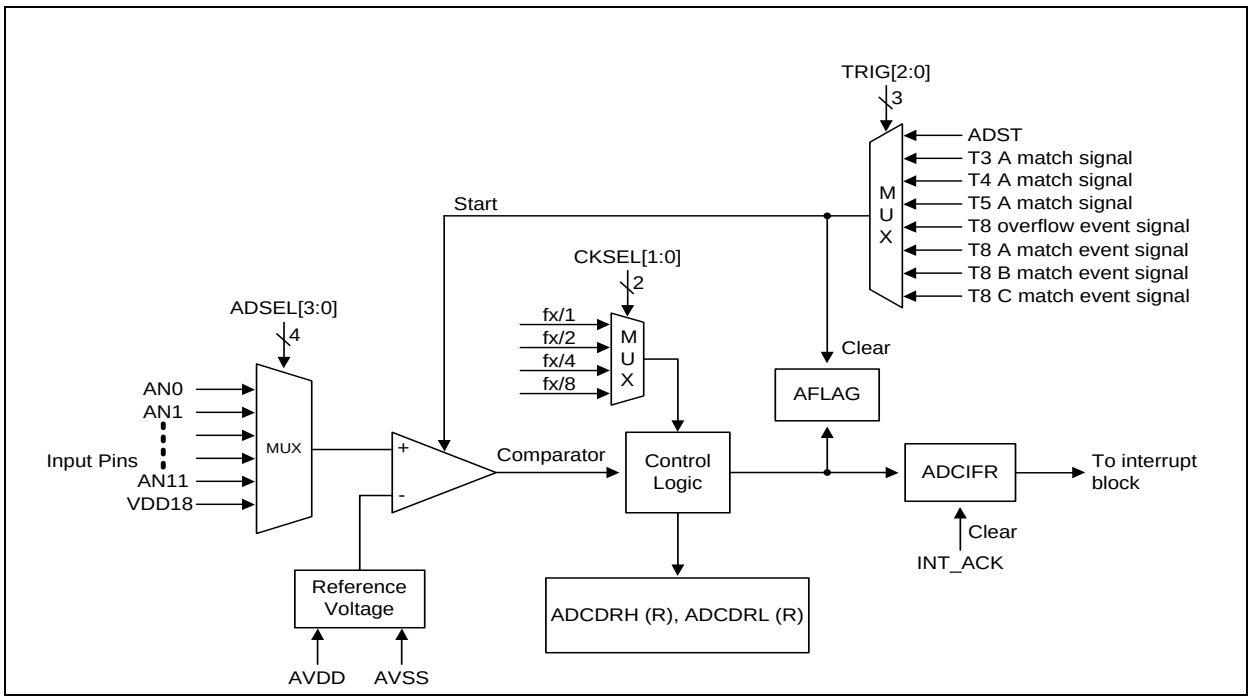



**Figure 11.80 12-bit ADC Block Diagram**

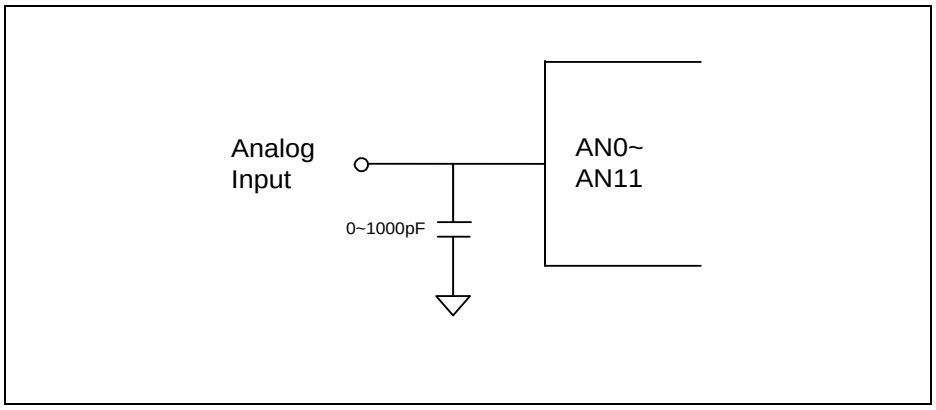



**Figure 11.81 A/D Analog Input Pin with Capacitor**

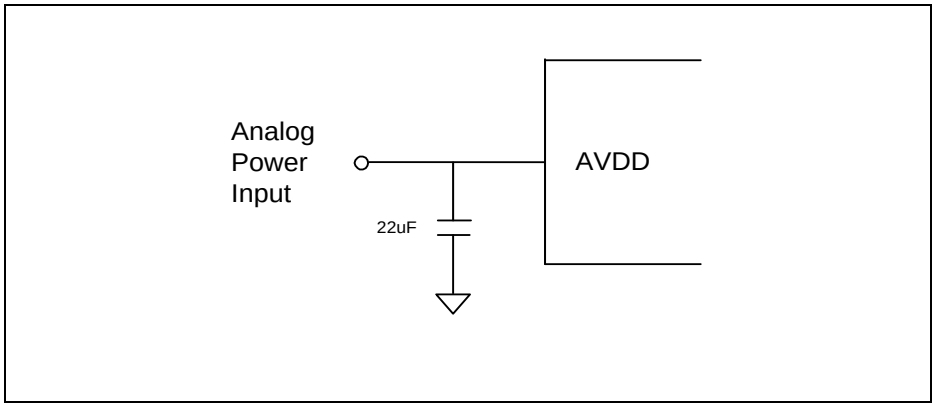



**Figure 11.82 A/D Power Pin with Capacitor**

## 11.13.4 ADC Operation

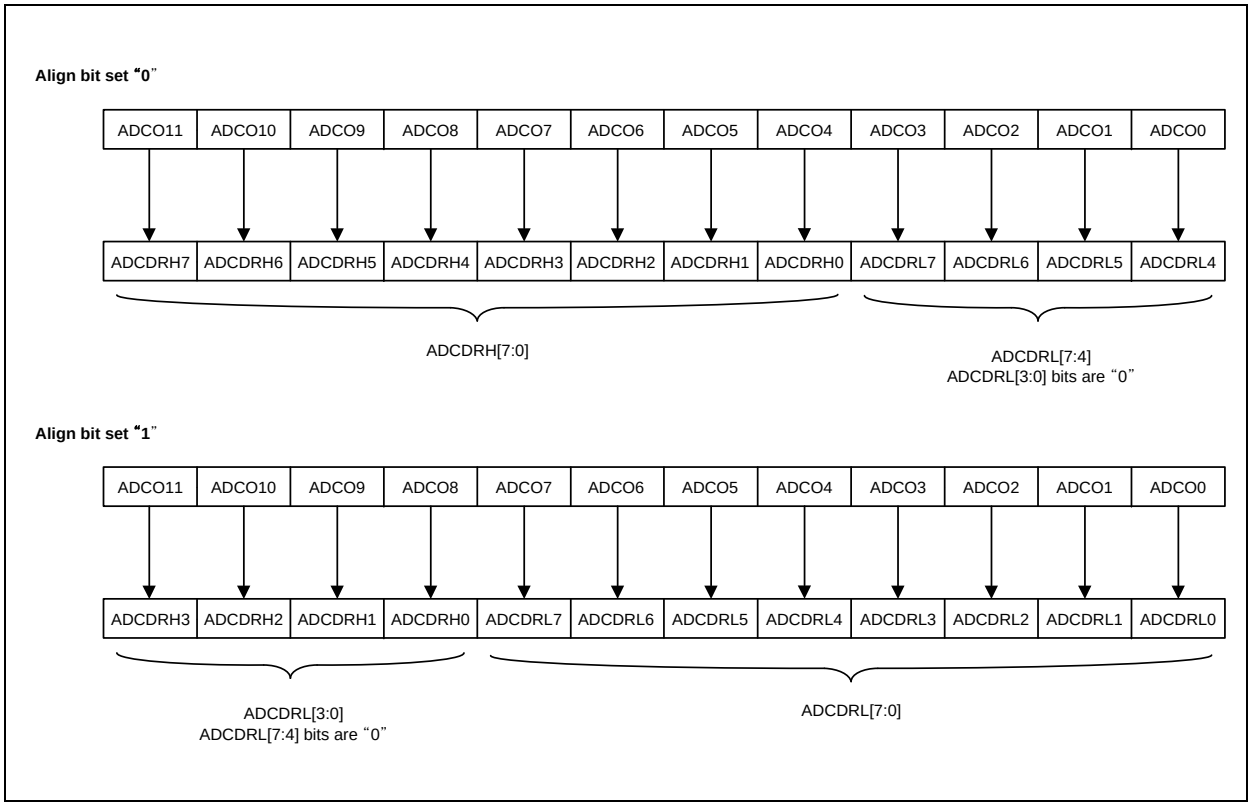



**Figure 11.83 ADC Operation for Align Bit**

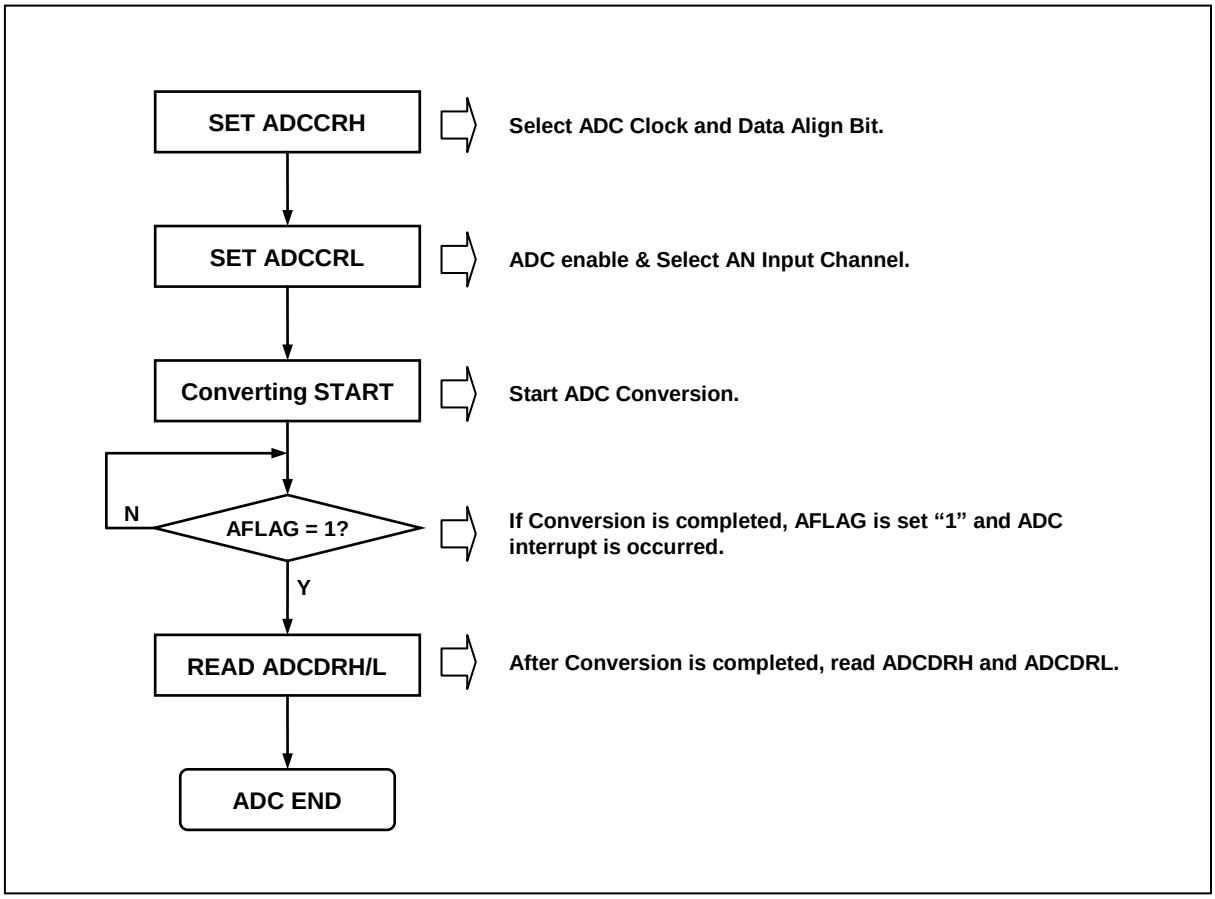



**Figure 11.84 A/D Converter Operation Flow**

## 11.13.5 Register Map

**Table 11-25 ADC Register Map**

| Name | Address | Dir | Default | Description |
|---|---|---|---|---|
| ADCDRH | E3H | R | xxH | A/D Converter Data High Register |
| ADCDRL | E2H | R | xxH | A/D Converter Data Low Register |
| ADCCRH | EBH | R/W | 00H | A/D Converter Control High Register |
| ADCCRL | EAH | R/W | 00H | A/D Converter Control Low Register |

## 11.13.6 ADC Register Description

The ADC register consists of A/D converter data high register (ADCDRH), A/D converter data low register (ADCDRL), A/D converter control high register (ADCCRH) and A/D converter control low register (ADCCRL).

## 11.13.7 Register Description for ADC

**ADCDRH (A/D Converter Data High Register) : E3H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ADDM11 | ADDM10 | ADDM9 | ADDM8 | ADDM7<br>ADDL11 | ADDM6<br>ADDL10 | ADDM5<br>ADDL9 | ADDM4<br>ADDL8 |
| R | R | R | R | R | R | R | R |

Initial value : xxH

**ADDM[11:4]** MSB align, A/D Converter High Result (8-bit)

**ADDL[11:8]** LSB align, A/D Converter High Result (4-bit)

**ADCDRL (A/D Converter Data Low Register) : E2H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ADDM3<br>ADDL7 | ADDM2<br>ADDL6 | ADDM1<br>ADDL5 | ADDM0<br>ADDL4 | ADDL3 | ADDL2 | ADDL1 | ADDL0 |
| R | R | R | R | R- | R | R | R |

Initial value : xxH

**ADDM[3:0]** MSB align, A/D Converter Low Result (4-bit)

**ADDL[7:0]** LSB align, A/D Converter Low Result (8-bit)

**ADCCRH (A/D Converter High Register) : EBH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| - | ADCIFR | TRIG2 | TRIG1 | TRIG0 | ALIGN | CKSEL1 | CKSEL0 |
| - | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**ADCIFR** When ADC interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or auto clear by INT_ACK signal.

0 ADC Interrupt no generation

1 ADC Interrupt generation

**TRIG[2:0]** A/D Trigger Signal Selection

| TRIG2 | TRIG1 | TRIG0 | Description |
|---|---|---|---|
| 0 | 0 | 0 | ADST |
| 0 | 0 | 1 | Timer 3 A match signal |
| 0 | 1 | 0 | Timer 4 A match signal |
| 0 | 1 | 1 | Timer 5 A match signal |
| 1 | 0 | 0 | Timer 8 overflow event signal |
| 1 | 0 | 1 | Timer 8 A match event signal |
| 1 | 1 | 0 | Timer 8 B match event signal |
| 1 | 1 | 1 | Timer 8 C match event signal |

**ALIGN** A/D Converter data align selection.

0 MSB align (ADCDRH[7:0], ADCDRL[7:4])

1 LSB align (ADCRDH[3:0], ADCDRL[7:0])

**CKSEL[1:0]** A/D Converter Clock selection

| CKSEL1 | CKSEL0 | Description |
|---|---|---|
| 0 | 0 | fx/1 |
| 0 | 1 | fx/2 |
| 1 | 0 | fx/4 |
| 1 | 1 | fx/8 |

**ADCCRL (A/D Converter Counter Low Register) : EAH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| STBY | ADST | - | AFLAG | ADSEL3 | ADSEL2 | ADSEL1 | ADSEL0 |
| R/W | R/W | - | R | R/W | R/W | R/W | R/W |

Initial value : 00H

**STBY** Control Operation of A/D
(The ADC module is automatically disabled at stop mode)

0 ADC module disable

1 ADC module enable

**ADST** Control A/D Conversion Start.

0 No effect

1 Trigger signal generation for conversion start

**AFLAG** A/D Converter Operation State (This bit is cleared to '0' when the STBY bit is set to '0' or when the CPU is at STOP mode)

0 During A/D Conversion

1 A/D Conversion finished

**ADSEL[3:0]** A/D Converter input selection

| ADSEL3 | ADSEL2 | ADSEL1 | ADSEL0 | Description |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | AN0 |
| 0 | 0 | 0 | 1 | AN1 |
| 0 | 0 | 1 | 0 | AN2 |
| 0 | 0 | 1 | 1 | AN3 |
| 0 | 1 | 0 | 0 | AN4 |
| 0 | 1 | 0 | 1 | AN5 |
| 0 | 1 | 1 | 0 | AN6 |
| 0 | 1 | 1 | 1 | AN7 |
| 1 | 0 | 0 | 0 | AN8 |
| 1 | 0 | 0 | 1 | AN9 |
| 1 | 0 | 1 | 0 | AN10 |
| 1 | 0 | 1 | 1 | AN11 |
| 1 | 1 | 0 | 0 | Not used |
| 1 | 1 | 0 | 1 | Not used |
| 1 | 1 | 1 | 0 | Not used |
| 1 | 1 | 1 | 1 | VDD18 |

## 11.14 LCD Driver

### 11.14.1 Overview

The LCD driver is controlled by the LCD control register (LCDCRH/L) and LCD driver contrast control register (LCDCCR). The LCLK[1:0] determines the frequency of COM signal scanning of each segment output. A RESET clears the LCD control register LCDCRH, LCDCRL and LCDCCR values to logic '0'.

The LCD display can continue operating during IDLE and STOP modes if a sub-frequency clock is used as system clock source.

The clock and duty for LCD driver is automatically initialized by hardware, whenever LCDCRH register data value is rewritten. So, don't rewrite LCDCRH frequently.

## 11.14.2 LCD Display RAM Organization

Display data are stored to the display data area in the external data memory.

The display data which stored to the display external data area (address 0000H-0027H) are read automatically and sent to the LCD driver by the hardware. The LCD driver generates the segment signals and common signals in accordance with the display data and drive method. Therefore, display patterns can be changed by only overwriting the contents of the display external data area with a program.

Figure 11-89 shows the correspondence between the display external data area and the COM/SEG pins. The LCD is turned on lights when the display data is “1” and turned off when “0”.

