MEGAWIN
MAKE YOU WIN

8051-Based MCU

MG82FG5D16
Data Sheet

Version: 0.75

Features

- 1-T 80C51 Central Processing Unit
- **MG82FG5D16** with **16K** Bytes flash ROM
  - ISP memory zone could be optioned as **0.5KB/1.0KB~7.5KB**
  - Flexible IAP size configured by software
  - Code protection for flash memory access
  - Flash write/erase cycle: 20,000
  - Flash data retention: 100 years at 25°C
  - **Default MG82FG5D16 Flash space mapping**
    - AP Flash default mapping (13.5KB, 0000h~35FFh)
    - IAP Flash default mapping (1.0KB, 3600h~39FFh)
    - ISP Flash default mapping (1.5KB, 3A00h~3FFFh), ISP Boot code
- Data RAM: **1K** Bytes
  - On-chip 256 bytes scratch-pad RAM
  - **768** bytes expanded RAM (XRAM) for **5D16**
  - Support page select on XRAM access in **5D16**
- Dual data pointer
- Interrupt controller
  - 14 sources, four-level-priority interrupt capability
  - Three external interrupt inputs, nINT0, nINT1 and nINT2, with glitch filter
  - All external interrupts support High/Low level or Rising/Falling edge trigger
- Total 6 timers in MG82FG5D16
  - RTC Timer and WDT Timer
  - Timer 0, Timer 1 and Timer 2
  - PCA0, Program Counter Array 0
  - If Timer 2 in split mode, then total 7 timers
- Three 16-bit timer/counters, Timer 0, Timer 1 and Timer 2
  - X12 mode and timer clock output function
  - New 5 operating modes in Timer 2 with 8 clock sources and 8 capture sources
  - Timer 2 can be split into two 8-bit timers
  - Clock Count Output (CCO) on T2CKO
  - All timers support PWM mode
- One Programmable 16-bit counter/timer Arrays (PCA0) with 6 CCP modules
  - PCA0 has 6 CCP (Capture/Compare/PWM) modules
  - Reloadable 16-bit base counter to support variable length PWM
  - Up to **100MHz** 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, each PCA 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, or
    - Up to 3 channels buffered 9~16-bit PWM
- 8 Inputs Keypad Interrupt
- 10-Bit Single-ended ADC
  - Programmable throughput up to 1M sps
  - 8 channel external inputs and one channel internal input (IVR/1.4V)
- Analog Comparator 0
  - Selectable internal voltage reference (IVR/1.4V) on ACNI0
  - 4 selectable ACPI0(+) inputs
  - Wake-up from power-down and idle
  - Glitch filter option and output to internal timer capture

- Enhanced UART (S0)
  - Framing Error Detection
  - Automatic Address Recognition
  - Speed improvement mechanism (X2/X4 mode), Max. UART baud rate up to 6MHz
  - Support SPI Master in Mode 4, up to 12MHz on SPICLK
- One Master/Slave SPI serial interface
  - Max. 12MHz on SPICLK
  - 8 bits data transfer
  - Up to 2 SPI masters including S0 in mode 4
  - Support daisy-chain function in SPI slave mode
- Two Master/Slave two wire serial interfaces: TWI0 and STWI (SID)
  - One Master/Slave hardware engine: TWI0
  - One software TWI, STWI, Start/Stop serial interface detection (SID)
- Programmable Watchdog Timer (WDT), clock sourced from ILRCO
  - One time enabled by CPU or power-on
  - To interrupt 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 or ILRCO
  - Programmable interrupt period from mini-second wakeup to monthly wakeup
  - 21-bit length system timer
- Beeper function
- On-Chip-Debug interface (OCD)
- Maximum 25 GPIOs in 28-pin package
  - 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 as open-drain output or push-pull output
  - P6.0, P6.1 and P4.7 shared with XTAL2, XTAL1 and RST
  - Programmable GPIO driving strength
  - On chip pull-up enable 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
- Two Brown-Out Detectors
  - BOD0: detect 1.7V
  - BOD1: selected detection level on 4.2V/3.7V/2.4V/2.0V
  - Interrupt CPU or reset CPU
  - Wake up CPU in Power-Down mode (BOD1)
- Multiple power 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
  - 10 Sources 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 crystal mode, 0 – 12MHz @ 2.0V – 5.5V, 0 – 25MHz @ 2.7V – 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
- Operating Temperature:
  - Industrial (-40°C to +105°C)*
- 16-Bytes Unique ID code
- Package Types:
  - SSOP28 (150 mil): MG82FG5D16AL28
  - SSOP20 (150 mil): MG82FG5D16AL20
  - QFN20 (3x3mm): MG82FG5D16AY20
  - SOP16: MG82FG5D16AS16

*: Tested by sampling.

List of Contents

List of Figures

List of Tables

1. General Description

The **MG82FG5D16** is a single-chip microcontroller based on a high performance 1-T architecture 80C51 CPU that executes instructions in 1~6 clock cycles (about 5~6 times fast then the standard 8051 device), and has an 8051 compatible instruction set. Therefore the **MG82FG5D16** can get the same performance by operating at a much lower speed to greatly reduce the power consumption.

The **MG82FG5D16** has **16K** bytes of embedded Flash memory for code and data. The Flash memory can be programmed either in serial writer mode (via ICP, In-Circuit Programming) or in In-System Programming mode. And, it also provides the In-Application Programming (IAP) capability. ICP and ISP that allows downloading new code without removing the microcontroller from the actual end product; IAP means that the device can write non-volatile data into the Flash memory while the application program is running. No need to provide external high voltage when the data writing into the Flash due to its built-in charge-pumping circuitry will provide it.

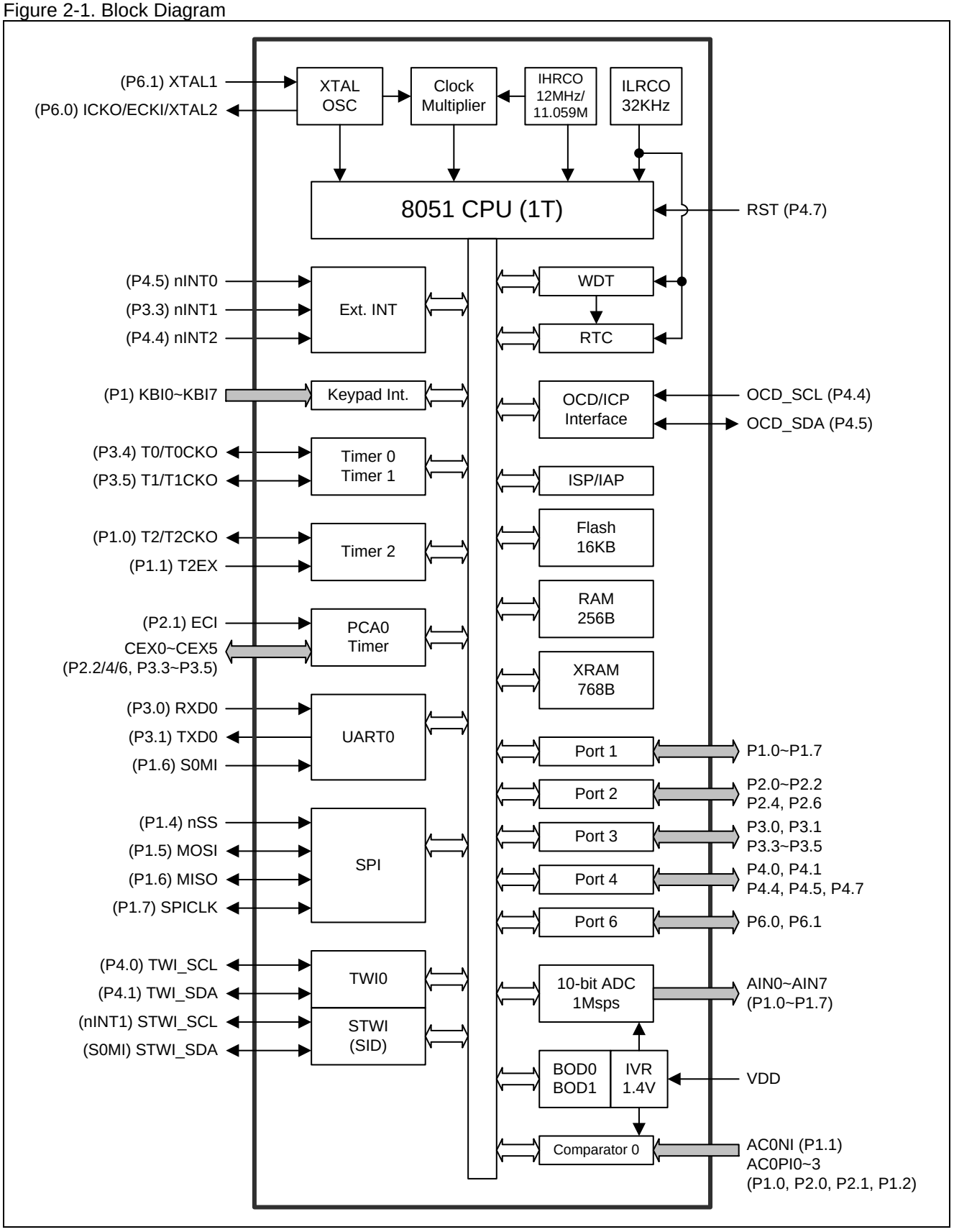
The **MG82FG5D16** retains all features of the standard 80C52 with 256 bytes of scratch-pad RAM, two external interrupts, a multi-source 4-level interrupt controller, a serial port (UART0) and three timer/counters. In addition, the **MG82FG5D16** has 25 I/O port pins, one XRAM of 768 bytes, one extra external interrupts with High/low trigger option, 1MHz 10-bit ADC, one analog comparators, one 6-channel PCA (PCA0), one 8 SPI, two TWIs (TWI0 and STWI), keypad interrupt, Watchdog Timer, Real-Time-Clock module, two Brown-out Detectors, 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 an enhanced serial function in UART0 that facilitates multiprocessor communication and a speed improvement mechanism (X2/X4 mode).

The **MG82FG5D16** 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 selecting sub-clock mode which its 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 **MG82FG5D16** 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. 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 1	*P6*	CH	CCAP0H	CCAP1H	CCAP2H	CCAP3H	CCAP4H	CCAP5H
F0	0 1	*B*	PAOE	PCAPWM0	PCAPWM1	PCAPWM2	PCAPWM3	PCAPWM4	PCAPWM5
E8	0 1	*P4*	CL	CCAP0L	CCAP1L	CCAP2L	CCAP3L	CCAP4L	CCAP5L
E0	0 1	*ACC*	WDTCR	IFD	IFADRH	IFADRL	IFMT	SCMD	ISPCR
D8	0 1	*CCON*	CMOD	CCAPM0	CCAPM1	CCAPM2	CCAPM3	CCAPM4	CCAPM5
D0	0 1	*PSW*	SIADR	SIDAT	SISTA	SICON	KBPATN	KBCON	KBMASK
C8	0 1	*T2CON*	T2MOD	RCAP2L	RCAP2H	TL2	TH2	CLRL	CHRL
C0	0	*XICON*	XICFG	--	ADCFG0	ADCON0	ADCDL	ADCDH	CKCON0
	1		XICFG1		ADCFG1				
	2		--		ADCFG2				
	3		--		ADCFG3				
B8	0 1	*IP0L*	SADEN	--	--	--	--	RTCCR	CKCON1
B0	0	*P3*	P3M0	P3M1	P4M0	--	--	RTCTM	IP0H
	1				--	--	P6M0		
	2				--	PDRVC0	--		
	3				--	PDRVC1	--		
A8	0 1	*IE*	SADDR	--	--	SFRPI	EIE1	EIP1L	EIP1H
A0	0	*P2*	AUXR0	AUXR1	AUXR2	AUXR3	--	--	--
	1					AUXR4			
	2					AUXR5			
	3					AUXR6			
	4					AUXR7			
98	0 1	*S0CON*	S0BUF			S0CFG	--	AC0CON	AC0MOD
90	0	*P1*	P1M0	P1M1	--	--	P2M0	BOREV	PCON1
	1			P2M1	T2MOD1	--	TREN0		
	2			P4M1	--	--	TRLC0		
	3			P6M1	--	--	TSPC0		
88	0 1	*TCON*	TMOD	TL0	TL1	TH0	TH1	SFIE	XRPS
80	0 1	--	SP	DPL	DPH	SPSTAT	SPCON	SPDAT	PCON0
		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 = xxxx-0000

7	6	5	4	3	2	1	0
--	--	--	--	PIDX3	PIDX2	PIDX1	PIDX0
W	**W**	**W**	**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. The available pages are only page **“0” to “F”**.

PIDX[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	BIT ADDRESS AND SYMBOL								RESET VALUE
			Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	
SP	Stack Pointer	81H	.7	.6	.5	.4	.3	.2	.1	.0	00000111
DPL	Data Pointer Low	82H	.7	.6	.5	.4	.3	.2	.1	.0	00000000
DPH	Data Pointer High	83H	.7	.6	.5	.4	.3	.2	.1	.0	00000000
SPSTAT	SPI Status Register	84H	SPIF	WCOL	THRF	SPIBSY	MODF	--	--	SPR2	00000xx0
SPCON	SPI Control Register	85H	SSIG	SPEN	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	00000100
SPDAT	SPI Data Register	86H	.7	.6	.5	.4	.3	.2	.1	.0	00000000
PCON0	Power Control 0	87H	SMOD1	SMOD0	--	POF0	GF1	GF0	PD	IDL	00010000
TCON	*Timer Control*	*88H*	*TF1*	*TR1*	*TF0*	*TR0*	*IE1*	*IT1*	*IE0*	*IT0*	*00000000*
TMOD	Timer Mode	89H	T1GATE	T1C/T	T1M1	T1M0	T0GATE	T0C/T	T0M1	T0M0	00000000
TL0	Timer Low 0	8AH	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TL1	Timer Low 1	8BH	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TH0	Timer High 0	8CH	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TH1	Timer High 1	8DH	.7	.6	.5	.4	.3	.2	.1	.0	00000000
SFIE	System Flag INT En.	8EH	SIDFIE	MCDRE	MCDFIE	RTCFIE	--	BOF1IE	BOF0IE	WDTFIE	0110x000
XRPS	XRAM Page Select	8FH	--	--	--	--	--	--	XRPS.1	XRPS.0	xxxxxx00
P1	*Port 1*	*90H*	*P1.7*	*P1.6*	*P1.5*	*P1.4*	*P1.3*	*P1.2*	*P1.1*	*P1.0*	*11111111*
P1M0	P1 Mode Register 0	91H	P1M0.7	P1M0.6	P1M0.5	P1M0.4	P1M0.3	P1M0.2	P1M0.1	P1M0.0	00000000
P1M1	P1 Mode Register 1	92H	P1M1.7	P1M1.6	P1M1.5	P1M1.4	P1M1.3	P1M1.2	P1M1.1	P1M1.0	11111111
P2M1	P2 Mode Register 1	92H	--	P2M1.6	--	P2M1.4	--	P2M1.2	P2M1.1	P2M1.0	x1x1x111
P4M1	P4 Mode Register 1	92H	P4M1.7	--	P4M1.5	P4M1.4	--	--	P4M1.1	P4M1.0	1x11xx11
P6M1	P6 Mode Register 1	92H	--	--	--	--	--	--	P6M1.1	P6M1.0	xxxxxx11
T2MOD1	Timer2 mode 1 Reg.	93H	TL2CS	TF2IG	TL2IS	T2CKS	T2MS1	CP2S2	CP2S1	CP2S0	00000000
P2M0	P2 Mode Register 0	95H	--	P2M0.6	--	P2M0.4	--	P2M0.2	P2M0.1	P2M0.0	x0x0x000
TREN0	Timer Run Enable Register 0	95H	--	--	TR2LE	--	--	TR2E	TR1E	TR0E	00000000
TRLC0	Timer Reload Control Register 0	95H	--	--	TL2RLC	--	--	T2RLC	T1RLC	T0RLC	00000000
TSPC0	Timer Stop Control Register 0	95H	--	--	TL2SC	--	--	T2SC	T1SC	T0SC	00000000
BOREV	Bit Order Reversed	96H	.7	.6	.5	.4	.3	.2	.1	.0	00000000
PCON1	Power Control 1	97H	SWRF	EXRF	MCDF	RTCF	--	BOF1	BOF0	WDTF	0000x000
S0CON	*Serial 0 Control*	*98H*	*SM00 /FE*	*SM10*	*SM20*	*REN0*	*TB80*	*RB80*	*TI0*	*RI0*	*00000000*
S0BUF	Serial 0 Buffer	99H	.7	.6	.5	.4	.3	.2	.1	.0	xxxxxxxx
S0CFG	S0 Configuration	9CH	GF	SMOD2	URM0X3	SM30	S0DOR	BTI	UTIE	--	x000100x
AC0CON	AC0 Control Reg.	9EH	AC0LP	AC0PDX	AC0OUT	AC0F	AC0EN	AC0INV	AC0M1	AC0M0	00x00000
AC0MOD	AC0 Mode Reg.	9FH	NVRS3	NVRS2	NVRS1	NVRS0	--	AC0FLT	AC0PIS1	AC0PIS0	0000x000
P2	*Port 2*	*A0H*	--	*P2.6*	--	*P2.4*	--	*P2.2*	*P2.1*	*P2.0*	*x1x1x111*
AUXR0	Auxiliary Register 0	A1H	P60OC1	P60OC0	P60FD	--	--	--	INT1H	INT0H	000xxx00
AUXR1	Auxiliary Register 1	A2H	--	--	--	--	--	--	--	DPS	xxxxxxx0
AUXR2	Auxiliary Register 2	A3H	STAF	STOF	--	--	T1X12	T0X12	T1CKOE	T0CKOE	00xx0000
AUXR3	Auxiliary Register 3	A4H	T0PS1	T0PS0	BPOC1	BPOC0	S0PS0	TWIPS1	TWIPS0	T0XL	00000000
AUXR4	Auxiliary Register 4	A4H	T2PS1	T2PS0	T1PS1	T1PS0	SPIPS0	--	AC0OE	AC0FLT1	00000x00
AUXR5	Auxiliary Register 5	A4H	C0IC4S0	C0IC2S0	C0PPS1	C0PPS0	C0PS1	C0PS0	ECIPS0	C0COPS	00000000
AUXR6	Auxiliary Register 5	A4H	KBI4PS1	KBI4PS0	KBI6PS0	KBI2PS0	KBI0PS0	--	S0MIPS	S0COPS	00000x00
AUXR7	Auxiliary Register 7	A4H	POE5	POE4	C0CKOE	SPI0M0	--	--	--	--	1100xxxx
IE	*Interrupt Enable*	*A8H*	*EA*	*GF4*	*ET2*	*ES0*	*ET1*	*EX1*	*ET0*	*EX0*	*0x000000*
SADDR	Slave Address	A9H	.7	.6	.5	.4	.3	.2	.1	.0	00000000
SFRPI	SFR Page Index	ACH	--	--	--	--	IDX3	IDX2	IDX1	IDX0	xxxx0000
EIE1	Extended INT Enable 1	ADH	EAC0	ETWI0	EKB	--	ESF	EPCA	EADC	ESPI	000x0000
EIP1L	Ext. INT Priority 1 Low	AEH	PAC0L	PTWI0L	PKBL	--	PSFL	PPCAL	PADCL	PSPIL	000x0000
EIP1H	Ext. INT Priority 1 High	AFH	PAC0H	PTWI0H	PKBH	--	PSFH	PPCAH	PADCH	PSPIH	000x0000
P3	*Port 3*	*B0H*	--	--	*P3.5*	*P3.4*	*P3.3*	--	*P3.1*	*P3.0*	*xx111x11*
P3M0	P3 Mode Register 0	B1H	--	--	P3M0.5	P3M0.4	P3M0.3	--	P3M0.1	P3M0.0	xx000x00
P3M1	P3 Mode Register 1	B2H	--	--	P3M1.5	P3M1.4	P3M1.3	--	P3M1.1	P3M1.0	xx000x00
P4M0	P4 Mode Register 0	B3H	P4M0.7	--	P4M0.5	P4M0.4	--	--	P4M0.1	P4M0.0	0x00xx00
PDRVC0	Port Driving Control 0	B4H	P3DC1	P3DC0	P2DC1	P2DC0	P1DC1	P1DC0	--	--	000000xx
PDRVC1	Port Driving Control 1	B4H	--	--	--	--	--	--	P4DC1	P4DC0	xxxxxx00
P6M0	P6 Mode Register 0	B5H	--	--	--	--	--	--	P6M0.1	P6M0.0	xxxxxx00
RTCTM	RTC Timer Register	B6H	RTCCS1	RTCCS0	RTCCT5	RTCCT4	RTCCT3	RTCCT2	RTCCT1	RTCCT0	01111111
IP0H	Interrupt Priority 0 High	B7H	--	PX2H	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	x0000000
IP0L	*Interrupt Priority Low*	*B8H*	--	*PX2L*	*PT2L*	*PSL*	*PT1L*	*PX1L*	*PT0L*	*PX0L*	*x0000000*
SADEN	Slave Address Mask	B9H	.7	.6	.5	.4	.3	.2	.1	.0	00000000

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	
RTCCR	RTC Control Reg.	BEH	RTCE	RTCO	RTCRL5	RTCRL4	RTCRL3	RTCRL2	RTCRL1	RTCRL0	00111111
CKCON1	Clock Control 1	BFH	XTOR	--	XCKS5	XCKS4	XCKS3	XCKS2	XCKS1	XCKS0	0x001011
XICON	*External INT Control*	*C0H*	--	--	--	--	*INT2H*	*EX2*	*IE2*	*IT2*	*xxxx0000*
XICFG	Ext. INT. Configure	C1H	INT1IS1	INT1IS0	INT0IS1	INT0IS0	--	X2FLT	X1FLT	X0FLT	0000x000
XICFG1	Ext. INT. Configure 1	C1H	INT1IS2	INT0IS2	INT2IS1	INT2IS0	--	X2FLT1	X1FLT1	X0FLT1	0000x000
ADCFG0	ADC Configuration 0	C3H	ADCKS2	ADCKS1	ADCKS0	ADRJ	--	--	ADTM1	ADTM0	0000xx00
ADCFG1	ADC Configuration 1	C3H	--	--	--	SIGN	AOS.3	AOS.2	AOS.1	AOS.0	xxx00000
ADCFG2	ADC Configuration 2	C3H	SHT.7	SHT.6	SHT.5	SHT.4	SHT.3	SHT.2	SHT.1	SHT.0	00000000
ADCFG3	ADC Configuration 3	C3H	ADPS1	ADPS0	HA	--	--	--	--	--	010xxxxx
ADCON0	ADC Control 0	C4H	ADCEN	--	CHS3	ADCI	ADCS	CHS2	CHS1	CHS0	0x000000
ADCDL	ADC Data Low	C5H	ADCV.1	ADCV.0	--	--	--	--	--	--	xxxxxxxx
ADCDH	ADC Data High	C6H	ADCV.9	ADCV.8	ADCV.7	ADCV.6	ADCV.5	ADCV.4	ADCV.3	ADCV.2	xxxxxxxx
CKCON0	Clock Control 0	C7H	AFS	ENCKM	CKMIS1	CKMIS0	CCKS	SCKS2	SCKS1	SCKS0	00010000
T2CON	*Timer 2 Control Reg.*	*C8H*	*TF2*	*EXF2*	*RCLK/ TF2L*	*TCLK/ TL2IE*	*EXEN2*	*TR2*	*C/T2*	*CP/RL2*	*00000000*
T2MOD	Timer2 mode Reg.	C9H	T2SPL	TL2X12	T2EXH	T2X12	TR2L	TR2LC	T2OE	T2MS0	00000000
RCAP2L	Timer2 Capture Low	CAH	.7	.6	.5	.4	.3	.2	.1	.0	00000000
RCAP2H	Timer2 Capture High	CBH	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TL2	Timer Low 2	CCH	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TH2	Timer High 2	CDH	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CLRL	CL Reload register	CEH	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CHRL	CH Reload register	CFH	.7	.6	.5	.4	.3	.2	.1	.0	00000000
PSW	*Program Status Word*	*D0H*	*CY*	*AC*	*F0*	*RS1*	*RS0*	*OV*	*F1*	*P*	*00000000*
SIADR	TWI0 Address Reg.	D1H	.7	.6	.5	.4	.3	.2	.1	GC	00000000
SIDAT	TWI0 Data Reg.	D2H	.7	.6	.5	.4	.3	.2	.1	.0	00000000
SISTA	TWI0 Status Reg.	D3H	SIS7	SIS6	SIS5	SIS4	SIS3	SIS2	SIS1	SIS0	11111000
SICON	TWI0 Control Reg.	D4H	CR2	ENSI	STA	STO	SI	AA	CR1	CR0	00000000
KBPATN	Keypad Pattern	D5H	.7	.6	.5	.4	.3	.2	.1	.0	11111111
KBCON	Keypad Control	D6H	KBCS1	KBCS0	--	--	--	--	PATN_ SEL	KBIF	00xxxx01
KBMASK	Keypad Int. Mask	D7H	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCON	*PCA0 Control Reg.*	*D8H*	*CF*	*CR*	*CCF5*	*CCF4*	*CCF3*	*CCF2*	*CCF1*	*CCF0*	*00000000*
CMOD	PCA0 Mode Reg.	D9H	CIDL	BME4	BME2	BME0	CPS2	CPS1	CPS0	ECF	00000000
CCAPM0	PCA0 Module0 Mode	DAH	--	ECOM0	CAPP0	CAPN0	MAT0	TOG0	PWM0	ECCF0	x0000000
CCAPM1	PCA0 Module1 Mode	DBH	--	ECOM1	CAPP1	CAPN1	MAT1	TOG1	PWM1	ECCF1	x0000000
CCAPM2	PCA0 Module2 Mode	DCH	--	ECOM2	CAPP2	CAPN2	MAT2	TOG2	PWM2	ECCF2	x0000000
CCAPM3	PCA0 Module3 Mode	DDH	--	ECOM3	CAPP3	CAPN3	MAT3	TOG3	PWM3	ECCF3	x0000000
CCAPM4	PCA0 Module4 Mode	DEH	--	ECOM4	CAPP4	CAPN4	MAT4	TOG4	PWM4	ECCF4	x0000000
CCAPM5	PCA0 Module5 Mode	DFH	--	ECOM5	CAPP5	CAPN5	MAT5	TOG5	PWM5	ECCF5	x0000000
ACC	*Accumulator*	*E0H*	*ACC.7*	*ACC.6*	*ACC.5*	*ACC.4*	*ACC.3*	*ACC.2*	*ACC.1*	*ACC.0*	*00000000*
WDTCR	WDT Control register	E1H	WREN	NSW	ENW	CLRW	WIDL	PS2	PS1	PS0	00000000
IFD	ISP Flash data	E2H	.7	.6	.5	.4	.3	.2	.1	.0	11111111
IFADRH	ISP Flash Addr. High	E3H	.7	.6	.5	.4	.3	.2	.1	.0	00000000
IFADRL	ISP Flash Addr. Low	E4H	.7	.6	.5	.4	.3	.2	.1	.0	00000000
IFMT	ISP Mode Table	E5H	--	--	--	--	--	MS.2	MS.1	MS.0	xxxxx000
SCMD	ISP Serial Command	E6H	.7	.6	.5	.4	.3	.2	.1	.0	xxxxxxxx
ISPCR	ISP Control Register	E7H	ISPEN	SWBS	SRST	CFAIL	--	--	--	--	0000xxxx
P4	*Port 4*	*E8H*	*P4.7*	--	*P4.5*	*P4.4*	--	--	*P4.1*	*P4.0*	*1x11xx11*
CL	PCA0 base timer Low	E9H	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP0L	PCA0 module0 capture Low	EAH	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP1L	PCA0 module1 capture Low	EBH	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP2L	PCA0 module2 capture Low	ECH	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP3L	PCA0 module3 capture Low	EDH	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP4L	PCA0 module4 capture Low	EEH	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP5L	PCA0 module5 capture Low	EFH	.7	.6	.5	.4	.3	.2	.1	.0	00000000
B	*B Register*	*F0H*	*B.7*	*B.6*	*B.5*	*B.4*	*B.3*	*B.2*	*B.1*	*B.0*	*00000000*
PAOE	PWM Additional Output Enable	F1H	POE3	POE2B	POE2A	POE2	POE1	POE0B	POE0A	POE0	10011001
PCAPWM0	PCA0 PWM0 Mode	F2H	P0RS1	P0RS0	--	--	--	P0INV	ECAP0H	ECAP0L	00xxx000
PCAPWM1	PCA0 PWM1 Mode	F3H	P1RS1	P1RS0	--	--	--	P1INV	ECAP1H	ECAP1L	00xxx000
PCAPWM2	PCA0 PWM2 Mode	F4H	P2RS1	P2RS0	--	--	--	P2INV	ECAP2H	ECAP2L	00xxx000
PCAPWM3	PCA0 PWM3 Mode	F5H	P3RS1	P3RS0	--	--	--	P3INV	ECAP3H	ECAP3L	00xxx000
PCAPWM4	PCA0 PWM4 Mode	F6H	P4RS1	P4RS0	--	--	--	P4INV	ECAP4H	ECAP4L	00xxx000

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	
PCAPWM5	PCA0 PWM5 Mode	F7H	P5RS1	P5RS0	--	--	--	P5INV	ECAP5H	ECAP5L	00xxx000
P6	*Port 6*	*F8H*	--	--	--	--	--	--	*P6.1*	*P6.0*	*xxxxxx11*
CH	PCA0 base timer High	F9H	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP0H	PCA0 Module0 capture High	FAH	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP1H	PCA0 Module1 capture High	FBH	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP2H	PCA0 Module2 capture High	FCH	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP3H	PCA0 Module3 capture High	FDH	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP4H	PCA0 Module4 capture High	FEH	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP5H	PCA0 Module5 capture High	FFH	.7	.6	.5	.4	.3	.2	.1	.0	00000000

3.3. Auxiliary SFR Map (Page P)

MG82FG5D16 has an auxiliary SFR page which is indexed by page P and its data write method is different from the standard 8051. 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 **8 physical bytes** and **8 logical bytes**. The 8 physical bytes include IAPLB, CKCON2, CKCON3, CKCON4, PCON2, PCON3, SPCON0 and DCON0. The 8 logical bytes include PCON0, PCON1, CKCON0, CKCON1, WDTCR, P4, P6 and RTCCR. **Access** on the 8 logical bytes gets the coherence content with the same SFR in Page 0~F. Please refer Section "27 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		--	--	--	--	--	--	RTCCR	CKCON1
B0		--	--	--	--	--	--	--	--
A8		--	--	--	--	--	--	--	--
A0		--	--	--	--	--	--	--	--
98		--	--	--	--	--	--	--	--
90		--	--	--	--	--	--	--	PCON1
88		--	--	--	--	--	--	--	
80		--	--	--	--	--	--	--	PCON0
78		--	--	--	--	--	--	--	--
70		--	--	--	--	--	--	--	--
68		--	--	--	--	--	--	--	--
60		--	--	--	--	--	--	--	--
58		--	--	--	--	--	--	--	--
50		--	--	--	--	--	--	--	--
48		SPCON0	--	--	--	DCON0	--	--	--
40		CKCON2	CKCON3	CKCON4	--	PCON2	PCON3	--	--
38		--	--	--	--	--	--	--	--
30		--	--	--	--	--	--	--	--
28		--	--	--	--	--	--	--	--
20		--	--	--	--	--	--	--	--
18		--	--	--	--	--	--	--	--
10		--	--	--	--	--	--	--	--
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	_
CKCON2	Clock Control 2	40H	XTGS1	XTGS0	XTALE	IHRCOE	MCKS1	MCKS0	OSCS1	OSCS0	01010000
CKCON3	Clock Control 3	41H	WDTCS1	WDTCS0	--	WDTFS	MCKD1	MCKD0	**1**	**0**	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	--	--	--	--	--	--	--	0xxxxxxx
SPCON0	SFR Page Control 0	48H	RTCCTL	P6CTL	P4CTL	WRCTL	CKCTL1	CKCTL0	PWCTL1	PWCTL0	00000000
DCON0	Device Control 0	4CH	HSE	IAPO	--	--	--	IORCTL	RSTIO	OCDE	10xxx011
Logical Bytes											
PCON0	Power Control 0	87H	SMOD1	SMOD0	--	POF0	GF1	GF0	PD	IDL	00010000
PCON1	Power Control 1	97H	SWRF	EXRF	MCDF	RTCF	--	BOF1	BOF0	WDTF	0000x000
RTCCR	RTC Control Register	BEH	RTCE	RTCO	RTCRL.5	RTCRL.4	RTCRL.3	RTCRL.2	RTCRL.1	RTCRL.0	00111111
CKCON1	Clock Control 1	BFH	XTOR	--	XCKS5	XCKS4	XCKS3	XCKS2	XCKS1	XCKS0	0x001011
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	--	P4.5	P4.4	--	--	P4.1	P4.0	1x11xx11
P6	Port 6	F8H	--	--	--	--	--	--	P6.1	P6.0	xxxxxx11

4. Pin Configurations

4.1. Package Instruction

Figure 4-1. SSOP28 Top View

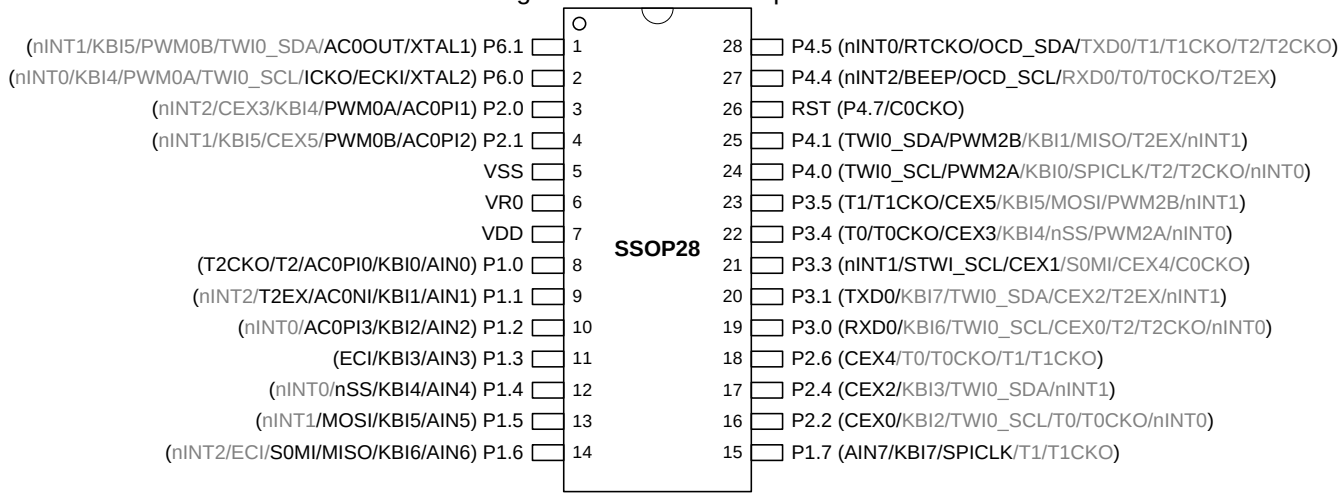


Figure 4-2. SSOP20 Top View

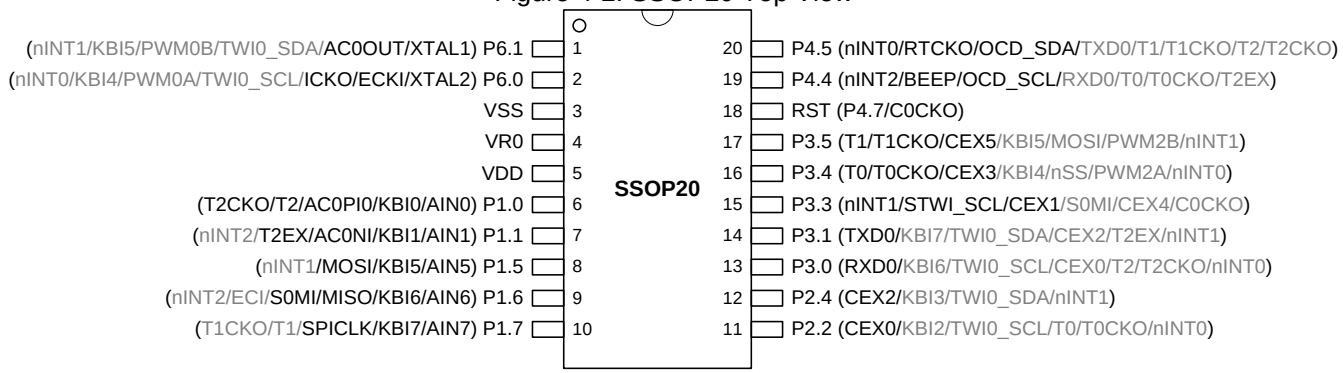


Figure 4-3. QFN20 Top View

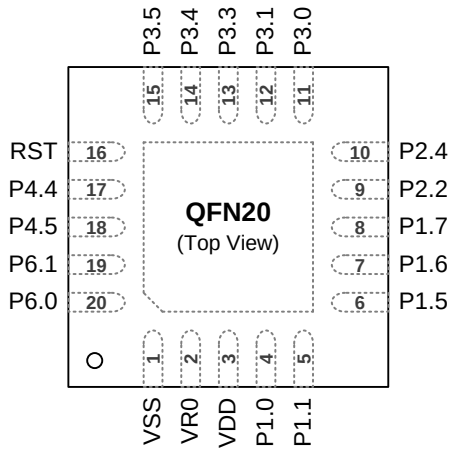
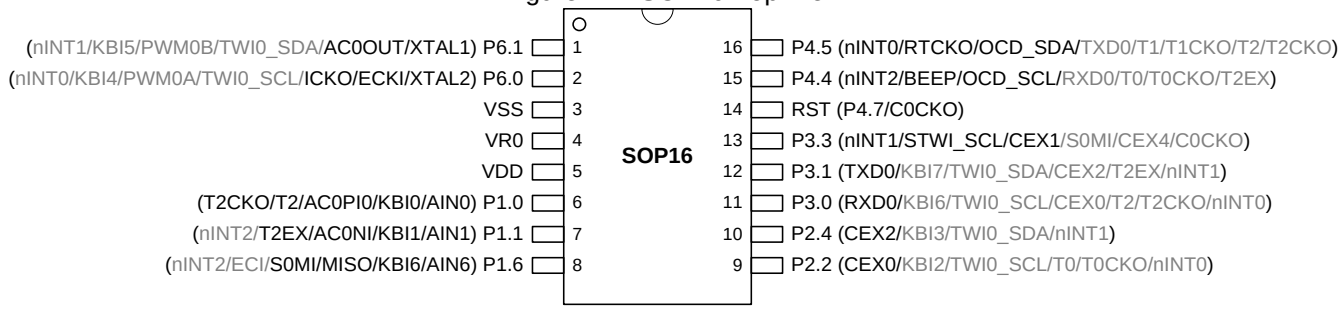


Figure 4-4. SOP16 Top View



4.2. Pin Description

Table 4-1. Pin Description

MNEMONIC	PIN NUMBER				I/O TYPE	DESCRIPTION
	28-Pin SSOP	20-Pin SSOP	20-Pin QFN20	16-Pin SOP		
P1.0 (AIN0) (KBI0) (AC0PI0) (T2) (T2CKO)	8	6	4	6	I/O	* Port 1.0. * AIN0: ADC channel-0 analog input. * KBI0: keypad input 0. * AC0PI0: Analog Comparator 0 positive input channel 0. * T2: Timer/Counter 2 external clock input. * T2CKO: Timer 2 programmable clock output.
P1.1 (AIN1) (KBI1) (AC0NI) (T2EX)	9	7	5	7	I/O	* Port 1.1. * AIN1: ADC channel-1 analog input. * KBI1: keypad input 1. * AC0NI: Analog Comparator 0 negative input. * T2EX: Timer/Counter 2 external control input.
P1.2 (AIN2) (KBI2) (AC0PI3)	10	--	--	--	I/O	* Port 1.2. * AIN2: ADC channel-2 analog input. * KBI2: keypad input 2. * AC0PI3: Analog Comparator 0 positive input channel 3.
P1.3 (AIN3) (KBI3) (ECI)	11	--	--	--	I/O	* Port 1.3. * AIN3: ADC channel-3 analog input. * KBI3: keypad input 3. * ECI: PCA0 external clock input.
P1.4 (AIN4) (KBI4) (nSS)	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)	13	8	6	--	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) (S0MI)	14	9	7	8	I/O	* Port 1.6. * AIN6: ADC channel-6 analog input. * KBI6: keypad input 6. * MISO: SPI master in & slave out. * S0MI: Serial Port 0 SPI Master mode data Input.
P1.7 (AIN7) (KBI7) (SPICLK)	15	10	8	--	I/O	* Port 1.7. * AIN7: ADC channel-7 analog input. * KBI7: keypad input 7. * SPICLK: SPI clock, output for master and input for slave.
P2.0 (AC0PI1) (PWM0A)	3	--	--	--	I/O	* Port 2.0. * AC0PI1: Analog Comparator 0 positive input channel 1. * PWM0A: PCA0 PWM0 output sub-channel A.
P2.1 (AC0PI2) (PWM0B)	4	--	--	--	I/O	* Port 2.1. * AC0PI2: Analog Comparator 0 positive input channel 2. * PWM0B: PCA0 PWM0 output sub-channel B.
P2.2 (CEX0)	16	11	9	9	I/O	* Port 2.2. * CEX0: PCA0 module-0 external I/O.
P2.4 (CEX2)	17	12	10	10	I/O	* Port 2.4. * CEX2: PCA0 module-2 external I/O.
P2.6 (CEX4)	18	--	--	--	I/O	* Port 2.6. * CEX4: PCA0 module-4 external I/O.
P3.0 (RXD0)	19	13	11	11	I/O	* Port 3.0. * RXD0: UART0 serial input port.
P3.1 (TXD0)	20	14	12	12	I/O	* Port 3.1. * TXD0: UART0 serial output port.
P3.3 (nINT1) (STWI_SCL) (CEX1)	21	15	13	13	I/O	* Port 3.3. * nINT1: external interrupt 1 input. * STWI_SCL: Software TWI serial clock. * CEX1: PCA0 module-1 external I/O.

