

Ultra Low Power Sub-1GHz Wireless MCU

MCU Features

- 1-T 80C51 CPU platform
- 16kB program area Flash with password access protects. Default space configuration:
 - AP program space (13.5 kB, 0000h ~ 35FFh)
 - IAP data space (1.0 kB, 3600h ~ 39FFh)
 - ISP boot code space (1.5 kB, 3A00h ~ 3FFFh)
- 1 kB data memory
 - 256-byte high-speed buffer
 - 768-byte of extended RAM (XRAM)
 - Extended RAM (XRAM) supporting page access
- On-chip debug interface (OCD)
- Multiple power control modes: power-down mode, idle mode, slow-frequency mode, sub-frequency mode, RTC mode, watch mode, and monitor mode
 - All interrupts supporting to wake up the CPU from IDLE mode
 - 10 interrupt sources supporting to wake up the CPU in power-down mode
 - Slow-frequency mode and sub-frequency mode supporting low-speed MCU operation
 - RTC mode supporting real-time clock (RTC) to wake up the CPU in power-down mode
 - Watch mode supporting watchdog (WDT) to wake up the CPU in power-down mode
 - Monitor mode supporting BOD1 to wake up the CPU in power-down mode
- Operating frequency range: up to 25 MHz
 - External crystal oscillator mode, 0–12 MHz at 2.0–3.6 V and 0–25 MHz at 2.4–3.6 V
 - CPU operating frequency can reach 12 MHz at 1.8–3.6 V and 25 MHz at 2.2–3.6 V
 - When on-chip clock frequency multiplier (CKM) is at 2.7–3.6 V, the CPU operating frequency can reach 36 MHz.
- Double data pointer
- Interrupt control
 - 16 interrupt sources, 4 priority levels
 - 3 external interrupts nINT0/1/2, with filtering
 - All external interrupts supporting high/low or rising/falling edge triggering
- 8-channel 12-bit single-ended ADC with a sampling rate greater than 500 kpsps

- 1 master/slave SPI serial interface, the rate reaching 12 MHz
- 2 master/slave two-wire serial interfaces: TWI0/I2C0 and STWI (SI2C)
- 1-channel DMA engine
 - P2P, M2P, P2M
 - Memory target: XRAM
 - Peripheral targets: UART0, UART1, SPI, TWI0/I2C0, ADC12 and CRC16
 - Timer 5 and Timer 6 are applied by DMA; they are independent timers when DMA is not enabled.
- Totally 9/11 timers/counters on-chip
 - RTC timer and WDT timer
 - Timer 0, 1, 2, 3
 - PCA0, programmable counter array 0
 - S0BRG and S1BRG
 - When timer 2/3 is used in separated mode, there are a total of 11 timers
- 8 keyboard interrupts
- 1 enhanced UART0 and 1 normal UART1

RF Features

- Operating frequency: 127-1020 MHz
- Modulation and demodulation methods: (G)FSK, (G)MSK, OOK
- Data rate: 0.5-300 kbps
- Sensitivity: -121 dBm @ 434 MHz, FSK
- Receive current: 8.5 mA @ 434 MHz, FSK
- Transmitting current: 72 mA @ 20 dBm, 434 MHz
- Configurable FIFO up to 64-Byte

System Features

- Operating voltage: 1.8 – 3.6 V
- Operating temperature: -40 – 85 °C
- QFN40 5x5 packaging

Application

- Automatic meter reading
- Home security and building automation
- Wireless sensor networks and industrial monitoring
- ISM band data communication

Description

Employed an 8-bit CPU core and an ultra-low power RF transceiver, the CMT2380F17 is a (G)FSK, (G)MSK, and OOK wireless MCU with high performance and ultra-low power applying to 127 to 1020 MHz band wireless applications. Operating with 1.8 to 3.6 V supply voltage, the CMT2380F17 consumes only 72 mA transmission current and 8.5 mA (exclusive of MCU consuming) receiving current while delivering up to 20 dBm power and reaching -121 dBm sensitivity.

The device employs a wide range of peripherals like support of standard UART, I2C and SPI interfaces, up to 25 general-purpose I/Os, support of internal high-speed, low-speed, low-power RC oscillators and 32.768 kHz external crystal oscillators, flexible data handling and packet handler, up to 64-byte Tx/Rx FIFO, feature-enriched RF GPIO, multiple low-power modes and fast-start mechanisms, high-precision RSSI, manual fast frequency hopping, multi-channel input 12-bit high-speed ADC, etc. Leading the industry in the aspect of the smallest package size, the CMT2380F17 is ideal for IoT applications with critical requests in size constraints and power-efficiency.

Table 1. The CMT2380F17 Resource List

Memory		Analog Peripherals		Digital Peripherals								
ROM	RAM	ADC	BOD	RTC	WDT	Timer	PCA	UART	SPI	TWI	KBI	GPIO
16K	1K	12bits x 8-ch 800ksps	√	1	1	16bits x 4	6-ch Capture/ Compare/PWM 2-ch Compare /PWM	2	1	2	8-ch	17

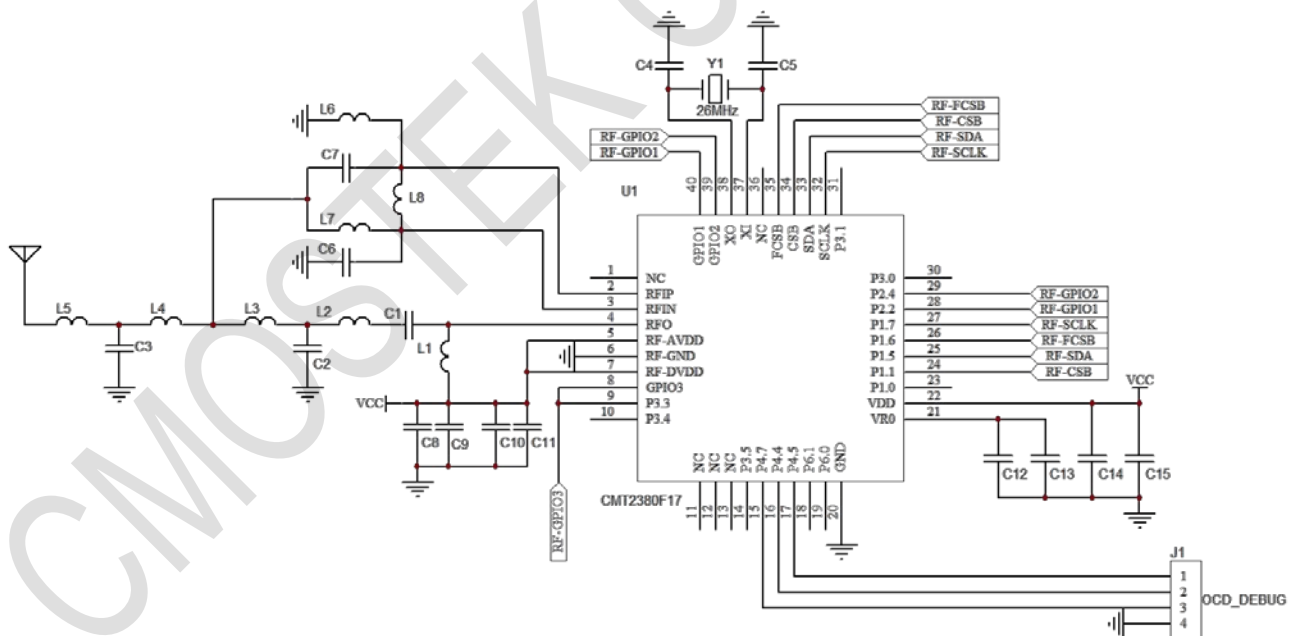


Figure 1. Typical Application Schematic for the CMT2380F17 (20dBm Output Power)

Table 2. Typical Application BOM (20dBm Output Power)

Label	Description	Component Value			Unit	Supplier
		434 MHz	868MHz	915MHz		
C1	±5%, 0402 NP0, 50 V	15	18	18	pF	-
C2	±5%, 0402 NP0, 50 V	3	3.6	3.6	pF	-
C3	±5%, 0402 NP0, 50 V	6.2	3.3	3.3	pF	-
C4	±5%, 0402 NP0, 50 V	24	24	24	pF	-
C5	±5%, 0402 NP0, 50 V	24	24	24	pF	-
C6	±5%, 0402 NP0, 50 V	4.7	2	1.8	pF	-
C7	±5%, 0402 NP0, 50 V	4.7	2	1.8	pF	-
C8	±20%, 0603 X7R, 25 V	4.7			uF	-
C9	±5%, 0402 NP0, 50 V	470			pF	-
C10	±20%, 0402 X7R, 25 V	0.1			uF	-
C11	±20%, 0402 X7R, 25 V	0.1			uF	-
C12	±20%, 0402 X7R, 25 V	0.1			uF	-
C13	±20%, 0603 X7R, 25 V	4.7			uF	-
C14	±20%, 0603 X7R, 25 V	1			uF	-
C15	±20%, 0603 X7R, 25 V	0.1			uF	-
L1	±10%, 0603 multilayer chip inductor	180	100	100	nH	Sunlord
L2	±10%, 0603 multilayer chip inductor	22	12	12	nH	Sunlord
L3	±10%, 0603 multilayer chip inductor	15pF	15	15	nH	Sunlord
L4	±10%, 0603 multilayer chip inductor	33	6.2	6.2	nH	Sunlord
L5	±10%, 0603 multilayer chip inductor	33	6.2	6.2	nH	Sunlord
L6	±10%, 0603 multilayer chip inductor	27	15	15	nH	Sunlord
L7	±10%, 0603 multilayer chip inductor	27	15	15	nH	Sunlord
L8	±10%, 0603 multilayer chip inductor	68	12	12	nH	Sunlord
Y1	±10 ppm, SMD32*25 mm	26			MHz	EPSON
U1	CMT2380F17, ultra-low power sub-1GHz wireless MCU	-			-	CMOSTEK

Table of Content

MCU Features	1
RF Features	1
System Features	1
Application	1
Description	2
1 Electrical Specifications	11
1.1 Recommended Operating Conditions	11
1.2 Absolute Maximum Ratings	11
1.3 RF Power Consumption	12
1.4 Receiver	12
1.5 Transmitter	14
1.6 RF Operating Mode Switching Time	15
1.7 RF Frequency Synthesizer	15
1.8 Requirement on Crystals for RF Section	17
1.9 Controller DC Specification	17
1.10 Controller Power Consumption Characteristics	18
1.11 BOD Characteristics	18
1.12 Controller IHRCO Characteristics	18
1.13 Controller ILRCO Characteristics	19
1.14 Controller CKM Characteristics	19
1.15 Controller Flash Characteristics	19
1.16 Controller ADC Characteristics	19
1.17 IVR Characteristics	20
1.18 Controller Serial Port Timing Characteristics	21
1.19 Controller SPI Timing Characteristics	21
1.20 Receive Current and Supply Voltage Correlation	23
1.21 Correlation Among Receive Current, Supply Voltage and Temperature	24
1.22 Receive Sensitivity and Supply Voltage Correlation	25
1.23 Receive Sensitivity and Temperature Correlation	25
1.24 Transmit Power and Supply Voltage Correlation	26
1.25 Phase Noise	27
2 Pin Description	28
3 Chip Structure	32
4 Sub-GHz Transceiver	35
4.1 Transmitter	35
4.2 Receiver	35
4.3 Transceiver Power-on Reset (POR)	35
4.4 Transceiver Crystal Oscillator	36
4.5 Transceiver Built-in Low Frequency Oscillator (LPOSC)	36
4.6 Transceiver Built-in Low Battery Detection	37
4.7 Receiver Signal Strength Indication (RSSI)	37
4.8 Phase Jump Detector (PJD)	37
4.9 Receiver Clock Data Recovery (CDR)	38
4.10 Fast Manual Frequency Hopping	39
4.11 Transceiver Control Interface and Operating Mode	39
4.11.1 Transceiver SPI Interface Timing	39
4.11.2 Transceiver FIFO Interface Timing	40
4.11.3 Transceiver Operating Status, Timing, and Power Consumption	41
4.11.4 Transceiver GPIO Function and Interrupt Mapping	44
5 80C51 CPU Function Description	47
5.1 CPU Register	47
5.2 CPU Timing	48
5.3 CPU Addressing Mode	49
5.3.1 Direct Addressing (DIR)	49

5.3.2	Indirect Addressing (IND)	49
5.3.3	Register Instruction (REG)	49
5.3.4	Register-Specific Instruction	49
5.3.5	Immediate Constant (IMM)	49
5.3.6	Index Addressing	49
6	Memory Organization	50
6.1	On-Chip Program Flash	50
6.2	On-Chip Data RAM	51
6.3	On-chip Expanded RAM (XRAM)	52
6.4	Off-Chip External Data Memory access	52
6.5	Declaration Identifiers in a C51-Compiler	52
7	XRAM Access	54
7.1	MOVX on 16-bit Address with dual DPTR	54
7.2	MOVX on 8-bit Address with XRPS	55
8	Direct Memory Access Controller (DMA)	57
8.1	DMA Structure	57
8.2	DMA Operation	58
8.2.1	DMA Transfer Types	58
8.2.2	DMA Transfer Mode	59
8.2.3	Transfer Count & Address Pointer	59
8.2.4	Start a DMA Transfer	60
8.2.5	Suspend or Stop DMA Transfer	60
8.2.6	DMA Interrupt	60
8.2.7	DMA Loop Mode	61
8.2.8	Error Handling in DMA	61
8.2.9	Data Copied to CRC16	61
8.2.10	Timer 5 & Timer 6	61
8.3	DMA Register	62
8.4	Timer5 Register	64
8.5	Timer 6 Register	66
9	System Clock	68
9.1	Clock Structure	68
9.2	Clock Source Switching	69
9.3	On-chip CKM (PLL)	69
9.4	Wake-up clock from CKM	69
9.5	Clock Register	70
10	Watch Dog Timer (WDT)	74
10.1	WDT Structure	74
10.2	WDT During Idle	74
10.3	WDT Register	75
10.4	WDT Hardware Option	77
11	Real-Time-Clock (RTC)/System-Timer	78
12	System Reset	82
12.1	Reset Source	82
12.2	Power-On Reset (POR)	82
	Note: POF0 must be cleared by software	83
12.3	External Reset	83
12.4	Software Reset	84
12.5	Brown-Out Reset	84
12.6	WDT Reset	85
12.7	Illegal Address Reset	85
13	Power Management	86

13.1 Brown-Out Detector	86
13.2 Power Saving Mode	86
13.2.1 Slow Mode	86
13.2.2 Sub-Clock Mode.....	87
13.2.3 RTC Mode.....	87
13.2.4 Watch Mode	87
13.2.5 Monitor Mode	87
13.2.6 Idle Mode	87
13.2.7 Power-down Mode	88
13.2.8 Interrupt Recovery from Power-down.....	89
13.2.9 Reset Recovery from Power-down	89
13.2.10 KBI wakeup Recovery from Power-down.....	89
13.3 Power Control Register	89
14 Configurable I/O Ports	93
14.1 IO Structure.....	93
14.1.1 Port 3 Quasi-Bidirectional IO Structure	93
14.1.2 Port 3 Push-Pull Output Structure.....	94
14.1.3 Port 3 Input-Only (High Impedance Input) Structure	94
14.1.4 Port 3 Open-Drain Output Structure.....	94
14.1.5 General Analog Input Only Structure.....	95
14.1.6 General Open-Drain Output with Pull-up Resistor Structure	95
14.1.7 General Open-Drain Output Structure.....	95
14.1.8 General Port Digital Input Configured	96
14.1.9 General Push-Pull Output Structure.....	96
14.1.10 Port Pin Output Driving Strength Selection	96
14.1.11 Port Pin Output Fast Driving Selection.....	96
14.2 I/O Port Register	96
14.2.1 Port 1 Register	97
14.2.2 Port 2 Register	98
14.2.3 Port 3 Register	99
14.2.4 Port 4 Register	100
14.2.5 Port 6 Register	101
14.2.6 Port Output Driving Strength Control Register	101
14.2.7 Port Output Fast Driving Control Register.....	103
14.3 Port Function Redirection.....	104
15 Interrupt.....	112
15.1 Interrupt Structure	112
15.2 Interrupt Source	114
15.3 Interrupt Enable.....	117
15.4 Interrupt Priority.....	117
15.5 Interrupt Process.....	118
15.6 nINTx Input Source Selection and input filter (x=0~2).....	119
15.7 Interrupt Register	120
16 Timers/Counters	130
16.1 Timer 0 and Timer 1	130
16.1.1 Timer 0/1 Mode 0	130
16.1.2 Timer 0/1 Mode 1	132
16.1.3 Timer 0/1 Mode 2	133
16.1.4 Timer 0/1 Mode 3	134
16.1.5 Timer 0/1 Programmable Clock-Out.....	134
16.1.6 Timer 0/1 Register.....	136
16.2 Timer 2.....	141

16.2.1	Timer 2 Mode 0 (Auto-Reload and External Interrupt)	141
16.2.2	Timer 2 Mode 1 (Auto-Reload with External Interrupt)	142
16.2.3	Timer 2 Mode 2 (Capture)	143
16.2.4	Timer 2 Mode 3 (Capture with Auto-Zero)	144
16.2.5	Split Timer 2 Mode 0 (AR and Ex. INT)	145
16.2.6	Split Timer 2 Mode 1 (AR with Ex. INT)	146
16.2.7	Split Timer 2 Mode 2 (Capture)	147
16.2.8	Split Timer 2 Mode 3 (Capture with Auto-Zero)	148
16.2.9	Split Timer 2 Mode 4 (8-bit PWM Mode)	149
16.2.10	Baud-Rate Generator Mode (BRG)	149
16.2.11	Timer 2 Programmable Clock Output	151
16.2.12	Timer 2 Register	153
16.3	Timer 3	157
16.3.1	Timer 3 Mode 0 (Auto-Reload and External Interrupt)	157
16.3.2	Timer 3 Mode 1 (Auto-Reload with External Interrupt)	157
16.3.3	Timer 3 Mode 2 (Capture)	158
16.3.4	Timer 3 Mode 3 (Capture and Auto-Zero)	159
16.3.5	Split Timer 3 Mode 0 (Auto-Reload and External Interrupt)	159
16.3.6	Split Timer 3 Mode 1 (Auto-Reload with External Interrupt)	160
16.3.7	Split Timer 3 Mode 2 (Capture)	162
16.3.8	Split Timer 3 Mode 3 (Capture with Auto-Zero)	163
16.3.9	Split Timer 3 Mode 4 (8-bit PWM Mode)	164
16.3.10	Timer 3 Programmable Clock Output	164
16.3.11	Timer 3 Register	166
16.4	Timer Global Control	170
16.4.1	Global Enable for all Timer Run	170
16.4.2	Global Control for all Timer Reload	171
16.4.3	Global Control for all Timer Stop	172
17	Programmable Counter Array (PCA0)	173
17.1	PCA Overview	173
17.2	PCA Timer/Counter	173
17.3	Compare/Capture Modules	179
17.4	Operation Modes of the PCA	182
17.4.1	Capture Mode	183
17.4.2	Buffered Capture Mode	183
17.4.3	16-bit Software Timer Mode (Compare mode)	185
17.4.4	High Speed Output Mode (Compare Output mode)	185
17.4.5	Buffered 8-bit PWM Mode	185
17.4.6	Un-buffered 10/12/16-bit PWM Mode	186
17.4.7	Buffered 10/12/16-bit PWM Mode	187
17.4.8	COPM Mode	188
17.4.9	Buffered COPM Mode	189
17.4.10	FIFO Data Mode	190
17.4.11	Enhanced PWM Control	190
17.4.12	PCA Module Output Control	194
17.4.13	Variable Resolution on Central Aligned PWM	198
18	Serial Port 0 (UART0)	200
18.1	Serial Port 0 Mode 0	201
18.2	Serial Port 0 Mode 1	202
18.3	Serial Port 0 Mode 2 and Mode 3	203
18.4	Frame Error Detection	204
18.5	Multiprocessor Communications	204

18.6 Automatic Address Recognition	204
18.7 Baud Rate Setting	206
18.7.1 Baud Rate Selection in S0	206
18.7.2 Baud Rate in Mode 0	207
18.7.3 Baud Rate in Mode 2	207
18.7.4 Baud Rate in Mode 1 & 3	208
18.7.4.1 Using Timer 1 as the Baud Rate Generator	208
18.7.4.2 Using Timer 2 as the Baud Rate Generator	215
18.7.4.3 Using S0 Baud Rate Timer as the Baud Rate Generator (S0BRG)	221
18.7.4.4 Using S1 Baud Rate Timer as the Baud Rate Generator	222
18.8 Serial Port 0 Mode 4 (SPI Master)	222
18.9 Serial Port 0 Register	224
18.10 Serial Port 0 Enhance function	227
18.10.1 S0 Baud Rate Generator (S0BRG)	229
18.10.2 Independent Baud Rate Generator S0BRG for S0	229
18.10.3 S0 Enhanced Mode	230
18.10.4 S0 LIN Bus Register	230
18.10.5 S0 acts as 8-bit Timer Mode	231
18.10.6 S0 acts as 16-bit Timer Mode	231
18.10.7 S0BRG Programmable Clock Output	232
19 Serial Port 1 (UART1)	233
19.1 Serial Port 1 Baud Rate Generator (S1BRG)	233
19.2 S1BRG configuration (S1TME=0)	233
19.2.1 Baud Rate in Mode 0	233
19.2.2 Baud Rate in Mode 2	234
19.2.3 Baud Rate in Mode 1 & 3	234
19.3 Serial Port 1 Mode 4 (SPI Master)	237
19.4 8-Bit Timer Mode on S1BRG	239
19.5 16-Bit Timer Mode on S1BRG	241
19.6 S1BRT Programmable Clock Output	241
How to Program 8-bit S1BRG in Clock-out Mode	242
19.7 S1 Baud Rate Generator for S0	243
19.8 Serial Port 1 Register	243
20 Serial Peripheral Interface (SPI)	248
20.1 Typical SPI Configurations	249
20.1.1 Single Master & Single Slave	249
20.1.2 Dual Device, where either can be a Master or a Slave	249
20.1.3 Single Master & Multiple Slaves	249
20.2 Configuring the SPI	250
20.2.1 Additional Considerations for a Slave	251
20.2.2 Additional Considerations for a Master	251
20.2.3 Mode Change on nSS-pin	251
20.2.4 Transmit Holding Register Full Flag	251
20.2.5 Write Collision	251
20.2.6 SPI Clock Rate Select	252
20.3 Data Mode	253
20.4 Daisy-Chain Connection	255
20.4.1 Configuring the Daisy-Chain	255
20.5 SPI Register	255
21 Two Wire serial Interface (TWI0/ I2C0)	259
21.1 Operating Modes	259
21.1.1 Master Transmitter Mode	260
21.1.2 Master Receiver Mode	261

21.1.3	Slave Transmitter Mode	261
21.1.4	Slave Receiver Mode	262
21.2	Miscellaneous States	262
21.3	Using the TWI/ I2C	263
The figure below shows how to read the flow charts.		263
21.4	TWI0/ I2C0 Register	269
22	Serial Interface Detection (STWI/SI2C)	273
22.1	SID Structure	273
22.2	SID Register	273
23	Beeper	275
24	Keypad Interrupt (KBI)	277
24.1	KBI Structure	277
24.2	KBI Register	277
25	General Purpose Logic (GPL-CRC)	280
25.1	GPL-CRC Structure	280
25.2	GPL-BOREV Structure	281
25.3	GPL Register	281
26	12-Bit ADC	283
26.1	ADC Structure	283
26.2	ADC Operation	284
26.2.1	ADC Input Channels	284
26.2.2	ADC Internal Voltage Reference	284
26.2.3	Starting a Conversion	284
26.2.4	ADC Conversion Rate	285
26.2.5	ADC Interrupts	285
26.2.6	ADC Window Detect	286
26.2.7	ADC Channel Scan Mode	287
26.2.8	Transfer ADC Data by DMA	287
26.2.9	I/O Pins Used with ADC Function	288
26.2.10	Idle and Power-Down Mode	288
26.2.11	How to improve ADC Accuracy	288
26.3	ADC Register	289
27	Internal Voltage Reference (IVR, 1.4V)	297
27.1	IVR (1.4V) Structure	297
27.2	IVR Register	297
27.3	How to read IVR (1.4V) ADC Prestored value	297
28	ISP and IAP	299
28.1	CMT2380F17 Flash Memory Configuration	299
28.2	CMT2380F17 Flash Access in ISP/IAP	299
28.2.1	ISP/IAP Flash Page Erase Mode	300
28.2.2	ISP/IAP Flash Byte Program Mode	302
28.2.3	ISP/IAP Flash Read Mode	304
28.3	ISP Operation	305
28.3.1	Hardware approached ISP	305
28.3.2	Software approached ISP	306
28.3.3	Notes for ISP	306
28.4	In-Application-Programming (IAP)	307
28.4.1	IAP-memory Boundary/Range	307
28.4.2	Update data in IAP-memory	307
28.4.3	Notes for IAP	308
28.5	ISP/IAP Register	308

28.6 ISP/IAP Sample Code	311
29 Page P SFR Access	313
30 Auxiliary SFRs	318
30.1 SFR Figure (Page 0~F)	318
30.2 SFR Bit Assignment (Page 0~F)	319
30.3 Auxiliary SFR Map (Page P)	323
30.4 Auxiliary SFR Bit Assignment (Page P)	323
30.5 Auxiliary SFR Register	325
31 Hardware Option.....	334
32 Application Notes	336
32.1 Power Supply Circuit.....	336
32.2 Reset Circuit.....	336
32.3 ICP and OCD Interface Circuit	336
32.4 In-Chip-Programming Function	337
32.5 On-Chip-Debug Function	338
33 Instruction Set	340
34 Ordering Information.....	343
35 Packaging Information	344
36 Top Marking	345
37 Reference Documents.....	346
38 Revise History.....	347
39 Contacts	348

1 Electrical Specifications

$V_{DD} = 3.3\text{ V}$, $T_{OP} = 25\text{ }^{\circ}\text{C}$, $F_{RF} = 433.92\text{ MHz}$, sensitivity is measured by receiving a PN9 sequence and matching to $50\text{ }\Omega$ impedance, 0.1% BER if nothing else stated. All measurement results are obtained using the evaluation board CMT2380F17-EM if nothing else stated.

1.1 Recommended Operating Conditions

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Operating supply voltage	V_{DD-RF}		1.8		3.6	V
	$V_{OP3-MCU}$	CPU operating speed 0-36MHz	2.7		3.6	V
	$V_{OP4-MCU}$	CPU operating speed 0-24MHz	2.2		3.6	V
	$V_{OP5-MCU}$	CPU operating speed 0-12MHz	1.8		3.6	V
Operating temperature	T_{OP}		-40		85	$^{\circ}\text{C}$
RF supply voltage slope	V_{RF-PSR}		1			mV/us
Controller supply voltage Slope	$V_{MCU-PSR}$		50			mV/us
Controller power-on reset effective voltage	$V_{MCU-POR}$				0.1	V

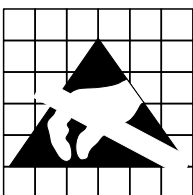
1.2 Absolute Maximum Ratings

Parameter	Symbol	Condition	Min.	Typ.	Max.
Supply voltage	V_{DD}		-0.3	3.6	V
Interface voltage	V_{IN}		-0.3	3.6	V
Junction temperature	T_J		-40	125	$^{\circ}\text{C}$
Storage temperature	T_{STG}		-50	150	$^{\circ}\text{C}$
Soldering temperature	T_{SDR}	Lasts for at least 30 seconds		255	$^{\circ}\text{C}$
ESD rating ^[2]		Human body model (HBM)	-2	2	kV
Latch-up current		@ 85 $^{\circ}\text{C}$	-100	100	mA
Maximum current from MCU-VDD to ground				200	mA
Maximum sink current of any pin of MCU				40	mA

Notes:

[1]. Exceeding *the Absolute Maximum Ratings* may cause permanent damage to the equipment. This value is a pressure rating and does not imply that the function of the equipment is affected under this pressure condition, but if it is exposed to absolute maximum ratings for extended periods of time, it may affect equipment reliability.

[2]. The CMT2380F17 is a high performance RF integrated circuit. The operation and assembly of this chip should only be performed on a workbench with good ESD protection.



Caution! ESD sensitive device. Precaution should be used when handling the device in order to prevent performance degradation or loss of functionality.

1.3 RF Power Consumption

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Sleep current	I_{SLEEP}	Sleep mode,sleep counter off		300		nA
		Sleep mode,sleep counter on		800		nA
Standbycurrent	$I_{Standby}$	Crystal oscillator on		1.45		mA
RFS current	I_{RFS}	433 MHz		5.7		mA
		868 MHz		5.8		mA
		915 MHz		5.8		mA
TFS current	I_{TFS}	433 MHz		5.6		mA
		868 MHz		5.9		mA
		915 MHz		5.9		mA
RX current (high performance)	I_{RX-HP}	FSK, 433 MHz, 10 kbps, 10 kHz F_{DEV}		8.5		mA
		FSK, 868 MHz, 10 kbps, 10 kHz F_{DEV}		8.6		mA
		FSK, 915 MHz, 10 kbps, 10 kHz F_{DEV}		8.9		mA
RX current (low power mode)	I_{RX-LP}	FSK, 433 MHz, 10 kbps, 10 kHz F_{DEV}		7.2		mA
		FSK, 868 MHz, 10 kbps, 10 kHz F_{DEV}		7.3		mA
		FSK, 915 MHz, 10 kbps, 10 kHz F_{DEV}		7.6		mA
TX current	I_{TX}	FSK, 433 MHz, +20 dBm (Direct-tie)		72		mA
		FSK, 433 MHz, +20 dBm (With RF switch)		77		mA
		FSK, 433 MHz, +13 dBm (Direct-tie)		23		mA
		FSK, 433 MHz, +10 dBm (Direct-tie)		18		mA
		FSK, 433 MHz, -10 dBm (Direct-tie)		8		mA
		FSK, 868 MHz, +20dBm (Direct-tie)		87		mA
		FSK, 868 MHz, +20dBm (With RF switch)		80		mA
		FSK, 868 MHz, +13 dBm (Direct-tie)		27		mA
		FSK, 868 MHz, +10 dBm (Direct-tie)		19		mA
		FSK, 868 MHz, -10 dBm (Direct-tie)		8		mA
		FSK, 915 MHz, +20 dBm (Direct-tie)		70		mA
		FSK, 915 MHz, +20 dBm (With RF switch)		75		mA
		FSK, 915 MHz, +13 dBm(Direct-tie)		28		mA
		FSK, 915 MHz, +10 dBm (Direct-tie)		19		mA
		FSK, 915 MHz, -10 dBm (Direct-tie)		8		mA

1.4 Receiver

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Data Rate	DR	OOK	0.5		40	kbps
		FSK 和 GFSK	0.5		300	kbps
Error	F_{DEV}	FSK 和 GFSK	2		200	kHz
Sensitivity @ 433 MHz	S_{433-HP}	DR = 2.0 kbps, F_{DEV} = 10 kHz		-121		dBm
		DR = 10 kbps, F_{DEV} = 10 kHz		-116		dBm

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
		DR = 10 kbps, $F_{DEV} = 10$ kHz (low-power configuration)		-115		dBm
		DR = 20 kbps, $F_{DEV} = 20$ kHz		-113		dBm
		DR = 20 kbps, $F_{DEV} = 20$ kHz (low-power configuration)		-112		dBm
		DR = 50 kbps, $F_{DEV} = 25$ kHz		-111		dBm
		DR = 100 kbps, $F_{DEV} = 50$ kHz		-108		dBm
		DR = 200 kbps, $F_{DEV} = 100$ kHz		-105		dBm
		DR = 300 kbps, $F_{DEV} = 100$ kHz		-103		dBm
Sensitivity @ 868 MHz	S _{868-HP}	DR = 2.0 kbps, $F_{DEV} = 10$ kHz		-119		dBm
		DR = 10 kbps, $F_{DEV} = 10$ kHz		-113		dBm
		DR = 10 kbps, $F_{DEV} = 10$ kHz (low-power configuration)		-111		dBm
		DR = 20 kbps, $F_{DEV} = 20$ kHz		-111		dBm
		DR = 20 kbps, $F_{DEV} = 20$ kHz (low-power configuration)		-109		dBm
		DR = 50 kbps, $F_{DEV} = 25$ kHz		-108		dBm
		DR = 100 kbps, $F_{DEV} = 50$ kHz		-105		dBm
		DR = 200 kbps, $F_{DEV} = 100$ kHz		-102		dBm
		DR = 300 kbps, $F_{DEV} = 100$ kHz		-99		dBm
Sensitivity @ 915 MHz	S _{915-HP}	DR = 2.0 kbps, $F_{DEV} = 10$ kHz		-117		dBm
		DR = 10 kbps, $F_{DEV} = 10$ kHz		-113		dBm
		DR = 10 kbps, $F_{DEV} = 10$ kHz (low-power configuration)		-111		dBm
		DR = 20 kbps, $F_{DEV} = 20$ kHz		-111		dBm
		DR = 20 kbps, $F_{DEV} = 20$ kHz (low-power configuration)		-109		dBm
		DR = 50 kbps, $F_{DEV} = 25$ kHz		-109		dBm
		DR = 100 kbps, $F_{DEV} = 50$ kHz		-105		dBm
		DR = 200 kbps, $F_{DEV} = 100$ kHz		-102		dBm
		DR = 300 kbps, $F_{DEV} = 100$ kHz		-99		dBm
Saturation	P _{LVL}				20	dBm
Image rejection ratio	IMR	F _{RF} =433 MHz		35		dBc
		F _{RF} =868 MHz		33		dBc
		F _{RF} =915 MHz		33		dBc
Receive channel bandwidth	BW	Receive channel bandwidth	50		500	kHz
Co-channel rejection	CCR	DR = 10 kbps, $F_{DEV} = 10$ kHz Interference with the same modulation		-7		dBc
Adjacent channel rejection	ACR-I	DR = 10 kbps, $F_{DEV} = 10$ kHz, BW=100 kHz, 200 kHz channel spacing, interference with the same modulation		30		dBc
Alternate channel rejection	ACR-II	DR = 10 kbps, $F_{DEV} = 10$ kHz, BW=100kHz, 400 kHz channel spacing, interference with the same modulation		45		dBc
Blocking	BI	DR = 10 kbps, $F_{DEV} = 10$ kHz, ± 1 MHz offset, continuous wave interference		70		dBc
		DR = 10 kbps, $F_{DEV} = 10$ kHz, ± 2 MHz offset,		72		dBc

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
		continuous wave interference				
		DR = 10 kbps, $F_{DEV} = 10$ kHz, ± 10 MHz offset, continuous wave interference		75		dBc
Input 3 rd order intercept point	IIP3	DR = 10 kbps, $F_{DEV} = 10$ kHz, 1 MHz and 20 MHz offset two tone test, maximum system gain setting		-25		dBm
RSSI range	RSSI		-120		20	dBm
More sensitivity (typical configuration)		433.92 MHz, DR = 1.2kbps, $F_{DEV} = 5$ kHz		-122.9		dBm
		433.92 MHz, DR = 1.2kbps, $F_{DEV} = 10$ kHz		-121.8		dBm
		433.92 MHz, DR = 1.2kbps, $F_{DEV} = 20$ kHz		-119.5		dBm
		433.92 MHz, DR = 2.4kbps, $F_{DEV} = 5$ kHz		-120.6		dBm
		433.92 MHz, DR = 2.4kbps, $F_{DEV} = 10$ kHz		-120.3		dBm
		433.92 MHz, DR = 2.4kbps, $F_{DEV} = 20$ kHz		-119.7		dBm
		433.92 MHz, DR = 9.6 kbps, $F_{DEV} = 9.6$ kHz		-116.0		dBm
		433.92 MHz, DR = 9.6 kbps, $F_{DEV} = 19.2$ kHz		-116.1		dBm
		433.92 MHz, DR = 20 kbps, $F_{DEV} = 10$ kHz		-114.2		dBm
		433.92 MHz, DR = 20 kbps, $F_{DEV} = 20$ kHz		-113.0		dBm
		433.92 MHz, DR = 50 kbps, $F_{DEV} = 25$ kHz		-110.6		dBm
		433.92 MHz, DR = 50 kbps, $F_{DEV} = 50$ kHz		-109.0		dBm
		433.92 MHz, DR = 100 kbps, $F_{DEV} = 50$ kHz		-107.8		dBm
		433.92 MHz, DR = 200 kbps, $F_{DEV} = 50$ kHz		-103.5		dBm
		433.92 MHz, DR = 200 kbps, $F_{DEV} = 100$ kHz		-104.3		dBm
		433.92 MHz, DR = 300 kbps, $F_{DEV} = 50$ kHz		-98.0		dBm
		433.92 MHz, DR = 300 kbps, $F_{DEV} = 150$ kHz		-101.6		dBm

1.5 Transmitter

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Output power	P_{OUT}	Specific matching network is required for different frequency bands	-20		+20	dBm
Output power step	P_{STEP}			1		dB
GFSK (Gaussian filter coefficient)	BT		0.3	0.5	1.0	-
Output power change with different temperature	$P_{OUT-TOP}$	Temperature range: -40 to +85 °C		1		dB
Spurious emissions		$P_{OUT} = +13$ dBm, 433 MHz, $F_{RF} < 1$ GHz			-54	dBm
		1 GHz to 12.75 GHz, including harmonics			-36	dBm
Harmonic output ^[1] for $F_{RF} = 433$ MHz	H2 ₄₃₃	2 nd harmonic, +20 dBm P_{OUT}		-46		dBm
	H3 ₄₃₃	3 rd harmonic, +20 dBm P_{OUT}		-50		dBm
Harmonic output ^[1] for $F_{RF} = 868$ MHz	H2 ₈₆₈	2 nd harmonic, +20 dBm P_{OUT}		-43		dBm
	H3 ₈₆₈	3 rd harmonic, +20 dBm P_{OUT}		-52		dBm
Harmonic output ^[1] for $F_{RF} = 915$ MHz	H2 ₉₁₅	2 nd harmonic, +20 dBm P_{OUT}		-48		dBm
	H3 ₉₁₅	3 rd harmonic, +20 dBm P_{OUT}		-53		dBm
Harmonic output ^[1] for $F_{RF} = 433$ MHz	H2 ₄₃₃	2 nd harmonic, +13 dBm P_{OUT}		-52		dBm
	H3 ₄₃₃	3 rd harmonic, +13 dBm P_{OUT}		-52		dBm

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Harmonic output ^[1] for $F_{RF}= 868$ MHz	H2 ₈₆₈	2 nd harmonic,+13 dBm P _{OUT}		-52		dBm
	H3 ₈₆₈	3 rd harmonic,+13 dBm P _{OUT}		-52		dBm
Harmonic output ^[1] for $F_{RF}= 915$ MHz	H2 ₉₁₅	2 nd harmonic,+13 dBm P _{OUT}		-52		dBm
	H3 ₉₁₅	3 rd harmonic,+13 dBm P _{OUT}		-52		dBm
Notes:						
[1] The harmonic level mainly depends on the matching network. Above parameters are measured based on the CMT2380F17-EM, users may get different results on their PCB designs.						

1.6 RF Operating Mode Switching Time

Parameter	Symbol	Condition	Min.	Typ.	Max.	Units
Stabilizing time	T _{SLP-RX}	From sleep to RX		1000		us
	T _{SLP-TX}	From sleep to TX		1000		us
	T _{STB-RX}	From standby to RX		350		us
	T _{STB-TX}	From standby to TX		350		us
	T _{RFS-RX}	From RFS to RX		20		us
	T _{TFS-RX}	From TFS to TX		20		us
	T _{TX-RX}	From TX to RX (Needs 2T _{symbol} to ramp down)		2T _{symbol} +350		us
	T _{RX-TX}	From RX to TX		350		us
Notes:						
[1] T _{SLP-RX} and T _{SLP-TX} are dominated by the crystal oscillator startup time, and the start-up time is to a large degree crystal dependent.						

1.7 RF Frequency Synthesizer

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Frequency range	F _{RF}	Need different matching networks.	760		1020	MHz
			380		510	MHz
			190		340	MHz
			127		170	MHz
Frequency resolution	F _{RES}			25		Hz
Frequency tuning time	t _{TUNE}			150		us
Phase noise @ 433 MHz	PN ₄₃₃	10 kHz deviation		-94		dBc/Hz
		100 kHz deviation		-99		dBc/Hz
		500 kHz deviation		-118		dBc/Hz
		1MHz deviation		-127		dBc/Hz
		10 MHz deviation		-134		dBc/Hz
Phase noise @ 868 MHz	PN ₈₆₈	10 kHz deviation		-92		dBc/Hz
		100 kHz deviation		-95		dBc/Hz
		500 kHz deviation		-114		dBc/Hz
		1MHz deviation		-121		dBc/Hz
		10 MHz deviation		-130		dBc/Hz
Phase noise @ 915 MHz	PN ₉₁₅	10 kHz deviation		-89		dBc/Hz
		100 kHz deviation		-92		dBc/Hz

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
		500 kHz deviation		-111		dBc/Hz
		1MHz deviation		-121		dBc/Hz
		10 MHz deviation		-130		dBc/Hz

1.8 Requirement on Crystals for RF Section

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Crystal frequency ^[1]	F _{XTAL}			26		MHz
Crystal frequency tolerance ^[2]	ppm			20		ppm
Load capacitance	C _{LOAD}			15		pF
ESR	R _m			60		Ω
Crystal startup time ^[3]	t _{XTAL}			400		us

Notes:

- [1]. An external reference clock can be used to drive the XI pin indirectly through a coupling capacitor if such a clock is available. The peak-to-peak level of the external reference clock is required between 0.3 and 0.7 V.
- [2]. It involves: (1) initial tolerance, (2) crystal loading, (3) aging, and (4) temperature changing. The acceptable crystal frequency tolerance is subject to the bandwidth of the receiver and the RF tolerance between the receiver and its paired transmitter.
- [3]. This parameter is to a large degree crystal dependent.

1.9 Controller DC Specification

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Input high level (all IO ports)	V _{IH1}	Except P6.0 and P6.1.	0.6			V _{DD}
Input high level (RST, P6.0, P6.1)	V _{IH2}		0.75			V _{DD}
Input low level (all IO ports)	V _{IL1}	Except P6.0 and P6.1.			0.15	V _{DD}
Input low level (RST, P6.0, P6.1)	V _{IL2}				0.2	V _{DD}
Input high leakage current (all IO ports)	I _{IH}	V _{PIN} = V _{DD}		0	±1	uA
Logic 0 input current (P3 is in quasi-bidirectional port mode or input port with on-chip pull-up resistor)	I _{IL1}	V _{PIN} = 0.4V		-20	-30	uA
Logic 0 input current (all input-only or open-drain output ports)	I _{IL2}	V _{PIN} = 0.4V		0	-1	uA
Logic 1 to 0 input conversion current (P3 is in quasi-bidirectional port mode)	I _{H2L}	V _{PIN} = 1.8V		-320	-450	uA
Output high current (P3 is in quasi-bidirectional port mode)	I _{OH1}	V _{PIN} = 2.4V, V _{DD} = 3.3V	-50	-80		uA
Output high current (all push-pull output ports)	I _{OH2}	V _{PIN} = 2.4V, V _{DD} = 3.3V	-8	-11		mA
Output low current (all I/O ports)	I _{OL1}	V _{PIN} = 0.4V, V _{DD} = 3.3V	14	17		mA
Output high current (low drive capability of all push-pull output ports, except RST pin)	I _{OL2}	V _{PIN} = 0.4V, V _{DD} = 3.3V	1.2	2.2		mA
Output low current (low drive capability of all I/O ports)	I _{OL2}	V _{PIN} = 0.4V, except P6.0, P6.1, P4.7	2			mA
Internal reset pull-down resistance	R _{RST}	V _{DD} = 3.3V		207		KΩ

1.10 Controller Power Consumption Characteristics

Parameter	Symbol	Condition	Note	Typ.	Max.	Unit
Normal mode operating current	I _{OP1}	SYSCLK = 32MHz @ IHRCO with PLL		10.5		mA
	I _{OP2}	SYSCLK = 24MHz @ IHRCO with PLL		9		mA
	I _{OP3}	SYSCLK = 12MHz @ IHRCO		5.3		mA
	I _{OP5}	SYSCLK = 12MHz @ IHRCO, V _{DD} =3.3V with ADC 400ksps		5.3		mA
Low speed mode operating current	I _{OPS1}	SYSCLK = 12MHz/128 @ IHRCO		0.6		mA
Idle mode operating current	I _{IDLE1}	SYSCLK = 12MHz @ IHRCO		1.1		mA
	I _{IDLE2}	SYSCLK = 12MHz/128 @ IHRCO		0.45		mA
	I _{IDLE3}	SYSCLK = 32KHz @ ILRCO		50		uA
Sub-frequency mode operating current	I _{SUB1}	SYSCLK = 32KHz @ ILRCO, BOD1 is disabled.		65		uA
	I _{SUB2}	SYSCLK = 32KHz/128 @ ILRCO, BOD1 is disabled.		60		uA
Watch mode operating current	I _{WAT}	WDT = 32KHz @ ILRCO in power-down mode.		5		uA
Monitor mode operating current	I _{MON}	BOD1 is enabled in power-down mode.		10		uA
RTC mode operating current	I _{RTC}	RTC operates in power-down mode, V _{DD} = 3.0V		4.5		uA
Power-down mode current	I _{PD}			2.5		uA

1.11 BOD Characteristics

Parameter	Symbol	Condition	Note	Typ.	Max.	Unit
BOD0 detection level	V _{BOD0}	TA = -40°C to +85°C		1.7		V
BOD1 detection level is 2.0V	V _{BOD10}	TA = -40°C to +85°C		2.0		V
BOD1 detection level is 2.4V	V _{BOD10}	TA = -40°C to +85°C		2.37		V

1.12 Controller IHRCO Characteristics

Parameter	Symbol	Condition	Note	Typ.	Max.	Unit
Supply power voltage			1.8		3.6	V
IHRCO frequency		TA = +25°C, AFS = 0		12		MHz
		TA = +25°C, AFS = 1		11.059		MHz
IHRCO frequency error (factory calibration)		TA = +25°C	-1.0		+1.0	%
		TA = -40°C ~ +85°C	-2.0 ^[1]		+2.0 ^[1]	%
IHRCO start time		TA = -40°C ~ +85°C			32 ^[1]	Us
IHRCO power consumption		TA = +25°C, V _{DD} = 3.3V		350 ^[1]		uA

Notes:

[1] The data is based on the characteristics, not from product test

1.13 Controller ILRCO Characteristics

Parameter	Symbol	Condition	Note	Typ.	Max.	Unit
Supply power voltage			1.8		3.6	V
ILRCO frequency		TA = +25°C		32		KHz
ILRCOfrequencydeviation		TA = +25°C	-8 ^[1]		+8 ^[1]	%
		TA = -40°C~+85°C	-20 ^[1]		+20 ^[1]	%
Notes:						
[1] The data is based on the characteristics, not from product test.						

1.14 Controller CKM Characteristics

Parameter	Symbol	Condition	Note	Typ.	Max.	Unit
Supply power voltage		TA = -40°C~+85°C	2.4		3.6	V
Clock input range		TA = -40°C~+85°C	4.5 ^[1]		6.5 ^[1]	MHz
CKM startup time		TA = -40°C~+85°C	30 ^[2]		100 ^[2]	us
CKM power consumption		TA = +25°C, V _{DD} = 3.6V, CKM=96MHz		350		uA
		TA = +25°C, V _{DD} = 3.6V, CKM=144MHz		450		uA
Notes:						
[1]. Data is guaranteed by design, not from test.						
[2]. The data is based on the characteristics, not from product test						

1.15 Controller Flash Characteristics

Parameter	Symbol	Condition	Note	Typ.	Max.	Unit
Supply power voltage		TA = -40°C~+85°C	1.8		3.6	V
Flash write (erase, program) voltage		TA = -40°C~+85°C	1.8		3.6	V
Flash erase, programming cycle		TA = -40°C~+85°C	20,000		、	次
Flash data retention period		TA = +25°C	100			年

1.16 Controller ADC Characteristics

Parameter	Symbol	Condition	Note	Typ.	Max.	Unit
Supply power voltage		TA = -40°C~+85°C	2.4		3.6	V
Resolution				12		位
Integral nonlinearity		V _{DD} = 2.4V~3.6V, 400Ksps	-3.2		+3	LSB
Differential nonlinearity		V _{DD} = 2.4V~3.6V, 400Ksps	-2		+2	LSB
Offset error		V _{DD} = 2.4V~3.6V		+6	+10	LSB
SAR conversion clock				24		MHz
Conversion time in SAR clock				30		Clocks
Conversion rate		V _{DD} ≥ 2.7V		533		Ksps
		V _{DD} ≥ 2.4V		400		Ksps
VADC input voltage range		Single break (AIN+ to GND)	0		V _{DD}	V
Input capacitance				9.17	10.58	pF

Parameter	Symbol	Condition	Note	Typ.	Max.	Unit
Input sampling switch resistance ^[1]		$V_{DD} = 3.3V$		968		Ω
		$V_{DD} = 2.7V$		1050		Ω
Switch from VDD to resistor divider ($V_{DD}/2$)		CH0(V_{DD}) \rightarrow CH1($V_{DD}/2$, 51Kresistor divider)		4.3		μS
		CH0(V_{DD}) \rightarrow CH1($V_{DD}/2$, 10Kresistor divider)		0.7		μS
Switch from GND to resistor divider ($V_{DD}/2$)		CH0(GND) \rightarrow CH1($V_{DD}/2$, 51Kresistor divider)		4.1		μS
		CH0(GND) \rightarrow CH1($V_{DD}/2$, 10Kresistor divider)		0.6		μS
ADC power supply current		ADPS<1:0> = 00	2.3		2.9	mA
		ADPS<1:0> = 01	2.2		2.8	mA
		ADPS<1:0> = 10	2.1		2.6	mA
		ADPS<1:0> = 11	2		2.5	mA

Notes:

[1]. Data is guaranteed by design, not from test.

1.17 IVR Characteristics

Parameter	Symbol	Condition	Note	Typ.	Max.	Unit
Supply power voltage		$T_A = -40^{\circ}C \sim +85^{\circ}C$	2.4		3.6	V
Operating current		Normal power consumption state	43		67	μA
		Low power state		0.1		μA
Output voltage		$T_A = -40^{\circ}C \sim +85^{\circ}C$	1.37	1.4	1.43	V
Spreadoverthetemperature range		$V_{DD}=3.3V \pm 10mV$			13	mV

1.18 Controller Serial Port Timing Characteristics

Parameter	Symbol	Condition	URM0X3 = 0		URM0X3 = 1		Unit
			Min.	Max.	Min.	Max.	
SerialPortClockCycleTime	t_{XLXL}		12T		4T		T_{SYSCLK}
OutputDataSetuptoClockRisingEdge	t_{QVXH}		10T-20		2T-20		ns
OutputDataHoldafterClockRisingEdge	t_{XHGX}		T-10		T-10		ns
InputDataHoldafterClockRisingEdge	t_{XHDX}		5		5		ns
ClockRisingEdgetoInputDataValid	t_{XHDX}			2T-10		2T-10	ns

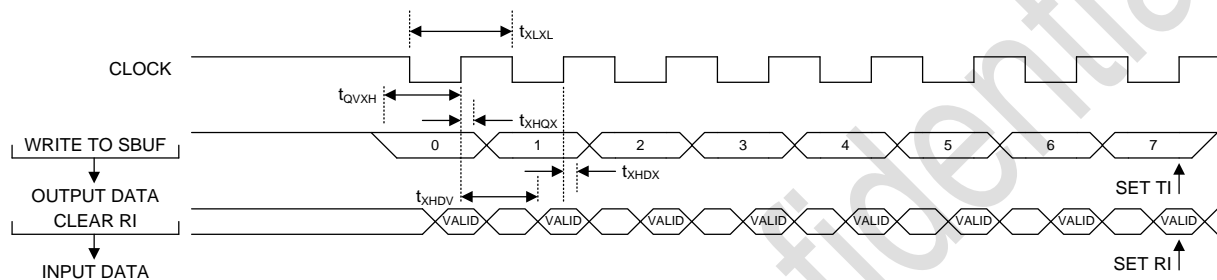


Figure 1-1. Shift Register Mode Timing Waveform

1.19 Controller SPI Timing Characteristics

Parameter	Symbol	Condition	Note	Typ.	Max.	Unit
SPICLKHighTime	t_{MCKH}	Mastermodetiming	1T			T_{SYSCLK}
SPICLKLowTime	t_{MCKL}		1T			T_{SYSCLK}
MISOValidtoSPICLKSampleEdge	t_{MIS}		10			ns
SPICLKShiftEdgeto MISOChange	t_{MIH}		0			ns
SPICLKShiftEdgetoMOSIChange	t_{MOH}				10	ns
nSSFallingto FirstSPICLKEdge	t_{SE}	Slavemodetiming	2T			T_{SYSCLK}
LastSPICLKEdgetonSSRising	t_{SD}		2T			T_{SYSCLK}
nSSFallingto MISOValid	t_{SEZ}				4T	T_{SYSCLK}
nSSRisingtoMISOHigh-Z	t_{SDZ}				4T	T_{SYSCLK}
SPICLKHighTime	t_{CKH}		2T			T_{SYSCLK}
SPICLKLowTime	t_{CKL}		2T			T_{SYSCLK}
MOSIValidtoSPICLKSampleEdge	t_{SIS}		1T			T_{SYSCLK}
SPICLKSampleEdgetoMOSIChange	t_{SIH}		1T			T_{SYSCLK}
SPICLKShiftEdgetoMISOChange	t_{SOH}				2T	T_{SYSCLK}
Last SPICLK Edge to MISO Change(CPHA= 1ONLY)	t_{SLH}		1T		2T	T_{SYSCLK}

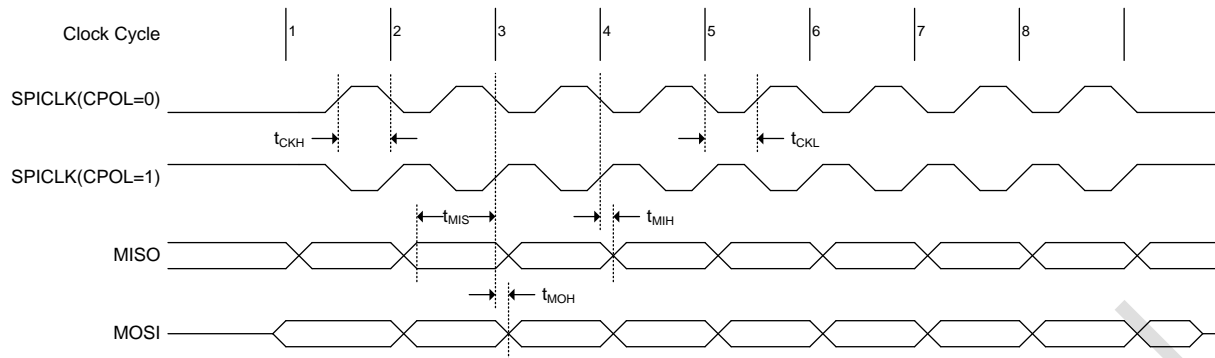


Figure 1-2. SPI Master Transfer Waveform with CPHA=0

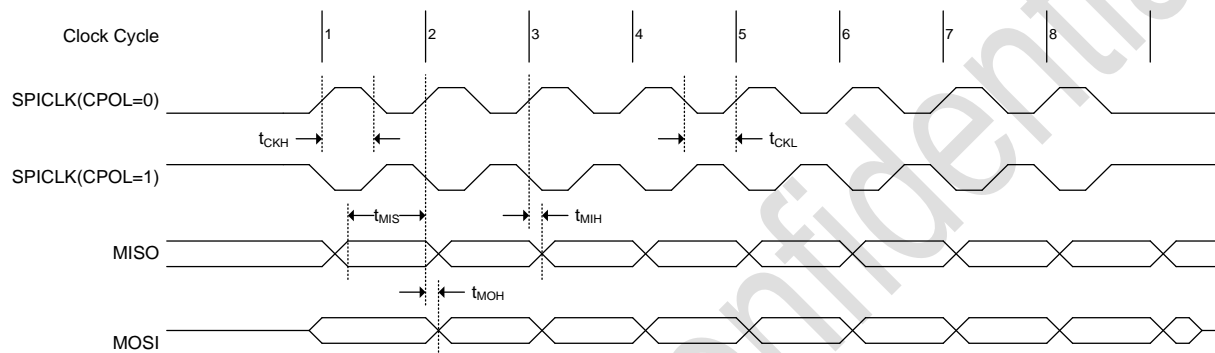


Figure 1-3. SPI Master Transfer Waveform with CPHA=1

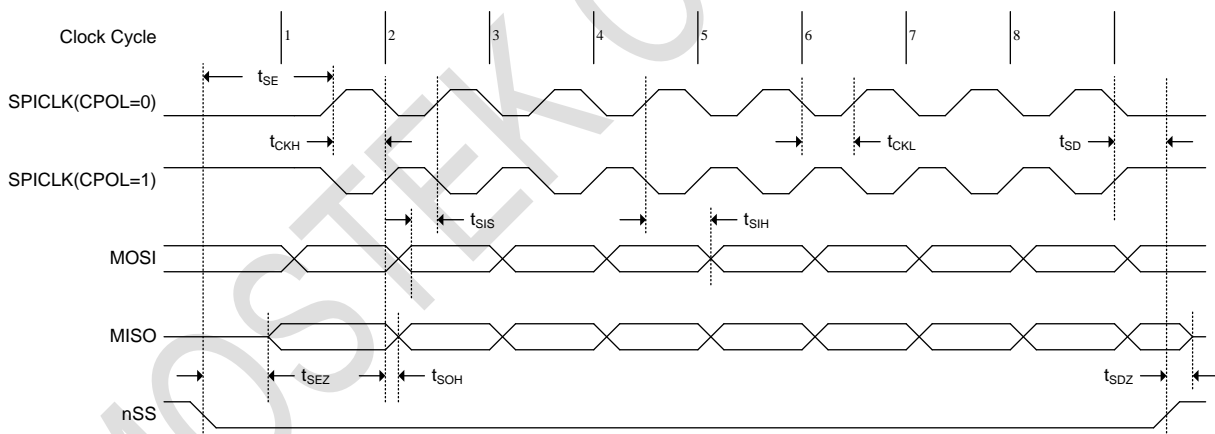


Figure 1-4. SPI Slave Transfer Waveform with CPHA=0

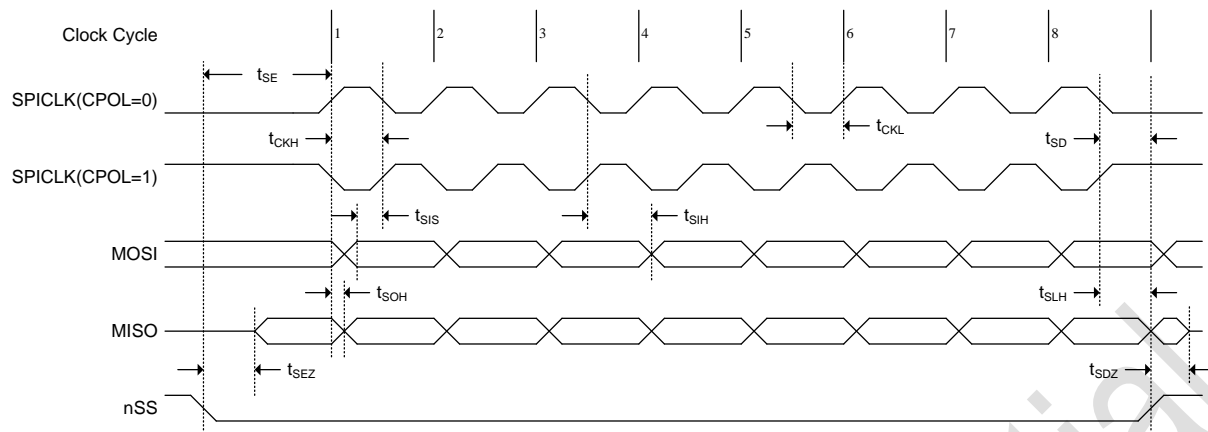
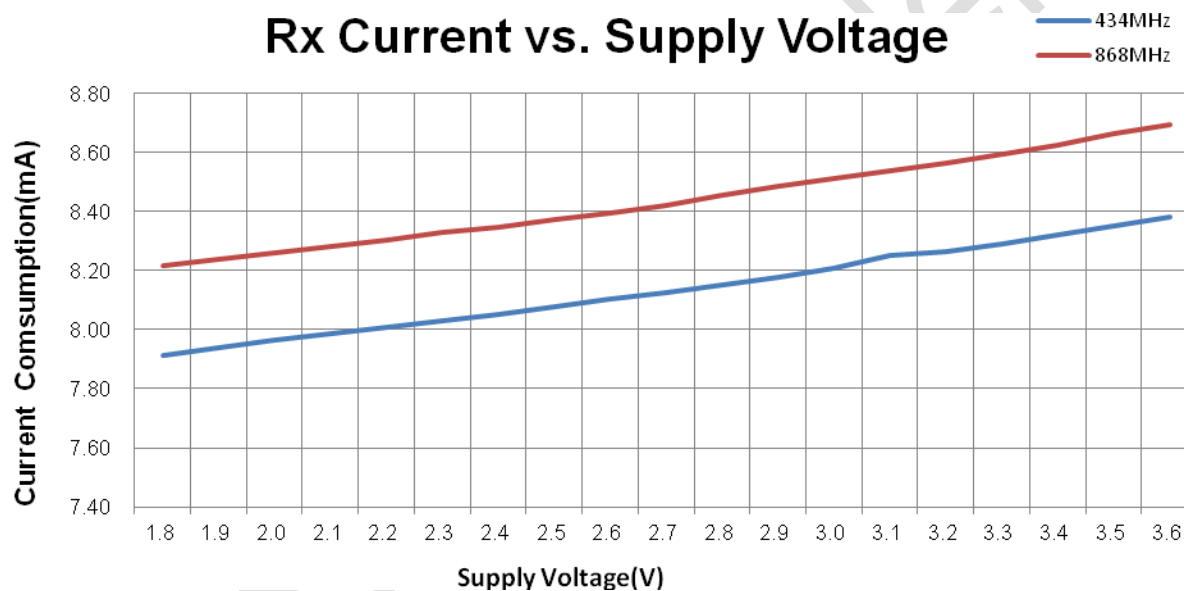


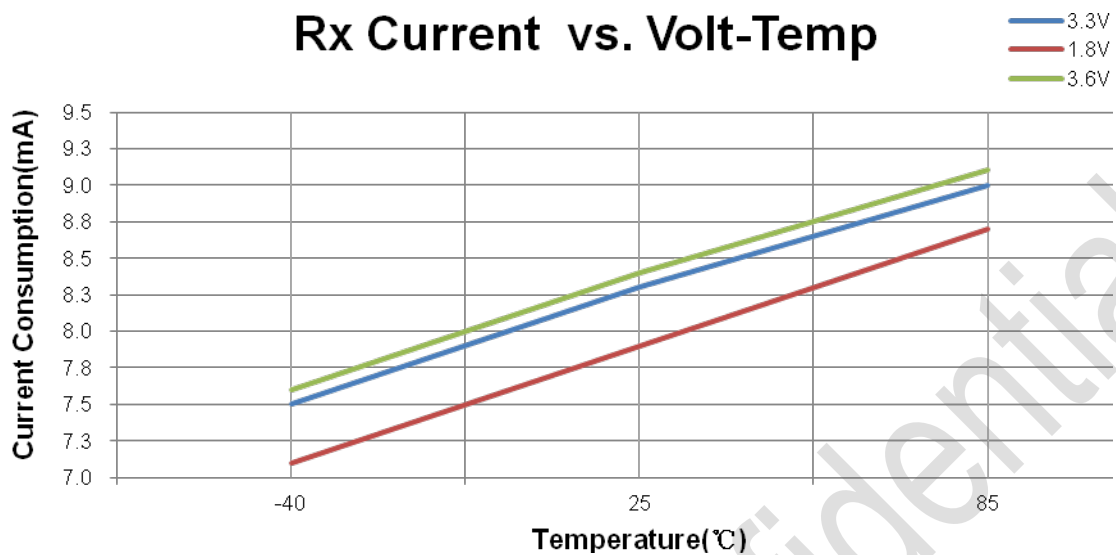
Figure1-5. SPI Slave Transfer Waveform with CPHA=1

1.20 Receive Current and Supply Voltage Correlation

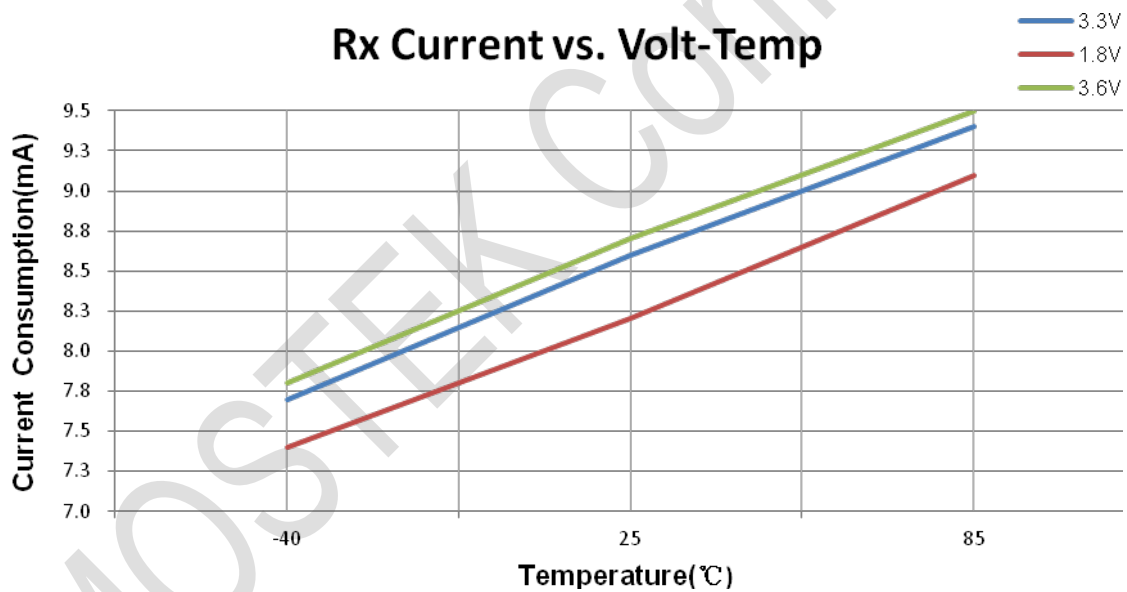


Test Conditions: Freq=434MHz / 868MHz, Fdev=10KHz, BR=10Kbps

1.21 Correlation Among Receive Current, Supply Voltage and Temperature

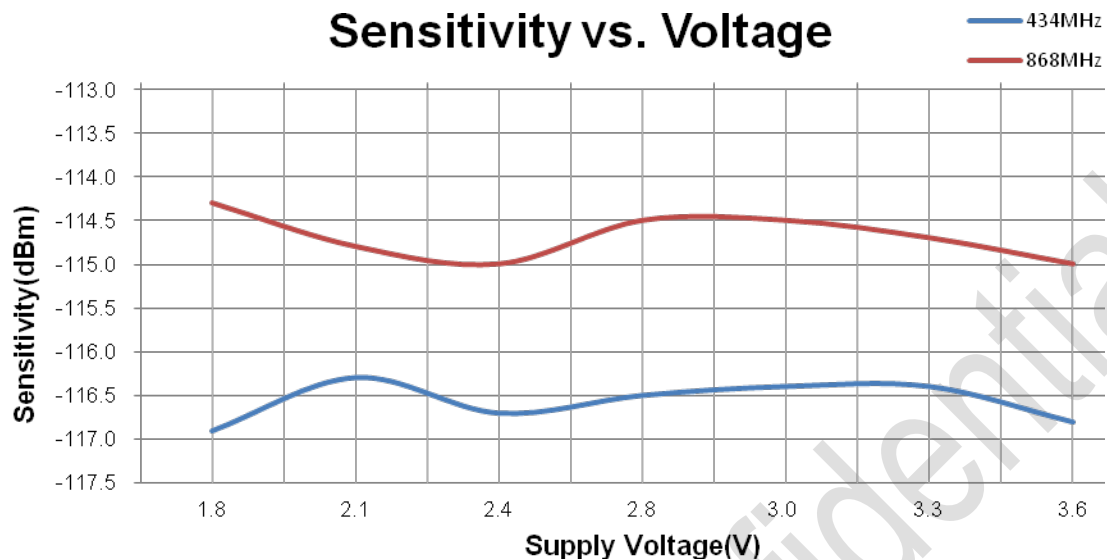


Test Conditions: Freq = 434MHz, Fdev = 10KHz, BR = 10Kbps



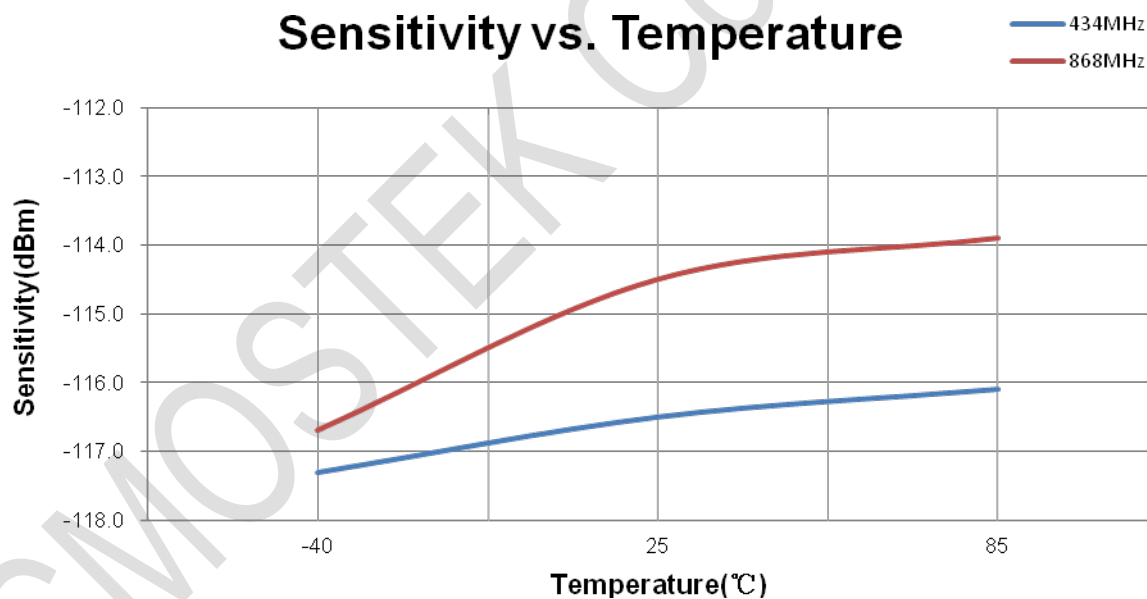
Test Conditions: Freq = 868MHz, Fdev = 10KHz, BR = 10Kbps

1.22 Receive Sensitivity and Supply Voltage Correlation



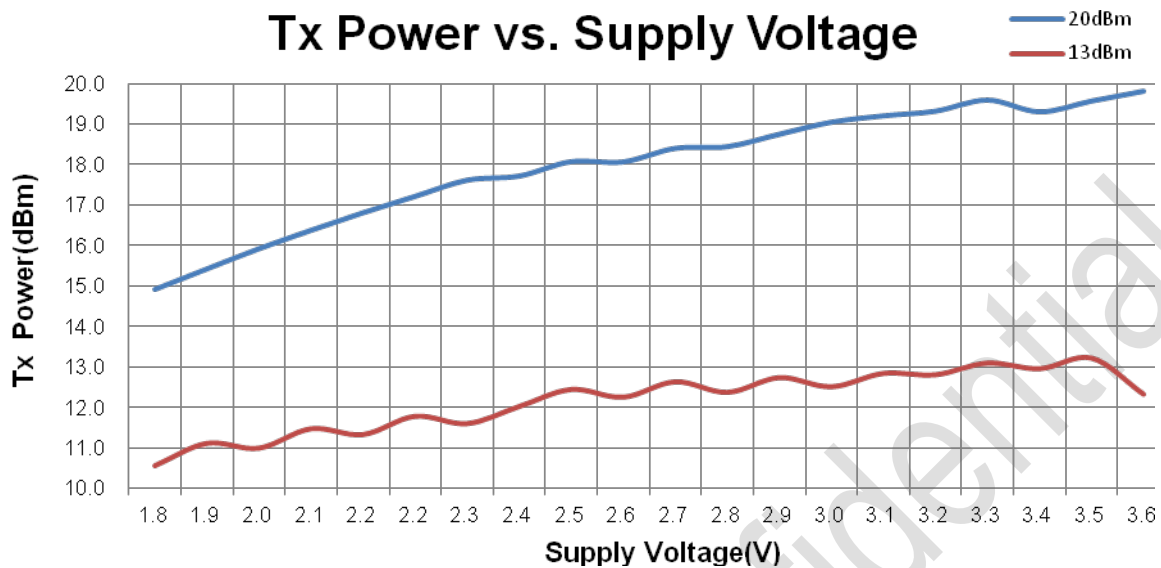
Test Conditions: FSKmodulation, DEV = 10KHz, BR = 10Kbps

1.23 Receive Sensitivity and Temperature Correlation

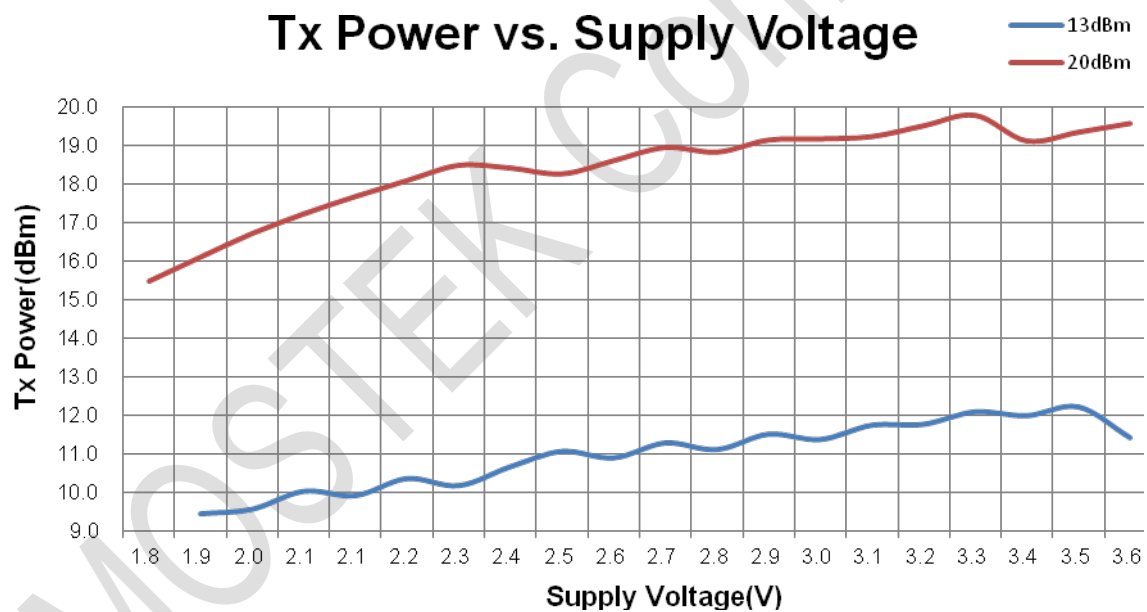


Test Conditions: FSKmodulation, DEV = 10KHz, BR = 10Kbps

1.24 Transmit Power and Supply Voltage Correlation



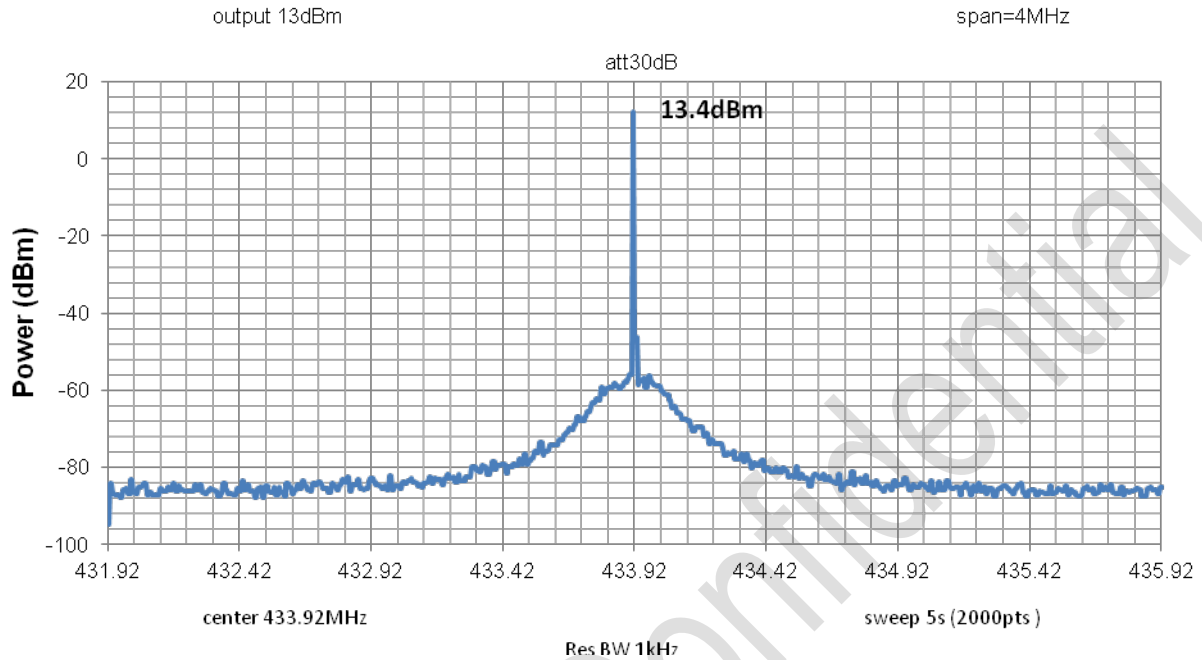
Test Conditions: Freq = 434 MHz, 20 dBm & 13 dBm matching network respectively



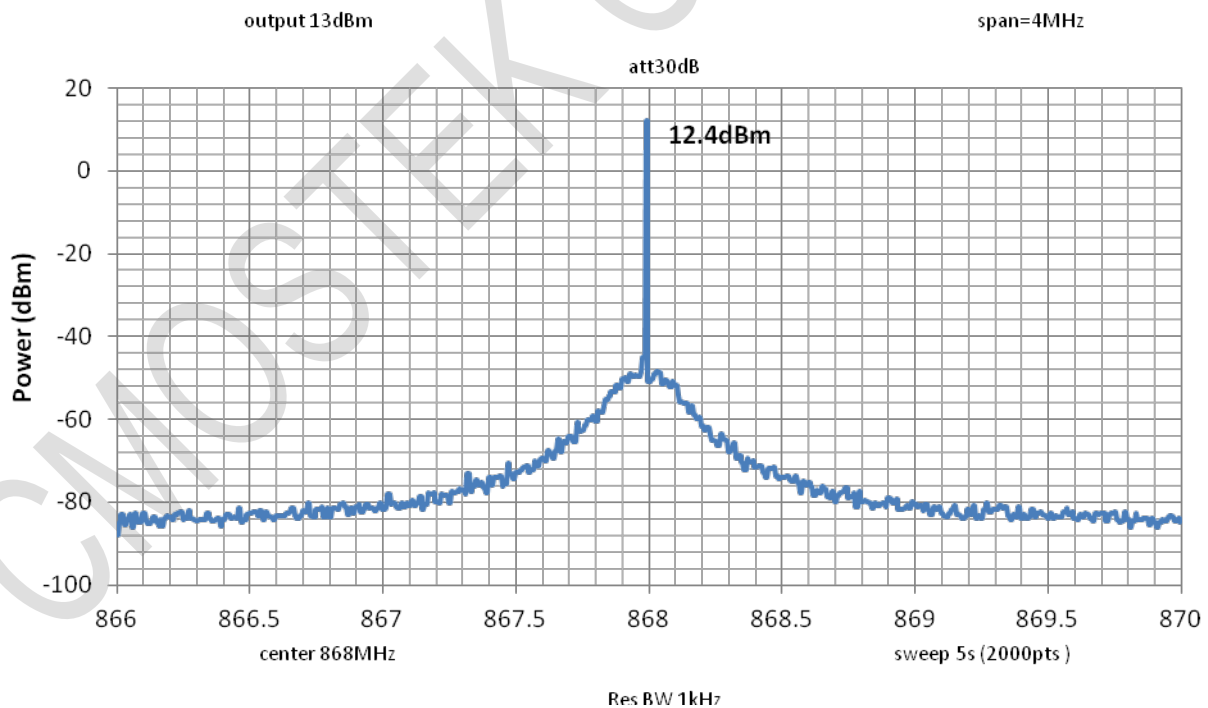
Test Conditions: Freq = 868MHz, 20dBm / 13dBm matching network

1.25 Phase Noise

433.92MHz Phase Noise



868MHz Phase Noise



2 Pin Description

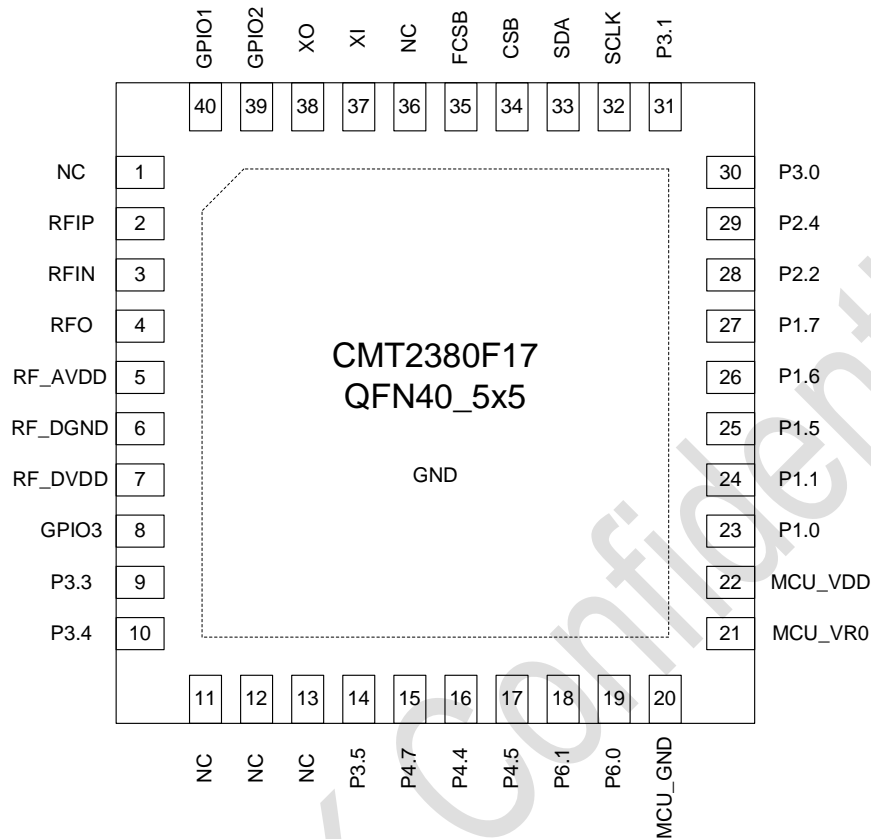


Figure1. CMT2380F17 Pin Arrangement Diagram

Table 2. CMT2380F17 Pin Description

Pin #	Pin Name	I/O	Description
0	GND	Analog	Chip substrate, must connect to ground
1	NC	-	No connection
2 - 3	RFIP/RFIN	Analog	Differential RF signal input port
4	FRO	Analog	PA output
5	RF-AVDD	Analog	RF circuit VDD, requires to connect to a 1.8-3.6 V power supply
6	RF-DGND	Digital	Digital GND of RF module.
7	RF-DVDD	Digital	Digital VDD of RF module, requiring to connect to a 1.8-3.6 V power supply.
8 ^[1]	GPIO3	IO	The GPIO3 of RF module. It can be configured as: CLK0, DOUT/DIN, INT2, and DCLK (TX/RX).
9	P3.3	IO	Port P3.3
	nINT1	I	Input signal of external interrupt 1
	KBI4	I	Channel 4 of keyboard input
	CEX1	I	External IO of PCA0 module 1
	nSS	I	Chip selection signal of SPI module slave mode
	T3/T3CKO	IO	External clock input/programmable clock output of Timer 3

Pin #	Pin Name	I/O	Description
10	P3.4	IO	Port P3.4
	T0/T0CKO	IO	External clock input/programmable clock output of Timer 0
	CEX3	IO	External IO of PCA0 module 3
	T3EX	I	External control input signal of Timer 3
	T5	I	External clock input signal of Timer 5
11	NC	-	Not connect
12	NC	-	Not connect
13	NC	-	Not connect
14	P3.5	IO	Port P3.5
	T1/T1CKO	IO	External clock input signal/programmable clock output signal of Timer 1
	CEX5	IO	External IO of PCA0 module 5
	T6	I	External clock input signal of Timer 6
15	P4.7	IO	Port P4.7
	RST	I	MCU reset pin
	C0CKO	O	Programmable clock output of PCA reference counter
	S0CKO	O	programmable S0BRG clock output
	S1CKO	O	programmable S1BRG clock output
16	P4.4	IO	Port P4.4
	OCD_SCL	IO	Clock signal of OCD debug interface
	nINT2	I	Input signal of external interrupt 2
	BEEP	O	Buzzer output
	ECI	I	External clock input of PCA
17	P4.5	IO	Port P4.5
	OCD_SDA	IO	Data signal of OCD debug interface
	nINT0	I	Input signal of external interrupt 0
	RTCKO	O	Programmable RTC clock output
18	P6.1	IO	Port P6.1
	S1MI	I	SPI master mode data input of UART1
	PWM7	O	PWM output of PCA0 module 7
	PWM2B	O	Sub-channel B of PCA PWM2 output
19	P6.0	IO	Port P6.0
	ECKI	I	Clock input pin in the external clock input mode
	ICKO	O	MCK output signal of internal clock
	PWM6	O	PWM output of PCA0 module 6
	PWM2A	O	Sub-channel A of PCA PWM2 output
20	MCU-GND	Analog	Digital Ground
21	MCU-VCAP	Analog	Reference voltage, connecting 0.1 uF and 4.7 uF parallel capacitors to ground.
22	MCU-VDD	Analog	MCU VDD, requiring to connect a 1.8-3.6 V power supply.
23	P1.0	IO	Port P1.0

Pin #	Pin Name	I/O	Description
	AIN0	I	Channel 0 of ADC analog input
	KBI0	I	Channel 0 of keyboard input
	T2/T2CKO	IO	Output signal of Timer 2 external clock input/programmable clock
	RxD1	I	RxD signal of UART1 module
24	P1.1	IO	Port P1.1
	AIN1	I	Channel 1 of ADC analog input
	KBI1	I	Channel 1 of keyboard input
	T2EX	I	Input signal for Timer 2 external control
	TxD1	O	TxD signal of UART1 module
25	P1.5	IO	Port P1.5
	AIN5	I	channel 5 of ADC analog input
	KBI5	I	Channel 5 of keyboard input
	MOSI	O	Signal of SPI module for master output and slave input
26	P1.6	IO	Port P1.6
	AIN6	I	Channel 6 of ADC analog input
	KBI6	I	Channel 6 of keyboard input
	MISO	I	Signal of SPI module for master input and slave output
	S0MI	I	Data input of UART0 SPI master mode
	PWM0A	O	Sub-channel A of PCA PWM0 module output
27	P1.7	IO	Port P1.7
	AIN7	I	Channel 7 of ADC analog input
	KBI7	I	Channel 7 of keyboard input
	SPICLK	I	Clock signal of SPI module
	CEX4	IO	External IO of PCA0 module 4
	PWM0B	O	Sub-channel B of PCA PWM0 output
28	P2.2	IO	Port P2.2
	AIN2	I	Channel 2 of ADC analog input
	CEX0	IO	External IO of PCA0 module 0
29	P2.4	IO	Port P2.4
	AIN3	I	Channel 3 of ADC analog input
	CEX2	IO	External IO of PCA0 module 2
30	P3.0	IO	Port P3.0
	AIN4	I	Channel 4 of ADC analog input
	KBI2	I	Channel 2 of keyboard input
	TWI0_SDA	IO	Data signal of TWI0/I2C0 module
	RxD0	I	RxD signal of UART0 module
31	P3.1	IO	Port P3.1
	KBI3	I	Channel 3 of keyboard input
	TWI0_SCL	O	Clock signal of TWI0/I2C0 module

Pin #	Pin Name	I/O	Description
	TxD0	O	TxD signal of UART0 module
32 ^[2]	SCLK	I	RF SPI clock
33 ^[2]	SDA	IO	RF SPI data input/output, connecting to 10 kΩ pull-up resistor externally
34 ^[2]	CSB	I	RF SPI chip selection for register access
35 ^[2]	FCSB	I	RF SPI chip selection for FIFO access
36	NC	-	Not connect
37	XI	I	26 MHz crystal circuit input
38	XO	O	26 MHz crystal circuit output
39 ^[2]	GPIO2	IO	GPIO2 of RF module; it can be configured as:INT1,INT2,DOUT/DIN,DCLK (TX/RX),RF_SWT.
40 ^[2]	GPIO1	IO	GPIO1 of RF module; it can be configured as:DOUT/DIN,INT1,INT2,DCLK (TX/RX),RF_SWT

Notes:

[1]. INT1 and INT2 refer to RF interrupts. DOUT refers to the demodulated data output. DIN refers to the modulation data input. DCLK refers to the modulation or demodulation data rate synchronous clock, which is switched automatically according to operation mode switch between TX and RX.

[2]. The SPI and GPIO of RF module are not connected to MCU ports. Users need to select the appropriate port of the MCU for off-chip connection according to application requirements (namely connecting to the PCB board).

CMT2380F17 can operate at a much lower speed and thereby greatly reduce the power consumption.

The CMT2380F17 has 16K bytes of embedded Flash memory for code and data. The Flash memory can be programmed either in serial writer mode (via ICP, In-Circuit Programming) or in In-System Programming mode. And, it also provides the In-Application Programming (IAP) capability. ICP and ISP allow the user to download new code without removing the microcontroller from the actual end product; IAP means that the device can write non-volatile data in the Flash memory while the application program is running. There needs no external high voltage for programming due to its built-in charge-pumping circuitry.

In addition to standard 80C51 functions, CMT2380F17 also adds enriched peripherals. Please see the table below for details.

Table 3-1. CMT2380F17 Peripheral Resource List

Resource Name	Standard 80C51 Peripheral	CMT2380F17 Peripheral	Notes
IRAM	256 Bytes	256 Bytes	
External Interrupt	2	3	Support high/low trigger selection
Interrupt source	4 Interrupt sources	16 interrupt sources, 4 priority levels	
UART0	Standard	Enhanced serial port	
UART1		Support	
Timer 0	Support	Support	
Timer 1	Support	Support	
Timer 2	Support	Support	
Timer 3		Support	
DMA		Support	
IO port		17	
XRAM		768 Bytes	
ADC		12-bit	≥500Ksps sampling rate
PCA0		8-ch	6ch counter/capture/PWM 2ch capture/PWM
SPI		1	
TWI		2	TWI & STWI
Keyboard interrupt (KBI)		8-ch	
Watchdog (WDT)		Support	
Real Time Clock (RTC)		Support	
Low voltage detector		2	
External crystal oscillator circuit		Support	
Internal high-speed RC oscillation		1	High-precision IHRC
Internal low-speed RC oscillation		1	
Clock multiplier		1	Core frequency multiplier

The CMT2380F17 supports a variety of operating modes: idle mode, power-down mode, slow-frequency mode, sub-frequency mode, RTC mode, watch mode and monitor mode. Applying or combining these operating modes can effectively reduce system power consumption.

- Idle mode: the CPU is frozen while the peripheral modules and interrupt system are still active.

- Power-down mode: the value of random access memory RAM and special function register SFR is saved, and all other functions are terminated. Most importantly, the microcontroller in power-down mode can be awakened by various interrupts or resets.
- Slow-frequency mode: users can slow down the system speed through the 8-bit system clock divider to reduce power consumption.
- Sub-frequency mode: the system clock applies the internal low-speed oscillator and the CPU runs at a particularly slow speed.。
- Real-time clock (RTC) mode: 支 it supports the real-time clock function in all modes.
- Watch mode: in power-down mode or idle mode, WDT overflow is used as an automatic timer to wake up the CPU.
- Monitor mode: detect the voltage in power-down mode, and reset when the voltage is particularly low.

The CMT2380F17 is equipped with a Megawin proprietary On-Chip Debug (OCD) interface for In-Circuit Emulator (ICE). The OCD interface provides on-chip and in-system non-intrusive debugging without any target resource occupied. Several operations necessary for an ICE are supported, such as Reset, Run, Stop, Step, Run to Cursor and Breakpoint Setting.

4 Sub-GHz Transceiver

4.1 Transmitter

The CMT2380F17 transmitter is based on direct RF synthesizer. Its carrier frequency is generated by a low noise fractional frequency synthesizer. The modulated data is transmitted by an efficient single-ended power amplifier (PA). The output power can be read and written by registers, which is configurable ranging from -20dBm to +20dBm with 1dB step.

When the PA switches quickly, its changed input impedance instantaneously interferes with the output frequency of the VCO. This effect becomes a VCO pull that produces spectral spurs and glitches near the target carrier. By ramping the PA output power, it can minimize the instantaneous glitch of the PA. The CMT2380F32 has a built-in ramping mechanism. When the PA ramp is enabled, the PA output power can be ramped to the required value by the configured speed in order to reduce the undesired spectral spectrum. In FSK mode, the transmitter supporting signal is transmitted after Gaussian filtering, meaning GFSK, to make the transmitting spectrum more concentrated.

Users can design a PA matching network based on specific application requirements to optimize the transmission efficiency at the required output power. Typical application schematics and required BOMs are detailed above. For more application schematic details and layout guidelines, please refer to AN141 CMT2300A Schematic and PCB Layout Guide.

The transmitter can work in pass-through mode and packet mode respectively. In pass-through mode, data is sent to the chip directly through the DIN pin of the chip and transmitted directly. In packet mode, data is preloaded into the FIFO of the chip in STBY status and transmitted then along with other packet elements.

4.2 Receiver

An ultra-low power, high performance low IF OOK, FSK receiver is built in the CMT2380F17. It follows processing steps as: 1) The RF signal sensed by the antenna is amplified by the low noise amplifier. 2) The signal is down-converted to the intermediate frequency by the quadrature mixer and then filtered by the image rejection filter. 3) The signal is further amplified by the limiting amplifier. 4) The signal is sent to the digital domain for digital demodulation processing. Each analog module is calibrated to an internal reference voltage during power-on reset (POR). This allows the chip to perform better at different temperatures and voltages. Baseband filtering and demodulation is done by a digital demodulator. When the chip is working in an environment with strong out-of-band interference, the automatic gain control loop adjusts the gain of the system through the wideband power detector and attenuation network next to the LNA to achieve optimal system linearity, selectivity, sensitivity, etc.

Following the low-power design techniques of CMOSTEK, an ultra-low power is consumed even when the receiver keeps operating for long period. Its periodic operating mode and air wakeup feature further reduce the average power consumption of the system, serving well in applications where power consumption is critical.

Similar to the transmitter, the CMT2380F17 receiver can operate in pass-through mode and packet mode respectively. In the pass-through mode, the data output by the demodulator can be directly output through the DOUT pin of the chip. DOUT can be configured from GPIO1/2/3. In the packet mode, data processing follows steps as: 1) The data output of the demodulator is firstly sent to the packet processor for decoding. 2) The data is filled into the FIFO. 3) The controller portion of the CMT2380F17 reads the FIFO through the SPI interface.

4.3 Transceiver Power-on Reset (POR)

The power-on reset circuit assists in power supply change detection and generates a corresponding reset signal to reset the entire RF system (the RF portion of the CMT2380F32). The CMT2380F32 controller can reinitialize the RF system after POR. The two cases for POR reset generation are as follows:

1. A rapid power supply mutation triggers POR reset under the condition that RF-VDD (RF system power supply, the same below) drops $0.9V \pm 20\%$ ($0.72V - 1.08V$) within less than 2 μs . Note that it monitors RF-VDD decrease instead of its absolute value as shown in the below figure.

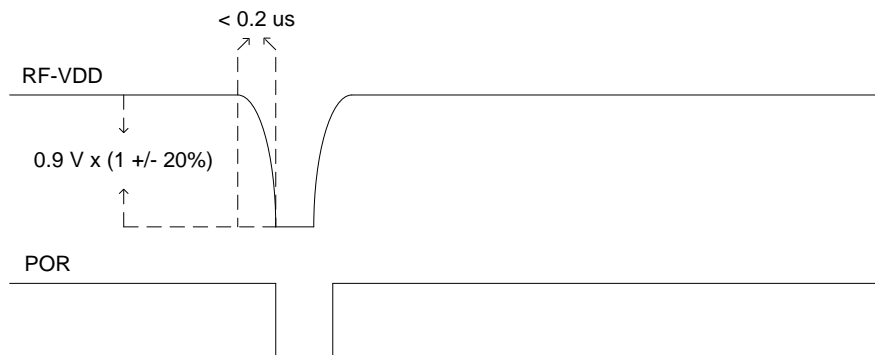


Figure 4-1. Rapid RF-VDD Drop Triggers POR Reset

2. A slow supply power drop triggers POR reset under the condition that RF-VDD drops to $1.45V \pm 20\%$ ($1.16 - 1.74V$) within less than 2 μs . Note that it monitors the absolute value of RF-VDD instead of RF-VDD decrease as shown in the below figure.

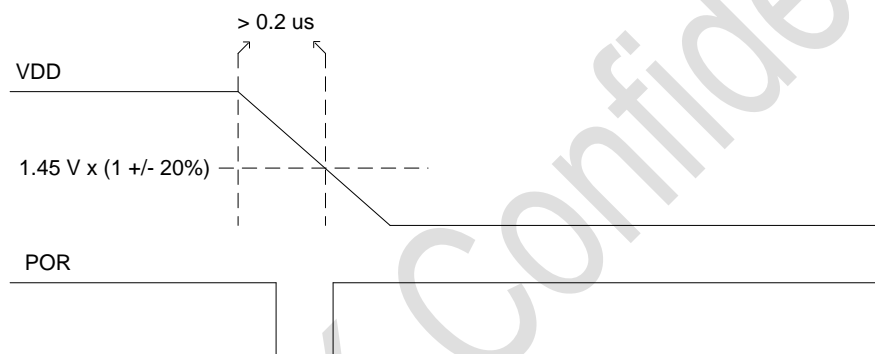


Figure 4-2. Slow RF-VDD Drop Triggers POR Reset

4.4 Transceiver Crystal Oscillator

The crystal oscillator provides both a reference clock for the phase-locked loop and a system clock for the digital part. The load capacitance depends on the crystal specified CL parameters. The total load capacitance between XI and XO should be equal to CL ensuring the crystal oscillates accurately at 26 MHz.

$$C_L = \frac{1}{1/C_{15} + 1/C_{16}} + C_{par} + 2.5pF$$

C15 and C16 are the load capacitances reside at both ends of the crystal. Cpar is the parasitic capacitance resides on the PCB. Each pin of the crystal has a 5pF parasitic capacitance inside, as an equivalence of 2.5pF altogether. The equivalent series resistance of the crystal should meet the specified specifications to ensure reliable crystal start-up. Alternatively the traditional crystal can be replaced with an external source connected to the XI pin. This clock signal is recommended to have a peak-to-peak level between 300 mV and 700 mV and be coupled to the XI pin with a coupling capacitor.

4.5 Transceiver Built-in Low Frequency Oscillator (LPOSC)

The CMT2380F17RF system integrates a sleep timer driven by a 32 kHz low power oscillator (LPOSC). When this function is enabled, the timer periodically wakes up the chip from sleep mode. Sleep time can be configured from 0.03125 ms to 41,922,560 ms when the chip is in periodical operating mode. Due to the low power oscillator frequency will change with the temperature and voltage drift, it will be automatically calibrated during power on and will be periodically calibrated since then. These calibrations will keep the oscillator's frequency tolerance within $\pm 1\%$.

4.6 Transceiver Built-in Low Battery Detection

The chip is employed with low battery detection function which is performed each time when the frequency is tuned. Frequency tuning occurs when the chip transitions from the SLEEP/STBY state to the RFS/TFS/TX/RX state. The detection result can be read by the LBD_VALUE register.

4.7 Receiver Signal Strength Indication (RSSI)

The RSSI is used to evaluate the strength of the signal within the tuned channel. The cascaded I/Q logarithmic amplifier amplifies the signal before it is sent to the demodulator. The receive signal indicators inside the logarithmic amplifiers of I channel and Q channel produce DC voltage that is proportional to the input signal strength. The RSSI output is the sum of the two signal values, which extend a dynamic range of 80 dB based on the sensitivity. After the signal strength is sampled by the ADC, a smoother RSSI value is obtained through a SAR filter and a smoothing filter. The order of the smoothing filter can be set via RSSI_AVG_MODE<2:0>. After filtering, the code value is converted into a dBm value. Users can obtain either the RSSI code value (RSSI_CODE<7:0>) or dBm value (RSSI_DBM<7:0>) by reading the register. With configuring the value of RSSI_DET_SEL<1:0>, users can choose to either output the RSSI value in real time or store RSSI value at each stage during packet receiving.

The CMT2380F17 supports users to set RSSI_TRIG_TH<7:0> threshold. After a comparison between the threshold value and detected RSSI value, the comparison outputs logic 1 if the RSSI detection value is more than the threshold value, otherwise it outputs logic 0. The comparison output can be output to the RSSI VLD interrupt or it can support the operation of the internal super-low power (SLP) mode.

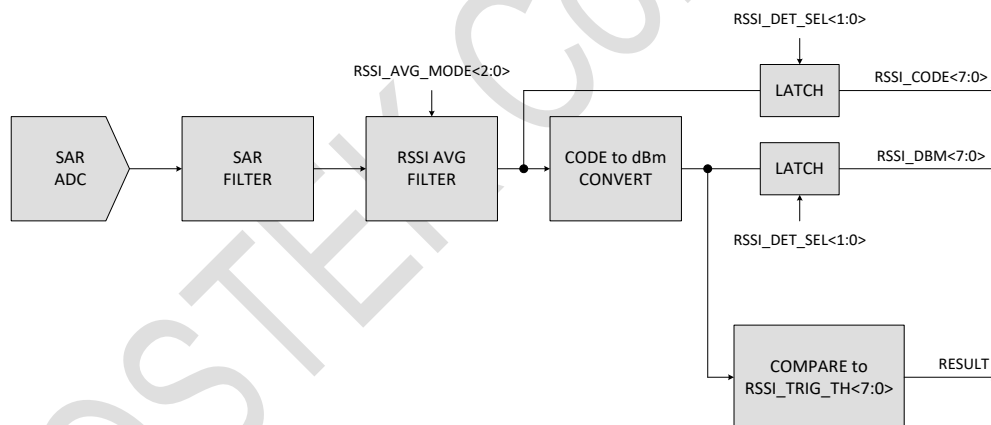


Figure 4-3. RSSI Measuring and Comparing Circuit Structure

The CMT2380F17 offers RSSI to meet the qualitative analysis requirements of users generally. However more accurate RSSI measurement results are needed in case of quantitative analysis, therefore users need to perform production calibration based on actual solutions. Please refer to AN144- CMT2300A RSSI User Guide for details.

4.8 Phase Jump Detector (PJD)

PJD refers to the phase jump detector. During the chip performing FSK demodulation, it can be used to identify useful signals from noise via observing the hopping characteristics of received signals.

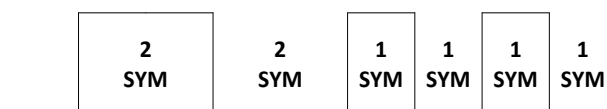


Figure 4-4.Receive Signal Jump Diagram

PJD identifies an input signal transition from 0 to 1 or from 1 to 0 as a phase jump. Users simply configure PJD_WIN_SEL<1:0> to indicate PJD how many signal transitions need to be detected before output the judgment result. As shown in above diagram, 6 phase jumps occurred among the total of 8 symbols received, meaning the jump number is not equal to the number of symbols. The jump count is equivalent to the number of symbols only if preamble is received. In general, the more signal transition detected, the more reliable the judgment is. The less, the faster the detection completes. If the receiving time window is small, the number of detections needs to be reduced to meet the window setting requirements. In general, 4 signal transitions detection can ensure reliable detection, that is, neither misjudging noise as a useful signal nor the detection failure of a useful signal will happen.

Monitoring signal transitions is essentially monitoring whether the signal meets the expected data rate, meanwhile, the PJD also automatically monitors whether the error of the signal meets the agreed value of the valid signal and determines if the SNR exceeds 7 dB. It outputs 1 if it determines a reliable signal is received based on the combined data rate, error and SNR monitoring results, otherwise it outputs 0 if it determines the signal is noise or interference signal. This result can be output to the RSSI VLD interrupt or to the implementation instance of the internal assisting super-low power (SLP) receive mode. In the direct mode, the FSK demodulation output can also be muted through setting the DOUT_MUTE register bit to 1 based on phase jump detect result.

PJD is similar to traditional carrier sense(CS) technique, more reliable though. A combine of RSSI monitoring and PJD technique ensures accurate channel state identification.

4.9 Receiver Clock Data Recovery (CDR)

CDR system is largely to recover the clock signal synchronized with the data rate while receiving data, either for decoding inside the chip or outputting to the GPIO for users to sample data. Error existing between the recovered clock frequency and the actual transmitted data rate will cause data acquisition errors, error codes and decoding errors during data reception. Therefore, CDR takes a simple but critical role.

The CMT2380F17 receiver supports three CDR systems according to different application requirements:

- **COUNTING system**

This system is designed for the case where the data rate is relatively accurate. If the data rate is 100% aligned, users can even receive unlimited lengths of 0 without error.

- **TRACING system**

This system is designed for dealing with large data rate errors. It has a tracking function that automatically detects the data rate transmitted by TX and adjusts the RX local data rate promptly to minimize the error between them. This system leads the industry by supporting up to 15.6% error.

- **MANCHESTER system**

This system comes from COUNTING system by inheriting its basic features except a difference on specific design for Manchester codec, which performs special treatment in case of a sudden change in the TX data rate to identify the signal part with sudden changes.

4.10 Fast Manual Frequency Hopping

Manual frequency hopping refers to switching an original basic frequency point obtained by RRPDK configuration, e.g. 433.92MHz, to another frequency point by simply setting 1 or 2 registers on MCU during the application, which simplify user's operation much for switching frequency points frequently in multi-channel applications.

$$FREQ = \text{Basic Frequency Point} + 2.5 \text{ kHz} \times FH_OFFSET <7:0> \times FH_CHANNEL <7:0>$$

General processing steps follow: 1) Set FH_OFFSET<7:0> during the initialization configuration of power-up. 2) Switch channels constantly as desired in the application by changing FH_CHANNEL<7:0>.

When fast manual frequency hopping in the receiving mode is performed, it needs to have special process on AFC parameters. Please refer to *AN197-CMT2300A-CMT2119B-CMT2219B Fast Manual Frequency Hopping* and *CMT2300as well as CMT2219B Frequency Hopping Calculation Table* for more details.

4.11 Transceiver Control Interface and Operating Mode

4.11.1 Transceiver SPI Interface Timing

The RF system of CMT2380F17 communicates with the controller section via a 4-wire SPI port (FCSB, CSB, SDA and SCLK). The low active CSB is the chip selection signal used to access the registers. The low active FCSB is the chip selection signal used to access the FIFO. The above two cannot be set to low both at the same time. SCLK is a serial clock with speed up to 5MHz. Data is sent on the falling edge of SCLK and collected on the rising edge for the chip itself or an external MCU. SDA is a bidirectional pin for inputting and outputting data. Both the address and data parts are transmitted from the MSB.

The CSB is pulled low when registers are accessed. An R/W bit followed by a 7-bit register address is sent. After the controller pulls CSB low, it must wait for at least half a SCLK cycle before it can start transmitting R/W bits. After the controller sends the falling edge of the last SCLK, it must wait for at least half of the SCLK cycle before pulling CSB high.

It should be noted that, as for read register operations, both the controller and the transceiver may generate a switch IO (SDA) port event among address 0 and data 7. At this point, SDA will switch the IO port from input to output, and the controller will switch the corresponding IO port from output to input. In the below figure, please notice the position of the dotted line in the middle, at this time, it is strongly recommended the controller switches the IO port to input before it sends the falling edge of SCLK. The transceiver will switch the IO to output after it receives the falling edge, which avoids situations where the both set SDA to output resulting in electrical conflicts.

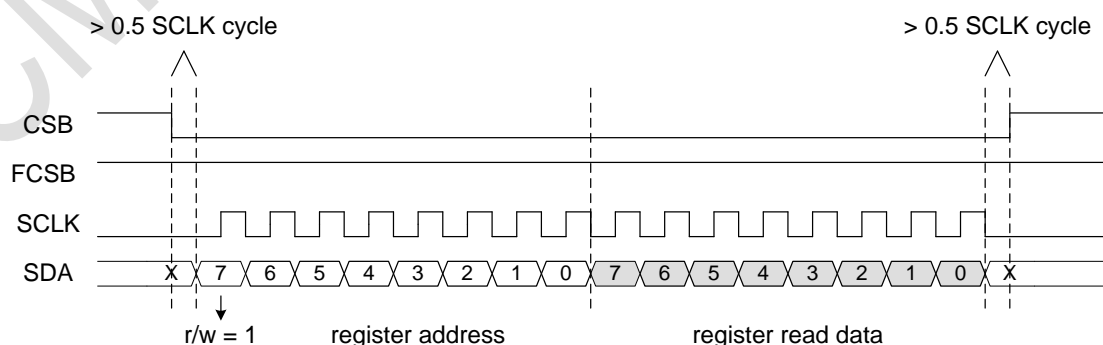


Figure 4-5. Transceiver SPI Read Register Timing

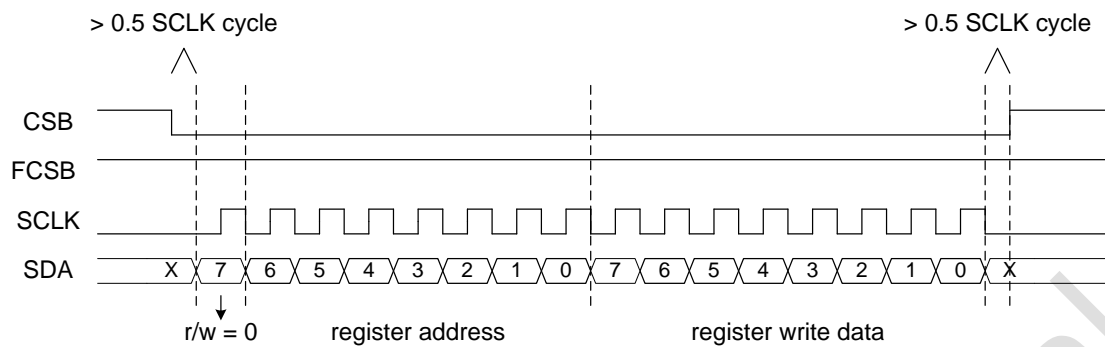


Figure 4-6. Transceiver SPI Write Register Timing

4.11.2 Transceiver FIFO Interface Timing

By default, the transceiver provides two independent 32-byte FIFOs for RX and TX. The RX FIFO is used to store received data in RX mode, and the TX FIFO is to store data to be transmitted in TX mode. Users can set FIFO_MARGE_EN to 1 as well, thus the two FIFOs are combined into a 64-byte FIFO, available both in TX and RX. By configuring FIFO_RX_TX_SEL, it indicates whether TX or RX currently is used. In no-combine case, when 32-byte RX FIFO is filled in, to save system operation time, users can fill in the 32-byte TX FIFO simultaneously for the next transmission.

The FIFO can be accessed via the SPI interface. Users can clear the FIFO by setting the FIFO_CLR_TX/FIFO_CLR_RX bit and transmit repeated data filled previously by setting FIFO_RESTORE, which avoids data refilling.

