

---

## CMOS Single Chip 8-bit MCU for Touch Sensing, Smart Proximity Sensing Microcontroller

---

UM Rev. 1.13

### Introduction

This user's manual targets application developers who use A96T418 for their specific needs. It provides complete information of how to use A96T418 device. Standard functions and blocks including corresponding register information of A96T418 are introduced in each chapter, while instruction set is in Appendix.

A96T418 is based on M8051 core, and provides standard features of 8051 such as 8-bit ALU, PC, 8-bit registers, timers and counters, serial data communication, PSW, DPTR, SP, 8-bit data bus and 2x16-bit address bus, and 8/11/16-bit operations.

In addition, A96T418 incorporates following features to offer highly flexible and cost effective solutions:

- 32Kbytes of FLASH, 256bytes of IRAM, and 1792bytes of XRAM
- Basic interval timer, watchdog timer, and 16-bit timer/counter
- 16-bit PPG output, I2C, and 12-bit ADC
- 20-Channel Self Capacitive Touch Sensing
- 8COM / 16SEG LED Driver  
(T-type max LED [8COM x 7SEG], M-type max LED [8COM x 8SEG], Max LED [8COM x 15SEG] - T-type, M-type mix)
- On-chip POR, LVR
- On-chip oscillator and clock circuitry.

As a field proven best seller, A96T418 introduces rich features such as excellent noise immunity, code optimization, cost effectiveness, and so on.

### Reference document

- A96T418 programming tools and manuals released by ABOV: They are available at ABOV website, [www.abovsemi.com](http://www.abovsemi.com).
- SDK-51 User's guide (System Design Kit) released by Intel in 1982: It contains all of components of a single-board computer based on Intel's 8051 single-chip microcomputer
- Information on Mentor Graphics 8051 microcontroller: The technical document is provided at Mentor® website: <https://www.mentor.com/products/ip/peripheral/microcontroller/>

## Contents

Introduction.....	1
Reference document.....	1
1 Description .....	16
1.1 Device overview.....	16
1.2 Block Diagram .....	18
2 Pinouts and pin description.....	19
2.1 Pinouts .....	19
2.2 Pin Description.....	24
3 Port Structures .....	30
3.1 General Purpose I/O Port .....	30
3.2 SEG I/O Port.....	31
3.3 COM & SEG I/O Port.....	32
4 Memory organization .....	33
4.1 Program Memory .....	33
4.2 Data Memory .....	35
4.3 External Data Memory.....	37
4.4 SFR Map.....	38
4.4.1 SFR Map Summary.....	38
4.4.2 SFR Map .....	40
4.4.3 Compiler Compatible SFR .....	46
5 I/O Ports .....	48
5.1 I/O Ports.....	48
5.2 Port Register.....	48
5.2.1 Data Register (Px).....	48
5.2.2 Direction Register (PxIO) .....	48
5.2.3 Pull-up Resistor Selection Register (PxPU).....	48
5.2.4 Open-drain Selection Register (PxOD).....	48
5.2.5 De-bounce Enable Register (PxDB) .....	48
5.2.6 Port Function Selection Register (PxFSR).....	49
5.2.7 Register Map .....	49
5.3 P0 Port.....	50
5.3.1 P0 Port Description .....	50
5.3.2 Register Description for P0 .....	50
5.4 P1 Port.....	54
5.4.1 P1 Port Description .....	54
5.4.2 Register description for P1 .....	54
5.5 P2 Port.....	58
5.5.1 P2 Port Description .....	58
5.5.2 Register Description for P2 .....	58
5.6 P3 Port.....	61
5.6.1 P3 Port Description .....	61
5.6.2 Register Description for P3 .....	61

6	Interrupt Controller .....	63
6.1	Overview .....	63
6.2	External Interrupt .....	65
6.3	Block Diagram .....	66
6.4	Interrupt Vector Table .....	67
6.5	Interrupt Sequence .....	68
6.6	Effective Timing after Controlling Interrupt Bit .....	69
6.7	Multi Interrupt .....	70
6.8	Interrupt Enable Accept Timing .....	71
6.9	Interrupt Service Routine Address .....	71
6.10	Saving/Restore General-Purpose Registers .....	71
6.11	Interrupt Timing .....	72
6.12	Interrupt Register Overview .....	72
6.12.1	Interrupt Enable Register (IE, IE1, IE2, IE3) .....	72
6.12.2	Interrupt Priority Register (IP, IP1) .....	72
6.12.3	External Interrupt Flag Register (EIFLAG0, EIFLAG1) .....	73
6.12.4	External Interrupt Polarity Register (EIPOL0L, EIPOL0H, EIPOL1) .....	73
6.12.5	Register Map .....	73
6.12.6	Interrupt Register Description .....	73
6.12.7	Register Description for Interrupt .....	74
7	Clock Generator .....	80
7.1	Overview .....	80
7.2	Block Diagram .....	81
7.3	Register Map .....	81
7.4	Clock Generator Register Description .....	81
7.5	Register Description for Clock Generator .....	82
8	Basic Interval Timer .....	84
8.1	Overview .....	84
8.2	Block Diagram .....	84
8.3	Register Map .....	85
8.4	Basic Interval Timer Register Description .....	85
8.5	Register Description for Basic Interval Timer .....	86
9	Watchdog Timer .....	87
9.1	Overview .....	87
9.2	WDT Interrupt Timing Waveform .....	87
9.3	Block Diagram .....	88
9.4	Register Map .....	88
9.5	Watchdog Timer Register Description .....	88
9.6	Register Description for Watchdog Timer .....	89
10	Watch Timer .....	90
10.1	Overview .....	90
10.2	Block Diagram .....	90
10.3	Register Map .....	91
10.4	Watch Timer Register Description .....	91
10.5	Register Description for Watch Timer .....	91

11	Timer 0/1/2/3/4/5 .....	93
11.1	Timer 0 .....	93
11.1.1	Overview .....	93
11.1.2	8-bit Timer/Counter Mode .....	94
11.1.3	8-bit PWM Mode .....	95
11.1.4	8-bit Capture Mode .....	98
11.1.5	Block Diagram .....	100
11.1.6	Register Map .....	100
11.1.7	Timer/Counter 0 Register Description .....	100
11.1.8	Register Description for Timer/Counter 0 .....	101
11.2	Timer 1 .....	103
11.2.1	Overview .....	103
11.2.2	16-bit Timer/Counter Mode .....	103
11.2.3	16-bit Capture Mode .....	105
11.2.4	16-bit PPG Mode .....	107
11.2.5	Block Diagram .....	109
11.2.6	Register Map .....	109
11.2.7	Timer/Counter 1 Register Description .....	110
11.2.8	Register Description for Timer/Counter 1 .....	110
11.3	Timer 2 .....	113
11.3.1	Overview .....	113
11.3.2	16-bit Timer/Counter Mode .....	114
11.3.3	16-bit Capture Mode .....	115
11.3.4	16-bit PPG Mode .....	118
11.3.5	Block Diagram .....	120
11.3.6	Register Map .....	120
11.3.7	Timer/Counter 2 Register Description .....	121
11.3.8	Register Description for Timer/Counter 2 .....	121
11.4	Timer 3 .....	124
11.4.1	Overview .....	124
11.4.2	16-bit Timer/Counter Mode .....	124
11.4.3	16-bit Capture Mode .....	126
11.4.4	16-bit PPG Mode .....	128
11.4.5	Block Diagram .....	130
11.4.6	Register Map .....	130
11.4.7	Timer/Counter 3 Register Description .....	131
11.4.8	Register Description for Timer/Counter 3 .....	131
11.5	Timer 4 .....	133
11.5.1	Overview .....	133
11.5.2	16-bit Timer/Counter Mode .....	134
11.5.3	16-bit Capture Mode .....	136
11.5.4	16-bit PPG Mode .....	138
11.5.5	Block Diagram .....	140
11.5.6	Register Map .....	140
11.5.7	Timer/Counter 4 Register Description .....	141

11.5.8	Register Description for Timer/Counter 4.....	141
11.6	Timer 5.....	143
11.6.1	Overview .....	143
11.6.2	16-bit Timer/Counter Mode .....	143
11.6.3	16-bit Capture Mode .....	144
11.6.4	16-bit PPG Mode.....	147
11.6.5	Block Diagram .....	149
11.6.6	Register Map .....	149
11.6.7	Timer/Counter 5 Register Description.....	150
11.6.8	Register Description for Timer/Counter 5.....	150
12	Buzzer Driver .....	152
12.1	Overview .....	152
12.2	Block Diagram .....	152
12.3	Register Map .....	153
12.4	Buzzer Driver Register Description .....	153
12.5	Register Description for Buzzer Driver .....	153
13	USI (USART + SPI + I2C).....	154
13.1	Overview .....	154
13.2	USIn UART Mode .....	155
13.3	USIn UART Block Diagram.....	156
13.4	USIn Clock Generation .....	157
13.5	USIn External Clock (SCKn).....	157
13.6	USIn Synchronous mode operation.....	158
13.7	USIn UART Data format .....	159
13.8	USIn UART Parity bit .....	159
13.9	USIn UART Transmitter .....	160
13.9.1	USIn UART Sending TX data.....	160
13.9.2	USIn UART Transmitter flag and interrupt .....	160
13.9.3	USIn UART Parity Generator .....	160
13.9.4	USIn UART Disabling Transmitter.....	161
13.10	USIn UART Receiver .....	161
13.10.1	USIn UART Receiving RX data.....	161
13.10.2	USIn UART Receiver Flag and Interrupt.....	162
13.10.3	USIn UART Parity Checker.....	162
13.10.4	USIn UART Disabling Receiver .....	162
13.10.5	USIn Asynchronous Data Reception.....	163
13.11	USIn SPI Mode .....	165
13.12	USIn SPI Clock Formats and Timing .....	165
13.13	USIn SPI Block Diagram .....	168
13.14	USIn I2C Mode .....	169
13.15	USIn I2C Bit Transfer .....	169
13.16	USIn I2C Start / Repeated Start / Stop .....	170
13.17	USIn I2C Data Transfer .....	170
13.18	USIn I2C Acknowledge .....	171
13.19	USIn I2C Synchronization / Arbitration .....	171

13.20 USIn I2C Operation .....	172
13.20.1 USIn I2C Master Transmitter.....	173
13.20.2 USIn I2C Master Receiver .....	175
13.20.3 USIn I2C Slave Transmitter.....	177
13.20.4 USIn I2C Slave Receiver .....	178
13.21 USIn I2C Block Diagram.....	179
13.22 Register Map .....	180
13.23 USIn Register Description .....	180
13.24 Register Description for USIn .....	181
13.25 Baud Rate setting (example) .....	190
14 USART1 .....	192
14.1 Overview .....	192
14.2 Block Diagram .....	193
14.3 Clock Generation .....	194
14.4 External Clock (XCK).....	195
14.5 Synchronous mode Operation.....	195
14.6 Data format.....	196
14.7 Parity bit.....	196
14.8 USART1 Transmitter.....	197
14.8.1 Sending Tx data .....	197
14.8.2 Transmitter flag and interrupt .....	197
14.8.3 Parity Generator.....	197
14.8.4 Disabling Transmitter .....	198
14.9 USART1 Receiver .....	198
14.9.1 Receiving Rx data .....	198
14.9.2 Receiver flag and interrupt .....	198
14.9.3 Parity Checker.....	199
14.9.4 Disabling Receiver .....	199
14.9.5 Asynchronous Data Reception .....	199
14.10 SPI Mode .....	201
14.10.1 SPI Clock formats and timing.....	201
14.11 Receiver Time Out (RTO) .....	203
14.12 Register Map .....	205
14.13 USART1 Register Description .....	205
14.14 Register Description for USART1 .....	206
14.15 Baud Rate setting (example) .....	213
14.16 0% Error Baud Rate.....	214
15 LED Driver .....	215
15.1 Overview .....	215
15.2 Block Diagram .....	215
15.3 Register Map .....	216
15.4 Register Description for LED Driver .....	217
15.5 LED RAM.....	225
15.6 Example Circuit.....	226
15.6.1 T-Type Matrix .....	226

15.6.2	M-Type Matrix .....	227
15.6.3	Circuit for LED and Touch by common pin.....	228
15.7	Mode Functions .....	229
15.7.1	LED AUTO Mode.....	229
15.7.2	LED Alone Mode .....	230
15.7.3	Hand Shake Mode .....	231
15.7.4	LED Stop Count Mode .....	232
15.7.5	Smart Share Mode .....	232
16	20-Channel Touch Switch .....	234
16.1	Features.....	234
16.2	Block Diagram .....	235
16.3	CAPN Port .....	235
16.4	Register Map .....	236
16.5	Register Description for Touch Sensing .....	237
16.6	User Programming Procedure.....	250
17	12-bit A/D Converter .....	251
17.1	Overview .....	251
17.2	Conversion Timing .....	251
17.3	Block Diagram .....	252
17.4	ADC Operation .....	253
17.5	Register Map .....	254
17.6	ADC Register Description.....	254
17.7	Register Description for ADC.....	255
18	Power Down Operation.....	258
18.1	Overview .....	258
18.2	Peripheral Operation in IDLE/STOP Mode .....	258
18.3	IDLE Mode.....	259
18.4	STOP Mode .....	260
18.5	Release Operation of STOP Mode .....	261
18.6	Register Map .....	262
18.7	Power Down Operation Register Description .....	262
18.8	Register Description for Power Down Operation.....	262
19	RESET .....	263
19.1	Overview .....	263
19.2	Reset Source .....	263
19.3	RESET Block Diagram .....	263
19.4	Power on RESET.....	264
19.5	External RESETB Input .....	267
19.6	Low Voltage Reset Processor .....	268
19.7	LVI Block Diagram .....	270
19.8	Register Map .....	270
19.9	Reset Operation Register Description .....	270
19.10	Register Description for Reset Operation.....	271
20	Memory Programming .....	274
20.1	Flash Control and Status Register.....	274

20.1.1	Register Map .....	274
20.1.2	Register Description for Flash.....	275
20.2	Memory Map .....	281
20.2.1	Flash Memory Map .....	281
20.3	Serial In-System Program Mode .....	282
20.3.1	Flash Operation.....	282
20.4	Mode Entrance Method of ISP Mode .....	289
20.4.1	Mode Entrance Method for ISP.....	289
20.5	Security .....	290
20.6	Configure Option.....	291
20.6.1	Configure Option Control .....	291
20.6.2	Register Description.....	291
21	Electrical Characteristics.....	292
21.1	Absolute Maximum Ratings .....	292
21.2	Recommended Operating Conditions .....	292
21.3	Touch Sensing Characteristics .....	293
21.4	A/D Converter Characteristics .....	293
21.5	Power-On Reset Characteristics .....	293
21.6	Low Voltage Reset and Low Voltage Indicator Characteristics .....	294
21.7	High Speed Internal RC Oscillator Characteristics.....	295
21.8	Low Speed Internal RC Oscillator Characteristics .....	295
21.9	DC Characteristics .....	296
21.10	AC Characteristics .....	297
21.11	USART Characteristics .....	298
21.12	SPI0 Characteristics .....	299
21.13	UART0 Characteristics .....	300
21.14	I2C0 Characteristics .....	301
21.15	Data Retention Voltage in Stop Mode .....	302
21.16	Internal Flash Rom Characteristics .....	303
21.17	Input/Output Capacitance .....	303
21.18	Sub Clock Oscillator Characteristics .....	304
21.19	Sub Oscillation Characteristics .....	305
21.20	Operating Voltage Range .....	305
21.21	Recommended Circuit and Layout .....	306
21.22	Typical Characteristics.....	307
22	Development Tools .....	309
22.1	Compiler .....	309
22.2	OCD(On-chip debugger) emulator and debugger .....	309
22.3	Programmer.....	310
22.3.1	E-PGM+ .....	310
22.4	Flash Programming .....	311
22.4.1	Overview .....	311
22.4.2	On-Board Programming .....	311
22.4.3	Circuit Design Guide .....	311
22.4.4	OCD Emulator.....	313

22.4.5	Gang Programmer .....	313
22.5	On-chip Debug System .....	314
22.5.1	Two-Pin External Interface .....	315
23	Package Diagram .....	319
24	Ordering Information .....	324
Appendix	.....	326
Revision History	.....	332

## List of Figures

Figure 1. Block diagram of A96T418.....	18
Figure 2. A96T418 28SOP pin assignment.....	19
Figure 3. A96T418 28TSSOP pin assignment.....	20
Figure 4. A96T418 24SOP pin assignment.....	21
Figure 5. A96T418 20SOP pin assignment.....	22
Figure 6. A96T418 20TSSOP pin assignment.....	23
Figure 7. General Purpose I/O Port .....	30
Figure 8. SEG I/O Port .....	31
Figure 9. COM & SEG I/O Port .....	32
Figure 10. Program Memory .....	34
Figure 11. Data Memory Map.....	35
Figure 12. Lower 128bytes RAM .....	36
Figure 13. XDATA Memory Area .....	37
Figure 14. External Interrupt Description .....	65
Figure 15. Block Diagram of Interrupt.....	66
Figure 16. Interrupt Sequence Flow.....	68
Figure 17. Effective Timing of Interrupt Enable Register .....	69
Figure 18. Effective Timing of Interrupt Flag Register.....	69
Figure 19. Effective Timing of Multi-Interrupt .....	70
Figure 20. Interrupt Response Timing Diagram .....	71
Figure 21. Correspondence between Vector Table Address and the Entry Address of ISR .....	71
Figure 22. Saving/Restore Process Diagram and Sample Source .....	71
Figure 23. Timing Chart of Interrupt Acceptance and Interrupt Return Instruction .....	72
Figure 24. Clock Generator Block Diagram .....	81
Figure 25. Basic Interval Timer Block Diagram.....	84
Figure 26. Watchdog Timer Interrupt Timing Waveform .....	87
Figure 27. Watchdog Timer Block Diagram .....	88
Figure 28. Watch Timer Block Diagram .....	90
Figure 29. 8-bit Timer/Counter Mode for Timer 0.....	94
Figure 30. 8-bit Timer/Counter 0 Example .....	95
Figure 31. 8-bit PWM Mode for Timer 0 .....	96
Figure 32. PWM Output Waveforms in PWM Mode for Timer 0 .....	97
Figure 33. 8-bit Capture Mode for Timer 0 .....	98
Figure 34. Input Capture Mode Operation for Timer 0 .....	99
Figure 35. Express Timer Overflow in Capture Mode .....	99
Figure 36. 8-bit Timer 0 Block Diagram .....	100
Figure 37. 16-bit Timer/Counter Mode for Timer 1.....	104
Figure 38. 16-bit Timer/Counter 1 Example .....	104
Figure 39. 16-bit Capture Mode for Timer 1 .....	105
Figure 40. Input Capture Mode Operation for Timer 1 .....	106
Figure 41. Express Timer Overflow in Capture Mode .....	106
Figure 42. 16-bit PPG Mode for Timer 1 .....	107
Figure 43. 16-bit PPG Mode Timming chart for Timer 1 .....	108
Figure 44. 16-bit Timer 1 Block Diagram .....	109
Figure 45. 16-bit Timer/Counter Mode for Timer 2 .....	114
Figure 46. 16-bit Timer/Counter 2 Example .....	115
Figure 47. 16-bit Capture Mode for Timer 2 .....	116
Figure 48. Input Capture Mode Operation for Timer 2 .....	117
Figure 49. Express Timer Overflow in Capture Mode .....	117

Figure 50. 16-bit PPG Mode for Timer 2 .....	118
Figure 51. 16-bit PPG Mode Timming chart for Timer 2 .....	119
Figure 52. 16-bit Timer 2 Block Diagram .....	120
Figure 53. 16-bit Timer/Counter Mode for Timer 3.....	125
Figure 54. 16-bit Timer/Counter 3 Example .....	125
Figure 55. 16-bit Capture Mode for Timer 3 .....	126
Figure 56. Input Capture Mode Operation for Timer 3 .....	127
Figure 57. Express Timer Overflow in Capture Mode .....	127
Figure 58. 16-bit PPG Mode for Timer 3 .....	128
Figure 59. 16-bit PPG Mode Timming chart for Timer 3 .....	129
Figure 60. 16-Bit Timer 3 Block Diagram .....	130
Figure 61. 16-bit Timer/Counter Mode for Timer 4.....	134
Figure 62. 16-bit Timer/Counter 4 Example .....	135
Figure 63. 16-bit Capture Mode for Timer 4 .....	136
Figure 64. Input Capture Mode Operation for Timer 4 .....	137
Figure 65. Express Timer Overflow in Capture Mode .....	137
Figure 66. 16-bit PPG Mode for Timer 4 .....	138
Figure 67. 16-bit PPG Mode Timming chart for Timer 4 .....	139
Figure 68. 16-bit Timer 4 Block Diagram .....	140
Figure 69. 16-bit Timer/Counter Mode for Timer 5.....	144
Figure 70. 16-bit Timer/Counter 4 Example .....	144
Figure 71. 16-bit Capture Mode for Timer 5 .....	145
Figure 72. Input Capture Mode Operation for Timer 5.....	146
Figure 73. Express Timer Overflow in Capture Mode .....	146
Figure 74. 16-bit PPG Mode for Timer 5 .....	147
Figure 75. 16-bit PPG Mode Timming chart for Timer 5 .....	148
Figure 76. 16-bit Timer 5 Block Diagram .....	149
Figure 77. Buzzer Driver Block Diagram.....	152
Figure 78. USIn USART Block Diagram(n = 0, 1).....	156
Figure 79. Clock Generation Block Diagram (USIn) .....	157
Figure 80. Synchronous Mode SCKn Timing (USIn) .....	158
Figure 81. Frame Format (USIn).....	159
Figure 82. Asynchronous Start Bit Sampling (USIn).....	163
Figure 83. Asynchronous Sampling of Data and Parity Bit (USIn).....	164
Figure 84. Stop Bit Sampling and Next Start Bit Sampling (USIn) .....	164
Figure 85. USIn SPI Clock Formats when CPHAn=0 .....	166
Figure 86. USIn SPI Clock Formats when CPHAn=1 .....	167
Figure 87. USIn SPI Block Diagram(n = 0, 1) .....	168
Figure 88. Bit Transfer on the I2C-Bus (USIn).....	169
Figure 89. START and STOP Condition (USIn) .....	170
Figure 90. Data Transfer on the I2C-Bus (USIn).....	170
Figure 91. Acknowledge on the I2C-Bus (USIn) .....	171
Figure 92. Clock Synchronization during Arbitration Procedure (USIn).....	172
Figure 93. Arbitration Procedure of Two Masters (USIn) .....	172
Figure 94. USIn I2C Block Diagram .....	179
Figure 95. USART1 Block Diagram .....	193
Figure 96. Clock Generation Block Diagram.....	194
Figure 97. Synchronous Mode XCK Timing .....	195
Figure 98. Start bit Sampling.....	200
Figure 99. Sampling of Data and Parity bit .....	200
Figure 100. Stop bit Sampling and Next Start bit Sampling .....	201

Figure 101. SPI Clock Formats when UCPHA=0 .....	202
Figure 102. SPI Clock Formats when UCPHA=1 .....	203
Figure 103. Example for RTO in USART1 .....	204
Figure 104. 0% Error Baud Rate Block Diagram .....	214
Figure 105. LED Driver Block Diagram .....	215
Figure 106. T-Type Matrix .....	226
Figure 107. M-Type Matrix .....	227
Figure 108. Circuit for LED and Touch by common pin .....	228
Figure 109. LED auto Mode .....	229
Figure 110. LED alone Mode .....	230
Figure 111. Hand Shake Mode .....	231
Figure 112. LED Stop Count Mode .....	232
Figure 113. Smart Share Mode .....	233
Figure 114. 16-bit Touch Block Diagram .....	235
Figure 115. CAPN Pin with Cs Capacitor .....	235
Figure 116. User Programming Procedure .....	250
Figure 117. 12-bit ADC Block Diagram .....	252
Figure 118. A/D Analog Input Pin with Capacitor .....	252
Figure 119. A/D Power (AVREF) Pin with Capacitor .....	252
Figure 120. ADC Operation for Align Bit .....	253
Figure 121. A/D Converter Operation Flow .....	254
Figure 122. IDLE Mode Release Timing by External Interrupt .....	259
Figure 123. STOP Mode Release Timing by External Interrupt .....	260
Figure 124. STOP Mode Release Flow .....	261
Figure 125. RESET Block Diagram .....	263
Figure 126. Fast VDD Rising Time .....	264
Figure 127. Internal RESET Release Timing On Power-Up .....	264
Figure 128. Configuration Timing when Power-on .....	265
Figure 129. Boot Process WaveForm .....	265
Figure 130. Timing Diagram after RESET .....	267
Figure 131. Oscillator generating waveform example .....	267
Figure 132. Block Diagram of LVR .....	268
Figure 133. Internal Reset at the power fail situation .....	268
Figure 134. Configuration timing when LVR RESET .....	269
Figure 135. LVI Diagram .....	270
Figure 136. Read device internal CkeckSum(Full size) .....	279
Figure 137. Read device internal CkeckSum(User define size) .....	279
Figure 138. Flash Memory Map .....	281
Figure 139. Address Configuration of Flash Memory .....	281
Figure 140. The Sequence of Page Program and Erase of Flash Memory .....	283
Figure 141. The Sequence of Bulk Erase of Flash Memory .....	284
Figure 142. ISP Mode .....	289
Figure 143. AC Timing .....	297
Figure 144. Waveform for USART Timing Characteristics .....	298
Figure 145. Timing Waveform for the USART Module .....	298
Figure 146. SPI0/1/2 Timing .....	299
Figure 147. Waveform for UART0 Timing Characteristics .....	300
Figure 148. Timing Waveform for the UART0 Module .....	300
Figure 149. I2C0 Timing .....	301
Figure 150. Stop Mode Release Timing when Initiated by an Interrupt .....	302
Figure 151. Stop Mode Release Timing when Initiated by RESETB .....	302

Figure 152. Crystal Oscillator.....	304
Figure 153. External Clock.....	304
Figure 154. Clock Timing Measurement at SXIN.....	305
Figure 155. Operating Voltage Range .....	305
Figure 156. Recommended Voltage Range.....	306
Figure 157. RUN (IDD1) Current .....	307
Figure 158. IDLE (IDD2) Current .....	307
Figure 159. STOP1 (IDD3) Current.....	308
Figure 160. STOP2 (IDD4) Current.....	308
Figure 161. Debugger(OCD1/OCD2) and pin description .....	309
Figure 162. E-PGMplus (Single writer) .....	310
Figure 163. PCB design guide for on board programming .....	312
Figure 164. E-GANG4 and E-GANG6 (for Mass Production).....	313
Figure 165. Block Diagram of On-Chip Debug System .....	314
Figure 166. 10-bit Transmission Packet.....	315
Figure 167. Data Transfer on the Twin Bus .....	316
Figure 168. Bit Transfer on the Serial Bus .....	316
Figure 169. Start and Stop Condition.....	316
Figure 170. Acknowledge on the Serial Bus .....	317
Figure 171. Clock Synchronization during Wait Procedure .....	317
Figure 172. Connection of Transmission .....	318
Figure 173. 28-Pin SOP Package .....	319
Figure 174. 28-Pin TSSOP Package .....	320
Figure 175. 24-Pin SOP Package .....	321
Figure 176. 20-Pin SOP Package .....	322
Figure 177. 20-Pin TSSOP Package .....	323
Figure 178. Device Nomenclature .....	325

## List of Tables

Table 1. A96T418 Device Features and Peripheral Counts .....	16
Table 1. A96T418 Device Features and Peripheral Counts (continued).....	17
Table 2. Normal Pin Description.....	24
Table 2. Normal Pin Description (continued).....	25
Table 2. Normal Pin Description (continued).....	26
Table 2. Normal Pin Description (continued).....	27
Table 2. Normal Pin Description (continued).....	28
Table 3. SFR Map Summary .....	38
Table 4. XSFR Map Summary .....	39
Table 5. SFR Map .....	40
Table 5. SFR Map (Continued).....	41
Table 5. SFR Map (Continued).....	42
Table 5. SFR Map (Continued).....	43
Table 6. XSFR Map .....	44
Table 6. XSFR Map (Continued) .....	45
Table 7. Port Register Map.....	49
Table 8. Interrupt Group Priority Level .....	64
Table 9. Interrupt Vector Address Table .....	67
Table 10. Interrupt Register Map.....	73
Table 11. Clock Generator Register Map .....	81
Table 12. Basic Interval Timer Register Map .....	85
Table 13. Watchdog Timer Register Map .....	88
Table 14. Watch Timer Register Map .....	91
Table 15. Timer 0 Operating Modes .....	93
Table 16. Timer 0 Register Map .....	100
Table 17. Timer 1 Operating Modes .....	103
Table 18. Timer 1 Register Map .....	109
Table 19. Timer 2 Operating Modes .....	113
Table 20. Timer 2 Register Map .....	120
Table 21. Timer 3 Operating Modes .....	124
Table 22. Timer 3 Register Map .....	130
Table 23. TIMER 4 Operating Modes .....	133
Table 24. Timer 4 Register Map .....	140
Table 25. TIMER 5 Operating Modes .....	143
Table 26. Timer 5 Register Map .....	149
Table 27. Buzzer Frequency at 16 MHz .....	152
Table 28. Buzzer Driver Register Map .....	153
Table 29. Equations for Calculating USIn Baud Rate Register Setting .....	157
Table 30. CPOLn Functionality.....	165
Table 31. USI Register Map .....	180
Table 32. Examples of USI0BD Settings for Commonly Used Oscillator Frequencies .....	190
Table 33. Examples of USI0BD Settings for Commonly Used Oscillator Frequencies (continued) ...	190
Table 33. Examples of USI0BD Settings for Commonly Used Oscillator Frequencies (continued) ...	191
Table 33. Equations for Calculating Baud Rate Register Setting.....	194
Table 34. CPOL Functionality.....	201
Table 35. USART1 Register Map .....	205
Table 36. Examples of UBAUD Settings for Commonly Used Oscillator Frequencies .....	213
Table 37. LED Driver Register Map .....	216
Table 38. LED RAM.....	225
Table 39. Recommended diodes .....	228

Table 40. Touch Register Map.....	236
Table 41. ADC Register Map.....	254
Table 42. Peripheral Operation during Power Down Mode.....	258
Table 43. Power Down Operation Register Map.....	262
Table 44. Reset State.....	263
Table 45. Boot Process Description .....	266
Table 46. Reset Operation Register Map.....	270
Table 47. Register Map .....	274
Table 48. Program/erase Time .....	280
Table 49. Operation Mode .....	288
Table 50. Security policy using lock-bits .....	290
Table 51. Absolute Maximum Ratings .....	292
Table 52. Recommended Operating Conditions .....	292
Table 53. Touch Switch Characteristics .....	293
Table 54. A/D Converter Characteristics .....	293
Table 55. Power-on Reset Characteristics.....	293
Table 56. LVR and LVI Characteristics.....	294
Table 57. High Internal RC Oscillator Characteristics .....	295
Table 58. Internal WDTRC Oscillator Characteristics .....	295
Table 59. DC Characteristics.....	296
Table 60. AC Characteristics .....	297
Table 61. USART Characteristics.....	298
Table 62. SPI0 Characteristics .....	299
Table 63. UART0 Characteristics .....	300
Table 64. I2C0 Characteristics .....	301
Table 65. Data Retention Voltage in Stop Mode .....	302
Table 66. Internal Flash Rom Characteristics .....	303
Table 67. Input / Output Capacitance.....	303
Table 68. Sub Clock Oscillator Characteristics .....	304
Table 69. Sub Oscillation Stabilization Characteristics .....	305
Table 70. Descriptions of pins which are used to programming/reading the Flash .....	311
Table 71. Descriptions of pins which are used to programming/reading the Flash .....	314
Table 72. Ordering Information of A96T418 .....	324

# 1 Description

The A96T418 is an advanced CMOS 8-bit microcontroller with 32Kbytes of FLASH. This is powerful microcontroller which provides a highly flexible and cost effective solution to many embedded control applications.

## 1.1 Device overview

In this section, features of A96T418 and peripheral counts are introduced.

**Table 1. A96T418 Device Features and Peripheral Counts**

<b>Peripherals</b>		<b>Description</b>
Core	CPU	8-bit CISC core (M8051, 2 clocks per cycle)
	Interrupt	Up to 23 peripheral interrupts supported. <ul style="list-style-type: none"> <li>• EINT0 to 7, EINT8, EINT10, EINT11, EINT12 (5)</li> <li>• Timer (0/1/2/3/4/5) (6)</li> <li>• WDT (1)</li> <li>• BIT (1)</li> <li>• WT (1)</li> <li>• USART Rx/Tx (2)</li> <li>• USI 1-ch. *Rx/Tx/I2C (3)</li> <li>• Touch (1)</li> <li>• LED (1)</li> <li>• ADC (1)</li> <li>• LVI (1)</li> </ul>
Memory	ROM (FLASH) capacity	<ul style="list-style-type: none"> <li>• 32Kbytes FLASH with self-read and write capability</li> <li>• In-system programming (ISP)</li> <li>• Endurance: 10,000 times</li> </ul>
	IRAM	256Bytes
	XRAM	1792Bytes
Programmable pulse generation		<ul style="list-style-type: none"> <li>• Pulse generation (by T1/T2/T3/T4/T5)</li> <li>• 8-bit PWM (by T0)</li> <li>• 16-bit PPG (by T1/T2/T3/T4/T5)</li> </ul>
Buzzer		8-bit × 1-ch
Minimum instruction execution time		125ns (@ 16MHz, NOP Instruction)
Power down mode		<ul style="list-style-type: none"> <li>• STOP1 mode</li> <li>• STOP2 mode</li> <li>• IDLE mode</li> </ul>
General Purpose I/O (GPIO)		<ul style="list-style-type: none"> <li>• Normal I/O : 26 ports</li> <li>• High sink current port : 8 ports (P2[7:0])</li> </ul>

**Table 1. A96T418 Device Features and Peripheral Counts (continued)**

Peripherals		Description
Reset	Power on reset	Power on reset : 1.2V
	Low voltage reset	<ul style="list-style-type: none"> <li>• 15 levels detect</li> <li>• 1.61/1.77/1.88/2.00/2.13/2.28/2.46/2.68/2.81/3.06/3.21/3.56/3.73/3.91/4.25V</li> </ul>
Low voltage indicator		level detect: 1.88~4.25V with 13 Levels
Watch Timer (WT)		<ul style="list-style-type: none"> <li>• 14-bit x 1-ch</li> <li>• 3.91ms/0.25s/0.5s/1s/1min interval at 32.768KHz</li> </ul>
Timer/counter		<ul style="list-style-type: none"> <li>• Basic interval timer (BIT) 8-bit x 1-ch.</li> <li>• Watchdog Timer (WDT) 8-bit x 1-ch.</li> <li>• 8-bit x 1-ch (Timer 0)</li> <li>• 16-bit x 5-ch (Timer 1/2/3/4/5)</li> </ul>
Communication function	USI	<ul style="list-style-type: none"> <li>• USART + SPI + I2C</li> <li>• 8-bit USART x 1-ch, 8-bit SPI x 1-ch and I2C x 1-ch</li> </ul>
	USART	<ul style="list-style-type: none"> <li>• USART + SPI</li> <li>• 8-bit USART x 1-ch and 8-bit SPI x 1-ch</li> <li>• Receiver timer out (RTO)</li> <li>• 0% error baud rate</li> </ul>
12-bit A/D converter		8 Input channels
20-Ch Self Capacitive Touch Switch		<ul style="list-style-type: none"> <li>• Fast Initial Self-Calibration</li> <li>• Key detection mode : Single/Multi-mode</li> <li>• CS Immunity : 10V</li> <li>• 16-bit Sensing Resolutions</li> <li>• Support Touch Key Int. Wakeup at Power down</li> </ul>
LED Driver		<ul style="list-style-type: none"> <li>• 8COM (250mA) / 16SEG (12.89mA)</li> <li>• T-type Max (8COM x 7SEG)</li> <li>• M-type Max (8COM x 8SEG)</li> <li>• Max LED (8COM x 15SEG) (T-type, M-type mix)</li> <li>• Constant Current Segment (4 level)</li> </ul>
Oscillator type		<ul style="list-style-type: none"> <li>• 2, 4, 8, 16MHz (internal RC oscillator)</li> <li>• 32.768kHz Crystal for sub clock</li> </ul>
Internal RC oscillator		<ul style="list-style-type: none"> <li>• HSIRC 16MHz <math>\pm</math>1.5% (TA=0~ +50°C)</li> <li>• HSIRC 16MHz <math>\pm</math>2.0% (TA=-10~ +70°C)</li> <li>• HSIRC 16MHz <math>\pm</math>2.5% (TA=-40~ +85°C)</li> <li>• HSIRC 16MHz <math>\pm</math>2.5% (TA=-40~ +105°C)</li> <li>• LSIRC 128kHz <math>\pm</math>20% (TA= -40~ +85°C)</li> <li>• LSIRC 128kHz <math>\pm</math>20% (TA= -40~ +105°C)</li> </ul>
Operating voltage and frequency		<ul style="list-style-type: none"> <li>• 2.0V~ 5.5V @ 16MHz</li> <li>• 2.7V~ 5.5V (Touch, ADC, LED Driver)</li> </ul>
Operating temperature		-40°C to +85°C, -40°C to +105°C
Package		<ul style="list-style-type: none"> <li>• 28 SOP</li> <li>• 28 TSSOP</li> <li>• 24 SOP</li> <li>• 20 SOP</li> <li>• 20 TSSOP</li> <li>• Pb-free packages</li> </ul>

## 1.2 Block Diagram

In this section, A96T418 devices and peripheral are described in a block diagram.

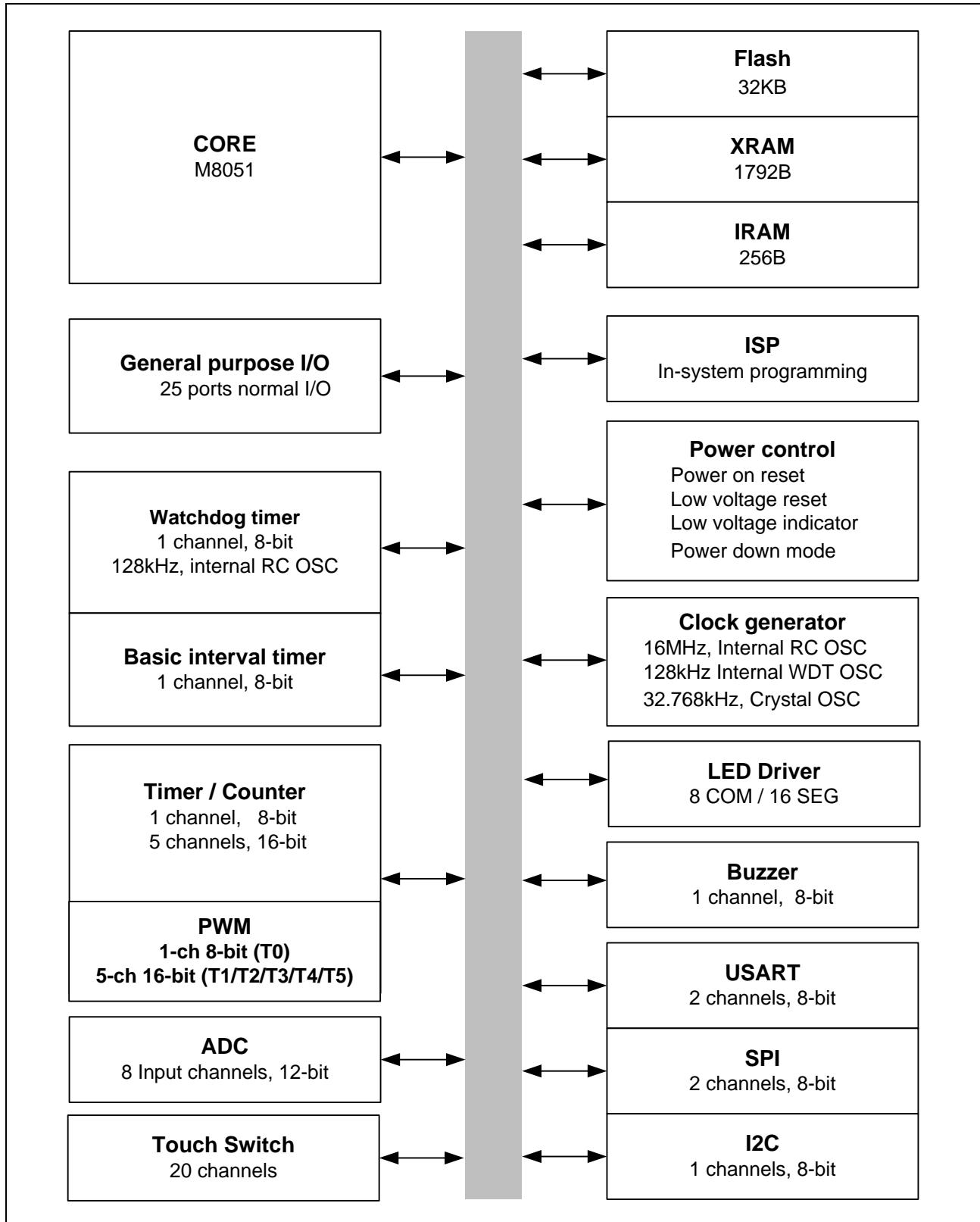
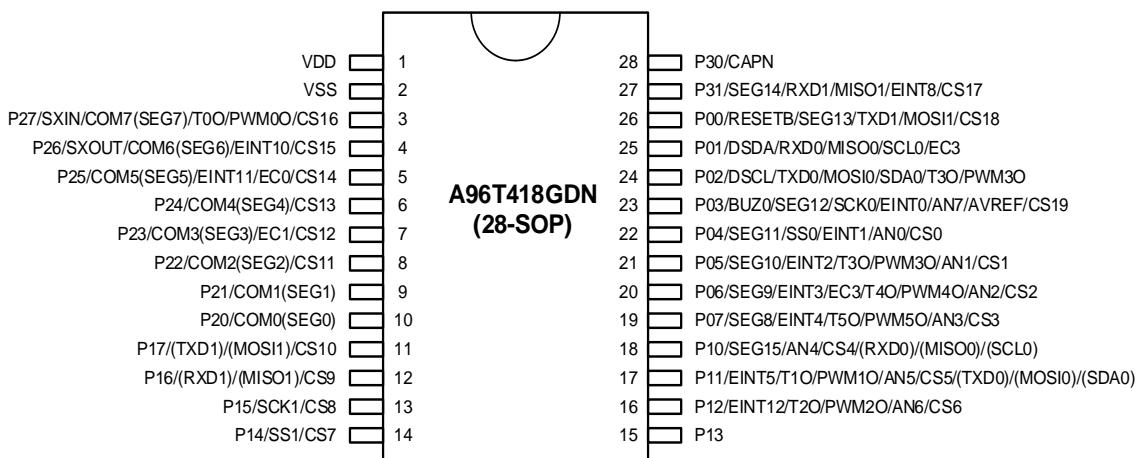


Figure 1. Block diagram of A96T418

## 2 Pinouts and pin description

Pinouts and pin descriptions of A96T418 device are introduce in the following sections.

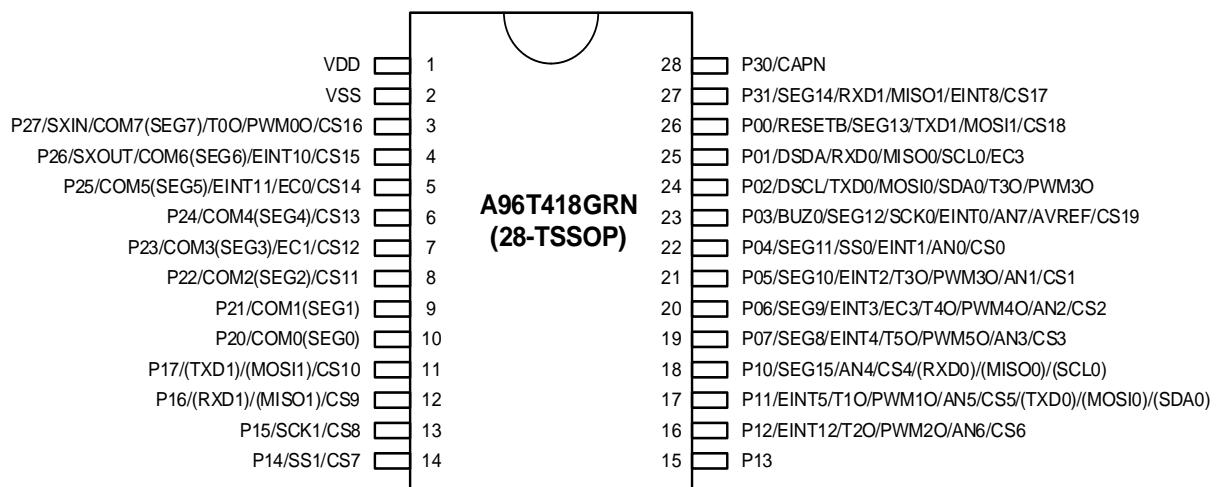
### 2.1 Pinouts



#### NOTES:

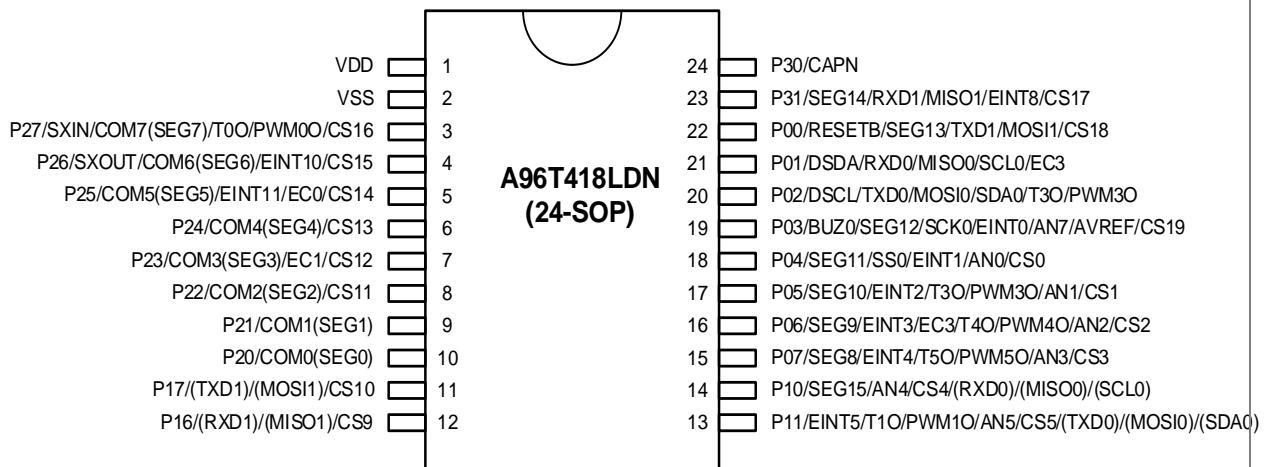
1. The programmer (E-PGM+, E-Gang4/6) uses P0[2:1] pin as DSCL, DSDA.

Figure 2. A96T418 28SOP pin assignment

**NOTES:**

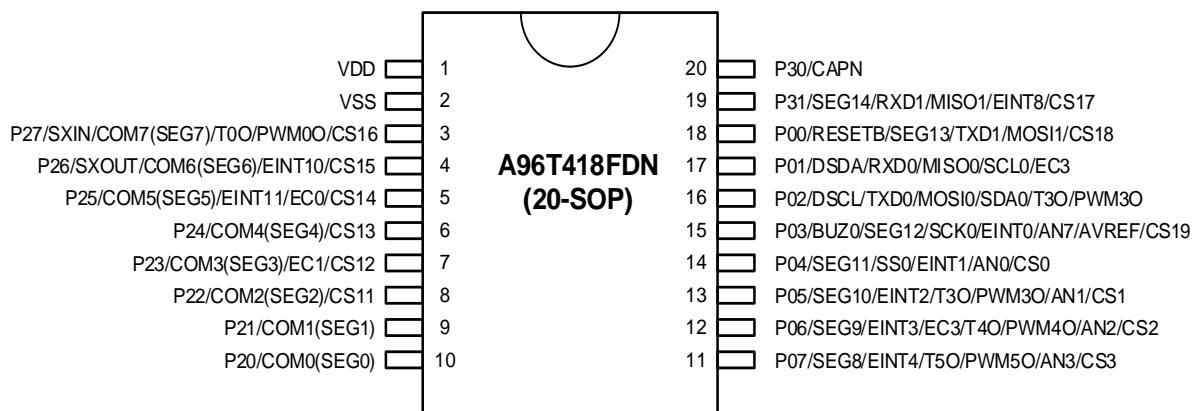
1. The programmer (E-PGM+, E-Gang4/6) uses P0[2:1] pin as DSCL, DSDA.

**Figure 3. A96T418 28TSSOP pin assignment**

**NOTES:**

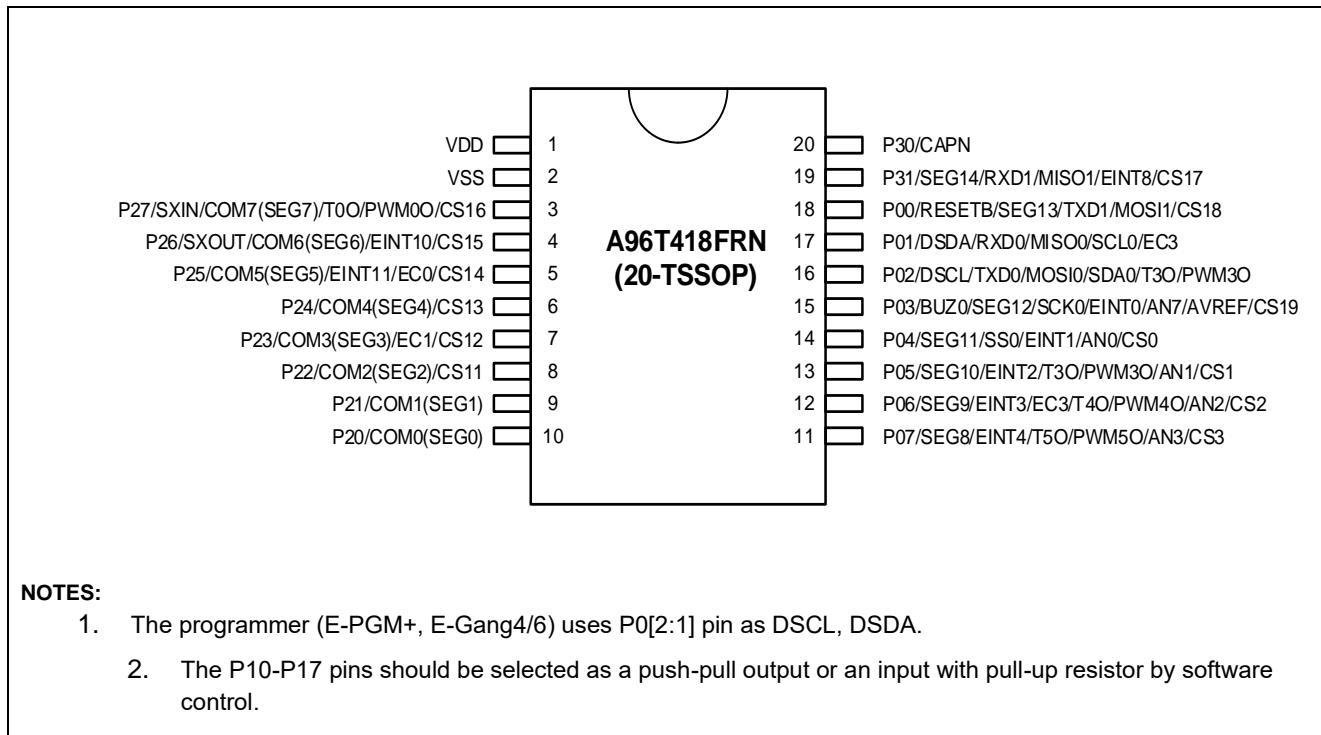
1. The programmer (E-PGM+, E-Gang4/6) uses P0[2:1] pin as DSCL, DSDA.
2. The P12-P15 pins should be selected as a push-pull output or an input with pull-up resistor by software control.

**Figure 4. A96T418 24SOP pin assignment**

**NOTES:**

1. The programmer (E-PGM+, E-Gang4/6) uses P0[2:1] pin as DSCL, DSDA.
2. The P10-P17 pins should be selected as a push-pull output or an input with pull-up resistor by software control.

**Figure 5. A96T418 20SOP pin assignment**

**Figure 6. A96T418 20TSSOP pin assignment**

## 2.2 Pin Description

**Table 2. Normal Pin Description**

PIN Name	I/O	Function	@RESET	Shared with
P00	I/O	Port P0 8-Bit I/O Port Can be set in input or output mode in 1-bit units Internal pull-up register can be used via software when this port is used as input port Open Drain enable register can be used via software when this port is used as output port	Input	RESETB/SEG13/TXD1/MOSI1/CS18
P01				DSDA/RXD0/MISO0/SCL0/EC3
P02				DSCL/TXD0/MOSI0/SDA0/T3O/PWM3O
P03				BUZ0/SEG12/SCK0/EINT0/AN7/AVREF/CS19
P04				SEG11/SS0/EINT1/AN0/CS0
P05				SEG10/EINT2/T3O/PWM3O/AN1/CS1
P06				SEG9/EINT3/EC3/T4O/PWM4O/AN2/CS2
P07				SEG8/EINT4/T5O/PWM5O/AN3/CS3
P10	I/O	Port P1 8-Bit I/O Port Can be set in input or output mode in 1-bit units Internal pull-up register can be used via software when this port is used as input port Open Drain enable register can be used via software when this port is used as output port. The P12 – P15 are not in the 24-pin package. The P10 – P17 are not in the 20-pin package.	Input	SEG15/AN4/CS4/(RXD0)/(MISO0)/(SCL0)
P11				EINT5/T1O/PWM1O/AN5/CS5/(TXD0)/(MOSI0)/(SDA0)
P12				EINT12/T2O/PWM2O/AN6/CS6
P13				
P14				SS1/CS7
P15				SCK1/CS8
P16				(RXD1)/(MISO1)/CS9
P17				(TXD1)/(MOSI1)/CS10
P20	I/O	Port P2 8-Bit I/O Port Can be set in input or output mode in 1-bit units Internal pull-up register can be used via software when this port is used as input port Open Drain enable register can be used via software when this port is used as output port.	Input	COM0(SEG0)
P21				COM1(SEG1)
P22				COM2(SEG2)/CS11
P23				COM3(SEG3)/EC1/CS12
P24				COM4(SEG4)/CS13
P25				COM5(SEG5)/EINT11/EC0/CS14
P26				SXOUT/COM6(SEG6)/EINT10/CS15
P27				SXIN/COM7(SEG7)/T0O/PWM0O/CS16

**Table 2. Normal Pin Description (continued)**

PIN Name	I/O	Function	@RESET	Shared with
P30	I/O	Port P3 2-Bit I/O Port Can be set in input or output mode in 1-bit units Internal pull-up register can be used via software when this port is used as input port	Input	CAPN
P31		Open Drain enable register can be used via software when this port is used as output port. The P31 is not in the package with VDDLED pin.		SEG14/RXD1/MISO1/EINT8/CS17
EINT0	I/O	External interrupt inputs	Input	P03/BUZ0/SEG12/SCK0/AN7/AVREF/CS19
EINT1				P04/SEG11/SS0/AN0/CS0
EINT2	I/O	External interrupt input and Timer 3 capture input	Input	P05/SEG10/T3O/PWM3O/AN1/CS1
EINT3	I/O	External interrupt input and Timer 4 capture input	Input	P06/SEG9/EC3/T4O/PWM4O/AN2/CS2
EINT4	I/O	External interrupt input and Timer 5 capture input	Input	P07/SEG8/T5O/PWM5O/AN3/CS3
EINT5	I/O	External interrupt input and Timer 1 capture input	Input	P11/T1O/PWM1O/AN5/CS5/(TXD0)/(MOSI0)/(SDA0)
EINT8	I/O	External Interrupt input	Input	P31/SEG14/RXD1/MISO1/CS17
EINT10	I/O	External interrupt input	Input	P26/SXOUT/COM6(SEG6)/CS15
EINT11	I/O	External interrupt input	Input	P25/COM5(SEG5)/EC0/CS14
EINT12	I/O	External interrupt input and Timer 2 capture input	Input	P12/T2O/PWM2O/AN6/CS6
T0O	I/O	Timer 0 interval output	Input	P27/SXIN/COM7(SEG7)/PWM0O/CS16
T1O	I/O	Timer 1 interval output	Input	P11/EINT5/PWM1O/AN5/CS5/(TXD0)/(MOSI0)/(SDA0)
T2O	I/O	Timer 2 interval output	Input	P12/EINT12/PWM2O/AN6/CS6

**Table 2. Normal Pin Description (continued)**

PIN Name	I/O	Function	@RESET	Shared with
T3O	I/O	Timer 3 interval output	Input	P02/DSCL/TXD0/MOSI0/SDA0/PWM3O P05/ SEG10/EINT2/PWM3O/AN1/CS1
T4O	I/O	Timer 4 interval output	Input	P06/SEG9/EINT3/EC3/PWM4O/AN2/CS2
T5O	I/O	Timer 5 interval output	Input	P07/SEG8/EINT4/PWM5O/AN3/CS3
PWM0O	I/O	Timer 0 PWM output	Input	P27/SXIN/T0O/COM7/CS16
PWM1O	I/O	Timer 1 PWM output	Input	P11/EINT5/T1O/AN5/CS5/(TXD0)/(MOSI0)/(SDA0)
PWM2O	I/O	Timer 2 PWM output	Input	P12/EINT12/T2O/AN6/CS6
PWM3O	I/O	Timer 3 PWM output	Input	P02/DSCL/TXD0/MOSI0/SDA0/T3O P05/ SEG10/EINT2/T3O/AN1/CS1
PWM4O	I/O	Timer 4 PWM output	Input	P06/SEG9/EINT3/EC3/T4O/AN2/CS2
PWM5O	I/O	Timer 5 PWM output	Input	P07/SEG8/EINT4/T5O/AN3/CS3
EC0	I/O	Timer 0 event count input	Input	P25/COM5(SEG5)/EINT11/CS14
EC1	I/O	Timer 1 event count input	Input	P23/COM3(SEG3)/CS12
EC3	I/O	Timer 3 event count input	Input	P01/DSDA/RXD0/MISO0/SCL0 P06/SEG9/EINT3/T4O/PWM4O/AN2/CS2
BUZ0	I/O	Buzzer signal output	Input	P03/SEG12/SCK0/AN7/EINT0/AVREF/CS19
SCK0	I/O	Serial 0 clock input/output	Input	P03/BUZ0/SEG12/AN7/EINT0/AVREF/CS19
SCK1	I/O	Serial 1 clock input/output	Input	P15/CS8
MOSI0	I/O	SPI 0 master output, slave input	Input	P02/DSCL/TXD0/SDA0/T3O/PWM3O P11/EINT5/T1O/PWM1O/AN5/CS5/(TXD0)/(SDA0)
MOSI1	I/O	SPI 1 master output, slave input	Input	P00/RESETB/SEG13/TXD1/CS18 P17/(TXD1)/CS10
MISO0	I/O	SPI 0 master input, slave output	Input	P01/DSDA/RXD0/SCL0/EC3 P10/SEG15/AN4/CS4/(RXD0)/(SCL0)
MISO1	I/O	SPI 1 master input, slave output	Input	P16/(RXD1)/CS9 P31/SEG14/RXD1/EINT8/CS17
SS0	I/O	SPI 0 slave select input (Slave mode only)	Input	P04/SEG11/SS0/EINT1/AN0/CS0
SS1	I/O	SPI 1 slave select input (Slave mode only)	Input	P14/CS7
TXD0	I/O	UART 0 data output	Input	P02/DSCL/MOSI0/SDA0/T3O/PWM3O P11/EINT5/T1O/PWM1O/AN5/CS5/(MOSI0)/(SDA0)
TXD1	I/O	UART 1 data output	Input	P00/RESETB/SEG13/MOSI1/CS18 P17/(MOSI1)/CS10
RXD0	I/O	UART 0 data input	Input	P01/DSDA/MISO0/SCL0/EC3 P10/SEG15/AN4/CS4/(MISO0)/(SCL0)
RXD1	I/O	UART 1 data input	Input	P16/(MISO1)/CS9 P31/SEG14/MISO1/EINT8/CS17

**Table 2. Normal Pin Description (continued)**

PIN Name	I/O	Function	@RESET	Shared with
SCL0	I/O	I2C 0 clock input/output	Input	P01/DSDA/RXD0/MISO0/EC3 P10/SEG15/AN4/CS4/(RXD0)/(MISO0)
SDA0	I/O	I2C 0 data input/output	Input	P02/DSCL/TXD0/MOSI0/T3O/PWM3O P11/EINT5/T1O/PWM1O/AN5/CS5/(TXD0)/(MOSI0)
AVREF	I/O	A/D converter reference voltage	Input	P03/BUZ0/SEG12/SCK0/AN7/EINT0/CS19
AN0	I/O	A/D converter analog input channels	Input	P04/SEG11/SS0/EINT1/CS0
AN1				P05/SEG10/EINT2/T3O/PWM3O/CS1
AN2				P06/SEG9/EINT3/EC3/T4O/PWM4O/CS2
AN3				P07/SEG8/EINT4/T5O/PWM5O/CS3
AN4				P10/SEG15/CS4/(RXD0)/(MISO0)/(SCL0)
AN5				P11/EINT5/T1O/PWM1O/CS5/(TXD0)/(MOSI0)/(SDA0)
AN6				P12/EINT12/T2O/PWM2O/CS6
AN7				P03/BUZ0/SEG12/SCK0/AVREF/EINT0/CS19
COM0	I/O	LED COM ports	Input	P20/(SEG0)
COM1				P21/(SEG1)
COM2				P22/(SEG2)/CS11
COM3				P23/(SEG3)EC1/CS12
COM4				P24/(SEG4)/CS13
COM5				P25/(SEG5)/EINT11/EC0/CS14
COM6				P26/(SEG6)/SXOUT/EINT10/CS15
COM7				P27/(SEG7)SXIN/T0D/PWM0O/CS16
SEG0	I/O	LED SEG ports	Input	P20/(COM0)
SEG1				P21/(COM1)
SEG2				P22/(COM2)/CS11
SEG3				P23/(COM3)EC1/CS12
SEG4				P24/(COM4)/CS13
SEG5				P25/(COM5)/EINT11/EC0/CS14
SEG6				P26/(COM6)/SXOUT/EINT10/CS15
SEG7				P27/(COM7)SXIN/T0D/PWM0O/CS16
SEG8				P07/EINT4/T5O/PWM5O/AN3/CS3
SEG9				P06/EINT3/EC3/T4O/PWM4O/AN2/CS2
SEG10				P05/EINT2/T3O/PWM3O/AN1/CS1
SEG11				P04/SS0/EINT1/AN0/CS0
SEG12				P03/BUZ0/SCK0/EINT0/AN7/AVREF/CS19
SEG13				P00/RESETB/TXD1/MOSI1/CS18
SEG14				P31/RXD1/MISO1/EINT8/CS17
CAPN	I/O	Modulation CAP	Input	P30

**Table 2. Normal Pin Description (continued)**

PIN Name	I/O	Function	@RESET	Shared with
CS0	I/O Touch Switch Inputs		Input	P04/SEG11/SS0/EINT1/AN0
CS1				P05/SEG10/EINT2/T3O/PWM3O/AN1
CS2				P06/SEG9/EINT3/EC3/T4O/PWM4O/AN2
CS3				P07/SEG8/EINT4/T5O/PWM5O/AN3
CS4				P10/SEG15/AN4/(RXD0)/(MISO0)/(SCL0)
CS5				P11/EINT5/T1O/PWM1O/AN5/(TXD0)/(MOSI0)/(SDA0)
CS6				P12/EINT12/T2O/PWM2O/AN6
CS7				P14/SS1
CS8				P15/SCK1
CS9				P16/(RXD1)/(MISO1)
CS10				P17/(TXD1)/(MOSI1)
CS11				P22/COM2(SEG2)
CS12				P23/COM3(SEG3)/EC1
CS13				P24/COM4(SEG4)
CS14				P25/COM5(SEG5)/EINT11/EC0
CS15				P26/SXOUT/COM6(SEG6)/EINT10
CS16				P27/SXIN/COM7(SEG7)/T0O/PWM0O
CS17				P31/SEG14/RXD1/MISO1/EINT8
CS18				P00/RESETB/SEG13/TXD1/MOSI1
CS19				P03/BUZ0/SEG12/SCK0/EINT0/AN7/AVREF
RESETB	I/O	System reset pin with a pull-up resistor when it is selected as the RESETB by CONFIGURE OPTION	Input	P00/SEG13/TXD1/MOSI1/CS18
DSDA	I/O	In-system programming data input/output	Input	P01/RXD0/MISO0/SCL0/EC3
DSCL	I/O	In-system programming clock input	Input	P02/TXD0/MOSI0/SDA0/T3O/PWM3O
SXIN	I/O	Sub oscillator pins	Input	P27/T0O/PWM0O/COM7(SEG7)/CS16
SXOUT				P26/COM6(SEG6)/EINT10/CS15
VDD, VSS	-	Power input pins	-	-
VDDLED	-	Power input pin for LED	-	-

**NOTES:**

1. The P12–P15 are not in the 24-pin package.
2. The P10–P17 are not in the 20-pin package.
3. The P00/RESETB pin is configured as one of the P00 and RESETB pin by the “CONFIGURE OPTION.”
4. If the P01/EC3/MISO0/RXD0/SCL0/DSDA and P02/T3O/PWM3O/MOSI0/TXD0/SDA0/DSCL pins are connected to the programmer during power-on reset, the pins are automatically configured as In-system programming pins.
5. The P01/EC3/MISO0/RXD0/SCL0/DSDA and P02/T3O/PWM3O/MOSI0/TXD0/SDA0/DSCL pins are configured as inputs with internal pull-up resistor only during the reset or power-on reset.
6. The P00/RESETB/SEG13/TXD1/MOSI1/CS18, P27/T0O/PWM0O/COM7/CS16/SXIN, and P26/COM6/EINT10/CS15/SXOUT pins are configured as a function pin by software control.

### 3 Port Structures

#### 3.1 General Purpose I/O Port

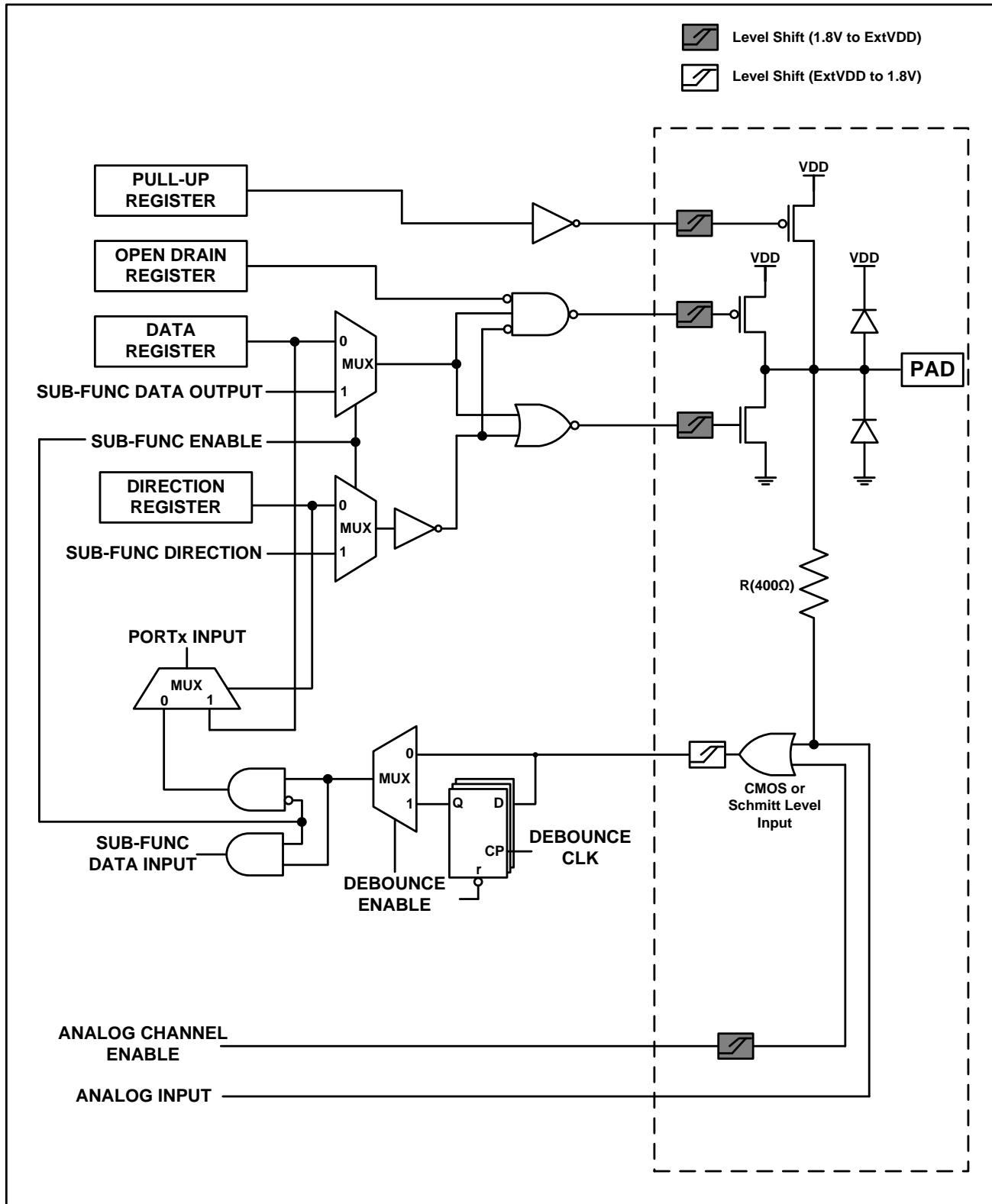


Figure 7. General Purpose I/O Port

### 3.2 SEG I/O Port

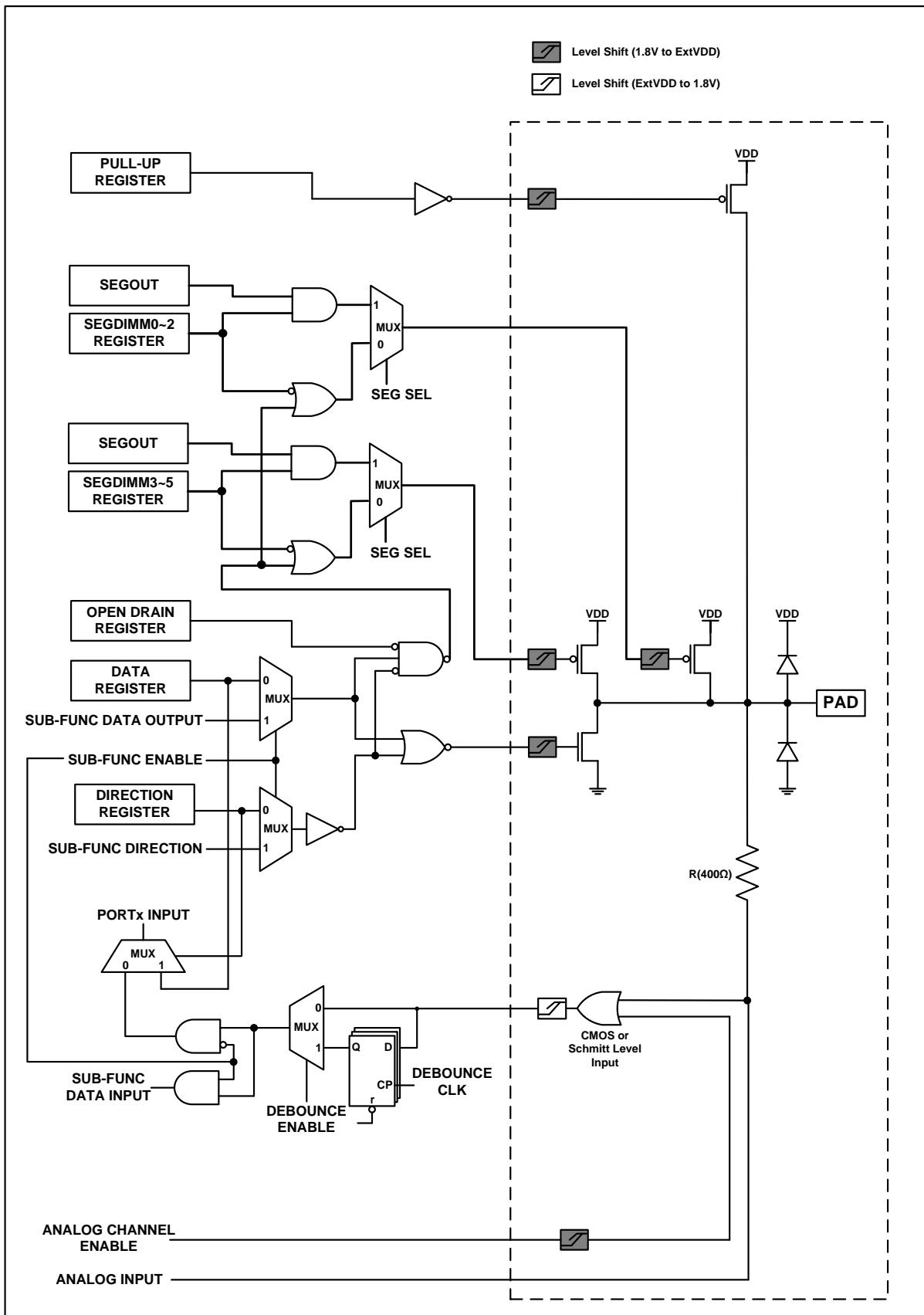


Figure 8. SEG I/O Port

### 3.3 COM & SEG I/O Port

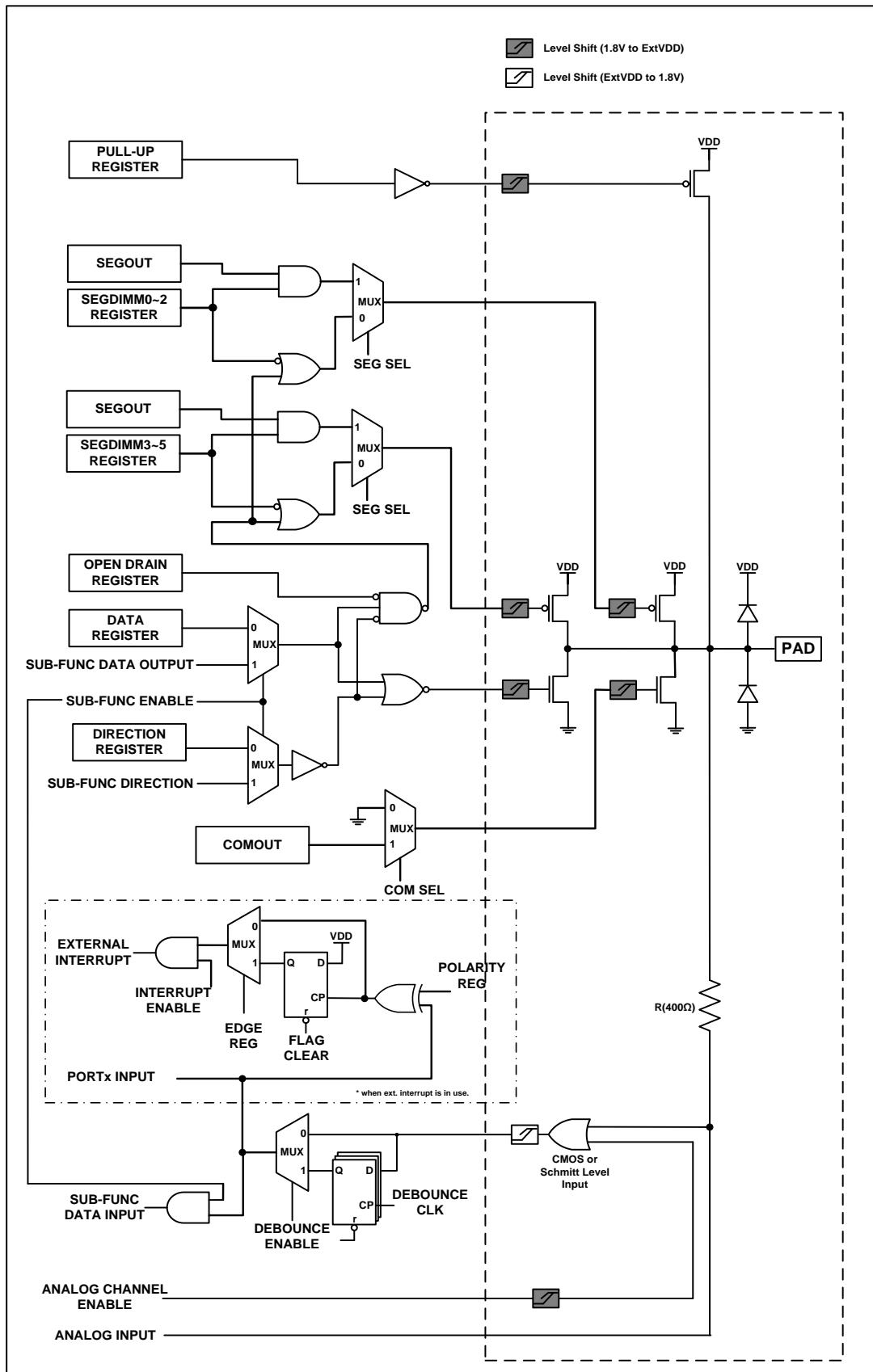


Figure 9. COM & SEG I/O Port

## 4 Memory organization

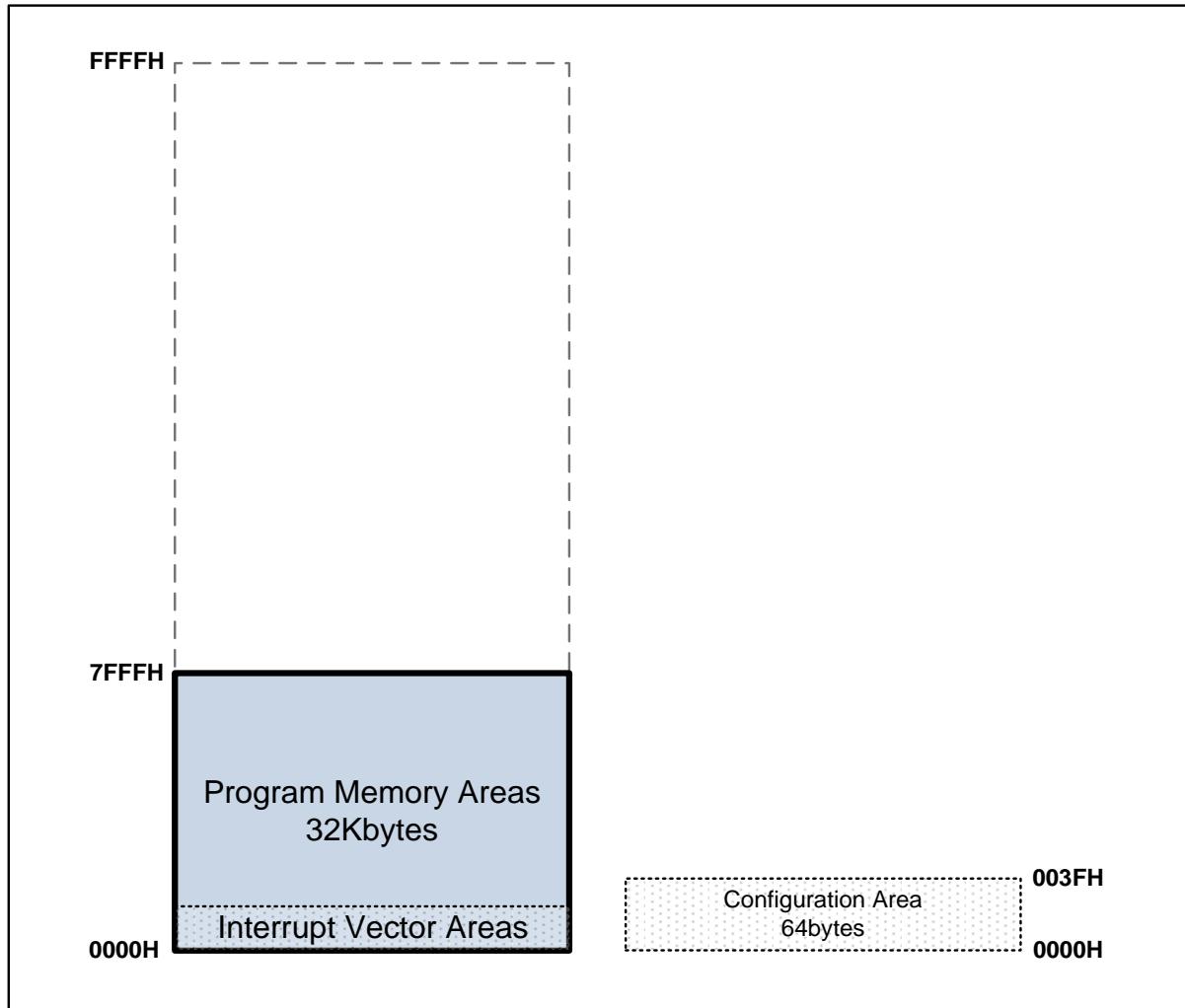
The A96T418 addresses two separate address memory stores: Program memory and Data memory. The logical separation of Program and Data memory allows Data memory to be accessed by 8-bit addresses, which makes the 8-bit CPU access the data memory more rapidly. Nevertheless, 16-bit Data memory addresses can also be generated through the DPTR register.

A96T418 provides on-chip 32Kbytes of the ISP type flash program memory, which can be read and written to. Internal data memory (IRAM) is 256bytes and it includes the stack area. External data memory (XRAM) is 1792bytes.

### 4.1 Program Memory

A 16-bit program counter is capable of addressing up to 64Kbytes, but this device has just 32Kbytes program memory space.

Figure 10 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 3, for example, is assigned to location 001BH. If external interrupt 3 is going to be used, its service routine must begin at location 001BH. 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 8bytes interval. Longer service routines can use a jump instruction to skip over subsequent interrupt locations, if other interrupts are in use.



**Figure 10. Program Memory**

**NOTES:**

1. 32Kbytes Including Interrupt Vector Region

## 4.2 Data Memory

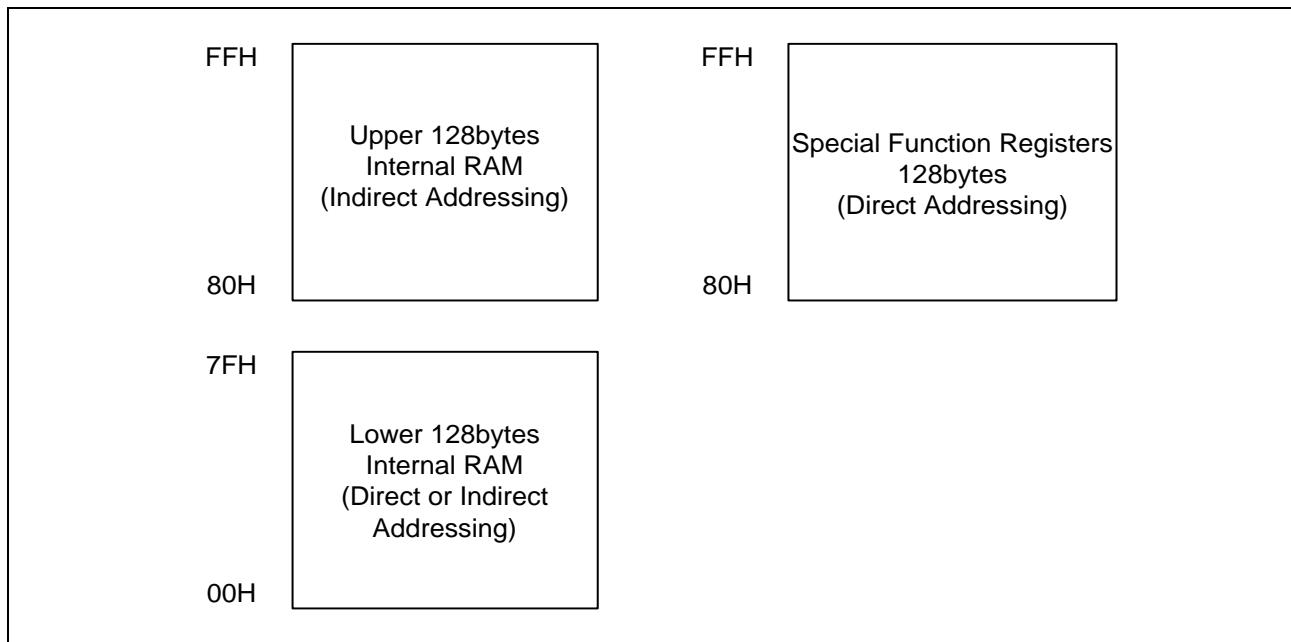


Figure 11. Data Memory Map

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

Internal data memory addresses are always one byte wide, which implies an address space of only 256bytes. However, in fact, the addressing modes for internal RAM can accommodate up to 384bytes 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 11 shows the upper 128bytes and SFR space occupying the same block of addresses, 80H through FFH, although they are physically separate entities.

The lower 128bytes of RAM are present in all 8051 devices as mapped in Figure 12. The lowest 32bytes 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 16bytes 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 128bytes can be accessed by either direct or indirect addressing. The upper 128bytes RAM can only be accessed by indirect addressing. These spaces are used for data RAM and stack.

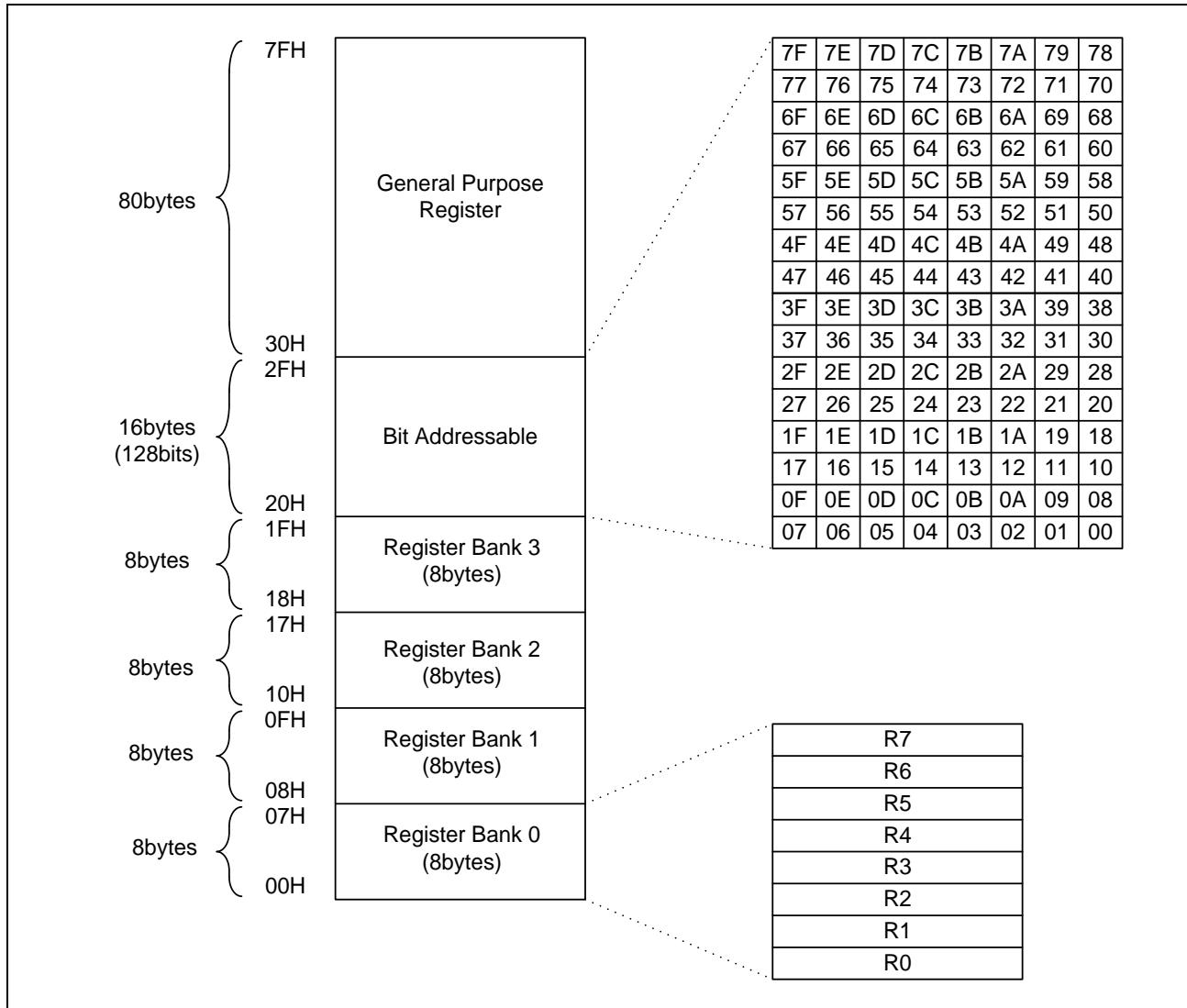


Figure 12. Lower 128bytes RAM

### 4.3 External Data Memory

A96T418 has 1792bytes XRAM and XSFR. This area has no relation with RAM/FLASH. It can be read and written to through SFR with 8-bit unit.

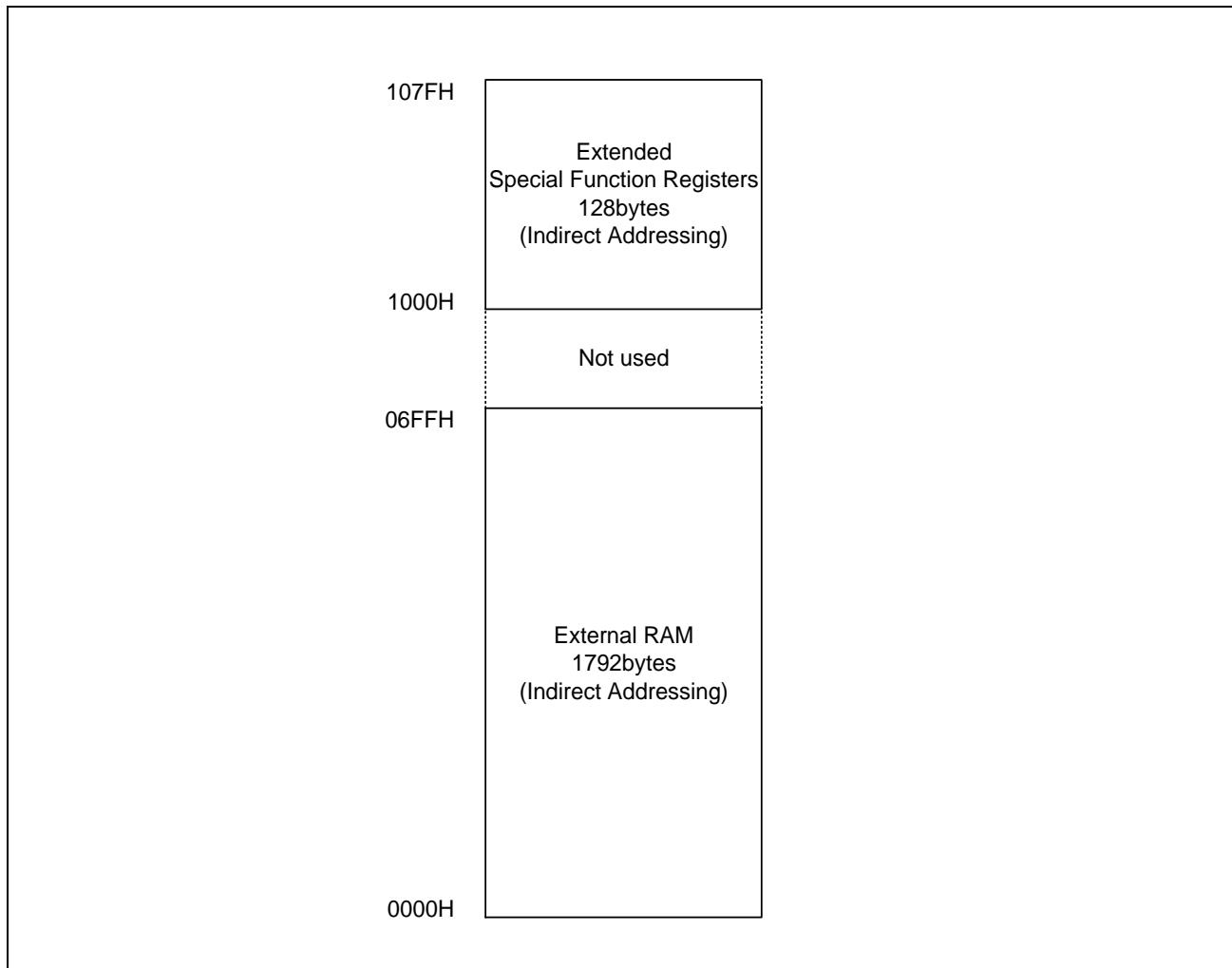


Figure 13. XDATA Memory Area

## 4.4 SFR Map

### 4.4.1 SFR Map Summary

**Table 3. SFR Map Summary**

	-	Reserved
		M8051 compatible

	00H/8H <sup>(1)</sup>	01H/9H	02H/0AH	03H/0BH	04H/0CH	05H/0DH	06H/0EH	07H/0FH
<b>0F8H</b>	IP1	–	–	–	UBAUD	UDATA	–	–
<b>0F0H</b>	B	–	–	–	–	–	–	–
<b>0E8H</b>	RSTFR	–	–	–	–	–	P3FSR	–
<b>0E0H</b>	ACC	USI0ST1	USI0ST2	USI0BD	USI0SDHR	USI0DR	USI0SCLR	USI0SCHR
<b>0D8H</b>	LVRCR	USI0CR1	USI0CR2	USI0CR3	USI0CR4	USI0SAR	P0DB	P123DB
<b>0D0H</b>	PSW	–	P0FSRL	P0FSRH	P1FSRL	P1FSRH	P2FSRL	P2FSRH
<b>0C8H</b>	OSCCR	–	–	UCTRL1	UCTRL2	UCTRL3	–	USTAT
<b>0C0H</b>	EIFLAG0	P3IO	T2CRL	T2CRH	T2ADRL	T2ADRH	T2BDRL	T2BDRH
<b>0B8H</b>	IP	P2IO	T1CRL	T1CRH	T1ADRL	T1ADRH	T1BDRL	T1BDRH
<b>0B0H</b>	–	P1IO	T0CR	T0CNT	T0DR/ T0CDR	–	–	–
<b>0A8H</b>	IE	IE1	IE2	IE3	P0PU	P1PU	P2PU	P3PU
<b>0A0H</b>	–	P0IO	EO	–	EIPOL0L	EIPOL0H	EIFLAG1	EIPOL1
<b>98H</b>	P3	–	–	–	ADCCRL	ADCCRH	ADCDRL	ADCDRH
<b>90H</b>	P2	P0OD	P1OD	P2OD	–	–	WTCR	BUZCR
<b>88H</b>	P1	WTDR/ WTCNT	SCCR	BITCR	BITCNT	WDTCR	WDTDR/ WDTCNT	BUZDR
<b>80H</b>	P0	SP	DPL	DPH	DPL1	DPH1	LVICR	PCON

**NOTES:**

1. 00H/8H, These registers are bit-addressable.