MNEMONIC	PIN NUMBER				I/O TYPE	DESCRIPTION
	28-Pin SSOP	20-Pin SSOP	20-Pin QFN20	16-Pin SOP		
P3.4 (T0) (T0CKO) (CEX3)	22	16	14	--	I/O	* Port 3.4. * T0: Timer/Counter 0 external input. * T0CKO: Timer 0 programmable clock output. * CEX3: PCA0 module-3 external I/O.
P3.5 (T1) (T1CKO) (CEX5)	23	17	15	--	I/O	* Port 3.5. * T1: Timer/Counter 1 external input. * T1CKO: Timer 1 programmable clock output. * CEX5: PCA0 module-5 external I/O.
P4.0 (TWI0_SCL) (PWM2A)	24	--	--	--	I/O	* Port 4.0. * TWI0_SCL: serial clock of TWI0. * PWM2A: PCA0 PWM2 output sub-channel A.
P4.1 (TWI0_SDA) (PWM2B)	25	--	--	--	I/O	* Port 4.1. * TWI0_SDA: serial data of TWI0. * PWM2B: PCA0 PWM2 output sub-channel B.
P4.4 (nINT2) (BEEP) (OCD_SCL)	27	19	17	15	I/O	* Port 4.4. * nINT2: external interrupt 2 input. * BEEP: Beeper output. * OCD_SCL: OCD interface, serial clock.
P4.5 (nINT0) (RTCKO) (OCD_SDA)	28	20	18	16	I/O	* Port 4.5. * nINT0: external interrupt 0 input. * OCD_SDA: OCD interface, serial data. * RTCKO: RTC programmable clock output.
P6.0 (XTAL2) (ECKI) (ICKO)	2	2	20	2	I/O 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.
P6.1 (XTAL1) (AC0OUT)	1	1	19	1	I/O	* Port 6.1. * XTAL1: Input of on-chip crystal oscillating circuit. * AC0OUT: Analog Comparator 0 output.
RST (P4.7) (C0CKO)	26	18	16	14	I I/O	* RST: External RESET input, high active. * Port 4.7. * C0CKO: Programmable clock output of PCA0 base counter.
VR0	6	4	2	4	I/O	* VR0. Voltage Reference 0. Connect 0.1uF and 4.7uF to VSS.
VDD	7	5	3	5	P	Power supply input.
VSS	5	3	1	3	G	Ground, 0 V reference.

4.3. Alternate Function Redirection

Many I/O pins, in addition to their normal I/O function, they 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~AUXR7 to serve their alternate function on the relocated ports.

AUXR0: Auxiliary Register 0

SFR Page = 0~F
SFR Address = 0xA1 RESET = 000x-xx00

7	6	5	4	3	2	1	0
P60OC1	P60OC0	P60FD	--	--	--	INT1H	INT0H
R/W	R/W	R/W	W	W	W	R/W	R/W

Bit 7~6: P6.0 function configured control bit 1 and 0. The two bits only act when internal RC oscillator (IHRCO or ILRCO) is selected as 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 P6M0.0
01	MCK	By P6M0.0
10	MCK/2	By P6M0.0
11	MCK/4	By P6M0.0

Please refer Section “8 System Clock” to get the more detailed clock information. For clock-out on P6.0 function, it is recommended to set P6M0.0 to “1” which selects P6.0 as push-push output mode.

Bit 5: P60FD, P6.0 Fast Driving.
0: P6.0 output with default driving.
1: P6.0 output with fast driving enabled. If P6.0 is configured to clock output, enable this bit when P6.0 output frequency is more than 12MHz at 5V application or more than 6MHz at 3V application.

AUXR1: Auxiliary Control Register 1

SFR Page = 0~F
SFR Address = 0xA2 RESET = xxxx-xxx0

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

AUXR2: Auxiliary Register 2

SFR Page = 0~F
SFR Address = 0xA3 RESET = 00xx-0000

7	6	5	4	3	2	1	0
STAF	STOF	--	--	T1X12	T0X12	T1CKOE	T0CKOE
R/W	R/W	W	W	R/W	R/W	R/W	R/W

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
00	P3.4
01	P4.4
10	P2.2
11	P2.6

Bit 3: S0PS0, Serial Port 0 pin Selection 0.

S0PS0	RXD0	TXD0
0	P3.0	P3.1
1	P4.4	P4.5

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

TWIPS1~0	TWI0_SCL	TWI0_SDA
00	P4.0	P4.1
01	P6.0	P6.1
10	P3.0	P3.1
11	P2.2	P2.4

AUXR4: Auxiliary Register 4

SFR Page = **1 only**

SFR Address = 0xA4 RESET = 0000-0x00

7	6	5	4	3	2	1	0
T2PS1	T2PS0	T1PS1	T1PS0	SPIPS0	--	AC0OE	AC0FLT1
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**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	P3.0	P3.1
10	P4.0	P4.1
11	P4.5	P4.4

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

T1PS1~0	T1/T1CKO
00	P3.5
01	P4.5
10	Reserved
11	P2.6

Bit 3: SPIPS0, SPI Port pin Selection 0.

SPIPS0	nSS	MOSI	MISO	SPICLK
0	P1.4	P1.5	P1.6	P1.7
1	P3.4	P3.5	P4.1	P4.0

Bit 1: AC0OE, AC0OUT output enable on port pin.
0: Disable AC0OUT output on port pin.
1: Enable AC0OUT output on P6.1.

AUXR5: Auxiliary Register 5

SFR Page = **2 only**

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
C0IC4S0	C0IC2S0	C0PPS1	C0PPS0	C0PS1	C0PS0	ECIPS0	C0COPS
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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

C0IC4S0	CEX4 input
0	CEX4 Port Pin
1	AC0OUT

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

C0IC2S0	CEX2 input
0	CEX2 Port Pin
1	ILRCO

Bit 5: C0PPS1, {PWM2A, PWM2B} Port pin Selection 0.

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	P2.0	P2.1
1	P6.0	P6.1

Bit 3: C0PS1, PCA0 Port pin Selection 1.

C0PS1	CEX3	CEX5
0	P3.4	P3.5
1	P2.0	P2.1

Bit 2: C0PS0, PCA0 Port pin Selection0.

C0PS0	CEX0	CEX2	CEX4
0	P2.2	P2.4	P2.6
1	P3.0	P3.1	P3.3

Notice: When CEX1 and CEX4 both have been selected output to port pin, please note CEX1 will output through P3.3, therefore the CEX4 only can use P2.6 to output the signal.

Bit 1: ECIPS0, PCA0 ECI Port pin Selection0.

ECIPS0	ECI
0	P1.3
1	P1.6

Bit 0: C0COPS, PCA0 Clock Output (C0CKO) port pin Selection.

C0COPS	C0CKO
0	P4.7
1	P3.3

AUXR6: Auxiliary Register 6

SFR Page = **3 only**

SFR Address = 0xA4 RESET = 0000-0x00

7	6	5	4	3	2	1	0
KBI4PS1	KB4IPS0	KBI6PS0	KBI2PS0	KBI0PS0	--	S0MIPS	S0COPS
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**W**	**R/W**	**R/W**

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

KBI4PS1~0	KBI4	KBI5
00	P1.4	P1.5
01	P3.4	P3.5
10	P6.0	P6.1
11	P2.0	P2.1

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

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

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

KBI2PS0	KBI2	KBI3
0	P1.2	P1.3
1	P2.2	P2.4

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

KBI0PS	KBI0	KBI1
0	P1.0	P1.1
1	P4.0	P4.1

Bit 2: Reserved. Software must write "**0**" on this bit when AUXR6 is written.

Bit 1: S0MIPS, S0MI Port pin Selection.

S0MIPS	S0MI
0	P1.6
1	P3.3

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

S0COPS	S0CKO
0	P4.7
1	P3.3

AUXR7: Auxiliary Register 7

SFR Page = **4 only**

SFR Address = 0xA4 RESET = 1100-xxxx

7	6	5	4	3	2	1	0
POE5	POE4	C0CKOE	SPI0M0	--	--	--	--
R/W	**R/W**	**R/W**	**R/W**	**W**	**W**	**W**	**W**

Bit 7: POE5, PCA0 PWM5 main channel (PWM5O) output control.
0: Disable PWM5O output on port pin.
1: Enable PWM5O output on port pin. **Default is enabled.**

Bit 6: POE4, PCA0 PWM4 main channel (PWM4O) output control.
0: Disable PWM4O output on port pin.
1: Enable PWM4O output on port pin. **Default is enabled.**

Bit 5: C0CKOE, PCA0 clock output enable.

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

Bit 4: SPIM0, SPI mode control 0.
0: Disable SPI daisy chain function.
1: Enable SPI daisy chain function in SPI slave mode.

XICFG: External Interrupt Configured Register

SFR Page = **0 only**
SFR Address = 0xC1 RESET = 0000-x000

7	6	5	4	3	2	1	0
INT1IS.1	INT1IS.0	INT0IS.1	INT0IS.0	--	X2FLT	X1FLT	X0FLT
R/W	**R/W**	**R/W**	**R/W**	**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
000	P3.3
001	P3.1
010	P3.5
011	P4.1
100	P6.1
101	P2.1
110	P1.5
111	P2.4

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

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

XICFG1: External Interrupt Configured 1 Register

SFR Page = **1 only**
SFR Address = 0xC1 RESET = 0000-x000

7	6	5	4	3	2	1	0
INT1IS.2	INT0IS.2	INT2IS.1	INT2IS.0	--	X2FLT1	X1FLT1	X0FLT1
R/W	**R/W**	**R/W**	**R/W**	**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 as following table.

INT2IS.1~0	Selected Port Pin of nINT2
00	P4.4
01	P2.0
10	P1.1
11	P1.6

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 **MG82FG5D16** 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 1~6 clock cycles (about 5~6 times fast then the standard 8051 device). It uses a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. The instruction timing is faster than the standard 8051.

In many 8051 implementations, a distinction between machine cycles and clock cycles is that machine cycles varying from 2 to 12 clock cycles. 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 “32 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)
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)
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 to use this method to access the registers are more efficiently because it eliminates the 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 is read-only. 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 **MG82FG5D16** has separated 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. It can be up to **16K** bytes of program memory. In the **MG82FG5D16**, 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.

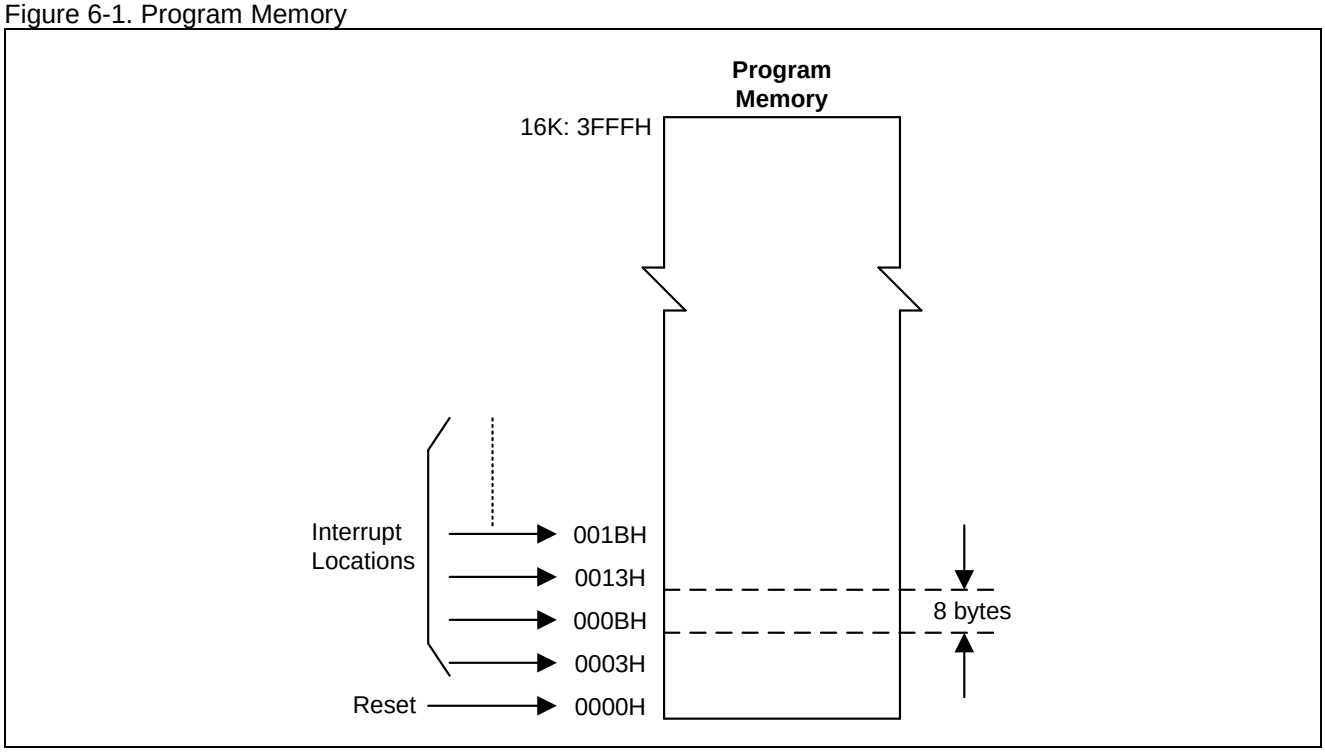
Data memory occupies a separate address space from program memory. In the **MG82FG5D16**, there are 256 bytes of internal scratch-pad RAM and **768** bytes of on-chip expanded RAM(XRAM).

6.1. On-Chip Program Flash

Program memory is the memory which stores the program codes for the CPU to be executed, 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 **MG82FG5D16** 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 used. This improves the code space usage, since register instructions are shorter than direct addressing instructions. The next 16 bytes above the register banks form a block of bit-addressable memory space. The 80C51 instruction set includes a wide selection of single-bit instructions, and the 128 bits in this area can be directly addressed by these instructions. The bit addresses in this area are 00H through 7FH.

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

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

Figure 6-2. Data Memory

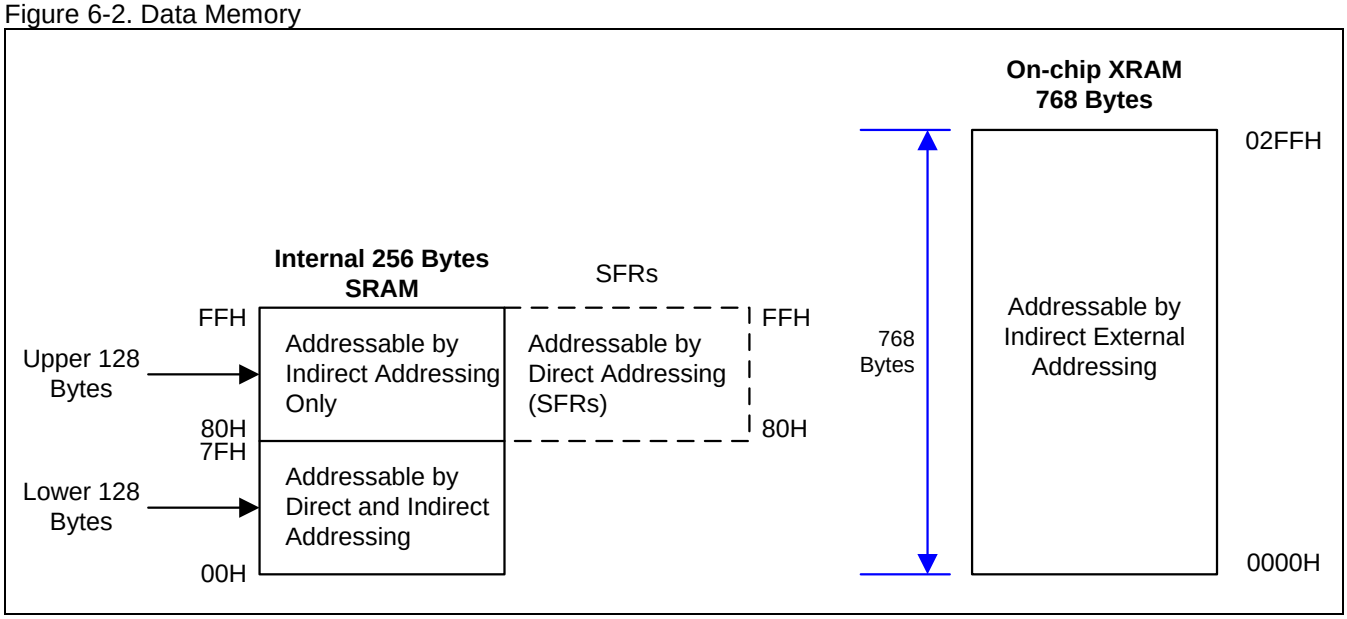


Figure 6-3. Lower 128 Bytes of Internal RAM

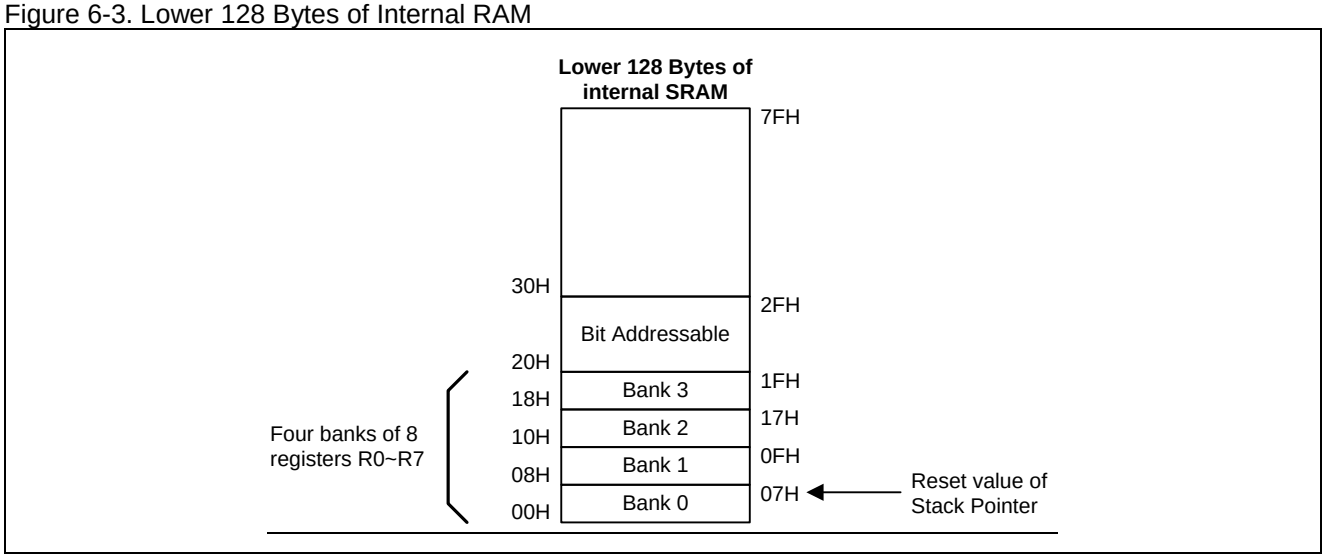
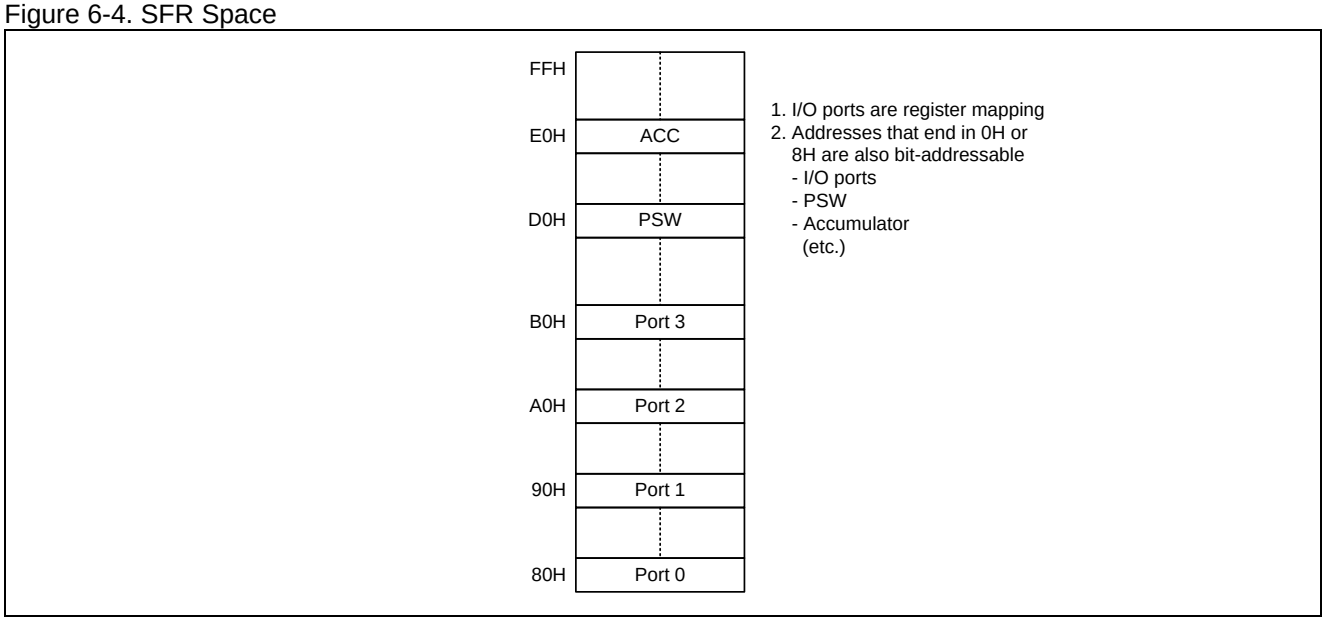


Figure 6-4. SFR Space



6.3. On-chip expanded RAM (XRAM)

To access the on-chip expanded RAM (XRAM), refer to Figure 6–2, the **768** bytes of XRAM (0000H to **02FFH**) are indirectly accessed by move external RAM 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 **MG82FG5D16** hardware can access them correctly.

6.4. Off-Chip External Data Memory access

The off-chip external data memory access function is not supported in **MG82FG5D16**.

6.5. Declaration Identifiers in a C51-Compiler

The declaration identifiers in a C51-compiler for the various **MG82FG5D16** 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 **MG82FG5D16** has **768** bytes of on-chip xdata memory.

pdata
Paged (256 bytes) external data or on-chip eXpanded RAM; duplicates the classic 80C51 256 bytes memory space addressed via the “MOVX @Ri” instruction. The **MG82FG5D16** has 256 bytes of on-chip pdata memory which is shared with on-chip xdata memory.

code
16K bytes of program memory space; accessed as part of program execution and via the “MOVC @A+DTPR” instruction. The **MG82FG5D16** has **16K** bytes of on-chip code memory.

7. XRAM Access

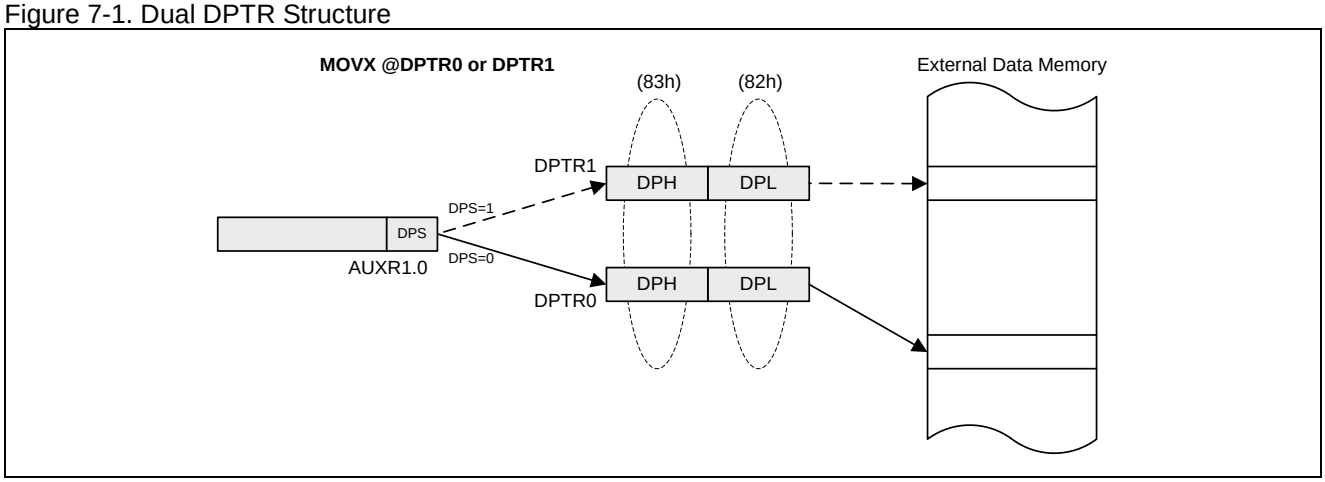
The **MG82FG5D16** MCUs include **768** bytes of on-chip RAM mapped into the external data memory space (XRAM). The external memory space may be accessed using the external move instruction (MOVX) and the data pointer (DPTR), or using the MOVX indirect addressing mode using R0 or R1. If the MOVX instruction is used with an 8-bit address operand (such as @R1), then the high byte of the 16-bit address is provided by the XRAM Page Select Register (XRPS).

The internal XRAM memory spaceis 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
MOVC  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 POR+RESET = xxxx-xxx0

7	6	5	4	3	2	1	0
--	--	--	--	--	--	--	DPS
W	**W**	**W**	**W**	**W**	**W**	**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

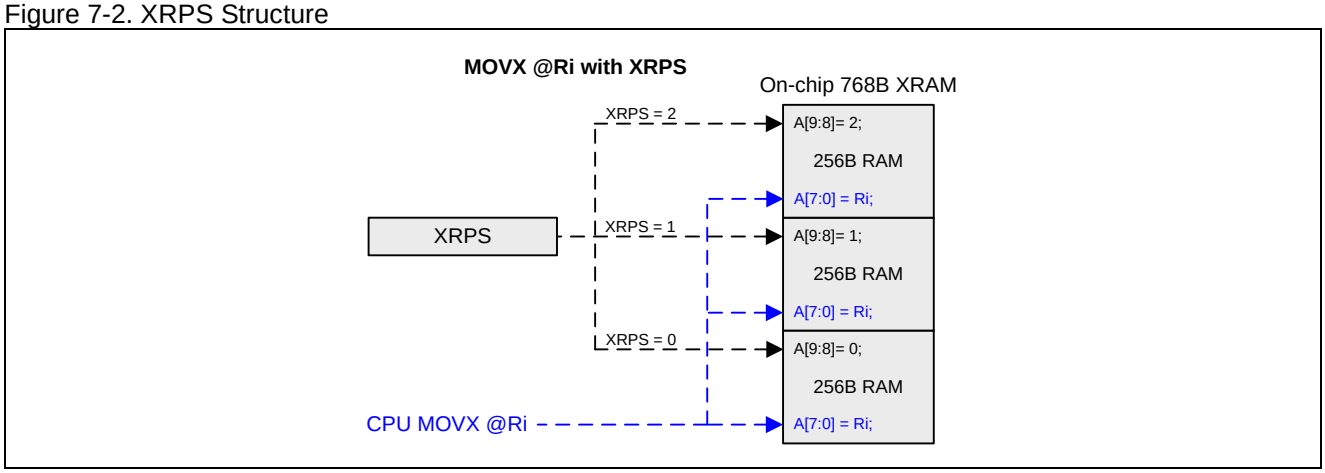
7	6	5	4	3	2	1	0
DPH.7	DPH.6	DPH.5	DPH.4	DPH.3	DPH.2	DPH.1	DPH.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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 is only provided in **MG82FG5D16** for 768 bytes XRAM access.

Figure 7-2. XRPS Structure



XRPS: XRAM Page Select Register

SFR Page = 0~F

SFR Address = 0x8F RESET = xxxx-xx00

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

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

Bit 1~0: XRPS, XRAM Page Select. The XRPS register provides the high byte of the 16-bit external data memory address when using an 8-bit MOVX command, effectively selecting a 256-byte page of RAM. Since the upper (reserved) bits of the register are always zero, the XRPS determines which page of XRAM is accessed. In **MG82FG5D16**, XRPS indexes the **three** pages 256-byte RAM.

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

8. 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 8–1 shows the structure of the system clock in **MG82FG5D16**.

The **MG82FG5D16** 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 wait the clock until the source be settle 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.

To 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. The second 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 “31.4 IHRCO Characteristics”). In IHRCO or ILRCO mode, P6.0 can be configured to internal *MCK* output or *MCK*/2 and *MCK*/4 for system application.

The built-in ILRCO provides the low power and low speed frequency about 32KHz to WDT and system clock source. MCU can select the ILRCO to system clock source by software for low power operation. To find the detailed ILRCO performance, please refer Section “31.5 ILRCO Characteristics”). In ILRCO mode, P6.0 can be configured to internal *MCK* output or *MCK*/2 and *MCK*/4 for system application.

The **MG82FG5D16** device includes a Clock Multiplier (CKM) to generate the high speed clock for system clock source. CKM applied in **MG82FG5D16** is shown in Figure 8–1 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 “31.6 CKM Characteristics”).

The system clock, *SYSCLK*, is obtained from one of these four clock sources through the clock divider, as shown in Figure 8–1. The user can program the divider control bits SCKS2~SCKS0 (in CKCON0 register) to get the desired system clock.

8.1. Clock Structure

Figure 8–1 presents the principal clock systems in the **MG82FG5D16**. The initial oscillator source of CPUCLK is set to IHROC 12MHz. It can use the combinations of the clock multiplier and divider for different frequencies. The maximum CPUCLK is as following:

- External crystal mode: Up to 12MHz @ 2.0V – 5.5V; Up to 25MHz @ 2.7V – 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

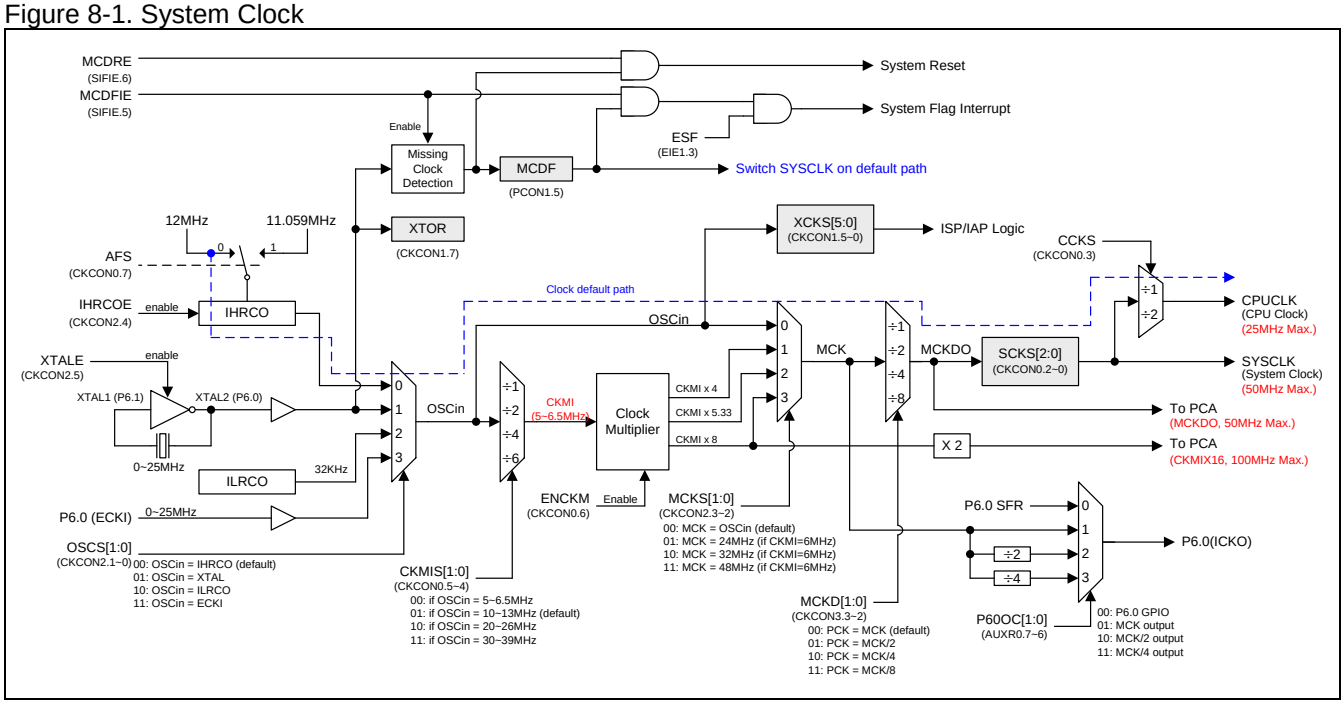
If the applications need higher performance, then HSE (DCON0 Bit 7) needs to be set when CPUCLK > 6MHz. Moreover, if needs ultra-high CPUCLK>25MHz, then HSE1 needs to be set.

The system clock can be sourced by the external oscillator circuit or either internal oscillator. It maximum frequency is 50MHz. Please note, when using Clock Multiplier (CKM) to raise the MCK frequency to get higher SYSCLK, the CPUCLK will be also changed. It is need 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 provide two more clock source for high speed PCA applications.

- MCKDO: Up to 50MHz
- CKMIX16: Up to 100MHz

Figure 8-1. System Clock



8.2. 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 8–1 shows the structure of the system clock in **MG82FG5D16**. The **MG82FG5D16** 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.

8.3. On-chip CKM (PLL)

The **MG82FG5BXX** includes a Clock Multiplier (CKM) to generate the high speed clock for system clock source. It is shown in Figure 8–1 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 “31.6 CKM Characteristics”).

8.4. Missing Clock Detection

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 an system reset. If user won’t apply this function, clear MCDRE to disable the reset function.

8.5. 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…….

8.6. Wake-up for 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…….

8.7. 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	OSCin/6 (when OSCin = 30 ~ 42MHz)

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 & P**

SFR Address = 0xBF RESET = 0x00-1011

7	6	5	4	3	2	1	0
XTOR	0	XCKS5	XCKS4	XCKS3	XCKS2	XCKS1	XCKS0
R	**W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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: Reserved. Software must write “0” on this bit when CKCON1 is written.

Bit 5~0: This is set the OSCin frequency value to define the time base of ISP/IAP programming. Fill with a proper value according to OSCin, as listed below.

[XCKS5~XCKS0] = OSCin – 1, w*here OSCin=1~40 (MHz).*

For examples,
(1) If OSCin=12MHz, then fill [XCKS5~XCKS0] with 11, i.e., 00-1011B.
(2) If OSCin=6MHz, then fill [XCKS5~XCKS0] with 5, i.e., 00-0101B.

OSCin	XCKS[5:0]
1MHz	00-0000
2MHz	00-0001
3MHz	00-0010
4MHz	00-0011
……	……
……	……
38MHz	10-0101
39MHz	10-0110
40MHz	10-0111

The default value of XCKS= 00-1011 for OSCin= 12MHz.

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, OSC Driving control Register.

XTGS1, XTGS0	Gain Define
0, 0	Gain for 32.768K
0, 1	Gain for 2MHz ~ 25MHz
Others	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 x 4 (ENCKM =1)	24MHz	22.118MHz
1 0	CKMI x 5.33 (ENCKM =1)	32MHz	29.491MHz
1 1	CKMI x 8 (ENCKM =1)	48MHz	44.236MHz

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	0	WDTFS	MCKD1	MCKD0	1	0
R/W	R/W	W	R/W	R/W	R/W	W	W

Bit 7~6: WDTCS1~0. WDT clock source selection.

Bit 5: Reserved. Software must write “0” on this bit when CKCON3 is written.

Bit 4: WDTFS. WDT overflow source selection.

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: Reserved. Software must write “**10**” on these bits when PCON1 is written.

AUXR0: Auxiliary Register 0

SFR Page = 0~F

SFR Address = 0xA1 RESET = 000x-xx00

7	6	5	4	3	2	1	0
P60OC1	P60OC0	P60FD	--	--	--	INT1H	INT0H
R/W	R/W	R/W	W	W	W	R/W	R/W

Bit 7~6: P6.0 function configured control bit 1 and 0. The two bits only act when internal RC oscillator (IHRCO or ILRCO) is selected for system clock source. In 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 P6M0.0
01	MCK	By P6M0.0
10	MCK/2	By P6M0.0
11	MCK/4	By P6M0.0

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

Bit 5: P60FD, P6.0 Fast Driving.
0: P6.0 output with default driving.
1: P6.0 output with fast driving enabled. If P6.0 is configured to clock output, enable this bit when P6.0 output frequency is more than 12MHz at 5V application or more than 6MHz at 3V application.

PCON1: Power Control Register 1

SFR Page = 0~F & P

SFR Address = 0x97 POR = 0000-x000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	--	BOF1	BOF0	WDTF
R/W	**R/W**	**R/W**	**R/W**	**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 on external crystal oscillating input. 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 again. To enable this function must combine XTALE & OSCS[1:0] = 01, external crystal mode.

SFIE: System Flag Interrupt Enable Register

SFR Page = 0~F

SFR Address = 0x8E POR = 0110-x000

7	6	5	4	3	2	1	0
SIDFIE	MCDRE	MCDFIE	RTCFIE	--	BOF1IE	BOF0IE	WDTFIE
R/W	**R/W**	**R/W**	**R/W**	**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. Default is enabled.

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

DCON0: Device Control Register 0

SFR Page = **P Only**

SFR Address = 0x4C POR = 1000-x011

7	6	5	5	4	3	2	1	0
HSE	IAPO	HSE1	--	--	--	IORCTL	RSTIO	OCDE
R/W	**R/W**	**W/R**	**W**	**W**	**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 SYSCLK, 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. ($F_{CPUCLK} > 25MHz$)

9. Watch Dog Timer (WDT)

9.1. WDT Structure

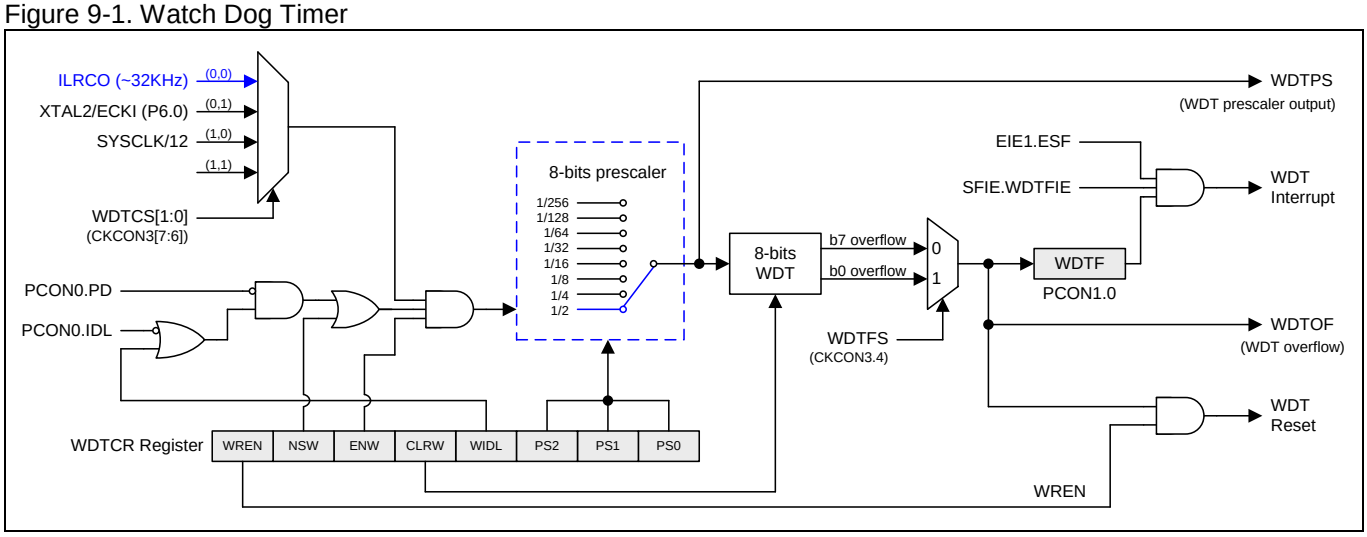
The Watch-dog Timer (WDT) is intended as a recovery method in situations where the CPU may be subjected to software upset. The WDT consists of a 8-bit free-running counter, a 8-bit prescaler and a control register (WDTCR). Figure 9–1 shows the WDT structure in **MG82FG5D16**.