When accessing the FIFO, users start from configuring a number of registers involving FIFO read/write mode and some operating modes settings, as well as some other working modes. Please see AN143-CMT2300A FIFO and Packet Format Usage Guide for more details. As shown in below timing diagram for reading and writing, it should be noted that the FCSB control and CSB control for register access are slightly different. At the beginning of the access, the FCSB pulls down one clock cycle before sending the rising edge of SCLK. After sending the falling edge of the last SCLK, it pulls up FCSB at least 2us later. The FCSB must keep pulled up for at least 4us between two consecutive read and write operations. When writing to the FIFO, the first bit of the data must be ready a half of clock cycle before the first rising edge of SCLK is sent.

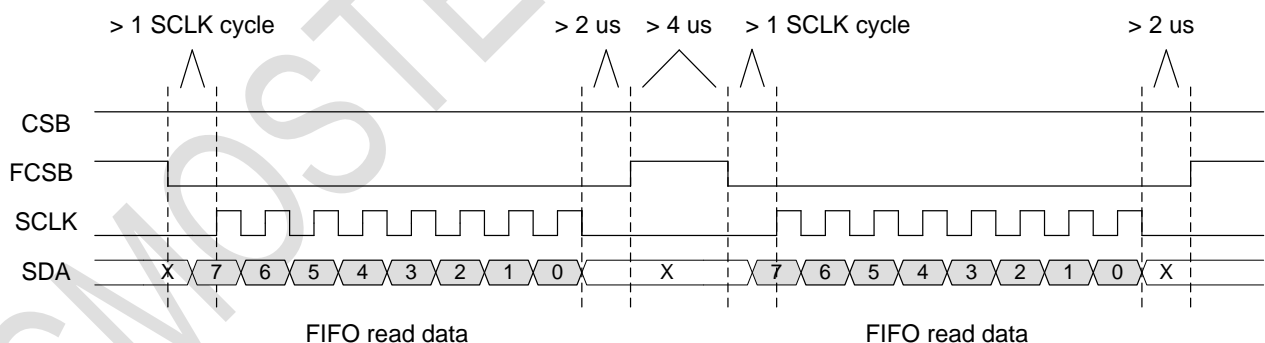


Figure 4-7. SPI Read FIFO Timing

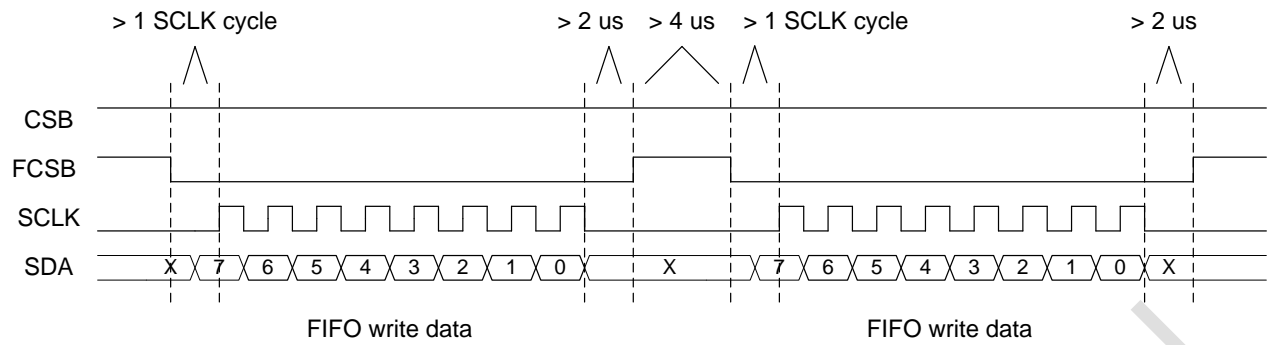


Figure 4-8. SPI Write FIFO Timing

The transceiver provides enriched FIFO-related interrupt sources helping for efficient operation of the chip. The Rx and Tx-related FIFO interrupt timing is shown in the figure below.

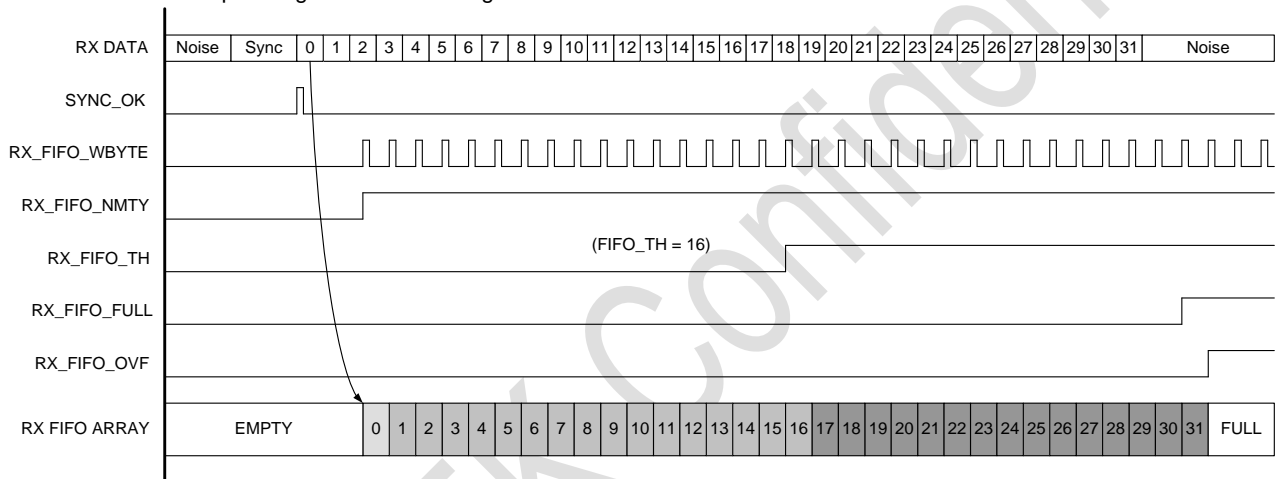


Figure 4-9. Transceiver RX FIFO Interrupt Timing Schematic

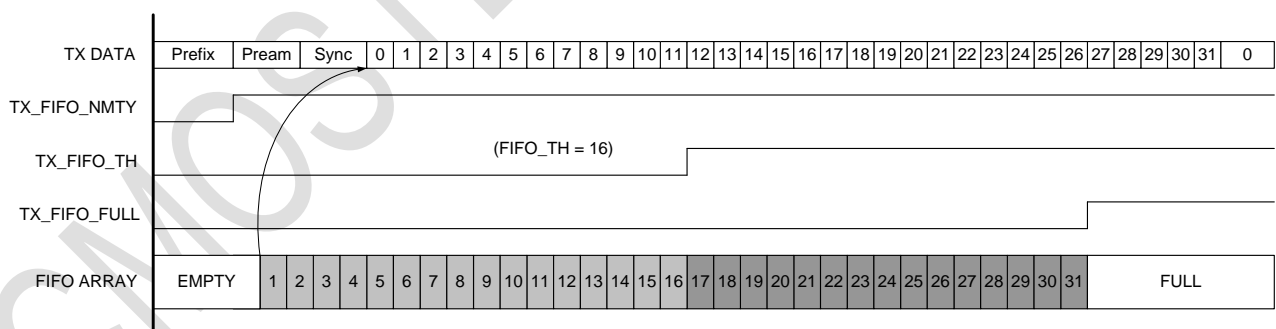


Figure 4-10. Transceiver TX FIFO Interrupt Timing Schematic

4.11.3 Transceiver Operating Status, Timing, and Power Consumption

● Startup Timing

After the transceiver is powered up on RF-VDD, generally it takes about 1 ms for POR release. The crystal starts after POR release. The startup time defaults to N ms depending on characteristics of the crystal itself. After startup, it needs to wait for a period of time for crystal stabilization then to start working. The default stabilization time is set as 2.48ms by default. The time can

be written later by writing XTAL_STB_TIME <2:0>. The chip stays in the IDLE state until the crystal is stable. After the crystal is stabilized, the chip will leave IDLE and begin calibration for each module. After calibration, the chip will stay at SLEEP, waiting for the user to do initialization configuration. The chip returns to IDLE and resume the power-on process any time when reset is performed.

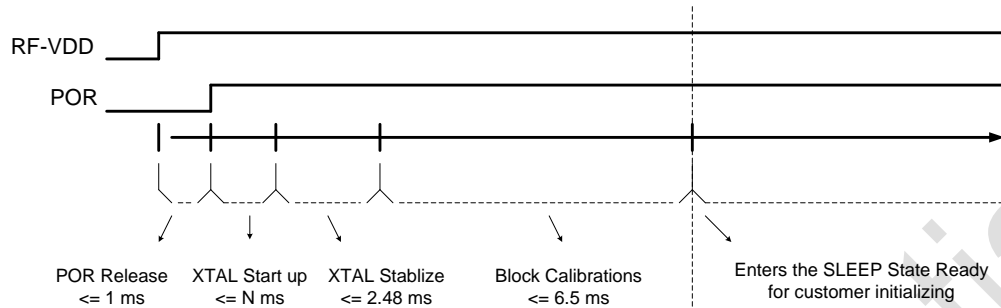


Figure 4-11.Startup Timing Diagram

The chip enters SLEEP mode after calibration completion. By now, the controller can switch the chip to different operating states by setting the register CHIP_MODE_SWT<7:0>.

● Transceiver Operating Status

The transceiver has a total of 7 working status: IDLE, SLEEP, STBY, RFS, RX, TFS and TX, as listed in the below table.

Table 4-1. Transceiver Status and Module Startup Table

Status	Binary Code	Switch Command	Startup Module	Optional Startup Module
IDLE	0000	soft_rst	SPI, POR	NA
SLEEP	0001	go_sleep	SPI, POR, FIFO	LFOSC, Sleep Timer
STBY	0010	go_stby	SPI, POR, XTAL, FIFO	CLKO
RFS	0011	go_rfs	SPI, POR, XTAL, PLL, FIFO	CLKO
TFS	0100	go_tfs	SPI, POR, XTAL, PLL, FIFO	CLKO
RX	0101	go_rx	SPI, POR, XTAL, PLL, LNA+MIXER+IF, FIFO	CLKO, RX Timer
TX	0110	go_tx	SPI, POR, XTAL, PLL, PA, FIFO	CLKO

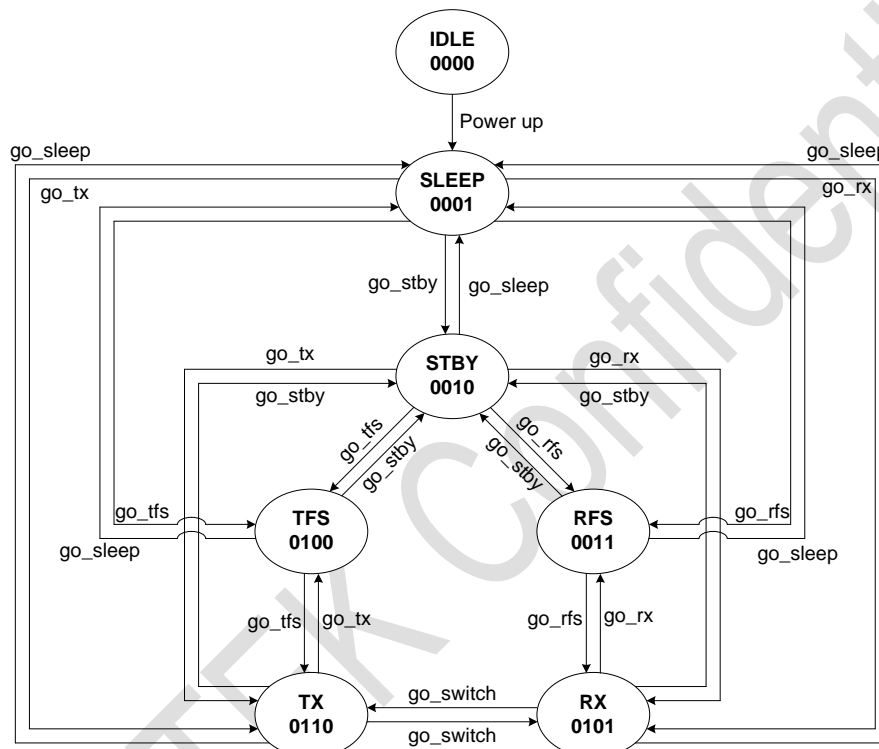


Figure 4-12. Status Switch Diagram

- **SLEEP Status**

The power consumption of the chip keeps the lowest in SLEEP mode with almost all modules disabled. The SPI is enabled, the registers in the configuration area and control area 1 can be accessed, and the contents previously filled in the FIFO is retained but the FIFO cannot be operated. If the periodical wake-up function is enabled, the LFOSC and sleep counters will start to work. The time period required for switching from IDLE to SLEEP is the power up processing time specified in above. Switching from the rest of the status to SLEEP is done immediately.

- **STBY Status**

In STBY, the crystal is enabled and the LDO of the digital circuit is enabled as well. The current is slightly increased. The FIFO can be operated. Users can choose whether to output CLKO (system clock) to the GPIO pin. As the crystal is enabled, the time required switching from STBY to transmit or receive is less than that in SLEEP status (switching from SLEEP to STBY requires more time to wait for the crystal startup and stabilizing for a while). Switching from other status to STBY will complete immediately.

- **RFS Status**

RFS is transitional status before switching to RX. Except the receiver's RF module, all others are enabled and the current will

be larger than that in STBY. As the PLL is already locked to the RX frequency point in RFS status, it cannot be switched to TX. Switching from STBY to RFS requires approximately 350us for PLL calibration and settling. Switching from SLEEP to RFS requires more time to wait for the crystal startup and stabilizing for a while. Switching from other status to RFS is done immediately.

- **TFS Status**

TFS is transitional status before switching to TX. Except the transmitter's RF module, all other modules are enabled and the current will be larger than STBY. As the PLL is already locked to the frequency of the TX in TFS status, it cannot be switched to RX. Switching from STBY to TFS requires approximately 350us for PLL calibration and settling. Switching from SLEEP to TFS requires more time to wait for the crystal startup and stabilizing for a while. Switching from other states to TFS is done immediately.

- **RX Status**

All RX related modules on the receiver will be turned on. Switching from RFS to RX takes only 20us. Switching from STBY to RX requires a 350us for PLL calibration and settling. Switching from SLEEP to RX requires more time to wait for the crystal startup and stabilizing for a while. In TX, users can switch to RX in a prompt way by sending the go_switch command. No matter whether the frequency points set by TX and RX are the same, it needs to wait for 350us for PLL recalibration and settling to accomplish switch.

- **TX Status**

All transmitter related modules will be turned on at TX. Switching from TFS to TX takes only 20us. Switching from STBY to TX requires 350us for PLL calibration and settling. Switching from SLEEP to TX requires more time to wait for the crystal startup and stabilizing for a while. In RX, users can switch to TX in a prompt way by sending the go_switch command. No matter whether the frequency points set by TX and RX are the same, it needs to wait for 350us for PLL recalibration and settling to complete switch.

4.11.4 Transceiver GPIO Function and Interrupt Mapping

The three GPIOs employed in the transceiver are GPIO1, GPIO2, and GPIO3. Each GPIO can be configured with different input or output functions. The transceiver is employed with two interrupt ports, which can be configured to different GPIOs for mapping output.

Table 4-2. Transceiver GPIO Function

Pin#	Name	I/O	Function
48	GPIO1	IO	Can configure as:DOUT/DIN,INT1,INT2,DCLK (TX/RX),RF_SWT
47	GPIO2	IO	Can configure as:INT1,INT2,DOUT/DIN,DCLK (TX/RX),RF_SWT
9	GPIO3	IO	Can configure as:CLKO,DOUT/DIN,INT2,DCLK (TX/RX)

The interrupt mapping table is given below. Mapping for INT1 and INT2 is the same. Take INT1 as an example as follows.

Table 4-3. Transceiver Interrupt Mapping

Name	INT1_SEL	Description	Clear Method
RX_ACTIVE	00000	Indicates that the interrupt is ready to enter RX or has entered RX, which is 1 in the PLL calibration and RX status, otherwise 0.	Auto
TX_ACTIVE	00001	Indicates that the interrupt is ready to enter TX or has entered TX, which is 1 in the PLL calibration and RX status, otherwise 0.	Auto
RSSI_VLD	00010	An interrupt Indicating whether the RSSI is valid	Auto
PREAM_OK	00011	An interrupt Indicating whether Preamble is received successfully.	by MCU
SYNC_OK	00100	An interrupt Indicating whether Sync Word is received successfully.	by MCU
NODE_OK	00101	An interrupt Indicating whether Node ID is received successfully.	by MCU
CRC_OK	00110	An interrupt Indicating whether receive and pass CRC check successfully	by MCU
PKT_OK	00111	An interrupt indicating receive a packet completely	by MCU
SL_TMO	01000	An interrupt indicating SLEEP counter time out	by MCU
RX_TMO	01001	An interrupt indicating RX counter time out	by MCU
TX_DONE	01010	An interrupt indicating TX completion	by MCU
RX_FIFO_NMTY	01011	An interrupt indicating RX counter is not empty	Auto
RX_FIFO_TH	01100	An interrupt indicating unread contents in RX FIFO exceed FIFO TH	Auto
RX_FIFO_FULL	01101	An interrupt indicating RX FIFO is full	Auto
RX_FIFO_WBYTE	01110	An interrupt indicating a BYTE is written in RX FIFO, which is a pulse	Auto
RX_FIFO_OVF	01111	An interrupt indicating RX FIFO overflow	Auto
TX_FIFO_NMTY	10000	An interrupt indicating TX FIFO is not empty	Auto
TX_FIFO_TH	10001	An interrupt indicating unread contents in TX FIFO exceed FIFO TH	Auto
TX_FIFO_FULL	10010	An interrupt indicating TX FIFO is full	Auto
STATE_IS_STBY	10011	An interrupt indicating current status is STBY	Auto
STATE_IS_FS	10100	An interrupt indicating current status is RFS or TFS	Auto
STATE_IS_RX	10101	An interrupt indicating current status is RX	Auto
STATE_IS_TX	10110	An interrupt indicating current status TX	Auto
LBD	10111	An interrupt indicating that low battery detection is active (VDD is lower than the set TH)	Auto
TRX_ACTIVE	11000	An interrupt indicating begin to enter RX or TX, already in RX or TX, which is 1 in PLL calibration, RX status or TX state, otherwise 0.	Auto
PKT_DONE	11001	Indicating that the current data packet has been received with 4 cases as below: 1. Receive the entire packet completely 2. Manchester decoding error, decoding circuit automatically restarts 3. NODE ID receives error, the decoding circuit automatically restarts 4. The signal conflict is found, the decoding circuit does not restart automatically, waiting for the MCU to process	by MCU

The interrupt is valid for value 1 by default. However value 0 becomes valid for all interrupts by setting the INT_POLAR register bit to 1. Still take INT1 as an example as shown in the below figure with the control and selection diagram of all interrupt sources shown. INT1 and INT2 are the same in control and mapping.

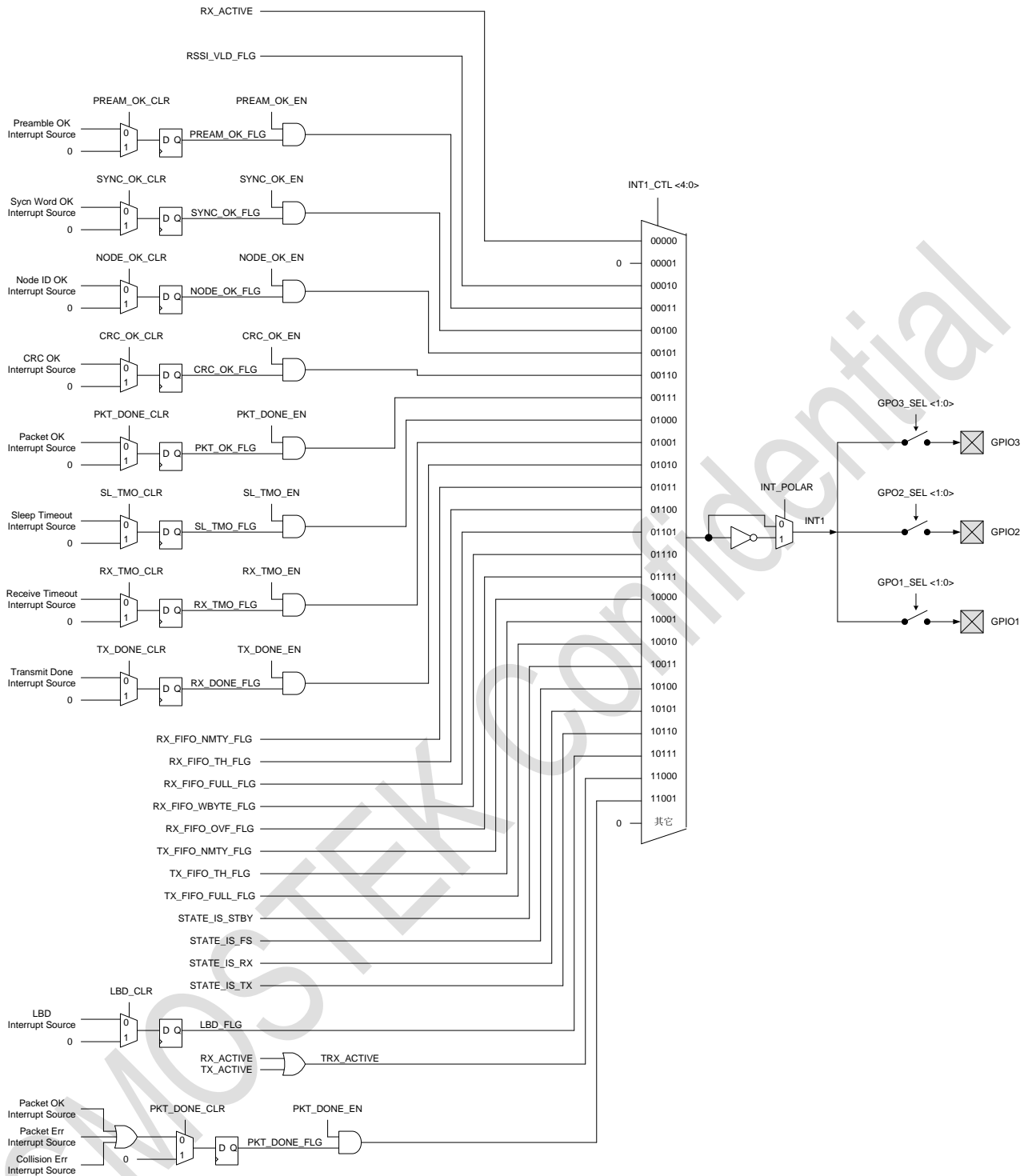


Figure 4-13. CMT2300A INT1 Interrupt Map

5 80C51 CPU Function Description

5.1 CPU Register

- **PSW:** Program Status Word
SFR Page = 0~F
SFR Address = 0xD0

Bit	7	6	5	4	3	2	1	0
Name	CY	AC	F0	RS1	RS0	OV	F1	P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

- CY: Carry bit.
- AC: Auxiliary carry bit.
- F0: General purpose flag 0.
- RS1: Register bank select bit 1.
- RS0: Register bank select bit 0.
- OV: Overflow flag.
- F1: General purpose flag 1.
- P: Parity bit.

The program status word (PSW) contains several status bits that reflect the current state of the CPU. The PSW, shown above, resides in the SFR space. It contains the Carry bit, the Auxiliary Carry(for BCD operation), the two register bank select bits, the Overflow flag, a Parity bit and two user-definable status flags.

The Carry bit, other than serving the function of a Carry bit in arithmetic operations, also serves as the "Accumulator" for a number of Boolean operations.

The bits RS0 and RS1 are used to select one of the four register banks shown in Section "6.2 On-Chip Data RAM". A number of instructions refer to these RAM locations as R0 through R7.

The Parity bit reflects the number of 1s in the Accumulator. P=1 if the Accumulator contains an odd number of 1s and otherwise P=0.

- **SP: Stack Pointer**
SFR Page = 0~F
SFR Address = 0x81

Bit	7	6	5	4	3	2	1	0
Name	SP[7:0]							
R/W	R/W							
Reset Value	0	0	0	0	0	1	1	1

The Stack Pointer holds the location of the top of the stack. The stack pointer is incremented before every PUSH operation. The SP register defaults to 0x07 after reset.

- **DPL: Data Pointer Low**
SFR Page = 0~F
SFR Address = 0x82

Bit	7	6	5	4	3	2	1	0
Name	DPTR[7:0]							
R/W	R/W							
Reset Value	0	0	0	0	0	0	0	0

The DPL register is the low byte of the 16-bit DPTR. DPTR is used to access indirectly addressed XRAM and Flash memory.

- **DPH: Data Pointer High**
SFR Page = 0~F
SFR Address = 0x83

Bit	7	6	5	4	3	2	1	0
Name	DPTR[15:8]							
R/W	R/W							
Reset Value	0	0	0	0	0	0	0	0

The DPH register is the high byte of the 16-bit DPTR. DPTR is used to access indirectly addressed XRAM and Flash memory.

- **ACC: Accumulator**
SFR Page = 0~F
SFR Address = 0xF0

Bit	7	6	5	4	3	2	1	0
Name	ACC[7:0]							
R/W	R/W							
Reset Value	0	0	0	0	0	0	0	0

This register is the accumulator for arithmetic operations.

- **B: B Register**
SFR Page = 0~F
SFR Address = 0xF0

Bit	7	6	5	4	3	2	1	0
Name	B[7:0]							
R/W	R/W							
Reset Value	0	0	0	0	0	0	0	0

This register serves as a second accumulator for certain arithmetic operations.

5.2 CPU Timing

CMT2380F17 is a single-chip microcontroller based on a high performance 1-T architecture 80C51 CPU that has an 8051 compatible instruction set, and executes instructions in 1~7 clock cycles (about 6~7 times

the rate of a standard 8051 device). It employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. The instruction timing is different than that of the standard 8051.

In many 8051 implementations, a distinction is made between machine cycles and clock cycles, with machine cycles varying from 2 to 12 clock cycles in length. However, the 1T-80C51 implementation is based solely on clock cycle timing. All instruction timings are specified in terms of clock cycles. For more detailed information about the 1T-80C51 instructions, please refer section “34 Instruction Set” which includes the mnemonic, number of bytes, and number of clock cycles for each instruction.

5.3 CPU Addressing Mode

5.3.1 Direct Addressing (DIR)

In direct addressing the operand is specified by an 8-bit address field in the instruction. Only internal data RAM and SFRs can be direct addressed.

5.3.2 Indirect Addressing (IND)

In indirect addressing the instruction specified a register which contains the address of the operand. Both internal and external RAM can be indirectly addressed. The address register for 8-bit addresses can be R0 or R1 of the selected bank, or the Stack Pointer. The address register for 16-bit addresses can only be the 16-bit data pointer register – DPTR.

5.3.3 Register Instruction (REG)

The register banks, containing registers R0 through R7, can be accessed by certain instructions which carry a 3-bit register specification within the op-code of the instruction. Instructions that access the registers this way are code efficient because this mode eliminates the need of an extra address byte. When such instruction is executed, one of the eight registers in the selected bank is accessed.

5.3.4 Register-Specific Instruction

Some instructions are specific to a certain register. For example, some instructions always operate on the accumulator or data pointer, etc. No address byte is needed for such instructions. The op-code itself does it.

5.3.5 Immediate Constant (IMM)

The value of a constant can follow the op-code in the program memory.

5.3.6 Index Addressing

Only program memory can be accessed with indexed addressing and it can only be read. This addressing mode is intended for reading look-up tables in program memory. A 16-bit base register (either DPTR or PC) points to the base of the table, and the accumulator is set up with the table entry number. Another type of indexed addressing is used in the conditional jump instruction.

In conditional jump, the destination address is computed as the sum of the base pointer and the accumulator.

6 Memory Organization

Like all 80C51 devices, the CMT2380F17 has separate address spaces for program and data memory. The logical separation of program and data memory allows the data memory to be accessed by 8-bit addresses, which can be quickly stored and manipulated by the 8-bit CPU.

Program memory (ROM) can only be read, not written to. There can be up to 16K bytes of program memory. In the CMT2380F17, all the program memory are on-chip Flash memory, and without the capability of accessing external program memory because of no External Access Enable (/EA) and Program Store Enable (/PSEN) signals designed.

Data memory occupies a separate address space from program memory. In the CMT2380F17, there are 256 bytes of internal scratch-pad RAM and 768 bytes of on-chip expanded RAM (XRAM).

6.1 On-Chip Program Flash

Program memory is the memory which stores the program codes for the CPU to execute, as shown in Figure 6–1. After reset, the CPU begins execution from location 0000H, where should be the starting of the user's application code. To service the interrupts, the interrupt service locations (called interrupt vectors) should be located in the program memory. Each interrupt is assigned a fixed location in the program memory. The interrupt causes the CPU to jump to that location, where it commences execution of the service routine. External Interrupt 0, for example, is assigned to location 0003H. If External Interrupt 0 is going to be used, its service routine must begin at location 0003H. If the interrupt is not going to be used, its service location is available as general purpose program memory.

The interrupt service locations are spaced at an interval of 8 bytes: 0003H for External Interrupt 0, 000BH for Timer 0, 0013H for External Interrupt 1, 001BH for Timer 1, etc. If an interrupt service routine is short enough (as is often the case in control applications), it can reside entirely within that 8-byte interval. Longer service routines can use a jump instruction to skip over subsequent interrupt locations, if other interrupts are in use.

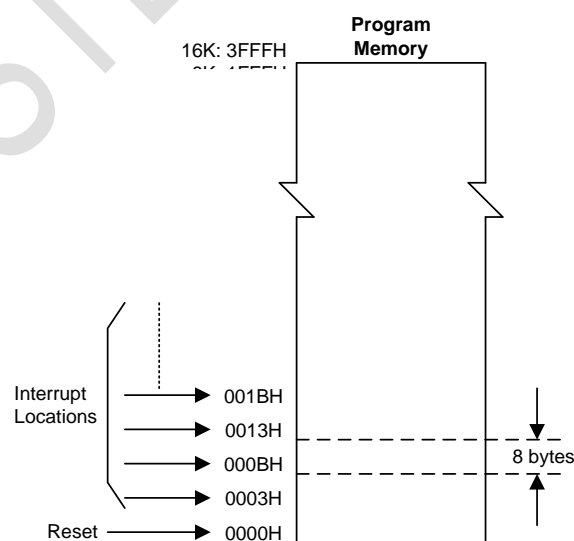


Figure 6-1. Program Memory

6.2 On-Chip Data RAM

Figure 6–2 shows the internal and external data memory spaces available to the CMT2380F17 user. Internal data memory can be divided into three blocks, which are generally referred to as the lower 128 bytes of RAM, the upper 128 bytes of RAM, and the 128 bytes of SFR space. Internal data memory addresses are always 8-bit wide, which implies an address space of only 256 bytes. Direct addresses higher than 7FH access the SFR space; and indirect addresses higher than 7FH access the upper 128 bytes of RAM. Thus the SFR space and the upper 128 bytes of RAM occupy the same block of addresses, 80H through FFH, although they are physically separate entities.

The lower 128 bytes of RAM are present in all 80C51 devices as mapped in Figure 6–3. The lowest 32 bytes are grouped into 4 banks of 8 registers. Program instructions call out these registers as R0 through R7. Two bits in the Program Status Word (PSW) select which register bank is in use. This allows more efficient use of code space, since register instructions are shorter than instructions that use direct addressing. The next 16 bytes above the register banks form a block of bit-addressable memory space. The 80C51 instruction set includes a wide selection of single-bit instructions, and the 128 bits in this area can be directly addressed by these instructions. The bit addresses in this area are 00H through 7FH.

All of the bytes in the Lower 128 can be accessed by either direct or indirect addressing while the Upper 128 can only be accessed by indirect addressing.

Figure 6–4 gives a brief look at the Special Function Register (SFR) space. SFRs include the Port latches, timers, peripheral controls, etc. These registers can only be accessed by direct addressing. Sixteen addresses in SFR space are both byte- and bit-addressable. The bit-addressable SFRs are those whose address ends in 0H or 8H.

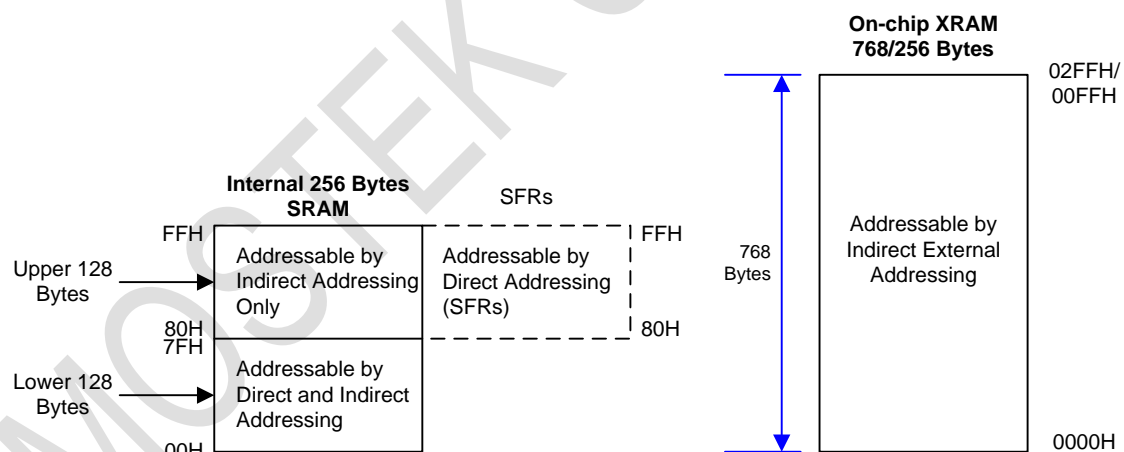


Figure 6-2. Data Memory

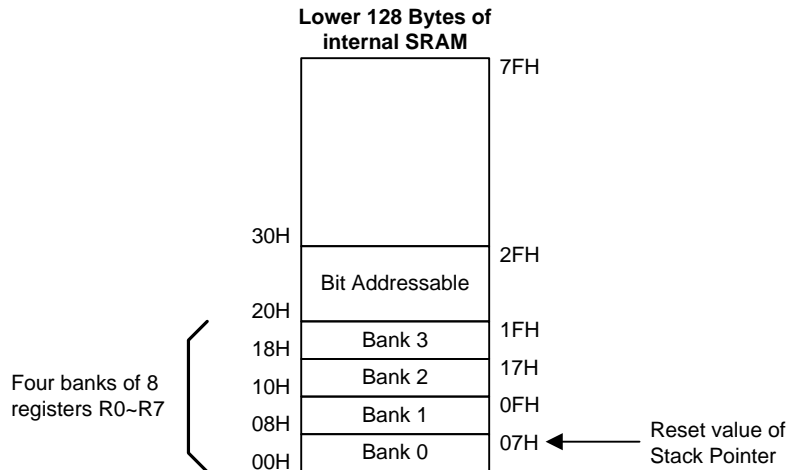


Figure 6-3. Lower 128 Bytes of Internal RAM

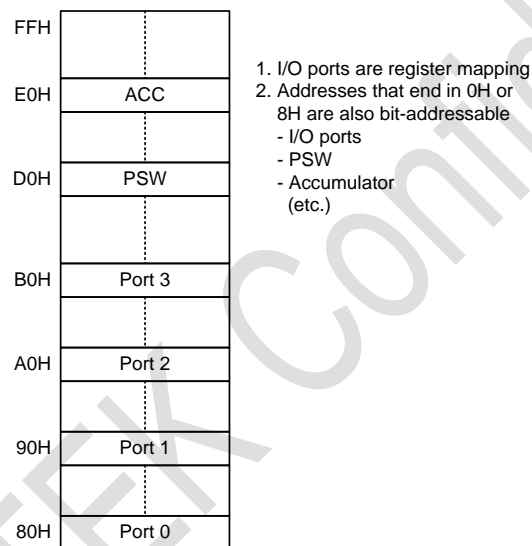


Figure 6-4. SFR Space

6.3 On-chip Expanded RAM (XRAM)

To access the on-chip expanded RAM (XRAM), refer to Figure 6-2, the 768 bytes of XRAM (0000H to 02FFH) are indirectly accessed by move external instruction, "MOVX @Ri" and "MOVX @DPTR". For C51 compiler, to assign the variables to be located at XRAM, the "pdata" or "xdata" definition should be used. After being compiled, the variables declared by "pdata" and "xdata" will become the memories accessed by "MOVX @Ri" and "MOVX @DPTR", respectively. Thus the CMT2380F17 hardware can access them correctly.

6.4 Off-Chip External Data Memory access

The off-chip external data memory access function is not supported in CMT2380F17.

6.5 Declaration Identifiers in a C51-Compiler

The declaration identifiers in a C51-compiler for the various CMT2380F17 memory spaces are as follows:

data

128 bytes of internal data memory space (00h~7Fh); accessed via direct or indirect addressing, using

instructions other than MOVX and MOVC. All or part of the Stack may be in this area.

idata

Indirect data; 256 bytes of internal data memory space (00h~FFh) accessed via indirect addressing using instructions other than MOVX and MOVC. All or part of the Stack may be in this area. This area includes the data area and the 128 bytes immediately above it.

sfr

Special Function Registers; CPU registers and peripheral control/status registers, accessible only via direct addressing.

xdata

External data or on-chip eXpanded RAM (XRAM); duplicates the classic 80C51 64KB memory space addressed via the "MOVX @DPTR" instruction. The CMT2380F17 has 768 bytes of on-chip xdata memory.

pdata

Paged (256 bytes) external data or on-chip eXpanded RAM; duplicates the classic 80C51 256 bytes memory space addressed via the "MOVX @Ri" instruction. The CMT2380F17 has 256 bytes of on-chip pdata memory which is shared with on-chip xdata memory.

code

16K bytes of program memory space; accessed as part of program execution and via the "MOVC @A+DTPR" instruction. The CMT2380F17 has 16K bytes of on-chip code memory.

7 XRAM Access

The CMT2380F17 MCUs include 768 bytes of on-chip RAM mapped into the external data memory space (XRAM). The external memory space may be accessed using the external move instruction (MOVX) and the data pointer (DPTR), or using the MOVX indirect addressing mode using R0 or R1. If the MOVX instruction is used with an 8-bit address operand (such as @R1), then the high byte of the 16-bit address is provided by the XRAM Page Select Register (XRPS).

The internal XRAM memory space is accessed using the MOVX instruction. The MOVX instruction has two forms, both of which use an indirect addressing method. The first method uses the Data Pointer, DPTR, a 16-bit register which contains the effective address of the XRAM location to be read or written. The second method uses R0 or R1 in combination with the XRPS register to generate the effective XRAM address.

7.1 MOVX on 16-bit Address with dual DPTR

The dual DPTR structure as shown in Figure 7–1 is a way by which the chip can specify the address of an external data memory location. There are two 16-bit DPTR registers that address the external memory, and a single bit called DPS (AUXR1.0) that allows the program code to switch between them.

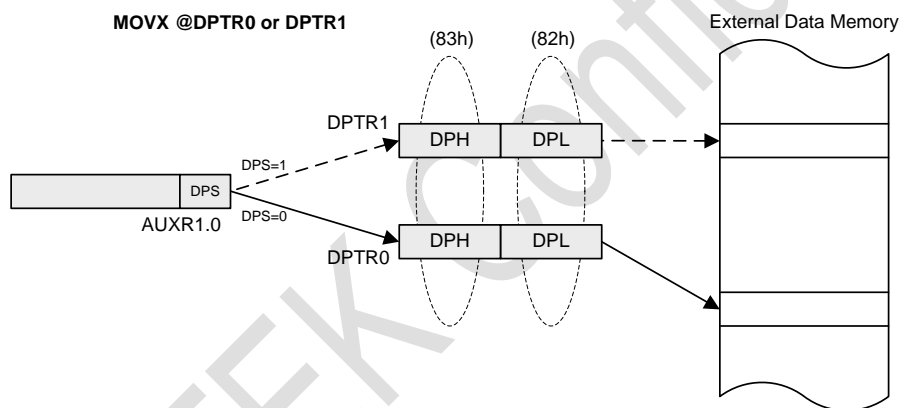


Figure7–1.DualDPTRStructure

DPTR Instructions

The six instructions that refer to DPTR currently selected using the DPS bit are as follows:

```

INC  DPTR           ;Increments the datapointer by 1
MOV  DPTR,#data16   ;Load the DPTR with a 16-bit constant
MOV  A,@A+DPTR      ;Move code byte relative to DPTR to ACC
MOVX A,@DPTR        ;Move external RAM (16-bit address) to ACC
MOVX @DPTR,A        ;Move ACC to external RAM (16-bit address)
JMP @A+DPTR         ;Jump indirect relative to DPTR
  
```

- **AUXR1:** Auxiliary Control Register 1
SFR Page = 0~F
SFR Address = 0xA2

Bit	7	6	5	4	3	2	1	0
Name	--	--	CRCDS1	CRCDS0	--	--	--	DPS
R/W	W	W	R/W	R/W	W	W	W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 0: DPS, DPTR select bit. Use to switch between DPTR0 and DPTR1.
 0: Select DPTR0.
 1: Select DPTR1.

DPS	Selected DPTR
0	DPTR0
1	DPTR1

- **DPL: Data Pointer Low**
 SFR Page = 0~F
 SFR Address = 0x82

Bit	7	6	5	4	3	2	1	0
Name	DPTR[7:0]							
R/W	R/W							
Reset Value	0	0	0	0	0	0	0	0

The DPL register is the low byte of the 16-bit DPTR. DPTR is used to access indirectly addressed XRAM and Flash memory.

- **DPH: Data Pointer High**
 SFR Page = 0~F
 SFR Address = 0x83

Bit	7	6	5	4	3	2	1	0
Name	DPTR[15:8]							
R/W	R/W							
Reset Value	0	0	0	0	0	0	0	0

The DPH register is the high byte of the 16-bit DPTR. DPTR is used to access indirectly addressed XRAM and Flash memory.

7.2 MOVX on 8-bit Address with XRPS

The 8-bit form of the MOVX instruction uses the contents of the XRPS SFR to determine the upper 8-bits of the effective address to be accessed and the contents of R0 or R1 to determine the lower 8-bits of the effective address to be accessed.

MOVX @Ri with XRPS on XRAM

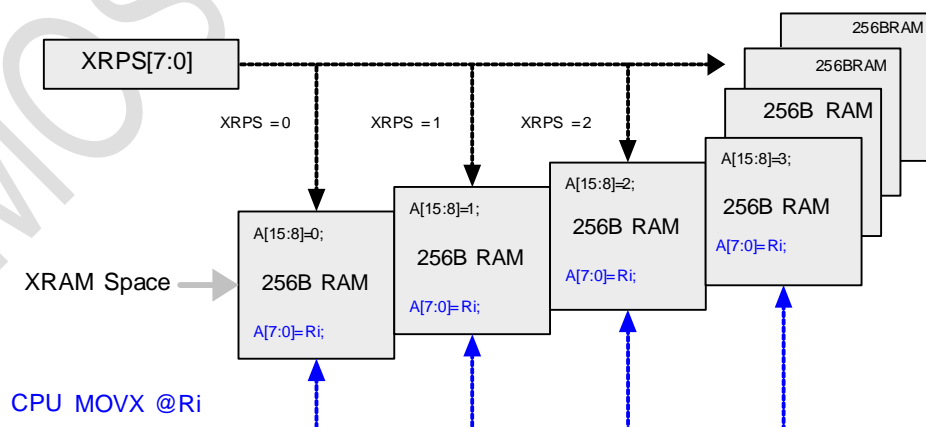


Figure 7-2. XRPS Structure

- **XRPS: XRAM Page Select Register**
 SFR Page = 0~F

SFR Address = 0x8F

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	--	--	--	XRPS.1	XRPS.0
R/W	W	W	W	W	W	W	R/W	R/W
Reset Value	X	X	X	X	X	X	0	0

Bit 7~2: Reserved. Software must write "0" on these bits when XRPS is written.

Bit 1~0: XRPS, XRAM Page Select. The XRPS register provides the high byte of the 16-bit external data memory address when using an 8-bit MOVX command, effectively selecting a 256-byte page of RAM. Since the upper (reserved) bits of the register are always zero, the XRPS determines which page of XRAM is accessed. In CMT2380F17, XRPS indexes the three pages 256-byte RAM.

For Example: If XRPS = 0x01, addresses 0x0100 through 0x01FF in XRAM will be accessed.

8 Direct Memory Access Controller (DMA)

The direct memory access (DMA) controller transfers data from data source to data destination, without CPU intervention, across the entire XRAM address range and the entire SFR address range. For example, the DMA controller can move data from the ADC12 conversion result register to 8051 XRAM. This keeps CPU resources free for other operations.

Using the DMA controller can increase the throughput of peripheral modules. It can also reduce system power consumption by allowing the CPU to remain in a low-power mode without having to awaken to move data to or from a peripheral.

The DMA controller features include: Easy use one channel DMA:

- Easy use one channel DMA
- Transfer type: Memory to peripheral (M2P), peripheral to memory (P2M), peripheral to peripheral (P2P)
- Configurable transfer trigger selections: CPU software or external hardware
- Support block transfer mode, transfer sizes up to 65536 transactions
- Capability to copy data to CRC engine during DMA transfer
- Auto-initialization for circular buffer management (loop mode)
- Capability to suspend and resume a DMA transfer.
- Capability to operate in low power modes (idle mode for interrupt)
- Option interrupt on End of DMA transfer

The DMA access diagram is shown in Figure 8–1.

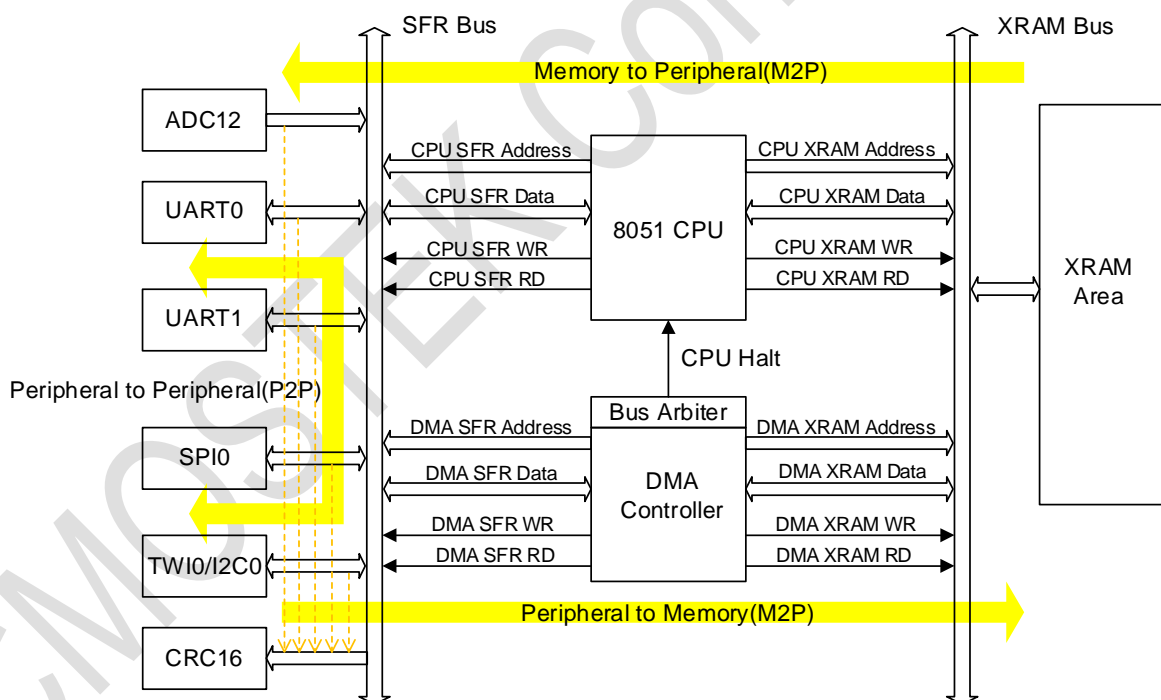


Figure 8-1. DMA DMA Access Diagram

8.1 DMA Structure

In CMT2380F17, the DMA controller provides one channel DMA to support 3 transfer types: transfer the data from XRAM to peripheral, from peripheral to XRAM and from peripheral to peripheral. DMADS0 register in DMA channel 0 defines the DMA transfer type to configure DMA controller behavior and defines the data path to generate the SFR address on peripheral access.

Timer 5 and Timer 6 are embedded in DMA module. The DMA controller supports the block mode

transfer by one DMA trigger, on CPU software trigger or external hardware trigger. The transfer size is programmable from 1 to 65536 and this function is implemented on Timer 5 for DMA transfer count. If DMA needs to access XRAM, the Timer 6 implements the XRAM address pointer. When DMA finishes one data transaction, DMA_CLK will trigger Timer 5 to increase the DMA transfer count and increase Timer 6 to point next XRAM address. Both of Timer 5 and Timer 6 only support up-count operation. When DMA function is not in used, the Timer 5 and Timer 6 can be traded as a general Timer 0 with 16-bit counter.

DMACR0 and DMACG0 are the SFRs for DMA operation mode control. It includes DMA start, suspend, interrupt enabled....etc. In DMA Operation section, will introduce the function in detailed.

The DMA controller block diagram is shown in Figure 8-2.

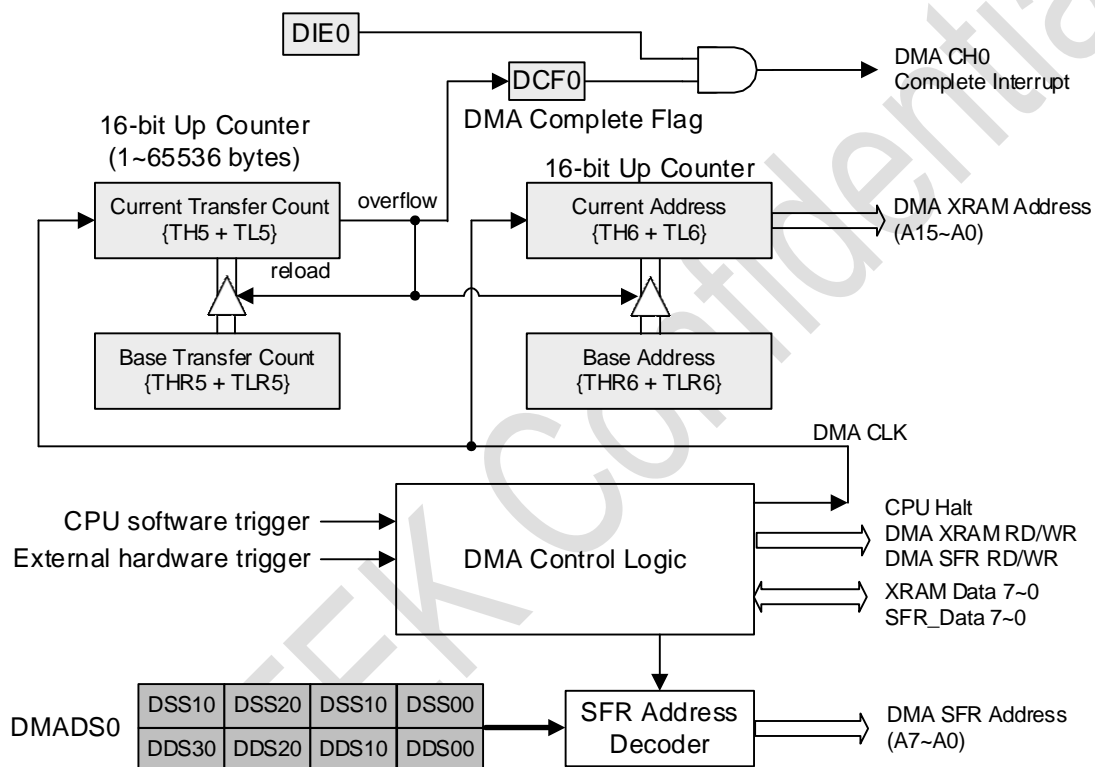


Figure 8-2. DMA Structure

8.2 DMA Operation

The DMA controller is configured with user software. The setup and operation of the DMA is discussed in the following sections.

8.2.1 DMA Transfer Types

The DMA controller in CMT2380F17 supports 3 type data transfer as following list::

- M2P: XRAM to Peripheral
- P2M: Peripheral to XRAM
- P2P: Peripheral to Peripheral

The DMA controller does not support the data transfer for XRAM to XRAM (M2M). It also cannot access the internal data RAM area and flash ROM area. The DMA transfer type is defined by DMADS0 when software configures the DMA data path of source and destination. Otherwise, if a peripheral is configured for DMA access, software must not access the data register of the peripheral.

For example 1:

The source data is selected on ADC12 and the destination data is selected on XRAM. Then, DMA controller will move data from the ADC12 conversion result register ADCDH and ADCDL to 8051 XRAM. The DMA transfer type is peripheral to XRAM. And software must avoid the read operation on ADCDH and ADCDL.

For example 2:

The source data is selected on ADC12 and the destination data is selected on S0 TX. Then, DMA controller will move data from the ADC12 conversion result register ADCDH and ADCDL to S0BUF. The DMA transfer type is peripheral to peripheral. Software must avoid to read ADCDH, ADCDL and to write data to S0BUF.

The configuration of DMA data path selection is listed in below Table 8–1.

Table8-1. DMA Data Path Selection

DSS30~00 (DMADS0[7:4])	Source Selection	DDS30~00 (DMADS0[3:0])	Destination Selection
0 0 0 0	Disabled	0 0 0 0	Disabled
0 0 0 1	S0RX	0 0 0 1	S0TX
0 0 1 0	S1RX	0 0 1 0	S1TX
0 1 0 1	TWI0RX	0 1 0 1	TWI0TX
0 1 1 1	SPI0RX	0 1 1 1	SPI0TX
1 0 0 1	ADC0	1 0 0 1	保留
1 1 0 1	保留	1 1 0 1	CRC
1 1 1 1	XRAM	1 1 1 1	XRAM

8.2.2 DMA Transfer Mode

The DMA controller in CMT2380F17 only supports block transfer mode. After DMA trigger active, DMA controller start to move data until the overflow event happened on DMA Current Transfer Count. That is one trigger input to activate a block data transfer by DMA controller.

The block data transfer size is defined in {TH5+TL5} as DMA Current Transfer Count. It supports the transfer size from 1 DMA transaction to 65536 DMA transactions. In CMT2380F17, one DMA transaction move one byte data from source to destination..

8.2.3 Transfer Count & Address Pointer

DMA transfer count control and memory address pointer are implemented on Timer 5 and Timer 6 in DMA module. The action of Timer 5 and Timer 6 likes general Timer 0 with 16-bit counter (TH5 + TL5, TH6 + TL6) and 16-bit reload register (THR5 + TLR5, THR6 + TLR6). If DMA enabled, Timer 5 controls the DMA transfer count and Timer 6 points to memory address. Both of Timer 5 and Timer 6 is always 16-bit up-count counter.

The Current Transfer Count implemented on {TH5 + TL5} register determines the number of transactions to be performed. The Base Transfer Count is implemented on {THR5 + TLR5}. It supports the maximum transfer count is up to 65536. The actual transfer count is equal to the value of (65536 – {TH5 + TL5}). The Current Transfer Count is increment after each DMA transaction. When the value in the register goes from FFFFH to 0000H, an event at “End of DMA transfer” is generated to stop the DMA transfer by clear DMAS0 and set DMA Complete Flag (DCF0). The event also reloads {THR5 + TLR5} to {TH5 + TL5} to initialize the new Current Transfer Count for next DMA transfer.

For examples on transfer count initial,

- A. If DMA transfer size is 65536, the {TH5 + TL5} will be programmed to 0000H.
- B. If DMA transfer size is 1, the {TH5 + TL5} will be written by FFFFH.

The Current Address implemented on {TH6 + TL6} register points the memory address for DMA access on XRAM. Based on {TH6 + TL6} up counting function, the addresses generated will be increased. There is a Base Address located on {THR6 + TLR6}. Each event on “End of DMA transfer” will reload the {THR6+ TLR6} to {TH6 + TL6} to initialize the new Current Address for next DMA transfer. The Current Address covers the entire XRAM memory space.

8.2.4 Start a DMA Transfer

It is an easy handling DMA controller in CMT2380F17. To starting a DMA transfer, software must issue the following sequence to construct a DMA operation:

1. Configure DMADS0 to determine the DMA transfer type and DMA data path on source and destination.
2. Configure DMA interrupt and its interrupt priority.
3. Configure the Current Transfer Count and Base Transfer Count
4. Configure the Current Address and Base Address if XRAM accessed by DMA is necessary
5. Configure the peripheral to ready state
6. Set DMAE0 to enable DMA FSM
7. Configure DMA trigger source and trigger DMA to start operation
 - If select software trigger, software sets DMAS0 to start DMA
 - If select external trigger, wait external active signal to start DMA
8. Software waits DMA Complete Flag (DCF0) that indicates the DMA transfer finished
9. Write 0 on DMAE0 to end DMA operation and configure DMADS0 to disable state.

In DMA external trigger operation, the external active signal will set DMAS0 automatically. Both of internal and external trigger, the DMAS0 will be cleared automatically when DMA transfer is finished, End of DMA transfer.

8.2.5 Suspend or Stop DMA Transfer

A DMA transaction can be suspended during the transfer (after DMAS0 set) by writing 0 on DMAS0. If the channel is suspended when a DMA data transaction is ongoing, the channel is effectively disabled only once the current data transaction is completed. Re-enabling the DMAS0 resumes the DMA transfer.

Software can write “0” on DMAE0 to stop current DMA transfer at any time or to end the DMA transfer after End of DMA transfer. It is recommended software must also configure data path (DMADS0) to disable state and clear DMAS0.

8.2.6 DMA Interrupt

DCF0 is set in any transfer mode or transfer type, when the corresponding Current Transfer Count register {TH5 + TL5} counts to overflow. If the corresponding DIE0 and EDMA (IE.6) are set, a DMA interrupt request is generated.

If the DMA function is disabled, the Timer 5 and Timer 6 of DMA module can be a general 16-bit timer. Each timer has its own timer flag, TF5 and TF6 with corresponding interrupt enable bit. They share the DMA interrupt with DAM complete flag. The following diagram shows the DMA interrupt architecture. If software

enables DMA transfer function, the interrupt enables of Timer 5 and Timer 6 must be disabled.

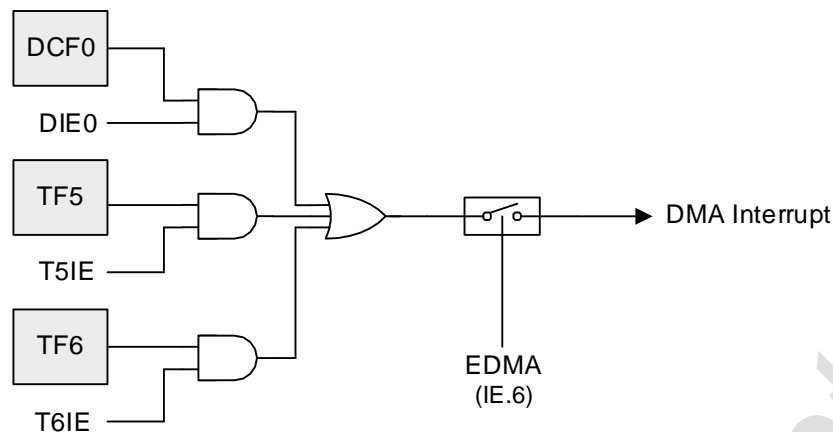


Figure 8-3. DMA Interrupt

8.2.7 DMA Loop Mode

Loop mode is available to handle circular buffers and continuous data flows (e.g. ADC scan mode). This feature can be enabled using the LOOP bit in the DMACG0 register. When loop mode is activated, the Current Transfer Count is automatically reloaded with the Base Transfer Count, the Current Address is automatically reloaded with the Base Address, and the DMA requests continue to be served without setting DMAS0.

8.2.8 Error Handling in DMA

There is no any error handling function in the DMA controller, software will take care on:

- A. Current Address cannot over the XRAM boundary. In CMT2380F17, XRAM boundary is 768 bytes (02FFH).
- B. Cannot support the even/odd parity check and generation on S0 and S1.
- C. Cannot handle the Not ACK status on TWI0/I2C0.。

8.2.9 Data Copied to CRC16

If DMA destination is not CRC16 module, enabled CRCW0 will copy the data content to CRC16 module on each DMA transaction. For example, moving data from S0 RX to SPI0 TX will fed the data to CRC16 simultaneously. This function is supported in any transfer type.

8.2.10 Timer 5 & Timer 6

When DMA enabled, Timer 5 behaves the function for DMA transfer counting. TH5 and TL5 are the Current Transfer Count registers. THR5 and TLR5 are the Base Transfer Count registers.

If DMA is disabled, Timer 5 is a 16-bit auto-reloadable timer/counter with Gate control function as Timer 0. The overflow flag, TF5, could be an interrupt source and shares the DMA interrupt vector. Following figure illustrates the Timer 5 structure.

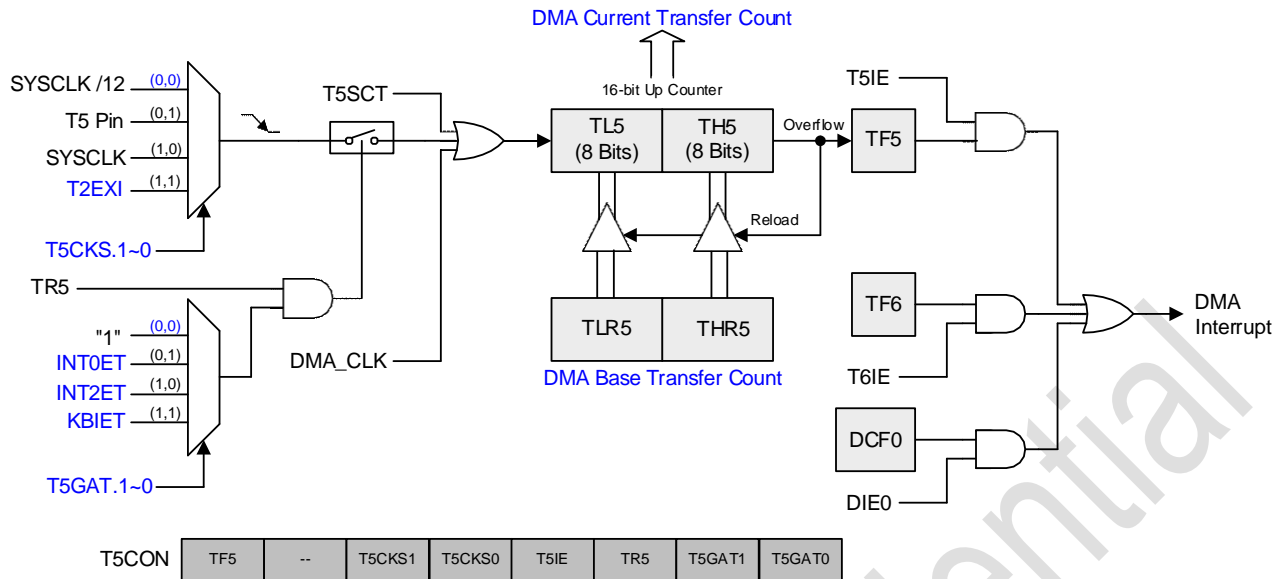


Figure 8-4. Timer 5 Structure

When DMA enabled, Timer 6 behaves the function for DMA memory address pointer. TH6 and TL6 are the Current Address registers. THR6 and TLR6 are the Base Address registers.

If DMA is disabled, Timer 6 is a 16-bit auto-reloadable timer/counter with Gate control function as Timer 0. The overflow flag, TF6, could be an interrupt source and shares the DMA interrupt vector. Following figure illustrates the Timer 5 structure.

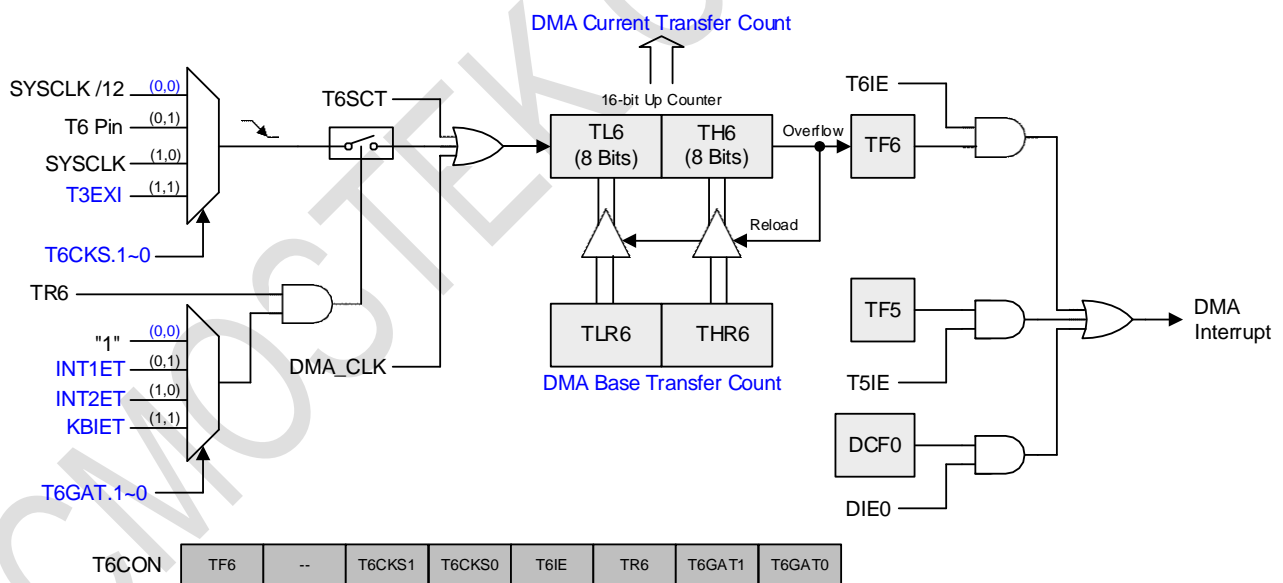


Figure 8-5. Timer 6 Structure

8.3 DMA Register

- DMACR0: DMA Control Register 0
SFR Page = 0~7
SFR Address = 0x94

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	--	DMAE0	DMA0	DIE0	DCF0

R/W	W	W	W	R/W	R/W	R/W	R/W	R/W
Reset Value	X	X	X	0	0	0	0	0

Bit 7~4: Reserved. Software must write "0" on these bits when DMACR0 is written.

Bit 3: DMAE0, DMA Enable 0.

0: Clear to disable DMA operation. 1: Set to enable DMA operation.

Bit 2: DMAS0, DMA transfer Start 0.

0: Cleared by H/W when DMA end-of-transfer. If Cleared by S/W will suspend DMA transfer. 1: Setting this bit by software starts or resume the DMA transfer.

Bit 1: DIE0, DCF0 Interrupt Enable. 0: Disable DCF0 interrupt.

1: Enable DCF0 interrupt to share the DMA interrupt vector.

Bit 0: DCF0, DMA Complete Flag 0.

0: DCF0 must be cleared by software writing 0.

1: DCF0 is set by DMA end-of-transfer.

- DMACG0: DMA ConfiGuration Register 0

SFR Page = 8 only

SFR Address = 0x94

Bit	7	6	5	4	3	2	1	0
Name	PDMAH	PDMAL	CRCW0	0	EXTS10	EXTS00	FAEN0	LOOP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~6: PDMAH/L, DMA interrupt priority control bits.

Bit 5: CRCW0, CRC16 Write (copied) enable.

0: Disable the DMA data is copied to CRC16 concurrently.

1: Enable the DMA data is copied to CRC16 concurrently.

Bit 4: Reserved. Software must write "0" on these bits when DMACG0 is written.

Bit 3~2: EXTS10~00, DMA external trigger source selection.

EXTS10, EXTS00	Selected Signal
0 0	Disabled, software trigger
0 1	INT2ET
1 0	Reserved
1 1	KBIET

Bit1: Reserved.

Bit 0: LOOP0.

0: Disable DMA loop operation.

1: Enable DMA loop operation.

- DMADS0: DMA Data path Selection Register 0
SFR Page = 9 only
SFR Address = 0x94

Bit	7	6	5	4	3	2	1	0
Name	DSS30	DSS20	DSS10	DSS00	DDS30	DDS20	DDS10	DDS00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~4: DMA data Source Selection.

Bit 3~0: DMA data Destination Selection.

DSS30~00	Source Selection	DDS30~00	Destination Selection
0 0 0 0	Disabled	0 0 0 0	Disabled
0 0 0 1	S0RX	0 0 0 1	S0TX
0 0 1 0	S1RX	0 0 1 0	S1TX
0 1 0 1	TWI0RX	0 1 0 1	TWI0TX
0 1 1 1	SPI0RX	0 1 1 1	SPI0TX
1 0 0 1	ADC0 ^[1]	1 0 0 1	Reserved
1 1 0 1	Reserved	1 1 0 1	CRC
1 1 1 1	XRAM	1 1 1 1	XRAM

Note 1: When use DMA to transfer ADC data, please watch out the Data Bit setting. Please reference "Transfer ADC Data by DMA" in ADC Module for details.

8.4 Timer5 Register

- T5CON: Timer 5 Control Register
SFR Page = 3 Only
SFR Address = 0xC8

Bit	7	6	5	4	3	2	1	0
Name	TF5	--	T5CKS1	T5CKS0	T5IE	TR5	T5GAT1	T5GAT0
R/W	R/W	W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: TF5, Timer 5 overflow flag.

0: TF5 must be cleared by software.

1: TF5 is set by a Timer 5 overflow happened.

Bit 6: --.

Bit 5~4: T5CKS.1~0, Timer 5 clock source selector.

T5CKS.1~0	T5 Clock Selection
00	SYSClk/12
01	T5 pin (P3.4)
10	SYSClk
11	T2EXI input selection

Bit 3: T5IE, TF5 interrupt enable.

0: Disable TF5 interrupt.

1: Enable TF5 interrupt to share the DMA interrupt vector.

Bit 2: TR5, Timer 5 Run control bit.

0: Disabled to stop the Timer/Counter 5. Before starting the DMA process, software must be disabled TR5.

1: Enabled to start the Timer/Counter 5.

Bit 1~0: T5GAT.1~0, Gating source selection of Timer 5.

T5GAT.1~0	T5 Gate source
-----------	----------------

00	Disable
01	INT0ET
10	INT2ET
11	KBIET

- TL5: Timer 5 Low byte Register
SFR Page = 3 Only
SFR Address = 0xCC

Bit	7	6	5	4	3	2	1	0
Name	TL5 [7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

- TH5: Timer 5 High byte Register
SFR Page = 3 Only
SFR Address = 0xCD

Bit	7	6	5	4	3	2	1	0
Name	TH5 [7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

- TLR5: Timer 5 Low byte Reload Register
SFR Page = 3 Only
SFR Address = 0xCA

Bit	7	6	5	4	3	2	1	0
Name	TLR5 [7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

- THR5: Timer 5 High byte Reload Register
SFR Page = 3 Only
SFR Address = 0xCB

Bit	7	6	5	4	3	2	1	0
Name	THR5 [7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

8.5 Timer 6 Register

- T6CON: Timer 6 Control Register
SFR Page = 4 Only
SFR Address = 0xC8

Bit	7	6	5	4	3	2	1	0
Name	TF6	--	T6CKS1	T6CKS0	T6IE	TR6	T6GAT1	T6GAT0
R/W	R/W	W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: TF6, Timer 6 overflow flag.
0: TF6 must be cleared by software.
1: TF6 is set by a Timer 6 overflow happened.

Bit 6: --.

Bit 5~4: T6CKS.1~0, Timer 6 clock source selector.

T6CKS.1~0	T6 Clock Selection
00	SYSCLK/12
01	T6 pin
10	SYSCLK
11	T3EXI input selection

Bit 3: T6IE, TF6 interrupt enable.
0: Disable TF6 interrupt.
1: Enable TF6 interrupt to share the DMA interrupt vector.

Bit 2: TR6, Timer 6 Run control bit.
0: Disabled to stop the Timer/Counter 6. Before starting the DMA process, software must be disabled TR6.
1: Enabled to start the Timer/Counter 6.

Bit 1~0: T6GAT.1~0, Gating source selection of Timer 6.

T6GAT.1~0	T6 Gate source
00	Disable
01	INT1ET
10	INT2ET
11	KBIET

TL6: Timer 6 Low byte Register

SFR Page = 4 Only

SFR Address = 0xCC

Bit	7	6	5	4	3	2	1	0
Name	TL6 [7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

TH6: Timer 6 High byte Register

SFR Page = 4 Only

SFR Address = 0xCD

Bit	7	6	5	4	3	2	1	0
Name	TH6 [7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

- TLR6: Timer 6 Low byte Reload Register

SFR Page = 4 Only

SFR Address = 0xCA

Bit	7	6	5	4	3	2	1	0
Name	TLR6 [7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

THR6: Timer 6 High byte Reload Register

SFR Page = 4 Only

SFR Address = 0xCB

Bit	7	6	5	4	3	2	1	0
Name	THR6 [7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

9 System Clock

There are three clock sources for the system clock: Internal High-frequency RC Oscillator (IHRCO), Internal Low-frequency RC Oscillator (ILRCO) and External Clock Input. Figure 9–1 shows the structure of the system clock in CMT2380F17.

The CMT2380F17 always boots from IHRCO on 12MHz. Software can select the OSCin input on one of the three clock sources application required and switches them on the fly. But software needs to settle the clock source stably before clock switching. In external clock input mode (ECKI), the clock source comes from P6.0 input.

The built-in IHRCO provides two kinds of frequency for software selected. Another frequency is 11.059MHz by software setting AFS on CKCON0.7. Both of 12MHz and 11.059 MHz in IHRCO provide high precision frequency for system clock source. To find the detailed IHRCO performance, please refer Section “33.4 IHRCO Characteristics”). In IHRCO or ILRCO mode, P6.0 can be configured to internal MCK output or MCK/2 and MCK/4 for system application.

The built-in ILRCO provides the low power and low speed frequency about 32KHz to WDT and system clock source. MCU can select the ILRCO to system clock source by software for low power operation. To find the detailed IHRCO performance, please refer Section “33.5 ILRCO Characteristics”). In ILRCO mode, P6.0 can be configured to internal MCK output or MCK/2 and MCK/4 for system application.

The CMT2380F17 device includes a Clock Multiplier (CKM) to generate the high speed clock for system clock source. CKM applied in CMT2380F17 is shown in Figure 9–1 and its typical input frequency is around 6MHz. Before enable CKM, software must configure the CKMIS1~0 (CKCON.5~4) to get the reasonable CKMI frequency for CKM input source. CKM can generate 4/5.33/8 times frequency of CKMI and setting MCKS1~0 (CKCON.2.3~2) selects different CKM outputs to provide the high speed operation on MCU without high-frequency clock source. To find the detailed CKM performance, please refer Section “33.6 CKM Characteristics”).

The system clock, SYSCLK, is obtained from one of these four clock sources through the clock divider, as shown in Figure 9–1. The user can program the divider control bits SCKS2~SCKS0 (in CKCON0 register) to get the desired system clock.

9.1 Clock Structure

Figure 9–1 presents the principal clock systems in the CMT2380F17. The initial oscillator source of CPUCLK is set to IHROC 12MHz. It can use the combinations of the clock multiplier and divider for different frequencies. The maximum CPUCLK is as following:

- External clock input mode: Up to 12MHz @ 2.0V – 5.5V; Up to 25MHz @ 2.4V – 5.5V
- CPU up to 12MHz @ 1.8V – 5.5V; Up to 25MHz @ 2.2V – 5.5V
- CPU up to 36MHz @ 2.7V -5.5V with on-chip CKM

If the applications need higher performance, then HSE (DCON0 Bit 7) needs to be set when CPUCLK > 6MHz. Moreover, if needs ultra-high CPUCLK>25MHz, then HSE1 needs to be set.

The system clock can be sourced by the external oscillator circuit or either internal oscillator. It maximum

frequency is 50MHz. Please note, when using Clock Multiplier (CKM) to raise the MCK frequency to get higher SYSCLK, the CPULCK will be also changed. It is need to set CCKS to slow down CPULCK before raise MCK frequency to avoid CPULCK over clock (CPULCK needs to lower then 25MHz or 36MHz).

The clock module also provide two more clock source for high speed PCA applications.

- MCKDO: Up to 72MHz
- CKMIX16: Up to 144MHz

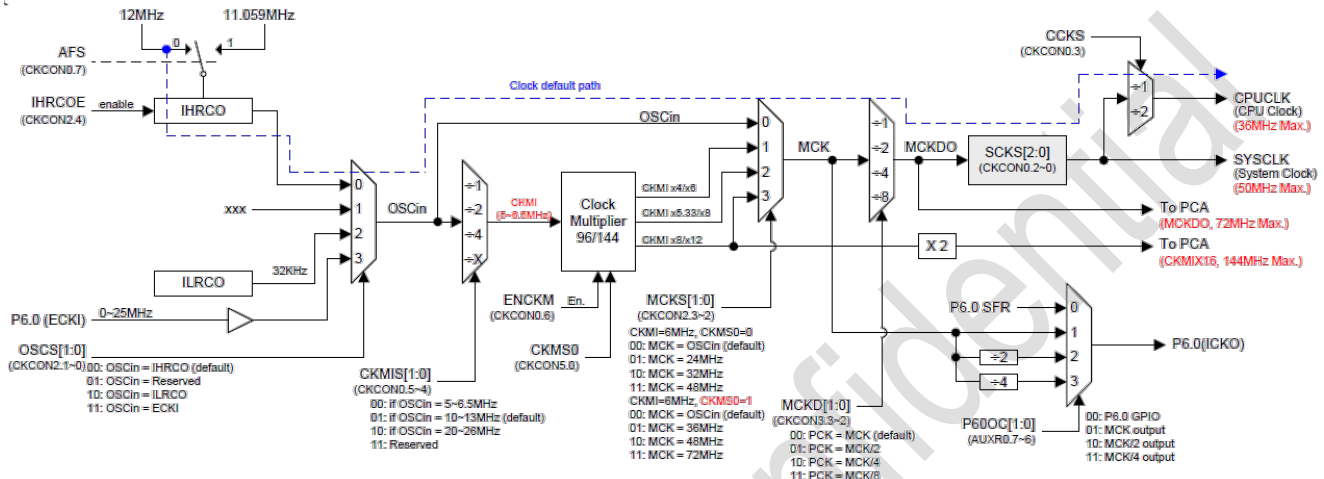


Figure 9-1. System Clock

9.2 Clock Source Switching

There are three clock sources for the system clock: Internal High-frequency RC Oscillator (IHRCO), Internal Low-frequency RC Oscillator (ILRCO) and External Clock Input. Figure 9–1 shows the structure of the system clock in CMT2380F17. The CMT2380F17 always boots from IHRCO on 12MHz. OSCS[1:0] are used to select the clock source by software setting, but the software need to wait until the clock be settle before switch the clock source..

9.3 On-chip CKM (PLL)

There are three clock sources for the system clock: Internal High-frequency RC Oscillator (IHRCO), Internal Low-frequency RC Oscillator (ILRCO) and External Clock Input. Figure 9–1 shows the structure of the system clock in CMT2380F17. The CMT2380F17 always boots from IHRCO on 12MHz. OSCS[1:0] are used to select the clock source by software setting, but the software need to wait until the clock be settle before switch the clock source..

9.4 Wake-up clock from CKM

When enable CKM circuit, it needs 100us to output stable frequency, within this uncertain frequency period, the input of the MCK needs to keep MCKS on OSCIn to guarantee system's satiability. Please reference the following procedure:

How to Program to Support wake-up with clock from CKM

- Program MCKS[1:0] (CKCON2.3~2) to "00" to select non-CKM output as clock source
- MCU enters power down
-
- MCU wakes up
- delay 100us to wait CKM working stable.
- Modify MCKS[1:0] (CKCON2.3~2) to select CKM output as clock source

- Continue program execution.....

9.5 Clock Register

CKCON0: Clock Control Register 0

SFR Page = 0~F & P

SFR Address = 0xC7

Bit	7	6	5	4	3	2	1	0
Name	AFS	ENCKM	CKMIS1	CKMIS0	CCKS	SCKS2	SCKS1	SCKS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	1	0	0	0	0

Bit 7: AFS, Alternated Frequency Selection.

0: Select IHRCO on 12MHz.

1: Select IHRCO on 11.059MHz.

Bit 6: ENCKM, Enable clock multiplier (X8/X12)

0: Disable the X8/X12 clock multiplier.

1: Enable the X8/X12 clock multiplier.

Bit 5~4: CKMIS1 ~ CKMIS0, Clock Multiplier Input Selection.

CKMIS[1:0]	Clock Multiplier Input Selection
0 0	OSCIn/1 (when OSCIn = 5 ~ 7MHz)
0 1	OSCIn/2 (when OSCIn = 10 ~ 14MHz)
1 0	OSCIn/4 (when OSCIn = 20 ~ 28MHz)
1 1	Reserved

Bit 3: CCKS, CPU Clock Select.

0: Select CPU Clock as SYSCLK.

1: Select CPU Clock as SYSCLK/2.

Bit2~0:SCKS2~SCKS0,programmableSystem ClockSelection.

SCKS[2:0]	SystemClock(SYSCLK)
0 0 0	MCKDO/1
0 0 1	MCKDO/2
0 1 0	MCKDO/4
0 1 1	MCKDO/8
1 0 0	MCKDO/16
1 0 1	MCKDO/32
1 1 0	MCKDO/64
1 1 1	MCKDO/128

CKCON2: Clock Control Register 2

SFR Page = P Only

SFR Address = 0x40

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	IHRCOE	MCKS[1:0]		OSCS[1:0]	
R/W	W	W	W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	1	0	0	0	0

Bit 4: IHRCOE, Internal High frequency RC Oscillator Enable.

0: Disable internal high frequency RC oscillator.

1: Enable internal high frequency RC oscillator. If this bit is set by CPU software, it needs 32 us to have stable output after IHRCOE is enabled.。

Bit 3~2: MCKS[1:0], MCK clock source selection.

MCKS[1:0]	MCK Source Selection	OSCIn =12MHz CKMIS = [01]		OSCIn =11.059MHz CKMIS = [01]	
		CKMS0 = 0	CKMS0 = 1	CKMS0 = 0	CKMS0 = 1
00	OSCIn	12MHz		11.059MHz	
01	CKMI x4 / x6	24MHz	36MHz	22.118MHz	33.177MHz
10	CKMI x5.33 / x8	32MHz	48MHz	29.491MHz	44.236MHz
11	CKMI x8 / x12	48MHz	72MHz	44.236MHz	66.354MHz

Note: It needs to set ENCKM = 1 to enable CKM.

Note: Needs to be careful of the limitation of CPUCLK and SYSCLK. Needs to use SCKS[2:0] and CCKS to choose proper range of CPUCLK and SYSCLK to not exceed the limitation. CPUCLK \leq 36MHz, SYSCLK \leq 50MHz.

Bit 1~0: OSCS[1:0], OSCIn Source selection.

OSCS[1:0]	OSCIn source Selection
00	IHRCO
01	ECKI, external clock input (P6.0) as OSCIn
10	ILRCO
11	Reserved

CKCON3: Clock Control Register 3

SFR Page = P only

SFR Address = 0x41

Bit	7	6	5	4	3	2	1	0
Name	WDTCS[1:0]		FWKP	WDTFS	MCKD[1:0]		--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	W	W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~6: WDT clock source selection.

Bit 5: FWKP, MCU Fast wake up control

0: Select MCU for normal wakeup time about 120us from power-down mode. 1: Select MCU for fast wakeup time about 30us from power-down mode.

Bit 4: WDTFS, WDT overflow source selection.

0: Select WDT bit-8 overflow as WDT event source. 1: Select WDT bit-0 overflow as WDT event source.

Bit 3~2: MCKD[1:0], MCK Divider Output selection.

MCKD[1:0]	MCKDO Frequency	Example MCK = 12MHz	Example MCK = 48MHz
00	MCKDO = MCK	MCKDO = 12MHz	MCKDO = 48MHz
01	MCKDO = MCK/2	MCKDO = 6MHz	MCKDO = 24MHz
10	MCKDO = MCK/4	MCKDO = 3MHz	MCKDO = 12MHz
11	MCKDO = MCK/8	MCKDO = 1.5MHz	MCKDO = 6MHz

- CKCON5: Clock Control Register 5

SFR Page = P only

SFR Address = 0x43

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	--	--	--	--	CKMS0
R/W	W	W	W	W	W	W	W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~1: Reserved. Software must write "0" on these bits when CKCON5 is written.

Bit 0: CKMS0, CKM mode selection 0.

0: Select CKM operating for 8X mode. (96MHz)

1: Select CKM operating for 12X mode. (144MHz)

- AUXR0: Auxiliary Register 0

SFR Page = 0~F

SFR Address = 0xA1

Bit	7	6	5	4	3	2	1	0
Name	P60OC[1:0]		P60FD	PBKF	--		INT1H	INT0H
R/W	R/W	R/W	R/W	R/W	W	W	R/W	R/W
Reset Value	0	0	0	X	X	X	0	0

Bit 7~6: P6.0 function configured control bit 1 and 0. The two bits only act when internal RC oscillator (IHRCO or ILRCO) is selected for system clock source. In external clock input mode, P6.0 is the dedicated clock input pin. In internal oscillator condition, P6.0 provides the following selections for GPIO or clock source generator. When P60OC[1:0] index to non-P6.0 GPIO function, P6.0 will drive the on-chip RC oscillator output to provide the clock source for other devices.

P60OC[1:0]	P60 Function	I/O Mode
00	P60	By P6M0.0
01	MCK	By P6M0.0
10	MCK/2	By P6M0.0
11	MCK/4	By P6M0.0

For clock-out on P6.0 function, it is recommended to set P6M0.0 to "1" which selects P6.0 as push-push output mode.

Bit 5: P60FD, P6.0 Fast Driving.

0: P6.0 output with default driving.

1: P6.0 output with fast driving enabled. If P6.0 is configured to clock output, enable this bit when P6.0 output frequency is more than 12MHz at 5V application or more than 6MHz at 3V application.

- DCON0: Device Control Register 0

SFR Page = P Only
SFR Address = 0x4C

Bit	7	6	5	4	3	2	1	0
Name	HSE	IAPO	HSE1	--	--	IORCTL	RSTIO	OCDE
R/W	R/W	R/W	W	W	W	R/W	R/W	R/W
Reset Value	1	0	0	0	X	0	1	1

Bit 7: HSE, High Speed operation Enable.

0: Select CPU running in lower speed mode ($FCPUCLK \leq 6\text{MHz}$) which is slow down internal circuit to reduce power consumption.

1: Enable CPU full speed operation if $FCPUCLK > 6\text{MHz}$. Before select high frequency clock ($> 6\text{MHz}$) on CPUCLK, software must set HSE to switch internal circuit for high speed operation.

Bit 5: HSE1, High Speed operation Enable 1. 0: No function.

1: Enable MCU for ultra-high speed operation. ($FCPUCLK > 25\text{MHz}$). It also needs to set HSE when use HSE1 = 1.

10 Watch Dog Timer (WDT)

10.1 WDT Structure

The Watch-dog Timer (WDT) is intended as a recovery method in situations where the CPU may be subjected to software upset. The WDT consists of a 8-bit free-running counter, a 8-bit prescaler and a control register (WDTCSR). Figure 10–1 shows the WDT structure in CMT2380F17.

There are four selections for WDT clock source. The clock source must be configured before WDT enabled. The default WDT clock source is 32 KHz ILRCO. The WDT overflow will set the WDTF (PCON1.0) which can be configured to generate an interrupt by enabled WDTFIE (SFIE.0) and enabled ESF (EIE1.3). The overflow can also trigger a system reset when WREN (WDTCSR.7) is set. To prevent WDT overflow, software needs to clear it by writing “1” to the CLRW bit (WDTCSR.4) before WDT overflows.

Once the WDT is enabled by setting ENW bit, there is no way to disable it except through power-on reset or page-p SFR over-write on ENW, which will clear the ENW bit. The WDTCSR register will keep the previous programmed value unchanged after hardware (RST-pin) reset, software reset and WDT reset.

WREN, NSW and ENW are implemented to one-time-enabled function, only writing “1” valid in general SFR page. Page-P SFR Access on WDTCSR can disable WREN, NSW and ENW, writing “0” on WDTCSR.7~5. Please refer Section “10.4 WDT Register” and Section “29 Page P SFR Access” for more detail information.

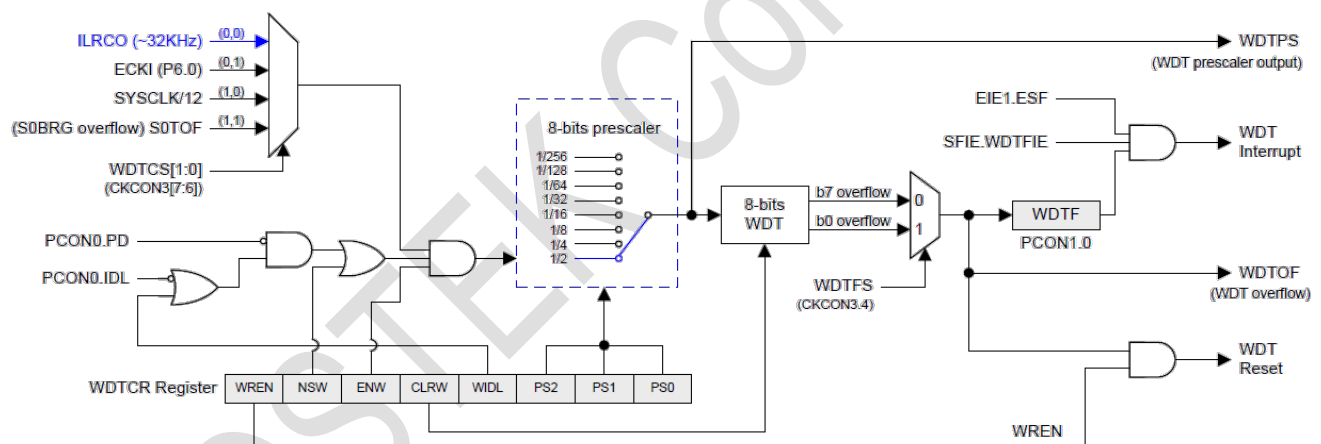


Figure 10-1. Watch Dog Timer

10.2 WDT During Idle

In the Idle mode, the WIDL bit (WDTCSR.3) determines whether WDT counts or not. Set this bit to let WDT keep counting in the Idle mode. If the hardware option NSWDT is enabled, the WDT always keeps counting regardless of WIDL bit.

In the Power down mode, the ILRCO won't stop if the NSW (WDTCSR.6) is enabled. The MUC enters Watch mode to behave an auto-wakeup function. That lets WDT keep counting even in Power down mode (Watch Mode). After WDT overflows, it will wake up the CPU from interrupt or reset by software configured. This function is only active when WDT clock source is come from ILRCO or P6.0 input which can be derived from external input.

10.3 WDT Register

- **WDTCR**: Watch-Dog-Timer Control Register

SFR Page = 0~F & P

SFR Address = 0xE1

POR = XXX0-XXXX (0000-0111)

Bit	7	6	5	4	3	2	1	0
Name	WREN	NSW	ENW	CLRW	WIDL	PS[2:0]		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	1	1	1

Bit 7: WREN, WDT Reset Enable. The initial value can be changed by hardware option, WREN0.

0: The overflow of WDT does not set the WDT reset. The WDT overflow flag, WDTF, may be polled by software or trigger an interrupt.

1: The overflow of WDT will cause a system reset. Once WREN has been set, it cannot be cleared by software in page 0~F. **In page P, software can modify it to "0" or "1".**

Bit 6: NSW. Non-Stopped WDT. The initial value can be changed by hardware option, NSWDT.0: WDT stop counting while the MCU is in power-down mode.

1: WDT always keeps counting while the MCU is in power-down mode (Watch Mode) or idle mode. Once NSW has been set, it cannot be cleared by software in page 0~F. **In page P, software can modify it to "0" or "1".**

Bit 5: ENW. Enable WDT.

0: Disable WDT running. This bit is only cleared by POR.

1: Enable WDT while it is set. Once ENW has been set, it can not be cleared by software in page 0~F. **In Page P, software can modify it as "0" or "1".**

Bit 4: CLRW. WDT clear bit.

0: Writing "0" to this bit is no operation in WDT.

1: Writing "1" to this bit will clear the 8-bit WDT counter to 00H. Note this bit has no need to be cleared by writing "0". Clear WDT to recount while it is set.

Bit 3: WIDL. WDT idle control.

0: WDT stops counting while the MCU is in idle mode. 1: WDT keeps counting while the MCU is in idle mode.

Bit 2~0: PS2~PS0, select prescaler output for WDT time base input.

When WDTFS (CKCON3.4) = 0, WDT clock source = ILRCO or SYSCLK/12

PS[2:0]	Prescaler Value	WDT Period (WDT clock = ILRCO)	WDT Period (WDT clock = SYSCLK/12) (SYSCLK = ILRCO, 12MHz)
0 0 0	2	16 ms	0.512 us
0 0 1	4	32 ms	1.024 ms
0 1 0	8	64 ms	2.048 ms
0 1 1	16	128 ms	4.096 ms
1 0 0	32	256 ms	8.192 ms
1 0 1	64	512 ms	16.384 ms
1 1 0	128	1024 ms	32.768 ms
1 1 1	256	2048 ms	65.536 ms

When WDTFS (CKCON3.4) = 1, WDT clock source = ILRCO

PS[2:0]	分频值	WDT Period Note ^[1] (clock source = ILRCO)
0 0 0	2	0.125 ms + 120us
0 0 1	4	0.25 ms + 120us
0 1 0	8	0.5 ms + 120us
0 1 1	16	1 ms + 120us
1 0 0	32	2 ms + 120us
1 0 1	64	4 ms + 120us
1 1 0	128	8 ms + 120us
1 1 1	256	16 ms + 120us

Note:

[1] When WDT clock source is ILRCO, the WDT internal logic latency is around 120us. Under this condition we suggest to add 120us of WDT period if shorter than 12ms.

● **CKCON3**: Clock Control Register 3

SFR Page = P only

SFR Address = 0x41

Bit	7	6	5	4	3	2	1	0
Name	WDTCS[1:0]		FWKP	WDTFS	MCKD1	MCKD0	1	0
R/W	R/W	R/W	W	R/W	R/W	R/W	W	W
Reset Value	0	0	0	0	0	0	1	0

Bit 7~6: WDTCS1~0, WDT Clock Source selection [1:0].

WDTCS1~0	WDT Clock Source
00	ILRCO
01	ECKI(P6.0)
10	SYSCLK/12
11	S0TOF

Bit 4: WDTFS. WDT overflow source selection.

0: Select WDT bit-8 overflow as WDT event source.

1: Select WDT bit-0 overflow as WDT event source.

● **PCON1**: Power Control Register 1

SFR Page = 0~F & P

SFR Address = 0x97

Bit	7	6	5	4	3	2	1	0
Name	SWRF	EXRF	--	RTCF	--	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	W	R/W	R/W	W
Reset Value	0	0	0	0	X	0	0	0

Bit 1: WDTF, WDT overflow flag.

0: This bit must be cleared by software writing "1" on it. Software writing "0" is no operation.

1: This bit is only set by hardware when WDT overflows. Writing "1" on this bit will clear WDTF.

● **SFIE**: System Flag Interrupt Enable Register

SFR Page = 0~F

SFR Address = 0x8E

Bit	7	6	5	4	3	2	1	0
Name	SIDFIE	--	--	RTCFIE	--	BOF1IE	BOF0IE	WDTFIE
R/W	R/W	W	W	R/W	W	R/W	R/W	R/W
Reset Value	0	0	0	0	X	0	0	0

Bit 0: WDTFIE, Enable WDTF (PCON1.0) Interrupt.

0: Disable WDTF interrupt.

1: Enable WDTF interrupt.

10.4 WDT Hardware Option

In addition to being initialized by software, the WDTCR register can also be automatically initialized at power-up by the hardware options WRENO, NSWDT, HWENW, HWWIDL and HWPS[2:0], which should be programmed by a universal Writer or Programmer, as described below.

If HWENW is programmed to "enabled", then hardware will automatically do the following initialization for the WDTCR register at power-up:

- (1) set ENW bit;
- (2) load WRENO into WREN bit;
- (3) load NSWDT into NSW bit;
- (4) load HWWIDL into WIDL bit;
- (5) load HWPS[2:0] into PS[2:0] bits.

If both of HWENW and WDSFWP are programmed to “enabled”, hardware still initializes the WDTCR register content by WDT hardware option at power-up. Then, any CPU writing on WDTCR bits will be inhibited except writing “1” on WDTCR.4 (CLRW), clear WDT, even though access through Page-P SFR mechanism.

WRENO:

- ☒ ☒: Enabled. Set WDTCR.WREN to enable the system reset function by WDTF.
- ☐: Disabled. Clear WDTCR.WREN to disable the system reset function by WDTF.

NSWDT: Non-Stopped WDT

- ☒: Enabled. Set WDTCR.NSW to enable the WDT running in powerdown mode (watch mode).
- ☐: Disabled. Clear WDTCR.NSW to disable the WDT running in powerdown mode (disable Watch mode).

HWENW: Hardware loaded for “ENW” of WDTCR.

- ☒: Enabled. Enable WDT and load the content of WRENO, NSWDT, HWWIDL and HWPS2~0 to WDTCR after power-on.
- ☐: Disabled. WDT is not enabled automatically after power-on.

HWWIDL, HWPS2, HWPS1, HWPS0:

When HWENW is enabled, the content on these four fused bits will be loaded to WDTCSR SFR after power-on.

WDSFWP:

- ☒: Enabled. The WDT SFRs, WREN, NSW, WIDL, PS2, PS1 and PS0 in WDTCR, will be write-protected.
- ☐: Disabled. The WDT SFRs, WREN, NSW, WIDL, PS2, PS1 and PS0 in WDTCR, are free for writing of software.

11 Real-Time-Clock (RTC)/System-Timer

The CMT2380F17 has a simple Real-Time clock that allows a user to continue running an accurate timer while the rest of the device is powered-down. The Real-Time clock can be a wake-up or an interrupt source. The Real-Time clock is a maximum 21-bit up counter comprised of a 0~15-bit prescaler and a 6-bit loadable up counter. When it overflows, the 6-bit counter will be reloaded again and the RTCF flag will be set. The clock source for this prescaler has 6 selections, and needs to set RCSS[2:0] to select one of source before enable WDT. Figure 11–1 shows the RTC structure in CMT2380F17.

To input 32.768 KHz from ECKI for the RTC module input will provide a programmable overflow period for 0.5S to 64S. The counter also provides a timer function with the clock derived from SYSCLK for a system timer function. The maximum overflow period for the system timer function is $\text{SYSCLK}/2^{21}$. The ILRCO provides the internal clock source for RTC module. The WDTPS and WDTOF come from WDT prescaler and WDT overflow to provide the extended prescaler source for more long wake-up time requirement. The RCT clock source must be configured before RTCE enabled.

RTCO enables the RTC overflow output on port pin. Only power-on reset will reset the Real-Time clock and its associated SFRs to the default state.

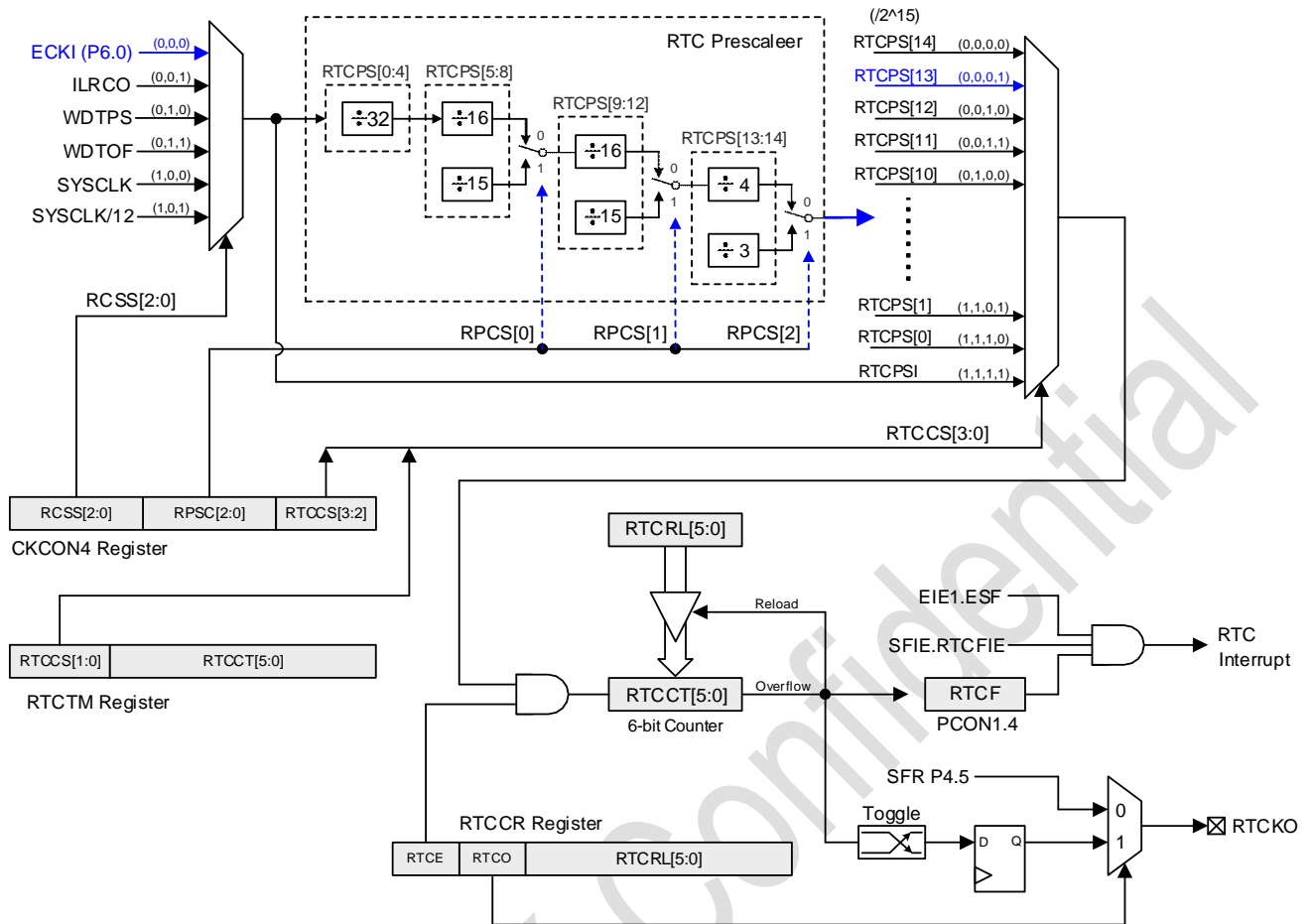


Figure 11-1. Real-Time-Clock Counter

● **RTCCR**: Real-Time-Clock Control Register

SFR Page = 0~7 & P

SFR Address = 0xBE/0x54

Bit	7	6	5	4	3	2	1	0
Name	RTCE	RTCO	RTCRL[5:0]					
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	1	1	1	1	1	1

Bit 7: RTCE, RTC Enable.

0: Stop RTC Counter, RTCCT.

1: Enable RTC Counter and set RTCF when RTCCT overflows. When RTCE is set, CPU can not access RTCTM. RTCTM must be accessed in RTCE cleared.

Bit 6: RTCO, RTC Output enabled. The frequency of RTCKO is (RTC overflow rate)/2. 0: Disable the RTCKO output.

1: Enable the RTCKO output on P4.5.

Bit 5~0: RTCRL[5:0], RTC counter reload value register. This register is accessed by CPU and the content in the register is reloaded to RTCCT when RTCCT overflows.

● **RTCTM**: Real-Time-Clock Timer Register

SFR Page = 0~7 & P

SFR Address = 0xB6/0x55

Bit	7	6	5	4	3	2	1	0
Name	RTCCS[1:0]		RTCCT[5:0]					

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	1	1	1	1	1	1

Bit 7~6: RTCCS.1~0, RTC Clock Selection. Default is "01".

RTCCS3~0	Clock Source	RTC Interrupt Duration	Minimum Step
0 0 0 0	RTCPs[14] ($/2^{15}$)	1S ~ 64S when P6.0 = 32768Hz	1S
0 0 0 1	RTCPs[13] ($/2^{14}$)	0.5S ~ 32S when P6.0 = 32768Hz	0.5S (默认)
0 0 1 0	RTCPs[13] ($/2^{13}$)	0.25S ~ 16S when P6.0 = 32768Hz	0.25S
.....
1 0 1 0	RTCPs[4] ($/2^5$)	976us ~ 62.46ms when P6.0 = 32768Hz	976 us
1 0 1 1	RTCPs[3] ($/2^4$)		488 us
1 1 0 0	RTCPs[2] ($/2^3$)		244 us
1 1 0 1	RTCPs[1] ($/2^2$)	122us ~ 3.9ms when P6.0 = 32768Hz	122 us
1 1 1 0	RTCPs[0] ($/2^1$)	61us ~ 1.952ms when P6.0 = 32768Hz	61 us
1 1 1 1	RTCPsI ($/2^0$)	30.5us ~ 976us when P6.0 = 32768Hz	30.5 us

Bit 5~0: RTCCT[5:0], RTC counter register. It is a counter for RTC function or System Timer function by different clock source selection on RTCCS[1:0]. When the counter overflows, it sets the RTCF flag which shares the system flag interrupt when RTCFIE is enabled. The maximum RTC overflow period is 64 seconds.

● **CKCON4:** Clock Control Register 4

SFR Page = P only

SFR Address = 0x42

Bit	7	6	5	4	3	2	1	0
Name	RCSS[2:0]			RPCS[2:0]			RTCCS[3:2]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~5: RTC Clock Source selection [2:0]

RCSS[2:0]	RTC Clock Selection
0 0 0	ECKI (P6.0)
0 0 1	ILRCO
0 1 0	WDTPS
0 1 1	WDTOF
1 0 0	SYSCLK
1 0 1	SYSCLK/12
Others	Reserved

Bit 4: RPSC2, RTC prescaler control bit 2.

Bit 3: RPSC1, RTC prescaler control bit 1.

Bit 2: RPSC0, RTC prescaler control bit0.

Bit 1~0: RTCCS[3:2], RTC counter clock selection, using together with RTCCS[1:0].

● **PCON1:** Power Control Register 1

SFR Page = 0~F & P

SFR Address = 0x97

Bit	7	6	5	4	3	2	1	0
-----	---	---	---	---	---	---	---	---

Name	SWRF	EXRF	--	RTCF	--	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	W	R/W	R/W	R/W
Reset Value	0	0	0	0	X	0	0	0

Bit 4: RTCF, RTC overflow flag.

0: This bit must be cleared by software writing "1" on it. Software writing "0" is no operation.

1: This bit is only set by hardware when RTCCT overflows. Writing "1" on this bit will clear RTCF.

● **SFIE: System Flag Interrupt Enable Register**

SFR Page = 0~F

SFR Address = 0x8E

Bit	7	6	5	4	3	2	1	0
Name	SIDFIE	--	--	RTCFIE	--	BOF1IE	BOF0IE	WDTFIE
R/W	R/W	R/W	R/W	R/W	W	R/W	R/W	R/W
Reset Value	0	1	1	0	X	0	0	0

Bit 4: RTCFIE, Enable RTCF (PCON1.4) Interrupt. 0: Disable RTCF interrupt.

1: Enable RTCF interrupt. If enabled, RTCF will wake up CPU in Idle mode or power-down mode.

12 System Reset

During reset, all I/O Registers are set to their initial values, and the program starts execution from the Reset Vector, 0000H, or ISP start address by OR setting. The CMT2380F17 has 7 sources of reset: power-on reset, external reset, software reset, illegal address reset, brown-out reset 0, brown-out reset 1 and WDT reset. Figure 12–1 shows the system reset source in CMT2380F17.

The following sections describe the reset happened source and corresponding control registers and indicating flags.

12.1 Reset Source

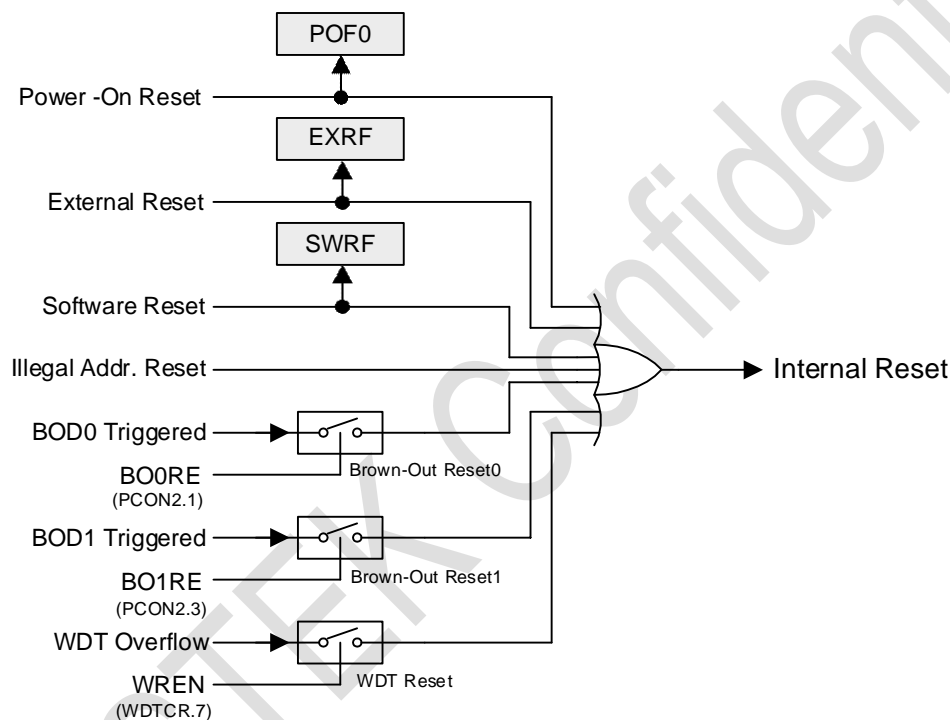


Figure 12-1. SystemResetSource

12.2 Power-On Reset (POR)

Power-on reset (POR) is used to internally reset the CPU during power-up. The CPU will keep in reset state and will not start to work until the VDD power rises above the voltage of Power-On Reset. And, the reset state is activated again whenever the VDD power falls below the POR voltage. During a power cycle, VDD must fall below the POR voltage before power is reapplied in order to ensure a power-on reset.

- **PCON0**: Power Control Register 0

SFR Page = 0~F & P

SFR Address = 0x87 POR = 0001-0000, 复位值= 000X-0000

Bit	7	6	5	4	3	2	1	0
Name	SMOD[1:0]		GF	POF0	GF[1:0]		PD	IDL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	1	0	0	0	0

Bit 4: POF0, Power-On Flag 0.

0: The flag must be cleared by software to recognize next reset type.

1: Set by hardware when VDD rises from 0 to its nominal voltage. POF0 can also be set by software.

The Power-on Flag, POF0, is set to “1” by hardware during power up or when VDD power drops below the POR voltage. It can be clear by firmware and is not affected by any warm reset such as external reset, Brown-Out reset, software reset (ISPCR.5) and WDT reset. It helps users to check if the running of the CPU begins from power up or not. Note that the POF0 must be cleared by firmware.

Note: POF0 must be cleared by software.

12.3 External Reset

A reset is accomplished by holding the RESET pin HIGH for a period of high time, please reference “33.3 External Reset Characteristics”. To ensure a reliable power-up reset, the hardware reset from RST pin is necessary.

- **PCON1**: Power Control Register 1

SFR Page = 0~F & P

SFR Address = 0x97

Bit	7	6	5	4	3	2	1	0
Name	SWRF	EXRF	--	RTCF	--	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	W	R/W	R/W	R/W
Reset Value	0	0	0	0	X	0	0	0

Bit 6: EXRF, External Reset Flag.

0: This bit must be cleared by software writing “1” on it. Software writing “0” is no operation.

1: This bit is only set by hardware if an External Reset occurs. Writing “1” on this bit will clear EXRF.

12.4 Software Reset

Software can trigger the CPU to restart by software reset, writing “1” on SWRST (ISPCR.5), and set the SWRF flag (PCON1.7). SWBS decides the CPU is boot from ISP or AP region after the reset action.

- ISPCR: ISP Control Register
SFR Page = 0~F
SFR Address = 0xE7

Bit	7	6	5	4	3	2	1	0
Name	ISPEN	SWBS	SWRST	CFAIL	--	--	--	--
R/W	R/W	R/W	R/W	R/W	W	W	W	W
Reset Value	0	0	0	0	X	X	X	X

Bit 6: SWBS, software boot selection control.
0: Boot from AP-memory after reset.
1: Boot from ISP memory after reset.

