MG82F6P32
Data Sheet

8Bit Single-Chip Microcontroller
Embedded OPA / PGA / ACMP

Version: V1.00

Features

- 1-T 80C51 Central Processing Unit
- **MG82F6P32** with **32K** Bytes flash ROM
  - ISP memory zone could be optioned as **0.5KB/1.0KB~7.5KB**
  - Flexible IAP size by software configured
  - Code protection for flash memory access
  - Flash write/erase cycle: 20,000
  - Flash data retention: 100 years at 25°C
  - **Default MG82F6P32 Flash space mapping**
    - AP Flash default mapping (29.5KB, 0000h~75FFh)
    - IAP Flash default mapping (1.0KB, F600h~F9FFh)
    - ISP Flash default mapping (1.5KB, FA00h~FFFFh), ISP Boot code
- Data RAM: **2K** Bytes
  - On-chip 256 bytes scratch-pad RAM
  - **MG82F6P32 - 1792** bytes expanded RAM (XRAM)
  - Support page select on XRAM access
- Dual data pointer
- Provide one channel DMA engine
  - P2P, M2P, P2M
  - Memory target: on-chip XRAM
  - Peripheral target: UART0/1, SPI, TWI0/I2C0, I2C1, ADC12 & CRC16/32
  - Timer 5 and Timer 6 are used for DMA, but it also can be traded as independent timer when DMA not in use
- Interrupt controller
  - **21** sources, four-level-priority interrupt capability
  - **Four** external interrupt inputs, nINT0, nINT1, nINT2 and nINT3 with glitch filter
  - All external interrupts support High/Low level or Rising/Falling edge trigger
  - AC0, AC1, OPA0 and OPA1 support MCU wakeup from Power Down mode
- Total **12/14** (with split mode) timers in **MG82F6P32**
  - RTC Timer and WDT Timer
  - Timer 0, Timer 1, Timer 2 and Timer 3
  - PTM0 (PWM Timer 0), PCA1 (Program Counter Array 1)
  - S0 BRG and S1 BRG
  - Timer 5/6 of DMA module also can be used as timer
  - If Timer 2/3 in split mode, **MG82F6P32** has total **14** timers
- **6** 16-bit timer/counters, Timer 0, Timer 1, Timer 2, Timer 3, Timer 5 and Timer 6
  - X12 mode and timer clock output function
  - New **6** operating modes in Timer 2/3 with 8 clock sources and 8 capture sources
  - Timer 2/3 can be split to two 8-bit timers
  - Clock Count Output (CCO) on T2CKO, T3CKO
  - Timer 0~3 support PWM mode
  - Timer 2/3 support Duty Capture function
- **One** 16-bit PWM Timer (PTM0)
  - PTM0 has 6 CP (Compare/PWM) modules
  - Reloadable 16-bit base counter to support variable length PWM
  - Up to **96MHz** clock source from on-chip CKM
  - 16-bit software timer mode and High-speed output mode
  - Variable 8/10/12/16-bit PWM mode, the PTM0 can be configured to:
    - Up to **6** channels un-buffered 10/12/16-bit PWM, or
    - Up to **6** channels buffered 2~8-bit PWM for various frequency setting, or
    - Up to **3** channels buffered 9~16-bit PWM for various frequency setting
  - PWM with dead-time control, break control and central-aligned option
- **One** Programmable 16-bit counter/timer Arrays (PCA1)

- PCA1 has 2 CCP (Capture/Compare/PWM) modules
- Reloadable 16-bit base counter to support variable length PWM
- Up to **96MHz** clock source from on-chip CKM
- Capture mode, 16-bit software timer mode and High-speed output mode
- Buffered capture mode to monitor narrow pulse input
- Variable 8/10/12/16-bit PWM mode, the PCA1 can be configured to:
  - Up to **2** channels un-buffered 10/12/16-bit PWM, or
  - Up to **2** channels buffered 2~8-bit PWM for various frequency setting, or
  - Up to **1** channel buffered 9~16-bit PWM for various frequency setting
- PWM with dead-time control, break control and central-aligned option

- Timer2/3, PTM0 and PCA1 has global control for output signal synchronization
- 8 Inputs Keypad Interrupt (KBI)
- 12-Bit Single-ended ADC
  - Programmable throughput up to **1M** sps (room temperature)/ **500K** sps (Full temperature)
  - **MG82F6P32** has **8** channel external inputs and **5** channels for internal reference voltage (IVR/ 2.4V), PGAO, OP0O, OP1O, 1/4VDD and internal VSS
  - Support window detect function on ADC result
  - Support channel scan mode
  - ADC VREF+ from internal IVR 2.4V
- On-chip voltage reference (IVR 24)
- 2 Operational Amplifiers OPA (OPA0/ OPA1)
  - Low input offset, 0.5mV ~ 1mV after calibration, 3mV factory trimmed.
  - Internal 1.2V reference on positive and negative input option to be DC bias or used for offset calibration
  - Support Input and Output I/O
  - PGA output on negative input as pre-amplifier.
  - Internal output path connects to ADC and AC0 input
  - Software offset calibration with 6-bit trimming range.
  - Low power mode
  - Support comparison mode
  - Comparison mode support interrupt and can be used under power down for edge wakeup trigger source.
- Programmable Gain Amplifier (PGA)
  - Low input offset, 0.5mV ~ 1mV after calibration, 3mV factory trimmed.
  - Support Input and Output I/O
  - Gain: x1, x2, x4, x8, x16, x32, x64, x128
  - Low power mode
  - Software offset calibration with 6-bit trimming range.
- Analog Comparator 0 (AC0)
  - Selectable internal voltage reference (IVR/2.4V) on ACNI0
  - **4** selectable ACPI0(+) inputs
  - Support OPAn output to ACPI0/ACNI0 for the signal comparison
  - Wake-up from power-down and idle
  - Glitch filter option and output to internal timer capture
  - Hysteresis Voltage: 0mV, ±10mV, ±20mV and ±60mV
  - 6 Bits Offset trimming
- Analog Comparator 1 (AC1)
  - Selectable internal voltage reference (IVR/2.4V)
  - Wake-up from power-down and idle
  - Glitch filter option and output to internal timer capture
  - Hysteresis Voltage: 0mV, ±10mV, ±20mV and ±60mV
  - 6 Bits Offset trimming
- Enhanced UART (S0)
  - Framing Error Detection
  - Automatic Address Recognition
  - Max. UART baud rate up to 6MHz
  - Support SPI Master in Mode 4, up to 12MHz on SPICLK

  - Built-in baud rate generator (S0BRG) to support TX or RX on different baud rate
  - S0BRG in timer mode cascaded with Timer 0/1 to be a 16/24-bit timer/counter
  - Support ARGB data transition
- 2nd UART (S1)
  - Max. UART baud rate up to 1.8432/3.0MHz
  - Support SPI Master in Mode 4, up to 12MHz on SPICLK
  - Provide BMC ENDEC for USB PD application
  - Support ARGB data transition
- One Master/Slave SPI serial interface
  - Max. 24MHz SPICLK on SPI master
  - Max 12MHz on SPI slave
  - Up to 3 SPI masters including S0/S1 in mode 4
  - Support daisy-chain function in SPI slave mode (mode 1)
- Three Master/Slave two wire serial interfaces: TWI0/I2C0, TWI1/I2C1 and STWI (SI2C)
  - Two Master/Slave hardware engine: TWI0/I2C0 and TWI1/I2C1
  - Max. 1MHz on I2C0/I2C1 master mode and Max. 400KHz on I2C0/I2C1 slave mode
  - One software TWI/I2C, STWI/ SI2C, Start/Stop serial interface detection (SID)
  - Multiple slave address recognition on I2C0/I2C1
- Programmable Watchdog Timer (WDT), clock sourced from ILRCO, XTAL or SYSCLK/12
  - One time enabled by CPU or power-on
  - Interrupt CPU or Reset CPU on WDT overflow
  - Support WDT function in power down mode (watch mode) for auto-wakeup function
- Real-Time-Clock (RTC) module, clock sourced from XTAL, ILRCO, WDTPS, WDTOF, SYSCLK or SYSCLK/12
  - Programmable interrupt period from mini-second wakeup to minute wakeup
  - 21-bit length system timer
- Beeper function
- General purpose logic (GPL/CRC)
  - Bit order reversed function
  - 16-bit CRC engine (polynomial: **0x1201**)
  - Support automatic CRC of flash content
  - Programmable initial seed function of CRC
  - 4b5b encoder/decoder (ENDEC)
  - 32-bit CRC engine (polynomial: **0x04C1_1DB7**)
- On-Chip-Debug interface (OCD)
- Maximum **29/25** GPIOs in 32/28-pin package
  - All I/O is Analog I/O mode as default
  - P3 can be configured to quasi-bidirectional, push-pull output, open-drain output and input only
  - P0, P1, P2, P4 and P6 can be configured to open-drain output or push-pull output
  - P6.0, P6.1 and P4.7 shared with XTAL2, XTAL1 and **nRST**
  - Programmable GPIO driving strength and driving speed
  - On chip pull-up enabled on each pin
- Clock Sources
  - Internal 12MHz/11.059MHz oscillator (IHRCO): factory calibrated to ±1%, typical
  - External crystal mode, support 32.768KHz oscillating and missing clock detection (MCD)
  - Internal Low power 32KHz RC Oscillator (ILRCO)
  - External clock input (ECKI) on P6.0/XTAL2, up to 25MHz
  - Internal RC Oscillator output on P6.0/XTAL2
  - On-chip Clock Multiplier (CKM) to provide high speed clock source 96MHz
- Two Brown-Out Detectors
  - BOD0: detect **1.7V**
  - BOD1: selected detection level on 4.2V/3.6V/2.4V/2.0V
  - Interrupt CPU or reset CPU
  - Wake up CPU in Power-Down mode (BOD1)

- Multiple power control modes: idle mode, power-down mode, slow mode, sub-clock mode, RTC mode, watch mode and monitor mode.
  - All interrupts can wake up IDLE mode
  - **19** sources with **23** pins to wake up Power-Down mode
  - Slow mode and sub-clock mode support low speed MCU operation
  - RTC mode supports RTC to resume CPU in power down
  - Watch mode supports WDT to resume CPU in power down
  - Monitor mode supports BOD1 to resume CPU in power down
- Operating voltage range: 1.8V – 5.5V
  - Minimum **1.8V** requirement in flash write operation (ISP/IAP/ICP)
- Operation frequency range: **32MHz** (max)
  - External clock input mode, 0 – 12MHz @ 2.0V – 5.5V, 0 – 25MHz @ 2.4V – 5.5V
  - CPU up to 12MHz @ **1.8V** – 5.5V, and up to 25MHz @ **2.2V** – 5.5V
  - **CPU up to 32MHz @ 2.7V -5.5V with on-chip CKM**
  - **Peripherial Clock (SYSCLK) up to 48MHz**
- 16-Bytes Unique ID code
- Operating Temperature:
  - Industrial (-40°C to +105°C)*
- Package Types:
  - LQFP32 (7 x 7 mm): MG82F6P32AD32 (32K)
  - QFN32 (4 x 4 x 0.55 mm): MG82F6P32AZ32 (32K)
  - SSOP28 (150 mil): MG82F6P32AL28 (32K)

*: Tested by sampling.

List of Contents

List of Figures

List of Tables

1. General Description

The **MG82F6P32** is a single-chip microcontroller based on a high performance 1-T architecture 80C51 CPU that executes instructions in 1~7 clock cycles (about 6~7 times the rate of a standard 8051 device), and has an 8051 compatible instruction set. Therefore at the same performance as the standard 8051, the **MG82F6P32** can operate at a much lower speed and thereby greatly reduce the power consumption.

The **MG82F6P32** has **32K** 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 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.

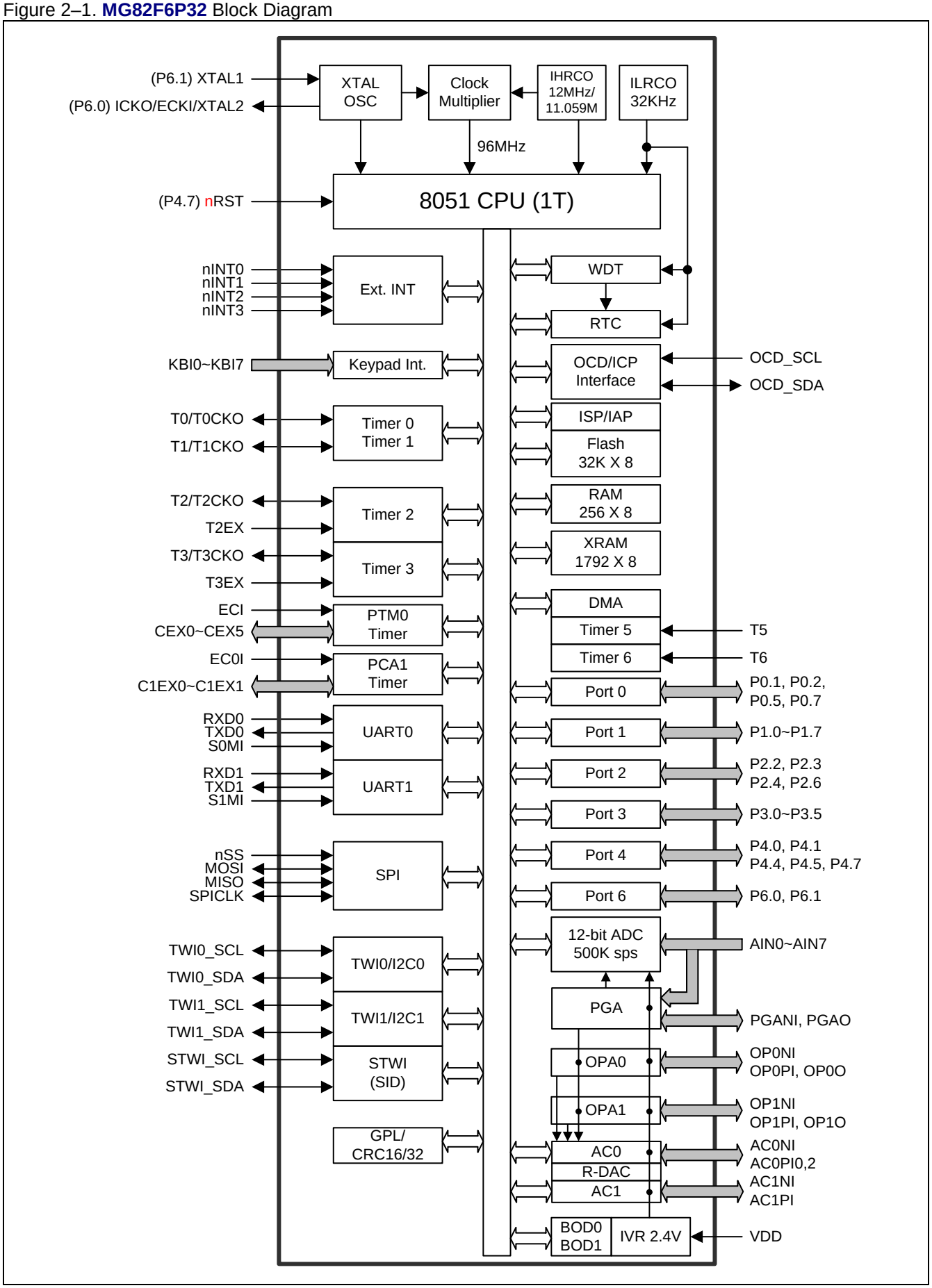
The **MG82F6P32** retains all features of the standard 80C52 with 256 bytes of scratch-pad RAM, **four** external interrupts with High/Low trigger option, a multi-source 4-level interrupt controller, 2 serial port (UART0/1) with ARGB and BMC ENDEC for USB PD protocol detection and **four** timer/counters. In addition, the **MG82F6P32** has **29** I/O port pins, one XRAM of **1792** bytes, **500K sps** 12-bit ADC, 2 OPA, 1 PGA, 5 16-bit timer, one 6-channel PTM with dead-time controlled PWM, one 2-channel PCA with dead-time controlled PWM, one 8-bit SPI, three TWI/I2C (TWI0/I2C0, TWI1/I2C1 and STWI/ SI2C), keypad interrupt, two Analog Comparators, Watchdog Timer, Real-Time-Clock module, two Brown-out Detectors, an ECKI external clock input (P6.0), an on-chip crystal oscillator(shared with P6.0 and P6.1), an internal high precision oscillator (IHRCO), an on-chip clock multiplier (CKM) to generate high speed clock source, an internal low speed RC oscillator (ILRCO) and two serial ports (UART0 ~ 1) which UART0 has enhanced serial function that facilitates multiprocessor communication, and a speed improvement mechanism (X2 mode). Support 3 different DMA transfer types, M2P (XRAM to Peripheral), P2M (Peripheral to XRAM) and P2P (Peripheral to Peripheral) to enhance transfer performance and reduce CPU loading.

The **MG82F6P32** has multiple operating modes to reduce the power consumption: idle mode, power down mode, slow mode, sub-clock mode, RTC mode, watch mode and monitor mode. In the Idle mode the CPU is frozen while the peripherals and the interrupt system are still operating. In the Power-Down mode the RAM and SFRs' value are saved and all other functions are inoperative; most importantly, in the Power-down mode the device can be waked up by many interrupt or reset sources. In slow mode, the user can further reduce the power consumption by using the 8-bit system clock pre-scaler to slow down the operating speed. Or select sub-clock mode which clock source is derived from internal low speed oscillator (ILRCO) for CPU to perform an ultra-low speed operation. The RTC module supports Real-Time-Clock function in all operating modes. In watch mode, it keeps WDT running in power-down or idle mode and resumes CPU as an auto-wakeup timer when WDT overflows. Monitor mode provides the Brown-Out detection in power down mode and resumes CPU when chip VDD reaches the specific detection level.

Additionally, the **MG82F6P32** is equipped with the 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. The user has no need to prepare any development board during firmware developing or the socket adapter used in the traditional ICE probe head. All the thing the user needs to do is to prepare a connector for the dedicated OCD interface. This powerful feature makes the developing very easy for any user.

2. Block Diagram

Figure 2–1. **MG82F6P32** Block Diagram



3. Special Function Register

3.1. SFR Map (Page 0~F)

Table 3–1. SFR Map (Page 0~F)

		0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
F8	0		CH	CCAP0H	CCAP1H	CCAP2H	CCAP3H	CCAP4H	CCAP5H
	1	*P6*	CH1	C1CAP0H	C1CAP1H				
	2~F								
F0	0	*B*	PAOE	PCAPWM0	PCAPWM1	PCAPWM2	PCAPWM3	PCAPWM4	PCAPWM5
	1			C1PWM0	C1PWM1				
	2~7								
	8		C0AOE1						
	9		C1AOE0						
	A		C1AOE1						
	B~F		--	--	--				
E8	0	*P4*	CL	CCAP0L	CCAP1L	CCAP2L	CCAP3L	CCAP4L	CCAP5L
	1		CL1	C1CAP0L	C1CAP1L				
	2~F								
E0	0~F	*ACC*	WDTCR	IFD	IFADRH	IFADRL	IFMT	SCMD	ISPCR
D8	0	*CCON*	CMOD	CCAPM0	CCAPM1	CCAPM2	CCAPM3	CCAPM4	CCAPM5
	1	C1CON	C1MOD	C1CAPM0	C1CAPM1				
	2~F								
D0	0	*PSW*	SIADR	SIDAT	SISTA	SICON	KBPATN	KBCON	KBMASK
	1		SI1ADR	SI1DAT	SI1STA	SI1CON			
	2		SIA2	SIA2M	--	SICR1			
	3		SI1A2	SI1A2M	--	SI1CR1			
	4~F		--	--	--	--			
C8	0	*T2CON*	T2MOD	RCAP2L	RCAP2H	TL2	TH2	CLRL	CHRL
	1	*T3CON*	T3MOD	RCAP3L	RCAP3H	TL3	TH3	CL1RL	CH1RL
	3	*T5CON*	--	TLR5	THR5	TL5	TH5	--	--
	4	*T6CON*	--	TLR6	THR6	TL6	TH6		
	5~F	--	--	**--**	--	--	--		
C0	0	*XICON*	XICFG	--	ADCFG0	ADCON0	ADCDL	ADCDH	CKCON0
	1		XICFG1		ADCFG1				
	2		XICFG2		ADCFG2				
	3		--		ADCFG3				
	4				ADCFG4				
	5				ADCFG5				
	6				--				
	7				ADCFG7				
	8~A				--				
	B				ADCFG11				
	C				ADCFG12				
	D				ADCFG13				
	E				ADCFG14				
	F				--				
B8	0	*IP0L*	SADEN/ S0CR1	OPSTA	--	PWMCR	CRC0DA	--	CKCON1
	1			OP0CON0		PDTCRA			
	2			OP1CON0		--			
	3~4			--		--			
	5			PGACON0		--			
	6			PGACON1		--			
	7			OPCMP		--			
	8~9			--		--			
	A	--		--		C1PWMCR			
	B	--		--		C1PDTCR			
	C~F	--		--		--			
B0	0	*P3*	P3M0	P3M1	P4M0	--	--	--	IP0H
	1				--	--	P6M0		
	2				--	PDRVC0			
	3				--	PDRVC1	--		
	4~F								
A8	0~F	*IE*	SADDR/ EDC45			SFRPI	EIE1	EIP1L	EIP1H

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			
	9					AUXR12			
	D					AUXR16			
	E					AUXR17			
	F								
98	0	*S0CON/ S0AGCR*	S0BUF	S0BRT	S0BRC	S0CFG	S0CFG1	AC0CON	AC0MOD
	1	*S1CON/ S1AGCR*	S1BUF	S1BRT	S1BRC/ S1CR1	S1CFG	S1CFG6	AC1CON	AC1MOD
	2								
	3						--	AC0CR1	AC0CF0
	4						--		AC1CF0
	5~F	--	--	--	--	--	--	--	--
90	0	*P1*	P1M0	P1M1	P0M0	DMACR0	P2M0	BOREV	PCON1
	1			P2M1	T2MOD1		TREN0		
	2			P4M1	T3MOD1		TRLC0		
	3			P6M1			TSPC0		
	4			P0M1	--		--		
	7			P3FDC	--		--		
	8			P1FDC	--	DMACG0	--		
	9			P2FDC	--	DMADS0	--		
	A			P4FDC	--	--	--		
	B			P0FDC					
	F			P3M2					
88	0~F	*TCON*	TMOD	TL0	TL1	TH0	TH1	SFIE	XRPS
80	0~7	*P0*	SP	DPL	DPH	SPSTAT	SPCON	SPDAT	PCON0
	8						SPCR1		
	9~F						--		
		0/8	**1/9**	**2/A**	**3/B**	**4/C**	**5/D**	**6/E**	**7/F**

***: User needs to set SFRPI as SFRPI=0x00 ~ 0x0F for SFR page access.**
（MCU will not keep SFRPI value in interrupt. User need to keep SFRPI value in software flow.）

SFRPI: SFR Page Index Register

SFR Page = 0~F

SFR Address = 0xAC RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	0	0	IDX3	IDX2	IDX1	IDX0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~4: Reserved. Software must write “0” on these bits when SFRPI is written.

Bit 3~0: SFR Page Index.

IDX[3:0]	Selected Page
0000	Page 0
0001	Page 1
0010	Page 2
0011	Page 3
……	……
……	……
……	……
1111	Page F

3.2. SFR Bit Assignment (Page 0~F)

Table 3–2. SFR Bit Assignment (Page 0~F)

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	
P0	*Port 0*	*80*	*0~F*	*P0.7*		*P0.5*			*P0.2*	*P0.1*		*10100110*
SP	Stack Pointer	81	*0~F*	.7	.6	.5	.4	.3	.2	.1	.0	00000111
DPL	Data Pointer Low	82	*0~F*	.7	.6	.5	.4	.3	.2	.1	.0	00000000
DPH	Data Pointer High	83	*0~F*	.7	.6	.5	.4	.3	.2	.1	.0	00000000
SPSTAT	SPI Status Register	84	*0~F*	SPIF	WCOL	THRF	SPIBSY	MODF				00000000
SPCON	SPI Control Register	85	*0~7*	SSIG	SPEN	DORD	MSTR	CPOL	CPHA			00000100
SPCR1	SPI Control Register 1	85	*8*	SPSOES	SPFACE			SPI0M0	SPR2	SPR1	SPR0	00000000
SPDAT	SPI Data Register	86	*0~F*	.7	.6	.5	.4	.3	.2	.1	.0	00000000
PCON0	Power Control 0	87	*0~F*	SMOD1	SMOD0	GF	POF0	GF1	GF0	PD	IDL	00010000
TCON	*Timer Control*	*88*	*0~F*	*TF1*	*TR1*	*TF0*	*TR0*	*IE1*	*IT1*	*IE0*	*IT0*	*00000000*
TMOD	Timer Mode	89	*0~F*	T1GATE	T1C/T	T1M1	T1M0	T0GATE	T0C/T	T0M1	T0M0	00000000
TL0	Timer Low 0	8A	*0~F*	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TL1	Timer Low 1	8B	*0~F*	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TH0	Timer High 0	8C	*0~F*	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TH1	Timer High 1	8D	*0~F*	.7	.6	.5	.4	.3	.2	.1	.0	00000000
SFIE	System Flag INT En.	8E	*0~F*	SIDFIE	MCDRE	MCDFIE	RTCFIE	SPWIE	BOF1IE	BOF0IE	WDTFIE	01100000
XRPS	XRAM Page Select	8F	*0~F*	0	0	0	0	0	**.2**	.1	.0	00000000
P1	*Port 1*	*90*	*0~F*	*P1.7*	*P1.6*	*P1.5*	*P1.4*	*P1.3*	*P1.2*	*P1.1*	*P1.0*	*11111111*
P1M0	P1 Mode Register 0	91	*0~F*	P1M0.7	P1M0.6	P1M0.5	P1M0.4	P1M0.3	P1M0.2	P1M0.1	P1M0.0	00000000
P1M1	P1 Mode Register 1	92	0	P1M1.7	P1M1.6	P1M1.5	P1M1.4	P1M1.3	P1M1.2	P1M1.1	P1M1.0	11111111
P2M1	P2 Mode Register 1	92	1	1	P2M1.6	1	P2M1.4	P2M1.3	P2M1.2	1	1	11111111
P4M1	P4 Mode Register 1	92	2	P4M1.7	1	P4M1.5	P4M1.4	1	1	P4M1.1	P4M1.0	11111111
P6M1	P6 Mode Register 1	92	3	1	1	1	1	1	1	P6M1.1	P6M1.0	11111111
P0M1	P0 Mode Register 1	92	4	P0M1.7	1	P0M1.5	1	1	P0M1.2	P0M1.1	1	11111111
P3FDC	P3 Fast Drv. Ctrl.	92	7	0	0	P3FDC.5	P3FDC.4	P3FDC.3	P3FDC.2	P3FDC.1	P3FDC.0	00000000
P1FDC	P1 Fast Drv. Ctrl.	92	8	P1FDC.7	P1FDC.6	P1FDC.5	P1FDC.4	P1FDC.3	P1FDC.2	P1FDC.1	P1FDC.0	00000000
P2FDC	P2 Fast Drv. Ctrl.	92	9	0	P2FDC.6	0	P2FDC.4	P2FDC.3	P2FDC.2	0	0	00000000
P4FDC	P4 Fast Drv. Ctrl.	92	A	P4FDC.7	0	P4FDC.5	P4FDC.4	0	0	P4FDC.1	P4FDC.0	00000000
P0FDC	P0 Fast Drv. Ctrl.	92	B	P0FDC.7	0	P0FDC.5	0	0	P0FDC.2	P0FDC.1	0	00000000
P3M2	P3 Mode Register 2	92	F	1	1	P3M2.5	P3M2.4	P3M2.3	P3M2.2	P3M2.1	P3M2.0	11111111
P0M0	P0 Mode Register 0	93	0	P0M0.7	0	P0M0.5	0	0	P0M0.2	P0M0.1	0	00000000
T2MOD1	Timer2 mode 1 Reg.	93	1	TL2CS	TF2IG	TL2IS	T2CKS	T2MS1	CP2S2	CP2S1	CP2S0	00000000
T3MOD1	Timer3 mode 1 Reg.	93	2	TL3CS	TF3IG	TL3IS	T3CKS	T3MS1	CP3S2	CP3S1	CP3S0	00000000
DMACR0	DMA Control Reg. 0	94	0~7	0	0	0	0	DMAE0	DMAS0	DIE0	DCF0	00000000
DMACG0	DMA Configured Reg. 0	94	8	PDMAH	PDMAL	CRCW0	0	EXTS10	EXTS00	0	LOOP0	00000000
DMADS0	DMA Data path Selection 0	94	9	DSS30	DSS20	DSS10	DSS00	DDS30	DDS20	DDS10	DDS00	00000000
P2M0	P2 Mode Register 0	95	0	0	P2M0.6	0	P2M0.4	P2M0.3	P2M0.2	0	0	00000000
TREN0	Timer Run Enable Register 0	95	1	TR2E	TR3E	0	0	0	0	C1CR	PTMCR	00000000
TRLC0	Timer Reload Control Register 0	95	2	T2RLC	T3RLC	0	0	0	0	C1RLC	PTMRLC	00000000
TSPC0	Timer Stop Control Register 0	95	3	T2SC	T3SC	0	0	0	0	C1SC	PTMSC	00000000
BOREV	Bit Order Reversed	96	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
PCON1	Power Control 1	97	0~F	SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF	00000000
S0CON	*Serial 0 Control*	*98*	*0*	*SM00 /FE*	*SM10*	*SM20*	*REN0*	*TB80*	*RB80*	*TI0*	*RI0*	*00000000*
S0AGCR	*Serial 0 ARGB Control*	*98*	*0*	*SM00*	*SM10*	*S0DTY1*	*S0GFIE*	*S0IDT11*	*S0IDT01*	*S0AGBF*	*S0AGBRF*	*00000000*
S1CON	*Serial 1 Control*	*98*	*1~2*	*SM01*	*SM11*	*SM21*	*REN1*	*TB81*	*RB81*	*TI1*	*RI1*	*00000000*
S1AGCR	*Serial 1 ARGB Control*	*98*	*1~2*	*SM01*	*SM11*	S1DTY1	S1GFIE	S1IDT11	S1IDT01	S1AGBF	S1AGBRF	*00000000*
S0BUF	Serial 0 Buffer	99	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
S1BUF	Serial 1 Buffer	99	1~2	.7	.6	.5	.4	.3	.2	.1	.0	00000000
S0BRT	S0 Baud-Rate Timer	9A	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
S1BRT	S1 Baud-Rate Timer	9A	1~2	.7	.6	.5	.4	.3	.2	.1	.0	00000000
S0BRC	S0 Baud-Rate Counter	9B	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
S1BRC	S1 Baud-Rate Counter	9B	1~2	.7	.6	.5	.4	.3	.2	.1	.0	00000000
S1CR1	S1 Control 1	9B	1~2	M6RER1	TXDO1	SOPWEC	TOTXD1	RMRI1	EXTFLT	TXINV1	RXINV1	00000000
S0CFG	S0 Configuration	9C	0	URTS	SMOD2	0	SM30	S0DOR	BTI	UTIE	SMOD3	00001000
S1CFG	S1 Configuration	9C	1~2	SM31	S1MOD3	S1DOR	S1TR	S1MOD1	S1TX12	S1CKOE	S1TME	00100000
S1CFG6	S1 Configuration 6	9D	1~2	PDOEC1	M6TGU1	M6TXB1	M6IDL1	M6TUR1	M6ROR1	PDOE1	M6PDE1	00010000
AC0CON	AC0 Control Reg.	9E	0	AC0LP	AC0PDX	AC0OUT	AC0F	AC0EN	AC0INV	AC0M1	AC0M0	00x00000
AC1CON	AC1 Control Reg.	9E	1	AC1LP	AC1PDX	AC1OUT	AC1F	AC1EN	AC1INV	AC1M1	AC1M0	00x00000
AC0CR1	AC0 Control Reg. 1	9E	3	GF	AC0HC1	AC0HC0		GF	AC0FLT1	AC0FLT0	AC0TR	00000000
AC0MOD	AC0 Mode Reg.	9F	0	NVRS3	NVRS2	NVRS1	NVRS0	NVRL		AC0PIS1	AC0PIS0	00000000
AC1MOD	AC1 Mode Reg.	9F	1	0	AC1HC1	AC1HC0	AC1NIS	NVRL1	AC1FLT1	AC1FLT0	AC1TR	00000000
P2	*Port 2*	*A0*	*0~F*	*0*	*P2.6*	*0*	*P2.4*	*P2.3*	*P2.2*	*0*	*0*	*01011100*
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	
AUXR0	Auxiliary Register 0	A1	*0~F*	0	0	C1BKF	PBKF	0	0	INT1H	INT0H	00000000

AUXR1	Auxiliary Register 1	A2	*0~F*	OP1Fr	OP0Fr	CRCDS1	CRCDS0	0	AC1Fr	AC0Fr	DPS	00000000
AUXR2	Auxiliary Register 2	A3	*0~F*	STAF	STOF	C1PLK	C0PLK	T1X12	T0X12	0	0	00000000
AUXR3	Auxiliary Register 3	A4	0	T0PS1	T0PS0	BPOC1	BPOC0	S0PS0	TWIPS1	TWIPS0	T0XL	00000000
AUXR4	Auxiliary Register 4	A4	1	T2PS1	T2PS0	T1PS1	T1PS0	AC1OE		AC0OE	--	00000000
AUXR5	Auxiliary Register 5	A4	2			C0PPS1	C0PPS0	EC1IPS0	C1COPS	ECIPS0	C0COPS	00000000
AUXR6	Auxiliary Register 6	A4	3	KBIHPS1	KBIHPS0	KBILPS1	KBILPS0	T3FCS	T2FCS	SnMIPS	S0COPS	00000000
AUXR7	Auxiliary Register 7	A4	4	--	--	C0CKOE	C1CKOE	0	0	0	0	00000000
AUXR8	Auxiliary Register 8	A4	5		C1ICS1	C1ICS0	0	0	S1COPS	T3PS1	T3PS0	00000000
AUXR9	Auxiliary Register 9	A4	6	SIDPS1	SIDPS0	T1G1	T0G1			S1PS1	S1PS0	00000000
AUXR10	Aux. Register 10	A4	7			SPIPS1	SPIPS0	S0PS1	P60OC1	P60OC0	P60FD	00000000
AUXR11	Aux. Register 11	A4	8			I2C1PS1	I2C1PS0	RX1S0		T1CKOE	T0CKOE	00000000
AUXR12	Aux. Register 12	A4	9	CRCDS2	CRCM0	PDOES1	PDOES0	EDCM0	GPLC0			00000000
AUXR16	Auxiliary Register 16	A4	D			C0OPS5	C0OPS4	C0OPS3	C0OPS2	C0OPS1	C0OPS0	00000000
AUXR17	Auxiliary Register 17	A4	E							C1OPS1	C1OPS0	00000000
EIE2	Extended INT Enable 2	A5	*0~F*	0	ETWI1	EPCA1	0	0	0	EAC1	ET3	00000000
EIP2L	Ext. INT Priority 2 Low	A6	*0~F*	0	PTWI1L	PPCA1L	0	0	0	PAC1L	PT3L	00000000
EIP2H	Ext. INT Priority 2 High	A7	*0~F*	0	PTWI1H	PPCA1H	0	0	0	PAC1H	PT3H	00000000
IE	*Interrupt Enable*	*A8*	*0~F*	*EA*	*EDMA*	*ET2*	*ES0*	*ET1*	*EX1*	*ET0*	*EX0*	*00000000*
SADDR	Slave Address	A9	*0~F*	.7	.6	.5	.4	.3	.2	.1	.0	00000000
EDC45	4b5b ENDEC	A9	*0~F*	.7	.6	.5	.4	.3	.2	.1	.0	00000000
SFRPI	SFR Page Index	AC	*0~F*	0	0	0	0	IDX3	IDX2	IDX1	IDX0	00000000
EIE1	Extended INT Enable 1	AD	*0~F*	EAC0	ETWI0	EKB	ES1	ESF	EPTM0	EADC	ESPI	00000000
EIP1L	Ext. INT Priority 1 Low	AE	*0~F*	PAC0L	PTWI0L	PKBL	PS1L	PSFL	PPCAL	PADCL	PSPIL	00000000
EIP1H	Ext. INT Priority 1 High	AF	*0~F*	PAC0H	PTWI0H	PKBH	PS1H	PSFH	PPCAH	PADCH	PSPIH	00000000
P3	*Port 3*	*B0*	*0~F*	*0*	*0*	*P3.5*	*P3.4*	*P3.3*	*P3.2*	*P3.1*	*P3.0*	*00111111*
P3M0	P3 Mode Register 0	B1	*0~F*	0	0	P3M0.5	P3M0.4	P3M0.3	P3M0.2	P3M0.1	P3M0.0	00000000
P3M1	P3 Mode Register 1	B2	*0~F*	0	0	P3M1.5	P3M1.4	P3M1.3	P3M1.2	P3M1.1	P3M1.0	00000000
P4M0	P4 Mode Register 0	B3	0	P4M0.7	0	P4M0.5	P4M0.4	0	0	P4M0.1	P4M0.0	10110000
PDRVC0	Port Driving Control 0	B4	2	P3DC1	P3DC0	P2DC1	P2DC0	P1DC1	P1DC0	P0DC1	P0DC0	00000000
PDRVC1	Port Driving Control 1	B4	3	0	0	0	0	0	0	P4DC1	P4DC0	00000000
P6M0	P6 Mode Register 0	B5	1							P6M0.1	P6M0.0	00000000
IP0H	Interrupt Priority 0 High	B7	*0~F*	PX3H	PX2H	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	00000000
IP0L	*Interrupt Priority Low*	*B8*	*0~F*	*PX3L*	*PX2L*	*PT2L*	*PSL*	*PT1L*	*PX1L*	*PT0L*	*PX0L*	*00000000*
SADEN	Slave Address Mask	B9	*0~F*	.7	.6	.5	.4	.3	.2	.1	.0	00000000
S0CR1	S0 Control 1	B9	*0~F*	S0TR	S0TX12	S0TCK	S0RCK	S0CKOE	ARTE	0	0	00000000
OPSTA	OPA Status	BA	0							OP1CMPF	OP0CMPF	00000000
OP0CON0	OPA0 Control 0	BA	1	OP0EN	OPPWR	OP0LP	OP0IG	OP0PIS1	OP0PIS0	OP0NIS1	OP0NIS0	00000000
OP1CON0	OPA1 Control 0	BA	2	OP1EN		OP1LP	OP1IG	OP1PIS1	OP1PIS0	OP1NIS1	OP1NIS0	00000000
PGACON0	PGA Control 0	BA	5	PGEN		PGLP	PGAOE	PGPIS1	PGPIS0			00000000
PGACON1	PGA Control 1	BA	6			PGENS1GN	0	PGGNM	PGGN2	PGGN1	PGGN0	00000000
OPCMP	OPA Comparison Mode	BA	7			OP1PDX	OP0PDX	OP1M1	OP1M0	OP0M1	OP0M0	00000000
PWMCR	PWM Control Reg.	BC	0	PCAE	C0OFS	PBKM	PBKE1.1	PBKE1.0	PBKE0.2	PBKE0.1	PBKE0.0	00000000
PDTCRA	PWM Dead-Time Control Reg. - A	BC	1	DTPS1	DTPS0	DT.5	DT.4	DT.3	DT.2	DT.1	DT.0	00000000
C1PWMCR	PWM Control Reg.	BC	A	PCA1E	C1OFS	C1BKM	C1BKE1.1	C1BKE1.0	C1BKE0.2	C1BKE0.1	C1BKE0.0	00000000
C1PDTCR	PWM Dead-Time Control Reg.	BC	B	C1DTPS1	C1DTPS0	C1DT.5	C1DT.4	C1DT.3	C1DT.2	C1DT.1	C1DT.0	00000000
CRC0DA	CRC0 Data Port	BD	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CKCON1	Clock Control 1	BF	0~F	XTOR	PLLOCKF	MCKSTA1	MCKSTA0	OSCSTA3	OSCSTA2	OSCSTA1	OSCSTA0	00010001
XICON	*External INT Control*	*C0*	0~F	*INT3H*	*EX3*	*IE3*	*IT3*	*INT2H*	*EX2*	*IE2*	*IT2*	*00000000*
XICFG	Ext. INT. Configured	C1	0	INT1IS1	INT1IS0	INT0IS1	INT0IS0	X3FLT	X2FLT	X1FLT	X0FLT	00000000
XICFG1	Ext. INT. Configured 1	C1	1	INT1IS2	INT0IS2	INT2IS1	INT2IS0	X3FLT1	X2FLT1	X1FLT1	X0FLT1	00000000
XICFG2	Ext. INT. Configured 2	C1	2	0	0	0	0	INT3IS2	INT3IS1	INT3IS0	INT2IS2	00000000
ADCFG0	ADC Configuration 0	C3	0	ADCKS2	ADCKS1	ADCKS0	ADRJ	ACHS	SMPF	ADTM1	ADTM0	00000000
ADCFG1	ADC Configuration 1	C3	1	IGADCI	EADCWI	SMPFIE	SIGN	AOS.3	AOS.2	AOS.1	AOS.0	00000000
ADCFG2	ADC Configuration 2	C3	2	SHT.7	SHT.6	SHT.5	SHT.4	SHT.3	SHT.2	SHT.1	SHT.0	00000000
ADCFG3	ADC Configuration 3	C3	3	ADPS1	ADPS0	ADHS	ADHS1	ARES1	ARES0	ADES0	0	01100000
ADCFG4	ADC Configuration 4	C3	4	ADMINS	ADWM0	ADTM3	ADTM2	0	VRS0	DBSD	DOSD	00000000
ADCFG5	ADC Configuration 5	C3	5	ASCE.7	ASCE.6	ASCE.5	ASCE.4	ASCE.3	ASCE.2	ASCE.1	ASCE.0	00000000
ADCFG7	ADC Configuration 7	C3	7	0	0	0	0	ASCS3	ASCS2	ASCS1	ASCS0	00000000
ADCFG11	ADC Configuration 11	C3	B	WHB.3	WHB.2	WHB.1	WHB.0	1	1	1	1	11111111
ADCFG12	ADC Configuration 12	C3	C	WHB.11	WHB.10	WHB.9	WHB.8	WHB.7	WHB.6	WHB.5	WHB.4	11111111
ADCFG13	ADC Configuration 13	C3	D	WLB.3	WLB.2	WLB.1	WLB.0	0	0	0	0	00000000
ADCFG14	ADC Configuration 14	C3	E	WLB.11	WLB.10	WLB.9	WLB.8	WLB.7	WLB.6	WLB.5	WLB.4	00000000
ADCON0	ADC Control 0	C4	0~F	ADCEN	ADCWI	CHS3	ADCI	ADCS	CHS2	CHS1	CHS0	00000000
ADCDL	ADC Data Low	C5	0~F	ADCV.3	ADCV.2	ADCV.1	ADCV.0	0	0	0	0	00000000
ADCDH	ADC Data High	C6	0~F	ADCV.11	ADCV.10	ADCV.9	ADCV.8	ADCV.7	ADCV.6	ADCV.5	ADCV.4	00000000
CKCON0	Clock Control 0	C7	0~F	AFS	ENCKM	CKMIS1	CKMIS0	CCKS	SCKS2	SCKS1	SCKS0	00010000
T2CON	*Timer 2 Control Reg.*	*C8*	*0*	*TF2*	*EXF2*	*RCLK/ TF2L*	*TCLK/ TL2IE*	*EXEN2*	*TR2*	*C/T2*	*CP/RL2*	*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**	
T3CON	*Timer 3 Control Reg.*	*C8*	*1*	*TF3*	*EXF3*	*TF3L*	*TL3IE*	*EXEN3*	*TR3*	*C/T3*	*CP/RL3*	*00000000*
T5CON	*Timer 5 Control Reg.*	*C8*	*3*	*TF5*	*0*	*T5CKS1*	*T5CKS0*	*T5IE*	*TR5*	*T5GAT1*	*T5GAT0*	*00000000*
T6CON	*Timer 6 Control Reg.*	*C8*	*4*	*TF6*	*0*	*T6CKS1*	*T6CKS0*	*T6IE*	*TR6*	*T6GAT1*	*T6GAT0*	*00000000*
T2MOD	Timer 2 mode Reg.	C9	0	T2SPL	TL2X12/	T2EXH	T2X12	TR2L	TR2LC	T2OE	T2MS0	00000000

					T2EIP							
T3MOD	Timer 3 mode Reg.	C9	1	T3SPL	TL3X12/ T3EIP	T3EXH	T3X12	TR3L	TR3LC	T3OE	T3MS0	00000000
RCAP2L	Timer2 Capture Low	CA	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
RCAP3L	Timer3 Capture Low	CA	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TLR5	TL5 reload Reg.	CA	3	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TLR6	TL6 reload Reg.	CA	4	.7	.6	.5	.4	.3	.2	.1	.0	00000000
RCAP2H	Timer2 Capture High	CB	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
RCAP3H	Timer3 Capture High	CB	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
THR5	TH5 reload Reg.	CB	3	.7	.6	.5	.4	.3	.2	.1	.0	00000000
THR6	TH6 reload Reg.	CB	4	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TL2	Timer Low 2	CC	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TL3	Timer Low 3	CC	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TL5	Timer Low 5	CC	3	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TL6	Timer Low 6	CC	4	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TH2	Timer High 2	CD	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TH3	Timer High 3	CD	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TH5	Timer High 5	CD	3	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TH6	Timer High 6	CD	4	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CLRL	CL Reload register	CE	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CL1RL	CL1 Reload register	CE	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CHRL	CH Reload register	CF	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CH1RL	CH1 Reload register	CF	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
PSW	*Program Status Word*	*D0*	0~F	*CY*	*AC*	*F0*	*RS1*	*RS0*	*OV*	*F1*	*P*	*00000000*
SIADR	TWI0 Address Reg.	D1	0	.7	.6	.5	.4	.3	.2	.1	GC	00000000
SI1ADR	TWI1 Address Reg.	D1	1	.7	.6	.5	.4	.3	.2	.1	GC1	00000000
SIA2	TWI0 2nd Addr Reg.	D1	2	.7	.6	.5	.4	.3	.2	.1	A2E	00000000
SI1A2	TWI1 2nd Addr Reg.	D1	3	.7	.6	.5	.4	.3	.2	.1	A2E1	00000000
SIDAT	TWI0 Data Reg.	D2	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
SI1DAT	TWI1 Data Reg.	D2	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
SIA2M	SIA2 Mask Reg.	D2	2	SIA2M.7	SIA2M.6	SIA2M.5	SIA2M.4	SIA2M.3	SIA2M.2	SIA2M.1	1	11111111
SI1A2M	SI1A2 Mask Reg.	D2	3	SI1A2M.7	SI1A2M.6	SI1A2M.5	SI1A2M.4	SI1A2M.3	SI1A2M.2	SI1A2M.1	1	11111111
SISTA	TWI0 Status Reg.	D3	0	.7	.6	.5	.4	.3	.2	.1	.0	11111000
SI1STA	TWI1 Status Reg.	D3	1	.7	.6	.5	.4	.3	.2	.1	.0	11111000
SICON	TWI0 Control Reg.	D4	0	CR2	ENSI	STA	STO	SI	AA	CR1	CR0	00000000
SI1CON	TWI1 Control Reg.	D4	1	CR21	ENSI1	STA1	STO1	SI1	AA1	CR11	CR01	00000000
SICR1	TWI0 Control Reg. 1	D4	2					F8IE	MTSE	PAA	CR3	00000000
SI1CR1	TWI1 Control Reg. 1	D4	3					F8IE1	MTSE1	PAA1	CR31	00000000
KBPATN	Keypad Pattern	D5	0~F	.7	.6	.5	.4	.3	.2	.1	.0	11111111
KBCON	Keypad Control	D6	0~F	KBCS1	KBCS0	KBES	0	0	0	PATN_SEL	KBIF	00000000
KBMASK	Keypad Int. Mask	D7	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCON	*PTM Control Reg.*	*D8*	0	*CF*	*CR*	*CCF5*	*CCF4*	*CCF3*	*CCF2*	*CCF1*	*CCF0*	*00000000*
C1CON	*PCA1 Control Reg.*	*D8*	1	*C1F*	*C1R*					*C1CF1*	*C1CF0*	*00000000*
CMOD	PTM Mode Reg.	D9	0	CIDL	BME4	BME2	BME0	CPS2	CPS1	CPS0	ECF	00000000
C1MOD	PCA1 Mode Reg.	D9	1	C1IDL			BME01	C1PS2	C1PS1	C1PS0	ECF1	00000000
CCAPM0	PTM Module0 Mode	DA	0	DTE0	ECOM0	0	0	MAT0	TOG0	PWM0	ECCF0	00000000
C1CAPM0	PCA1 Module0 Mode	DA	1	DT1E0	ECOM01	CAP1P0	CAP1N0	MAT01	TOG01	PWM01	EC1CF0	00000000
CCAPM1	PTM Module1 Mode	DB	0	0	ECOM1	0	0	MAT1	TOG1	PWM1	ECCF1	00000000
C1CAPM1	PCA1 Module0 Mode	DB	1	0	ECOM11	CAP1P1	CAP1N1	MAT11	TOG11	PWM11	EC1CF1	00000000
CCAPM2	PTM Module2 Mode	DC	0~F	DTE2	ECOM2	0	0	MAT2	TOG2	PWM2	ECCF2	00000000
CCAPM3	PTM Module3 Mode	DD	0~F	0	ECOM3	0	0	MAT3	TOG3	PWM3	ECCF3	00000000
CCAPM4	PTM Module4 Mode	DE	0~F	DTE4	ECOM4	0	0	MAT4	TOG4	PWM4	ECCF4	00000000
CCAPM5	PTM Module5 Mode	DF	0~F	0	ECOM5	0	0	MAT5	TOG5	PWM5	ECCF5	00000000
ACC	*Accumulator*	*E0*	0~F	*ACC.7*	*ACC.6*	*ACC.5*	*ACC.4*	*ACC.3*	*ACC.2*	*ACC.1*	*ACC.0*	*00000000*
WDTCR	WDT Control register	E1	0~F	WREN	NSW	ENW	CLRW	WIDL	PS2	PS1	PS0	00000000
IFD	ISP Flash data	E2	0~F	.7	.6	.5	.4	.3	.2	.1	.0	11111111
IFADRH	ISP Flash Addr. High	E3	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
IFADRL	ISP Flash Addr. Low	E4	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
IFMT	ISP Mode Table	E5	0~F	MS.7	0	0	0	MS.3	MS.2	MS.1	MS.0	00000000
SCMD	ISP Serial Command	E6	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
ISPCR	ISP Control Register	E7	0~F	ISPEN	SWBS	SRST	CFAIL	0	0	0	0	00000000
P4	*Port 4*	*E8*	0~F	*P4.7*		*P4.5*	*P4.4*			*P4.1*	*P4.0*	*10110011*
CL	PTM base timer Low	E9	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CL1	PCA1 base timer Low	E9	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP0L	PTM module0 compare Low	EA	0	.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**	
CCAP1L	PTM module1 compare Low	EB	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
C1CAP1L	PCA1 module1 capture Low	EB	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP2L	PTM module2 compare Low	EC	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000

CCAP3L	PTM module3 compare Low	ED	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP4L	PTM module4 compare Low	EE	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP5L	PTM module5 compare Low	EF	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
B	*B Register*	*F0*	0~F	*B.7*	*B.6*	*B.5*	*B.4*	*B.3*	*B.2*	*B.1*	*B.0*	*00000000*
PAOE	PWM Additional Output Enable	F1	0~7	POE3	POE2B	POE2A	POE2	POE1	POE0B	POE0A	POE0	10011001
C0AOE1	PTM0 Additional Output Enable 1	F1	8	POEM0				POE5			POE4	0xxx1xx1
C1AOE0	PCA1 Additional Output Enable 0	F1	9					C1POE1			C1POE0	xxxx1001
C1AOE1	PCA1 Additional Output Enable 1	F1	A	C1POEM0								0xxxxxxx
PCAPWM0	PTM PWM0 Mode	F2	0	P0RS1	P0RS0	0	0	0	P0INV	ECAP0H	ECAP0L	00000000
C1PWM0	PCA1 PWM0 Mode	F2	1	P0RS11	P0RS01	0	0	0	P0INV1	ECAP0H1	ECAP0L1	00000000
PCAPWM1	PTM PWM1 Mode	F3	0	P1RS1	P1RS0	0	0	0	P1INV	ECAP1H	ECAP1L	00000000
C1PWM1	PCA1 PWM1 Mode	F3	1	P1RS11	P1RS01	0	0	0	P1INV1	ECAP1H1	ECAP1L1	00000000
PCAPWM2	PTM PWM2 Mode	F4	0~F	P2RS1	P2RS0	0	0	0	P2INV	ECAP2H	ECAP2L	00000000
PCAPWM3	PTM PWM3 Mode	F5	0~F	P3RS1	P3RS0	0	0	0	P3INV	ECAP3H	ECAP3L	00000000
PCAPWM4	PTM PWM4 Mode	F6	0~F	P4RS1	P4RS0	0	0	0	P4INV	ECAP4H	ECAP4L	00000000
PCAPWM5	PTM PWM5 Mode	F7	0~F	P5RS1	P5RS0	0	0	0	P5INV	ECAP5H	ECAP5L	00000000
P6	*Port 6*	*F8*	*1*	*0*	*0*	*0*	*0*	*0*	*0*	*P6.1*	*P6.0*	*00000011*
CH	PTM base timer High	F9	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CH1	PCA1 base timer High	F9	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP0H	PTM Module0 compare High	FA	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
C1CAP0H	PCA1 Module0 capture High	FA	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP1H	PTM Module1 compare High	FB	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
C1CAP1H	PCA1 Module1 capture High	FB	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP2H	PTM Module2 compare High	FC	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP3H	PTM Module3 compare High	FD	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP4H	PTM Module4 compare High	FE	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP5H	PTM Module5 compare High	FF	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000

3.3. Auxiliary SFR Map (Page P)

MG82F6P32 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.** The 11 physical bytes include IAPLB, CKCON2, CKCON3, CKCON4, CKCON5, PCON2, PCON3, SPCON0, DCON0, RTCTM and RTCCR. The 6 logical bytes include PCON0, PCON1, CKCON0, WDTCR, P4 and P6. Access on the 6 logical bytes gets the coherence content with the same SFR in Page 0~F. Please refer Section "33 Page P SFR Access" for more detail information.

Table 3–3. Auxiliary SFR Map (Page P)

		0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
F8		P6	--	--	--	--	--	--	--
F0		--	--	--	--	--	--	--	--
E8		P4	--	--	--	--	--	--	--
E0		--	WDTCR	--	--	--	--	--	--
D8		--	--	--	--	--	--	--	--
D0		--	--	--	--	--	--	--	--
C8		--	--	--	--	--	--	--	--
C0		--	--	--	--	--	--	--	CKCON0
B8		--	--	--	--	--	--	--	--
B0		--	--	--	--	--	--	--	--
A8		--	--	--	--	--	--	--	--
A0		--	--	--	--	--	--	--	--
98		--	--	--	--	--	--	--	--
90		--	--	--	--	--	--	--	PCON1
88		--	--	--	--	--	--	--	--
80		--	--	--	--	--	--	--	PCON0
78		--	--	--	--	--	--	--	--
70		--	--	--	--	--	--	--	--
68		--	--	--	--	--	--	--	--
60		--	--	--	--	--	--	--	--
58		--	--	--	--	--	--	--	--
50		--	--	--	SPHB	RTCCR	RTCTM	--	--
48		SPCON0	--	--	--	DCON0	--	--	--
40		CKCON2	CKCON3	CKCON4	--	PCON2	PCON3	--	--
38		--	--	--	--	--	--	--	--
30		--	--	--	--	--	--	--	--
28		--	--	--	--	--	--	--	--
20		--	--	--	--	--	--	--	--
18		--	--	--	--	--	--	--	--
10		--	--	--	OP0CFG0	OP1CFG0	PGACFG0	AC0CF0	AC1CF0
08		--	--	--	--	--	--	--	--
00		--	--	--	IAPLB	--	--	--	--
		0/8	**1/9**	**2/A**	**3/B**	**4/C**	**5/D**	**6/E**	**7/F**

3.4. Auxiliary SFR Bit Assignment (Page P)

Table 3–4. 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	IAP Low Boundary	03H	IAPLB6	IAPLB5	IAPLB4	IAPLB3	IAPLB2	IAPLB1	IAPLB0	0	
OP0CFG0	OP0 Configuration 0	13H	--	--	*OP0OS5*	*OP0OS4*	*OP0OS3*	*OP0OS2*	*OP0OS1*	*OP0OS0*	*00xxxxxx*
OP1CFG0	OP1 Configuration 0	14H	--	--	*OP1OS5*	*OP1OS4*	*OP1OS3*	*OP1OS2*	*OP1OS1*	*OP1OS0*	*00011111*
PGACFG0	PGA Configuration 0	15H	--	--	*PGAOS5*	*PGAOS4*	*PGAOS3*	*PGAOS2*	*PGAOS1*	*PGAOS0*	*00011111*
AC0CF0	AC0 Configuration 0	16H	*AC0OSE*	*AC0OS6*	*AC0OS5*	*AC0OS4*	*AC0OS3*	*AC0OS2*	*AC0OS1*	*AC0OS0*	*10000000*
AC1CF0	AC1 Configuration 0	17H	*AC1OSE*	*AC1OS6*	*AC1OS5*	*AC1OS4*	*AC1OS3*	*AC1OS2*	*AC1OS1*	*AC1OS0*	*10000000*
CKCON2	Clock Control 2	40H	XTGS1	XTGS0	XTALE	IHRCOE	MCKS1	MCKS0	OSCS1	OSCS0	01010000
CKCON3	Clock Control 3	41H	WDTCS1	WDTCS0	FWKP	WDTFS	MCKD1	MCKD0	MCDS1	MCDS0	00000010
CKCON4	Clock Control 4	42H	RCSS2	RCSS1	RCSS0	RPSC2	RPSC1	RPSC0	RTCCS3	RTCCS2	00000000
PCON2	Power Control 2	44H	AWBOD1	0	BO1S1	BO1S0	BO1RE	EBOD1	BO0RE	1	0000x1x1
PCON3	Power Control 3	45H	IVREN	IVRPDE	0	SPWRE	0	0	0	0	00000000
SPCON0	SFR Page Control 0	48H	0	P6CTL	P4CTL	WRCTL	0	CKCTL0	PWCTL1	PWCTL0	00000000
DCON0	Device Control 0	4CH	HSE	IAPO	HSE1	0	0	IORCTL	RSTIO	OCDE	10000011
SPHB	SP High Boundary	53H	1	1	1	1	SPHB.3	SPHB.2	SPHB.1	SPHB.0	11111111
RTCCR	RTC Control Reg.	54H	RTCE	RTCO	RTCRL5	RTCRL4	RTCRL3	RTCRL2	RTCRL1	RTCRL0	00111111
RTCTM	RTC Timer Register	55H	RTCCS1	RTCCS0	RTCCT5	RTCCT4	RTCCT3	RTCCT2	RTCCT1	RTCCT0	01111111
Logical Bytes											
PCON0	Power Control 0	87H	SMOD1	SMOD0	GF	POF0	GF1	GF0	PD	IDL	00010000
PCON1	Power Control 1	97H	SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF	00000000
CKCON0	Clock Control 0	C7H	AFS	ENCKM	CKMIS1	CKMIS0	CCKS	SCKS2	SCKS1	SCKS0	00010000
WDTCR	Watch-dog-timer Control register	E1H	WREN	NSW	ENW	CLRW	WIDL	PS2	PS1	PS0	00000000
P4	Port 4	E8H	P4.7	1	P4.5	P4.4	1	1	P4.1	P4.0	11111111
P6	Port 6	F8H	1	1	1	1	1	1	P6.1	P6.0	11111111

Sample Code of Page-P SFR write：

```
IFADRH = 0x00;
ISPCR = ISPEN;                  //enable IAP/ISP
IFMT = MS2;                     // Page-P write，IFMT =0x04
IFADRH = 0x00;
IFADRL = SPCON0;                //Set Page-P SFR address
IFD |= CKCTL0;                  // set CKCTL0
SCMD = 0x46;                    //
SCMD = 0xB9;                    //
IFMT = Flash_Standby;           // IAP/ISP standby，IFMT =0x00
ISPCR &= ~ISPEN;
```

4. Pin Configurations

4.1. Package Instruction

Figure 4–1. **MG82F6P32** LQFP32 Top View

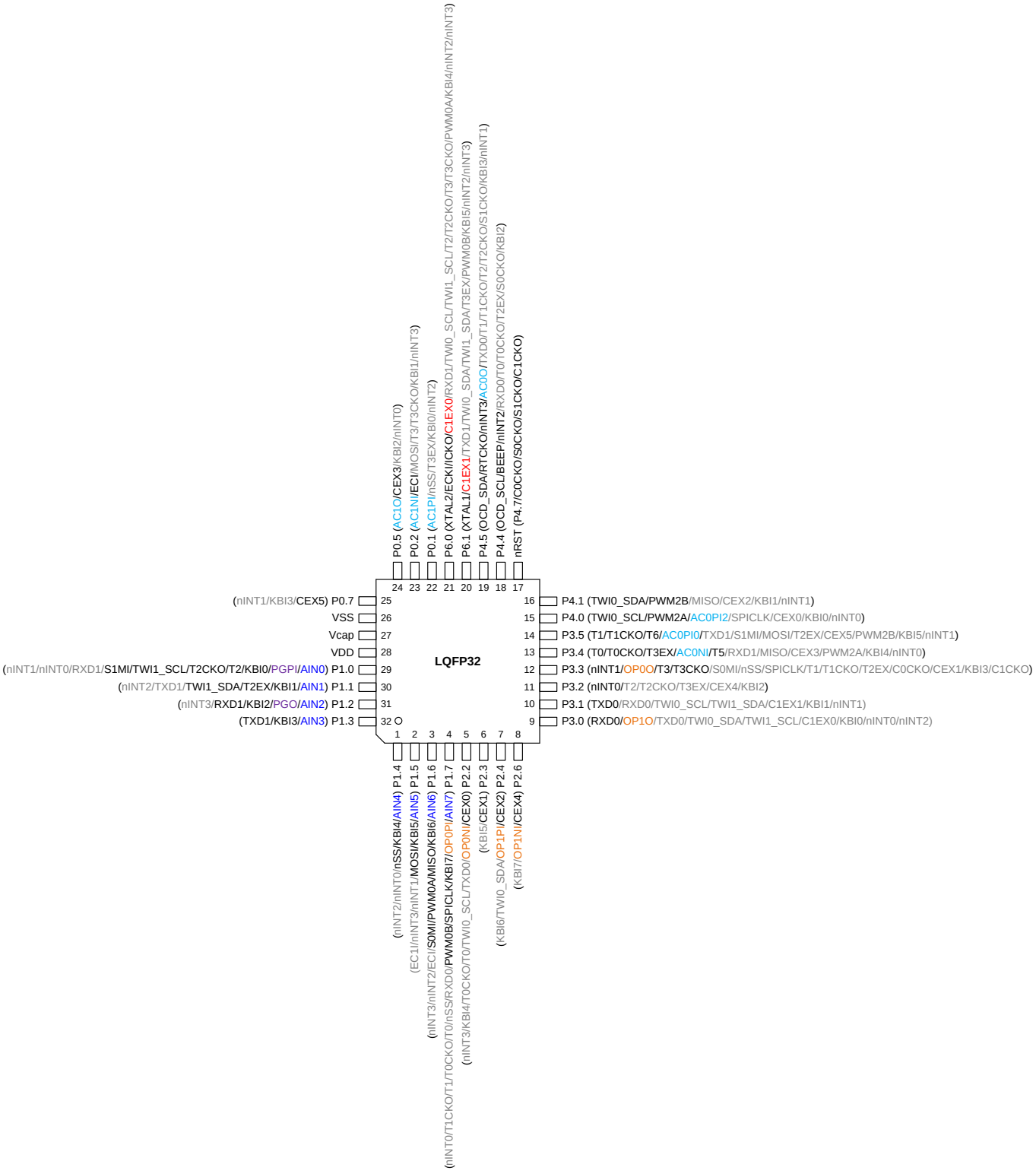


Figure 4–2. **MG82F6P32** QFN32 Top View

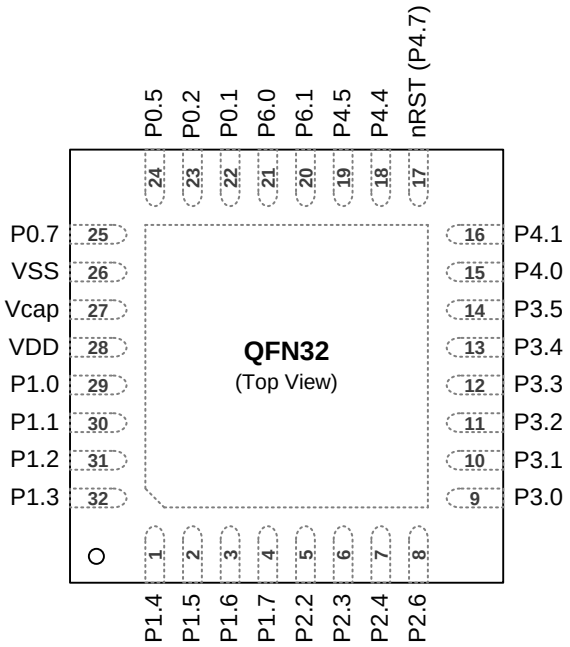
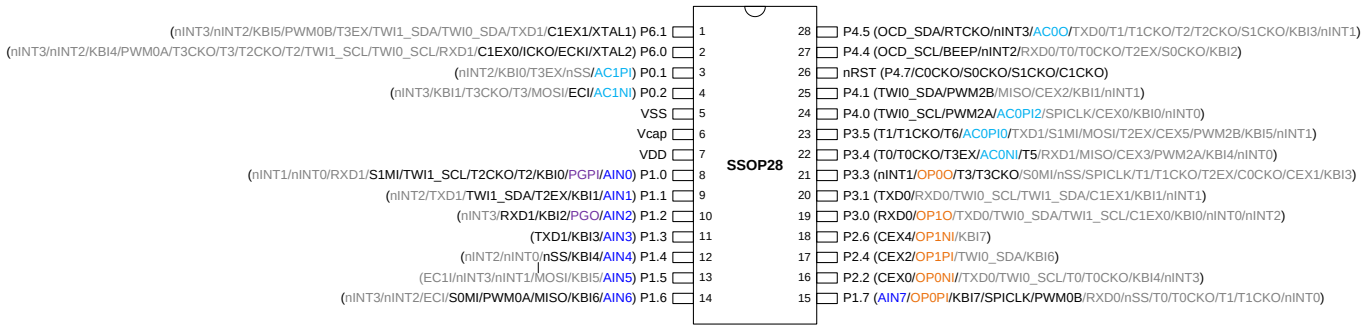


Figure 4–3. **MG82F6P32** SSOP28 Top View



4.2. Pin Description

Table 4–1. Pin Description

MNEMONIC	PIN NUMBER		I/O TYPE	DESCRIPTION
	32-Pin LQFP/ QFN	28-Pin SSOP		
P0.1 (AC1PI)	22	3	I/O	* Port 0.1. * AC1PI: Analog Comparator 1 negative input.
P0.2 (AC1NI) (ECI)	23	4	I/O	* Port 0.2. * AC1NI: Analog Comparator 1 negative input. * ECI: PTM external clock input.
P0.5 (AC1O) (CEX3)	24	--	I/O	* Port 0.5. * AC1OUT: Analog Comparator 1 output. * CEX3: PTM module-3 external I/O.
P0.7 (CEX5)	25	--	I/O	* Port 0.7. * CEX5: PTM module-5 external I/O.
P1.0 (AIN0) (PGPI) (KBI0) (T2) (T2CKO) (TWI1_SCL) (S1MI)	29	8	I/O	* Port 1.0. * AIN0: ADC channel-0 analog input. * PGPI: PGA positive input * KBI0: keypad input 0. * T2: Timer/Counter 2 external clock input. * T2CKO: Timer 2 programmable clock output. * TWI1_SCL: serial clock of TWI1/I2C1. * S1MI: Serial Port 1 SPI Master mode data Input.
P1.1 (AIN1) (KBI1) (T2EX) (TWI1_SDA)	30	9	I/O	* Port 1.1. * AIN1: ADC channel-1 analog input. * KBI1: keypad input 1. * T2EX: Timer/Counter 2 external control input. * TWI1_SDA: serial data of TWI1/I2C1.
P1.2 (AIN2) (PGO) (KBI2) (RXD1)	31	10	I/O	* Port 1.2. * AIN0: ADC channel-2 analog input. * PGO: PGA output * KBI2: keypad input 2. * RXD1: Serial Port 1 (UART1) serial input port.
P1.3 (AIN3) (KBI3) (TXD1)	32	11	I/O	* Port 1.3. * AIN3: ADC channel-3 analog input. * KBI3: keypad input 3. * TXD1: Serial Port 1 (UART1) serial output port.
P1.4 (AIN4) (KBI4) (nSS)	1	12	I/O	* Port 1.4. * AIN4: ADC channel-4 analog input. * KBI4: keypad input 4. * nSS: SPI slave select.
P1.5 (AIN5) (KBI5) (MOSI)	2	13	I/O	* Port 1.5. * AIN5: ADC channel-5 analog input. * KBI5: keypad input 5. * MOSI: SPI master out & slave in
P1.6 (AIN6) (KBI6) (MISO) (PWM0A) (S0MI)	3	14	I/O	* Port 1.6. * AIN6: ADC channel-6 analog input. * KBI6: keypad input 6. * MISO: SPI master in & slave ou. * PWM0A: PTM PWM0 output sub-channel A. * S0MI: Serial Port 0 SPI Master mode data Input.
P1.7 (AIN7) (OP0PI) (KBI7) (SPICLK) (PWM0B)	4	15	I/O	* Port 1.7. * AIN7: ADC channel-7 analog input. * OP0PI: OPA0 positive input * KBI7: keypad input 7. * SPICLK: SPI clock, output for master and input for slave. * PWM0B: PTM PWM0 output sub-channel B.
P2.2 (CEX0) (OP0NI)	5	16	I/O	* Port 2.2. * CEX0: PTM module-0 external I/O. * OP0NI: OPA0 negative input
P2.3 (CEX1)	6	--	I/O	* Port 2.3. * CEX1: PTM module-1 external I/O.
P2.4 (CEX2) (OP1PI)	7	17	I/O	* Port 2.4. * CEX2: PTM module-2 external I/O. * OP1PI: OPA1 positive input
P2.6 (CEX4)	8	18	I/O	* Port 2.6. * CEX4: PTM0 module-4 external I/O.

(OP1NI)				* OP1NI: OPA1 negative input
P3.0 (RXD0) (OP1O)	9	19	I/O	* Port 3.0. * RXD0: Serial Port 0 (UART0) serial input port. * OP1O: OPA1 output
P3.1 (TXD0)	10	20	I/O	* Port 3.1. * TXD0: Serial Port 0 (UART0) serial output port.
P3.2 (nINT0)	11	21	I/O	* Port 3.2. * nINT0: external interrupt 0 input.
P3.3 (nINT1) (T3) (T3CKO) (OP0O)	12	21	I/O	* Port 3.3. * nINT1: external interrupt 1 input. * T3: Timer/Counter 3 external clock input. * T3CKO: Timer 3 programmable clock output. * OP0O: OPA0 output
P3.4 (T0) (T0CKO) (T3EX) (AC0NI) (T5)	13	22	I/O	* Port 3.4. * T0: Timer/Counter 0 external input. * T0CKO: Timer 0 programmable clock output. * T3EX: Timer/Counter 3 external control input. * AC0NI: Analog Comparator 0 negative input. * T5: Timer/Counter 5 external clock input.
P3.5 (T1) (T1CKO) (T6) (AC0PI0)	14	23	I/O	* Port 3.5. * T1: Timer/Counter 1 external input. * T1CKO: Timer 1 programmable clock output. * T6: Timer/Counter 6 external clock input. * AC0PI0: Analog Comparator 0 positive input channel 0.
P4.0 (TWI0_SCL) (PWM2A) (AC0PI2)	15	24	I/O	* Port 4.0. * TWI0_SCL: serial clock of TWI0/I2C0. * PWM2A: PTM0 PWM2 output sub-channel A. * AC0PI2: Analog Comparator 0 positive input channel 2.
P4.1 (TWI0_SDA) (PWM2B)	16	25	I/O	* Port 4.1. * TWI0_SDA: serial data of TWI0/I2C0. * PWM2B: PTM0 PWM2 output sub-channel B.
P4.4 (OCD_SCL) (BEEP) (nINT2)	18	27	I/O	* Port 4.4. * OCD_SCL: OCD interface, serial clock. * BEEP: Beeper output. * nINT2: external interrupt 2 input.
P4.5 (OCD_SDA) (RTCKO) (nINT3) (AC0O)	19	28	I/O	* Port 4.5. * OCD_SDA: OCD interface, serial data. * RTCKO: RTC programmable clock output. * nINT3: external interrupt 3 input. * AC0OUT: Analog Comparator 0 output.
P6.0 (XTAL2) (ECKI) (ICKO) (C1EX1)	21	2	I/O I O	* Port 6.0. * XTAL2: Output of on-chip crystal oscillating circuit. * ECKI: In external clock input mode, this is clock input pin. * ICKO: Internal Clock (MCK) Output. * C1EX1: PCA1 module-1 external I/O.
P6.1 (XTAL1) (C1EX0)	20	1	I/O	* Port 6.1. * XTAL1: Input of on-chip crystal oscillating circuit. * C1EX0: PCA1 module-0 external I/O.
nRST (P4.7) (C0CKO) (S0CKO) (S1CKO) (C1CKO)	17	26	I **O**	* nRST: External RESET input, **low** active. * Port 4.7 output only. *Note: When P4.7/nRST use as port pin, it is not suggested to program it as Input to avoid MCU is locked in reset in bootup period when level high send into this pin.* * C0CKO: Programmable clock output of PTM0 base counter. * S0CKO: S0BRG programmable clock output. * S1CKO: S1BRG programmable clock output. * C1CKO: Programmable clock output of PCA1 base counter.
Vcap	27	6	I/O	* Vcap. Internal LDO should add output capacitor for voltage stabilize. Connect 0.1uF and 4.7uF to VSS.
VDD	28	7	P	Power supply input.
VSS	26	5	G	Ground, 0 V reference.

4.3. Alternate 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.

AUXR3: Auxiliary Register 3

SFR Page = **0 only**

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
T0PS1	T0PS0	BPOC1	BPOC0	S0PS0	TWIPS1	TWIPS0	T0XL
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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

T0PS1~0	T0/T0CKO
0 0	P3.4
0 1	P4.4
1 0	P2.2
1 1	P1.7

Bit 5~4: BPOC1~0, Beeper output control bits.

BPOC[1:0]	P4.4 function	I/O mode
0 0	P4.4	By P4M0.4 & P4M1.4
0 1	ILRCO/32	By P4M0.4 & P4M1.4
1 0	ILRCO/16	By P4M0.4 & P4M1.4
1 1	ILRCO/8	By P4M0.4 & P4M1.4

For beeper on P4.4 function, it is recommended to configure P4.4 as push-push output mode.

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

S0PS1~0	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/I2C0 Port pin Selection [1:0].

TWIPS1~0	TWI0_SCL	TWI0_SDA
0 0	P4.0	P4.1
0 1	P6.0	P6.1
1 0	P3.1	P3.0
1 1	P2.2	P2.4

AUXR4: Auxiliary Register 4

SFR Page = **1 only**

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
T2PS1	T2PS0	T1PS1	T1PS0	AC1OE	0	AC0OE	0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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

T2PS1~0	T2/T2CKO	T2EX
0 0	P1.0	P1.1
0 1	P3.2	P3.3
1 0	P6.0	P3.5
1 1	P4.5	P4.4

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

T1PS1~0	T1/T1CKO
0 0	P3.5
0 1	P4.5
1 0	P1.7
1 1	P3.3

Bit 3: AC1OE, AC1OUT output enable on port pin.
0: Disable AC1OUT output on port pin.
1: Enable AC1OUT output on P0.5.

Bit 1: AC0OE, AC0OUT output enable on port pin.
0: Disable AC0OUT output on port pin.
1: Enable AC0OUT output on P4.5.

AUXR5: Auxiliary Register 5

SFR Page = **2 only**
SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	C0PPS1	C0PPS0	EC1IPS0	C1COPS	ECIPS0	C0COPS
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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

C0PPS1	PWM2A	PWM2B
0	P4.0	P4.1
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: EC1IPS0, PCA1 ECI Port pin Selection0.

EC1IPS0	EC1I
0	ECI
1	P1.5

Bit 2: C1COPS, PCA1 Clock Output (C1CKO) port pin Selection.

C1COPS	C1CKO
0	P4.7
1	P3.3

Bit 1: ECIPS0, PTM0 ECI Port pin Selection0.

ECIPS0	ECI
0	P0.2
1	P1.6

Bit 0: C0COPS, PTM0 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 RESET = 0000-0000

7	6	5	4	3	2	1	0
KBIHPS1	KBIHPS0	KBILPS1	KBILPS0	T3FCS	T2FCS	SnMIPS	S0COPS
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~6: KBIHPS1~0, KBI4~7 Port pin Selection [1:0].

KBIHPS1~0	KBI7	KBI6	KBI5	KBI4
0 0	P1.7	P1.6	P1.5	P1.4
0 1	P1.7	P1.6	P3.5	P3.4
1 0	P2.6	P2.4	P2.3	P2.2
1 1	P1.7	P1.6	P6.1	P6.0

Bit 5~4: KBILPS1~0, KBI0~3 Port pin Selection 0.

KBILPS1~0	KBI3	KBI2	KBI1	KBI0
0 0	P1.3	P1.2	P1.1	P1.0
0 1	P3.3	P3.2	P3.1	P3.0
1 0	P0.7	P0.5	P0.2	P0.1
1 1	P4.5	P4.4	P4.1	P4.0

Bit 3: T3FCS, Reserved for chip test.

Bit 2: T2FCS, Reserved for chip test.

Bit 1: SnMIPS, S0MI, S1MI, S2MI & S3MI Port pin Selection.

SnMIPS	S0MI	S1MI
0	P1.6	P1.0
1	P3.3	P3.5

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

S0COPS	S0CKO
0	P4.7
1	P4.4

AUXR7: Auxiliary Register 7

SFR Page = 4 only

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	C0CKOE	C1CKOE	0	0	0	0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~6: Reserved. Software must write “0” on these bits when AUXR7 is written.

Bit 5: C0CKOE, PTM clock output (C0CKO) enable.
0: Disable PTM clock output.
1: Enable PTM clock output with PTM base timer overflow rate/2.

Bit 4: C1CKOE, PCA1 clock output (C1CKO) enable.
0: Disable PCA1 clock output.
1: Enable PCA1 clock output with PCA1 base timer overflow rate/2.

Bit 3~0: Reserved. Software must write “0” on these bits when AUXR7 is written.

AUXR8: Auxiliary Register 8

SFR Page = 5 only

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	C1ICS1	C1ICS0	0	0	S1COPS	T3PS1	T3PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: Reserved. Software must write “0” on these bits when AUXR8 is written.

Bit 6: C1ICS1, PCA1 Input Channel 1 input port pin Selection.

C1ICS1	C1EX1 input
0	C1EX1 Port Pin
1	T2EXI

Bit 5: C1ICS0, PCA1 Input Channel 0 input port pin Selection.

C1ICS0	C1EX0 input
0	C1EX0 Port Pin
1	T3EXI

Bit 4~3: Reserved. Software must write “0” on these bits when AUXR8 is written.

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

S1COPS	S1CKO
0	P4.7
1	P4.5

Bit 1~0: T3PS1~0, Timer 3 Port pin Selection [1:0].

T3PS1~0	T3/T3CKO	T3EX
0 0	P3.3	P3.4
0 1	P3.3	P3.2
1 0	P0.2	P0.1
1 1	P6.0	P6.1

AUXR9: Auxiliary Register 9

SFR Page = 6 only

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
SIDPS1	SIDPS0	T1G1	T0G1	0	0	S1PS1	S1PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: SID/STWI Port pin Selection [1:0].

SIDPS1~0	STWI_SCL	STWI_SDA
0 0	nINT1	S0MI
0 1	TWI0_SCL	TWI0_SDA
1 0	TWI1_SCL	TWI1_SDA
1 1	T2EXI	T3EXI

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

S1PS1~0	RXD1	TXD1
0 0	P1.2	P1.3
0 1	P6.0	P6.1
1 0	P1.0	P1.1
1 1	P3.4	P3.5

AUXR10: Auxiliary Register 10

SFR Page = 7 only
SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	SPIPS1	SPIPS0	S0PS1	P60OC1	P60OC0	P60FD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5~4: SPIPS1~0, SPI Port pin Selection 1~0.

Pin Options	SPIPS1~0	nSS	MOSI	MISO	SPICLK
0	0 0	P1.4	P1.5	P1.6	P1.7
1	0 1	P0.1	P0.2	P4.1	P4.0
2	1 0	P3.3	P1.5	P1.6	P1.7
3	1 1	P1.7	P3.5	P3.4	P3.3

Bit 3: S0PS1, Serial Port 0 pin Selection 1. (Its function is illustrated at AUXR3.3, S0PS0)

Bit 2~1: 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 crystal mode, XTAL2 and XTAL1 are the alternated function of P6.0 and P6.1. 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]	P6.0 function	I/O mode
0 0	P6.0	By P6M1.0 & P6M0.0
0 1	MCK	By P6M1.0 & P6M0.0
1 0	MCK/2	By P6M1.0 & P6M0.0
1 1	MCK/4	By P6M1.0 & 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 {P6M1.0, P6M0.0} to "01" which selects P6.0 as push-push output mode.

Bit 0: 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.

AUXR11: Auxiliary Register 11

SFR Page = 8 only
SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	I2C1PS1	I2C1PS0	RX1S0	0	T1CKOE	T0CKOE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5~4: I2C1PS1~0, TWI1/I2C1 Port pin in Selection [1:0].

I2C1PS1~0	TWI1_SCL	TWI1_SDA
0 0	P1.0	P1.1
0 1	P6.0	P6.1
1 0	P3.0	P3.1
1 1	TWI0_SCL	TWI0_SDA

Bit 3: RX1S0, RXD1 Selection 0.
0: RXD1 is selected by S1PS1~0 (AUXR9.1~0).
1: Force RXD1 is selected on AC0OUT.

AUXR12: Auxiliary Register 12

SFR Page = 9 only

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
CRCDS2	CRCM0	PDOES1	PDOES0	EDCM0	GPLC0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5~4: PDOES1~0, PDTXOE1 port pin Selection [1:0].

PDOES1~0	PDTXOE
0 0	P1.5
0 1	P3.3
1 0	P6.1
1 1	P3.2

AUXR16: Auxiliary Register 16

SFR Page = D only

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	C0OPS5	C0OPS4	C0OPS3	C0OPS2	C0OPS1	C0OPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5~0: C0OPS5 ~0, PTM0 Output Pin Selection.

	C0OPS5	C0OPS4	C0OPS3	C0OPS2	C0OPS1	C0OPS0
C0OPSn	CEX5	CEX4	CEX3	CEX2	CEX1	CEX0
0	P0.7	P2.6	P0.5	P2.4	P2.3	P2.2
1	P3.5	P3.2	P3.4	P4.1	P3.3	P4.0

AUXR17: Auxiliary Register 17

SFR Page = E only

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	0	0	0	0	C0OPS1	C0OPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 1~0: C1OPS1 ~0, PCA1 Output Pin Selection.

	C1OPS1	C1OPS0
C1OPSn	C1EX1	C1EX0
0	P6.1	P6.0
1	P3.1	P3.0

XICFG: External Interrupt Configured Register

SFR Page = 0 only

SFR Address = 0xC1 RESET = 0000-0000

7	6	5	4	3	2	1	0
INT1IS.1	INT1IS.0	INT0IS.1	INT0IS.0	X3FLT	X2FLT	X1FLT	X0FLT
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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
0 0 0	P3.3
0 0 1	P3.1
0 1 0	P3.5
0 1 1	P4.1
1 0 0	P4.5
1 0 1	P1.5
1 1 0	P1.0
1 1 1	P0.7

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

0 0 0	P3.2
0 0 1	P3.0
0 1 0	P3.4
0 1 1	P4.0
1 0 0	P1.0
1 0 1	P1.4
1 1 0	P1.7
1 1 1	P0.5

XICFG1: External Interrupt Configured 1 Register

SFR Page = 1 only

SFR Address = 0xC1 RESET = 0000-0000

7	6	5	4	3	2	1	0
INT1IS.2	INT0IS.2	INT2IS.1	INT2IS.0	X3FLT1	X2FLT1	X1FLT1	X0FLT1
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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 with INT2IS.2 as following table.

INT2IS.2~0	Selected Port Pin of nINT2
0 0 0	P4.4
0 0 1	P3.0
0 1 0	P1.1
0 1 1	P1.6
1 0 0	P1.4
1 0 1	P6.0
1 1 0	P0.1
1 1 1	P6.1

XICFG2: External Interrupt Configured 2 Register

SFR Page = 2 only

SFR Address = 0xC1 RESET = XXXX-0000

7	6	5	4	3	2	1	0
--	--	--	--	INT3IS.2	INT3IS.1	INT3IS.0	INT2IS.2
W	**W**	**W**	**W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 3~1: INT3IS.2~0, nINT3 input port pin selection bits which function is defined as following table.

INT3IS.2~0	Selected Port Pin of nINT3
0 0 0	P4.5
0 0 1	P2.2
0 1 0	P1.5
0 1 1	P1.6
1 0 0	P0.2
1 0 1	P6.1
1 1 0	P1.2
1 1 1	P6.0

Bit 0: INT2IS2, nINT2 input port pin selection bit which function is defined with INT2IS.1~0.

5. 8051 CPU Function Description

5.1. CPU Register

PSW: Program Status Word

SFR Page = 0~F
SFR Address = 0xD0 RESET = 0000-0000

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

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 RESET = 0000-0111

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

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 RESET = 0000-0000

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

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 RESET = 0000-0000

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

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 = 0xE0 RESET = 0000-0000

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

This register is the accumulator for arithmetic operations.

B: B Register

SFR Page = 0~F

SFR Address = 0xF0 RESET = 0000-0000

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

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

5.2. CPU Timing

The **MG82F6P32** 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 “38 Instruction Set” which includes the mnemonic, number of bytes, and number of clock cycles for each instruction.

5.3. CPU Addressing Mode

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.

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.

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.

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.

Immediate Constant (IMM)
The value of a constant can follow the op-code in the program memory.

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 **MG82F6P32** 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 **32K** bytes of program memory. In the **MG82F6P32**, 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.

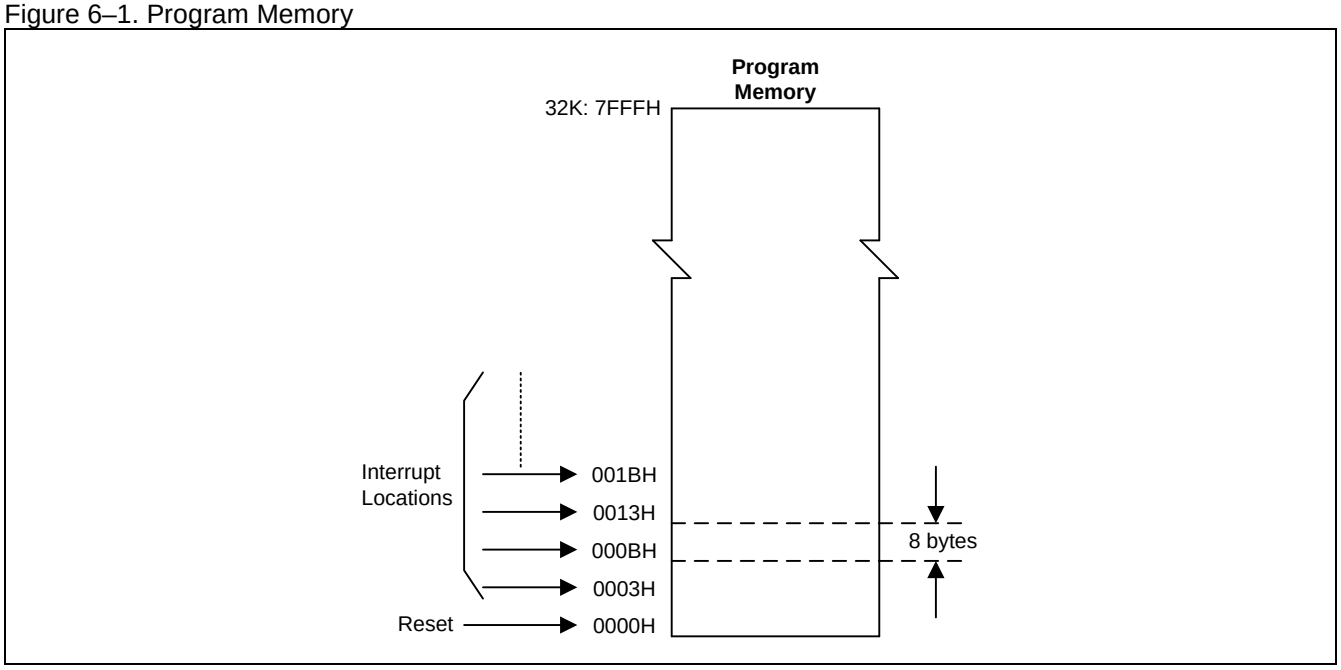
Data memory occupies a separate address space from program memory. In the **MG82F6P32**, there are 256 bytes of internal scratch-pad RAM and **1792** bytes of on-chip expanded RAM (XRAM).

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 **MG82F6P32** 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 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

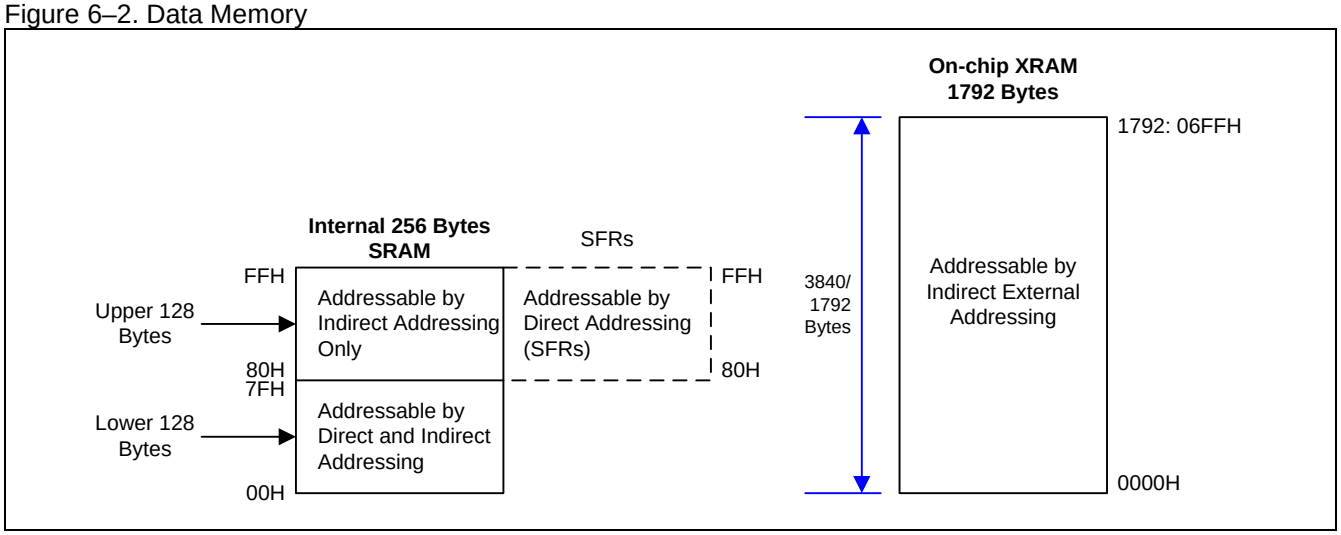


Figure 6–3. Lower 128 Bytes of Internal RAM

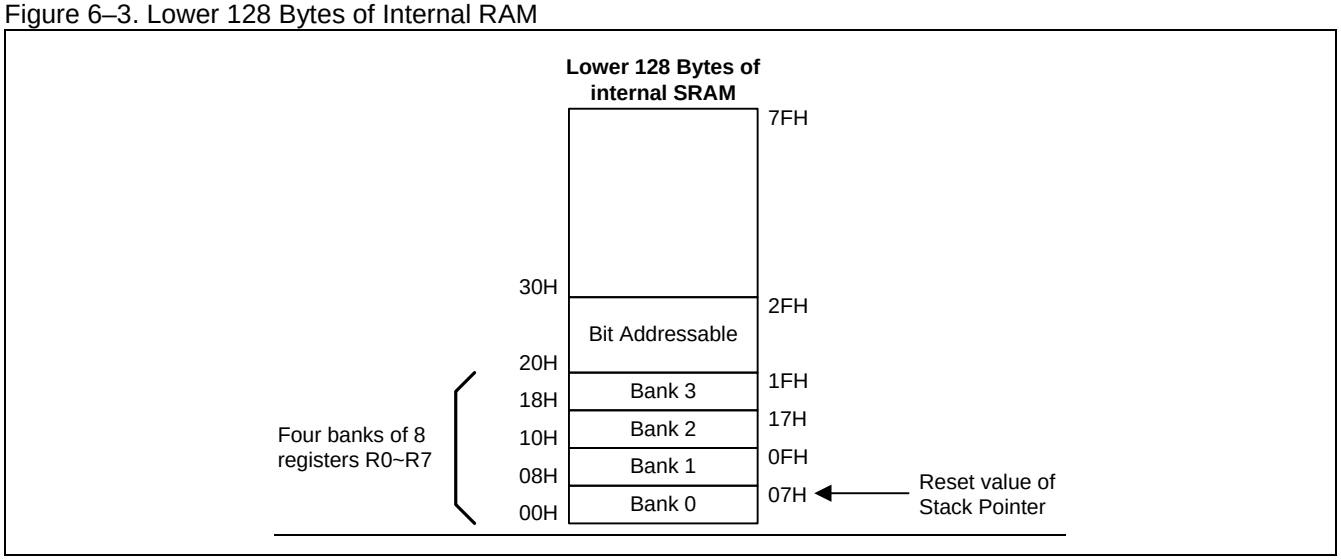
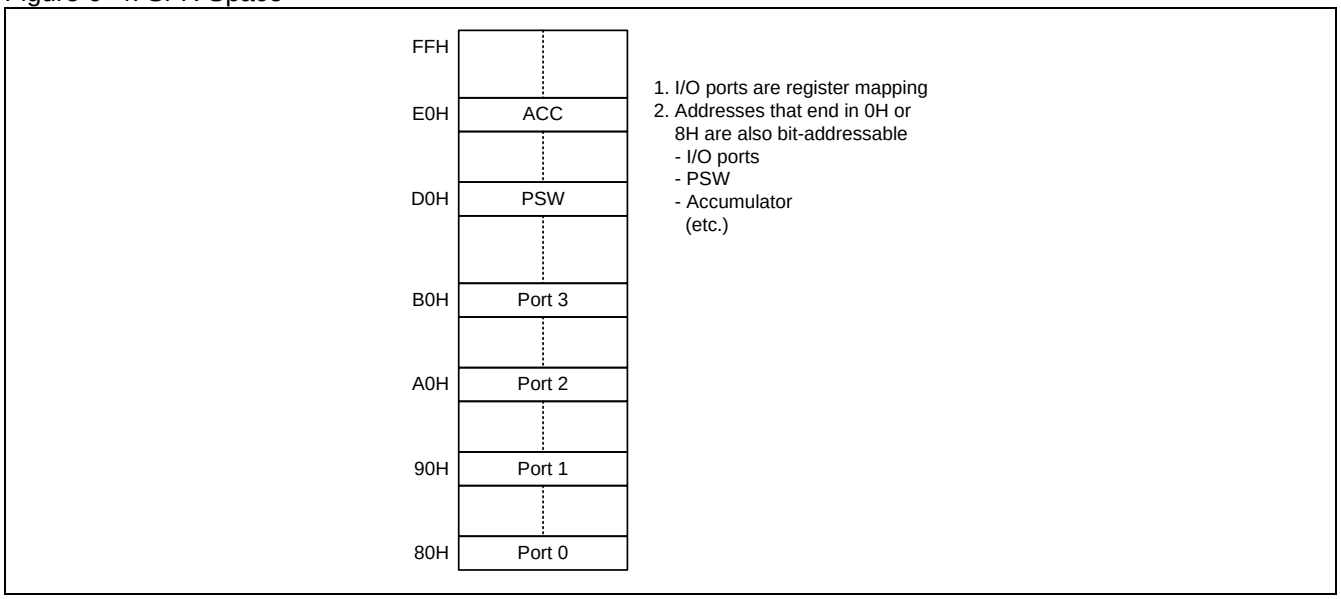


Figure 6–4. SFR Space



6.3. On-chip expanded RAM (XRAM)

To access the on-chip expanded RAM (XRAM), refer to Figure 6–2, the **1792** bytes of XRAM (0000H to **06FFH**) 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 **MG82F6P32** hardware can access them correctly.

6.4. EMB, Off-Chip External data Memory Bus

The off-chip external data memory access function is not supported in **MG82F6P32**.

6.5. Declaration Identifiers in a C51-Compiler

The declaration identifiers in a C51-compiler for the various **MG82F6P32** 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 **MG82F6P32** has **1792** 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 **MG82F6P32** has 256 bytes of on-chip pdata memory which is shared with on-chip xdata memory.

code
32K bytes of program memory space; accessed as part of program execution and via the “MOVC @A+DTPR” instruction. The **MG82F6P32** has **32K** bytes of on-chip code memory.

7. XRAM Access

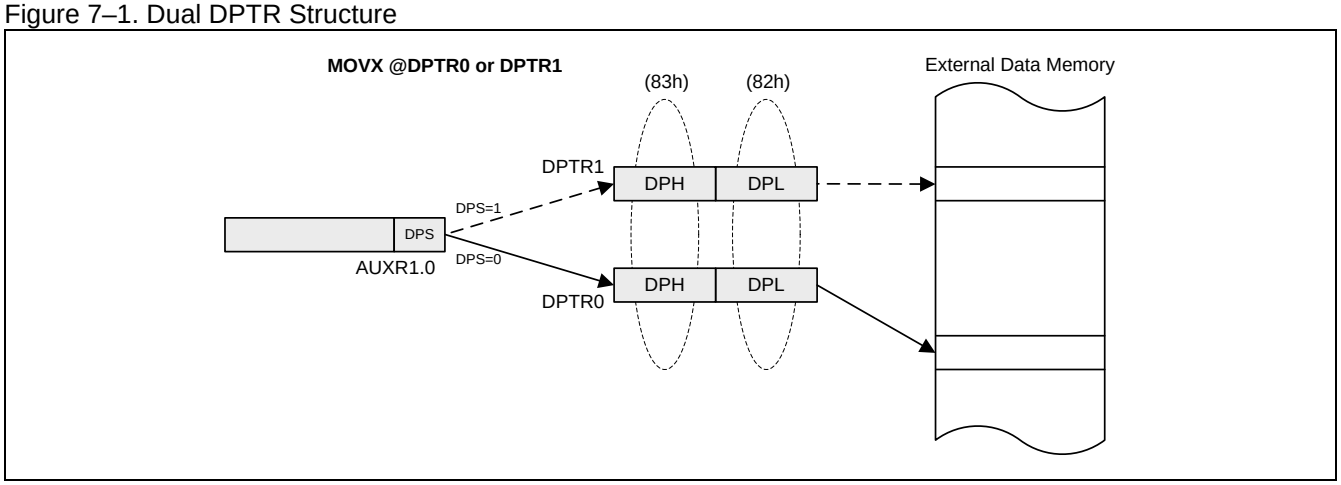
The **MG82F6P32** MCUs include **1792** bytes of on-chip RAM mapped into the external data memory space (XRAM). The external memory space may be accessed using the external move instruction (MOVX) and the data pointer (DPTR), or using the MOVX indirect addressing mode using R0 or R1. If the MOVX instruction 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.