**Table 4. XSFR Map Summary**

	<b>00H/8H<sup>(1)</sup></b>	<b>01H/9H</b>	<b>02H/0AH</b>	<b>03H/0BH</b>	<b>04H/0CH</b>	<b>05H/0DH</b>	<b>06H/0EH</b>	<b>07H/0FH</b>
<b>1078H</b>	–	–	–	–	–	–	–	–
<b>1070H</b>	–	–	–	–	–	–	–	–
<b>1068H</b>	–	–	–	–	–	–	–	–
<b>1060H</b>	–	–	–	–	–	–	–	–
<b>1058H</b>	–	–	–	–	–	–	–	–
<b>1050H</b>	–	–	–	–	–	–	–	–
<b>1048H</b>	–	–	–	–	–	–	–	–
<b>1040H</b>	–	–	–	–	–	–	–	–
<b>1038H</b>	–	–	–	–	–	–	–	–
<b>1030H</b>	–	–	–	–	–	–	–	–
<b>1028H</b>	FEARH	FEARM	FEARL	FEDR	FETR	–	–	–
<b>1020H</b>	FEMR	FECR	FESR	FETCR	FEARM1	FEARL1	–	–
<b>1018H</b>	UCTRL4	FPCR	RTOCH	RTOCL	–	–	–	–
<b>1010H</b>	T5CRH	T5CRL	T5ADRH	T5ADRL	T5BDRH	T5BDRL	–	–
<b>1008H</b>	T4CRH	T4CRL	T4ADRH	T4ADRL	T4BDRH	T4BDRL	–	–
<b>1000H</b>	T3CRH	T3CRL	T3ADRH	T3ADRL	T3BDRH	T3BDRL	–	–

#### 4.4.2 SFR Map

**Table 5. SFR Map**

<b>Address</b>	<b>Function</b>	<b>Symbol</b>	<b>R/W</b>	@Reset							
				<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
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
8BH	Basic Interval Timer Control Register	BITCR	R/W	0	1	0	0	0	1	0	1
8CH	Basic Interval Timer Counter Register	BITCNT	R	0	0	0	0	0	0	0	0
8DH	Watchdog Timer Control Register	WDTCR	R/W	0	0	0	—	—	—	0	0
8EH	Watchdog Timer Data Register	WDTDR	W	1	1	1	1	1	1	1	1
	Watchdog 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	P0 Open-drain Selection Register	P0OD	R/W	0	0	0	0	0	0	0	0
92H	P1 Open-drain Selection Register	P1OD	R/W	0	0	0	0	0	0	0	0
93H	P2 Open-drain Selection Register	P2OD	R/W	0	0	0	0	0	0	0	0
94H	Reserved	—	—	—	—	—	—	—	—	—	—
95H	Reserved	—	—	—	—	—	—	—	—	—	—
96H	Watch Timer Control Register	WTCR	R/W	0	—	—	0	0	0	0	0
97H	BUZZER Control Register	BUZCR	R/W	—	—	—	—	—	0	0	0
98H	P3 Data Register	P3	R/W	—	—	—	—	—	—	0	0
99H	Reserved	—	—	—	—	—	—	—	—	—	—
9AH	Reserved	—	—	—	—	—	—	—	—	—	—
9BH	Reserved	—	—	—	—	—	—	—	—	—	—
9CH	A/D Converter Control Low Register	ADCCRL	R/W	0	0	0	0	—	0	0	0
9DH	A/D Converter Control High Register	ADCCRH	R/W	0	0	0	0	0	0	0	1
9EH	A/D Converter Data Low Register	ADCDRL	R	0	0	0	0	0	0	0	0
9FH	A/D Converter Data High Register	ADCDRH	R	0	0	0	0	0	0	0	0

**Table 5. SFR Map (Continued)**

<b>Address</b>	<b>Function</b>	<b>Symbol</b>	<b>R/W</b>	<b>@Reset</b>							
				<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
A0H	Reserved	—	—	—	—	—	—	—	—	—	—
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	Reserved	—	—	—	—	—	—	—	—	—	—
A4H	External Interrupt Polarity 0 Low Register	EIPOL0L	R/W	0	0	0	0	0	0	0	0
A5H	External Interrupt Polarity 0 High Register	EIPOL0H	R/W	—	—	—	—	0	0	0	0
A6H	External Interrupt Flag 1 Register	EIFLAG1	R/W	0	0	—	—	0	0	0	0
A7H	External Interrupt Polarity 1 Register	EIPOL1	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
AAH	Interrupt Enable Register 2	IE2	R/W	—	—	0	0	0	0	0	0
ABH	Interrupt Enable Register 3	IE3	R/W	—	—	0	0	0	0	0	0
ACH	P0 Pull-up Resistor Selection Register	P0PU	R/W	0	0	0	0	0	0	0	0
ADH	P1 Pull-up Resistor Selection Register	P1PU	R/W	0	0	0	0	0	0	0	0
AEH	P2 Pull-up Resistor Selection Register	P2PU	R/W	0	0	0	0	0	0	0	0
AFH	P3 Pull-up Resistor Selection Register	P3PU	R/W	—	—	—	—	—	—	0	0
B0H	Reserved	—	—	—	—	—	—	—	—	—	—
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	Reserved	—	—	—	—	—	—	—	—	—	—
B6H	Reserved	—	—	—	—	—	—	—	—	—	—
B7H	Reserved	—	—	—	—	—	—	—	—	—	—
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 Low Register	T1CRL	R/W	0	0	0	0	—	0	0	0
BBH	Timer 1 Counter High Register	T1CRH	R/W	0	—	0	0	—	—	—	0
BCH	Timer 1 A Data Low Register	T1ADRL	R/W	1	1	1	1	1	1	1	1
BDH	Timer 1 A Data High Register	T1ADRH	R/W	1	1	1	1	1	1	1	1
BEH	Timer 1 B Data Low Register	T1BDRL	R/W	1	1	1	1	1	1	1	1
BFH	Timer 1 B Data High Register	T1BDRH	R/W	1	1	1	1	1	1	1	1

**Table 5. SFR Map (Continued)**

<b>Address</b>	<b>Function</b>	<b>Symbol</b>	<b>R/W</b>	<b>@Reset</b>							
				<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
C0H	External Interrupt Flag 0 Register	EIFLAG0	R/W	-	-	0	0	0	0	0	0
C1H	P3 Direction Register	P3IO	R/W	-	-	-	-	-	-	0	0
C2H	Timer 2 Control Low Register	T2CRL	R/W	0	0	0	0	-	0	-	0
C3H	Timer 2 Control High Register	T2CRH	R/W	0	-	0	0	-	-	-	0
C4H	Timer 2 A Data Low Register	T2ADRL	R/W	1	1	1	1	1	1	1	1
C5H	Timer 2 A Data High Register	T2ADRH	R/W	1	1	1	1	1	1	1	1
C6H	Timer 2 B Data Low Register	T2BDRL	R/W	1	1	1	1	1	1	1	1
C7H	Timer 2 B Data High Register	T2BDRH	R/W	1	1	1	1	1	1	1	1
C8H	Oscillator Control Register	OSCCR	R/W	0	0	1	0	1	0	-	0
C9H	Reserved	-	-	-	-	-	-	-	-	-	-
CAH	Reserved	-	-	-	-	-	-	-	-	-	-
CBH	USART Control Register 1	UCTRL1	R/W	0	0	0	0	0	0	0	0
CCH	USART Control Register 2	UCTRL2	R/W	0	0	0	0	0	0	0	0
CDH	USART Control Register 3	UCTRL3	R/W	0	0	0	0	-	0	0	0
CEH	Reserved	-	-	-	-	-	-	-	-	-	-
CFH	USART Status Register	USTAT	R/W	1	0	0	0	0	0	0	0
D0H	Program Status Word Register	PSW	R/W	0	0	0	0	0	0	0	0
D1H	Reserved	-	-	-	-	-	-	-	-	-	-
D2H	P0 Function Selection Low Register	P0FSRL	R/W	0	0	0	0	0	0	0	0
D3H	P0 Function Selection High Register	P0FSRH	R/W	0	0	0	0	0	0	0	0
D4H	P1 Function Selection Low Register	P1FSRL	R/W	0	0	0	0	0	0	0	0
D5H	P1 Function Selection High Register	P1FSRH	R/W	0	0	0	0	0	0	0	0
D6H	P2 Function Selection Low Register	P2FSRL	R/W	0	0	0	0	0	0	0	0
D7H	P2 Function Selection High Register	P2FSRH	R/W	0	0	0	0	0	0	0	0
D8H	Low Voltage Reset Control Register	LVRCR	R/W	-	-	-	0	0	0	0	0
D9H	USI0 Control Register 1	USI0CR1	R/W	0	0	0	0	0	0	0	0
DAH	USI0 Control Register 2	USI0CR2	R/W	0	0	0	0	0	0	0	0
DBH	USI0 Control Register 3	USI0CR3	R/W	0	0	0	0	0	0	0	0
DCH	USI0 Control Register 4	USI0CR4	R/W	0	-	0	0	0	0	0	0
DDH	USI0 Slave Address Register	USI0SAR	R/W	0	0	0	0	0	0	0	0
DEH	P0 De-bounce Enable Register	P0DB	R/W	0	0	-	0	0	0	0	0
DFH	P1/P2/P3 De-bounce Enable Register	P123DB	R/W	-	-	-	0	0	0	0	0

**Table 5. SFR Map (Continued)**

<b>Address</b>	<b>Function</b>	<b>Symbol</b>	<b>R/W</b>	<b>@Reset</b>							
				<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
E0H	Accumulator Register	ACC	R/W	0	0	0	0	0	0	0	0
E1H	USI0 Status Register 1	USI0ST1	R/W	1	0	0	0	0	0	0	0
E2H	USI0 Status Register 2	USI0ST2	R	0	0	0	0	0	0	0	0
E3H	USI0 Baud Rate Generation Register	USI0BD	R/W	1	1	1	1	1	1	1	1
E4H	USI0 SDA Hold Time Register	USI0SHDR	R/W	0	0	0	0	0	0	0	1
E5H	USI0 Data Register	USI0DR	R/W	0	0	0	0	0	0	0	0
E6H	USI0 SCL Low Period Register	USI0SCLR	R/W	0	0	1	1	1	1	1	1
E7H	USI0 SCL High Period Register	USI0SCHR	R/W	0	0	1	1	1	1	1	1
E8H	Reset Flag Register	RSTFR	R/W	1	x	0	0	x	-	-	-
E9H	Reserved	-	-	-	-	-	-	-	-	-	-
EAH	Reserved	-	-	-	-	-	-	-	-	-	-
EBH	Reserved	-	-	-	-	-	-	-	-	-	-
ECH	Reserved	-	-	-	-	-	-	-	-	-	-
EDH	Reserved	-	-	-	-	-	-	-	-	-	-
EEH	P3 Function Selection Register	P3FSR	R/W	-	-	-	-	0	0	0	0
EFH	Reserved	-	-	-	-	-	-	-	-	-	-
F0H	B Register	B	R/W	0	0	0	0	0	0	0	0
F1H	Reserved	-	-	-	-	-	-	-	-	-	-
F2H	Reserved	-	-	-	-	-	-	-	-	-	-
F3H	Reserved	-	-	-	-	-	-	-	-	-	-
F4H	Reserved	-	-	-	-	-	-	-	-	-	-
F5H	Reserved	-	-	-	-	-	-	-	-	-	-
F6H	Reserved	-	-	-	-	-	-	-	-	-	-
F7H	Reserved	-	-	-	-	-	-	-	-	-	-
F8H	Interrupt Priority Register 1	IP1	R/W	-	-	0	0	0	0	0	0
F9H	Reserved	-	-	-	-	-	-	-	-	-	-
FAH	Reserved	-	-	-	-	-	-	-	-	-	-
FBH	Reserved	-	-	-	-	-	-	-	-	-	-
FCH	USART Baud Rate Generation Register	UBAUD	R/W	1	1	1	1	1	1	1	1
FDH	USART Data Register	UDATA	R/W	0	0	0	0	0	0	0	0
FEH	Reserved	-	-	-	-	-	-	-	-	-	-
FFH	Reserved	-	-	-	-	-	-	-	-	-	-

**Table 6. XSFR Map**

<b>Address</b>	<b>Function</b>	<b>Symbol</b>	<b>R/W</b>	<b>@Reset</b>							
				<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
1000H	Timer 3 Control High Register	T3CRH	R/W	0	-	0	0	-	-	-	0
1001H	Timer 3 Control Low Register	T3CRL	R/W	0	0	0	0	-	0	0	0
1002H	Timer 3 A Data High Register	T3ADRH	R/W	1	1	1	1	1	1	1	1
1003H	Timer 3 A Data Low Register	T3ADRL	R/W	1	1	1	1	1	1	1	1
1004H	Timer 3 B Data High Register	T3BDRH	R/W	1	1	1	1	1	1	1	1
1005H	Timer 3 B Data Low Register	T3BDRL	R/W	1	1	1	1	1	1	1	1
1006H	Reserved	-	-	-	-	-	-	-	-	-	-
1007H	Reserved	-	-	-	-	-	-	-	-	-	-
1008H	Timer 4 Control High Register	T4CRH	R/W	0	-	0	0	-	-	-	0
1009H	Timer 4 Control Low Register	T4CRL	R/W	0	0	0	0	-	0	-	0
100AH	Timer 4 A Data High Register	T4ADRH	R/W	1	1	1	1	1	1	1	1
100BH	Timer 4 A Data Low Register	T4ADRL	R/W	1	1	1	1	1	1	1	1
100CH	Timer 4 B Data High Register	T4BDRH	R/W	1	1	1	1	1	1	1	1
100DH	Timer 4 B Data Low Register	T4BDRL	R/W	1	1	1	1	1	1	1	1
100EH	Reserved	-	-	-	-	-	-	-	-	-	-
100FH	Reserved	-	-	-	-	-	-	-	-	-	-
1010H	Timer 5 Control High Register	T5CRH	R/W	0	-	0	0	-	-	-	0
1011H	Timer 5 Control Low Register	T5CRL	R/W	0	0	0	0	-	0	-	0
1012H	Timer 5 A Data High Register	T5ADRH	R/W	1	1	1	1	1	1	1	1
1013H	Timer 5 A Data Low Register	T5ADRL	R/W	1	1	1	1	1	1	1	1
1014H	Timer 5 B Data High Register	T5BDRH	R/W	1	1	1	1	1	1	1	1
1015H	Timer 5 B Data Low Register	T5BDRL	R/W	1	1	1	1	1	1	1	1
1016H	Reserved	-	-	-	-	-	-	-	-	-	-
1017H	Reserved	-	-	-	-	-	-	-	-	-	-
1018H	USART Control Register 4	UCTRL4	R/W	-	-	-	0	0	0	0	0
1019H	USART Floating Point Counter	FPCR	R/W	0	0	0	0	0	0	0	0
101AH	Receiver Time Out Counter High Register	RTOCH	R	0	0	0	0	0	0	0	0
101BH	Receiver Time Out Counter Low Register	RTOCL	R	0	0	0	0	0	0	0	0
101CH	Reserved	-	-	-	-	-	-	-	-	-	-
101DH	Reserved	-	-	-	-	-	-	-	-	-	-
101EH	Reserved	-	-	-	-	-	-	-	-	-	-
101FH	Reserved	-	-	-	-	-	-	-	-	-	-

**Table 6. XSFR Map (Continued)**

<b>Address</b>	<b>Function</b>	<b>Symbol</b>	<b>R/W</b>	<b>@Reset</b>							
				<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
1020H	Flash Mode Register	FEMR	R/W	0	-	0	0	0	0	0	0
1021H	Flash Control Register	FECR	R/W	0	-	0	0	0	0	1	1
1022H	Flash Status Register	FESR	R/W	1	-	-	-	0	0	0	0
1023H	Flash Time Control Register	FETCR	R/W	0	0	0	0	0	0	0	0
1024H	Flash Address Middle Register 1	FEARM1	R/W	0	0	0	0	0	0	0	0
1025H	Flash Address Low Register 1	FEARL1	R/W	0	0	0	0	0	0	0	0
1026H	Reserved	-	-	-	-	-	-	-	-	-	-
1027H	Reserved	-	-	-	-	-	-	-	-	-	-
1028H	Flash Address High Register	FEARH	R/W	0	0	0	0	0	0	0	0
1029H	Flash Address Middle Register	FEARM	R/W	0	0	0	0	0	0	0	0
102AH	Flash Address Low Register	FEARL	R/W	0	0	0	0	0	0	0	0
102BH	Flash Data Register	FEDR	R/W	0	0	0	0	0	0	0	0
102CH	Flash Test Register	FETR	R/W	0	0	0	0	0	0	0	0
102DH	Reserved	-	-	-	-	-	-	-	-	-	-
102EH	Reserved	-	-	-	-	-	-	-	-	-	-
102FH	Reserved	-	-	-	-	-	-	-	-	-	-
1030H	Reserved	-	-	-	-	-	-	-	-	-	-
1031H	Reserved	-	-	-	-	-	-	-	-	-	-
1032H	Reserved	-	-	-	-	-	-	-	-	-	-
1033H	Reserved	-	-	-	-	-	-	-	-	-	-
1034H	Reserved	-	-	-	-	-	-	-	-	-	-
1035H	Reserved	-	-	-	-	-	-	-	-	-	-
1036H	Reserved	-	-	-	-	-	-	-	-	-	-
1037H	Reserved	-	-	-	-	-	-	-	-	-	-

#### 4.4.3 Compiler Compatible SFR

##### ACC (Accumulator Register): E0H

7	6	5	4	3	2	1	0
ACC							
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

ACC

Accumulator

##### B (B Register): F0H

7	6	5	4	3	2	1	0
B							
RW							

Initial value: 00H

B

B Register

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

7	6	5	4	3	2	1	0
SP							
RW							

Initial value: 07H

SP

Stack Pointer

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

7	6	5	4	3	2	1	0
DPL							
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

DPL

Data Pointer Low

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

7	6	5	4	3	2	1	0
DPH							
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

DPH

Data Pointer High

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

7	6	5	4	3	2	1	0
DPL1							
RW	RW	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							
RW	RW	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
RW	RW	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

## 5 I/O Ports

### 5.1 I/O Ports

The A96T418 has four groups of I/O ports (P0 ~ P3). 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 includes function that can generate interrupt according to change of state of the pin.

### 5.2 Port Register

#### 5.2.1 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.

#### 5.2.2 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.

#### 5.2.3 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.

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

There are internally open-drain selection registers (PxOD) for P0 ~ P2. 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.

#### 5.2.5 De-bounce Enable Register (PxDB)

P0[7:3], P1[2:1], P2[6:5] and P31 support debounce function. Debounce clocks of each ports are fx/1, fx/4, and fx/4096.

### 5.2.6 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.

### 5.2.7 Register Map

**Table 7. Port Register Map**

Name	Address	Direction	Default	Description
P0	80H	R/W	00H	P0 Data Register
P0IO	A1H	R/W	00H	P0 Direction Register
P0PU	ACH	R/W	00H	P0 Pull-up Resistor Selection Register
P0OD	91H	R/W	00H	P0 Open-drain Selection Register
P0DB	DEH	R/W	00H	P0 De-bounce Enable Register
P0FSRH	D3H	R/W	00H	P0 Function Selection High Register
P0FSRL	D2H	R/W	00H	P0 Function Selection Low Register
P1	88H	R/W	00H	P1 Data Register
P1IO	B1H	R/W	00H	P1 Direction Register
P1PU	ADH	R/W	00H	P1 Pull-up Resistor Selection Register
P1OD	92H	R/W	00H	P1 Open-drain Selection Register
P123DB	DFH	R/W	00H	P1/P2/P3 De-bounce Enable Register
P1FSRH	D5H	R/W	00H	P1 Function Selection High Register
P1FSRL	D4H	R/W	00H	P1 Function Selection Low Register
P2	90H	R/W	00H	P2 Data Register
P2IO	B9H	R/W	00H	P2 Direction Register
P2PU	AEH	R/W	00H	P2 Pull-up Resistor Selection Register
P2OD	93H	R/W	00H	P2 Open-drain Selection Register
P2FSRH	D7H	R/W	00H	P2 Function Selection High Register
P2FSRL	D6H	R/W	00H	P2 Function Selection Low Register
P3	98H	R/W	00H	P3 Data Register
P3IO	C1H	R/W	00H	P3 Direction Register
P3PU	AFH	R/W	00H	P3 Pull-up Resistor Selection Register
P3FSR	EEH	R/W	00H	P3 Function Selection Register

## 5.3 P0 Port

### 5.3.1 P0 Port Description

P0 is 8-bit I/O port. P0 control registers consist of P0 data register (P0), P0 direction register (P0IO), de-bounce 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.

### 5.3.2 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
RW	RW	R/W	R/W	R/W	R/W	R/W	RW

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
RW	RW	R/W	R/W	R/W	R/W	R/W	RW

Initial value: 00H

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

0	Input
1	Output

**NOTES:**

- 1. EC3(P06) possible when P0FSRH[5:4] = '01'

#### P0PU (P0 Pull-up Resistor Selection Register): ACH

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

Initial value: 00H

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

0	Disable
1	Enable

**P0OD (P0 Open-drain Selection Register): 91H**

7	6	5	4	3	2	1	0
P07OD	P06OD	P05OD	P04OD	P03OD	P02OD	P01OD	P00OD
RW							

Initial value: 00H

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

- 0 Push-pull output
- 1 Open-drain output

**P0DB (P0 De-bounce Enable Register): DEH**

7	6	5	4	3	2	1	0
DBCLK1	DBCLK0	-	P07DB	P06DB	P05DB	P04DB	P03DB
RW	RW	-	RW	RW	RW	RW	RW

Initial value: 00H

DBCLK[1:0] Configure De-bounce Clock of Port

DBCLK1	DBCLK0	Description
0	0	fx/1
0	1	fx/4
1	0	fx/4096
1	1	LSIRC (128KHz)

P07DB Configure De-bounce of P07 Port

- 0 Disable
- 1 Enable

P06DB Configure De-bounce of P06 Port

- 0 Disable
- 1 Enable

P05DB Configure De-bounce of P05 Port

- 0 Disable
- 1 Enable

P04DB Configure De-bounce of P04Port

- 0 Disable
- 1 Enable

P03DB Configure De-bounce of P03 Port

- 0 Disable
- 1 Enable

**NOTES:**

1. If the same 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 clocks or more to be actually detected as a valid edge.
3. The port de-bounce is automatically disabled at stop mode and recovered after stop mode release.

**P0FSRH (Port 0 Function Selection High Register): D3H**

7	6	5	4	3	2	1	0
P0FSRH7	P0FSRH6	P0FSRH5	P0FSRH4	P0FSRH3	P0FSRH2	P0FSRH1	P0FSRH0
R/W							

Initial value: 00H

## P0FSRH[7:6]

## P07 Function Select

P0FSRH7 P0FSRH6 Description

0 0 I/O Port (EINT4 function possible when input)

0 1 Reserved

1 0 AN3 Function

1 1 PWM5O/T5O Function

## P0FSRH[5:4]

## P06 Function Select

P0FSRH5 P0FSRH4 Description

0 0 I/O Port (EINT3 function possible when input)

0 1 EC3 Function

1 0 AN2 Function

1 1 PWM4O/T4O Function

## P0FSRH[3:2]

## P05 Function Select

P0FSRH3 P0FSRH2 Description

0 0 I/O Port (EINT2 function possible when input)

0 1 Reserved

1 0 AN1 Function

1 1 PWM3O/T3O Function

## P0FSRH[1:0]

## P04 Function Select

P0FSRH1 P0FSRH0 Description

0 0 I/O Port (EINT1 function possible when input)

0 1 Reserved

1 0 AN0 Function

1 1 SS0 Function

**P0FSRL (Port 0 Function Selection Low Register): D2H**

7	6	5	4	3	2	1	0
P0FSRL7	P0FSRL6	P0FSRL5	P0FSRL4	P0FSRL3	P0FSRL2	P0FSRL1	P0FSRL0
RW	RW	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P0FSRL[7:6]	P03 Function Select	
P0FSRL7	P0FSRL6	Description
0	0	I/O Port (EINT0 function possible when input)
0	1	BUZ0 Function
1	0	AN7/AVREF Function
1	1	SCK0 Function
P0FSRL[5:4]	P02 Function Select	
P0FSRL5	P0FSRL4	Description
0	0	I/O Port
0	1	Reserved
1	0	Reserved
1	1	TXD0/MOSI0/SDA0 Function
P0FSRL[3:2]	P01 Function Select	
P0FSRL3	P0FSRL2	Description
0	0	I/O Port (EC3 function possible when input)
0	1	T3O/PWM3O Function(P02 Output)
1	0	Reserved
1	1	RXD0/MISO0/SCL0 Function
P0FSRL[1:0]	P00 Function Select	
P0FSRL1	P0FSRL0	Description
0	0	I/O Port
0	1	Reserved
1	0	Reserved
1	1	TXD1/MOSI1 Function

**NOTES:**

1. EINT0 ~ 4 function possible when input
2. Both TXD0 at P02 and RXD0 at P01 function should be selected even for TXD0 only function or RXD0 function only function.
3. T3O/PWM3O Function(P02 Output) should be set at P01 Function Select position.

## 5.4 P1 Port

### 5.4.1 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 (P123DB), 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.

### 5.4.2 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
RW							

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
RW							

Initial value: 00H

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

0	Input
1	Output

#### P1PU (P1 Pull-up Resistor Selection Register): ADH

7	6	5	4	3	2	1	0
P17PU	P16PU	P15PU	P14PU	P13PU	P12PU	P11PU	P10PU
RW							

Initial value: 00H

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

0	Disable
1	Enable

**P1OD (P1 Open-drain Selection Register): 92H**

7	6	5	4	3	2	1	0
P17OD	P16OD	P15OD	P14OD	P13OD	P12OD	P11OD	P10OD
RW	RW	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

**P123DB (P1/P2/P3 De-bounce Enable Register): DFH**

7	6	5	4	3	2	1	0
-	-	-	P31DB	P26DB	P25DB	P12DB	P11DB
-	-	-	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P31DB      Configure De-bounce of P31 Port  
 0      Disable  
 1      Enable  
 P26DB      Configure De-bounce of P26 Port  
 0      Disable  
 1      Enable  
 P25DB      Configure De-bounce of P25 Port  
 0      Disable  
 1      Enable  
 P12DB      Configure De-bounce of P12 Port  
 0      Disable  
 1      Enable  
 P11DB      Configure De-bounce of P11 Port  
 0      Disable  
 1      Enable

**NOTES:**

1. If the same 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 clocks or more to be actually detected as a valid edge.
3. The port de-bounce is automatically disabled at stop mode and recovered after stop mode release.
4. Refer to the port 0 de-bounce enable register (P0DB) for the de-bounce clock of port 1, port 2 and port 3.

**P1FSRH (Port 1 Function Selection High Register): D5H**

7	6	5	4	3	2	1	0
P1FSRH7	P1FSRH6	P1FSRH5	P1FSRH4	P1FSRH3	P1FSRH2	P1FSRH1	P1FSRH0
RW							

Initial value: 00H

P1FSRH[7:6]	P17 Function Select		
	P1FSRH7	P1FSRH6	Description
	0	0	I/O Port
	0	1	Reserved
	1	0	Reserved
	1	1	TXD1/MOSI1 Function
P1FSRH[5:4]	P16 Function Select		
	P1FSRH5	P1FSRH4	Description
	0	0	I/O Port
	0	1	Reserved
	1	0	Reserved
	1	1	RXD1/MISO1 Function
P1FSRH[3:2]	P15 Function Select		
	P1FSRH3	P1FSRH2	Description
	0	0	I/O Port
	0	1	Reserved
	1	0	Reserved
	1	1	SCK1 Function
P1FSRH[1:0]	P14 Function Select		
	P1FSRH1	P1FSRH0	Description
	0	0	I/O Port
	0	1	Reserved
	1	0	Reserved
	1	1	SS1 Function

**P1FSRL (Port 1 Function Selection Low Register): D4H**

7	6	5	4	3	2	1	0
P1FSRL7	P1FSRL6	P1FSRL5	P1FSRL4	P1FSRL3	P1FSRL2	P1FSRL1	P1FSRL0
R/W							

Initial value: 00H

P1FSRL[7:6]

P13 Function Select

P1FSRL7	P1FSRL6	Description
0	0	I/O Port
0	1	Reserved
1	0	Reserved
1	1	Reserved

P1FSRL[5:4]

P12 Function Select

P1FSRL5	P1FSRL4	Description
0	0	I/O Port (EINT12 function possible when input)
0	1	Reserved
1	0	AN6 Function
1	1	T2O/PWM2O Function

P1FSRL[3:2]

P11 Function Select

P1FSRL3	P1FSRL2	Description
0	0	I/O Port (EINT5 function possible when input)
0	1	T1O/PWM1O Function
1	0	AN5 Function
1	1	TXD0/MOSI0/SDA0 Function

P1FSRL[1:0]

P10 Function Select

P1FSRL1	P1FSRL0	Description
0	0	I/O Port
0	1	Reserved
1	0	AN4 Function
1	1	RXD0/MISO0/SCL0 Function

## 5.5 P2 Port

### 5.5.1 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.

### 5.5.2 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							

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							

Initial value: 00H

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

0	Input
1	Output

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

7	6	5	4	3	2	1	0
P27PU	P26PU	P25PU	P24PU	P23PU	P22PU	P21PU	P20PU
RW	RW	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): 93H**

7	6	5	4	3	2	1	0
P27OD	P26OD	P25OD	P24OD	P23OD	P22OD	P21OD	P20OD
RW	RW	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): D7H**

7	6	5	4	3	2	1	0
P2FSRH7	P2FSRH6	P2FSRH5	P2FSRH4	P2FSRH3	P2FSRH2	P2FSRH1	P2FSRH0
RW	RW	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

P2FSRH[7:6]	P27 Function Select	
P2FSRH7	P2FSRH6	Description
0	0	I/O Port
0	1	SXIN Function
1	0	T0O/PWM0O Function
1	1	Reserved
P2FSRH[5:4]	P26 Function Select	
P2FSRH5	P2FSRH4	Description
0	0	I/O Port (EINT10 function possible when input)
0	1	SXOUT Function
1	0	Reserved
1	1	Reserved
P2FSRH[3:2]	P25 Function Select	
P2FSRH3	P2FSRH2	Description
0	0	I/O Port (EINT11 function possible when input)
0	1	EC0 Function
1	0	Reserved
1	1	Reserved
P2FSRH[1:0]	P24 Function Select	
P2FSRH1	P2FSRH0	Description
0	0	I/O Port
0	1	Reserved

1	0	Reserved
1	1	Reserved

**P2FSRL (Port 2 Function Selection Low Register): D6H**

7	6	5	4	3	2	1	0
P2FSRL7	P2FSRL6	P2FSRL5	P2FSRL4	P2FSRL3	P2FSRL2	P2FSRL1	P2FSRL0
R/W							

Initial value: 00H

P2FSRL[7:6]	P23 Function Select		
	P2FSRL7	P2FSRL6	Description
	0	0	I/O Port
	0	1	EC1 Function
	1	0	Reserved
	1	1	Reserved
P2FSRL[5:4]	P22 Function Select		
	P2FSRL5	P2FSRL4	Description
	0	0	I/O Port
	0	1	Reserved
	1	0	Reserved
	1	1	Reserved
P2FSRL[3:2]	P21 Function Select		
	P2FSRL3	P2FSRL2	Description
	0	0	I/O Port
	0	1	Reserved
	1	0	Reserved
	1	1	Reserved
P2FSRL[1:0]	P20 Function Select		
	P2FSRL3	P2FSRL2	Description
	0	0	I/O Port
	0	1	Reserved
	1	0	Reserved
	1	1	Reserved

## 5.6 P3 Port

### 5.6.1 P3 Port Description

P3 is 2-bit I/O port. P3 control registers consist of P3 data register (P3), P3 direction register (P3IO) and P3 pull-up resistor selection register (P3PU). Refer to the port function selection registers for the P3 function selection.

### 5.6.2 Register Description for P3

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

7	6	5	4	3	2	1	0
-	-	-	-	-	-	P31	P30
-	-	-	-	-	-	RW	RW

Initial value: 00H

P3[1:0]      I/O Data

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

7	6	5	4	3	2	1	0
-	-	-	-	-	-	P31IO	P30IO
-	-	-	-	-	-	RW	RW

Initial value: 00H

P3IO[1:0]      P3 Data I/O Direction

0      Input  
1      Output

#### P3PU (P3 Pull-up Resistor Selection Register): AFH

7	6	5	4	3	2	1	0
-	-	-	-	-	-	P31PU	P30PU
-	-	-	-	-	-	RW	RW

Initial value: 00H

P3PU[1:0]      Configure Pull-up Resistor of P3 Port

0      Disable  
1      Enable

**P3FSR (Port 3 Function Selection Register): EEH**

7	6	5	4	3	2	1	0
-	-	-	-	P3FSR3	P3FSR2	P3FSR1	P3FSR0
-	-	-	-	R/W	R/W	R/W	R/W

Initial value: 00H

P3FSR[3:2]	P31 Function Select	
	P3FSR3	P3FSR2
	0	0
	0	1
P3FSR[1:0]	P30 Function Select	
	P3FSR1	P3FSR0
	0	0
	0	1

P3FSR3	P3FSR2	Description
0	0	I/O Port (EINT8 function possible when input)
0	1	Reserved
1	0	Reserved
1	1	RXD1/MISO1 Function

P3FSR1	P3FSR0	Description
0	0	I/O Port
0	1	Reserved
1	0	Reserved
1	1	Reserved

## 6 Interrupt Controller

### 6.1 Overview

The A96T418 supports up to 23 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 23 interrupt source
- 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, and 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 A96T418 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 8 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 8. Interrupt Group Priority Level**

Interrupt Group	Highest → Lowest					Highest	Lowest
0 (Bit0)	Interrupt 0	Interrupt 6	Interrupt 12	Interrupt 18			
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			

## 6.2 External Interrupt

The external interrupt on INT0, INT1, INT5, INT6 and INT11 receive various interrupt request depending on the external interrupt polarity 0 high/low register (EIPOL0H/L) and external interrupt polarity 1 register (EIPOL1) as shown in Figure 14. Also each external interrupt source has enable/disable bits. The External interrupt flag 0 register (EIFLAG0) and external interrupt flag 1 register 1 (EIFLAG1) provides the status of external interrupts.

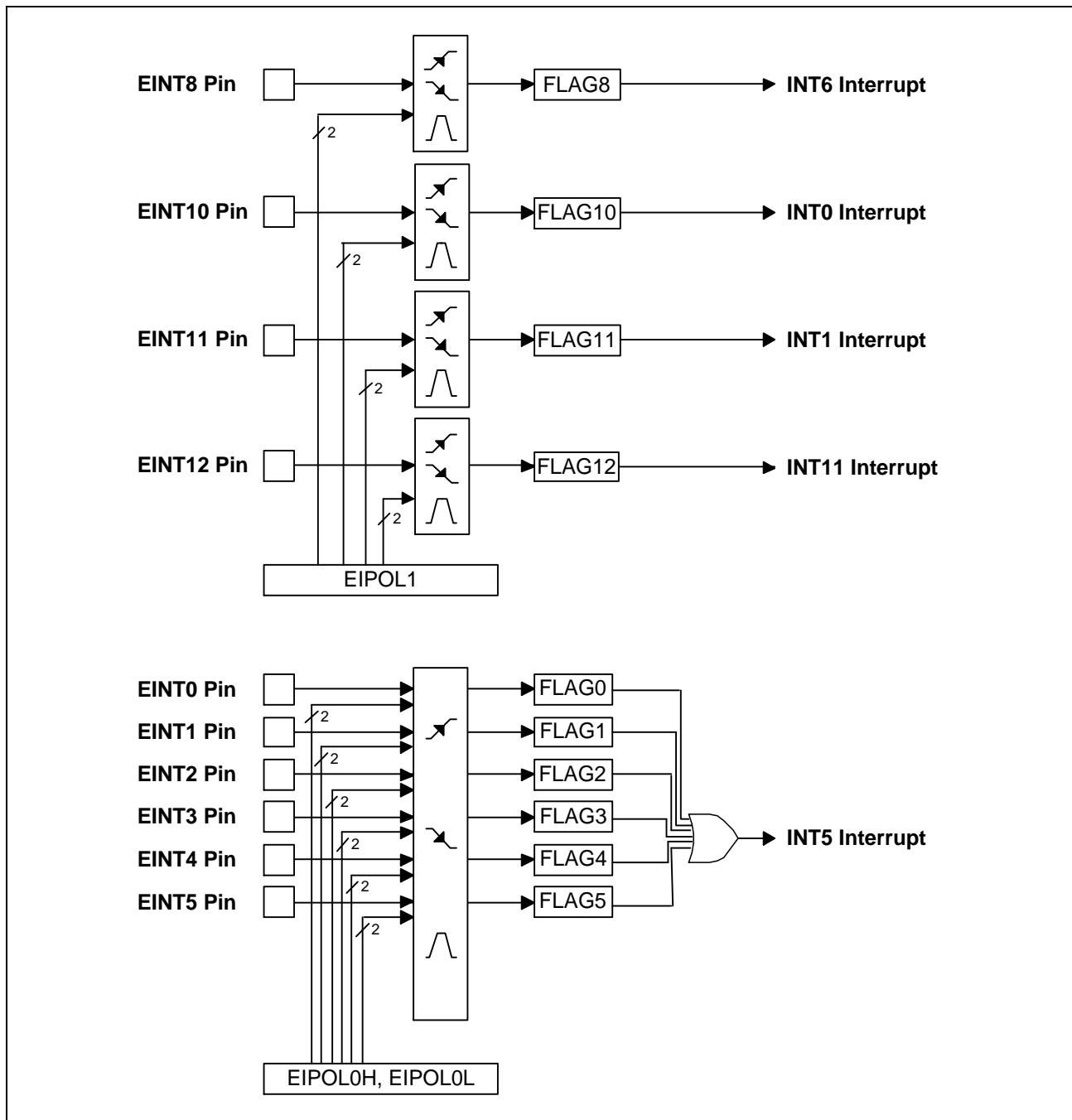


Figure 14. External Interrupt Description

### 6.3 Block Diagram

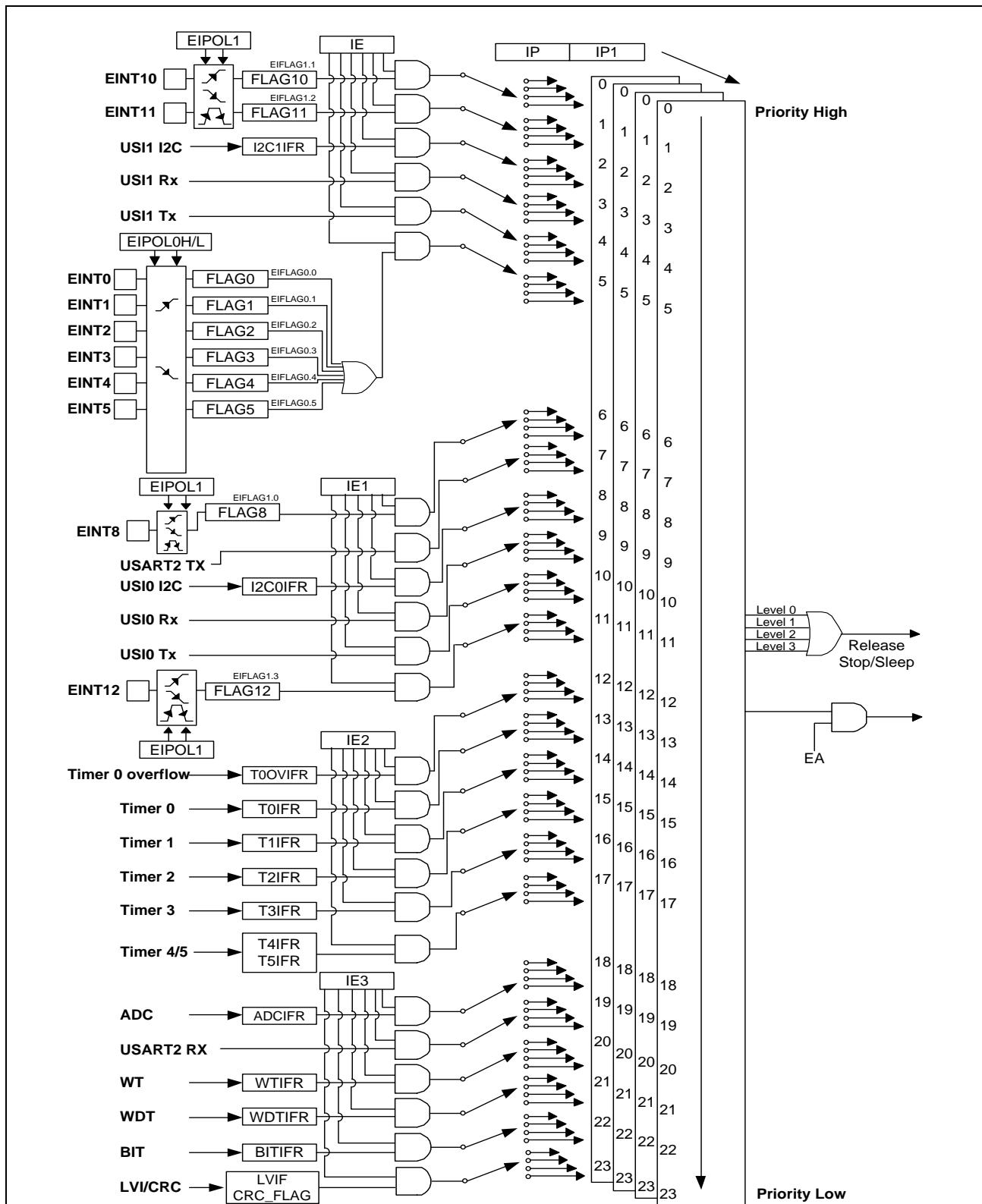


Figure 15. Block Diagram of Interrupt

**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.

## 6.4 Interrupt Vector Table

The interrupt controller supports 23 interrupt sources as shown in the Table 9. 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 9. Interrupt Vector Address Table**

Interrupt Source	Symbol	Interrupt Enable Bit	Polarity	Mask	Vector Address
Hardware Reset	RESETB	-	0	Non-Maskable	0000H
External Interrupt 10	INT0	IE.0	1	Maskable	0003H
External Interrupt 11	INT1	IE.1	2	Maskable	000BH
LED Interrupt	INT2	IE.2	3	Maskable	0013H
Touch Interrupt	INT3	IE.3	4	Maskable	001BH
Reserved	INT4	IE.4	5	Maskable	0023H
External Interrupt 0 – 5	INT5	IE.5	6	Maskable	002BH
External Interrupt 8	INT6	IE1.0	7	Maskable	0033H
USART1 TX Interrupt	INT7	IE1.1	8	Maskable	003BH
USI0 I2C Interrupt	INT8	IE1.2	9	Maskable	0043H
USI0 RX Interrupt	INT9	IE1.3	10	Maskable	004BH
USI0 TX Interrupt	INT10	IE1.4	11	Maskable	0053H
External Interrupt 12	INT11	IE1.5	12	Maskable	005BH
T0 Overflow Interrupt	INT12	IE2.0	13	Maskable	0063H
T0 Match Interrupt	INT13	IE2.1	14	Maskable	006BH
T1 Match Interrupt	INT14	IE2.2	15	Maskable	0073H
T2 Match Interrupt	INT15	IE2.3	16	Maskable	007BH
T3 Match Interrupt	INT16	IE2.4	17	Maskable	0083H
T4/T5 Match Interrupt	INT17	IE2.5	18	Maskable	008BH
ADC Interrupt	INT18	IE3.0	19	Maskable	0093H
USART1 RX Interrupt	INT19	IE3.1	20	Maskable	009BH
WT Interrupt	INT20	IE3.2	21	Maskable	00A3H
WDT Interrupt	INT21	IE3.3	22	Maskable	00ABH
BIT Interrupt	INT22	IE3.4	23	Maskable	00B3H
LVI Interrupt	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.

## 6.5 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.

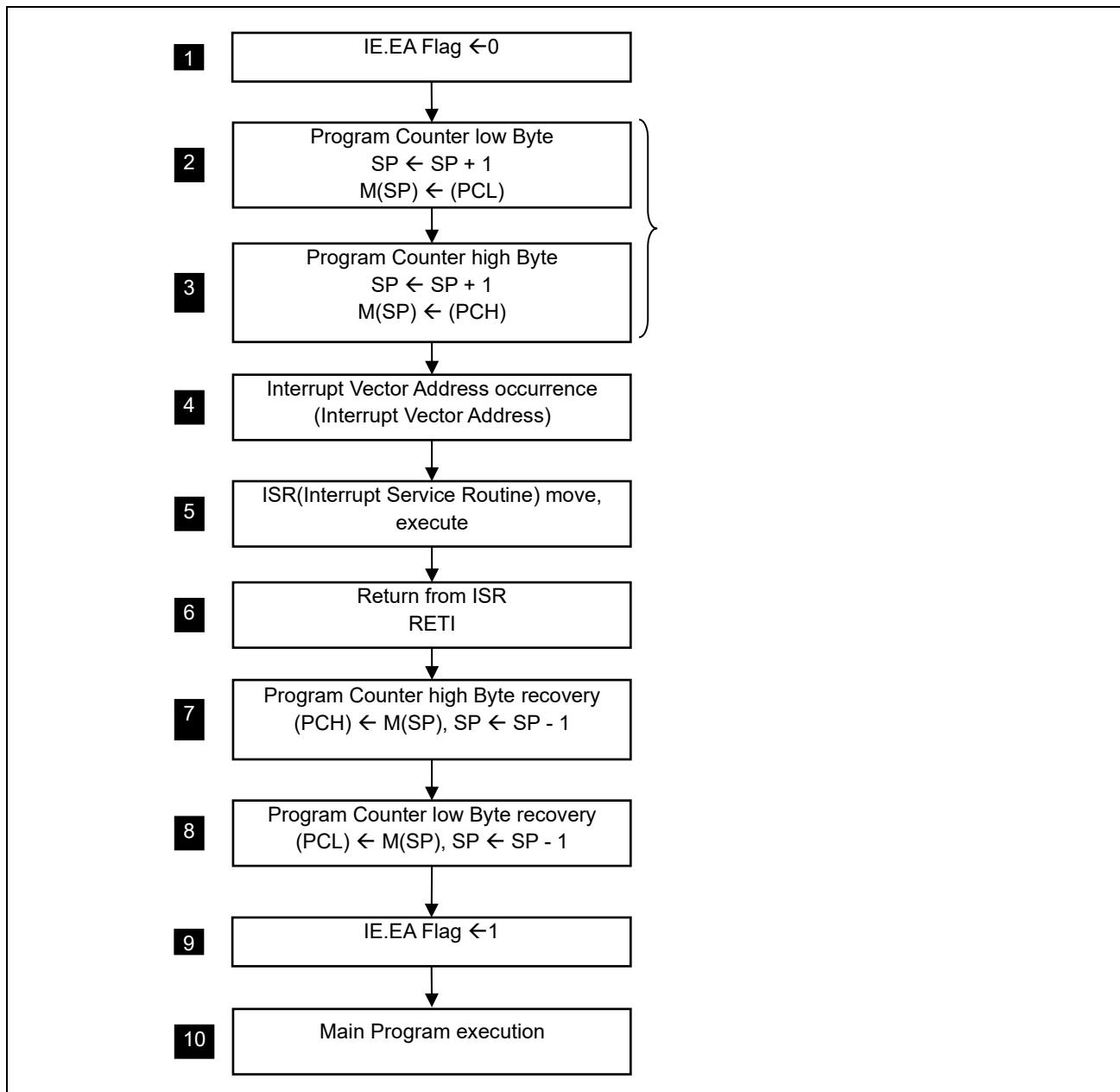


Figure 16. Interrupt Sequence Flow

## 6.6 Effective Timing after Controlling Interrupt Bit

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

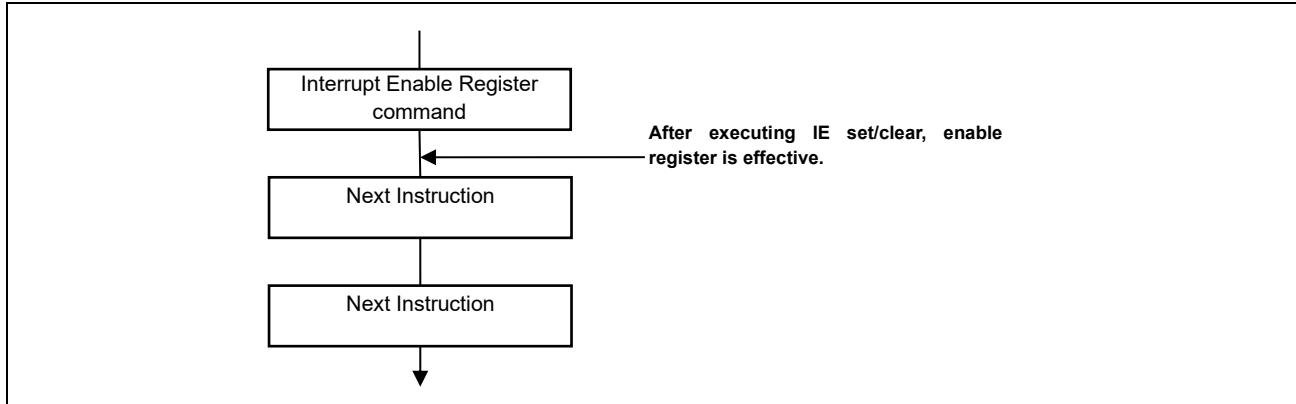


Figure 17. Effective Timing of Interrupt Enable Register

Case b) Interrupt flag Register

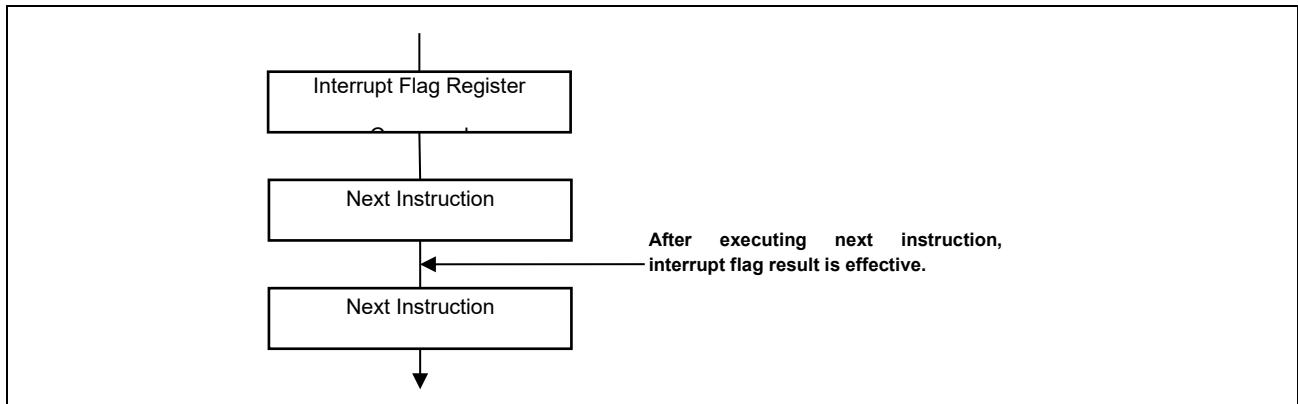
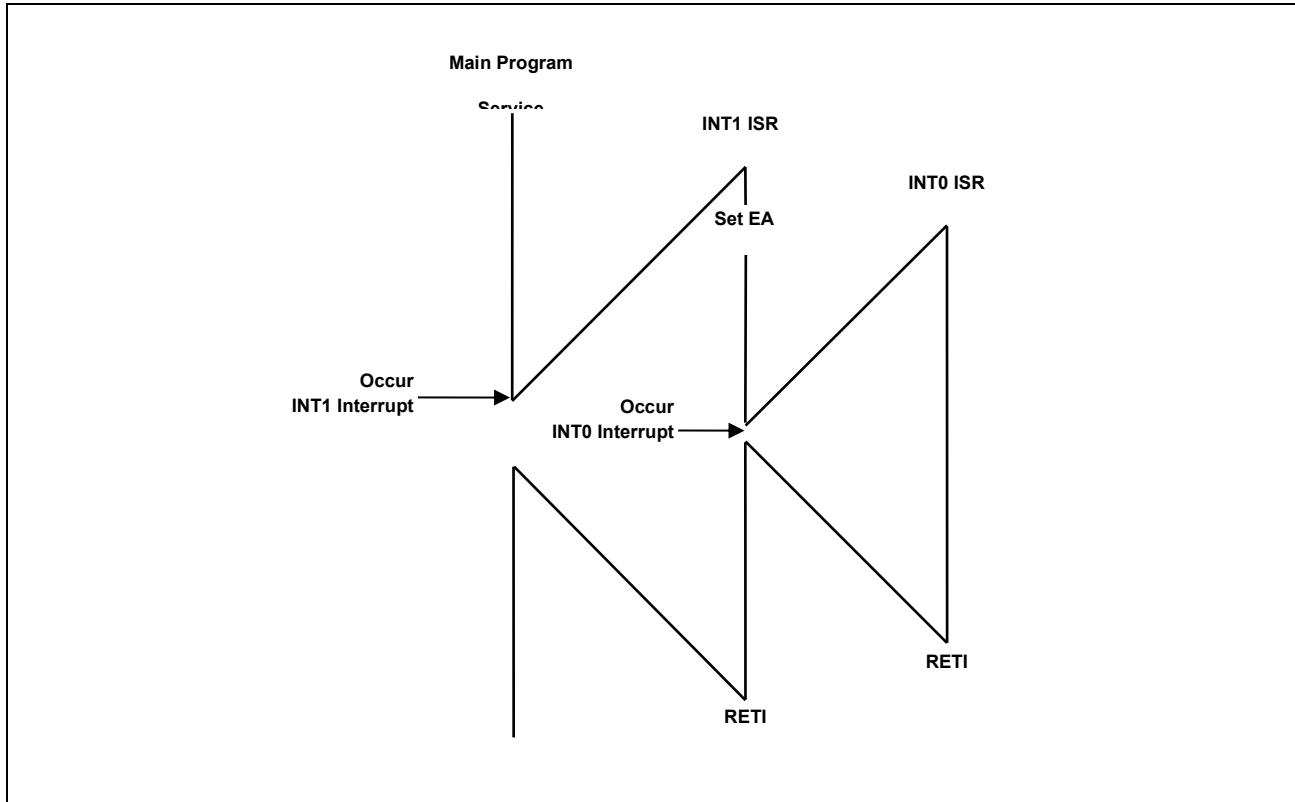


Figure 18. Effective Timing of Interrupt Flag Register

## 6.7 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 19. Effective Timing of Multi-Interrupt**

Figure 19 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.

## 6.8 Interrupt Enable Accept Timing

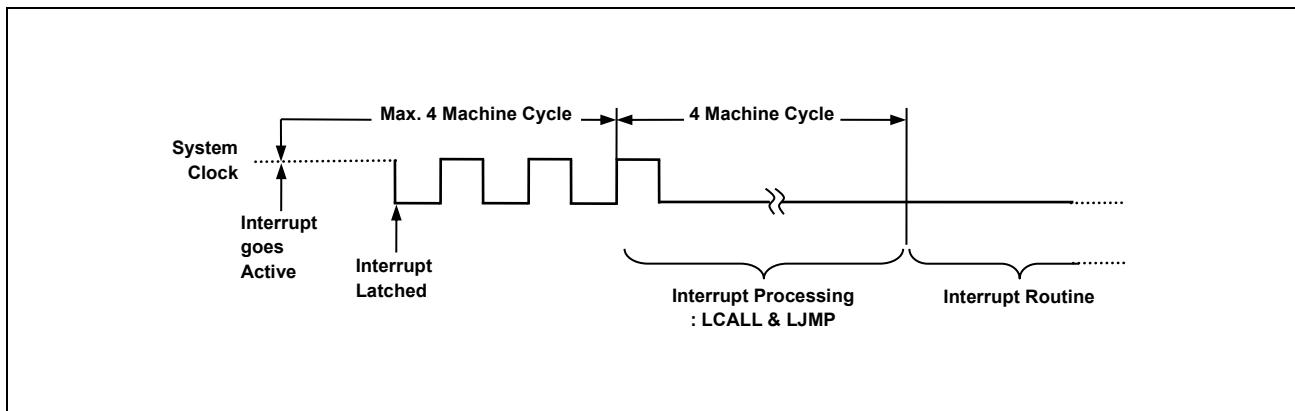


Figure 20. Interrupt Response Timing Diagram

## 6.9 Interrupt Service Routine Address

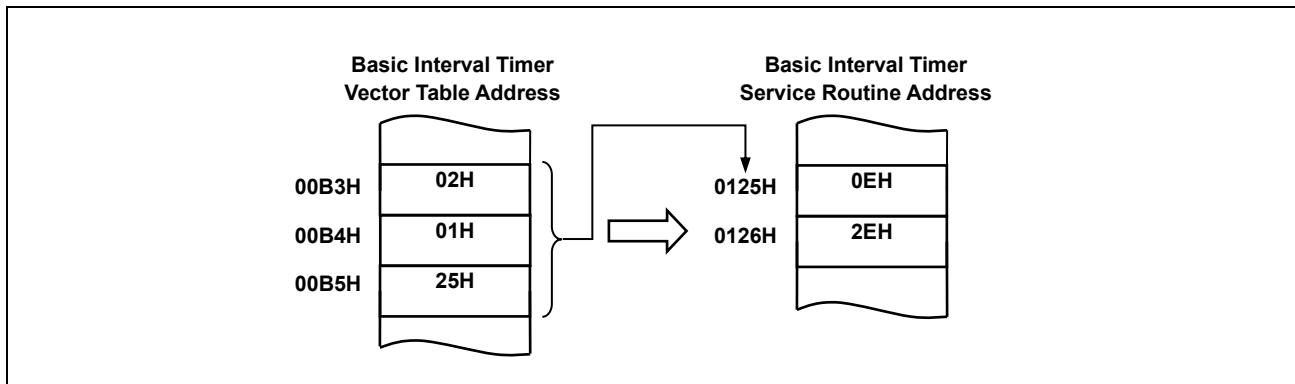


Figure 21. Correspondence between Vector Table Address and the Entry Address of ISR

## 6.10 Saving/Restore General-Purpose Registers

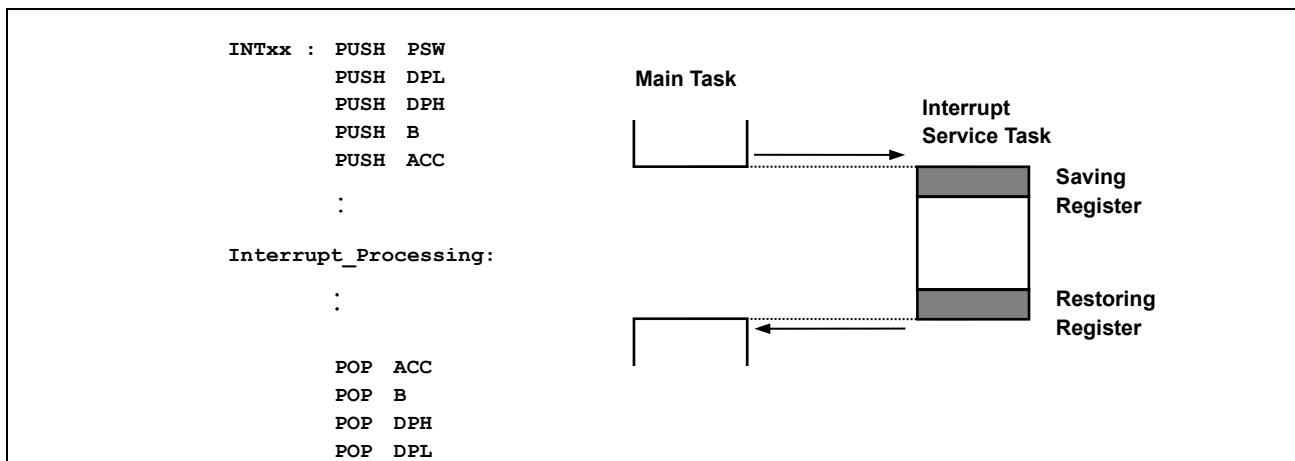


Figure 22. Saving/Restore Process Diagram and Sample Source

## 6.11 Interrupt Timing

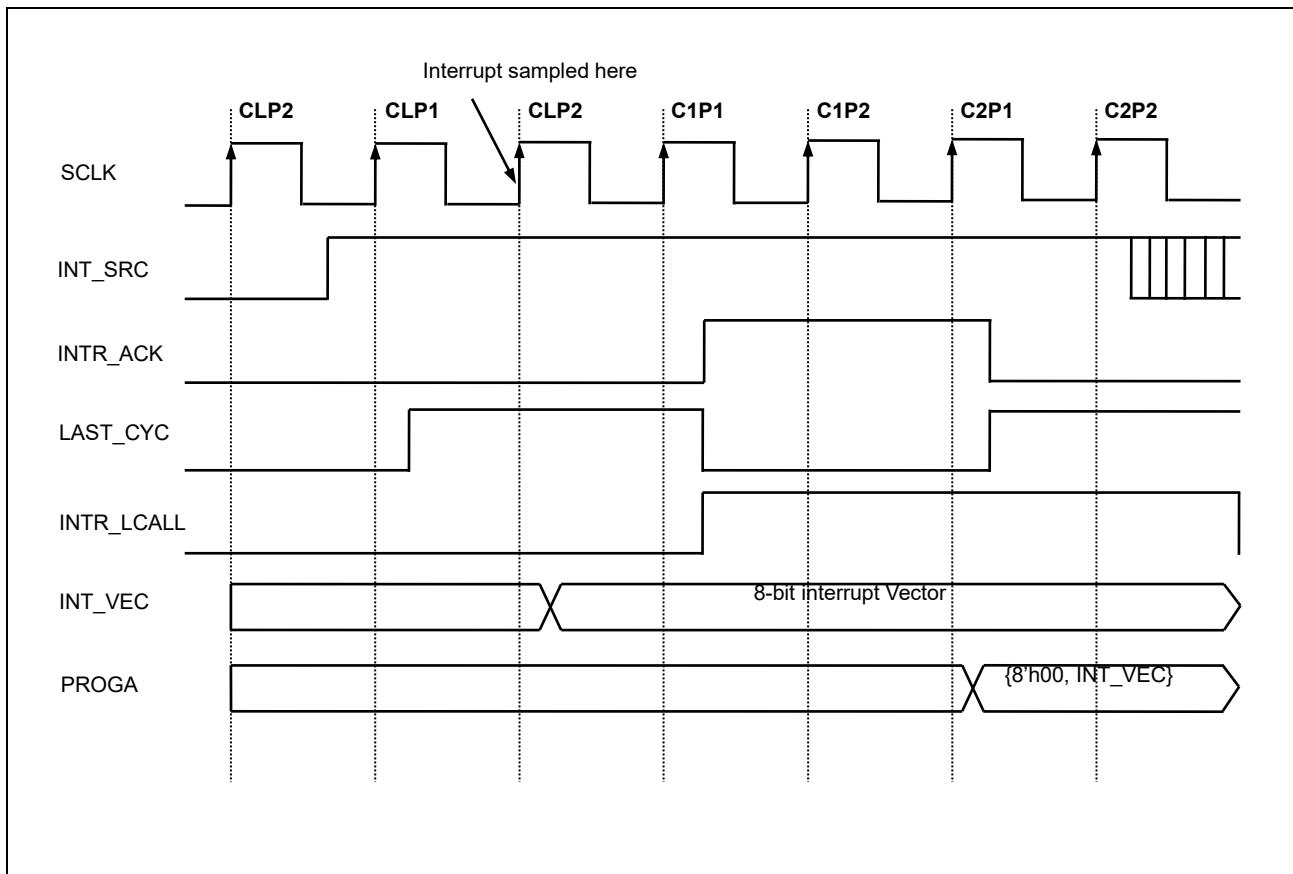


Figure 23. Timing Chart of Interrupt Acceptance and Interrupt Return Instruction

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.

**NOTES:**

1. command cycle CLPx: L=Last cycle, 1=1<sup>st</sup> cycle or 1<sup>st</sup> phase, 2=2<sup>nd</sup> cycle or 2<sup>nd</sup> phase

## 6.12 Interrupt Register Overview

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

Interrupt enable register consists of global interrupt control bit (EA) and peripheral interrupt control bits. Total 23 peripherals are able to control interrupt.

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

The 23 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.

### 6.12.3 External Interrupt Flag Register (EIFLAG0, EIFLAG1)

The external interrupt flag 0 register (EIFLAG0) and external interrupt flag 1 register (EIFLAG1) 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.

### 6.12.4 External Interrupt Polarity Register (EIPOL0L, EIPOL0H, EIPOL1)

The external interrupt polarity0 high/low register (EIPOL0H/L) and external interrupt polarity1 register (EIPOL1) determines which type of rising/falling/both edge interrupt. Initially, default value is no interrupt at any edge.

### 6.12.5 Register Map

**Table 10. Interrupt Register Map**

Name	Address	Direction	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	C0H	R/W	00H	External Interrupt Flag 0 Register
EIPOL0L	A4H	R/W	00H	External Interrupt Polarity 0 Low Register
EIPOL0H	A5H	R/W	00H	External Interrupt Polarity 0 High Register
EIFLAG1	A6H	R/W	00H	External Interrupt Flag 1 Register
EIPOL1	A7H	R/W	00H	External Interrupt Polarity 1 Register

### 6.12.6 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) and external interrupt polarity 1 register(EIPOL1).

### 6.12.7 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 ~ 5 (EINT0 ~ EINT5)
0	Disable
1	Enable
INT4E	Reserved
0	Disable
1	Enable
INT3E	Enable or Disable Touch Interrupt
0	Disable
1	Enable
INT2E	Enable or Disable LED Interrupt
0	Disable
1	Enable
INT1E	Enable or Disable External Interrupt 11 (EINT11)
0	Disable
1	Enable
INT0E	Enable or Disable External Interrupt 10 (EINT10)
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 External Interrupt 12 (EINT12)
0	Disable
1	Enable
INT10E	Enable or Disable USI0 TX Interrupt
0	Disable
1	Enable
INT9E	Enable or Disable USI0 RX Interrupt
0	Disable
1	Enable
INT8E	Enable or Disable USI0 I2C Interrupt
0	Disable
1	Enable
INT7E	Enable or Disable USART1 TX Interrupt
0	Disable
1	Enable
INT6E	Enable or Disable External Interrupt 8 (EINT8)
0	Disable
1	Enable

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

7	6	5	4	3	2	1	0
—	—	INT17E	INT16E	INT15E	INT14E	INT13E	INT12E
—	—	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

INT17E	Enable or Disable Timer 4/5 Match Interrupt
0	Disable
1	Enable
INT16E	Enable or Disable Timer 3 Match Interrupt
0	Disable
1	Enable
INT15E	Enable or Disable Timer 2 Match Interrupt
0	Disable
1	Enable
INT14E	Enable or Disable Timer 1 Match Interrupt
0	Disable
1	Enable
INT13E	Enable or Disable Timer 0 Match Interrupt
0	Disable
1	Enable
INT12E	Enable or Disable Timer 0 Overflow 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 LVI Interrupt
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 Interrupt
0	Disable
1	Enable
INT19E	Enable or Disable USART1 RX Interrupt
0	Disable
1	Enable
INT18E	Enable or Disable ADC 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 Flag0 Register): C0H**

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

Initial value: 00H

EIFLAG0[5:0]	When an External Interrupt 0~5 is occurred, the flag becomes '1'. The flag is cleared only by writing '0' to the bit. So, the flag should be cleared by software.
0	External Interrupt 0 ~ 5 not occurred
1	External Interrupt 0 ~ 5 occurred

**EIPOL0H (External Interrupt Polarity 0 High Register): A5H**

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

Initial value: 00H

EIPOL0H[3:0]	External interrupt (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 and 5

**EIPOL0L (External Interrupt Polarity 0 Low Register): A4H**

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

Initial value: 00H

EIPOL0[7:0]	External interrupt (EINT0, EINT1, EINT2, EINT3) 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): A6H**

7	6	5	4	3	2	1	0
T0OVIFR	T0IFR	-	-	FLAG12	FLAG11	FLAG10	FLAG8
R/W	R/W	-	-	R/W	R/W	R/W	R/W

Initial value: 00H

T0OVIFR	When T0 overflow interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or automatically clear by INT_ACK signal. Writing "1" has no effect.
	0 T0 overflow Interrupt no generation
	1 T0 overflow Interrupt generation
T0IFR	When T0 interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or automatically clear by INT_ACK signal. Writing "1" has no effect.
	0 T0 Interrupt no generation
	1 T0 Interrupt generation
EIFLAG1[3:0]	When an External Interrupt (EINT8, EINT10-EINT12) is occurred, the flag becomes '1'. The flag is cleared by writing '0' to the bit or automatically cleared by INT_ACK signal. Writing "1" has no effect.
	0 External Interrupt not occurred
	1 External Interrupt occurred

**EIPOL1 (External Interrupt Polarity 1 Register): A7H**

7	6	5	4	3	2	1	0
POL12		POL11		POL10		POL8	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

EIPOL1[7:2]	External interrupt (EINT8, EINT10,EINT11,EINT12) 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, 10, 11 and 12

## 7 Clock Generator

### 7.1 Overview

As shown in Figure 24, 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 sub-frequency clock oscillator. The sub clock operation can be easily obtained by attaching a crystal between the SXIN and SXOUT pin. The sub clock can be also obtained from the external oscillator. In this case, it is necessary to put the external clock signal into the SXIN pin and open the SXOUT pin. The default system clock is 1MHz INT-RC Oscillator. In order to stabilize system internally, it is used 128kHz LOW INT-RC oscillator on POR.

2. Calibrated High Internal RC Oscillator ( $f_{HSIRC} = 16\text{MHz}$ )
3. HSIRC OSC (16MHz, Default system clock)
4. HSIRC OSC/2 (8MHz)
5. HSIRC OSC/4 (4MHz)
6. HSIRC OSC/8 (2MHz)
7. HSIRC OSC/16 (1MHz)
8. HSIRC OSC/32 (0.5MHz)
9. SubCrystal Oscillator (32.768kHz)
10. Internal LSIRC Oscillator (128kHz)

## 7.2 Block Diagram

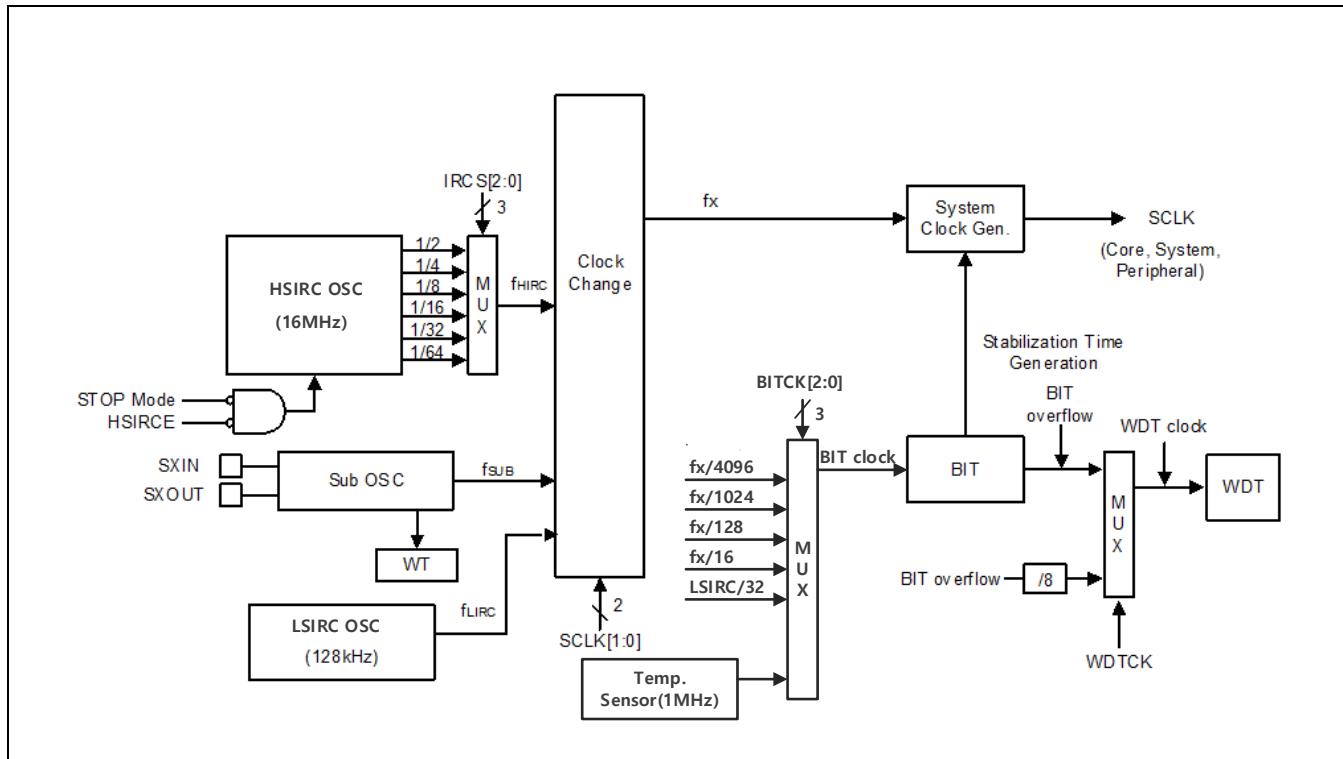


Figure 24. Clock Generator Block Diagram

## 7.3 Register Map

Table 11. Clock Generator Register Map

Name	Address	Direction	Default	Description
SCCR	8AH	R/W	00H	System and Clock Control Register
OSCCR	C8H	R/W	28H	Oscillator Control Register

## 7.4 Clock Generator Register Description

The clock generator register uses clock control for system operation. The clock generation consists of System and clock control register and oscillator control register.

## 7.5 Register Description for Clock Generator

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

7	6	5	4	3	2	1	0
-	-	-	-	-	-	SCLK1	SCLK0
-	-	-	-	-	-	RW	RW

Initial value: 00H

SCLK [1:0]		System Clock Selection Bit		
SCLK1	SCLK0	Description		
0	0	Internal 16MHz RC OSC ( $f_{HSIRC}$ ) for system clock		
0	1	Reserved		
1	0	External Sub OSC ( $f_{SUB}$ ) for system clock		
1	1	Internal 128kHz RC OSC ( $f_{LSIRC}$ ) for system clock		

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

7	6	5	4	3	2	1	0
TIRCE	LSIRCE	IRCS2	IRCS1	IRCS0	HSIRCE	-	SCLKE
R/W	R/W	R/W	R/W	R/W	R/W	-	R/W

Initial value: 28H

TIRCE	Control the Operation of the Touch (16MHz) Internal RC Oscillator			
	0 Disable operation of TIRC OSC			
	1 Enable operation of TIRC OSC			
LSIRCE	Control the Operation of the Low Frequency (128kHz) internal RC Oscillator at Stop mode			
	0 Disable operation of LSIRC OSC			
	1 Enable operation of LSIRC OSC			
IRCS[2:0]	Internal RC Oscillator Post-divider Selection			
	IRCS2	IRCS1	IRCS0	Description
	0	0	0	$f_{HSIRC} /32$ (0.5MHz)
	0	0	1	$f_{HSIRC} /16$ (1MHz)
	0	1	0	$f_{HSIRC} /8$ (2MHz)
	0	1	1	$f_{HSIRC} /4$ (4MHz)
	1	0	0	$f_{HSIRC} /2$ (8MHz)
	1	0	1	$f_{HSIRC}$ (16MHz)
	1	1	0	Test only
	Other Values			reserved
HSIRCE	Control the Operation of the High Frequency (16MHz) Internal RC Oscillator			
	0 Enable operation of HSIRC OSC			
	1 Disable operation of HSIRC OSC			
SCLKE	Control the Operation of the External Sub Oscillator			
	0 Disable operation of SX-TAL			
	1 Enable operation of SX-TAL			

## 8 Basic Interval Timer

### 8.1 Overview

The A96T418 has one 8-bit basic interval timer that is free-run and can't stop. Block diagram is shown in Figure 25. In addition, the basic interval timer generates the time base for watchdog timer counting. It also provides a basic interval timer interrupt (BITIFR).

The A96T418 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

### 8.2 Block Diagram

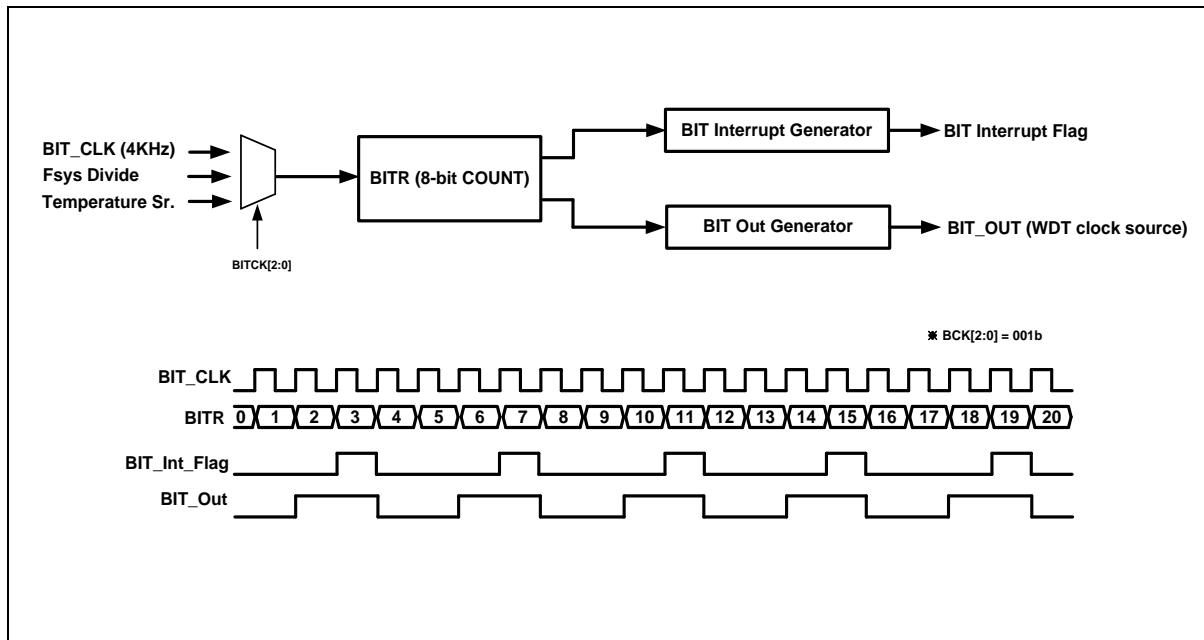


Figure 25. Basic Interval Timer Block Diagram

### 8.3 Register Map

**Table 12. Basic Interval Timer Register Map**

Name	Address	Direction	Default	Description
BITCNT	8CH	R	00H	Basic Interval Timer Counter Register
BITCR	8BH	R/W	45H	Basic Interval Timer Control Register

### 8.4 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.

## 8.5 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	BITCK2	BITCK1	BITCK0	BCLR	BCK2	BCK1	BCK0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 45H

**BITIFR** When BIT Interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or signal.  
Writing "1" has no auto clear by INT\_ACK effect.

- 0 BIT interrupt no generation
- 1 BIT interrupt generation

**BITCK[2:0]** Select BIT clock source

BITCK2	BITCK1	BITCK0	Description
0	0	0	fx/4096
0	0	1	fx/1024
0	1	0	fx/128
0	1	1	fx/16
1	0	0	LSIRC/32 (Default)
1	0	1	LSIRC/32
1	1	0	LSIRC/32
1	1	1	Temperature Sensor

**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 (fx= LSIRC 128k)
0	0	0	0.5ms (BIT Clock * 2)
0	0	1	1ms (BIT Clock * 4)
0	1	0	2ms (BIT Clock * 8)
0	1	1	4ms (BIT Clock * 16)
1	0	0	8ms (BIT Clock * 32)
1	0	1	16ms (BIT Clock * 64) (default)
1	1	0	32ms (BIT Clock * 128)
1	1	1	64ms (BIT Clock * 256)

## 9 Watchdog Timer

### 9.1 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 watchdog 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 watchdog 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

$$\text{WDT Interrupt Interval} = (\text{BIT Interrupt Interval}) \times (\text{WDTDR Value}+1)$$

### 9.2 WDT Interrupt Timing Waveform

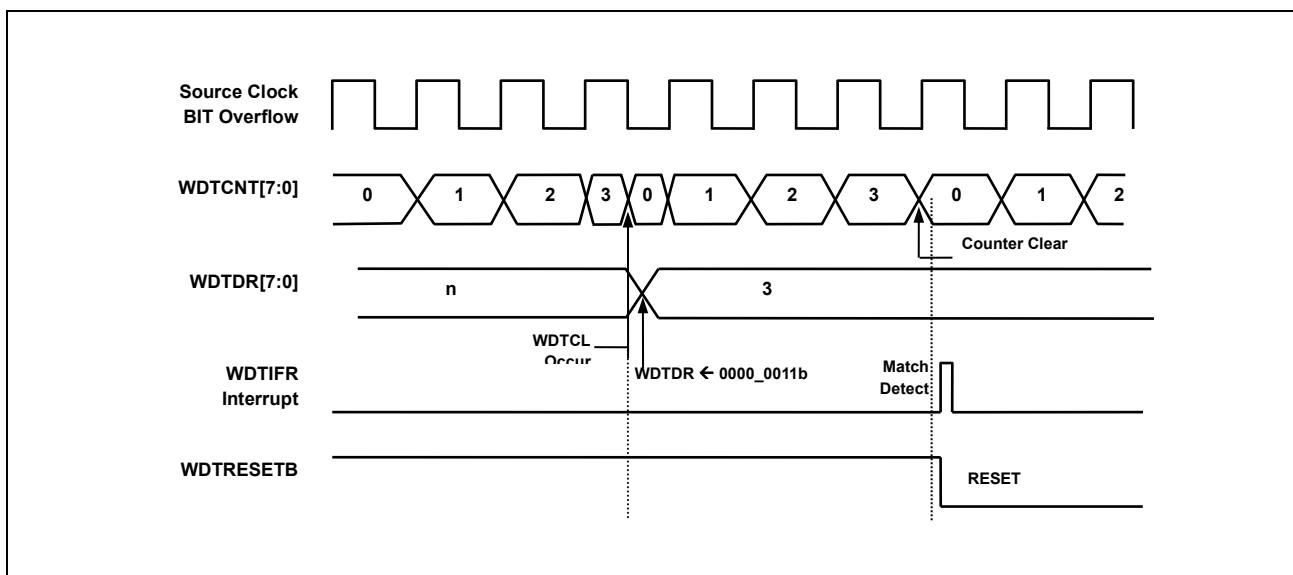


Figure 26. Watchdog Timer Interrupt Timing Waveform

### 9.3 Block Diagram

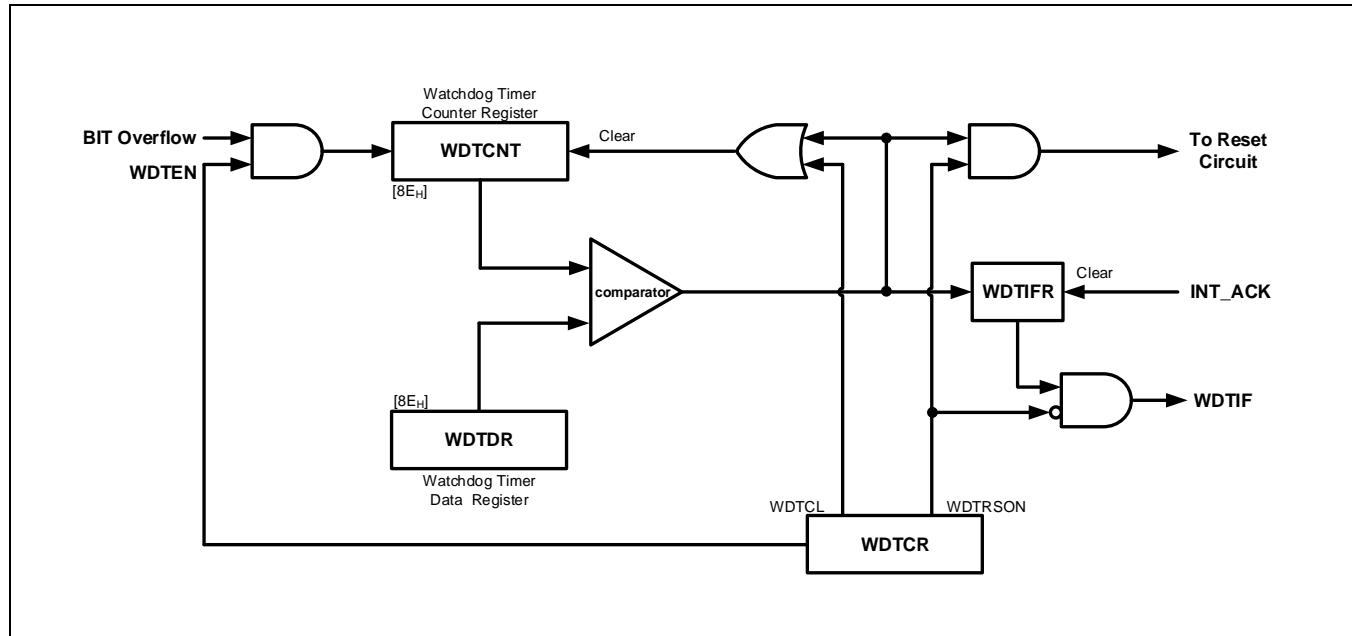


Figure 27. Watchdog Timer Block Diagram

### 9.4 Register Map

Table 13. Watchdog Timer Register Map

Name	Address	Direction	Default	Description
WDTCNT	8EH	R	00H	Watchdog Timer Counter Register
WDTDR	8EH	W	FFH	Watchdog Timer Data Register
WDTCR	8DH	R/W	00H	Watchdog Timer Control Register

### 9.5 Watchdog Timer Register Description

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

## 9.6 Register Description for Watchdog Timer

### WDTCNT (Watchdog 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 (Watchdog 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)

**NOTES:**

- Do not write "0" in the WDTDR register.

### WDTCR (Watchdog 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 Watchdog 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

1 BIT overflow/8 for WDT clock

WDTIFR When WDT Interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or auto clear by INT\_ACK signal. Writing "1" has no effect.

0 WDT Interrupt no generation

1 WDT Interrupt generation

## 10 Watch Timer

### 10.1 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.

### 10.2 Block Diagram

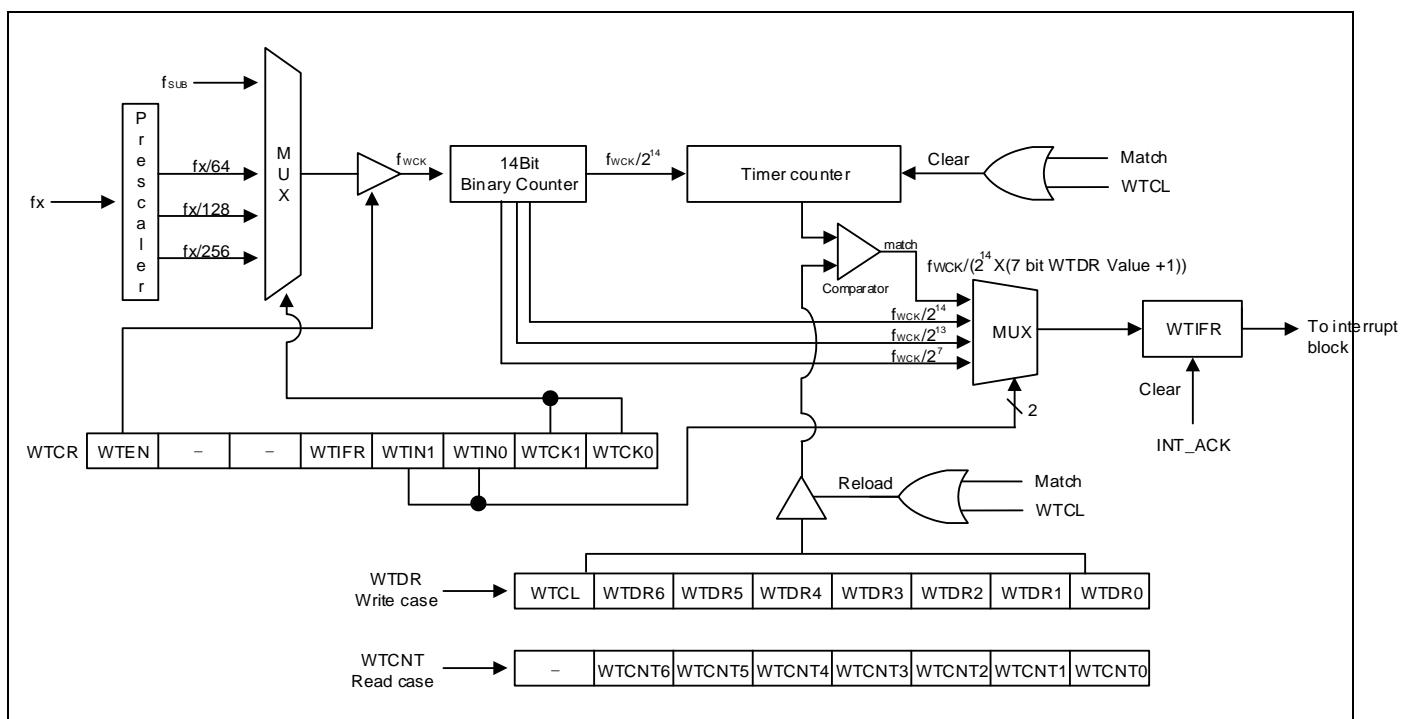


Figure 28. Watch Timer Block Diagram

### 10.3 Register Map

**Table 14. Watch Timer Register Map**

Name	Address	Direction	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

### 10.4 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 6-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).

### 10.5 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))

**NOTES:**

- 1. Do not write "0" in the WTDR register.

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

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

Initial value: 00H

WTEN	Control Watch Timer		
	0 Disable		
	1 Enable		
WTIFR	When WT Interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or automatically clear by INT_ACK signal. Writing "1" has no effect.		
	0 WT Interrupt no generation		
	1 WT Interrupt generation		
WTIN[1:0]	Determine interrupt interval		
	WTIN1	WTIN0	Description
	0	0	fwck/2^7
	0	1	fwck/2^13
	1	0	fwck/2^14
	1	1	fwck/(2^14 x (7bit WTDR Value+1))
WTCK[1:0]	Determine Source Clock		
	WTCK1	WTCK0	Description
	0	0	fSUB
	0	1	fx/256
	1	0	fx/128
	1	1	fx/64

**NOTES:**

1. fx— System clock frequency (Where fx= 4.19MHz)
2. fSUB— Sub clock oscillator frequency (32.768kHz)
3. fwck— Selected Watch timer clock

## 11 Timer 0/1/2/3/4/5

### 11.1 Timer 0

#### 11.1.1 Overview

The 8-bit timer 0 consists of multiplexer, timer 0 counter register, timer 0 data register, timer 0 capture data register and timer 0 control register (T0CNT, T0DR, T0CDR, and T0CR).

It has three operating modes:

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

The timer/counter 0 can be clocked by an internal or an external clock source (EC0). The clock source is selected by clock selection logic which is controlled by the clock selection bits (T0CK[2:0]).

TIMER0 clock source:  $f_x/2$ , 4, 8, 32, 128, 512, 2048 and EC0

In the capture mode, by EINT10, the data is captured into input capture data register (T0CDR). In timer/counter mode, whenever counter value is equal to T0DR, T0O port toggles. Also the timer 0 outputs PWM waveform through PWM0O port in the PWM mode.

**Table 15. Timer 0 Operating Modes**

<b>T0EN</b>	<b>T0MS[1:0]</b>	<b>T0CK[2:0]</b>	<b>Timer 0</b>
1	00	XXX	8-bit Timer/Counter Mode
1	01	XXX	8-bit PWM Mode
1	1X	XXX	8-bit Capture Mode

### 11.1.2 8-bit Timer/Counter Mode

The 8-bit timer/counter mode is selected by control register as shown in Figure 29.

The 8-bit timer have counter and data register. The counter register is increased by internal or external clock input. Timer 0 can use the input clock with one of 2, 4, 8, 32, 128, 512 and 2048 prescaler division rates ( $T0CK[2:0]$ ). When the value of  $T0CNT$  and  $T0DR$  is identical in timer 0, a match signal is generated and the interrupt of Timer 0 occurs.  $T0CNT$  value is automatically cleared by match signal. It can be also cleared by software ( $T0CC$ ).

The external clock (EC0) counts up the timer at the rising edge. If the EC0 is selected as a clock source by  $T0CK[2:0]$ , EC0 port should be set to the input port by P25IO bit.

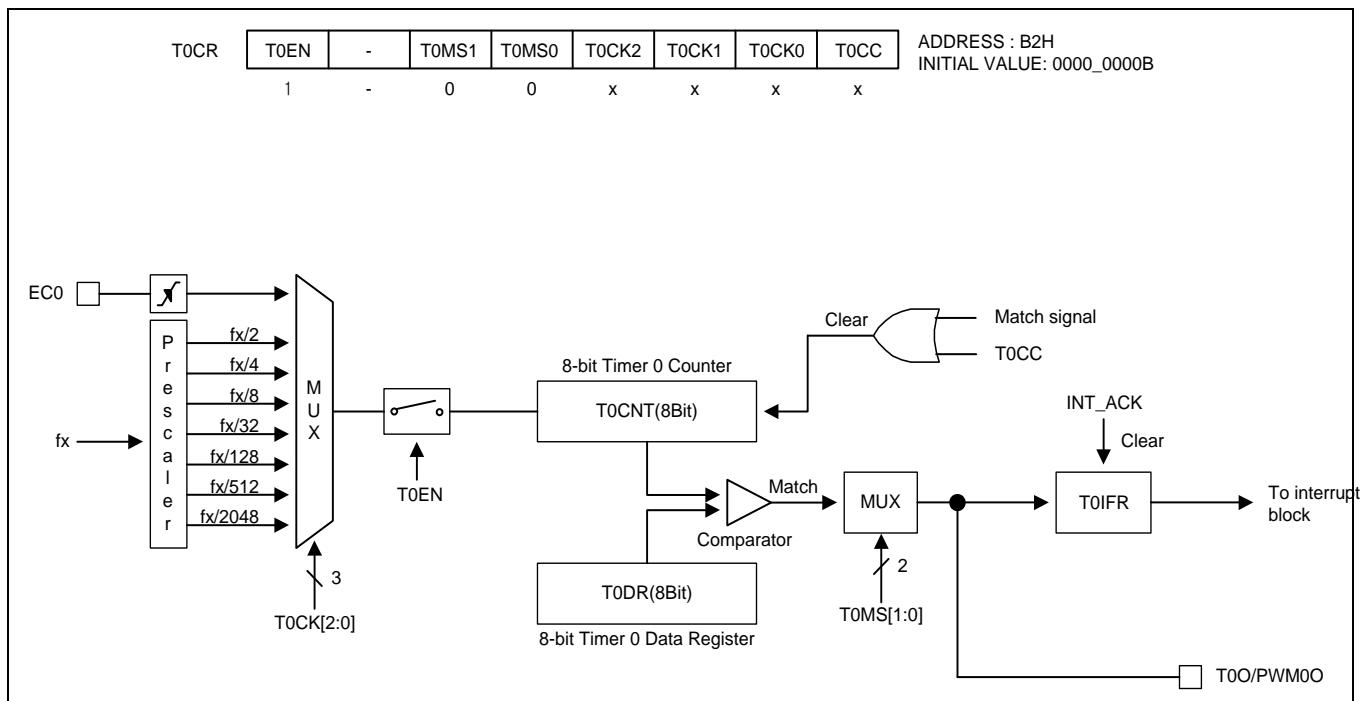


Figure 29. 8-bit Timer/Counter Mode for Timer 0

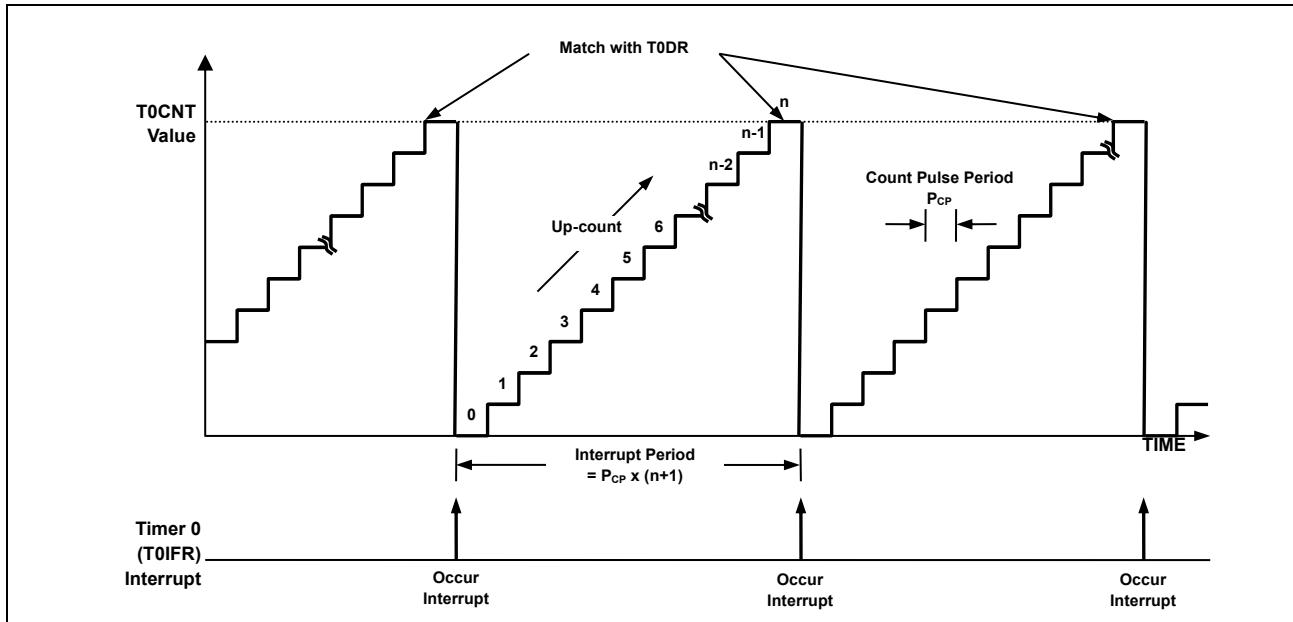


Figure 30. 8-bit Timer/Counter 0 Example

### 11.1.3 8-bit PWM Mode

The timer 0 has a high speed PWM (Pulse Width Modulation) function. In PWM mode, T0O/PWM0O pin outputs up to 8-bit resolution PWM output. This pin should be configured as a PWM output by setting the T0O/PWM0O function by P2FSRH[7:6] bits. In the 8-bit timer/counter mode, a match signal is generated when the counter value is identical to the value of T0DR. When the value of T0CNT and T0DR is identical in timer 0, a match signal is generated and the interrupt of timer 0 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 overflow interrupt is generated whenever a counter overflow occurs. T0CNT value is cleared by software (T0CC) bit.

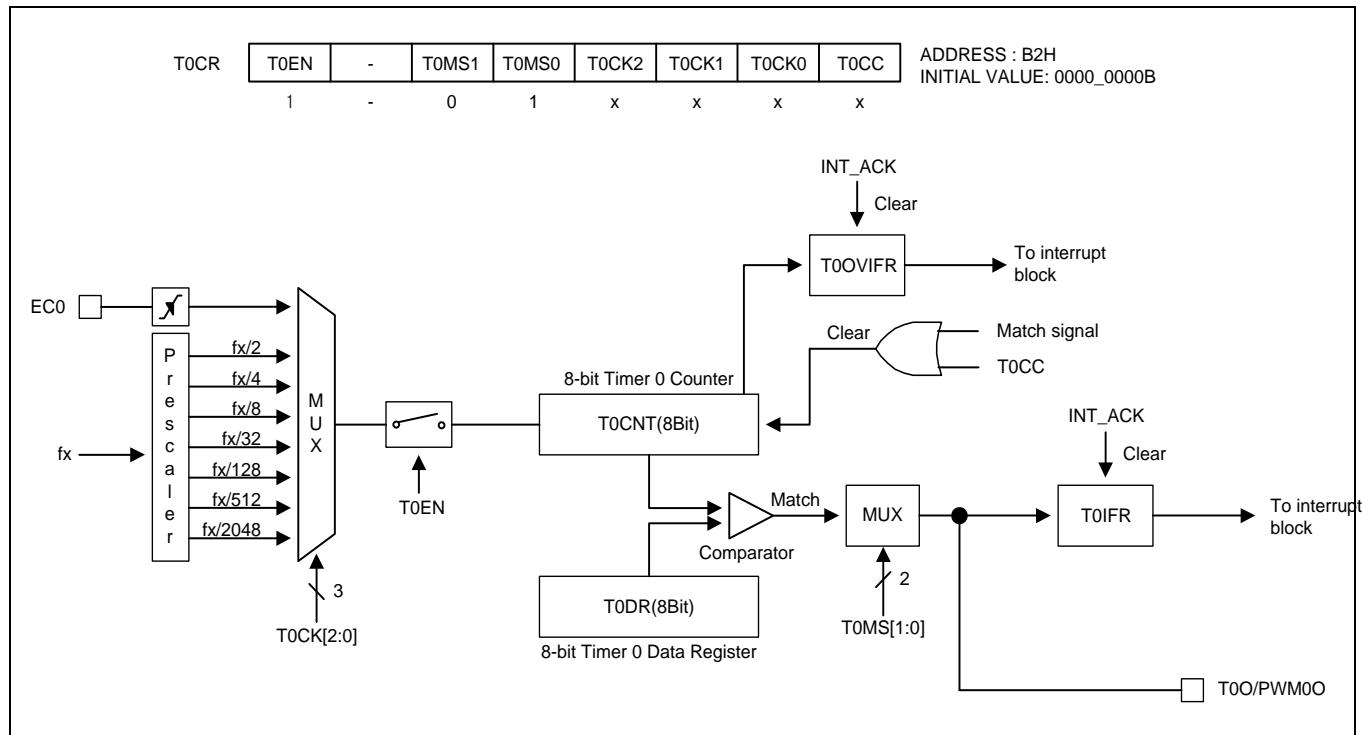


Figure 31. 8-bit PWM Mode for Timer 0

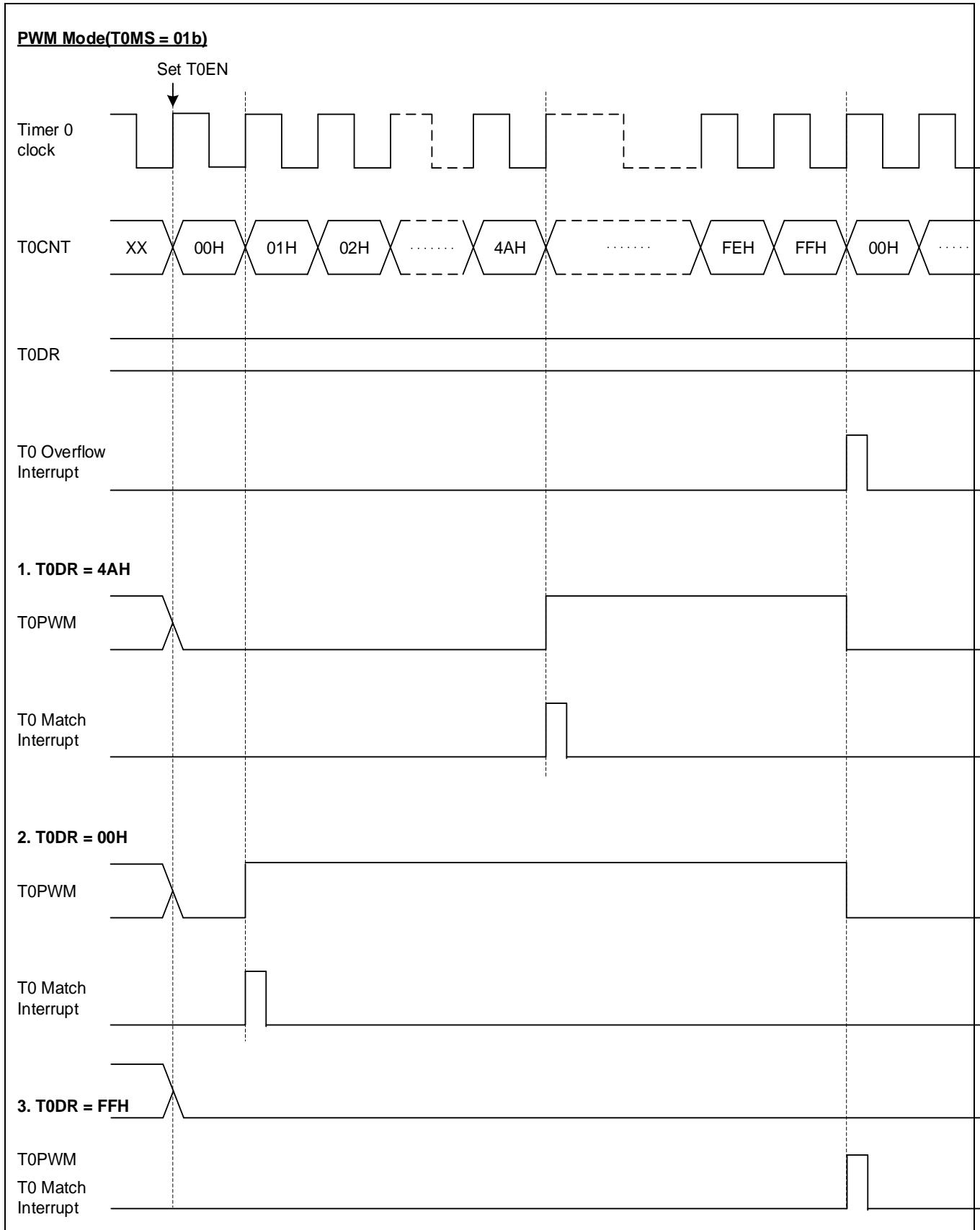


Figure 32. PWM Output Waveforms in PWM Mode for Timer 0

### 11.1.4 8-bit Capture Mode

The timer 0 capture mode is set by T0MS[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 T0CNT is equal to T0DR. T0CNTvalue is automatically cleared by match signal and it can be also cleared by software (T0CC).