There are four/three 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 (RST-pin) reset, software reset and WDT reset.

WREN, NSW and ENW are implemented to one-time-enabled function, only writing “1” valid in general SFR page. Page-P SFR Access on WDTCR can disable WREN, NSW and ENW, writing “0” on WDTCR.7~5. Please refer Section “9.4 WDT Register” and Section “27 Page P SFR Access” for more detail information.

Figure 9-1. Watch Dog Timer



9.2. 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.

9.3. Auto Weak 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.

9.4. WDT Register

WDTCR: Watch-Dog-Timer Control Register

SFR Page = 0~F & P

SFR Address = 0xE1 POR = XXX0-XXXX (0000-0000)

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 us
0 0 1	4	32 ms	1.024 ms
0 1 0	8	64 ms	2.048 ms
0 1 1	16	128 ms	4.096 ms
1 0 0	32	256 ms	8.192 ms
1 0 1	64	512 ms	16.384 ms
1 1 0	128	1024 ms	32.768 ms
1 1 1	256	2048 ms	65.536 ms

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

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

CKCON3: Clock Control Register 3

SFR Page = **P only**

SFR Address = 0x41 POR = 0000-0010

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

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-x000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	--	BOF1	BOF0	WDTF
R/W	**R/W**	**R/W**	**R/W**	**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-x000

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

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

9.5. 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.

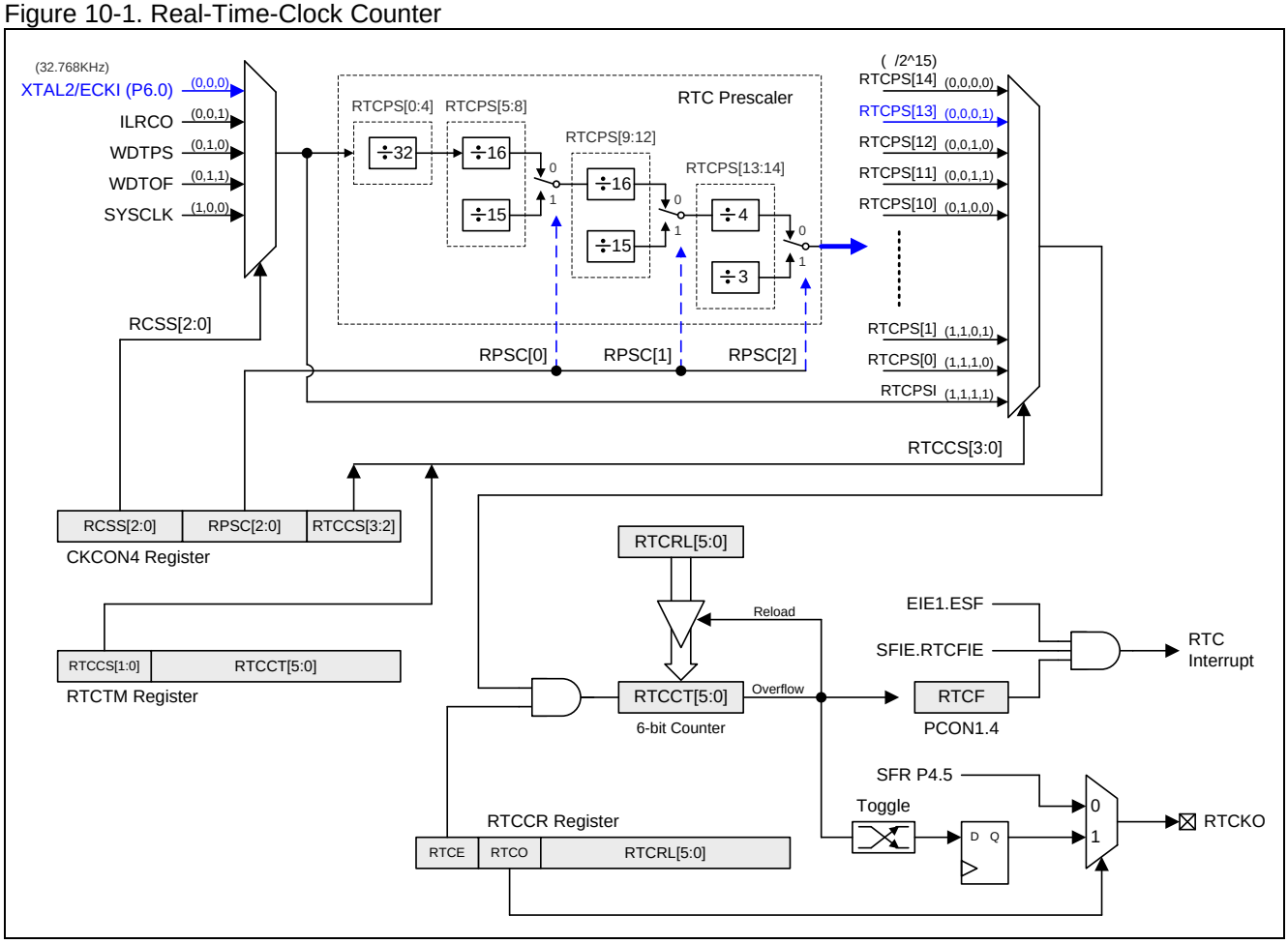
10. Real-Time-Clock(RTC)/System-Timer

The **MG82FG5D16** 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 **5** selections including the XTAL oscillator in default, provided that the XTAL oscillator is not being used as the system clock. Figure 10–1 shows the RTC structure in **MG82FG5D16**.

The 32.768KHz crystal for the RTC module input will provide a programmable overflow period for 30uS 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 more long 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 10-1. Real-Time-Clock Counter



10.1. RTC Register

RTCCR: Real-Time-Clock Control Register

SFR Page = 0~F & P
SFR Address = 0xBE 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 = 0~F
SFR Address = 0xB6 POR = 0111-1111

7	6	5	4	3	2	1	0
RTCCS1	RTCCS0	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: RTCCS3~0, RTC Counter Clock Selection. Default is “01”.

RTCCS3~0	Clock Source	RTC Interrupt Duration	Min. Step
0 0 0 0	RTCPS[14] (/2^15)	1S ~ 64S when P6.0 = 32768Hz	1S
0 0 0 1	RTCPS[13] (/2^14)	0.5S ~ 32S when P6.0 = 32768Hz	0.5S (default)
0 0 1 0	RTCPS[13] (/2^13)	0.25S ~ 16S when P6.0 = 32768Hz	0.25S
……	……	……	……
1 0 1 0	RTCPS[4] (/2^5)	976us ~ 62.46ms when P6.0 = 32768Hz	976 us
1 0 1 1	RTCPS[3] (/2^4)		488 us
1 1 0 0	RTCPS[2] (/2^3)		244 us
1 1 0 1	RTCPS[1] (/2^2)	122us ~ 3.9ms when P6.0 = 32768Hz	122 us
1 1 1 0	RTCPS[0] (/2^1)	61us ~ 1.952ms when P6.0 = 32768Hz	61 us
1 1 1 1	RTCPSI (/2^0)	30.5us ~ 976us when P6.0 = 32768Hz	30.5 us

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

CKCON4: Clock Control Register 4

SFR Page = **P only**

SFR Address = 0x42 POR = 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: RCSS2~0, RTC module Clock Source Selection bits 2~0.

RCSS[2:0]	RTC module Clock Source
0 0 0	XTAL2/ECKI (P6.0)
0 0 1	ILRCO
0 1 0	WDTPS
0 1 1	WDTOF
1 0 0	SYSCLK
Others	Reserved.

Bit 4: RPSC2, RTC PreScaler Control 2.
Bit 3: RPSC1, RTC PreScaler Control 1.
Bit 2: RPSC1, RTC PreScaler Control 0.

Bit 1~0: RTCCS3~2, RTC Counter Clock Selection. The function is active with RTCCS1~0.

PCON1: Power Control Register 1

SFR Page = 0~F & P

SFR Address = 0x97 POR = 0000-x000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	--	BOF1	BOF0	WDTF
R/W	**R/W**	**R/W**	**R/W**	**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-x000

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

11. System Reset

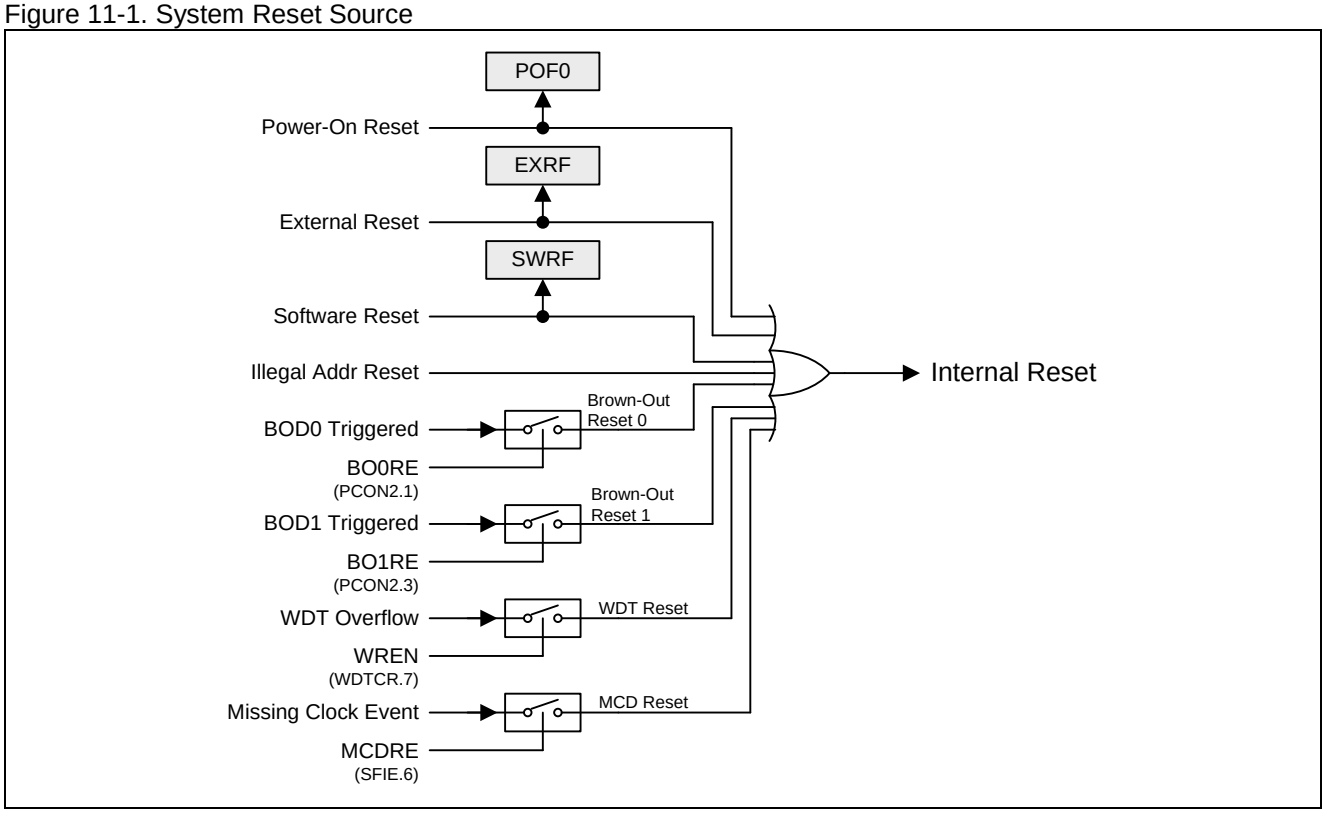
During reset, all I/O Registers are set to their initial values, and the program starts execution from the Reset Vector, 0000H, or ISP start address by OR setting. The **MG82FG5D16** has eight sources of reset: power-on reset, external reset, software reset, illegal address reset, brown-out reset 0, brown-out reset 1 WDT reset and Missing-Clock-Detection reset. Figure 11–1 shows the system reset source in **MG82FG5D16**.

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

11.1. Reset Source

Figure 11–1 presents the reset systems in the **MG82FG5D16** and all of its reset sources.

Figure 11-1. System Reset Source



11.2. 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 in order to ensure a power-on reset

PCON0: Power Control Register 0

SFR Page = 0~F & P

SFR Address = 0x87 POR = 0001-0000, 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 CPU starts up from power up. Note that the POF0 must be cleared by firmware.

11.3. External Reset

A reset is accomplished by holding the RESET pin HIGH for at least 24 oscillator periods while the oscillator is running. To ensure a reliable power-up reset, the hardware reset from RST pin is necessary.

PCON1: Power Control Register 1
SFR Page = 0~F & P
SFR Address = 0x97 POR = 0000-x000

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

11.4. 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 RESET = 0000-xxxx

7	6	5	4	3	2	1	0
ISPEN	SWBS	SWRST	CFAIL	0	0	0	0
R/W	**R/W**	**R/W**	**R/W**	**W**	**R/W**	**R/W**	**R/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-x000

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

11.5. Brown-Out Reset

In **MG82FG5D16**, there are two Brown-Out Detectors (BOD0 & BOD1) to monitor VDD power. BOD0 services the fixed detection level at VDD=1.7V. BOD1 detects the VDD level by software selecting 4.2V, 3.7V, 2.4V or 2.0V. If VDD power drops below BOD0 or BOD1 monitor level. Associated flag, BOF0 and BOF1, is set. If BO0RE (PCON2.1) is enabled, BOF0 indicates a BOD0 Reset occurred. If BO1RE (PCON2.3) is enabled, BOF1 indicates a BOD1 Reset occurred.

PCON1: Power Control Register 1

SFR Page = 0~F & P

SFR Address = 0x97 POR = 0000-x000

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

11.6. 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-x000

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

11.7. 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 an 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-x000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	--	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	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. To enable this function must combine XTALE & OSCS[1:0] = 01, external crystal mode.

SFIE: System Flag Interrupt Enable Register

SFR Page = 0~F
SFR Address = 0x8E POR = 0110-x000

7	6	5	4	3	2	1	0
SIDFIE	MCDRE	MCDFIE	RTCFIE	--	BOF1IE	BOF0IE	WDTFIE
R/W	R/W	R/W	R/W	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. Default is enabled.

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

11.8. Illegal Address Reset

In **MG82FG5D16**, if software program runs to illegal address such as over program ROM limitation, it triggers a RESET to CPU.

12. Power Management

The **MG82FG5D16** 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, PCON0, PCON1, PCON2, PCON3, RTCCR and WDTCR register.

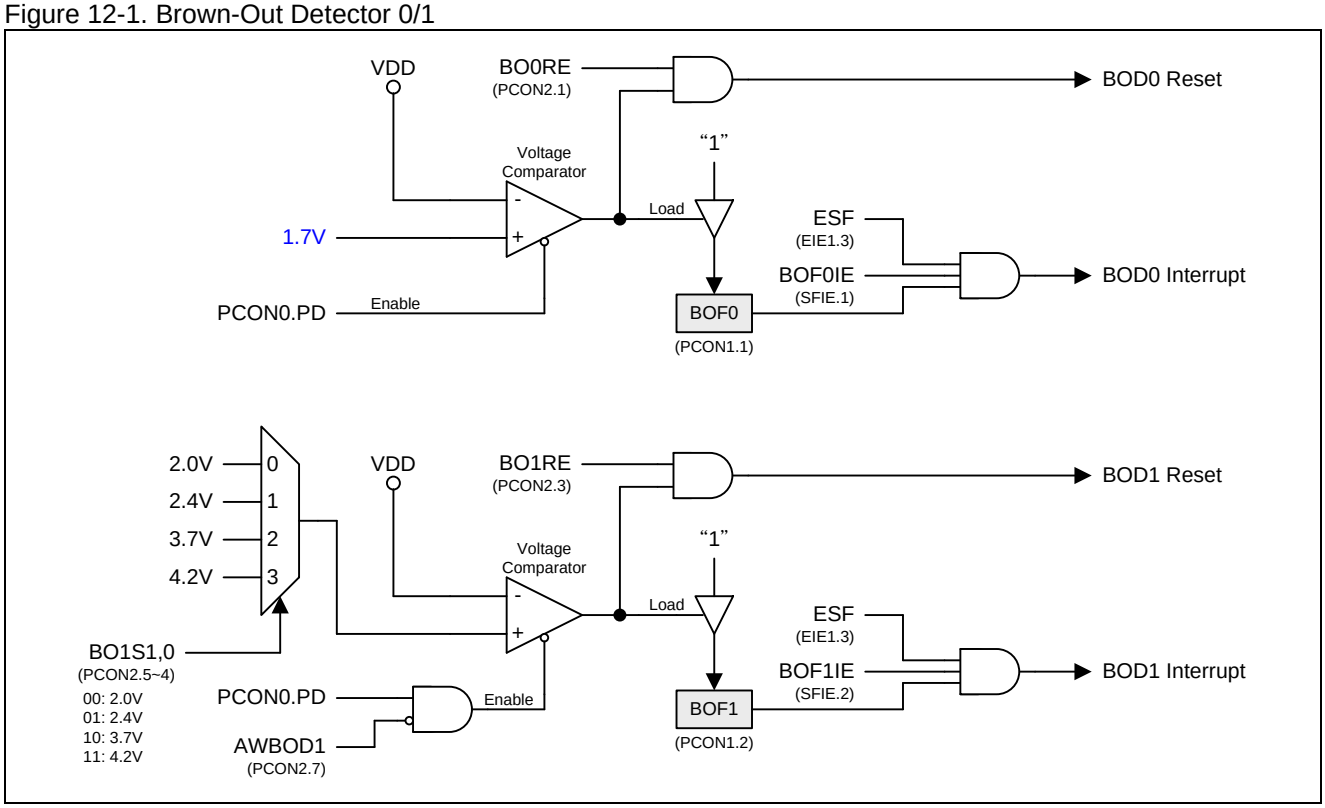
12.1. Brown-Out Detector

In **MG82FG5D16**, there are two Brown-Out Detectors (BOD0 & BOD1) to monitor VDD power. Figure 12–1 shows the functional diagram of BOD0 and BOD1. BOD0 services the fixed detection level at VDD=1.7V and BOD1 detects the software selection levels (4.2V/3.7V/2.4V/2.0V) on VDD. Associated flag, BOF0 (PCON1.1), is set when BOD0 meets the detection level. If both of ESF (EIE1.3) and BOF0IE (SFIE.1) are enabled, a set BOF0 will generate a system flag interrupt. It can interrupt CPU either CPU in normal mode or idle mode. The BOD1 has the same flag function, BOF1, and same interrupt function. The BOD1 interrupt also wakes up CPU in power down mode if AWBOD1 (PCON2.7) is enabled.

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

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

Figure 12-1. Brown-Out Detector 0/1



12.2. Power Saving Mode

12.2.1. Slow Mode

The alternative way to save the operating power is to slow the MCU's operating speed by programming SCKS2~SCKS0 bits (in CKCON0 register, see Section "8 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.

12.2.2. Sub-Clock Mode

The alternative way to slow down the MCU's operating speed by programming OSCS1~0 to select the ILRCO for system clock. The 32KHz ILRCO provides the ultra-low speed and low power operation for the MCU. Additional programming SCKS2~SCKS0 bits (in CKCON0 register, see Section "8 System Clock"), the user could put the MCU speed down to 250Hz slowest.

12.2.3. RTC Mode

The **MG82FG5D16** has a simple RTC module that allows the user to continue running an accurate timer while the rest of the device is powered-down. In RTC mode, the RTC module behaves as a "Clock" function and can be a wake-up source from chip power down by RTC overflow rate. Please refer Section "10 Real-Time-Clock(RTC)/System-Timer" for more detail information.

12.2.4. 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 **MG82FG5D16**. 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 "9 Watch Dog Timer (WDT)" and Section "14 Interrupt" for more detail information.

12.2.5. Monitor Mode

If AWBOD1 (PCON2.3) is set, BOD1 will keep VDD monitor in power down mode. It is the Monitor Mode in **MG82FG5D16**. When BOD1 meets the detection level, set BOF1 and wakeup CPU from interrupt or system reset by software configured. Please refer Section "12.1 Brown-Out Detector" and Section "14 Interrupt" for more detail information.

12.2.6. 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 when Idle was activated. Idle mode leaves the peripherals running in order to allow them to wake up the CPU when an interrupt is generated. Timer 0, Timer 1, Timer 2, SPI, KBI, ADC, S0, TWI0, RTC, MCD, BOD0 and BOD1 will continue to function during Idle mode. PCA0 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 or analog comparator input channels must be set to "***Analog Input Only***" when MCU is in idle mode or power-down mode to reduce power consumption.

12.2.7. Power-down Mode

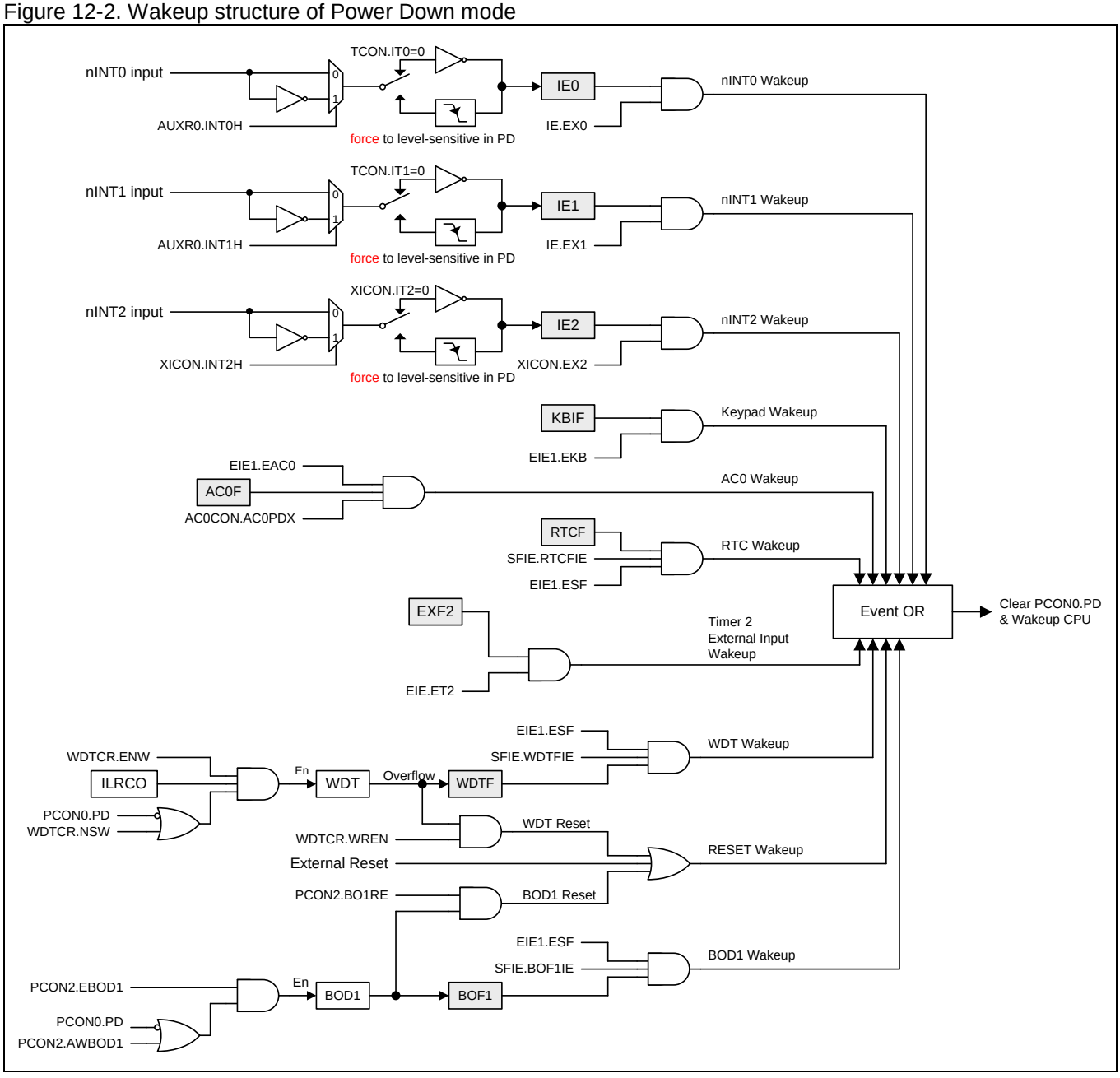
Setting the PD bit in PCON0 enters Power-down mode. Power-down mode stops the oscillator and powers down the Flash memory in order to minimize power consumption. Only the power-on circuitry will continue to draw power during Power-down. During Power-down the power supply voltage may be reduced to the RAM keep-alive voltage. The RAM contents will be retained; however, the SFR contents are not guaranteed once VDD has been reduced.

Power-down may be excited by external reset, power-on reset, enabled external interrupts, enabled KBI, enabled RTC (RTC mode), enabled BOD1 (monitor mode) or enabled Non-Stop WDT (watch mode).

The user should not attempt to enter (or re-enter) the power-down mode for a minimum of 4 µs until 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 12–2 shows the wakeup mechanism of power-down mode in **MG82FG5D16**.

Figure 12-2. Wakeup structure of Power Down mode



12.2.8. Interrupt Recovery from Power-down

Four external interrupts may be configured to terminate Power-down mode. External interrupts nINT0, nINT1and nINT2 may be used to exit Power-down. To wake up by external interrupt nINT0, nINT1 or nINT2, the interrupt must be enabled and configured for level-sensitive operation. If the enabled external interrupts are configured to edge-sensitive operation (Falling or Rising), they will be **forced** to level-sensitive operation (Low level or High level) by hardware in power-down mode.

When terminating Power-down by an interrupt, the wake up period is internally timed. At the falling edge on the interrupt pin, Power-down is exited, the oscillator is restarted, and an internal timer begins counting. The internal clock will not be allowed to propagate and the CPU will not resume execution until the timer counter has been 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.

12.2.9. Reset Recovery from Power-down

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

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

12.2.10. KBI wakeup Recovery from Power-down

The Keypad Interrupt of **MG82FG5D16**, KBI.7~0 have wakeup CPU capability that are enabled by the control registers in KBI module. OR software can configure AUXR6.7~3 to select port pins to serve the KBI function. Please refer Section “28 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.

12.2.11. Fast wakeup for XTAL mode

Reference “8.5 Fast wake-up for XTAL mode”

12.3. Power Control Register

PCON0: Power Control Register 0

SFR Page = 0~F & P
SFR Address = 0x87 POR = 0001-0000, 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: Setting this bit activates power down operation.

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

PCON1: Power Control Register 1

SFR Page = 0~F & P
SFR Address = 0x97 POR = 0000-x000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	**0**	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	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. To enable this function must combine XTALE & OSCS[1:0] = 01, external crystal mode.

Bit 4: RTCF, RTC overflow flag.
0: This bit must be cleared by software writing **"1"** on it. Software writing ":0" is no operation.
1: This bit is only set by hardware when RTCCT overflows. Writing **"1"** on this bit will clear RTCF.

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

Bit 2: BOF1, Brown-Out Detection flag 1.
0: This bit must be cleared by software writing **"1"** to it.
1: This bit is set by hardware if the operating voltage matches the detection level of Brown-Out Detector 1 (4.2V/3.7/2.4/2.0).

Bit 1: BOF0, Brown-Out Detection flag 0.
0: This bit must be cleared by software writing **"1"** to it.
1: This bit is set by hardware if the operating voltage matches the detection level of Brown-Out Detector 0 (1.7V).

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

PCON2: Power Control Register 2

SFR Page = **P Only**
SFR Address = 0x44 POR = 0011-0101

7	6	5	4	3	2	1	0
AWBOD1	**0**	BO1S1	BO1S0	BO1RE	EBOD1	BO0RE	**1**
R/W	**W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**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	3.7V
1 1	4.2V

Bit 3: BO1RE, BOD1 Reset Enabled.
0: Disable BOD1 to trigger a system reset when BOF1 is set.
1: Enable BOD1 to trigger a system reset when BOF1 is set.

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

Bit 1: BO0RE, BOD0 Reset Enabled.
0: Disable BOD0 to trigger a system reset when BOF0 is set.
1: Enable BOD0 to trigger a system reset when BOF0 is set (VDD meets 1.7V).

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

PCON3: Power Control Register 3

SFR Page = **P Only**
SFR Address = 0x45 POR = 0xxx-xxxx

7	6	5	4	3	2	1	0
IVREN	--	--	--	--	--	--	--
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 (1.4V).
1: Enable on-chip IVR (1.4V).

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

13. Configurable I/O Ports

The **MG82FG5D16** has following I/O ports: P1.0~P1.7, P2.0~P2.2, P2.4, P2.6, P3.0, P3.1, P3.3~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 13–1.

Table 13-1. Number of I/O Pins Available

Package Type	I/O Pins	Number of I/O ports
28-pin	P1.0~P1.7, P2.0~P2.2, P2.4, P2.6 P3.0, P3.1, P3.3~P3.5, P4.0, P4.1, P4.4, P4.5, P4.7(RST), P6.0 (ECKI/XTAL2), P6.1 (XTAL1)	25 or 24 (RST selected) or 23 (RST & ECKI selected) or 22 (RST & XTAL selected)
20-pin	P1.0, P1.1, P1.5~P1.7, P2.2, P2.4, P3.0, P3.1, P3.3~P3.5, P4.4, P4.5, P4.7(RST), P6.0 (ECKI/XTAL2), P6.1 (XTAL1)	17 or 16 (RST selected) or 15 (RST & ECKI selected) or 14 (RST & XTAL selected)
16-pin	P1.0, P1.1, P1.6, P2.2, P2.4, P3.0, P3.1, P3.3, P4.4, P4.5, P4.7(RST), P6.0 (ECKI/XTAL2), P6.1 (XTAL1)	13 or 12 (RST selected) or 11 (RST & ECKI selected) or 10 (RST & XTAL selected)

13.1. IO Structure

The I/O operating modes are divided into two groups in **MG82FG5D16**. The first group is only for Port 3 to support four configurations on I/O operating. These are: quasi-bidirectional (standard 8051 I/O port), push-pull output, input-only (high-impedance input) and open-drain output. The Port 3 default setting is quasi-bidirectional mode with weakly pull-up resistance.

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

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

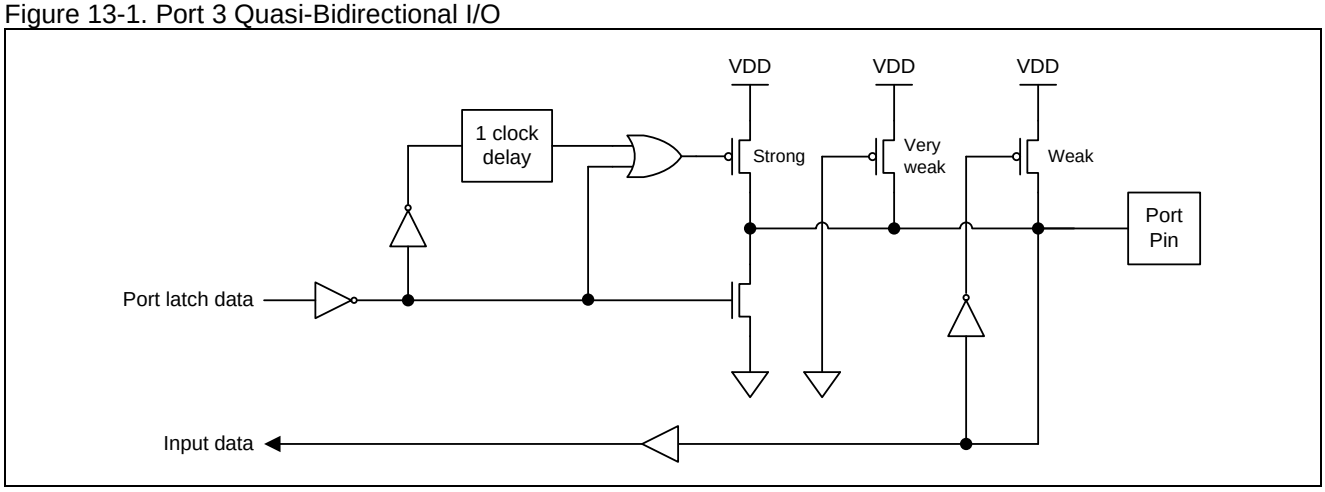
13.1.1. Port 3 Quasi-Bidirectional IO Structure

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 13–1.

Figure 13-1. Port 3 Quasi-Bidirectional I/O

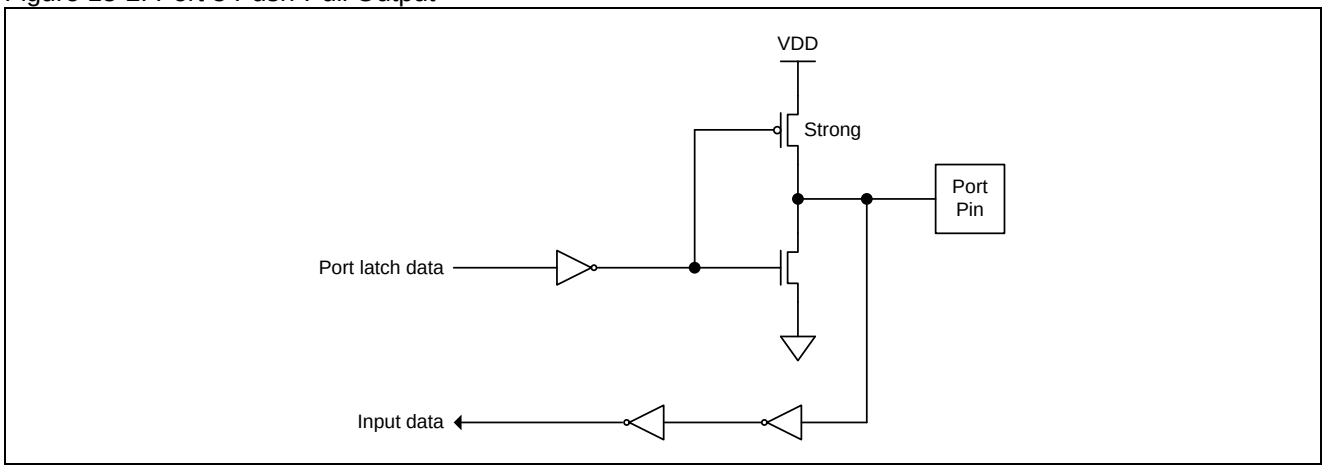


13.1.2. Port 3 Push-Pull Output Structure

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 13–2.

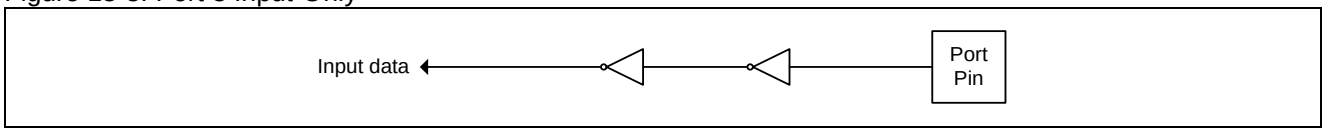
Figure 13-2. Port 3 Push-Pull Output



13.1.3. 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 13–3.

Figure 13-3. Port 3 Input-Only

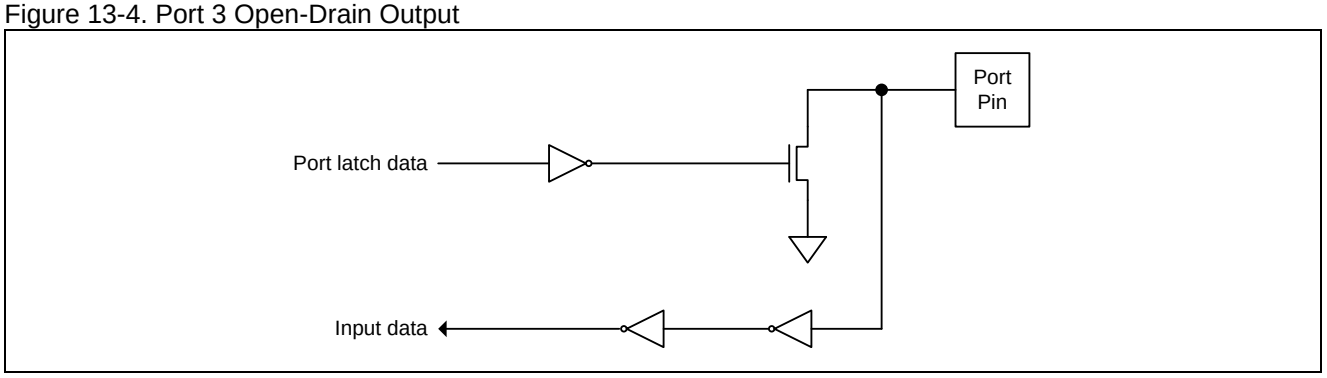


13.1.4. Port 3 Open-Drain Output Structure

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 13–4.

Figure 13-4. Port 3 Open-Drain Output

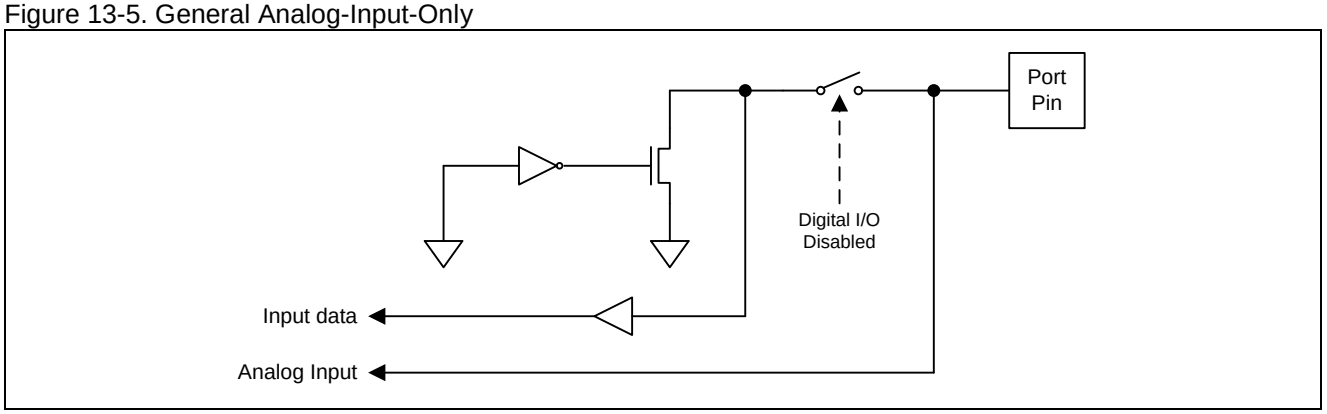


13.1.5. General Analog Input Only Structure

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

The analog-input-only port configuration is shown in Figure 13–7.