Figure 7–1. Dual DPTR Structure



DPTR Instructions

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

```
INC  DPTR              ; Increments the data pointer by 1
MOV  DPTR,#data16      ; Loads 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 RESET = 0000-0000

7	6	5	4	3	2	1	0
OP1Fr	OP0Fr	CRCDS1	CRCDS0	0	AC1Fr	AC0Fr	DPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 RESET = 0000-0000

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

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 RESET = 0000-0000

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

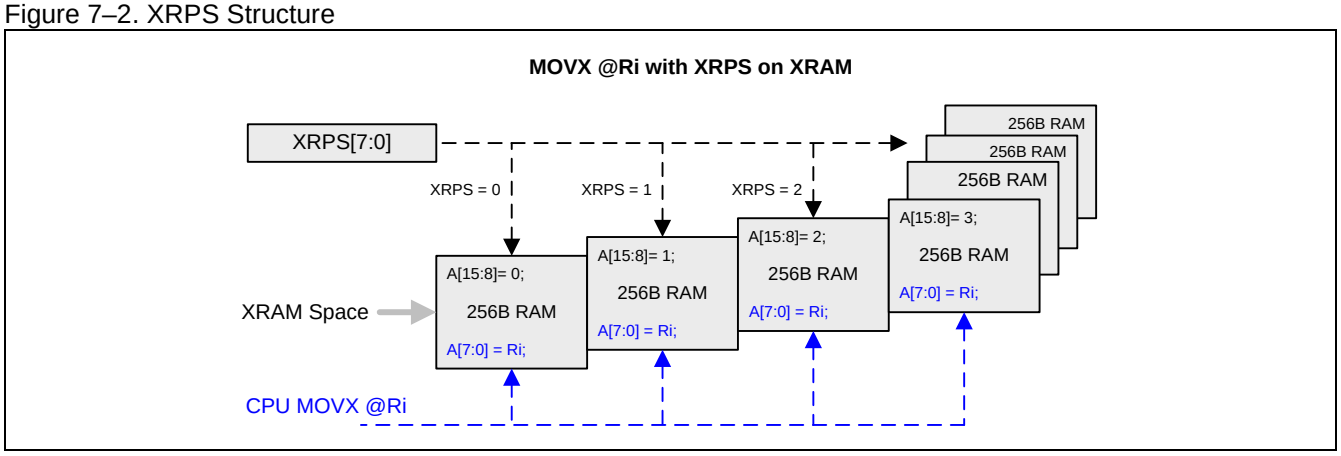
The DPH register is the high byte of the 16-bit DPTR. DPTR is used to access indirectly addressed XRAM and Flash memory.

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.

This function can give the designer to get more efficiency code to access XRAM. To access whole range of XRAM will need to use 2 bytes address. The software complier will compile the 2 bytes address by using DPTR to access specific memory location, it results to add more instructions and will slow down the efficiency. But if use the XRPS with global “pdata” variables, the complier will translate it to MOVX@Ri to reduce many extra instructions to enhance the memory access performance.

Figure 7–2. XRPS Structure



XRPS: XRAM Page Select Register

SFR Page = 0~F

SFR Address = 0x8F RESET = 0000-0000

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

Bit 7~4: Reserved. Software must write “0” on these bits when XRPS is written.

Bit 3~0: 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 **MG82F6P32**, XRPS indexes the **7** pages 256-byte RAM (XRPS = 0000 ~ 0110).

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.

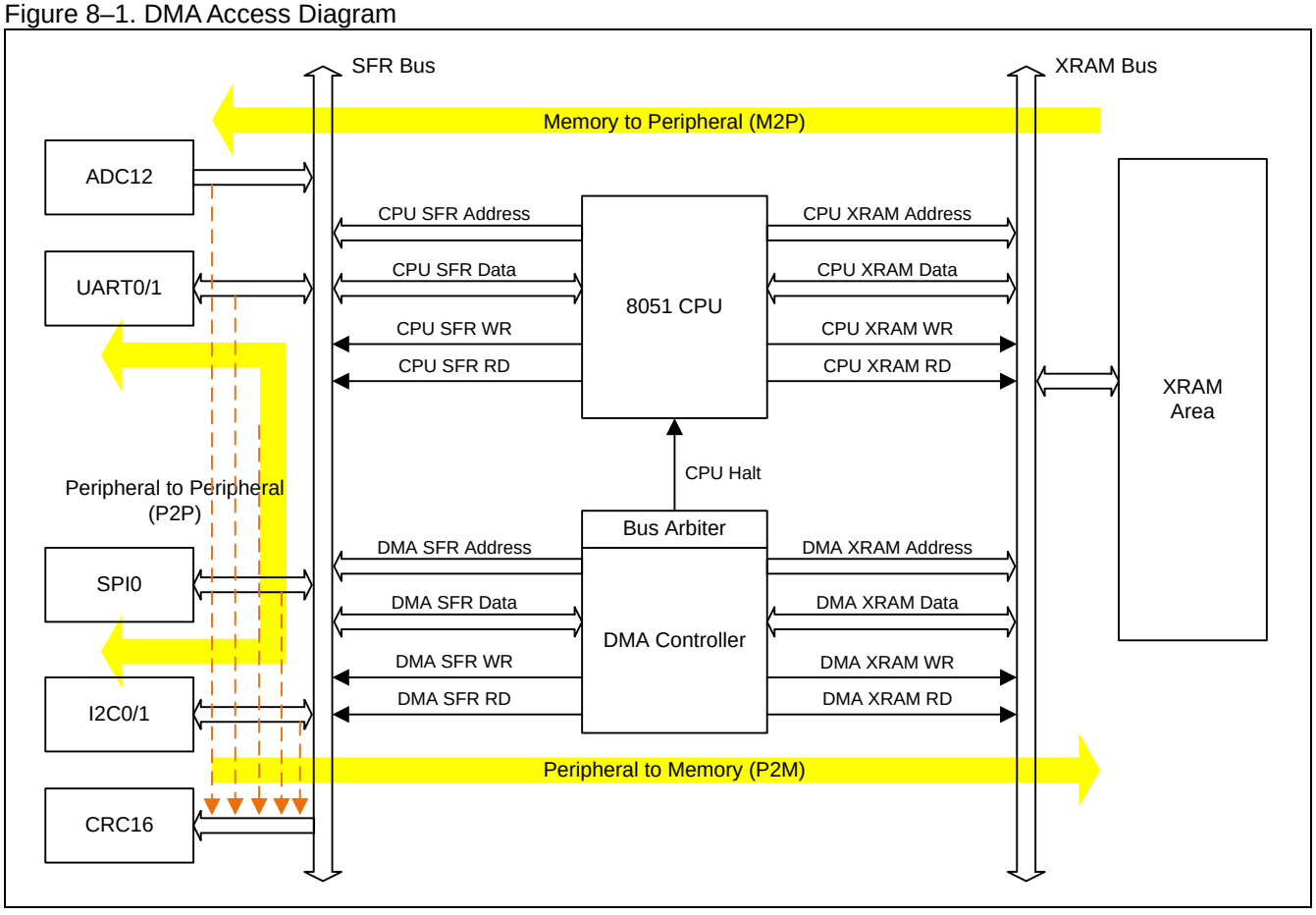
Using the DMA controller can increase the throughput of peripheral modules. It can also reduce system power consumption by allowing the CPU to remain in a low-power mode without having to awaken to move data to or from a peripheral.

The DMA controller features include:

- Easy use one channel DMA
- 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 Access Diagram



8.1. DMA Structure

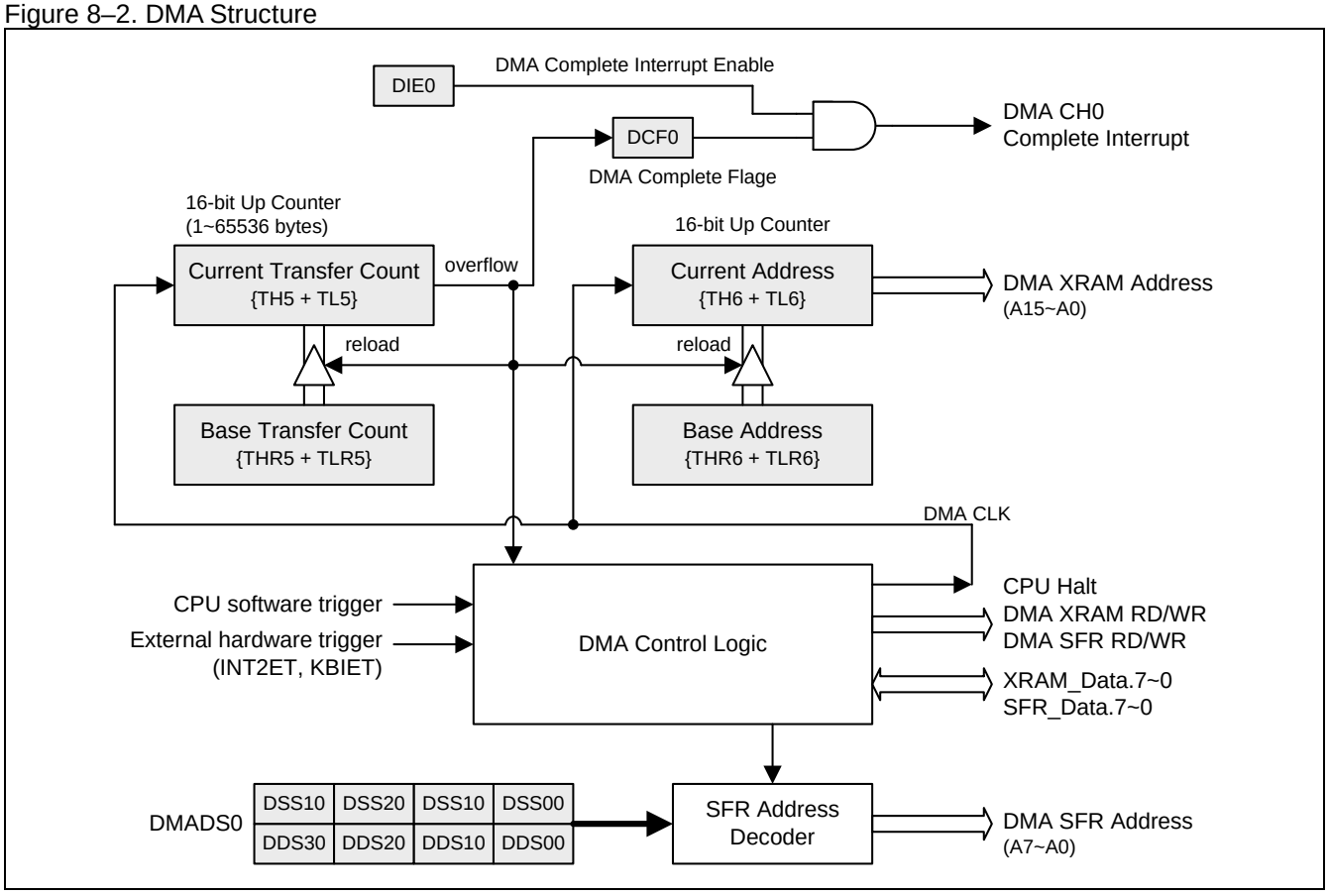
In **MG82F6P32**, the DMA controller provides one channel DMA to support 3 transfer types: transfer the data from XRAM to peripheral, from peripheral to XRAM and from peripheral to peripheral. DMADS0 register in DMA channel 0 defines the DMA transfer type to configure DMA controller behavior and defines 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 **MG82F6P32** 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.

Table 8–1. 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	S0 RX	0 0 0 1	S0 TX
0 0 1 0	S1 RX	0 0 1 0	S1 TX
0 0 1 1		0 0 1 1	
0 1 0 0		0 1 0 0	
0 1 0 1	TWI0 RX	0 1 0 1	TWI0 TX
0 1 1 0	TWI1 RX	0 1 1 0	TWI1 TX
0 1 1 1	SPI0 RX	0 1 1 1	SPI0 TX
1 0 0 0		1 0 0 0	
1 0 0 1	ADC0	1 0 0 1	
1 0 1 0		1 0 1 0	
1 0 1 1		1 0 1 1	
1 1 0 0		1 1 0 0	
1 1 0 1		1 1 0 1	CRC
1 1 1 0		1 1 1 0	
1 1 1 1	XRAM	1 1 1 1	XRAM

Note:
The DMA does not support the transfer of "XRAM to XRAM". Other paths are supported in DMA transfer.

8.2.2. DMA Transfer Mode

The DMA controller in **MG82F6P32** 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 **MG82F6P32**, 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 **MG82F6P32**.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.

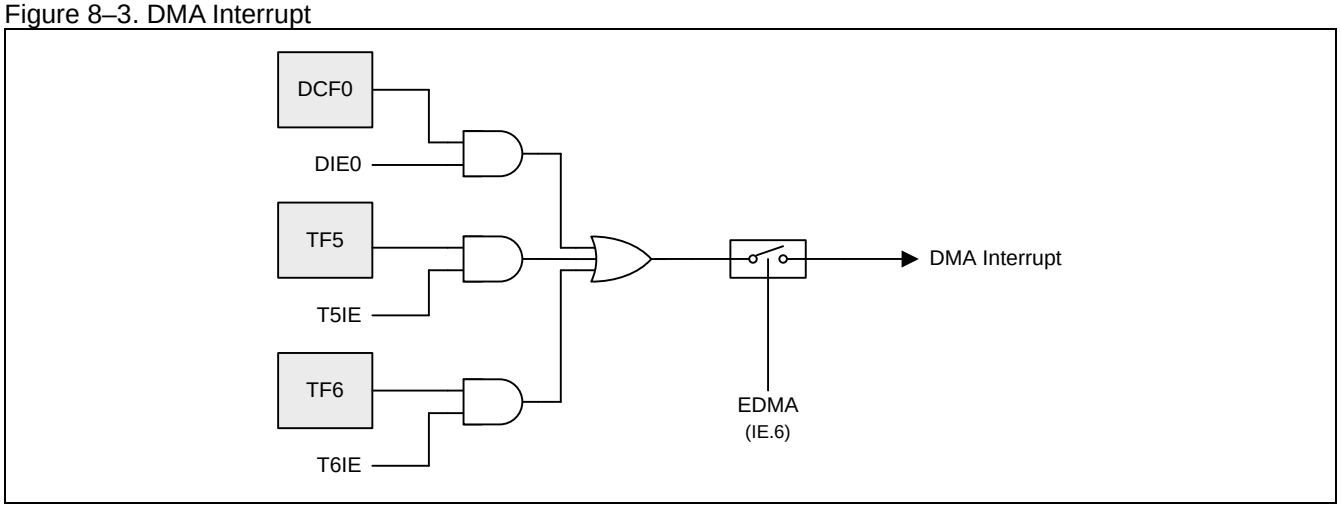
Software can write "0" on DMAE0 to stop current DMA transfer at any time or to end the DMA transfer after End of DMA transfer. It is recommended software must also configure data path (DMADS0) to disable state and clear DMAS0.

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 not any error handling function in the DMA controller, software will take care on:
a. Current Address cannot over the XRAM boundary. In **MG82F6P32**, XRAM boundary is **1792** bytes (**06FFH**).
b. Cannot support the even/odd parity check and generation on S0, S1.
c. Cannot handle the Not ACK status on TWI0/I2C0 and TWI1/I2C1.

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 feed the data to CRC16 simultaneously. This function is supported in any transfer type.

8.2.10. Timer 5 & Timer 6

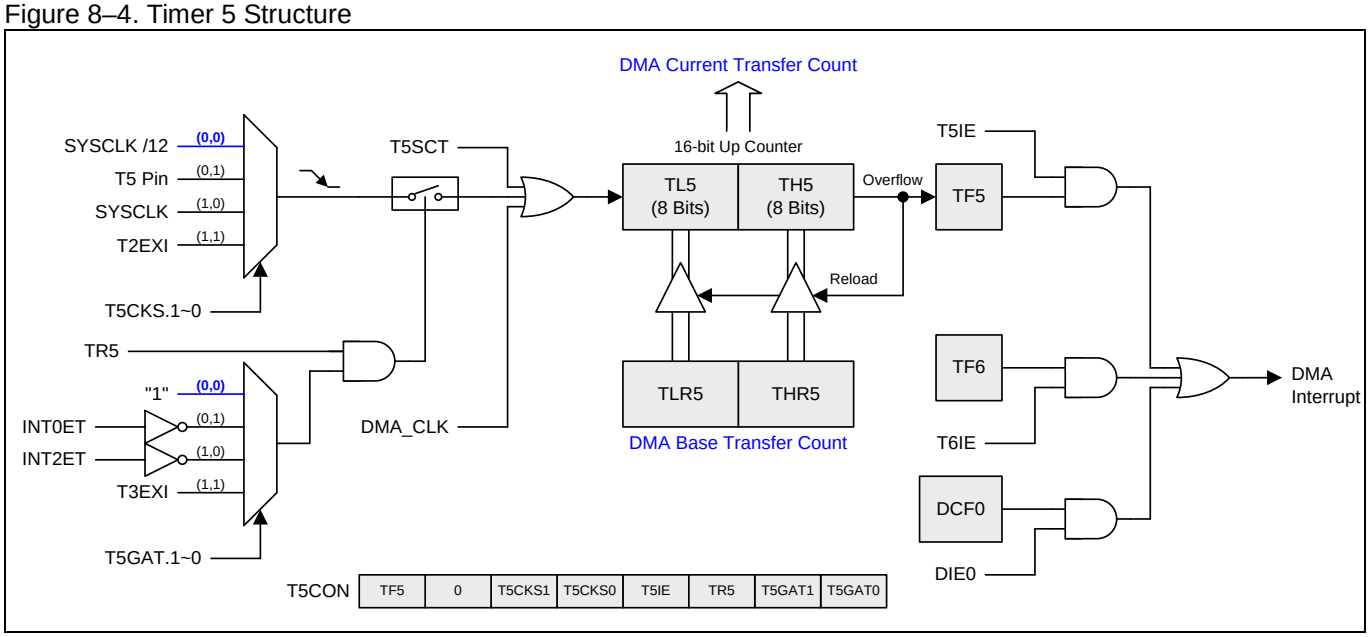
When DMA enabled, Timer 5 behaves the function for DMA transfer counting. TH5 and TL5 are the Current Transfer Count registers. THR5 and TLR5 are the Base Transfer Count registers.

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.

Timer 5 Pin configuration is as following:

T5
P3.4

Figure 8–4. Timer 5 Structure



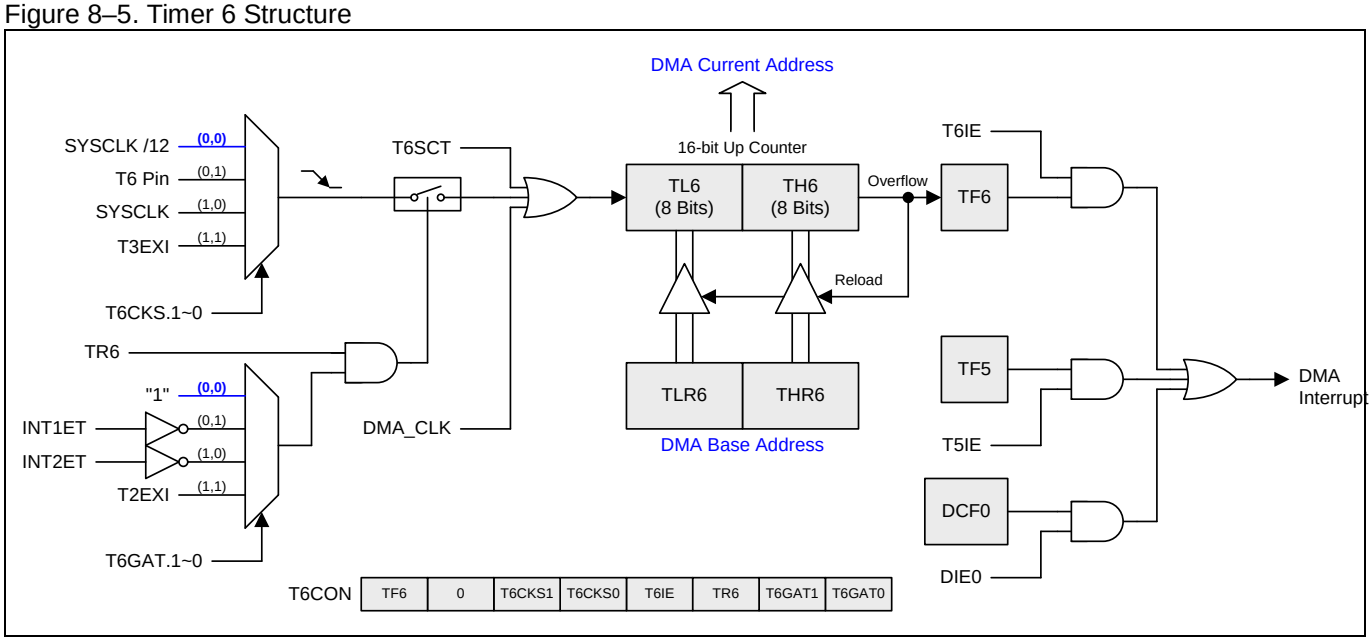
When DMA enabled, Timer 6 behaves the function for DMA memory address pointer. TH6 and TL6 are the Current Address registers. THR6 and TLR6 are the Base Address registers.

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.

Timer 6 Pin configuration is as following:

T6
P3.5

Figure 8–5. Timer 6 Structure



8.3. DMA Register

DMACR0: DMA Control Register 0

SFR Page = **0~7**

SFR Address = 0x94 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	0	0	DMAE0	DMAS0	DIE0	DCF0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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 RESET = 0000-0000

7	6	5	4	3	2	1	0
PDMAH	PDMAL	CRCW0	0	EXTS10	EXTS00	0	LOOP0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~6: PDMAH/L, DMA interrupt priori/y 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

Bit 1: Reserved. Software must write “0” on this bit when DMACG0 is written.

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 RESET = 0000-0000

7	6	5	4	3	2	1	0
DSS30	DSS20	DSS10	DSS00	DDS30	DDS20	DDS10	DDS00
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~4: DMA data Source Selection.

Bit 3~0: DMA data Destination 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	S0 RX	0 0 0 1	S0 TX
0 0 1 0	S1 RX	0 0 1 0	S1 TX
0 0 1 1	Reserved	0 0 1 1	Reserved
0 1 0 0	Reserved	0 1 0 0	Reserved
0 1 0 1	TWI0 RX	0 1 0 1	TWI0 TX
0 1 1 0	TWI1 RX	0 1 1 0	TWI1 TX
0 1 1 1	SPI0 RX	0 1 1 1	SPI0 TX
1 0 0 0	Reserved	1 0 0 0	Reserved
1 0 0 1	ADC0	1 0 0 1	Reserved
1 0 1 0	Reserved	1 0 1 0	Reserved
1 0 1 1	Reserved	1 0 1 1	Reserved
1 1 0 0	Reserved	1 1 0 0	Reserved
1 1 0 1	Reserved	1 1 0 1	CRC
1 1 1 0	Reserved	1 1 1 0	Reserved
1 1 1 1	On-chip XRAM	1 1 1 1	On-chip XRAM

Note 1: When use DMA to transfer ADC data, please watch out the Data Bit setting. Please reference 29.2.9 Transfer ADC Data by DMA

8.4. Timer 5 Register

T5CON: Timer 5 Control Register

SFR Page = **3 Only**

SFR Address = 0xC8 RESET = 0000-0000

7	6	5	4	3	2	1	0
TF5	0	T5CKS1	T5CKS0	T5IE	TR5	T5GAT1	T5GAT0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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: Reserved. Software must write “0” on this bit when T5CON is written.

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

T5CKS.1~0	T5 Clock Selection
00	SYSCLK/12
01	T5 Pin
10	SYSCLK
11	T2EXI input selection

T5 Pin: P3.4

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	Inverted INT0ET
10	Inverted INT2ET
11	T3EXI input selection

TL5: Timer 5 Low byte Register

SFR Page = **3 Only**

SFR Address = 0xCC RESET = 0000-0000

7	6	5	4	3	2	1	0
TL5.7	TL5.6	TL5.5	TL5.4	TL5.3	TL5.2	TL5.1	TL5.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

TH5: Timer 5 High byte Register

SFR Page = **3 Only**

SFR Address = 0xCD RESET = 0000-0000

7	6	5	4	3	2	1	0
TH5.7	TH5.6	TH5.5	TH5.4	TH5.3	TH5.2	TH5.1	TH5.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

TLR5: Timer 5 Low byte Reload Register

SFR Page = **3 Only**

SFR Address = 0xCA RESET = 0000-0000

7	6	5	4	3	2	1	0
TLR5.7	TLR5.6	TLR5.5	TLR5.4	TLR5.3	TLR5.2	TLR5.1	TLR5.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

THR5: Timer 5 High byte Reload Register

SFR Page = **3 Only**

SFR Address = 0xCB RESET = 0000-0000

7	6	5	4	3	2	1	0
THR5.7	THR5.6	THR5.5	THR5.4	THR5.3	THR5.2	THR5.1	THR5.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

8.5. Timer 6 Register

T6CON: Timer 6 Control Register

SFR Page = **4 Only**

SFR Address = 0xC8 RESET = 0000-0000

7	6	5	4	3	2	1	0
TF6	0	T6CKS1	T6CKS0	T6IE	TR6	T6GAT1	T6GAT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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: Reserved. Software must write “0” on this bit when T6CON is written.

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

T6 Pin: P3.5

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 5.

T6GAT.1~0	T6 Gate source
00	Disable
01	Inverted INT1ET
10	Inverted INT2ET
11	T2EXI input selection

TL6: Timer 6 Low byte Register

SFR Page = **4 Only**

SFR Address = 0xCC RESET = 0000-0000

7	6	5	4	3	2	1	0
TL6.7	TL6.6	TL6.5	TL6.4	TL6.3	TL6.2	TL6.1	TL6.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TH6: Timer 6 High byte Register

SFR Page = **4 Only**

SFR Address = 0xCD RESET = 0000-0000

7	6	5	4	3	2	1	0
TH6.7	TH6.6	TH6.5	TH6.4	TH6.3	TH6.2	TH6.1	TH6.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TLR6: Timer 6 Low byte Reload Register

SFR Page = **4 Only**

SFR Address = 0xCA RESET = 0000-0000

7	6	5	4	3	2	1	0
TLR6.7	TLR6.6	TLR6.5	TLR6.4	TLR6.3	TLR6.2	TLR6.1	TLR6.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

THR6: Timer 6 High byte Reload Register

SFR Page = **4 Only**

SFR Address = 0xCB RESET = 0000-0000

7	6	5	4	3	2	1	0
THR6.7	THR6.6	THR6.5	THR6.4	THR6.3	THR6.2	THR6.1	THR6.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

9. System Clock

There are **four** clock sources for the system clock: Internal High-frequency RC Oscillator (IHRCO), external crystal oscillator, Internal Low-frequency RC Oscillator (ILRCO) and External Clock Input. Figure 9–1 shows the structure of the system clock in **MG82F6P32**.

The **MG82F6P32** always boots from IHRCO on 12MHz and reserves crystal pads as P6.0/P6.1 GPIO function. Software can select the OSCin input on one of the **four** clock sources application required and switches them on the fly. But software needs to settle the clock source stably before clock switching. If software selects external crystal mode, port pin of P6.0 and P6.1 will be assigned to XTAL2 and XTAL1. And P6.0/P6.1 GPIO function will be inhibited. In external clock input mode (ECKI), the clock source comes from P6.0 input and P6.1 still reserves GPIO function.

Set XTALE (CKCON2.5) to enable external crystal oscillating. When frequency from external crystal had been stable, the XTOR (CKCON1.7) will be set automatically. Software need to poll this bit before switch the system clock source (OSCin) to external crystal. XTOR is read only.

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 “37.5 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, RTC, 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 “37.6 ILRCO Characteristics”). In ILRCO mode, P6.0 can be configured to internal *MCK* output or *MCK*/2 and *MCK*/4 for system application.

The **MG82F6P32** device includes a Clock Multiplier (CKM) to generate the high-speed clock for system clock source. CKM applied in **MG82F6P32** 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 (CKCON2.3~2) 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 “37.7 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 **MG82F6P32**. The initial oscillator source of CPUCLK is set to IHRCO 12MHz. It can use the combinations of the clock multiplier and divider for different frequencies. The maximum CPUCLK is as following:

- External crystal 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 32MHz @ 2.7V -5.5V with on-chip CKM

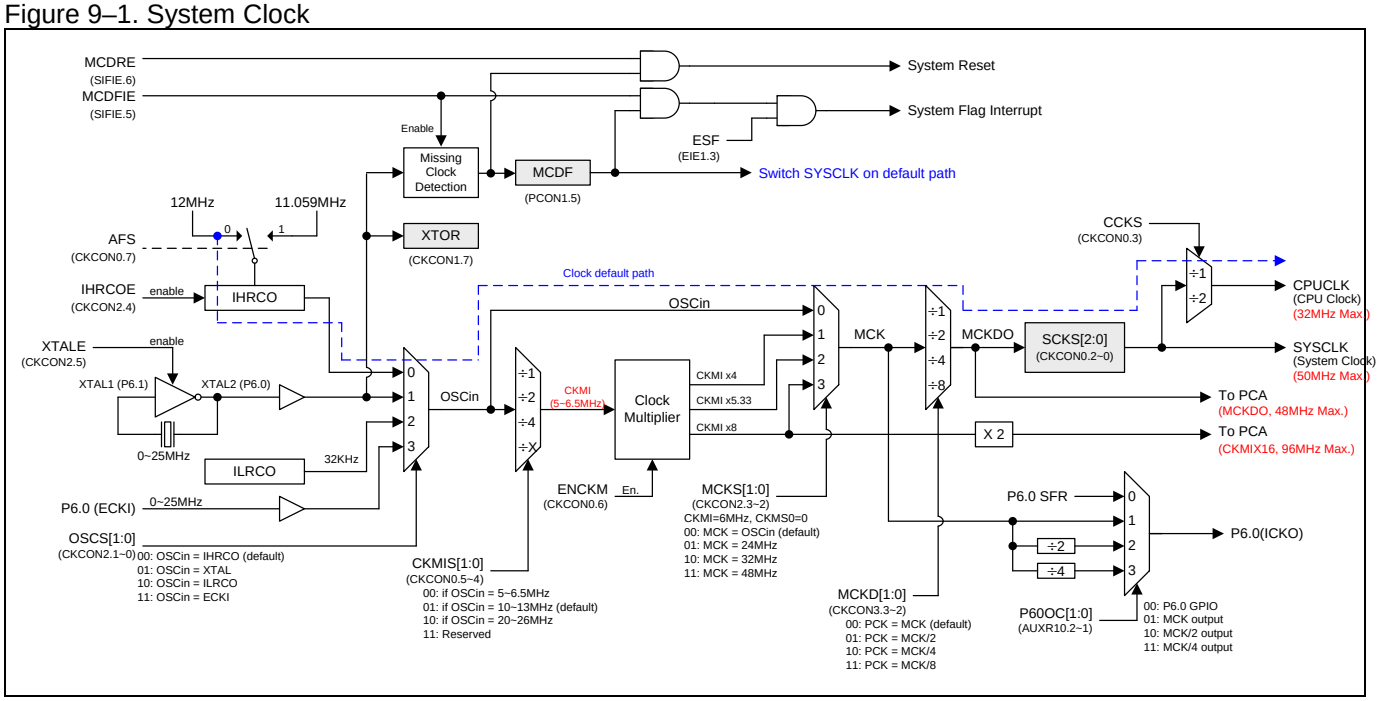
If the applications need higher performance, then HSE (DCON0 Bit 7) needs to be set when CPUCLK > 6MHz. Moreover, if needs ultra-high CPUCLK>25MHz, then HSE1 needs to be set.

The system clock can be sourced by the external oscillator circuit or internal oscillator. Its maximum frequency is 50MHz. Please note, when using Clock Multiplier (CKM) to raise the MCK frequency to get higher SYSCLK, the CPUCLK will be also changed. It is needed to set CCKS to slow down CPUCLK before raise MCK frequency to avoid CPUCLK over clock (CPUCLK needs to lower then 25MHz or 32MHz).

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

- MCKDO: Up to 48MHz
- CKMIX16: Up to 96MHz

Figure 9–1. System Clock



9.2. Clock Source Switching

There are four clock sources for the system clock: Internal High-frequency RC Oscillator (IHRCO), external crystal oscillator, Internal Low-frequency RC Oscillator (ILRCO) and External Clock Input. Figure 9–1 shows the structure of the system clock in **MG82F6P32**. The **MG82F6P32** 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)

The **MG82F6P32** includes a Clock Multiplier (CKM) to generate the high-speed clock for system clock source. It 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 suitable CKMI frequency for CKM input source. CKM can generate 4/5.33/8 times frequency of CKMI and setting MCKS1~0 (CKCON2.3~2) selects different CKM outputs on MCK to provide the high-speed operation on MCU without high-frequency clock source. To find the detailed CKM performance, please refer Section "37.7 CKM Characteristics").

9.4. Missing Clock Detection (MCD)

When using the external crystal oscillator as the clock source, it can be monitored by the missing clock detector MCD to notify if the crystal is out of function. The Missing-Clock-Detection module is enabled by MCDFIE. If MCDFIE is cleared, the Missing-Clock-Detection module is inactive. PCON1 Bit 5, MCDF, Missing Clock Detection (MCD) flag set by hardware to detect a Missing-Clock event on external crystal oscillating input. Writing **"1"** on this bit will clear MCDF. Once a missing clock event happened, software must clear MCDF before switching OSCin to XTAL again. In default, the MCD event will trigger a system reset. If user won't apply this function, clear MCDRE to disable the reset function.

9.5. Fast wake-up for XTAL mode

Normally the settle time of the external crystal is 0.6ms ~ 2ms (depends on the applications). The MCU stable time is 200us and is fast then crystal. It can be used to speed up the wake-up time for XTAL mode. Under XTAL mode, user can switch the OSCin from XTAL to IHRCO before the system go into Power-Down Mode.

How to Program with Fast wake-up function for XTAL mode

- Set IHRCOE (CKCON2.4) to "1" to enable IHRCO.
- delay 32us to wait IHRCO working stable
- Program OSCS[1:0] (CKCON2.1~0) to "00" to select IHRCO as clock source
- NOP x 10
- Set XTALE (CKCON2.5) to "0" to disable external crystal oscillating circuit.
- MCU enter power down
- ……..
- MCU wake up
- Set XTALE (CKCON2.5) to "1" to enable external crystal oscillating circuit.
- Poll XTOR (CKCON1.7) to "1" to wait external crystal oscillating ready.
- Program OSCS[1:0] (CKCON2.1~0) to "01" to select XTAL as clock source
- NOP x 10
- Set IHRCOE (CKCON2.4) to "0" to disable IHRCO.
- Continue program execution…….

9.6. 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…….

9.7. Clock Register

CKCON0: Clock Control Register 0

SFR Page = **0~F & P**

SFR Address = 0xC7 RESET = 0001-0000

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

Bit 7: AFS, Alternated Frequency Selection.
0: Select IHRCO on 12MHz.
1: Select IHRCO on 11.059MHz.

Bit 6: ENCKM, Enable clock multiplier (X8)
0: Disable the x8 clock multiplier.
1: Enable the x8 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.

Bit 2~0: SCKS2 ~ SCKS0, programmable System Clock Selection.

SCKS[2:0]	System Clock (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

CKCON1: Clock Control Register 1

SFR Page = **0~F**

SFR Address = 0xBF RESET = 0001-0001

7	6	5	4	3	2	1	0
XTOR	PLLOCKF	MCKSTA1	MCKSTA0	OSCSTA3	OSCSTA2	OSCSTA1	OSCSTA0
R	R	R	R	R	R	R	R

Bit 7: XTOR, Crystal Oscillating Ready. Read Only.
0: Crystal Oscillating not Ready.
1: Crystal Oscillating Ready. When XTALE is enabled, XTOR reports the crystal oscillator reached start-up count.

Bit 6: PLLOCKF, Read only, to indicate the PLL lock ready flag
0: PLL is not lock
1: PLL is locked.

Bit 5~4: MCKSTA[1:0], MCK MUX Status
00: MCK MUX going on switching clock
01: MCK MUX is using OSCin clock source
10: MCK MUX is using CKMIx4/x5.33/x8 clock source
11: MCU MUX is going on switching clock

Bit 3~0: OSCSTA[3:0]
0001: OSCin MUX is using IHRCO clock source

0010: OSCin MUX is using XTAL clock source
0100: OSCin MUX is using ILRCO clock source
1000: OSCin MUX is using ECKI clock source
Others: OSCin MUX is going on switching clock

CKCON2: Clock Control Register 2

SFR Page = **P Only**
SFR Address = 0x40 RESET = 0101-0000

7	6	5	4	3	2	1	0
XTGS1	XTGS0	XTALE	IHRCOE	MCKS1	MCKS0	OSCS1	OSCS0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~6: XTGS1~XTGS0, Crystal Gain Selection register.

XTGS1, XTGS0	Gain Define	Applied Crystal
0, 0	Low Gain	32.768KHz
0, 1	Medium Gain	2MHz ~ 25MHz
1, 0	Lower Gain	32.768KHz
1, 1	Reserved	Reserved

Bit 5: XTALE, external Crystal (XTAL) Enable.
0: Disable XTAL oscillating circuit. In this case, XTAL2 and XTAL1 behave as Port 6.0 and Port 6.1.
1: Enable XTAL oscillating circuit. If this bit is set by CPU software, software polls the XTOR (CKCON1.7) true to indicate the crystal oscillator is ready for OSCin clock selected.

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 Source Selection	OSCin =12MHz CKMIS = [01]	OSCin =11.059MHz CKMIS = [01]
0 0	OSCin	12MHz	11.059MHz
0 1	CKMI x4	24MHz	22.118MHz
1 0	CKMI x5.33	32MHz	29.491MHz
1 1	CKMI x8	48MHz	44.236MHz

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 ≤ 32MHz, SYSCLK ≤ 50MHz.

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

OSCS[1:0]	OSCin source Selection
0 0	IHRCO
0 1	XTAL
1 0	ILRCO
1 1	ECKI, External Clock Input (P6.0) as OSCin.

CKCON3: Clock Control Register 3

SFR Page = **P only**

SFR Address = 0x41 RESET = 0000-0010

7	6	5	4	3	2	1	0
WDTCS1	WDTCS0	FWKP	WDTFS	MCKD1	MCKD0	MCDS1	MCDS0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~6: WDTCS1~0. 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	if MCK = 12MHz	if MCK = 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: MCDS[1:0], Reserve for test

AUXR10: Auxiliary Register 10

SFR Page = 7 only

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	SPIPS1	SPIPS0	S0PS1	P60OC1	P60OC0	P60FD
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 2~1: 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 crystal mode, XTAL2 and XTAL1 are the alternated function of P6.0 and P6.1. 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 P6M1.0 & P6M0.0
01	MCK	By P6M1.0 & P6M0.0
10	MCK/2	By P6M1.0 & P6M0.0
11	MCK/4	By P6M1.0 & P6M0.0

For clock-out on P6.0 function, it is recommended to set {P6M1.0, P6M0.0} to “01” which selects P6.0 as push-push output mode.

Bit 0: 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.

PCON1: Power Control Register 1

SFR Page = 0~F & P

SFR Address = 0x97 POR = 0000-0000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 5: MCDF, Missing Clock Detection 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 which detects a Missing-Clock event. Writing **"1"** on this bit will clear MCDF. The Missing-Clock-Detection module is enabled by MCDFIE. If MCDFIE is cleared, the Missing-Clock-Detection module is inactive. Once a missing clock event happened, software must clear MCDF before switching OSCin to XTAL.

SFIE: System Flag Interrupt Enable Register

SFR Page = 0~F
SFR Address = 0x8E POR = 0110-0000

7	6	5	4	3	2	1	0
SIDFIE	MCDRE	MCDFIE	RTCFIE	SPWIE	BOF1IE	BOF0IE	WDTFIE
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 6: MCDRE, Enable Missing-Clock-Detection event causes a system reset.
0: Disable MCD event to trigger a system Reset.
1: Enable MCD event to trigger a system Reset.

Bit 5: MCDFIE, Enable MCDF (PCON1.5) Interrupt.
0: Disable MCDF interrupt.
1: Enable MCD module and enable MCDF interrupt.

DCON0: Device Control Register 0

SFR Page = P Only
SFR Address = 0x4C POR = 1000-0011

7	6	5	4	3	2	1	0
HSE	IAPO	HSE1	0	0	IORCTL	RSTIO	OCDE
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: HSE, High Speed operation Enable.
0: Select CPU running in lower speed mode ($F_{CPUCLK} \leq 6MHz$) which is slow down internal circuit to reduce power consumption.
1: Enable CPU full speed operation if $F_{CPUCLK} > 6MHz$. Before select high frequency clock (> 6MHz) 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. ($F_{CPUCLK} > 25MHz$). It also needs to set HSE when use HSE1 = 1.

10. Watch Dog Timer (WDT)

10.1. WDT Structure

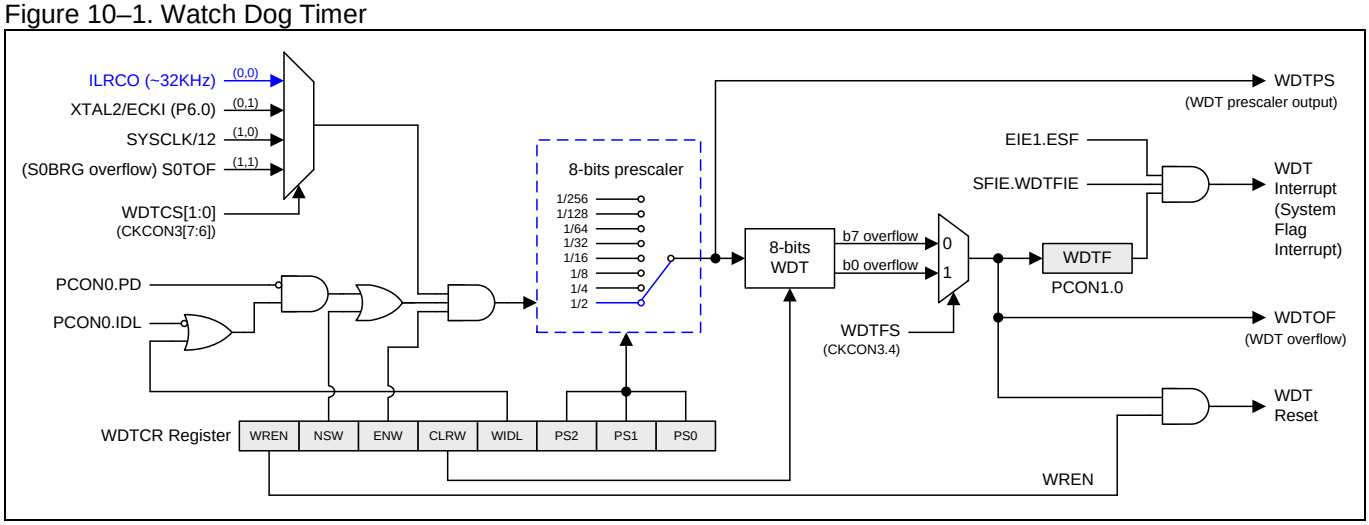
The Watch-dog Timer (WDT) is intended as a recovery method in situations where the CPU may be subjected to software upset. The WDT consists of an 8-bit free-running counter, a 8-bit prescaler and a control register (WDTCR). Figure 10–1 shows the WDT structure in **MG82F6P32**.

There are four selections for WDT clock source. The clock source must be configured before WDT enabled. The default WDT clock source is 32KHz 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 (WDTCR.7) is set. To prevent WDT overflow, software needs to clear it by writing “1” to the CLRW bit (WDTCR.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 WDTCR register will keep the previous programmed value unchanged after hardware (nRST-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 WDTCR can disable WREN, NSW and ENW, writing “0” on WDTCR.7~5. Please refer Section “10.4 WDT Register” and Section “33 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 (WDTCR.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.

10.3. WDT During Power Down (Auto Wake Up)

In the Power down mode, the ILRCO won’t stop if the NSW (WDTCR.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 or crystal oscillating circuit (XTAL1/XTAL2) enabled.

10.4. WDT Register

WDTCR: Watch-Dog-Timer Control Register

SFR Page = 0~F & P

SFR Address = 0xE1 POR = XXX0-XXXX (X: Depends on Hardware Options)

7	6	5	4	3	2	1	0
WREN	NSW	ENW	CLRW	WIDL	PS2	PS1	PS0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: WREN, WDT Reset Enable. The initial value can be changed by hardware option, WRENO.
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 can not 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 can not 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 = IHRCO, 12MHz)
0 0 0	2	16 ms	0.512 ms
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]	Prescaler Value	WDT Period (clock source = ILRCO)	
0 0 0	2	0.125 ms	245 us= 125+120
0 0 1	4	0.25 ms	370 us= 250+120
0 1 0	8	0.5 ms	620 us= 500+120
0 1 1	16	1 ms	1.12 ms= 1ms+120
1 0 0	32	2 ms	2.12 ms= 2ms+120
1 0 1	64	4 ms	4.12 ms= 4ms +120
1 1 0	128	8 ms	8.12 ms= 8ms +120
1 1 1	256	16 ms	16.12ms= 16ms+120

CKCON3: Clock Control Register 3

SFR Page = **P only**

SFR Address = 0x41 RESET = 0000-0010

7	6	5	4	3	2	1	0
WDTCS1	WDTCS0	FWKP	WDTFS	MCKD1	MCKD0	MCDS1	MCDS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

WDTCS1~0	WDT Clock Source
00	ILRCO
01	XTAL2/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 POR = 0000-0000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 POR = 0110-0000

7	6	5	4	3	2	1	0
SIDFIE	MCDRE	MCDFIE	RTCFIE	SPWIE	BOF1IE	BOF0IE	WDTFIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

10.5. 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, and (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 a 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 power down mode (watch mode).
☐: Disabled. Clear WDTCR.NSW to disable the WDT running in power down 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 WDTCR 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.

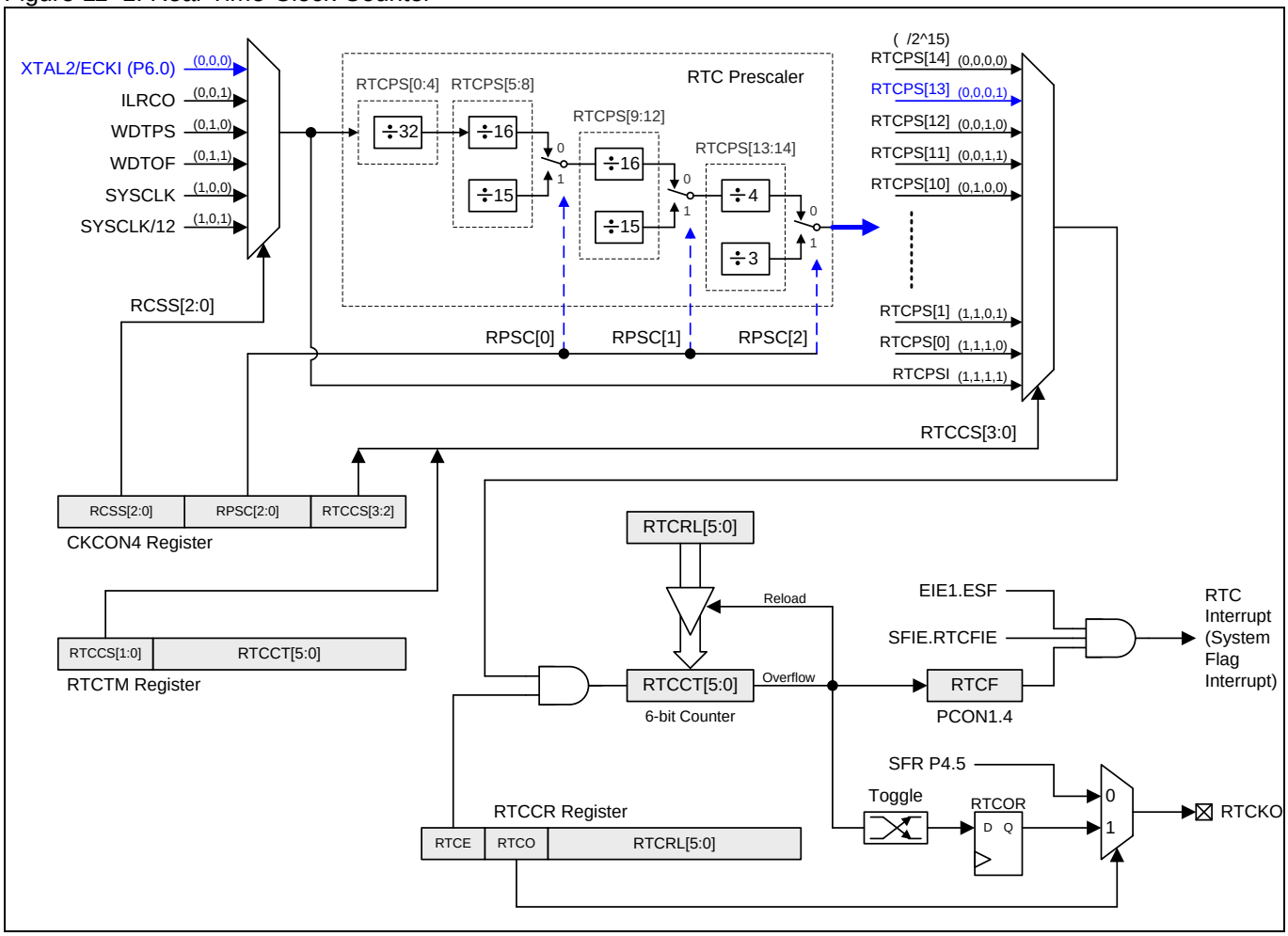
11. Real-Time-Clock (RTC)/System-Timer

The **MG82F6P32** 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 **MG82F6P32**. The settings of RTCCS[3:0], RTCCT[5:0], RTCRL[5:0] and RTCO should be set before enable the RTC function, when RTC has been enabled, the settings change will be ignored to prevent abnormal time period.

To input 32.768 KHz crystal 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 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 longer wake-up time requirement. The RCT clock source must be configured before RTCE enabled.

If the XTAL oscillator is used as the system clock, then the RTC still uses P6.0 input as its clock source. 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



11.1. RTC Register

RTCCR: Real-Time-Clock Control Register

SFR Page = **P**

SFR Address = **0x54** POR = 0011-1111

7	6	5	4	3	2	1	0
RTCE	RTCO	RTCRL.5	RTCRL.4	RTCRL.3	RTCRL.2	RTCRL.1	RTCRL.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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 = **P**

SFR Address = **0x55** POR = 0111-1111

7	6	5	4	3	2	1	0
RTCCS.1	RTCCS.0	RTCCT.5	RTCCT.4	RTCCT.3	RTCCT.2	RTCCT.1	RTCCT.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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

RTCCS.3~0	Clock Source	RTC Interrupt Duration if XTAL2/ECKI(P60) = 32768Hz	Min. Step
0 0 0 0	RTCPS[14] (/2^15)	1 sec ~ 64 sec	1 sec
0 0 0 1	RTCPS[13] (/2^14)	0.5 sec ~ 32 sec	0.5 sec (default)
0 0 1 0	RTCPS[13] (/2^13)	0.25 sec ~ 16 sec	0.25 sec
……	……	……	……
1 0 1 0	RTCPS[4] (/2^5)	976 us ~ 62.46 ms	976 us
1 0 1 1	RTCPS[3] (/2^4)	488 us ~ 31.23 ms	488 us
1 1 0 0	RTCPS[2] (/2^3)	244 us ~ 15.61 ms	244 us
1 1 0 1	RTCPS[1] (/2^2)	122 us ~ 7.805 ms	122 us
1 1 1 0	RTCPS[0] (/2^1)	61 us ~ 3.903 ms	61 us
1 1 1 1	RTCPSI (/2^0)	30.5 us ~ 1.952 ms	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 RESET = 0000-0000

7	6	5	4	3	2	1	0
RCSS2	RCSS1	RCSS0	RPSC2	RPSC1	RPSC0	RTCCS3	RTCCS2
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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

RCSS2, RCSS1, RCSS0	RTC Clock Selection
0 0 0	XTAL2/ECKI (P6.0)
0 0 1	ILRCO
0 1 0	WDTPS
0 1 1	WDTOF
1 0 0	SYSCLK
1 0 1	SYSCLK/12
1 1 0	Reserved
1 1 1	Reserved

PCON1: Power Control Register 1

SFR Page = 0~F & P

SFR Address = 0x97 POR = 0000-0000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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 POR = 0110-0000

7	6	5	4	3	2	1	0
SIDFIE	MCDRE	MCDFIE	RTCFIE	SPWIE	BOF1IE	BOF0IE	WDTFIE
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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

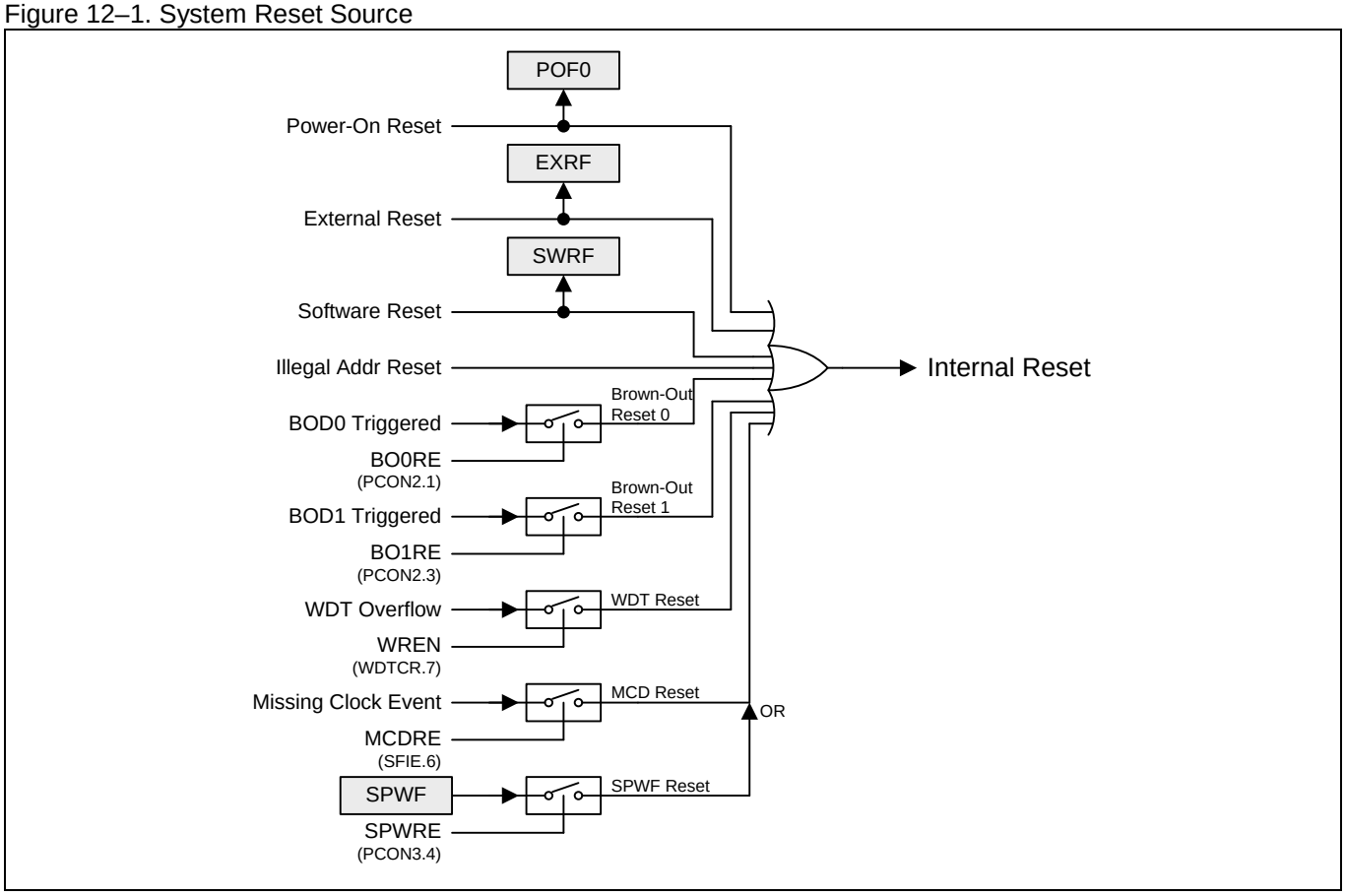
During reset, all I/O Registers are set to their initial values, and the program starts execution from the Reset Vector, 0000H, or ISP start address by OR setting. The **MG82F6P32** has **9** sources of reset: power-on reset, external reset, software reset, illegal address reset, brown-out reset 0, brown-out reset 1, WDT reset, and Missing-Clock-Detection reset. Figure 12–1 shows the system reset source in **MG82F6P32**.

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

12.1. Reset Source

Figure 12–1 presents the reset systems in the **MG82F6P32** and all its reset sources.

Figure 12–1. System Reset Source



12.2. Power-On Reset

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 to ensure a power-on reset

PCON0: Power Control Register 0

SFR Page = 0~F & P POR = 0001-0000
SFR Address = 0x87 RESET = 000X-0000

7	6	5	4	3	2	1	0
SMOD1	SMOD0	GF	POF0	GF1	GF0	PD	IDL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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.

12.3. External Reset

A reset is accomplished by holding the RESET pin LOW for a period of high time, please reference “37.3 External Reset Characteristics”. To ensure a reliable power-up reset, the hardware reset from nRST pin is necessary.

PCON1: Power Control Register 1

SFR Page = 0~F & P
SFR Address = 0x97 POR = 0000-0000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 POR = 0000-XXXX

7	6	5	4	3	2	1	0
ISPEN	SWBS	SWRST	CFAIL	--	--	--	--
R/W	R/W	R/W	R/W	W	W	W	W

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 POR = 0000-0000

7	6	5	4	3	2	1	0

SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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 **MG82F6P32**, 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, 3.6V, 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.

Notice of BOD1 Reset

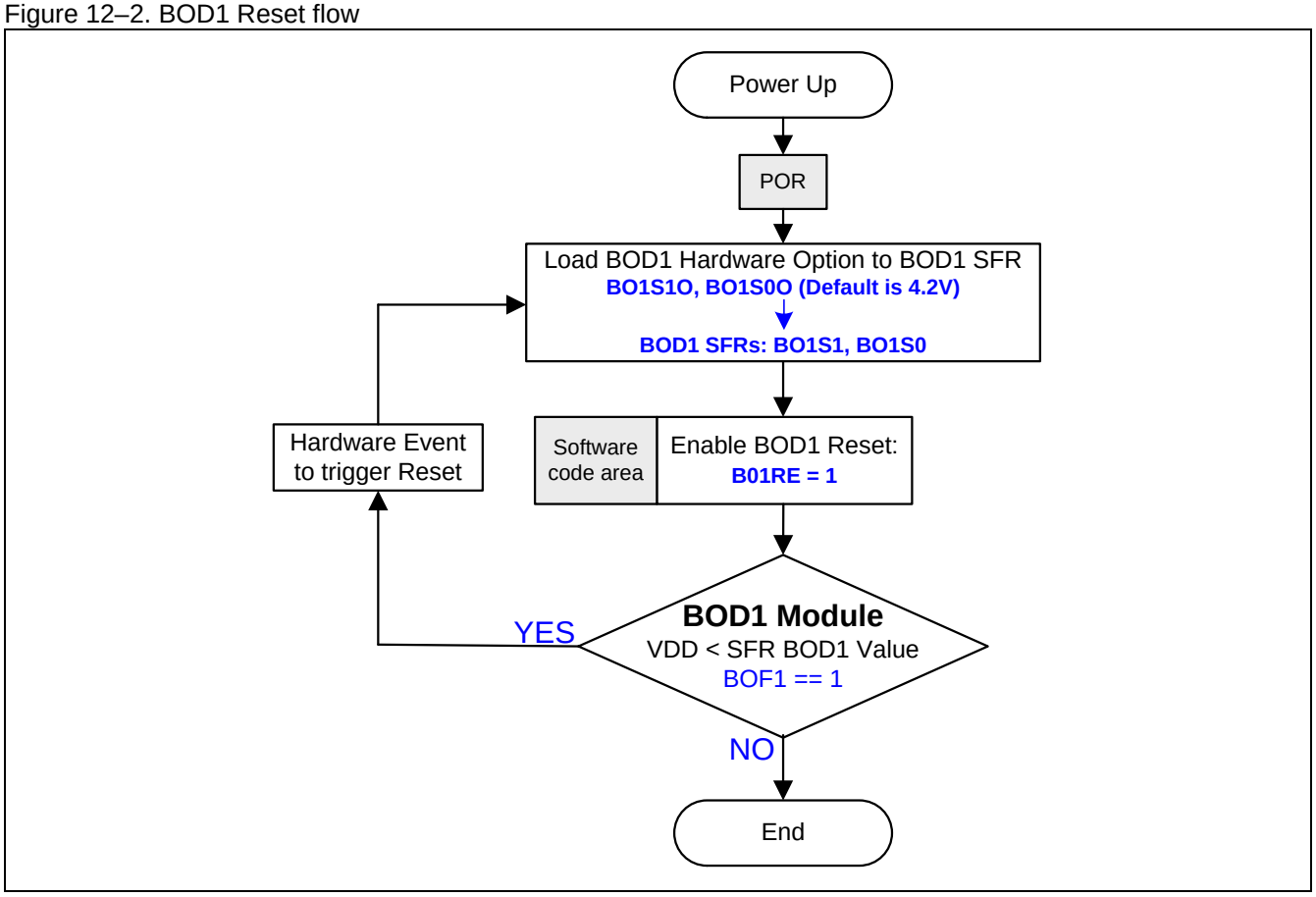
When BOD1 reset, it will reload the BOD1 Hardware Option value (BO1S1O, BO1S0O, please refer Section "35 Hardware Option" for detail) into BOD1 SFR (BO1S1, BO1S0). However, if the Hardware option BO1REO has also been enabled and user re-set a lower BOD1 level in software, when VDD lower than the software set level, the BOD1 event will be triggered. Normally the BOD1 event should be released after VDD voltage higher the software set value, but BO1REO is set in this case, so the BOD1 Hardware Option value will be reload, and induce the BOD1 be released until VDD higher than the Hardware Option value.

There are two ways to control BOD1 Reset:

1. Hardware Option:
   a. Using H/W option to set the BOD1 voltage detect level.
   b. To set H/W option BO1REO to enable BOD1 Reset. User also can reserve the BOD1 Reset and to enable it in the software code when needed.

Figure 12–2 shows the BOD1 reset flow.

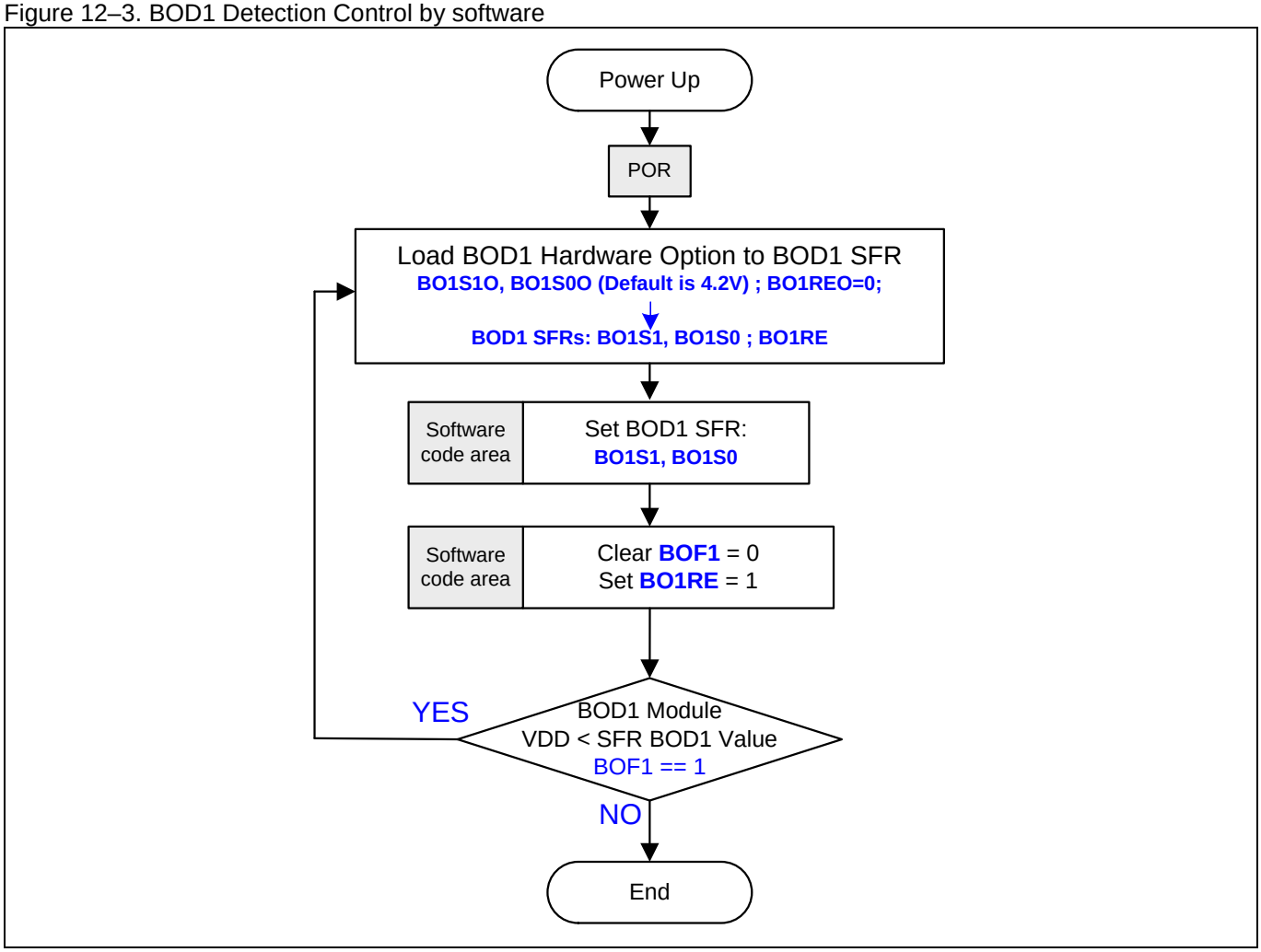
Figure 12–2. BOD1 Reset flow



2. Software control BOD1:
   a. Do not enable BOD1 Reset by Hardware Option BO1REO. (Default is disabled)
   b. Set BOD1 detect voltage by BO1S[1:0]
   c. Clear BOD1 flag, BOF1 = 0
   d. Enable BOD1 Reset

Figure 12–3 is the software control flow for BOD1.

Figure 12–3. BOD1 Detection Control by software



PCON1: Power Control Register 1

SFR Page = 0~F & P

SFR Address = 0x97 POR = 0000-0000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 POR = 0000-0000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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. MCD Reset

When XTAL is selected to MCU clock source by software, MCDF will be set and trigger a system reset by XTAL input signal lost. After MCD triggered system reset, the MCU clock source will be switched to IHRCO. In default, the MCD event will trigger a system reset. If user won't apply this function, clear MCDRE to disable the reset function.

PCON1: Power Control Register 1

SFR Page = 0~F & P

SFR Address = 0x97 POR = 0000-0000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5: MCDF, Missing Clock Detection 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 which detects a Missing-Clock event. Writing "1" on this bit will clear MCDF. The Missing-Clock-Detection module is enabled by MCDFIE. If MCDFIE is cleared, the Missing-Clock-Detection module is inactive. Once a missing clock event happened, software must clear MCDF before switching OSCin to XTAL.

SFIE: System Flag Interrupt Enable Register

SFR Page = 0~F

SFR Address = 0x8E POR = 0110-0000

7	6	5	4	3	2	1	0
SIDFIE	MCDRE	MCDFIE	RTCFIE	SPWIE	BOF1IE	BOF0IE	WDTFIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 6: MCDRE, Enable Missing-Clock-Detection event causes a system reset.
0: Disable MCD event to trigger a system Reset.
1: Enable MCD event to trigger a system Reset.

Bit 5: MCDFIE, Enable MCDF (PCON1.5) Interrupt.
0: Disable MCDF interrupt.
1: Enable MCD module and enable MCDF interrupt.

12.8. Illegal Address Reset

In **MG82F6P32**, if software program runs to illegal address such as over program ROM limitation, it triggers a RESET to CPU.

12.9. Stack Pointer Warning Reset

SPHB: Stack Pointer High Boundary

SFR Page = P Only
SFR Address = 0x53 RESET = 1111-1111

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

SPHB, it is used for the detection boundary of Stack Pointer warning.
If SPHB == 1111-1111, SPWF will be set when SP ≥ 1111-1111.
If SPHB == 1111-0000, SPWF will be set when SP ≥ 1111-0000.

PCON1: Power Control Register 1

SFR Page = 0~F & P
SFR Address = 0x97 POR = 0000-0000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 3: SPWF, SP Warning 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 SP ≥ SPHB. Writing **"1"** on this bit will clear SPWF when SP < SPHB.

PCON3: Power Control Register 3

SFR Page = **P Only**
SFR Address = 0x45 POR = 0000-0000

7	6	5	4	3	2	1	0
IVREN	IVRPDE	0	SPWRE	0	0	0	0
R/W	R/W	W	R/W	W	W	W	W

Bit 4: SPWRE, SPWF trigger a MCU reset.
0: Disable SPWF to trigger a MCU reset.
1: Enable SPWF to trigger a MCU reset.

13. Power Management

The **MG82F6P32** 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.

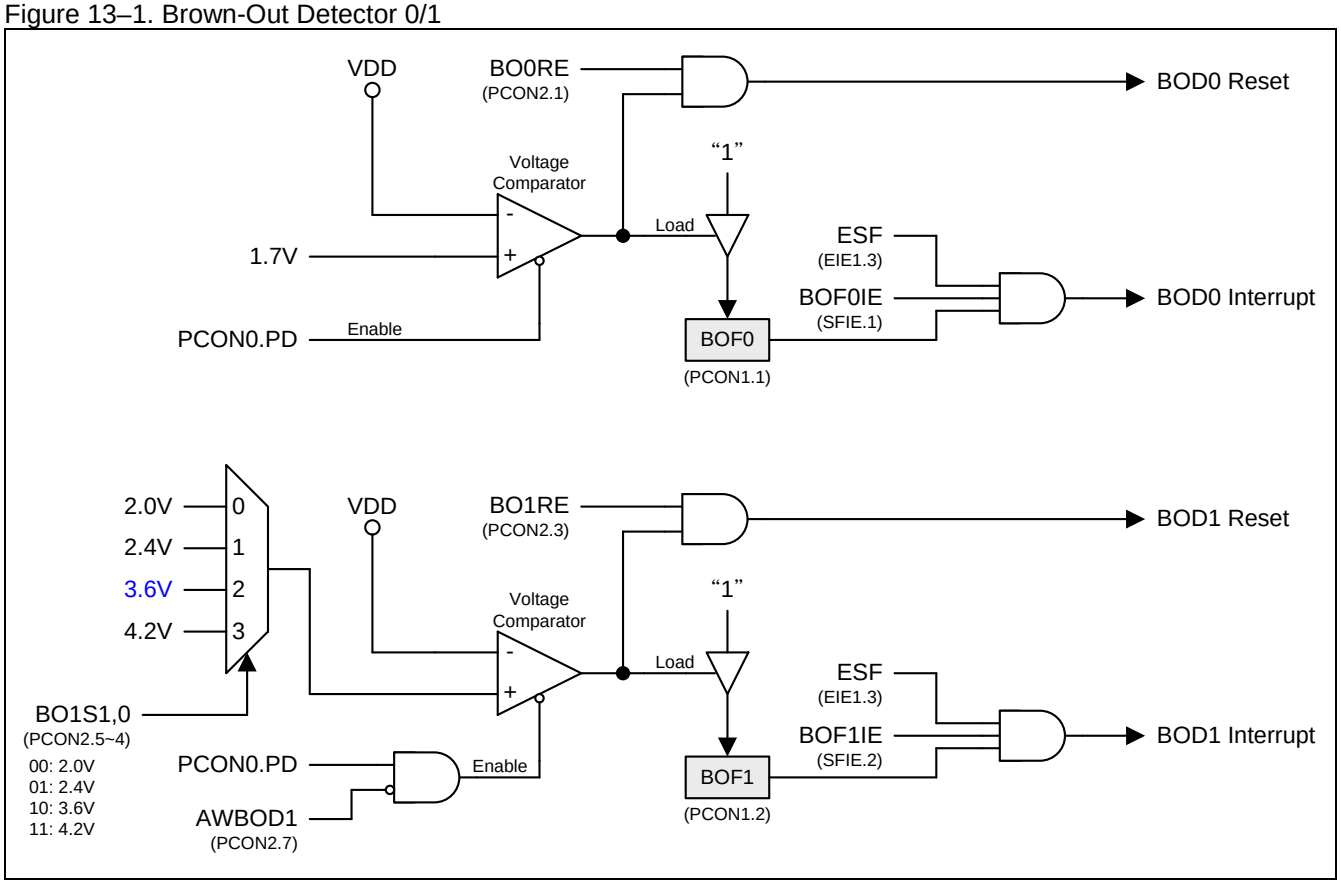
13.1. Brown-Out Detector

In **MG82F6P32**, 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 VDD and compare with the software selected levels (4.2V/3.6V/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 13–1. Brown-Out Detector 0/1



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 operates 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 **MG82F6P32** 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 **MG82F6P32**. 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 **MG82F6P32**. 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

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 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, AC0, AC1, OPA0, OPA1, PGA, S0~S1, TWI0/I2C0, TWI1/I2C1, 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.

The ADC, OPA0/1, PGA or analog comparator input channels must be set to "***Analog I/O***" when MCU is in idle mode or power-down mode to reduce power consumption.

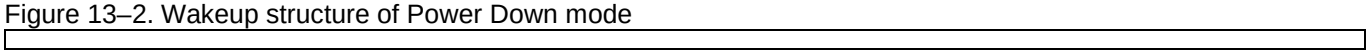
13.2.7. Power-down Mode

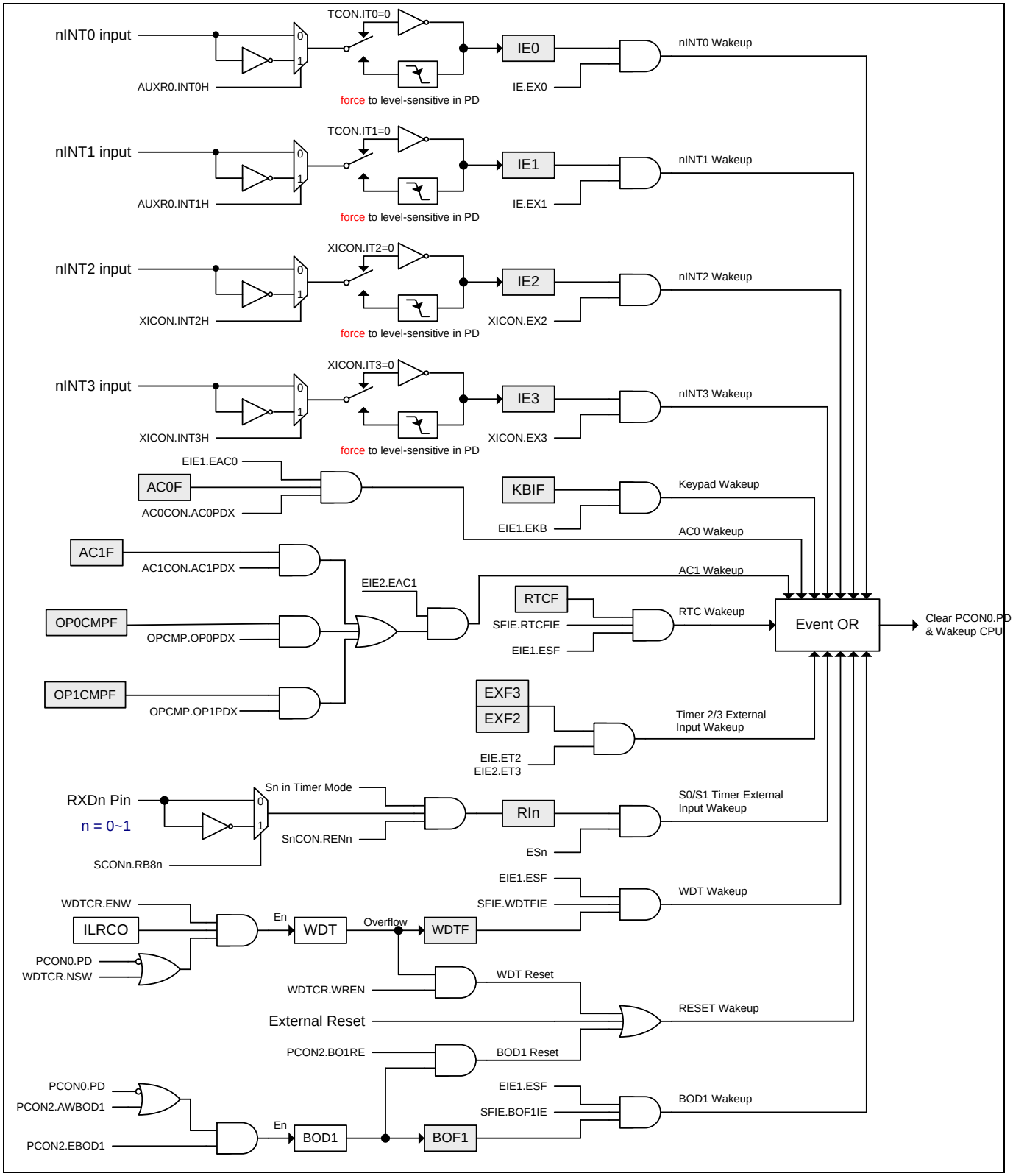
Setting the PD bit in PCON0 enters Power-down mode. Power-down mode stops the oscillator and powers down the Flash memory to minimize power consumption. Only the power-on circuitry will continue to draw power during Power-down. During Power-down the power supply voltage may be reduced to the RAM keep-alive voltage. The RAM contents will be retained; however, the SFR contents are not guaranteed once VDD has been reduced. Power-down may be excited by external reset, enabled external interrupts, AC0 ~ 1 power down enabled mode, OPA0~1 ACMP with power down enabled mode, 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 μs 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 shows the wakeup mechanism of power-down mode in **MG82F6P32**.

Figure 13–2. Wakeup structure of Power Down mode




nINT0 input
AUXR0.INT0H
TCON.IT0=0
force to level-sensitive in PD
IE0
IE.EX0
nINT0 Wakeup
nINT1 input
AUXR0.INT1H
TCON.IT1=0
force to level-sensitive in PD
IE1
IE.EX1
nINT1 Wakeup
nINT2 input
XICON.INT2H
XICON.IT2=0
force to level-sensitive in PD
IE2
XICON.EX2
nINT2 Wakeup
nINT3 input
XICON.INT3H
XICON.IT3=0
force to level-sensitive in PD
IE3
XICON.EX3
nINT3 Wakeup
EIE1.EAC0
AC0F
AC0CON.AC0PDX
KBIF
EIE1.EKB
Keypad Wakeup
AC0 Wakeup
AC1F
AC1CON.AC1PDX
AC1 Wakeup
EIE2.EAC1
OP0CMPF
OPCMP.OP0PDX
OP1CMPF
OPCMP.OP1PDX
RTCF
SFIE.RTCFIE
EIE1.ESF
RTC Wakeup
Event OR
Clear PCON0.PD
& Wakeup CPU
EXF3
EXF2
EIE.ET2
EIE2.ET3
Timer 2/3 External
Input Wakeup
RXDn Pin
n = 0~1
SCONn.RB8n
Sn in Timer Mode
SnCON.RENn
RIn
ESn
S0/S1 Timer External
Input Wakeup
WDTCR.ENW
ILRCO
PCON0.PD
WDTCR.NSW
En
WDT
Overflow
WDTF
EIE1.ESF
SFIE.WDTFIE
WDT Wakeup
WDTCR.WREN
WDT Reset
External Reset
RESET Wakeup
PCON2.BO1RE
BOD1 Reset
PCON0.PD
PCON2.AWBOD1
PCON2.EBOD1
En
BOD1
BOF1
EIE1.ESF
SFIE.BOF1IE
BOD1 Wakeup

13.2.8. Interrupt Recovery from Power-down

Four external interrupts may be configured to terminate Power-down mode. External interrupts nINT0, nINT1, nINT2 and nINT3 may be used to exit Power-down. To wake up by external interrupt nINT0, nINT1, nINT2 or nINT3, 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 falling edge of nRST, 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 nRST pin must be held low for longer than the timeout period to ensure that the device is reset properly. The device will begin executing once nRST is brought high.

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 **MG82F6P32**, 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”34 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 POR = 0001-0000
SFR Address = 0x87 RESET = 000X-0000

7	6	5	4	3	2	1	0
SMOD1	SMOD0	GF	POF0	GF1	GF0	PD	IDL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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: Set 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: Set this bit activates idle mode operation.

PCON1: Power Control Register 1

SFR Page = 0~F & P
SFR Address = 0x97 POR = 0000-0000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 5: MCDF, Missing Clock Detection 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 which detects a Missing-Clock event. Writing "1" on this bit will clear MCDF. The Missing-Clock-Detection module is enabled by MCDFIE. If MCDFIE is cleared, the Missing-Clock-Detection module is inactive. Once a missing clock event happened, software must clear MCDF before switching OSCin to XTAL.

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: SPWF, SP Warning 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 SP ≥ SPHB. Writing **"1"** on this bit will clear SPWF when SP < SPHB.

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 POR = 0000-0101

7	6	5	4	3	2	1	0
AWBOD1	0	BO1S1	BO1S0	BO1RE	EBOD1	BO0RE	
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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	3.6V
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 = 0000-0000

7	6	5	4	3	2	1	0
IVREN	IVRPDE	0	SPWRE	0	0	0	0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: IVREN, Internal Voltage Reference Enable.
0: Disable on-chip IVR (2.4V).
1: Enable on-chip IVR (2.4V).

Bit 6 IVRPDE, IVR can be used under power down.
0: Disable IVR keep awake at Power down mode.
1: Enable IVR keep awake at Power down mode

Bit 5: Reserved. Software must write “0” on these bits when PCON3 is written.

Bit 4: SPWRE, SPWF trigger a MCU reset.
0: Disable SPWF to trigger a MCU reset.
1: Enable SPWF to trigger a MCU reset.

Bit 3~0: Reserved. Software must write “0” on these bits when PCON3 is written.

14. Configurable I/O Ports (GPIO)

The **MG82F6P32** has following I/O ports: P0.1, P0.2, P0.5, P0.7, P1.0~P1.7, P2.2~P2.4, P2.6, P3.0~P3.5, P4.0~P4.1, P4.4, P4.5, P4.7 and P6.0~P6.1. If enable external crystal oscillator as system clock or RTC input, Port 6.0 and Port 6.1 are configured to XTAL2 and XTAL1. 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.

Table 14–1. Number of I/O Pins Available

Package Type	I/O Pins	Number of I/O ports
32-pin	P0.1, P0.2, P0.5, P0.7, P1.0~P1.7, P2.2~P2.4, P2.6, P3.0~P3.5, P4.0~P4.1, P4.4~P4.5, P4.7(nRST), P6.0(ECKI/XTAL2), P6.1(XTAL1)	29 or 28 (nRST selected) or 27 (nRST & ECKI selected) or 26 (nRST & XTAL selected)
28-pin	P0.1, P0.2, P1.0~P1.7, P2.2, P2.4, P2.6, P3.0~P3.1, P3.3~P3.5, P4.0~P4.1, P4.4~P4.5, P4.7(nRST), P6.0(ECKI/XTAL2), P6.1(XTAL1)	25 or 24 (nRST selected) or 23 (nRST & ECKI selected) or 22 (nRST & XTAL selected)

14.1. IO Structure

The I/O operating modes are distinguished two groups in **MG82F6P32**. The first group is Port 3 only features, which are quasi-bidirectional (standard 8051 I/O port), push-pull output, input-only (high-impedance input) and open-drain output. But the default mode of Port 3 and most of the rest I/O port is analog I/O mode expect OCD interface (P4.4/P4.5) and the reset pin. The default mode of the OCD interface is Open-Drain with pull-up resistor.

All other general port pins belong to the second group. They can be programmed to four operating modes, which include analog I/O, 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/output, which means the port pin in high impedance state.

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

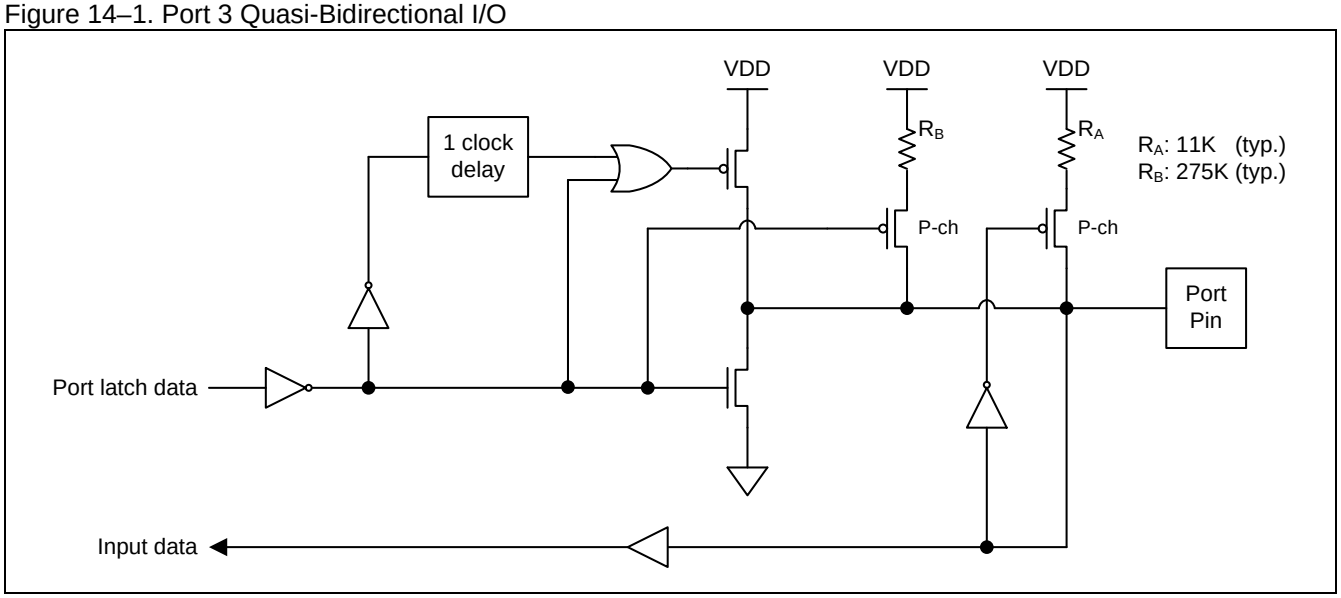
14.1.1. Port 3 Quasi-Bidirectional IO Structure

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.

The quasi-bidirectional port configuration is shown in Figure 14–1.

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

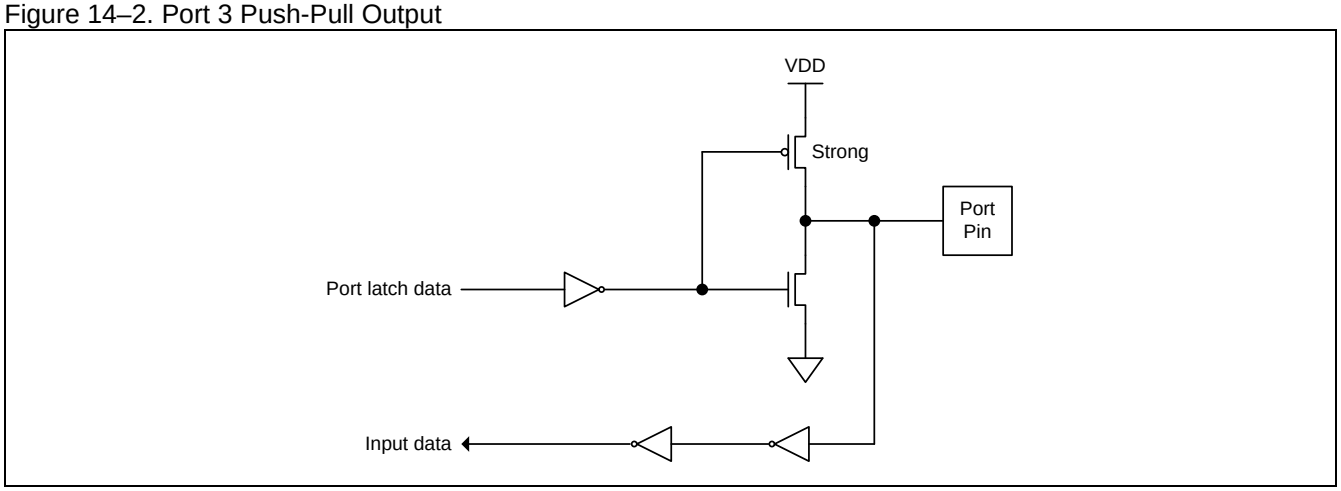


14.1.2. Port 3 Push-Pull Output Structure

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.

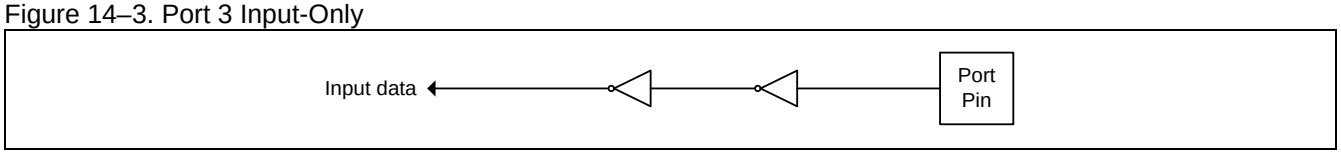
Figure 14–2. Port 3 Push-Pull Output



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

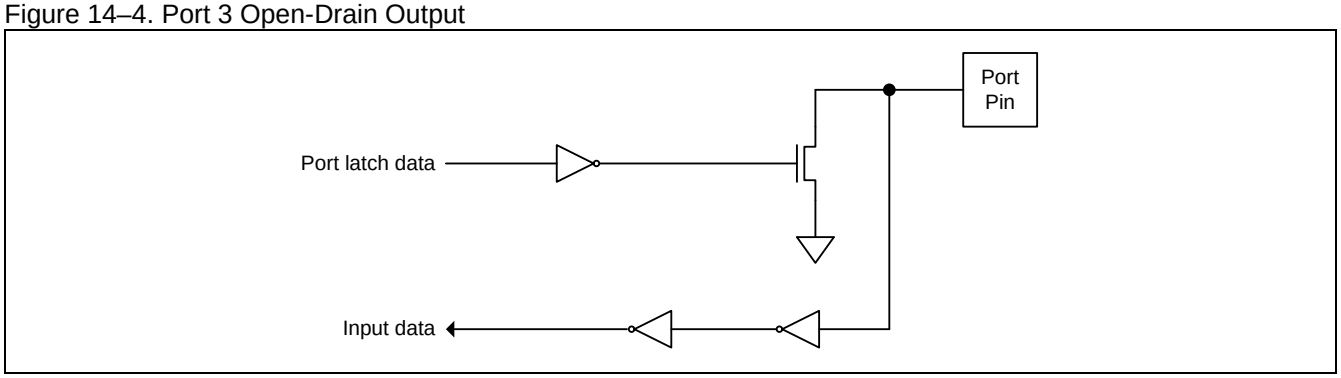


14.1.4. Port 3 Open-Drain Output Structure

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

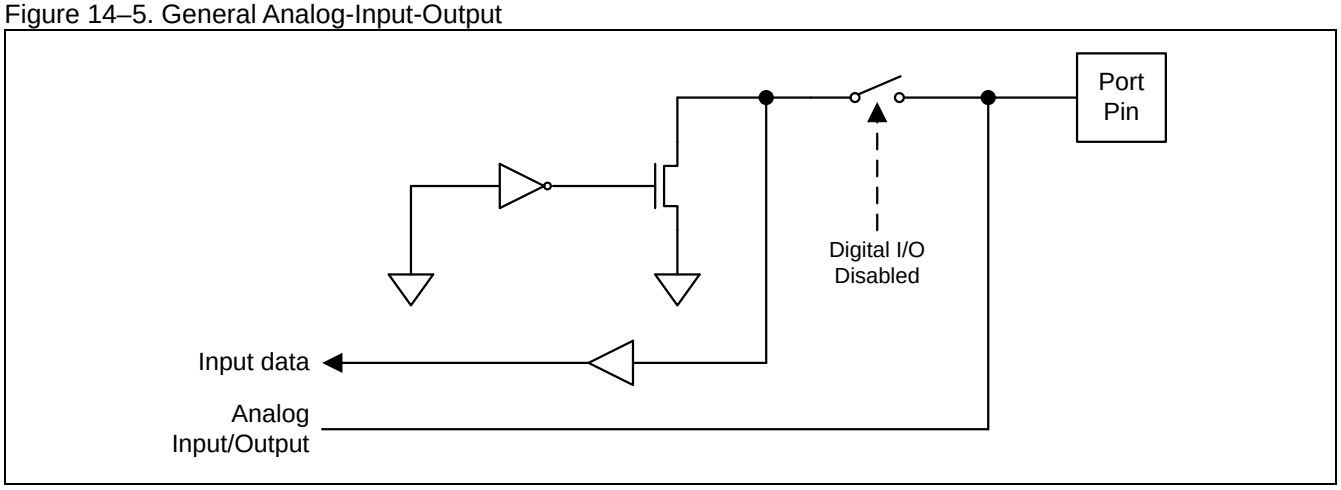


14.1.5. General Analog Input Output Structure

The analog-input-output configuration on general port pins is the default setting, which are including Port 0, 1, 2, 3, 4 and 6, but not include OCD_ICE (P4.4 and P4.5) interface and nRST. For ADC, Analog Comparator, OPA and PGA input/output 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. Under analog I/O mode to read the I/O SFR it will be always 0, due to the digital input path is disabled and it will be tied to low to prevent internal floating state.

The analog-input-output port configuration is shown in Figure 14–5.

Figure 14–5. General Analog-Input-Output

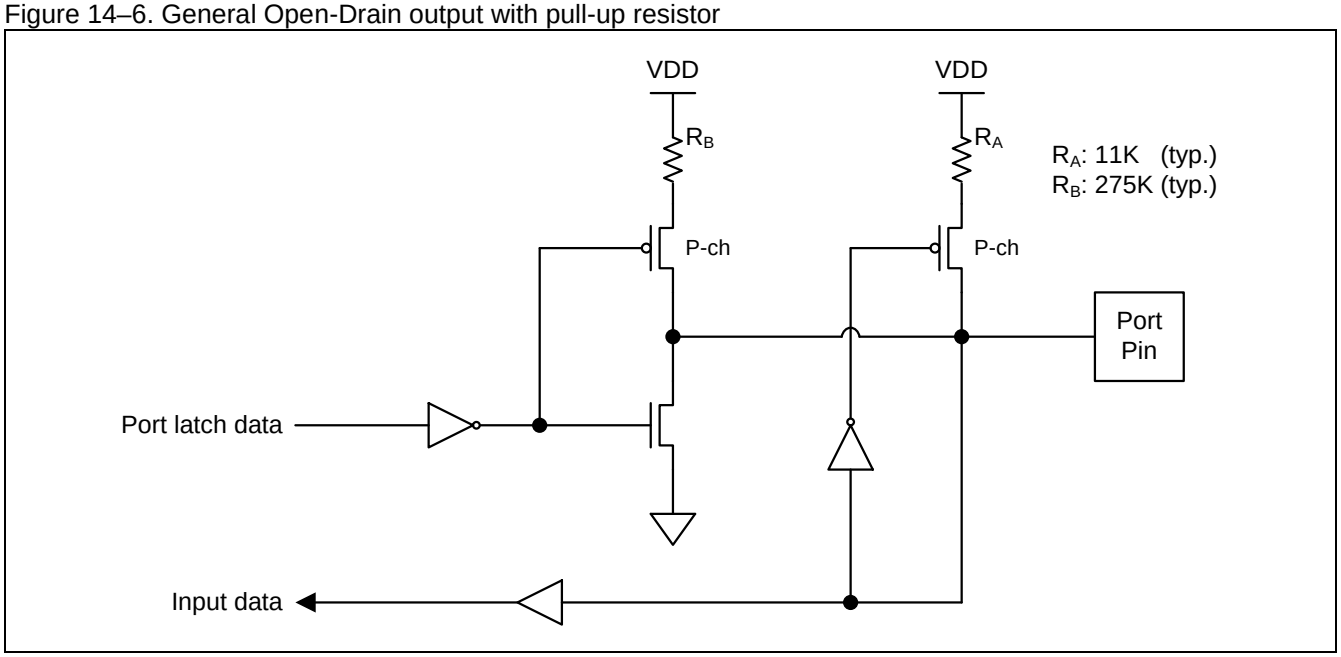


14.1.6. General Open-Drain Output with Pull-up Resistor Structure

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

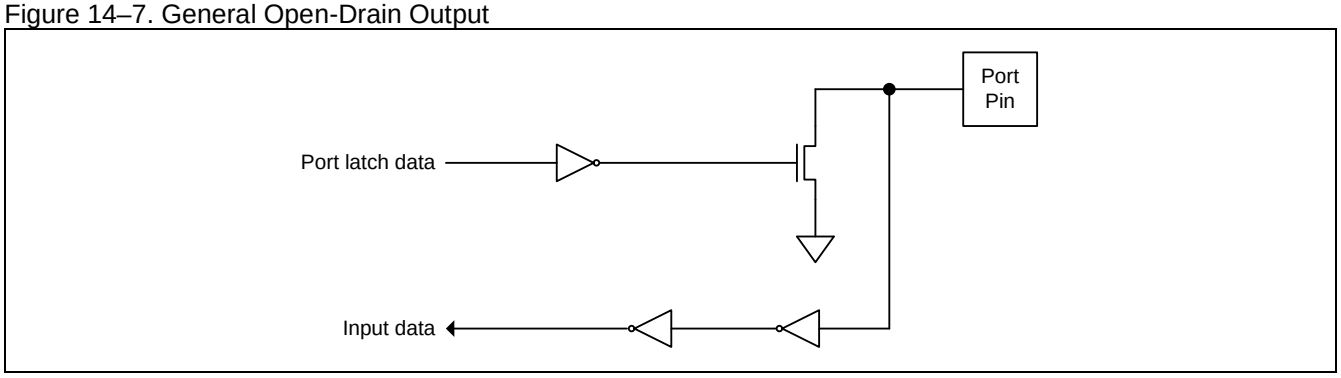


14.1.7. General Open-Drain Output Structure

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

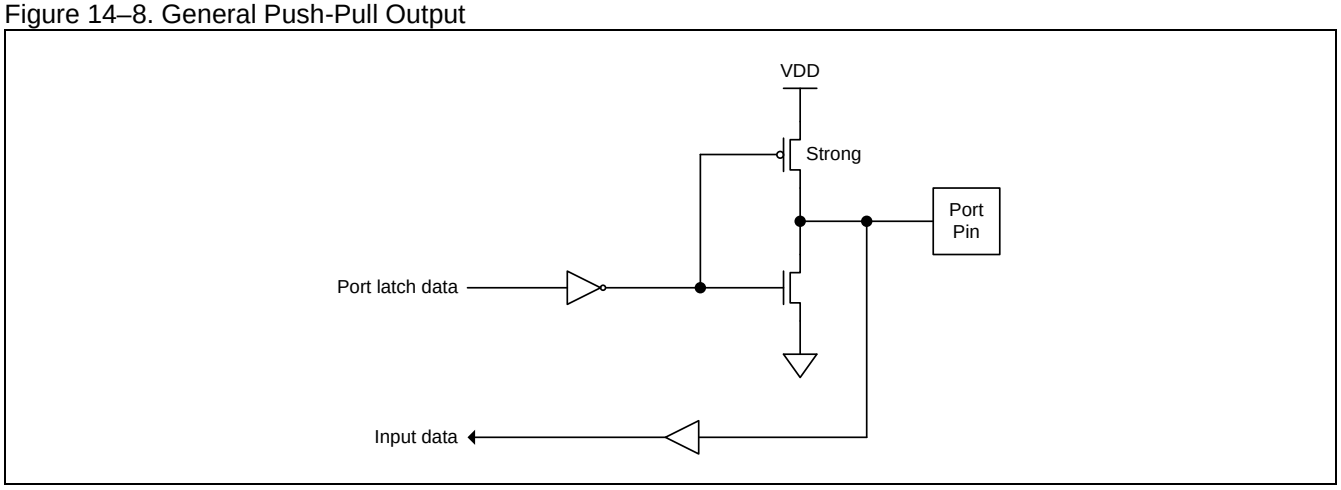
A Port pin is configured as a digital input by setting its output mode to “Open-Drain” and writing a logic “1” to the associated bit in the Port Data register. For example, P1.0 is configured as a digital input by setting P1M0.0 to a logic 0, P1M1.0 to a logic 0 and P1.0 to a logic 1.

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

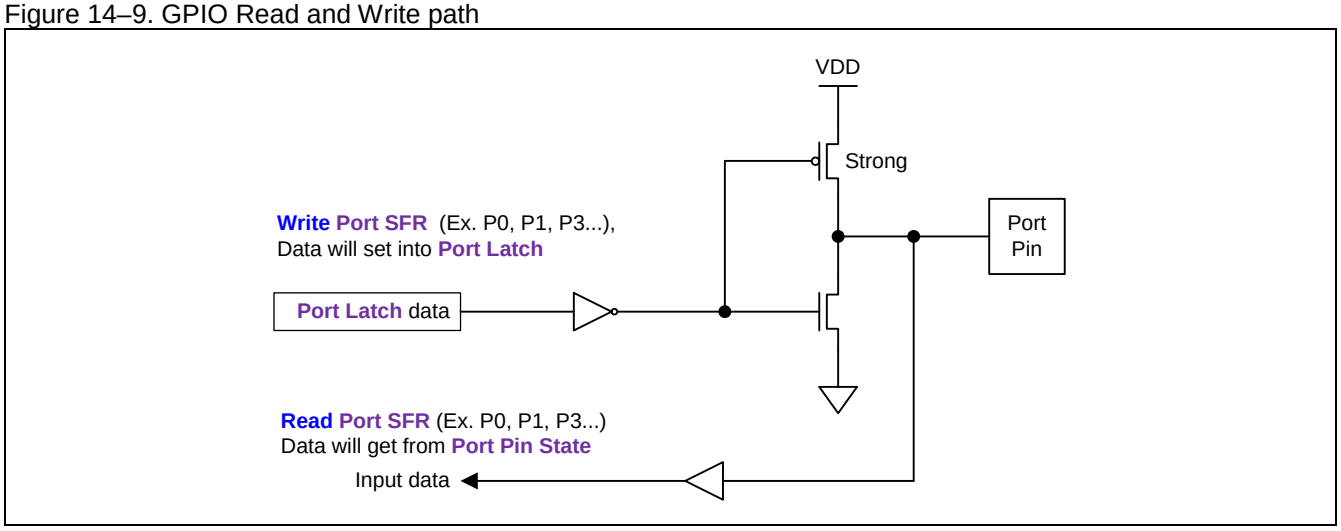


14.1.10. GPIO Read and Write -- Port Latch and Port state

When read or write I/O port SFR it will go through different path, please reference Figure 14–9.
Write data to Port SFR (For example, P0, P1…) it will be set into Port Latch.
Read data from Port State, it means when the voltage on the port pin is higher than VIH, the read back data will be "1", and if the voltage one the port pin is lower than VIL, the read back data will be "0". But the read back data will not affect the Port Latch value.

For example, if the I/O Port mode is open drain, and write "1" to port latch, but the port pin voltage is pulled down to "0". Then the read back value will be "0", but the port latch is maintaining "1".

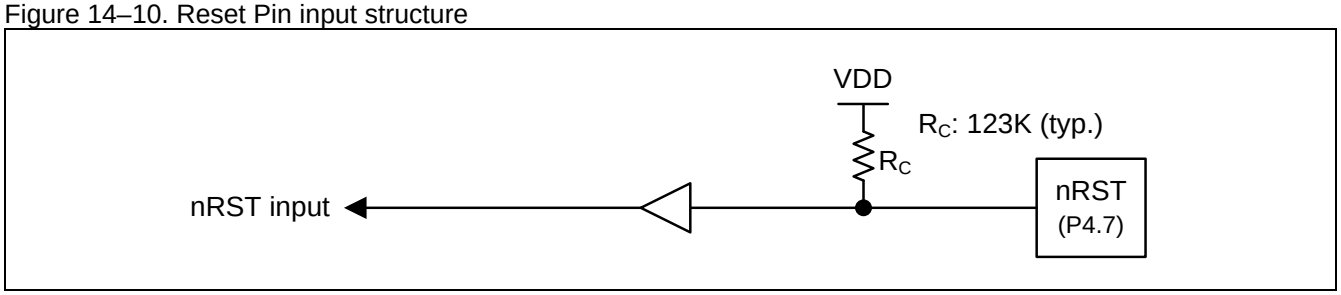
Figure 14–9. GPIO Read and Write path



14.1.11. Reset Pin Structure

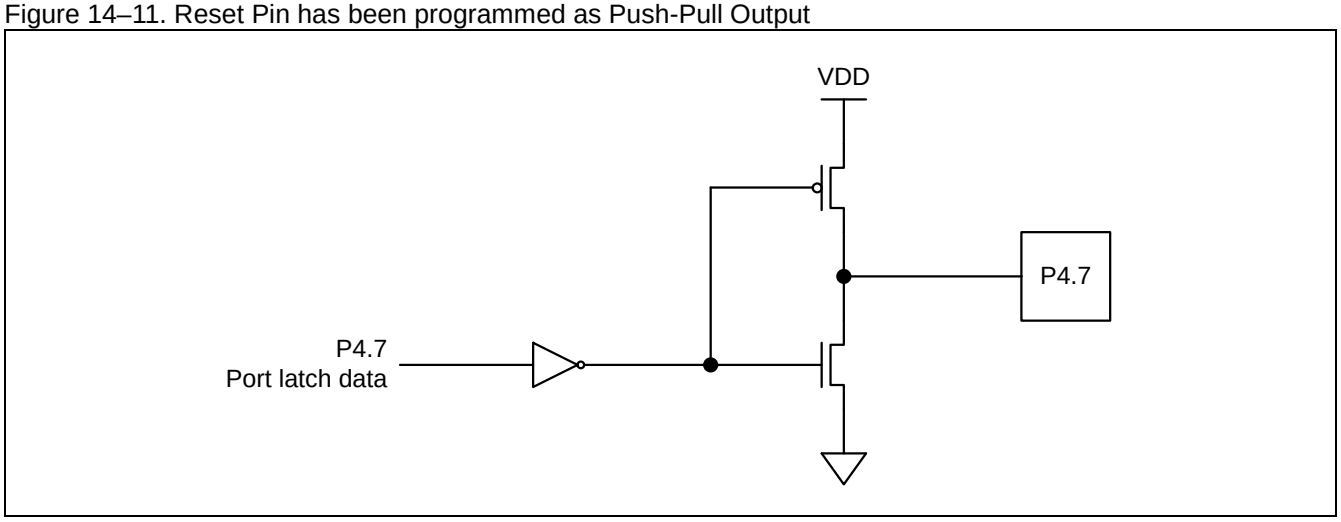
The Reset Pin is used for the external reset trigger circuit. Its structure is showed as in Figure 14–10.

Figure 14–10. Reset Pin input structure



The Reset Pin also can be set as output pin, its structure is showed in Figure 14–11

Figure 14–11. Reset Pin has been programmed as Push-Pull Output



14.1.12. Port Pin Output Driving Strength Selection

The I/O of the **MG82F6P32** has two driving strength can be selected for different kinds of the application to match the output impedance. Please reference 14.2.7 Port Output Driving Strength Control Register.

14.1.13. Port Pin Output Fast Driving Selection

The I/O of the **MG82F6P32** has two driving speed can be selected for different kinds of the I/O frequency. Please reference 14.2.8 Port Output Fast Driving Control Register.

14.2. I/O Port Register

All I/O port pins on the **MG82F6P32** may be individually and independently configured by software to select its operating modes. Port 3 has five operating modes, as shown in Table 14–2. Two mode registers select the output type for each port 3 pin.

Table 14–2. Port 3 Configuration Settings

P3M2.y	P3M1.y	P3M0.y	Port Mode
0	0	0	Quasi-Bidirectional
0	0	1	Input Only (High Impedance Input)
0	1	0	Push-Pull Output
0	1	1	Open-Drain Output
1	0	0	**Analog I/O(default)**
1	others		Reserved

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.

Table 14–3. General Port Configuration Settings

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

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 0 Register

P0: Port 0 Register

SFR Page = 0~F
SFR Address = 0x80 RESET = 1010-0110

7	6	5	4	3	2	1	0
P0.7	0	P0.5	0	0	P0.2	P0.1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

P0M0: Port 0 Mode Register 0

SFR Page = **0 only**
SFR Address = 0x93 RESET = 0000-0000

7	6	5	4	3	2	1	0
P0M0.7	0	P0M0.5	0	0	P0M0.2	P0M0.1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P0M1: Port 0 Mode Register 1

SFR Page = **4 only**
SFR Address = **0x92** RESET = 1111-1111

7	6	5	4	3	2	1	0
P0M1.7	1	P0M1.5	1	1	P0M1.2	P0M1.1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

14.2.2. Port 1 Register

P1: Port 1 Register

SFR Page = 0~F
SFR Address = 0x90 RESET = 1111-1111

7	6	5	4	3	2	1	0
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: 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 RESET = 0000-0000

7	6	5	4	3	2	1	0
P1M0.7	P1M0.6	P1M0.5	P1M0.4	P1M0.3	P1M0.2	P1M0.1	P1M0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P1M1: Port 1 Mode Register 1

SFR Page = **0 only**
SFR Address = **0x92** RESET = 1111-1111

7	6	5	4	3	2	1	0
P1M1.7	P1M1.6	P1M1.5	P1M1.4	P1M1.3	P1M1.2	P1M1.1	P1M1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

14.2.3. Port 2 Register

P2: Port 2 Register

SFR Page = 0~F
SFR Address = 0xA0 RESET = 0101-1100

7	6	5	4	3	2	1	0
0	P2.6	0	P2.4	P2.3	P2.2	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: 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 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	P2M0.6	0	P2M0.4	P2M0.3	P2M0.2	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P2M1: Port 2 Mode Register 1

SFR Page = 1 only
SFR Address = 0x92 RESET = 1111-1111

7	6	5	4	3	2	1	0
1	P2M1.6	1	P2M1.4	P2M1.3	P2M1.2	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

14.2.4. Port 3 Register

P3: Port 3 Register

SFR Page = 0~F
SFR Address = 0xB0 RESET = 0011-1111

7	6	5	4	3	2	1	0
0	0	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: Port 3 output data latch could be only set/cleared by CPU.

P3M0: Port 3 Mode Register 0

SFR Page = 0~F
SFR Address = 0xB1 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	P3M0.5	P3M0.4	P3M0.3	P3M0.2	P3M0.1	P3M0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P3M1: Port 3 Mode Register 1

SFR Page = 0~F
SFR Address = 0xB2 RESET = 0000-0000

7	6	5	4	3	2	1	0
1	1	P3M1.5	P3M1.4	P3M1.3	P3M1.2	P3M1.1	P3M1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P3M2: Port 3 Mode Register 2

SFR Page = F
SFR Address = 0x92 RESET = 1111-1111

7	6	5	4	3	2	1	0
1	1	P3M2.5	P3M2.4	P3M2.3	P3M2.2	P3M2.1	P3M2.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

14.2.5. Port 4 Register

P4: Port 4 Register

SFR Page = 0~F
SFR Address = 0xE8 RESET = 1011-0011

7	6	5	4	3	2	1	0
P4.7	0	P4.5	P4.4	0	0	P4.1	P4.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 nRST input.

P4M0: Port 4 Mode Register 0

SFR Page = 0 only
SFR Address = 0xB3 RESET = 1011-0000

7	6	5	4	3	2	1	0
P4M0.7	0	P4M0.5	P4M0.4	0	0	P4M0.1	P4M0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: When P4.7/ nRST use as port pin, it is not suggested to program it as Input to avoid MCU is locked in reset in boot-up period when level high send into this pin.

P4M1: Port 4 Mode Register 1

SFR Page = **2 only**

SFR Address = **0x92** RESET = 1111-1111

7	6	5	4	3	2	1	0
P4M1.7	1	P4M1.5	P4M1.4	1	1	P4M1.1	P4M1.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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

14.2.6. Port 6 Register

P6: Port 6 Register

SFR Page = **1 only**

SFR Address = **0xF8** RESET = 0000-0011

7	6	5	4	3	2	1	0
0	0	0	0	0	0	P6.1	P6.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~0: 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** RESET = 0000-0000

7	6	5	4	3	2	1	0
7	6	5	4	3	2	P6M0.1	P6M0.0
0	0	0	0	0	0	**R/W**	**R/W**

P6M1 : Port 6 Mode Register 1

SFR Page = **3 only**

SFR Address = **0x92** RESET = 1111-1111

7	6	5	4	3	2	1	0
1	1	1	1	1	1	P6M1.1	P6M1.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

14.2.7. Port Output Driving Strength Control Register

In **MG82F6P32,** 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** RESET = 0000-0000

7	6	5	4	3	2	1	0
P3DC1	P3DC0	P2DC1	P2DC0	P1DC1	P1DC0	P0DC1	P0DC0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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: P0DC1, Port 0 output driving strength control on high nibble.
0: Select the P0.7 ~ P0.4 output with high driving strength.
1: Select the P0.7 ~ P0.4 output with low driving strength.

Bit 0: P0DC0, Port 0 output driving strength control on low nibble.
0: Select the P0.3 ~ P0.0 output with high driving strength.
1: Select the P0.3 ~ P0.0 output with low driving strength.

PDRVC1: Port Drive Control Register 1

SFR Page = **3 only**
SFR Address = **0xB4** RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	0	0	0	0	P4DC1	P4DC0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7 ~ 2: Reserved. Software must write “0” on this bit 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: P4DC0, Port 4 output driving strength control on low nibble.
0: Select the P4.3 ~ P4.0 output with high driving strength.
1: Select the P4.3 ~ P4.0 output with low driving strength.

14.2.8. Port Output Fast Driving Control Register

In **MG82F6P32,** all port pins have two driving speed selection by software configured except P6.1. 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** RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	P3FDC.5	P3FDC.4	P3FDC.3	P3FDC.2	P3FDC.1	P3FDC.0
R/W	R/W	R/W	R/W	R/W	R/W	RW	RW

Bit 5~0: Port 3 output fast driving control
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** RESET = 0000-0000

7	6	5	4	3	2	1	0
P1FDC.7	P1FDC.6	P1FDC.5	P1FDC.4	P1FDC.3	P1FDC.2	P1FDC.1	P1FDC.0
R/W	R/W	R/W	R/W	R/W	R/W	RW	RW

Bit 7~0: Port 1 output fast driving control.
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** RESET = 0000-0000

7	6	5	4	3	2	1	0
0	P2FDC.6	0	P2FDC.4	P2FDC.3	P2FDC.2	0	0
R/W	R/W	R/W	R/W	R/W	R/W	RW	RW

Bit 7~0: Port 2 output fast driving control
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** RESET = 0000-0000

7	6	5	4	3	2	1	0
P4FDC.7	0	P4FDC.5	P4FDC.4	0	0	P4FDC.1	P4FDC.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: Port 4 output fast driving control
0: Disable fast driving on port pin output.
1: Enable fast driving on port pin output.

P0FDC: Port 0 Fast Driving Control Register

SFR Page = B **only**
SFR Address = **0x92** RESET = 0000-0000

7	6	5	4	3	2	1	0
P0FDC.7	0	P0FDC.5	0	0	P0FDC.2	P0FDC.1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: Port 0 output fast driving control.
0: Disable fast driving on port pin output.
1: Enable fast driving on port pin output.

AUXR10: Auxiliary Register 10

SFR Page = 7 only

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	SPIPS1	SPIPS0	S0PS1	P60OC1	P60OC0	P60FD
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 0: 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.

15. Interrupt

The **MG82F6P32** has **19** 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.

Figure 15–1 shows the interrupt system. Each of these interrupts will be briefly described in the following sections.

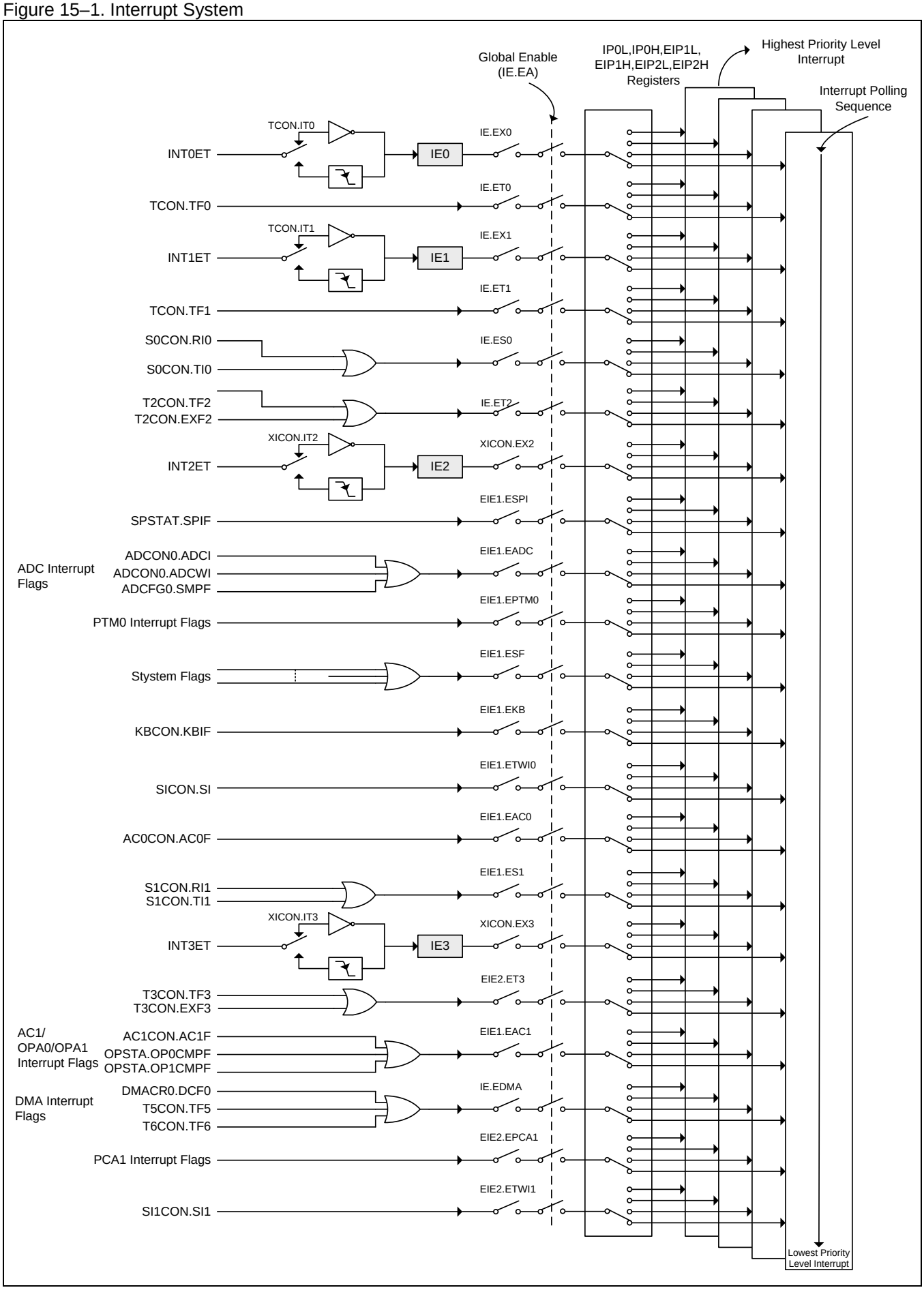
Table 15–1. Interrupt Sources

No	Source Name	Enable Bit	Request Bits	Priority Bits	Polling Priority	Vector Address
#0	External Interrupt 0, nINT0	EX0	IE0	[PX0H, PX0L]	(Highest)	0003H
#1	Timer 0	ET0	TF0	[PT0H, PT0L]	…	000Bh
#2	External Interrupt 1, nINT1	EX1	IE1	[PX1H, PX1L]	...	0013H
#3	Timer 1	ET1	TF1	[PT1H, PT1L]	**…**	001BH
#4	Serial Port 0	ES0	RI0, TI0	[PSH, PSL]	**…**	0023H
#5	Timer 2	ET2	TF2, EXF2 (TF2L)	[PT2H, PT2L]	**…**	002Bh
#6	External Interrupt 2, nINT2	EX2	IE2	[PX2H, PX2L]	**…**	0033H
#7	SPI	ESPI	SPIF	[PSPIH, PSPIL]	**…**	003BH
#8	ADC	EADC	ADCI, ADCWI, SMPF	[PADCH, PADCL]	**…**	0043H
#9	PTM0	EPTM0	CF, CCFn (n=0~ 5)	[PPTMH, PPTML]	…	004Bh
#10	System Flag	ESF	(Note 1)	[PSFH, PSFL]	**…**	0053H
#11	Keypad Interrupt	EKB	KBIF	[PKBH, PKBL]	**…**	005BH
#12	TWI0/I2C0	ETWI0	SI	[PTWI0H, PTWI0L]	**…**	0063H
#13	Analog Comparator 0	EAC0	AC0F	[PAC0H, PAC0L]	**…**	006BH
#14	Serial Port 1	ES1	RI1, TI1	[PS1H, PS1L]	…	0073H
#15	External Interrupt 3, nINT3	EX3	IE3	[PX3H, PX3L]	…	007BH
#16	Timer 3	ET3	TF3, EXF3 (TF3L)	[PT3H, PT3L]	**…**	0083H
#17	Analog Comparator 1/ OPA0/OPA1	EAC1	AC1F, OP0CMPF, OP1CMPF	[PAC1H, PAC1L]	**…**	008BH
#18	DMA	EDMA	(Note 2)	[PDMAH, PDMAL]	…	0093H
#19	PCA1	EPCA1	C1F, C1CFn (n=0~1)	[PAC2H, PAC2L]	…	009BH
#20	Reserved					00A3H
#21	Reserved					00ABH
#22	Reserved					00B3H
#23	TWI1	ETWI1	SI1	[PTWI1H, PTWI1L]	(Lowest)	00BBH

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

Note 2: The DMA interrupt flags include: DCF0, TF5 and TF6.

Figure 15–1. Interrupt System



15.2. Interrupt Source

Table 15–2. Interrupt Source Flag

No	Source Name	Request Bits	Bit Location
#0	External Interrupt 0, nINT0	IE0	TCON.1
#1	Timer 0	TF0	TCON.5
#2	External Interrupt 1, nINT1	IE1	TCON.3
#3	Timer 1	TF1	TCON.7
#4	Serial Port 0	RI0, TI0	S0CON.0 S0CON.1
#5	Timer 2	TF2, EXF2, (TF2L)	T2CON.7 T2CON.6 T2CON.5
#6	External Interrupt 2, nINT2	IE2	XICON.1
#7	SPI	SPIF	SPSTAT.7
#8	ADC	ADCI, ADCWI, SMPF	ADCON0.4 ADCON0.6 ADCFG0.2
#9	PTM0	CF, CCFn (n=0~5)	CCON.7 CCON.5~0
#10	System Flag	WDTF, BOF0, BOF1, SPWF, RTCF, MCDF, STAF, STOF, (TI0)	PCON1.0 PCON1.1 PCON1.2 PCON1.3 PCON1.4 PCON1.5 AUXR2.7 AUXR2.6 S0CON.1
#11	Keypad Interrupt	KBIF	KBCON.0
#12	TWI0/I2C0	SI	SICON.3
#13	Analog Comparator 0	AC0F	AC0CON.4
#14	Serial Port 1	RI1, TI1	S1CON.0 S1CON.1
#15	External Interrupt 3, nINT3	IE3	XICON.5
#16	Timer 3	TF3, EXF3, (TF3L)	T3CON.7 T3CON.6 T3CON.5
#17	Analog Comparator 1 OPA0/OPA1	AC1F OP0CMPF OP1CMPF	AC1CON.4 OPSTA.0 OPSTA.1
#18	DMA	DCF0 TF5 TF6	DMACR0.0 T5CON.7 T6CON.7
#19	PCA1	C1F, C1CFn	C1CON.7 C1CON.1~0
#20	Reserved		
#21	Reserved		
#22	Reserved		
#23	TWI1/I2C1	SI1	SI1CON.3

The external interrupt nINT0, nINT1, nINT2 and nINT3 can each be either level-activated or transition-activated, depending on bits IT0 and IT1 in register TCON, IT2 and IT3 in register XICON. The flags that actually generate these interrupts are bits IE0 and IE1 in TCON, IE2 and IE3 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 DMA interrupt is generated by the logical OR of DCF0, TF5 and TF6. All flags will not be cleared by hardware when the service routine is vectored to.

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.

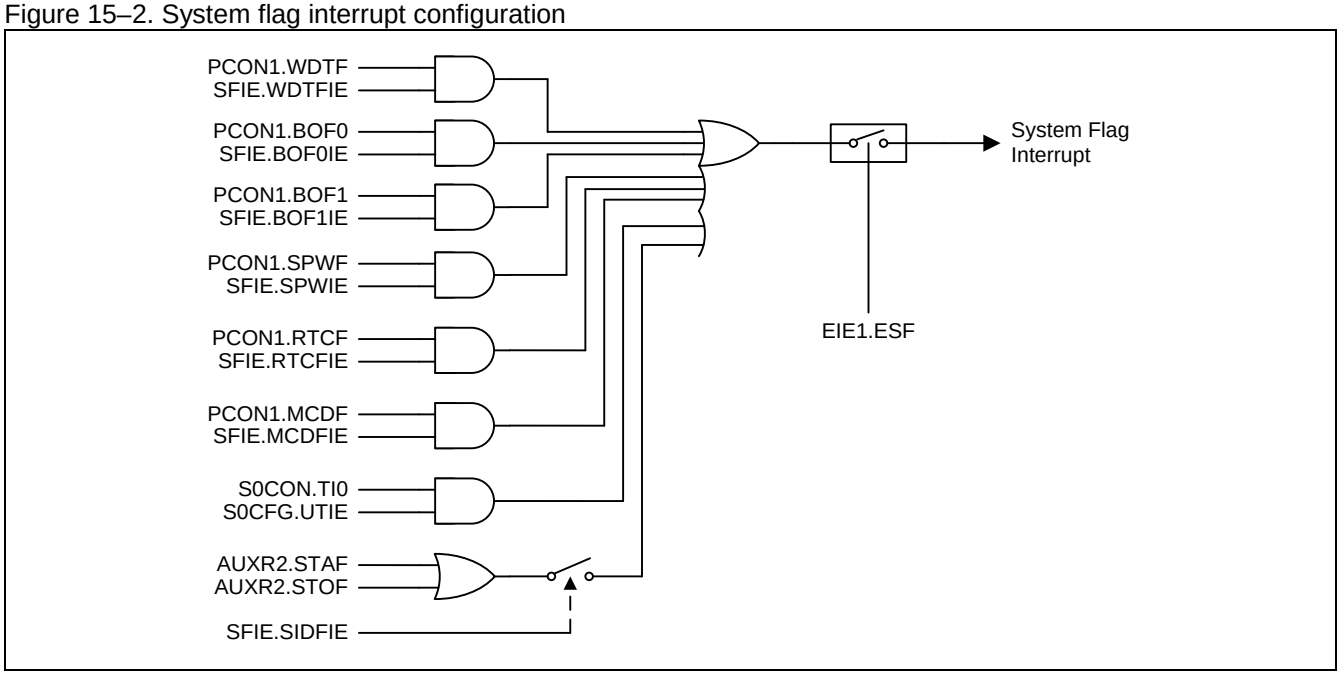
SPI interrupt is generated by SPIF in SPSTAT, which are set by SPI engine finishes a SPI transfer. It will not be cleared by hardware when the service routine is vectored to.

The ADC interrupt is generated by ADCI, ADCWI in ADCON0 and SMPF in ADCFG0. These flags will not be cleared by hardware when the service routine is vectored to.

The PTM0 interrupt is generated by the logical OR of CF, CCF5, CCF4, CCF3, CCF2, CCF1 and CCF0 in CCON. The PCA1 interrupts is generated by the C1CF1 and C1CF0 in C1CON. 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, SPWF, MCDF, 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. SPWF is set by SP monitor to indicate the warning for stack pointer overflow coming. MCDF is set by detection event of missing clock on XTAL. 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 generated 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.

The TWI1/I2C1 interrupt is generated by SI1 in SI1CON, which function is same as TWI0/I2C.

The AC0 interrupt is generated by AC0F in AC0CON, which is set by AC0OUT changed detecting on rising, falling or dual edge. It will not be cleared by hardware when the service routine is vectored to.