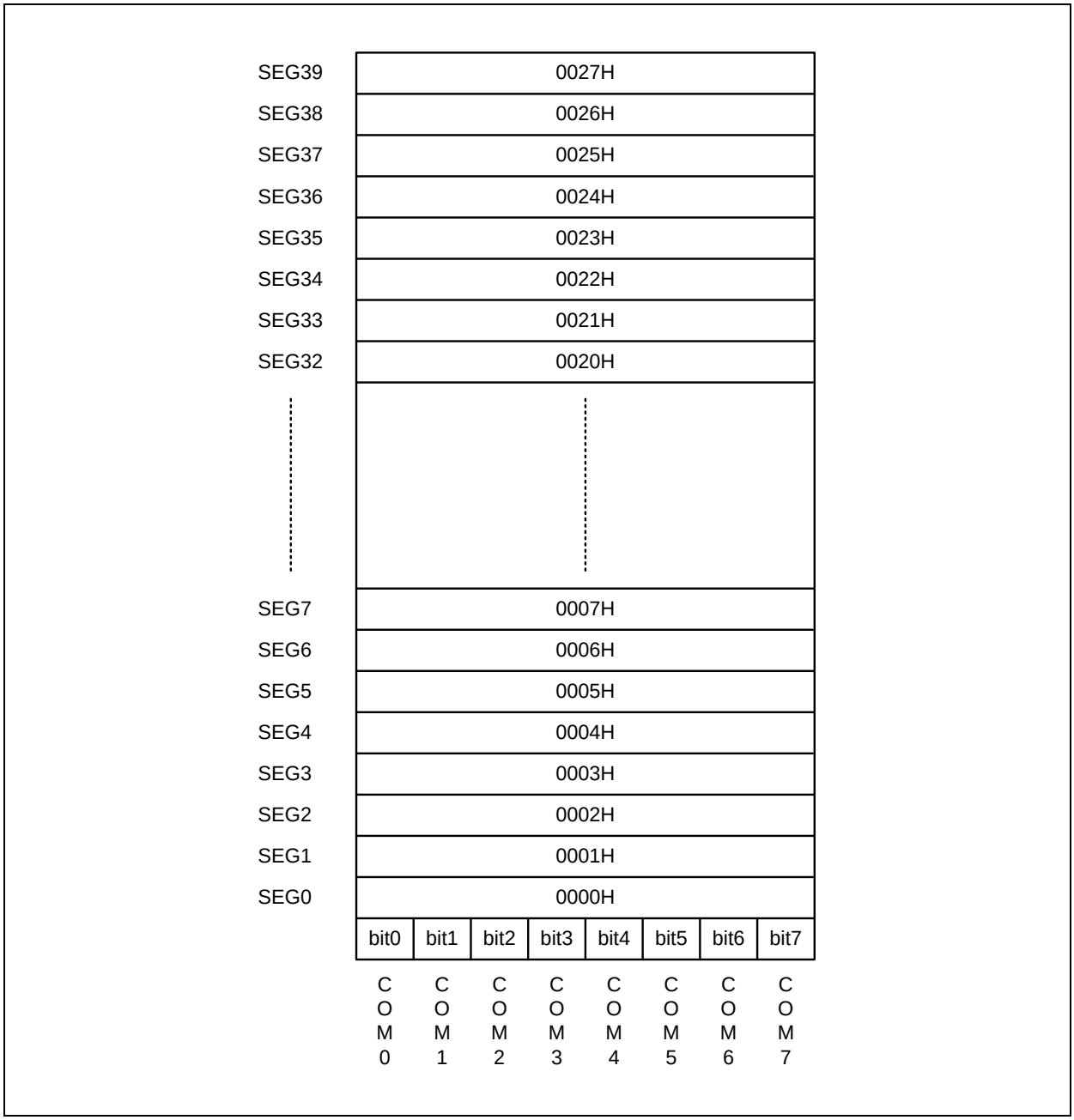



**Figure 11.85 LCD Circuit Block Diagram**

### 11.14.3 LCD Signal Waveform

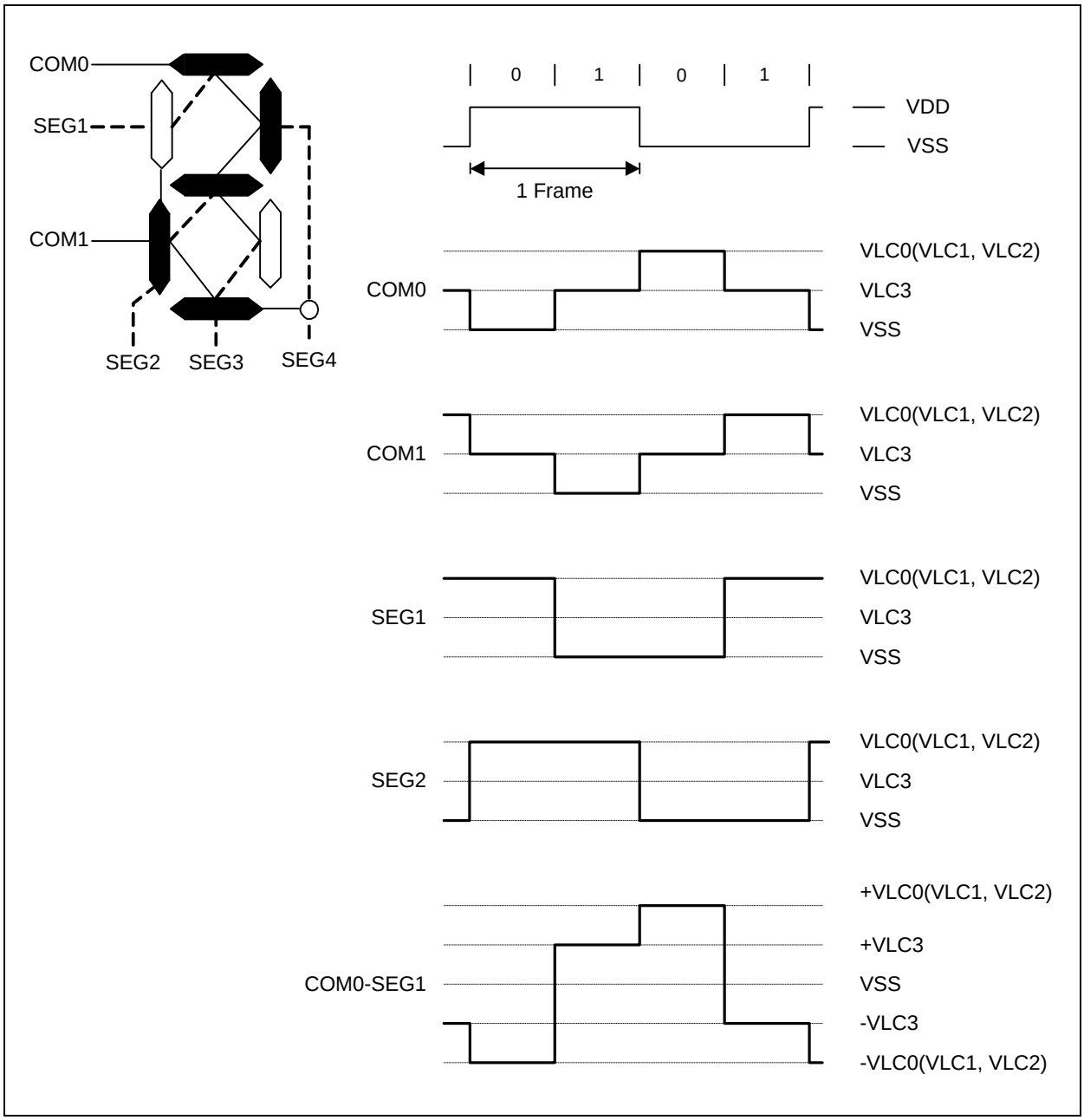



**Figure 11.86 LCD Signal Waveforms (1/2Duty, 1/2Bias)**

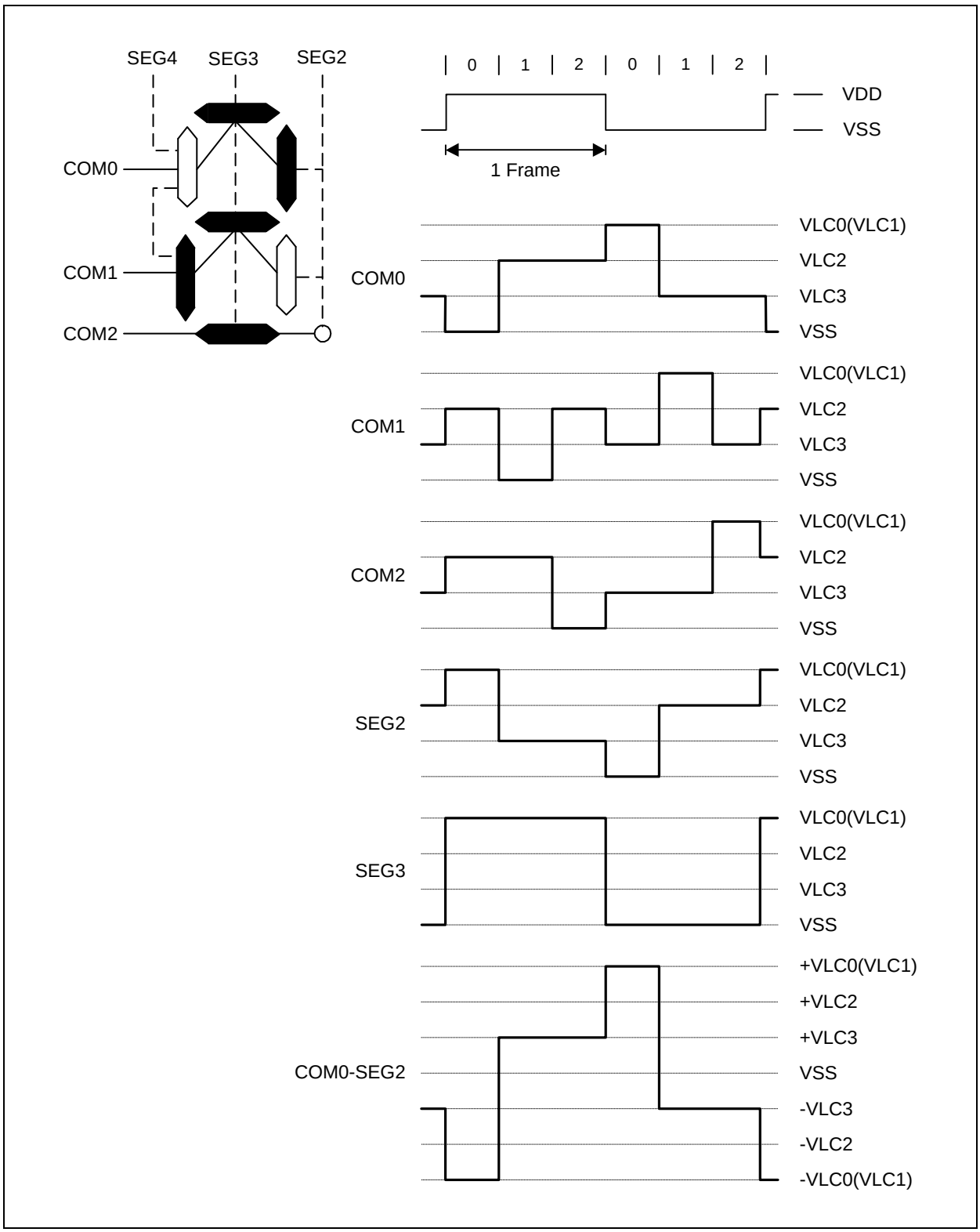



**Figure 11.87 LCD Signal Waveforms (1/3Duty, 1/3Bias)**

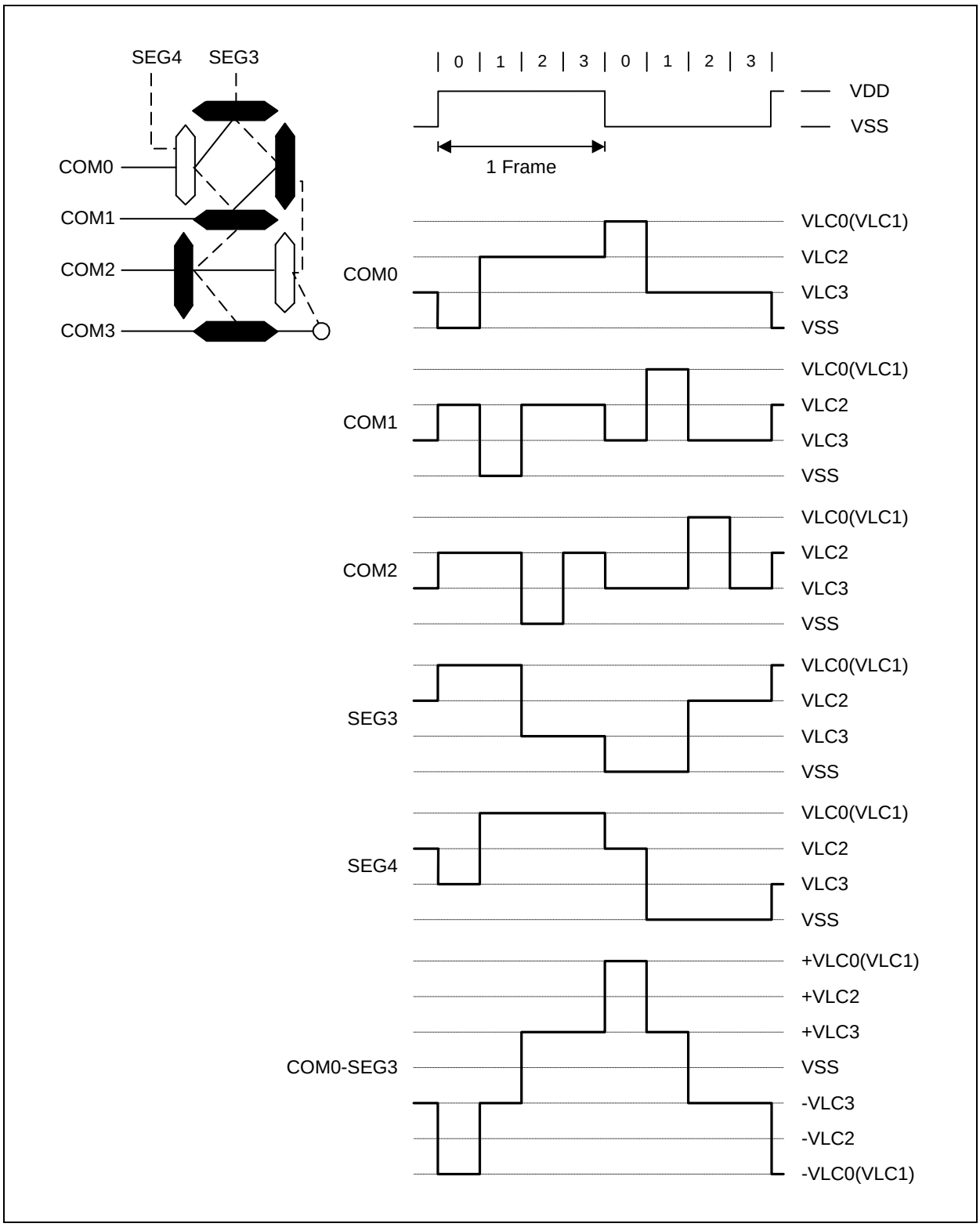



**Figure 11.88 LCD Signal Waveforms (1/4Duty, 1/3Bias)**

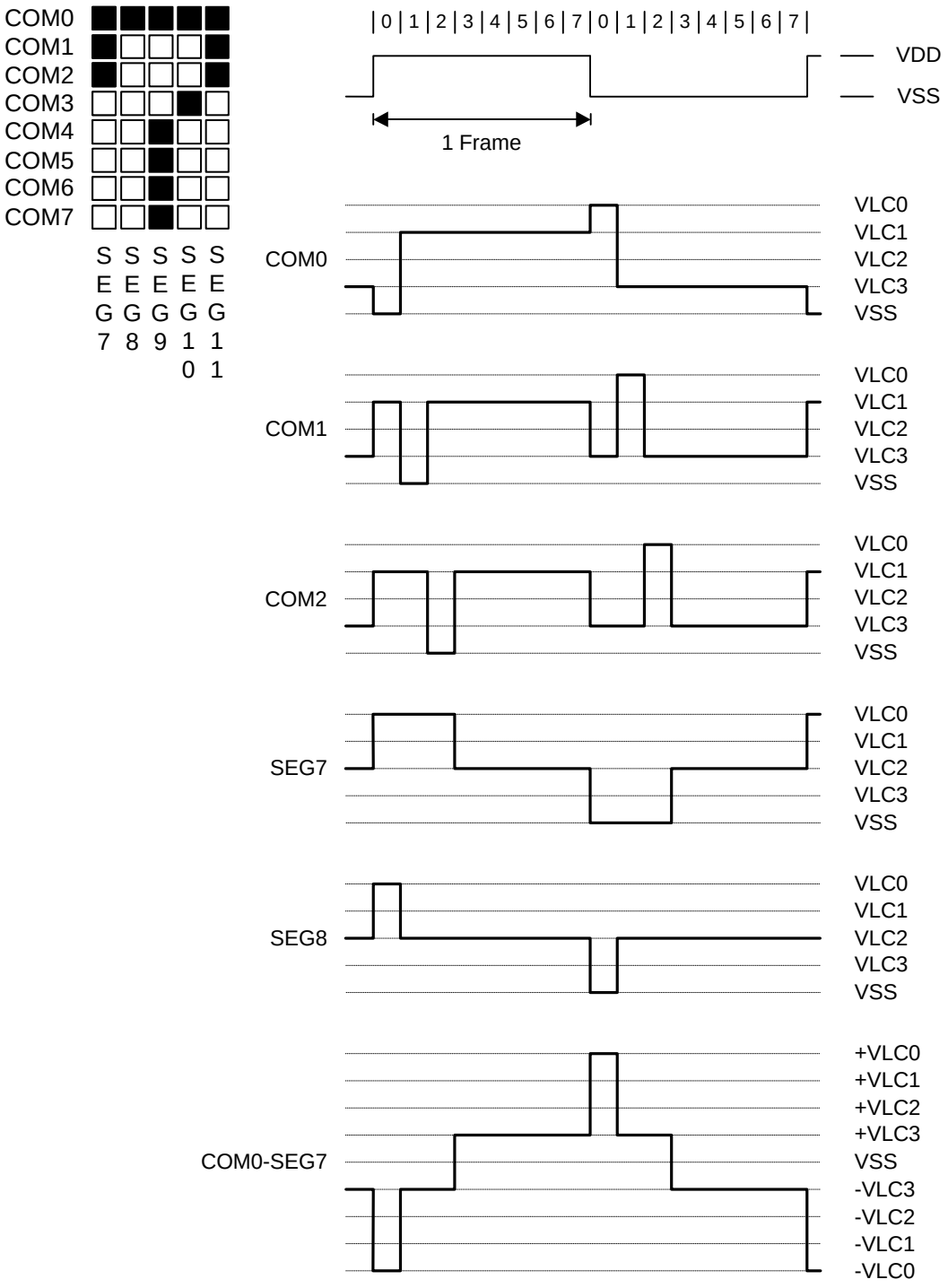
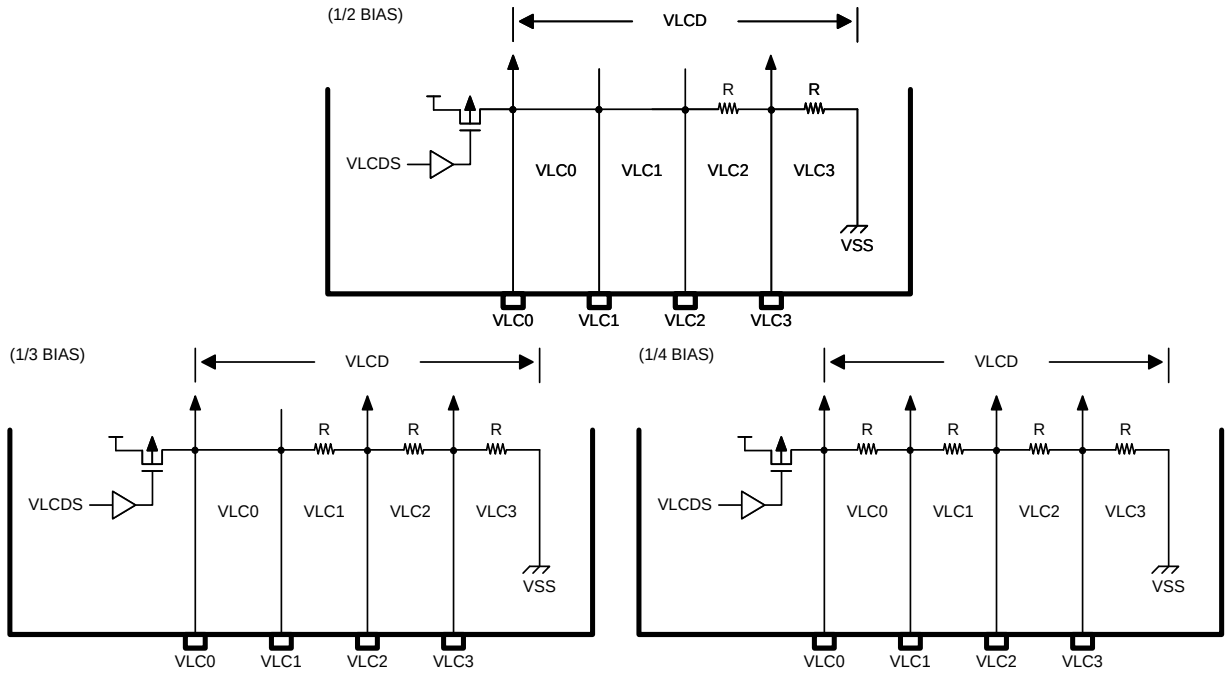