Figure 13-5. General Analog-Input-Only

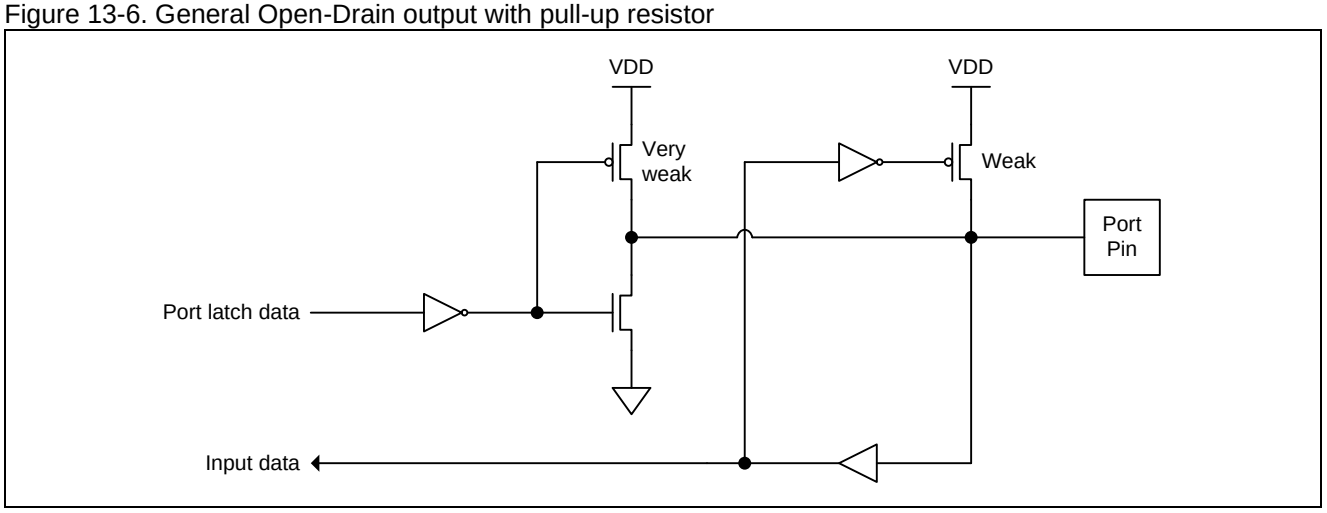


13.1.6. General Open-Drain Output with Pull-up Resistor Structure

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 13–7.

Figure 13-6. General Open-Drain output with pull-up resistor

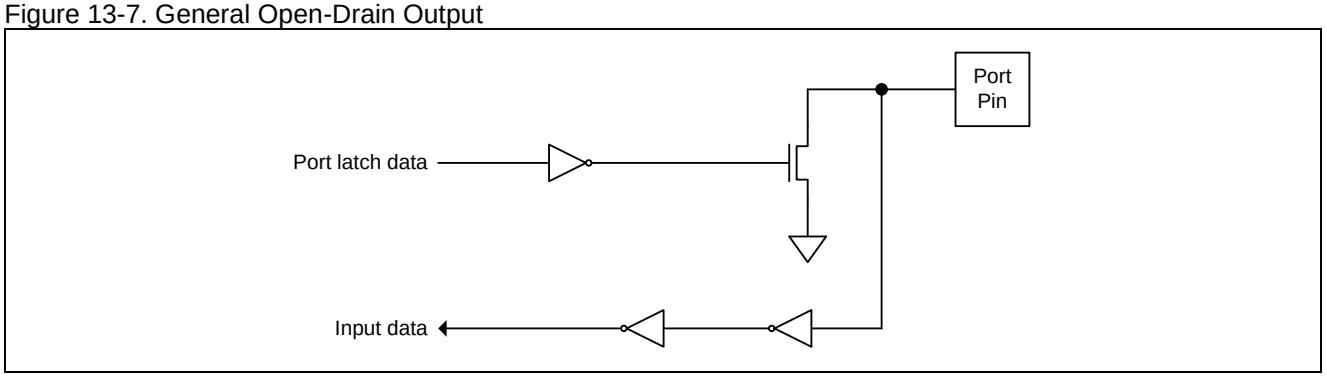


13.1.7. General Open-Drain Output Structure

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 13–7.

Figure 13-7. General Open-Drain Output



13.1.8. General Port Digital Input Configured

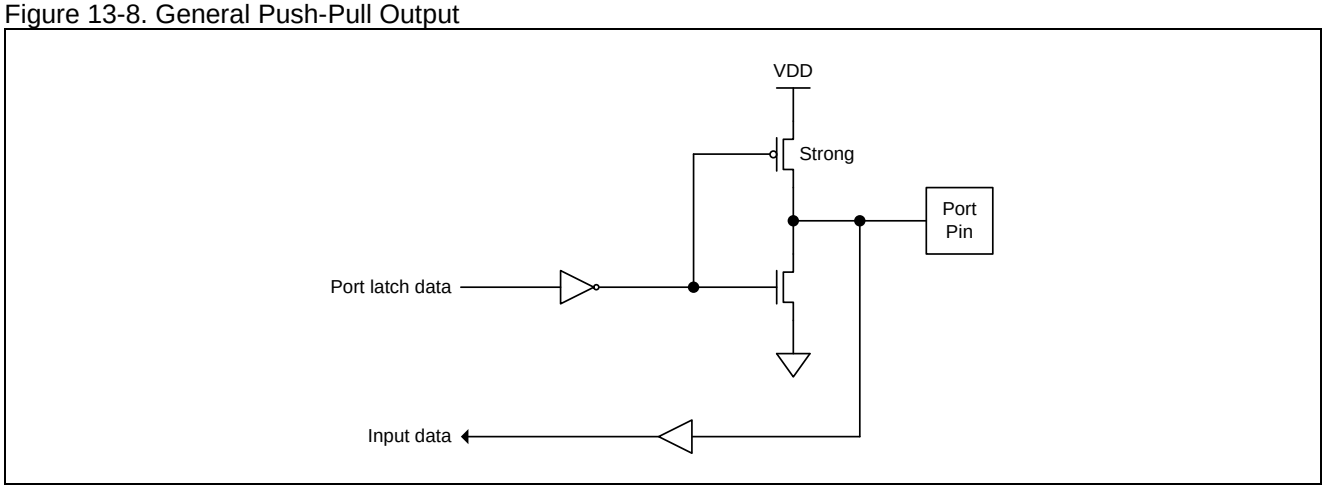
A Port pin is configured as a digital input by setting its output mode to “Open-Drain” and writing a logic “1” to the associated bit in the Port Data register. For example, P1.0 is configured as a digital input by setting P1M0.0 to a logic 0, P1M1.0 to a logic 0 and P1.0 to a logic 1.

13.1.9. 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 13–8.

Figure 13-8. General Push-Pull Output



13.2. I/O Port Register

All I/O port pins on the **MG82FG5D16** may be individually and independently configured by software to select its operating modes. Port 3 has four operating modes, as shown in Table 13–2. Two mode registers select the output type for each port 3 pin. Only Port 3 supports quasi-bidirectional mode and setting them to quasi-bidirectional mode after system reset.

Table 13-2. Port 3 Configuration Settings

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

Where y=0, 1, 3~5 (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 13–3. 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 13-3. General Port Configuration Settings

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

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

13.2.1. 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**

13.2.2. Port 2 Register

P2: Port 2 Register

SFR Page = 0~F
SFR Address = 0xA0 RESET = x1x1-x111

7	6	5	4	3	2	1	0
--	P2.6	--	P2.4	--	P2.2	P2.1	P2.0
W	**R/W**	**W**	**R/W**	**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 = x0x0-x000

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

P2M1: Port 2 Mode Register 1

SFR Page = **1 only**
SFR Address = **0x92** RESET = x1x1-x111

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

13.2.3. Port 3 Register

P3: Port 3 Register

SFR Page = 0~F
SFR Address = 0xB0 RESET = xx11-1x11

7	6	5	4	3	2	1	0
--	--	P3.5	P3.4	P3.3	--	P3.1	P3.0
W	**W**	**R/W**	**R/W**	**R/W**	**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 = xx00-0x00

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

P3M1: Port 3 Mode Register 1

SFR Page = 0~F
SFR Address = 0xB2 RESET = xx00-0x00

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

13.2.4. Port 4 Register

P4: Port 4 Register

SFR Page = 0~F & P
SFR Address = 0xE8 RESET = 1x11-xx11

7	6	5	4	3	2	1	0
P4.7	--	P4.5	P4.4	--	--	P4.1	P4.0
R/W	**W**	**R/W**	**R/W**	**W**	**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 RST input.

P4M0: Port 4 Mode Register 0

SFR Page = 0 only
SFR Address = 0xB3 RESET = 0x00-xx00

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

Bit 7, 5, 4, 1, 0:
0: Port pin output mode is configured to open-drain.
1: Port pin output mode is configured to push-pull.

Bit 6, 3, 2: Reserved. Software must write “0” on these bits when P4M0 is written.

P4M1: Port 4 Mode Register 1

SFR Page = **2 only**
SFR Address = **0x92** RESET = 1x11-xx11

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

Bit 7, 5, 4, 1, 0:
0: Port pin output mode is configured to open-drain.
1: Port pin output mode is configured to push-pull.

Bit 6, 3, 2: Reserved. Software must write “0” on these bits when P4M0 is written.

13.2.5. Port 6 Register

P6: Port 6 Register

SFR Page = **0 ~ F**
SFR Address = **0xF8** RESET = xxxx-xx11

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

Bit 7~0: Port 6 output data latch could be only set/cleared by CPU.
P6.1 and P6.0 have the alternated function for crystal oscillating circuit, XTAL1 and XTAL2.

P6M0: Port 6 Mode Register 0

SFR Page = **1 only**

SFR Address = **0xB5** RESET = xxxx-xx00

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

Reference Table 13–3 .

P6M1: Port 6 Mode Register 1

SFR Page = **3 only**

SFR Address = **0x92** RESET = xxxx-xx11

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

Reference Table 13–3 .

13.2.6. Port Output Driving Strength Control Register

In **MG82FG5D16,** 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 = 0000-00xx

7	6	5	4	3	2	1	0
P3DC1	P3DC0	P2DC1	P2DC0	P1DC1	P1DC0	0	0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**W**	**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 1~0: Reserved. Software must write “0” on these bits when PDRVC0 is written.

PDRVC1: Port Drive Control Register 1

SFR Page = **3 only**
SFR Address = **0xB4** RESET = xxxx-xx00

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

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

Bit 1: P4DC1, Port 4 output driving strength control on high nibble.
0: Select the P4.5 ~ P4.4 output with high driving strength.
1: Select the P4.5 ~ P4.4 output with low driving strength.

Bit 0: P0DC0, Port 0 output driving strength control on low nibble.
0: Select the P4.1 ~ P4.0 output with high driving strength.
1: Select the P4.1 ~ P4.0 output with low driving strength.

14. Interrupt

The **MG82FG5D16** has **14** 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 and XICON. The IP0H (Interrupt Priority 0 High) and EIP1H (Extended Interrupt Priority 1 High) registers make the four-level interrupt structure possible. The four priority level interrupt structure allows great flexibility in handling these interrupt sources.

14.1. Interrupt Structure

Table 14–1 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 be set previously. The 'Request Bits' can be set or cleared by software as well as hardware. It means the interrupts can be generated by software or the pending interrupts can be cancelled by 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.

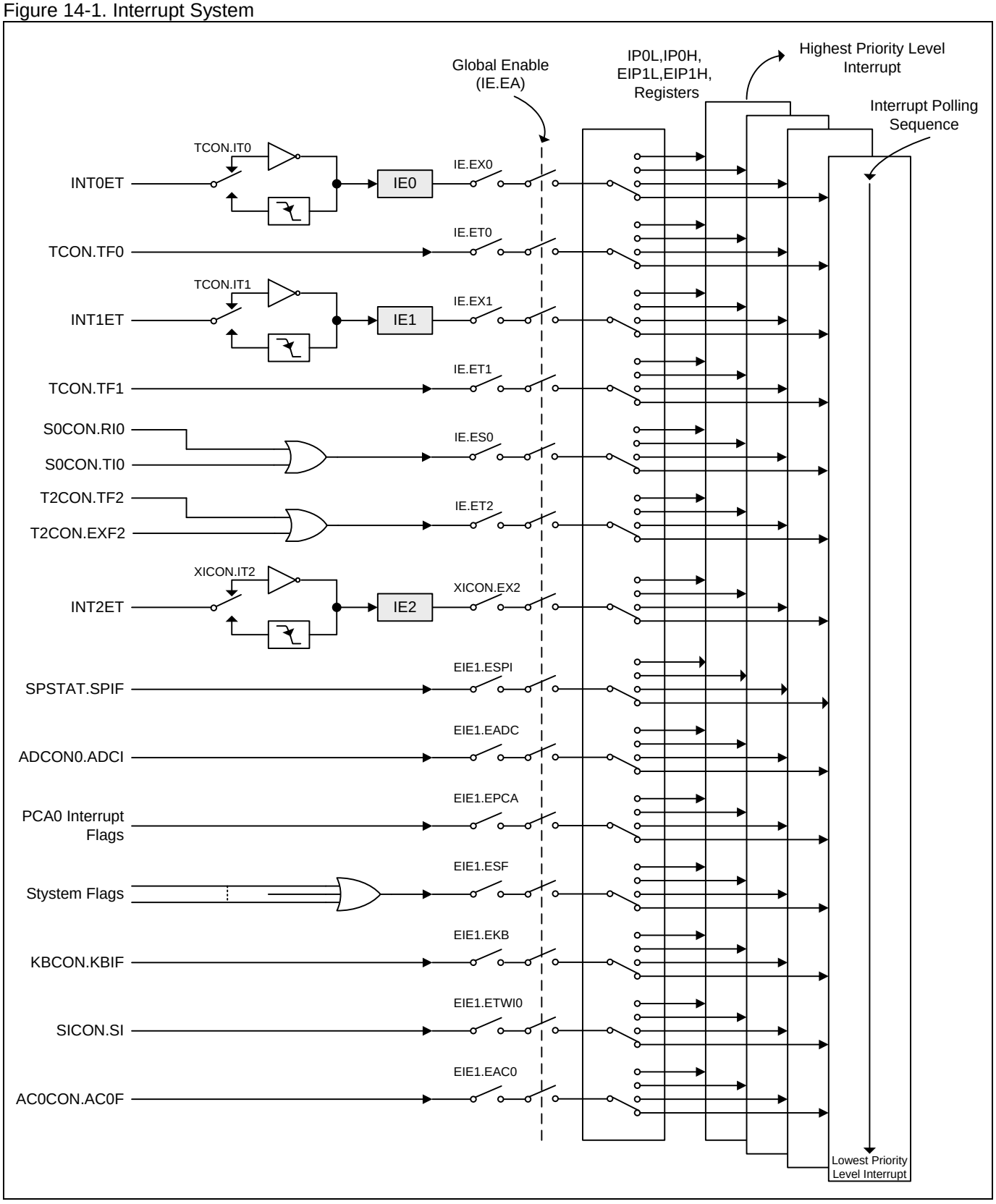
Figure 14–1 shows the interrupt system. Each of these interrupts will be briefly described in the following sections.

Table 14-1. Interrupt Sources

Interrupt #	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	[PS0H, PS0L]	...	0023H
5	Timer 2	ET2	TF2, EXF2	[PT2H, PT2L]	...	002Bh
6	External Interrupt 2, nINT2	EX2	IE2	[PX2H, PX2L]	...	0033H
7	SPI	ESPI	SPIF	[PSPIH, PSPIL]	...	003BH
8	ADC	EADC	ADCI	[PADCH, PADCL]	...	0043h
9	PCA0	EPCA	CF, CCFn (n=0~5)	[PPCAH, PPCAL]	...	004BH
10	System Flag	ESF	(Note 1)	[PSFH, PSFL]	...	0053H
11	Keypad Interrupt	EKB	KBIF	[PKBH, PKBL]	...	005BH
12	TWI0	ETWI0	SI	[PTWI0H, PTWI0L]	...	0063H
13	Analog Comparator 0	EAC0	AC0F	[PAC0H, PAC0L]	(Lowest)	006BH

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

Figure 14-1. Interrupt System



14.2. Interrupt Source

Table 14-2. Interrupt Source Flag

Interrupt #	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	T2CON.7 T2CON.6
6	External Interrupt 2,nINT2	IE2	XICON.1
7	SPI	SPIF	SPSTAT.7
8	ADC	ADCI	ADCON0.4
9	PCA0	CF, CCFn (n=0~5)	CCON.7 CCON.5~0
10	System Flag	WDTF, BOF1, BOF0, RTCF, MCDF, STAF, STOF, (TI0)	PCON1.0 PCON1.1 PCON1.2 PCON1.4 PCON1.5 AUXR2.7 AUXR2.6 S0CON.1
11	Keypad Interrupt	KBIF	KBCON.0
12	TWI0	SI	SICON.3
13	Analog Comparator 0	AC0F	AC0CON.4

The external interrupt nINT0, nINT1 and nINT2 can be either level-activated or transition-activated, depending on bits IT0 and IT1 in register TCON, and IT2 in register XICON. The flags that actually generate these interrupts are bits IE0 and IE1 in TCON, IE2 in XICON. When the interrupt is set to the transition-activated mode, the flag which is set by the external interrupt can be cleared by the on-chip hardware when the service vectored to the routine. Otherwise, the flag will be controlled by the request from external sources.

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 can be 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 timer2 interrupt is generated by the logical OR of TF2 and EXF2. 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 is set after SPI engine finishes a SPI transfer. It will not be cleared by hardware when the service routine is vectored to.

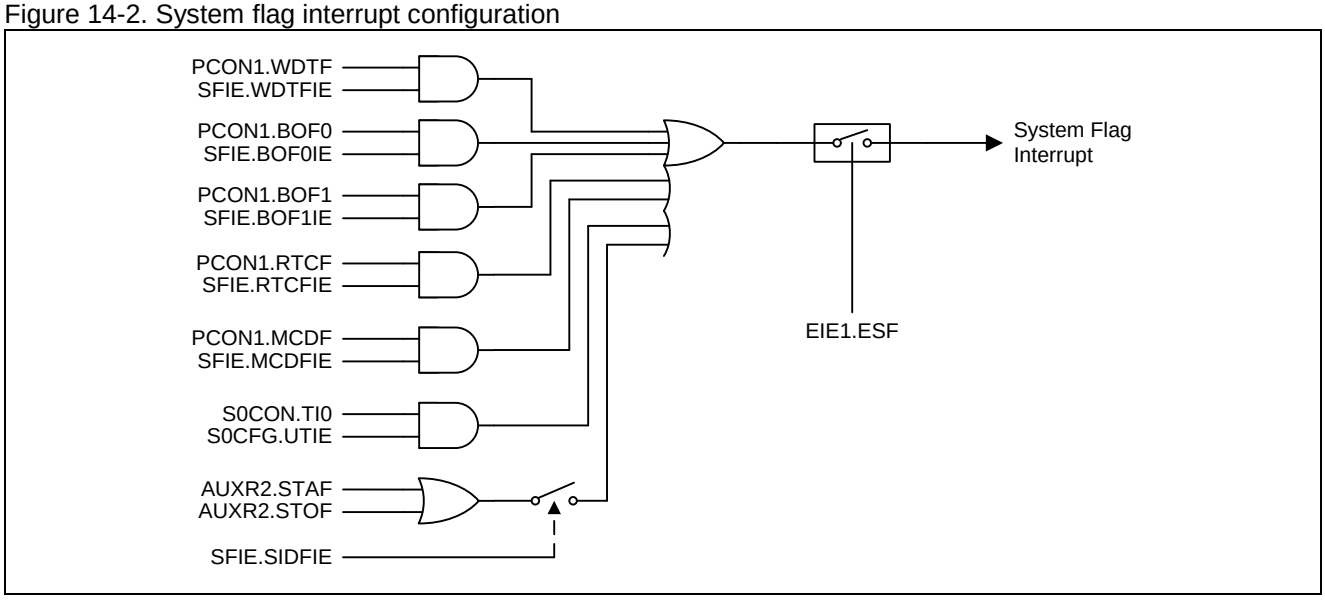
The ADC interrupt is generated by ADCI in ADCON0. It will not be cleared by hardware when the service routine is vectored to.

The PCA0 interrupt is generated by the logical OR of CF, CCF5, CCF4, CCF3, CCF2, CCF1 and CCF0 in CCON. None 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 MCDF, RTCF, BOF1, BOF0, WDTF, TI0, STAF and STOF. STAF and STOF are set by serial interface detection and stored in AUXR2. The Serial Port TI flag is optional to locate the

interrupt vector shared with system flag interrupt which is enabled by UTIE set. The rest flags are stored in PCON1. MCDF is set by Missing-Clock-Detection activated. RTCF is set by RTC counter overflow. BOF1 and BOF0 are set by on chip Brownout-Detector (BOD1 and BOD0) met the low voltage event. WDTF is set by Watch-Dog-Timer overflow. These flags will not be cleared by hardware when the service routine is vectored to. Figure 14–2 shows the system flag interrupt configuration.

Figure 14-2. 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 interrupt is generate by SI in SICON, which is set by TWI0 engine detecting a new bus state updated. It will not be cleared by hardware when the service routine is vectored to.

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.

All of the bits that generate interrupts can be set or cleared by software, which have the same result as hardware made. In other words, the software can generate the interrupts or cancel the pending interrupts.

14.3. Interrupt Enable

Table 14-3. Interrupt Enable

Interrupt #	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	PCA0	EPCA	EIE1.2
10	System Flag	ESF	EIE1.3
11	Keypad Interrupt	EKB	EIE1.5
12	TWI0	ETWI0	EIE1.6
13	Analog Comparator 0	EAC0	EIE1.7

There are **14** interrupt sources available in **MG82FG5D16**. Each of these interrupt sources can be individually enabled or disabled by setting or clearing an interrupt enable bit in the registers IE, EIE1 and XICON. IE also contains a global disable bit, EA, which can be cleared to disable all interrupts at once. If EA is set to ‘1’, the interrupts are individually enabled or disabled by their corresponding enable bits. If EA is cleared to ‘0’, all interrupts are disabled.

14.4. 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 14–1) determine the priority level of each interrupt. IP0L, IP0H, EIP1L and EIP1H are combined to 4-level priority interrupt. Table 14–4 shows the bit values and priority levels associated with each combination.

Table 14-4. 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 not be 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 14–2 shows the internal polling sequence in the same priority level and the interrupt vector address.

14.5. 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:

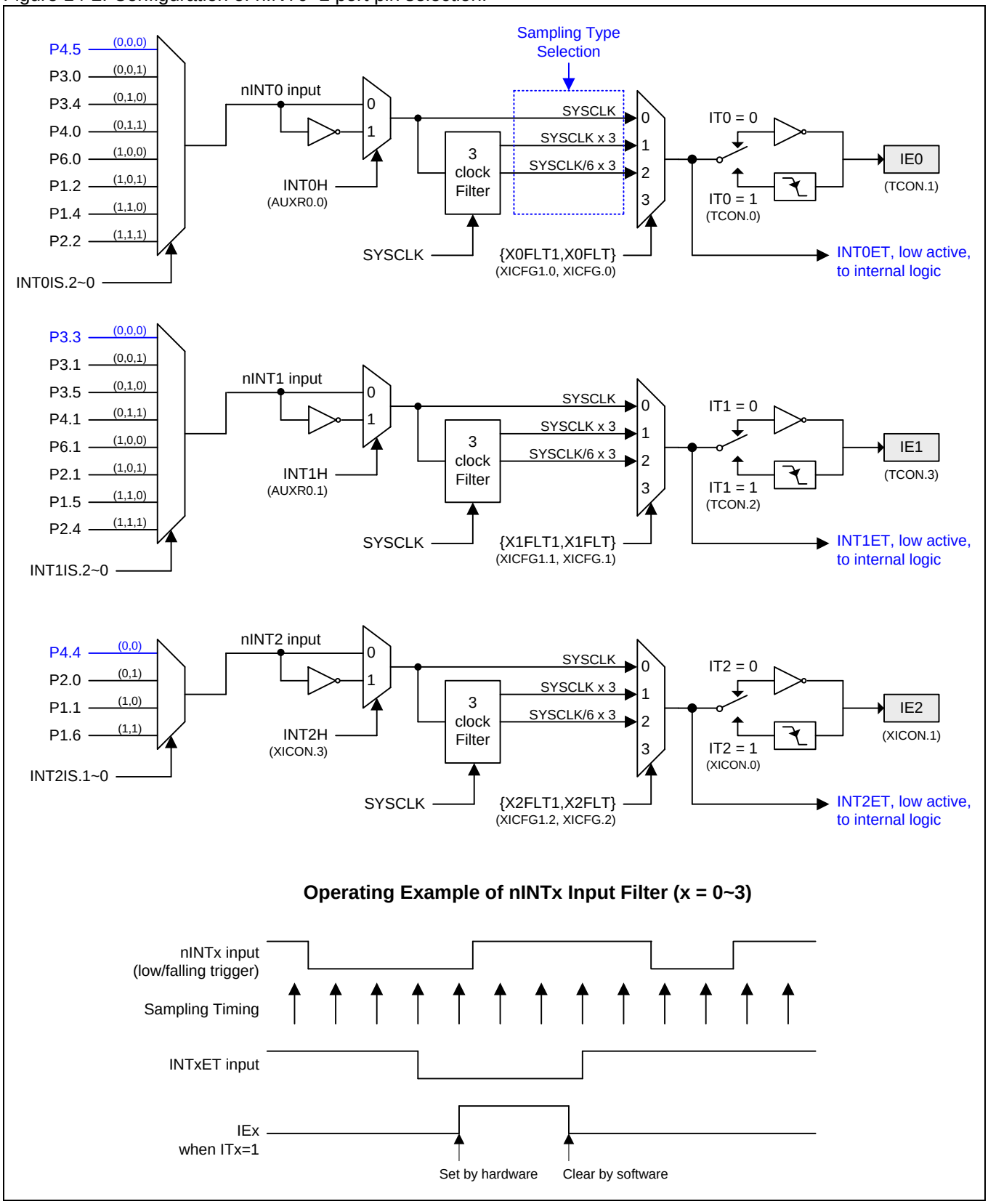
- An interrupt of equal or higher priority level is already in progress.
- The current cycle (polling cycle) is not the final cycle in the execution of the instruction in progress.
- The instruction in progress is RETI or any write to the IE, IP0L, IPH, EIE1, EIP1L, EIP1H, and XICON registers.

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.

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

The **MG82FG5D16** provides flexible port pin selection for nINT0, nINT1 and nINT2 input.

Figure 14-2. Configuration of nINT0~2 port pin selection.



14.7. 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 = 0x00-0000

7	6	5	4	3	2	1	0
EA	0	ET2	ES0	ET1	EX1	ET0	EX0
R/W	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: Reserved. Software must write “0” on this bit when IE is written.

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

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

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

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

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

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

AUXR0: Auxiliary Register 0

SFR Page = 0~F
SFR Address = 0xA1 RESET = 000x-xx00

7	6	5	4	3	2	1	0
P60OC1	P60OC0	P60FD	--	--	--	INT1H	INT0H
R/W	**R/W**	**R/W**	**W**	**W**	**W**	**R/W**	**R/W**

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 = xxxx-0000

7	6	5	4	3	2	1	0
0	0	0	0	INT2H	EX2	IE2	IT2
W	**W**	**W**	**W**	**R/W**	**R/W**	**R/W**	**R/W**

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

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 = x000-0000

7	6	5	4	3	2	1	0
0	PX2L	PT2L	PSL	PT1L	PX1L	PT0L	PX0L
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 IP0L is written.
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 = x000-0000

7	6	5	4	3	2	1	0
0	PX2H	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
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 IP0H is written.
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 = 000x-0000

7	6	5	4	3	2	1	0
EAC0	ETWI0	EKB	--	ESF	EPCA	EADC	ESPI
R/W	**R/W**	**R/W**	**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 interrupt.
0: Disable TWI0 interrupt.
1: Enable TWI0 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: Reserved. Software must write “0” on this bit when EIE1 is written.

Bit 3: ESF, Enable System Flag interrupt.
0: Disable the interrupt when the group of {MCDF, RTCF, BOF1, BOF0, WDTF} in PCON1, {STAF, STOF} in AUXR3 or TI0 with UTIE is set.
1: Enable the interrupt of the flags of {MCDF, RTCF, BOF1, BOF0, WDTF} in PCON1, {STAF, STOF} in AUXR3, or TI0 with UTIE when the associated system flag interrupt is enabled in SFIE.

Bit 2: EPCA, Enable PCA0 interrupt.
0: Disable PCA0 interrupt.
1: Enable PCA0 interrupt.

Bit 1: EADC, Enable ADC Interrupt.
0: Disable the interrupt when ADCON0.ADCI is set in ADC module.
1: Enable the interrupt when ACCON0.ADCI is set in ADC module.

Bit 0: ESPI, Enable SPI Interrupt.
0: Disable the interrupt when SPSTAT.SPIF is set in SPI module.
1: Enable the interrupt when SPSTAT.SPIF is set in SPI module.

EIP1L: Extended Interrupt Priority 1 Low Register

SFR Page = 0~F
SFR Address = 0xAE RESET = 000x-0000

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

Bit 7: PAC0L, AC0 interrupt priority-L register.
Bit 6: PTWI0L, TWI0 interrupt priority-L register.
Bit 5: PKBL, keypad interrupt priority-L register.
Bit 4: Reserved. Software must write “0” on this bit when EIP1L is written.
Bit 3: PSFL, system flag interrupt priority-L register.
Bit 2: PPCAL, PCA0 interrupt priority-L register.
Bit 1: PADCL, ADC interrupt priority-L register.
Bit 0: PSPIL, SPI interrupt priority-L register.

EIP1H: Extended Interrupt Priority 1 High Register

SFR Page = 0~F
SFR Address = 0xAF RESET = 000x-0000

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

Bit 7: PAC0H, AC0 priority-H register.
Bit 6: PTWI0H, TWI0 interrupt priority-H register.
Bit 5: PKBH, keypad interrupt priority-H register.
Bit 7: Reserved. Software must write “0” on this bit when EIP1H is written.
Bit 3: PSFH, system flag interrupt priority-H register.
Bit 2: PPCAH, PCA0 interrupt priority-H register.
Bit 1: PADCH, ADC interrupt priority-H register.
Bit 0: PSPIH, SPI interrupt priority-H register.

XICFG: External Interrupt Configured Register

SFR Page = **0 only**
SFR Address = 0xC1 RESET = 0000-x000

7	6	5	4	3	2	1	0
INT1IS.1	INT1IS.0	INT0IS.1	INT0IS.0	0	X2FLT	X1FLT	X0FLT
R/W	**R/W**	**R/W**	**R/W**	**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
000	P3.3
001	P3.1
010	P3.5
011	P4.1
100	P6.1
101	P2.1
110	P1.5
111	P2.4

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

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

Bit 3: Reserved. Software must write “0” on this bit when XICFG is written.

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

X2FLT1, X2FLT	nINT2 input filter mode
00	Disabled
01	SYSCLK x 3
10	SYSCLK/6 x 3
11	Reserved

Bit 1: X1FLT, nINT1 Filter mode control. It selects nINT1 input filter mode with X1FLT1 (XICFG1.1)

X1FLT1, X1FLT	nINT1 input filter mode
00	Disabled
01	SYSCLK x 3
10	SYSCLK/6 x 3
11	Reserved

Bit 0: X0FLT, nINT0 Filter mode control. It selects nINT0 input filter mode with X0FLT1 (XICFG1.0)

X0FLT1, X0FLT	nINT0 input filter mode
00	Disabled
01	SYSCLK x 3
10	SYSCLK/6 x 3
11	Reserved

XICFG1: External Interrupt Configured 1 Register

SFR Page = **1 only**

SFR Address = 0xC1 RESET = 0000-x000

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

INT2IS.1~0	Selected Port Pin of nINT2
00	P4.4
01	P2.0
10	P1.1
11	P1.6

Bit 3: Reserved. Software must write “0” on this bit when XICFG1 is written.

Bit 2: X2FLT1, nINT2 Filter mode control. It selects nINT2 input filter mode with X2FLT (XICFG.2). Refer XICFG description for nINT2 input filter mode definition.

Bit 1: X1FLT1, nINT1 Filter mode control. It selects nINT1 input filter mode with X1FLT (XICFG.1). Refer XICFG description for nINT1 input filter mode definition.

Bit 0: X0FLT1, nINT0 Filter mode control. It selects nINT0 input filter mode with X0FLT (XICFG.0). Refer XICFG description for nINT0 input filter mode definition.

SFIE: System Flag Interrupt Enable Register

SFR Page = 0~F

SFR Address = 0x8E RESET = 0110-x000

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

Bit 7: SIDFIE, Serial Interface 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: Reserved. Software must write “0” on this bit when SFIE is written.

Bit 2: BOF1IE, Enable BOF1 (PCON1.2) Interrupt.
0: Disable BOF1 interrupt.
1: Enable BOF1 interrupt.

Bit 1: BOF0IE, Enable BOF0 (PCON1.1) Interrupt.
0: Disable BOF0 interrupt.
1: Enable BOF0 interrupt.

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

PCON1: Power Control Register 1

SFR Page = 0~F & P

SFR Address = 0x97 POR = 0000-x000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	0	BOF1	BOF0	WDTF
R/W	**R/W**	**R/W**	**R/W**	**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. To enable this function must combine XTALE & OSCS[1:0] = 01, external crystal mode.

Bit 4: RTCF, RTC overflow flag.
0: This bit must be cleared by software writing **"1"** on it. Software writing "0" is no operation.
1: This bit is only set by hardware when RTCCT overflows. Writing **"1"** on this bit will clear RTCF.

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

Bit 2: BOF1, Brown-Out Detection flag 1.
0: This bit must be cleared by software writing **"1"** to it.
1: This bit is set by hardware if the operating voltage matches the detection level of Brown-Out Detector 1 (4.2V/3.7/2.4/2.0).

Bit 1: BOF0, Brown-Out Detection flag 0.
0: This bit must be cleared by software writing **"1"** to it.
1: This bit is set by hardware if the operating voltage matches the detection level of Brown-Out Detector 0 (2.2V).

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~F

SFR Address = 0xA3 RESET = 00xx-0000

7	6	5	4	3	2	1	0
STAF	STOF	--	--	T1X12	T0X12	T1CKOE	T0CKOE
R/W	**R/W**	**W**	**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.
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.

15. Timers/Counters

MG82FG5D16 has **three** 16-bit Timers/Counters: Timer 0, Timer 1 and Timer 2. 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 serves to count the standard C51 machine cycle. AUXR2.T0X12, AUXR2.T1X12 and T2MOD.T2X12 are the functions for Timer 0/1/2 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, or T2. 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.

15.1. Timer 0 and Timer 1

15.1.1. 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. The counted input is enabled to the timer when TRx = 1 and either TxGATE=0 or INTxET = 1. Mode 0 operation is the same for Timer0 and Timer1. The PWM function of Timer 0/1 is shown in Figure 15–1 and Figure 15–2.

Figure 15-1. Timer 0 Mode 0 Structure

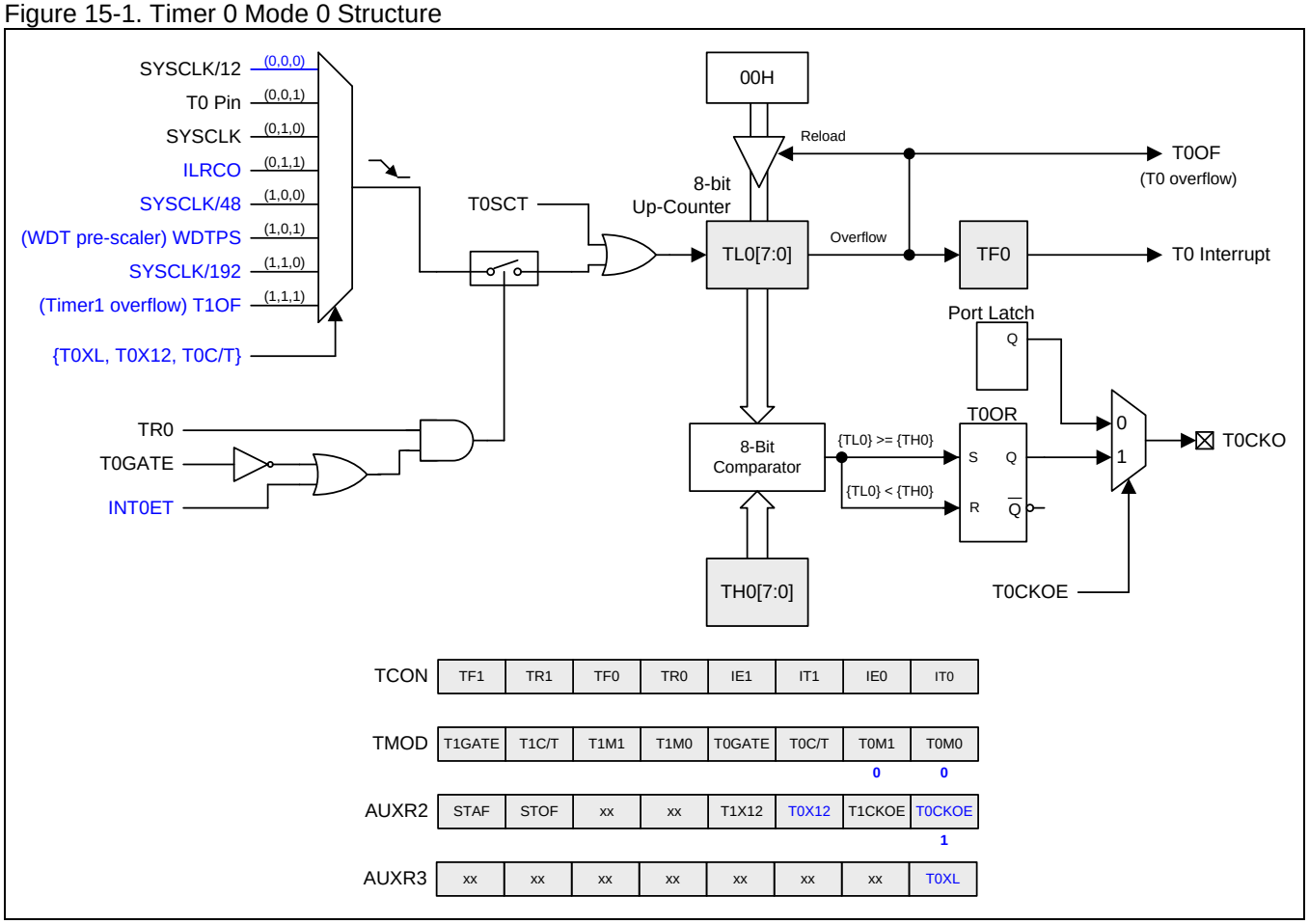
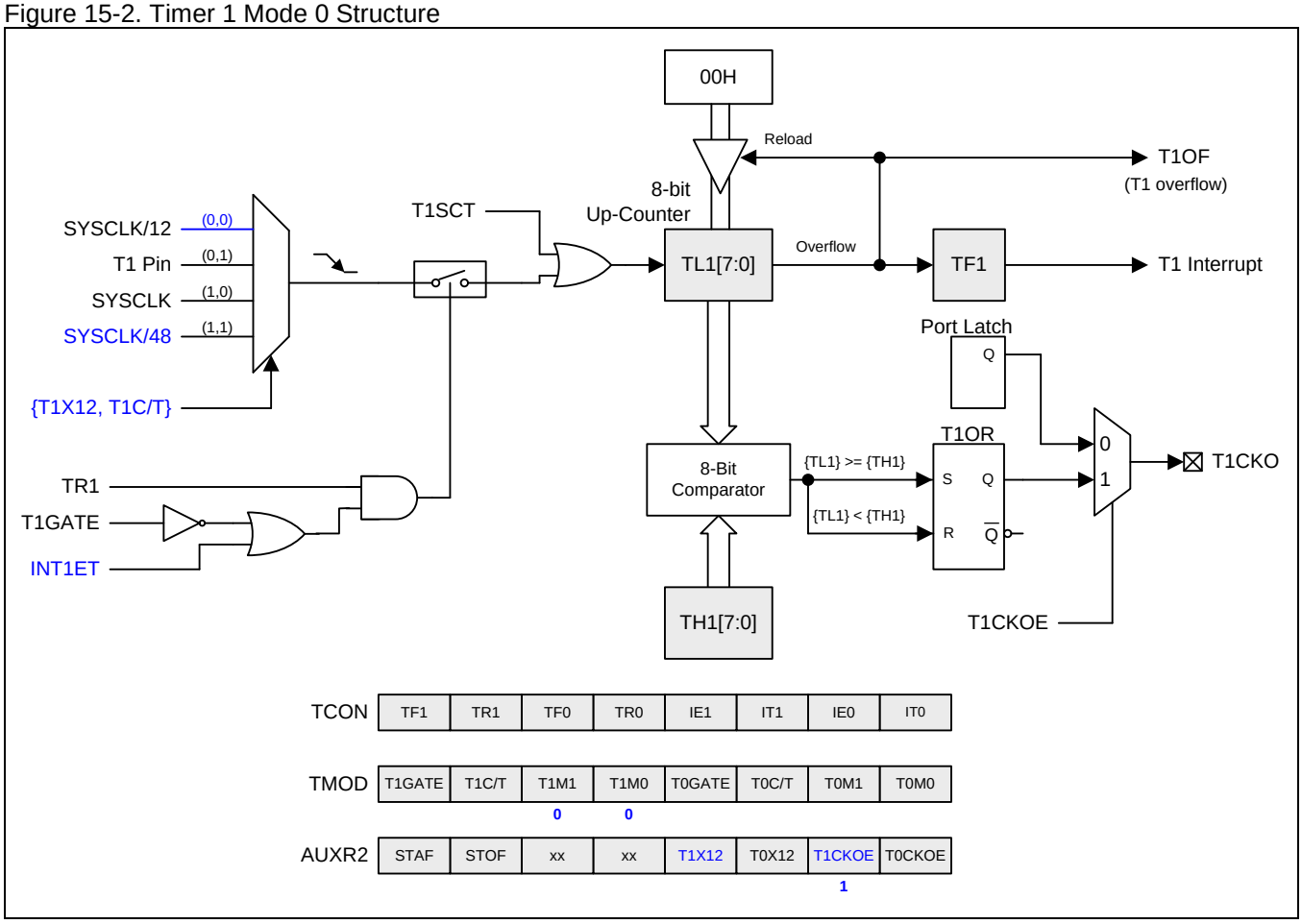


Figure 15-2. Timer 1 Mode 0 Structure



15.1.2. Timer 0/1 Mode 1

Timer 0/1 in Mode1 is configured as a 16 bit timer or counter. The function of TxGATE, INTxET and TRx is same as mode 0. Figure 15–3 and Figure 15–4 show the mode 1 structure of Timer 0 and Timer 1.