The AC1 interrupt is generated by AC1F in AC1CON.

When OPA0 and OPA1 are in ACMP mode also can induce interrupt. It was sharing the enable bit EAC1 to control the interrupt induce by OPA0/1. The OPA0 interrupt is generated by OP0CMPF in OPASTS. The OPA1 interrupt is generated by OP1CMPF in OPASTA.

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	Source Name	Enable Bit	Bit Location
#0	External Interrupt 0, nINT0	EX0	IE.0
#1	Timer 0	ET0	IE.1
#2	External Interrupt 1, nINT1	EX1	IE.2
#3	Timer 1	ET1	IE.3
#4	Serial Port 0	ES0	IE.4
#5	Timer 2	ET2	IE.5
#6	External Interrupt 2,nINT2	EX2	XICON.2
#7	SPI	ESPI	EIE1.0
#8	ADC	EADC	EIE1.1
#9	PTM0	EPTM0	EIE1.2
#10	System Flag	ESF	EIE1.3
#11	Keypad Interrupt	EKB	EIE1.5
#12	TWI0/I2C0	ETWI0	EIE1.6
#13	Analog Comparator 0, AC0	EAC0	EIE1.7
#14	Serial Port 1	ES1	EIE1.4
#15	External Interrupt 3,nINT3	EX3	XICON.6
#16	Timer 3	ET3	EIE2.0
#17	Analog Comparator 1/ OPA0/OPA1	EAC1	EIE2.1
#18	DMA	EDMA	IE.6
#19	PCA1	EPCA1	EIE2.5
#20	Reserved		
#21	Reserved		
#22	Reserved		
#23	TWI1/I2C1	ETWI1	EIE2.6

There are **19** interrupt sources available in **MG82F6P32**. 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 disabled 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 a hardware LCALL to the appropriate service routine as long as it is not blocked by any of the following conditions.

Block conditions:

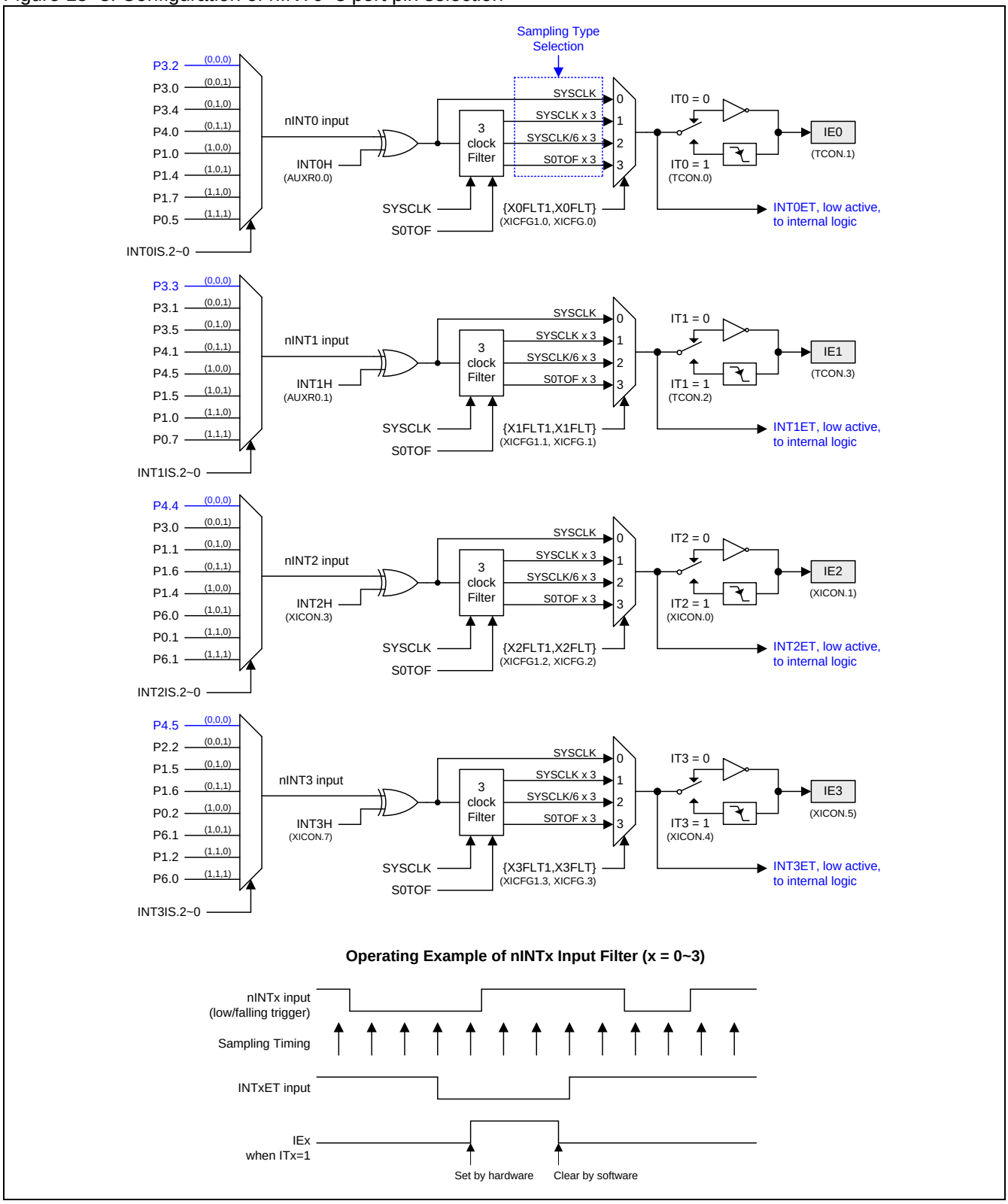
- An interrupt of equal or higher priority level is already in progress.
- The current cycle (polling cycle) is not the final cycle in the execution of the instruction in progress.
- The instruction in progress is RETI or any write to the IE, IP0L, IPH, EIE1, EIP1L, EIP1H, EIE2, EIP2L, EIP2H and XICON 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=0~3)

The **MG82F6P32** provides flexible nINT0, nINT1, nINT2 and nINT3 source selection to share the port pin inputs by mux to give most IO can enable the external interrupt event. It also provides digital filter to reduce input bunce

Figure 15–3. Configuration of nINT0~3 port pin selection



15.7. Interrupt Register

TCON: Timer/Counter Control Register

SFR Page = 0~F
SFR Address = 0x88 RESET = 0000-0000

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

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: Interrupt Enable Register

SFR Page = 0~F
SFR Address = 0xA8 RESET = 0000-0000

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

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 0 interrupt.

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

AUXR0: Auxiliary Register 0

SFR Page = 0~F
SFR Address = 0xA1 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	C1BKF	PBKF	0	0	INT1H	INT0H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 RESET = 0000-0000

7	6	5	4	3	2	1	0
INT3H	EX3	IE3	IT3	INT2H	EX2	IE2	IT2
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: INT3H, nINT3 High/Rising trigger enable.
0: Maintain nINT3 triggered on low level or falling edge on selected port pin input.
1: Set nINT3 triggered on high level or rising edge on selected port pin input.

Bit 6: EX3, external interrupt 3 (nINT3) enable register.
0: Disable external interrupt 3.
1: Enable external interrupt 3.
When CPU in IDLE and PD mode, nINT3 event will trigger IE3 and have wake-up CPU capability if EX3 is enabled. If EX3 is disabled, IE3 on nINT3 will not wake-up CPU from IDLE or PD mode.

Bit 5: IE3, External interrupt 3 (nINT3) 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 4: IT3, Interrupt 3 type control bit.
0: Cleared by CPU to specify low level triggered on nINT3. If INT3H is set, this bit specifies high level triggered on nINT3.
1: Set by CPU to specify falling edge triggered on nINT3. If INT3H is set, this bit specifies rising edge triggered on nINT3.

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 RESET = 0000-0000

7	6	5	4	3	2	1	0
PX3L	PX2L	PT2L	PSL	PT1L	PX1L	PT0L	PX0L
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: PX3L, external interrupt 3 priority-L register.
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 RESET = 0000-0000

7	6	5	4	3	2	1	0
PX3H	PX2H	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: PX3H, external interrupt 3 priority-H register.
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 RESET = 0000-0000

7	6	5	4	3	2	1	0
EAC0	ETWI0	EKB	ES1	ESF	EPTM0	EADC	ESPI
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: EAC0, Enable Analog Comparator 0 (AC0) Interrupt.
0: Disable AC0 interrupt.
1: Enable AC0 interrupt.

Bit 6: ETWI0, Enable TWI0/I2C0 interrupt.
0: Disable TWI0/I2C0 interrupt.

1: Enable TWI0/I2C0 interrupt.

Bit 5: EKBI, 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, or TI0 with UTIE when the associated system flag interrupt is enabled in SFIE.

Bit 2: EPTM0, Enable PTM0 interrupt.
0: Disable PTM0 interrupt.
1: Enable PTM0 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 RESET = 0000-0000

7	6	5	4	3	2	1	0
PAC0L	PTWI0L	PKBL	PS1L	PSFL	PPCAL	PADCL	PSPIL
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: PAC0L, AC0 interrupt priority-L register.
Bit 6: PTWI0L, 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, PTM0 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 RESET = 0000-0000

7	6	5	4	3	2	1	0
PAC0H	PTWI0H	PKBH	PS1H	PSFH	PPCAH	PADCH	PSPIH
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: PAC0H, AC0 priority-H register.
Bit 6: PTWI0H, 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, PTM0 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 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	ETWI1	EPCA1	0	0	0	EAC1	ET3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: Reserved. Software must write "**0**" on this bit when EIE2 is written.

Bit 6: ETWI1, Enable TWI1 interrupt.
0: Disable TWI1 interrupt.
1: Enable TWI1 interrupt.

Bit 5: EPCA1, Enable PCA1 interrupt.
0: Disable PCA1 interrupt.
1: Enable PCA1 interrupt.

Bit 4~2: Reserved. Software must write "**0**" on this bit when EIE2 is written.

Bit 1: EAC1, Enable Analog Comparator 1 (AC1) Interrupt.
0: Disable AC1 interrupt.
1: Enable AC1 interrupt.

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 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	PTWI1L	PPCA1L	0	0	0	PAC1L	PT3L
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: Reserved. Software must write "**0**" on this bit when EIP2L is written.
Bit 6: PTWI1L, TWI1 interrupt priority-L register.
Bit 5: PPCA1L, PCA1 interrupt priority-L register.
Bit 4~2: Reserved. Software must write "**0**" on this bit when EIP2L is written.
Bit 1: PAC1L, AC1 interrupt priority-L register.
Bit 0: PT3L, Timer 3 interrupt priority-L register.

EIP2H: Extended Interrupt Priority 2 High Register

SFR Page = 0~F
SFR Address = 0xA7 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	PTWI1H	PPCA1H	0	0	0	PAC1H	PT3H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: Reserved. Software must write "**0**" on this bit when EIP2H is written.
Bit 6: PTWI1H, TWI1 interrupt priority-H register.
Bit 5: PPCA1H, PCA1 interrupt priority-H register.
Bit 4~2: Reserved. Software must write "**0**" on this bit when EIP2H is written.
Bit 1: PAC1H, AC1 interrupt priority-H register.
Bit 0: PT3H, Timer 3 interrupt priority-H register.

DMACG0: DMA Configuration Register 0

SFR Page = **8 only**

SFR Address = 0x94 RESET = 0000-0000

7	6	5	4	3	2	1	0
PDMAH	PDMAL	CRCW0	--	EXTS10	EXTS00	FAEN0	LOOP0
R/W	**R/W**	**R/W**	**W**	**R/W**	**R/W**	**R/W**	**R/W**

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 RESET = 0000-0000

7	6	5	4	3	2	1	0
INT1IS.1	INT1IS.0	INT0IS.1	INT0IS.0	X3FLT	X2FLT	X1FLT	X0FLT
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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
0 0 0	P3.3
0 0 1	P3.1
0 1 0	P3.5
0 1 1	P4.1
1 0 0	P4.5
1 0 1	P1.5
1 1 0	P1.0
1 1 1	P0.7

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
0 0 0	P3.2
0 0 1	P3.0
0 1 0	P3.4
0 1 1	P4.0
1 0 0	P1.0
1 0 1	P1.4
1 1 0	P1.7
1 1 1	P0.5

Bit 3: X3FLT, nINT3 Filter mode control. It selects nINT3 input filter mode with X3FLT1 (XICFG1.3)

X3FLT1, X3FLT	nINT3 input filter mode
0 0	Disabled
0 1	SYSCLK x 3
1 0	SYSCLK/6 x 3
1 1	S0TOF x 3

Bit 2: X2FLT, nINT2 Filter mode control. It selects nINT2 input filter mode with X2FLT1 (XICFG1.2)

X2FLT1, X2FLT	nINT2 input filter mode
0 0	Disabled
0 1	SYSCLK x 3
1 0	SYSCLK/6 x 3
1 1	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
0 0	Disabled
0 1	SYSCLK x 3
1 0	SYSCLK/6 x 3
1 1	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
0 0	Disabled
0 1	SYSCLK x 3
1 0	SYSCLK/6 x 3
1 1	S0TOF x 3

XICFG1: External Interrupt Configured 1 Register

SFR Page = **1 only**

SFR Address = **0xC1** RESET = 0000-0000

7	6	5	4	3	2	1	0
INT1IS.2	INT0IS.2	INT2IS.1	INT2IS.0	X3FLT1	X2FLT1	X1FLT1	X0FLT1
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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 with INT2IS.2 as following table.

INT2IS.2~0	Selected Port Pin of nINT2
0 0 0	P4.4
0 0 1	P3.0
0 1 0	P1.1
0 1 1	P1.6
1 0 0	P1.4
1 0 1	P6.0
1 1 0	P0.1
1 1 1	P6.1

Bit 3: X3FLT1, nINT3 Filter mode control. It selects nINT3 input filter mode with X3FLT (XICFG.3). Refer XICFG description for nINT3 input filter mode definition.

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.

XICFG2: External Interrupt Configured 2 Register

SFR Page = **2 only**

SFR Address = **0xC1** RESET = XXXX-0000

7	6	5	4	3	2	1	0
--	--	--	--	INT3IS.2	INT3IS.1	INT3IS.0	INT2IS.2
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 3~1: INT3IS.2~0, nINT3 input port pin selection bits which function is defined as following table.

INT3IS.2~0	Selected Port Pin of nINT3
0 0 0	P4.5
0 0 1	P2.2
0 1 0	P1.5
0 1 1	P1.6
1 0 0	P0.2
1 0 1	P6.1
1 1 0	P1.2
1 1 1	P6.0

Bit 0: INT2IS2, nINT2 input port pin selection bit which function is defined with INT2IS.1~0.

SFIE: System Flag Interrupt Enable Register

SFR Page = 0~F

SFR Address = 0x8E POR = 0110-0000

7	6	5	4	3	2	1	0
SIDFIE	MCDRE	MCDFIE	RTCFIE	SPWIE	BOF1IE	BOF0IE	WDTFIE
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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 6: MCDRE, Enable Missing-Clock-Detection event causes a system reset.
0: Disable MCD event to trigger a system Reset.
1: Enable MCD event to trigger a system Reset.

Bit 5: MCDFIE, Enable MCDF (PCON1.5) Interrupt.
0: Disable MCDF interrupt.
1: Enable MCD module and enable MCDF interrupt.

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

Bit 3: SPWIE, Enable SPWF (PCON1.3) Interrupt.
0: Disable SPWF interrupt.
1: Enable SPWF interrupt.

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 POR = 0000-0000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 5: MCDF, Missing Clock Detection 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 which detects a Missing-Clock event. Writing "1" on this bit will clear MCDF. The Missing-Clock-Detection module is enabled by MCDFIE. If MCDFIE is cleared, the Missing-Clock-Detection module is inactive. Once a missing clock event happened, software must clear MCDF before switching OSCin to XTAL.

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: SPWF, SP Warning 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 SP ≥ SPHB. Writing **"1"** on this bit will clear SPWF when SP < SPHB.

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 RESET = 0000-0000

7	6	5	4	3	2	1	0
STAF	STOF	C1PLK	C0PLK	T1X12	T0X12	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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.

16. Timers/Counters

MG82F6P32 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.

Figure 16–1. Timer 0 Mode 0 Structure

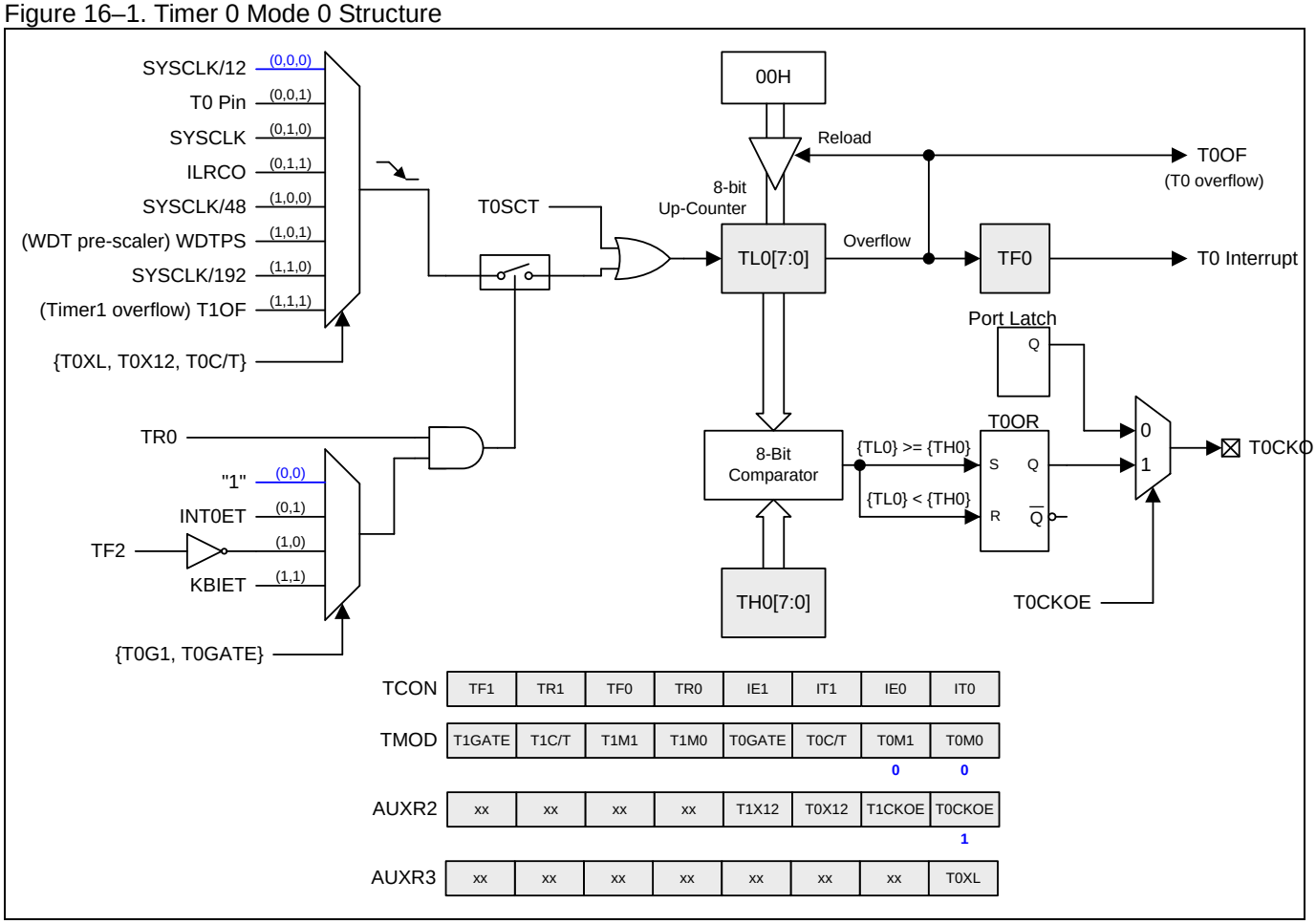
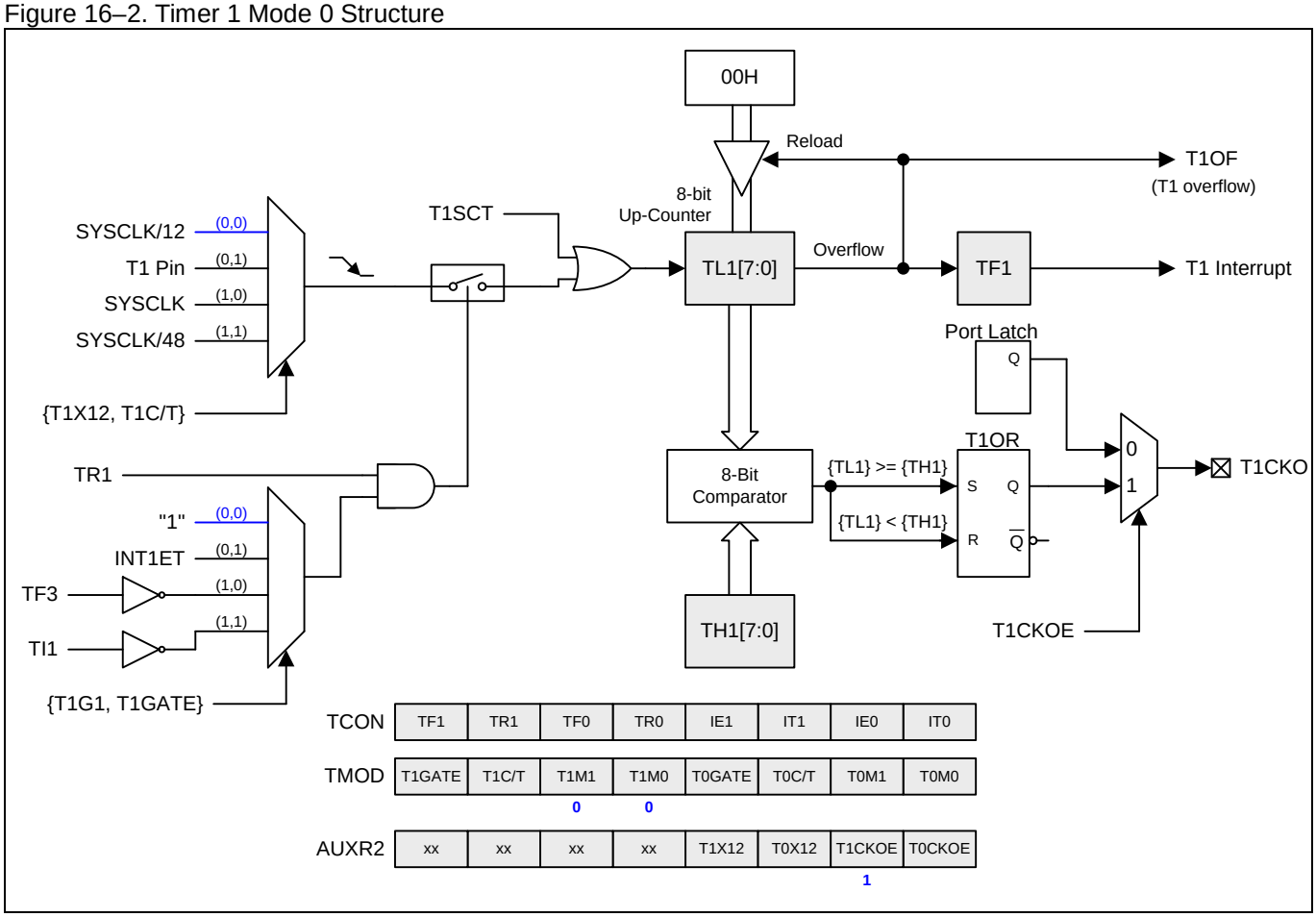


Figure 16–2. Timer 1 Mode 0 Structure



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 16–3. Timer 0 Mode 1 Structure

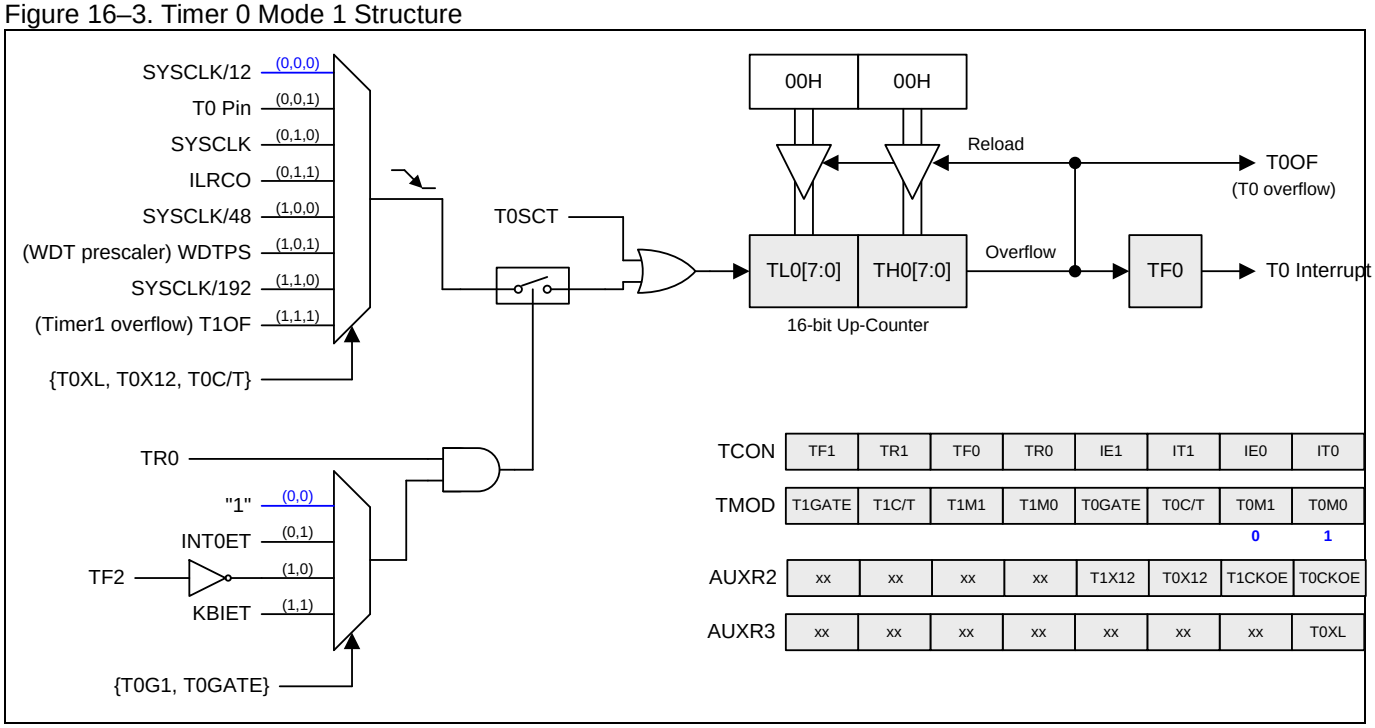
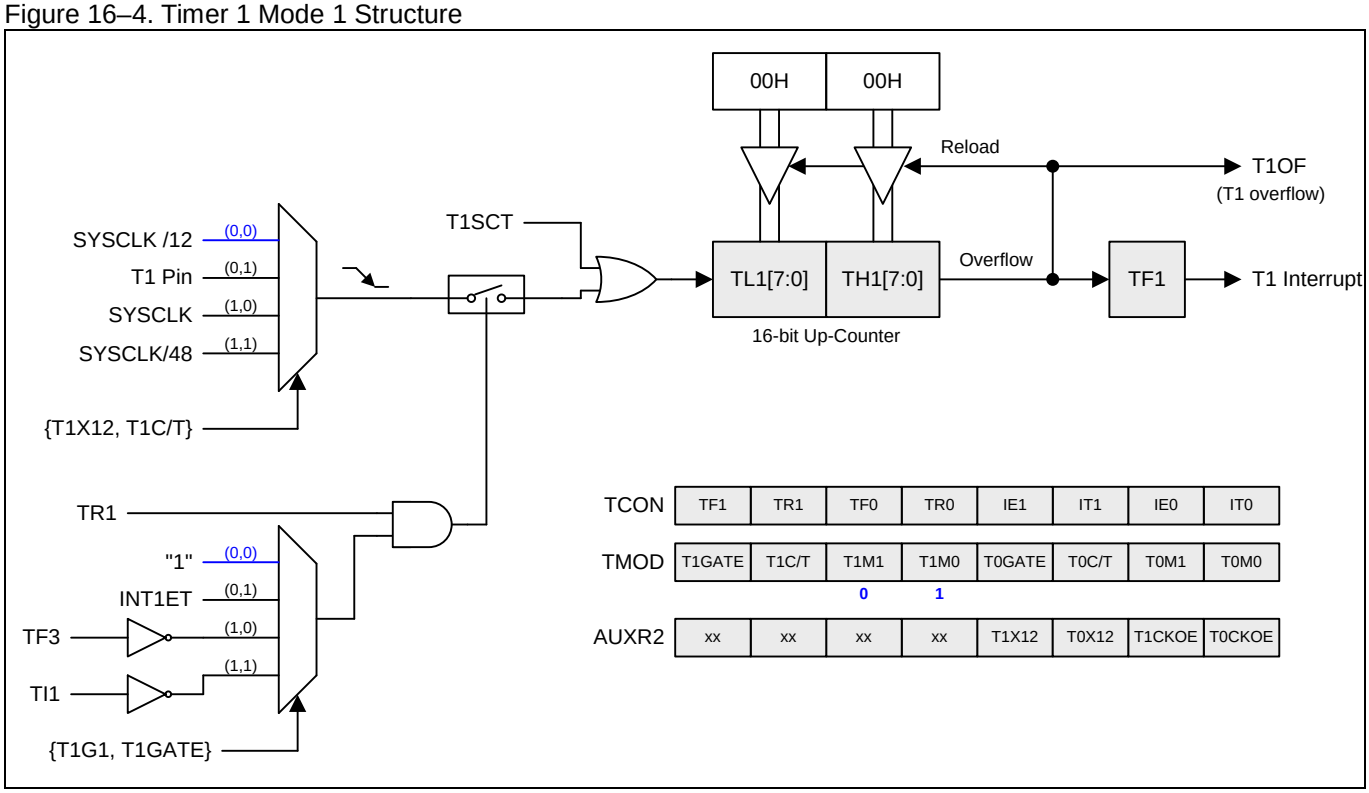


Figure 16–4. Timer 1 Mode 1 Structure



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

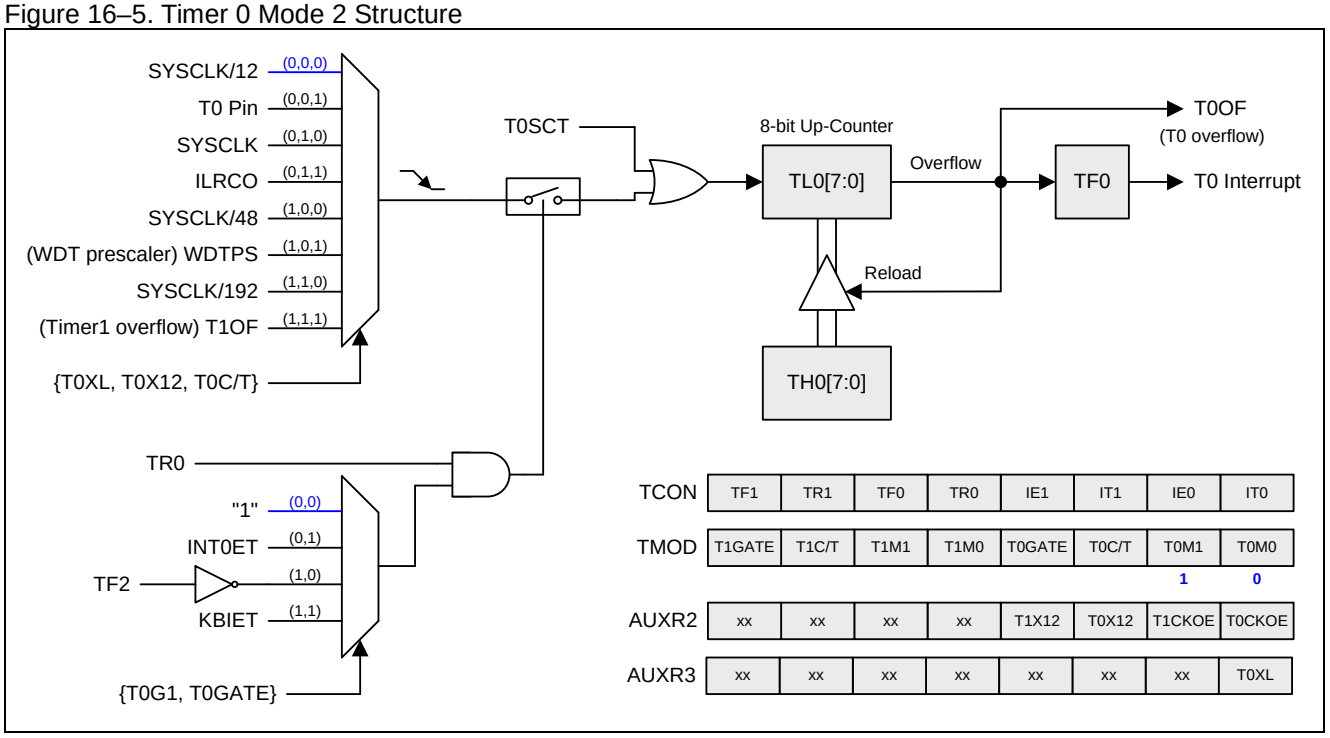
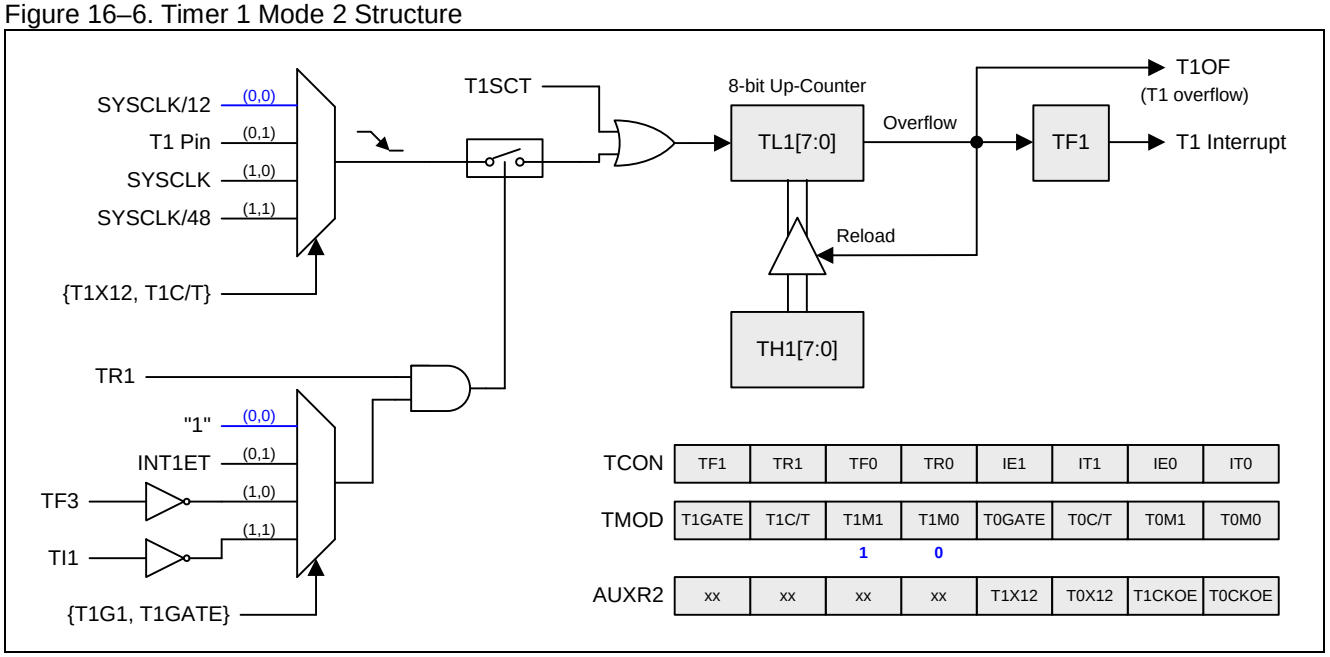


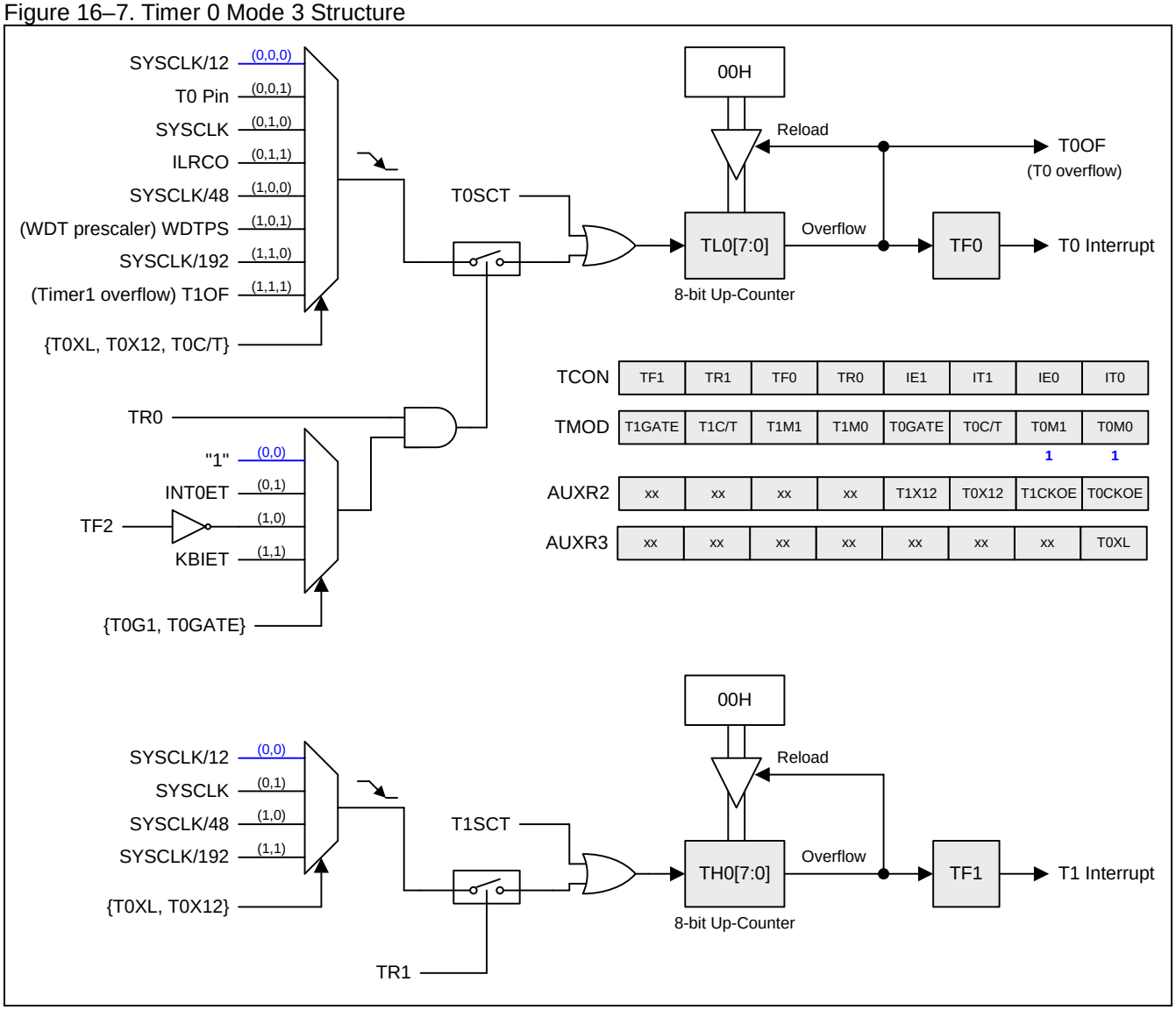
Figure 16–6. Timer 1 Mode 2 Structure



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 (cannot 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 and output control.

Figure 16–12 shows the clock-out structure of Timer 1.

Figure 16–8. Timer 0 clock out equation

$$\text{T0 Clock-out Frequency} = \frac{\text{T0 Clock Frequency}}{2 \times (256 - \text{TH0})}$$

Figure 16–9. Timer 0 clock out equation

$$\text{T1 Clock-out Frequency} = \frac{\text{T1 Clock Frequency}}{2 \times (256 - \text{TH1})}$$

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 16–10. Timer 0 in Clock Output Mode

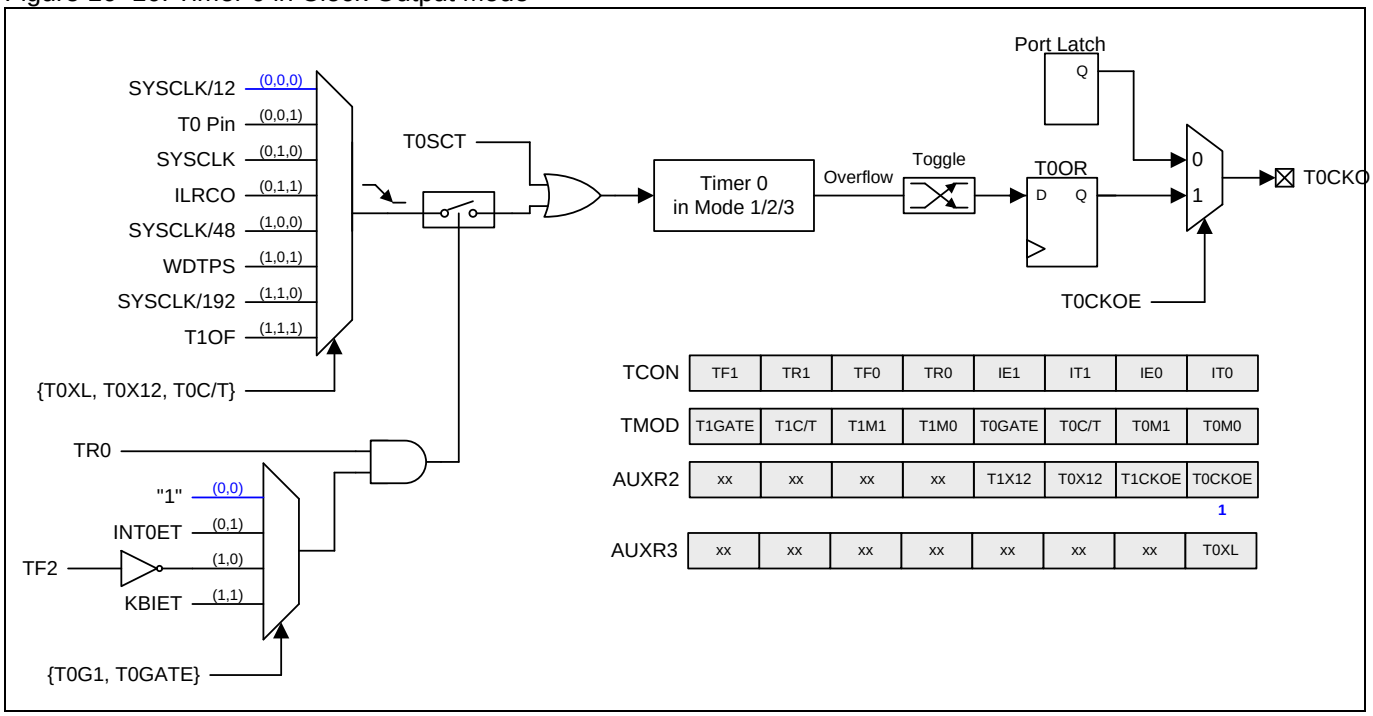


Figure 16–11. Timer 0 Clock Output Control

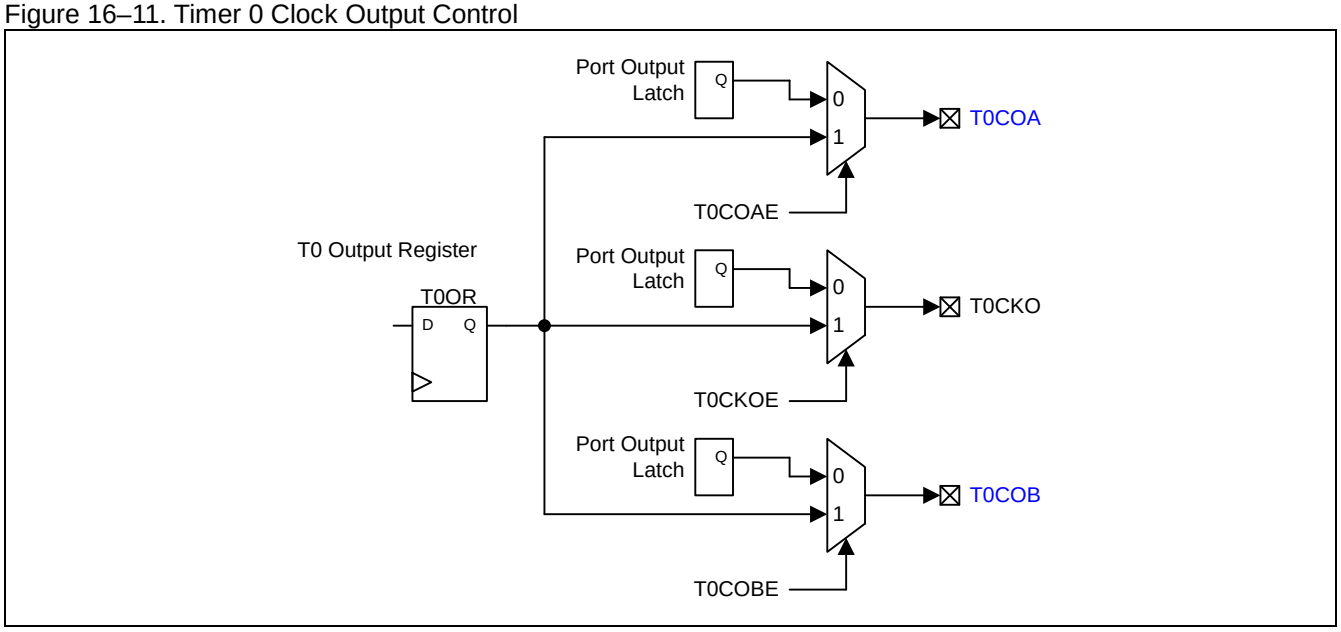
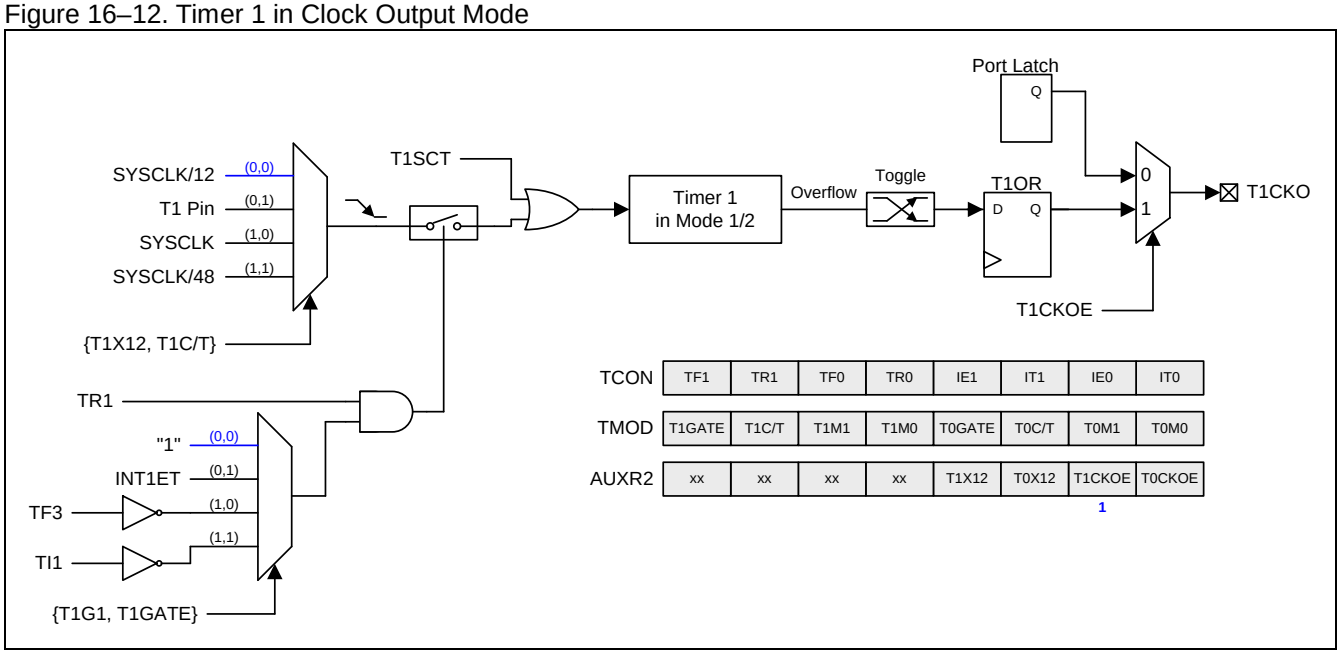


Figure 16–12. Timer 1 in Clock Output Mode



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 AUXR10 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 RESET = 0000-0000

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

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 RESET = 0000-0000

7	6	5	4	3	2	1	0
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
|←---------- Timer1 ----------→				|←---------- Timer0 ----------→			

Bit 7: T1GATE, Gating control for Timer1.

T1G1, T1GATE	T1 Gate source
0 0	Disable
0 1	INT1 active
1 0	TF3 active
1 1	TI1 active

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	T1M0	Timer 1 Operating Mode
0	0	8-bit PWM generator for Timer1
0	1	16-bit timer/counter for Timer1
1	0	8-bit timer/counter with automatic reload for Timer1
1	1	Timer/Counter1 Stopped

Bit 3: T0GATE, Gating control for Timer0.

T0G1, T0GATE	T0 Gate source
0 0	Disable
0 1	INT0 active
1 0	TF2 active
1 1	KBI active

.

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	T0M0	Timer 0 Operating Mode

0	0	8-bit PWM generator for Timer0
0	1	16-bit timer/counter for Timer0
1	0	8-bit timer/counter with automatic reload for Timer0
1	1	TL0 is 8-bit timer/counter, TH0 is locked into 8-bit timer

TL0: Timer 0 Low byte Register

SFR Page = 0~F

SFR Address = 0x8A RESET = 0000-0000

7	6	5	4	3	2	1	0
TL0.7	TL0.6	TL0.5	TL0.4	TL0.3	TL0.2	TL0.1	TL0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TH0: Timer 0 High byte Register

SFR Page = 0~F

SFR Address = 0x8C RESET = 0000-0000

7	6	5	4	3	2	1	0
TH0.7	TH0.6	TH0.5	TH0.4	TH0.3	TH0.2	TH0.1	TH0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TL1: Timer 1 Low byte Register

SFR Page = 0~F

SFR Address = 0x8B RESET = 0000-0000

7	6	5	4	3	2	1	0
TL1.7	TL1.6	TL1.5	TL1.4	TL1.3	TL1.2	TL1.1	TL1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TH1: Timer 1 High byte Register

SFR Page = 0~F

SFR Address = 0x8D RESET = 0000-0000

7	6	5	4	3	2	1	0
TH1.7	TH1.6	TH1.5	TH1.4	TH1.3	TH1.2	TH1.1	TH1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

AUXR2: Auxiliary Register 2

SFR Page = 0~F

SFR Address = 0xA3 RESET = 0000-0000

7	6	5	4	3	2	1	0
STAF	STOF	C1PLK	C0PLK	T1X12	T0X12	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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	T1 Pin
1 0	SYSCLK
1 1	SYSCLK/48

Bit 2: T0X12, Timer 0 clock source selection with T0C/T and T0XL control.

T0XL, T0X12, T0C/T	Timer 0 Clock Selection
0 0 0	SYSCLK/12
0 0 1	T0 Pin
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

AUXR3: Auxiliary Register 3

SFR Page = **0 only**

SFR Address = 0xA4 RESET = 0000-0000

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

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

T0PS1~0	T0/T0CKO
0 0	P3.4
0 1	P4.4
1 0	P4.6
1 1	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 RESET = 0000-0000

7	6	5	4	3	2	1	0
T2PS1	T2PS0	T1PS1	T1PS0	AC1OE	0	AC0OE	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

T1PS1~0	T1/T1CKO
0 0	P3.5
0 1	P4.5
1 0	P1.7
1 1	P3.3

AUXR9: Auxiliary Register 9

SFR Page = 6 only

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
SIDPS1	SIDPS0	T1G1	T0G1	0	0	S1PS1	S1PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5: T1G1, Gating source selection of Timer 1.

T1G1, T1GATE	T1 Gate source
00	Disable
01	INT1 active
10	TF3 active
11	TI1 active

Bit 4: T0G1, Gating source selection of Timer 0.

T0G1, T0GATE	T0 Gate source
00	Disable
01	INT0 active
10	TF2 active
11	KBI active

AUXR11: Auxiliary Register 11

SFR Page = 8 only

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	I2C1PS1	I2C1PS0	RX1S0	0	T1CKOE	T0CKOE

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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.

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 T2CKS, T2X12 and C/T2 Timer 2 has serval operating modes: Capture, Auto-Reload (up counting), 8-bit PWM, Baud Rate Generator and Programmable Clock-Out, which are selected by bits in the T2CON, T2MOD and T2MOD1 registers.

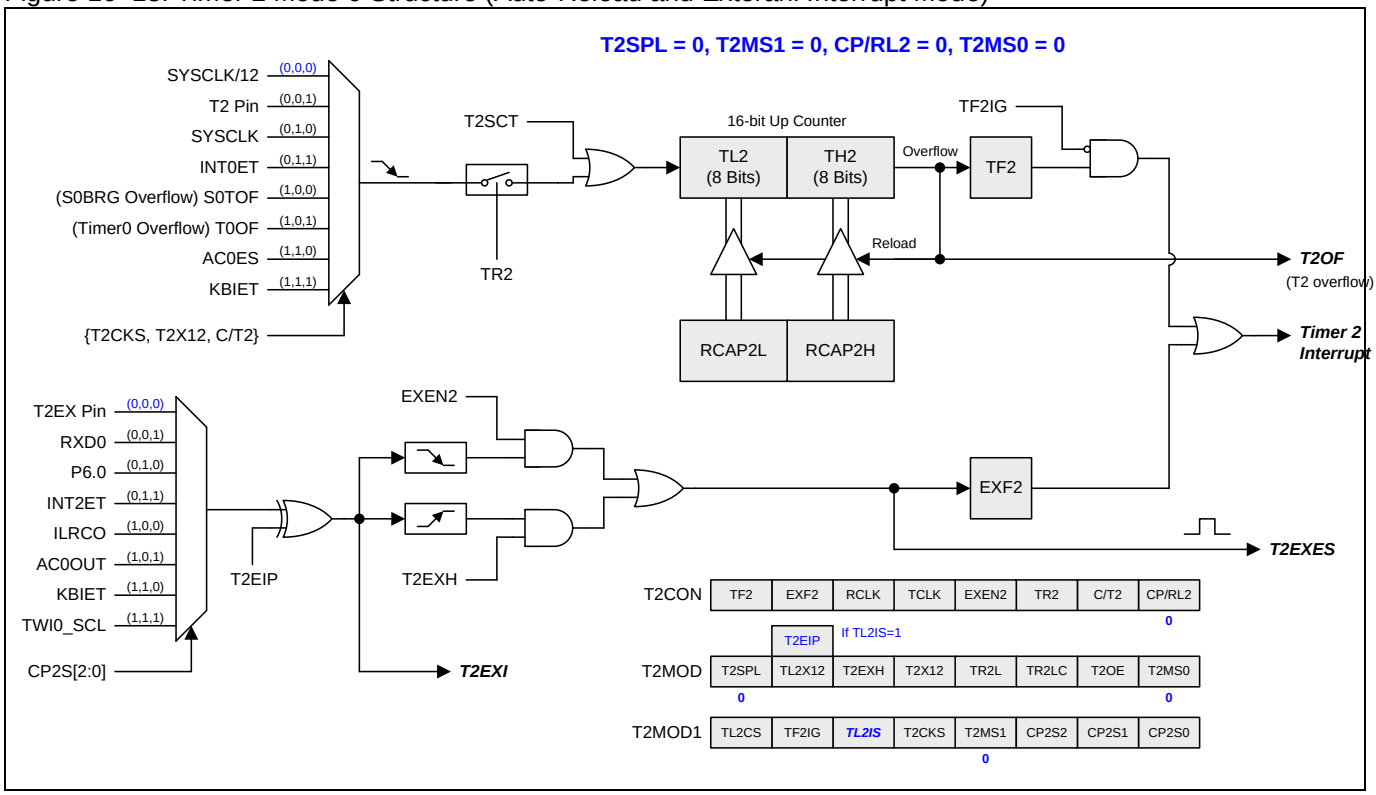
16.2.1. Timer 2 Mode 0 (Auto-Reload and External Interrupt)

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 16–13.

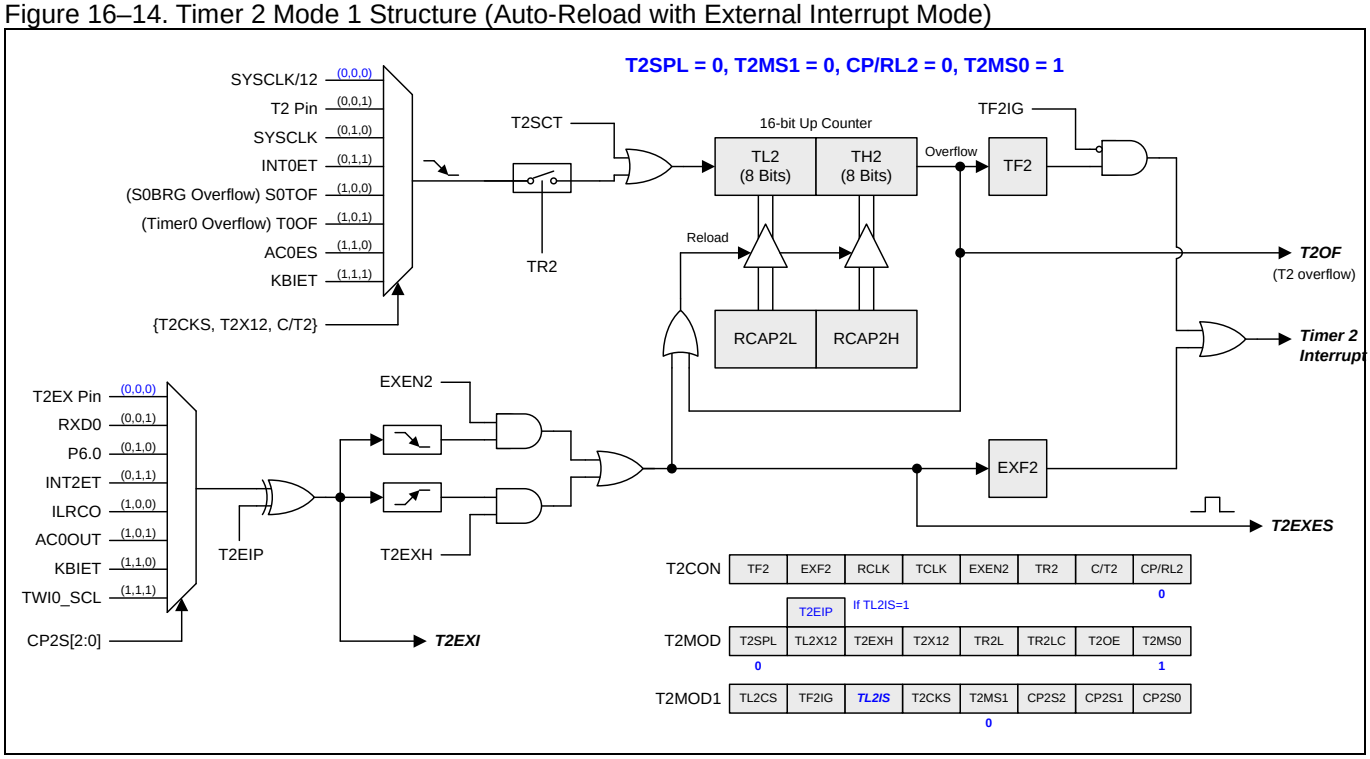
Figure 16–13. Timer 2 Mode 0 Structure (Auto-Reload and Exteranl Interrupt Mode)



16.2.2. Timer 2 Mode 1 (Auto-Reload with External Interrupt)

Figure 16–14 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 a T2 overflow or by a 1-to-0 transition of T2EXI, which is chosen from one of 8 external trigger 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.

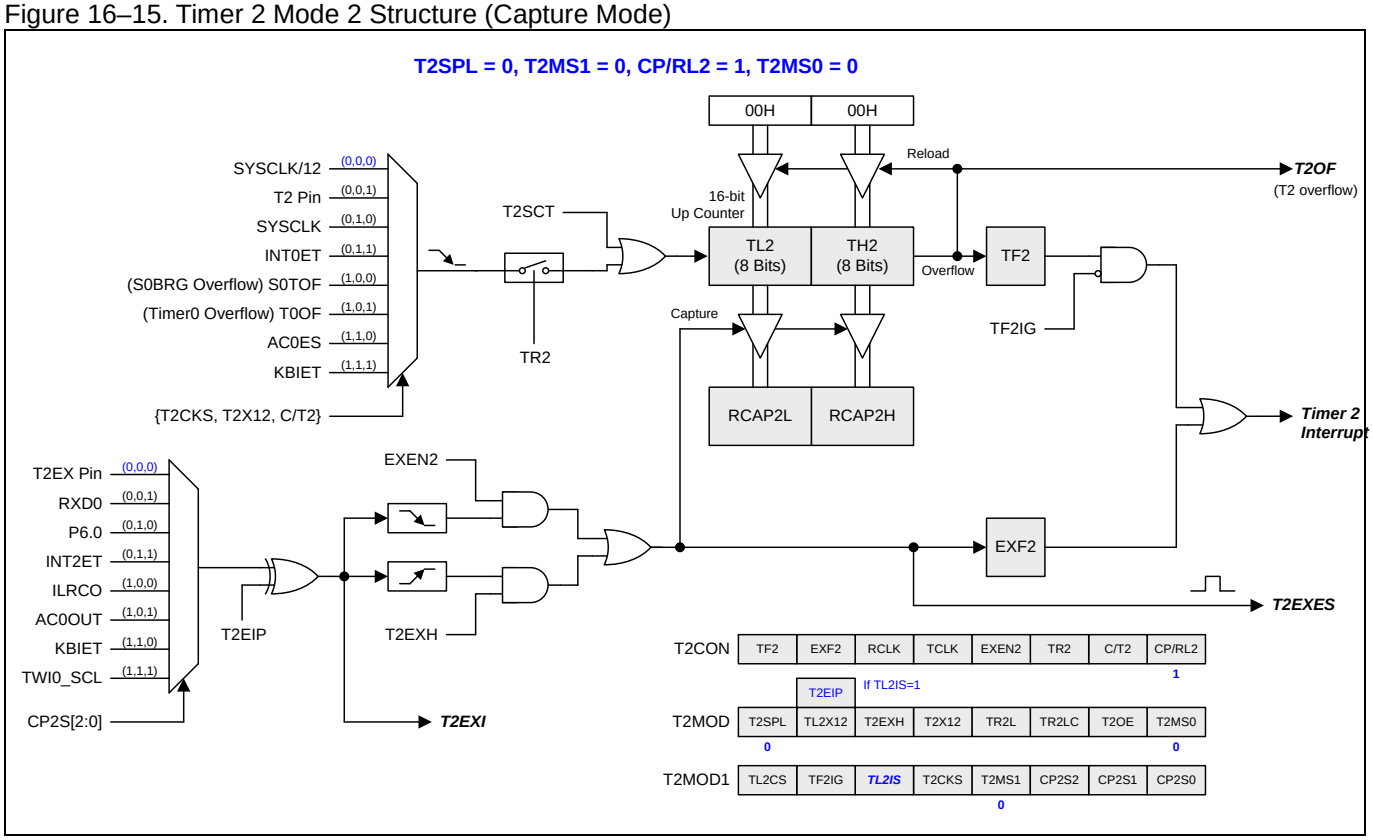
Figure 16–14. Timer 2 Mode 1 Structure (Auto-Reload with External Interrupt Mode)



16.2.3. Timer 2 Mode 2 (Capture)

Figure 16–15 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 16–15. Timer 2 Mode 2 Structure (Capture Mode)

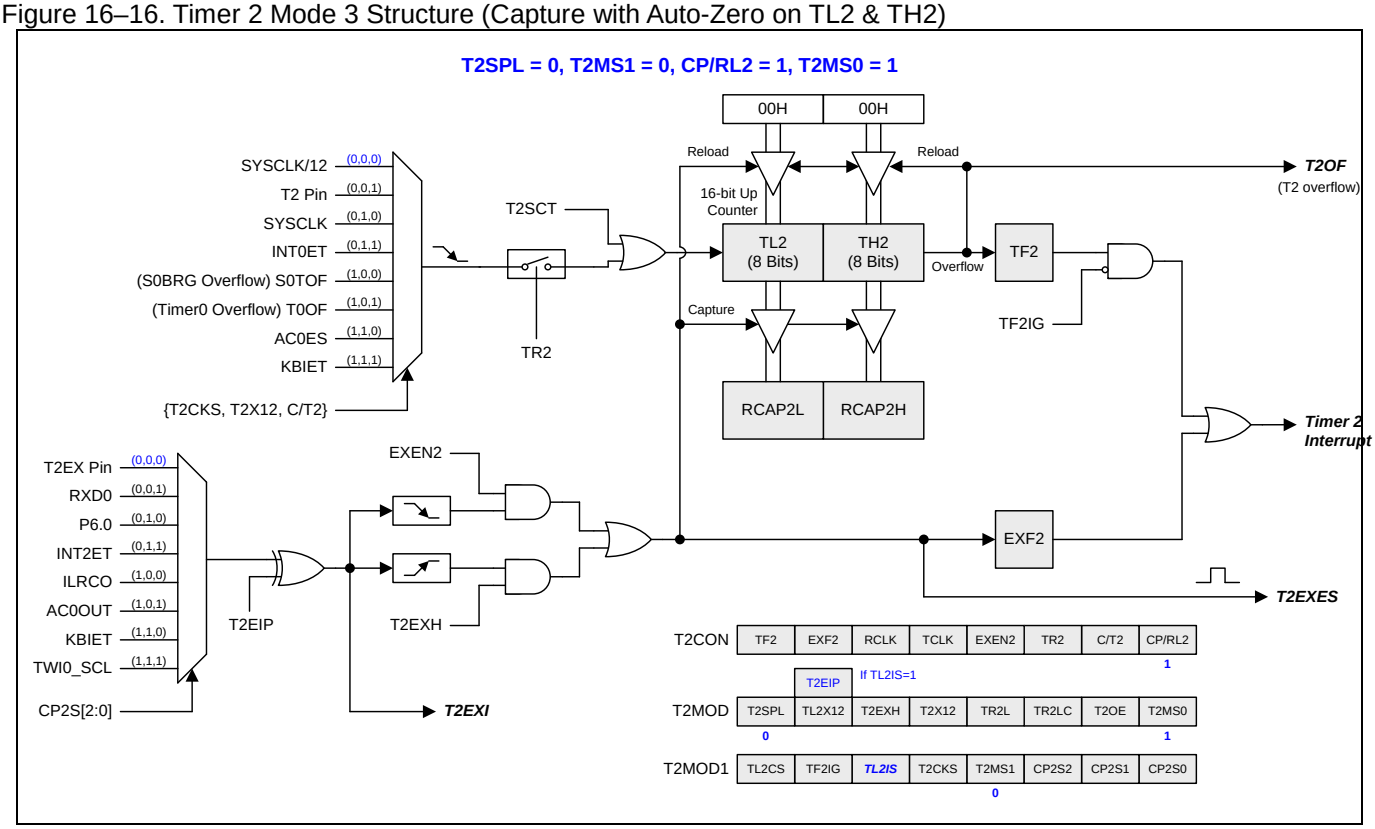


16.2.4. Timer 2 Mode 3 (Capture with Auto-Zero)

Timer 2 Mode 3 is the similar function with Timer 2 Mode 2. There is one difference that the T2EXES, EXF2 event set signal, not only is the capture source of Timer 2 but also clears the content of TL2 and TH2 to 0x0000H.

Timer 2 Mode 3 is illustrated in Figure 16–16.

Figure 16–16. Timer 2 Mode 3 Structure (Capture with Auto-Zero on TL2 & TH2)



16.2.5. Timer 2 Mode 6 (Duty Capture)

Timer 2 Mode 6 provides ability to capture cycle time or duty of the input waveform. Three edges of the signal can calculate the cycle and duty. In the timer duty capture mode it needs to clear the TH2:TL2 to 00H. And then start duty capture mode by setting TR2, but the counter will not be started yet, it will wait until the first edge send into the external trigger channel, for example T2EX Pin. It means the first edge value is 00H. And after first edge triggered, the counter start to count. Please note, the first trigger edge of the T2EXI must be the rising edge. And even you set the T2EXH, the first edge will be ignored to trigger EXF2.

Second, if only want to calculate the pulse width, then can set EXEN2 to pass the second edge to trigger EXF2. In this case, when second edge trigger timer to capture its value into RCAP2H: RCAP2L, it also trigger EXF2 for interrupt to know it has been done.

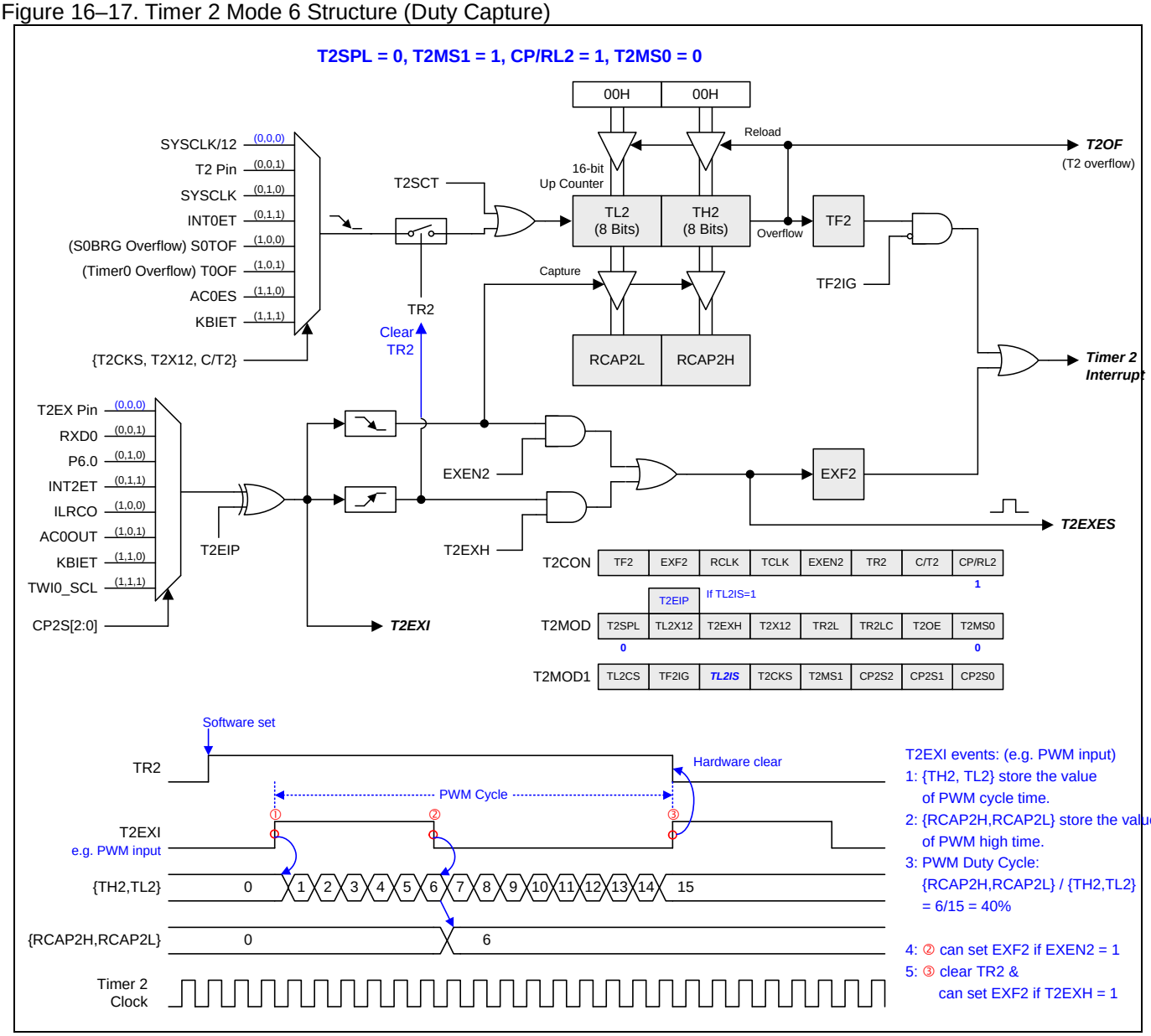
But if you want to get cycle time, then the second edge needs to be blocked by clear EXEN2.

When third edge arrives, it will automatically clear TR2 to stop the counter.

Using the TH2: TL2 (3rd edge), RCAP2H: RCAP2L (2nd edge) and 0 (1set edge) to calculate the cycle time.

Timer 2 Mode 6 is illustrated in Figure 16–17.

Figure 16–17. Timer 2 Mode 6 Structure (Duty Capture)



16.2.6. Split Timer 2 Mode 0 (AR and Ext. INT)

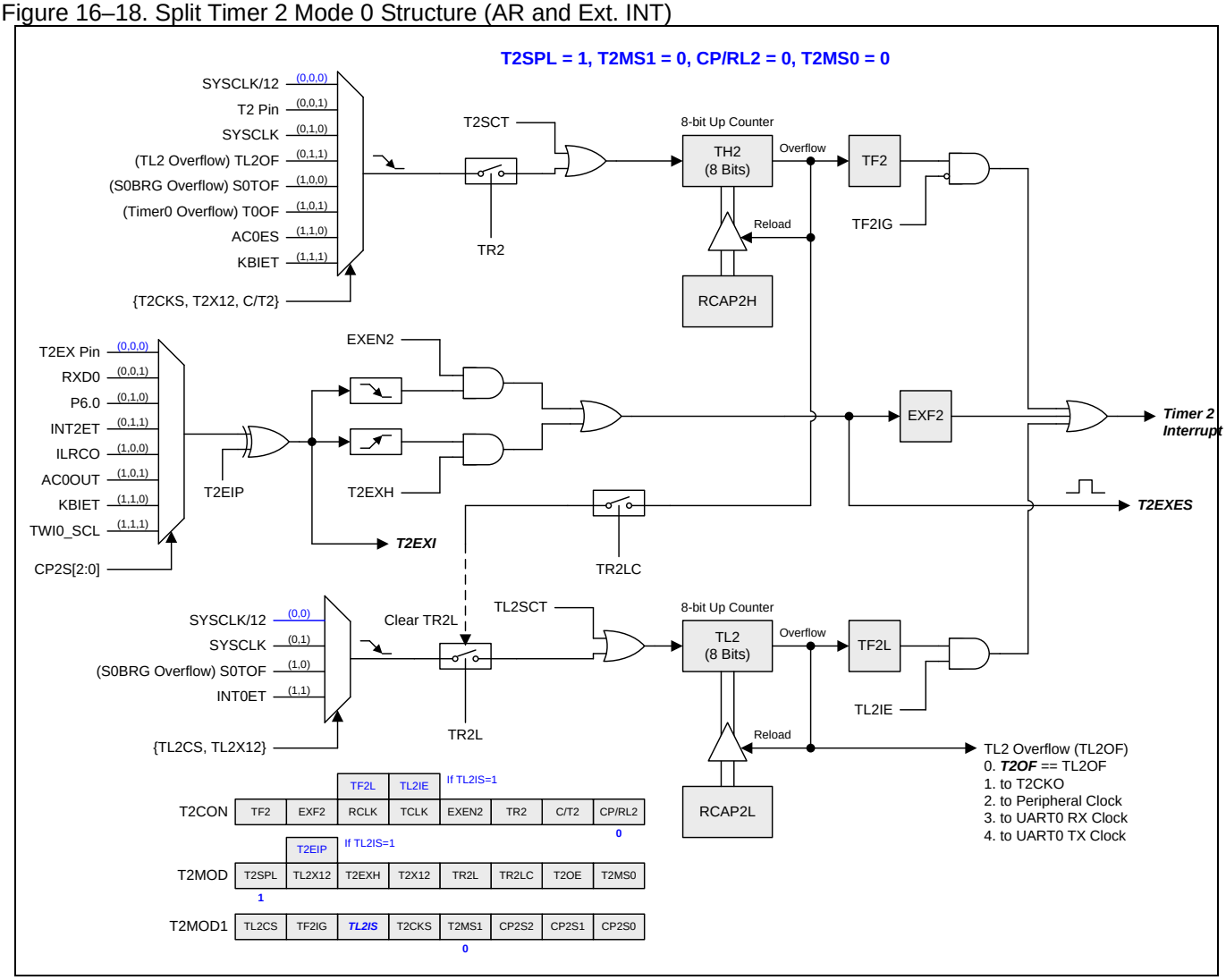
When T2SPL is set in this mode, Timer 2 operates as two 8-bit timers (TH2 and TL2). Both 8-bit timers operate in up-counter as shown in Figure 16–18. TH2 holds the reload value for RCAP2H and keep the same 8 clock source inputs selection as 16-bit mode. It behaves the 8-bit function liked Timer 2 Mode 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 and T2MOD.6 are the function of RCLK, TCLK and TL2X12. If TL2IS is “1”, the bits on T2CON.5~4 and T2MOD.6 are the function of TF2L, TL2IE and T2EIP.

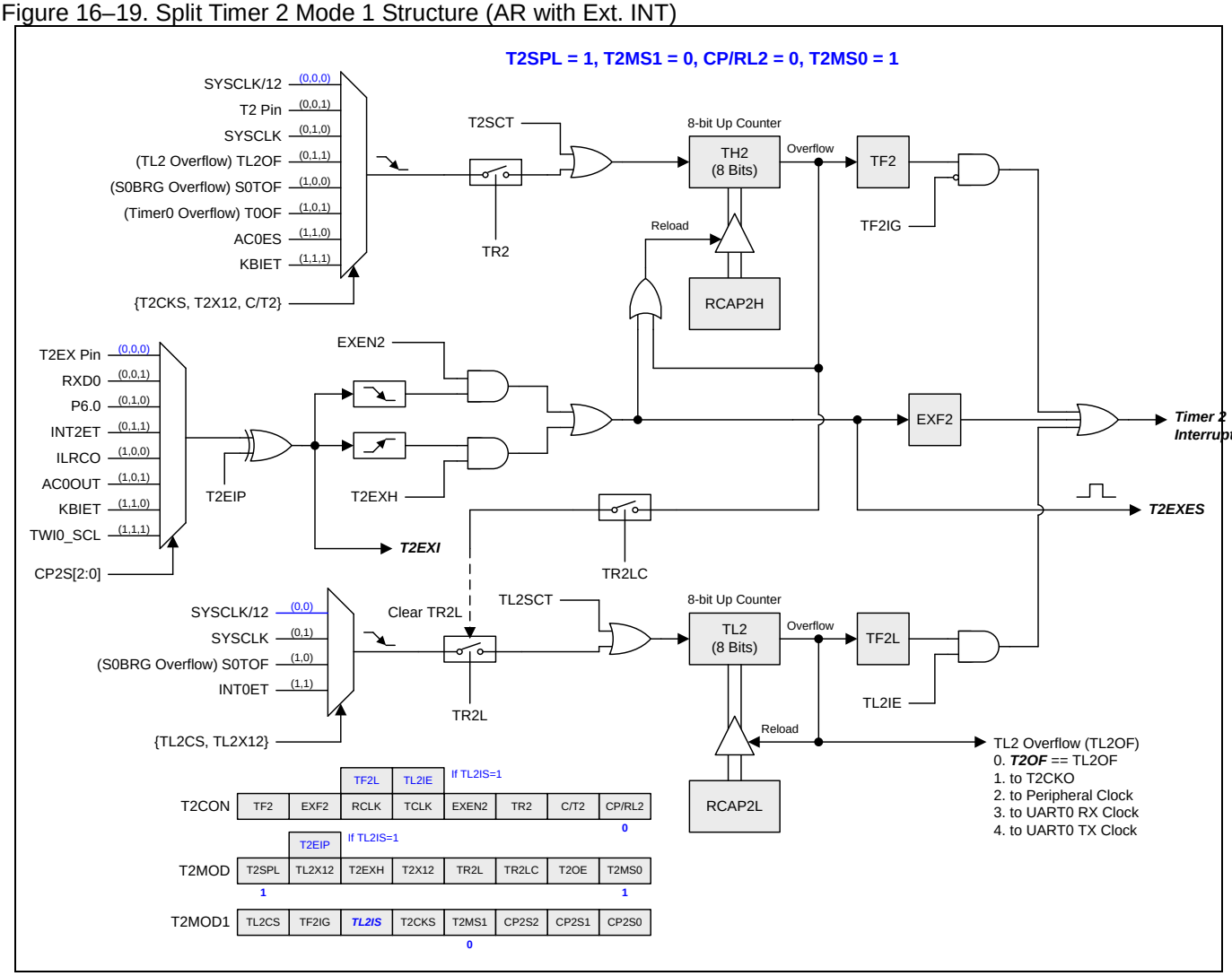
Figure 16–18. Split Timer 2 Mode 0 Structure (AR and Ext. INT)



16.2.7. Split Timer 2 Mode 1 (AR with Ext. INT)

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 1 and keeps the same interrupt scheme in Split Timer 2 Mode 0.

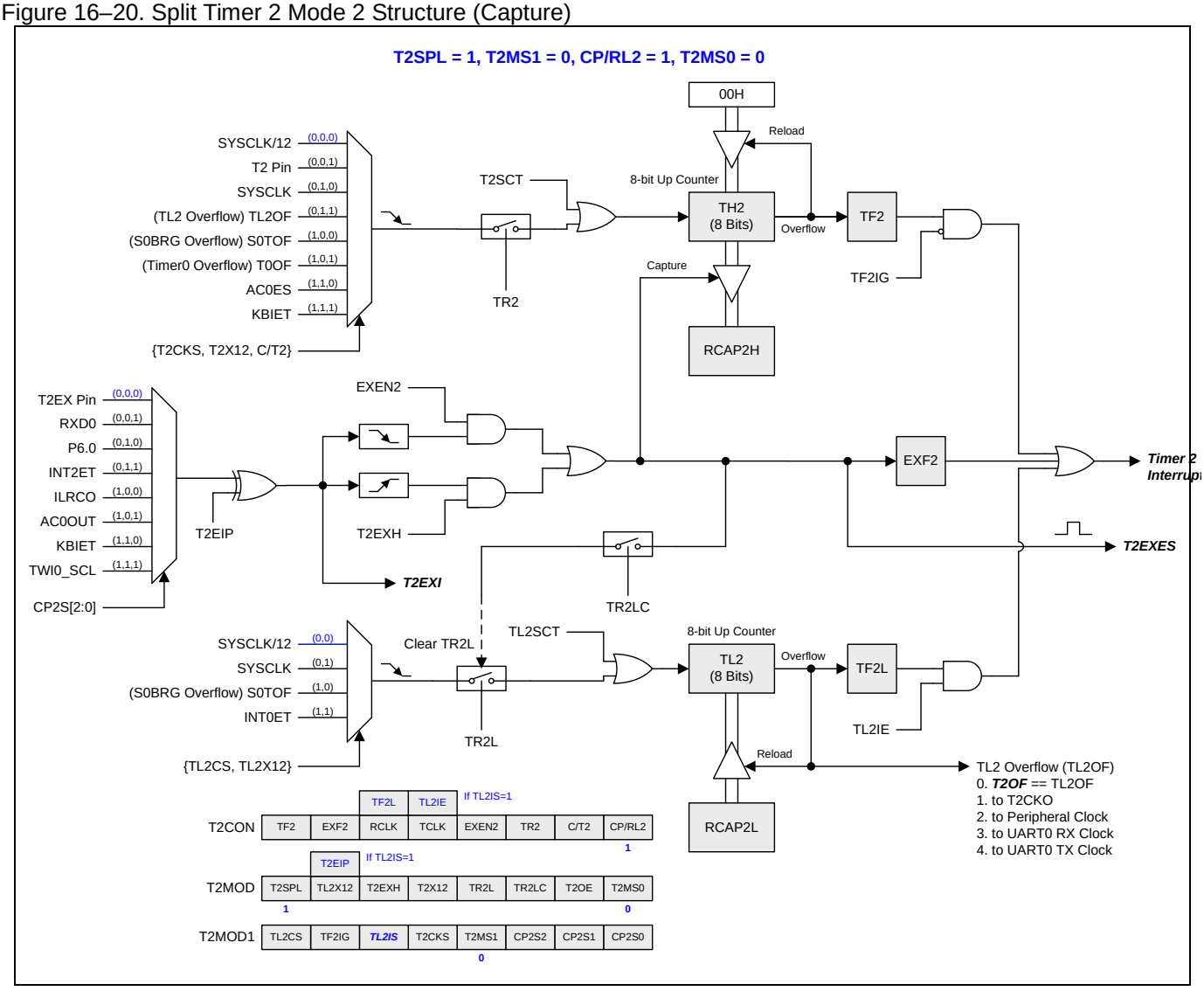
Figure 16–19. Split Timer 2 Mode 1 Structure (AR with Ext. INT)



16.2.8. 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 16–20. It is similar function as Timer 2 Mode 2 and keeps the same interrupt scheme in Split Timer 2.

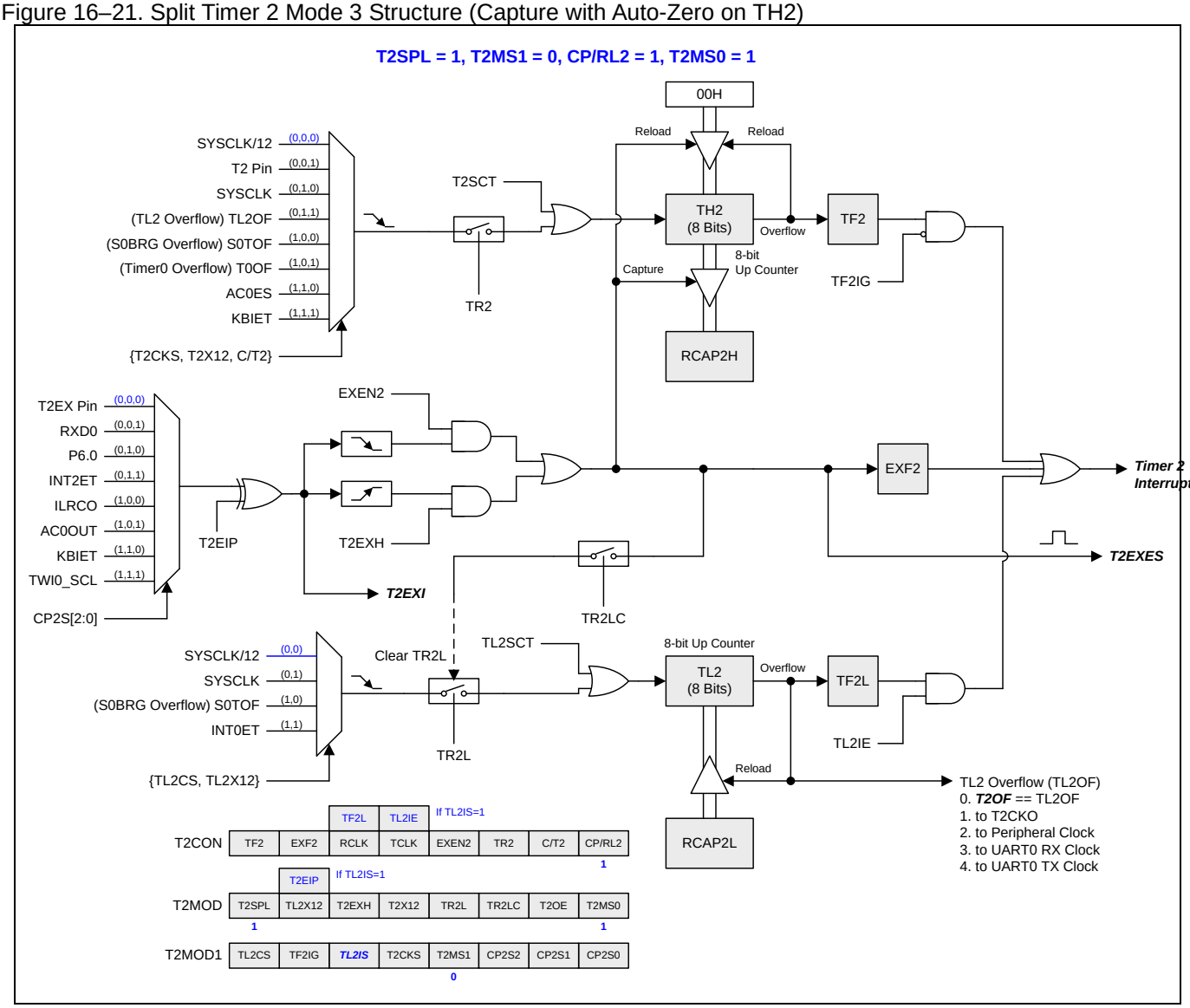
Figure 16–20. Split Timer 2 Mode 2 Structure (Capture)



16.2.9. 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 16–21. It is similar function as Timer 2 Mode 3 and keeps the same interrupt scheme in Split Timer 2 Mode 0.

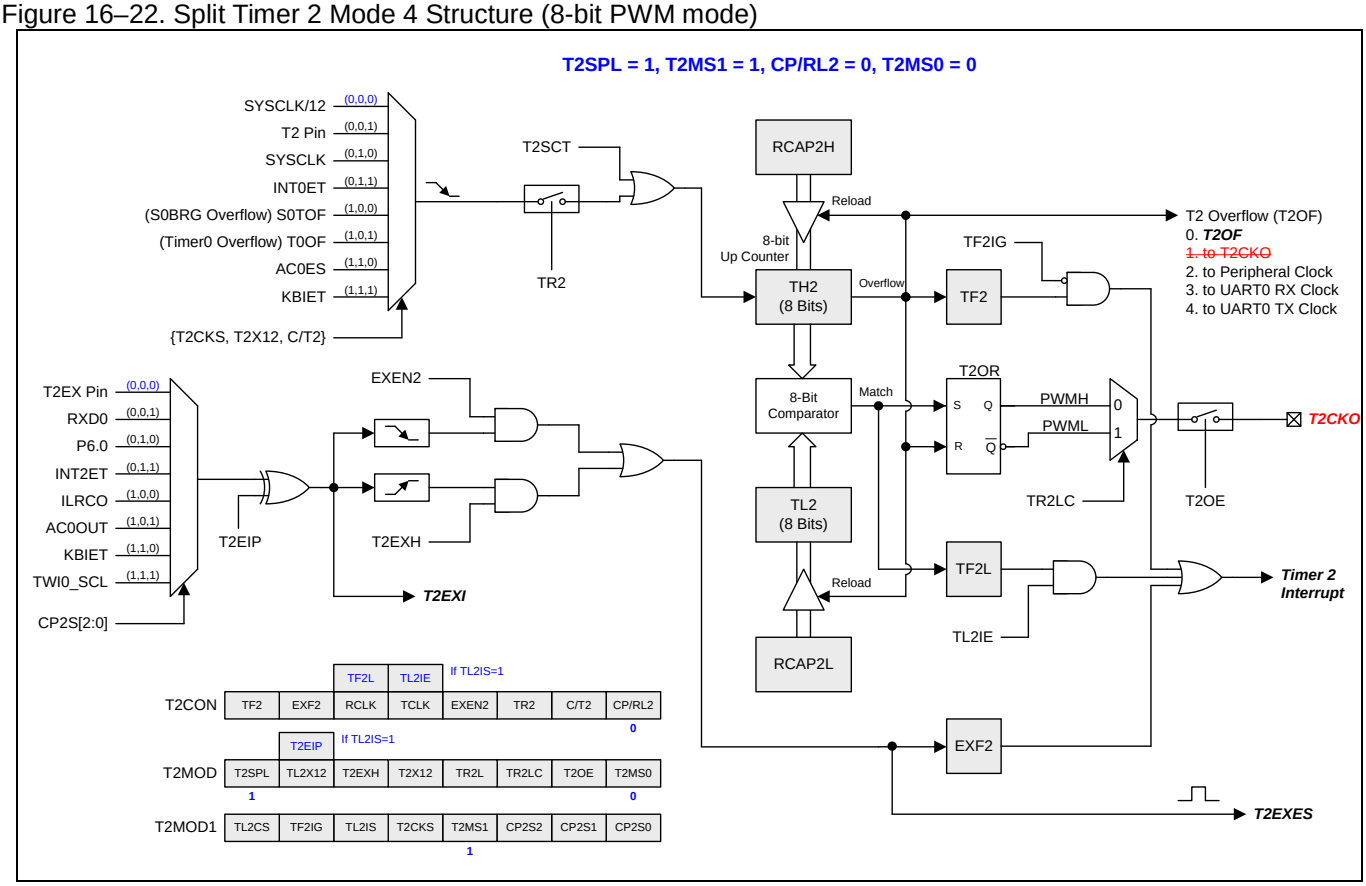
Figure 16–21. Split Timer 2 Mode 3 Structure (Capture with Auto-Zero on TH2)



16.2.10. Split Timer 2 Mode 4 (8-bit PWM Mode)

In this mode, Timer 2 is an 8-bit PWM mode as shown in Figure 16–22. 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.

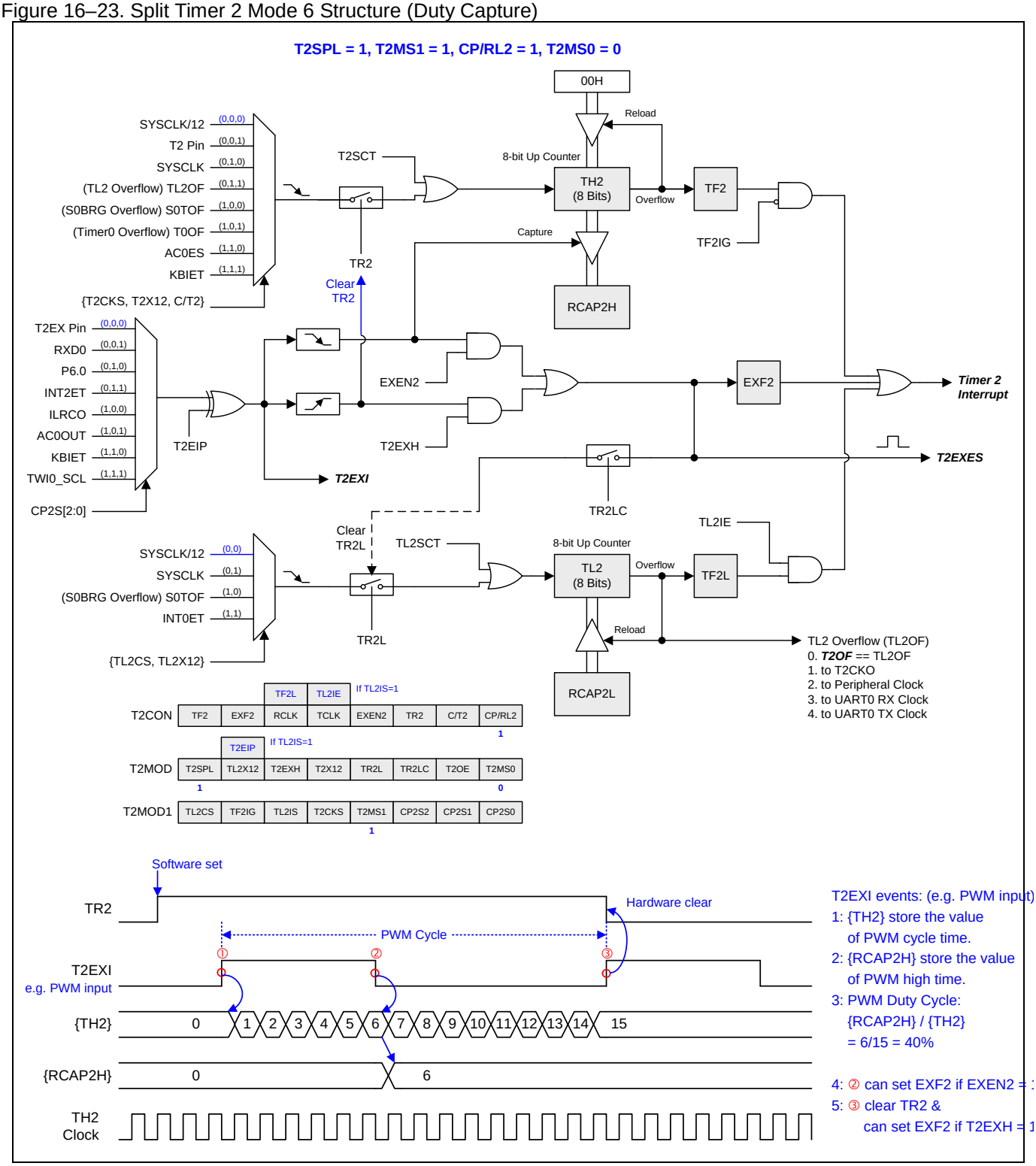
Figure 16–22. Split Timer 2 Mode 4 Structure (8-bit PWM mode)



16.2.11. Split Timer 2 Mode 6 (Duty Capture)

When T2SPL is set in this mode, Timer 2 is split to two 8-bit timers as shown in Figure 16–23. It is similar function as Timer 2 Mode 6 and keeps the same interrupt scheme in Split Timer 2.

Figure 16–23. Split Timer 2 Mode 6 Structure (Duty Capture)



16.2.12. 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 16–24 shows the Timer 2 in baud rate generation mode to generate RX Clock and TX Clock into UART engine (See Figure 19–6.). 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 rate generator, T2EX can be used as an additional external interrupt, if needed.

When Timer 2 is in the baud rate generator mode, one should not try to read or write TH2 and TL2. As a baud rate generator, Timer 2 is incremented at 1/2 the system clock or asynchronously from pin T2; under these conditions, a read or write of TH2 or TL2 may not be accurate. The RCAP2 registers may be read, but should not be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.

Note:

Refer to Section "19.9.4 Baud Rate in Mode 1 & 3" to get baud rate setting value when using Timer 2 as the baud rate generator.

If Timer 2 in Split Mode 0, TL2 and RCAP2L are combined to an 8-bit baud-rate generator as shown in Figure 16–25. TL2 overflow sets the TF2L which interrupt is enabled by TL2IE. TH2 and RCAP2H act as an 8-bit auto-reload timer/counter function with Timer 2 interrupt capability.

Figure 16–24. Timer 2 in Baud-Rate Generator Mode

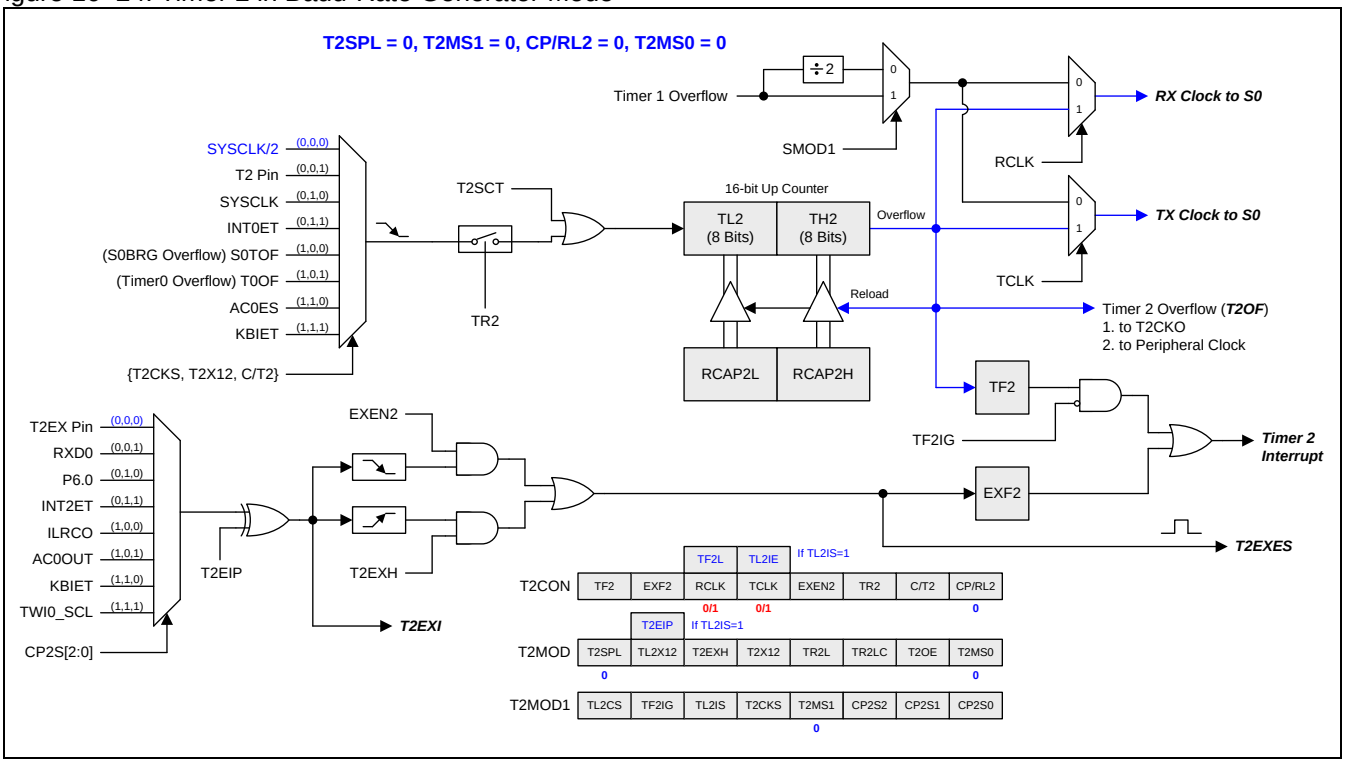
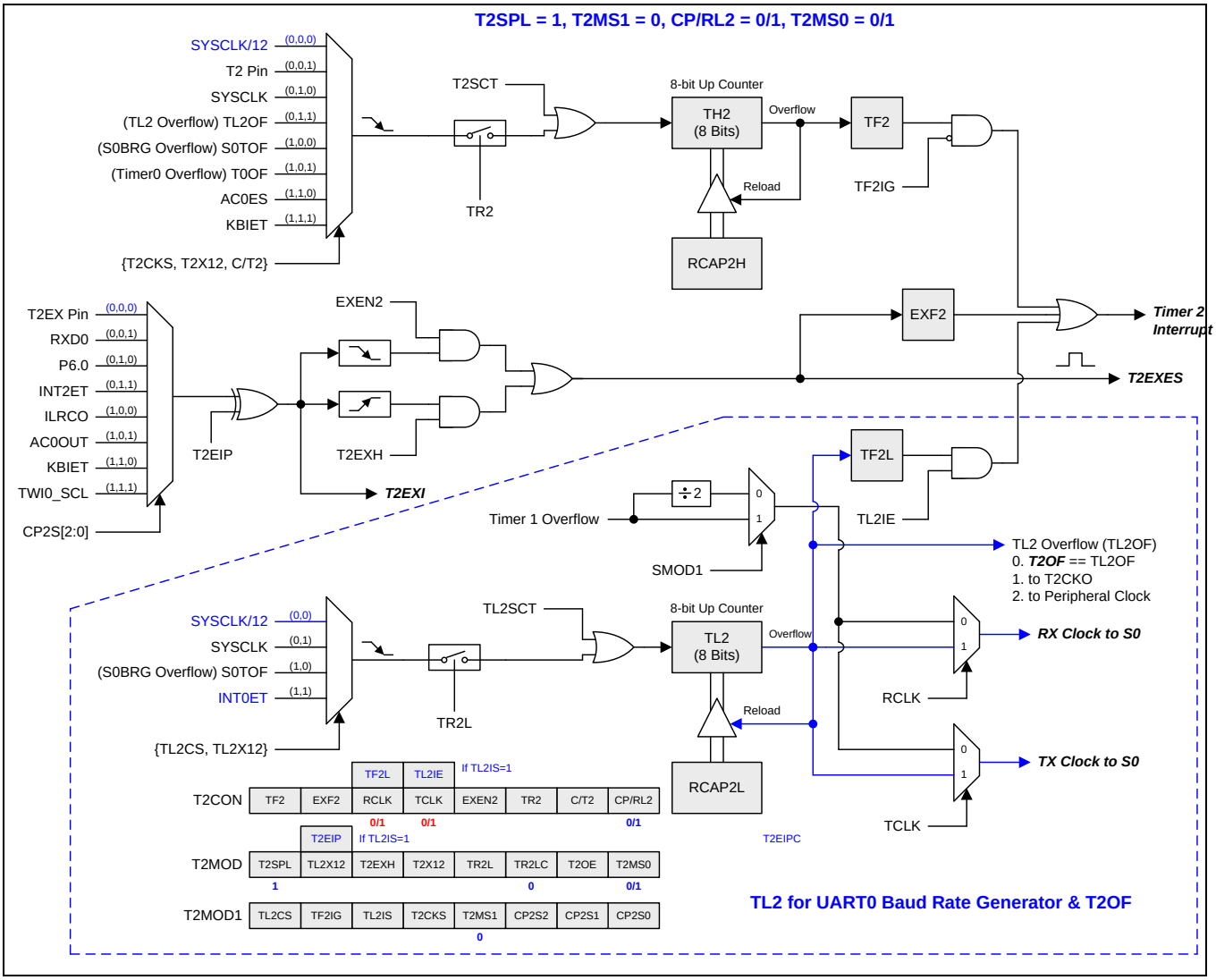


Figure 16–25. Split Timer 2 in Baud-Rate Generator Mode


T2SPL = 1, T2MS1 = 0, CP/RL2 = 0/1, T2MS0 = 0/1
SYSCLK/12 (0,0,0)
T2 Pin (0,0,1)
SYSCLK (0,1,0)
(TL2 Overflow) TL2OF (0,1,1)
(S0BRG Overflow) S0TOF (1,0,0)
(Timer0 Overflow) T0OF (1,0,1)
AC0ES (1,1,0)
KBIET (1,1,1)
{T2CKS, T2X12, C/T2}
T2SCT
TR2
8-bit Up Counter
TH2 (8 Bits)
Overflow
TF2
TF2IG
Reload
RCAP2H
T2EX Pin (0,0,0)
RXD0 (0,0,1)
P6.0 (0,1,0)
INT2ET (0,1,1)
ILRCO (1,0,0)
AC0OUT (1,0,1)
KBIET (1,1,0)
TWI0_SCL (1,1,1)
CP2S[2:0]
T2EIP
EXEN2
T2EXH
T2EXI
EXF2
Timer 2 Interrupt
T2EXES
TF2L
TL2IE
÷2
Timer 1 Overflow
SMOD1
TL2 Overflow (TL2OF)
0. T2OF == TL2OF
1. to T2CKO
2. to Peripheral Clock
SYSCLK/12 (0,0)
SYSCLK (0,1)
(S0BRG Overflow) S0TOF (1,0)
INT0ET (1,1)
{TL2CS, TL2X12}
TL2SCT
TR2L
8-bit Up Counter
TL2 (8 Bits)
Overflow
Reload
RCAP2L
RX Clock to S0
RCLK
TX Clock to S0
TCLK
T2EIPC
T2CON
TF2 EXF2 RCLK TCLK EXEN2 TR2 C/T2 CP/RL2
TF2L TL2IE If TL2IS=1
0/1 0/1 0/1
T2MOD
T2EIP If TL2IS=1
T2SPL T2X12 T2EXH T2X12 TR2L TR2LC T2OE T2MS0
1 0 0/1
T2MOD1
TL2CS TF2IG TL2IS T2CKS T2MS1 CP2S2 CP2S1 CP2S0
0
TL2 for UART0 Baud Rate Generator & T2OF

16.2.13. Timer 2 Programmable Clock Output

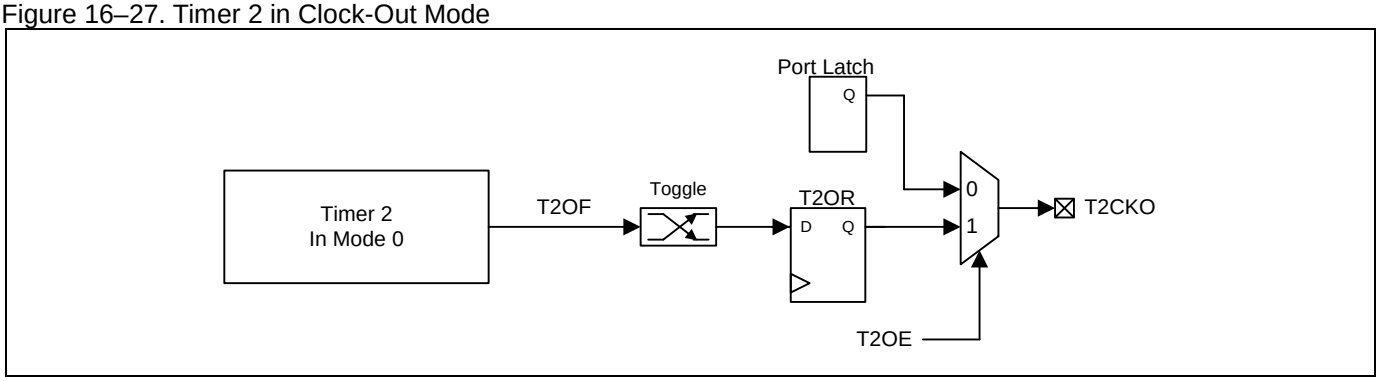
Timer 2 has a Clock-Out Mode (while CP/RL2=0 & T2OE=1). In this mode, Timer 2 operates as a programmable clock generator with 50% duty-cycle. The generated clocks come out on T2CKO port pin. The input clock (SYSCLK/2, SYSCLK/12 or SYSCLK) increments the 16-bit timer (TH2, TL2). The timer repeatedly counts to overflow from a loaded value. Once overflows occur, the contents of (RCAP2H, RCAP2L) are loaded into (TH2, TL2) for the consecutive counting. Figure 16–26 gives the formula of Timer 2 clock-out frequency: Figure 16–27 shows the clock structure of Timer 2.

Figure 16–26. Timer 2 clock out equation

$$\text{T2 Clock-out Frequency} = \frac{\text{T2 Clock Frequency}}{2 \times (65536 - (\text{RCAP2H, RCAP2L}))}$$

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 16–27. Timer 2 in Clock-Out Mode



How to Program 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.

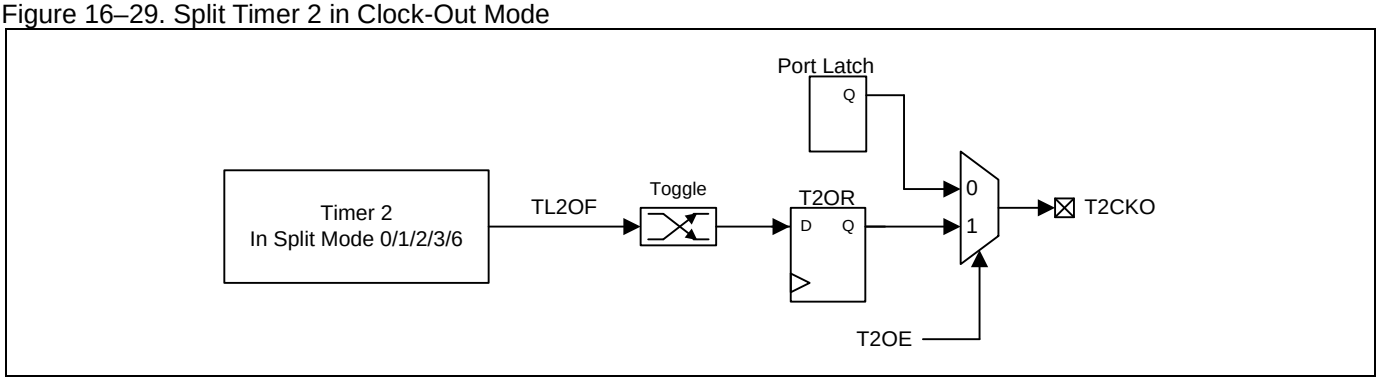
If Timer 2 in split mode, the clock output function is generated by TL2 overflow, and the output clock frequency is TL2 overflow rate /2. RCAP2L is the TL2's reload value when TL2 overflow. There are four clock source selections for TL2. Before enabling split Timer 2 clock output function, software must finish the TL2 clock source configuration. Figure 16–28 gives the formula of TL2 clock-out frequency: Figure 16–29 shows the clock structure of Split Timer 2.

Figure 16–28. Split Timer 2 clock out equation

$$\text{Split T2 Clock-out Frequency} = \frac{\text{TL2 Clock Frequency}}{2 \times (256 - \text{RCAP2L})}$$

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 16–29. 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.

16.2.14. Timer 2 Register

T2CON: Timer 2 Control Register

SFR Page = **0 Only**

SFR Address = 0xC8 RESET = 0000-0000

7	6	5	4	3	2	1	0
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

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	T2 Pin	T2 Pin
0 1 0	SYSCLK	SYSCLK
0 1 1	INT0ET	TL2OF
1 0 0	S0TOF	S0TOF
1 0 1	T0OF	T0OF
1 1 0	AC0ES	AC0ES
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 RESET = 0000-0000

7	6	5	4	3	2	1	0
T2SPL	TL2X12/ T2EIP	T2EXH	T2X12	TR2L	TR2LC	T2OE	T2MS0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: T2SPL, Timer 2 split mode control.
0: Disable Timer 2 to split mode.
1: Enable Timer 2 to split mode.

TL2IS (T2MOD1.5) must be cleared to enable access to the TL2X12 bit.
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

TL2IS (T2MOD1.5) must be set to enable access to the T2EIP bit.
Bit 6: T2EIP, T2EXI input signal inversion control bit.
0: T2EXI input signal is not inverted.
1: T2EXI input signal is inverted.

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-to1 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	Mode 0: Auto-Reload and External Interrupt
0 0 1	Mode 1: Auto-Reload with External Interrupt
0 1 0	Mode 2: Capture mode
0 1 1	Mode 3: Capture with Auto-Zero
1 0 0	Mode 4: 8-bit PWM if T2SPL = 1
1 1 0	Mode 6: Duty Capture
Others	Reserved

T2MOD1: Timer 2 Mode Register 1

SFR Page = **1 Only**

SFR Address = 0x93 RESET = 0000-0000

7	6	5	4	3	2	1	0
TL2CS	TF2IG	TL2IS	T2CKS	T2MS1	CP2S2	CP2S1	CP2S0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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, TL2IE and T2EIP access control.
0: Enable RCLK and TCLK access function on T2CON.5~4, TL2X12 access on T2MOD.6.
1: Enable TF2L and TL2IE access function on T2CON.5~4, T2EIP access on T2MOD.6.

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 Pin
0 1 1	INT2ET
1 0 0	ILRCO
1 0 1	AC0OUT
1 1 0	KBIET
1 1 1	TWI0_SCL

TL2: Timer 2 Low byte Register

SFR Page = **0 Only**

SFR Address = 0xCC RESET = 0000-0000

7	6	5	4	3	2	1	0
TL2.7	TL2.6	TL2.5	TL2.4	TL2.3	TL2.2	TL2.1	TL2.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

TH2: Timer 2 High byte Register

SFR Page = **0 Only**

SFR Address = 0xCD RESET = 0000-0000

7	6	5	4	3	2	1	0
TH2.7	TH2.6	TH2.5	TH2.4	TH2.3	TH2.2	TH2.1	TH2.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

RCAP2L: Timer 2 Capture Low byte Register

SFR Page = **0 Only**

SFR Address = 0xCA RESET = 0000-0000

7	6	5	4	3	2	1	0
RCAP2L.7	RCAP2L.6	RCAP2L.5	RCAP2L.4	RCAP2L.3	RCAP2L.2	RCAP2L.1	RCAP2L.1
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

RCAP2H: Timer 2 Capture High byte Register

SFR Page = **0 Only**

SFR Address = 0xCB RESET = 0000-0000

7	6	5	4	3	2	1	0
RCAP2H.7	RCAP2H.6	RCAP2H.5	RCAP2H.4	RCAP2H.3	RCAP2H.2	RCAP2H.1	RCAP2H.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

AUXR4: Auxiliary Register 4

SFR Page = **1 only**

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
T2PS1	T2PS0	T1PS1	T1PS0	AC1OE	AC1FLT1	AC0OE	AC0FLT1
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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

T2PS1~0	T2/T2CKO	T2EX
00	P1.0	P1.1
01	P3.2	P3.3
10	P6.7	P5.7
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 T3CKS, T3X12 and C/T3. Timer 3 has serval operating modes: Capture, Auto-Reload (up counting), 8-bit PWM, and Programmable Clock-Out, which are selected by bits in the T3CON, T3MOD and T3MOD1 registers.

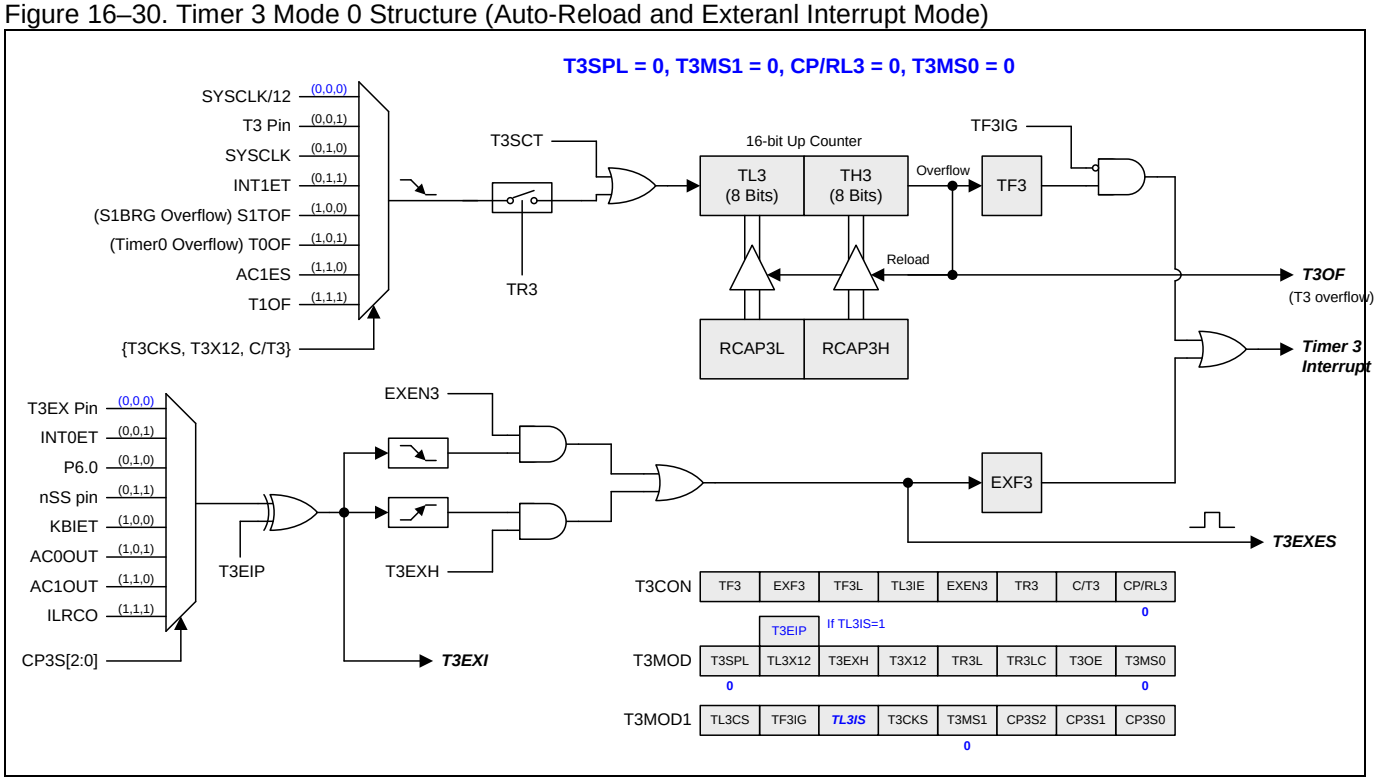
16.3.1. Timer 3 Mode 0 (Auto-Reload and External Interrupt)

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 sources 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.

Timer 3 Mode 0 is illustrated in Figure 16–30.

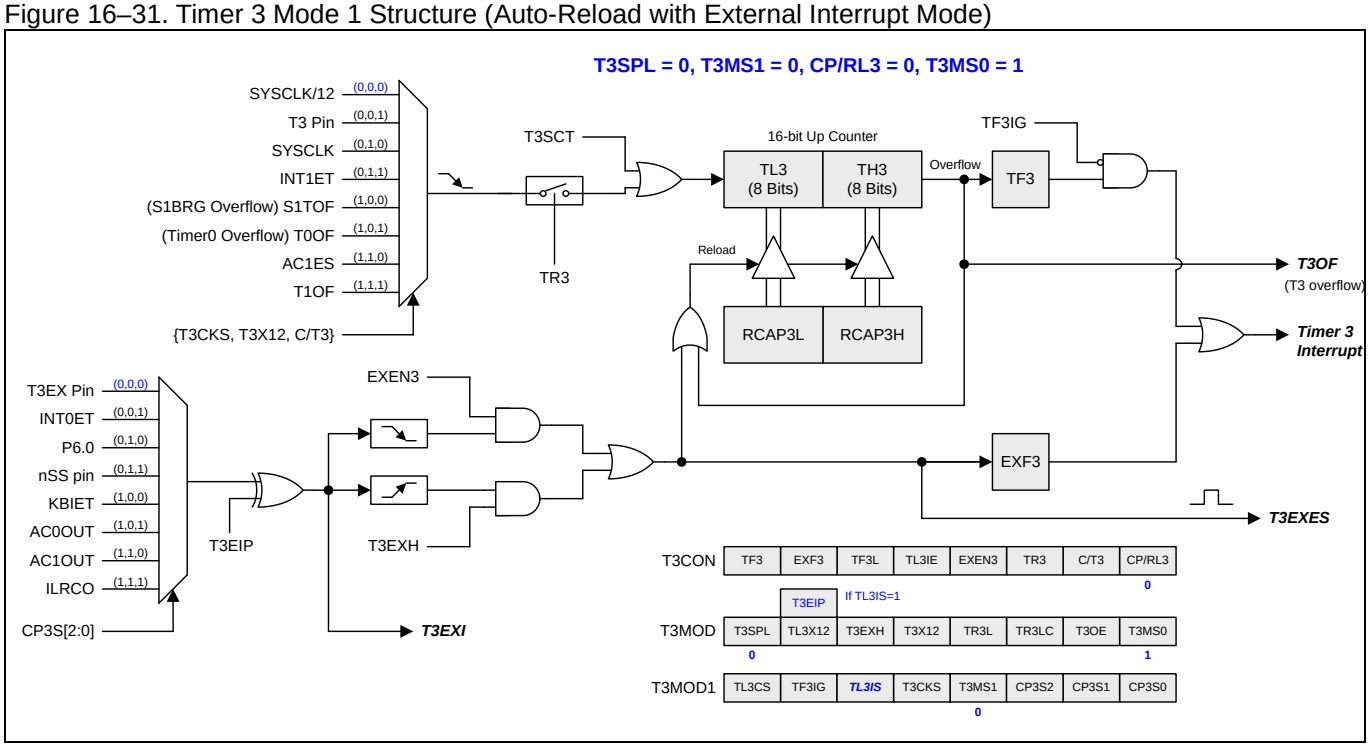
Figure 16–30. Timer 3 Mode 0 Structure (Auto-Reload and Exteranl Interrupt Mode)



16.3.2. Timer 3 Mode 1 (Auto-Reload with External Interrupt)

Figure 16–31 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 a T3 overflow or by a 1-to-0 transition of T3EXI, which is chosen from one of 8 external trigger inputs. 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.

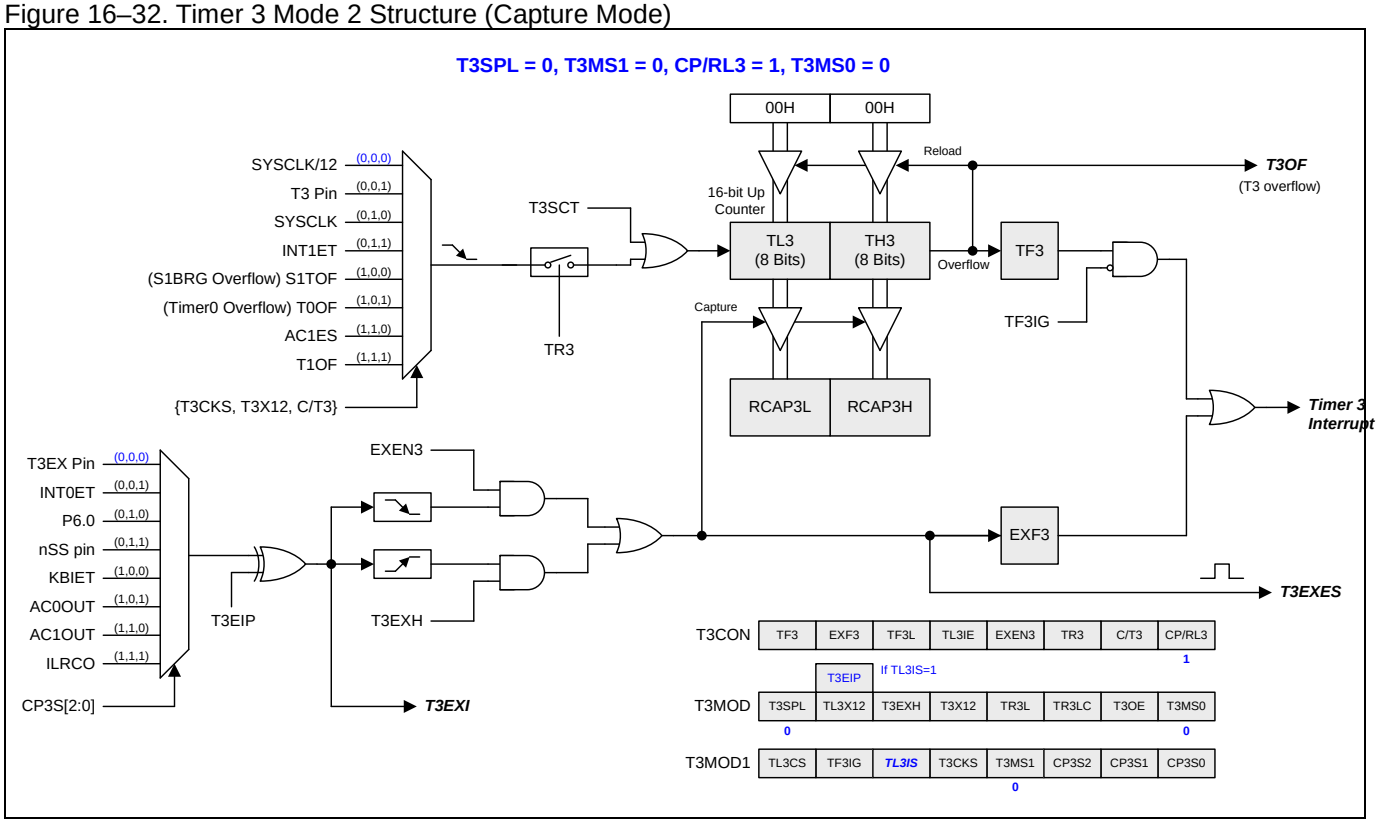
Figure 16–31. Timer 3 Mode 1 Structure (Auto-Reload with External Interrupt Mode)



16.3.3. Timer 3 Mode 2 (Capture)

Figure 16–32 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 16–32. Timer 3 Mode 2 Structure (Capture Mode)

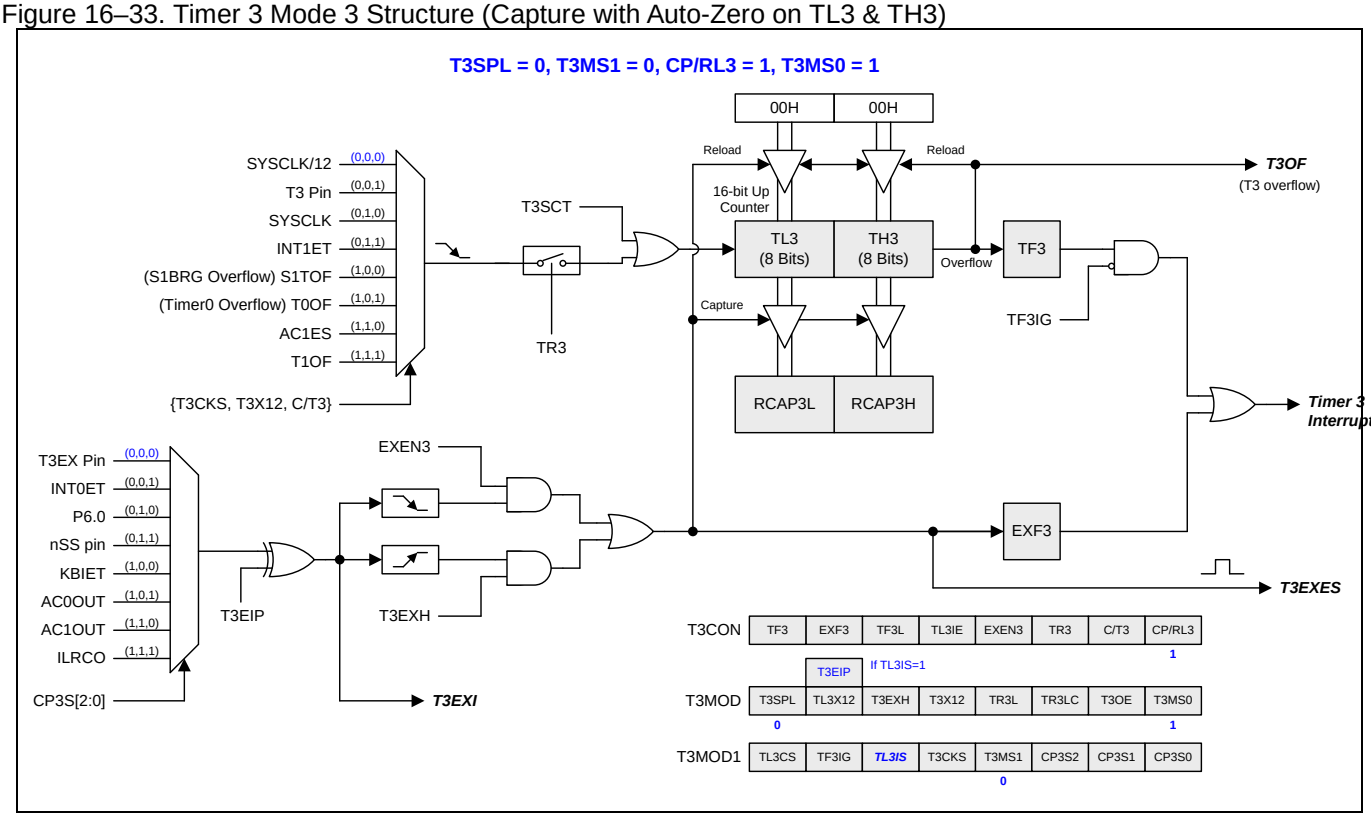


16.3.4. Timer 3 Mode 3 (Capture and Auto-Zero)

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.

Timer 3 Mode 3 is illustrated in Figure 16–33.

Figure 16–33. Timer 3 Mode 3 Structure (Capture with Auto-Zero on TL3 & TH3)



16.3.5. Timer 3 Mode 6 (Duty Capture)

Timer 3 Mode 6 provides ability to capture cycle time or duty of the input waveform. Three edges of the signal can calculate the cycle and duty. In the timer duty capture mode, it needs to clear the TH3:TL3 to 00H. And then start duty capture mode by setting TR3, but the counter will not be started yet, it will wait until the first edge send into the external trigger channel, for example T3EX Pin. It means the first edge value is 00H. And after first edge triggered, the counter starts to count. Please note, the first trigger edge of the T3EXI must be the rising edge. And even you set the T3EXH, the first edge will be ignored to trigger EXF3.

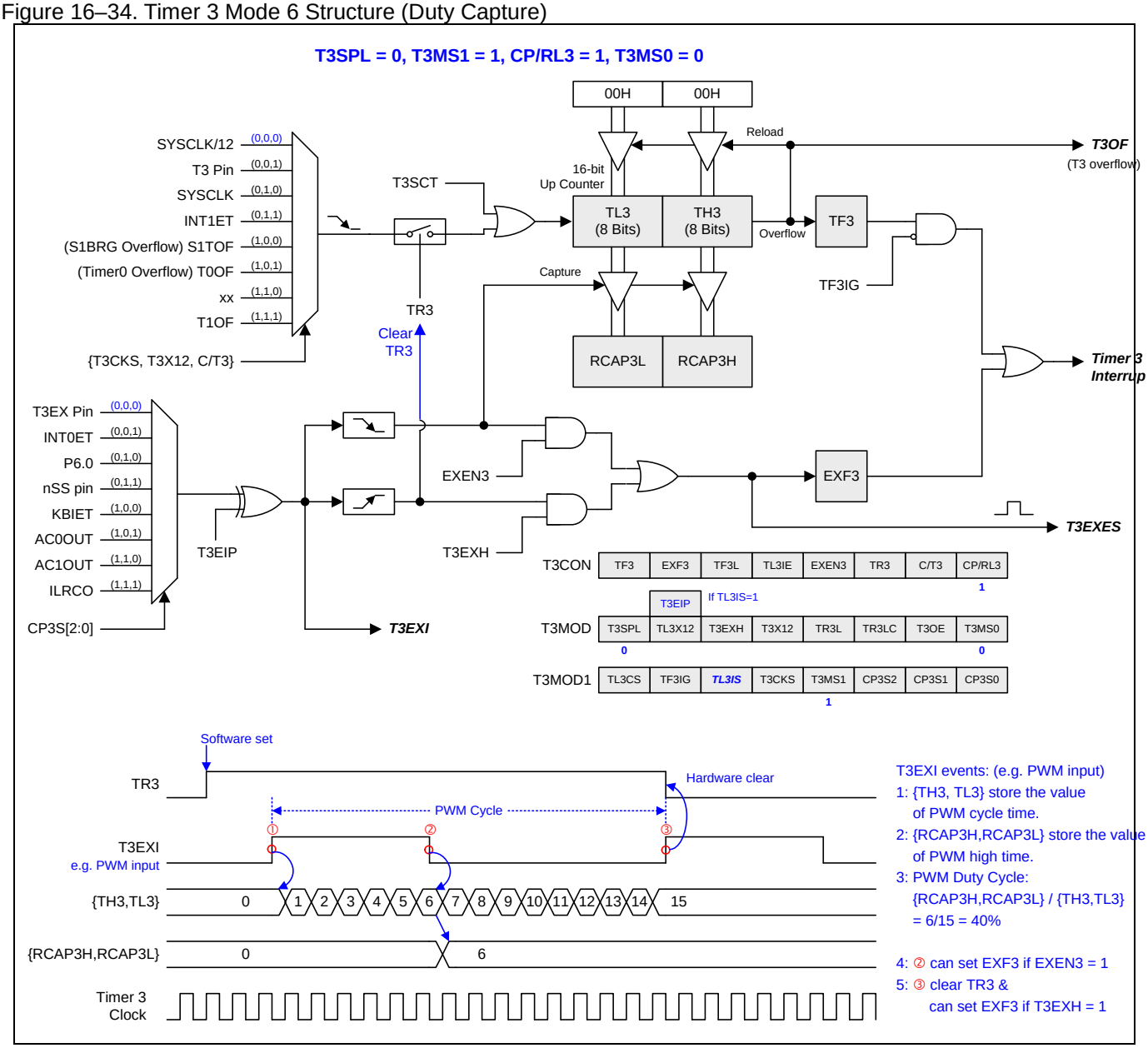
Second, if only want to calculate the pulse width, then can set EXEN3 to pass the second edge to trigger EXF3. In this case, when second edge trigger timer to capture its value into RCAP3H: RCAP3L, it also triggers EXF3 for interrupt to know it has been done.

But if you want to get cycle time, then the second edge needs to be blocked by clear EXEN3.

When third edge arrive, it will automatically clear TR3 to stop the counter.

Using the TH3: TL3 (3rd edge), RCAP3H: RCAP3L (2nd edge) and 0 (1set edge) to calculate the cycle time.
Timer 3 Mode 6 is illustrated in Figure 16–34.

Figure 16–34. Timer 3 Mode 6 Structure (Duty Capture)



16.3.6. Split Timer 3 Mode 0 (AR and Ext. INT)

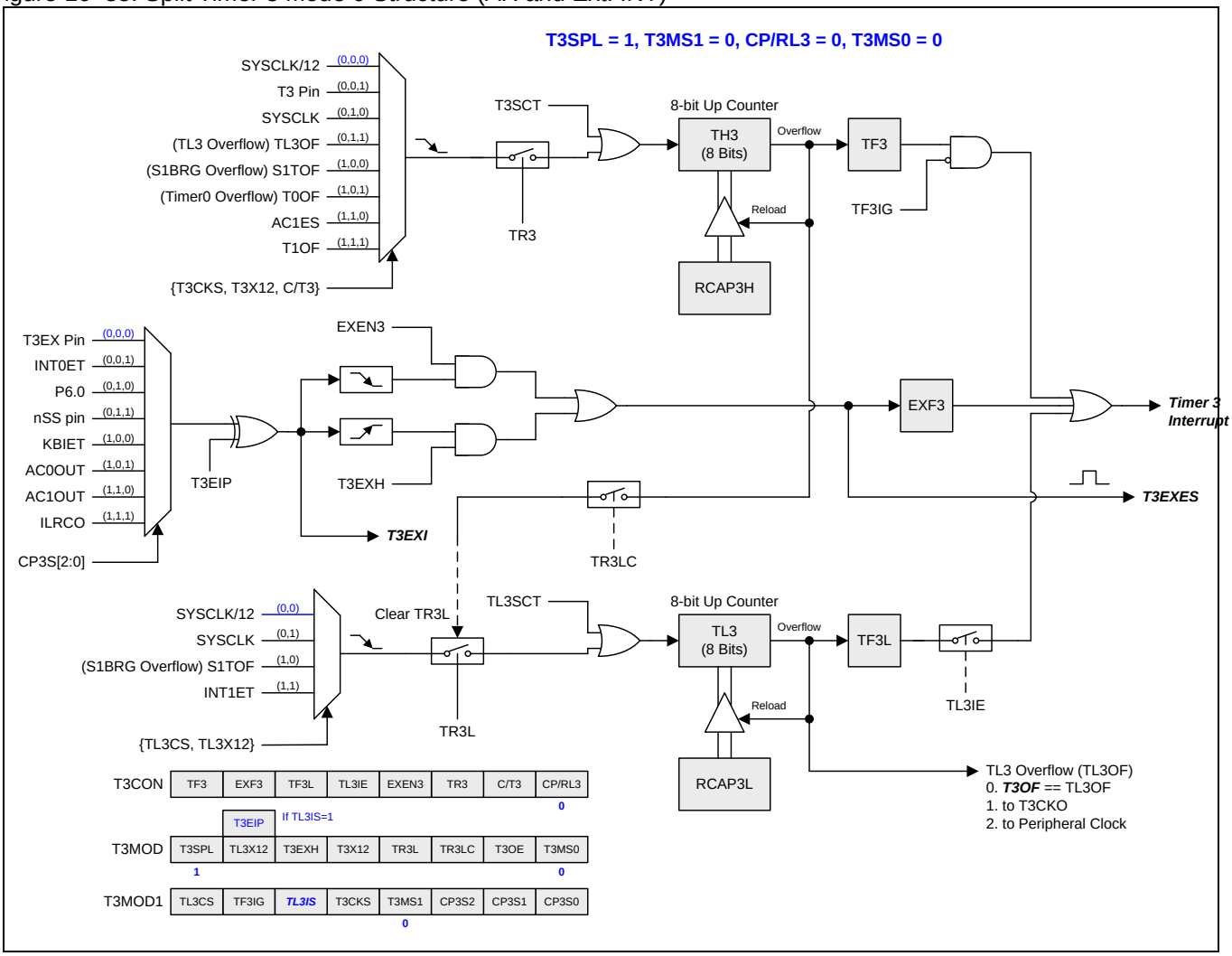
When T3SPL is set in this mode, Timer 3 operates as two 8-bit timers (TH3 and TL3), both 8-bit timers operate in up-counter as shown in Figure 16–35. TH3 holds the reload value for RCAP3H and keep the same 8 clock source inputs selection as 16-bit mode. It behaves the 8-bit function liked Timer 3 Mode 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 “0”, the bit on T3MOD.6 is the function of TL3X12. If TL3IS is “1”, the bit on T3MOD.6 is the function of T3EIP.

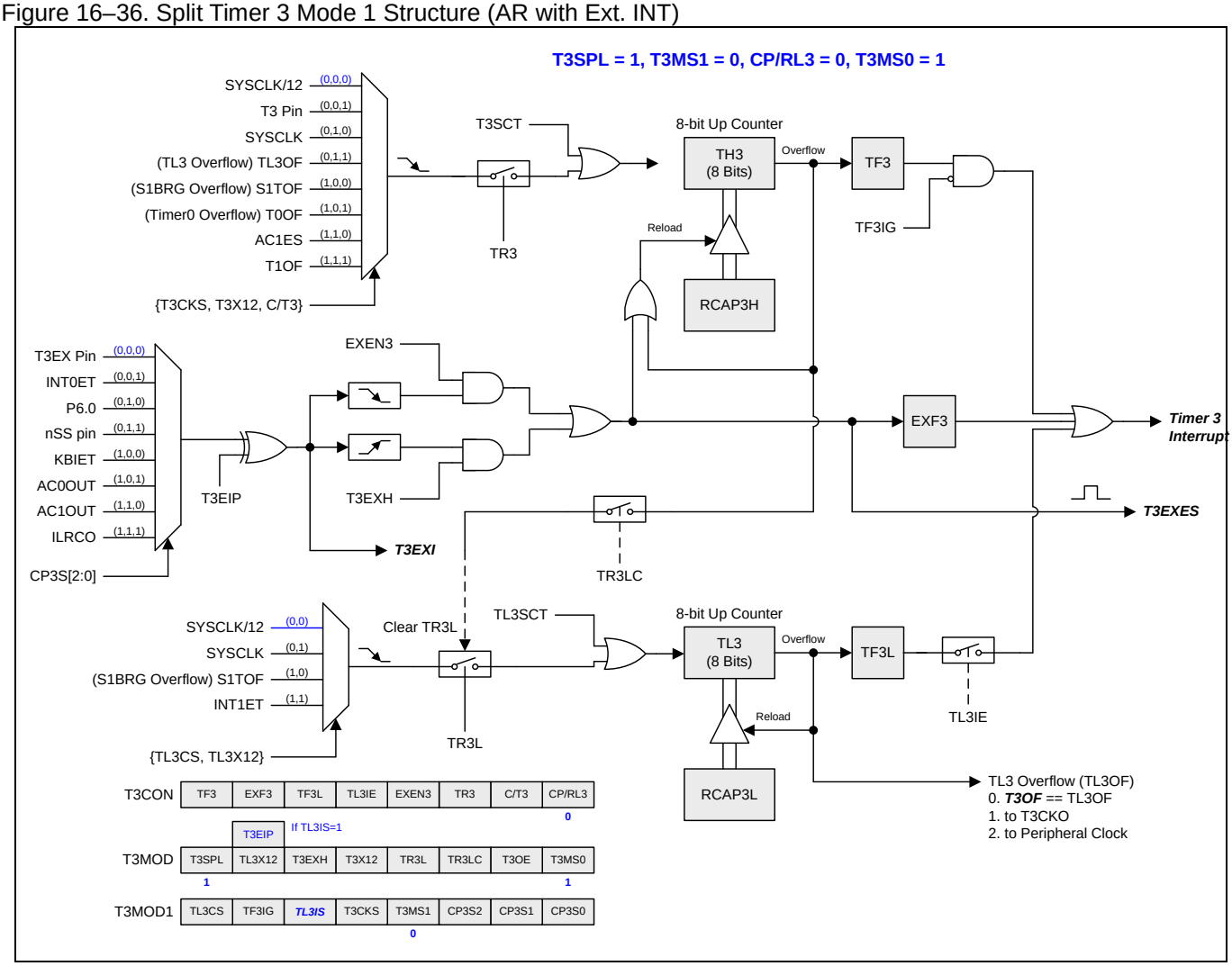
Figure 16–35. Split Timer 3 Mode 0 Structure (AR and Ext. INT)



16.3.7. Split Timer 3 Mode 1 (AR with Ext. INT)

When T3SPL is set in this mode, Timer 3 is split to two 8-bit timers as shown in Figure 16–36. It is similar function as Timer 3 Mode 1 and keeps the same interrupt scheme in Split Timer 3 Mode 0.

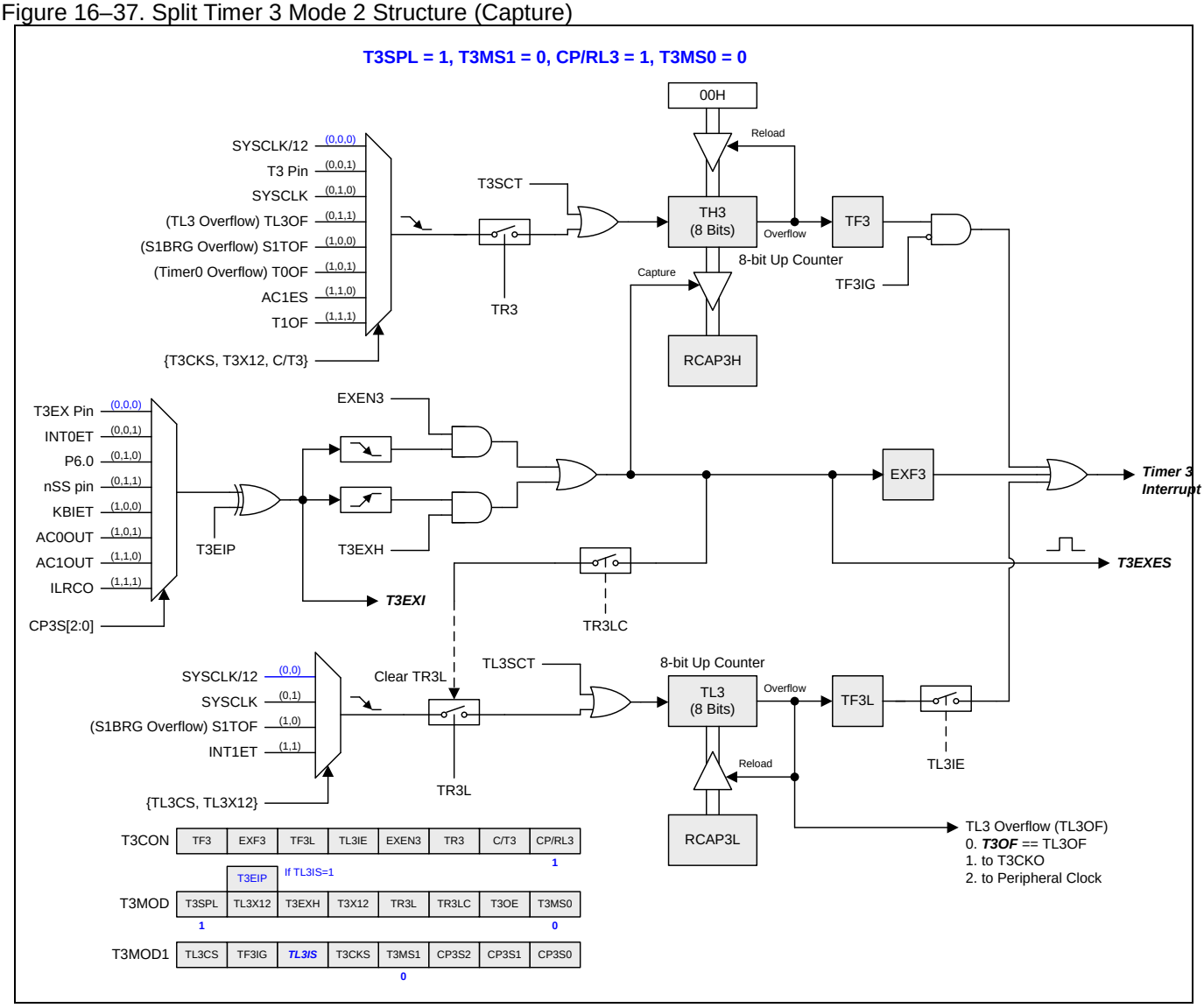
Figure 16–36. Split Timer 3 Mode 1 Structure (AR with Ext. INT)



16.3.8. 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 16–37. It is similar function as Timer 3 Mode 2 and keeps the same interrupt scheme in Split Timer 3 Mode 0.

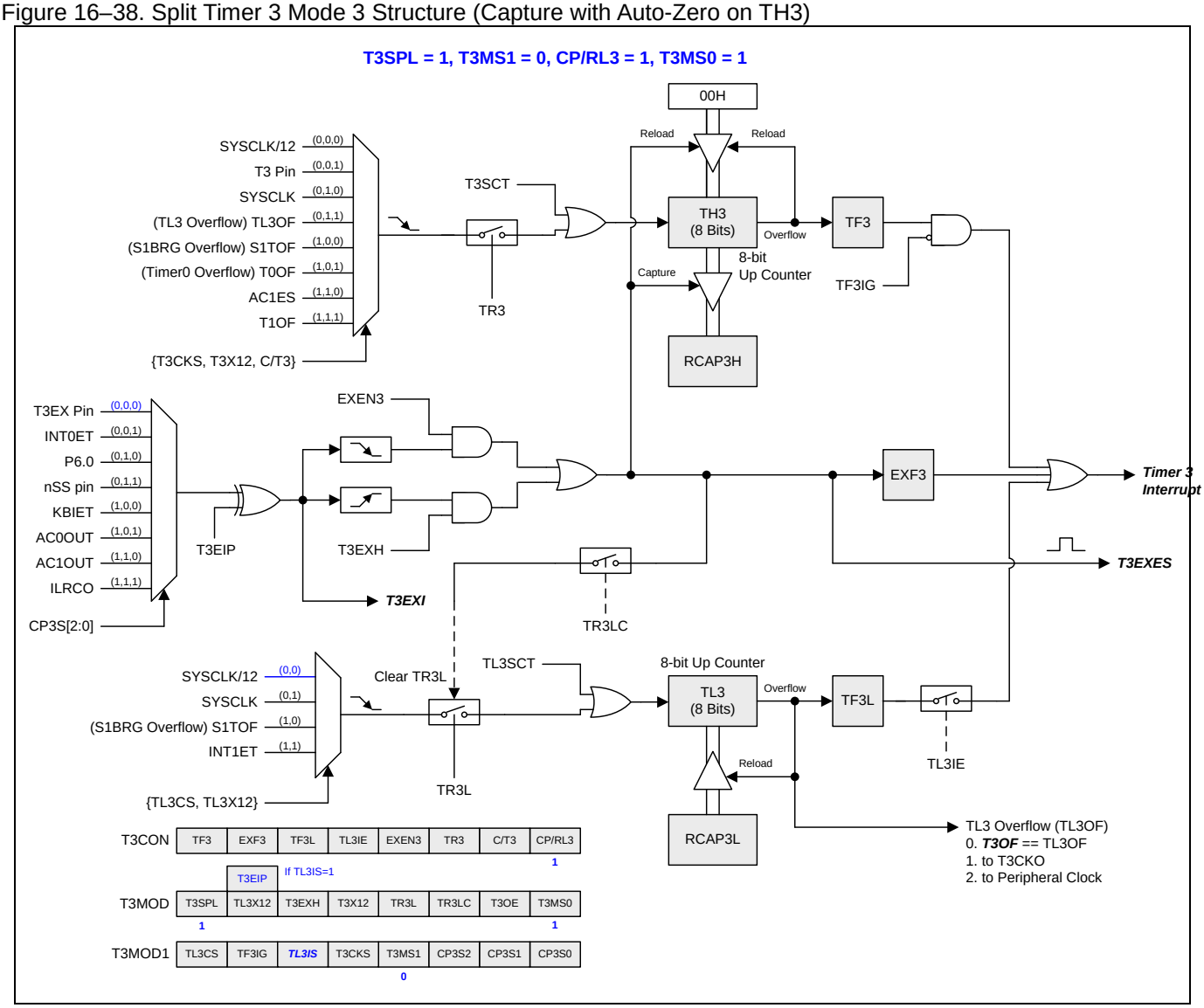
Figure 16–37. Split Timer 3 Mode 2 Structure (Capture)



16.3.9. 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 16–38. It is similar function as Timer 3 Mode 3 and keeps the same interrupt scheme in Split Timer 3 Mode 0.

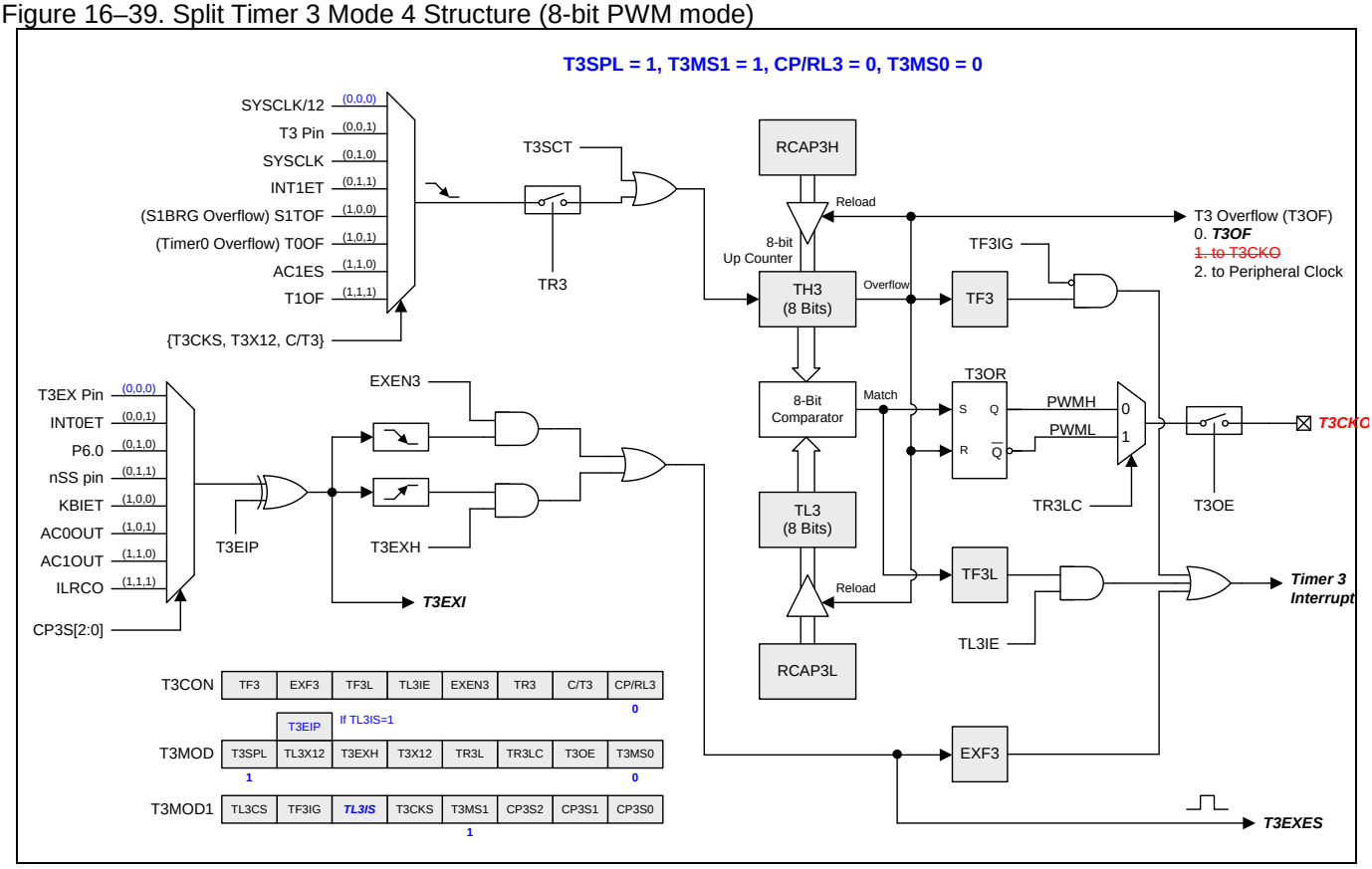
Figure 16–38. Split Timer 3 Mode 3 Structure (Capture with Auto-Zero on TH3)



16.3.10. Split Timer 3 Mode 4 (8-bit PWM Mode)

In this mode, Timer 3 is an 8-bit PWM mode as shown in Figure 16–39. 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.

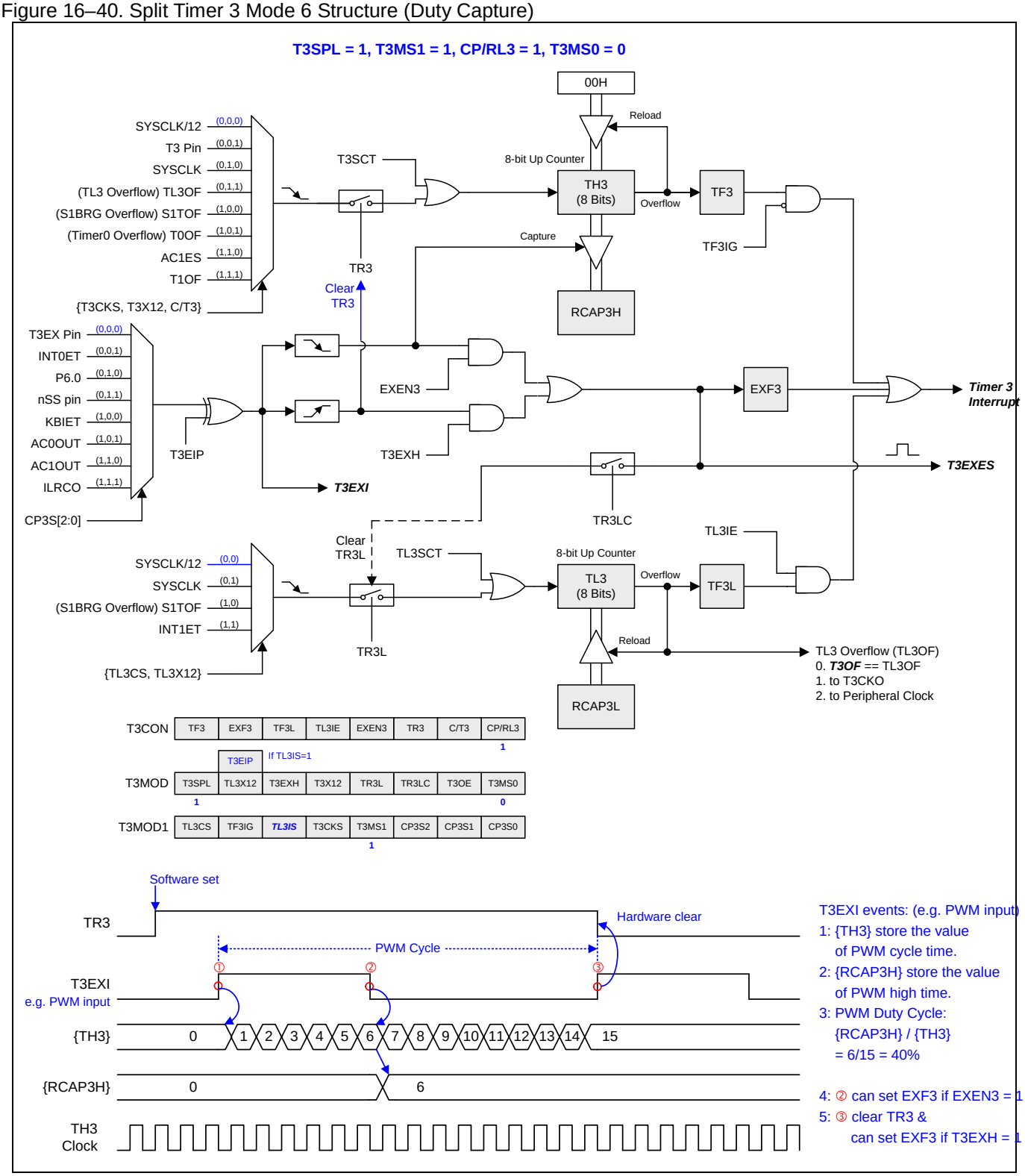
Figure 16–39. Split Timer 3 Mode 4 Structure (8-bit PWM mode)



16.3.11. Split Timer 3 Mode 6 (Duty Capture)

When T3SPL is set in this mode, Timer 3 is split to two 8-bit timers as shown in Figure 16–40. It is similar function as Timer 3 Mode 6 and keeps the same interrupt scheme in Split Timer 3 Mode 0.

Figure 16–40. Split Timer 3 Mode 6 Structure (Duty Capture)



16.3.12. Timer 3 Programmable Clock Output

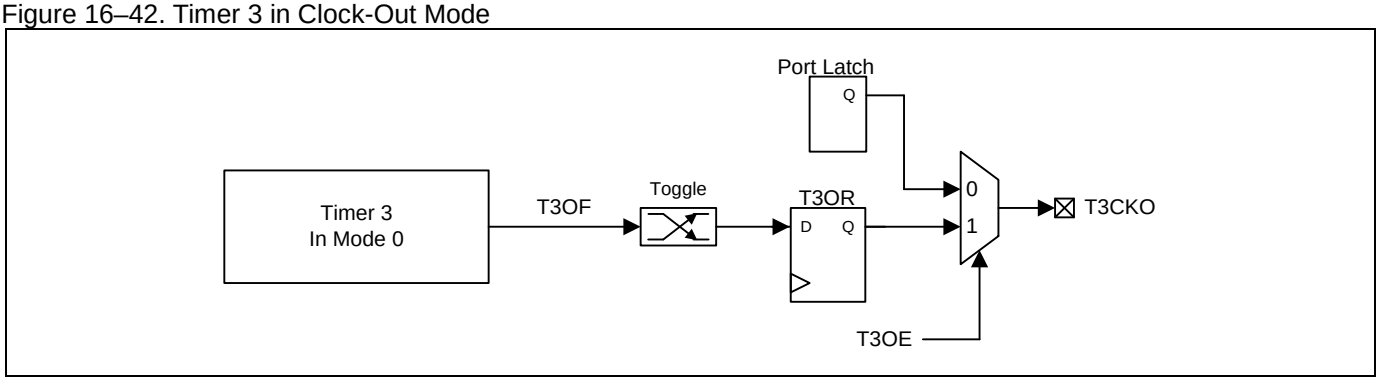
Timer 3 has a Clock-Out Mode (while CP/RL3=0 & T3OE=1). In this mode, Timer 3 operates as a programmable clock generator with 50% duty-cycle. The generated clocks come out on T3CKO port pin. The input clock (SYSCLK/12 or SYSCLK) increments the 16-bit timer (TH3, TL3). The timer repeatedly counts to overflow from a loaded value. Once overflows occur, the contents of (RCAP3H, RCAP3L) are loaded into (TH3, TL3) for the consecutive counting. Figure 16–41 gives the formula of Timer 3 clock-out frequency: Figure 16–42 shows the clock structure of Timer 3.

Figure 16–41. Timer 3 clock out equation

$$\text{T3 Clock-out Frequency} = \frac{\text{T3 Clock Frequency}}{2 \times (65536 - (\text{RCAP3H, RCAP3L}))}$$

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 SYSCLK=12MHz and select SYSCLK/12 as Timer 3 clock source, Timer 3 has a programmable output frequency range from 45.7Hz to 3MHz.
(3) For SYSCLK=12MHz and select SYSCLK as Timer 3 clock source, Timer 3 has a programmable output frequency range from 91.5Hz to 6MHz.

Figure 16–42. 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.

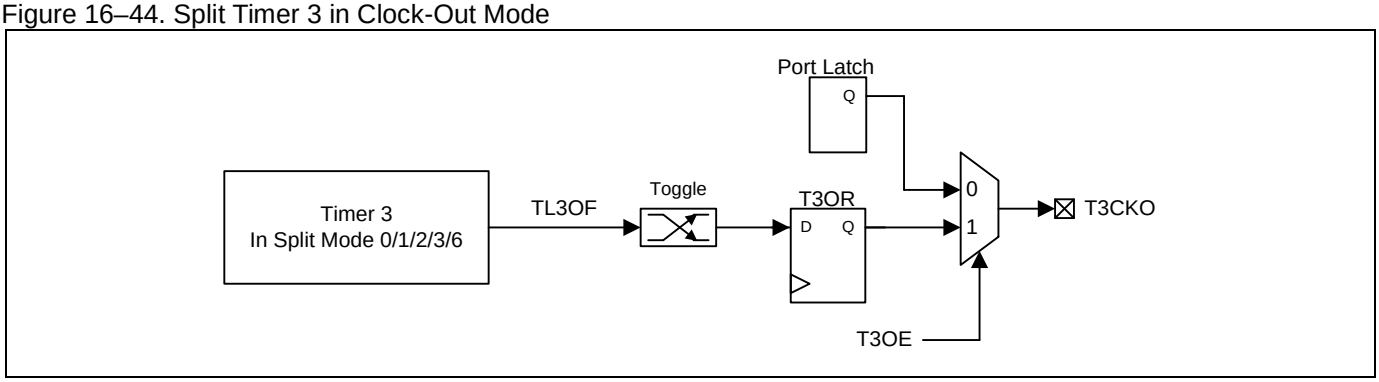
If Timer 3 in split mode, the clock output function is generated by TL3 overflow, and the output clock frequency is TL3 overflow rate /2. RCAP3L is the TL3's reload value when TL3 overflow. There are four clock source selections for TL3. Before enabling split Timer 3 clock output function, software must finish the TL3 clock source configuration. Figure 16–43 gives the formula of TL3 clock-out frequency: Figure 16–44 shows the clock structure of Split Timer 3.

Figure 16–43. Split Timer 3 clock out equation

$$\text{Split T3 Clock-out Frequency} = \frac{\text{TL3 Clock Frequency}}{3 \times (256 - \text{RCAP3L})}$$

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 SYSCLK=12MHz and select SYSCLK/12 as TL3 clock source, TL3 has a programmable output frequency range from 1.95KHz to 500KHz.
(3) For SYSCLK=12MHz and select SYSCLK as TL3 clock source, TL3 has a programmable output frequency range from 23.44Hz to 6MHz.

Figure 16–44. 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.

16.3.13. Timer 3 Register

T3CON: Timer 3 Control Register

SFR Page = **1 Only**

SFR Address = 0xC8 RESET = 0000-0000

7	6	5	4	3	2	1	0
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**

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 no 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	AC1ES	AC1ES
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 RESET = 0000-0000

7	6	5	4	3	2	1	0
T3SPL	TL3X12/ T3EIP	T3EXH	T3X12	TR3L	TR3LC	T3OE	T3MS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: T3SPL, Timer 3 split mode control.
0: Disable Timer 3 to split mode.
1: Enable Timer 3 to split mode.

TL3IS (T3MOD1.5) must be cleared to enable access to the TL3X12 bit.
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

TL3IS (T3MOD1.5) must be set to enable access to the T3EIP bit.
Bit 6: T3EIP, T3EXI input signal inversion control bit.
0: T3EXI input signal is not inverted.
1: T3EXI input signal is inverted.

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-to1 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	Mode 0: Auto-Reload and External Interrupt
0 0 1	Mode 1: Auto-Reload with External Interrupt
0 1 0	Mode 2: Capture mode
0 1 1	Mode 3: Capture with Auto-Zero
1 0 0	Mode 4: 8-bit PWM if T3SPL = 1
1 1 0	Mode 6: Duty Capture
Others	Reserved

T3MOD1: Timer 3 Mode Register 1

SFR Page = **2 Only**

SFR Address = 0x93 RESET = 0000-0000

7	6	5	4	3	2	1	0
TL3CS	TF3IG	TL3IS	T3CK2	T3MS1	CP3S2	CP3S1	CP3S0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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: TL3IS, TL3X12/T3EIP access control.
0: Enable TL3X12 access on T3MOD.6.
1: Enable T3EIP access on T3MOD.6.

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	T3EX Pin
0 0 1	INT0ET
0 1 0	P6.0 Pin
0 1 1	nSS Pin
1 0 0	KBIET
1 0 1	AC0OUT
1 1 0	AC1OUT
1 1 1	ILRCO

TL3: Timer 3 Low byte Register

SFR Page = **1 Only**

SFR Address = 0xCC RESET = 0000-0000

7	6	5	4	3	2	1	0
TL3.7	TL3.6	TL3.5	TL3.4	TL3.3	TL3.2	TL3.1	TL3.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

TH3: Timer 3 High byte Register

SFR Page = **1 Only**

SFR Address = 0xCD RESET = 0000-0000

7	6	5	4	3	2	1	0
TH3.7	TH3.6	TH3.5	TH3.4	TH3.3	TH3.2	TH3.1	TH3.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

RCAP3L: Timer 3 Capture Low byte Register

SFR Page = **1 Only**

SFR Address = 0xCA RESET = 0000-0000

7	6	5	4	3	2	1	0
RCAP3L.7	RCAP3L.6	RCAP3L.5	RCAP3L.4	RCAP3L.3	RCAP3L.2	RCAP3L.1	RCAP3L.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

RCAP3H: Timer 3 Capture High byte Register

SFR Page = **1 Only**

SFR Address = 0xCB RESET = 0000-0000

7	6	5	4	3	2	1	0
RCAP3H.7	RCAP3H.6	RCAP3H.5	RCAP3H.4	RCAP3H.3	RCAP3H.2	RCAP3H.1	RCAP3H.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

AUXR8: Auxiliary Register 8

SFR Page = 5 only

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	C1ICS1	C1ICS0	0	0	S1COPS	T3PS1	T3PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 1~0: T3PS1~0, Timer 3 Port pin Selection [1:0].

T3PS1~0	T3/T3CKO	T3EX
0 0	P3.3	P3.4
0 1	P3.3	P3.2
1 0	P0.2	P0.1
1 1	P6.0	P6.1

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 = 0000-0000

7	6	5	4	3	2	1	0
TR2E	TR3E	0	0	0	0	C1CR	PTMCR
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7, TR2E, write “1” on this bit to set TR2 enabled (TR2=1) of Timer 2. This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

Bit 6, TR3E, write “1” on this bit to set TR3 enabled (TR3=1) of Timer 3. This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

Bit 5~2: Reserved. Software must write “**0**” on this bit when TREN0 is written.

Bit 1, C1CR, write “1” on this bit to set C1R enabled (C1R=1) of PCA1. This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

Bit 0, PTMCRE, write “1” on this bit to set CR enabled (CR=1) of PTM0. 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

7	6	5	4	3	2	1	0
T2RLC	T3RLC	0	0	0	0	C1RLC	PTMRLC
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7, T2RLC, write “1” on this bit to force T2 reload. This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

Bit 6, T3RLC, write “1” on this bit to force T3 reload. This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

Bit 5~2: Reserved. Software must write “**0**” on this bit when TRLC0 is written.

Bit 1, C1RLC, write “1” on this bit to force PCA1 reload. This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

Bit 0, PTMRLC, write “1” on this bit to force PTM0 reload. 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

7	6	5	4	3	2	1	0
T2SC	T3SC	0	0	0	0	C1SC	PTMSC
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7, T2SC, write “1” on this bit to set TR2 disabled (TR2=0) of Timer 2. This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

Bit 6, T3SC, write “1” on this bit to set TR3 disabled (TR3=0) of Timer 3. This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

Bit 5~2: Reserved. Software must write “**0**” on this bit when TSPC0 is written.

Bit 1, C1SC, write “1” on this bit to set C1R disabled (C1R=0) of PCA1. This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

Bit 0, PTMSC, write “1” on this bit to set CR disabled (CR0=0) of PTM0. This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

17. PWM Timer (PTM0)

The **MG82F6P32** is equipped with one programmable Counter Array (PCA1) and one PWM Timer (PTM0) module, which provides more timing capabilities with less CPU intervention than the standard timer/counters. Its advantages include reduced software overhead and improved accuracy.

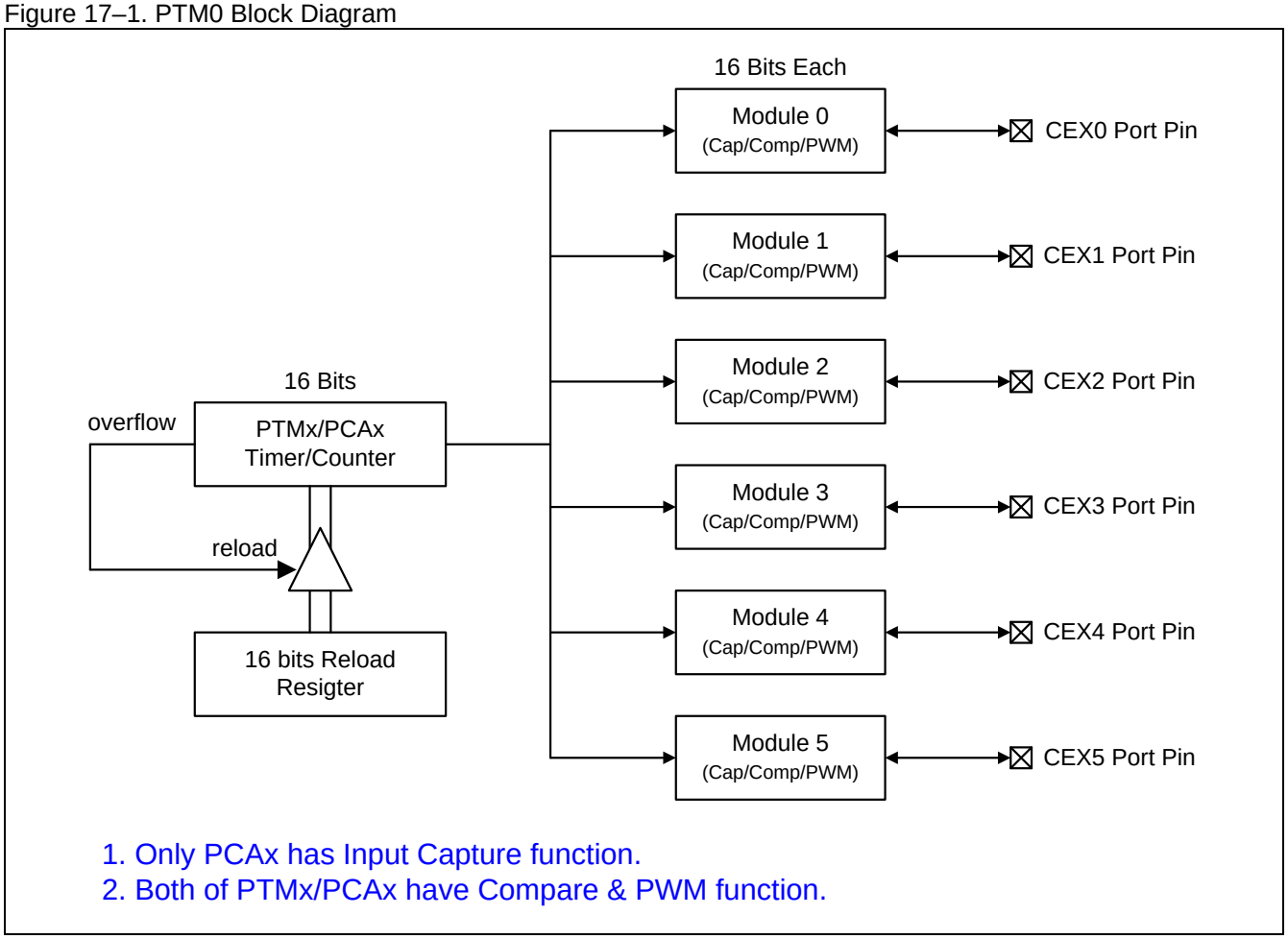
17.1. PTM0 Overview

The PTM0 consists of a dedicated timer/counter which serves as the time base for an array of **6** compare/PWM modules. Figure 17–1 shows a block diagram of the PTM0. Notice that the PTM0 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.

PTM0 Module 0~5 can be programmed in any one of the following modes:
- Software Timer (Compare)
- High Speed Output (Compare Output)
- Phase Variable Compare Output
- Pulse Width Modulator Output (PWM)

All of these modes will be discussed later in detail. However, let's first look at how to set up the PTM0 timer and modules.

Figure 17–1. PTM0 Block Diagram



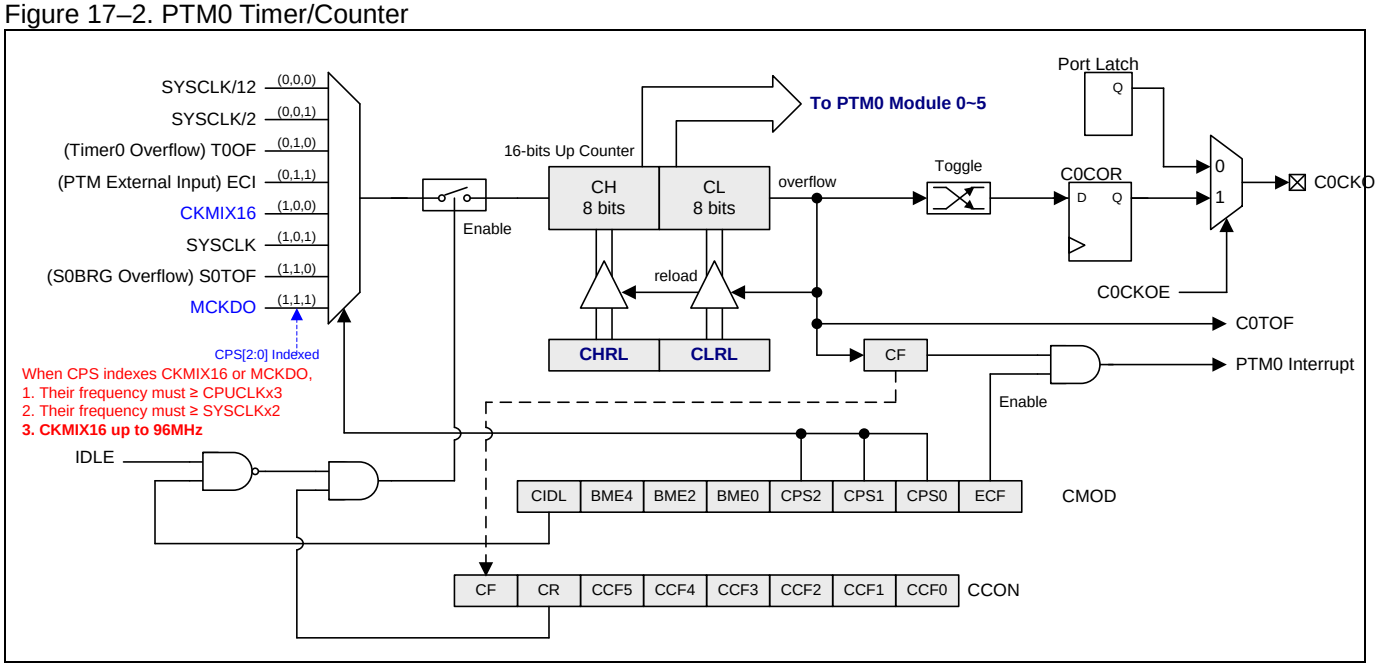
17.2. PTM0 Timer/Counter

The timer/counter for the PTM0 is 16-bit auto-reload 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 PTM0 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 PTM0 timer input. When CPS[2:0] indexes CKMIX16 or MCKDO, the frequency of the input clock sources must ≥ CPUCLK x3 and ≥ SYSCLK x2.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 PTM0 timer during Idle Mode by setting the Counter Idle bit (CIDL). This can further reduce power consumption during Idle mode.

Figure 17–2. PTM0 Timer/Counter



CMOD: PTM0 Counter Mode Register

SFR Page = 0 Only

SFR Address = 0xD9 RESET = 0000-0000

7	6	5	4	3	2	1	0
CIDL	BME4	BME2	BME0	CPS2	CPS1	CPS0	ECF
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: CIDL, PTM0 counter Idle control.
0: Lets the PTM0 counter continue functioning during Idle mode.
1: Lets the PTM0 counter be gated off during Idle mode.

Bit 6: BME4, Buffer Mode Enable on PTM0 module 4/5. It is only valid on both of PTM0 module 4 and module 5 in capture mode, PWM mode or COPM mode.
0: PTM0 Module 4/5 buffer mode disabled.
1: PTM0 Module 4/5 buffer mode enabled.

Bit 5: BME2, Buffer Mode Enable on PTM0 module 2/3. It is only valid on both of PTM0 module 2 and module 3 in capture mode, PWM mode or COPM mode.
0: PTM0 Module 2/3 buffer mode disabled.
1: PTM0 Module 2/3 buffer mode enabled.

Bit 4: BME0, Buffer Mode Enable on PTM0 module 0/1. It is only valid on both of PTM0 module 0 and module 1 in capture mode, PWM mode or COPM mode.
0: PTM0 Module 0/1 buffer mode disabled.
1: PTM0 Module 0/1 buffer mode enabled.

Bit 3~1: CPS2-CPS0, PCA counter clock source select bits.

CPS2	CPS1	CPS0	PTM0 Clock Source
0	0	0	Internal clock, (system clock)/12
0	0	1	Internal clock, (system clock)/2
0	1	0	Timer 0 overflow
0	1	1	External clock at the ECI pin
1	0	0	CKMIX16 output
1	0	1	Internal clock, (system clock)/1
1	1	0	S0BRT overflow
1	1	1	MCK Divider Output, MCKDO

Note: When CPS indexes CKMIX16 or MCKDO, needs to follow the conditions:

1. *The source frequency must ≥ CPUCLK x3.*
2. *The source frequency must ≥SYSCLK x2.*

Bit 0: ECF, Enable PTM0 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 PTM0 and the flags for the PTM0 timer and each module. To run the PTM0 the CR bit (CCON.6) must be set by software. The PTM0 will be shut off by clearing this bit. The CF bit (CCON.7) is set when the PTM0 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 a match occurs. These flags also can only be cleared by software. The PTM0 interrupt system is shown Figure 17–3.

CCON: PTM0 Counter Control Register

SFR Page = 0 only
SFR Address = 0xD8 RESET = 0000-0000

7	6	5	4	3	2	1	0
CF	CR	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: CF, PTM0 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, PTM0 Counter Run control bit.
0: Must be cleared by software to turn the PTM0 counter off.
1: Set by software to turn the PTM0 counter on.

Bit 5: CCF5, PTM0 Module 5 interrupt flag.
0: Must be cleared by software.
1: Set by hardware when a match occurs.

Bit 4: CCF4, PTM0 Module 4 interrupt flag.

0: Must be cleared by software.
1: Set by hardware when a match occurs.

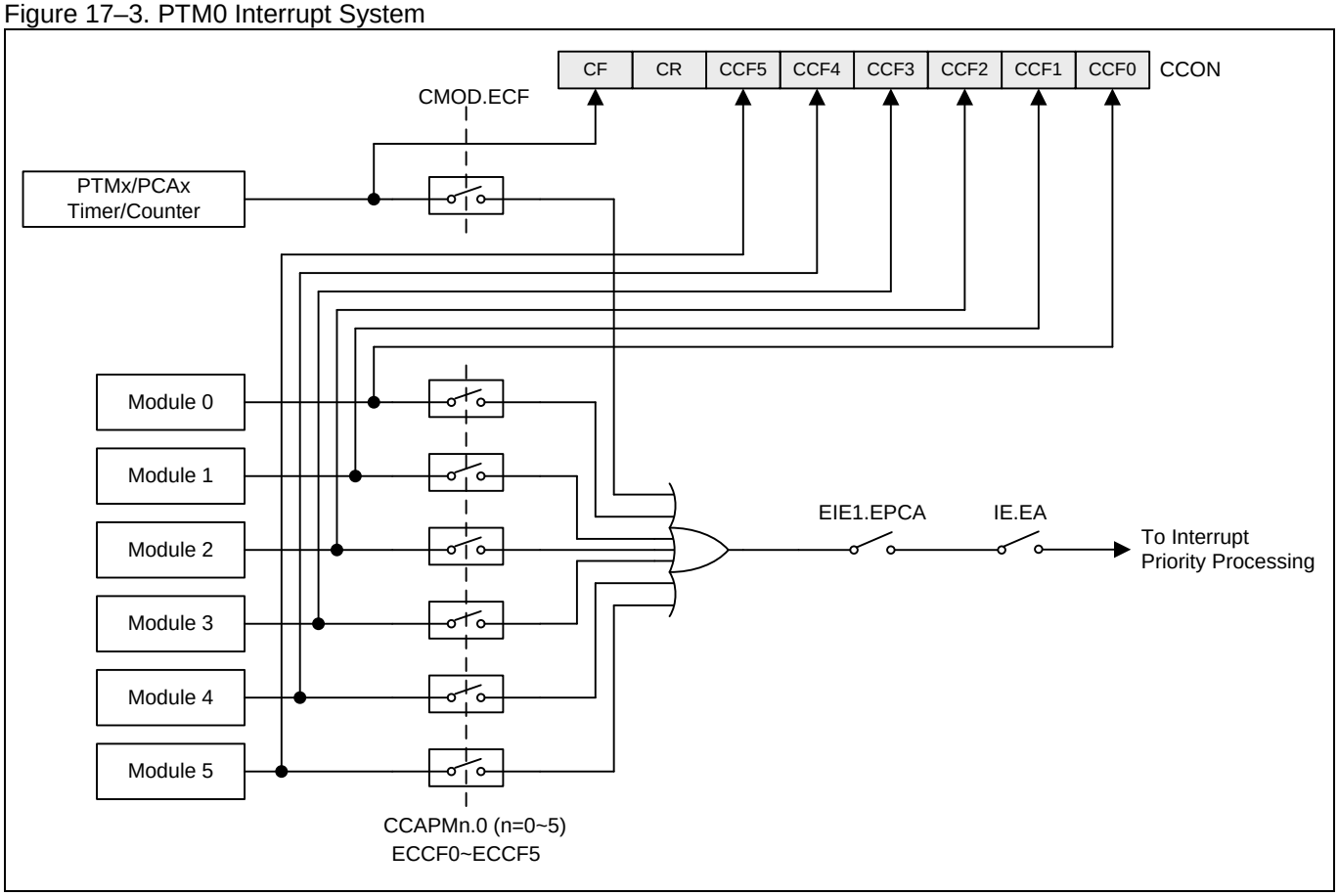
Bit 3: CCF3, PTM0 Module 3 interrupt flag.
0: Must be cleared by software.
1: Set by hardware when a match occurs.

Bit 2: CCF2, PTM0 Module 2 interrupt flag.
0: Must be cleared by software.
1: Set by hardware when a match occurs.

Bit 1: CCF1, PTM0 Module 1 interrupt flag.
0: Must be cleared by software.
1: Set by hardware when a match occurs.

Bit 0: CCF0, PTM0 Module 0 interrupt flag.
0: Must be cleared by software.
1: Set by hardware when a match occurs.

Figure 17–3. PTM0 Interrupt System



CH: PTM0 base timer High

SFR Page = **0 only**

SFR Address = 0xF9 RESET = 0000-0000

7	6	5	4	3	2	1	0
CH.7	CH.6	CH.5	CH.4	CH.3	CH.2	CH.1	CH.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

CL: PTM0 base timer Low

SFR Page = **0 only**

SFR Address = 0xE9 RESET = 0000-0000

7	6	5	4	3	2	1	0
CL.7	CL.6	CL.5	CL.4	CL.3	CL.2	CL.1	CL.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

CHRL: PTM0 CH Reload Register

SFR Page = **0 only**

SFR Address = 0xCF RESET = 0000-0000

7	6	5	4	3	2	1	0
CHRL.7	CHRL.6	CHRL.5	CHRL.4	CHRL.3	CHRL.2	CHRL.1	CHRL.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~0: CHRL, reload value of CH.

CLRL: PTM0 CL Reload Register

SFR Page = **0 only**

SFR Address = 0xCE RESET = 0000-0000

7	6	5	4	3	2	1	0
CLRL.7	CLRL.6	CLRL.5	CLRL.4	CLRL.3	CLRL.2	CLRL.1	CLRL.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~0: CLRL, reload value of CL.

17.3. Compare Modules

Each of the compare module 0~5 has a mode register called CCAPMn (n = 0,1,2,3,4 or 5) 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: PTM Module Compare Register, n=0~5

SFR Page = 0 only

SFR Address = 0xDA~0xDF RESET = 0000-0000

7	6	5	4	3	2	1	0
DTEn	ECOMn	0	0	MATn	TOGn	PWMn	ECCFn
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: DTEn. Enable Dead-Time control on PWMHn/ PWMLn 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 PWMn output.
1: Enable the Dead-Time control on PWMn output.

Bit 6: ECOMn, Enable Comparator.
0: Disable the digital comparator function.
1: Enables the digital comparator function.

Bit 3: MATn, Match control.
0: Disable the digital comparator match event to set CCFn.
1: A match of the PTM counter with this module's compare/capture register causes the CCFn bit in CCON to be set.

Bit 2: TOGn, Toggle control.
0: Disable the digital comparator match event to toggle CEXn.
1: A match of the PTM counter with this module's compare/capture register causes the CEXn pin to toggle.

Bit 1: PWMn, PWM control.
0: Disable the PWM mode in PTM module.
1: Enable the PWM function and cause CEXn pin to be used as a pulse width modulated output.

Bit 0: ECCFn, Enable CCFn interrupt.
0: Disable compare/capture flag CCFn in the CCON register to generate an interrupt.
1: Enable compare/capture flag CCFn in the CCON register to generate an interrupt.

Each module also has a pair of 8-bit compare 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 17.4.8 and 17.4.9.

CCAPnH: PTM0 Module n Capture High Register, n=0~5

SFR Page = 0 only

SFR Address = 0xFA~0xFF RESET = 0000-0000

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

CCAPnL: PTM0 Module n Capture Low Register, n=0~5

SFR Page = 0 only

SFR Address = 0xEA~0xEF RESET = 0000-0000

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

PCAPWMn: PWM Mode Auxiliary Register, n=0~5

SFR Page = 0 only

SFR Address = 0xF2~0xF7 RESET = 0000-0000

7	6	5	4	3	2	1	0
PnRS1	PnRS0	0	0	0	PnINV	ECAPnH	ECAPnL
R/W	**R/W**	**W**	**W**	**R/W**	**R/W**	**R/W**	**R/W**

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-1111-1111-111 → 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 5~3: Reserved. Software must write “0” on these bits when PCAPWMn is written.

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 PCAx/PTMx

Table 17–1 shows the CCAPMn register settings for the various PTM0 functions.

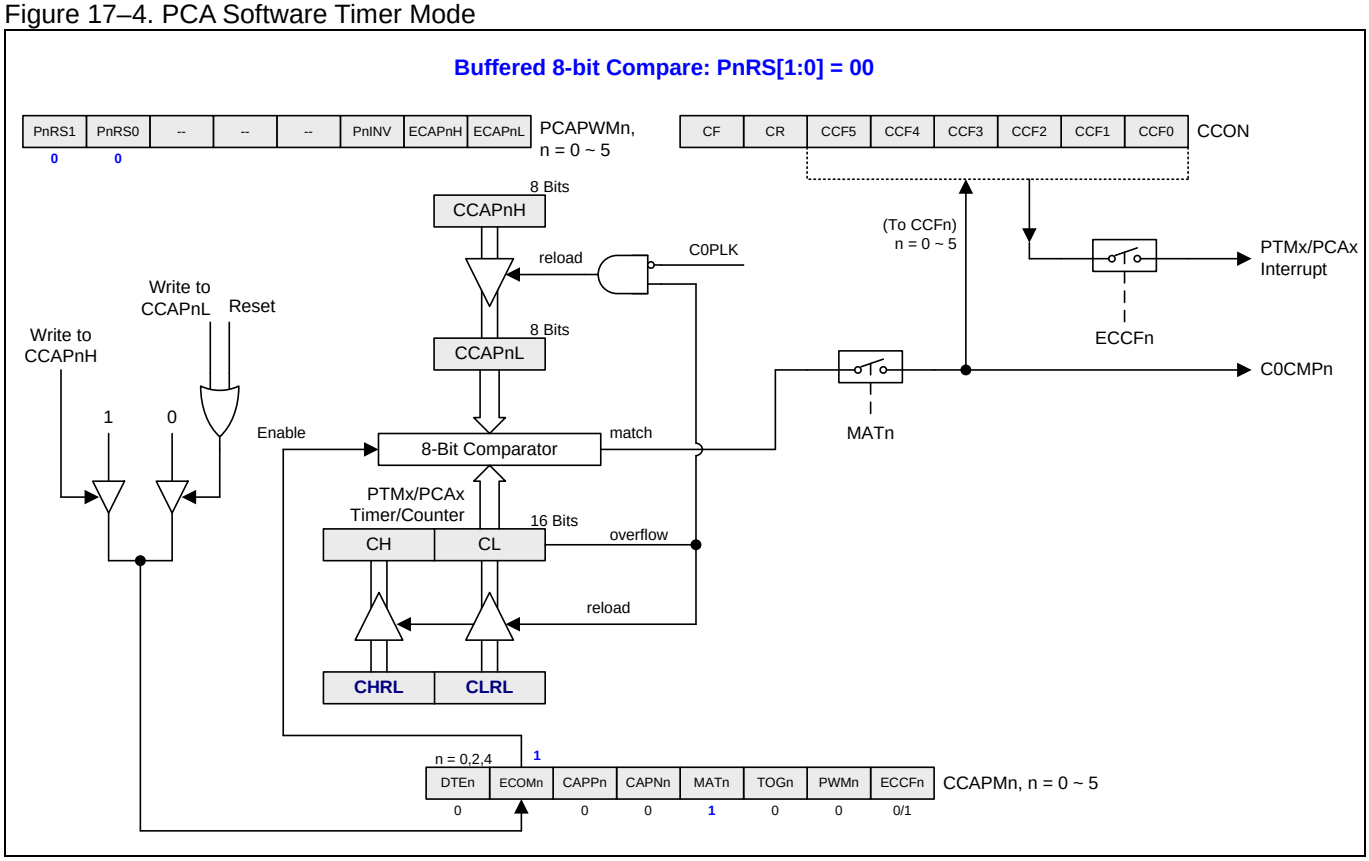
Table 17–1. PCAx/PTMx Module Modes

ECOMn	MATn	TOGn	PWMn	BMEn	Module Function
0	0	0	0	0	No operation
1	1	0	0	0	Software Timer (Compare)
1	1	0	0	1	Buffered Software Timer (Compare)
1	1	1	0	0	High Speed Output (Output Compare)
1	1	1	0	1	Phase Variable Compare Output
1	0/1	0	1	0	Pulse Width Modulator (PWM)
1	0/1	0	1	1	Buffered 10/12/16-bit PWM

17.4.1. 8-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 17–4. PCA Software Timer Mode

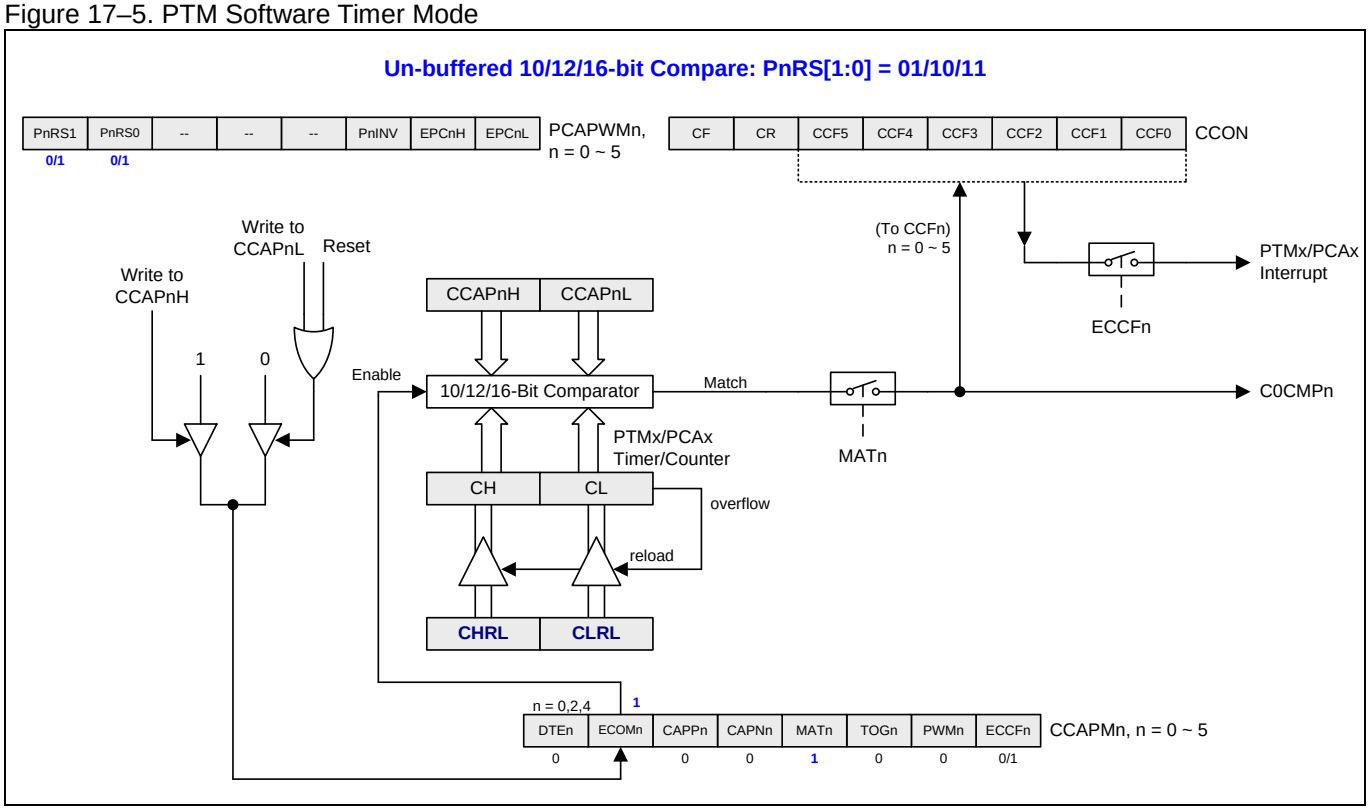


17.4.2. 16-bit Software Timer Mode (Compare mode)

The PTM modules can be used as software timers by setting both the ECOM and MAT bits in the module's CCAPMn register. The PTM 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.

Due to CL counter will be changed more frequently, and it may cause mismatch when writing the compare value into CCAPnL. To prevent the mismatch when writing CCAPnL, it will automatically set ECOMn to 0 to disable comparator after write command. And the ECOMn will be resumed to 1 after the write command to CCAPnH. If the application only changes the CCAPnL, it should set the ECOMn to 1 after write to CCAPnL to resume the comparator

Figure 17–5. PTM Software Timer Mode



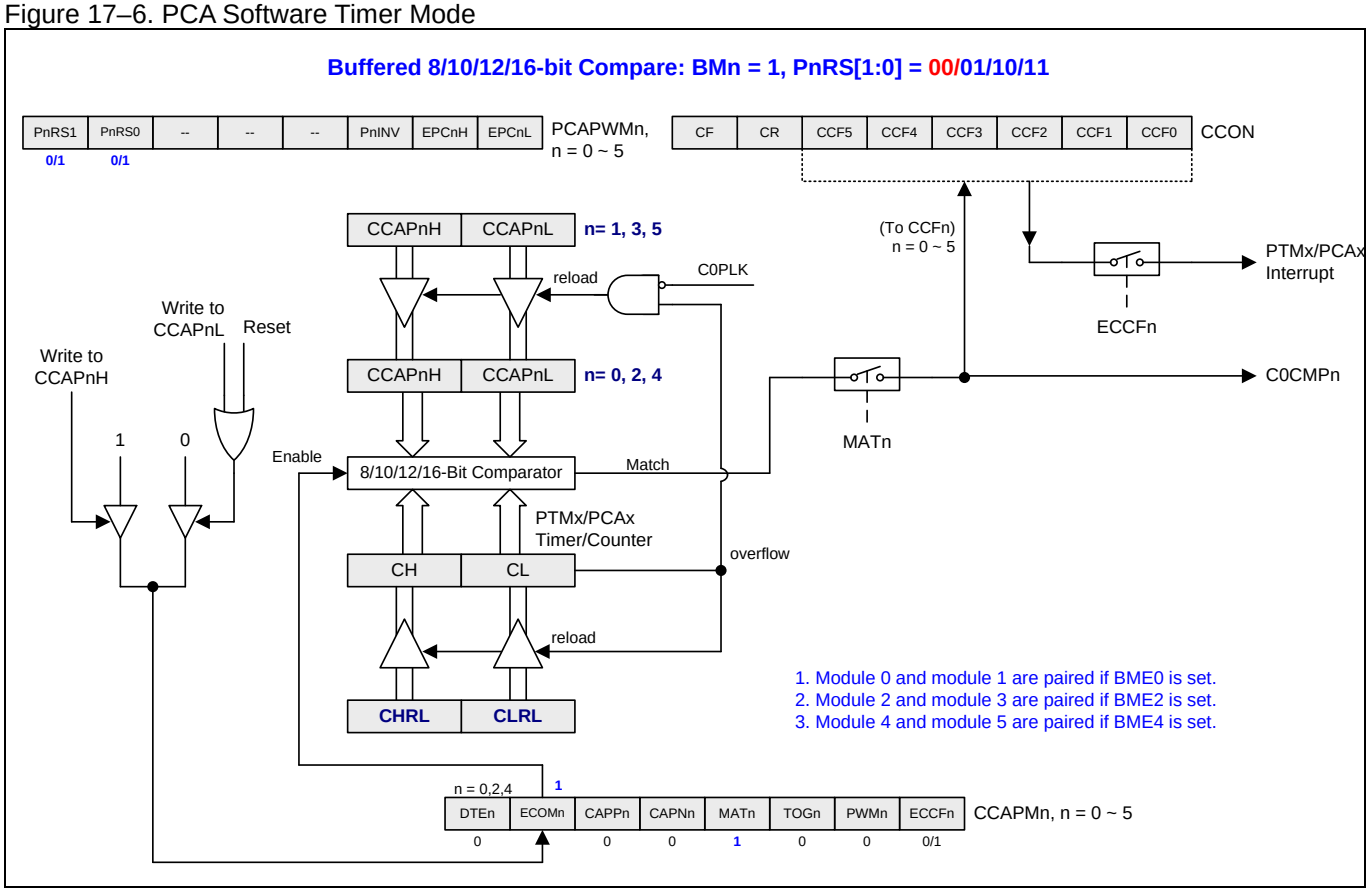
17.4.3. Buffered 16-bit Software Timer Mode (Compare mode)

To avoid the mismatch during CCAPnL or CCAPnH has been changed, **MG82F6P32** provide buffered mode to prevent it.
The CCAPnL and CCAPnH of odd modules (n=1, 3 and 5) has been used as buffer, user can write the new data into CCAPnH, CCAPnL (n=1, 3 and 5) and it will be reloaded into CCAPnH, CCAPnL (n=0, 2 and 4) after CH, CL overflow.
Or user can set C0PLK to reload the new code manually.

Due to the odd buffers have been used as buffer, so it should only use even MATn for match detection.

When use buffered mode to write data into buffer CCAPnL (n= 1, 3 and 5) the auto inhibition function will be disabled (It means the ECOMn will not be auto set to 1.) But if user still wants to access CCAPnL (n = 0, 2 and 4) the auto inhibition function still exists. The auto inhibition function please reference 17.4.2

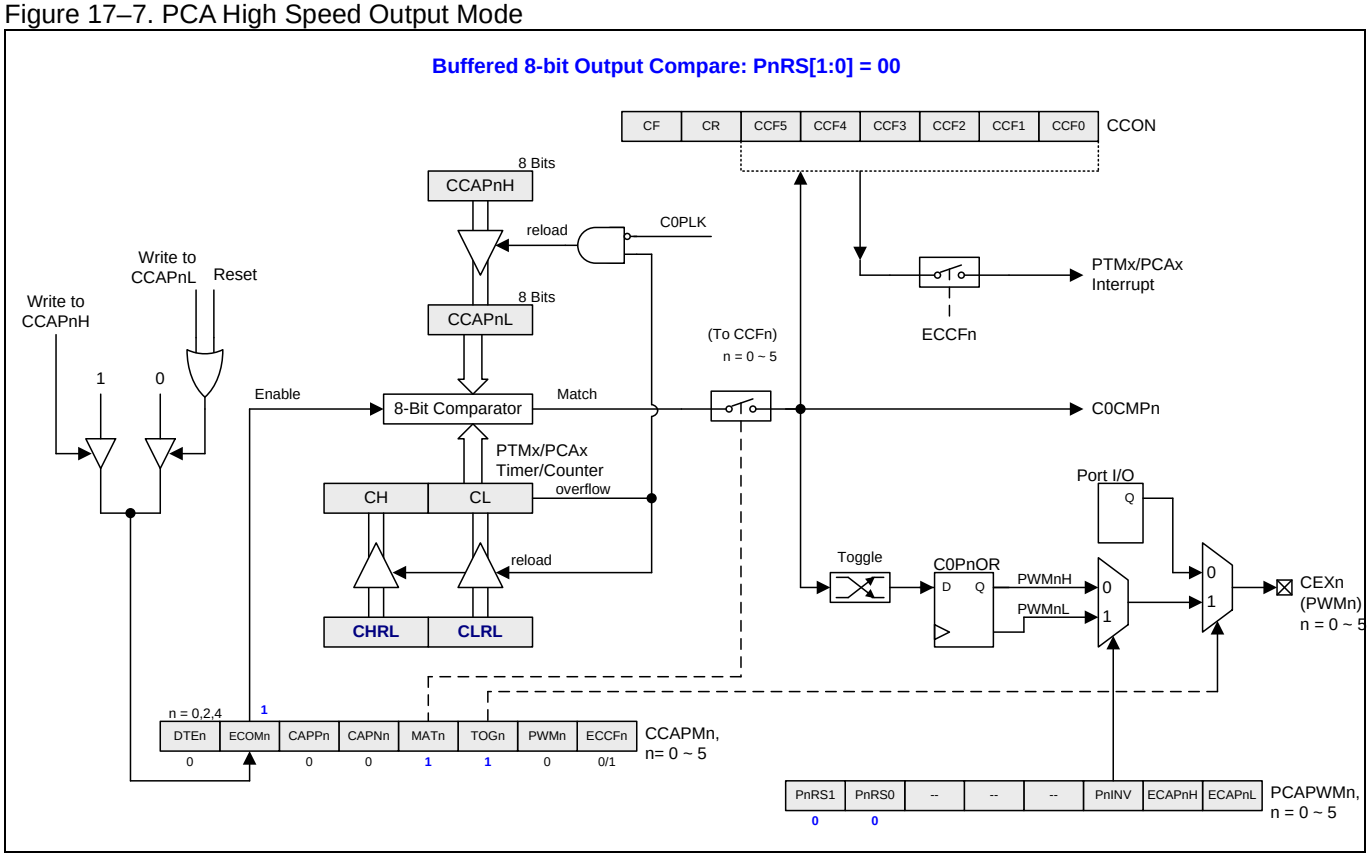
Figure 17–6. PCA Software Timer Mode



17.4.4. 8-bit High Speed Output Mode (Output Compare 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.

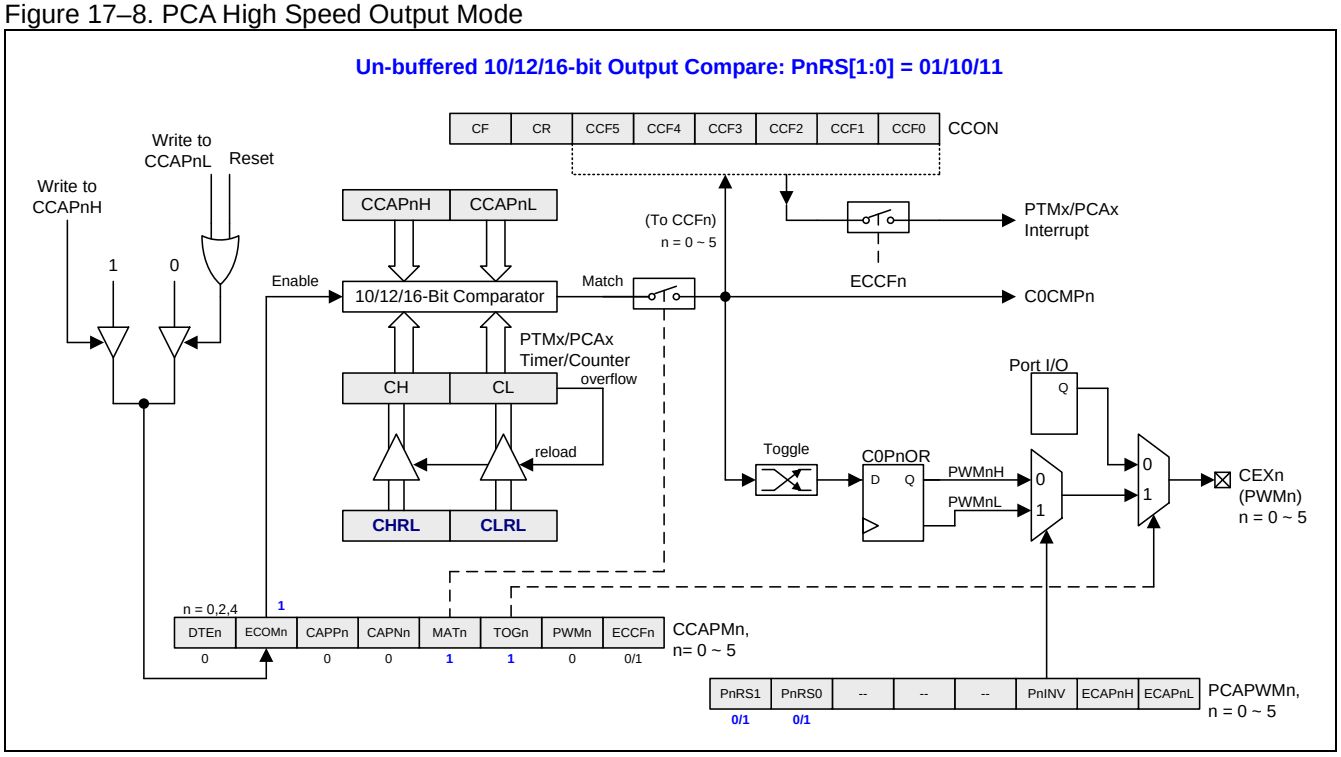
Figure 17–7. PCA High Speed Output Mode



17.4.5. 16-bit High Speed Output Mode (Output Compare 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.

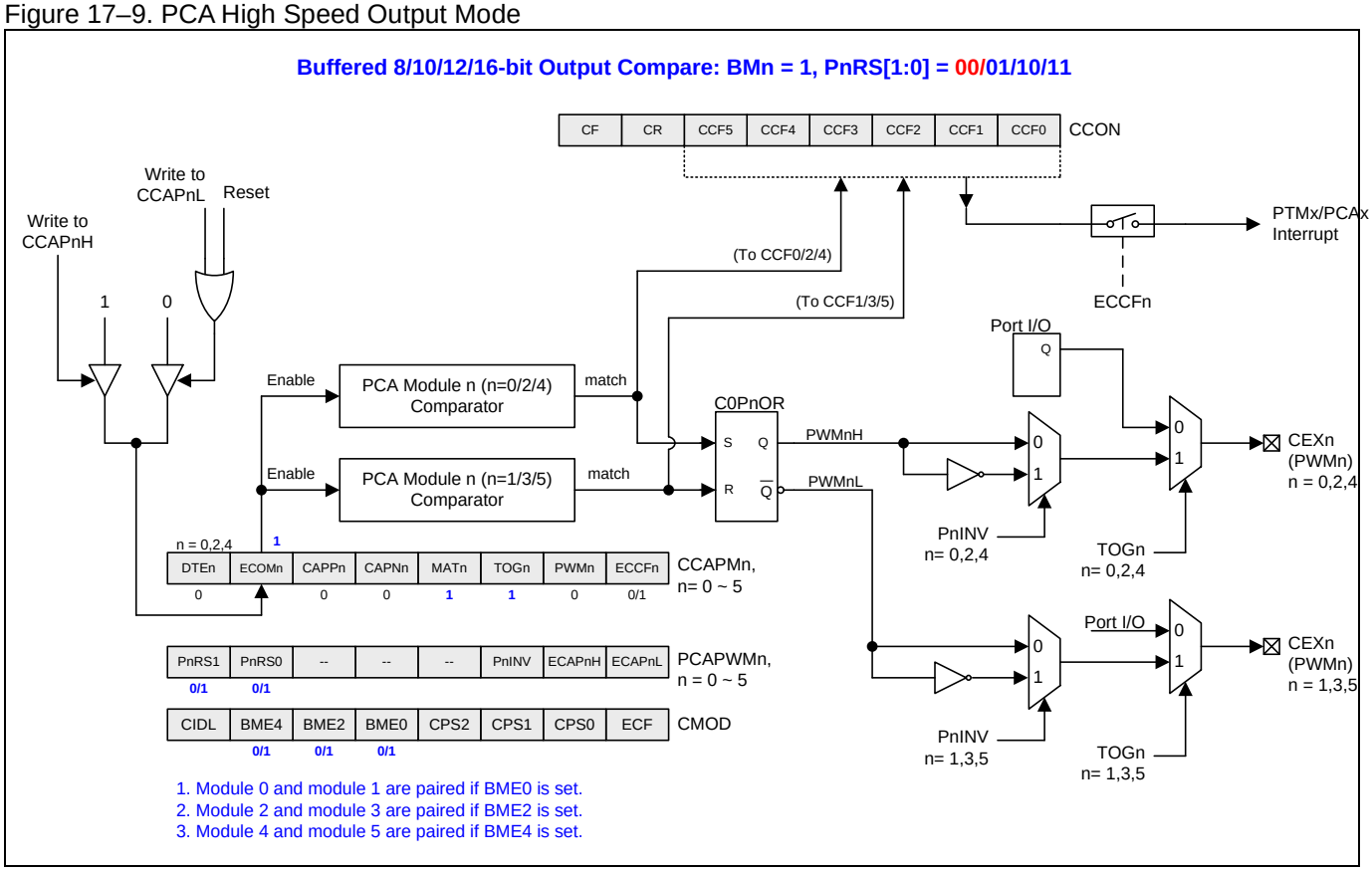
Figure 17–8. PCA High Speed Output Mode



17.4.6. Phase Variable 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 17–9. PCA High Speed Output Mode



17.4.7. Buffered 8-bit PWM Mode

All of the PTM0 modules can be used as PWM outputs. The frequency of the output depends on the clock source for the PTM0 timer. All of the modules will have the same frequency of output because they all share the PTM 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.

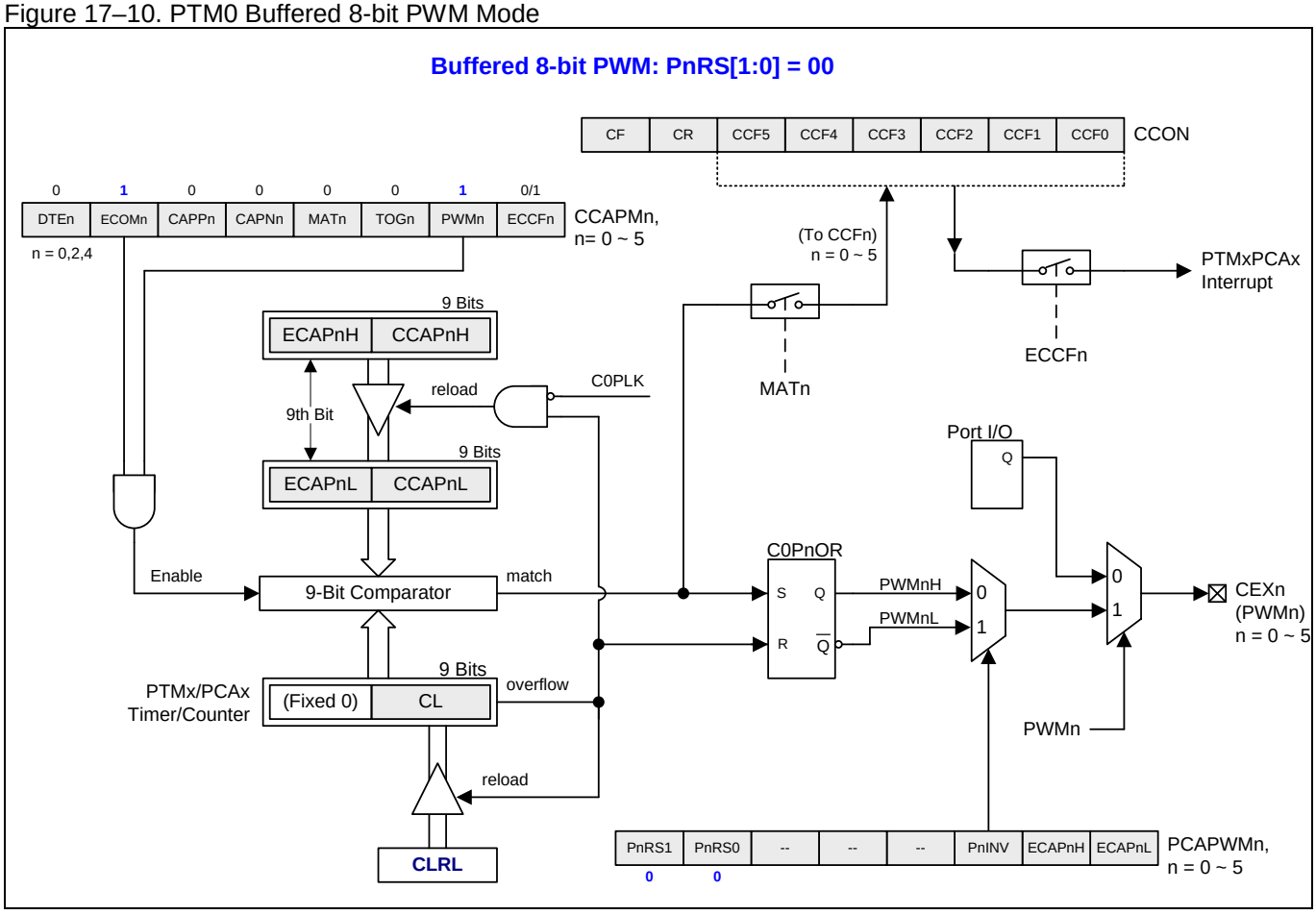
Using the 9-bit comparison, the duty cycle of the output can be improved to really start from 0%, and up to 100%. The formula for the duty cycle is:

Duty Cycle = 1 – { ECAPnH, [CCAPnH] } / 256.

Where, *[CCAPnH]* is the 8-bit value of the CCAPnH register, and *ECAPnH* (bit-1 in the PTMPWMn register) is 1-bit value. So, *{ ECAPnH, [CCAPnH] }* forms a 9-bit value for the 9-bit comparator.

For examples,
a. If ECAPnH=0 & CCAPnH=0x00 (i.e., 0x000), the duty cycle is 100%.
b. If ECAPnH=0 & CCAPnH=0x40 (i.e., 0x040) the duty cycle is 75%.
c. If ECAPnH=0 & CCAPnH=0xC0 (i.e., 0x0C0), the duty cycle is 25%.
d. If ECAPnH=1 & CCAPnH=0x00 (i.e., 0x100), the duty cycle is 0%.

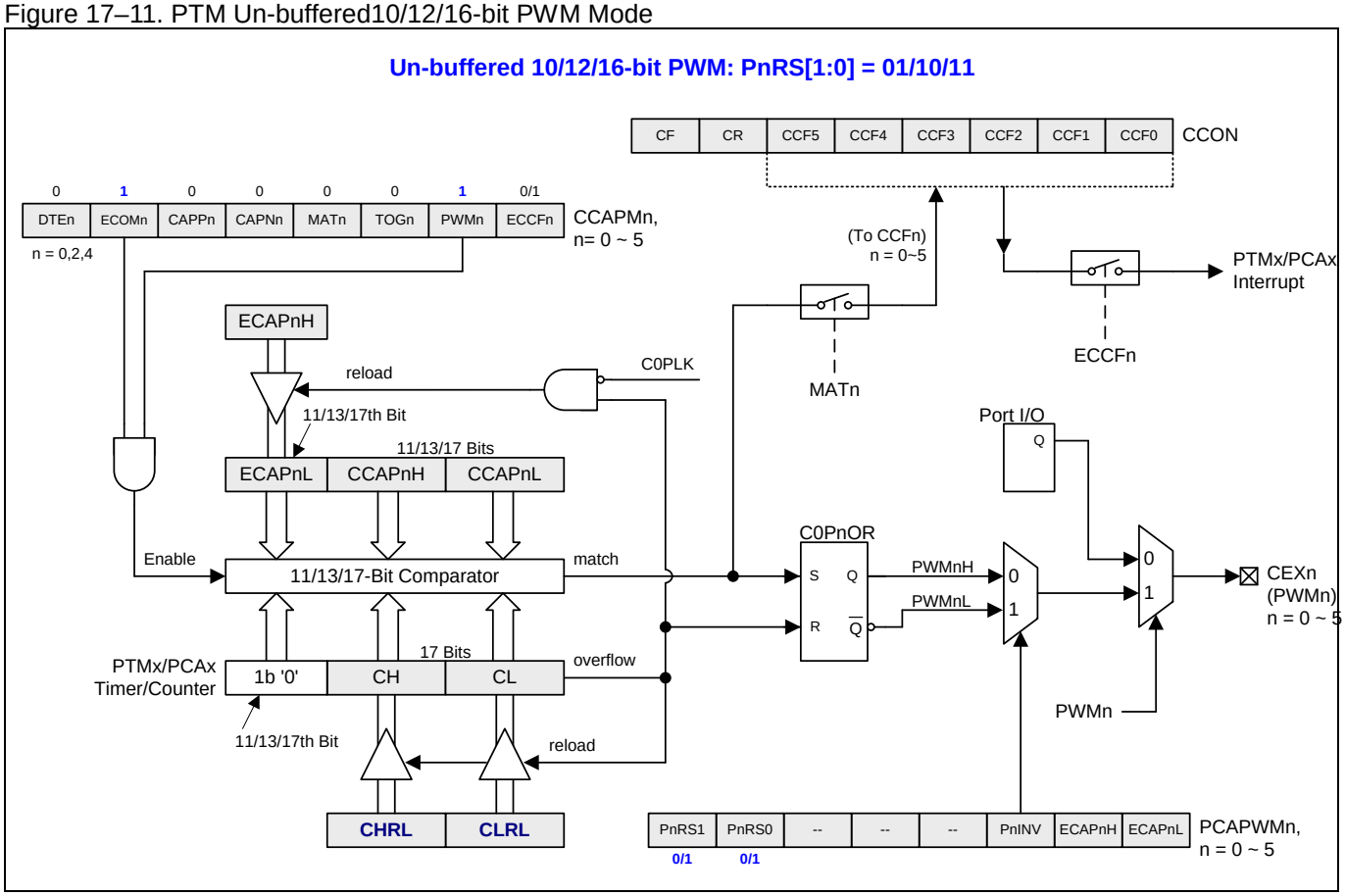
Figure 17–10. PTM0 Buffered 8-bit PWM Mode



17.4.8. Un-buffered 10/12/16-bit PWM Mode

The PTM 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.

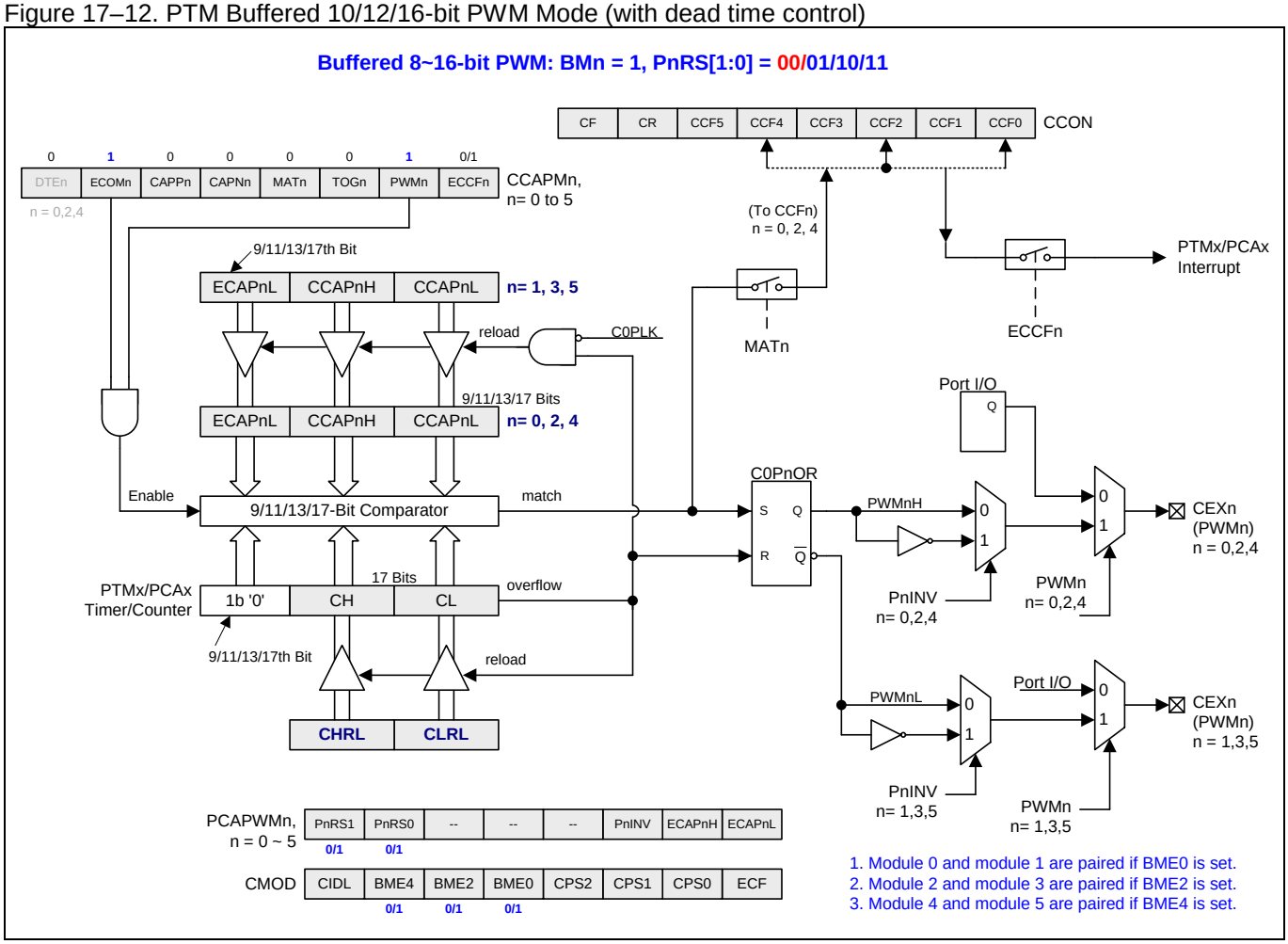
Figure 17–11. PTM Un-buffered10/12/16-bit PWM Mode



17.4.9. 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 has been written. If the applications need accurate control when change the duty cycle, it needs to use the Buffered PWM mode.

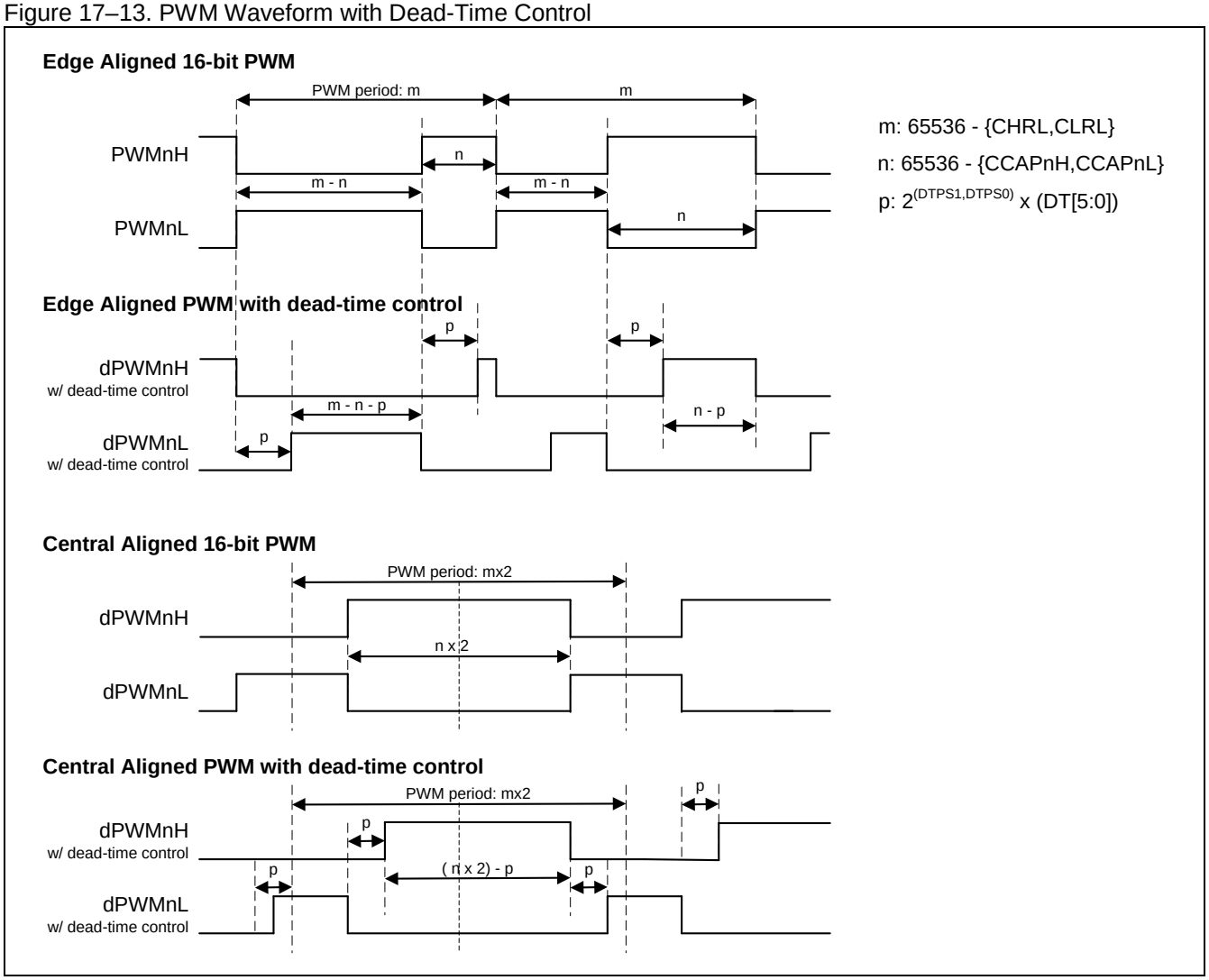
Figure 17–12. PTM Buffered 10/12/16-bit PWM Mode (with dead time control)



17.4.10. PTM0 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.

Figure 17–13. PWM Waveform with Dead-Time Control



CCAPMn: PTM Module Compare/Capture Register, n=0~5

SFR Page = 0 only

SFR Address = 0xDA~0xDF RESET = 0000-0000

7	6	5	4	3	2	1	0
DTEn	ECOMn	0	0	MATn	TOGn	PWMn	ECCFn
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: DTEn. Enable Dead-Time control on PWMHn/PWMLn 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 PWMn output.
1: Enable the Dead-Time control on PWMn output.

PDTCRA: PWM Dead-Time Control Register -A

SFR Page = **1 only**

SFR Address = **0xBC** RESET = 0000-0000

7	6	5	4	3	2	1	0
DTPS1	DTPS0	DT5	DT4	DT3	DT2	DT1	DT0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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-Time Disabled
000001	Pre-Scaler Clock X 1
000010	Pre-Scaler Clock X 2
000011	Pre-Scaler Clock X 3
…..	……
111110	Pre-Scaler Clock X 62
111111	Pre-Scaler Clock X 63

PWMCR: PWM Control Register

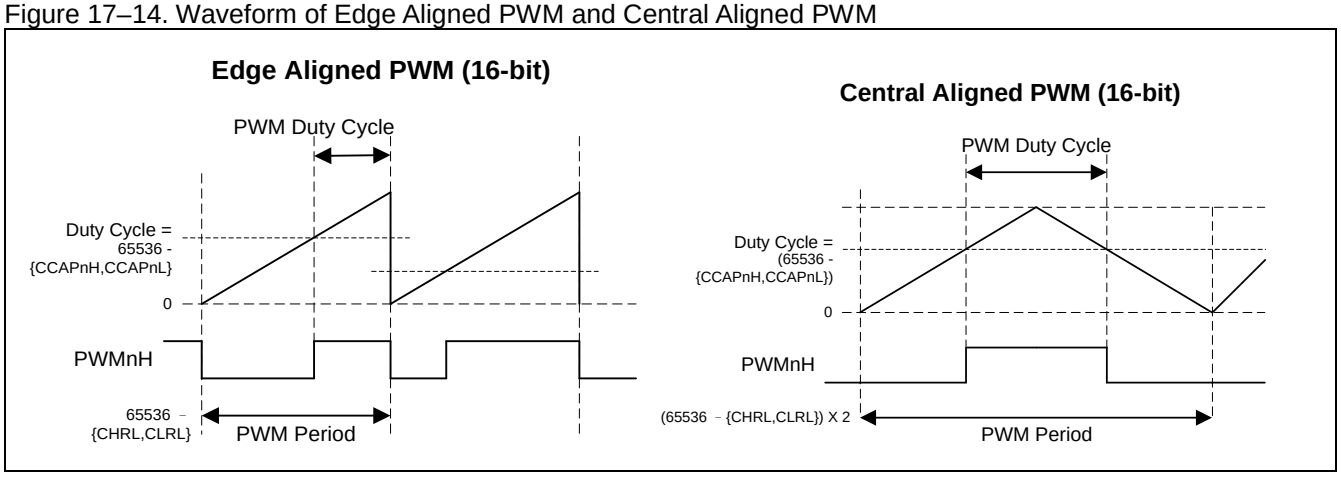
SFR Page = **0 only**

SFR Address = **0xBC** RESET = 0000-0000

7	6	5	4	3	2	1	0
PCAE	C0OFS	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**

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 is recommended to set all PWM channel to same resolution.

Figure 17–14. Waveform of Edge Aligned PWM and Central Aligned PWM



Bit 6: C0OFS, PTM0 overflow flag selection
0: CF is set on the top of central aligned PWM cycle.
1: CF is set on the bottom of central aligned PWM cycle.

Bit 5: PBKM, PWM Break Mode selection.
0: Latched Mode.
1: Cycle-by-cycle Mode.

Figure 17–15. Latch Mode Waveform of PWM Break control

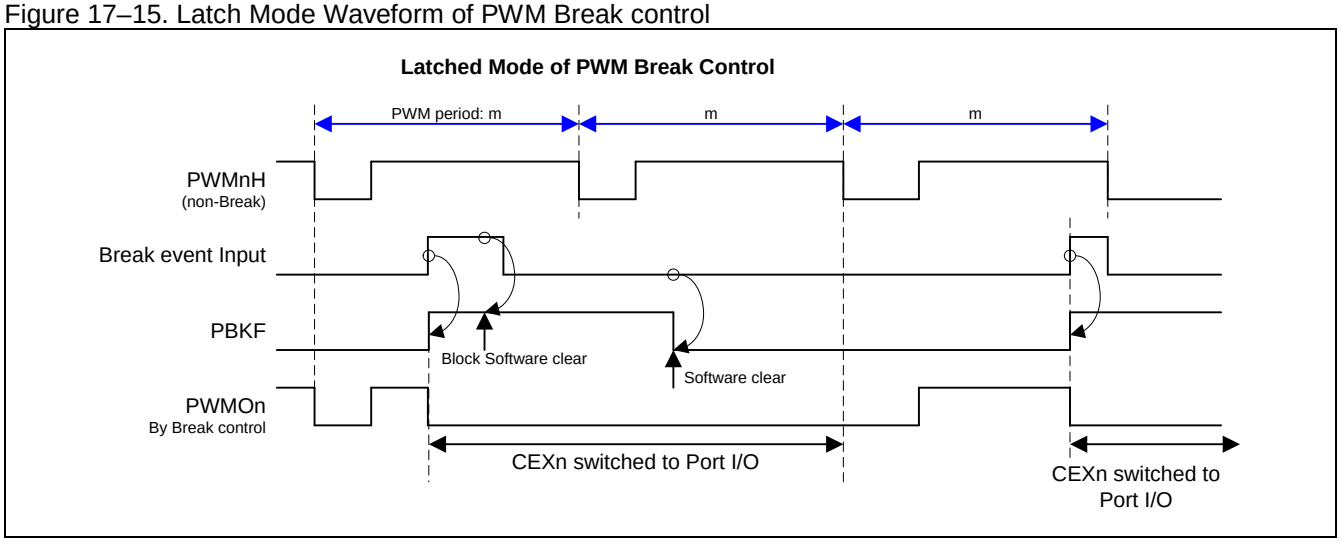
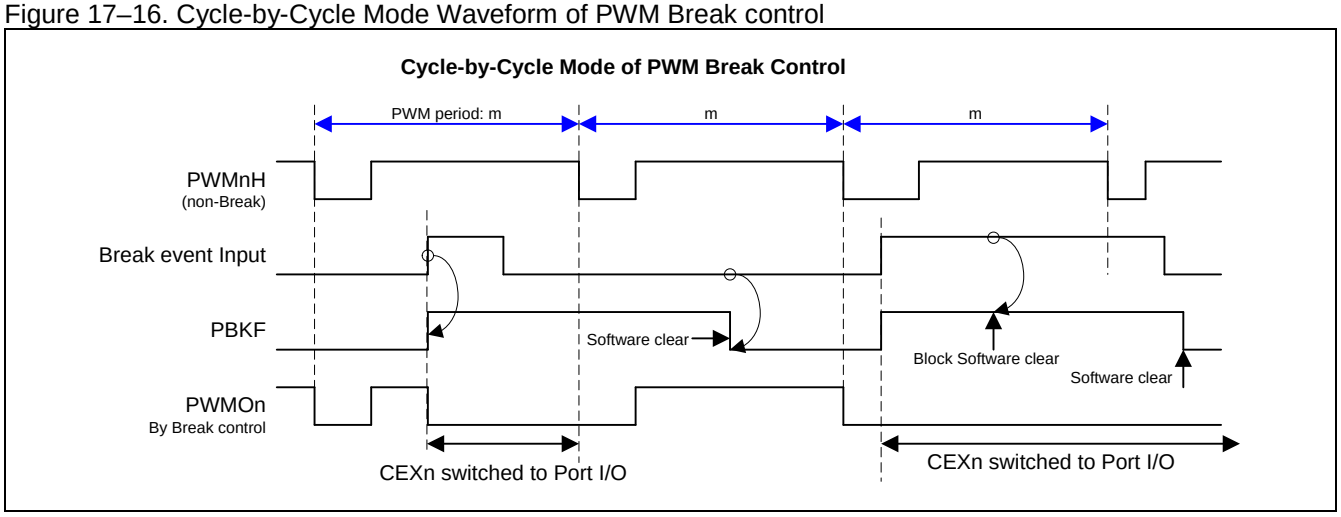


Figure 17–16. Cycle-by-Cycle Mode Waveform of PWM Break control



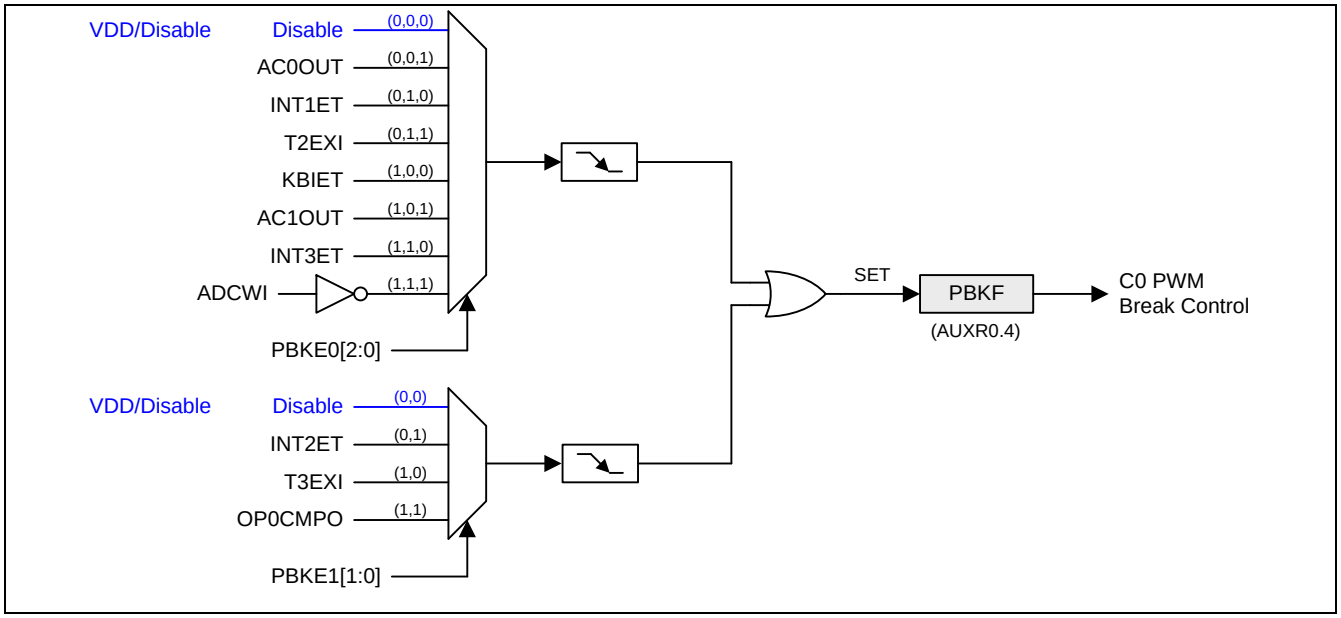