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 T0CDR. In the timer 0 capture mode, timer 0 output (T0O) waveform is not available.

T0CDR and T0DR are in the same address. In the capture mode, reading operation reads T0CDR, not T0DR and writing operation will update T0DR.

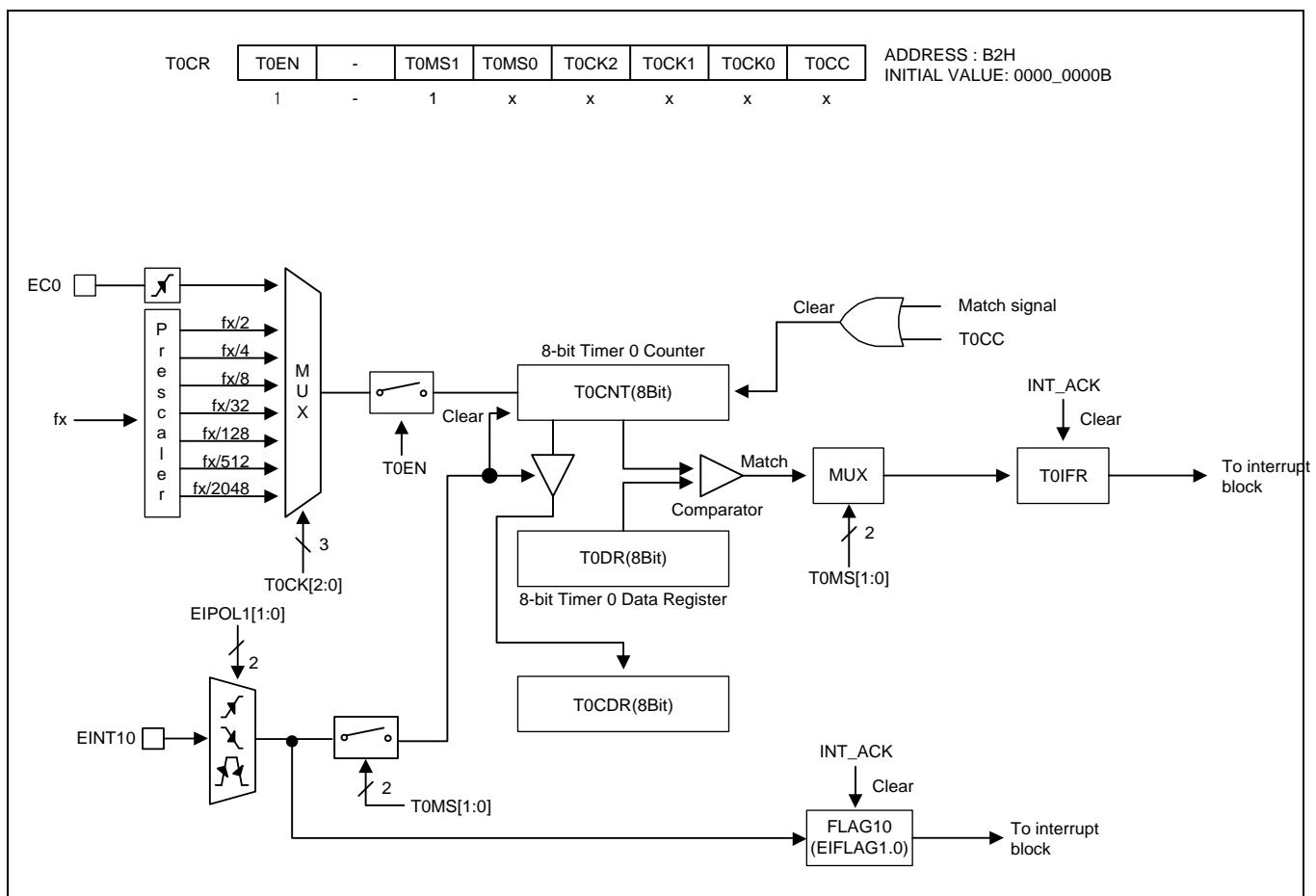


Figure 33. 8-bit Capture Mode for Timer 0

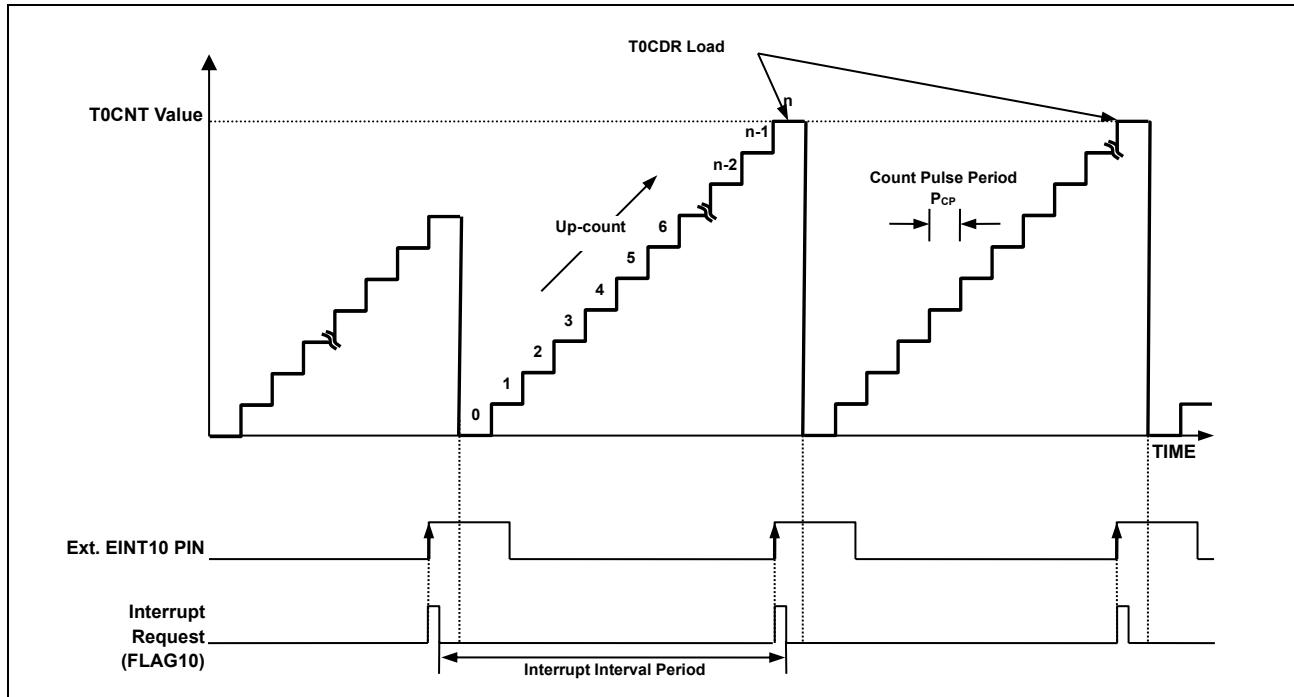


Figure 34. Input Capture Mode Operation for Timer 0

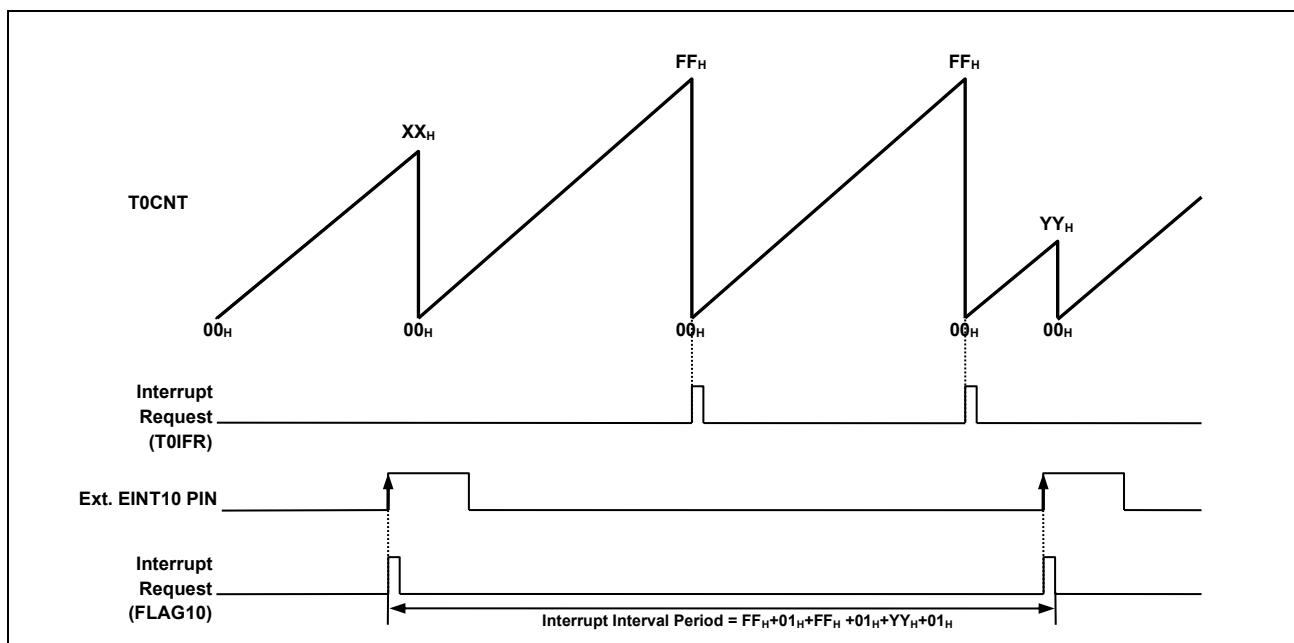


Figure 35. Express Timer Overflow in Capture Mode

### 11.1.5 Block Diagram

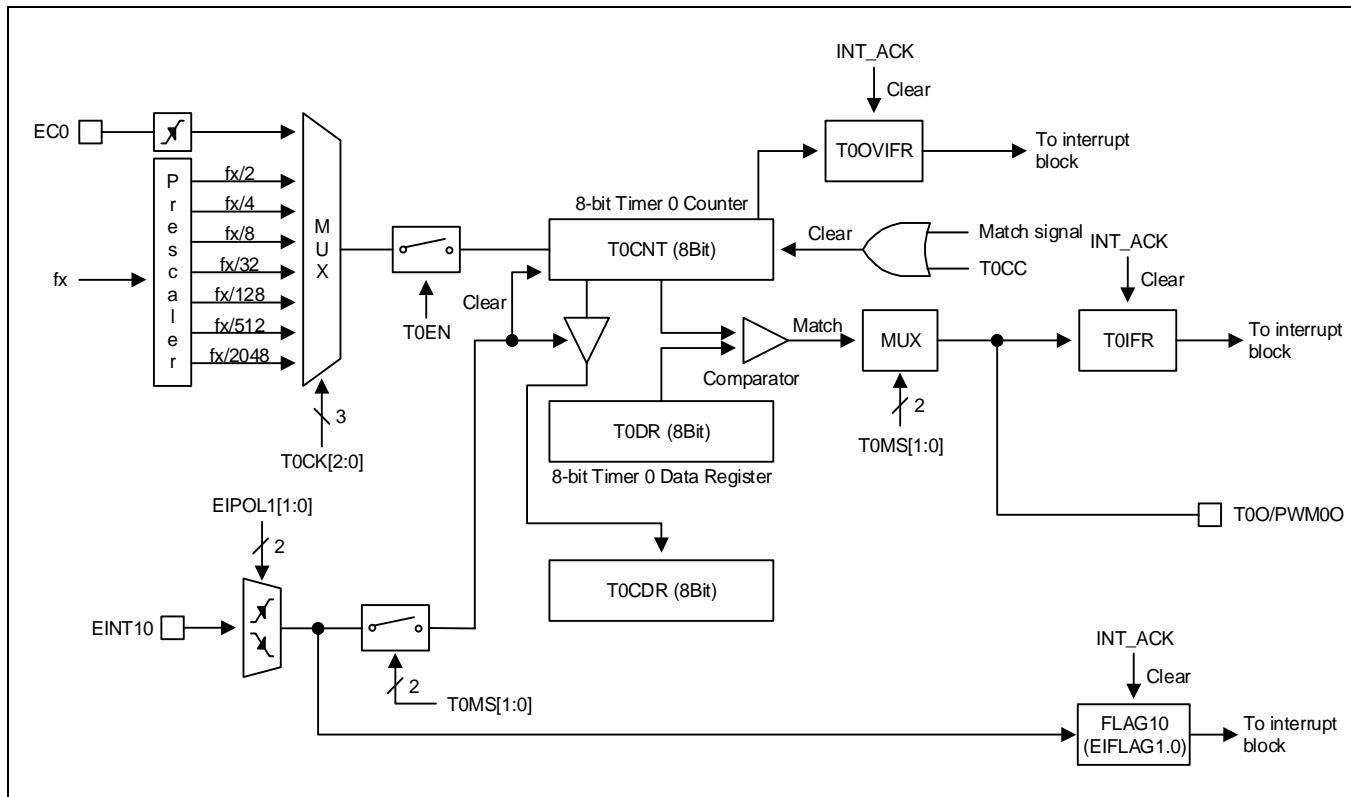


Figure 36. 8-bit Timer 0 Block Diagram

### 11.1.6 Register Map

Table 16. Timer 0 Register Map

Name	Address	Direction	Default	Description
T0CNT	B3H	R	00H	Timer 0 Counter Register
T0DR	B4H	R/W	FFH	Timer 0 Data Register
T0CDR	B4H	R	00H	Timer 0 Capture Data Register
T0CR	B2H	R/W	00H	Timer 0 Control Register

### 11.1.7 Timer/Counter 0 Register Description

The timer/counter 0 register consists of timer 0 counterregister (T0CNT), timer 0 data register (T0DR), timer 0 capture dataregister (T0CDR), andtimer 0 controlregister (T0CR). T0IFR and T0OVIFR bits are in the external interrupt flag 1 register (EIFLAG1).

### 11.1.8 Register Description for Timer/Counter 0

#### T0CNT (Timer 0 Counter Register): B3H

7	6	5	4	3	2	1	0
T0CNT7	T0CNT6	T0CNT5	T0CNT4	T0CNT3	T0CNT2	T0CNT1	T0CNT0
R	R	R	R	R	R	R	R

Initial value: 00H

T0CNT[7:0]      T0 Counter

#### T0DR (Timer 0 Data Register): B4H

7	6	5	4	3	2	1	0
T0DR7	T0DR6	T0DR5	T0DR4	T0DR3	T0DR2	T0DR1	T0DR0
R/W							

Initial value: FFH

T0DR[7:0]      T0 Data

#### T0CDR (Timer 0 Capture Data Register: Read Case, Capture mode only): B4H

7	6	5	4	3	2	1	0
T0CDR7	T0CDR6	T0CDR5	T0CDR4	T0CDR3	T0CDR2	T0CDR1	T0CDR0
R	R	R	R	R	R	R	R

Initial value: 00H

T0CDR[7:0]      T0 Capture Data

**T0CR (Timer 0 Control Register): B2H**

7	6	5	4	3	2	1	0
TOEN	-	T0MS1	T0MS0	T0CK2	T0CK1	T0CK0	T0CC
RW	-	RW	RW	RW	RW	RW	RW

Initial value: 00H

T0EN	Control Timer 0										
	0 Timer 0 disable										
	1 Timer 0 enable										
T0MS[1:0]	Control Timer 0 Operation Mode										
	T0MS	T0MS	Description								
	1	0									
	0	0	Timer/counter mode								
	0	1	PWM mode								
	1	x	Capture mode								
T0CK[2:0]	Select Timer 0 clock source. fx is a system clock frequency										
	T0CK2	T0CK	T0CK	Description							
	1	0									
	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 (EC0)							
T0CC	Clear timer 0 Counter										
	0	No effect									
	1	Clear the Timer 0 counter (When write, automatically cleared "0" after being cleared counter)									

**NOTES:**

1. Match Interrupt is generated in Capture mode.
2. Refer to the external interrupt flag 1 register (EIFLAG1) for the T0 interrupt flags.

## 11.2 Timer 1

### 11.2.1 Overview

The 16-bit timer 1 consists of multiplexer, timer 1 A data register high/low, timer 1 B data register high/low and timer 1 control register high/low (T1ADR<sub>H</sub>, T1ADR<sub>L</sub>, T1BDR<sub>H</sub>, T1BDR<sub>L</sub>, T1CR<sub>H</sub>, T1CR<sub>L</sub>).

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 1 can be clocked by an internal or an external clock source (EC1). The clock source is selected by clock selection logic which is controlled by the clock selection bits (T1CK[2:0]).

TIMER 1 clock source: f<sub>x</sub>/1, 2, 4, 8, 64, 512, 2048 and EC1

In the capture mode, by EINT11, the data is captured into input capture data register (T1BDR<sub>H</sub>/T1BDR<sub>L</sub>). Timer 1 outputs the comparison result between counter and data register through T1O port in timer/counter mode. Also Timer 1 outputs PWM wave form through PWM1O port in the PPG mode.

**Table 17. Timer 1 Operating Modes**

T1EN	P1FSRL[1:0]	T1MS[1:0]	T1CK[2:0]	Timer 1
1	11	00	XXX	16-bit Timer/Counter Mode
1	00	01	XXX	16-bit Capture Mode
1	11	10	XXX	16-bit PPG Mode(one-shot mode)
1	11	11	XXX	16-bit PPG Mode(repeat mode)

### 11.2.2 16-bit Timer/Counter Mode

The 16-bit timer/counter mode is selected by control register as shown in Figure 37.

The 16-bit timer have counter and data register. The counter register is increased by internal or external clock input. Timer 1 can use the input clock with one of 1, 2, 4, 8, 64, 512 and 2048 prescaler division rates (T1CK[2:0]). When the value of T1CNTH, T1CNTL and the value of T1ADR<sub>H</sub>, T1ADR<sub>L</sub> are identical in Timer 1 respectively, a match signal is generated and the interrupt of Timer1occurs. The T1CNTH, T1CNTL value is automatically cleared by match signal. It can be also cleared by software (T1CC).

The external clock (EC1) counts up the timer at the rising edge. If the EC1 is selected as a clock source by T1CK[2:0], EC1 port should be set to the input port by P23IO bit.

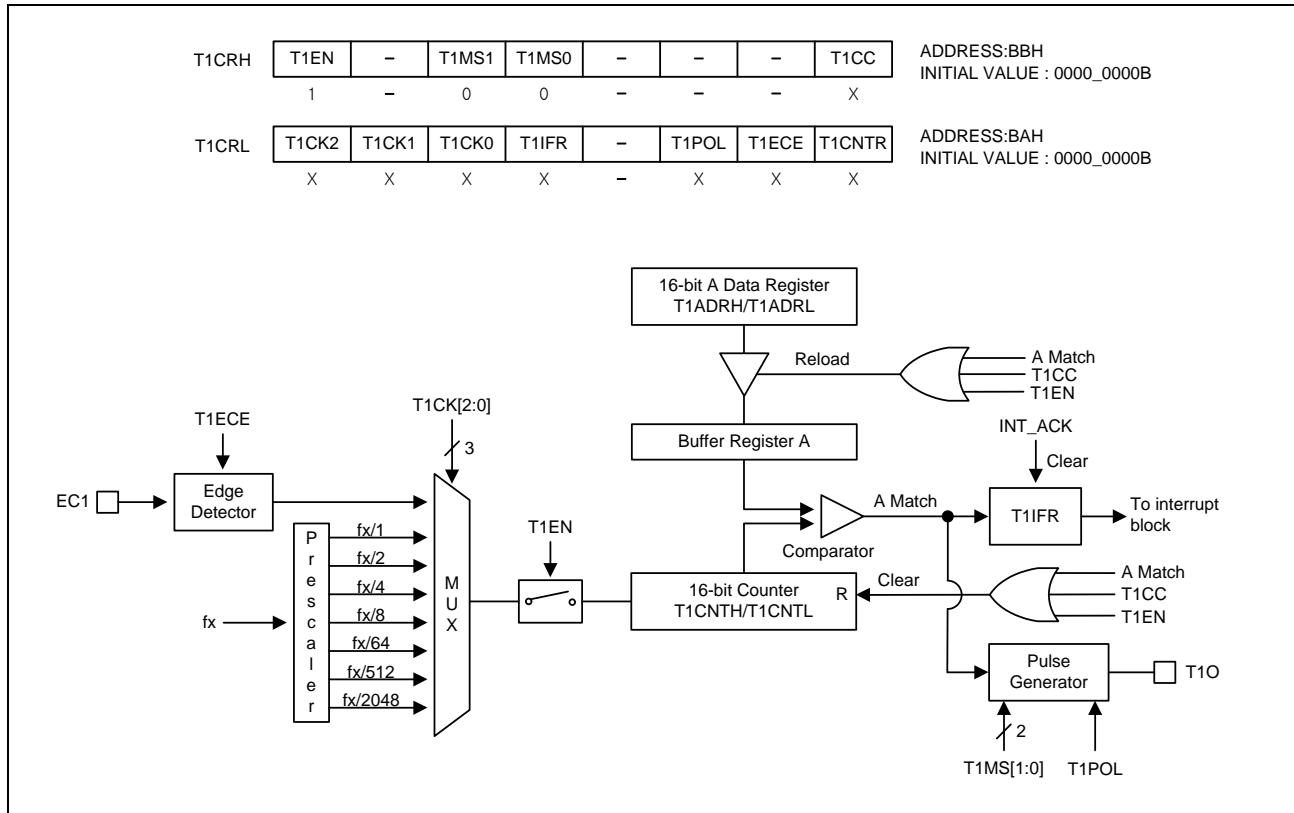


Figure 37. 16-bit Timer/Counter Mode for Timer 1

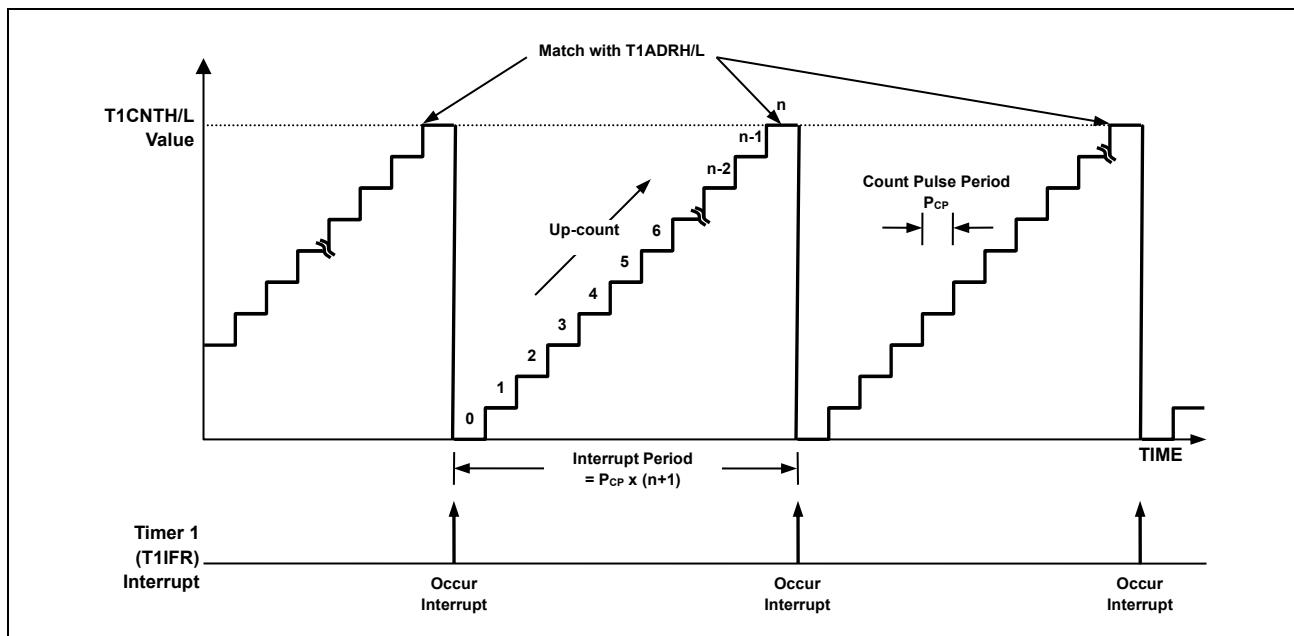


Figure 38. 16-bit Timer/Counter 1 Example

### 11.2.3 16-bit Capture Mode

The 16-bit timer 1 capture mode is set by T1MS[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 T1CNTH/T1CNTL is equal to T1ADRH/T1ADRL. The T1CNTH, T1CNTL values are automatically cleared by match signal. It can be also cleared by software (T1CC).

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 T1BDRH/T1BDRL.

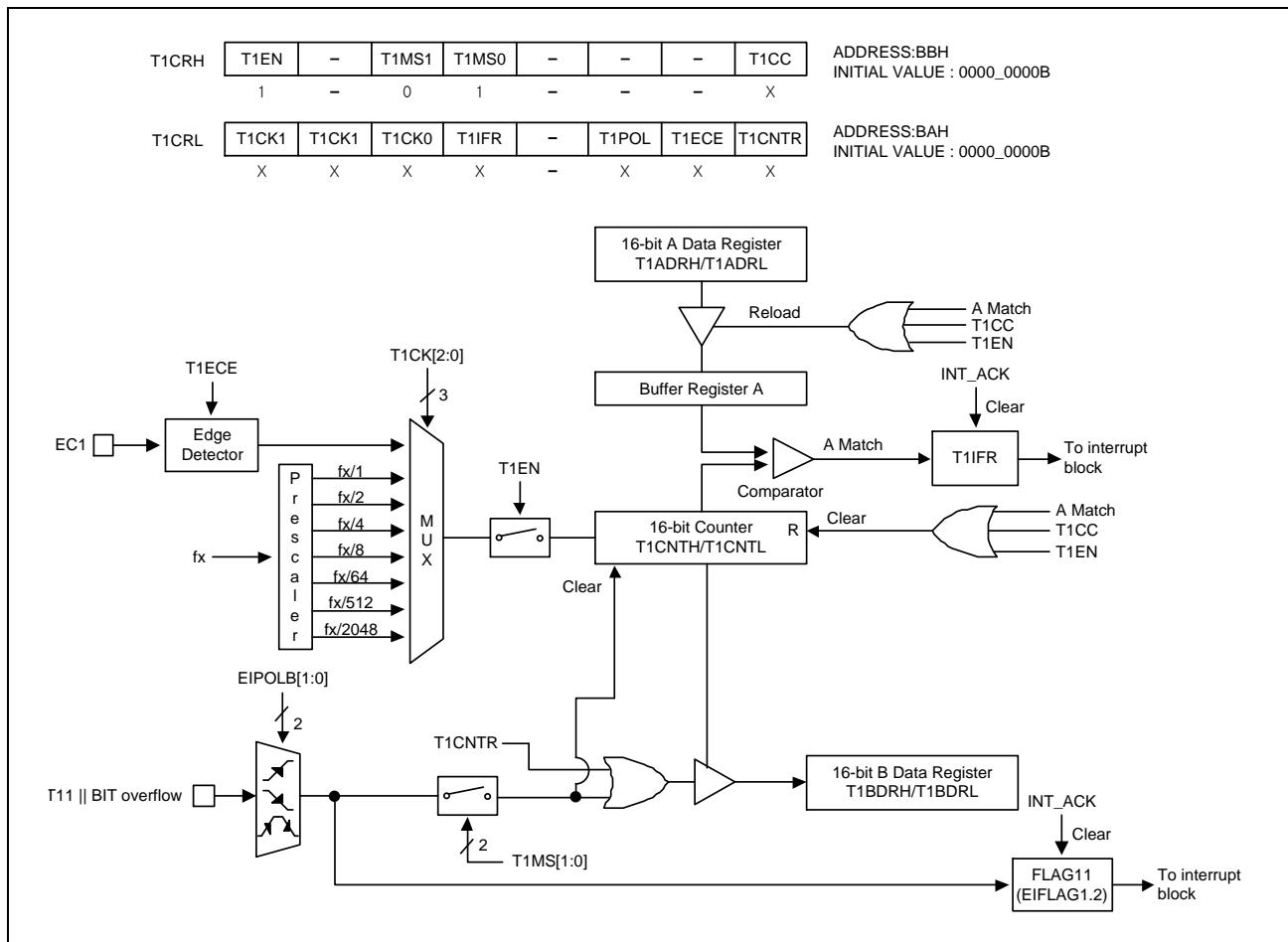


Figure 39. 16-bit Capture Mode for Timer 1

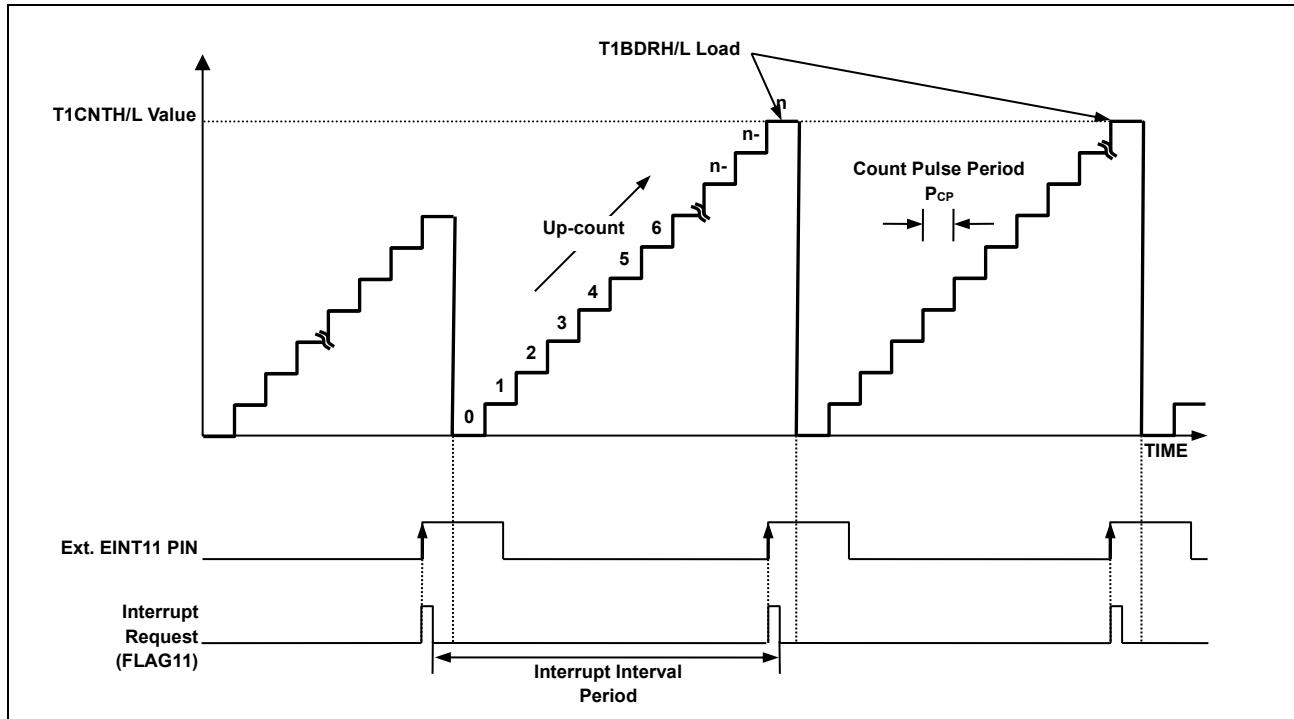


Figure 40. Input Capture Mode Operation for Timer 1

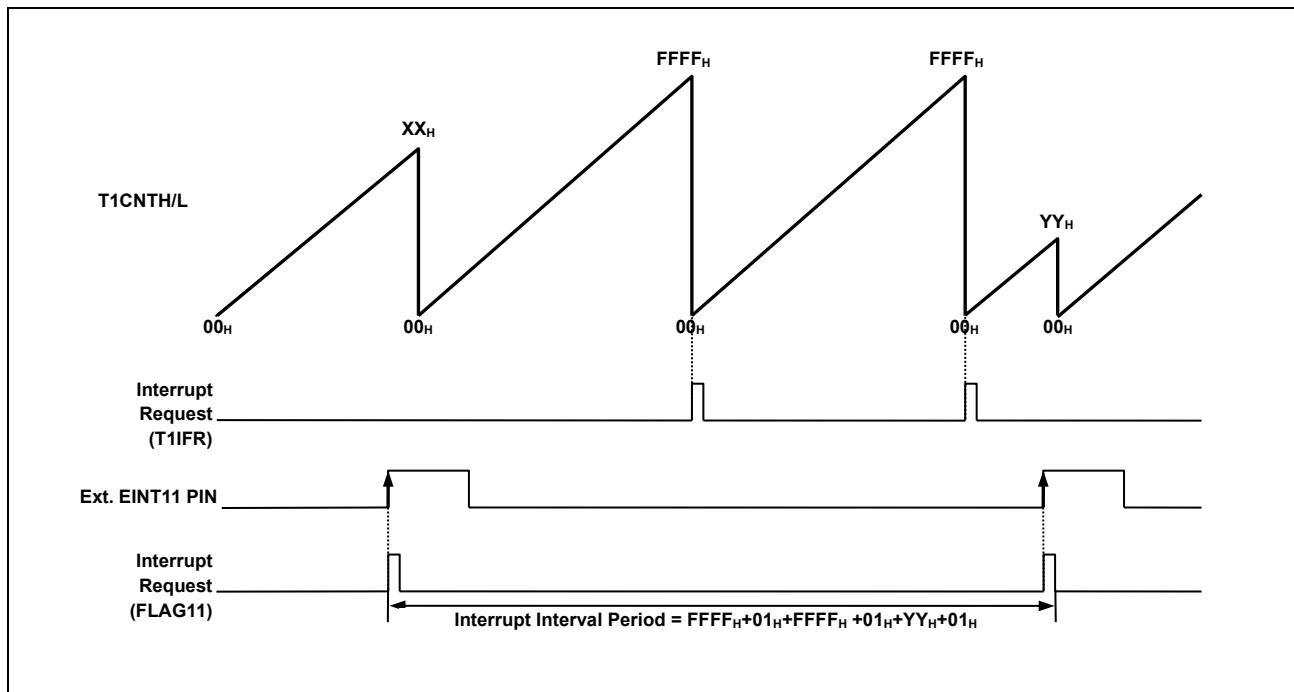


Figure 41. Express Timer Overflow in Capture Mode

### 11.2.4 16-bit PPG Mode

The timer 1 has a PPG (Programmable Pulse Generation) function. In PPG mode, T1O/PWM1O pin outputs up to 16-bit resolution PWM output. This pin should be configured as a PWM output by setting P1FSRL[3:2] to '11'. The period of the PWM output is determined by the T1ADRH/T1ADRL. And the duty of the PWM output is determined by the T1BDRH/T1BDRL.

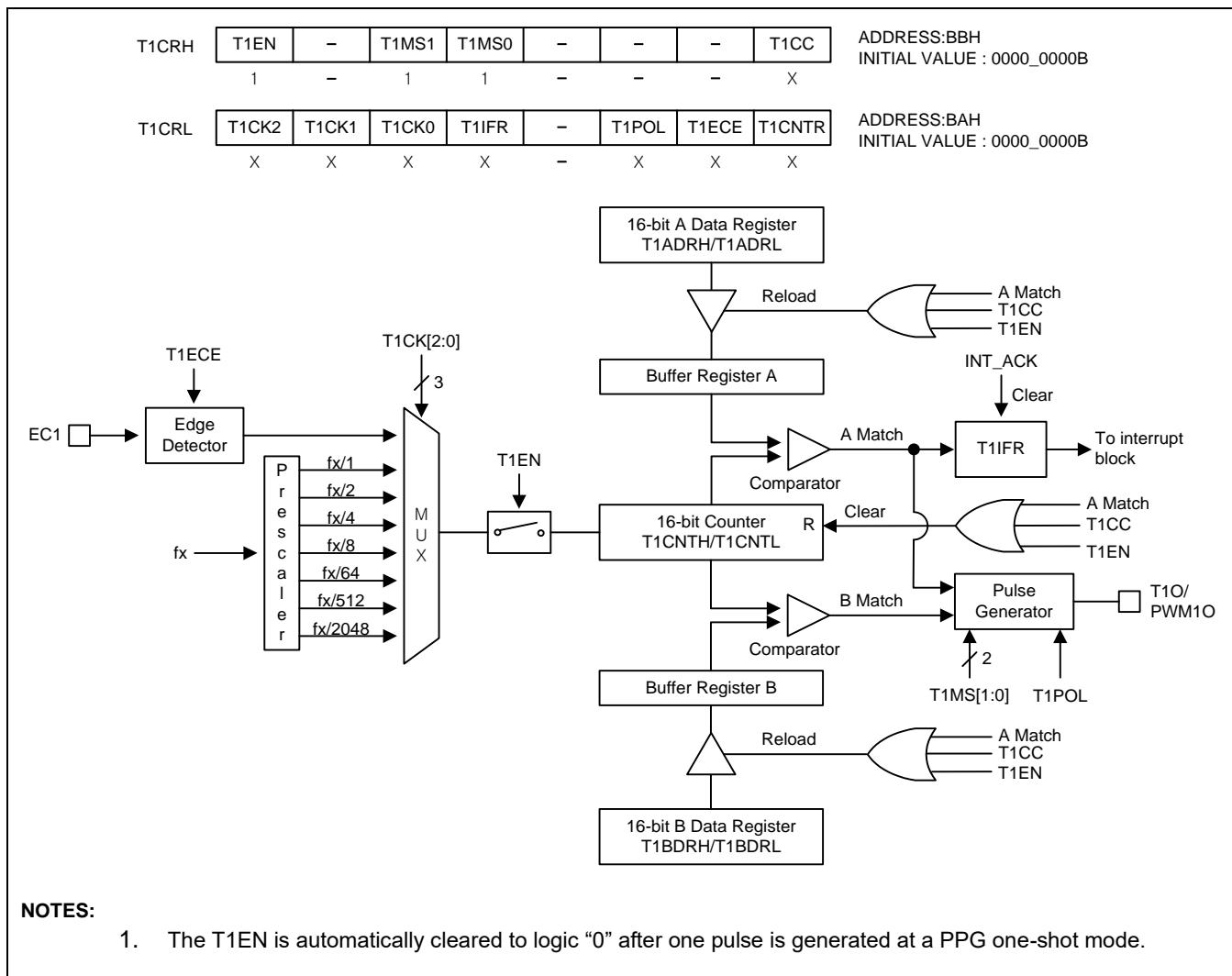


Figure 42. 16-bit PPG Mode for Timer 1

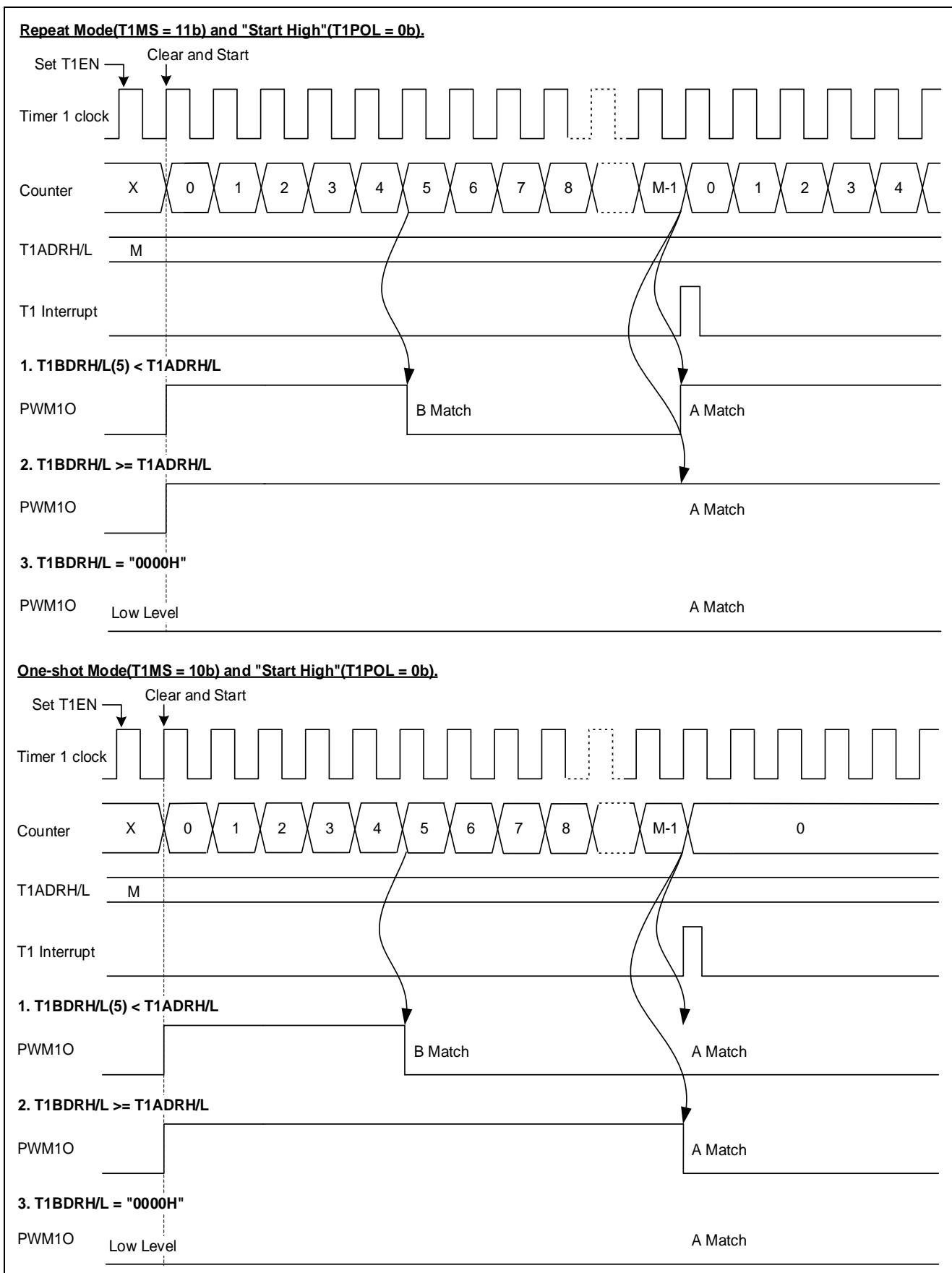
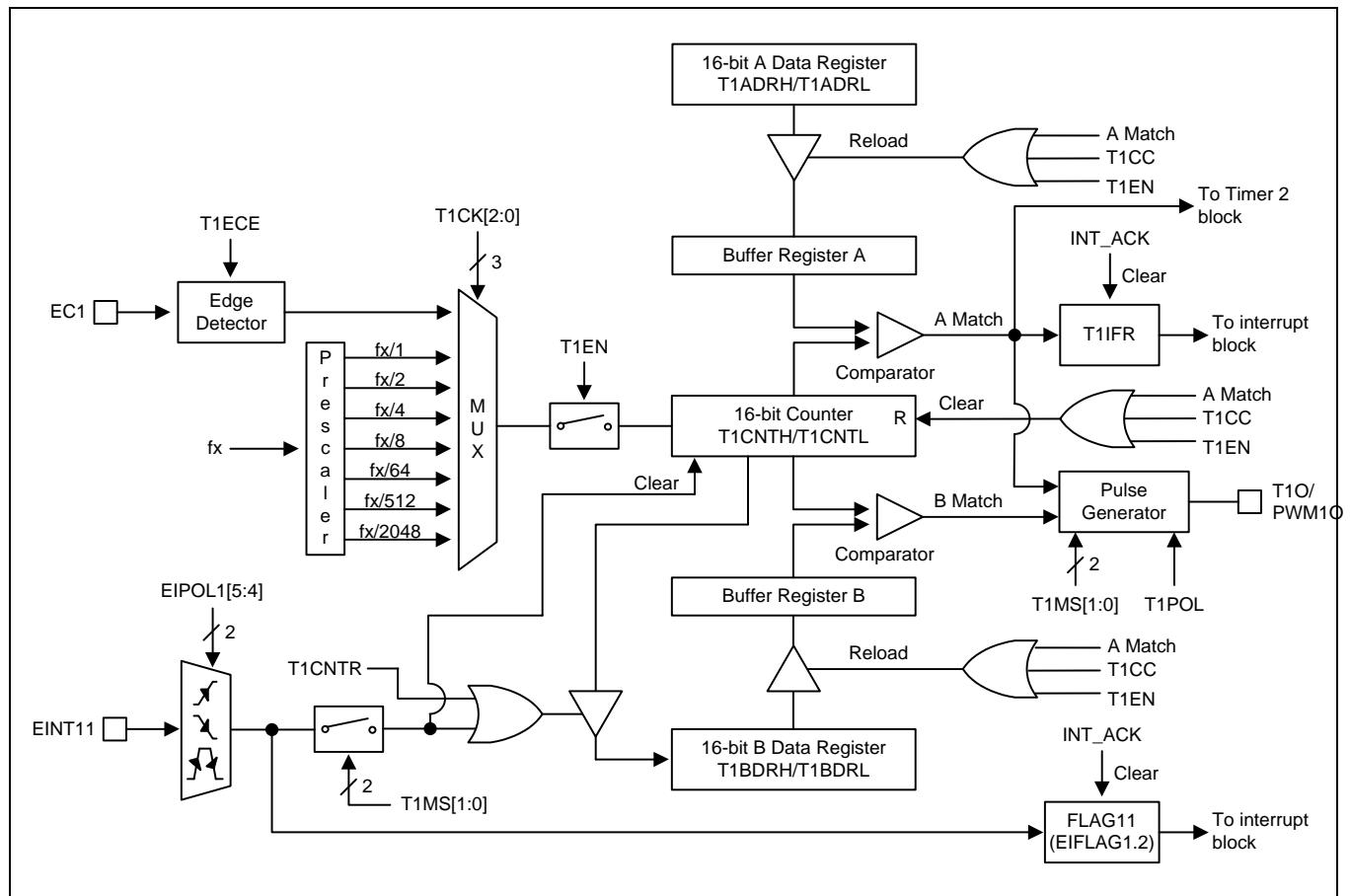


Figure 43. 16-bit PPG Mode Timming chart for Timer 1

### 11.2.5 Block Diagram



**Figure 44. 16-bit Timer 1 Block Diagram**

### 11.2.6 Register Map

**Table 18. Timer 1 Register Map**

Name	Address	Direction	Default	Description
T1ADRH	BDH	R/W	FFH	Timer 1 A Data High Register
T1ADRL	BCH	R/W	FFH	Timer 1 A Data Low Register
T1BDRH	BFH	R/W	FFH	Timer 1 B Data High Register
T1BDRL	BEH	R/W	FFH	Timer 1 B Data Low Register
T1CRH	BBH	R/W	00H	Timer 1 Control High Register
T1CRL	BAH	R/W	00H	Timer 1 Control Low Register

### 11.2.7 Timer/Counter 1 Register Description

The timer/counter 1 register consists of timer 1 A data high register (T1ADRH), timer 1 A data low register (T1ADRL), timer 1 B data high register (T1BDRH), timer 1 B data low register (T1BDRL), timer 1 control high register (T1CRH) and timer 1 control low register (T1CRL).

### 11.2.8 Register Description for Timer/Counter 1

#### T1ADRH (Timer 1 A Data High Register): BDH

7	6	5	4	3	2	1	0
T1ADRH7	T1ADRH6	T1ADRH5	T1ADRH4	T1ADRH3	T1ADRH2	T1ADRH1	T1ADRHO
R/W							

Initial value: FFH

T1ADRH[7:0] T1 A Data High Byte

#### T1ADRL (Timer 1 A Data Low Register): BCH

7	6	5	4	3	2	1	0
T1ADRL7	T1ADRL6	T1ADRL5	T1ADRL4	T1ADRL3	T1ADRL2	T1ADRL1	T1ADRL0
R/W							

Initial value: FFH

T1ADRL[7:0] T1 A Data Low Byte

#### NOTES:

Do not write "0000H" in the T1ADRH/T1ADRL register when PPG mode

#### T1BDRH (Timer 1 B Data High Register): BFH

7	6	5	4	3	2	1	0
T1BDRH7	T1BDRH6	T1BDRH5	T1BDRH4	T1BDRH3	T1BDRH2	T1BDRH1	T1BDRHO
R/W							

Initial value: FFH

T1BDRH[7:0] T1 B Data High Byte

#### T1BDRL (Timer 1 B Data Low Register): BEH

7	6	5	4	3	2	1	0
T1BDRL7	T1BDRL6	T1BDRL5	T1BDRL4	T1BDRL3	T1BDRL2	T1BDRL1	T1BDRL0
R/W							

Initial value: FFH

T1BDRL[7:0] T1 B Data Low Byte

**T1CRH (Timer 1 Control High Register): BBH**

7	6	5	4	3	2	1	0
T1EN	-	T1MS1	T1MS0	-	-	-	T1CC
R/W	-	R/W	R/W	-	-	-	R/W

Initial value: 00H

T1EN	Control Timer 1		
	0 Timer 1 disable		
	1 Timer 1 enable (Counter clear and start)		
T1MS[1:0]	Control Timer 1 Operation Mode		
	T1MS1	T1MS0	Description
	0	0	Timer/counter mode (T1O: toggle at A match)
	0	1	Capture mode (The A match interrupt can occur)
	1	0	PPG one-shot mode (PWM1O)
	1	1	PPG repeat mode (PWM1O)
T1CC	Clear Timer 1 Counter		
	0	No effect	
	1	Clear the Timer 1 counter (When write, automatically cleared "0" after being cleared counter)	

**T1CRL (Timer 1 Control Low Register): BAH**

7	6	5	4	3	2	1	0
T1CK2	T1CK1	T1CK0	T1IFR	-	T1POL	T1ECE	T1CNTR
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W

Initial value: 00H

T1CK[2:0] Select Timer 1 clock source. fx is main system clock frequency

T1CK2	T1CK1	T1CK0	Description
0	0	0	fx/2048
0	0	1	fx/512
0	1	0	fx/64
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 (EC1)

T1IFR When T1 Interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or auto clear by INT\_ACK signal. Writing "1" has no effect.

0	T1 Interrupt no generation
1	T1 Interrupt generation

T1POL T1O/PWM1O Polarity Selection

0	Start High (T1O/PWM1O is low level at disable)
1	Start Low (T1O/PWM1O is high level at disable)

T1ECE Timer 1 External Clock Edge Selection

0	External clock falling edge
1	External clock rising edge

T1CNTR Timer 1 Counter Read Control

0	No effect
1	Load the counter value to the B data register (When write, automatically cleared "0" after being loaded)

## 11.3 Timer 2

### 11.3.1 Overview

The 16-bit timer 2 consists of multiplexer, timer 2 A data high/low register, timer 2 B data high/low register and timer 2 control high/low register (T2ADRH, T2ADRL, T2BDRH, T2BDRL, T2CRH, and T2CRL).

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 2 can be divided clock of the system clock selected from prescaler output and T1 A Match (timer 1 A match signal). The clock source is selected by clock selection logic which is controlled by the clock selection bits (T2CK[2:0]).

TIMER 2 clock source: fx/1, fx/2, fx/4,fx/8,fx/32, fx/128, fx/512 and T1 A Match

In the capture mode, by EINT12, the data is captured into input capture data register (T2BDRH/T2BDRL). In timer/counter mode, whenever counter value is equal to T2ADRH/L, T2O port toggles. Also the timer 2 outputs PWM wave form to PWM2O port in the PPG mode.

**Table 19. Timer 2 Operating Modes**

T2EN	P1FSRL[5:4]	T2MS[1:0]	T2CK[2:0]	Timer 2
1	11	00	XXX	16-bit Timer/Counter Mode
1	00	01	XXX	16-bit Capture Mode
1	11	10	XXX	16-bit PPG Mode (one-shot mode)
1	11	11	XXX	16-bit PPG Mode (repeat mode)

### 11.3.2 16-bit Timer/Counter Mode

The 16-bit timer/counter mode is selected by control register as shown in Figure 45.

The 16-bit timer have counter and data register. The counter register is increased by internal or timer 1 A match clock input. Timer 2 can use the input clock with one of 1, 2, 4, 8, 32, 128, 512 and T1 A Match prescaler division rates (T2CK[2:0]). When the values of T2CNTH/T2CNTL and T2ADRH/T2ADRL are identical in timer 2, a match signal is generated and the interrupt of Timer 2 occurs. The T2CNTH/T2CNTL values are automatically cleared by match signal. It can be also cleared by software (T2CC).

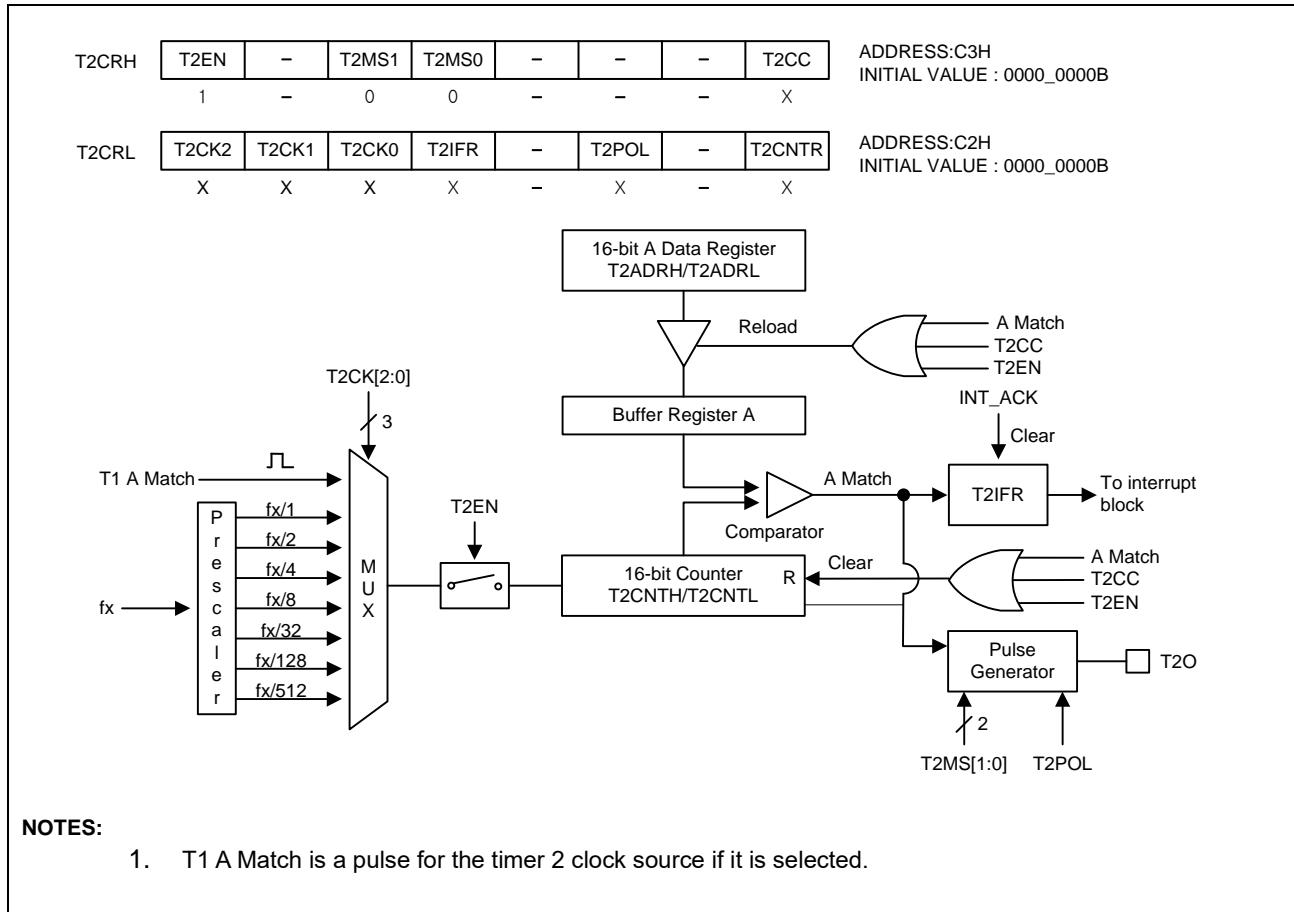


Figure 45. 16-bit Timer/Counter Mode for Timer 2

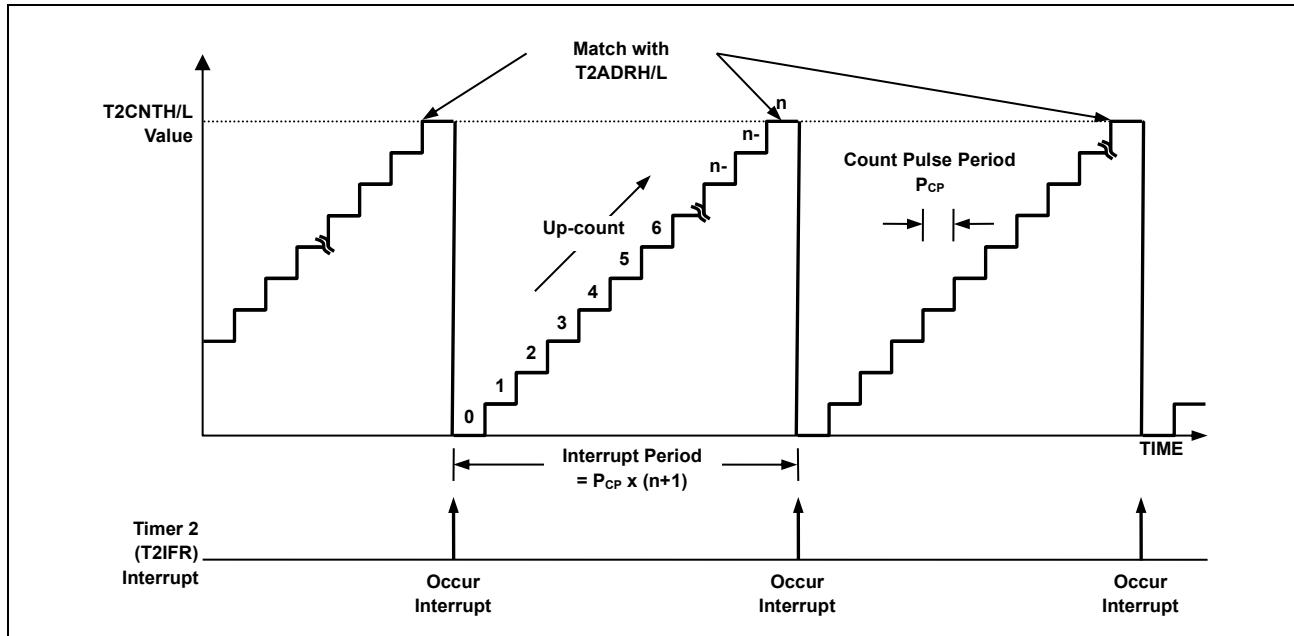


Figure 46. 16-bit Timer/Counter 2 Example

### 11.3.3 16-bit Capture Mode

The timer 2 capture mode is set by T2MS[1:0] as '01'. The clock source can use the internal clock. Basically, it has the same function as the 16-bit timer/counter mode and the interrupt occurs when T2CNTH/T2CNTL is equal to T2ADRH/T2ADRL. T2CNTH/T2CNTL values are automatically cleared by match signal and it can be also cleared by software (T2CC).

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 T2BDRH/T2BDRL. In the timer 2 capture mode, timer 2 output(T2O) waveform is not available.

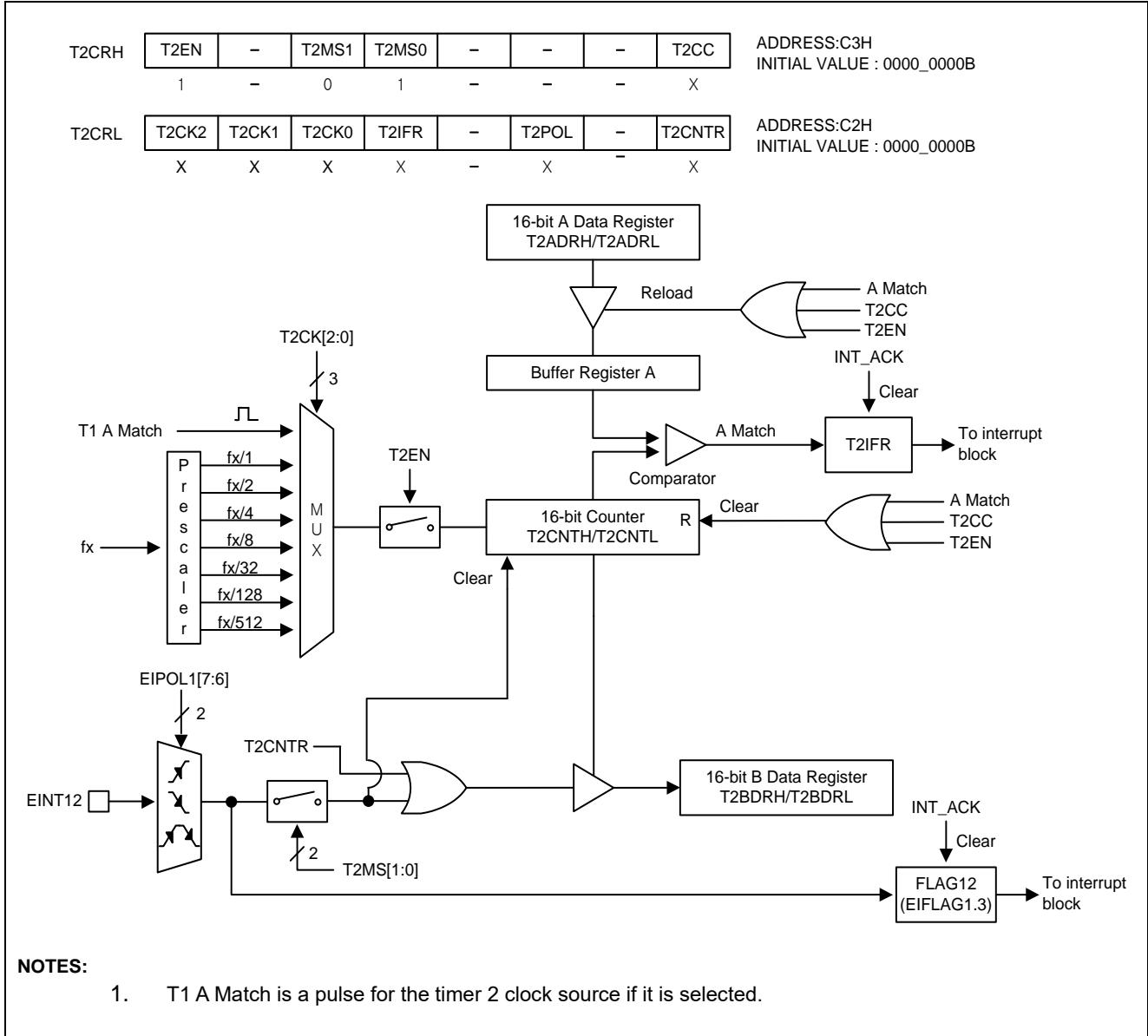


Figure 47. 16-bit Capture Mode for Timer 2

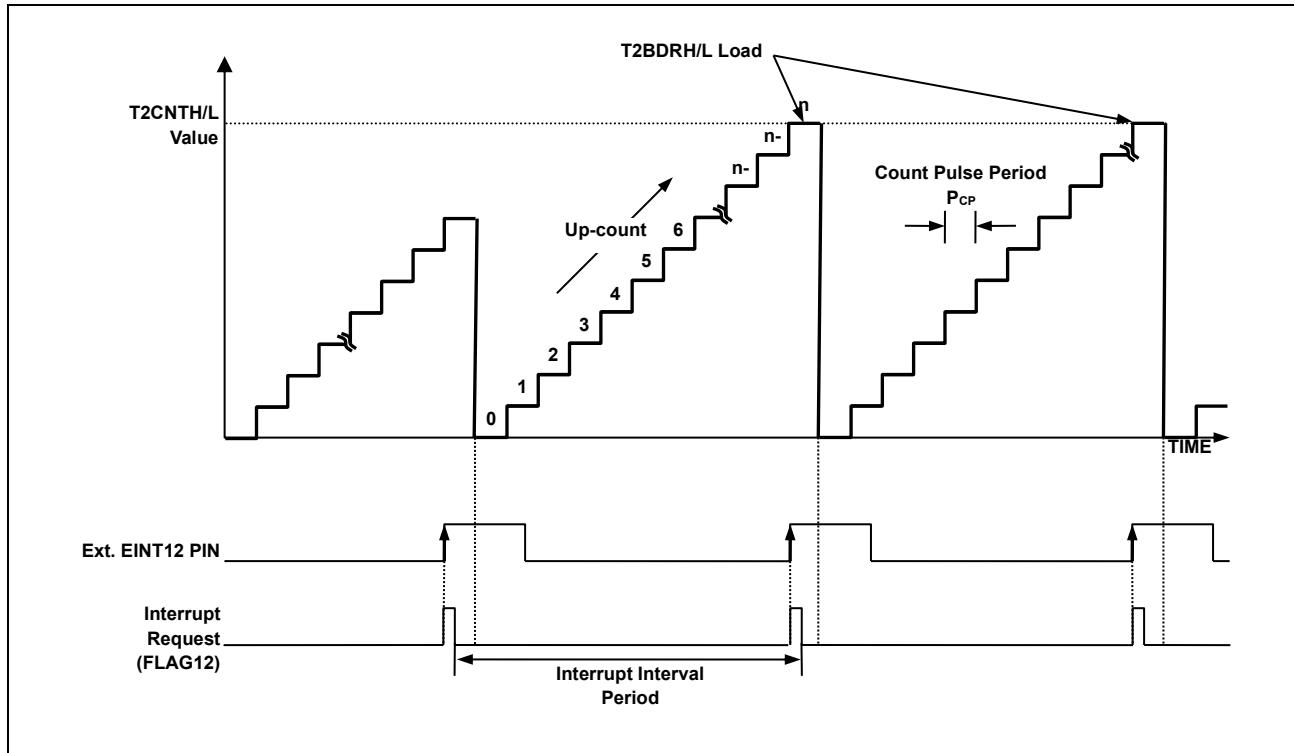


Figure 48. Input Capture Mode Operation for Timer 2

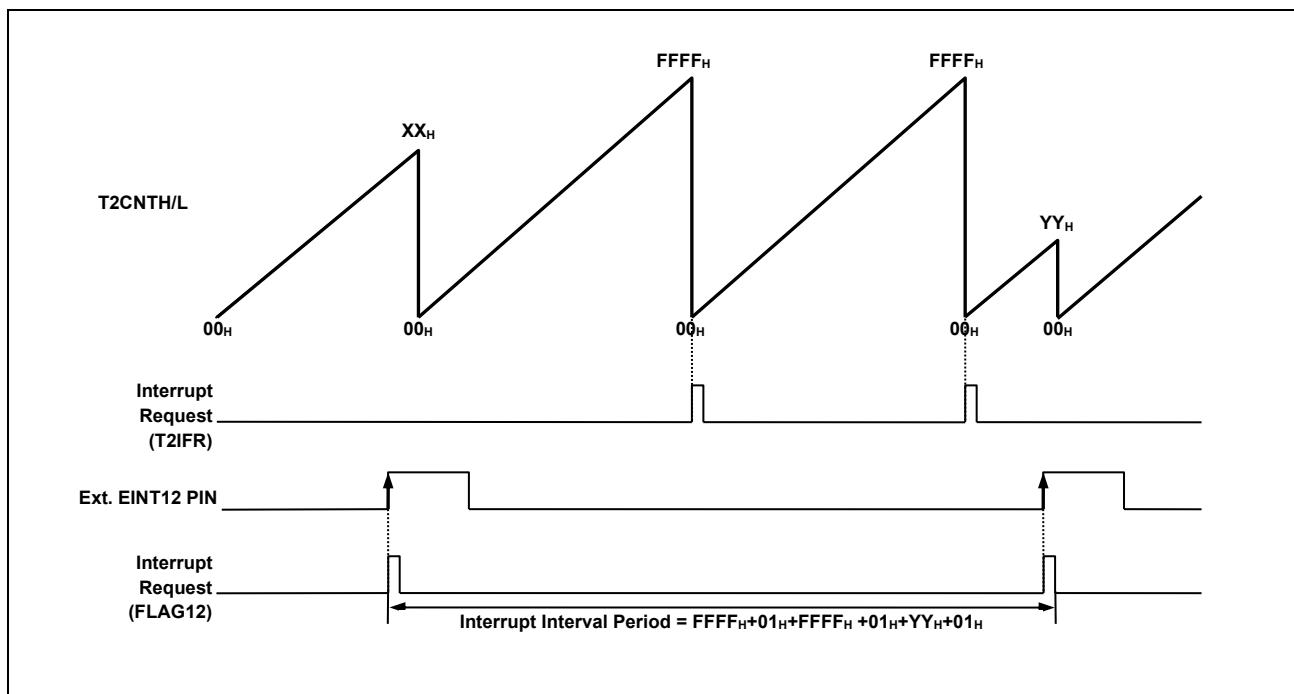


Figure 49. Express Timer Overflow in Capture Mode

### 11.3.4 16-bit PPG Mode

The timer 2 has a PPG (Programmable Pulse Generation) function. In PPG mode, the T2O/PWM2O pin outputs up to 16-bit resolution PWM output. This pin should be configured as a PWM output by set P1FSRL[5:4] to '11'. The period of the PWM output is determined by the T2ADRH/T2ADRL. And the duty of the PWM output is determined by the T2BDRH/T2BDRL.

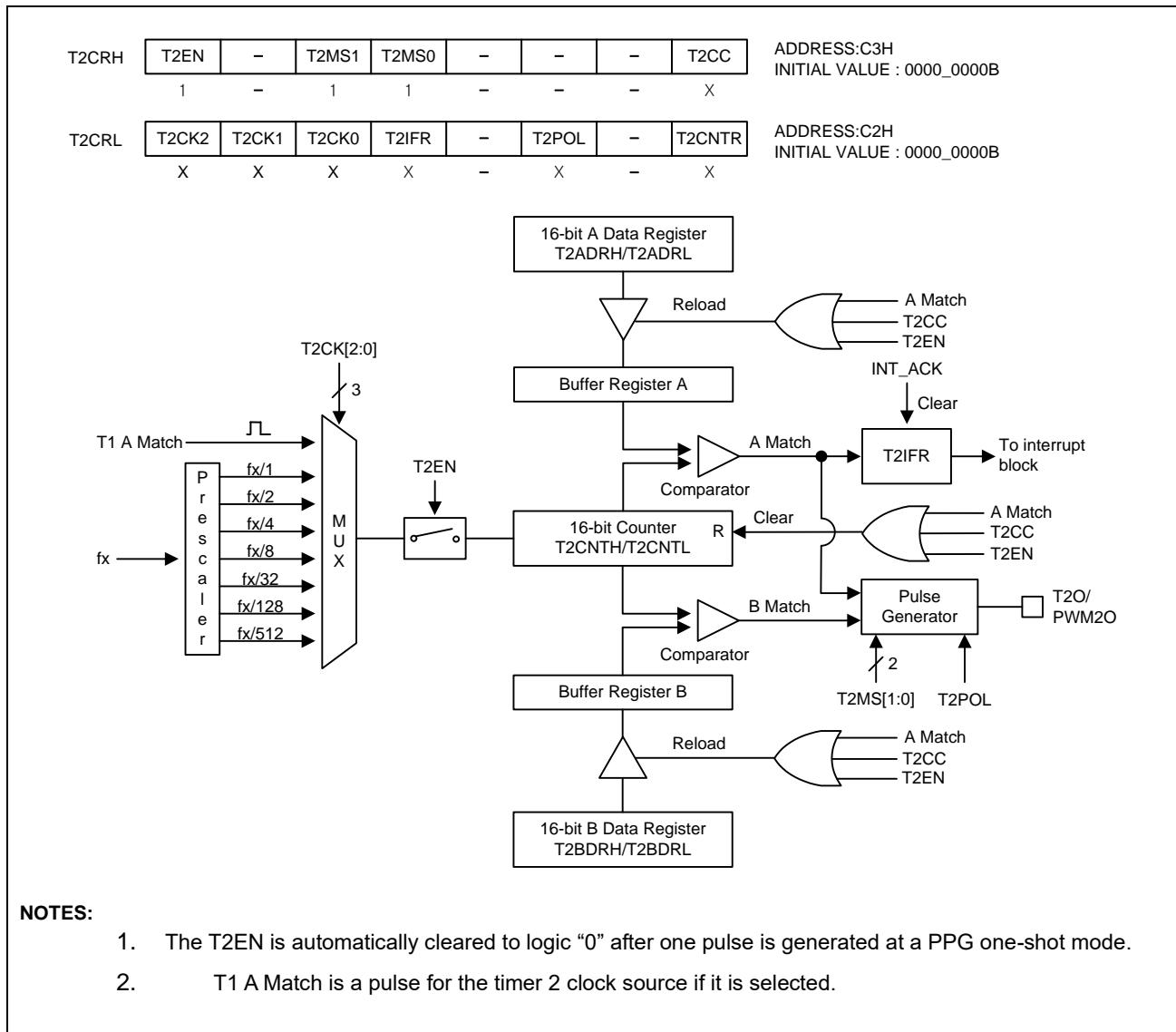


Figure 50. 16-bit PPG Mode for Timer 2

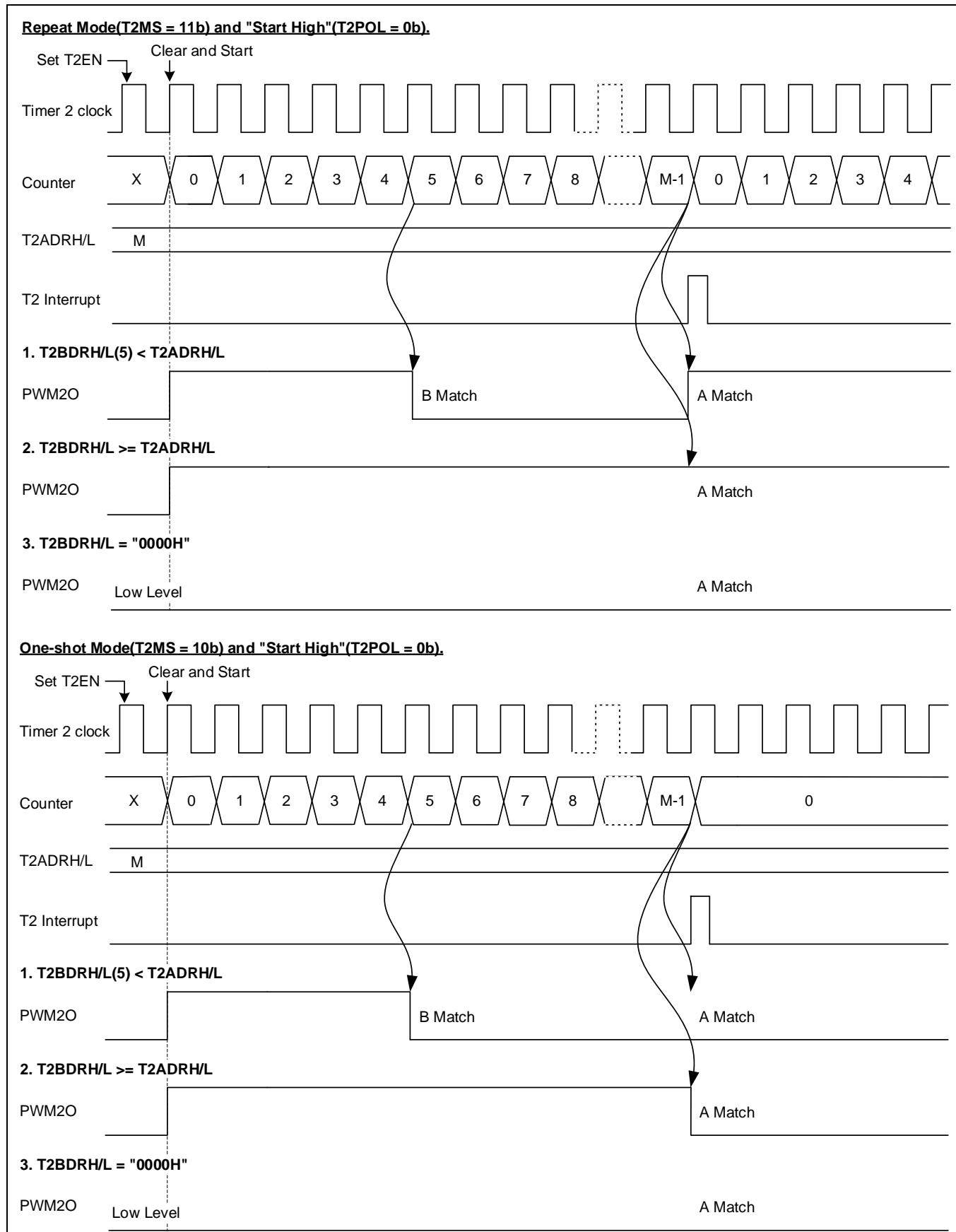


Figure 51. 16-bit PPG Mode Timming chart for Timer 2

### 11.3.5 Block Diagram

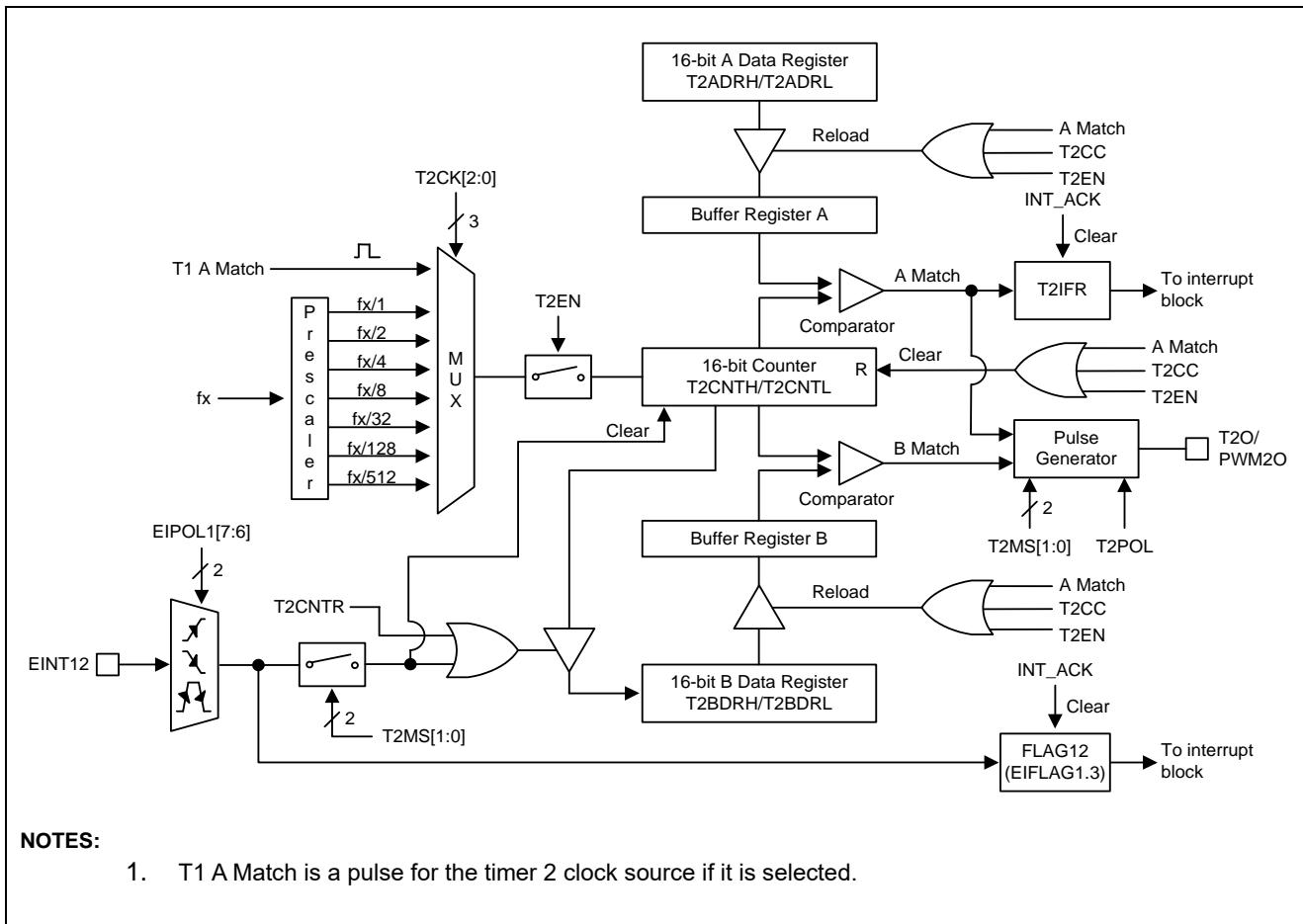


Figure 52. 16-bit Timer 2 Block Diagram

### 11.3.6 Register Map

Table 20. Timer 2 Register Map

Name	Address	Direction	Default	Description
T2ADRH	C5H	R/W	FFH	Timer 2 A Data High Register
T2ADRL	C4H	R/W	FFH	Timer 2 A Data Low Register
T2BDRH	C7H	R/W	FFH	Timer 2 B Data High Register
T2BDRL	C6H	R/W	FFH	Timer 2 B Data Low Register
T2CRH	C3H	R/W	00H	Timer 2 Control High Register
T2CRL	C2H	R/W	00H	Timer 2 Control Low Register

### 11.3.7 Timer/Counter 2 Register Description

The timer/counter 2 register consists of timer 2 A data high register (T2ADR<sub>H</sub>), timer 2 A data low register (T2ADR<sub>L</sub>), timer 2 B data high register (T2BDR<sub>H</sub>), timer 2 B data low register (T2BDR<sub>L</sub>), timer 2 control high register (T2CR<sub>H</sub>) and timer 2 control low register (T2CR<sub>L</sub>).

### 11.3.8 Register Description for Timer/Counter 2

#### T2ADR<sub>H</sub> (Timer 2 A Data High Register): C5H

7	6	5	4	3	2	1	0
T2ADR <sub>H</sub> 7	T2ADR <sub>H</sub> 6	T2ADR <sub>H</sub> 5	T2ADR <sub>H</sub> 4	T2ADR <sub>H</sub> 3	T2ADR <sub>H</sub> 2	T2ADR <sub>H</sub> 1	T2ADR <sub>H</sub> 0
R/W							

Initial value: FFH

T2ADR<sub>H</sub>[7:0]      T2 A Data High Byte

#### T2ADR<sub>L</sub> (Timer 2 A Data Low Register): C4H

7	6	5	4	3	2	1	0
T2ADR <sub>L</sub> 7	T2ADR <sub>L</sub> 6	T2ADR <sub>L</sub> 5	T2ADR <sub>L</sub> 4	T2ADR <sub>L</sub> 3	T2ADR <sub>L</sub> 2	T2ADR <sub>L</sub> 1	T2ADR <sub>L</sub> 0
R/W							

Initial value: FFH

T2ADR<sub>L</sub>[7:0]      T2 A Data Low Byte

**NOTES:**

Do not write “0000H” in the T2ADR<sub>H</sub>/T2ADR<sub>L</sub> register when PPG mode.

#### T2BDR<sub>H</sub> (Timer 2 B Data High Register): C7H

7	6	5	4	3	2	1	0
T2BDR <sub>H</sub> 7	T2BDR <sub>H</sub> 6	T2BDR <sub>H</sub> 5	T2BDR <sub>H</sub> 4	T2BDR <sub>H</sub> 3	T2BDR <sub>H</sub> 2	T2BDR <sub>H</sub> 1	T2BDR <sub>H</sub> 0
R/W							

Initial value: FFH

T2BDR<sub>H</sub>[7:0]      T2 B Data High Byte

#### T2BDR<sub>L</sub> (Timer 2 B Data Low Register): C6H

7	6	5	4	3	2	1	0
T2BDR <sub>L</sub> 7	T2BDR <sub>L</sub> 6	T2BDR <sub>L</sub> 5	T2BDR <sub>L</sub> 4	T2BDR <sub>L</sub> 3	T2BDR <sub>L</sub> 2	T2BDR <sub>L</sub> 1	T2BDR <sub>L</sub> 0
R/W							

Initial value: FFH

T2BDR<sub>L</sub>[7:0]      T2 B Data Low Byte

**T2CRH (Timer 2 Control High Register): C3H**

7	6	5	4	3	2	1	0
T2EN	–	T2MS1	T2MS0	–	–	–	T2CC
R/W	–	R/W	R/W	–	–	–	R/W

Initial value: 00H

T2EN	Control Timer 2		
	0 Timer 2 disable		
	1 Timer 2 enable (Counter clear and start)		
T2MS[1:0]	Control Timer 2 Operation Mode		
	T2MS1	T2MS0	Description
	0	0	Timer/counter mode (T2O: toggle at A match)
	0	1	Capture mode (The A match interrupt can occur)
	1	0	PPG one-shot mode (PWM2O)
	1	1	PPG repeat mode (PWM2O)
T2CC	Clear Timer 2 Counter		
	0	No effect	
	1	Clear the Timer 2 counter (When write, automatically cleared "0" after being cleared counter)	

**T2CRL (Timer 2 Control Low Register): C2H**

7	6	5	4	3	2	1	0
T2CK2	T2CK1	T2CK0	T2IFR	-	T2POL	-	T2CNTR
R/W	R/W	R/W	R/W	-	R/W	-	R/W

Initial value: 00H

T2CK[2:0] Select Timer 2 clock source. fx is main system clock frequency

T2CK2	T2CK1	T2CK0	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	T1 A Match

T2IFR When T2 Match Interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or auto clear by INT\_ACK signal. Writing "1" has no effect.

0	T2 interrupt no generation
1	T2 interrupt generation

T2POL T2O/PWM2O Polarity Selection

0	Start High (T2O/PWM2O is low level at disable)
1	Start Low (T2O/PWM2O is high level at disable)

T2CNTR Timer 2 Counter Read Control

0	No effect
1	Load the counter value to the B data register (When write, automatically cleared "0" after being loaded)

## 11.4 Timer 3

### 11.4.1 Overview

The 16-bit timer 3 consists of multiplexer, timer 3 A data register high/low, timer 3 B data register high/low and timer 3 control register high/low (T3ADRH, T3ADRL, T3BDRH, T3BDRL, T3CRH, and T3CRL).

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 can be clocked by an internal or an external clock source (EC3). The clock source is selected by clock selection logic which is controlled by the clock selection bits (T3CK[2:0]).

TIMER 3 clock source: fx/1, 2, 4, 8, 64, 512, 2048 and EC3

In the capture mode, by EINT3, the data is captured into input capture data register (T3BDRH/T3BDRL). Timer 3 outputs the comparison result between counter and data register through T3O port in timer/counter mode. Also Timer 3 outputs PWM wave form through PWM3O port in the PPG mode.

**Table 21. Timer 3 Operating Modes**

T3EN	P0FSRL[3:2]	T3MS[1:0]	T3CK[2:0]	Timer 3
1	11	00	XXX	16-bit Timer/Counter Mode
1	00	01	XXX	16-bit Capture Mode
1	11	10	XXX	16-bit PPG Mode(one-shot mode)
1	11	11	XXX	16-bit PPG Mode(repeat mode)

### 11.4.2 16-bit Timer/Counter Mode

The 16-bit timer/counter mode is selected by control register as shown in Figure 53.

The 16-bit timer have counter and data register. The counter register is increased by internal or external clock input. Timer 3 can use the input clock with one of 1, 2, 4, 8, 64, 512 and 2048 prescaler division rates (T3CK[2:0]). When the value of T3CNTH, T3CNTL and the value of T3ADRH, T3ADRL are identical in Timer 3 respectively, a match signal is generated and the interrupt of Timer 3 occurs. The T3CNTH, T3CNTL value is automatically cleared by match signal.

It can be also cleared by software (T3CC).

The external clock (EC3) counts up the timer at the rising edge. If the EC3 is selected as a clock source by T3CK[2:0], EC3 port should be set to the input port by P00IO bit.

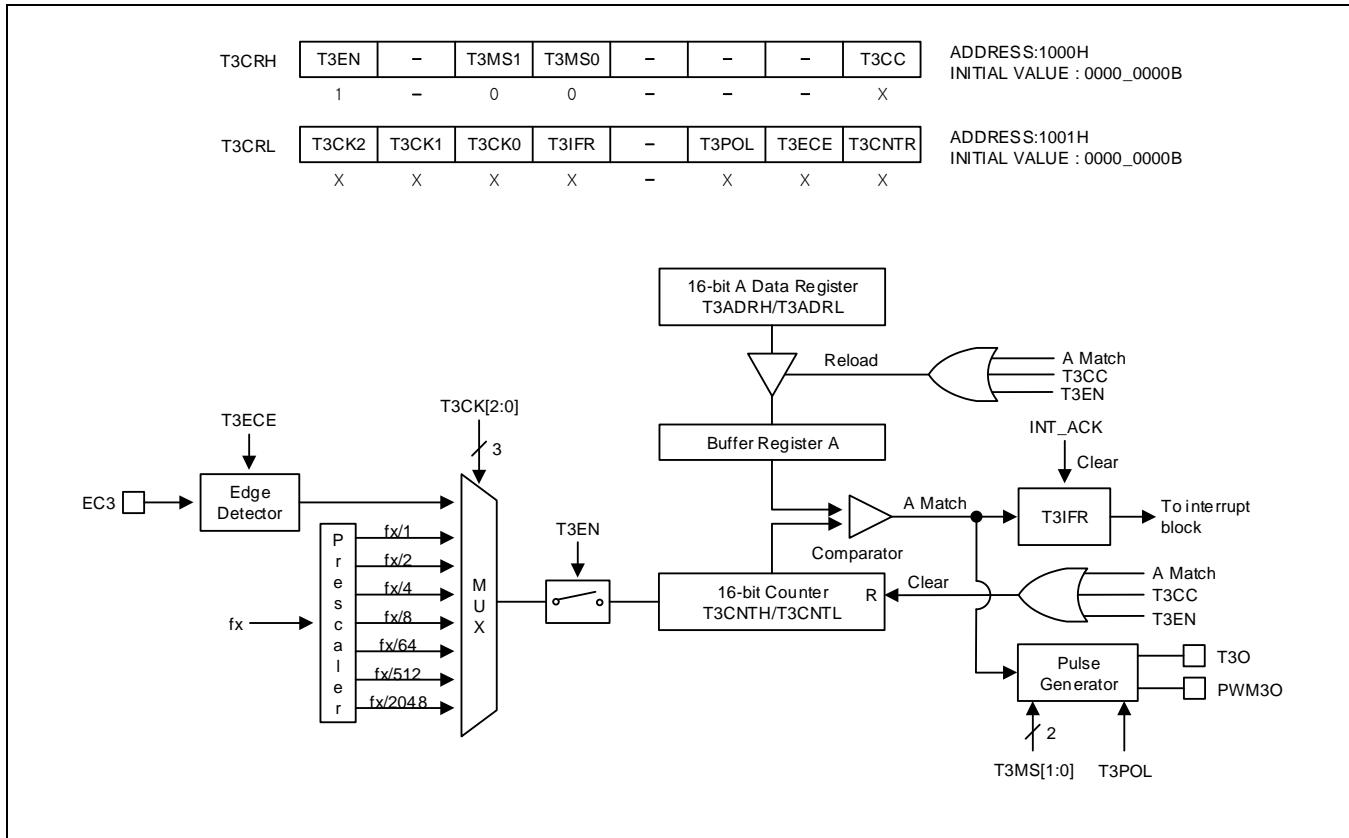


Figure 53. 16-bit Timer/Counter Mode for Timer 3

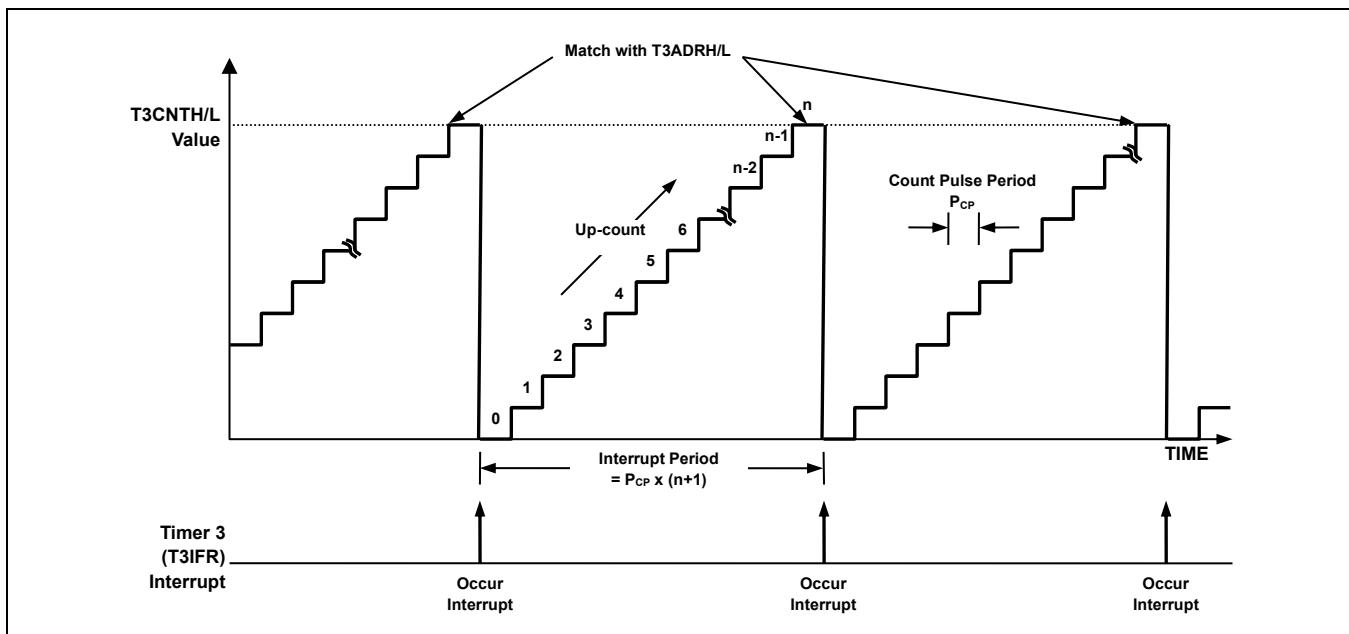


Figure 54. 16-bit Timer/Counter 3 Example

### 11.4.3 16-bit Capture Mode

The 16-bit timer 3 capture mode is set by T3MS[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 T3CNTH/T3CNTL is equal to T3ADRH/T3ADRL. The T3CNTH, T3CNTL values are automatically cleared by match signal. It can be also cleared by software (T3CC).

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 T3BDRH/T3BDRL.

According to EIPOL0 registers setting, the external interrupt EINT3 function is chosen. Of course, the EINT3 pin must be set as an input port.

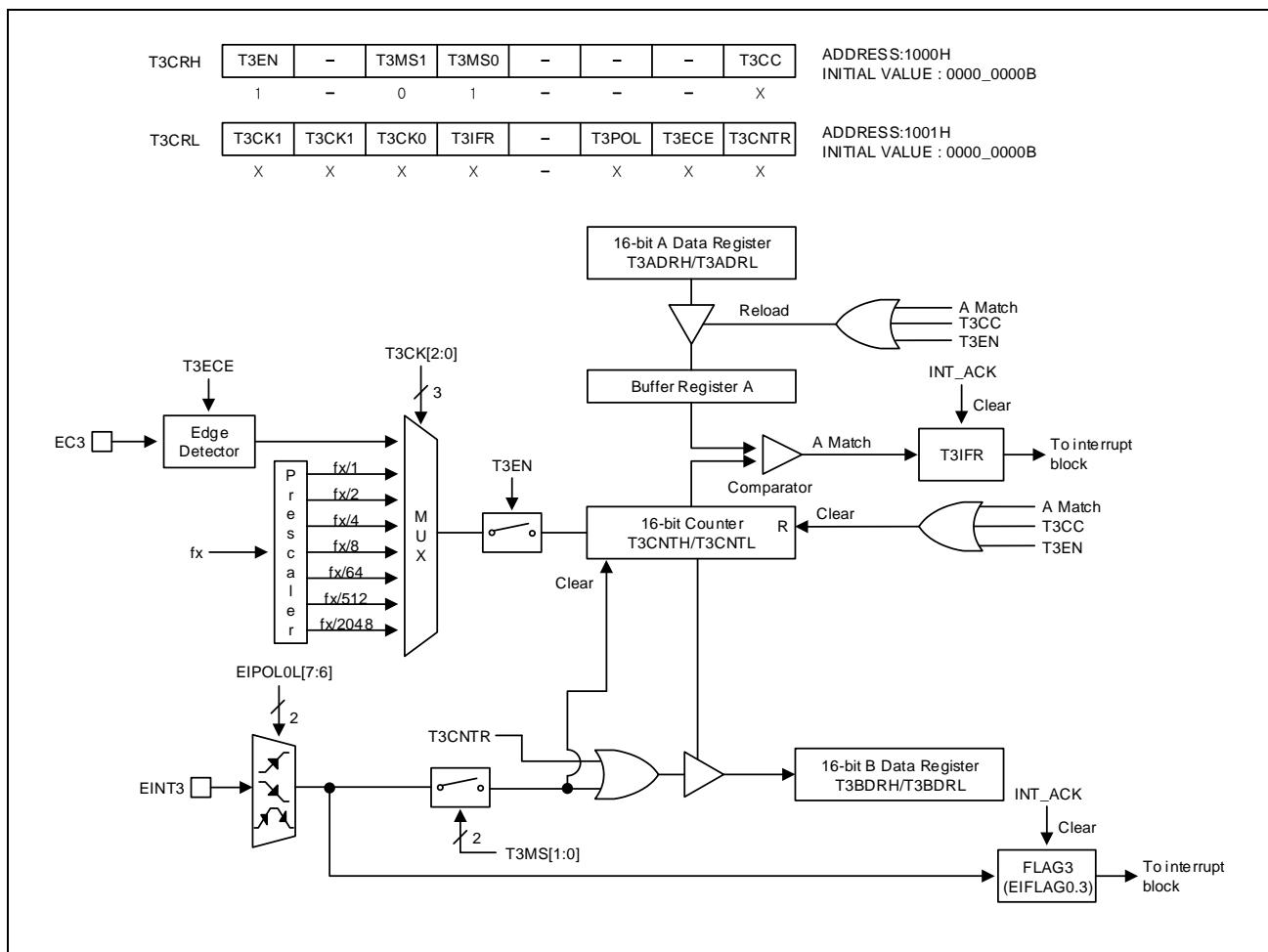


Figure 55. 16-bit Capture Mode for Timer 3

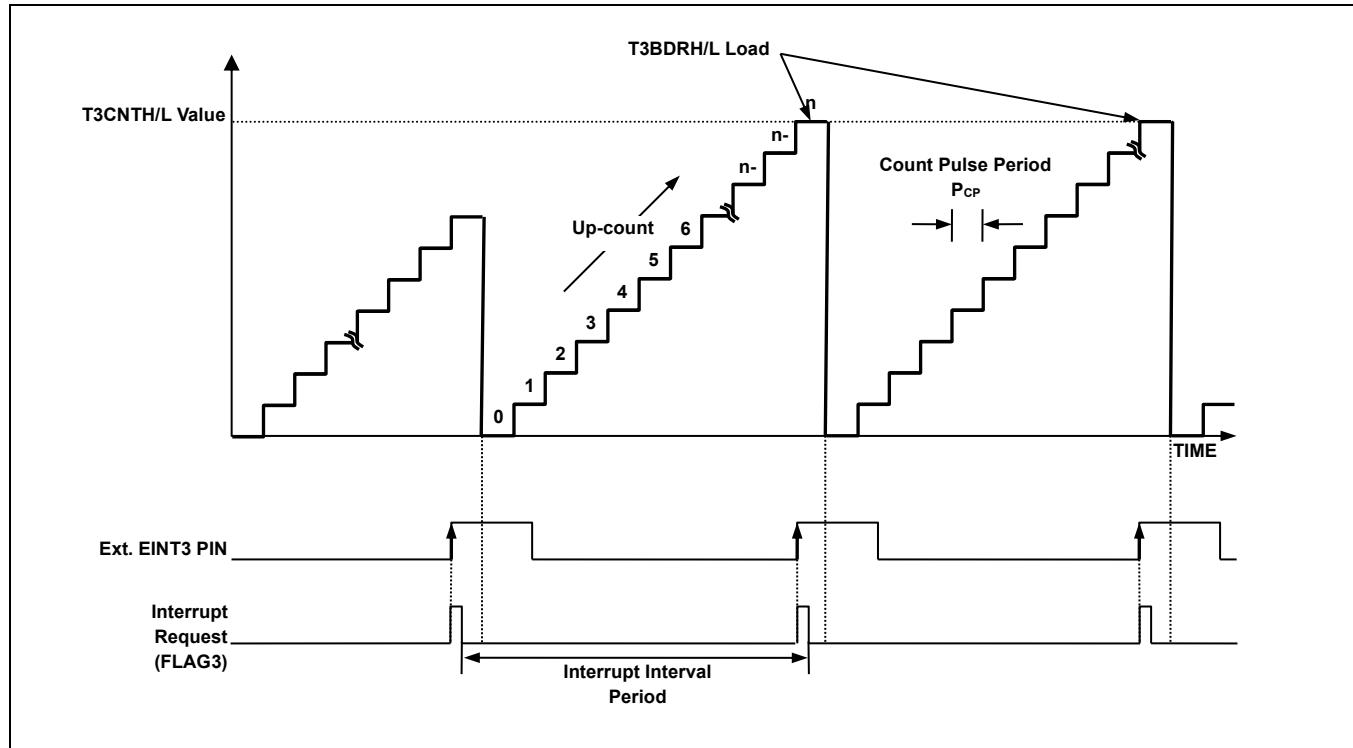


Figure 56. Input Capture Mode Operation for Timer 3

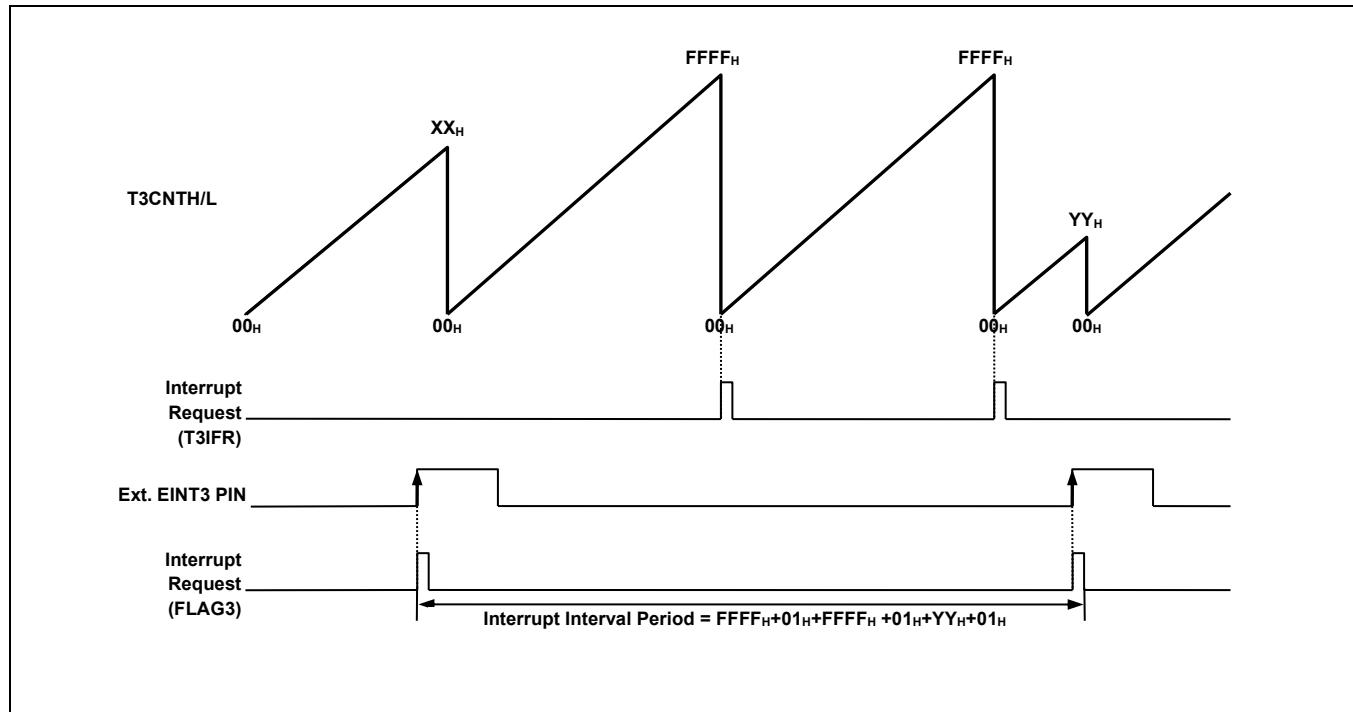


Figure 57. Express Timer Overflow in Capture Mode

### 11.4.4 16-bit PPG Mode

The timer 3 has a PPG (Programmable Pulse Generation) function. In PPG mode, T3O/PWM3O pin outputs up to 16-bit resolution PWM output. This pin should be configured as a PWM output by setting P0FSRH[3:2] to '11'. The period of the PWM output is determined by the T3ADRH/T3ADRL. And the duty of the PWM output is determined by the T3BDRH/T3BDRL.