Figure 15-3. Timer 0 Mode 1 Structure

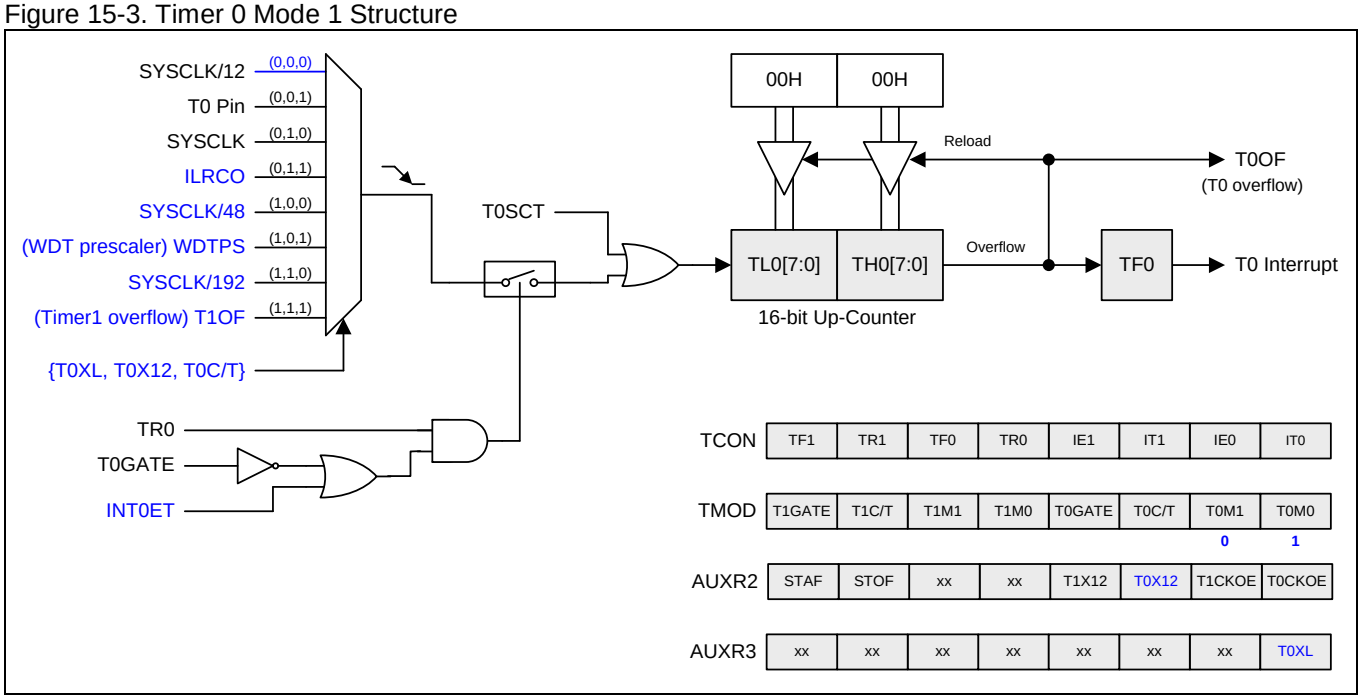
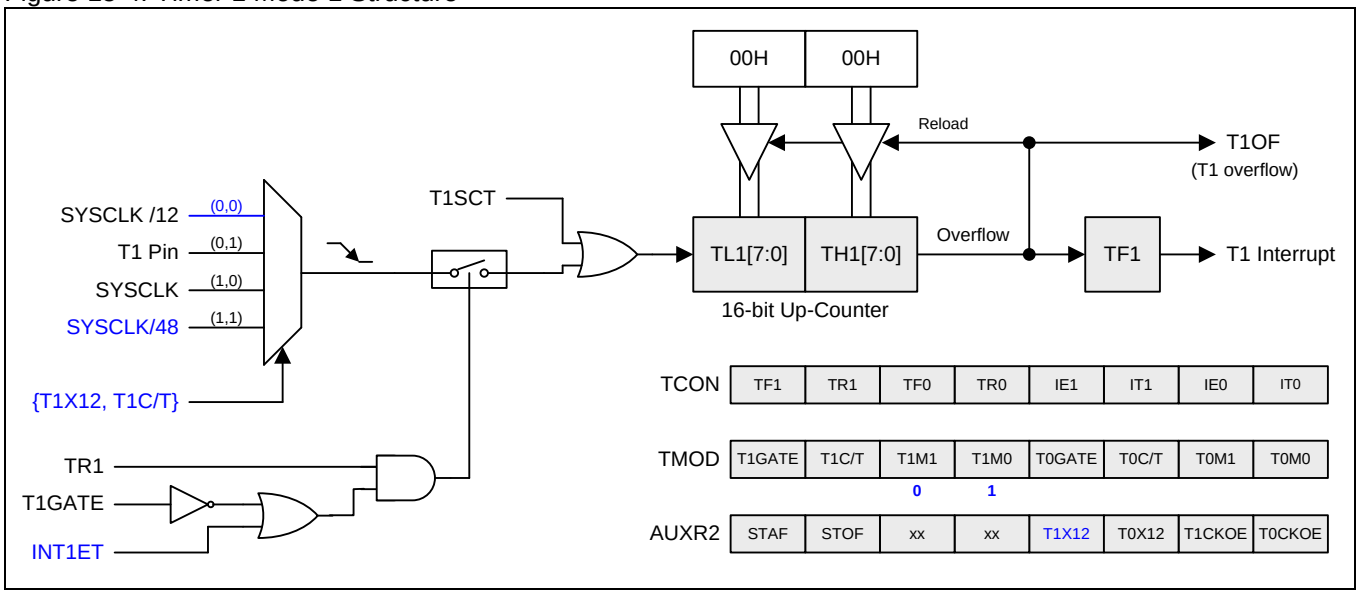


Figure 15-4. Timer 1 Mode 1 Structure



15.1.3. 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 15–5 and Figure 15–6 show the mode 2 structure of Timer 0 and Timer 1.

Figure 15-5. Timer 0 Mode 2 Structure

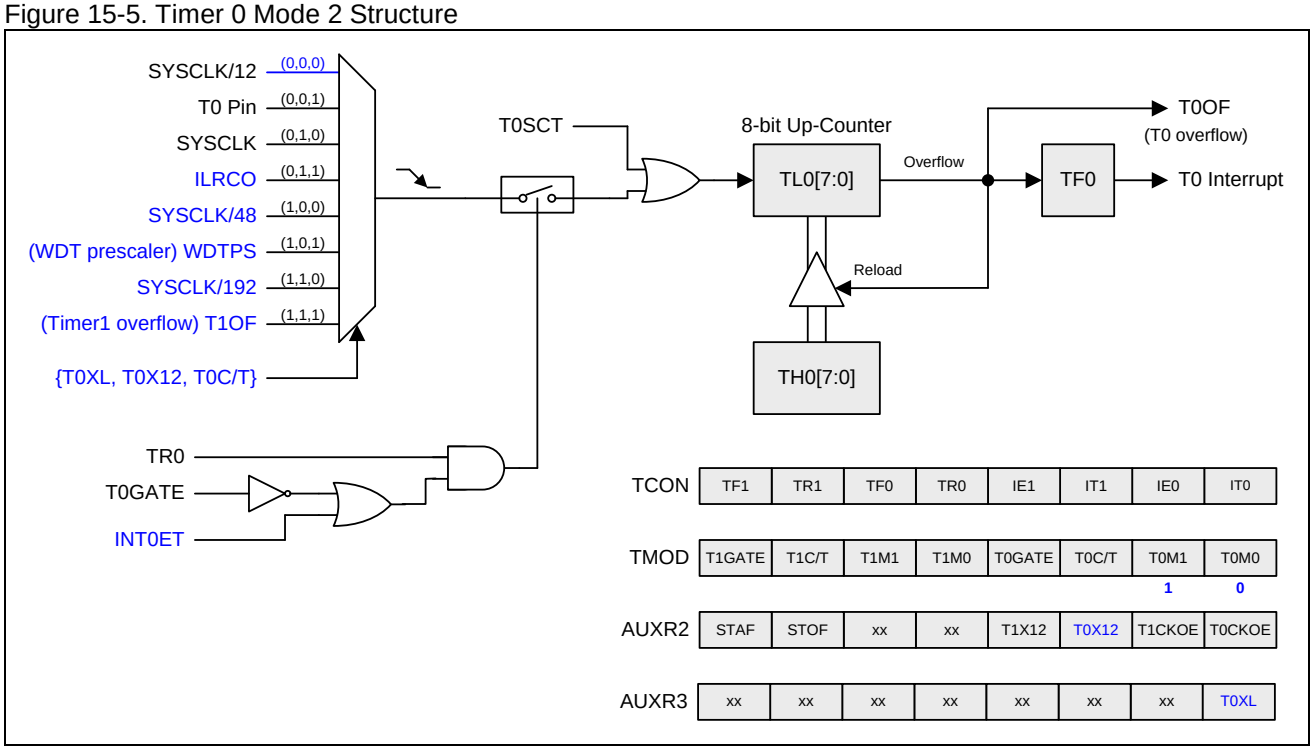
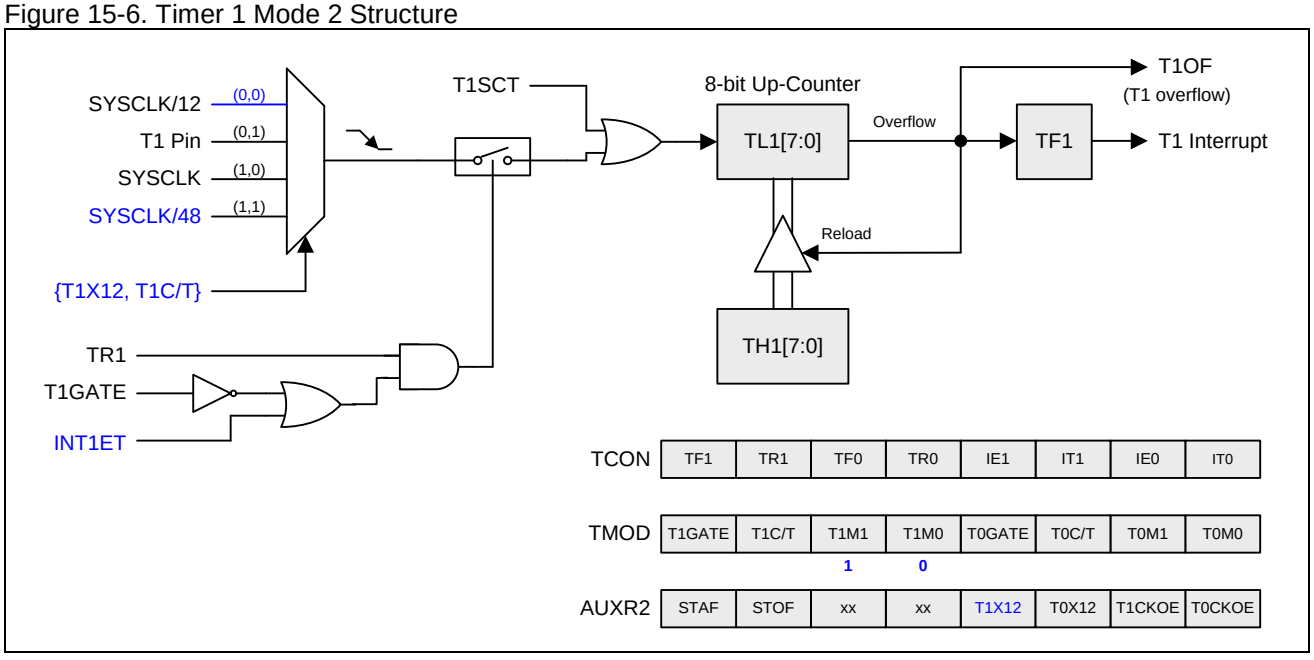


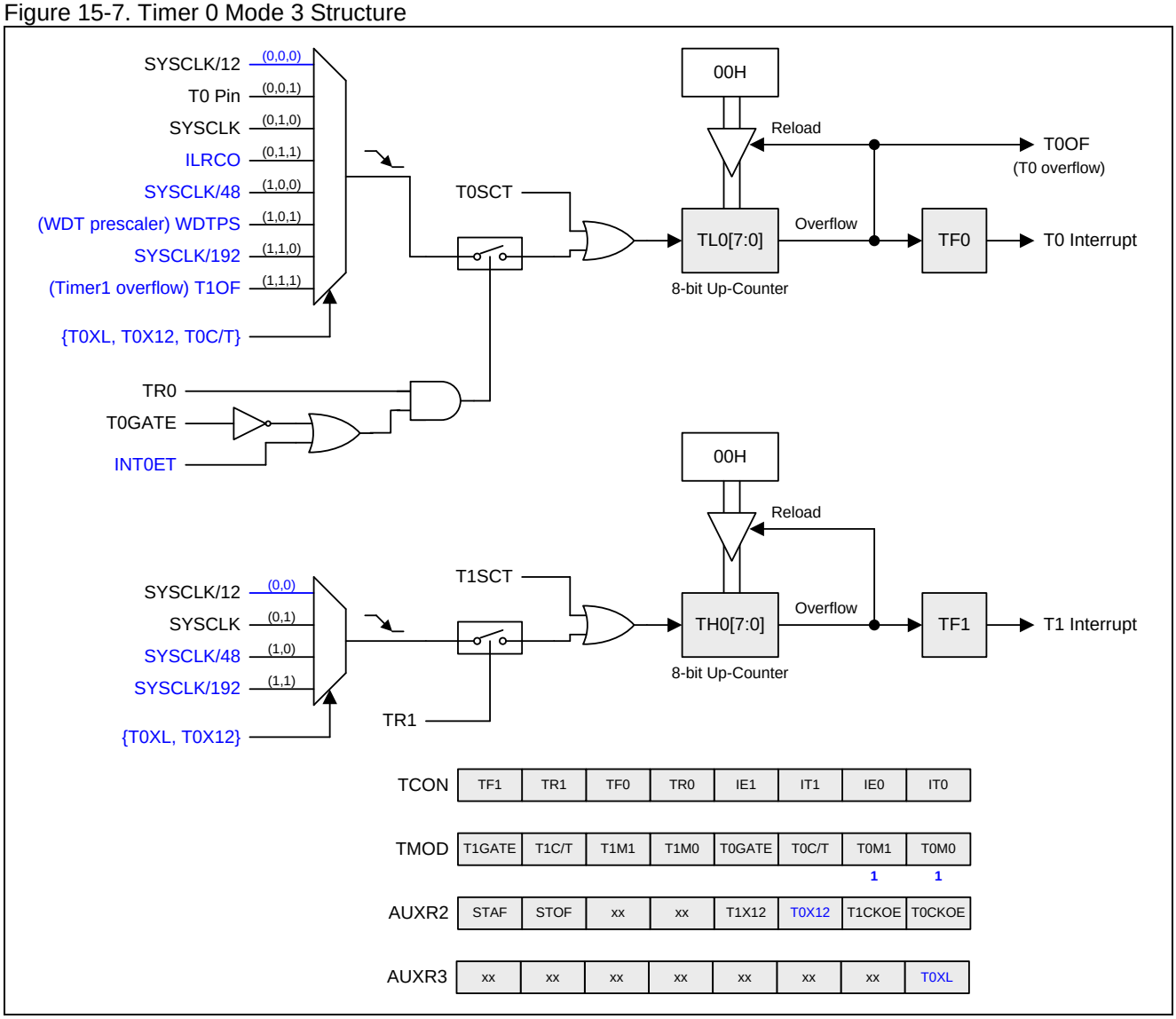
Figure 15-6. Timer 1 Mode 2 Structure



15.1.4. Timer 0/1 Mode 3

Timer1 in Mode3 simply holds its count, the effect is the same as setting TR1 = 1. Timer0 in Mode 3 enables TL0 and TH0 as two separate 8-bit counters. TL0 uses the Timer0 control bits such like C/T, T0GATE, TR0, INT0ET and TF0. TH0 is locked into a timer function (can not be external event counter) and take over the use of TR1, TF1 from Timer1. TH0 now controls the Timer1 interrupt. Figure 15–7 shows the mode 3 structure of Timer 0.

Figure 15-7. Timer 0 Mode 3 Structure



15.1.5. 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 15–8 and Figure 15–9 formula gives the formula of Timer 0 and Timer 1 clock-out frequency. Figure 15–10 and Figure 15–11 show the clock-out structure of Timer 0 and Timer 1.

Figure 15-8. Timer 0 clock out equation

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

Figure 15-9. 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 15-10. Timer 0 in Clock Output Mode

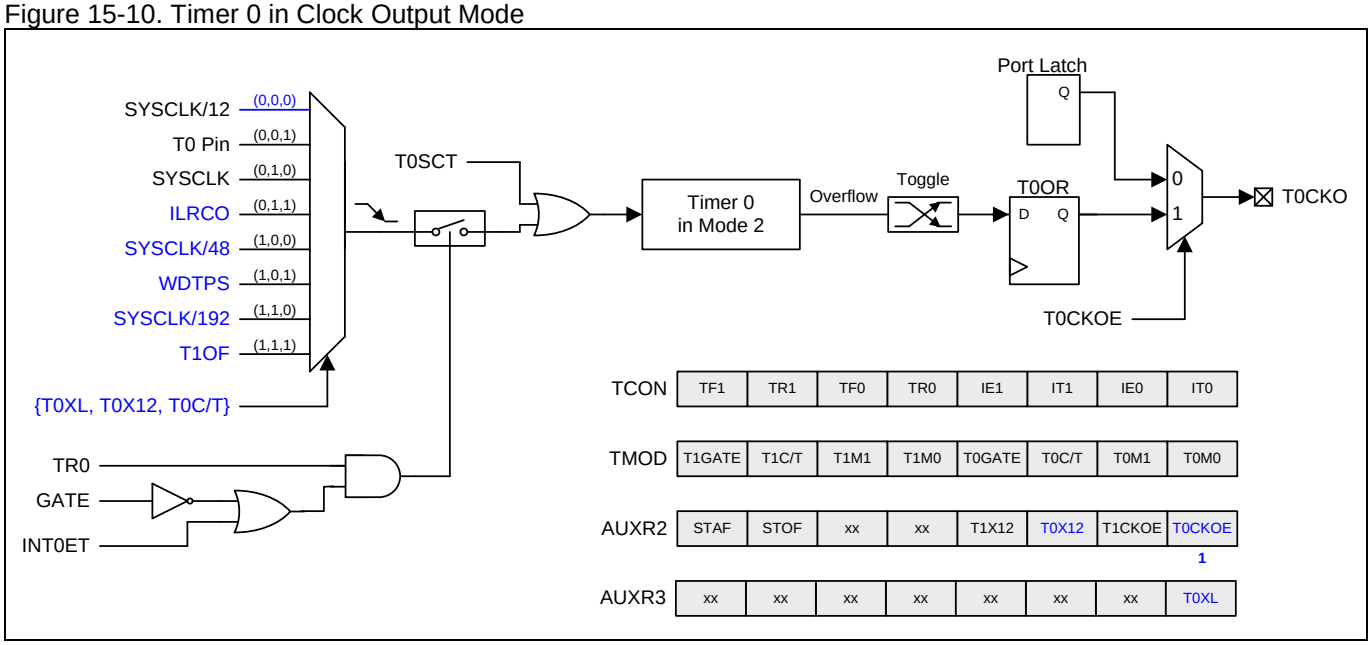
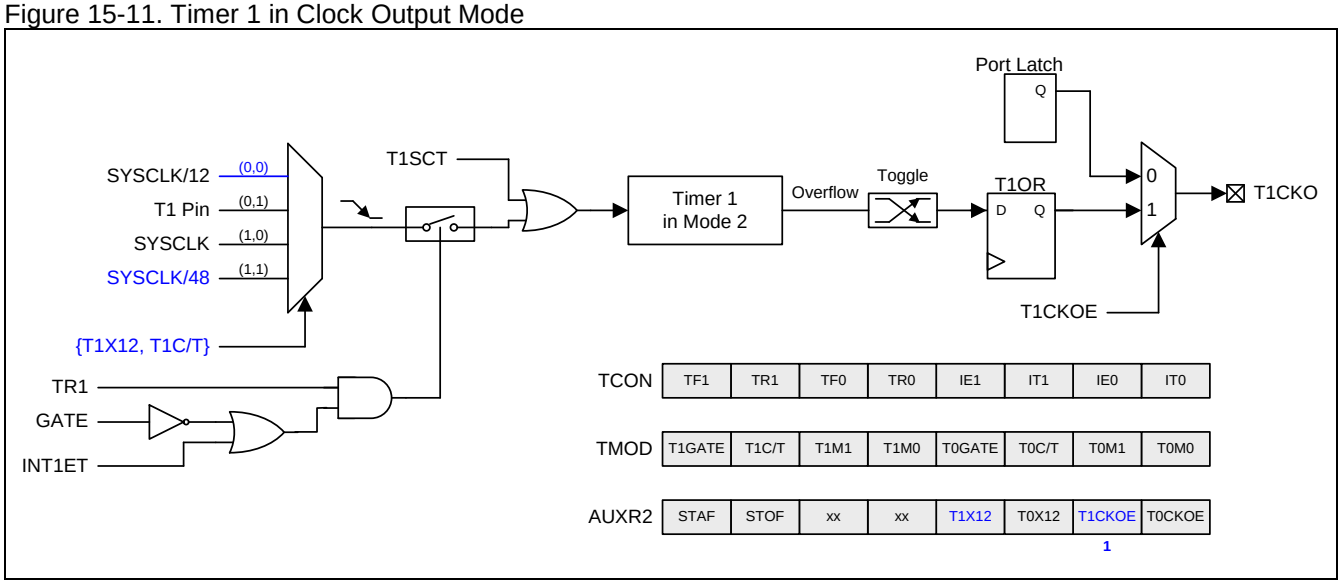


Figure 15-11. Timer 1 in Clock Output Mode



How to Program Timer 0/1 in Clock-out Mode

- Select Timer 0/1 clock source.
- Determine the 8-bit reload value from the formula and enter it in the TH0/TH1 register.
- Enter the same reload value as the initial value in the TL0/TL1 register.
- Set T0CKOE/T1CKOE bit in AUXR2 register.
- Set TR0/TR1 bit in TCON register to start the Timer 0/1.

In the Clock-Out mode, Timer 0/1 rollovers will 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.

15.1.6. 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.
0: Disable gating control for Timer1.
1: Enable gating control for Timer1. When set, Timer1 or Counter1 is enabled only when iINT1 input is high and TR1 control bit is set.

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: Timer 1 operating mode selection.

T1M1~0	Timer 1 Operating Mode
00	8-bit PWM generator for Timer1
01	16-bit timer/counter for Timer1
10	8-bit timer/counter with automatic reload for Timer1
11	Timer/Counter1 Stopped

Bit 3: T0Gate, Gating control for Timer0.
0: Disable gating control for Timer0.
1: Enable gating control for Timer0. When set, Timer0 or Counter0 is enabled only when iINT0 input is high and TR0 control bit is set.

Bit 6: 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: Timer 0 operating mode selection.

T0M1~0	Timer 0 Operating Mode
00	8-bit PWM generator for Timer0
01	16-bit timer/counter for Timer0
10	8-bit timer/counter with automatic reload for Timer0
11	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 = 00xx-0000

7	6	5	4	3	2	1	0
STAF	STOF	--	--	T1X12	T0X12	T1CKOE	T0CKOE
R/W	R/W	W	W	R/W	R/W	R/W	R/W

When selecting the asynchronous clock (external clock from I/O pin or ILRCO) as the timer clock source, it's frequency needs to lower then 1/2 of the system clock to ensure the edge event can be latched to trigger the counter.

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

Bit 1: T1CKOE, Timer 1 Clock Output Enable.
0: Disable Timer 1 clock output.
1: Enable Timer 1 clock output on T1CKO Port pin.

Bit 0: T0CKOE, Timer 0 Clock Output Enable.
0: Disable Timer 0 clock output.
1: Enable Timer 0 clock output on T0CKO Port pin.

AUXR3: Auxiliary Register 3

SFR Page = **0 only**
SFR Address = 0xA4 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
00	P3.4
01	P4.4
10	P2.2
11	P2.6

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 = 0000-0x00

7	6	5	4	3	2	1	0
T2PS1	T2PS0	T1PS1	T1PS0	SPIPS0	--	AC0OE	AC0FLT1
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**W**	**R/W**	**R/W**

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

T1PS1~0	T1/T1CKO
00	P3.5
01	P4.5
10	Reserved
11	P2.6

15.2. Timer 2

Timer 2 is a 16-bit Timer/Counter which can operate either as a timer or an event counter which function is defined on 8 clock sources, as selected by the bits in T2CON, T2MOD and T2MOD1 register. Timer 2 has five main functions: Capture, Reload Timer, PWM, Baud Rate Generator and Programmable Clock-Out by operation mode definition. It also provides a selection on 8 external signals for capture event or Timer 2 external interrupt source, EXF2. There are two interrupt sources in Timer2, TF2 and EXF2. TF2 is the TH2 overflow flag and its interrupt function can be blocked by TF2IG.

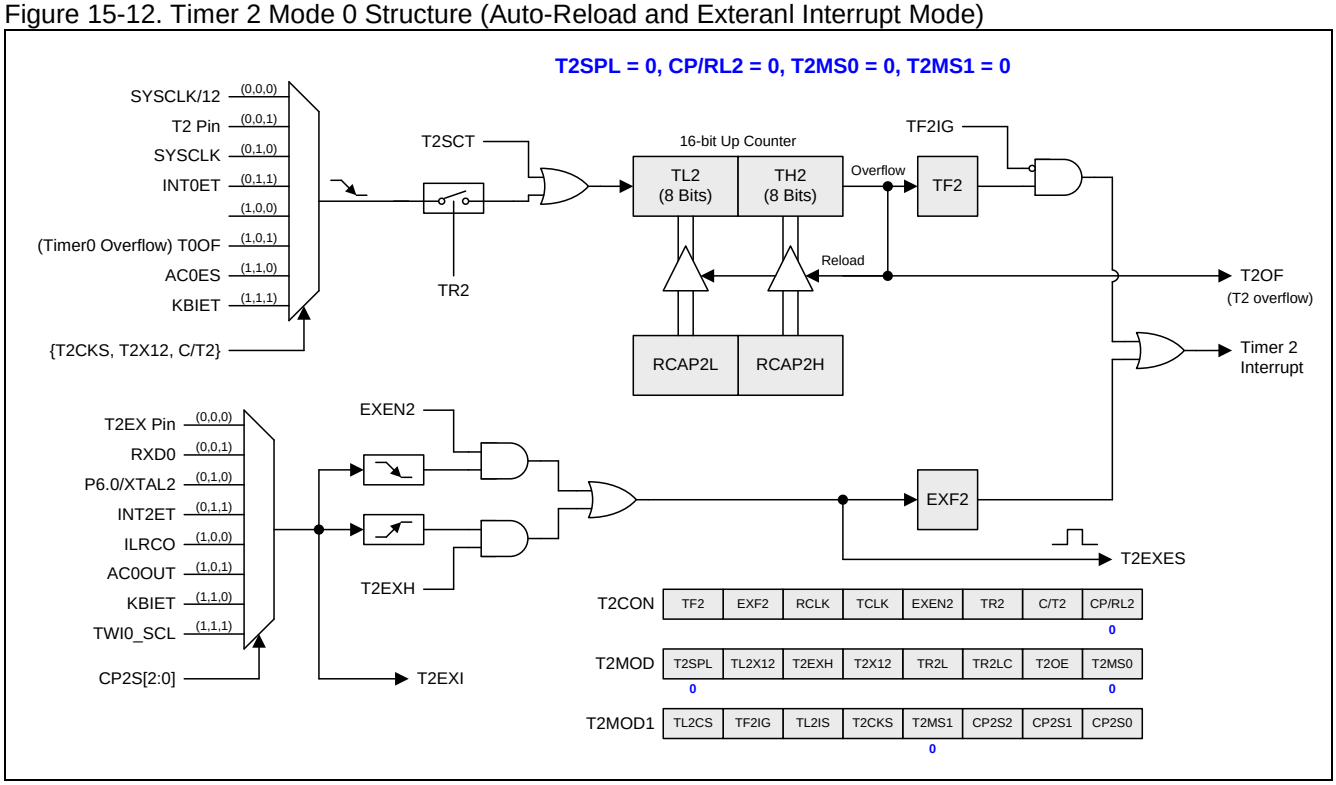
15.2.1. Timer 2 Mode 0 (Auto-Reload and External Interrupt)

In this mode, Timer 2 provides an 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 15–12.

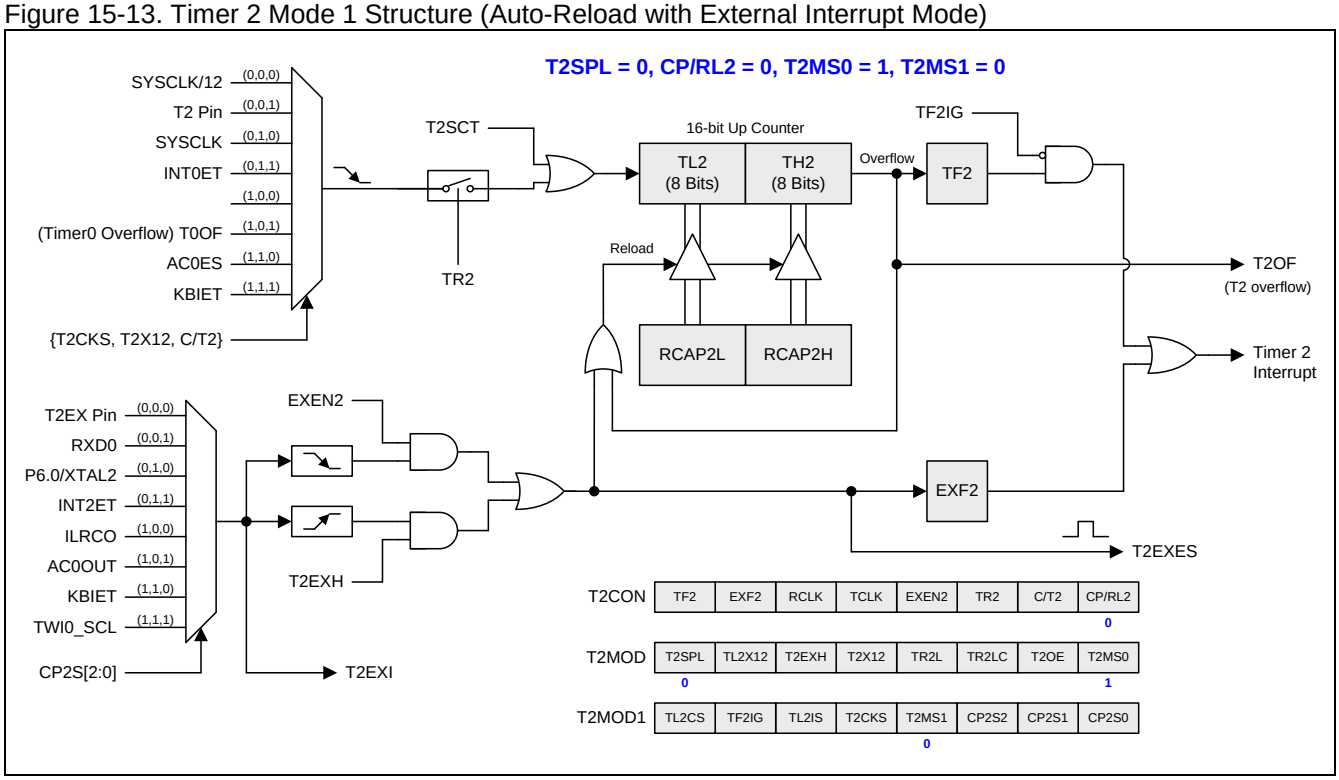
Figure 15-12. Timer 2 Mode 0 Structure (Auto-Reload and Exteranl Interrupt Mode)



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

Figure 15–13 shows Timer 2 Mode 1, which enables Timer 2 to count up automatically. In this mode there are two options selected by bit EXEN2 in T2CON register. If EXEN2=0, then Timer 2 counts up to 0xFFFFH and sets the TF2 (Overflow Flag) bit upon overflow. This causes the Timer 2 registers to be reloaded with the 16-bit value in RCAP2L and RCAP2H. The values in RCAP2L and RCAP2H are preset by firmware. If EXEN2=1, then a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at T2EXI, one of 8 Timer 2 external inputs. This transition also sets the EXF2 bit. The Timer 2 interrupt, if enabled, can be generated when either TF2 or EXF2 are 1. T2EXH performs the same function as EXEN2 but it enables the detecting a 0-to-1 transition at T2EXI input.

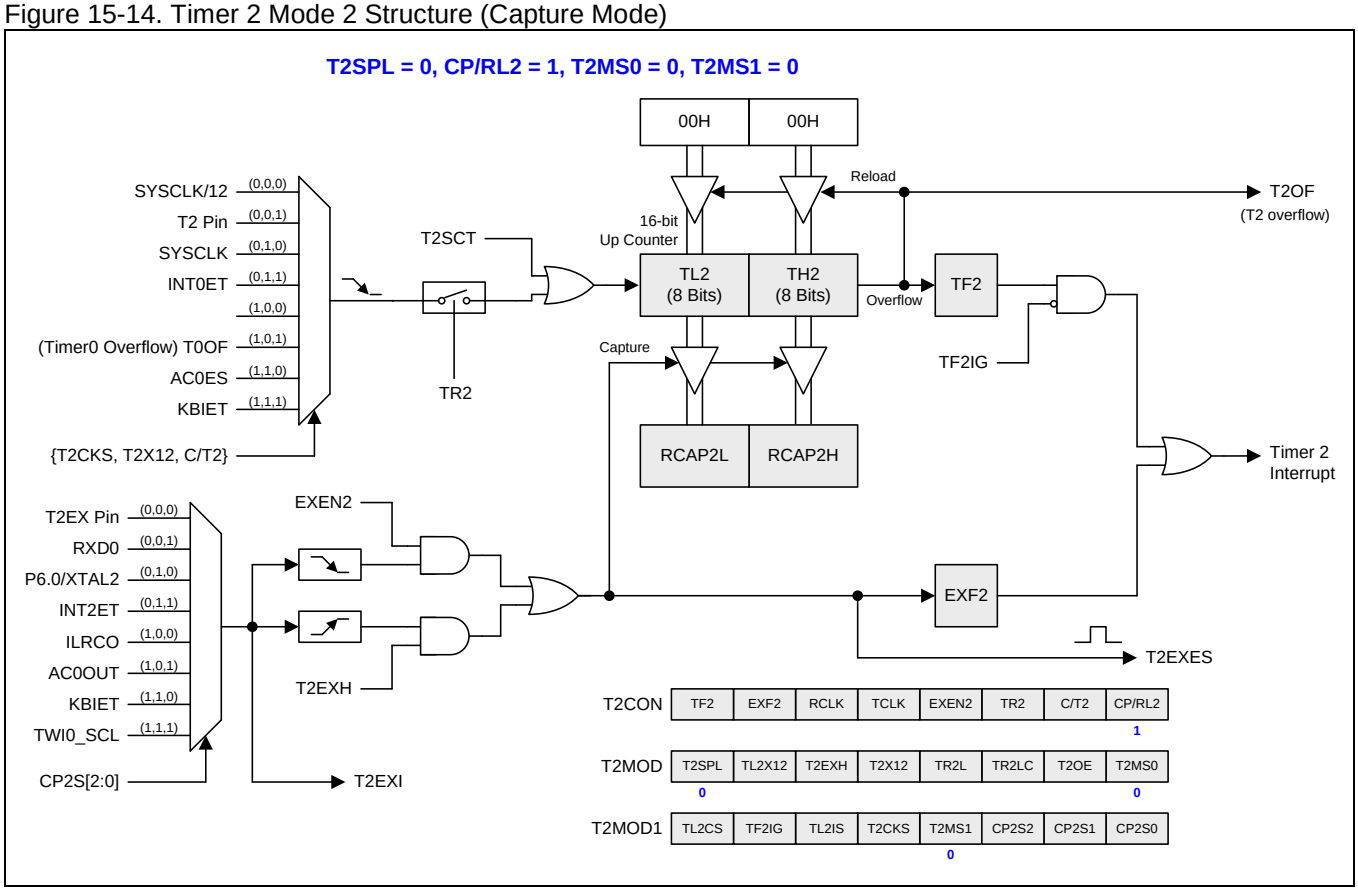
Figure 15-13. Timer 2 Mode 1 Structure (Auto-Reload with External Interrupt Mode)



15.2.3. Timer 2 Mode 2 (Capture)

Figure 15–14 shows the capture mode there are two options selected by bit EXEN2 in T2CON. If EXEN2=0, Timer 2 is a 16-bit timer or counter which, upon overflow, sets bit TF2 (Timer 2 overflow flag). This bit can then be used to generate an interrupt (by enabling the Timer 2 interrupt bit in the IE register). If EXEN2=1, Timer 2 still does the above, but with the added feature that a 1-to-0 transition at T2EXI, one of 8 Timer 2 external inputs, that causes the current value in the Timer 2 registers, TH2 and TL2, to be captured into registers RCAP2H and RCAP2L, respectively. In addition, the transition at T2EXI causes bit EXF2 in T2CON to be set, and the EXF2 bit (like TF2) can generate an interrupt which vectors to the same location as Timer 2 overflow interrupt. T2EXH performs the same function as EXEN2 but it enables the detecting a 0-to-1 transition at T2EXI input.