Bit 4~3: PBKE1.1~0, PWM Break Enable 1 selection. This function is only active on CEXn output mode (n=0~5).

PBKE1[1:0]	PWM Break Source
0 0	Disable PWM break source 1
0 1	INT2ET, nINT2 active
1 0	T3EXI
1 1	OP0ES

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
0 0 0	Disable PWM break source 0
0 0 1	AC0OUT
0 1 0	INT1ET, nINT1 active
0 1 1	T2EXI
1 0 0	KBIET, KBI match active
1 0 1	AC1OUT
1 1 0	INT3ET, nINT3 active
1 1 1	ADCWI active

Figure 17–17. PTM0 PWM Break control source



AUXR0: Auxiliary Register 0

SFR Page = 0~F

SFR Address = 0xA1 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	C1BKF	PBKF	0	0	INT1H	INT0H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 4: PBKF, PTM0 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.

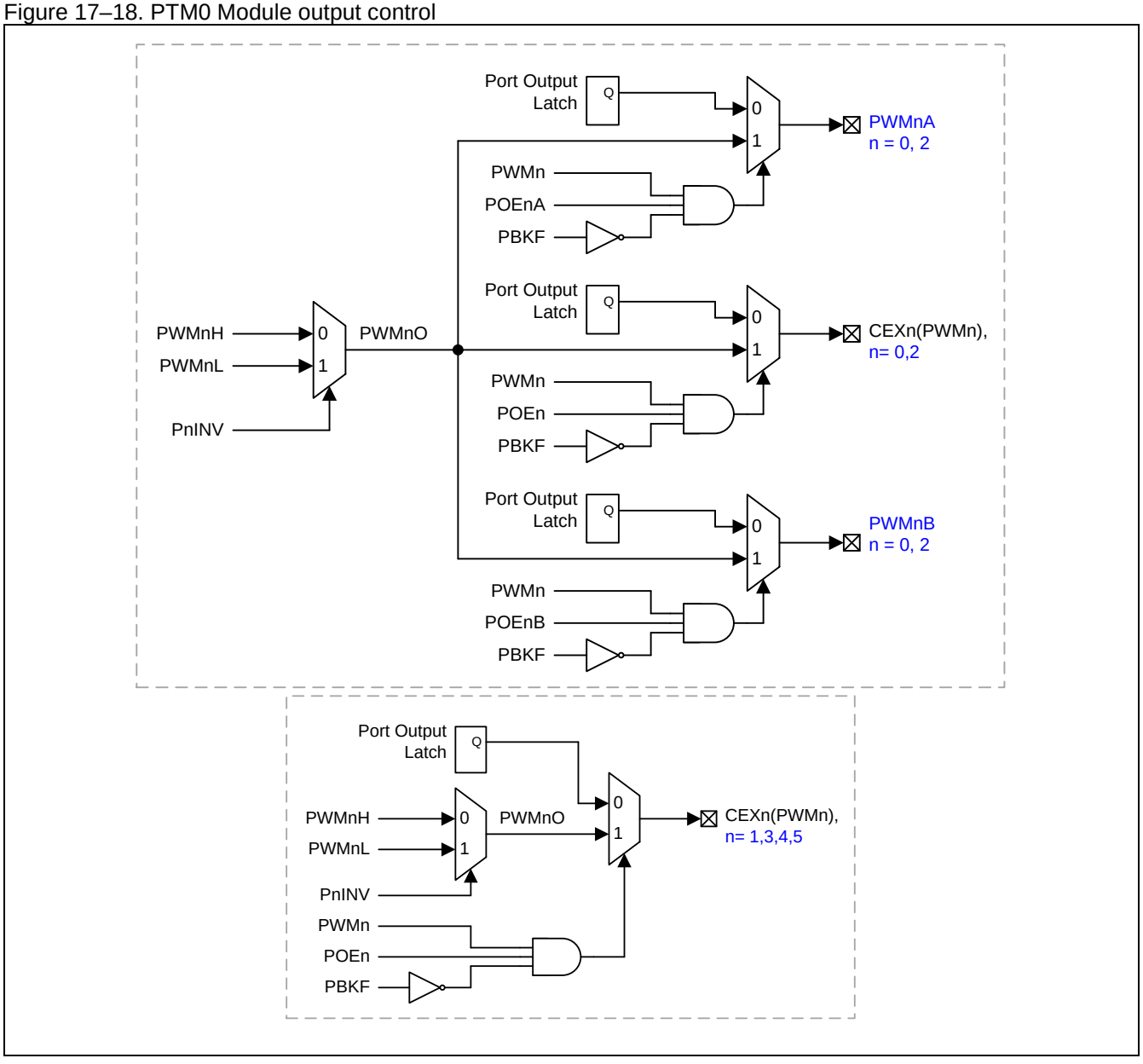
17.4.11. PTM0 Module Output Control

PTM0 modules have multi output control mode can be selected for different applications. The CEXn (n=0 ~ 5) can be programed as general I/O port or the output of the PTM0 module (PWM) 0 ~ 5. When PWM has been assigned to the CEXn, 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 CEXn (n=0 ~ 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.

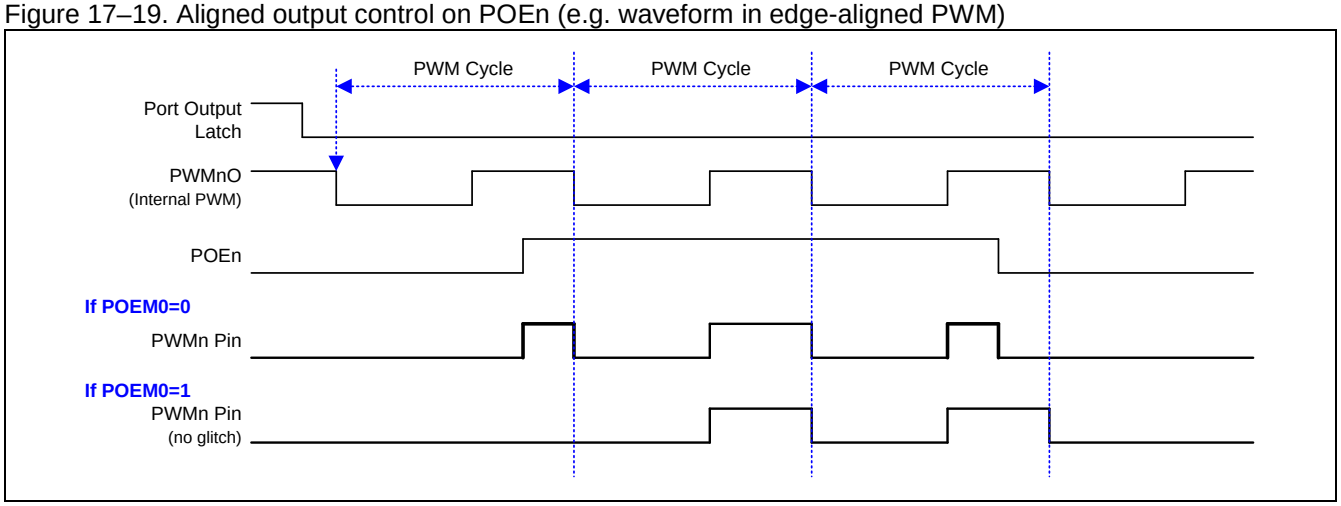
In addition, PTM0 module 0 and 2 have 2 “Cloned” signals to the different port pin. These three the same PWM signals can be masked particular cycles by the POEnA or POEnB or PWMn for the applications which need phase control.

Figure 17–18. PTM0 Module output control



POEM0 in **MG82F6P32** controls the POEn output timing to align with PWM cycle. The configuration and waveform of the aligned function is shown in Figure 17–19.

Figure 17–19. Aligned output control on POEn (e.g. waveform in edge-aligned PWM)



PAOE: PWM Additional Output Enable Register

SFR Page = **0~7**

SFR Address = **0xF1** RESET = 1001-1001

7	6	5	4	3	2	1	0
POE3	POE2B	POE2A	POE2	POE1	POE0B	POE0A	POE0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: POE3, PTM0 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, PTM0 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, PTM0 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, PTM0 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, PTM0 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, PTM0 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, PTM0 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, PTM0 PWM0 main channel (PWM0O) output control.
0: Disable PWM0O output on port pin.
1: Enable PWM0O output on port pin. **Default is enabled.**

C0AOE1: PWM Additional Output Enable Register

SFR Page = **8**

SFR Address = **0xF1** RESET = 0XXX-1001

7	6	5	4	3	2	1	0
POEM0				POE5			POE4
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: POEM0, PTM0 POEn control 0.
0: POEn function is active immediately after CPU writing.
1: POEn function is aligned to PWM cycle.

Bit 3: POE5, PTM0 PWM5 main channel (PWM5O) output control.
0: Disable PWM5O output on port pin.
1: Enable PWM5O output on port pin. **Default is enabled.**

Bit 0: POE4, PTM0 PWM4 main channel (PWM4O) output control.
0: Disable PWM4O output on port pin.
1: Enable PWM4O output on port pin. **Default is enabled.**

AUXR5: Auxiliary Register 5

SFR Page = **2 only**

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	C0PPS1	C0PPS0	EC1IPS0	C1COPS	ECIPS0	C0COPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

C0PPS1	PWM2A	PWM2B
0	P4.0	P4.1
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: EC1IPS0, PCA1 ECI Port pin Selection0.

EC1IPS0	EC1I
0	ECI
1	P1.5

Bit 2: C1COPS, PCA1 Clock Output (C1CKO) port pin Selection.

C1COPS	C1CKO
0	P4.7
1	P3.3

Bit 1: ECIPS0, PTM0 ECI Port pin Selection0.

ECIPS0	ECI
0	P0.2
1	P1.6

Bit 0: C0COPS, PTM0 Clock Output (C0CKO) port pin Selection.

C0COPS	C0CKO
0	P4.7
1	P3.3

AUXR7: Auxiliary Register 7

SFR Page = 4 Only

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	C0CKOE	C1CKOE	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5: C0CKOE, PTM0 clock output enable.

0: Disable PTM0 clock output.

1: Enable PTM0 clock output with PTM0 base timer overflow rate/2.

AUXR16: Auxiliary Register 16

SFR Page = D only

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	C0OPS5	C0OPS4	C0OPS3	C0OPS2	C0OPS1	C0OPS0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 5~0: C0OPS5 ~0, PTM Output Pin Selection.

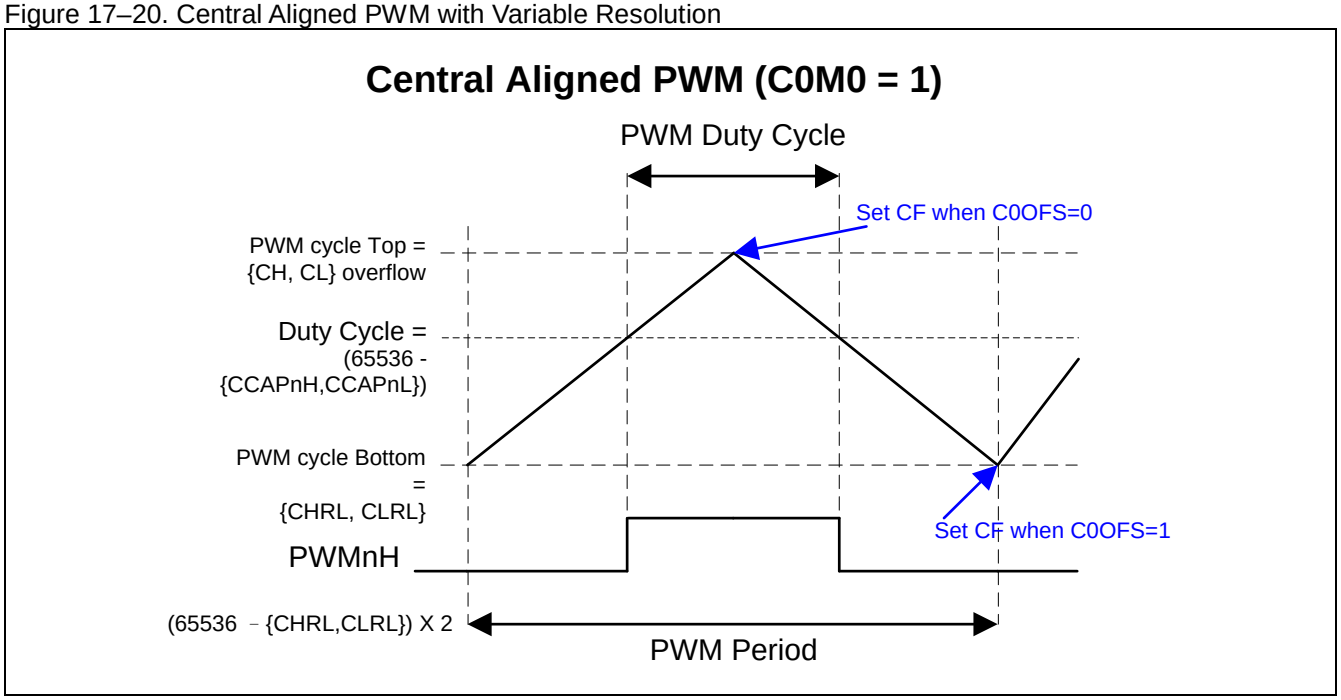
	C0OPS5	C0OPS4	C0OPS3	C0OPS2	C0OPS1	C0OPS0
C0OPSn	CEX5	CEX4	CEX3	CEX2	CEX1	CEX0
0	P0.7	P2.6	P0.5	P2.4	P2.3	P2.2
1	P3.5	P3.2	P3.4	P4.1	P3.3	P4.0

17.4.12. Variable Resolution on Central Aligned PWM

In Section “PTM0 Enhanced PWM Control”, it defines the central aligned PWM only support the 8/10/12/16-bit resolution. And in that mode, all of PTM0 functions, compare, on other non-PWM modules are still available.

Please note when using Central Aligned PWM, we suggest setting 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 17–20. Central Aligned PWM with Variable Resolution



PWMCR: PWM Control Register

SFR Page = **0 only**

SFR Address = **0xBC** RESET = 0000-0000

7	6	5	4	3	2	1	0
PCAE	C0OFS	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**

Bit 0: C0OFS, PTM0 overflow flag selection.
0: CF is set on the top of central aligned PWM cycle.
1: CF is set on the bottom of central aligned PWM cycle.

17.4.13. PWM Global control

PTM0, PCA1 and Timer 2~4 can use the global control method to sync the frequency and phase. Please reference “16.4 Timer Global Control”.

18. Programmable Counter Array (PCA1)

The **MG82F6P32** is equipped with one programmable Counter Array (PCA1), which provides more timing capabilities with less CPU intervention than the standard timer/counters. Its advantages include reduced software overhead and improved accuracy.

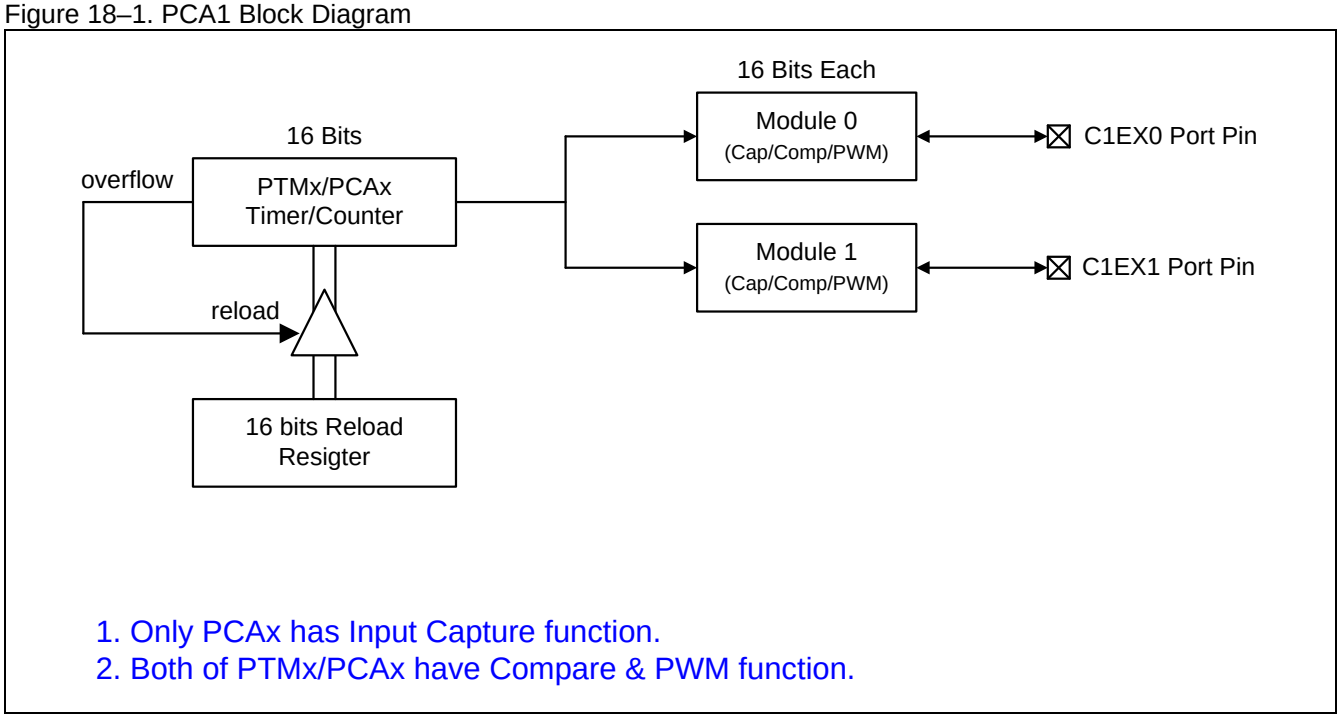
18.1. PCA1 Overview

The PCA1 consists of a dedicated timer/counter which serves as the time base for an array of **2** compare/PWM modules. Figure 18–1 shows a block diagram of the PCA1. Notice that the PCA1 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.

PCA1 Module 0~1 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)

PCA1 Module 0~1 also support the modes listed on top but add the capture function. All of these modes will be discussed later in detail. However, let's first look at how to set up the PCA1 timer and modules.

Figure 18–1. PCA1 Block Diagram



1. Only PCAx has Input Capture function.
2. Both of PTMx/PCAx have Compare & PWM function.

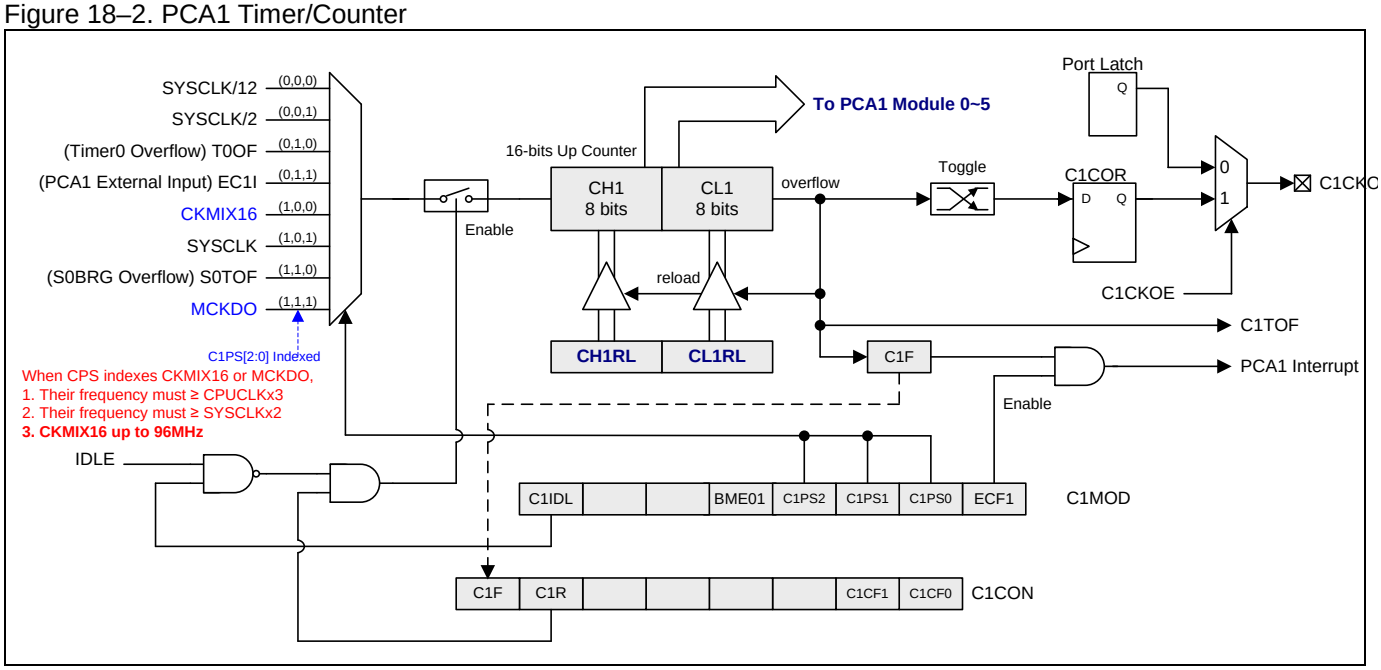
18.2. PCA1 Timer/Counter

The timer/counter for the PCA1 is 16-bit auto-reload timer consisting of registers CH1 and CL1 (the high and low bytes of the count values), CH1RL, CL1RL (the high and low bytes reload registers), as shown in Figure 18–2 . CH1RL and CL1RL are reloaded to CH1 and CL1 at each time overflow on {CH1+CL1} counter which can change the PCA1 cycle time for variable PWM resolution, such as 7-bit or 9-bit PWM.

{CH1 + CL1} 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 C1MOD contains the Count Pulse Select bits (C1PS2, C1PS1 and C1PS0) to specify the PCA1 timer input. When C1PS[2:0] indexes CKMIX16 or MCKDO, the frequency of the input clock sources must ≥ CPUCLK x3 and ≥ SYSCLK x2.This register also contains the ECF bit which enables an interrupt when the counter {CH1+CL1} overflows. And the counter overflow toggles C1COR, it will output on port pin when C1CKOE is enabled. In addition, the user has the option of turning off the PCA1 timer during Idle Mode by setting the Counter Idle bit (C1IDL). This can further reduce power consumption during Idle mode.

Figure 18–2. PCA1 Timer/Counter



C1MOD: PCA1 Counter Mode Register

SFR Page = 1 Only
SFR Address = 0xD9 RESET = 0000-0000

7	6	5	4	3	2	1	0
C1IDL	0	0	BME01	CPS21	CPS11	CPS01	ECF1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: C1IDL, PCA1 counter Idle control.
0: Lets the PCA1 counter continue functioning during Idle mode.
1: Lets the PCA1 counter be gated off during Idle mode.

Bit 4: BME01, Buffer Mode Enable on PCA1 module 0/1. It is only valid on both of PCA1 module 0 and module 1 in capture mode, PWM mode or COPM mode.
0: PCA1 Module 0/1 buffer mode disabled.
1: PCA1 Module 0/1 buffer mode enabled.

Bit 3~1: C1PS2-C1PS0, PCA counter clock source select bits.

CPS21	CPS11	CPS01	PCA1 Clock Source
0	0	0	Internal clock, (system clock)/12
0	0	1	Internal clock, (system clock)/2
0	1	0	Timer 0 overflow
0	1	1	External clock at the ECI pin
1	0	0	CKMIX16 output
1	0	1	Internal clock, (system clock)/1
1	1	0	S0BRT overflow
1	1	1	MCK Divider Output, MCKDO

Note: When CPS indexes CKMIX16 or MCKDO, needs to follow the conditions:

3. *The source frequency must ≥ CPUCLK x3.*
4. *The source frequency must ≥SYSCLK x2.*

Bit 0: ECF1, Enable PCA1 counter overflow interrupt.
0: Disables an interrupt when C1F bit (in C1CON register) is set.
1: Enables an interrupt when C1F bit (in C1CON register) is set.

The C1CON register shown below contains the run control bit for the PCA1 and the flags for the PCA1 timer and each module. To run the PCA1 the C1R bit (C1CON.6) must be set by software. The PCA1 is shut off by clearing this bit. The C1F bit (C1ON.7) is set when the PCA1 counter overflows and an interrupt will be generated if the ECF1 bit in the C1MOD register is set. The C1F bit can only be cleared by software. C1CF0 to C1CF5 are the interrupt flags for module 0 to module 5, respectively, and they are set by hardware when match occurs. These flags also can only be cleared by software. The PCA1 interrupt system is shown Figure 18–3.

C1CON: PCA1 Counter Control Register

SFR Page = 1 only
SFR Address = 0xD8 RESET = 0000-0000

7	6	5	4	3	2	1	0
C1F	C1R	0	0	0	0	C1CF1	C1CF0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: C1F, PCA1 Counter Overflow flag.
0: Only be cleared by software.
1: Set by hardware when the counter rolls over. C1F flag can generate an interrupt if bit ECF1 in C1MOD is set. C1F may be set by either hardware or software.

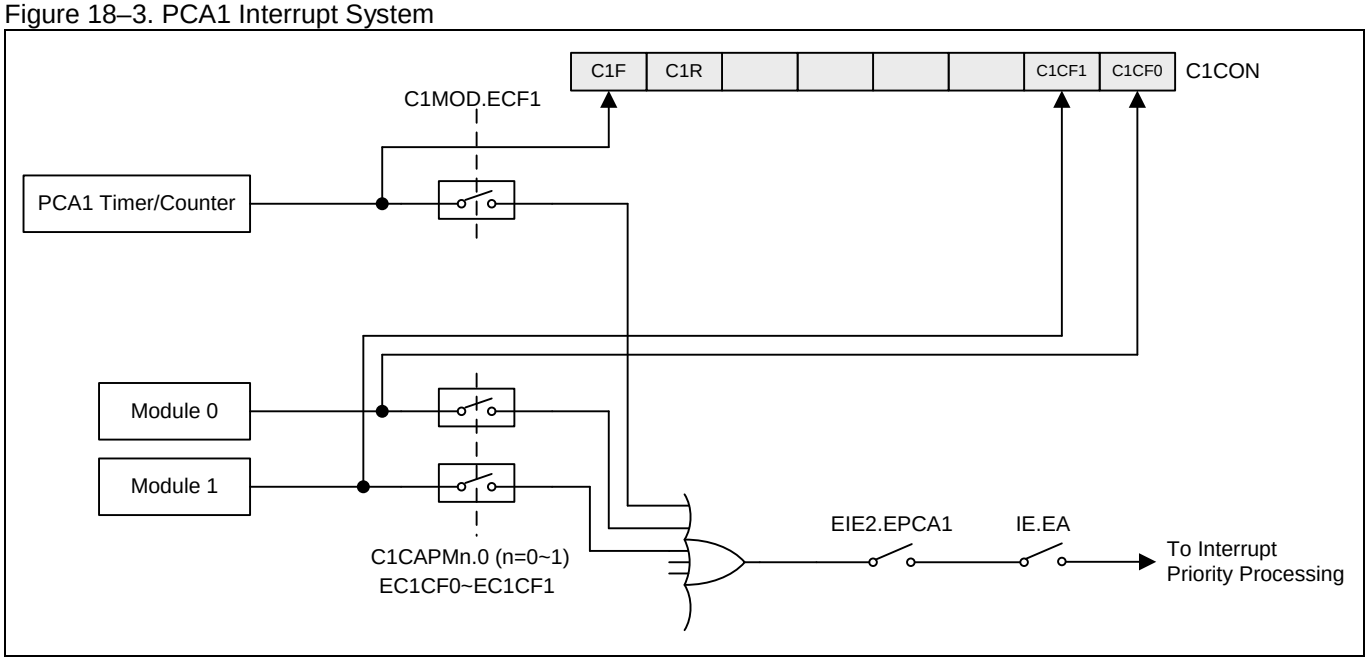
Bit 6: C1R, PCA1 Counter Run control bit.
0: Must be cleared by software to turn the PCA1 counter off.
1: Set by software to turn the PCA1 counter on.

Bit 5~2: Reserved. Software must write “0” on these bits when C1CON is written.

Bit 1: C1CF1, PCA1 Module 1 interrupt flag.
0: Must be cleared by software.
1: Set by hardware when a match or capture occurs.

Bit 0: C1CF0, PCA1 Module 0 interrupt flag.
0: Must be cleared by software.
1: Set by hardware when a match or capture occurs.

Figure 18–3. PCA1 Interrupt System



C1PWMn: PWM Mode Auxiliary Register, n=0~1

SFR Page = 1 only

SFR Address = 0xF2 RESET = 0000-0000

7	6	5	4	3	2	1	0
PnRS11	PnRS01	--	--	--	PnINV1	ECAPnH1	ECAPnL1
R/W	R/W	W	W	R/W	R/W	R/W	R/W

Bit 7~6: PnRS11~01, PCA1 PWMn Resolution Setting 1~0.
00: 8 bit PWMn, the overflow is active when [CH1, CL1] counts XXXX-XXXX-1111-1111 → XXXX-XXXX-0000-0000.
01: 10 bit PWMn, the overflow is active when [CH1, CL1] counts XXXX-XX11-1111-1111 → XXXX-XX00-0000-0000.
10: 12 bit PWMn, the overflow is active when [CH1, CL1] counts XXXX-1111-1111-111 → XXXX-0000-0000-0000.
11: 16 bit PWMn, the overflow is active when [CH1, CL1] counts 1111-1111-1111-1111→ 0000-0000-0000-0000.

Bit 5~3: Reserved. Software must write “0” on these bits when C1PWMn is written.

Bit 2: PnINV1, Invert Compare/PWM output (C1PnOR) on C1EXn pin.
0: Non-inverted Compare/PWM output (C1PnOR).
1: Inverted Compare/PWM output (C1PnOR).

Bit 1: ECAPnH1, Extended 9th bit (MSB bit), associated with C1CAPnH to become a 9-bit register used in PWM mode.

Bit 0: ECAPnL1, Extended 9th bit (MSB bit), associated with C1CAPnL to become a 9-bit register used in PWM mode.

CH1: PCA1 base timer High

SFR Page = **1 only**

SFR Address = 0xF9 RESET = 0000-0000

7	6	5	4	3	2	1	0
CH1.7	CH1.6	CH1.5	CH1.4	CH1.3	CH1.2	CH1.1	CH1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CL1: PCA1 base timer Low

SFR Page = **1 only**

SFR Address = 0xE9 RESET = 0000-0000

7	6	5	4	3	2	1	0
CL1.7	CL1.6	CL1.5	CL1.4	CL1.3	CL1.2	CL1.1	CL1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CH1RL: PCA1 CH Reload Register

SFR Page = **1 only**

SFR Address = 0xCF RESET = 0000-0000

7	6	5	4	3	2	1	0
CH1RL.7	CH1RL.6	CH1RL.5	CH1RL.4	CH1RL.3	CH1RL.2	CH1RL.1	CH1RL.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: CH1RL, reload value of CH1.

CL1RL: PCA1 CL Reload Register

SFR Page = 1 **only**

SFR Address = 0xCE RESET = 0000-0000

7	6	5	4	3	2	1	0
CL1RL.7	CL1RL.6	CL1RL.5	CL1RL.4	CL1RL.3	CL1RL.2	CL1RL.1	CL1RL.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~0: CL1RL, reload value of CL1.

18.3. Compare/Capture Modules

Each of the compare/capture module 0~1 has a mode register called C1CAPMn (n = 0,1) to select which function it will perform. Note the ECCFn1 bit which enables an interrupt to occur when a module's interrupt flag is set.

C1CAPMn: PCA1 Module Compare/Capture Register, n=0~1

SFR Page = 1 only

SFR Address = 0xDA ~ 0xDB RESET = 0000-0000

7	6	5	4	3	2	1	0
DT1En	ECOMn1	CAP1Pn	CAP1Nn	MATn1	TOGn1	PWMn1	EC1CFn
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: DT1En. Enable Dead-Time control on PWMHn/PWMLn output pair. This bit is only valid on n= 0 the dead-time function is active when PWM channel is operating in buffer mode. The channel buffer mode is enabled by BME01 in C1MOD.
0: Disable the Dead-Time control on PWMn output.
1: Enable the Dead-Time control on PWMn output.

Bit 6: ECOMn1, Enable Comparator.
0: Disable the digital comparator function.
1: Enables the digital comparator function.

Bit 5: CAP1Pn, Capture Positive enabled.
0: Disable the PCA1 capture function on C1EXn positive edge detected.
1: Enable the PCA1 capture function on C1EXn positive edge detected.

Bit 4: CAP1Nn, Capture Negative enabled.
0: Disable the PCA1 capture function on C1EXn negative edge detected.
1: Enable the PCA1 capture function on C1EXn negative edge detected.

Bit 3: MATn1, Match control.
0: Disable the digital comparator match event to set C1CFn.
1: A match of the PCA1 counter with this module’s compare/capture register causes the C1CFn bit in CCON1 to be set.

Bit 2: TOGn1, Toggle control.
0: Disable the digital comparator match event to toggle C1EXn.
1: A match of the PCA1 counter with this module’s compare/capture register causes the C1EXn pin to toggle.

Bit 1: PWMn1, PWM control.
0: Disable the PWM mode in PCA1 module.
1: Enable the PWM function and cause C1EXn pin to be used as a pulse width modulated output.

Bit 0: EC1CFn, Enable C1CFn interrupt.
0: Disable compare/capture flag C1CFn in the C1CON register to generate an interrupt.
1: Enable compare/capture flag C1CFn in the C1CON register to generate an interrupt.

Note: The bits CAP1Nn (C1CAPMn.4) and CAP1Pn (C1CAPMn.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 (C1CAPnH, C1CAPnL) 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 C1PWMn 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 17.4.8 and 17.4.9.

C1CAPnH: PCA1 Module n Capture High Register, n=0~1

SFR Page = 1 only

SFR Address = 0xFA~0xFB RESET = 0000-0000

7	6	5	4	3	2	1	0
C1CAPnH.7	C1CAPnH.6	C1CAPnH.5	C1CAPnH.4	C1CAPnH.3	C1CAPnH.2	C1CAPnH.1	C1CAPnH.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

C1CAPnL: PCA1 Module n Capture Low Register, n=0~1

SFR Page = 1 only

SFR Address = 0xEA~0xEB RESET = 0000-0000

7	6	5	4	3	2	1	0
C1CAPnL.7	C1CAPnL.6	C1CAPnL.5	C1CAPnL.4	C1CAPnL.3	C1CAPnL.2	C1CAPnL.1	C1CAPnL.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

C1PWMn: PCA1 PWM Mode Auxiliary Register, n=0~1

SFR Page = 1 only

SFR Address = 0xF2~0xF3 RESET = 0000-0000

7	6	5	4	3	2	1	0
PnRS11	PnRS01	--	--	--	PnINV1	ECAPnH1	ECAPnL1
R/W	R/W	W	W	R/W	R/W	R/W	R/W

Bit 7~6: PnRS11~0, PWMn Resolution Setting 1~0.
00: 8 bit PWMn, the overflow is active when [CH1, CL1] counts XXXX-XXXX-1111-1111 → XXXX-XXXX-0000-0000.
01: 10 bit PWMn, the overflow is active when [CH1, CL1] counts XXXX-XX11-1111-1111 → XXXX-XX00-0000-0000.
10: 12 bit PWMn, the overflow is active when [CH1, CL1] counts XXXX-1111-1111-111 → XXXX-0000-0000-0000.
11: 16 bit PWMn, the overflow is active when [CH1, CL1] counts 1111-1111-1111-1111→ 0000-0000-0000-0000.

Bit 5~4: Reserved. Software must write “0” on these bits when C1PWMn is written.

Bit 2: PnINV1, Invert Compare/PWM output (C1PnOR) on C1EXn pin.
0: Non-inverted Compare/PWM output (C1PnOR).
1: Inverted Compare/PWM output (C1PnOR).

Bit 1: ECAPnH1, Extended 9th bit (MSB bit), associated with C1CAPnH to become a 9-bit register used in PWM mode.

Bit 0: ECAPnL1, Extended 9th bit (MSB bit), associated with C1CAPnL to become a 9-bit register used in PWM mode.

18.4. Operation Modes of the PCA1

Table 18–1 shows the C1CAPMn register settings for the various PCA1 functions.

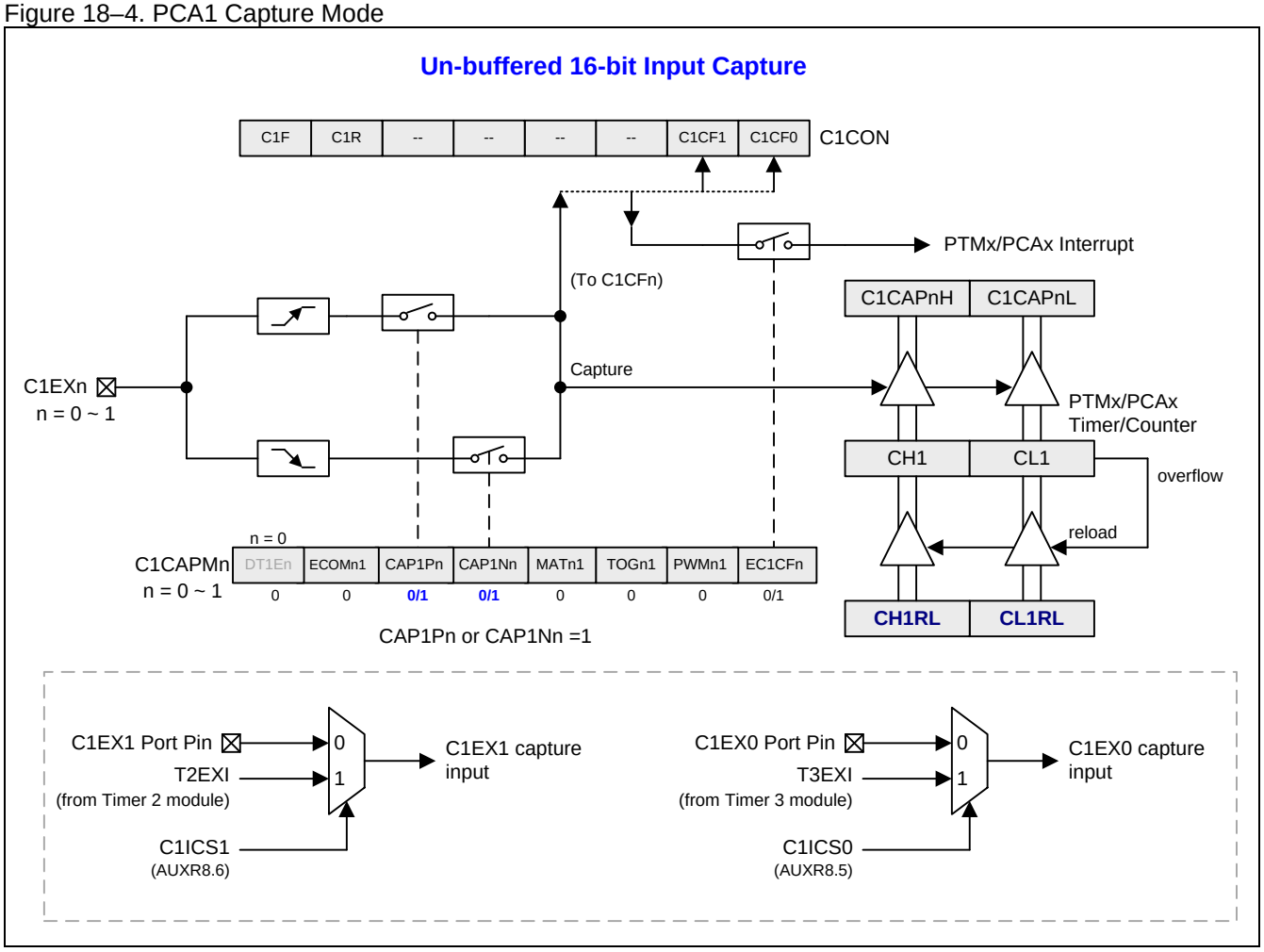
Table 18–1. PCA1 Module Modes

ECOMn1	CAP1Pn	CAP1Nn	MATn1	TOGn1	PWMn1	BMEn1	Module Function
0	0	0	0	0	0	0	No operation
X	1	0	0	0	0	0	Input Capture by a positive-edge trigger on C1EXn
X	0	1	0	0	0	0	Input Capture by a negative-edge trigger on C1EXn
X	1	1	0	0	0	0	Input Capture by a transition on C1EXn
X	1/0	0/1	0	0	0	1	Buffered Input Capture by a transition on C1EXn
1	0	0	1	0	0	0	Software Timer (Compare)
1	0	0	1	0	0	1	Buffered Software Timer (Compare)
1	0	0	1	1	0	0	High Speed Output (Output Compare)
1	0	0	1	1	0	1	Buffered High Speed Output (Output Compare)
1	0	0	0/1	0	1	0	Pulse Width Modulator (PWM)
1	0	0	0/1	0	1	1	Buffered 10/12/16-bit PWM

18.4.1. Capture Mode

To use one of the PCA1 modules in the capture mode, either one or both of the bits CAP1N and CAP1P for that module must be set. The external C1EX input for the module is sampled for a transition. When a valid transition occurs the PCA1 hardware loads the value of the PCA1 counter registers (CH1 and CL1) into the module's capture registers (C1CAPnL and C1CAPnH). If the C1CFn and the EC1CFn bits for the module are both set, an interrupt will be generated.

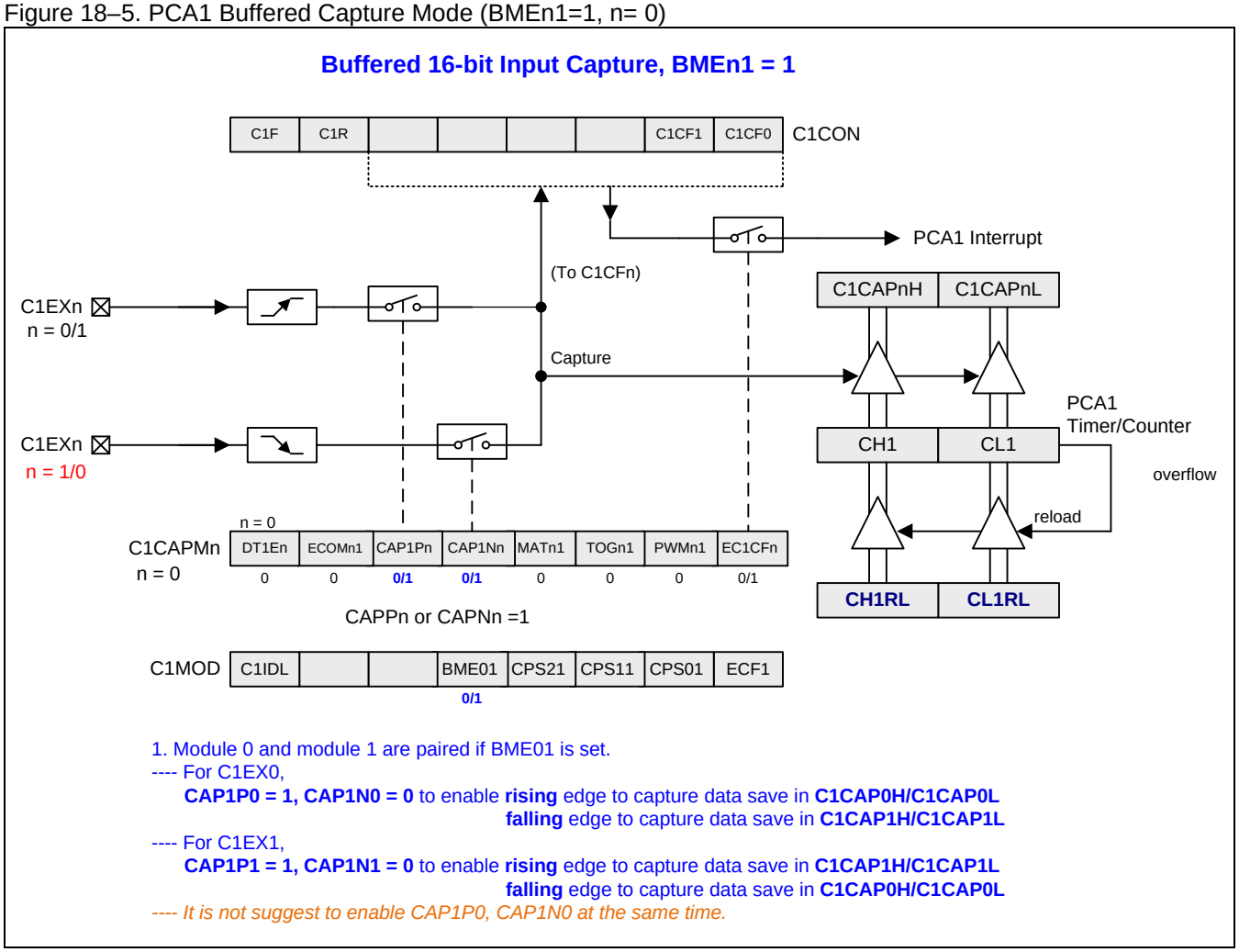
Figure 18–4. PCA1 Capture Mode



18.4.2. Buffered Capture Mode

To capture narrow input signal, buffered capture mode is necessary. If enabled, it put the odd module capture data registers (C1CAPnH, C1CAPnL, n= 1) to be the buffer register of even module capture data registers (channel 0). There is no influence on module 0 capture operation. BME01 enables the buffer operation of channel 0 and channel 1.

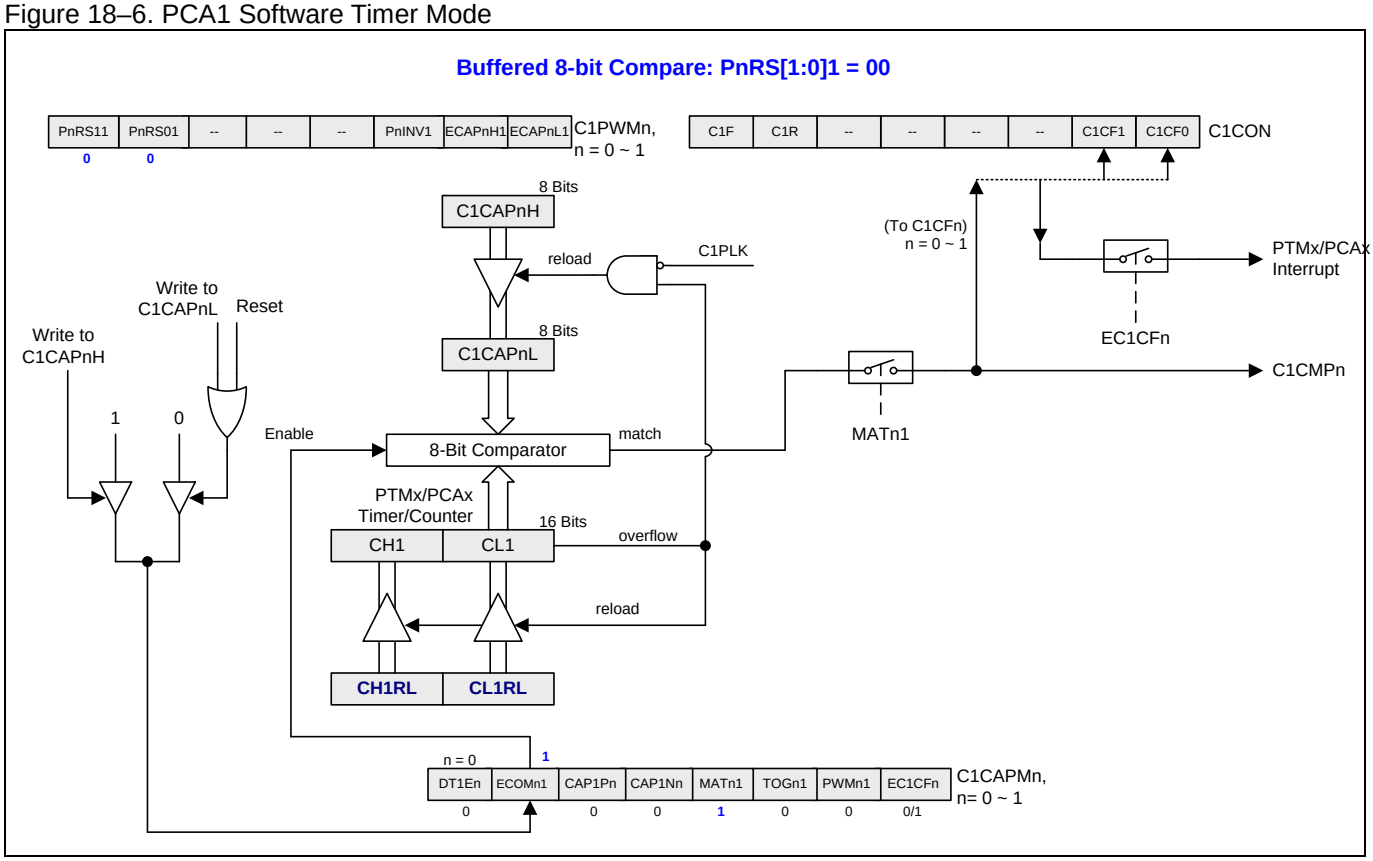
Figure 18–5. PCA1 Buffered Capture Mode (BMEn1=1, n= 0)



18.4.3. 8-bit Software Timer Mode (Compare mode)

The PCA1 modules can be used as software timers by setting both the ECOMn1 and MATn1 bits in the module's C1CAPMn register. The PCA1 timer will be compared to the module's capture registers, and when a match occurs an interrupt will occur if the C1CFn and the EC1CFn bits for the module are both set.

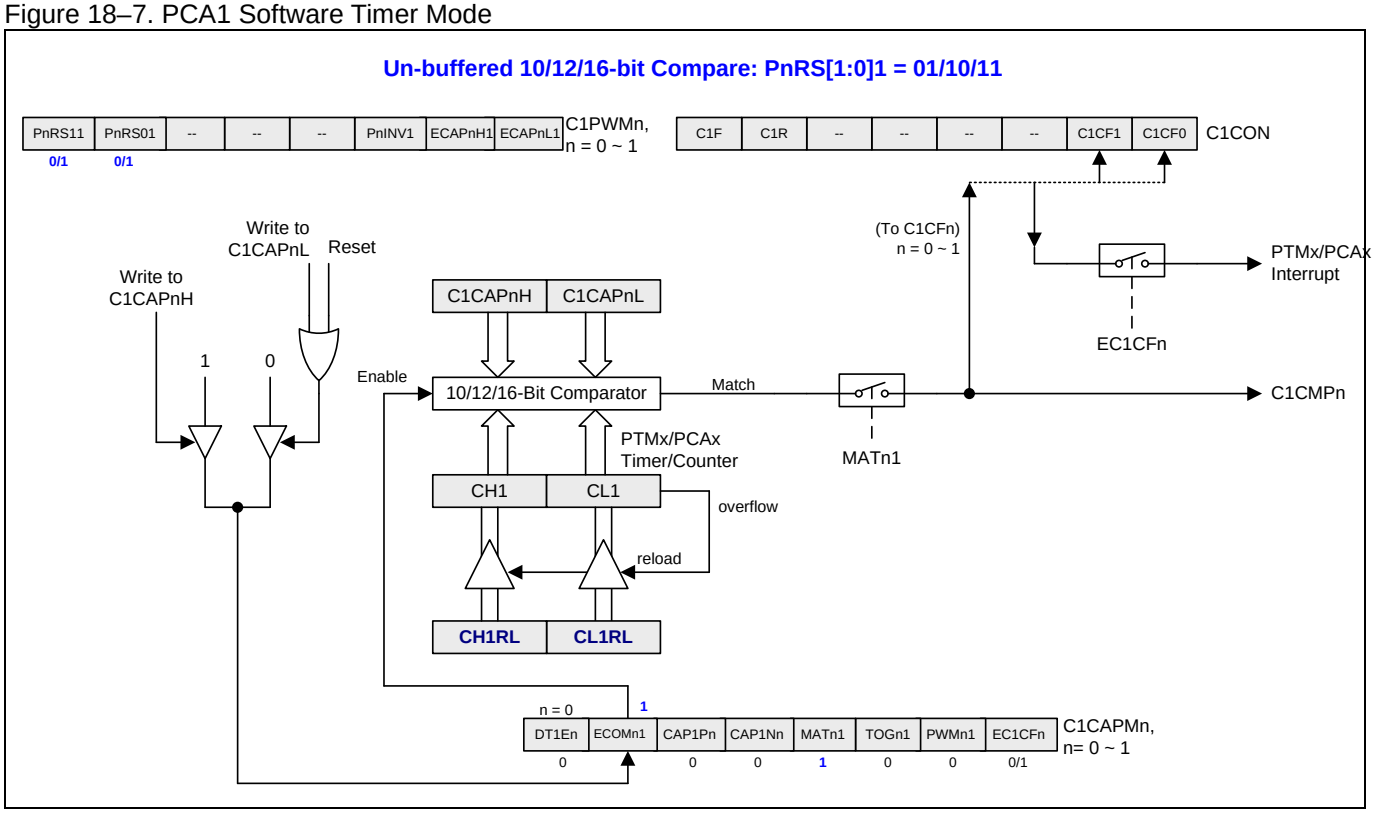
Figure 18–6. PCA1 Software Timer Mode



18.4.4. 16-bit Software Timer Mode (Compare mode)

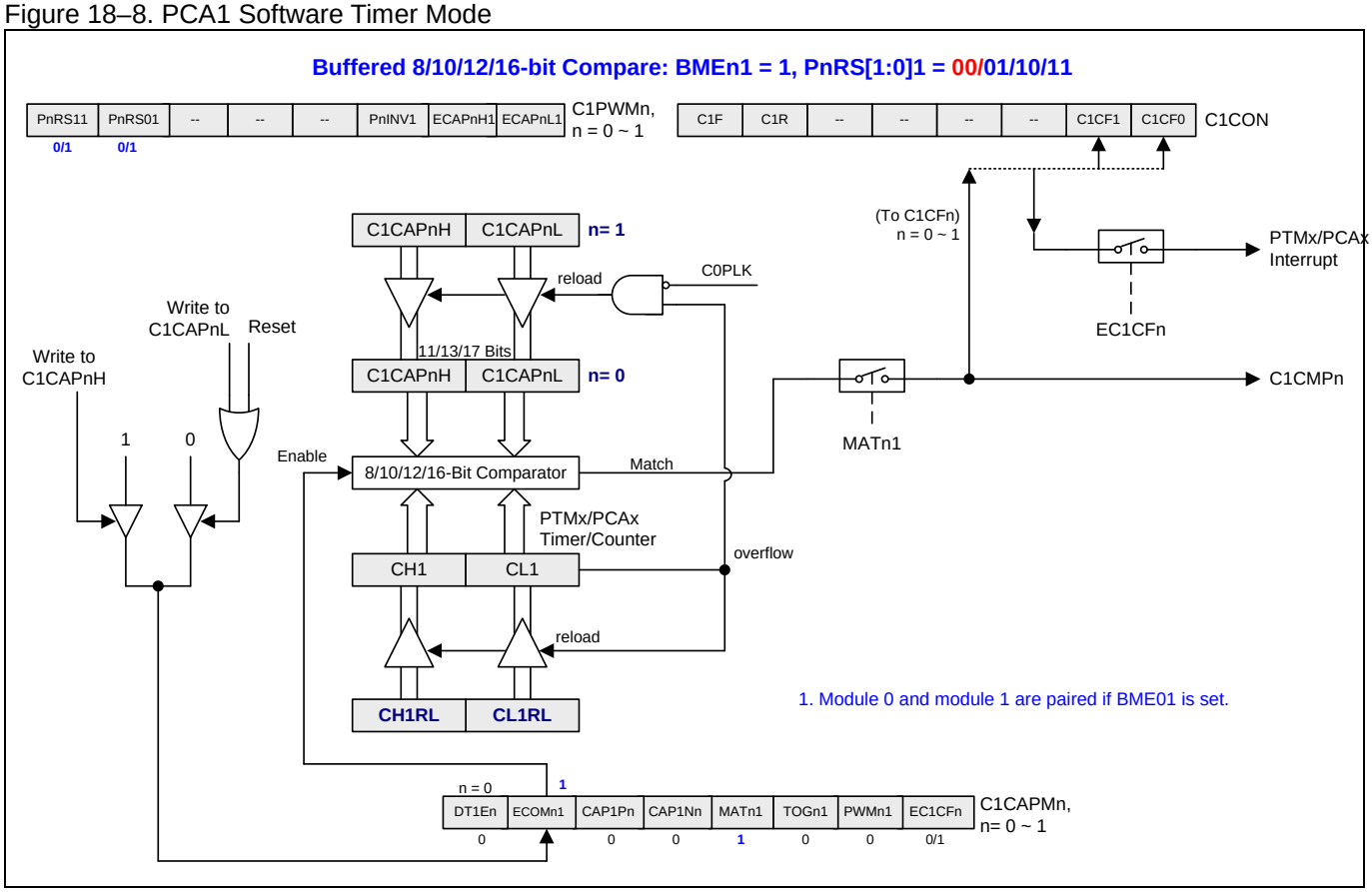
The PCA1 modules can be used as software timers by setting both the ECOMn1 and MATn1 bits in the module's C1CAPMn register. The PCA1 timer will be compared to the module's capture registers, and when a match occurs an interrupt will occur if the C1CFn and the EC1CFn bits for the module are both set.

Figure 18–7. PCA1 Software Timer Mode



18.4.5. Buffered 16-bit Software Timer Mode (Compare mode)

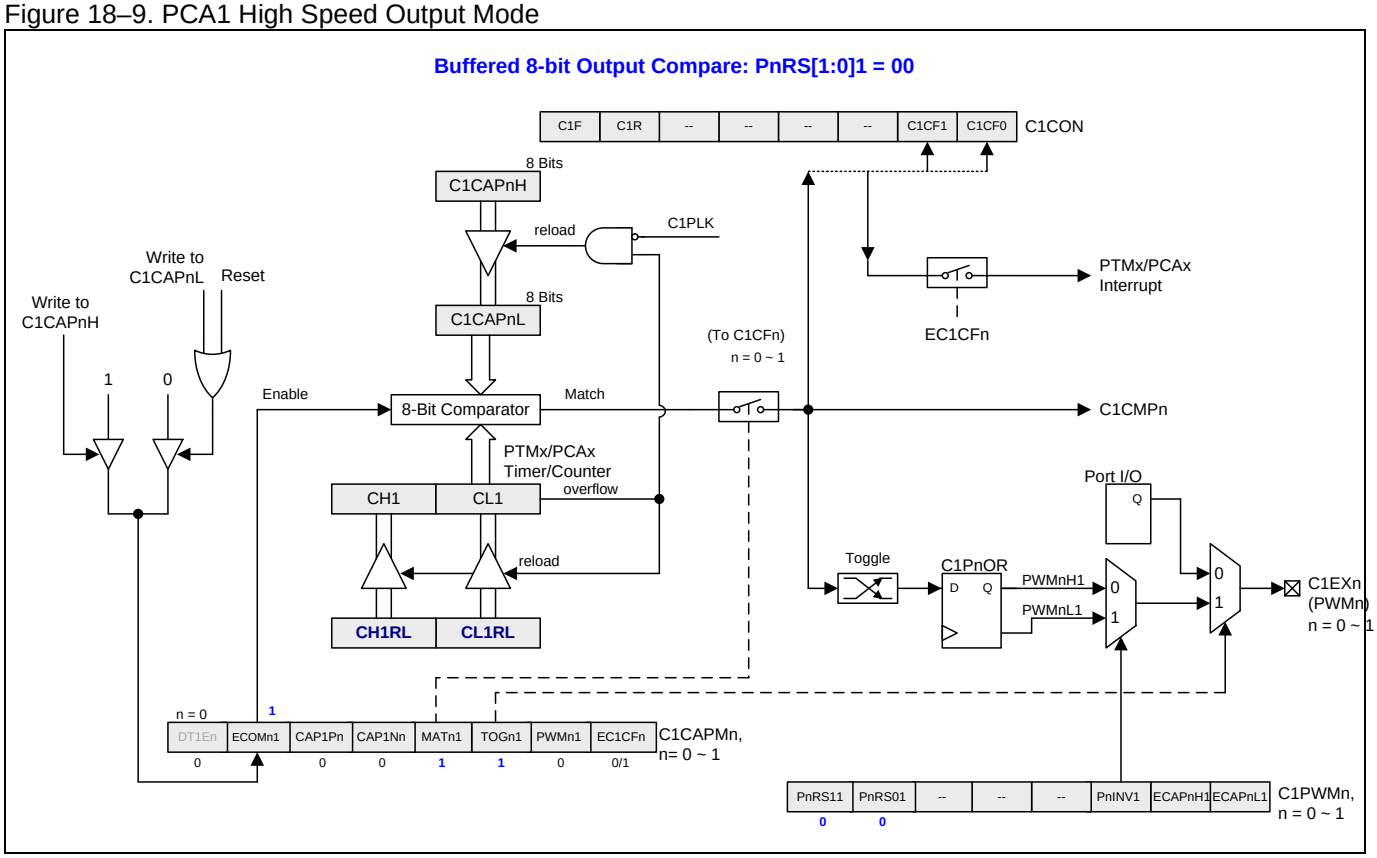
Figure 18–8. PCA1 Software Timer Mode



18.4.6. 8-bit High Speed Output Mode (Output Compare mode)

In this mode the C1EX output associated with the PCA1 module will toggle each time a match occurs between the PCA1 counter and the module's capture registers. To activate this mode, the TOGn1, MATn1 and ECOMn1 bits in the module's C1CAPMn register must be set.

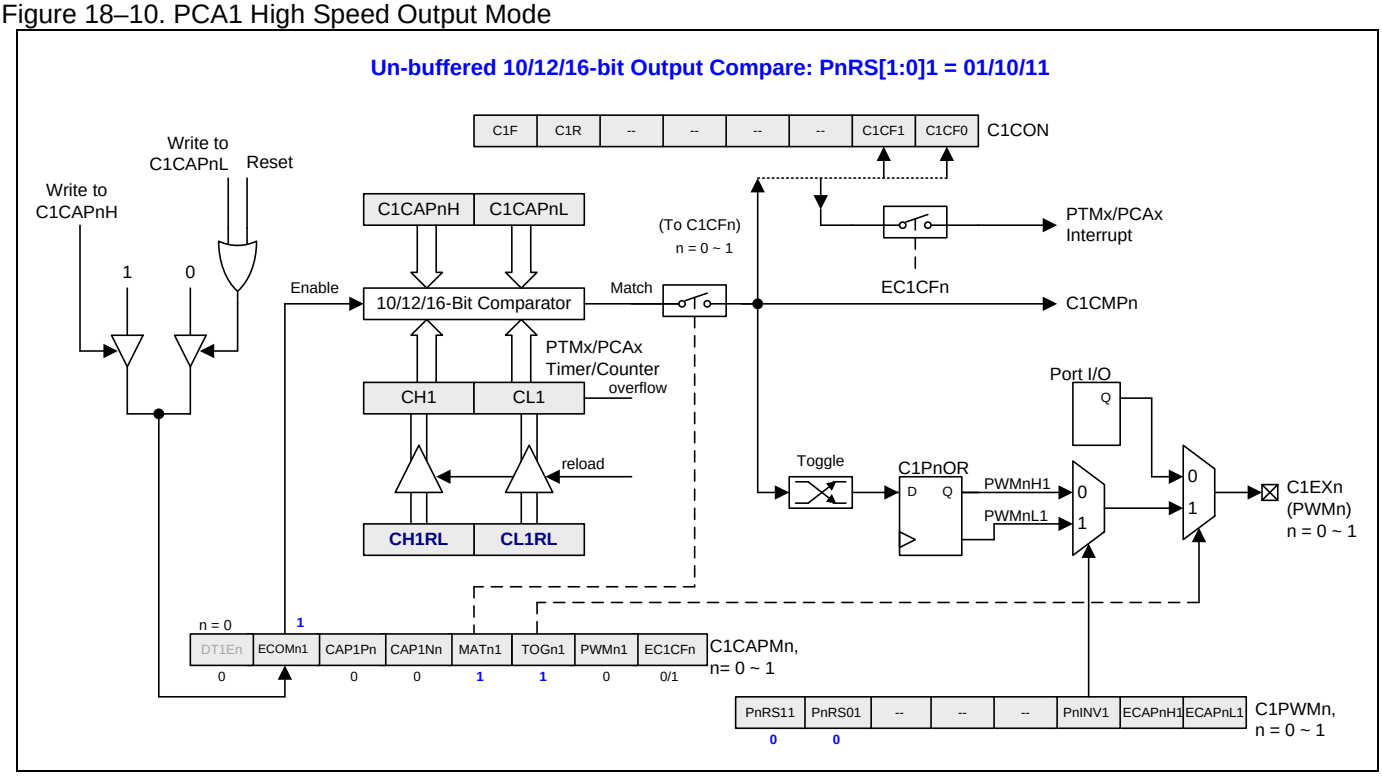
Figure 18–9. PCA1 High Speed Output Mode



18.4.7. 16-bit High Speed Output Mode (Output Compare mode)

In this mode the C1EX output associated with the PCA1 module will toggle each time a match occurs between the PCA1 counter and the module's capture registers. To activate this mode, the TOGn1, MATn1 and ECOMn1 bits in the module's C1CAPMn register must be set.

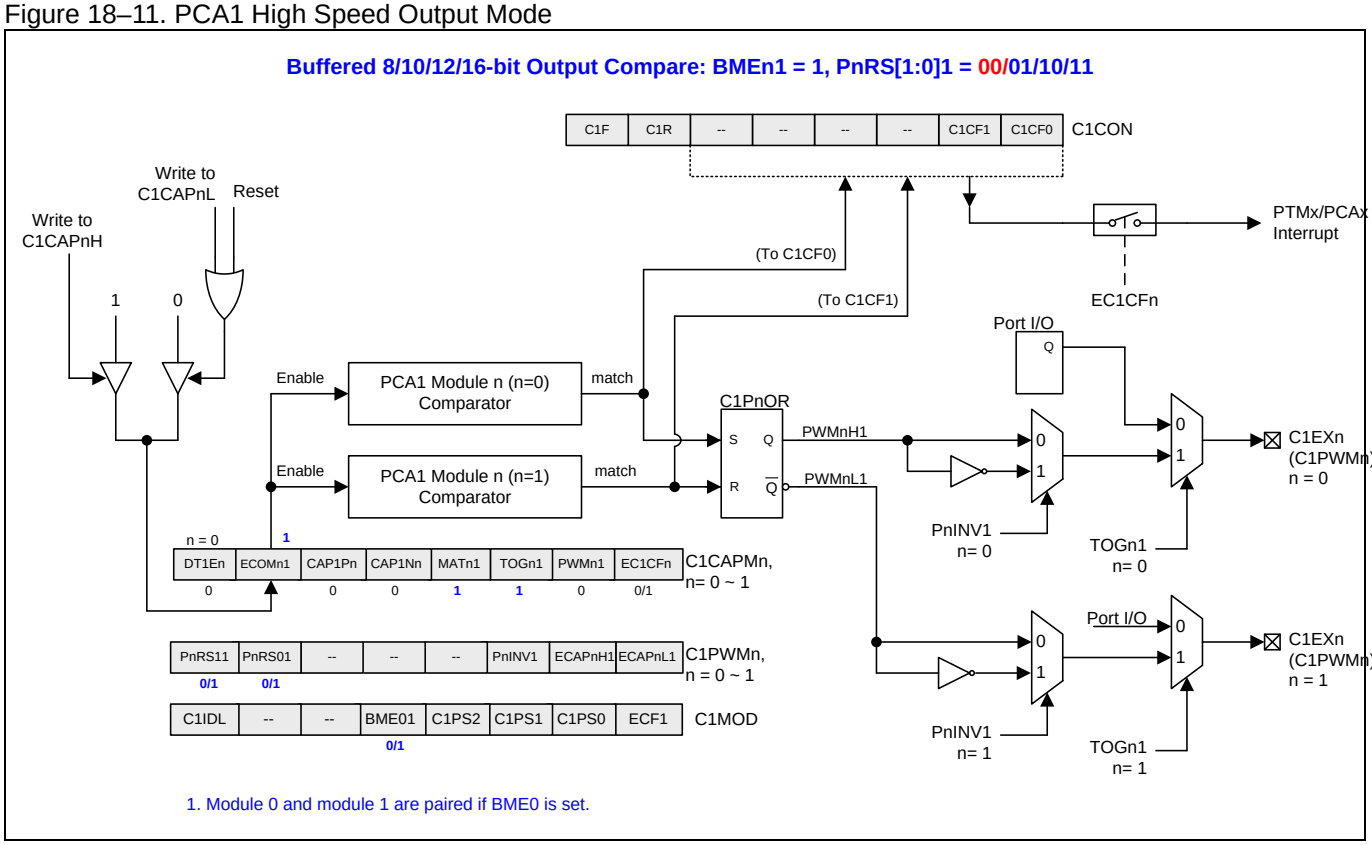
Figure 18–10. PCA1 High Speed Output Mode



18.4.8. Phase Variable Compare Output Mode

In this mode the C1EX output associated with the PCA1 module will toggle each time a match occurs between the PCA1 counter and the module's capture registers. To activate this mode, the TOGn1, MATn1 and ECOMn1 bits in the module's C1CAPMn register must be set.

Figure 18–11. PCA1 High Speed Output Mode



18.4.9. Buffered 8-bit PWM Mode

All of the PCA1 modules can be used as PWM outputs. The frequency of the output depends on the clock source for the PCA1 timer. All of the modules will have the same frequency of output because they all share the PCA1 timer.

The duty cycle of each module is determined by the module's capture register C1CAPnL and the extended 9th bit, ECAPnL1. When the 9-bit value of { 0, [CL1] } is *less than* the 9-bit value of { ECAPnL1, [C1CAPnL] } the output will be low, and if *equal to or greater than* the output will be high.

When CL1 overflows from 0xFF to 0x00, { ECAPnL1, [C1CAPnL] } is reloaded with the value of { ECAPnH1, [C1CAPnH] }. This allows updating the PWM without glitches. The PWMn and ECOMn1 bits in the module's C1CAPMn register must be set to enable the PWM mode.

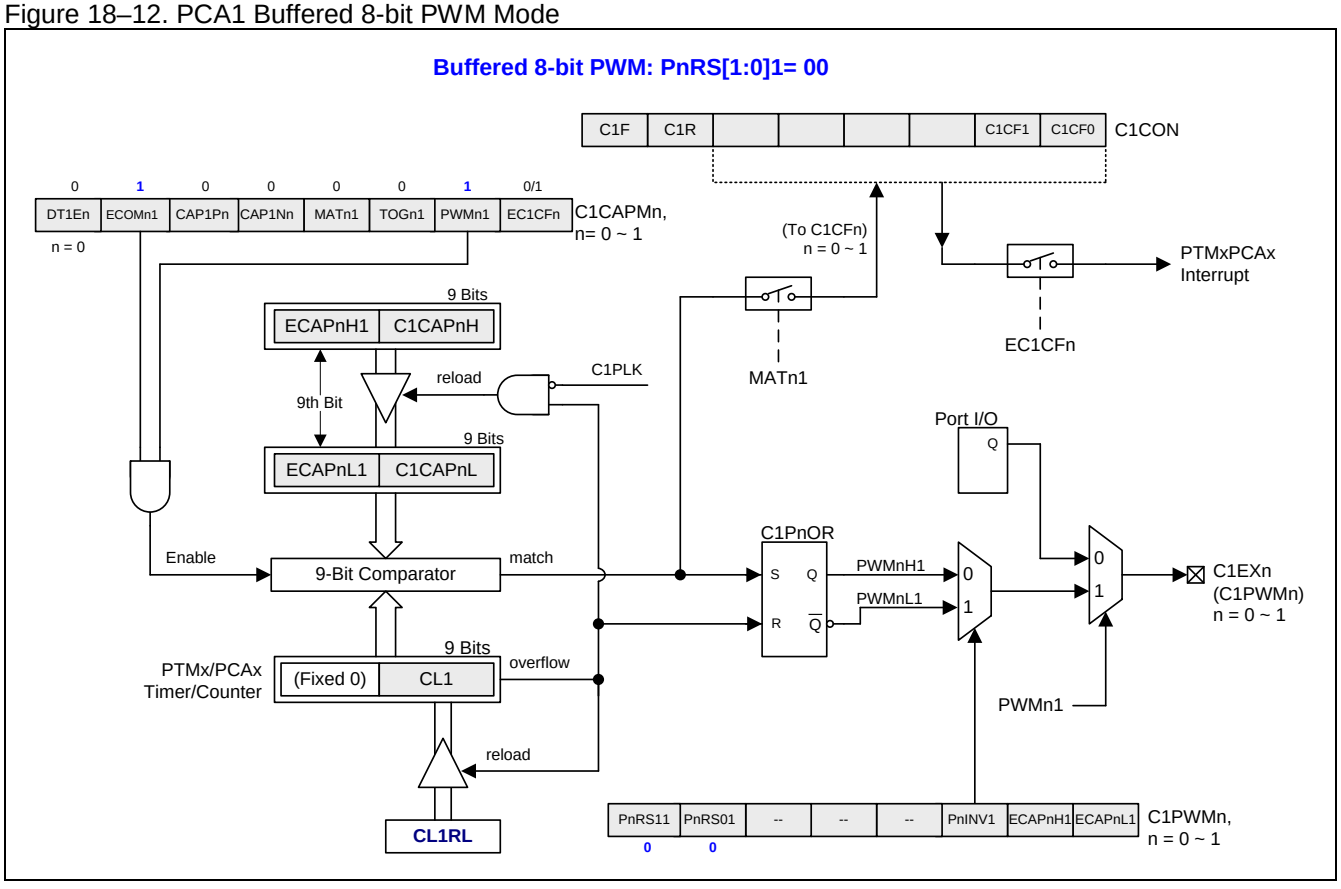
Using the 9-bit comparison, the duty cycle of the output can be improved to really start from 0%, and up to 100%. The formula for the duty cycle is:

Duty Cycle = 1 – { ECAPnH1, [C1CAPnH] } / 256.

Where, *[C1CAPnH]* is the 8-bit value of the C1CAPnH register, and *ECAPnH1* (bit-1 in the C1PWMn register) is 1-bit value. So, *{ ECAPnH1, [C1CAPnH] }* forms a 9-bit value for the 9-bit comparator.

For examples,
a. If ECAPnH1=0 & C1CAPnH=0x00 (i.e., 0x000), the duty cycle is 100%.
b. If ECAPnH1=0 & C1CAPnH=0x40 (i.e., 0x040) the duty cycle is 75%.
c. If ECAPnH1=0 & C1CAPnH=0xC0 (i.e., 0x0C0), the duty cycle is 25%.
d. If ECAPnH1=1 & C1CAPnH=0x00 (i.e., 0x100), the duty cycle is 0%.

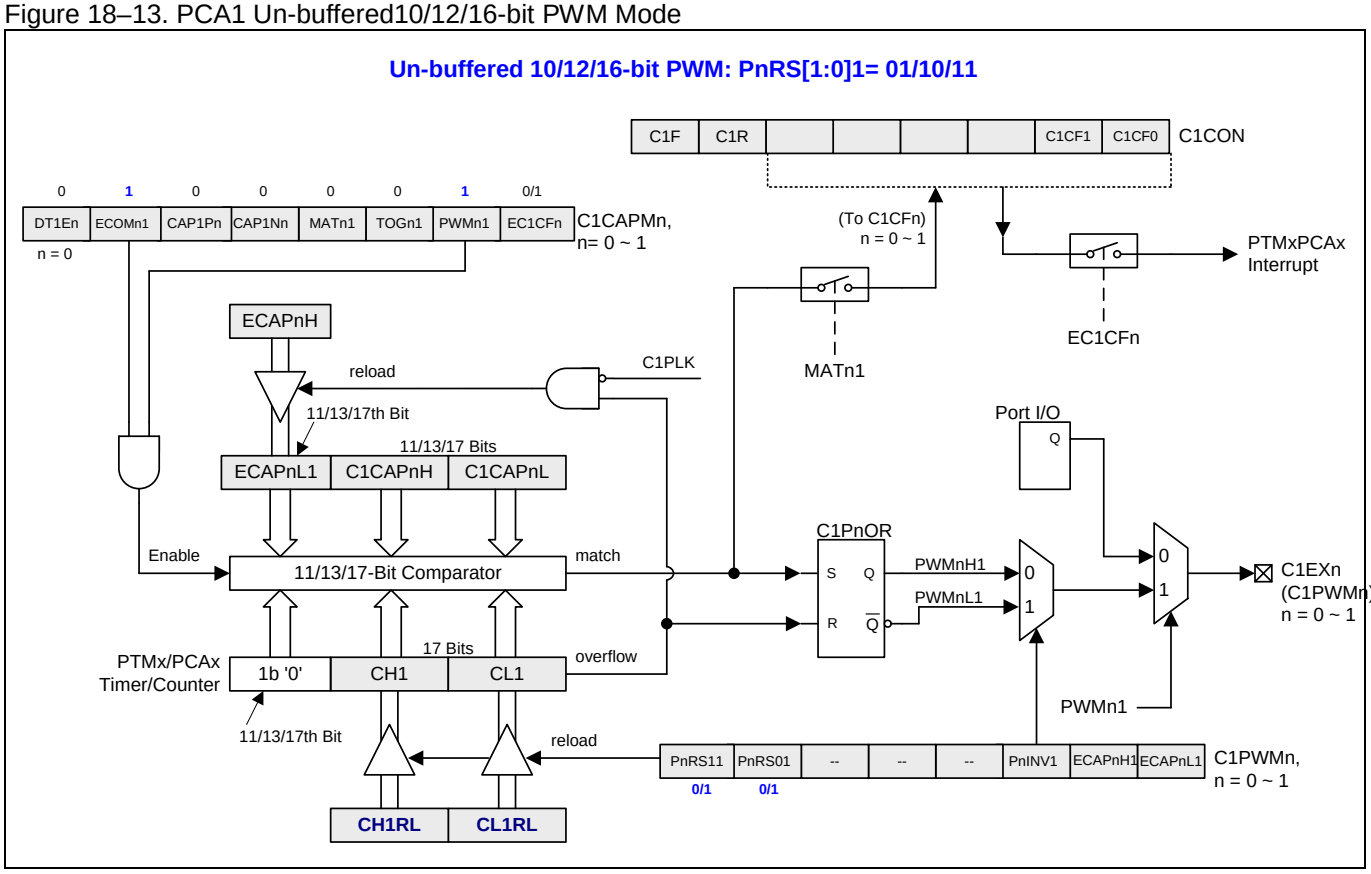
Figure 18–12. PCA1 Buffered 8-bit PWM Mode



18.4.10. Un-buffered 10/12/16-bit PWM Mode

The PCA1 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.

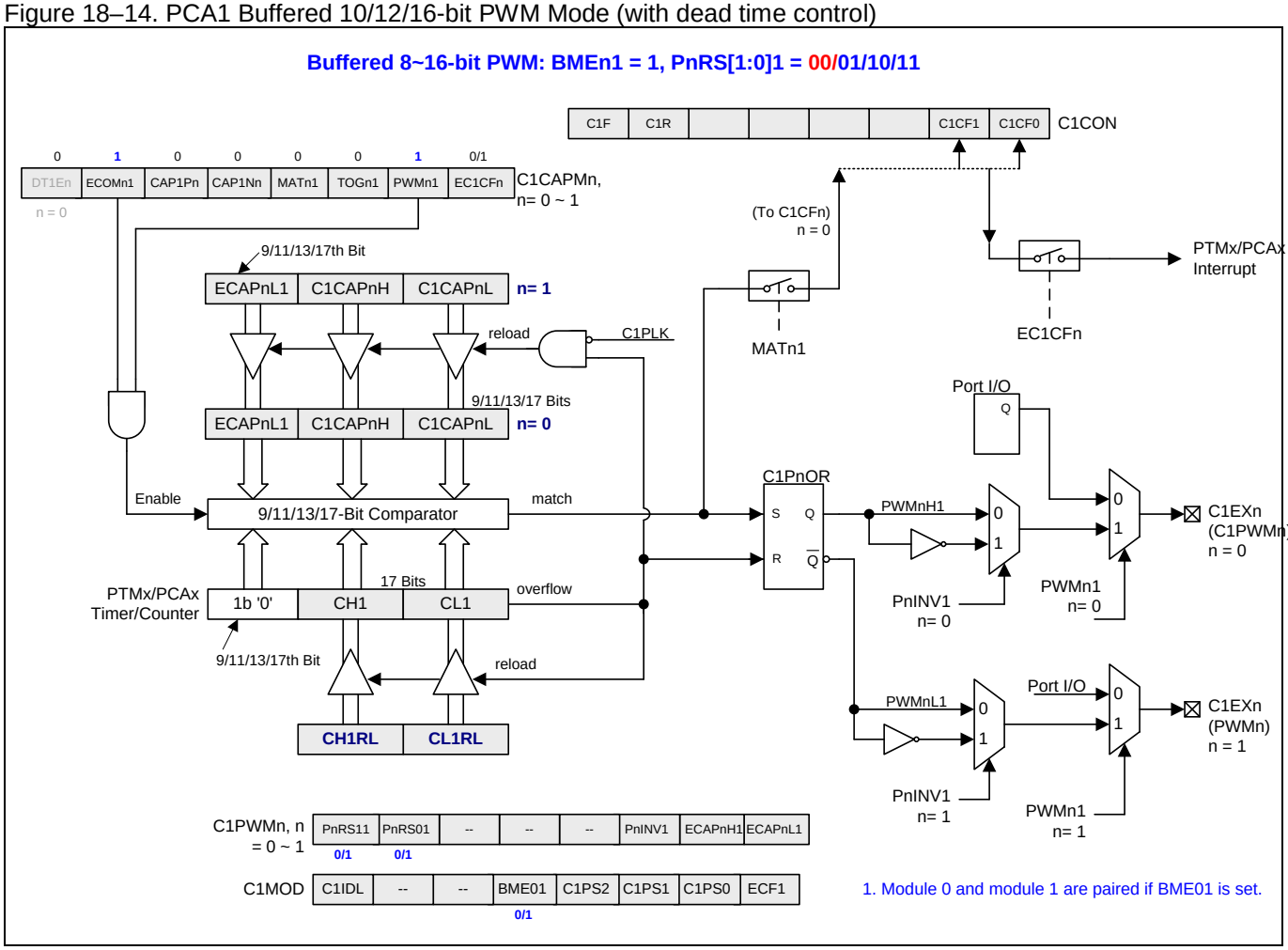
Figure 18–13. PCA1 Un-buffered10/12/16-bit PWM Mode



18.4.11. 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 C1CAPnH and C1CAPnL, 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 has been written. If the applications need accurate control when change the duty cycle, it needs to use the Buffered PWM mode.

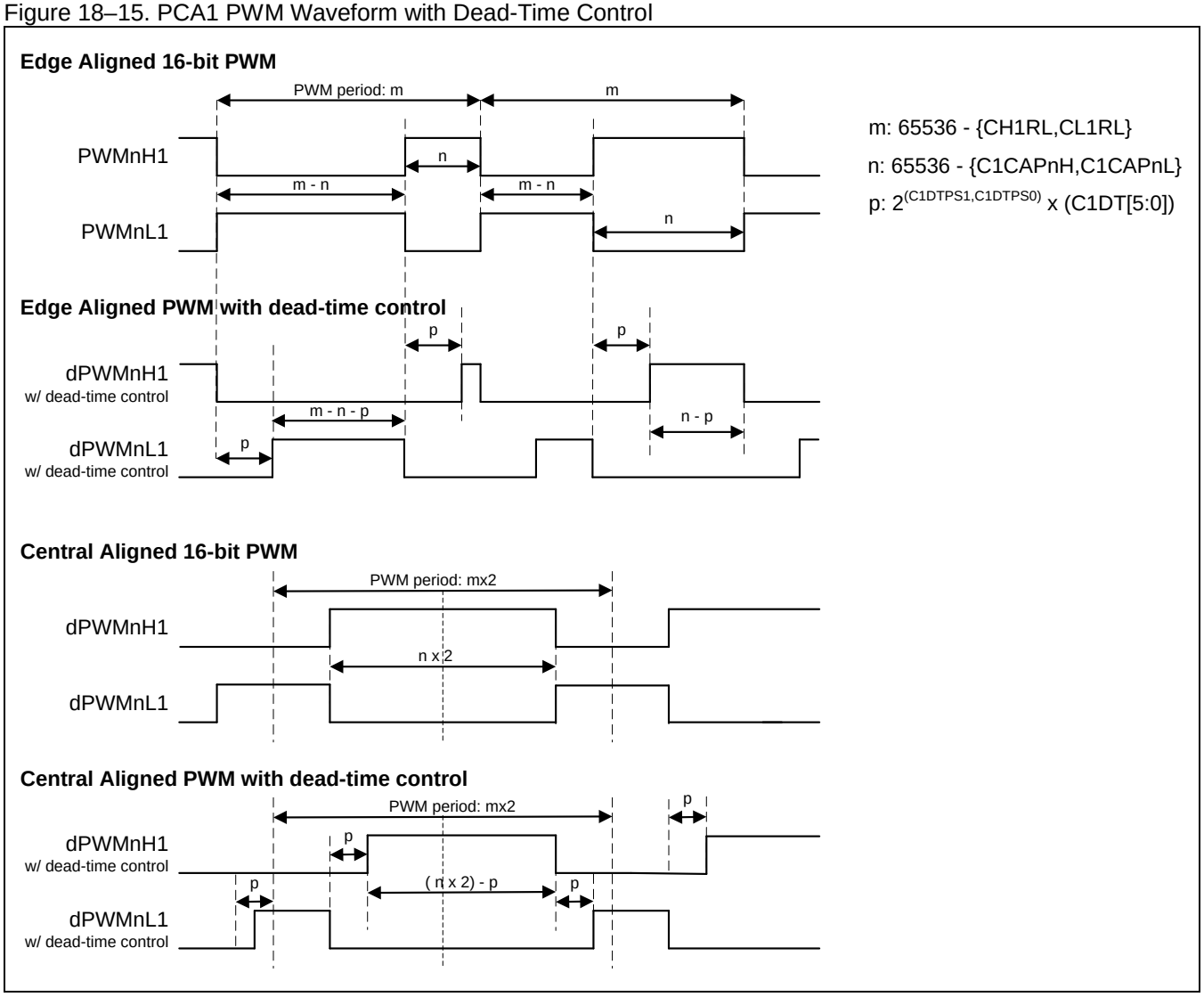
Figure 18–14. PCA1 Buffered 10/12/16-bit PWM Mode (with dead time control)



18.4.12. PCA1 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.

Figure 18–15. PCA1 PWM Waveform with Dead-Time Control



C1CAPMn: PCA Module Compare/Capture Register, n=0~1

SFR Page = 1 only

SFR Address = 0xDA~0xDF RESET = 0000-0000

7	6	5	4	3	2	1	0
DT1En	ECOMn1	CAP1Pn	CAP1Nn	MATn1	TOGn1	PWMn1	ECCFn1
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: DT1En. Enable Dead-Time control on PWMHn/PWMLn output pair. This bit is only valid on n= 0 and the dead-time function is active when PWM channel is operating in buffer mode. The channel buffer mode is enabled by BME01 in C1MOD.
0: Disable the Dead-Time control on PWMn output.
1: Enable the Dead-Time control on PWMn output.

C1PDTCR: PCA1 Dead-Time Control Register -A

SFR Page = **B only**

SFR Address = 0xBC RESET = 0000-0000

7	6	5	4	3	2	1	0
C1DTPS1	C1DTPS0	C1DT.5	C1DT.4	C1DT.3	C1DT.2	C1DT.1	C1DT.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~6: C1DTPS1~0, Clock Pre-Scaler of Dead-Time counter.

C1DTPS[1:0]	Pre-Scaler Selection
00	SYSCLK
01	SYSCLK/2
10	SYSCLK/4
11	SYSCLK/8

Bit 5~0: C1DT5~0, Dead-Time period control bits.

DT[5:0]	Dead-Time Period
000000	Dead-Time Disabled
000001	Pre-Scaler Clock X 1
000010	Pre-Scaler Clock X 2
000011	Pre-Scaler Clock X 3
…..	……
111110	Pre-Scaler Clock X 62
111111	Pre-Scaler Clock X 63

C1PWMCR: PAC1 Control Register

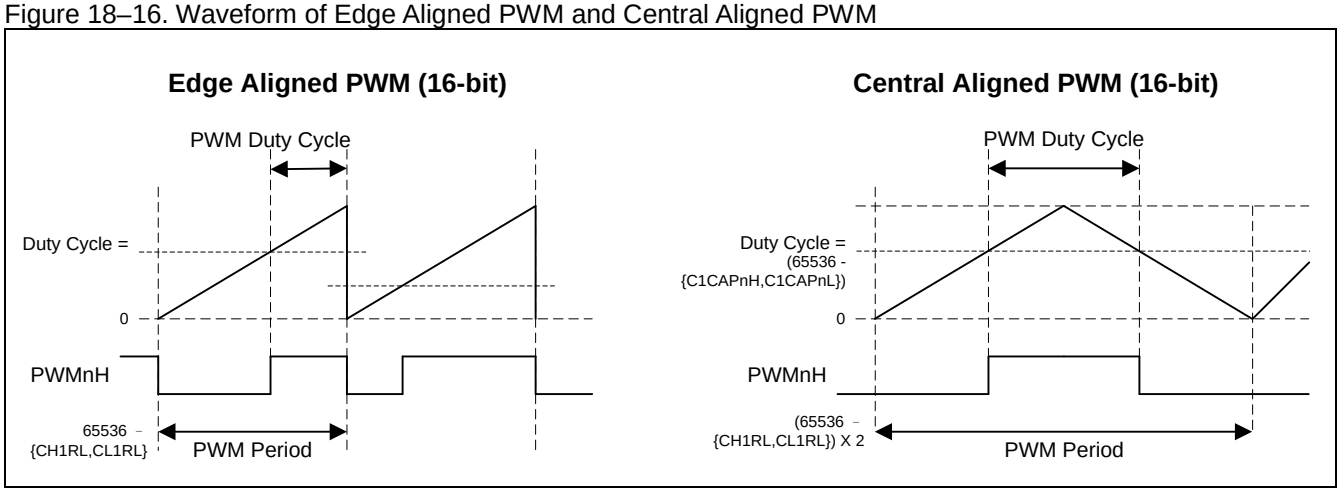
SFR Page = **A only**

SFR Address = **0**xBC RESET = 0000-0000

7	6	5	4	3	2	1	0
PCA1E	C1OFS	C1BKM	C1BKE1.1	C1BKE1.0	C1BKE0.2	C1BKE0.1	C1BKE0.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: PCA1E, PWM Central Aligned Enabled. PCA1E 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 PWMO0~1.
0: Set the PWM function with edge aligned modulation.
1: Enable the PWM function with central aligned modulation. It is recommended to set all PWM channel to same resolution.

Figure 18–16. Waveform of Edge Aligned PWM and Central Aligned PWM



Bit 6: C1OFS, PCA1 overflow flag selection.
0: CF is set on the top of central aligned PWM cycle.
1: CF is set on the bottom of central aligned PWM cycle.

Bit 5: C1BKM, PWM Break Mode selection.
0: Latched Mode.
1: Cycle-by-cycle Mode.

Figure 18–17. Latch Mode Waveform of PWM Break control

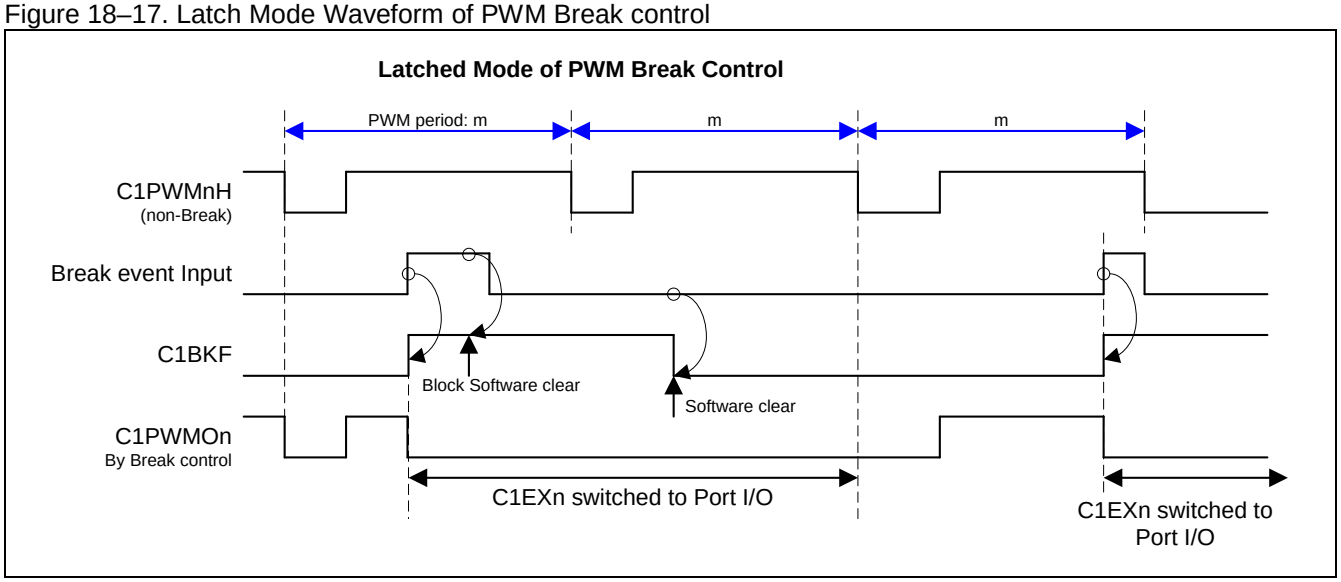
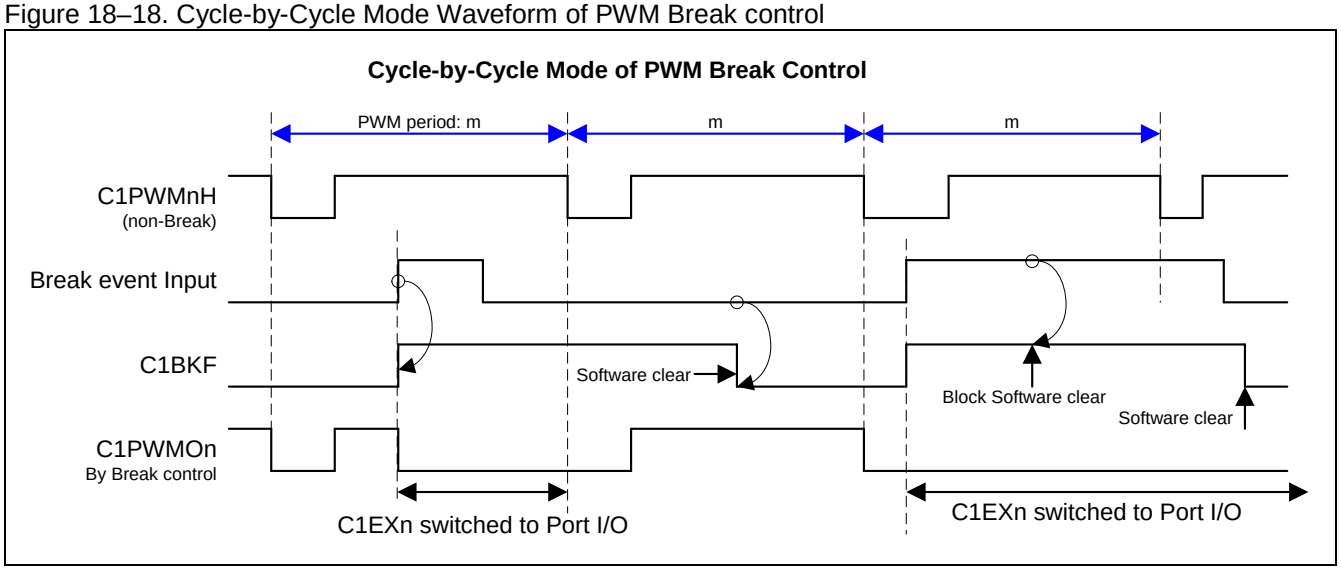


Figure 18–18. Cycle-by-Cycle Mode Waveform of PWM Break control



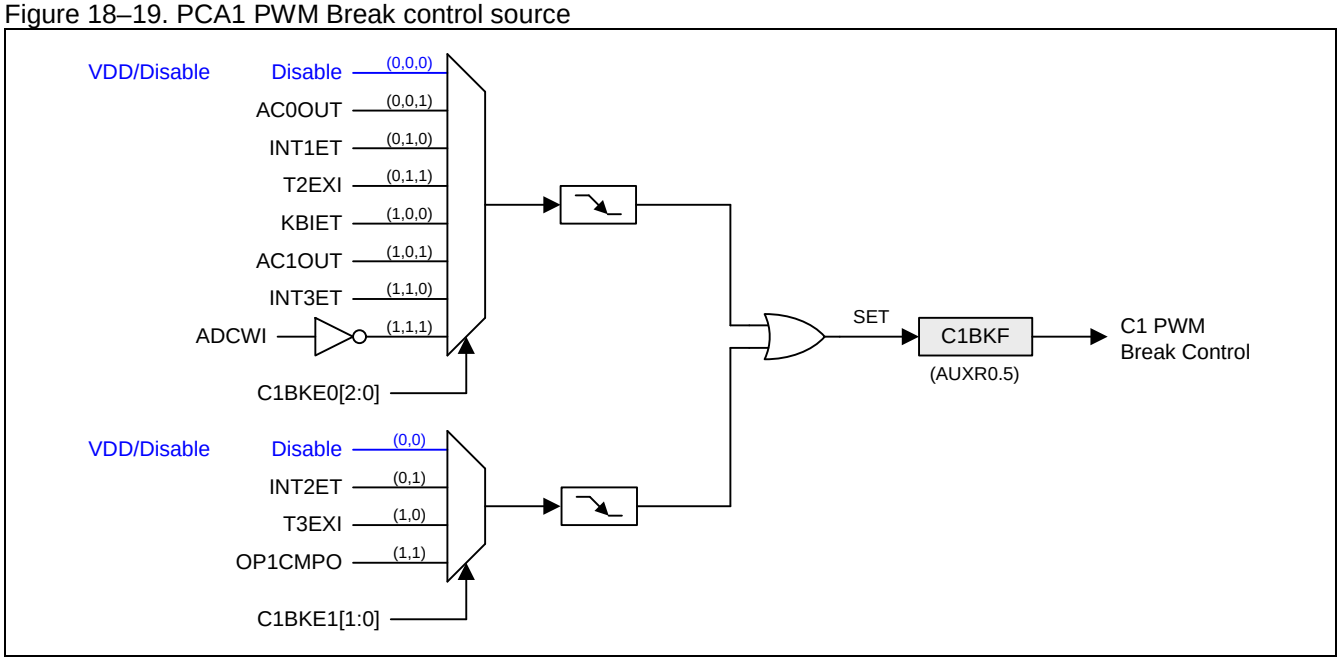
Bit 4~3: C1BKE1.1~0, PWM Break Enable 1 selection. This function is only active on CEXn output mode (n=0~5).

C1BKE1[1:0]	PWM Break Source
0 0	Disable PWM break source 1
0 1	INT2ET, nINT2 active
1 0	T3EXI
1 1	OP1ES

Bit 2~0: C1BKE0.2~0, PWM Break Enable 0 selection. This function is only active on CEXn output mode (n=0~5).

C1BKE0[2:0]	PWM Break Source
0 0 0	Disable PWM break source 0
0 0 1	AC0OUT
0 1 0	INT1ET, nINT1 active
0 1 1	T2EXI
1 0 0	KBIET, KBI match active
1 0 1	AC1OUT
1 1 0	INT3ET, nINT3 active
1 1 1	ADCWI active

Figure 18–19. PCA1 PWM Break control source



AUXR0: Auxiliary Register 0

SFR Page = 0~F

SFR Address = 0xA1 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	C1BKF	PBKF	0	0	INT1H	INT0H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

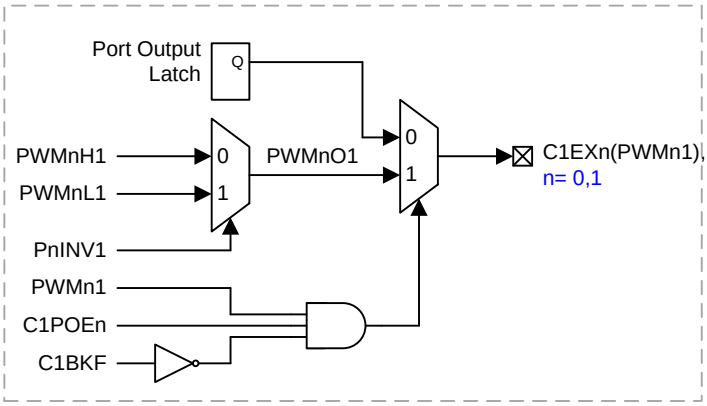
Bit 5: C1BKF, PCA1 PWM Break Flag. This bit is set by PWM break source enabled. If this flag is set, the enabled PWM channel 0~5 will be blocked, and the output pins keep the original GPIO state.
0: There is no PWM Break event happened. It is only cleared by software.
1: There is a PWM Break event happened or software triggers a PWM Break.

18.4.13. PCA1 Module Output Control

PCA1 modules have multi output control mode can be selected for different applications. The C1EXn (n=1) can be programed as general I/O port or the output of the PCA1 module (PWM) 1. When PWM has been assigned to the C1EXn, the PnINV1 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.

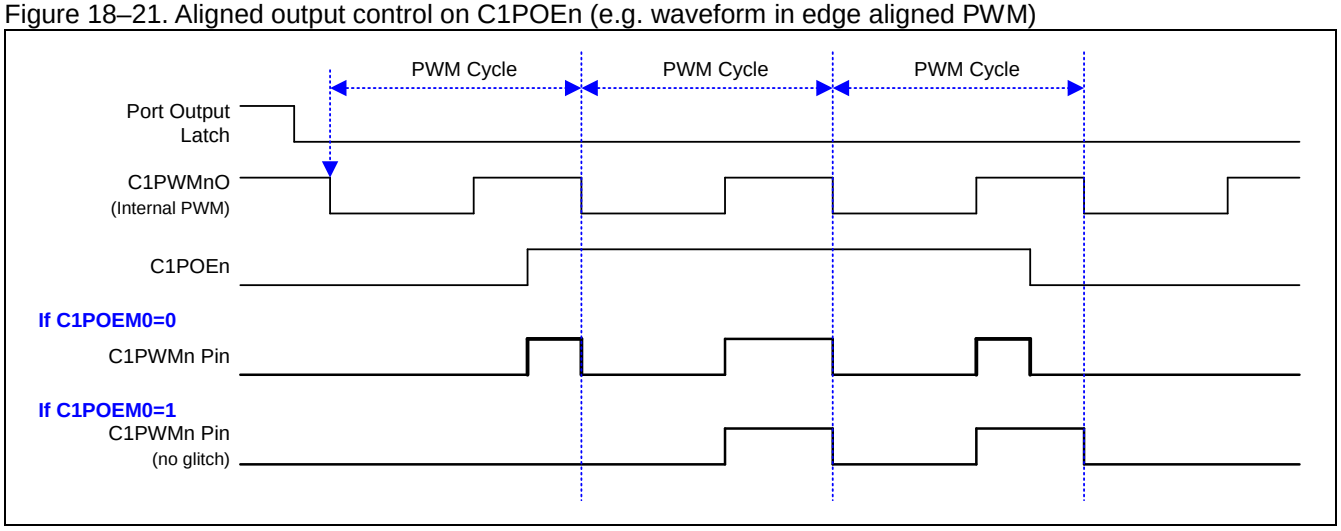
The C1EXn (n=1) also can use C1BKF, PWM Break Flag, to break PWM output. If this flag is set, the enabled PWM channel 0~5 will be blocked, and the output pins keep the original GPIO state.

Figure 18–20. PCA1 Module output control



C1POEM0 in **MG82F6P32** controls the C1POEn output timing to align with PWM cycle. The configuration and waveform of the aligned function is shown in Figure 17–19.

Figure 18–21. Aligned output control on C1POEn (e.g. waveform in edge aligned PWM)



C1AOE0: PWM Additional Output Enable Register

SFR Page = **8 only**

SFR Address = 0xF1 RESET = XXXX-1001

7	6	5	4	3	2	1	0
0	0	0	0	C1POE1	0	0	C1POE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~4: Reserved. Software must write “0” on these bits when C1AOE0 is written.

Bit 3: C1POE1, PCA1 PWM1 main channel (C1PWM1O) output control.
0: Disable C1PWM1O output on port pin.
1: Enable C1PWM1O output on port pin. **Default is enabled.**

Bit 2~1: Reserved. Software must write “0” on these bits when C1AOE0 is written.

Bit 0: C1POE0, PCA1 PWM0 main channel (C1PWM0O) output control.
0: Disable C1PWM0O output on port pin.
1: Enable C1PWM0O output on port pin. **Default is enabled.**

C1AOE1: PWM Additional Output Enable Register

SFR Page = **A**

SFR Address = 0xF1 RESET = 1XXX-XXXX

7	6	5	4	3	2	1	0
C1POEM0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: C1POEM0, PCA1 C1POEn control 0.
0: C1POEn function is active immediately after CPU writing.
1: C1POEn function is aligned to PWM cycle.

Bit 6~0: Reserved. Software must write “0” on these bits when C1AOE1 is written.

AUXR7: Auxiliary Register 7

SFR Page = 4 only

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	C0CKOE	C1CKOE	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 4: C1CKOE, PCA1 clock output (C1CKO) enable.
0: Disable PCA1 clock output.
1: Enable PCA1 clock output with PCA1 base timer overflow rate/2.

AUXR8: Auxiliary Register 8

SFR Page = 5 only

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	C1ICS1	C1ICS0	0	0	S1COPS	T3PS1	T3PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 6: C1ICS1, PCA1 Input Channel 1 input port pin Selection.

C1ICS1	C1EX1 input
0	C1EX1 Port Pin
1	T2EXI

Bit 5: C1ICS0, PCA1 Input Channel 0 input port pin Selection.

C1ICS0	C1EX0 input
0	C1EX0 Port Pin
1	T3EXI

AUXR17: Auxiliary Register 17

SFR Page = E only

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	0	0	0	0	C0OPS1	C0OPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 1~0: C1OPS1 ~0, PCA1 Output Pin Selection.

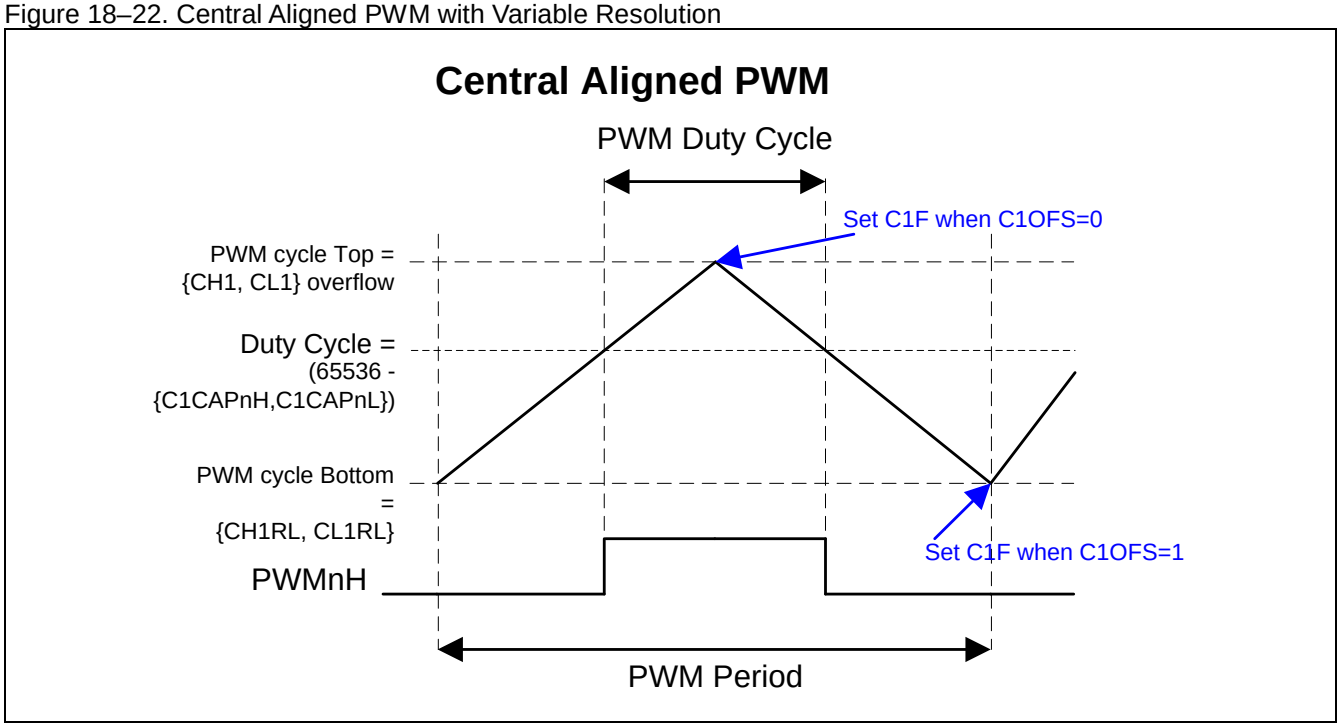
	C1OPS1	C1OPS0
C1OPSn	C1EX1	C1EX0
0	P6.1	P6.0
1	P3.1	P3.0

18.4.14. 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 PCA1 functions, capture or compare, on other non-PWM modules are still available.

Please note when using Central Aligned PWM, 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 18–22. Central Aligned PWM with Variable Resolution



C1PWMCR: PCA1 PWM Control Register

SFR Page = **A only**

SFR Address = 0xBC RESET = 0000-0000

7	6	5	4	3	2	1	0
PCA1E	C1OFS	C1BKM	C1BKE1.1	C1BKE1.0	C1BKE0.2	C1BKE0.1	C1BKE0.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 6: C1OFS, PCA1 overflow flag selection.
0: C1F is set on the top of central aligned PWM cycle.
1: C1F is set on the bottom of central aligned PWM cycle.

18.4.15. PWM Global control

PTM0, PCA1 and Timer 2~4 can use the global control method to sync the frequency and phase. Please reference “16.4 Timer Global Control”.

19. Serial Port 0 (UART0)

The serial port 0 of **MG82F6P32** supports 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 receives 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 receive register.

19.1. Serial Port 0 Mode Selection

The serial port can operate in **5** standard modes and **8** enhance 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. For the enhance modes please reference section 19.11

Table 19–1. Serial Port 0 Mode Selection

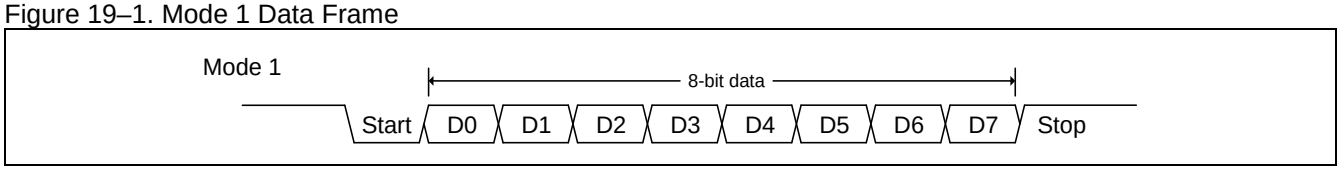
SM30,SM00, SM10	S0RCK	S0TCK	MODE	Function	Baud Rate Time Base	
000	0	0	0	shift register	SYSCLK/12 or SYSCLK/4	
001	0	0	1	8-bit UART	Timer 1 or Timer 2 overflow	
010	0	0	2	9-bit UART	SYSCLK/64, /32, or /16	
011	0	0	3	9-bit UART	Timer 1 or Timer 2 overflow	
100	0	0	4	SPI Master	SYSCLK/12 or SYSCLK/4 (SMOD1=1)	
000	0	1	Enhanced	shift register	S0BRG overflow	
000	1	1	Enhanced	ARGB	S0BRG overflow	
001	0/1	0/1	Enhanced	8-bit UART	Selectable S0BRG overflow on TX or RX	
010	0	1	Enhanced	9-bit UART	TX: S0BRG overflow RX: SYSCLK/64, /32 or /16	
010	1	0	Enhanced	9-bit UART	TX: SYSCLK/64, /32 or /16 RX: S0BRG overflow	
010	1	1	Enhanced	Pure Timer	Only Timer function	
011	0/1	0/1	Enhanced	9-bit UART	Selectable S0BRG overflow on TX or RX	
100	0	1	Enhanced	SPI Master	S0BRG overflow	
Others				Reserved.		

Note: Mode 0 ~ 4 use the default value of S0RCK and S0TCK, for the reset enhance mode please reference section 19.11 for detail description.

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 SMOD1 setting in PCON0 register. In **MG82F6P32**, 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 19–4 and Figure 19–5 show the clock polarity waveform in Mode 0.

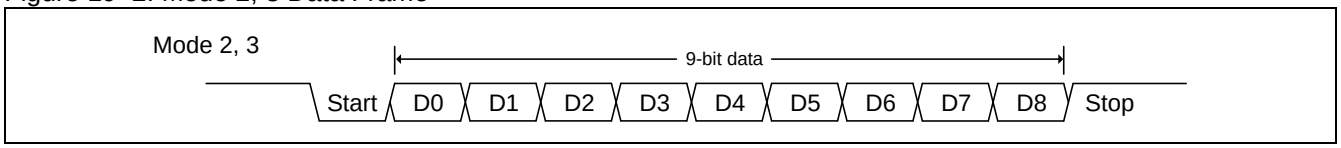
Mode 1: 10 bits are transmitted through TXD0 or received through RXD0. The frame data includes a start bit (0), 8 data bits (LSB first), and a stop bit (1), as shown in Figure 19–1. On receive, the stop bit would be loaded into RB80 in S0CON register. The baud rate is variable.

Figure 19–1. Mode 1 Data Frame



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 19–2. 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 19–2. 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.

19.2. 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 SMOD1 setting in PCON0 register.

Figure 19–3 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 initial status of the Tx can be defined by SM20. When SM20 =0, the initial level of Tx is high. And if SM20 = 1, it will be low. 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 19–4 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. The initial status of the Rx can be defined by S0DOR. When S0DOR =1, the initial level of Tx is high. And if S0OR = 0, it will be low.

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 19–5 shows the reception waveform in Mode 0.

Figure 19–3. Serial Port 0 Mode 0

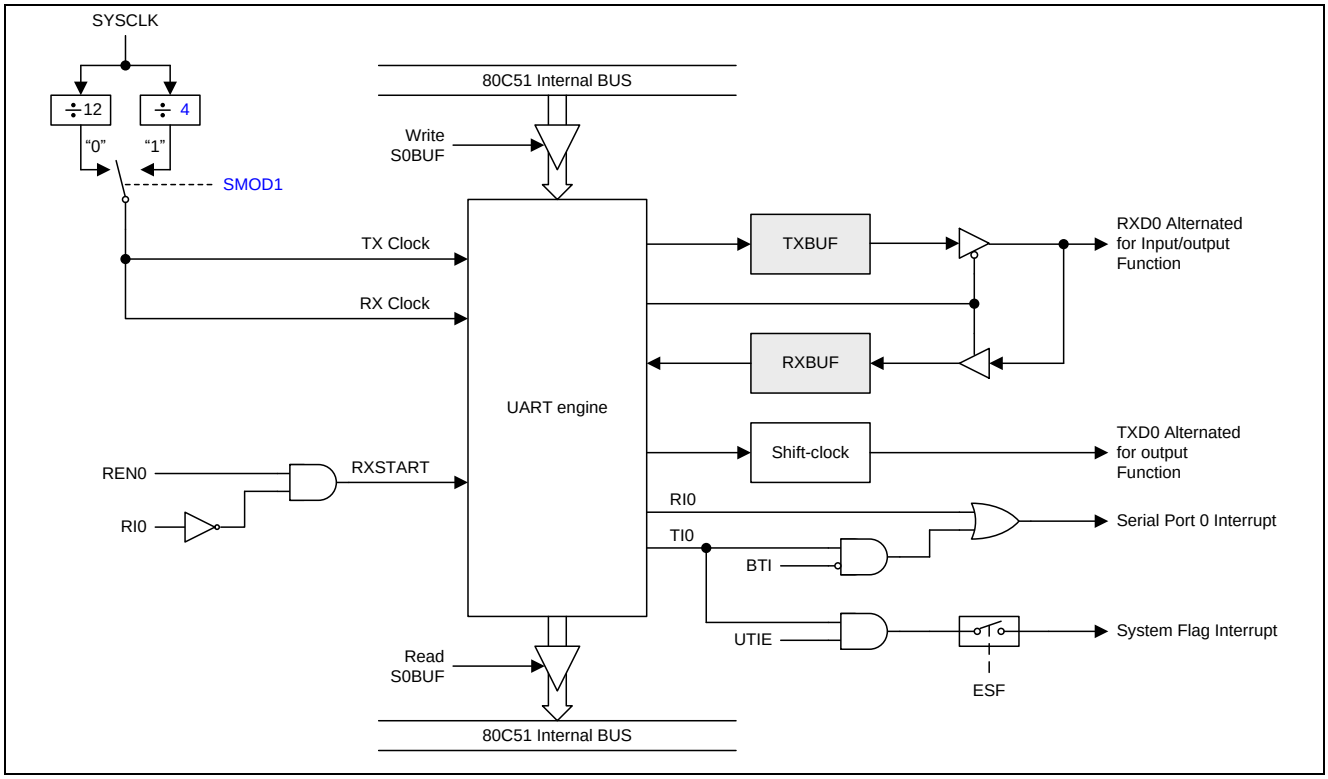


Figure 19–4. Mode 0 Transmission Waveform

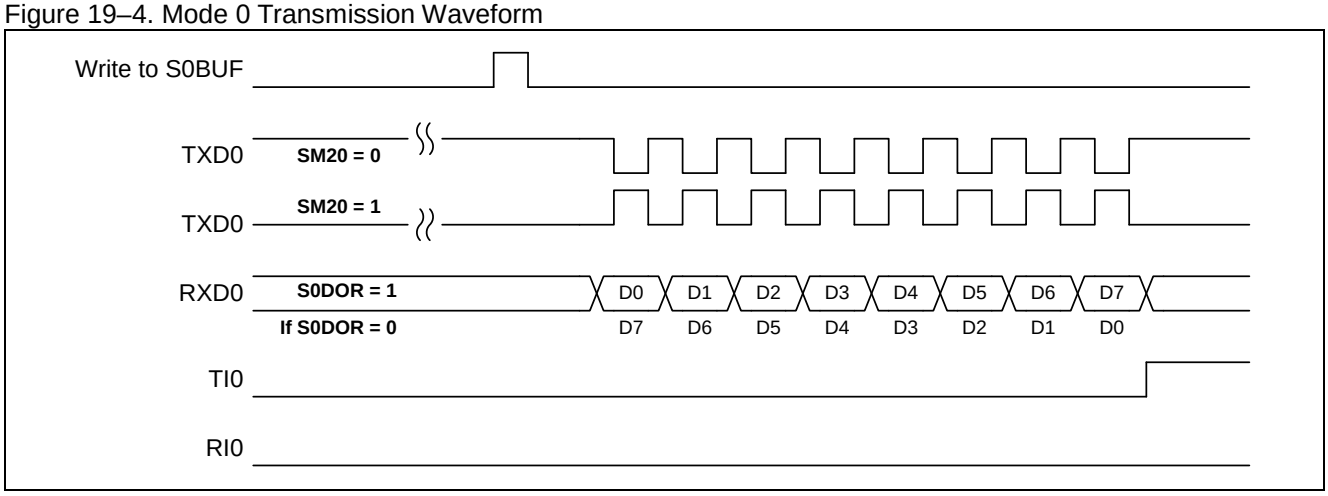
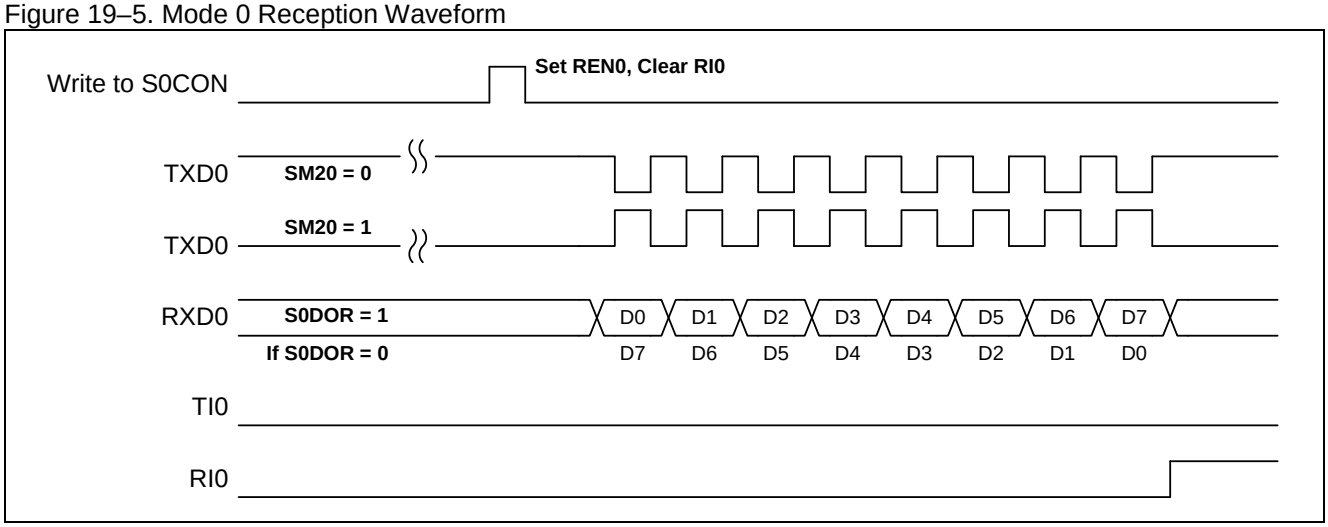


Figure 19–5. Mode 0 Reception Waveform



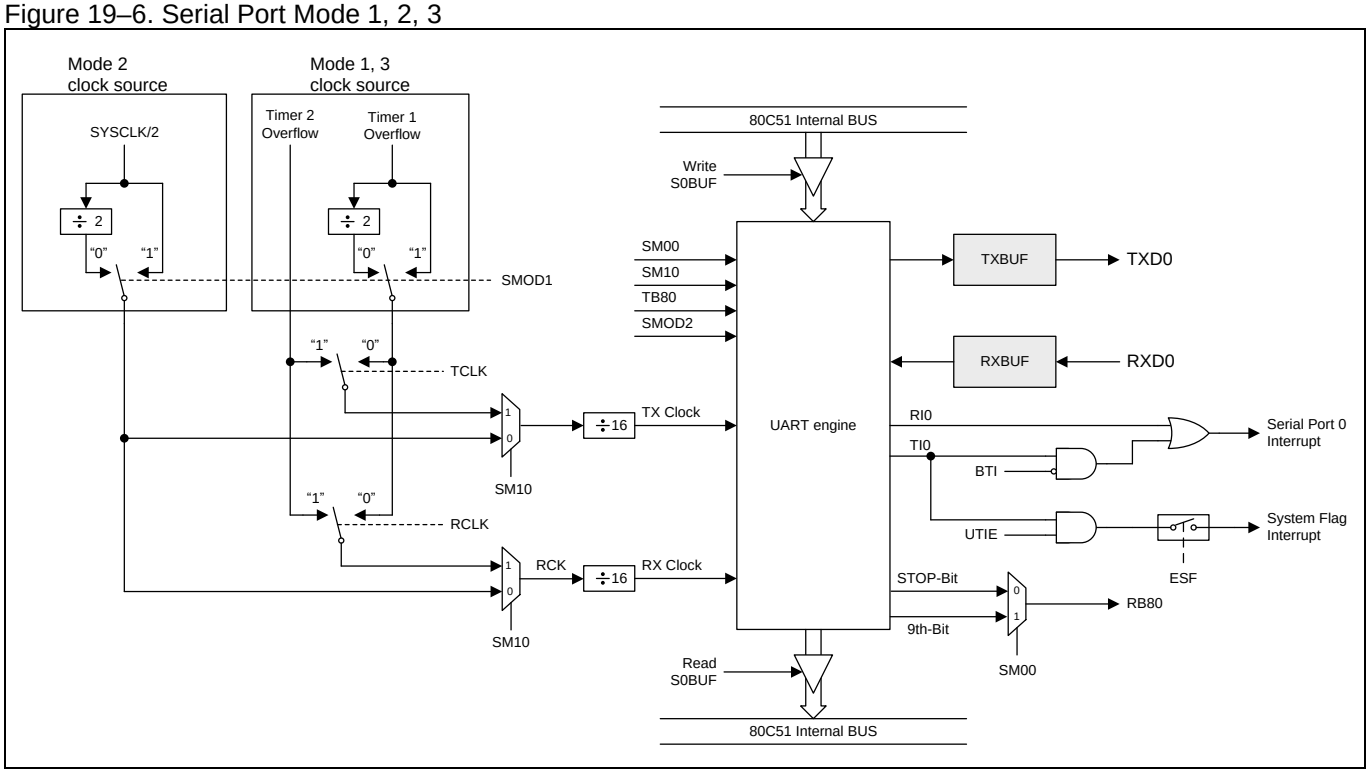
19.3. Serial Port 0 Mode 1

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 19–1 shows the data frame in Mode 1 and Figure 19–6 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 19–1 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 19–6. Serial Port Mode 1, 2, 3



19.4. 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 19–2 shows the data frame in Mode 2 and Mode 3. Figure 19–5 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 shit register and starts 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 19–2 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.

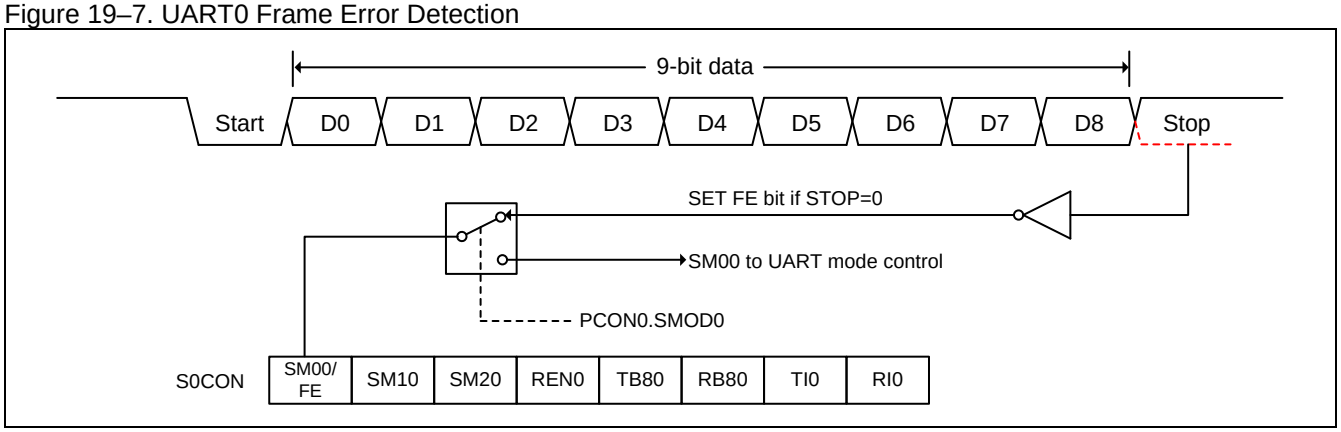
Reception is initiated when the UART0 engine detected 1-to-0 transition at RXD0 sampled by RCK. The data on the RXD0 pin would be sampled by Bit Detector in UART0 engine. After the end of 9th data bit reception, RI0 would be asserted by hardware to indicate the end of data reception and load the 9th 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.

19.5. 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 19–7.

Figure 19–7. UART0 Frame Error Detection



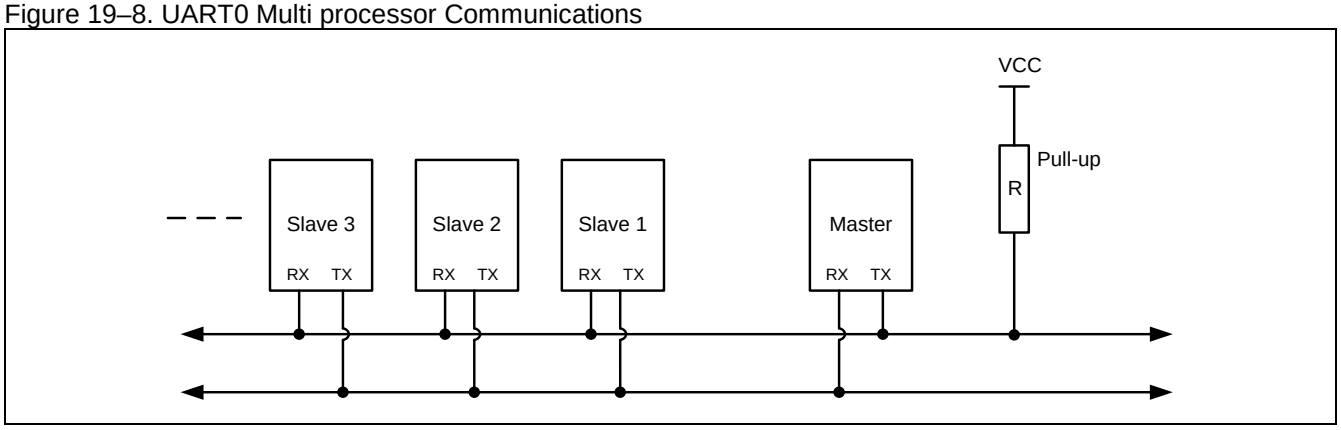
19.6. Multiprocessor Communications

Modes 2 and 3 have a special provision for multiprocessor communications as shown in Figure 19–8. 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 19–8. UART0 Multi processor Communications

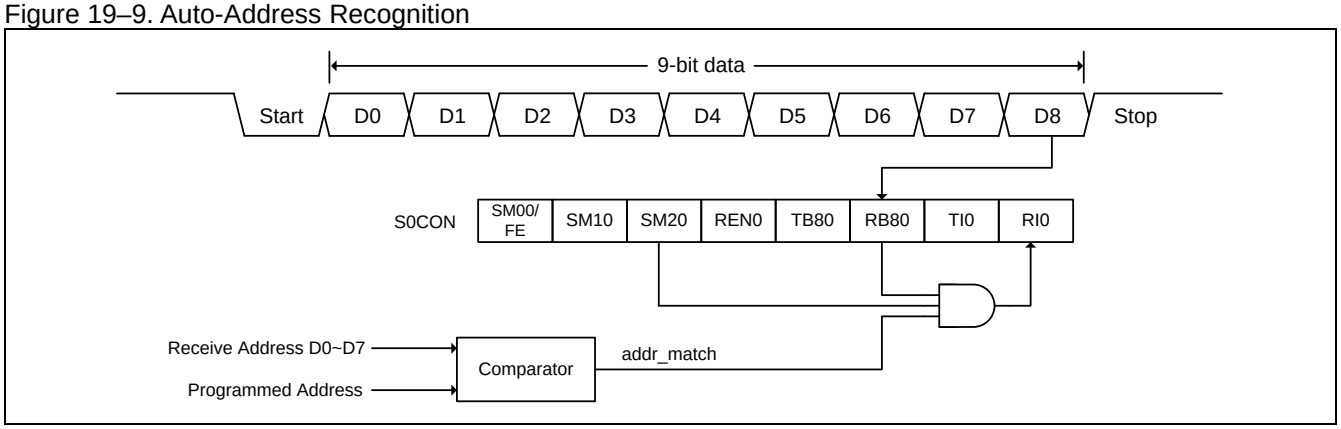


19.7. Automatic Address Recognition

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 19–9.

Figure 19–9. 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.

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 the bits in the SADDR which bits are available, and the remaining bits are "don't care". To use the SADEN mask, perform a logical AND operation on SADDR to create a "Given" address that will be used as the address of slave devices and the master can send the "Given" address on bus to identify the slave out from multiple slaves.

The following examples will help to show the versatility of this scheme:

Slave 0	**Slave 1**
SADDR = 1100 0000	SADDR = 1100 0000
SADEN = 1111 1101	SADEN = 1111 1110
Given = 1100 00X**0**	Given = 1100 00**0**X

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:

Slave 0	**Slave 1**	**Slave 2**
SADDR = 1100 0000	SADDR = 1110 0000	SADDR = 1110 0000
SADEN = 1111 1001	SADEN = 1111 1010	SADEN = 1111 1100
Given = 1100 0XX**0**	Given = 1110 0X**0**X	Given = 1110 0**0**XX

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.

When RESET, the SADDR (SFR address 0xA9) and SADEN (SFR address 0xB9) are loaded with 0s. This generates a "Given" address as "don't care" ("XXXXXXXXb") and a "Broadcast" address as "don't care" ("XXXXXXXXb"). 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.

19.8. Serial Port 0 Mode 4 (SPI Master)

The Serial Port 0 of **MG82F6P32** is embedded an additional Mode 4 to support SPI master engine. The Mode 4 is selected by SM30, SM00 and SM10. Please reference "Table 19–1. Serial Port 0 Mode Selection".

SMOD1 also controls the SPI transfer speed. If SMOD1 = 0, the SPI clock frequency is SYSCLK/12. If SMOD1 = 1, the SPI clock frequency is SYSCLK/4. When choose S0BRG as clock source the SPI clock frequency is not controlled by SMOD1, it is equal S0TOF/4.

The SPI master in **MG82F6P32** uses the TXD0 as SPICLK, RXD0 as MOSI, and S0MI as MISO. nSS is selected by MCU software on other port pin. Figure 19–10 shows the SPI connection. It also can support the configuration for multiple slave communication in Figure 19–11.

Figure 19–10. Serial Port 0 Mode 4, Single Master and Single Slave configuration (n = 0)

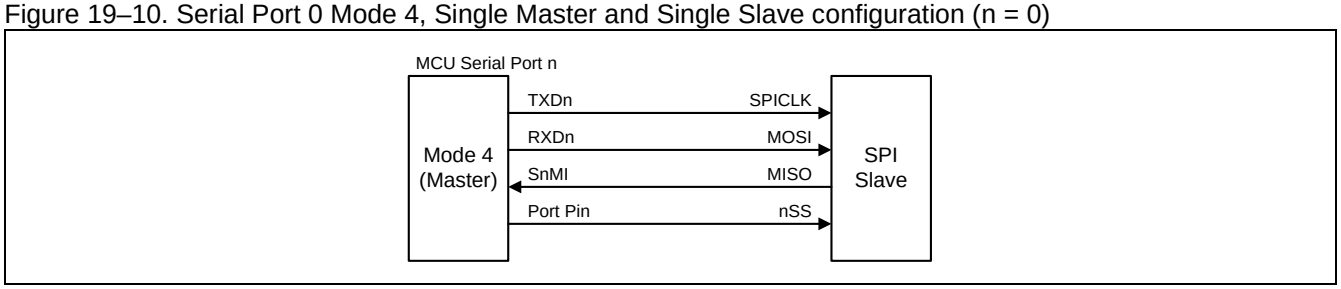
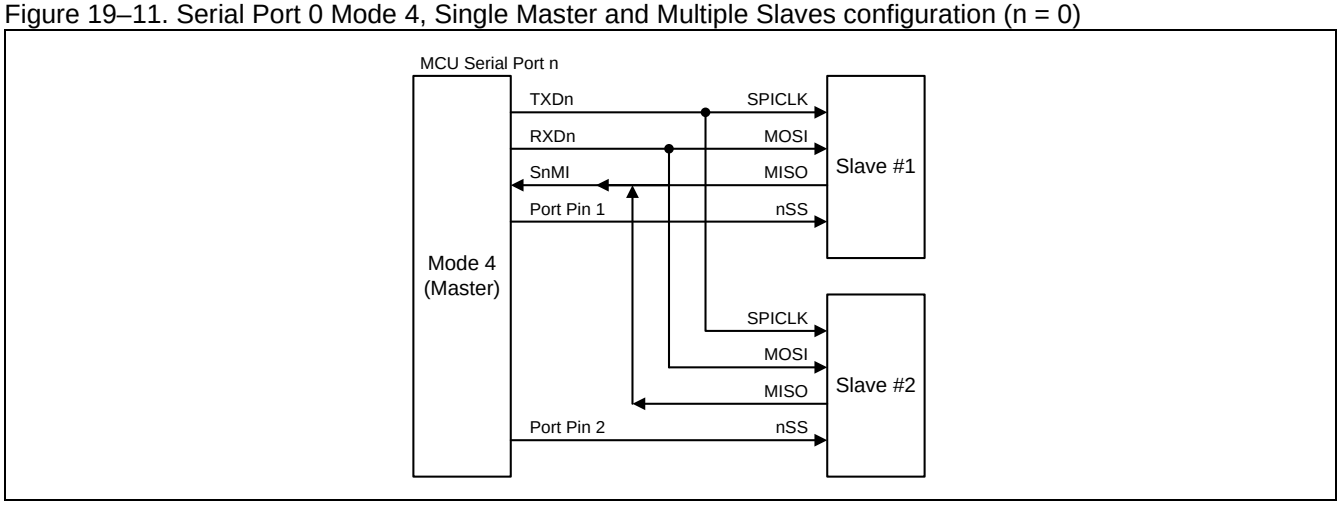


Figure 19–11. Serial Port 0 Mode 4, Single Master and Multiple Slaves configuration (n = 0)



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, **MG82F6P32** uses an easy way by initialize SPI clock assigned port pin (TXD0) polarity to fit them. Table 19–2 shows the serial port Mode 4 mapping with the four SPI operating mode.

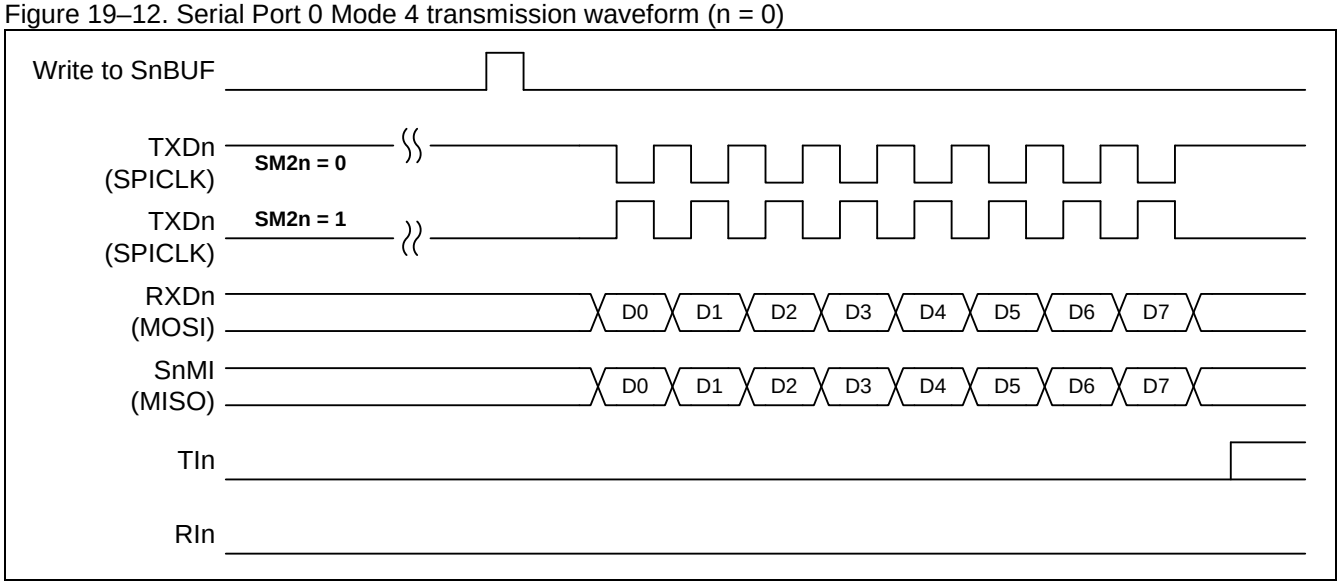
Table 19–2. SPI mode mapping with Serial Port Mode 4 configuration

SPI Mode	CPOL	CPHA	Configure SM2n & TB8n in SnCON
0	0	0	SM2n = 1, TB8n = 1
1	0	1	SM2n = 1, TB8n = 0
2	1	0	SM2n = 0, TB8n = 1
3	1	1	SM2n = 0, TB8n = 0

For bit order control (DORD) on SPI serial transfer, **MG82F6P32** 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. The initial status of the TxD can be defined by SM2n. When SM2n =0, the initial level of TxD is high. And if SM2n = 1, it will be low. 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 19–12 shows the transmission waveform in Mode 0. RI0 will not be asserted in Mode 4.

Figure 19–12. Serial Port 0 Mode 4 transmission waveform (n = 0)



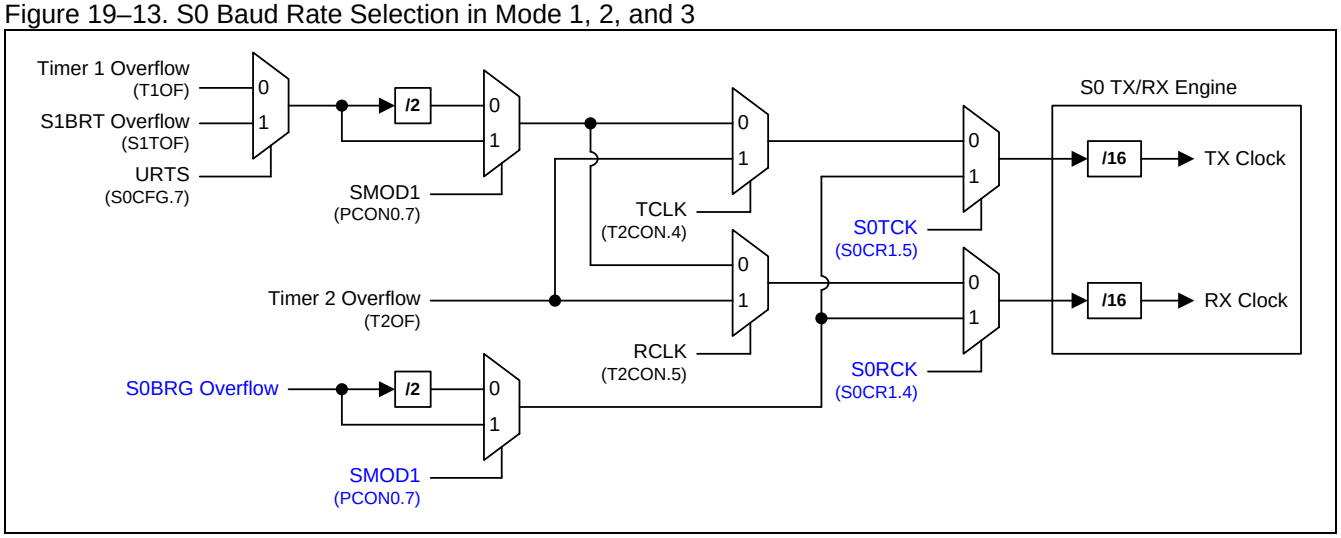
19.9. Baud Rate Setting

Bits T2X12 (T2MOD.4), T1X12 (AUXR2.3), and SMOD2 (S0CFG.6) provide a new option for the baud rate setting, as listed below.

19.9.1. 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 19–13. S0 Baud Rate Selection in Mode 1, 2, and 3



19.9.2. Baud Rate (Serial clock frequency) in Mode 0 & 4

S0 Standard Mode 0 & 4 Baud Rate equation

$$\text{Baud Rate} = \frac{F_{SYSCLK}}{12} \quad ; \text{SMOD1=0}$$

$$\text{or} = \frac{F_{SYSCLK}}{4} \quad ; \text{SMOD1=1}$$

Note:

*If **SMOD1**=0, the baud rate formula is as same as standard 8051.*

S0 Enhanced Mode 0 & 4 Baud Rate equation (Clock source from S0BRG)

$$\text{Baud Rate} = \frac{F_{SYSCLK}}{4 \times (256 - \text{S0BRT})}$$

19.9.3. Baud Rate in Mode 2

$$\text{Mode 2 Baud Rate} = \frac{2^{SMOD1} \times 2^{(SMOD2 \times 2)}}{64} \times F_{SYSCLK}$$

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 19–3 defines the Baud Rate setting with SMOD2 factor in Mode 2 baud rate generator.

Table 19–3. SMOD2 application criteria in Mode 2

SMOD2	SMOD1	Baud Rate	Note	Recommended Max. Receive Error (%)
0	0	Default Baud Rate	Standard function	± 3%
0	1	Double Baud Rate	Standard function	± 3%
1	0	Double Baud Rate **X2**	Enhanced function	± 2%
1	1	Reserved		

Table 19–4. S0 Mode 2 Baud Rates @ F_{SYSCLK}=11.0592MHz

Baud Rate	SMOD2	SMOD1	Error
172,800	0	0	0.0%
345,600	0	1	0.0%
691,200	1	0	0.0%

Table 19–5. S0 Mode 2 Baud Rates @ F_{SYSCLK}=12.00MHz

Baud Rate	SMOD2	SMOD1	Error
187,500	0	0	0.0%
375,000	0	1	0.0%
750,000	1	0	0.0%

19.9.4. Baud Rate in Mode 1 & 3

19.9.4.1. Using Timer 1 and SnBRG as the Baud Rate Generator

Depends on Figure 19–13. S0 Baud Rate Selection in Mode 1, 2, and 3, the S0 can also uses Timer 1, S0BRG and S1BGR as Baud Rate clock source.

The S0BRG is the embedded baud rate generator, which detailed function is described in Section 19.11.1. When S0BRG is used as the baud rate generator of S0, the baud rate is as follows.

The secondary UART (S1) in **MG82F6P32** has an independent baud-rate generator S1BRG. S0 can set URTS (S0CFG.7) to select the S1BRT as the timer source for UART Mode 1 and Mode 3. See Section "20.9 S1 Baud Rate Generator for S0" for the details on S0 baud rate select.

Timer 1:

$$\text{Mode 1, 3 Baud Rate} = \frac{2^{SMOD1} \times 2^{(SMOD2 \times 2)}}{32} \times \frac{F_{SYSCLK}}{12 \times (256 - TH1)} \quad ; T1X12=0$$

$$\text{or} = \frac{2^{SMOD1} \times 2^{(SMOD2 \times 2)}}{32} \times \frac{F_{SYSCLK}}{1 \times (256 - TH1)} \quad ; T1X12=1$$

SnBRG (n = 0, 1):

$$\text{Mode 1, 3 Baud Rate} = \frac{2^{SMOD1} \times 2^{(SMOD2 \times 2)}}{32} \times \frac{F_{SYSCLK}}{12 \times (256 - SnBRT)} \quad ; S0TX12=0$$

$$\text{or} = \frac{2^{SMOD1} \times 2^{(SMOD2 \times 2)}}{32} \times \frac{F_{SYSCLK}}{1 \times (256 - SnBRT)} \quad ; S0TX12=1$$

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 19–6 defines the Baud Rate setting with SMOD2 factor in Timer 1 and S1BGR baud rate generator. Please reference section 20.9 for S1BRG setting

Table 19–6. SMOD2 application criteria in Mode 1 & 3 using Timer 1 & SnBRG

SMOD2	SMOD1	Baud Rate	Note	Recommended Max. Receive Error (%)
0	0	Default Baud Rate	Standard function	± 3%
0	1	Double Baud Rate	Standard function	± 3%
1	0	Double Baud Rate **X2**	Enhanced function	± 2%
1	1	Reserved		

Table 19–7 ~ Table 19–14 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 F_{SYSCLK} = 48MHz).

Table 19–7. Timer 1 or SnBRG Generated Commonly Used Baud Rates @ F_{SYSCLK}=11.0592MHz

TH1 or SnBRT Reload Value								
SMOD1	0	1	0	Error	0	1	0	Error
SMOD2	0	0	1		0	0	1	
Baud Rate	**T1X12=0**				**T1X12=1**			
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	250	0.0%
460.8K	--	--	--	--	--	--	253	0.0%
691.2K	--	--	--	--	--	--	254	0.0%
1.3824M	--	--	--	--	--	--	255	0.0%

Table 19–8. Timer 1 or SnBRG Generated Commonly Used Baud Rates @ F_{SYSCLK}=22.1184MHz

TH1 or SnBRT Reload Value								
SMOD1	0	1	0	Error	0	1	0	Error
SMOD2	0	0	1		0	0	1	
Baud Rate	**T1X12=0**				**T1X12=1**			
1200	208	160	--	0.0%	--	--	--	--
2400	232	208	--	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	--	--	250	244	--	0.0%
230400	--	--	--	--	253	250	--	0.0%
460.8K	--	--	--	--	--	253	250	0.0%
691.2K	--	--	--	--	--	--	252	0.0%
921.6K	--	--	--	--	--	--	253	0.0%
1.3824M	--	--	--	--	--	--	254	0.0%
1.8432M	--	--	--	--	--	--	--	--
2.7648M	--	--	--	--	--	--	255	0.0%

Table 19–9. Timer 1 or SnBRG Generated Commonly Used Baud Rates @ F_{SYSCLK}=12.0MHz

TH1 or SnBRT Reload Value								
SMOD1	0	1	0	Error	0	1	0	Error
SMOD2	0	0	1		0	0	1	
Baud Rate	**T1X12=0**				**T1X12=1**			
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	--	--	--	--	--	--	243	0.16%

Table 19–10. Timer 1 or SnBRG Generated Commonly Used Baud Rates @ F_{SYSCLK}=24.0MHz

TH1 or SnBRT Reload Value								
SMOD1	0	1	0	Error	0	1	0	Error
SMOD2	0	0	1		0	0	1	
Baud Rate	**T1X12=0**				**T1X12=1**			
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%
230400	--	--	--	--	--	--	243	0.16%

Table 19–11. Timer 1 or SnBRG Generated Commonly Used Baud Rates @ F_{SYSCLK}=29.4912MHz

TH1 or SnBRT Reload Value								
SMOD1	0	1	0	Error	0	1	0	Error
SMOD2	0	0	1		0	0	1	
Baud Rate	**T1X12=0**				**T1X12=1**			
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%
230400	--	--	--	--	252	248	--	0.0%
460.8K	--	--	--	--	254	252	--	0.0%
921.6K	--	--	--	--	255	254	--	0.0%

1.8432M	--	--	--	--	--	255	254	0.0%

Table 19–12. Timer 1 or SnBRG Generated Commonly Used Baud Rates @ F_{SYSCLK}=44.2368MHz

TH1 or SnBRT Reload Value								
SMOD1	0	1	0	Error	0	1	0	Error
SMOD2	0	0	1		0	0	1	
Baud Rate	**T1X12=0**				**T1X12=1**			
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%
230400	--	255	--	0.0%	250	244	--	0.0%
460.8K	--	--	--	--	253	250	--	0.0%
691.2K	--	--	--	--	--	--	--	--
921.6K	--	--	--	--	--	253	--	0.0%
1.3824M	--	--	--	--	--	--	--	--
1.8432M	--	--	--	--	--	--	--	--
2.7648M	--	--	--	--	--	255	254--	0.0%

Table 19–13. Timer 1 or SnBRG Generated Commonly Used Baud Rates @ F_{SYSCLK}=32MHz

TH1 or SnBRT Reload Value								
SMOD1	0	1	0	Error	0	1	0	Error
SMOD2	0	0	1		0	0	1	
Baud Rate	**T1X12=0**				**T1X12=1**			
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	152	0.16%
57600	--	--	--	--	239	222	--	2.12%
115200	--	--	--	--	--	239	--	2.12%

Table 19–14. Timer 1 or SnBRG Generated Commonly Used Baud Rates @ F_{SYSCLK}=48.0MHz

TH1 or SnBRT Reload Value								
SMOD1	0	1	0	Error	0	1	0	Error
SMOD2	0	0	1		0	0	1	
Baud Rate	**T1X12=0**				**T1X12=1**			
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%
230400	--	--	--	--	--	243	230	0.16%
460.8K	--	--	--	--	--	--	243	0.16%

19.9.4.2. 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.

$$\text{Mode 1, 3 Baud Rate} = \frac{2^{\text{SMOD2 X (SMOD1 + 1)}} \times F_{SYSCLK}}{32 \times (65536 - \text{(RCAP2H, RCAP2L)})} \text{ ; T2X12=0}$$

$$\text{or} = \frac{2^{\text{SMOD2 X (SMOD1 + 1)}} \times F_{SYSCLK}}{16 \times (65536 - \text{(RCAP2H, RCAP2L)})} \text{ ; T2X12=1}$$

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 19–15 defines the Baud Rate setting with SMOD2 factor in Timer 2 baud rate generator.

Table 19–15. SMOD2 application criteria in Mode 1 & 3 using Timer 2

SMOD2	SMOD1	Baud Rate	Note	Recommended Max. Receive Error (%)
0	X	Default Baud Rate	Standard function	± 3%
1	0	Double Baud Rate	Enhanced function	± 2%
1	1	Reserved		

Table 19–16 ~ Table 19–23 list various commonly used baud rates and how they can be obtained from Timer 2 in its Baud-Rate Generator Mode.

Table 19–16. Timer 2 Generated Commonly Used Baud Rates @ F_{SYSCLK}=11.0592MHz

[RCAP2H, RCAP2L] Reload Value								
SMOD1	0	1	0	Error	0	1	0	Error
SMOD2	0	0	1		0	0	1	
Baud Rate	**T2X12=0**				**T2X12=1**			
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	0.0%	65533	65533	65530	0.0%
460.8K	--	--	--	--	--	--	65533	0.0%
691.2K	--	--	65535	0.0%	--	--	65534	0.0%
921.6K	--	--	--	--	--	--	--	--
1.3824M	--	--	--	--	--	--	65535	0.0%
1.8432M	--	--	--	--	--	--	--	--
2.7648M	--	--	--	--	--	--	--	--

Table 19–17. Timer 2 Generated Commonly Used Baud Rates @ F_{SYSCLK}=22.1184MHz

[RCAP2H, RCAP2L] Reload Value								
SMOD1	0	1	0	Error	0	1	0	Error
SMOD2	0	0	1		0	0	1	
Baud Rate	**T2X12=0**				**T2X12=1**			
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%
460.8K	--	--	65533	0.0%	65533	65533	65530	0.0%
691.2K	--	--	65534	0.0%	--	--	65532	0.0%
921.6K	--	--	--	--	--	--	65533	0.0%
1.3824M	--	--	65535	0.0%	--	--	65534	0.0%
1.8432M	--	--	--	--	--	--	--	--
2.7648M	--	--	--	--	--	--	65535	0.0%

Table 19–18. Timer 2 Generated Commonly Used Baud Rates @ F_{SYSCLK}=12.0MHz

[RCAP2H, RCAP2L] Reload Value								
SMOD1	0	1	0	Error	0	1	0	Error
SMOD2	0	0	1		0	0	1	
Baud Rate	**T2X12=0**				**T2X12=1**			
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	--	--	--	--	--	--	65523	0.16%

Table 19–19. Timer 2 Generated Commonly Used Baud Rates @ F_{SYSCLK}=24.0MHz

[RCAP2H, RCAP2L] Reload Value								
SMOD1	0	1	0	Error	0	1	0	Error
SMOD2	0	0	1		0	0	1	
Baud Rate	**T2X12=0**				**T2X12=1**			
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%
230400	--	--	--	--	--	--	65523	0.16%

Table 19–20. Timer 2 Generated Commonly Used Baud Rates @ F_{SYSCLK}=29.4912MH z

[RCAP2H, RCAP2L] Reload Value								
SMOD1	0	1	0	Error	0	1	0	Error
SMOD2	0	0	1		0	0	1	
Baud Rate	**T2X12=0**				**T2X12=1**			
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%
230400	65532	65532	--	0.0%	65528	65528	--	0.0%
460.8K	65534	65534	--	0.0%	65532	65532	--	0.0%
921.6K	65535	65535	65534	0.0%	65534	65534	65532	0.0%
1.8432M	--	--	--	--	--	--	--	--

Table 19–21. Timer 2 Generated Commonly Used Baud Rates @ F_{SYSCLK}=44.2368MHz

[RCAP2H, RCAP2L] Reload Value								
SMOD1	0	1	0	Error	0	1	0	Error
SMOD2	0	0	1		0	0	1	
Baud Rate	**T2X12=0**				**T2X12=1**			
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%
230400	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	--	--	65534	65534	--	0.0%
1.8432M	--	--	--	--	--	--	--	--
2.7648M	--	--	--	--	65535	65535	65534	0.0%

5.5296M	--	--	--	--	--	--	65535	0.0%

Table 19–22. Timer 2 Generated Commonly Used Baud Rates @ F_{SYSCLK}=32MHz

[RCAP2H, RCAP2L] Reload Value								
SMOD1	0	1	0	Error	0	1	0	Error
SMOD2	0	0	1		0	0	1	
Baud Rate	**T2X12=0**				**T2X12=1**			
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%
115200	--	--	--	--	--	--	65501	0.64%

Table 19–23. Timer 2 Generated Commonly Used Baud Rates @ F_{SYSCLK}=48.0MHz

[RCAP2H, RCAP2L] Reload Value								
SMOD1	0	1	0	Error	0	1	0	Error
SMOD2	0	0	1		0	0	1	
Baud Rate	**T2X12=0**				**T2X12=1**			
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%
230400	--	--	65523	0.16%	65523	65523	65510	0.16%
460.8K	--	--	--	--	--	--	65523	0.16%

19.9.4.3. Using Split Timer 2 as the Baud Rate Generator

When Timer 2 is in Split mode and used as the baud rate generator (either TCLK or RCLK in T2CON is '1'), the baud rate is as follows.

$$\text{Mode 1, 3 Baud Rate} = \frac{2^{\text{SMOD2 X (SMOD1 + 1)}} \times F_{SYSCLK}}{16 \times 12 \times (256 - \text{RCAP2L})} \text{ ; TL2X12=0}$$

$$\text{or} = \frac{2^{\text{SMOD2 X SMOD1 +1)}} \times F_{SYSCLK}}{16 \times 1 \times (256 - \text{RCAP2L})} \text{ ; TL2X12=1}$$

Note:

Table 19–24 defines the Baud Rate setting with SMOD2 and SMOD1 factors in Split Timer 2 baud rate generator.

Table 19–24. SMOD2 application criteria in Mode 1 & 3 using Split Timer 2

SMOD2	SMOD1	Baud Rate	Note	Recommended Max. Receive Error (%)
0	X	Default Baud Rate	Standard function	± 3%
1	0	Double Baud Rate	Enhanced function	± 2%
1	1	Reserved		

19.10. ARGB Transmitter

- ARGB data output on TXD0 when S0 is set in enhanced mode.
- Programmable bit rate by S0BRG (S0 Baud Rate Generator).
- Selectable duty cycle on 1/4 or 1/3
- Selectable bus reset time detection on 60us, 260us, 200us and 320us.

Initial S0 to ARGB operating mode

- Set { SM30, SM00, SM10, S0RCK, S0TCK} to ARGB mode.
- Write S0BRC/S0BRT SFR & S0TX12/SMOD1 control bits to configure the Baud Rate on ARGB.
- Define S0DTY1 to configure ARGB duty cycle on 1/4 or 1/3.
- Select the bus reset time detection duration on S0IDT11 and S0IDT01.

Flow for Transmit a PD data frame

- Wait RI0(S0AGBRF) for bus reset event.
- Clear RI0, then start to write ARGB data on S0BUF.
- Wait TI0(S0AGBF) for S0 TX buffer empty.
- Clear TI0, then write next byte ARGB data on S0BUF…..and so on, until all ARGB data has been transmitted.
- Wait RI0 to start a new frame to transmit ARGB data.

19.10.1. S0 ARGB Baud Rate

Figure 19–14. S0 ARGB Mode Baud Rate equation: (n=0)

$$\text{Sn ARGB Baud Rate} = \frac{1}{m} \times \frac{F_{SYSCLK}}{12 \times (256 - \text{SnBRT})} \text{ ; SnTX12=0} \quad \begin{array}{l}\text{; m=4, if SnDTY1 = 0} \\ \text{; m=3, if SnDTY1 = 1}\end{array}$$

$$\text{or} = \frac{1}{m} \times \frac{F_{SYSCLK}}{1 \times (256 - \text{SnBRT})} \text{ ; SnTX12=1} \quad \begin{array}{l}\text{; m=4, if SnDTY1 = 0} \\ \text{; m=3, if SnDTY1 = 1}\end{array}$$

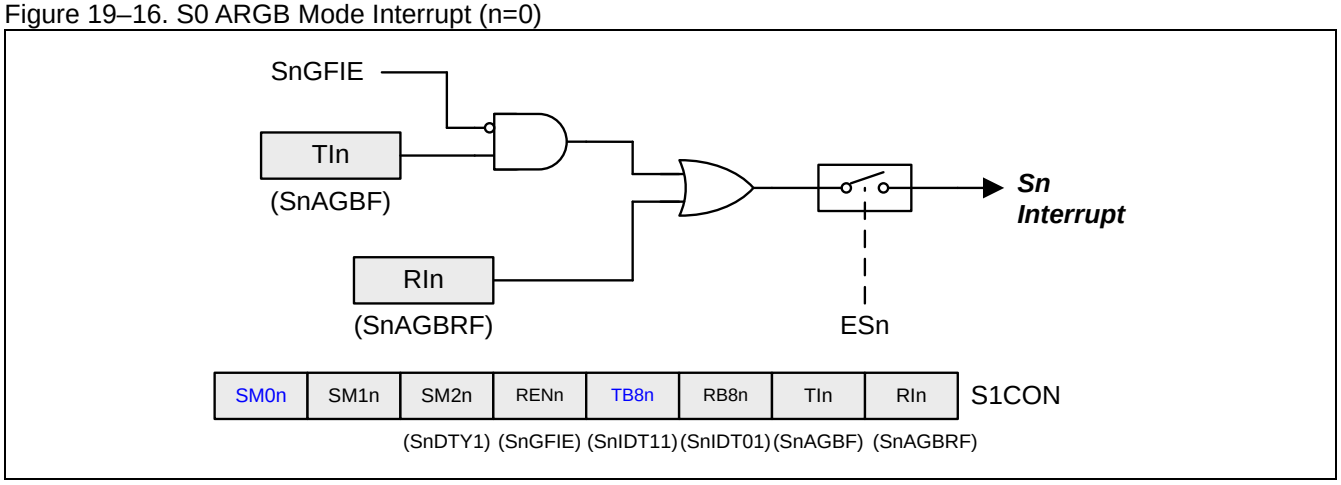
19.10.2. S0 ARGB Bus Reset Time

Figure 19–15. S0 ARGB Mode Bus Reset Time (n=0)

$$\text{Sn ARGB Reset Time} = \frac{8 \times m}{\text{Sn ARGB Baud Rate}}$$; m=6/20/26/32, indexed by SnIDT1n~0n

19.10.3. S0 ARGB Interrupt

Figure 19–16. S0 ARGB Mode Interrupt (n=0)



19.10.4. S0 ARGB Register

The following SPI special function registers are re-defined to the ARGB operation:

S0CON/ S0AGCR: Serial port 0 ARGB Control Register

SFR Page = **0**

SFR Address = 0x98 RESET = 0000-0000

7	6	5	4	3	2	1	0
SM00/FE	SM10	SM20	REN0	TB80	RB80	TI0	RI0
SM00	SM10	S0DTY1	S0GFIE	S0IDT11	S0IDT01	S0AGBF	S0AGBRF
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 5: S0DTY1, ARGB signal Duty selection.
0: Select ARGB signal Duty to 3/4(H) or 1/4(L).
1: Select ARGB signal Duty to 2/3(H) or 1/3(L).

Bit 4: S0GFIE, TI0 interrupt Ignored.
0: Enabled TI0 interrupt. Default is enabled.
1: Disable TI0 interrupt.

Bit 3~2: S0IDT1~0, Selection for ARGB bus reset time duration.

S0IDT11~01	ARGB Bus Reset Time (when Baud Rate 800K)
0 0	**260us**
0 1	200us
1 0	**60us**
1 1	320us

Bit 1: S0AGBF, the flag reports Transmit Holding Register finishes the transfer to Output Shift Register in ARGB mode.
0: **The S0AGBF is cleared in software by writing "0" to this bit.**
1: When S0BUF empty, the S0AGBF bit is set and an interrupt is generated if S0 interrupt is enabled.

Bit 0: S0AGBRF, ARGB bus Reset flag in ARGB mode.
0: The S0AGBRF flag is cleared in software by writing "0" to this bit.
1: This bit is set to logic 1 when bus meets the ARGB reset time. The reset time is defined by S0IDT11~01.

19.11. Serial Port 0 Enhanced 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 RESET = 0000-0000

7	6	5	4	3	2	1	0
S0TR	S0TX12	S0TCK	S0RCK	S0CKOE	ARTE	--	--
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**W**	**W**

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.
To use with SM20 to select the clock source.

S0TX12, SM20	S0BRG Clock Selection
0 0	SYSCLK/12
0 1	T1OF
1 0	SYSCLK
1 1	T0OF

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 RESET = 0000-0000

7	6	5	4	3	2	1	0
S0BRT.7	S0BRT.6	S0BRT.5	S0BRT.4	S0BRT.3	S0BRT.2	S0BRT.1	S0BRT.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~0: It is used as the reload value register for baud rate timer generator that works in a similar as Timer 1.

S0BRC: Serial port 0 Baud Rate Counter Register

SFR Page = **0 only**

SFR Address = 0x9B RESET = 0000-0000

7	6	5	4	3	2	1	0
S0BRC.7	S0BRC.6	S0BRC.5	S0BRC.4	S0BRC.3	S0BRC.2	S0BRC.1	S0BRC.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~0: It is used as the reload value register for baud rate timer generator that works in a similar 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.

19.11.1. S0 Baud Rate Generator (S0BRG)

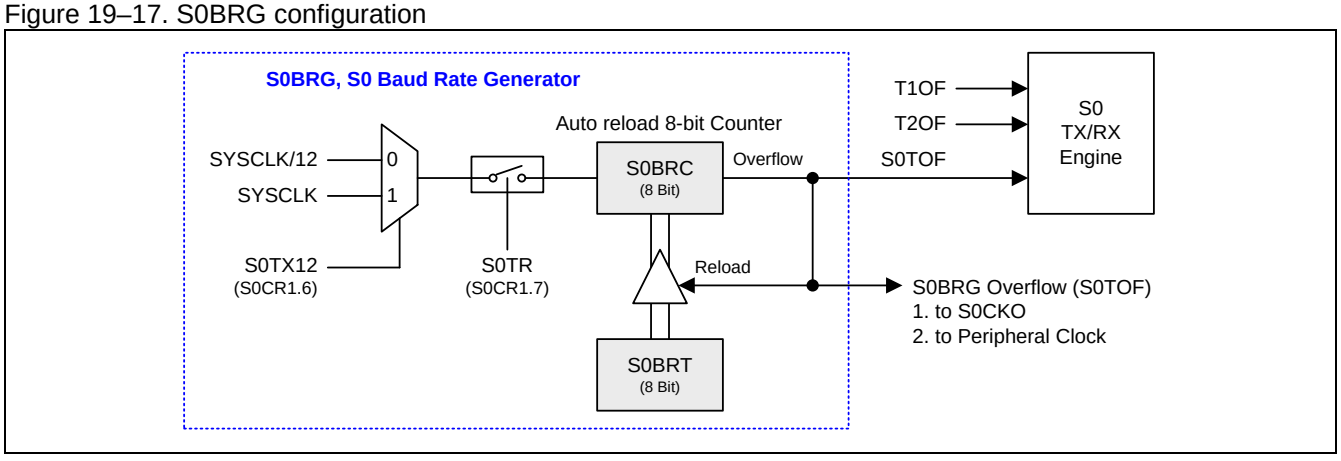
To give S0 more flexibility, S0 Baud Rate Generator S0BRG can be selected as Baud Rate source, the **MG82F6P32** has an embedded Baud Rate Generator to generate the clock for serial port 0 operation. It is constructed by an 8-bit up-counter, S0BRC, and an 8-bit reload register, S0BRT. The overflow (S0TOF) 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. The configuration of the Serial Port 0 baud rate selection please reference Figure 19–13. S0 Baud Rate Selection.

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 19–17.

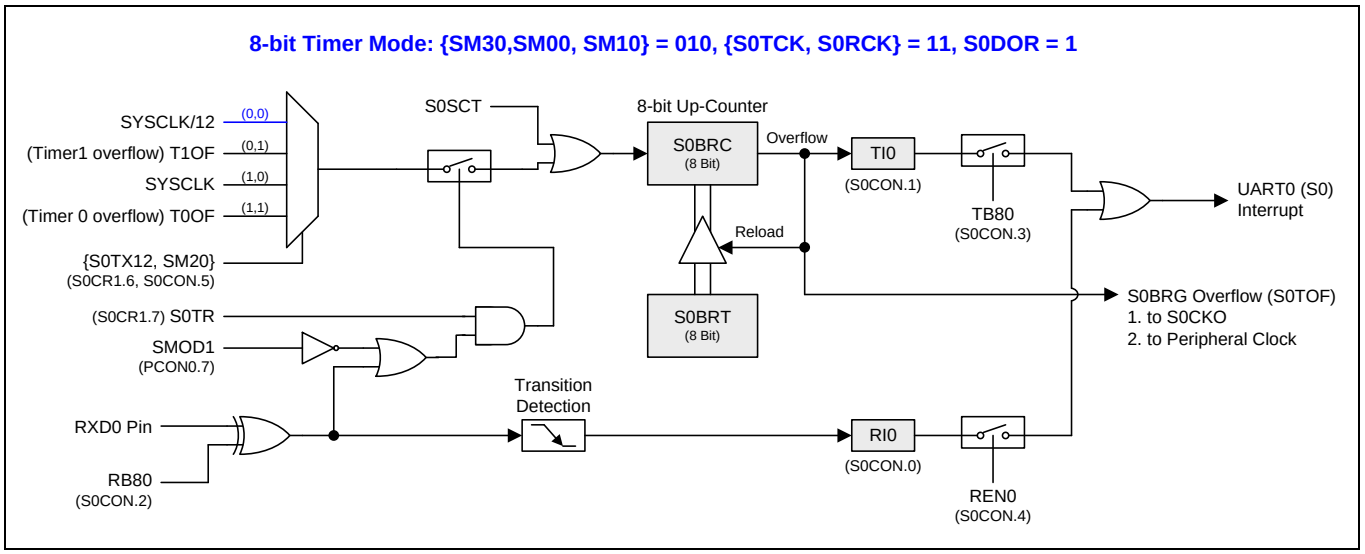
Figure 19–17. S0BRG configuration



19.11.2. S0 acts as 8-bit Timer Mode

S0 8-bit Timer Mode is shown in Figure 19–18.

Figure 19–18. S0 8-bit Timer Mode



19.11.3. S0BRG Programmable Clock Output

S0BRG has a clock output mode is shown in Figure 19–19 and Figure 19–20.

Figure 19–19. S0BRG Clock Output (S0BRG in 8-bit Timer Mode)

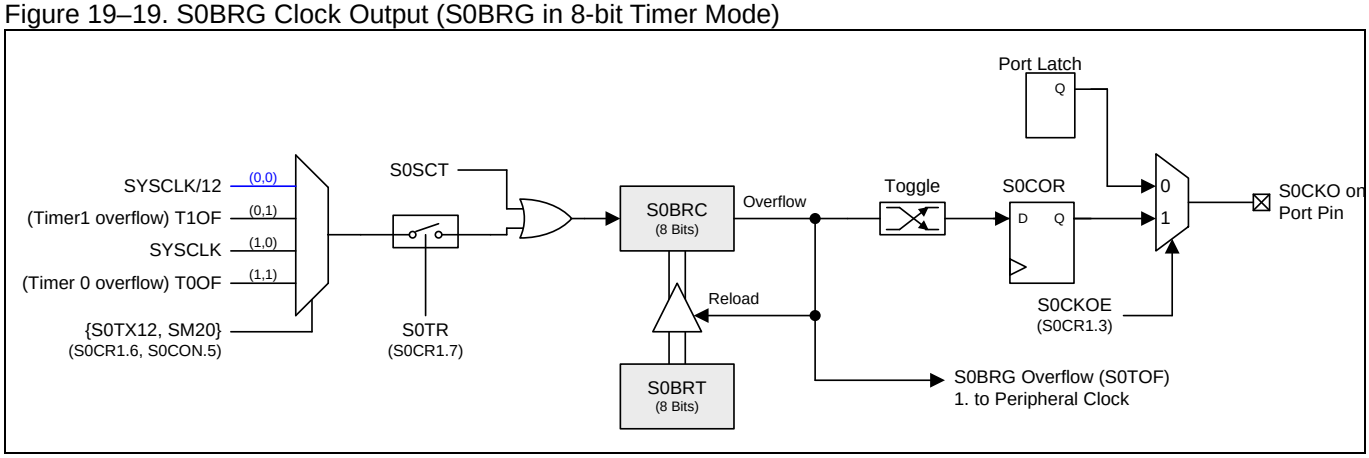
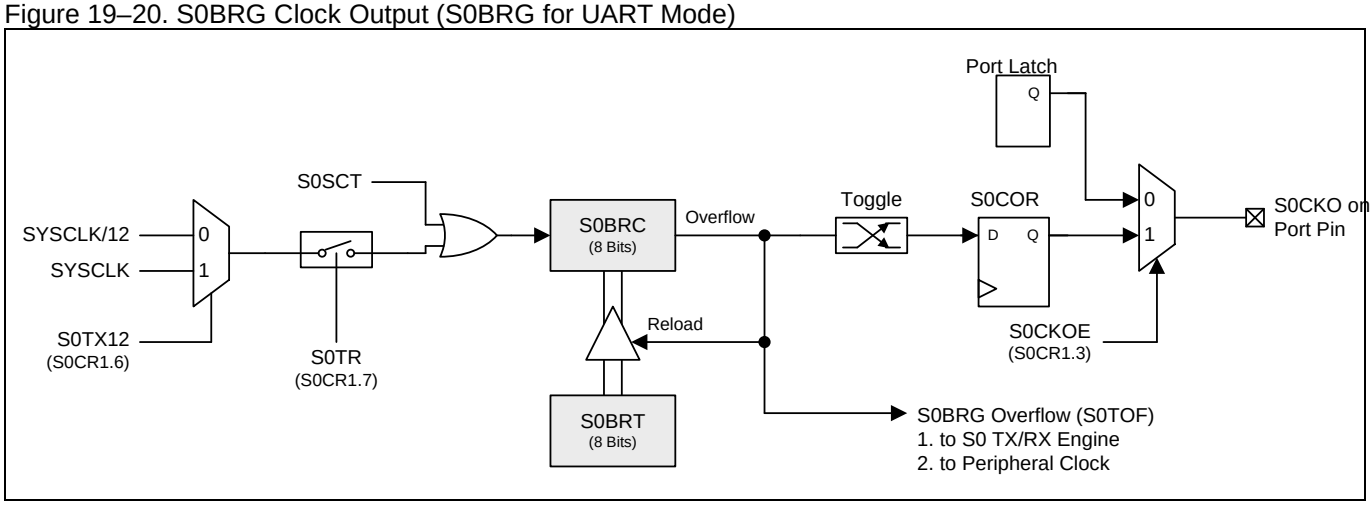


Figure 19–20. S0BRG Clock Output (S0BRG for UART Mode)



AUXR6: Auxiliary Register 6

SFR Page = **3 only**
SFR Address = 0xA4 RESET = 0000-0000

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

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

S0COPS	S0CKO
0	P4.7
1	P4.4

19.12. 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 RESET = 0000-0000

7	6	5	4	3	2	1	0
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

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	Baud Rate
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, /16, /8
0	1	1	3	9-bit UART	variable