Bit 5: SWRST, software reset trigger control.
0: Write “0” is no operation
1: Write “1” to generate software system reset. It will be cleared by hardware automatically.

- PCON1: Power Control Register 1
SFR Page = 0~F & P
SFR Address = 0x97

Bit	7	6	5	4	3	2	1	0
Name	SWRF	EXRF	--	RTCF	--	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	W	R/W	R/W	R/W
Reset Value	0	0	0	0	X	0	0	0

Bit 7: SWRF, Software Reset Flag.
0: This bit must be cleared by software writing “1” on it. Software writing “0” is no operation.
1: This bit is only set by hardware if a Software Reset occurs. Writing “1” on this bit will clear SWRF.

12.5 Brown-Out Reset

In CMT2380F17, there are two Brown-Out Detectors (BOD0 & BOD1) to monitor VDD power. BOD0 services the fixed detection level at VDD=1.7V. BOD1 detects the VDD level by software selecting 4.2V, 3.7V, 2.4V or 2.0V. If VDD power drops below BOD0 or BOD1 monitor level. Associated flag, BOF0 and BOF1, is set. If BO0RE (PCON2.1) is enabled, BOF0 indicates a BOD0 Reset occurred. If BO1RE (PCON2.3) is enabled, BOF1 indicates a BOD1 Reset occurred.

- PCON1: Power Control Register 1
SFR Page = 0~F & P
SFR Address = 0x97

Bit	7	6	5	4	3	2	1	0
Name	SWRF	EXRF	MCDF	RTCF	--	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	W	R/W	R/W	R/W
Reset Value	0	0	0	0	X	0	0	0

Bit 2: BOF1, BOF1 (Reset) Flag.
0: This bit must be cleared by software writing “1” on it. Software writing “0” is no operation.
1: This bit is only set by hardware when VDD meets BOD1 monitored level. Writing “1” on this bit will clear BOF1. If BO1RE (PCON2.3) is enabled, BOF1 indicates a BOD1 Reset occurred.

Bit 1: BOF0, BOF0 (Reset) Flag.

0: This bit must be cleared by software writing “1” on it. Software writing “0” is no operation.

1: This bit is only set by hardware when VDD meets BOD0 monitored level. Writing “1” on this bit will clear BOF0. If BO0RE (PCON2.1) is enabled, BOF0 indicates a BOD0 Reset occurred.

12.6 WDT Reset

When WDT is enabled to start the counter, WDTF will be set by WDT overflow. If WREN (WDTCR.7) is enabled, the WDT overflow will trigger a system reset that causes CPU to restart. Software can read the WDTF to recognize the WDT reset occurred.

- PCON1: Power Control Register 1

SFR Page = 0~F & P

SFR Address = 0x97

Bit	7	6	5	4	3	2	1	0
Name	SWRF	EXRF	--	RTCF	--	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	W	R/W	R/W	R/W
Reset Value	0	0	0	0	X	0	0	0

Bit 0: WDTF, WDT Overflow/Reset Flag.

0: This bit must be cleared by software writing “1” on it. Software writing “0” is no operation.

1: This bit is only set by hardware when WDT overflows. Writing “1” on this bit will clear WDTF. If WREN (WDTCR.7) is set, WDTF indicates a WDT Reset occurred.

12.7 Illegal Address Reset

In CMT2380F17, if software program runs to illegal address such as over program ROM limitation, it triggers a RESET to CPU.

13 Power Management

The CMT2380F17 supports two power monitor modules, Brown-Out Detector 0 (BOD0) and Brown-Out Detector 1 (BOD1), and 7 power-reducing modes: Idle mode, Power-down mode, Slow mode, Sub-Clock mode, RTC mode, Watch mode and Monitor mode.

BOD0 and BOD1 report the chip power status on the flags, BOF0 and BOF1, which provide the capability to interrupt CPU or to reset CPU by software configured. The seven power-reducing modes provide the different power-saving scheme for chip application. These modes are accessed through the CKCON0, CKCON2, CKCON3, CKCON4, CKCON5, PCON0, PCON1, PCON2, PCON3, RTCCR and WDTCR register.

13.1 Brown-Out Detector

In CMT2380F17, there are two Brown-Out Detectors (BOD0 & BOD1) to monitor VDD power. Figure 13–1 shows the functional diagram of BOD0 and BOD1. BOD0 services the fixed detection level at VDD=1.7V and BOD1 detects the software selection levels (4.2V/3.7V/2.4V/2.0V) on VDD. Associated flag, BOF0 (PCON1.1), is set when BOD0 meets the detection level. If both of ESF (EIE1.3) and BOF0IE (SFIE.1) are enabled, a set BOF0 will generate a system flag interrupt. It can interrupt CPU either CPU in normal mode or idle mode. The BOD1 has the same flag function, BOF1, and same interrupt function. The BOD1 interrupt also wakes up CPU in power down mode if AWBOD1 (PCON2.7) is enabled.

If BO0RE (PCON2.1) is enabled, the BOD0 event will trigger a system reset and set BOF0 to indicate a BOD0 Reset occurred. The BOD0 reset restart the CPU either CPU in normal mode or idle mode. BOD1 also has the same reset capability with associated control bit, BO1RE (PCON2.3). The BOD1 reset also restart CPU in power down mode if AWBOD1 (PCON2.7) is enabled in BOD1 reset operation.

To reduce power consumption, software may clear EBOD1 (PCON2.2) to disable BOD1 if the BOD1 is not applied in user application.

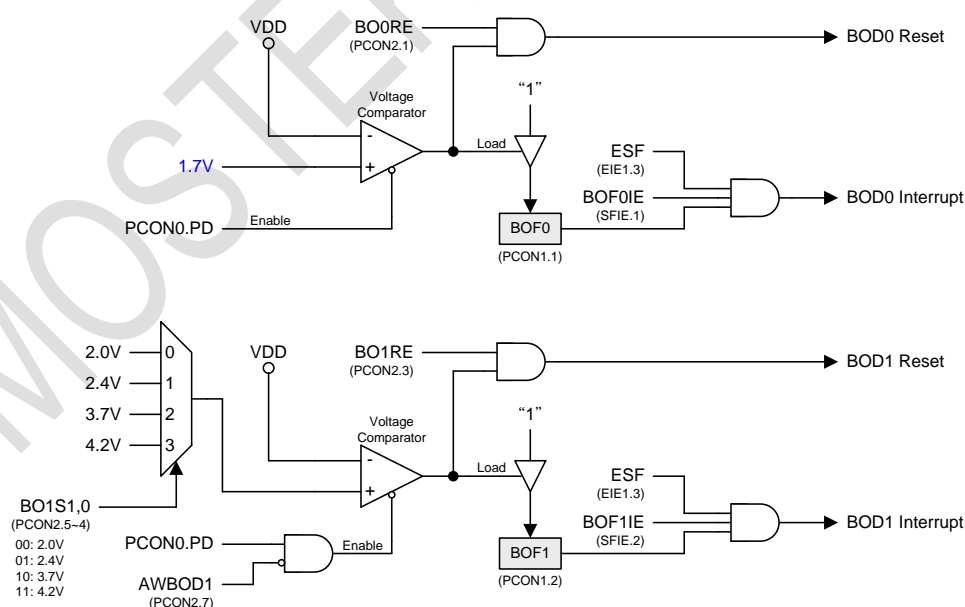


Figure 12-1. BOD0 and BOD1

13.2 Power Saving Mode

13.2.1 Slow Mode

The alternative to save the operating power is to slow the MCU's operating speed by programming SCKS2~SCKS0 bits (in CKCON0 register, see Section "9 System Clock") to a non-0/0/0 value. The user should examine which program segments are suitable for lower operating speed. In principle, the lower operating speed should not affect the system's normal function. Then, restore its normal speed in the other program segments.

13.2.2 Sub-Clock Mode

The alternative to slow down the MCU's operating speed by programming OSCS1~0 can select the ILRCO for system clock. The 32KHz ILRCO provides the MCU to operate in an ultra-low speed and low power operation. Additional programming SCKS2~SCKS0 bits (in CKCON0 register, see Section "9 System Clock"), the user could put the MCU speed down to 250Hz slowest.

13.2.3 RTC Mode

The CMT2380F17 has a simple RTC module that allows a user to continue running an accurate timer while the rest of the device is powered-down. In RTC mode, the RTC module behaves a "Clock" function and can be a wake-up source from chip power down by RTC overflow rate. Please refer Section "11 Real-Time-Clock (RTC)/System-Timer" for more detail information.

13.2.4 Watch Mode

If Watch-Dog-Timer is enabled and NSW is set, Watch-Dog-Timer will keep running in power down mode to support an auto-wakeup function, which named Watch Mode in CMT2380F17. When WDT overflows, set WDTF and wakeup CPU from interrupt or system reset by software configured. The maximum wakeup period is about 2 seconds that is defined by WDT pre-scaler. Please refer Section "10 Watch Dog Timer (WDT)" and Section "15 Interrupt" for more detail information.

13.2.5 Monitor Mode

If AWBOD1 (PCON2.3) is set, BOD1 will keep VDD monitor in power down mode. It is the Monitor Mode in CMT2380F17. When BOD1 meets the detection level, set BOF1 and wakeup CPU from interrupt or system reset by software configured. Please refer Section "13.1 Brown-Out Detector" and Section "15 Interrupt" for more detail information.

13.2.6 Idle Mode

可 Setting the IDL bit in PCON enters idle mode. Idle mode halts the internal CPU clock. The CPU state is preserved in its entirety, including the RAM, stack pointer, program counter, program status word, and accumulator. The Port pins hold the logical states they had at the time that Idle was activated. Idle mode leaves the peripherals running in order to allow them to wake up the CPU when an interrupt is generated. Timer 0, Timer 1, Timer 2, Timer 3, DMA, SPI, KBI, ADC, S0, S1, TWI0/I2C0, RTC, MCD, BOD0 and BOD1 will continue to function during Idle mode. PCA Timer and WDT are conditional enabled during Idle mode to wake up CPU. Any enabled interrupt source or reset may terminate Idle mode. When exiting Idle mode with an interrupt, the interrupt will immediately be serviced, and following RETI, the next instruction to be executed will be the one following the instruction that put the device into Idle.

Note: When the MCU is in idle mode and power-down mode, the ADC input channel must be set to "analog input only".

13.2.7 Power-down Mode

Setting the PD bit in PCON0 enters Power-down mode. Power-down mode stops the oscillator and powers down the Flash memory in order to minimize power consumption. Only the power-on circuitry will continue to draw power during Power-down. During Power-down the power supply voltage may be reduced to the RAM keep-alive voltage. The RAM contents will be retained; however, the SFR contents are not guaranteed once VDD has been reduced. Power-down may be exited by external reset, power-on reset, enabled external interrupts, enabled KBI, enabled RTC (RTC mode), enabled BOD1 (monitor mode) or enabled Non-Stop WDT (watch mode).

The user should not attempt to enter (or re-enter) the power-down mode for a minimum of 4 μ s until after one of the following conditions has occurred: Start of code execution (after any type of reset), or Exit from power-down mode. To ensure minimum power consumption in power down mode, software must confirm all I/O not in floating state.

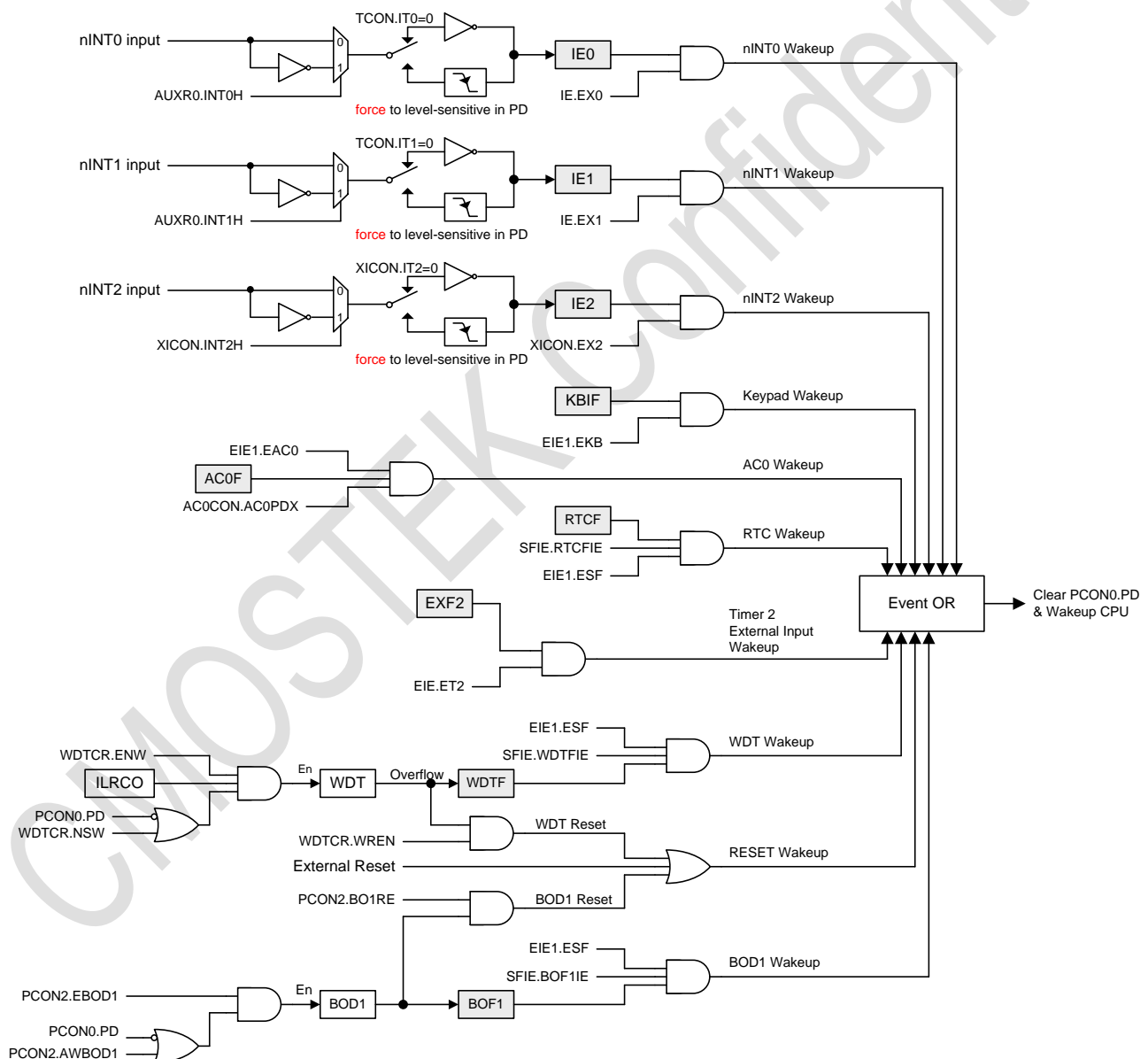


Figure 13-2. Wakeup structure of Power Down mode of CMT2380F17

13.2.8 Interrupt Recovery from Power-down

3 external interrupts may be configured to terminate Power-down mode. External interrupts nINT0, nINT1, nINT2 may be used to exit Power-down. To wake up by external interrupt nINT0, nINT1, nINT2, the interrupt must be enabled and configured for level-sensitive operation. If the enabled external interrupts are configured to edge-sensitive operation (Falling or Rising), they will be forced to level-sensitive operation (Low level or High level) by hardware in power-down mode.

When terminating Power-down by an interrupt, the wake up period is internally timed. At the falling edge on the interrupt pin, Power-down is exited, the oscillator is restarted, and an internal timer begins counting. The internal clock will not be allowed to propagate and the CPU will not resume execution until after the timer has reached internal counter full. After the timeout period, the interrupt service routine will begin. To prevent the interrupt from re-triggering, the ISR should disable the interrupt before returning. The interrupt pin should be held low until the device has timed out and begun executing.

13.2.9 Reset Recovery from Power-down

Wakeup from Power-down through an external reset is similar to the interrupt. At the rising edge of RST, Power-down is exited, the oscillator is restarted, and an internal timer begins counting. The internal clock will not be allowed to propagate to the CPU until after the timer has reached internal counter full. The RST pin must be held high for longer than the timeout period to ensure that the device is reset properly. The device will begin executing once RST is brought low.

It should be noted that when idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

13.2.10 KBI wakeup Recovery from Power-down

The Keypad Interrupt of CMT2380F17, KBI.7~0 have wakeup CPU capability that are enabled by the control registers in KBI module. OR software can configure the KBI inputs on different port pins. Please refer Section “30 Auxiliary SFRs” for more detailed AUXR6 information.

Wakeup from Power-down through an enabled wakeup KBI is same to the interrupt. At the matched condition of enabled KBI pattern and enabled KBI interrupt (EIE1.5, EKB), Power-down is exited, the oscillator is restarted, and an internal timer begins counting. The internal clock will not be allowed to propagate to the CPU until after the timer has reached internal counter full. After the timeout period, CPU will meet a KBI interrupt and execute the interrupt service routine.

13.3 Power Control Register

- **PCON0**: Power Control Register 0

SFR Page = 0~F & P

SFR Address = 0x87

POR = 0001-0000, RESET = 000X-0000

Bit	7	6	5	4	3	2	1	0
Name	SMOD[1:0]		GF	POF0	GF[1:0]		PD	IDL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	1	0	0	0	0

Bit 4: POF0, Power-On Flag 0.

0: This bit must be cleared by software writing one to it. 1: This bit is set by hardware if a Power-On Reset occurs.

Bit 1: PD, Power-Down control bit.

0: This bit could be cleared by CPU or any exited power-down event. 1: Setting this bit activates power down operation.

Bit 0: IDL, Idle mode control bit.

0: This bit could be cleared by CPU or any exited Idle mode event. 1: Setting this bit activates idle mode operation.

- **PCON1**: Power Control Register 1

SFR Page = 0~F & P

SFR Address = 0x97

Bit	7	6	5	4	3	2	1	0
Name	SWRF	EXRF	--	RTCF	--	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	W	R/W	R/W	R/W
Reset Value	0	0	0	0	X	0	0	0

Bit 7: SWRF, Software Reset Flag.

0: This bit must be cleared by software writing "1" to it. 1: This bit is set by hardware if a Software Reset occurs.

Bit 6: EXRF, External Reset Flag.

0: This bit must be cleared by software writing "1" to it. 1: This bit is set by hardware if an External Reset occurs.

Bit 4: RTCF, RTC overflow flag.

0: This bit must be cleared by software writing "1" on it. Software writing "0" is no operation.

1: This bit is only set by hardware when RTCCT overflows. Writing "1" on this bit will clear RTCF.

Bit 3: Reserved. Software must write "0" on this bit when PCON1 is written.

Bit 2: BOF1, Brown-Out Detection flag 1.

0: This bit must be cleared by software writing "1" to it.

1: This bit is set by hardware if the operating voltage matches the detection level of Brown-Out Detector 1 (4.2V/3.7/2.4/2.0).

Bit 1: BOF0, Brown-Out Detection flag 0.

0: This bit must be cleared by software writing "1" to it.

1: This bit is set by hardware if the operating voltage matches the detection level of Brown-Out Detector 0 (1.7V).

Bit 0: WDTF, WDT overflow flag.

0: This bit must be cleared by software writing "1" to it. 1: This bit is set by hardware if a WDT overflow occurs.

- **PCON2:** Power Control Register 2
SFR Page = P Only
SFR Address = 0x44

Bit	7	6	5	4	3	2	1	0
Name	AWBOD1	0	BO1S[1:0]		BO1RE	EBOD1	BO0RE	1
R/W	R/W	W	R/W	R/W	R/W	R/W	R/W	W
Reset Value	0	0	0	0	0	1	0	1

Bit 7: AWBOD1, Awaked BOD1 in PD mode.
0: BOD1 is disabled in power-down mode.
1: BOD1 keeps operation in power-down mode.

Bit 6: Reserved. Software must write "0" on this bit when PCON2 is written.

Bit 5~4: BO1S[1:0]. Brown-Out detector 1 monitored level Selection..

BO1S[1:0]	BOD1 detecting level
0 0	2.0V
0 1	2.4V
1 0	3.7V
1 1	4.2V

Bit 3: BO1RE, BOD1 Reset Enabled.

0: Disable BOD1 to trigger a system reset when BOF1 is set. 1: Enable BOD1 to trigger a system reset when BOF1 is set.

Bit 2: EBOD1, Enable BOD1 that monitors VDD power dropped at a BO1S1~0 specified voltage level. 0: Disable BOD1 to slow down the chip power consumption.
1: Enable BOD1 to monitor VDD power dropped.

Bit 1: BO0RE, BOD0 Reset Enabled.

0: Disable BOD0 to trigger a system reset when BOF0 is set.
1: Enable BOD0 to trigger a system reset when BOF0 is set (VDD meets 1.7V).

Bit 0: Reserved. Software must write "1" on this bit when PCON2 is written.

- **PCON3:** Power Control Register 3

SFR Page = P Only

SFR Address = 0x45

POR = 0xxx-xxxx

Bit	7	6	5	4	3	2	1	0
Name	IVREN	--	--	--	--	--	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	X	X	X	X	X	X	X

Bit 7: IVREN, Internal Voltage Reference Enable. 0: Disable on-chip IVR (1.4V).

1: Enable on-chip IVR (1.4V).

Bit 6~0: Reserved. Software must write "0" on these bits when PCON3 is written.

14 Configurable I/O Ports

The CMT2380F17 has following I/O ports: P0.0~P0.7, P1.0~P1.7, P2.0~P2.7, P3.0~P3.7, P4.0~P4.7 and P6.0~P6.3/P6.4. If disable external reset function, P4.7 function is valid. The exact number of I/O pins available depends upon the package types. See Table 14 – 1.

14.1 IO Structure

The I/O operating modes are distinguished two groups in CMT2380F17. The first group is only for Port 3 to support four configurations on I/O operating. These are: quasi-bidirectional (standard 8051 I/O port), push-pull output, input-only (high-impedance input) and open-drain output. The Port 3 default setting is quasi-bidirectional mode with weakly pull-up resistance.

All other general port pins belong to the second group. They can be programmed to four operating modes, which include analog input only, open-drain output with pull-up resistor, open-drain output and push-pull output. The default setting of this group I/O is analog input only, which means the port pin in high impedance state.

Following sections describe the configuration of the all types I/O mode.

14.1.1 Port 3 Quasi-Bidirectional IO Structure

Port 3 pins in quasi-bidirectional mode are similar to the standard 8051 port pins. A quasi-bidirectional port can be used as an input and output without the need to reconfigure the port. This is possible because when the port outputs a logic high, it is weakly driven, allowing an external device to pull the pin low. When the pin outputs low, it is driven strongly and able to sink a large current. There are three pull-up transistors in the quasi-bidirectional output that serve different purposes.

One of these pull-ups, called the “very weak” pull-up, is turned on whenever the port register for the pin contains a logic “1”. This very weak pull-up sources a very small current that will pull the pin high if it is left floating. A second pull-up, called the “weak” pull-up, is turned on when the port register for the pin contains a logic “1” and the pin itself is also at a logic “1” level. This pull-up provides the primary source current for a quasi-bidirectional pin that is outputting a 1. If this pin is pulled low by the external device, this weak pull-up turns off, and only the very weak pull-up remains on. In order to pull the pin low under these conditions, the external device has to sink enough current to over-power the weak pull-up and pull the port pin below its input threshold voltage. The third pull-up is referred to as the “strong” pull-up. This pull-up is used to speed up low-to-high transitions on a quasi-bidirectional port pin when the port register changes from a logic “0” to a logic “1”. When this occurs, the strong pull-up turns on for one CPU clocks, quickly pulling the port pin high.

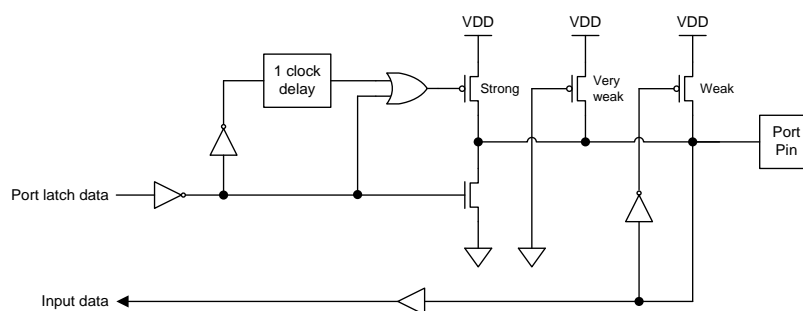


Figure 14-1. Port 3 Quasi-Bidirectional I/O

14.1.2 Port 3 Push-Pull Output Structure

The push-pull output configuration on Port 3 has the same pull-down structure as both the open-drain and the quasi-bidirectional output modes, but provides a continuous strong pull-up when the port register contains a logic “1”. The push-pull mode may be used when more source current is needed from a port output. In addition, the input path of the port pin in this configuration is also the same as quasi-bidirectional mode. The push-pull port configuration is shown in Figure 14–2.

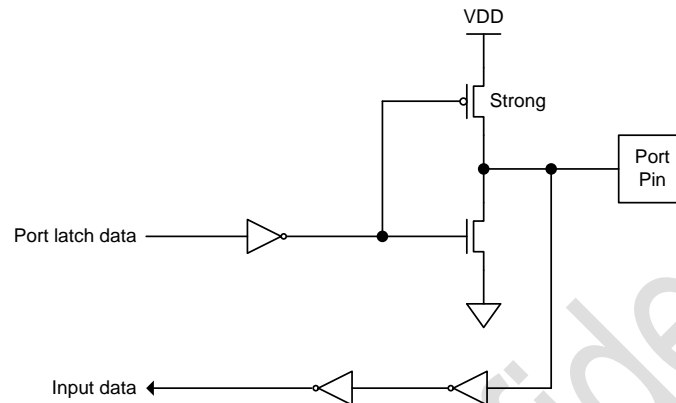


Figure 14-2. Port 3 Push-Pull Output

14.1.3 Port 3 Input-Only (High Impedance Input) Structure

The input-only configuration on Port 3 is an input without any pull-up resistors on the pin, as shown in Figure 14–3.

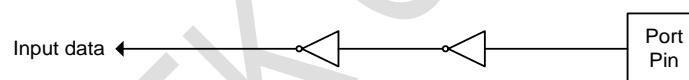


Figure 14-3. Port 3 Input-Only

14.1.4 Port 3 Open-Drain Output Structure

The open-drain output configuration on Port 3 turns off all pull-ups and only drives the pull-down transistor of the port pin when the port register contains a logic “0”. To use this configuration in application, a port pin must have an external pull-up, typically a resistor tied to VDD. The pull-down for this mode is the same as for the quasi-bidirectional mode. In addition, the input path of the port pin in this configuration is also the same as quasi-bidirectional mode.

The open-drain port configuration is shown in Figure 14–4.

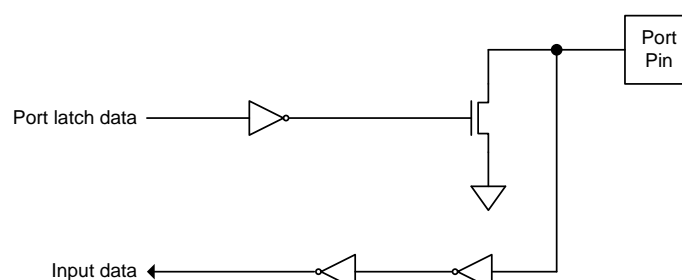
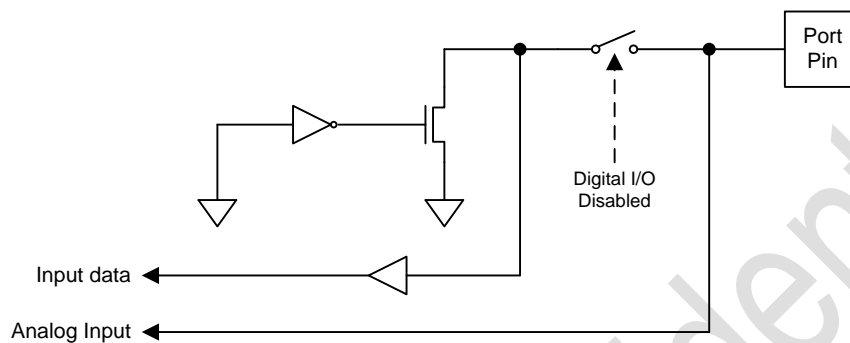


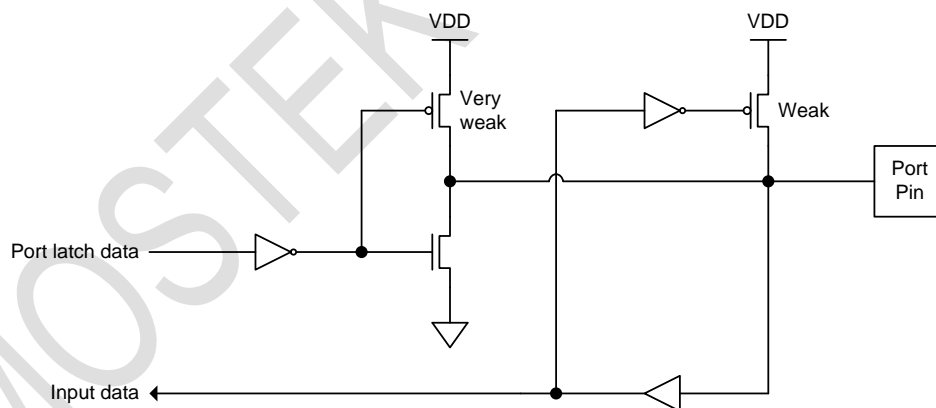
Figure 14-4. Port 3 Open-Drain Output**14.1.5 General Analog Input Only Structure**

The analog-input-only configuration on general port pins is the default setting. For ADC or Analog Comparator input application, user may keep the port setting in this configuration. If apply the port pin to digital function, user must program the port pin to associated configuration. The analog-input-only port configuration is shown in Figure 14–5.

**Figure 14-5. General Analog-Input-Only****14.1.6 General Open-Drain Output with Pull-up Resistor Structure**

The open-drain output with pull-up resistor configuration on general port pins enables the on-chip pull-up resistor in open-drain output mode.

The open-drain output with pull-up resistor port configuration is shown in Figure 14–6.

**Figure 14-6. General Open-Drain output with Pull-up Resistor****14.1.7 General Open-Drain Output Structure**

The open-drain output configuration on general port pins is the same function as port 3 open-drain output mode. The general open-drain port configuration is shown in Figure 14–7.

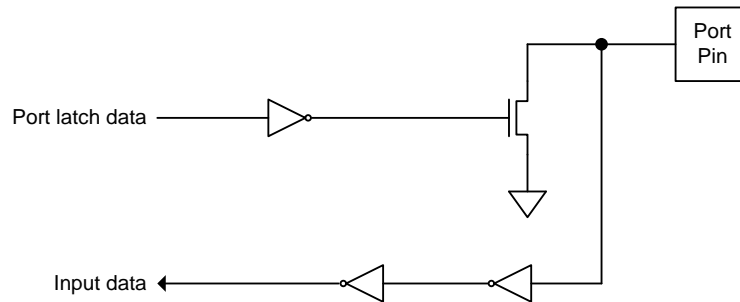


Figure 14-7. General Open-Drain Output

14.1.8 General Port Digital Input Configured

A Port pin is configured as a digital input by setting its output mode to “Open-Drain” and writing a logic “1” to the associated bit in the Port Data register. For example, P1.0 is configured as a digital input by setting P1M0.0 to a logic 0, P1M1.0 to a logic 0 and P1.0 to a logic 1.

14.1.9 General Push-Pull Output Structure

The push-pull output configuration on general port pins has the same function with port 3 push-pull output mode. The push-pull port configuration is shown in Figure 14–8.

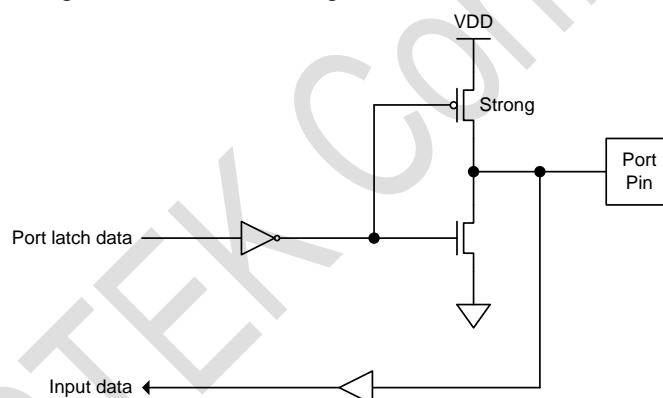


Figure 14-8. General Push-Pull Output

14.1.10 Port Pin Output Driving Strength Selection

The I/O of the CMT2380F17 has two driving strength can be selected for different kinds of the application to match the output impedance. Please reference 14.2.6 Port Output Driving Strength Control Register.。

14.1.11 Port Pin Output Fast Driving Selection

The I/O of the CMT2380F17 has two driving speed can be selected for different kinds of the I/O frequency. Please reference 14.2.7 Port Output Fast Driving Control Register.

14.2 I/O Port Register

All I/O port pins on the CMT2380F17 may be individually and independently configured by software to select its operating modes. Port 3 has four operating modes, as shown in Table 14–2. Two mode registers select the output type for each port 3 pin. Only Port 3 supports quasi-bidirectional mode and setting them to quasi-bidirectional mode after system reset.

Figure 14-1. Port 3 Configuration Settings

P3M0.y	P3M1.y	Port Mode
0	0	Quasi-Bidirectional (default)
0	1	Push-Pull Output
1	0	Input Only (High Impedance Input)
1	1	Open-Drain Output

Note: Where y=0~7 (port pin). The registers P3M0 and P3M1 are listed in each port description.

Other general port pins also support four operating modes, as shown in Table 14-3. Two mode registers select the I/O type for each port pin and setting to analog-input-only on these port pins after system reset.

Figure 14-2. General Port Configuration Settings

PxM0.y	PxM1.y	Port Mode
0	1	Analog Input Only (default)
1	1	Open-Drain with Pull-up resistor
0	0	Open-Drain Output / General Digital Input (Port Pin set to "1")
1	0	Push-Pull Output

Note: Where x= 0, 1, 2, 4, 6 (port number), and y=0~7 (port pin). The registers PxM0 and PxM1 are listed in each port description.

14.2.1 Port 1 Register

- P1: Port 1 Register
SFR Page = 0~F
SFR Address = 0x90

Bit	7	6	5	4	3	2	1	0
Name	P1.7	P1.6	P1.5	--	--	--	P1.1	P1.0
R/W	R/W	R/W	R/W	W	W	W	R/W	R/W
Reset Value	1	1	1	1	1	1	1	1

Bit 7, 6, 5, 1,0: Port 1 output data latch could be only set/cleared by CPU.

- P1M0: Port 1 Mode Register 0
SFR Page = 0~F
SFR Address = 0x91

Bit	7	6	5	4	3	2	1	0
Name	P1M0.7	P1M0.6	P1M0.5	--	--	--	P1M0.1	P1M0.0
R/W	R/W	R/W	R/W	W	W	W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

- P1M1: Port 1 Mode Register 1
SFR Page = 0 only
SFR Address = 0x92

Bit	7	6	5	4	3	2	1	0
Name	P1M1.7	P1M1.6	P1M1.5	--	--	--	P1M1.1	P1M1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	1	1	1	1	1	1	1	1

14.2.2 Port 2 Register

- P2: Port 2 Register
SFR Page = 0~F
SFR Address = 0xA0

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	P2.4	--	P2.2	--	--
R/W	W	W	W	R/W	W	R/W	W	W
Reset Value	1	1	1	1	1	1	1	1

Bit 4, 2: Port 2 output data latch could be only set/cleared by CPU.

- P2M0: Port 2 Mode Register 0
SFR Page = 0 only
SFR Address = 0x95

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	P2M0.4	--	P2M0.2	--	--
R/W	W	W	W	R/W	W	R/W	W	W
Reset Value	0	0	0	0	0	0	0	0

- P2M1: Port 2 Mode Register 0

SFR Page = 0 only

SFR Address = 0x95

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	P2M1.4	--	P2M1.2	--	--
R/W	W	W	W	R/W	W	R/W	W	W
Reset Value	1	1	1	1	1	1	1	1

14.2.3 Port 3 Register

- P3: Port 3 Register

SFR Page = 0~F

SFR Address = 0xB0

Bit	7	6	5	4	3	2	1	0
Name	--	--	P3.5	P3.4	P3.3	--	P3.1	P3.0
R/W	W	W	R/W	R/W	R/W	W	R/W	R/W
Reset Value	1	1	1	1	1	1	1	1

Bit 5, 4, 3, 1, 0: Port 3 output data is set/cleared by the CPU.

- P3M0: Port 3 Mode Register 0

SFR Page = 0~F

SFR Address = 0xB1

Bit	7	6	5	4	3	2	1	0
Name	--	--	P3M0.5	P3M0.4	P3M0.3	--	P3M0.1	P3M0.0
R/W	W	W	R/W	R/W	R/W	W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

- P3M1: Port 3 Mode Register 1

SFR Page = 0~F

SFR Address = 0xB2

Bit	7	6	5	4	3	2	1	0
Name	--	--	P3M1.5	P3M1.4	P3M1.3	--	P3M1.1	P3M1.0
R/W	W	W	R/W	R/W	R/W	W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

- AUXR11: Auxiliary Register 11
SFR Page = 8 only
SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	P30AM	--	--	--	--	--	COM0	COOFS
R/W	W	W	W	W	W	W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: P30AM, P3.0 Analog input Mode enable.

0: The P3.0 GPIO mode is controlled by P3M0 and P3M1.

1: Force P3.0 to be analog input mode for the AIN4 input of ADC12.

14.2.4 Port 4 Register

- P4: Port 4 Mode Register 0
SFR Page = 0 only
SFR Address = 0xB3

Bit	7	6	5	4	3	2	1	0
Name	P4.7	--	P4.5	P4.4	--	--	--	--
R/W	R/W	W	R/W	R/W	W	W	W	W
Reset Value	1	X	1	1	X	X	1	1

Bit 7~0: Port 4 output data latch could be set/cleared by CPU.

P4.5 and P4.4 have the alternated function for OCD_SDA and OCD_SCL.

P4.7 has the alternated function for RST input.

- P4M0: Port 4 Mode Register 0
SFR Page = 0 only
SFR Address = 0xB3

Bit	7	6	5	4	3	2	1	0
Name	P4M0.7	--	P4M0.5	P4M0.4	--	--	--	--
R/W	R/W	W	R/W	R/W	W	W	W	W
Reset Value	1	0	1	1	0	0	0	0

Note: When P4.7/RST use as port pin, it is not suggest to program it as Input to avoid MCU is locked in reset in bootup period when level high send into this pin.

- P4M1: Port 4 Mode Register 1

SFR Page = 2 only

SFR Address = 0x92

Bit	7	6	5	4	3	2	1	0
Name	P4M1.7	--	P4M1.5	P4M1.4	--	--	--	--
R/W	R/W	W	R/W	R/W	W	W	W	W
Reset Value	1	1	1	1	1	1	1	1

14.2.5 Port 6 Register

- P6: Port 6 Register

SFR Page = 1 only

SFR Address = 0xF8

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	--	--	--	P6.1	P6.0
R/W	W	W	W	W	W	W	R/W	R/W
Reset Value	X	X	X	1	1	1	1	1

Bit 1~0: Port 6 output data latch could be only set/cleared by CPU.

- P6M0: Port 6 Mode Register 0

SFR Page = 1 only

SFR Address = 0xB5

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	--	--	--	P6M0.1	P6M0.0
R/W	W	W	W	W	W	W	R/W	R/W
Reset Value	X	X	X	0	0	0	0	0

- P6M1: Port 6 Mode Register 1

SFR Page = 3 only

SFR Address = 0x92

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	--	--	--	P6M1.1	P6M1.0
R/W	W	W	W	W	W	W	R/W	R/W
Reset Value	1	1	1	1	1	1	1	1

14.2.6 Port Output Driving Strength Control Register

In CMT2380F17, all port pins have two driving strength selection by software configured except P4.7, P6.1 and P6.0. Please refer to get the driving strength information on the port pins.

- PDRVC0: Port Drive Control Register 0

SFR Page = 2 only

SFR Address = 0xB4

Bit	7	6	5	4	3	2	1	0
Name	P3DC1	P3DC0	P2DC1	P2DC1	P1DC1	P1DC0	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	W	W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: P3DC1, Port 3 output driving strength control on high nibble.

0: Select the P3.7 ~ P3.4 output with high driving strength.

1: Select the P3.7 ~ P3.4 output with low driving strength.

Bit 6: P3DC0, Port 3 output driving strength control on low nibble.

0: Select the P3.3 ~ P3.0 output with high driving strength.

1: Select the P3.3 ~ P3.0 output with low driving strength.

Bit 5: P2DC1, Port 2 output driving strength control on high nibble.

0: Select the P2.7 ~ P2.4 output with high driving strength.

1: Select the P2.7 ~ P2.4 output with low driving strength.

Bit 4: P2DC0, Port 2 output driving strength control on low nibble.

0: Select the P2.3 ~ P2.0 output with high driving strength.

1: Select the P2.3 ~ P2.0 output with low driving strength.

Bit 3: P1DC1, Port 1 output driving strength control on high nibble.

0: Select the P1.7 ~ P1.4 output with high driving strength.

1: Select the P1.7 ~ P1.4 output with low driving strength.

Bit 2: P1DC0, Port 1 output driving strength control on low nibble.

0: Select the P1.3 ~ P1.0 output with high driving strength.

1: Select the P1.3 ~ P1.0 output with low driving strength.

Bit 1~0: Reserved bits. When writing the PDRVC0 register, the software must write "0" to these bits.

- PDRVC1: Port Drive Control Register 1
SFR Page = 3 only
SFR Address = 0xB4

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	--	--	--	P4DC1	--
R/W	W	W	W	W	W	W	R/W	R/W
Reset Value	X	X	X	0	X	X	0	0

Bit 7~2: Reserved. Software must write "0" on these bits when PDRVC1 is written.

Bit 1: P4DC1, Port 4 output driving strength control on high nibble.

0: Select the P4.6 ~ P4.4 output with high driving strength.

1: Select the P4.6 ~ P4.4 output with low driving strength.

Bit 0: reserved bit. When writing the PDRVC1 register, the software bit must write to "0".

14.2.7 Port Output Fast Driving Control Register

In CMT2380F17, all port pins have two driving speed selection by software configured except P4.7. Please refer to get the driving strength information on the port pins.

- P3FDC: Port 3 Fast Driving Control Register
SFR Page = 7 only
SFR Address = 0x92

Bit	7	6	5	4	3	2	1	0
Name	--	--	P3FDC.5	P3FDC.4	P3FDC.3	--	P3FDC.1	P3FDC.0
R/W	W	W	R/W	R/W	R/W	W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~0: Port 3 output fast driving control could be only set/cleared by CPU. 0: Disable fast driving on port pin output.

1: Enable fast driving on port pin output.

- P1FDC: Port 1 Fast Driving Control Register
SFR Page = 8 only
SFR Address = 0x92

Bit	7	6	5	4	3	2	1	0
Name	P1FDC.7	P1FDC.6	P1FDC.5	--	--	--	P1FDC.1	P1FDC.0
R/W	R/W	R/W	R/W	W	W	W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~0: Port 1 output fast driving control could be only set/cleared by CPU. 0: Disable fast driving on port pin output.

1: Enable fast driving on port pin output.

- P2FDC: Port 2 Fast Driving Control Register
SFR Page = 9 only
SFR Address = 0x92

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	P2FDC.4	--	P2FDC.2	--	--
R/W	W	W	W	R/W	W	R/W	W	W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~0: Port 2 output fast driving control could be only set/cleared by CPU. 0: Disable fast driving on port pin output.

1: Enable fast driving on port pin output.

- P4FDC: Port 4 Fast Driving Control Register
SFR Page = A only
SFR Address = 0x92

Bit	7	6	5	4	3	2	1	0
Name	--	--	P4FDC.5	P4FDC.4	--	--	--	--
R/W	W	W	R/W	R/W	W	W	W	W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~0: Port 4 output fast driving control could be only set/cleared by CPU. 0: Disable fast driving on port pin output.

1: Enable fast driving on port pin output.

14.3 Port Function Redirection

Many I/O pins, in addition to their normal I/O function, also serve the alternate function for internal peripherals. For the digital peripherals, all GPIOs serve the alternate function in the default state. However, the user may set the corresponding control bits in AXUR0~AUXR3 to serve their alternate function on the relocated ports.

- AUXR0: Auxiliary Register 0
SFR Page = 0~F
SFR Address = 0xA1

Bit	7	6	5	4	3	2	1	0
Name	P60OC1	P60OC0	P60FD	PBKF	--	--	INT1H	INT0H
R/W	R/W	R/W	R/W	R/W	W	W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~6: P6.0 function configured control bit 1 and 0. The two bits only act when internal RC oscillator (IHRCO or ILRCO) is selected for system clock source. In external clock input mode, P6.0 is the dedicated clock input pin. In internal oscillator condition, P6.0 provides the following selections for GPIO or clock source generator. When P60OC[1:0] index to non-P6.0 GPIO function, P6.0 will drive the on-chip RC oscillator output to provide the clock source for other devices.

P60OC[1:0]	P60function	I/Omode
00	P60	By P6M0.0
01	MCK	By P6M0.0
10	MCK/2	By P6M0.0
11	MCK/4	By P6M0.0

Please refer Section “9 System Clock” to get the more detailed clock information. For clock-out on P6.0 function, it is recommended to set P6M0.0 to “1” which selects P6.0 as push-push output mode.

Bit 5: P60FD, P6.0 Fast Driving. 0: P6.0 output with default driving.

1: P6.0 output with fast driving enabled. If P6.0 is configured to clock output, enable this bit when P6.0 output frequency is more than 12MHz at 5V application or more than 6MHz at 3V application.

- AUXR3: Auxiliary Register 3

SFR Page = 0 only

SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	T0PS1	T0PS0	BPOC1	BPOC0	S0PS0	TWIPS1	TWIPS0	T0XL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~6: T0PS1~0, Timer 0 Port pin Selection [1:0].

T0PS1~0	T0/T0CKO
00	P3.4
01	P4.4
10	P2.2
11	P1.7

Bit 0: T0XL is the timer 0 prescaler control bit. For T0XL function definition, please refer to T0X12 (AUXR2.2).

Bit 3: S0PS0, Serial Port 0 pin Selection 0. (Add new S0PS1 at AUXR10.3)

S0PS1, S0PS0	RxD0	TxD0
00	P3.0	P3.1
01	P4.4	P4.5
10	P3.1	P3.0
11	P1.7	P2.2

Bit 2~1: TWIPS1~0, TWI0/I2C0 Port pin Selection [1:0].

TWIPS[1:0]	TWI0/I2C0_SCL	TWI0/I2C0_SDA
00	P3.1	P3.0
01	P6.0	P6.1
10	P3.0	P3.1
11	P2.2	P2.4

- AUXR4: Auxiliary Register 4

SFR Page = 1 only

SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	T2PS1	T2PS0	T1PS1	T1PS0	--	--	--	--
R/W	R/W	R/W	R/W	R/W	W	W	W	W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~6: T2PS1~0, Timer 2 Port pin Selection [1:0].

T2PS1~0	T2/T2CKO	T2EX
00	P1.0	P1.1
01	P3.0	P3.1
10	P6.0	P3.5
11	P4.5	P4.4

Bit 5~4: T1PS1~0, Timer 1 Port pin Selection [1:0].

T1PS1~0	T1/T1CKO
00	P3.5
01	P4.5
10	P1.7
11	P3.3

- AUXR5: Auxiliary Register 5
SFR Page = 2 only
SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	C0IC4S0	C0IC2S0	C0PPS1	C0PPS0	--	C0PS0	ECIPS0	C0COPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: C0IC4S0, PCA0 Input Channel 4 input port pin Selection.

C0IC4S0	CEX4Input
0	CEX4 _{port} pin
1	T2EXI

Bit 6: C0IC2S0, PCA0 Input Channel 2 input port pin Selection.

C0IC2S0	CEX2Input
0	CEX2 _{port} pin
1	T3EXI

Bit 5: C0PPS1, {PWM2A, PWM2B} Port pin Selection 1.

C0PPS1	PWM2A	PWM2B
0	P6.0	P6.1
1	P3.4	P3.5

Bit 4: C0PPS0, {PWM0A, PWM0B} Port pin Selection 0.

C0PPS0	PWM0A	PWM0B
0	P1.6	P1.7
1	P6.0	P6.1

Bit 3: Reserved.

Bit 2: C0PS0, PCA0 Port pin Selection 0.

C0PS0	CEX0	CEX2	CEX4
0	P2.2	P2.4	P1.7
1	P3.0	P2.4	P3.1

Bit1:ECIPS0,PCA0ECIPortpinSelection0.

ECIPS0	ECI
0	P4.4
1	P1.6

Bit0:C0COPS,PCA0Clock Output(C0CKO)portpinSelection.

C0COPS	C0CKO
0	P4.7
1	P3.3

- AUXR6: Auxiliary Register 6
SFR Page = 3 only
SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	KBI4PS1	KBI4PS0	KBI6PS0	KBI2PS0	T3FCS	T2FCS	SnMIPS	S0COPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~6: KBI4PS1~0, KBI4~5 Port pin Selection [1:0].

KBI4PS1~0	KBI4	KBI5
00	P3.3	P1.5
01	P3.4	P3.5
10	P6.0	P6.1
11	P1.5	P3.3

Bit 5: KBI6PS0, KBI6~7 Port pin Selection 0.

KBI6PS0	KBI6	KBI7
0	P1.6	P1.7
1	P3.0	P3.1

Bit 4: KBI2PS0, KBI2~3 Port pin Selection 0.

KBI2PS0	KBI2	KBI3
0	P3.0	P3.1
1	P2.2	P2.4

Bit 1: SnMIPS, S0MI & S1MI Port pin Selection.

SnMIPS	S0MI	S1MI
0	P1.6	P6.1
1	P3.3	P4.7

Bit 0: S0COPS, S0BRG Clock Output (S0CKO) port pin Selection.

S0COPS	S0CKO
0	P4.7
1	P3.3

- AUXR7: Auxiliary Register 7

SFR Page = 4 only

SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	POE5	POE4	C0CKOE	SPI0M0	--	--	--	--
R/W	R/W	R/W	R/W	R/W	W	W	W	W
Reset Value	1	1	0	0	0	0	0	0

Bit 7: POE5, PCA0 PWM5 main channel (PWM5O) output control.

0: Disable PWM5O output on port pin.

1: Enable PWM5O output on port pin. Default is enabled.

Bit 6: POE4, PCA0 PWM4 main channel (PWM4O) output control.

0: Disable PWM4O output on port pin.

1: Enable PWM4O output on port pin. Default is enabled.

Bit 5: C0CKOE, PCA0 clock output (C0CKO) enable.

0: Disable PCA0 clock output.

1: Enable PCA0 clock output with PCA0 base timer overflow rate/2.

- AUXR8: Auxiliary Register 8

SFR Page = 5 only

SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	POE7	POE6	C0PPS2	--	KBI0PS0	S1COPS	--	--
R/W	R/W	R/W	R/W	W	R/W	R/W	W	W
Reset Value	1	1	0	0	0	0	0	0

Bit 7: POE7, PCA0 PWM7 main channel (PWM7O) output control.

0: Disable PWM7O output on port pin.

1: Enable PWM7O output on port pin. Default is enabled.

Bit 6: POE6, PCA0 PWM6 main channel (PWM6O) output control.

0: Disable PWM6O output on port pin.

1: Enable PWM6O output on port pin. Default is enabled.

Bit 5: C0PPS2, {PWM6, PWM7} Port pin Selection 2.

C0PPS2	PWM6	PWM7
0	P6.0	P6.1
1	P3.0	P3.1

Bit 3: KBI0PS0, KBI0~1 Port pin Selection 0.

KBI0PS0	KBI0	KBI1
0	P1.0	P1.1
1	P4.7	P3.3

Bit 2: S1COPS, S1BRG Clock Output (S1CKO) port pin Selection.

S1COPS	S1CKO
0	P4.7
1	P6.1

- AUXR9: Auxiliary Register 9

SFR Page = 6 only

SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	--	--	T1G1	T0G1	C0FDC1	C0FDC0	S1PS1	S1PS0
R/W	W	W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 1~0: S1PS1~0, Serial Port 1 pin Selection [1:0].

S1PS1, S1PS0	RxD1	TxD1
00	P1.0	P1.1
01	P6.0	P6.1
10	P4.4	P4.5
11	P3.4	P3.5

- AUXR10: Auxiliary Register 10
SFR Page = 7 only
SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	SPIPS0	S0PS1	--	TWICF	PAA
R/W	W	W	W	R/W	R/W	W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 4: SPIPS0, SPI Port pin Selection 0.

SPIPS0	nSS	MOSI	MISO	SPICLK
0	P3.3	P1.5	P1.6	P1.7
1	P1.7	P3.5	P3.4	P3.3

- XICFG: External Interrupt Configured Register
SFR Page = 0 only
SFR Address = 0xC1

Bit	7	6	5	4	3	2	1	0
Name	INT1IS[1:0]		INT0IS[1:0]		--	X2FLT	X1FLT	X0FLT
R/W	R/W	R/W	R/W	R/W	W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~6: nINT1 input pin selection bits determined by INT1IS[2:0]. As shown in the table below.

INT1IS[2:0]	Selected Port Pin of nINT1
000	P3.3
001	P3.1
010	P3.5
011	P1.0
100	P6.1
101	P3.4
110	P1.5
111	P2.4

Bit 5~4: INT0IS.1~0, nINT0 input port pin selection bits which function is defined with INT0IS.2 as following table.

INT0IS[2:0]	Selected Port Pin of nINT0
000	P4.5
001	P3.0
010	P3.4
011	P4.7
100	P6.0
101	P1.1
110	P1.7
111	P2.2

- XICFG1: External Interrupt Configured 1 Register
SFR Page = 1 only
SFR Address = 0xC1

Bit	7	6	5	4	3	2	1	0
Name	INT1IS.2	INT0IS.2	INT2IS[1:0]		--	X2FLT1	X1FLT1	X0FLT1
R/W	R/W	R/W	R/W	R/W	W	R/W	R/W	R/W
Reset Value	0	0	0	0	X	0	0	0

Bit 7: INT1IS2, nINT1 input port pin selection bit which function is defined with INT1IS.1~0.

Bit 6: INT0IS2, nINT0 input port pin selection bit which function is defined with INT0IS.1~0.

Bit 5~4: INT2IS1~0, nINT2 input port pin selection bits which function is defined as following table.

INT2IS[1:0]	Selected Port Pin of nINT2
00	P4.4
01	P3.0
10	P1.1
11	P1.6

15 Interrupt

The CMT2380F17 has 16 interrupt sources with a four-level interrupt structure. There are several SFRs associated with the four-level interrupt. They are the IE, IP0L, IP0H, EIE1, EIP1L, EIP1H, EIE2, EIP2L, EIP2H and XICON. The IP0H (Interrupt Priority 0 High), EIP1H (Extended Interrupt Priority 1 High) and EIP2H (Extended Interrupt Priority 2 High) registers make the four-level interrupt structure possible. The four priority level interrupt structure allows great flexibility in handling these interrupt sources.

15.1 Interrupt Structure

Table 15–1 lists all the interrupt sources. The ‘Request Bits’ are the interrupt flags that will generate an interrupt if it is enabled by setting the ‘Enable Bit’. Of course, the global enable bit EA (in IE0 register) should have been set previously. The ‘Request Bits’ can be set or cleared by software, with the same result as though it had been set or cleared by hardware. That is, interrupts can be generated or pending interrupts can be cancelled in software. The ‘Priority Bits’ determine the priority level for each interrupt. The ‘Priority within Level’ is the polling sequence used to resolve simultaneous requests of the same priority level. The ‘Vector Address’ is the entry point of an interrupt service routine in the program memory.

Table 15-1. Interrupt Sources

No	SourceName	Enable Bit	Request Bits	Priority Bits	Polling Priority	Vector Address
#0	ExternalInterrupt0,nINT0	EX0	IE0	[PX0H, PX0L]	(Highest)	0003H
#1	Timer0	ET0	TF0	[PT0H, PT0L]	...	000Bh
#2	ExternalInterrupt1, nINT1	EX1	IE1	[PX1H, PX1L]	...	0013H
#3	Timer1	ET1	TF1	[PT1H, PT1L]	...	001BH
#4	SerialPort0	ES0	RI0, TI0	[PS0H, PS0L]	...	0023H
#5	Timer2	ET2	TF2, EXF2	[PT2H, PT2L]	...	002Bh
#6	ExternalInterrupt2, nINT2	EX2	IE2	[PX2H, PX2L]	...	0033H
#7	SPI	ESPI	SPIF	[PSPIH, PSPIL]	...	003BH
#8	ADC	EADC	ADCI	[PADCH, PADCL]	...	0043h
#9	PCA0	EPCA	CF, CCFn (n=0~5)	[PPCAH, PPCAL]	...	004BH
#10	SystemFlag	ESF	(注 1)	[PSFH, PSFL]	...	0053H
#11	KeypadInterrupt	EKB	KBIF	[PKBH, PKBL]	...	005BH
#12	TWI0/I2C0	ETWI0	SI	[PTWI0H, PTWI0L]	...	0063H
#13	Reserved	--	--	--	...	006BH
#14	SerialPort1	ES1	RI1, TI1	[PS1H, PS1L]	...	0073H
#15	Reserved	--	--	--	...	007BH
#16	Timer3	ET3	TF3, EXF3 (TF3L)	[PT3H, PT3L]	...	0083H
#17	Reserved	--	--	--	...	008BH
#18	DMA	EDMA	(注 2)	[PDMAH, PDMAL]	(Lowest)	0093H

Note 1: The System Flag interrupt flags include: WDTF, BOF0, BOF1 and RTCF in PCON1, TI0 in S0CON, STAF and STOF in AUXR2.

Note2: The DMA interrupt flags include: DCF0, TF5 and TF6.

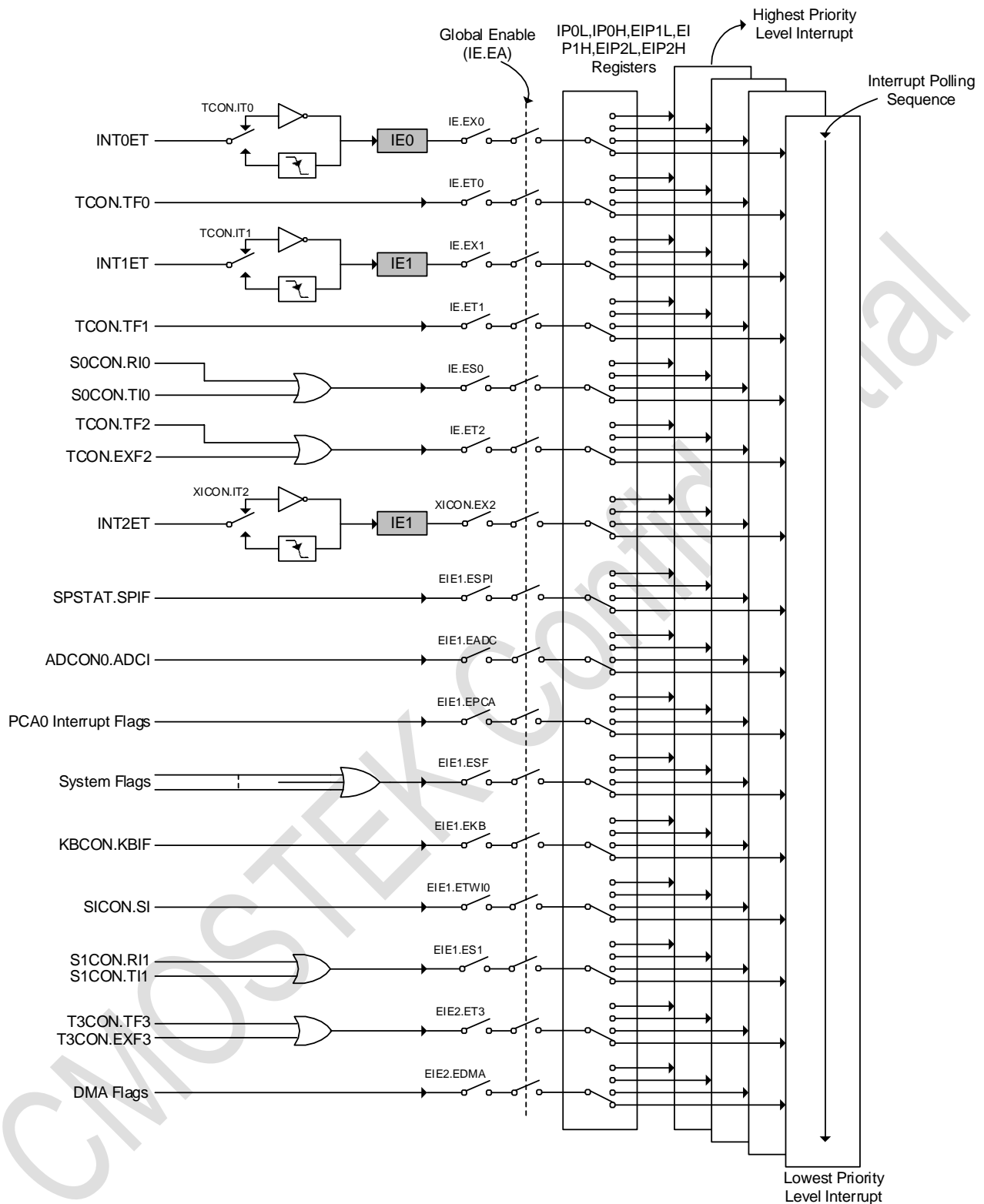


Figure 15-1. Interrupt System

15.2 Interrupt Source

Table15-2. Interrupt Source Flag

No	SourceName	RequestBits	BitLocation
#0	ExternalInterrupt0,nINT0	IE0	TCON.1
#1	Timer0	TF0	TCON.5
#2	ExternalInterrupt1,nINT1	IE1	TCON.3
#3	Timer1	TF1	TCON.7
#4	SerialPort0	RI0,TI0	S0CON.0S0CON.1
#5	Timer2	TF2,EXF2, (TF2L)	T2CON.7T2CON.6 T2CON.5
#6	ExternalInterrupt2,nINT2	IE2	XICON.1
#7	SPI	SPIF	SPSTAT.7
#8	ADC	ADCI, ADCWI,SMPF	ADCON0.4 ADCON0.6ADCFG0.2
#9	PCA0	CF, CCFn (n=0~5),CCFn(n=6~7)	CCON.7CCON.5~0PCAPWMn.3
#10	SystemFlag	WDTF,BOF1,BOF0,RTCF, STAF,STOF,BM0F,BM1F, (TI0)	PCON1.0PCON1.1PCON1.2PCON1.4A UXR2.7AUXR2.6AUXR0.2AUXR0.3 S0CON.1
#11	KeypadInterrupt	KBIF	KBCON.0
#12	TWI0/I2C0	SI	SICON.3
#13	Reserved	--	--
#14	SerialPort1	RI1,TI1	S1CON.0S1CON.1
#15	Reserved	--	--

# 1 6	Timer3	TF3, EXF3,(TF3L)	T3CON.7 T3CON.6T3CON.5
# 1 7	Reserved	--	--
# 1 8	DMA	DCF0 TF5TF6	DMACR0.0 T5CON.7T6CON.7

The external interrupt nINT0, nINT1 and nINT2 can each be either level-activated or transition-activated, depending on bits IT0 and IT1 in register TCON and IT2 in register XICON. The flags that actually generate these interrupts are bits IE0 and IE1 in TCON and IE2 in XICON. When an external interrupt is generated, the flag that generated it is cleared by the hardware when the service routine is vectored to only if the interrupt was transition –activated, then the external requesting source is what controls the request flag, rather than the on-chip hardware.

The Timer0 and Timer1 interrupts are generated by TF0 and TF1, which are set by a rollover in their respective Timer/Counter registers in most cases. When a timer interrupt is generated, the flag that generated it is cleared by the on-chip hardware when the service routine is vectored to.

The serial port 0 interrupt is generated by the logical OR of RI0 and TI0. Neither of these flags is cleared by hardware

when the service routine is vectored to. The service routine should poll RI0 and TI0 to determine which one to request service and it will be cleared by software.

The serial port 1 interrupt is generated by the logical OR of RI1 and TI1. Neither of these flags is cleared by hardware when the service routine is vectored to. The service routine should poll RI1 and TI1 to determine which one to request service and it will be cleared by software.

The timer2 interrupt is generated by the logical OR of TF2 and EXF2. If the timer 2 in split mode, the TL2 overflow will set another interrupt flag, TF2L. Just the same as serial port, neither of these flags is cleared by hardware when the service routine is vectored to.

The timer3 interrupt is generated by the logical OR of TF3 and EXF3. If the timer 3 in split mode, the TL3 overflow will set another interrupt flag, TF3L. Just the same as serial port, neither of these flags is cleared by hardware when the service routine is vectored to.

SPI interrupt is generated by SPIF in SPSTAT, which are set by SPI engine finishes a SPI transfer. It will not be cleared by hardware when the service routine is vectored to.

The ADC interrupt is generated by ADCI in ADCON0. It will not be cleared by hardware when the service routine is vectored to.

The PCA0 interrupt is generated by the logical OR of CF, CCF5, CCF4, CCF3, CCF2, CCF1 and CCF0 in CCON. Neither of these flags is cleared by hardware when the service routine is vectored to. The service routine should poll these flags to determine which one to request service and it will be cleared by software.

The System Flag interrupt is generated by RTCF, BOF1, BOF0, WDTF, TI0, STAF and STOF. STAF and

STOF are set by serial interface detection and stored in AUXR2. The Serial Port TI flag is optional to locate the interrupt vector shared with system flag interrupt which is enabled by UTIE set. The rest flags are stored in PCON1. RTCF is set by RTC counter overflow. BOF1 and BOF0 are set by on chip Brownout-Detector (BOD1 and BOD0) met the low voltage event. WDTF is set by Watch-Dog-Timer overflow. These flags will not be cleared by hardware when the service routine is vectored to. Figure 15–2 shows the system flag interrupt configuration.

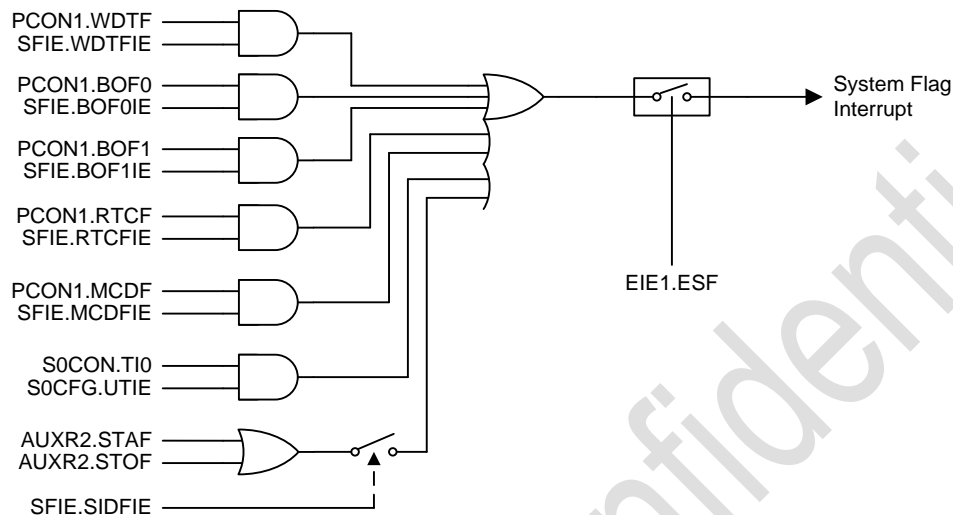


Figure 15-2. System flag interrupt configuration

The keypad interrupt is generated by KBCON.KBIF, which is set by Keypad module meets the input pattern. It will not be cleared by hardware when the service routine is vectored to.

The TWI0/ I2C0 interrupt is generate by SI in SICON, which is set by TWI0/ I2C0 engine detecting a new bus state updated. It will not be cleared by hardware when the service routine is vectored to.

All of the bits that generate interrupts can be set or cleared by software, with the same result as though it had been set or cleared by hardware. In other words, interrupts can be generated or pending interrupts can be canceled in software.

15.3 Interrupt Enable

Table 15-3. Interrupt Enable

No	SourceName	EnableBit	BitLocation
#0	ExternalInterrupt0,nINT0	EX0	IE.0
#1	Timer0	ET0	IE.1
#2	ExternalInterrupt1,nINT1	EX1	IE.2
#3	Timer1	ET1	IE.3
#4	SerialPort0	ES0	IE.4
#5	Timer2	ET2	IE.5
#6	ExternalInterrupt2,nINT2	EX2	XICON.2
#7	SPI	ESPI	EIE1.0
#8	ADC	EADC	EIE1.1
#9	PCA	EPCA	EIE1.2
#10	SystemFlag	ESF	EIE1.3
#11	KeypadInterrupt	EKB	EIE1.5
#12	TWI0/I2C0	ETWI0	EIE1.6
#13	Reserved	--	
#14	SerialPort1	ES1	EIE1.4
#15	Reserved	--	
#16	Timer3	ET3	EIE2.0
#17	Reserved	--	
#18	DMA	EDMA	IE.6

There are 16 interrupt sources available in CMT2380F17. Each of these interrupt sources can be individually enabled or disabled by setting or clearing an interrupt enable bit in the registers IE, EIE1, EIE2 and XICON. IE also contains a global disable bit, EA, which can be cleared to disable all interrupts at once. If EA is set to '1', the interrupts are individually enabled or disabled by their corresponding enable bits. If EA is cleared to '0', all interrupts are disabled.

15.4 Interrupt Priority

The priority scheme for servicing the interrupts is the same as that for the 80C51, except there are four interrupt levels rather than two as on the 80C51. The Priority Bits (see Table 15–1) determine the priority level of each interrupt. IP0L, IP0H, EIP1L, EIP1H, EIP2L and EIP2H are combined to 4-level priority interrupt. Table 15–4 shows the bit values and priority levels associated with each combination.

Table 15–4. Interrupt Priority

{IPnH.x , IPnL.x}	Priority Level
11	1 (highest)
10	2
01	3
00	4

Each interrupt source has two corresponding bits to represent its priority. One is located in SFR named IPnH and the other in IPnL register. Higher-priority interrupt will be not interrupted by lower-priority interrupt request. If two interrupt requests of different priority levels are received simultaneously, the request of higher priority is serviced. If interrupt requests of the same priority level are received simultaneously, an internal polling sequence determine which request is serviced. Table 15–2 shows the internal polling sequence in the

same priority level and the interrupt vector address.

15.5 Interrupt Process

Each interrupt flag is sampled at every system clock cycle. The samples are polled during the next system clock. If one of the flags was in a set condition at first cycle, the second cycle (polling cycle) will find it and the interrupt system will generate an hardware LCALL to the appropriate service routine as long as it is not blocked by any of the following conditions.

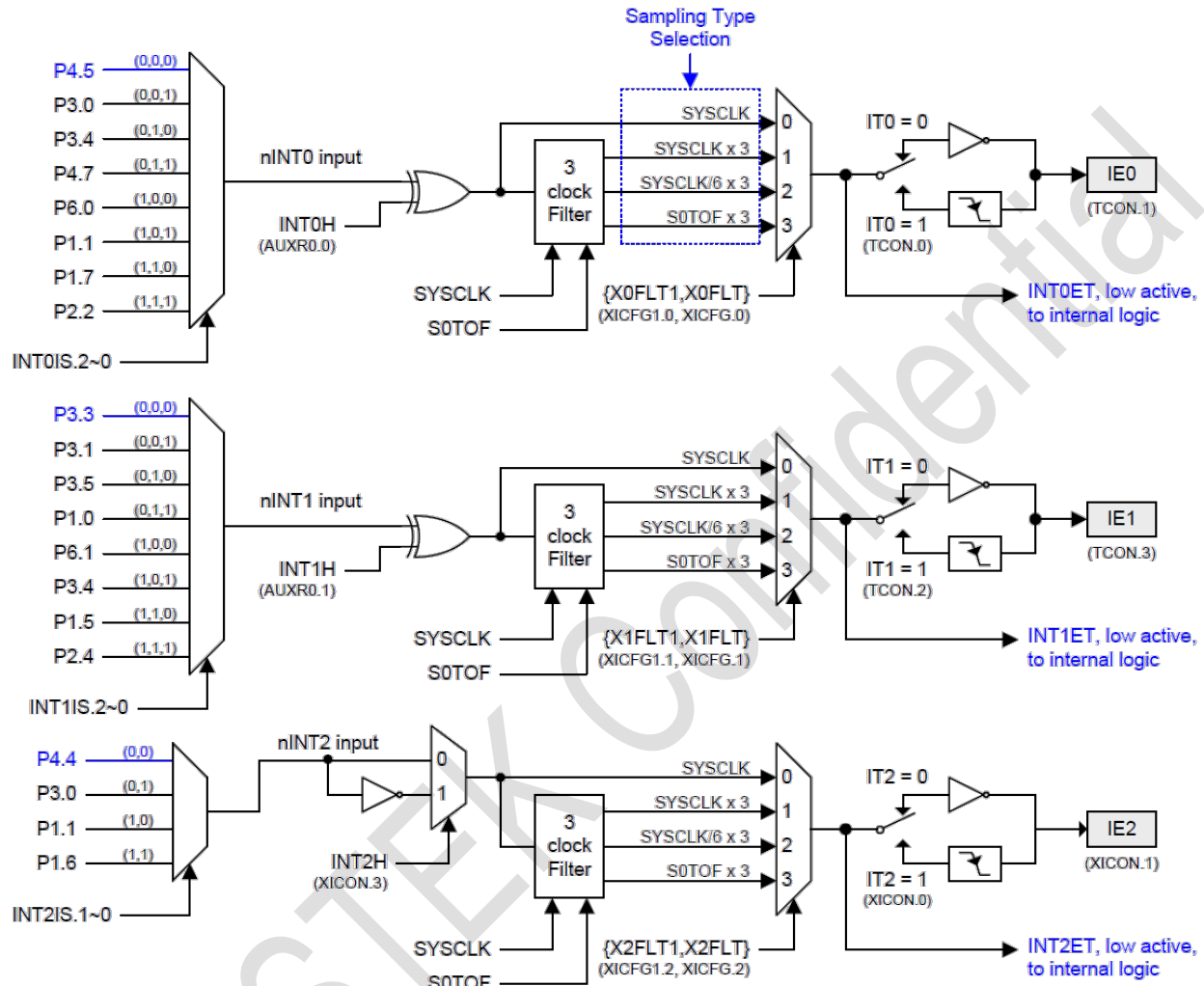
Block conditions:

- An interrupt of equal or higher priority level is already in progress.
- The current cycle (polling cycle) is not the final cycle in the execution of the instruction in progress.
- The instruction in progress is RETI or any write to the IE, IP0L, IPH, EIE1, EIP1L, EIP1H, EIE2, EIP2L, EIP2H and XICON registers.◦

Any of these three conditions will block the generation of the hardware LCALL to the interrupt service routine. Condition 2 ensures that the instruction in progress will be completed before vectoring into any service routine. Condition 3 ensures that if the instruction in progress is RETI or any access to IE or IP, then at least one or more instruction will be executed before any interrupt is vectored to.

15.6 nINTx Input Source Selection and input filter (x=0~2)

The CMT2380F17 provides flexible nINT0, nINT1 and nINT2 source selection to share the port pin inputs...



Operating Example of nINTx Input Filter (x = 0~2)

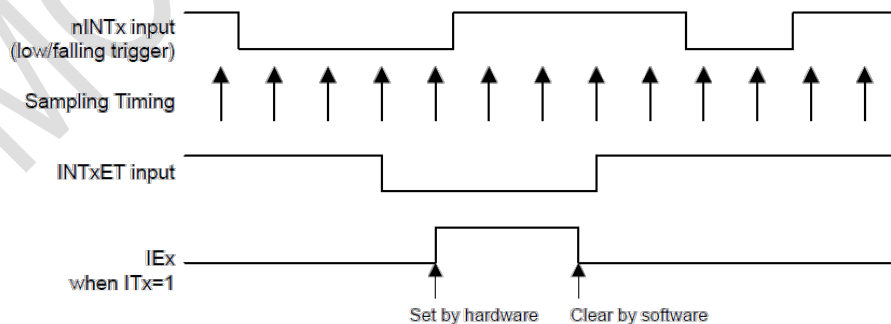


Figure 15-2. Configuration of nINT0~2 Port Pin Selection

15.7 Interrupt Register

- TCON: Timer/Counter Control Register

SFR Page = 0~F

SFR Address = 0x88

Bit	7	6	5	4	3	2	1	0
Name	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 3: IE1, Interrupt 1 (nINT1) Edge flag.

0: Cleared when interrupt processed on if transition-activated.

1: Set by hardware when external interrupt 1 (nINT1) edge is detected (transmitted or level-activated).

Bit 2: IT1: Interrupt 1 (nINT1) Type control bit.

0: Cleared by software to specify low level triggered external interrupt 1 (nINT1). If INT1H (AUXR0.1) is set, this bit specifies high level triggered on nINT1.

1: Set by software to specify falling edge triggered external interrupt 1 (nINT1). If INT1H (AUXR0.1) is set, this bit specifies rising edge triggered on nINT1.

Bit 1: IE0, Interrupt 0 (nINT0) Edge flag.

0: Cleared when interrupt processed on if transition-activated.

1: Set by hardware when external interrupt 0 (nINT0) edge is detected (transmitted or level-activated).

Bit 0: IT0: Interrupt 0 (nINT0) Type control bit.

0: Cleared by software to specify low level triggered external interrupt 0 (nINT0). If INT0H (AUXR0.0) is set, this bit specifies high level triggered on nINT0.

1: Set by software to specify falling edge triggered external interrupt 0 (nINT0). If INT0H (AUXR0.0) is set, this bit specifies rising edge triggered on nINT0.

- IE: InterruptEnableRegister

SFRPage = 0~F

SFRAddress = 0xA8

Bit	7	6	5	4	3	2	1	0
Name	EA	EDMA	ET2	ES0	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: EA, All interrupts enable register. 0: Global disables all interrupts.
1: Global enables all interrupts.

Bit 6: EDMA, DMA group interrupt enable register. 0: Disable DMA group interrupt.
1: Enable DMA group interrupt.

Bit 5: ET2, Timer 2 interrupt enable register. 0: Disable Timer 2 interrupt.
1: Enable Timer 2 interrupt.

Bit 4: ES, Serial port 0 interrupt (UART0) enable register. 0: Disable serial port 0 interrupt.
1: Enable serial port 0 interrupt.

Bit 3: ET1, Timer 1 interrupt enable register. 0: Disable Timer 1 interrupt.
1: Enable Timer 1 interrupt.

Bit 2: EX1, External interrupt 1 (nINT1) enable register. 0: Disable external interrupt 1.
1: Enable external interrupt 1.

Bit 1: ET0, Timer 0 interrupt enable register. 0: Disable Timer 0 interrupt.
1: Enable Timer 1 interrupt.

Bit 0: EX0, External interrupt 0 (nINT0) enable register. 0: Disable external interrupt 0.
1: Enable external interrupt 1.

- AUXR0: Auxiliary Register 0

SFR Page = 0~F

SFR Address = 0xA1

Bit	7	6	5	4	3	2	1	0
Name	P60OC1	P60OC0	P60FD	PBKF	--	--	INT1H	INT0H
R/W	R/W	R/W	R/W	R/W	W	W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 4: PBKF, PWM Break Flag. This bit is set by PWM break source enabled. If this flag is set, the enabled PWM channel 0~5 will be blocked and the output pins keep the original GPIO state.

0: There is no PWM Break event happened. It is only cleared by software.

1: There is a PWM Break event happened or software triggers a PWM Break.

Bit 1: INT1H, INT1 High/Rising trigger enable.

0: Remain INT1 triggered on low level or falling edge on selected port pin input.

1: Set INT1 triggered on high level or rising edge on selected port pin input.

Bit 0: INT0H, INT0 High/Rising trigger enable.

0: Remain INT0 triggered on low level or falling edge on selected port pin input.

1: Set INT0 triggered on high level or rising edge on selected port pin input.

- XICON: External Interrupt Control Register

SFR Page = 0~F

SFR Address = 0xC0

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	--	INT2H	EX2	IE2	IT2
R/W	W	W	W	W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 3: INT2H, nINT2 High/Rising trigger enable.

0: Maintain nINT2 triggered on low level or falling edge on selected port pin input. 1: Set nINT2 triggered on high level or rising edge on selected port pin input.

Bit 2: EX2, external interrupt 2 (nINT2) enable register. 0: Disable external interrupt 2.

1: Enable external interrupt 2.

When CPU in IDLE and PD mode, nINT2 event will trigger IE2 and have wake-up CPU capability if EX2 is enabled. If EX2 is disabled, IE2 on nINT2 will not wake-up CPU from IDLE or PD mode.

Bit 1: IE2, External interrupt 2 (nINT2) Edge flag.

0: Cleared by hardware when the interrupt is starting to be serviced. It also could be cleared by CPU. 1: Set by hardware when external interrupt edge detected. It also could be set by CPU.

Bit 0: IT2, Interrupt 2 type control bit.

0: Cleared by CPU to specify low level triggered on nINT2. If INT2H is set, this bit specifies high level triggered on nINT2.

1: Set by CPU to specify falling edge triggered on nINT2. If INT2H is set, this bit specifies rising edge triggered on nINT2.

- IP0L: Interrupt Priority 0 Low Register

SFR Page = 0~F

SFR Address = 0xB8

Bit	7	6	5	4	3	2	1	0
Name	--	PX2L	PT2L	PSL	PT1L	PX1L	PT0L	PX0L
R/W	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: reserved. In case of writing IP0L, this bit must be set to 0 by software.

Bit 6: PX2L, external interrupt 2 priority-L register.

Bit 5: PT2L, Timer 2 interrupt priority-L register.

Bit 4: PSL, Serial port interrupt priority-L register.

Bit 3: PT1L, Timer 1 interrupt priority-L register.

Bit 2: PX1L, external interrupt 1 priority-L register.

Bit 1: PT0L, Timer 0 interrupt priority-L register.

Bit 0: PX0L, external interrupt 0 priority-L register.

IP0H: Interrupt Priority 0 High Register

SFR Page = 0~F

SFR Address = 0xB7

Bit	7	6	5	4	3	2	1	0
Name	--	PX2H	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
R/W	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: reserved. In case of writing IP0H, this bit must be set to 0 by software.

Bit 6: PX2H, external interrupt 2 priority-H register.

Bit 5: PT2H, Timer 2 interrupt priority-H register.

Bit 4: PSH, Serial port interrupt priority-H register.

Bit 3: PT1H, Timer 1 interrupt priority-H register.

Bit 2: PX1H, external interrupt 1 priority-H register.

Bit 1: PT0H, Timer 0 interrupt priority-H register.

Bit 0: PX0H, external interrupt 0 priority-H register.。

● EIE1: Extended Interrupt Enable 1 Register

SFR Page = 0~F

SFR Address = 0xAD

Bit	7	6	5	4	3	2	1	0
Name	--	ETWIO	EKB	ES1	ESF	EPCA	EADC	ESPI
R/W	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

B Bit 6: ETWIO, Enable TWIO/ I2C0 interrupt.

0: Disable TWIO/ I2C0 interrupt.

1: Enable TWIO/ I2C0 interrupt.

Bit 5: EKB, Enable Keypad Interrupt.

0: Disable the interrupt when KBCON.KBIF is set in Keypad control module.

1: Enable the interrupt when KBCON.KBIF is set in Keypad control module.

Bit 4: ES1, Enable Serial Port 1 (UART1) interrupt.

0: Disable Serial Port 1 interrupt.

1: Enable Serial Port 1 interrupt.

Bit 3: ESF, Enable System Flag interrupt.

0: Disable the interrupt when the group of {RTCF, BOF1, BOF0, WDTF} in PCON1, {STAF, STOF} in AUXR2, {BM1F, BM0F} in AUXR0, or TI0 with UTIE is set.

1: Enable the interrupt of the flags of { RTCF, BOF1, BOF0, WDTF} in PCON1, {STAF, STOF} in AUXR2, {BM1F, BM0F} in AUXR0, or TI0 with UTIE when the associated system flag interrupt is enabled in SFIE.

Bit 2: EPCA, Enable PCA0 interrupt.

0: Disable PCA0 interrupt.

1: Enable PCA0 interrupt.

Bit 1: EADC, Enable ADC Interrupt.

0: Disable the interrupt when ADCON0.ADCI is set in ADC module.

1: Enable the interrupt when ACCON0.ADCI is set in ADC module.

Bit 0: ESPI, Enable SPI Interrupt.

0: Disable the interrupt when SPSTAT.SPIF is set in SPI module.

1: Enable the interrupt when SPSTAT.SPIF is set in SPI module.

- EIP1L: Extended Interrupt Priority 1 Low Register

SFR Page = 0~F

SFR Address = 0xAE

Bit	7	6	5	4	3	2	1	0
Name	--	PTWIO0L	PKBL	PS1L	PSFL	PPCAL	PADCL	PSPIL
R/W	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 6: PTWIO0L, TWI0/ I2C0 interrupt priority-L register.

Bit 5: PKBL, keypad interrupt priority-L register.

Bit 4: PS1L, UART1 interrupt priority-L register.

Bit 3: PSFL, system flag interrupt priority-L register.

Bit 2: PPCAL, PCA0 interrupt priority-L register.

Bit 1: PADCL, ADC interrupt priority-L register.

Bit 0: PSPIL, SPI interrupt priority-L register.

- EIP1H: Extended Interrupt Priority 1 High Register

SFR Page = 0~F

SFR Address = 0xAF

Bit	7	6	5	4	3	2	1	0
Name	--	PTWIO0H	PKBH	PS1H	PSFH	PPCAH	PADCH	PSPIH
R/W	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	X	0	0	0	0

Bit 6: PTWIO0H, TWI0/ I2C0 interrupt priority-H register.

Bit 5: PKBH, keypad interrupt priority-H register.

Bit 4: PS1H, UART1 interrupt priority-H register.

Bit 3: PSFH, system flag interrupt priority-H register.

Bit 2: PPCAH, PCA0 interrupt priority-H register.

Bit 1: PADCH, ADC interrupt priority-H register.

Bit 0: PSPIH, SPI interrupt priority-H register.

- EIE2: Extended Interrupt Enable 2 Register

SFR Page = 0~F

SFR Address = 0xA5

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	--	--	--	--	ET3
R/W	W	W	W	W	W	W	W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~1: Reserved. Software must write "0" on these bits when EIE2 is written.

Bit 0: ET3, Timer 3 interrupt enable register.

0: Disable Timer 3 interrupt.

1: Enable Timer 3 interrupt.

- EIP2L: Extended Interrupt Priority 2 Low Register

SFR Page = 0~F

SFR Address = 0xA6

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	--	--	--	--	PT3L
R/W	W	W	W	W	W	W	W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~1: Reserved. Software must write "0" on these bits when EIP2L is written.

Bit 0: PT3L, Timer 3 interrupt priority-L register.

- EIP2H: Extended Interrupt Priority 2 High Register

SFR Page = 0~F

SFR Address = 0xA7

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	--	--	--	--	PT3H
R/W	W	W	W	W	W	W	W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~1: Reserved. Software must write "0" on these bits when EIP2H is written.

Bit 0: PT3H, Timer 3 interrupt priority-H register.

- DMACG0: DMA Configuration Register 0

SFR Page = 8 only

SFR Address = 0x94

Bit	7	6	5	4	3	2	1	0
Name	PDMAH	PDMAL	CRCW0	--	EXTS10	EXTS00	FAEN0	LOOP0
R/W	R/W	R/W	R/W	W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: PDMAH, DMA group interrupt priority-H register.

Bit 6: PDMAL, DMA group interrupt priority-L register.

- XICFG: External Interrupt Configured Register

SFR Page = 0 only

SFR Address = 0xC1

Bit	7	6	5	4	3	2	1	0
Name	INT1IS[1:0]		INT0IS[1:0]		--	X2FLT	X1FLT	X0FLT
R/W	R/W	R/W	R/W	R/W	W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~6: INT1IS.1~0, nINT1 input port pin selection bits which function is defined with INT1IS.2 as following table.

INT1IS[2:0]	Selected Port Pin of nINT1
000	P3.3
001	P3.1
010	P3.5
011	P1.0
100	P6.1
101	P3.4
110	P1.5
111	P2.4

Bit 5~4: INT0IS.1~0, nINT0 input port pin selection bits which function is defined with INT0IS.2 as following table.

INT0IS[2:0]	Selected Port Pin of nINT0
000	P4.5
001	P3.0
010	P3.4
011	P4.7
100	P6.0
101	P1.1
110	P1.7
111	P2.2

Bit 3: reserved. In case of writing XICFG, this bit must be written to 0 by software.

Bit 2: X2FLT, nINT2 Filter mode control. It selects nINT2 input filter mode with X2FLT1 (XICFG1.2).

X2FLT1, X2FLT	nINT2 input filter mode
00	Disabled
01	SYSCLK x 3
10	SYSCLK/6 x 3
11	S0TOF x 3

Bit 1: X1FLT, nINT1 Filter mode control. It selects nINT1 input filter mode with X1FLT1 (XICFG1.1).

X1FLT1, X1FLT	nINT1 input filter mode
00	Disabled
01	SYSCLK x 3
10	SYSCLK/6 x 3
11	S0TOF x 3

Bit 0: X0FLT, nINT0 Filter mode control. It selects nINT0 input filter mode with X0FLT1 (XICFG1.0).

X0FLT1, X0FLT	nINT0 input filter mode
00	Disabled
01	SYSCLK x 3
10	SYSCLK/6 x 3
11	S0TOF x 3

- XICFG1: External Interrupt Configured 1 Register

SFR Page = 1 only

SFR Address = 0xC1

Bit	7	6	5	4	3	2	1	0
Name	INT1IS.2	INT0IS.2	INT2IS[1:0]		--	X2FLT1	X1FLT1	X0FLT1
R/W	R/W	R/W	R/W	R/W	W	R/W	R/W	R/W
Reset Value	0	0	0	0	X	0	0	0

Bit 7: INT1IS2, nINT1 input port pin selection bit which function is defined with INT1IS.1~0.

Bit 6: INT0IS2, nINT0 input port pin selection bit which function is defined with INT0IS.1~0.

Bit 5~4: INT2IS1~0, nINT2 input port pin selection bits which function is defined as following table.

INT2IS[1:0]	Selected Port Pin of nINT2
00	P4.4
01	P3.0
10	P1.1
11	P1.6

Bit 3: reserved. In case of writing XICFG1, this bit must be written to 0 by software.

Bit 2: X2FLT1, nINT2 Filter mode control. It selects nINT2 input filter mode with X2FLT (XICFG.2). Refer XICFG description for nINT2 input filter mode definition.

Bit 1: X1FLT1, nINT1 Filter mode control. It selects nINT1 input filter mode with X1FLT (XICFG.1). Refer XICFG description for nINT1 input filter mode definition.

Bit 0: X0FLT1, nINT0 Filter mode control. It selects nINT0 input filter mode with X0FLT (XICFG.0). Refer XICFG description for nINT0 input filter mode definition.

- SFIE: System Flag Interrupt Enable Register

SFR Page = 0~F

SFR Address = 0x8E

Bit	7	6	5	4	3	2	1	0
Name	SIDFIE	--	--	RTCFIE	--	BOF1IE	BOF0IE	WDTFIE
R/W	R/W	W	W	R/W	W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: SIDFIE, Serial Interface (STWI/SI2C) Detection Flag Interrupt Enabled. 0: Disable SIDF (STAF or STOF) interrupt.
1: Enable SIDF (STAF or STOF) interrupt to share the system flag interrupt.

Bit 4: RTCFIE, Enable RTCF (PCON1.4) Interrupt.
0: Disable RTCF interrupt.
1: Enable RTCF interrupt.

Bit 3: Reserved. Software must write "0" on this bit when SFIE is written.

Bit 2: BOF1IE, Enable BOF1 (PCON1.2) Interrupt.
0: Disable BOF1 interrupt.
1: Enable BOF1 interrupt.

Bit 1: BOF0IE, Enable BOF0 (PCON1.1) Interrupt.

0: Disable BOF0 interrupt.

1: Enable BOF0 interrupt.

Bit 0: WDTFIE, Enable WDTF (PCON1.0) Interrupt.

0: Disable WDTF interrupt.

1: Enable WDTF interrupt.

- PCON1: Power Control Register 1

SFR Page = 0~F & P

SFR Address = 0x97

Bit	7	6	5	4	3	2	1	0
Name	SWRF	EXRF	--	RTCF	--	BOF1	BOF0	WDTF
R/W	R/W	R/W	W	R/W	W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: SWRF, Software Reset Flag.

0: This bit must be cleared by software writing "1" to it.

1: This bit is set by hardware if a Software Reset occurs.

Bit 6: EXRF, External Reset Flag.

0: This bit must be cleared by software writing "1" to it.

1: This bit is set by hardware if an External Reset occurs.

Bit 4: RTCF, RTC overflow flag.

0: This bit must be cleared by software writing "1" on it. Software writing "0" is no operation.

1: This bit is only set by hardware when RTCCT overflows. Writing "1" on this bit will clear RTCF.

Bit 3: Reserved. Software must write "0" on this bit when PCON1 is written.

Bit 2: BOF1, Brown-Out Detection flag 1.

0: This bit must be cleared by software writing "1" to it.

1: This bit is set by hardware if the operating voltage matches the detection level of Brown-Out Detector 1 (4.2V/3.7/2.4/2.0).

Bit 1: BOF0, Brown-Out Detection flag 0.

0: This bit must be cleared by software writing "1" to it.

1: This bit is set by hardware if the operating voltage matches the detection level of Brown-Out Detector 0 (1.7V).

Bit 0: WDTF, WDT overflow flag.

0: This bit must be cleared by software writing "1" to it. 1: This bit is set by hardware if a WDT overflow occurs.

- AUXR2: Auxiliary Register 2

SFR Page = 0 only

SFR Address = 0xA3

Bit	7	6	5	4	3	2	1	0
Name	STAF	STOF	--	--	T1X12	T0X12	T1CKOE	T0CKOE
R/W	R/W	R/W	W	W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: STAF, Start Flag detection of STWI (SID).

0: Clear by firmware by writing "0" on it. STAF might be held within MCU reset period, so needs to clear STAF in firmware initial.

1: Set by hardware to indicate the START condition occurred on STWI bus.

Bit 6: STOF, Stop Flag detection of STWI (SID).

0: Clear by firmware by writing "0" on it.

1: Set by hardware to indicate the STOP condition occurred on STWI bus. STOF might be held within MCU reset period, so needs to clear STOF in firmware initial.

CMOSTEK Confidential

16 Timers/Counters

CMT2380F17 has four 16-bit Timers/Counters: Timer 0, Timer 1, Timer 2 and Timer 3. All of them can be configured as timers or event counters.

In the “timer” function, the timer rate is prescaled by 12 clock cycle to increase register value. In other words, it is to count the standard C51 machine cycle. AUXR2.T0X12, AUXR2.T1X12, T2MOD.T2X12 and T3MOD.T3X12 are the function for Timer 0/1/2/3 to set the timer rate on every clock cycle. It performs at a speed 12 times than standard C51 timer function. Other prescaler values can be selected by combining T0C/T, T0XL and T0X12 for Timer 0 clock input.

In the “counter” function, the register is increased in response to a 1-to-0 transition at its corresponding external input pin, T0, T1, T2 or T3. In this function, the external input is sampled by every timer rate cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register at the end of the cycle following the one in which the transition was detected.

16.1 Timer 0 and Timer 1

16.1.1 Timer 0/1 Mode 0

The timer register is configured as a PWM generator. As the count rolls over from all 1s to all 0s, it sets the timer interrupt flag TFX. Timer0 uses the control bits {T0XL, T0X12, T0C/T} to set the clock source to count. And it also uses TR0 and {T0G1, T0GATE} to select the gating sources to block the trigger signal to stop the counting. Timer1 uses the control bits {T1X12, T1C/T} to set the clock source to count. And it uses TR1 and {T1G1, T1GATE} to select the gating sources to block the trigger signal to stop the counting. Mode 0 operation is the same for Timer0 and Timer1. The PWM function of Timer 0/1 is shown in Figure 16–1 and Figure 16–2.



16.1.2 Timer 0/1 Mode 1

Timer 0/1 in Mode1 is configured as a 16 bit timer or counter. The function of GATE, TxG1 and TRx is same as mode 0. Figure 16–3 and Figure 16–4 show the mode 1 structure of Timer 0 and Timer 1.

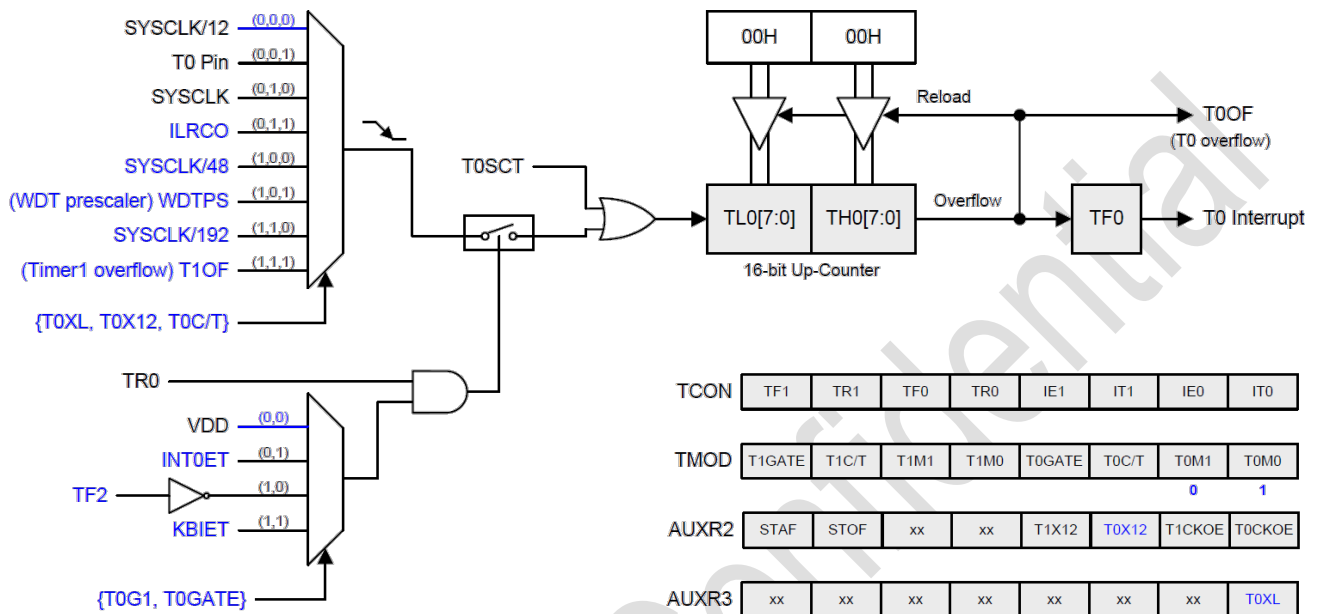


Figure 6-3. Timer 0 Mode 1 Structure

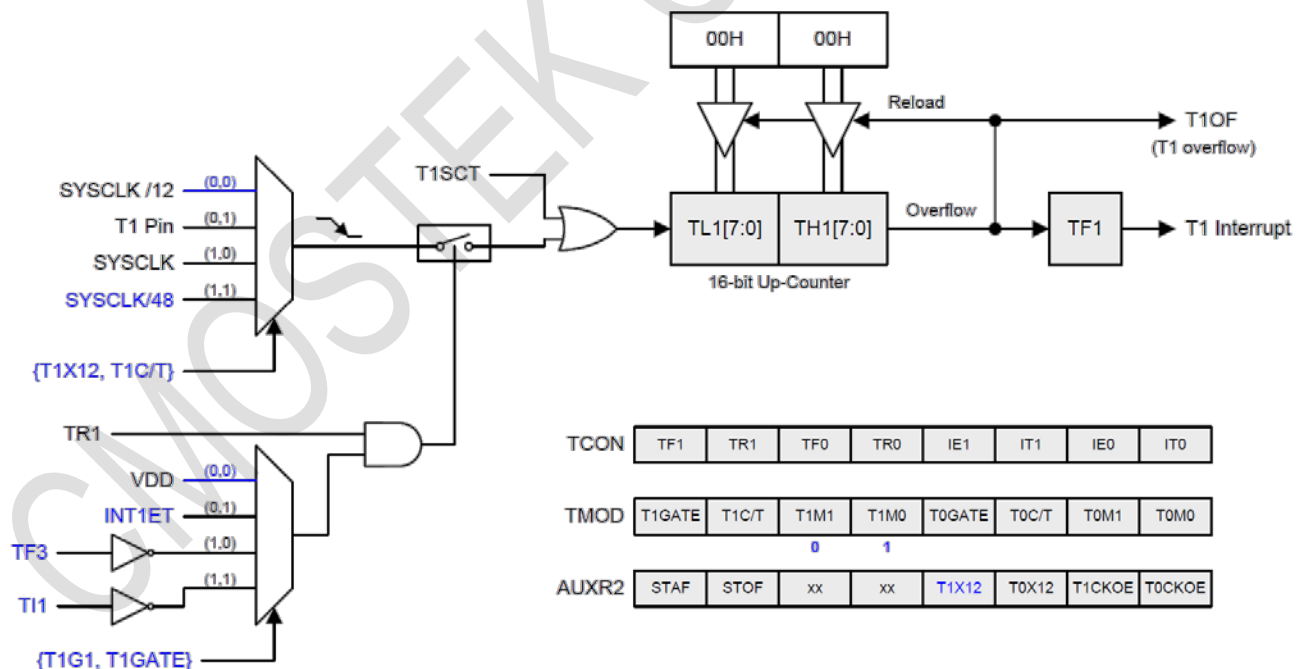


Figure 16-4. Timer 1 Mode 1 Structure

16.1.3 Timer 0/1 Mode 2

Mode 2 configures the timer register as an 8-bit counter (TLx) with automatic reload. Overflow from TLx not only set TFx, but also reload TLx with the content of THx, which is determined by software. The reload leaves THx unchanged. Mode 2 operation is the same for Timer0 and Timer1. Figure 16–5 and Figure 16–6 show the mode 2 structure of Timer 0 and Timer 1.

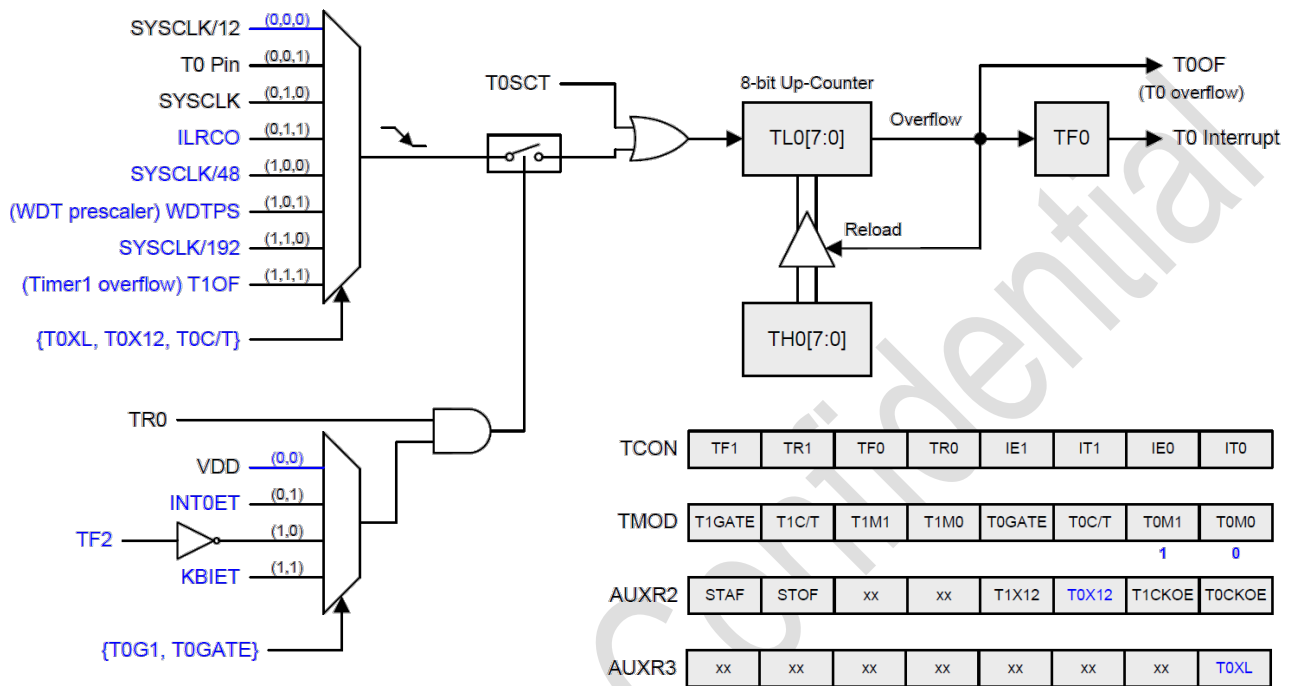


Figure 16-5. Timer 0 Mode 2 Structure

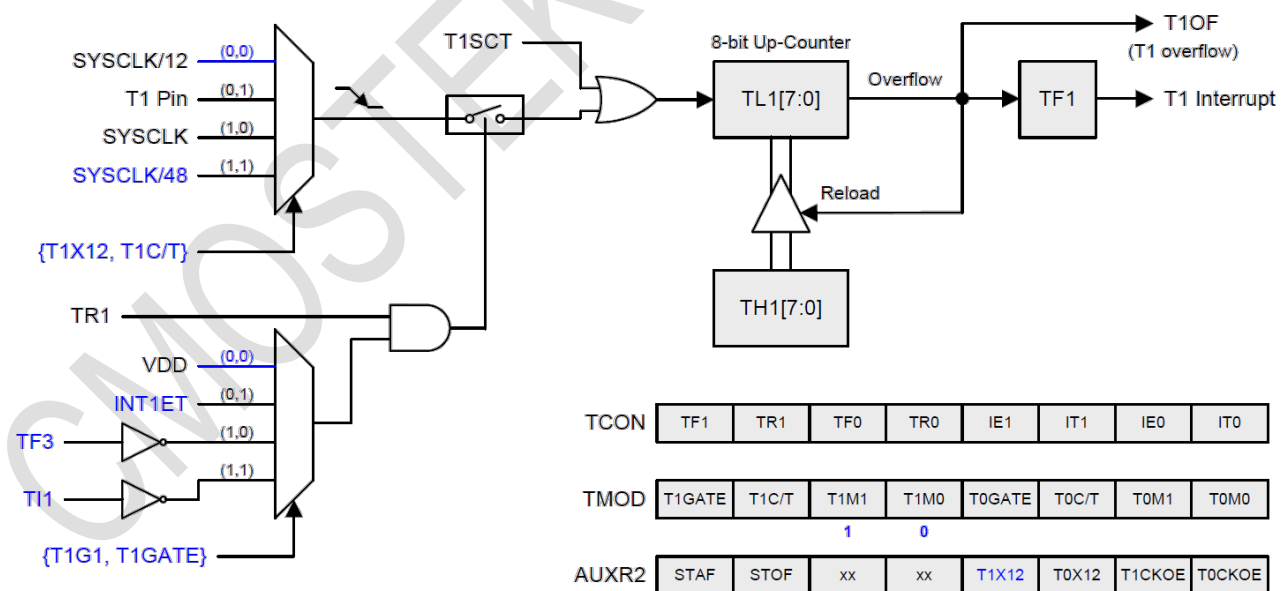


Figure 16-6. Timer 1 Mode 2 Structure

16.1.4 Timer 0/1 Mode 3

Timer1 in Mode3 simply holds its count, the effect is the same as setting TR1 = 0. Timer0 in Mode 3 enables TL0 and TH0 as two separate 8-bit counters. TL0 uses the Timer0 control bits such like T0XL, T0X12, T0C/T, T0G1, T0GATE, TR0 and TF0. TH0 is locked into a timer function (can not be external event counter) and take over the use of TR1, TF1 from Timer1. TH0 now controls the Timer1 interrupt. Figure 16–7 shows the mode 3 structure of Timer 0.

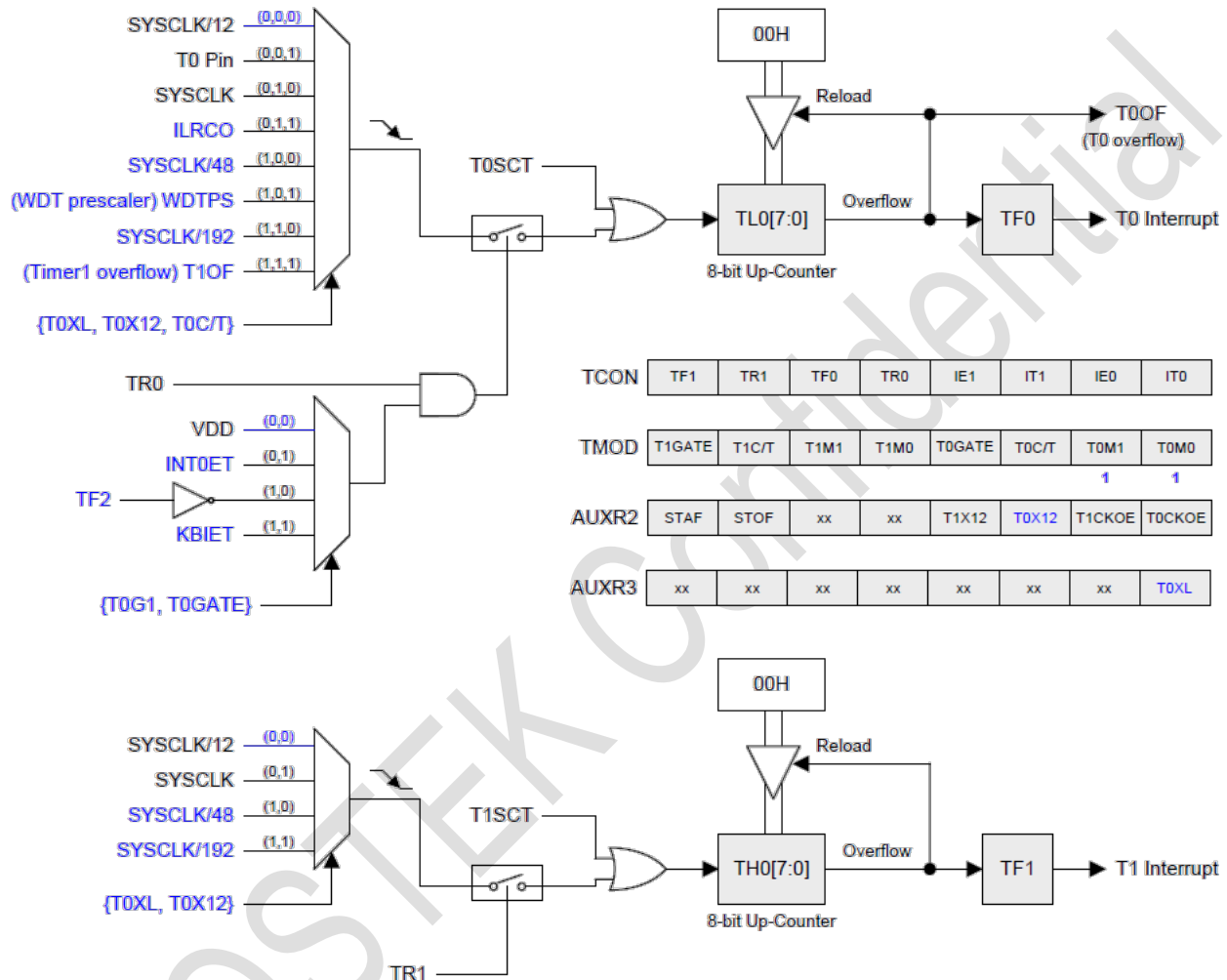


Figure 16-7.Timer 0 Mode 3 Structure

16.1.5 Timer 0/1 Programmable Clock-Out

Timer 0 and Timer 1 have a Clock-Out Mode (while TxCKOE=1). In this mode, Timer 0 or Timer 1 operates as 8-bit auto-reload timer for a programmable clock generator with 50% duty-cycle. The generated clocks come out on T0CKO (P3.4) and T1CKO (P3.5) individually. The input clock of Timer 0 increases the 8-bit timer, TL0, in Timer 0 module. The input clock of Timer 1 increases the 8-bit timer, TL1, in Timer 1 module. The timer repeatedly counts to overflow from a loaded value. Once overflows occur, the contents of (TH0 and TH1) are loaded into (TL0, TL1) for the consecutive counting. Figure 16–8 and Figure 16–9 formula gives the formula of Timer 0 and Timer 1 clock-out frequency. Figure 16–10 and Figure 16–11 show the clock-out structure of Timer 0 and Timer 1.

$$\text{T0 Clock-out Frequency} = \frac{\text{T0 Clock Frequency}}{2 \times (256 - \text{TH0})}$$

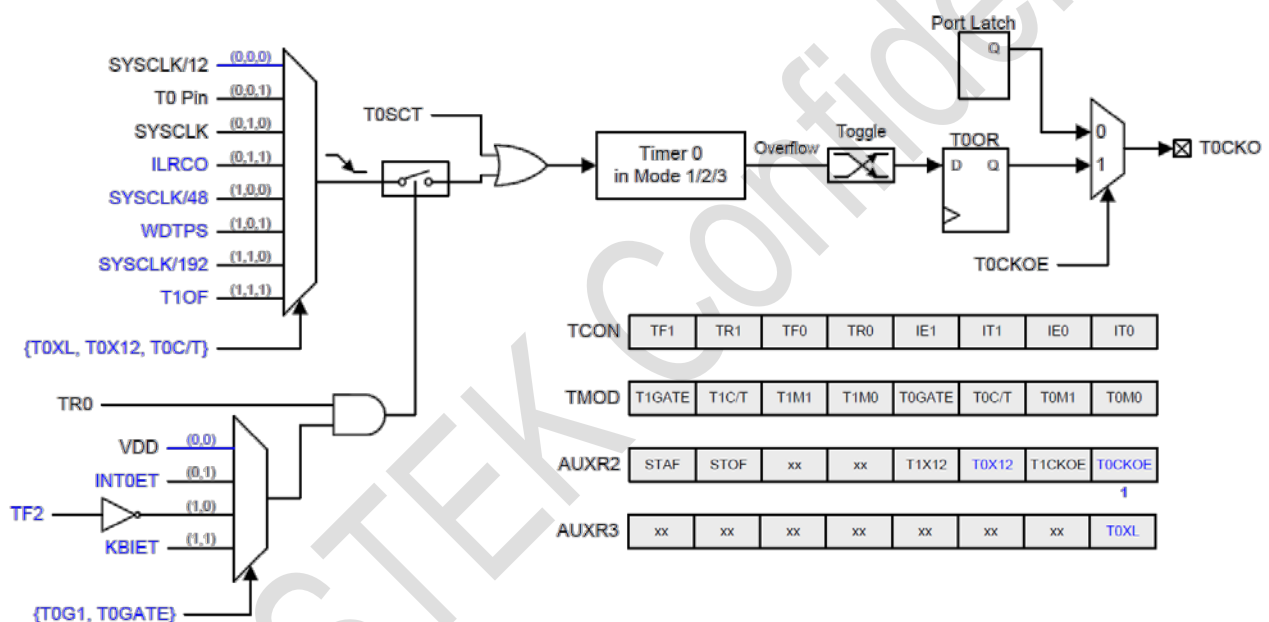
Figure 16-8. Timer 0 clock out equation

$$\text{T1 Clock-out Frequency} = \frac{\text{T1 Clock Frequency}}{2 \times (256 - \text{TH1})}$$

Figure 16-9. Timer 0 clock out equation

Note:

- (1) Timer 0/1 overflow flag, TF0/1, will be set when Timer 0/1 overflows
- (2) For SYSCLK=12MHz and select SYSCLK/12 as Timer 0/1 clock source, Timer 0/1 has a programmable output frequency range from 1.95KHz to 500KHz.
- (3) For SYSCLK=12MHz and select SYSCLK as Timer 0/1 clock source, Timer 0/1 has a programmable output frequency range from 23.44KHz to 6MHz. .