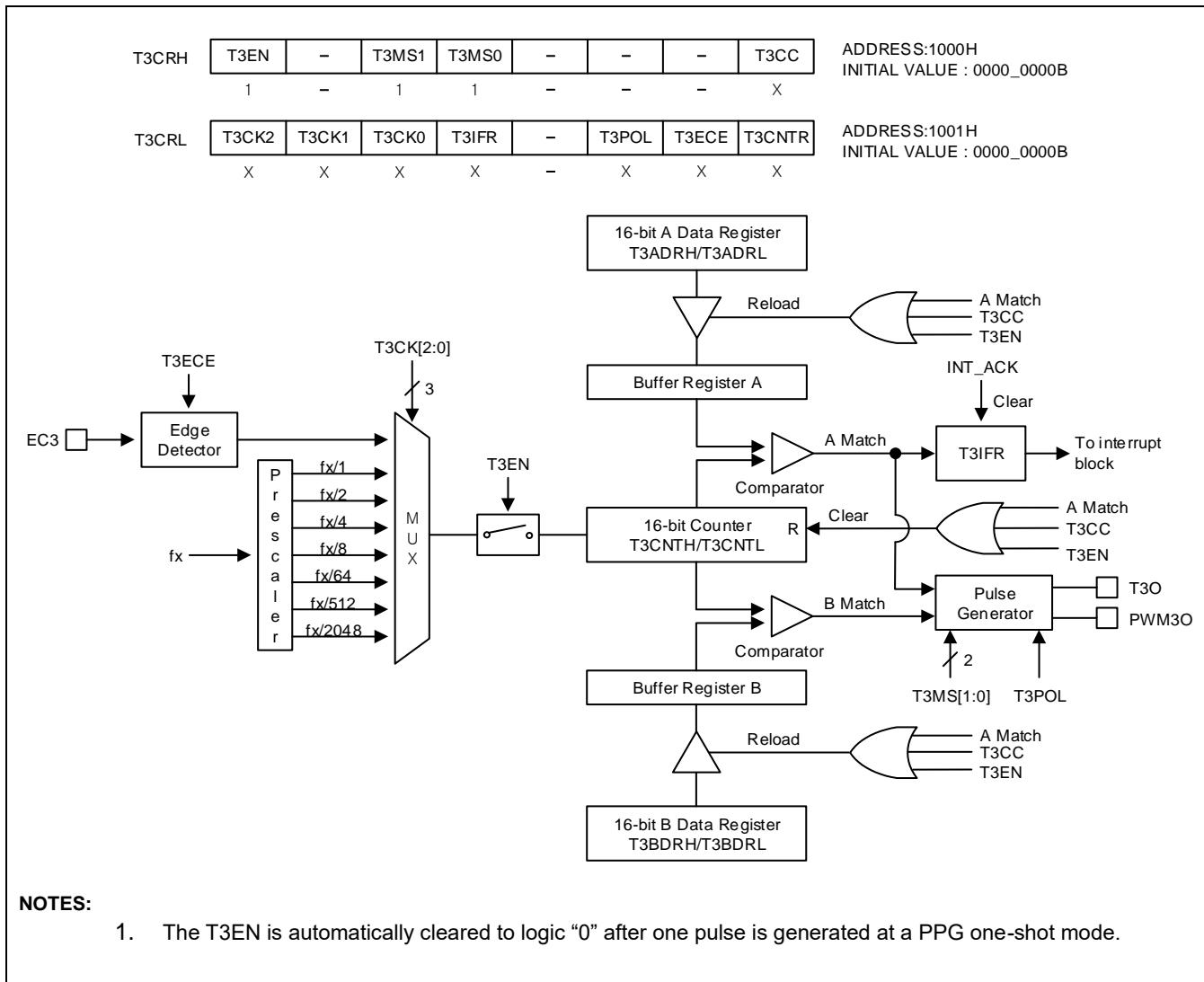


Figure 58. 16-bit PPG Mode for Timer 3

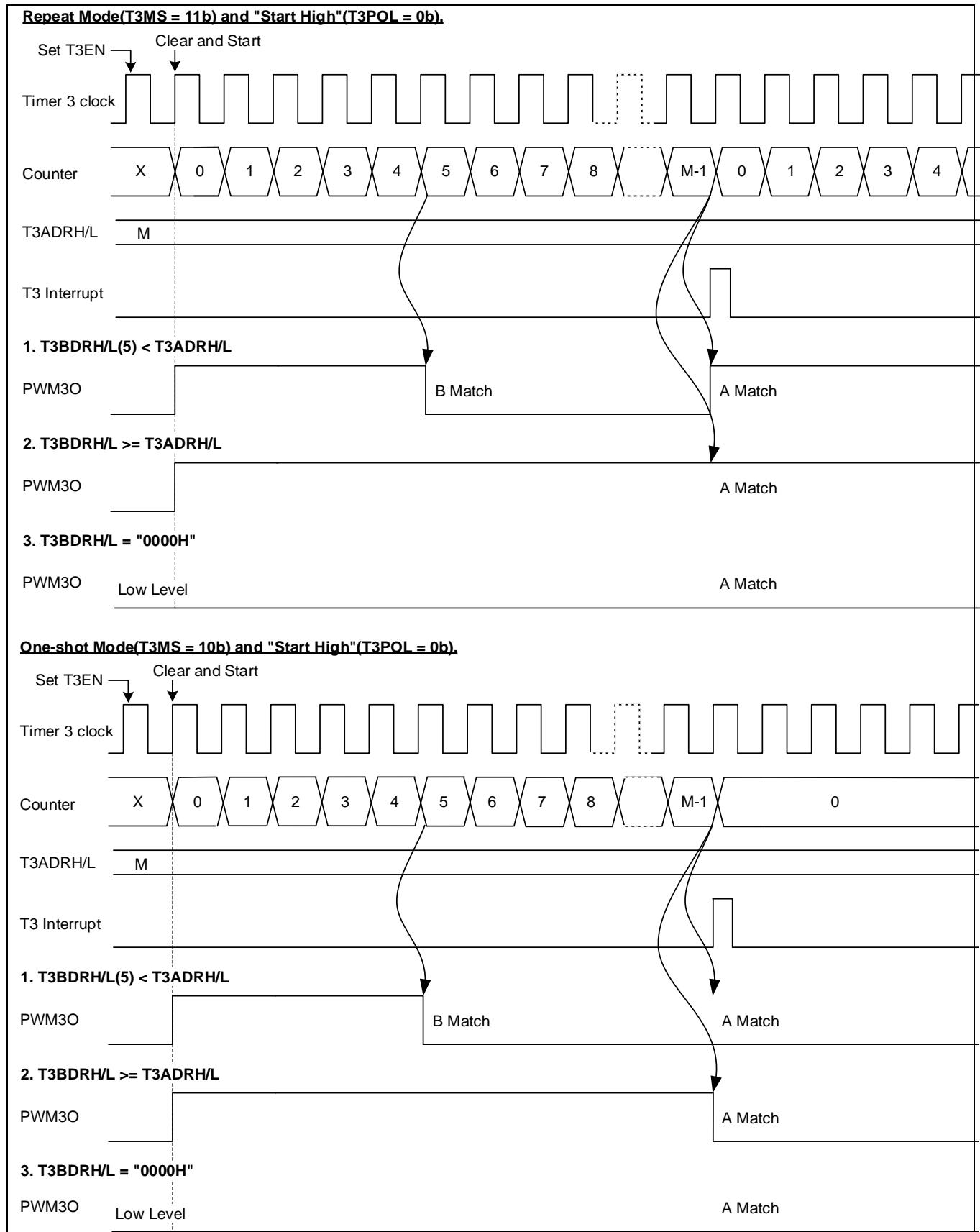


Figure 59. 16-bit PPG Mode Timming chart for Timer 3

### 11.4.5 Block Diagram

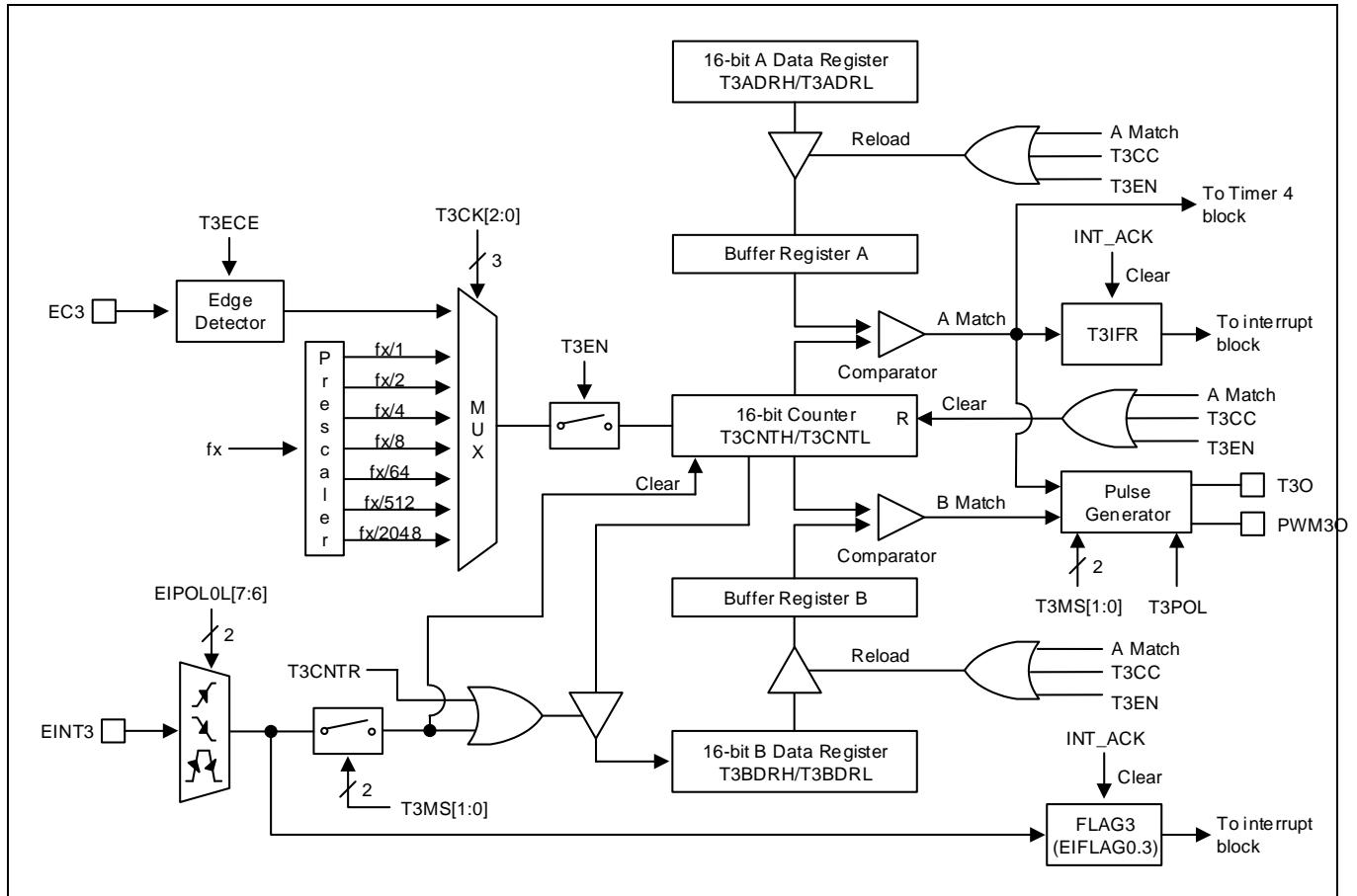


Figure 60. 16-Bit Timer 3 Block Diagram

### 11.4.6 Register Map

Table 22. Timer 3 Register Map

Name	Address	Direction	Default	Description
T3ADRH	1002H	R/W	FFH	Timer 3 A Data High Register
T3ADRL	1003H	R/W	FFH	Timer 3 A Data Low Register
T3BDRH	1004H	R/W	FFH	Timer 3 B Data High Register
T3BDRL	1005H	R/W	FFH	Timer 3 B Data Low Register
T3CRH	1000H	R/W	00H	Timer 3 Control High Register
T3CRL	1001H	R/W	00H	Timer 3 Control Low Register

### 11.4.7 Timer/Counter 3 Register Description

The timer/counter 3 register consists of timer 3 A data high register (T3ADRH), timer 3 A data low register (T3ADRL), timer 3 B data high register (T3BDRH), timer 3 B data low register (T3BDRL), timer 3 control high register (T3CRH) and timer 3 control low register (T3CRL).

### 11.4.8 Register Description for Timer/Counter 3

#### T3ADRH (Timer 3 A Data High Register): 1002H

7	6	5	4	3	2	1	0
T3ADR7	T3ADR6	T3ADR5	T3ADR4	T3ADR3	T3ADR2	T3ADR1	T3ADR0
R/W							

Initial value: FFH

T3ADR7[7:0]      T3 A Data High Byte

#### T3ADRL (Timer 3 A Data Low Register): 1003H

7	6	5	4	3	2	1	0
T3ADRL7	T3ADRL6	T3ADRL5	T3ADRL4	T3ADRL3	T3ADRL2	T3ADRL1	T3ADRL0
R/W							

Initial value: FFH

T3ADRL7[7:0]      T3 A Data Low Byte

**NOTES:**

Do not write “0000H” in the T3ADR7/T3ADRL register when PPG mode

#### T3BDRH (Timer 3 B Data High Register): 1004H

7	6	5	4	3	2	1	0
T3BDRH7	T3BDRH6	T3BDRH5	T3BDRH4	T3BDRH3	T3BDRH2	T3BDRH1	T3BDRH0
R/W							

Initial value: FFH

T3BDRH7[7:0]      T3 B Data High Byte

#### T3BDRL (Timer 3 B Data Low Register): 1005H

7	6	5	4	3	2	1	0
T3BDRL7	T3BDRL6	T3BDRL5	T3BDRL4	T3BDRL3	T3BDRL2	T3BDRL1	T3BDRL0
R/W							

Initial value: FFH

T3BDRL7[7:0]      T3 B Data Low Byte

**T3CRH (Timer 3 Control High Register): 1000H**

7	6	5	4	3	2	1	0
T3EN	-	T3MS1	T3MS0	-	-	-	T3CC
RW	-	R/W	R/W	-	-	-	RW

Initial value: 00H

T3EN	Control Timer 3		
	0 Timer 3 disable		
	1 Timer 3 enable (Counter clear and start)		
T3MS[1:0]	Control Timer 3 Operation Mode		
	T3MS1	T3MS0	Description
	0	0	Timer/counter mode (T3O: toggle at A match)
	0	1	Capture mode (The A match interrupt can occur)
	1	0	PPG one-shot mode (PWM3O)
	1	1	PPG repeat mode (PWM3O)
T3CC	Clear Timer 3 Counter		
	0	No effect	
	1	Clear the Timer 3 counter (When write, automatically cleared "0" after being cleared counter)	

**T3CRL (Timer 3 Control Low Register): 1001H**

7	6	5	4	3	2	1	0
T3CK2	T3CK1	T3CK0	T3IFR	-	T3POL	T3ECE	T3CNTR
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W

Initial value: 00H

T3CK[2:0]	Select Timer 3 clock source. fx is main system clock frequency		
	T3CK2 T3CK T3CK Description		
	1	0	
	0	0	fx/2048
	0	1	fx/512
	0	1	fx/64
	0	1	fx/8
	1	0	fx/4
	1	0	fx/2
	1	1	fx/1
	1	1	External clock (EC3)
T3IFR	When T3 Interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or auto clear by INT_ACK signal. Writing "1" has no effect.		
	0	T3 Interrupt no generation	
	1	T3 Interrupt generation	
T3POL	T3O/PWM3O Polarity Selection		
	0	Start High (T3O/PWM3O is low level at disable)	
	1	Start Low (T3O/PWM3O is high level at disable)	
T3ECE	Timer 3 External Clock Edge Selection		
	0	External clock falling edge	
	1	External clock rising edge	
T3CNTR	Timer 3 Counter Read Control		

0	No effect
1	Load the counter value to the B data register (When write, automatically cleared "0" after being loaded)

## 11.5 Timer 4

### 11.5.1 Overview

The 16-bit TIMER 4 consists of multiplexer, timer 4 A data high/low register, timer 4 B data high/low register and timer 4 control high/low register (T4ADRH, T4ADRL, T4BDRH, T4BDRL, T4CRH, and T4CRL).

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 4 can be divided clock of the system clock selected from prescaler output and T3 A Match (timer 3 A match signal). The clock source is selected by clock selection logic which is controlled by the clock selection bits (T4CK[2:0]).

TIMER 4 clock source: fx/1, fx/2, fx/4, fx/8, fx/32, fx/128, fx/512 and T3 A Match

In the capture mode, by EINT4, the data is captured into input capture data register (T4BDRH/T4BDRL). In timer/counter mode, whenever counter value is equal to T4ADRH/L, T4O port toggles. Also the TIMER 4 outputs PWM wave form to PWM4O port in the PPG mode.

**Table 23. TIMER 4 Operating Modes**

T4EN	P0FSRH[5:4]	T4MS[1:0]	T4CK[2:0]	TIMER 4
1	11	00	XXX	16-bit Timer/Counter Mode
1	00	01	XXX	16-bit Capture Mode
1	11	10	XXX	16-bit PPG Mode (one-shot mode)
1	11	11	XXX	16-bit PPG Mode (repeat mode)

### 11.5.2 16-bit Timer/Counter Mode

The 16-bit timer/counter mode is selected by control register as shown in Figure 61.

The 16-bit timer have counter and data register. The counter register is increased by internal or timer 3 A match clock input. Timer 4 can use the input clock with one of 1, 2, 4, 8, 32, 128, 512 and T3 A Match prescaler division rates (T4CK[2:0]). When the values of T4CNTH/T4CNTL and T4ADRH/T4ADRL are identical in timer 4, a match signal is generated and the interrupt of Timer 4 occurs. The T4CNTH/T4CNTL values are automatically cleared by match signal. It can be also cleared by software (T4CC).

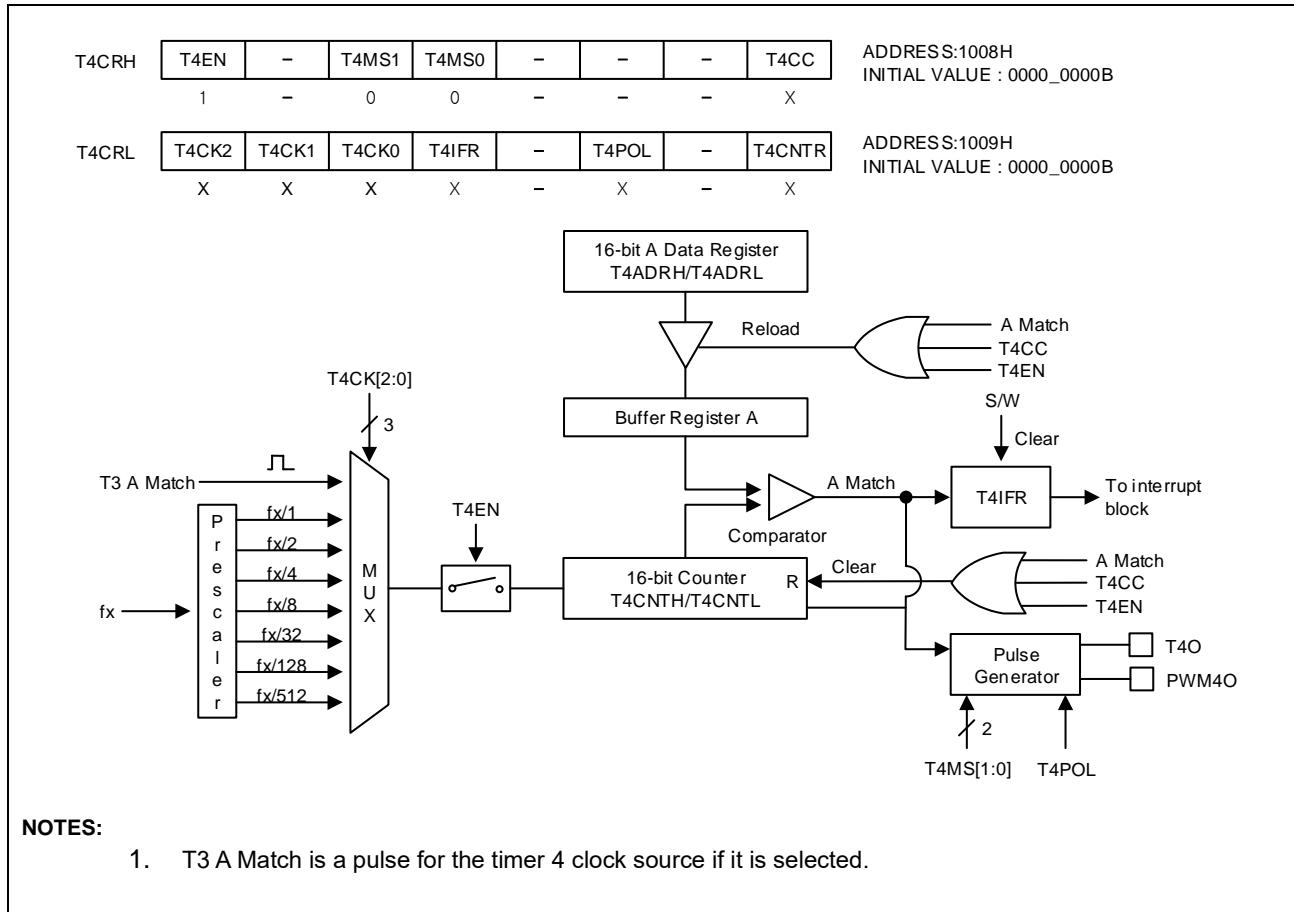


Figure 61. 16-bit Timer/Counter Mode for Timer 4

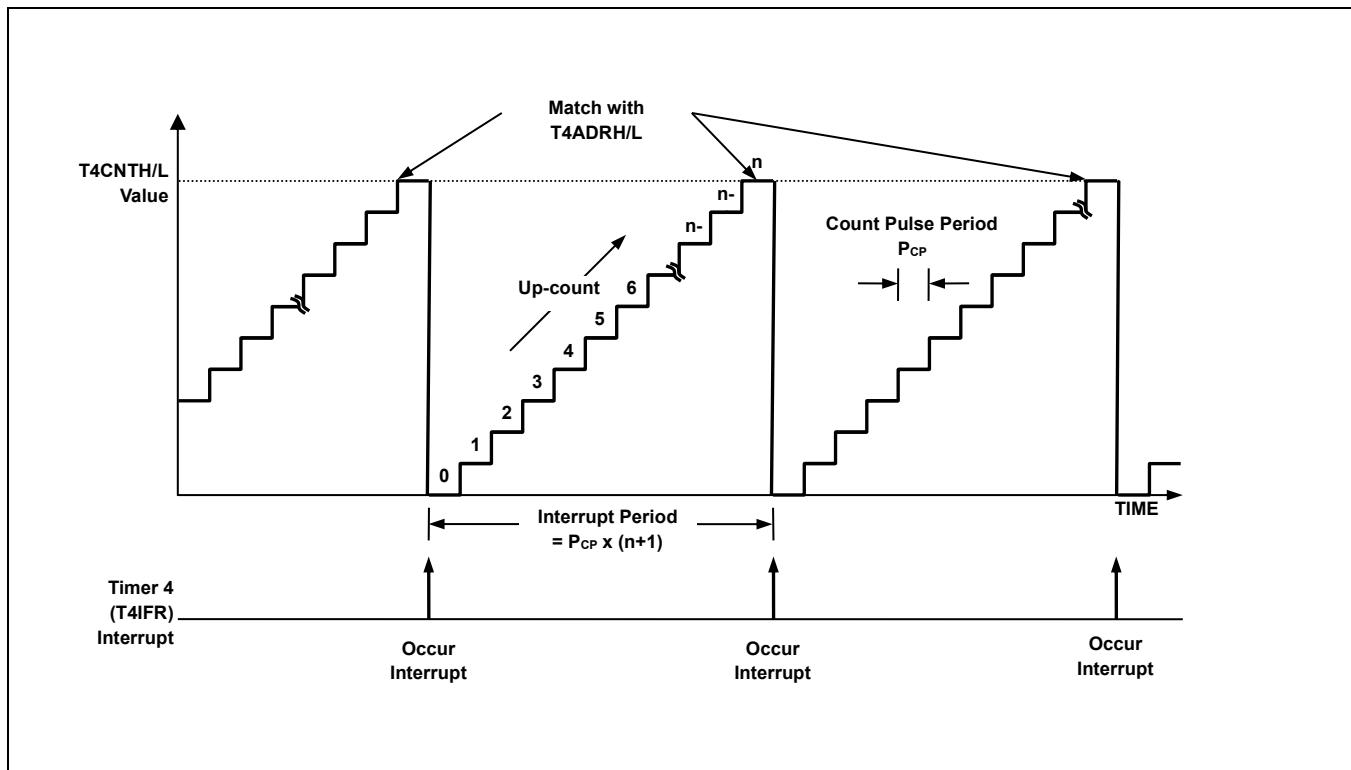


Figure 62. 16-bit Timer/Counter 4 Example

### 11.5.3 16-bit Capture Mode

The timer 4 capture mode is set by T4MS[1:0] as '01'. The clock source can use the internal clock. Basically, it has the same function as the 16-bit timer/counter mode and the interrupt occurs when T4CNTH/T4CNTL is equal to T4ADRH/T4ADRL. T4CNTH/T4CNTL values are automatically cleared by match signal and it can be also cleared by software (T4CC).

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 T4BDRH/T4BDRL. In the timer 4 capture mode, timer 4 output (T4O) waveform is not available.

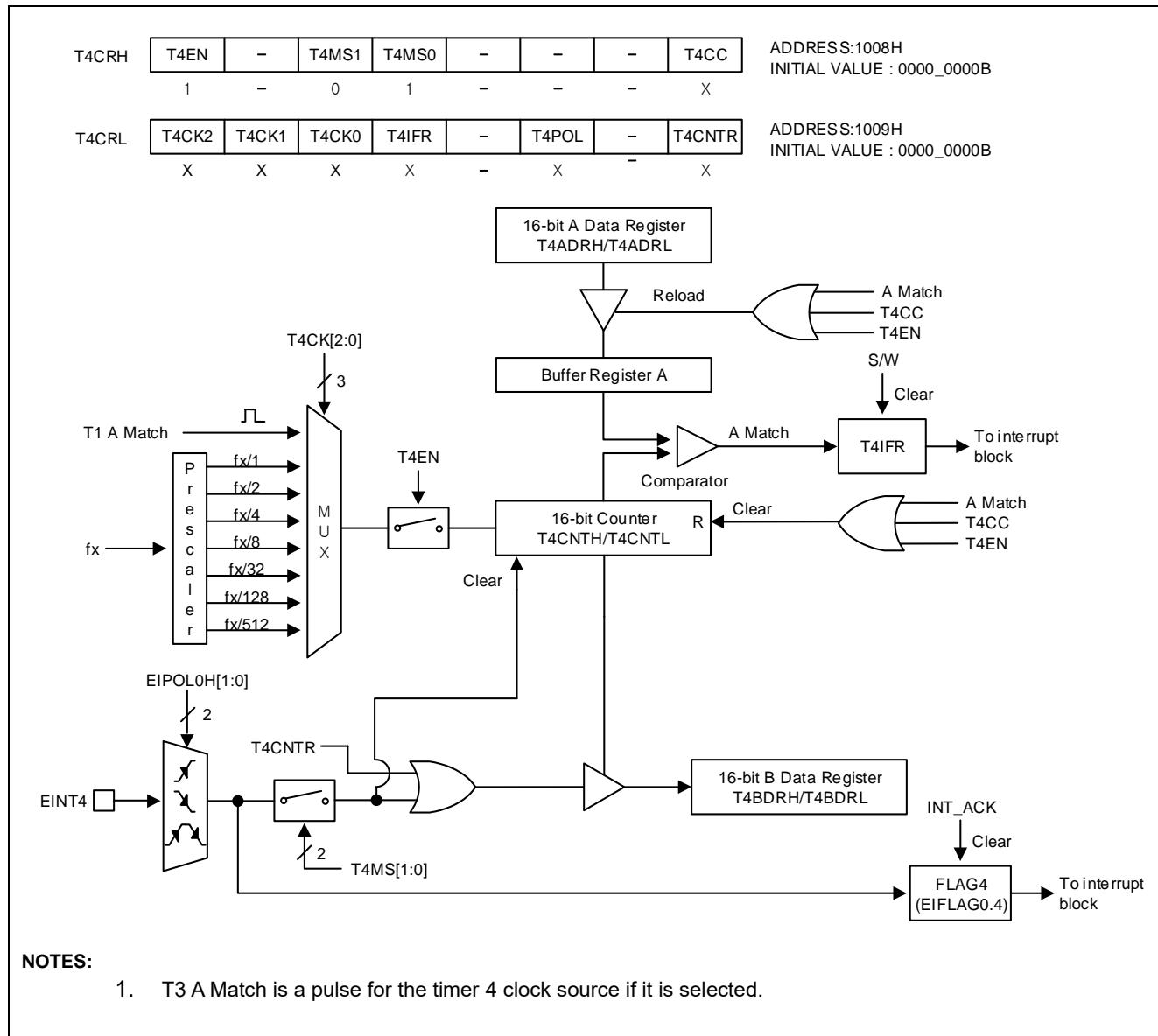


Figure 63. 16-bit Capture Mode for Timer 4

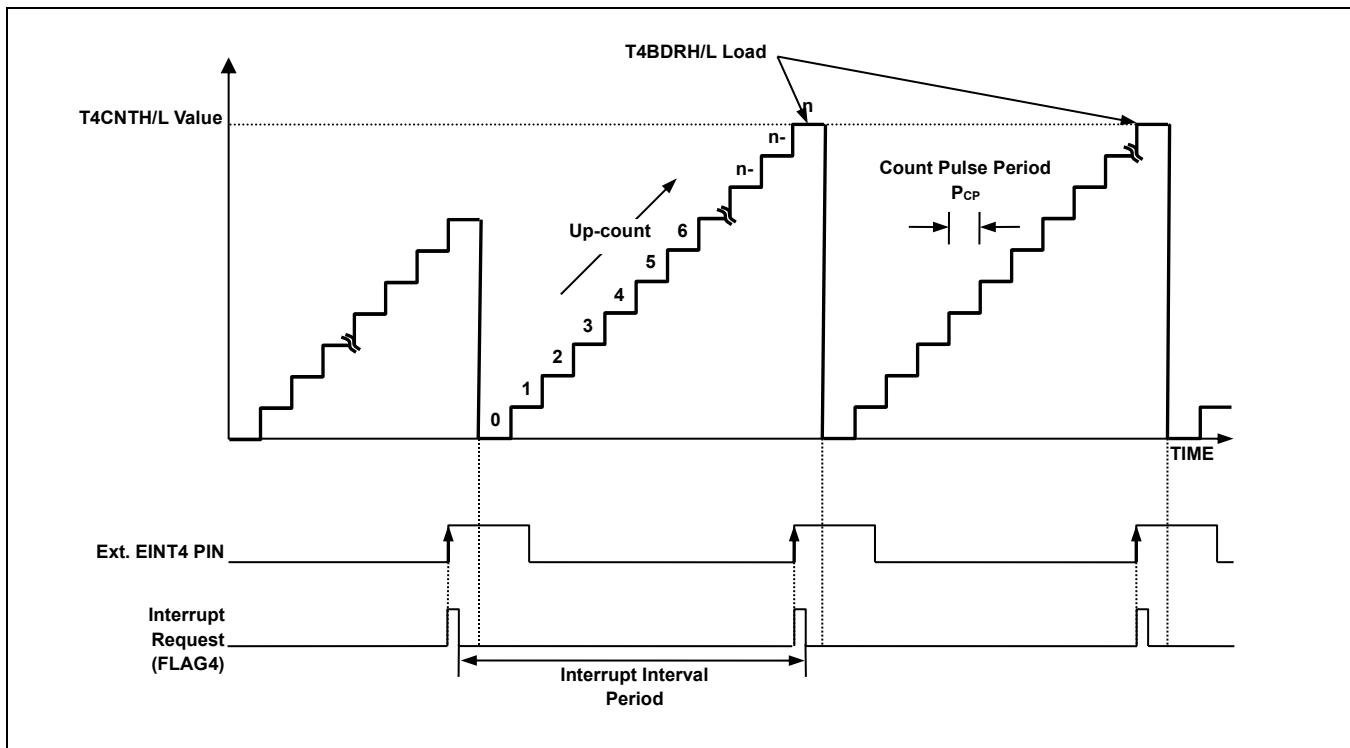


Figure 64. Input Capture Mode Operation for Timer 4

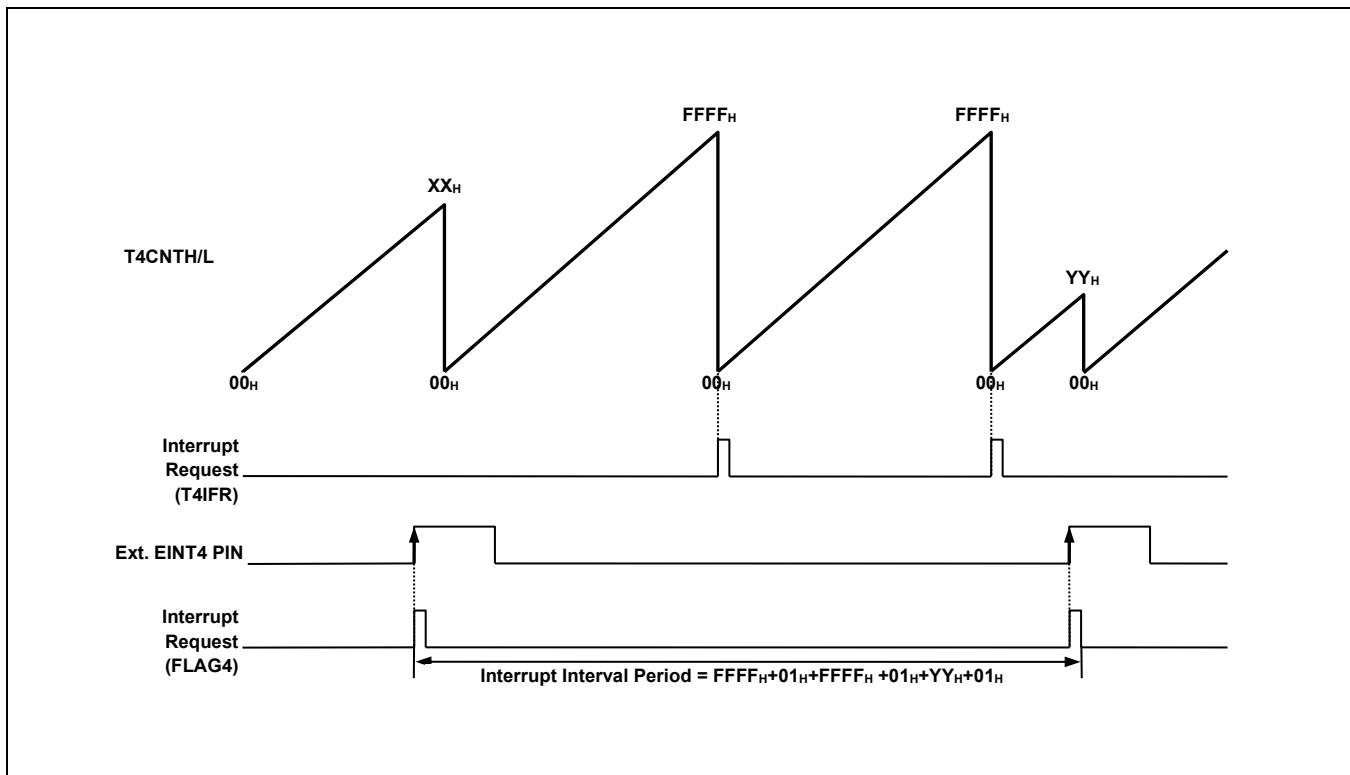


Figure 65. Express Timer Overflow in Capture Mode

### 11.5.4 16-bit PPG Mode

The timer 4 has a PPG (Programmable Pulse Generation) function. In PPG mode, the T4O/PWM4O pin outputs up to 16-bit resolution PWM output. This pin should be configured as a PWM output by set P0FSRH[5:4] to '11'. The period of the PWM output is determined by the T4ADRH/T4ADRL. And the duty of the PWM output is determined by the T4BDRH/T4BDRL.

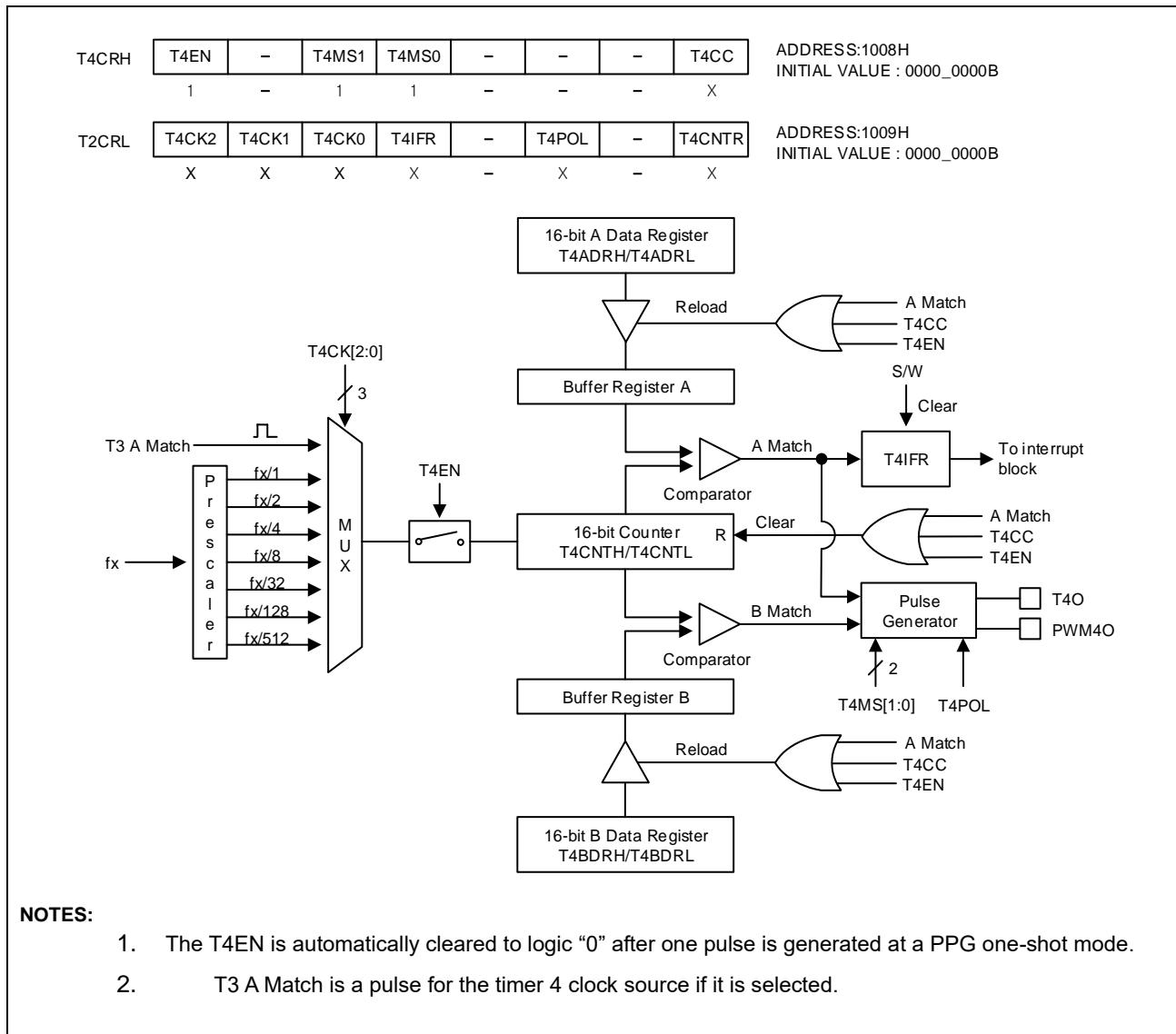


Figure 66. 16-bit PPG Mode for Timer 4

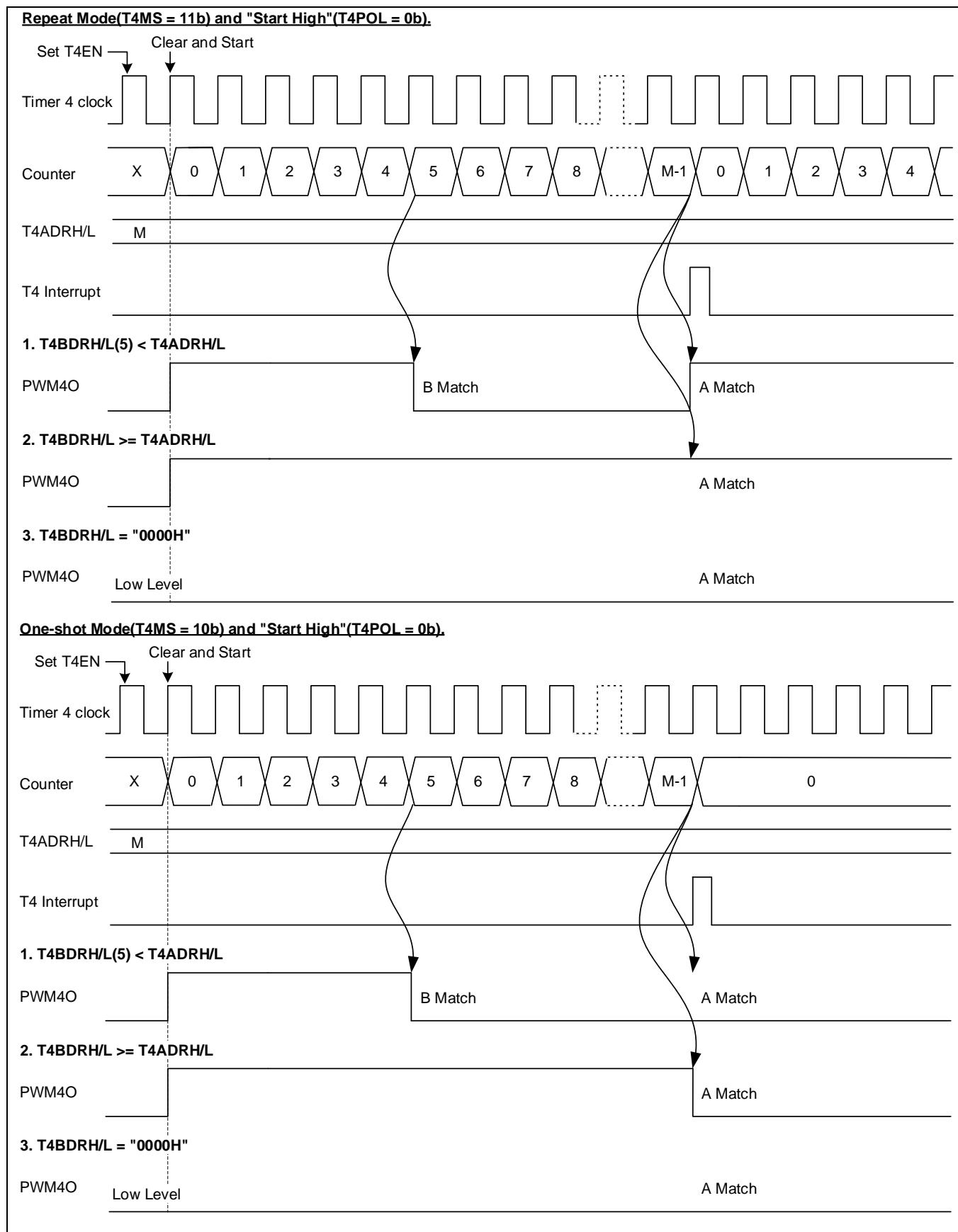


Figure 67. 16-bit PPG Mode Timming chart for Timer 4

### 11.5.5 Block Diagram

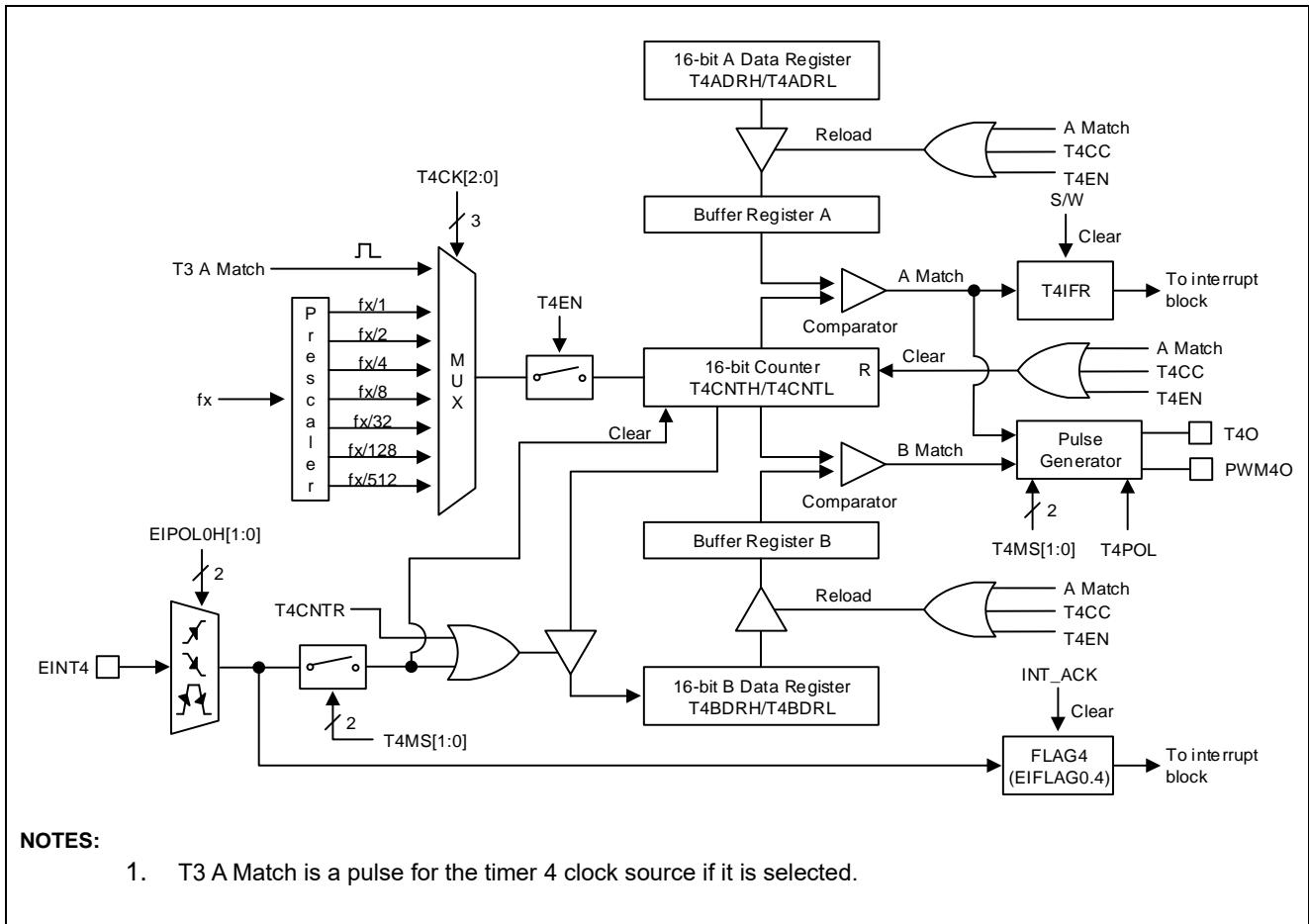


Figure 68. 16-bit Timer 4 Block Diagram

### 11.5.6 Register Map

Name	Address	Direction	Default	Description
T4ADRH	100AH	R/W	FFH	Timer 4 A Data High Register
T4ADRL	100BH	R/W	FFH	Timer 4 A Data Low Register
T4BDRH	100CH	R/W	FFH	Timer 4 B Data High Register
T4BDRL	100DH	R/W	FFH	Timer 4 B Data Low Register
T4CRH	1008H	R/W	00H	Timer 4 Control High Register
T4CRL	1009H	R/W	00H	Timer 4 Control Low Register

Table 24. Timer 4 Register Map

### 11.5.7 Timer/Counter 4 Register Description

The timer/counter 4 register consists of timer 4 A data high register (T4ADRH), timer 4 A data low register (T4ADRL), timer 4 B data high register (T4BDRH), timer 4 B data low register (T4BDRL), timer 4 control high register (T4CRH) and timer 4 control low register (T4CRL).

### 11.5.8 Register Description for Timer/Counter 4

#### T4ADRH (Timer 4 A Data High Register): 100AH

7	6	5	4	3	2	1	0
T4ADR7	T4ADR6	T4ADR5	T4ADR4	T4ADR3	T4ADR2	T4ADR1	T4ADR0
R/W							

Initial value: FFH

T4ADR7[7:0]      T4 A Data High Byte

#### T4ADRL (Timer 4 A Data Low Register): 100BH

7	6	5	4	3	2	1	0
T4ADRL7	T4ADRL6	T4ADRL5	T4ADRL4	T4ADRL3	T4ADRL2	T4ADRL1	T4ADRL0
R/W							

Initial value: FFH

T4ADRL7[7:0]      T4 A Data Low Byte

**NOTES:**  
Do not write "0000H" in the T4ADR7/T4ADRL7 register when PPG mode.

#### T4BDRH (Timer 4 B Data High Register): 100CH

7	6	5	4	3	2	1	0
T4BDRH7	T4BDRH6	T4BDRH5	T4BDRH4	T4BDRH3	T4BDRH2	T4BDRH1	T4BDRH0
R/W							

Initial value: FFH

T4BDRH7[7:0]      T4 B Data High Byte

#### T4BDRL (Timer 4 B Data Low Register): 100DH

7	6	5	4	3	2	1	0
T4BDRL7	T4BDRL6	T4BDRL5	T4BDRL4	T4BDRL3	T4BDRL2	T4BDRL1	T4BDRL0
R/W							

Initial value: FFH

T4BDRL7[7:0]      T4 B Data Low

**T4CRH (Timer 4 Control High Register): 1008H**

7	6	5	4	3	2	1	0
T4EN	-	T4MS1	T4MS0	-	-	-	T4CC
RW	-	R/W	R/W	-	-	-	RW

Initial value: 00H

T4EN	Control Timer 4			
	0 Timer 4 disable			
	1 Timer 4 enable (Counter clear and start)			
T4MS[1:0]	Control Timer 4 Operation Mode			
	T4MS1	T4MS0	Description	
	0	0	Timer/counter mode (T4O: toggle at A match)	
	0	1	Capture mode (The A match interrupt can occur)	
	1	0	PPG one-shot mode (PWM4O)	
	1	1	PPG repeat mode (PWM4O)	
T4CC	Clear Timer 4 Counter			
	0	No effect		
	1	Clear the Timer 4 counter (When write, automatically cleared "0" after being cleared counter)		

**T4CRL (Timer 4 Control Low Register): 1009H**

7	6	5	4	3	2	1	0
T4CK2	T4CK1	T4CK0	T4IFR	-	T4POL	-	T4CNTR
R/W	R/W	R/W	R/W	-	RW	-	RW

Initial value: 00H

T4CK[2:0]	Select Timer 4 clock source. fx is main system clock frequency			
	T4CK2	T4CK1	T4CK0	
	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	T3 A Match
T4IFR	When T4 Match Interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit. Writing "1" has no effect.			
	0	T4 interrupt no generation		
	1	T4 interrupt generation		
T4POL	T4O/PWM4O Polarity Selection			
	0	Start High (T4O/PWM4O is low level at disable)		
	1	Start Low (T4O/PWM4O is high level at disable)		
T4CNTR	Timer 4 Counter Read Control			
	0	No effect		
	1	Load the counter value to the B data register (When write, automatically cleared "0" after being loaded)		

## 11.6 Timer 5

### 11.6.1 Overview

The 16-bit TIMER 5 consists of multiplexer, timer 5 A data high/low register, timer 5 B data high/low register and timer 5 control high/low register (T5ADR<sub>H</sub>, T5ADRL, T5BDR<sub>H</sub>, T5BDRL, T5CR<sub>H</sub>, and T5CRL).

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 5 can be divided clock of the system clock selected from prescaler output.

The clock source is selected by clock selection logic which is controlled by the clock selection bits (T5CK[2:0]).

TIMER 5 clock source: fx/1, fx/2, fx/4, fx/8, fx/32, fx/128, fx/512 and HSIRC

In the capture mode, by EINT5, the data is captured into input capture data register (T5BDR<sub>H</sub>/T5BDRL). In timer/counter mode, whenever counter value is equal to T5ADR<sub>H/L</sub>, T5O port toggles. Also the TIMER 5 outputs PWM wave form to PWM5O port in the PPG mode.

**Table 25. TIMER 5 Operating Modes**

T5EN	P0FSRH[5:4]	T5MS[1:0]	T5CK[2:0]	TIMER 5
1	11	00	XXX	16-bit Timer/Counter Mode
1	00	01	XXX	16-bit Capture Mode
1	11	10	XXX	16-bit PPG Mode (one-shot mode)
1	11	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 72.

The 16-bit timer have counter and data register. The counter register is increased by internal clock input. Timer 5 can use the input clock with one of 1, 2, 4, 8, 32, 128, 512 and High Frequency Internal Oscillator (HSIRC) prescaler division rates (T5CK[2:0]). When the values of T5CNTH/T5CNTL and T5ADR<sub>H</sub>/T5ADRL are identical in timer 5, a match signal is generated and the interrupt of Timer 5 occurs. The T5CNTH/T5CNTL values are automatically cleared by match signal. It can be also cleared by software (T5CC).

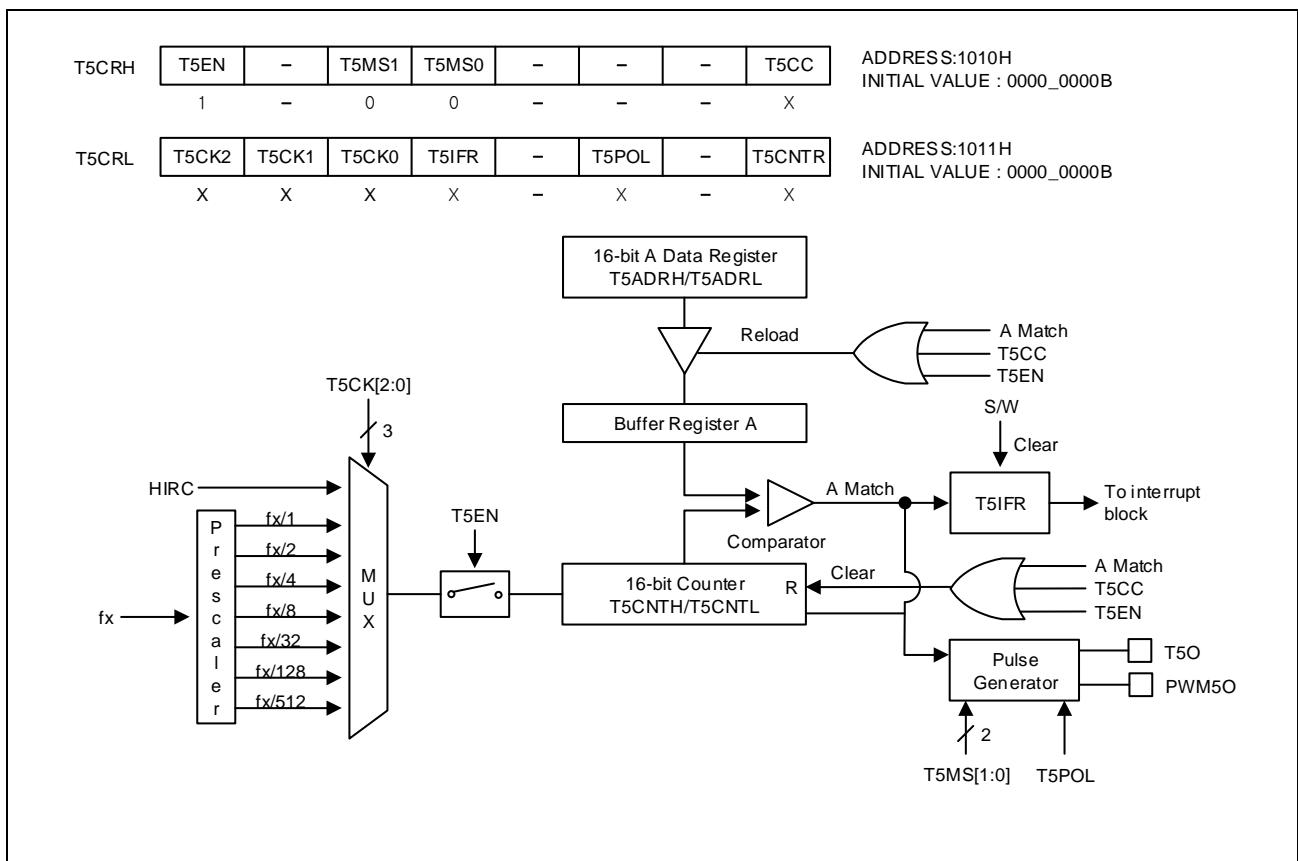


Figure 69. 16-bit Timer/Counter Mode for Timer 5

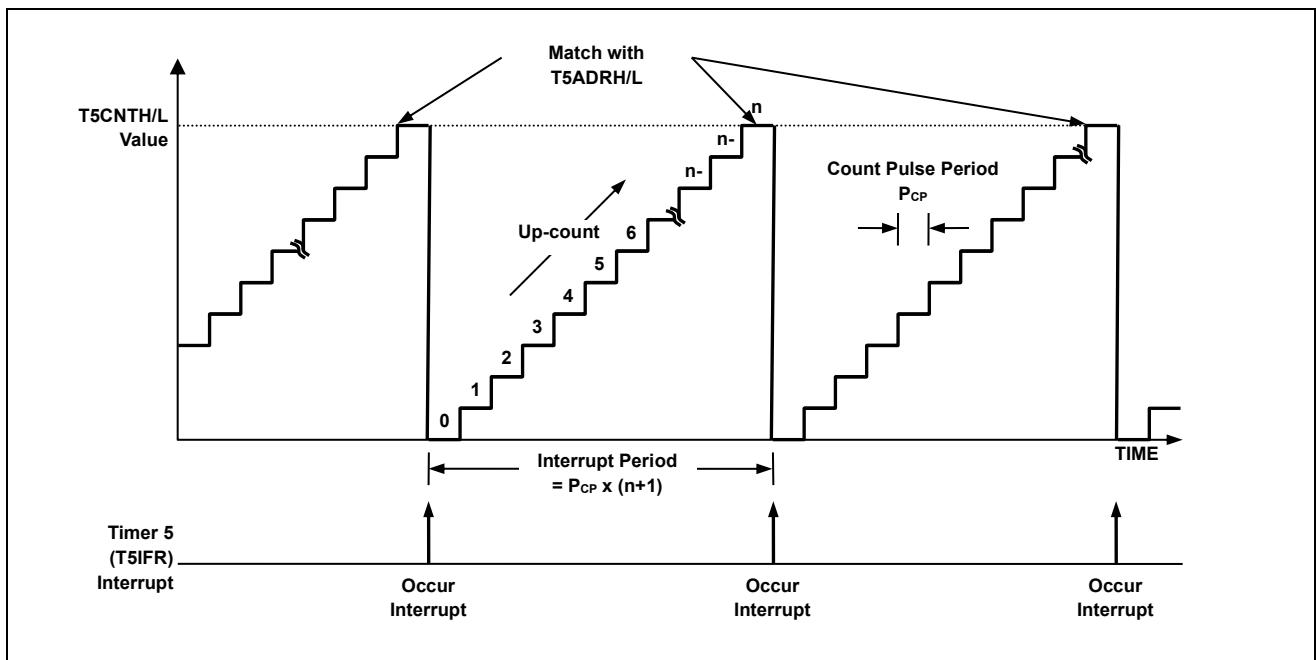


Figure 70. 16-bit Timer/Counter 4 Example

### 11.6.3 16-bit Capture Mode

The timer 5 capture mode is set by T5MS[1:0] as '01'. The clock source can use the internal clock. Basically, it has the same function as the 16-bit timer/counter mode and the interrupt occurs when T5CNTH/T5CNTL is equal to T5ADRH/T5ADRL. T5CNTH/T5CNTL values are automatically cleared by match signal and it can be also cleared by software (T5CC).

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 T5BDRH/T5BDRL. In the timer 5 capture mode, timer 5 output (T5O) waveform is not available.

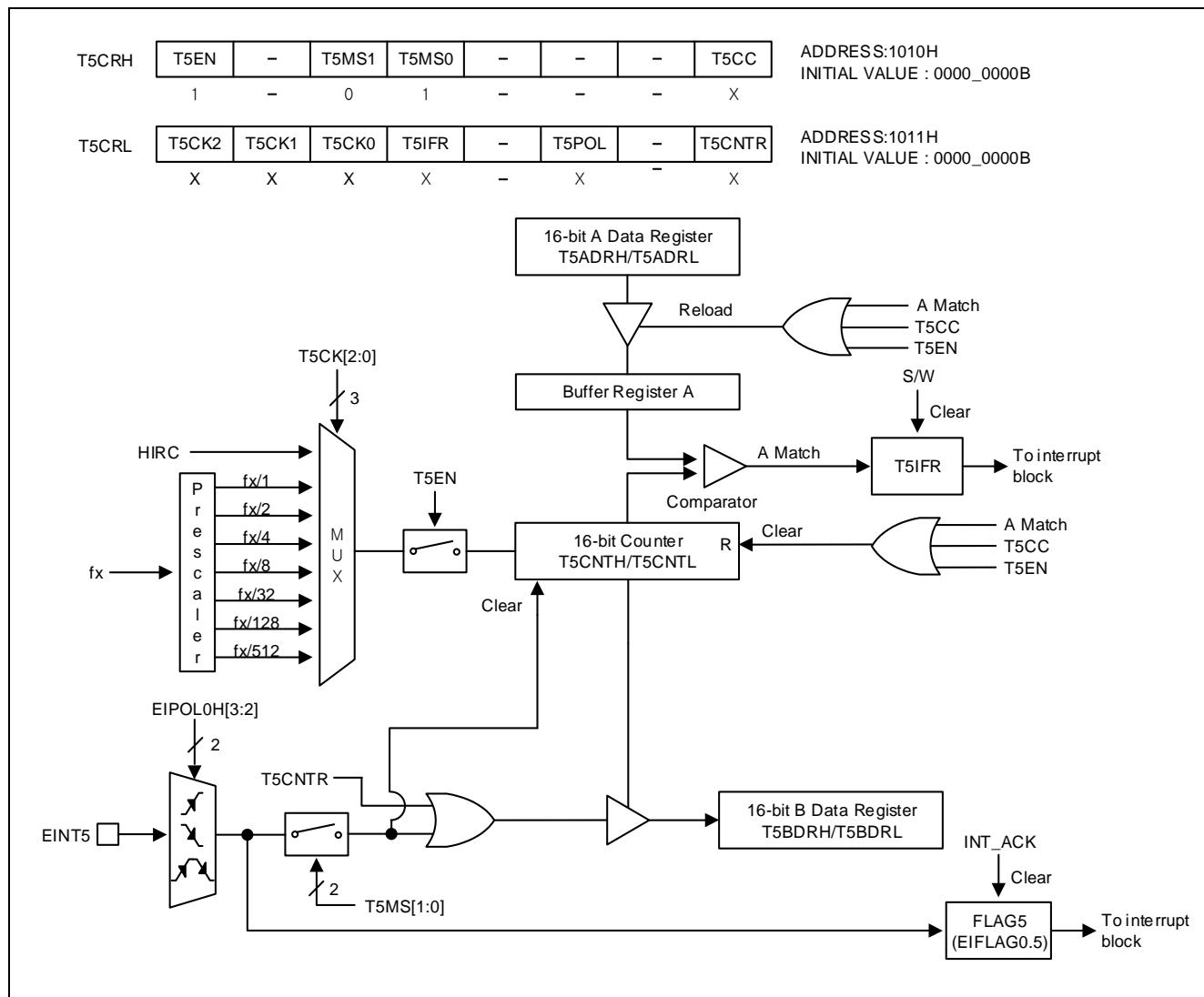


Figure 71. 16-bit Capture Mode for Timer 5

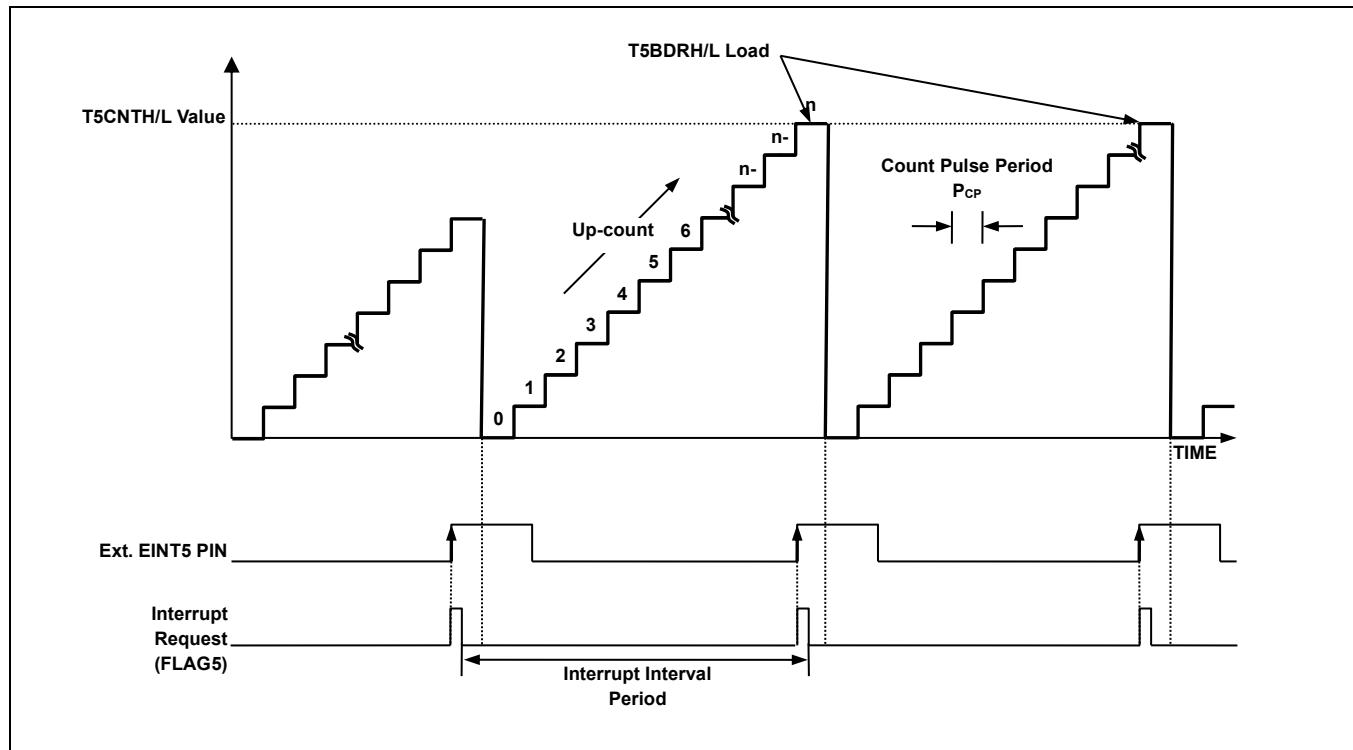


Figure 72. Input Capture Mode Operation for Timer 5

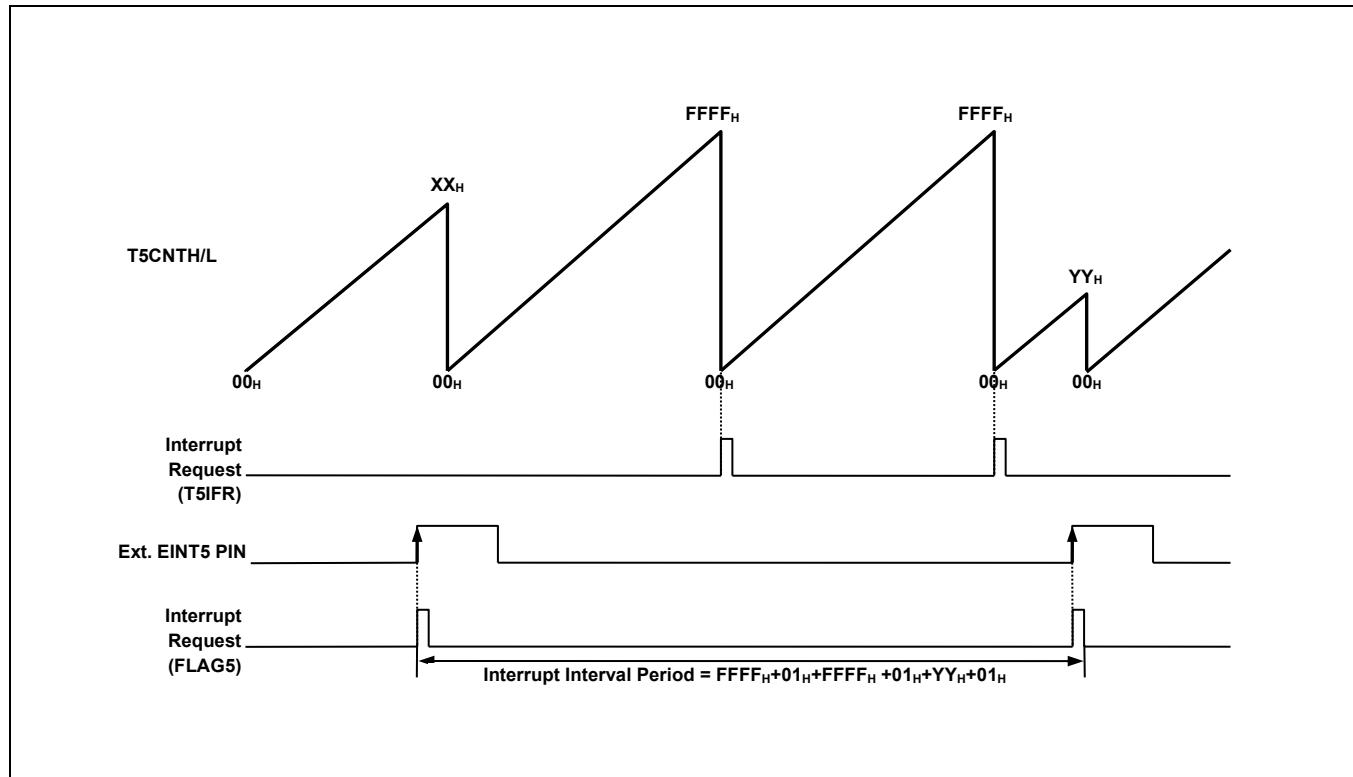


Figure 73. Express Timer Overflow in Capture Mode

### 11.6.4 16-bit PPG Mode

The timer 5 has a PPG (Programmable Pulse Generation) function. In PPG mode, the T5O/PWM5O pin outputs up to 16-bit resolution PWM output. This pin should be configured as a PWM output by set P0FSRH[7:6] to '11'. The period of the PWM output is determined by the T5ADRH/T5ADRL. And the duty of the PWM output is determined by the T5BDRH/T5BDRL.

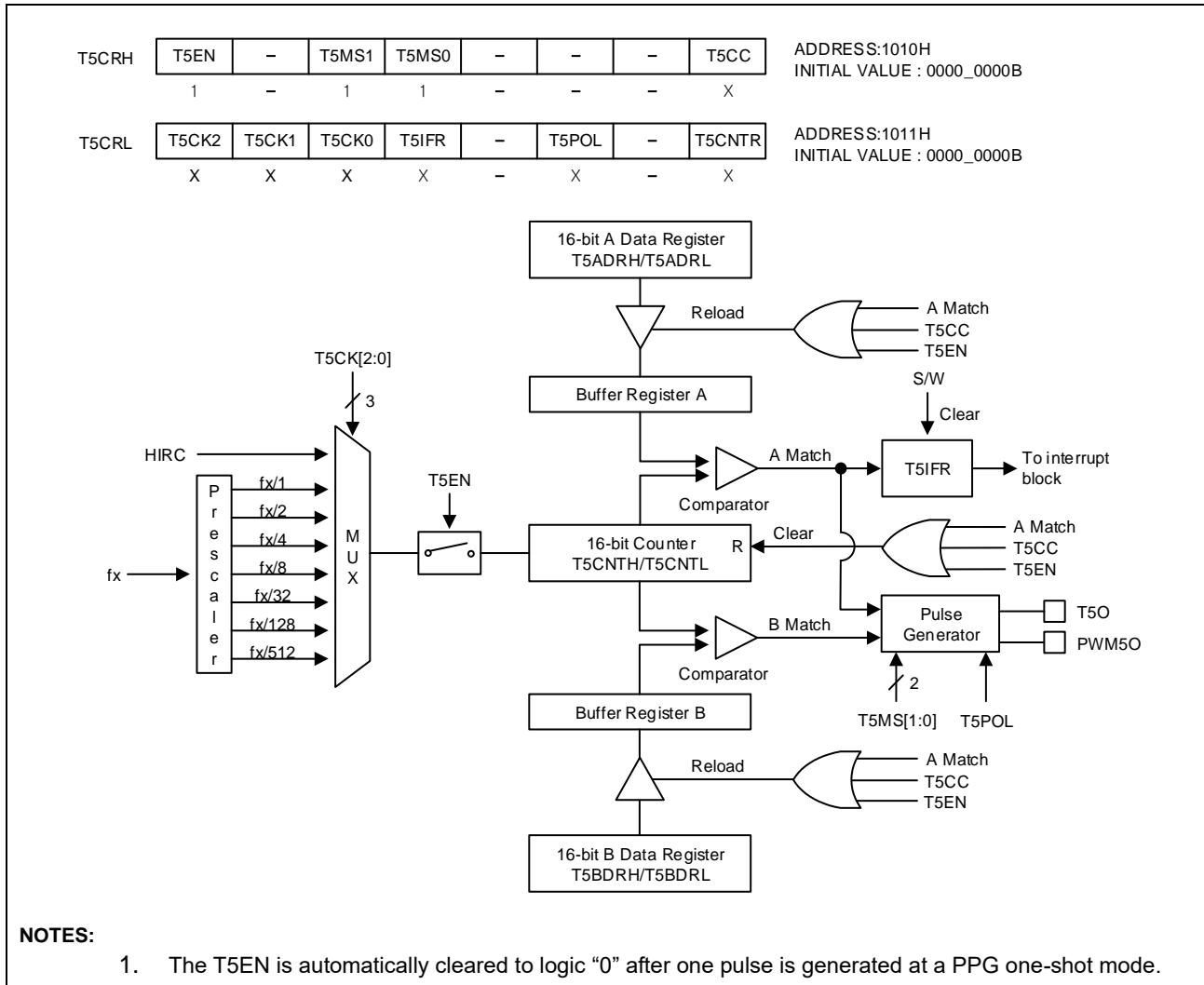


Figure 74. 16-bit PPG Mode for Timer 5

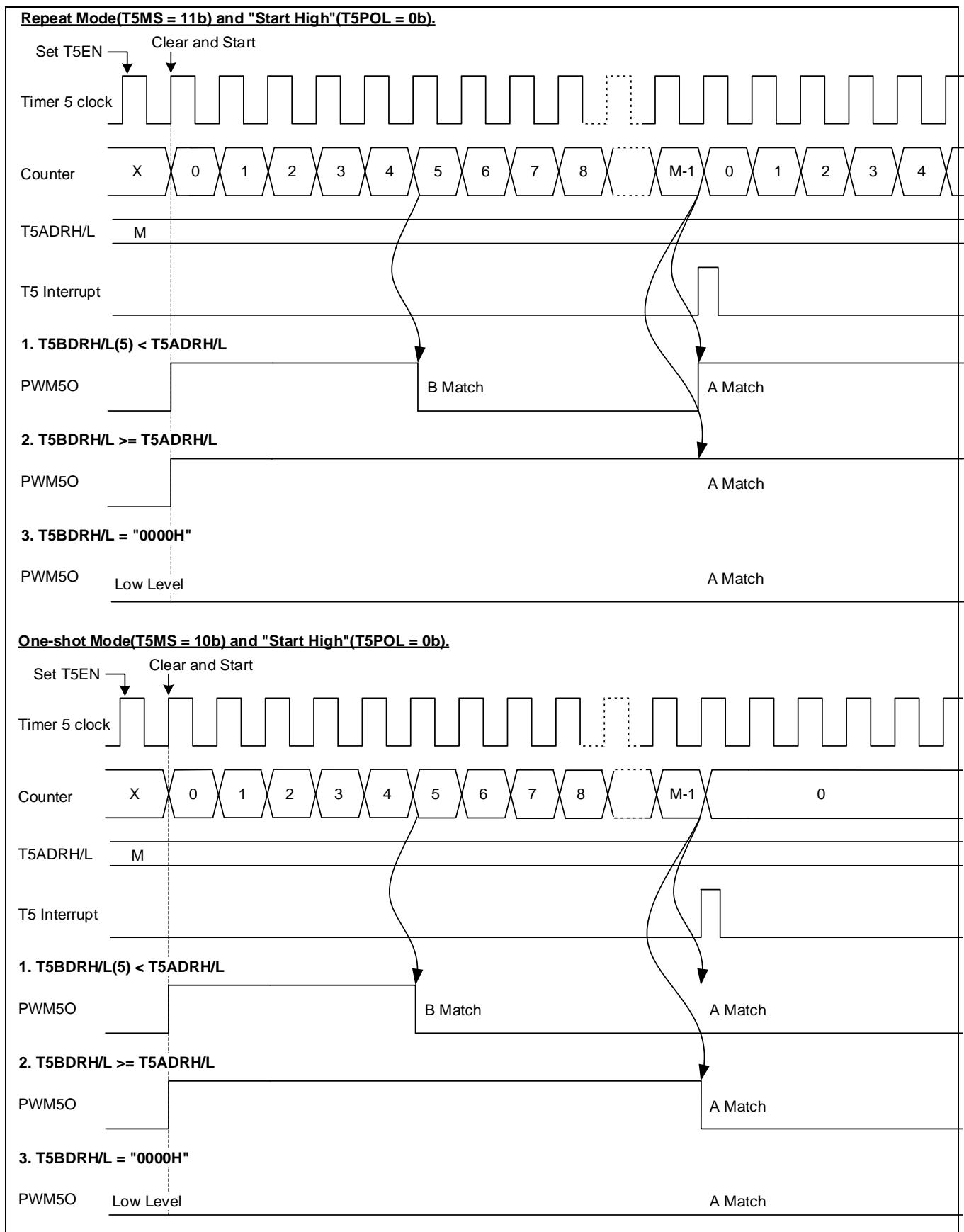
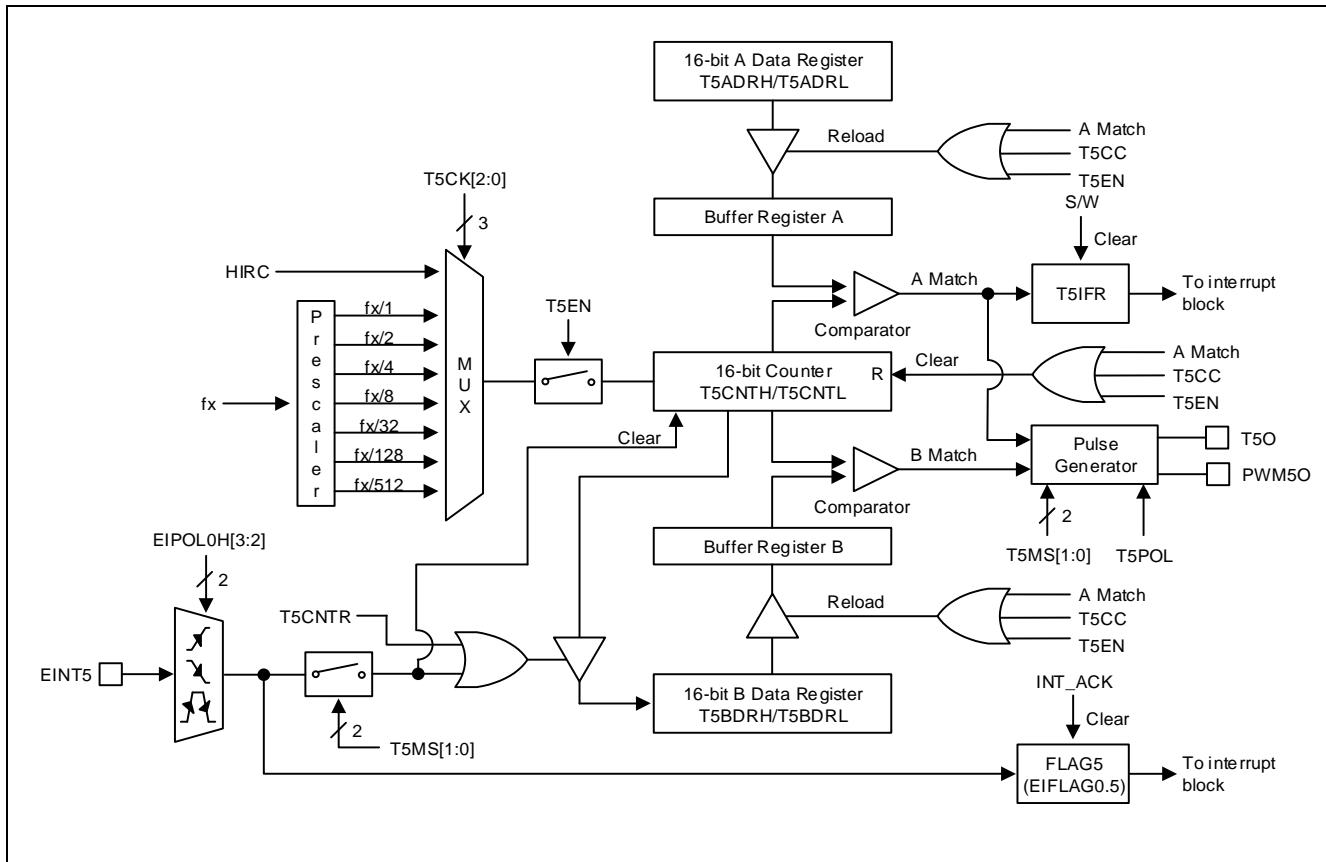


Figure 75. 16-bit PPG Mode Timming chart for Timer 5

### 11.6.5 Block Diagram



**Figure 76. 16-bit Timer 5 Block Diagram**

### 11.6.6 Register Map

**Table 26. Timer 5 Register Map**

Name	Address	Direction	Default	Description
T5ADRH	1012H	R/W	FFH	Timer 5 A Data High Register
T5ADRL	1013H	R/W	FFH	Timer 5 A Data Low Register
T5BDRH	1014H	R/W	FFH	Timer 5 B Data High Register
T5BDRL	1015H	R/W	FFH	Timer 5 B Data Low Register
T5CRH	1010H	R/W	00H	Timer 5 Control High Register
T5CRL	1011H	R/W	00H	Timer 5 Control Low Register

### 11.6.7 Timer/Counter 5 Register Description

The timer/counter 5 register consists of timer 5 A data high register (T5ADRH), timer 5 A data low register (T5ADRL), timer 5 B data high register (T5BDRH), timer 5 B data low register (T5BDRL), timer 5 control high register (T5CRH) and timer 5 control low register (T5CRL).

### 11.6.8 Register Description for Timer/Counter 5

#### T5ADR<sub>H</sub> (Timer 5 A Data High Register): 1012H

7	6	5	4	3	2	1	0
T5ADR <sub>H7</sub>	T5ADR <sub>H6</sub>	T5ADR <sub>H5</sub>	T5ADR <sub>H4</sub>	T5ADR <sub>H3</sub>	T5ADR <sub>H2</sub>	T5ADR <sub>H1</sub>	T5ADR <sub>H0</sub>
R/W							

Initial value: FFH

T5ADR<sub>H[7:0]</sub> T5 A Data High Byte

#### T5ADRL (Timer 5 A Data Low Register): 1013H

7	6	5	4	3	2	1	0
T5ADRL <sub>7</sub>	T5ADRL <sub>6</sub>	T5ADRL <sub>5</sub>	T5ADRL <sub>4</sub>	T5ADRL <sub>3</sub>	T5ADRL <sub>2</sub>	T5ADRL <sub>1</sub>	T5ADRL <sub>0</sub>
R/W							

Initial value: FFH

T5ADRL[7:0] T5 A Data Low Byte

#### NOTES:

Do not write "0000H" in the T5ADR<sub>H</sub>/T5ADRL register when PPG mode.

#### T5BDR<sub>H</sub> (Timer 5 B Data High Register): 1014H

7	6	5	4	3	2	1	0
T5BDR <sub>H7</sub>	T5BDR <sub>H6</sub>	T5BDR <sub>H5</sub>	T5BDR <sub>H4</sub>	T5BDR <sub>H3</sub>	T5BDR <sub>H2</sub>	T5BDR <sub>H1</sub>	T5BDR <sub>H0</sub>
R/W							

Initial value: FFH

T5BDR<sub>H[7:0]</sub> T5 B Data High Byte

#### T5BDRL (Timer 5 B Data Low Register): 1015H

7	6	5	4	3	2	1	0
T5BDRL <sub>7</sub>	T5BDRL <sub>6</sub>	T5BDRL <sub>5</sub>	T5BDRL <sub>4</sub>	T5BDRL <sub>3</sub>	T5BDRL <sub>2</sub>	T5BDRL <sub>1</sub>	T5BDRL <sub>0</sub>
R/W							

Initial value: FFH

T5BDRL[7:0] T5 B Data Low Byte

**T5CRH (Timer 5 Control High Register): 1010H**

7	6	5	4	3	2	1	0
T5EN	-	T5MS1	T5MS0	-	-	-	T5CC
R/W	-	R/W	R/W	-	-	-	R/W

Initial value: 00H

T5EN	Control Timer 5			
	0 Timer 5 disable			
	1 Timer 5 enable (Counter clear and start)			
T5MS[1:0]	Control Timer 5 Operation Mode			
	T5MS1	T5MS0	Description	
	0	0	Timer/counter mode (T5O: toggle at A match)	
	0	1	Capture mode (The A match interrupt can occur)	
	1	0	PPG one-shot mode (PWM5O)	
	1	1	PPG repeat mode (PWM5O)	
T5CC	Clear Timer 5 Counter			
	0	No effect		
	1	Clear the Timer 5 counter (When write, automatically cleared "0" after being cleared counter)		

**T5CRL (Timer 5 Control Low Register): 1011H**

7	6	5	4	3	2	1	0
T5CK2	T5CK1	T5CK0	T5IFR	-	T5POL	-	T5CNTR
R/W	R/W	R/W	R/W	-	R/W	-	R/W

Initial value: 00H

T5CK[2:0]	Select Timer 5 clock source. fx is main system clock frequency			
	T5CK2	T5CK1	T5CK0	
	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	HSIRC Direct (16MHz)
T5IFR	When T5 Match Interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit. Writing "1" has no effect.			
	0	T5 interrupt no generation		
	1	T5 interrupt generation		
T5POL	T5O/PWM5O Polarity Selection			
	0	Start High (T5O/PWM5O is low level at disable)		
	1	Start Low (T5O/PWM5O is high level at disable)		
T5CNTR	Timer 5 Counter Read Control			
	0	No effect		
	1	Load the counter value to the B data register (When write, automatically cleared "0" after being loaded)		

## 12 Buzzer Driver

### 12.1 Overview

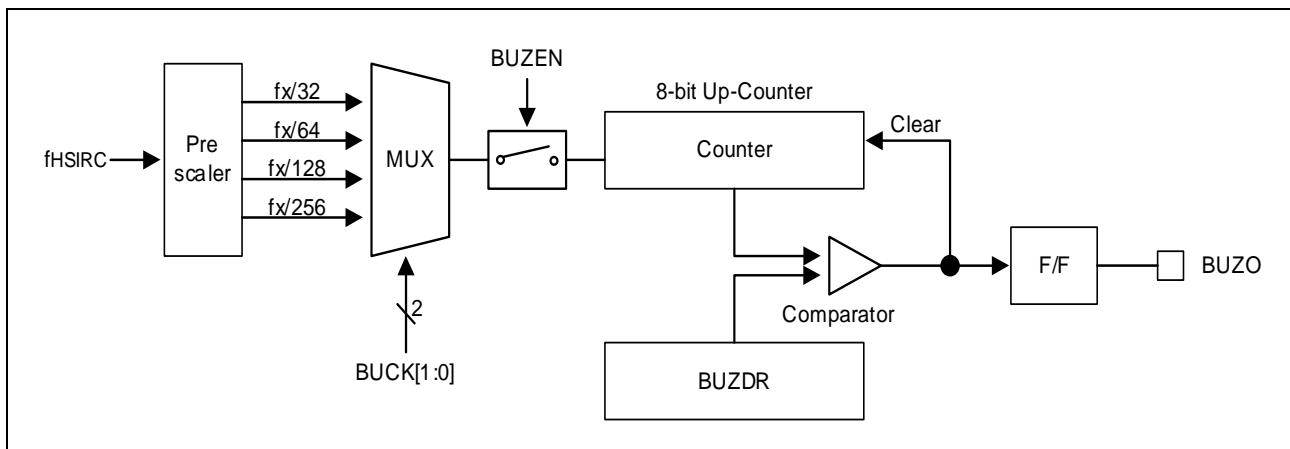
The Buzzer consists of 8bit counter, buzzer data register (BUZDR), and buzzer control register (BUZCR). The Square Wave (122.07Hz~250KHz @16MHz) is outputted through P03/AN1/EINT1BUZ0 pin. The buzzer data register (BUZDR) controls the buzzer frequency (look at the following expression). In buzzer control register (BUZCR), BUCK[1:0] selects source clock divided by prescaler.

$$f_{BUZ}(\text{Hz}) = \frac{\text{Oscillator Frequency}}{2 \times \text{Prescaler Ratio} \times (\text{BUZDR} + 1)}$$

**Table 27. Buzzer Frequency at 16 MHz**

BUZDR[7:0]	Buzzer Frequency (kHz)			
	BUZCR[2:1]=00	BUZCR[2:1]=01	BUZCR[2:1]=10	BUZCR[2:1]=11
0000_0000	250kHz	125.00kHz	62.50kHz	31.250kHz
0000_0001	125kHz	62.50kHz	31.25kHz	15.625kHz
...	...	...	...	...
1111_1101	984.252Hz	492.126Hz	246.063Hz	123.031Hz
1111_1110	980.392Hz	490.196Hz	245.098Hz	122.549Hz
1111_1111	976.563Hz	488.281Hz	244.141Hz	122.070Hz

### 12.2 Block Diagram



**Figure 77. Buzzer Driver Block Diagram**

## 12.3 Register Map

**Table 28. Buzzer Driver Register Map**

Name	Address	Direction	Default	Description
BUZDR	8FH	R/W	FFH	Buzzer Data Register
BUZCR	97H	R/W	00H	Buzzer Control Register

## 12.4 Buzzer Driver Register Description

Buzzer driver consists of buzzer data register (BUZDR) and buzzer control register (BUZCR).

### 12.5 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
RW							

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
-	-	-	-	-	BUCK1	BUCK0	BUZEN
-	-	-	-	-	RW	RW	RW

Initial value: 00H

BUCK[1:0]      Buzzer Driver Source Clock Selection  

BUCK1	BUCK0	Description
0	0	fHSIRC/32
0	1	fHSIRC/64
1	0	fHSIRC/128
1	1	fHSIRC/256

  
 BUZEN      Buzzer Driver Operation Control  

	Description
0	Buzzer Driver disable
1	Buzzer Driver enable

#### NOTES:

1. fx: System clock oscillation frequency.

## 13 USI (USART + SPI + I2C)

### 13.1 Overview

The USI is an acronym of USART, SPI and I2C, A96T418 has two USI function blocks. Each USI consists of USI control register1/2/3/4, USI status register 1/2, USI baud-rate generation register, USI data register, USI SDA hold time register, USI SCL high period register, USI SCL low period register, and USI slave address register (USInCR1, USInCR2, USInCR3, USInCR4, USInST1, USInST2, USInBD, USInDR, where n is 0, and USInSDHR, USInSCHR, USInSCLR, USInSAR where n is only 0).

The operation mode is selected by the operation mode of USIn selection bits (USInMS[1:0]).

It has four operating modes:

- Asynchronous mode (UART)
- Synchronous mode (USART)
- SPI mode
- I2C mode

## 13.2 USIn UART Mode

The universal synchronous and asynchronous serial receiver and transmitter (USART) 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 are Supported by Hardware
- Data OverRun Detection
- Framing Error Detection
- Three Separate Interrupts on TX Completion, TX Data Register Empty and RX Completion
- Double Speed Asynchronous communication mode

The USIn comprises clock generator, transmitter and receiver. The clock generation logic consists of synchronization logic for external clock input used by synchronizing 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.