1	0	0	4	**SPI Master**	SYSCLK/12 or 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.
In Mode0 and Mode4:
The Tx initial signal level can be defined by SM20
0: To define the initial signal level of Tx is high.
1: To define the initial signal level of Tx is low.

In Modes 2 and 3:
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 pure timer mode:
To use with S0TX12 to select the clock source.

S0TX12, SM20	S0BRG Clock Selection
0 0	SYSCLK/12
0 1	T1OF
1 0	SYSCLK
1 1	T0OF

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.
In mode 0 and mode 4, it is the selection on RX data sample edge.
0: Sample the RX data on trailing edge of the serial clock (TXD0).
1: Sample the RX data on leading edge of the serial clock (TXD0).

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 RESET = XXXX-XXXX

7	6	5	4	3	2	1	0
S0BUF.7	S0BUF.6	S0BUF.5	S0BUF.4	S0BUF.3	S0BUF.2	S0BUF.1	S0BUF.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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 RESET = 0000-0000

7	6	5	4	3	2	1	0
SADDR.7	SADDR.6	SADDR.5	SADDR.4	SADDR.3	SADDR.2	SADDR.1	SADDR.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

SADEN: Slave Address Mask Register (SMOD3 = 0)

SFR Page = 0~F
SFR Address = 0xB9 RESET = 0000-0000

7	6	5	4	3	2	1	0
SADEN.7	SADEN.6	SADEN.5	SADEN.4	SADEN.3	SADEN.2	SADEN.1	SADEN.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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 POR = 0001-0000
SFR Address = 0x87 RESET = 0000-0000

7	6	5	4	3	2	1	0
SMOD1	SMOD0	GF	POF	GF1	GF0	PD	IDL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

7	6	5	4	3	2	1	0
URTS	SMOD2	0	SM30	S0DOR	BTI	UTIE	SMOD3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 "19.9.4 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: Reserved. Software must write "**0**" on this bit when S0CFG is written.

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 RESET = 0000-0000

7	6	5	4	3	2	1	0
STAF	STOF	C1PLK	C0PLK	T1X12	T0X12	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 RESET = 0000-0000

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

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

S0PS1~0	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

AUXR6: Auxiliary Register 6

SFR Page = **3 only**

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
KBIHPS1	KBIHPS0	KBILPS0	KBILPS0	T3FCS	T2FCS	SnMIPS	S0COPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 1: SnMIPS, S0MI, S1MI, S2MI & S3MI Port pin Selection.

SnMIPS	S0MI	S1MI
0	P1.6	P1.0
1	P3.3	P3.5

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

S0COPS	S0CKO
0	P4.7
1	P4.4

20. Serial Port 1 (UART1)

The **MG82F6P32** 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.

The UART1 and UART0 in **MG82F6P32** can operate simultaneously in identical or different modes and communication speeds.

20.1. Serial Port 1 Mode Selection

The serial port 1 can operate in **4** standard modes and **3/4** enhance 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 6 provides the BMC ENDEC (encoder/decoder) on transmit and receive. Mode 8 is a pure timer function. Mode 9 supports the function for ARGB transmitter.

Table 20–1. Serial Port 1 Mode Selection

S1TME	SM31	SM01, SM11	MODE	Function	Baud Rate Time Base	Note
0	0	0 0	0	shift register	SYSCLK/12 or SYSCLK/4 (S1MOD1=1)	
0	0	0 1	1	8-bit UART	S1BRG overflow	
0	0	1 0	2	9-bit UART	SYSCLK/64, or /32	
0	0	1 1	3	9-bit UART	S1BRG overflow	
0	1	0 0	4	SPI Master	SYSCLK/12 or SYSCLK/4 (S1MOD1=1)	
0	1	0 1	5	Reserved		
0	1	1 0	6	BMC ENDEC	S1BRG overflow	
0	1	1 1	7	Reserved		
1	0	0 0	8	Pure Timer	8-bit timer function	
1	0	0 1	9	ARGB Transmitter	S1BRG overflow	
Others				Reserved		

20.2. Serial Port 1 Baud Rate Generator (S1BRG)

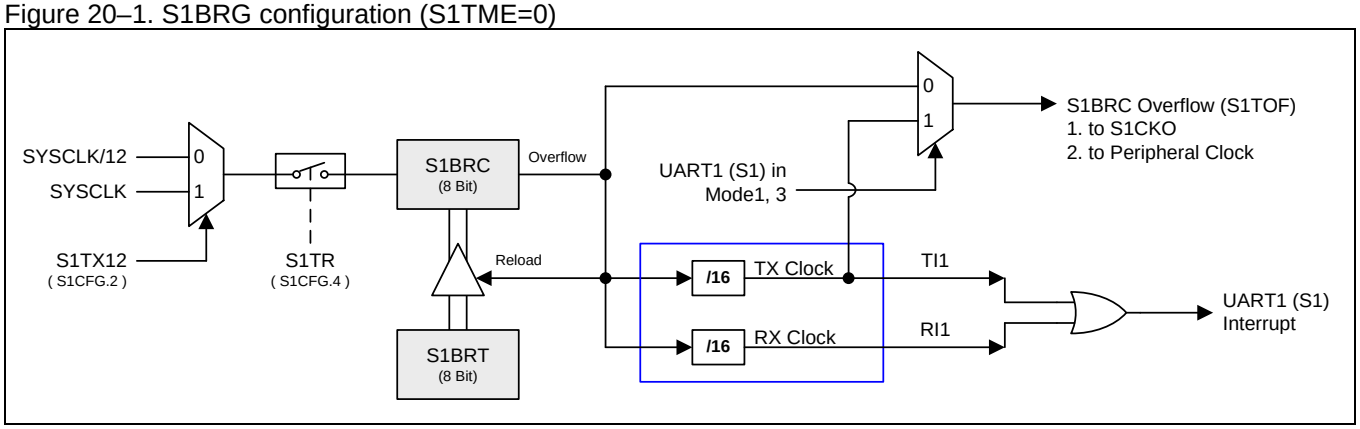
The **MG82F6P32** has an embedded Baud Rate Generator to generate the UART clock for serial port 1 operation in mode 1 and mode 3. It is constructed by an 8-bit up-counter, S1BRC, and an 8-bit reload register, S1BRT. The overflow (S1TOF) of S1BRC is the time base of UART1 serial engine in mode 1 and 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 20–1.

Figure 20–1. S1BRG configuration (S1TME=0)



20.3. Serial Port 1 Baud Rate Setting

20.3.1. Baud Rate (Serial clock frequency) in Mode 0 & 4

S1 Mode 0 & 4 Baud Rate equation: (n=1)

$$\text{Baud Rate} = \frac{F_{SYSCLK}}{12} \quad ; \text{SnMOD1=0}$$

$$\text{or} = \frac{F_{SYSCLK}}{4} \quad ; \text{SnMOD1=1}$$

20.3.2. Baud Rate in Mode 2

S1 Mode 2 Baud Rate equation: (n=1)

$$\text{Sn Mode 2 Baud Rate} = \frac{2^{SnMOD1}}{64} \times F_{SYSCLK}$$

Table 20–2. S1 Mode 2 Baud Rates @ F_{SYSCLK}=11.0592MHz

Baud Rate	S1MOD1	Error
172800	0	0.0%
345600	1	0.0%

Table 20–3. S1 Mode 2 Baud Rates @ F_{SYSCLK}=12.00MHz

Baud Rate	S1MOD1	Error
187500	0	0.0%
375000	1	0.0%

20.3.3. Baud Rate in Mode 1 & 3

S1 Mode 1, 3 Baud Rate equation: (n=1)

$$\text{Sn Mode 1, 3 Baud Rate} = \frac{2^{SnMOD1}}{32} \times \frac{F_{SYSCLK}}{12 \times (256 - SnBRT)} \quad ; \text{SnTX12=0}$$

$$\text{or} = \frac{2^{SnMOD1}}{32} \times \frac{F_{SYSCLK}}{1 \times (256 - SnBRT)} \quad ; \text{SnTX12=1}$$

S1MOD1	Baud Rate	Note	Recommended Max. Receive Error (%)
0	Default Baud Rate	Standard function	± 3%
1	Double Baud Rate	Standard function	± 3%

Table 20–4 ~ Table 20–11 list various commonly used baud rates and how they can be obtained from S1BRG, serial port 1 baud rate generator.

Table 20–4. S1BRG Generated Commonly Used Baud Rates @ F_{SYSCLK}=11.0592MHz

S1BRT Reload Value						
S1MOD1	0	1	Error	0	1	Error
Baud Rate	**S1TX12=0**			**S1TX12=1**		
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 20–5. S1BRG Generated Commonly Used Baud Rates @ F_{SYSCLK}=22.1184MHz

S1BRT Reload Value						
S1MOD1	0	1	Error	0	1	Error
Baud Rate	**S1TX12=0**			**S1TX12=1**		
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%
460.8K	--	--	--	--	253	0.0%

Table 20–6. S1BRG Generated Commonly Used Baud Rates @ F_{SYSCLK}=12.0MHz

S1BRT Reload Value						
S1MOD1	0	1	Error	0	1	Error
Baud Rate	**S1TX12=0**			**S1TX12=1**		
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%

Table 20–7. S1BRG Generated Commonly Used Baud Rates @ F_{SYSCLK}=24.0MHz

S1BRT Reload Value						
S1MOD1	0	1	Error	0	1	Error
Baud Rate	**S1TX12=0**			**S1TX12=1**		
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 20–8. S1BRG Generated Commonly Used Baud Rates @ F_{SYSCLK}=29.4912MHz

S1BRT Reload Value						
S1MOD1	0	1	Error	0	1	Error
Baud Rate	**S1TX12=0**			**S1TX12=1**		
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%
230400	--	--	--	252	248	0.0%
460.8K	--	--	--	254	252	0.0%
921.6K	--	--	--	255	254	0.0%
1.8432M	--	--	--	--	255	0.0%

Table 20–9. S1BRG Generated Commonly Used Baud Rates @ FSYSCLK=44.2368MHz

S1BRT Reload Value						
S1MOD1	0	1	Error	0	1	Error
Baud Rate	**S1TX12=0**			**S1TX12=1**		
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%
230400	--	255	0.0%	250	244	0.0%
460.8K	--	--	--	253	250	0.0%

921.6K	--	--	--	--	253	0.0%

Table 20–10. S1BRG Generated Commonly Used Baud Rates @ F_{SYSCLK}=32.0MHz

S1BRT Reload Value						
S1MOD1	0	1	Error	0	1	Error
Baud Rate	**S1TX12=0**			**S1TX12=1**		
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 20–11. S1BRG Generated Commonly Used Baud Rates @ F_{SYSCLK}=48.0MHz

S1BRT Reload Value						
S1MOD1	0	1	Error	0	1	Error
Baud Rate	**S1TX12=0**			**S1TX12=1**		
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%
230400	--	--	--	--	243	0.16%

20.4. Serial Port 1 Mode 4 (SPI Master)

The Serial Port 1 of **MG82F6P32** is embedded Mode 4 to support SPI master engine. The Mode 4 is selected by SM31, SM01 and SM11. Table 20–12 shows the serial port mode definition in **MG82F6P32**.

Table 20–12. Serial Port 1 Mode Selection

SM31	SM01	SM11	Mode	Description	Baud Rate
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, or /32
0	1	1	3	9-bit UART	variable
1	0	0	4	SPI Master	SYSCLK/12 or SYSCLK/4
1	0	1	5	Reserved	variable
1	1	0	6	BMC endec	Variable
1	1	1	7	Reserved	variable

S1MOD1 also controls the SPI transfer speed. If **S1MOD1** = 1, the SPI clock frequency is SYSCLK/4. Otherwise, the SPI clock frequency is SYSCLK/12. When choose S1BRG as clock source the SPI clock frequency is not controlled by S1MOD1, it is equal S1TOF/4.

The SPI master in **MG82F6P32** uses the TXD1 as SPICLK, RXD1 as MOSI, and S1MI as MISO. nSS is selected by MCU software on other port pin. Figure 20–2 shows the SPI connection. It also can support the configuration for multiple slaves' communication in Figure 20–3.

Figure 20–2. Serial Port 1 Mode 4, Single Master and Single Slave configuration (n = 1)

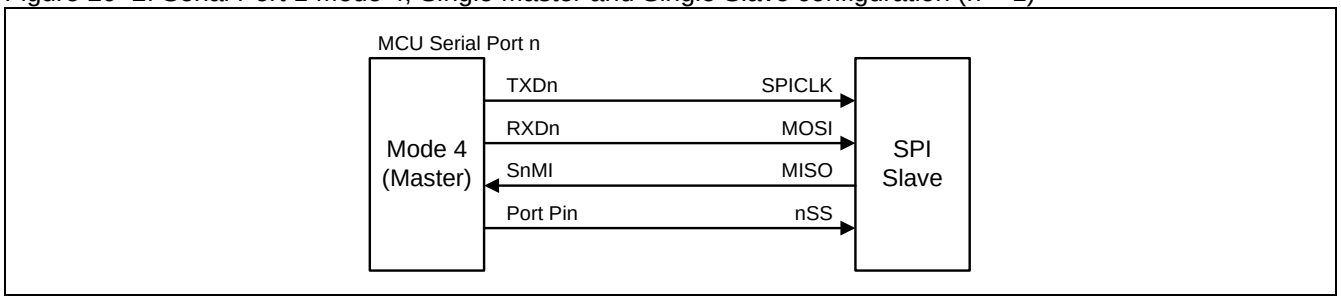
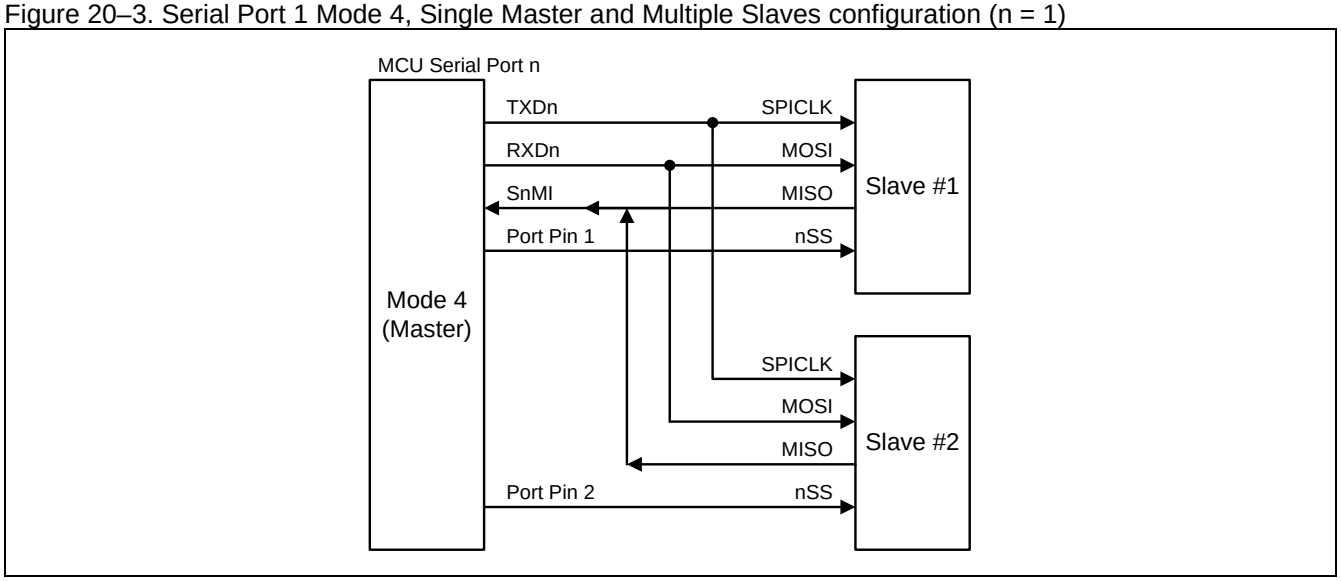


Figure 20–3. Serial Port 1 Mode 4, Single Master and Multiple Slaves configuration (n = 1)



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, **MG82F6P32** uses an easy way by initialize SPI clock polarity to fit them. Table 20–13 shows the serial port Mode 4 mapping with the four SPI operating mode.

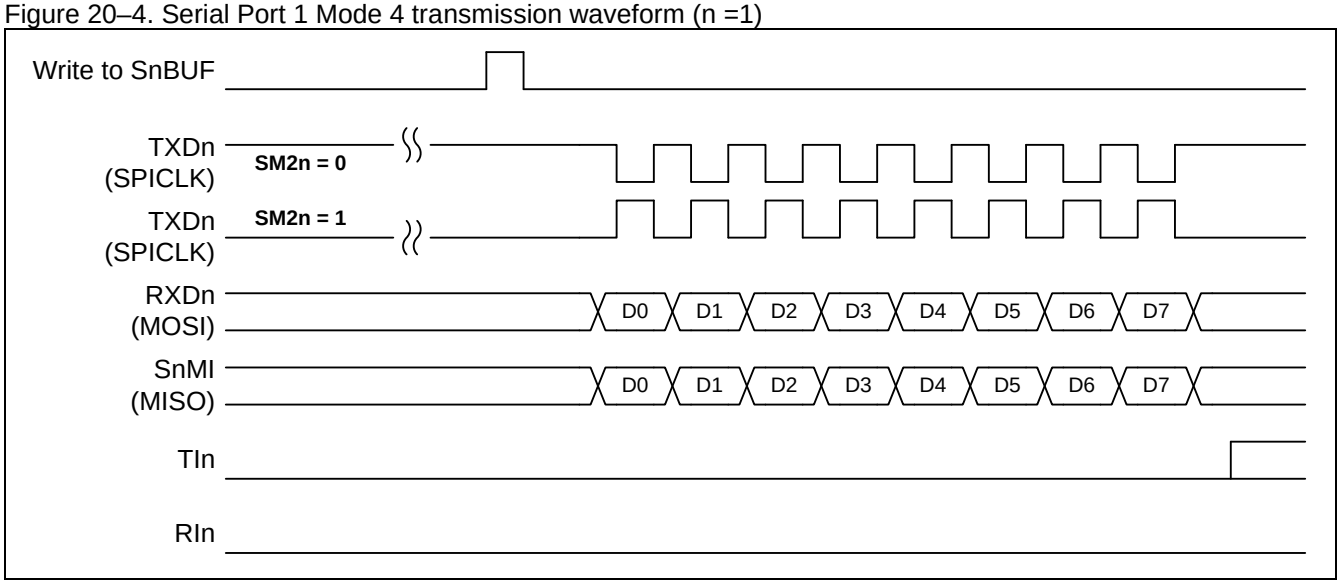
Table 20–13. SPI mode mapping with Serial Port Mode 4 configuration

SPI Mode	CPOL	CPHA	Configure SM2n & TB8n in SnCON
0	0	0	SM2n = 1, TB8n = 1
1	0	1	SM2n = 1, TB8n = 0
2	1	0	SM2n = 0, TB8n = 1
3	1	1	SM2n = 0, TB8n = 0

For bit order control (DORD) on SPI serial transfer, **MG82F6P32** 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. The initial status of the TxD can be defined by SM2n. When SM2n =0, the initial level of TxD is high. And if SM2n = 1, it will be low. 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 20–4 shows the transmission waveform in Mode 0. RI1 will not be asserted in Mode 4.

Figure 20–4. Serial Port 1 Mode 4 transmission waveform (n =1)



20.5. Timer Mode on S1BRG (Mode 8)

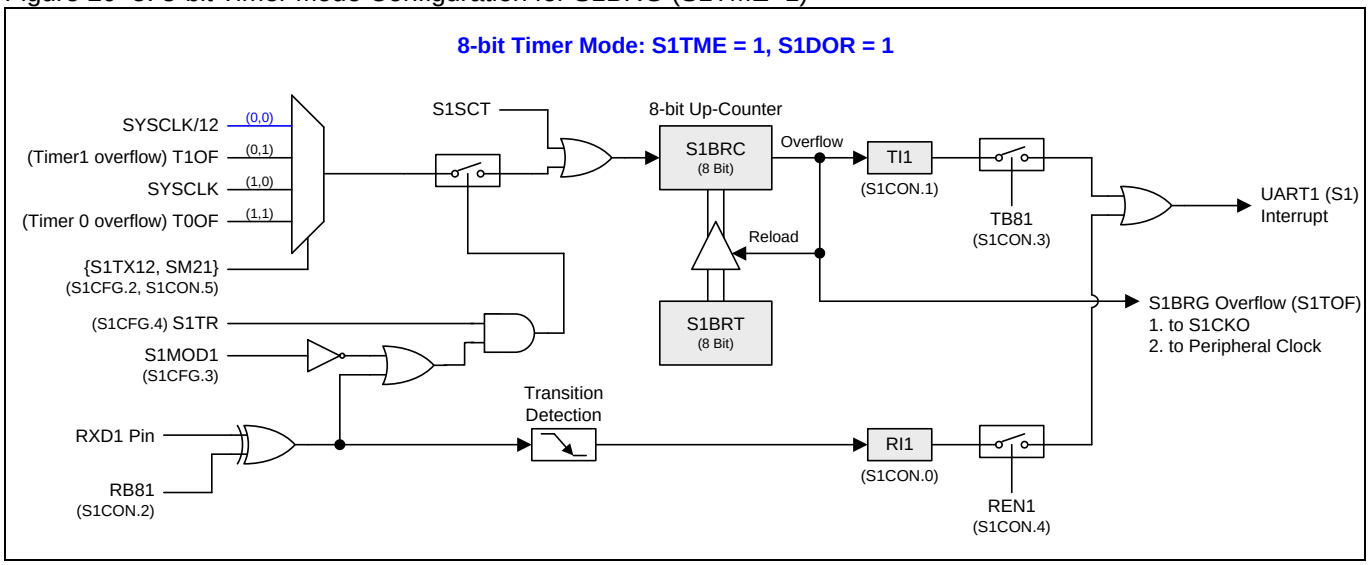
20.5.1. 8-bit Timer Mode

If the UART1 is not necessary in application or pending by software, setting S1TME=1 in the **MG82F6P32** 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 8-bit Timer mode on S1BRG is shown in Figure 20–5.

Figure 20–5. 8-bit Timer Mode Configuration for S1BRG (S1TME=1)



20.6. BMC ENDEC (Mode 6)

- Support 8-bit BMC data ENDEC (encoder/decoder), encoder output on TXD1, decoder from RXD1 input
- Programmable bit rate by S1BRG (S1 Baud Rate Generator).
- Special 5-bit BMC mode to fit USB PD format
- Configurable PDTXOE to enable off-chip driver to drive TXD1 data

Initial S1 to Mode 6 for USB PD operating mode

- Set {S1TME, SM31, SM01, SM11} to operating mode 6, BMC ENDEC.
- Write S1BRC/S1BRT SFR & S1TX12/S1MOD1 control bits to configure the S1 Baud Rate
- Set M6PDE1to select USB PD for 5-bit BMC ENDEC.
- Set REN and S1TR to start the S1 TX/RX engine.

Flow for Transmit a PD data frame

- Write 0xAA (preamble) in S1BUF
- Wait TI1, then write 0xAA in S1BUF again……. Total transmit 8 times 0xAA in S1BUF.
- Wait TI1, then send 5-bit K-code…….. Repeat 4 times for K-code sending.
- If need, wait TI1 to send the 5-bit Data0~DataF (include CRC32 value)
- Wait TI1, then send 5-bit EOP K-code.
- Wait TI1, then write 0xFF in S1BUF to end the transmission.

Flow for Receive a PD data frame

- Wait RI to get 1st data in S1BUF. If data bit 7~5 = 100, its 5 bit data is non Data0~DataF of 4b5b. If bit 7~5 = 000, its 5 bit data is Data0~DataF of 4b5b (include CRC32 value)
- Wait RI to get 2nd data in S1BUF and perform the same check on data bit 7~5.
- ……
- Wait RI, if the data is 0xFF, it indicates the end of the receive process.

20.6.1. Baud Rate in Mode 6

S1 Mode 6 Baud Rate equation: (n=1)

$$\text{Sn Mode 6 Baud Rate} = \frac{2^{SnMOD1}}{32} \times \frac{F_{SYSCLK}}{12 \times (256 - SnBRT)} \text{ ; SnTX12=0}$$

$$\text{or} = \frac{2^{SnMOD1}}{32} \times \frac{F_{SYSCLK}}{1 \times (256 - SnBRT)} \text{ ; SnTX12=1}$$

Note:
(1) For SYSCLK=24MHz & S1TX12=1 & S1MOD1=1, S1BRT=0xFB will generate 300K baud Rate for Mode 6.

S1CFG6: Serial Port 1 Configuration Register 6
SFR Page = **1 and 2**
SFR Address = **0x9D & S1 in Mode 6** RESET = 0001-0000

7	6	5	4	3	2	1	0
PDOEC1	M6TGU1	M6TXB1	M6IDL1	M6TUR1	M6ROR1	PDOE1	M6PDE1
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Before access S1CFG6 on SFR address 0x9D with SFR page 1 or 2, software must set the S1 module operating in mode 6.

Bit 7: PDOEC, PDOE output control.
0: Select active low on PDTXOE output.
1: Select active high on PDTXOE1 output.
Bit 6: M6TGU1, TX Give-Up Flag in S1 Mode 6.
0: Must be cleared by software.

1: Set by hardware to indicate the data transmission is give up because the engine is going on BMC data receive.

Bit 5: M6TXB1, TX (BMC data transmission) Busy status in S1 Mode 6. Read Only.
0: TX engine is not in busy.
1: TX engine is busy on data transmitting in S1 Mode 6.

Bit 4: M6IDL1. The BMC bus is IDLE on receive. Read Only.
0: The engine is going on receive RXD1 in S1 Mode 6.
1: Bus is idle on RXD1.

Bit 3: M6TUR1, TX Under-Run flag.
0: Must be cleared by software.
1: Set by hardware to indicate the last bit data is sent and the S1 TX buffer is empty in S1 mode 6.

Bit 2: M6ROR1, RX Over-Run flag.
0: Must be cleared by software.
1: Set by hardware to indicate the data received is overrun in S1 mode 6.

Bit 1: PDOE1, PDTXOE output enable that controls the external buffer to drive TXD1 on USB PD bus.
0: Disable PDTXOE function on port pin.
1: Enable PDTXOE function on port pin.

Bit 0: M6PDE1, USB PD BMC function enable control in S1 Mode 6.
0: Disable USB PD BMC function in S1 Mode 6. (8-bit BMC ENDEC)
1: Enable USB PD BMC function in S1 Mode 6. (5-bit BMC ENDEC)

S1CFG: Serial Port 1 Configuration Register

SFR Page = **1 and 2**
SFR Address = **0x9C** RESET = 0010-0000

7	6	5	4	3	2	1	0
SM31	S1MOD3	S1DOR	S1TR	S1MOD1	S1TX12	S1CKOE	S1TME
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 6: S1MOD3, S1MOD3 enables the S1CR1 access on S1BRC address

S1CR1: Serial Port 1 Control Register 2

SFR Page = **1 and 2**
SFR Address = **0x9B & S1MOD3=1** RESET = 0000-0000

7	6	5	4	3	2	1	0
M6RER1	TXDO1	SOPWEC	TOTXD1	RMRI1	EXTFLT	TXINV1	RXINV1
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Before access S1CR1 on SFR address 0x9B with SFR page 1 or 2, software must set the S1MOD3=1. If S1MOD3=0, software accessed SFR address 0x9B with SFR page 1 or 2 would be S1BRC.

Bit 7: M6RER1, BMC Received data Error.
0: Must be cleared by software.
1: Set by hardware to indicate the BMC decoder error on receive data.

Bit 6: TXOD1, TXD1 signal output state in S1 module. Read only.
0: TXD1 signal output is logic low.
1: TXD1 signal output is logic high.

Bit 5: SOPWEC, 1st SOP packet decode including E-Code.
0: Disable SOPWEC function.
1: Enable SOPWEC function.

Bit 4: TOTXD1, Toggle TXD1 output signal. Software read it is always “0”.
0: No action when software writes “0” on this bit.
1: Software writes “1” on this bit will toggle the TXD1 signal output state.

Bit 3: RMRI1, Release Mask RI1 function.
0: RI1 function will be masked in PD TX mode.
1: Enable RI1 function in PD TX mode.

Bit 2: EXTFLT, Extend input Filter control. (Add TB81 to support EXTFLT1 function in S1 mode 6)
0: Disable EXTFLT function.
1: Enable EXTFLT function.

Bit 1: TXINV, TXD1 output inverted.
0: Disable signal inverted on TXD1 output.
1: Enable signal inverted on TXD1 output.

Bit 0: RXINV, RXD1 input inverted.
0: Disable signal inverted on RXD1 input.
1: Enable signal inverted on RXD1 input.

AUXR12: Auxiliary Register 12

SFR Page = 9 only
SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
CRCDS2	CRCM0	PDOES1	PDOES0	EDCM0	GPLC0	0	0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: CRCDS2. CRC0 Data port Selection bit 2.

Bit 6: CRCM0, CRC mode selection 0.
0: Select CRC16: 0x1021.
1: Select CRC32: 0x04C1_1DB7.

Bit 5~4: PDOES1~0, PDTXOE1 port pin Selection [1:0].

PDOES1~0	PDTXOE
0 0	P1.5
0 1	P3.3
1 0	P6.1
1 1	P3.2

Bit 3: EDCM0. EDC45 mode selection 0.

EDCM0	Mode #	EDC45 operation
0	0	4-bit to 5-bit encoder
1	1	5-bit to 4-bit decoder

Bit 2: GPLC0. GPL Control 0.
0: CPU accesses SFR address 0xA9 on SADDR.
1: CPU accesses SFR address 0xA9 on EDC45.

Bit 1~0: Reserved. Software must write “0” on these bits when AUXR12 is written.

20.7. ARGB Transmitter

- ARGB data output on TXD1 when S1 is set in enhanced mode.
- Programmable bit rate by S1BRG (S1 Baud Rate Generator).
- Selectable duty cycle on 1/4 or 1/3
- Selectable bus reset time detection on 60us, 260us, 200us and 320us.

Initial S1 to ARGB operating mode

- Set {S1TME, SM31, SM01, SM11} to operating ARGB mode.
- Write S1BRC/S1BRT SFR & S1TX12/S1MOD1 control bits to configure the Baud Rate on ARGB.
- Define S1DTY1 to configure ARGB duty cycle on 1/4 or 1/3.
- Select the bus reset time detection duration on S1IDT11 and S1IDT01.

Flow for Transmit a PD data frame

- Wait RI1(S1AGBRF) for bus reset event.
- Clear RI1, then start to write ARGB data on S1BUF.
- Wait TI1(S1AGBF) for S1 TX buffer empty.
- Clear TI1, then write next byte ARGB data on S1BUF…..and so on, until all ARGB data has been transmitted.
- Wait RI1to start a new frame to transmit ARGB data.

20.7.1. S1 ARGB Baud Rate

Figure 20–6. S1 ARGB Mode Baud Rate equation: (n=1)

$$\text{Sn ARGB Baud Rate} = \frac{1}{m} \times \frac{F_{SYSCLK}}{12 \times (256 - \text{SnBRT})} \quad ; \text{SnTX12=0} \quad \begin{matrix} ; \text{m=4, if SnDTY1 = 0} \\ ; \text{m=3, if SnDTY1 = 1} \end{matrix}$$

$$\text{or} = \frac{1}{m} \times \frac{F_{SYSCLK}}{1 \times (256 - \text{SnBRT})} \quad ; \text{SnTX12=1} \quad \begin{matrix} ; \text{m=4, if SnDTY1 = 0} \\ ; \text{m=3, if SnDTY1 = 1} \end{matrix}$$

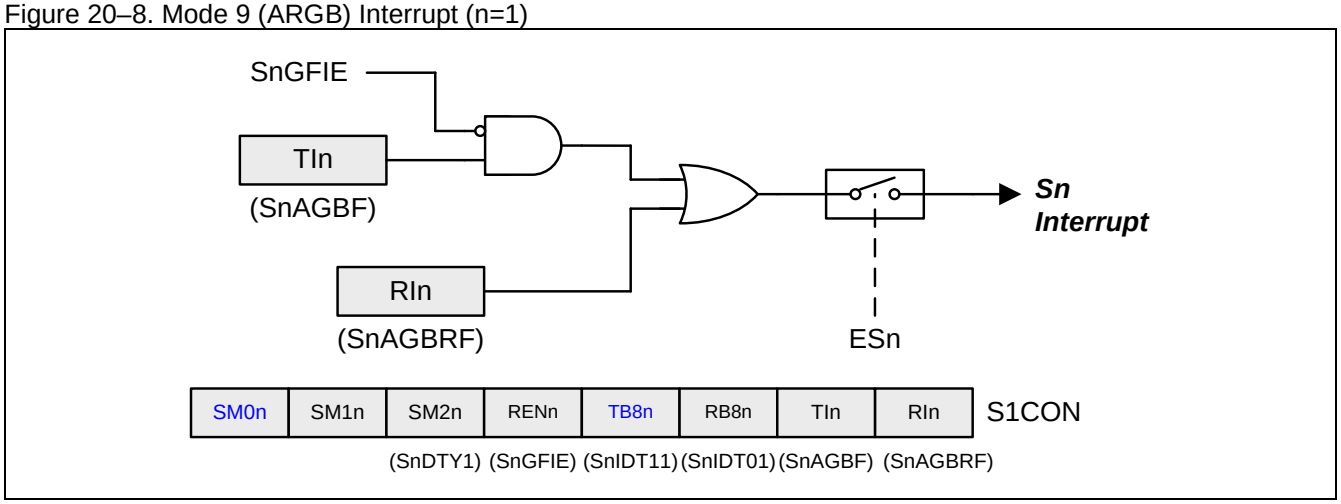
20.7.2. S1 ARGB Bus Reset Time

Figure 20–7. S1 ARGB Mode Bus Reset Time (n=1)

$$\text{Sn ARGB Reset Time} = \frac{8 \times m}{\text{Sn ARGB Baud Rate}} \quad ; \text{m=6/20/26/32, indexed by SnIDT1n~0n}$$

20.7.3. S1 ARGB Interrupt

Figure 20–8. Mode 9 (ARGB) Interrupt (n=1)



20.7.4. S1 ARGB Register

The following SPI special function registers are re-defined to the ARGB operation:

S1CON/ S1AGCR: Serial port 1 ARGB Control Register

SFR Page = **1 and 2**

SFR Address = 0x98 RESET = 0000-0000

7	6	5	4	3	2	1	0
SM01	SM11	SM21	REN1	TB81	RB81	TI1	RI1
SM01	SM11	S1DTY1	S1GFIE	S1IDT11	S1IDT01	S1AGBF	S1AGBRF
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 5: S1DTY1, ARGB signal Duty selection.
0: Select ARGB signal Duty to 3/4(H) or 1/4(L).
1: Select ARGB signal Duty to 2/3(H) or 1/3(L).

Bit 4: S1GFIE, TI1 interrupt Ignored.
0: Enabled TI1 interrupt. Default is enabled.
1: Disable TI1 interrupt.

Bit 3~2: S1IDT11~01, Selection for ARGB bus reset time duration.

S1IDT11~01	ARGB Bus Reset Time (when Baud Rate 800K)
0 0	**260us**
0 1	200us
1 0	**60us**
1 1	320us

Bit 1: S1AGBF, the flag reports Transmit Holding Register finishes the transfer to Output Shift Register in ARGB mode.
0: **The S1AGBF is cleared in software by writing "0" to this bit.**
1: When S1BUF empty, the S1AGBF bit is set, and an interrupt is generated if S1 interrupt is enabled.

Bit 0: S1AGBRF, ARGB bus Reset flag in ARGB mode.
0: The S1AGBRF flag is cleared in software by writing "0" to this bit.
1: This bit is set to logic 1 when bus meets the ARGB reset time. The reset time is defined by S1IDT11~01.

20.8. 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 20–9 and Figure 20–10 show 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 20–9. S1BRG Clock Output (S1BRG in 8-bit Timer Mode)

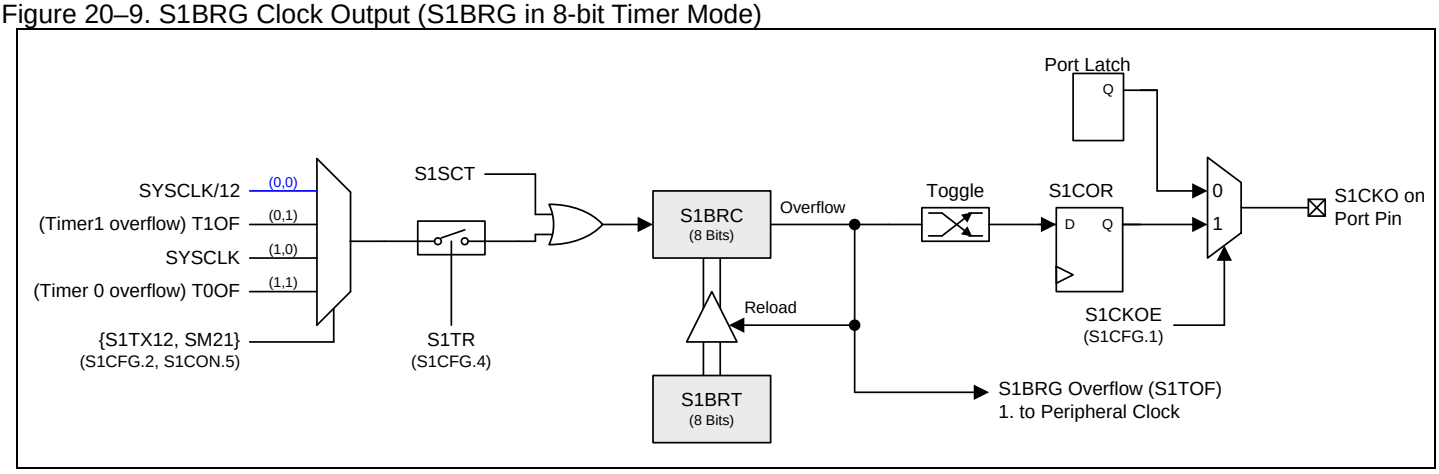
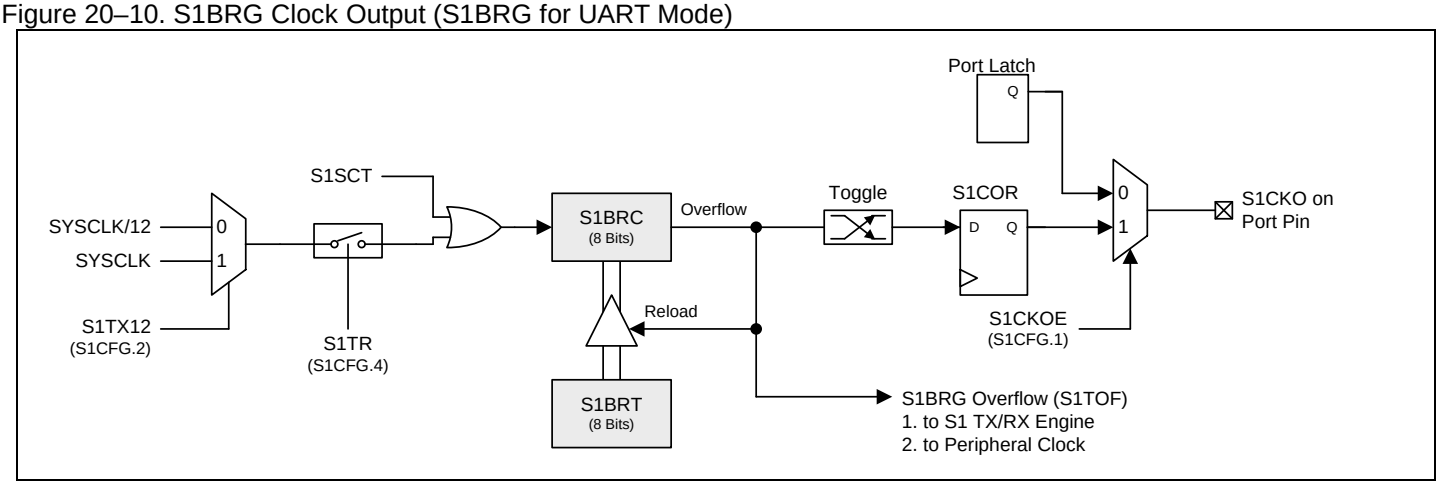


Figure 20–10. S1BRG Clock Output (S1BRG for UART Mode)



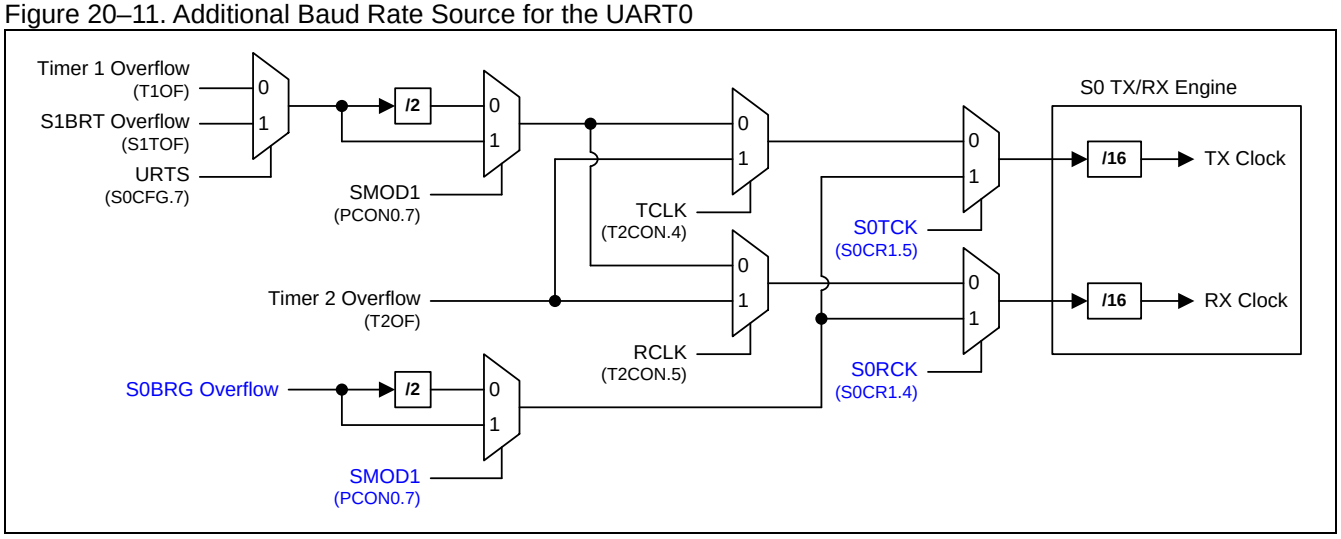
How to Program 8-bit S1BRG in Clock-out Mode

- 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.

20.9. 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 20–11. Additional Baud Rate Source for the UART0



When S1BRG is used as the baud rate generator of S0, the baud rate is as follows.

$$\text{Mode 1, 3 Baud Rate} = \frac{2^{SMOD1} \times 2^{(SMOD2 \times 2)}}{32} \times \frac{F_{SYSCLK}}{12 \times (256 - S1BRT)} \quad ; S1TX12=0$$

$$\text{or} = \frac{2^{SMOD1} \times 2^{(SMOD2 \times 2)}}{32} \times \frac{F_{SYSCLK}}{1 \times (256 - S1BRT)} \quad ; S1TX12=1$$

Note:
Please reference Section 19.9.4.1 for detail Baud Rate setting

20.10. 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 and 2**
SFR Address = 0x98 RESET = 0000-0000

7	6	5	4	3	2	1	0
SM01	SM11	SM21	REN1	TB81	RB81	TI1	RI1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: SM01, Serial port 1 mode bit 0.
Bit 6: SM11, Serial port 1 mode bit 1.

SM31	SM01	SM11	Mode	Description	Baud Rate
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, or /32
0	1	1	3	9-bit UART	variable
1	0	0	4	SPI Master	SYSCLK/12 or SYSCLK/4
1	0	1	5	Reserved	Reserved
1	1	0	6	BMC endec	Variable
1	1	1	7	Reserved	Reserved

Bit 5: Serial port 1 mode bit 2.
In Mode0 and Mode4:
The Tx initial signal level can be defined by SM21
0: To define the initial signal level of Tx is high.
1: To define the initial signal level of Tx is low.

In Modes 2 and 3:
0: Disable SM21 function.
1: Enable the automatic address recognition feature. 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 pure timer mode (Mode 8):
To use with S1TX12 to select the clock source.

S1TX12, SM21	S1BRG Clock Selection
0 0	SYSCLK/12
0 1	T1OF
1 0	SYSCLK
1 1	T0OF

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.
In mode 0 and mode 4, it is the selection on RX data sample edge.
0: Sample the RX data on trailing edge of the serial clock (TXD1).
1: Sample the RX data on leading edge of the serial clock (TXD1).

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 and 2**
SFR Address = 0x99 RESET = XXXX-XXXX

7	6	5	4	3	2	1	0
S1BUF.7	S1BUF.6	S1BUF.5	S1BUF.4	S1BUF.3	S1BUF.2	S1BUF.1	S1BUF.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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 and 2**
SFR Address = 0x9A RESET = 0000-0000

7	6	5	4	3	2	1	0
S1BRT.7	S1BRT.6	S1BRT.5	S1BRT.4	S1BRT.3	S1BRT.2	S1BRT.1	S1BRT.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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 Counter Register

SFR Page = **1 and 2**
SFR Address = 0x9B RESET = 0000-0000

7	6	5	4	3	2	1	0
S1BRC.7	S1BRC.6	S1BRC.5	S1BRC.4	S1BRC.3	S1BRC.2	S1BRC.1	S1BRC.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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 and 2**
SFR Address = **0x9C** RESET = 0010-0000

7	6	5	4	3	2	1	0
SM31	S1MOD3	S1DOR	S1TR	S1MOD1	S1TX12	S1CKOE	S1TME
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: SM31, Serial Port 1 Mode control bit 3.

SM31	SM01	SM11	Mode	Description	Baud Rate
0	0	0	0	shift register	SYSCLK/12 or SYSCL/4
0	0	1	1	8-bit UART	variable
0	1	0	2	9-bit UART	SYSCLK/64, or /32
0	1	1	3	9-bit UART	Variable
1	0	0	4	SPI Master	SYSCLK/12 or SYSCLK/4
1	0	1	5	Reserved	Reserved
1	1	0	6	BMC endec	Variable
1	1	1	7	Reserved	Reserved

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 6: S1MOD3, S1MOD3 enables the S1CR1 access on S1BRC address

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.

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.

Bit 2: S1TX12, UART1 Baud Rate Generator clock source select
To use with S1TX12 to select the clock source.

S1TX12, SM21	S1BRG Clock Selection
0 0	SYSCLK/12
0 1	T1OF
1 0	SYSCLK
1 1	T0OF

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 RESET = 0000-0000

7	6	5	4	3	2	1	0
SIDPS1	SIDPS0	T1G0	T0G1	0	0	S1PS1	S1PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

S1PS1~0	RXD1	TXD1
0 0	P1.2	P1.3
0 1	P6.0	P6.1
1 0	P1.0	P1.1
1 1	P3.4	P3.5

AUXR11: Auxiliary Register 11

SFR Page = 8 only
SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	I2C1PS1	I2C1PS0	RX1S0	0	T1CKOE	T0CKOE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 3: RX1S0, RXD1 Selection 0.
0: RXD1 is selected by S1PS1~0 (AUXR9.1~0).
1: Force RXD1 is selected on AC0OUT.

AUXR6: Auxiliary Register 6

SFR Page = **3 only**

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
KBIHPS1	KBIHPS0	KBILPS0	KBILPS0	T3FCS	T2FCS	SnMIPS	S0COPS
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 1: SnMIPS, S0MI, S1MI, S2MI & S3MI Port pin Selection.

SnMIPS	S0MI	S1MI
0	P1.6	P1.0
1	P3.3	P3.5

AUXR8: Auxiliary Register 8

SFR Page = 5 only

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	C1ICS1	C1ICS0	0	0	S1COPS	T3PS1	T3PS0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

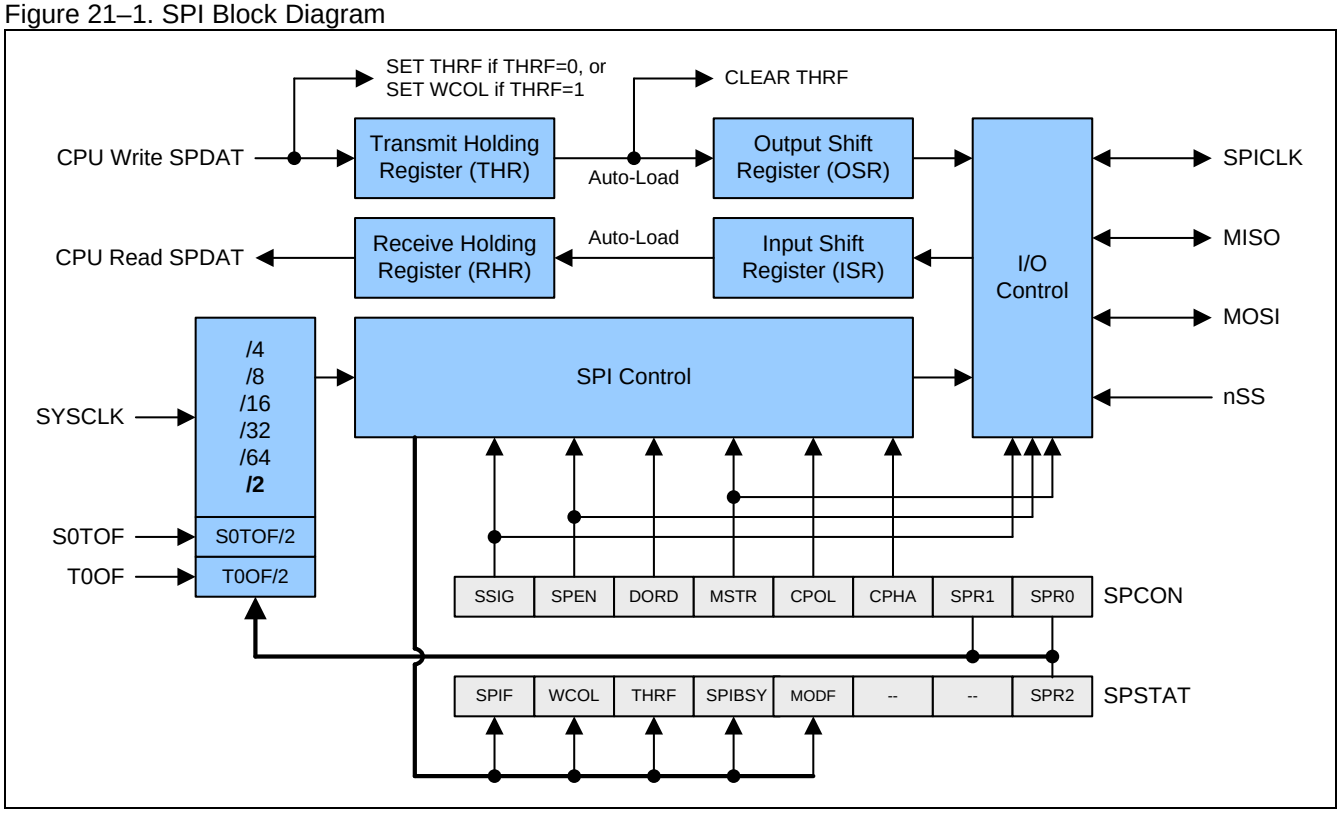
Bit 2: S1COPS, S1BRG Clock Output (S1CKO) port pin Selection.

S1COPS	S1CKO
0	P4.7
1	P4.5

21. Serial Peripheral Interface (SPI)

The **MG82F6P32** 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 **12**MHz 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.

Figure 21–1. 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.

• nSS (/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 nSS 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 nSS pin is ignored, i.e. SSIG (SPCTL.7) bit = 1, this pin is configured for port functions.

Note: See the AUXR10 in Section "4.3 Alternate Function Redirection", for its alternate pin-out option.

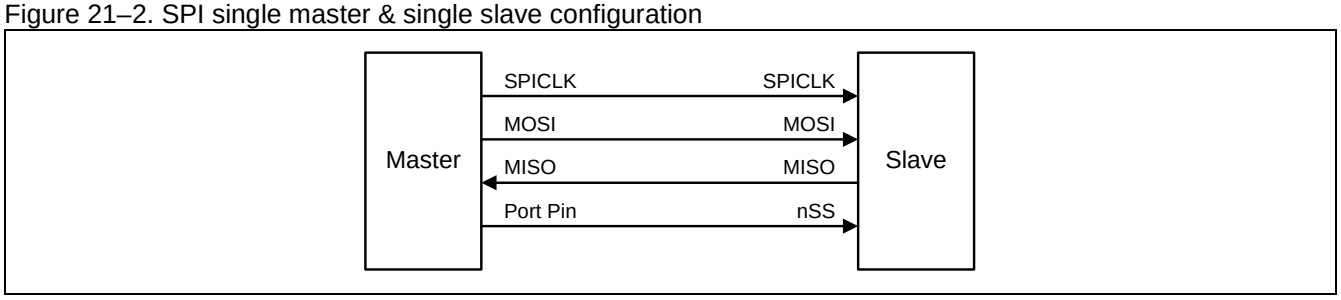
Note that even if the SPI is configured as a master (MSTR=1), it can still be converted to slave mode by the logic low of nSS pin input (if SSIG=0). Should this happen, the SPIF bit (SPSTAT.7) will be set and SPEN will be cleared. (See Section "21.2.3 Mode Change on nSS-pin")

21.1. Typical SPI Configurations

21.1.1. 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.

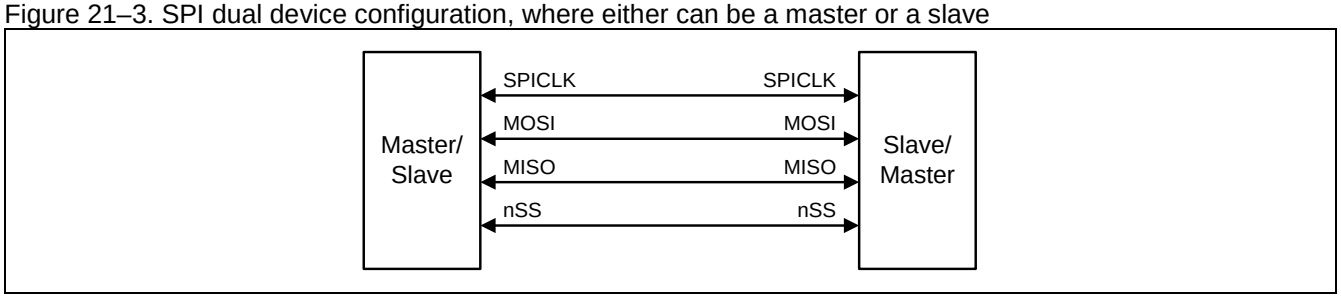
Figure 21–2. SPI single master & single slave configuration



21.1.2. 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 port pin configured in quasi-bidirectional mode or in open-drain mode with pull-up resistor. When any device initiates a transfer, it can configure nSS port pin as an output and drive it low to force a "mode change to slave" in the other device. (See Section "21.2.3 Mode Change on nSS-pin")

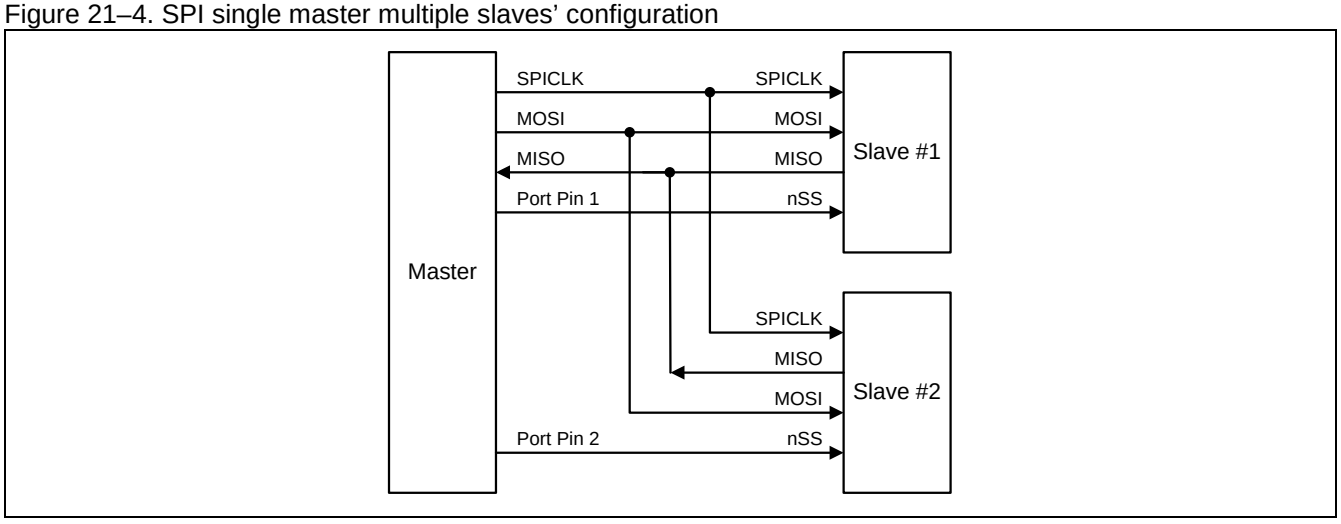
Figure 21–3. SPI dual device configuration, where either can be a master or a slave



21.1.3. 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 21–4. SPI single master multiple slaves' configuration



21.2. Configuring the SPI

Table 21–1 shows configuration for the master/slave modes as well as usages and directions for the modes.

Table 21–1. 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 pint are used as general port pins.
1	0	0	0	Salve (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 H/W 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".

21.2.1. 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.

21.2.2. 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.

21.2.3. Mode Change on nSS-pin

If SPEN=1, SSIG=0, MSTR=1 and nSS 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.

21.2.4. 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 mater 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).

21.2.5. Write Collision

The SPI in **MG82F6P32** 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.

21.2.6. 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 21–2.

Table 21–2. 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	3 MHz	12 MHz
0	0	1	SYSCLK/8	1.5 MHz	6 MHz
0	1	0	SYSCLK/16	750 KHz	3 MHz
0	1	1	SYSCLK/32	375 KHz	1.5 MHz
1	0	0	SYSCLK/64	187.5 KHz	750 KHz
1	0	1	SYSCLK/2	6 MHz	24 MHz
1	1	0	S0TOF/2	Variable	Variable
1	1	1	T0OF/2	Variable	Variable

Note:

1. *SYSCLK is the system clock.*
2. *S0TOF is UART0 Baud-Rate Generator Overflow.*
3. *T0OF is Timer 0 Overflow.*

21.3. 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 21–3. 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 21–5. SPI Slave Transfer Format with CPHA=0

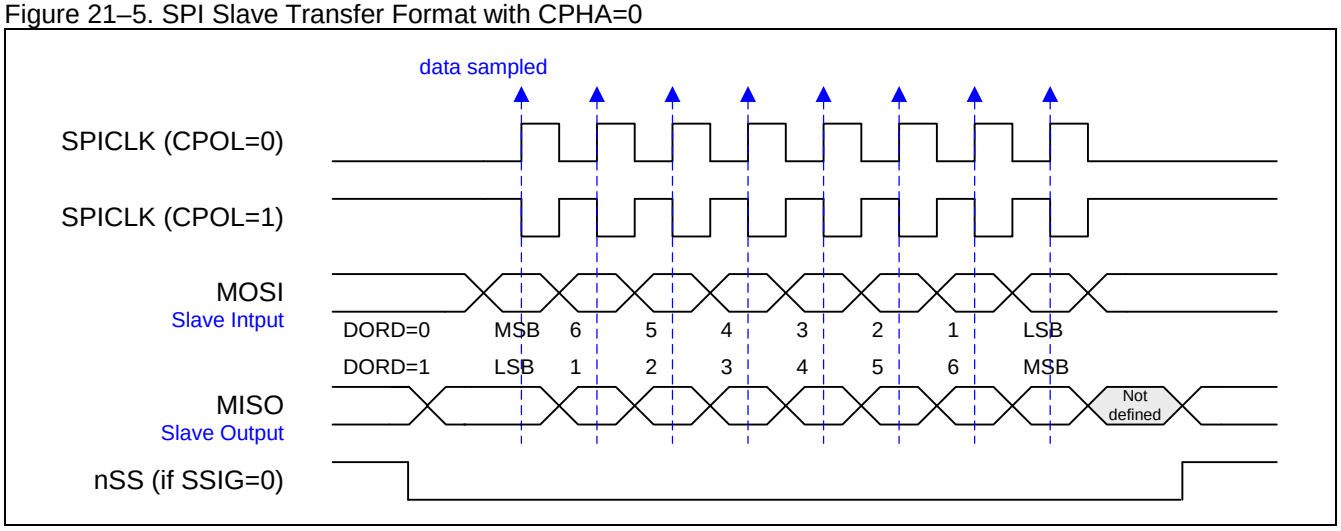


Figure 21–6. Slave Transfer Format with CPHA=1

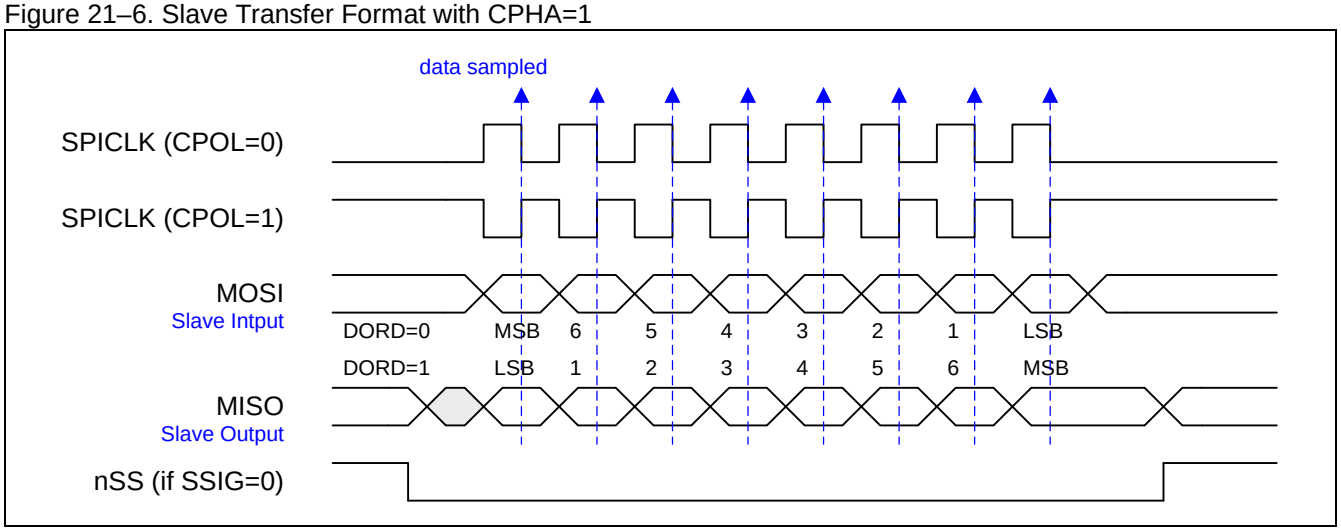


Figure 21–7. SPI Master Transfer Format with CPHA=0

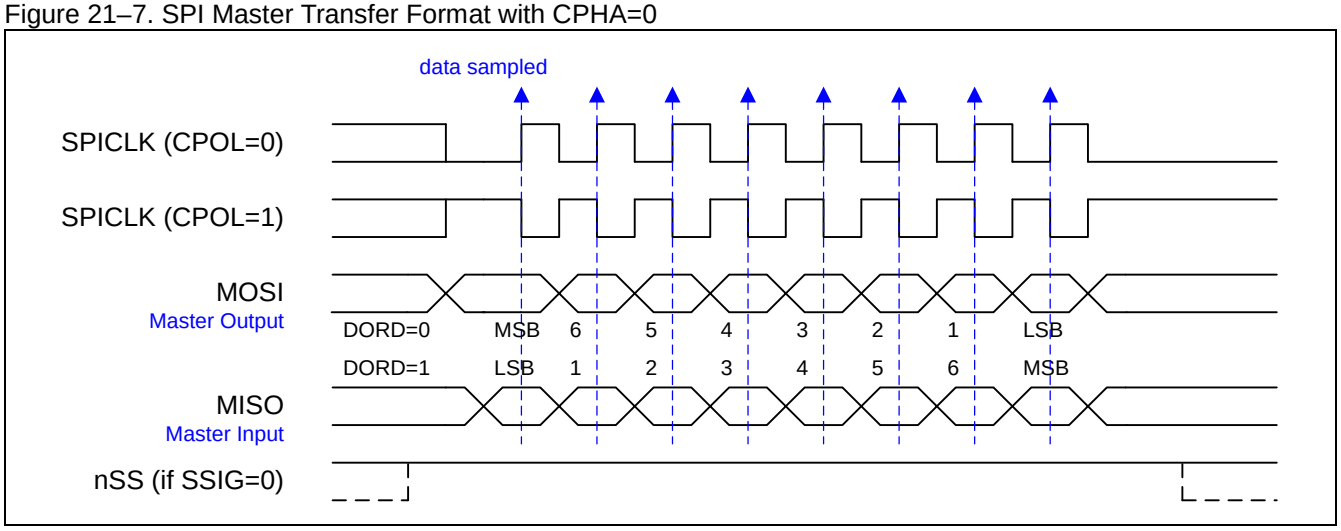
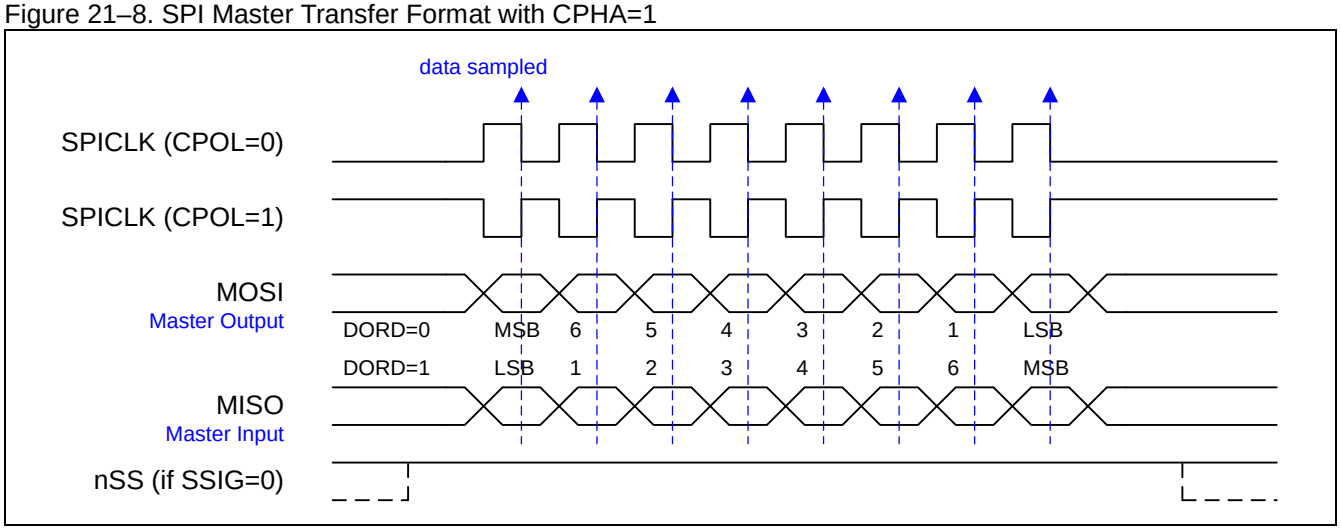


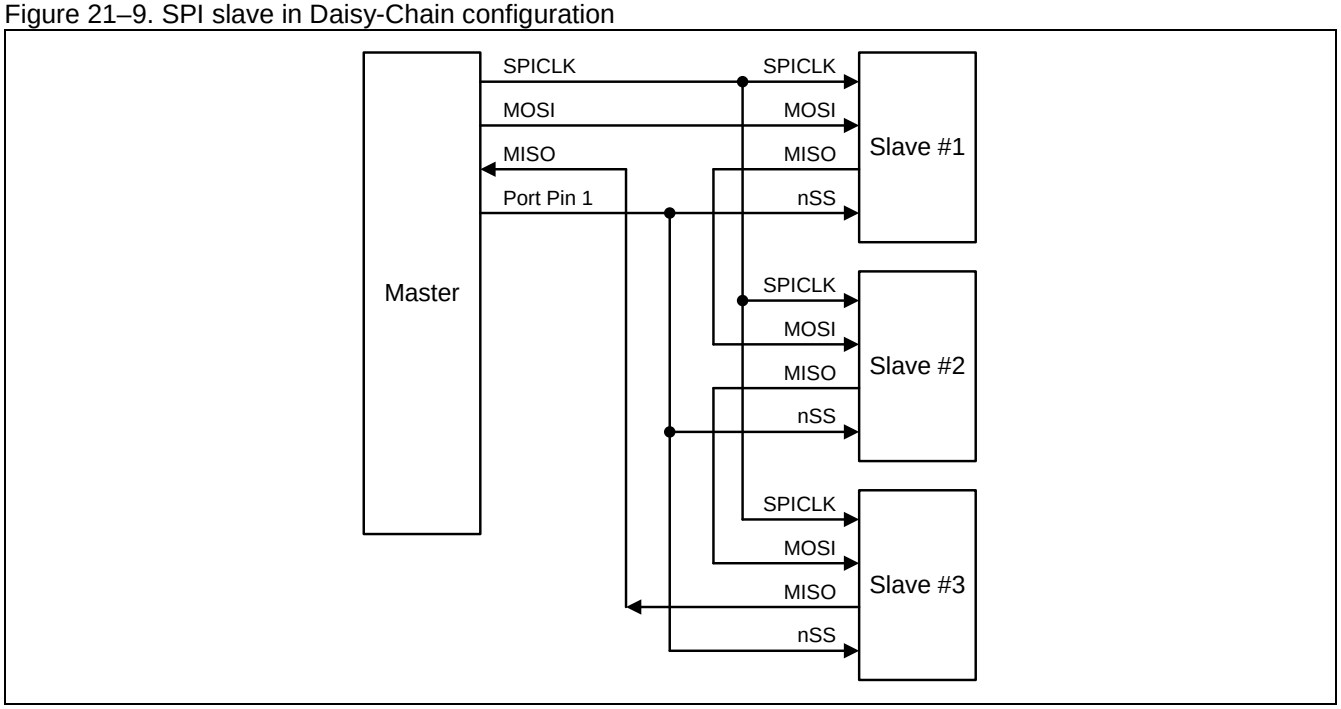
Figure 21–8. SPI Master Transfer Format with CPHA=1



21.4. 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 21–9. SPI slave in Daisy-Chain configuration



21.4.1. 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.

21.5. SPI Register

The following special function registers are related to the SPI operation:

SPCON: SPI Control Register

SFR Page = 0~7

SFR Address = 0x85 RESET = 0000-0100

7	6	5	4	3	2	1	0
SSIG	SPEN	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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 nSS 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	3 MHz	12 MHz
0	0	1	SYSCLK/8	1.5 MHz	6 MHz
0	1	0	SYSCLK/16	750 KHz	3 MHz
0	1	1	SYSCLK/32	375 KHz	1.5 MHz
1	0	0	SYSCLK/64	187.5 KHz	750 KHz
1	0	1	SYSCLK/2	6 MHz	24 MHz
1	1	0	S0TOF/2	Variable	Variable
1	1	1	T0OF/2	Variable	Variable

Note:

1. SYSCLK is the system clock.
2. S0TOF is UART0 Baud-Rate Generator Overflow.
3. T0OF is Timer 0 Overflow.

Note: Software must write "0" on SPR1~SPR0 when QPIEN is enabled.

SPCR1: SPI Control Register 1

SFR Page = 8 only

SFR Address = 0x85 RESET = 0000-0100

7	6	5	4	3	2	1	0
--	--	SPI0M0	SPSOES	SPFACE	SPR2	SPR1	SPR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5: SPI0M0, SPI mode, daisy-chain mode & ARGB mode.

SPI0M0	Operation Mode of SPI module
0	Standard SPI mode
1	SPI Daisy-Chain mode

Bit 4: SPSOES, SPI SO Early sample edge

0: Disable, SPI SO normal sample edge.

1: Enable. SPI SO early sample edge for faster SPI slave access rate.

Bit 3: SPFACE, SPIF Auto-Cleared Enable.

0: Disable, SPIF is only cleared by CPU software.

1: Enable. SPIF is also cleared by CPU read/write SPDAT operation.

Bit 2 ~ 0: SPR[2:0], SPI clock source selection.

SPR2	SPR1	SPR0	SPI Clock Selection	SPI Clock Rate @ SYSCLK=12MHz	SPI Clock Rate @ SYSCLK=48MHz
0	0	0	SYSCLK/4	3 MHz	12 MHz
0	0	1	SYSCLK/8	1.5 MHz	6 MHz
0	1	0	SYSCLK/16	750 KHz	3 MHz
0	1	1	SYSCLK/32	375 KHz	1.5 MHz
1	0	0	SYSCLK/64	187.5 KHz	750 KHz
1	0	1	SYSCLK/2	6 MHz	24 MHz
1	1	0	S0TOF/2	Variable	Variable
1	1	1	T0OF/2	Variable	Variable

SPSTAT: SPI Status Register

SFR Page = 0~F

SFR Address = 0x84 RESET = 0000-0000

7	6	5	4	3	2	1	0
SPIF	WCOL	THRF	SPIBSY	MODF	0	0	0
R/W	R/W	R	R	R/W	R/W	R/W	R/W

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 “21.2.5 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".

SPDAT: SPI Data Register

SFR Page = 0~F
SFR Address = 0x86 RESET = 0000-0000

7	6	5	4	3	2	1	0
(MSB)							(LSB)
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

SPDAT has two physical buffers for writing to and reading from during transmit and receive, respectively.

AUXR10: Auxiliary Register 10

SFR Page = 7 only
SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	SPIPS1	SPIPS0	S0PS1	P60OC1	P60OC0	P60FD
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 5~4: SPIPS1~0, SPI Port pin Selection 1~0.

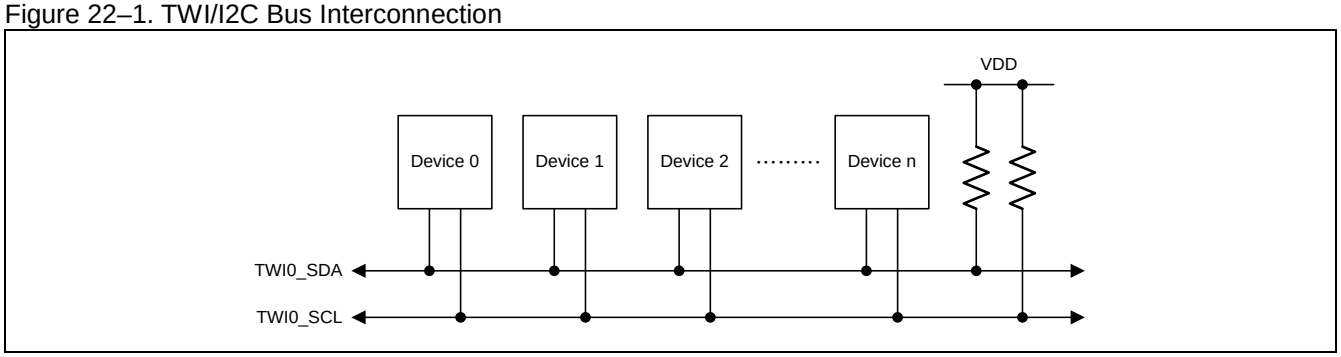
Pin Options	SPIPS1~0	nSS	MOSI	MISO	SPICLK
0	0 0	P1.4	P1.5	P1.6	P1.7
1	0 1	P0.1	P0.2	P4.1	P4.0
2	1 0	P3.3	P1.5	P1.6	P1.7
3	1 1	P1.7	P3.5	P3.4	P3.3

22. Two Wire serial Interface (TWI0/I2C0 & TWI1/I2C1)

The Two-Wire serial Interface is a two-wire, bi-directional serial bus. It is ideally suited for typical microcontroller applications. There are two TWI/I2C embedded in the **MG82F6P32**, TWI0/I2C0 and TWI1/I2C1. Both of them support the multiple slave address recognition.

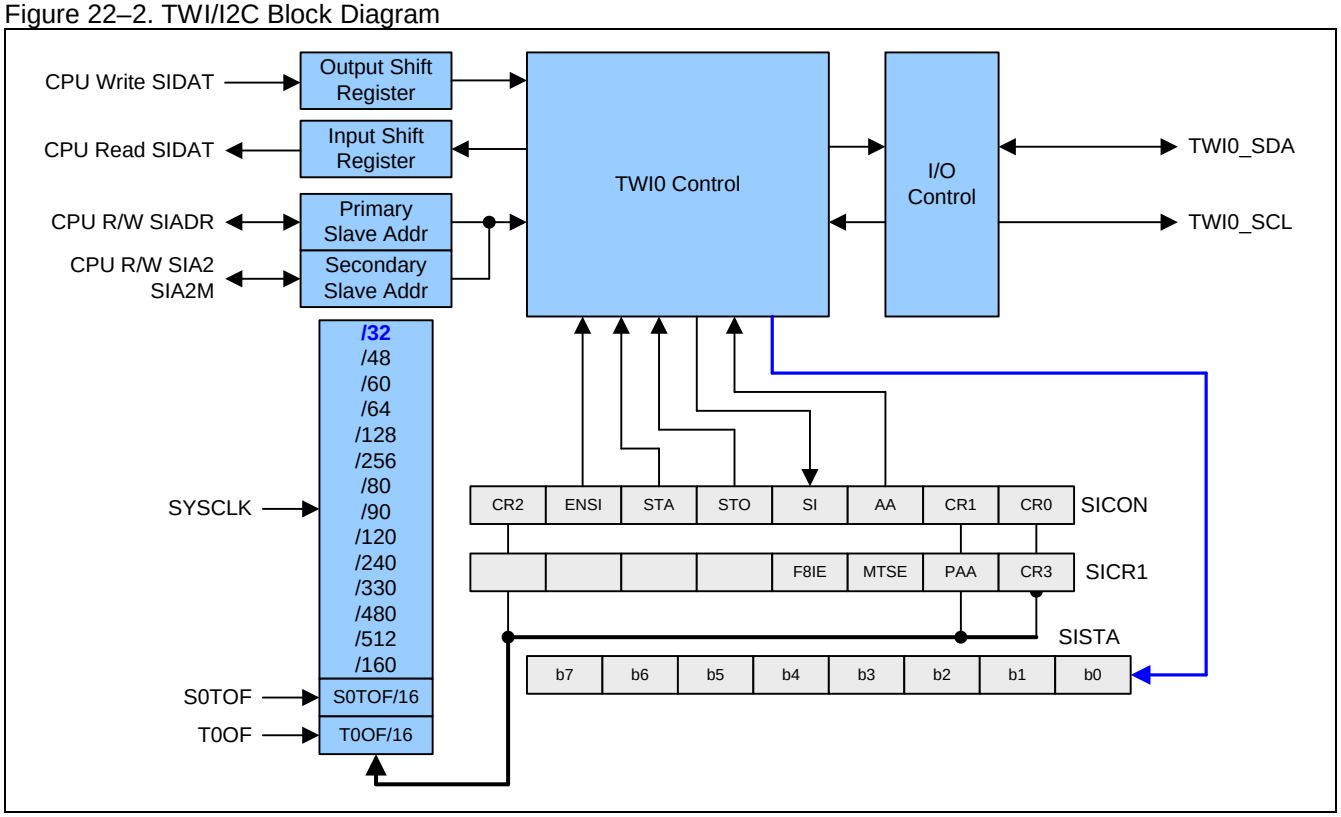
The TWI/I2C protocol allows the systems designer to interconnect up to 128 different devices using only two bi-directional bus lines, one for clock (TWI0_SCL) and one for data (TWI0_SDA). The TWI bus provides control of TWI0_SDA (serial data), TWI0_SCL (serial clock) generation and synchronization, arbitration 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 22–1. 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 and SIA2 are the slave address registers. And the TWI/I2C hardware interfaces to the serial bus via two lines: SDA (serial data line) and SCL (serial clock line).

Figure 22–2. TWI/I2C Block Diagram



22.1. Operating Modes

There are four operating modes for the TWI/I2C: 1) Master/Transmitter mode, 2) Master/Receiver mode, 3) Slave/Transmitter mode and 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.

22.1.1. 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

7	6	5	4	3	2	1	0
CR2	ENSI	STA	STO	SI	AA	CR1	CR0
Bit rate	1	0	0	0	x	Bit rate	

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.

22.1.2. 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.

22.1.3. 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

7	6	5	4	3	2	1	0
X	X	X	X	X	X	X	GC

|<---------- 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

7	6	5	4	3	2	1	0
CR2	ENSI	STA	STO	SI	AA	CR1	CR0
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.

22.1.4. 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.

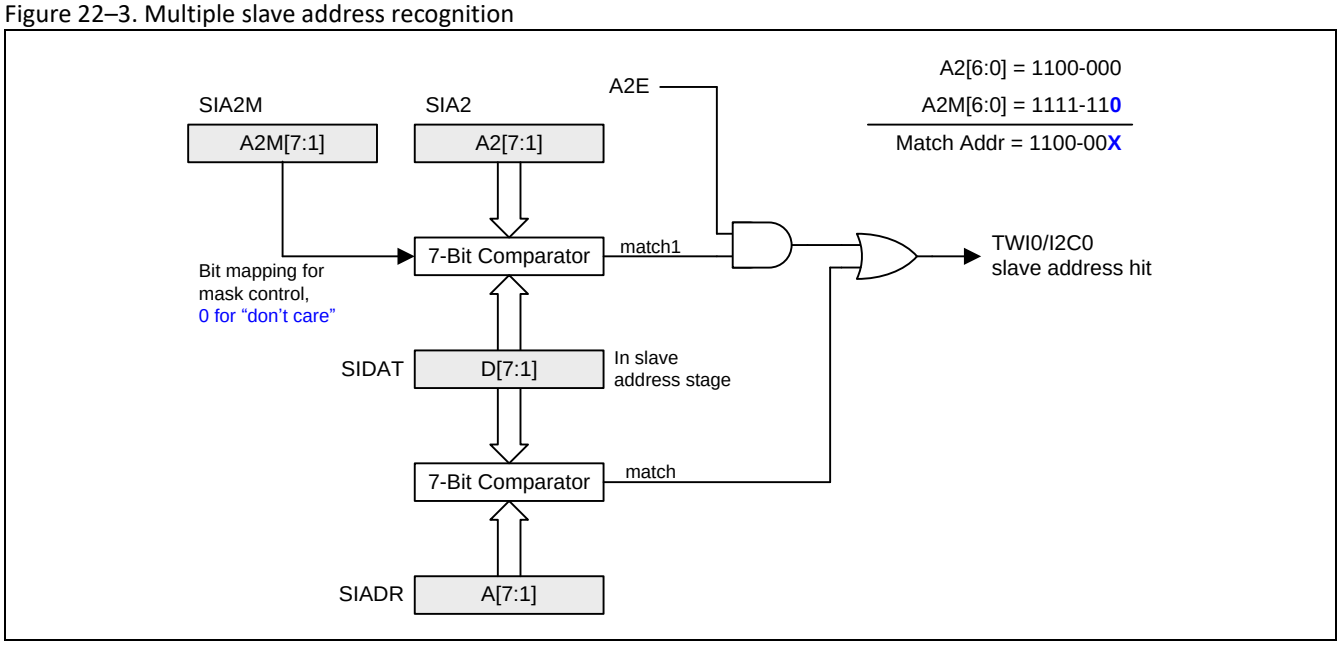
When SIADR and SICON have been initialized, TWI/I2C waits until it is addressed by its own slave address followed by the data direction bit which must be “0” (W) for TWI/I2C to operate in the slave receiver mode. After its own slave address and the W bit have been received, the serial interrupt flag (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.

22.1.5. Multiple slave address recognition

SIADR defines the primary slave address in **MG82F6P32** TWI/I2C. **MG82F6P32** also provide the secondary slave address with mask function that is implemented on SIA2 and SIA2M. A 1 in bit positions of the slave address mask SIA2M[7:1] enable a comparison between the received slave address and the secondary hardware's slave address SIA2[7:1] for those bits. A 0 in a bit of the slave address mask means that bit will be treated as a "don't care" for comparison purposes. In this case, either a 1 or a 0 value are acceptable on the incoming slave address.

Figure 22–3. Multiple slave address recognition



22.2. 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.

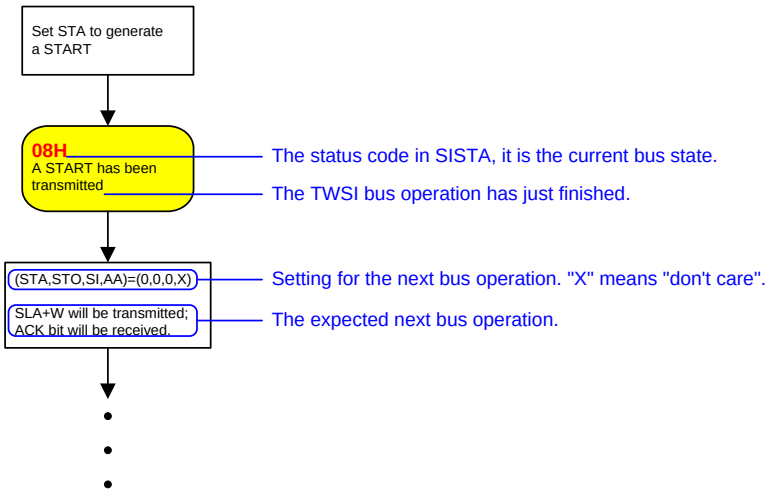
S1STA = 00H:
This status code indicates that a bus error has occurred during a TWI/I2C serial transfer. A bus error is caused when a START or STOP condition occurs at an illegal position in the format frame. Examples of such illegal positions are during the serial transfer of an address byte, a data 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 TWI0_SDA and TWI0_SCL lines are released (a STOP condition is not transmitted).

22.3. Using the TWI/I2C

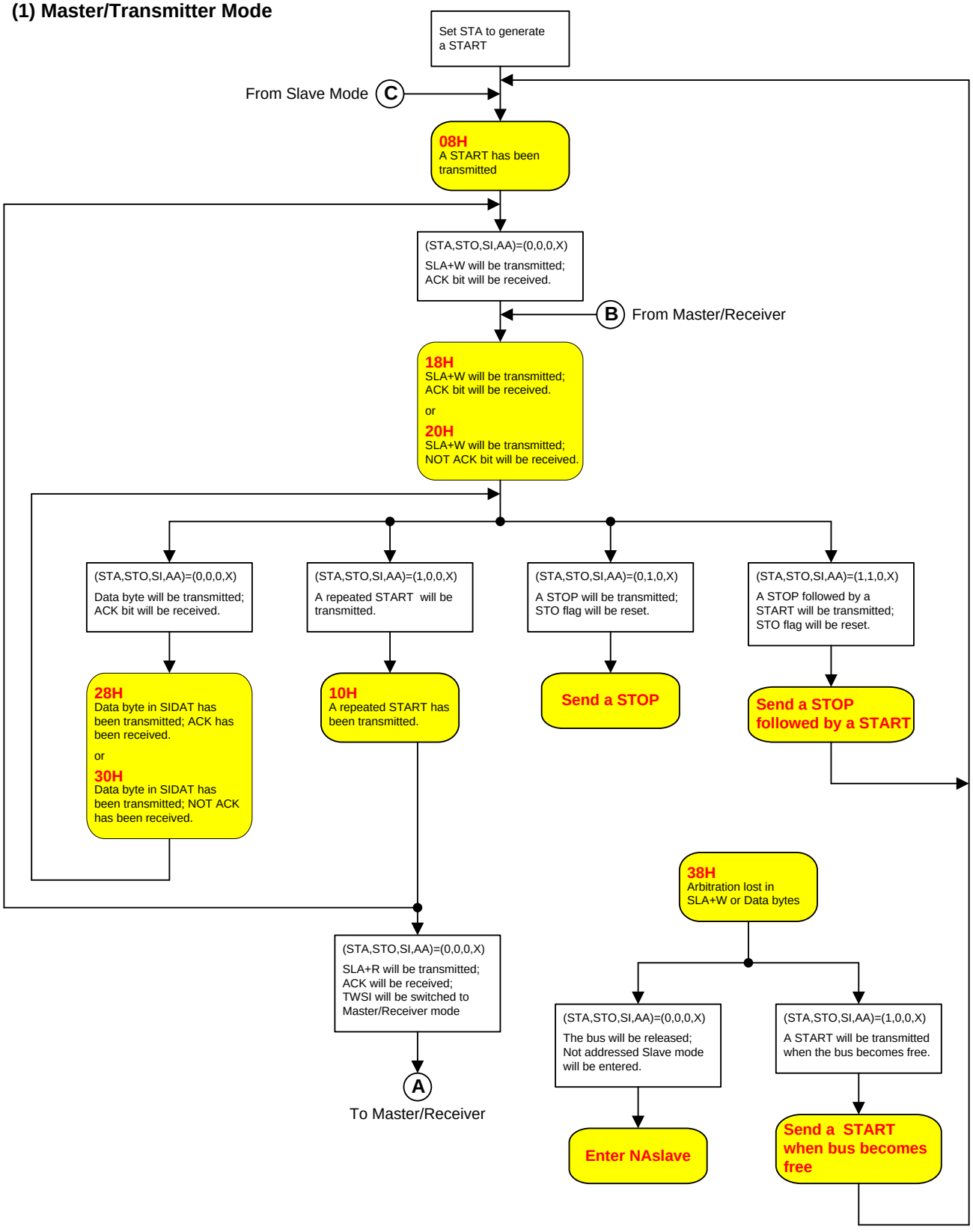
The TWI/I2C is byte-oriented and interrupt based. Interrupts are issued after all bus events, like reception of a byte or transmission of a START condition. Because the TWI/I2C is interrupt-based, the application software is free to carry on other operations during a TWI/I2C byte transfer. Note that the TWI0/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”

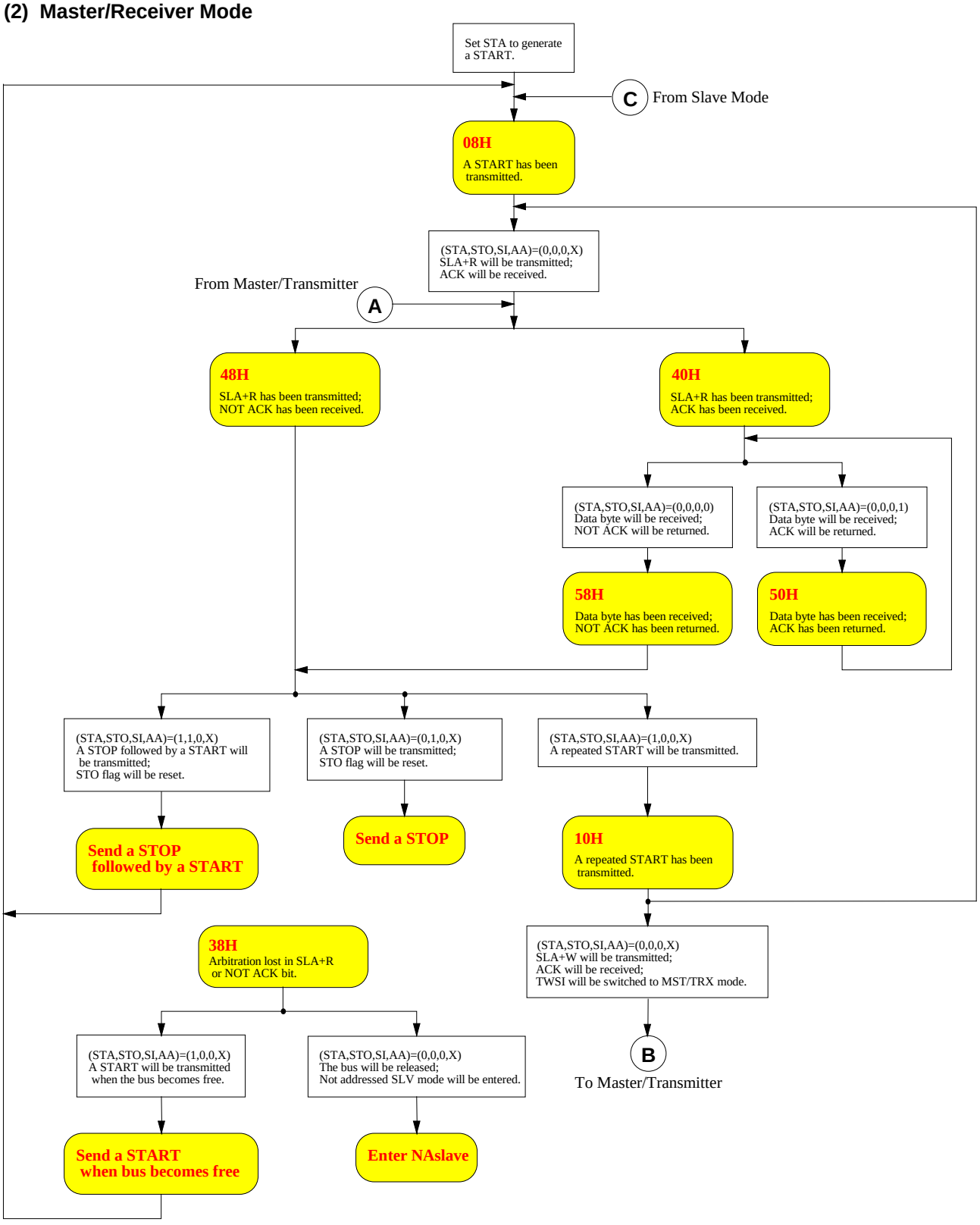
The figure below shows how to read the flow charts.



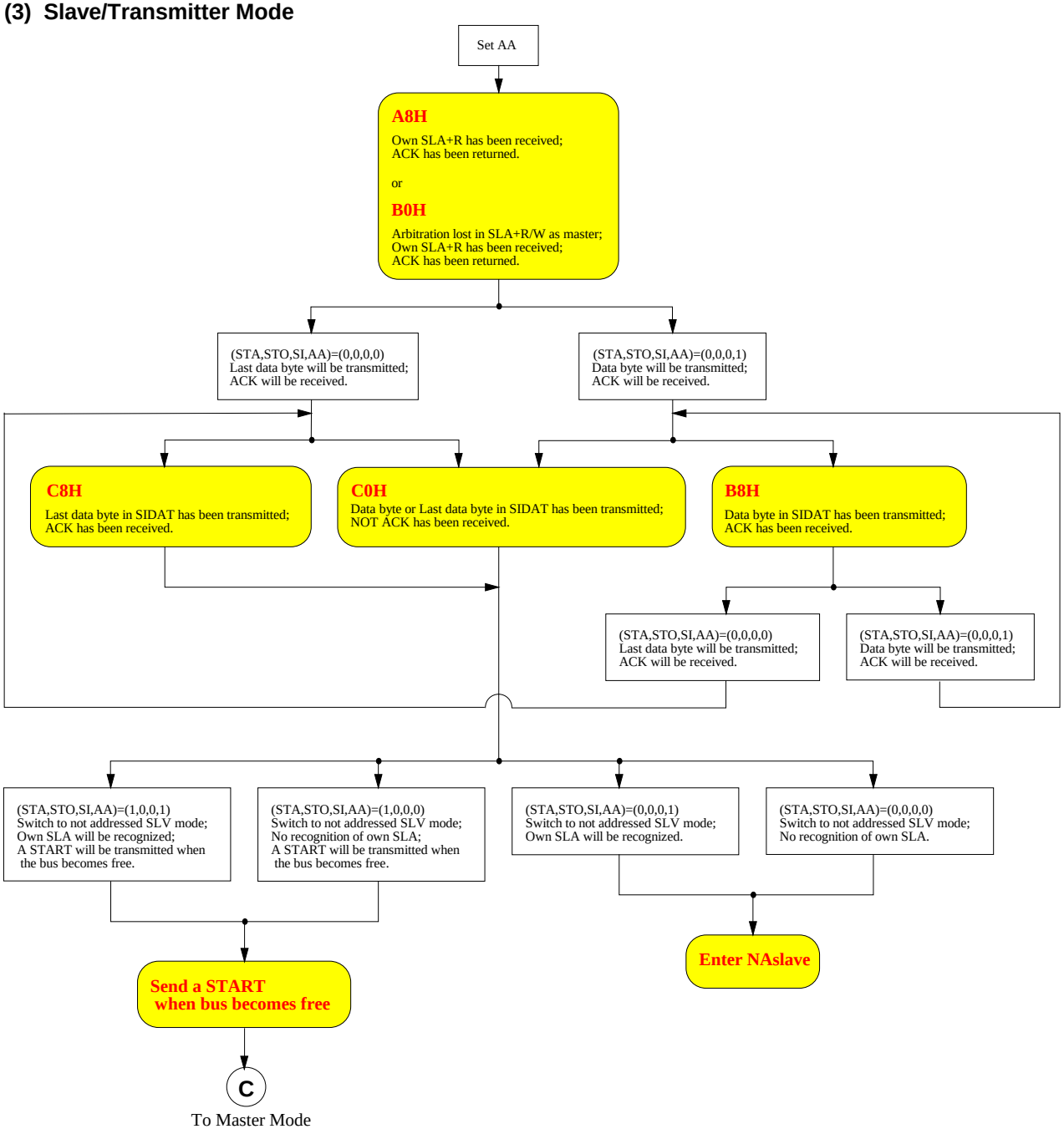
(1) Master/Transmitter Mode



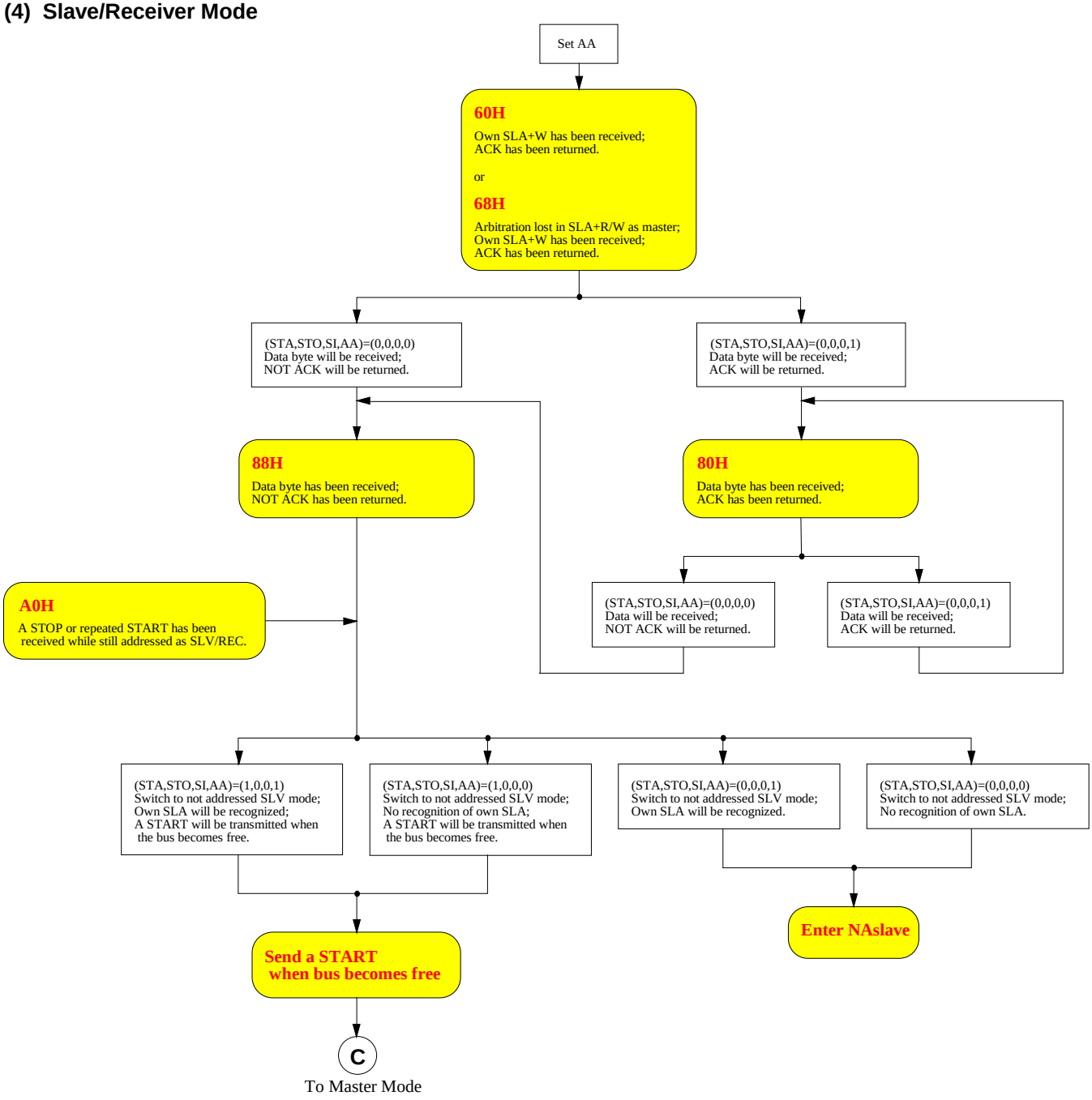
(2) Master/Receiver Mode



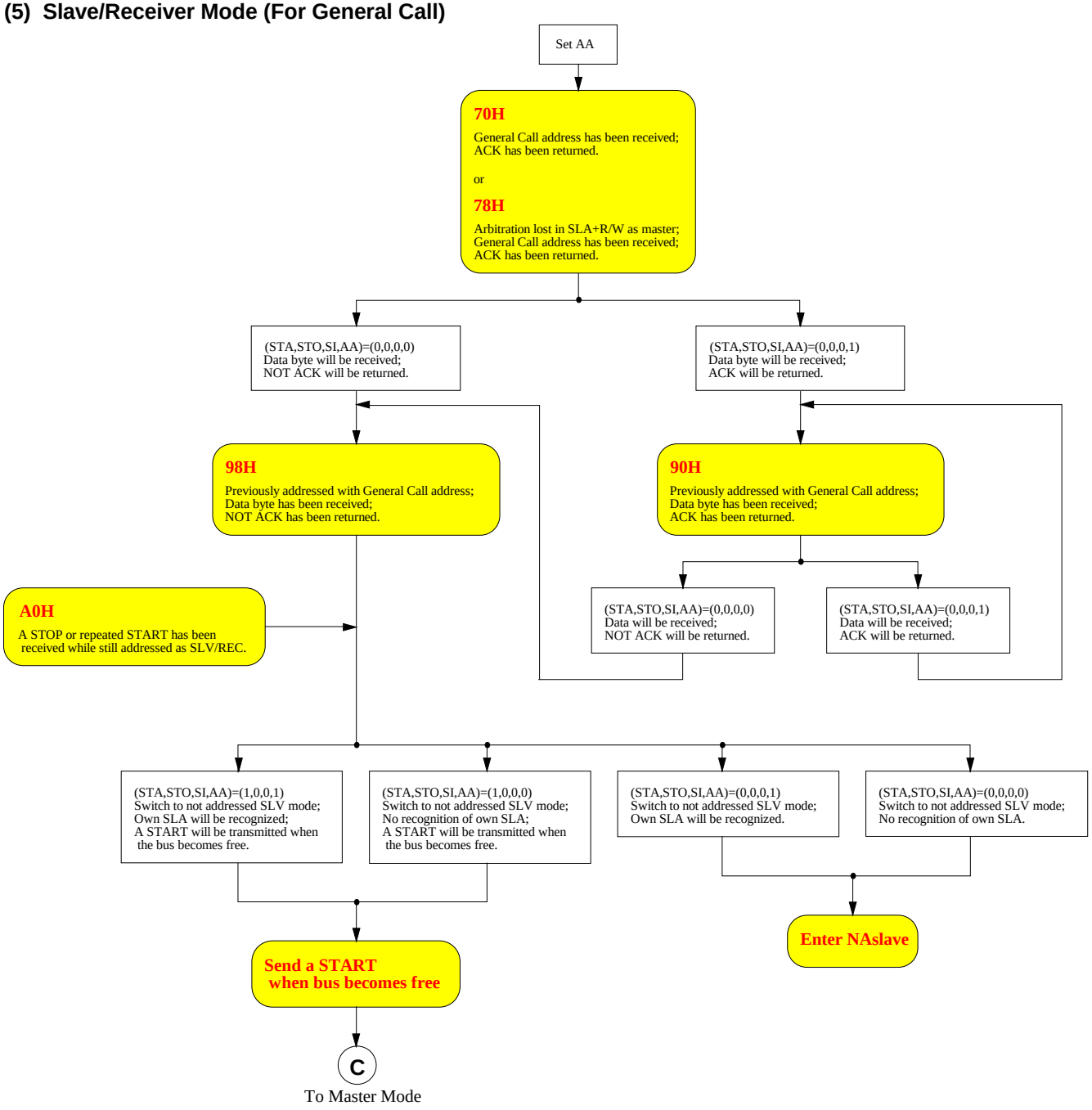
(3) Slave/Transmitter Mode



(4) Slave/Receiver Mode



(5) Slave/Receiver Mode (For General Call)



22.4. TWI0/I2C0 Register

SIADR: TWI0/I2C0 Address Register

SFR Page = **0 Only**

SFR Address = 0xD1 RESET = 0000-0000

7	6	5	4	3	2	1	0
A6	A5	A4	A3	A2	A1	A0	GC
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7 ~ 1: TWI0/I2C0 Address, 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. The most significant bit corresponds to the first bit received from the TWI0/I2C0 bus after a START condition.

Bit 0: GC, if GC is set, the general call address (00H) is recognized; otherwise it is ignored.

SIDAT: TWI0/I2C0 Data Register

SFR Page = **0 Only**

SFR Address = 0xD2 RESET = 0000-0000

7	6	5	4	3	2	1	0
SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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_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_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 Only**

SFR Address = 0xD4 RESET = 0000-0000

7	6	5	4	3	2	1	0
CR2	ENSI	STA	STO	SI	AA	CR1	CR0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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 combine with CR3 to 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 22–1.

Table 22–1. TWI0/I2C0 Serial Clock Rates

CR3	CR2	CR1	CR0	TWI0/I2C0 Clock Selection	TWI0/I2C0 Clock around 100KHz		TWI0/I2C0 Clock around 400KHz	
					System Clock	TWI0/I2C0 Clock	System Clock	TWI0/I2C0 Clock
0	0	0	0	SYSCLK/32	N/A	N/A	12MHz	375KHz
0	0	0	1	SYSCLK/48	6M Hz	125 KHz	N/A	N/A
0	0	1	0	SYSCLK/60	6M Hz	100 KHz	24M Hz	400 KHz
0	0	1	1	SYSCLK/64	6M Hz	93.75 KHz	24M Hz	375 KHz
0	1	0	0	SYSCLK/128	12M Hz	93.75 KHz	48M Hz	375 KHz
0	1	0	1	SYSCLK/256	24M Hz	93.75 KHz	N/A	N/A
0	1	1	0	S0TOF/16	Variable	Variable	Variable	Variable
0	1	1	1	T0OF/16	Variable	Variable	Variable	Variable
1	0	0	0	SYSCLK/80	8M Hz	100 KHz	32M Hz	400 KHz
1	0	0	1	SYSCLK/90	8M Hz	88.89 KHz	36M Hz	400 KHz
1	0	1	0	SYSCLK/120	12M Hz	100 KHz	48M Hz	400 KHz
1	0	1	1	SYSCLK/240	24M Hz	100 KHz	N/A	N/A
1	1	0	0	SYSCLK/330	32M Hz	96.97 KHz	N/A	N/A
1	1	0	1	SYSCLK/480	48M Hz	100 KHz	N/A	N/A
1	1	1	0	SYSCLK/512	48M Hz	93.75 KHz	N/A	N/A
1	1	1	1	SYSCLK/160	16M Hz	100 KHz	N/A	N/A

Note:1. The Maximum TWI0/I2C0 *clock Rate should under 1MHz, to set SYSCLK = 32MHz to generate 1MHz.*
2. SYSCLK is the system clock.
3. S0TOF is UART0 Baud-Rate Generator Overflow.
4. T0OF is Timer 0 Overflow.

SICR1: TWI0/I2C0 Control Register 1

SFR Page = **2 Only**
SFR Address = 0xD4 RESET = 0000-0000

7	6	5	4	3	2	1	0
				F8IE	MTSE	PAA	CR3
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 3: F8IE, I2C idle state to generate SI.
0: Disable, STSTA == 0xF8 (I2C idle state) to generate SI (interrupt flag)
1: Enable STSTA == 0xF8 (I2C idle state) to generate SI (interrupt flag)

Bit 2: MTSE, Multi-Speed Enable (disable input filter); Define the clock source of the digital filer of input signal
0: Use a fixed scale of the input signal for digital filter
1: Use SYSCLK x 3 of digital filte for the input signal for multi speed on the I2C bus.

Bit 1: PAA, Pre-Assert Acknowledge for TWI0/I2C0.
0: Disable PAA function.
1: Enable PAA function on DMA transfer with TWI0/I2C0 master RX and slave TX/RX.

Bit 0: CR3, to use with CR2 ~ CR0, the Clock Rate select Bits

SISTA: TWI0/I2C0 Status Register

SFR Page = **0 Only**
SFR Address = 0xD3 RESET = 1111-1000

7	6	5	4	3	2	1	0
SIS7	SIS6	SIS5	SIS4	SIS3	SIS2	SIS1	SIS0
R	R	R	R	R	R	R	R

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.

SIA2: TWI0/I2C0 2nd Address Register

SFR Page = **2 Only**
SFR Address = 0xD1 RESET = 0000-0000

7	6	5	4	3	2	1	0
A2.6	A2.5	A2.4	A2.3	A2.2	A2.1	A2.0	A2E
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~1: 2nd slave address content of TWI0/I2C0.

Bit 0: A2E, Enable control of 2nd slave address recognition.
0: Disable 2nd slave address recognition.
1: Enable 2nd slave address recognition.

SIA2M: TWI0/I2C0 2nd Address Mask Register

SFR Page = **2 Only**
SFR Address = 0xD2 RESET = 1111-1111

7	6	5	4	3	2	1	0
A2M.6	A2M.5	A2M.4	A2M.3	A2M.2	A2M.1	A2M.0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SIA2 register is combined with ISA2M register for 2nd address recognition. In fact, SIA2M functions as the “mask” register for SIA2 register. The following is the example for it.

SIA2[7:1] = 1100 000
SIA2M[7:1] = 1111 110
2nd ADR[7:1] = 1100 00x ⟶ The 2nd slave address will be checked except bit 1 is treated as “don’t care”

Bit 0: Reserved. Software must write “1” on this bit when SIA2M is written.

AUXR3: Auxiliary Register 3

SFR Page = **0 only**
SFR Address = 0xA4 RESET = 0000-0000

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

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

TWIPS1~0	TWI0_SCL	TWI0_SDA
0 0	P4.0	P4.1
0 1	P6.0	P6.1
1 0	P3.1	P3.0
1 1	P2.2	P2.4

22.5. TWI1/I2C1 Register

SI1ADR: TWI1/I2C1 Address Register

SFR Page = **1 Only**

SFR Address = 0xD1 RESET = 0000-0000

7	6	5	4	3	2	1	0
A61	A51	A41	A31	A21	A11	A01	GC1
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7 ~ 1: TWI1/I2C1 Address, the CPU can read from and write to this register directly. SI1ADR is not affected by the TWI1/I2C1 hardware. The contents of this register are irrelevant when TWI1/I2C1 is in a master mode. In the slave mode, the seven most significant bits must be loaded with the microcontroller's own slave address. The most significant bit corresponds to the first bit received from the TWI1/I2C1 bus after a START condition.

Bit 0: GC1, if GC1 is set, the general call address (00H) is recognized; otherwise it is ignored.

SI1DAT: TWI1/I2C1 Data Register

SFR Page = **1 Only**

SFR Address = 0xD2 RESET = 0000-0000

7	6	5	4	3	2	1	0
SD71	SD61	SD51	SD41	SD31	SD21	SD11	SD01
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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 TWI1/I2C1 is in a defined state and the serial interrupt flag (SI1) is set. Data in SI1DAT remains stable as long as SI is set. While data is being shifted out, data on the bus is simultaneously being shifted in; SI1DAT 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 SI1DAT.

SI1DAT 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 TWI1/I2C1 hardware and cannot be accessed by the CPU. Serial data is shifted through the ACK flag into SI1DAT on the rising edges of serial clock pulses on the TWI1_SCL line. When a byte has been shifted into SI1DAT, the serial data is available in SI1DAT, and the acknowledge bit is returned by the control logic during the 9th clock pulse. Serial data is shifted out from SI1DAT on the falling edges of clock pulses on the TWI1_SCL line.

When the CPU writes to SI1DAT, the bit SD71 is the first bit to be transmitted to the SDA line. After nine serial clock pulses, the eight bits in SI1DAT 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 SI1DAT.

SI1CON: TWI1/I2C1 Control Register

SFR Page = **1 Only**

SFR Address = 0xD4 RESET = 0000-0000

7	6	5	4	3	2	1	0
CR21	ENSI1	STA1	STO1	SI1	AA1	CR11	CR01
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

The CPU can read and write to this register directly. Two bits are affected by the TWI/I2C1 hardware: the SI1 will be set when a serial interrupt occurred, and the STO1 will be cleared when a STOP condition is present on the bus. The STO1 bit is also cleared when ENSI1="0".

Bit 7: CR21, TWI1/I2C1 Clock Rate select bit 2 (associated with CR11 and CR01).

Bit 6: ENSI1, the TWI1/I2C1 Hardware Enable Bit
When ENSI1 is "0", the TWI1 _SDA and TWI1_SCL outputs are in a high impedance state, and it will ignore the input signals. Under this condition, the TWI1/I2C1 is in the not-addressed slave state, and STO1 is forced to "0". No other bits are affected, and the TWI1_SDA and TWI1_SCL can be used as general purpose I/O pins. When ENSI1 is "1", TWI1 is enabled, the TWI1_SDA and TWI1_SCL assign to port pin latch, such as P4.3 and P4.2. 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: STA1, the START Flag
When sets the STA1 to enter master mode, the TWI1/I2C1 hardware will check the status of the serial bus. It will generate a START condition if the bus is free. Otherwise TWI1/I2C1 will wait for a STOP condition and generates a START condition after a delay. If STA1 is set while TWI1/I2C1 is already in a master mode and one or more bytes are transmitting or receiving, TWI1/I2C1 will send a repeated START condition. STA1 may be set at any time. STA1 may also be set when TWI1/I2C1 is an addressed slave mode. When the STA1 bit is reset, no START condition or repeated START condition will be generated.

Bit 4: STO1, the STOP Flag
When the STO1 is set while TWI1/I2C1 is in a master mode, a STOP condition is transmitted to the serial bus. When the STOP condition is detected on the bus, the TWI1/I2C1 hardware clears the STO1 flag. In a slave mode, the STO1 flag may be set to recover from a bus error condition. In this case, no STOP condition is transmitted to the bus. However, the TWI1/I2C1 hardware behaves as if a STOP condition has been received and switches to the defined not addressed slave receiver mode. The STO1 flag is automatically cleared by hardware. If the STA1 and STO1 bits are both set, then a STOP condition is transmitted to the bus if TWI1/I2C1 is in a master mode (in a slave mode, TWI1/I2C1 generates an internal STOP condition which is not transmitted), and then transmits a START condition.

Bit 3: SI1, the Serial Interrupt Flag
When a new TWI1/I2C1 state is present in the SI1STA register, the SI1 flag is set by hardware. And, if the TWI1/I2C1 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 SI1 is set, the low period of the serial clock on the TWI1_SCL line is stretched, and the serial transfer is suspended. A high level on the TWI1_SCL line is unaffected by the serial interrupt flag. SI1 must be cleared by software writing “0” on this bit. When the SI1 flag is reset, no serial interrupt is requested, and there is no stretching on the serial clock on the TWI1_SCL line.

Bit 2: AA1, the Assert Acknowledge Flag
If the AA1 flag is set to “1”, an Acknowledge (low level to TWI1_SDA) will be returned during the acknowledge clock pulse on the TWI1_SCL line when:
1) The own slave address has been received.
2) A data byte has been received while TWI1/I2C1 is in the master/receiver mode.
3) A data byte has been received while TWI1/I2C1 is in the addressed slave/receiver mode.

If the AA1 flag is reset to “0”, a not acknowledge (high level to TWI1_SDA) will be returned during the acknowledge clock pulse on TWI1_SCL when:
1) A data has been received while TWI1/I2C1 is in the master/receiver mode.
2) A data byte has been received while TWI1/I2C1 is in the addressed slave/receiver mode.

Bit 7, 1~0: CR21, CR11 and CR01, the Clock Rate select Bits
These three bits determine the serial clock frequency when TWI1/I2C1 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. TWI1/I2C1 will automatically synchronize with any clock frequency from master, which is up to 400KHz. The various serial clock rates are shown in Table 22–2.

Table 22–2. TWI1/I2C1 Serial Clock Rates

CR31	CR21	CR11	CR01	TWI1/I2C1 Clock Selection	TWI1/I2C1 Clock around 100KHz		TWI1/I2C1 Clock around 400KHz	
					System Clock	TWI0/I2C0 Clock	System Clock	TWI0/I2C0 Clock
0	0	0	0	SYSCLK/32	N/A	N/A	12MHz	375KHz
0	0	0	1	SYSCLK/48	6M Hz	125 KHz	N/A	N/A
0	0	1	0	SYSCLK/60	6M Hz	100 KHz	24M Hz	400 KHz
0	0	1	1	SYSCLK/64	6M Hz	93.75 KHz	24M Hz	375 KHz
0	1	0	0	SYSCLK/128	12M Hz	93.75 KHz	93.75 KHz	48M Hz
0	1	0	1	SYSCLK/256	24M Hz	93.75 KHz	N/A	N/A
0	1	1	0	S0TOF/16	Variable	Variable	Variable	Variable
0	1	1	1	T0OF/16	Variable	Variable	Variable	Variable
1	0	0	0	SYSCLK/80	8M Hz	100 KHz	32M Hz	400 KHz
1	0	0	1	SYSCLK/90	8M Hz	88.89 KHz	36M Hz	400 KHz
1	0	1	0	SYSCLK/120	12M Hz	100 KHz	48M Hz	400 KHz
1	0	1	1	SYSCLK/240	24M Hz	100 KHz	N/A	N/A
1	1	0	0	SYSCLK/330	32M Hz	96.97 KHz	N/A	N/A
1	1	0	1	SYSCLK/480	48M Hz	100 KHz	N/A	N/A
1	1	1	0	SYSCLK/512	48M Hz	93.75 KHz	N/A	N/A
1	1	1	1	SYSCLK/160	16M Hz	100 KHz	N/A	N/A

Note:1. The Maximum TWI1/I2C1 *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.

SI1CR1: TWI1/I2C1 Control Register 1

SFR Page = **3 Only**
SFR Address = 0xD4 RESET = 0000-0000

7	6	5	4	3	2	1	0
				F8IE1	MTSE1	PAA1	CR31
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 3: F8IE1, I2C1 idle state to generate SI.
0: Disable, STSTA == 0xF8 (I2C idle state) to generate SI (interrupt flag)
1: Enable STSTA == 0xF8 (I2C idle state) to generate SI (interrupt flag)

Bit 2: MTSE1, Multi-Speed Enable (disable input filter); Define the clock source of the digital filer of input signal
0: Use a fixed scale of the input signal for digital filter
1: Use SYSCLK x 3 of digital filte for the input signal.

Bit 1: PAA1, Pre-Assert Acknowledge for TWI0/I2C1.
0: Disable PAA1 function.
1: Enable PAA1 function on DMA transfer with TWI1/I2C1 master RX and slave TX/RX.

Bit 0: CR31, to use with CR21 ~ CR01, the Clock Rate select Bits

SI1STA: TWI1/I2C1 Status Register

SFR Page = **1 Only**
SFR Address = 0xD3 RESET = 1111-1000

7	6	5	4	3	2	1	0
SI1S7	SI1S6	SI1S5	SI1S4	SI1S3	SI1S2	SI1S1	SI1S0
R	**R**	**R**	**R**	**R**	**R**	**R**	**R**

SI1STA 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 SI1STA contains F8H, no serial interrupt is requested. All other SI1STA values correspond to defined TWI1/I2C1 states. When each of these states

is entered, a status interrupt is requested (SI1=1). A valid status code is present in SI1STA 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.

SI1A2: TWI1/I2C1 2nd Address Register

SFR Page = **3 Only**
SFR Address = 0xD1 RESET = 0000-0000

7	6	5	4	3	2	1	0
A21.6	A21.5	A21.4	A21.3	A21.2	A21.1	A21.0	A2E1
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~1: 2nd slave address content of TWI1/I2C1.

Bit 0: A2E1, Enable control of 2nd slave address recognition.
0: Disable 2nd slave address recognition.
1: Enable 2nd slave address recognition.

SI1A2M: TWI1/I2C1 2nd Address Mask Register

SFR Page = **3 Only**
SFR Address = 0xD2 RESET = 1111-1111

7	6	5	4	3	2	1	0
A2M1.6	A2M1.5	A2M1.4	A2M1.3	A2M1.2	A2M1.1	A2M1.0	1
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

SI1A2 register is combined with SI1A2M register for 2nd address recognition. In fact, SI1A2M functions as the "mask" register for SI1A2 register. The following is the example for it.

SI1A2[7:1] = 1100 000
SI1A2M[7:1] = 1111 110
2nd ADR[7:1] = 1100 00x ⟶ The 2nd slave address will be checked except bit 1 is treated as "don't care"

Bit 0: Reserved. Software must write "1" on this bit when SI1A2M is written.

AUXR11: Auxiliary Register 11

SFR Page = 8 only
SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	I2C1PS1	I2C1PS0	RX1S0	0	T1CKOE	T0CKOE
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 6: PAA1, Pre-Assert Acknowledge for TWI1/I2C1.
0: Disable PAA1 function.
1: Enable PAA1 function on DMA transfer with TWI1/I2C1 master RX and slave TX/RX.

Bit 5~4: I2C1PS1~0, TWI1/I2C1 Port pin in Selection [1:0].

I2C1PS1~0	TWI1_SCL	TWI1_SDA
0 0	P1.0	P1.1
0 1	P6.0	P6.1
1 0	P3.0	P3.1
1 1	TWI0_SCL	TWI0_SDA

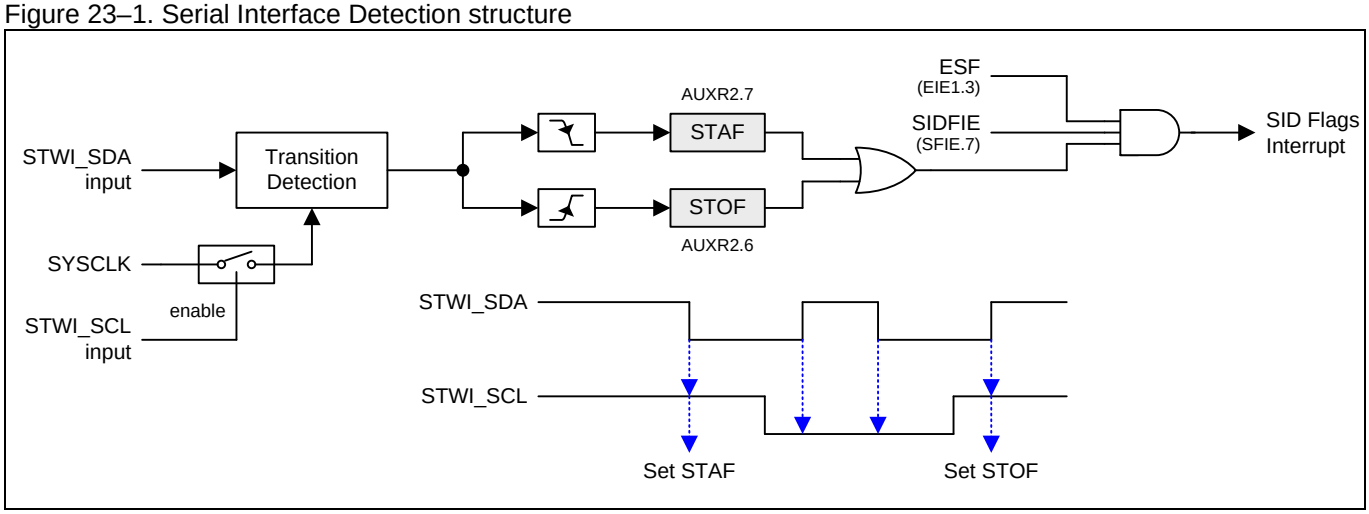
23. Serial Interface Detection (STWI/SI2C)

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/I2C slave device.

23.1. SID Structure

Figure 23–1 shows the configuration of STAF and STOF detection, interrupt architecture and event detecting waveform.

Figure 23–1. Serial Interface Detection structure



23.2. SID Register

AUXR2: Auxiliary Register 2

SFR Page = 0~F
SFR Address = 0xA3 RESET = 0000-0000

7	6	5	4	3	2	1	0
STAF	STOF	C1PLK	C0PLK	T1X12	T0X12	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 POR = 0110-0000

7	6	5	4	3	2	1	0
SIDFIE	MCDRE	MCDFIE	RTCFIE	SPWIE	BOF1IE	BOF0IE	WDTFIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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.

AUXR9: Auxiliary Register 9

SFR Page = 6 only
SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
SIDPS1	SIDPS0	T1G1	T0G1	0	0	S1PS1	S1PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

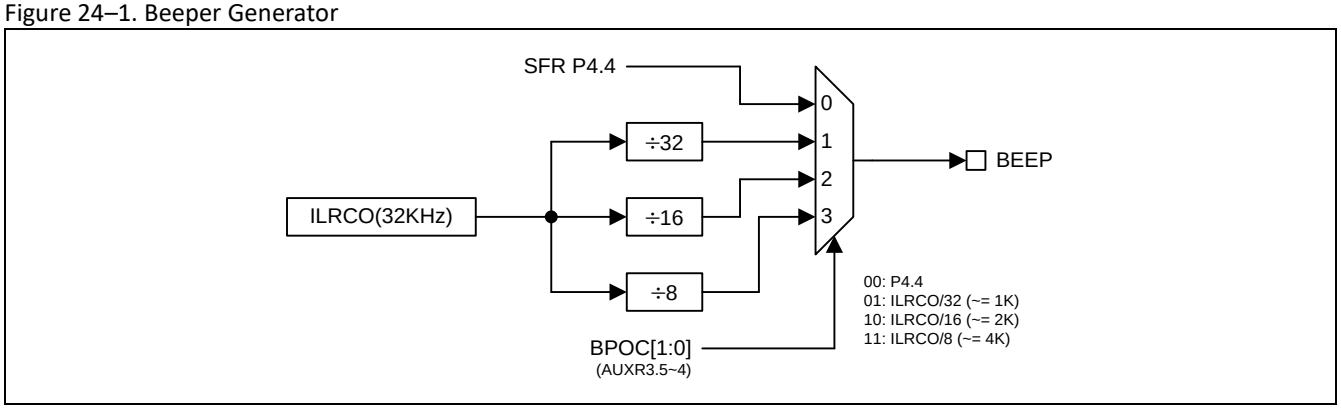
Bit 7~6: SID/STWI Port pin Selection [1:0].

SIDPS1~0	STWI_SCL	STWI_SDA
0 0	nINT1	S0MI
0 1	TWI0_SCL	TWI0_SDA
1 0	TWI1_SCL	TWI1_SDA
1 1	T2EXI	T3EXI

24. 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 24–1 shows the beeper generator circuit. But ILRCO is not the precision clock source. Please refer Section "37.6 ILRCO Characteristics" for more detailed ILRCO frequency deviation range.

Figure 24–1. Beeper Generator



24.1. Beeper Register

AUXR3: Auxiliary Register 3

SFR Page = **0 only**

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
T0PS1	T0PS0	BPOC1	BPOC0	S0PS0	TWIPS1	TWIPS0	T0XL
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 5~4: BPOC1~0, Beeper output control bits.

BPOC[1:0]	P4.4 function	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

For beeper on P4.4 function, it is recommended to configure P4.4 as push-push output mode.

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 RESET = 100x-x011

7	6	5	4	3	2	1	0
HSE	IAPO	HSE1	--GF	--GF	IORCTL	RSTIO	OCDE
R/W	**R/W**	**R/W**	**W**	**W**	**W**	**R/W**	**R/W**

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.

25. Keypad Interrupt (KBI)

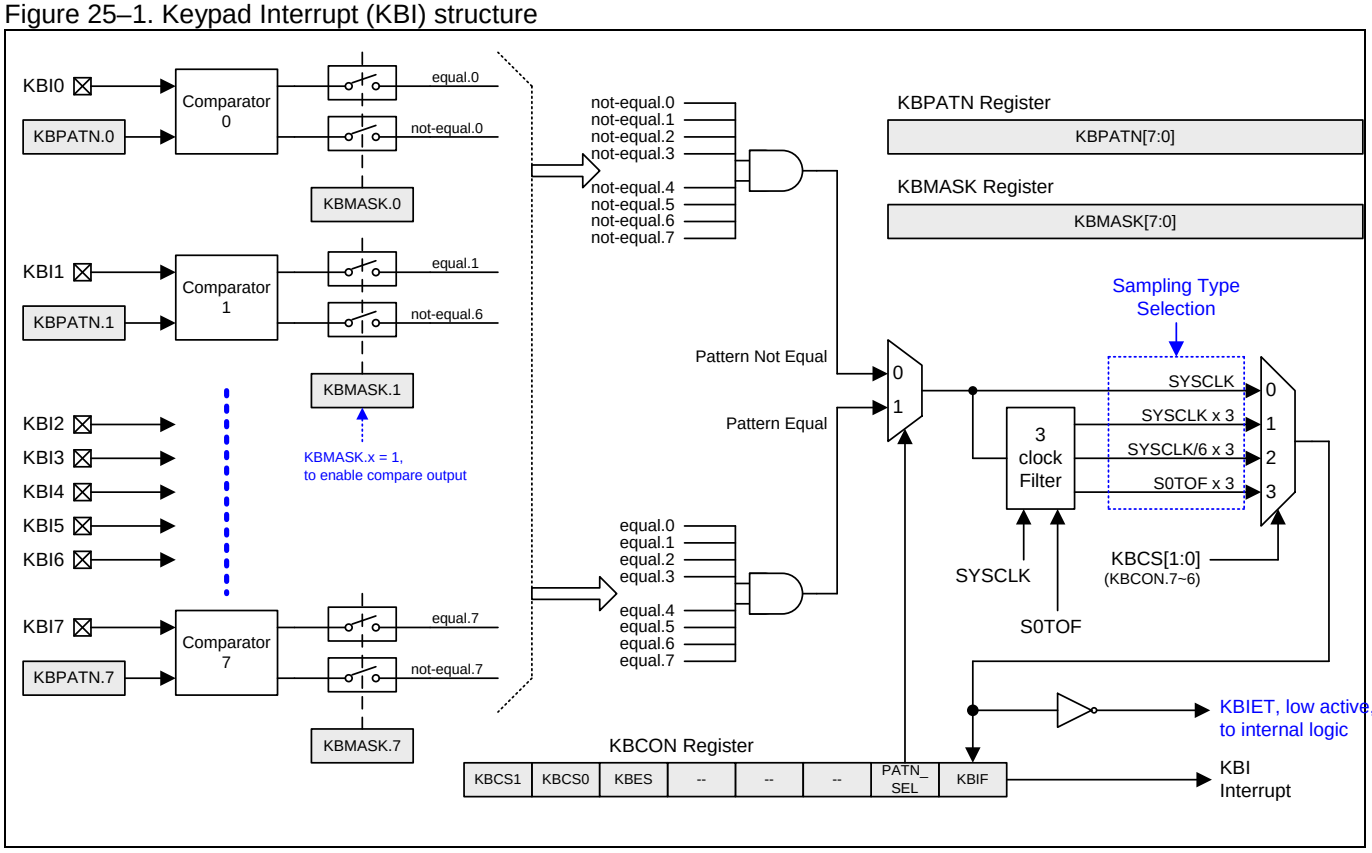
The Keypad Interrupt function is intended primarily to allow a single interrupt to be generated when KBI.7~0 is equal to or not equal to a certain pattern. This function can be used for bus address recognition or keypad recognition.

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 induce 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.

25.1. KBI Structure

Figure 25–1. Keypad Interrupt (KBI) structure



25.2. KBI Register

The following special function registers are related to the KBI operation:

KBPATN: Keypad Pattern Register

SFR Page = 0~F
SFR Address = 0xD5 RESET = 1111-1111

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

Bit 7~0: KBPATN.7~0: The keypad pattern, reset value is 0xFF.

KBCON: Keypad Control Register

SFR Page = 0~F
SFR Address = 0xD6 RESET = 0000-0000

7	6	5	4	3	2	1	0
KBCS1	KBCS0	KBES				PATN_SEL	KBIF
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Bit 7~6: KBCS1~0, KBI Filter mode control.

KBCS1~0	KBI input filter mode
00	Disabled
01	SYSCLK x 3
10	SYSCLK/6 x 3
11	S0TOF x 3

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 RESET = 0000-0000

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

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 RESET = 0000-0000

7	6	5	4	3	2	1	0
KBIHPS1	KBIHPS0	KBILPS1	KBILPS0	T3FCS	T2FCS	SnMIPS	S0COPS
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~6: KBIHPS1~0, KBI4~7 Port pin Selection [1:0].

KBIHPS1~0	KBI7	KBI6	KBI5	KBI4
0 0	P1.7	P1.6	P1.5	P1.4
0 1	P1.7	P1.6	P3.5	P3.4
1 0	P2.6	P2.4	P2.3	P2.2
1 1	P1.7	P1.6	P6.1	P6.0

Bit 5~4: KBILPS1~0, KBI0~3 Port pin Selection 0.

KBILPS1~0	KBI3	KBI2	KBI1	KBI0
0 0	P1.3	P1.2	P1.1	P1.0
0 1	P3.3	P3.2	P3.1	P3.0
1 0	P0.7	P0.5	P0.2	P0.1
1 1	P4.5	P4.4	P4.1	P4.0

26. General Purpose Logic (GPL-CRC)

The **MG82F6P32** 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.

Set CRCM0 to select CRC32: 0x04C1_1DB7.

26.1. 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-Read 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-Read mode.
4. Sequentially write 0x46h then 0xB9h to SCMD register to trigger CRC calculation.

Figure 26–1. CRC16 structure

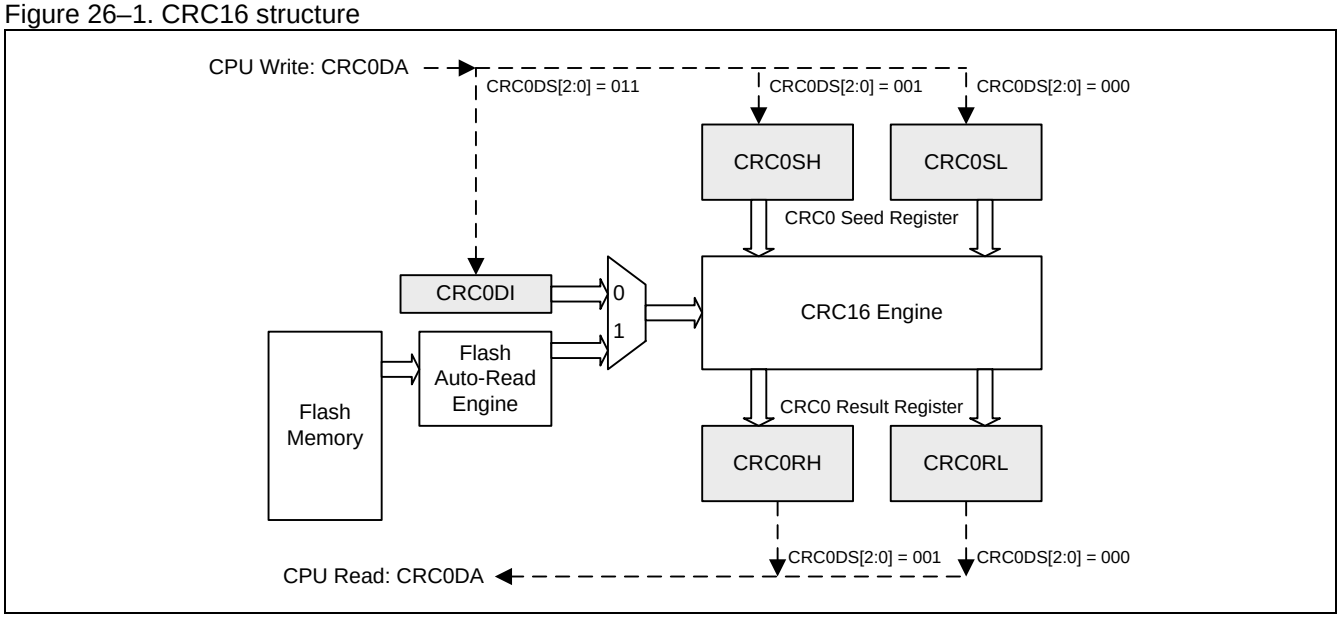
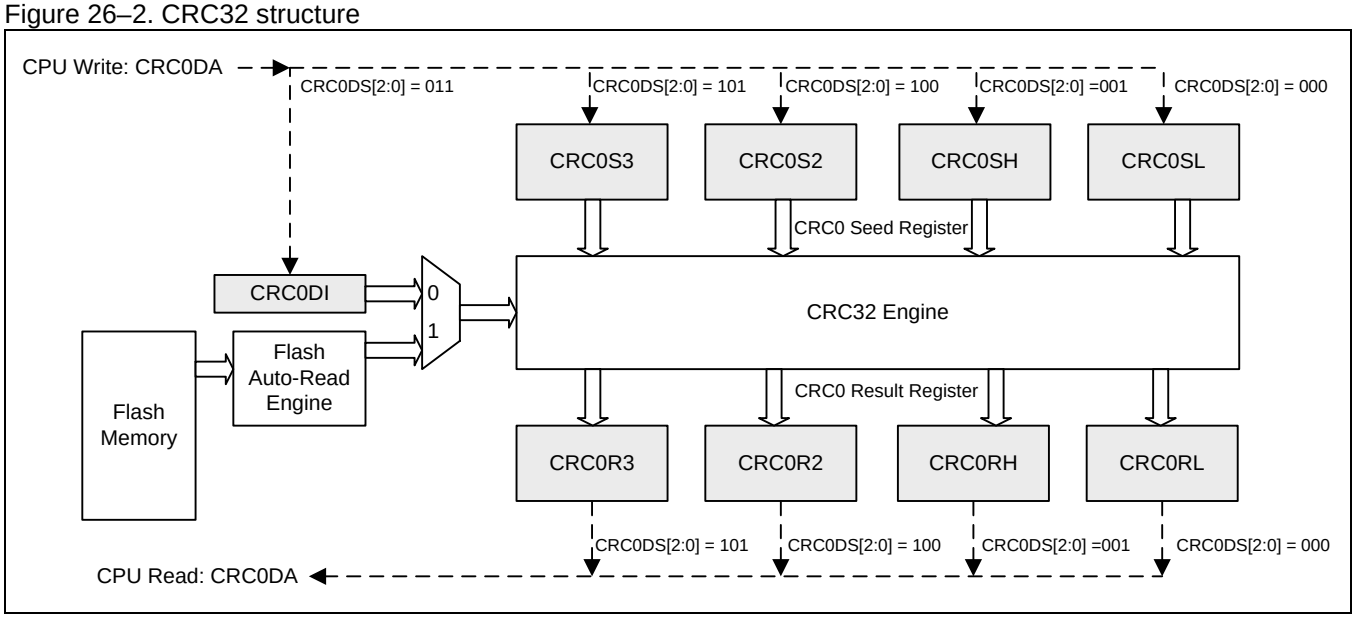
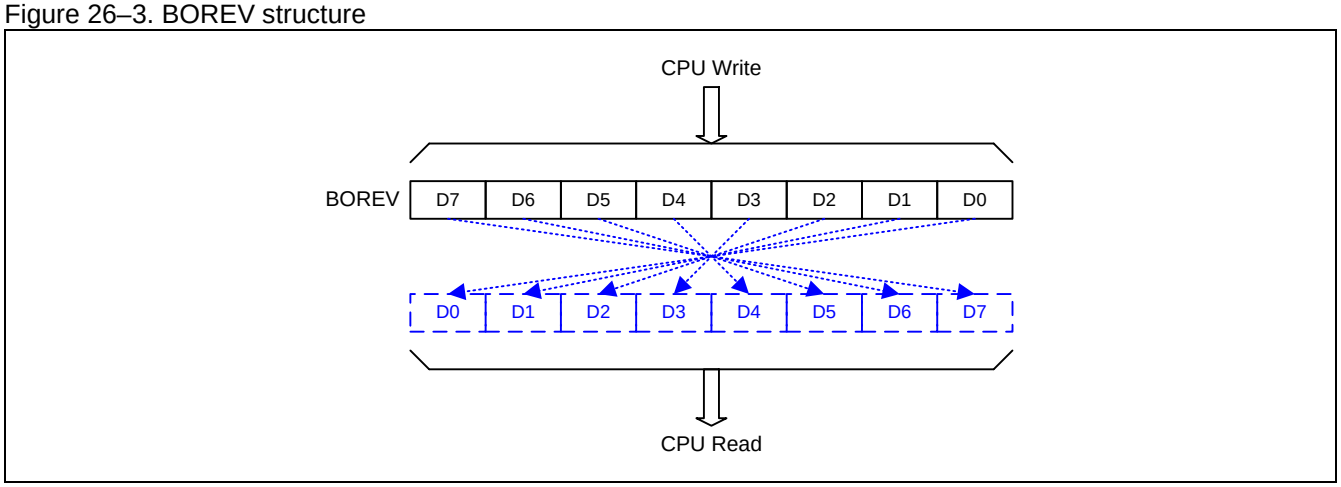


Figure 26–2. CRC32 structure



26.2. GPL-BOREV Structure

Figure 26–3. BOREV structure



26.3. GPL-EDC45 Structure

When using USB PD to do the communication, it is 4bit/ 5bit format with BMC (Bi-phase Mark Coding) signaling. To transfer 4-bit data to 5-bit data for the transmission to enhance correctness.

Table 26–1. 4b/5b transfer table

4BIT/ 5BIT table		
HEX data	**4 Bit**	**5 Bit**
0	0000	11110
1	0001	01001
2	0010	10100
3	0011	10101
4	0100	01010
5	0101	01011
6	0110	01110
7	0111	01111
8	1000	10010
9	1001	10011
A	1010	10110
B	1011	10111
C	1100	11010
D	1101	11011

E	1110	11100
F	1111	11101

To do the USB PD communication, the data need to use SFR EDC45 of S0 to transfer or receive, which is used for 4b/5b data formant transfer. The EDC45 is sharing the same address of SADDR. To use GPLC0 set the SFR to SADDR or EDC45. When GPLC0 = 0, this SFR will be SADDR, and GPLC0 = 1, it will be EDC45.

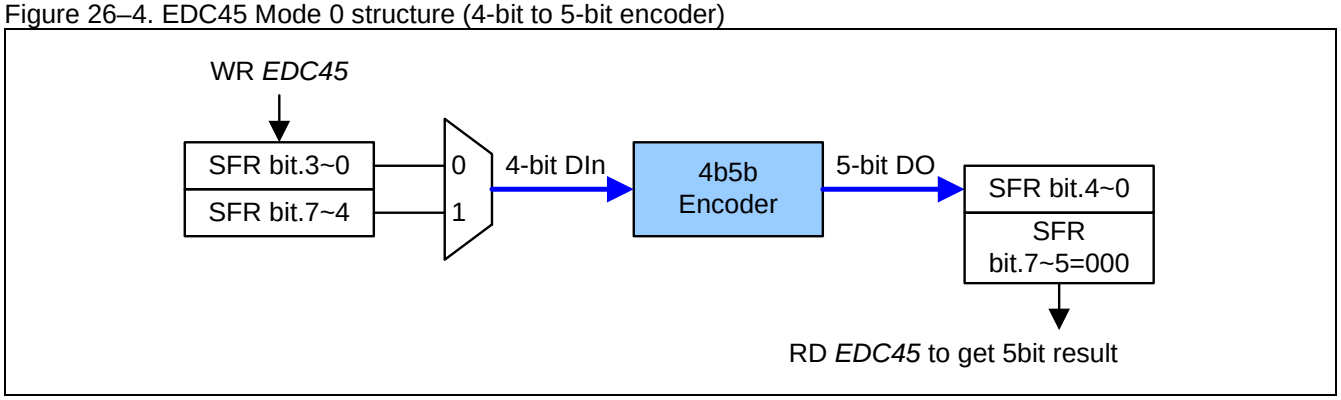
For USB PD data transfer, the GPLC0 should be set to 1.

When GPL-EDC45 process data transformation, it takes two 4b/5b data to finish the transfer or receive cycle.