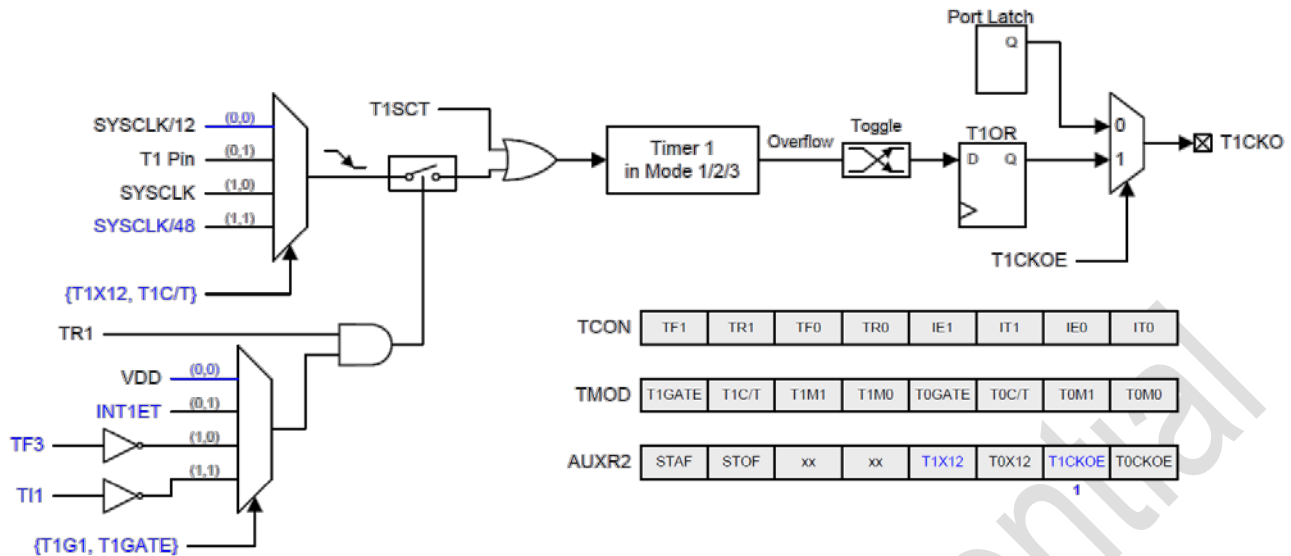


Figure 16-11. Timer 1 in Clock Output Mode

How to Program Timer 0/1 in Clock-out Mode

- Select Timer 0/1 clock source.
- Determine the 8-bit reload value from the formula and enter it in the TH0/TH1 register.
- Enter the same reload value as the initial value in the TL0/TL1 register.
- Set T0CKOE/T1CKOE bit in AUXR2 register.
- Set TR0/TR1 bit in TCON register to start the Timer 0/1.

In the Clock-Out mode, Timer 0/1 rollovers will not generate an interrupt. This is similar to when Timer 1 is used as a baud-rate generator. It is possible to use Timer 1 as a baud rate generator and a clock generator simultaneously. Note, however, that the baud-rate and the clock-out frequency depend on the same overflow rate of Timer 1. So, software usually disables the Timer 0/1 interrupt in this kind of application.

16.1.6 Timer 0/1 Register

- **TCON:** Timer/Counter Control Register
SFR Page = 0~F
SFR Address = 0x88

Bit	7	6	5	4	3	2	1	0
Name	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: TF1, Timer 1 overflow flag.

0: Cleared by hardware when the processor vectors to the interrupt routine, or cleared by software. 1: Set by hardware on Timer/Counter 1 overflow, or set by software.

Bit 6: TR1, Timer 1 Run control bit. 0: Disabled to stop Timer/Counter 1.

1: Enabled to start Timer/Counter 1.

Bit 5: TF0, Timer 0 overflow flag.

0: Cleared by hardware when the processor vectors to the interrupt routine, or cleared by software. 1: Set by hardware on Timer/Counter 0 overflow, or set by software.

Bit 4: TR0, Timer 0 Run control bit. 0: Disabled to stop Timer/Counter 0.

1: Enabled to start Timer/Counter 0.

- **TMOD:** Timer/Counter Mode Control Register

SFR Page = 0~F

SFR Address = 0x89

Bit	7	6	5	4	3	2	1	0
Name	T1GATE	T1C/T	T1M1	T1M0	T0GATE	T0C/T	T0M1	T0M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: T1GATE, Gating control for Timer1.

T1G1, T1GATE	T1Gate source
0 0	Disable
0 1	INT1active
1 0	TF3active
1 1	TI1active

Bit 6: T1C/T, Timer 1 clock source selector. It controls the Timer 1 as timer or counter with 4 clock sources. Refer to T1X12 description in the AUXR2.

Bit 5~4: Operating mode selection.

T1M1~0	Timer 1 Operating Mode
00	8-bitPWMgeneratorforTimer1
01	16-bittimer/counterforTimer1
10	8-bittimer/counterwithautomaticreloadforTimer1
11	Timer/Counter1 Stopped

Bit 3: T0GATE, Gating control for Timer0.

T0G1, T0GATE	T0Gate source
0 0	Disable
0 1	INT0active
1 0	TF2active
1 1	KBIactive

Bit 2: T0C/T, Timer 0 clock source selector. It controls the Timer 0 as timer or counter with 8 clock sources. Refer to T0X12 description in the AUXR2.

Bit 1~0: Operating mode selection.

T0M1~0	Timer 0 Operating Mode
00	8-bitPWMgeneratorforTimer0
01	16-bittimer/counterforTimer0
10	8-bittimer/counterwithautomaticreloadforTimer0
11	TL0is8-bittimer/counter, TH0islockedinto8-bittimer

- TL0: Timer 0 Low byte Register

SFR Page = 0~F

SFR Address = 0x8A

Bit	7	6	5	4	3	2	1	0
Name	TL0[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

- TH0: Timer 0 High byte Register

SFR Page = 0~F

SFR Address = 0x8C

Bit	7	6	5	4	3	2	1	0
Name	TH0[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

- TL1: Timer 1 High byte Register
SFR Page = 0~F
SFR Address = 0x8B

Bit	7	6	5	4	3	2	1	0
Name	TL1[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

- TH1: Timer 1 High byte Register

SFR Page = 0~F

SFR Address = 0x8D

Bit	7	6	5	4	3	2	1	0
Name	TH1[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

- AUXR2: Auxiliary Register 2

SFR Page = 0~F

SFR Address = 0xA3

Bit	7	6	5	4	3	2	1	0
Name	STAF	STOF	--	--	T1X12	T0X12	T1CKOE	T0CKOE
R/W	R/W	R/W	W	W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 3: T1X12, Timer 1 clock source selection with T1C/T control.

T1X12, T1C/T	Timer 1 Clock Selection
0 0	SYSClk/12
0 1	T1Pin
1 0	SYSClk
1 1	SYSClk/48

Bit 2: T0X12, T0XL and T0C/T together control the timer 0 clock source selection.

T0XL, T0X12, T0C/T	Timer 0 Clock Selection
0 0 0	SYSClk/12
0 0 1	T0Pin
0 1 0	SYSClk
0 1 1	ILRCO
1 0 0	SYSClk/48
1 0 1	WDTPS
1 1 0	SYSClk/192
1 1 1	T1OF

Bit 1: T1CKOE, Timer 1 Clock Output Enable. 0: Disable Timer 1 clock output.

1: Enable Timer 1 clock output on T1CKO Port pin.

Bit 0: T0CKOE, Timer 0 Clock Output Enable. 0: Disable Timer 0 clock output.

1: Enable Timer 0 clock output on T0CKO Port pin.

- **AUXR3: Auxiliary Register 3**
SFR Page = 0 only
SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	T0PS1	T0PS0	BPOC1	BPOC0	S0PS0	TWIPS1	TWIPS0	T0XL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~6: T0PS1~0, Timer 0 Port pin Selection [1:0].

T0PS1~0	T0/T0CKO
00	P3.4
01	P4.4
10	P2.2
11	P1.7

Bit 0: T0XL is the Timer 0 per-scaler control bit. Please refer T0X12 (AUXR2.2) for T0XL function definition.

- **AUXR4: Auxiliary Register 4**
SFR Page = 1 only
SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	T2PS1	T2PS0	T1PS1	T1PS0	--	--	--	--
R/W	R/W	R/W	R/W	R/W	W	W	W	W
Reset Value	0	0	0	0	0	0	0	0

Bit 5~4: T1PS1~0, Timer 1 Port pin Selection [1:0].

T1PS1~0	T1/T1CKO
00	P3.5
01	P4.5
10	P1.7
11	P3.3

- **AUXR9: Auxiliary Register 9**
SFR Page = 6 only
SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	--	--	T1G1	T0G1	C0FDC1	C0FDC0	S1PS1	S1PS0
R/W	W	W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 5: T1G1, Gating source selection of Timer 1.

T1G1, T1GATE	T1Gate source
00	Disable
01	INT1active
10	TF3active
11	TI1active

Bit 4: T0G1, Gating source selection of Timer 0.

T0G1, T0GATE	T0 Gate source
00	Disable
01	INT0active
10	TF2active
11	KBIactive

16.2 Timer 2

Timer 2 is a 16-bit Timer/Counter which can operate either as a timer or an event counter, as selected by C/T2 in T2CON register. Timer 2 has four operating modes: Capture, Auto-Reload (up or down counting), Baud Rate Generator and Programmable Clock-Out, which are selected by bits in the T2CON, T2MOD and T2MOD1 registers.

16.2.1 Timer 2 Mode 0 (Auto-Reload and External Interrupt)

In this mode, Timer 2 provides a 16-bit auto-reload timer/counter. The TF2, Timer 2 overflow flag, is one of the Timer 2 interrupt source which interrupt function can be blocked by TF2IG. EXEN2 enables a 1-to-0 transition at T2EXI to set the flag, EXF2, for an external input interrupt to share the Timer 2 interrupt with TF2. T2EXI is the selection result of 8 Timer 2 external inputs. T2EXH performs the same function as EXEN2 but it enables the detecting a 0-to-1 transition at T2EXI input.

The Timer 2 overflow event (T2OF) in this module will be output to other peripheral as clock input or event source. Timer 2 Mode 0 is illustrated in Figure 16–12.

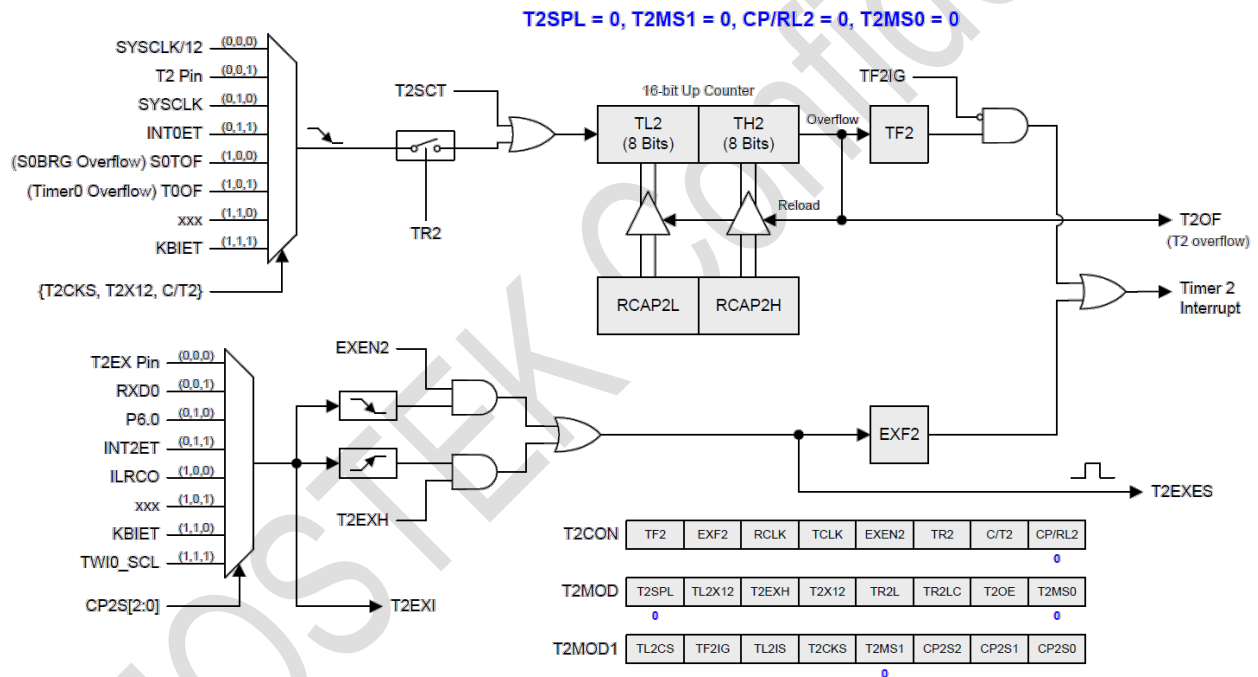


Figure 16-12. Timer 2 Mode 0 Structure (Auto-Reload and External Interrupt Mode)

16.2.2 Timer 2 Mode 1 (Auto-Reload with External Interrupt)

Figure 16–13 shows Timer 2 Mode 1, which enables Timer 2 to count up automatically. In this mode there are two options selected by bit EXEN2 in T2CON register. If EXEN2=0, then Timer 2 counts up to 0FFFFH and sets the TF2 (Overflow Flag) bit upon overflow. This causes the Timer 2 registers to be reloaded with the 16-bit value in RCAP2L and RCAP2H. The values in RCAP2L and RCAP2H are preset by firmware. If EXEN2=1, then a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at input T2EXI, one of 8 Timer 2 external inputs. This transition also sets the EXF2 bit. The Timer 2 interrupt, if enabled, can be generated when either TF2 or EXF2 are 1. T2EXH performs the same function as EXEN2 but it enables the detecting a 0-to-1 transition at input T2EXI.

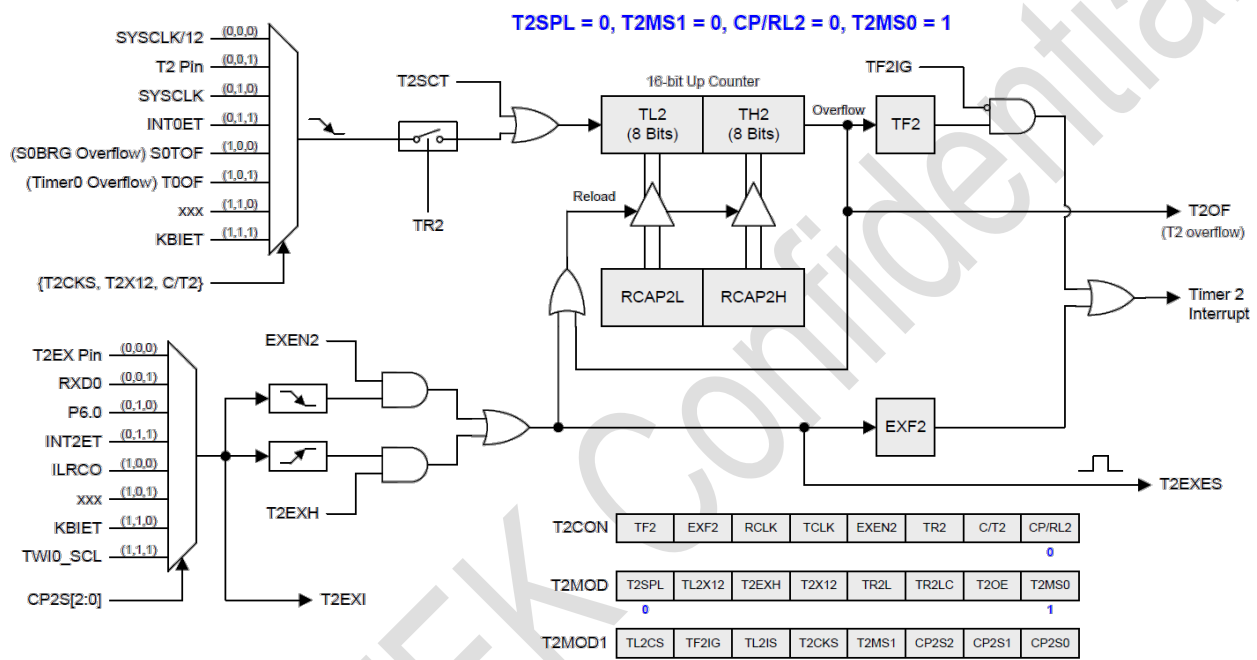


Figure 16-13. Timer 2 Mode 1 Structure (Auto-Reload with External Interrupt Mode)

16.2.3 Timer 2 Mode 2 (Capture)

Figure 16–14 shows the capture mode there are two options selected by bit EXEN2 in T2CON. If EXEN2=0, Timer 2 is a 16-bit timer or counter which, upon overflow, sets bit TF2 (Timer 2 overflow flag). This bit can then be used to generate an interrupt (by enabling the Timer 2 interrupt bit in the IE register). If EXEN2=1, Timer 2 still does the above, but with the added feature that a 1-to-0 transition at T2EXI, one of 8 Timer 2 external inputs, that causes the current value in the Timer 2 registers, TH2 and TL2, to be captured into registers RCAP2H and RCAP2L, respectively. In addition, the transition at T2EXI causes bit EXF2 in T2CON to be set, and the EXF2 bit (like TF2) can generate an interrupt which vectors to the same location as Timer 2 overflow interrupt. T2EXH performs the same function as EXEN2 but it enables the detecting a 0-to-1 transition at T2EXI input.

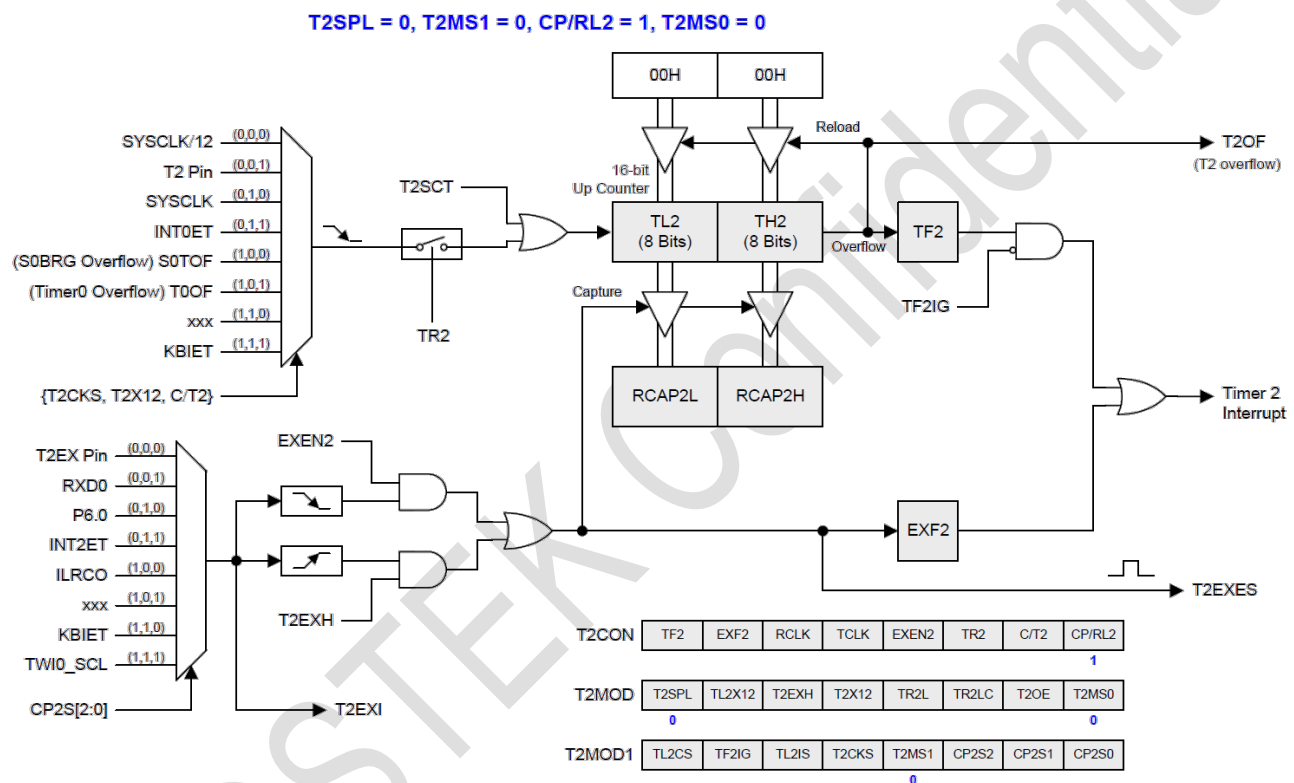
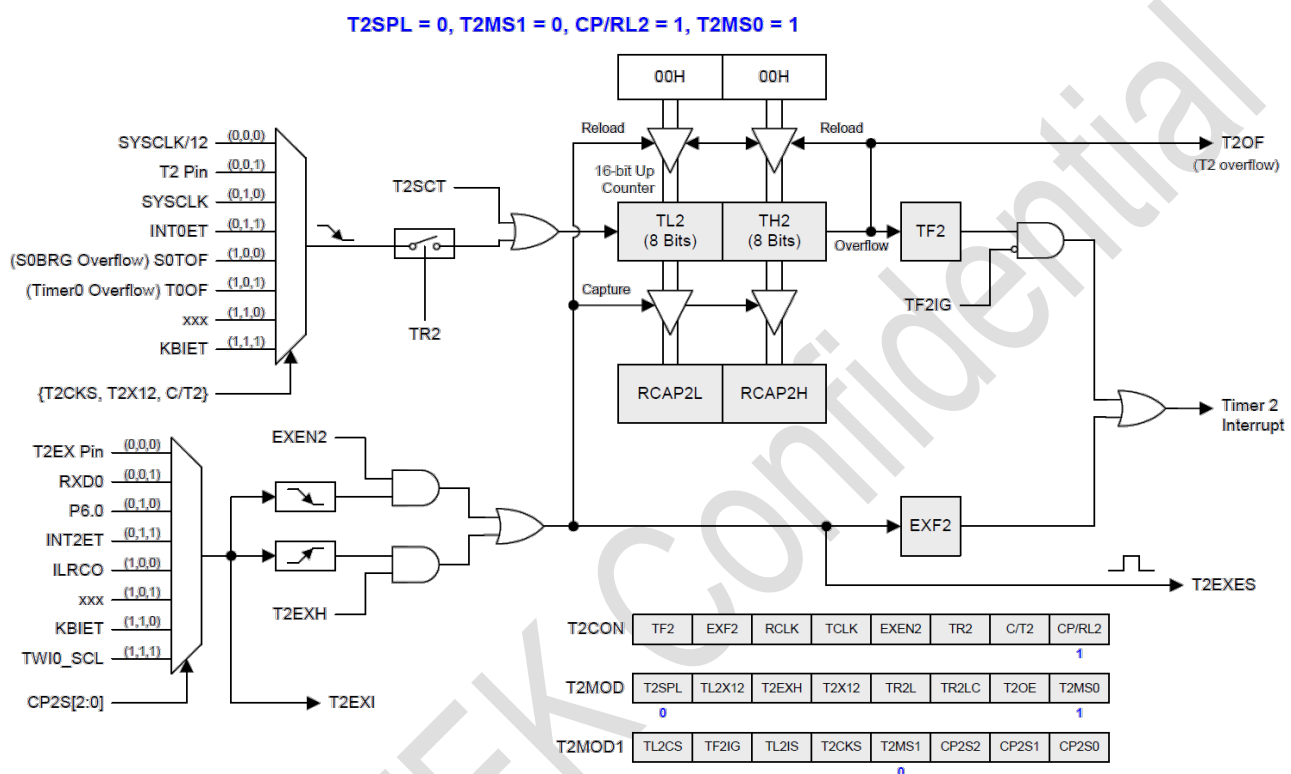


Figure 16-14. Timer 2 Mode 2 Structure (Capture Mode)



16.2.5 Split Timer 2 Mode 0 (AR and Ex. INT)

When T2SPL is set in this mode, Timer 2 operates as two 8-bit timers (TH2 and TL2). Both 8-bit timers operate in up-counter as shown in Figure 16-16. TH2 holds the reload value for RCAP2H and keep the same 8 clock source

inputs selection as 16-bit mode. It behaves the 8-bit function liked Timer 2 Mode 0 in 16-bit mode. TL2 holds the reload value for RCAP2L with 4 clock inputs selection. The TR2 bit in T2CON handles the run control for TH2. The TR2L bit in T2MOD handles the run control for TL2. And TH2 overflow can stop the TR2L running when TR2LC is set.

There are 3 interrupt flags in split mode, EXF2, TF2 and TF2L. EXF2 has the same function as 16-bit mode to detect the transition on T2EXI. TF2 is set when TH2 overflows from 0xFF to 0x00 with TF2IG control. TF2L is set when TL2 overflows from 0xFF to 0x00 with interrupt enabled by TL2IE. The EXF2, TF2 and TF2L interrupt flags are not cleared by hardware and must be cleared by software.

By the way, the Timer 2 overflow event (T2OF) in 16-bit timer is replaced by TL2 overflow event (TL2OF) in this split mode.

If TL2IS in T2MOD1 is 0, the bits on T2CON.5~4 are the function of RCLK and TCLK. If TL2IS is 1, the bits on T2CON.5~4 are the function of TF2L and TL2IE.

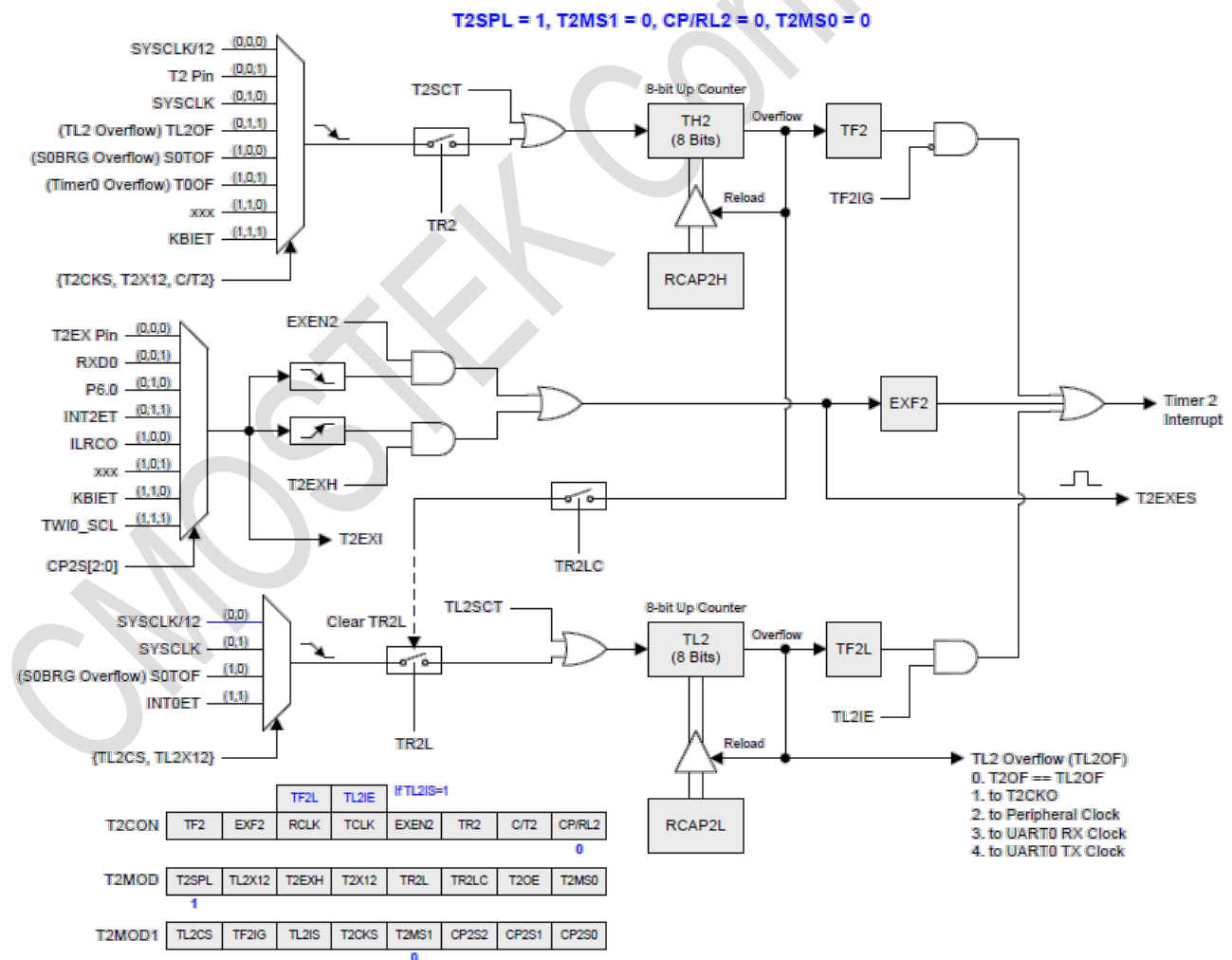


Figure 16-16. Split Timer 2 Mode 0 Structure (AR and Ex. INT)

16.2.6 Split Timer 2 Mode 1 (AR with Ex. INT)

When T2SPL is set in this mode, Timer 2 is split to two 8-bit timers as shown in Figure 16–17. It is similar function as Timer 2 Mode 1 and keeps the same interrupt scheme in Split Timer 2 Mode 0.

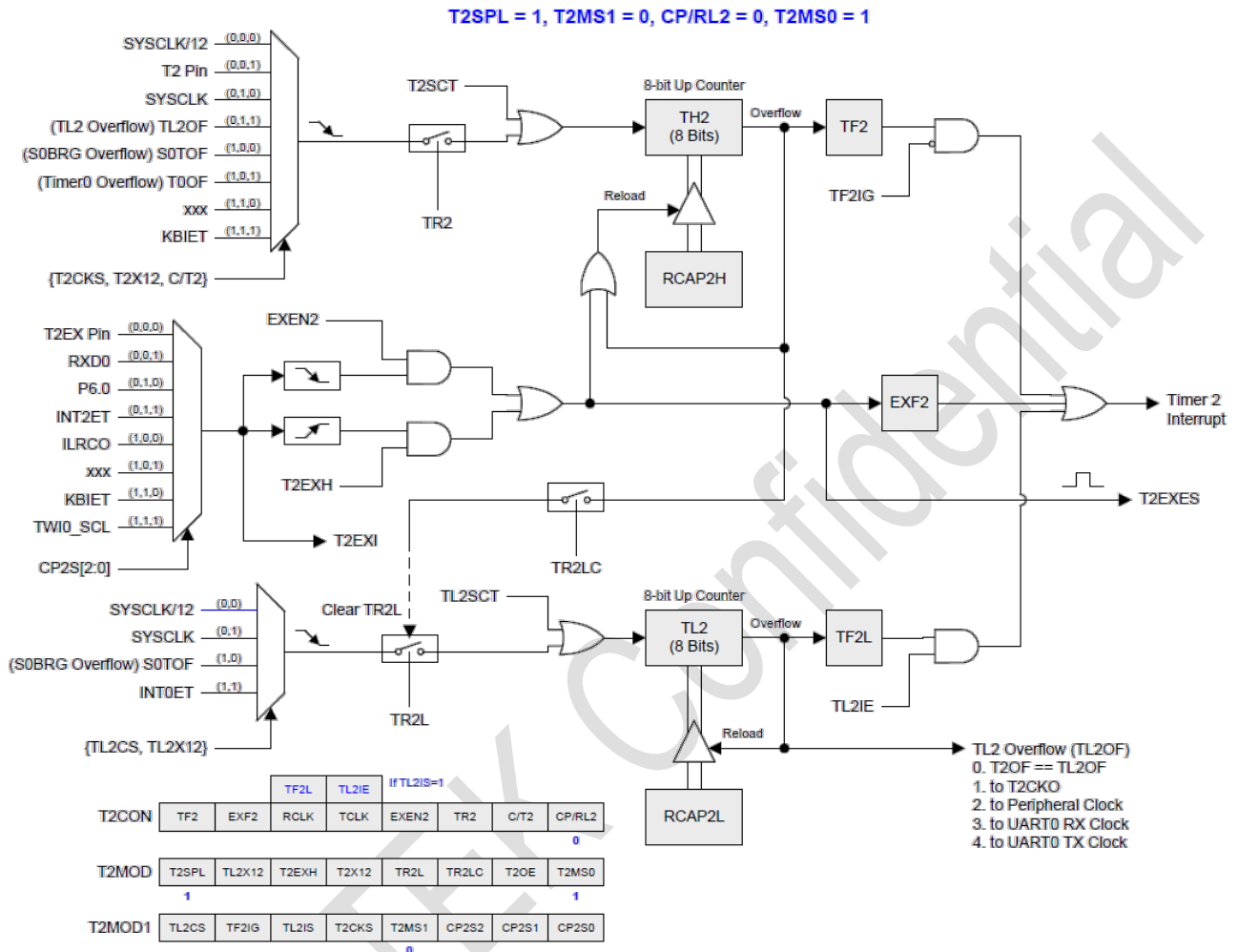


Figure 16-17. Split Timer 2 Mode 1 Structure (AR with Ex. INT)

16.2.7 Split Timer 2 Mode 2 (Capture)

When T2SPL is set in this mode, Timer 2 is split to two 8-bit timers as shown in Figure 16–18. It is similar function as Timer 2 Mode 2 and keeps the same interrupt scheme in Split Timer 2.

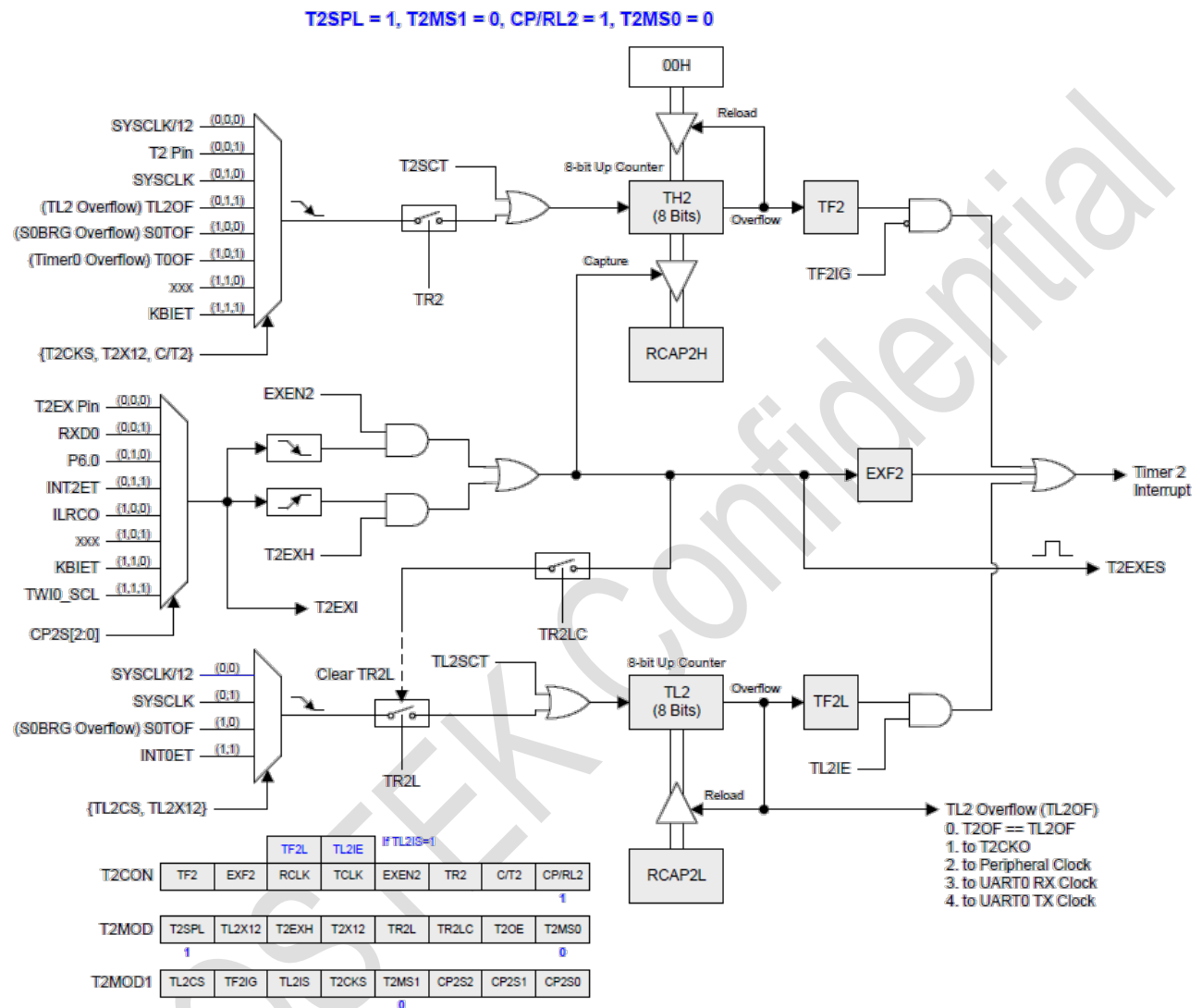


Figure 16-18. Split Timer 2 Mode 2 Structure (Capture)

16.2.8 Split Timer 2 Mode 3 (Capture with Auto-Zero)

When T2SPL is set in this mode, Timer 2 is split to two 8-bit timers as shown in Figure 16–19. It is similar function as Timer 2 Mode 3 and keeps the same interrupt scheme in Split Timer 2 Mode 0.

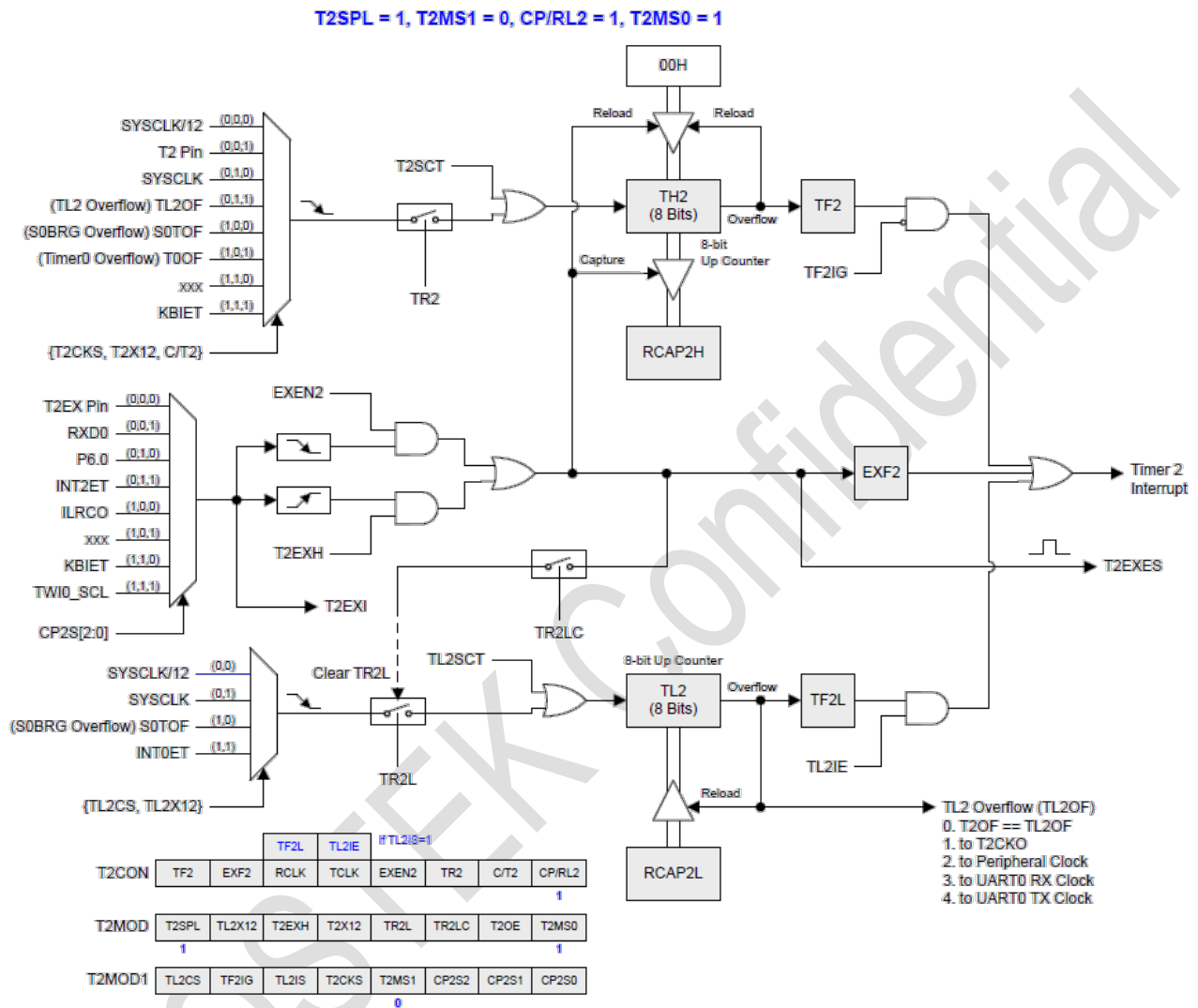


Figure 16-19. Split Timer 2 Mode 3 Structure (Capture with Auto-Zero on TH2)

16.2.9 Split Timer 2 Mode 4 (8-bit PWM Mode)

In this mode, Timer 2 is an 8-bit PWM mode as shown in Figure 16–20. TH2 and RCAP2H are combined to an 8-bit auto-reload counter. Software configures these two registers to decide the PWM cycle time. TL2 is the PWM compare register to generate PWM waveform. RCAP2L is the PWM buffer register and software will update PWM data in this register. Each TH2 overflow event will set TF2 and load RCAP2L value into TL2. The PWM signal will be output on T2CKO function pin and the output is gated by T2OE in T2MOD register.

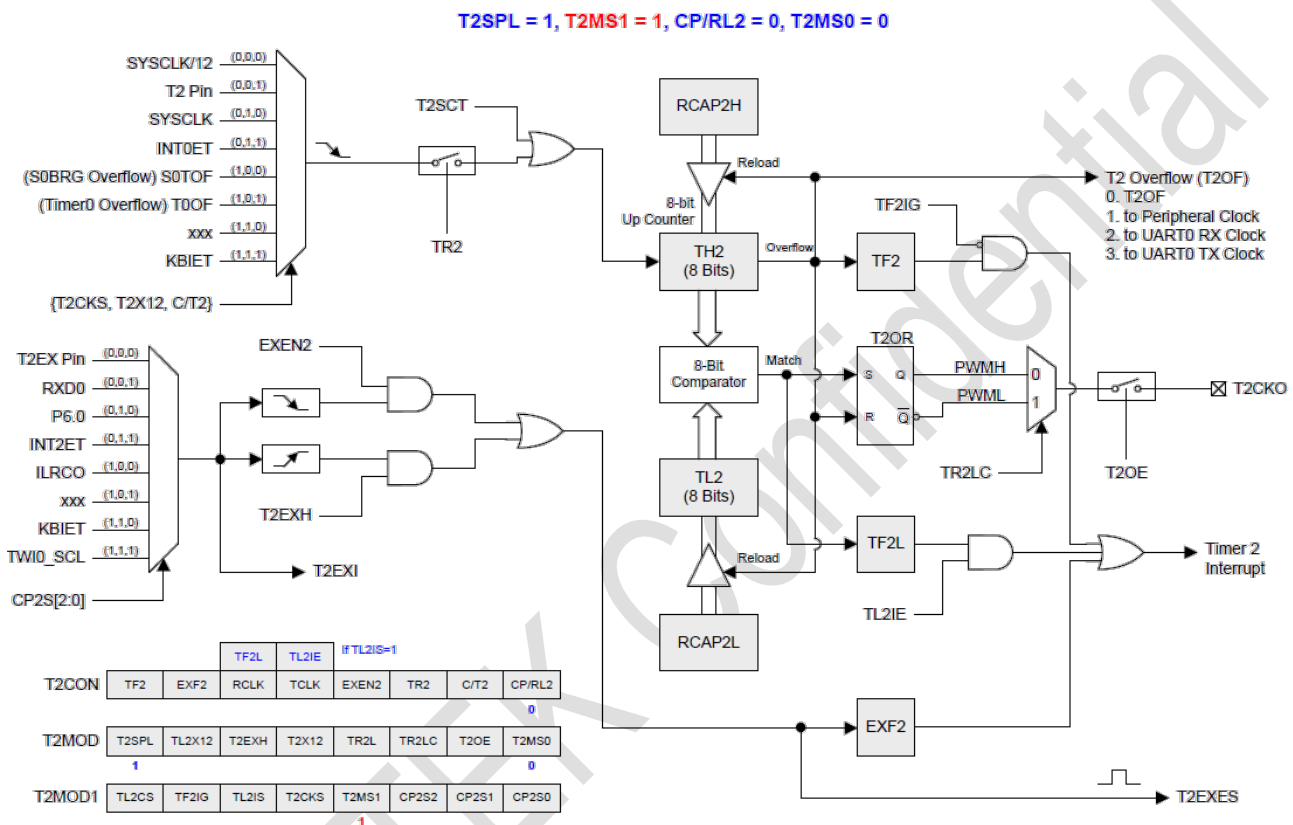


Figure 16-20. Split Timer 2 Mode 4 Structure (8-bit PWM mode)

16.2.10 Baud-Rate Generator Mode (BRG)

Bits TCLK and/or RCLK in T2CON register allow the serial port transmit and receive baud rates to be derived from either Timer 1 or Timer 2. When TCLK=0, Timer 1 is used as the serial port transmit baud rate generator. When TCLK=1, Timer 2 is used as the serial port transmit baud rate generator. RCLK has the same effect for the serial port receive baud rate. With these two bits, the serial port can have different receive and transmit baud rates – one generated by Timer 1, the other by Timer 2.

Figure 16–21 shows the Timer 2 in baud rate generation mode to generate RX Clock and TX Clock into UART engine (See Figure 18–6.). The baud rate generation mode is like the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by firmware.

The Timer 2 as a baud rate generator mode is valid only if RCLK and/or TCLK=1 in T2CON register. Note that a rollover in TH2 does set TF2. Thus, the Timer 2 interrupt does not have to be disabled when Timer 2 is in the baud rate generator mode by setting TF2IG to block TF2 interrupt. Also if the EXEN2 (T2 external enable bit) is set, a 1-to-0 transition in T2EX (Timer/counter 2 trigger input) will set EXF2 (T2 external flag) but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Therefore when Timer 2 is in use as a baud

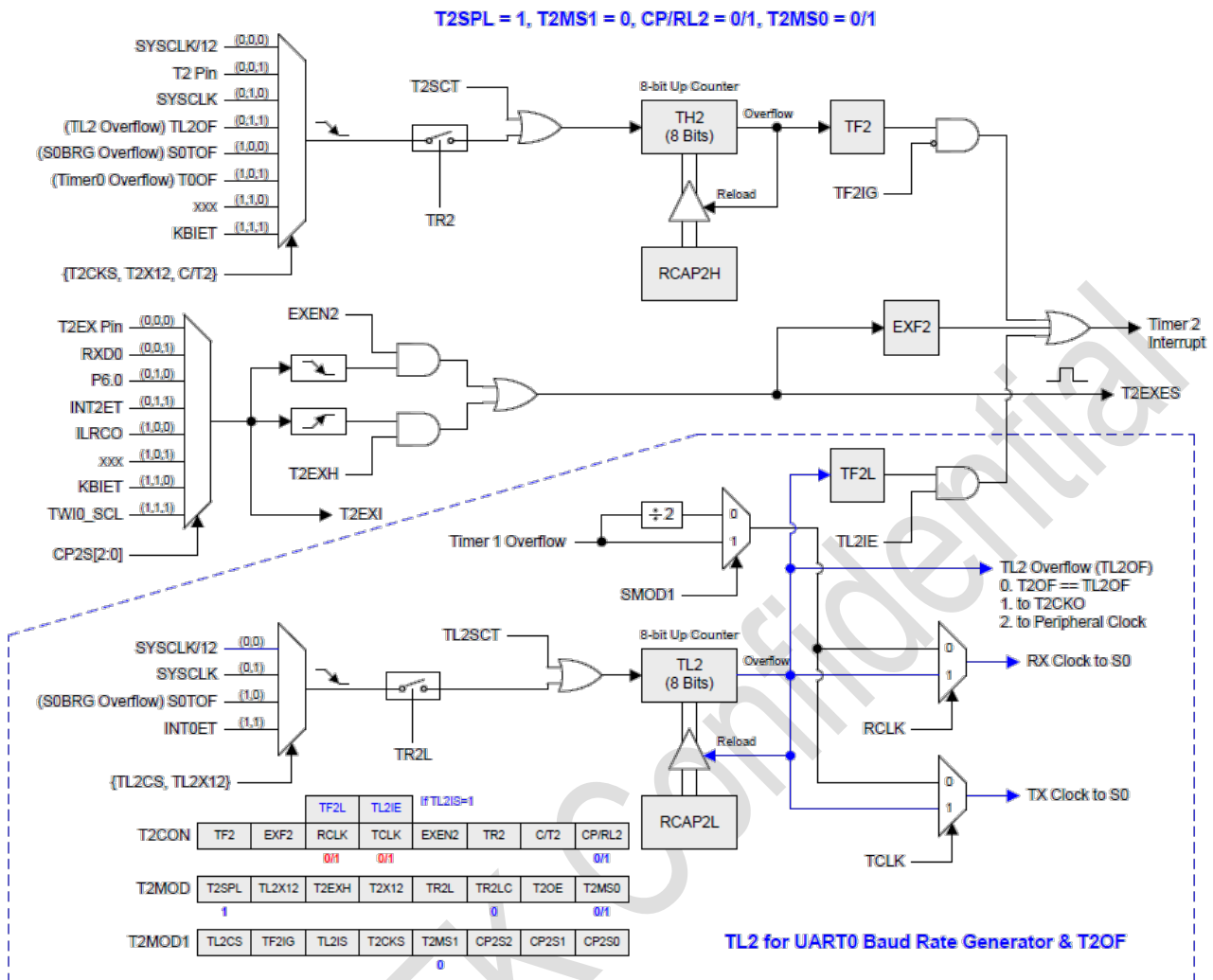


Figure 16-22. Split Timer 2 in Baud-Rate Generator Mode

16.2.11 Timer 2 Programmable Clock Output

Timer 2 has a Clock-Out Mode (while CP/RL2=0 & T2OE=1). In this mode, Timer 2 operates as a programmable clock generator with 50% duty-cycle. The generated clocks come out on P1.0. The input clock (SYSCLK/2 or SYSCLK) increments the 16-bit timer (TH2, TL2). The timer repeatedly counts to overflow from a loaded value. Once overflows occur, the contents of (RCAP2H, RCAP2L) are loaded into (TH2, TL2) for the consecutive counting. Figure 16–23 gives the formula of Timer 2 clock-out frequency: Figure 16–24 shows the clock structure of Timer 2.

$$\text{T2 Clock-out Frequency} = \frac{\text{T2 Clock Frequency}}{2 \times (65536 - (\text{RCAP2H}, \text{RCAP2L}))}$$

Figure 16-23. Timer 2 clock out equation

Note:

- (1) Timer 2 overflow flag, TF2, will be set when Timer 2 overflows to generate interrupt. But, the TF2 interrupt can be blocked by TF2IG in T2MOD1 register.
- (2) For SYSCLK=12MHz and select SYSCLK/12 as Timer 2 clock source, Timer 2 has a programmable output frequency range from 45.7Hz to 3MHz.
- (3) For SYSCLK=12MHz and select SYSCLK as Timer 2 clock source, Timer 2 has a programmable output

frequency range from 91.5Hz to 6MHz.

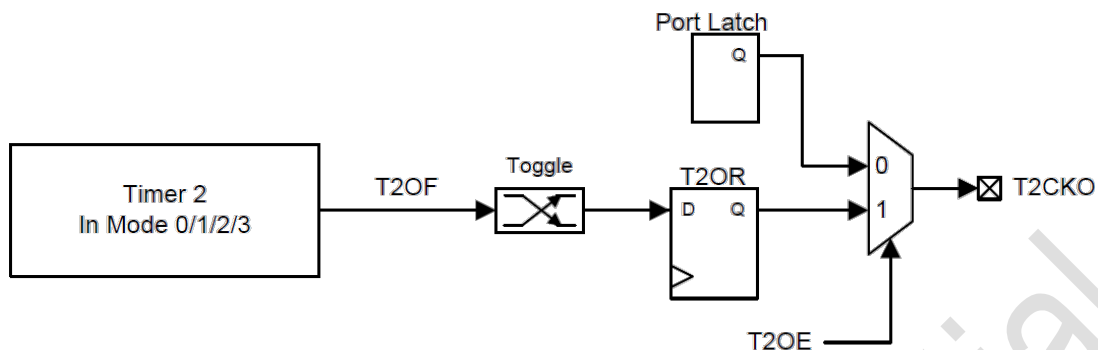


Figure 16-24. Timer 2 in Clock-Out Mode

Timer 2 in Clock-Out Mode

- Select Timer 2 clock source.
- Determine the 16-bit reload value from the formula and enter it in the RCAP2H and RCAP2L registers. Enter the same reload value as the initial value in the TH2 and TL2 registers.
- Set T2OE bit in T2MOD register.
- Set TR2 bit in T2CON register to start the Timer 2.

In the Clock-Out mode, Timer 2 rollovers will also generate a TF2 interrupt. This is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud rate generator and a clock generator simultaneously. Note, however, that the baud-rate and the clock-out frequency depend on the same overflow rate of Timer 2 and its interrupt will be blocked by TF2IG.

If Timer 2 in split mode, the clock output function is generated by TL2 overflow and the output clock frequency is TL2 overflow rate /2. RCAP2L is the TL2's reload value when TL2 overflow. There are four clock source selections for TL2. Before enable split Timer 2 clock output function, software must finish the TL2 clock source configuration. Figure 16–25 gives the formula of TL2 clock-out frequency: Figure 16–26 shows the clock structure of Split Timer 2.

$\text{Split T2 Clock-out Frequency} = \frac{\text{TL2 Clock Frequency}}{2 \times (256 - \text{RCAP2L})}$

Figure16-25. Split Timer 2 clock out equation

Note:

- (1) TL 2 overflow flag, TF2L, will be set when TL2 overflows to generate interrupt. But, the TF2L interrupt is enabled by TL2IE in T2CON register.
- (2) For SYSCLK=12MHz and select SYSCLK/12 as TL2 clock source, TL2 has a programmable output frequency range from 1.95KHz to 500KHz.
- (3) For SYSCLK=12MHz and select SYSCLK as TL2 clock source, TL2 has a programmable output frequency range from 23.44Hz to 6MHz.

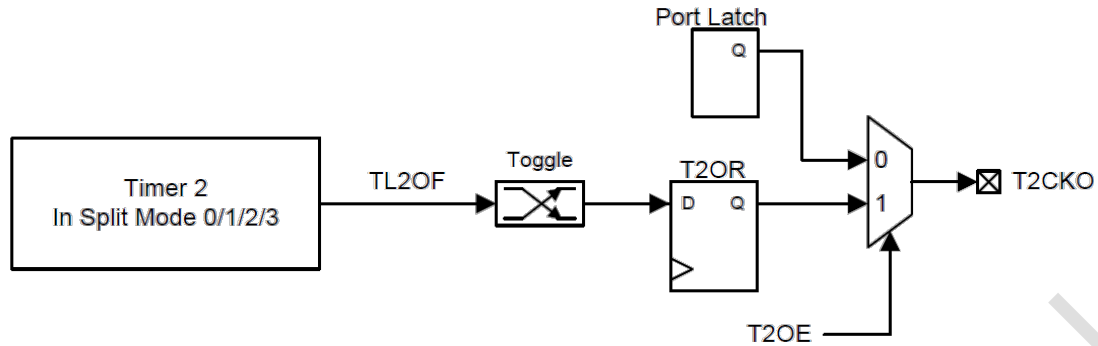


Figure 15-26. Split Timer 2 in Clock-Out Mode

How to Program Split Timer 2 in Clock-out Mode

- Select TL2 clock source.
- Determine the 8-bit reload value from the formula and enter it in the RCAP2L register. Enter the same reload value as the initial value in the TL2 register.
- Set T2OE bit in T2MOD register.
- Set TR2L bit in T2CON register to start the Timer 2.

In the Clock-Out mode, TL2 rollovers will also generate an interrupt, TF2L. This is similar to when TL2 is used as a baud-rate generator. It is possible to use TL2 as a baud rate generator and a clock generator simultaneously. Note, however, that the baud-rate and the clock-out frequency depend on the same overflow rate of TL2 in split Timer 2. The TF2L interrupt is enabled by TL2IE in T2CON register.

16.2.12 Timer 2 Register

- T2CON: Timer 2 Control Register

SFR Page = 0 Only

SFR Address = 0xC8

Bit	7	6	5	4	3	2	1	0
Name	TF2	EXF2	RCLK/ TF2L	TCLK/ TL2IE	EXEN2	TR2	C/T2	CP/RL2
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: TF2, Timer 2 overflow flag.

0: TF2 must be cleared by software.

1: TF2 is set by a Timer 2 overflow happens. TF2 will not be set when either RCLK=1 or TCLK=1.

Bit 6: EXF2, Timer 2 external flag.

0: EXF2 must be cleared by software.

1: Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX pin and EXEN2=1 or a positive transition on T2EX and T2EXH=1. When Timer 2 interrupt is enabled, EXF2=1 will cause the CPU to vector to the Timer 2 interrupt routine.

TL2IS (T2MOD1.5) must be cleared to enable access to the RCLK bit.

Bit 5: RCLK, Receive clock flag.

0: Causes Timer 1 overflow to be used for the receive clock.

1: Causes the serial port to use Timer 2 overflow pulses for its receive clock in modes 1 and 3.

TL2IS (T2MOD1.5) must be set to enable access to the TF2L bit.

Bit 5: TF2L, TL2 overflow flag in Timer 2 split mode.

0: TF2L must be cleared by software.

1: TF2L is set by TL2 overflow happened in Timer 2 split mode.

TL2IS (T2MOD1.5) must be cleared to enable access to the TCLK bit.

Bit 4: TCLK, Transmit clock flag.

0: Causes Timer 1 overflows to be used for the transmit clock.

1: Causes the serial port to use Timer 2 overflow pulses for its transmit clock in modes 1 and 3.

TL2IS (T2MOD1.5) must be set to enable access to the TL2IE bit.

Bit 4: TL2IE, TF2L interrupt enable.

0: Disable TF2L interrupt.

1: Enable TF2L interrupt to share the Timer 2 interrupt vector.

Bit 3: EXEN2, Timer 2 external enable flag on a negative transition of T2EX pin.

0: Cause Timer 2 to ignore negative transition events at T2EX pin.

1: Allows a capture or reload to occur as a result of a 1-to-0 transition on T2EX pin if Timer 2 is not being used to clock the serial port 0. If Timer 2 is configured to clock the serial port 0, the T2EX remains the external transition detection and reports on EXF2 flag with Timer 2 interrupt.

Bit 2: TR2, Timer 2 Run control bit. If in Timer 2 split mode, it only controls the TH2.

0: Disabled to stop the Timer/Counter 2.

1: Enabled to start the Timer/Counter 2.

Bit 1: C/T2, Timer 2 clock or counter source selector. The function is active with T2X12 and T2CKS as following definition:

T2CKS, T2X12, C/T2	Timer 2 Clock Selection	TH2 Clock Selection in split mode
0 0 0	SYSClk/12	SYSClk/12
0 0 1	T2Pin	T2Pin
0 1 0	SYSClk	SYSClk
0 1 1	INT0ET	TL2OF
1 0 0	S0TOF	S0TOF
1 0 1	T0OF	T0OF
1 1 0	Reserved	Reserved
1 1 1	KBIET	KBIET

Bit 0: CP/RL2, Timer 2 mode control bit. Refer T2MOD.T2MS0 description for the function definition.

- T2MOD: Timer 2 Mode Register

SFR Page = 0 Only

SFR Address = 0xC9

Bit	7	6	5	4	3	2	1	0
Name	T2SPL	TL2X12	T2EXH	T2X12	TR2L	TR2LC	T2OE	T2MS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: T2SPL, Timer 2 split mode control.

0: Disable Timer 2 to split mode.

1: Enable Timer 2 to split mode.

Bit 6: TL2X12, the clock control bit of TL2 in Timer 2 split mode.

TL2CS, TL2X12	TL2 Clock Selection
0 0	SYSClk/12
0 1	SYSClk
1 0	S0TOF
1 1	INT0ET

Bit 5: T2EXH, Timer 2 external enable flag on a positive transition of T2EX pin. 0: Cause Timer 2 to

ignore positive transition events at T2EX pin.

1: Allows a capture or reload to occur as a result of a 0-to-1 transition on T2EX pin if Timer 2 is not being used to clock the serial port 0. If Timer 2 is configured to clock the serial port 0, the T2EX remains the external transition detection and reports on EXF2 flag with Timer 2 interrupt.

Bit 4: T2X12, Timer 2 clock source selector. Refer to C/T2 description for the function defined. Bit 3: TR2L, TL2 Run control bit in Timer 2 split mode.

0: Disabled to stop the TL2.

1: Enabled to start the TL2.

Bit 2: TR2LC, TR2L Cleared control.

0: Disabled the TR2L cleared by hardware event.

1: Enabled the TR2L cleared by the TH2 overflow (Timer 2 in mode 0/1) or capture input (Timer 2 in mode 2/3).

Bit 1: T2OE, Timer 2 clock-out enable bit. 0: Disable Timer 2 clock output.

1: Enable Timer 2 clock output.

Bit 0: T2MS0, Timer 2 mode select bit 0.

T2MS1, CP/RL2, T2MS0	Timer 2 Mode Selection
0 0 0	Mode0:Auto-ReloadandExternalInterrupt
0 0 1	Mode 1:Auto-Reload with ExternalInterrupt
0 1 0	Mode2:Capturemode
0 1 1	Mode3:CapturewithAuto-Zero
1 0 0	Mode4:8-bitPWMifT2SPL=1
Others	Reserved

- T2MOD1: Timer 2 Mode Register 1

SFR Page = 1 Only

SFR Address = 0x93

Bit	7	6	5	4	3	2	1	0
Name	TL2CS	TF2IG	TL2IS	T2CK2	T2MS1	CP2S[2:0]		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: TL2CS. TL2 clock selection in Timer 2 split mode. Refer to T2MOD.TL2X12 description for the function defined. Bit 6: TF2IG, TF2 interrupt Ignored.

0: Enabled TF2 interrupt. Default is enabled.

1: Disable TF2 interrupt.

Bit 5: TL2IS, TF2L and TL2IE access control.

0: Enable RCLK and TCLK access function on T2CON.5~4.

1: Enable TF2L and TL2IE access function on T2CON.5~4.

Bit 4: T2CKS, Timer 2 clock selection. Refer to C/T2 description for the function defined.

Bit 3: T2MS1, Timer 2 mode selection bit 1. Refer T2MOD.T2MS0 description for the function definition.

Bit 2~0: CP2S.2~0. These bits define the capture source selector of Timer 2.

CP2S[2:0]	Timer 2 Capture Source Selection
0 0 0	T2EX Pin
0 0 1	RXD0
0 1 0	P6.0
0 1 1	INT2ET
1 0 0	Reserved
1 0 1	Reserved
1 1 0	KBIET
1 1 1	TWI0_SCL

- TL2: Timer 2 Low byte Register
SFR Page = 0 Only
SFR Address = 0xCC

Bit	7	6	5	4	3	2	1	0
Name	TL2[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

- TH2: Timer 2 High byte Register
SFR Page = 0 Only
SFR Address = 0xCD

Bit	7	6	5	4	3	2	1	0
Name	TH2[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

- RCAP2L: Timer 2 Capture Low byte Register
SFR Page = 0 Only
SFR Address = 0xCA

Bit	7	6	5	4	3	2	1	0
Name	RCAP2[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

- RCAP2H: Timer 2 Capture High byte Register
SFR Page = 0 Only
SFR Address = 0xCB

Bit	7	6	5	4	3	2	1	0
Name	RCAP2[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

- AUXR4: Auxiliary Register 4
SFR Page = 1 only
SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	T2PS[1:0]		T1PS1	T1PS0	SPIPS0	--	--	--
R/W	R/W	R/W	R/W	R/W	R/W	W	W	W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~6: T2PS1~0, Timer 2 Port pin Selection [1:0].

T2PS1~0	T2/T2CKO	T2EX
00	P1.0	P1.1
01	P3.0	P3.1
10	P6.0	P3.5
11	P4.5	P4.4

16.3 Timer 3

Timer 3 is a 16-bit Timer/Counter which can operate either as a timer or an event counter, as selected by C/T3 in T3CON register. Timer 3 has four operating modes: Capture, Auto-Reload (up or down counting) and Programmable Clock-Out, which are selected by bits in the T3CON, T3MOD and T3MOD1 registers.

Timer 3 Pin configuration are as following:

T3/T3CKO	T3EX
P3.3	P3.4

16.3.1 Timer 3 Mode 0 (Auto-Reload and External Interrupt)

In this mode, Timer 3 provides a 16-bit auto-reload timer/counter. The TF3, Timer 3 overflow flag, is one of the Timer 3 interrupt source which interrupt function can be blocked by TF3IG. EXEN3 enables a 1-to-0 transition at T3EXI to set the flag, EXF3, for an external input interrupt to share the Timer 3 interrupt with TF3. T3EXI is the selection result of 8 Timer 3 external inputs. T3EXH performs the same function as EXEN3 but it enables the detecting a 0-to-1 transition at T3EXI input.

The Timer 3 overflow event (T3OF) in this module will be output to other peripheral as clock input or event source.

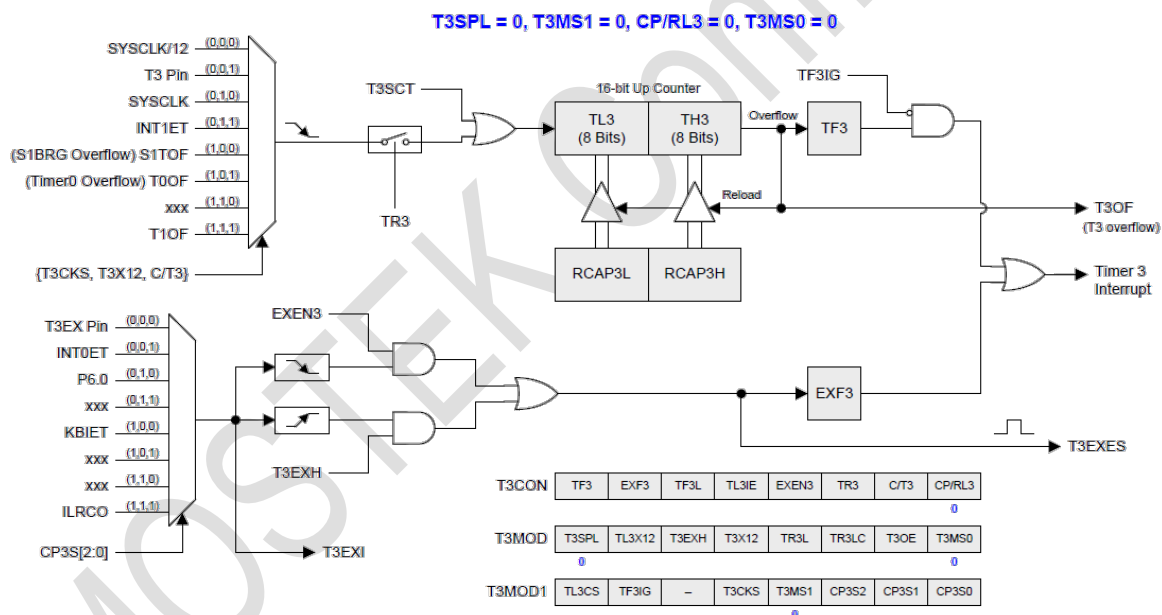


Figure 16-27. Timer 3 Mode 0 Structure (Auto-Reload and External Interrupt Mode)

16.3.2 Timer 3 Mode 1 (Auto-Reload with External Interrupt)

Figure 16-28 shows Timer 3 Mode 1, which enables Timer 3 to count up automatically. In this mode there are two options selected by bit EXEN3 in T3CON register. If EXEN2=0, then Timer 3 counts up to 0FFFFH and sets the TF3 (Overflow Flag) bit upon overflow. This causes the Timer 3 registers to be reloaded with the 16-bit value in RCAP3L and RCAP3H. The values in RCAP3L and RCAP3H are preset by firmware. If EXEN3=1, then a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at input T3EX. This transition also sets the EXF3 bit. The Timer 3 interrupt, if enabled, can be generated when either TF3 or EXF3 are 1. T3EXH performs the same function as EXEN3 but it enables the detecting a 0-to-1 transition at input T3EXI.

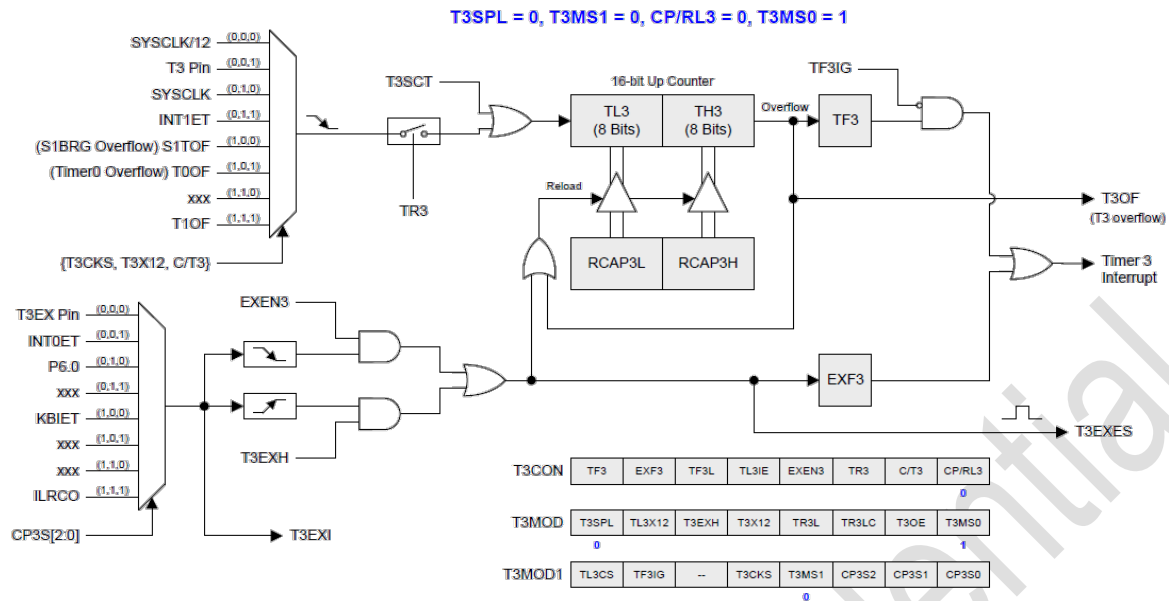


Figure 16-28. Timer 3 Mode 1 Structure (Auto-Reload with External Interrupt Mode)

16.3.3 Timer 3 Mode 2 (Capture)

Figure 16–29 shows the capture mode there are two options selected by bit EXEN3 in T3CON. If EXEN3=0, Timer 3 is a 16-bit timer or counter which, upon overflow, sets bit TF3 (Timer 3 overflow flag). This bit can then be used to generate an interrupt (by enabling the Timer 3 interrupt bit in the EIE2 register). If EXEN3=1, Timer 3 still does the above, but with the added feature that a 1-to-0 transition at T3EXI, one of 8 Timer 3 external inputs, that causes the current value in the Timer 3 registers, TH3 and TL3, to be captured into registers RCAP3H and RCAP3L, respectively. In addition, the transition at T3EXI causes bit EXF3 in T3CON to be set, and the EXF3 bit (like TF3) can generate an interrupt which vectors to the same location as Timer 3 overflow interrupt. T3EXH performs the same function as EXEN3 but it enables the detecting a 0-to-1 transition at T3EXI input.

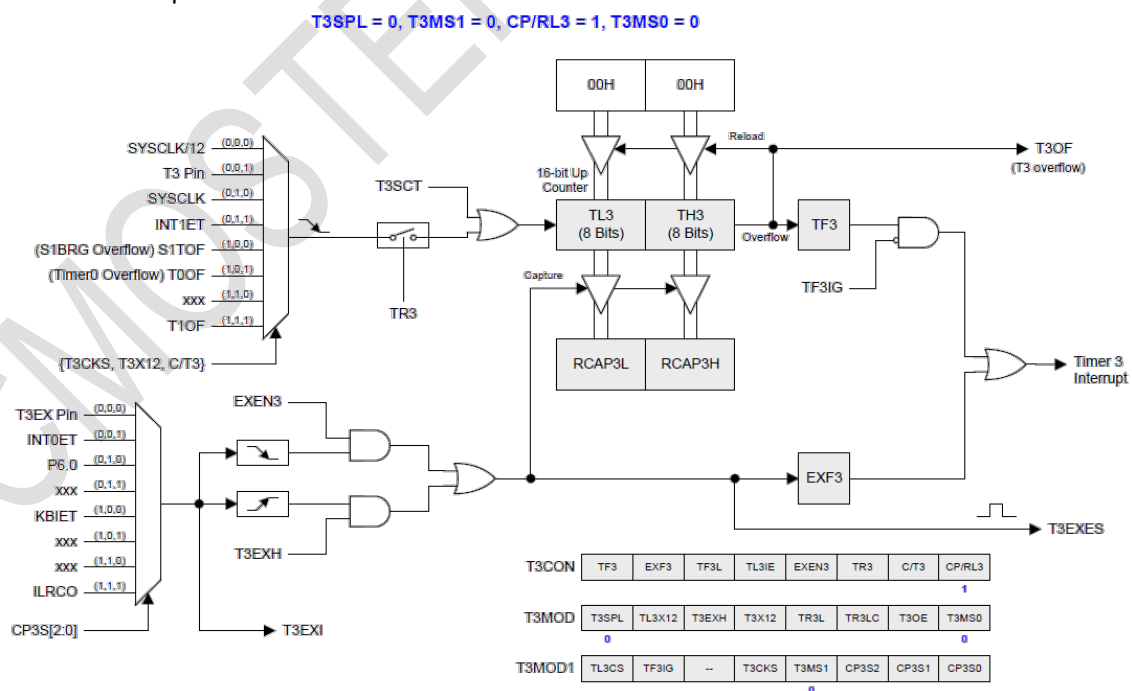


Figure 16-29. Timer 3 Mode 2 Structure (Capture Mode)

16.3.4 Timer 3 Mode 3 (Capture and Auto-Zero)

Timer 3 Mode 3 is the similar function with Timer 3 Mode 2. There is one difference that the T3EXES, EXF3 event set signal, not only is the capture source of Timer 3 but also clears the content of TL3 and TH3 to 0x0000H.

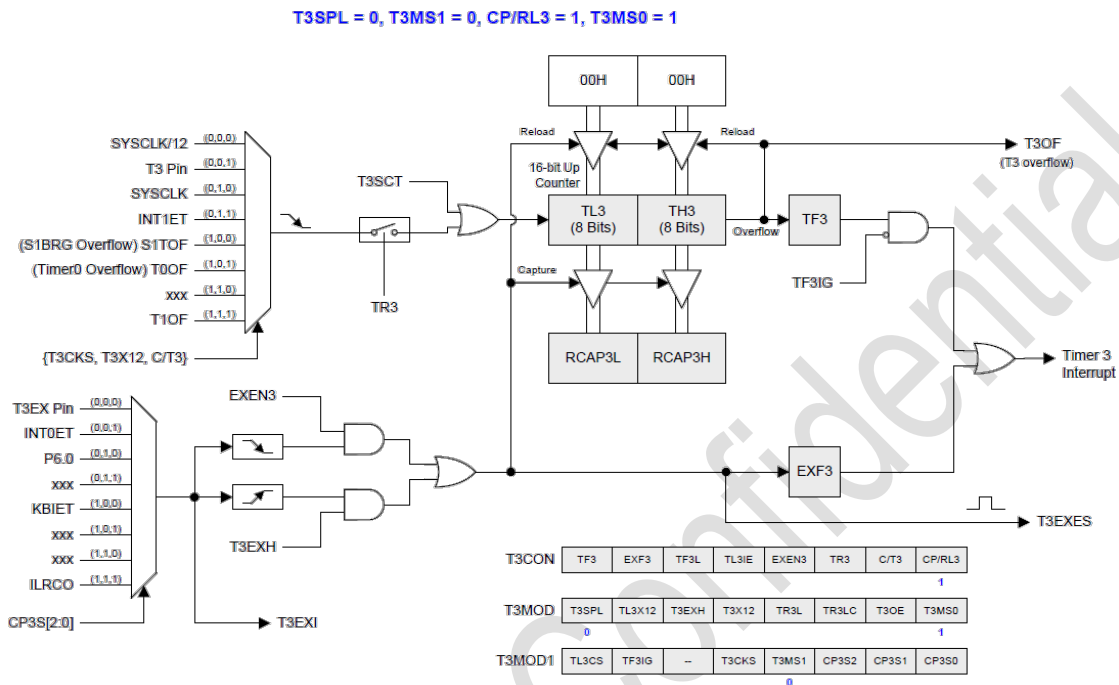


Figure 16-30. Timer 3 Mode 3 Structure (Capture with Auto-Zero on TL3 & TH3)

16.3.5 Split Timer 3 Mode 0 (Auto-Reload and External Interrupt)

When T3SPL is set in this mode, Timer 3 operates as two 8-bit timers (TH3 and TL3). Both 8-bit timers operate

in up counter as shown in Figure Figure 16–31 . TH3 holds the reload value for RCAP3H and keep the same 8 clock source inputs selection as 16-bit mode. It behaves the 8-bit function liked Timer 3 Mode 0 in 16-bit mode. TL3 holds the reload value for RCAP3L with 4 clock inputs selection. The TR3 bit in T3CON handles the run control for TH3. The TR3L bit in T3MOD handles the run control for TL3. And TH3 overflow can stop the TR3L running when TR3LC is set.

There are 3 interrupt flags in split mode, EXF3, TF3 and TF3L. EXF3 has the same function as 16-bit mode to detect the transition on T3EXI. TF3 is set when TH3 overflows from 0xFF to 0x00 with TF3IG control. TF3L is set when TL3 overflows from 0xFF to 0x00 with interrupt enabled by TL3IE. The EXF3, TF3 and TF3L interrupt flags are not cleared by hardware and must be cleared by software.

By the way, the Timer 3 overflow event (T3OF) in 16-bit timer is replaced by TL3 overflow event (TL3OF) in this split mode.

If TL3IS in T3MOD1 is 1, the bits on T3CON.5~4 are the function of TF3L and TL3IE.

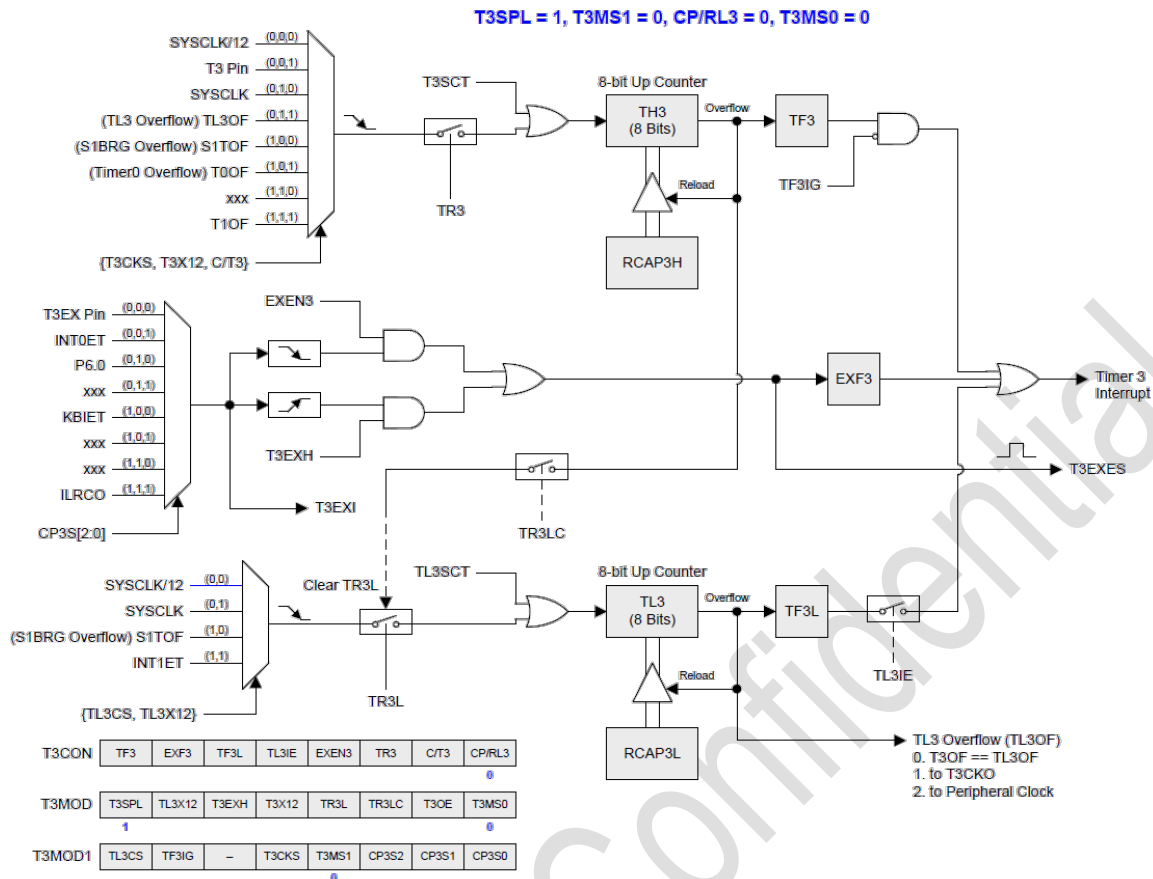


Figure 16-31. Split Timer 3 Mode 0 Structure (AR and Ex. INT).

16.3.6 Split Timer 3 Mode 1 (Auto-Reload with External Interrupt)

When T3SPL is set in this mode, Timer 3 is split to two 8-bit timers as shown in Figure 16–32. It is similar function as Timer 3 Mode 1 and keeps the same interrupt scheme in Split Timer 3 Mode 0.

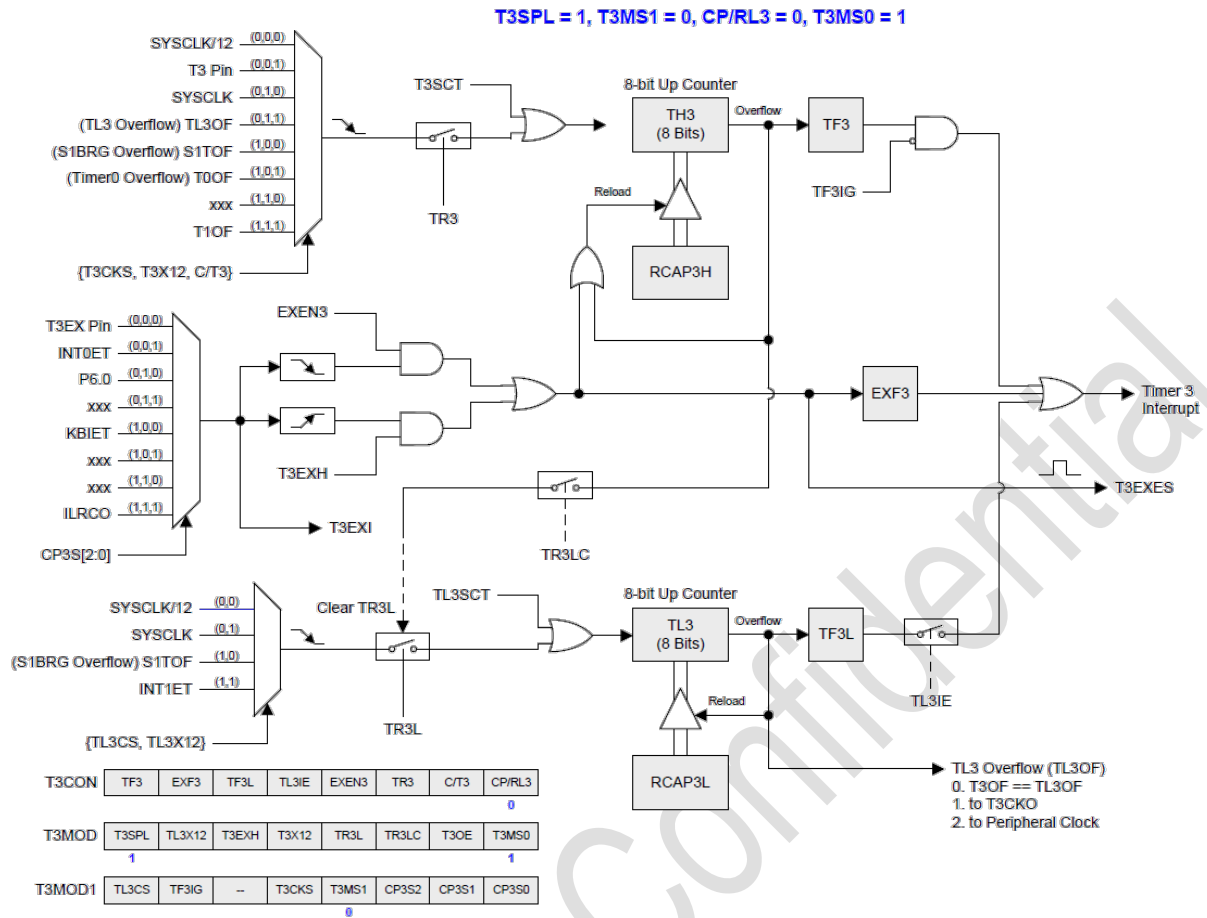


Figure 16-32. Split Timer 3 Mode 1 Structure (AR with Ex. INT)

16.3.7 Split Timer 3 Mode 2 (Capture)

When T3SPL is set in this mode, Timer 3 is split to two 8-bit timers as shown in Figure 16–33. It is similar function as Timer 3 Mode 2 and keeps the same interrupt scheme in Split Timer 3 Mode 0.

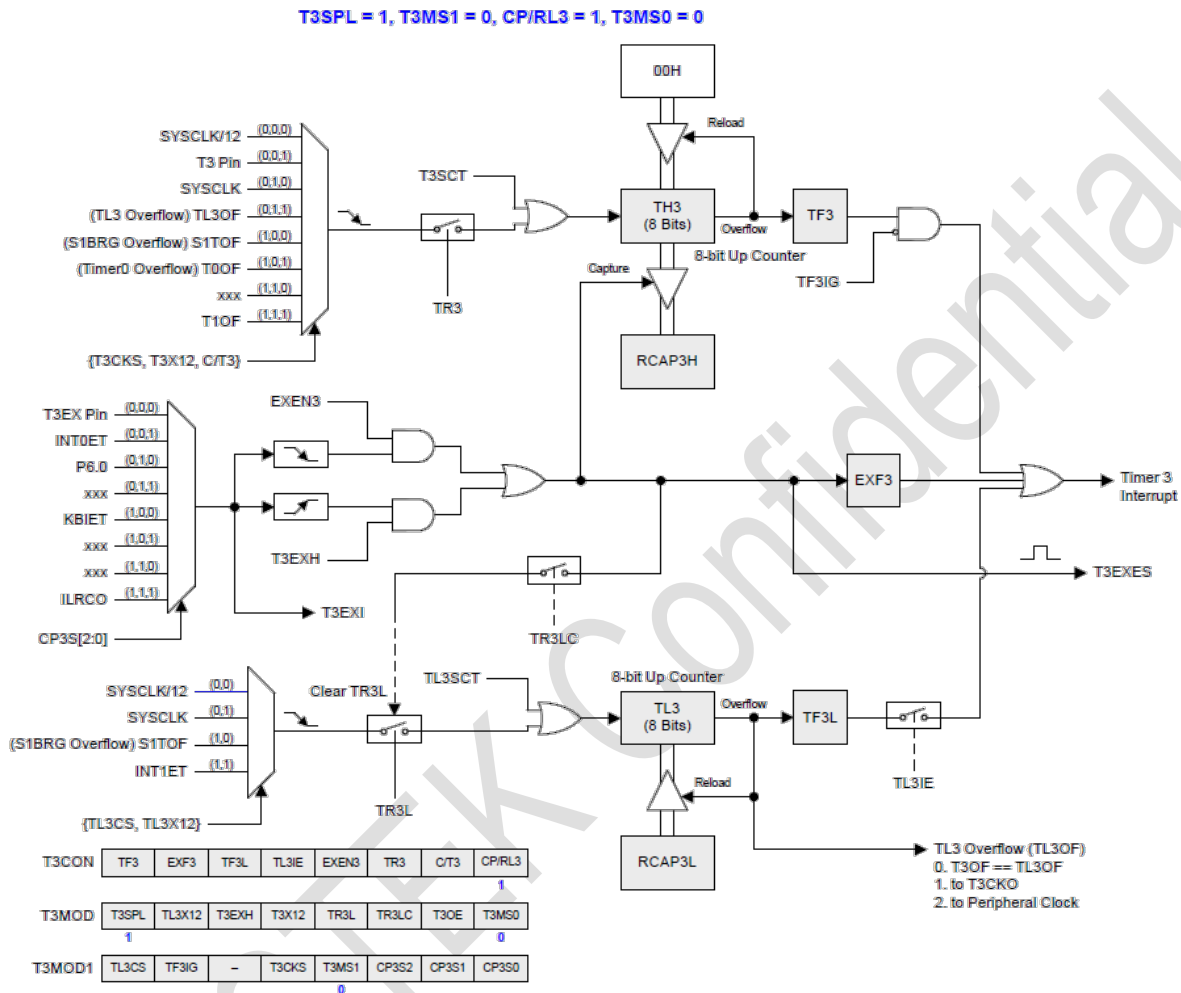


Figure 16-33. Split Timer 3 Mode 2 Structure (Capture)

16.3.8 Split Timer 3 Mode 3 (Capture with Auto-Zero)

When T3SPL is set in this mode, Timer 3 is split to two 8-bit timers as shown in Figure 16–34. It is similar function as Timer 3 Mode 3 and keeps the same interrupt scheme in Split Timer 3 Mode 0.

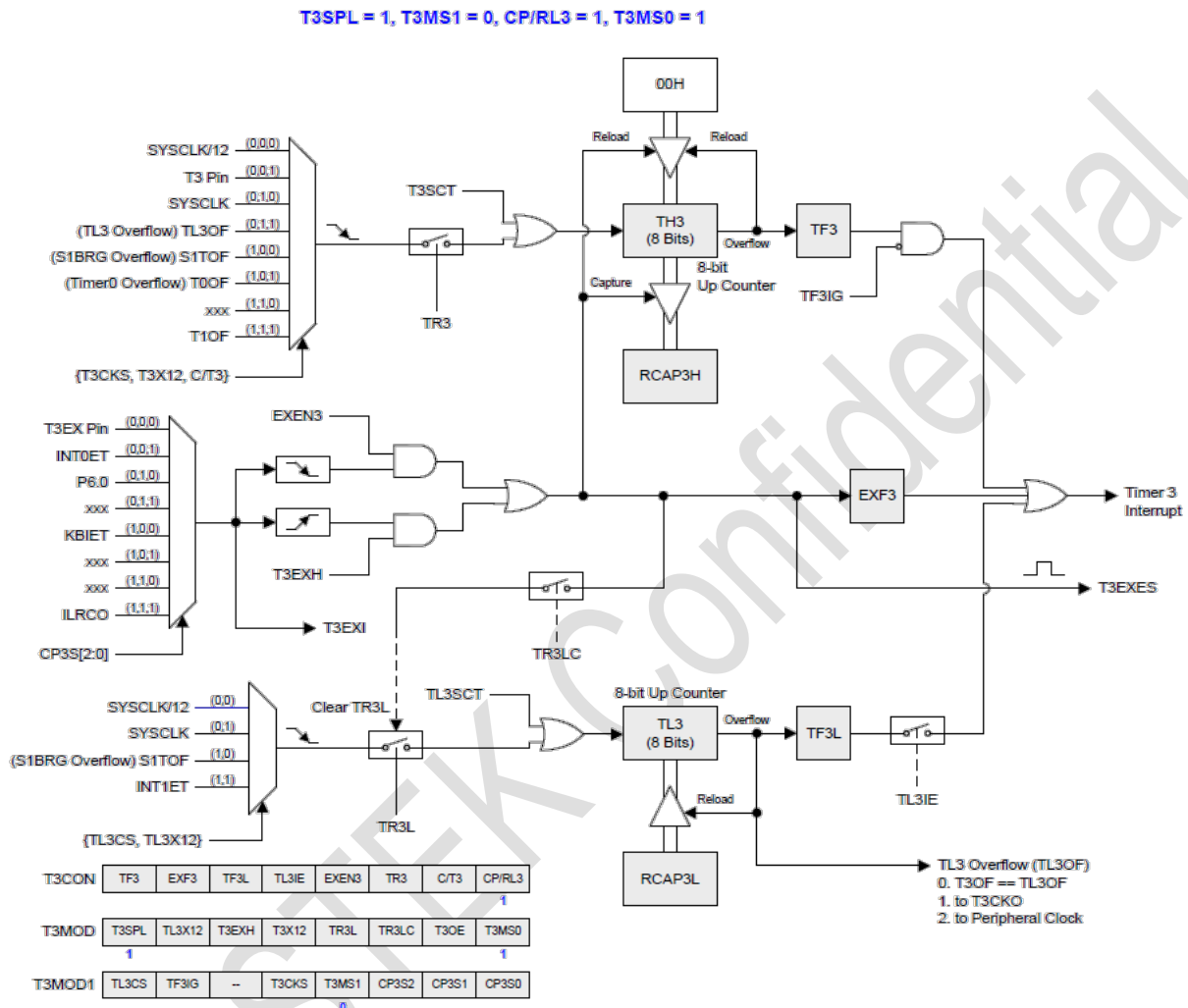


Figure 16-34. Split Timer 3 Mode 3 Structure (Capture with Auto-Zero on TH3)

16.3.9 Split Timer 3 Mode 4 (8-bit PWM Mode)

In this mode, Timer 3 is an 8-bit PWM mode as shown in Figure 16–35. TH3 and RCAP3H are combined to an 8-bit auto-reload counter. Software configures these two registers to decide the PWM cycle time. TL3 is the PWM compare register to generate PWM waveform. RCAP3L is the PWM buffer register and software will update PWM data in this register. Each TH3 overflow event will set TF3 and load RCAP3L value into TL3. The PWM signal will be output on T3CKO function pin and the output is gated by T3OE in T3MOD register.

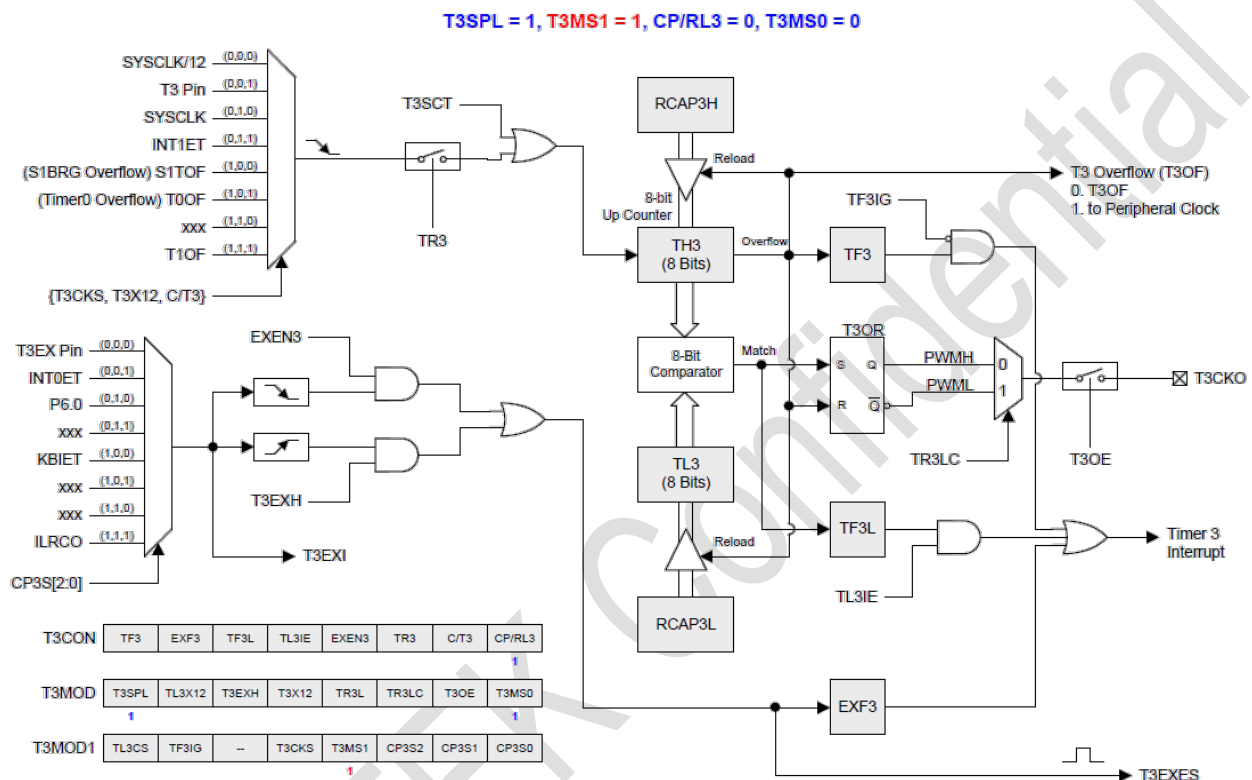


Figure 16-35. Split Timer 3 Mode 4 Structure (8-bit PWM mode)

16.3.10 Timer 3 Programmable Clock Output

Timer 3 has a Clock-Out Mode (while CP/RL3=0 & T3OE=1). In this mode, Timer 3 operates as a programmable clock generator with 50% duty-cycle. The generated clocks come out on T3CKO port pin. The input clock (SYSCLK/2 or SYSCLK) increments the 16-bit timer (TH3, TL3). The timer repeatedly counts to overflow from a loaded value. Once overflows occur, the contents of (RCAP3H, RCAP3L) are loaded into (TH3, TL3) for the consecutive counting. Figure 16–32 gives the formula of Timer 3 clock-out frequency: Figure 16–37 shows the clock structure of Timer 3.

$$\text{T3 Clock-out Frequency} = \frac{\text{T3 Clock Frequency}}{2 \times (65536 - (\text{RCAP3H}, \text{RCAP3L}))}$$

Figure 16-36. Timer 3 clock out equation

Note:

(1) Timer 3 overflow flag, TF3, will be set when Timer 3 overflows to generate interrupt. But, the TF3 interrupt can be blocked by TF3IG in T3MOD1 register.

- (2) For $\text{SYSCLK}=12\text{MHz}$ and select $\text{SYSCLK}/12$ as Timer 3 clock source, Timer 3 has a programmable output frequency range from 45.7Hz to 3MHz.
- (3) For $\text{SYSCLK}=12\text{MHz}$ and select SYSCLK as Timer 3 clock source, Timer 3 has a programmable output frequency range from 91.5Hz to 6MHz.

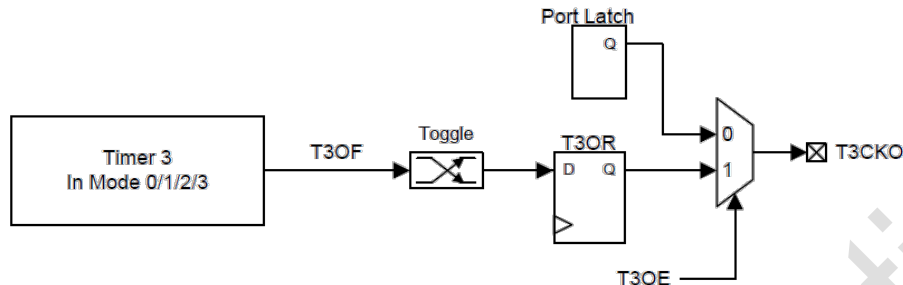


Figure 16-37. Timer 3 in Clock-Out Mode

How to Program Timer 3 in Clock-out Mode

- Select Timer 3 clock source.
- Determine the 16-bit reload value from the formula and enter it in the RCAP3H and RCAP3L registers.
- Enter the same reload value as the initial value in the TH3 and TL3 registers.
- Set T3OE bit in T3MOD register.
- Set TR3 bit in T3CON register to start the Timer 3.

In the Clock-Out mode, Timer 3 rollovers will also generate a TF3 interrupt. Its interrupt will be blocked by TF3IG.

If Timer 3 in split mode, the clock output function is generated by TL3 overflow and the output clock frequency is TL3 overflow rate /2. RCAP3L is the TL3's reload value when TL3 overflow. There are four clock source selections for TL3. Before enable split Timer 3 clock output function, software must finish the TL3 clock source configuration. Figure 16–25 gives the formula of TL3 clock-out frequency: Figure 16–26 shows the clock structure of Split Timer 3.

$$\text{Split T3 Clock-out Frequency} = \frac{\text{TL3 Clock Frequency}}{3 \times (256 - \text{RCAP3L})}$$

Figure 16-38. Split Timer 3 Clock Out Equation

Note:

- (1) TL3 overflow flag, TF3L, will be set when TL3 overflows to generate interrupt. But, the TF3L interrupt is enabled by TL3IE in T3CON register.
- (2) For $\text{SYSCLK}=12\text{MHz}$ and select $\text{SYSCLK}/12$ as TL3 clock source, TL3 has a programmable output frequency range from 1.95KHz to 500KHz.
- (3) For $\text{SYSCLK}=12\text{MHz}$ and select SYSCLK as TL3 clock source, TL3 has a programmable output frequency range from 23.44Hz to 6MHz.

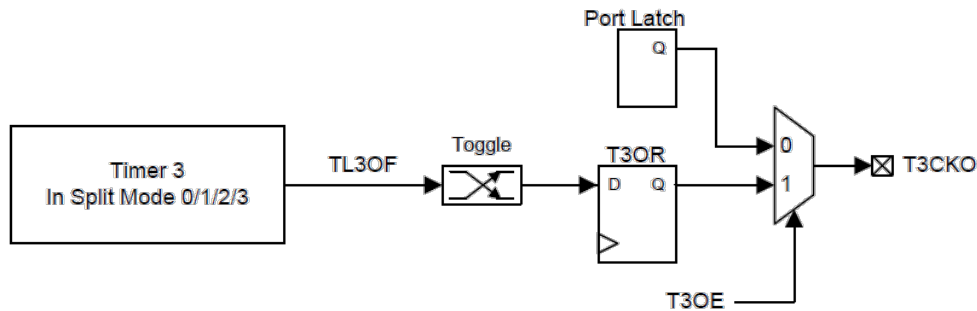


Figure 16-39. Split Timer 3 in Clock-Out Mode

How to Program Split Timer 3 in Clock-out Mode

- Select TL3 clock source.
- Determine the 8-bit reload value from the formula and enter it in the RCAP3L register.
- Enter the same reload value as the initial value in the TL3 register.
- Set T3OE bit in T3MOD register.
- Set TR3L bit in T3CON register to start the Timer 3.

In the Clock-Out mode, TL3 rollovers will not generate an interrupt, TF3L. This is similar to when TL3 is used as a baud-rate generator. It is possible to use TL3 as a baud rate generator and a clock generator simultaneously. Note, however, that the baud-rate and the clock-out frequency depend on the same overflow rate of TL3 in split Timer 3. The TF3L interrupt is enabled by TL3IE in T3CON register.

16.3.11 Timer 3 Register

- T3CON: Timer 3 Control Register

SFR Page = 1 Only

SFR Address = 0xC8

Bit	7	6	5	4	3	2	1	0
Name	TF3	EXF3	TF3L	TL3IE	EXEN3	TR3	C/T3	CP/RL3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: TF3, Timer 3 overflow flag.

0: TF3 must be cleared by software.

1: TF3 is set by a Timer 3 overflow happens.

Bit 6: EXF3, Timer 3 external flag.

0: EXF3 must be cleared by software.

1: Timer 3 external flag set when either a capture or reload is caused by a negative transition on T3EX pin and EXEN3=1 or a positive transition on T3EX and T3EXH=1. When Timer 3 interrupt is enabled, EXF3=1 will cause the CPU to vector to the Timer 3 interrupt routine. When the MCU is in power-down mode and Timer 3 interrupt is enabled, the EXF3 is forced to level-sensitive triggered with wake-up MCU capability.

Bit 5: TF3L, TL3 overflow flag in Timer 3 split mode.

0: TF3L must be cleared by software.

1: TF3L is set by TL3 overflow happened in Timer 3 split mode.

Bit 4: TL3IE, TF3L interrupt enable.

0: Disable TF3L interrupt.

1: Enable TF3L interrupt to share the Timer 3 interrupt vector.

Bit 3: EXEN3, Timer 3 external enable flag on a negative transition of the Timer 3 external input.

0: Cause Timer 3 to ignore negative transition events at Timer 3 external input.

1: Allows a capture or reload to occur as a result of a 1-to-0 transition on Timer 3 external input. If Timer 3

is configured to mode 0 which does not behave capture or reload function, the Timer 3 external input remains the external transition detection and reports on EXF3 flag with Timer 3 interrupt.

Bit 2: TR3, Timer 3 Run control bit. If in Timer 3 split mode, it only controls the TH3.

0: Disabled to stop the Timer/Counter 3.

1: Enabled to start the Timer/Counter 3.

Bit 1: C/T3, Timer 3 clock or counter source selector. The function is active with T3X12 and T3CKS as following definition:

T3CKS, T3X12, C/T3	Timer 3 Clock Selection	TH3 Clock Selection in split mode
0 0 0	SYSCLK/12	SYSCLK/12
0 0 1	T3 Pin	T3 Pin
0 1 0	SYSCLK	SYSCLK
0 1 1	INT1ET	TL3OF
1 0 0	S1TOF	S1TOF
1 0 1	T0OF	T0OF
1 1 0	Reserved	Reserved
1 1 1	T1OF	T1OF

Bit 0: CP/RL3, Timer 3 mode control bit. Refer T3MOD.T3MS0 description for the function definition.

- T3MOD: Timer 3 Mode Register

SFR Page = 1 Only

SFR Address = 0xC9

Bit	7	6	5	4	3	2	1	0
Name	T3SPL	TL3X12	T3EXH	T3X12	TR3L	TR3LC	T3OE	T3MS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: T3SPL, Timer 3 split mode control.

0: Disable Timer 3 to split mode.

1: Enable Timer 3 to split mode.

Bit 6: TL3X12, the clock control bit of TL3 in Timer 3 split mode.

TL3CS, TL3X12	TL3 Clock Selection
0 0	SYSCLK/12
0 1	SYSCLK
1 0	S1TOF
1 1	INT1ET

Bit 5: T3EXH, Timer 3 external enable flag on a positive transition of T3EX pin.

0: Cause Timer 3 to ignore positive transition events at T3EX pin.

1: Allows a capture or reload to occur as a result of a 0-to-1 transition on T3EX pin and set EXF3.

Bit 4: T3X12, Timer 3 clock source selector. Refer to C/T3 description for the function defined.

Bit 3: TR3L, TL3 Run control bit in Timer 3 split mode.

0: Disabled to stop the TL3.

1: Enabled to start the TL3.

Bit 2: TR3LC, TR3L Cleared control.

0: Disabled the TR3L cleared by hardware event.

1: Enabled the TR3L cleared by the TH3 overflow (Timer 3 in mode 0/1) or capture input (Timer 3 in mode 2/3).

Bit 1: T3OE, Timer 3 clock-out enable bit. 0: Disable Timer 3 clock output.

1: Enable Timer 3 clock output.

Bit 0: T3MS0, Timer 3 mode select bit 0.

T3MS1, CP/RL3, T3MS0	Timer 3 Mode Selection
0 0 0	Mode0:Auto-ReloadandExternalInterrupt
0 0 1	Mode 1:Auto-Reload with ExternalInterrupt
0 1 0	Mode2:Capturemode
0 1 1	Mode3:CapturewithAuto-Zero
1 0 0	Mode4:8-bitPWMifT3SPL=1
Others	Reserved

T3MOD1: Timer 3 Mode Register 1

SFR Page = 2 Only

SFR Address = 0x93

Bit	7	6	5	4	3	2	1	0
Name	TL3CS	TF3IG	--	T3CK2	T3MS1	CP3S[2:0]		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: TL3CS. TL3 clock selection in Timer 3 split mode. Refer to T3MOD.TL3X12 description for the function defined. Bit 6: TF3IG, TF3 interrupt Ignored.

0: Enabled TF3 interrupt. Default is enabled.

1: Disable TF3 interrupt.

Bit 5: Reserved. Software must write "0" on this bit when T3MOD1 is written.