### 13.3 USIn UART Block Diagram

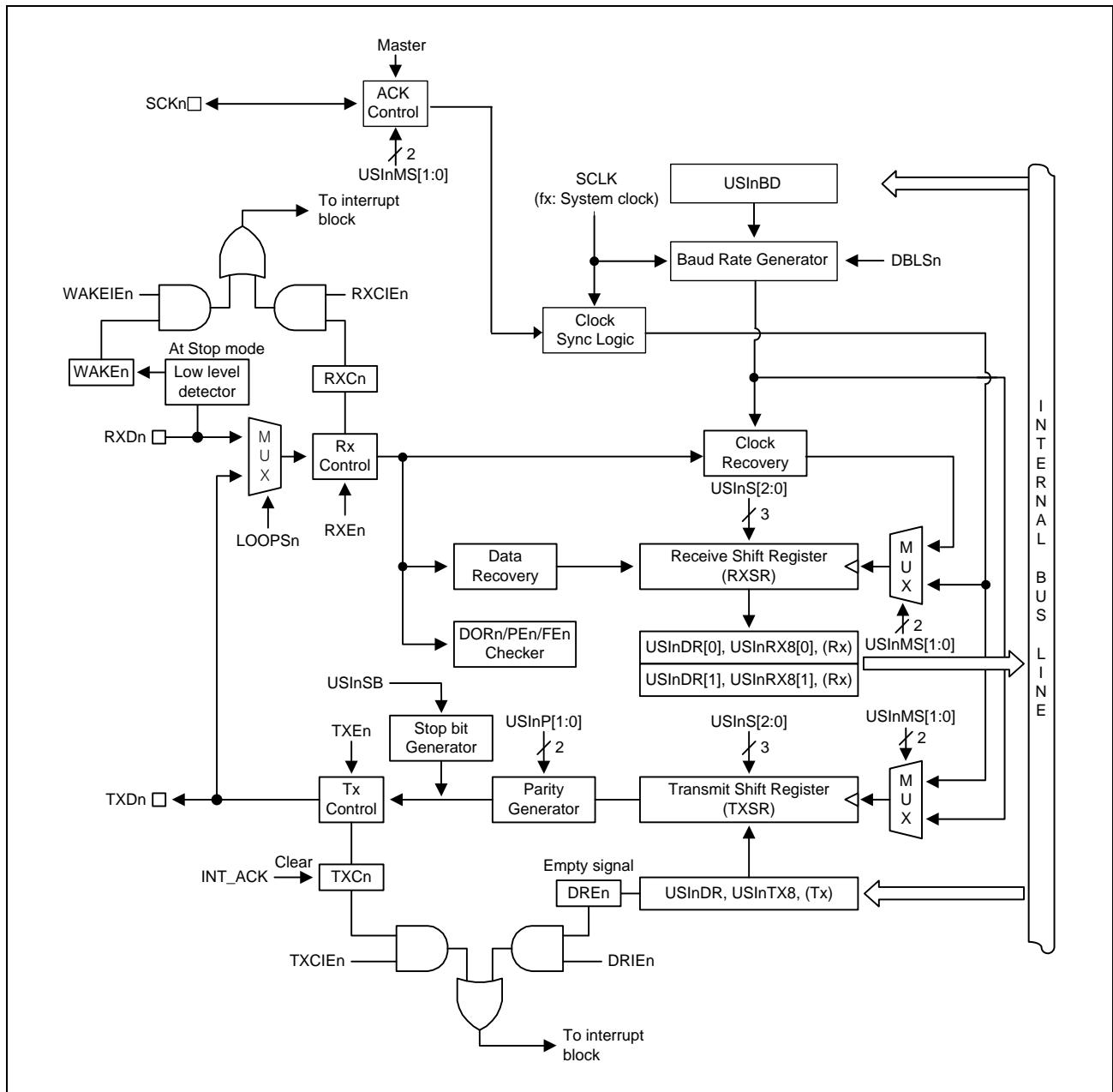


Figure 78. USIn USART Block Diagram(n = 0, 1)

### 13.4 USIn Clock Generation

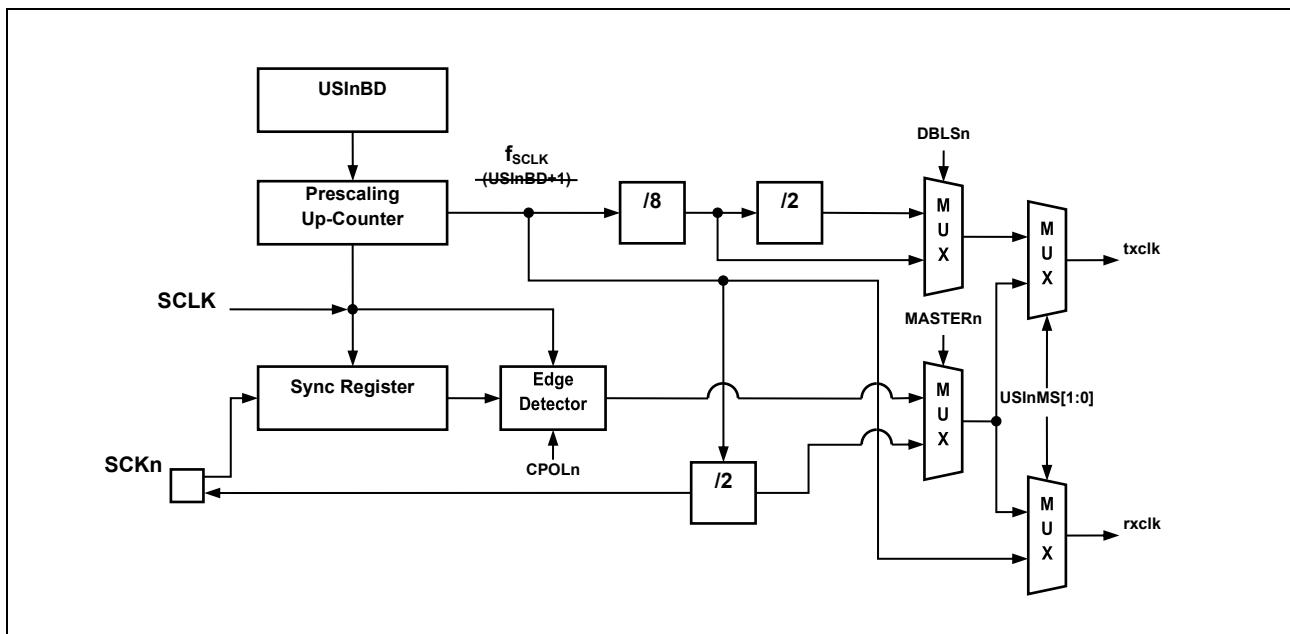


Figure 79. Clock Generation Block Diagram (USIn)

The clock generation logic generates the base clock for the transmitter and receiver. The USIn 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 USIn operates in synchronous or SPI mode.

Following table shows the equations for calculating the baud rate (in bps).

Table 29. Equations for Calculating USIn Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate
Asynchronous Normal Mode ( <b>DBLSn=0</b> )	Baud Rate = $\frac{fx}{16(USInBD + 1)}$
Asynchronous Double Speed Mode ( <b>DBLSn=1</b> )	Baud Rate = $\frac{fx}{8(USInBD + 1)}$
Synchronous or SPI Master Mode	Baud Rate = $\frac{fx}{2(USInBD + 1)}$

### 13.5 USIn 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.

## 13.6 USIn 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 (MISO 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.

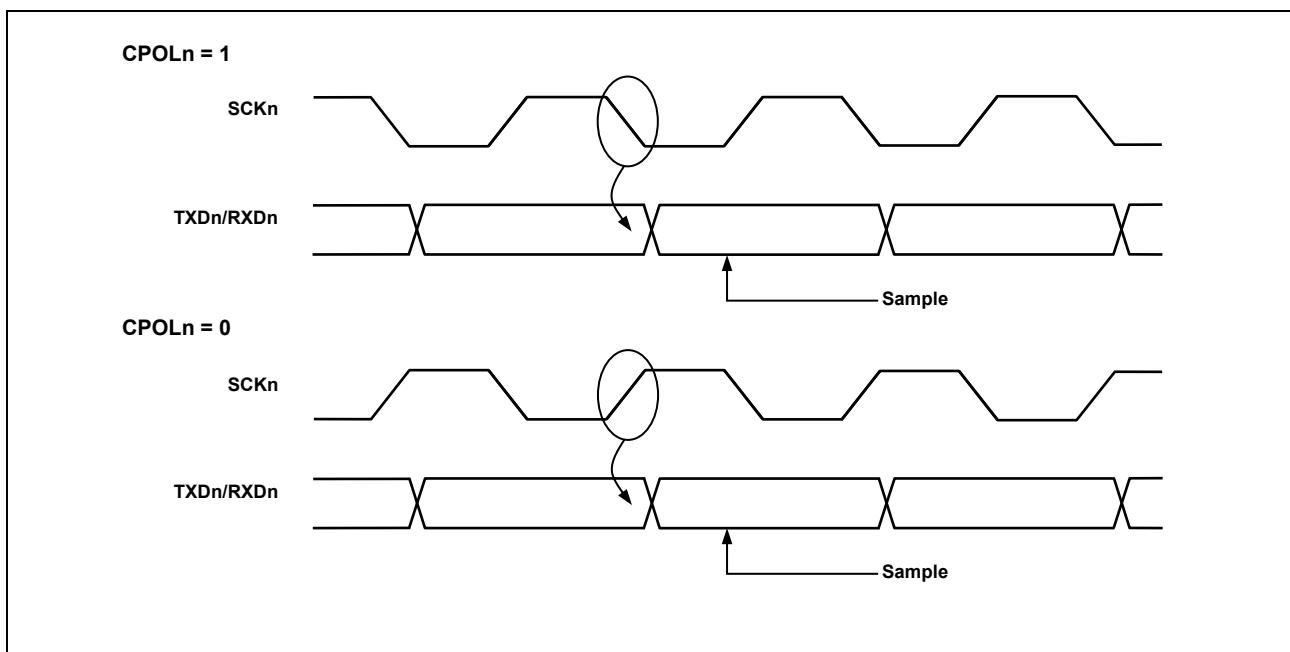


Figure 80. Synchronous Mode SCKn Timing (USIn)

### 13.7 USIn 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 USART 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.

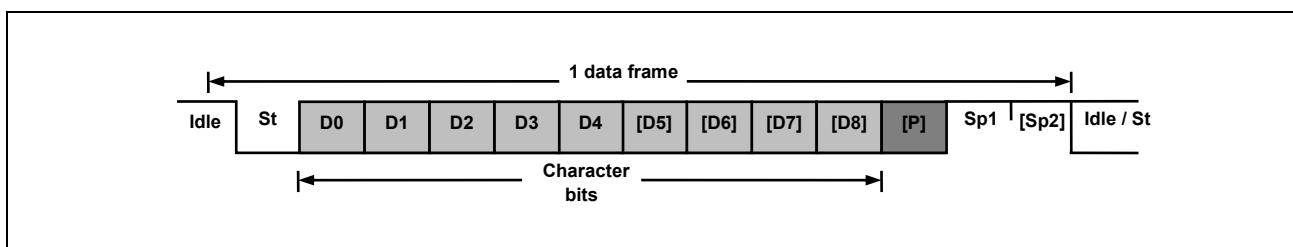


Figure 81. Frame Format (USIn)

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 USART 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 figures.

### 13.8 USIn 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-OR is inverted. The parity bit is located between the MSB and first stop bit of a serial frame.

$$P_{\text{even}} = D_{n-1} \wedge \dots \wedge D_3 \wedge D_2 \wedge D_1 \wedge D_0 \wedge 0$$

$$P_{\text{odd}} = D_{n-1} \wedge \dots \wedge D_3 \wedge D_2 \wedge D_1 \wedge D_0 \wedge 1$$

$P_{\text{even}}$  : Parity bit using even parity

$P_{\text{odd}}$  : Parity bit using odd parity

$D_n$  : Data bit n of the character

## 13.9 USIn 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 P1FSRH[7:6], P1FSRL[3:2], P0FSRL[5:4] and P0FSRL[1:0]. The baud-rate, operation mode and frame format must be set up 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 P1FSRH[3:2] and P0FSRL[7:6].

### 13.9.1 USIn UART Sending TX data

A data transmission is initiated by loading the transmit buffer (USInDRregister I/O location) with the data to be transmitted. The data be 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).

### 13.9.2 USIn UART Transmitter flag and interrupt

The USART transmitter has two flags which indicate its state. One is USART data register empty flag (DREn) and the other is transmit completion 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 prohibited.

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 serve 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, USART transmit complete interrupt is generated while TXCn flag is set.

### 13.9.3 USIn 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.

#### 13.9.4 USIn 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).

### 13.10 USIn UART Receiver

The USART 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 P1FSRH[5:4], P1FSRL[1:0], P0FSRL[3:2] and P3FSR[3:2]. 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 input, so it should be selected to do SCKn function by P1FSRH[3:2] and P0FSRL[7:6]. In SPI operation mode the SS<sub>n</sub> input pin in slave mode or can be configured as SS<sub>n</sub> output pin in master mode. This can be done by setting USInSSEN bit in USInCR3 register.

#### 13.10.1 USIn 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 RXD0 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 the second stop bit in the frame, the second stop bit is ignored by the receiver. That is, receiving the first stop bit means that a complete serial frame is presented 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 ninth 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.

### 13.10.2 USIn UART Receiver Flag and Interrupt

The UARTReceiver 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 is 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 datas 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 presented 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 avoid 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 PE bit is always read "0".

### 13.10.3 USIn 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.

### 13.10.4 USIn 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).

### 13.10.5 USIn 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 does sampling and low pass filtering the incoming bits, and removing 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.

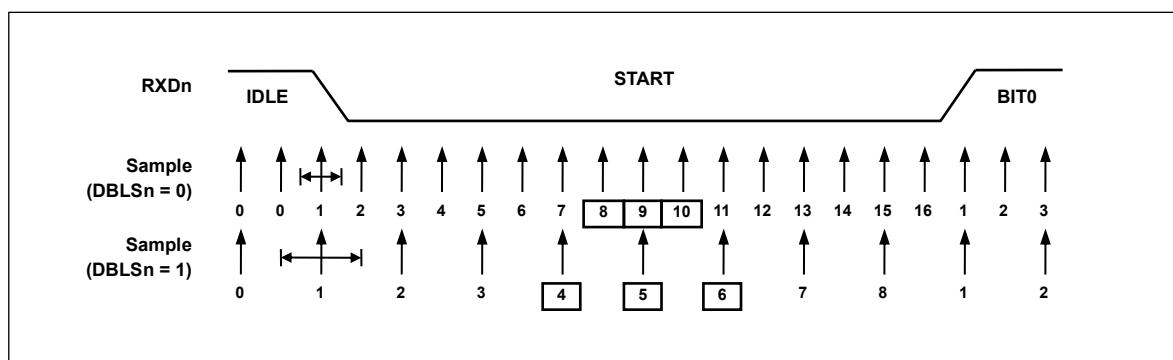
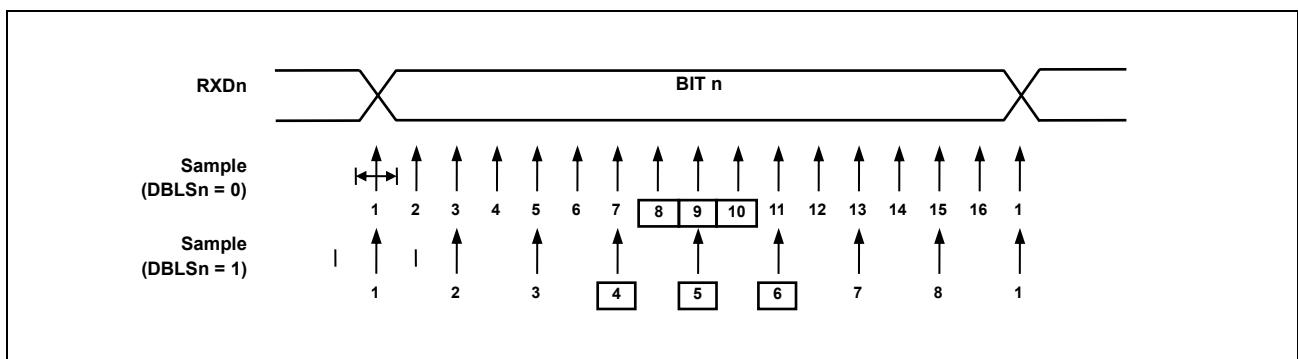


Figure 82. Asynchronous Start Bit Sampling (USIn)

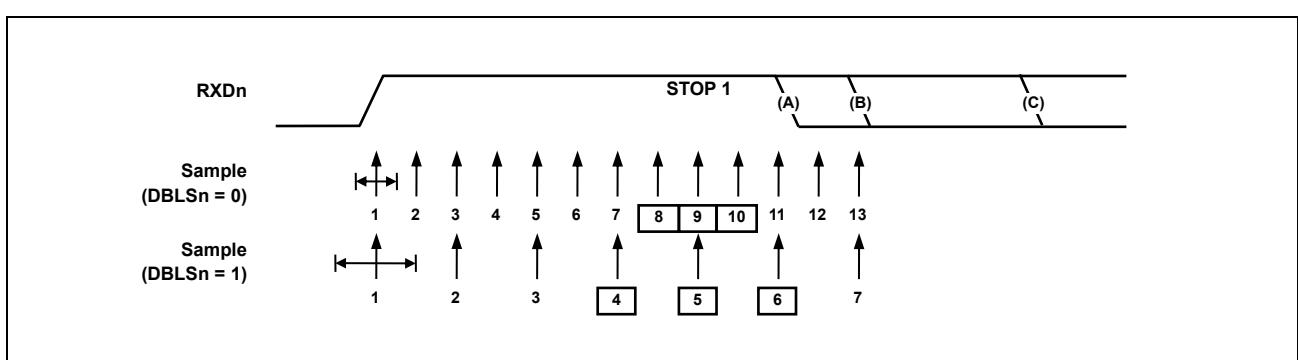
When the receiver is enabled ( $\text{RXEn}=1$ ), the clock recovery logic tries to find a high-to-low transition on the  $\text{RXDn}$  line, the start bit condition. After detecting high to low transition on  $\text{RXDn}$  line, the clock recovery logic uses samples 8, 9 and 10 for normal mode to 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 same 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 83. Asynchronous Sampling of Data and Parity Bit (USIn)**

The process for detecting stop bit is same as 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  $\text{RXDn}$  line to check a valid high to low transition is detected (start bit detection).



**Figure 84. Stop Bit Sampling and Next Start Bit Sampling (USIn)**

### 13.11 USIn SPI Mode

The USIn can be set to operate in industrial standard SPI compliant mode. The SPI mode has the following features.

11. Full Duplex, Three-wire synchronous data transfer
12. Master and Slave Operation
13. Supports all four SPI modes of operation (mode 0, 1, 2, and 3)
14. Selectable LSB first or MSB first data transfer
15. Double buffered transmit and receive
16. 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 MISO and TXDn is renamed as MOSI for compatibility to other SPI devices.

### 13.12 USIn SPI Clock Formats and Timing

To accommodate a wide variety of synchronous serial peripherals from different manufacturers, the USIn 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 inserts 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 USIn.

Table below shows four combinations of CPOLn and CPHAn for SPI mode 0, 1, 2, and 3.

**Table 30. CPOLn Functionality**

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)

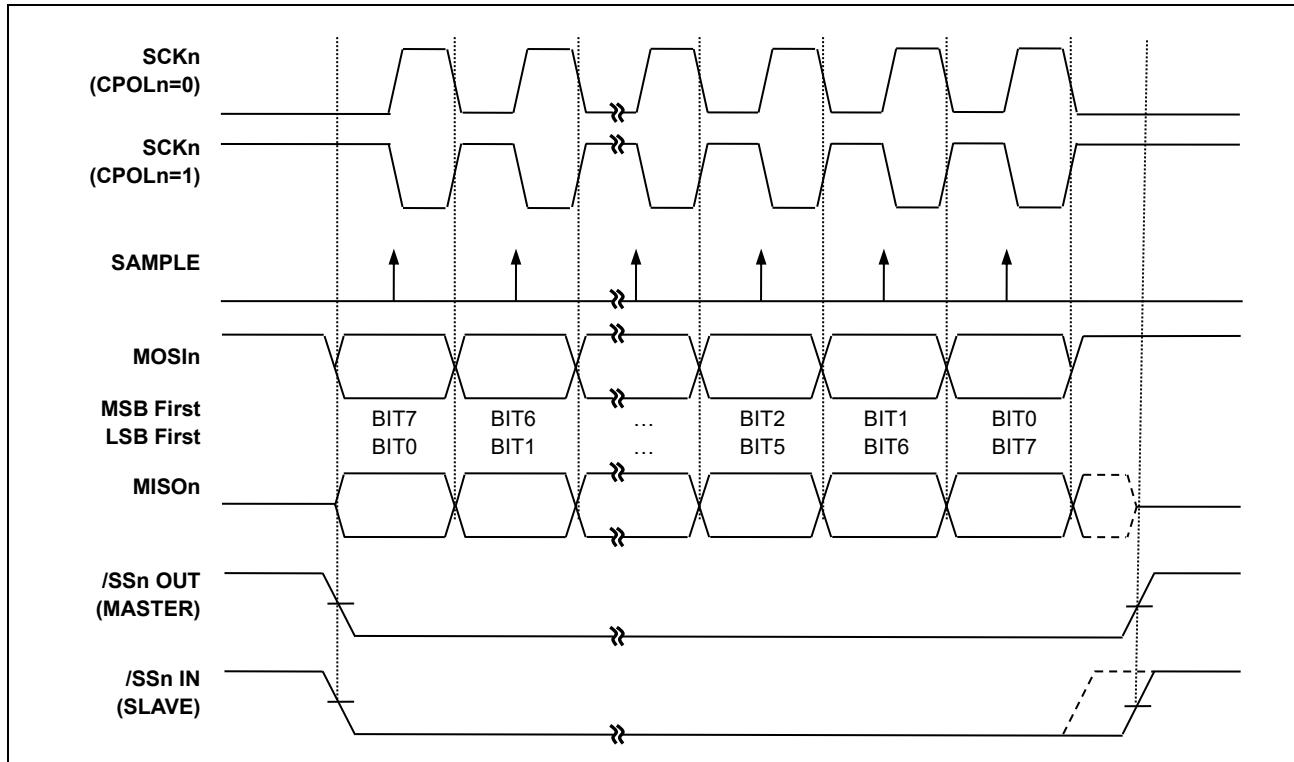
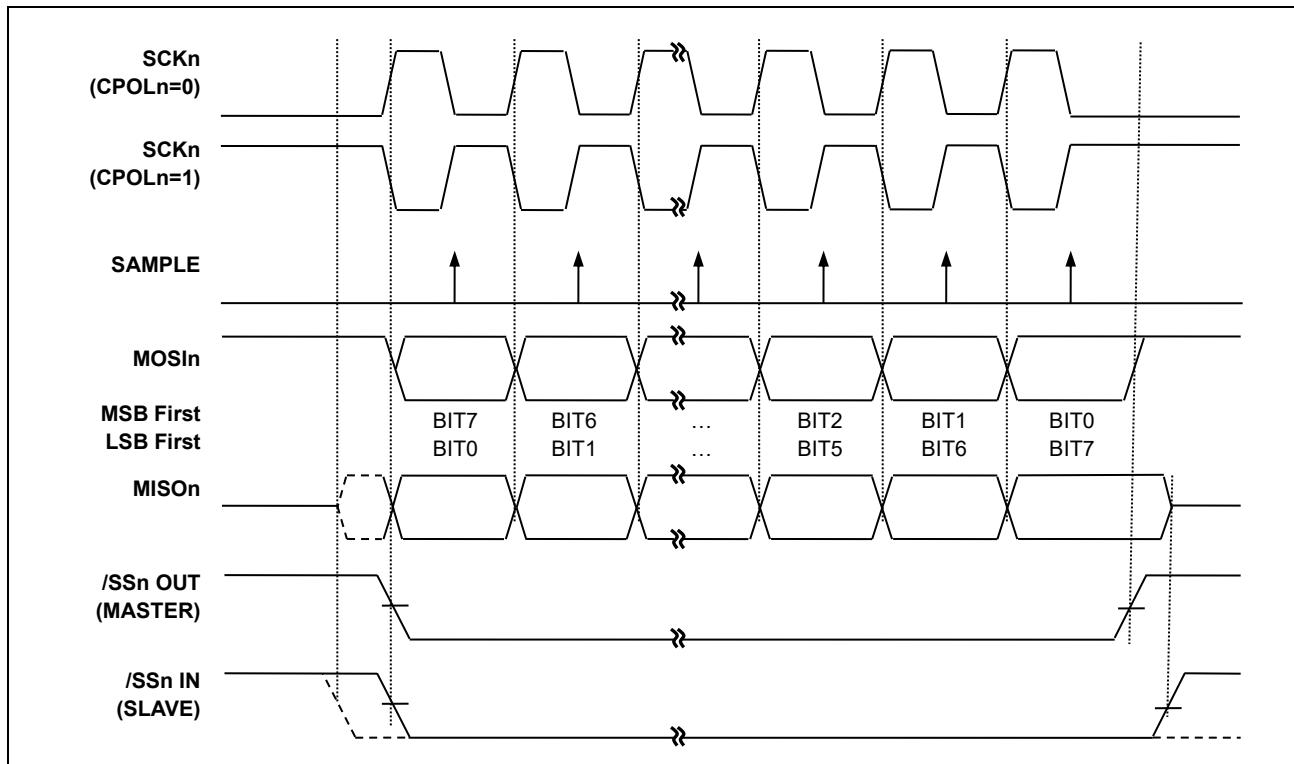


Figure 85. USIn SPI Clock Formats when  $\text{CPHAn}=0$

When  $\text{CPHAn}=0$ , the slave begins to drive its  $\text{MISON}$  output with the first data bit value when  $\text{SSn}$  goes to active low. The first  $\text{SCKn}$  edge causes both the master and the slave to sample the data bit value on their  $\text{MISON}$  and  $\text{MOSIn}$  inputs, respectively. At the second  $\text{SCKn}$  edge, the USIn shifts the second data bit value out to the  $\text{MOSIn}$  and  $\text{MISON}$  outputs of the master and slave, respectively. Unlike the case of  $\text{CPHAn}=1$ , when  $\text{CPHAn}=0$ , the slave's  $\text{SSn}$  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  $\text{SSn}$  input.



**Figure 86. USIn SPI Clock Formats when CPHAn=1**

When CPHAn=1, the slave begins to drive its MISON output when SS<sub>n</sub> goes active low, but the data is not defined until the first SCK<sub>n</sub> edge. The first SCK<sub>n</sub> 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 SCK<sub>n</sub> edge causes both the master and slave to sample the data bit value on their MISON and MOSIn inputs, respectively. At the third SCK<sub>n</sub> edge, the USIn 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 SS<sub>n</sub> input is not required to go to its inactive high level between transfers.

Because the SPI logic reuses the USIn resources, SPI mode of operation is similar to that of synchronous or asynchronous operation. An SPI transfer is initiated by checking for the USIn 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 SCK<sub>n</sub> is generated from transmitter block.

### 13.13 USIn SPI Block Diagram

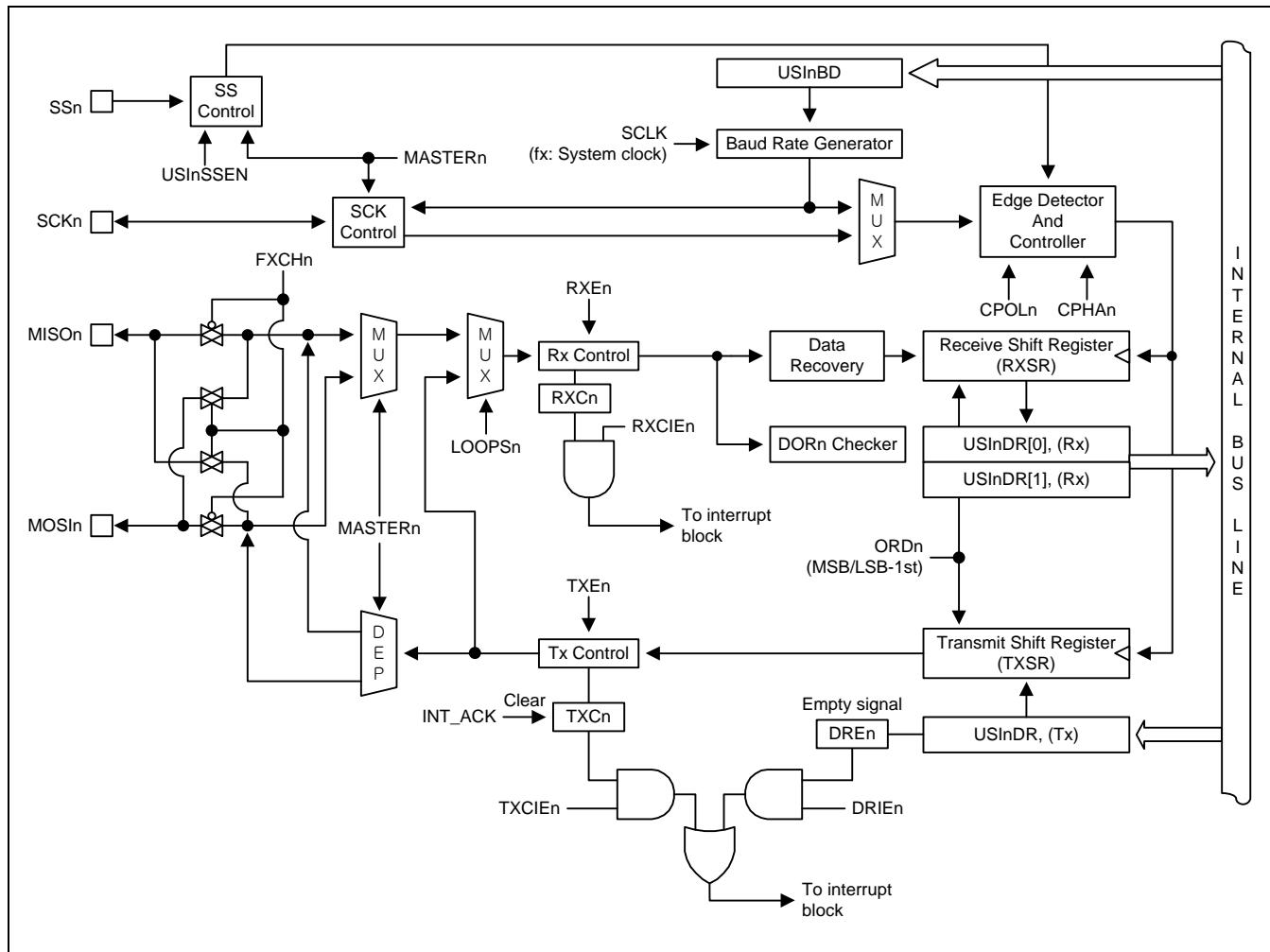


Figure 87. USIn SPI Block Diagram(n = 0, 1)

## 13.14 USIn I2C Mode

The USIn can be set to operate in industrial standard serial communication 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. A96T418 supports only one I2C, so n=0. 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
- I2C Slave mode does not support repeated START bit mode

## 13.15 USIn 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.

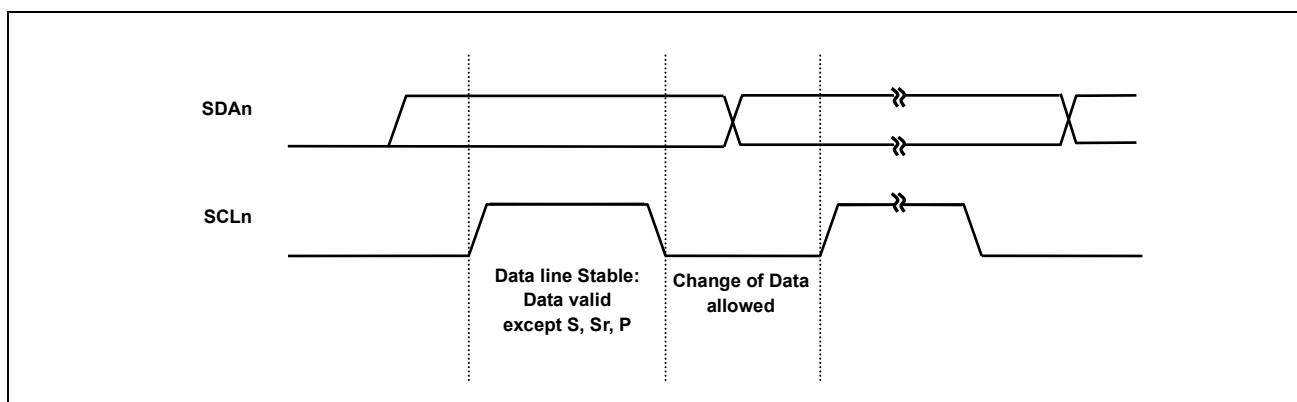


Figure 88. Bit Transfer on the I2C-Bus (USIn)

### 13.16 USIn I2C Start / Repeated Start / Stop

One master can issue a START (S) condition to notice other devices connected to the SCL<sub>n</sub>, 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 SCL<sub>n</sub> is high defines a START (S) condition.

A low to high transition on the SDAn line while SCL<sub>n</sub> 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.

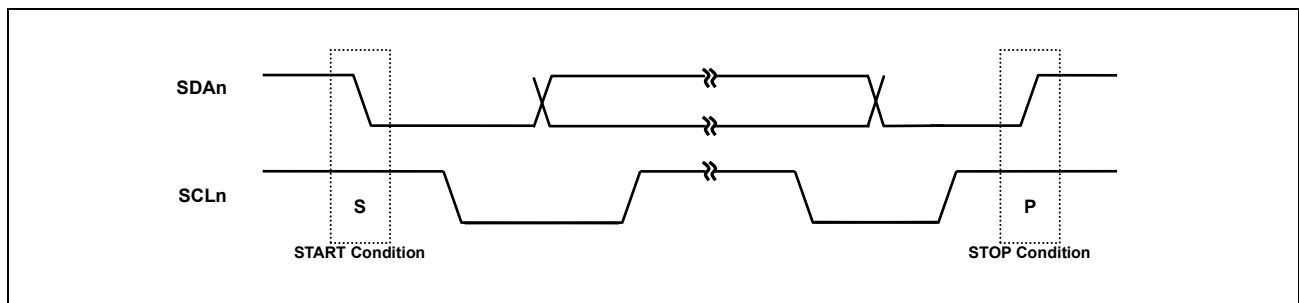


Figure 89. START and STOP Condition (USIn)

### 13.17 USIn 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 SCL<sub>n</sub> 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 SCL<sub>n</sub>.

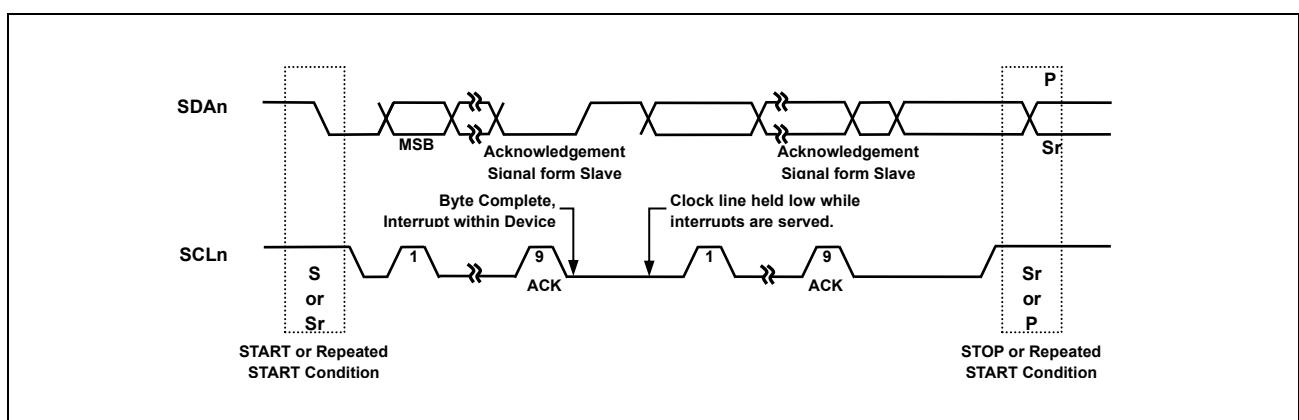


Figure 90. Data Transfer on the I2C-Bus (USIn)

### 13.18 USIn 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.

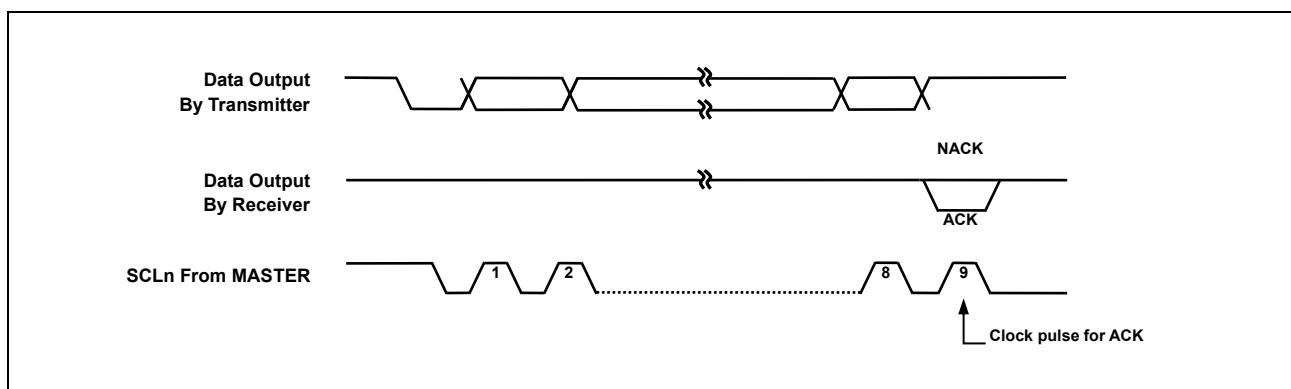


Figure 91. Acknowledge on the I2C-Bus (USIn)

### 13.19 USIn 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.

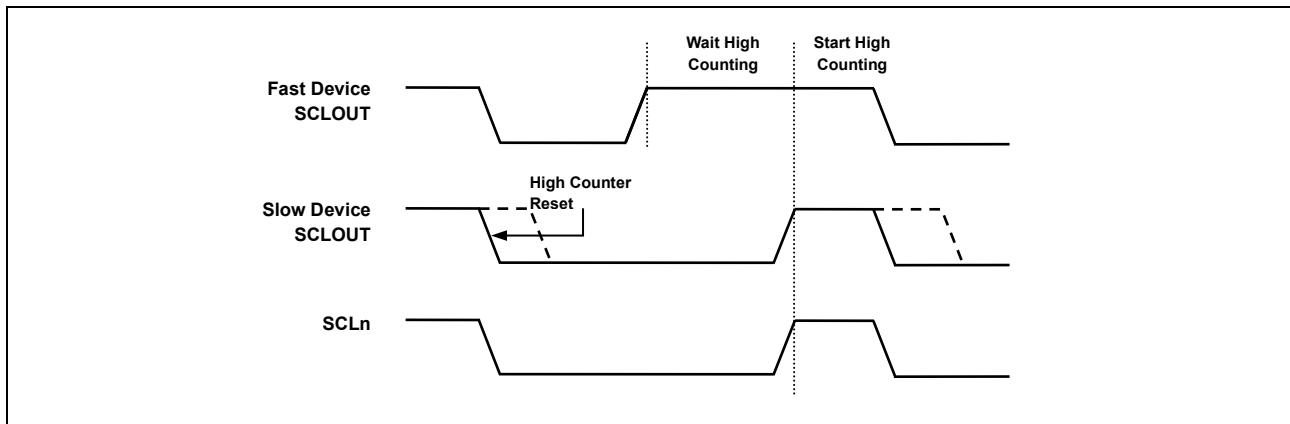


Figure 92. Clock Synchronization during Arbitration Procedure (USIn)

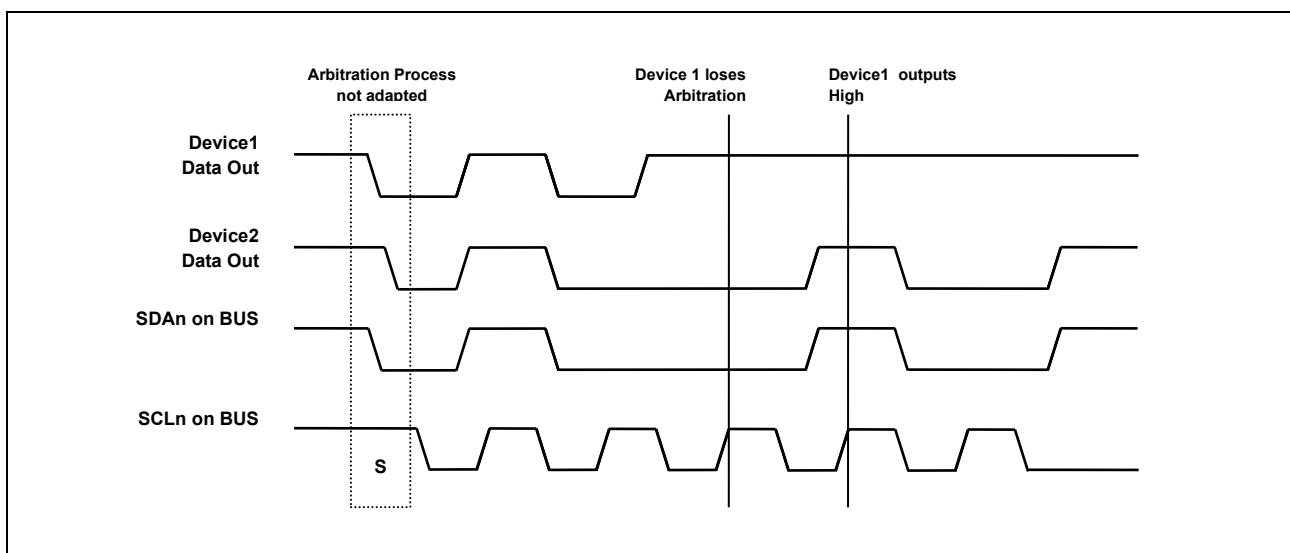


Figure 93. Arbitration Procedure of Two Masters (USIn)

### 13.20 USIn 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.

### 13.20.1 USIn I2C Master Transmitter

To operate I2C in master transmitter, follow the recommended steps below.

Enable I2C by setting USInMS[1:0]bits in USInCR1 and USInEN bit in USInCR2. This provides main clock to the peripheral.

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.

Configure baud rate by writing desired value to both USInSCLR and USInSCHR for the Low and High period of SCLn line.

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.

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.

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<sup>th</sup> 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 stops 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 stops 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.

1-Byte of data is being transmitted. During data transfer, bus arbitration continues.

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 stops 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.

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.

### 13.20.2 USIn I2C Master Receiver

To operate I2C in master receiver, follow the recommended steps below.

Enable I2C by setting USInMS[1:0]bits in USInCR1 and USInEN bit in USInCR2. This provides main clock to the peripheral.

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.

Configure baud rate by writing desired value to both USInSCLR and USInSCHR for the Low and High period of SCLn line.

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.

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.

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<sup>th</sup> 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 stops 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 stops 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.

1-Byte of data is being received.

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.

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.

### 13.20.3 USIn I2C Slave Transmitter

To operate I2C in slave transmitter, follow the recommended steps below.

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.

Enable I2C by setting USInMS[1:0]bits in USInCR1, IICnIEbit in USInCR4 and USInEN bit in USInCR2. This provides main clock to the peripheral.

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.

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.

1-Byte of data is being transmitted.

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 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.

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.

### 13.20.4 USIn I2C Slave Receiver

To operate I2C in slave receiver, follow the recommended steps below.

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.

Enable I2C by setting USInMS[1:0] bits in USInCR1, IICnIEbit in USInCR4 and USInEN bit in USInCR2. This provides main clock to the peripheral.

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, I2Cn compares the received data with value 0x00, the general call address.

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.

Byte of data is being received.

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 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.

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.

### 13.21 USIn I2C Block Diagram

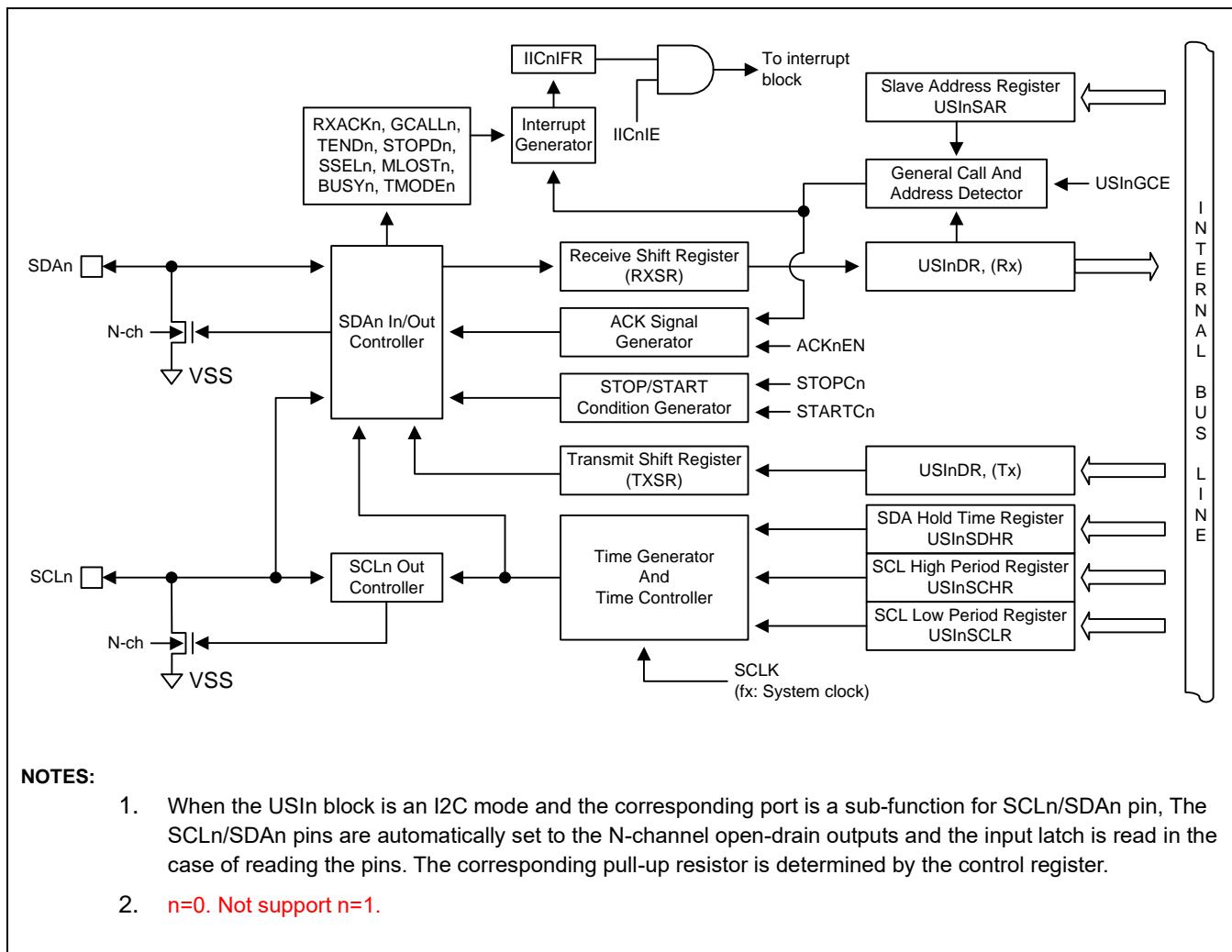


Figure 94. USIn I2C Block Diagram

## 13.22 Register Map

**Table 31. USI Register Map**

Name	Address	Direction	Default	Description
USI0BD	E3H	R/W	FFH	USI0 Baud Rate Generation Register
USI0DR	E5H	R/W	00H	USI0 Data Register
USI0SDHR	E4H	R/W	01H	USI0 SDA Hold Time Register
USI0SCHR	E7H	R/W	3FH	USI0 SCL High Period Register
USI0SCLR	E6H	R/W	3FH	USI0 SCL Low Period Register
USI0SAR	DDH	R/W	00H	USI0 Slave Address Register
USI0CR1	D9H	R/W	00H	USI0 Control Register 1
USI0CR2	DAH	R/W	00H	USI0 Control Register 2
USI0CR3	DBH	R/W	00H	USI0 Control Register 3
USI0CR4	DCH	R/W	00H	USI0 Control Register 4
USI0ST1	E1H	R/W	80H	USI0 Status Register 1
USI0ST2	E2H	R	00H	USI0 Status Register 2

## 13.23 USIn Register Description

USIn module consists of USIn baud rate generation register (USInBD), USIn data register (USInDR), USIn SDA hold time register (USInSDHR), USIn SCL high period register (USInSCHR), USIn SCL low period Register (USInSCLR), USIn slave address register (USInSAR), USIn control register 1/2/3/4 (USInCR1/2/3/4), and USIn status register 1/2 (USInST1/2).

## 13.24 Register Description for USIn

### USInBD (USIn Baud- Rate Generation Register: For UART and SPI mode): E3H, n = 0

7	6	5	4	3	2	1	0
USInBD7	USInBD6	USInBD5	USInBD4	USInBD3	USInBD2	USInBD1	USInBD0
RW							

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.

#### NOTES:

1. In common with USInSAR register, USInBD register is used for slave address register when the USIn I2C mode.

### USInDR (USIn Data Register: For UART, SPI, and I2C mode): E5H, n = 0

7	6	5	4	3	2	1	0
USInDR7	USInDR6	USInDR5	USInDR4	USInDR3	USInDR2	USInDR1	USInDR0
RW							

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 (USInSDA Hold Time Register: For I2C mode): E4H, n = 0**

7	6	5	4	3	2	1	0
USInSDHR7	USInSDHR6	USInSDHR5	USInSDHR4	USInSDHR3	USInSDHR2	USInSDHR1	USInSDHR0
RW	RW	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 SCI in I2C mode.

**NOTES:**

- That SDAn is changed after  $t_{SCLK} X$  (USInSDHR+2), in master SDAn change in the middle of SCLn.
- In slave mode, configure this register regarding the frequency of SCLn from master.
- The SDAn is changed after  $t_{SCLK} X$  (USInSDHR+2) in master mode. So, to insure operation in slave mode, the value
- $t_{SCLK} X$  (USInSDHR+2) must be smaller than the period of SCL.

**USInSCHR (USInSCL High Period Register: For I2C mode): E7H, n = 0**

7	6	5	4	3	2	1	0
USInSCHR7	USInSCHR6	USInSCHR5	USInSCHR4	USInSCHR3	USInSCHR2	USInSCHR1	USInSCHR0
RW	RW	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:  $t_{SCLK} X (4 \times USInSCLR + 2)$  where  $t_{SCLK}$  is the period of SCLK.

So, the operating frequency of I2C master mode is calculated by the following equation.

$$f_{I2C} = \frac{1}{t_{SCLK} X (4 \times (USInSCLR + USInSCHR) + 4)}$$

**USInSCLR (USInSCL Low Period Register: For I2C mode): E6H, n = 0**

7	6	5	4	3	2	1	0
USInSCLR7	USInSCLR6	USInSCLR5	USInSCLR4	USInSCLR3	USInSCLR2	USInSCLR1	USInSCLR0
RW							

Initial value: 3FH

**USInSCLR[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:  $t_{SCLK} \times (4 \times USInSCLR + 2)$  where  $t_{SCLK}$  is the period of SCLK.

**USInSAR (USIn Slave Address Register: For I2C mode): DDH, n = 0**

7	6	5	4	3	2	1	0
USInSLA6	USInSLA5	USInSLA4	USInSLA3	USInSLA2	USInSLA1	USInSLA0	USInGCE
RW	RW						

Initial value: 00H

**USInSLA[6:0]** These bits configure the slave address of I2C when it operates 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 (USIn Control Register 1: For UART, SPI, and I2C mode): D9H, n = 0**

7	6	5	4	3	2	1	0
USInMS1	USInMS0	USInPM1	USInPM0	USInS2	USInS1 ORDn	USInS0 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				
	1	0	I2C mode (if n=1, I2C mode is not supported)				
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				
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			
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' (only SPI mode)						
	0	LSB-first					
	1	MSB-first					
	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 SCKn (only SPI mode).						
CPHAn	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)			
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					

**USInCR2 (USIn Control Register 2: For UART, SPI, and I2C mode): DAH, n = 0**

7	6	5	4	3	2	1	0
DRIEn	TXCIEn	RXCIEn	WAKEIEn	TXEn	RXEn	USInEN	DBLSn
RW	RW	RW	RW	RW	RW	RW	RW

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 (USIn Control Register 3: For UART, SPI, and I2C mode): DBH, n = 0**

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 SCKn). 1 Master mode operation (Internal clock for SCKn).
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
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 <sup>th</sup> bit) to be transmitted is '0' 1 MSB (9 <sup>th</sup> 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 <sup>th</sup> bit) received is '0' 1 MSB (9 <sup>th</sup> bit) received is '1'

**USInCR4 (USIn Control Register 4: For I2C mode): DCH, n = 0**

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 the USInST2. Writing "1" has no effect.
	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 SCLn period. 0 No ACK signal is generated (SDAn =1) 1 ACK signal is generated (SDAn =0) <b>NOTES: ACK signal is output (SDA =0) for the following 3 cases.</b> 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 (USIn Status Register 1: For UART and SPI mode): E1H, n = 0**

7	6	5	4	3	2	1	0
DREn	TXCn	RXCn	WAKEn	USInRST	DORn	FEn	PEn
RW	RW	R	RW	RW	R	RW	RW

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 an 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 (USIn Status Register 2: For I2C mode): E2H, n = 0**

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<sup>(NOTES:</sup>** 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 indicate general call.

0 General call address is not detected (Slave mode)

1 General call address is detected (Slave mode)

**TENDn<sup>(NOTES:</sup>** 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<sup>(NOTES:</sup>** This bit is set when a STOP condition is detected.

0 No STOP condition is detected

1 STOP condition is detected

**SSELn<sup>(NOTES:</sup>** 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<sup>(NOTES:</sup>** This bit represents the result of bus arbitration in master mode.

0 I2C maintains bus mastership

1 I2C has lost 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

**NOTES:**

1. These bits can be source of interrupt.

2. When an I2C interrupt occurs except for STOP mode, the SCLn line is hold LOW. To release SCLn, write arbitrary value to USInST2. When USInST2 is written, the TENDn, STOPDn, SSELn, MLOSTn, and RXACKn bits are cleared.

### 13.25 Baud Rate setting (example)

**Table 32. Examples of USI0BD Settings for Commonly Used Oscillator Frequencies**

Baud Rate (bps)	fx=1.00MHz		fx=1.8432MHz		fx=2.00MHz	
	USI0BD	ERROR	USI0BD	ERROR	USI0BD	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	-	-	-	-	-	-

**Table 33. Examples of USI0BD Settings for Commonly Used Oscillator Frequencies (continued)**

Baud Rate (bps)	fx=1.00MHz		fx=1.8432MHz		fx=2.00MHz	
	USI0BD	ERROR	USI0BD	ERROR	USI0BD	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	-	-	-	-	-	-

**Table 33. Examples of USI0BD Settings for Commonly Used Oscillator Frequencies (continued)**

<b>Baud Rate (bps)</b>	<b>fx=8.00MHz</b>		<b>fx=11.0592MHz</b>	
	<b>USI0BD</b>	<b>ERROR</b>	<b>USI0BD</b>	<b>ERROR</b>
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	-	-	-	-

## 14 USART1

### 14.1 Overview

The Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART1) 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
- Master or Slave Clocked Synchronous and SPI Operation
- Supports all four SPI Modes of Operation (Mode 0, 1, 2, 3)
- LSB First or MSB First Data Transfer @SPI mode
- High Resolution 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
- Digital Low Pass Filter
- Three Separate Interrupts on TX Complete, TX Data Register Empty and RX Complete
- Double Speed Asynchronous Communication Mode

USART1 has three main parts of Clock Generator, Transmitter and Receiver. The Clock Generation logic consists of synchronization logic for external clock input 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 a continuous transfer of data without any delay between frames. The receiver is the most complex part of the USART1 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 (UDATA) and control logic. The Receiver supports the same frame formats as the Transmitter and can detect Frame Error, Data OverRun and Parity Errors.

## 14.2 Block Diagram

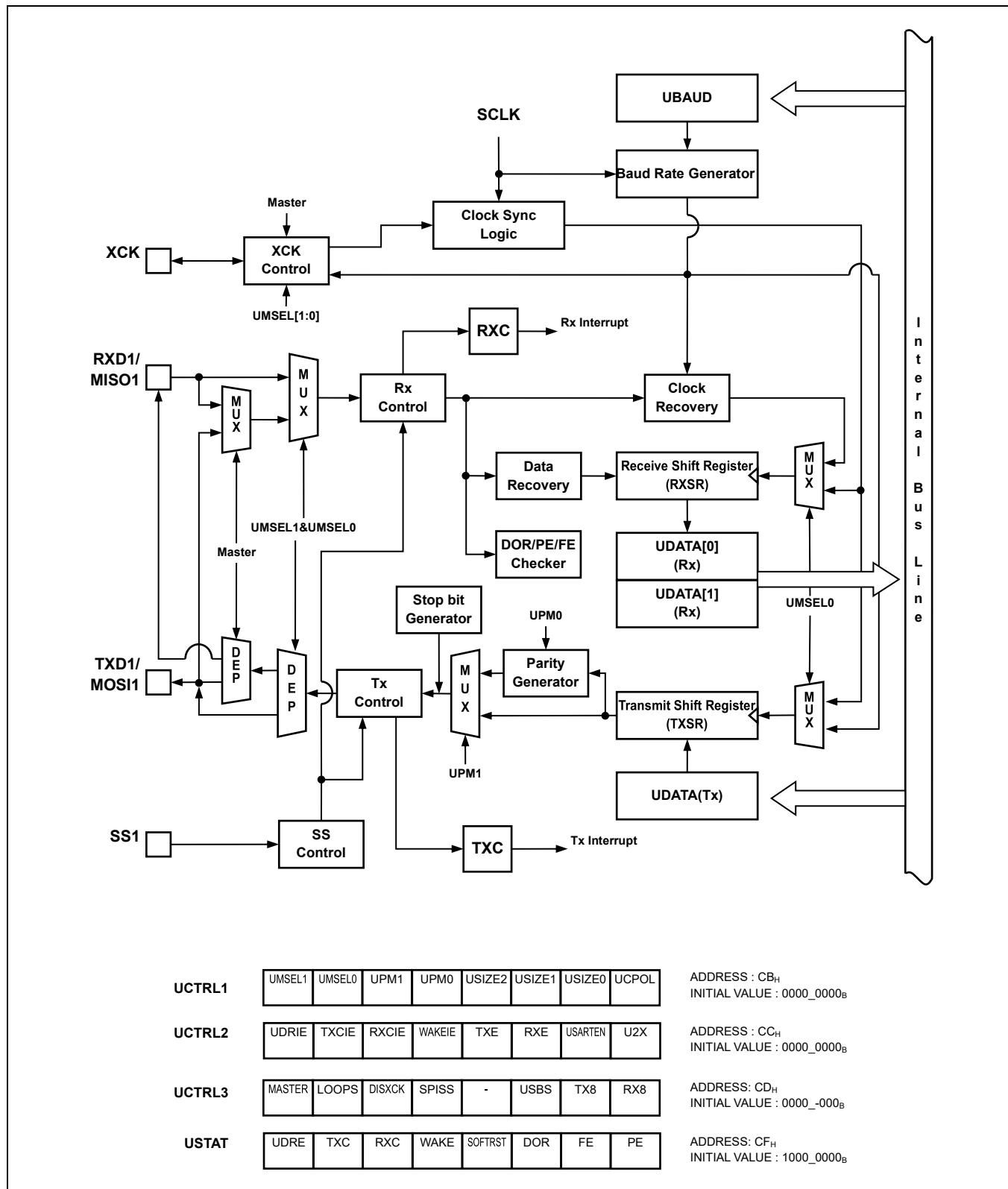


Figure 95. USART1 Block Diagram

### 14.3 Clock Generation

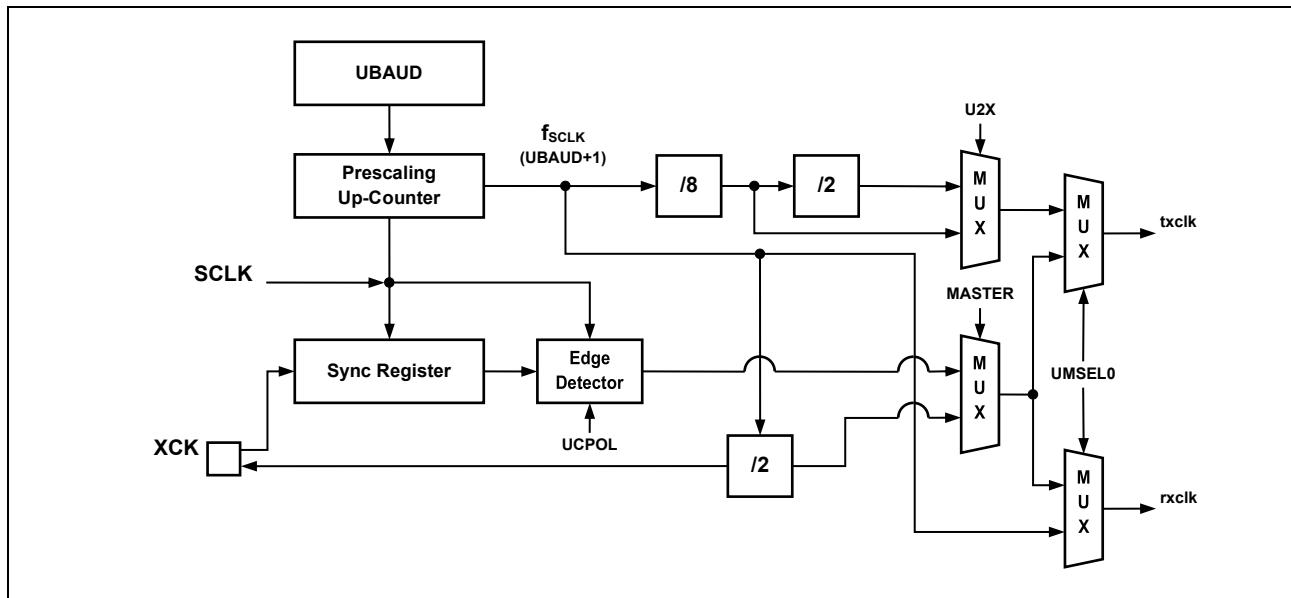


Figure 96. Clock Generation Block Diagram

The Clock generation logic generates the base clock for the Transmitter and Receiver. The USART1 supports four modes of clock operation and those are Normal Asynchronous, Double Speed Asynchronous, Master Synchronous and Slave Synchronous. The clock generation scheme for Master SPI and Slave SPI mode is the same as Master Synchronous and Slave Synchronous operation mode. The UMSEL<sub>n</sub> bit in UCTRL1 register selects between asynchronous and synchronous operation. Asynchronous Double Speed mode is controlled by the U2X bit in the UCTRL2 register. The MASTER bit in UCTRL2 register controls whether the clock source is internal (Master mode, output port) or external (Slave mode, input port). The XCK pin is only active when the USART1 operates in Synchronous or SPI mode.

Table below contains equations for calculating the baud rate (in bps).

Table 33. Equations for Calculating Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate
Asynchronous Normal Mode (U2X=0)	Baud Rate = $\frac{f_{SCLK}}{16(UBAUDx + 1)}$
Asynchronous Double Speed Mode (U2X=1)	Baud Rate = $\frac{f_{SCLK}}{8(UBAUDx + 1)}$
Synchronous or SPI Master Mode	Baud Rate = $\frac{f_{SCLK}}{2(UBAUDx + 1)}$

## 14.4 External Clock (XCK)

External clocking is used by the synchronous or SPI slave modes of operation.

External clock input from the XCK pin is sampled by a synchronization logic to remove meta-stability. The output from the synchronization logic must then pass through an edge detector before it can be used by the Transmitter and Receiver. This process introduces a two CPU clock period delay and the maximum frequency of the external XCK pin is limited by the following equation.

$$f_{XCK} = \frac{f_{SCLK}}{4}$$

Where  $f_{XCK}$  is the frequency of XCK and  $f_{SCLK}$  is the frequency of main system clock (SCLK).

## 14.5 Synchronous mode Operation

When synchronous or SPI mode is used, the XCK pin will be used as either clock input (slave) or clock output (master). The dependency between the clock edges and data sampling or data change is the same. The basic principle is that data input on RXD1 (MISO1 in SPI mode) pin is sampled at the opposite XCK clock edge of the edge in the data output on TXD1 (MOSI1 in SPI mode) pin is changed.

The UCPO1 bit in UCTRL1 register selects which XCK clock edge is used for data sampling and which is used for data change. As shown in the figure below, when UCPO1 is zero the data will be changed at XCK rising edge and sampled at XCK falling edge.

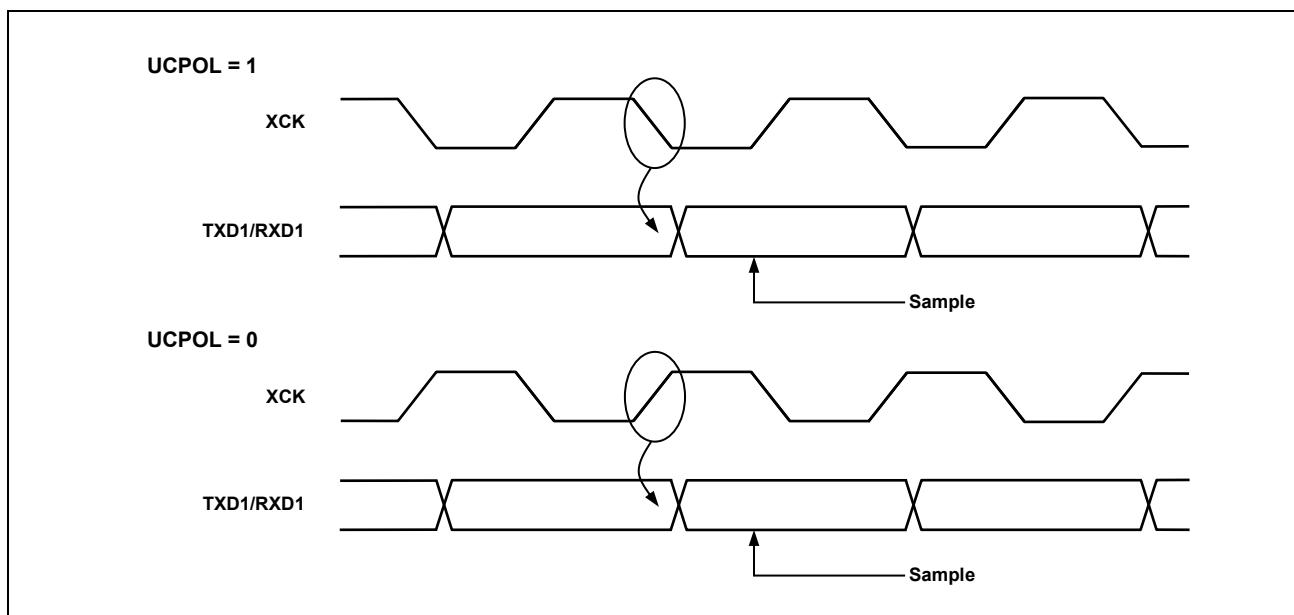


Figure 97. Synchronous Mode XCK Timing

## 14.6 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 checking.

The USART1 supports all 30 combinations of the following as valid frame formats.

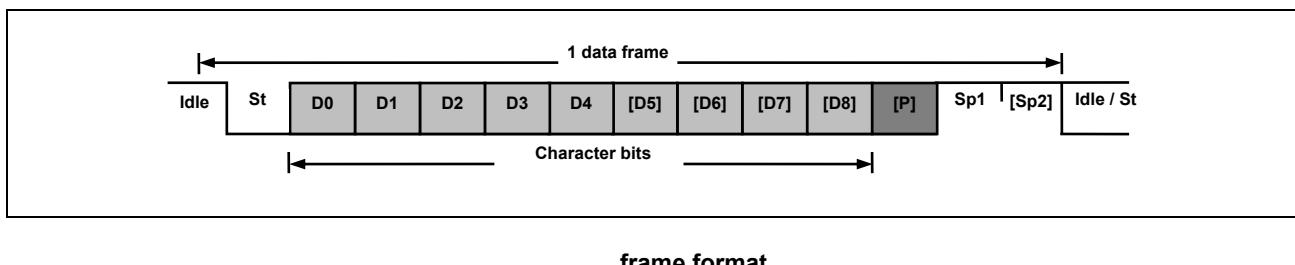
1 start bit

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 a total of nine, are succeeding, ending with the most significant bit (MSB). If enabled the parity bit is inserted after the data bits, before the stop bits. 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 next figure shows the possible combinations of the frame formats. Bits inside brackets are optional.



1 data frame consists of the following bits

Idle            No communication on communication line (Tx D1/Rx D1)

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 USART1 is set by the USIZE[2:0], UPM[1:0] and USBS bits in UCTRL1 register. The Transmitter and Receiver use the same setting.

## 14.7 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 St + bits and first stop bit of a serial frame.

$$P_{even} = D_{n-1} \wedge \dots \wedge D_3 \wedge D_2 \wedge D_1 \wedge D_0 \wedge 0$$

$$P_{odd} = D_{n-1} \wedge \dots \wedge D_3 \wedge D_2 \wedge D_1 \wedge D_0 \wedge 1$$

$P_{even}$  : Parity bit using even parity

$P_{odd}$  : Parity bit using odd parity

$D_n$  : Data bit n of the character

## 14.8 USART1 Transmitter

The USART1 Transmitter is enabled by setting the TXE bit in UCTRL1 register. When the Transmitter is enabled, the normal port operation of the TXD1 pin is overridden by the serial output pin of USART1. The baud-rate, operation mode and frame format must be setup once before doing any transmissions. If synchronous or SPI operation is used, the clock on the XCK pin will be overridden and used as transmission clock. If USART1 operates in SPI mode, SS1 pin is used as SS1 input pin in slave mode or can be configured as SS1 output pin in master mode. This can be done by setting SPISS bit in UCTRL3 register.

### 14.8.1 Sending Tx data

A data transmission is initiated by loading the transmit buffer (UDATA 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 at the settings of control registers. If the 9-bit characters are used in asynchronous or synchronous operation mode (USIZE[2:0]=7), the ninth bit must be written to the TX8 bit in UCTRL3 register before loading transmit buffer (UDATA register).

### 14.8.2 Transmitter flag and interrupt

The USART1 Transmitter has 2 flags which indicate its state. One is USART1 Data Register Empty (UDRE) and the other is Transmit Complete (TXC). Both flags can be used as interrupt sources.

UDRE flag indicates whether the transmit buffer is ready to be loaded with new data. This bit is set when the transmit buffer is empty and cleared when the transmit buffer contains transmission data which 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 not valid.

When the Data Register Empty Interrupt Enable (UDRIE) bit in UCTRL2 register is set and the Global Interrupt is enabled, USART1 Data Register Empty Interrupt is generated while UDRE flag is set.

The Transmit Complete (TXC) flag bit is set when the entire frame in the transmit shift register has been shifted out and there are no more data in the transmit buffer. The TXC flag is automatically cleared when the Transmit Complete Interrupt service routine is executed, or it can be cleared by writing '0' to TXC bit in USTAT register.

When the Transmit Complete Interrupt Enable (TXCIE) bit in UCTRL2 register is set and the Global Interrupt is enabled, USART1 Transmit Complete Interrupt is generated while TXC flag is set.

### 14.8.3 Parity Generator

The Parity Generator calculates the parity bit for the sending serial frame data. When parity bit is enabled (UPM[1]=1), the transmitter control logic inserts the parity bit between the bits and the first stop bit of the sending frame.

#### 14.8.4 Disabling Transmitter

Disabling the Transmitter by clearing the TXE bit will not become effective until ongoing transmission is completed. When the Transmitter is disabled, the TxD1 pin is used as normal General Purpose I/O (GPIO) or primary function pin.

### 14.9 USART1 Receiver

The USART1 Receiver is enabled by setting the RXE bit in the UCTRL1 register. When the Receiver is enabled, the normal pin operation of the RxD1 pin is overridden by the USART1 as the serial input pin of the Receiver. The baud-rate, mode of operation and frame format must be set before serial reception. If synchronous or SPI operation is used, the clock on the XCK pin will be used as transfer clock. If USART1 operates in SPI mode, SS1 pin is used as SS1 input pin in slave mode or can be configured as SS1 output pin in master mode. This can be done by setting SPISS bit in UCTRL3 register.

#### 14.9.1 Receiving Rx data

When USART1 is in synchronous or asynchronous operation mode, the Receiver starts data reception when it detects a valid start bit (LOW) on RxD1 pin. Each bit after start bit is sampled at pre-defined baud-rate (asynchronous) or sampling edge of XCK (synchronous), and shifted into the receive shift register until the first stop bit of a frame is received. Even if there's 2nd stop bit in the frame, the 2nd 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 UDATA register.

If 9-bit characters are used (USIZE[2:0] = 7), the ninth bit is stored in the RX8 bit position in the UCTRL3 register. The 9th bit must be read from the RX8 bit before reading the low 8 bits from the UDATA register. Likewise, the error flags FE, DOR, PE must be read before reading the data from UDATA register. This is because the error flags are stored in the same FIFO position of the receive buffer.

#### 14.9.2 Receiver flag and interrupt

The USART1 Receiver has one flag that indicates the Receiver state.

The Receive Complete (RXC) flag indicates whether there are unread data present 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 (RXE=0), the receiver buffer is flushed and the RXC flag is cleared.

When the Receive Complete Interrupt Enable (RXCIE) bit in the UCTRL2 register is set and Global Interrupt is enabled, the USART1 Receiver Complete Interrupt is generated while RXC flag is set.

The USART1 Receiver has three error flags which are Frame Error (FE), Data OverRun (DOR) and Parity Error (PE). These error flags can be read from the USTAT register. As data received 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 UDATA register, read the USTAT register first which contains error flags.

The Frame Error (FE) flag indicates the state of the first stop bit. The FE flag is set when the stop bit was correctly detected as “1”, and the FE flag is cleared when the stop bit was incorrect, ie detected as “0”. This flag can be used for detecting out-of-sync conditions between data frames.

The Data OverRun (DOR) flag indicates data loss due to a receive buffer full condition. A DOR 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 DOR flag is set, all the incoming data are lost. To prevent data loss or clear this flag, read the receive buffer.

The Parity Error (PE) flag indicates that the frame in the receive buffer had a Parity Error when received. If Parity Check function is not enabled (UPM[1]=0), the PE bit is always read “0”.

**NOTES:** The error flags related to receive operation are not used when USART1 is in SPI mode.

#### 14.9.3 Parity Checker

If Parity bit is enabled (UPM[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.

#### 14.9.4 Disabling Receiver

In contrast to Transmitter, disabling the Receiver by clearing RXE bit makes the Receiver inactive immediately. When the Receiver is disabled the Receiver flushes the receive buffer and the remaining data in the buffer is all reset. The RXD1 pin is not overridden the function of USART1, so RXD1 pin becomes normal GPIO or primary function pin.

#### 14.9.5 Asynchronous Data Reception

To receive asynchronous data frame, the USART1 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 RXD1 pin.

The Data recovery logic samples incoming bits and low pass filters them, and this removes the noise of RXD1 pin.

The next figure illustrates the sampling process of the start bit of an incoming frame. The sampling rate is 16 times the baud-rate for normal mode, and 8 times the baud rate for Double Speed mode (U2X=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.

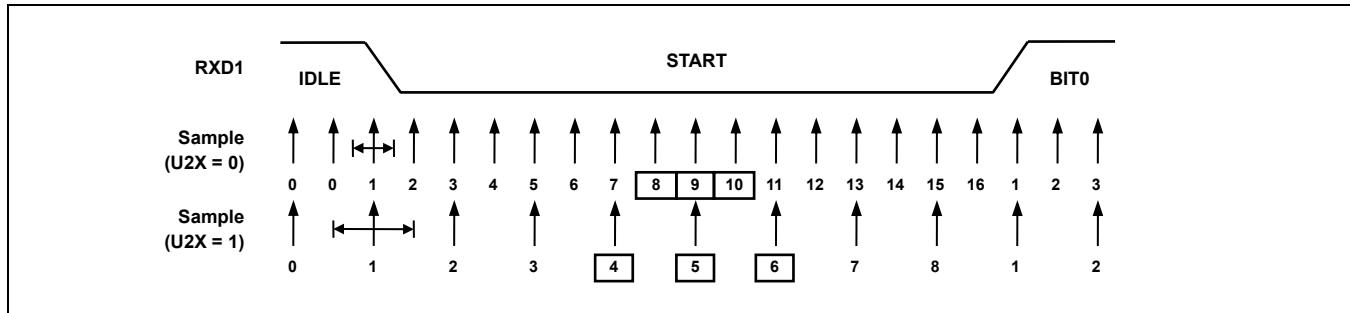


Figure 98. Start bit Sampling

When the Receiver is enabled ( $\text{RXE}=1$ ), the clock recovery logic tries to find a high to low transition on the RXD1 line, the start bit condition. After detecting high to low transition on RXD1 line, the clock recovery logic uses samples 8,9, and 10 for Normal mode, and samples 4, 5, and 6 for Double Speed 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 for Normal mode, samples 4, 5, and 6 for Double Speed mode. If more than 2 samples have low levels, the received bit is considered to a logic 0 and 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.

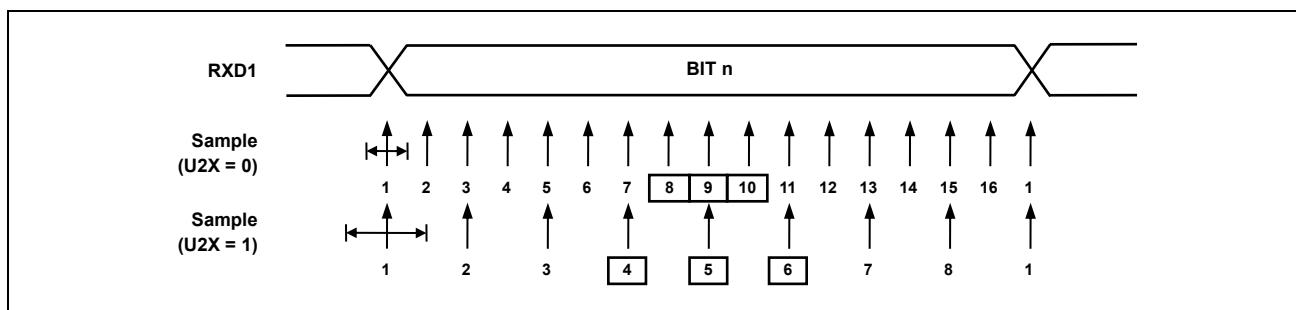


Figure 99. Sampling of Data and Parity bit

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 flag is set. After deciding first stop bit whether a valid stop bit is received or not, the Receiver enters into idle state and monitors the RXD1 line to check a valid high to low transition is detected (start bit detection).

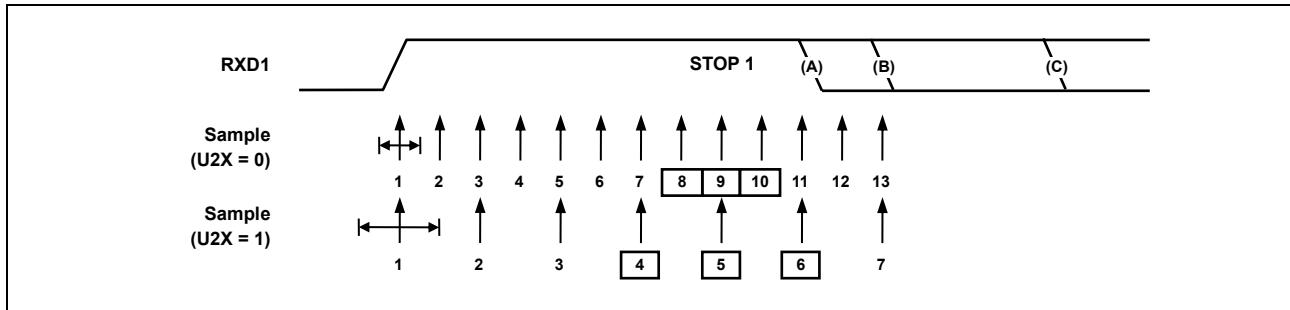


Figure 100. Stop bit Sampling and Next Start bit Sampling

## 14.10 SPI Mode

The USART1 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
- Master or Slave operation
- Supports all four SPI modes of operation (mode0, 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 (UMSEL[1:0]=3), the Slave Select (SS1) pin becomes active low input in slave mode operation, or can be output in master mode operation if SPISS bit is set.

Note that during SPI mode of operation, the pin RXD1 is renamed as MISO1 and TxD1 is renamed as MOSI1 for compatibility to other SPI devices.

### 14.10.1 SPI Clock formats and timing

To accommodate a wide variety of synchronous serial peripherals from different manufacturers, the USART1 has a clock polarity bit (UCPOL) and a clock phase control bit (UCPHA) to select one of four clock formats for data transfers. UCPOL selectively insert an inverter in series with the clock. UCPHA selects one of two different clock phase relationships between the clock and data. Note that UCPHA and UCPOL bits in UCTRL1 register have different meanings according to the UMSEL[1:0] bits which decides the operating mode of USART1.

Table below shows four combinations of UCPOL and UCPHA for SPI mode 0, 1, 2, and 3.

Table 34. CPOL Functionality

SPI Mode	UCPOL	UCPHA	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)

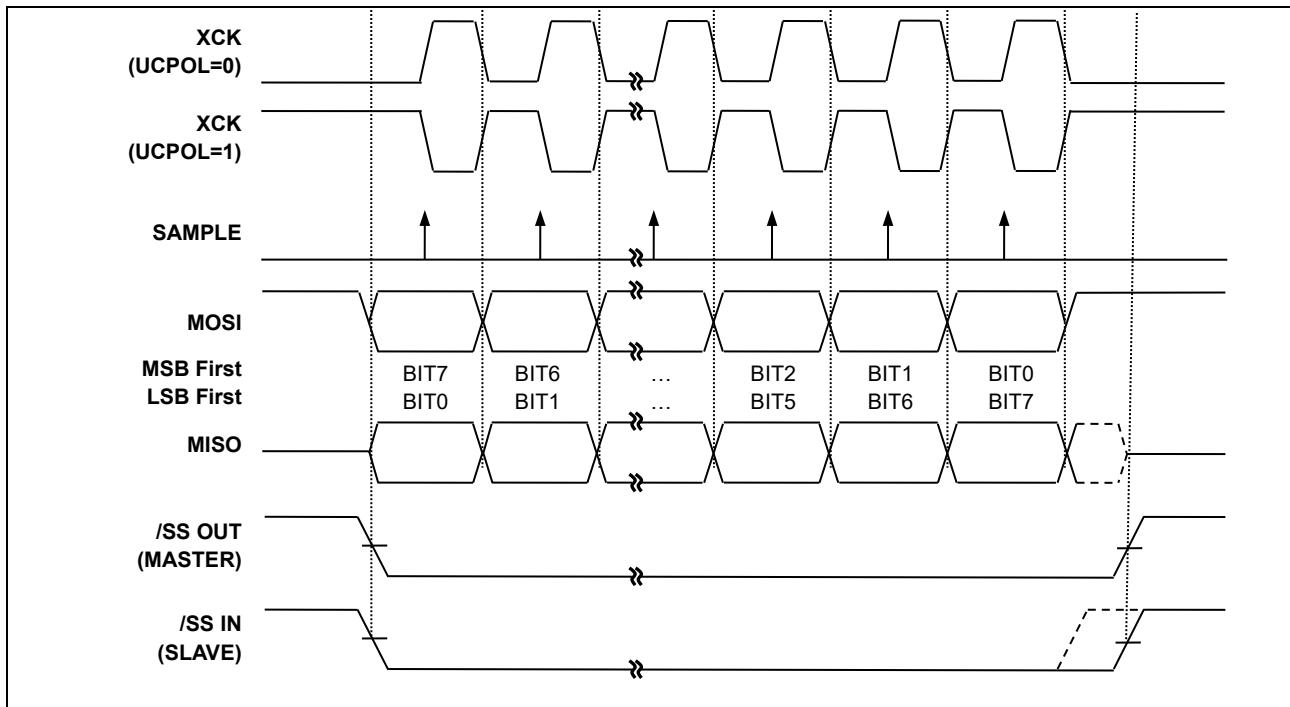
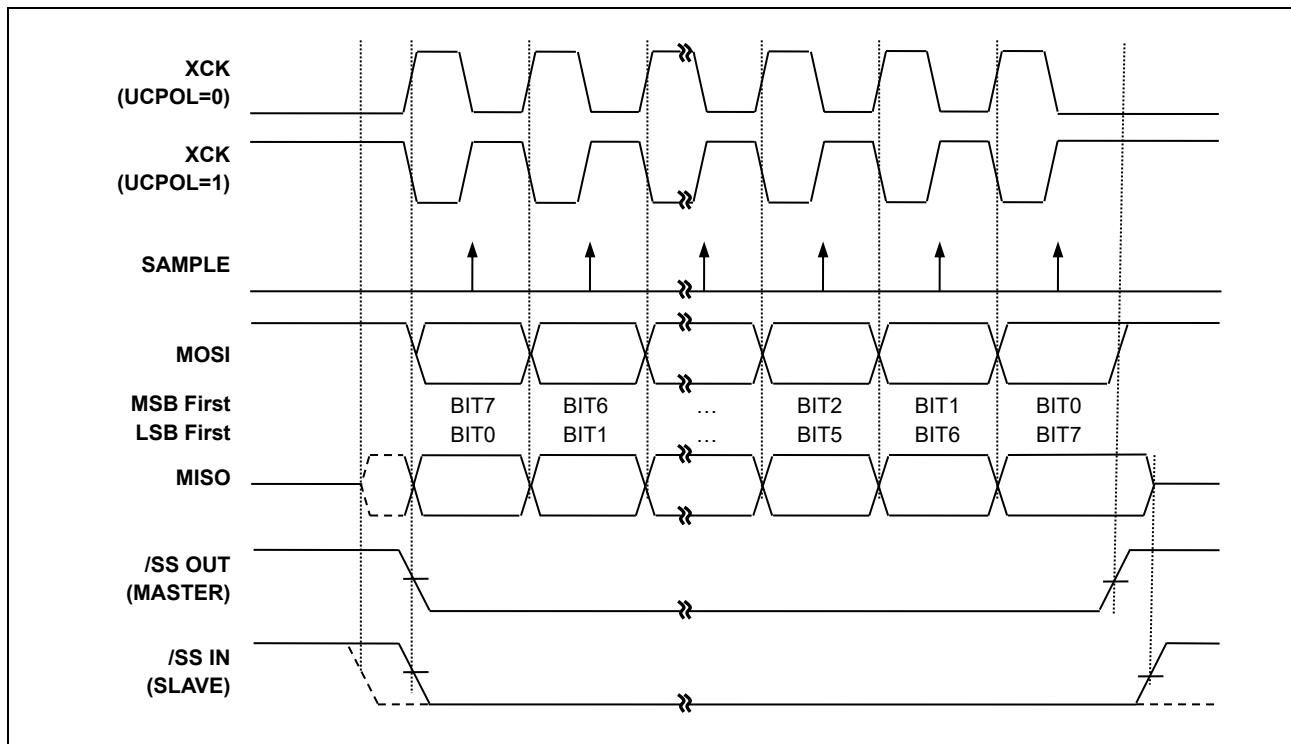


Figure 101. SPI Clock Formats when  $UCPHA=0$

When  $UCPHA=0$ , the slave begins to drive its MISO1 output with the first data bit value when SS goes to active low. The first XCK edge causes both the master and the slave to sample the data bit value on their MISO1 and MOSI1 inputs, respectively. At the second XCK edge, the USART1 shifts the second data bit value out to the MOSI1 and MISO1 outputs of the master and slave, respectively. Unlike the case of  $UCPHA=1$ , when  $UCPHA=0$ , the slave's SS1 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 SS1 input.



**Figure 102. SPI Clock Formats when UCPHA=1**

When UCPHA=1, the slave begins to drive its MISO1 output when SS1 goes active low, but the data is not defined until the first XCK edge. The first XCK edge shifts the first bit of data from the shifter onto the MOSI1 output of the master and the MISO1 output of the slave. The next XCK edge causes both the master and slave to sample the data bit value on their MISO1 and MOSI1 inputs, respectively. At the third XCK edge, the USART1 shifts the second data bit value out to the MOSI1 and MISO1 output of the master and slave respectively. When UCPHA=1, the slave's SS1 input is not required to go to its inactive high level between transfers.

Because the SPI logic reuses the USART1 resources, SPI mode of operation is similar to that of synchronous or asynchronous operation. An SPI transfer is initiated by checking for the USART1 Data Register Empty flag (UDRE=1) and then writing a byte of data to the UDATA Register. In master mode of operation, even if transmission is not enabled (TXE=0), writing data to the UDATA register is necessary because the clock XCK is generated from transmitter block.

## 14.11 Receiver Time Out (RTO)

This USART1 system supports the time out function. This function occurs when stop bit are not in RX line during URTOC setting value. RTO count stops in RXD signal live state and RTO clear and start is executed by stop bit recognition.

**Example1)** Condition : sysclk = 16MHz, Baud rate = 115,200 bps, Asynchronous Normal Mode (U2X = 0)

$$\text{Baud rate} = \text{sysclk} / 16 \times (\text{UBAUD} + 1)$$

$$\text{Calculated UBAUD} = (1000000 / \text{Target Baud rate}) - 1 = 7.68, \text{ Error rate} = 0.68 \Rightarrow \text{UBAUD} = 8$$

$$\text{Real baud rate at sysclk 16MHz} = 111,111 \text{ bps.}$$

1 bit time = 9ms

Maximum count time = 9ms \* 65536(16bit count) = 589.8ms

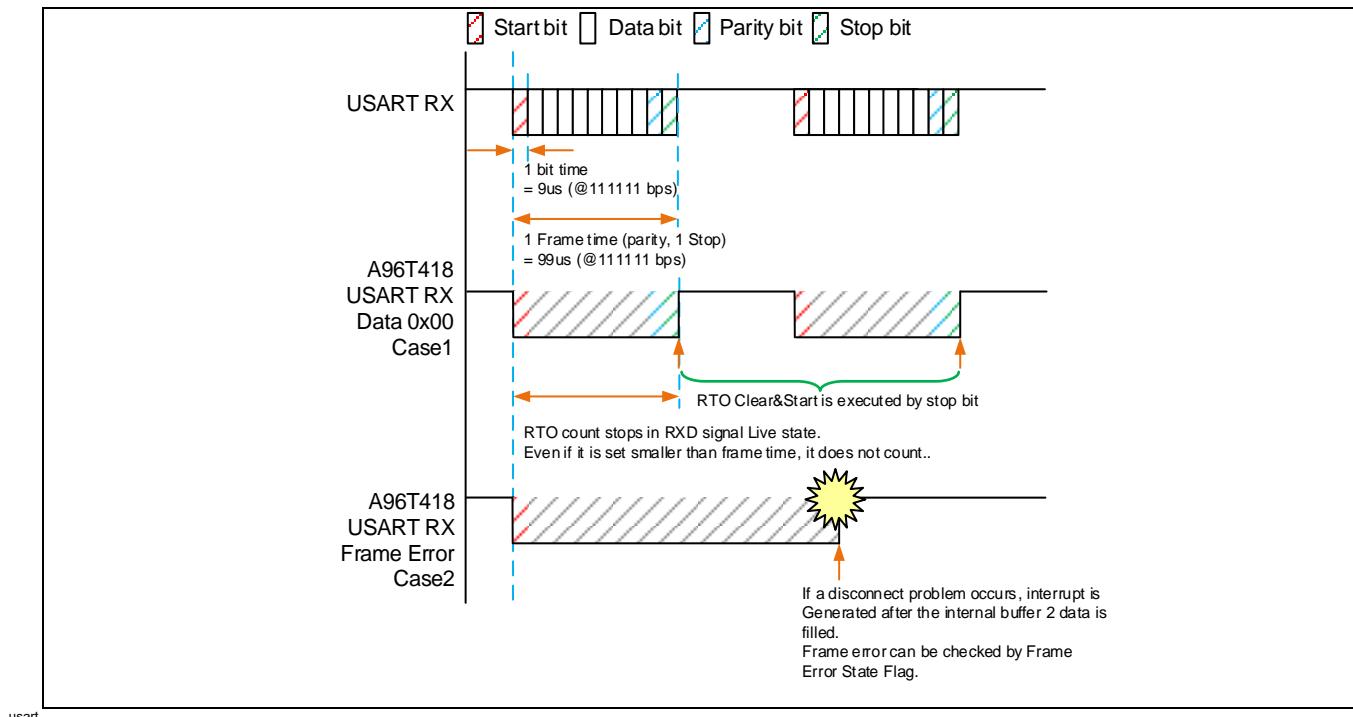


Figure 103. Example for RTO in USART1

## 14.12 Register Map

**Table 35. USART1 Register Map**

Name	Address	Direction	Default	Description
UCTRL1	CBH	R/W	00H	USART1 Control 1 Register
UCTRL2	CCH	R/W	00H	USART1 Control 2 Register
UCTRL3	CDH	R/W	00H	USART1 Control 3 Register
UCTRL4	1018H	R/W	00H	USART1 Control 4 Register
USTAT	CFH	R	80H	USART1 Status Register
UBAUD	FCH	R/W	FFH	USART1 Baud Rate Generation Register
UDATA	FDH	R/W	00H	USART1 Data Register
FPCR	1019H	R/W	00H	USART1 Floating Point Counter Register
RTOCH	101AH	R	00H	Receiver Time Out Counter High Register
RTOCL	101BH	R	00H	Receiver Time Out Counter Low Register

## 14.13 USART1 Register Description

USART1 module consists of USART1 Control 1 Register (UCTRL1), USART1 Control 2 Register (UCTRL2), USART1 Control 3 Register (UCTRL3), USART1 Control 4 Register (UCTRL4), USART1 Floating Point Counter (FPCR), USART1 Status Register (USTAT), USART1 Data Register (UDATA), and USART1 Baud Rate Generation Register (UBAUD).

## 14.14 Register Description for USART1

### UCTRL1 (USART1 Control 1 Register): CBH

7	6	5	4	3	2	1	0
UMSEL1	UMSEL0	UPM1	UPM0	USIZE2	USIZE1 UDORD	USIZE0 UCPHA	UCPOL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

UMSEL[1:0]	Selects operation mode of USART1		
UMSEL1	UMSEL0	Operating Mode	
0	0	Asynchronous Mode (Normal UART)	
0	1	Synchronous Mode (Synchronous UART)	
1	0	Reserved	
1	1	SPI Mode	
UPM[1:0]	Selects Parity Generation and Check methods		
UPM1	UPM0	Parity mode	
0	0	No Parity	
0	1	Reserved	
1	0	Even Parity	
1	1	Odd Parity	
USIZE[2:0]	When in asynchronous or synchronous mode of operation, selects the length of data bits in frame.		
USIZE2	USIZE1	USIZE0	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
UDORD	This bit is in the same bit position with USIZE1. In SPI mode, when set to one the MSB of the data byte is transmitted first. When set to zero the LSB of the data byte is transmitted first.		
0	LSB First		
1	MSB First		
UCPOL	Selects polarity of XCK in synchronous or SPI mode		
0	TXD1 change @Rising Edge, RXD1 change @Falling Edge		
1	TXD1 change @ Falling Edge, RXD1 change @ Rising Edge		
UCPHA	This bit is in the same bit position with USIZE0. In SPI mode, along with UCPOL bit, selects one of two clock formats for different kinds of synchronous serial peripherals. Leading edge means first XCK edge and trailing edge means 2 <sup>nd</sup> or last clock edge of XCK in one XCK pulse. And Sample means detecting of incoming receive bit, Setup means preparing transmit data.		
UCPOL	UCPHA	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)

**UCTRL2 (USART1 Control 2 Register): CCH**

7	6	5	4	3	2	1	0
UDRIE	TXCIE	RXCIE	WAKEIE	TXE	RXE	USARTEN	U2X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

UDRIE	Interrupt enable bit for USART1 Data Register Empty.
	0      Interrupt from UDRE is inhibited (use polling)
	1      When UDRE is set, request an interrupt
TXCIE	Interrupt enable bit for Transmit Complete.
	0      Interrupt from TXC is inhibited (use polling)
	1      When TXC is set, request an interrupt
RXCIE	Interrupt enable bit for Receive Complete
	0      Interrupt from RXC is inhibited (use polling)
	1      When RXC is set, request an interrupt
WAKEIE	Interrupt enable bit for Asynchronous Wake in STOP mode. When device is in stop mode, if RxD1 goes to LOW level an interrupt can be requested to wake-up system.
	0      Interrupt from Wake is inhibited
	1      When WAKE is set, request an interrupt
TXE	Enables the transmitter unit.
	0      Transmitter is disabled
	1      Transmitter is enabled
RXE	Enables the receiver unit.
	0      Receiver is disabled
	1      Receiver is enabled
USARTEN	Activate USART1 module by supplying clock.
	0      USART1 is disabled (clock is halted)
	1      USART1 is enabled
U2X	This bit only has effect for the asynchronous operation and selects receiver sampling rate.
	0      Normal asynchronous operation
	1      Double Speed asynchronous operation

**UCTRL3 (USART1 Control 3 Register): CDH**

7	6	5	4	3	2	1	0
MASTER	LOOPS	DISXCK	SPISS	-	USBS	TX8	RX8
R/W	R/W	R/W	R/W	-	R/W	R/W	R
Initial value: 00H							
<p><b>MASTER</b>      Selects master or slave in SPI or Synchronous mode operation and controls the direction of XCK pin.</p> <p>0      Slave mode operation and XCK is input pin. 1      Master mode operation and XCK is output pin</p>							
<p><b>LOOPS</b>      Controls the Loop Back mode of USART1, for test mode</p> <p>0      Normal operation 1      Loop Back mode</p>							
<p><b>DISXCK</b>      In Synchronous mode of operation, selects the waveform of XCK output.</p> <p>0      XCK is free-running while USART is enabled in synchronous master mode. 1      XCK is active while any frame is on transferring.</p>							
<p><b>SPISS</b>      Controls the functionality of SS1 pin in master SPI mode.</p> <p>0      SS1 pin is normal GPIO or other primary function 1      SS1 output to other slave device</p>							
<p><b>USBS</b>      Selects the length of stop bit in Asynchronous or Synchronous mode of operation.</p> <p>0      1 Stop bit 1      2 Stop bit</p>							
<p><b>TX8</b>      The ninth bit of data frame in Asynchronous or Synchronous mode of operation. Write this bit first before loading the UDATA register.</p> <p>0      MSB (9<sup>th</sup> bit) to be transmitted is '0' 1      MSB (9<sup>th</sup> bit) to be transmitted is '1'</p>							
<p><b>RX8</b>      The ninth bit of data frame in Asynchronous or Synchronous mode of operation. Read this bit first before reading the receive buffer.</p> <p>0      MSB (9<sup>th</sup> bit) received is '0' 1      MSB (9<sup>th</sup> bit) received is '1'</p>							

**UCTRL4 (USART1 Control 4 Register): 1018H**

7	6	5	4	3	2	1	0
-	-	-	RTOEN	RTO_FLAG	FPCREN	AOVSSEL	AOVSEN
-	-	-	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

RTOEN	Enable receiver time out.
	0 Disable
	1 Enable
RTO_FLAG	This bit is set when RTO count overflows. Writing '0' to this bit position will clear RTO_FLAG.
	0 RTO count dose not overflow.
	1 RTO count overflow.
FPCREN	Enable baud rate compensation
	0 Disable
	1 Enable
AOVSEN	Enable additional oversampling rates selection
	0 Disable
	1 Enable
AOVSSEL	Select additional oversampling rates
	0 Select X13
	1 Select X4

**USTAT (USART1 Status Register): CFH**

7	6	5	4	3	2	1	0
UDRE	TXC	RXC	WAKE	SOFTRST	DOR	FE	PE
R/W	R/W	R/W	R/W	R/W	R	R	R

Initial value: 80H

UDRE	The UDRE flag indicates if the transmit buffer (UDATA) is ready to be loaded with new data. If UDRE is '1', it means the transmit buffer is empty and can hold one or two new data. This flag can generate an UDRE interrupt. Writing '0' to this bit position will clear UDRE flag.
	0      Transmit buffer is not empty. 1      Transmit buffer is empty.
TXC	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 TXC interrupt is executed. It is also cleared by writing '0' to this bit position. This flag can generate a TXC interrupt.
	0      Transmission is ongoing. 1      Transmit buffer is empty and the data in transmit shift register are shifted out completely.
RXC	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 RXC flag can be used to generate a RXC interrupt.
	0      There is no data unread in the receive buffer 1      There are more than 1 data in the receive buffer
WAKE	This flag is set when the RX pin is detected low while the CPU is in stop mode. This flag can be used to generate a WAKE interrupt. This bit is set only when in asynchronous mode of operation. <small>NOTE</small>
	0      No WAKE interrupt is generated. 1      WAKE interrupt is generated.
SOFTRST	This is an internal reset and only has effect on USART. Writing '1' to this bit initializes the internal logic of USART and is auto cleared.
	0      No operation 1      Reset USART
DOR	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
FE	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
PE	This bit is set if the next character in the receive buffer has a Parity Error when received while Parity Checking is enabled. This bit is valid until the receive buffer is read.
	0      No Parity Error 1      Parity Error detected

**NOTES:** When the WAKE function of USART is used as a release source from STOP mode, it is required to clear this bit in the RX interrupt service routine. Else the device will not wake-up from STOP mode again by the change of RX pin.

**UBAUD (USART Baud-Rate Generation Register): FCH**

7	6	5	4	3	2	1	0
UBAUD7	UBAUD6	UBAUD5	UBAUD4	UBAUD3	UBAUD2	UBAUD1	UBAUD0
R/W							

Initial value: FFH

**UBAUD [7:0]** The value in this register is used to generate internal baud rate in asynchronous mode or to generate XCK clock in synchronous or SPI mode. To prevent malfunction, do not write '0' in asynchronous mode, and do not write '0' or '1' in synchronous or SPI mode.

**UDATA (USART Data Register): FDH**

7	6	5	4	3	2	1	0
UDATA7	UDATA6	UDATA5	UDATA4	UDATA3	UDATA2	UDATA1	UDATA0
R/W							

Initial value: 00H

**UDATA [7:0]** The USART 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 UDATA register. Reading the UDATA register returns the contents of the Receive Buffer. Write this register only when the UDRE flag is set. In SPI or synchronous master mode, write this register even if TX is not enabled to generate clock, XCK.