Transfer Data (Data out):

1. Set GPLC0 = 1, to use the SFR as EDC45.
2. Set EDCM0 to 0, to set GPL-EDC45 operation as 4-bit to 5-bit encoder.
3. Write the data to SFR EDC45.
4. Read SFR EDC45 to S0BUF (To use S0 Mode 6 BMC ENDEC mode, reference section 20.6) to transfer low nibble data
5. Read SFR EDC45 again to S0BUF to transfer high nibble data

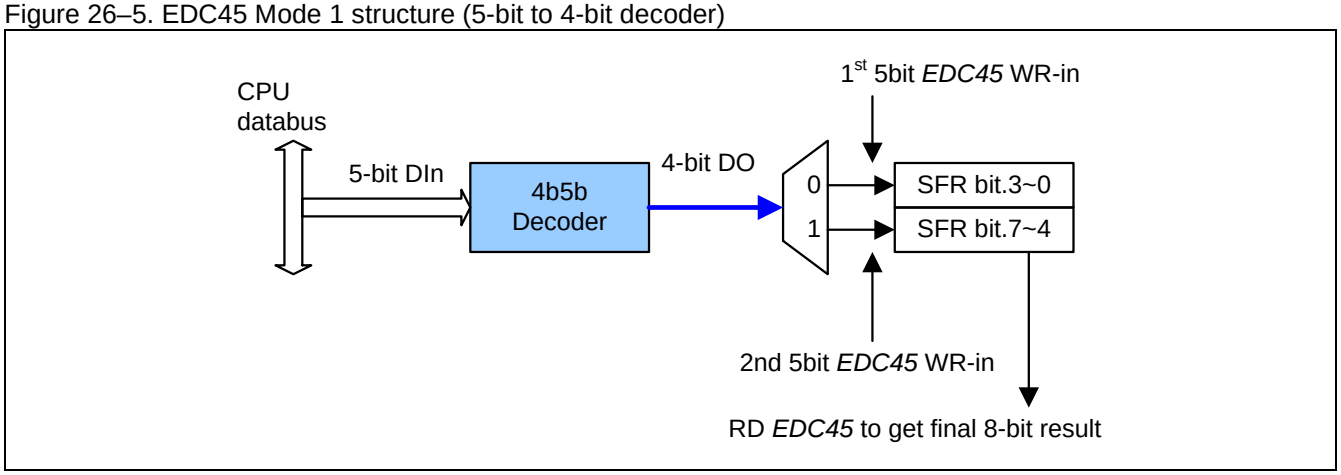
Figure 26–4. EDC45 Mode 0 structure (4-bit to 5-bit encoder)



Receive Data (Data In):

1. Set GPLC0 = 1, to use SFR EDC45.
2. Set EDCM0 to 1, to set GPL-EDC45 operation as 5-bit to 4-bit decoder.
3. Read first data from S0BUF (S0 set to Mode 6 BMC ENDEC mode, reference section 20.6) to SFR EDC45.
4. Read second data from S0BUF to SFR EDC45.
5. Read SFR EDC45 to get the input data.

Figure 26–5. EDC45 Mode 1 structure (5-bit to 4-bit decoder)



26.4. GPL Register

The following special function registers are related to the CRC operation:

CRC0DA: CRC0 Data Port

SFR Page = 0~F
SFR Address = 0xB6 RESET = 0000-0000

7	6	5	4	3	2	1	0
CRC0DA.7	CRC0DA.6	CRC0DA.5	CRC0DA.4	CRC0DA.3	CRC0DA.2	CRC0DA.1	CRC0DA.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: CRC0 Data Port. The CRC0 data access is defined as following table:

CRCDS2~0	CPU R/W	CRC0 Data Selection	Description
0 0 0	Write	CRC0SL	CRC0 Data Seed register-L.
0 0 1	Write	CRC0SH	CRC0 Data Seed register-H.
0 1 0	Write	--	Reserved.
0 1 1	Write	CRC0DI	CRC0 Data Input register.
1 0 0	Write	CRC0S2	CRC0 Seed register Byte 2 for CRC32
1 0 1	Write	CRC0S3	CRC0 Seed register Byte 3 for CRC32
0 0 0	Read	CRC0RL	CRC0 Result register-L.
0 0 1	Read	CRC0RH	CRC0 Result register-H.
0 1 0	Read	CRC0R2	CRC0 Result register byte 2 for CRC32.
0 1 1	Read	CRC0R3	CRC0 Result register byte 3 for CRC32.
Others	Read/Write	--	Reserved

AUXR1: Auxiliary Control Register 1

SFR Page = 0~F
SFR Address = 0xA2 RESET = 0000-0000

7	6	5	4	3	2	1	0
OP1Fr	OP0Fr	CRCDS1	CRCDS0	0	AC1Fr	AC0Fr	DPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5~4: CRCDS1~0. CRC0 Data port Selection bit 1~0.

AUXR12: Auxiliary Register 12

SFR Page = 9 only
SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
CRCDS2	CRCM0	PDOES1	PDOES0	EDCM0	GPLC0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: CRCDS2. CRC0 Data port Selection bit 2.

Bit 6: CRCM0, CRC mode selection 0.
0: Select CRC16: 0x1021.
1: Select CRC32: 0x04C1_1DB7.

Bit 3: EDCM0. EDC45 mode selection 0.

EDCM0	Mode #	EDC45 operation
0	0	4-bit to 5-bit encoder
1	1	5-bit to 4-bit decoder

Bit 2: GPLC0. GPL Control 0.
0: CPU accesses SFR address 0xA9 on SADDR.
1: CPU accesses SFR address 0xA9 on EDC45.

BOREV: Bit Order Reversed data register

SFR Page = 0~F
SFR Address = 0x96 RESET = 0000-0000

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

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 in mode 0.
If 0x01 is written to BOREV, the data read back will be 0x80 in mode 0.

IFMT: ISP/IAP Flash Mode Table

SFR Page = 0~F
SFR Address = 0xE5 RESET = xxxx-x000

7	6	5	4	3	2	1	0
MS.7	MS.6	MS.5	MS.4	MS.3	MS.2	MS.1	MS.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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
0 0 0 0-0 0 0 0	Standby
0 0 0 0-0 0 0 1	Flash byte read of AP/IAP-memory
0 0 0 0-0 0 1 0	Flash byte program of AP/IAP-memory
0 0 0 0-0 0 1 1	Flash page erase of AP/IAP-memory
0 0 0 0-0 1 0 0	Page P SFR Write
0 0 0 0-0 1 0 1	Page P SFR Read
1 0 0 0-0 0 0 0	Automatic flash read for CRC.
1 0 0 0-0 0 0 1	Flash byte read with address increased function
1 0 0 0-0 0 1 0	Flash byte 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.

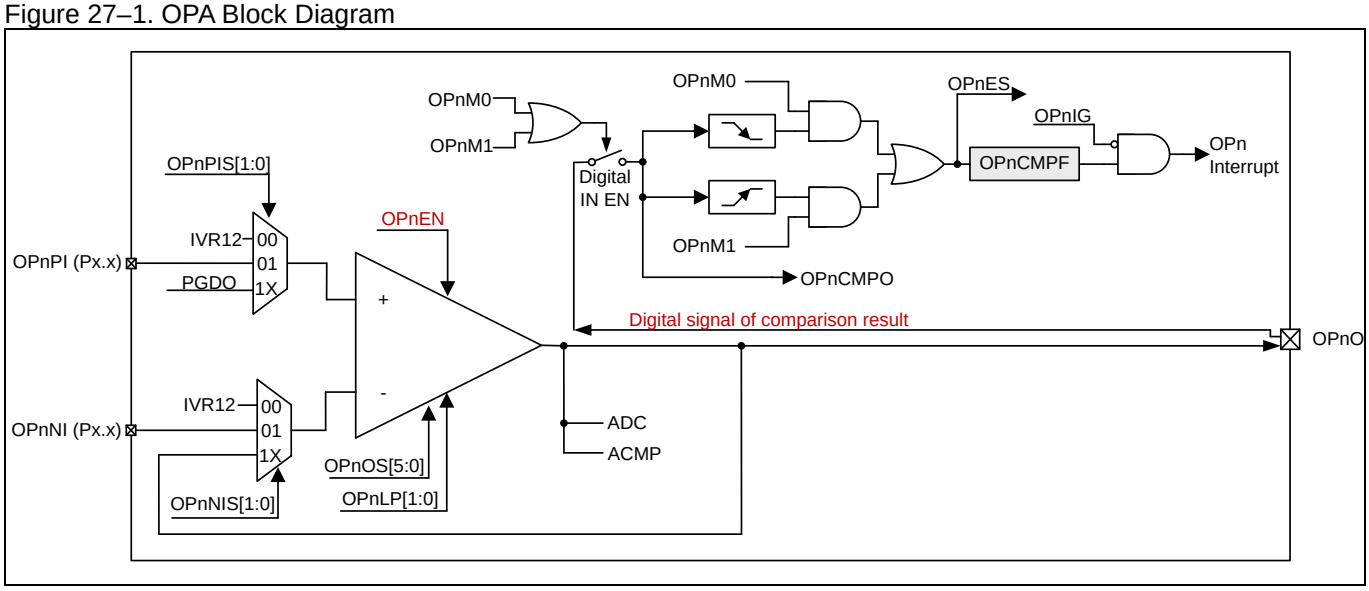
If software selects the mode on automatic flash read for CRC, the flash start-address is defined in IFARDH and IFADRL. The flash end-address is defined at {IAPLB + 9’b1-1111-1111}.

27. Operational Amplifiers (OPA0/ OPA1)

MG82F6P32 embeds **two** operational amplifiers, each operational amplifier has two inputs and one output. Each input terminal contains three signals for selection. These three I/Os can be connected to external pins and can be interconnected with any type of device. The OPA also supports 4 power levels and comparison mode. The output can be connected to internal ADC and ACx.

27.1. OPA Structure

Figure 27–1. OPA Block Diagram



Input pins:
OPA0 Positive Input: P1.7
OPA0 Negative Input: P2.2
OPA0 Output: P3.3
OPA1 Positive Input: P2.4
OPA1 Negative Input: P2.6
OPA0 Output: P3.0

27.2. OPA Operation

Each OPA can be enabled or disable individually, and both OPA has two power modes, normal power and low power mode. And it has one power level control bit OPPWR for OPA0 and OPA1. The OPPWR and OPnLP give total 4 power levels can be used.

OPA modules got 2 operation modes, one is OPA mode another is comparison (CMP) mode. When under CMP mode it can be act as a comparison and the compare result will be set in OPnCMPo. It can be used to trigger interrupt, and it can be a wakeup source to wake up the system when power down.

The OPA modules support input offset calibration to reduce input offset. It has factory trimmed value will be load within MCU initial period. But the offset might be different base on different system conditions, for example, the VDD voltage, ambient temperature. User can set the OPA to trim mode to get the optimized offset value.

27.2.1. OPA Input Channels

Two analog multiplexers (AMUX) select the inputs of the OPA positive input and negative input. Positive input has three chooses which is controlled by **OPnPIS[1:0]** to select signal from internal Bandgap 1.2V reference voltage, I/O and PGADO. When connect with PGADO, which is from PGA direct output, can be cascaded as two stages system.

Negative input also has three chooses which is controlled by **OPnNIS[1:0]** to select signal from internal Bandgap 1.2V reference voltage, I/O and OPnO (OPA output) as a Uni-Gain buffer.

When using OPA0/ OPA1 under OPA mode and input or output signals are from I/O, these I/O should be set to Analog input mode. When the OPA is disabled, the output is high impedance. When OPA under CMP mode the output pin will auto trigger the I/O to enable digital input path.

27.2.2. OPA Operation Modes

The OPAn has serval operation modes show in Table 27–1. OPAn Mode Selection

Table 27–1. OPAn Mode Selection

OPnM1	OPnM0	Mode	Descriptions
0	0	OPA	Operational amplifiers
0	1	CMP M0	Comparison Mode Falling edge trigger
1	0	CMP M1	Comparison Mode Rising edge trigger
1	1	CMP M2	Comparison Mode Dual edge trigger

27.2.3. OPA Power Modes

MG82F6P32 OPA has 4 power levels. OPPWR is the main control SFR to control the power level of both OPA0/1 at the same time. And user can use the OP0LP and OP1LP to get lower power mode of the OPA0 and OPA1 separately.

Table 27–2. OPAn Power Levels Selection

OPPWR:OPnLP	OPAn Power Saving control
0 0	High power, high speed
0 1	Medium high power, medium high speed
1 0	Medium low power, medium low speed
1 1	Low power, low speed

27.2.4. OPA Mode

- To set the I/O mode to negative input (OPnNI), positive input (OPnPI) and Output (OPnO) to Analog Input mode
- To set OPnEN to enable the OPA
- Wait for the OPA stable time
- Start to use the OPA

27.2.5. Comparison Mode

The OPA support signal comparison mode, its response time is longer then the AC0 and AC1.
Following steps show how to set the OPA to comparison mode.

- To set the I/O mode to negative input (OPnNI), positive input (OPnPI) and Output (OPnO) will be auto enabling digital input path
- The CMP has 3 modes to select different edge of the comparison result, falling edge, rising edge and dual edges.
- To set the power mode of the OPA by OPPWR and OPnLP.
- If wants to use interrupt, set EOPn. OPnCMPF set by hardware and needs to be cleared by software.
- To set OPnEN to enable the OPA in signal comparison mode
- Wait for the OPA stable time
- Start to use the OPA

27.2.6. OPA input Offset Trimming

MG82F6P32 has factory trimmed to reduce the OPA input offset voltage under normal power mode. But the offset might be different depends on different environmental conditions, for example, VDD voltage, ambient temperature and power mode. If users want to trim the offset under specific condition, it can set the OPA into compare mode and give the NI and PI the same voltage to do the calibration to get the best input signal performance.

When enable CMP mode for trimming, the digital I/O input path of the OPnO will be automatically enabled. In this case, if user wants to trim the offset, it is needed to consider the loading effects on the output pin, because the trimming result will be showed on the output through the I/O pad to digital input. If the components on the I/O pad have strong drive or sink current, it might cause the incorrect result when reading from the I/O.

Please follow the steps as below to do the offset calibration:
Method 1: For system manufacture trim mode, can use the external voltage supply to PI and NI of the OPA:

- To set the I/O mode of negative input (OPnNI), positive input (OPnPI) to analog input. Output (OPnO) will be auto enable digital input path
- Connect OPnNI and OPnPI together and apply DC voltage to these two PINs. The DC voltage which should be close to the target signal voltage level.
- To set OPnM[1:0] = 11 to compare mode for trimming.
- If wants to trim the offset of the low power mode, it should be set OPPWR and OPnLP. Please reference Table 27–2. OPAn Power Levels Selection
- To read the OP
- To set OPnEN to enable the OPA to start trimming mode
- Wait for the OPA stable time(<5uS)
- To read the OPnO status:
  1. If OPnO = 0, to increase OPnOS[5:0]; If OPnO = 1, to decrease OPnOS[5:0]
  2. Wait for OPnO stable(<600uS)
  3. Check OPnCMPF, if equal 1 it means the offset has been trimmed. The OPnCMPF needs to be cleared by software.

Method 2: User trim mode, needs to use the internal voltage supply to PI and NI of the OPA:

- To select Bandgap12 of positive input (OPnPI) and Output (OPnO) will be auto enable digital input path
- In this case the trimmed offset will be based on 1.2V as reference voltage. If the target signal RMS voltage is not at this level, it might have minor different for the optimized offset value.
- To set OPnM[1:0] = 11 to compare mode for trimming.
- If wants to trim the offset of the low power mode, it should be set OPPWR and OPnLP. Please reference Table 27–2. OPAn Power Levels Selection
- To set OPnEN to enable the OPA to start trimming mode
- Wait for the OPA stable time(<5uS)
- To read the the OPnO status:
  1. If OPnO = 0, to increase OPnOS[5:0]; If OPnO = 1, to decrease OPnOS[5:0]
  2. Wait for OPnO stable(<600uS)
  3. Check OPnCMPF, if equal 1 it means the offset has been trimmed. The OPnCMPF needs to be cleared by software.

27.2.7. Idle and Power-Down Mode

In Power-Down mode, the OPA can be enabled to use in CMP mode as the event trigger to wake up the system.

27.3. OPAn Register

OPSTA: OPA Status

SFR Page = 0 Only
SFR Address = 0xBA RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	0	0	0	0	OP1CMPF	OP0CMPF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 1: OP1CMPF, OPA1 compare result flag.
0: This bit must be cleared by software writing **"1"** on it. Software writing ":0" is no operation.
1: When OPA1 is under comparison or offset trimming mode, this bit is set by hardware when the comparison result come out.

Bit 0: OP0CMPF, OPA0 compare result flag.
0: This bit must be cleared by software writing **"1"** on it. Software writing ":0" is no operation.
1: When OPA0 is under comparison or offset trimming mode, this bit is set by hardware when the comparison result come out.

OP0CON0: OPA0 Control Register 0

SFR Page = 1 Only
SFR Address = 0xBA RESET = 0000-0000

7	6	5	4	3	2	1	0
OP0EN	OPPWR	OP0LP	OP0IG	OP0PIS1	OP0PIS0	OP0NIS1	OP0NIS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: OP0EN, OPA0 Enable.
0: Clear to turn off the OPA0 block.
1: Set to turn on the OPA0 block. At least **5us** stable time after OPA0 is enabled.

Bit 6: OPPWR OPA0/OPA1 Power level control. Used with OPnLP for different operating power level. This SFR to control OPA0 and OPA1 power mode at the same time.

Bit 5: OP0LP: OPA0 low power mode.

OPPWR:OP0LP	OPA0 Power Saving control
0 0	High power, high speed
0 1	Medium high power, medium high speed
1 0	Medium low power, medium low speed
1 1	Low power, low speed

Bit 4: OP0IG: OPA0 to ignore the comparison result of OP0CMPF to induce interrupt when OPA0 signal comparison mode has been enabled.

Bit 3 ~ 2: OP0PIS0[1:0], OPA0 positive input signal selection.

OP0PIS[1:0]	OPA0 positive input signal selection
0 0	Internal reference voltage Bandgap 1.2V (Need to enable IVR24. The voltage accuracy is not grantee.)
0 1	OP0PI (P1.7)
1 x	PGADO

Bit 1 ~ 0: OP0NIS[1:0], OPA0 negative input signal selection.

OP0NIS[1:0]	OPA0 negative input signal selection
0 0	Internal reference voltage Bandgap 1.2V (Need to enable IVR24. The voltage accuracy is not grantee.)
0 1	OP0NI (P2.2)
1 x	OP0O, Operate as Uni-gain buffer

OP1CON0: OPA1 Control Register 0

SFR Page = 2 Only
SFR Address = 0xBA RESET = 0000-0000

7	6	5	4	3	2	1	0
OP1EN		OP1LP	OP1IG	OP1PIS1	OP1PIS0	OP1NIS1	OP1NIS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: OP1EN, OPA1 Enable.
0: Clear to turn off the OPA1 block.
1: Set to turn on the OPA1 block. At least **5us** stable time after OPA1 is enabled.

Bit 5: OP1LP, OPA1 low power mode.

OPPWR:OP1LP	OPA1 Power Saving control
0 0	High power, high speed
0 1	Medium high power, medium high speed
1 0	Medium low power, medium low speed
1 1	Low power, low speed

Bit 4: OP1IG: OPA1 Interrupt ignore the comparison result of OP1CMPF to induce interrupt when OPA1 signal comparison mode has been enabled.

Bit 3 ~ 2: OP1PIS0[1:0], OPA1 positive input signal selection.

OP1PIS[1:0]	OPA1 positive input signal selection
0 0	Internal reference voltage Bandgap 1.2V (Need to enable IVR24. The voltage accuracy is not grantee.)
0 1	OP1PI (P2.4)
1 x	PGADO

Bit 1 ~ 0: OP1NIS[1:0], OPA1 negative input signal selection.

OP1NIS[1:0]	OPA1 negative input signal selection
0 0	Internal reference voltage Bandgap 1.2V (Need to enable IVR24. The voltage accuracy is not grantee.)
0 1	OP1NI (P2.6)
1 x	OP1O, Operate as Uni-gain buffer

OPCMP: OPA Comparison Mode

SFR Page = 7 Only
SFR Address = 0xBA RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	OP1PDX	OP0PDX	OP1M1	OP1M0	OP0M1	OP0M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7 ~ 6: Reserved. Software must write "**0**" on this bit when OPCMP is written.

Bit 5: OP1PDX: OPA1 control in PD mode.
0: Program the OPA1 to be gated off during PD mode.
1: Program the OPA1 to continue its function during PD mode.
If OP1EN, OP1PDX and EAC1 have been set, the OPA1 in PD function can only wake up CPU in low level or high-level mode.

Bit 4: OP0PDX: OPA0 control in PD mode.
0: Program the OPA0 to be gated off during PD mode.
1: Program the OPA0 to continue its function during PD mode.
If OP0EN, OP0PDX and EAC1 have been set, the OPA0 in PD function can only wake up CPU in low level or high-level mode.

Bit 3~2: OP1M[1:0], OPA1 mode selection:

OP0M1	OP0M0	Mode	Descriptions
0	0	OPA	Operational amplifiers
0	1	CMP M0	Falling edge trigger
1	0	CMP M1	Rising edge trigger
1	1	CMP M2	Dual edge trigger

Bit 1~0: OP0M[1:0], OPA0 mode selection:

OP0M1	OP0M0	Mode	Descriptions
0	0	OPA	Operational amplifiers
0	1	CMP M0	Falling edge trigger
1	0	CMP M1	Rising edge trigger
1	1	CMP M2	Dual edge trigger

OP0CFG0: OPA0 Configuration Register 0

SFR Page = P Only

SFR Address = 0x13 RESET = 00xx-xxxx

7	6	5	4	3	2	1	0
0	0	OP0OS5	OP0OS4	OP0OS3	OP0OS2	OP0OS1	OP0OS0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 5 ~ 0: OP0OS[5:0] Offset adjust value.

It has factory trimmed offset value will be loaded into MCU during initialization period. The factory trimmed value is trimmed under normal power mode. If customer wants to trim the offset value under customize environment, the OP0O should read to know if the offset has been trimmed.

- If OP0O = 0, to increase OP0OS[5:0]; If OP0O = 1, to decrease OP0OS[5:0]
- Wait for OP0O stable
- Check OP0CMPF, if it is toggled, it means the NI and PI voltage is the closest value that means the offset has been trimmed.

OPnOS[5:0]	OPA Offset adjust
111111	OPAn offset trim base +31
111110	OPAn offset trim base +30
……	……
100010	OPAn offset trim base +2
100001	OPAn offset trim base +1
100000	OPAn offset trim base
011111	OPAn offset trim base -1
011110	OPAn offset trim base -2
……	……
000001	OPAn offset trim base -31
000000	OPAn offset trim base -32

OP1CFG0: OPA1 Configuration Register 0

SFR Page = P Only

SFR Address = 0x14 RESET = 00xx-xxxx

7	6	5	4	3	2	1	0
0	0	OP1OS5	OP1OS4	OP1OS3	OP1OS2	OP1OS1	OP1OS0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 5 ~ 0: OP1OS[5:0] Offset adjust value.

It has factory trimmed offset value will be loaded into MCU during initialization period. The factory trimmed value is trimmed under normal power mode. If customer wants to trim the offset value under customize environment, the OP1O should read to know if the offset has been trimmed.

- If OP1O = 0, to increase OP1OS[5:0]; If OP1O = 1, to decrease OP1OS[5:0]
- Wait for OP1O stable
- Check OP1CMPF, if it is toggled, it means the NI and PI voltage is the closest value that means the offset has been trimmed.

OPnOS[5:0]	OPA Offset adjust

111111	OPAn offset trim base +31
111110	OPAn offset trim base +30
……	……
100010	OPAn offset trim base +2
100001	OPAn offset trim base +1
100000	OPAn offset trim base 0
011111	OPAn offset trim base -1
011110	OPAn offset trim base -2
……	……
000001	OPAn offset trim base -31
000000	OPAn offset trim base -32

AUXR1: Auxiliary Control Register 1

SFR Page = 0~F

SFR Address = 0xA2 RESET = 0000-0000

7	6	5	4	3	2	1	0
OP1Fr	OP0Fr	CRCDS1	CRCDS0	0	AC1Fr	AC0Fr	DPS
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: OP1Fr. If OP1CMPF is set to 1, then OP1Fr is also set to 1 according to hardware settings. It is read-only.

Bit 6: OP0Fr. If OP0CMPF is set to 1, then OP0Fr is also set to 1 according to hardware settings. It is read-only.

Bit 2: AC1Fr. If AC1F is set to 1, then AC1Fr is also set to 1 according to hardware settings. It is read-only.

Bit 1: AC0Fr. If AC0F is set to 1, then AC0Fr is also set to 1 according to hardware settings. It is read-only.

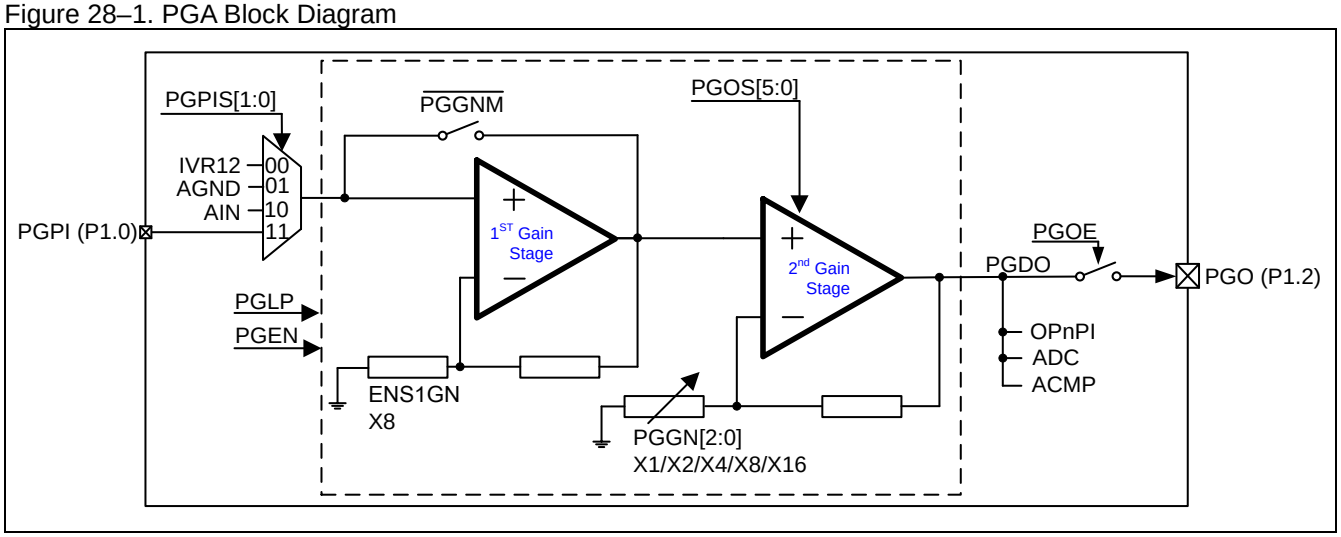
These 4 bit are in the same SFR, user can polling this SFR to get these 4 comparison result at the same time.

28. Programmable Gain Amplifier, PGA

MG82F6P32 embeds **one** PGA, it has positive input and one output. These two I/Os can be connected to external pins and can be interconnected with any type of device. The output can be controlled to the I/O pad or not. The PGA has two stages internal OPA with different gain can be used, x1, x2, x4, x8, x16, x32, x64 and x128. The output can be connected to internal ADC, ACx, OPA0/1.

28.1. PGA Structure

Figure 28–1. PGA Block Diagram



28.2. PGA Operation

The PGA can be enabled or disable by PGEN, and the PGA has two power modes, normal power and low power mode.

The PGA module supports input offset calibration to reduce input offset. It has factory trimmed value will be load within MCU bootup period. But the offset might be different base on different system conditions, for example, the VDD voltage, ambient temperature. User can set the PGA to trim mode to get the optimized offset value.

28.2.1. PGA Input Channels

Positive input has two chooses which is controlled by PGPIS[1:0] to select signal from internal 1.2V reference voltage, AGND, AIN and I/O.

Table 28–1. PGA Positive Input Selections

PGPIS1	PGPIS0	Mode	Descriptions
0	0	internal 1.2V reference voltage	Internal 1.2V reference voltage. (Need to enable IVR24. The voltage accuracy is not grantee.)
0	1	AGND	Use internal AGND for negative input mode.
1	0	AIN	From ADC AMUX Output – AIN, to use cascade mode with ADC as pre-amplifier.
1	1	PGPI PIN	I/O

28.2.2. PGA Operation

Embedded gain for Non-Inverting amplifier

- To set the I/O mode of positive input (PGPI) and Output (PGO) to Analog mode
- If only use the PGA as pre-amplifier for OPn, ADC or ACx, it is no need to set PGOE to the PGO pin. Otherwise, it should set the PGOE to assign output to PGO pin for external circuit.
- To set PGPIS0 = 1 to assign PGPI to positive input
- If the gain requirement is under 16 can use PGGNM = 0 to skip 1st gain stage.
- If the gain requirement is over 16, can set PGGNM = 1 to keep 1st gain stage. And set ENS1GN = 1 for x8 gain of the 1st gain stage. And then set the gain of the 2nd gain stage by PGGN[2:0], default gain are x1, x2, x4, x8, x16. The combined 1st and 2nd total maximum gain is up to 128.
- To set PGLP, if wants to use the PGA in low power mode
- To set PGEN to enable the PGA
- Wait for the PGA stable time
- Start to use the PGA

28.2.3. PGA input Offset Trimming Mode

MG82F6P32 has factory trimmed for the PGA input offset under normal power mode. But the offset might be different depends on different environment conditions, for example, VDD voltage, ambient temperature and power mode. User can ADC to trim the PGA offset for the calibration to get the best input signal performance.

MG82F6P32 PGA has two gain stages, but only the 2nd gain stage has the offset trimming mode. Each trimming step is around 0.6mV, and it has total 64 steps (+31 ~ -32). The offset of the 1st stage may use 2nd trimming method to do the calibration. For example, if 1st stage gain is x1, and its offset is +2mV, the offset trim value can be set around -3. If 1st stage gain is x8, then the 2nd input should “see” +16mV offset, the offset trim value can be set around -26. If the offset of the 1st stage is too large, user can use the ADC to record the offset value and to do the calibration in the software.

Please follow the steps as below to do the offset calibration:

User trim mode:

- To set AIN to IVR or 1/4VDD, and enable ADC
- To set ADMINS = 0 to set the ADC input direct from AIN.
- To start ADC conversion and keep the ADC result.
- To set ADMINS to 1 to set the ADC input source from PGA output.
- To set PGGNM = 0 to bypass the 1st gain stage, and to calibrate the offset of the 2nd gain stage.
- To start ADC conversion and compare the result with the value which direct measure by the ADC. To tune the trim code in opposite direction and then do the same routinely, until find the best solution.
- To change the PGGNM = 1 to enable 1st and use the same method of the 2nd gain stage to do the calibration.
- If wants to trim the offset of the low power mode, it should be set PGLP.

28.3. PGA Register

PGACON0: OPA0 Control Register 0

SFR Page = 5 Only

SFR Address = 0xBA RESET = 0000-0000

7	6	5	4	3	2	1	0
PGEN	0	PGLP	PGAOE	PGPIS1	PGPIS0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: PGEN, PGA Enable.
0: Clear to turn off the PGA block.
1: Set to turn on the PGA block. At least **1us** stable time after PGA is enabled.

Bit 5: PGLP, PGA low power mode.
0: PGA works under normal power mode.
1: PGA works under low power mode.

Bit 4: PGAOE, PGA output signal enable.
0: Disconnect from I/O
1: Connect to PGO pin

Bit 3 ~ 2: PGPIS[1:0], PGA positive input signal selection.

PGPIS[1:0]	PGA positive input signal selection
0 0	Internal 1.2V reference voltage
0 1	AGND
1 0	AIN (From ADC AMUX output)
1 1	OP0PI (P3.0)

Bit 1~0: Reserved. Software must write "**0**" on this bit when PGACON0 is written.

PGACON1: PGA Control Register 1

SFR Page = 6 Only

SFR Address = 0xBA RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	PGENS1GN	0	PGGNM	PGGN2	PGGN1	PGGN0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~6: Reserved. Software must write "**0**" on this bit when PGACON0 is written.

Bit 5: PGENS1GN, PGA first stage gain control.
0: Reserved.
1: PGA Gain x8

Bit 4: Reserved. Software must write "**0**" on this bit when PGACON0 is written.

Bit 3: PGGNM, PGA first stage bypass control.
0: Bypass PGA first stage, equivalent to set the first stage gain to x1
1: PGA first stage enable

Bit 2 ~ 0: PGGN[2:0], PGA 2nd gain stage gain control.

PGGN2	PGGN1	PGGN0	Gain
0	0	0	x1
0	0	1	x2
0	1	0	x4
0	1	1	x8
1	0	0	x16
1	0	1	reserved
1	1	0	reserved
1	1	1	reserved

PGACFG0: PGA Configuration Register 0

SFR Page = **P Only**

SFR Address = 0x15 RESET = 0001-1111

7	6	5	4	3	2	1	0
0	0	PGAOS5	PGAOS4	PGAOS3	PGAOS2	PGAOS1	PGAOS0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~6: Reserved. Software must write "**0**" on this bit when PGACFG0 is written.

Bit 5 ~ 0: PGAOS[5:0] trim value.

If has factory trimmed offset value will be loaded into MCU during initialization period. The factory trimmed value is trimmed under normal power mode. If customer wants to trim the offset value under customize environment, should read the PGA output pin PGO to get the current status.

- If PGO = 0, to increase PGAOS[5:0]; If PGO = 1, to decrease PGAOS[5:0]
- Wait for PGO stable
- Check PGO, if it toggles, means the NI and PI voltage is the closest value that means the offset has been trimmed.

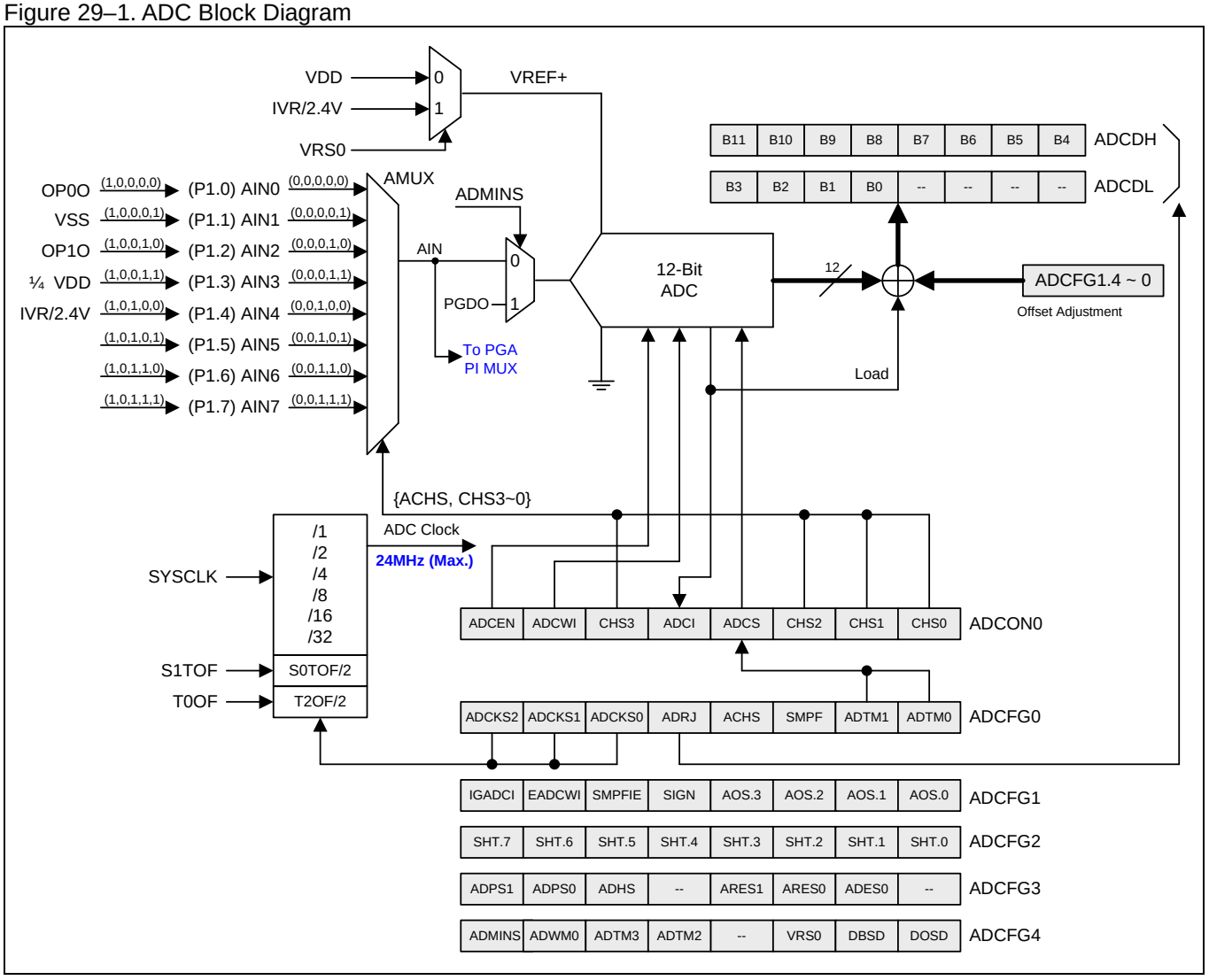
PGAOS[5:0]	PGA Offset adjust
111111	PGAn offset trim base +31
111110	PGAn offset trim base +30
……	……
100010	PGAn offset trim base +2
100001	PGAn offset trim base +1
100000	PGAn offset trim base 0
011111	PGAn offset trim base -1
011110	PGAn offset trim base -2
……	……
000001	PGAn offset trim base -31
000000	PGAn offset trim base -32

29. 12-Bit ADC

The ADC subsystem for the **MG82F6P32** consists of an analog multiplexer (AMUX) for input channels selection, and one AMUX for voltage reference input selection and a **1M** sps (room temperature)/ **500K** sps (Full temperature), **12**-bit successive-approximation-register ADC. The AMUX can be configured via the Special Function Registers shown in Figure 29–1. ADC operates in Single-ended mode and may be configured to measure any of the pins on AIN0 ~ AIN7 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.

29.1. ADC Structure

Figure 29–1. ADC Block Diagram



29.2. ADC Operation

ADC has a maximum conversion speed of **1M** sps (room temperature)/ **500K** sps (Full temperature). 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 24MHz.

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

$$\text{ADC Result} = \frac{V_{IN} \times 4096}{\text{VDD Voltage}}$$

29.2.1. ADC Reference voltage VREF+ selection

ADC has two reference voltage sources VDD and IVR 2.4V can be selected by VRS0.
When using IVR 2.4V, it will be easier to calculate the least signification bit (LSB) voltage. But if user want to get faster conversion rate, can choose VDD to supply higher voltage.

29.2.2. 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, 2.4V). 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 29–1. The selected pin is measured with respect to GND.

29.2.3. ADC Internal Voltage Reference

The default ADC reference is VDD. If the VDD is not fixed at a certain voltage, it should read a certain voltage to calculate the ADC value to know the voltage.

MG82F6P32 provides 3 internal reference voltage can be used to save external components and use less pin counts for other In/Out control. To use {ACHS, CHS[3:0]} to select:

1. **OP0O** @ {ACHS, CHS[3:0]} = {1, 0000}, to use OPA0 output as pre-amplifier.
2. **VSS** @ {ACHS, CHS[3:0]} = {1, 0001}, this can be used to do the Offset voltage cancelling.
3. **OP1O** @ {ACHS, CHS[3:0]} = {1, 0010}, to use OPA1 output as pre-amplifier.
4. **¼VDD** @ {ACHS, CHS[3:0]} = {1, 0011}, to detect VDD or battery voltage.
5. **IVR (2.4V)** @ {ACHS, CHS[3:0]} = {1, 0100}, this can be used as reference voltage

The following steps can be referenced to use internal voltage reference to calculate the step voltage (LSB voltage):

1) To set the analog multiplexer (AMUX) to Internal Voltage Reference.
2) Convert and store the reference voltage value by ADC.
3) To use the IVR read back reference value to calculate the VDD value to get LSB voltage.
4) To use the LSB voltage to calculate the input voltage.

29.2.4. Starting a Conversion

Prior to using the ADC function, the user should:

1) Choose VREF+ voltage by VRS0.
2) Turn on the ADC hardware by setting the ADCEN bit,
3) Configure the ADC input clock by bits ADCKS2, ADCKS1 and ADCKS0
4) Select the analog input channel by bits ACHS, CHS3, CHS2, CHS1 and CHS0
5) Configure the ADC voltage reference source
6) Configure the selected port input to the Analog-Input-Only mode
7) 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 overflow

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.

29.2.5. ADC Conversion Rate

The user can select the appropriate conversion speed according to the frequency of the analog input signal.
This ADC provide 2 transfer clock cycle options which can be used with different maximum ADC input clock speed.
It can be used ADHS (ADCFG3.5) to choose as following:

1. ADHS = 0: To use **30** ADC clocks for ADC conversion. The maximum input clock frequency should be under **29.47MHz**.
2. ADHS = 1: To use **24** ADC clocks for ADC conversion. The maximum input clock frequency should be under **24MHz**

User can configure the ADCKS2~0 (ADCFG0.7~5), SHT (ADCFG2.7~0) and ADHS (ADCFG3.5) to specify the conversion rate. The following equation is the clock number of one ADC conversion:

$$\text{ADC Conversion Rate} = \frac{\text{ADC Clock Freq.}}{(30 + X)} \quad ; X = \text{SHT}, 0\sim255$$

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 500K Sample Rate:
   If SYSCLK= 12MHz and the ADCKS = SYSCLK is selected, SHT = 0,
   Then conversion rate f_S = 12MHz/(24+0) = 500K sps.
   (In this case, the AC input signal f_N frequency should lower than 250KHz to ensure the measurement accuracy.)
2. To get 150K Sample Rate:
   If SYSCLK= 12MHz and the ADCKS = SYSCLK/2 is selected, SHT = 10,
   Then conversion rate f_S = 12MHz/2/(30+10) = 150K sps.
   (In this case, the AC input signal f_N frequency should lower than 75KHz to ensure the measurement accuracy.)

29.2.6. ADC Interrupts

The ADC interrupt of **MG82F6P32** 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)

Figure 29–2. ADC Interrupt

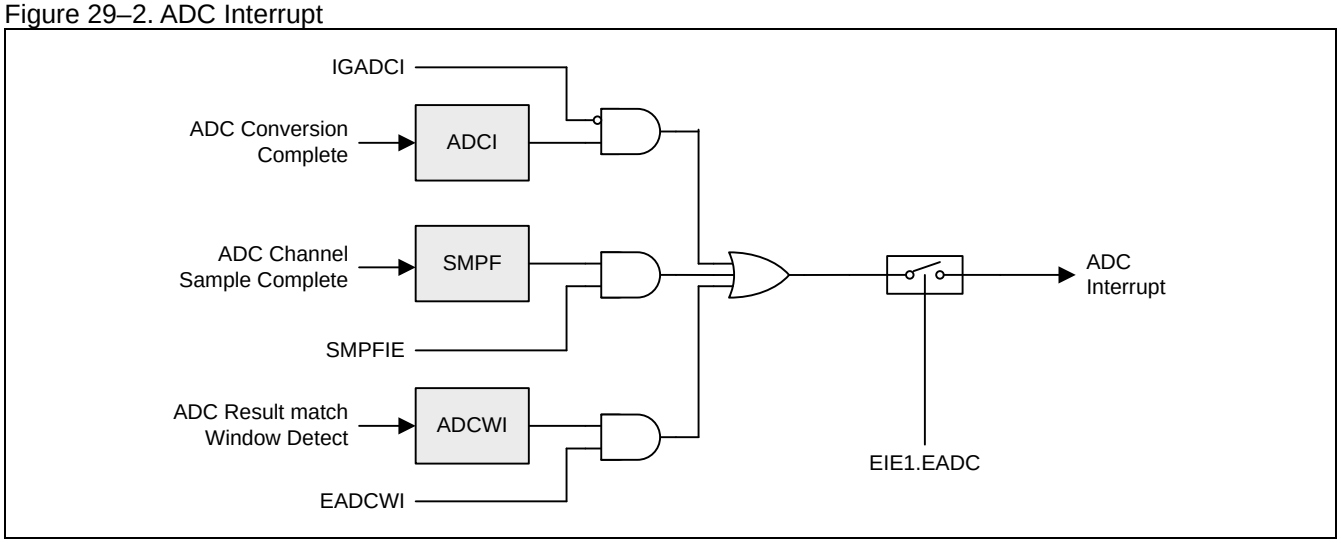
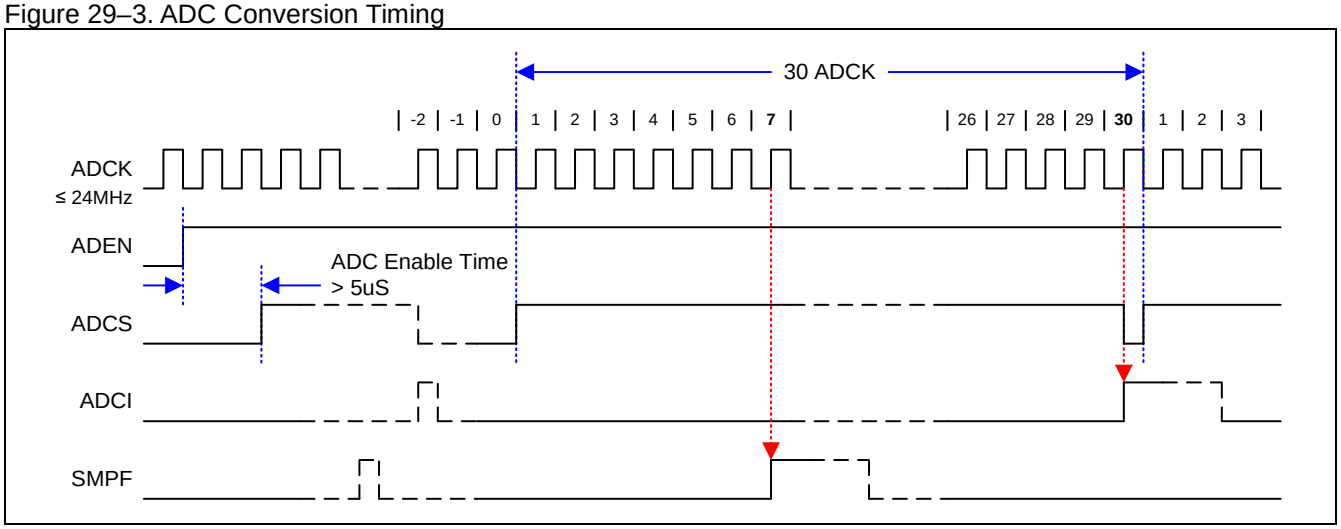


Figure 29–3. ADC Conversion Timing

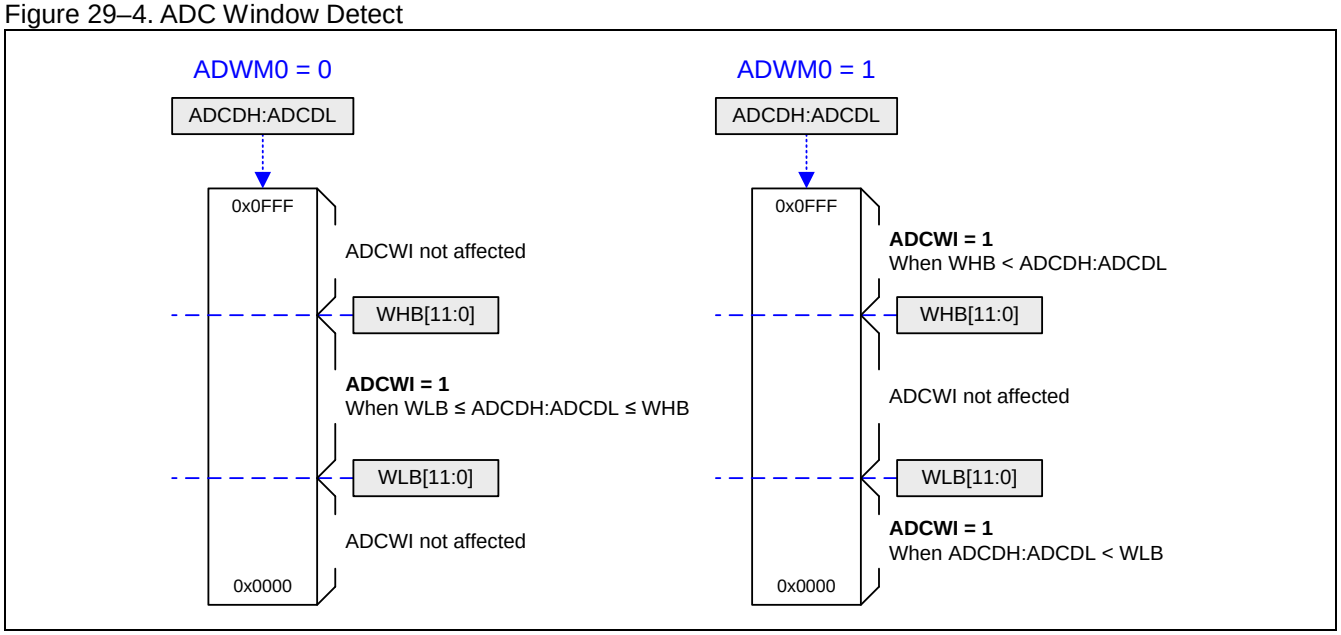


29.2.7. ADC Window Detect

The **MG82F6P32** 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:ADCDL) 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] ≤ ADCDH:ADCDL ≤ 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:ADCDL or ADCDH:ADCDL< WHB[11:0] is true, ADCWI will be held.

Figure 29–4. ADC Window Detect



Another application of ADC Window Detect is to specify the voltage is lager or less than a specific voltage. For example:

1. The target voltage ≥ the condition: ADWM0 = 0, to set condition value in WLB and set WHB = 0xFFF
2. The voltage less ≤ 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 = 0xFFF

29.2.8. ADC Channel Scan Mode

MG82F6P32 has 8 channels used as ADC input. If the application needs to watch several voltages 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.

When channel scan mode has been enabled, the channel information will be automatically asserted into the ADCDH or ADCDL.

Table 29–1. Add ADC channel information when channel Scan has enabled

ADRJ	ARES[1:0] ADC Data Resolution	ADCDH	ADCDL
0	**00** (12-bit)	0xFF	0xF + ADC CH Info
	01 (10-bit)	0xFF	0xC + ADC CH Info
	1x (8-bit)	0xFF	0x0 + ADC CH Info
1	**00** (12-bit)	0x ADC CH Info + F	0xFF
	01 (10-bit)	0x ADC CH Info + 3	0xFF
	1x (8-bit)	0x ADC CH Info + 0	0xFF

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 port input 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]

29.2.9. 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 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 0xFFF, with different combinations the ADC Data will be transfer by DMA as following:

Table 29–2. ADC DMA data transfer order without Channel Scan

ADRJ	ARES[1:0] ADC Data Resolution	DBSD ADC Data Byte Transfer by DMA	Data Transfer Order	
			1st Data	2nd Data
0	**00** (12-bit)	**0** (2-byte Data)	0xF0	0xFF
		1 (1-byte Data)	0xFF	X
	01 (10-bit)	**0** (2-byte Data)	0xC0	0xFF
		1 (1-byte Data)	0xFF	X
	1x (8-bit)	**0** (2-byte Data)	0x00	0xFF
		1 (1-byte Data)	0xFF	X
1	**00** (12-bit)	**0** (2-byte Data)	0xFF	0x 0F
		1 (1-byte Data)	0xFF	X
	01 (10-bit)	**0** (2-byte Data)	0xFF	0x 03
		1 (1-byte Data)	0xFF	X
	1x (8-bit)	**0** (2-byte Data)	0xFF	0x 00
		1 (1-byte Data)	0xFF	X

Table 29–3. ADC DMA data transfer order with Channel Scan

ADRJ	ARES[1:0] ADC Data Resolution	DBSD ADC Data Byte Transfer by DMA	Data Transfer Order	
			1st Data	2nd Data
0	**00** (12-bit)	**0** (2-byte Data)	0xF+ADC CH Info	0xFF
		1 (1-byte Data)	0xFF	X
	01 (10-bit)	**0** (2-byte Data)	0xC+ADC CH Info	0xFF
		1 (1-byte Data)	0xFF	X
	1x (8-bit)	**0** (2-byte Data)	0x0 + ADC CH Info	0xFF
		1 (1-byte Data)	0xFF	X
1	**00** (12-bit)	**0** (2-byte Data)	0xFF	0x ADC CH Info + F
		1 (1-byte Data)	0xFF	X
	01 (10-bit)	**0** (2-byte Data)	0xFF	0x ADC CH Info +3
		1 (1-byte Data)	0xFF	X
	1x (8-bit)	**0** (2-byte Data)	0xFF	0x ADC CH Info +0
		1 (1-byte Data)	0xFF	X

29.2.10. 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 analog-input-only mode to AIN7~0. The port pin configuration for

analog input function is described in "Table 14–3. General Port Configuration Settings" and regarding the AIN port pin setting please reference Section "14.2 I/O Port Register".

29.2.11. 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.

29.2.12. 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. **MG82F6P32** 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 Presorted 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)

$$IVR\ Voltage = \frac{IVR_{ADC\_PreStored\_Value} * 3300}{4096}(mV) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

- To use the proportional relationship calculate the I/O pin voltage

$$AIN\ Voltage = \frac{IVR\ Voltage * AIN_{ADC\_Value}}{IVR_{ADC\_Value}}(mV) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2)$$

Note: To read the IVR ADC Presorted value please reference 31.3 How to read IVR (2.4V) ADC Pre-stored value.

29.3. ADC Register

ADCON0: ADC Control Register 0

SFR Page = **0~F**

SFR Address = 0xC4 RESET = 0000-0000

7	6	5	4	3	2	1	0
ADCEN	ADCWI	CHS3	ADCI	ADCS	CHS2	CHS1	CHS0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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 CHS2~0 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 ~ CHS0, Input Channel Selection for ADC analog multiplexer.

In Single-ended mode:

ACHS	CHS3~0	Selected Channel
0	0 0 0 0	AIN0 (P1.0)
0	0 0 0 1	AIN1 (P1.1)
0	0 0 1 0	AIN2 (P1.2)
0	0 0 1 1	AIN3 (P1.3)
0	0 1 0 0	AIN4 (P1.4)
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	OP0O
1	0 0 0 1	AVSS
1	0 0 1 0	OP1O
1	0 0 1 1	1/4 VDD
1	0 1 0 0	Int. VREF (IVR/2.4V)
	Others	Reserved

ADCFG0: ADC Configuration Register 0

SFR Page = **0 Only**

SFR Address = **0xC3** RESET = 0000-0000

7	6	5	4	3	2	1	0
ADCKS2	ADCKS1	ADCKS0	ADRJ	ACHS	SMPF	ADTM1	ADTM0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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[7:6].

If ADRJ = 0

ADCDH: ADC Date High Byte Register

SFR Page = 0~F

SFR Address = 0xC6 RESET = xxxx-xxxx

7	6	5	4	3	2	1	0
(B11)	(B10)	(B9)	(B8)	(B7)	(B6)	(B5)	(B4)
R	R	R	R	R	R	R	R

ADCDL: ADC Data Low Byte Register

SFR Page = 0~F

SFR Address = **0xC5** RESET = xxxx-xxxx

7	6	5	4	3	2	1	0
(B3)	(B2)	(B1)	(B0)	--	--	--	--
R	R	R	R	R	R	R	R

If ADRJ = 1

ADCDH

7	6	5	4	3	2	1	0
--	--	--	--	(B11)	(B10)	(B9)	(B8)
R	R	R	R	R	R	R	R

ADCDL

7	6	5	4	3	2	1	0
(B7)	(B6)	(B5)	(B4)	(B3)	(B2)	(B1)	(B0)
R	R	R	R	R	R	R	R

When in Single-ended Mode, conversion codes are represented as 12-bit unsigned integers. Inputs are measured from ‘0’ to VDD(VREF) x 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 x 4095/4096	0xFFF0	0x0FFF
VDD x 2048/4096	0x8000	0x0800
VDD x 1024/4096	0x4000	0x0400
VDD x 512/4096	0x2000	0x0200
VDD x 256/4096	0x1000	0x0100
VDD x 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]	ADC Conversion Start Selection
0 0	Set ADCS
0 1	Timer 0 overflow
1 0	Free running mode
1 1	S0 BRG overflow

ADCFG1: ADC Configuration Register 1
SFR Page = **1 Only**
SFR Address = 0xC3 RESET = 0000-0000

7	6	5	4	3	2	1	0
IGADCI	EADCWI	SMPFIE	SIGN	AOS.3	AOS.2	AOS.1	AOS.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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 store the value in **MG82F6P32** IAP zone to use it as a constant parameter for ADC offset cancellation. The following table lists the adjustment value for ADC transfer result.

{Sign, AOS.[3:0]}	Value in {ADCDH, ADCDL}
0_1111	ADC transfer value + 15
0_1110	ADC transfer value + 14
……	……
0_0010	ADC transfer value + 2
0_0001	ADC transfer value + 1
0_0000	ADC transfer value + 0
1_1111	ADC transfer value – 1
1_1110	ADC transfer value – 2
……	……
1_0001	ADC transfer value – 15
1_0000	ADC transfer value – 16

ADCFG2: ADC Configuration Register 2
SFR Page = **2 only**
SFR Address = **0xC3** RESET = 0000-0000

7	6	5	4	3	2	1	0
SHT.7	SHT.6	SHT.5	SHT.4	SHT.3	SHT.2	SHT.1	SHT.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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 RESET = 0110-0000

7	6	5	4	3	2	1	0
ADPS1	ADPS0	ADHS	ADHS1	ARES1	ARES0	ADES0	
R/W	**R/W**	**W**	**W**	**R/W**	**R/W**	**R/W**	**W**

Bit 7~6: ADPS1~0, ADC Trigger Mode selection bit 3~2.

ADPS[1:0]	ADC Power Saving control
0 0	High power, high speed
0 1	Medium high power, medium high speed (default)
1 0	Medium low power, medium low speed
1 1	Low power, low speed

Bit 5 ~ 4: ADHS and ADHS1, ADC High Speed selection.

ADHS1, ADHS	ADC High Speed selection
0 0	30 ADC clocks for ADC conversion
0 1	24 ADC clocks for ADC conversion
1 0	28 ADC clocks for ADC conversion
1 1	29 ADC clocks for ADC conversion

Bit 3~2: ARES1~0, ADC data Resolution selection bit 1~0.

ARES[1:0]	ADC Data Resolution Selection
0 0	12-bit Data
0 1	10-bit Data
1 0	8-bit Data
1 1	Reserved

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 4, 0: Reserved. Software must write “0” on these bits when ADCFG3 is written.

ADCFG4: ADC Configuration Register 4

SFR Page = **4 only**

SFR Address = 0xC3 RESET = 0000-0000

7	6	5	4	3	2	1	0
ADMINS	ADWM0	ADTM3	ADTM2	0	VRS0	DBSD	DOSD
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: ADMINS. ADC input source selection from AMUX or PGA output.
0: ADC input source is set to AMUX of AIN.
1: ADC input source is set to PGDO which is PGA output as pre-amplifier.

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]	ADC Conversion Start Selection	Source
0 0 0 0	Set ADCS	Software
0 0 0 1	Timer 0 overflow (T0OF)	Timer 0
0 0 1 0	Free running mode	ADC
0 0 1 1	S0 BRG overflow (S0TOF)	S0 BRG
0 1 0 0	KBIET	KBI
0 1 0 1	INT1ET	nINT1
0 1 1 0	INT2ET	nINT2
0 1 1 1	INT3ET	nINT3
1 0 0 0	T2EXES	Timer 2

1 0 0 1	AC0ES	AC0
1 0 1 0	T3EXES	Timer 3
1 0 1 1	OP0ES	OPA0
1 1 0 0	PTM0 Overflow (C0TOF)	PTM0 Counter
1 1 0 1	C0CMP5 (Note1)	TM0 CH5 Compare
1 1 1 0	C1CMP0 or C1CMP1 (Note1)	PCA1 CH0/CH1 Compare
1 1 1 1	OP1ES	OPA1

*Note1: C0CMPx: Reference*Figure 17–5 *&* Figure 17–8

Bit 3: Reserved. Software must write “0” on these bits when ADCFG4 is written.

Bit 2: VRS0, Voltage reference, VREF+ source selection.
0: VDD
1: IVR/ 2.4V

Bit 1: DBSD, ADC Data Bit transfer by DMA.

DBSD	ADC Data Byte Transfer by DMA
0	2 Bytes Data
1	1 Byte Data (ADCH only with ADRJ=0)

Bit 0: DOSD, ADC Data Order Selection by DMA.

DBSD	DOSD	ADC Data Order transfer by DMA
0	0	ADCDL first
0	1	ADCDH first
1	0/1	ADCDH only with ADRJ=0

ADCFG5: ADC Configuration Register 5

SFR Page = **5 only**
SFR Address = 0xC3 RESET = 0000-0000

7	6	5	4	3	2	1	0
ASCE.7	ASCE.6	ASCE.5	ASCE.4	ASCE.3	ASCE.2	ASCE.1	ASCE.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~0: AIN7~AIN0 auto-scan enabled.
0: Disable ADC channel auto-scan.
1: Enable ADC channel auto-scan.

ADCFG7: ADC Configuration Register 7

SFR Page = **7 only**
SFR Address = 0xC3 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	0	0	ASCS3	ASCS2	ASCS1	ASCS0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~4: Reserved. Software must write “0” on these bits when ADCFG7 is written.

Bit 3: ASCS3 auto-scan channel selection 3.
0: Use AIN3 for ADC auto-scan channel 3.
1: Use internal channel of ¼ VDD for ADC auto-scan channel 3.

Bit 2: ASCS2 auto-scan channel selection 2.
0: Use AIN2 for ADC auto-scan channel 2.
1: Use internal channel of OP1O for ADC auto-scan channel 2.

Bit 1: ASCS1 auto-scan channel selection 1.
0: Use AIN1 for ADC auto-scan channel 1.
1: Use internal channel of VSS for ADC auto-scan channel 1.

Bit 0: ASCS0 auto-scan channel selection 0.
0: Use AIN0 for ADC auto-scan channel 0.
1: Use internal channel of OP0O for ADC auto-scan channel 0.

ADCFG11: ADC Configuration Register 11

SFR Page = **B only**

SFR Address = 0xC3 RESET = 1111-1111

7	6	5	4	3	2	1	0
WHB.3	WHB.2	WHB.1	WHB.0	1	1	1	1
R/W	**R/W**	**R/W**	**R/W**	**W**	**W**	**W**	**W**

ADCFG12: ADC Configuration Register 12

SFR Page = **C only**

SFR Address = **0xC3** RESET = 1111-1111

7	6	5	4	3	2	1	0
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**

WHB.9~0: ADC Window High Boundary value.

ADCFG13: ADC Configuration Register 13

SFR Page = **D only**

SFR Address = **0xC3** RESET = 0000-0000

7	6	5	4	3	2	1	0
WLB.3	WLB.2	WLB.1	WLB.0	0	0	0	0
R/W	**R/W**	**R/W**	**R/W**	**W**	**W**	**W**	**W**

ADCFG14: ADC Configuration Register 14

SFR Page = **E only**

SFR Address = **0xC3** RESET = 0000-0000

7	6	5	4	3	2	1	0
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**

WLB.9~0: ADC Window Low Boundary value.

PCON3: Power Control Register 3

SFR Page = **P Only**

SFR Address = 0x45 POR = 0000-0000

7	6	5	4	3	2	1	0
IVREN	IVRPDE	0	SPWRE	0	0	0	0
R/W	**R/W**	**W**	**R/W**	**W**	**W**	**W**	**W**

Bit 7: IVREN, Internal Voltage Reference Enable.
0: Disable on-chip IVR (2.4V).
1: Enable on-chip IVR (2.4V).

Bit 6 IVRPDE, IVR can be used under power down.
0: Disable IVR keep awake at Power down mode.
1: Enable IVR keep awake at Power down mode

Bit 5: Reserved. Software must write “0” on these bits when PCON3 is written.

Bit 3~0: Reserved. Software must write “0” on these bits when PCON3 is written.

30. Analog Comparator 0/1 (AC0/AC1)

The **MG82F6P32** consist of two analog comparator modules. It is useful to transfer analog signal into digital information by comparing the voltage level between V_{IN}+ and V_{IN}-. The result of these modules can send to port pin of internal logics.

1. Input Signals on V_{IN}+:
   a. For the AC0: It has 4 I/O inputs which can be selected by AC0PIS [1:0].
   b. For the AC1: It has single I/O inputs.
2. There are 3 kinds of the reference voltage on V_{IN}- can be used:
   a. VDD voltage divider by R-ladder to make different voltage. The
   b.
   c. Table 30–1 & Table 30–2 give detail setting.
   d. From port pin AC0NI, AC1NI: If the application need a much precise voltage, then it can apply a precise voltage source into this I/O pin.
   e. IVR: Internal Voltage Reference, 2.4V
   f. The AC1 V_{IN}- can set to use ACNI0 as the reference voltage, which can set the three comparators under the same reference voltage for different input signals.
3. Combine clock filter, which can set 3 different sample rates to filer different noise, which can reduce the software de-bounced effort to improve whole system efficiency.
4. Interrupt Mode selection: The interrupt of AC0, AC1 can be triggered by raising edge, falling edge and dual edge.

30.1. AC0/AC1 Structure

Figure 30–1. Analog Comparator 0 Block Diagram

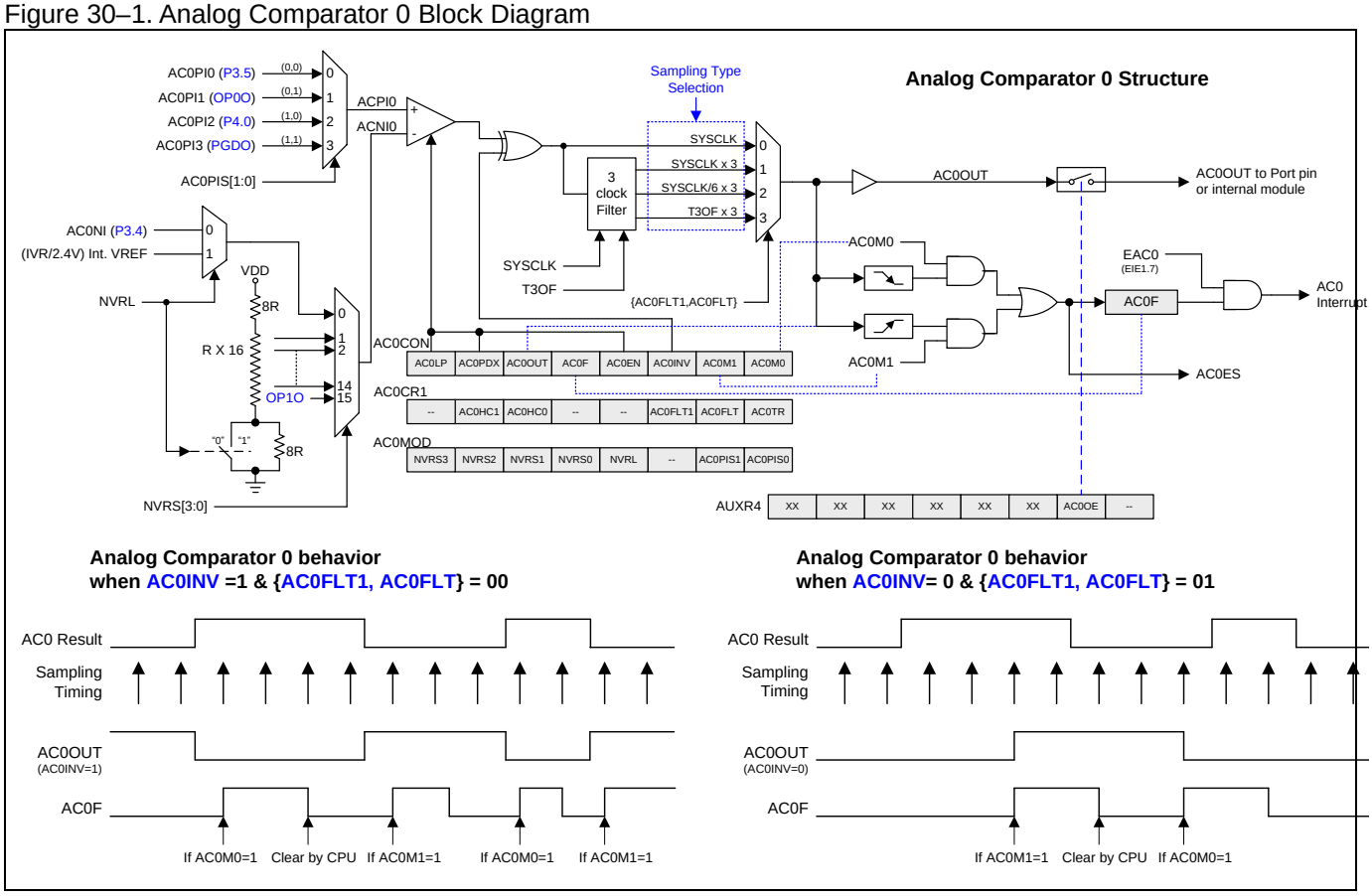
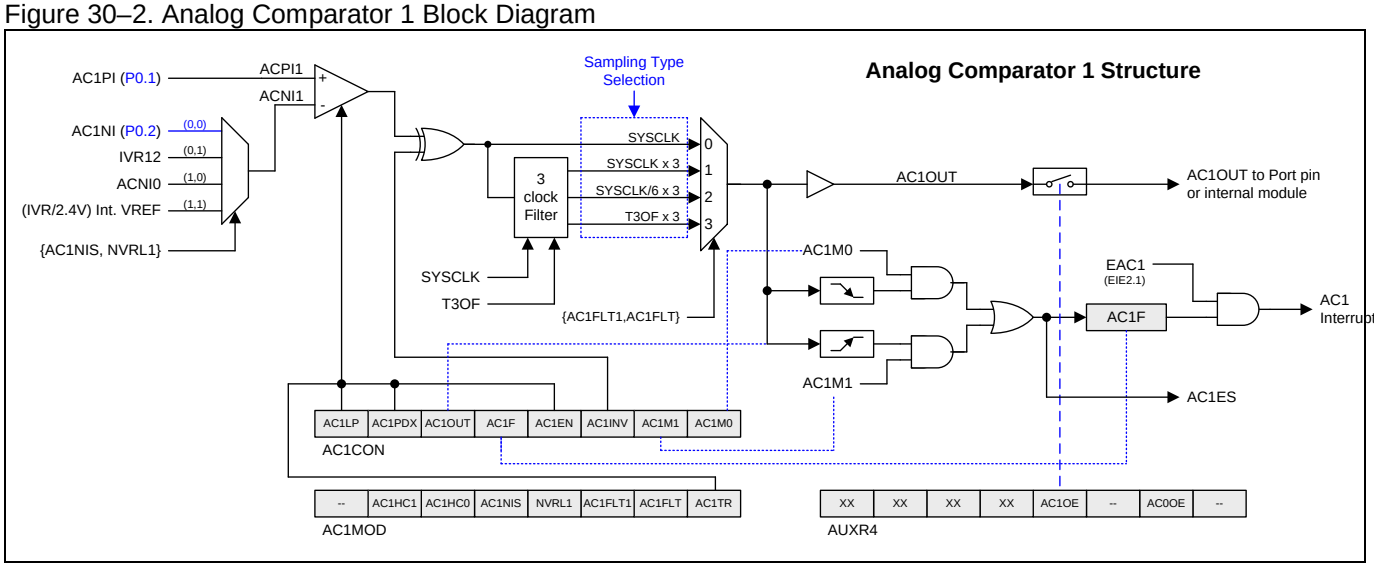


Figure 30–2. Analog Comparator 1 Block Diagram



30.2. AC0/AC1 Register

AC0CON: Analog Comparator 0 Control & Status Register

SFR Page = **0 Only**

SFR Address = 0x9E RESET = 00X0-0000

7	6	5	4	3	2	1	0
AC0LP	AC0PDX	AC0OUT	AC0F	AC0EN	AC0INV	AC0M1	AC0M0
R/W	**R/W**	**R**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: AC0LP, Analog Comparator 0 Low Power Enable.
0: Disable AC0 low power mode.
1: Enable AC0 low power mode.

Bit 6: AC0PDX, Analog Comparator 0 control in PD mode.
0: Program the Analog Comparator 0 to be gated off during PD mode.
1: Program the Analog Comparator 0 to continue its function during PD mode.
If AC0EN, AC0PDX and EAC0 have been set, the comparator in PD function can only wake up CPU in low level or high-level mode.

Bit 5: AC0OUT, this is a read only bit from comparator output.

AC0 Input	AC0INV = 0	AC0INV = 1
ACPI0(+) > ACNI0(-)	AC0OUT = 1	AC0OUT = 0
ACPI0(+) < ACNI0(-)	AC0OUT = 0	AC0OUT = 1

Bit 4: AC0F. Analog Comparator 0 Interrupt Flag.
0: The flag must be cleared by software.
1: Set when the comparator output meets the conditions specified by the AC0M [1:0] bits and AC0EN is set. The interrupt may be enabled/disabled by setting/clearing bit 7 of EIE1.

Bit 3: AC0EN. Analog Comparator 0 Enable.
0: Clearing this bit will force the comparator output low and prevent further events from setting AC0F.
1: Set this bit to enable the comparator.

Bit 2: AC0INV, Analog Comparator 0 output inversion bit.
0: AC0 output not inverted.
1: AC0 output inverted.

Bit 1~0: AC0M[1:0], Analog Comparator 0 Interrupt trigger edge selection.

AC0M[1:0]	AC0 Interrupt Mode
0 0	Reserved
0 1	Comparator 0 detects output Falling edge
1 0	Comparator 0 detects output Rising edge
1 1	Comparator 0 detects output Toggle

AC0MOD: Analog Comparator 0 Mode Register

SFR Page = **0 Only**

SFR Address = 0x9F RESET = 0000-0000

7	6	5	4	3	2	1	0
NVRS3	NVRS2	NVRS1	NVRS0	NVRL	0	AC0PIS1	AC0PIS0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~5: NVRS[3:0], Negative input on Voltage Reference selector of analog comparator 0. The four bits determine the analog comparator (V-) input source as following:

Table 30–1. AC0 Reference voltage selection, NVRL = 0, select high range

NVRS[3:0]	(V-) Input	NVRS[3:0]	(V-) Input
0000	**AC0NI(P3.4)**	1000	16/32 VDD
0001	9/32 VDD	1001	17/32 VDD
0010	10/32 VDD	1010	18/32 VDD
0011	11/32 VDD	1011	19/32 VDD
0100	12/32 VDD	1100	20/32 VDD
0101	13/32 VDD	1101	21/32 VDD
0110	14/32 VDD	1110	22/32 VDD
0111	15/32 VDD	1111	OP1O

Table 30–2. AC0 Reference voltage selection, NVRL = 1, select low range

NVRS[3:0]	(V-) Input	NVRS[3:0]	(V-) Input
0000	**Int. VREF (2.4V)**	1000	8/24 VDD
0001	1/24 VDD	1001	9/24 VDD
0010	2/24 VDD	1010	10/24 VDD
0011	3/24 VDD	1011	11/24 VDD
0100	4/24 VDD	1100	12/24 VDD
0101	5/24 VDD	1101	13/24 VDD
0110	6/24 VDD	1110	14/24 VDD
0111	7/24 VDD	1111	Reserved

Bit 3: NVRL, Negative Voltage Reference Low range select.
0: Select NVRS on high range.
1: Select NVRS on low range.

Bit 2: Reserved. Software must write “0” on these bits when AC0MOD is written.

Bit 1~0: AC0PIS[1:0], Positive input on I/O channel selector of analog comparator 0. The two bits determine the analog comparator (V+) input source as following:

AC0PIS[1:0]	(V+) Input Select
0 0	AC0PI0(P3.5)
0 1	AC0PI1(OP0O)
1 0	AC0PI2(P4.0)
1 1	AC0PI3(PGDO)

AC0CR1: Analog Comparator 0 Control & Status Register 1

SFR Page = **3 Only**

SFR Address = 0x9E RESET = 0000-0000

7	6	5	4	3	2	1	0
0	AC0HC1	AC0HC0	0	0	AC0FLT1	AC0FLT0	AC0TR
R/W	**R/W**	**R**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: Reserved. Software must write “0” on these bits when AC0CR1 is written.

Bit 6 ~ 5 : AC0HC[1:0], AC0 Hysteresis Control.

AC0HC[1:0]	Input Hysteresis select
0 0	**0**
0 1	**±10mV**
1 0	**±20mV**
1 1	**±60mV**

Bit 4~3: Reserved. Software must write “0” on these bits when AC0CR1 is written.

Bit 2 ~ 1: AC0FLT1, AC0FLT0, Analog Comparator 0 output Filter control. It selects AC0OUT filter mode

AC0FLT1, AC0FLT0	AC0OUT filter mode
0 0	Disabled
0 1	SYSCLK x 3
1 0	SYSCLK/6 x 3
1 1	**T3OF x 3**

Bit 0: AC0TR, AC0 Offset voltage trimming mode selection
0: Normal mode
1: Offset voltage trimming mode. Will short AC0 NI and PI automatically and trim the offset value to check the output toggle.

AC0CF0: AC0 Configuration Register 0

SFR Page = P Only

SFR Address = **0x14** RESET = 1001-1111

7	6	5	4	3	2	1	0
AC0OSE	AC0OS6	AC0OS5	AC0OS4	AC0OS3	AC0OS2	AC0OS1	AC0OS0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: AC0OSE, set AC0 to set offset mode.
0: Normal mode
1: Enable AC0 offset trimming mode. In this mode it will short NI and PI internally, and user can apply voltage on NI or PI or to use internal reference voltage for offset trimming.

Bit 6 ~ 0: AC0OS[6:0] Offset adjust value.
It has factory trimmed offset value will be loaded into MCU during initiation period. The factory trimmed value is trimmed under normal power mode. If customer wants trim the offset value under customize environment, the AC0OUT should read to know if the offset has been trimmed.

- If AC0OUT = 0, to increase AC0OS[6:0]; If AC0OUT = 1, to decrease AC0OS[6:0]
- Wait for AC0OUT stable
- Check AC0OUT, if it is toggled, it means the NI and PI voltage is the closest value that means the offset has been trimmed.

ACnOS[6:0]	ACn Offset adjust
1111111	ACn offset trim base +63
1111110	ACn offset trim base +62
……	……
1000010	ACn offset trim base +2
1000001	ACn offset trim base +1
1000000	ACn offset trim base
0111111	ACn offset trim base -1
0111110	ACn offset trim base -2
……	……

0000001	ACn offset trim base -63
0000000	ACn offset trim base -64

AC1CON: Analog Comparator 1 Control & Status Register

SFR Page = **1 Only**

SFR Address = 0x9E RESET = 0000-0000

7	6	5	4	3	2	1	0
AC1LP	AC1PDX	AC1OUT	AC1F	AC1EN	AC1INV	AC1M1	AC1M0
R/W	**R/W**	**R**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: AC1LP, Analog Comparator 1 Low Power Enable.
0: Disable AC1 low power mode.
1: Enable AC1 low power mode.

Bit 6: AC1PDX, Analog Comparator 1 control in PD mode.
0: Program the Analog Comparator 1 to be gated off during PD mode.
1: Program the Analog Comparator 1 to continue its function during PD mode.
If AC1EN, AC1PDX and EAC1 have been set, the comparator in PD function can only wake up CPU in low level or high-level mode.

Bit 5: AC1OUT, this is a read only bit from comparator output.

AC1 Input	AC1INV = 0	AC1INV = 1
ACPI1(+) > ACNI1(-)	AC1OUT = 1	AC1OUT = 0
ACPI1(+) < ACNI1(-)	AC1OUT = 0	AC1OUT = 1

Bit 4: AC1F. Analog Comparator 1 Interrupt Flag.
0: The flag must be cleared by software.
1: Set when the comparator output meets the conditions specified by the AC1M [1:0] bits and AC1EN is set. The interrupt may be enabled/disabled by setting/clearing bit 2 of EIE2.

Bit 3: AC1EN. Analog Comparator 1 Enable.
0: Clearing this bit will force the comparator output low and prevent further events from setting AC1F.
1: Set this bit to enable the comparator.

Bit 2: AC1INV, Analog Comparator 1 output inversion bit.
0: AC1 output not inverted.
1: AC1 output inverted.

Bit 1~0: AC1M[1:0], Analog Comparator 1 Interrupt Mode.

AC1M[1:0]	AC1 Interrupt Mode
0 0	Reserved.
0 1	Comparator 1 detects output Falling edge
1 0	Comparator 1 detects output Rising edge
1 1	Comparator 1 detects output Toggle

AC1MOD: Analog Comparator 1 Mode Register

SFR Page = **1 Only**

SFR Address = 0x9F RESET = 0000-0000

7	6	5	4	3	2	1	0
0	AC1HC1	AC1HC0	AC1NIS	NVRL1	AC1FLT1	AC1FLT0	AC1TR
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: Reserved. Software must write “0” on these bits when AC1MOD is written.

Bit 6 ~ 5 : AC1HC[1:0], AC1 Hysteresis Control.

AC1HC[1:0]	Input Hysteresis select
0 0	**0**
0 1	**±10mV**
1 0	**±20mV**
1 1	**±60mV**

Bit 4~3: AC1NIS & NVRL1, Analog Comparator 1 Negative Input Selection.

AC1NIS, NVRL1	ACNI1 Selection	Source Description
0 0	AC1NI port pin	P0.2
0 1	IVR12	Internal Voltage Reference 1.2V
1 0	ACNI0	Analog Comparator 0 Negative Input
1 1	IVR24	Internal Voltage Reference 2.4V

Bit 2 ~ 1: AC1FLT1, AC1FLT0, Analog Comparator 1 output Filter control. It selects AC1OUT filter mode

AC1FLT1, AC1FLT0	AC1OUT filter mode
0 0	Disabled
0 1	SYSCLK x 3
1 0	SYSCLK/6 x 3
1 1	T3OF x 3

Bit 0: AC1TR, AC1 Offset voltage trimming mode selection
0: Normal mode
1: Offset voltage trimming mode. Will short AC1 NI and PI automatically and trim the offset value to check the output toggle.

AC1CF0: AC1 Configuration Register 0

SFR Page = P Only
SFR Address = **0x15** RESET = 1001-1111

7	6	5	4	3	2	1	0
AC1OSE	AC1OS6	AC1OS5	AC1OS4	AC1OS3	AC1OS2	AC1OS1	AC1OS0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: AC1OSE, set AC1 to set offset mode.
0: Normal mode
1: Enable AC1 offset trimming mode. In this mode it will short NI and PI internally, and user can apply voltage on NI or PI or to use internal reference voltage for offset trimming.

Bit 6 ~ 0: AC1OS[6:0] Offset adjust value.
It has factory trimmed offset value will be loaded into MCU during initiation period. The factory trimmed value is trimmed under normal power mode. If customer wants to trim the offset value under customize environment, the AC1OUT should read to know if the offset has been trimmed.

- If AC1OUT = 0, to increase AC1OS[6:0]; If AC1OUT = 1, to decrease AC1OS[6:0]
- Wait for AC1OUT stable
- Check AC1OUT, if it is toggled, it means the NI and PI voltage is the closest value that means the offset has been trimmed.

ACnOS[6:0]	ACn Offset adjust
1111111	ACn offset trim base +63
1111110	ACn offset trim base +62
……	……
1000010	ACn offset trim base +2
1000001	ACn offset trim base +1
1000000	ACn offset trim base
0111111	ACn offset trim base -1
0111110	ACn offset trim base -2
……	……
0000001	ACn offset trim base -63
0000000	ACn offset trim base -64

AUXR4: Auxiliary Register 4

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
T2PS1	T2PS0	T1PS1	T1PS0	AC1OE	0	AC0OE	0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 3: AC1OE, AC1OUT output enable on port pin.
0: Disable AC1OUT output on port pin.
1: Enable AC1OUT output on P0.5.
Bit 1: AC0OE, AC0OUT output enable on port pin.
0: Disable AC0OUT output on port pin.
1: Enable AC0OUT output on P4.5.

PCON3: Power Control Register 3

SFR Page = **P Only**
SFR Address = 0x45 POR = 0000-0000

7	6	5	4	3	2	1	0
IVREN	IVRPDE	0	SPWRE	0	0	0	0
R/W	**R/W**	**W**	**R/W**	**W**	**W**	**W**	**W**

Bit 7: IVREN, Internal Voltage Reference Enable.
0: Disable on-chip IVR (2.4V).
1: Enable on-chip IVR (2.4V).

Bit 6 IVRPDE, IVR can be used under power down.
0: Disable IVR keep awake at Power down mode.
1: Enable IVR keep awake at Power down mode

Bit 5: Reserved. Software must write “0” on these bits when PCON3 is written.

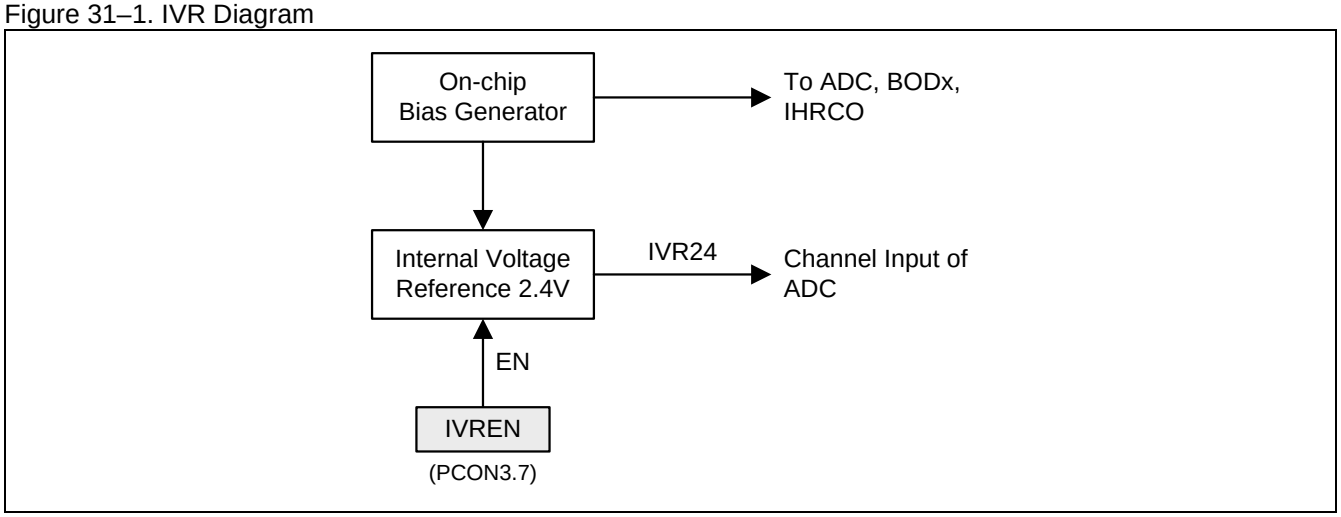
Bit 3~0: Reserved. Software must write “0” on these bits when PCON3 is written.

31. Internal Voltage Reference (IVR, 2.4V)

The IVR can be used as the reference voltage of the AC0 and ADC. The typical output is 2.4V. It can be disabling by IVREN.

31.1. IVR (2.4V) Structure

Figure 31–1. IVR Diagram



31.2. IVR Register

PCON3: Power Control Register 3

SFR Page = **P Only**

SFR Address = 0x45 POR = 0000-0000

7	6	5	4	3	2	1	0
IVREN	IVRPDE	0	SPWRE	0	0	0	0
R/W	**W**	**W**	**R/W**	**W**	**W**	**W**	**W**

Bit 7: IVREN, Internal Voltage Reference Enable.
0: Disable on-chip IVR (2.4V).
1: Enable on-chip IVR (2.4V).

Bit 6 IVRPDE, IVR can be used under power down.
0: Disable IVR keep awake at Power down mode.
1: Enable IVR keep awake at Power down mode

Bit 5: Reserved. Software must write “0” on these bits when PCON3 is written.

Bit 3~0: Reserved. Software must write “0” on these bits when PCON3 is written.

31.3. How to read IVR (2.4V) ADC Pre-stored 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 need 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 "29.2.12 How to improve ADC Accuracy" to understand how to improve ADC measurement accuracy.