Bit 4: T3CKS, Timer 3 clock selection. Refer to C/T3 description for the function defined.

Bit 3: T3MS1, Timer 3 mode selection bit 1. Refer T3MOD.T3MS0 description for the function definition.

Bit 2~0: CP3S.2~0. These bits define the capture source selector of Timer 3.

CP3S[2:0]	Timer 3 Capture Source Selection
0 0 0	T3EXPIn
0 0 1	INT0ET
0 1 0	P6.0
0 1 1	Reserved
1 0 0	KBIET
1 0 1	Reserved
1 1 0	Reserved
1 1 1	ILRCO

- TL3: Timer 3 Low byte Register

SFR Page = 1 Only

SFR Address = 0xCC

Bit	7	6	5	4	3	2	1	0
Name	TL2[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

- TH3: Timer 3 High byte Register

SFR Page = 1 Only

SFR Address = 0xCD

Bit	7	6	5	4	3	2	1	0
-----	---	---	---	---	---	---	---	---

Name	TH2[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

- RCAP3L: Timer 3 Capture Low byte Register

SFR Page = 1 Only

SFR Address = 0xCA

Bit	7	6	5	4	3	2	1	0
Name	RCAP2[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

- RCAP3H: Timer 3 Capture High byte Register

SFR Page = 1 Only

SFR Address = 0xCB

Bit	7	6	5	4	3	2	1	0
Name	RCAP2[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

16.4 Timer Global Control

When the applications are asking all timers work together in sync mode, it can set the registers to Start, Reload and Stop the timers.

16.4.1 Global Enable for all Timer Run

When the applications are asking all timers work together in sync mode, just need to set the TRxE or TRxLE in TREN0 to start the timer at the same time. Those registers will be auto cleared by hardware after writing “1” into it.

- **TREN0:** Timer Run Enable Register 0
SFR Page = 1 Only
SFR Address = 0x95 Reset Value=0000-0000

Bit	7	6	5	4	3	2	1	0
Name	--	TR3LE	TR2LE	--	TR3E	TR2E	TR1E	TR0E
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: Reserved. Software must write “0” on this bit when TREN0 is written.

Bit 6, TR3LE, write “1” on this bit to set TR3L enabled (TR3L=1) when Timer 3 in split mode. This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

Bit 5, TR2LE, write “1” on this bit to set TR2L enabled (TR2L=1) when Timer 2 in split mode. This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

Bit 4: Reserved. Software must write “0” on this bit when TREN0 is written.

Bit 3, TR3E, write “1” on this bit to set TR3 enabled (TR3=1). This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

Bit 2, TR2E, write “1” on this bit to set TR2 enabled (TR2=1). This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

Bit 1, TR1E, write “1” on this bit to set TR1 enabled (TR1=1). This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

Bit 0, TR0E, write “1” on this bit to set TR0 enabled (TR0=1). This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

16.4.2 Global Control for all Timer Reload

- **TRLC0:** Timer Reload Control Register 0

SFR Page = 2 Only

SFR Address = 0x95 RESET= 0000-0000

Bit	7	6	5	4	3	2	1	0
Name	--	TL3RLC	TL2RLC	--	T3RLC	T2RLC	T1RLC	T0RLC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: Reserved. Software must write "0" on this bit when TRLC0 is written.

Bit 6, TL3RLC, write "1" on this bit to force TL3 reload condition happened when Timer 3 in split mode. This bit is auto-cleared by hardware after writing "1" operation. Write "0" on this bit is no action.

Bit 5, TL2RLC, write "1" on this bit to force TL2 reload condition happened when Timer 2 in split mode. This bit is auto-cleared by hardware after writing "1" operation. Write "0" on this bit is no action.

Bit 4: Reserved. Software must write "0" on this bit when TRLC0 is written.

Bit 3, T3RLC, write "1" on this bit to force TH3 and TL3 reload condition happened when Timer 3 not in split mode. Or force TH3 reload condition happened when Timer 3 in split mode. The force reload is not available if the timer in capture mode. This bit is auto-cleared by hardware after writing "1" operation. Write "0" on this bit is no action.

Bit 2, T2RLC, write "1" on this bit to force TH2 and TL2 reload condition happened when Timer 2 not in split mode. Or force TH2 reload condition happened when Timer 2 in split mode. The force reload is not available if the timer in capture mode. This bit is auto-cleared by hardware after writing "1" operation. Write "0" on this bit is no action.

Bit 1, T1RLC, write "1" on this bit to force TH1/TL1 reload condition happened. This bit is auto-cleared by hardware after writing "1" operation. Write "0" on this bit is no action.

Bit 0, T0RLC, write "1" on this bit to force TH0/TL0 reload condition happened. This bit is auto-cleared by hardware after writing "1" operation. Write "0" on this bit is no action.

16.4.3 Global Control for all Timer Stop

- **TSPC0:** Timer Stop Control Register 0

SFR Page = 3 Only

SFR Address = 0x95 RESET= 0000-0000

Bit	7	6	5	4	3	2	1	0
Name	--	TL3SC	TL2SC	--	T3SC	T2SC	T1SC	T0SC
R/W	R/W	R/W	R/W	R/W	R/W	W	R/W	R/W
Reset Value	0	0	0	0	0	X	0	0

Bit 7: Reserved. Software must write "0" on this bit when TSPC0 is written.

Bit 6, TL3SC, write "1" on this bit to set TR3L disabled (TR3L=0) when Timer 3 in split mode. This bit is auto-cleared by hardware after writing "1" operation. Write "0" on this bit is no action.

Bit 5, TL2SC, write "1" on this bit to set TR2L disabled (TR2L=0) when Timer 2 in split mode. This bit is auto-cleared by hardware after writing "1" operation. Write "0" on this bit is no action.

Bit 4: Reserved. Software must write "0" on this bit when TSPC0 is written.

Bit 3, T3SC, write "1" on this bit to set TR3 disabled (TR3=0). This bit is auto-cleared by hardware after writing "1" operation. Write "0" on this bit is no action.

Bit 2, T2SC, write "1" on this bit to set TR2 disabled (TR2=0). This bit is auto-cleared by hardware after writing "1" operation. Write "0" on this bit is no action.

Bit 1, T1SC, write "1" on this bit to set TR1 disabled (TR1=0). This bit is auto-cleared by hardware after writing "1" operation. Write "0" on this bit is no action.

Bit 0, T0SC, write "1" on this bit to set TR0 disabled (TR0=0). This bit is auto-cleared by hardware after writing "1" operation. Write "0" on this bit is no action.

17 Programmable Counter Array (PCA0)

The CMT2380F17 is equipped with a Programmable Counter Array (PCA0), which provides more timing capabilities with less CPU intervention than the standard timer/counters. Its advantages include reduced software overhead and improved accuracy.

17.1 PCA Overview

The PCA consists of a dedicated timer/counter which serves as the time base for an array of Six capture/compare/PWM modules and Two compare/PWM modules. Figure 17–1 shows a block diagram of the PCA. Notice that the PCA timer and modules are all 16-bits. If an external event is associated with a module, that function is shared with the corresponding Port pin. If the module is not using the port pin, the pin can still be used for standard I/O.

Module 0~5 can be programmed in any one of the following modes::

- Rising and/or Falling Edge Capture
- Software Timer (Compare)
- High Speed Output (Compare Output)
- Pulse Width Modulator Output (PWM)
- Compare Output on PWM Match case (COPM)

Module 6~7 also support the upper modes except Rising and/or Falling Edge capture. All of these modes will be discussed later in detail. However, let's first look at how to set up the PCA timer and modules.

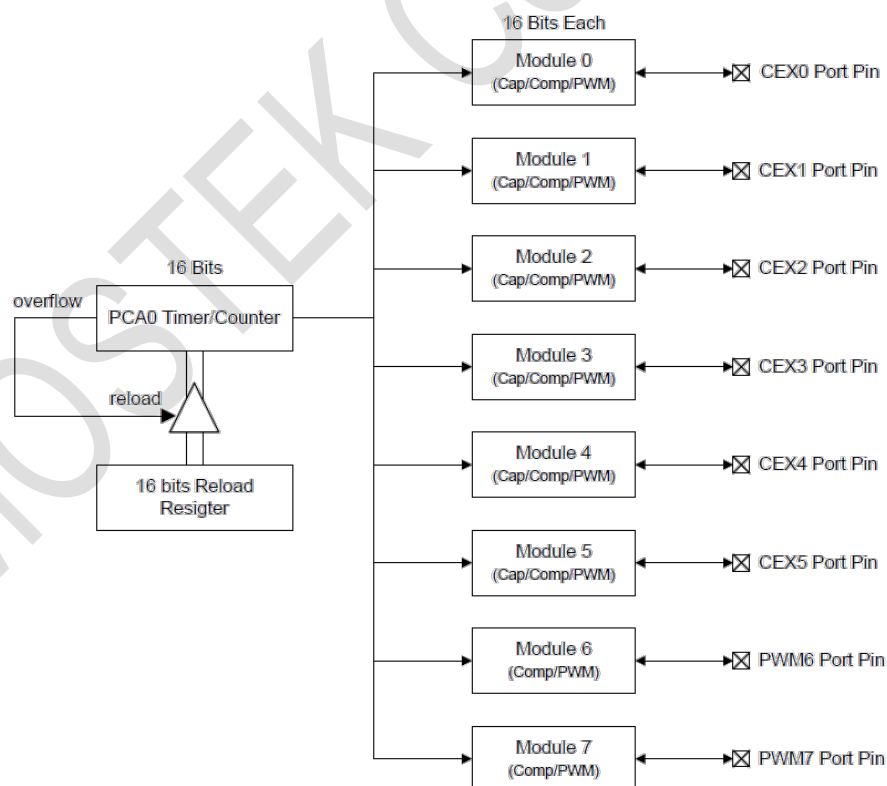


Figure 17-1. PCA Block Diagram

17.2 PCA Timer/Counter

The timer/counter for the PCA is a auto-reload 16-bit timer consisting of registers CH and CL (the high

and low bytes of the count values), CHRL, CLRL (the high and low bytes reload registers), as shown in Figure 17–2. CHRL and CLRL are reloaded to CH and CL at each time overflow on {CH+CL} counter which can change the PCA cycle time for variable PWM resolution, such as 7-bit or 9-bit PWM.

{CH + CL} is the common time base for all modules and its clock input can be selected from the following source:

- 1/12 the system clock frequency,
- 1/2 the system clock frequency,
- The Timer 0 overflow, which allows for a range of slower clock inputs to the timer,
- External clock input, 1-to-0 transitions, on ECI pin,
- CKMIX16, refer Section “9.1 Clock Structure”,
- Directly from the system clock frequency,
- The S0BRG overflow, S0TOF,
- MCKDO, refer Section “9.1 Clock Structure”.

Special Function Register CMOD contains the Count Pulse Select bits (CPS2, CPS1 and CPS0) to specify the PCA timer input. When CPS[2:0] indexes CKMIX16 or MCKDO, the frequency of the input clock sources must $\geq \text{CPUCLK} \times 3$ and $\geq \text{SYSCLK} \times 2$. This register also contains the ECF bit which enables an interrupt when the counter {CH+CL} overflows. And the counter overflow toggles C0COR, it will output on port pin when C0CKOE is enabled. In addition, the user has the option of turning off the PCA timer during Idle Mode by setting the Counter Idle bit (CIDL). This can further reduce power consumption during Idle mode.

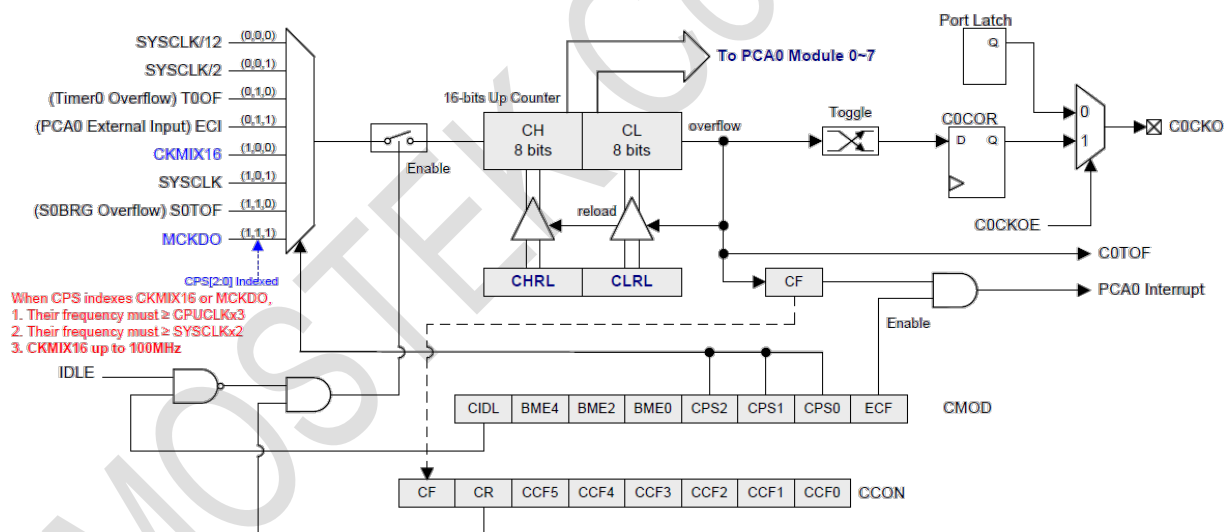


Figure 17-2. PCA Timer/Counter

- CMOD: PCA Counter Mode Register
SFR Page = 0 Only
SFR Address = 0xD9

Bit	7	6	5	4	3	2	1	0
Name	CIDL	BME4	BME2	BME0	CPS2	CPS1	CPS0	ECF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: CIDL, PCA counter Idle control.

0: Lets the PCA counter continue functioning during Idle mode. 1: Lets the PCA counter be gated off during Idle mode.

Bit 6: BME4, Buffer Mode Enable on PCA module 4/5. It is only valid on both of PCA module 4 and module 5 in capture mode, PWM mode or COPM mode.

0: PCA Module 4/5 buffer mode disabled. 1: PCA Module 4/5 buffer mode enabled.

Bit 5: BME2, Buffer Mode Enable on PCA module 2/3. It is only valid on both of PCA module 2 and module 3 in capture mode, PWM mode or COPM mode.

0: PCA Module 2/3 buffer mode disabled. 1: PCA Module 2/3 buffer mode enabled.

Bit 4: BME0, Buffer Mode Enable on PCA module 0/1. It is only valid on both of PCA module 0 and module 1 in capture mode, PWM mode or COPM mode.

0: PCA Module 0/1 buffer mode disabled. 1: PCA Module 0/1 buffer mode enabled.

Bit 3~1: CPS2-CPS0, PCA counter clock source select bits.

CPS2	CPS1	CPS0	PCA Clock Source
0	0	0	Internalclock,(system clock)/12
0	0	1	Internalclock,(system clock)/2
0	1	0	Timer0overflow
0	1	1	ExternalclockattheECIpin
1	0	0	CKMIX16output
1	0	1	Internalclock,(system clock)/1
1	1	0	S0BRToverflow
1	1	1	MCKDividerOutput,MCKDO

Note: When CPS indexes CKMIX16 or MCKDO, needs to follow the conditions:

1. The source frequency must $\geq CPUCLK \times 3$.
2. The source frequency must $\geq SYSCLK \times 2$.

Bit 0: ECF, Enable PCA counter overflow interrupt.

0: Disables an interrupt when CF bit (in CCON register) is set. 1: Enables an interrupt when CF bit (in CCON register) is set.

The CCON register shown below contains the run control bit for the PCA and the flags for the PCA timer and each module. To run the PCA the CR bit (CCON.6) must be set by software. The PCA is shut off by clearing this bit. The CF bit (CCON.7) is set when the PCA counter overflows and an interrupt will be generated if the ECF bit in the CMOD register is set. The CF bit can only be cleared by software. CCF0 to CCF5 are the interrupt flags for module 0 to module 5, respectively, and they are set by hardware when either a match or a capture occurs. These flags also can only be cleared by software. The PCA interrupt system is shown Figure 17–3.

- CCON: PCA Counter Control Register

SFR Page = 0 only

SFR Address = 0xD8

Bit	7	6	5	4	3	2	1	0
Name	CF	CR	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: CF, PCA Counter Overflow flag.

0: Only be cleared by software.

1: Set by hardware when the counter rolls over. CF flag can generate an interrupt if bit ECF in CMOD is set. CF may be set by either hardware or software.

Bit 6: CR, PCA Counter Run control bit.

0: Must be cleared by software to turn the PCA counter off.

1: Set by software to turn the PCA counter on.

Bit 5: CCF5, PCA Module 5 interrupt flag.

0: Must be cleared by software.

1: Set by hardware when a match or capture occurs.

Bit 4: CCF4, PCA Module 4 interrupt flag.

0: Must be cleared by software.

1: Set by hardware when a match or capture occurs.

Bit 3: CCF3, PCA Module 3 interrupt flag.

0: Must be cleared by software.

1: Set by hardware when a match or capture occurs.

Bit 2: CCF2, PCA Module 2 interrupt flag.

0: Must be cleared by software.

1: Set by hardware when a match or capture occurs.

Bit 1: CCF1, PCA Module 1 interrupt flag.

0: Must be cleared by software.

1: Set by hardware when a match or capture occurs.

Bit 0: CCF0, PCA Module 0 interrupt flag.

0: Must be cleared by software.

1: Set by hardware when a match or capture occurs.

Figure 17-3. PCA Interrupt System

- **PCAPWMn**: PWM Mode Auxiliary Register, n=0~7

SFR Page = 0 only for n= 0~1 (n=2~5 for all page) SFR Page = 1 only for n= 6~7

SFR Address = 0xF2~0xF7

Bit	7	6	5	4	3	2	1	0
Name	PnRS1	PnRS0	--	--	CCFn	PnINV	ECAPnH	ECAPnL
R/W	R/W	R/W	W	W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 3: CCFn, only CCF6 and CCF7 are valid for the interrupt flag in module 6 and module 7. 0: Must be cleared by software.

1: Set by hardware when a match occurs.

- **CH**: PCA base timer High

SFR Page = 0 ~ F

SFR Address = 0xF9

Bit	7	6	5	4	3	2	1	0
Name	CH[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

- **CL**: PCA base timer Low

SFR Page = 0 ~ F

SFR Address = 0xE9

Bit	7	6	5	4	3	2	1	0
Name	CL[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

- **CHRL**: PCA CH Reload Register

SFR Page = 0 ~ F

SFR Address = 0xCF

Bit	7	6	5	4	3	2	1	0
Name	CHRL[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~0: CHRL, reload value of CH.

- **CLRL:** PCA CL Reload Register

SFR Page = 0 ~ F

SFR Address = 0xCE

Bit	7	6	5	4	3	2	1	0
Name	CLRL[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~0: CLRL, reload value of CL.

17.3 Compare/Capture Modules

Each of the compare/capture module 0~7 has a mode register called CCAPMn (n = 0,1,2,3,4,5,6 or 7) to select which function it will perform. Note the ECCFn bit which enables an interrupt to occur when a module's interrupt flag is set.

- **CCAPMn:** PCA Module Compare/Capture Register, n=0~5

SFR Page = 0 only for n= 0~1 (n=2~5 for all page)

SFR Address = 0xDA~0xDF

Bit	7	6	5	4	3	2	1	0
Name	DTE _n	ECOM _n	CAPP _n	CAPN _n	MAT _n	TOG _n	PWM _n	ECCF _n
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: DTE_n. Enable Dead-Time control on PWMH_n/PWML_n output pair. This bit is only valid on n= 0, 2 and 4 and the dead-time function is active when PWM channel is operating in buffer mode. The channel buffer mode is enabled by BME0, BME2 or BME4 in CMOD.

0: Disable the Dead-Time control on PWM_n output. 1: Enable the Dead-Time control on PWM_n output.

Bit 6: ECOM_n, Enable Comparator.

0: Disable the digital comparator function. 1: Enables the digital comparator function.

Bit 5: CAPP_n, Capture Positive enabled. Module 6 and module 7 don't support the capture mode. 0: Disable the PCA capture function on CEX_n positive edge detected.

1: Enable the PCA capture function on CEX_n positive edge detected.

Bit 4: CAPN_n, Capture Negative enabled. Module 6 and module 7 don't support the capture mode. 0: Disable the PCA capture function on CEX_n negative edge detected.

1: Enable the PCA capture function on CEX_n negative edge detected.

Bit 3: MAT_n, Match control.

0: Disable the digital comparator match event to set CCF_n.

1: A match of the PCA counter with this module's compare/capture register causes the CCF_n bit in CCON to be set.

Bit 2: TOG_n, Toggle control.

0: Disable the digital comparator match event to toggle CEX_n.

1: A match of the PCA counter with this module's compare/capture register causes the CEX_n pin to toggle.

Bit 1: PWM_n, PWM control.

0: Disable the PWM mode in PCA module.

1: Enable the PWM function and cause CEX_n pin to be used as a pulse width modulated output.

Bit 0: ECCF_n, Enable CCF_n interrupt.

0: Disable compare/capture flag CCF_n in the CCON register to generate an interrupt. 1: Enable

compare/capture flag CCFn in the CCON register to generate an interrupt.

Note: The bits CAPNn (CCAPMn.4) and CAPPn (CCAPMn.5) determine the edge on which a capture input will be active. If both bits are set, both edges will be enabled and a capture will occur for either transition.

Each module also has a pair of 8-bit compare/capture registers (CCAPnH, CCAPnL) associated with it. These registers are used to store the time when a capture event occurred or when a compare event should occur. When a module is used in the PWM mode, in addition to the above two registers, an extended register PCAPWMn is used to improve the range of the duty cycle of the output. The improved range of the duty cycle starts from 0%, up to 100%, with a step of 1/256. About 10/12/16 bit PWM please reference 17.4.6 and 17.4.7.

● **CCAPMn:** PCA Module Compare/Capture Register, n=6~7

SFR Page = 1 only for n= 6~7

SFR Address = 0xDB, 0xDA

Bit	7	6	5	4	3	2	1	0
Name	DTEn	ECOMn	--	CAPNn	MATn	TOGn	PWMn	ECCFn
R/W	R/W	R/W	W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit7:BME6(This is only valid in CCAPM6) Buffer Mode Enable on PCA module 6/7. It is only valid on both of PCA module 6 and module 7 in capture mode, PWM mode or COPM mode.
0: PCA module 6/7 buffer mode is disabled. 1: PCA module 6/7 buffer mode is enabled.

Bit6: ECOMn, Enable Comparator.
0: Disable the digital comparator function. 1: Enable the digital comparator function.

Bit5: Reserved. Module 6 and module 7 don't support the capture mode.

Bit4: CAPNn, Capture Negative enabled. Module 6 and module 7 don't support the capture mode. CAPN6 and CAPN7 is used for other PCA mode setting, please reference "Table 17-1. PCA Module Modes" for details.

Bit3: MATn, Match control.
0: Disable the digital comparator match event to set CCFn.
1: A match of the PCA counter with this module's compare/capture register causes the CCFn bit in PCAPWMn to be set.

Bit2: TOGn, Toggle control.
0: Disable the digital comparator match event to toggle CEXn.
1: A match of the PCA counter with this module's compare/capture register causes the CEXn pin to toggle.

Bit1: PWMn, PWM control.
0: Disable the PWM mode in PCA module.
1: Enable the PWM function and cause CEXn pin to be used as a pulse width modulated output.

Bit0: ECCFn, Enable CCFn interrupt.
0: Disable compare/capture flag CCFn in the PCAPWMn register to generate an interrupt. 1: Enable compare/capture flag CCFn in the PCAPWMn register to generate an interrupt.

Each module also has a pair of 8-bit compare/capture registers (CCAPnH, CCAPnL) associated with it. These registers are used to store the time when a compare event occurs.

When a module is used in the PWM mode, in addition to the above two registers, an extended register PCAPWMn is used to improve the range of the duty cycle of the output. The improved range of the duty cycle starts from 0%, up to 100%, with a step of 1/256. About 10/12/16-bit PWM please reference 17.4.6 and 17.4.7.

- CCAPnH: PCA Module n Capture High Register, n=0~7

SFR Page = 0 only for n= 0~1 (n=2~5 for all page) SFR Page = 1 only for n= 6~7

SFR Address = 0xFA~0xFF

Bit	7	6	5	4	3	2	1	0
Name	CCAPnH[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

- CCAPnL: PCA Module n Capture Low Register, n=0~7 SFR Page = 0 only for n= 0~1 (n=2~5 for all page)
SFR Page = 1 only for n= 6~7
SFR Address = 0xEA~0xEF

Bit	7	6	5	4	3	2	1	0
Name	CCAPnL[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

- PCAPWMn: PWM Mode Auxiliary Register, n=0~7
SFR Page = 0 only for n= 0~1 (n=2~5 for all page)
SFR Page = 1 only for n= 6~7
SFR Address = 0xF2~0xF7

Bit	7	6	5	4	3	2	1	0
Name	PnRS1	PnRS0	--	--	CCFn	PnINV	ECAPnH	ECAPnL
R/W	R/W	R/W	W	W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	X	0	0	0

Bit 7~6: PnRS1~0, PWMn Resolution Setting 1~0.

00: 8 bit PWMn, the overflow is active when [CH, CL] counts XXXX-XXXX-1111-1111 □
XXXX-XXXX-0000-0000.

01: 10 bit PWMn, the overflow is active when [CH, CL] counts XXXX-XX11-1111-1111 □
XXXX-XX00-0000-0000.

10: 12 bit PWMn, the overflow is active when [CH, CL] counts XXXX-1111-1111-1111 □
XXXX-0000-0000-0000.

11: 16 bit PWMn, the overflow is active when [CH, CL] counts 1111-1111-1111-1111 □
0000-0000-0000-0000.

Bit 5~4: Reserved. Software must write "0" on these bits when PCAPWMn is written.

Bit 3: CCFn, only CCF6 and CCF7 are valid for the interrupt flag in module 6 and module 7. 0: Must be cleared by software.

1: Set by hardware when a match occurs.

Bit 2: PnINV, Invert Compare/PWM output (C0PnOR) on CEXn pin. 0: Non-inverted Compare/PWM output (C0PnOR).

1: Inverted Compare/PWM output (C0PnOR).

Bit 1: ECAPnH, Extended 9th bit (MSB bit), associated with CCAPnH to become a 9-bit register used in PWM mode.

Bit 0: ECAPnL, Extended 9th bit (MSB bit), associated with CCAPnL to become a 9-bit register used in PWM mode.

17.4 Operation Modes of the PCA

Table 17-1 shows the CCAPMn register settings for the various PCA functions.

Table 17-1. PCA Module Modes

ECOMn	CAPPn	CAPn	MATn	TOGn	PWMn	ECCFn	模块功能
0	0	0	0	0	0	0	Nooperation
X	1	0	0	0	0	X	16-bit capture by a positive-edge trigger on CEXn
X	0	1	0	0	0	X	16-bit capture by a negative-edge trigger on CEXn

X	1	1	0	0	0	X	16-bit capture by a transition on CEXn
1	0	0	1	0	0	X	16-bit Software Timer (Compare)
1	0	0	1	1	0	X	16-bit High Speed Output (HSO)
1	0	0	0	0	1	X	Pulse Width Modulator (PWM)
1	0	0	0	1	1	X	Compare Output on PWM match case (COPM)
1	0	1	0	0	1	X	FIFO Data Mode

Note: PCA Module 6 and module 7 don't support the capture mode.

17.4.1 Capture Mode

To use one of the PCA modules in the capture mode, either one or both of the bits CAPN and CAPP for that module must be set. The external CEX input for the module is sampled for a transition. When a valid transition occurs the PCA hardware loads the value of the PCA counter registers (CH and CL) into the module's capture registers (CCAPnL and CCAPnH). If the CCFn and the ECCFn bits for the module are both set, an interrupt will be generated.

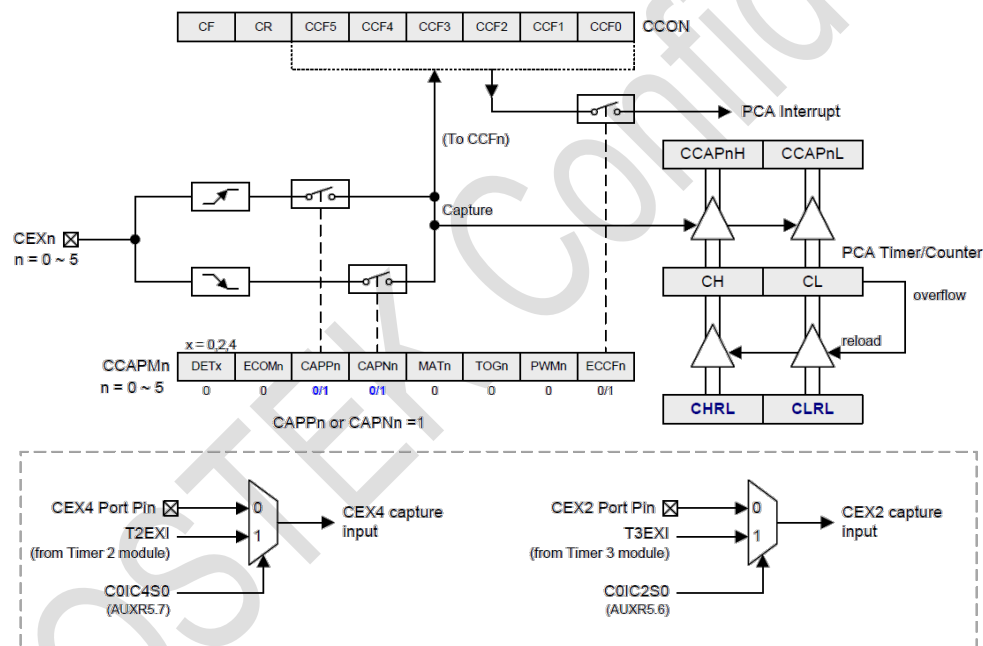


Figure 17-4. PCA Capture Mode

17.4.2 Buffered Capture Mode

To capture narrow input signal, buffered capture mode is necessary. If enabled, it put the odd module capture data registers (CCAPnH, CCAPnL, n= 1, 3, 5) to be the buffer register of even module capture data registers (channel 0, 2, 4). There is no influence on module 0/2/4 capture operation. BME0 enables the buffer operation of channel 0 and channel 1. BME2 and BME4 control the module 2/3 and module 4/5.

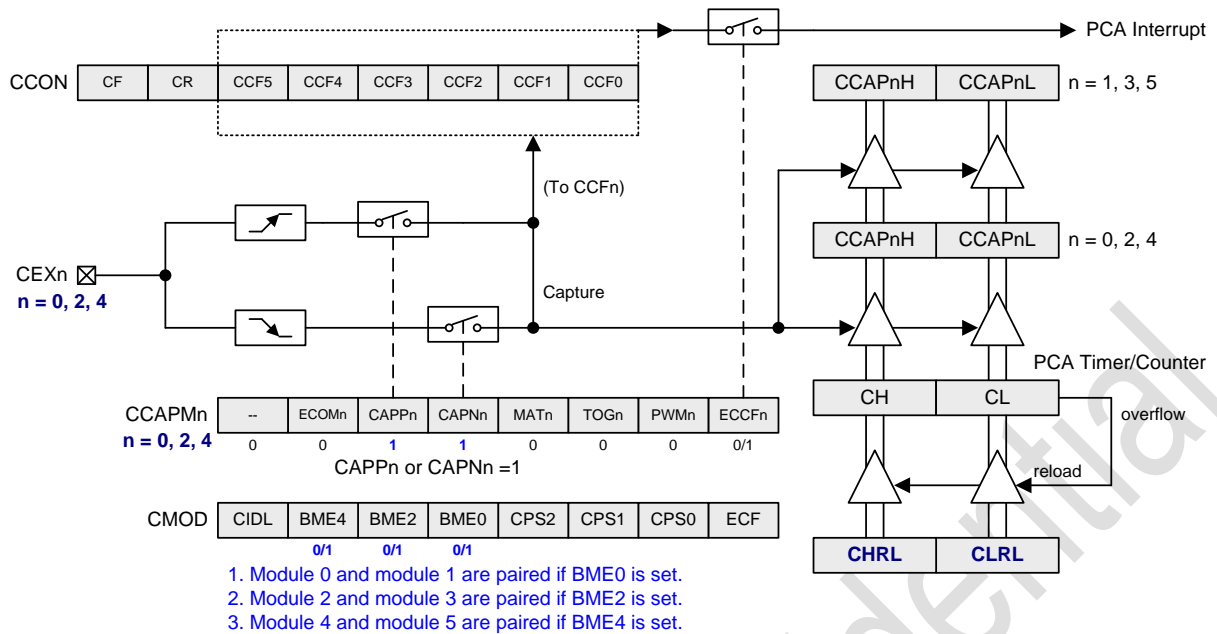


Figure 17-5. PCA Buffered Capture Mode (BMEn=1, n= 0, 2, 4)

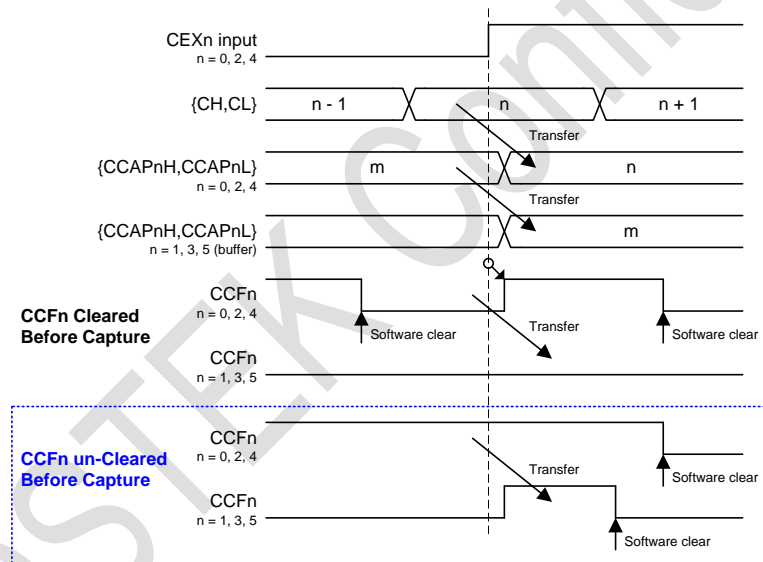


Figure 17-6. PCABufferedCaptureModeWaveform

17.4.3 16-bit Software Timer Mode (Compare mode)

The PCA modules can be used as software timers by setting both the ECOM and MAT bits in the module's CCAPMn register. The PCA timer will be compared to the module's capture registers, and when a match occurs an interrupt will occur if the CCFn and the ECCFn bits for the module are both set.

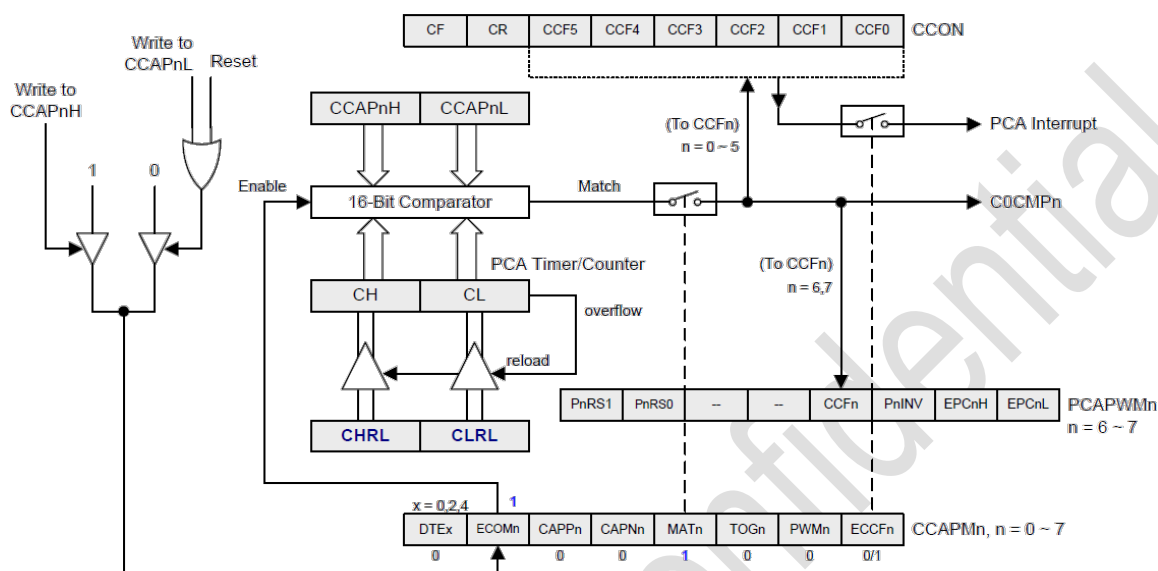


Figure 17-7. PCA Software Timer Mode

17.4.4 High Speed Output Mode (Compare Output mode)

In this mode the CEX output associated with the PCA module will toggle each time a match occurs between the PCA counter and the module's capture registers. To activate this mode, the TOG, MAT and ECOM bits in the module's CCAPMn register must be set.

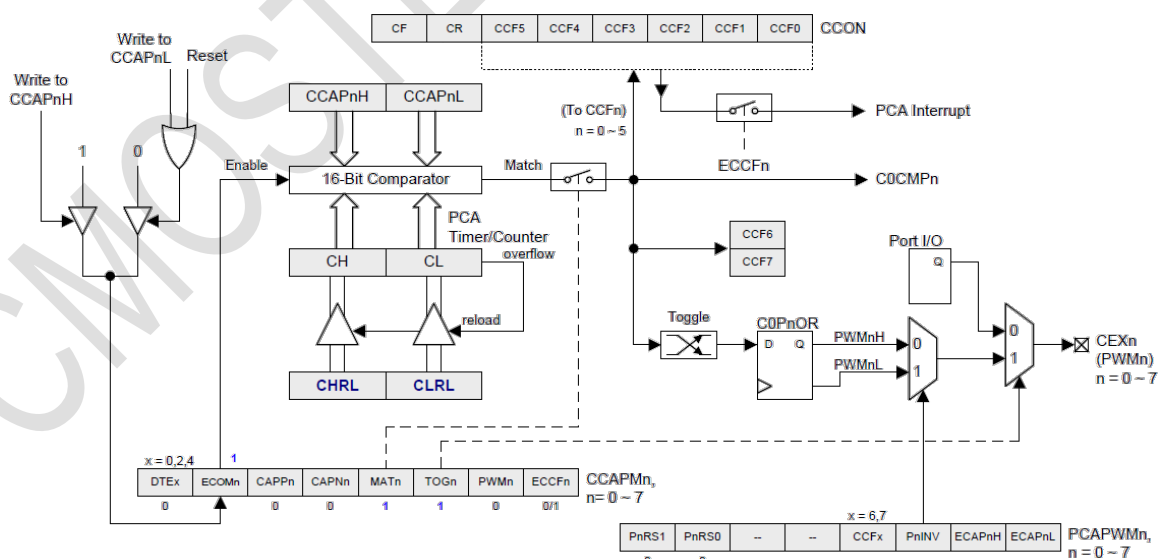


Figure 17-8. PCA High Speed Output Mode

17.4.5 Buffered 8-bit PWM Mode

All of the PCA modules can be used as PWM outputs. The frequency of the output depends on the clock source for the PCA timer. All of the modules will have the same frequency of output because they all share the PCA timer.

The duty cycle of each module is determined by the module's capture register CCAPnL and the extended 9th bit, ECAPnL. When the 9-bit value of { 0, [CL] } is *less than* the 9-bit value of { ECAPnL, [CCAPnL] } the output will be low, and if *equal to or greater than* the output will be high.

When CL overflows from 0xFF to 0x00, { ECAPnL, [CCAPnL] } is reloaded with the value of { ECAPnH, [CCAPnH] }. This allows updating the PWM without glitches. The PWMn and ECOMn bits in the module's CCAPMn register must be set to enable the PWM mode.

Using the 9-bit comparison, the duty cycle of the output can be improved to really start from 0%, and up to 100%. The formula for the duty cycle is:

$$\text{DutyCycle} = 1 - \{\text{ECAPnH}, [\text{CCAPnH}]\} / 256.$$

Where, $[CCAPnH]$ is the 8-bit value of the CCAPnH register, and $ECAPnH$ (bit-1 in the PCAPWMn register) is 1-bitvalue. So, $\{ECAPnH, [CCAPnH]\}$ forms a 9-bitvalue for the 9-bit comparator.

Forexamples,

- If $ECAPnH=0$ & $CCAPnH=0x00$ (i.e., $0x000$), the duty cycle is 100%.
- If $ECAPnH=0$ & $CCAPnH=0x40$ (i.e., $0x040$) the duty cycle is 75%.
- If $ECAPnH=0$ & $CCAPnH=0xC0$ (i.e., $0x0C0$), the duty cycle is 25%.
- If $ECAPnH=1$ & $CCAPnH=0x00$ (i.e., $0x100$), the duty cycle is 0%.

Buffered 8-bit PWM: PnRS[1:0] = 00

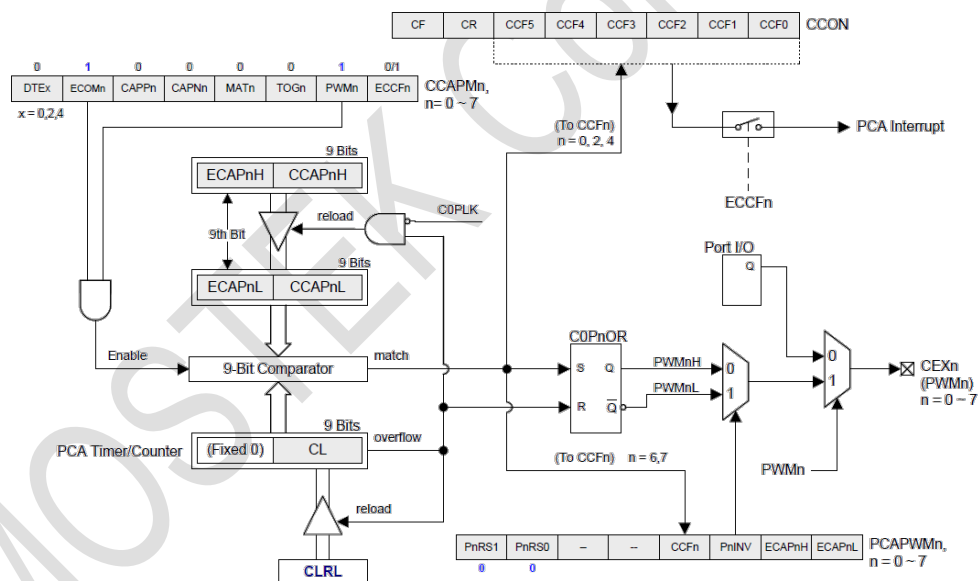


Figure17-9. PCA Buffered 8-bit PWM Mode

17.4.6 Un-buffered 10/12/16-bit PWM Mode

The PCA provides the variable PWM mode to enhance the control capability on PWM application. There are additional un-buffered 10/12/16 bits PWM can be assigned in each channel and each PWM channel with different resolution can operate concurrently.

Un-buffered 10/12/16-bit PWM: PnRS[1:0] = 01/10/11

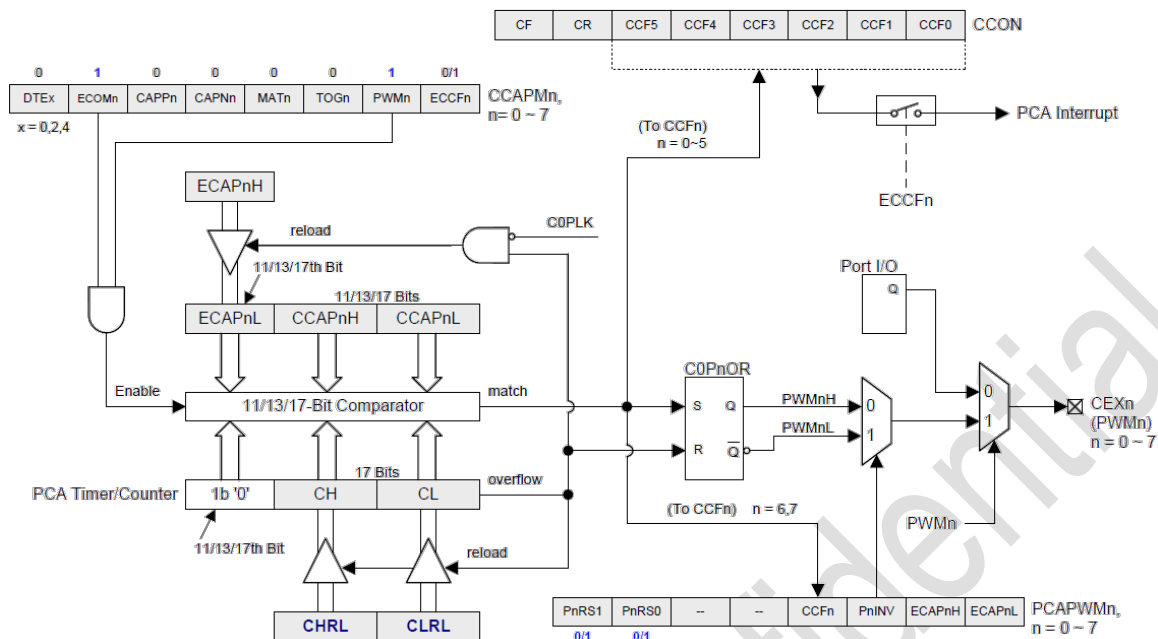
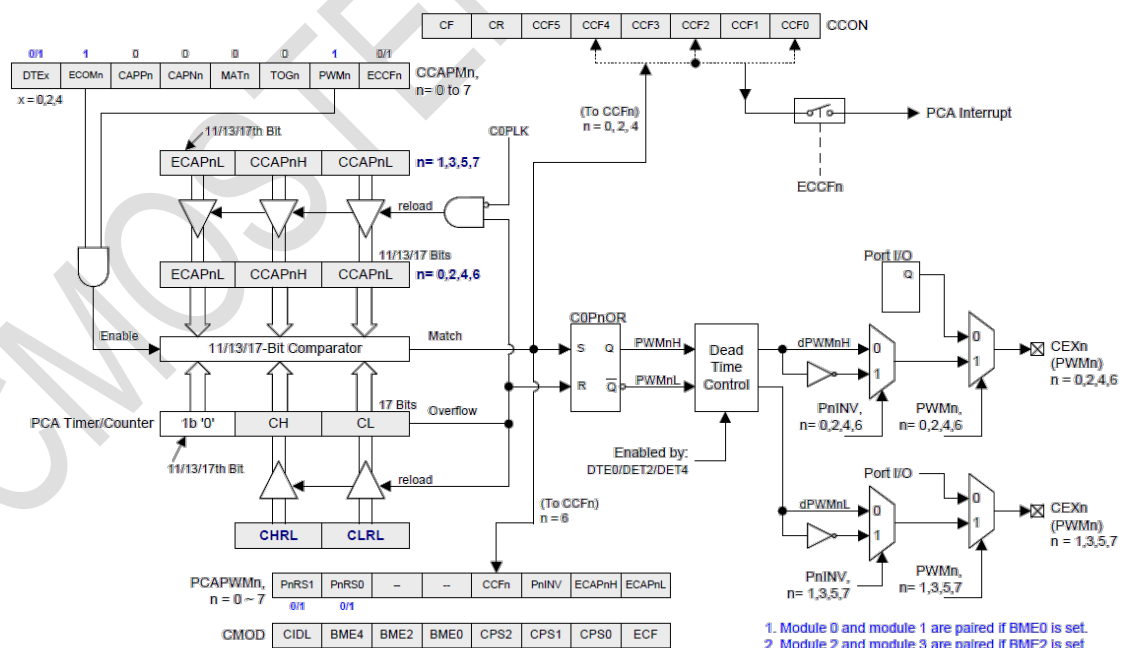


Figure 17-10. PCA Un-buffered 10/12/16-bit PWM Mode

17.4.7 Buffered 10/12/16-bit PWM Mode

To use 10/12/16-bit PWM mode might cause unexpected duty cycle when change the duty cycle setting by writing data into CCAPnH and CCAPnL, because the 8 bit CPU can only write one byte at a time. To finish fully setting it will take two write cycles, and the comparator will output unexpected duty cycle when the first byte have been written. If the applications need accurate control when change the duty cycle, it needs to use the Buffered PWM mode.

Buffered 9~16-bit PWM: BM0/BM2/BM4 = 1, PnRS[1:0] = 01/10/11



1. Module 0 and module 1 are paired if BME0 is set.
2. Module 2 and module 3 are paired if BME2 is set.
3. Module 4 and module 5 are paired if BME4 is set.
4. Module 6 and module 7 are paired if BME6 is set. (BME6 is located at CCAPM6.7)
5. Only pair 0/1, 2/3, 4/5 support dead-time control, central-aligned & break control.

Figure 17-11. PCA Buffered 10/12/16-bit PWM Mode (with dead time control)

17.4.8 COPM Mode

Compare Output on PWM Match mode is similar to High Speed Output Mode, but it uses PCA0 PWM comparators instead of fixed 16-bit comparators. It gives more flexibility to the applications. For example, if it uses 8-Bit PWM for the PCA0 comparator, the output toggles frequency can higher than High Speed Output Mode.

Un-buffered COPM: PnRS[1:0] = 00/01/10/11

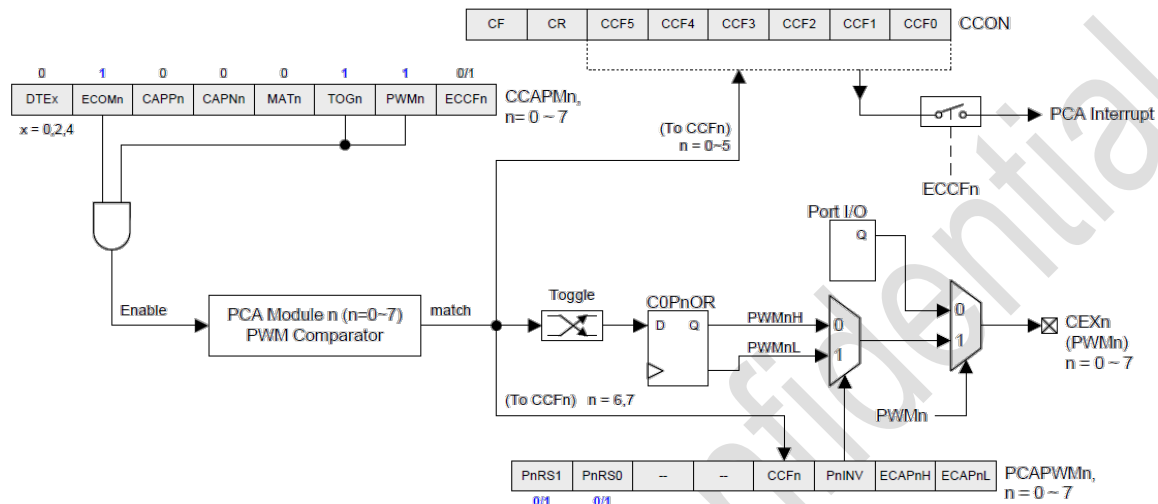


Figure 17-12. PCA COPM Mode

17.4.9 Buffered COPM Mode

If the applications need to have any phase control of the PWM signals, it needs to set the PCA0 modules in buffered COPM mode. One pair of the PCA0 module (n=0&1 / 2&3 / 4&5) can program the time delay of the two edges of one cycle of the PWM signal. It means you can set the start and end point of the waveform. This is useful when the 2 or 3 correlation PWM signals can set the phase shift between each other.

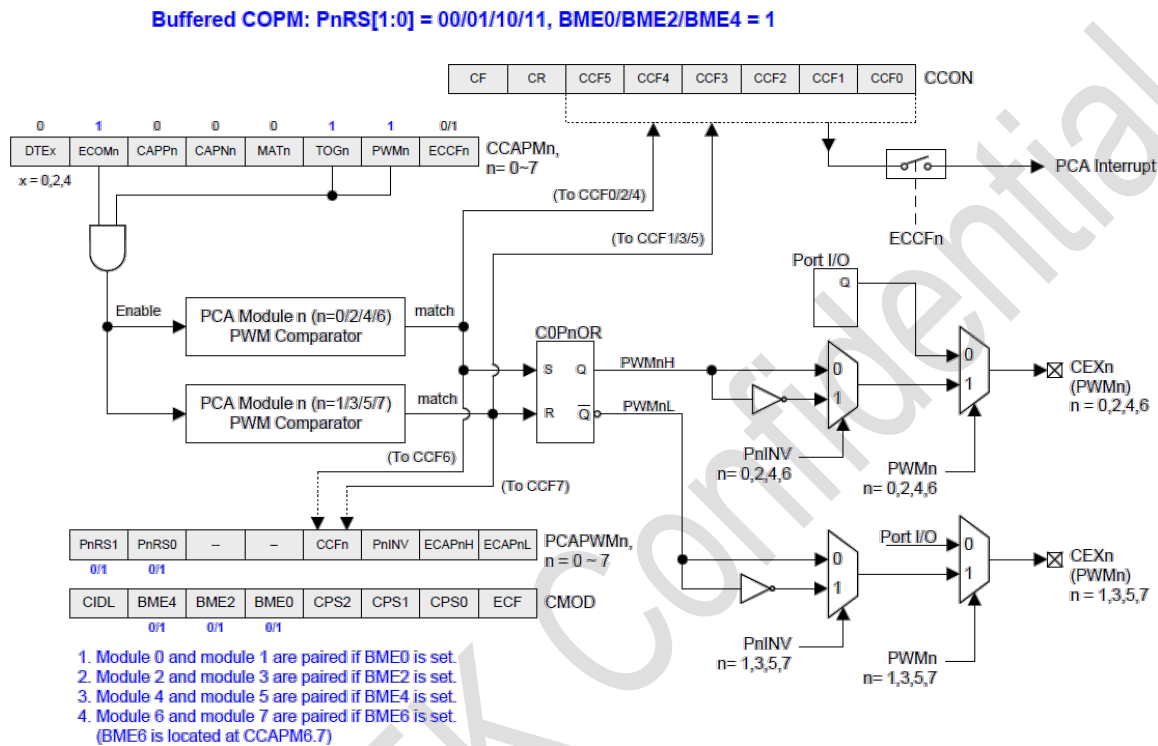


Figure 17-13. PCA Buffered COPM Mode

17.4.10 FIFO Data Mode

In this mode the user can set the CCAPnL, CCAPnH, CCAP(n+1)L and CCAP(n+1)H as a buffer chain. After all these buffers are set, it can change the duty sequentially trigger by T0OF, T1OF, T3OF or S0TOF. This function is enabled, the CPU can leave it to run by itself to earn more time slot to do other operations. For example, when the power converter start to raise the voltage from light load to heave load, it might useful to set the duty larger than the target it the beginning period, and then reduce the duty step by step close to the target duty. It can just set all duties in the buffer and leave it to finish.

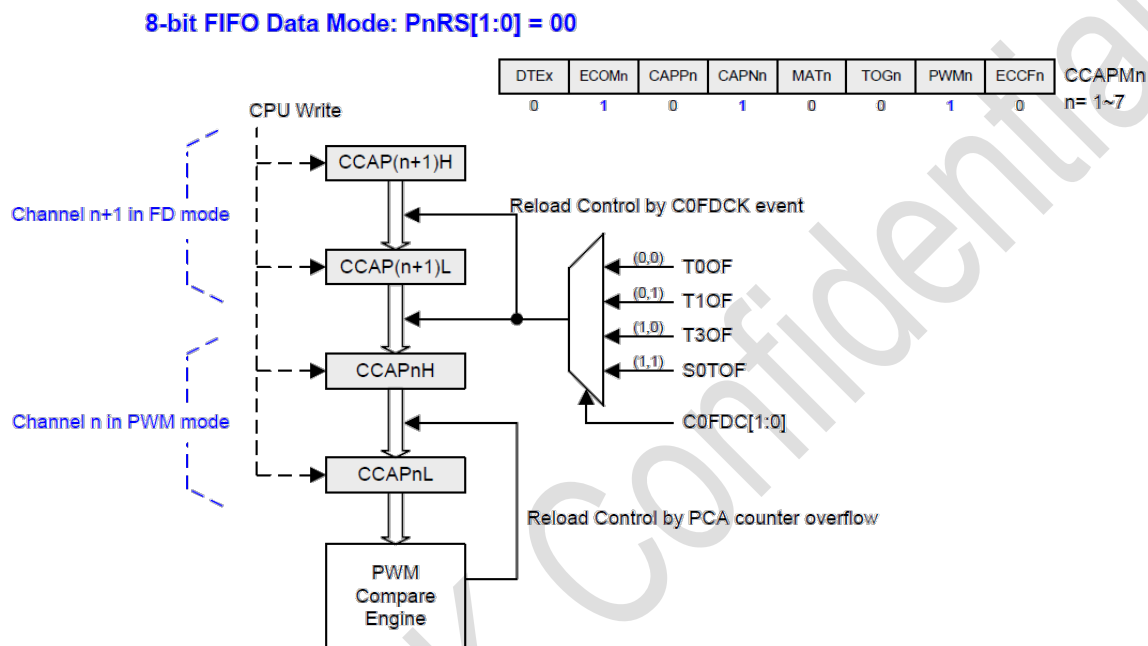


Figure 17-14. PCA channel for FIFO Data Mode

Channel FIFO data mode that is moved on C0FDCK.

C0FDCK source selection, updated clock selection of PCA0 FIFO Data mode.

C0FDC1~0	C0FDCK
00	T0OF
01	T1OF
10	T3OF
11	S0TOF

- AUXR9: Auxiliary Register 9**

SFR Page = 6 only

SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	--	--	T1G0	T0G1	C0FDC1	C0FDC0	S1PS1	S1PS0
R/W	W	W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

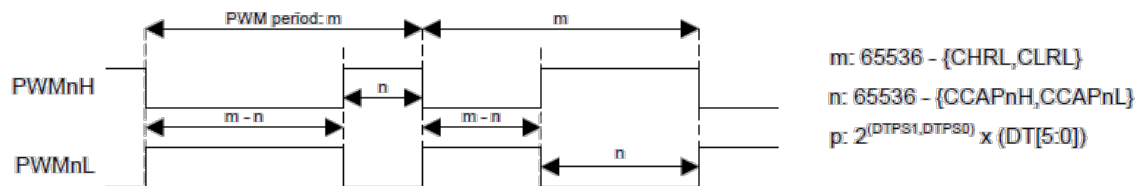
Bit 3~2: C0FDC1~0, C0FDCK Selection [1:0].

17.4.11 Enhanced PWM Control

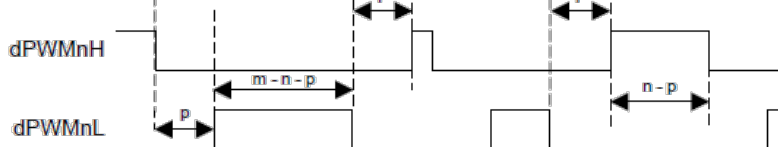
The PCA provides the variable PWM mode to enhance the control capability on PWM application. There are additional 10/12/16 bits PWM can be assigned in each channel and each PWM channel with different

resolution and different phase delay can operate concurrently.

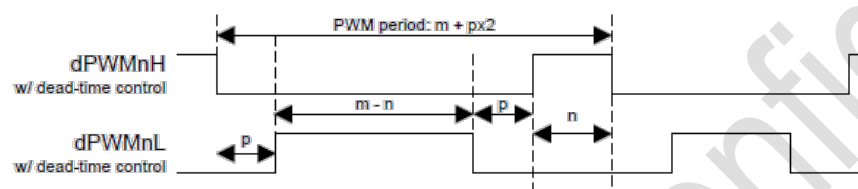
Normal 16-bit PWM



PWM with dead-time control



PWM with dead-time control & EXDT enabled & Edge Aligned



PWM with dead-time control & EXDT enabled & Central Aligned

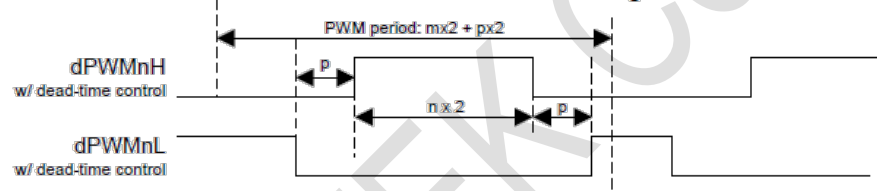


Figure 17-15. PWM Waveform with Dead-Time Control

- CCAPMn: PCA Module Compare/Capture Register, n=0~5
 SFR Page = 0 only for n= 0~1 (n=2~5 for all page)
 SFR Address = 0xDA~0xDF

Bit	7	6	5	4	3	2	1	0
Name	DTE _n	ECOM _n	CAPP _n	CAPN _n	MAT _n	TOG _n	PWM _n	ECCF _n
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: DTE_n. Enable Dead-Time control on PWMH_n/PWML_n output pair. This bit is only valid on n= 0, 2 and 4 and the dead-time function is active when PWM channel is operating in buffer mode. The channel buffer mode is enabled by BME0, BME2 or BME4 in CMOD.

0: Disable the Dead-Time control on PWM_n output.

1: Enable the Dead-Time control on PWM_n output.

- PDTCRA: PWM Dead-Time Control Register -A
 SFR Page = 1 only
 SFR Address = 0xBC

Bit	7	6	5	4	3	2	1	0
Name	DTPS1	DTPS0	DT[5:0]					
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~6: DTPS1~0, Clock Pre-Scaler of Dead-Time counter.

DTPS[1:0]	Pre-Scaler Selection
00	SYSCLK
01	SYSCLK/2
10	SYSCLK/4
11	SYSCLK/8

Bit 5~0: DT5~0, Dead-Time period control bits.

DT[5:0]	Dead-Time Period
000000	Dead-TimeDisabled
000001	Pre-ScalerClockX1
000010	Pre-ScalerClockX2
000011	Pre-ScalerClockX3
.....
111110	Pre-ScalerClock X62
111111	Pre-ScalerClock X63

● PWMCR: PWM Control Register

SFR Page = 0 only

SFR Address = 0xBC

Bit	7	6	5	4	3	2	1	0
Name	PCAE	EXDT	PBKM	PBKE1.1	PBKE1.0	PBKE0.2	PBKE0.1	PBKE0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: PCAE, PWM Central Aligned Enabled. PCAE controls the enabled PWM channels to central aligned modulation including buffer mode PWM or non-buffer mode PWM. In this PWM mode, the PWM frequency is the half of edge aligned mode. This function is only active on PWMO0~5.

0: Set the PWM function with edge aligned modulation.

1: Enable the PWM function with central aligned modulation. It only supports 8/10/12/16-bit resolution on CHRL and CLRL setting.

Bit 6: EXDT: Extend Dead-Time in PWM Period. This function will corrupt the non-PWM channel function. Such as capture mode, software timer mode and high speed output mode.

0: Disable M + 2P.

1: Enable M + 2P on enabled PWM channel.

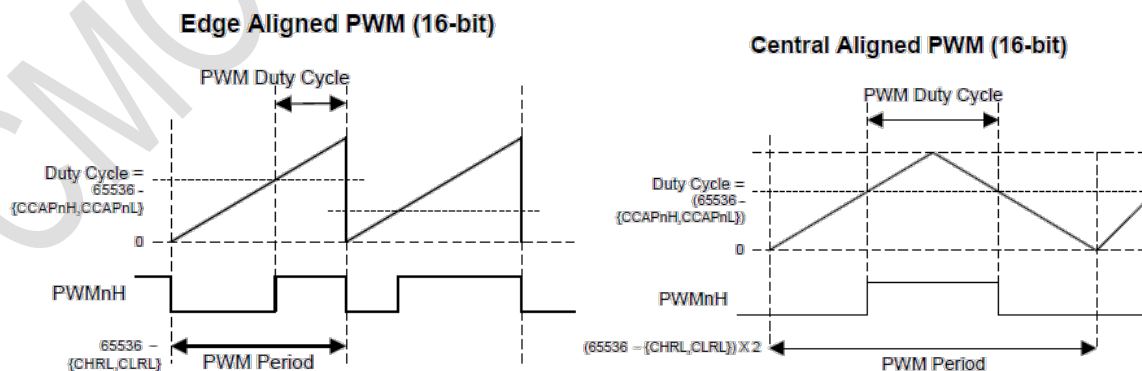


Figure 17-16. Waveform of Edge Aligned PWM and Central Aligned PWM

Bit 5: PBKM, PWM Break Mode selection.

0: Latched Mode.

1: Cycle-by-cycle Mode.

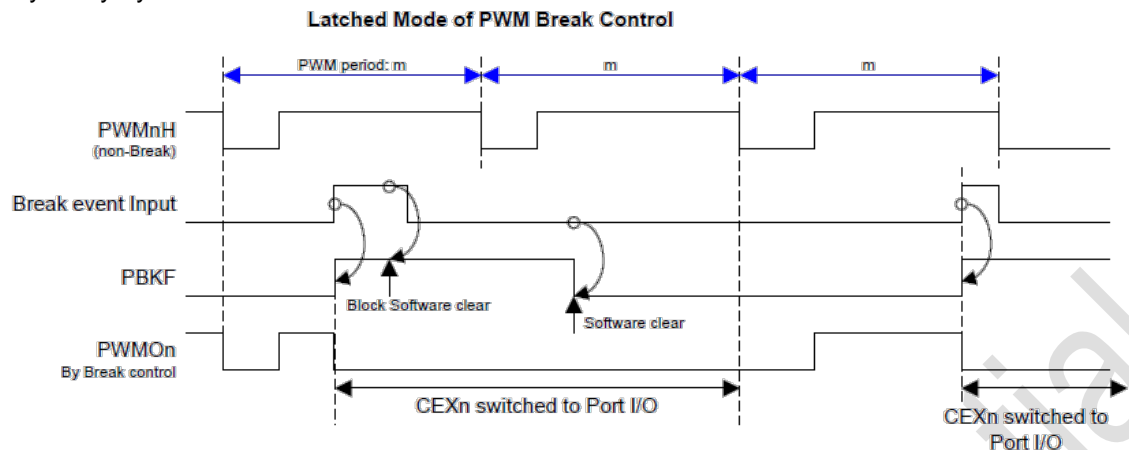


Figure 17-17. Latch Mode Waveform of PWM Break control

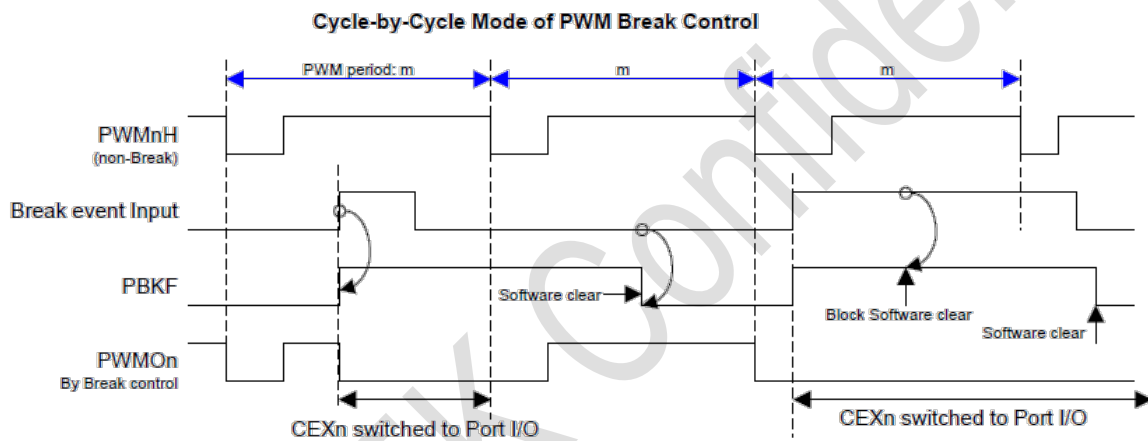


Figure 17-18. Cycle-by-Cycle Mode Waveform of PWM Break control

Bit 4~3: PBKE1.1~0, PWM Break Enable 1 selection. This function is only active on CEXn output mode (n=0~5).

PBKE1[1:0]	PWM Break Source
00	DisablePWMbreaksource1
01	INT2ET,nINT2active
10	Reserved
11	KBIET,KBIatchactive

Bit 2~0: PBKE0.2~0, PWM Break Enable 0 selection. This function is only active on CEXn output mode (n=0~5).

PBKE0[2:0]	PWM Break Source
000	DisablePWMbreaksource0
001	Reserved
010	INT1ET,nINT1active
011	T2EXI
100	KBIET,KBIatchactive
101	Reserved
110	Reserved
Other	Reserved.

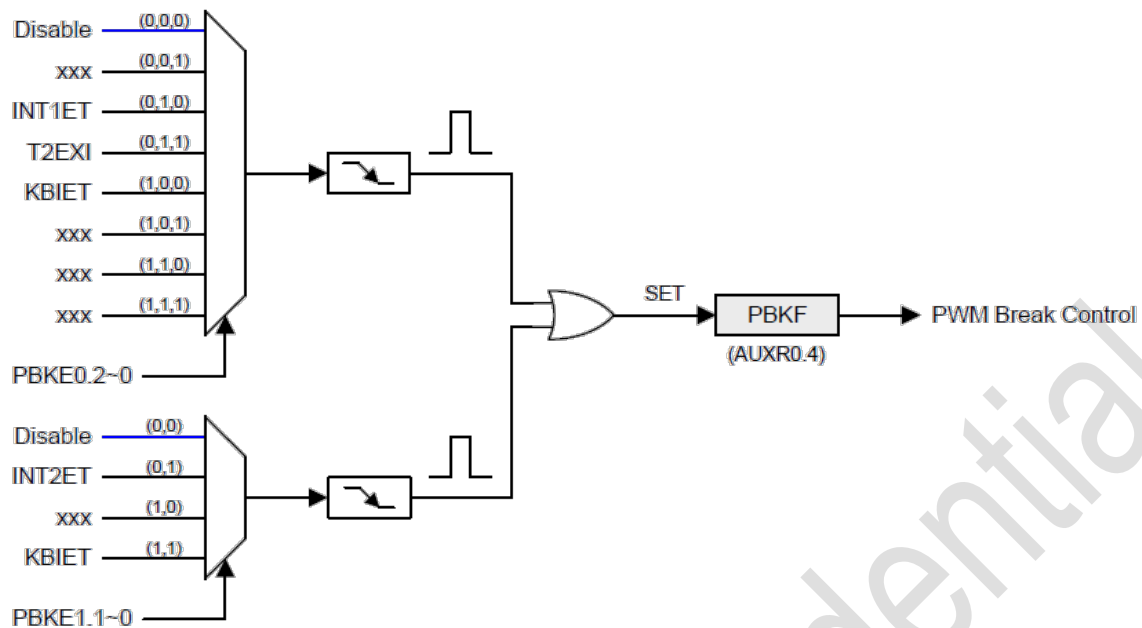


Figure 17-19. PCA PWM Break control source

- AUXR0: Auxiliary Register 0
SFR Page = 0~F
SFR Address = 0xA1

Bit	7	6	5	4	3	2	1	0
Name	P60OC1	P60OC0	P60FD	PBKF	--	--	INT1H	INT0H
R/W	R/W	R/W	R/W	R/W	W	W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 4: PBKF, PWM Break Flag. This bit is set by PWM break source enabled. If this flag is set, the enabled PWM channel 0~5 will be blocked and the output pins keep the original GPIO state.

0: There is no PWM Break event happened. It is only cleared by software.

1: There is a PWM Break event happened or software triggers a PWM Break.

17.4.12 PCA Module Output Control

PCA0 modules have multi output control mode can be selected for different applications. The CEX_n (n=1,3,4,5,6,7) can be programed as general I/O port or the output of the PCA0 module (PWM) 1, 3, 4, 5, 6 and 7. When PWM has been assigned to the CEX_n, the PnINV can switch between the normal PWM signal or inverted PWM signal. POEn can be used to enable or disable the PWM output to the port pin.

The CEX_n (n=1,3,4,5) also can use PBKF, PWM Break Flag, to break PWM output. If this flag is set, the enabled PWM channel 0~5 will be blocked and the output pins keep the original GPIO state.

In addition, PCA0 module 0 and 2 have 2 "Cloned" signals to the different port pin. These three the same PWM signals can be masked particular cycles by the POEnA or POEnB or PWM_n for the applications which need phase control.

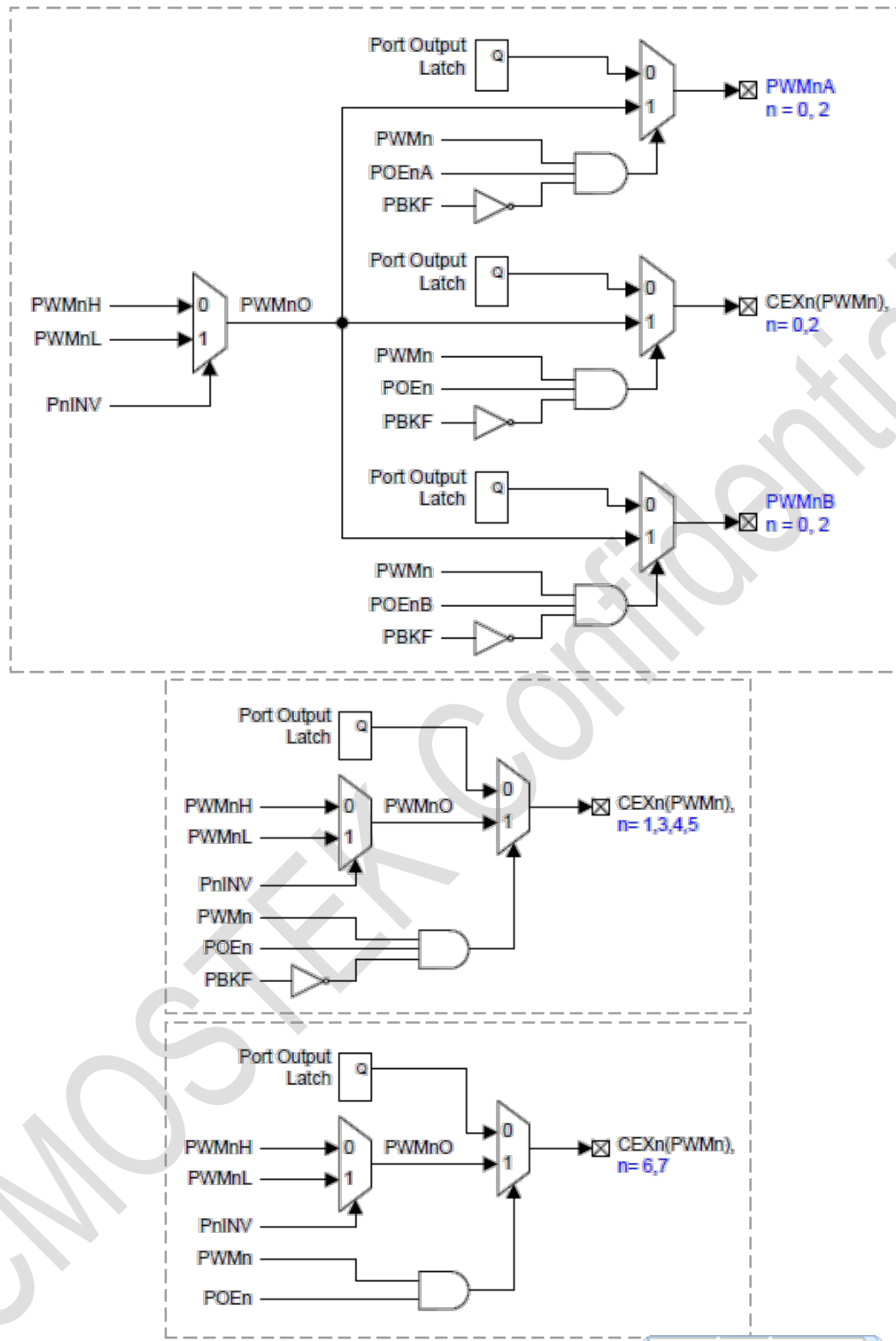


Figure 17-20. PCA Module output control

- **PAOE:** PWM Additional Output Enable Register
SFR Page = 0~F
SFR Address = 0xF1

Bit	7	6	5	4	3	2	1	0
Name	POE3	POE2B	POE2A	POE2	POE1	POE0B	POE0A	POE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reset Value	1	0	0	1	1	0	0	1
-------------	---	---	---	---	---	---	---	---

Bit 7: POE3, PCA0 PWM3 main channel (PWM3O) output control.

0: Disable PWM3O output on port pin.

1: Enable PWM3O output on port pin. Default is enabled.

Bit 6: POE2B, PCA0 PWM2 3rd channel (PWM2B) output control.

0: Disable PWM2B output on port pin. Default is disabled.

1: Enable PWM2B output on port pin.

Bit 5: POE2A, PCA0 PWM2 2nd channel (PWM2A) output control.

0: Disable PWM2A output on port pin. Default is disabled.

1: Enable PWM2A output on port pin.

Bit 4: POE2, PCA0 PWM2 main channel (PWM2O) output control.

0: Disable PWM2O output on port pin.

1: Enable PWM2O output on port pin. Default is enabled.

Bit 3: POE1, PCA0 PWM1 main channel (PWM1O) output control.

0: Disable PWM1O output on port pin.

1: Enable PWM1O output on port pin. Default is enabled.

Bit 2: POE0B, PCA0 PWM0 3rd channel (PWM0B) output control. 0: Disable PWM0B output on port pin. Default is disabled.

1: Enable PWM0B output on port pin.

Bit 1: POE0A, PCA0 PWM0 2nd channel (PWM0A) output control.

0: Disable PWM0A output on port pin. Default is disabled.

1: Enable PWM0A output on port pin.

Bit 0: POE0, PCA0 PWM0 main channel (PWM0O) output control.

0: Disable PWM0O output on port pin.

1: Enable PWM0O output on port pin. Default is enabled.

- AUXR7: Auxiliary Register 7**

SFR Page = 4 Only

SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	POE5	POE4	C0CKOE	SPI0M0	--	--	--	--
R/W	R/W	R/W	R/W	R/W	W	W	W	W
Reset Value	1	1	0	0	0	0	0	0

Bit 7: POE5, PCA0 PWM5 main channel (PWM5O) output control.

0: Disable PWM5O output on port pin.

1: Enable PWM5O output on port pin. Default is enabled.

Bit 6: POE4, PCA0 PWM4 main channel (PWM4O) output control.

0: Disable PWM4O output on port pin.

1: Enable PWM4O output on port pin. Default is enabled.

Bit 5: C0CKOE, PCA0 clock output enable.

0: Disable PCA0 clock output.

1: Enable PCA0 clock output with PCA0 base timer overflow rate/2.

- AUXR5: Auxiliary Register 5**

SFR Page = 2 only

SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
-----	---	---	---	---	---	---	---	---

Name	C0IC4S0	C0IC2S0	C0PPS1	C0PPS0	--	C0PS0	ECIPS0	C0COPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: C0IC4S0, PCA0 Input Channel 4 input port pin Selection.

C0IC4S0	CEX4 input
0	CEX4 Port Pin
1	T2EXI

Bit 6: C0IC2S0, PCA0 Input Channel 2 input port pin Selection.

C0IC2S0	CEX2 input
0	CEX2 Port Pin
1	T3EXI

Bit 5: C0PPS1, {PWM2A, PWM2B} Port pin Selection 1.

C0PPS1	PWM2A	PWM2B
0	P6.0	P6.1
1	P3.4	P3.5

Bit 4: C0PPS0, {PWM0A, PWM0B} Port pin Selection 0.

C0PPS0	PWM0A	PWM0B
0	P1.6	P1.7
1	P6.0	P6.1

Bit 3: Reserved.

Bit 2: C0PS0, PCA0 Port pin Selection 0.

C0PS0	CEX0	CEX2	CEX4
0	P2.2	P2.4	P1.7
1	P3.0	P2.4	P3.1

Bit 2: C0PS0, PCA0 Port pin Selection 0.

ECIPS0	ECI
0	P4.4
1	P1.6

Bit 0: C0COPS, PCA0 Clock Output (C0CKO) port pin Selection.

C0COPS	C0CKO
0	P4.7
1	P3.3

- **AUXR8:** Auxiliary Register 8
SFR Page = 5 only
SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	POE7	POE6	C0PPS2	--	KBIOPS0	S1COPS	--	--
R/W	R/W	R/W	R/W	W	R/W	R/W	W	W
Reset Value	1	1	0	0	0	0	0	0

Bit 7: POE7, PCA0 PWM7 main channel (PWM7O) output control. 0: Disable PWM7O output on port pin.
1: Enable PWM7O output on port pin. Default is enabled.

Bit 6: POE6, PCA0 PWM6 main channel (PWM6O) output control. 0: Disable PWM6O output on port pin. 1: Enable PWM6O output on port pin. Default is enabled.

Bit 5: C0PPS2, {PWM6, PWM7} Port pin Selection 2.

C0PPS2	PWM6	PWM7
0	P6.0	P6.1
1	P3.0	P3.1

17.4.13 Variable Resolution on Central Aligned PWM

In Section “Enhanced PWM Control”, it defines the central aligned PWM only support the 8/10/12/16-bit resolution. And in that mode, all of PCA functions, capture or compare, on other non-PWM modules are still available.

If it is necessary to apply the variable resolution on central aligned PWM, software must set C0M0 to enable the PCA0 to support this function operating. In this mode, PCA0 can support all compare or PWM modes. Otherwise, not support. Please note when using Central Aligned PWM with C0M0 = 1, Please note when using Central Aligned PWM with C0M0 = 1, we suggest to set the PWM module under 16-bit mode and the base timer need to use 16-bit 0xFFFF to minus the value to prevent unexpected error.

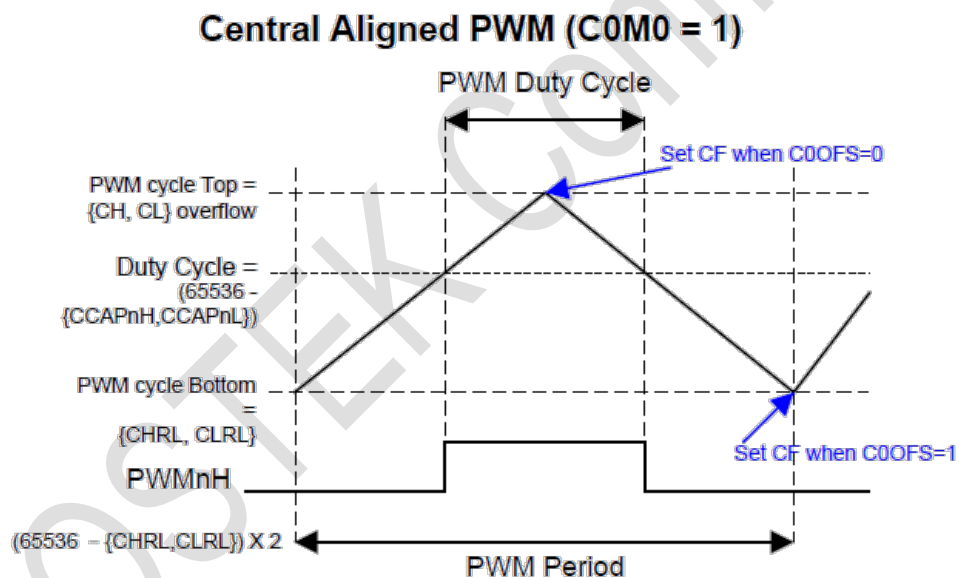


Figure 17-21. Central Aligned PWM with Variable Resolution

- **AUXR11:** Auxiliary Register 11

SFR Page = 8 only

SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	P30AM	--	--	--	--	--	C0M0	C0OFS
R/W	R/W	W	W	W	W	W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 1: C0M0, PCA0 Mode control 0.

0: Not support variable resolution on central aligned PWM.

1: Enable PCA0 variable resolution central aligned PWM. To enable this function, the PCAE also needs to be set.

Bit 0: C0OFS, PCA0 overflow flag selection when C0M0 is enabled.

0: CF is set on the top of central aligned PWM cycle.

1: CF is set on the bottom of central aligned PWM cycle.

18 Serial Port 0 (UART0)

The serial port 0 of CMT2380F17 support full-duplex transmission, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the register. However, if the first byte still hasn't been read by the time reception of the second byte is complete, one of the bytes will be lost. The serial port receive and transmit registers are both accessed at special function register S0BUF. Writing to S0BUF loads the transmit register, and reading from S0BUF accesses a physically separate receiver register.

The serial port can operate in **5** modes: Mode 0 provides *synchronous* communication while Modes 1, 2, and 3 provide *asynchronous* communication. The asynchronous communication operates as a full-duplex Universal

Asynchronous Receiver and Transmitter (UART), which can transmit and receive simultaneously and at different baud rates. Mode 4 in UART0 supports SPI master operation which data rate setting is same as Mode 0.

Mode 0: 8 data bits (LSB first) are transmitted or received through RXD0. TXD0 always outputs the shift clock. The baud rate can be selected to 1/12 or 1/4 the system clock frequency by URM0X3 setting in S0CFG register. The clock polarity of serial port Mode 0 can be selected by software. It is decided by P3.1 state before serial data shift in or shift out. [Figure 18-4](#) and [Figure 18-5](#) show the clock polarity waveform in Mode 0.

Mode 1: 10 bits are transmitted through TXD0 or received through RXD0. The framed data includes a start bit (0), 8 data bits (LSB first), and a stop bit (1), as shown in [Figure 18-1](#). On receive, the stop bit would be loaded into RB80 in S0CON register. The baud rate is variable.

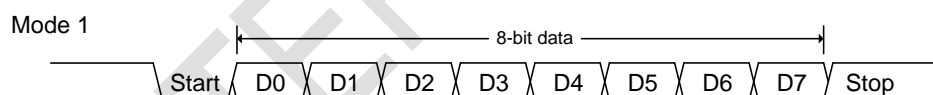


Figure 18-1. Mode 1 Data Frame

- **Mode 2** Mode 2: 11 bits are transmitted through TXD0 or received through RXD0. The frame data includes a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1), as shown in [Figure 18-2](#). On Transmit, the 9th data bit comes from TB80 in S0CON register can be assigned the value of 0 or 1. On receive, the 9th data bit would be loaded into RB80 in S0CON register, while the stop bit is ignored. The baud rate can be configured to 1/32 or 1/64 the system clock frequency.



Figure 18-2. Mode 2, 3 Data Frame

- **Mode 3:** Mode 3 is the same as Mode 2 except the baud rate is variable.

In all four modes, transmission is initiated by any instruction that uses S0BUF as a destination register. In Mode 0, reception is initiated by the condition RI0=0 and REN0=1. In the other modes, reception is initiated by the incoming start bit with 1-to-0 transition if REN0=1.

In addition to the standard operation, the UART0 can perform framing error detection by looking for missing stop bits, and automatic address recognition. .

18.1 Serial Port 0 Mode 0

Serial data enters and exits through RXD0. TXD0 outputs the shift clock. 8 bits are transmitted/received: 8 data bits (LSB first). The shift clock source can be selected to 1/12 or 1/4 the system clock frequency by URM0X3 setting in SOC_CFG register.

Figure 18–3 shows a simplified functional diagram of the serial port 0 in Mode 0.

Transmission is initiated by any instruction that uses S0BUF as a destination register. The “write to S0BUF” signal triggers the UART0 engine to start the transmission. The data in the S0BUF would be shifted into the RXD0(P3.0) pin by each raising edge shift clock on the TXD0(P3.1) pin. After eight raising edge of shift clocks passing, TI0 would be asserted by hardware to indicate the end of transmission and its interrupt vector can be switched to System Flag interrupt by BTI and UTIE gated. Figure 18–4 shows the transmission waveform in Mode 0.

Reception is initiated by the condition REN0=1 and RI0=0. At the next instruction cycle, the Serial Port 0 Controller writes the bits 11111110 to the receive shift register, and in the next clock phase activates Receive.

Receive enables Shift Clock which directly comes from RX Clock to the alternate output function of TXD0 pin. When Receive is active, the contents on the RXD0 pin would be sampled and shifted into shift register by falling edge of shift clock. After eight falling edge of shift clock, RI0 would be asserted by hardware to indicate the end of reception. Figure 18–5 shows the reception waveform in Mode 0.

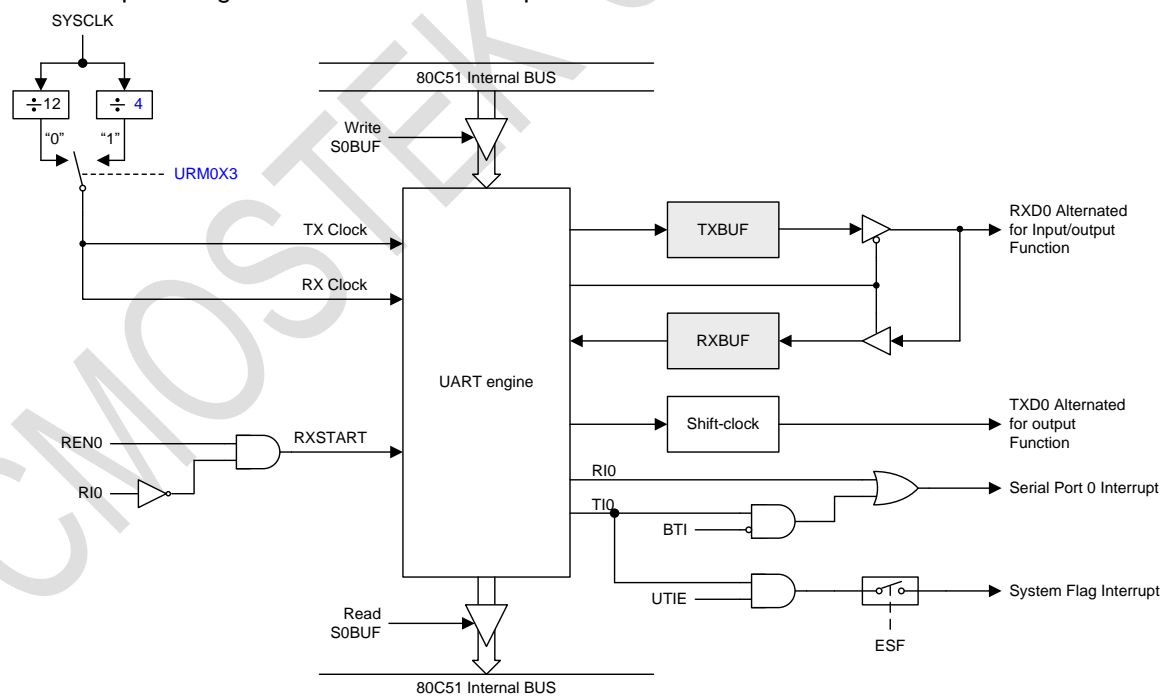


Figure 18-3. Serial Port 0 Mode 0

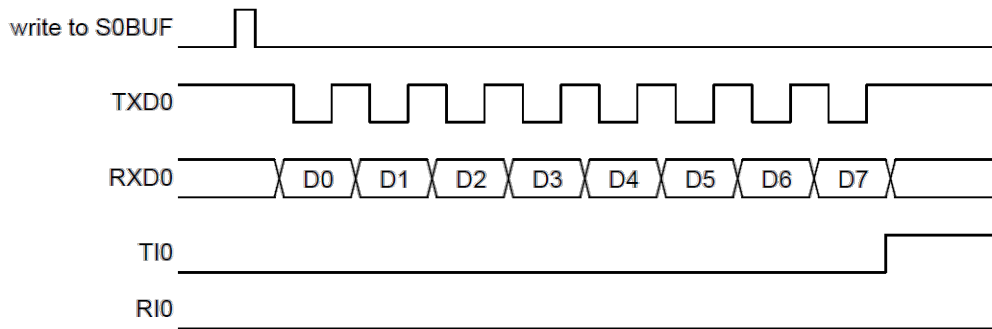


Figure 18-4. Mode 0 Transmission Waveform

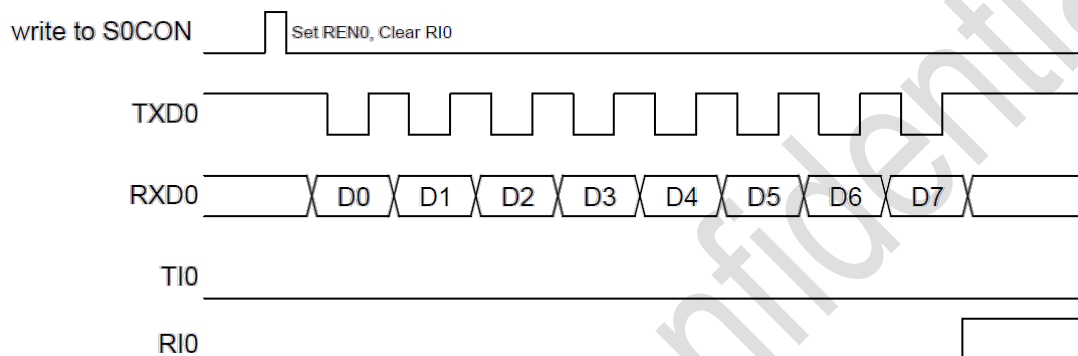


Figure 18-5. Mode 0 Reception Waveform

18.2 Serial Port 0 Mode 1

10 bits are transmitted through TXD0, or received through RXD0: a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB80 in S0CON. The baud rate is determined by the Timer 1 or Timer 2 overflow rate. Figure 18–1 shows the data frame in Mode 1 and Figure 18–6 shows a simplified functional diagram of the serial port in Mode 1.

Transmission is initiated by any instruction that uses S0BUF as a destination register. The “write to S0BUF” signal requests the UART0 engine to start the transmission. After receiving a transmission request, the UART0 engine would start the transmission at the raising edge of TX Clock. The data in the S0BUF would be serial output on the TXD0 pin with the data frame as shown in Figure 18–1 and data width depend on TX Clock. After the end of 8th data transmission, TI0 would be asserted by hardware to indicate the end of data transmission and its interrupt vector can be switched to System Flag interrupt by BTI and UTIE gated.

Reception is initiated when Serial Port 0 Controller detected 1-to-0 transition at RXD0 sampled by RCK. The data on the RXD0 pin would be sampled by Bit Detector in Serial Port 0 Controller. After the end of STOP-bit reception, RI0 would be asserted by hardware to indicate the end of data reception and load STOP-bit into RB80 in S0CON register.

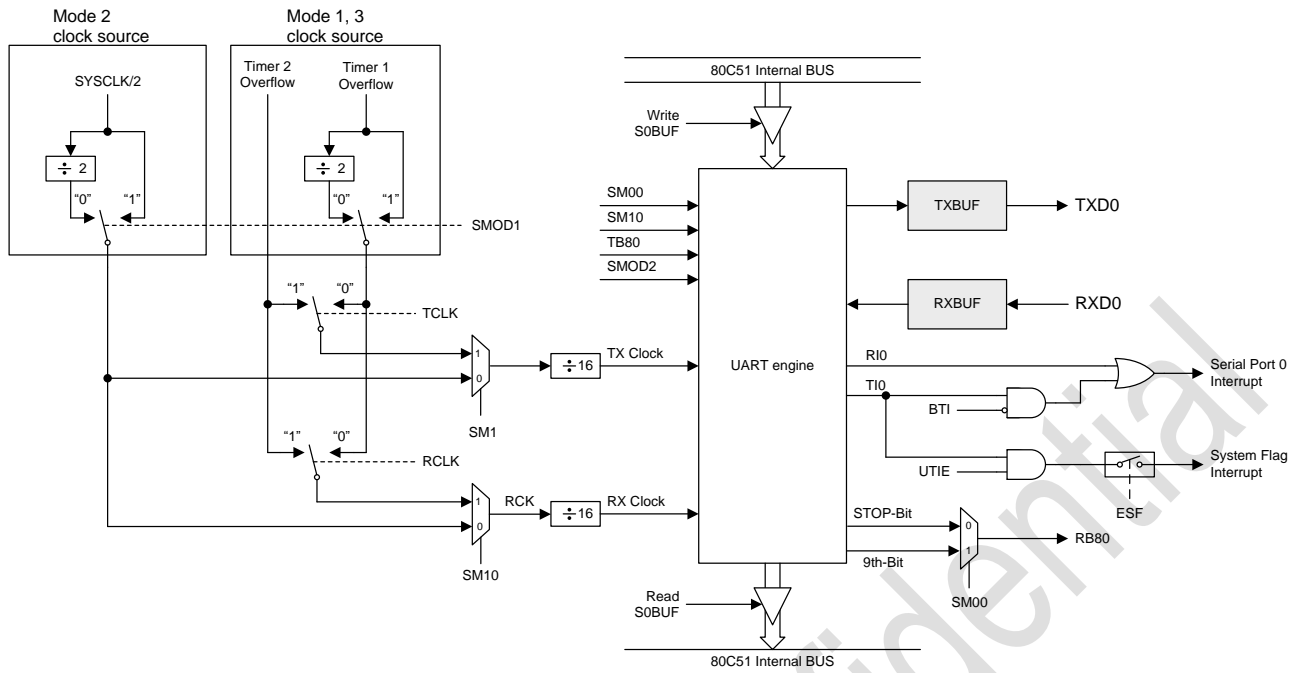


Figure 18-6. Serial Port Mode 1, 2, 3

18.3 Serial Port 0 Mode 2 and Mode 3

11 bits are transmitted through TXD0, or received through RXD0: a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On transmit, the 9th data bit (TB80) can be assigned the value of 0 or 1. On receive, the 9th data bit goes into RB80 in S0CON. The baud rate is programmable to select one of 1/16, 1/32 or 1/64 the system clock frequency in Mode 2. Mode 3 may have a variable baud rate generated from Timer 1 or Timer 2.

Figure 18–2 shows the data frame in Mode 2 and Mode 3. Figure 18–5 shows a functional diagram of the serial port in Mode 2 and Mode 3. The receive portion is exactly the same as in Mode 1. The transmit portion differs from Mode 1 only in the 9th bit of the transmit shift register.

The “write to S0BUF” signal requests the Serial Port 0 Controller to load TB80 into the 9th bit position of the transmit shift register and starts the transmission. After receiving a transmission request, the UART0 engine would start the transmission at the rising edge of TX Clock. The data in the S0BUF would be serial output on the TXD0 pin with the data frame as shown in Figure 18–2 and data width depend on TX Clock. After the end of 9th data transmission, TI0 would be asserted by hardware to indicate the end of data transmission and its interrupt vector can be switched to System Flag interrupt by BTI and UTIE gated.

Reception is initiated when the UART0 engine detected 1-to-0 transition at RXD0 sampled by RCK. The data on the RXD0 pin would be sampled by Bit Detector in UART0 engine. After the end of 9th data bit reception, RI0 would be asserted by hardware to indicate the end of data reception and load the 9th data bit into RB80 in S0CON register.

In all four modes, transmission is initiated by any instruction that use S0BUF as a destination register. Reception is initiated in mode 0 by the condition RI0 = 0 and REN0 = 1. Reception is initiated in the other modes by the incoming start bit with 1-to-0 transition if REN0=1.

18.4 Frame Error Detection

When used for framing error detection, the UART0 looks for missing stop bits in the communication. A missing stop bit will set the FE bit in the S0CON register. The FE bit shares the S0CON.7 bit with SM00 and the function of S0CON.7 is determined by SMOD0 bit (PCON.6). If SMOD0 is set then S0CON.7 functions as FE. S0CON.7 functions as SM00 when SMOD0 is cleared. When S0CON.7 functions as FE, it can only be cleared by firmware. Refer to Figure 18–7.

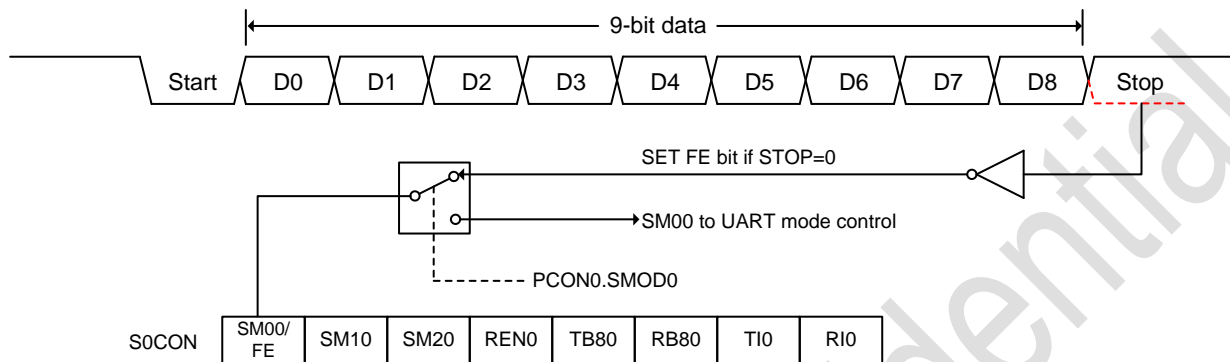


Figure 18-7. UART0 Frame Error Detection

18.5 Multiprocessor Communications

Modes 2 and 3 have a special provision for multiprocessor communications as shown in Figure 18–8. In these two modes, 9 data bits are received. The 9th bit goes into RB80. Then comes a stop bit. The port can be programmed such that when the stop bit is received, the serial port interrupt will be activated only if RB80=1. This feature is enabled by setting bit SM20 (in S0CON register). A way to use this feature in multiprocessor systems is as follows:

When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. With SM20=1, no slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and check if it is being addressed. The addressed slave will clear its SM20 bit and prepare to receive the data bytes that will be coming. The slaves that weren't being addressed leave their SM20 set and go on about their business, ignoring the coming data bytes.

SM20 has no effect in Mode 0, and in Mode 1 can be used to check the validity of the stop bit. In a Mode 1 reception, if SM20=1, the receive interrupt will not be activated unless a valid stop bit is received.

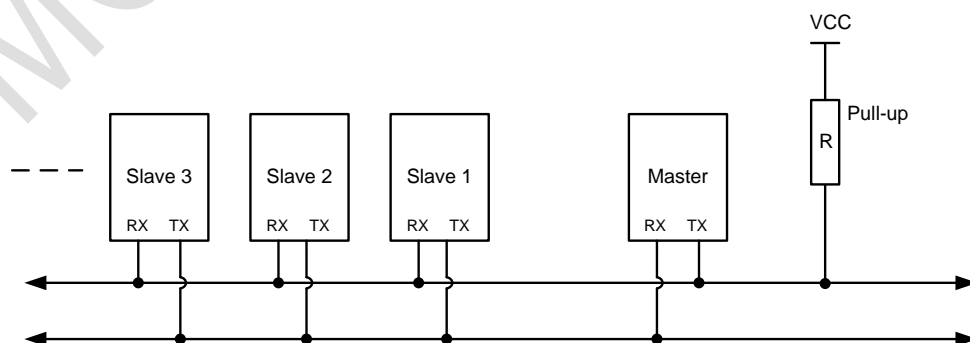


Figure 18-8. UART0 Multiprocessor Communications

18.6 Automatic Address Recognition

Automatic Address Recognition is a feature which allows the UART0 to recognize certain addresses in the serial bit stream by using hardware to make the comparisons. This feature saves a great deal of firmware overhead by eliminating the need for the firmware to examine every serial address which passes by the serial port. This feature is enabled by setting the SM20 bit in S0CON.

In the 9 bit UART modes, mode 2 and mode 3, the Receive Interrupt flag (RI0) will be automatically set when the received byte contains either the "Given" address or the "Broadcast" address. The 9-bit mode requires that the 9th information bit is a 1 to indicate that the received information is an address and not data. Automatic address recognition is shown in Figure 18–9. The 8 bit mode is called Mode 1. In this mode the RI flag will be set if SM20 is enabled and the information received has a valid stop bit following the 8 address bits and the information is either a Given or Broadcast address. Mode 0 is the Shift Register mode and SM20 is ignored.

Using the Automatic Address Recognition feature allows a master to selectively communicate with one or more slaves by invoking the Given slave address or addresses. All of the slaves may be contacted by using the Broadcast address. Two special Function Registers are used to define the slave's address, SADDR, and the address mask, SADEN.

SADEN is used to define which bits in the SADDR are to be used and which bits are "don't care". The SADEN mask can be logically ANDed with the SADDR to create the "Given" address which the master will use for addressing each of the slaves. Use of the Given address allows multiple slaves to be recognized while excluding others.

The following examples will help to show the versatility of this scheme:

Slave0	Slave1
SADDR=11000000	SADDR=11000000
SADEN = 1111	SADEN = 1111
1101Given=110000	1110Given=110000
X0	0X

In the above example SADDR is the same and the SADEN data is used to differentiate between the two slaves. Slave 0 requires a 0 in bit 0 and it ignores bit 1. Slave 1 requires a 0 in bit 1 and bit 0 is ignored. A unique address for Slave 0 would be 1100 0010 since slave 1 requires a 0 in bit 1. A unique address for slave 1 would be 1100 0001 since a 1 in bit 0 will exclude slave 0. Both slaves can be selected at the same time by an address which has bit 0 = 0 (for slave 0) and bit 1 = 0 (for slave 1). Thus, both could be addressed with 1100 0000.

In a more complex system the following could be used to select slaves 1 and 2 while excluding slave 0:

Slave0	Slave1	Slave2
SADDR=11000000	SADDR=11100000	SADDR=11100000
SADEN=11111001G	SADEN=11111010G	SADEN=11111100G
iven=11000XX0	iven=11100X0X	iven=111000XX

In the above example the differentiation among the 3 slaves is in the lower 3 address bits. Slave 0 requires that bit 0 = 0 and it can be uniquely addressed by 1110 0110. Slave 1 requires that bit 1 = 0 and it can be uniquely addressed by 1110 0101. Slave 2 requires that bit 2 = 0 and its unique address is 1110 0011. To select Slaves 0 and 1 and exclude Slave 2 use address 1110 0100, since it is necessary to make bit 2 = 1 to exclude slave 2.

The Broadcast Address for each slave is created by taking the logical OR of SADDR and SADEN. Zeros

in this result are treated as don't-cares. In most cases, interpreting the don't-cares as ones, the broadcast address will be FF hexadecimal.

Upon reset SADDR (SFR address 0xA9) and SADEN (SFR address 0xB9) are loaded with 0s. This produces a given address of all “don't cares” as well as a Broadcast address of all “don't cares”. This effectively disables the Automatic Addressing mode and allows the micro-controller to use standard 80C51 type UART drivers which do not make use of this feature.

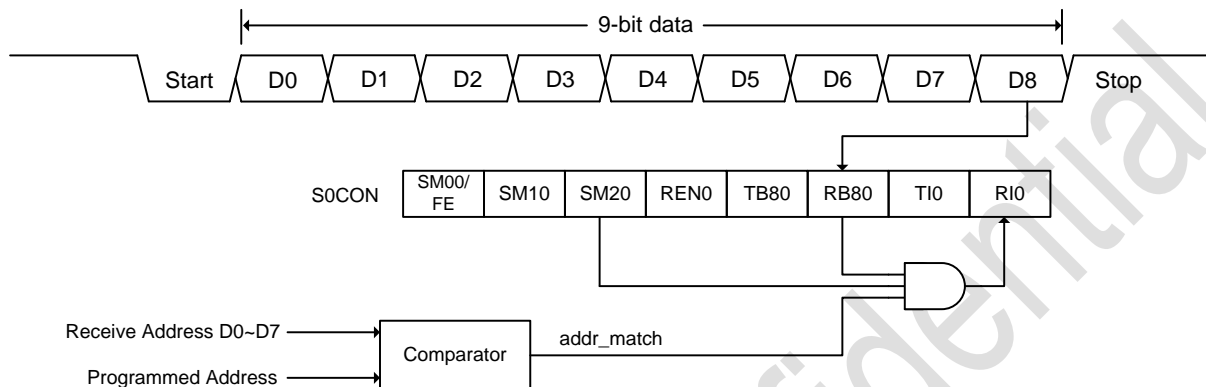


Figure 18-9. Auto-Address Recognition

Note:

- (1) After address matching ($addr_match=1$), Clear SM20 to receive data bytes
- (2) After all data bytes have been received, Set SM20 to wait for next address.

18.7 Baud Rate Setting

Bits T2X12 (T2MOD.4), T1X12 (AUXR2.3), URM0X3 (S0CFG.5) and SMOD2 (S0CFG.6) provide a new option for the baud rate setting, as listed below.

18.7.1 Baud Rate Selection in S0

In the Mode 1 and Mode 3 operation of the UART0, the software can select Timer 1 as the Baud Rate Generator by clearing bits TCLK and RCLK in T2CON register. At this time, if URTS bit (S0CFG.7) is set, then Timer 1 overflow signal will be replaced by the overflow signal of the UART1 Baud Rate Generator (S1BRG). In other words, the user can adopt S1BRG as the Baud Rate Generator for Mode 1 or Mode 3 of the UART0 as long as RCLK=0, TCLK=0 and URTS=1. In this condition, Timer 1 is free for other application. Of course, if UART1 (Mode 1 or Mode 3) is also operated at this time, these two UARTs will have the same baud rates.

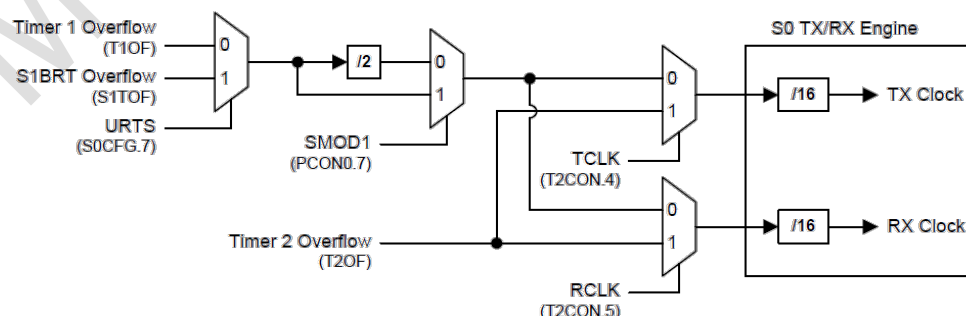


Figure 18-10. Baud Rate Source for the UART0

Table 18-1. SMOD2 application criteria in Mode 2

SMOD 2	SMOD 1	BaudRate	Not e	RecommendedMax.ReceiveError(%)
0	0	DefaultBaudRate	Standardfunction	± 3%
0	1	DoubleBaudRate	Standardfunction	± 3%
1	0	DoubleBaudRateX 2	Enhancedfunction	± 2%
1	1	DoubleBaudRateX 4	Enhancedfunction	± 1%

Note: When Timer 1 in Double Baud Rate x4 (SMOD1=1 & SMOD2=1) mode, the TH1 can not equal to 254 & 255.

Table 18-2. S0 Mode 2 Baud Rates @ FSYSClk=11.0592MHz

BaudRate	URM0X3	SMOD2	SMOD1	Error
172,800	0	0	0	±0.0%
345,600	0	0	1	±0.0%
691,200	0	1	0	±0.0%
1,382,400	0	1	1	±0.0%
57,600	1	0	0	±0.0%
115,200	1	0	1	±0.0%
230,400	1	1	0	±0.0%
460,800	1	1	1	±0.0%

Table 18-3. S0 Mode 2 Baud Rates @ FSYSClk=12.00MHz

BaudRate	URM0X3	SMOD2	SMOD1	Error
187,500	0	0	0	±0.0%
375,000	0	0	1	±0.0%
750,000	0	1	0	±0.0%
1,500,000	0	1	1	±0.0%
62,500	1	0	0	±0.0%
125,000	1	0	1	±0.0%
250,000	1	1	0	±0.0%
500,000	1	1	1	±0.0%

18.7.4 Baud Rate in Mode 1 & 3

18.7.4.1 Using Timer 1 as the Baud Rate Generator

$$\text{Mode 1, 3 Baud Rate} = \frac{2^{\text{SMOD1}} \times 2^{(\text{SMOD2} \times 2)}}{32} \times \frac{F_{\text{SYSClk}}}{12 \times (256 - \text{TH1})} ; \text{T1X12}=0$$

$$\text{or} = \frac{2^{\text{SMOD1}} \times 2^{(\text{SMOD2} \times 2)}}{32} \times \frac{F_{\text{SYSClk}}}{1 \times (256 - \text{TH1})} ; \text{T1X12}=1$$

Note:

If SMOD2=0, T1X12=0, the baud rate formula is as same as standard 8051. If SMOD2=1, there is an enhanced function for baud rate setting. Table 18-4 defines the Baud Rate setting with SMOD2 factor in Timer 1 baud rate generator.