**FPCR (USART Floating Point Register): 1019H**

7	6	5	4	3	2	1	0
FPCR7	FPCR6	FPCR5	FPCR4	FPCR3	FPCR2	FPCR1	FPCR0
R/W							

Initial value: 00H

**FPCR [7:0]** USART Floating Point Counter  
8-bit floating point counter

**NOTES:** BAUD RATE compensation can be used in the following ways:

**Example1)** Condition: sysclk = 16MHz, Baud rate = 9600 bps, Asynchronous Normal Mode (U2X = 0)

$$\text{Baud rate} = \text{sysclk} / 16 \times (\text{UBAUD} + 1)$$

$$\text{Calculated UBAUD} = (1000000 / \text{Target Baud rate}) - 1 = 103.17, \text{ Error rate} = 0.17 \Rightarrow \text{UBAUD} = 104$$

UCTRL4 = 0x04, Enable baudrate Compensation

$$\text{Calculated FPCR} = (\text{UBAUD} - \text{Calculated UBAUD}) \times 256 = (104 - 103.17) \times 256 = 212.48 \Rightarrow \text{FPCR} = 213$$

**Example2)** Condition: sysclk = 16MHz, Baud rate = 115,200 bps, Asynchronous Normal Mode (U2X = 0)

$$\text{Baud rate} = \text{sysclk} / 16 \times (\text{UBAUD} + 1)$$

$$\text{Calculated UBAUD} = (1000000 / \text{Target Baud rate}) - 1 = 7.68, \text{ Error rate} = 0.68 \Rightarrow \text{UBAUD} = 8$$

UCTRL4 = 0x04, Enable baudrate Compensation

$$\text{Calculated FPCR} = (\text{UBAUD} - \text{Calculated UBAUD}) \times 256 = (8 - 7.68) \times 256 = 81.92 \Rightarrow \text{FPCR} = 82$$

**RTOCH (Receiver Time Out Counter High Register): 101AH**

7	6	5	4	3	2	1	0
RTOCH7	RTOCH6	RTOCH5	RTOCH4	RTOCH3	RTOCH2	RTOCH1	RTOCH0
R/W							

Initial value: 00H

**RTOCL (Receiver Time Out Counter Low Register): 101BH**

7	6	5	4	3	2	1	0
RTOCL7	RTOCL6	RTOCL5	RTOCL4	RTOCL3	RTOCL2	RTOCL1	RTOCL0
R/W							

Initial value: 00H

## 14.15 Baud Rate setting (example)

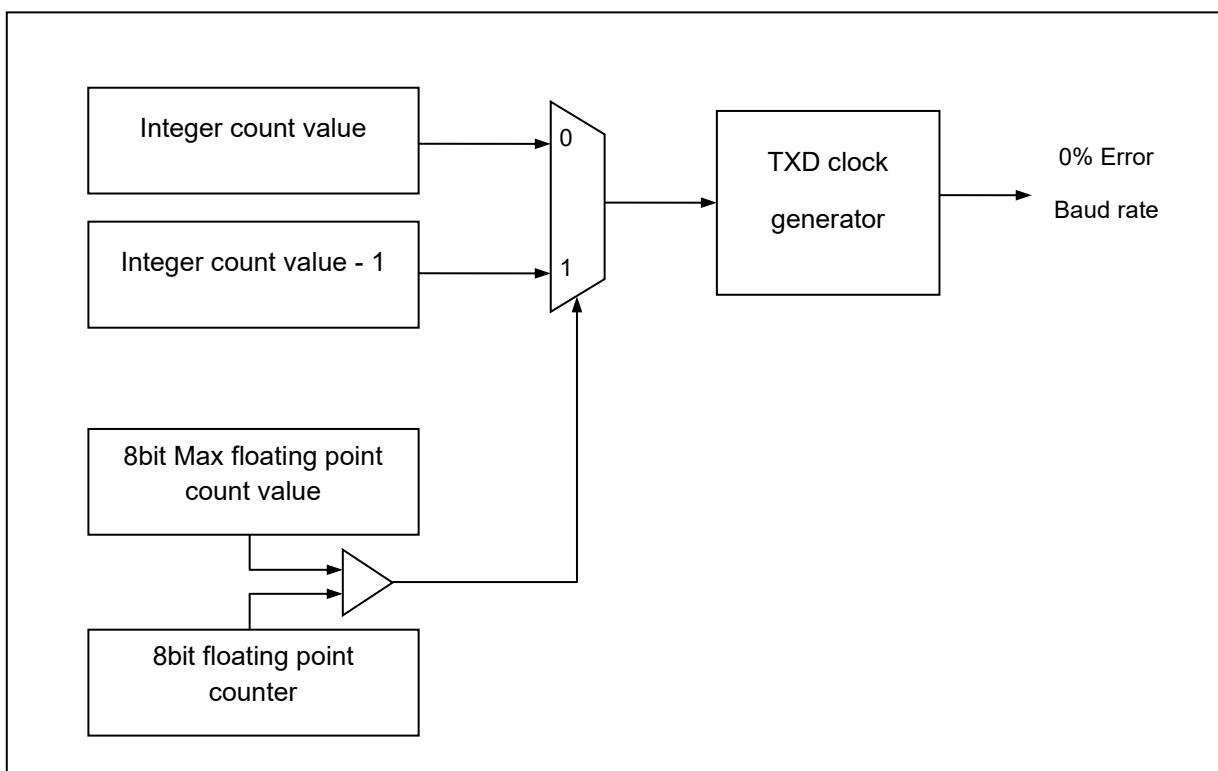
**Table 36. Examples of UBAUD Settings for Commonly Used Oscillator Frequencies**

Baud Rate	fOSC=1.00MHz				fOSC=1.8432MHz				fOSC=2.00MHz			
	U2X=0		U2X=1		U2X=0		U2X=1		U2X=0		U2X=1	
	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR
2400	25	0.2%	51	0.2%	47	0.0%	95	0.0%	51	0.2%	103	0.2%
4800	12	0.2%	25	0.2%	23	0.0%	47	0.0%	25	0.2%	51	0.2%
9600	6	-7.0%	12	0.2%	11	0.0%	23	0.0%	12	0.2%	25	0.2%
14.4K	3	8.5%	8	-3.5%	7	0.0%	15	0.0%	8	-3.5%	16	2.1%
19.2K	2	8.5%	6	-7.0%	5	0.0%	11	0.0%	6	-7.0%	12	0.2%
28.8K	1	8.5%	3	8.5%	3	0.0%	7	0.0%	3	8.5%	8	-3.5%
38.4K	1	-18.6%	2	8.5%	2	0.0%	5	0.0%	2	8.5%	6	-7.0%
57.6K	-	-	1	8.5%	1	-25.0%	3	0.0%	1	8.5%	3	8.5%
76.8K	-	-	1	-18.6%	1	0.0%	2	0.0%	1	-18.6%	2	8.5%
115.2K	-	-	-	-	-	-	1	0.0%	-	-	1	8.5%
230.4K	-	-	-	-	-	-	-	-	-	-	-	-
Baud Rate	fOSC=3.6864MHz				fOSC=4.00MHz				fOSC=7.3728MHz			
	U2X=0		U2X=1		U2X=0		U2X=1		U2X=0		U2X=1	
	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR
2400	95	0.0%	191	0.0%	103	0.2%	207	0.2%	191	0.0%	-	-
4800	47	0.0%	95	0.0%	51	0.2%	103	0.2%	95	0.0%	191	0.0%
9600	23	0.0%	47	0.0%	25	0.2%	51	0.2%	47	0.0%	95	0.0%
14.4K	15	0.0%	31	0.0%	16	2.1%	34	-0.8%	31	0.0%	63	0.0%
19.2K	11	0.0%	23	0.0%	12	0.2%	25	0.2%	23	0.0%	47	0.0%
28.8K	7	0.0%	15	0.0%	8	-3.5%	16	2.1%	15	0.0%	31	0.0%
38.4K	5	0.0%	11	0.0%	6	-7.0%	12	0.2%	11	0.0%	23	0.0%
57.6K	3	0.0%	7	0.0%	3	8.5%	8	-3.5%	7	0.0%	15	0.0%
76.8K	2	0.0%	5	0.0%	2	8.5%	6	-7.0%	5	0.0%	11	0.0%
115.2K	1	0.0%	3	0.0%	1	8.5%	3	8.5%	3	0.0%	7	0.0%
230.4K	-	-	1	0.0%	-	-	1	8.5%	1	0.0%	3	0.0%
250K	-	-	1	-7.8%	-	-	1	0.0%	1	-7.8%	3	-7.8%
0.5M	-	-	-	-	-	-	-	-	-	-	1	-7.8%
Baud Rate	fOSC=8.00MHz				fOSC=11.0592MHz				fOSC=14.7456MHz			
	U2X=0		U2X=1		U2X=0		U2X=1		U2X=0		U2X=1	
	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR
2400	207	0.2%	-	-	-	-	-	-	-	-	-	-
4800	103	0.2%	207	0.2%	143	0.0%	-	-	191	0.0%	-	-
9600	51	0.2%	103	0.2%	71	0.0%	143	0.0%	95	0.0%	191	0.0%
14.4K	34	-0.8%	68	0.6%	47	0.0%	95	0.0%	63	0.0%	127	0.0%
19.2K	25	0.2%	51	0.2%	35	0.0%	71	0.0%	47	0.0%	95	0.0%
28.8K	16	2.1%	34	-0.8%	23	0.0%	47	0.0%	31	0.0%	63	0.0%
38.4K	12	0.2%	25	0.2%	17	0.0%	35	0.0%	23	0.0%	47	0.0%
57.6K	8	-3.5%	16	2.1%	11	0.0%	23	0.0%	15	0.0%	31	0.0%
76.8K	6	-7.0%	12	0.2%	8	0.0%	17	0.0%	11	0.0%	23	0.0%
115.2K	3	8.5%	8	-3.5%	5	0.0%	11	0.0%	7	0.0%	15	0.0%
230.4K	1	8.5%	3	8.5%	2	0.0%	5	0.0%	3	0.0%	7	0.0%
250K	1	0.0%	3	0.0%	2	-7.8%	5	-7.8%	3	-7.8%	6	5.3%
0.5M	-	-	1	0.0%	-	-	2	-7.8%	1	-7.8%	3	-7.8%
1M	-	-	-	-	-	-	-	-	-	-	1	-7.8%

## 14.16 0% Error Baud Rate

This USART1 system supports the floating point counter logic for the 0% error of baud rate. By using the 8bits floating point counter logic, the cumulative error to below the decimal point can be removed.

The floating point counter value is defined by baud rate error. In the baud rate formula, BAUD is presented the integer count value. For example, If you want to use the 57600 baud rate ( $f_{SCLK} = 16\text{MHz}$ ), integer count value must be 16.36 value ( $\text{BAUD}+1 = 16000000/(16 \times 57600) = 17.36$ ). Here, the accurate BAUD value is 16.36. To realize 0% error of baud rate, floating point counter value must be 164 ( $((17-16.36) \times 256 \approx 164)$ ) and BAUD value must be 17. Namely you have to write the 164(decimal number) in USART\_FPCR and 17(decimal number) in USART\_BAUD.



**Figure 104. 0% Error Baud Rate Block Diagram**

## 15 LED Driver

### 15.1 Overview

LED drive contains 8 COM / 16 SEG output pins. They are also shared with Touch sensing pins. By setting LED CONTROL REGISTER 1 (LEDCON1), there are 5 modes that can be shared(or not shared) with touch sensing function. The controller consists of display data RAM memory, COM and SEG generator. COM0-COM7 are shared with SEG0-SEG7. It is selected by CSER register. SEG8~SEG15 is dedicated only SEG function in LED function. COM and SEG pin can also be used as I / O pins. COMOE, and SEGOE1,SEGOE2 registers are used to select SEG0-SEG14, COM0- COM7.

During the power-on reset, reset pin, BOD reset or watchdog reset, LED are turned off.

### 15.2 Block Diagram

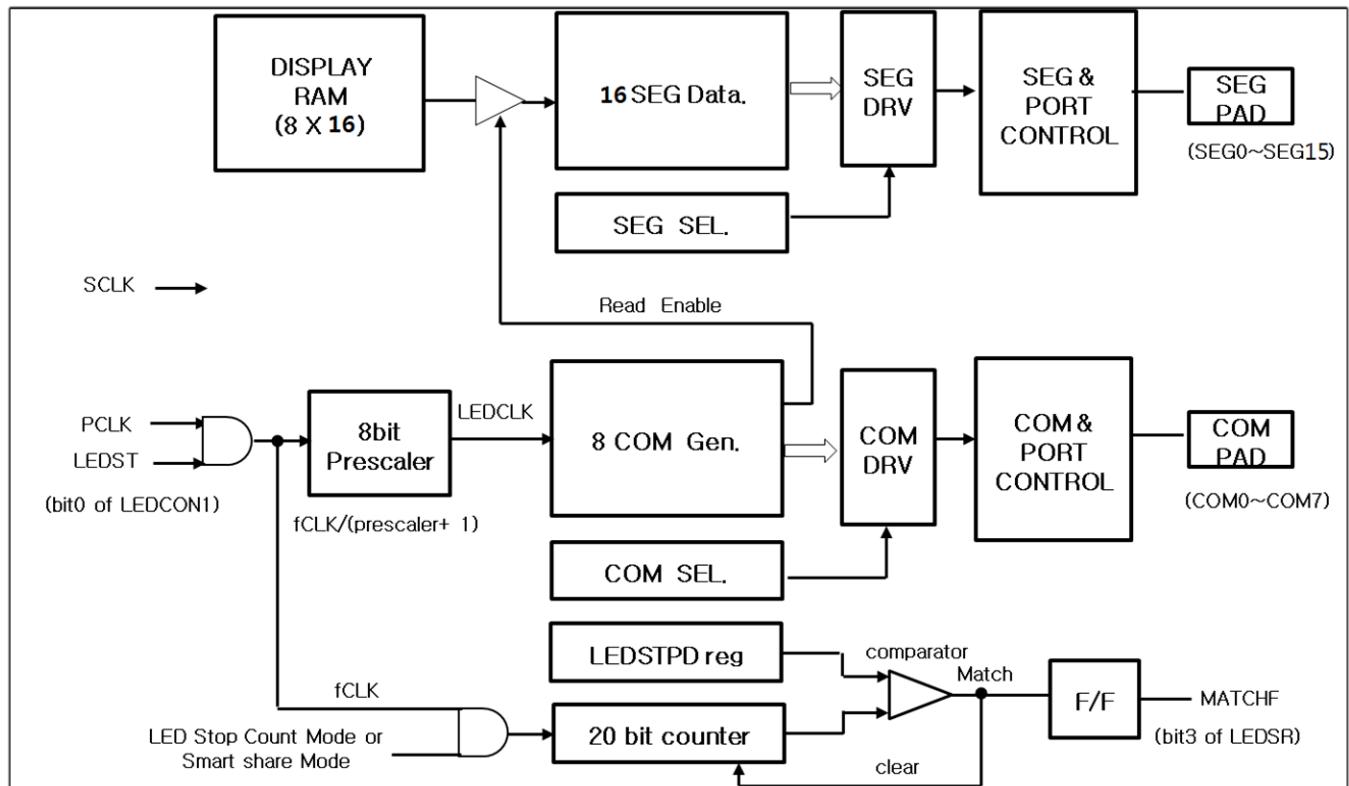


Figure 105. LED Driver Block Diagram

### 15.3 Register Map

**Table 37. LED Driver Register Map**

Name	Address	Direction	Default	Description
COMOE	2F00H	R/W	00H	LED COM Output Enable Register
SEGOE1	2F01H	R/W	00H	LED SEG Output Enable 1 Register
SEGOE2	2F02H	R/W	00H	LED SEG Output Enable 2 Register
PRESD	2F03H	R/W	00H	LED Prescale Data Register
CSER	2F04H	R/W	00H	LED COM or SEG Enable Register
COMPWID	2F05H	R/W	00H	LED COM Pulse Width Control Register
STPD0	2F0AH	R/W	00H	LED Stop Duration 0 Register
STPD1	2F0BH	R/W	00H	LED Stop Duration 1 Register
STPD2	2F0CH	R/W	00H	LED Stop Duration 2 Register
LEDSR	2F0DH	R/W	01H	LED Status Register
LEDCON3	2F0EH	R/W	00H	LED Control3 Register
LEDCON2	2F0FH	R/W	00H	LED Control 2 Register
LEDCON1	2F10H	R/W	00H	LED Control 1 Register
CCLSR	2F30H	R/W	01H	Constant Current level select Register
TEST_COM	2F7CH	R/W	00H	COM Register for test
TEST_SEG0	2F7DH	R/W	00H	SEG Register for test
TEST_SEG1	2F7EH	R/W	00H	SEG Register for test

## 15.4 Register Description for LED Driver

### COMOE (LED COM Output Enable Register): 2F00H

7	6	5	4	3	2	1	0
COMOE7	COMOE6	COMOE5	COMOE4	COMOE3	COMOE2	COMOE1	COMOE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value: 00H							
COMOE7      COM7 Output Enable 0      Disable 1      Enable							
COMOE6      COM6 Output Enable 0      Disable 1      Enable							
COMOE5      COM5 Output Enable 0      Disable 1      Enable							
COMOE4      COM4 Output Enable 0      Disable 1      Enable							
COMOE3      COM3 Output Enable 0      Disable 1      Enable							
COMOE2      COM2 Output Enable 0      Disable 1      Enable							
COMOE1      COM1 Output Enable 0      Disable 1      Enable							
COMOE0      COM0 Output Enable 0      Disable 1      Enable							

**SEGOE1 (LED SEG Output Enable 1 Register): 2F01H**

7	6	5	4	3	2	1	0
SEGOE1.7	SEGOE1.6	SEGOE1.5	SEGOE1.4	SEGOE1.3	SEGOE1.2	SEGOE1.1	SEGOE1.0
RW	RW	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

SEGOE1.7	SEG7 Output Enable
0	Disable
1	Enable
SEGOE1.6	SEG6 Output Enable
0	Disable
1	Enable
SEGOE1.5	SEG5 Output Enable
0	Disable
1	Enable
SEGOE1.4	SEG4 Output Enable
0	Disable
1	Enable
SEGOE1.3	SEG3 Output Enable
0	Disable
1	Enable
SEGOE1.2	SEG2 Output Enable
0	Disable
1	Enable
SEGOE1.1	SEG1 Output Enable
0	Disable
1	Enable
SEGOE1.0	SEG0 Output Enable
0	Disable
1	Enable

**SEGOE2 (LED SEG Output Enable 2 Register): 2F02H**

7	6	5	4	3	2	1	0
SEGOE2.7	SEGOE2.6	SEGOE2.5	SEGOE2.4	SEGOE2.3	SEGOE2.2	SEGOE2.1	SEGOE2.0
RW	RW	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

SEGOE2.7	SEG15 Output Enable
0	Disable
1	Enable
SEGOE2.6	SEG14 Output Enable
0	Disable
1	Enable
SEGOE2.5	SEG13 Output Enable
0	Disable
1	Enable
SEGOE2.4	SEG12 Output Enable
0	Disable
1	Enable
SEGOE2.3	SEG11 Output Enable
0	Disable
1	Enable
SEGOE2.2	SEG10 Output Enable
0	Disable
1	Enable
SEGOE2.1	SEG9 Output Enable
0	Disable
1	Enable
SEGOE2.0	SEG8 Output Enable
0	Disable
1	Enable

**PRESD (LED Prescale Data Register): 2F03H**

7	6	5	4	3	2	1	0
PRESD7	PRESD6	PRESD5	PRESD4	PRESD3	PRESD2	PRESD1	PRESD0
R/W							

Initial value: 00H

PRESD[7:0]	Prescale value of LED Clock LED Clock = fCLK / (PRESD + 1) (fCLK is a selected input clock)
------------	---

**CSER (LED COM or SEG Enable Register): 2F04H**

7	6	5	4	3	2	1	0
CS_SEL7	CS_SEL6	CS_SEL5	CS_SEL4	CS_SEL3	CS_SEL2	CS_SEL1	CS_SEL0
RW	RW	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

CS_SEL7	Select COM7/SEG7
0	Select SEG7
1	Select COM7
CS_SEL6	Select COM6/SEG6
0	Select SEG6
1	Select COM6
CS_SEL5	Select COM5/SEG5
0	Select SEG5
1	Select COM5
CS_SEL4	Select COM4/SEG4
0	Select SEG4
1	Select COM4
CS_SEL3	Select COM3/SEG3
0	Select SEG3
1	Select COM3
CS_SEL2	Select COM2/SEG2
0	Select SEG2
1	Select COM2
CS_SEL1	Select COM1/SEG1
0	Select SEG1
1	Select COM1
CS_SEL0	Select COM0/SEG0
0	Select SEG0
1	Select COM0

**COMPWID (LED COM Pulse Width Control Register): 2F05H**

7	6	5	4	3	2	1	0
COMPWID7	COMPWID6	COMPWID5	COMPWID4	COMPWID3	COMPWID2	COMPWID1	COMPWID0
RW	RW	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

COMPWID[7:0] Maximum Pulse width for COM  
 COM Width = LED CLK / (COMPWID + 1)  
 (LED CLK is a Prescaler output clock)

**STPD0 (LED stop Duration 0 Register): 2F0AH**

7	6	5	4	3	2	1	0
STPD7	STPD6	STPD5	STPD4	STPD3	STPD2	STPD1	STPD0
RW							

Initial value: 00H

**STPD1 (LED stop Duration 1 Register): 2F0BH**

7	6	5	4	3	2	1	0
STPD15	STPD14	STPD13	STPD12	STPD11	STPD10	STPD9	STPD8
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

**STPD2 (LED stop Duration 2 Register): 2F0CH**

7	6	5	4	3	2	1	0
-	-	-	-	STPD19	STPD18	STPD17	STPD16
-	-	-	-	RW	RW	RW	RW

Initial value: 00H

STPD[19:0]      LED stop duration. These registers are valid in LED Stop Count mode or smart share mode.

LED Stop Duration Width = fCLK / (STPD + 1)  
(fCLK is a selected input clock)

**LEDSR (LED Status Register): 2F0DH**

7	6	5	4	3	2	1	0
-	-	-	-	MATCHF	LED_INT	LED_INTE	LED_ENDF
-	-	-	-	RW	RW	RW	RW

Initial value: 01H

MATCHF      This bit is set when the value of internal counter becomes the same with STPD register. After LED display, this bit will be cleared automatically.

0      Unmatched

1      Matched

LED\_INT      LED Interrupt flag when LED\_INTE is set.

0      LED interrupt not generated

1      LED interrupt generated

LED\_INTE      LED Interrupt Enable

0      Disable

1      Enable

LED\_ENDF      LED Operation Flag

0      Under LED Operation

1      LED Operation is ended

**LEDCON3 (LED Control 3 Register): 2F0EH**

7	6	5	4	3	2	1	0
SET_TIME3	SET_TIME2	SET_TIME1	SET_TIME0	CLR_TIME3	CLR_TIME2	CLR_TIME1	CLR_TIME0
RW	RW	R/W	R/W	R?W-	R/W-	RW	RW

Initial value: 00H

**SET\_TIME[3:0]** SEG GND Start position These bits are valid only when SRTEN(LEDCON2[3]) is High. fCLK = 16MHZ

0000	1us
0001	2us
0010	4us
0011	6us
0100	8us
0101	10us
0110	12us
0111	14us
1000	16us
1001	18us
1010	20us
1011	22us
1100	24us
1101	26us
1110	28us
1111	32us

**CLR\_TIME[3:0]** SEG GND Start position These bits are valid only when SRTEN(LEDCON2[3]) is High. fCLK = 16MHZ, OVERTS is bit[6:4] of LEDCON2

0000	OVERTS – 1us
0001	OVERTS – 2us
0010	OVERTS – 4us
0011	OVERTS – 6us
0100	OVERTS – 8us
0101	OVERTS – 10us
0110	OVERTS – 12us
0111	OVERTS – 14us
1000	OVERTS – 16us
1001	OVERTS – 18us
1010	OVERTS – 20us
1011	OVERTS – 22us
1100	OVERTS – 24us
1101	OVERTS – 26us
1110	OVERTS – 28us
1111	OVERTS – 32us

**LEDCON2 (LED Control 2 Register): 2F0FH**

7	6	5	4	3	2	1	0
OVERLAP	OVERTS2	OVERTS1	OVERTS0	SRTEN	-	-	-
RW	RW	R/W	R/W	-	-	-	-

Initial value: 00H

OVERLAP	How to make Overlap Time	
	0      Overlap time = 3us ~ 64us @ fCLK = 16MHZ	
	1      Overlap time = fCLK/2	
OVERTS[2:0]	OVERLAP TIME SELECT. These bits are valid only when OVERLAP is LOW.	
	000	64us
	001	32us
	010	24us
	011	21us
	100	12us
	101	9us
	110	6us
	111	3us
SRTEN	SEG GND Select	
	0	non select
	1	Select

**LEDCON1 (LED Control 1 Register): 2F10H**

7	6	5	4	3	2	1	0
-	-	-	MD2	MD1	MD0	LEDEN	LEDST
-	-	-	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

MD[2:0]	Mode Select	
	000    LED Auto Mode	
	001    Hand Shake Mode	
	010    LED STOP Count Mode	
	011    Smart Share Mode	
	1xx    LED Alone Mode	
LEDEN	LED Enable	
	0	LED Disable
	1	LED Enable
LEDST	LED Start or Stop under LEDEN = 1	
	0	Stop LED Operation
	1	Start LED Operation

**CCLSR (Constant Current Level Select Register): 2F30H**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	COA1	COA0
-	-	-	-	-	-	RW	RW

Initial value: 01H

COA[1:0]	SEG(mA)
00	8.66mA
01	13.22mA (default)
10	19.49mA
11	22.06mA

**TEST\_COM (TEST COM Register for Test): 2F7CH**

7	6	5	4	3	2	1	0
TCOM7	TCOM6	TCOM5	TCOM4	TCOM3	TCOM2	TCOM1	TCOM0
RW							

Initial value: 00H

**TEST\_SEG0 (TEST SEG Low Byte Register for Test): 2F7DH**

7	6	5	4	3	2	1	0
TSEG07	TSEG06	TSEG05	TSEG04	TSEG03	TSEG02	TSEG01	TSEG00
RW							

Initial value: 00H

**TEST\_SEG0 (TEST SEG High Byte Register for Test): 2F7EH**

7	6	5	4	3	2	1	0
TSEG17	TSEG16	TSEG15	TSEG14	TSEG13	TSEG12	TSEG11	TSEG10
RW							

Initial value: 00H

## 15.5 LED RAM

**Table 38. LED RAM**

<b>Address</b>		<b>bit7/15</b>	<b>bit6/14</b>	<b>bit5/13</b>	<b>bit4/12</b>	<b>bit3/11</b>	<b>bit2/10</b>	<b>bit1/9</b>	<b>bit0/8</b>
2F13H	COM0L	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	-
2F14H	COM0H	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8
2F15H	COM1L	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	-	SEG0
2F16H	COM1H	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8
2F17H	COM2L	SEG7	SEG6	SEG5	SEG4	SEG3	-	SEG1	SEG0
2F18H	COM2H	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8
2F19H	COM3L	SEG7	SEG6	SEG5	SEG4	-	SEG2	SEG1	SEG0
2F1AH	COM3H	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8
2F1BH	COM4L	SEG7	SEG6	SEG5	-	SEG3	SEG2	SEG1	SEG0
2F1CH	COM4H	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8
2F1DH	COM5L	SEG7	SEG6	-	SEG4	SEG3	SEG2	SEG1	SEG0
2F1EH	COM5H	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8
2F1FH	COM6L	SEG7	-	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
2F20H	COM6H	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8
2F21H	COM7L	-	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
2F22H	COM7H	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8

"-" means no affect to SEG output

## 15.6 Example Circuit

### 15.6.1 T-Type Matrix

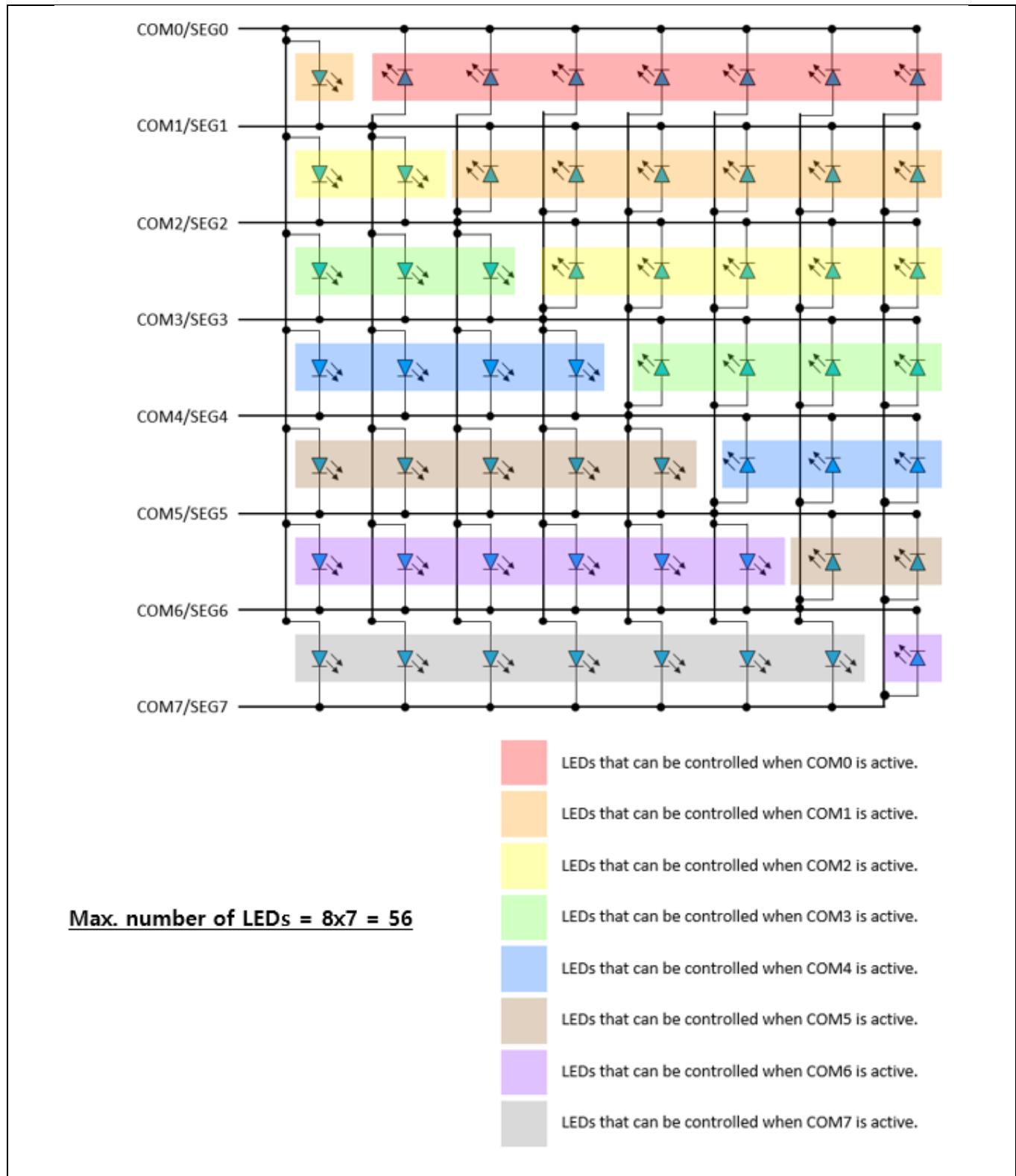


Figure 106. T-Type Matrix

### 15.6.2 M-Type Matrix

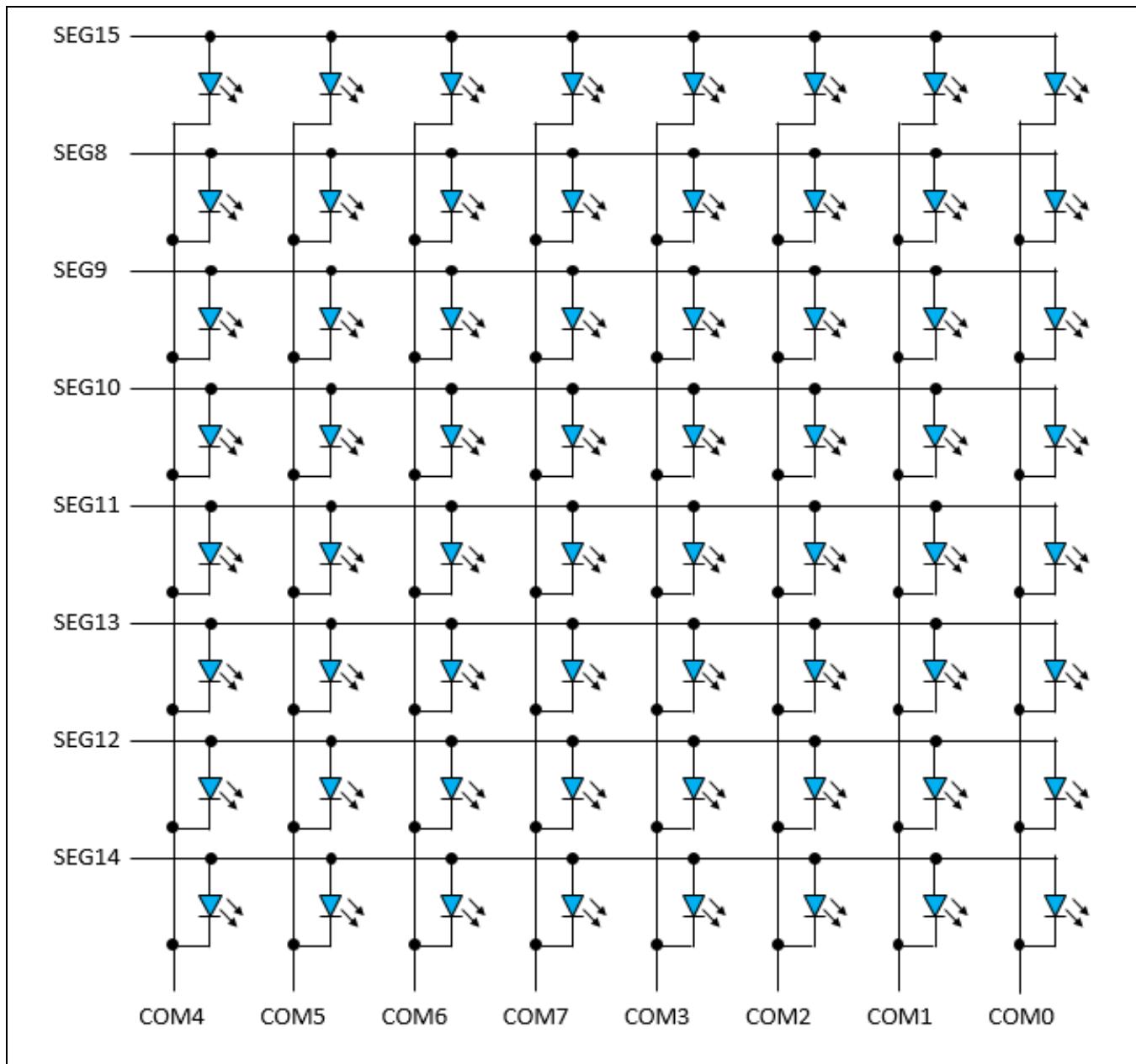


Figure 107. M-Type Matrix

### 15.6.3 Circuit for LED and Touch by common pin

When implementing touch and LED functions on a common pin by time-division, the touch sensitivity may be very poor due to the parasitic capacitance caused by LED matrix structure. This problem can be minimized by adding a diode in front of the LED matrix. Refer to the diode's capacitance specification, and select a diode less than 4pF. It is recommended to use the following circuit and diodes.

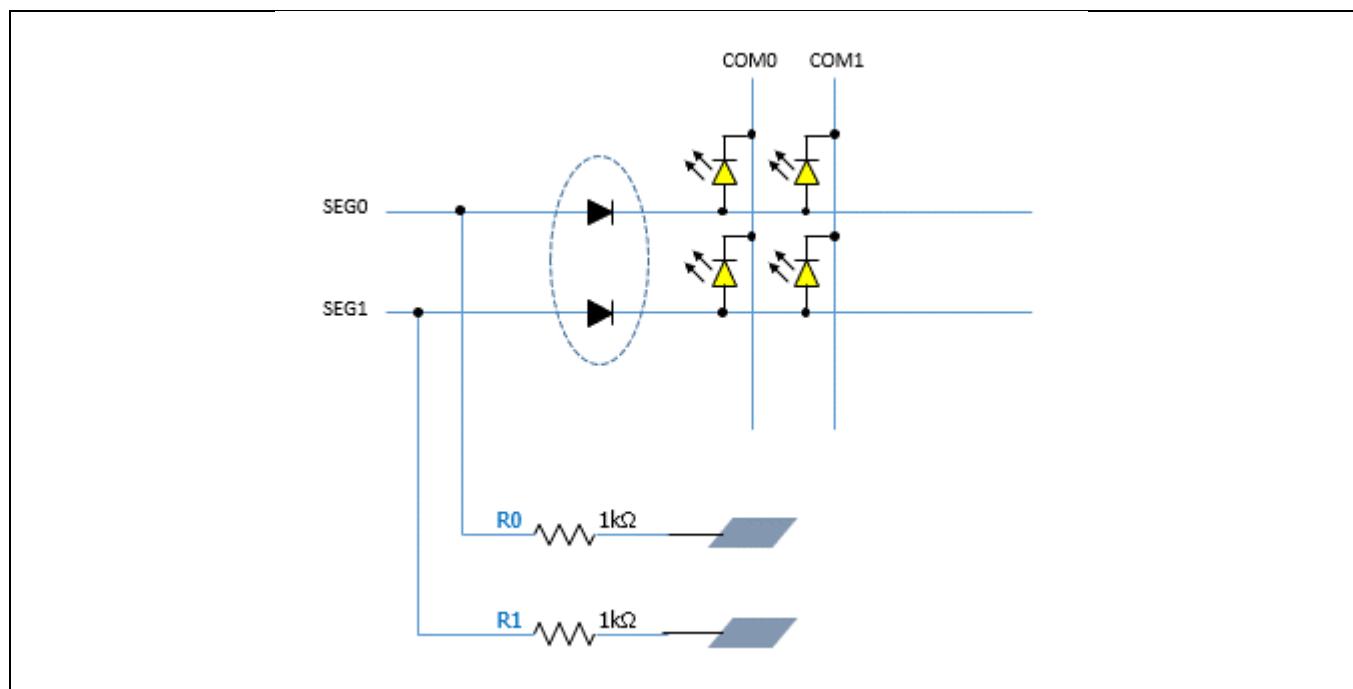


Figure 108. Circuit for LED and Touch by common pin

Table 39. Recommended diodes

No	IC-NAME	VENDOR	Capacitance
1	1N4148-B	RECTRON	4pF
2	1N4148W	TAIWAN SEMI	4pF
3	1N4148W-7-F	DIODES	2pF
4	1N4148W-E3-08	VISHAY	4pF
5	1N4148W-G3-18	VISHAY	4pF
6	1N4148WS	ON SEMI	4pF
7	1N4148WSF-7	DIODES	1.5pF

## 15.7 Mode Functions

By depending on how set MD values (LEDCON1[4:2]), there are fifth operation modes.

### 15.7.1 LED AUTO Mode

First mode is LED auto mode irrelevant to touch sensing function that can start by write "1" LEDST by program.

Below all mode is possible only after LEDEN (LEDCON1[1]) is set "1".

Figure 109 show the LED auto mode in case of CSER = 0100\_0001B. (k) shows the overlaptimes that exists between the COM and the COM that are controllable by the program.

When enabling the LED by write LEDST "1" as in (A), LED can operate continually..

An interrupt occurs at point (B) where the LED 1 frame operation terminated when LED\_INTE is set.

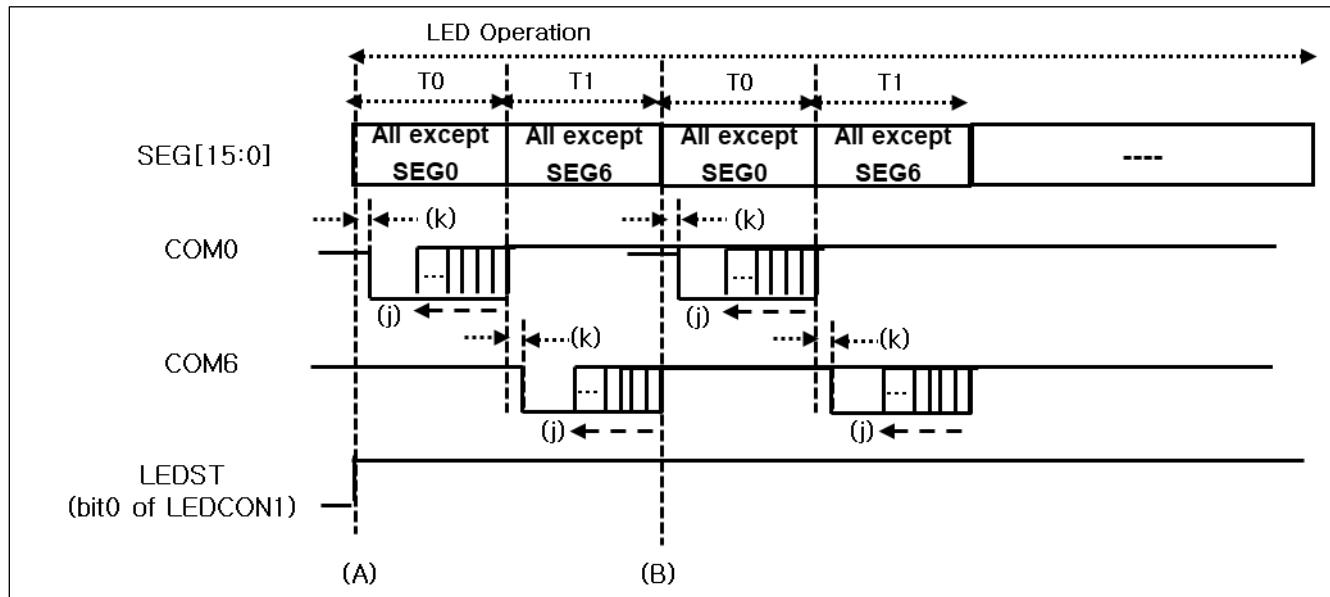


Figure 109. LED auto Mode

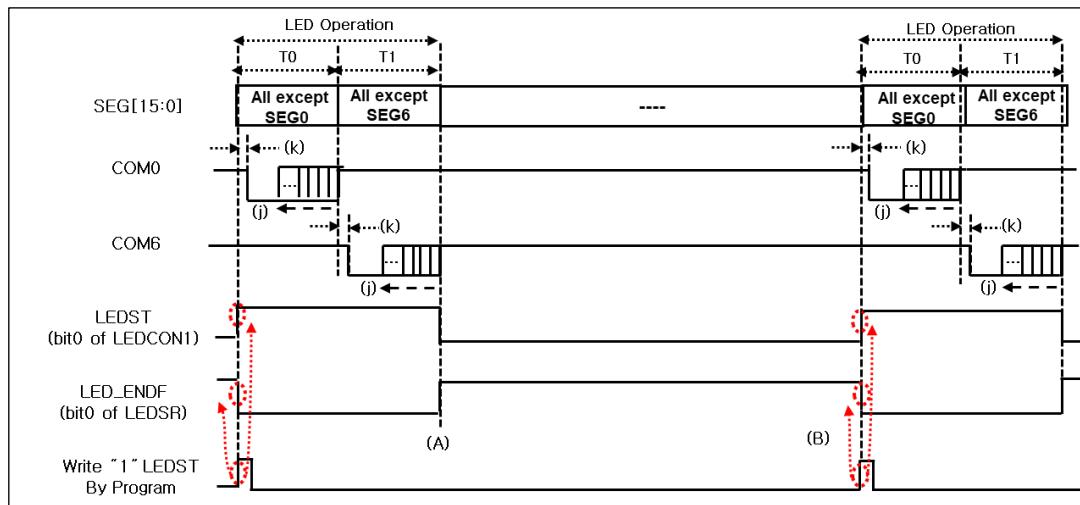
### 15.7.2 LED Alone Mode

First mode is LED alone mode irrelevant to touch sensing function that can re-start by write "1" LEDST by program.

Figure 110 show the LED alone mode in case of CSER = 0100\_0001B. (k) shows the overlaptimes that exists between the COM and the COM that are controllable by the program.

An interrupt occurs at point (A) where the LED operation terminated when LED\_INTE is set.

When enabling the LED by rewrite LEDST "1" as in (B), LED can starts again.



**Figure 110. LED alone Mode**

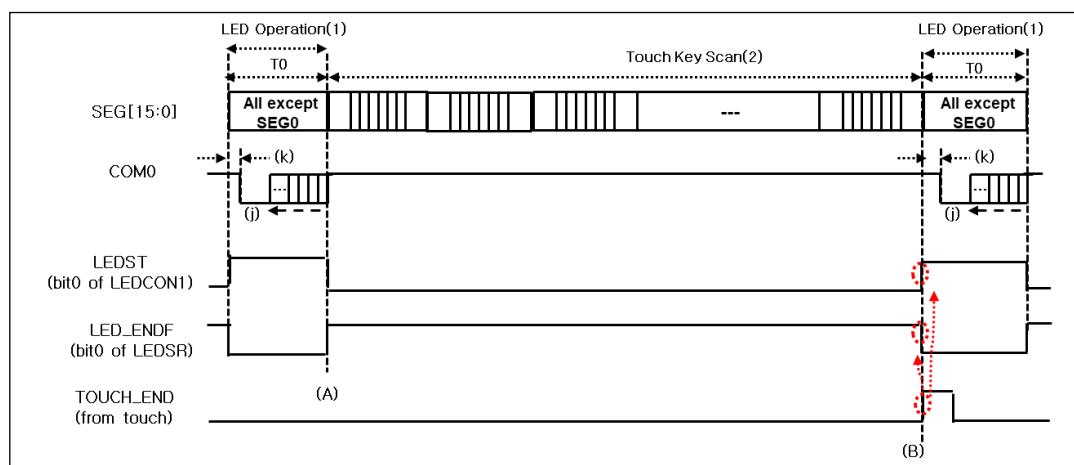
### 15.7.3 Hand Shake Mode

The second mode is hand shake mode to inform the touch when the LED is finished and to make LED start again by the TOUCH\_END signal from the touch sensing.

Figure 111 show the hand shake mode in the case of CSER = 0000\_0001B. Frame cycle is defined by sum of LED operation (1) and touch key scan (2). It is the minimum cycle without flicker.

An interrupt occurs at point (A) where the LED operation terminated when LED\_INTE is set.

Touch sensing continues to send the “1” until LED\_ENDF changes to “0” in (B), and then LED start again.



**Figure 111. Hand Shake Mode**

### 15.7.4 LED Stop Count Mode

The third mode is LED stop count mode that the beginning of the next LED operation is determined by LED stop duration register value after the previous LED operation end.

It is need when deciding the period of frame that LED can operate without flicker.

Figure 112 show LED STOP Count mode in case of CSER = 1000\_0000B. An interrupt occurs at point (A) here the LED operation terminated when LED\_INTE is set.

At the same time, 20-bit counter starts to increase by the internal enable bit. If 20-bit LEDSTPD register is matched with the counter, MATCHF(bit3 of LEDSR) is set “1”. LED\_ENDF is cleared “0”, 20-bit counter is cleared “0” and LED start again as in (B).

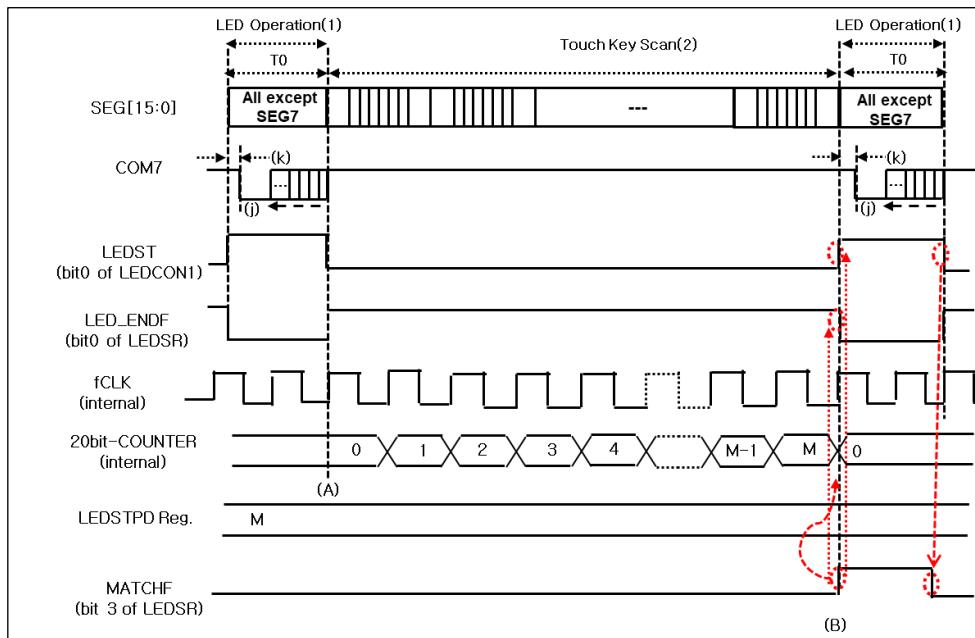


Figure 112. LED Stop Count Mode

### 15.7.5 Smart Share Mode

The fourth mode is smart share mode that the beginning of the next LED operation can't do until current pending touch key scan ends although MATCHF (LEDSR[3]) generate after the previous LED operation end.

It is need when as many as touch key scan want to be executed during touch key scan time.

Figure 113 show smart share mode in case of CSER = 1000\_0000B. An interrupt occurs at point (A) here the LED operation terminated when LED\_INTE is set. At the same time, 20-bit counter starts to increase by the internal enable bit although 20-bit STPD register is matched with the counter and then MATCHF (LEDSR[3]) is set “1”.

LED\_ENDF can't clear “0” and LED can't start again until the TOUCH\_END signal from the touch sensing is active.

If any more LED display is not needed, it is highly recommended LEDEN (LEDCON1[1]) setting as “0” to be executed between the end of LED operation (A) and LED restart (B) to prevent flicker.

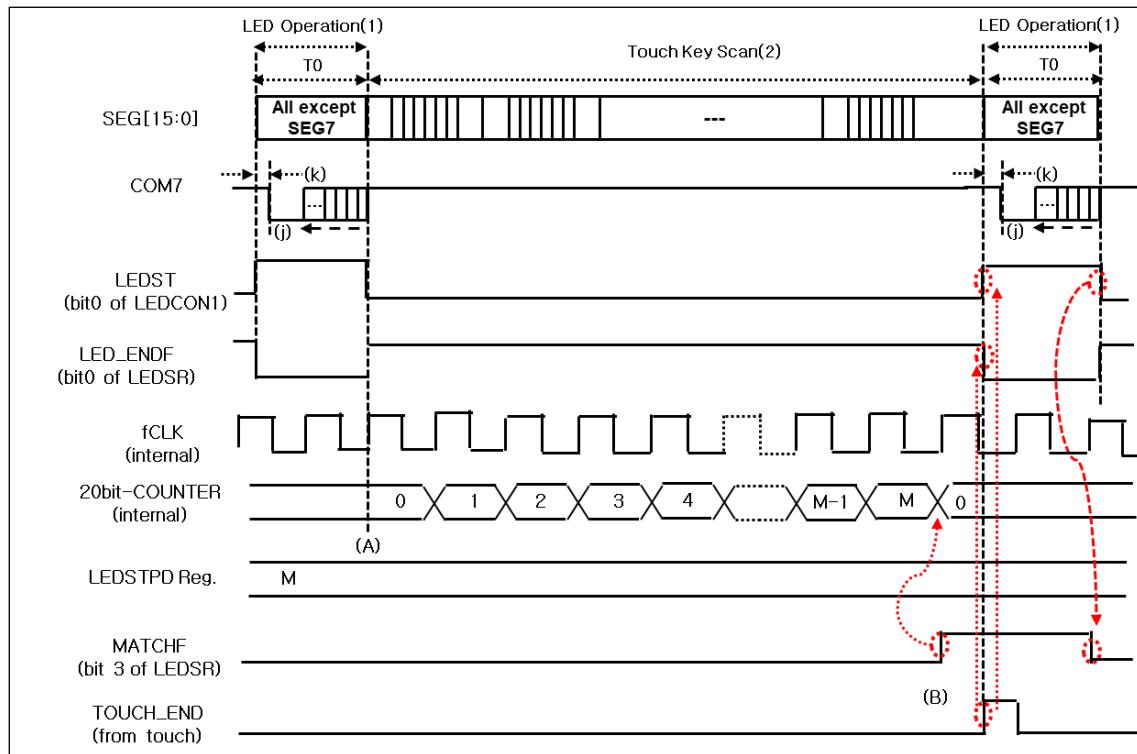


Figure 113. Smart Share Mode

## 16 20-Channel Touch Switch

### 16.1 Features

- 10V Conducted Susceptibility (CS) Immunity
- Self-Capacitive Touch Key Sensor.
- Total 20-channel Touch Key Support.
- 16-bits Sensing Resolutions.
- Fast Initial Self Calibration.
- Key Detection Mode : Single/Multi-Mode.
- Clock Frequency during Sensing Operation : 16MHz.
- The Improvement of the SNR by Bias-Calibration in Analog Sensing Block.
- VDD Operating Voltage : 2.7V ~ 5.5V.
- Current Consumption : T.B.D.
- Current Consumption@STOPmode :< 1uA.
- Operation Temperature : -40°C ~ 105°C.

## 16.2 Block Diagram

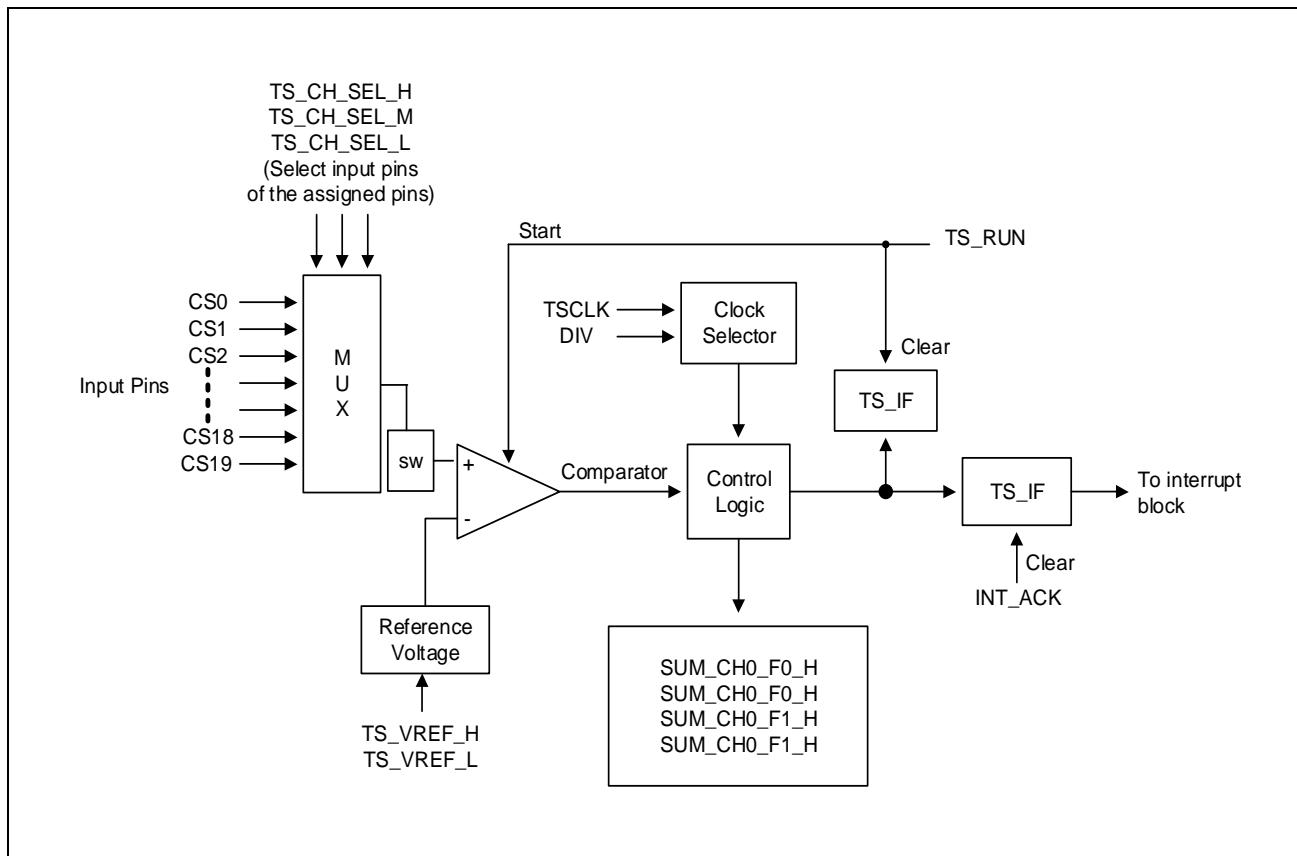


Figure 114. 16-bit Touch Block Diagram

## 16.3 CAPN Port

Connect the negative terminal of the reference capacitor  $C_s$  to CAPN port, and the positive terminal of the  $C_s$  capacitor to VDD.  $C_s$  capacitors must use 5% precision polyester plug-in capacitors, 10% high precision NPO or X7R chip capacitors. C0G type or Mylar Capacitors recommended for applications with severe temperature changes.

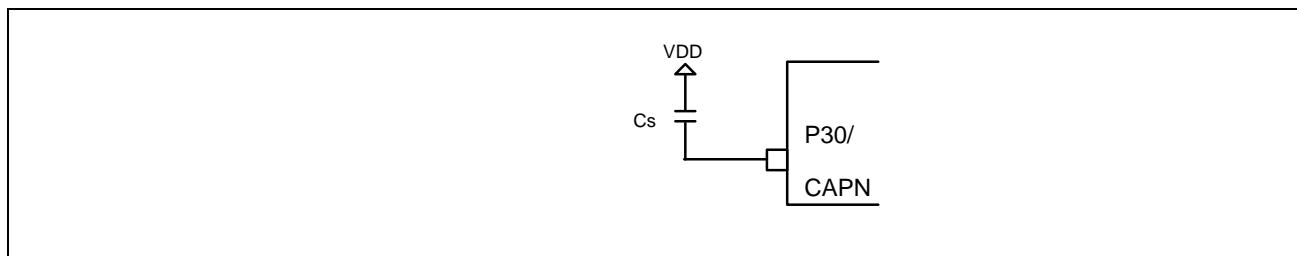


Figure 115. CAPN Pin with  $C_s$  Capacitor

## 16.4 Register Map

**Table 40. Touch Register Map**

Name	Address	Direction	Default	Description
SUM_CH0_F0_H	2E00H	R	00H	Touch Sensor Ch0 Freq0 Sum High-Byte Register
SUM_CH0_F0_L	2E01H	R	00H	Touch Sensor Ch0 Freq0 Sum Low-Byte Register
SUM_CH0_F1_H	2E02H	R	00H	Touch Sensor Ch0 Freq1 Sum High-Byte Register
SUM_CH0_F1_L	2E03H	R	00H	Touch Sensor Ch0 Freq1 Sum Low-Byte Register
...				
SUM_CH19_F0_H	2E4CH	R	00H	Touch Sensor Ch19 Freq0 Sum High-Byte Register
SUM_CH19_F0_L	2E4DH	R	00H	Touch Sensor Ch19 Freq0 Sum Low-Byte Register
SUM_CH19_F1_H	2E4EH	R	00H	Touch Sensor Ch19 Freq1 Sum High-Byte Register
SUM_CH19_F1_L	2E4FH	R	00H	Touch Sensor Ch19 Freq1 Sum Low-Byte Register
SCO0_H	2E50H	R/W	00H	Touch Sensor Offset Capacitor Selection High-Byte Register for Ch0
SCO0_L	2E51H	R/W	00H	Touch Sensor Offset Capacitor Selection Low-Byte Register for Ch0
...				
SCO19_H	2E76H	R/W	00H	Touch Sensor Offset Capacitor Selection High-Byte Register for Ch19
SCO19_L	2E77H	R/W	00H	Touch Sensor Offset Capacitor Selection Low-Byte Register for Ch19
TS_CON	2E80H	R/W	00H	Touch Sensor Control Register
TS_MODE	2E81H	R/W	00H	Touch Sensor Mode Register
TS_SUM_CNT	2E82H	R/W	01H	Touch Sensor Sum Repeat Count Register
TS_CH_SELH	2E83H	R/W	00H	Touch Sensor Channel Selection High-Byte Register
TS_CH_SELM	2E84H	R/W	00H	Touch Sensor Channel Selection Middle-Byte Register
TS_CH_SELL	2E85H	R/W	00H	Touch Sensor Channel Selection Low-Byte Register
TS_S1_WIDH	2E86H	R/W	15H	S1 Minimum Time Register
TS_SLP_CON	2E87H	R/W	74H	Touch Sensor Low Pass Filter Control Register
Reserved	2E88H			
TS_TRIM	2E89B	R/W	07H	Touch Sensor Trimming Register
TS_CLK_CFG	2E8AH	R/W	02H	Touch Sensor Clock Configuration Register
TRIM_OSC	2E8BH	R/W	20H	Touch Sensor RING Oscillator Trimming Selection Register
DELTA_OSC	2E8CH	R/W	01H	Touch Sensor RING Oscillator Delta Register
TLED	2E8DH	R/W	30H	LED stable time Register
TS_VHS_H	2E8EH	R/W	02H	Touch Sensor High Sense Voltage High-Byte Register
TS_VHS_L	2E8FH	R/W	80H	Touch Sensor High Sense Voltage Low-Byte Register
TS_VREF_H	2F90H	R/W	02H	Touch Sensor COMP Reference Voltage High-Byte Register
TS_VREF_L	2F91H	R/W	10H	Touch Sensor COMP Reference Voltage Low-Byte Register

## 16.5 Register Description for Touch Sensing

**SUM\_CH0\_F0\_H (Touch Sensor Ch0 Freq0 Sum High-Byte Register):** **2E00H**

7	6	5	4	3	2	1	0
SUM_CH0_F0_H							
R	R	R	R	R	R	R	R

Initial value: 00H

**SUM\_CH0\_F0\_L (Touch Sensor Ch0 Freq0 Sum Low-Byte Register):** **2E01H**

7	6	5	4	3	2	1	0
SUM_CH0_F0_L							
R	R	R	R	R	R	R	R

Initial value: 00H

**SUM\_CH0\_F1\_H (Touch Sensor Ch0 Freq1 Sum High-Byte Register):** **2E02H**

**SUM\_CH0\_F1\_L (Touch Sensor Ch0 Freq1 Sum Low-Byte Register):** **2E03H**

**SUM\_CH1\_F0\_H (Touch Sensor Ch1 Freq0 Sum High-Byte Register):** **2E04H**

**SUM\_CH1\_F0\_L (Touch Sensor Ch1 Freq0 Sum Low-Byte Register):** **2E05H**

**SUM\_CH1\_F1\_H (Touch Sensor Ch1 Freq1 Sum High-Byte Register):** **2E06H**

**SUM\_CH1\_F1\_L (Touch Sensor Ch1 Freq1 Sum Low-Byte Register):** **2E07H**

**SUM\_CH2\_F0\_H (Touch Sensor Ch2 Freq0 Sum High-Byte Register):** **2E08H**

**SUM\_CH2\_F0\_L (Touch Sensor Ch2 Freq0 Sum Low-Byte Register):** **2E09H**

**SUM\_CH2\_F1\_H (Touch Sensor Ch2 Freq1 Sum High-Byte Register):** **2E0AH**

**SUM\_CH2\_F1\_L (Touch Sensor Ch2 Freq1 Sum Low-Byte Register):** **2E0BH**

**SUM\_CH3\_F0\_H (Touch Sensor Ch3 Freq0 Sum High-Byte Register):** **2E0CH**

**SUM\_CH3\_F0\_L (Touch Sensor Ch3 Freq0 Sum Low-Byte Register):** **2E0DH**

**SUM\_CH3\_F1\_H (Touch Sensor Ch3 Freq1 Sum High-Byte Register):** **2E0EH**

**SUM\_CH3\_F1\_L (Touch Sensor Ch3 Freq1 Sum Low-Byte Register):** **2E0FH**

**SUM\_CH3\_F0\_H (Touch Sensor Ch3 Freq0 Sum High-Byte Register):** **2E0CH**

**SUM\_CH3\_F0\_L (Touch Sensor Ch3 Freq0 Sum Low-Byte Register):** **2E0DH**

**SUM\_CH3\_F1\_H (Touch Sensor Ch3 Freq1 Sum High-Byte Register):** **2E0EH**

**SUM\_CH3\_F1\_L (Touch Sensor Ch3 Freq1 Sum Low-Byte Register):** **2E0FH**

**SUM\_CH4\_F0\_H (Touch Sensor Ch4 Freq0 Sum High-Byte Register):** **2E10H**

**SUM\_CH4\_F0\_L (Touch Sensor Ch4 Freq0 Sum Low-Byte Register):** **2E11H**

**SUM\_CH4\_F1\_H (Touch Sensor Ch4 Freq1 Sum High-Byte Register):** **2E12H**

**SUM\_CH4\_F1\_L (Touch Sensor Ch4 Freq1 Sum Low-Byte Register):** **2E13H**

**SUM\_CH5\_F0\_H (Touch Sensor Ch5 Freq0 Sum High-Byte Register):** **2E14H**

<b>SUM_CH5_F0_L (Touch Sensor Ch5 Freq0 Sum Low-Byte Register):</b>	<b>2E15H</b>
<b>SUM_CH5_F1_H (Touch Sensor Ch5 Freq1 Sum High-Byte Register):</b>	<b>2E16H</b>
<b>SUM_CH5_F1_L (Touch Sensor Ch5 Freq1 Sum Low-Byte Register):</b>	<b>2E17H</b>
<b>SUM_CH6_F0_H (Touch Sensor Ch6 Freq0 Sum High-Byte Register):</b>	<b>2E18H</b>
<b>SUM_CH6_F0_L (Touch Sensor Ch6 Freq0 Sum Low-Byte Register):</b>	<b>2E19H</b>
<b>SUM_CH6_F1_H (Touch Sensor Ch6 Freq1 Sum High-Byte Register):</b>	<b>2E1AH</b>
<b>SUM_CH6_F1_L (Touch Sensor Ch6 Freq1 Sum Low-Byte Register):</b>	<b>2E1BH</b>
<b>SUM_CH7_F0_H (Touch Sensor Ch7 Freq0 Sum High-Byte Register):</b>	<b>2E1CH</b>
<b>SUM_CH7_F0_L (Touch Sensor Ch7 Freq0 Sum Low-Byte Register):</b>	<b>2E1DH</b>
<b>SUM_CH7_F1_H (Touch Sensor Ch7 Freq1 Sum High-Byte Register):</b>	<b>2E1EH</b>
<b>SUM_CH7_F1_L (Touch Sensor Ch7 Freq1 Sum Low-Byte Register):</b>	<b>2E1FH</b>
<b>SUM_CH8_F0_H (Touch Sensor Ch8 Freq0 Sum High-Byte Register):</b>	<b>2E20H</b>
<b>SUM_CH8_F0_L (Touch Sensor Ch8 Freq0 Sum Low-Byte Register):</b>	<b>2E21H</b>
<b>SUM_CH8_F1_H (Touch Sensor Ch8 Freq1 Sum High-Byte Register):</b>	<b>2E22H</b>
<b>SUM_CH8_F1_L (Touch Sensor Ch8 Freq1 Sum Low-Byte Register):</b>	<b>2E23H</b>
<b>SUM_CH9_F0_H (Touch Sensor Ch9 Freq0 Sum High-Byte Register):</b>	<b>2E24H</b>
<b>SUM_CH9_F0_L (Touch Sensor Ch9 Freq0 Sum Low-Byte Register):</b>	<b>2E25H</b>
<b>SUM_CH9_F1_H (Touch Sensor Ch9 Freq1 Sum High-Byte Register):</b>	<b>2E26H</b>
<b>SUM_CH9_F1_L (Touch Sensor Ch9 Freq1 Sum Low-Byte Register):</b>	<b>2E27H</b>
<b>SUM_CH10_F0_H (Touch Sensor Ch10 Freq0 Sum High-Byte Register):</b>	<b>2E28H</b>
<b>SUM_CH10_F0_L (Touch Sensor Ch10 Freq0 Sum Low-Byte Register):</b>	<b>2E29H</b>
<b>SUM_CH10_F1_H (Touch Sensor Ch10 Freq1 Sum High-Byte Register):</b>	<b>2E2AH</b>
<b>SUM_CH10_F1_L (Touch Sensor Ch10 Freq1 Sum Low-Byte Register):</b>	<b>2E2BH</b>
<b>SUM_CH11_F0_H (Touch Sensor Ch11 Freq0 Sum High-Byte Register):</b>	<b>2E2CH</b>
<b>SUM_CH11_F0_L (Touch Sensor Ch11 Freq0 Sum Low-Byte Register):</b>	<b>2E2DH</b>
<b>SUM_CH11_F1_H (Touch Sensor Ch11 Freq1 Sum High-Byte Register):</b>	<b>2E2EH</b>
<b>SUM_CH11_F1_L (Touch Sensor Ch11 Freq1 Sum Low-Byte Register):</b>	<b>2E2FH</b>
<b>SUM_CH12_F0_H (Touch Sensor Ch12 Freq0 Sum High-Byte Register):</b>	<b>2E30H</b>
<b>SUM_CH12_F0_L (Touch Sensor Ch12 Freq0 Sum Low-Byte Register):</b>	<b>2E31H</b>
<b>SUM_CH12_F1_H (Touch Sensor Ch12 Freq1 Sum High-Byte Register):</b>	<b>2E32H</b>
<b>SUM_CH12_F1_L (Touch Sensor Ch12 Freq1 Sum Low-Byte Register):</b>	<b>2E33H</b>
<b>SUM_CH13_F0_H (Touch Sensor Ch13 Freq0 Sum High-Byte Register):</b>	<b>2E34H</b>
<b>SUM_CH13_F0_L (Touch Sensor Ch13 Freq0 Sum Low-Byte Register):</b>	<b>2E35H</b>

SUM_CH13_F1_H (Touch Sensor Ch13 Freq1 Sum High-Byte Register):	2E36H
SUM_CH13_F1_L (Touch Sensor Ch13 Freq1 Sum Low-Byte Register):	2E37H
SUM_CH14_F0_H (Touch Sensor Ch14 Freq0 Sum High-Byte Register):	2E38H
SUM_CH14_F0_L (Touch Sensor Ch14 Freq0 Sum Low-Byte Register):	2E39H
SUM_CH14_F1_H (Touch Sensor Ch14 Freq1 Sum High-Byte Register):	2E3AH
SUM_CH14_F1_L (Touch Sensor Ch14 Freq1 Sum Low-Byte Register):	2E3BH
SUM_CH15_F0_H (Touch Sensor Ch15 Freq0 Sum High-Byte Register):	2E3CH
SUM_CH15_F0_L (Touch Sensor Ch15 Freq0 Sum Low-Byte Register):	2E3DH
SUM_CH15_F1_H (Touch Sensor Ch15 Freq1 Sum High-Byte Register):	2E3EH
SUM_CH15_F1_L (Touch Sensor Ch15 Freq1 Sum Low-Byte Register):	2E3FH
SUM_CH16_F0_H (Touch Sensor Ch16 Freq0 Sum High-Byte Register):	2E40H
SUM_CH16_F0_L (Touch Sensor Ch16 Freq0 Sum Low-Byte Register):	2E41H
SUM_CH16_F1_H (Touch Sensor Ch16 Freq1 Sum High-Byte Register):	2E42H
SUM_CH16_F1_L (Touch Sensor Ch16 Freq1 Sum Low-Byte Register):	2E43H
SUM_CH17_F0_H (Touch Sensor Ch17 Freq0 Sum High-Byte Register):	2E44H
SUM_CH17_F0_L (Touch Sensor Ch17 Freq0 Sum Low-Byte Register):	2E45H
SUM_CH17_F1_H (Touch Sensor Ch17 Freq1 Sum High-Byte Register):	2E46H
SUM_CH17_F1_L (Touch Sensor Ch17 Freq1 Sum Low-Byte Register):	2E47H
SUM_CH18_F0_H (Touch Sensor Ch18 Freq0 Sum High-Byte Register):	2E48H
SUM_CH18_F0_L (Touch Sensor Ch18 Freq0 Sum Low-Byte Register):	2E49H
SUM_CH18_F1_H (Touch Sensor Ch18 Freq1 Sum High-Byte Register):	2E4AH
SUM_CH18_F1_L (Touch Sensor Ch18 Freq1 Sum Low-Byte Register):	2E4BH
SUM_CH19_F0_H (Touch Sensor Ch19 Freq0 Sum High-Byte Register):	2E4CH
SUM_CH19_F0_L (Touch Sensor Ch19 Freq0 Sum Low-Byte Register):	2E4DH
SUM_CH19_F1_H (Touch Sensor Ch19 Freq1 Sum High-Byte Register):	2E4EH
SUM_CH19_F1_L (Touch Sensor Ch19 Freq1 Sum Low-Byte Register):	2E4FH

**SCO0\_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH0): 2E50H**

7	6	5	4	3	2	1	0
SCO0_H							RW
Initial value: 00H							

**SCO0\_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH0): 2E51H**

7	6	5	4	3	2	1	0
SCO0_L							
RW							RW

Initial value: 00H

**SCO1\_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH1): 2E52H**

**SCO1\_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH1): 2E53H**

**SCO2\_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH2): 2E54H**

**SCO2\_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH2): 2E55H**

**SCO3\_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH3): 2E56H**

**SCO3\_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH3): 2E57H**

**SCO4\_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH4): 2E58H**

**SCO4\_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH4): 2E59H**

**SCO5\_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH5): 2E5AH**

**SCO5\_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH5): 2E5BH**

**SCO6\_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH6): 2E5CH**

**SCO6\_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH6): 2E5DH**

**SCO7\_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH7): 2E5EH**

**SCO7\_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH7): 2E5FH**

**SCO8\_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH8): 2E60H**

**SCO8\_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH8): 2E61H**

**SCO9\_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH9): 2E62H**

**SCO9\_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH9): 2E63H**

**SCO10\_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH10): 2E64H**

**SCO10\_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH10): 2E65H**

**SCO11\_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH11): 2E66H**

**SCO11\_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH11): 2E67H**

**SCO12\_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH12): 2E68H**

**SCO12\_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH12): 2E69H**

**SCO13\_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH13): 2E6AH**

**SCO13\_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH13): 2E6BH**

**SCO14\_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH14): 2E6CH**

**SCO14\_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH14): 2E6DH**

**SCO15\_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH15): 2E6EH**

**SCO15\_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH15): 2E6FH**

**SCO16\_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH16): 2E70H**

**SCO16\_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH16): 2E71H**

**SCO17\_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH17): 2E72H**

**SCO17\_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH17): 2E73H**

**SCO18\_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH18): 2E74H**

**SCO18\_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH18): 2E75H**

**SCO19\_H (Touch Sensor Offset Capacitor Selection High-Byte Register for CH19): 2E76H**

**SCO19\_L (Touch Sensor Offset Capacitor Selection Low-Byte Register for CH19): 2E77H**

**TS\_CON (Touch Sensor Control Register): 2E80H**

7	6	5	4	3	2	1	0
-	-	-	-	-	TS_IF	-	TS_RUN
-	-	-	-	-	RW	-	RW

Initial value: 00H

**TS\_IF** Touch Sensor Interrupt Flag. To start the new sensing, you must clear this bit after reading the SUM registers. If this bit is changed from 1 to 0, the SUM registers will be reset to 0s.

0 No new sensing results

1 In normal mode, this flag indicates that the new sensing results are generated.

**TS\_RUN** Touch Sensor Enable

0 Touch Sensor Disable (Default)

1 Touch Sensor Enable

**TS\_MODE (Touch Sensor Mode Register): 2E81H**

7	6	5	4	3	2	1	0
-	SC_GAIN	S1_SWEEP_DIS	MODE2	MODE1	MODE0	PORT1	PORT0
-	RW	RW	RW	-	RW	RW	RW

Initial value: 00H

SC_GAIN	Gain Calibration Capacitor Enable
0	Disable
1	Enable
S1_SWEEP_DIS	S1 width sweep disable
0	Sweep enable (Default)
1	Sweep disable
MODE[2:0]	Touch Sensor mode
000	Normal sensing mode
001	High sensing mode
010	Channel Adjust sensing mode
011	Offset Calibration sensing mode
100	Simultaneous sensing for all selected channels to reduce the sensing time
POR[T][1:0]	Port Configuration During Inactive Status
00	Input Floating
01	Output Low
10	Output High

**TS\_SUM\_CNT (Touch Sensor Sum Repeat Count Register): 2E82H**

7	6	5	4	3	2	1	0
TS_SUM_CNT							
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 01H

**TS\_CH\_SEL\_H (Touch Sensor Channel Selection High-Byte Register): 2E83H**

7	6	5	4	3	2	1	0
-	-	-	-	CH19_SEL	CH18_SEL	CH17_SEL	CH16_SEL
-	-	-	-	RW	RW	RW	RW

Initial value: 00H

CH19_SEL	Channel 19 Enable
0	Disable (Default)
1	Enable Touch Key by P03
CH18_SEL	Channel 18 Enable
0	Disable (Default)
1	Enable Touch Key by P00
CH17_SEL	Channel 17 Enable
0	Disable (Default)
1	Enable Touch Key by P31
CH16_SEL	Channel 16 Enable
0	Disable (Default)
1	Enable Touch Key by P27

**TS\_CH\_SEL\_M (Touch Sensor Channel Selection Middle-Byte Register): 2E84H**

7	6	5	4	3	2	1	0
CH15_SEL	CH14_SEL	CH13_SEL	CH12_SEL	CH11_SEL	CH10_SEL	CH9_SEL	CH8_SEL
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

CH15_SEL	Channel 15 Enable
0	Disable (Default)
1	Enable Touch Key by P26
CH14_SEL	Channel 14 Enable
0	Disable (Default)
1	Enable Touch Key by P25
CH13_SEL	Channel 13 Enable
0	Disable (Default)
1	Enable Touch Key by P24
CH12_SEL	Channel 12 Enable
0	Disable (Default)
1	Enable Touch Key by P23
CH11_SEL	Channel 11 Enable
0	Disable (Default)
1	Enable Touch Key by P22
CH10_SEL	Channel 10 Enable
0	Disable (Default)
1	Enable Touch Key by P17
CH9_SEL	Channel 9 Enable
0	Disable (Default)
1	Enable Touch Key by P16
CH8_SEL	Channel 8 Enable
0	Disable (Default)
1	Enable Touch Key by P15

**TS\_CH\_SEL\_L (Touch Sensor Channel Selection Low-Byte Register): 2E85H**

7	6	5	4	3	2	1	0
CH7_SEL	CH6_SEL	CH5_SEL	CH4_SEL	CH3_SEL	CH2_SEL	CH1_SEL	CH0_SEL
RW	RW	R/W	R/W	R/W	R/W	R/W	R/W
Initial value: 00H							
CH7_SEL      Channel 7 Enable 0      Disable (Default) 1      Enable Touch Key by P14							
CH6_SEL      Channel 6 Enable 0      Disable (Default) 1      Enable Touch Key by P12							
CH5_SEL      Channel 5 Enable 0      Disable (Default) 1      Enable Touch Key by P11							
CH4_SEL      Channel 4 Enable 0      Disable (Default) 1      Enable Touch Key by P10							
CH3_SEL      Channel 3 Enable 0      Disable (Default) 1      Enable Touch Key by P07							
CH2_SEL      Channel 2 Enable 0      Disable (Default) 1      Enable Touch Key by P06							
CH1_SEL      Channel 1 Enable 0      Disable (Default) 1      Enable Touch Key by P05							
CH0_SEL      Channel 0 Enable 0      Disable (Default) 1      Enable Touch Key by P04							

**TS\_S1\_WIDTH (S1 Timing Register): 2E86H**

7	6	5	4	3	2	1	0
TS_S1_WIDTH							
RW	R/W						
Initial value: 20H							

Actually, S1 width will be changed randomly from 2\*TS\_S1\_WIDTH to 4\*TS\_S1\_WIDTH.

**TS\_SLP\_CON (Touch Sensor Low Pass Filter Control Register): 2E87H**

7	6	5	4	3	2	1	0
SLP_C3	SLP_C2	SLP_C1	SLP_C0	SLP_R3	SLP_R2	SLP_R1	SLP_R0
	RW						

Initial value: 74H

SLP\_C[3:0] Capacitor Trimming for Input Low Pass Filter

- 0000 0.0pF
- 0001 1.5pF
- 0010 3.0pF
- 0011 4.5pF
- 0100 6.0pF
- 0101 7.5pF
- 0110 9.0pF
- 0111 10.5pF (Default)
- 1000 12.0pF
- 1001 13.5pF
- 1010 15.0pF
- 1011 16.5pF
- 1100 18.0pF
- 1101 19.5pF
- 1110 21.0pF
- 1111 22.5pF

SLP\_R[3:0] Resistor Trimming for Input Low Pass Filter

- 0000 Open
- 0001 0K
- 0010 5K
- 0100 100K (Default)
- 1000 200K
- 0110 33K
- 1100 67K
- 1010 40K
- 1110 28K

**TS\_TRIM (Touch Sensor Trimming Register): 2E89H**

7	6	5	4	3	2	1	0
-	-	-	HC	IB_TRIM3	IB_TRIM2	IB_TRIM1	IB_TRIM0
-	-	-	R/W	R/W	R/W	R/W	R/W

Initial value: 07H

HC	High Current Enable
0	Disable
1	Enable
IB_TRIM[3:0]	Current Bias Trimming Value
0000	1uA(N)
0001	1.5uA(N)
0010	2uA(N)
0011	2.5uA(N)
0100	3uA(N)
0101	3.5uA(N)
0110	4uA(N)
0111	4.5uA(N) (Default)
1000	5uA(N)
1001	5.5uA(N)
1010	6uA(N)
1011	6.5uA(N)
1100	7uA(N)
1101	7.5uA(N)
1110	8uA(N)
1111	8.5uA(N)

**TS\_CLK\_CFG (Touch Sensor Clock Configuration Register) : 2E8AH**

7	6	5	4	3	2	1	0
SCLK_EN	PAT_EN	CLKSEL1	CLKSEL0	TSCLKOE	TSCLKDIV2	TSCLKDIV1	TSCLKDIV0
RW	RW	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 02H

SCLK_EN	Enable or disable system clock as touch clock
0	Disable (Default)
1	Enable
PAT_EN	Enable or disable of using pattern for S2 & S3
0	Disable (Default)
1	Enable
CLKSEL[1:0]	Touch Clock source select
00	Two touch sensor clocks will be used
01	Touch clock A only
1x	Touch clock B only
TSCLKOE	Divided Touch Sensor Clock Output Enable
0	Clock Output Disable (Default)
1	Clock Output Enable
TSCLKDIV[2:0]	Touch Sensor Clock Divider (Refer to TRIM_OSC)
000	$OSC_{ts} / 1$ (16MHz, Default)
001	$OSC_{ts} / 2$
010	$OSC_{ts} / 4$
011	$OSC_{ts} / 8$
100	$OSC_{ts} / 16$
101	$OSC_{ts} / 32$
110	$OSC_{ts} / 64$
111	$OSC_{ts} / 128$

**TRIM\_OSC (Touch Sensor RING Oscillator Trimming Selection Register): 2E8BH**

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

Initial value: 20H

TRIM_OSC[5:0]	Touch Sensor RING Oscillator Trimming Selection
3FH	+23.25% (19.72MHz)
3EH	+22.50%
...	...
21H	+0.75%
20H	16MHz (default)
1FH	-0.75%
...	...
01H	-23.25%
00H	-24.00% (12.16MHz)

Frequency of Touch clock B = 16MHz + (TRIM\_OSC – 20H) \* 0.75%

\*Frequency = 16MHz + 16MHz \* ((TRIM\_OSC – 20H) \* 0.0075)

**DELTA\_OSC (Touch Sensor RING Oscillator Delta Register): 2E8CH**

7	6	5	4	3	2	1	0
DELTA_OSC							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 01H

Frequency of Touch clock A = 16MHz + ((TRIM\_OSC + DELTA\_OSC) – 20H) \* 0.75%

\*Frequency = 16MHz + 16MHz \* (((TRIM\_OSC + DELTA\_OSC) – 20H) \* 0.0075)

**TLED (LED stable time Register) : 2E8DH**

7	6	5	4	3	2	1	0
TLED							
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

To run touch sensing again after LED working, we need 25~30us stable time. This counter is working based on system clock.

**TS\_VHS\_H (Touch Sensor High Sense Voltage High-Byte Register): 2E8EH**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	TS_VHS_H	
-	-	-	-	-	-	RW	RW

Initial value: 02H

**TS\_VHS\_L (Touch Sensor High Sense Voltage Low-Byte Register): 2E8FH**

7	6	5	4	3	2	1	0
TS_VHS_L							
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 80H

**TS\_VREF\_H (Touch Sensor COMP Reference Voltage High-Byte Register): 2E90H**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	TS_VREF_H	
-	-	-	-	-	-	RW	RW

Initial value: 02H

**TS\_VREF\_L(Touch Sensor COMP Reference Voltage Low-Byte Register): 2E91H**

7	6	5	4	3	2	1	0
TS_VREF_L							
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 10H

## 16.6 User Programming Procedure

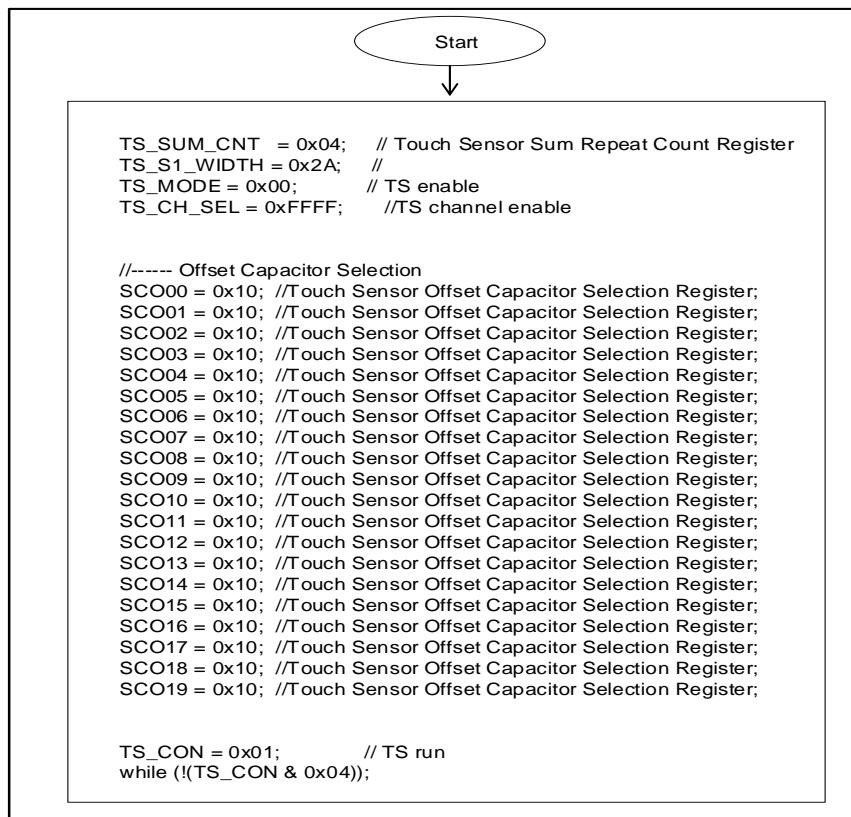


Figure 116. User Programming Procedure

## 17 12-bit A/D Converter

### 17.1 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 eight 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[2:0]. 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'.

### 17.2 Conversion Timing

The A/D conversion process requires 4 steps (4 clock edges) to convert each bit and 12 clocks to set up A/D conversion. Therefore, total of 48 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  $\mu$ s. Each bit conversion requires 4 clocks, the conversion rate is calculated as follows:

$$4 \text{ clocks/bit} \times 12 \text{ bits} + \text{set-up time} = 60 \text{ clocks}$$

$$\text{ADC Conversion Time} = \text{ADCLK} * 60 \text{ cycles}$$

#### NOTES:

1. The A/D converter needs at least 7.5us for conversion time. So you must set the conversion time more than 7.5us.

### 17.3 Block Diagram

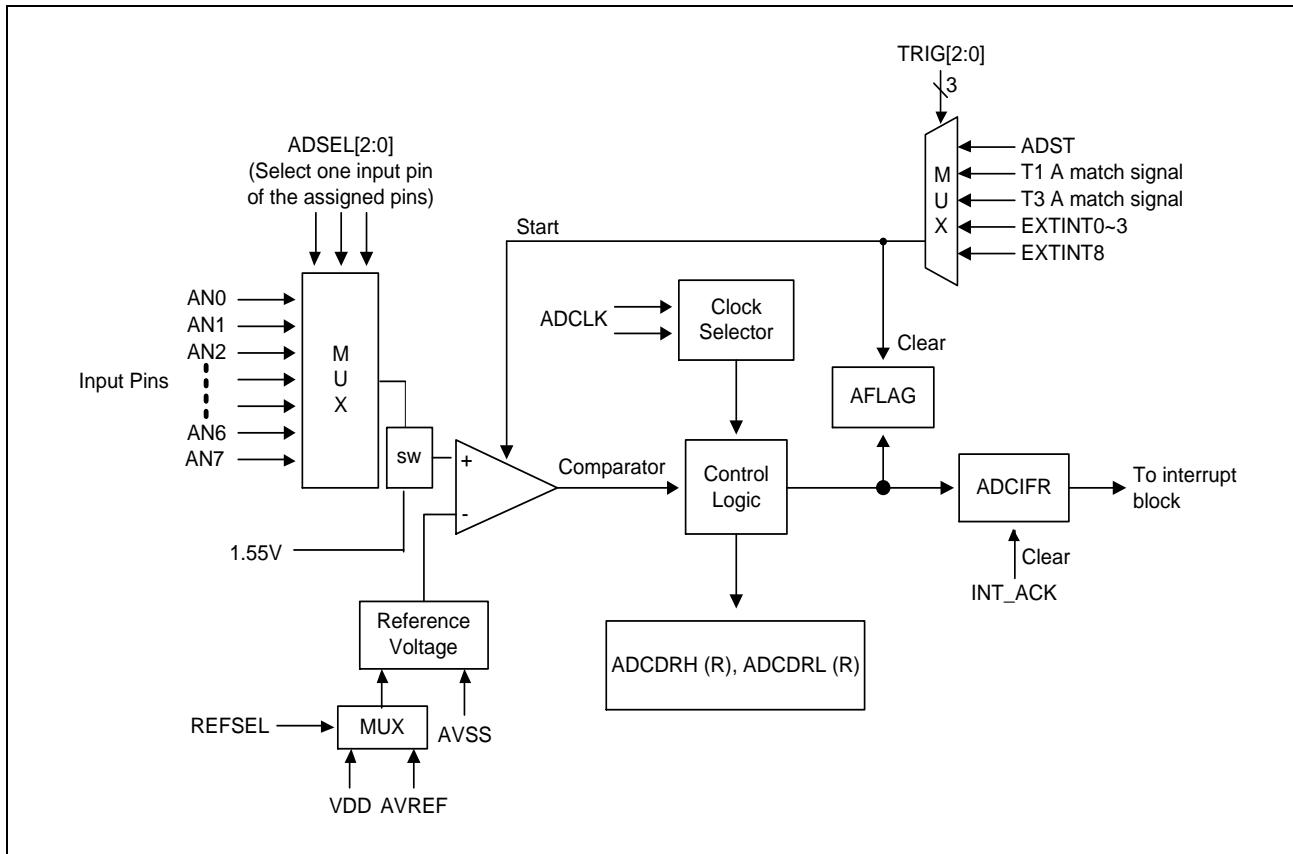


Figure 117. 12-bit ADC Block Diagram

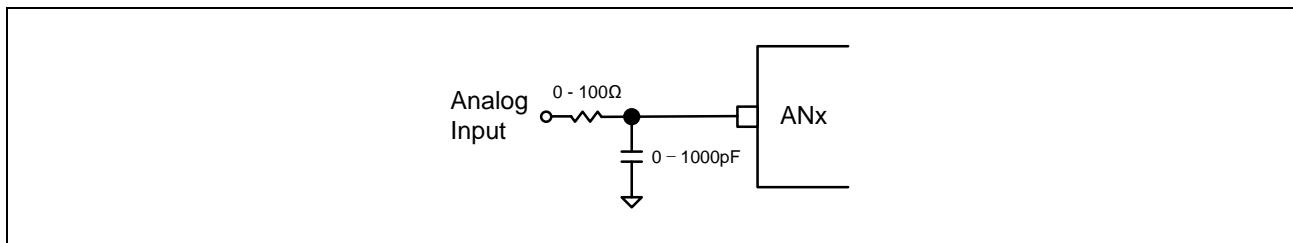


Figure 118. A/D Analog Input Pin with Capacitor

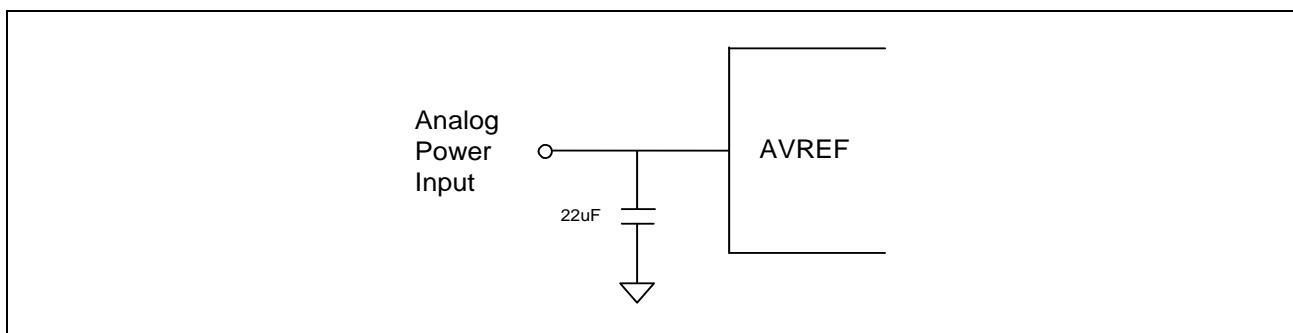


Figure 119. A/D Power (AVREF) Pin with Capacitor

## 17.4 ADC Operation

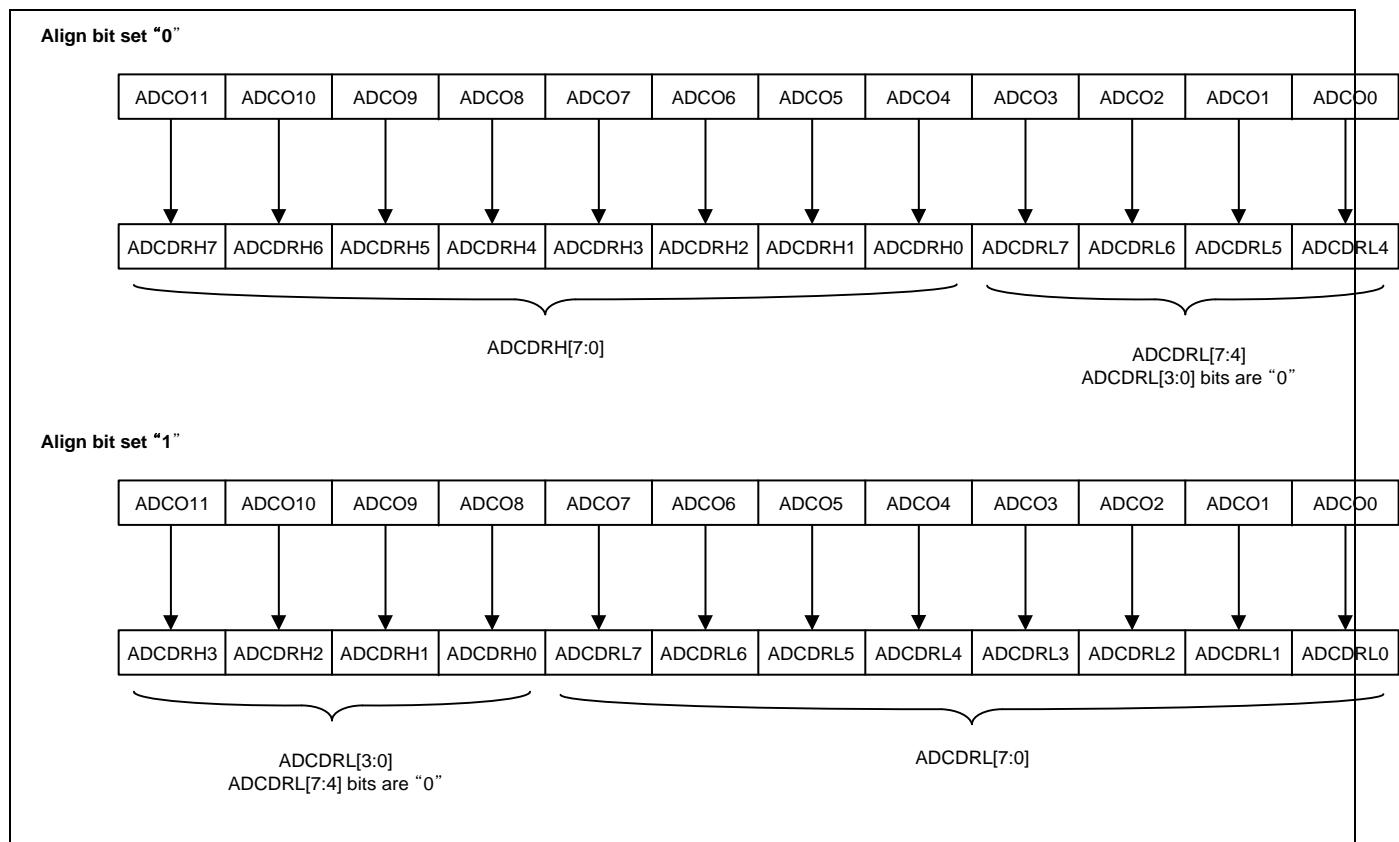


Figure 120. ADC Operation for Align Bit

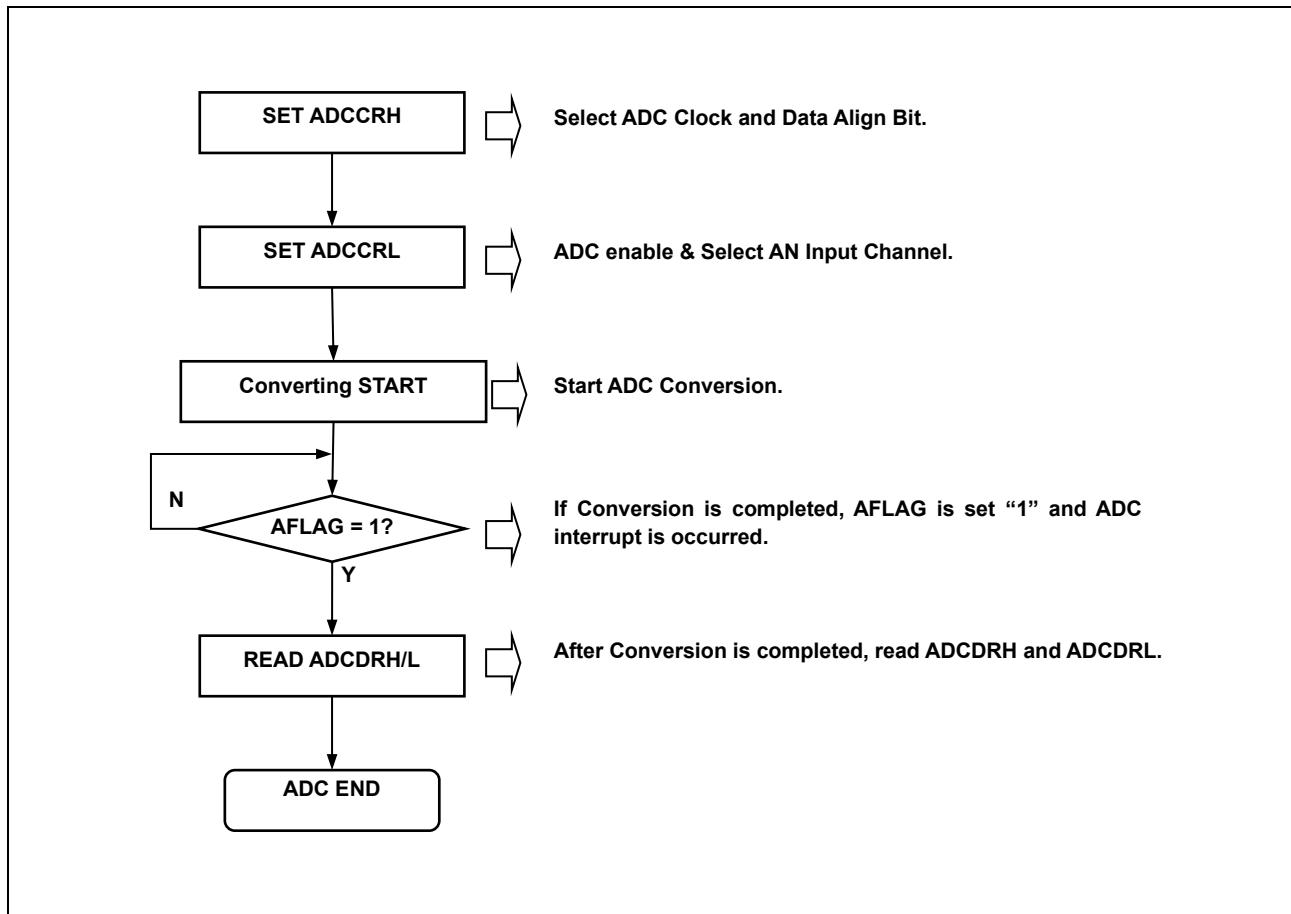


Figure 121. A/D Converter Operation Flow

## 17.5 Register Map

Table 41. ADC Register Map

Name	Address	Direction	Default	Description
ADCDRH	9FH	R	00H	A/D Converter Data High Register
ADCDRL	9EH	R	00H	A/D Converter Data Low Register
ADCCRH	9DH	R/W	00H	A/D Converter Control High Register
ADCCRL	9CH	R/W	00H	A/D Converter Control Low Register

## 17.6 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).

## 17.7 Register Description for ADC

### ADCDRH (A/D Converter Data High Register): 9FH

7	6	5	4	3	2	1	0
ADDM11	ADDM10	ADDM9	ADDM8	ADDM7 ADDL11	ADDM6 ADDL10	ADDM5 ADDL9	ADDM4 ADDL8
R	R	R	R	R	R	R	R

Initial value: xxH

ADDM[11:4] MSB align, A/D Converter High Data (8-bit)

ADDL[11:8] LSB align, A/D Converter High Data (4-bit)

### ADCDRL (A/D Converter Data Low Register): 9EH

7	6	5	4	3	2	1	0
ADDM3	ADDM2	ADDM1 ADDL5	ADDM0 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 Data (4-bit)

ADDL[7:0] LSB align, A/D Converter Low Data (8-bit)

**ADCCRH (A/D Converter High Register): 9DH**

7	6	5	4	3	2	1	0
ADCIFR	-	TRIG2	TRIG1	TRIG0	ALIGN	CKSEL1	CKSEL0
RW	RW	R/W	R/W	R/W	R/W	RW	RW

Initial value: 01H

ADCIFR	When ADC interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or auto clear by INT_ACK signal. Writing "1" has no effect.
	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 1 A match signal
	0        1        0        Timer 3 A match signal
	0        1        1        EXTINT0~3
	1        0        0        EXTINT8
	1        0        1        Not used
	Other Values                          Not used
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 (default)
	1        0        fx/4
	1        1        fx/8

**NOTES:** fx : system clock

ADC clock should use below 8MHz

**ADCCRL (A/D Converter Counter Low Register): 9CH**

7	6	5	4	3	2	1	0
STBY	ADST	REFSEL	AFLAG	-	ADSEL2	ADSEL1	ADSEL0
R/W	R/W	R/W	R	-	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 stop/start.
	0      No effect
	1      ADC Conversion Start and auto clear
REFSEL	A/D Converter Reference Selection
	0      Internal Reference (VDD)
	1      External Reference (AVREF)
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[2:0]	A/D Converter input selection
	ADSEL2   ADSEL1   ADSEL0   Description
	0      0      0      AN0
	0      0      1      AN1
	0      1      0      AN2
	0      1      1      AN3
	1      0      0      AN4
	1      0      1      AN5
	1      1      0      AN6
	1      1      1      AN7

## 18 Power Down Operation

### 18.1 Overview

The A96T418 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.