```
void Get_Prestored_IVR(void)
{
        ISPCR = ISP_ENABLE;
        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;
}
```

32. ISP and IAP

The flash memory of **MG82F6P32** 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.

32.1. MG82F6P32 Flash Memory Configuration

There are total **32K** bytes of Flash Memory in **MG82F6P32** Figure 32–1 show the device flash configuration of **MG82F6P32**. 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 **32K** bytes flash memory.

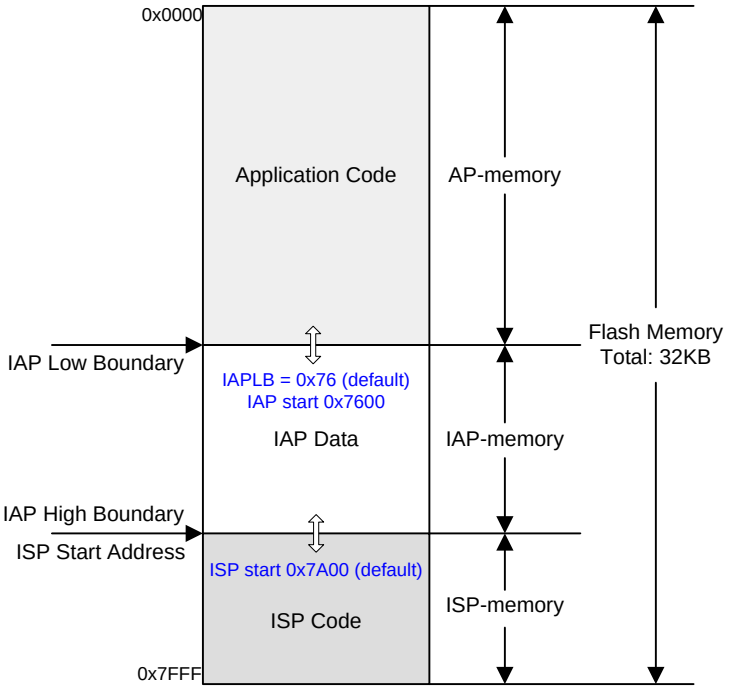
Figure 32–1. **6P32** Flash Memory Configuration

Note:
(1) ISP Start Address:
0x6200 if ISP Size = 7.5KB
0x6400 if ISP Size = 7KB
……
0x7000 if ISP Size = 4KB
……
0x7C00 if ISP Size = 1KB
0x7E00 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 = 0x7FFF
IAP Low Boundary = 0x7FFF – IAP Size + 1



Note:

*In default, the **MG82F6P32** 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.*

32.2. MG82F6P32 Flash Access in ISP/IAP

There are 3 flash access modes are provided in **MG82F6P32** 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 ISPEN in ISPCR to enable the ISP/IAP flow.
Step 2: Set MS= 0x03 in IFMT register to select Page Erase Mode.
Step 3: Fill page address in IFADRH & IFADRL registers.
Step 4: Sequentially write 0x46h then 0xB9h to SCMD register to trigger an ISP processing.
Step 5: Clear ISPEN= 0 and MS= 0x00 to close the ISP/IAP flow.

To do Byte Program

Step 1: Set ISPEN in ISPCR to enable the ISP/IAP flow.
Step 2: Set MS= 0x02 in IFMT register to select Byte Program Mode.
Step 3: Fill byte address in IFADRH & IFADRL registers.
Step 4: Fill data to be programmed in IFD register.
Step 5: Sequentially write 0x46h then 0xB9h to SCMD register to trigger an ISP processing.
Step 6: Clear ISPEN= 0 and MS= 0x00 to close the ISP/IAP flow.

To do Read

Step 1: Set ISPEN in ISPCR to enable the ISP/IAP flow.
Step 2: Set MS= 0x01 in IFMT register to select Read Mode.
Step 3: Fill byte address in IFADRH & IFADRL registers.
Step 4: Sequentially write 0x46h then 0xB9h to SCMD register to trigger an ISP processing.
Step 5: Now, the Flash data is in IFD register.
Step 6: Clear ISPEN= 0 and MS= 0x00 to close the ISP/IAP flow.

The detailed descriptions of flash page erase, byte program and flash read in **MG82F6P32** is listed in the following sections:

32.2.1. ISP/IAP Flash Page Erase Mode

The any bit in flash data of **MG82F6P32** 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 **MG82F6P32** 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 **MG82F6P32** 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 32–2 shows the flash page erase flow in ISPIAP operation.

Figure 32–2. ISP/IAP Page Erase Flow

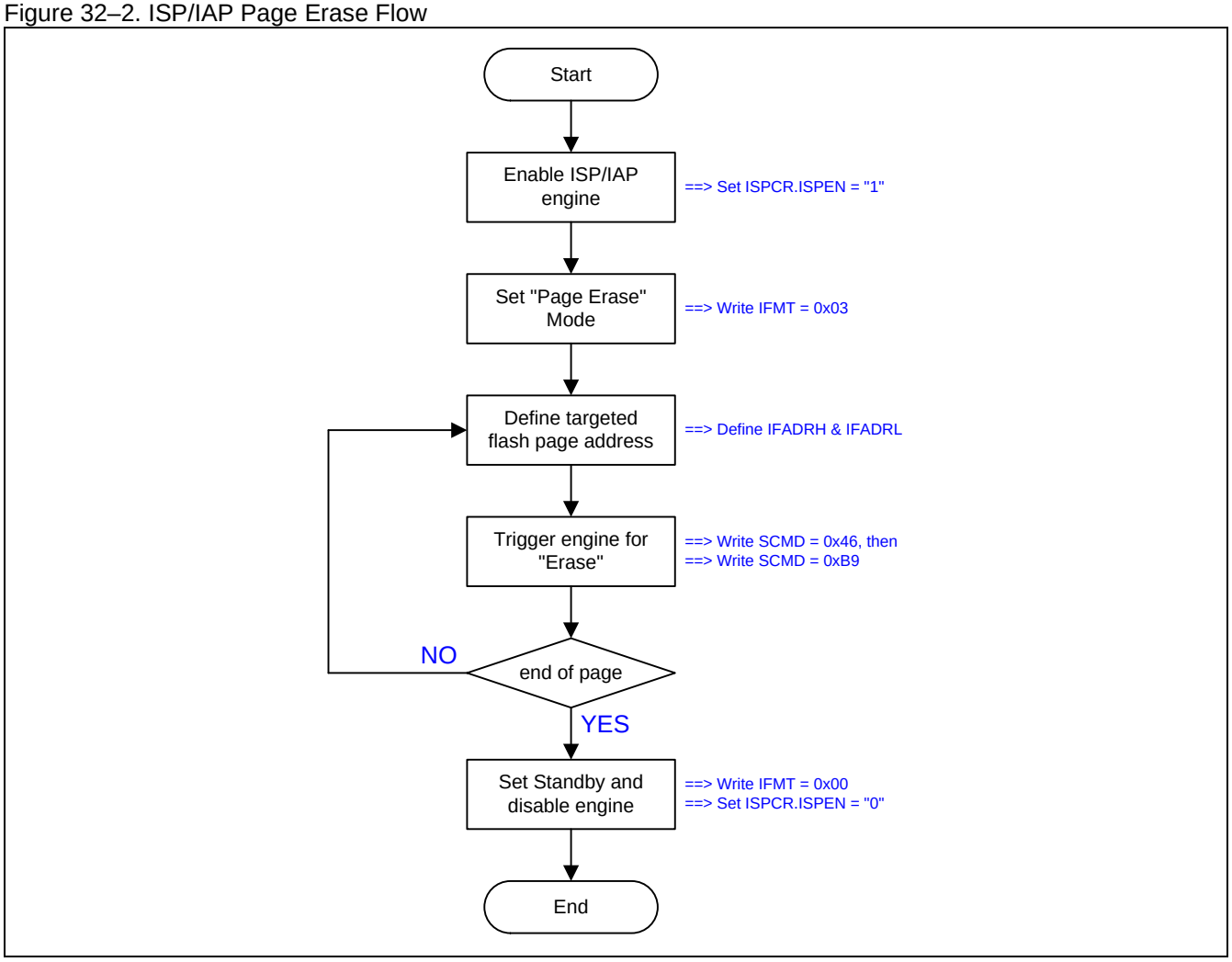


Figure 32–3 shows the demo code of the ISP/IAP page erase operation.

Figure 32–3. 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/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
```