Table 18-4. SMOD2 application criteria in Mode 1 & 3 using Timer 1

SMOD2	SMOD1	BaudRate	Note	RecommendedMax. ReceiveError(%)
0	0	DefaultBaudRate	Standardfunction	±3%
0	1	DoubleBaudRate	Standardfunction	±3%
1	0	DoubleBaudRateX2	Enhancedfunction	±2%
1	1	DoubleBaudRateX4	Enhancedfunction	±1%

Note: When Timer 1 in Double Baud Rate x4 (SMOD1=1 & SMOD2=1) mode, the TH1 can not equal to 254 & 255.

Table 18-5 ~ Table 18-20 list various commonly used baud rates and how they can be obtained from Timer 1 in its 8-Bit Auto-Reload Mode. For the non-standard Baud Rate, the maximum frequency is 6MHz when FSYSCCLK = 48MHz).

Table 18-5. Timer 1 Generated Commonly Used Baud Rates @ FSYSCCLK=11.0592MHz

Baud Rate	TH1,theReloadValue					
	T1X12=0 & SMOD2=0			T1X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	232	208	0.0%	--	--	--
2400	244	232	0.0%	112	--	0.0%
4800	250	244	0.0%	184	112	0.0%
9600	253	250	0.0%	220	184	0.0%
14400	254	252	0.0%	232	208	0.0%
19200	--	253	0.0%	238	220	0.0%
28800	255	254	0.0%	244	232	0.0%
38400	--	--	--	247	238	0.0%
57600	--	255	0.0%	250	244	0.0%
115200	--	--	--	253	250	0.0%
230400	--	--	--	--	253	0.0%

Table 18-6. Timer 1 Generated High Baud Rates @ FSYSCCLK =11.0592MHz

Baud Rate	TH1,theReloadValue					
	T1X12=0 & SMOD2=1			T1X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	Error	SMOD=0	SMOD=1	Error
230.4K	--	255	0.0%	250	244	0.0%
460.8K	--	--	--	253	250	0.0%
691.2K	--	--	--	254	252	0.0%
921.6K	--	--	--	--	253	0.0%
1.3824M	--	--	--	255	254	0.0%
2.7648M	--	--	--	--	255	0.0%

Table 18-7. Timer 1 Generated Commonly Used Baud Rates @ FSYSCLK=22.1184MHz

Baud Rate	TH1,theReloadValue					
	T1X12=0 & SMOD2=0			T1X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	208	160	0.0%	--	--	--
2400	232	208	0.0%	--	--	0.0%
4800	244	232	0.0%	112	--	0.0%
9600	250	244	0.0%	184	112	0.0%
14400	252	248	0.0%	208	160	0.0%
19200	253	250	0.0%	220	184	0.0%
28800	254	252	0.0%	232	208	0.0%
38400	--	253	0.0%	238	220	0.0%
57600	255	254	0.0%	244	232	0.0%
115200	--	255	0.0%	250	244	0.0%
230400	--	--	--	253	250	0.0%
460800	--	--	--	--	253	0.0%

Table 18-8. Timer 1 Generated High Baud Rates @ FSYSCLK=22.1184MHz

Baud Rate	TH1,theReloadValue					
	T1X12=0 & SMOD2=1			T1X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	Error	SMOD=0	SMOD=1	Error
460.8K	--	255	0.0%	250	244	0.0%
691.2K	--	--	--	252	248	0.0%
921.6K	--	--	--	253	250	0.0%
1.3824M	--	--	--	254	252	0.0%
1.8432M	--	--	--	--	253	0.0%
2.7648M	--	--	--	255	254	0.0%
5.5296M	--	--	--	--	255	0.0%

Table 18-9. Timer 1 Generated Commonly Used Baud Rates @ FSYSCLK=12.0MHz

Baud Rate	TH1,theReloadValue					
	T1X12=0 & SMOD2=0			T1X12=1 & SMOD2=0		
	SMOD=0	SMOD=1	Error	SMOD=0	SMOD=1	Error
1200	230	204	0.16%	--	--	--
2400	243	230	0.16%	100	--	0.16%
4800	--	243	0.16%	178	100	0.16%
9600	--	--	--	217	178	0.16%
14400	--	--	--	230	204	0.16%
19200	--	--	--	--	217	0.16%
28800	--	--	--	243	230	0.16%
38400	--	--	--	246	236	2.34%
57600	--	--	--	--	243	0.16%

Baud Rate	TH1,theReloadValue					
	T1X12=0 & SMOD2=0			T1X12=1 & SMOD2=0		
	SMOD=0	SMOD=1	Error	SMOD=0	SMOD=1	Error
115200	--	--	--	--	--	--

Table 18-10. Timer 1 Generated High Baud Rates @ FSYCLK=12.0MHz

Baud Rate	TH1,theReloadValue					
	T1X12=0 & SMOD2=1			T1X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	Error	SMOD=0	SMOD=1	Error
115.2K	--	--	--	243	230	0.16%
230.4K	--	--	--	--	243	0.16%
460.8K	--	--	--	--	--	--

Table 18-11. Timer 1 Generated Commonly Used Baud Rates @ FSYCLK=24.0MHz

Baud Rate	TH1,theReloadValue					
	T1X12=0 & SMOD2=0			T1X12=1 & SMOD2=0		
	SMOD=0	SMOD=1	Error	SMOD=0	SMOD=1	Error
1200	204	152	0.16%	--	--	--
2400	230	204	0.16%	--	--	--
4800	243	230	0.16%	100	--	0.16%
9600	--	243	0.16%	178	100	0.16%
14400	--	--	--	204	152	0.16%
19200	--	--	--	217	178	0.16%
28800	--	--	--	230	204	0.16%
38400	--	--	--	--	217	0.16%
57600	--	--	--	243	230	0.16%
115200	--	--	--	--	243	0.16%

Table 18-12. Timer 1 Generated High Baud Rates @ FSYCLK=24.0MHz

Baud Rate	TH1,theReloadValue					
	T1X12=0 & SMOD2=1			T1X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	Error	SMOD=0	SMOD=1	Error
230.4K	--	--	--	243	230	0.16%
460.8K	--	--	--	--	243	0.16%
691.2K	--	--	--	--	--	--
921.6K	--	--	--	--	--	--

Table 18-13. Timer 1 Generated Commonly Used Baud Rates @ FSYSCLK=29.4912MHz

Baud Rate	TH1, the Reload Value					
	T1X12=0 & SMOD2=0			T1X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	192	128	0.0%	--	--	--
2400	224	192	0.0%	--	--	--
4800	240	224	0.0%	64	--	0.0%
9600	248	240	0.0%	160	64	0.0%
14400	--	--	--	192	128	0.0%
19200	252	248	0.0%	208	160	0.0%
28800	--	--	--	224	192	0.0%
38400	--	--	--	232	208	0.0%
57600	--	--	--	240	224	0.0%
115200	--	--	--	248	240	0.0%
230.4K	--	--	--	252	248	0.0%
460.8K	--	--	--	254	252	0.0%
921.6K	--	--	--	255	254	0.0%
1.8432M	--	--	--	--	255	0.0%

Table 18-14. Timer 1 Generated High Baud Rates @ FSYSCLK=29.4912MHz

Baud Rate	TH1, the Reload Value					
	T1X12=0 & SMOD2=1			T1X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1.8432M	--	--	--	254	252	0.0%
2.7648M	--	--	--	--	--	--
3.6864M	--	--	--	--	--	--

Table 18-15. Timer 1 Generated Commonly Used Baud Rates @ FSYSCLK=44.2368MHz

Baud Rate	TH1, the Reload Value					
	T1X12=0 & SMOD2=0			T1X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	160	64	0.0%	--	--	--
2400	208	160	0.0%	--	--	--
4800	232	208	0.0%	--	--	--
9600	244	232	0.0%	112	--	0.0%
14400	248	240	0.0%	160	64	0.0%
19200	250	244	0.0%	184	112	0.0%
28800	252	248	0.0%	208	160	0.0%
38400	253	250	0.0%	220	184	0.0%
57600	254	252	0.0%	232	208	0.0%
115200	255	254	0.0%	244	232	0.0%
230.4K	--	255	0.0%	250	244	0.0%

Baud Rate	TH1, the Reload Value					
	T1X12=0 & SMOD2=0			T1X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
460.8K	--	--	--	253	250	0.0%
921.6K	--	--	--	--	253	0.0%
1.8432M	--	--	--	--	--	--
2.7648M	--	--	--	--	255	0.0%

Table 18-16. Timer 1 Generated High Baud Rates @ FSYSCLK=44.2368MHz

Baud Rate	TH1, the Reload Value					
	T1X12=0 & SMOD2=1			T1X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
2.7648M	--	--	--	254	252	0.0%
3.6864M	--	--	--	--	253	0.0%
5.5296M	--	--	--	--	--	0.0%

Table 18-17. Timer 1 Generated Commonly Used Baud Rates @ FSYSCLK=32MHz

Baud Rate	TH1, the Reload Value					
	T1X12=0 & SMOD2=0			T1X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	187	118	0.64%	--	--	--
2400	221	186	-0.79%	--	--	--
4800	239	222	2.12%	48	--	0.16%
9600	--	239	2.12%	152	48	0.16%
14400	--	--	--	187	118	0.64%
19200	--	--	--	204	152	0.16%
28800	--	--	--	221	186	-0.79%
38400	--	--	--	230	204	0.16%
57600	--	--	--	239	222	2.12%
115200	--	--	--	--	239	2.12%

Table 18-18. Timer 1 Generated High Baud Rates @ FSYSCLK=32MHz

Baud Rate	TH1, the Reload Value					
	T1X12=0 & SMOD2=1			T1X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
38400	--	--	--	152	48	0.16%
57600	--	--	--	--	117	-0.08%
115200	--	--	--	--	187	0.64%
230.4K	--	--	--	--	221	-0.79%

Table 18-19. Timer 1 Generated Commonly Used Baud Rates @ FSYSCLK=48.0MHz

Baud Rate	TH1, the Reload Value					
	T1X12=0 & SMOD2=0			T1X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	152	48	0.16%	--	--	--
2400	204	152	0.16%	--	--	--
4800	230	204	0.16%	--	--	--
9600	243	230	0.16%	100	--	0.16%
14400	--	239	2.12%	152	48	0.16%
19200	--	243	0.16%	178	100	0.16%
28800	--	--	--	204	152	0.16%
38400	--	--	--	217	178	0.16%
57600	--	--	--	230	204	0.16%
115200	--	--	--	243	230	0.16%
230.4K	--	--	--	--	243	0.16%

Table 18-20. Timer 1 Generated High Baud Rates @ FSYSCLK=48.0MHz

Baud Rate	TH1, the Reload Value					
	T1X12=0 & SMOD2=1			T1X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
230.4K	--	--	--	230	204	0.16%
460.8K	--	--	--	243	230	0.16%
921.6K	--	--	--	--	243	0.16%
1.8432M	--	--	--	--	--	--
2.7648M	--	--	--	--	--	--
3.6864M	--	--	--	--	--	--
5.5296M	--	--	--	--	--	--

18.7.4.2 Using Timer 2 as the Baud Rate Generator

When Timer 2 is used as the baud rate generator (either TCLK or RCLK in T2CON is '1'), the baud rate is as follows.

$$\text{Mode 1, 3 Baud Rate} = \frac{2^{\text{SMOD2} \times (\text{SMOD1} + 1)} \times F_{\text{SYSCLK}}}{32 \times (65536 - (\text{RCAP2H}, \text{RCAP2L}))}; \text{T2X12}=0$$

$$\text{or} = \frac{2^{\text{SMOD2} \times (\text{SMOD1} + 1)} \times F_{\text{SYSCLK}}}{16 \times (65536 - (\text{RCAP2H}, \text{RCAP2L}))}; \text{T2X12}=1$$

Note:

If SMOD2=0, the baud rate formula is as same as standard 8051. If SMOD2=1, there is an enhanced function for baud rate setting. Table 18–21 defines the Baud Rate setting with SMOD2 factor in Timer 2 baud rate generator.

Table 18-21. SMOD2 application criteria in Mode 1 & 3 using Timer 2

SMOD2	SMOD1	BaudRate	Note	Recommended Max.ReceiveError(%)
0	X	DefaultBaudRate	Standardfunction	±3%
1	0	DoubleBaudRate	Enhancedfunction	±3%
1	1	DoubleBaudRateX2	Enhancedfunction	±2%

Note: When Timer 2 in Double Baud Rate x2 (SMOD1=1 & SMOD2=1) mode, the RCAP2H & RPAC2L can not equal to 65534 & 65535.

Table 18–22 ~ Table 18–37 list various commonly used baud rates and how they can be obtained from Timer 2 in its Baud-Rate Generator Mode.

Table 18-22. Timer 2 Generated Commonly Used Baud Rates @ FSYCLK=11.0592MHz

Baud Rate	[RCAP2H, RCAP2L], the Reload Value					
	T2X12=0 & SMOD2=0			T2X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	65248	65248	0.0%	64960	64960	0.0%
2400	65392	65392	0.0%	65248	65248	0.0%
4800	65464	65464	0.0%	65392	65392	0.0%
9600	65500	65500	0.0%	65464	65464	0.0%
14400	65512	65512	0.0%	65488	65488	0.0%
19200	65518	65518	0.0%	65500	65500	0.0%
28800	65524	65524	0.0%	65512	65512	0.0%
38400	65527	65527	0.0%	65518	65518	0.0%
57600	65530	65530	0.0%	65524	65524	0.0%
115200	65533	65533	0.0%	65530	65530	0.0%
230400	--	--	--	65533	65533	0.0%

CMOSTEK Confidential

Table 18-23. Timer 2 Generated High Baud Rates @ FSYCLK=11.0592MHz

Baud Rate	[RCAP2H, RCAP2L], the Reload Value					
	T2X12=0 & SMOD2=1			T2X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	Error	SMOD=0	SMOD=1	Error
230.4K	65533	65530	0.0%	65530	65524	0.0%
460.8K	--	65533	0.0%	65533	65530	0.0%
691.2K	65535	65534	0.0%	65534	65532	0.0%
921.6K	--	--	--	--	65533	0.0%
1.3824M	--	65535	0.0%	65535	65534	0.0%
2.7648M	--	--	--	--	65535	0.0%

Table 18-24. Timer 2 Generated Commonly Used Baud Rates @ FSYCLK=22.1184MHz

Baud Rate	[RCAP2H, RCAP2L], the Reload Value					
	T2X12=0 & SMOD2=0			T2X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	64960	64960	0.0%	64384	64384	0.0%
2400	65248	65248	0.0%	64960	64960	0.0%
4800	65392	65392	0.0%	65248	65248	0.0%
9600	65464	65464	0.0%	65392	65392	0.0%
14400	65488	65488	0.0%	65440	65440	0.0%
19200	65500	65500	0.0%	65464	65464	0.0%
28800	65512	65512	0.0%	65488	65488	0.0%
38400	65518	65518	0.0%	65500	65500	0.0%
57600	65524	65524	0.0%	65512	65512	0.0%
115200	65530	65530	0.0%	65524	65524	0.0%
230400	65533	65533	0.0%	65530	65530	0.0%
460800	--	--	--	65533	65533	0.0%

Table 18-25. Timer 2 Generated High Baud Rates @ FSYCLK=22.1184MHz

Baud Rate	[RCAP2H, RCAP2L], the Reload Value					
	T2X12=0 & SMOD2=1			T2X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	Error	SMOD=0	SMOD=1	Error
460.8K	65533	65530	0.0%	65530	65524	0.0%
691.2K	65534	65532	0.0%	65532	65528	0.0%
921.6K	--	65533	0.0%	65533	65530	0.0%
1.3824M	65535	65534	0.0%	65534	65532	0.0%
1.8432M	--	--	--	--	65533	0.0%
2.7648M	--	65535	0.0%	65535	65534	0.0%
5.5296M	--	--	--	--	65535	0.0%

Table 18-26. Timer 2 Generated Commonly Used Baud Rates @ FSYCLK=12.0MHz

Baud Rate	[RCAP2H, RCAP2L], the Reload Value					
	T2X12=0 & SMOD2=0			T2X12=1 & SMOD2=0		
	SMOD=0	SMOD=1	Error	SMOD=0	SMOD=1	Error
1200	65224	65224	0.16%	64912	64912	0.16%
2400	65380	65380	0.16%	65224	65224	0.16%
4800	65458	65458	0.16%	65380	65380	0.16%
9600	65497	65497	0.16%	65458	65458	0.16%
14400	65510	65510	0.16%	65484	65484	0.16%
19200	65516	65516	2.34%	65497	65497	0.16%
28800	65523	65523	0.16%	65510	65510	0.16%
38400	--	--	--	65516	65516	2.34%
57600	--	--	--	65523	65523	0.16%
115200	--	--	--	--	--	--

Table 18-27. Timer 2 Generated High Baud Rates @ FSYCLK=12.0MHz

Baud Rate	[RCAP2H, RCAP2L], the Reload Value					
	T2X12=0 & SMOD2=1			T2X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	Error	SMOD=0	SMOD=1	Error
115.2K	--	65523	0.16%	65523	65510	0.16%
230.4K	--	--	--	--	65523	0.16%
460.8K	--	--	--	--	--	--

Table 18-28. Timer 2 Generated Commonly Used Baud Rates @ FSYCLK=24.0MHz

Baud Rate	[RCAP2H, RCAP2L], the Reload Value					
	T2X12=0 & SMOD2=0			T2X12=1 & SMOD2=0		
	SMOD=0	SMOD=1	Error	SMOD=0	SMOD=1	Error
1200	64912	64912	0.16%	64288	64288	0.16%
2400	65224	65224	0.16%	64912	64912	0.16%
4800	65380	65380	0.16%	65224	65224	0.16%
9600	65458	65458	0.16%	65380	65380	0.16%
14400	65484	65484	0.16%	65432	65432	0.16%
19200	65497	65497	0.16%	65458	65458	0.16%
28800	65510	65510	0.16%	65484	65484	0.16%
38400	65516	65516	2.34%	65497	65497	0.16%
57600	65523	65523	0.16%	65510	65510	0.16%
115200	--	--	--	65523	65523	0.16%

Table 18-29. Timer 2 Generated High Baud Rates @ FSYSCLK=24.0MHz

Baud Rate	[RCAP2H, RCAP2L], the Reload Value					
	T2X12=0 & SMOD2=1			T2X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	Error	SMOD=0	SMOD=1	Error
230.4K	--	65523	0.16%	65523	65510	0.16%
460.8K	--	--	--	--	65523	0.16%
691.2K	--	--	--	--	--	--
921.6K	--	--	--	--	--	--

Table 18-30. Timer 2 Generated Commonly Used Baud Rates @ FSYSCLK=29.4912MHz

Baud Rate	[RCAP2H, RCAP2L], the Reload Value					
	T2X12=0 & SMOD2=0			T2X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	64768	64768	0.0%	64000	64000	0.0%
2400	65152	65152	0.0%	64768	64768	0.0%
4800	65344	65344	0.0%	65152	65152	0.0%
9600	65440	65440	0.0%	65344	65344	0.0%
14400	65472	65472	0.0%	65408	65408	0.0%
19200	65488	65488	0.0%	65440	65440	0.0%
28800	65504	65504	0.0%	65472	65472	0.0%
38400	65512	65512	0.0%	65488	65488	0.0%
57600	65520	65520	0.0%	65504	65504	0.0%
115200	65528	65528	0.0%	65520	65520	0.0%
230.4K	65532	65532	0.0%	65528	65528	0.0%
460.8K	65534	65534	0.0%	65532	65532	0.0%
691.2K	--	--	--	--	--	--
921.6K	65535	65535	0.0%	65534	65534	0.0%

Table 18-31. Timer 2 Generated High Baud Rates @ FSYSCLK=29.4912MHz

Baud Rate	[RCAP2H, RCAP2L], the Reload Value					
	T2X12=0 & SMOD2=1			T2X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
921.6K	65534	65532	0.0%	65532	65528	0.0%

Table 18-32. Timer 2 Generated Commonly Used Baud Rates @ FSYCLK=44.2368MHz

Baud Rate	[RCAP2H, RCAP2L], the Reload Value					
	T2X12=0 & SMOD2=0			T2X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	64384	64384	0.0%	63232	63232	0.0%
2400	64960	64960	0.0%	64384	64384	0.0%
4800	65248	65248	0.0%	64960	64960	0.0%
9600	65392	65392	0.0%	65248	65248	0.0%
14400	65440	65440	0.0%	65344	65344	0.0%
19200	65464	65464	0.0%	65392	65392	0.0%
28800	65488	65488	0.0%	65440	65440	0.0%
38400	65500	65500	0.0%	65464	65464	0.0%
57600	65512	65512	0.0%	65488	65488	0.0%
115200	65524	65524	0.0%	65512	65512	0.0%
230.4K	65530	65530	0.0%	65524	65524	0.0%
460.8K	65533	65533	0.0%	65530	65530	0.0%
691.2K	65534	65534	0.0%	65532	65532	0.0%
921.6K	--	--	--	65533	65533	0.0%
1.3824M	65535	65535	0.0%	65534	65534	0.0%
2.7648M	--	--	--	65535	65535	0.0%

Table 18-33. Timer 2 Generated High Baud Rates @ FSYCLK=44.2368MHz

Baud Rate	[RCAP2H, RCAP2L], the Reload Value					
	T2X12=0 & SMOD2=1			T2X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
2.7648M	65535	65534	0.0%	65534	65532	0.0%
5.5296M	--	65535	0.0%	65535	--	0.0%
11.0592M	--	--	--	--	--	0.0%

Table 18-34. Timer 2 Generated Commonly Used Baud Rates @ FSYCLK=32MHz

Baud Rate	[RCAP2H, RCAP2L], the Reload Value					
	T2X12=0 & SMOD2=0			T2X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	64703	64703	0.04%	63870	63870	0.04%
2400	65120	65120	0.16%	64703	64703	0.04%
4800	65328	65328	-0.16%	65120	65120	0.16%
9600	65432	65432	-0.16%	65328	65328	0.16%
14400	65467	65467	0.64%	65398	65398	0.64%
19200	65484	65484	0.16%	65432	65432	0.16%
28800	65502	65502	2.12%	65467	65467	0.64%
38400	65510	65510	0.16%	65484	65484	0.16%
57600	65519	65519	2.12%	65502	65502	2.12%

Baud Rate	[RCAP2H, RCAP2L], the Reload Value					
	T2X12=0 & SMOD2=0			T2X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
115200	--	--	--	65519	65519	2.12%

Table 18-35. Timer 2 Generated High Baud Rates @ F_{SYSC}CLK=32MHz

Baud Rate	[RCAP2H, RCAP2L], the Reload Value					
	T2X12=0 & SMOD2=1			T2X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
115200	65519	65502	2.12%	65501	65467	0.64%
230.4K	--	--	--	--	65501	-0.79%

Table 18-36. Timer 2 Generated Commonly Used Baud Rates @ F_{SYSC}CLK=48.0MHz

Baud Rate	[RCAP2H, RCAP2L], the Reload Value					
	T2X12=0 & SMOD2=0			T2X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	64286	64286	0.00%	63036	63036	0.00%
2400	64911	64911	0.00%	64286	64286	0.00%
4800	65224	65224	0.16%	64911	64911	0.00%
9600	65380	65380	0.16%	65224	65224	0.16%
14400	65432	65432	0.16%	65328	65328	0.16%
19200	65458	65458	0.16%	65380	65380	0.16%
28800	65484	65484	0.16%	65432	65432	0.16%
38400	65497	65497	0.16%	65458	65458	0.16%
57600	65510	65510	0.16%	65484	65484	0.16%
115200	65523	65523	0.16%	65510	65510	0.16%
230.4K	--	--	--	65523	65523	0.16%

Table 18-37. Timer2GeneratedHighBaudRates@F_{SYSC}CLK=48.0MHz

Baud Rate	[RCAP2H, RCAP2L], the Reload Value					
	T2X12=0 & SMOD2=1			T2X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
230.4K	65523	65510	0.16%	65510	65484	0.16%
460.8K	--	65522	0.16%	65523	65510	0.16%
691.2K	--	--	--	--	--	--
921.6K	--	--	--	--	65523	0.16%

18.7.4.3 Using S0 Baud Rate Timer as the Baud Rate Generator (S0BRG)

The S0 of CMT2380F17 has embedded a dedicated baud rate generator (S0BRG), which detailed function is described in Section “18.10.2 Independent Baud Rate Generator S0BRG for S0”. When S0BRG is used as the baud rate generator of S0, the baud rate is as follows.

$$\text{Mode 1, 3 Baud Rate} = \frac{2^{(\text{SMOD2})}}{32} \times \frac{F_{\text{SYSCLK}}}{12 \times (256 - \text{S0BRT})} ; \text{S0TX12}=0, \text{SMOD1}=0$$

$$\text{or} = \frac{2^{(\text{SMOD2})}}{32} \times \frac{F_{\text{SYSCLK}}}{1 \times (256 - \text{S0BRT})} ; \text{S0TX12}=1, \text{SMOD1}=0$$

18.7.4.4 Using S1 Baud Rate Timer as the Baud Rate Generator

The secondary UART (S1) in CMT2380F17 has an independent baud-rate generator. S0 can set URTS (S0CFG.7) to select the S1BRT as the timer source for UART Mode 1 and Mode 3. See Section “19.7 S1 Baud Rate Generator for S0” for the details on S0 baud rate select.

18.8 Serial Port 0 Mode 4 (SPI Master)

The Serial Port 0 of CMT2380F17 is embedded an additional Mode 4 to support SPI master engine. The Mode 4 is selected by SM30, SM00 and SM10. Table 18–38 shows the serial port mode definition in CMT2380F17.

Table 18-38. Serial Port 0 Mode Selection

SM30	SM00	SM10	Mode	Description	BaudRate
0	0	0	0	shiftregister	SYSCCLK/12or SYSCCLK/4
0	0	1	1	8-bitUART	Variable
0	1	0	2	9-bitUART	SYSCCLK/64,/32,/16,/8or /192,/96,/48,/24
0	1	1	3	9-bitUART	variable
1	0	0	4	SPI Master	SYSCCLK/12or SYSCCLK/4
1	0	1	5	Reserved	Reserved
1	1	0	6	Reserved	Reserved
1	1	1	7	Reserved	Reserved

URM0X3 also controls the SPI transfer speed. If URM0X3 = 0, the SPI clock frequency is SYSCCLK/12. If URM0X3 = 1, the SPI clock frequency is SYSCCLK/4.

The SPI master in CMT2380F17 uses the TXD0 as SPICLK, RXD0 as MOSI, and S0MI as MISO. nSS is selected by MCU software on other port pin. Figure 18–11 shows the SPI connection. It also can support the configuration for multiple slave communication in Figure 18–12.

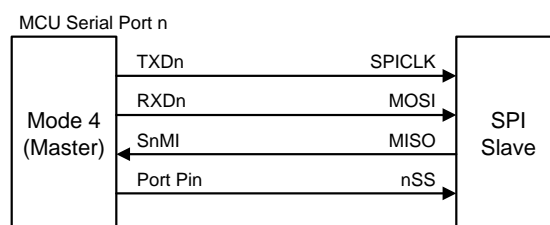


Figure 18-11. Serial Port 0 Mode 4, Single Master and Single Slave configuration (n = 0)

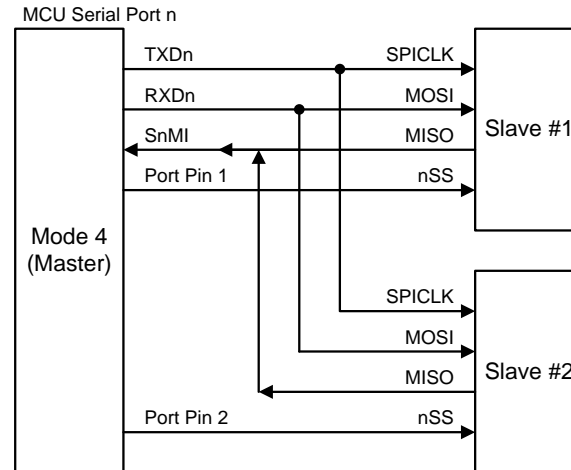


Figure 18-12. SerialPort0Mode4, Single Master and Multiple Slaves configuration (n=0)

The SPI master satisfies the transfer with the full function SPI module of Megawin MG82/84 series MCU with CPOL, CPHA and DORD selection. For CPOL and CPHA condition, CMT2380F17 uses an easy way by initialize SPI clock assigned port pin (TXD0, P3.1/P4.5) polarity to fit them. Table 18–39 shows the serial port Mode 4 mapping with the four SPI operating mode.

Table 18-39. SPI mode mapping with Serial Port Mode 4 configuration

SPI Mode	CPOL	CPHA	Configuration in TXD0
0	0	0	Clear TXD0 output register to "0"
1	0	1	Clear TXD0 output register to "0"
2	1	0	Set TXD0 output register to "1"
3	1	1	Set TXD0 output register to "1"

For bit order control (DORD) on SPI serial transfer, CMT2380F17 provides a control bit, S0DOR, to control the data bit order by software program. S0DOR default is "1", LSB first.

Transmission is initiated by any instruction that uses S0BUF as a destination register. The "write to S0BUF" signal triggers the UART engine to start the transmission. The data in the S0BUF would be shifted into the RXD0 pin as MOSI serial data. The SPI shift clock is built on the TXD0 pin for SPICLK output. After eight raising edge of shift clocks passing, TI0 would be asserted by hardware to indicate the end of transmission. And the contents on the S0MI pin would be sampled and shifted into shift register. Then, "read S0BUF" can get the SPI shift-in data. Figure 18–13 shows the transmission waveform in Mode 0. RI0 will not be asserted in Mode 4.

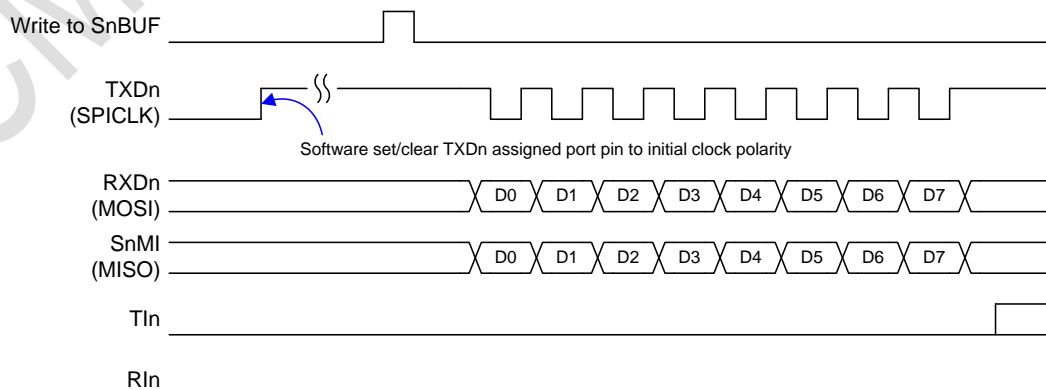


Figure 18-13. Serial Port 0 Mode 4 transmission waveform (n = 0)

18.9 Serial Port 0 Register

All the four operation modes of the serial port are the same as those of the standard 8051 except the baud rate setting. Three registers, PCON, AUXR2 and S0CFG, are related to the baud rate setting:

- **S0CON**: Serial port 0 Control Register

SFR Page = 0 only

SFR Address = 0x98

Bit	7	6	5	4	3	2	1	0
Name	SM00/FE	SM10	SM20	REN0	TB80	RB80	TI0	RI0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: FE, Framing Error bit. The SMOD0 bit must be set to enable access to the FE bit. 0: The FE bit is not cleared by valid frames but should be cleared by software.

1: This bit is set by the receiver when an invalid stop bit is detected.

Bit 7: Serial port 0 mode bit 0, (SMOD0 must = 0 to access bit SM00)

Bit 6: Serial port 0 mode bit 1.

SM30	SM00	SM10	Mode	Description	BaudRate
0	0	0	0	shiftregister	SYSClk/12or SYSClk/4
0	0	1	1	8-bitUART	variable
0	1	0	2	9-bitUART	SYSClk/64,/32,/16,/8or /192,/96,/48,/24
0	1	1	3	9-bitUART	variable
1	0	0	4	SPIMaster	SYSClk/12or SYSClk/4
1	0	1	5	Reserved	Reserved
1	1	0	6	Reserved	Reserved
1	1	1	7	Reserved	Reserved

Bit 5: Serial port 0 mode bit 2.

0: Disable SM20 function.

1: Enable the automatic address recognition feature in Modes 2 and 3. If SM20=1, RI0 will not be set unless the received 9th data bit is 1, indicating an address, and the received byte is a Given or Broadcast address. In mode1, if SM20=1 then RI0 will not be set unless a valid stop Bit was received, and the received byte is a Given or Broadcast address. In Mode 0, SM20 should be 0.

Bit 4: REN0, Enable serial reception.

0: Clear by software to disable reception.

1: Set by software to enable reception.

Bit 3: TB80, The 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired.

Bit 2: RB80, In Modes 2 and 3, the 9th data bit that was received. In Mode 1, if SM20 = 0, RB80 is the stop bit that was received. In Mode 0, RB80 is not used.

Bit 1: TI0. Transmit interrupt flag.

0: Must be cleared by software.

1: Set by hardware at the end of the 8th bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission.

Bit 0: RI0. Receive interrupt flag.

0: Must be cleared by software.

1: Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM20).

- **S0BUF:** Serial port 0 Buffer Register

SFR Page = 0 only

SFR Address = 0x99

Bit	7	6	5	4	3	2	1	0
Name	S0BUF[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	X	X	X	X	X	X	X	X

Bit 7~0: It is used as the buffer register in transmission and reception.

- **SADDR:** Slave Address Register

SFR Page = 0~F

SFR Address = 0xA9

Bit	7	6	5	4	3	2	1	0
Name	SADDR[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

- **SADEN:** Slave Address Mask Register (SMOD3 = 0)

SFR Page = 0~F

SFR Address = 0xB9

Bit	7	6	5	4	3	2	1	0
Name	SADEN[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

SADDR register is combined with SADEN register to form Given/Broadcast Address for automatic address recognition. In fact, SADEN functions as the “mask” register for SADDR register. The following is the example for it.

SADDR = 1100 0000

SADEN = 1111 1101

Given = 1100 00x0
The Given slave address will be checked except bit 1 is treated as “don’t care”

The Broadcast Address for each slave is created by taking the logical OR of SADDR and SADEN. Zero in this result is considered as “don’t care”. Upon reset, SADDR and SADEN are loaded with all 0s. This produces a Given Address of all “don’t care” and a Broadcast Address of all “don’t care”. This disables the automatic address detection feature.

- **PCON0:** Power Control Register 0

SFR Page = 0~F

SFR Address = 0x87 POR = 0001-0000, RESET = 0000-0000

Bit	7	6	5	4	3	2	1	0
Name	SMOD1	SMOD0	GF	POF	GF1	GF0	PD	IDL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: SMOD1, double Baud rate control bit. 0: Disable double Baud rate of the UART.

1: Enable double Baud rate of the UART in mode 1, 2, or 3.

Bit 6: SMOD0, Frame Error select. 0: S0CON.7 is SM0 function.

1: S0CON.7 is FE function. Note that FE will be set after a frame error regardless of the state of SMOD0.

- **S0CFG: Serial Port 0 Configuration Register**

SFR Page = 0 only

SFR Address = 0x9C RESET = 0000-1000

Bit	7	6	5	4	3	2	1	0
Name	URTS	SMOD2	URM0X3	SM30	S0DOR	BTI	UTIE	SMOD3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	1	0	0	0

Bit 7: URTS, UART0 Timer Selection.

0: Timer 1 or Timer 2 can be used as the Baud Rate Generator in Mode 1 and Mode 3.

1: Timer 1 overflow signal is replaced by the UART1 Baud Rate Timer overflow signal when Timer 1 is selected as the Baud Rate Generator in Mode1 or Mode 3 of the UART0. (Refer Section "18.7.4 Baud Rate in Mode 1 & 3".)

Bit 6: SMOD2, UART0 extra double baud rate selector. 0: Disable extra double baud rate for UART0.

1: Enable extra double baud rate for UART0.

Bit 5: URM0X3, this bit control the baud rate in S0 mode 0, mode 2 and mode 4. S1 in mode 0 and mode4:

0: Clear to select SYSCLK/12 as the baud rate for S0 Mode 0 and Mode 4. 1: Set to select SYSCLK/4 as the baud rate for S0 Mode 0 and Mode 4.

S0 in mode 2:

0: Clear to select UART0 baud rate as SYSCLK/32 or /64. 1: Set to select UART0 baud rate as SYSCLK/96 or /192.

Bit 4: SM30, Serial Port Mode control bit 3.

Bit 3: S0DOR, Serial Port 0 data order control in all operating modes. If S0 is not in Timer mode:

0: The MSB of the data byte is transmitted first.

1: The LSB of the data byte is transmitted first. S0DOR is set to "1" in default. If S0 is in Timer mode:

0: Set the S0BRG to 8-bit reload timer/counter mode. 1: Set the S0BRG to 16-bit timer/counter mode.

Bit 2: BTI, Block TI0 in Serial Port 0 Interrupt.

0: Retain the TI0 to be a source of Serial Port 0 Interrupt.

1: Block TI0 to be a source of Serial Port 0 Interrupt.

Bit 1: UTIE, S0 TI0 Enabled in system flag interrupt.

0: Disable the interrupt vector sharing for TI0 in system flag interrupt.

1: Set TI0 flag will share the interrupt vector with system flag interrupt.

Bit 0: SMOD3, S0CR1 access control.

0: Disable S0CR1 access. CPU accesses SFR address 0xB9 to read/write SADEN.

1: Enable S0CR1 access. CPU accesses SFR address 0xB9 to read/write S0CR1.

- **AUXR2: Auxiliary Register 2**

SFR Page = 0~F

SFR Address = 0xA3

Bit	7	6	5	4	3	2	1	0
Name	STAF	STOF	--	--	T1X12	T0X12	T1CKOE	T0CKOE
R/W	R/W	R/W	W	W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 3: T1X12, Timer 1 clock source selector while C/T=0.

0: Clear to select SYSCLK/12.

1: Set to select SYSCLK as the clock source. If set, the UART0 baud rate by Timer 1 in Mode 1 and Mode 3 is 12 times than standard 8051 function.。

- **AUXR3:** Auxiliary Register 3

SFR Page = 0 only

SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	T0PS1	T0PS0	BPOC1	BPOC0	S0PS0	TWIPS1	TWIPS0	T0XL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 3: S0PS0, Serial Port 0 pin Selection 0. (S0PS1 at AUXR10.3)

S0PS1, S0PS0	RxD0	TxD0
00	P3.0	P3.1
01	P4.4	P4.5
10	P3.1	P3.0
11	P1.7	P2.2

- **AUXR6:** Auxiliary Register 6

SFR Page = 3 only

SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	KBI4PS1	KBI4PS0	KBI6PS0	KBI2PS0	T3FCS	T2FCS	SnMIPS	S0COPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 1: SnMIPS, S0MI & S1MI Port pin Selection.

SnMIPS	S0MI	S1MI
0	P1.6	P6.1
1	P3.3	P4.7

18.10 Serial Port 0 Enhance function

If SMOD3 (S0CFG.0) is set, SFR address 0xB9 will be accessed on S0CR1. S0CR1 control the enhanced function of serial port 0 including :

- Enable S0 embedded baud rate generator, S0BRG
- Enable the S0 TX or RX to select the baud rate time base by S0BRG
- Enable S0BRG to behave a general timer
- Enable S0 to enter LIN bus mode

- **S0CR1:** Serial Port 0 Control Register 1 (SMOD3 = 1)

SFR Page = 0~F

SFR Address = 0xB9

Bit	7	6	5	4	3	2	1	0
Name	S0TR	S0TX12	S0TCK	S0RCK	S0CKOE	ARTE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	W	W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: S0TR, UART0 Baud Rate Generator control bit. 0: Clear to stop S0BRG operation.

1: Set to start S0BRG operation.

Bit 6: S0TX12, S0BRG clock source selection.

0: Clear to select SYSCLK/12 as the clock source of S0BRG. 1: Set to select SYSCLK as the clock source of S0BRG.

Bit 5: S0TCK, S0 control bit to select S0BRG overflow for UART0 transmit clock.

0: Cause Timer 1 or Timer 2 overflow to be used for the transmit clock.

1: Cause the S0 to use S0BRG overflow for it's transmit clock and operating mode control.

Bit 4: S0RCK, S0 control bit to select S0BRG overflow for UART0 receive clock.

0: Cause Timer 1 or Timer 2 overflow to be used for the receive clock.

1: Cause the S0 to use S0BRG overflow for it's receive clock and operating mode control.

Bit 3: S0CKOE, S0BRG clock output control.

0: Disable S0BRG clock output on S0CKO. 1: Enable S0BRG clock output on S0CKO.

Bit 2: ARTE, Auto Repeat Transmit Enable.

0: Disable auto repeat transmit.

1: Auto repeat transmit enable.

Bit 1~0: Reserved. Software must write "0" on these bits when S0CR1 is written.

- **S0BRT**: Serial port 0 Baud Rate Timer Reload Register

SFR Page = 0 only

SFR Address = 0x9A

Bit	7	6	5	4	3	2	1	0
Name	S0BRT[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~0: It is used as the reload value register for baud rate timer generator that works in a similar manner as Timer 1.

- **S0BRC**: Serial port 0 Baud Rate Counter Register

SFR Page = 0 only

SFR Address = 0x9B

Bit	7	6	5	4	3	2	1	0
Name	S0BRC[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~0: It is used as the reload value register for baud rate timer generator that works in a similar manner as Timer 1. This register can be always read/written by software. If S0TR (S0CR1.7) = 0, software writing S0BRT will store the data content to S0BRT and S0BRC concurrently. If S0TR = 1, software writing S0BRT will not store the data to S0BRC.

18.10.1 S0 Baud Rate Generator (S0BRG)

The CMT2380F17 has an embedded Baud Rate Generator to generate the clock for serial port 0 operation. It is constructed by an 8-bit up-counter, S0BRC, and an 8-bit reload register, S0BRT. The overflow (S0TOF) of S0BRC is the time base of UART0 serial engine in all operation modes and triggers the S0BRT content reloaded into S0BRC for the consecutive counting.

If S0TR = 0, software writing S0BRT will modify S0BRC simultaneously. After S0TR enabled to start the S0BRC counting, it is no influence on S0BRC when S0BRT is writing. Modifying S0BRC is always independent with S0BRT content.

This baud rate generator can also provide the time base for clock output, S0CKO, from the S0BRC overflow rate by 2 (S0TOF/2). S0TOF also supplies the toggle source for other peripherals' clock input. Regardless S0 engine is running or pending, S0BRG always serves the time base function for these peripherals.

The configuration of the Serial Port 0 Baud Rate Generator is shown in Figure 18–14.

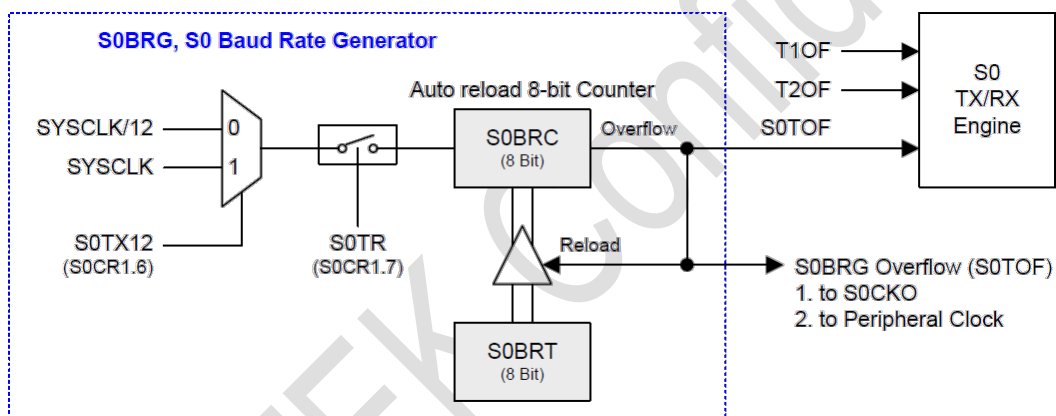


Figure 18-14. S0BRG configuration

18.10.2 Independent Baud Rate Generator S0BRG for S0

To give S0 more flexibility, S0 Baud Rate Generator S0BRG can be selected as Baud Rate source. The configuration of the Serial Port 0 baud rate selection is shown in Figure 18–15.

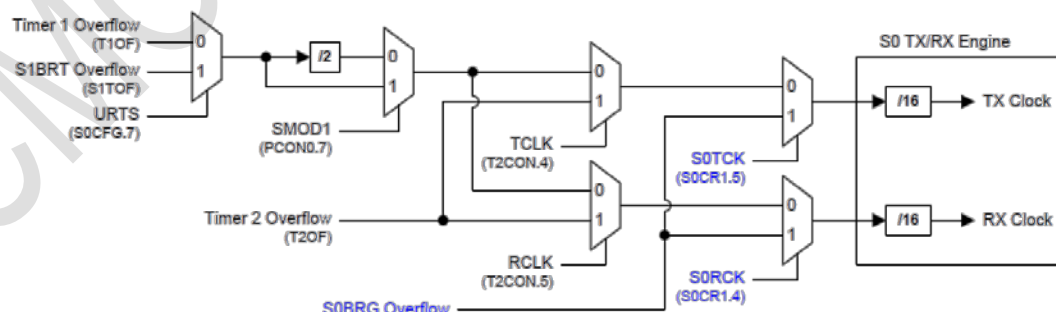


Figure 18-15. S0 Baud Rate Selection

18.10.3 S0 Enhanced Mode

SM30,SM00,SM10	S0RCK	S0TCK	Function	BaudRateTimeBase	Note
000	0	0	shiftregister	SYSCCLK/12 orSYSCCLK/4(URM0X3=1)	
001	0	0	8-bitUART	Timer1orTimer2overflow	WhenSMOD1&SMOD2=1, counter cannot be Full-1 orFull-2(Ex.254,255,65534,65535)
010	0	0	9-bitUART	SYSCCLK/64,/32,/16,or/8	
011	0	0	9-bitUART	Timer1orTimer2overflow	WhenSMOD1&SMOD2=1, counter cannot be Full-1 orFull-2(Ex.254,255,65534,65535)
100	0	0	SPI Master	SYSCCLK/12 orSYSCCLK/4(URM0X3=1)	
000	0	1	shiftregister	S0BRGoverflow	S0BRTcannotbe255
001	0/1	0/1	8-bitUART	Selectable S0BRG overflowonTXor RX	SMOD1 & SMOD2 cannot be 1atthesametime
010	0	1	9-bitUART	TX:S0BRGoverflow RX:SYSCCLK/64,/32or /16	SMOD1 & SMOD2 cannot be 1atthesametime
010	1	0	9-bitUART	TX:SYSCCLK/64,/32or/16 RX:S0BRGoverflow	SMOD1 & SMOD2 cannot be 1atthesametime
010	1	1	Pure Timer	OnlyTimerfunction	
011	0/1	0/1	9-bitUART	Selectable S0BRG overflowonTXor RX	SMOD1 & SMOD2 cannot be 1atthesametime
100	0	1	SPI Master	S0BRGoverflow	S0BRTcannotbe255
101	1	1	LINBus	S0BRG overflow and autobaud rate	SMOD1 & SMOD2 cannot be 1atthesametime
Others			Reserved.		

18.10.4 S0 LIN Bus Register

- S0CFG1: Serial Port 0 Configuration Register 1

SFR Page = 0 only

SFR Address = 0x9D

Bit	7	6	5	4	3	2	1	0
Name	SBF0	TXER0	S0SB16	ATBR0	TXRX0	SYNC0		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	W	W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: SBF0, Sync-Break Flag on S0. 0: Write 1 to clear by software.

1: Set by hardware at the end of the break event detection on LIN bus. In master mode, it is set combined with TI0 flag. In slave mode, it is set combined with RI0.

Bit 6: TXER0, LIN Transmit Error on S0. 0: Write 1 to clear by software.

1: In TX mode, set by hardware at the transmit error detection on LIN bus.

Bit 5: S0SB16, Sync-Break 16 Bit enable on S0.

0: Select 13-bit Sync-Break transmitting in master mode. 1: Select 16-bit Sync-Break transmitting in

Bit 1~0: Reserved. Software must write "0" on these bits when S0CFG1 is written.

8-bit Timer Mode: {SM30, SM00, SM10} = 010, {S0TCK, S0RCK} = 11, S0DOR = 1



16-bit Timer Mode: {SM30, SM00, SM10} = 010, {S0TCK, S0RCK} = 11, S0DOR = 0



Figure 18–17.S0 16-bitTimer Mode

18.10.7 S0BRG Programmable Clock Output

S0BRG has a clock output mode is shown in Figure 18–18.

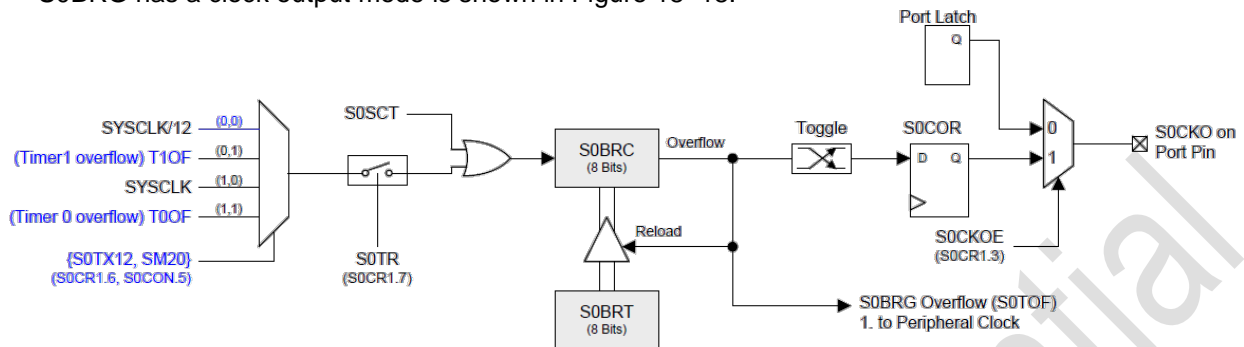


Figure 18–18. S0BRG Clock Output (S0BRG in 8-bit Timer Mode)

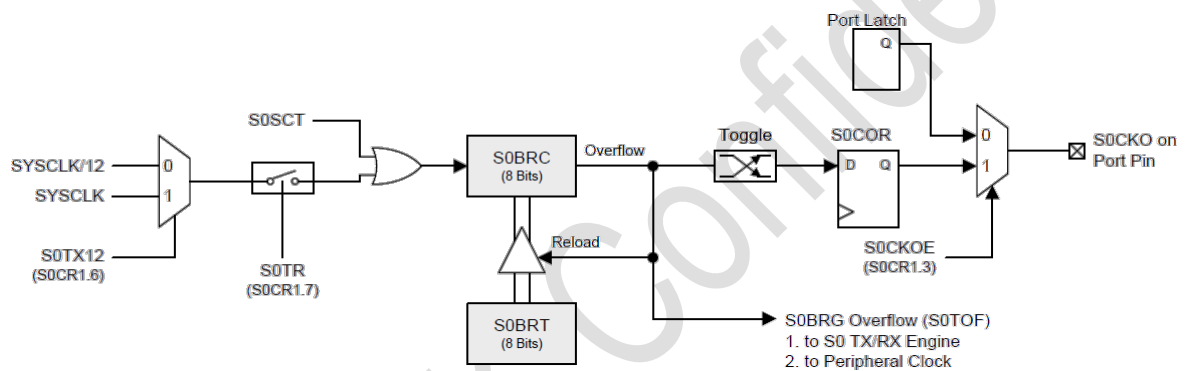


Figure 18–19. S0BRG Clock Output (S0BRG for UART Mode)

- **AUXR6:** Auxiliary Register 6
SFR Page = 3 only
SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	KBI4PS1	KBI4PS0	KBI6PS0	KBI2PS0	T3FCS	T2FCS	SnMIPS	S0COPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 0: S0COPS, S0BRG Clock Output (S0CKO) port pin Selection.

S0COPS	S0CKO
0	P4.7
1	P3.3

19 Serial Port 1 (UART1)

The CMT2380F17 is equipped with a secondary UART (hereafter, called UART1), which has 5 operation modes, Mode 0 ~ Mode 4, the same as the first UART (UART0) except the following differences:

- (1) The UART1 has no enhanced functions: Framing Error Detection and Auto Address Recognition.
- (2) The UART1 use the dedicated Baud Rate Timer as its Baud Rate Generator (S1BRG).
- (3) The UART1 uses TXD1 and RXD1 for transmit and receive, respectively.
- (4) The Baud Rate Generator provides the toggle source for S1CKO and peripheral clock.
- (5) S1 + S1BRG can be configured to an 8-bit auto-reload timer with port change detection.
- (6) In mode 0 and mode 4, S1TX12 of UART1 is the same function as URM0X3 in UART0.

The UART1 and UART0 in CMT2380F17 can operate simultaneously in identical or different modes and communication speeds.

19.1 Serial Port 1 Baud Rate Generator (S1BRG)

The CMT2380F17 has an embedded Baud Rate Generator to generate the UART clock for serial port 1 operation in mode 1 and mode 3. It is constructed by an 8-bit up-counter, S1BRC, and an 8-bit reload register, S1BRT. The overflow (S1TOF) of S1BRC is the time base of UART1 serial engine in mode 1 and mode 3 and triggers the S1BRT content reloaded into S1BRC for the consecutive counting.

If S1TR = 0, software writing S1BRT will modify S1BRC simultaneously. After S1TR enabled to start the S1BRC counting, it is no influence on S1BRC when S1BRT is writing. Modifying S1BRC is always independent with S1BRT content.

This baud rate generator can also provide the time base for serial port 0 by software configured. There is an addition clock output, S1CKO, from the S1BRC overflow rate by 2 (S1TOF/2). S1TOF also supplies the toggle source for other peripherals' clock input. Regardless S1 engine is running or pending, S1BRG always serves the time base function for these peripherals.

The configuration of the Serial Port 1 Baud Rate Generator is shown in Figure 19–1.

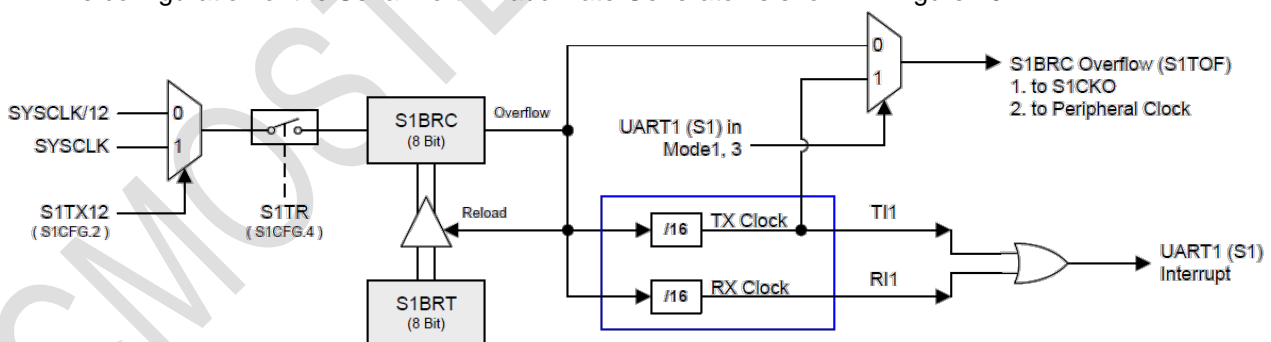


Figure 19–1. S1BRGStructure(S1TME=0)

19.2 S1BRG configuration (S1TME=0)

19.2.1 Baud Rate in Mode 0

$\text{S1 Mode 0 Baud Rate} = \frac{F_{\text{SYSCLK}}}{n}$; n=12, if S1TX12=0 ; n=4, if S1TX12=1
--	---

19.2.2 Baud Rate in Mode 2

When S1M0X3 = 0,

$$\text{S1 Mode 2 Baud Rate} = \frac{2^{\text{S1MOD1}}}{64} \times \text{F}_{\text{SYSCLK}}$$

When S1M0X3 = 1,

$$\text{S1 Mode 2 Baud Rate} = \frac{2^{\text{S1MOD1}}}{192} \times \text{F}_{\text{SYSCLK}}$$

Table 19–1. S1 Mode 2 Baud Rates @ F_{SYSCLK}=11.0592MHz

BaudRate	S1M0X3	S1MOD1	Error
172,800	0	0	±0.0%
345,600	0	1	±0.0%
57,600	1	0	±0.0%
115,200	1	1	±0.0%

Table 19–2. S1 Mode 2 Baud Rates @ F_{SYSCLK}=12.00MHz

BaudRate	S1M0X3	S1MOD1	Error
187,500	0	0	±0.0%
375,000	0	1	±0.0%
62,500	1	0	±0.0%
125,000	1	1	±0.0%

19.2.3 Baud Rate in Mode 1 & 3

$$\text{S1 Mode 1, 3 Baud Rate} = \frac{2^{\text{S1MOD1}}}{32} \times \frac{\text{F}_{\text{SYSCLK}}}{12 \times (256 - \text{S1BRT})} ; \text{S1TX12}=0$$

$$\text{or} = \frac{2^{\text{S1MOD1}}}{32} \times \frac{\text{F}_{\text{SYSCLK}}}{1 \times (256 - \text{S1BRT})} ; \text{S1TX12}=1$$

Table 19–3 ~ Table 19–6 list various commonly used baud rates and how they can be obtained from S1BRG, serial port 1 baud rate generator.

Table 19–3. S1BRG Generated Commonly Used Baud Rates @ F_{SYSCLK}=11.0592MHz

Baud Rate	S1BRT, Reload Value of S1BRG					
	S1TX12 = 0			S1TX12 = 1		
	S1MOD1=0	S1MOD1=1	Error	S1MOD1=0	S1MOD1=1	Error
1200	232	208	0.0%	--	--	--
2400	244	232	0.0%	112	--	0.0%
4800	250	244	0.0%	184	112	0.0%
9600	253	250	0.0%	220	184	0.0%
14400	254	252	0.0%	232	208	0.0%
19200	--	253	0.0%	238	220	0.0%
28800	255	254	0.0%	244	232	0.0%
38400	--	--	--	247	238	0.0%

57600	--	255	0.0%	250	244	0.0%
115200	--	--	--	253	250	0.0%
230400	--	--	--	--	253	0.0%

Table 19–4.S1BRG Generated Commonly Used Baud Rates @ FSYSCLK=22.1184MHz

Baud Rate	S1BRT, Reload Value of S1BRG					
	S1TX12 = 0			S1TX12 = 1		
	S1MOD1=0	S1MOD1=1	Error	S1MOD1=0	S1MOD1=1	Error
1200	208	160	0.0%	--	--	--
2400	232	208	0.0%	--	--	0.0%
4800	244	232	0.0%	112	--	0.0%
9600	250	244	0.0%	184	112	0.0%
14400	252	248	0.0%	208	160	0.0%
19200	253	250	0.0%	220	184	0.0%
28800	254	252	0.0%	232	208	0.0%
38400	--	253	0.0%	238	220	0.0%
57600	255	254	0.0%	244	232	0.0%
115200	--	255	0.0%	250	244	0.0%
230400	--	--	--	253	250	0.0%
460800	--	--	--	--	253	0.0%

Table 19–5.S1BRG Generated Commonly Used Baud Rates @ FSYSCLK=12.0MHz

Baud Rate	S1BRT, Reload Value of S1BRG					
	S1TX12 = 0			S1TX12 = 1		
	S1MOD=0	S1MOD=1	Error	S1MOD=0	S1MOD=1	Error
1200	230	204	0.16%	--	--	--
2400	243	230	0.16%	100	--	0.16%
4800	--	243	0.16%	178	100	0.16%
9600	--	--	--	217	178	0.16%
14400	--	--	--	230	204	0.16%
19200	--	--	--	--	217	0.16%
28800	--	--	--	243	230	0.16%
38400	--	--	--	246	236	2.34%
57600	--	--	--	--	243	0.16%
115200	--	--	--	--	--	--

Table 19–6.S1BRG Generated Commonly Used Baud Rates @ FSYSCLK=24.0MHz

Baud Rate	S1BRT, Reload Value of S1BRG					
	S1TX12 = 0			S1TX12 = 1		
	S1MOD=0	S1MOD=1	Error	S1MOD=0	S1MOD=1	Error
1200	204	152	0.16%	--	--	--
2400	230	204	0.16%	--	--	--
4800	243	230	0.16%	100	--	0.16%

9600	--	243	0.16%	178	100	0.16%
14400	--	--	--	204	152	0.16%
19200	--	--	--	217	178	0.16%
28800	--	--	--	230	204	0.16%
38400	--	--	--	--	217	0.16%
57600	--	--	--	243	230	0.16%
115200	--	--	--	--	243	0.16%

Table 19–7.S1BRG Generated Commonly Used Baud Rates @ FSYSCLK=29.4912MHz

Baud Rate	S1BRT, Reload Value of S1BRG					
	S1TX12 = 0			S1TX12 = 1		
	S1MOD1=0	S1MOD1=1	Error	S1MOD1=0	S1MOD1=1	Error
1200	192	128	0.0%	--	--	--
2400	224	192	0.0%	--	--	--
4800	240	224	0.0%	64	--	0.0%
9600	248	240	0.0%	160	64	0.0%
14400	--	--	--	192	128	0.0%
19200	252	248	0.0%	208	160	0.0%
28800	--	--	--	224	192	0.0%
38400	--	--	--	232	208	0.0%
57600	--	--	--	240	224	0.0%
115200	--	--	--	248	240	0.0%
230.4K	--	--	--	252	248	0.0%
460.8K	--	--	--	254	252	0.0%
921.6K	--	--	--	255	254	0.0%
1.8432M	--	--	--	--	255	0.0%

Table 19–8.S1BRG Generated Commonly Used Baud Rates @ FSYSCLK=44.2368MHz

Baud Rate	S1BRT, Reload Value of S1BRG					
	S1TX12 = 0			S1TX12 = 1		
	S1MOD1=0	S1MOD1=1	Error	S1MOD1=0	S1MOD1=1	Error
1200	160	64	0.0%	--	--	--
2400	208	160	0.0%	--	--	--
4800	232	208	0.0%	--	--	--
9600	244	232	0.0%	112	--	0.0%
14400	248	240	0.0%	160	64	0.0%
19200	250	244	0.0%	184	112	0.0%
28800	252	248	0.0%	208	160	0.0%
38400	253	250	0.0%	220	184	0.0%
57600	254	252	0.0%	232	208	0.0%
115200	255	254	0.0%	244	232	0.0%
230.4K	--	255	0.0%	250	244	0.0%
460.8K	--	--	--	253	250	0.0%
921.6K	--	--	--	--	253	0.0%

1.8432M	--	--	--	--	--	--
2.7648M	--	--	--	--	255	0.0%

Table 19–9.S1BRG Generated Commonly Used Baud Rates @ FSYCLK=32.0MHz

Baud Rate	S1BRT, Reload Value of S1BRG					
	S1TX12 = 0			S1TX12 = 1		
	S1MOD1=0	S1MOD1=1	Error	S1MOD1=0	S1MOD1=1	Error
1200	187	118	0.64%	--	--	--
2400	221	186	-0.79%	--	--	--
4800	239	222	2.12%	48	--	0.16%
9600	--	239	2.12%	152	48	0.16%
14400	--	--	--	187	118	0.64%
19200	--	--	--	204	152	0.16%
28800	--	--	--	221	186	-0.79%
38400	--	--	--	230	204	0.16%
57600	--	--	--	239	222	2.12%
115200	--	--	--	--	239	2.12%

Table 19–10.S1BRG Generated Commonly Used Baud Rates @ FSYCLK=48.0MHz

Baud Rate	S1BRT, Reload Value of S1BRG					
	S1TX12 = 0			S1TX12 = 1		
	S1MOD1=0	S1MOD1=1	Error	S1MOD1=0	S1MOD1=1	Error
1200	152	48	0.16%	--	--	--
2400	204	152	0.16%	--	--	--
4800	230	204	0.16%	--	--	--
9600	243	230	0.16%	100	--	0.16%
14400	--	239	2.12%	152	48	0.16%
19200	--	243	0.16%	178	100	0.16%
28800	--	--	--	204	152	0.16%
38400	--	--	--	217	178	0.16%
57600	--	--	--	230	204	0.16%
115200	--	--	--	243	230	0.16%
230.4K	--	--	--	--	243	0.16%

19. 3Serial Port 1 Mode 4 (SPI Master)

The Serial Port of CMT2380F17 is embedded Mode 4 to support SPI master engine. The Mode 4 is selected by SM31, SM01 and SM11. Table 19–11 shows the serial port mode definition in CMT2380F17..

Table 19-11. Serial Port 1 Mode Selection

SM31	SM01	SM11	Mode	Description	BaudRate
0	0	0	0	shiftregister	SYSCCLK/12or SYSCCLK/4

0	0	1	1	8-bitUART	variable
0	1	0	2	9-bitUART	SYSClk/64,/32or/192,/96
0	1	1	3	9-bitUART	variable
1	0	0	4	SPI Master	SYSClk/12or SYSClk/4
1	0	1	5	Reserved	variable
1	1	0	6	Reserved	Reserved
1	1	1	7	Reserved	variable

S1M0X3 also controls the SPI transfer speed. If S1M0X3 = 1, the SPI clock frequency is SYSClk/4. Otherwise, the SPI clock frequency is SYSClk/12.

The SPI master in CMT2380F17 uses the TXD1 as SPICLK, RXD1 as MOSI, and S1MI as MISO. nSS is selected by MCU software on other port pin. Figure 19–2 shows the SPI connection. It also can support the configuration for multiple slaves communication in Figure 19–3.

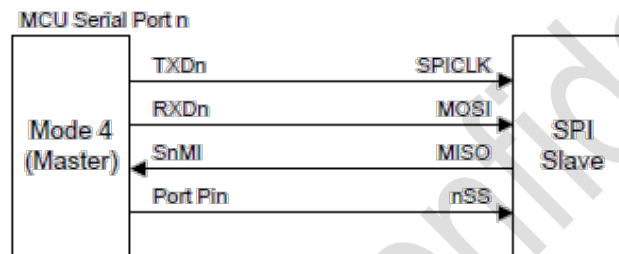


Figure 19–2. Serial Port 1 Mode 4, Single Master and Single Slave configuration (n = 1)

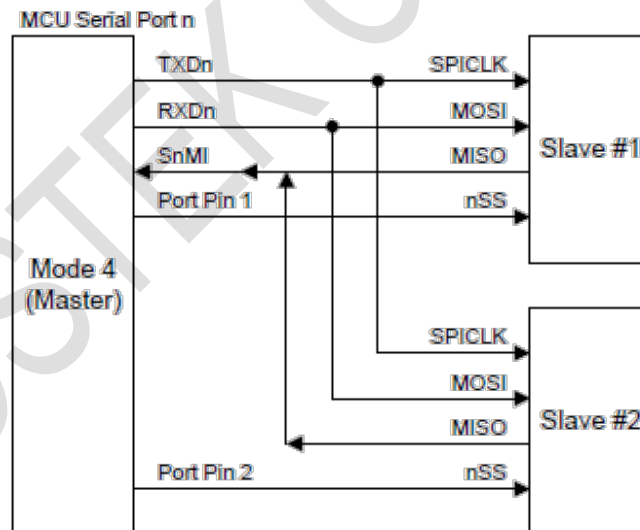


Figure 19–3. Serial Port 1 Mode 4, Single Master and Multiple Slaves configuration (n = 1)

The SPI master satisfies the transfer with the full function SPI module of Megawin MG82/84 series MCU with CPOL, CPHA and DORD selection. For CPOL and CPHA condition, CMT2380F17 uses an easy way by initialize SPI clock polarity to fit them. Table 18–12 shows the serial port Mode 4 mapping with the four SPI operating mode.

Table 19–12. SPI mode mapping with Serial Port Mode 4 configuration

SPI Mode	CPOL	CPHA	Configuration in TXD1
0	0	0	Clear TXD1 output register to "0"

1	0	1	Clear TXD1 output register to "0"
2	1	0	Set TXD1 output register to "1"
3	1	1	Set TXD1 output register to "1"

For bit order control (DORD) on SPI serial transfer, CMT2380F17 provides a control bit, S1DOR, to control the data bit order by software program. The default value of S1DOR is "1", LSB first.

Transmission is initiated by any instruction that uses S1BUF as a destination register. The "write to S1BUF" signal triggers the UART engine to start the transmission. The data in the S1BUF would be shifted into the RXD1 pin as MOSI serial data. The SPI shift clock is built on the TXD1 pin for SPICLK output. After eight raising edge of shift clocks passing, TI1 would be asserted by hardware to indicate the end of transmission. And the contents on the S1MI pin would be sampled and shifted into shift register. Then, "read S1BUF" can get the SPI shift-in data. Figure 19–4 shows the transmission waveform in Mode 0. RI1 will not be asserted in Mode 4.

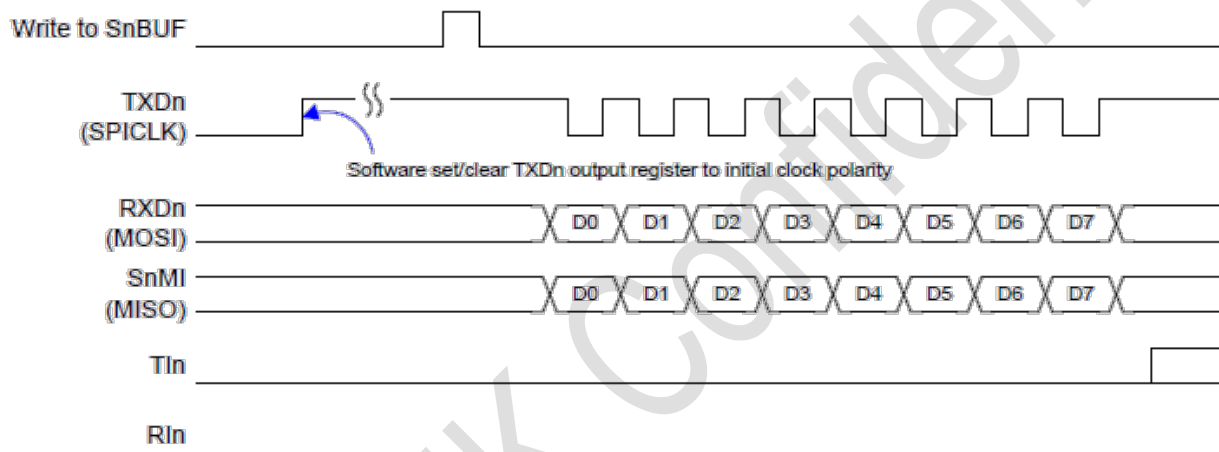


Figure 19–4. Serial Port 1 Mode 4 transmission waveform (n =1)

19.4 8-Bit Timer Mode on S1BRG

If the UART1 is not necessary in application or pending by software, setting S1TME=1 in the CMT2380F17 provides the pure timer operating mode on S1 Baud Rate Generator (S1BRG). This timer operates as an 8-bit auto-reload timer and provides the overflow flag which is set on the TI1 (S1CON.1). The RI1 (S1CON.0) serves the port change detector on RXD1 port pin. Both of TI1 and RI1 in this mode keep the interrupt capability on UART1 interrupt resource and have the individual interrupt enabled control (TB81 & REN1). RB81 selects the RI1 detection level on RXD1 port input. If RB81=0, RI1 will be set by REN1=1 and RXD1 pin falling edge detecting. Otherwise, RI1 will detect the rising edge on RXD1 port pin. In MCU power-down mode, the RI1 is forced to level-sensitive operation and has the capability to wake up CPU if UART1 interrupt is enabled.

This pure timer mode has a clock input option from Timer 1 overflow which is a cascaded counter to perform a 16-bit timer. When S1BRC overflows, it can be the clock source of UART0 or toggle the port pin output. "S1CKOE=1" enables the S1CKO output on port pin and masks the RI1 interrupt.

The configuration of the Pure Timer mode of S1BRG is shown in Figure 19–5.

8-bit Timer Mode: S1TME = 1, S1DOR = 1

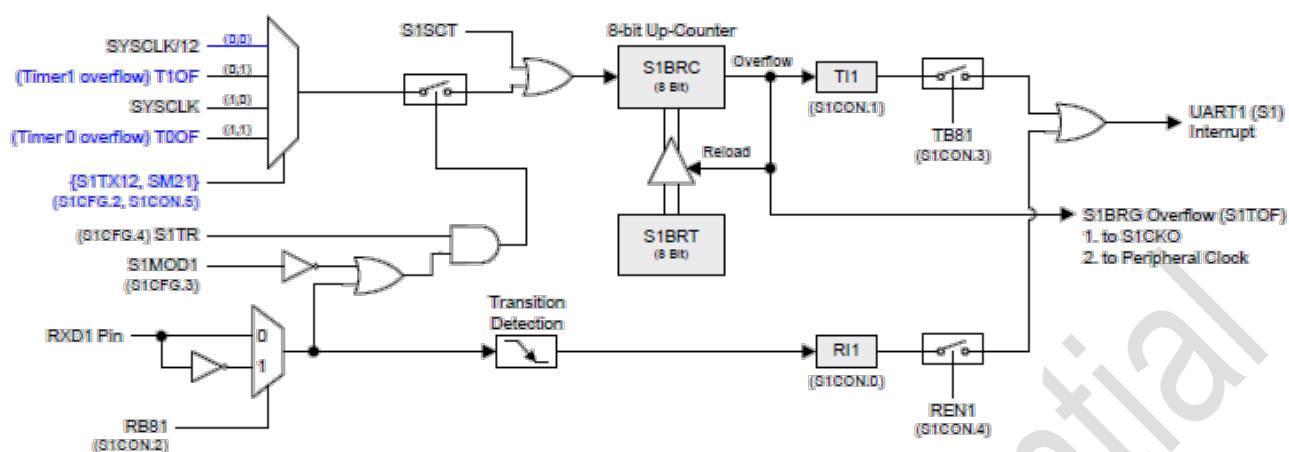


Figure 19–5. 8-bit Timer Mode Configuration for S1BRG (S1TME=1)

19.5 16-Bit Timer Mode on S1BRG

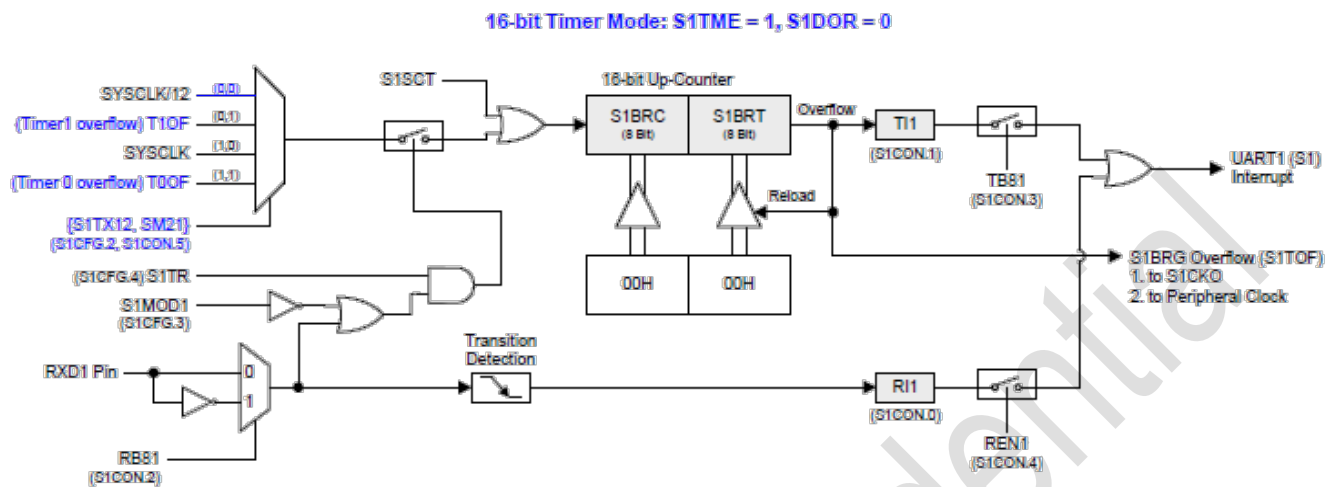


Figure 19–6. 16-Bit Timer Mode on S1BRG(S1TME=1)

19.6 S1BRT Programmable Clock Output

When S1BRC overflows, the overflow flag, S1TOF, provides the toggle source for S1CKO and peripheral clock. The input clock (SYSCLK/12 or SYSCLK) increases the 8-bit timer, S1BRC. The timer repeatedly counts to overflow from a loaded value. Once overflows occur, the content of S1BRT is loaded into S1BRC for the consecutive counting. Figure 19–7 shows the block diagram for the Clock Output mode of S1 Baud Rate Generator. The following formula gives the clock-out frequency.

$$\text{S1T Clock-out Frequency} = \frac{\text{SYSCLK Frequency}}{n \times (256 - \text{S1BRT})}$$

; n=24, if S1TX12=0
; n=2, if S1TX12=1

Note:

- (1) For SYSCLK=12MHz & S1TX12=0, S1BRG has a programmable output frequency range from 1.95KHz to 500KHz.
- (2) For SYSCLK=12MHz & S1TX12=1, S1BRG has a programmable output frequency range from 23.43KHz to 6MHz.

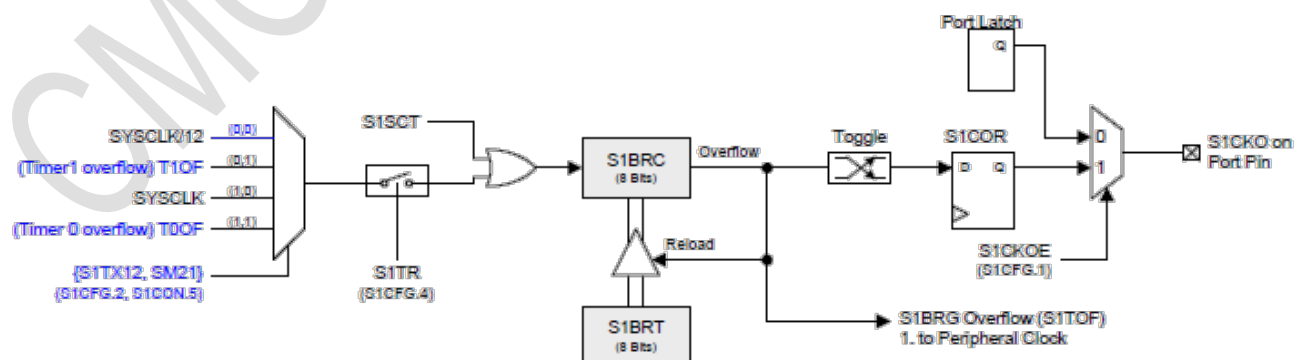


Figure 19–7. S1BRG Clock Output(S1BRG in 8-bit Timer Mode)

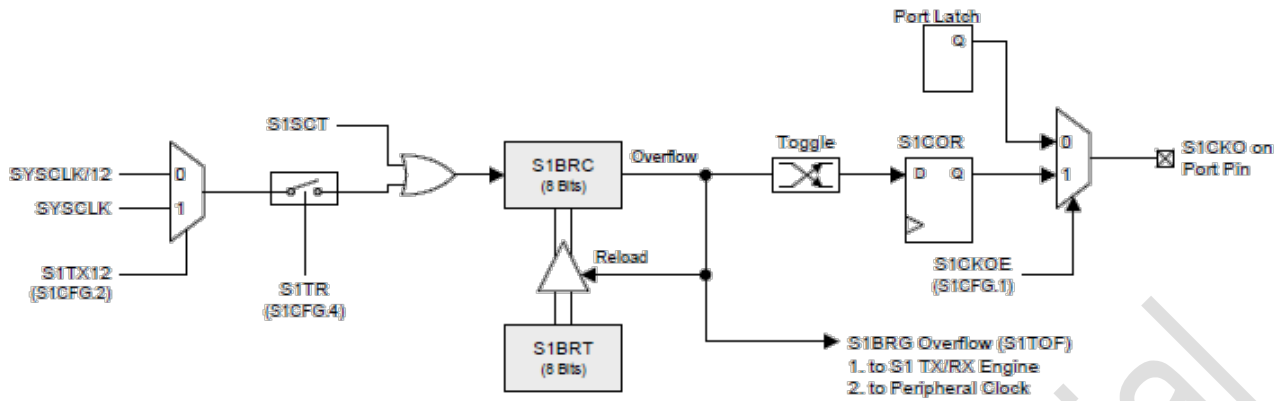


Figure 19–8. S1BRG Clock Output (S1BRG for UART Mode)

How to Program 8-bit S1BRG in Clock-out Mode

- Select S1CFG.S1TX12 bit and S1CON.SM21 bit to decide the S1BRG clock source.
- Determine the 8-bit reload value from the formula and enter it in the S1BRT and S1BRC registers.
- Set S1CKOE bit in S1CFG register.
- Set S1TR to start the S1BRC timer.

19.7 S1 Baud Rate Generator for S0

In the Mode 1 and Mode 3 operation of the UART0, the software can select Timer 1 as the Baud Rate Generator by clearing bits TCLK and RCLK in T2CON register. At this time, if URTS bit (S0CFG.7) is set, then Timer 1 overflow signal will be replaced by the overflow signal of the UART1 Baud Rate Generator (S1BRG). In other words, the user can adopt S1BRG as the Baud Rate Generator for Mode 1 or Mode 3 of the UART0 as long as RCLK=0, TCLK=0 and URTS=1. In this condition, Timer 1 is free for other application. Of course, if UART1 (Mode 1 or Mode 3) is also operated at this time, these two UARTs will have the same baud rates.

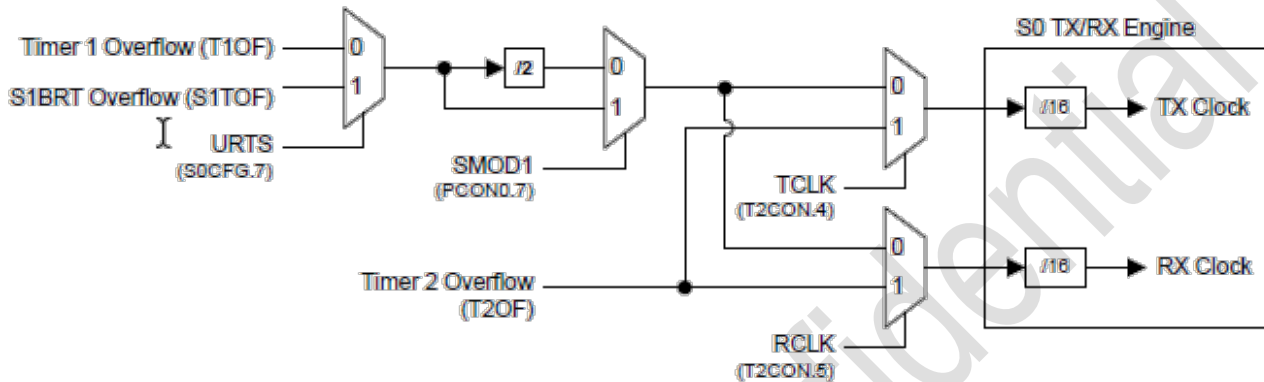


Figure 19-9. Additional Baud Rate Source for the UART0

When S1BRG is used as the baud rate generator of S0, the baud rate is as follows.

$$\text{Mode 1, 3 Baud Rate} = \frac{2^{SMOD1} \times 2^{(SMOD2 \times 2)}}{32} \times \frac{F_{SYSCLK}}{12 \times (256 - S1BRT)} ; S1TX12=0$$

$$\text{or} = \frac{2^{SMOD1} \times 2^{(SMOD2 \times 2)}}{32} \times \frac{F_{SYSCLK}}{1 \times (256 - S1BRT)} ; S1TX12=1$$

19.8 Serial Port 1 Register

The following special function registers are related to the operation of the UART1:

- **S1CON:** Serial port 1 Control Register

SFR Page = 1 only

SFR Address = 0x98

Bit	7	6	5	4	3	2	1	0
Name	SM01	SM11	SM21	REN1	TB81	RB81	TI1	RI1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: SM01, Serial port 1 mode bit 0.

Bit 6: SM11, Serial port 1 mode bit 1.

SM31	SM01	SM11	Mode	Description	BaudRate
0	0	0	0	shift register	SYSCLK/12 or SYSCLK/4
0	0	1	1	8-bit UART	variable
0	1	0	2	9-bit UART	SYSCLK/64, /32 or /192, /96
0	1	1	3	9-bit UART	variable

1	0	0	4	SPIMaster	SYSCLK/12or SYSCLK/4
1	0	1	5	Reserved	Reserved
1	1	0	6	Reserved	Reserved
1	1	1	7	Reserved	Reserved

Bit 5: Serial port 1 mode bit 2. 0: Disable SM21 function.

1: Enable the automatic address recognition feature in Modes 2 and 3. If SM21=1, RI1 will not be set unless the received 9th data bit is 1, indicating an address, and the received byte is a given or Broadcast address. In mode1, if SM21=1 then RI1 will not be set unless a valid stop Bit was received, and the received byte is a given or Broadcast address. In Mode 0, SM21 should be 0.

Bit 4: REN1, Enable serial reception.

0: Clear by software to disable reception. 1: Set by software to enable reception.

Bit 3: TB81, The 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired.

Bit 2: RB81, In Modes 2 and 3, the 9th data bit that was received. In Mode 1, if SM21 = 0, RB81 is the stop bit that was received. In Mode 0, RB81 is not used.

Bit 1: TI1. Transmit interrupt flag. 0: Must be cleared by software.

1: Set by hardware at the end of the 8th bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission.

Bit 0: RI1. Receive interrupt flag. 0: Must be cleared by software.

1: Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM21).

- S1BUF: Serial port 1 Buffer Register

SFR Page = 1 only

SFR Address = 0x99

Bit	7	6	5	4	3	2	1	0
Name	S1BUF[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	X	X	X	X	X	X	X	X

Bit 7~0: It is used as the buffer register in transmission and reception.

- S1BRT: Serial port 1 Baud Rate Timer Reload Register

SFR Page = 1 only

SFR Address = 0x9A

Bit	7	6	5	4	3	2	1	0
Name	S0BRT[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~0: It is used as the reload value register for baud rate timer generator that works in a similar manner as Timer 1.

- S1BRC: Serial port 1 Baud Rate Timer Reload Register

SFR Page = 1 only

SFR Address = 0x9A

Bit	7	6	5	4	3	2	1	0
Name	S0BRC[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~0: It is used as the reload value register for baud rate timer generator that works in a similar manner as Timer 1. This register can be always read/written by software. If S1CFG.S1TME = 0, software writing S1BRT will store the data content to S1BRT and S1BRC concurrently.

- S1CFG: Serial Port 1 Configuration Register

SFR Page = 1 only

SFR Address = 0x9C,

Bit	7	6	5	4	3	2	1	0
Name	SM31	S1M0X3	S1DOR	S1TR	S1MOD1	S1TX12	S1CKOE	S1TME
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	1	0	0	0	0	0

Bit 7: SM31, Serial Port 1 Mode control bit 3.

SM31	SM01	SM11	Mode	Description	BaudRate
0	0	0	0	shiftregister	SYSClk/12orSYSCl/4
0	0	1	1	8-bitUART	variable
0	1	0	2	9-bitUART	SYSClk/64,/32or/192,/96
0	1	1	3	9-bitUART	Variable
1	0	0	4	SPIMaster	SYSClk/12or SYSClK/4
1	0	1	5	Reserved	Reserved
1	1	0	6	Reserved	Reserved
1	1	1	7	Reserved	Reserved

Bit 6: S1M0X3, this bit control the baud rate in S1 mode 0, mode 2 and mode 4. S1 in mode 0 and mode4:

0: Clear to select SYSClk/12 as the baud rate for S1 Mode 0 and Mode 4. 1: Set to select SYSClk/4 as

the baud rate for S1 Mode 0 and Mode 4.

S1 in mode 2:

0: Clear to select UART1 baud rate as SYSCLK/32 or /64. 1: Set to select UART1 baud rate as SYSCLK/96 or /192.

Bit 5: S1DOR, Serial Port 1 data order control in all operating modes. If S1TME = 0:

0: The MSB of the data byte is transmitted first.

1: The LSB of the data byte is transmitted first. S1DOR is set to "1" in default.

If S1TME = 1:

0: Set the S1BRG to 8-bit reload timer/counter mode. 1: Set the S1BRG to 16-bit timer/counter mode.

Bit 4: S1TR, UART1 Baud Rate Generator control bit. 0: Clear to turn off the S1BRG.

1: Set to turn on S1BRG.

Bit 3: S1MOD1, UART1 double baud rate enable bit. 0: Disable the double baud rate function for UART1.

1: Enable the double baud rate function for UART1.

Bit 2: S1TX12, UART1 Baud Rate Generator clock source select 0: Clear to select SYSCLK/12 as the clock source for S1BRG.

1: Set to select SYSCLK as the clock source for S1BRG.

Bit 1: S1CKOE, Serial Port 1 BRG Clock Output Enable. 0: Disable the S1CKO output on the port pin.

1: Enable the S1CKO output on the port pin.

Bit 0: S1TME, Serial port 1 BRG Timer Mode Enabled. 0: Keep S1BRT to service Serial Port 1 (UART1).

1: Disable Serial Port 1 function and release the S1BRT as an 8-bit auto-reload timer. In this mode, there is an additional function for RXD1 port pin change detector.

AUXR9: Auxiliary Register 9

SFR Page = 6 only

SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	---	---	T1G1	T0G1	C0FDC1	C0FDC0	S1PS1	S1PS0
R/W	W	W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 1~0: S1PS1~0, Serial Port 1 pin Selection [1:0].

S1PS1, S1PS0	RxD1	TxD1
00	P1.0	P1.1
01	P6.0	P6.1
10	P4.4	P4.5
11	P3.4	P3.5

AUXR6: Auxiliary Register 6

SFR Page = 3 only

SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	KBI4PS1	KBI4PS0	KBI6PS0	KBI2PS0	T3FCS	T2FCS	SnMIPS	S0COPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 1: SnMIPS, S0MI & S1MI Port pin Selection.

SnMIPS	S0MI	S1MI
0	P1.6	P6.1

SnMIPS	S0MI	S1MI
1	P3.3	P4.7

CMOSTEK Confidential

20 Serial Peripheral Interface (SPI)

The CMT2380F17 provides a high-speed serial communication interface, the SPI interface. SPI is a full-duplex,

high-speed and synchronous communication bus with two operation modes: Master mode and Slave mode. Up to 24 Mbps can be supported in Master or 12MHz in Slave mode under a 48MHz system clock. It has a Transfer Completion Flag (SPIF), Write Collision Flag (WCOL) and Mode Fault flag (MODF) in the SPI status register (SPSTAT). And a specially designed Transmit Holding Register (THR) improves the transmit performance compared to the conventional SPI and THRF flag indicates the THR is full or empty. SPIBSY read-only flag reports the Busy state in SPI engine. SPIBSY read-only flag reports the Busy state in SPI engine.

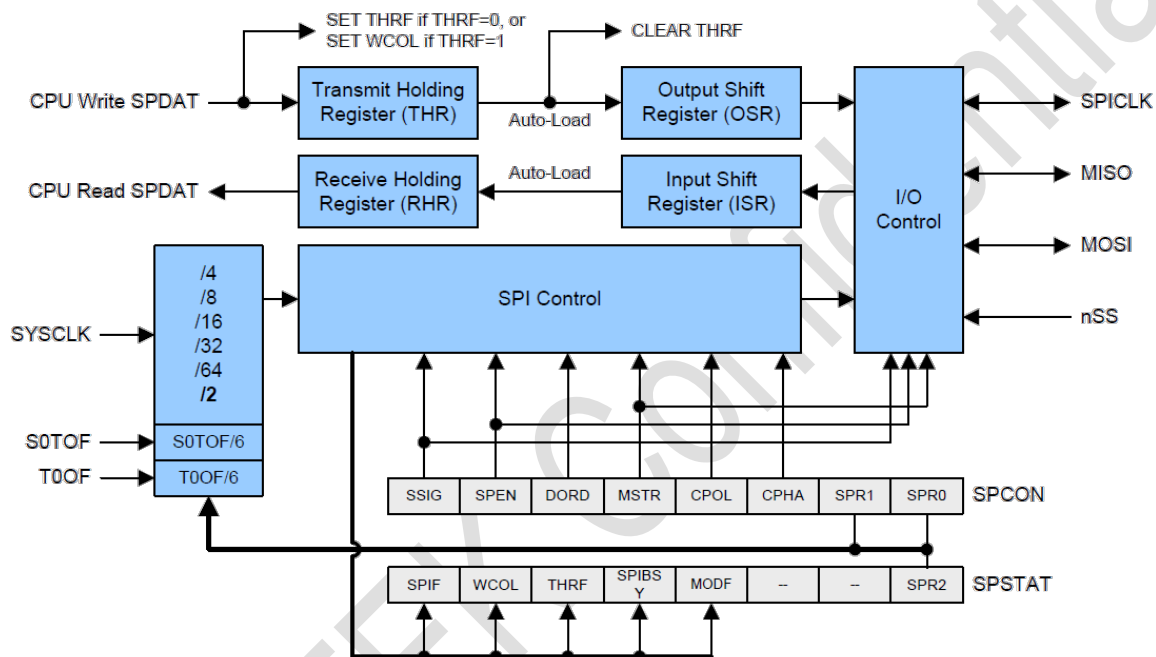


Figure 20-1. SPI Block Diagram

The SPI interface has four pins: MISO, MOSI, SPICLK and nSS:

- SPICLK, MOSI and MISO are typically tied together between two or more SPI devices. Data flows from master to slave on the MOSI pin (Master Out / Slave In) and flows from slave to master on the MISO pin (Master In / Slave Out). The SPICLK signal is output in the master mode and is input in the slave mode. If the SPI system is disabled, i.e., SPEN (SPCTL.6) = 0, these pins function as normal I/O pins.
- /SS is the optional slave select pin. In a typical configuration, an SPI master asserts one of its port pins to select one SPI device as the current slave. An SPI slave device uses its nSS pin to determine whether it is selected. The /SS is ignored if any of the following conditions are true:
 - If the SPI system is disabled, i.e. SPEN (SPCTL.6) = 0 (reset value).
 - If the SPI is configured as a master, i.e., MSTR (SPCTL.4) = 1, and nSS GPIO is configured as an output.
 - If the /SS pin is ignored, i.e. SSIG (SPCTL.7) bit = 1, this pin is configured for port functions.用

Note:

1. See the AUXR8 in Section “4.3 Alternate Function Redirection”, for its alternate pin-out option.
2. Even if the SPI is configured as a master (MSTR=1), it can still be converted to slave mode by the logic low of nSS pin input (if SSIG=0). Should this happen, the SPIF bit (SPSTAT.7) will be set and SPEN will be cleared. (See Section “20.2.3 Mode Change on nSS-pin”).

20.1 Typical SPI Configurations

20.1.1 Single Master & Single Slave

For the master: any port pin, including nSS GPIO, can be used to drive the nSS pin of the slave. For the slave: SSIG is '0', and nSS pin is used to determine whether it is selected.

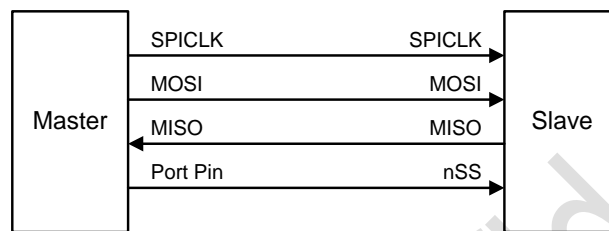


Figure 20-2. SPI single master & single slave configuration

20.1.2 Dual Device, where either can be a Master or a Slave

Two devices are connected to each other and either device can be a master or a slave. When no SPI operation is occurring, both can be configured as masters with MSTR=1, SSIG=0 and nSS GPIO configured in quasi-bidirectional mode. When any device initiates a transfer, it can configure P1.4 as an output and drive it low to force a “mode change to slave” in the other device. (See Section “20.2.3 Mode Change on nSS-pin”).

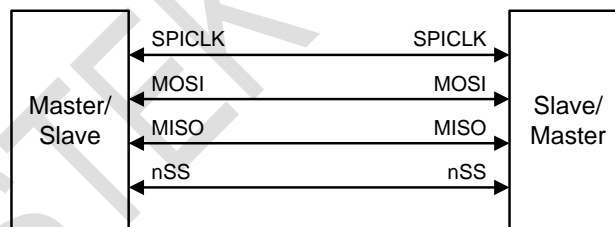


Figure 18-3. SPI dual device configuration, where either can be a master or a slave

20.1.3 Single Master & Multiple Slaves

For the master: any port pin, including nSS GPIO, can be used to drive the nSS pins of the slaves. For all the slaves: SSIG is '0', and nSS pin are used to determine whether it is selected.

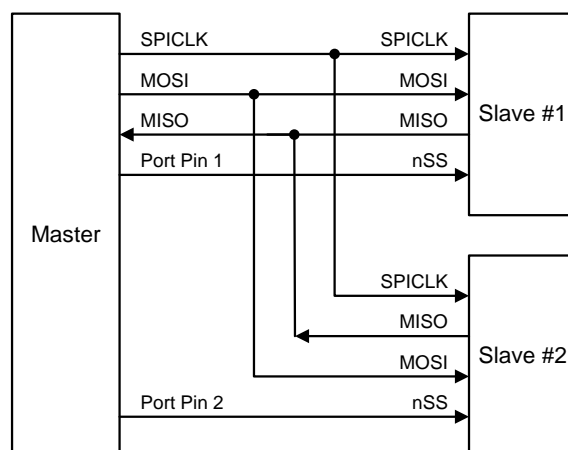


Figure 20-4.SPI single master multiple slaves configuration

20.2 Configuring the SPI

Table 20–1 shows configuration for the master/slave modes as well as usages and directions for the modes.

Table 201. SPI Master and Slave Selection

SPEN (SPCON.6)	SSIG (SPCON.7)	nSS -pin	MSTR (SPCON.4)	Mode	MISO -pin	MOSI -pin	SPICLK -pin	Remarks
0	X	X	X	SPI disabled	input	input	input	SPI assigned port pins are used as general port pins.
1	0	0	0	Slave (selected)	output	input	input	Selected as slave.
1	0	1	0	Slave (not selected)	Hi-Z	input	input	Not selected.
1	0	0	1→0	Slave (by mode change)	output	input	input	Mode change to slave if nSS pin is driven low, then MSTR will be cleared to '0' by HW automatically, and SPEN is cleared, MODF is set.
1	0	1	1	Master (idle)	input	Hi-Z	Hi-Z	MOSI and SPICLK are at high impedance to avoid bus contention when the Master is idle.
				Master (active)		output	output	MOSI and SPICLK are push-pull when the Master is active.
1	1	X	0	Slave	output	input	input	
1	1	X	1	Master	input	output	output	

“X” means “don’t care”.

20.2.1 Additional Considerations for a Slave

When CPHA is 0, SSIG must be 0 and nSS pin must be negated and reasserted between each successive serial byte transfer. Note the SPDAT register cannot be written while nSS pin is active (low), and the operation is undefined if CPHA is 0 and SSIG is 1.

When CPHA is 1, SSIG may be 0 or 1. If SSIG=0, the nSS pin may remain active low between successive transfers (can be tied low at all times). This format is sometimes preferred for use in systems having a single fixed master and a single slave configuration.

20.2.2 Additional Considerations for a Master

In SPI, transfers are always initiated by the master. If the SPI is enabled (SPEN=1) and selected as master, writing to the SPI data register (SPDAT) by the master starts the SPI clock generator and data transfer. The data will start to appear on MOSI about one half SPI bit-time to one SPI bit-time after data is written to SPDAT.

Before starting the transfer, the master may select a slave by driving the nSS pin of the corresponding device low. Data written to the SPDAT register of the master is shifted out of MOSI pin of the master to the MOSI pin of the slave. And, at the same time the data in SPDAT register of the selected slave is shifted out on MISO pin to the MISO pin of the master.

After shifting one byte, the SPI clock generator stops, setting the transfer completion flag (SPIF) and an interrupt will be created if the SPI interrupt is enabled. The two shift registers in the master CPU and slave CPU can be considered as one distributed 16-bit circular shift register. When data is shifted from the master to the slave, data is also shifted in the opposite direction simultaneously. This means that during one shift cycle, data in the master and the slave are interchanged.

20.2.3 Mode Change on nSS-pin

If SPEN=1, SSIG=0, MSTR=1 and /SS pin=1, the SPI is enabled in master mode. In this case, another master can drive this pin low to select this device as an SPI slave and start sending data to it. To avoid bus contention, the SPI becomes a slave. As a result of the SPI becoming a slave, the MOSI and SPICLK pins are forced to be an input and MISO becomes an output. The SPIF flag in SPSTAT is set, and if the SPI interrupt is enabled, an SPI interrupt will occur. User software should always check the MSTR bit. If this bit is cleared by a slave select and the user wants to continue to use the SPI as a master, the user must set the MSTR bit again, otherwise it will stay in slave mode.

20.2.4 Transmit Holding Register Full Flag

To speed up the SPI transmit performance, a specially designed Transmit Holding Register (THR) improves the latency time between byte to byte transmitting in CPU data moving. And a set THR-Full flag, THRF (SPSTAT.5), indicates the data in THR is valid and waiting for transmitting. If THR is empty (THRF=0), software writes one byte data to SPDAT will store the data in THR and set the THRF flag. If Output Shift Register (OSR) is empty, hardware will move THR data into OSR immediately and clear the THRF flag. In SPI master mode, valid data in OSR triggers a SPI transmit. In SPI slave mode, valid data in OSR is waiting for another SPI master to shift out the data. If THR is full (THRF=1), software writes one byte data to SPDAT will set a write collision flag, WCOL (SPSTAT.6).

20.2.5 Write Collision

The SPI in CMT2380F17 is double buffered data both in the transmit direction and in the receive direction. New data for transmission cannot be written to the THR until the THR is empty. The read-only flag, THRF, indicates the THR is full or empty. The WCOL (SPSTAT.6) bit is set to indicate data collision when the data register is written during set THRF. In this case, the SPDAT writing operation is ignored.

While write collision is detected for a master or a slave, it is uncommon for a master because the master has full control of the transfer in progress. The slave, however, has no control over when the master will initiate a transfer and therefore collision can occur.

WCOL can be cleared in software by writing '1' to the bit.

20.2.6 SPI Clock Rate Select

The SPI clock rate selection (in master mode) uses the SPR1 and SPR0 bits in the SPCON register and SPR2 in the SPSTAT register, as shown in Table 20–2.

Table 20-2. SPI Serial Clock Rates

SPR2	SPR1	SPR0	SPI Clock Selection	SPI Clock Rate @ SYSCLK=12MHz	SPI Clock Rate @ SYSCLK=48MHz
0	0	0	SYSCLK/4	3MHz	12MHz
0	0	1	SYSCLK/8	1.5MHz	6MHz
0	1	0	SYSCLK/16	750KHz	3MHz
0	1	1	SYSCLK/32	375KHz	1.5MHz
1	0	0	SYSCLK/64	187.5KHz	750KHz
1	0	1	SYSCLK/2	6MHz	24MHz
1	1	0	S0TOF/6	Variable	Variable
1	1	1	T0OF/6	Variable	Variable

Note:

1. SYSCLK is the system clock.
2. S0TOF is UART0 Baud-Rate Generator Overflow.
3. T0OF is Timer 0 Overflow.

20.3 Data Mode

Clock Phase Bit (CPHA) allows the user to set the edges for sampling and changing data. The Clock Polarity bit, CPOL, allows the user to set the clock polarity. The following figures show the different settings of Clock Phase Bit, CPHA.

Table 20–3. SPI mode definition

SPI Mode	CPOL	CPHA	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)

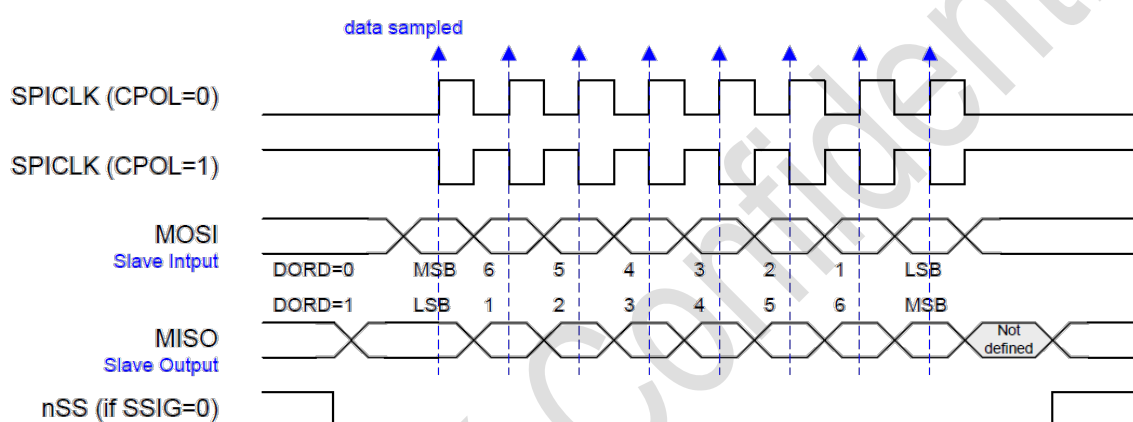


Figure 20-5. SPI Slave Transfer Format with CPHA=0

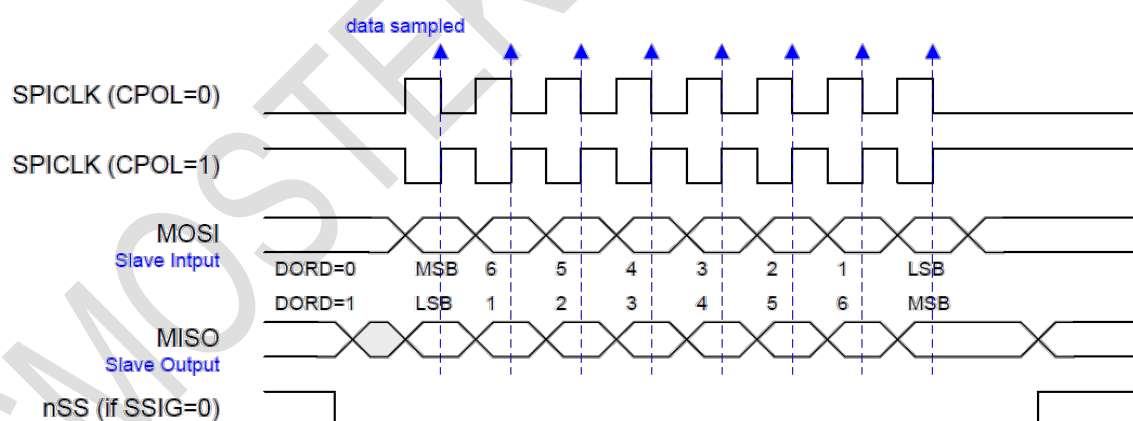


Figure 20-6. Slave Transfer Format with CPHA=1

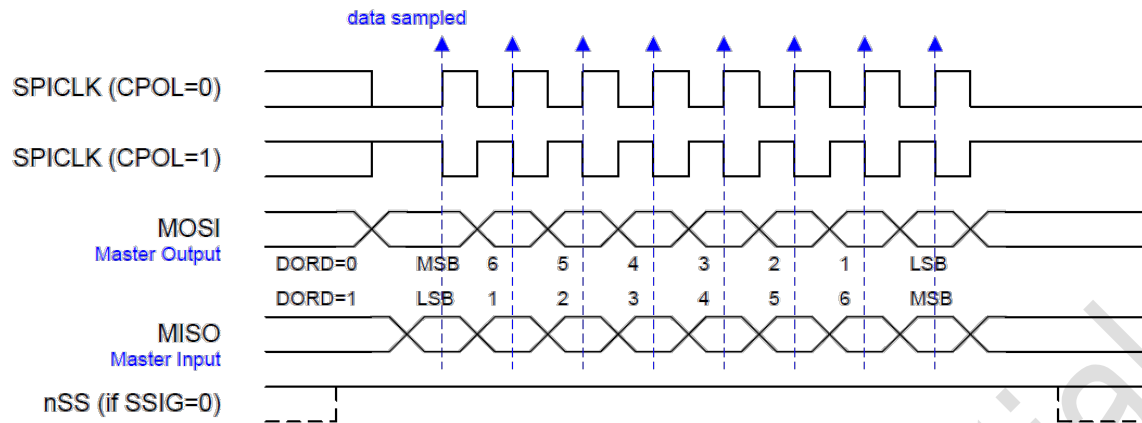


Figure 20-7. SPI Master Transfer Format with CPHA=0

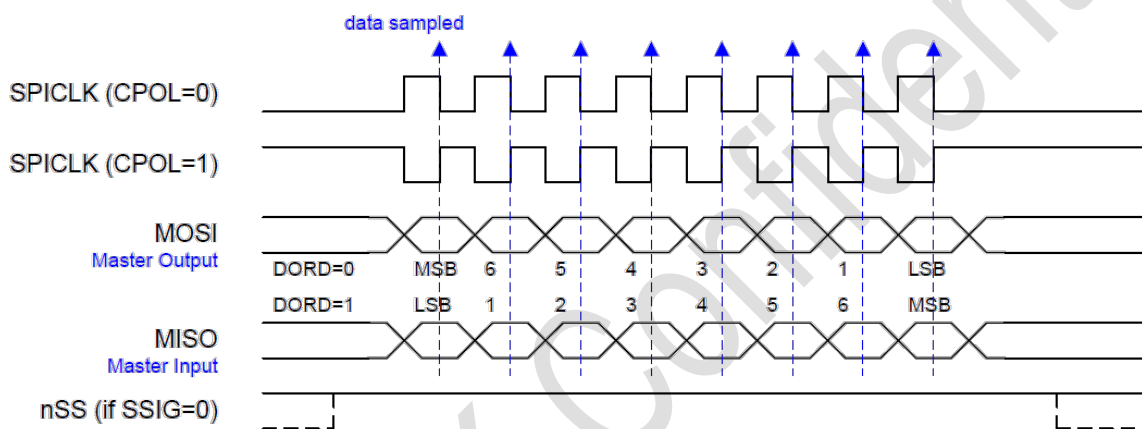


Figure 20-8. SPI Master Transfer Format with CPHA=1

20.4 Daisy-Chain Connection

If SPI0 is defined in slave mode, it can be connected in a daisy-chain configuration. The first slave output is connected to the second slave input, the second slave output is connected to the third slave input, and so on. The final slave output is connected to the master input. Each slave sends out, during a second group of clock pulses, an exact copy of what was received during the first group of clock pulses. The whole chain acts as one large communication shift register. The daisy-chain feature only requires a single Slave Select line (nSS) from the master device.

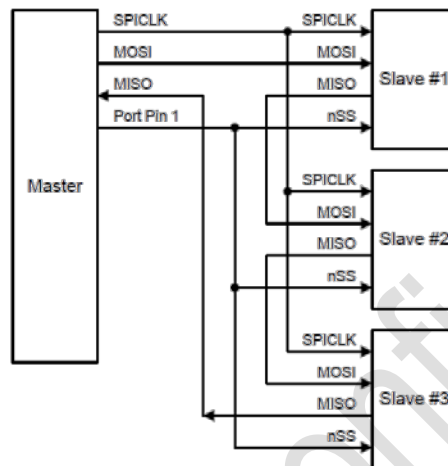


Figure 20–9. SPI single master multiple slaves configuration

20.4.1 Configuring the Daisy-Chain

How to Configure SPI Slave in Daisy-Chain

- Configure SPCON to define the data mode and select SPI0 in slave mode.
- Set SPI0M0 (AUXR7.4) to enable SPI0 in Daisy-Chain mode.
- Service SPIF to get daisy-chain communication.。

20.5 SPI Register

The following special function registers are related to the SPI operation:

- **SPCON:** SPI Control Register

SFR Page = 0~F

SFR Address = 0x85

Bit	7	6	5	4	3	2	1	0
Name	SSIG	SPEN	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	1	0	0

Bit 7: SSIG, nSS is ignored.

0: The nSS pin decides whether the device is a master or slave. 1: MSTR decides whether the device is a master or slave.

Bit 6: SPEN, SPI enable.

0: The SPI interface is disabled and all SPI pins will be general-purpose I/O ports. 1: The SPI is enabled.

Bit 5: DORD, SPI data order.

0: The MSB of the data byte is transmitted first. 1: The LSB of the data byte is transmitted first.

Bit 4: MSTR, Master/Slave mode select 0: Selects slave SPI mode.

1: Selects master SPI mode.

Bit 3: CPOL, SPI clock polarity select

0: SPICLK is low when Idle. The leading edge of SPICLK is the rising edge and the trailing edge is the falling edge. 1: SPICLK is high when Idle. The leading edge of SPICLK is the falling edge and the trailing edge is the rising edge.