### 18.2 Peripheral Operation in IDLE/STOP Mode

**Table 42. 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 (Can be operated with WDTRC OSC)
Watchdog Timer	Operates Continuously	Stop (Can be operated with WDTRC OSC)
Watch Timer	Operates Continuously	Stop (Can be operated with sub clock)
Timer0~4	Operates Continuously	Halted (Only when the Event Counter Mode is Enabled, Timer operates Normally)
ADC	Operates Continuously	Stop
BUZ	Operates Continuously	Stop
USI0/1	Operates Continuously	Only operate with external clock
Internal OSC (16MHz)	Oscillation	Stop when the system clock (fx) is fHSIRC
WDTRC OSC (128kHz)	Can be operated with setting value	Can be operated programmable
Sub OSC (32.768kHz)	Oscillation	Can be operated programmable
Touch OSC (16MHz)	Oscillation	Stop when the system clock (fx) is fHSIRC
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, EC1, EC3), External Interrupt, USART by RX, WT (sub clock), WDT USI0/1 by RX, I2C (Slave mode)

### 18.3 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 stops. 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.

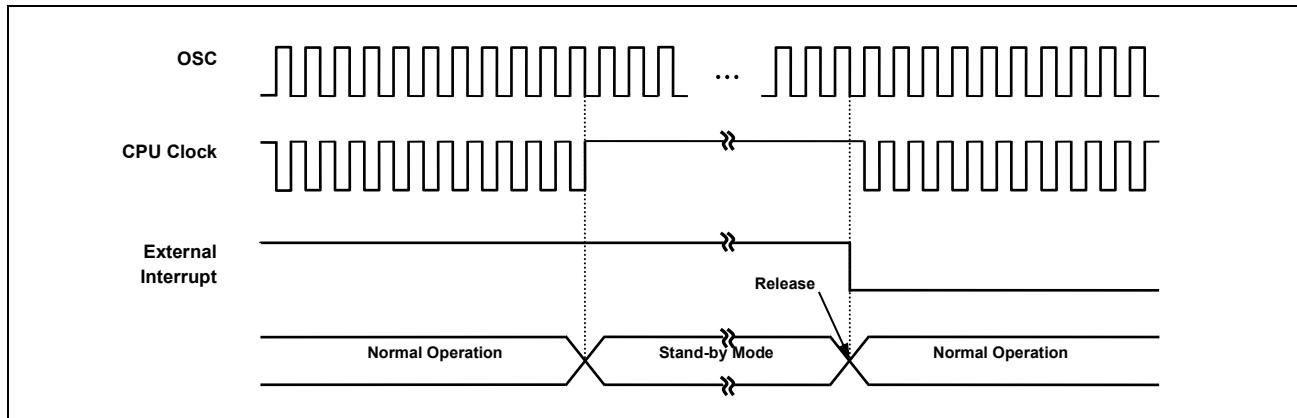


Figure 122. IDLE Mode Release Timing by External Interrupt

## 18.4 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 (fIRC) is selected for the system clock and the sub clock (fSUB) is oscillated, the internal RC oscillator stops oscillation and the sub clock is continuously oscillated in stop mode. At that time, the watch timer 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 123 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.

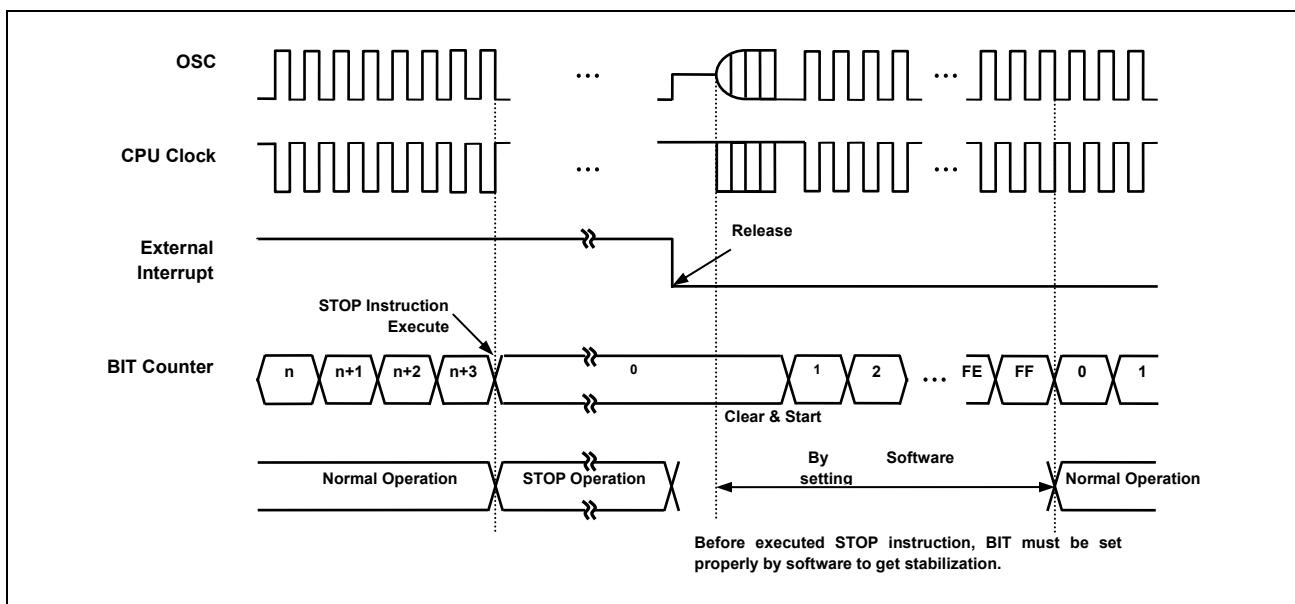


Figure 123. STOP Mode Release Timing by External Interrupt

## 18.5 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 124). 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'.

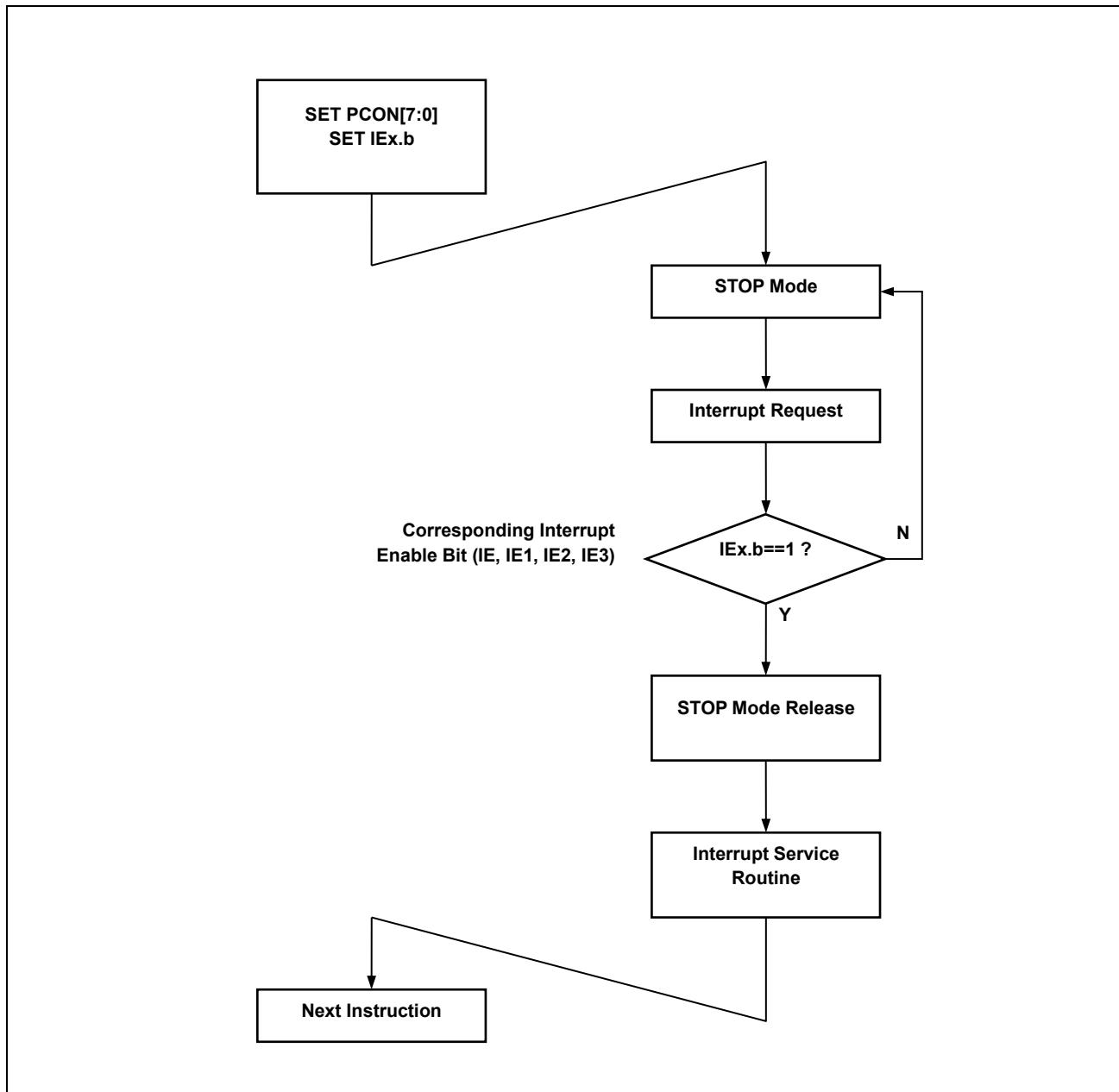


Figure 124. STOP Mode Release Flow

## 18.6 Register Map

**Table 43. Power Down Operation Register Map**

Name	Address	Direction	Default	Description
PCON	87H	R/W	00H	Power Control Register

## 18.7 Power Down Operation Register Description

The power down operation register consists of the power control register (PCON).

## 18.8 Register Description for Power Down Operation

### PCON (Power Control Register): 87H

7	6	5	4	3	2	1	0
PCON7	-	-	-	PCON3	PCON2	PCON1	PCON0
RW	-	-	-	RW	RW	RW	RW
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	Ex2)	MOV PCON, #03H ; STOP mode
	NOP		NOP
	NOP		NOP
	NOP		NOP
	•		•
	•		•
	•		•

## 19 RESET

### 19.1 Overview

The following is the hardware setting value.

**Table 44. Reset State**

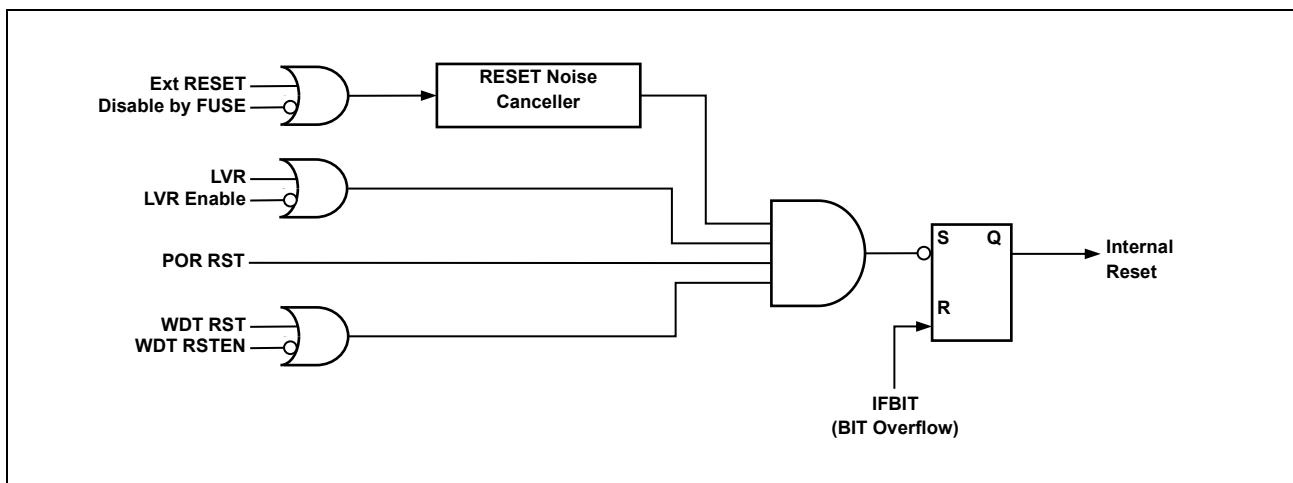
On Chip Hardware	Initial Value
Program Counter (PC)	0000h
Accumulator	00h
Stack Pointer (SP)	07h
Peripheral Clock	On
Control Register	Refer to the Peripheral Registers

### 19.2 Reset Source

The A96T418 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')
- OCD Reset

### 19.3 RESET Block Diagram



**Figure 125. RESET Block Diagram**

## 19.4 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.

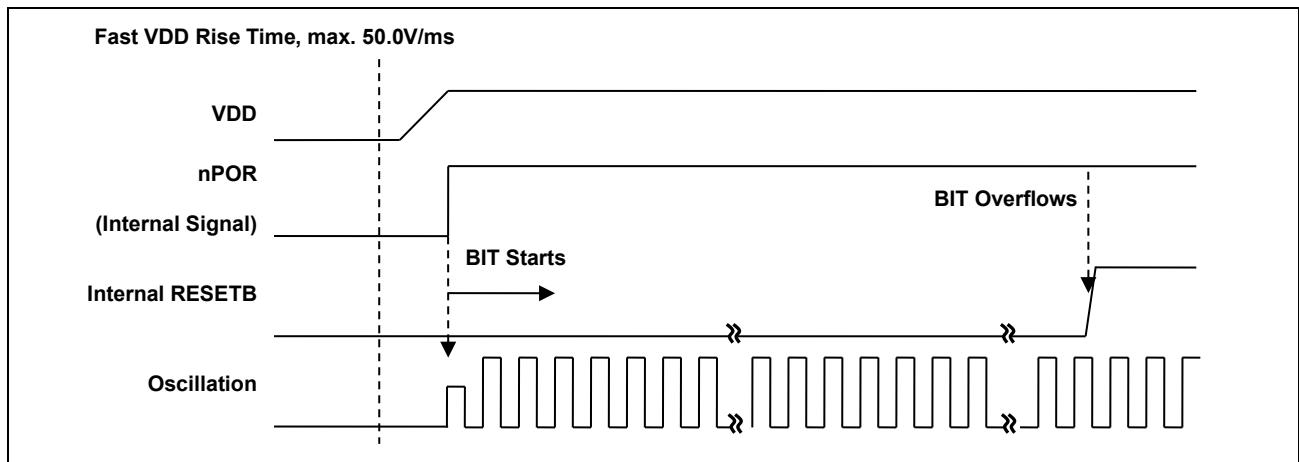


Figure 126. Fast VDD Rising Time

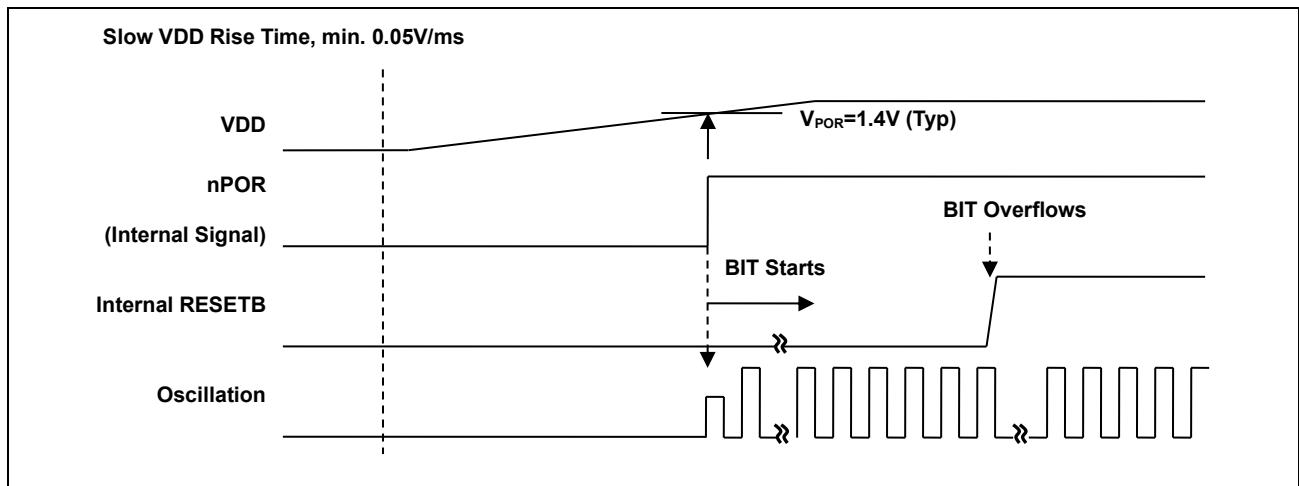


Figure 127. Internal RESET Release Timing On Power-Up

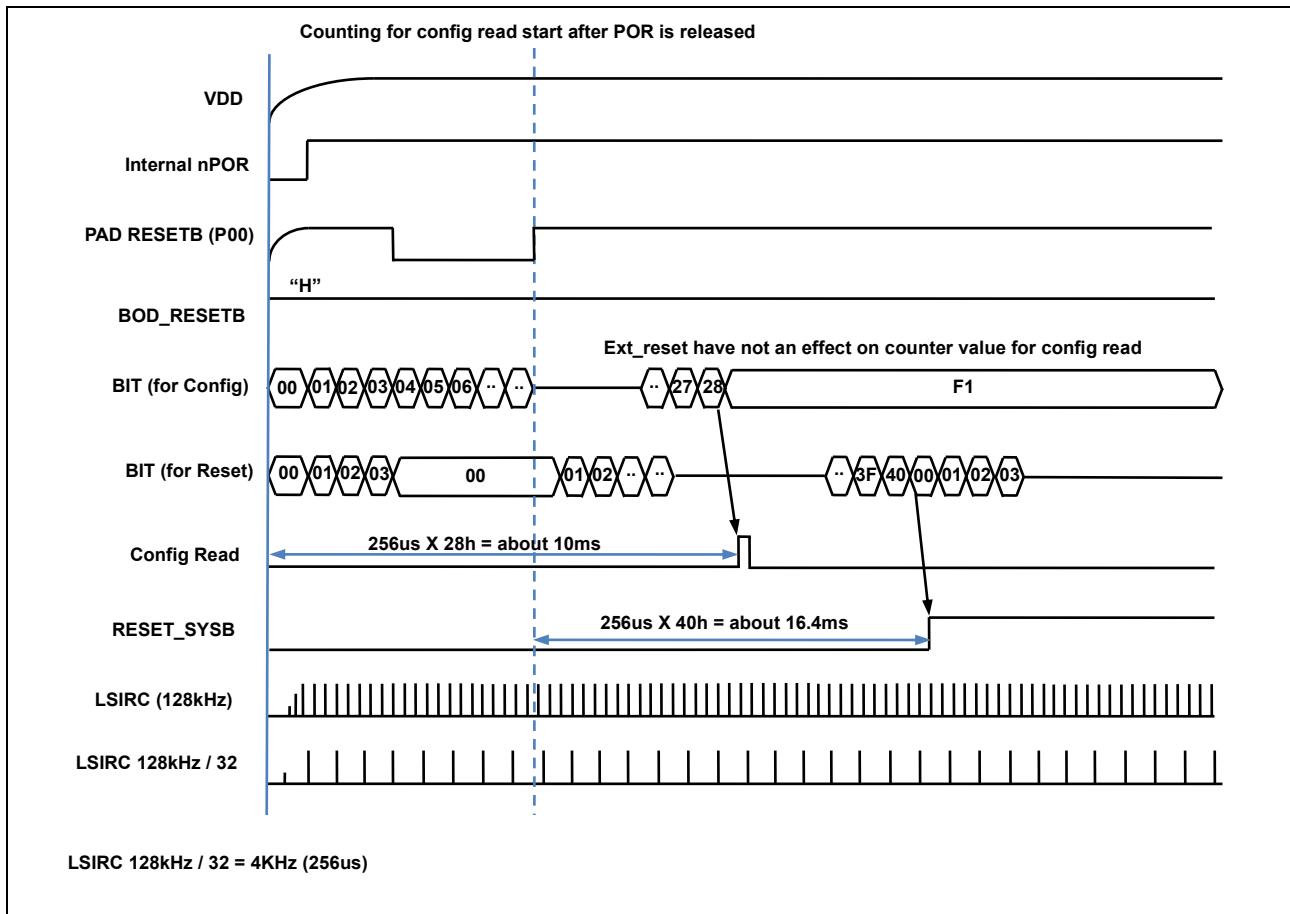


Figure 128. Configuration Timing when Power-on

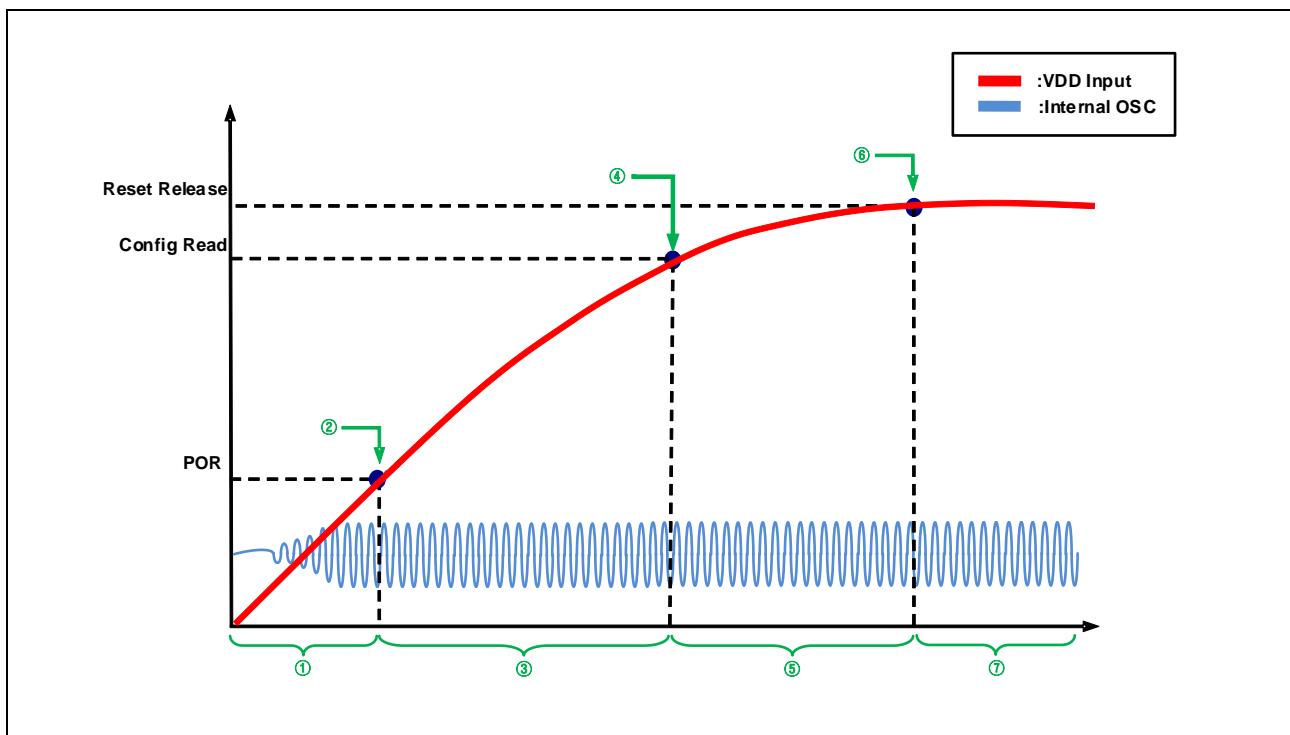


Figure 129. Boot Process WaveForm

**Table 45. Boot Process Description**

<b>Process</b>	<b>Description</b>	<b>Remarks</b>
①	- No Operation - LSIRC (128KHz) ON	0.7V~0.9V
②	-1st POR level Detection	-about 1.1V~1.3V
③	- (LSIRC 128KHz/32)x32h Delay section (=10ms) -VDD input voltage must rise over than flash operating voltage for Configure option read	-Slew Rate >= 0.025V/ms
④	- Configure option read point	-about 1.6V ~ 1.8V -Configure 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) i) after16ms, after External Reset Release (External reset) ii) 16ms point after POR (POR only)	- BIT is used for Peripheral stability
⑦	-Normal operation	

## 19.5 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.

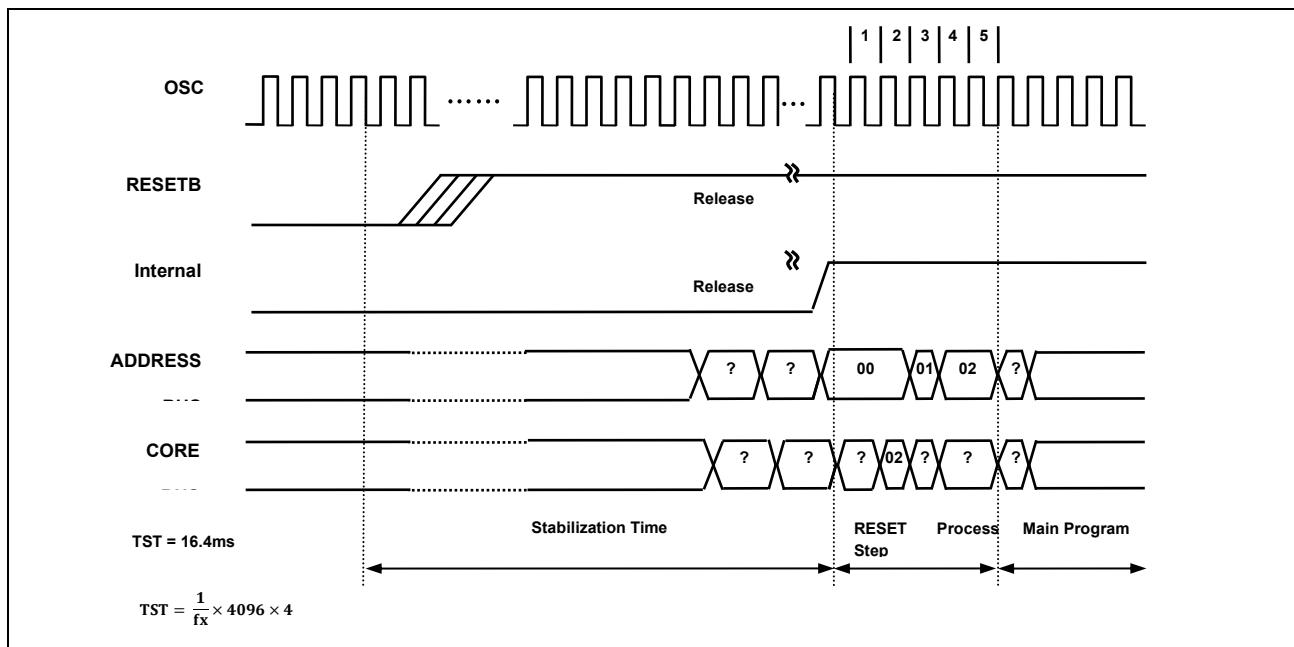


Figure 130. Timing Diagram after RESET

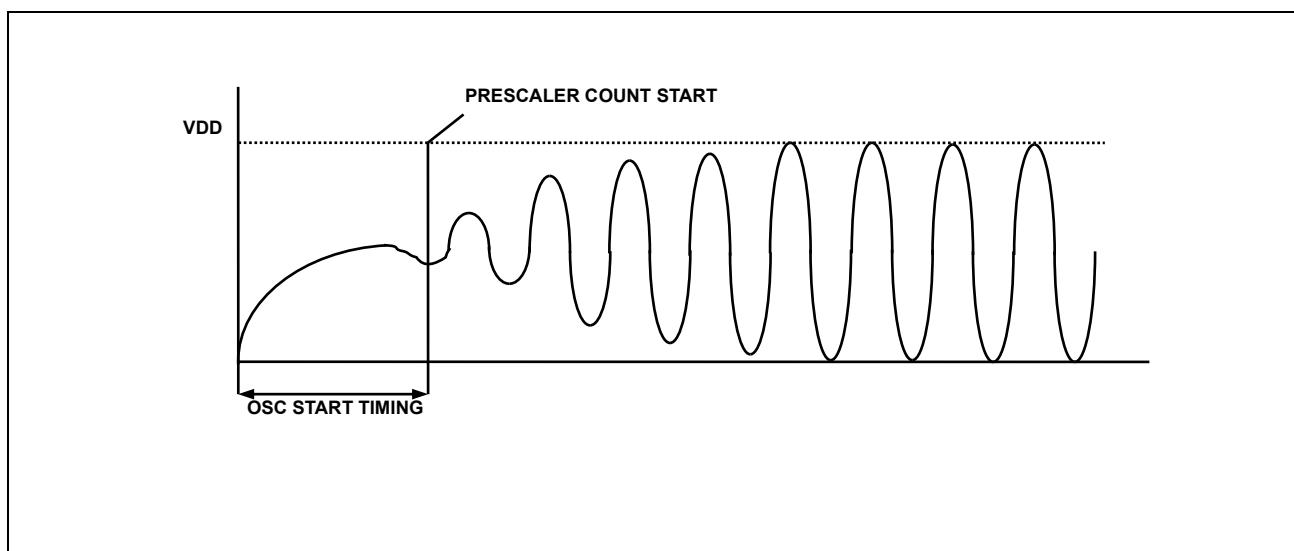


Figure 131. Oscillator generating waveform example

### NOTES:

1. As shown Figure 130, the stable generating time is not included in the start-up time.
2. The RESETB pin has a Pull-up register by hardware

## 19.6 Low Voltage Reset Processor

The A96T418 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 1.62V, 1.77V, 1.88V, 2.00V, 2.13V, 2.28V, 2.46V, 2.67V, 2.80V, 3.04V, 3.20V, 3.55V, 3.75V, 3.99V, 4.25V. 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.

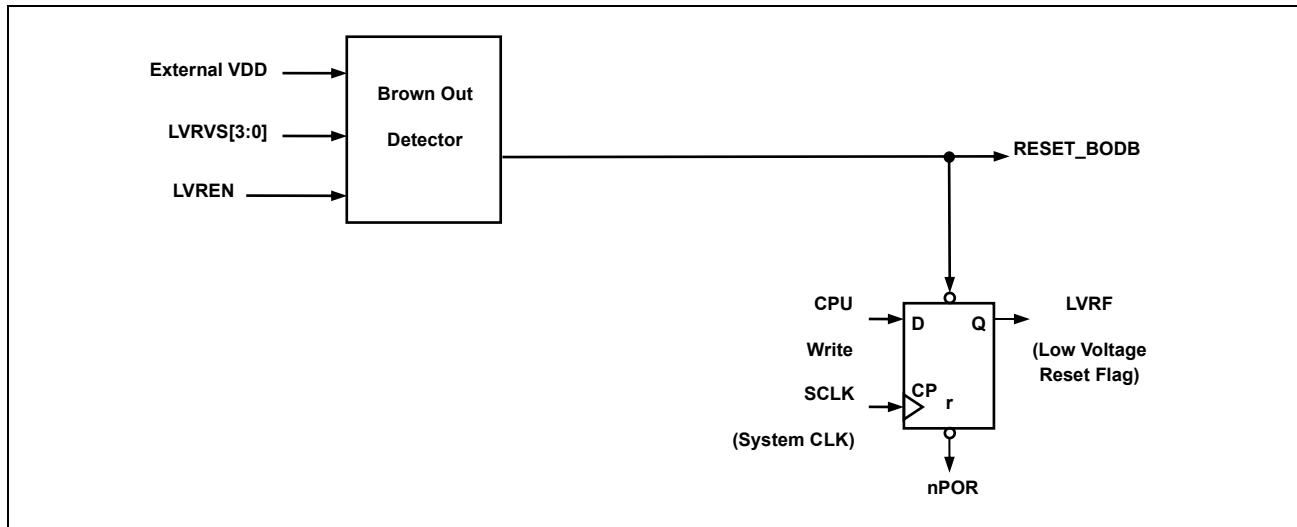


Figure 132. Block Diagram of LVR

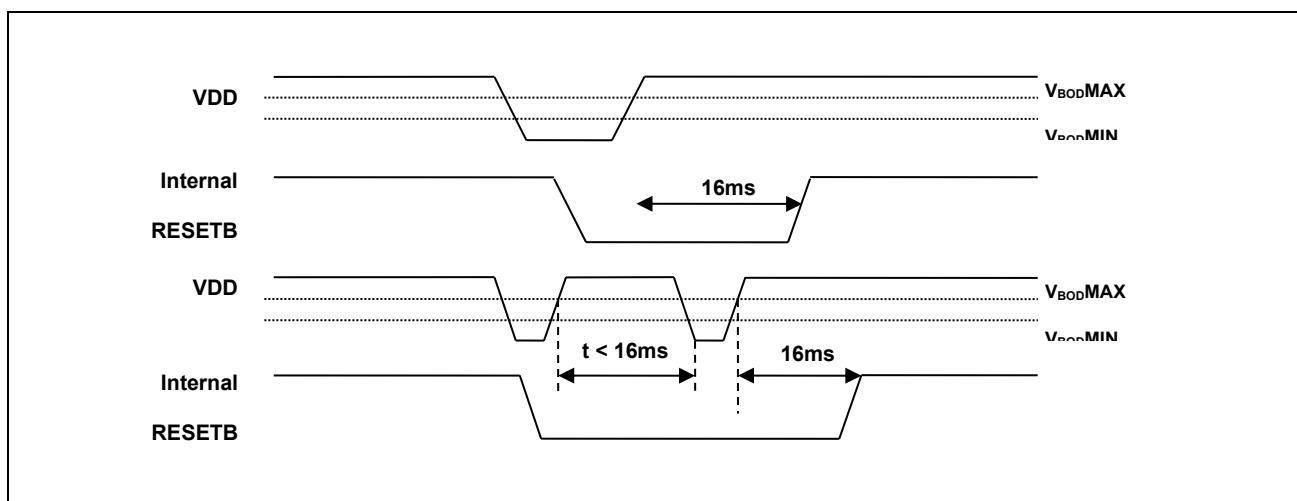


Figure 133. Internal Reset at the power fail situation

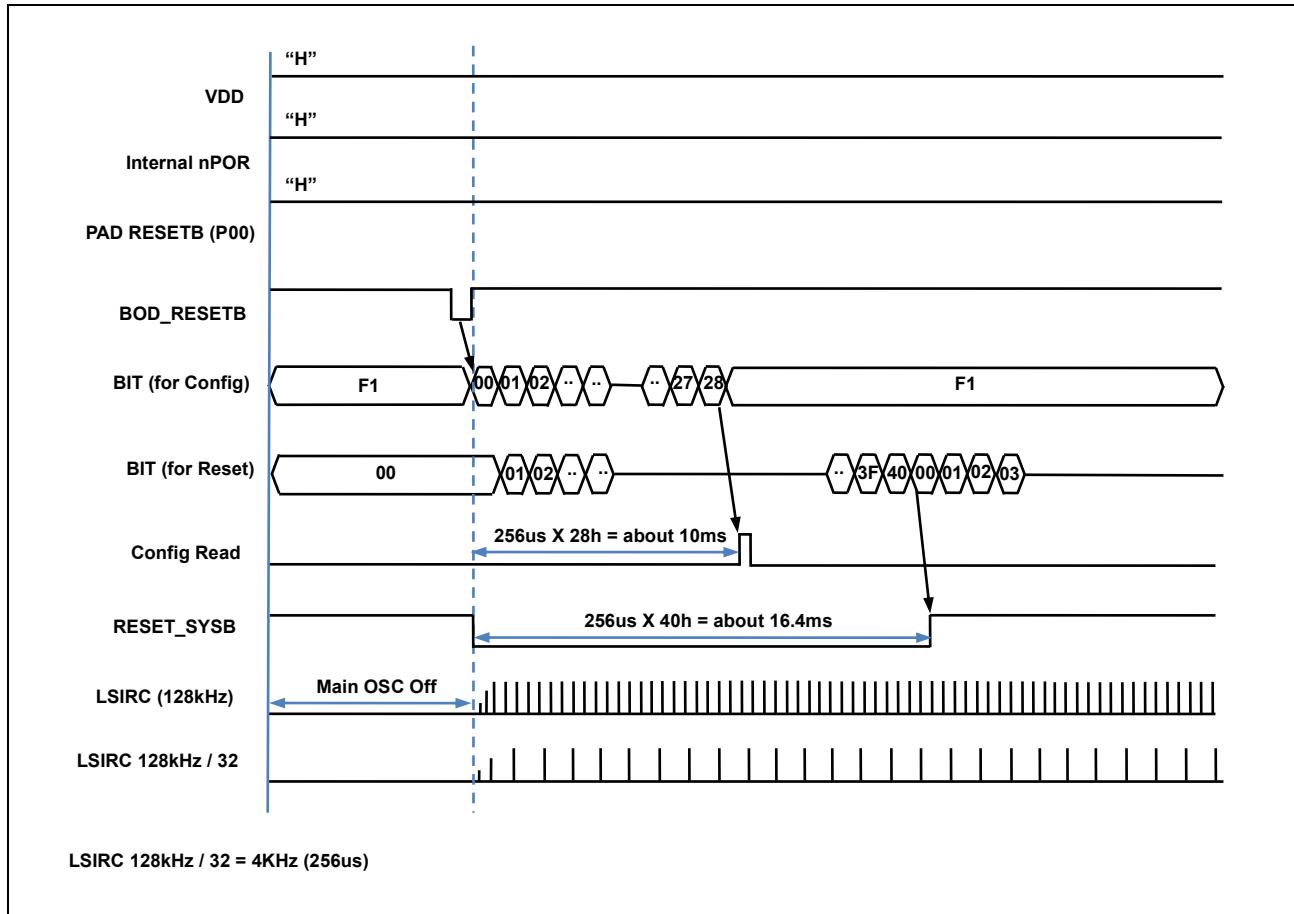


Figure 134. Configuration timing when LVR RESET

## 19.7 LVI Block Diagram

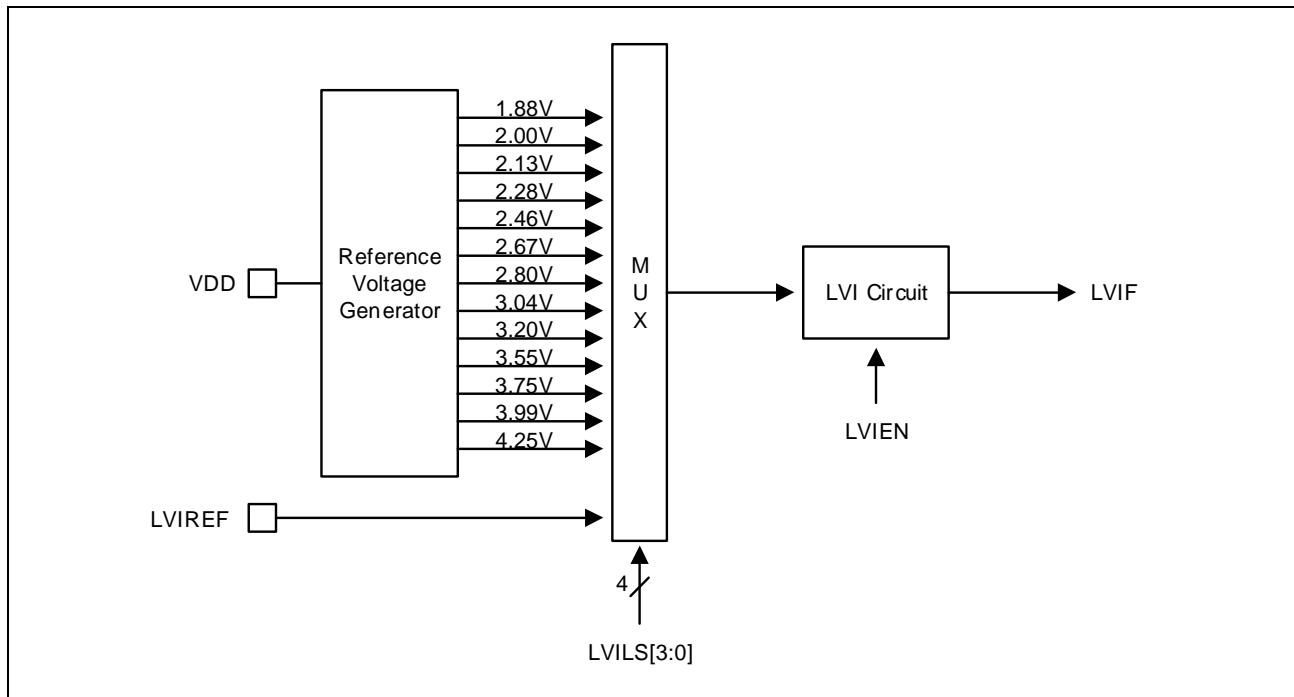


Figure 135. LVI Diagram

## 19.8 Register Map

Table 46. Reset Operation Register Map

Name	Address	Direction	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

## 19.9 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).

## 19.10 Register Description for Reset Operation

### RSTFR (Reset Flag Register): E8H

7	6	5	4	3	2	1	0
PORF	EXTRF	WDTRF	OCDRF	LVRF	-	-	-
RW	RW	RW	RW	RW	-	-	-

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	Watchdog 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 debugger 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

#### NOTES:

- When the Power-On Reset occurs, the PORF bit is only set to "1", the other flag (WDTRF) bits are all cleared to "0".
- 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.
- 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.
- 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
-	-	-	LVRVS3	LVRVS2	LVRVS1	LVRVS0	LVREN
-	-	-	RW	R/W	R/W	RW	R/W

Initial value: 00H

**LVRVS[3:0]****LVR Voltage Select**

LVRVS3	LVRVS2	LVRVS1	LVRVS0	Description
0	0	0	0	1.61V
0	0	0	1	Not available
0	0	1	0	1.77V
0	0	1	1	1.88V
0	1	0	0	2.00V
0	1	0	1	2.13V
0	1	1	0	2.28V
0	1	1	1	2.46V
1	0	0	0	2.68V
1	0	0	1	2.81V
1	0	1	0	3.06V
1	0	1	1	3.21V
1	1	0	0	3.56V
1	1	0	1	3.73V
1	1	1	0	3.91V
1	1	1	1	4.25V

**LVREN****LVR Operation**

0	LVR Enable
1	LVR Disable

**NOTES:**

1. The LVRVS[3:0] bits are cleared by a power-on reset but are retained by other reset signals.
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	LVEN	LVLS3	LVLS2	LVLS1	LVLS0
–	–	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				
LVIVS[3:0]	LVI Level Select				
	LVIVS3	LVIVS2	LVIVS1	LVIVS0	Description
	0	0	0	0	Not available
	0	0	0	1	Not available
	0	0	1	0	Not available
	0	0	1	1	Not available
	0	1	0	0	Not available
	0	1	0	1	Not available
	0	1	1	0	2.28V
	0	1	1	1	2.46V
	1	0	0	0	2.68V
	1	0	0	1	2.81V
	1	0	1	0	3.06V
	1	0	1	1	3.21V
	1	1	0	0	3.56V
	1	1	0	1	3.73V
	1	1	1	0	3.91V
	1	1	1	1	4.25V

## 20 Memory Programming

A96T418 has flash memory to which a program can be written, erased, and overwritten while mounted on the board. Serial ISP mode is supported.

Flash of A96T418 features the followings:

- Flash Size : 32Kbytes
- Single power supply program and erase
- Command interface for fast program and erase operation
- Up to 10,000 program/erase cycles at typical voltage and temperature for flash memory
- Security feature

### 20.1 Flash Control and Status Register

Registers to control Flash and Data EEPROM are Mode Register (FEMR), Control Register (FEGR), Status Register (FESR), Time Control Register (FETCR), Address Low Register x (FEARLx), Address Middle Register x (FEARMx), address High Register (FEARH). They are mapped to SFR area and can be accessed only in programming mode.

#### 20.1.1 Register Map

**Table 47. Register Map**

Name	Address	Direction	Default	Description
FEMR	1020H	R/W	00H	Flash Mode Register
FEGR	1021H	R/W	03H	Flash Control Register
FESR	1022H	R/W	80H	Flash Status Register
FETCR	1023H	R/W	00H	Flash Time Control Register
FEARL1	1025H	R/W	00H	Flash Address Low Register 1
FEARM1	1024H	R/W	00H	Flash Address Middle Register 1
FEARL	102AH	R/W	00H	Flash Address Low Register
FEARM	1029H	R/W	00H	Flash Address Middle Register
FEARH	1028H	R/W	00H	Flash Address High Register
FEDR	102BH	R/W	00H	Flash Data Register
FETR	102CH	R/W	00H	Flash Test Register

### 20.1.2 Register Description for Flash

#### FEMR (Flash Mode Register): 1020H

7	6	5	4	3	2	1	0
FSEL	-	PGM	ERASE	PBUFF	OTPE	VFY	FEEN
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

FSEL	Select flash memory.
0	Deselect flash memory
1	Select flash memory
PGM	Enable program or program verify mode with VFY
0	Disable program or program verify mode
1	Enable program or program verify mode
ERASE	Enable erase or erase verify mode with VFY
0	Disable erase or erase verify mode
1	Enable erase or erase verify mode
PBUFF	Select page buffer
0	Deselect page buffer
1	Select page buffer
OTPE	Select OTP area instead of program memory
0	Deselect OTP area
1	Select OTP area
VFY	Set program or erase verify mode with PGM or ERASE Program Verify: PGM=1, VFY=1 Erase Verify: ERASE=1, VFY=1
FEEN	Enable program and erase of Flash. When inactive, it is possible to read as normal mode
0	Disable program and erase
1	Enable program and erase

**FECR (Flash Control Register): 1021H**

7	6	5	4	3	2	1	0
AEF	-	EXIT1	EXIT0	WRITE	READ	nFERST	nPBRST
RW	-	RW	RW	RW	RW	RW	RW

Initial value: 03H

AEF	Enable flash bulk erase mode		
	0 Disable bulk erase mode of Flash memory		
	1 Enable bulk erase mode of Flash memory		
EXIT[1:0]	Exit from program mode. It is cleared automatically after 1 clock		
	EXIT1	EXIT0	Description
	0	0	Don't exit from program mode
	0	1	Don't exit from program mode
	1	0	Don't exit from program mode
	1	1	Exit from program mode
WRITE	Start to program or erase of Flash. It is cleared automatically after 1 clock		
	0 No operation		
	1 Start to program or erase of Flash		
READ	Start auto-verify of Flash. It is cleared automatically after 1 clock		
	0 No operation		
	1 Start auto-verify of Flash (Checksum or CRC16)		
nFERST	Reset Flash control logic. It is set automatically after 1 clock		
	0 Reset Flash control logic		
	1 No operation (default)		
nPBRST	Reset page buffer with PBUFF. It is set automatically after 1 clock		
	PBUFF	nPBRST	Description
	0	0	Page buffer reset
	1	0	Page buffer select register reset
	X	1	No operation (default)

WRITE and READ bits can be used in program, erase and verify mode with FEAR registers. Read or writes for memory cell or page buffer uses read and write enable signals from memory controller. Indirect address mode with FEAR is only allowed to program, erase and verify.

**FESR (Flash Status Register): 1022H**

7	6	5	4	3	2	1	0
PEVBSY	REMAPSI	REMAPS	-	ROMINT	WMODE	EMODE	VMODE
R	R	R	R	R/W	R	R	R

Initial value: 80H

PEVBSY	Operation status flag. It is cleared automatically when operation starts. Operations are program, erase or verification 0      Busy (Operation processing) 1      Complete Operation
REMAPSI	Remapping for check the serial ID. 0      No operation 1      Remapping OTP area to 81C0~81FF.
REMAPS	Test Only.
ROMINT	Flash interrupt request flag. Auto-cleared when program/erase/verify starts. Active in program/erase/verify completion 0      No interrupt request. 1      Interrupt request.
WMODE	Write mode flag
EMODE	Erase mode flag
VMODE	Verify mode flag

In Flash page write mode or Flash page erase mode, recommended procedure is :

Start program(FECR = 0x0B) -&gt; one NOP instruction(\_nop\_( )) -&gt; Read FESR until PEVBSY is 1(while( !( FESR &amp; 0x80)).

**FEARL1 (Flash Address Low Register 1): 1025H**

7	6	5	4	3	2	1	0
ARL17	ARL16	ARL15	ARL14	ARL13	ARL12	ARL11	ARL10
W	W	W	W	W	W	W	W

Initial value: 00H

ARL1[7:0]      Flash address low 1

**FEARM1 (Flash Address Middle Register 1): 1024H**

7	6	5	4	3	2	1	0
ARM17	ARM16	ARM15	ARM14	ARM13	ARM12	ARM11	ARM10
W	W	W	W	W	W	W	W

Initial value: 00H

ARM1[7:0]      Flash address middle 1

**FEARL (Flash Address Low Register): 102AH**

7	6	5	4	3	2	1	0
ARL7	ARL6	ARL5	ARL4	ARL3	ARL2	ARL1	ARL0
W	W	W	W	W	W	W	W

Initial value: 00H

ARL[7:0]      Flash address low

**FEARM (Flash Address Middle Register): 1029H**

7	6	5	4	3	2	1	0
ARM7	ARM6	ARM5	ARM4	ARM3	ARM2	ARM1	ARM0
W	W	W	W	W	W	W	W

Initial value: 00H

ARM[7:0]      Flash address middle

**FEARH (Flash Address High Register): 1028H**

7	6	5	4	3	2	1	0
ARH7	ARH6	ARH5	ARH4	ARH3	ARH2	ARH1	ARH0
W	W	W	W	W	W	W	W

Initial value: 00H

ARH[7:0]      Flash address high

FEAR registers are used for program, erase and auto-verify. In program and erase mode, it is page address and ignored the same least significant bits as the number of bits of page address. In auto-verify mode, address increases automatically by one.

FEARs are write-only register. Reading these registers returns 24-bit checksum result.

When calculating flash checksum, the lower 4 bits of start address are calculated as 0x0000 and the lower 4 bits of end address as 0x1111 for protection.

This device can support internal CheckSum calculation, device verification time will be decreased dramatically.

CheckSum cannot detect error address or error bit, but it is quite good feature in mass product programming.

Device data read out time takes few seconds. The execution time per byte is 4~5ms based on 16MHz.

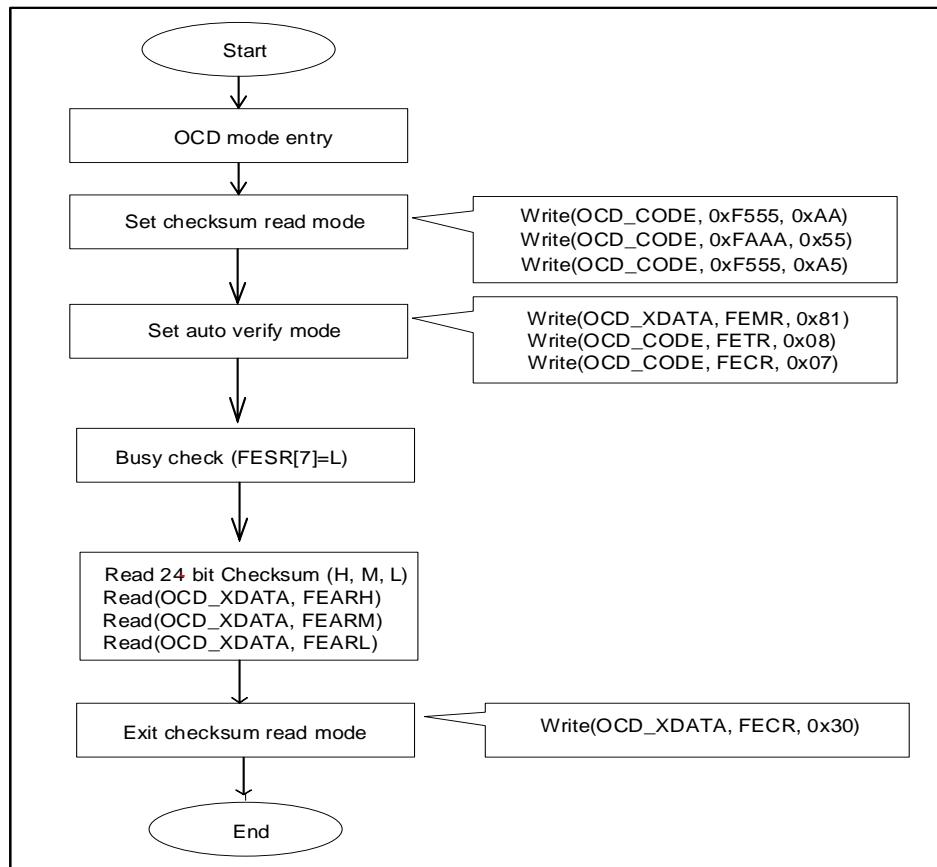


Figure 136. Read device internal CkeckSum(Full size)

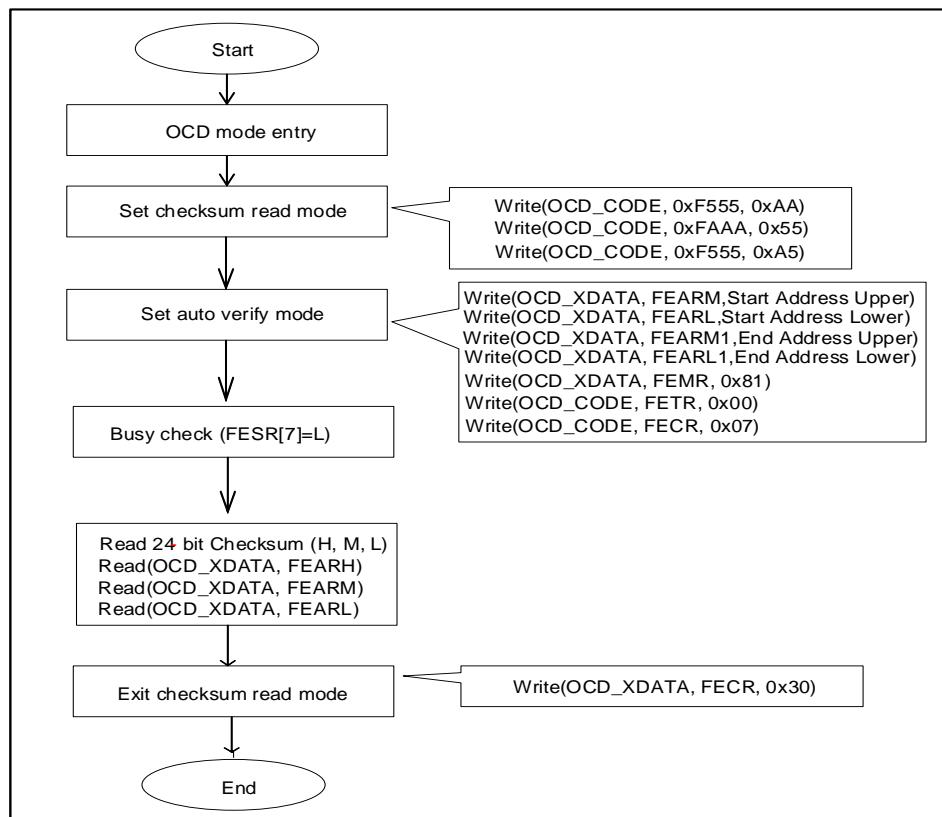


Figure 137. Read device internal CkeckSum(User define size)

**FETCR (Flash Time control Register): 1023H**

7	6	5	4	3	2	1	0
TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0
RW	RW	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

TCR[7:0] Flash Time control

Program and erase time is controlled by setting FETCR register. Program and erase timer uses 10-bit counter. It increases by one at each RING clock frequency ( $f_{LSIRC}=128\text{kHz}$ ). It is cleared when program or erase starts. Timer stops when 10-bit counter is same to FETCR. PEVBSY is cleared when program, erase or verify starts and set when program, erase or verify stops.

Max program/erase time at INTRC/256 clock:  $(255+1) * 2 * (7.8125\mu\text{s}) = 4.0\text{ms}$

In the case of  $\pm 10\%$  of error rate of counter source clock, program or erase time is 3.6~4.4ms.

\* Program/erase time calculation

For page write or erase =  $T_{pe} = (TCON+1) * 2 * (f_{LSIRC})$

For bulk erase,  $T_{be} = (TCON+1) * 4 * (f_{LSIRC})$

Recommended bulk erase time: FETCR = 57h

Recommended program / page erase time: FETCR = AFh

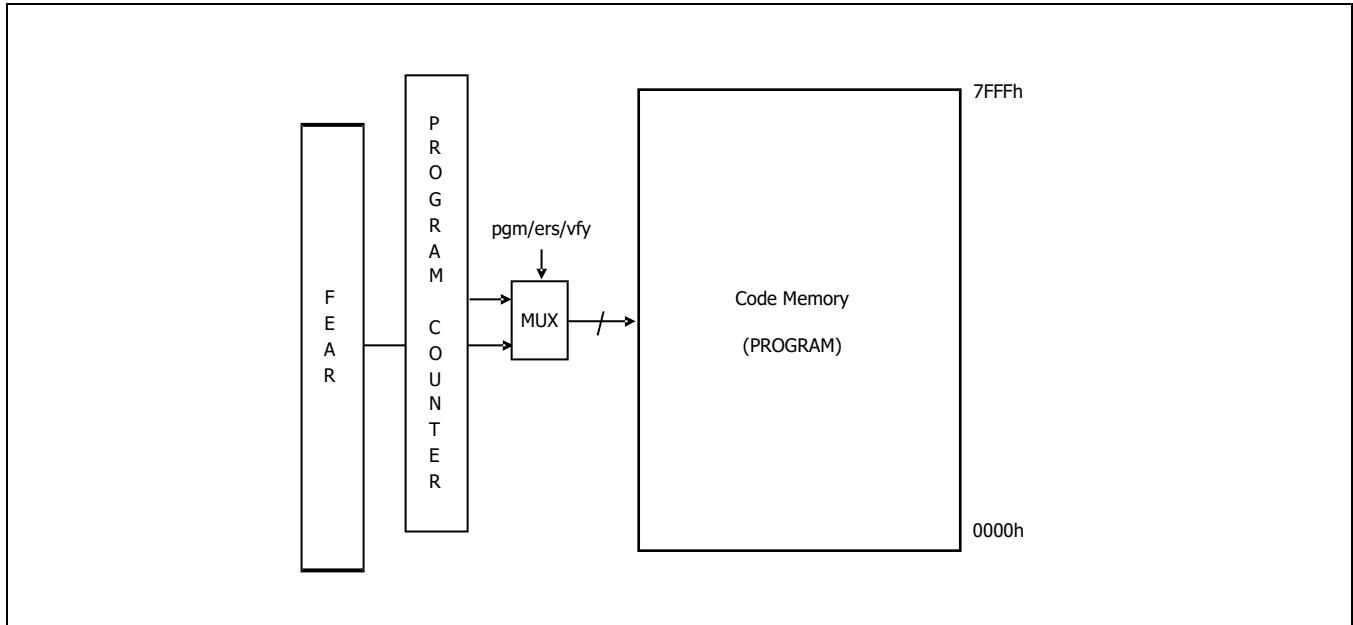
**Table 48. Program/erase Time**

	Min	Typ	Max	Unit
program/erase Time	2.4	2.5	2.6	Ms

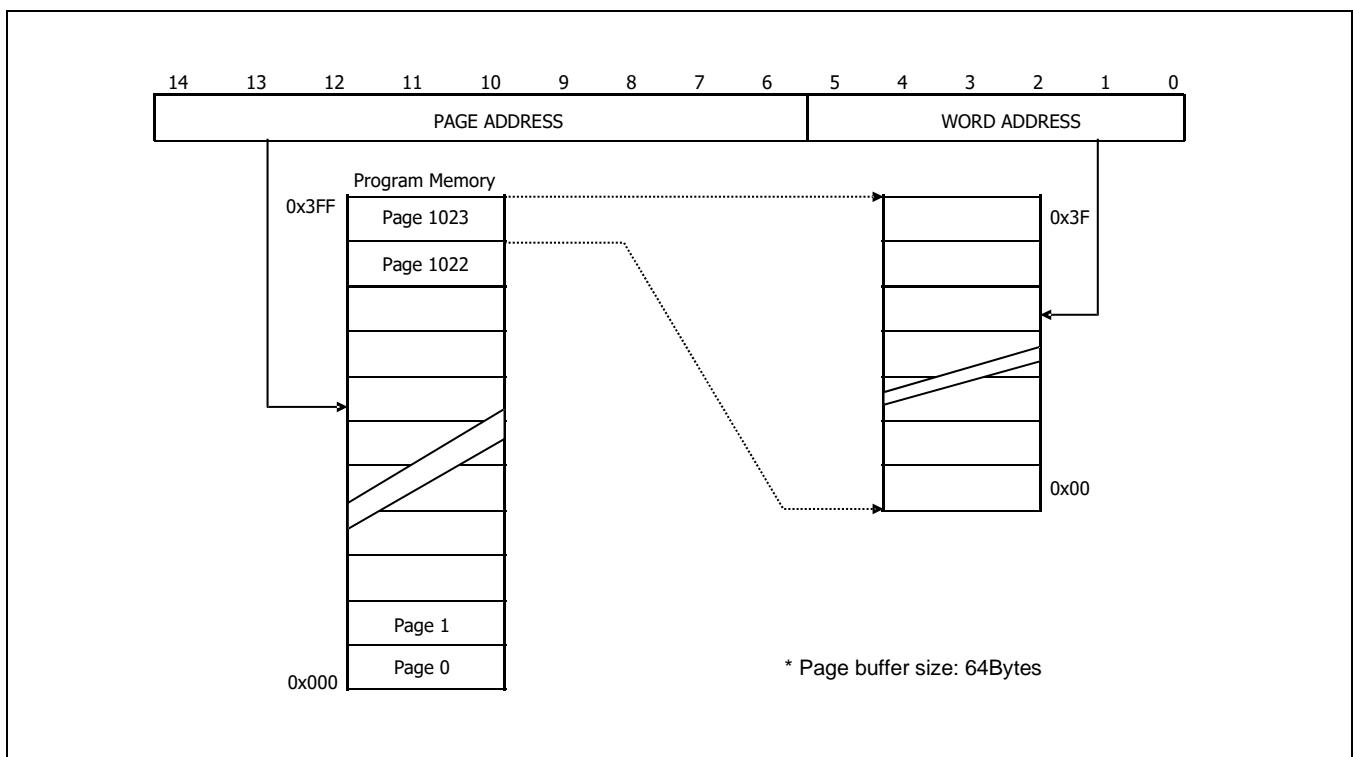
## 20.2 Memory Map

### 20.2.1 Flash Memory Map

Program memory uses 32KBytes of Flash memory. It is read by byte and written by byte or page. One page is 64-bytes.



**Figure 138. Flash Memory Map**



**Figure 139. Address Configuration of Flash Memory**

## 20.3 Serial In-System Program Mode

Serial in-system program uses the interface of debugger which uses two wires. Refer to chapter 14 in details about debugger.

### 20.3.1 Flash Operation

**Configuration** (This Configuration is just used for follow description)

7	6	5	4	3	2	1	0
-	FEMR[4]&[1]	FEMR[5]&[1]	-	-	FEMR[2]	FECR[6]	FECR[7]
-	ERASE&VFY	PGM&VFY	-	-	OTPE	AEE	AEF

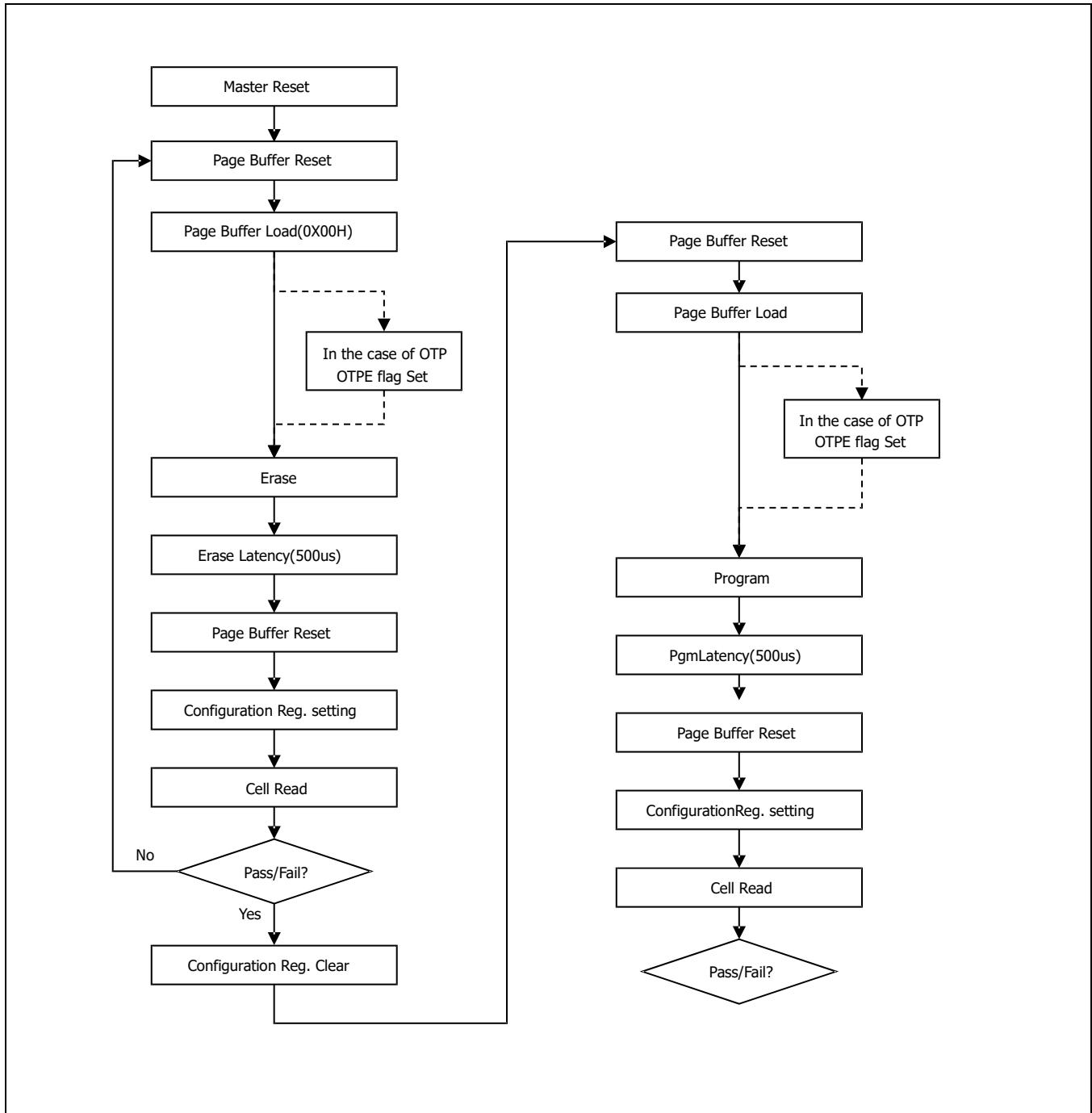


Figure 140. The Sequence of Page Program and Erase of Flash Memory

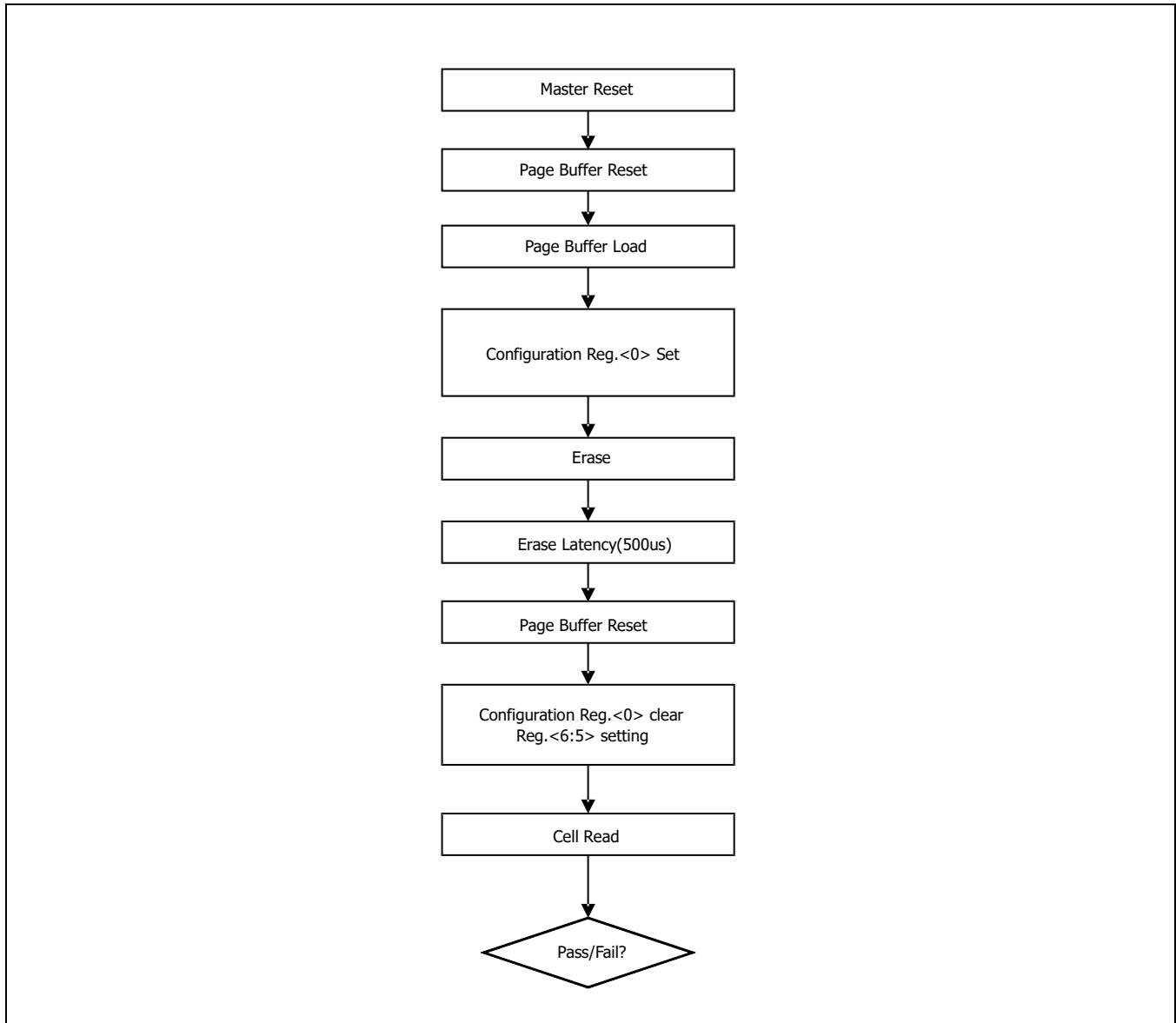


Figure 141. The Sequence of Bulk Erase of Flash Memory

### Flash Read

- Step 1. Enter OCD(=ISP) mode.
- Step 2. Set ENBDM bit of BCR.
- Step 3. Enable debug and Request debug mode.
- Step 4. Read data from Flash.

### Enable program mode

- Step 1. Enter OCD(=ISP) mode.<sup>1</sup>
- Step 2. Set ENBDM bit of BCR.

Step 3. Enable debug and Request debug mode.

Step 4. Enter program/erase mode sequence.<sup>2</sup>

(1) Write AAH to F555H.

(2) Write 55H to FAAAH.

(3) Write A5H to F555H.

<sup>1</sup> Refer to how to enter ISP mode.

<sup>2</sup> Command sequence to activate Flash write/erase mode. It is composed of sequentially writing data of Flash memory.

### **Flash write mode**

Step 1. Enable program mode.

Step 2. Reset page buffer. FEMR: 1000\_0001B, FECR: 0000\_0010B

Step 3. Select page buffer. FEMR: 1000\_1001B

Step 4. Write data to page buffer. (Address automatically increases by twin.)

Step 5. Set write mode. FEMR: 1010\_0001B

Step 6. Set page address. FEARH:FEARM:FEARL=20'hx\_xxxx

Step 7. Set FETCR.

Step 8. Start program. FECR: 0000\_1011B

Step 9. Insert one NOP operation.

Step 10. Read FESR until PEVBSY is 1.

Step 11. Repeat step 2 to step 8 until all pages are written.

### **Flash page erase mode**

Step 1. Enable program mode.

Step 2. Reset page buffer. FEMR: 1000\_0001B, FECR: 0000\_0010B

Step 3. Select page buffer. FEMR: 1000\_1001B

Step 4. Write 00H to page buffer. (Data value is not important.)

Step 5. Set erase mode. FEMR: 1001\_0001B

Step 6. Set page address. FEARH:FEARM:FEARL=20'hx\_xxxx

Step 7. Set FETCR.

Step 8. Start erase. FECR: 0000\_1011B

Step 9. Insert one NOP operation

Step 10. Read FESR until PEVBSY is 1.

Step 11. Repeat step 2 to step 8 until all pages are erased.

#### **Flash bulk erase mode**

Step 1. Enable program mode.

Step 2. Reset page buffer. FEMR: 1000\_0001B, FECR: 0000\_0010B

Step 3. Select page buffer. FEMR: 1000\_1001B

Step 4. Write 00H to page buffer. (Data value is not important.)

Step 5. Set erase mode. FEMR: 1001\_0001B

(Only main cell area is erased. For bulk erase including OTP area, select OTP area. (set FEMR to 1000\_1101B.)

Step 6. Set FETCR.

Step 7. Start bulk erase. FECR: 1000\_1011B

Step 8. Insert one NOP operation.

Step 9. Read FESR until PEVBSY is 1.

#### **Flash OTP area read mode**

Step 1. Enter OCD(=ISP) mode.

Step 2. Set ENBDM bit of BCR.

Step 3. Enable debug and Request debug mode.

Step 4. Select OTP area. FEMR: 1000\_0101B

Step 5. Read data from Flash.

#### **Flash OTP area write mode**

Step 1. Enable program mode.

Step 2. Reset page buffer. FEMR: 1000\_0001B, FECR: 0000\_0010B

Step 3. Select page buffer. FEMR: 1000\_1001B

Step 4. Write data to page buffer. (Address automatically increases by twin.)

Step 5. Set write mode and select OTP area. FEMR: 1010\_0101B

Step 6. Set page address. FEARH:FEARM:FEARL=20'hx\_xxxx

Step 7. Set FETCR.

Step 8. Start program. FECR: 0000\_1011B

Step 9. Insert one NOP operation.

Step 10. Read FESR until PEVBSY is 1.

**Flash OTP area erase mode**

- Step 1. Enable program mode.
- Step 2. Reset page buffer. FEMR: 1000\_0001B, FECR: 0000\_0010B
- Step 3. Select page buffer. FEMR: 1000\_1001B
- Step 4. Write 00H to page buffer. (Data value is not important.)
- Step 5. Set erase mode and select OTP area. FEMR: 1001\_0101B
- Step 6. Set page address. FEARH:FEARM:FEARL=20'hx\_xxxx
- Step 7. Set FETCR.
- Step 8. Start erase. FECR: 0000\_1011B
- Step 9. Insert one NOP operation.
- Step 10. Read FESR until PEVBSY is 1.

**Flash program verify mode**

- Step 1. Enable program mode.
- Step 2. Set program verify mode. FEMR: 1010\_0011B
- Step 3. Read data from Flash.

**OTP program verify mode**

- Step 1. Enable program mode.
- Step 2. Set program verify mode. FEMR: 1010\_0111B
- Step 3. Read data from Flash.

**Flash erase verify mode**

- Step 1. Enable program mode.
- Step 2. Set erase verify mode. FEMR: 1001\_0011B
- Step 3. Read data from Flash.

**Flash page buffer read**

- Step 1. Enable program mode.
- Step 2. Select page buffer. FEMR: 1000\_1001B
- Step 3. Read data from Flash.

**Summary of Flash Program/Erase Mode**

**Table 49. Operation Mode**

<b>Operation mode</b>		<b>Description</b>
F L A S H	Flash read	Read cell by byte.
	Flash write	Write cell by bytes or page.
	Flash page erase	Erase cell by page.
	Flash bulk erase	Erase the whole cells.
	Flash program verify	Read cell in verify mode after programming.
	Flash erase verify	Read cell in verify mode after erase.
	Flash page buffer load	Load data to page buffer.

## 20.4 Mode Entrance Method of ISP Mode

### 20.4.1 Mode Entrance Method for ISP

TARGET MODE	DSDA	DSCL	DSDA
OCD(ISP)	'hC	'hC	'hC

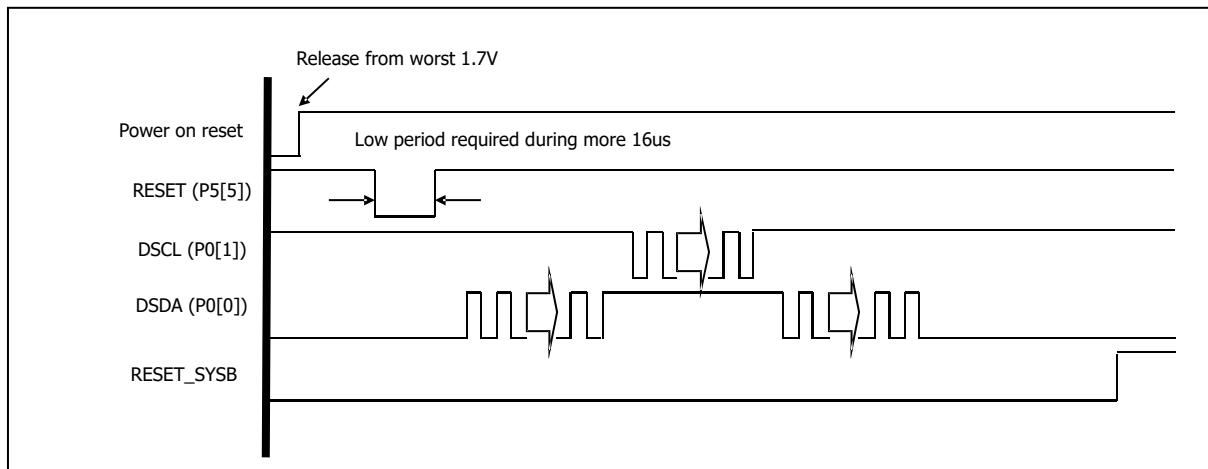


Figure 142. ISP Mode

## 20.5 Security

+  
A96T418 provides Lock bits which can be left un-programmed ("0") or can be programmed ("1") to obtain the additional features listed in Table 50. The Lock bit can only be erased to "0" with the bulk erase command and a value of more than 0x40 at FETCR.

**Table 50. Security policy using lock-bits**

LOCK MODE	USER MODE								ISP MODE							
	FLASH				OTP				FLASH				OTP			
LOCKF	R	W	PE	BE	R	W	PE	BE	R	W	PE	BE	R	W	PE	BE
0	O	O	O	X	X	X	X	X	O	O	O	O	O	O	O	O
1	O	O	O	X	X	X	X	X	X	X	X	O	O	X	X	O

- LOCKF: Lock bit of Flash memory
- R: Read
- W: Write
- PE: Page erase
- BE: Bulk Erase
- O: Operation is possible.
- X: Operation is impossible.

## 20.6 Configure Option

### 20.6.1 Configure Option Control

The data for configure option should be written in the configure option area (003EH – 003FH) by programmer (Writer tools).

### 20.6.2 Register Description

#### CONFIGURE OPTION 2: ROM Address 0001H

7	6	5	4	3	2	1	0
R_P	HL	-	VAPEN	-	-	-	RSTS

Initial value: 00H

R_P	Code Read Protection
	0 Disable
	1 Enable
HL	Code Write Protection
	0 Disable
	1 Enable
VAPEN	Vector area (00H-FFH) Protection
	0 Disable Protection
	1 Enable Protection
RSTS	Select RESETB pin
	0 Disable RESETB pin(P31)
	1 Enable RESETB pin

#### CONFIGURE OPTION 1: ROM Address 0000H (A96T418 32K Series)

7	6	5	4	3	2	1	0
-	-	-	-	PAEN	PASS2	PASS1	PASS0

Initial value: 00H

PAEN	Enable Specific Area Write Protection		
	0 Disable Protection		
	1 Enable Protection		
PASS [2:0]	Select Specific Area for Write Protection		
<b>NOTES:</b>			
	1. When PAEN = '1', it is applied.		
PASS2	PASS1	PASS0	Description
0	0	0	0.7Kbytes (Address 0100H – 03FFH)
0	0	1	1.7Kbytes (Address 0100H – 07FFH)
0	1	0	2.7Kbytes (Address 0100H – 0BFFFH)
0	1	1	3.7Kbytes (Address 0100H – 0FFFFH)
1	0	0	29.75Kbytes (Address 0100H – 77FFH)
1	0	1	30.75Kbytes (Address 0100H – 7BFFFH)
1	1	0	31.25Kbytes (Address 0100H – 7DFFFH)
1	1	1	31.50Kbytes (Address 0100H – 7EFFH)

## 21 Electrical Characteristics

### 21.1 Absolute Maximum Ratings

**Table 51. Absolute Maximum Ratings**

Parameter	Symbol	Rating	Unit	Note
Supply Voltage	VDD	-0.3~+6.5	V	–
Normal Voltage Pin	V <sub>I</sub>	-0.3~VDD+0.3	V	Voltage on any pin with respect to VSS
	V <sub>O</sub>	-0.3~VDD+0.3	V	
	I <sub>OH</sub>	42.5	mA	Maximum current output sourced by (I <sub>OH</sub> per I/O pin)
	ΣI <sub>OH</sub>	112	mA	Maximum current (ΣI <sub>OH</sub> )
	I <sub>OL1</sub>	50	mA	Maximum current (I <sub>OL1</sub> per I/O pin)
	ΣI <sub>OL1</sub>	101	mA	Maximum current (ΣI <sub>OL1</sub> )
	I <sub>OL2</sub>	160	mA	Maximum current sunk by (I <sub>OL2</sub> per I/O pin)
	ΣI <sub>OL2</sub>	250	mA	Maximum current by LED Drive (ΣI <sub>OL2</sub> )
Total Power Dissipation	P <sub>T</sub>	600	mW	–
Storage Temperature	T <sub>STG</sub>	-65~+150	°C	–

**NOTES:**

1. 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.

### 21.2 Recommended Operating Conditions

**Table 52. Recommended Operating Conditions**

(T<sub>A</sub>=-40°C ~ 85°C or T<sub>A</sub>=-40°C ~ 105°C)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Operating Voltage	VDD	f <sub>X</sub> = 16MHz HSIRC	2.0	–	5.5	V
		f <sub>X</sub> = 128kHz LSIRC	2.0	–	5.5	
		f <sub>X</sub> = 32 ~ 38kHz Ext. Sub Crystal	2.0	–	5.5	
		Touch, ADC, LED Driver	2.7	–	5.5	
Operating Temperature	T <sub>OPR</sub>	VDD=2.0~5.5V	-40	–	85	°C
			-40	–	105	

### 21.3 Touch Sensing Characteristics

**Table 53. Touch Switch Characteristics**

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Load Capacitance	Cload	-	-	50	100	pF
Operating Frequency	Fop	Cload=50pF(@typ.) Cload=100pF(@min.)	2	4	4	MHz
High-Sense Voltage	VHS	VDD=5V	2.6	3.5	5	V
COMP Reference Voltage	VCOM	VDD=5V	2.5	3	3.5	V

### 21.4 A/D Converter Characteristics

**Table 54. A/D Converter Characteristics**

(TA=-40°C ~ +85°C or TA=-40°C ~ 105°C, VDD=2.7V ~ 5.5V, VSS=0V)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Resolution	-	-	--	12	-	bit
Integral Linear Error	ILE	AVREF= 2.7V – 5.5V fx= 8MHz	-	-	±6	LSB
Differential Linearity Error	DLE		-	-	±1	
Zero Offset Error	ZOE		-	-	±5	
Full Scale Error	FSE		-	-	±5	
Conversion Time	tCON	12-bit resolution, 8MHz	8	-	-	us
Analog Input Voltage	VAN	-	VSS	-	AVREF	V
Analog Reference Voltage	AVREF	*Note 3	2.7	-	VDD	
Analog Input Leakage Current	I <sub>AN</sub>	AVREF=5.12V	-	-	2	uA
ADC Operating Current	I <sub>ADC</sub>	Enable	VDD=5.12V	1	2	mA
		Disable	-	-	0.1	uA

**NOTES:**

1. Zero offset error is the difference between 000000000000 and the converted output for zero input voltage (VSS).
2. Full scale error is the difference between 111111111111 and the converted output for full-scale input voltage (AVREF).
3. When AVREF is lower than 2.7V, the ADC resolution is worse.