**Figure 11.89 LCD Signal Waveforms (1/8Duty, 1/4Bias)**

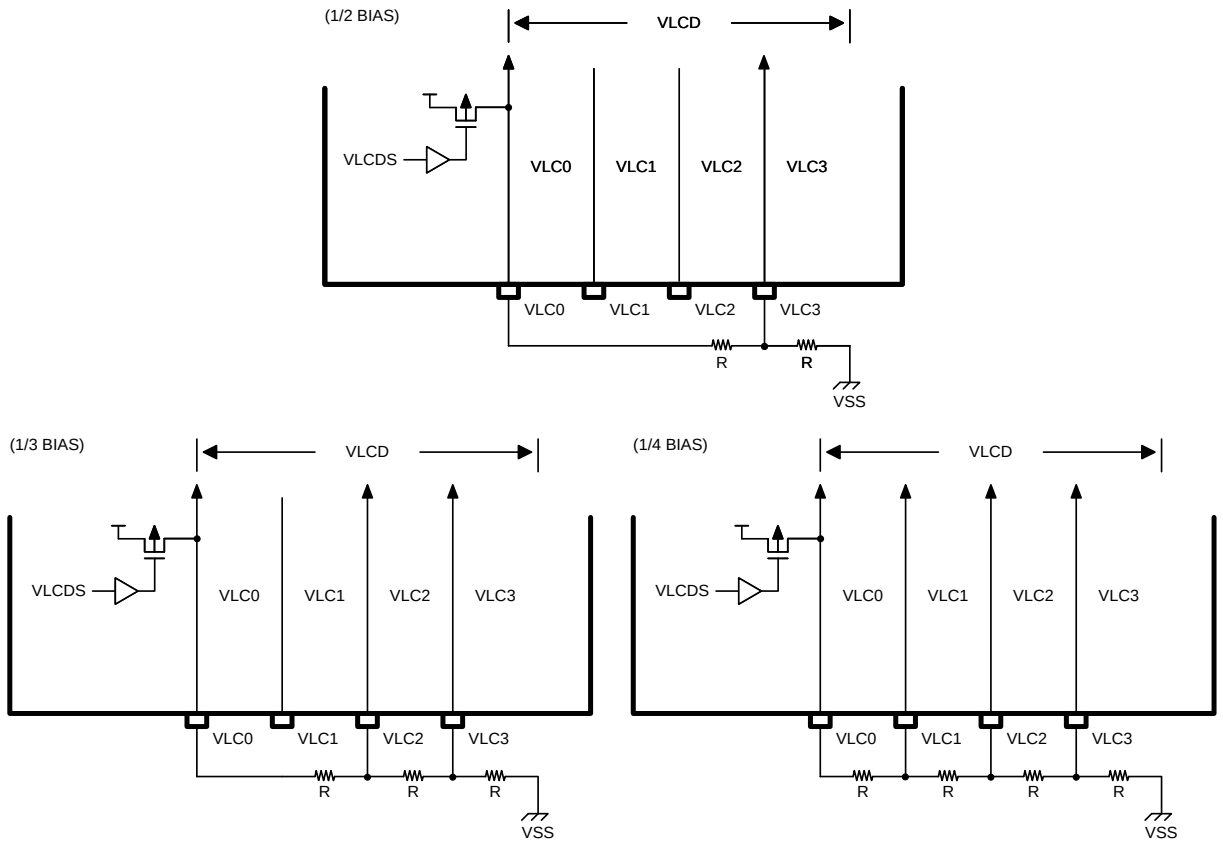
### 11.14.4 LCD Voltage Dividing Connection




NOTES)

1. When 1/2 bias are selected, the VLC0, VLC1 and VLC2 are internally connected.
2. When 1/3 bias are selected, the VLC0 and VLC1 are internally connected.
3. When the internal resistors are selected, all the P73/VLC0, P72/VLC1, P71/VLC2, P70/VLC3, P60/CAPH, and P61/CAPL pins can be used for normal I/O.

**Figure 11.90 Internal Resistor Bias Connection**




NOTES)

1. When the external resistor bias is selected, the internal resistors for bias are disconnected.
2. When the external resistor bias is selected, the dividing resistors should be connected like the above figure and the needed bias pins should be selected as the LCD bias function pins (VLC0, VLC1, VLC2, and VLC3) by P6FSRL/P7FSRL register.
   - When it is 1/2 bias, the P73/VLC0 and P70/VLC3 pins should be selected as VLC0 and VLC3 functions. The other pins can be used for normal I/O.
   - When it is 1/3 bias, the P73/VLC0, P71/VLC2, and P70/VLC3 pins should be selected as VLC0, VLC2, and VLC3 functions. The P72/VLC1 pin can be used for normal I/O.
   - When it is 1/4 bias, the P73/VLC0, P72/VLC1, P71/VLC2, and P70/VLC3 pins should be selected as VLC0, VLC1, VLC2, and VLC3 functions.

**Figure 11.91 External Resistor Bias Connection**

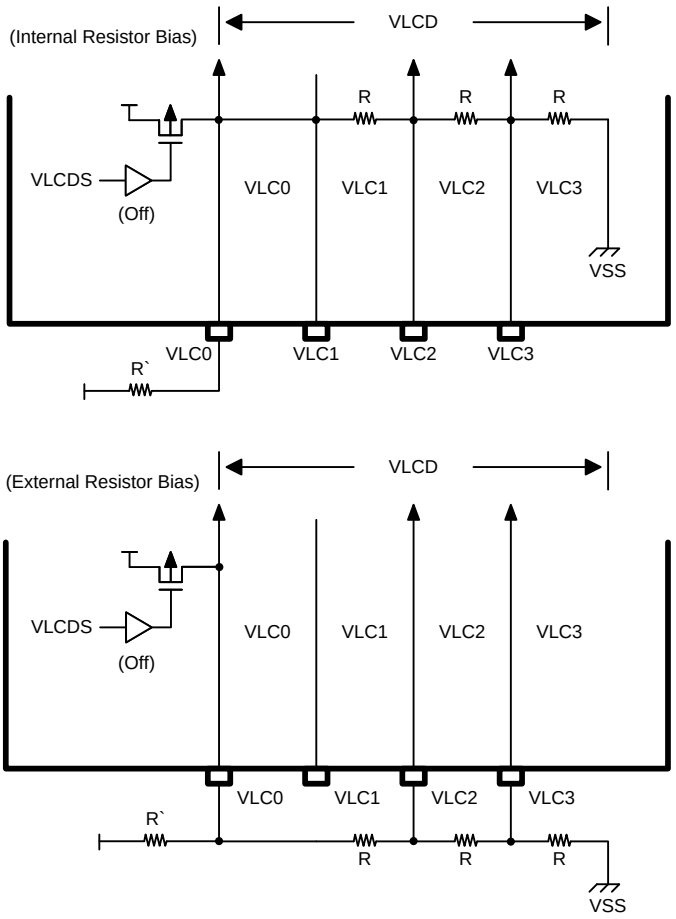
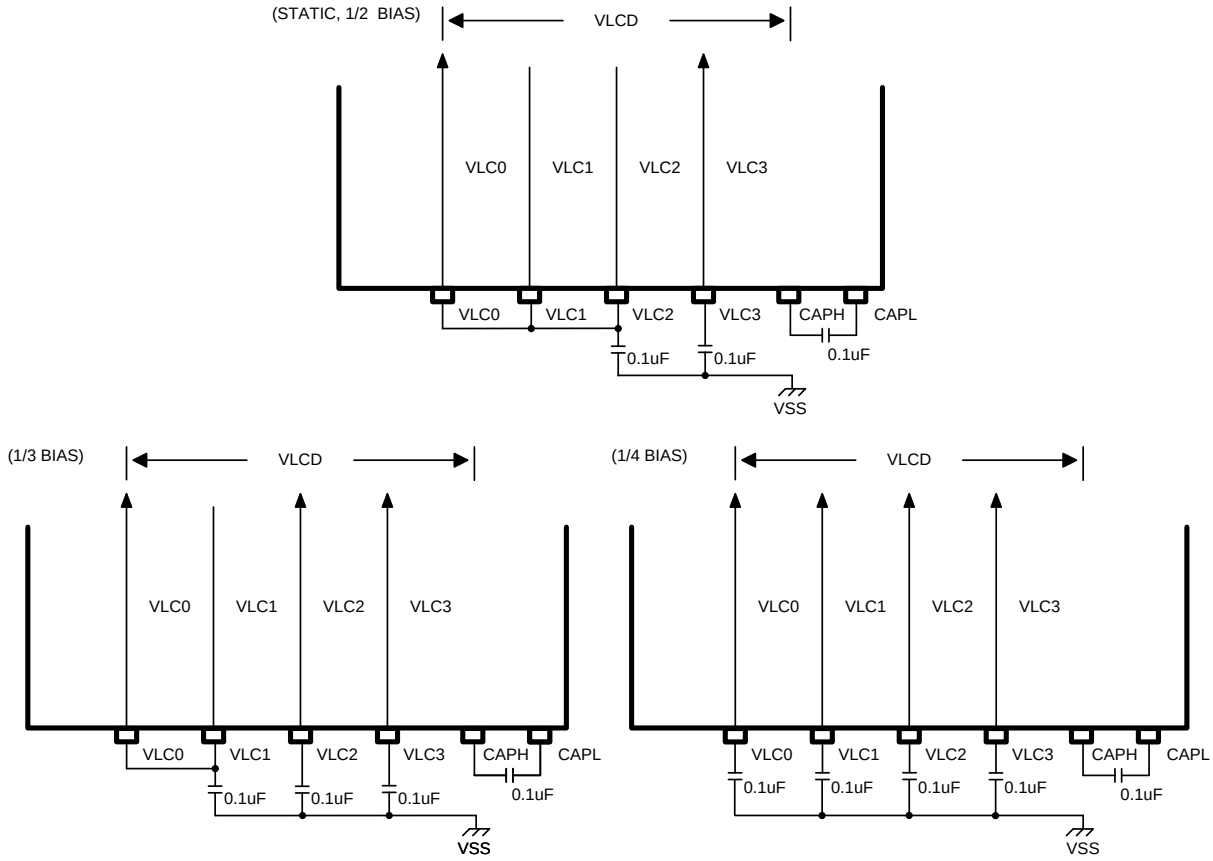



**Figure 11.92 Resistor Bias Connection for External VLCD**




NOTES) When the capacitor bias is selected for the VLCD voltage, all the P73/VLC0, P72/VLC1, P71/VLC2, P70/VLC3, P60/CAPH, and P61/CAPL pins should be selected for LCD bias function pins by the P6FSRL/P7FSRL and should be connected like the above figure.

**Figure 11.93 Capacitor Bias Connection**

### 11.14.5 Block Diagram

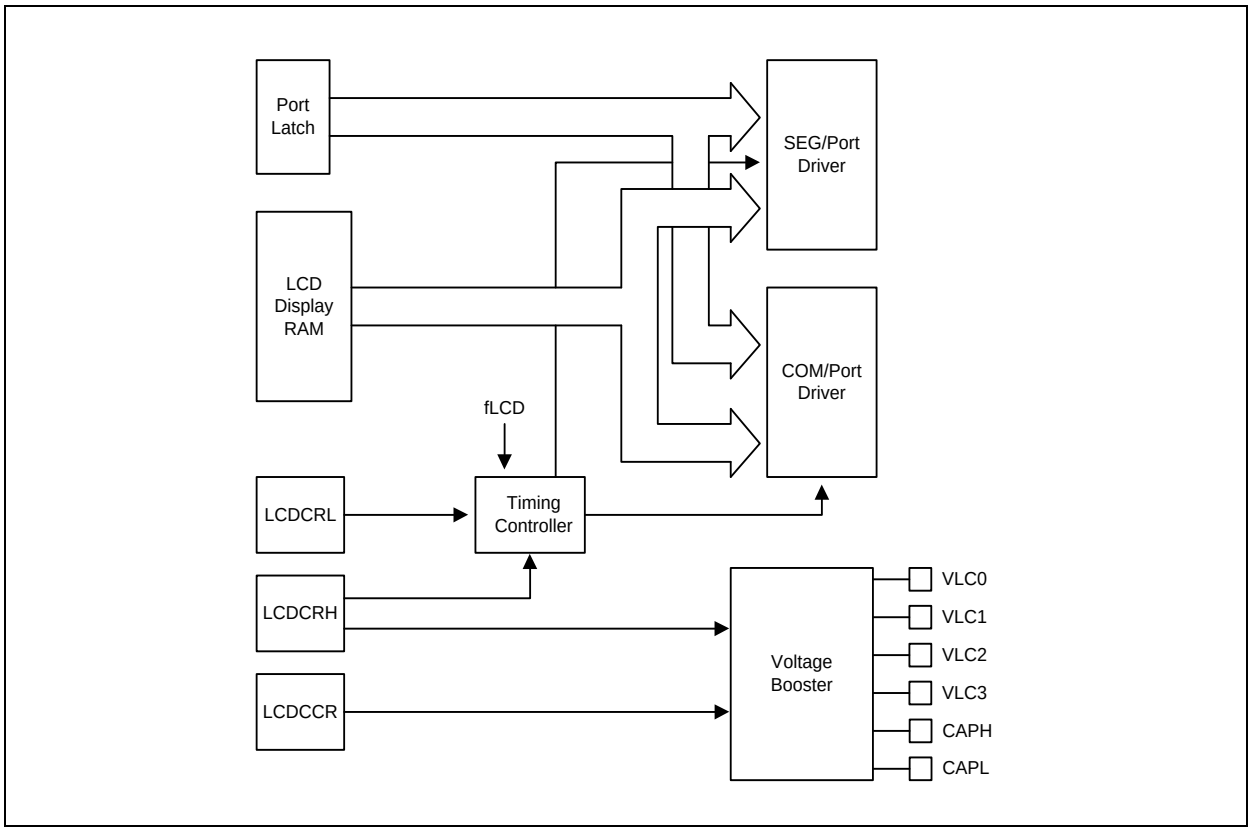



**Figure 11.94 LCD Circuit Block Diagram**

NOTE) The clock and duty for LCD driver is automatically initialized by hardware, whenever LCDCRL register data value is rewritten. So, don't rewrite LCDCRL frequently

### 11.14.6 Register Map

**Table 11-26 LCD Register Map**

| Name | Address | Dir | Default | Description |
|---|---|---|---|---|
| LCDCRH | F3H | R/W | 00H | LCD Driver Control High Register |
| LCDCRL | F2H | R/W | 00H | LCD Driver Control Low Register |
| LCDCCR | F1H | R/W | 00H | LCD Driver Contrast Control Register |

### 11.14.7 LCD Driver Register Description

LCD driver register has three control registers, LCD driver control high register (LCDCRH), LCD driver control low register (LCDCRL) and LCD driver contrast control register (LCDCCR).

## 11.14.8 Register Description for LCD Driver

**LCDCRH (LCD Driver Control High Register) : F3H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VLCDS | - | - | IRSEL1 | IRSEL0 | BTYPE1 | BTYPE0 | DISP |
| R/W | - | - | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**VLCDS** VLCD Power Select (only when a resistor bias type is selected)

0 Internal power for VLCD

1 External power for VLCD

**IRSEL[1:0]** Internal LCD Bias Dividing Resistor Select

| IRSEL1 | IRSEL0 | description |
|---|---|---|
| 0 | 0 | High resistance (RLCD: about 100kΩ) |
| 0 | 1 | Low resistance (RLCD: about 10kΩ) |
| 1 | 0 | Mid resistance (RLCD: about 50kΩ) |
| 1 | 1 | Not available |

**BTYPE[1:0]** LCD Duty and Bias Select (NOTE)

| BTYPE1 | BTYPE0 | Description |
|---|---|---|
| 0 | 0 | Internal resistor bias |
| 0 | 1 | External resistor bias |
| 1 | 0 | Capacitor bias (Voltage booster) |
| 1 | 1 | Not available |

NOTES)

1. All the VLC0 – VLC3, CAPH, and CAPL pins must be used as bias functions (P7FSRL = “55H”, P6FSRL.3-.0 = “0101b”) when the capacitor bias is selected for the LCD bias type.
2. Refer to the P6FSRL/P7FSRL register for pin functions

**DISP** LCD Display Control

0 Display off (The LCD block and voltage booster are off)

1 Normal display on (When the BTYPE[1:0] = “10b”, the voltage booster is turn on)

**LCDCRL (LCD Driver Control Low Register) : F2H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| - | - | - | DBS2 | DBS1 | DBS0 | LCLK1 | LCLK0 |
| - | - | - | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**DBS[2:0]** LCD Duty and Bias Select

| DBS2 | DBS1 | DBS0 | Description |
|---|---|---|---|
| 0 | 0 | 0 | 1/8Duty, 1/4Bias |
| 0 | 0 | 1 | 1/6Duty, 1/4Bias |
| 0 | 1 | 0 | 1/5Duty, 1/3Bias |
| 0 | 1 | 1 | 1/4Duty, 1/3Bias |
| 1 | 0 | 0 | 1/3Duty, 1/3Bias |
| 1 | 0 | 1 | 1/3Duty, 1/2Bias |
| 1 | 1 | 0 | 1/2Duty, 1/2Bias |
| 1 | 1 | 1 | Not available |

**LCLK[1:0]** LCD Clock Select (When $f_{WCK}$(Watch timer clock)= 32.768 kHz)

| LCLK1 | LCLK0 | Description |
|---|---|---|
| 0 | 0 | $f_{LCD}$ = 128Hz |
| 0 | 1 | $f_{LCD}$ = 256Hz |
| 1 | 0 | $f_{LCD}$ = 512Hz |
| 1 | 1 | $f_{LCD}$ = 1024Hz |

NOTE) The LCD clock is generated by watch timer clock ($f_{WCK}$). So the watch timer should be enabled when the LCD display is turned on.

**Table 11-27 LCD Frame Frequency**

| LCD Clock Frequency ($f_{LCD}$) | LCD Frame Frequency ($f_{FRAME}$) | | | | | | Unit |
|---|---|---|---|---|---|---|---|
| | 1/2 Duty | 1/3 Duty | 1/4 Duty | 1/5 Duty | 1/6 Duty | 1/8 Duty | |
| 128 | 64 | 43 | 32 | 26 | 21 | 16 | Hz |
| 256 | 128 | 85 | 64 | 51 | 43 | 32 | |
| 512 | 256 | 171 | 128 | 102 | 85 | 64 | |
| 1024 | 512 | 341 | 256 | 205 | 171 | 128 | |

The LCD frame frequency is calculated by the following formula:

$$\text{LCD Frame Frequency } (f_{FRAME}) = f_{LCD} \times \text{Duty[Hz]}$$

Ex) In cace of 1/4 duty and $f_{LCD}$ = 512Hz, $f_{FRAME} = f_{LCD} \times 1/4 = 512 \times 1/4 = 128$[Hz]

**LCDCCR (LCD Driver Contrast Control Register) : F1H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| - | - | - | - | VLCD3 | VLCD2 | VLCD1 | VLCD0 |
| - | - | - | - | R/W | R/W | R/W | R/W |

Initial value : 00H

**VLCD[3:0]** VLCD3 Voltage Control when the capacitor bias bias is selected

| VLCD3 | VLCD2 | VLCD1 | VLCD0 | Description | | |
|---|---|---|---|---|---|---|
| | | | | 1/2 | 1/3 | 1/4 Bias |
| 0 | 0 | 0 | 0 | VLC3= 1.00V, | 0.90V, | 0.75V |
| 0 | 0 | 0 | 1 | VLC3= 1.05V, | 0.95V, | 0.79V |
| 0 | 0 | 1 | 0 | VLC3= 1.10V, | 1.00V, | 0.83V |
| 0 | 0 | 1 | 1 | VLC3= 1.15V, | 1.05V, | 0.86V |
| 0 | 1 | 0 | 0 | VLC3= 1.20V, | 1.10V, | 0.90V |
| 0 | 1 | 0 | 1 | VLC3= 1.25V, | 1.15V, | 0.94V |
| 0 | 1 | 1 | 0 | VLC3= 1.30V, | 1.20V, | 0.98V |
| 0 | 1 | 1 | 1 | VLC3= 1.35V, | 1.25V, | 1.01V |
| 1 | 0 | 0 | 0 | VLC3= 1.40V, | 1.30V, | 1.05V |
| 1 | 0 | 0 | 1 | VLC3= 1.45V, | 1.35V, | 1.09V |
| 1 | 0 | 1 | 0 | VLC3= 1.50V, | 1.40V, | 1.13V |
| 1 | 0 | 1 | 1 | VLC3= 1.55V, | 1.45V, | 1.16V |
| 1 | 1 | 0 | 0 | VLC3= 1.60V, | 1.50V, | 1.20V |
| 1 | 1 | 0 | 1 | VLC3= 1.65V, | 1.55V, | 1.24V |
| 1 | 1 | 1 | 0 | VLC3= 1.70V, | 1.60V, | 1.28V |
| 1 | 1 | 1 | 1 | VLC3= 1.75V, | 1.65V, | 1.31V |

NOTES)
The VLC0 voltage can be calculated by the below formulas.
– VLC0 = VLC3 X 2 ; 1/2 bias
– VLC0 = VLC3 X 3 ; 1/3 bias
– VLC0 = VLC3 X 4 ; 1/4 bias

# 12. Power Down Operation

## 12.1 Overview

The MC96F7864 has two power-down modes to minimize the power consumption of the device. In power down mode, power consumption is reduced considerably. The device provides three kinds of power saving functions, Main-IDLE, Sub-IDLE and STOP mode. In three modes, program is stopped.