Figure 15-14. Timer 2 Mode 2 Structure (Capture Mode)

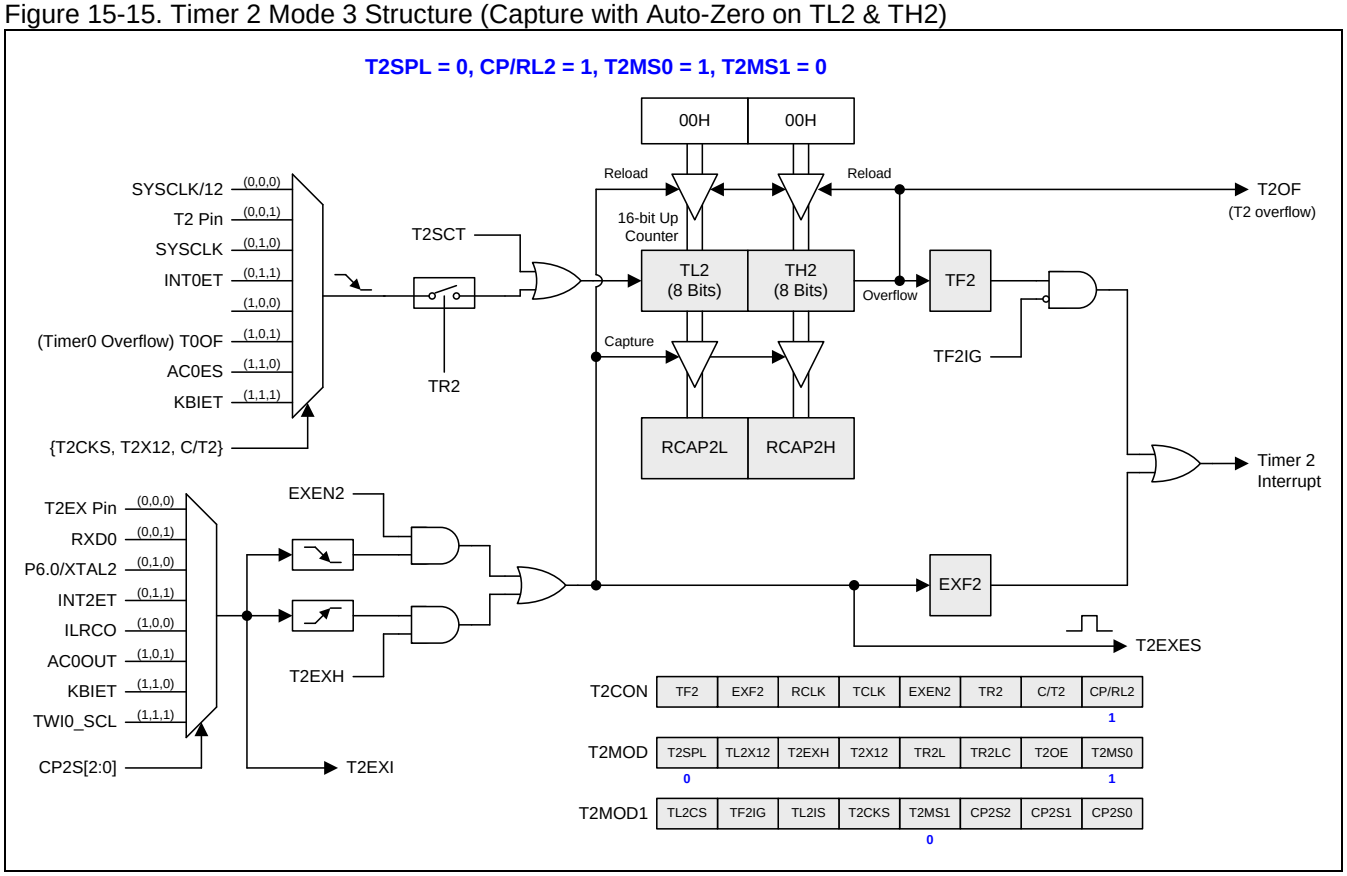


15.2.4. Timer 2 Mode 3 (Capture with Auto-Zero)

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 Time 2 but also clears the content of TL2 and TH2 to 0x0000H.

Timer 2 Mode 3 is illustrated in Figure 15–15.

Figure 15-15. Timer 2 Mode 3 Structure (Capture with Auto-Zero on TL2 & TH2)



15.2.5. Split Timer 2 Mode 0 (AR and Ex. INT)

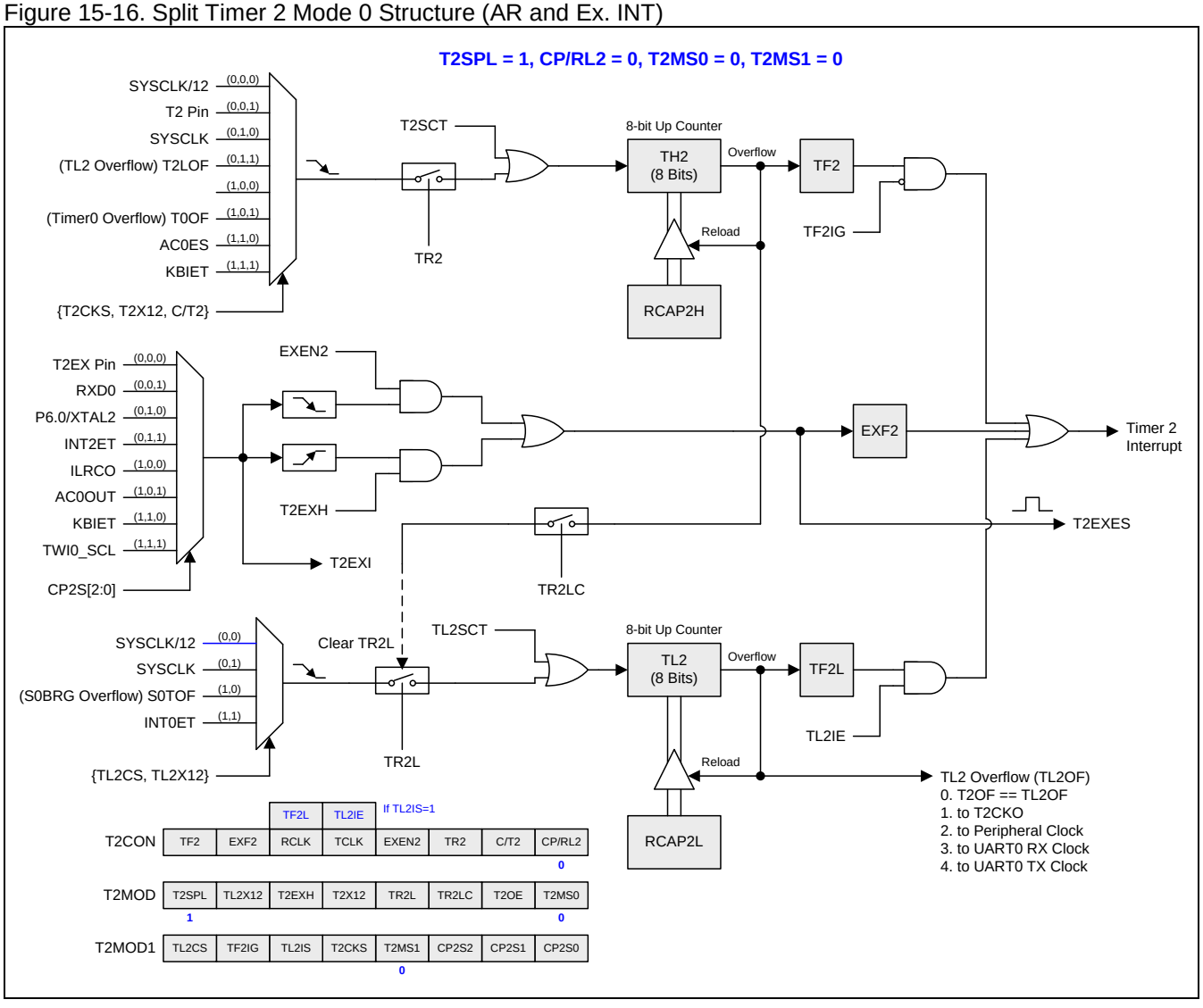
When T2SPLIT is set in this mode, Timer 2 operates as two 8-bit timers (TH2 and TL2). Both 8-bit timers operate in up counter as shown in Figure 15–16. TH2 holds the reload value for RCAP2H and keep the same 8 clock source inputs selection as 16-bit mode. It behaves the 8-bit function liked Timer 2 Mode 0 in 16-bit mode. TL2 holds the reload value for RCAP2L with 4 clock inputs selection. The TR2 bit in T2CON handles the run control for TH2. The TR2L bit in T2MOD handles the run control for TL2. And TH2 overflow can stop the TR2L running when TR2LC is set.

There are 3 interrupt flags in split mode, EXF2, TF2 and TF2L. EXF2 has the same function as 16-bit mode to detect the transition on T2EXI. TF2 is set when TH2 overflows from 0xFF to 0x00 with TF2IG control. TF2L is set when TL2 overflows from 0xFF to 0x00 with interrupt enabled by TL2IE. The EXF2, TF2 and TF2L interrupt flags are not cleared by hardware and must be cleared by software.

By the way, the Timer 2 overflow event (T2OF) in 16-bit timer is replaced by TL2 overflow event (TL2OF) in this split mode.

If TL2IS in T2MOD1 is 0, the bits on T2CON.5~4 are the function of RCLK and TCLK. If TL2IS is 1, the bits on T2CON.5~4 are the function of TF2L and TL2IE.

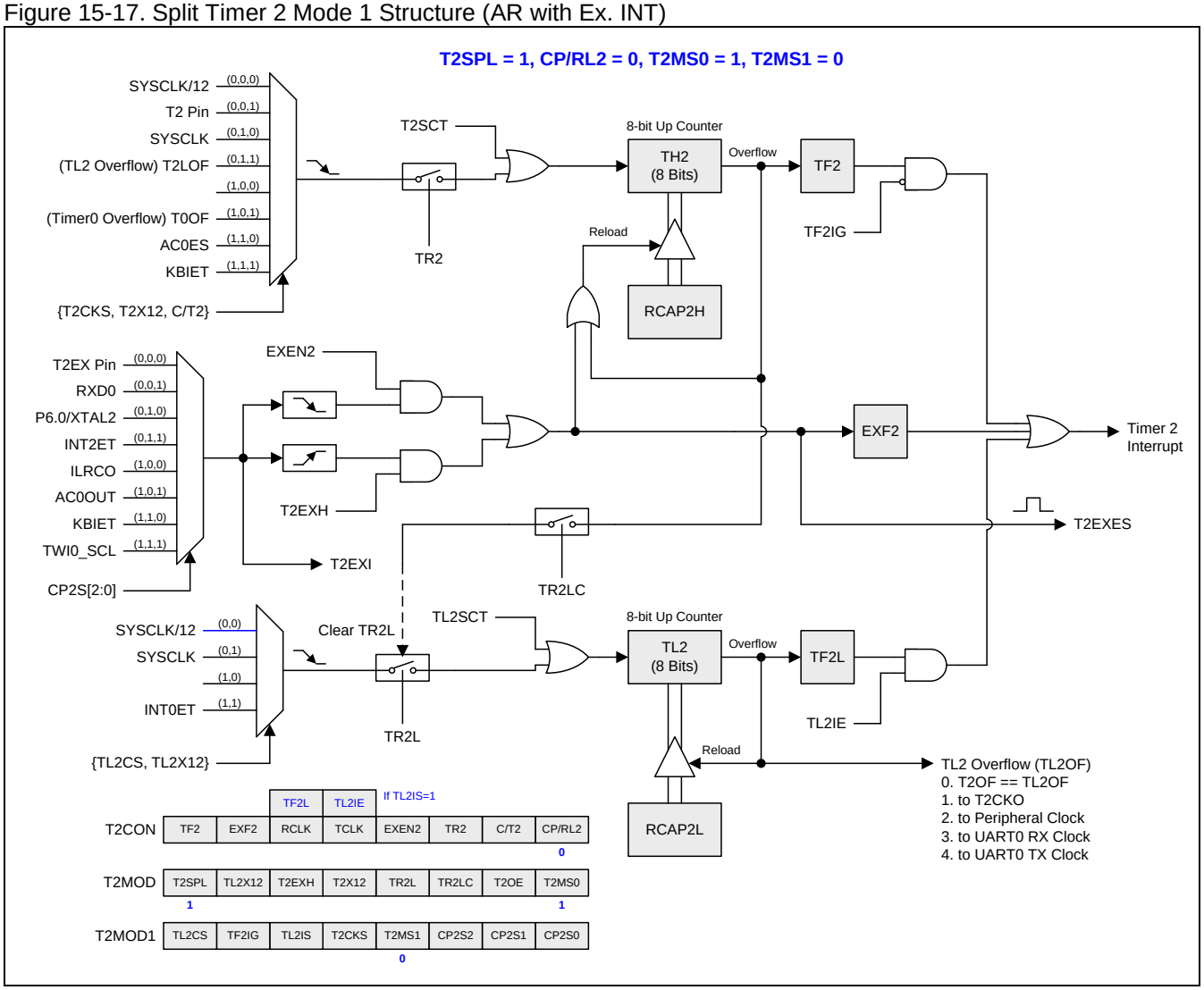
Figure 15-16. Split Timer 2 Mode 0 Structure (AR and Ex. INT)



15.2.6. Split Timer 2 Mode 1 (AR with Ex. INT)

When T2SPLIT is set in this mode, Time 2 is split to two 8-bit timers as shown in Figure 15–17. It is similar function as Timer 2 Mode 1 and keeps the same interrupt scheme in Split Timer 2 Mode 0.

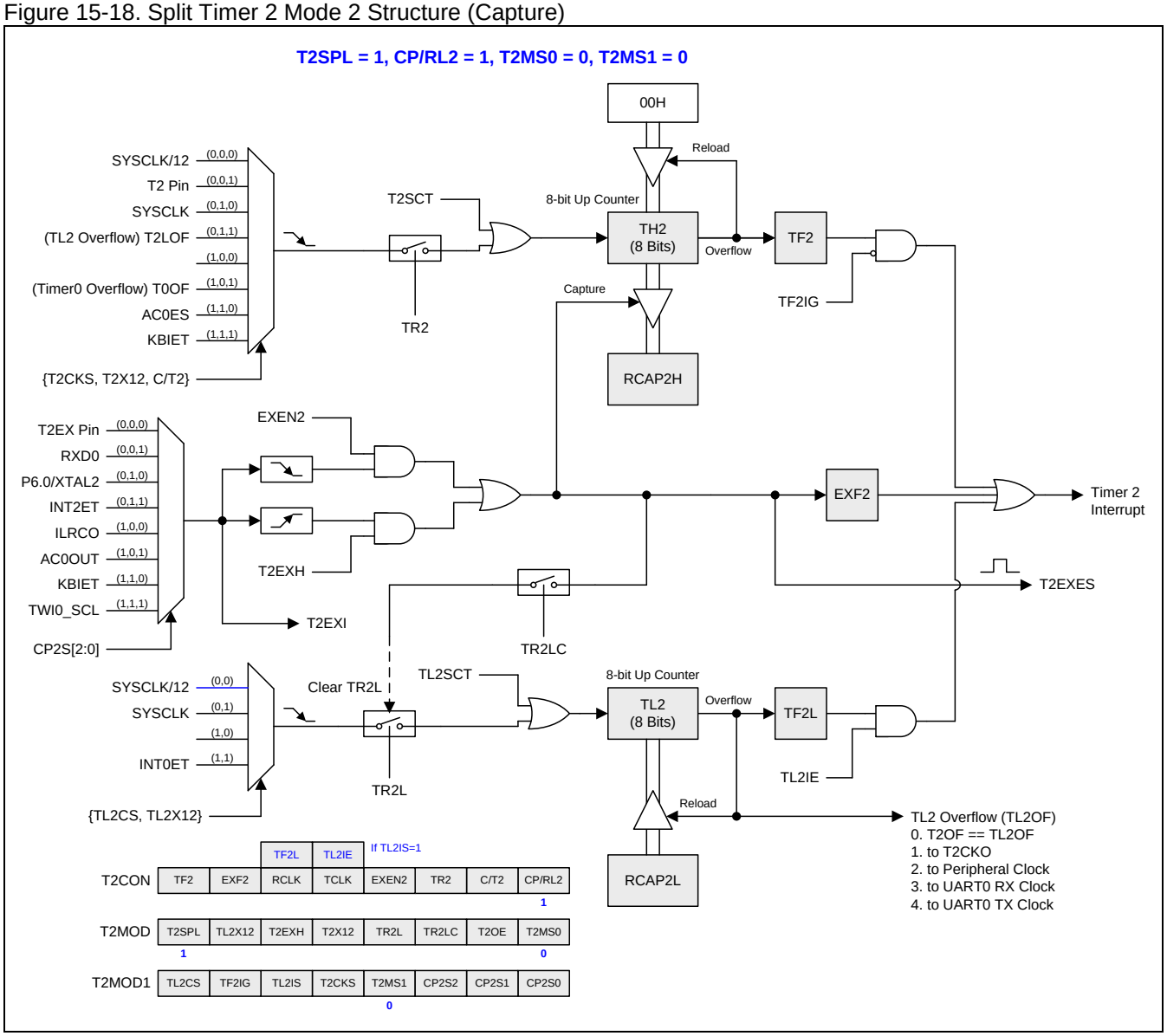
Figure 15-17. Split Timer 2 Mode 1 Structure (AR with Ex. INT)



15.2.7. Split Timer 2 Mode 2 (Capture)

When T2SPLIT is set in this mode, Time 2 is split to two 8-bit timers as shown in Figure 15–18. It is similar function as Timer 2 Mode 2 and keeps the same interrupt scheme in Split Timer 2 Mode 0.

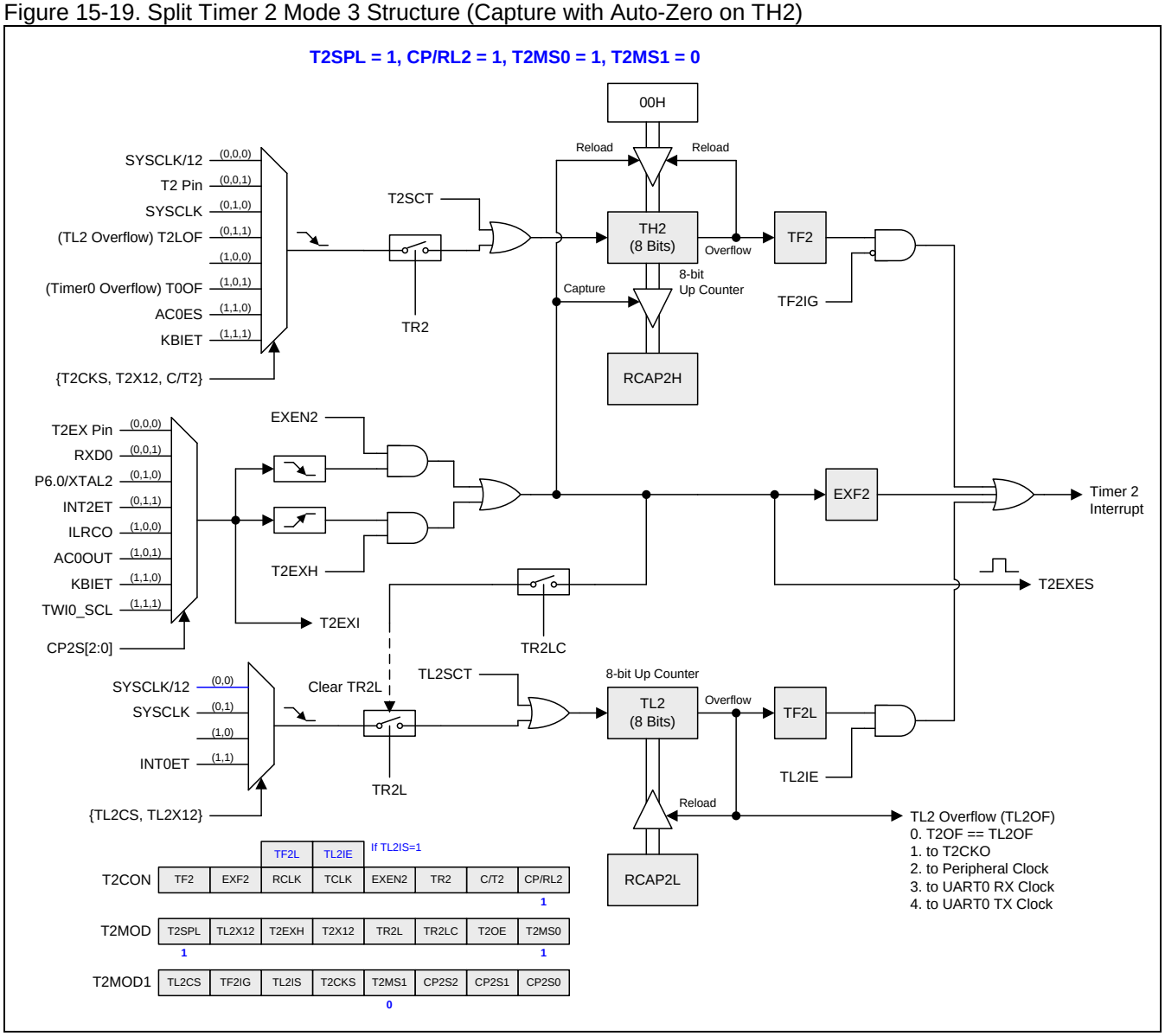
Figure 15-18. Split Timer 2 Mode 2 Structure (Capture)



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

When T2SPLIT is set in this mode, Time 2 is split to two 8-bit timers as shown in Figure 15–19. It is similar function as Timer 2 Mode 3 and keeps the same interrupt scheme in Split Timer 2 Mode 0.

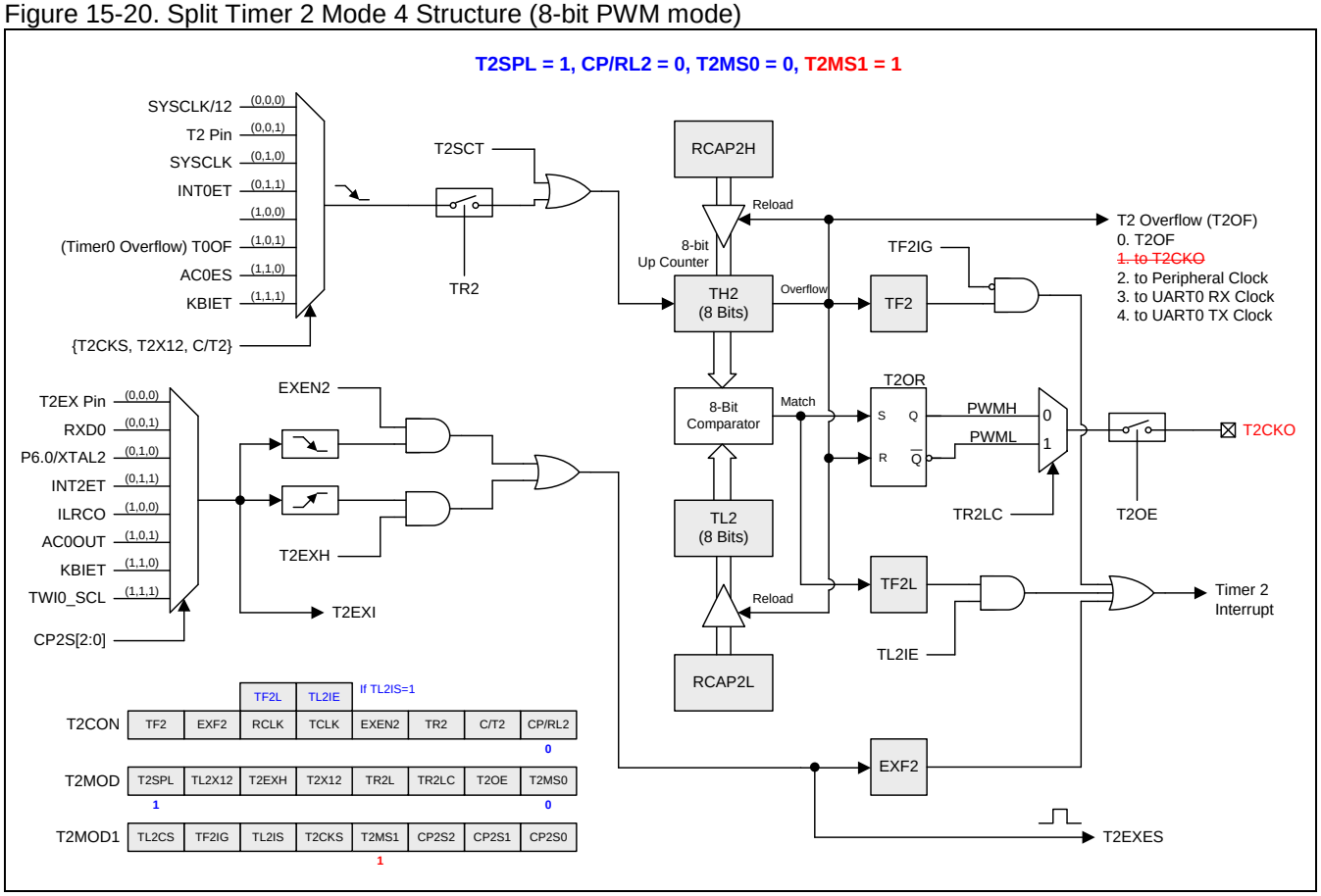
Figure 15-19. Split Timer 2 Mode 3 Structure (Capture with Auto-Zero on TH2)



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

In this mode, Timer 2 is an 8-bit PWM mode as shown in Figure 15–20. TH2 and RCAP2H are combined to an 8-bit auto-reload counter. Software configures these two registers to decide the PWM cycle time. TL2 is the PWM compare register to generate PWM waveform. RCAP2L is the PWM buffer register and software will update PWM data in this register. Each TH2 overflow event will set TF2 and load RCAP2L value into TL2. The PWM signal will be output on T2CKO function pin and the output is gated by T2OE in T2MOD register.

Figure 15-20. Split Timer 2 Mode 4 Structure (8-bit PWM mode)



15.2.10.Baud-Rate Generator Mode (BRG)

If Timer 2 in Mode 0, 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 15–21 shows the Timer 2 in baud rate generation mode to generate RX Clock and TX Clock into UART engine (See Figure 17–6.). 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, but will not generate an interrupt. Thus, the Timer 2 interrupt does not have to 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 at T2EXI (one of 8 Timer 2 trigger inputs) will set EXF2 (T2 external flag) but will not cause a reload from (RCAP2H, RCAP2L) to (TH2,TL2) in Mode 0. Therefore when Timer 2 is in use as a baud rate generator, T2EXI can be used as an additional external interrupt, if needed. T2EXH performs the same function as EXEN2 but it enables the detecting a 0-to-1 transition at T2EXI input.

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 "17.7.4 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 15–22. 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 15-21. Timer 2 in Baud-Rate Generator Mode

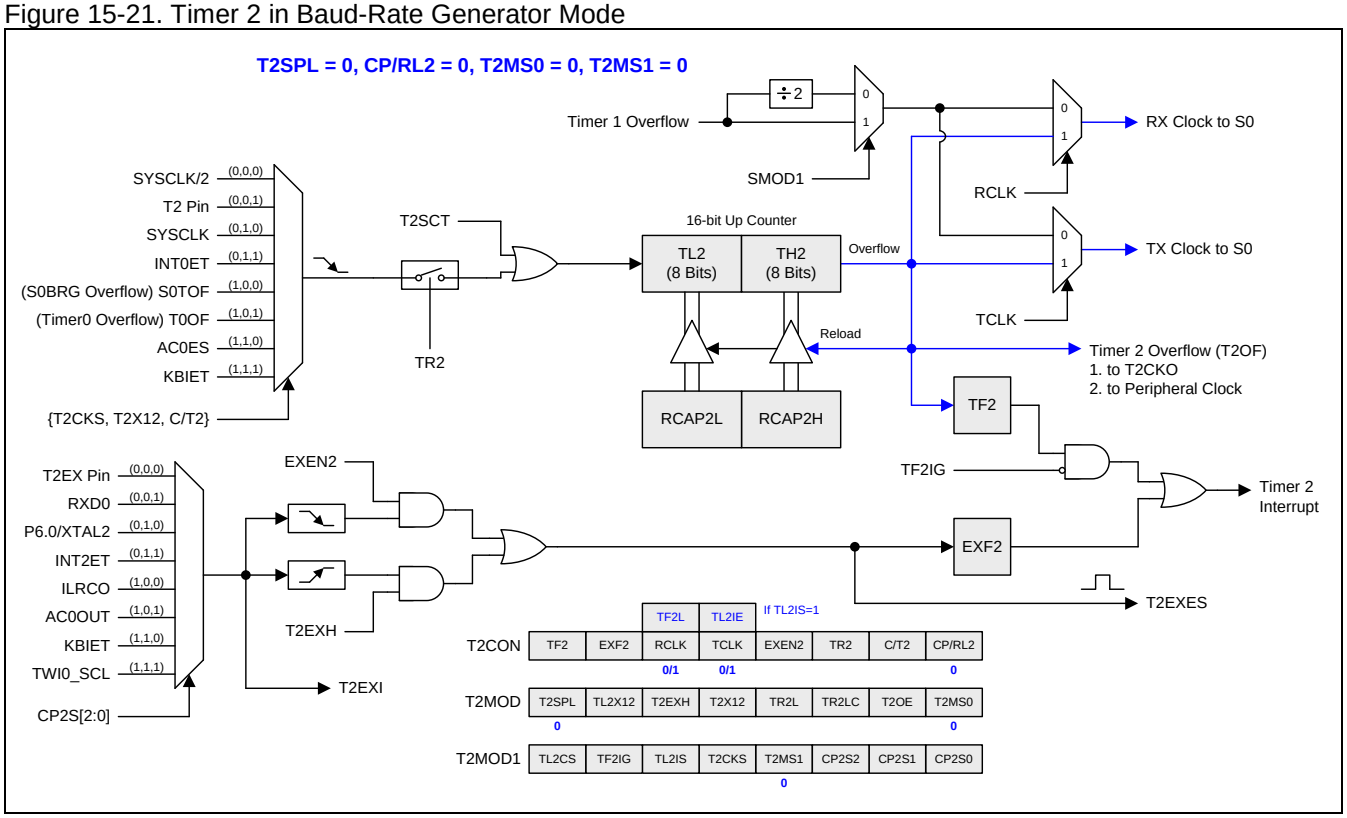
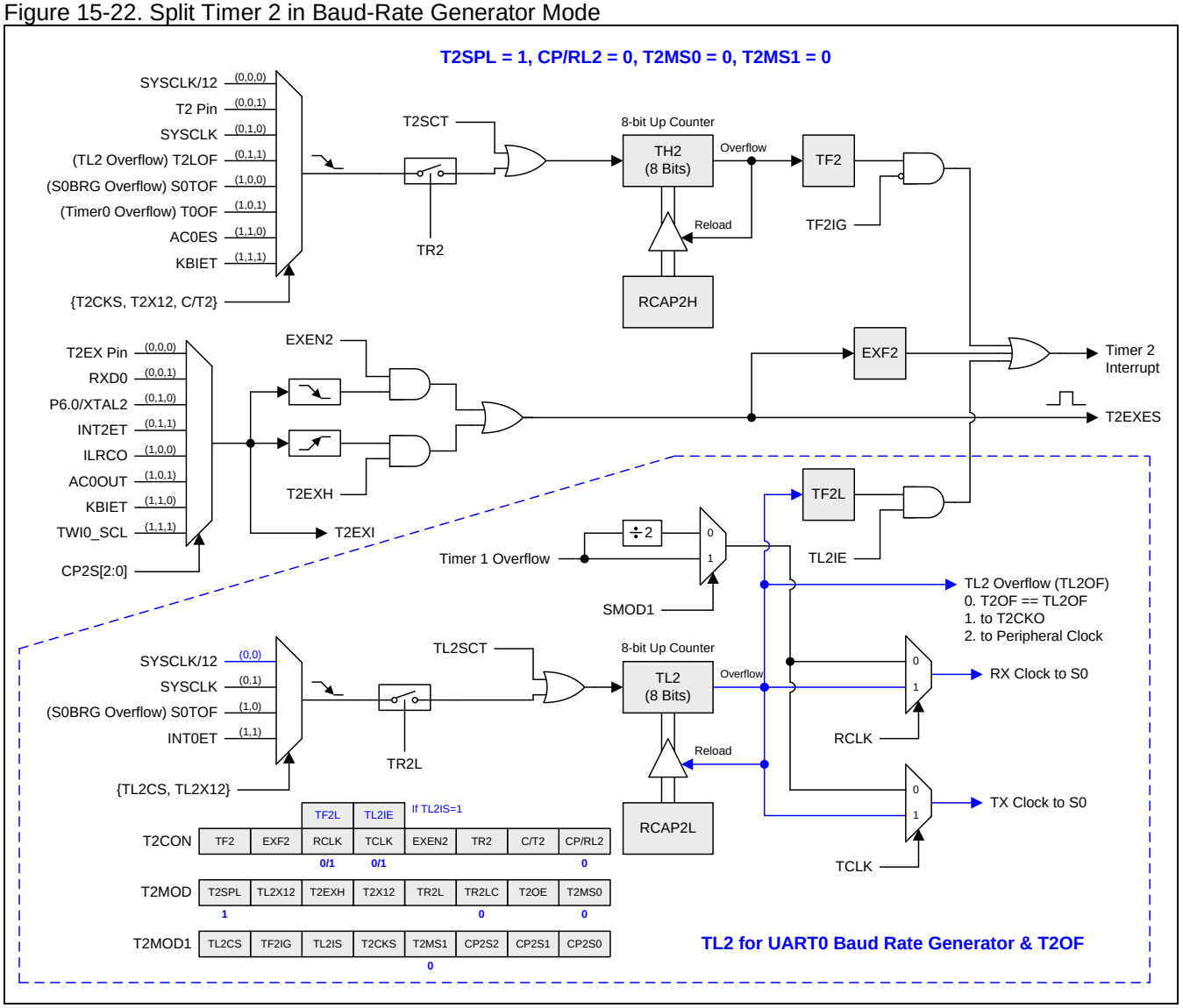


Figure 15-22. Split Timer 2 in Baud-Rate Generator Mode



15.2.11.Timer 2 Programmable Clock Output

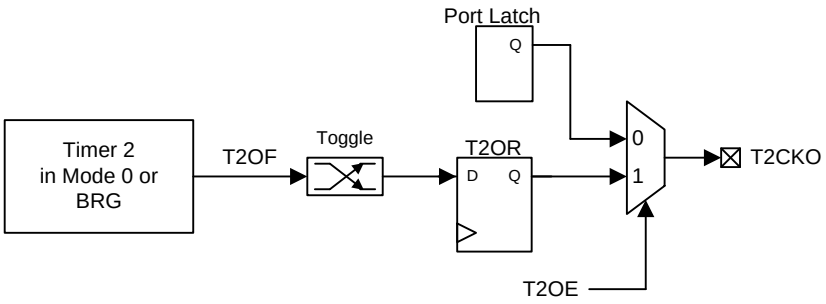
Timer 2 has a Clock-Out Mode (while Mode 0 & T2OE=1). In this mode, Timer 2 operates as a programmable clock generator with 50% duty-cycle. The generated clocks come out on P1.0. The input clock increments the 16-bit timer (TH2, TL2). The timer repeatedly counts to overflow from a loaded value. Once overflows occur, the contents of (RCAP2H, RCAP2L) are loaded into (TH2, TL2) for the consecutive counting. Figure 15–23 gives the formula of Timer 2 clock-out frequency: Figure 15–24 shows the clock structure of Timer 2.

Figure 15-23. 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 15-24. 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 generate an interrupt. This is different 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 will not generate interrupt.

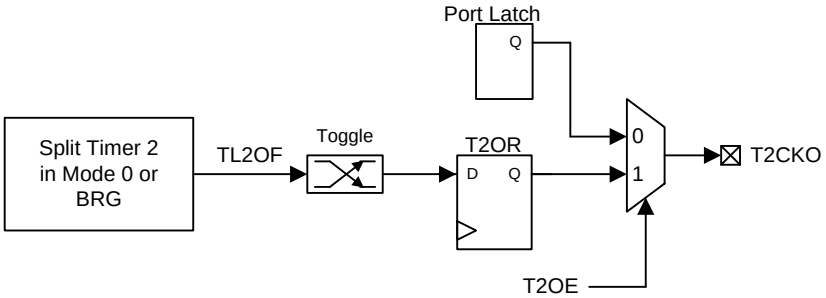
If Timer 2 in split mode, the clock output function is generated by TL2 overflow and the output clock frequency is TL2 overflow rate /2. RCAP2L is the TL2's reload value when TL2 overflow. There are four clock source selections for TL2. Before enable split Timer 2 clock output function, software must finish the TL2 clock source configuration. Figure 15–25 gives the formula of TL2 clock-out frequency: Figure 15–26 shows the clock structure of Split Timer 2.

Figure 15-25. 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 15-26. Split Timer 2 in Clock-Out Mode



How to Program Split Timer 2 in Clock-out Mode

- Select TL2 clock source.
- Determine the 8-bit reload value from the formula and enter it in the RCAP2L register.
- Enter the same reload value as the initial value in the TL2 register.
- Set T2OE bit in T2MOD register.
- Set TR2L bit in T2MOD register to start the Timer 2.

In the Clock-Out mode, TL2 rollovers will 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.

15.2.12.Timer 2 Register

T2CON: Timer 2 Control Register
SFR Page = **0 ~ F**
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 is set when 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. When selecting the asynchronous clock (external clock from I/O pin or ILRCO) as the timer clock source, it's frequency needs to lower then 1/2 of the system clock to ensure the edge event can be latched to trigger the counter. 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	--	--
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 ~ F**

SFR Address = 0xC9 RESET= 0000-0000

7	6	5	4	3	2	1	0
T2SPL	TL2X12	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.

Bit 6: TL2X12, the clock control bit of TL2 in Timer 2 split mode. When selecting the asynchronous clock (external clock from I/O pin or ILRCO) as the timer clock source, it's frequency needs to lower then 1/2 of the system clock to ensure the edge event can be latched to trigger the counter.

TL2CS, TL2X12	TL2 Clock Selection
0 0	SYSCLK/12
0 1	SYSCLK
1 0	S0TOF
1 1	INT0ET

Bit 5: T2EXH, Timer 2 external enable flag on a positive transition of T2EX pin.
0: Cause Timer 2 to ignore positive transition events at T2EX pin.
1: Allows a capture or reload to occur as a result of a 0-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.

T2SPL, T2MS1, CP/RL2, T2MS0	Timer 2 Mode Selection
0 0 0 0	Mode 0: Auto-Reload and External Interrupt
0 0 0 1	Mode 1: Auto-Reload with External Interrupt
0 0 1 0	Mode 2: Capture mode
0 0 1 1	Mode 3: Capture with Auto-Zero on Timer 2
1 0 0 0	Split Mode 0
1 0 0 1	Split Mode 1
1 0 1 0	Split Mode 2
1 0 1 1	Split Mode 3
1 1 0 0	8-bit PWM Mode
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 iGnore.
0: Enabled TF2 interrupt. Default is enabled.
1: Disable TF2 interrupt.

Bit 5: TL2IS, TF2L and TL2IE access control.
0: Enable RCLK and TCLK access function on T2CON.5~4.
1: Enable TF2L and TL2IE access function on T2CON.5~4.

Bit 4: T2CKS, Timer 2 clock selection. Refer to C/T2 description for the function defined.

Bit 3: T2MS1, Timer 2 mode selection bit 1. Refer T2MOD.T2MS0 description for the function definition.

Bit 2~0: CP2S.2~0. These bits define the capture source selector of Timer 2.

CP2S.2~0	Timer 2 Capture Source Selection
0 0 0	T2EX Pin
0 0 1	RXD0
0 1 0	P6.0/XTAL2
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 ~ F**

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 ~ F**

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 ~ F**

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 ~ F**

SFR Address = 0xCB POR+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 POR+RESET = 0000-0x00

7	6	5	4	3	2	1	0
T2PS1	T2PS0	T1PS1	T1PS0	SPIPS0	--	AC0OE	AC0FLT1
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**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	P3.0	P3.1
10	P4.0	P4.1
11	P4.5	P4.4

15.3. Timer Global Control

When the applications are asking all timers work together in sync mode, just need to set the TRxE or TR2LE 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 Enalbe Register 0

SFR Page = **1 Only**

SFR Address = 0x95 RESET= 0000-0000

7	6	5	4	3	2	1	0
--	--	TR2LE	--	--	TR2E	TR1E	TR0E
W	**W**	**W**	**W**	**W**	**W**	**W**	**W**

Bit 7~6, Reserved, Software must write “0” on this bit when TREN0 is written.

Bit 5, TR2LE, write “1” on this bit to control TL2 by setting TR2L enabled (TR2L=1) when Timer 2 in split mode. This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

Bit 4~3, Reserved, Software must write “0” on this bit when TREN0 is written.

Bit 2, TR2E, write “1” on this bit to set TR2 enabled (TR2=1). This bit control whole timer when Timer 2 in Full mode (16Bit timer). But when Timer 2 in split mode, this bit only control TH2. This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

Bit 1, TR1E, write “1” on this bit to set TR1 enabled (TR1=1). This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

Bit 0, TR0E, write “1” on this bit to set TR0 enabled (TR0=1). This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

TRLC0: Timer Reload Control Register 0

SFR Page = **2 Only**

SFR Address = 0x95 RESET= 0000-0000

7	6	5	4	3	2	1	0
--	--	TL2RLC	--	--	T2RLC	T1RLC	T0RLC
W	**W**	**W**	**W**	**W**	**W**	**W**	**W**

Bit 7~6, Reserved, Software must write “0” on this bit when TRLC0 is written.

Bit 5, TL2RLC, write “1” on this bit to force TL2 reload condition active when Timer 2 in split mode. This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

Bit 4~3, Reserved, Software must write “0” on this bit when TRLC0 is written.