32.2.2. ISP/IAP Flash Byte Program Mode

The “program” mode of **MG82F6P32** 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 32–4 shows the flash byte program flow in ISP/IAP operation.

Figure 32–4. ISP/IAP byte Program Flow

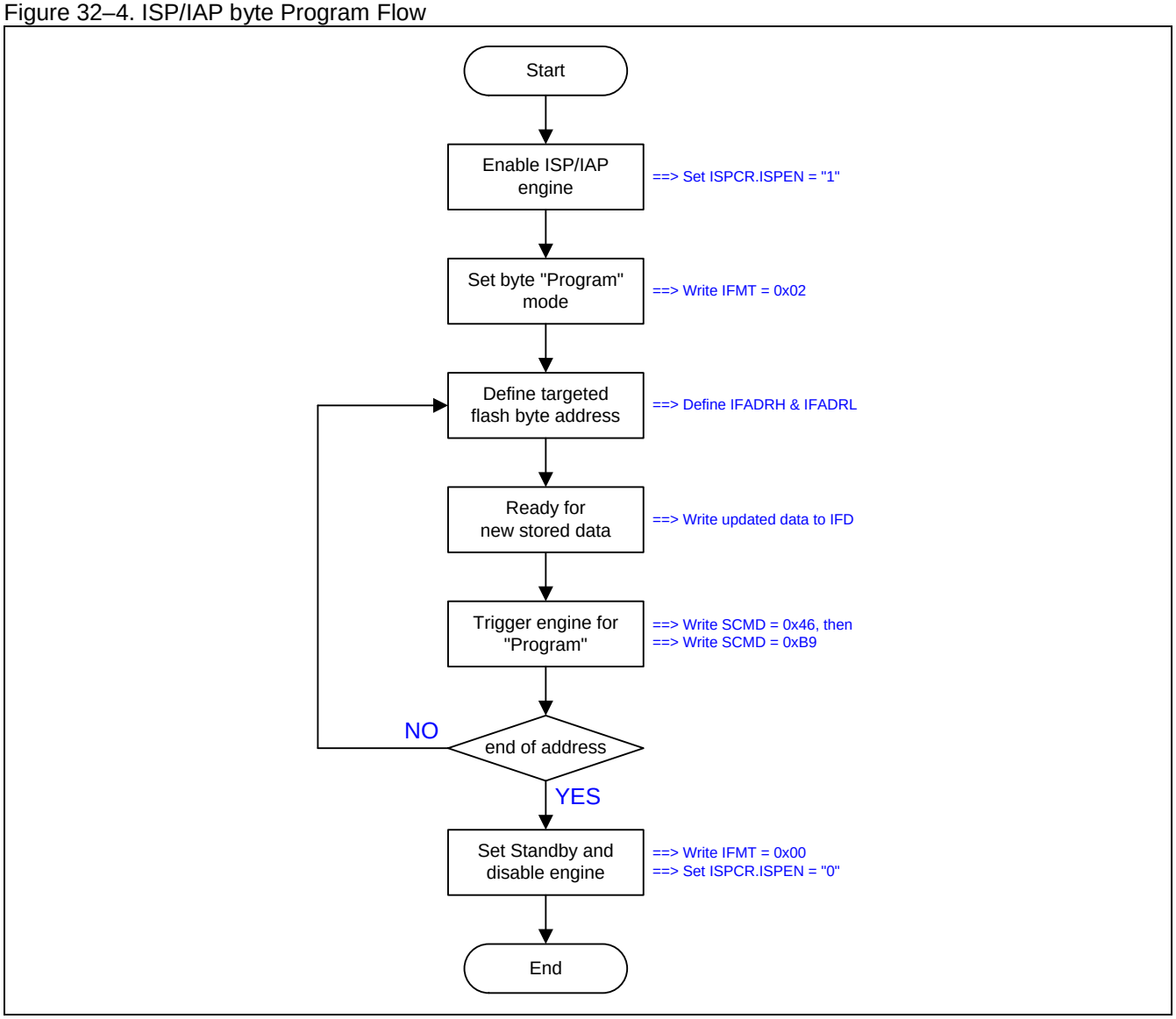


Figure 32–5 shows the demo code of the ISP/IAP byte program operation.

Figure 32–5. 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 address
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
```

32.2.3. ISP/IAP Flash Read Mode

The “read” mode of **MG82F6P32** 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 32–6 shows the flash byte read flow in ISP/IAP operation.

Figure 32–6. ISP/IAP byte Read Flow

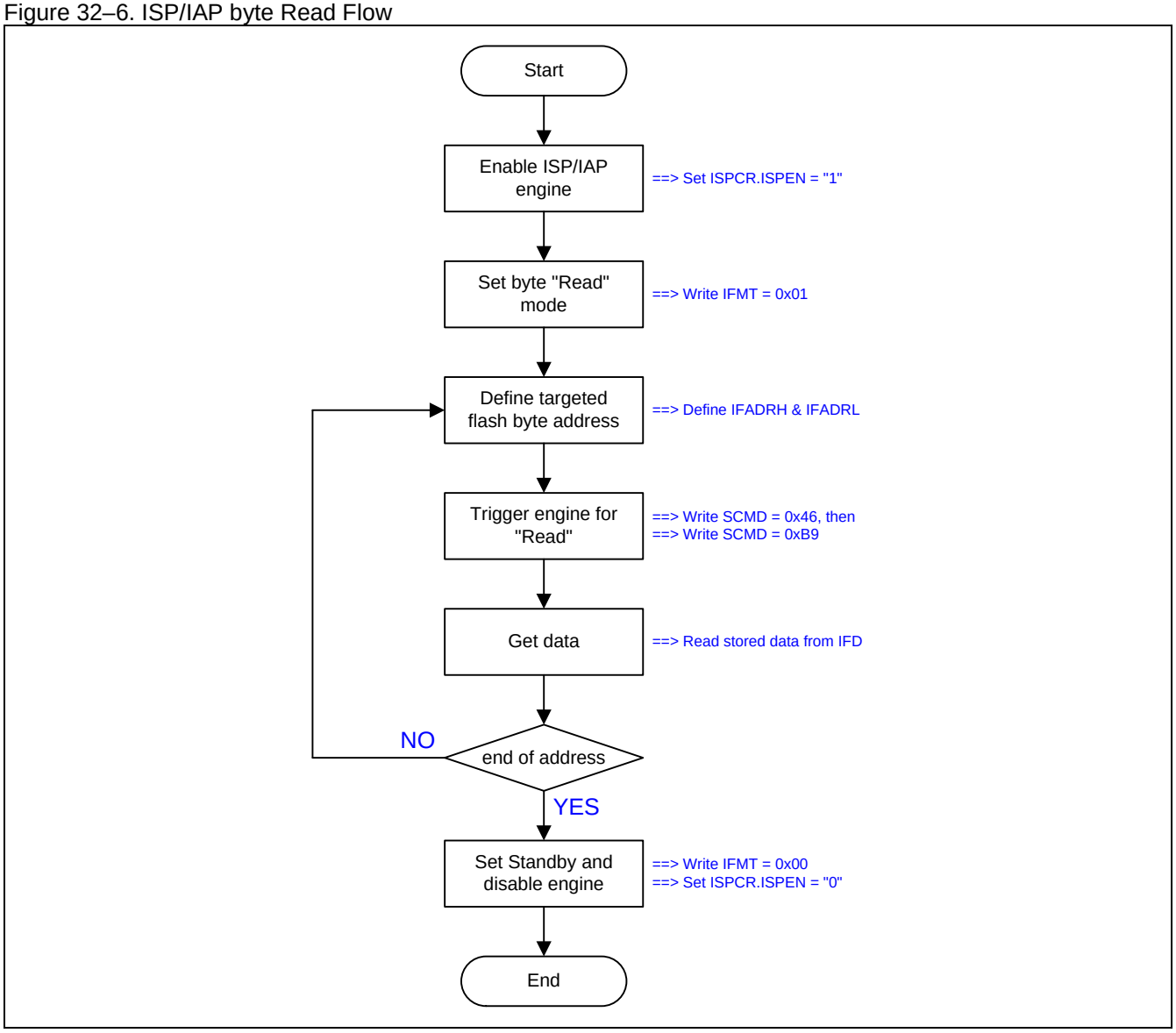


Figure 32–7 shows the demo code of the ISP/IAP byte read operation.

Figure 32–7. 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        ; trigger ISP/IAP processing
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, disable ISP
```

32.3. 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 a 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.

32.3.1. 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 has finished. It provides another hardware approached way to enter ISP mode by external reset signal. After power-on, **MG82F6P32** can perform ISP operation by external reset trigger and doesn't need to power down and then power-on again. It suits for the non-power-off system to apply the hardware approached ISP function. The SWBS will be held after HWBS2, and it should be cleared after reset from ISP code.

32.3.2. 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.

32.3.3. 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 32–1 for **MG82F6P32**, 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 cannot 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

MG82F6P32 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.

32.4. In-Application-Programming (IAP)

The **MG82F6P32** 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 **MG82F6P32** 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.*

32.4.1. IAP-memory Boundary/Range for MG82F6P32

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 = **0x7FFF**.*
*IAP low boundary = **0x7FFF** – IAP + 1.*

For example, if ISP-memory is 1K, so that ISP start address is **0x7C00**, and IAP-memory is 1K, then the IAP-memory range is located at **0x7800** ~ **0x7BFF**. The IAP low boundary in **MG82F6P32** is defined by IAPLB register which can be modified by software to adjust the IAP size in user's AP program.

32.4.2. Update data in IAP-memory

The special function registers are related to ISP/IAP would be shown in Section "32.5 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.

32.4.3. 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 cannot 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

MG82F6P32 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.

32.5. 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 RESET = 1111-1111

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

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 RESET = 0000-0000

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

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 RESET = 0000-0000

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

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 RESET = xxxx-x000

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

Bit 7~0: ISP/IAP/Page-P operating mode selection

MS[7:0]	Mode
0 0 0 0-0 0 0 0	Standby
0 0 0 0-0 0 0 1	Flash byte read of AP/IAP-memory
0 0 0 0-0 0 1 0	Flash byte program of AP/IAP-memory
0 0 0 0-0 0 1 1	Flash page erase of AP/IAP-memory
0 0 0 0-0 1 0 0	Page P SFR Write
0 0 0 0-0 1 0 1	Page P SFR Read
1 0 0 0-0 0 0 0	Automatic flash read for CRC.
1 0 0 0-0 0 0 1	Flash byte read with address increased function
1 0 0 0-0 0 1 0	Flash byte 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 RESET = xxxx-xxxx

7	6	5	4	3	2	1	0
SCMD							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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: ISP Control Register

SFR Page = **0~F**
SFR Address = 0xE7 POR = 0000-0000

7	6	5	4	3	2	1	0
ISPEN	SWBS	SWRST	CFAIL	--	--	--	--
R/W	R/W	R/W	R/W	W	W	W	W

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 RESET = 0111-000x

7	6	5	4	3	2	1	0
IAPLB							0
W	W	W	W	W	W	W	W

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 IMFT 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 IMFT, 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.

32.5.1. ISP/IAP Sample Code

The following Figure 32–8 shows a sample code for ISP operation.

Figure 32–8. Sample Code for ISP

```
;*************************************************************************************
; 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. Page Erase Mode (512 bytes per page)
;=========================================================================
       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 processing
       MOV    SCMD,#0B9h  ;
       ;Now in processing...(CPU will halt here until complete)

;=========================================================================
; 2. Byte Program Mode
;=========================================================================
       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   ;trigger ISP 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   ;trigger ISP 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:
       ...
;
```

33. Page P SFR Access

MG82F6P32 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 RESET = 1111-111x

7	6	5	4	3	2	1	0
IAPLB							0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 IFADRL for SFR address in Page-P, the IMFT 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 IMFT, 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 = IAPLBx256, 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 RESET = 0101-0000

7	6	5	4	3	2	1	0
XTGS1	XTGS0	XTALE	IHRCOE	MCKS1	MCKS0	OSCS1	OSCS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: XTGS1~XTGS0, Crystal Gain Selection register.

XTGS1, XTGS0	Gain Define	Applied Crystal
0, 0	Low Gain	32.768KHz
0, 1	Medium Gain	2MHz ~ 25MHz
1, 0	Lower Gain	32.768KHz
1, 1	Reserved	Reserved

Bit 5: XTALE, external Crystal(XTAL) Enable.
0: Disable XTAL oscillating circuit. In this case, XTAL2 and XTAL1 behave as Port 6.0 and Port 6.1.
1: Enable XTAL oscillating circuit. If this bit is set by CPU software, software polls the XTOR (CKCON1.7) true to indicate the crystal oscillator is ready for OSCin clock selected.

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 Source Selection	OSCin =12MHz CKMIS = [01]	OSCin =11.059MHz CKMIS = [01]
0 0	OSCin	12MHz	11.059MHz
0 1	CKMI x4	24MHz	22.118MHz
1 0	CKMI x5.33	32MHz	29.491MHz
1 1	CKMI x8	48MHz	44.236MHz

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 ≤ 32MHz, SYSCLK ≤ 50MHz.

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

OSCS[1:0]	OSCin source Selection
0 0	IHRCO
0 1	XTAL
1 0	ILRCO
1 1	ECKI, External Clock Input (P6.0) as OSCin.

CKCON3: Clock Control Register 3

SFR Page = **P only**

SFR Address = 0x41 RESET = 0000-0010

7	6	5	4	3	2	1	0
WDTCS1	WDTCS0	FWKP	WDTFS	MCKD1	MCKD0	MCDS1	MCDS0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~6: WDTCS1~0. 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	if MCK = 12MHz	if MCK = 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: MCDS[1:0], Reserve for test

CKCON4: Clock Control Register 4

SFR Page = **P only**

SFR Address = 0x42 RESET = 0000-0000

7	6	5	4	3	2	1	0
RCSS2	RCSS1	RCSS0	RPSC2	RPSC1	RPSC0	RTCCS3	RTCCS2
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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

RCSS2, RCSS1, RCSS0	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
1 1 0	Reserved
1 1 1	Reserved

PCON2: Power Control Register 2

SFR Page = **P Only**

SFR Address = 0x44 POR = 0000-0101

7	6	5	4	3	2	1	0
AWBOD1	0	BO1S1	BO1S0	BO1RE	EBOD1	BO0RE	1
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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.BO1S1O and OR1.BO1S0O.

BO1S[1:0]	BOD1 detecting level
0 0	2.0V
0 1	2.4V
1 0	3.6V
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 = 0000-0000

7	6	5	4	3	2	1	0
IVREN	IVRPDE	0	SPWRE	0	0	0	0
R/W	**R/W**	**W**	**R/W**	**W**	**W**	**W**	**W**

Bit 7: IVREN, Internal Voltage Reference Enable.
0: Disable on-chip IVR (2.4V).
1: Enable on-chip IVR (2.4V).

Bit 6 IVRPDE, IVR can be used under power down.
0: Disable IVR keep awake at Power down mode.
1: Enable IVR keep awake at Power down mode

Bit 5: Reserved. Software must write “0” on these bits when PCON3 is written.

Bit 4: SPWRE, SPWF trigger a MCU reset.
0: Disable SPWF to trigger a MCU reset.
1: Enable SPWF to trigger a MCU reset.

Bit 3~0: Reserved. Software must write “0” on these bits when PCON3 is written.

SPCON0: SFR Page Control 0

SFR Page = **P Only**

SFR Address = 0x48 POR = 0000-0000

7	6	5	4	3	2	1	0
--	P6CTL	P4CTL	WRCTL	--	CKCTL0	PWCTL1	PWCTL0
W	**R/W**	**R/W**	**R/W**	**W**	**R/W**	**R/W**	**R/W**

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. WDTCR SFR access Control.
If WRCTL is set, it will disable the WDTCR SFR modified in Page 0~F. WDTCR 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 RESET = 1000-0011

7	6	5	4	3	2	1	0
HSE	IAPO	HSE1	0	0	IORCTL	RSTIO	OCDE
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: HSE, High Speed operation Enable.
0: Select CPU running in lower speed mode ($F_{CPUCLK} \leq 6MHz$) which is slow down internal circuit to reduce power consumption.
1: Enable CPU full speed operation if $F_{CPUCLK} > 6MHz$. Before selecting high frequency clock (> 6MHz) on CPUCLK, software must set HSE to switch internal circuit for high-speed operation.

Bit 6: IAPO, 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. ($F_{CPUCLK} > 25MHz$). 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/Ext Reset/BOR0/BOR1 (if BOR0/1 is enabled).

Bit 1: RSTIO, **n**RST function on I/O,
0: Select I/O pad function for P47.
1: Select I/O pad function for external reset input, nRST.

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.

SPHB: Stack Pointer High Boundary

SFR Page = P Only
SFR Address = 0x53 RESET = 1111-1111

7	6	5	4	3	2	1	0
1	1	1	1	SPHB.3	SPHB.2	SPHB.1	SPHB.0
R	**R**	**R**	**R**	**R/W**	**R/W**	**R/W**	**R/W**

SPHB, it is used for the detection boundary of Stack Pointer warning.
If SPHB == 1111-1111, SPWF will be set when SP ≥ 1111-1111.
If SPHB == 1111-0000, SPWF will be set when SP ≥ 1111-0000.

34. Auxiliary SFRs

AUXR0: Auxiliary Register 0

SFR Page = 0~F
SFR Address = 0xA1 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	C1BKF	PBKF	0	0	INT1H	INT0H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: Reserved. Software must write "**0**" on this bit when AUXR0 is written.

Bit 5: C1BKF, PCA1 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 4: PBKF, PTM0 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 3~2: Reserved. Software must write "**0**" on this bit when AUXR0 is written.

Bit 1: INT1H, INT1 High/Rising trigger enable.
0: Remain nINT1 triggered on low level or falling edge on nINT1 port pin.
1: Set nINT1 triggered on high level or rising edge on nINT1 port pin.

Bit 0: INT0H, INT0 High/Rising trigger enable.
0: Remain nINT0 triggered on low level or falling edge on nINT0 port pin.
1: Set nINT0 triggered on high level or rising edge on nINT0 port pin.

AUXR1: Auxiliary Control Register 1

SFR Page = 0~F
SFR Address = 0xA2 RESET = 0000-0000

7	6	5	4	3	2	1	0
OP1Fr	OP0Fr	CRCDS1	CRCDS0	0	AC1Fr	AC0Fr	DPS
R	R	R/W	R/W	R/W	R	R	R/W

Bit 7: OP1Fr. If OP1CMPF is set to 1, then OP1Fr is also set to 1 according to hardware settings. It is read-only.

Bit 6: OP0Fr. If OP0CMPF is set to 1, then OP0Fr is also set to 1 according to hardware settings. It is read-only.

Bit 5~4: CRCDS1~0. CRC0 Data port Selection bit 1~0.

Bit 3: Reserved. Software must write "**0**" on these bits when AUXR1 is written.
Bit 2: AC1Fr. If AC1F is set to 1, then AC1Fr is also set to 1 according to hardware settings. It is read-only.

Bit 1: AC0Fr. If AC0F is set to 1, then AC0Fr is also set to 1 according to hardware settings. It is read-only.

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

AUXR2: Auxiliary Register 2

SFR Page = 0~F

SFR Address = 0xA3 RESET = 0000-0000

7	6	5	4	3	2	1	0
STAF	STOF	C1PLK	C0PLK	T1X12	T0X12	0	0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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: C1PLK, PCA1 buffered PWM/COPM update control.
0: Buffered PWM/COPM is auto-updated on PTM0 base timer overflow.
1: Disable the buffered PWM/COPM auto-updated.

Bit 4: C0PLK, PTM0 buffered PWM/COPM update control.
0: Buffered PWM/COPM is auto-updated on PTM0 base timer overflow.
1: Disable the buffered PWM/COPM auto-updated.

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.

Bit 2: T0X12, Timer 1 clock source selector while C/T=0.
0: Clear to select SYSCLK/12.
1: Set to select SYSCLK as the clock source.

Bit 1~0: Reserved. Software must write "**0**" on these bits when AUXR0 is written.

AUXR3: Auxiliary Register 3

SFR Page = **0 only**

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
T0PS1	T0PS0	BPOC1	BPOC0	S0PS0	TWIPS1	TWIPS0	T0XL
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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

T0PS1~0	T0/T0CKO
0 0	P3.4
0 1	P4.4
1 0	P2.2
1 1	P1.7

Bit 5~4: BPOC1~0, Beeper output control bits.

BPOC[1:0]	P4.4 function	I/O mode
0 0	P4.4	By P4M0.4 & P4M1.4
0 1	ILRCO/32	By P4M0.4 & P4M1.4
1 0	ILRCO/16	By P4M0.4 & P4M1.4
1 1	ILRCO/8	By P4M0.4 & P4M1.4

For beeper on P4.4 function, it is recommended to configure P4.4 as push-push output mode.

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

S0PS1~0	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/I2C0 Port pin Selection [1:0].

TWIPS1~0	TWI0_SCL	TWI0_SDA
0 0	P4.0	P4.1
0 1	P6.0	P6.1
1 0	P3.1	P3.0
1 1	P2.2	P2.4

Bit 0: T0XL is the Timer 0 clock source selection bit. Please refer T0X12 for T0XL function definition.

AUXR4: Auxiliary Register 4

SFR Page = **1 only**

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
T2PS1	T2PS0	T1PS1	T1PS0	AC1OE	0	AC0OE	0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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

T2PS1~0	T2/T2CKO	T2EX
0 0	P1.0	P1.1
0 1	P3.2	P3.3
1 0	P6.0	P3.5
1 1	P4.5	P4.4

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

T1PS1~0	T1/T1CKO
0 0	P3.5
0 1	P4.5
1 0	P1.7
1 1	P3.3

Bit 3: AC1OE, AC1OUT output enable on port pin.
0: Disable AC1OUT output on port pin.
1: Enable AC1OUT output on P0.5.

Bit 2: Reserved. Software must write "**0**" on these bits when AUXR4 is written.

Bit 1: AC0OE, AC0OUT output enable on port pin.
0: Disable AC0OUT output on port pin.
1: Enable AC0OUT output on P4.5.

Bit 0: Reserved. Software must write "**0**" on these bits when AUXR4 is written.

AUXR5: Auxiliary Register 5

SFR Page = **2 only**

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	C0PPS1	C0PPS0	EC1IPS0	C1COPS	ECIPS0	C0COPS
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~6: Reserved. Software must write "**0**" on these bits when AUXR5 is written.

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

C0PPS1	PWM2A	PWM2B
0	P4.0	P4.1
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: EC1IPS0, PCA1 ECI Port pin Selection0.

EC1IPS0	EC1I
0	ECI
1	P1.5

Bit 2: C1COPS, PCA1 Clock Output (C1CKO) port pin Selection.

C1COPS	C1CKO
0	P4.7
1	P3.3

Bit 1: ECIPS0, PTM0 ECI Port pin Selection0.

ECIPS0	ECI
0	P0.2
1	P1.6

Bit 0: C0COPS, PTM0 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 RESET = 0000-0000

7	6	5	4	3	2	1	0
KBIHPS1	KBIHPS0	KBILPS1	KBILPS0	T3FCS	T2FCS	SnMIPS	S0COPS
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~6: KBIHPS1~0, KBI4~7 Port pin Selection [1:0].

KBIHPS1~0	KBI7	KBI6	KBI5	KBI4
0 0	P1.7	P1.6	P1.5	P1.4
0 1	P1.7	P1.6	P3.5	P3.4
1 0	P2.6	P2.4	P2.3	P2.2
1 1	P1.7	P1.6	P6.1	P6.0

Bit 5~4: KBILPS1~0, KBI0~3 Port pin Selection 0.

KBILPS1~0	KBI3	KBI2	KBI1	KBI0
0 0	P1.3	P1.2	P1.1	P1.0
0 1	P3.3	P3.2	P3.1	P3.0
1 0	P0.7	P0.5	P0.2	P0.1
1 1	P4.5	P4.4	P4.1	P4.0

Bit 3: T3FCS, Reserved for chip test.

Bit 2: T2FCS, Reserved for chip test.

Bit 1: SnMIPS, S0MI, S1MI, S2MI & S3MI Port pin Selection.

SnMIPS	S0MI	S1MI
0	P1.6	P1.0
1	P3.3	P3.5

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

S0COPS	S0CKO
0	P4.7
1	P4.4

AUXR7: Auxiliary Register 7

SFR Page = 4 only
SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	C0CKOE	C1CKOE	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: Reserved. Software must write “0” on these bits when AUXR7 is written.

Bit 5: C0CKOE, PTM clock output (C0CKO) enable.
0: Disable PTM clock output.
1: Enable PTM clock output with PTM base timer overflow rate/2.

Bit 4: C1CKOE, PCA1 clock output (C1CKO) enable.
0: Disable PCA1 clock output.
1: Enable PCA1 clock output with PCA1 base timer overflow rate/2.

Bit 3~0: Reserved. Software must write “0” on these bits when AUXR7 is written.

AUXR8: Auxiliary Register 8

SFR Page = 5 only
SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	C1ICS1	C1ICS0	0	0	S1COPS	T3PS1	T3PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: Reserved. Software must write “0” on these bits when AUXR8 is written.

Bit 6: C1ICS1, PCA1 Input Channel 1 input port pin Selection.

C1ICS1	C1EX1 input
0	C1EX1 Port Pin
1	T2EXI

Bit 5: C1ICS0, PCA1 Input Channel 0 input port pin Selection.

C1ICS0	C1EX0 input
0	C1EX0 Port Pin
1	T3EXI

Bit 4~3: Reserved. Software must write “0” on these bits when AUXR8 is written.

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

S1COPS	S1CKO
0	P4.7
1	P4.5

Bit 1~0: T3PS1~0, Timer 3 Port pin Selection [1:0].

T3PS1~0	T3/T3CKO	T3EX
0 0	P3.3	P3.4

0 1	P3.3	P3.2
1 0	P0.2	P0.1
1 1	P6.0	P6.1

AUXR9: Auxiliary Register 9

SFR Page = 6 only

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
SIDPS1	SIDPS0	T1G1	T0G1	0	0	S1PS1	S1PS0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~6: SID/STWI Port pin Selection [1:0].

SIDPS1~0	STWI_SCL	STWI_SDA
0 0	nINT1	S0MI
0 1	TWI0_SCL	TWI0_SDA
1 0	TWI1_SCL	TWI1_SDA
1 1	T2EXI	T3EXI

Bit 5: T1G1, Gating source selection of Timer 1.

T1G1, T1GATE	T1 Gate source
00	Disable
01	INT1 active
10	TF3 active
11	TI1 active

Bit 4: T0G1, Gating source selection of Timer 0.

T0G1, T0GATE	T0 Gate source
00	Disable
01	INT0 active
10	TF2 active
11	KBI active

Bit 3~2: Reserved. Software must write “0” on these bits when AUXR9 is written.

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

S1PS1~0	RXD1	TXD1
0 0	P1.2	P1.3
0 1	P6.0	P6.1
1 0	P1.0	P1.1
1 1	P3.4	P3.5

AUXR10: Auxiliary Register 10

SFR Page = 7 only

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	SPIPS1	SPIPS0	S0PS1	P60OC1	P60OC0	P60FD
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~6: Reserved. Software must write “0” on these bits when AUXR10 is written.

Bit 5~4: SPIPS1~0, SPI Port pin Selection 1~0.

Pin Options	SPIPS1~0	nSS	MOSI	MISO	SPICLK
0	0 0	P1.4	P1.5	P1.6	P1.7
1	0 1	P0.1	P0.2	P4.1	P4.0
2	1 0	P3.3	P1.5	P1.6	P1.7
3	1 1	P1.7	P3.5	P3.4	P3.3

Bit 3: S0PS1, Serial Port 0 pin Selection 1. (Its function is illustrated at AUXR3.3, S0PS0)

Bit 2~1: 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 crystal mode, XTAL2 and XTAL1 are the alternated function of P6.0 and P6.1. 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]	P6.0 function	I/O mode
0 0	P6.0	By P6M1.0 & P6M0.0
0 1	MCK	By P6M1.0 & P6M0.0
1 0	MCK/2	By P6M1.0 & P6M0.0
1 1	MCK/4	By P6M1.0 & 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 {P6M1.0, P6M0.0} to "01" which selects P6.0 as push-push output mode.

Bit 0: 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.

AUXR11: Auxiliary Register 11

SFR Page = 8 only
SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	I2C1PS1	I2C1PS0	RX1S0	0	T1CKOE	T0CKOE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: Reserved. Software must write "0" on these bits when AUXR11 is written.

Bit 5~4: I2C1PS1~0, TWI1/I2C1 Port pin in Selection [1:0].

I2C1PS1~0	TWI1_SCL	TWI1_SDA
0 0	P1.0	P1.1
0 1	P6.0	P6.1
1 0	P3.0	P3.1
1 1	TWI0_SCL	TWI0_SDA

Bit 3: RX1S0, RXD1 Selection 0.
0: RXD1 is selected by S1PS1~0 (AUXR9.1~0).
1: Force RXD1 is selected on AC0OUT.

Bit 2: Reserved. Software must write "0" on these bits when AUXR11 is written.

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.

AUXR12: Auxiliary Register 12

SFR Page = 9 only
SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
CRCDS2	CRCM0	PDOES1	PDOES0	EDCM0	GPLC0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: CRCDS2. CRC0 Data port Selection bit 2.

Bit 6: CRCM0, CRC mode selection 0.
0: Select CRC16: 0x1021.
1: Select CRC32: 0x04C1_1DB7.

Bit 5~4: PDOES1~0, PDTXOE1 port pin Selection [1:0].

PDOES1~0	PDTXOE
0 0	P1.5
0 1	P3.3
1 0	P6.1
1 1	P3.2

Bit 3: EDCM0. EDC45 mode selection 0.

EDCM0	Mode #	EDC45 operation
0	0	4-bit to 5-bit encoder
1	1	5-bit to 4-bit decoder

Bit 2: GPLC0. GPL Control 0.
0: CPU accesses SFR address 0xA9 on SADDR.
1: CPU accesses SFR address 0xA9 on EDC45.

Bit 1~0: Reserved. Software must write “0” on these bits when AUXR12 is written.

AUXR16: Auxiliary Register 16

SFR Page = D only
SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	C0OPS5	C0OPS4	C0OPS3	C0OPS2	C0OPS1	C0OPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: Reserved. Software must write “0” on these bits when AUXR16 is written.

Bit 5~0: C0OPS5 ~0, PTM Output Pin Selection.

	C0OPS5	C0OPS4	C0OPS3	C0OPS2	C0OPS1	C0OPS0
C0OPSn	CEX5	CEX4	CEX3	CEX2	CEX1	CEX0
0	P0.7	P2.6	P0.5	P2.4	P2.3	P2.2
1	P3.5	P3.2	P3.4	P4.1	P3.3	P4.0

AUXR17: Auxiliary Register 17

SFR Page = E only

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	0	0	0	0	C0OPS1	C0OPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~2: Reserved. Software must write “0” on these bits when AUXR17 is written.

Bit 1~0: C1OPS1 ~0, PCA1 Output Pin Selection.

	C1OPS1	C1OPS0
C1OPSn	C1EX1	C1EX0
0	P6.1	P6.0
1	P3.1	P3.0

SFRPI: SFR Page Index Register

SFR Page = **0~F**

SFR Address = 0xAC RESET = 0000-0000

7	6	5	4	3	2	1	0
0	0	0	0	IDX3	IDX2	IDX1	IDX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~4: Reserved. Software must write “0” on these bits when SFRPI is written.

Bit 3~0: SFR Page Index.

IDX[3:0]	Selected Page
0000	Page 0
0001	Page 1
0010	Page 2
0011	Page 3
……	……
……	……
……	……
1111	Page F

35. 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 "MLink" (The ICE adapter also supports ICP programming function. Refer Section "36.5 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 **MG82F6P32** 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, **MG82F6P32** 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	**MG82F6P32** ISP Start Address
7.5K bytes	6200
7.0K bytes	6400
6.5K bytes	6600
6.0K bytes	6800
5.5K bytes	6A00
5.0K bytes	6C00
4.5K bytes	6E00
4.0K bytes	7000
3.5K bytes	7200
3.0K bytes	7400
2.5K bytes	7600
2.0K bytes	7800
1.5K bytes	7A00
1.0K bytes	7C00
0.5K bytes	7E00
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 **n**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 IAPLB. In default, it is configured to **1K** bytes.

BO1S1O, BO1S0O:

☑,☑: Select BOD1 to detect 2.0V.
☑,☐: Select BOD1 to detect 2.4V.
☐,☑: Select BOD1 to detect 3.6V
☐,☐: 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, 3.6V, 2.4V or 2.0V)
☐: Disabled. BOD1 cannot trigger a RESET to CPU.

WRENO:
☑: Enabled. Set WDTCR.WREN to enable a 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 power down mode (watch mode).
☐: Disabled. Clear WDTCR.NSW to disable the WDT running in power down 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 WDTCR SFR after power-on.

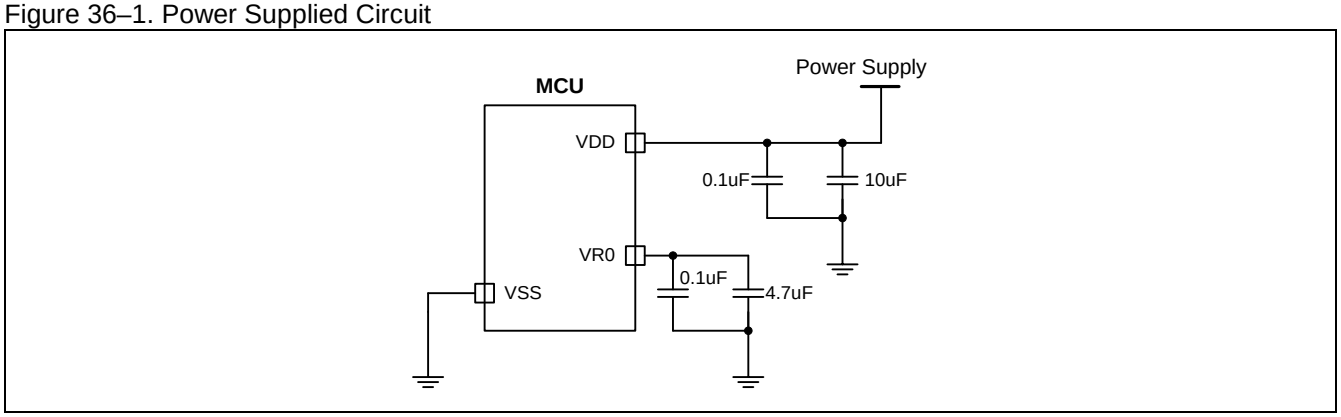
WDSFWP:
☑: Enabled. The WDT SFRs, WREN, NSW, WIDL, PS2, PS1 and PS0 in WDTCR, will be write-protected.
☐: Disabled. The WDT SFRs, WREN, NSW, WIDL, PS2, PS1 and PS0 in WDTCR, are free for writing of software.

36. Application Notes

36.1. Power Supply Circuit

To have the **MG82F6P32** work with power supply varying from 2.0V to 5.5V, adding some external decoupling and bypass capacitors is necessary, as shown in Figure 36–1.

Figure 36–1. Power Supplied Circuit



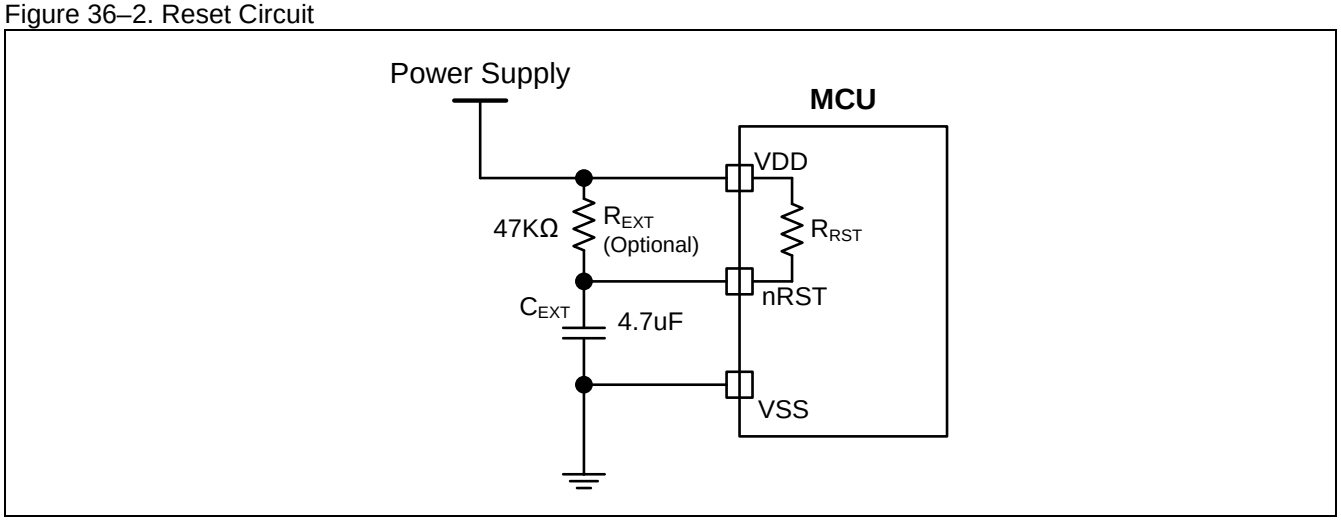
36.2. Reset Circuit

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 36–2 shows the external reset circuit, which consists of a capacitor C_{EXT} connected to VSS (ground) and a resistor R_{EXT} connected to VDD (power supply).

In general, R_{EXT} is optional because the **n**RST pin has an internal pull-up resistor (R_{RST}). This internal diffused resistor to VDD permits a power-up reset using only an external capacitor C_{EXT} to VSS.

See Section "37.2 DC Characteristics" for R_{RST} value.

Figure 36–2. Reset Circuit



36.3. XTAL Oscillating Circuit

To achieve successful and exact oscillating (up to 24MHz), the capacitors C1 and C2 are necessary, as shown in Figure 36–3. Normally, C1 and C2 have the same value. Table 36–1 lists the C1 & C2 value for the different frequency crystal application.

Figure 36–3. XTAL Oscillating Circuit

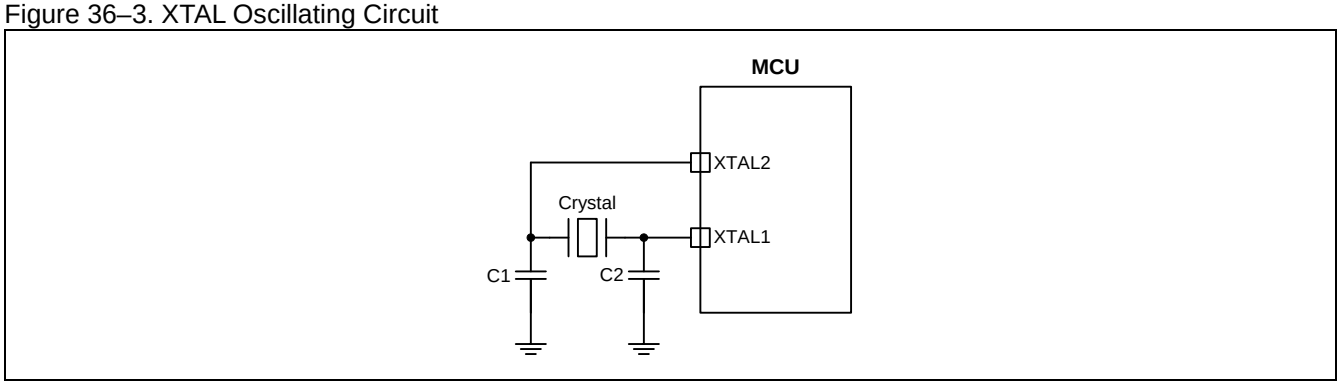


Table 36–1. Reference Capacitance of C1 & C2 for crystal oscillating circuit

Crystal	C1, C2 Capacitance
16MHz ~ 25MHz	10pF
6MHz ~ 16MHz	15pF
2MHz ~ 6MHz	33pF
32768Hz	7pF

36.4. ICP and OCD Interface Circuit

MG82F6P32 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.

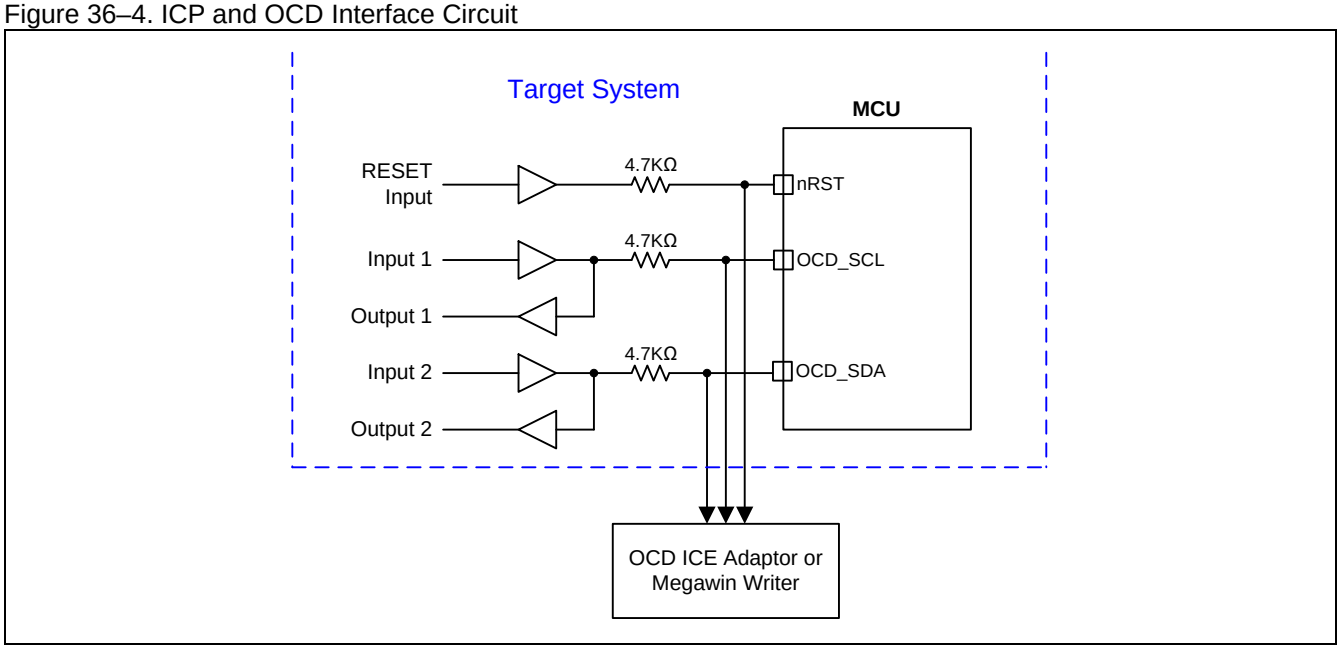
The ICP interface allows the ICP_SCL/ICP_SDA pins to be shared with user functions so that In-Chip Flash Programming function could be performed. This is practicable because ICP communication is performed when the device is in the halt state, where the on-chip peripherals and user software are stalled. 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 36–4.

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 MG82F6P32 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 mentioned ICP interface isolation in Figure 36–4, 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 OCDE as "1". Or "Erase" the on-chip flash by ICP which cleans the user software to stop the port pins switching.

Figure 36–4. ICP and OCD Interface Circuit



36.5. 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.

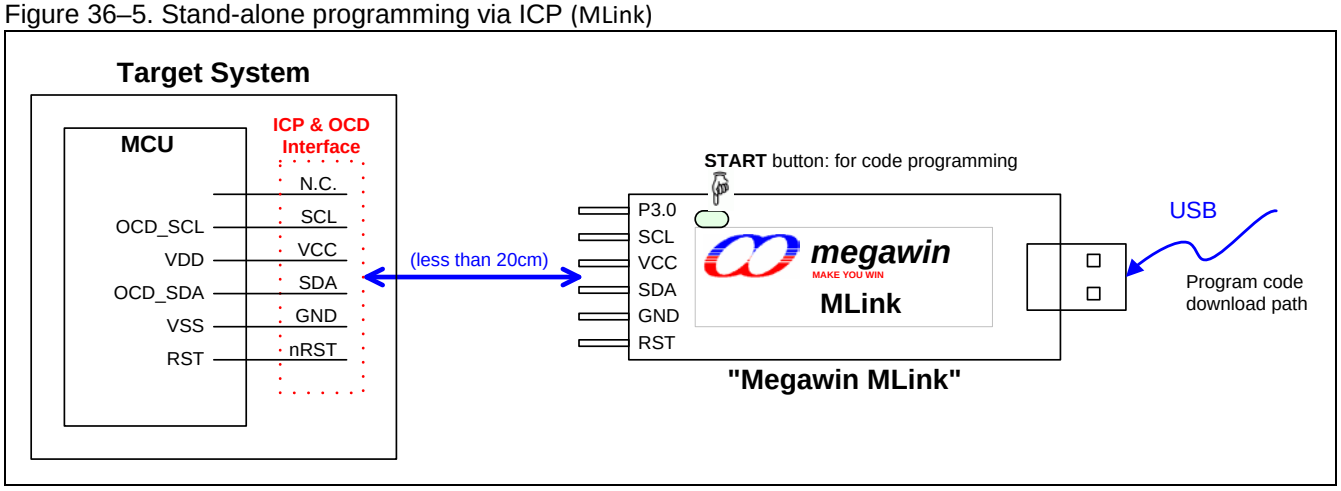
The proprietary 6-pin "MLink" can support the In-Circuit Programming of **MG82F6P32**. "MLink" has the in-system storage to store the user program code and device options. So, the tools can perform a portable and stand-alone programming without a host on-line, such as connecting the tool to PC. Following lists the features of the ICP function:

Features

- 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 36–5. 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 "MLink" to the target MCU; the RST line to halt the MCU, and the VCC & GND are the power supply entry of the 6-pin "MLink" 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 "MLink".

Figure 36–5. Stand-alone programming via ICP (MLink)



36.6. On-Chip-Debug Function

The **MG82F6P32** 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 “MLink” for the user, as shown in Figure 36–6. 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 36–6.

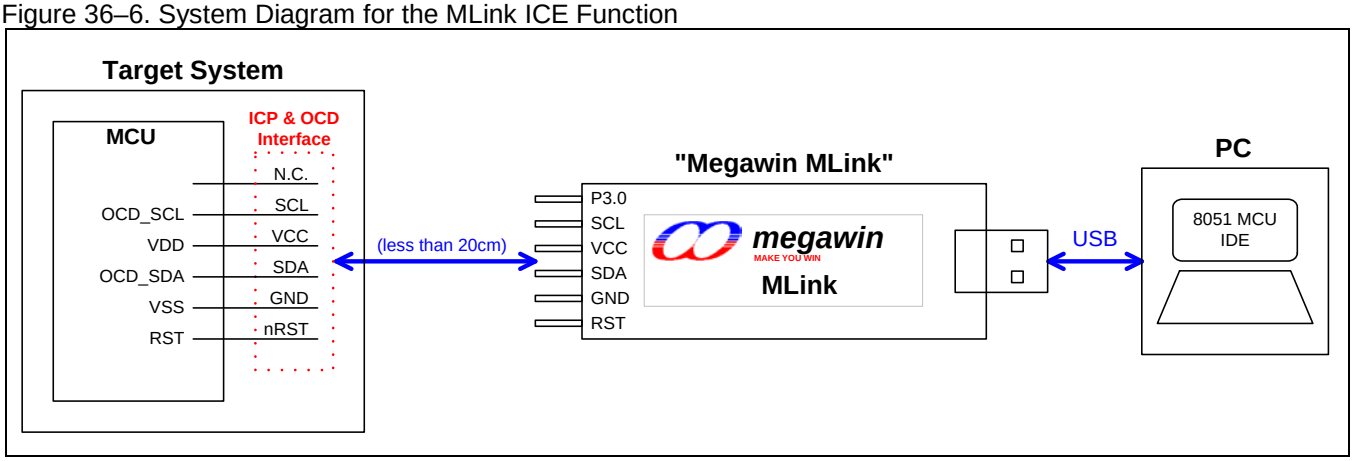
In addition, the most powerful feature is that it can directly connect the user’s target system to the Keil 8051 IDE software for debugging, which directly utilizes the Keil IDE’s dScope-Debugger function. Of course, all the advantages are based on your using Keil 8051 IDE software.

Note: “Keil” is the trade mark of “Keil Elektronik GmbH and Keil Software, Inc.”.

Features

- Megawin proprietary 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 36–6. System Diagram for the MLink ICE Function



Note: For more detailed information about the OCD ICE, please feel free to contact Megawin.

37. Electrical Characteristics

37.1. Absolute Maximum Rating

Parameter	Rating	Unit
Ambient temperature under bias	-40 ~ +105	°C
Storage temperature	-65 ~ + 150	°C
Voltage on any Port I/O Pin or nRST with respect to VSS	-0.5 ~ VDD + 0.5	V
Voltage on VDD with respect to VSS	-0.5 ~ +6.0	V
Maximum total current through VDD and VSS	200	mA
Maximum output current sunk by any Port pin	40	mA

*Note: stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the devices at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

37.2. DC Characteristics

VDD = 5.0V±10%, VSS = 0V, T_A = 25 °C and execute NOP for each machine cycle, unless otherwise specified

Symbol	Parameter	Test Condition	Limits			Unit
			min	typ	max	
Input/ Output Characteristics						
V_{IH1}	Input High voltage (All I/O Ports)	Except P6.0, P6.1	0.6			VDD
V_{IH2}	Input High voltage (nRST, P6.0, P6.1)		0.75			VDD
V_{IL1}	Input Low voltage (All I/O Ports)	Except P6.0, P6.1			0.15	VDD
V_{IL2}	Input Low voltage (nRST, P6.0, P6.1)				0.2	VDD
I_{IH}	Input High Leakage current (All I/O Ports)	V_{PIN} = VDD		0	±1	uA
I_{IL1}	Logic 0 input current (P3 in quasi-mode)	V_{PIN} = 0.4V		-20	-30	uA
I_{IL2}	Logic 0 input current (All Input only or open-drain Ports)	V_{PIN} = 0.4V		0	-1	uA
I_{H2L}	Logic 1 to 0 input transition current (P3 in quasi-mode) (2)	V_{PIN} =1.8V		-300	-450	uA
I_{OH1}	Output High current (P3 in quasi-Mode)	VDD=5V; V_{PIN} =2.4V	-180	-260		uA
		VDD=3.3V; V_{PIN} =2.4V	-50	-80		uA
		VDD=1.8V; V_{PIN} =1.4V	-10	-15		uA
I_{OH2}	Output High current (All push-pull output ports)	VDD=5V; V_{PIN} =2.4V	-30	-34		mA
		VDD=3.3V; V_{PIN} =2.4V	-8	-11		mA
		VDD=1.8V; V_{PIN} =1.4V	-2	-2.6		mA
I_{OH3}	Output High current (All push-pull output ports on low driving strength, except nRST Pin)	VDD=5V; V_{PIN} =2.4V	-8	-13.6		mA
		VDD=3.3V; V_{PIN} =2.4V	-3	-4.6		mA
		VDD=1.8V; V_{PIN} =1.4V	0.7	-1.1		mA
I_{OL1}	Output Low current (All I/O Ports)	VDD=5V; V_{PIN} =0.4V	18	24		mA
		VDD=3.3V; V_{PIN} =0.4V	14	17		mA
		VDD=1.8V; V_{PIN} =0.4V	6	8		mA
I_{OL2}	Output Low current (All push-pull output ports on low driving strength, except nRST Pin)	VDD=5V; V_{PIN} =0.4V	1.8	3.1		mA
		VDD=3.3V; V_{PIN} =0.4V	1.2	2.2		mA
		VDD=1.8V; V_{PIN} =0.4V	0.55	1.1		mA
R_{RST}	Internal reset pull-down resistance	VDD=5V	265	270	290	Kohm
		VDD=3.3V	450	460	490	Kohm
		VDD=1.8V	1430	1500	1600	Kohm
Power Consumption						
I_{OP1}	Normal mode operating current	CPUCLK=SYSCLK = 32MHz @ IHRCO with PLL, HSE1=HSE=1		7.5		mA
I_{OP2}		CPUCKL=SYSCLK = 24MHz @ IHRCO with PLL, HSE1=0, HSE=1		6.5		mA
I_{OP3}		CPUCLK=SYSCLK = 24MHz @ XTAL 24MHz , HSE1=0, HSE=1		7.5		mA
I_{OP4}		CPUCLK=SYSCLK = 24MHz @ ECKI 24MHz , HSE1=0, HSE=1		5.8		mA
I_{OP5}		CPUCKL=SYSCLK = 12MHz @ IHRCO, HSE1=0, HSE=0		3.8		mA
I_{OP6}		CPUCLK=SYSCLK = 12MHz @ XTAL 12MHz , HSE1=0, HSE=0		4.6		mA
I_{OP7}		CPUCLK=SYSCLK = 12MHz @ ECKI 12MHz , HSE1=0, HSE=0		3.6		mA
I_{OP8}		CPUCLK=SYSCLK = 2MHz @ XTAL 2MHz		1.8		mA
I_{OP9}		CPUCLK=SYSCLK = 4MHz @ XTAL 4MHz		2.2		mA
I_{OP10}		CPUCLK=SYSCLK = 6MHz @ XTAL 6MHz		2.8		mA

I_{OP11}		CPUCLK=SYSCLK = 32KHz @ XTAL 32KHz		65		uA
I_{OPS1}	Slow mode operating current	CPUCLK=SYSCLK = 12MHz/128 @ IHRCO		2.8		mA
I_{IDLE1}	Idle mode operating current	SYSCLK = 12MHz @ IHRCO		1.7		mA
I_{IDLE2}		SYSCLK = 12MHz/128 @ IHRCO		0.38		mA
I_{IDLE3}		SYSCLK = 32KHz @ ILRCO		51		uA
I_{IDLE4}		SYSCLK = 32KHz @ XTAL 32KHz		57		uA
I_{SUB1}	Sub-clock mode operating current	SYSCLK = 32KHz @ ILRCO, BOD1 disabled		4.2		uA
I_{SUB2}		SYSCLK = 32KHz/128 @ ILRCO, BOD1 disabled		4		uA
I_{WAT}	Watch mode operating current	WDT = 32KHz @ ILRCO in PD mode		2.5		uA
I_{MON1}	Monitor Mode operating current	BOD1 enabled in PD mode		8.6		uA
I_{RTC1}	RTC Mode operating current	RTC operating in PD mode, VDD = 5.0V		2.55		uA
I_{PD1}	Power down mode current			1.5	4	uA
BOD0/BOD1 Characteristics						
V_{BOD0}	BOD0 detection level	T_A = -40°C to +105°C		1.7		V
V_{BOD10}	BOD1 detection level for 2.0V	T_A = -40°C to +105°C		2.0		V
V_{BOD10}	BOD1 detection level for 2.4V	T_A = -40°C to +105°C		2.37		V
V_{BOD11}	BOD1 detection level for 3.6V	T_A = -40°C to +105°C		3.6		V
V_{BOD11}	BOD1 detection level for 4.2V	T_A = -40°C to +105°C		4.2		V
I_{BOD1}	BOD1 Power Consumption	T_A = +25°C, VDD=5.0V		6.5		uA
		T_A = +25°C, VDD=3.3V		5		
Operating Condition						
V_{PSR}	Power-on Slop Rate	T_A = -40°C to +85°C	0.05			V/ms
V_{POR1}	Power-on Reset Valid Voltage	T_A = -40°C to +85°C			0.1	V
V_{OP1}	CPU Operating Speed 0-32MHz	T_A = -40°C to +105°C	2.7		5.5	V
V_{OP2}	CPU Operating Speed 0-24MHz	T_A = -40°C to +105°C	2.2		5.5	V
V_{OP3}	CPU Operating Speed 0-12MHz	T_A = -40°C to +105°C	1.8		5.5	V

(1) Data based on characterization results, not tested in production.

(2) I/O under Quasi-Bidirectional mode, when input voltage High transfer to Low and across the threshold voltage, the internal “Weak” pull up will be turn off. I_{H2L} indicates the current near the threshold voltage. Please reference “Figure 14–1. Port 3 Quasi-Bidirectional I/O”.

(3) All current flowing into the chip has a positive value, and current flowing out of the chip has negative value.

37.3. External Reset Characteristics

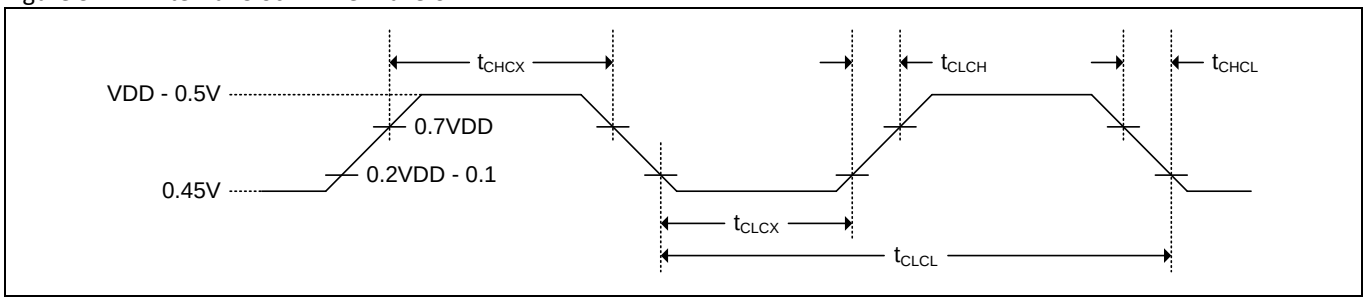
Parameter	Test Condition	Limits			Unit
		min	typ	max	
Supply Voltage		1.8		5.5	V
External nRST low time to generate System Reset	SYSCLK = IHRCO	32			us

37.4. External Clock Characteristics

VDD = 2.4V ~ 5.5V, VSS = 0V, T_A = -40°C to +105°C, unless otherwise specified

Symbol	Parameter	Oscillator				Unit
		Crystal Mode		ECKI Mode		
		Min.	Max	Min.	Max	
$1/t_{CLCL}$	Oscillator Frequency	0.032	25	0	25	MHz
$1/t_{CLCL}$	Oscillator Frequency (VDD = 2.0V ~ 5.5V)	0.032	12	0	12	MHz
t_{CLCL}	Clock Period	41.6		27.7		ns
t_{CHCX}	High Time	0.4T	0.6T	0.4T	0.6T	t_{CLCL}
t_{CLCX}	Low Time	0.4T	0.6T	0.4T	0.6T	t_{CLCL}
t_{CLCH}	Rise Time		5		5	ns
t_{CHCL}	Fall Time		5		5	ns

Figure 37–1. External Clock Drive Waveform



37.5. IHRCO Characteristics

Parameter	Test Condition	Limits			Unit
		min	typ	max	
Supply Voltage		1.8		5.5	V
IHRCO Frequency	T_A = +25°C, AFS = 0		12		MHz
	T_A = +25°C, AFS = 1		11.059		MHz
IHRCO Frequency Deviation (factory calibrated)	TA = +25°C	-1.0		+1.0	%
	TA = -40°C to +105°C	-2.0(1)		+2.0(1)	%
IHRCO Start-up Time	TA = -40°C to +105°C			32(1)	us
IHRCO Power Consumption	TA = +25°C, VDD=5.0V		(1)		uA

(1) Data based on characterization results, not tested in production.

37.6. ILRCO Characteristics

Parameter	Test Condition	Limits			Unit
		min	typ	max	
Supply Voltage		1.8		5.5	V
ILRCO Frequency	TA = +25°C		32		KHz
ILRCO Frequency Deviation	TA = +25°C	-8(1)		+8(1)	%
	TA = -40°C to +105°C	-15(1)		+15(1)	%

(1) Data based on characterization results, not tested in production.

37.7. CKM Characteristics

Parameter	Test Condition	Limits			Unit
		min	typ	max	
Supply Voltage	TA = -40°C to +105°C	2.2		5.5	V
Clock Input Range	TA = -40°C to +105°C	4.5(1)		6.5(1)	MHz
CKM Start-up Time	TA = -40°C to +105°C	30(2)		100(2)	us
CKM Power Consumption	TA = +25°C, VDD=5.0V, CKM = 96MHz		350		uA

(1) Data guaranteed by design, not tested in production.
(2) Data based on characterization results, not tested in production.

37.8. Flash Characteristics

Parameter	Test Condition	Limits			Unit
		min	typ	max	
Supply Voltage	TA = -40°C to +105°C	1.8		5.5	V
Flash Write (Erase/Program) Voltage	TA = -40°C to +105°C	1.8		5.5	V
Flash Erase/Program Cycle	TA = -40°C to +105°C	20,000			times
Flash Data Retention	TA = +25°C	100			year

37.9. OPA Characteristics

VDD=5.0V, T_A= -40°C ~ +85°C unless otherwise specified

Parameter	Test Condition	Limits			Unit
		min	typ	max	
Supply Voltage		2.4	--	5.5	V
Single OPA I_Q	High power, OPPWR:OPnLP = 00	0.8	1.13	1.45	mA
	Medium high power, OPPWR:OPnLP = 01	--	0.65	--	
	Medium low power, OPPWR:OPnLP = 10	--	0.55	--	
	Low power, OPPWR:OPnLP = 11	0.15	0.3	0.35	
Offset V_{OS}	Calibrated @ Vi = 1.2V		±0.5	±1.0	mV
V_{CM}	High power, Error Gain=±0.5%	0.53	--	VDD-0.1	V
	High power, Error Gain=±1.0%	0.30	--	VDD-0.1	
	Medium high power, Error Gain=±0.5%	0.53	--	VDD-0.05	
	Medium High power, Error Gain=±1.0%	0.30	--	VDD-0.05	
	Medium low power, Error Gain=±0.5%	0.53	--	VDD-0.05	
	Medium low power, Error Gain=±1.0%	0.30	--	VDD-0.05	
	low power, Error Gain=±0.5%	0.63	--	VDD-0.05	
	low power, Error Gain=±1.0%	0.43	--	VDD-0.05	
V_{OL}	High power	0.1	--	--	V
	Medium high power	0.13	--	--	
	Medium low power	0.1	--	--	
	low power	0.13	--	--	
V_{OH}	High power	--	--	VDD-0.12	V
	Medium high power	--	--	VDD-0.15	
	Medium low power	--	--	VDD-0.15	
	low power	--	--	VDD-0.20	
BandWidth	High power, C_{LOAD}// R_{LOAD} = 110pF // 2k ohm (peaking ,Gain=110%)	15.5	19.0	--	MHz
	High power, C_{LOAD}// R_{LOAD} = 30pF // 2k ohm	--	--	25.0	
	High power, C_{LOAD}// R_{LOAD} = 20pF // 2k ohm	--	--	22.0	
	Medium high power, C_{LOAD}// R_{LOAD} = 110pF // 2k ohm (peaking ,Gain=130%)	10.5	13.0	--	
	Medium low power, C_{LOAD}// R_{LOAD} = 110pF // 2k ohm (peaking ,Gain=116%)	8.3	10.3	--	

	Low power, C_{LOAD}// R_{LOAD} = 110pF // 2k ohm (peaking ,Gain=136%)	8.5	9.4	--	
Slew Rate	High power, C_{LOAD}// R_{LOAD} = 110pF // 2k ohm(from 20 and 80% of output voltage)	--	7.0	--	V/us
	Medium high power, C_{LOAD}// R_{LOAD} = 110pF // 2k ohm(from 20 and 80% of output voltage)	--	7.5	--	
	Medium low power, C_{LOAD}// R_{LOAD} = 110pF // 2k ohm(from 20 and 80% of output voltage)	--	2.7	--	
	Low power, C_{LOAD}// R_{LOAD} = 110pF // 2k ohm(from 20 and 80% of output voltage)	--	3.6	--	
stable time		--	--	2.5	us
A_{OL}					dB
CMRR					dB
PSRR					dB

Note1: Data guaranteed by design, not tested in production.

37.10. PGA Characteristics

VDD=5.0V, T_A= -40℃ ~ +85℃ unless otherwise specified

Parameter	Test Condition	Limits			Unit
		min	typ	max	
Supply Voltage		2.4	--	5.5	V
Single OPA I_Q	Normal power	0.45	0.56	0.69	mA
	Low power	0.22	0.28	0.35	
Offset V_{OS}	1st	--	--	±4.40	mV
	2ed	--	±0.4	±1.0	
V_{CM}(software calibrated)	G=1	0.028		VDD-0.8	V
	G=2	0.014		VDD-2.5	
	G=4	0.010	1	--	
	G=8	0.009	0.6	--	
	G=16	0.008	0.31	--	
	G=32	0.007	0.155	--	
	G=64	0.006	0.077	--	
	G=128	0.005	0.038	--	
V_{OL}(software calibrated)		0.002	--	--	V
V_{OH}(software calibrated)		--	--	VDD-0.1	V
Normal Power Mode Bandwidth	G=1		3.3		MHz
	G=2		7		
	G=4		5		
	G=8		4		
	G=16		2		
	G=32		3		
	G=64		--		
	G=128		--		
Slew Rate	Normal power	--	15	--	V/us
	Low power	--	7.3	--	
stable time		--	--	1	us
CMRR					dB
PSRR					dB

Note: Data guaranteed by design, not tested in production.

37.11. ADC Characteristics

VDD=5.0V, T_A= -40℃ ~ +85℃ unless otherwise specified

Parameter	Test Condition	Limits			Unit
		min	typ	max	

Supply Voltage			2.4		5.5	V
Resolution				12		bits
DC Accuracy T_A= 25°C						
Integral Nonlinearity	VDD = VREF+ ≥ 5V, 1M sps Power level: High and Medium high		-2.4		2.9	LSB
	VDD = VREF+ ≥ 2.4V, 500K sps Power level: not include low power mode		-3.3		2.6	
Differential Nonlinearity	VDD = VREF+ ≥ 5V, 1M sps Power level: High and Medium high		-1		3.3	LSB
	VDD = VREF+ ≥ 2.4V, 500K sps		-1		2	
Offset Error	VDD= 2.4V~5.5V			1	3.5	LSB
DC Accuracy T_A= -40°C ~ +85°C						
Integral Nonlinearity	VDD = VREF+ ≥ 5V, 500K sps Power level: High		-2.9		3	LSB
	VDD = VREF+ ≥ 2.4V, 250K sps Power level: High and Medium high		-4		3.1	
Differential Nonlinearity	VDD = VREF+ ≥ 5V, 500K sps Power level: High		-1		3.5	LSB
	VDD = VREF+ ≥ 2.4V, 250K sps		-1		2.1	
Offset Error	VDD= 2.4V~5.5V			1	3.5	LSB
Conversion Rate						
SAR Conversion Clock				12	24	MHz
Conversion Time in SAR Clocks				24		clocks
Suggested Conversion Rate				500	1000	K sps
Analog Inputs						
Input Voltage Range	Single Ended (AIN+ – GND)		0		VDD	V
C_{ADC} Input Capacitance note1				5		pF
Input Sampling switch resistance note1	VDD = 5V	Vin = 0.1V	450	525	610	Ω
		Vin = 2.5V	660	800	1000	Ω
		Vin = 5V	950	1100	1300	Ω
	VDD = 2.4V	Vin = 0.1V	690	830	995	Ω
		Vin = 1.2V	1.95	3.66	13	KΩ
		Vin = 2.4V	1.5	1.9	2.4	KΩ
Switch Channel Stable Time						
Original pin and Target pin voltage switch between VDD or GND	CH0(VDD)→CH1(GND)			0		us
	CH0(GND)→CH1(VDD)			0		
Original pin voltage = VDD switch to target pin with pulldown resistor	CH0(VDD)→CH1(51K Pulldown)			4.41		
	CH0(VDD)→CH1(10K Pulldown)			0.5		
Original pin voltage = GND switch to target pin with Pullup resistor	CH0(GND)→CH1(51K Pullup)			7.5		
	CH0(GND)→CH1(10K Pullup)			1.25		
Original pin voltage = VDD switch to resistor divider (VDD/2)	CH0(VDD)→CH1(VDD/2，51K resistor divider)			3.91		
	CH0(VDD)→CH1(VDD/2，10K resistor divider)			3.08		
Original pin voltage = GND switch to resistor divider (VDD/2)	CH0(GND)→CH1(VDD/2，51K resistor divider)			0.33		us
	CH0(GND)→CH1(VDD/2，10K resistor divider)			0.5		
Power Consumption						
Power Supply Current	ADPS<1:0>=00		1.2	1.4	1.8	mA
	ADPS<1:0>=01		--	1.38	--	
	ADPS<1:0>=10		--	1.37	--	
	ADPS<1:0>=11		--	1.33	--	

Note1: Data guaranteed by design, not tested in production.

37.12. IVR Characteristics

VDD=5.0V±10%, VSS=0V, T_A =-40°C to +105°C, C_{LOAD}=4.7upF/0.1ohm-ESR unless otherwise specified

Parameter	Test Condition	Limits			Unit
		Min.	Typ.	Max.	
Supply Range					
Supply Voltage		2.7	5.0	5.5	V
Operation Current	Normal Power State		850		uA

DC Accuracy					
Output Supply Voltage	-40°C ~ +85°C		2.4		V
Spread over the temperature range	VDD = 3.3V±10mV	-15		+15	mV

37.13. Analog Comparator AC0/AC1 Characteristics

VDD=5.0V, T_A= -40°C ~ +85°C unless otherwise specified

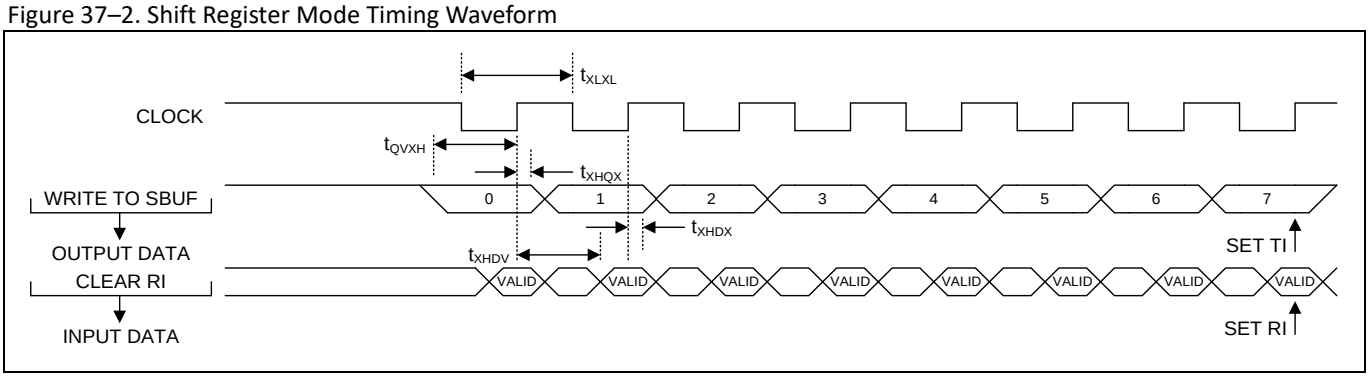
Parameter	Test Condition	Limits			Unit
		Min.	Typ.	Max.	
Supply Range					
Supply Voltage		2.0	5.0	5.5	V
Operation Current	Normal Power State		57		uA
	Low Power State		12		uA
R-Ladder (32 levels)	VDD= 5.0V		95		uA
R-Ladder (24 levels)	VDD= 5.0V		117		uA
DC Accuracy					
Input Voltage Range		0		VDD-1	V
Input Offset Voltage-Trimmed	VDD= 5.0V		0.8		mV
Input Offset Voltage-Non-Trimmed	VDD= 5.0V	-20		20	mV
Input Common Mode Voltage		0.02		VDD-1	V
Comparator Hysteresis (Input Hysteresis select = **0**V)	VDD= 5.0V		0		mV
	VDD= 3.0V		0		
	VDD= 2.0V		0		
Comparator Hysteresis (Input Hysteresis select = **±10mV**)	VDD= 5.0V	10	18	24	mV
	VDD= 3.0V	16	21	30	
	VDD= 2.0V	18	23	32	
Comparator Hysteresis (Input Hysteresis select = **±20mV**)	VDD= 5.0V	31	42	59	mV
	VDD= 3.0V	35	46	65	
	VDD= 2.0V	37	50	70	
Comparator Hysteresis (Input Hysteresis select = **±60mV**)	VDD= 5.0V	80	135	190	mV
	VDD= 3.0V	75	140	195	
	VDD= 2.0V	70	130	200	
Response Time	Rising(V_{DD} = 5V, Ta = 25°C, V_{OD}=50mV, V_{CM} < 50mV)		43		ns
	Falling(V_{DD} = 5V, Ta = 25°C, V_{OD}=50mV, V_{CM} < 50mV)		54		ns
	Rising(V_{DD} ≥ 3V, V_{OD}=100mV, 0.5V ≤ V_{CM} ≤ VDD-1.5V)		65	75	ns
	Falling(V_{DD} ≥ 3V, V_{OD}=100mV, 0.5V ≤ V_{CM} ≤ VDD-1.5V)		70	100	ns
Power on Time (from Power-down)	Normal mode (V_{DD} = 5V, SYSCLK = CPUCLK = 12MHz)		--		us
	Low power mode (V_{DD} = 5V, SYSCLK = CPUCLK = 12MHz)		--		us

37.14. Serial Port Timing Characteristics

VDD = 5.0V±10%, VSS = 0V, T_A = -40°C to +105°C, unless otherwise specified

Symbol	Parameter	Min.	Max	Unit
t_{XLXL}	Serial Port Clock Cycle Time	12T		T_{SYSCLK}
t_{QVXH}	Output Data Setup to Clock Rising Edge	10T-20		ns
t_{XHQX}	Output Data Hold after Clock Rising Edge	T-10		ns
t_{XHDX}	Input Data Hold after Clock Rising Edge	5		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		2T-10	ns

Figure 37–2. Shift Register Mode Timing Waveform



37.15. SPI Timing Characteristics

VDD = 5.0V±10%, VSS = 0V, T_A = -40°C to +105°C, unless otherwise specified

Symbol	Parameter	Min	Max	Units
Master Mode Timing				
1/(t_{MCKH} + t_{MCKL})	SPI Clock Frequency @VDD = 3.0V ~ 5.5V		24MHz	MHz
	SPI Clock Frequency @VDD = 1.8V ~ 3.0V		16MHz	MHz
t_{MCKH}	SPICLK High Time	1T		T_{SYSCLK}
t_{MCKL}	SPICLK Low Time	1T		T_{SYSCLK}
t_{MIS}	MISO Valid to SPICLK Sample Edge	10		ns
t_{MIH}	SPICLK Shift Edge to MISO Change	0		ns
t_{MOH}	SPICLK Shift Edge to MOSI Change		10	ns
Slave Mode Timing				
1/(t_{CKH} + t_{CKL})	SPI Clock Frequency @VDD = 3.0V ~ 5.5V		16MHz	MHz
	SPI Clock Frequency @VDD = 1.8V ~ 3.0V		12MHz	MHz
t_{SE}	nSS Falling to First SPICLK Edge	2T		T_{SYSCLK}
t_{SD}	Last SPICLK Edge to nSS Rising	2T		T_{SYSCLK}
t_{SEZ}	nSS Falling to MISO Valid		4T	T_{SYSCLK}
t_{SDZ}	nSS Rising to MISO High-Z		4T	T_{SYSCLK}
t_{CKH}	SPICLK High Time	2T		T_{SYSCLK}
t_{CKL}	SPICLK Low Time	2T		T_{SYSCLK}
t_{SIS}	MOSI Valid to SPICLK Sample Edge	1T		T_{SYSCLK}
t_{SIH}	SPICLK Sample Edge to MOSI Change	1T		T_{SYSCLK}
t_{SOH}	SPICLK Shift Edge to MISO Change		2T	T_{SYSCLK}
t_{SLH}	Last SPICLK Edge to MISO Change (CPHA = 1 ONLY)	1T	2T	T_{SYSCLK}

Figure 37–3. SPI Master Transfer Waveform with CPHA=0

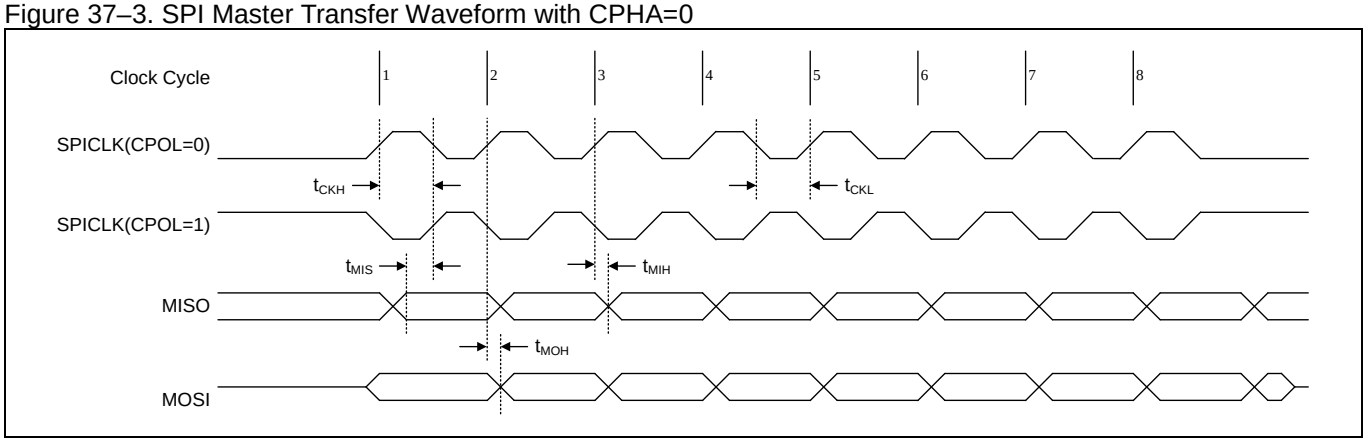


Figure 37–4. SPI Master Transfer Waveform with CPHA=1

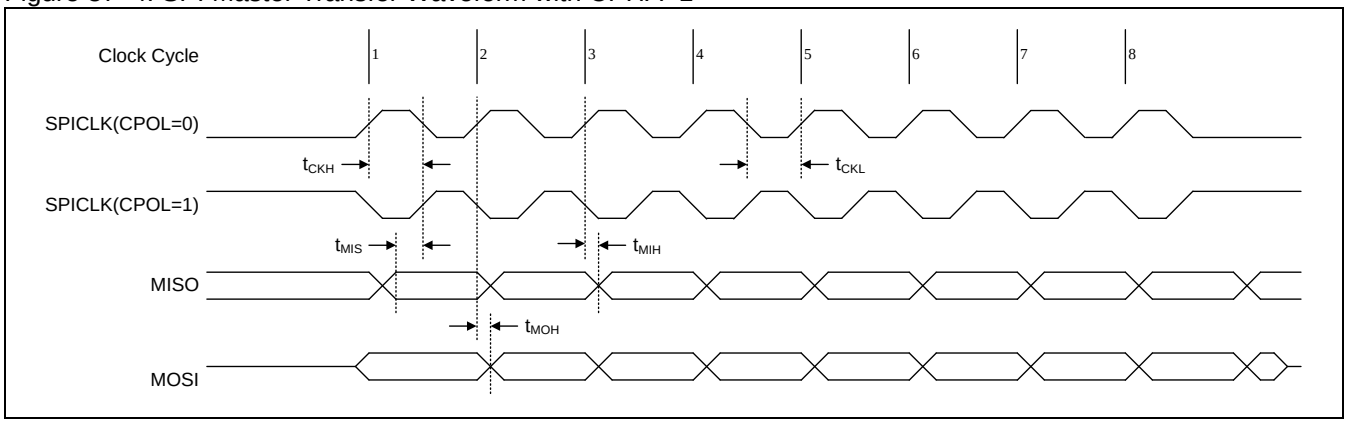


Figure 37–5. SPI Slave Transfer Waveform with CPHA=0

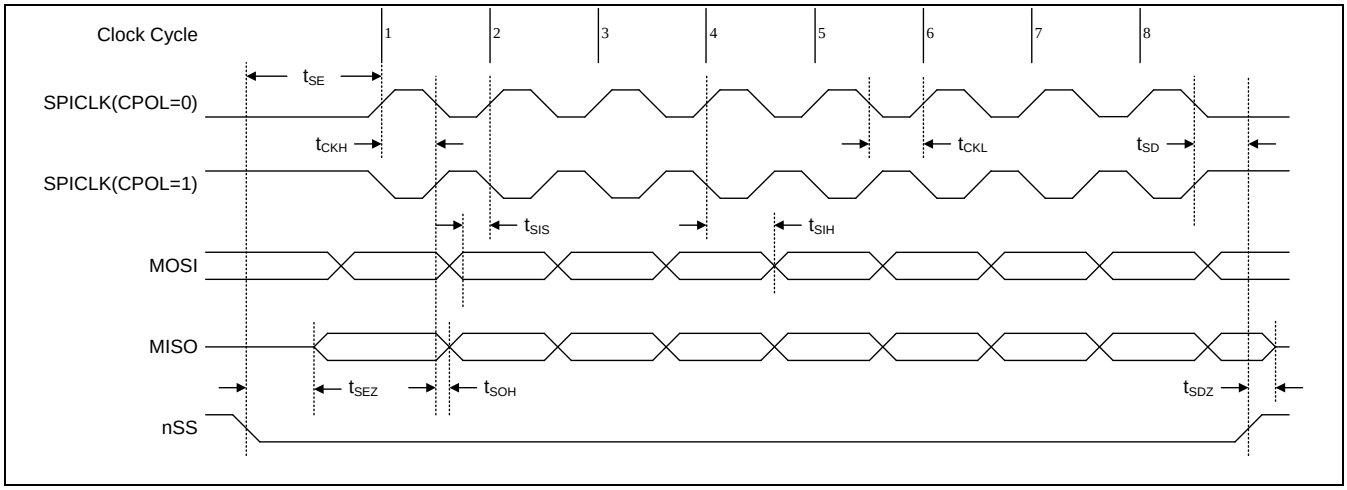
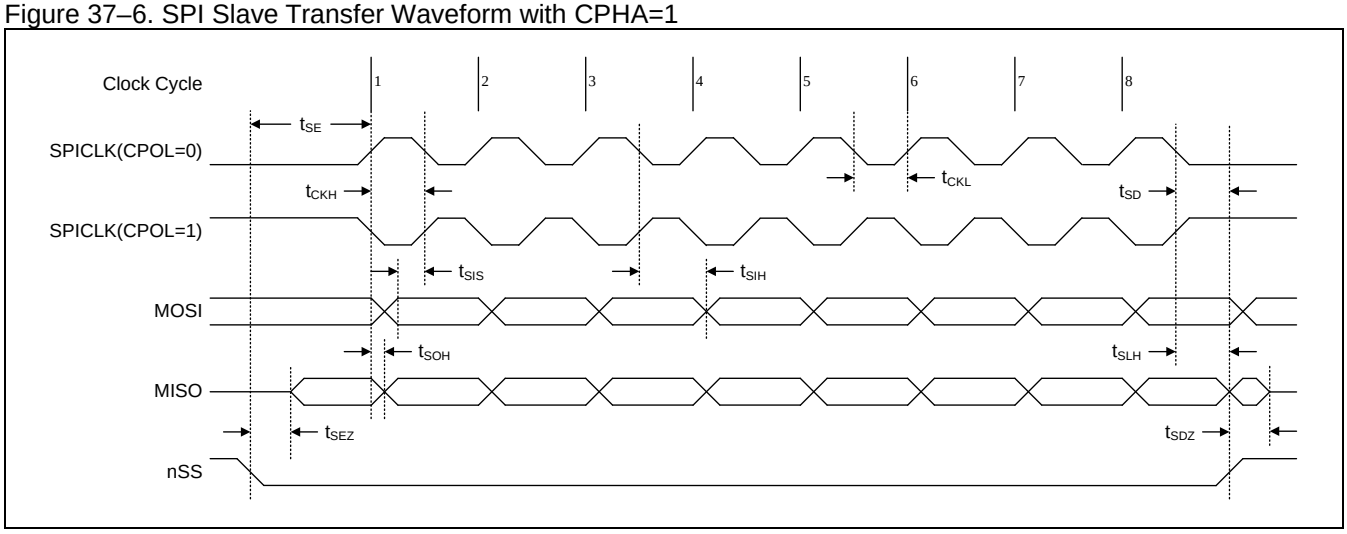


Figure 37–6. SPI Slave Transfer Waveform with CPHA=1

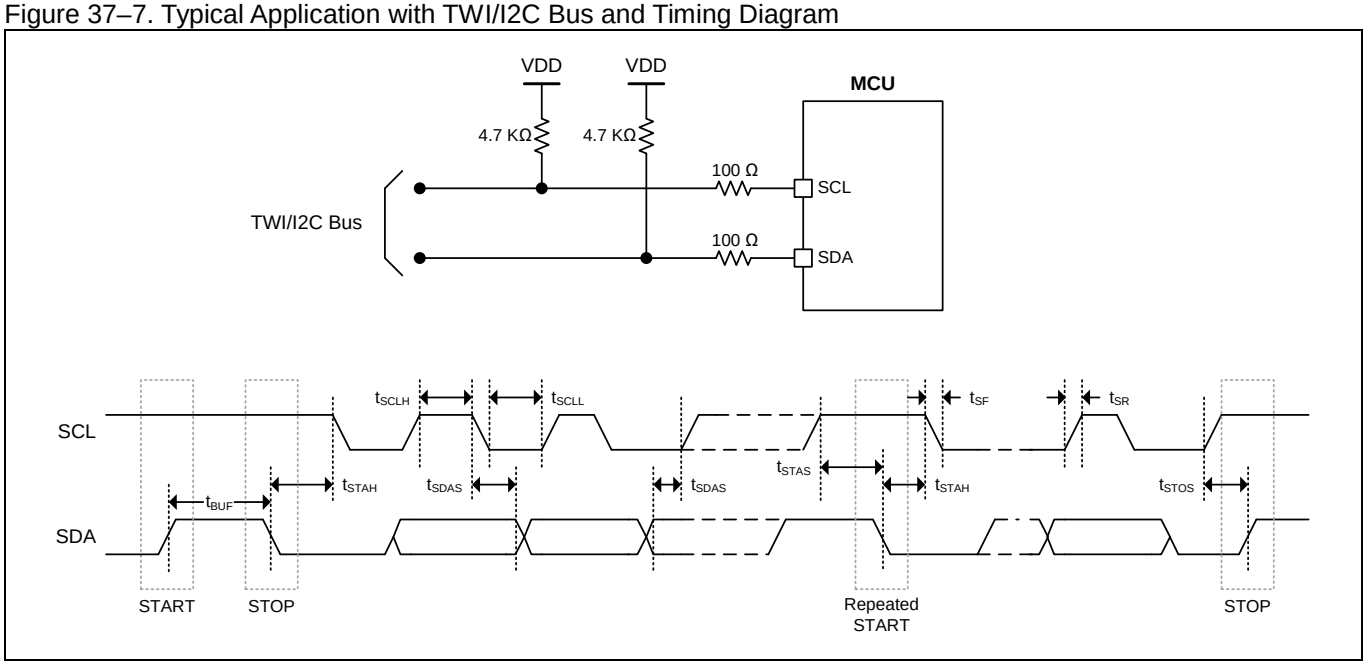


37.16. TWI/I2C Timing Characteristics

VDD = 5.0V±10%, VSS = 0V, T_A = -40°C to +105°C, unless otherwise specified

Symbol	Parameter	Min	Max	Units
Master Mode Timing				
t_{BUF}	Bus Free Time between STOP and START Condition			us
t_{STAH}	Hold Time after (Repeated) START Condition			us
t_{STAS}	Repeated START Condition Setup Time			us
t_{STOS}	STOP Condition Setup Time			us
t_{SCLH}	SCL High Time			T_{SYSCLK}
t_{SCLL}	SCL Low Time			T_{SYSCLK}
t_{SDAS}	SDA Data Setup Time			ns
t_{SDAH}	SDA Data Hold Time			ns
t_{SR}	SCL/SDA Rise Time		1000 ~ 300	ns
t_{SF}	SCL/SDA Fall Time			us
C_L	Capacitive Load for SCL and SDA	400	400	pF

Figure 37–7. Typical Application with TWI/I2C Bus and Timing Diagram



38. Instruction Set

Table 38–1. Instruction Set

MNEMONIC	DESCRIPTION	BYTE	EXECUTION Cycles
DATA TRASFER			
MOV A,Rn	Move register to Acc	1	1
MOV A,direct	Move direct byte o Acc	2	2
MOV A,@Ri	Move indirect RAM to Acc	1	2
MOV A,#data	Move immediate data to Acc	2	2
MOV Rn,A	Move Acc to register		2
MOV Rn,direct	Move direct byte to register	2	4
MOV Rn,#data	Move immediate data to register	2	2
MOV direct,A	Move Acc to direct byte	2	3
MOV direct,Rn	Move register to direct byte	2	3
MOV direct,direct	Move direct byte to direct byte	3	4
MOV direct,@Ri	Move indirect RAM to direct byte	2	4
MOV direct,#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
MOV DPTR,#data16	Load DPTR with a 16-bit constant	3	3
MOVC A,@A+DPTR	Move code byte relative to DPTR to Acc	1	4
MOVC A,@A+PC	Move code byte relative to PC to Acc	1	4
MOVX A,@Ri	Move on-chip auxiliary RAM(8-bit address) to Acc	1	3
MOVX A,@DPTR	Move on-chip auxiliary RAM(16-bit address) to	1	3
MOVX @Ri,A	Move Acc to on-chip auxiliary RAM(8-bit	1	3
MOVX @DPTR,A	Move Acc to on-chip auxiliary RAM(16-bit	1	3
MOVX A,@Ri	Move external RAM(8-bit address) to Acc	1	3 ~ 12*Note1
MOVX A,@DPTR	Move external RAM(16-bit address) to Acc	1	3 ~ 12*Note1
MOVX @Ri,A	Move Acc to external RAM(8-bit address)	1	3 ~ 12*Note1
MOVX @DPTR,A	Move Acc to external RAM(16-bit address)	1	3 ~ 12*Note1
PUSH direct	Push direct byte onto Stack	2	4
POP direct	Pop direct byte from Stack	2	3
XCH A,Rn	Exchange register with Acc	1	3
XCH A,direct	Exchange direct byte with Acc	2	4
XCH A,@Ri	Exchange indirect RAM with Acc	1	4
XCHD A,@Ri	Exchange low-order digit indirect RAM with Acc	1	4
ARITHEMATIC OPERATIONS			
ADD A,Rn	Add register to Acc	1	2
ADD A,direct	Add direct byte to Acc	2	3
ADD A,@Ri	Add indirect RAM to Acc	1	3
ADD A,#data	Add immediate data to Acc	2	2
ADDC A,Rn	Add register to Acc with Carry	1	2
ADDC A,direct	Add direct byte to Acc with Carry	2	3
ADDC A,@Ri	Add indirect RAM to Acc with Carry	1	3
ADDC A,#data	Add immediate data to Acc with Carry	2	2
SUBB A,Rn	Subtract register from Acc with borrow	1	2
SUBB A,direct	Subtract direct byte from Acc with borrow	2	3
SUBB A,@Ri	Subtract indirect RAM from Acc with borrow	1	3

MNEMONIC	DESCRIPTION	BYTE	EXECUTION Cycles
SUBB A,#data	Subtract immediate data from Acc with borrow	2	2
INC A	Increment Acc	1	2
INC Rn	Increment register	1	3
INC direct	Increment direct byte	2	4
INC @Ri	Increment indirect RAM	1	4
DEC A	Decrement Acc	1	2
DEC Rn	Decrement register	1	3
DEC direct	Decrement direct byte	2	4
DEC @Ri	Decrement indirect RAM	1	4
INC DPTR	Increment DPTR	1	1
MUL AB	Multiply A and B	1	4
DIV AB	Divide A by B	1	5
DA A	Decimal Adjust Acc	1	4
LOGIC OPERATION			
ANL A,Rn	AND register to Acc	1	2
ANL A,direct	AND direct byte to Acc	2	3
ANL A,@Ri	AND indirect RAM to Acc	1	3
ANL A,#data	AND immediate data to Acc	2	2
ANL direct,A	AND Acc to direct byte	2	4
ANL direct,#data	AND immediate data to direct byte	3	4
ORL A,Rn	OR register to Acc	1	2
ORL A,direct	OR direct byte to Acc	2	3
ORL A,@Ri	OR indirect RAM to Acc	1	3
ORL A,#data	OR immediate data to Acc	2	2
ORL direct,A	OR Acc to direct byte	2	4
ORL direct,#data	OR immediate data to direct byte	3	4
XRL A,Rn	Exclusive-OR register to Acc	1	2
XRL A,direct	Exclusive-OR direct byte to Acc	2	3
XRL A,@Ri	Exclusive-OR indirect RAM to Acc	1	3
XRL A,#data	Exclusive-OR immediate data to Acc	2	2
XRL direct,A	Exclusive-OR Acc to direct byte	2	4
XRL direct,#data	Exclusive-OR immediate data to direct byte	3	4
CLR A	Clear Acc	1	1
CPL A	Complement Acc	1	2
RL A	Rotate Acc Left	1	1
RLC A	Rotate Acc Left through the Carry	1	1
RR A	Rotate Acc Right	1	1
RRC A	Rotate Acc Right through the Carry	1	1
SWAP A	Swap nibbles within the Acc	1	1
BOOLEAN VARIABLE MANIPULATION			
CLR C	Clear Carry	1	1
CLR bit	Clear direct bit	2	4
SETB C	Set Carry	1	1
SETB bit	Set direct bit	2	4
CPL C	Complement Carry	1	1
CPL bit	Complement direct bit	2	4
ANL C,bit	AND direct bit to Carry	2	3
MNEMONIC	**DESCRIPTION**	**BYTE**	**EXECUTION Cycles**

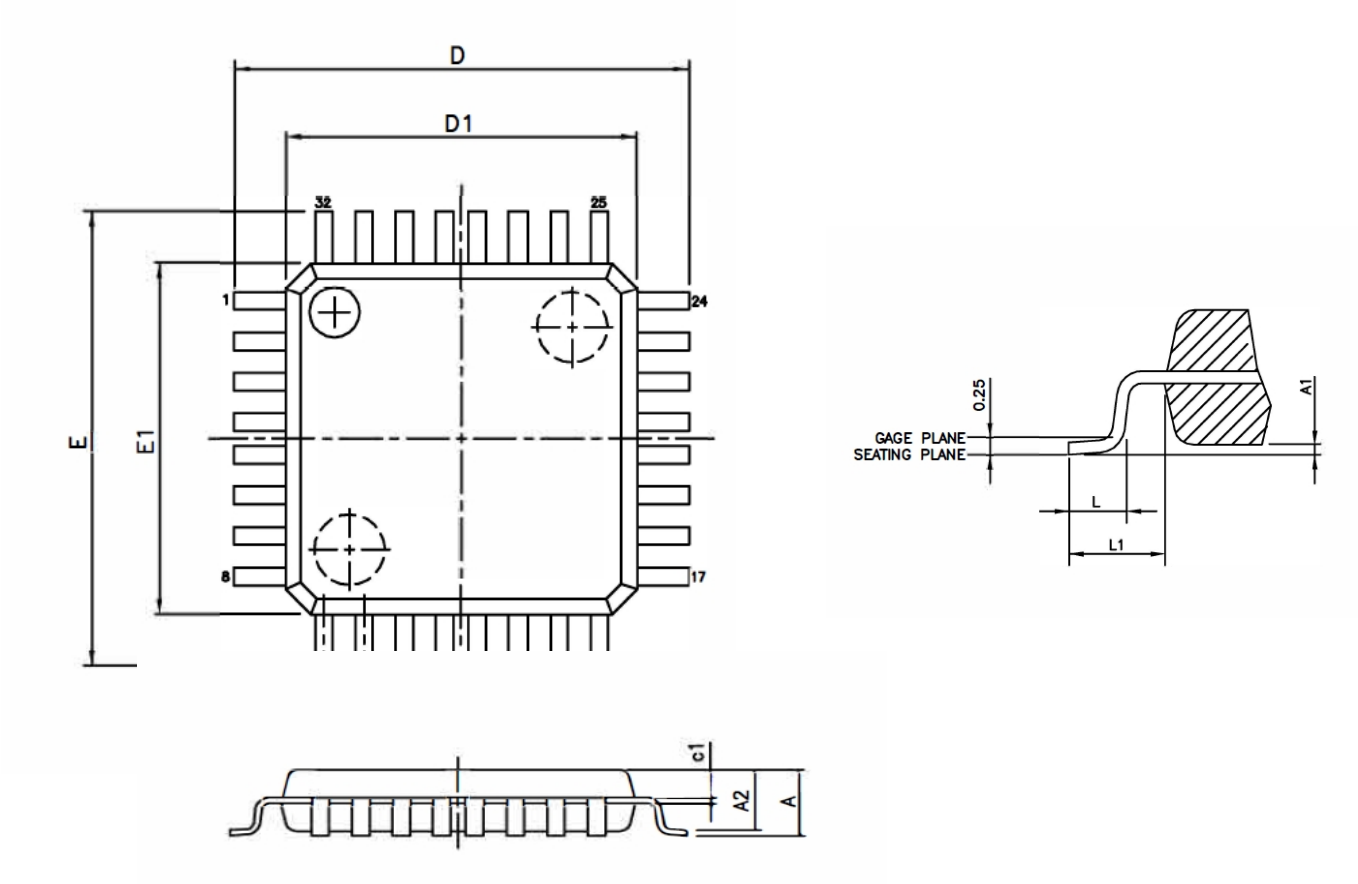
ANL C,/bit	AND complement of direct bit to Carry	2	3
ORL C,bit	OR direct bit to Carry	2	3
ORL C,/bit	OR complement of direct bit to Carry	2	3
MOV C,bit	Move direct bit to Carry	2	3
MOV bit,C	Move Carry to direct bit	2	4
BOOLEAN VARIABLE MANIPULATION			
JC rel	Jump if Carry is set	2	3
JNC rel	Jump if Carry not set	2	3
JB bit,rel	Jump if direct bit is set	3	4
JNB bit,rel	Jump if direct bit not set	3	4
JBC bit,rel	Jump if direct bit is set and then clear bit	3	5
PROAGRAM BRACHING			
ACALL addr11	Absolute subroutine call	2	6
LCALL addr16	Long subroutine call	3	6
RET	Return from subroutine	1	4
RETI	Return from interrupt subroutine	1	4
AJMP addr11	Absolute jump	2	3
LJMP addr16	Long jump	3	4
SJMP rel	Short jump	2	3
JMP @A+DPTR	Jump indirect relative to DPTR	1	3
JZ rel	Jump if Acc is zero	2	3
JNZ rel	Jump if Acc not zero	2	3
CJNE A,direct,rel	Compare direct byte to Acc and jump if not equal	3	5
CJNE A,#data,rel	Compare immediate data to Acc and jump if not	3	4
CJNE Rn,#data,rel	Compare immediate data to register and jump if	3	4
CJNE @Ri,#data,rel	Compare immediate data to indirect RAM and	3	5
DJNZ Rn,rel	Decrement register and jump if not equal	2	4
DJNZ direct,rel	Decrement direct byte and jump if not equal	3	5
NOP	No Operation	1	1

Note 1: The cycle time for access of external auxiliary RAM (EMB) is:
EMAI1 = 1: 3 + RW_Stretch + 2 x RWSH; (3~12)

39. Package Dimension

39.1. LQFP-32 (7mm X 7mm) Package Dimension

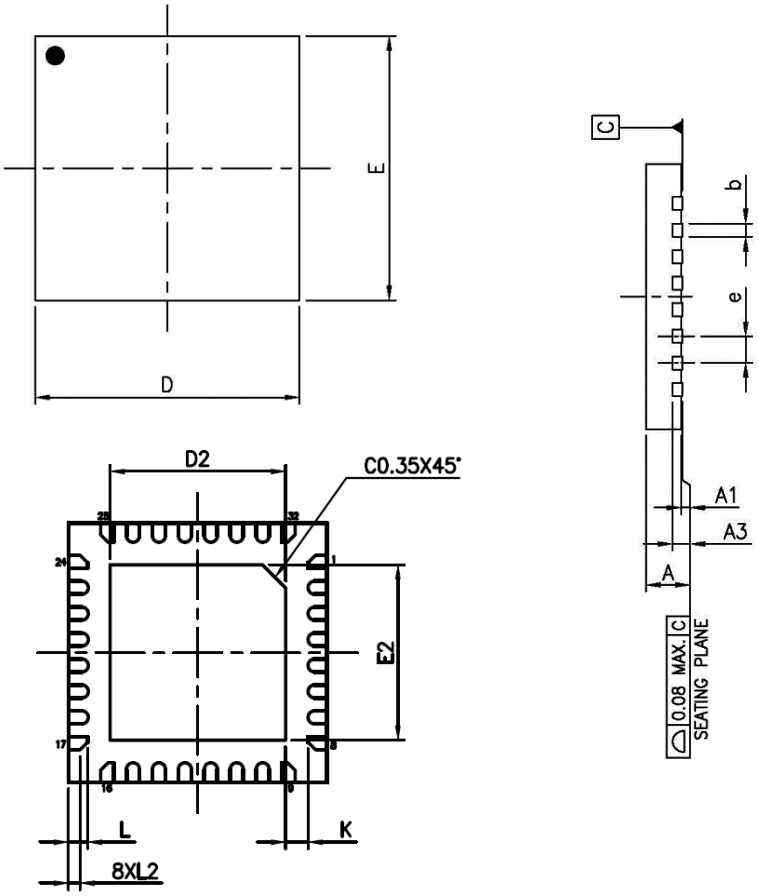
Figure 39-1. LQFP-32 (7mm X 7mm) Package Dimension



Unit	mm		inch	
Symbols mm	Min.	Max.	Min.	Max.
A	---	1.6	---	0.062
A1	0.05	0.15	0.000	0.005
A2	1.35	1.45	0.053	0.057
c1	0.09	0.16	0.003	0.006
D	9.00 BSC		0.354 BSC	
D1	7.00 BSC		0.275 BSC	
E	9.00 BSC		0.354 BSC	
E1	7.00 BSC		0.275 BSC	
e	0.8 BSC		0.0314 BSC	
b	0.30	0.45	0.011	0.017
L	0.45	0.75	0.017	0.029
L1	1REF		0.039REF	

39.2. QFN-32 (4mm X 4mm X 0.55mm) Dimension

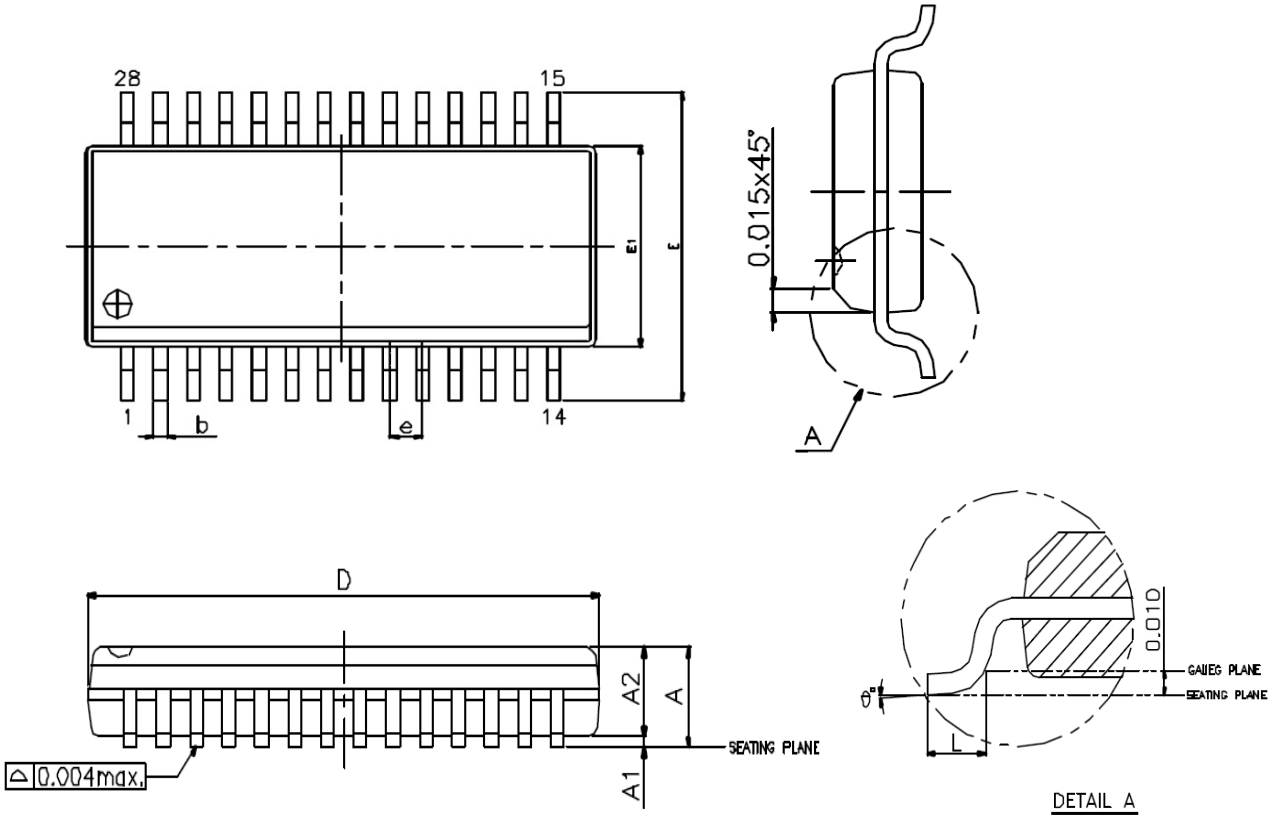
Figure 39-2. QFN-32 (4mm X 4mm X 0.55mm) Package Dimension



Unit	mm			inch		
JEDEC	N/A			N/A		
PKG CODE	UQFN(W432)			UQFN(W432)		
Symbols	Min.	Nom.	Max.	Min.	Nom.	Max.
A	0.50	0.55	0.60	0.020	0.022	0.024
A1	0.00	0.02	0.05	0.000	0.001	0.002
A3	0.150 REF.			0.006 REF.		
b	0.15	0.20	0.25	0.006	0.008	0.010
D	4.00 BSC			0.157 BSC		
E	4.00 BSC			0.157 BSC		
e	0.40 BSC			0.016 BSC		
L	0.25	0.30	0.35	0.010	0.012	0.014
L2	0.13	0.18	0.23	0.005	0.007	0.009
K	0.20	----	----	0.008	----	----
D2	2.65	2.70	2.75	0.104	0.106	0.108
E2	2.65	2.70	2.75	0.104	0.106	0.108

39.3. SSOP28 (150 mil) Package dimension

Figure 39-3. SSOP-28 (150 mil) Package dimension



Symbols	Dimensions in mm	
	Min.	Max.
A	1.346	1.752
A1	0.101	0.254
A2	----	1.498
b	0.203	0.304
D	9.804	10.007
E1	3.81	3.987
e	0.635 BASIC	
E	5.791	6.198
L	0.406	1.270
θ °	0	8

40. Revision History

Table 40–1. Revision History

Rev	Descriptions	Date
V0.20	Initial version preliminary released	2024/02/23
V0.21	1. Modify Fig 9-1 2. Fix CPU and MCKDO maximum frequency 3. Fix typo of Analog Input Only to Analog I/O 4. To add description of OCD and nRST in section 14.1 and 14.1.5 5. Fix CRC16 structure and add CRC32 structure.	2024/02/29
V0.22	1. Updated PGA block diagram. 2. Updated System block diagram. 3. Added I/O pin description in OPA block diagram section 27.1. 4. Modify the power mode description of the OPA in section 27.2.3. 5. Added description of OPnIG. 6. Fixed PGGAIN to PGGN. 7. Added ADMINS description. 8. Fixed OP1PIS and OP1NIS bit location in OP1CON0. 9. Delete S0CFG bit 5 description, and removed S0CFG1. 10. Removed TR4E, T4RLC, T4SC 11. Removed CCFn in PCAPWMN which have been moved to CCON. 12. Fixed typo of the C1PWMCR bit 6 in section 18.4.14 13. Modify description of GC and GC1 description. 14. To add description of SM20, SM21 15. Fixed typo in Figure 32-1 Flash memory configuration 16. Fixed typo in SnAGBRF description, SnIDT11~01 17. Fixed typo in ADCFG4 bit5~4 table, 1011 from P0ES to OP0ES 18. Fixed typo in AC1NIS, NVRL1 table 19. Removed PTM0 description in section 17.1 20. Fixed AC0NI Pin to P3.4 in Table 30-1, and IVR2.4V in Table 30-2 21. Fixed T1 pin assignment in section 16.1.6 22. Fixed Read/Write ability of AC1MOD in section 30.2 23. Fixed the CEX4 alternated pin from P44 to P32 in Figure 4-1 and Figure 4-3 24. Update OPA and PGA electrical characteristics. 25. To add descriptions of OP1Fr, OP0Fr, AC1Fr and AC0Fr 26. Modified external Reset capacitor and resistor description.	2024/06/04
V1.00	1. Fixed typo of OP1PIS0 2. Add description for IVR12, it needs to enable IVR24 for IVR12. 3. Modify the OPA offset trimming stable time to 600uS. 4. Fixed typo of the IVR voltage in the description of chapter 30 from 1.4V to 2.4V 5. Change description of AC0M[1:0] to improve readability. 6. Modified ADC conversion rate.	2024/09/09

41. Disclaimers

Herein, Megawin stands for "***Megawin Technology Co., Ltd.***"

Life Support — This product is not designed for use in medical, life-saving or life-sustaining applications, or systems where malfunction of this product can reasonably be expected to result in personal injury. Customers using or selling this product for use in such applications do so at their own risk and agree to fully indemnify Megawin for any damages resulting from such improper use or sale.

Right to Make Changes — Megawin reserves the right to make changes in the products - including circuits, standard cells, and/or software - described or contained herein in order to improve design and/or performance. When the product is in mass production, relevant changes will be communicated via an Engineering Change Notification (ECN).