## 12.2 Peripheral Operation in IDLE/STOP Mode

**Table 12-1 Peripheral Operation during Power Down Mode**

| Peripheral | IDLE Mode | STOP Mode |
|---|---|---|
| CPU | ALL CPU Operation are Disable | ALL CPU Operation are Disable |
| RAM | Retain | Retain |
| Basic Interval Timer | Operates Continuously | Stop |
| Watch Dog Timer | Operates Continuously | Stop (Can be operated with WDTRC OSC) |
| Watch Timer | Operates Continuously | Stop (Can be operated with sub clock) |
| Timer0~8 | Operates Continuously | Halted (Only when the Event Counter Mode is Enabled, Timer operates Normally) |
| 10-bit PWM | Operates Continuously | Stop |
| ADC | Operates Continuously | Stop |
| BUZ | Operates Continuously | Stop |
| SPI2/3 | Operates Continuously | Only operate with external clock |
| UART2/3/4 | Operates Continuously | Only operate with external clock |
| USI0/1 | Operates Continuously | Only operate with external clock |
| LCD Controller | Operates Continuously | Stop (Can be operated with sub clock) |
| PLL | Oscillation | Stop when the system clock (fx) is $f_{PLL}$ |
| Internal OSC (16MHz) | Oscillation | Stop when the system clock (fx) is $f_{IRC}$ |
| WDTRC OSC (5kHz) | Stop | Can be operated with setting value |
| Main OSC (0.4~12MHz) | Oscillation | Stop when fx = $f_{XIN}$ |
| Sub OSC (32.768kHz) | Oscillation | Stop when fx = $f_{SUB}$ |
| I/O Port | Retain | Retain |
| Control Register | Retain | Retain |
| Address Data Bus | Retain | Retain |
| Release Method | By RESET, all Interrupts | By RESET, Timer Interrupt (EC0-EC7), SPI (External clock), External Interrupt, UART by Rx, WT (sub clock), WDT |

## 12.3 IDLE Mode

The power control register is set to '01h' to enter the IDLE Mode. In this mode, the internal oscillation circuits remain active. Oscillation continues and peripherals are operated normally but CPU stop. It is released by reset or interrupt. To be released by interrupt, interrupt should be enabled before IDLE mode. If using reset, because the device becomes initialized state, the registers have reset value.

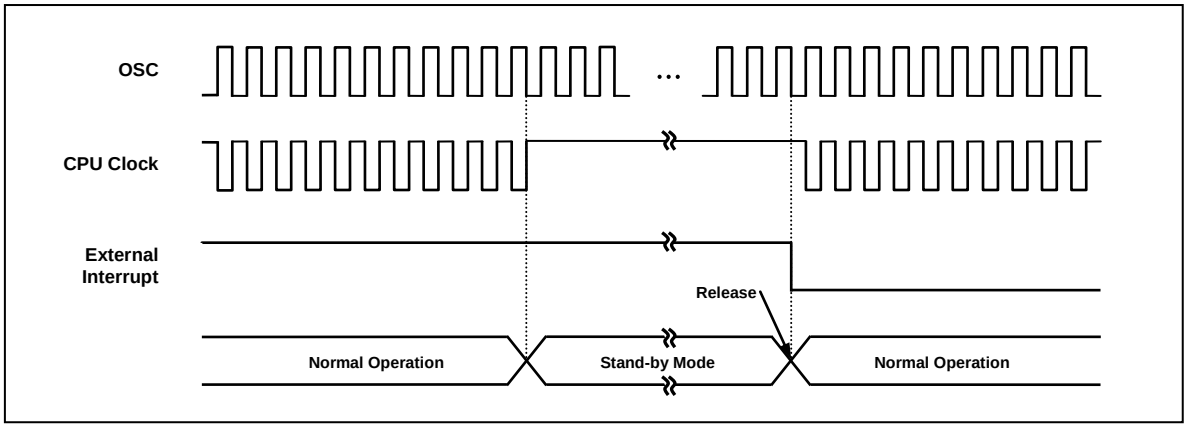



**Figure 12.1 IDLE Mode Release Timing by External Interrupt**

## 12.4 STOP Mode

The power control register is set to '03H' to enter the STOP Mode. In the stop mode, the selected oscillator, system clock and peripheral clock is stopped, but watch timer can be continued to operate with sub clock. With the clock frozen, all functions are stopped, but the on-chip RAM and control registers are held. For example, If the internal RC oscillator ($f_{IRC}$) is selected for the system clock and the sub clock ($f_{SUB}$) is oscillated, the internal RC oscillator STOP oscillation and the sub clock is continuously oscillated in stop mode. At that time, the watch timer and LCD controller can be operated with the sub clock.

The source for exit from STOP mode is hardware reset and interrupts. The reset re-defines all the control registers.

When exit from STOP mode, enough oscillation stabilization time is required to normal operation. Figure 12.2 shows the timing diagram. When released from STOP mode, the Basic interval timer is activated on wake-up. Therefore, before STOP instruction, user must be set its relevant prescale divide ratio to have long enough time. This guarantees that oscillator has started and stabilized.

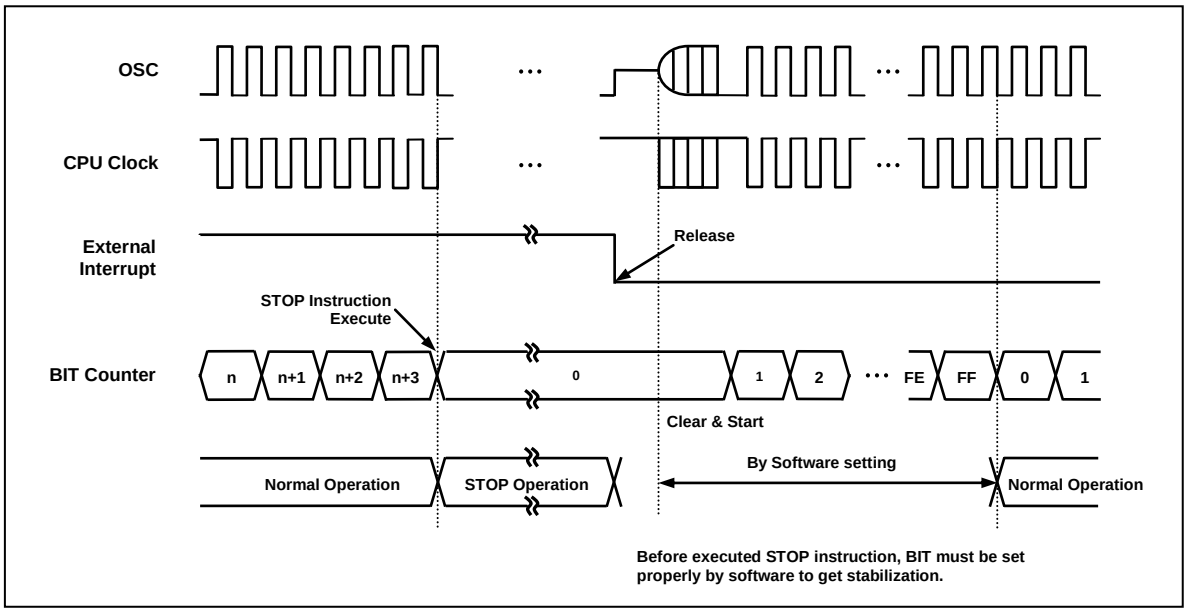



**Figure 12.2 STOP Mode Release Timing by External Interrupt**

## 12.5 Release Operation of STOP Mode

After STOP mode is released, the operation begins according to content of related interrupt register just before STOP mode start (Figure 12.3). If the global interrupt Enable Flag (IE.EA) is set to `1`, the STOP mode is released by the interrupt which each interrupt enable flag = `1` and the CPU jumps to the relevant interrupt service routine. Even if the IE.EA bit is cleared to '0', the STOP mode is released by the interrupt of which the interrupt enable flag is set to '1'.

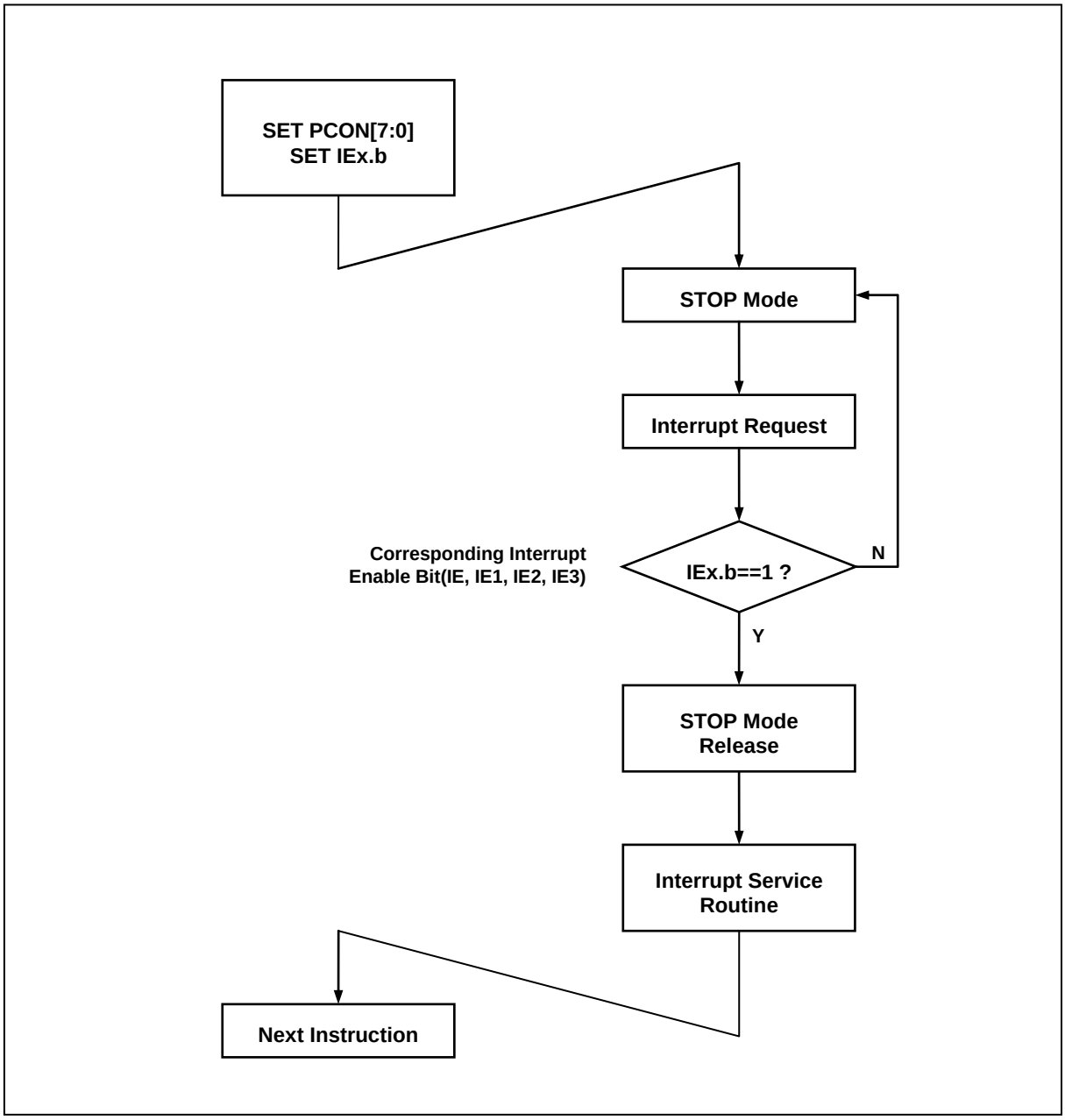



**Figure 12.3 STOP Mode Release Flow**

### 12.5.1 Register Map

**Table 12-2 Power Down Operation Register Map**

| Name | Address | Dir | Default | Description |
|---|---|---|---|---|
| PCON | 87H | R/W | 00H | Power Control Register |

### 12.5.2 Power Down Operation Register Description

The power down operation register consists of the power control register (PCON).

### 12.5.3 Register Description for Power Down Operation

**PCON (Power Control Register) : 87H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PCON7 | – | – | – | PCON3 | PCON2 | PCON1 | PCON0 |
| R/W | – | – | – | R/W | R/W | R/W | R/W |

Initial value : 00H

| | | |
|---|---|---|
| **PCON[7:0]** | Power Control | |
| | 01H | IDLE mode enable |
| | 03H | STOP mode enable |
| | Other Values | Normal operation |

NOTES) 1. To enter IDLE mode, PCON must be set to '01H'.

2. To enter STOP mode, PCON must be set to '03H'.
3. The PCON register is automatically cleared by a release signal in STOP/IDLE mode.
4. Three or more NOP instructions must immediately follow the instruction that make the device enter STOP/IDLE mode. Refer to the following examples.

```
Ex1)  MOV   PCON, #01H   ; IDLE mode
      NOP
      NOP
      NOP
       •
       •
       •
```

```
Ex2)  MOV   PCON, #03H   ; STOP mode
      NOP
      NOP
      NOP
       •
       •
       •
```

# 13. RESET

## 13.1 Overview

The following is the hardware setting value.

**Table 13-1 Reset State**

| On Chip Hardware | Initial Value |
|---|---|
| Program Counter (PC) | 0000h |
| Accumulator | 00h |
| Stack Pointer (SP) | 07h |
| Peripheral Clock | On |
| Control Register | Refer to the Peripheral Registers |

## 13.2 Reset Source

The MC96F7864 has five types of reset sources. The following is the reset sources.

- External RESETB
- Power ON RESET (POR)
- WDT Overflow Reset (In the case of WDTEN = `1`)
- Low Voltage Reset (In the case of LVREN = `0 `)
- Flash fail detection ( Parity Check )
- OCD Reset

## 13.3 RESET Block Diagram

`

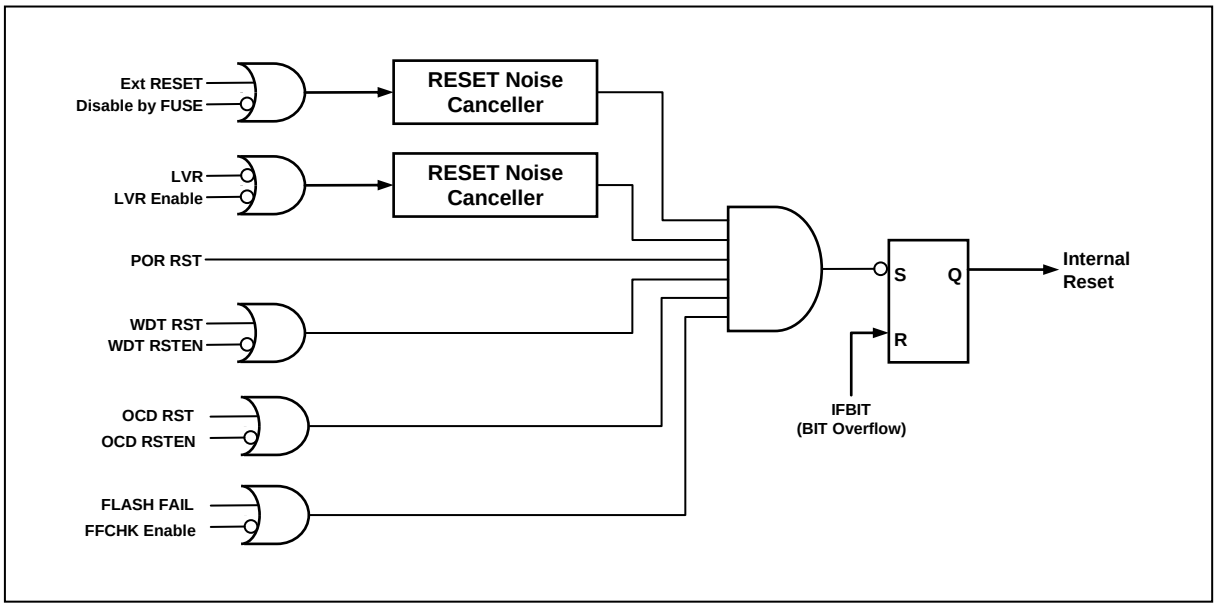



**Figure 13.1 RESET Block Diagram**

## 13.4 RESET Noise Canceller

The Figure 13.2 is the noise canceller diagram for noise cancellation of RESET. It has the noise cancellation value of about 2us (@$V_{DD}$=5V) to the low input of system reset.

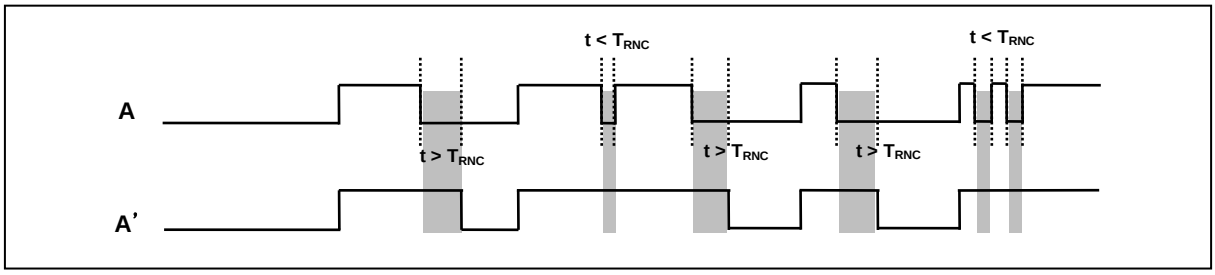



**Figure 13.2 Reset noise canceller timer diagram**

## 13.5 Power on RESET

When rising device power, the POR (Power On Reset) has a function to reset the device. If POR is used, it executes the device RESET function instead of the RESET IC or the RESET circuits.