Bit 2, T2RLC, write “1” on this bit to force TH2 and TL2 reload when Timer 2 not in split mode. Or force TH2 reload when Timer 2 in split mode. This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

Bit 1, T1RLC, write “1” on this bit to force TL1 reload in Timer 1 mode 2. This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

Bit 0, T0RLC, write “1” on this bit to force TL0 reload in Timer 0 mode 2. This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

TSPC0: Timer Stop Control Register 0

SFR Page = **3 Only**

SFR Address = 0x95 RESET= 0000-0000

7	6	5	4	3	2	1	0
--	--	TL2SC	--	--	T2SC	T1SC	T0SC
W	**W**	**W**	**W**	**W**	**W**	**W**	**W**

Bit 7~6, Reserved, Software must write “0” on this bit when TSPC0 is written.

Bit 5, TL2SC, write “1” on this bit to set TR2L disabled (TR2L=0) when Timer 2 in split mode. This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

Bit 4~3, Reserved, Software must write “0” on this bit when TSPC0 is written.

Bit 2, T2SC, write “1” on this bit to set TR2 disabled (TR2=0). This bit control whole timer when Timer 2 in Full mode (16-Bit timer). But when Timer 2 in split mode, this bit only control TH2. This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

Bit 1, T1SC, write “1” on this bit to set TR1 enabled (TR1=0). This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

Bit 0, T0SC, write “1” on this bit to set TR0 enabled (TR0=0). This bit is auto-cleared by hardware after writing “1” operation. Write “0” on this bit is no action.

16. Programmable Counter Array (PCA0)

The **MG82FG5D16** is equipped with a Programmable Counter Array (PCA0), which provides more timing capabilities with less CPU intervention than the standard timer/counters. Its advantages include reduced software overhead and improved accuracy.

16.1. PCA0 Overview

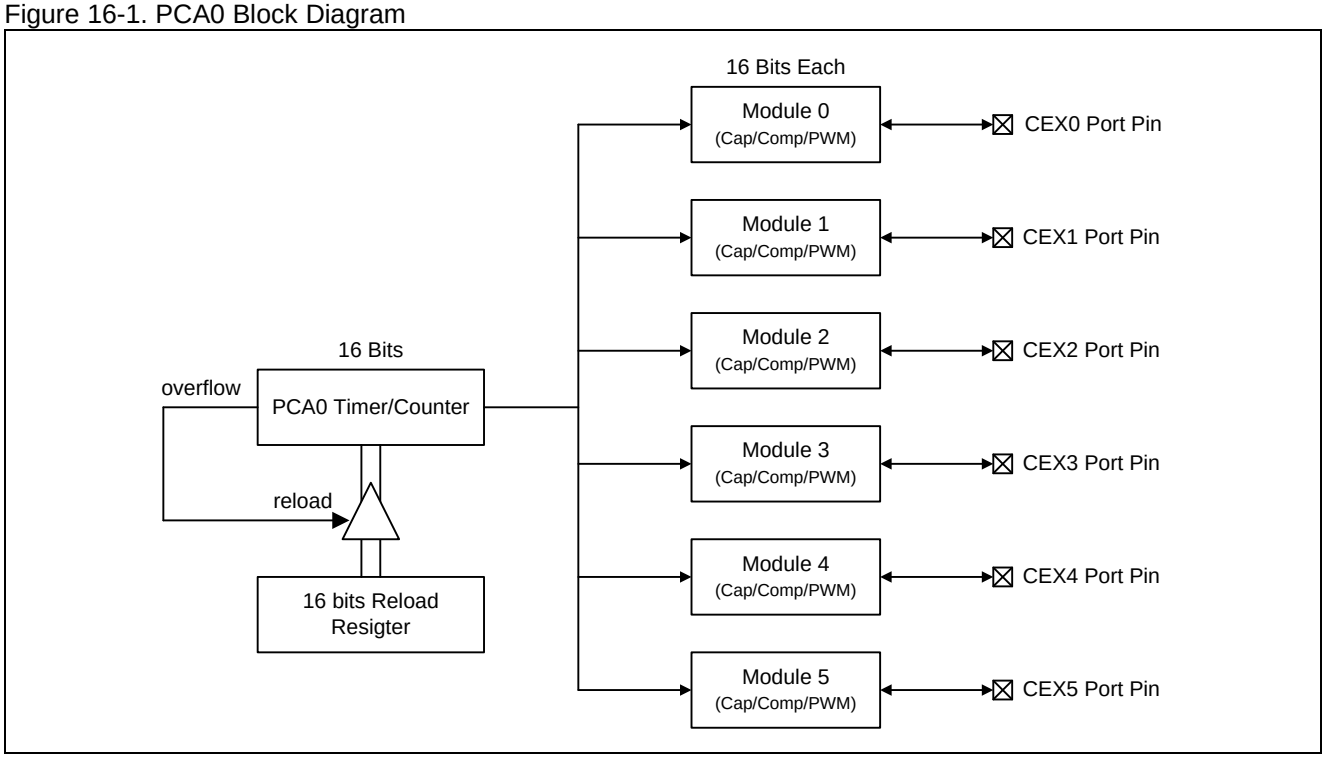
The PCA0 consists of a dedicated timer/counter which serves as the time base for an array of **Six** compare/capture modules. Figure 16–1 shows a block diagram of the PCA0. Notice that the PCA0 timer and modules are all 16-bits. If an external event is associated with a module, that function is shared with the corresponding Port pin. If the module is not using the port pin, the pin can still be used for standard I/O.

Module 0~5 can be programmed in any one of the following modes:

- Rising and/or Falling Edge Capture
- Software Timer (Compare)
- High Speed Output (Compare Output)
- Pulse Width Modulator Output (PWM)
- Compare Output on PWM Match case (COPM)

All of these modes will be discussed later in detail. However, let's first look at how to set up the PCA0 timer and modules.

Figure 16-1. PCA0 Block Diagram



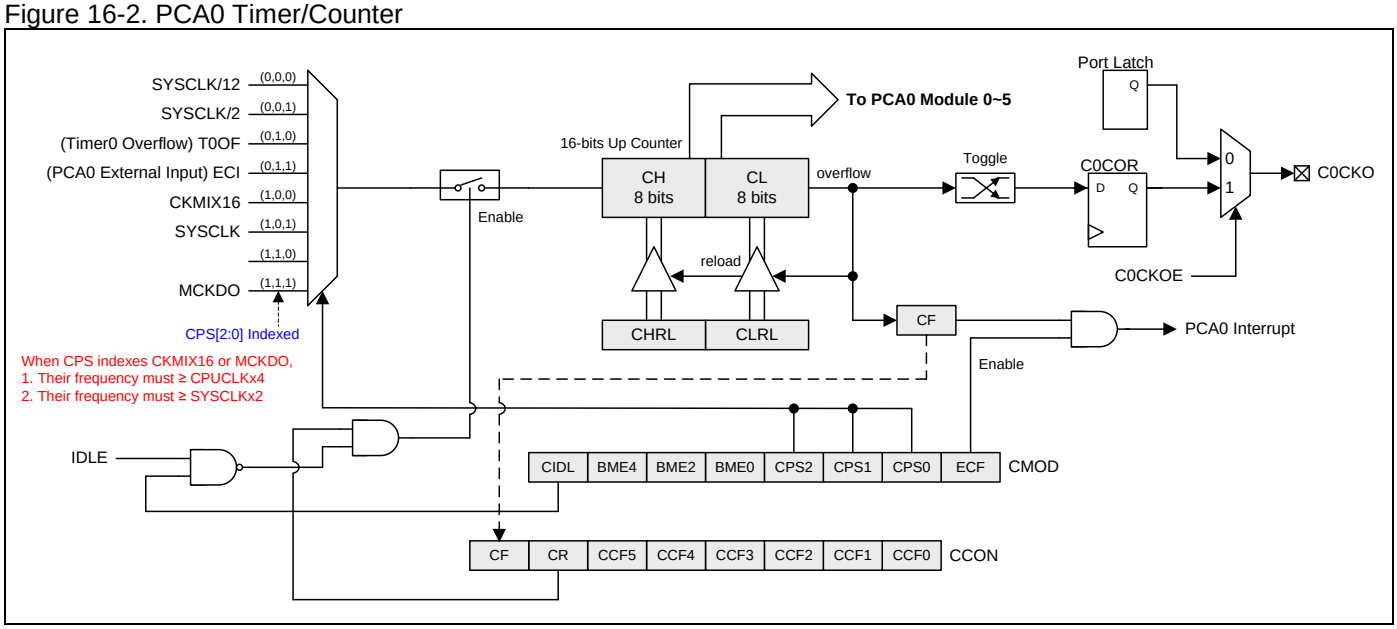
16.2. PCA0 Timer/Counter

The timer/counter for the PCA0 is an auto-reload 16-bit timer consisting of registers CH and CL (the high and low bytes of the count values), CHRL, CLRL (the high and low bytes reload registers), as shown in Figure 16–2. CHRL and CLRL are reloaded to CH and CL at each time overflow on {CH+CL} counter which can change the PCA cycle time for variable PWM resolution, such as 7-bit or 9-bit PWM.

{CH + CL} is the common time base for all modules and its clock input can be selected from the following source:
- 1/12 the system clock frequency,
- 1/2 the system clock frequency,
- the Timer 0 overflow, which allows for a range of slower clock inputs to the timer,
- external clock input, 1-to-0 transitions, on ECI pin,
- CKMIX16, refer Section “8.1 Clock Structure”,
- directly from the system clock frequency,
- MCKDO, refer Section “8.1 Clock Structure”.

Special Function Register CMOD contains the Count Pulse Select bits (CPS2, CPS1 and CPS0) to specify the PCA0 timer input. 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 PCA0 timer during Idle Mode by setting the Counter Idle bit (CIDL). This can further reduce power consumption during Idle mode.

Figure 16-2. PCA0 Timer/Counter



CMOD: PCA0 Counter Mode Register

SFR Page = 0 ~ **F**

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, PCA0 counter Idle control.
0: To keep the PCA0 counter continue function available during Idle mode.
1: The PCA0 counter be gated off during Idle mode.

Bit 6: BME4, Buffer Mode Enable on PCA0 module 4/5. It is only valid on both of PCA0 module 4 and module 5 in capture mode, PWM mode or COPM mode.
0: PCA0 Module 4/5 buffer mode disabled.
1: PCA0 Module 4/5 buffer mode enabled.

Bit 5: BME2, Buffer Mode Enable on PCA0 module 2/3. It is only valid on both of PCA0 module 2 and module 3 in capture mode, PWM mode or COPM mode.
0: PCA0 Module 2/3 buffer mode disabled.
1: PCA0 Module 2/3 buffer mode enabled.

Bit 4: BME0, Buffer Mode Enable on PCA0 module 0/1. It is only valid on both of PCA0 module 0 and module 1 in capture mode, PWM mode or COPM mode.
0: PCA0 Module 0/1 buffer mode disabled.
1: PCA0 Module 0/1 buffer mode enabled.

Bit 3~1: CPS2~0, PCA0 counter clock source select bits.

CPS2	CPS1	CPS0	PCA0 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	Reserved.
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 x4.*
2. *The source frequency must ≥SYSCLK x2.*

Bit 0: ECF, Enable PCA0 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 PCA0 and the flags for the PCA0 timer and each module. To run the PCA0 counter, the CR bit (CCON.6) must be set by software. The PCA0 is shut off by clearing this bit. The CF bit (CCON.7) is set when the PCA0 counter overflows and an interrupt will be generated if the ECF bit in the CMOD register is set. The CF bit can only be cleared by software. CCF0 to CCF5 are the interrupt flags for module 0 to module 5, respectively, and they are set by hardware when either a match or a capture occurs. These flags can only be cleared by software. The PCA0 interrupt system is shown Figure 16–3.

CCON: PCA0 Counter Control Register

SFR Page = 0 **~ F**
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, PCA0 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, PCA0 Counter Run control bit.
0: Must be cleared by software to turn the PCA0 counter off.
1: Set by software to turn the PCA0 counter on.

Bit 5: CCF5, PCA0 Module 5 interrupt flag.
0: Must be cleared by software.
1: Set by hardware when a match or capture occurs.

Bit 4: CCF4, PCA0 Module 4 interrupt flag.
0: Must be cleared by software.
1: Set by hardware when a match or capture occurs.

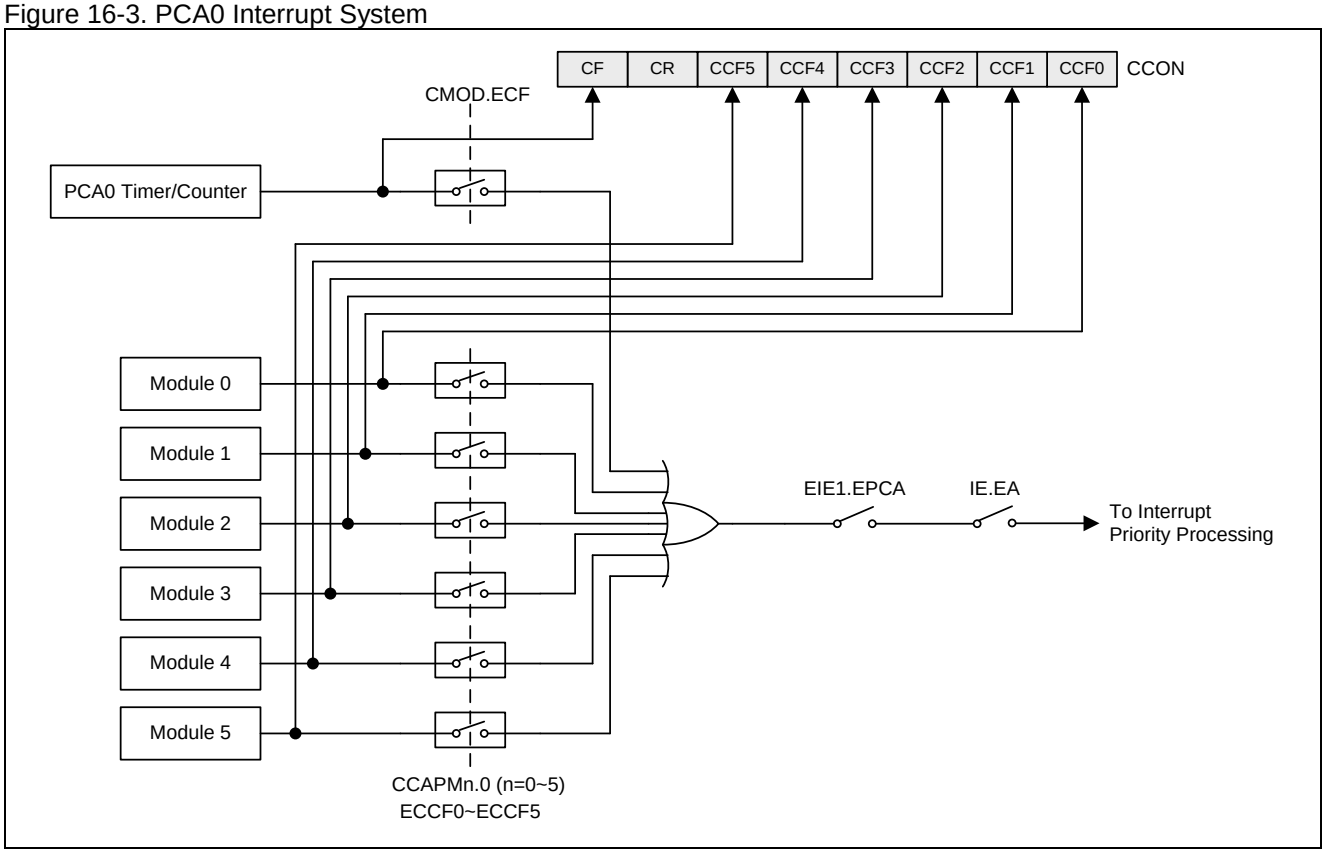
Bit 3: CCF3, PCA0 Module 3 interrupt flag.
0: Must be cleared by software.
1: Set by hardware when a match or capture occurs.

Bit 2: CCF2, PCA0 Module 2 interrupt flag.
0: Must be cleared by software.
1: Set by hardware when a match or capture occurs.

Bit 1: CCF1, PCA0 Module 1 interrupt flag.
0: Must be cleared by software.
1: Set by hardware when a match or capture occurs.

Bit 0: CCF0, PCA0 Module 0 interrupt flag.
0: Must be cleared by software.
1: Set by hardware when a match or capture occurs.

Figure 16-3. PCA0 Interrupt System



CH: PCA0 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: PCA0 base timer Low

SFR Page = 0 **~ F**
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: PCA0 CH Reload Register

SFR Page = 0 **~ F**

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: PCA0 CL Reload Register

SFR Page = 0 **~ F**

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.

16.3. Compare/Capture Modules

Each of the compare/capture 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: PCA0 Module Compare/Capture Register, n=0~5

SFR Page = 0 ~ F

SFR Address = 0xDA~0xDF RESET = x000-0000

7	6	5	4	3	2	1	0
--	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn
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 CCAPMn is written.

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

Bit 5: CAPPn, Capture Positive enabled.
0: Disable the PCA0 capture function on CEXn positive edge detected.
1: Enable the PCA0 capture function on CEXn positive edge detected.

Bit 4: CAPNn, Capture Negative enabled.
0: Disable the PCA0 capture function on CEXn negative edge detected.
1: Enable the PCA0 capture function on CEXn negative edge detected.

Bit 3: MATn, Match control.
0: Disable the digital comparator match event to set CCFn.
1: A match of the PCA0 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 PCA0 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 PCA0 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.

Note: The bits CAPNn (CCAPMn.4) and CAPPn (CCAPMn.5) determine the edge on which a capture input will be active. If both bits are set, both edges will be enabled and a capture will occur for either transition.

Each module also has a pair of 8-bit compare/capture registers (CCAPnH, CCAPnL) associated with it. These registers are used to store the time when a capture event occurred or when a compare event should occur. When a module is used in the PWM mode, in addition to the above two registers, an extended register PCAPWMn is used to improve the range of the duty cycle of the output. The improved range of the duty cycle starts from 0%, up to 100%, with a step of 1/256. About 10/12/16 bit PWM setting, please reference 16.4.6 & 16.4.7.

CCAPnH: PCA0 Module n Capture High Register, n=0~5

SFR Page = **0 ~ F**
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: PCA0 Module n Capture Low Register, n=0~5

SFR Page = **0 ~ F**
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 ~ F**
SFR Address = 0xF2~0xF7 RESET = 00xx-x000

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

16.4. Operation Modes of the PCA

Table 16–1 shows the CCAPMn register settings for the various PCA functions.

Table 16-1. PCA Module Modes

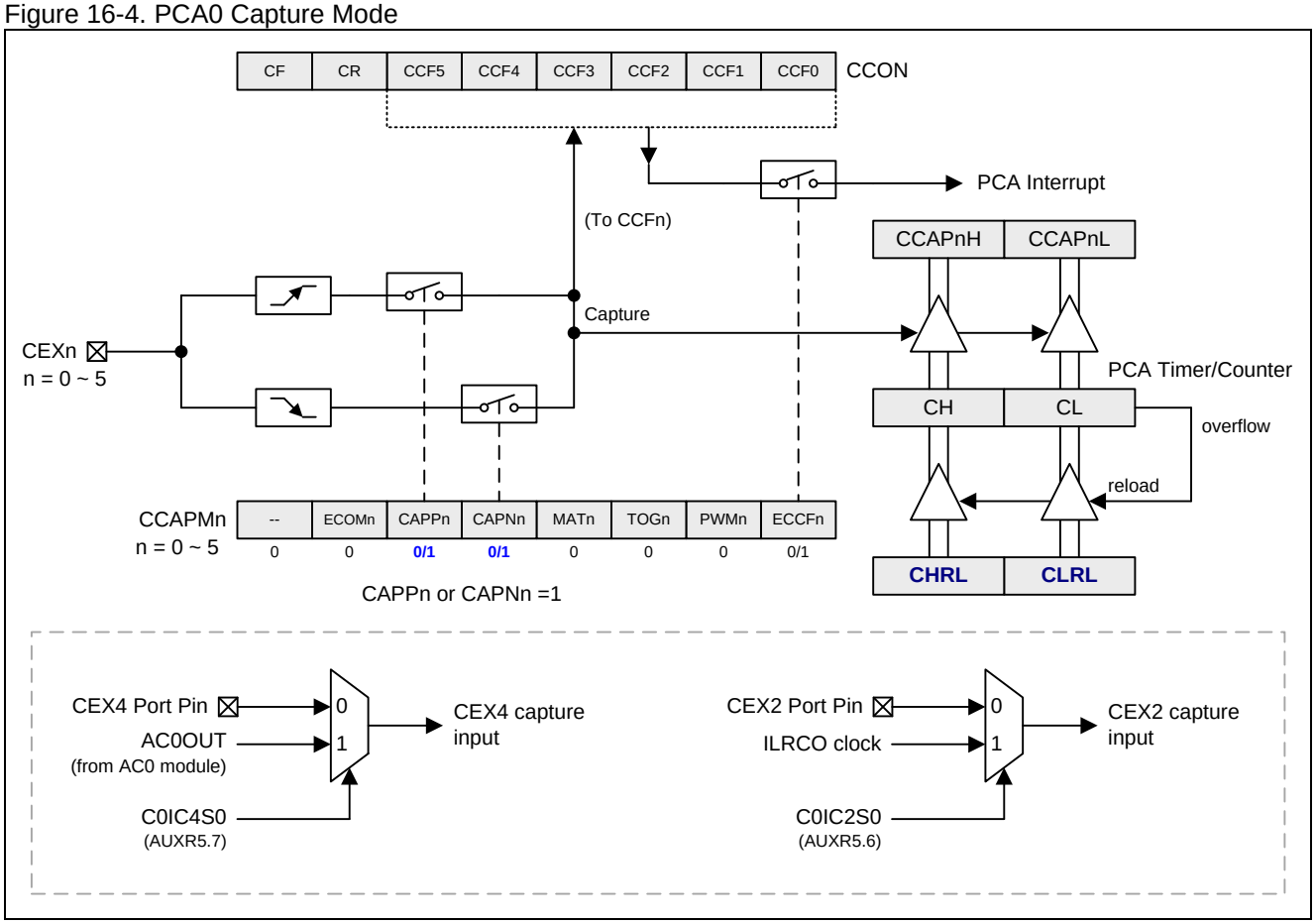
ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn	Module Function
0	0	0	0	0	0	0	No operation
X	1	0	0	0	0	X	16-bit capture by a positive-edge trigger on CEXn
X	0	1	0	0	0	X	16-bit capture by a negative-edge trigger on CEXn
X	1	1	0	0	0	X	16-bit capture by a transition on CEXn
1	0	0	1	0	0	X	16-bit Software Timer (Compare)
1	0	0	1	1	0	X	16-bit High Speed Output (HSO, Compare Output)
1	0	0	0	0	1	X	Pulse Width Modulator (PWM)
1	0	0	0	1	1	X	Compare Output on PWM Match case (COPM)

16.4.1. Capture Mode

To use one of the PCA0 modules in the capture mode, either one or both of the bits CAPN and CAPP for that module must be set. The external CEX input for the module is sampled for a transition. When a valid transition occurs the PCA0 hardware loads the value of the PCA0 counter registers (CH and CL) into the module's capture registers (CCAPnL and CCAPnH). If the CCFn and the ECCFn bits for the module are both set, an interrupt will be generated.

Capture input of channel 4 can be selected by software between CEX4 port pin and AC0OUT of AC0 output. Also, Channel 2 input has the alternated selection on ILRCO clock source.

Figure 16-4. PCA0 Capture Mode



16.4.2. Buffered Capture Mode

To capture narrow input signal, buffered capture mode is necessary. If enabled, it set the odd module capture data registers (CCAPnH, CCAPnL, n= 1, 3, 5) to be the buffer register of even module capture data registers (channel 0, 2, 4). There is no influence on module 0/2/4 capture operation. BME0 enables the buffer operation of channel 0 and channel 1. BME2 and BME4 control the module 2/3 and module 4/5.

Figure 16-5. PCA0 Buffered Capture Mode (BMEn=1, n= 0, 2, 4)

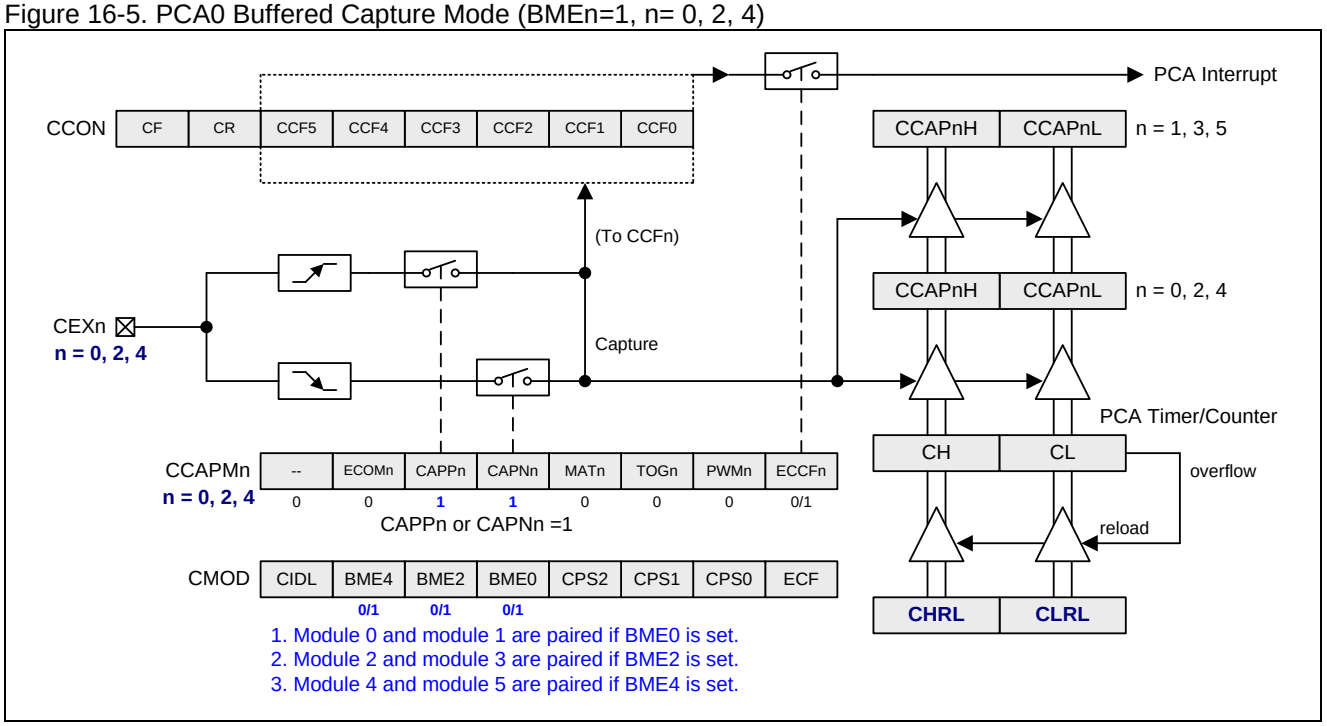
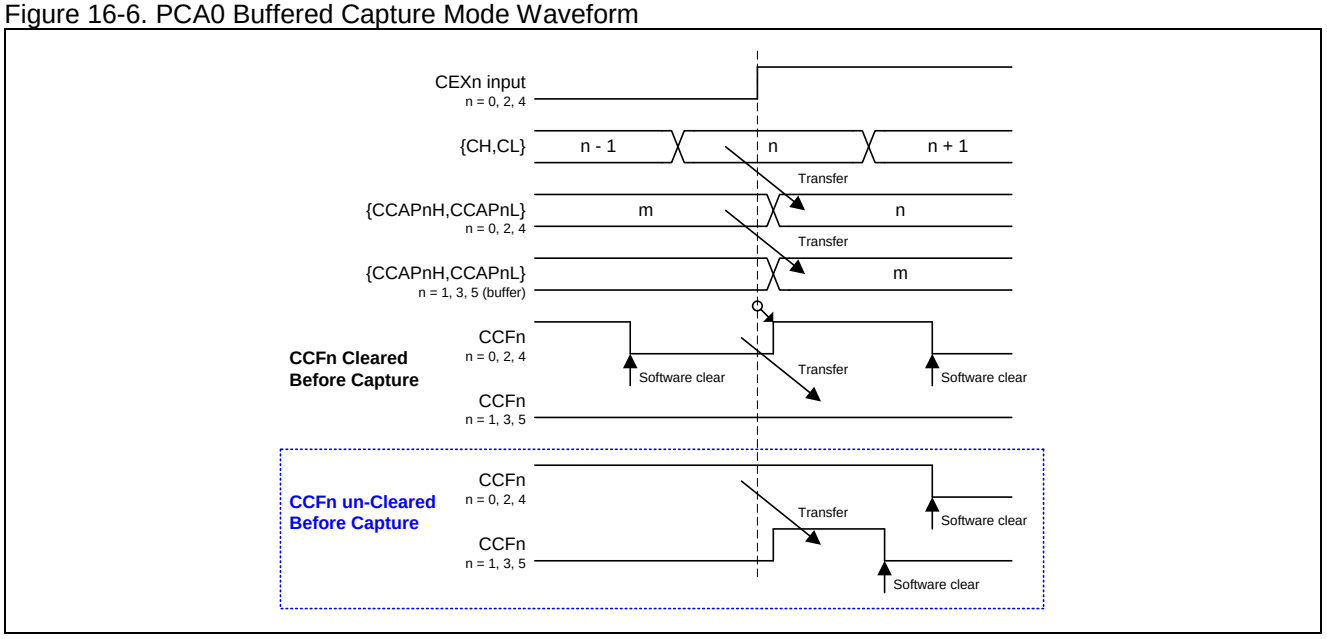


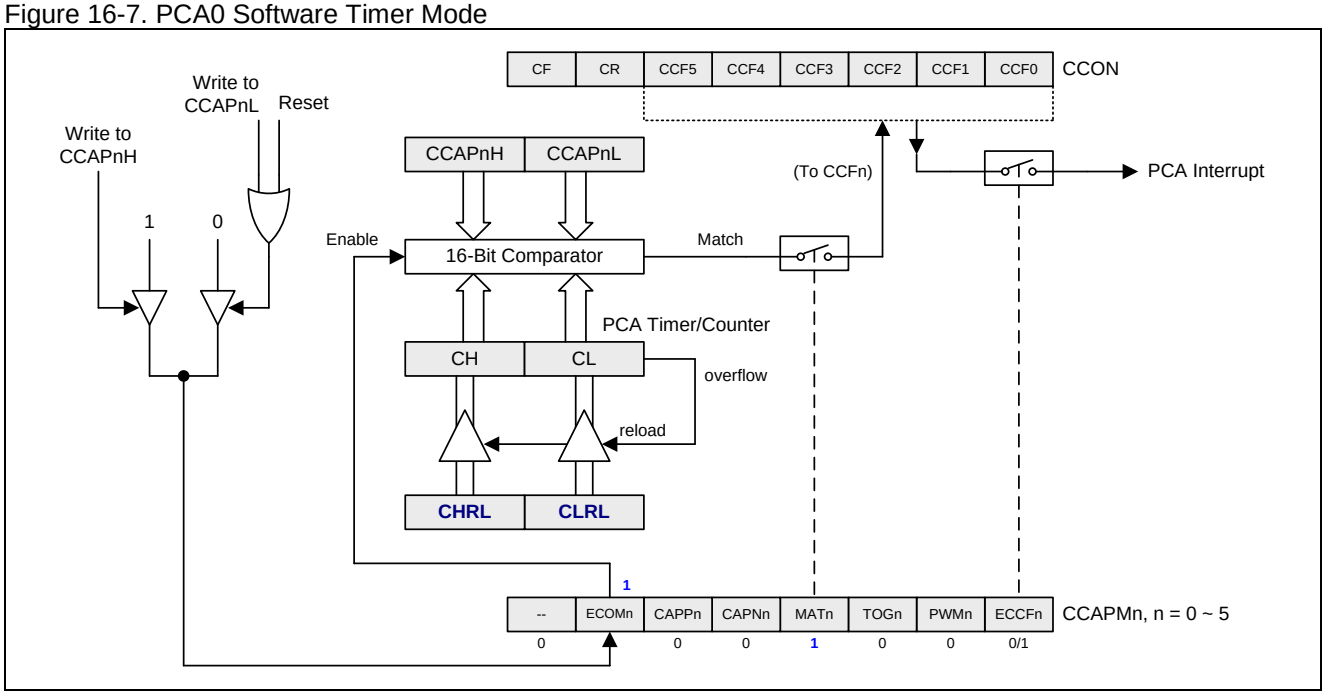
Figure 16-6. PCA0 Buffered Capture Mode Waveform



16.4.3. 16-bit Software Timer Mode (Compare mode)

The PCA0 modules can be used as software timers by setting both the ECOM and MAT bits in the module's CCAPMn register. The PCA0 timer will be compared to the module's capture registers, and when a match occurs an interrupt will be held if the CCFn and the ECCFn bits for the module are both set.

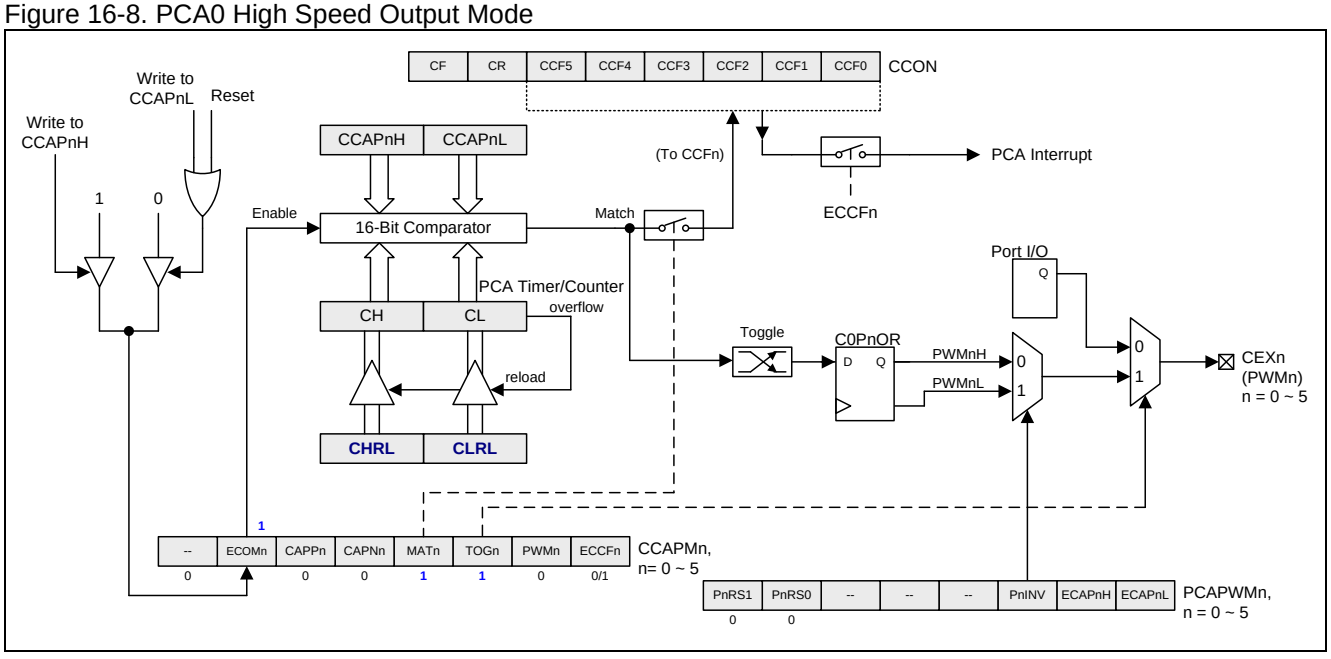
Figure 16-7. PCA0 Software Timer Mode



16.4.4. High Speed Output Mode (Compare Output mode)

In this mode the CEX output associated with the PCA0 module will toggle each time a match occurs between the PCA0 counter and the module's capture registers (CCAPnH & CCAPnL). To activate this mode, the TOG, MAT and ECOM bits in the module's CCAPMn register must be set.

Figure 16-8. PCA0 High Speed Output Mode



16.4.5. Buffered 8-bit PWM Mode

All of the PCA0 modules can be used as PWM outputs. The frequency of the output depends on the clock source for the PCA timer. All of the modules will have the same frequency of output because they all share the PCA0 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.

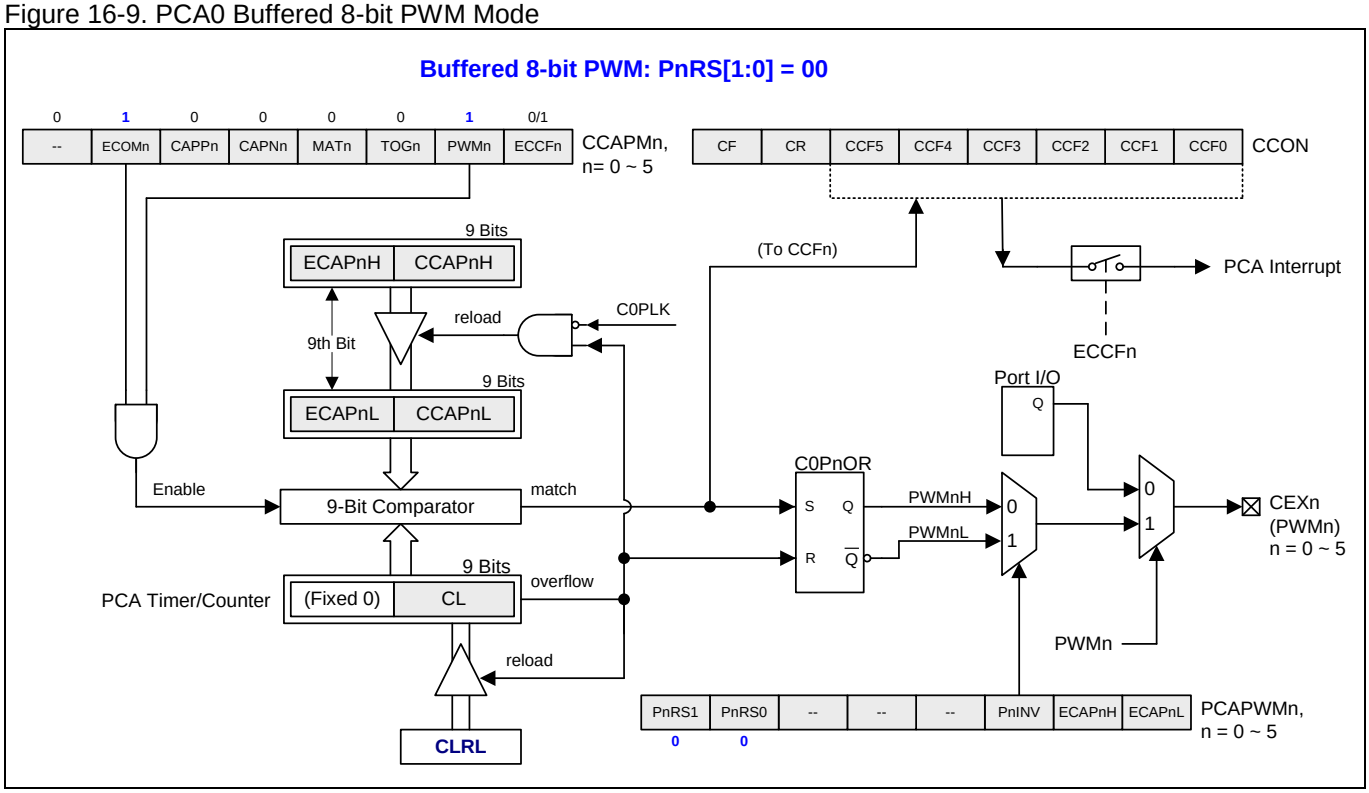
Using the 9-bit comparison, the duty cycle of the output can be improved to really start from 0%, and up to 100%. The formula for the duty cycle is:

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

Where, *[CCAPnH]* is the 8-bit value of the CCAPnH register, and *ECAPnH* (bit-1 in the PCAPWMn 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%.