### 21.5 Power-On Reset Characteristics

**Table 55. Power-on Reset Characteristics**

(TA=-40°C ~ +85°C or TA=-40°C ~ 105°C, VDD=2.0V ~ 5.5V, VSS=0V)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
RESET Release Level	V <sub>POR</sub>	-	-	1.32	-	V
VDD Voltage Rising Time	t <sub>R</sub>	-	0.05	-	50.0	V/ms
Minimum Pulse Width	t <sub>LW</sub>	-	100			us
POR Current	I <sub>POR</sub>	-	-	0.2	-	uA

## 21.6 Low Voltage Reset and Low Voltage Indicator Characteristics

**Table 56. LVR and LVI Characteristics**

( $T_A = -40^\circ\text{C} \sim +85^\circ\text{C}$  or  $T_A = -40^\circ\text{C} \sim 105^\circ\text{C}$ ,  $VDD = 2.0\text{V} \sim 5.5\text{V}$ ,  $VSS = 0\text{V}$ )

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit	
Detection Level	$V_{LVR}$ $V_{LVI}$	The LVR can select all levels but LVI can select other levels except 1.61/1.77/1.88/2.00/2.13V. (Falling level)	-	1.61	1.75	V	
			1.63	1.77	1.91		
			1.73	1.88	2.03		
			1.84	2.00	2.16		
			1.96	2.13	2.30		
			2.10	2.28	2.46		
			2.26	2.46	2.66		
			2.47	2.68	2.89		
			2.59	2.81	3.03		
			2.82	3.06	3.30		
			2.95	3.21	3.47		
			3.28	3.56	3.84		
			3.43	3.73	4.03		
			3.60	3.91	4.22		
			3.91	4.25	4.59		
Hysteresis	$\Delta V$	-	-	30	180	mV	
Minimum Pulse Width	$t_{LW}$	-	100	-	-	us	
LVR and LVI Current	$I_{BL}$	Enable (Both)	VDD= 3V, RUN Mode	-	14.0	24.0	uA
		Enable (One of two)		-	10.0	18.0	
		Disable (Both)	VDD= 3V	-	-	0.1	

**NOTES:**

- Guaranteed by design

## 21.7 High Speed Internal RC Oscillator Characteristics

**Table 57. High Internal RC Oscillator Characteristics**

( $T_A = -40^\circ\text{C} \sim +85^\circ\text{C}$  or  $T_A = -40^\circ\text{C} \sim 105^\circ\text{C}$ ,  $VDD = 2.0\text{V} \sim 5.5\text{V}$ ,  $VSS = 0\text{V}$ )

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Frequency	$f_{HSIRC}$	$VDD = 2.0 - 5.5\text{V}$	—	16	—	MHz
Tolerance	—	$T_A = 0^\circ\text{C} \text{ to } +50^\circ\text{C}$	—	—	$\pm 1.5$	%
		$T_A = -10^\circ\text{C} \text{ to } +70^\circ\text{C}$	—	—	$\pm 2.0$	
		$T_A = -40^\circ\text{C} \text{ to } +85^\circ\text{C}$	—	—	$\pm 2.5$	
		$T_A = -40^\circ\text{C} \text{ to } +105^\circ\text{C}$	—	—	$\pm 5.0$	
Clock Duty Ratio	$TOD$	—	40	50	60	%
Stabilization Time	$T_{HFS}$	—	—	—	100	us
HSIRC Current	$I_{HSIRC}$	Enable	—	0.2	—	mA
		Disable	—	—	-0.1	uA

**NOTES:**

1. A 0.1uF bypass capacitor should be connected to VDD and VSS.

## 21.8 Low Speed Internal RC Oscillator Characteristics

**Table 58. Internal WDTRC Oscillator Characteristics**

( $T_A = -40^\circ\text{C} \sim +85^\circ\text{C}$  or  $T_A = -40^\circ\text{C} \sim 105^\circ\text{C}$ ,  $VDD = 2.0\text{V} \sim 5.5\text{V}$ ,  $VSS = 0\text{V}$ )

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Frequency	$f_{LSIRC}$	$T_A = -40^\circ\text{C} \sim +85^\circ\text{C}$	102	128	154	kHz
		$T_A = -40^\circ\text{C} \sim +105^\circ\text{C}$	90	128	166	
Stabilization Time	$T_{LSIRC}$	—	—	—	1	ms
LSIRC Current	$I_{LSIRC}$	Enable	—	1	—	uA
		Disable	—	—	0.1	

## 21.9 DC Characteristics

**Table 59. DC Characteristics**

( $T_A = -40^\circ\text{C} \sim +85^\circ\text{C}$  or  $T_A = -40^\circ\text{C} \sim 105^\circ\text{C}$ ,  $VDD = 2.0\text{V} \sim 5.5\text{V}$ ,  $VSS = 0\text{V}$ ,  $f_{SCLK} = 16\text{MHz}$ )

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Input High Voltage	$V_{IH}$	All input pins	0.7VDD	—	VDD	V
Input Low Voltage	$V_{IL}$	All input pins except $V_{IL1}$	—	—	0.3VDD	V
Output High Voltage	$V_{OH1}$	$VDD = 4.5\text{V}$ , $I_{OH} = -8.57\text{mA}$ , All output ports;	VDD-1.0	—	—	V
	$V_{OH2}$	$VDD = 4.5\text{V}$ , $I_{OH} = -19\text{ mA}$ , All output ports;	VDD-2.0	—	—	V
SEG Output Current	$I_{OH30}$	$VDD = 5.0\text{V}$ , $V_{OH} = 4.5\text{V}$ , $T_A = 25^\circ\text{C}$	- 6.72	- 8.2	- 10.17	mA
	$I_{OH31}$	$VDD = 5.0\text{V}$ , $V_{OH} = 4.5\text{V}$ , $T_A = 25^\circ\text{C}$	- 10.56	- 12.89	- 15.99	mA
	$I_{OH32}$	$VDD = 5.0\text{V}$ , $V_{OH} = 4.5\text{V}$ , $T_A = 25^\circ\text{C}$	- 16.17	- 19.72	- 24.46	mA
	$I_{OH33}$	$VDD = 5.0\text{V}$ , $V_{OH} = 4.5\text{V}$ , $T_A = 25^\circ\text{C}$	- 19.3	- 23.54	- 29.19	mA
SEG Current Matching	$ITOSEG$	$VDD = 5.0\text{V}$ , $V_{OH} = 4.5\text{V}$ $T_A = 25^\circ\text{C}$ , $V_{oh31}$	-5	-	5	%
Output Low Voltage	$V_{OL1}$	$VDD = 4.5\text{V}$ , $I_{OL} = 10\text{mA}$ ; All output ports except $V_{OL2}$	—	—	1.0	V
	$V_{OL2}$	$VDD = 5\text{V}$ , $I_{OL} = 250\text{mA}$ , $T_A = 25^\circ\text{C}$ ; P2x LED High sink current output	—	1.5	-	V
Input High Leakage Current	$I_{IH}$	All input ports	—	—	1	uA
Input Low Leakage Current	$I_{IL}$	All input ports	-1	—	—	uA
Pull-Up Resistor	$R_{PU}$	$VDD = 5.0\text{V}$ , $V_I = 0\text{V}$ , $T_A = 25^\circ\text{C}$ , All Input ports	25	50	100	kΩ
OSC feedback resistor	$R_x$	$SXIN = VDD$ , $SXOUT = VSS$ $T_A = 25^\circ\text{C}$ , $VDD = 5\text{V}$	6.25	13.53	36.98	MΩ
Power Supply Current	$I_{DD1}$ (RUN)	$f_{HSIRC} = 16\text{MHz}$ , $VDD = 5\text{V} \pm 10\%$	0.7	—	4.0	mA
	$I_{DD2}$ (IDLE)	$f_{HSIRC} = 16\text{MHz}$ , $VDD = 5\text{V} \pm 10\%$	0.5	—	3.0	mA
	$I_{DD3(STOP1)}$	STOP @ WDT on, $VDD = 5.5\text{V} \pm 10\%$ , $T_A = 25^\circ\text{C}$	—	—	22.0	uA
	$I_{DD4(STOP2)}$	STOP @ WDT off & LVR off, $VDD = 5.5\text{V} \pm 10\%$ , $T_A = 25^\circ\text{C}$	—	—	7.0	

**NOTES:**

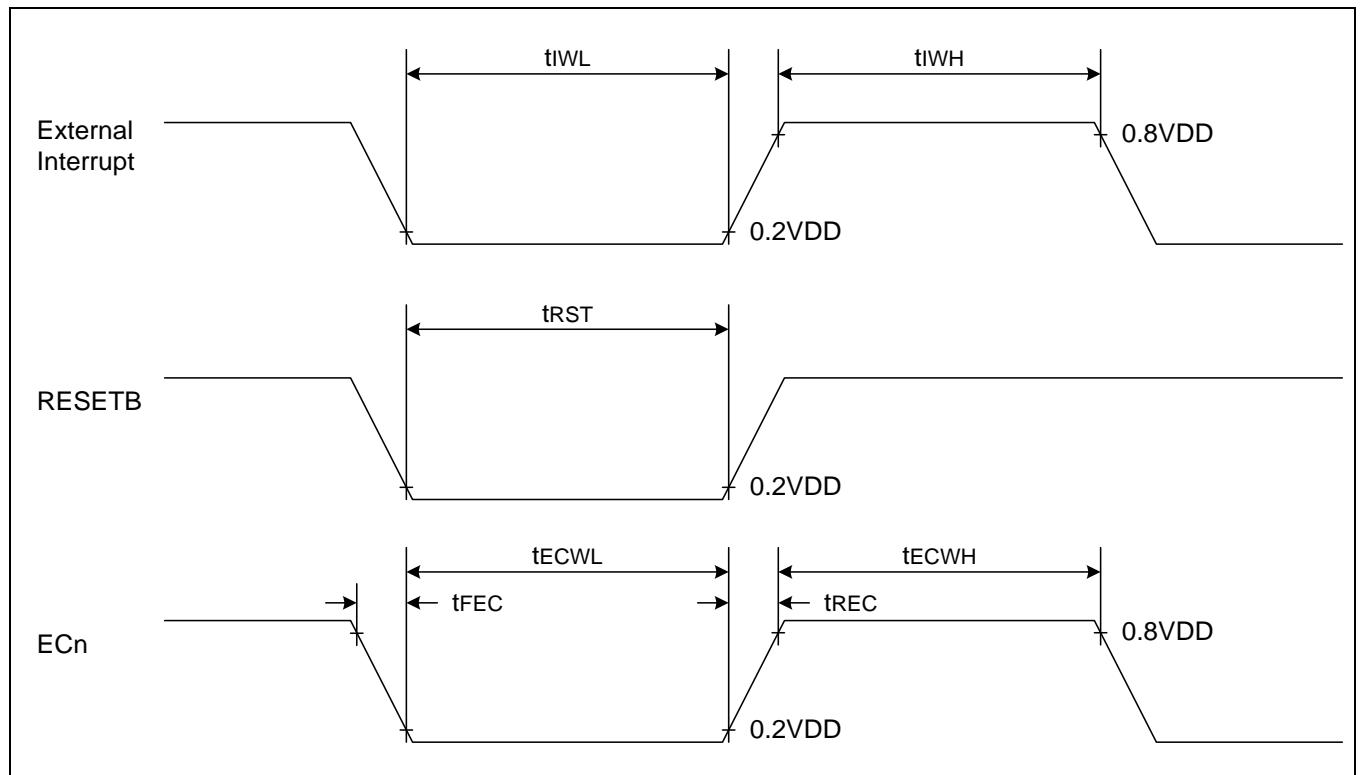
- Where  $f_{SUB}$  is an external sub oscillator, the  $f_{HSIRC}$  and  $f_{LSIRC}$  are an internal RC oscillator, and the  $f_{SCLK}$  is the selected system clock.
- All supply current items don't include the current of an internal Watch-dog timer RC (WDTRC) oscillator and a peripheral block.
- All supply current items include the current of the power-on reset (POR) block.
- SEG Current Matching represents ( ISEG-ISEGAVR ) / ISEGAVR )

## 21.10 AC Characteristics

**Table 60. AC Characteristics**

( $T_A = -40^\circ\text{C} \sim +85^\circ\text{C}$  or  $T_A = -40^\circ\text{C} \sim 105^\circ\text{C}$ ,  $VDD = 2.0\text{V} \sim 5.5\text{V}$ )

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
RESETB input low width	$t_{RSL}$	Input, $VDD = 5\text{V}$	10	—	—	us
Interrupt input high, low width	$t_{INTH}$ , $t_{INTL}$	All interrupt, $VDD = 5\text{V}$	200	—	—	ns
External Counter Input High, Low Pulse Width	$t_{ECWH}$ , $t_{ECWL}$	$EC_n$ , $VDD = 5\text{V}$ ( $n = 0, 1, 3$ )	200	—	—	
External Counter Transition Time	$t_{REC}$ , $t_{FEC}$	$EC_n$ , $VDD = 5\text{V}$ ( $n = 0, 1, 3$ )	20	—	—	



**Figure 143. AC Timing**

## 21.11 USART Characteristics

**Table 61. USART Characteristics**

( $T_A = -40^\circ\text{C} \sim +85^\circ\text{C}$  or  $T_A = -40^\circ\text{C} \sim 105^\circ\text{C}$ ,  $VDD = 2.0\text{V} \sim 5.5\text{V}$ ,  $f_{SCLK} = 8\text{MHz}$ )

Parameter	Symbol	MIN	Typ	MAX	Unit
Serial port clock cycle time	$t_{SCK}$	1800	$t_{CPU} \times 16$	2200	ns
Output data setup to clock rising edge	$t_{S1}$	8100	$t_{CPU} \times 13$	—	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	$t_{HIGH}, t_{LOW}$	720	$t_{CPU} \times 8$	1280	ns

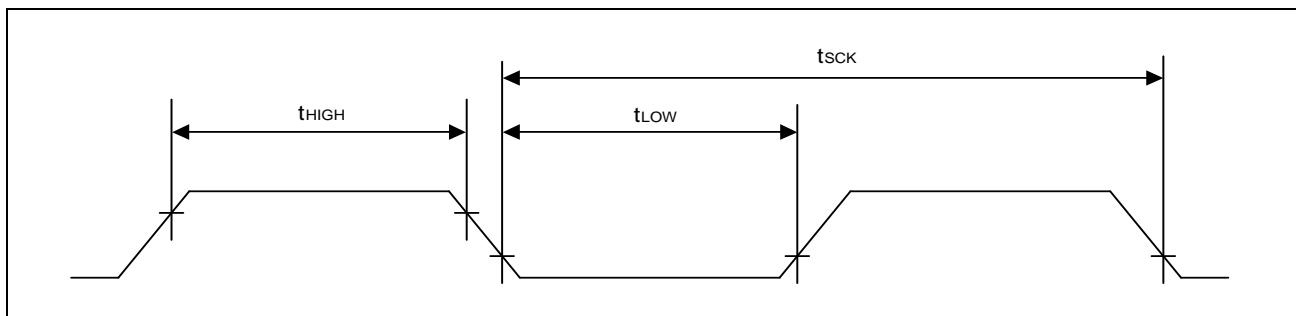


Figure 144. Waveform for USART Timing Characteristics

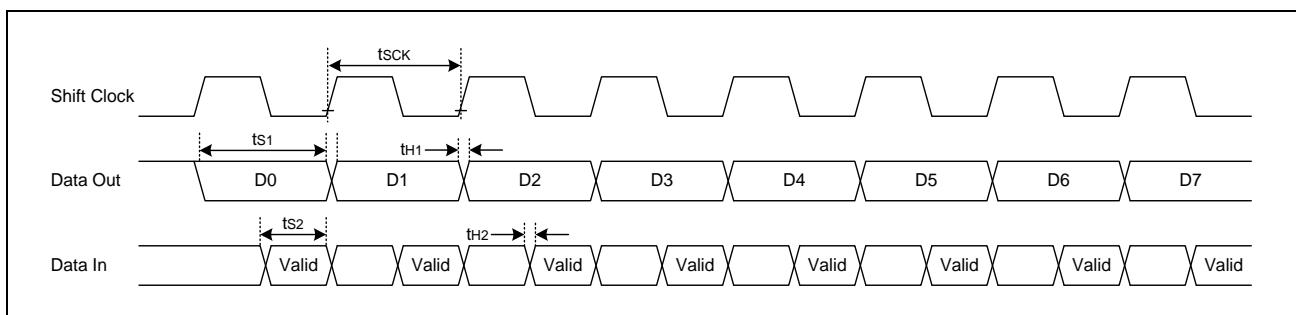


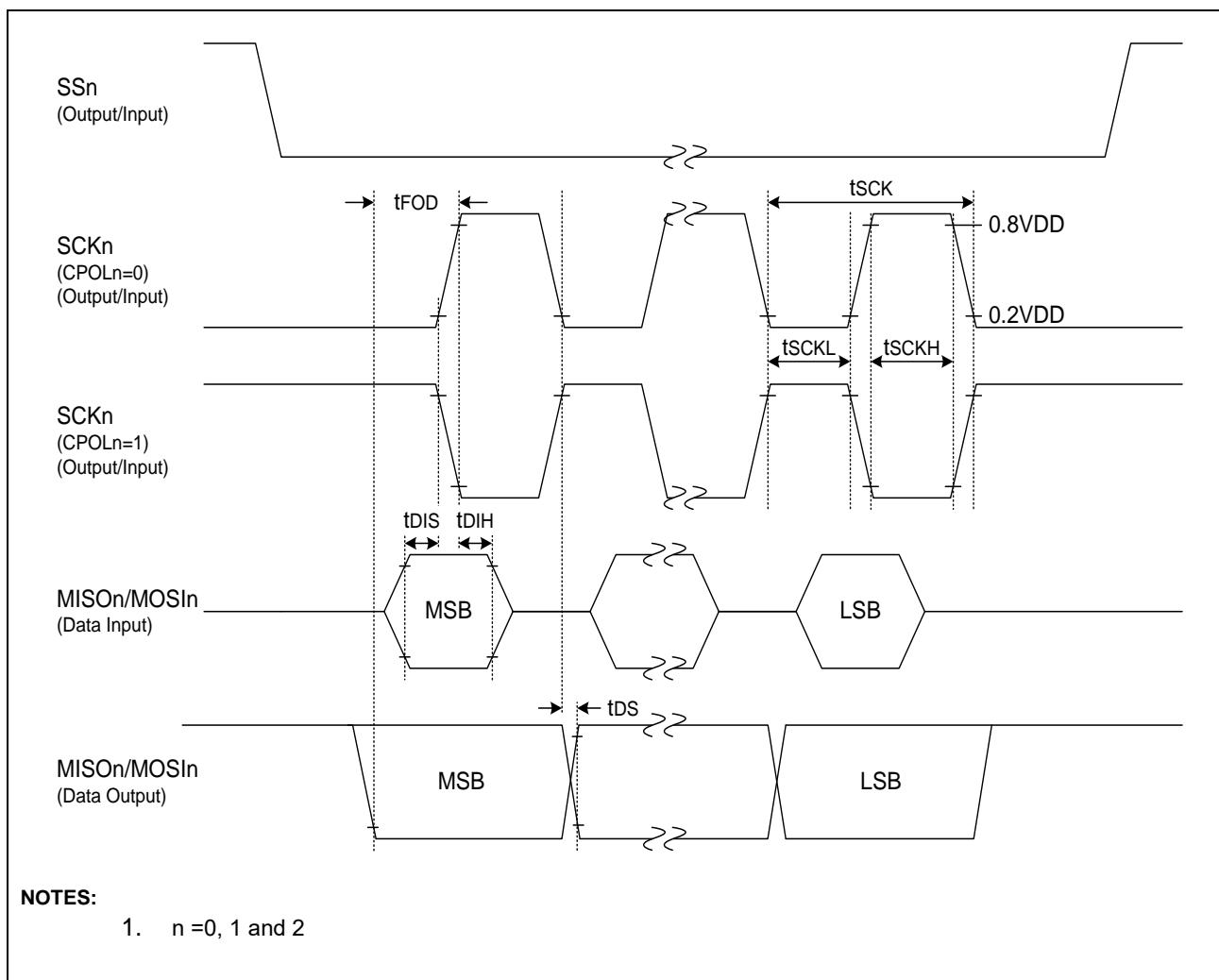
Figure 145. Timing Waveform for the USART Module

## 21.12 SPI0 Characteristics

**Table 62. SPI0 Characteristics**

( $T_A = -40^\circ\text{C} \sim +85^\circ\text{C}$  or  $T_A = -40^\circ\text{C} \sim 105^\circ\text{C}$ ,  $VDD = 2.0\text{V} \sim 5.5\text{V}$ )

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Output Clock Pulse Period	tSCK	Internal SCK source	200	—	—	ns
Input Clock Pulse Period		External SCK source	200	—	—	
Output Clock High, Low Pulse Width	tSCKH, tSCKL	Internal SCK source	70	—	—	
Input Clock High, Low Pulse Width		External SCK source	70	—	—	
First Output Clock Delay Time	tFOD	Internal/External SCK source	100	—	—	
Output Clock Delay Time	tDS	—	—	—	50	
Input Setup Time	tDIS	—	100	—	—	
Input Hold Time	tDIH	—	150	—	—	



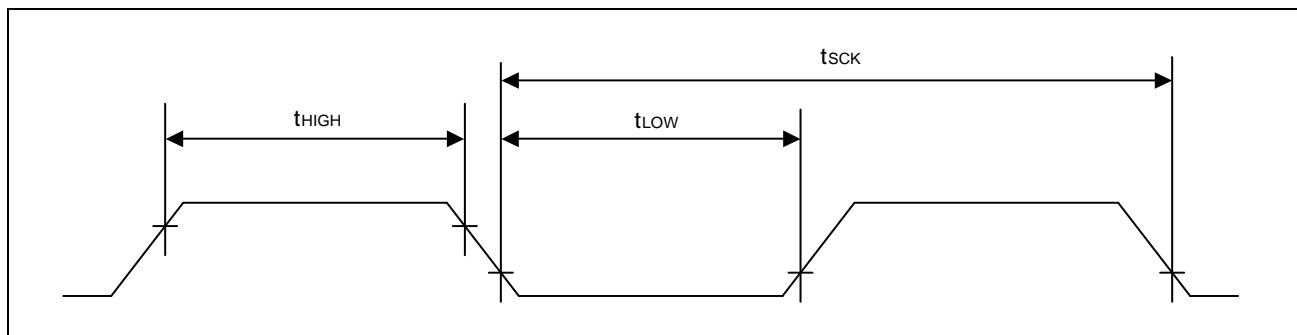
**Figure 146. SPI0/1/2 Timing**

## 21.13 UART0 Characteristics

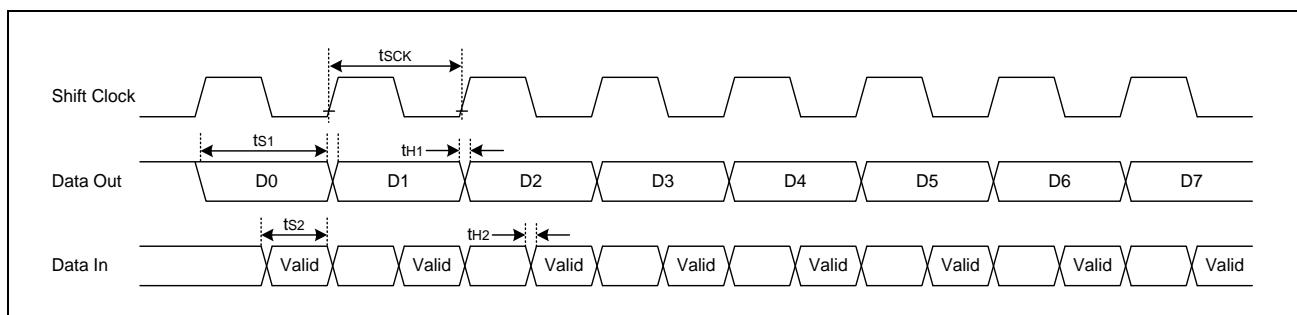
**Table 63. UART0 Characteristics**

( $T_A = -40^\circ\text{C} \sim +85^\circ\text{C}$  or  $T_A = -40^\circ\text{C} \sim 105^\circ\text{C}$ ,  $VDD = 2.0\text{V} \sim 5.5\text{V}$ ,  $f_{SCLK} = 8\text{MHz}$ )

Parameter	Symbol	MIN	Typ	MAX	Unit
Serial port clock cycle time	$t_{SCK}$	1800	$t_{CPU} \times 16$	2200	ns
Output data setup to clock rising edge	$t_{S1}$	810	$t_{CPU} \times 13$	—	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	$t_{HIGH}, t_{LOW}$	720	$t_{CPU} \times 8$	1280	ns



**Figure 147. Waveform for UART0 Timing Characteristics**



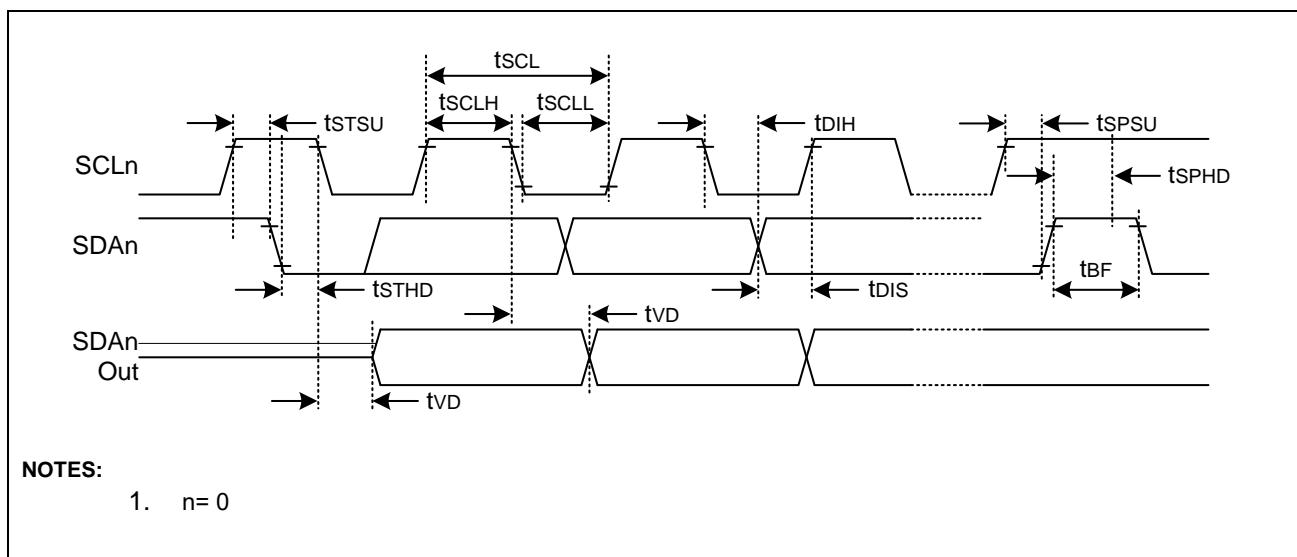
**Figure 148. Timing Waveform for the UART0 Module**

## 21.14 I<sub>2</sub>C0 Characteristics

**Table 64. I<sub>2</sub>C0 Characteristics**

( $T_A = -40^\circ\text{C} \sim +85^\circ\text{C}$  or  $T_A = -40^\circ\text{C} \sim 105^\circ\text{C}$ ,  $VDD = 2.0\text{V} \sim 5.5\text{V}$ )

<b>Parameter</b>	<b>Symbol</b>	<b>Standard Mode</b>		<b>High-Speed Mode</b>		<b>Unit</b>
		<b>MIN</b>	<b>MAX</b>	<b>MIN</b>	<b>MAX</b>	
Clock frequency	tSCL	0	100	0	400	kHz
Clock High Pulse Width	tSCLH	4.0	—	0.6	—	
Clock Low Pulse Width	tSCLL	4.7	—	1.3	—	
Bus Free Time	tBF	4.7	—	1.3	—	
Start Condition Setup Time	tSTSU	4.7	—	0.6	—	
Start Condition Hold Time	tSTHD	4.0	—	0.6	—	
Stop Condition Setup Time	tSPSU	4.0	—	0.6	—	
Stop Condition Hold Time	tSPHD	4.0	—	0.6	—	
Output Valid from Clock	tVD	0	—	0	—	
Data Input Hold Time	tDIH	0	—	0	1.0	
Data Input Setup Time	tDIS	250	—	100	—	ns



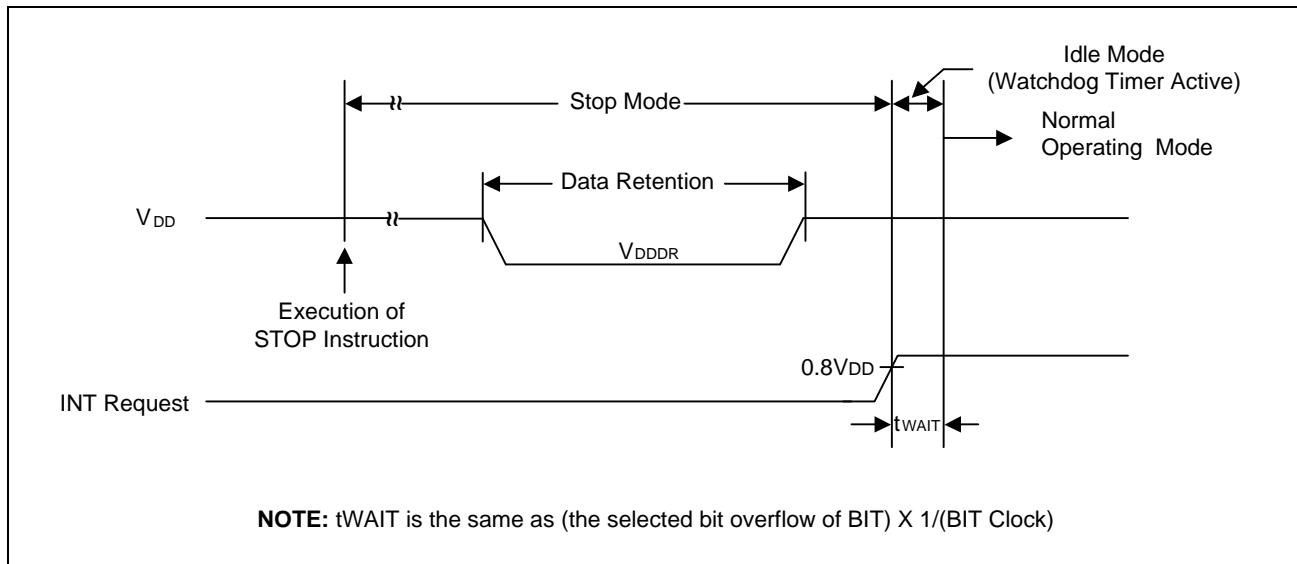
**Figure 149. I<sub>2</sub>C0 Timing**

## 21.15 Data Retention Voltage in Stop Mode

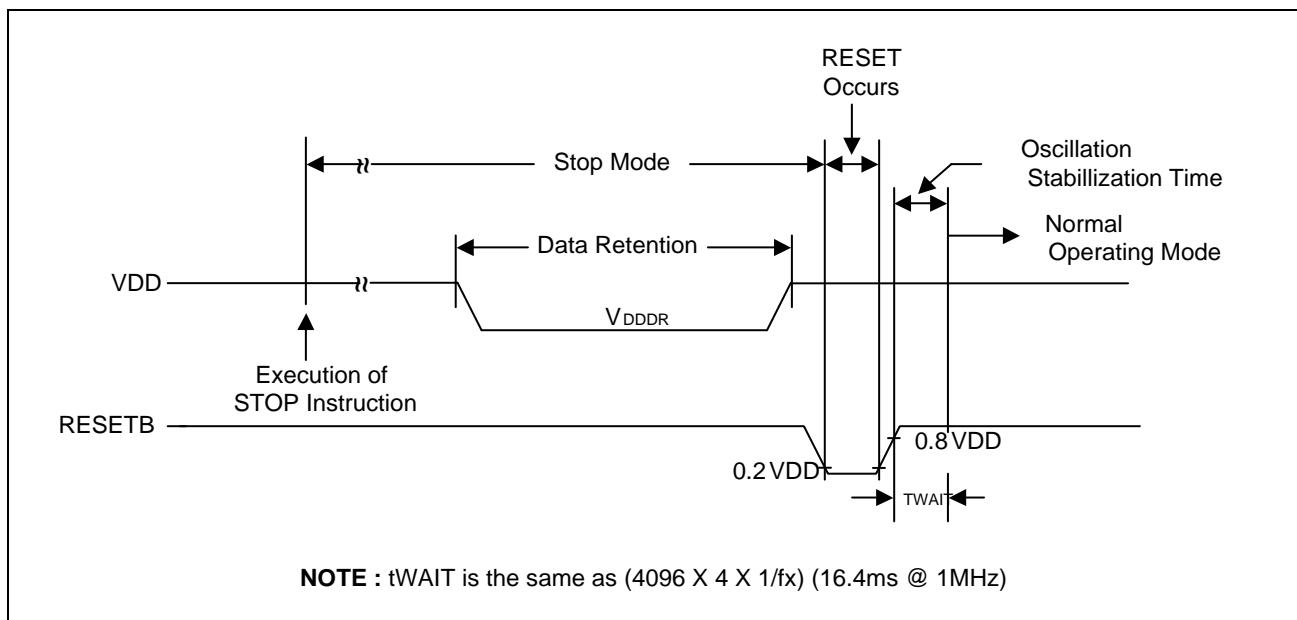
**Table 65. Data Retention Voltage in Stop Mode**

( $T_A = -40^\circ\text{C} \sim +85^\circ\text{C}$  or  $T_A = -40^\circ\text{C} \sim 105^\circ\text{C}$ ,  $V_{DD} = 2.0\text{V} \sim 5.5\text{V}$ )

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Data retention supply voltage	$V_{DDDR}$	—	2.0	—	5.5	V
Data retention supply current	$I_{DDDR}$	$V_{DDDR} = 2.0\text{V}$ , ( $T_A = 25^\circ\text{C}$ ), Stop mode	—	—	1	$\mu\text{A}$



**Figure 150. Stop Mode Release Timing when Initiated by an Interrupt**



**Figure 151. Stop Mode Release Timing when Initiated by RESETB**

## 21.16 Internal Flash Rom Characteristics

**Table 66. Internal Flash Rom Characteristics**

( $T_A = -40^\circ\text{C} \sim +85^\circ\text{C}$  or  $T_A = -40^\circ\text{C} \sim 105^\circ\text{C}$ ,  $VDD = 2.0\text{V} \sim 5.5\text{V}$ ,  $VSS = 0\text{V}$ )

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	
Code Write Protection Time	$t_{FHL}$	—	—	2.5	2.7	
Page Buffer Reset Time	$t_{FBR}$	—	—	—	5	us
Flash Programming Frequency	$f_{PGM}$	—	0.4	—	—	MHz
Endurance of Write/Erase	$NF_{WE}$	—	—	—	10,000	times

**NOTES:**

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

## 21.17 Input/Output Capacitance

**Table 67. Input / Output Capacitance**

( $T_A = -40^\circ\text{C} \sim +85^\circ\text{C}$  or  $T_A = -40^\circ\text{C} \sim 105^\circ\text{C}$ ,  $VDD = 0\text{V}$ )

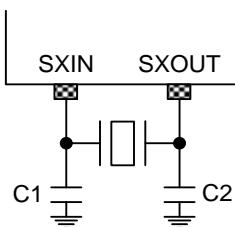
Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Input Capacitance	$C_{IN}$	$f_x = 1\text{MHz}$ Unmeasured pins are connected to VSS	—	—	10	pF
Output Capacitance	$C_{OUT}$					
I/O Capacitance	$C_{IO}$					

## 21.18 Sub Clock Oscillator Characteristics

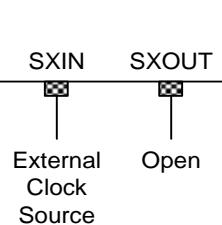
**Table 68. Sub Clock Oscillator Characteristics**

( $T_A = -40^\circ\text{C} \sim +85^\circ\text{C}$  or  $T_A = -40^\circ\text{C} \sim 105^\circ\text{C}$ ,  $VDD = 2.0\text{V} \sim 5.5\text{V}$ )

Oscillator	Parameter	Condition	MIN	TYP	MAX	Unit
Crystal	Sub oscillation frequency	$2.0\text{V} - 5.5\text{V}$	32	32.768	38	kHz
External Clock	SXIN input frequency		32	-	100	kHz



**Figure 152. Crystal Oscillator**



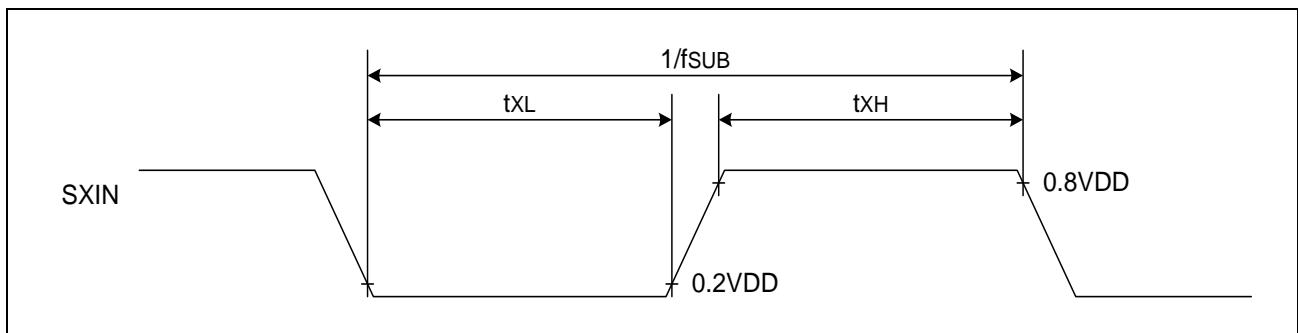
**Figure 153. External Clock**

## 21.19 Sub Oscillation Characteristics

**Table 69. Sub Oscillation Stabilization Characteristics**

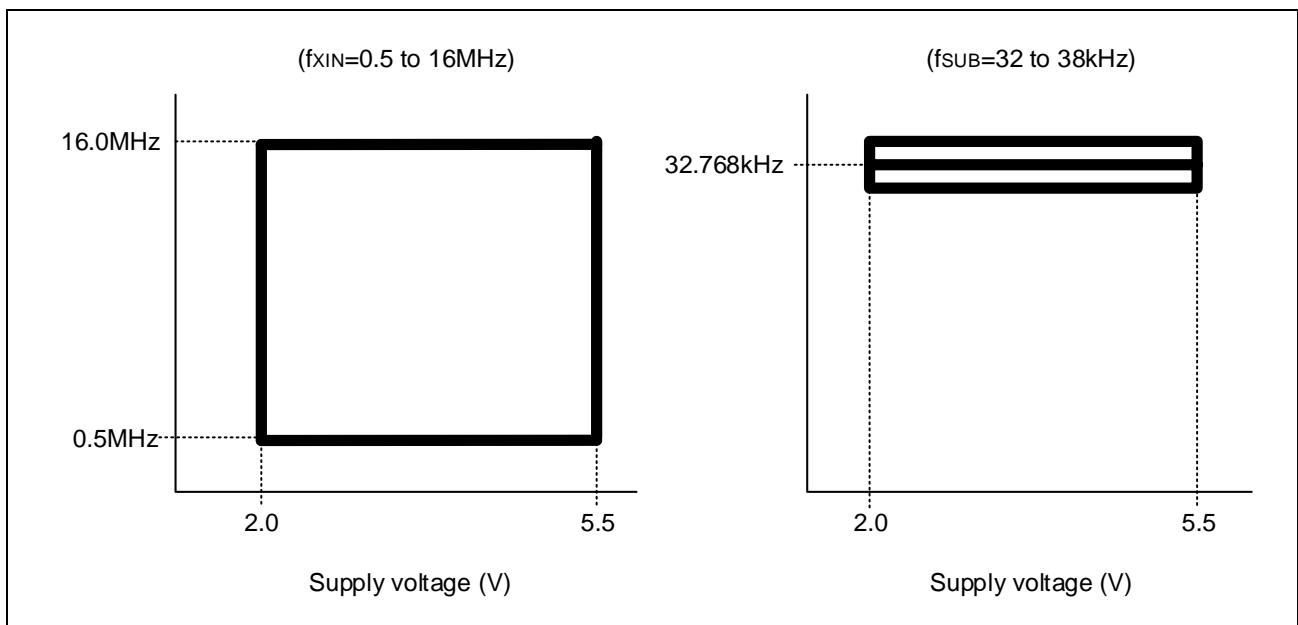
( $T_A = -40^\circ\text{C} \sim +85^\circ\text{C}$  or  $T_A = -40^\circ\text{C} \sim 105^\circ\text{C}$ ,  $VDD = 2.0\text{V} \sim 5.5\text{V}$ )

Oscillator	Parameter	MIN	TYP	MAX	Unit
Crystal	$T_A = -40^\circ\text{C} \sim +85^\circ\text{C}$	–	–	10	s
	$T_A = 25^\circ\text{C}$		500		ms
External Clock	SXIN input high and low width ( $t_{XL}$ , $t_{XH}$ )	5	–	15	us



**Figure 154. Clock Timing Measurement at SXIN**

## 21.20 Operating Voltage Range



**Figure 155. Operating Voltage Range**

## 21.21 Recommended Circuit and Layout

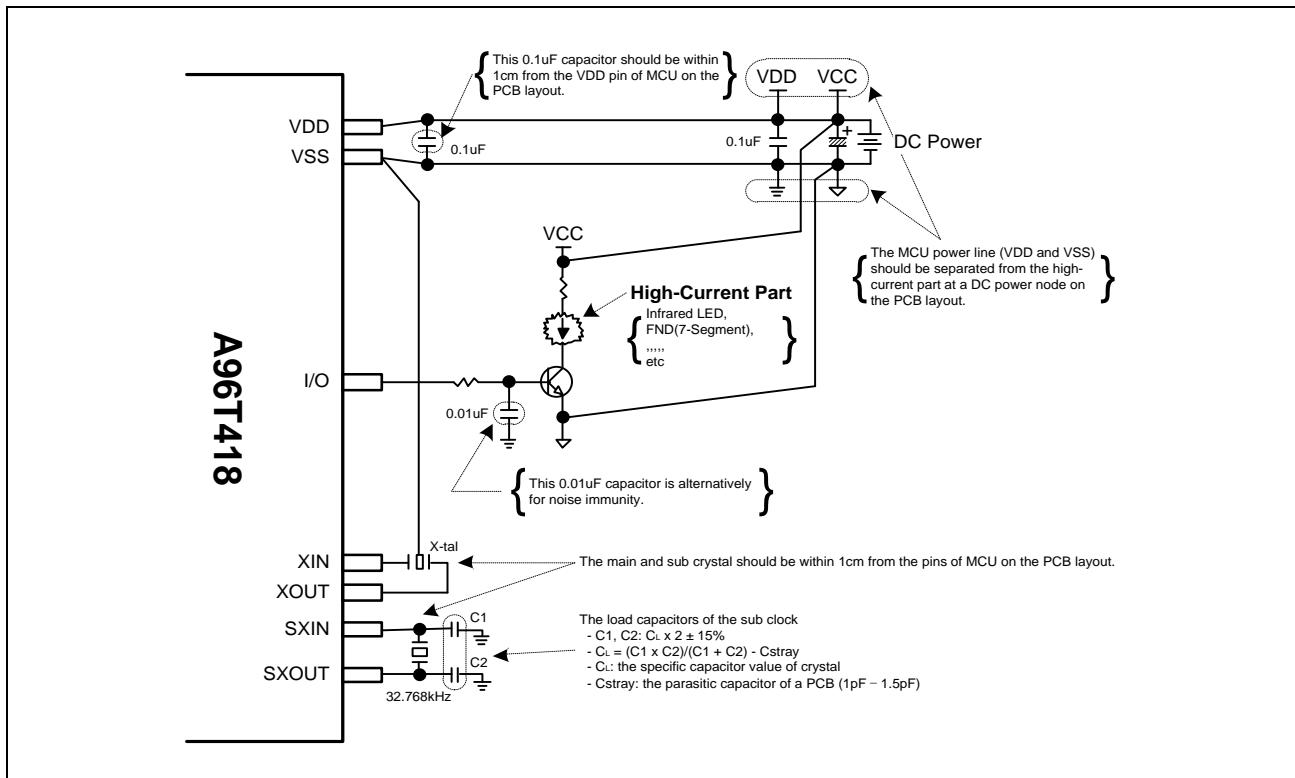


Figure 156. Recommended Voltage Range

## 21.22 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 + 3 $\sigma$ ) and (mean - 3 $\sigma$ ) respectively where  $\sigma$  is standard deviation.

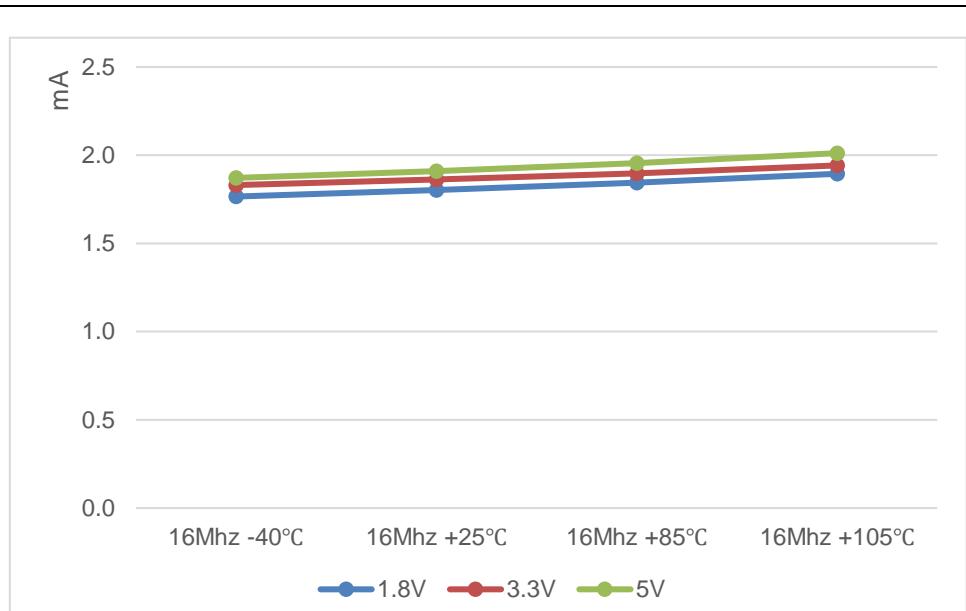


Figure 157. RUN (IDD1) Current

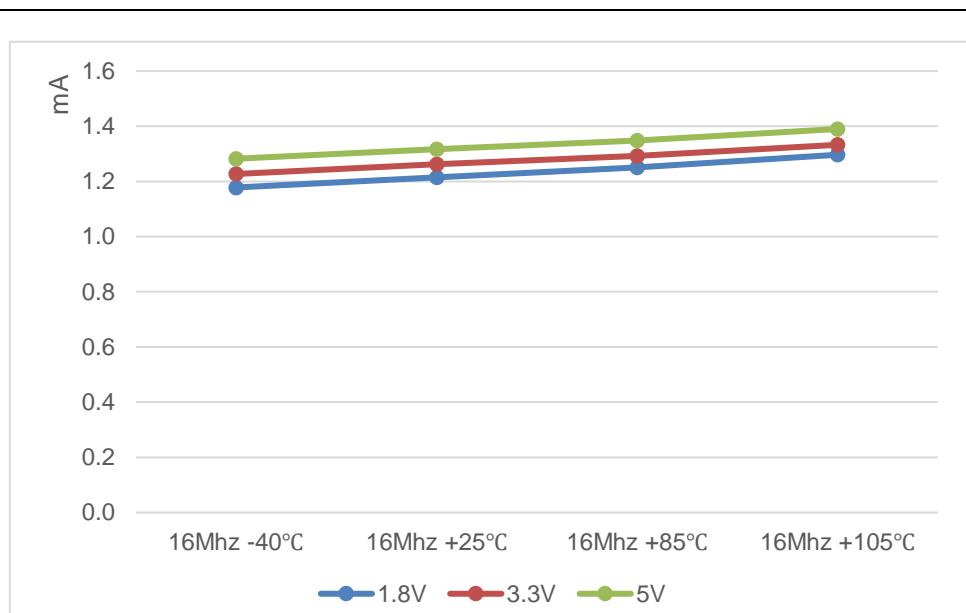


Figure 158. IDLE (IDD2) Current

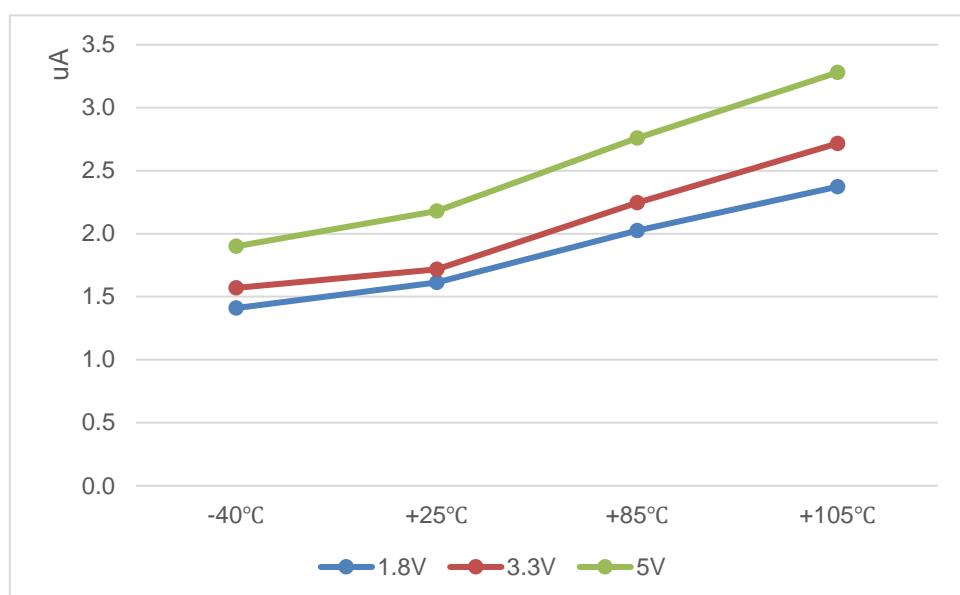


Figure 159. STOP1 (IDD3) Current

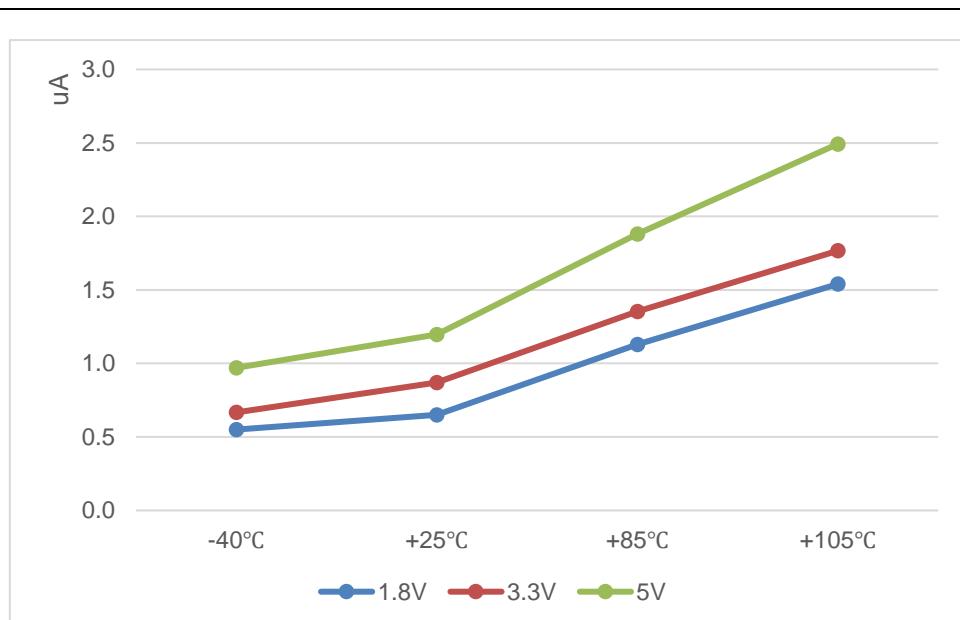


Figure 160. STOP2 (IDD4) Current

## 22 Development Tools

This chapter introduces wide range of development tools for A96T418. ABOV offers software tools, debuggers, and programmers to help a user in generating right results to match target applications. ABOV supports entire development ecosystem of the customers.

### 22.1 Compiler

ABOV Semiconductor does not provide compiler. It is recommended that you consult a compiler provider. It is recommended to consult a compiler provider. Since A96T418 has Mentor 8051 as its core, and ROM is smaller than 64Kbytes in size, a developer can use any standard 8051 compiler of other providers.

### 22.2 OCD(On-chip debugger) emulator and debugger

The OCD emulator supports ABOV's 8051 series MCU emulation. The OCD uses two wires interfacing between PC and MCU, which is attached to user's system. The OCD can read or change the value of MCU's internal memory and I/O peripherals. In addition, the OCD controls MCU's internal debugging logic. This means OCD controls emulation, step run, monitoring and many more functions regarding debugging.

The OCD debugger program runs underneath MS operating system such as MS-Windows NT/ 2000/ XP/ Vista (32-bit). If a user wants to see more details, it is recommended to refer to OCD debugger manual by visiting ABOV's website (<http://www.abovsemi.com>) and downloading debugger S/W and corresponding manuals.

- Connection:
  - DSCL (A96T418 P02 port)
  - DSDA (A96T418 P01 port)

Figure 161 shows pinouts of OCD connector.

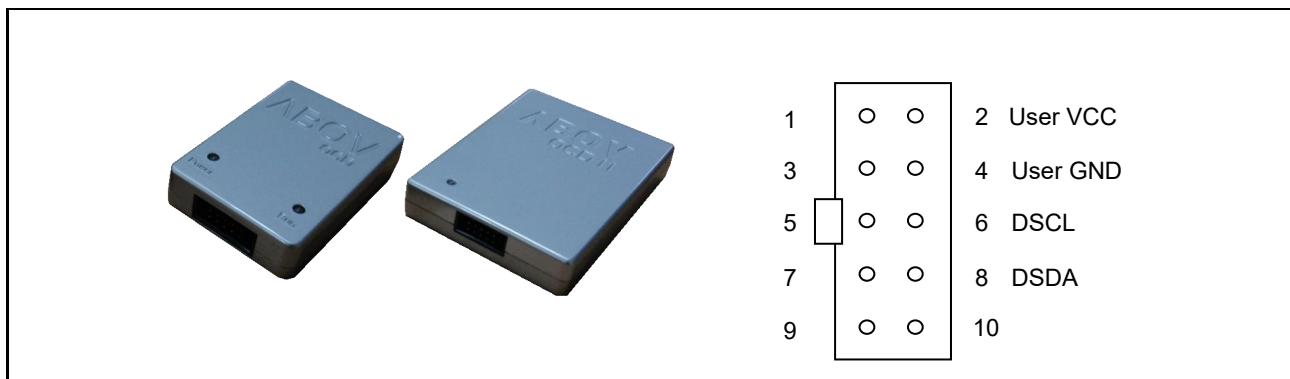


Figure 161. Debugger(OCD1/OCD2) and pin description

## 22.3 Programmer

### 22.3.1 E-PGM+

E-PGM+ USB is a single programmer. A user can program A96T418 directly using the E-PGM+.

- Support ABOV / ADAM devices
- 2~5 times faster than S-PGM+
- Main controller: 32 bit MCU @ 72MHz
- Buffer memory: 1Mbyte

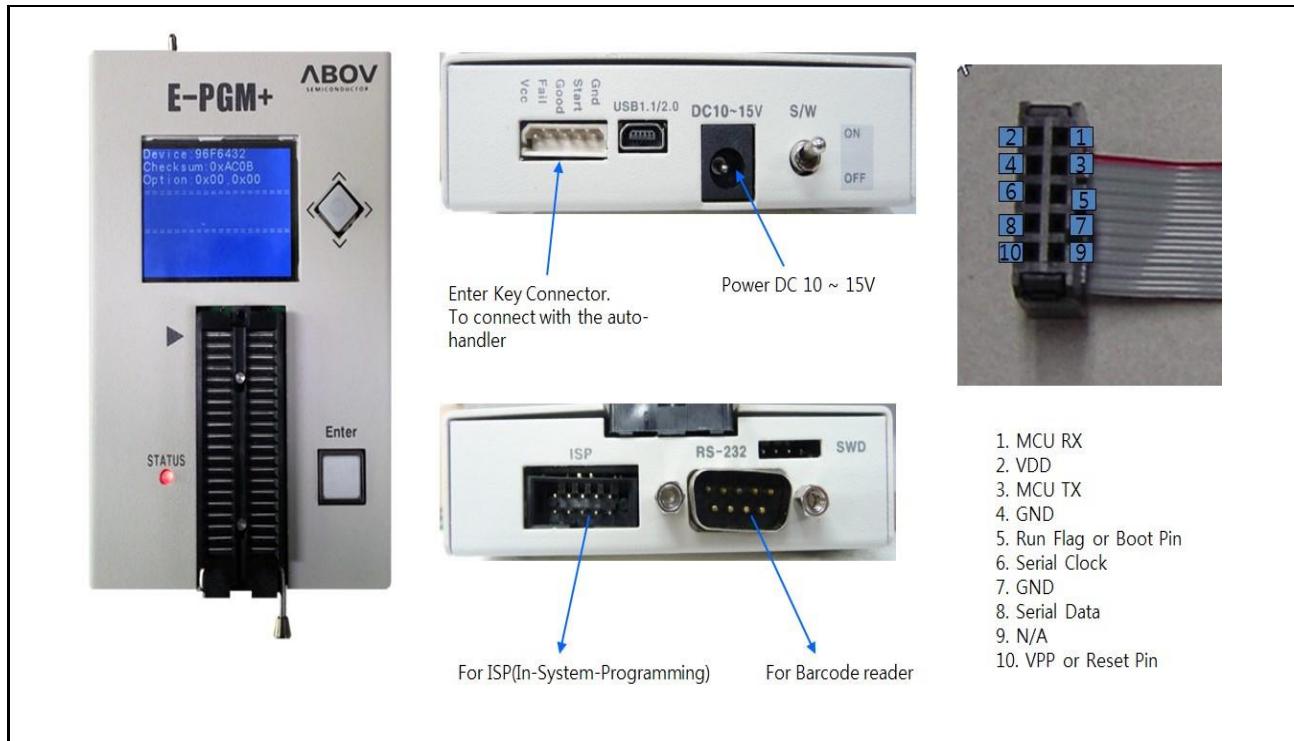


Figure 162. E-PGMplus (Single writer)

## 22.4 Flash Programming

### 22.4.1 Overview

The program memory of A96T418 is Flash Type. This flash is accessed by serial data format. There are four pins(DSCL, DSDA, VDD, and VSS) for programming/reading the flash.

**Table 70. Descriptions of pins which are used to programming/reading the Flash**

Pin name	Main chip pin name	During programming	
		I/O	Description
DSCL	P02	I	Serial clock pin. Input only pin.
DSDA	P01	I/O	Serial data pin. Output port when reading and input port when programming. Can be assigned as input/push-pull output port.
VDD, VSS	VDD, VSS	-	Logic power supply pin.

### 22.4.2 On-Board Programming

The A96T418 needs only four signal lines including VDD and VSS pins for programming FLASH with serial communication protocol. Therefore the on-board programming is possible if the programming signal lines are ready at the PCB of application board is designed.

### 22.4.3 Circuit Design Guide

When programming flash memory, the programming tool needs 4 signal lines, DSCL, DSDA, VDD, and VSS. If a user designs a PCB circuit, the user should consider the usage of these 4 signal lines for the on-board programming.

Please be careful to design the related circuit of these signal pins because rising/falling timing of DSCL and DSDA is very important for proper programming.

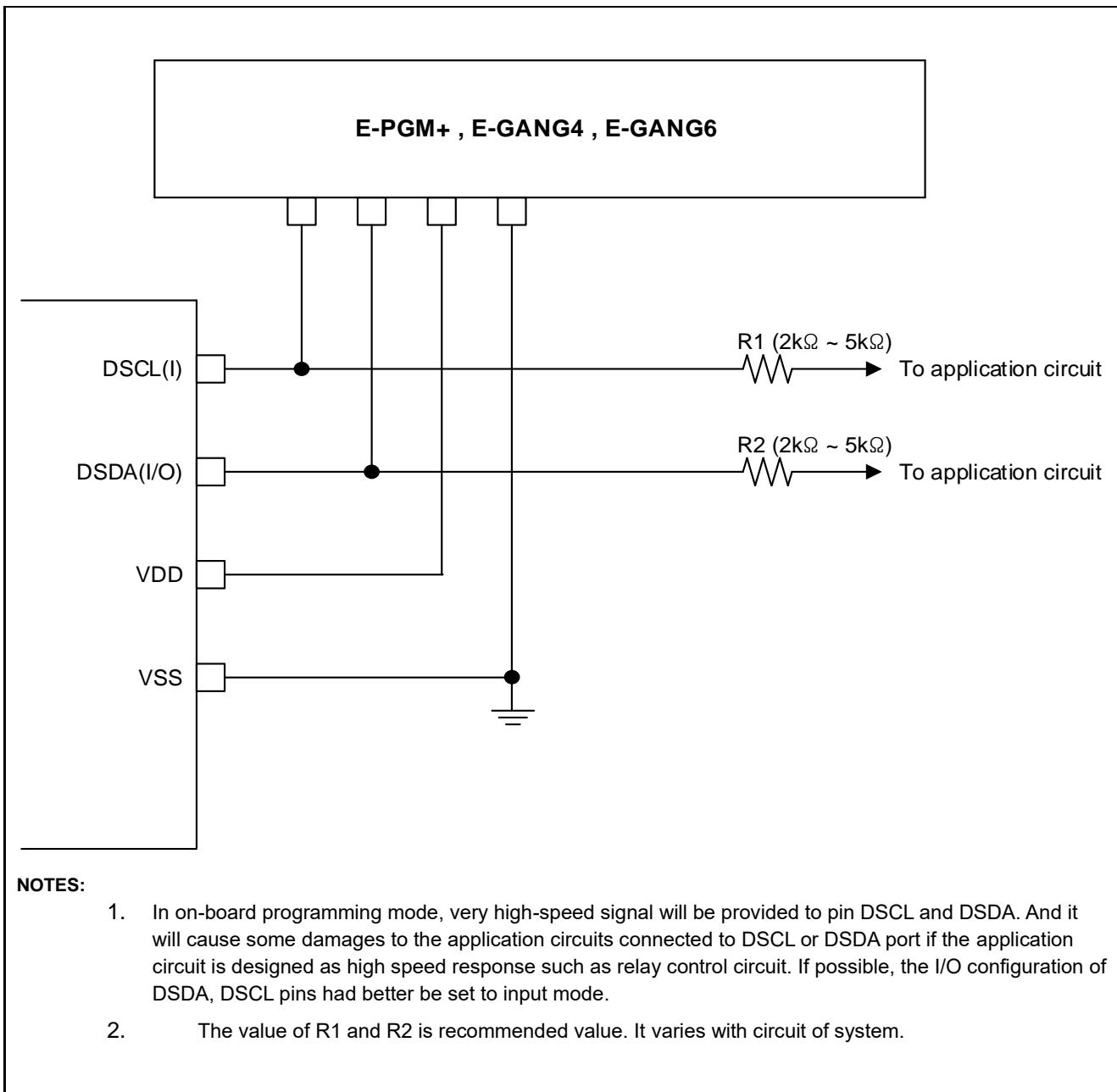


Figure 163. PCB design guide for on board programming

#### 22.4.4 OCD Emulator

OCD emulator allows a user to write code on the device too, since OCD debugger supports ISP (In System Programming). It doesn't require additional H/W, except developer's target system.

#### 22.4.5 Gang Programmer

E-Gang4 and E-Gang6 allows a user to program on multiple devices at a time. They run not only in PC controlled mode but also in standalone mode without PC control. USB interface is available and it is easy to connect to the handler.



Figure 164. E-GANG4 and E-GANG6 (for Mass Production)

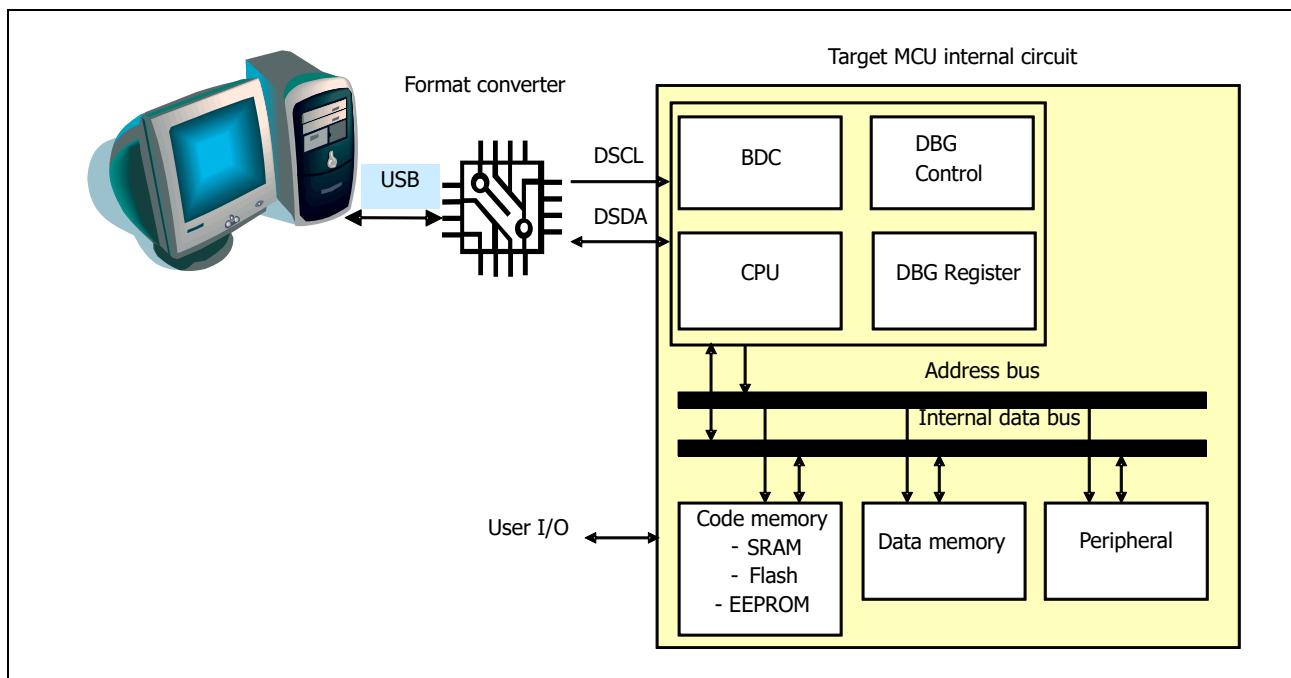
## 22.5 On-chip Debug System

A96T418 supports On-Chip Debug (OCD) system. On-chip debug system of A96T418 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 this section. Features of the OCD is introduced in Table 71.

**Table 71. Descriptions of pins which are used to programming/reading the Flash**

Two wire external interface	<ul style="list-style-type: none"> <li>• 1 for serial clock input</li> <li>• 1 for bi-directional serial data bus</li> </ul>
Debugger accesses	<ul style="list-style-type: none"> <li>• All internal peripherals</li> <li>• Internal data RAM</li> <li>• Program Counter</li> <li>• Flash memory and data EEPROM memory</li> </ul>
Extensive On-Chip Debugging supports for Break Conditions	<ul style="list-style-type: none"> <li>• Break instruction</li> <li>• Single step break</li> <li>• Program memory break points on single address</li> <li>• Programming of Flash, EEPROM, Fuses, and Lock bits through the two-wire interface</li> <li>• On-Chip Debugging supported by Dr. Choice®</li> </ul>
Operating frequency	The maximum frequency of a target MCU

Figure 165 shows a block diagram of the OCD interface and the OCD system.



**Figure 165. Block Diagram of On-Chip Debug System**

### 22.5.1 Two-Pin External Interface

#### Basic Transmission Packet

17. 10-bit packet transmission using two-pin interface.
18. packet consists of 8-bit data, 1-bit parity and 1-bit acknowledge.
19. Parity is even of '1' for 8-bit data in transmitter.
20. Receiver generates acknowledge bit as '0' when transmission for 8-bit data and its parity has no error.
21. When transmitter has no acknowledge (Acknowledge bit is '1' at tenth clock), error process is executed in transmitter.
22. When acknowledge error is generated, host PC makes stop condition and transmits command which has error again.
23. Background debugger command is composed of a bundle of packet.
24. Start condition and stop condition notify the start and the stop of background debugger command respectively.

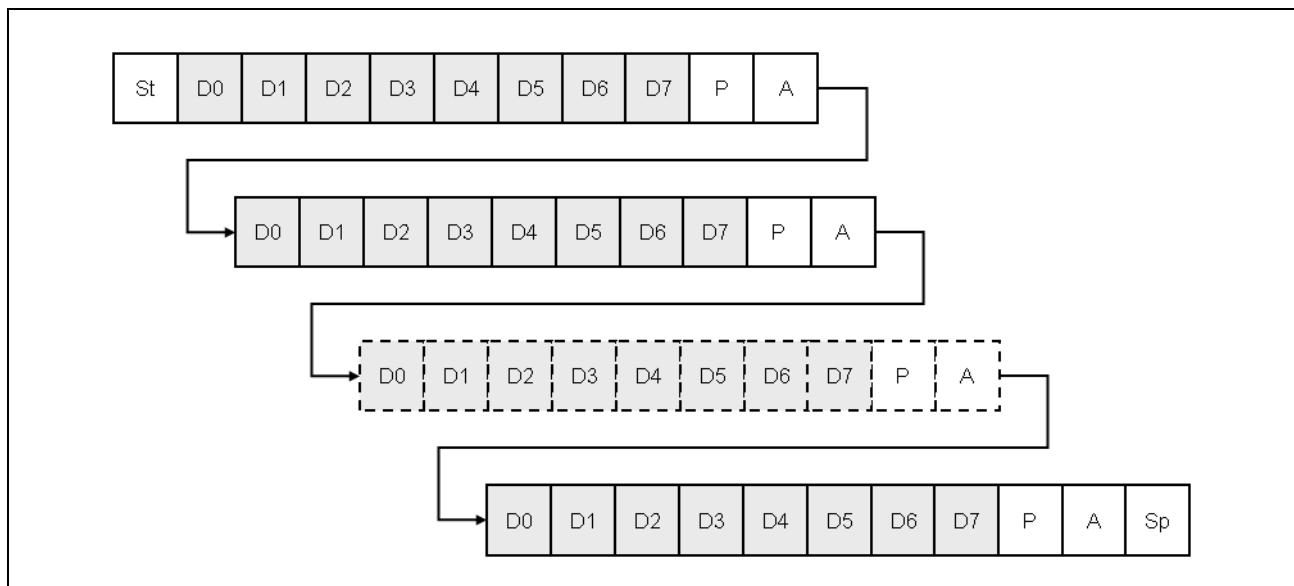


Figure 166. 10-bit Transmission Packet

### Packet Transmission Timing

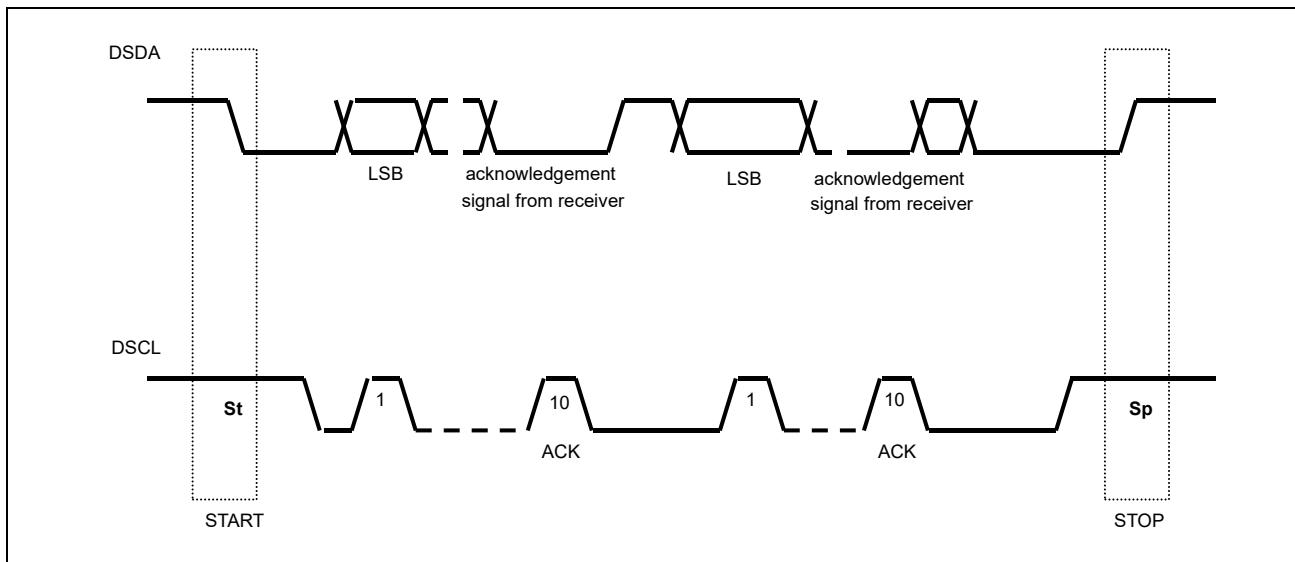


Figure 167. Data Transfer on the Twin Bus

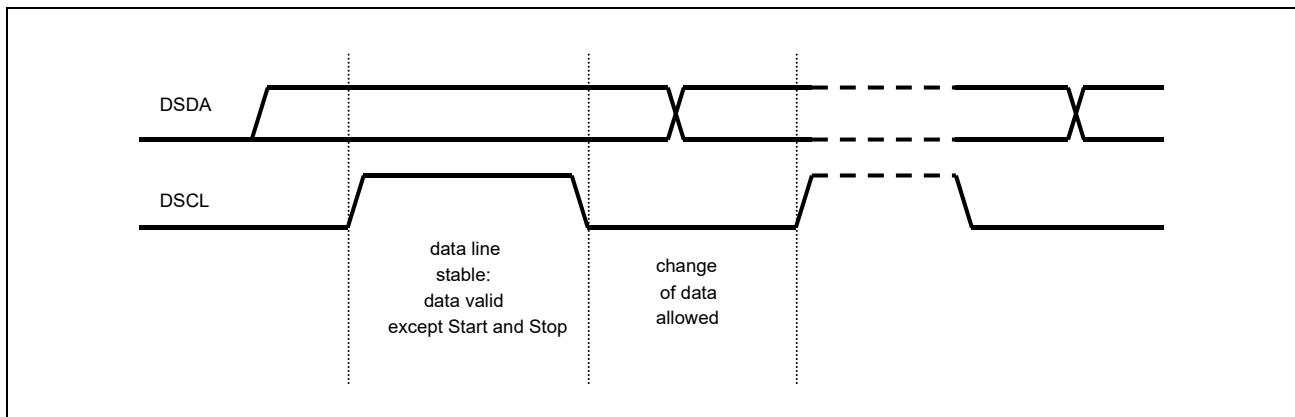


Figure 168. Bit Transfer on the Serial Bus

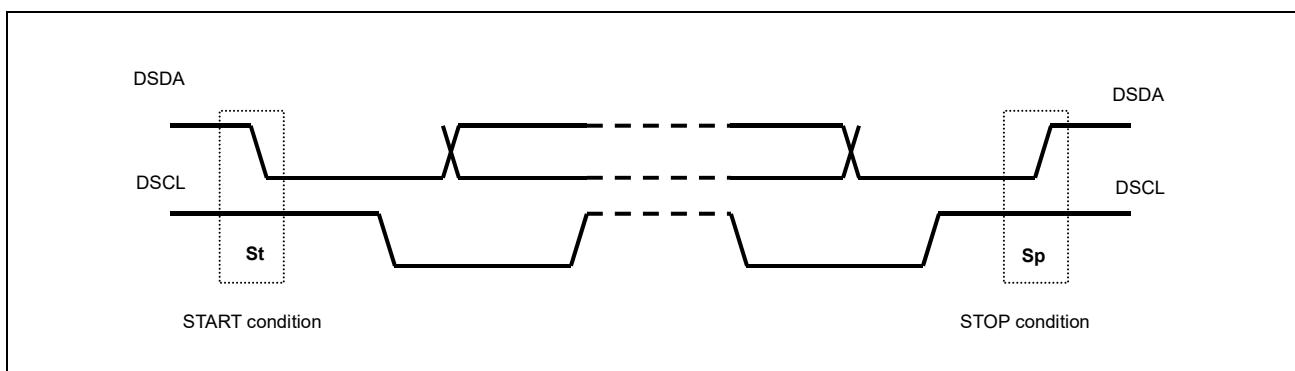


Figure 169. Start and Stop Condition

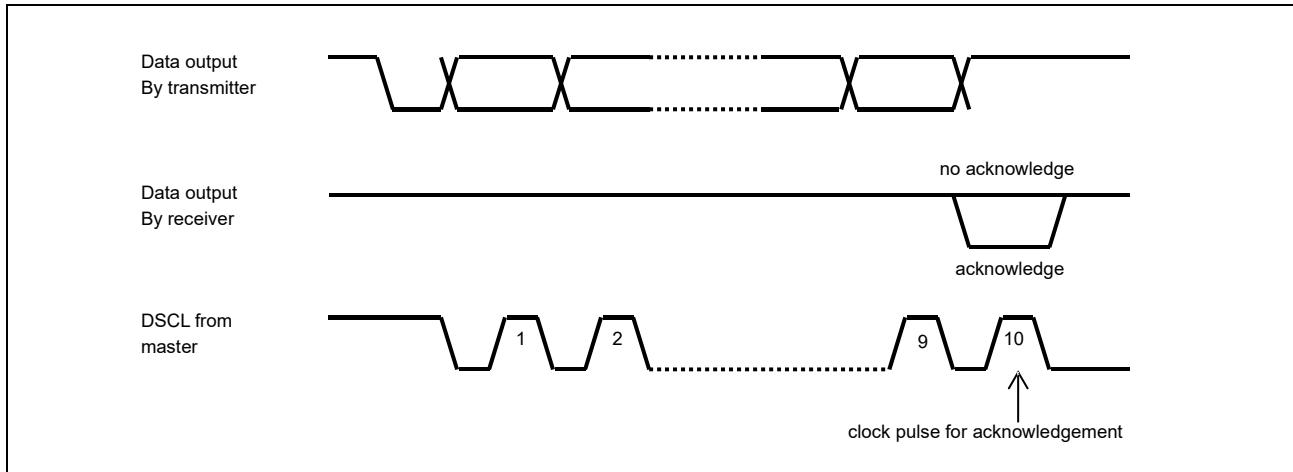


Figure 170. Acknowledge on the Serial Bus

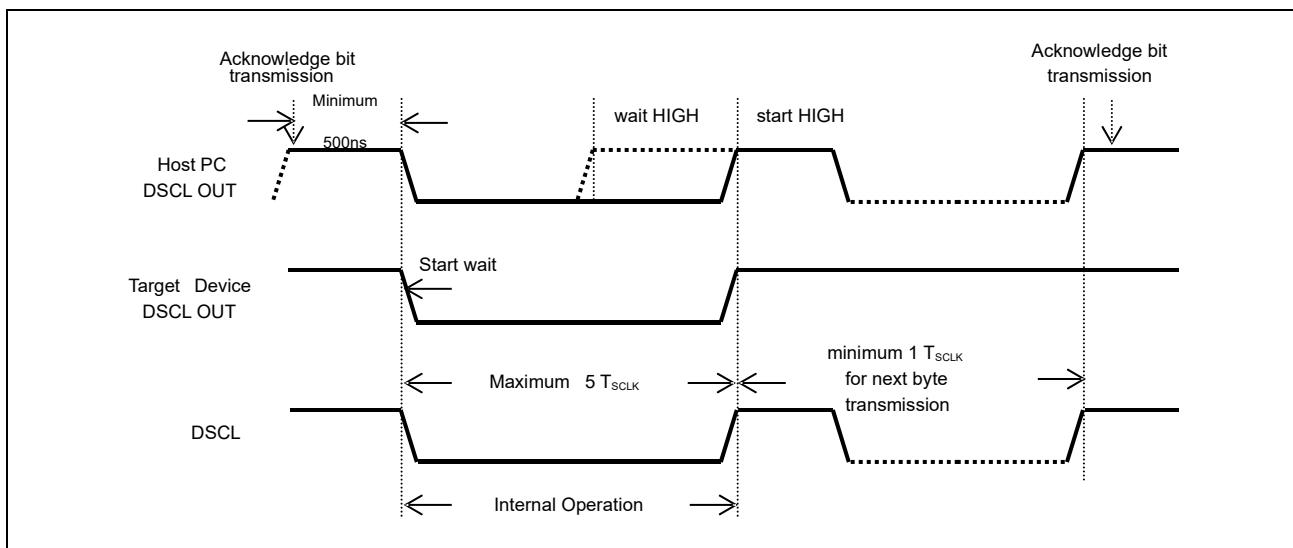
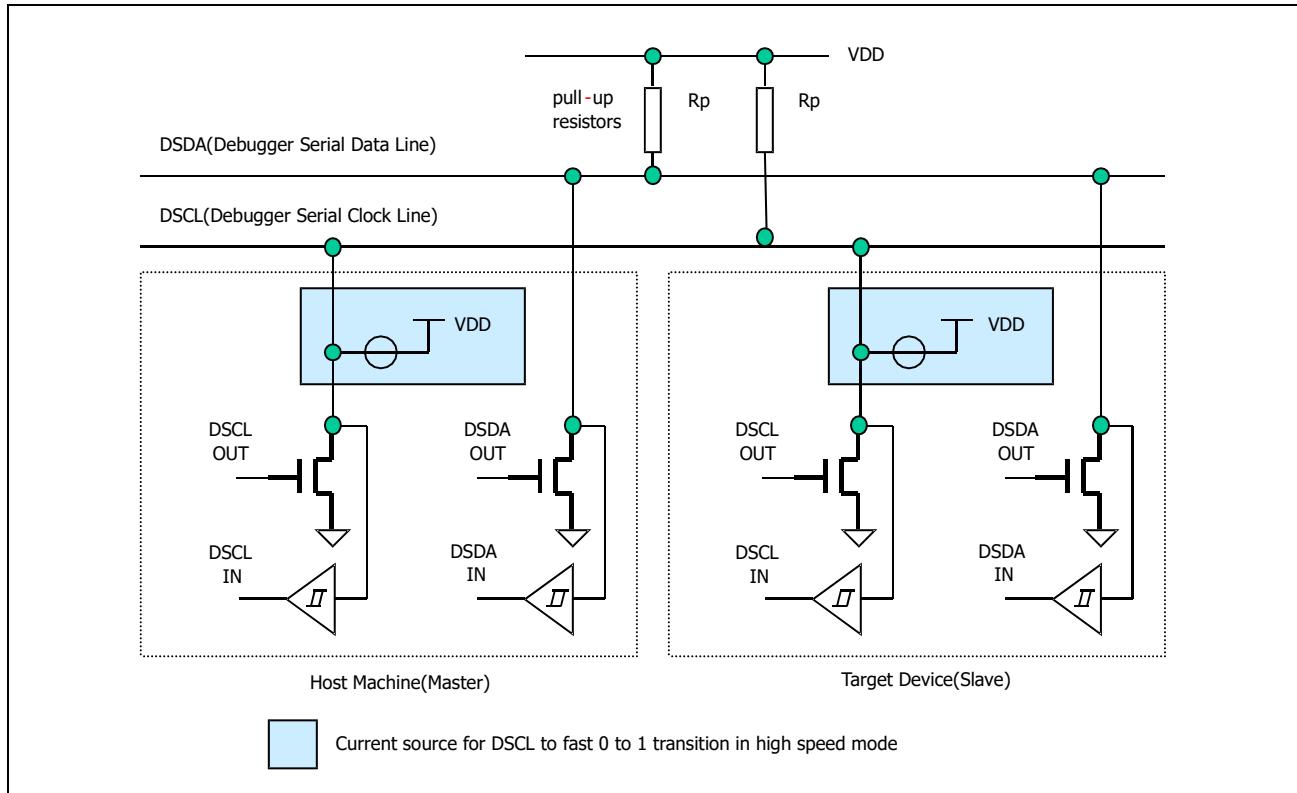


Figure 171. Clock Synchronization during Wait Procedure

### Connection of Transmission

Two-pin interface connection uses open-drain(wire-AND bidirectional I/O).



**Figure 172. Connection of Transmission**

## 23 Package Diagram

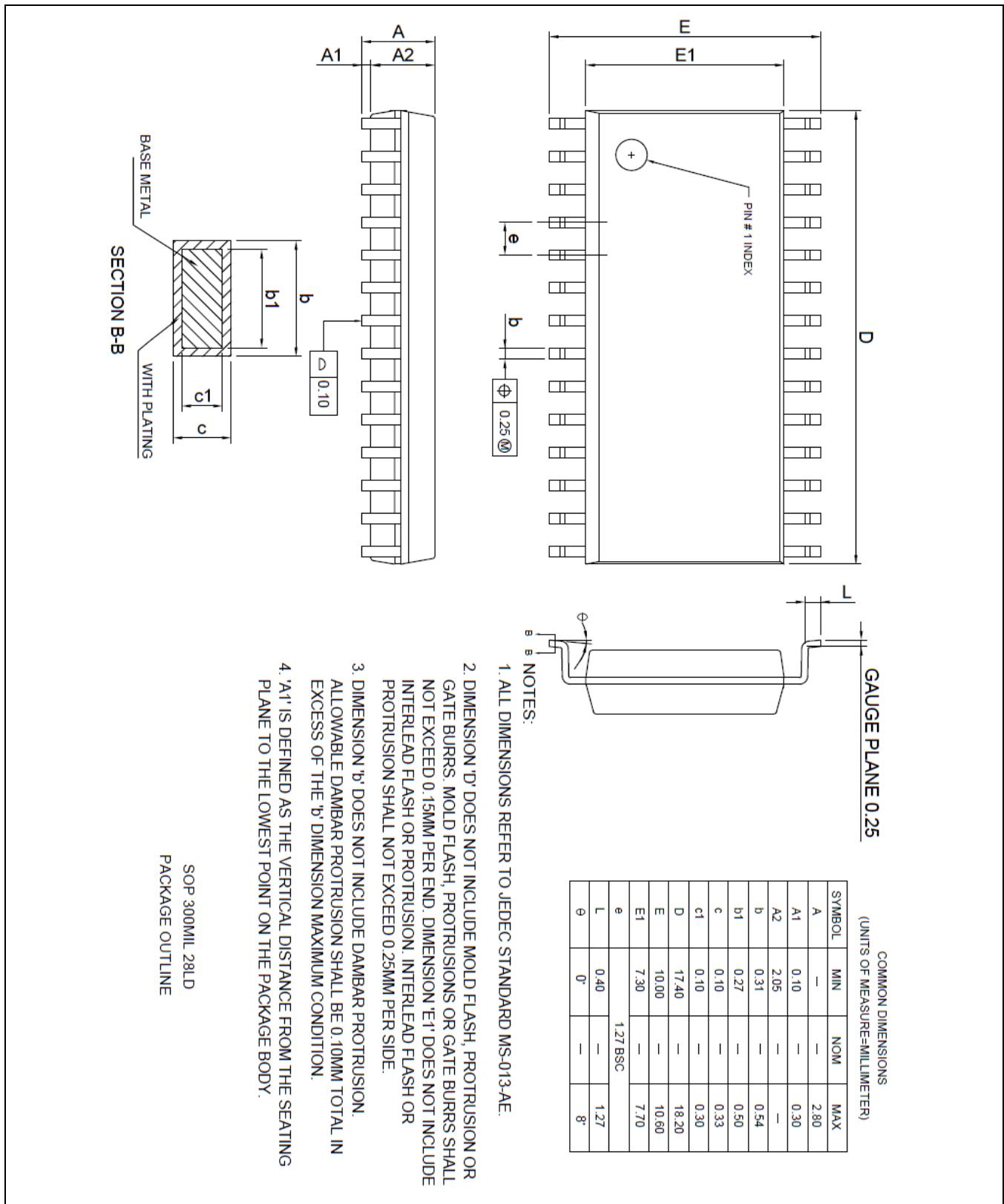


Figure 173. 28-Pin SOP Package

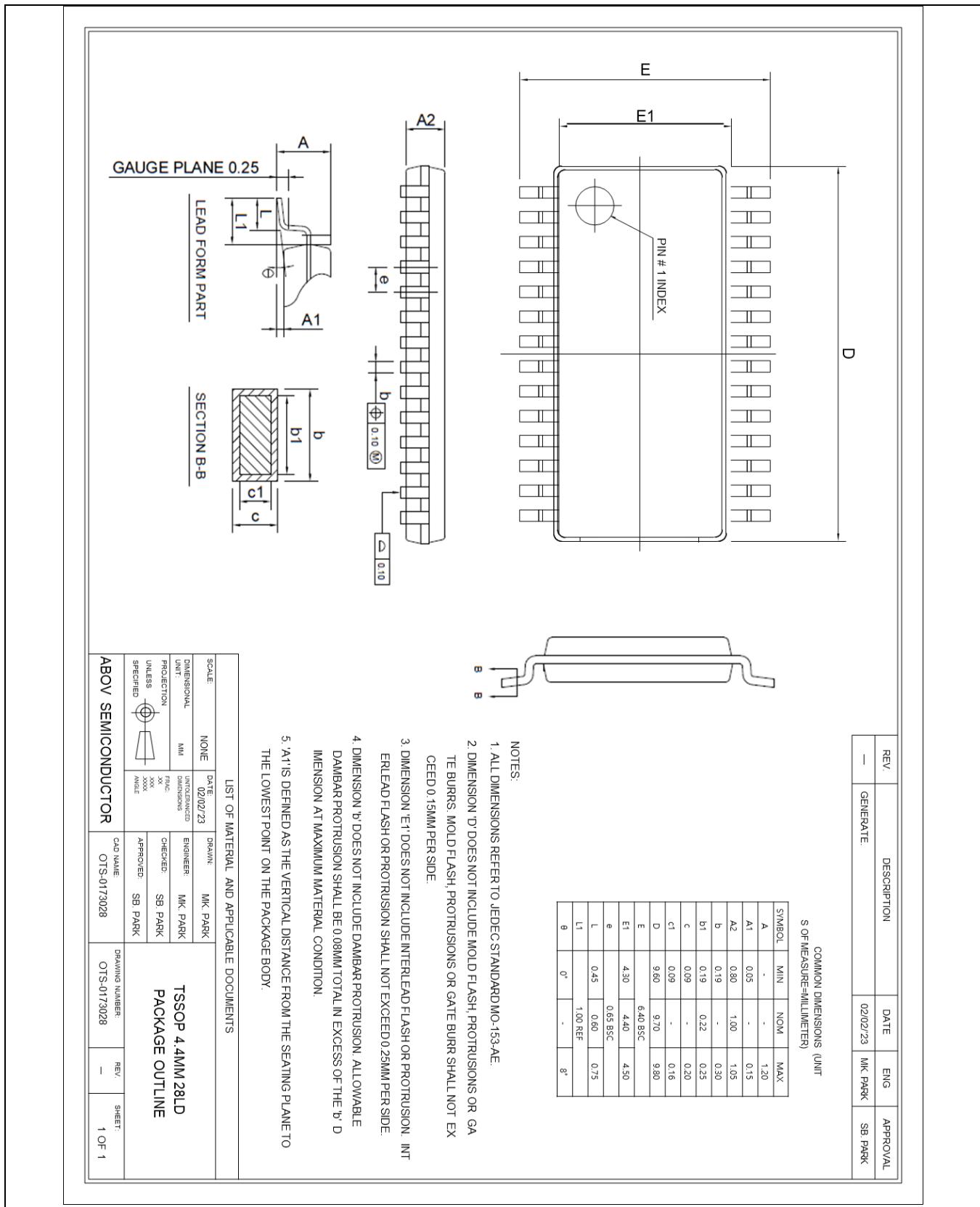


Figure 174. 28-Pin TSSOP Package

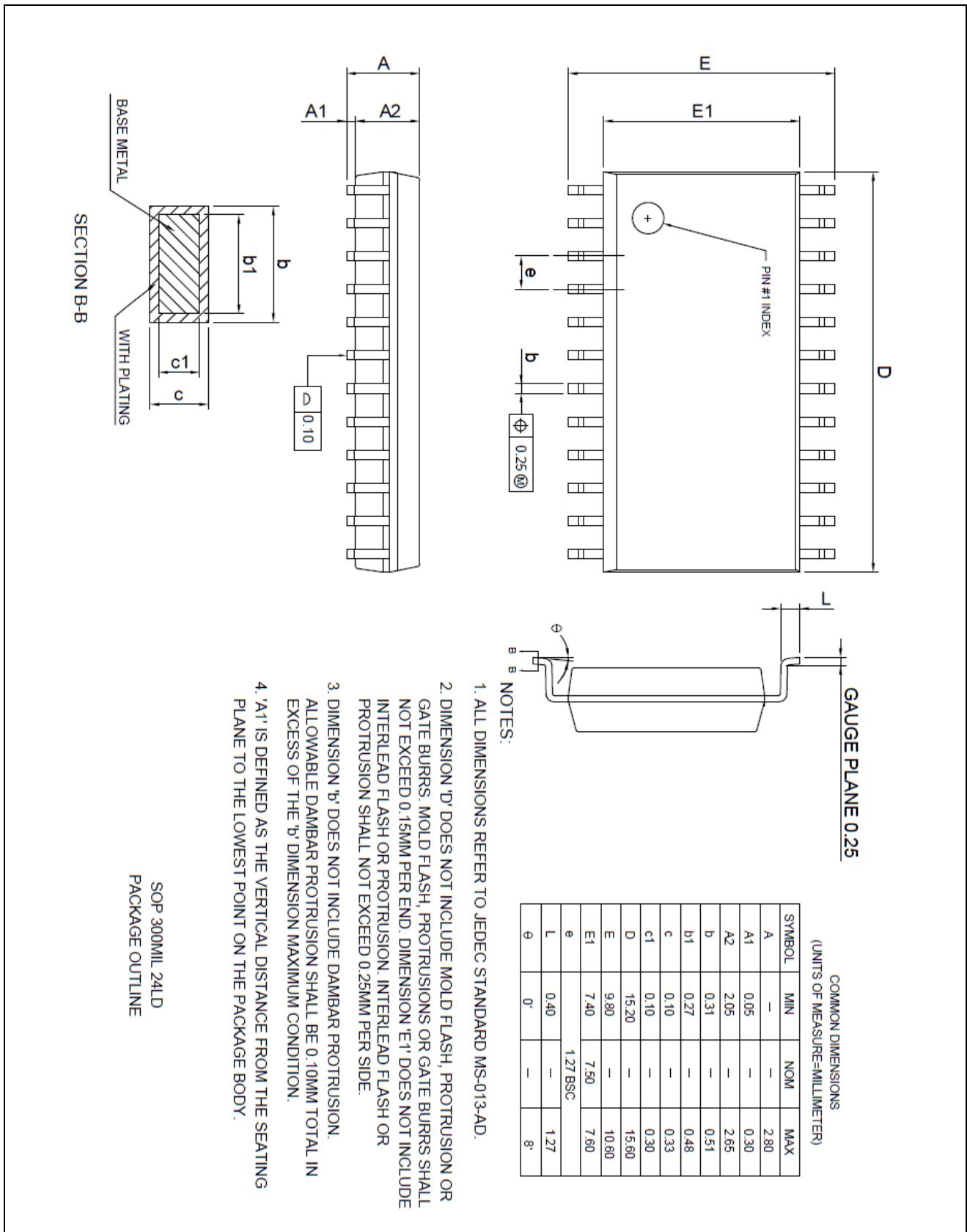


Figure 175. 24-Pin SOP Package

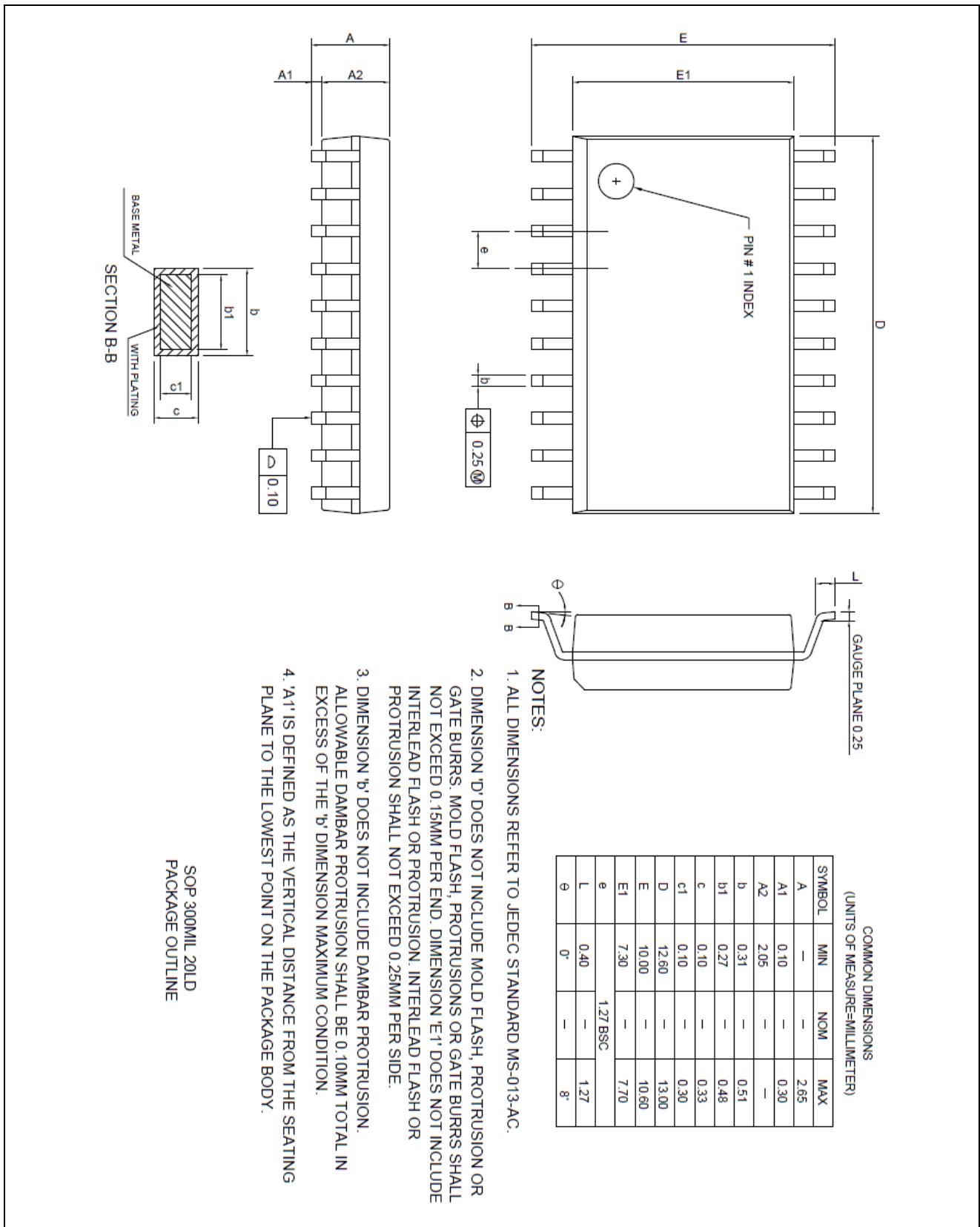


Figure 176. 20-Pin SOP Package

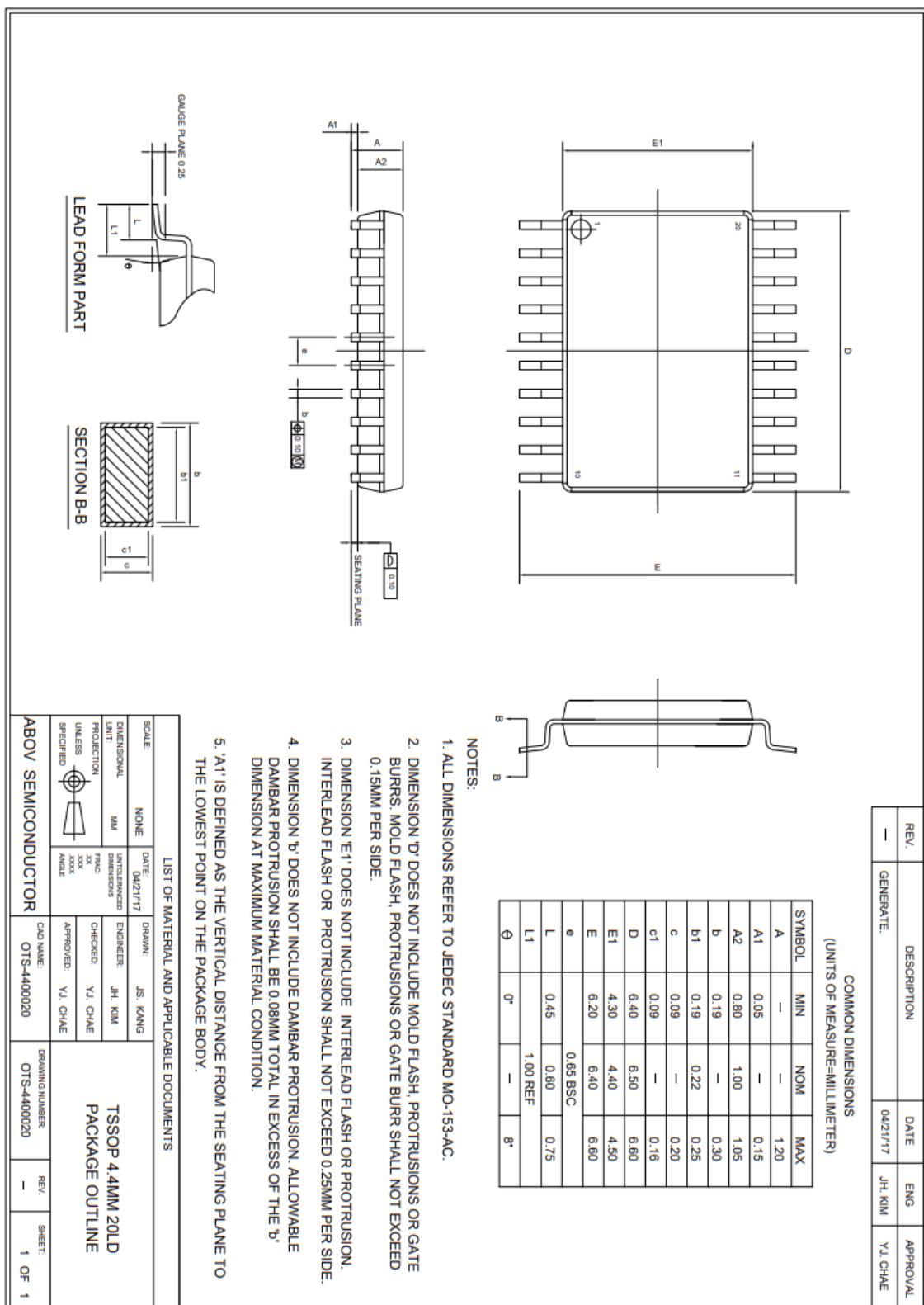


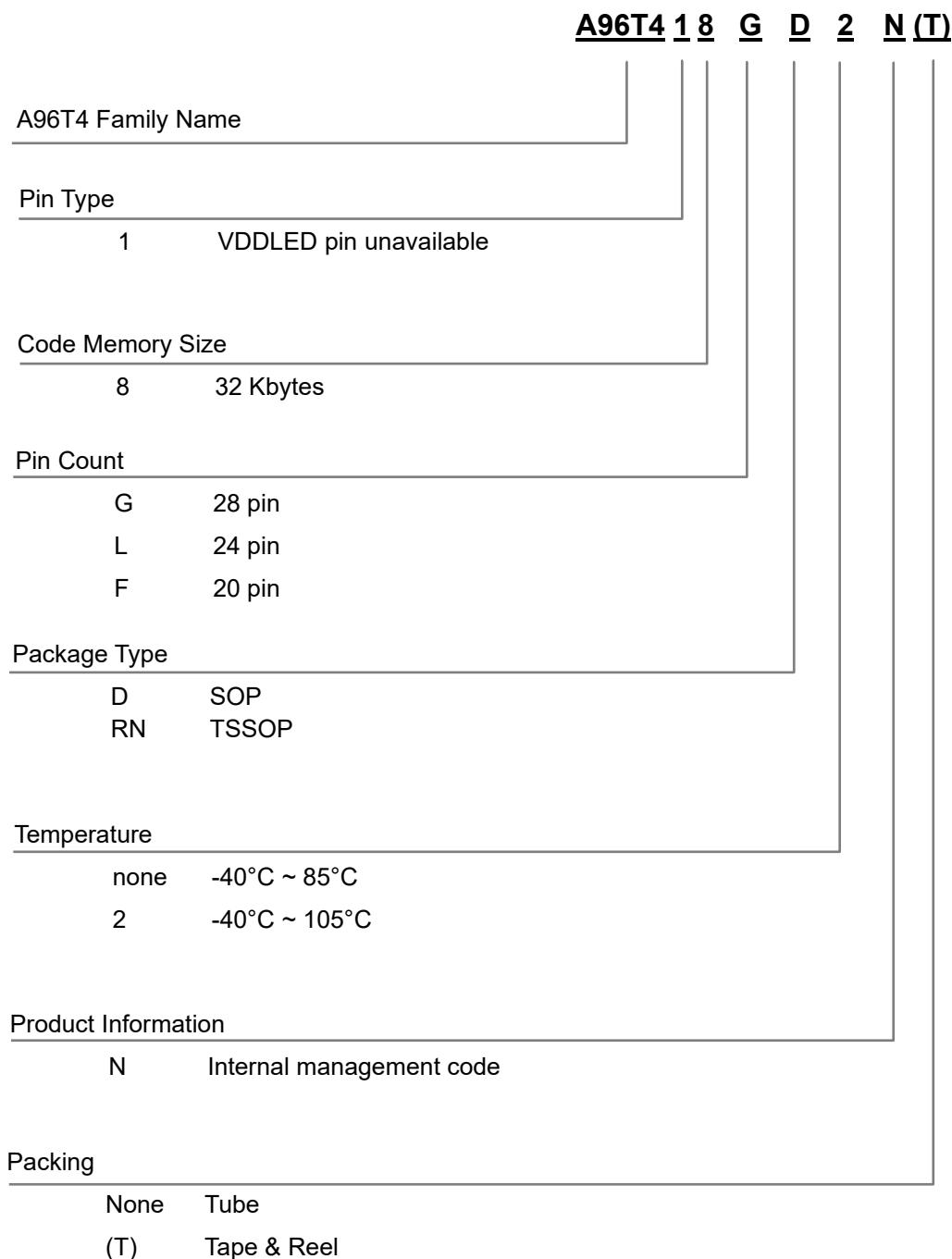
Figure 177. 20-Pin TSSOP Package

## 24 Ordering Information

**Table 72. Ordering Information of A96T418**

Device Name	FLASH	XRAM	IRAM	Touch Switch	LED Driver	ADC	I/O PORT	Package	Temperature Range
A96T418GDN	32Kbytes	1792bytes	256bytes	20-ch	8COM / 16SEG	8inputs	25	28 SOP	-40°C ~ 85°C
A96T418GRN				20-ch	8COM / 16SEG	8inputs	25	28 TSSOP	
A96T418LDN*				17-ch	8COM / 16SEG	8inputs	21	24 SOP	
A96T418FDN*				13-ch	8COM / 15SEG	8inputs	17	20 SOP	
A96T418FRN*				13-ch	8COM / 15SEG	8inputs	17	20 TSSOP	
A96T418GD2N	32Kbytes	1792bytes	256bytes	20-ch	8COM / 16SEG	8inputs	25	28 SOP	-40°C ~ 105°C
A96T418LD2N				17-ch	8COM / 16SEG	8inputs	21	24 SOP	
A96T418FD2N				13-ch	8COM / 15SEG	8inputs	17	20 SOP	

\* For available options or further information on the device with an “\*” mark, please contact [the ABOV sales office](#).



**Figure 178. Device Nomenclature**

## Appendix

### Instruction Table

Instructions are either 1, 2 or 3bytes 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.

<b>ARITHMETIC</b>				
<b>Mnemonic</b>	<b>Description</b>	<b>Bytes</b>	<b>Cycles</b>	<b>Hex code</b>
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
DAA	Decimal Adjust A	1	1	D4

<b>LOGICAL</b>				
<b>Mnemonic</b>	<b>Description</b>	<b>Bytes</b>	<b>Cycles</b>	<b>Hex code</b>
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

<b>DATA TRANSFER</b>				
<b>Mnemonic</b>	<b>Description</b>	<b>Bytes</b>	<b>Cycles</b>	<b>Hex code</b>
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

<b>BOOLEAN</b>				
<b>Mnemonic</b>	<b>Description</b>	<b>Bytes</b>	<b>Cycles</b>	<b>Hex code</b>
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

<b>BRANCHING</b>				
<b>Mnemonic</b>	<b>Description</b>	<b>Bytes</b>	<b>Cycles</b>	<b>Hex code</b>
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	2	2	D8-DF
DJNZ dir,rel	Decrement direct byte, jnz relative	3	2	D5

<b>MISCELLANEOUS</b>				
<b>Mnemonic</b>	<b>Description</b>	<b>Bytes</b>	<b>Cycles</b>	<b>Hex code</b>
NOP	No operation	1	1	00

<b>ADDITIONAL INSTRUCTIONS (selected through EO[7:4])</b>				
<b>Mnemonic</b>	<b>Description</b>	<b>Bytes</b>	<b>Cycles</b>	<b>Hex code</b>
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.

## Revision History

Version	Date	Revision list
1.00	2020.07.15	The First Release.
1.01	2020.11.11	detailed DC Characteristics of LED drive current.
1.02	2020.12.01	Circuit for LED and Touch by common pin and recommended diodes added.
1.03	2021.02.10	Figure of Configuration Timing when Power-on updated. HSIRC tolerance table updated.
1.04	2021.04.28	HSIRC tolerance table updated.
1.05	2021.10.28	P0FSRL reg. and LED driver of Feature table updated.
1.06	2021.11.23	Buzzer block diagram, frequency table, and BUZCR reg. updated.
1.07	2022.07.22	24SOP/20SOP P31/SEG14/RXD1 pin name updated. FESR (Flash Status Register) explanation updated.
1.08	2022.11.21	Updated font style of this document.
1.09	2022.12.12	SEG Output High voltage change to LED SEG output current. (Change voltage notation to current notation) Change LED Example Circuit. Change LED COM / SEG notation.
1.10	2022.02.02	A96T418 GRN 28TSSOP package added.
1.11	2023.07.20	A96T418 FRN 20TSSOP package added.
1.12	2023.12.22	105°C Temperature specification added.
1.13	2024.12.02	Updated the disclaimer.

**Korea****Regional Office, Seoul**

R&D, Marketing & Sales  
8th Fl., 330, Yeongdong-daero,  
Gangnam-gu, Seoul,  
06177, Korea

Tel: +82-2-2193-2200  
Fax: +82-2-508-6903  
<http://www.abovsemi.com/>

**HQ, Ochang**

R&D, QA, and Test Center  
93, Gangni 1-gil, Ochang-eup, Cheongwon-  
gun,  
Chungcheongbuk-do,  
28126, Korea

Tel: +82-43-219-5200  
Fax: +82-43-217-3534  
<http://www.abovsemi.com/>

**Domestic Sales Manager**

Tel: +82-2-2193-2206  
Fax: +82-2-508-6903  
Email: [sales\\_kr@abov.co.kr](mailto:sales_kr@abov.co.kr)

**Global Sales Manager**

Tel: +82-2-2193-2281  
Fax: +82-2-508-6903  
Email: [sales\\_gl@abov.co.kr](mailto:sales_gl@abov.co.kr)

**China Sales Manager**

Tel: +86-755-8287-2205  
Fax: +86-755-8287-2204  
Email: [sales\\_cn@abov.co.kr](mailto:sales_cn@abov.co.kr)

**ABOV Disclaimer****IMPORTANT NOTICE – PLEASE READ CAREFULLY**

ABOV Semiconductor (“ABOV”) reserves the right to make changes, corrections, enhancements, modifications, and improvements to ABOV products and/or to this document at any time without notice. **ABOV DOES NOT GIVE WARRANTIES AS TO THE ACCURACY OR COMPLETENESS OF THE INFORMATION INCLUDED HEREIN.** Purchasers should obtain the latest relevant information of ABOV products before placing orders. Purchasers are entirely responsible for the choice, selection, and use of ABOV products and ABOV assumes no liability for application assistance or the design of purchasers' products. **NO LICENSE, EXPRESS OR IMPLIED, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY ABOV HEREIN.** ABOV **DISCLAIMES ALL EXPRESS AND IMPLIED WARRANTIES AND SHALL NOT BE RESPONSIBLE OR LIABLE FOR ANY INJURIES OR DAMAGES RELATED TO USE OF ABOV PRODUCTS IN SUCH UNAUTHORIZED APPLICATIONS.** ABOV and the ABOV logo are trademarks of ABOV. For additional information about ABOV trademarks, please refer to [https://www.abov.co.kr/en/about/corporate\\_identity.php](https://www.abov.co.kr/en/about/corporate_identity.php). All other product or service names are the property of their respective owners. Information in this document supersedes and replaces the information previously supplied in any former versions of this document.

**© 2020 ABOV Semiconductor – All rights reserved**