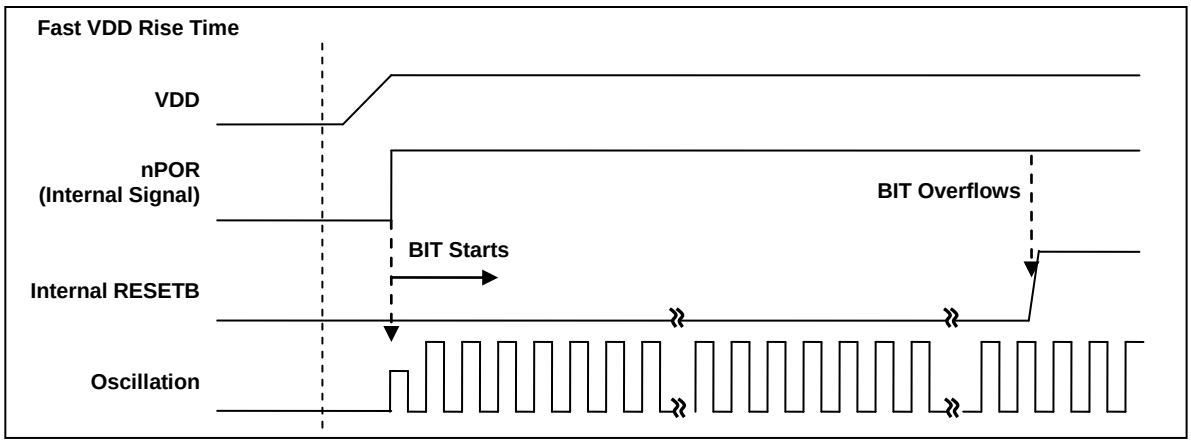



**Figure 13.3 Fast VDD Rising Time**

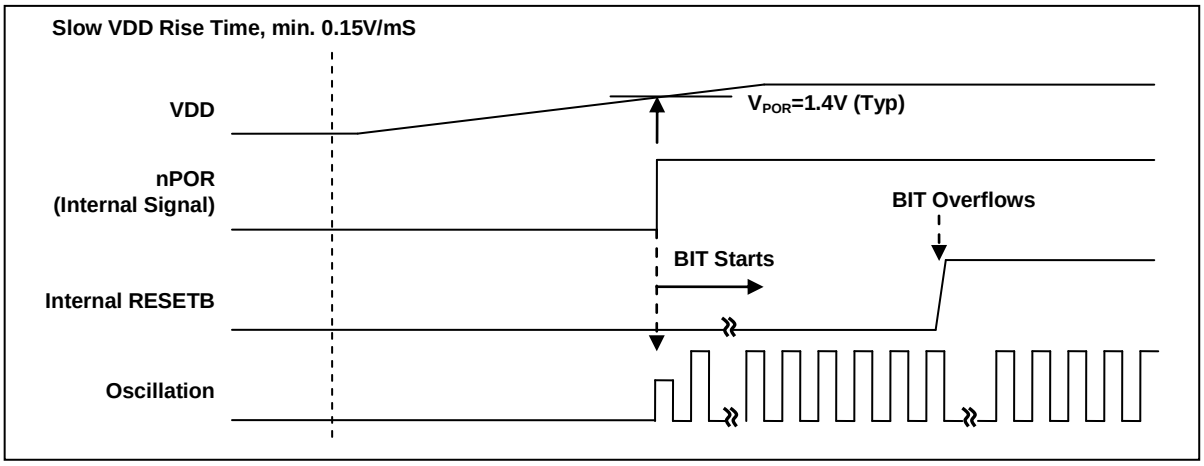



**Figure 13.4 Internal RESET Release Timing On Power-Up**

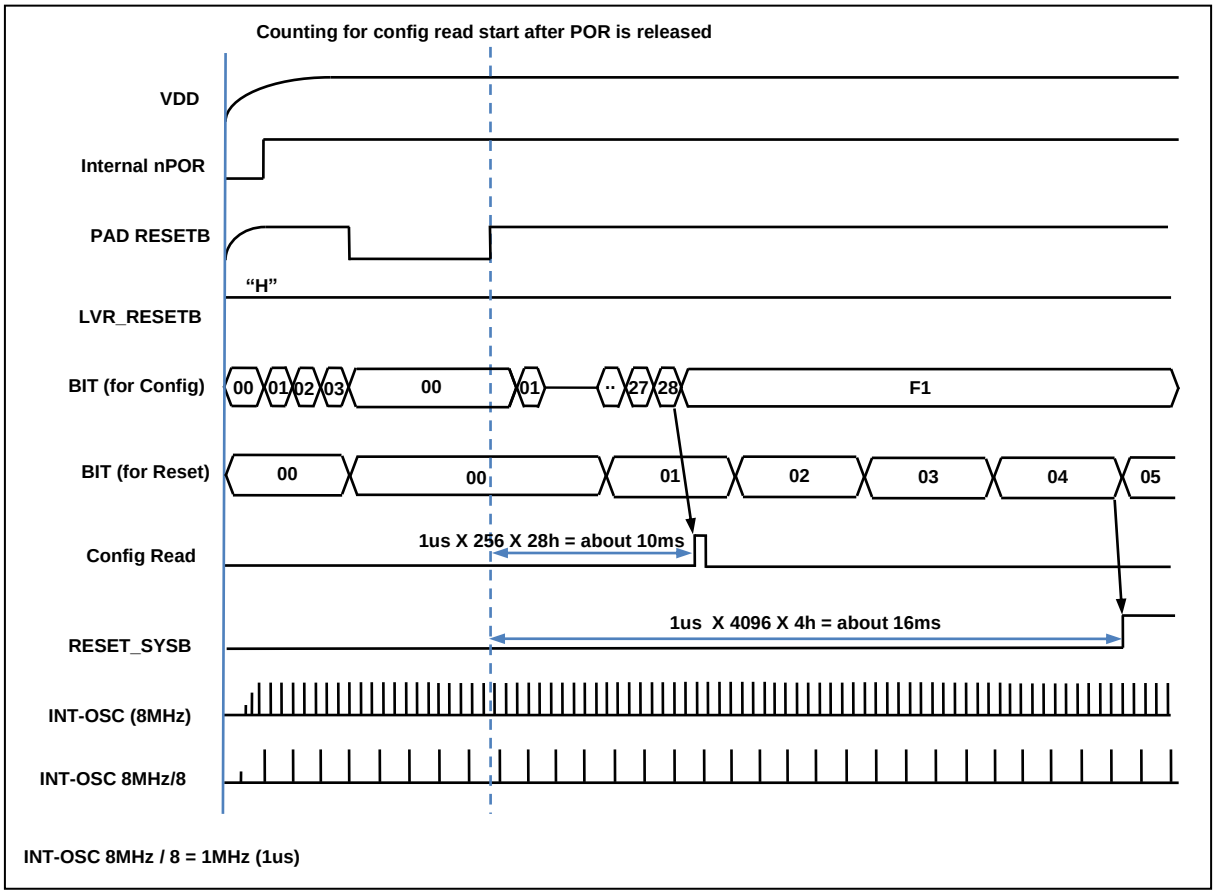



**Figure 13.5 Configuration Timing when Power-on**

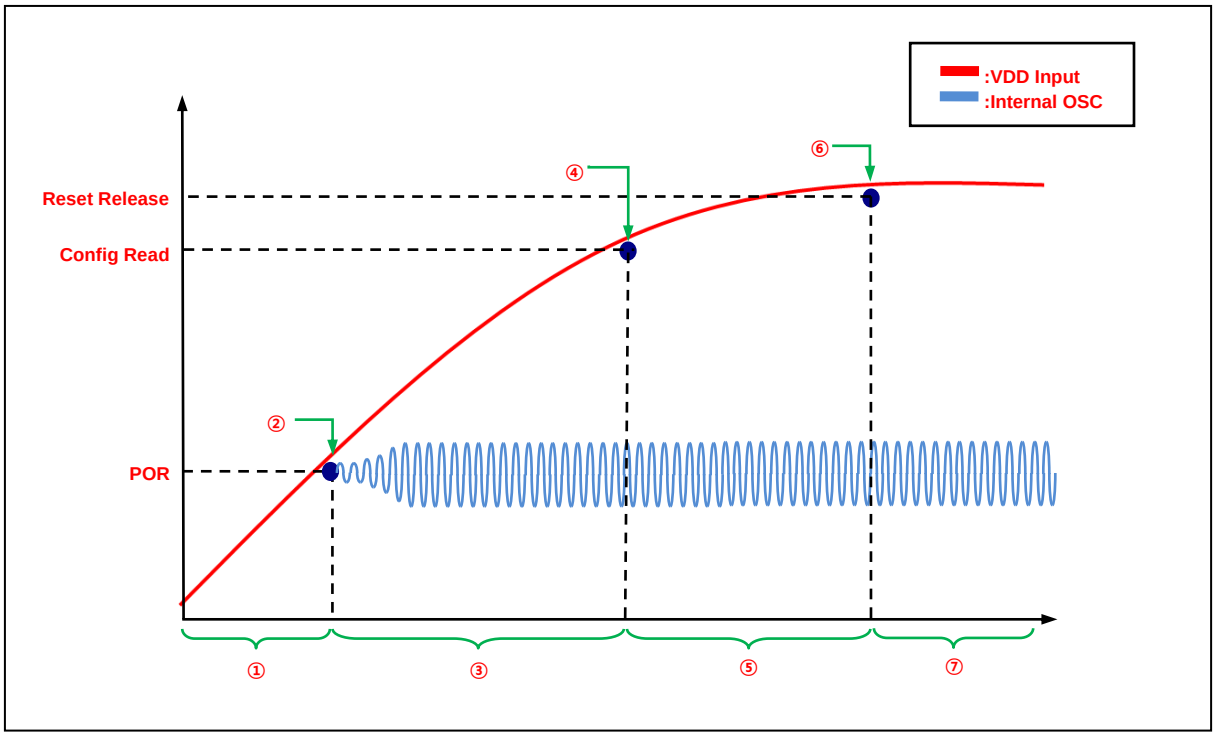



**Figure 13.6 Boot Process WaveForm**

**Table 13-2 Boot Process Description**

| Process | Description | Remarks |
|---|---|---|
| ① | -No Operation | |
| ② | -1st POR level Detection | -about 1.4V |
| ③ | - (INT-OSC 8MHz/8)x256x28h Delay section (=10ms)<br>-VDD input voltage must rise over than flash operating voltage for Config read | -Slew Rate >= 0.15V/ms |
| ④ | - Config read point | -about 1.5V ~ 1.6V<br>-Config Value is determined by Writing Option |
| ⑤ | - Rising section to Reset Release Level | -16ms point after POR or Ext_reset release |
| ⑥ | - Reset Release section (BIT overflow)<br>i) after16ms, after External Reset Release (External reset)<br>ii) 16ms point after POR (POR only) | - BIT is used for Peripheral stability |
| ⑦ | -Normal operation | |

## 13.6 External RESETB Input

The External RESETB is the input to a Schmitt trigger. If RESETB pin is held with low for at least 10us over within the operating voltage range and stable oscillation, it is applied and the internal state is initialized. After reset state becomes '1', it needs the stabilization time with 16ms and after the stable state, the internal RESET becomes '1'. The Reset process step needs 5 oscillator clocks. And the program execution starts at the vector address stored at address 0000H.

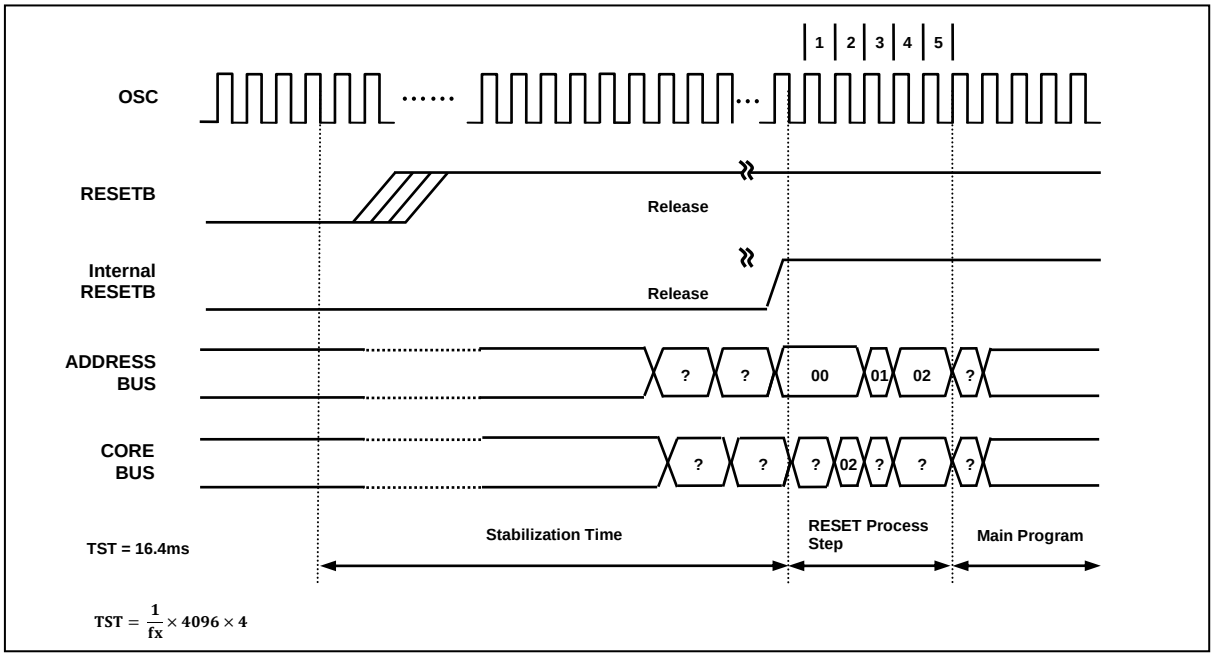



**Figure 13.7 Timing Diagram after RESET**

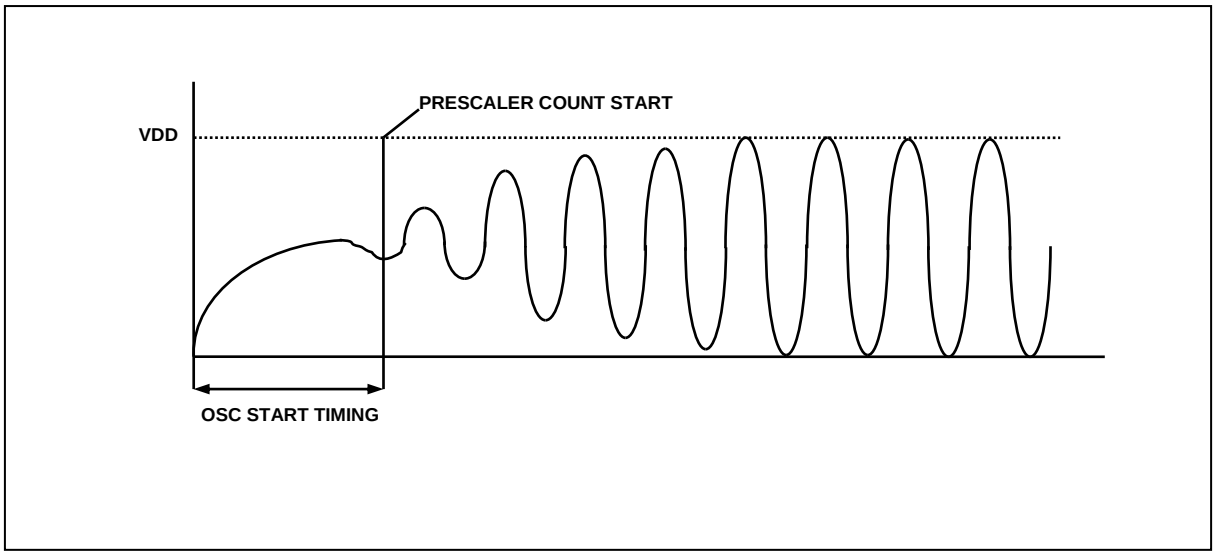



**Figure 13.8 Oscillator generating waveform example**

NOTE) As shown Figure 13.8, the stable generating time is not included in the start-up time.
The RESETB pin has a Pull-up resistor by H/W.

## 13.7 Brown Out Detector Processor

The MC96F7864 has an On-chip brown-out detection circuit (BOD) for monitoring the VDD level during operation by comparing it to a fixed trigger level. The trigger level for the BOD can be selected by LVRVS[3:0] bit to be 1.60V, 2.00V, 2.10V, 2.20V,2.32V, 2.44V, 2.59V, 2.75V, 2.93V, 3.14V, 3.38V, 3.67V, 4.00V, 4.40V. In the STOP mode, this will contribute significantly to the total current consumption. So to minimize the current consumption, the LVREN bit is set to off by software.