Bit 2: CPHA, SPI clock phase select

0: Data is driven when /SS pin is low (SSIG=0) and changes on the trailing edge of SPICLK. Data is sampled on the leading edge of SPICLK.

1: Data is driven on the leading edge of SPICLK, and is sampled on the trailing edge. (Note: If SSIG=1, CPHA must not be 1, otherwise the operation is not defined.)

Bit 1~0: SPR1-SPR0, SPI clock rate select 0 & 1 (associated with SPR2, when in master mode)

SPR2	SPR1	SPR0	SPI Clock Selection	SPI Clock Rate @ SYSCLK=12MHz	SPI Clock Rate @ SYSCLK=48MHz
0	0	0	SYSCLK/4	3MHz	12MHz
0	0	1	SYSCLK/8	1.5MHz	6MHz
0	1	0	SYSCLK/16	750KHz	3MHz
0	1	1	SYSCLK/32	375KHz	1.5MHz
1	0	0	SYSCLK/64	187.5KHz	750KHz
1	0	1	SYSCLK/2	6MHz	24MHz
1	1	0	S0TOF/6	Variable	Variable
1	1	1	T0OF/6	Variable	Variable

Note:

- SYSCLK is the system clock.
- S0TOF is UART0 Baud-Rate Generator Overflow.
- T0OF is Timer 0 Overflow.

- SPSTAT: SPI Status Register

SFR Page = 0~F

SFR Address = 0x84

Bit	7	6	5	4	3	2	1	0
Name	SPIF	WCOL	THRF	SPIBSY	MODF	--	--	SPR2
R/W	R/W	R/W	R	R	R/W	W	W	R/W
Reset Value	0	0	0	0	X	X	X	0

Bit 7: SPIF, SPI transfer completion flag

0: The SPIF is cleared in software by writing "1" to this bit.

1: When a serial transfer finishes, the SPIF bit is set and an interrupt is generated if SPI interrupt is enabled. If nSS pin is driven low when SPI is in master mode with SSIG=0, SPIF will also be set to signal the "mode change".

Bit 6: WCOL, SPI write collision flag.

0: The WCOL flag is cleared in software by writing "1" to this bit.

1: The WCOL bit is set if the SPI data register, SPDAT, is written during a data transfer (see Section "20.2.5 Write Collision").

Bit 5: THRF, Transmit Holding Register (THR) Full flag. Read only.

0: Means the THR is "empty". This bit is cleared by hardware when the THR is empty. That means the data in THR is loaded (by H/W) into the Output Shift Register to be transmitted, and now the user can write the next data byte to SPDAT for next transmission.

1: Means the THR is "full". This bit is set by hardware just when SPDAT is written by software.

Bit 4: SPIBSY, SPI Busy flag. Read only.

0: It indicates SPI engine is idle and all shift registers are empty.

1: It is set to logic 1 when a SPI transfer is in progress (Master or slave Mode).

Bit 3: Mode Fault Flag. This bit is set to logic 1 by hardware when a master mode collision is detected (nSS is low, MSTEN = 1, and SSIG = 0). If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software writing "1".

Bit 2~1: Reserved. Software must write "0" on these bits when SPSTAT is written. Bit 0: SPR2, SPI clock rate select 2 (associated with SPR1 and SPR0)

- SPDAT: SPI Data Register

SFR Page = 0~F

SFR Address = 0x86

Bit	7	6	5	4	3	2	1	0
Name	SPDAT[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

SPDAT has two physical buffers for writing to and reading from during transmit and receive, respectively.

- AUXR7: Auxiliary Register 7

SFR Page = 4 only

SFR Address = 0xA44

Bit	7	6	5	4	3	2	1	0
Name	POE5	POE4	COCKOE	SPI0M0	--	--	--	--
R/W	R/W	R/W	R/W	R/W	W	W	W	W
Reset Value	1	1	0	0	X	X	X	0

Bit 4: SPI0M0, SPI0 model control bit 0. It controls the SPI application with daisy-chain connection. 0: Disable the mode control.
1: Enable the mode control.

- AUXR10: Auxiliary Register 10

SFR Page = 7 only

SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	SPIPS0	S0PS1	--	TWICF	PAA
R/W	W	W	W	R/W	R/W	W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 4: SPIPS0, SPI Port pin Selection 0.

SPIPS0	nSS	MOSI	MISO	SPICLK
0	P3.3	P1.5	P1.6	P1.7
1	P1.7	P3.5	P3.4	P3.3

21 Two Wire serial Interface (TWI0/ I2C0)

The Two-Wire serial Interface is a two-wire, bi-directional serial bus. It is ideally suited for typical microcontroller applications.

The TWI/ I2C protocol allows the systems designer to interconnect up to 128 different devices using only two bi-directional bus lines, one for clock (TWI0_SCL) and one for data (TWI0_SDA). The TWI bus provides control of TWI0_SDA (serial data), TWI0_SCL (serial clock) generation and synchronization, arbitration logic, and START/STOP control and generation. The only external hardware needed to implement this bus is a single pull-up resistor for each of the TWI bus lines. All devices connected to the bus have individual addresses, and mechanisms for resolving bus contention are inherent in the TWI/ I2C protocol.

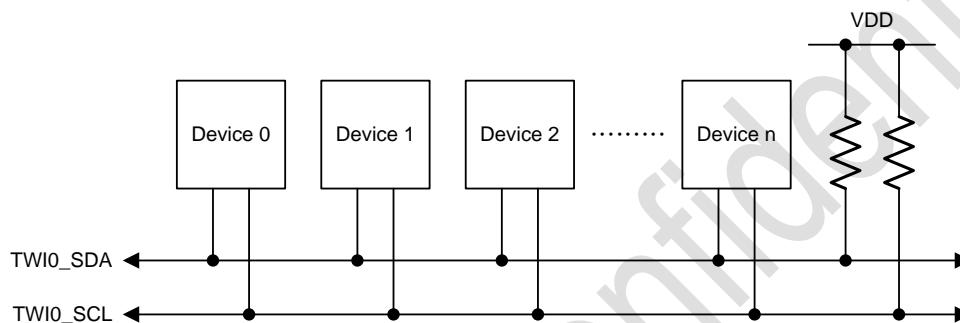


Figure 21-1. TWI/ I2C Bus Interconnection

The TWI/ I2C bus may operate as a master and/or slave, and may function on a bus with multiple masters. The CPU interfaces to the TWI/ I2C through the following four special function registers: SICON configures the TWI/ I2C bus; SISTA reports the status code of the TWI/ I2C bus; and SIDAT is the data register, used for both transmitting and receiving TWI/ I2C data. SIADR is the slave address register. And, the TWI/ I2C hardware interfaces to the serial bus via two lines: SDA (serial data line) and SCL (serial clock line).

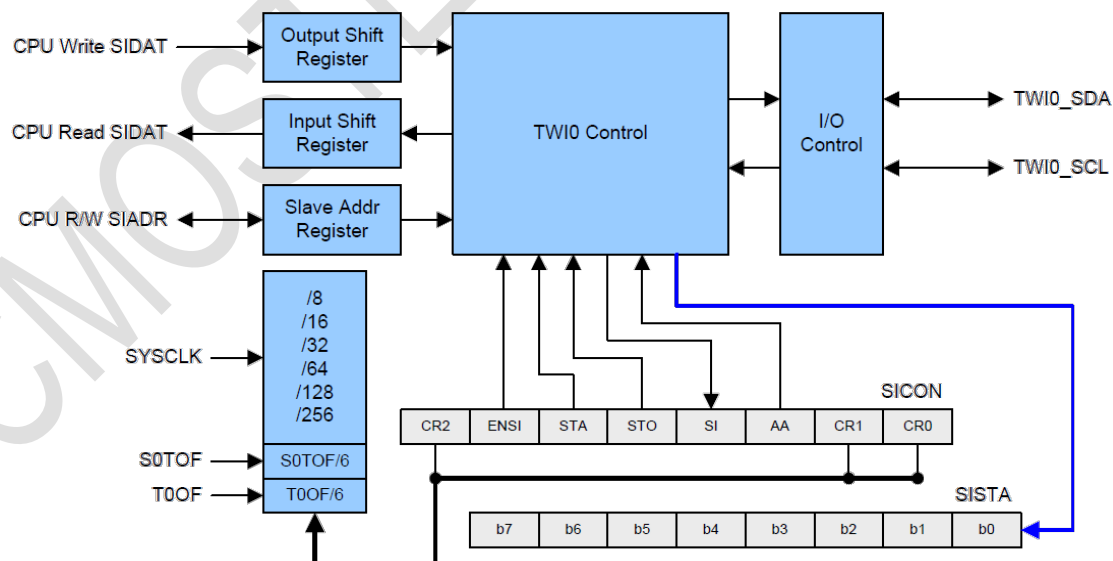


Figure 21-2. TWI/ I2C Block Diagram

21.1 Operating Modes

There are four operating modes for the TWI/ I2C:

- 1) Master/Transmitter mode;
- 2) Master/Receiver mode;
- 3) Slave/Transmitter mode;
- 4) Slave/Receiver mode.

Bits STA, STO and AA in SICON decide the next action which the TWI hardware will take after SI is cleared by software. When the next action is completed, a new status code in SISTA will be updated and SI will be set by hardware in the same time. Now, the interrupt service routine is entered (if the TWI/ I2C interrupt is enabled), and the new status code can be used to determine which appropriate routine the software is to branch to.

21.1.1 Master Transmitter Mode

In the master transmitter mode, a number of data bytes are transmitted to a slave receiver. Before the master transmitter mode can be entered, SICON must be initialized as follows:

● SICON

Bit	7	6	5	4	3	2	1	0
Name	CR2	ENSI	STA	STO	SI	AA	CR1	CR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	Bitrate	1	0	0	0	X	Bitrate	

CR0, CR1, and CR2 define the serial bit rate. ENSI must be set to logic 1 to enable TWI/ I2C. If the AA bit is reset, TWI/ I2C will not acknowledge its own slave address or the general call address in the event of another device becoming master of the bus. In other words, if AA is reset, TWI/ I2C cannot enter a slave mode. STA, STO, and SI must be reset.

The master transmitter mode may now be entered by software setting the STA bit. The TWI/ I2C logic will now test the serial bus and generate a START condition as soon as the bus becomes free. When a START condition is transmitted, the serial interrupt flag (SI) is set, and the status code in the status register (SISTA) will be 08H. This status code must be used to vector to an interrupt service routine that loads SIDAT with the slave address and the data direction bit (SLA+W). The SI bit in SICON must then be reset before the serial transfer can continue.

When the slave address and the direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again, and a number of status codes in SISTA are possible. There are 18H, 20H, or 38H for the master mode and also 68H, 78H, or B0H if the slave mode was enabled (AA=1). The appropriate action to be taken for each of these status codes is detailed in the following operating flow chart. After a repeated START condition (state 10H), TWI/ I2C may switch to the master receiver mode by loading SIDAT with SLA+R.

21.1.2 Master Receiver Mode

In the master receiver mode, a number of data bytes are received from a slave transmitter. SICON must be initialized as in the master transmitter mode. When the start condition has been transmitted, the interrupt service routine must load SIDAT with the 7-bit slave address and the data direction bit (SLA+R). The SI bit in SICON must then be cleared before the serial transfer can continue.

When the slave address and the data direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again, and a number of status codes in SISTA are possible. They are 40H, 48H, or 38H for the master mode and also 68H, 78H, or B0H if the slave mode was enabled (AA=1). The appropriate action to be taken for each of these status codes is detailed in the following operating flow chart. After a repeated start condition (state 10H), TWI/ I2C may switch to the master transmitter mode by loading SIDAT with SLA+W.

21.1.3 Slave Transmitter Mode

In the slave transmitter mode, a number of data bytes are transmitted to a master receiver. To initiate the slave transmitter mode, SIADR and SICON must be loaded as follows:

- **SIADR**

Bit	7	6	5	4	3	2	1	0
Name	A6	A5	A4	A3	A2	A1	A0	GC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	X	0	0

|<----- Own Slave Address ----->|

The upper 7 bits are the address to which TWI/ I2C will respond when addressed by a master. If the LSB (GC) is set, TWI/ I2C will respond to the general call address (00H); otherwise it ignores the general call address.

- **SICON**

Bit	7	6	5	4	3	2	1	0
Name	CR2	ENSI	STA	STO	SI	AA	CR1	CR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	X	1	0	0	0	1	X	X

CR0, CR1, and CR2 do not affect TWI/ I2C in the slave mode. ENSI must be set to “1” to enable TWI/ I2C. The AA bit must be set to enable TWI/ I2C to acknowledge its own slave address or the general call address. STA, STO, and SI must be cleared to “0”.

When SIADR and SICON have been initialized, TWI/ I2C waits until it is addressed by its own slave address followed by the data direction bit which must be “1” (R) for TWI/ I2C to operate in the slave transmitter mode. After its own slave address and the “R” bit have been received, the serial interrupt flag (SI) is set and a valid status code can be read from SISTA. This status code is used to vector to an interrupt service routine, and the appropriate action to be taken for each of these status codes is detailed in the following operating flow chart. The slave transmitter mode may also be entered if arbitration is lost while TWI/ I2C is in the master mode (see state B0H).

If the AA bit is reset during a transfer, TWI/ I2C will transmit the last byte of the transfer and enter state C0H or C8H. TWI/ I2C is switched to the not-addressed slave mode and will ignore the master receiver if it continues the transfer. Thus the master receiver receives all 1s as serial data. While AA is reset, TWI/ I2C does not respond to its own slave address or a general call address. However, the serial bus is still monitored,

and address recognition may be resumed at any time by setting AA. This means that the AA bit may be used to temporarily isolate TWI/ I2C from the bus.

21.1.4 Slave Receiver Mode

In the slave receiver mode, a number of data bytes are received from a master transmitter. Data transfer is initialized as in the slave transmitter mode.

When SIADR and SICON have been initialized, TWI/ I2C waits until it is addressed by its own slave address followed by the data direction bit which must be "0" (W) for TWI/ I2C to operate in the slave receiver mode. After its own slave address and the W bit have been received, the serial interrupt flag (SI) is set and a valid status code can be read from SISTA. This status code is used to vector to an interrupt service routine, and the appropriate action to be taken for each of these status codes is detailed in the following operating flow chart. The slave receiver mode may also be entered if arbitration is lost while TWI/ I2C is in the master mode (see status 68H and 78H).

If the AA bit is reset during a transfer, TWI/ I2C will return a not acknowledge (logic 1) to SDA after the next received data byte. While AA is reset, TWI/ I2C does not respond to its own slave address or a general call address. However, the serial bus is still monitored and address recognition may be resumed at any time by setting AA. This means that the AA bit may be used to temporarily isolate from the bus.

21.2 Miscellaneous States

There are two SISTA codes that do not correspond to a defined TWI/ I2C hardware state, as described below.

- **S1STA=F8H**

This status code indicates that no relevant information is available because the serial interrupt flag, SI, is not yet set. This occurs between other states and when TWI/ I2C is not involved in a serial transfer.

- **S1STA = 00H:**

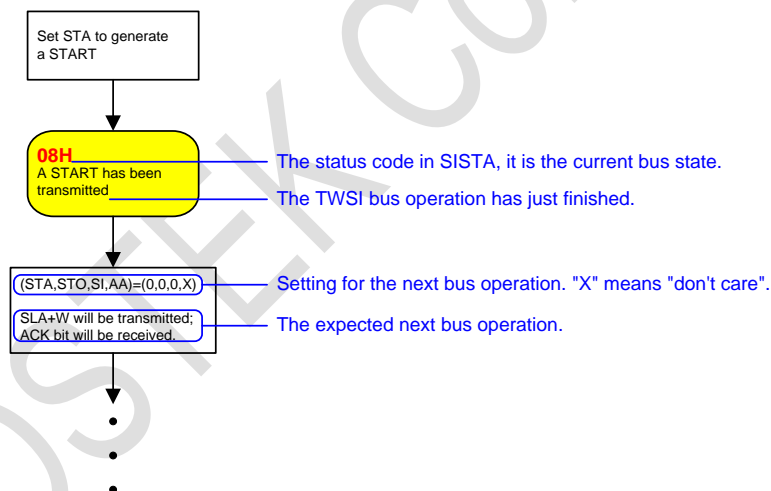
This status code indicates that a bus error has occurred during a TWI/ I2C serial transfer. A bus error is caused when a START or STOP condition occurs at an illegal position in the format frame. Examples of such illegal positions are during the serial transfer of an address byte, a data byte, or an acknowledge bit. A bus error may also be caused when external interference disturbs the internal TWI/ I2C signals. When a bus error occurs, SI is set. To recover from a bus error, the STO flag must be set and SI must be cleared by software. This causes TWI/ I2C to enter the "not-addressed" slave mode (a defined state) and to clear the STO flag (no other bits in SICON are affected). The TWI/ I2C0_SDA and TWI/ I2C0_SCL lines are released (a STOP condition is not transmitted).

21.3 Using the TWI/ I2C

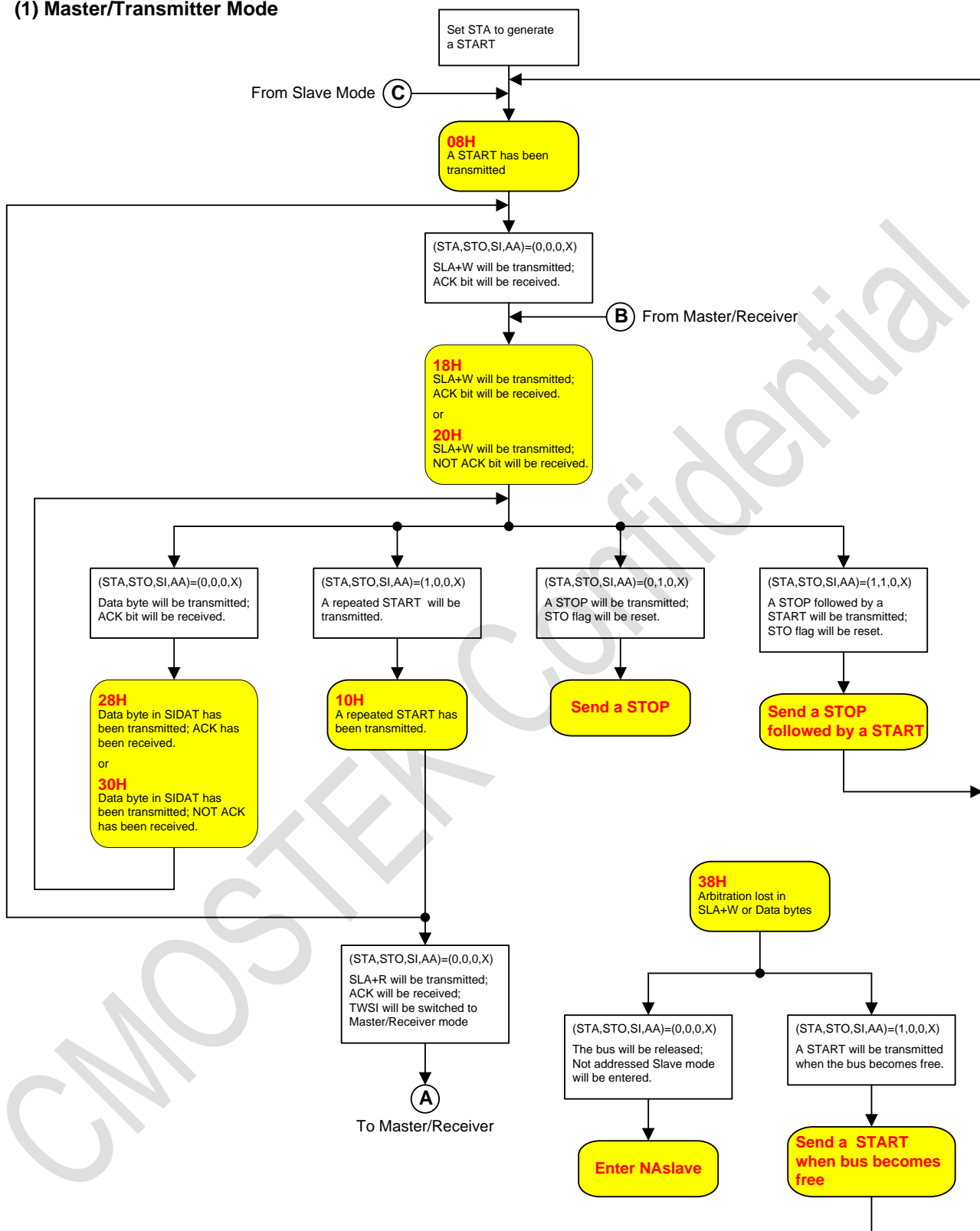
The TWI/ I2C is byte-oriented and interrupt based. Interrupts are issued after all bus events, like reception of a byte or transmission of a START condition. Because the TWI/ I2C is interrupt-based, the application software is free to carry on other operations during a TWI/ I2C byte transfer. Note that the TWI/ I2C0 interrupt enable bit ETWI/ I2C0 bit (EIE1.6) together with the EA bit allow the application to decide whether or not assertion of the SI Flag should generate an interrupt request. When the SI flag is asserted, the TWI/ I2C has finished an operation and awaits application response. In this case, the status register SISTA contains a status code indicating the current state of the TWI/ I2C bus. The application software can then decide how the TWI/ I2C should behave in the next TWI/ I2C bus operation by properly programming the STA, STO and AA bits (in SICON).

The following operating flow charts will instruct the user to use the TWI/ I2C using state-by-state operation. First, the user should fill SIADR with its own Slave address (refer to the previous description about SIADR). To act as a master, after initializing the SICON, the first step is to set "STA" bit to generate a START condition to the bus. To act as a slave, after initializing the SICON, the TWI/ I2C waits until it is addressed. And then follow the operating flow chart for a number a next actions by properly programming (STA,STO,SI,AA) in the SICON. Since the TWI/ I2C hardware will take next action when SI is just cleared, it is recommended to program (STA,STO,SI,AA) by two steps, first STA, STO and AA, then clear SI bit (may use instruction "CLR SI") for safe operation. "don't care"

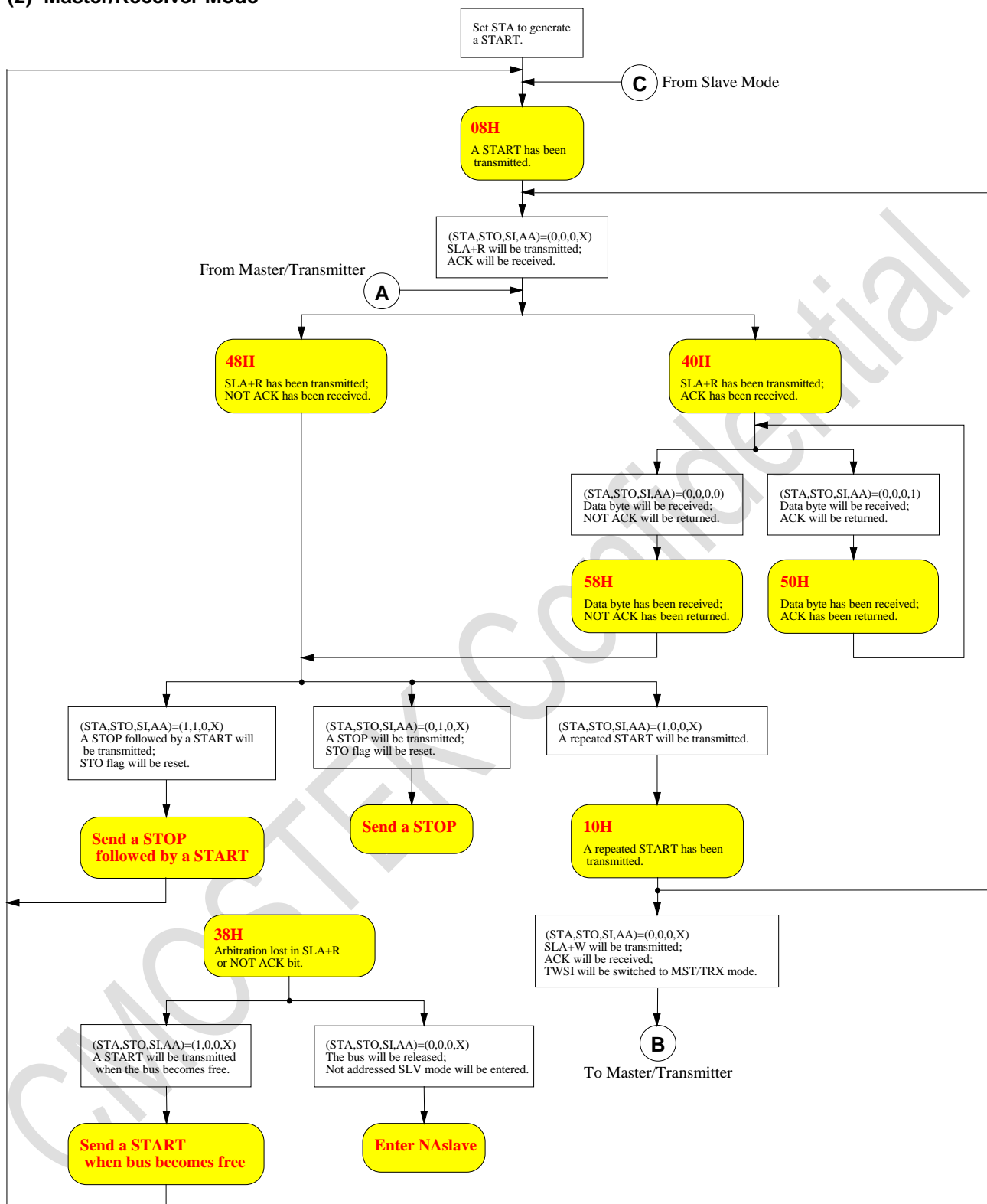
The figure below shows how to read the flow charts.



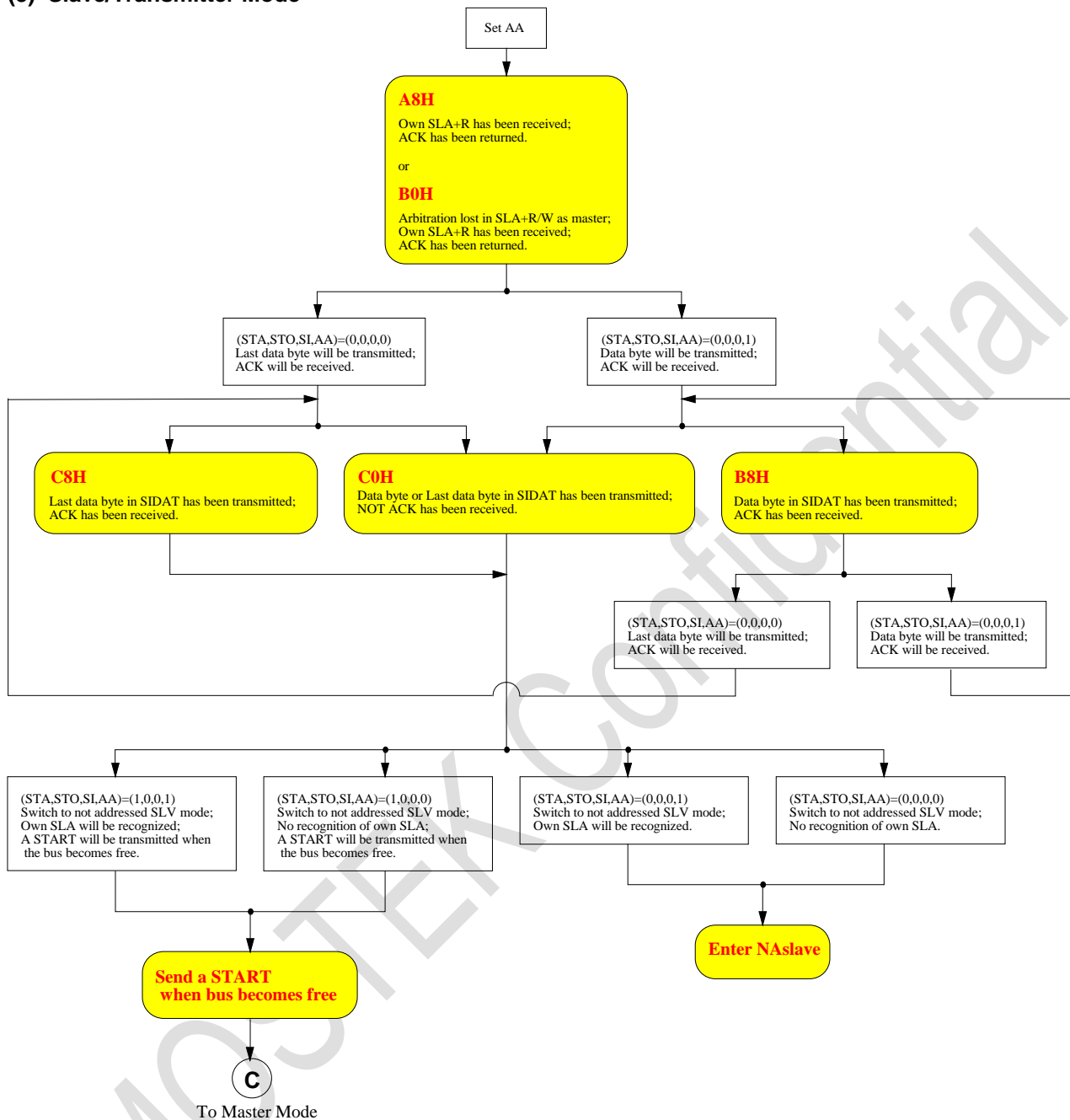
(1) Master/Transmitter Mode



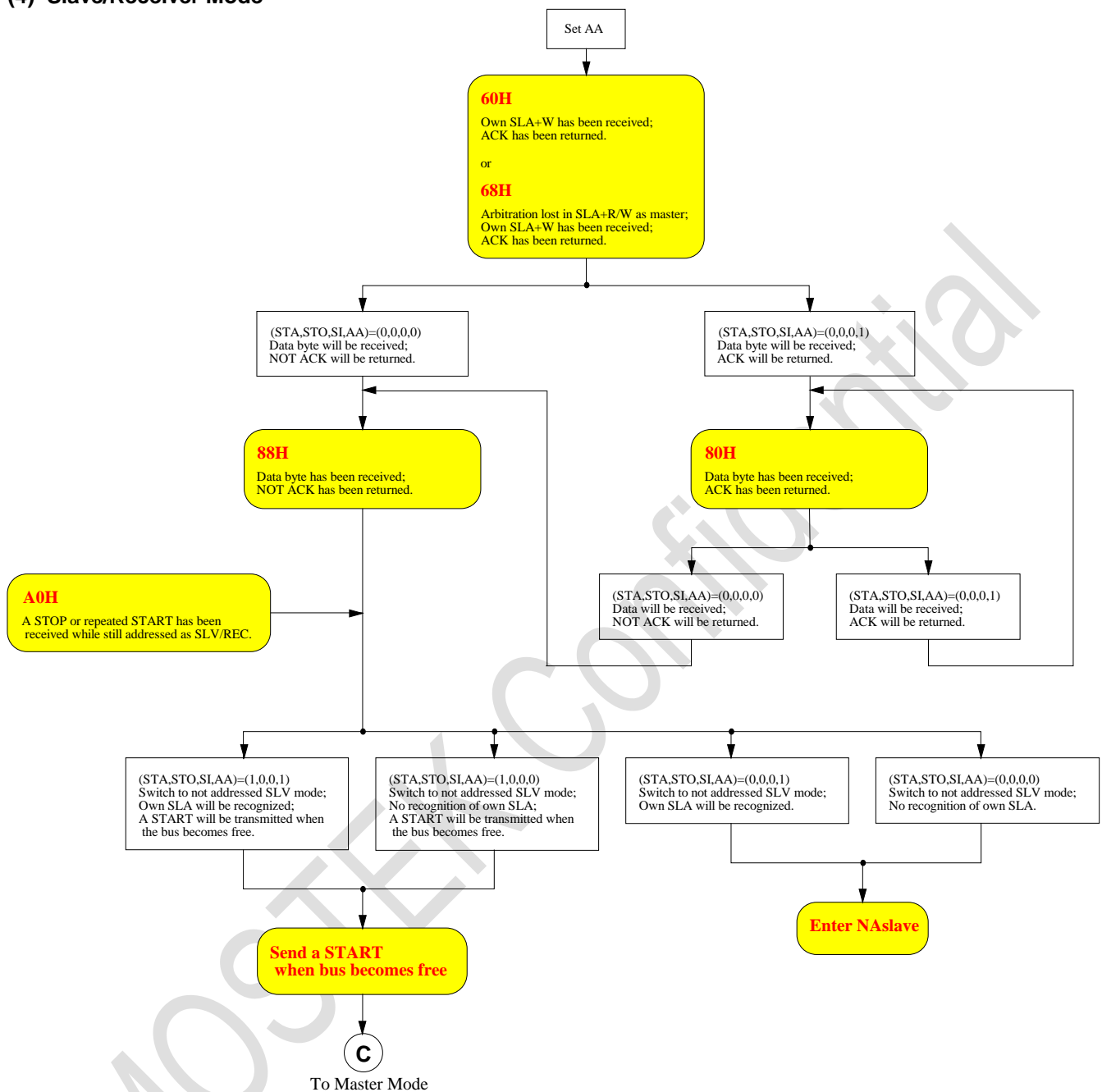
(2) Master/Receiver Mode



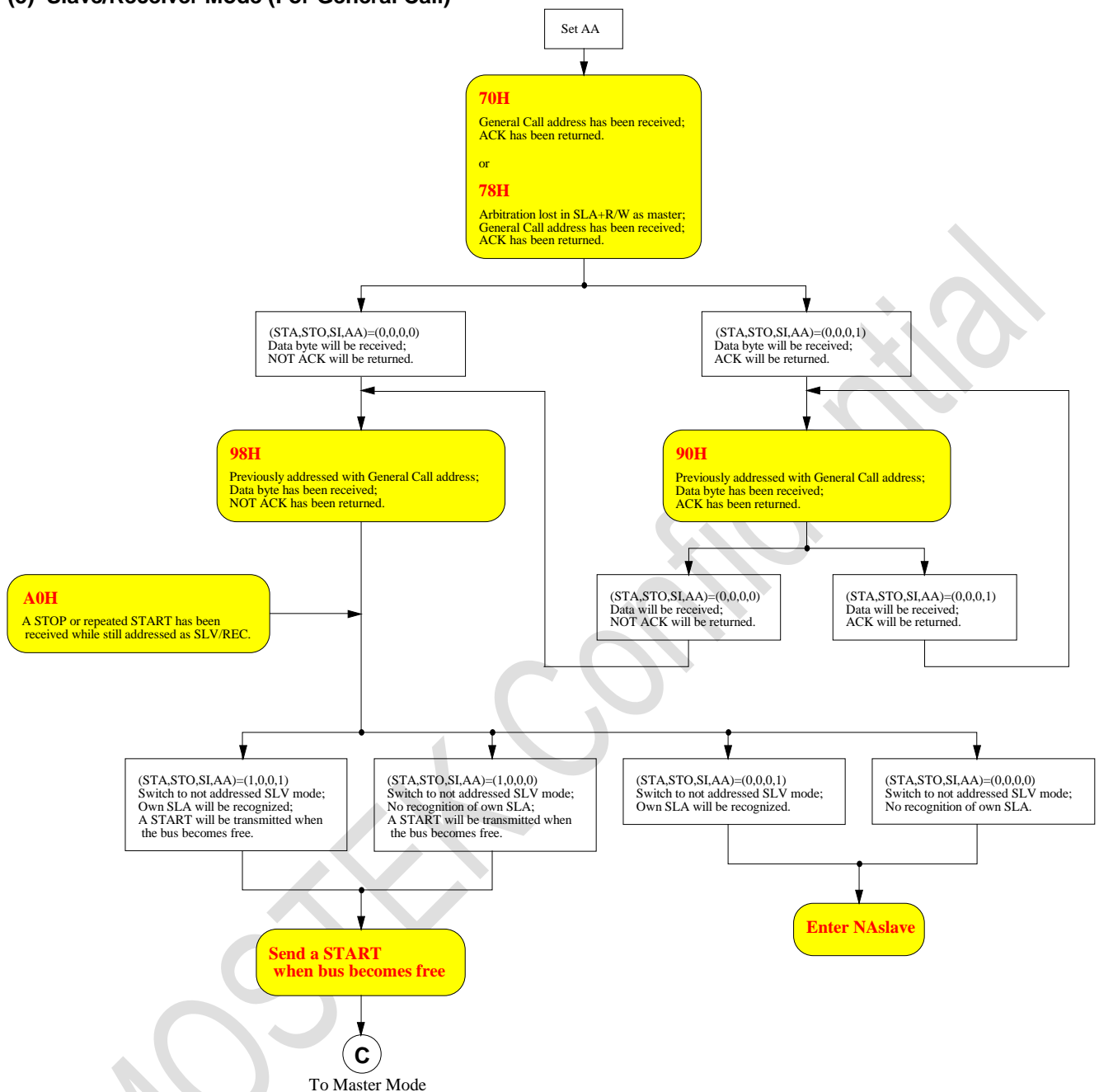
(3) Slave/Transmitter Mode



(4) Slave/Receiver Mode



(5) Slave/Receiver Mode (For General Call)



21.4 TWI0/ I2C0 Register

- **SIADR:** TWI0/ I2C0 Address Register

SFR Page = 0~F

SFR Address = 0xD1

Bit	7	6	5	4	3	2	1	0
Name	A6	A5	A4	A3	A2	A1	A0	GC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

The CPU can read from and write to this register directly. SIADR is not affected by the TWI0/ I2C0 hardware. The contents of this register are irrelevant when TWI0/ I2C0 is in a master mode. In the slave mode, the seven most significant bits must be loaded with the microcontroller's own slave address, and, if the least significant bit (GC) is set, the general call address (00H) is recognized; otherwise it is ignored. The most significant bit corresponds to the first bit received from the TWI0/ I2C0 bus after a START condition.

- **SIDAT:** TWI0/ I2C0 Data Register

SFR Page = 0~F

SFR Address = 0xD2

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

This register contains a byte of serial data to be transmitted or a byte which has just been received. The CPU can read from or write to this register directly while it is not in the process of shifting a byte. This occurs when TWI0/ I2C0 is in a defined state and the serial interrupt flag (SI) is set. Data in SIDAT remains stable as long as SI is set. While data is being shifted out, data on the bus is simultaneously being shifted in; SIDAT always contains the last data byte present on the bus. Thus, in the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data in SIDAT.

SIDAT and the ACK flag form a 9-bit shift register which shifts in or shifts out an 8-bit byte, followed by an acknowledge bit. The ACK flag is controlled by the TWI0/ I2C0 hardware and cannot be accessed by the CPU. Serial data is shifted through the ACK flag into SIDAT on the rising edges of serial clock pulses on the TWI0/ I2C0_SCL line. When a byte has been shifted into SIDAT, the serial data is available in SIDAT, and the acknowledge bit is returned by the control logic during the 9th clock pulse. Serial data is shifted out from SIDAT on the falling edges of clock pulses on the TWI0/ I2C0_SCL line.

When the CPU writes to SIDAT, the bit SD7 is the first bit to be transmitted to the SDA line. After nine serial clock pulses, the eight bits in SIDAT will have been transmitted to the SDA line, and the acknowledge bit will be present in the ACK flag. Note that the eight transmitted bits are shifted back into SIDAT.

- SICON: TWI0/ I2C0 Control Register

SFR Page = 0~F

SFR Address = 0xD4

Bit	7	6	5	4	3	2	1	0
Name	CR2	ENSI	STA	STO	SI	AA	CR1	CR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

The CPU can read and write to this register directly. Two bits are affected by the TWI0/I2C0 hardware: the SI will be set when a serial interrupt occurred, and the STO will be cleared when a STOP condition is present on the bus. The STO bit is also cleared when ENSI="0".

Bit 7: CR2, TWI0/ I2C0 Clock Rate select bit 2 (associated with CR1 and CR0). Bit 6: ENSI, the TWI0/I2C0 Hardware Enable Bit

When ENSI is "0", the TWI0 _SDA and TWI0_SCL outputs are in a high impedance state, and it will ignore the input signals. Under this condition, the TWI0/I2C0 is in the not-addressed slave state, and STO is forced to "0". No other bits are affected, and the TWI0_SDA and TWI0_SCL can be used as general purpose I/O pins. When ENSI is "1", TWI0 is enabled, the TWI0_SDA and TWI0_SCL assign to port pin latch, such as P4.1 and P4.0. The port pin latch must be set to logic 1 and I/O mode must be configured to open-drain mode for the serial communication.

Bit 5: STA, the START Flag

When sets the STA to enter master mode, the TWI0/I2C0 hardware will check the status of the serial bus. It will generate a START condition if the bus is free. Otherwise TWI0/I2C0 will wait for a STOP condition and generates a START condition after a delay. If STA is set while TWI0/I2C0 is already in a master mode and one or more bytes are transmitting or receiving, TWI0/I2C0 will send a repeated START condition. STA may be set at any time. STA may also be set when TWI0/I2C0 is an addressed slave mode. When the STA bit is reset, no START condition or repeated START condition will be generated.

Bit 4: STO, the STOP Flag

When the STO is set while TWI0/I2C0 is in a master mode, a STOP condition is transmitted to the serial bus. When the STOP condition is detected on the bus, the TWI0/I2C0 hardware clears the STO flag. In a slave mode, the STO flag may be set to recover from a bus error condition. In this case, no STOP condition is transmitted to the bus. However, the TWI0/I2C0 hardware behaves as if a STOP condition has been received and switches to the defined not addressed slave receiver mode. The STO flag is automatically cleared by hardware. If the STA and STO bits are both set, then a STOP condition is transmitted to the bus if TWI0/I2C0 is in a master mode (in a slave mode, TWI0/I2C0 generates an internal STOP condition which is not transmitted), and then transmits a START condition.

Bit 3: SI, the Serial Interrupt Flag

When a new TWI0/I2C0 state is present in the SISTA register, the SI flag is set by hardware. And, if the TWI0/I2C0 interrupt is enabled, an interrupt service routine will be serviced. The only state that does not cause SI to be set is state F8H, which indicates that no relevant state information is available. When SI is set, the low period of the serial clock on the TWI0_SCL line is stretched, and the serial transfer is suspended. A high level on the TWI0_SCL line is unaffected by the serial interrupt flag. SI must be cleared by software writing "0" on this bit. When the SI flag is reset, no serial interrupt is requested, and there is no stretching on the serial clock on the TWI0_SCL line.

Bit 2: AA, the Assert Acknowledge Flag

If the AA flag is set to "1", an Acknowledge (low level to TWI0_SDA) will be returned during the acknowledge clock pulse on the TWI0_SCL line when:

- 1) The own slave address has been received.
- 2) A data byte has been received while TWI0/I2C0 is in the master/receiver mode.
- 3) A data byte has been received while TWI0/I2C0 is in the addressed slave/receiver mode.

If the AA flag is reset to "0", a not acknowledge (high level to TWI0_SDA) will be returned during the acknowledge clock pulse on TWI0_SCL when:

- 1) A data has been received while TWI0/I2C0 is in the master/receiver mode.
- 2) A data byte has been received while TWI0/I2C0 is in the addressed slave/receiver mode.

Bit 7, 1~0: CR2, CR1 and CR0, the Clock Rate select Bits

These three bits determine the serial clock frequency when TWI0/I2C0 is in a master mode. The highest master mode clock frequency is limited to 1MHz. In slave mode, it is no need to select the clock rate. TWI0/I2C0 will automatically synchronize with any clock frequency from master, which is up to 400KHz. The various serial clock rates are shown in Table 21-1.

Table 21-1. TWI0/ I2C0 Serial Clock Rates

CR2	CR1	CR0	TWI0Clock Selection	TWI0Clock Rate @ SYSCLK=12MHz
0	0	0	SYSCLK/8	1.5 MHz
0	0	1	SYSCLK/16	750 KHz
0	1	0	SYSCLK/32	375 KHz
0	1	1	SYSCLK/64	187.5 KHz
1	0	0	SYSCLK/128	93.75 KHz
1	0	1	SYSCLK/256	46.875 KHz
1	1	0	S0TOF/6	可变的
1	1	1	T0OF/6	可变的

Note:

1. The Maximum TWI0/I2C0 clock Rate should under 1MHz, to set SYSCLK = 8MHz to generate 1MHz.
2. SYSCLK is the system clock.
3. S0TOF is UART0 Baud-Rate Generator Overflow.
4. T0OF is Timer 0 Overflow.意:

- SISTA: TWI0/ I2C0 Status Register

SFR Page = 0~F

SFR Address = 0xD3

Bit	7	6	5	4	3	2	1	0
Name	SIS7	SIS6	SIS5	SIS4	SIS3	SIS2	SIS1	SIS0
R/W	R	R	R	R	R	R	R	R
Reset Value	1	1	1	1	1	0	0	0

SISTA is an 8-bit read-only register. The three least significant bits are always 0. The five most significant bits contain the status code. There are a number of possible status codes. When SISTA contains F8H, no serial interrupt is requested. All other SISTA values correspond to defined TWI0/ I2C0 states. When each of these states is entered, a status interrupt is requested (SI=1). A valid status code is present in SISTA when SI is set by hardware.

In addition, state 00H stands for a Bus Error. A Bus Error occurs when a START or STOP condition is present at an illegal position, such as inside an address/data byte or just on an acknowledge bit.

- AUXR3: Auxiliary Register 3

SFR Page = 0 only

SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	T0PS1	T0PS0	BPOC1	BPOC0	S0PS0	TWIPS[1:0]		T0XL

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 2~1: TWIPS1~0, TWI0/ I2C0 Port Selection [1:0].

TWIPS[1:0]	TWI0/I2C0_SCL	TWI0/I2C0_SDA
00	P3.1	P3.0
01	P6.0	P6.1
10	P3.0	P3.1
11	P2.2	P2.4

● **AUXR10:** Auxiliary Register 10

SFR Page = 7 only

SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	SIIPS0	S0PS1	--	TWICF	PAA
R/W	W	W	W	R/W	R/W	W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 1: TWICF, TWI0/I2C0 serial Clock input Filter. 0: Disable TWICF function.

1: Enable TWICF function.

Bit 0: PAA, Pre-Assert Acknowledge. 0: Disable PAA function.

1: Enable PAA function on DMA transfer with TWI0/I2C0 master RX and slave TX/RX.

22 Serial Interface Detection (STWI/SI2C)

The serial interface detection module (SID) is always monitoring the “Start” and “Stop” condition on software two-wire-interface (STWI/SI2C). STWI_SCL is the serial clock signal and STWI_SDA is the serial data signal. If any matched condition is detected, hardware set the flag on STAF and STOF. Software can poll these two flags or set SIDFIE (SFIE.7) to share the interrupt vector on System Flag. And STWI_SCL is located on nINT1 which helps MCU to strobe the serial data by nINT1 interrupt. Software can use these resources to implement a variable TWI slave device.

22.1 SID Structure

Figure 22–1 shows the configuration of STAF and STOF detection, interrupt architecture and event detecting waveform.

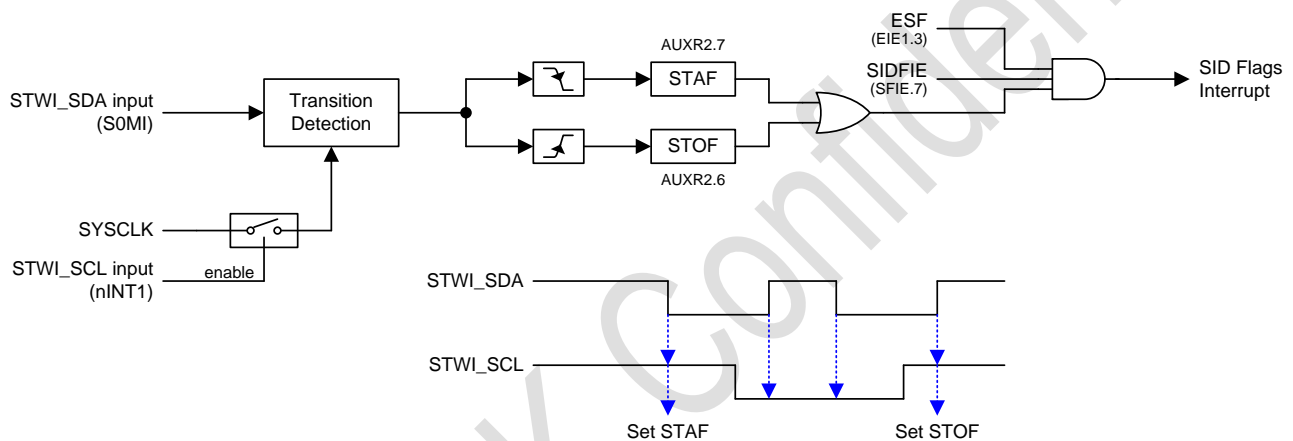


Figure 22-1. Serial Interface Detection structure

22.2 SID Register

- AUXR2: Auxiliary Register 2
SFR Page = 0~F
SFR Address = 0xA3

Bit	7	6	5	4	3	2	1	0
Name	STAF	STOF	--	--	T1X12	T0X12	T1CKOE	T0CKOE
R/W	R/W	R/W	W	W	W	W	W	W
Reset Value	0	0	X	X	0	0	0	0

Bit 7: STAF, Start Flag detection of STWI (SID).

0: Clear by firmware by writing “0” on it. STAF might be held within MCU reset period, so needs to clear STAF in firmware initial.

1: Set by hardware to indicate the START condition occurred on STWI bus.

Bit 6: STOF, Stop Flag detection of STWI (SID). 0: Clear by firmware by writing “0” on it.

1: Set by hardware to indicate the STOP condition occurred on STWI bus. STOF might be held within MCU reset period, so needs to clear STOF in firmware initial.

- SFIE: System Flag Interrupt Enable Register
SFR Page = 0~F

SFR Address = 0x8E

RESET = 0110-x000

Bit	7	6	5	4	3	2	1	0
Name	SIDFIE	--	--	RTCFIE	--	BOF1IE	BOF0IE	WDTFIE
R/W	R/W	R/W	R/W	R/W	W	R/W	R/W	R/W
Reset Value	0	1	1	0	X	0	0	0

Bit 7: SIDFIE, Serial Interface (STWI/SI2C) Detection Flag Interrupt Enabled.

0: Disable SID Flags (STAF or STOF) interrupt.

1: Enable SID Flags (STAF or STOF) interrupt.

23 Beeper

The beeper function outputs a signal on the BEEP pin for sound generation. The signal is in the range about 1, 2 or 4 kHz which is divided from ILRCO. Figure 23–1 shows the beeper generator circuit. But ILRCO is not the precision clock source. Please refer Section “33.5 ILRCO Characteristics” for more detailed ILRCO frequency deviation range.

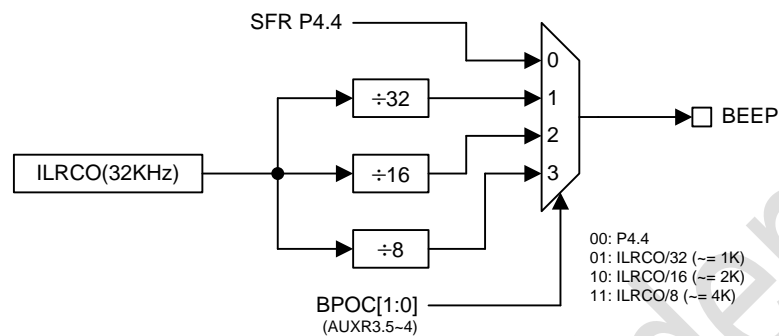


Figure 23-1. Beeper Generator

- **AUXR3:** Auxiliary Register 3
SFR Page = 0 only
SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	T0PS1	T0PS0	BPOC[1:0]		S0PS0	TWIPS1	TWIPS0	T0XL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 5~4: BPOC1~0, Beeper output control bits.

BPOC[1:0]	P4.4 功能	I/O 模式
00	P4.4	By P4M0.4 & P4M1.4
01	ILRCO/32	By P4M0.4 & P4M1.4
10	ILRCO/16	By P4M0.4 & P4M1.4
11	ILRCO/8	By P4M0.4 & P4M1.4

Note:

1. For beeper on P4.4 function, it is recommended to configure P4.4 as push-push output mode.
2. Beeper will use P4.4, and please disable OCD function before enable Beeper function.

- **DCON0:** Device Control 0
SFR Page = P Only
SFR Address = 0x4C

Bit	7	6	5	4	3	2	1	0
Name	HSE	IAPO	HSE1	--	--	IORCTL	RSTIO	OCDE
R/W	R/W	R/W	W	W	W	W	R/W	W
Reset Value	0	0	0	0	0	0	0	0

Bit 0: OCDE, OCD enable.

0: Disable OCD interface on P4.4 and P4.5

1: Enable OCD interface on P4.4 and P4.5.

24 Keypad Interrupt (KBI)

The Keypad Interrupt function is intended primarily to allow a single interrupt to be generated when KBI.7~0 is equal to or not equal to a certain pattern. This function can be used for bus address recognition or keypad recognition.

There are three SFRs used for this function. The Keypad Interrupt Mask Register (KBMASK) is used to define which KBI input pins are enabled to trigger the interrupt. The Keypad Pattern Register (KBPATN) is used to define a pattern that is compared to the value of keypad input. The Keypad Interrupt Flag (KBIF) in the Keypad Interrupt Control Register (KBCON) is set by hardware when the condition is matched. An interrupt will be generated if it has been enabled by setting the EKBI bit in EIE1 register and EA=1. The PATN_SEL bit in the Keypad Interrupt Control Register (KBCON) is used to define “equal” or “not-equal” for the comparison. The keypad input can be assigned on the different port pins, please refer Section “4.3 Alternate Function Redirection” for more detailed information.

In order to use the Keypad Interrupt as the “Keyboard” Interrupt, the user needs to set KBPATN=0xFF and PATN_SEL=0 (not equal), then any key connected to keypad input which is enabled by KBMASK register will cause the hardware to set the interrupt flag KBIF and generate an interrupt if it has been enabled. The interrupt may wake up the CPU from Idle mode or Power-Down mode. This feature is particularly useful in handheld, battery powered systems that need to carefully manage power consumption but also need to be convenient to use.

24.1 KBI Structure

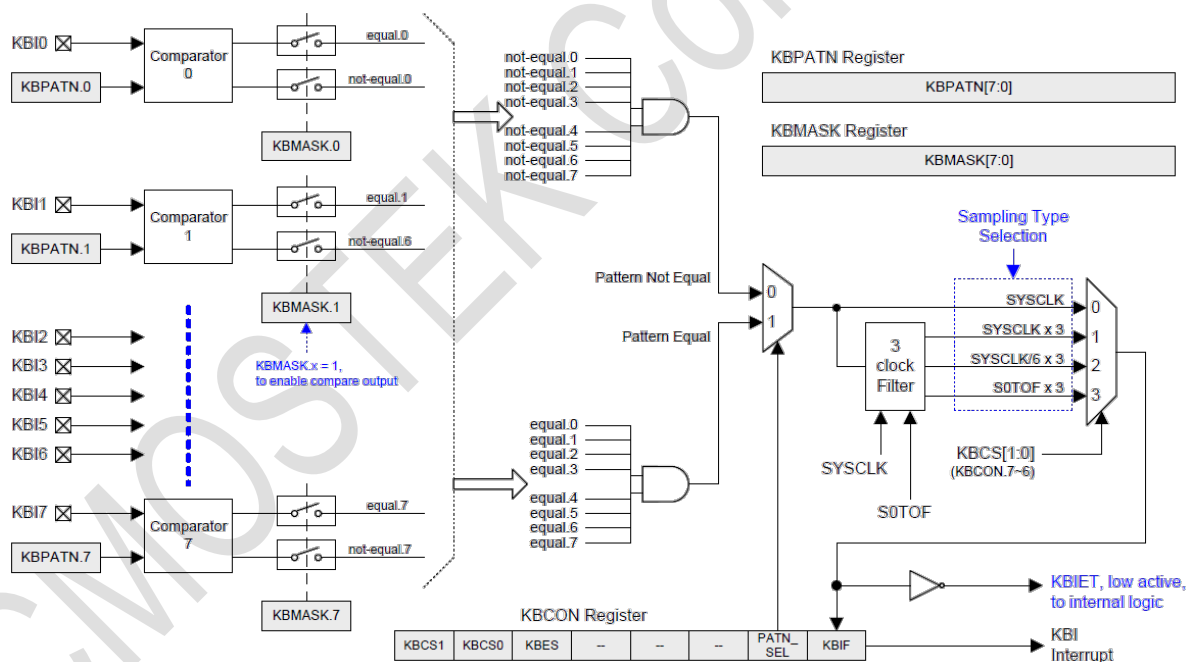


Figure 24-1. Keypad Interrupt (KBI) structure

24.2 KBI Register

The following special function registers are related to the KBI operation:

- KBPATN: Keypad Pattern Register
SFR Page = 0~F
SFR Address = 0xD5

Bit	7	6	5	4	3	2	1	0
Name	KBPATN[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	1	1	1	1	1	1	1	1

Bit 7~0: KBPATN.7~0: The keypad pattern, reset value is 0xFF.

- **KBCON**: Keypad Control Register

SFR Page = 0~F

SFR Address = 0xD6

Bit	7	6	5	4	3	2	1	0
Name	KBCS[1:0]		KBES	--	--	--	PATN_SEL	KBIF
R/W	R/W	R/W	R/W	W	W	W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	1

Bit 7~6: KBCS1~0, KBI Filter mode control.

KBCS[1:0]	KBI input filter mode
00	禁止
01	SYSCLK x3
10	SYSCLK/6 x3
11	S0TOF x3

Bit 5: KBES, KBI Edge mode select.

0: Set KBI module to level detection mode. 1: Set KBI module to edge detection mode.

Bit 1: PATN_SEL, Pattern Matching Polarity selection.

0: The keypad input has to be not equal to user-defined keypad pattern in KBPATN to generate the interrupt. 1: The keypad input has to be equal to the user-defined keypad pattern in KBPATN to generate the interrupt.

Bit 0: KBIF, Keypad Interrupt Flag. The default value of KBIF is set to "1". 0: Must be cleared by software by writing "0".

1: Set when keypad input matches user defined conditions specified in KBPATN, KBMASK, and PATN_SEL.

- **KBMASK**: Keypad Interrupt Mask Register

SFR Page = 0~F

SFR Address = 0xD7

Bit	7	6	5	4	3	2	1	0
Name	KBMASK[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

KBMASK.7: When set, enables KBI7 input as a cause of a Keypad Interrupt.

KBMASK.6: When set, enables KBI6 input as a cause of a Keypad Interrupt.

KBMASK.5: When set, enables KBI5 input as a cause of a Keypad Interrupt.

KBMASK.4: When set, enables KBI4 input as a cause of a Keypad Interrupt.

KBMASK.3: When set, enables KBI3 input as a cause of a Keypad Interrupt.

KBMASK.2: When set, enables KBI2 input as a cause of a Keypad Interrupt.

KBMASK.1: When set, enables KBI1 input as a cause of a Keypad Interrupt.

KBMASK.0: When set, enables KBI0 input as a cause of a Keypad Interrupt.

- **AUXR6**: Auxiliary Register 6

SFR Page = 3 only

SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	KBI4PS1	KB4IPS0	KBI6PS0	KBI2PS0	T3FCS	T2FCS	SnMIPS	S0COPS
R/W	R/W	R/W	R/W	R/W	R/W	W	R/W	R/W
Reset Value	0	0	0	0	0	X	0	0

Bit 7~6: KBI4PS1~0, KBI4~5 Port pin Selection [1:0].

KBI4PS1~0	KBI4	KBI5
00	P3.3	P1.5
01	P3.4	P3.5
10	P6.0	P6.1
11	P1.5	P3.3

Bit 5: KBI6PS0, KBI6~7 Port pin Selection 0.

KBI6PS0	KBI6	KBI7
0	P1.6	P1.7
1	P3.0	P3.1

Bit 4: KBI2PS0, KBI2~3 Port pin Selection 0.

KBI2PS0	KBI2	KBI3
0	P3.0	P3.1
1	P2.2	P2.4

- AUXR8: Auxiliary Register 8

SFR Page = 5 only

SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	POE7	POE6	C0PPS2	--	KBI0PS0	S1COPS	--	--
R/W	R/W	R/W	R/W	W	R/W	R/W	W	W
Reset Value	1	1	0	0	0	0	0	0

Bit 3: KBI0PS0, KBI0~1 Port pin Selection 0.

KBI0PS0	KBI0	KBI1
0	P1.0	P1.1
1	P4.7	P3.3

25 General Purpose Logic (GPL-CRC)

The CMT2380F17 builds in a general purpose logic cyclic redundancy check function with CCITT16 (CRC16 0x1021) polynomial. The CRC accepts a stream of 8-bit data written to the CRC0DI. Its initial value (seed value) is programmable for multi-purpose applications. The 16-bit initial value (seed value) is set to high byte CRC0SH (CRCDS0~1=01) and low byte CRC0SL (CRCDS0~1=00). The result is stored in CRC0RH (CRCDS0~1=01) and CRC0RL (CRCDS0~1=00).

The GPL-CRC has another data path direct from Flash memory by the Flash Auto-Reload Engine to dynamically check the data correctness in the Flash.

The GPL-CRC can also combine the data inverse function. To write the data byte into BOREV register and it will be flipped automatically when read it back from BOREV. The MSB becomes the LSB.

25.1 GPL-CRC Structure

In the normal mode, it needs to set the seed in CRC0SH and CRC0SL and then write the data into CRC0DI to start the conversion.

In the Flash Auto-Reload mode, it needs to keep CRCDS1~0 at "0x11". And follow the steps show in below:

1. To set the start address of the reload sector, this is defined in IFADRH and IFADRL.
2. To set its end-address is combined the IAPLB (7 bits) and 9'b1-1111-1111.
3. Set IFMT register (ISP/IAP Flash Mode) to 0x80 for Flash Auto-Reload mode.
4. Sequentially write 0x46h then 0xB9h to SCMD register to trigger CRC calculation.

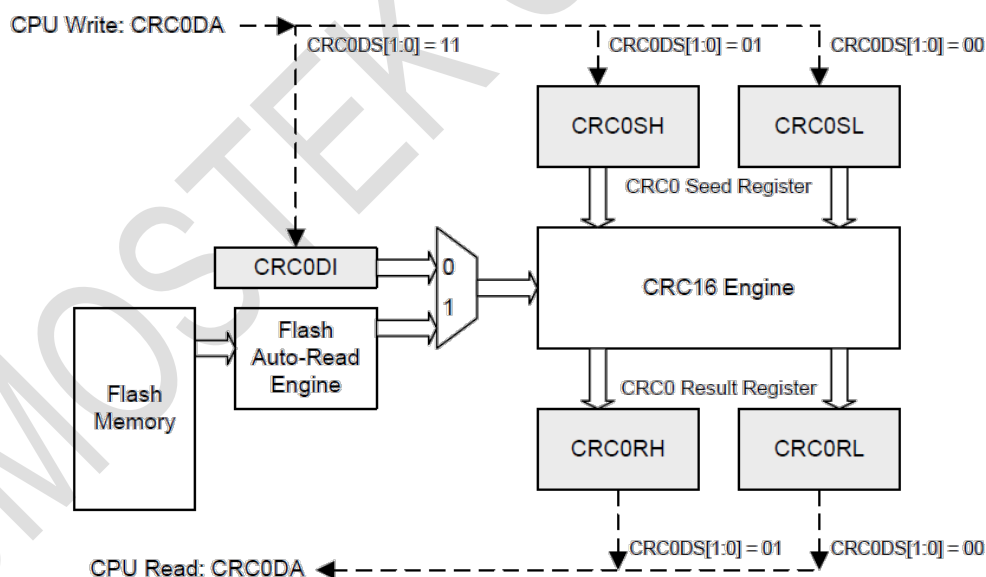


Figure 25-1. CRC structure

25.2 GPL-BOREV Structure

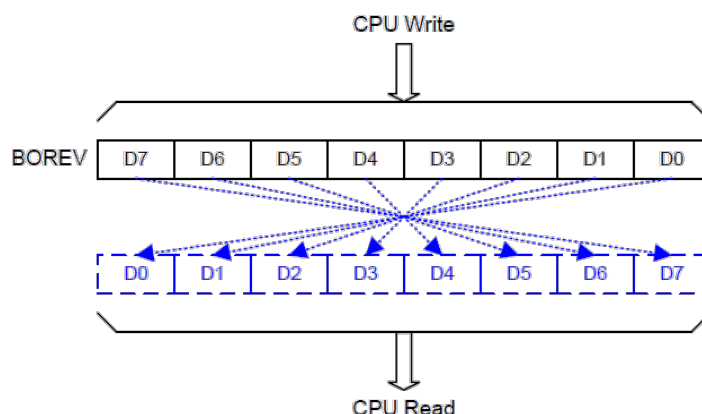


Figure 25–2. GPL-BOREV Structure

25.3 GPL Register

The following special function registers are related to the CRC operation:

- CRC0DA:** CRC0 Data Port
 SFR Page = 0~F
 SFR Address = 0xB6

Bit	7	6	5	4	3	2	1	0
Name	CRC0DA[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~0: CRC0 Data Port. The CRC0 data access is defined as following table:

CRCDS1~0	CPUR/W	CRC0Data Selection	Description
00	Write	CRC0SL	CRC0DataSeedregister-L.
01	Write	CRC0SH	CRC0DataSeedregister-H.
10	Write	--	Reserved.
11	Write	CRC0DI	CRC0DataInputregister.
00	Read	CRC0RL	CRC0Resultregister-L.
01	Read	CRC0RH	CRC0Resultregister-H.
10	Read	--	Reserved.
11	Read	--	Reserved.

- AUXR1:** Auxiliary Control Register 1
 SFR Page = 0~F
 SFR Address = 0xA2

Bit	7	6	5	4	3	2	1	0
Name	--	--	CRCDS1	CRCDS0	--	--	--	DPS
R/W	W	W	R/W	R/W	W	W	W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 5~4: CRCDS1~0. CRC0 Data port Selection bit 1~0.

- BOREV:** Bit Order Reversed data register

SFR Page = 0~F
SFR Address = 0x96

Bit	7	6	5	4	3	2	1	0
Name	BOREV[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~0: BOREV7~0, data read/write for Bit-Order-Reversed function.

Any byte written to BOREV is read back in a bit-reversed order, i.e., the written LSB becomes the MSB.

For example:

If 0xA0 is written to BOREV, the data read back will be 0x05.

If 0x01 is written to BOREV, the data read back will be 0x80.

- **IFMT:** ISP/IAP Flash Mode Table

SFR Page = 0~F
SFR Address = 0xE5

Bit	7	6	5	4	3	2	1	0
Name	MS[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~0: ISP/IAP/Page-P operating mode selection

MS[7:0]	Mode
000 0-00 00	Standby
000 0-00 01	FlashbytereadofAP/IAP-memory
000 0-00 10	FlashbyteprogramofAP/IAP-memory
000 0-00 11	FlashpageeraseofAP/IAP-memory
000 0-01 00	PagePSFRWrite
000 0-01 01	PagePSFRRead
100 0-00 00	AutomaticflashreadforCRC.
100 0-00 01	Flashbyteread withaddressincreasedfunction
100 0-00 10	Flashbyteprogram withaddressincreasedfunction.
Others	Reserved

IFMT is used to select the flash mode for performing numerous ISP/IAP function or to select page P SFR access.

If software selects the mode on automatic flash read for CRC, the flash start-address is defined in IFARDH and IFADRL. The flash end-address is defined at {IAPLB + 9'b1-1111-1111}.

2612-Bit ADC

The ADC subsystem for the CMT2380F17 consists of an analog multiplexer (AMUX), and a 800K sps, 12-bit successive-approximation-register ADC. The AMUX can be configured via the Special Function Registers shown in Figure 26–1. ADC operates in Single-ended mode, and may be configured to measure any of the pins on Port 1 or internal reference. The ADC subsystem is enabled only when the ADEN bit in the ADC Control register (ADCON0) is set to logic 1. The ADC subsystem is in low power shutdown when this bit is logic 0.

26.1 ADC Structure

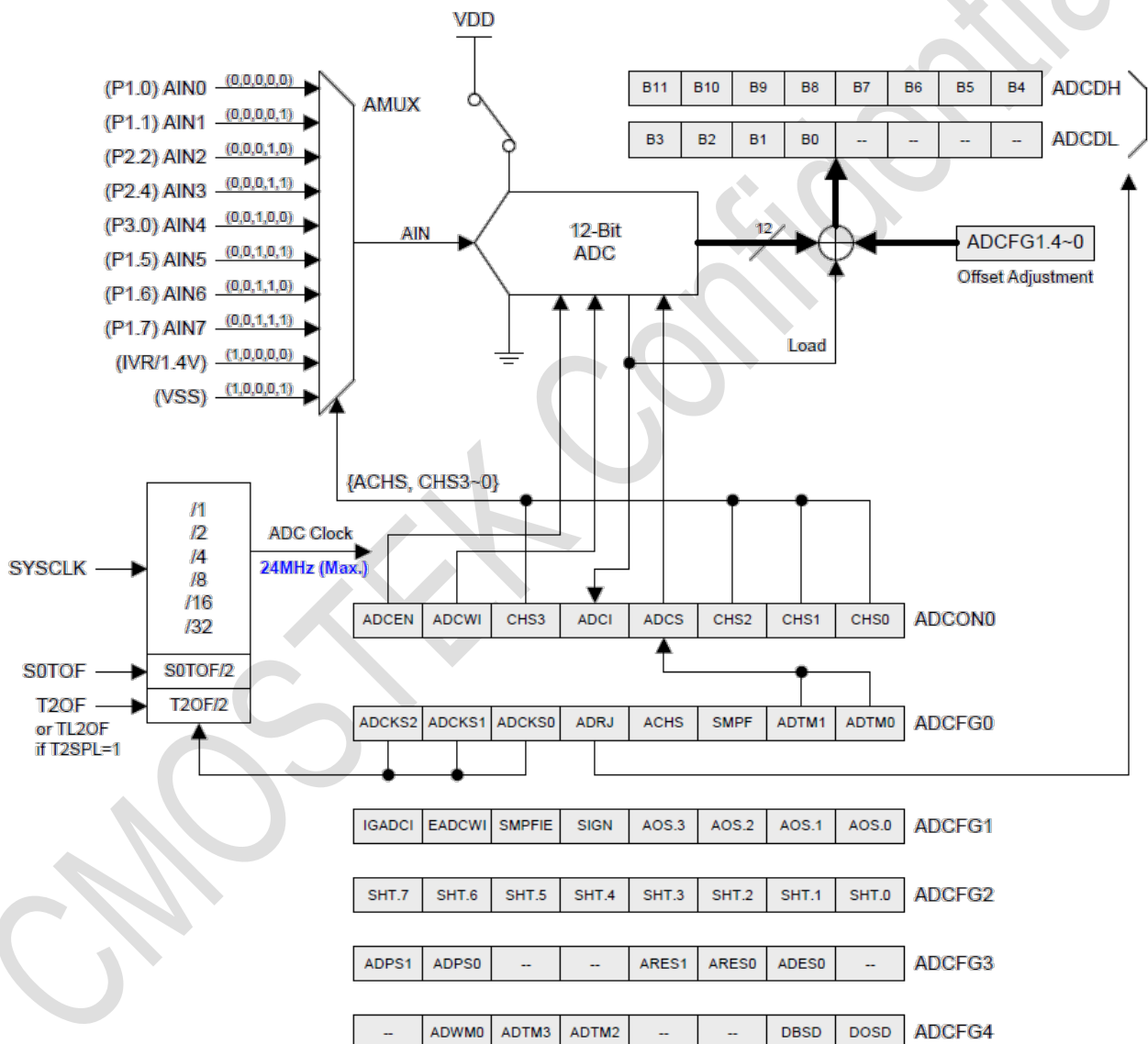


Figure 26-1. ADC Block Diagram

26.2 ADC Operation

ADC has a maximum conversion speed of 800 ksp/s. The ADC conversion clock is a divided version of the system clock, S0 BRG overflow or Timer 2 overflow, determined by the ADCKS2~0 bits in the ADCFG0 register. The ADC conversion clock should be no more than 24 MHz.

After the conversion is complete (ADCI is high), the conversion result can be found in the ADC Result Registers (ADCDH, ADCDL). For single ended conversion, the result is:

$$\text{ADC Result} = \frac{V_{\text{IN}} \times 4096}{\text{VDD Voltage}}$$

26.2.1 ADC Input Channels

The analog multiplexer (AMUX) selects the inputs to the ADC, allowing any of the pins on AIN7~0 to be measured in single-ended mode and one internal voltage reference (IVR, 1.4V). The ADC input channels are configured and selected by CHS3~0 in the ADCON0 register and ACHS in the ADCFG0 register as shown in Figure 26–1. The selected pin is measured with respect to GND.

26.2.2 ADC Internal Voltage Reference

The default ADC reference is VDD. If the VDD is not fixed at a certain voltage, then use the following steps to read voltage:

- 1) To set the analog multiplexer (AMUX) to IVR.
- 2) Convert and store the IVR value by ADC. (Hint: Different VDD voltage will get different IVR read back value, but IVR is fixed at 1.4V. So this read back value can be treated as the reference value.)
- 3) To use the IVR read back reference value to calculate the VDD value. Now the VDD get a certain value, and can be treated as the reference voltage.
- 4) To use the reference voltage converts the input voltage.

26.2.3 Starting a Conversion

Prior to using the ADC function, the user should:

- 1) Turn on the ADC hardware by setting the ADCEN bit,
- 2) Configure the ADC input clock by bits ADCKS2, ADCKS1 and ADCKS0
- 3) Select the analog input channel by bits ACHS, CHS3, CHS2, CHS1 and CHS0
- 4) Configure the selected input (shared with P1) to the Analog-Input-Only mode, and
- 5) Configure ADC result arrangement using ADRJ bit.

Now, user can set the ADCS bit to start the A-to-D conversion. The conversion time is controlled by the bits ADCKS2, ADCKS1 and ADCKS0. Once the conversion is completed, the hardware will automatically clear the ADCS bit, set the interrupt flag ADCI and load the 12 bits of conversion result into ADCDH and ADCDL (according to ADRJ bit) simultaneously. If user sets the ADCS and selects the ADC trigger mode to S0BRG/Timer2 over flow or free-run, then the ADC will keep conversion continuously unless ADCEN is cleared or configure ADC to manual mode.

As described above, the interrupt flag ADCI, when set by hardware, shows a completed conversion. Thus two ways may be used to check if the conversion is completed:

- (1) Always polling the interrupt flag ADCI by software;

(2) Enable the ADC interrupt by setting bits EADC (in EIE1 register) and EA (in IE register), and then the CPU will jump into its Interrupt Service Routine when the conversion is completed. Regardless of (1) or (2), the ADCI flag should be cleared by software before next conversion.使用

26.2.4 ADC Conversion Rate

The user can select the appropriate conversion speed according to the frequency of the analog input signal. The maximum input clock of the ADC is 24MHz and it operates a minimum conversion time with 30 ADC clocks. User can configure the ADCKS2~0 (ADCFG0.7~5), SHT (ADCFG2.7~0) and HA (ADCFG3.5) to specify the conversion rate. The following equation is the clock number of one ADC conversion:

$$\text{ADC Conversion Rate} = \frac{\text{ADC Clock Freq.}}{(30 + X)} \quad ; X = \text{SHT}, 0 \sim 255$$

Please note is the input signal is AC signal, f_N , and assume the sample rate is f_S , based on Nyquist theorem, f_S should large than 2 times f_N to ensure the measurement accuracy.

For example,

1. To get 800K Sample Rate:

If SYSCLK= 24MHz and the ADCKS = SYSCLK is selected, SHT = 0, Then conversion rate $f_S = 24\text{MHz}/(30+0) = 800\text{K}$ sps.

(In this case, the AC input signal f_N frequency should lower than 400KHz to ensure the measurement accuracy.)

2. To get 150K Sample Rate:

If SYSCLK= 24MHz and the ADCKS = SYSCLK/4 is selected, SHT = 10, Then conversion rate $f_S = 24\text{MHz}/4/(30+10) = 150\text{K}$ sps.

(In this case, the AC input signal f_N frequency should lower than 75KHz to ensure the measurement accuracy.)

26.2.5 ADC Interrupts

The ADC interrupt of CMT2380F17 includes 3 sources:

1. ADCI, when an A/D conversion is completed, ADCI will be set to invoke an interrupt. The interrupt on this flag can be blocked by IGADCI (ADCFG1.7).

2. SMPF, it is set when an ADC channel sample & hold is completed to invoke an interrupt. The interrupt on this flag can be blocked by SMPFIE (ADCFG1.5).

3. ADCWI, under ADC Window Compare mode, this Interrupt flag will be held when Window Comparison Data match has occurred. An interrupt is invoked if it is enabled. The interrupt on this flag can be enabled by EADCWI. (ADCFG1.6)有3种ADC中断源:

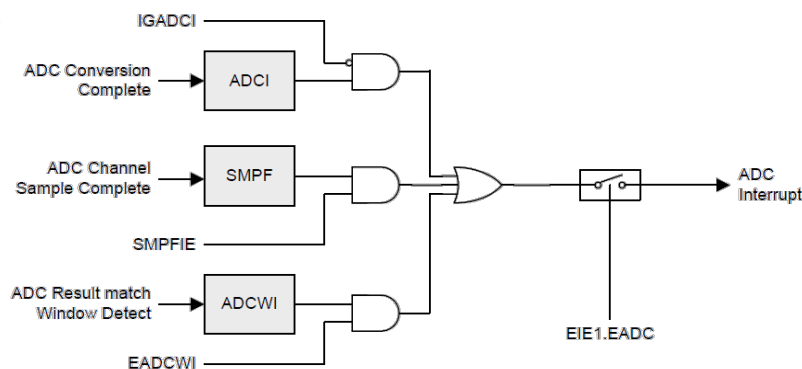


Figure 26–2. ADC Interrupt

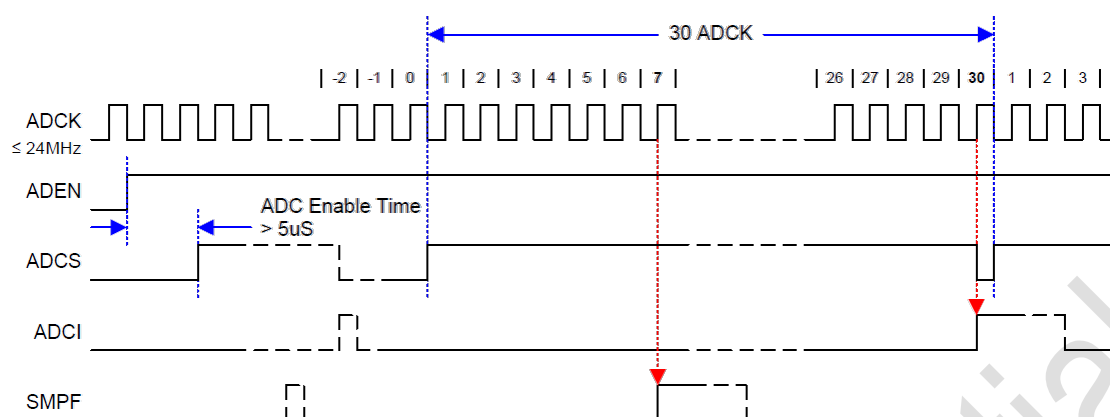


Figure 26-3. ADC Conversion Timing

26.2.6 ADC Window Detect

The CMT2380F17 ADC's programmable window detector continuously compares the ADC output registers with

user-programmed limits, and notifies the system when a desired condition is detected. This is especially effective in an interrupt driven system, saving code space and CPU bandwidth while delivering faster response times. The window detector interrupt flag (ADCWI) can also be used in polled mode. The Window-High-Boundary (WHB[11:0], {ADCFG12, ADCFG11}) and Window-Low-Boundary (WLB[11:0], {ADCFG14, ADCFG13}) registers hold the boundary values. The Window-Boundary flags can be programmed to catch the ADC convert value (ADCDH:ADC DL) when it is inside or outside of the user-defined boundary. The following figure shows the two window detect modes:

1. ADWM0 = 0: When ADC convert value is "inside" the boundary the interrupt flag ADCWI will be held. Which means the condition $WLB[11:0] \leq ADCDH:ADC DL \leq WHB[11:0]$ is true, ADCWI will be held.
2. ADWM0 = 1: When ADC convert value is "outside" the boundary the interrupt flag ADCWI will be held. Which means the condition $WLB[11:0] > ADCDH:ADC DL$ or $ADCDH:ADC DL < WHB[11:0]$ is true, ADCWI will be held.

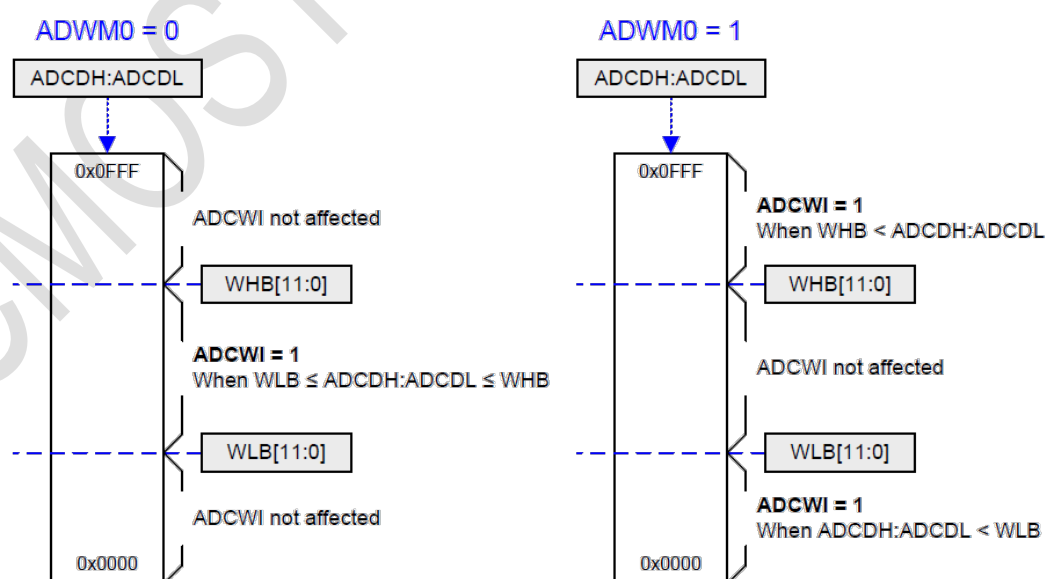


Figure 26-4. ADC Window Detect

Another application of ADC Window Detect is to specify the voltage is larger or less than a specific voltage. For example:

1. The target voltage \geq the condition: ADWM0 = 0, to set condition value in WLB and set WHB = 0xFFFF
2. The voltage less \leq the condition: ADWM0 = 0, to set the condition value in WHB and set WLB = 0
3. The target voltage $>$ the condition: ADWM0 = 1, to set condition value in WHB and set WLB = 0
4. The target voltage $<$ the condition: ADWM0 = 1, to set condition value in WLB and set WHB = 0xFFFF

26.2.7 ADC Channel Scan Mode

CMT2380F17 has 8 channels used as ADC input. If the application needs to watch several voltage by different input pad sequentially, to use ADC Channel Scan Mode can be the easy way to implement and save the channel switch time. To set the ASCE.7 ~ ASCE.0 in ADCFG5 to indicate the input channels, and the channels will be changed to next channel after ADC convert finish. To use this function with different ADC trigger mode to auto switch between the channels. To stop this mode just clear ASCE.7 ~ ASCE.0 to disable this function. When the ADC Channel Scan mode is enabled, please do not write the CHS3~0 manually to change channel, otherwise it will cause unexpected channel to be selected. If you want to clear ADCWI (ADC Window Compare Interrupt flag), you need to read modify write of the ADCON0 to prevent the CHS3 ~ CHS0 to be changed. And please note, when using this mode, the ACHS needs to be "0", to prevent the internal ADC channel be selected.

- 1) Turn on the ADC hardware by setting the ADCEN bit,
- 2) Configuring the ADC input clock by bits ADCKS2, ADCKS1 and ADCKS0
- 3) Configuring the selected input (shared with P1) to the Analog-Input-Only mode
- 4) Configure ADC result arrangement using ADRJ bit.
- 5) Select the analog input channels by setting ASCE.7 ~ ASCE.0 in ADCFG5
- 6) Select ADC Trigger Mode by setting ADTM [3:0]

26.2.8 Transfer ADC Data by DMA

When using ADC with DMA transfer, it needs to check following settings:

1. DMA controller will transfer ADCDL and then ADCDH
2. ADRJ (ADC result Right-Justified selection).
3. ADC Data Resolution: There are 3 ADC data resolution can be selected, 12-bit, 10-bit and 8-bit. To use ARES[1:0] to set the resolution.
4. ADC Data Bit Transfer by DMA: There are 2 different options can be chosen for DMA transfer, 2-byte and 1-byte which is selected by DBSD. When 8-bit is selected, the DMA controller will automatically to detect the register which hold the ADC value to transfer.

For example when ADC Data Resolution is 8 bit mode:

- i. ADRJ = 0 (Left-Justified): The ADC value will store in ADCDH. If DMA transfer mode is 8-bit, then ADCDH will be transferred.
- ii. ADRJ = 1 (Right-Justified): The ADC value will store in ADCDL. If DMA transfer mode is 8-bit, then ADCDL will be transferred.

For example, when VIN = VDD the ADC value is 0xFFFF, with different combinations the ADC Data will be transfer by DMA as following:

ADRJ	ARES[1:0] ADCDataResolution	DBSD ADCDataByteTransferbyDMA	DataTransferOrder	
			1 st Data	2 nd Data
0	00 (12-bit)	0(2-byteData)	0xF0	0xFF
		1(1-byteData)	0xFF	X
	01 (10-bit)	0(2-byteData)	0xC0	0xFF
		1(1-byteData)	0xFF	X
	1x (8-bit)	0(2-byteData)	0x00	0xFF
		1(1-byteData)	0xFF	X
1	00 (12-bit)	0(2-byteData)	0xFF	0x0F
		1(1-byteData)	0xFF	X
	01 (10-bit)	0(2-byteData)	0xFF	0x03
		1(1-byteData)	0xFF	X
	1x (8-bit)	0(2-byteData)	0xFF	0x00
		1(1-byteData)	0xFF	X

26.2.9 I/O Pins Used with ADC Function

The analog input pins used for the A/D converters also have its I/O port's digital input and output function. In order to give the proper analog performance, a pin that is being used with the ADC should have its digital output as disabled. It is done by putting the port pin into the input-only mode. And when an analog signal is applied to the ADCI7~0 pin and the digital input from this pin is not needed, software could set the corresponding pin to analog-input-only mode to reduce power consumption in the digital input buffer. The port pin configuration for analog input function is described in Table 14–3. General Port Configuration Settings and Section "" and "14.2.1 Port 1 Register".

26.2.10 Idle and Power-Down Mode

If the ADC is turned on in Idle mode and Power-Down mode, it will consume a little power. So, power consumption can be reduced by turning off the ADC hardware (ADCEN=0) before entering Idle mode and Power-Down mode.

In Power-Down mode, the ADC does not function. If software triggers the ADC operation in Idle mode, the ADC will finish the conversion and set the ADC interrupt flag, ADCI. When the ADC interrupt enable (EADC, EIE1.1) is set, the ADC interrupt will wake up CPU from Idle mode.

26.2.11 How to improve ADC Accuracy

To use ADC measure the voltage, its accuracy might be affected by many factors, for example, the power noise of the MCU VDD or tolerance of the reference voltage. MG82F6D64/32 has trimmed the internal reference voltage – IVR under VDD equals to 3.3V, and use the ADC to read its ADC value to store in flash ROM as the Pre-stored value. To use this value by following formulas to calculate the AIN voltage instead of measuring VDD to calculate the 1 LSB voltage.

- To push back the IVR voltage (which was measured under VDD=3.3V)

$$\text{IVR Voltage} = \frac{3300 \times \text{IVR}_{\text{ADC_PreStored_Value}}}{4096} (\text{mV})$$

- o use the proportional relationship calculate the I/O pin voltage

$$\text{AIN Voltage} = \frac{\text{IVR Voltage} \times \text{AIN}_{\text{ADC_value}}}{\text{IVR}_{\text{ADC_value}}} (\text{mV})$$

Note: To read the IVR ADC Presorted value please reference 27.3 How to read IVR (1.4V) ADC Prestored value .

26.3 ADC Register

● **ADCON0:** ADC Control Register 0

SFR Page = 0~F

SFR Address = 0xC4

Bit	7	6	5	4	3	2	1	0
Name	ADCEN	ADCWI	CHS3	ADCI	ADCS	CHS[2:0]		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: ADCEN, ADC Enable.

0: Clear to turn off the ADC block.

1: Set to turn on the ADC block. At least 5us ADC enabled time is required before set ADCS.

Bit 6: ADCWI, ADC Window Compare Interrupt flag.

0: ADC0 Window Comparison Data match has not occurred since this flag was last cleared. The flag must be cleared by software.

1: This flag is set when ADC Window Comparison Data match has occurred. An interrupt is invoked if it is enabled. The interrupt on this flag can be enabled by EADCWI. (ADCFG1.6)

Bit 5: CHS3. Combined CH2~0 to select ADC input channel.

Bit 4: ADCI, ADC Interrupt Flag.

0: The flag must be cleared by software.

1: This flag is set when an A/D conversion is completed. An interrupt is invoked if it is enabled. The interrupt on this flag can be blocked by IGADCI (ADCFG1.7).

Bit 3: ADCS. ADC Start of conversion. 0: ADCS cannot be cleared by software.

1: Setting this bit by software starts an A/D conversion. On completion of the conversion, the ADC hardware will clear ADCS and set the ADCI. A new conversion may not be started while either ADCS or ADCI is high.

Bit 2~0: CHS2 ~ CHS1, Input Channel Selection for ADC analog multiplexer.

In Single-ended mode:

ACHS	CHS3~0	SelectedChannel
0	0 0 0 0	AIN0(P1.0)
0	0 0 0 1	AIN1(P1.1)
0	0 0 1 0	AIN2(P2.2)
0	0 0 1 1	AIN3(P2.4)
0	0 1 0 0	AIN4(P3.0)
0	0 1 0 1	AIN5(P1.5)
0	0 1 1 0	AIN6(P1.6)
0	0 1 1 1	AIN7(P1.7)
1	0 0 0 0	Int.VREF(IVR/1.4V)
1	0 0 0 1	AVSS
1	0 0 1 0	Reserved
1	0 0 1 1	Reserved
	Others	Reserved

CMOSTEK Confidential

● **ADCFG0:** ADC Configuration Register 0

SFR Page = 0 Only

SFR Address = 0xC3

Bit	7	6	5	4	3	2	1	0
Name	ADCKS[2:0]			ADRJ	ACHS	SMPF	ADTM1	ADTM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	X	X	0	0

Bit 7~5: ADC Conversion Clock Select bits.

ADCKS[2:0]	ADC Clock Selection
0 0 0	SYSCLK
0 0 1	SYSCLK/2
0 1 0	SYSCLK/4
0 1 1	SYSCLK/8
1 0 0	SYSCLK/16
1 0 1	SYSCLK/32
1 1 0	S0TOF/2
1 1 1	T2OF/2

Note:

1. SYSCLK is the system clock.
2. S0TOF is UART0 Baud-Rate Generator Overflow.
3. T2OF is Timer2 Overflow.

Bit 4: ADRJ, ADC result Right-Justified selection.

0: The most significant 8 bits of conversion result are saved in ADCDH [7:0], while the least significant 2 bits in ADCDL[7:6].

1: The most significant 2 bits of conversion result are saved in ADCDH [1:0], while the least significant 8 bits in ADCDL[7:0].

(1) If ADRJ = 0

● **ADCDH:** ADC Date High Byte Register

SFR Page = 0~F

SFR Address = 0xC6

Bit	7	6	5	4	3	2	1	0
Name	ADCD[11:4]							
R/W	R	R	R	R	R	R	R	R
Reset Value	X	X	X	X	X	X	X	X

- **ADCL:** ADC Data Low Byte Register

SFR Page = 0~F

SFR Address = 0xC5

Bit	7	6	5	4	3	2	1	0
Name	ADCD[3:0]				--	--	--	--
R/W	R	R	R	R	R	R	R	R
Reset Value	X	X	X	X	X	X	X	X

(2) If ADRJ = 1

- **ADCDH**

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	--	ADCD[11:8]			
R/W	R	R	R	R	R	R	R	R
Reset Value	X	X	X	X	X	X	X	X

- **ADCDL**

Bit	7	6	5	4	3	2	1	0
Name	ADCD[7:0]							
R/W	R	R	R	R	R	R	R	R
Reset Value	X	X	X	X	X	X	X	X

When in Single-ended Mode, conversion codes are represented as 12-bit unsigned integers. Inputs are measured from '0' to $VDD(VREF) \times 4095/4096$. Example codes are shown below for both right-justified and left-justified data. Unused bits in the ADCDH and ADCDL registers are set to '0'.

Input Voltage (Single-Ended)	ADCDH:ADCDL (ADRJ = 0)	ADCDH:ADCDL (ADRJ = 1)
$VDD \times 4095/4096$	0xFFFF0	0x0FFF
$VDD \times 2048/4096$	0x8000	0x0800
$VDD \times 1023/4096$	0x4000	0x0400
$VDD \times 512/4096$	0x2000	0x0200
$VDD \times 256/4096$	0x1000	0x0100
$VDD \times 128/4096$	0x0800	0x0080
0	0x0000	0x0000

Bit 3: ACHS, ADC Auxiliary CHannel Select. Decode ACHS and CHS3~0 to select ADC input channel. Bit 2: SMPF, ADC channel sample & hold flag.

0: The flag must be cleared by software.

1: This flag is set when an ADC channel sample & hold is completed. An interrupt is invoked if it is enabled. The interrupt on this flag can be enabled by SMPFIE (ADCFG1.5).

Bit 1~0: ADC Trigger Mode selection.

ADTM[1:0]	ADCConversionStartSelection
00	SetADCS
01	Timer0overflow
10	Freerunningmode
11	S0BRGoverflow

- **ADCFG1:** ADC Configuration Register 1

SFR Page = 1 Only

SFR Address = 0xC3

Bit	7	6	5	4	3	2	1	0
Name	IGADCI	EADCWI	SMPFIE	SIGN	AOS.3	AOS.2	AOS.1	AOS.0

R/W	W	W	W	R/W	R/W	R/W	R/W	R/W
Reset Value	X	X	X	0	0	0	0	0

Bit 7: IGADCI, Ignore ADCI interrupt.

0: Enabled ADCI interrupt. Default is enabled. 1: Disable ADCI interrupt.

Bit 6: EADCWI, ADCWI interrupt enable. 0: Disable ADCWI interrupt.

1: Enable ADCWI interrupt to share the ADC interrupt vector.

Bit 5: SMPFIE, SMPF interrupt enable. 0: Disable SMPF interrupt.

1: Enable SMPF interrupt to share the ADC interrupt vector.

Bit 4~0: SIGN and AOS.3~0. The register value adjusts the ADC result in {ADCDH, ADCDL} for offset cancellation. Software can dynamically collect the ADC offset value. Software can also stores the value in CMT2380F17 IAP zone to use it as a constant parameter for ADC offset cancellation. The following table lists the AD0ROC adjustment value for ADC transfer result.

{Sign,AOS.[3:0]}	Value in {ADCDH, ADCDL}
0_1111	ADCtransfervalue +15
0_1110	ADCtransfervalue +14
.....
0_0010	ADCtransfervalue+2
0_0001	ADCtransfervalue+1
0_0000	ADCtransfervalue+0
1_1111	ADCtransfervalue-1
1_1110	ADCtransfervalue-2
.....
1_0001	ADCtransfervalue-15
1_0000	ADCtransfervalue-16

- ADCFG2: ADC Configuration Register 2

SFR Page = 2 only

SFR Address = 0xC3

Bit	7	6	5	4	3	2	1	0
Name	SHT[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~0: SHT[7:0], extend ADC sample time. The value of SHT is 0~255 ADC clocks.

- ADCFG3: ADC Configuration Register 3

SFR Page = 3 only

SFR Address = 0xC3

Bit	7	6	5	4	3	2	1	0
Name	ADPS1	ADPS0	--	--	ARES1	ARES0	ADES0	--
R/W	R/W	R/W	W	W	R/W	R/W	R/W	W
Reset Value	0	1	0	0	0	0	0	0

Bit 7~6: ADPS1~0, ADC Trigger Mode selection bit 3~2.

ADPS[1:0]	ADCPowerSavingcontrol
0 0	Highpower,highspeed
0 1	Mediumhighpower,mediumhighspeed(default)

1	0	Mediumlowpower,mediumlowspeed
1	1	Lowpower,lowspeed

Bit 5~4: Reserved. Software must write "0" on these bits when ADCFG3 is written.

Bit 3~2: ARES1~0, ADC data Resolution selection bit 1~0.

ARES[1:0]		ADCDataResolutionSelection
0	0	12-bitData
0	1	10-bitData
1	0	8-bitData
1	1	Res.(8-bitData)

Bit 1: ADES0, ADC DMA Event request Selection.

0: Request DMA to service ADC data transfer on ADCI setting. 1: Request DMA to service ADC data transfer on ADCWI setting

Bit 0: Reserved. Software must write "0" on this bit when ADCFG3 is written.

- ADCFG4: ADC Configuration Register 4

SFR Page = 4 only

SFR Address = 0xC3

Bit	7	6	5	4	3	2	1	0
Name	--	ADWM0	ADTM3	ADTM2	--	--	DBSD	--
R/W	W	R/W	R/W	R/W	W	W	R/W	W
ResetValue	0	0	0	0	0	0	0	0

Bit 7: Reserved. Software must write "0" on this bit when ADCFG4 is written. Bit 6: ADWM0. Mode selection of ADC Window Detector.

0: ADCWI will be set when ADCDH: ADCDL value is within the range defined by WHB and WLB.

1: ADCWI will be set when ADCDH: ADCDL value is outside of the range defined by WHB and WLB.

Bit 5~4: ADC Trigger Mode selection bit 3~2.

ADTM[3:0]	ADCConversionStartSelection	Source
000 0	SetADCS	Software
000 1	Timer0overflow(T0OF)	Timer0
001 0	Freerunningmode	ADC
001 1	S0BRGoverflow(S0TOF)	S0BRG
010 0	KBIET	KBI
010 1	INT1ET	nINT1
011 0	INT2ET	nINT2
011 1	Reserved	Reserved
100 0	T2EXES	Timer2
100 1	Reserved	Reserved
101 0	T3EXES	Timer3
101 1	Reserved	Reserved
110 0	PCA0Overflow(C0TOF)	PCA0Counter
110 1	C0CMP6 ^(Note1)	PCA0CH6Compare
111 0	C0CMP6orC0CMP7 ^(Note1)	PCA0CH6/CH7Compare
111 1	Reserved	Reserved

Note1: C0CMPx: ReferenceFigure 17-7 & Figure 17-8

Bit 3~2: Reserved. Software must write "0" on these bits when ADCFG4 is written.

Bit 1: DBSD, ADC Data Bit transfer by DMA.

DBSD	ADCDataByteTransferbyDMA
------	--------------------------

0	2BytesData
1	1ByteData

Bit 0: Reserved. Software must write “0” on this bit when ADCFG4 is written.

- ADCFG5: ADC Configuration Register 5

SFR Page = 5 only

SFR Address = 0xC3

Bit	7	6	5	4	3	2	1	0
Name	ASCE[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~0: AIN7~AIN0 auto-scan enabled.

0: Disable ADC channel auto-scan.

1: Enable ADC channel auto-scan.

- ADCFG11: ADC Configuration Register 11

SFR Page = B only

SFR Address = 0xC3

Bit	7	6	5	4	3	2	1	0
Name	WHB.3	WHB.2	WHB.1	WHB.0	--	--	--	--
R/W	R/W	R/W	R/W	R/W	W	W	W	W
Reset Value	1	1	1	1	1	1	1	1

ADCFG12: ADC Configuration Register 12

SFR Page = C only

SFR Address = 0xC3

Bit	7	6	5	4	3	2	1	0
Name	WHB.11	WHB.10	WHB.9	WHB.8	WHB.7	WHB.6	WHB.5	WHB.4
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	1	1	1	1	1	1	1	1

WHB.9~0: ADC Window High Boundary value.

- ADCFG13: ADC Configuration Register 13

SFR Page = D only

SFR Address = 0xC3

Bit	7	6	5	4	3	2	1	0
Name	WLB.3	WLB.2	WLB.1	WLB.0	--	--	--	--
R/W	R/W	R/W	R/W	R/W	W	W	W	W
Reset Value	1	1	1	1	1	1	1	1

- ADCFG14: ADC Configuration Register 14

SFR Page = E only

SFR Address = 0xC3

Bit	7	6	5	4	3	2	1	0
Name	WLB.11	WLB.10	WLB.9	WLB.8	WLB.7	WLB.6	WLB.5	WLB.4
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	1	1	1	1	1	1	1	1

WLB.9~0: ADC Window Low Boundary value.

- PCON3: Power Control Register 3

SFR Page = P Only

SFR Address = 0x45

Bit	7	6	5	4	3	2	1	0
Name	IVREN							
R/W	R/W	W	W	W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: IVREN, Internal Voltage Reference Enable.

0: Disable on-chip IVR (1.4V).

1: Enable on-chip IVR (1.4V).

Bit 6~0: Reserved. Software must write "0" on these bits when PCON3 is written.

27 Internal Voltage Reference (IVR, 1.4V)

The IVR can be used as the reference voltage of the ADC. The typical output is 1.4V. It can be disabling by IVREN.

27.1 IVR (1.4V) Structure

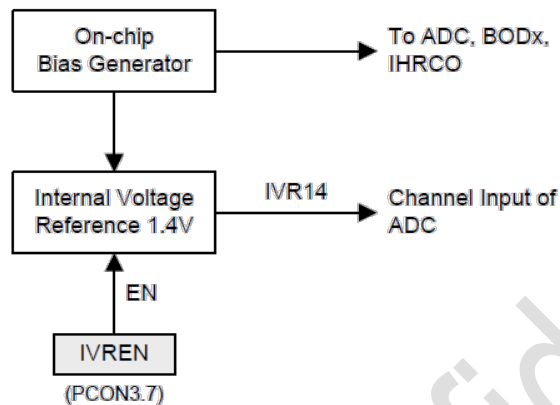


Figure 25-1. IVR Diagram

27.2 IVR Register

- PCON3: Power Control Register 3
SFR Page = P Only
SFR Address = 0x45

Bit	7	6	5	4	3	2	1	0
Name	IVREN	--	--	--	--	--	--	--
R/W	R/W	W	W	W	W	W	W	W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: IVREN, Internal Voltage Reference Enable.

0: Disable on-chip IVR (1.4V).

1: Enable on-chip IVR (1.4V).

Bit 6~0: Reserved. Software must write "0" on these bits when PCON3 is written..

27.3 How to read IVR (1.4V) ADC Prestored value

IVR had been trimmed @VDD=3.3V in factory. And its ADC value had been stored in reserved area in Flash ROM for customer calculation the voltage value from the ADC value. It means customer don't needs to do the calibration of the ADC in the production line. It can save the test time and cost. Please reference the following sample code to read the prestored IVR ADC value. And reference 26.2.11 How to improve ADC Accuracy to understand how to improve ADC measurement accuracy.

```

void Get_Prestored_IVR(void)
{
    ISPCR    = ISP_ENABLE;
    BOREV    = 0x22;
    IFMT     = 0x06;
    IFADRH   = 0x00;
    IFADRL   = 0xC0;
  
```

```
SCMD      = 0x46;
SCMD      = 0xB9;
Trim_IVR_ADC_Value.B[0] = IFD;
IFADRL ++;

SCMD      = 0x46;
SCMD      = 0xB9;
Trim_IVR_ADC_Value.B[1] = IFD;

ISPCR = ISP_DISABLE;
}
```

28ISP and IAP

The flash memory of CMT2380F17 is partitioned into AP-memory, IAP-memory and ISP-memory. AP-memory is used to store user's application program; IAP-memory is used to store the non-volatile application data; and, ISP-memory is used to store the boot loader program for In-System Programming. When MCU is running in ISP region, MCU could modify the AP and IAP memory for software upgraded. If MCU is running in AP region, software could only modify the IAP memory for storage data updated.

28.1 CMT2380F17 Flash Memory Configuration

There are total 16K bytes of Flash Memory in CMT2380F17 and Figure 28–1 shows the device flash configuration of CMT2380F17. The ISP-memory can be configured as disabled or up to 7.5K bytes space by hardware option setting with 0.5KB step. The flash size of IAP memory is located between the IAP low boundary and IAP high boundary. The IAP low boundary is defined by the value of IAPLB register. The IAP high boundary is associated with ISP start address which decides ISP memory size by hardware option. The IAPLB register value is configured by hardware option or AP software programming. All of the AP, IAP and ISP memory are shared the total 16K bytes flash memory.

Note:

(1) ISP Start Address:

0x2200 if ISP Size = 7.5KB

0x2400 if ISP Size = 7KB

.....

0x3000 if ISP Size = 4KB

.....

0x3C00 if ISP Size = 1KB

0x3E00 if ISP Size = 0.5KB

(2) IAP Size :

IAPLB = IAP Low Boundary (ROM High-Byte address)

IAP Start Address = {IAPLB, 00H}

IAP Size = ISP Start Address - IAP Start Address

Set IAPLB = Change IAP Size

(3) If ISP is enabled:

IAP High Boundary = ISP Start Address - 1

IAP Low Boundary = ISP Start Address - IAP Size

(4) If ISP is disabled:

IAP High Boundary = 0x3FFF

IAP Low Boundary = 0x3FFF - IAP Size + 1

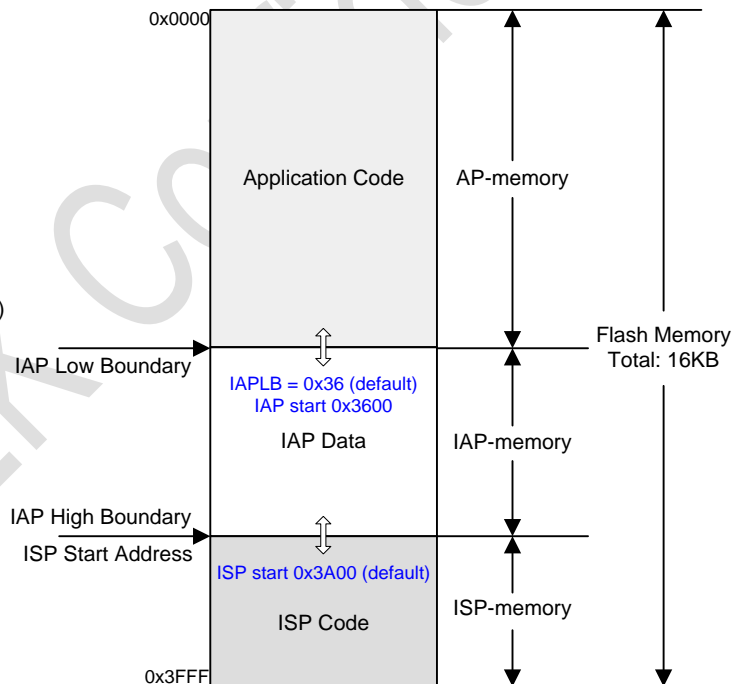


Figure 28-1. CMT2380F17 Flash Flash Memory Configuration

Note:

In default, the CMT2380F17 that Megawin shipped had configured the flash memory for 1.5K ISP, 1K IAP and Lock enabled. The 1.5K ISP region is inserted Megawin proprietary COMBO ISP code to perform In-System-Programming through Megawin 1-Line ISP protocol and COM port ISP. The 1K IAP size can be re-configured by software for application required.

28.2 CMT2380F17 Flash Access in ISP/IAP

There are 3 flash access modes are provided in CMT2380F17 for ISP and IAP application: page erase mode, byte program mode and read mode. MCU software uses these three modes to update new data into flash storage and get flash content. This section shows the flow chart and demo code for the various flash modes.

To do Page Erase (512 Bytes per Page)

Step 1: Set MS[2:0]=[0,1,1] in ISPCR register to select Page Erase Mode. Step 2: Fill page address in IFADRH & IFADRL registers.

Step 3: Sequentially write 0x46h then 0xB9h to SCMD register to trigger an ISP processing.

To do Byte Program

Step 1: Set MS[2:0]=[0,1,0] in ISPCR register to select Byte Program Mode. Step 2: Fill byte address in IFADRH & IFADRL registers.

Step 3: Fill data to be programmed in IFD register.

Step 4: Sequentially write 0x46h then 0xB9h to SCMD register to trigger an ISP processing.

To do Read

Step 1: Set MS[2:0]=[0,0,1] in ISPCR register to select Read Mode. Step 2: Fill byte address in IFADRH & IFADRL registers.

Step 3: Sequentially write 0x46h then 0xB9h to SCMD register to trigger an ISP processing. Step 4: Now, the Flash data is in IFD register.

The detailed descriptions of flash page erase, byte program and flash read in CMT2380F17 is listed in the following sections:

28.2.1 ISP/IAP Flash Page Erase Mode

The any bit in flash data of CMT2380F17 only can be programmed to "0". If user would like to write a "1" into flash data, the flash erase is necessary. But the flash erase in CMT2380F17 ISP/IAP operation only support "page erase" mode, a page erase will write all data bits to "1" in one page. There are 512 bytes in one page of CMT2380F17 and the page start address is aligned to A8~A0 = 0x000. The targeted flash address is defined in IFADRH and IFADRL. So, in flash page erase mode, the IFADRH.0(A8) and IFADRL.7~0(A7~A0) must be written to "0" for right page address selection. Figure 28–2 shows the flash page erase flow in ISPIAP operation.

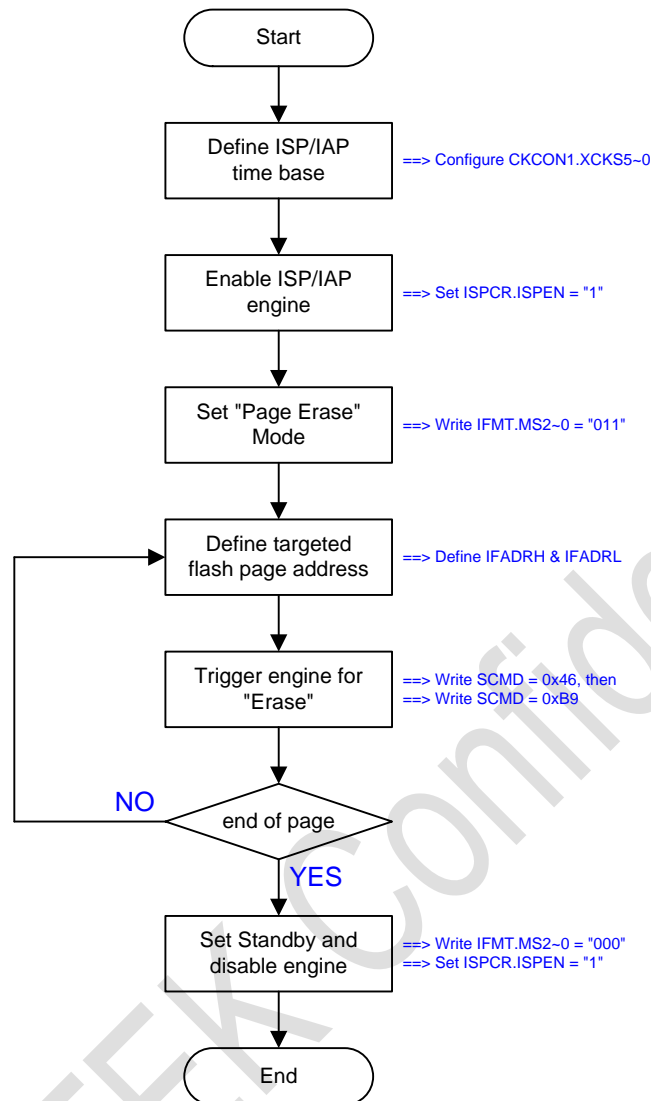


Figure 28-2. ISP/IAP Page Erase Flow

Figure 28-3. Demo Code for ISP/IAP Page Erase

```

MOV    ISPCR,#10000000b ;ISPCR.7 = 1, enable ISP
MOV    IFMT,#03h        ;select Page Erase Mode
MOV    IFADRH,??        ;fill [IFADRH,IFADRL] with page address
MOV    IFADRL,??        ;
MOV    SCMD,#46h        ; trigger ISP/IAPprocessing
MOV    SCMD,#0B9h       ;

;Now, MCU will halt here until processing completed

MOV    IFMT,#00h;select Standby Mode
MOV    ISPCR,#00000000b ;ISPCR.7= 0, disable ISP

```

28.2.2 ISP/IAP Flash Byte Program Mode

The “program” mode of CMT2380F17 provides the byte write operation into flash memory for new data updated. The IFADRH and IFADRL point to the physical flash byte address. IFD stores the content which will be programmed into the flash. Figure 28–4 shows the flash byte program flow in ISP/IAP operation.