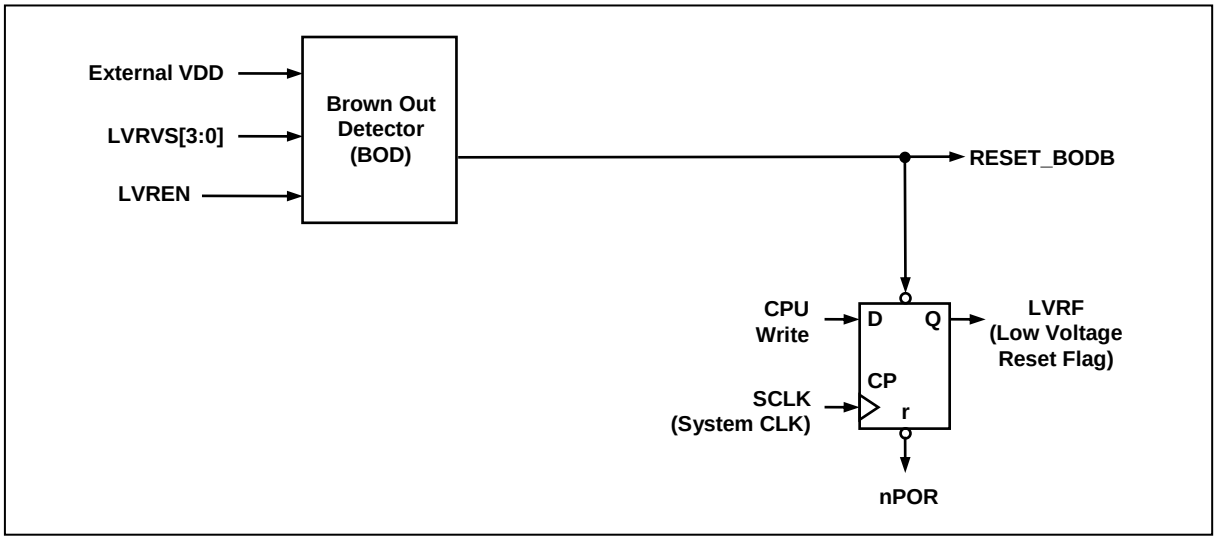



**Figure 13.9 Block Diagram of BOD**

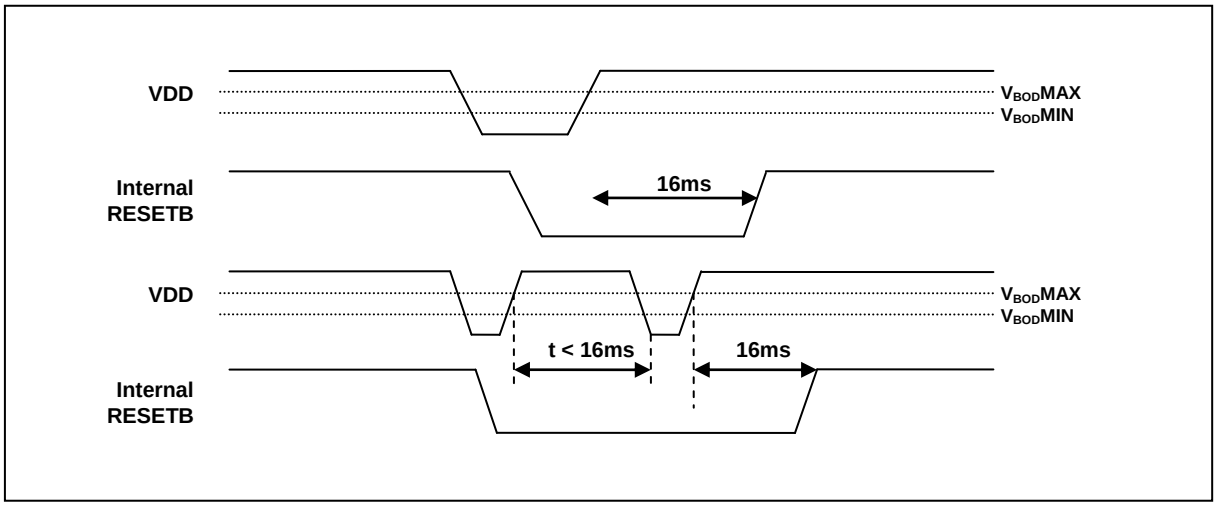



**Figure 13.10 Internal Reset at the power fail situation**

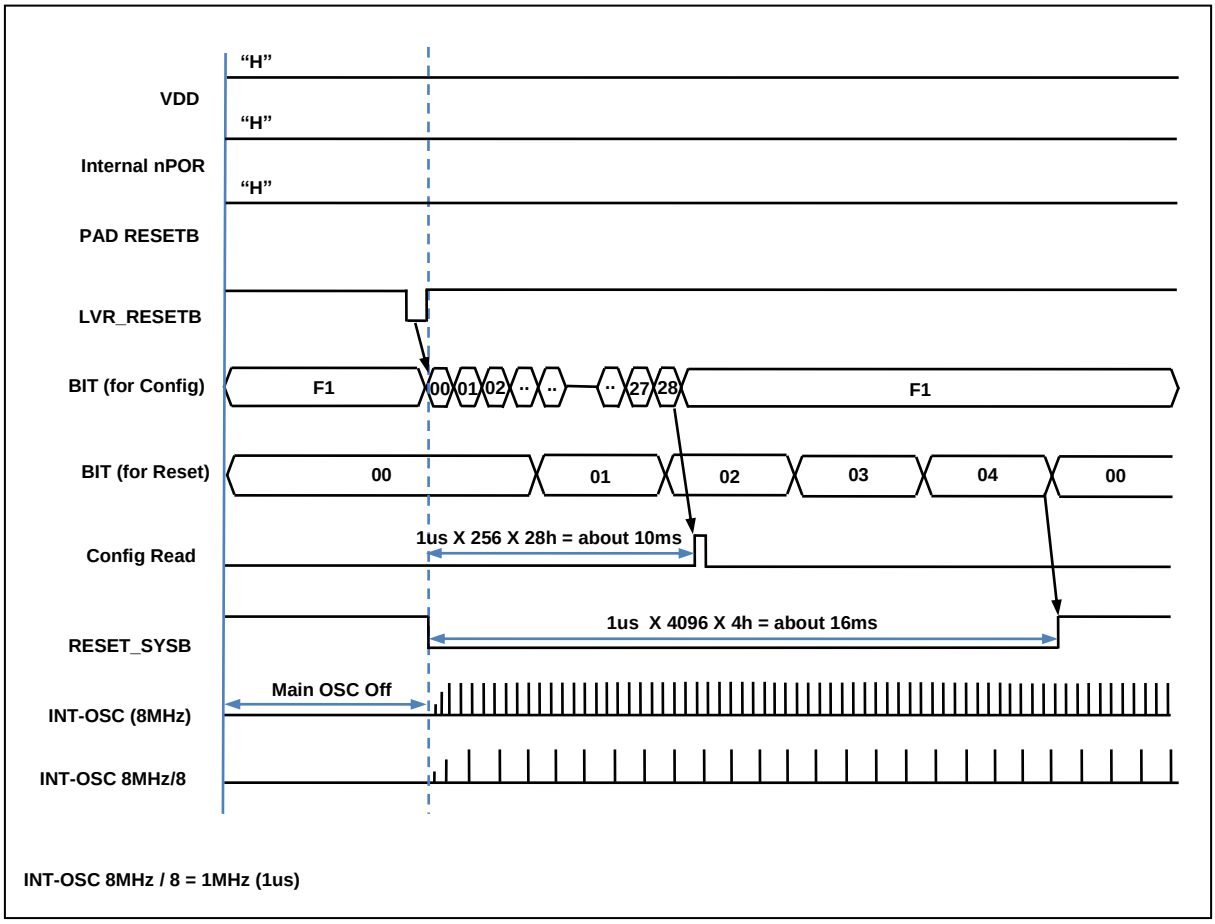



**Figure 13.11 Configuration timing when BOD RESET**

## 13.8 LVI Block Diagram

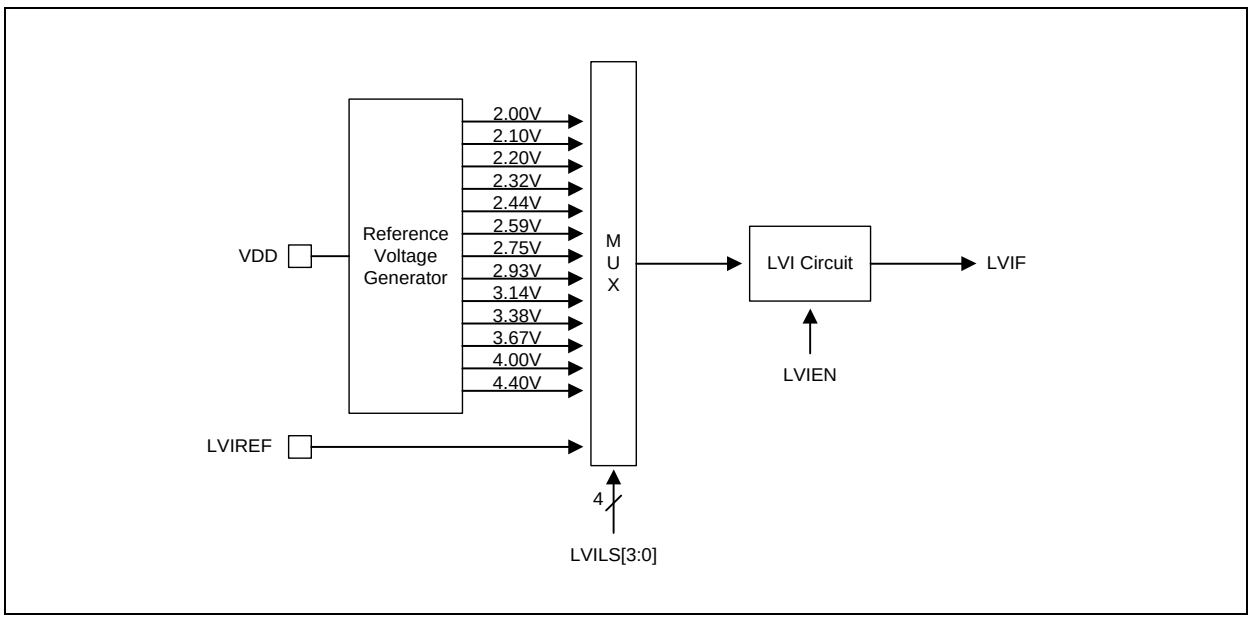



**Figure 13.12 LVI Diagram**

### 13.8.1 Register Map

**Table 13-3 Reset Operation Register Map**

| Name | Address | Dir | Default | Description |
|---|---|---|---|---|
| RSTFR | E8H | R/W | 80H | Reset Flag Register |
| LVRCR | D8H | R/W | 00H | Low Voltage Reset Control Register |
| LVICR | 86H | R/W | 00H | Low Voltage Indicator Control Register |

### 13.8.2 Reset Operation Register Description

The reset control register consists of the reset flag register (RSTFR), low voltage reset control register (LVRCR), and low voltage indicator control register (LVICR).

### 13.8.3 Register Description for Reset Operation

**RSTFR (Reset Flag Register) : E8H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PORF | EXTRF | WDTRF | OCDRF | LVRF | – | – | FPRIRF |
| R/W | R/W | R/W | R/W | R/W | – | – | R/W |

Initial value : 80H

**PORF** Power-On Reset flag bit. The bit is reset by writing '0' to this bit.

0 No detection

1 Detection

**EXTRF** External Reset (RESETB) flag bit. The bit is reset by writing '0' to this bit or by Power-On Reset.

0 No detection

1 Detection

**WDTRF** Watch Dog Reset flag bit. The bit is reset by writing '0' to this bit or by Power-On Reset.

0 No detection

1 Detection

**OCDRF** On-Chip Debug Reset flag bit. The bit is reset by writing '0' to this bit or by Power-On Reset.

0 No detection

1 Detection

**LVRF** Low Voltage Reset flag bit. The bit is reset by writing '0' to this bit or by Power-On Reset.

0 No detection

1 Detection

**FPRIRF** Flash ROM Parity Fail Reset flag bit. This bit is set to '1' by a reset of flash ROM parity fail detection during the CPU fetches the next flash ROM data. This bit is reset by writing '0' to this bit or by Power-On Reset.

0 No detection

1 Detection

NOTES) 1. When the Power-On Reset occurs, the PORF bit is only set to "1", the other flag (WDTRF and OCDRF) bits are all cleared to "0".

2. When the Power-On Reset occurs, the EXTRF bit is unknown, At that time, the EXTRF bit can be set to "1" when External Reset (RESETB) occurs.

3. When the Power-On Reset occurs, the LVRF bit is unknown, At that time, the LVRF bit can be set to "1" when LVR Reset occurs.

4. When a reset except the POR occurs, the corresponding flag bit is only set to "1", the other flag bits are kept in the previous values.

**LVRCR (Low Voltage Reset Control Register) : D8H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| LVRST | – | – | LVRVS3 | LVRVS2 | LVRVS1 | LVRVS0 | LVREN |
| R/W | – | – | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**LVRST** LVR Enable when Stop Release

0 Not effect at stop release

1 LVR enable at stop release

NOTES)

When this bit is '1', the LVREN bit is cleared to '0' by stop mode release. (LVR enable)

When this bit is '0', the LVREN bit is not effect by stop mode release.

**LVRVS[3:0]** LVR Voltage Select

| LVRVS3 | LVRVS2 | LVRVS1 | LVRVS0 | Description |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1.60V |
| 0 | 0 | 0 | 1 | 2.00V |
| 0 | 0 | 1 | 0 | 2.10V |
| 0 | 0 | 1 | 1 | 2.20V |
| 0 | 1 | 0 | 0 | 2.32V |
| 0 | 1 | 0 | 1 | 2.44V |
| 0 | 1 | 1 | 0 | 2.59V |
| 0 | 1 | 1 | 1 | 2.75V |
| 1 | 0 | 0 | 0 | 2.93V |
| 1 | 0 | 0 | 1 | 3.14V |
| 1 | 0 | 1 | 0 | 3.38V |
| 1 | 0 | 1 | 1 | 3.67V |
| 1 | 1 | 0 | 0 | 4.00V |
| 1 | 1 | 0 | 1 | 4.40V |
| 1 | 1 | 1 | 0 | Not available |
| 1 | 1 | 1 | 1 | Not available |

**LVREN** LVR Operation

0 LVR Enable

1 LVR Disable

NOTES) 1. The LVRVS[3:0] and LVREN bits are cleared by a power-on reset but are retained by other reset.

2. The LVRVS[3:0] bits should be set to '0000b' while LVREN bit is "1".

**LVICR (Low Voltage Indicator Control Register) : 86H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | LVIF | LVIEN | LVILS3 | LVILS2 | LVILS1 | LVILS0 |
| – | – | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**LVIF** Low Voltage Indicator Flag Bit

0 No detection

1 Detection

**LVIEN** LVI Enable/Disable

0 Disable

1 Enable

**LVILS[3:0]** LVI Level Select

| LVILS3 | LVILS2 | LVILS1 | LVILS0 | Description |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 2.00V |
| 0 | 0 | 0 | 1 | 2.10V |
| 0 | 0 | 1 | 0 | 2.20V |
| 0 | 0 | 1 | 1 | 2.32V |
| 0 | 1 | 0 | 0 | 2.44V |
| 0 | 1 | 0 | 1 | 2.59V |
| 0 | 1 | 1 | 0 | 2.75V |
| 0 | 1 | 1 | 1 | 2.93V |
| 1 | 0 | 0 | 0 | 3.14V |
| 1 | 0 | 0 | 1 | 3.38V |
| 1 | 0 | 1 | 0 | 3.67V |
| 1 | 0 | 1 | 1 | 4.00V |
| 1 | 1 | 0 | 0 | 4.40V |
| 1 | 1 | 0 | 1 | External Reference (LVIREF) |
| Other Values | | | | Not available |

# 14. On-chip Debug System

## 14.1 Overview

### 14.1.1 Description

On-chip debug system (OCD) of MC96F7864 can be used for programming the non-volatile memories and on-chip debugging. Detail descriptions for programming via the OCD interface can be found in the following chapter.

Figure 14.1 shows a block diagram of the OCD interface and the On-chip Debug system.

### 14.1.2 Feature

- Two-wire external interface: 1-wire serial clock input, 1-wire bi-directional serial data bus
- Debugger Access to:
  - All Internal Peripheral Units
  - Internal data RAM
  - Program Counter
  - Flash and Data EEPROM Memories
- Extensive On-chip Debug Support for Break Conditions, Including
  - Break Instruction
  - Single Step Break
  - Program Memory Break Points on Single Address
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the two-wire Interface
  - On-chip Debugging Supported by Dr.Choice®
- Operating frequency

  Supports the maximum frequency of the target MCU

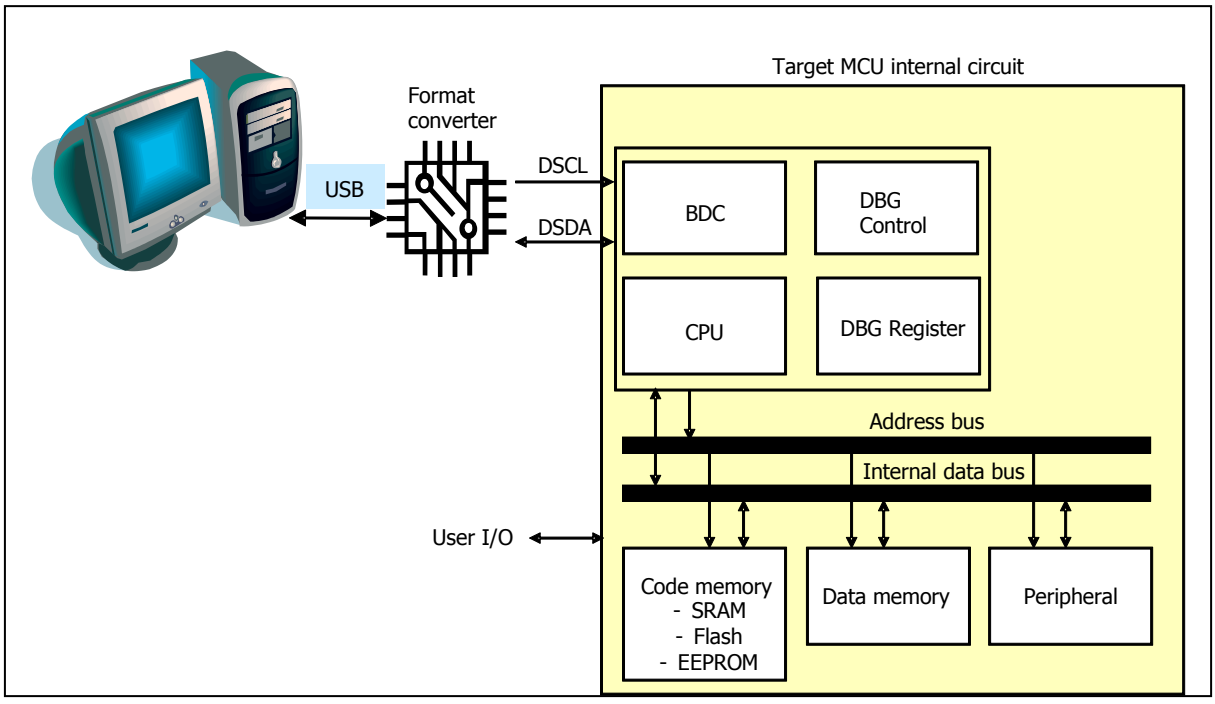



**Figure 14.1 Block Diagram of On-Chip Debug System**

## 14.2 Two-Pin External Interface

### 14.2.1 Basic Transmission Packet

- 10-bit packet transmission using two-pin interface.
- 1-packet consists of 8-bit data, 1-bit parity and 1-bit acknowledge.
- Parity is even of '1' for 8-bit data in transmitter.
- Receiver generates acknowledge bit as '0' when transmission for 8-bit data and its parity has no error.
- When transmitter has no acknowledge (Acknowledge bit is '1' at tenth clock), error process is executed in transmitter.
- When acknowledge error is generated, host PC makes stop condition and transmits command which has error again.
- Background debugger command is composed of a bundle of packet.
- Start condition and stop condition notify the start and the stop of background debugger command respectively.