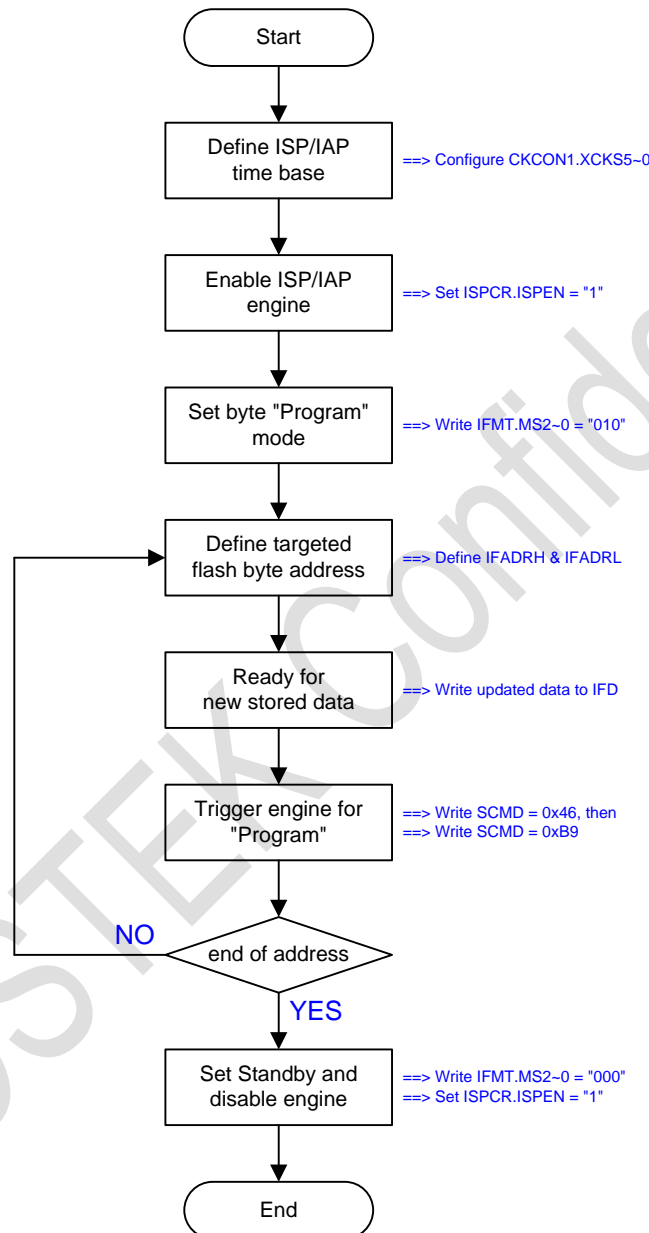


Figure 28-4. ISP/IAP byte Program Flow

Figure 28-5. Demo Code for ISP/IAP Byte Program

```
MOV    ISPCR,#10000011b ; ISPCR.7=1, enable ISP
MOV    IFMT,#02h        ;select Program Mode
MOV    IFADRH,??        ;fill [IFADRH,IFADRL] with byte addres
MOV    IFADRL,??        ;
MOV    IFD,??           ; fill IFD with the data to be programmed
MOV    SCMD,#46h        ;trigger ISP/IAP processing
MOV    SCMD,#0B9h      ;
```

;Now, MCU will halt here until processing completed

```
MOV    IFMT,#00h;select Standby Mode
MOV    ISPCR,#00000000b ;ISPCR.7= 0, disable ISP
```

28.2.3 ISP/IAP Flash Read Mode

The “read” mode of CMT2380F17 provides the byte read operation from flash memory to get the stored data. The IFADRH and IFADRL point to the physical flash byte address. IFD stores the data which is read from the flash content. It is recommended to verify the flash data by read mode after data programmed or page erase.

Figure 28–6 shows the flash byte read flow in ISP/IAP operation.

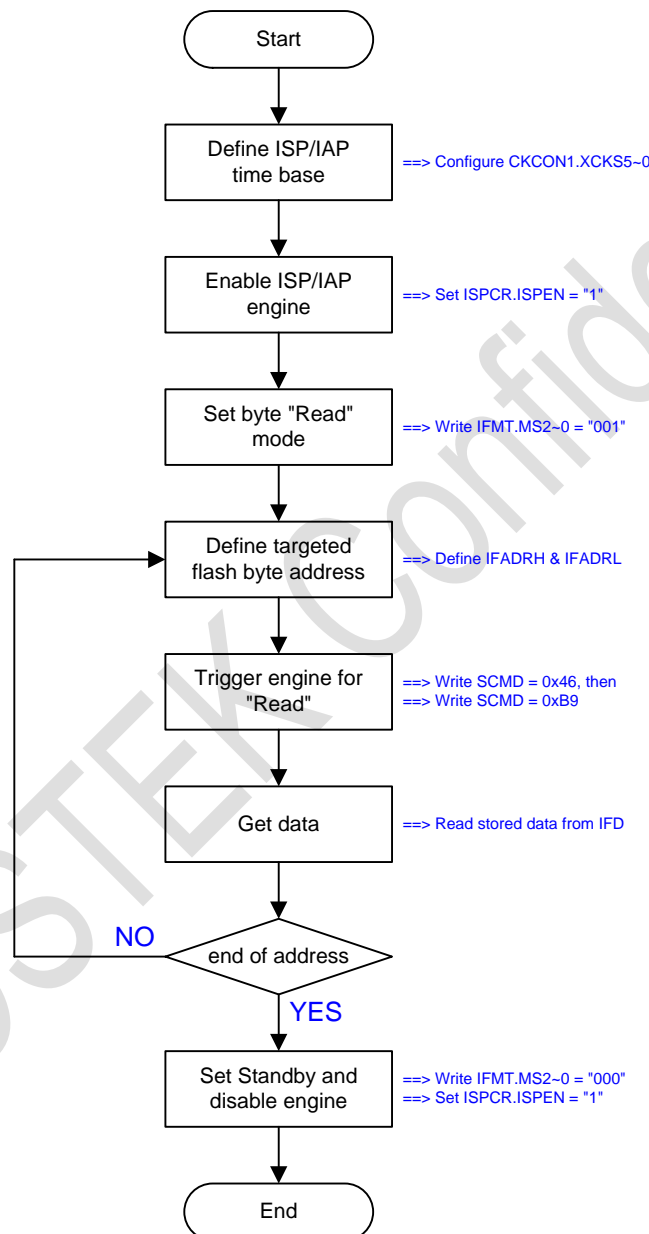


Figure 28-6. ISP/IAP byte Read Flow

Figure 28-7. Demo Code for ISP/IAP byte Read

```

MOV    ISPCR,#10000011b ;ISPCR.7=1, enable ISP
MOV    IFMT,#01h        ;select Read Mode
MOV    IFADRH,??        ;fill [IFADRH,IFADRL] with byte address
MOV    IFADRL,??        ;
MOV    SCMD,#46h        ;fill [IFADRH,IFADRL] with byte address
MOV    SCMD,#0B9h       ;

;Now, MCU will halt here until processing completed

MOV    A,IFD            ;now, the read data exists in IFD
MOV    IFMT,#00h;select Standby Mode
MOV    ISPCR,#00000000b ;ISPCR.7= 0, disableISP

```

28.3 ISP Operation

ISP means In-System-Programming which makes it possible to update the user's application program (in AP-memory) and non-volatile application data (in IAP-memory) without removing the MCU chip from the actual end product. This useful capability makes a wide range of field-update applications possible. The ISP mode is used in the loader program to program both the AP-memory and IAP-memory.

Note:

- (1) Before using the ISP feature, the user should configure an ISP-memory space and pre-program the ISP code (boot loader program) into the ISP-memory by a universal Writer/Programmer or Megawin proprietary Writer/Programmer.
- (2) ISP code in the ISP-memory can only program the AP-memory and IAP-memory.

After ISP operation has been finished, software writes "001" on ISPCR.7 ~ ISPCR.5 which triggers an software RESET and makes CPU reboot into application program memory (AP-memory) on the address 0x0000.

As we have known, the purpose of the ISP code is to program both AP-memory and IAP-memory. Therefore, the MCU must boot from the ISP-memory in order to execute the ISP code. There are two methods to implement In-System Programming according to how the MCU boots from the ISP-memory.

28.3.1 Hardware approached ISP

To make the MCU directly boot from the ISP-memory when it is just powered on, the MCU's hardware options HWBS and ISP Memory must be enabled. The ISP entrance method by hardware option is named hardware approached. Once HWBS and ISP Memory are enabled, the MCU will always boot from the ISP-memory to execute the ISP code (boot loader program) when it is just powered on. The first thing the ISP code should do is to check if there is an ISP request. If there is no ISP requested, the ISP code should trigger a software reset (setting ISPCR.7~5 to "101" simultaneously) to make the MCU re-boot from the AP-memory to run the user's application program..

If the additional hardware option, HWBS2, is enabled with HWBS and ISP Memory, the MCU will always boot from ISP memory after power-on or external reset finished. It provides another hardware approached way to enter ISP mode by external reset signal. After first time power-on, CMT2380F17 can perform ISP operation by external reset trigger and doesn't wait for next time power-on, which suits the non-power-off system to apply the hardware approached ISP function.

28.3.2 Software approached ISP

The software approached ISP to make the MCU boot from the ISP-memory is to trigger a software reset while the MCU is running in the AP-memory. In this case, neither HWBS nor HWBS2 is enabled. The only way for the MCU to boot from the ISP-memory is to trigger a software reset, setting ISPCR.7~5 to "111" simultaneously, when running in the AP-memory. Note: the ISP memory must be configured a valid space by hardware option to reserve ISP mode for software approached ISP application.

Note: ISP storage space must configure a valid space through the hardware option to reserve the ISP mode to start the ISP application by software.

28.3.3 Notes for ISP

- **Developing of the ISP Code**

Although the ISP code is programmed in the ISP-memory that has an ISP Start Address in the MCU's Flash (see Figure 28–1 for CMT2380F17, it doesn't mean you need to put this offset (= ISP Start Address) in your source code. The code offset is automatically manipulated by the hardware. User just needs to develop it like an application program in the AP-memory.

- **Interrupts during ISP**

After triggering the ISP/IAP flash processing, the MCU will halt for a while for internal ISP processing until the processing is completed. At this time, the interrupt will queue up for being serviced if the interrupt is enabled previously. Once the processing is completed, the MCU continues running and the interrupts in the queue will be serviced immediately if the interrupt flag is still active. The user, however, should be aware of the following:

- (1) Any interrupt can not be in-time serviced when the MCU halts for ISP processing.
- (2) The low/high-level triggered external interrupts, nINTx, should keep activated until the ISP is completed, or they will be neglected.

- **ISP and Idle mode**

CMT2380F17 does not make use of idle-mode to perform ISP function. Instead, it freezes CPU running to release the flash memory for ISP/IAP engine operating. Once ISP/IAP operation finished, CPU will be resumed and advanced to the instruction which follows the previous instruction that invokes ISP/AP activity.

- **Accessing Destination of ISP**

As mentioned previously, the ISP is used to program both the AP-memory and the IAP-memory. Once the accessing destination address is beyond that of the last byte of the IAP-memory, the hardware will automatically neglect the triggering of ISP processing. That is the triggering of ISP is invalid and the hardware does nothing.

- **Flash Endurance for ISP**

The endurance of the embedded Flash is 20,000 erase/write cycles, that is to say, the erase-then-write cycles shouldn't exceed 20,000 times. Thus the user should pay attention to it in the application which needs to frequently update the AP-memory and IAP-memory.

28.4 In-Application-Programming (IAP)

The CMT2380F17 has built a function as In Application Programmable (IAP), which allows some region in the Flash memory to be used as non-volatile data storage while the application program is running. This useful feature can be applied to the application where the data must be kept after power off. Thus, there is no need to use an external serial EEPROM (such as 93C46, 24C01, .., and so on) for saving the non-volatile data.

In fact, the operating of IAP is the same as that of ISP except the Flash range to be programmed is different. The programmable Flash range for ISP operating is located within the AP and IAP memory, while the range for IAP operating is only located within the configured IAP-memory.

Note:

(1) For CMT2380F17 IAP feature, the software should specify an IAP-memory space by writing IAPLB in IFMT defined. The IAP-memory space can be also configured by a universal Writer/Programmer or Megawin proprietary Writer/Programmer which configuration is corresponding to IAPLB initial value.

(2) The program code to execute IAP is located in the AP-memory and just only program IAP-memory not ISP-memory.

28.4.1 IAP-memory Boundary/Range

If ISP-memory is specified, the range of the IAP-memory is determined by IAP and the ISP starts address as listed below.

- IAP high boundary = ISP start address - 1.
- IAP low boundary = ISP start address - IAP.

If ISP-memory is not specified, the range of the IAP-memory is determined by the following formula.

- IAP high boundary = 0x3FFF.
- IAP low boundary = 0x3FFF - IAP + 1.

For example, if ISP-memory is 1K, so that ISP start address is 0x3C00, and IAP-memory is 1K, then the IAP-memory range is located at 0x3800 ~ 0x3BFF. The IAP low boundary in CMT2380F17 is defined by IAPLB register which can be modified by software to adjust the IAP size in user's AP program.

28.4.2 Update data in IAP-memory

The special function registers are related to ISP/IAP would be shown in Section "28.5 ISP/IAP Register".

Because the IAP-memory is a part of Flash memory, only Page Erase, no Byte Erase, is provided for Flash erasing. To update "one byte" in the IAP-memory, users can not directly program the new datum into that byte. The following steps show the proper procedure:

Step 1: Save the whole page flash data (with 512 bytes) into XRAM buffer which contains the data to be updated. Step 2: Erase this page (using ISP/IAP Flash Page Erase mode).

Step 3: Modify the new data on the byte(s) in the XRAM buffer.

Step 4: Program the updated data out of the XRAM buffer into this page (using ISP/IAP Flash Program mode). To read the data in the IAP-memory, users can use the ISP/IAP Flash Read mode to get the targeted data.

28.4.3 Notes for IAP

Interrupts during IAP

After triggering the ISP/IAP flash processing for In-Application Programming, the MCU will halt for a while for internal IAP processing until the processing is completed. At this time, the interrupt will queue up for being serviced if the interrupt is enabled previously. Once the processing is completed, the MCU continues running and the interrupts in the queue will be serviced immediately if the interrupt flag is still active. Users, however, should be aware of the following:

- (1) Any interrupt can not be in-time serviced during the MCU halts for IAP processing.
- (2) The low/high-level triggered external interrupts, nINTx, should keep activated until the IAP is completed, or they will be neglected.

IAP and Idle mode

CMT2380F17 does not make use of idle-mode to perform IAP function. Instead, it freezes CPU running to release the flash memory for ISP/IAP engine operating. Once ISP/IAP operation finished, CPU will be resumed and advanced to the instruction which follows the previous instruction that invokes ISP/AP activity.

Accessing Destination of IAP

As mentioned previously, the IAP is used to program only the IAP-memory. Once the accessing destination is not within the IAP-memory, the hardware will automatically neglect the triggering of IAP processing. That is the triggering of IAP is invalid and the hardware does nothing.

An Alternative Method to Read IAP Data

To read the Flash data in the IAP-memory, in addition to using the Flash Read Mode, the alternative method is using the instruction "MOVC A, @A+DPTR". Where, DPTR and ACC are filled with the wanted address and the offset, respectively. And, the accessing destination must be within the IAP-memory, or the read data will be indeterminate. Note that using 'MOVC' instruction is much faster than using the Flash Read Mode.

Flash Endurance for IAP

The endurance of the embedded Flash is 20,000 erase/write cycles, that is to say, the erase-then-write cycles shouldn't exceed 20,000 times. Thus the user should pay attention to it in the application which needs to frequently update the IAP-memory.s

28.5 ISP/IAP Register

The following special function registers are related to the access of ISP, IAP and Page-P SFR:

- **IFD:** ISP/IAP Flash Data Register
SFR Page = 0~F
SFR Address = 0xE2

Bit	7	6	5	4	3	2	1	0
Name	IFD[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	1	1	1	1	1	1	1	1

IFD is the data port register for ISP/IAP/Page-P operation. The data in IFD will be written into the desired address in operating ISP/IAP/Page-P write and it is the data window of readout in operating ISP/IAP read.

- IFADRH: ISP/IAP Address for High-byte addressing

SFR Page = 0~F

SFR Address = 0xE3

Bit	7	6	5	4	3	2	1	0
Name	IFADRH[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

IFADRH is the high-byte address port for all ISP/IAP modes. It is not defined in Page-P mode.

- IFADRL: ISP/IAP Address for Low-byte addressing

SFR Page = 0~F

SFR Address = 0xE4

Bit	7	6	5	4	3	2	1	0
Name	IFADRL[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

IFADRL is the low byte address port for all ISP/IAP/Page-P modes. In flash page erase operation, it is ignored.

- IFMT: ISP/IAP Flash Mode Table

SFR Page = 0~F

SFR Address = 0xE5

Bit	7	6	5	4	3	2	1	0
Name	MS[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	X	X	X	X	X	0	0	0

Bit 7~4: Reserved. Software must write "0000_0" on these bits when IFMT is written.

Bit 3~0: ISP/IAP/Page-P operating mode selection.

MS[7:0]	Mode
000 0-00 00	Standby
000 0-00 01	Flashbyte read of AP/IAP-memory
000 0-00 10	Flashbyte program of AP/IAP-memory
000 0-00 11	Flashpage erase of AP/IAP-memory
000 0-01 00	Page PSFR Write
000 0-01 01	Page PSFR Read
100 0-00 00	Automatic flash read for CRC.
100 0-00 01	Flashbyte read with address increased function
100 0-00 10	Flashbyte program with address increased function.
Others	Reserved

IFMT is used to select the flash mode for performing numerous ISP/IAP function or to select page P SFR access.

If software selects the mode on automatic flash read for CRC, the flash start-address is defined in IFARDH and IFADRH. The flash end-address is defined at {IAPLB + 9'b1-1111-1111}.

- SCMD: Sequential Command Data register

SFR Page = 0~F

SFR Address = 0xE6

Bit	7	6	5	4	3	2	1	0
Name	SCMD[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	X	X	X	X	X	X	X	X

SCMD is the command port for triggering ISP/IAP/Page-P activity. If SCMD is filled with sequential 0x46h, 0xB9h and if ISPCR.7 = 1, ISP/IAP/Page-P activity will be triggered.

- ISPCR: SP Control Register

SFR Page = 0~F

SFR Address = 0xE7

Bit	7	6	5	4	3	2	1	0
Name	ISPEN	SWBS	SWRST	CFAIL	--	--	--	--
R/W	R/W	R/W	R/W	R/W	W	W	W	W
Reset Value	0	0	0	0	X	X	X	X

Bit 7: ISPEN, ISP/IAP/Page-P operation enable.

0: Global disable all ISP/IAP/Page-P program/erase/read function. 1: Enable ISP/IAP/Page-P program/erase/read function.

Bit 6: SWBS, software boot selection control. 0: Boot from main-memory after reset.

1: Boot from ISP memory after reset.

Bit 5: SWRST, software reset trigger control. 0: No operation

1: Generate software system reset. It will be cleared by hardware automatically.

Bit 4: CFAIL, Command Fail indication for ISP/IAP operation. 0: The last ISP/IAP command has finished successfully.

1: The last ISP/IAP command fails. It could be caused since the access of flash memory was inhibited. Bit 3~0: Reserved. Software must write "0" on these bits when ISPCR is written.

IAPLB: IAP Low Boundary

SFR Page = P Only

SFR Address = 0x03

Bit	7	6	5	4	3	2	1	0
Name	IAPLB[7:0]							
R/W	W	W	W	W	W	W	W	W
Reset Value	0	1	1	1	0	0	0	X

Bit 7~0: The IAPLB determines the IAP-memory lower boundary. Since a Flash page has 512 bytes, the IAPLB must be an even number.

To read IAPLB, MCU need to define the IFMT for mode selection on IAPLB Read and set ISPCR.ISPEN. And then write 0x46h & 0xB9h sequentially into SCMD. The IAPLB content is available in IFD. If write IAPLB, MCU will put new IAPLB setting value in IFD firstly. And then select IFMT, enable ISPCR.ISPEN and then set SCMD. The IAPLB content has already finished the updated sequence.

The range of the IAP-memory is determined by IAPLB and the ISP start address as listed below.

IAP lower boundary = IAPLB[7:0] x 256, and

IAP higher boundary = ISP start address – 1.

For example, if IAPLB=0x20 and ISP start address is 0x3000, then the IAP-memory range is located at 0x2000 ~ 0x2FFF.

Additional attention point, the IAP low boundary address must not be higher than ISP start address.

28.6 ISP/IAP Sample Code

The following Figure 28–8 shows a sample code for ISP operation.

```

;*****
; Demo Program for the ISP
;*****
IFD      DATA    0E2h
IFADRH   DATA    0E3h
IFADRL   DATA    0E4h
IFMT     DATA    0E5h
SCMD     DATA    0E6h
ISPCR    DATA    0E7h
;
; MOV    ISPCR,#10000000b ;ISPCR.7=1, enable ISP
;=====
;1.PageEraseMode(512bytesperpage)
;=====
; ORL    IFMT,#03h      ;MS[2:0]=[0,1,1], select Page Erase Mode
; MOV    IFADRH,??      ;fill page address in IFADRH & IFADRL
; MOV    IFADRL,??      ;
; MOV    SCMD,#46h      ;trigger ISP processingMOV    SCMD,#0B9h;
;Nowinprocessing...(CPUwillhalthereuntilcomplete)
;=====
;2.ByteProgramMode
;=====

```

```

    ORL    IFMT,#02h      ;MS[2:0]=[0,1,0], select Byte Program Mode
    ANL    ISPCR,#0FAh;
    MOV    IFADRH,??      ;fill byte address in IFADRH & IFADRL
    MOV    IFADRL,??      ;
    MOV    IFD,??         ;fill the data to be programmed in IFD
    MOV    SCMD,#46h      ;triggerISP processing
    MOV    SCMD,#0B9h;
    ;Now in processing...(CPU will halt here until complete)

```

```

;=====
;3.Verify using Read Mode
;=====

```

```

    ANL    IFMT,#0F9h      ;MS1[2:0]=[0,0,1],select Byte Read Mode
    ORL    IFMT,#01h      ;
    MOV    IFADRH,??      ;fill byte address in IFADRH & IFADRL
    MOV    IFADRL,??      ;
    MOV    SCMD,#46h      ;triggerISP processing
    MOV    SCMD,#0B9h      ;
    ;Now in processing...(CPU will halt here until complete)
    MOV    A,IFD          ;data will be in IFD
    CJNE   A,wanted,ISP_error;compare with the wanted value

```

```

    ...
ISP_error:
    ...
;

```


29 Page P SFR Access

CMT2380F17 builds a special SFR page (Page P) to store the control registers for MCU operation. These SFRs can be accessed by the ISP/IAP operation with different IFMT. In page P access, IFADRH must set to "00" and IFADRL indexes the SFR address in page P. If IFMT= 04H for Page P writing, the content in IFD will be loaded to the SFR in IFADRL indexed after the SCMD triggered. If IFMT = 05H for Page P reading, the content in IFD is stored the SFR value in IFADRL indexed after the SCMD triggered.

Following descriptions are the SFR function definition in Page P:

- IAPLB: IAP Low Boundary

SFR Page = P

SFR Address = 0x03

Bit	7	6	5	4	3	2	1	0
Name	IAPLB[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	1	1	1	1	1	1	1	0

Bit7~0: The IAPLB determines the IAP-memory lower boundary. Since a Flash page has 512 bytes, the IAPLB must be an even number.

To read IAPLB, MCU need to define the IFADRL for SFR address in Page-P, the IFMT for mode selection on Page-P Read and set ISPCR.ISPEN. And then write 0x46h & 0xB9h sequentially into SCMD. The IAPLB content is available in IFD. If write IAPLB, MCU will put new IAPLB setting value in IFD firstly. And index IFADRL, select IFMT, enable ISPCR.ISPEN and then set SCMD. The IAPLB content has already finished the updated sequence.

The range of the IAP-memory is determined by IAPLB and the ISP start address as listed below.

IAP lower boundary = IAPLB x 256, and

IAP higher boundary = ISP start address - 1.

For example, if IAPLB=0xE0 and ISP start address is 0xF000, then the IAP-memory range is located at 0xE000 ~0xEFFF.

Additional attention point, the IAP low boundary address must not be higher than ISP start address.

- CKCON2: Clock Control Register 2

SFR Page = P Only

SFR Address = 0x40

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	IHRCOE	MCKS1	MCKS0	OSCS1	OSCS0
R/W	W	W	W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	1	0	0	0	0

Bit 4: IHRCOE, Internal High frequency RC Oscillator Enable. 0: Disable internal high frequency RC oscillator.

1: Enable internal high frequency RC oscillator. If this bit is set by CPU software, it needs 32 us to have stable output after IHRCOE is enabled.

Bit 3~2: MCKS[1:0], MCK Source Selection.

MCKS[1:0]	MCK 时钟源选择	OSCI _n = 12MHz CKMIS = [01]	OSCI _n = 11.059MHz CKMIS = [01]
-----------	-----------	---	---

		CKMS0=0	CKMS0=1	CKMS0=0	CKMS0=1
0 0	OSCIn	12MHz		11.059MHz	
0 1	CKMI x4/x6	24MHz	36MHz	22.118MHz	33.177MHz
1 0	CKMI x5.33/x8	32MHz	48MHz	29.491MHz	44.236MHz
1 1	CKMI x8/x12	48MHz	72MHz	44.236MHz	66.354MHz

Note: It needs to set ENCKM = 1 to enable CKM.

Note: Needs to be careful of the limitation of CPUCLK and SYSCLK. Needs to use SCKS[2:0] and CCKS to choose proper range of CPUCLK and SYSCLK to not exceed the limitation. CPUCLK \leq 36MHz, SYSCLK \leq 50MHz.

Bit 1~0: OSCS[1:0], OSCIn Source selection.

OSCS[1:0]	OSCIn source Selection
0 0	IHRCO
0 1	ECKI
1 0	ILRCO
1 1	Reserved

- CKCON3: Clock Control Register 3

SFR Page = P

SFR Address = 0x41

Bit	7	6	5	4	3	2	1	0
Name	WDTCS1	WDTCS0	FWKP	WDTFS	MCKD1	MCKD0	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	W	R/W
Reset Value	0	0	0	0	0	0	1	0

Bit 7~6: WDTCS1~0, WDT Clock Source selection [1:0].

WDTCS1~0	WDT 时钟源
00	ILRCO
01	ECKI
10	SYSCLK/12
11	S0TOF

Bit 5: FWKP, MCU Fast wake up control.

0: Select MCU for normal wakeup time about 120us from power-down mode. 1: Select MCU for fast wakeup time about 30us from power-down mode.

Bit 4: WDTFS, WDT overflow source selection.

0: Select WDT bit-8 overflow as WDT event source. 1: Select WDT bit-0 overflow as WDT event source.

Bit 3~2: MCKD[1:0], MCK Divider Output selection.

MCKD[1:0]	MCKDOFrequency	ifMCK =12MHz	ifMCK =48MHz
0 0	MCKDO = MCK	MCKDO = 12MHz	MCKDO = 48MHz
0 1	MCKDO = MCK/2	MCKDO = 6MHz	MCKDO = 24MHz
1 0	MCKDO = MCK/4	MCKDO = 3MHz	MCKDO = 12MHz
1 1	MCKDO = MCK/8	MCKDO = 1.5MHz	MCKDO = 6MHz

Bit 1~0: Reserved. In case of writing CKCON3, these two bits must be written to 0" by software.

- CKCON4: Clock Control Register 4

SFR Page = P only

SFR Address = 0x42

Bit	7	6	5	4	3	2	1	0
Name	RCSS2	RCSS1	RCSS0	RPSC2	RPSC1	RPSC0	RTCCS3	RTCCS2
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~5: RTC Clock Source selection [2:0]

RCSS[2:0]	RTC Clock Selection
0 0 0	ECKI(P6.0)
0 0 1	ILRCO
0 1 0	WDTPS
0 1 1	WDT0F
1 0 0	SYSCLK
1 0 1	SYSCLK/12
1 1 0	Reserved
1 1 1	Reserved

- CKCON5: Clock Control Register 5

SFR Page = P only

SFR Address = 0x43

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	--	--	--	--	CKMS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 0: CKMS0, CKM mode selection 0.

0: Select CKM operating for 8X mode. (96MHz)

1: Select CKM operating for 12X mode. (144MHz)

- PCON2: Power Control Register 2

SFR Page = P Only

SFR Address = 0x44

Bit	7	6	5	4	3	2	1	0
Name	AWBOD1	--	BO1S1	BO1S0	BO1RE	EBOD1	BO0RE	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	1	1	0	1	0	1

Bit 7: AWBOD1, Awaked BOD1 in PD mode. 0: BOD1 is disabled in power-down mode.

1: BOD1 keeps operation in power-down mode.

Bit 6: Reserved. Software must write "0" on this bit when PCON2 is written.

Bit 5~4: BO1S[1:0]. Brown-Out detector 1 monitored level Selection. The initial values of these two bits are loaded from OR1.BO1S10 and OR1.BO1S00.

BO1S[1:0]	BOD1 detecting level
0 0	2.0V
0 1	2.4V
1 0	3.7V
Others	Not allowed

Bit 3: BO1RE, BOD1 Reset Enabled.

0: Disable BOD1 to trigger a system reset when BOF1 is set. 1: Enable BOD1 to trigger a system reset when BOF1 is set.

Bit 2: EBOD1, Enable BOD1 that monitors VDD power dropped at a BO1S1~0 specified voltage level. 0: Disable BOD1 to slow down the chip power consumption. 1: Enable BOD1 to monitor VDD power dropped.

Bit 1: BO0RE, BOD0 Reset Enabled.

0: Disable BOD0 to trigger a system reset when BOF0 is set.

1: Enable BOD0 to trigger a system reset when BOF0 is set (VDD meets 1.7V).

Bit 0: Reserved. Software must write "1" on this bit when PCON2 is written.

- **PCON3:** Power Control Register 3

SFR Page = P Only

SFR Address = 0x45

Bit	7	6	5	4	3	2	1	0
Name	IVREN	--	--	--	--	--	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: IVREN, Internal Voltage Reference Enable. 0: Disable on-chip IVR (1.4V).

1: Enable on-chip IVR (1.4V).

Bit 6~0: Reserved. Software must write "0" on these bits when PCON3 is written.

- **SPCON0:** SFR Page Control 0

SFR Page = P Only

SFR Address = 0x488

Bit	7	6	5	4	3	2	1	0
Name	--	P6CTL	P4CTL	WRCTL	--	CKCTL0	PWCTL1	PWCTL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: Reserved. Software must write "0" on this bit when SPCON is written. Bit 6: P6CTL. P6 SFR access Control.

If P6CTL is set, it will disable the P6 SFR modified in Page 0~F. P6 in Page 0~F only keeps the SFR read function. But software always owns the modification capability in SFR Page P.

Bit 5: P4CTL. P4 SFR access Control.

If P4CTL is set, it will disable the P4 SFR modified in Page 0~F. P4 in Page 0~F only keeps the SFR read function. But software always owns the modification capability in SFR Page P.

Bit 4: WRCTL. WDTCSR SFR access Control.

If WRCTL is set, it will disable the WDTCSR SFR modified in Page 0~F. WDTCSR in Page 0~F only keeps the SFR read function. But software always owns the modification capability in SFR Page P.

Bit 2: CKCTL0. CKCON0 SFR access Control.

If CKCTL0 is set, it will disable the CKCON0 SFR modified in Page 0~F. CKCON0 in Page 0~F only keeps the SFR read function. But software always owns the modification capability in SFR Page P.

Bit 1: PWCTL1. PCON1 SFR access Control.

If PWCTL1 is set, it will disable the PCON1 SFR modified in Page 0~F. PCON1 in Page 0~F only keeps the SFR read function. But software always owns the modification capability in SFR Page P.

Bit 0: PWCTL0. PCON0 SFR access Control.

If PWCTL0 is set, it will disable the PCON0 SFR modified in Page 0~F. PCON0 in Page 0~F only keeps the SFR read function. But software always owns the modification capability in SFR Page P.

● DCON0: Device Control 0

SFR Page = P Only

SFR Address = 0x4C C

Bit	7	6	5	4	3	2	1	0
Name	HSE	IAP0	HSE1	--	--	IORCTL	RSTIO	OCDE
R/W	R/W	R/W	R/W	W	W	W	R/W	W
Reset Value	1	0	0	X	X	0	1	1

Bit 7: HSE, High Speed operation Enable.

0: Select CPU running in lower speed mode ($FCPUCLK \leq 6\text{MHz}$) which is slow down internal circuit to reduce power consumption.

1: Enable CPU full speed operation if $FCPUCLK > 6\text{MHz}$. Before select high frequency clock ($> 6\text{MHz}$) on CPUCLK, software must set HSE to switch internal circuit for high speed operation.

Bit 6: IAP0, IAP function only.

0: Maintain IAP region to service IAP function and code execution.

1: Disable the code execution in IAP region and the region only service IAP function.

Bit 5: HSE1, High Speed operation Enable 1. 0: No function.

1: Enable MCU for ultra-high speed operation. ($FCPUCLK > 25\text{MHz}$). It also needs to set HSE when use $HSE1 = 1$. Bit 4~3: Reserved. Software must write "0" on these bits when DCON0 is written.

Bit 2: IORCTL, GPIO Reset Control.

0: Port 6 keeps reset condition for all reset events.

1: If this bit is set, Port 6 is only reset by POR/LVR/Ext Reset/BOR0/BOR1 (if BOR0/1 is enabled).

Bit 1: RSTIO, RST function on I/O,

0: Select I/O pad function for P47.

1: Select I/O pad function for external reset input, RST.

Bit 0: OCDE, OCD enable.

0: Disable OCD interface on P4.4 and P4.5

1: Enable OCD interface on P4.4 and P4.5.

30 Auxiliary SFRs

30.1 SFR Figure (Page 0~F)

		0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
F8	0	P6	CH	CCAP0H	CCAP1H	CCAP2H	CCAP3H	CCAP4H	CCAP5H
	1			CCAP6H	CCAP7H				
F0	0	B	PAOE	PCAPWM0	PCAPWM1	PCAPWM2	PCAPWM3	PCAPWM4	PCAPWM5
	1			PCAPWM6	PCAPWM7				
E8	0	P4	CL	CCAP0L	CCAP1L	CCAP2L	CCAP3L	CCAP4L	CCAP5L
	1			CCAP6L	CCAP7L				
E0	0	ACC	WDTCR	IFD	IFADRH	IFADRL	IFMT	SCMD	ISPCR
	1								
D8	0	CCON	CMOD	CCAPM0	CCAPM1	CCAPM2	CCAPM3	CCAPM4	CCAPM5
	1			CCAPM6	CCAPM7				
D0	0	PSW	SIADR	SIDAT	SISTA	SICON	KBPATN	KBCON	KBMASK
	1								
C8	0	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2	CLRL	CHRL
	1	T3CON	T3MOD	RCAP3L	RCAP3H	TL3	TH3		
	3	T5CON	--	TLR5	THR5	TL5	TH5		
	4	T6CON	--	TLR6	THR6	TL6	TH6		
C0	0	XICON	XICFG	--	ADCFG0	ADCON0	ADCDL	ADCDH	CKCON0
	1		XICFG1		ADCFG1				
	2				ADCFG2				
	3				ADCFG3				
	4				ADCFG4				
	5				ADCFG5				
	B				ADCFG11				
	C				ADCFG12				
	D				ADCFG13				
	E				ADCFG14				
B8	0	IP0L	SADEN/ S0CR1	--	--	PWMCR PDT CRA	CRC0DA	RTCCR	--
	1								
B0	0	P3	P3M0	P3M1	P4M0	--	--	RTCTM	IP0H
	1				--	--	P6M0		
	2				--	PDRV C0	--		
	3				--	PDRV C1	--		
A8	0	IE	SADDR	--	--	SFRPI	EIE1	EIP1L	EIP1H
	1								
A0	0	P2	AUXR0	AUXR1	AUXR2	AUXR3	EIE2	EIP2L	EIP2H
	1					AUXR4			
	2					AUXR5			
	3					AUXR6			
	4					AUXR7			
	5					AUXR8			
	6					AUXR9			
	7					AUXR10			
	8					AUXR11			
98	0	S0CON	S0BUF	S0BRT	S0BRC	S0CFG	S0CFG1		
	1	S1CON	S1BUF	S1BRT	S1BRC	S1CFG			
90	0	P1	P1M0	P1M1	--	DMACR0	P2M0	BOREV	PCON1
	1			P2M1	T2MOD1	DMACR0	TREN0		
	2			P4M1	T3MOD1	DMACR0	TRLC0		
	3			P6M1	--	DMACR0	TSPC0		
	4			--	--	DMACR0	--		
	5			--	--	DMACR0	--		
	6			--	--	DMACR0	--		
	7			P3FDC	--	DMACR0	--		
	8			P1FDC	--	DMACG0	--		
	9			P2FDC	--	DMADS0	--		
	A			P4FDC	--	--	--		
88	0	TCON	TMOD	TL0	TL1	TH0	TH1	SFIE	XRPS
	1								
80	0	--	SP	DPL	DPH	SPSTAT	SPCON	SPDAT	PCON0
	1								

Note: Users need to set SFRPI=0x00~0x0F for SFR page access. (The MCU will not retain the value of SFRPI during interrupts. Users need to retain the value of SFRPI by software.)

- SFRPI: SFR Page Index Register

SFR Page = 0~F

SFR Address = 0xAC

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	--	PIDX3	PIDX2	PIDX1	PIDX0
R/W	W	W	W	W	R/W	R/W	R/W	R/W
Reset Value	X	X	X	X	0	0	0	0

Bit 7~4: Reserved. Software must write "0" on these bits when SFRPI is written.

Bit 3~0: SFR Page Index.

PIDX[3:0]	SelectedPage
0000	Page0
0001	Page1
0010	Page2
0011	Page3
.....
.....
.....
1111	PageF

30.2 SFR Bit Assignment (Page 0~F)

SYMBOL	DESCRIPTION	ADDR (HEX)	PAGE (HEX)	BITADDRESSANDSYMBOL								RESET VALUE
				Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	
SP	StackPointer	81	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000111
DPL	DataPointerLow	82	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
DPH	DataPointerHigh	83	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
SPSTAT	SPIStatusRegister	84	0-F	SPIF	WCOL	THRF	SPIBSY	MODF	--	--	SPR2	00000xx0
SPCON	SPIControlRegister	85	0-F	SSIG	SPEN	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	00000100
SPDAT	SPI DataRegister	86	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
PCON0	PowerControl0	87	0-F	SMOD1	SMOD0	GF	POF0	GF1	GF0	PD	IDL	00010000
TCON	TimerControl	88	0-F	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00000000
TMOD	TimerMode	89	0-F	T1GATE	T1C/T	T1M1	T1M0	T0GATE	T0C/T	T0M1	T0M0	00000000
TL0	TimerLow0	8A	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TL1	TimerLow1	8B	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TH0	TimerHigh0	8C	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TH1	TimerHigh1	8D	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
SFIE	SystemFlagINTEn.	8E	0-F	SIDFIE	--	--	RTCFIE	--	BOF1IE	BOF0IE	WDTFIE	0000x000
XRPS	XRAMPPageSelect	8F	0-F	--	--	--	--	--	--	.1	.0	xxxxx000
P1	Port1	90	0-F	P1.7	P1.6	P1.5	--	--	--	P1.1	P1.0	11111111
P1M0	P1ModeRegister0	91	0-F	P1M0.7	P1M0.6	P1M0.5	--	--	--	P1M0.1	P1M0.0	00000000
P1M1	P1ModeRegister1	92	0	P1M1.7	P1M1.6	P1M1.5	--	--	--	P1M1.1	P1M1.0	11111111
P2M1	P2ModeRegister1	92	1	--	--	--	P2M1.4	--	P2M1.2	--	--	11111111
P4M1	P4ModeRegister1	92	2	P4M1.7	--	P4M1.5	P4M1.4	--	--	--	--	11111111
P6M1	P6ModeRegister1	92	3	--	--	--	--	--	--	P6M1.1	P6M1.0	11111111
P3FDC	P3FastDrv.Ctrl.	92	7	--	--	.5	.4	.3	--	.1	.0	00000000
P1FDC	P1FastDrv.Ctrl.	92	8	.7	.6	.5	--	--	--	.1	.0	00000000
P2FDC	P2FastDrv.Ctrl.	92	9	--	--	--	.4	--	.2	--	--	00000000
P4FDC	P4FastDrv.Ctrl.	92	A	.7	--	.5	.4	--	--	--	--	00000000
T2MOD1	Timer2mode1Reg.	93	1	TL2CS	TF2IG	TL2IS	T2CKS	T2MS1	CP2S2	CP2S1	CP2S0	00000000
T3MOD1	Timer3mode1Reg.	93	2	TL3CS	TF3IG	--	T3CKS	T3MS1	CP3S2	CP3S1	CP3S0	00x00000
DMACR0	DMAControlReg.0	94	0-7	--	--	--	--	DMAE0	DMAS0	DIE0	DCF0	xxx00000
DMACG0	DMAConfiguredReg.0	94	8	PDMAH	PDMAL	CRCW0	0	EXTS10	EXTS00	FAEN0	LOOP0	00000000
DMADS0	DMADataPath Selection0	94	9	DSS30	DSS20	DSS10	DSS00	DDS30	DDS20	DDS10	DDS00	00000000
P2M0	P2ModeRegister0	95	0	--	--	--	P2M0.4	--	P2M0.2	--	--	00000000

SYMBOL	DESCRIPTION	ADDR (HEX)	PAGE (HEX)	BITADDRESSANDSYMBOL								RESET VALUE
				Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	
TREN0	TimerRunEnable Register 0	95	1	--	TR3LE	TR2LE	--	TR3E	TR2E	TR1E	TR0E	x00x0000
TRLC0	TimerReloadControl Register 0	95	2	--	TL3RLC	TL2RLC	--	T3RLC	T2RLC	T1RLC	T0RLC	x00x0000
TSPC0	TimerStopControl Register 0	95	3	--	TL3SC	TL2SC	--	T3SC	T2SC	T1SC	T0SC	x00x0000
BOREV	BitOrderReversed	96	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
PCON1	PowerControl1	97	0-F	SWRF	EXRF	--	RTCF	--	BOF1	BOF0	WDTF	0000x000
S0CON	Serial0Control	98	0	SM00 /FE	SM10	SM20	REN0	TB80	RB80	T10	R10	00000000
S1CON	Serial1Control	98	1	SM01	SM11	SM21	REN1	TB81	RB81	T11	R11	00000000
S0BUF	Serial0Buffer	99	0	.7	.6	.5	.4	.3	.2	.1	.0	xxxxxxxx
S1BUF	Serial1Buffer	99	1	.7	.6	.5	.4	.3	.2	.1	.0	xxxxxxxx
S0BRT	S0Baud-RateTimer	9A	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
S1BRT	S1Baud-RateTimer	9A	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
S0BRC	S0Baud-RateCounter	9B	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
S1BRC	S1Baud-RateCounter	9B	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
S0CFG	S0Configuration	9C	0	URTS	SMOD2	URM0X3	SM30	S0DOR	BTI	UTIE	SMOD3	00001000
S1CFG	S1Configuration	9C	1	SM31	S1M0X3	S1DOR	S1TR	S1MOD1	S1TX12	S1CKOE	S1TME	00100000
S0CFG1	S0Configuration1 (LINCFG)	9D	0	SBF0	TXER0	S0SB16	ATBR0	TXRX0	SYNC0	--	--	000000xx
P2	Port2	A0	0-F	--	--	--	P2.4	--	P2.2	--	--	11111111
AUXR0	AuxiliaryRegister0	A1	0-F	P60OC1	P60OC0	P60FD	PBKF	--	--	INT1H	INT0H	00000000
AUXR1	AuxiliaryRegister1	A2	0-F	--	--	CRCDS1	CRCDS0	--	--	--	DPS	00000000
AUXR2	AuxiliaryRegister2	A3	0-F	STAF	STOF	--	--	T1X12	T0X12	T1CKOE	T0CKOE	00000000
AUXR3	AuxiliaryRegister3	A4	0	T0PS1	T0PS0	BPOC1	BPOC0	S0PS0	TWIPS1	TWIPS0	T0XL	00000000
AUXR4	AuxiliaryRegister4	A4	1	T2PS1	T2PS0	T1PS1	T1PS0	--	--	--	--	00000000
AUXR5	AuxiliaryRegister5	A4	2	C0IC4S0	C0IC2S0	C0PPS1	C0PPS0	--	C0PS0	ECIPS0	C0COPS	00000000
AUXR6	AuxiliaryRegister6	A4	3	KBI4PS1	KBI4PS0	KBI6PS0	KBI2PS0	T3FCS	T2FCS	SnMIPS	S0COPS	00000000
AUXR7	AuxiliaryRegister7	A4	4	POE5	POE4	C0CKOE	SPI0M0	--	--	--	--	11000000
AUXR8	AuxiliaryRegister8	A4	5	POE7	POE6	C0PPS2	--	KBI0PS0	S1COPS	--	--	11000000
AUXR9	AuxiliaryRegister9	A4	6	--	--	T1G1	T0G1	C0FDC1	C0FDC0	S1PS1	S1PS0	00000000
AUXR10	Aux. Register10	A4	7	--	--	--	SPIPS0	S0PS1	--	TWICF	PAA	00000000
AUXR11	Aux.Register11	A4	8	P30AM	--	--	--	--	--	C0M0	C0OFS	00000000
EIE2	ExtendedINTEnable2	A5	0-F	--	--	--	--	--	--	--	ET3	xxxxxxx0
EIP2L	Ext. INTPriority2Low	A6	0-F	--	--	--	--	--	--	--	PT3L	xxxxxxx0
EIP2H	Ext. INTPriority2High	A7	0-F	--	--	--	--	--	--	--	PT3H	xxxxxxx0
IE	InterruptEnable	A8	0-F	EA	EDMA	ET2	ES0	ET1	EX1	ET0	EX0	00000000
SADDR	SlaveAddress	A9	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
SFRPI	SFRPage Index	AC	0-F	--	--	--	--	IDX3	IDX2	IDX1	IDX0	xx000000
EIE1	ExtendedINTEnable1	AD	0-F	--	ETW10	EKB	ES1	ESF	EPCA	EADC	ESPI	00000000
EIP1L	Ext. INTPriority1Low	AE	0-F	--	PTW10L	PKBL	PS1L	PSFL	PPCAL	PADCL	PSPIL	00000000
EIP1H	Ext. INTPriority1High	AF	0-F	--	PTW10H	PKBH	PS1H	PSFH	PPCAH	PADCH	PSPIH	00000000
P3	Port3	B0	0-F	--	--	P3.5	P3.4	P3.3	--	P3.1	P3.0	11111111
P3M0	P3ModeRegister0	B1	0-F	--	--	P3M0.5	P3M0.4	P3M0.3	--	P3M0.1	P3M0.0	00000000
P3M1	P3ModeRegister1	B2	0-F	--	--	P3M1.5	P3M1.4	P3M1.3	--	P3M1.1	P3M1.0	00000000
P4M0	P4ModeRegister0	B3	0	P4M0.7	--	P4M0.5	P4M0.4	--	--	--	--	10110000
PDRVC0	Port DrivingControl0	B4	2	P3DC1	P3DC0	P2DC1	P2DC0	P1DC1	P1DC0	--	--	00000000
PDRVC1	Port DrivingControl1	B4	3	--	--	--	--	--	--	P4DC1	--	xxx0xx00
P6M0	P6ModeRegister0	B5	1	--	--	--	--	--	--	P6M0.1	P6M0.0	00000000
RTCTM	RTCTimerRegister	B6	0-F	RTCCS1	RTCCS0	RTCCT5	RTCCT4	RTCCT3	RTCCT2	RTCCT1	RTCCT0	01111111
IP0H	InterruptPriority0High	B7	0-F	--	PX2H	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	00000000
IP0L	InterruptPriorityLow	B8	0-F	--	PX2L	PT2L	PSL	PT1L	PX1L	PT0L	PX0L	00000000
SADEN	SlaveAddressMask	B9	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
S0CR1	S0Control1	B9	0-F	S0TR	S0TX12	S0TCK	S0RCK	S0CKOE	ARTE	--	--	00000000
PWMCR	PWMControl Reg.	BC	0	PCAE	EXDT	PBKM	PBKE1.1	PBKE1.0	PBKE0.2	PBKE0.1	PBKE0.0	00000000
PDTCRA	PWMDead-Time ControlReg. -A	BC	1	DTPS1	DTPS0	DT.5	DT.4	DT.3	DT.2	DT.1	DT.0	00000000
CRC0DA	CRC0DataPort	BD	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
RTCCR	RTCCControlReg.	BE	0-F	RTCE	RTCO	RTCRL5	RTCRL4	RTCRL3	RTCRL2	RTCRL1	RTCRL0	00111111
XICON	ExternalINTControl	C0	0-F	--	--	--	--	INT2H	EX2	IE2	IT2	xxxx0000
XICFG	Ext.INT.Configured	C1	0	INT1IS1	INT1IS0	INT0IS1	INT0IS0	--	X2FLT	X1FLT	X0FLT	00000000
XICFG1	Ext.INT.Configured1	C1	1	INT1IS2	INT0IS2	INT2IS1	INT2IS0	--	X2FLT1	X1FLT1	X0FLT1	00000000
ADCFG0	ADCCConfiguration0	C3	0	ADCKS2	ADCKS1	ADCKS0	ADRJ	ACHS	SMPF	ADTM1	ADTM0	00000000
ADCFG1	ADCCConfiguration1	C3	1	IGADCI	EADCW1	SMPFIE	SIGN	AOS.3	AOS.2	AOS.1	AOS.0	00000000
ADCFG2	ADCCConfiguration2	C3	2	SHT.7	SHT.6	SHT.5	SHT.4	SHT.3	SHT.2	SHT.1	SHT.0	00000000
ADCFG3	ADCCConfiguration3	C3	3	ADPS1	ADPS0	--	--	ARES1	ARES0	ADES0	--	010000xx
ADCFG4	ADCCConfiguration4	C3	4	--	ADWM0	ADTM3	ADTM2	--	--	DBSD	--	0000xx00

SYMBOL	DESCRIPTION	ADDR (HEX)	PAGE (HEX)	BITADDRESSANDSYMBOL								RESET VALUE
				Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	
ADCFG5	ADCConfiguration5	C3	5	ASCE.7	ASCE.6	ASCE.5	ASCE.4	ASCE.3	ASCE.2	ASCE.1	ASCE.0	00000000
ADCFG11	ADCConfiguration11	C3	B	WHB.3	WHB.2	WHB.1	WHB.0	1	1	1	1	11111111
ADCFG12	ADCConfiguration12	C3	C	WHB.11	WHB.10	WHB.9	WHB.8	WHB.7	WHB.6	WHB.5	WHB.4	11111111
ADCFG13	ADCConfiguration13	C3	D	WLB.3	WLB.2	WLB.1	WLB.0	0	0	0	0	00000000
ADCFG14	ADCConfiguration14	C3	E	WLB.11	WLB.10	WLB.9	WLB.8	WLB.7	WLB.6	WLB.5	WLB.4	00000000
ADCON0	ADCControl0	C4	0-F	ADCEN	ADCWI	--	ADCI	ADCS	CHS2	CHS1	CHS0	0x000000
ADCDL	ADCData Low	C5	0-F	ADCV.3	ADCV.2	ADCV.1	ADCV.0	--	--	--	--	0000xxxx
ADCDH	ADCData High	C6	0-F	ADCV.11	ADCV.10	ADCV.9	ADCV.8	ADCV.7	ADCV.6	ADCV.5	ADCV.4	00000000
CKCON0	ClockControl0	C7	0-F	AFS	ENCKM	CKMIS1	CKMIS0	CCKS	SCKS2	SCKS1	SCKS0	00010000
T2CON	Timer2ControlReg.	C8	0	TF2	EXF2	RCLK/ TF2L	TCLK/ TL2IE	EXEN2	TR2	C/T2	CP/RL2	00000000
T3CON	Timer3ControlReg.	C8	1	TF3	EXF3	TF3L	TL3IE	EXEN3	TR3	C/T3	CP/RL3	00000000
T5CON	Timer5ControlReg.	C8	3	TF5	--	T5CKS1	T5CKS0	T5IE	TR5	T5GAT1	T5GAT0	00000000
T6CON	Timer6ControlReg.	C8	4	TF6	--	T6CKS1	T6CKS0	T6IE	TR6	T6GAT1	T6GAT0	00000000
T2MOD	Timer2modeReg.	C9	0	T2SPL	TL2X12	T2EXH	T2X12	TR2L	TR2LC	T2OE	T2MS0	00000000
T3MOD	Timer3modeReg.	C9	1	T3SPL	TL3X12	T3EXH	T3X12	TR3L	TR3LC	T3OE	T3MS0	00000000
RCAP2L	Timer2CaptureLow	CA	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
RCAP3L	Timer3CaptureLow	CA	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TLR5	TL5reloadReg.	CA	3	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TLR6	TL6reloadReg.	CA	4	.7	.6	.5	.4	.3	.2	.1	.0	00000000
RCAP2H	Timer2CaptureHigh	CB	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
RCAP3H	Timer3CaptureHigh	CB	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
THR5	TH5reloadReg.	CB	3	.7	.6	.5	.4	.3	.2	.1	.0	00000000
THR6	TH6reloadReg.	CB	4	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TL2	TimerLow2	CC	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TL3	TimerLow3	CC	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TL5	TimerLow5	CC	3	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TL6	TimerLow6	CC	4	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TH2	TimerHigh2	CD	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TH3	TimerHigh3	CD	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TH5	TimerHigh5	CD	3	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TH6	TimerHigh6	CD	4	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CLRL	CLRReloadregister	CE	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CHRL	CHReloadregister	CF	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
PSW	ProgramStatusWord	D0	0-F	CY	AC	F0	RS1	RS0	OV	F1	P	00000000
SIADR	TWIOAddressReg.	D1	0-F	.7	.6	.5	.4	.3	.2	.1	GC	00000000
SIDAT	TWIODataReg.	D2	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
SISTA	TWIOStatusReg.	D3	0-F	.7	.6	.5	.4	.3	.2	.1	.0	11111000
SICON	TWIOControlReg.	D4	0-F	CR2	ENSI	STA	STO	SI	AA	CR1	CR0	00000000
KBPATN	KeypadPattern	D5	0-F	.7	.6	.5	.4	.3	.2	.1	.0	11111111
KBCON	KeypadControl	D6	0-F	KBCS1	KBCS0	KBES	--	--	--	PATN_ SEL	KBIF	00000001
KBMASK	KeypadInt. Mask	D7	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCON	PCAControlReg.	D8	0-F	CF	CR	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0	00000000
CMOD	PCAModeReg.	D9	0-F	CIDL	BME4	BME2	BME0	CPS2	CPS1	CPS0	ECF	00000000
CCAPM0	PCAModule0Mode	DA	0	DTE0	ECOM0	CAPP0	CAPN0	MAT0	TOG0	PWM0	ECCF0	00000000
CCAPM6	PCAModule6Mode	DA	1	BME6	ECOM6	--	CAPN6	MAT6	TOG6	PWM6	ECCF6	00xx0000
CCAPM1	PCAModule1Mode	DB	0	--	ECOM1	CAPP1	CAPN1	MAT1	TOG1	PWM1	ECCF1	x0000000
CCAPM7	PCAModule7Mode	DB	1	--	ECOM7	--	CAPN7	MAT7	TOG7	PWM7	ECCF7	x0xx0000
CCAPM2	PCAModule2Mode	DC	0-F	DTE2	ECOM2	CAPP2	CAPN2	MAT2	TOG2	PWM2	ECCF2	00000000
CCAPM3	PCAModule3Mode	DD	0-F	--	ECOM3	CAPP3	CAPN3	MAT3	TOG3	PWM3	ECCF3	x0000000
CCAPM4	PCAModule4Mode	DE	0-F	DTE4	ECOM4	CAPP4	CAPN4	MAT4	TOG4	PWM4	ECCF4	00000000
CCAPM5	PCAModule5Mode	DF	0-F	--	ECOM5	CAPP5	CAPN5	MAT5	TOG5	PWM5	ECCF5	x0000000
ACC	Accumulator	E0	0-F	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	00000000
WDTCR	WDTControlregister	E1	0-F	WREN	NSW	ENW	CLRW	WIDL	PS2	PS1	PS0	00000000
IFD	ISPFflashdata	E2	0-F	.7	.6	.5	.4	.3	.2	.1	.0	11111111
IFADRH	ISP FlashAddr.High	E3	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
IFADRL	ISP FlashAddr.Low	E4	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
IFMT	ISPModeTable	E5	0-F	MS.7	--	--	--	MS.3	MS.2	MS.1	MS.0	00000000
SCMD	ISPSerialCommand	E6	0-F	.7	.6	.5	.4	.3	.2	.1	.0	xxxxxxxx
ISPCR	ISPControlRegister	E7	0-F	ISPEN	SWBS	SRST	CFAIL	--	--	--	--	00000xxx
P4	Port4	E8	0-F	P4.7	--	P4.5	P4.4	--	--	--	--	11111111
CL	PCAbasetimerLow	E9	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP0L	PCAModule0captureLow	EA	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP6L	PCAModule6captureLow	EA	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP1L	PCAModule1captureLow	EB	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP7L	PCAModule7captureLow	EB	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP2L	PCAModule2captureLow	EC	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000

SYMBOL	DESCRIPTION	ADDR (HEX)	PAGE (HEX)	BIT ADDRESS AND SYMBOL								RESET VALUE
				Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	
CCAP3L	PCAModule3captureLow	ED	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP4L	PCAModule4captureLow	EE	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP5L	PCAModule5captureLow	EF	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
<i>B</i>	<i>BRegister</i>	<i>F0</i>	<i>0-F</i>	<i>B.7</i>	<i>B.6</i>	<i>B.5</i>	<i>B.4</i>	<i>B.3</i>	<i>B.2</i>	<i>B.1</i>	<i>B.0</i>	<i>00000000</i>
PAOE	PWMAdditionalOutput Enable	F1	0-F	POE3	POE2B	POE2A	POE2	POE1	POE0B	POE0A	POE0	10011001
PCAPWM0	PCAPWM0Mode	F2	0	P0RS1	P0RS0	--	--	--	P0INV	ECAP0H	ECAP0L	00xxx000
PCAPWM6	PCAPWM6Mode	F2	1	P6RS1	P6RS0	--	--	CCF6	P6INV	ECAP6H	ECAP6L	00xx0000
PCAPWM1	PCAPWM1Mode	F3	0	P1RS1	P1RS0	--	--	--	P1INV	ECAP1H	ECAP1L	00xxx000
PCAPWM7	PCAPWM7Mode	F3	1	P7RS1	P7RS0	--	--	CCF7	P7INV	ECAP7H	ECAP7L	00xx0000
PCAPWM2	PCAPWM2Mode	F4	0-F	P2RS1	P2RS0	--	--	--	P2INV	ECAP2H	ECAP2L	00xxx000
PCAPWM3	PCAPWM3Mode	F5	0-F	P3RS1	P3RS0	--	--	--	P3INV	ECAP3H	ECAP3L	00xxx000
PCAPWM4	PCAPWM4Mode	F6	0-F	P4RS1	P4RS0	--	--	--	P4INV	ECAP4H	ECAP4L	00xxx000
PCAPWM5	PCAPWM5Mode	F7	0-F	P5RS1	P5RS0	--	--	--	P5INV	ECAP5H	ECAP5L	00xxx000
<i>P6</i>	<i>Port6</i>	<i>F8</i>	<i>0-F</i>	--	--	--	--	--	--	<i>P6.1</i>	<i>P6.0</i>	<i>xxx11111</i>
CH	PCAbasetimerHigh	F9	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP0H	PCA Module0 captureHigh	FA	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP6H	PCA Module6 capture High	FA	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP1H	PCA Module1 captureHigh	FB	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP7H	PCA Module7 capture High	FB	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP2H	PCA Module2 captureHigh	FC	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP3H	PCA Module3 capture High	FD	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP4H	PCA Module4 captureHigh	FE	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP5H	PCA Module5 capture High	FF	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000

30.3 Auxiliary SFR Map (Page P)

CMT2380F17 has an auxiliary SFR page which is indexed by page P and the SFRs' write is a different way from standard 8051 SFR page. The registers in auxiliary SFR map are addressed by IFMT and SCMD like ISP/IAP access flow. Page P has 256 bytes space that can target to 11 physical bytes and 6 logical bytes. IAPLB, CKCON2, CKCON3, CKCON4, CKCON5, PCON2, PCON3, SPCON0, DCON0, RTCTM and RTCCR. The 6 logical bytes include PCON0, PCON1, CKCON0, WDTCSR, P4 and P6. Access on the 6 logical bytes gets the coherence content with the same SFR in Page 0~F. Please refer Section "29 Page P SFR Access" for more detail information.

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
F8	P6	--	--	--	--	--	--	--
F0	--	--	--	--	--	--	--	--
E8	P4	--	--	--	--	--	--	--
E0	--	WDTCSR	--	--	--	--	--	--
D8	--	--	--	--	--	--	--	--
D0	--	--	--	--	--	--	--	--
C8	--	--	--	--	--	--	--	--
C0	--	--	--	--	--	--	--	CKCON0
B8	--	--	--	--	--	--	--	--
B0	--	--	--	--	--	--	--	--
A8	--	--	--	--	--	--	--	--
A0	--	--	--	--	--	--	--	--
98	--	--	--	--	--	--	--	--
90	--	--	--	--	--	--	--	PCON1
88	--	--	--	--	--	--	--	--
80	--	--	--	--	--	--	--	PCON0
78	--	--	--	--	--	--	--	--
70	--	--	--	--	--	--	--	--
68	--	--	--	--	--	--	--	--
60	--	--	--	--	--	--	--	--
58	--	--	--	--	--	--	--	--
50	--	--	--	--	RTCCR	RTCTM	--	--
48	SPCON0	--	--	--	DCON0	--	--	--
40	CKCON2	CKCON3	CKCON4	CKCON5	PCON2	PCON3	--	--
38	--	--	--	--	--	--	--	--
30	--	--	--	--	--	--	--	--
28	--	--	--	--	--	--	--	--
20	--	--	--	--	--	--	--	--
18	--	--	--	--	--	--	--	--
10	--	--	--	--	--	--	--	--
08	--	--	--	--	--	--	--	--
00	--	--	--	IAPLB	--	--	--	--

30.4 Auxiliary SFR Bit Assignment (Page P)

SYMBOL	DESCRIPTION	ADDR	BIT ADDRESS AND SYMBOL								RESET VALUE
			Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	
Physical Bytes											
IAPLB	IAPLow Boundary	03H	IAPLB6	IAPLB5	IAPLB4	IAPLB3	IAPLB2	IAPLB1	IAPLB0	0	
CKCON2	ClockControl2	40H	--	--	--	IHRCOE	MCKS1	MCKS0	OSCS1	OSCS0	0001-0000
CKCON3	ClockControl3	41H	WDTCS1	WDTCS0	FWKP	WDTFS	MCKD1	MCKD0	--	--	00000000

CKCON4	ClockControl4	42H	RCSS2	RCSS1	RCSS0	RPSC2	RPSC1	RPSC0	RTCCS3	RTCCS2	00000000
CKCON5	ClockControl5	43H	--	--	--	--	--	--	--	CKMS0	00000000
PCON2	PowerControl2	44H	AWBOD1	0	BO1S1	BO1S0	BO1RE	EBOD1	BO0RE	1	0000x1x1
PCON3	PowerControl3	45H	IVREN	0	0	0	0	0	0	0	00000000
SPCON0	SFR Page Control 0	48H	--	P6CTL	P4CTL	WRCTL	--	CKCTL0	PWCTL1	PWCTL0	00000000
DCON0	DeviceControl 0	4CH	HSE	IAPO	HSE1	--	--	IORCTL	RSTIO	OCDE	100xx011
RTCCR	RTCCControlReg.	54H	RTCE	RTCO	RTCRL5	RTCRL4	RTCRL3	RTCRL2	RTCRL1	RTCRL0	00111111
RTCTM	RTC TimerRegi ster	55H	RTCCS1	RTCCS0	RTCCT5	RTCCT4	RTCCT3	RTCCT2	RTCCT1	RTCCT0	01111111
LogicalBytes											
PCON0	PowerControl0	87H	SMOD1	SMOD0	GF	POF0	GF1	GF0	PD	IDL	00010000
PCON1	PowerControl1	97H	SWRF	EXRF	--	RTCF	--	BOF1	BOF0	WDTF	0000x000
CKCON0	ClockControl0	C7H	AFS	ENCKM	CKMIS1	CKMIS0	CCKS	SCKS2	SCKS1	SCKS0	00010000
WDTCR	Watch-dog-timer Controlregister	E1H	WREN	NSW	ENW	CLRW	WIDL	PS2	PS1	PS0	00000000
P4	Port4	E8H	P4.7	--	P4.5	P4.4	--	--	--	--	1x11xx11
P6	Port6	F8H	--	--	--	--	--	--	P6.1	P6.0	xxxxxx11

Sample Code of

Page-P

SFR write:

IFADRH=0

x00;

ISPCR =ISPEN;

//enableIAP/ISP

IFMT=MS2;

//Page-Pwrite, IFMT=0x04

IFADRL=SPCON0;

//SetPage-PSFRaddress

IFD |=CKCTL0;

//setCKCTL0

SCMD=0x46;

//

SCMD=0xB9;

//

IFMT=Flash_Standby;

//IAP/ISPst

andby, IFMT=0x00ISPCR&= ~ISPEN;

30.5 Auxiliary SFR Register

- **AUXR0:** Auxiliary Register 0
SFR Page = 0~F
SFR Address = 0xA1

Bit	7	6	5	4	3	2	1	0
Name	P6OOC1	P6OOC0	P60FD	PBKF	--	--	INT1H	INT0H
R/W	R/W	R/W	R/W	R/W	W	W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~6: P6.0 function configured control bit 1 and 0. The two bits only act when internal RC oscillator (IHRCO or ILRCO) is selected for system clock source. In external clock input mode, P6.0 is the dedicated clock input pin. In internal oscillator condition, P6.0 provides the following selections for GPIO or clock source generator. When P6OOC[1:0] index to non-P6.0 GPIO function, P6.0 will drive the on-chip RC oscillator output to provide the clock source for other devices.

P6OOC[1:0]	P60 功能	I/O mode
00	P60	By P6M0.0
01	MCK	By P6M0.0
10	MCK/2	By P6M0.0
11	MCK/4	By P6M0.0

Please refer Section “9 System Clock” to get the more detailed clock information. For clock-out on P6.0 function, it is recommended to set P6M0.0 to “1” which selects P6.0 as push-push output mode.

Bit 5: P60FD, P6.0 Fast Driving.

0: P6.0 output with default driving.

1: P6.0 output with fast driving enabled. If P6.0 is configured to clock output, enable this bit when P6.0 output frequency is more than 12MHz at 5V application or more than 6MHz at 3V application.

Bit 4: PBKF, PWM termination flag. This bit is set by PWM termination source enabling. If this bit is set, the enabled PWM channels 0~5 will be locked and the output pins will maintain the original GPIO state.

0: No PWM termination event occurs. Only cleared by software.

1: PWM termination event occurs or software triggers a PWM termination.

Bit 1: INT1H, INT1 high level/rising edge triggering enabling.

0: Reserve INT1 to trigger on low level or falling edge on the selected port pin.

1: Set INT1 to trigger on high level or rising edge on the selected port pin.

Bit 0: INT0H, INT0 high level/rising edge trigger enable.

0: Reserve INT0 to trigger on low level or falling edge on the selected port pin.

1: Set INT0 to trigger on high level or rising edge on the selected port pin.

- **AUXR1:** Auxiliary Control Register 1
SFR Page = 0~F
SFR Address = 0xA2

Bit	7	6	5	4	3	2	1	0
Name	--	--	CRCDS1	CRCDS0	--	--	--	DPS
R/W	W	W	R/W	R/W	W	W	W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~6: Reserved. When AUXR1 is written, the software must write "0" to these bits.

Bit 5~4: CRCDS1~0. CRC0 data port selection 1~0.

Bit 3~1: Reserved bits. When AUXR1 is written, the software must write "0" to these bits.

Bit 0: DPS, DPTR select bit. Use to switch between DPTR0 and DPTR1.

0: Select DPTR0.

1: Select DPTR1.

DPS	DPTR Selection
0	DPTR0
1	DPTR1

- AUXR2: Auxiliary Register 2

SFR Page = 0 only

SFR Address = 0xA3

Bit	7	6	5	4	3	2	1	0
Name	STAF	STOF	--	--	T1X12	T0X12	T1CKOE	T0CKOE
R/W	R/W	R/W	W	W	R/W	R/W	R/W	R/W
Reset Value	0	0	X	X	0	0	0	0

Bit 7: STAF, Start Flag detection of STWI (SID).

0: Clear by firmware by writing "0" on it. STAF might be held within MCU reset period, so needs to clear STAF in firmware initial.

1: Set by hardware to indicate the START condition occurred on STWI bus.

Bit 6: STOF, Stop Flag detection of STWI (SID). 0: Clear by firmware by writing "0" on it.

1: Set by hardware to indicate the STOP condition occurred on STWI bus. STOF might be held within MCU reset period, so needs to clear STOF in firmware initial.

Bit 5~4: 保留位。当写 AUXR2 寄存器时软件必须在这些位写 "0"。

Bit 5~4: Reserved bits. When writing the AUXR2 register, the software must write "0" in these bits.

Bit 3: T1X12, Timer 1 clock source selection when C/T=0.

0: Clear to select SYSCLK/12 as the timer 1 clock source.

1: Set to select SYSCLK as the timer 1 clock source.

Bit 2: T0X12, Timer 0 clock source selection when C/T=0.

0: Clear to select SYSCLK/12 as the timer 0 clock source.

1: Set to select SYSCLK as the timer 0 clock source.

Bit 1: T1CKOE, Timer 1 clock output enable.

0: Disable timer 1 clock output.

1: Enable Timer 1 clock output on the T1CKO port pin.

Bit 0: T0CKOE, Timer 0 clock output enable.

0: Disable timer 0 clock output.

1: Enable timer 0 clock output on the T0CKO port pin.

- AUXR3: Auxiliary Register 3

SFR Page = 0 only

SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	T0PS1	T0PS0	BPOC1	BPOC0	S0PS0	TWIPS1	TWIPS0	T0XL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~6: T0PS1~0, Timer 0 Port pin Selection [1:0].

T0PS1~0	T0/T0CKO
---------	----------

00	P3.4
01	P4.4
10	P2.2
11	P1.7

Bit 5~4: T1PS1~0, Timer 1 Port pin Selection [1:0].

BPOC[1:0]	P4.4Function	I/O Mode
00	P4.4	By P4M0.4 & P4M1.4
01	ILRCO/32	By P4M0.4 & P4M1.4
10	ILRCO/16	By P4M0.4 & P4M1.4
11	ILRCO/8	By P4M0.4 & P4M1.4

The buzzer function is in P4.4, it is recommended to set P4.4 to work in push-pull output mode.

Bit 3: S0PS0, serial port 0 (UART0) port pin selection bit 0. (S0PS1 is in AUXR10.3)

S0PS1, S0PS0	RXD0	TXD0
0 0	P3.0	P3.1
0 1	P4.4	P4.5
1 0	P3.1	P3.0
1 1	P1.7	P2.2

Bit 2~1: TWIPS1~0, TWI0 port pin selection bits[1:0].

TWIPS1~0	TWI0_SCL	TWI0_SDA
0 0	P3.1	P3.0
0 1	P6.0	P6.1
1 0	P3.0	P3.1
1 1	P2.2	P2.4

Bit 0: T0XL is the timer 0 prescaler control bit. Please refer to T0X12 for T0XL function definition.

- AUXR4: Auxiliary Register 4
SFR Page = 1 only
SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	T2PS1	T2PS0	T1PS1	T1PS0	--	--	--	--
R/W	R/W	R/W	R/W	R/W	W	W	W	W
Reset Value	0	0	0	0	0	X	0	0

Bit 7~6: T2PS1~0, Timer 2 port pin selection bits[1:0].

T2PS1~0	T2/T2CKO	T2EX
0 0	P1.0	P1.1
0 1	P3.0	P3.1
1 0	P6.0	P3.5
1 1	P4.5	P4.4

Bit 5~4: T1PS1~0, Timer 1 port pin selection bits[1:0].

T1PS1~0	T1/T1CKO
0 0	P3.5
0 1	P4.5
1 0	P1.7
1 1	P3.3

- AUXR5: Auxiliary Register 5
SFR Page = 2 only
SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	C0IC4S0	C0IC2S0	C0PPS1	C0PPS0	--	C0PS0	ECIPS0	C0COPS
R/W	R/W	R/W	R/W	R/W	W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7: C0IC4S0, PCA0 Input Channel 4 input port pin Selection.

C0IC4S0	CEX4 input
0	CEX4 Port Pin
1	T2EXI

Bit 6: C0IC2S0, PCA0 Input Channel 2 input port pin Selection.

C0IC2S0	CEX2 input
0	CEX2 Port Pin
1	T3EXI

Bit 5: C0PPS1, {PWM2A, PWM2B} Port pin Selection 1.

C0PPS1	PWM2A	PWM2B
0	P6.0	P6.1
1	P3.4	P3.5

Bit 4: C0PPS0, {PWM0A, PWM0B} Port pin Selection 0.

C0PPS0	PWM0A	PWM0B
0	P1.6	P1.7
1	P6.0	P6.1

Bit 3: Reserved.

Bit 2: C0PS0, PCA0 Port pin Selection 0.

C0PS0	CEX0	CEX2	CEX4
0	P2.2	P2.4	P1.7
1	P3.0	P2.4	P3.1

Bit 1: ECIPS0, PCA0 ECI Port pin Selection0.

ECIPS0	ECI
0	P4.4
1	P1.6

Bit 0: C0COPS, PCA0 Clock Output (C0CKO) port pin Selection.

C0COPS	C0CKO
0	P4.7
1	P3.3

- AUXR6: Auxiliary Register 6
SFR Page = 3 only
SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	KBI4PS1	KB4IPS0	KBI6PS0	KBI2PS0	T3FCS	T2FCS	SnMIPS	S0COPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 7~6: KBI4PS1~0, KBI4~5 Port pin Selection [1:0].

KBI4PS1~0	KBI4	KBI5
-----------	------	------

00	P3.3	P1.5
01	P3.4	P3.5
10	P6.0	P6.1
11	P1.5	P3.3

Bit 5: KBI6PS0, KBI6~7 Port pin Selection 0.

KBI6PS0	KBI6	KBI7
0	P1.6	P1.7
1	P3.0	P3.1

Bit 4: KBI2PS0, KBI2~3 Port pin Selection 0.

KBI2PS0	KBI2	KBI3
0	P3.0	P3.1
1	P2.2	P2.4

Bit 3: T3FCS, Reserved for chip test.

Bit 2: T2FCS, Reserved for chip test.

Bit 1: SnMIPS, S0MI & S1MI Port pin Selection.

SnMIPS	S0MI	S1MI
0	P1.6	P6.1
1	P3.3	P4.7

Bit 0: S0COPS, S0BRG Clock Output (S0CKO) port pin Selection.

S0COPS	S0CKO
0	P4.7
1	P3.3

● AUXR7: Auxiliary Register 7

SFR Page = 4 only

SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	POE5	POE4	C0CKOE	SPI0M0	--	--	--	--
R/W	R/W	R/W	R/W	R/W	W	W	W	W
Reset Value	1	1	0	0	0	0	0	0

Bit 7: POE5, PCA0 PWM5 main channel (PWM5O) output control. 0: Disable PWM5O output on port pin. 1: Enable PWM5O output on port pin. Default is enabled.

Bit 6: POE4, PCA0 PWM4 main channel (PWM4O) output control. 0: Disable PWM4O output on port pin. 1: Enable PWM4O output on port pin. Default is enabled.

Bit 5: C0CKOE, PCA0 clock output (C0CKO) enable. 0: Disable PCA0 clock output. 1: Enable PCA0 clock output with PCA0 base timer overflow rate/2.

Bit 4: SPIM0, SPI mode control 0.
0: Disable SPI daisy chain function.
1: SPI daisy chain function in SPI slave mode is enabled.

Bit 3~0: Reserved bits. When writing the AUXR7 register, the software must write "0" in these bits.

- AUXR8: Auxiliary Register 8
SFR Page = 5 only
SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	POE7	POE6	C0PPS2	--	KBI0PS0	S1COPS	--	--
R/W	R/W	R/W	R/W	W	R/W	R/W	W	W
Reset Value	1	1	0	0	0	0	0	0

Bit 7: POE7, PCA0 PWM7 main channel (PWM7O) output control.

0: Disable PWM7O output on port pin.

1: Enable PWM7O output on port pin. Default is enabled.

Bit 6: POE6, PCA0 PWM6 main channel (PWM6O) output control.

0: Disable PWM6O output on port pin.

1: Enable PWM6O output on port pin. Default is enabled.

Bit 5: C0PPS2, {PWM6, PWM7} Port pin Selection 2.

C0PPS2	PWM6	PWM7
0	P6.0	P6.1
1	P3.0	P3.1

Bit 3: KBI0PS0, KBI0~1 Port pin Selection 0.

KBI0PS0	KBI0	KBI1
0	P1.0	P1.1
1	P4.7	P3.3

Bit 2: S1COPS, S1BRG Clock Output (S1CKO) port pin Selection.

S1COPS	S1CKO
0	P4.7
1	P6.1

- AUXR9: Auxiliary Register 9
SFR Page = 6 only
SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	--	--	T1G1	T0G1	C0FDC1	C0FDC0	S1PS1	S1PS0
R/W	W	W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 1~0: S1PS1~0, Serial Port 1 pin Selection [1:0].

T1G1, T1GATE	T1 Gate Control Source
00	Disabled
01	INT1 activation
10	TF3 activation
11	TI1 activation

Bit 4: T0G1, Timer 0 gate control source selection.

T0G1, T0GATE	T0 Gate Control Source
00	Disabled
01	INT0 activation
10	TF2 activation
11	KBI activation

Bit 3~2: C0FDC1~0, C0FDCKselection [1:0].

C0FDC1~0	C0FDCK
00	T0OF
01	T1OF
10	T3OF
11	S0TOF

Bit 1~0: S1PS1~0, serial port 1 (UART1) port pin selection bits [1:0].

S1PS1, S1PS0	RxD1	TxD1
00	P1.0	P1.1
01	P6.0	P6.1
10	P4.4	P4.5
11	P3.4	P3.5

- AUXR10: Auxiliary Register 10

SFR Page = 7 only

SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	SPIPS0	S0PS1	--	TWICF	PAA
R/W	W	W	W	R/W	R/W	W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 4: SPIPS0, SPI Port pin Selection 0.

SPIPS0	nSS	MOSI	MISO	SPICLK
0	P3.3	P1.5	P1.6	P1.7
1	P1.7	P3.5	P3.4	P3.3

Bit 1: TWICF, TWI0/I2C0 serial clock input filter.

0: Disable TWICF function.

1: Enable TWICF function.

Bit 0: PAA, pre-assertion response.

0: Disable PAA function.

1: TWI0/I2C0 master RX and slave TX/RX enable PAA function during DMA transmission.

- AUXR11: Auxiliary Register 11

SFR Page = 8 only

SFR Address = 0xA4

Bit	7	6	5	4	3	2	1	0
Name	P30AM	--	--	--	--	--	COM0	C0OFS

R/W	R/W	W	W	W	W	W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit 1: C0M0, PCA0 Mode control 0.

0: Not support variable resolution on central aligned PWM.

1: Enable PCA0 variable resolution central aligned PWM. To enable this function, the PCAE also needs to be set.

Bit 0: C0OFS, PCA0 overflow flag selection when C0M0 is enabled.

0: CF is set on the top of central aligned PWM cycle.

1: CF is set on the bottom of central aligned PWM cycle.

- SFRPI: SFR Page Index Register

SFR Page = 0~F

SFR Address = 0xAC

Bit	7	6	5	4	3	2	1	0
Name	--	--	--	--	PIDX3	PIDX2	PIDX1	PIDX0
R/W	W	W	W	W	R/W	R/W	R/W	R/W
Reset Value	X	X	X	X	0	0	0	0

Bit 7~4: Reserved. Software must write "0" on these bits when SFRPI is written.

Bit 3~0: SFR Page Index.

PIDX[3:0]	Selected Page
0000	Page 0
0001	Page 1
0010	Page 2
0011	Page 3
.....
.....
1111	Page F

31 Hardware Option

The MCU's Hardware Option defines the device behavior which cannot be programmed or controlled by software. The hardware options can only be programmed by a Universal Programmer, the "Megawin 8051 Writer U1" or the "Megawin 8051 ICE Adapter" (The ICE adapter also supports ICP programming function. Refer Section "32.4 In-Chip-Programming Function"). After whole-chip erased, all the hardware options are left in "disabled" state and there is no ISP-memory and IAP-memory configured. The CMT2380F17 has the following Hardware Options:

LOCK:

- ☒ : Enabled. Code dumped on a universal writer or programmer is locked to 0xFF for security.
☐ : Disabled. Not locked.

ISP-memory Space:

The ISP-memory space is specified by its starting address. And, its higher boundary is limited by the Flash end address, i.e., **0x3FFF**. The following table lists the ISP space option in this chip. In default setting, CMT2380F17 ISP space is configured to **1.5K** that had been embedded Megawin proprietary COMBO ISP code to perform device firmware upgrade through Megawin 1-Line ISP protocol and COM port ISP.

ISP-memory Size	ISP Start Address
7.5K bytes	2200
7.0K bytes	2400
6.5K bytes	2600
6.0K bytes	2800
5.5K bytes	2A00
5.0K bytes	2C00
4.5K bytes	2E00
4.0K bytes	3000
3.5K bytes	3200
3.0K bytes	3400
2.5K bytes	3600
2.0K bytes	3800
1.5K bytes	3A00
1.0K bytes	3C00
0.5K bytes	3E00
No ISP Space	--

HWBS:

- ☒ : Enabled. When powered up, MCU will boot from ISP-memory if ISP-memory is configured.
☐ : Disabled. MCU always boots from AP-memory.

HWBS2:

- ☒ : Enabled. Activating the RST-pin can induce MCU to boot from ISP-memory if ISP-memory is configured.
☐ : Disabled. Where MCU boots from is determined by HWBS.

IAP-memory Space:

The IAP-memory space specifies the user-defined IAP space. The IAP-memory space can be configured by hardware option or MCU software by modifying IAP_LB. In default, it is configured to **1K** bytes.

BO1S10, BO1S00:

- ☒ , ☒ : Select BOD1 to detect 2.0V.

- ☒, ☐: Select BOD1 to detect 2.4V.
☐, ☒: Select BOD1 to detect 3.7V.
☐, ☐: Select BOD1 to detect 4.2V.

BO0REO:

- ☒: Enabled. BOD0 will trigger a RESET event to CPU on AP program start address. (1.7V)
☐: Disabled. BOD0 cannot trigger a RESET to CPU.

BO1REO:

- ☒: Enabled. BOD1 will trigger a RESET event to CPU on AP program start address. (4.2V, 3.7V, 2.4V or 2.0V)
☐: Disabled. BOD1 cannot trigger a RESET to CPU.

WRENO:

- ☒: Enabled. Set WDTCSR.WREN to enable as system reset function by WDTCR.
☐: Disabled. Clear WDTCSR.WREN to disable the system reset function by WDTCR.

NSWDT: Non-Stopped WDT

- ☒: Enabled. Set WDTCSR.NSW to enable the WDT running in powerdown mode (watch mode).
☐: Disabled. Clear WDTCSR.NSW to disable the WDT running in powerdown mode (disable Watch mode).

HWENW: Hardware loaded for "ENW" of WDTCSR.

- ☒: Enabled. Enable WDT and load the content of WRENO, NSWDT, HWWIDL and HWPS2~0 to WDTCSR after power-on.
☐: Disabled. WDT is not enabled automatically after power-on.

HWWIDL, HWPS2, HWPS1, HWPS0:

When HWENW is enabled, the content on these four fused bits will be loaded to WDTCSR SFR after power-on.

WDSFWP:

- ☒: Enabled. The WDT SFRs, WREN, NSW, WIDL, PS2, PS1 and PS0 in WDTCSR, will be write-protected.
☐: Disabled. The WDT SFRs, WREN, NSW, WIDL, PS2, PS1 and PS0 in WDTCSR, are free for writing of software.

32 Application Notes

32.1 Power Supply Circuit

To have the CMT2380F17 work with power supply varying from 2.0V to 5.5V, adding some external decoupling and bypass capacitors is necessary, as shown in Figure 32–1.

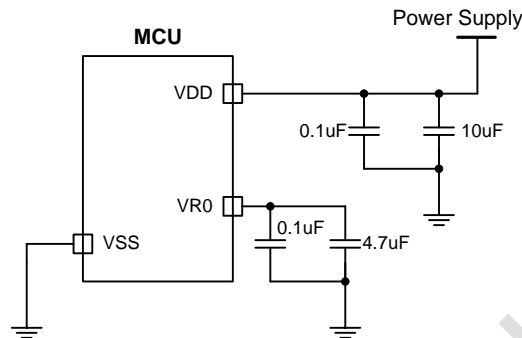


Figure 32-1. Power Supplied Circuit

32.2 Reset Circuit

Normally, the power-on reset can be successfully generated during power-up. However, to further ensure the MCU a reliable reset during power-up, the external reset is necessary. Figure 32–2 shows the external reset circuit, which consists of a capacitor C_{EXT} connected to VDD (power supply) and a resistor R_{EXT} connected to VSS (ground).

In general, R_{EXT} is optional because the RST pin has an internal pull-down resistor (R_{RST}). This internal diffused resistor to VSS permits a power-up reset using only an external capacitor C_{EXT} to VDD.

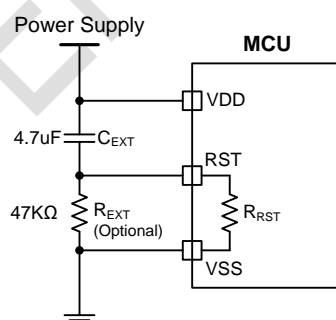


Figure 30-2. Reset Circuit

32.3 ICP and OCD Interface Circuit

CMT2380F17 devices include an on-chip Megawin proprietary debug interface to allow In-Chip-Programming (ICP) and in-system On-Chip-Debugging (OCD) with the production part installed in the end application. The ICP and OCD share the same interface to use a clock signal (ICP_SCL/OCD_SCL) and a bi-directional data signal (ICP_SDA/OCD_SDA) to transfer information between the device and a host system.

The ICP interface allows the ICP_SCL/ICP_SDA pins to be shared with user functions so that In-Chip Flash Programming function could be performed. This is practicable because ICP communication is performed when the device is in the halt state, where the on-chip peripherals and user software are stalled. In this halted state, the ICP interface can safely 'borrow' the ICP_SCL (P4.4) and ICP_SDA (P4.5) pins. In most applications, external resistors are required to isolate ICP interface traffic from the user application. A typical isolation configuration is shown in Figure 32– 3.

It is strongly recommended to build the ICP interface circuit on target system. It will reserve the whole capability for software programming and device options configured.

After power-on, the P4.4 and P4.5 of CMT2380F17 are configured to OCD_SCL/OCD_SDA for in-system On-Chip-Debugging function. This is possible because OCD communication is typically performed when the CPU is in the halt state, where the user software is stalled. In this halted state, the OCD interface can safely 'use' the OCD_SCL (P4.4) and OCD_SDA (P4.5) pins. As the mentioned ICP interface isolation in Figure 32–3, external resistors are required to isolate OCD interface traffic from the user application.

If user gives up the OCD function, software can configure the OCD_SCL and OCD_SDA to port pins: P4.4 and P4.5 by clearing OCDE on bit 0 of DCON0. When user would like to regain the OCD function, user can predict an event that triggers the software to switch the P4.4 and P4.5 back to OCD_SCL and OCD_SDA by setting OCED as "1". Or "Erase" the on-chip flash by ICP which cleans the user software to stop the port pins switching.

However, for the CMT2380F17 SOP8 package it does not support ICP due to the pin limitation. For SOP8 package, it is necessary to use other package within code development phase. Once the code has been done and then use the ISP to download the code for physical evaluation. .

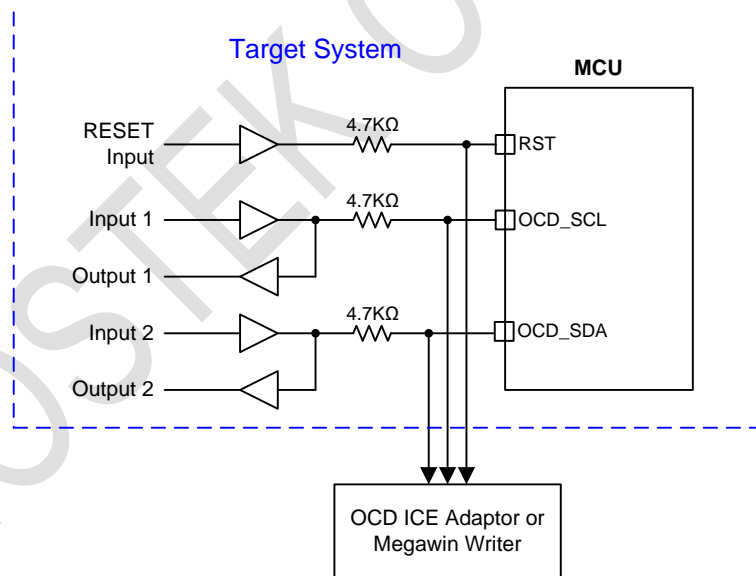


Figure 32-3. ICP and OCD Interface Circuit

32.4 In-Chip-Programming Function

The ICP, like the traditional parallel programming method, can be used to program anywhere in the MCU, including the Flash and MCU's Hardware Option. And, owing to its dedicated serial programming interface (via the On-Chip Debug path), the ICP can update the MCU without removing the MCU chip from the actual end product, just like the ISP does.

The proprietary 6-pin “Megawin 8051 ICE Adapter” can support the In-Circuit Programming of CMT2380F17. “Megawin 8051 ICE Adapter” has the in-system storage to store the user program code and device options. So, the tools can perform a portable and stand-alone programming without a host on-line, such as connecting the tool to PC. Following lists the features of the ICP function:

- No need to have a loader program pre-programmed in the target MCU.
- Dedicated serial interface; no port pin is occupied.
- The target MCU needn't be in running state; it just needs to be powered.
- Capable of portable and stand-alone working without host's intervention.

The above valuable features make the ICP function very friendly to the user. Particularly, it is capable of stand-alone working after the programming data is downloaded. This is especially useful in the field without a PC. The system diagrams of the ICP function for the stand-alone programming are shown in Figure 32–4. Only five pins are used for the ICP interface: the SDA line and SCL line function as serial data and serial clock, respectively, to transmit the programming data from the 6-pin “Megawin 8051 ICE Adapter” to the target MCU; the RST line to halt the MCU, and the VCC & GND are the power supply entry of the 6-pin “Megawin 8051 ICE Adapter” for portable programming application. The USB connector can be directly plugged into the PC's USB port to download the programming data from PC to the 6-pin “Megawin 8051 ICE Adapter”.

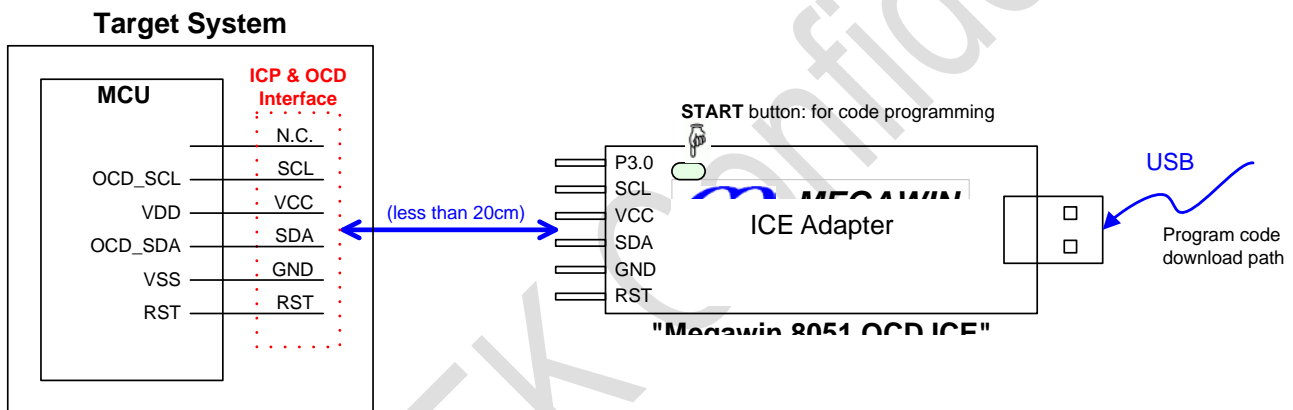


Figure 32-4. Stand-alone programming via ICP

32.5 On-Chip-Debug Function

The CMT2380F17 is equipped with a Megawin proprietary On-Chip Debug (OCD) interface for In-Circuit Emulator (ICE). The OCD interface provides on-chip and in-system non-intrusive debugging without any target resource occupied.

Several operations necessary for an ICE are supported, such as Reset, Run, Stop, Step, Run to Cursor and Breakpoint Setting.

Using the OCD technology, Megawin provides the “Megawin 8051 OCD ICE” for the user, as shown in Figure 32–5. The user has no need to prepare any development board during developing, or the socket adapter used in the traditional ICE probe. All the thing the user needs to do is to reserve a 6-pin connector on the system for the dedicated OCD interface: P3.0, RST, VCC, OCD_SDA, OCD_SCL and GND as shown in Figure 32–5.

In addition, the most powerful feature is that it can directly connect the user's target system to the Keil 8051 IDE software for debugging, which directly utilizes the Keil IDE's dScope-Debugger function. Of course, all the advantages are based on your using Keil 8051 IDE software.。

Note: “Keil” is the trade mark of “Keil Elektronik GmbH and Keil Software, Inc.”.

● Features

- OCD (On-Chip-Debug) technology
- On-chip & in-system real-time debugging
- 5-pin dedicated serial interface for OCD, no target resource occupied
- Directly linked to the debugger function of the Keil 8051 IDE Software
- USB connection between target and host (PC)
- Helpful debug actions: Reset, Run, Stop, Step and Run to Cursor
- Programmable breakpoints, up to 4 breakpoints can be inserted simultaneously
- Several debug-helpful windows: Register/Disassembly/Watch/Memory Windows
- Source-level (Assembly or C-language) debugging capability

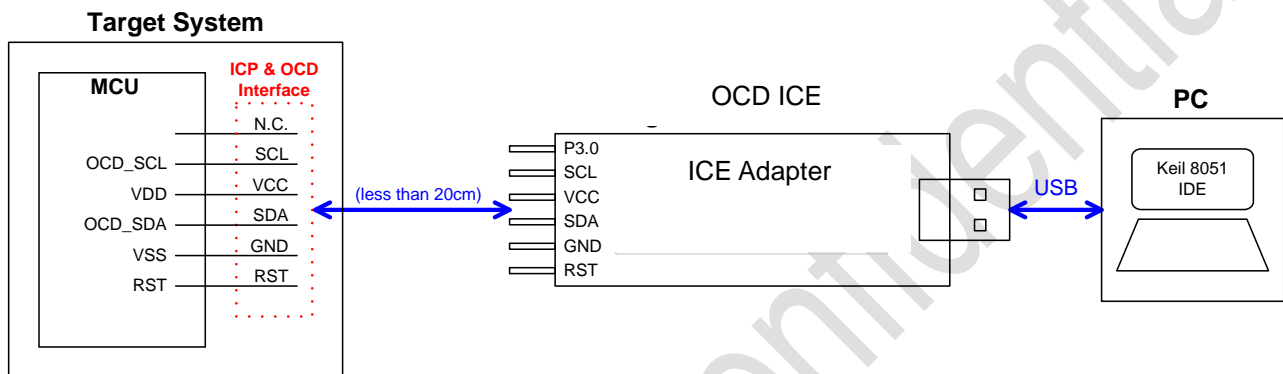


Figure 32-5. System Diagram for the ICE Function

33 Instruction Set

MNEMONIC	DESCRIPTION	BYTE	EXECUTION Cycles
DATATRASFER			
MOVA,Rn	Move register to Acc	1	1
MOVA,direct	Move direct byte to Acc	2	2
MOVA,@Ri	Move indirect RAM to Acc	1	2
MOVA,#data	Move immediate data to Acc	2	2
MOVRn,A	Move Acc to register	1	2
MOVRn,direct	Move direct byte to register	2	4
MOVRn,#data	Move immediate data to register	2	2
MOVdirect,A	Move Acc to direct byte	2	3
MOVdirect,Rn	Move register to direct byte	2	3
MOVdirect,direct	Move direct byte to direct byte	3	4
MOVdirect,@Ri	Move indirect RAM to direct byte	2	4
MOVdirect,#data	Move immediate data to direct byte	3	3
MOV@Ri,A	Move Acc to indirect RAM	1	3
MOV@Ri,direct	Move direct byte to indirect RAM	2	3
MOV@Ri,#data	Move immediate data to indirect RAM	2	3
MOVDPTR,#data16	Load DPTR with a 16-bit constant	3	3
MOVCA,@A+DPTR	Move code byte relative to DPTR to Acc	1	4
MOVCA,@A+PC	Move code byte relative to PC to Acc	1	4
MOVXA,@Ri	Move on-chip auxiliary RAM (8-bit address) to Acc	1	3
MOVXA,@DPTR	Move on-chip auxiliary RAM (16-bit address) to Acc	1	3
MOVX@Ri,A	Move Acc to on-chip auxiliary RAM (8-bit address)	1	3
MOVX@DPTR,A	Move Acc to on-chip auxiliary RAM (16-bit address)	1	3
MOVXA,@Ri	Move external RAM (8-bit address) to Acc	1	not support
MOVXA,@DPTR	Move external RAM (16-bit address) to Acc	1	not support
MOVX@Ri,A	Move Acc to external RAM (8-bit address)	1	not support
MOVX@DPTR,A	Move Acc to external RAM (16-bit address)	1	not support
PUSHdirect	Push direct byte onto Stack	2	4
POPdirect	Pop direct byte from Stack	2	3
XCHA,Rn	Exchange register with Acc	1	3
XCHA,direct	Exchange direct byte with Acc	2	4
XCHA,@Ri	Exchange indirect RAM with Acc	1	4
XCHDA,@Ri	Exchange low-order digit indirect RAM with Acc	1	4
ARITHMETIC OPERATIONS			
ADDA,Rn	Add register to Acc	1	2
ADDA,direct	Add direct byte to Acc	2	3
ADDA,@Ri	Add indirect RAM to Acc	1	3
ADDA,#data	Add immediate data to Acc	2	2
ADDCA,Rn	Add register to Acc with Carry	1	2
ADDCA,direct	Add direct byte to Acc with Carry	2	3
ADDCA,@Ri	Add indirect RAM to Acc with Carry	1	3

ADDCA,#data	AddimmediatedatatoAccwithCarry	2	2
SUBBA,Rn	SubtractregisterfromAccwithborrow	1	2
SUBBA,direct	SubtractdirectbytefromAccwithborrow	2	3
SUBBA,@Ri	SubtractindirectRAMfrom Accwith borrow	1	3
SUBBA,#data	SubtractimmediatedatafromAccwithborrow	2	2
INCA	IncrementAcc	1	2
INCRn	Incrementregister	1	3
INCdirect	Incrementdirectbyte	2	4
INC@Ri	IncrementindirectRAM	1	4
DECA	DecrementAcc	1	2
DECRn	Decrementregister	1	3
DECdirect	Decrementdirectbyte	2	4
DEC@Ri	DecrementindirectRAM	1	4
INCDPTR	IncrementDPTR	1	1
MULAB	MultiplyAand B	1	4
DIVAB	Divide AbyB	1	5
DAA	DecimalAdjustAcc	1	4

LOGICOPERATION

ANLA,Rn	ANDregistertoAcc	1	2
ANLA,direct	ANDdirectbyte toAcc	2	3
ANLA,@Ri	ANDindirect RAMtoAcc	1	3
ANLA,#data	ANDimmediatedatatoAcc	2	2
ANLdirect,A	ANDAcc todirectbyte	2	4
ANLdirect,#data	ANDimmediatedatatodirectbyte	3	4
ORLA,Rn	ORregisterto Acc	1	2
ORLA,direct	ORdirectbyte to Acc	2	3
ORLA,@Ri	ORindirect RAMtoAcc	1	3
ORLA,#data	ORimmediatedatatoAcc	2	2
ORLdirect,A	ORAcc todirectbyte	2	4
ORLdirect,#data	ORimmediatedatatodirectbyte	3	4
XRLA,Rn	Exclusive-ORregistertoAcc	1	2
XRLA,direct	Exclusive-ORdirectbytetoAcc	2	3
XRLA,@Ri	Exclusive-ORindirectRAMto Acc	1	3
XRLA,#data	Exclusive-ORimmediatedatatoAcc	2	2
XRLdirect,A	Exclusive-ORAcctodirectbyte	2	4
XRLdirect,#data	Exclusive-ORimmediatedatatodirectbyte	3	4
CLRA	ClearAcc	1	1
CPLA	ComplementAcc	1	2
RLA	Rotate AccLeft	1	1
RLCA	RotateAccLeftthroughtheCarry	1	1
RRA	RotateAccRight	1	1
RRCA	RotateAccRightthroughtheCarry	1	1
SWAPA	SwapnibbleswithintheAcc	1	1

BOOLEANVARIABLEMANIPULATION

CLRC	ClearCarry	1	1
CLRbit	Cleardirectbit	2	4
SETBC	SetCarry	1	1
SETBbit	Setdirectbit	2	4

CPLC	ComplementCarry	1	1
CPLbit	Complementdirectbit	2	4
ANLC,bit	ANDdirectbittoCarry	2	3
ANLC,/bit	ANDcomplementofdirectbittoCarry	2	3
ORLC,bit	ORdirectbittoCarry	2	3
ORLC,/bit	ORcomplementofdirectbittoCarry	2	3
MOVC,bit	Move directbitto Carry	2	3
MOVbit,C	MoveCarrytodirectbit	2	4
BOOLEANVARIABLEMANIPULATION			
JC rel	Jump ifCarryisset	2	3
JNCrel	Jump ifCarrynotset	2	3
JBbit,rel	Jump ifdirectbitisset	3	4
JNBbit,rel	Jump ifdirectbitnotset	3	4
JBCbit,rel	Jump ifdirectbitissetandthenclearbit	3	5
PROGRAMBRANCHING			
ACALLAddr11	Absolutesubroutinecall	2	6
LCALLAddr16	Longsubroutinecall	3	6
RET	Returnfromsubroutine	1	4
RETI	Returnfrominterruptsubroutine	1	4
AJMPAddr11	Absolutejump	2	3
LJMPAddr16	Longjump	3	4
SJMPrel	Shortjump	2	3
JMP@A+DPTR	JumpindirectrelativetoDPTR	1	3
JZ rel	Jumpif Acciszero	2	3
JNZrel	JumpifAccnotzero	2	3
CJNEA,direct,rel	Comparedirectbyte toAccand jumpifnotequal	3	5
CJNEA,#data,rel	CompareimmediatedatatoAccandjump ifnotequal	3	4
CJNERn,#data,rel	Compareimmediatedatatoregisterandjumpifnotequal	3	4
CJNE@Ri,#data,rel	CompareimmediatedatatoindirectRAMand jumpifnotequal	3	5
DJNZRn,rel	Decrementregisterandjumpifnotequal	2	4
DJNZdirect,rel	Decrementdirectbyteandjumpifnotequal	3	5
NOP	NoOperation	1	1

34 Ordering Information

Table 34-1. CMT2380F17 订购信息

Model	Description	Packaging	Packaging Option	Operating Condition	Minimum Order Quantity
CMT2380F17-EQR ^[1]	CMT2380F17, ultra-low power sub-1GHz wireless MCU	QFN40(5x5)	Tape and tray	1.8 to 3.6V, -40 to 85°C	3,000
Notes: [1] E refers to extended Industrial product rating, which supports temperature range from -40 to +85 °C. Q refers to the package type QFN40. R refers to tape and tray type, and the minimum order quantity (MOQ) is 3,000 pieces.					

Please visit www.cmostek.com for more product/product line information.

Please contact sales@cmotek.com or your local sales representative for sales or pricing requirements.

35 Packaging Information

The packaging information of the CMT2380F17 is shown in the below figure.

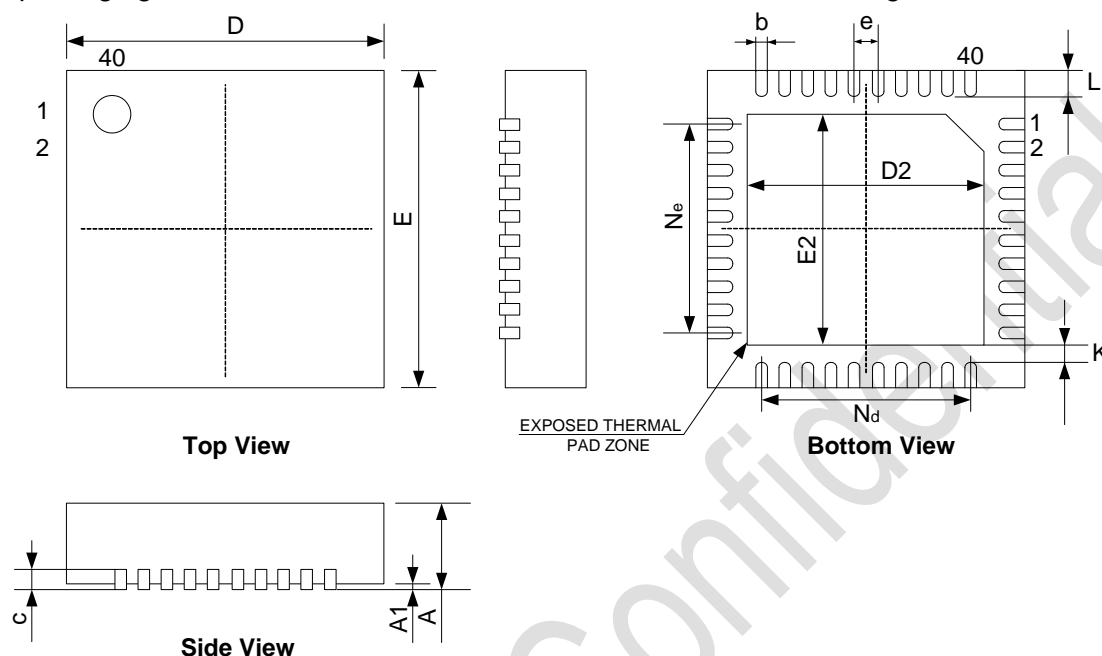


Figure 35-1. QFN405x5 Packaging

Table 35-1. QFN405x5 Packaging Scale

Symbol	Scale (mm)		
	Min.	Typ.	Max.
A	0.70	0.75	0.80
A1	0	0.02	0.05
b	0.15	0.20	0.25
b1	0.14REF		
C	0.18	0.20	0.25
D	4.90	5.00	5.10
D2	3.60	3.70	3.80
e	0.40 BSC		
Ne	3.60 BSC		
Nd	3.60 BSC		
E	4.90	5.00	5.10
E2	3.60	3.70	3.80
L	0.35	0.40	0.45
L1	0.10REF		
K	0.20	-	-
h	0.30	0.35	0.40

36Top Marking

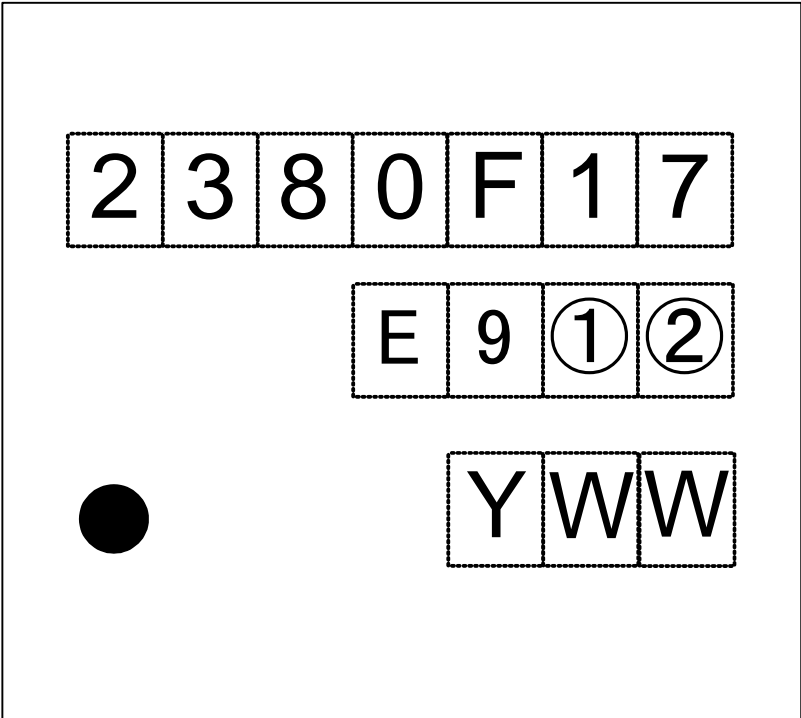


Figure 36-1. The CMT2380F17 Top Marking

Table 36-2. The CMT2380F17 Top Marking Information

Marking Method	Laser
Pin 1 Mark	Diameter of the circle = 0.3 mm
Font Size	0.5 mm, align right
Line 1 Marking	2380F17referring to model CMT2380F17
Line 2 Marking	E9①② is internal tracing code
Line 3 Marking	The date code is assigned by the package factory. Y is the last digit of the year. WW is the working week.

37 Reference Documents

Table 37-1.Reference Documents

Doc No.	Name	Description
AN141	CMT2300A Schematic and PCB Layout Guide	CMT2380F17 RF matching design guidelines.
AN142	CMT2300A Quick Start Guide	CMT2380F17 RF quick start guidelines.
AN143	CMT2300A FIFO and Packet Format Usage Guide	CMT2380F17 RF transceiver message usage guide.
AN144	CMT2300A RSSI User Guide	CMT2380F17 RF RSSI user guide.
AN146	CMT2300A Low Power Mode User Guide	CMT2380F17 RF Low power design guidelines.
AN147	CMT2300A Features Usage Guide	CMT2380F17 RF features description.
AN149	CMT2300A RF Parameter Configuration Guide	CMT2380F17 RF Frequency matching parameters.
AN150	CMT2300A Low Voltage Transmit Power Compensation	CMT2380F17 RF Low-voltage transmit power compensation description.
AN197	CMT2300A-CMT2119B-CMT2219B Fast Manual Frequency Hopping	CMT2380F17 fast manual frequency hopping description.
AN198	CMT2300A-CMT2119B-CMT2219B Status Switching Precautions	CMT2380F17 RF status switching considerations.
AN199	CMT2300A-CMT2119B-CMT2219B RF Frequency Calculation Guide	CMT2380F17 RF frequency calculation description.

38 Revise History

Table 38-1. Revise Records

Version No.	Chapter	Description	Date
0.1	All	Initial version	2021-10-19
0.2	All	Update part number error	2022-7-14
0.3	All	Update contacts	2022-9-14

39 Contacts

Shenzhen Hope Microelectronics Co., Ltd.

Address: 30th floor of 8th Building, C Zone, Vanke Cloud City, Xili Sub-district, Nanshan, Shenzhen, GD, P.R. China

Tel: +86-755-82973805 / 4001-189-180

Fax: +86-755-82973550

Post Code: 518052

Sales: sales@hoperf.com

Website: www.hoperf.com

Copyright. Shenzhen Hope Microelectronics Co., Ltd. All rights are reserved.

The information furnished by HOPERF is believed to be accurate and reliable. However, no responsibility is assumed for inaccuracies and specifications within this document are subject to change without notice. The material contained herein is the exclusive property of HOPERF and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of HOPERF. HOPERF products are not authorized for use as critical components in life support devices or systems without express written approval of HOPERF. The HOPERF logo is a registered trademark of Shenzhen Hope Microelectronics Co., Ltd. All other names are the property of their respective owners.