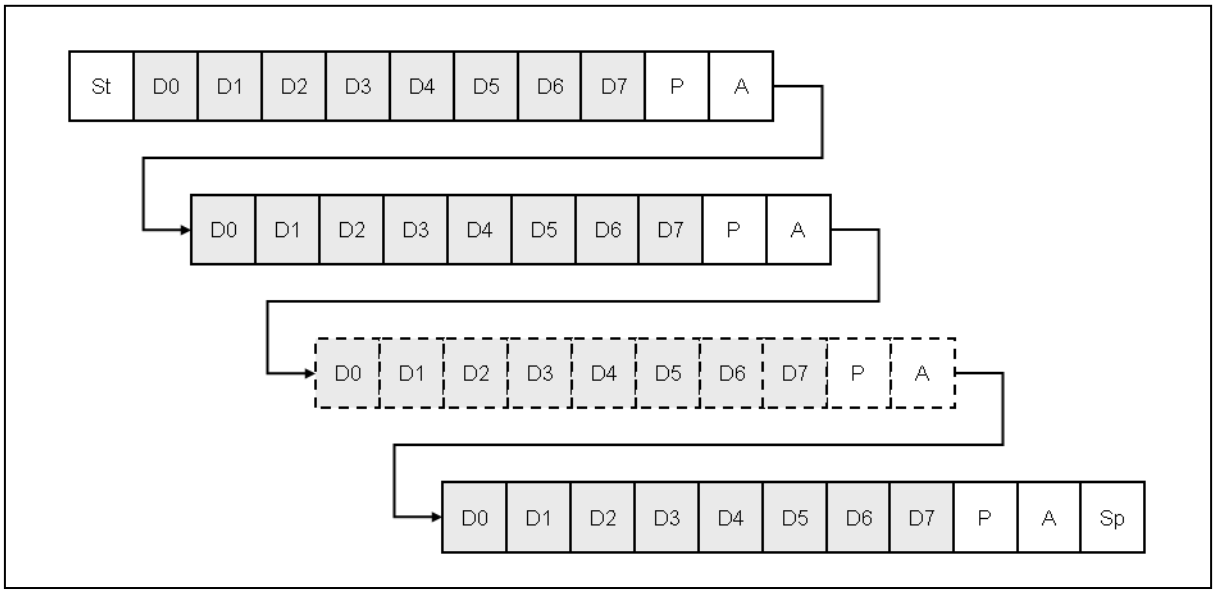



**Figure 14.2 10-bit Transmission Packet**

### 14.2.2 Packet Transmission Timing

#### 14.2.2.1 Data Transfer

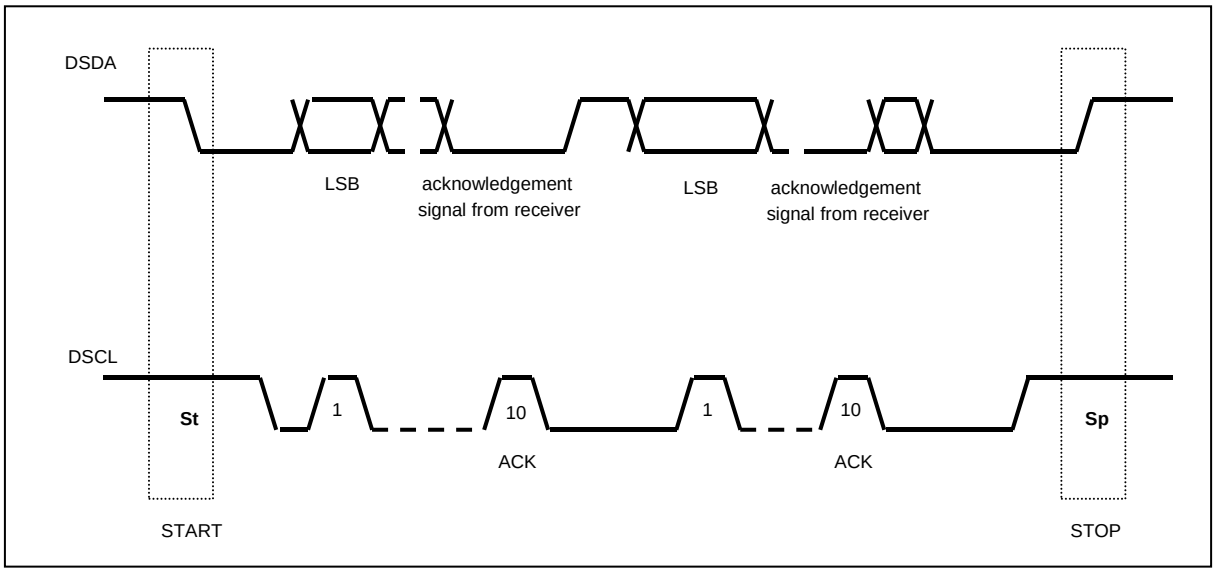



**Figure 14.3 Data Transfer on the Twin Bus**

#### 14.2.2.2 Bit Transfer

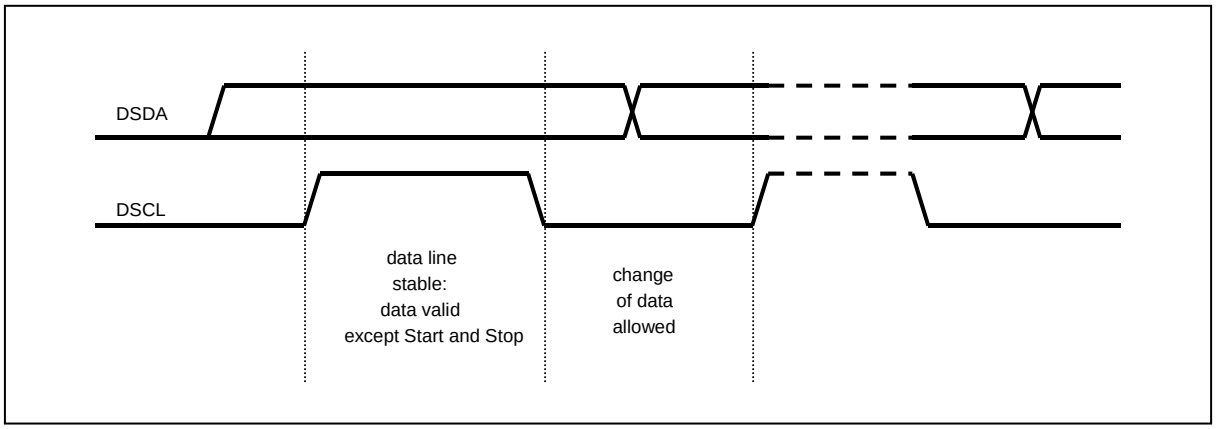



**Figure 14.4 Bit Transfer on the Serial Bus**

### 14.2.2.3 Start and Stop Condition

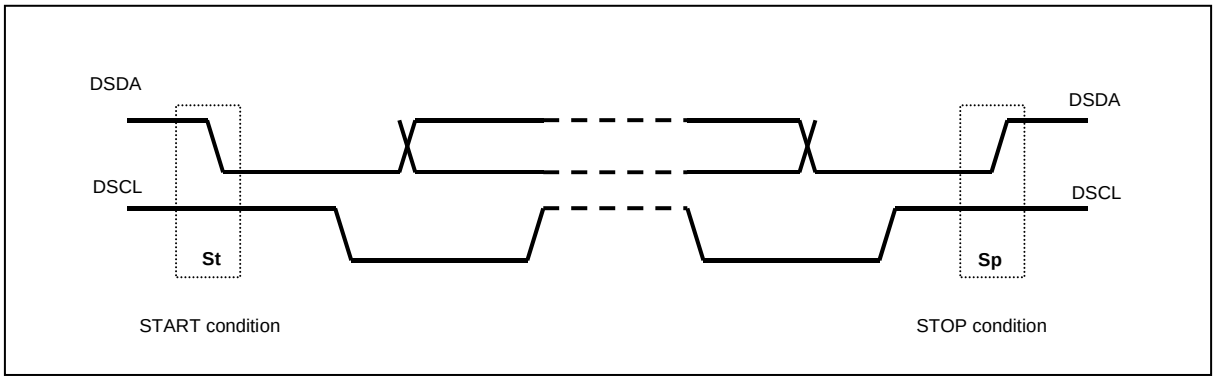



**Figure 14.5 Start and Stop Condition**

### 14.2.2.4 Acknowledge Bit

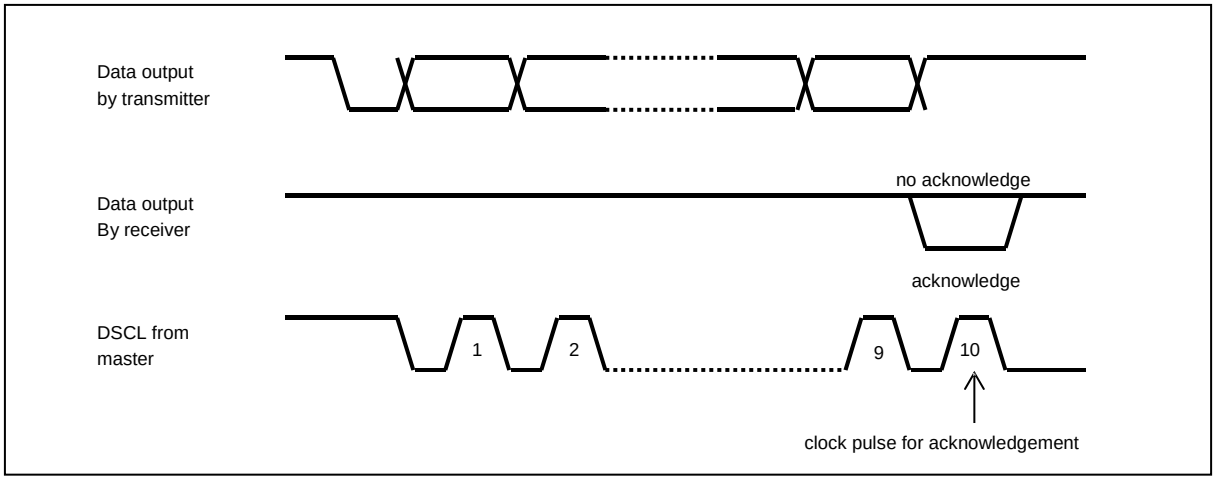



**Figure 14.6 Acknowledge on the Serial Bus**

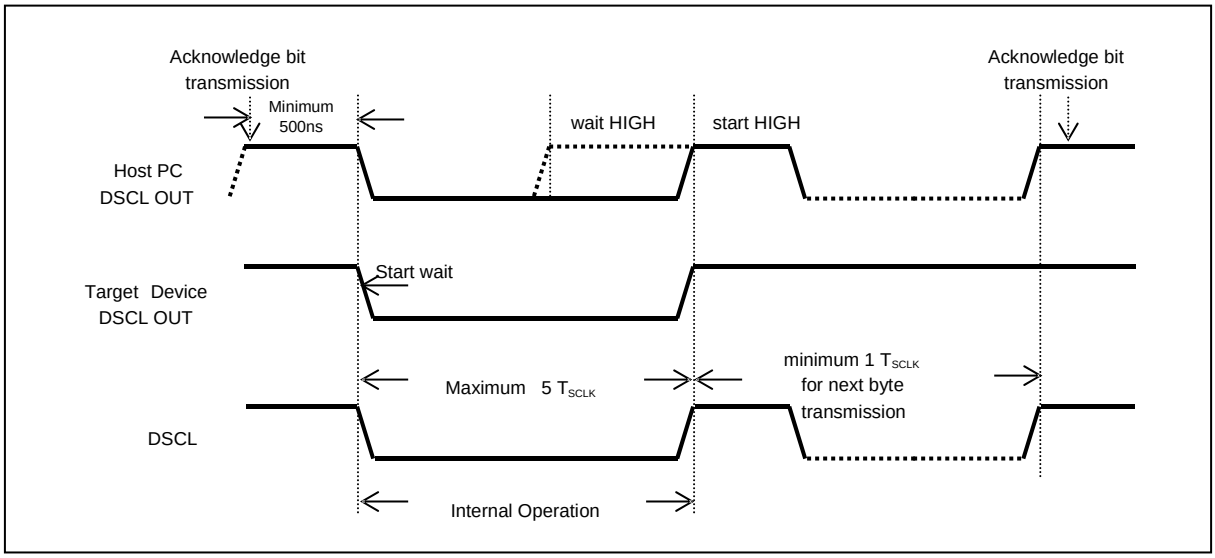



**Figure 14.7 Clock Synchronization during Wait Procedure**

### 14.2.3 Connection of Transmission

Two-pin interface connection uses open-drain (wire-AND bidirectional I/O).

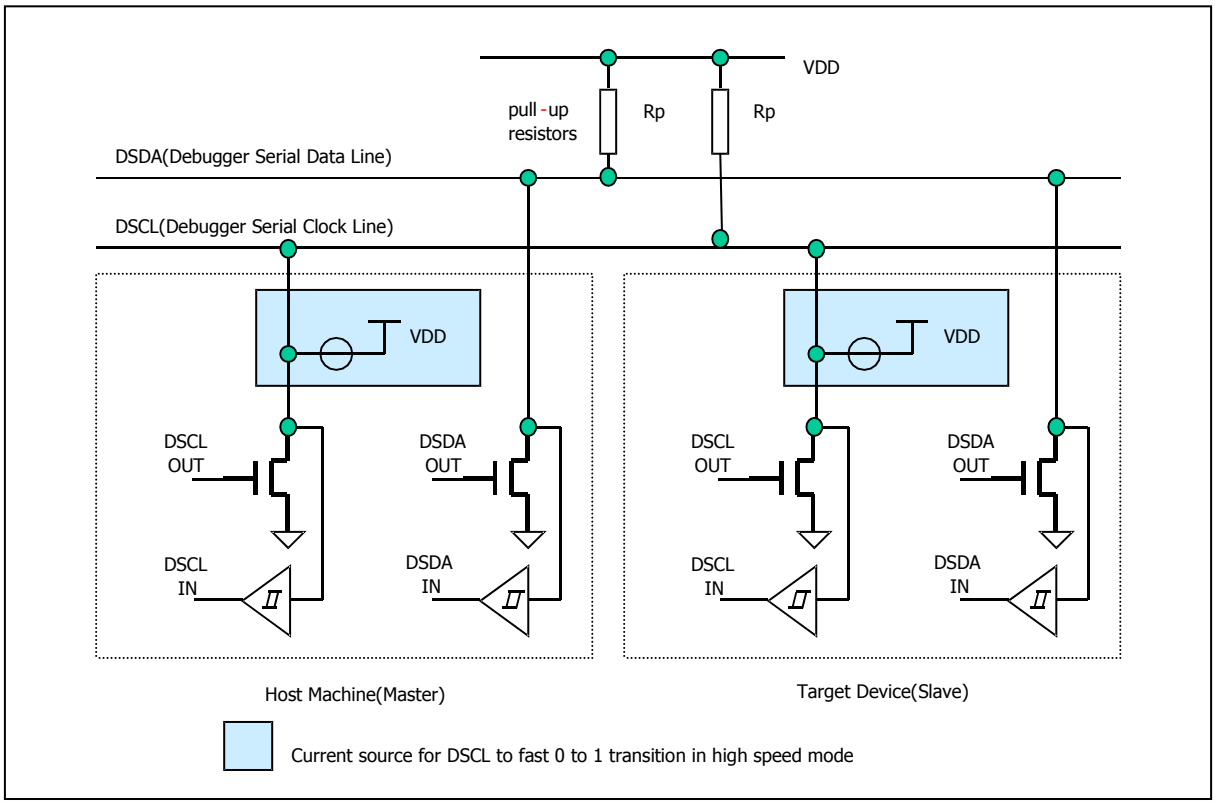



**Figure 14.8 Connection of Transmission**

# 15. Flash Memory

## 15.1 Overview

### 15.1.1 Description

MC96F7864 incorporates flash memory to which a program can be written, erased, and overwritten while mounted on the board. The flash memory can be read by 'MOVC' instruction and it can be programmed in OCD, serial ISP mode or user program mode.

- Flash Size : 64kbytes
- Single power supply program and erase
- Command interface for fast program and erase operation
- Up to 10,000(Sector 0~1019)/100,000(Sector 1020~1023) program/erase cycles at typical voltage and temperature for flash memory

### 15.1.2 Flash Program ROM Structure

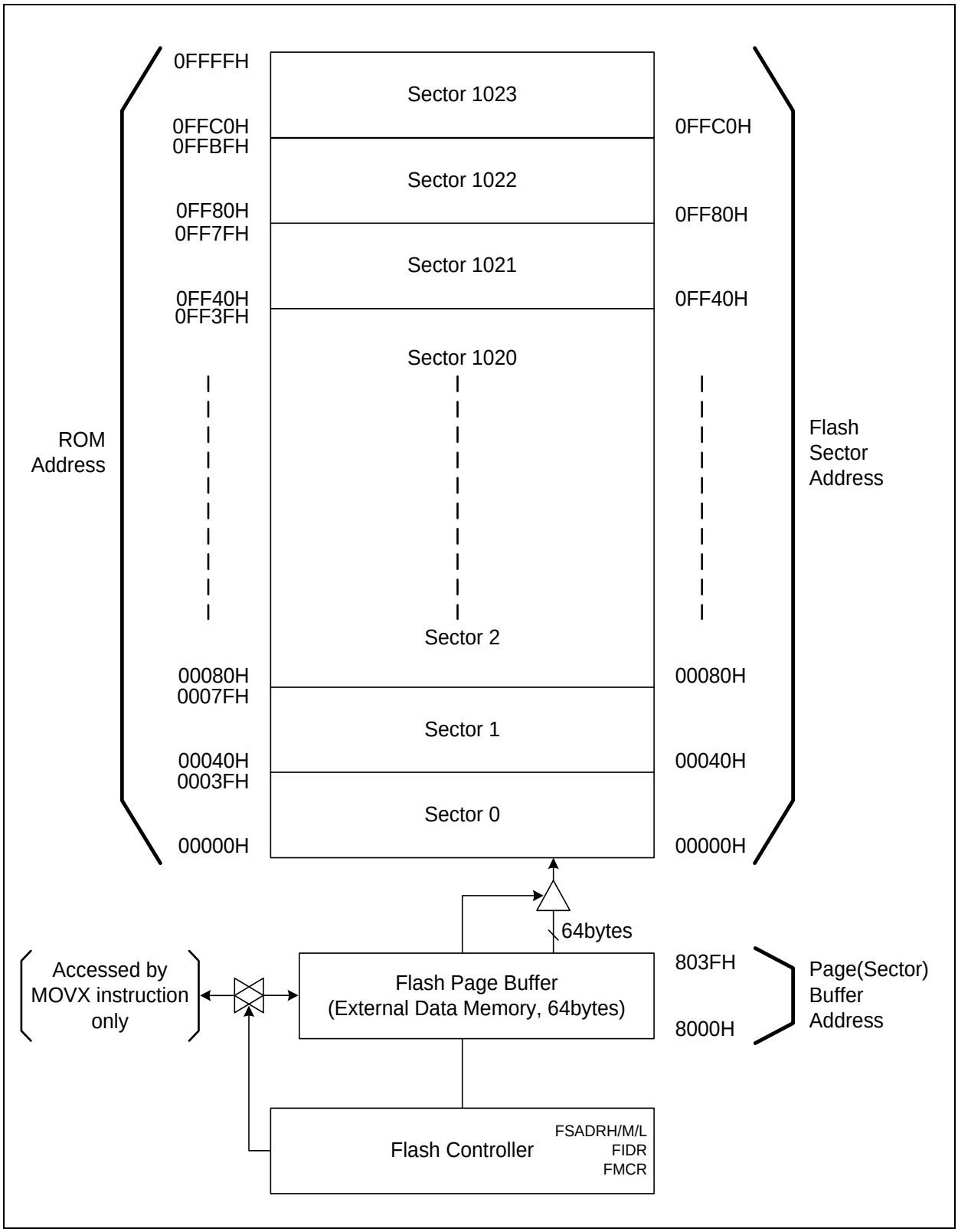



**Figure 15.1 Flash Program ROM Structure**

### 15.1.3 Register Map

**Table 15-1Flash Memory Register Map**

| Name | Address | Dir | Default | Description |
|---|---|---|---|---|
| FSADRH | FAH | R/W | 00H | Flash Sector Address High Register |
| FSADRM | FBH | R/W | 00H | Flash Sector Address Middle Register |
| FSADRL | FCH | R/W | 00H | Flash Sector Address Low Register |
| FIDR | FDH | R/W | 00H | Flash Identification Register |
| FMCR | FEH | R/W | 00H | Flash Mode Control Register |

### 15.1.4 Register Description for Flash Memory Control and Status

Flash control register consists of the flash sector address high register (FSADRH), flash sector address middle register (FSADRM), flash sector address low register (FSADRL), flash identification register (FIDR), and flash mode control register (FMCR). They are mapped to SFR area and can be accessed only in programming mode.

## 15.1.5 Register Description for Flash

**FSADRH (Flash Sector Address High Register) : FAH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | FSADRH3 | FSADRH2 | FSADRH1 | FSADRH0 |
| – | – | – | – | R/W | R/W | R/W | R/W |

Initial value : 00H

**FSADRH[3:0]** Flash Sector Address High

**FSADRM (Flash Sector Address Middle Register) : FBH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FSADRM7 | FSADRM6 | FSADRM5 | FSADRM4 | FSADRM3 | FSADRM2 | FSADRM1 | FSADRM0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**FSADRM[7:0]** Flash Sector Address Middle

**FSADRL (Flash Sector Address Low Register) : FCH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FSADRL7 | FSADRL6 | FSADRL5 | FSADRL4 | FSADRL3 | FSADRL2 | FSADRL1 | FSADRL0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**FSADRL[7:0]** Flash Sector Address Low

**FIDR (Flash Identification Register) : FDH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FIDR7 | FIDR6 | FIDR5 | FIDR4 | FIDR3 | FIDR2 | FIDR1 | FIDR0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Initial value : 00H

**FIDR[7:0]** Flash Identification

Others No identification value

10100101 Identification value for a flash mode

(These bits are automatically cleared to logic ‘00H’ immediately after one time operation)

**FMCR (Flash Mode Control Register) : FEH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FMBUSY | – | – | – | – | FMCR2 | FMCR1 | FMCR0 |
| R | – | – | – | – | R/W | R/W | R/W |

Initial value : 00H

**FMBUSY** Flash Mode Busy Bit. This bit will be used for only debugger.

0 No effect when “1” is written

1 Busy

**FMCR[2:0]** Flash Mode Control Bits. During a flash mode operation, the CPU is hold and the global interrupt is on disable state regardless of the IE.7 (EA) bit.

| FMCR2 | FMCR1 | FMCR0 | Description |
|---|---|---|---|
| 0 | 0 | 1 | Select flash page buffer reset mode and start regardless of the FIDR value (Clear all 64bytes to ‘0’) |
| 0 | 1 | 0 | Select flash sector erase mode and start operation when the FIDR=”10100101b’ |
| 0 | 1 | 1 | Select flash sector write mode and start operation when the FIDR=”10100101b’ |
| 1 | 0 | 0 | Select flash sector hard lock and start operation when the FIDR=”10100101b’ |

Others Values: No operation

(These bits are automatically cleared to logic ‘00H’ immediately after one time operation)

### 15.1.6 Serial In-System Program (ISP) Mode

Serial in-system program uses the interface of debugger which uses two wires. Refer to chapter 14 in details about debugger

### 15.1.7 Protection Area (User program mode)

MC96F7864 can program its own flash memory (protection area). The protection area can not be erased or programmed. The protection areas are available only when the PAEN bit is cleared to '0', that is, enable protection area at the configure option 2 if it is needed. If the protection area isn't enabled (PAEN ='1'), this area can be used as a normal program memory.

The size of protection area can be varied by setting of configure option 2.

**Table 15-2 Protection Area size**

| Protection Area Size Select | | Size of Protection Area | Address of Protection Area |
|---|---|---|---|
| **PASS1** | **PASS0** | | |
| 0 | 0 | 3.8k Bytes | 0100H – 0FFFH |
| 0 | 1 | 1.7k Bytes | 0100H – 07FFH |
| 1 | 0 | 768 Bytes | 0100H – 03FFH |
| 1 | 1 | 256 Bytes | 0100H – 01FFH |

NOTE) Refer to chapter 16 in configure option control.

### 15.1.8 Erase Mode

## The sector erase program procedure in user program mode

1. Page buffer clear (FMCR=0x01)
2. Write '0' to page buffer
3. Set flash sector address register (FSADRH/FSADRM/FSADRL).
4. Set flash identification register (FIDR).
5. Set flash mode control register (FMCR).
6. Erase verify

**Program Tip – sector erase**

```
            MOV     FMCR,#0x01          ;page buffer clear
            NOP                         ;Dummy instruction, This instruction must be needed.
            NOP                         ;Dummy instruction, This instruction must be needed.
            NOP                         ;Dummy instruction, This instruction must be needed.

            MOV     A,#0
            MOV     R0,#64              ;Sector size is 64bytes
            MOV     DPH,#0x80
            MOV     DPL,#0

Pgbuf_clr:  MOVX    @DPTR,A
            INC     DPTR
            DJNZ    R0, Pgbuf_clr       ;Write '0' to all page buffer

            MOV     FSADRH,#0x00
            MOV     FSADRM,#0x7F
            MOV     FSADRL,#0x40        ;Select sector 509
            MOV     FIDR,#0xA5          ;Identification value
            MOV     FMCR,#0x02          ;Start flash erase mode
            NOP                         ;Dummy instruction, This instruction must be needed.
            NOP                         ;Dummy instruction, This instruction must be needed.
            NOP                         ;Dummy instruction, This instruction must be needed.

            MOV     A,#0                ;erase verify
            MOV     R0,#64              ;Sector size is 64bytes
            MOV     R1,#0
            MOV     DPH,#0x7F
            MOV     DPL,#0x40

Erase_verify:
            MOVC    A,@A+DPTR
            SUBB    A,R1
            JNZ     Verify_error
            INC     DPTR
            DJNZ    R0, Erase_verify


Verify_error:
```

## The Byte erase program procedure in user program mode

1. Page buffer clear (FMCR=0x01)
2. Write '0' to page buffer
3. Set flash sector address register (FSADRH/FSADRM/FSADRL).
4. Set flash identification register (FIDR).
5. Set flash mode control register (FMCR).
6. Erase verify

**Program Tip – byte erase**

```
MOV     FMCR,#0x01          ;page buffer clear
NOP                         ;Dummy instruction, This instruction must be needed.
NOP                         ;Dummy instruction, This instruction must be needed.
NOP                         ;Dummy instruction, This instruction must be needed.

MOV     A,#0
MOV     DPH,#0x80
MOV     DPL,#0
MOVX    @DPTR,A

MOV     DPH,#0x80
MOV     DPL,#0x05
MOVX    @DPTR,A             ;Write '0' to page buffer

MOV     FSADRH,#0x00
MOV     FSADRM,#0x7F
MOV     FSADRL,#0x40        ;Select sector 509
MOV     FIDR,#0xA5          ;Identification value
MOV     FMCR,#0x02          ;Start flash erase mode
NOP                         ;Dummy instruction, This instruction must be needed.
NOP                         ;Dummy instruction, This instruction must be needed.
NOP                         ;Dummy instruction, This instruction must be needed.

MOV     A,#0                ;erase verify
MOV     R1,#0
MOV     DPH,#0x7F
MOV     DPL,#0x40
MOVC    A,@A+DPTR
SUBB    A,R1                ;0x7F40 = 0 ?
JNZ     Verify_error

MOV     A,#0
MOV     R1,#0
MOV     DPH,#0x7F
MOV     DPL,#0x45
MOVC    A,@A+DPTR
SUBB    A,R1                ;0x7F45 = 0 ?
JNZ     Verify_error


Verify_error:
```

### 15.1.9 Write Mode

## The sector Write program procedure in user program mode

1. Page buffer clear (FMCR=0x01)
2. Write data to page buffer
3. Set flash sector address register (FSADRH/FSADRM/FSADRL).
4. Set flash identification register (FIDR).
5. Set flash mode control register (FMCR).
6. Erase verify

**Program Tip – sector write**

```
            MOV     FMCR,#0x01          ;page buffer clear
            NOP                         ;Dummy instruction, This instruction must be needed.
            NOP                         ;Dummy instruction, This instruction must be needed.
            NOP                         ;Dummy instruction, This instruction must be needed.

            MOV     A,#0
            MOV     R0,#64              ;Sector size is 64bytes
            MOV     DPH,#0x80
            MOV     DPL,#0

Pgbuf_WR:   MOVX    @DPTR,A
            INC     A
            INC     DPTR
            DJNZ    R0, Pgbuf_WR        ;Write data to all page buffer

            MOV     FSADRH,#0x00
            MOV     FSADRM,#0x7F
            MOV     FSADRL,#0x40        ;Select sector 509
            MOV     FIDR,#0xA5          ;Identification value
            MOV     FMCR,#0x03          ;Start flash write mode
            NOP                         ;Dummy instruction, This instruction must be needed.
            NOP                         ;Dummy instruction, This instruction must be needed.
            NOP                         ;Dummy instruction, This instruction must be needed.

            MOV     A,#0                ;write verify
            MOV     R0,#64              ;Sector size is 64bytes
            MOV     R1,#0
            MOV     DPH,#0x7F
            MOV     DPL,#0x40

Write_verify:
            MOVC    A,@A+DPTR
            SUBB    A,R1
            JNZ     Verify_error
            INC     R1
            INC     DPTR
            DJNZ    R0, Write_verify


Verify_error:
```

## The Byte Write program procedure in user program mode

1. Page buffer clear (FMCR=0x01)
2. Write data to page buffer
3. Set flash sector address register (FSADRH/FSADRM/FSADRL).
4. Set flash identification register (FIDR).
5. Set flash mode control register (FMCR).
6. Erase verify

**Program Tip – byte write**

```
MOV     FMCR,#0x01          ;page buffer clear
NOP                         ;Dummy instruction, This instruction must be needed.
NOP                         ;Dummy instruction, This instruction must be needed.
NOP                         ;Dummy instruction, This instruction must be needed.

MOV     A,#5
MOV     DPH,#0x80
MOV     DPL,#0
MOVX    @DPTR,A             ;Write data to page buffer

MOV     A,#6
MOV     DPH,#0x80
MOV     DPL,#0x05
MOVX    @DPTR,A             ;Write data to page buffer

MOV     FSADRH,#0x00
MOV     FSADRM,#0x7F
MOV     FSADRL,#0x40        ;Select sector 509
MOV     FIDR,#0xA5          ;Identification value
MOV     FMCR,#0x03          ;Start flash write mode
NOP                         ;Dummy instruction, This instruction must be needed.
NOP                         ;Dummy instruction, This instruction must be needed.
NOP                         ;Dummy instruction, This instruction must be needed.

MOV     A,#0                ;write verify
MOV     R1,#5
MOV     DPH,#0x7F
MOV     DPL,#0x40
MOVC    A,@A+DPTR
SUBB    A,R1                ;0x7F40 = 5 ?
JNZ     Verify_error

MOV     A,#0
MOV     R1,#6
MOV     DPH,#0x7F
MOV     DPL,#0x45
MOVC    A,@A+DPTR
SUBB    A,R1                ;0x7F45 = 6 ?
JNZ     Verify_error

Verify_error:
```

### 15.1.10 Read Mode

## The Reading program procedure in user program mode

1. Load receive data from flash memory on MOVC instruction by indirectly addressing mode.

**Program Tip – reading**

```
MOV     A,#0
MOV     DPH,#0x7F
MOV     DPL,#0x40               ;flash memory address

MOVC    A,@A+DPTR               ;read data from flash memory
```

### 15.1.11 Hard Lock Mode

## The Reading program procedure in user program mode

1. Set flash identification register (FIDR).
2. Set flash mode control register (FMCR).

**Program Tip – reading**

```
MOV     FIDR,#0xA5              ;Identification value
MOV     FMCR,#0x04              ;Start flash hard lock mode
NOP                             ;Dummy instruction, This instruction must be needed.
NOP                             ;Dummy instruction, This instruction must be needed.
NOP                             ;Dummy instruction, This instruction must be needed.
```

# 16. Configure Option

## 16.1 Configure Option Control

The data for configure option should be written in the configure option area (003EH – 003FH) by programmer (Writer tools).

**CONFIGURE OPTION 1 : ROM Address 003FH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| R_P | HL | FFCHK | – | – | – | – | RSTS |

Initial value : 00H

**R_P** Read Protection
- 0 Disable “Read protection”
- 1 Enable “Read protection”

**HL** Hard-Lock
- 0 Disable “Hard-lock”
- 1 Enable “Hard-lock”

**FFCHK** Enable/Disable Parity bit Check function for Flash Fail Detection
- 0 Disable flash fail detection function
- 1 Enable flash fail detection function

NOTE) Whenever ROM code is fetched, parity (even parity) bit is Checked. If this bit is ‘1b’ and parity bit is incorrect, reset occurs.

**RSTS** RESETB Select
- 0 P66 port
- 1 RESETB port with a pull-up resistor

**CONFIGURE OPTION 2: ROM Address 003EH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | PAEN | PASS1 | PASS0 |

Initial value : 00H

**PAEN** Protection Area Enable/Disable
- 0 Disable Protection (Erasable by instruction)
- 1 Enable Protection (Not erasable by instruction)

**PASS [1:0]** Protection Area Size Select

| PASS1 | PASS0 | Description |
|---|---|---|
| 0 | 0 | 3.8k Bytes (Address 0100H – 0FFFH) |
| 0 | 1 | 1.7k Bytes (Address 0100H – 07FFH) |
| 1 | 0 | 768 Bytes (Address 0100H – 03FFH) |
| 1 | 1 | 256 Bytes (Address 0100H – 01FFH) |

# 17. APPENDIX

## A. Instruction Table

Instructions are either 1, 2 or 3 bytes long as listed in the 'Bytes' column below.

Each instruction takes either 1, 2 or 4 machine cycles to execute as listed in the following table. 1 machine cycle comprises 2 system clock cycles.

| ARITHMETIC | | | | |
|---|---|---|---|---|
| **Mnemonic** | **Description** | **Bytes** | **Cycles** | **Hex code** |
| ADD A,Rn | Add register to A | 1 | 1 | 28-2F |
| ADD A,dir | Add direct byte to A | 2 | 1 | 25 |
| ADD A,@Ri | Add indirect memory to A | 1 | 1 | 26-27 |
| ADD A,#data | Add immediate to A | 2 | 1 | 24 |
| ADDC A,Rn | Add register to A with carry | 1 | 1 | 38-3F |
| ADDC A,dir | Add direct byte to A with carry | 2 | 1 | 35 |
| ADDC A,@Ri | Add indirect memory to A with carry | 1 | 1 | 36-37 |
| ADDC A,#data | Add immediate to A with carry | 2 | 1 | 34 |
| SUBB A,Rn | Subtract register from A with borrow | 1 | 1 | 98-9F |
| SUBB A,dir | Subtract direct byte from A with borrow | 2 | 1 | 95 |
| SUBB A,@Ri | Subtract indirect memory from A with borrow | 1 | 1 | 96-97 |
| SUBB A,#data | Subtract immediate from A with borrow | 2 | 1 | 94 |
| INC A | Increment A | 1 | 1 | 04 |
| INC Rn | Increment register | 1 | 1 | 08-0F |
| INC dir | Increment direct byte | 2 | 1 | 05 |
| INC @Ri | Increment indirect memory | 1 | 1 | 06-07 |
| DEC A | Decrement A | 1 | 1 | 14 |
| DEC Rn | Decrement register | 1 | 1 | 18-1F |
| DEC dir | Decrement direct byte | 2 | 1 | 15 |
| DEC @Ri | Decrement indirect memory | 1 | 1 | 16-17 |
| INC DPTR | Increment data pointer | 1 | 2 | A3 |
| MUL AB | Multiply A by B | 1 | 4 | A4 |
| DIV AB | Divide A by B | 1 | 4 | 84 |
| DA A | Decimal Adjust A | 1 | 1 | D4 |

| LOGICAL | | | | |
|---|---|---|---|---|
| **Mnemonic** | **Description** | **Bytes** | **Cycles** | **Hex code** |
| ANL A,Rn | AND register to A | 1 | 1 | 58-5F |
| ANL A,dir | AND direct byte to A | 2 | 1 | 55 |
| ANL A,@Ri | AND indirect memory to A | 1 | 1 | 56-57 |
| ANL A,#data | AND immediate to A | 2 | 1 | 54 |
| ANL dir,A | AND A to direct byte | 2 | 1 | 52 |
| ANL dir,#data | AND immediate to direct byte | 3 | 2 | 53 |
| ORL A,Rn | OR register to A | 1 | 1 | 48-4F |
| ORL A,dir | OR direct byte to A | 2 | 1 | 45 |
| ORL A,@Ri | OR indirect memory to A | 1 | 1 | 46-47 |
| ORL A,#data | OR immediate to A | 2 | 1 | 44 |
| ORL dir,A | OR A to direct byte | 2 | 1 | 42 |
| ORL dir,#data | OR immediate to direct byte | 3 | 2 | 43 |
| XRL A,Rn | Exclusive-OR register to A | 1 | 1 | 68-6F |
| XRL A,dir | Exclusive-OR direct byte to A | 2 | 1 | 65 |
| XRL A, @Ri | Exclusive-OR indirect memory to A | 1 | 1 | 66-67 |

| | | | | |
|---|---|---|---|---|
| XRL A,#data | Exclusive-OR immediate to A | 2 | 1 | 64 |
| XRL dir,A | Exclusive-OR A to direct byte | 2 | 1 | 62 |
| XRL dir,#data | Exclusive-OR immediate to direct byte | 3 | 2 | 63 |
| CLR A | Clear A | 1 | 1 | E4 |
| CPL A | Complement A | 1 | 1 | F4 |
| SWAP A | Swap Nibbles of A | 1 | 1 | C4 |
| RL A | Rotate A left | 1 | 1 | 23 |
| RLC A | Rotate A left through carry | 1 | 1 | 33 |
| RR A | Rotate A right | 1 | 1 | 03 |
| RRC A | Rotate A right through carry | 1 | 1 | 13 |

| **DATA TRANSFER** | | | | |
|---|---|---|---|---|
| **Mnemonic** | **Description** | **Bytes** | **Cycles** | **Hex code** |
| MOV A,Rn | Move register to A | 1 | 1 | E8-EF |
| MOV A,dir | Move direct byte to A | 2 | 1 | E5 |
| MOV A,@Ri | Move indirect memory to A | 1 | 1 | E6-E7 |
| MOV A,#data | Move immediate to A | 2 | 1 | 74 |
| MOV Rn,A | Move A to register | 1 | 1 | F8-FF |
| MOV Rn,dir | Move direct byte to register | 2 | 2 | A8-AF |
| MOV Rn,#data | Move immediate to register | 2 | 1 | 78-7F |
| MOV dir,A | Move A to direct byte | 2 | 1 | F5 |
| MOV dir,Rn | Move register to direct byte | 2 | 2 | 88-8F |
| MOV dir,dir | Move direct byte to direct byte | 3 | 2 | 85 |
| MOV dir,@Ri | Move indirect memory to direct byte | 2 | 2 | 86-87 |
| MOV dir,#data | Move immediate to direct byte | 3 | 2 | 75 |
| MOV @Ri,A | Move A to indirect memory | 1 | 1 | F6-F7 |
| MOV @Ri,dir | Move direct byte to indirect memory | 2 | 2 | A6-A7 |
| MOV @Ri,#data | Move immediate to indirect memory | 2 | 1 | 76-77 |
| MOV DPTR,#data | Move immediate to data pointer | 3 | 2 | 90 |
| MOVC A,@A+DPTR | Move code byte relative DPTR to A | 1 | 2 | 93 |
| MOVC A,@A+PC | Move code byte relative PC to A | 1 | 2 | 83 |
| MOVX A,@Ri | Move external data(A8) to A | 1 | 2 | E2-E3 |
| MOVX A,@DPTR | Move external data(A16) to A | 1 | 2 | E0 |
| MOVX @Ri,A | Move A to external data(A8) | 1 | 2 | F2-F3 |
| MOVX @DPTR,A | Move A to external data(A16) | 1 | 2 | F0 |
| PUSH dir | Push direct byte onto stack | 2 | 2 | C0 |
| POP dir | Pop direct byte from stack | 2 | 2 | D0 |
| XCH A,Rn | Exchange A and register | 1 | 1 | C8-CF |
| XCH A,dir | Exchange A and direct byte | 2 | 1 | C5 |
| XCH A,@Ri | Exchange A and indirect memory | 1 | 1 | C6-C7 |
| XCHD A,@Ri | Exchange A and indirect memory nibble | 1 | 1 | D6-D7 |

| **BOOLEAN** | | | | |
|---|---|---|---|---|
| **Mnemonic** | **Description** | **Bytes** | **Cycles** | **Hex code** |
| CLR C | Clear carry | 1 | 1 | C3 |
| CLR bit | Clear direct bit | 2 | 1 | C2 |
| SETB C | Set carry | 1 | 1 | D3 |
| SETB bit | Set direct bit | 2 | 1 | D2 |
| CPL C | Complement carry | 1 | 1 | B3 |
| CPL bit | Complement direct bit | 2 | 1 | B2 |
| ANL C,bit | AND direct bit to carry | 2 | 2 | 82 |
| ANL C,/bit | AND direct bit inverse to carry | 2 | 2 | B0 |

| | | | | |
|---|---|---|---|---|
| ORL C,bit | OR direct bit to carry | 2 | 2 | 72 |
| ORL C,/bit | OR direct bit inverse to carry | 2 | 2 | A0 |
| MOV C,bit | Move direct bit to carry | 2 | 1 | A2 |
| MOV bit,C | Move carry to direct bit | 2 | 2 | 92 |

| **BRANCHING** | | | | |
|---|---|---|---|---|
| **Mnemonic** | **Description** | **Bytes** | **Cycles** | **Hex code** |
| ACALL addr 11 | Absolute jump to subroutine | 2 | 2 | 11→F1 |
| LCALL addr 16 | Long jump to subroutine | 3 | 2 | 12 |
| RET | Return from subroutine | 1 | 2 | 22 |
| RETI | Return from interrupt | 1 | 2 | 32 |
| AJMP addr 11 | Absolute jump unconditional | 2 | 2 | 01→E1 |
| LJMP addr 16 | Long jump unconditional | 3 | 2 | 02 |
| SJMP rel | Short jump (relative address) | 2 | 2 | 80 |
| JC rel | Jump on carry = 1 | 2 | 2 | 40 |
| JNC rel | Jump on carry = 0 | 2 | 2 | 50 |
| JB bit,rel | Jump on direct bit = 1 | 3 | 2 | 20 |
| JNB bit,rel | Jump on direct bit = 0 | 3 | 2 | 30 |
| JBC bit,rel | Jump on direct bit = 1 and clear | 3 | 2 | 10 |
| JMP @A+DPTR | Jump indirect relative DPTR | 1 | 2 | 73 |
| JZ rel | Jump on accumulator = 0 | 2 | 2 | 60 |
| JNZ rel | Jump on accumulator ≠ 0 | 2 | 2 | 70 |
| CJNE A,dir,rel | Compare A,direct jne relative | 3 | 2 | B5 |
| CJNE A,#d,rel | Compare A,immediate jne relative | 3 | 2 | B4 |
| CJNE Rn,#d,rel | Compare register, immediate jne relative | 3 | 2 | B8-BF |
| CJNE @Ri,#d,rel | Compare indirect, immediate jne relative | 3 | 2 | B6-B7 |
| DJNZ Rn,rel | Decrement register, jnz relative | 3 | 2 | D8-DF |
| DJNZ dir,rel | Decrement direct byte, jnz relative | 3 | 2 | D5 |

| **MISCELLANEOUS** | | | | |
|---|---|---|---|---|
| **Mnemonic** | **Description** | **Bytes** | **Cycles** | **Hex code** |
| NOP | No operation | 1 | 1 | 00 |

| **ADDITIONAL INSTRUCTIONS (selected through EO[7:4])** | | | | |
|---|---|---|---|---|
| **Mnemonic** | **Description** | **Bytes** | **Cycles** | **Hex code** |
| MOVC @(DPTR++),A | M8051W/M8051EW-specific instruction supporting software download into program memory | 1 | 2 | A5 |
| TRAP | Software break command | 1 | 1 | A5 |

In the above table, an entry such as E8-EF indicates a continuous block of hex opcodes used for 8 different registers, the register numbers of which are defined by the lowest three bits of the corresponding code. Non-continuous blocks of codes, shown as 11→F1 (for example), are used for absolute jumps and calls, with the top 3 bits of the code being used to store the top three bits of the destination address.

The CJNE instructions use the abbreviation #d for immediate data; other instructions use #data.