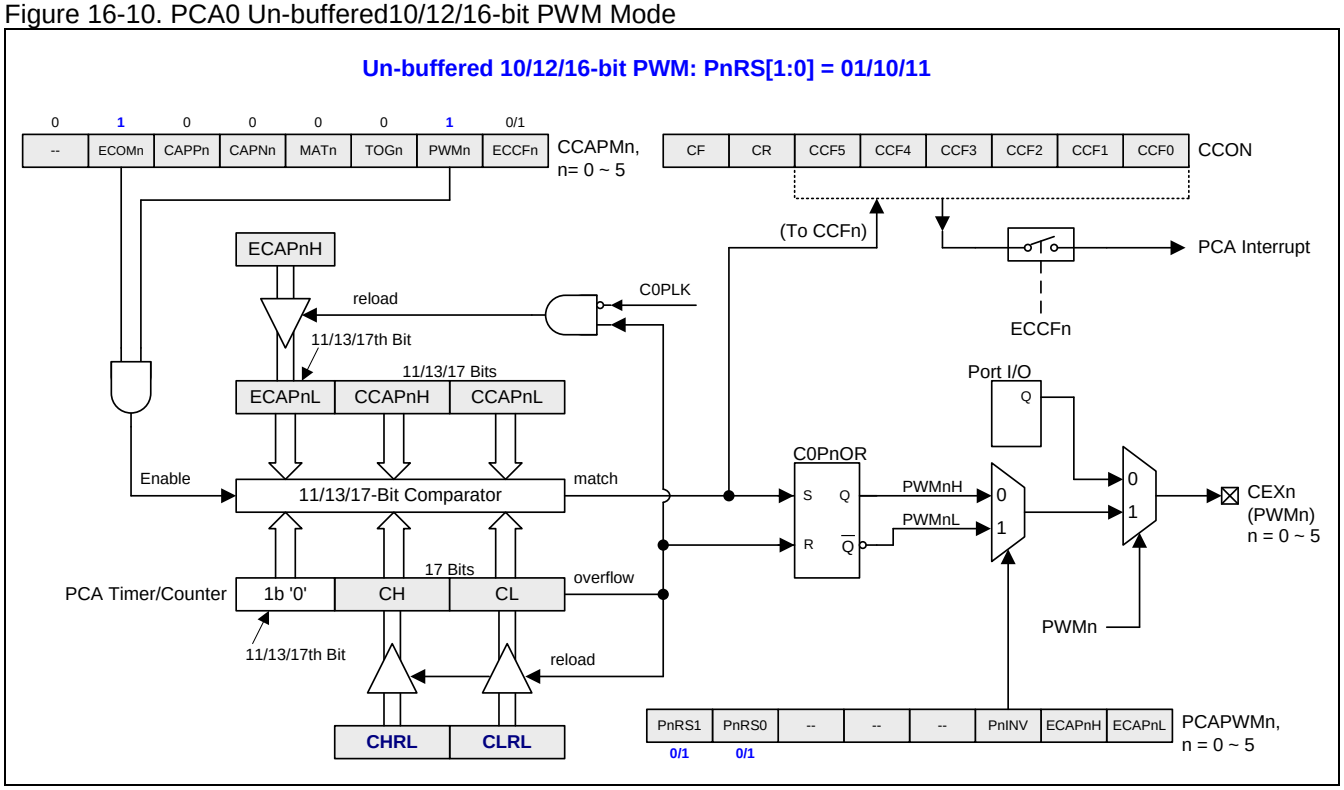
Figure 16-9. PCA0 Buffered 8-bit PWM Mode



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

The PCA0 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.

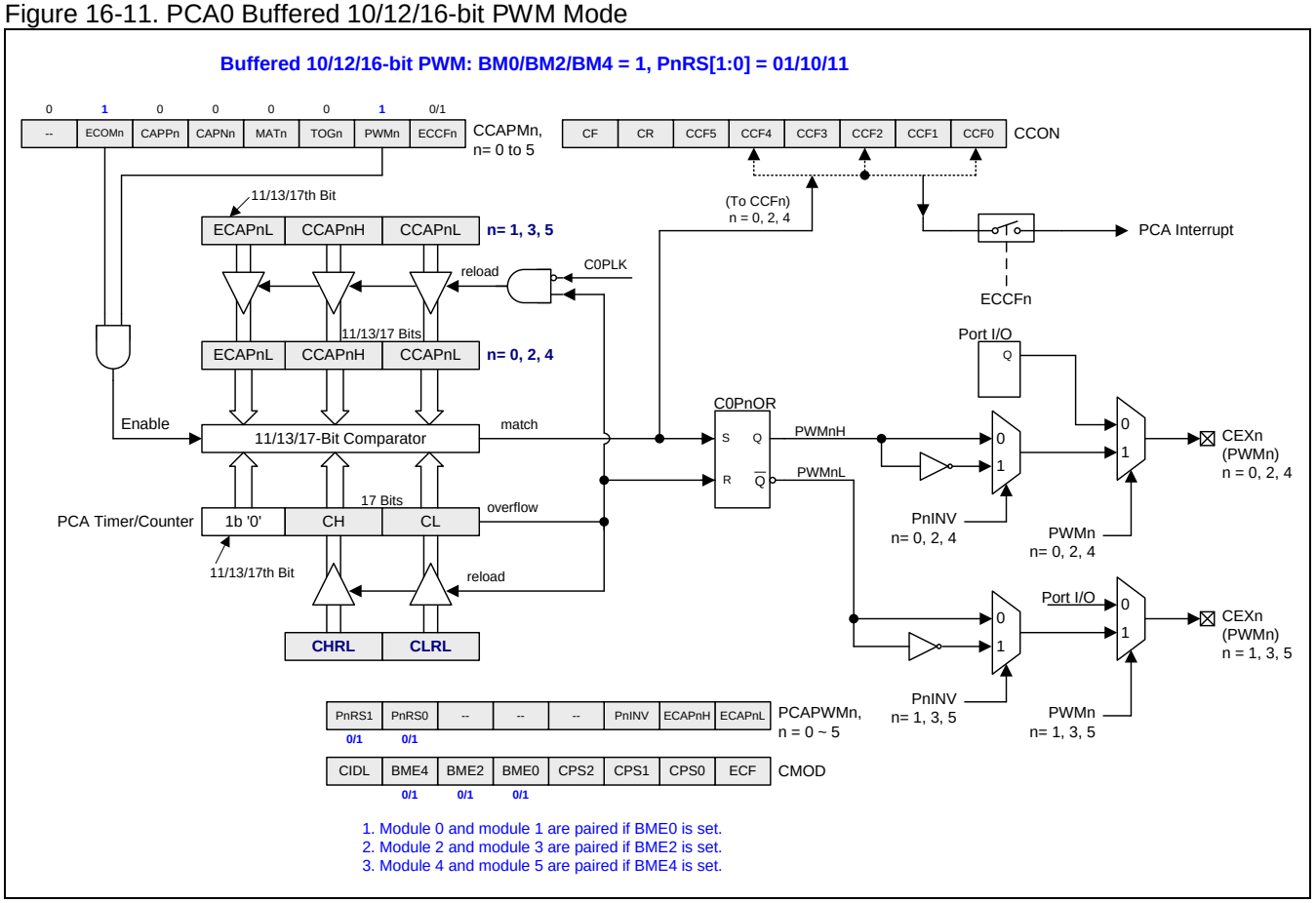
Figure 16-10. PCA0 Un-buffered10/12/16-bit PWM Mode



16.4.7. Buffered 10/12/16-bit PWM Mode

To use 10/12/16-bit PWM mode will cause unexpected duty cycle when change the duty cycle setting by writing data into CCAPnH and CCAPnL, because the 8 bit CPU can only write one byte at a time. To finish fully setting it will take two write cycles, and the comparator will output unexpected duty cycle when the first byte have been written. If the applications need accurate control when change the duty cycle, it needs to use the Buffered PWM mode.

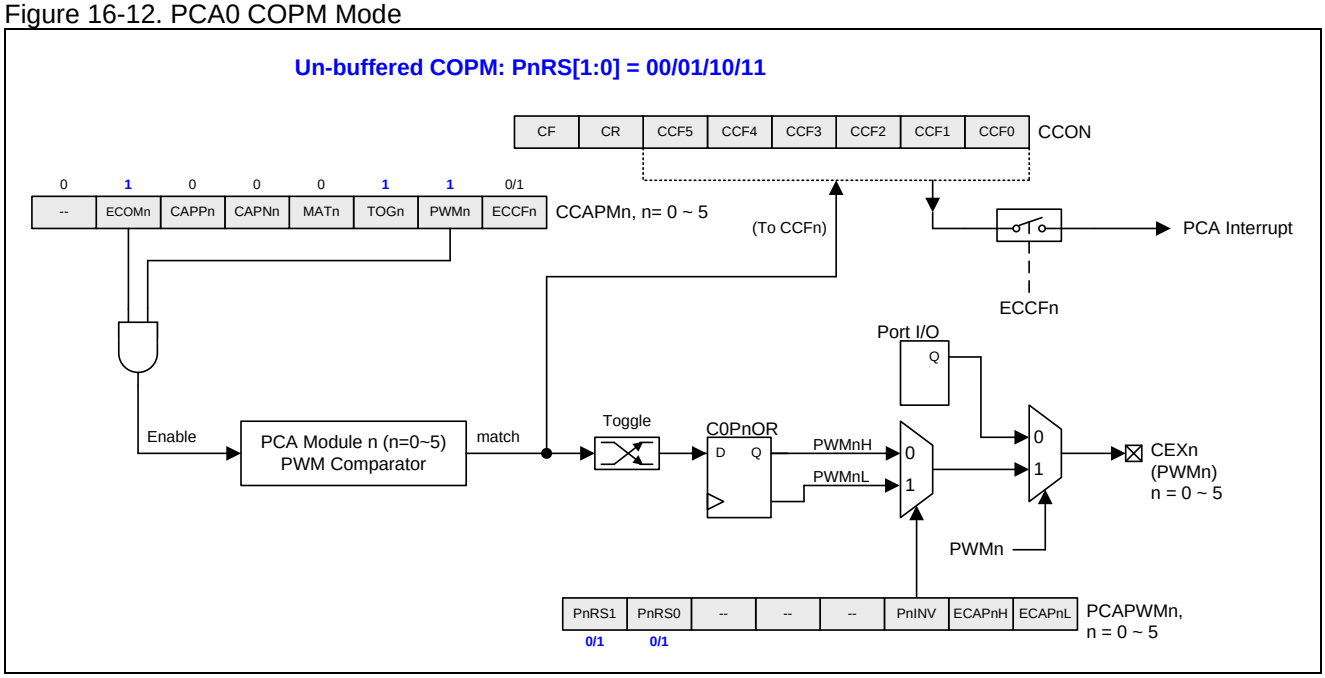
Figure 16-11. PCA0 Buffered 10/12/16-bit PWM Mode



16.4.8. COPM Mode

Compare Output on PWM Match mode is similar to High Speed Output Mode, but it uses PCA0 PWM comparators instead of fixed 16-bit comparators. It gives more flexibility to the applications. For example, if it uses 8-Bit PWM for the PCA0 comparator, the output toggle frequency can higher then High Speed Output Mode.

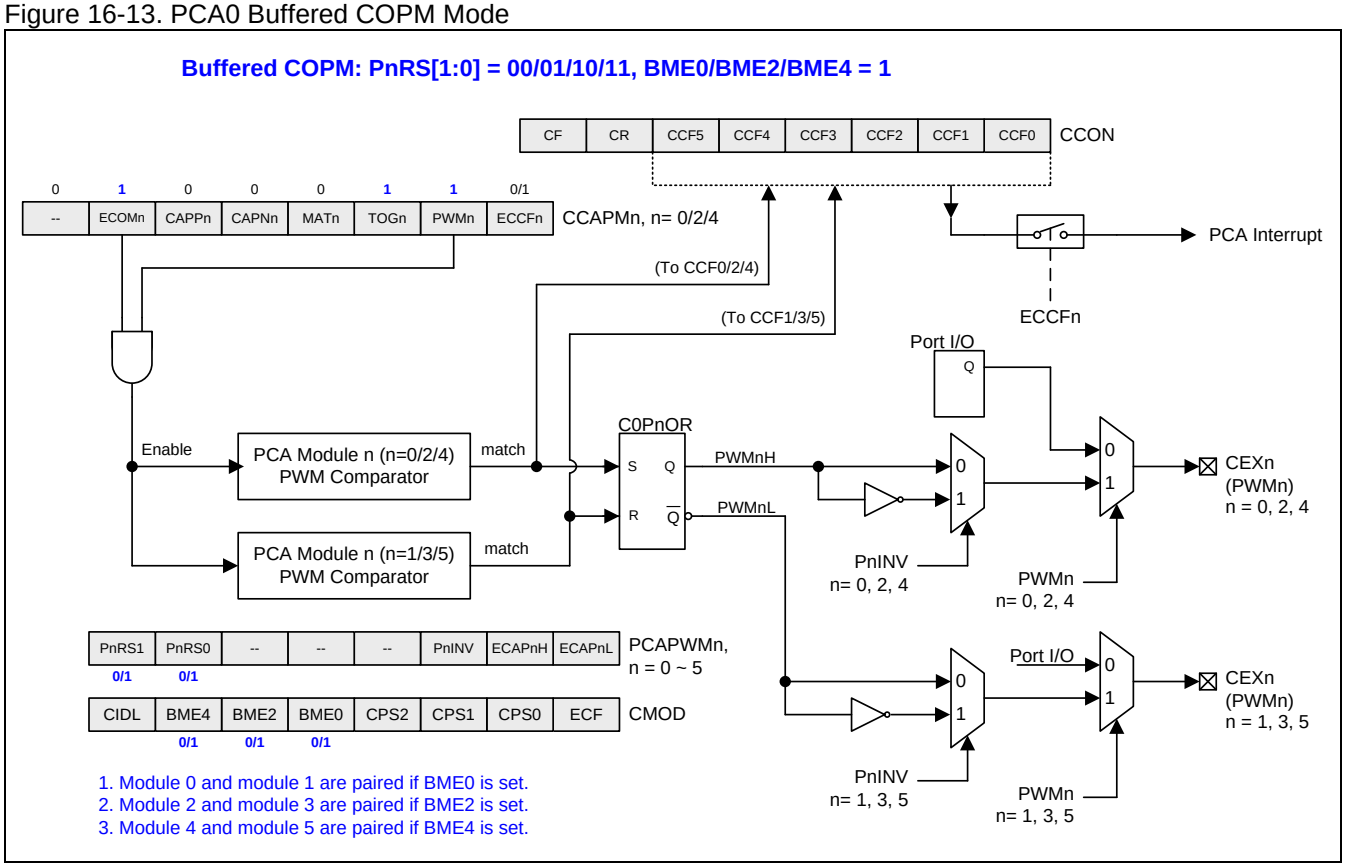
Figure 16-12. PCA0 COPM Mode



16.4.9. Buffered COPM Mode

If the applications needs to have any phase control of the PWM signals, it needs to set the PCA0 modules in buffered COPM mode. One pair of the PCA0 module (n=0&1 / 2&3 / 4&5) can program the time delay of the two edges of one cycle of the PWM signal. It means you can set the start and end point of the waveform. This is useful when the 2 or 3 correlation PWM signals can set the phase shift between each other.

Figure 16-13. PCA0 Buffered COPM Mode

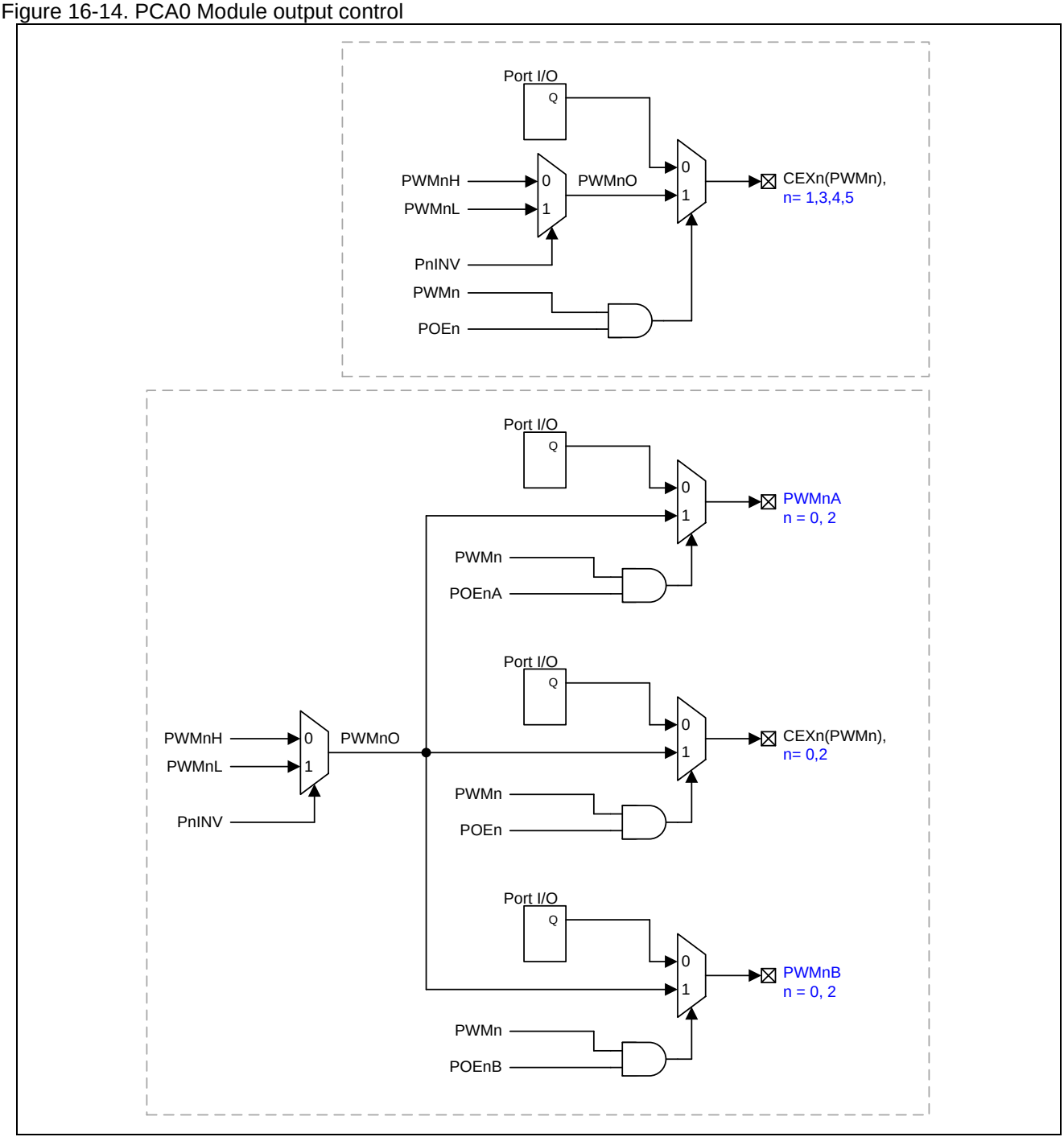


16.4.10.PCA0 Module Output Control

PCA0 modules have multi output control mode can be selected for different applications. The CEXn (n=1,3,4,5) can be programed as general I/O port or the output of the PCA0 module (PWM) 1, 3, 4 and 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.

In addition, PCA0 module 0 and 2 have 2 “Cloned” signals to the different port pin. These three the same PWM signals can be masked particular cycles by the POEnA or POEnB or PWMn for the applications which need phase control.

Figure 16-14. PCA0 Module output control



PAOE: PWM Additional Output Enable Register

SFR Page = 0~F
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, PCA0 PWM3 main channel (PWM3O) output control.
0: Disable PWM3O output on port pin.
1: Enable PWM3O output on port pin. **Default is enabled.**

Bit 6: POE2B, PCA0 PWM2 3rd channel (PWM2B) output control.
0: Disable PWM2B output on port pin. **Default is disabled.**
1: Enable PWM2B output on port pin.

Bit 5: POE2A, PCA0 PWM2 2nd channel (PWM2A) output control.
0: Disable PWM2A output on port pin. **Default is disabled.**
1: Enable PWM2A output on port pin.

Bit 4: POE2, PCA0 PWM2 main channel (PWM2O) output control.
0: Disable PWM2O output on port pin.
1: Enable PWM2O output on port pin. **Default is enabled.**

Bit 3: POE1, PCA0 PWM1 main channel (PWM1O) output control.
0: Disable PWM1O output on port pin.
1: Enable PWM1O output on port pin. **Default is enabled.**

Bit 2: POE0B, PCA0 PWM0 3rd channel (PWM0B) output control.
0: Disable PWM0B output on port pin. **Default is disabled.**
1: Enable PWM0B output on port pin.

Bit 1: POE0A, PCA0 PWM0 2nd channel (PWM0A) output control.
0: Disable PWM0A output on port pin. **Default is disabled.**
1: Enable PWM0A output on port pin.

Bit 0: POE0, PCA0 PWM0 main channel (PWM0O) output control.
0: Disable PWM0O output on port pin.
1: Enable PWM0O output on port pin. **Default is enabled.**

AUXR7: Auxiliary Register 7

SFR Page = 4 Only
SFR Address = 0xA4 RESET = 1100-xxxx

7	6	5	4	3	2	1	0
POE5	POE4	C0CKOE	SPI0M0	--	--	--	--
R/W	**R/W**	**R/W**	**R/W**	**W**	**W**	**W**	**W**

Bit 7: POE5, PCA0 PWM5 main channel (PWM5O) output control.
0: Disable PWM5O output on port pin.
1: Enable PWM5O output on port pin. **Default is enabled.**

Bit 6: POE4, PCA0 PWM4 main channel (PWM4O) output control.
0: Disable PWM4O output on port pin.
1: Enable PWM4O output on port pin. **Default is enabled.**

Bit 5: C0CKOE, PCA0 clock output enable.
0: Disable PCA0 clock output.
1: Enable PCA0 clock output with PCA0 base timer overflow rate/2.

AUXR5: Auxiliary Register 5

SFR Page = **2 only**

SFR Address = 0xA4 RESET = 0000-0000

7	6	5	4	3	2	1	0
C0IC4S0	C0IC2S0	C0PPS1	C0PPS0	C0PS1	C0PS0	ECIPS0	C0COPS
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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

C0IC4S0	CEX4 input
0	CEX4 Port Pin
1	AC0OUT

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

C0IC2S0	CEX2 input
0	CEX2 Port Pin
1	ILRCO

Bit 5: C0PPS1, {PWM2A, PWM2B} Port pin Selection 0.

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	P2.0	P2.1
1	P6.0	P6.1

Bit 3: C0PS1, PCA0 Port pin Selection 1.

C0PS1	CEX3	CEX5
0	P3.4	P3.5
1	P2.0	P2.1

Bit 2: C0PS0, PCA0 Port pin Selection0.

C0PS0	CEX0	CEX2	CEX4
0	P2.2	P2.4	P2.6
1	P3.0	P3.1	P3.3

Notice: When CEX1 and CEX4 both have been selected output to port pin, please note CEX1 will output through P3.3, therefore the CEX4 only can use P2.6 to output the signal.

Bit 1: ECIPS0, PCA0 ECI Port pin Selection0.

ECIPS0	ECI
0	P1.3
1	P1.6

Bit 0: C0COPS, PCA0 Clock Output (C0CKO) port pin Selection.

C0COPS	C0CKO
0	P4.7
1	P3.3

17. Serial Port 0 (UART0)

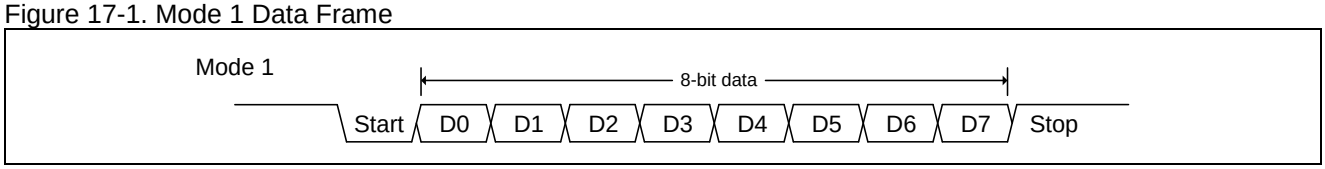
The serial port 0 of **MG82FG5D16** support full-duplex transmission, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of the second byte before a previously received byte has been read from the register. However, if the first byte hasn't been read when the second byte reception is completed, one of the byte will be lost. The serial port receive and transmit registers are both accessed at special function register S0BUF. Writing to S0BUF loads the transmit register, and reading from S0BUF accesses a physically separate receive register.

The serial port can operate in **5** modes: Mode 0 provides *synchronous* communication while Modes 1, 2, and 3 provide *asynchronous* communication. The asynchronous communication operates as a full-duplex Universal Asynchronous Receiver and Transmitter (UART), which can transmit and receive simultaneously and at different baud rates. Mode 4 in UART0 supports SPI master operation which data rate setting is same as Mode 0.

Mode 0: 8 data bits (LSB first) are transmitted or received through RXD0. TXD0 always outputs the shift clock. The baud rate can be selected to 1/12 or 1/4 the system clock frequency by URM0X3 setting in S0CFG register. In **MG82FG5D16**, 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 17–4 and Figure 17–5 show the clock polarity waveform in Mode 0.

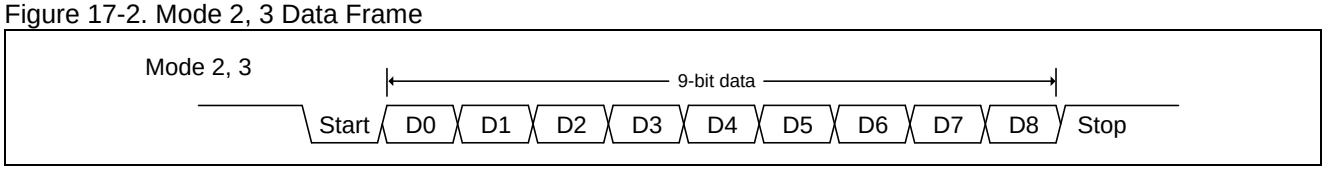
Mode 1: 10 bits are transmitted through TXD0 or received through RXD0. The frame data includes a start bit (0), 8 data bits (LSB first), and a stop bit (1), as shown in Figure 17–1. On receive, the stop bit would be loaded into RB80 in S0CON register. The baud rate is variable.

Figure 17-1. 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 17–2. 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 17-2. 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.

17.1. Serial Port 0 Mode 0

Serial data enters and exits through RXD0. TXD0 outputs the shift clock. 8 bits are transmitted/received: 8 data bits (LSB first). The shift clock source can be selected to 1/12 or 1/4 the system clock frequency by URM0X3 setting in S0CFG register. Figure 17–3 shows a simplified functional diagram of the serial port 0 in Mode 0.

Transmission is initiated by any instruction that uses S0BUF as a destination register. The “write to S0BUF” signal triggers the UART0 engine to start the transmission. The data in the S0BUF would be shifted into the RXD0(P3.0) pin by each raising edge of the shift clock on the TXD0(P3.1) pin. After eight raising edge of shift clocks passed, 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 17–4 shows the transmission waveform of Mode 0.

Reception is initiated by the condition REN0=1 and RI0=0. At the next instruction cycle, the Serial Port 0 Controller writes the bits 11111110 to the receive shift register, and in the next clock phase activates Receive.

Receive is enabled by the 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 17–5 shows the reception waveform in Mode 0.

Figure 17-3. Serial Port 0 Mode 0

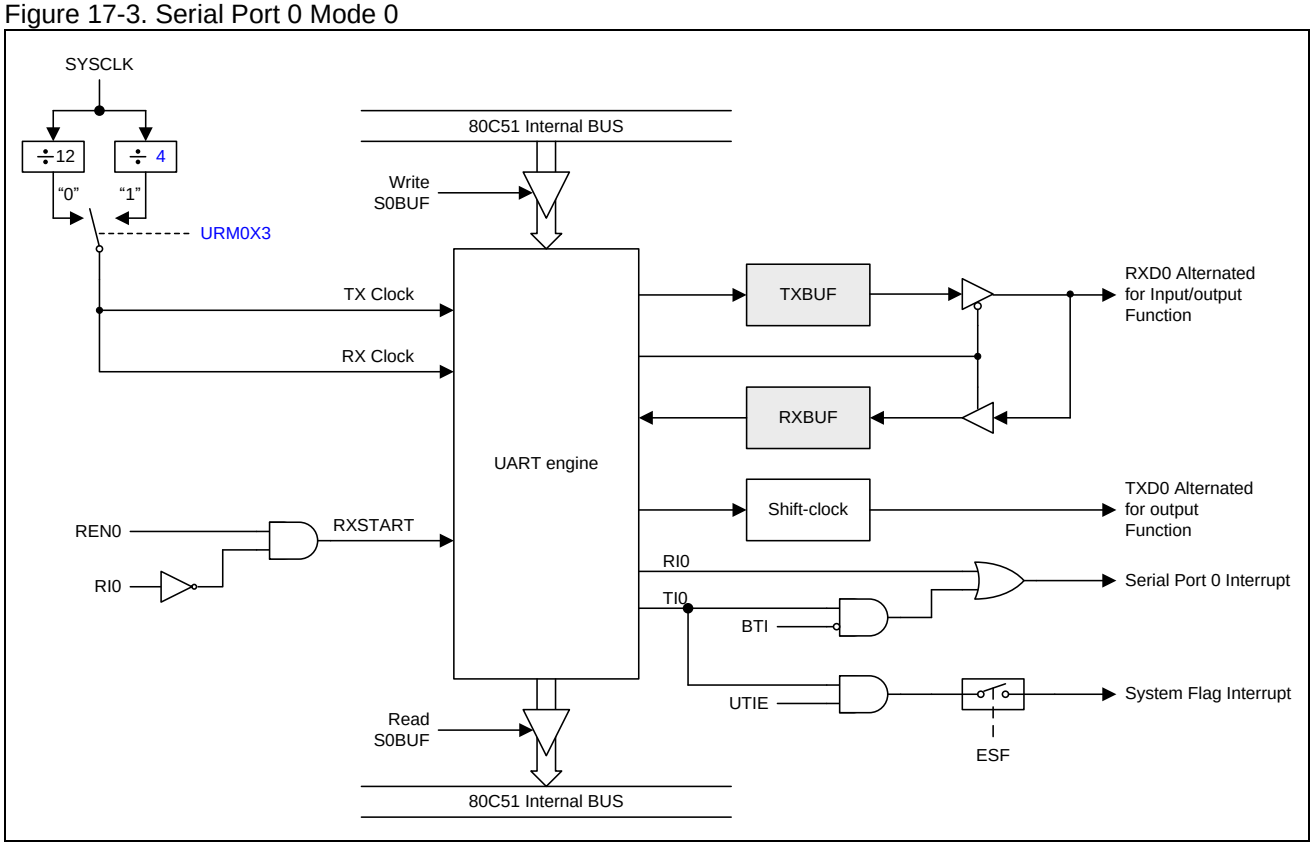


Figure 17-4. Mode 0 Transmission Waveform

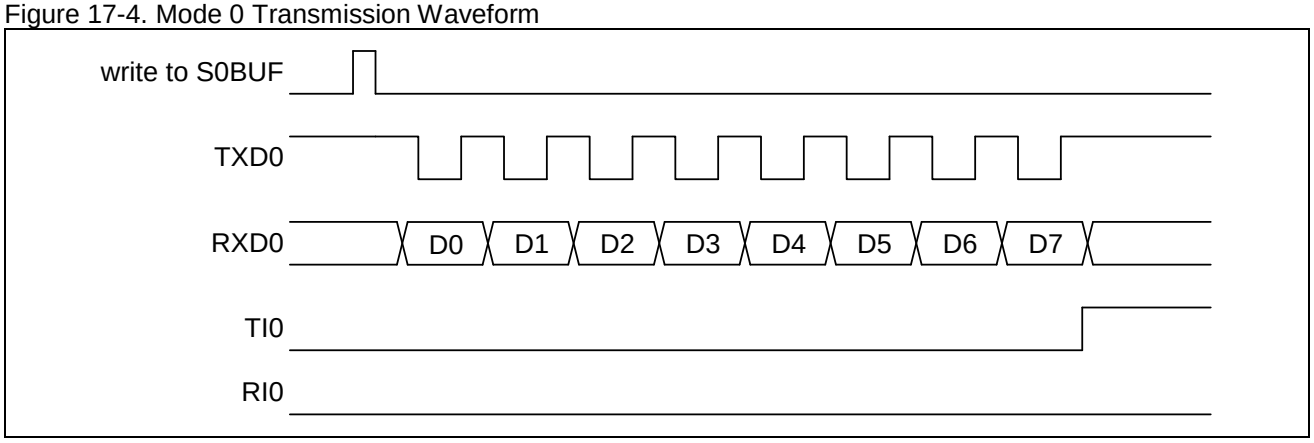
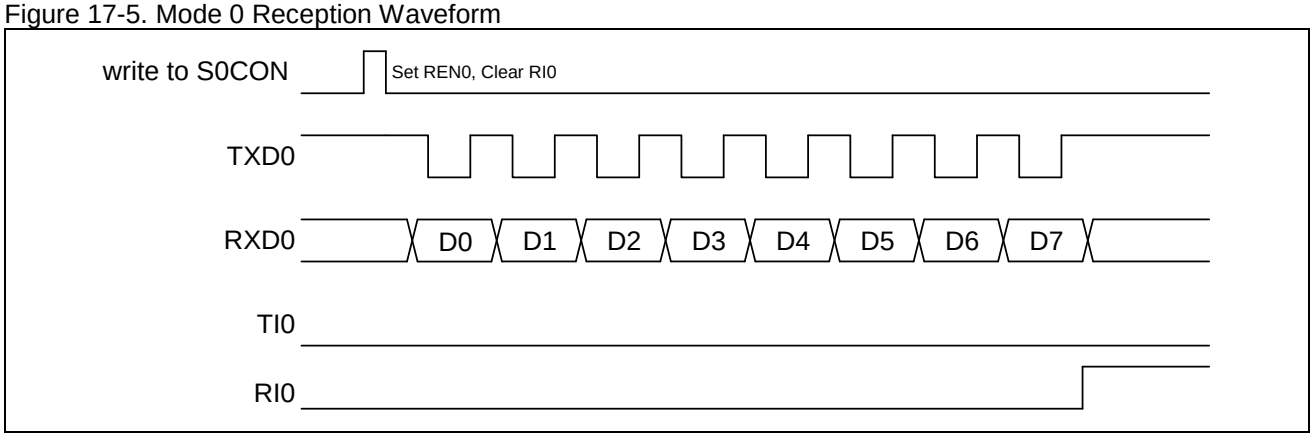


Figure 17-5. Mode 0 Reception Waveform



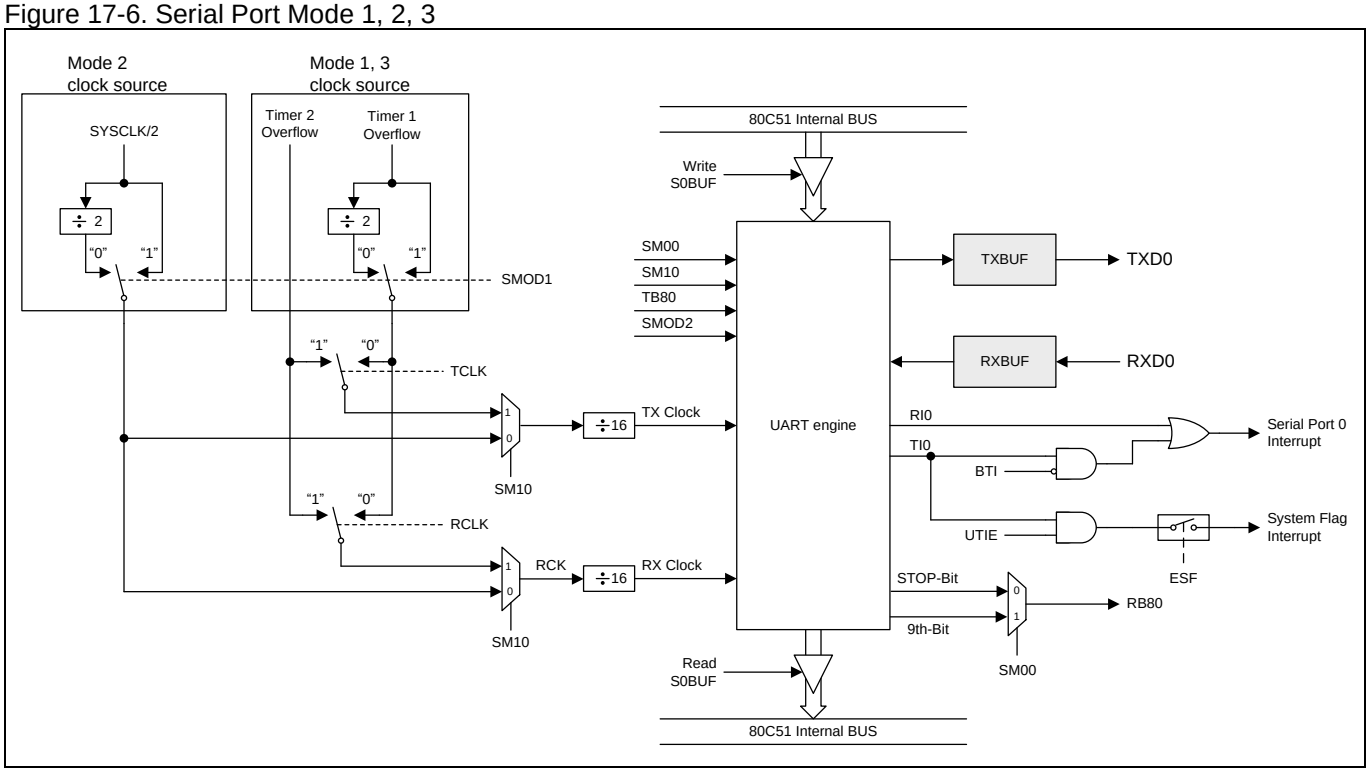
17.2. Serial Port 0 Mode 1

10 bits are transmitted through TXD0, or received through RXD0: a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB80 in S0CON. The baud rate is determined by the Timer 1 or Timer 2 overflow rate. Figure 17–1 shows the data frame in Mode 1 and Figure 17–6 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 17–1 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 17-6. Serial Port Mode 1, 2, 3



17.3. Serial Port 0 Mode 2 and Mode 3

11 bits are transmitted through TXD0, or received through RXD0: a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On transmit, the 9th data bit (TB80) can be assigned either 0 or 1. For receive, the 9th data bit goes into RB80 in S0CON. The baud rate is programmable to 1/16, 1/32 or 1/64 of the system clock frequency in Mode 2. Mode 3 may have a variable baud rate generated from Timer 1 or Timer 2.

Figure 17–2 shows the data frame in Mode 2 and Mode 3. Figure 17–5 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 17–2 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.

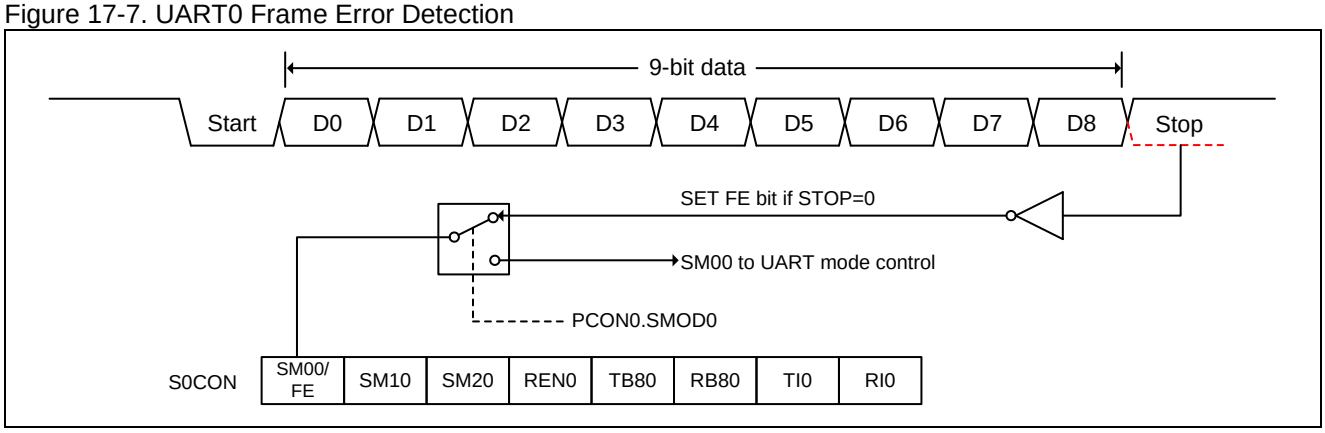
Reception is initiated when the UART0 engine detected 1-to-0 transition at RXD0 sampled by RCK. The data on the RXD0 pin would be sampled by Bit Detector in UART0 engine. After the end of 9th data bit reception, RI0 would be asserted by hardware to indicate the end of data reception and load the 9th 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.

17.4. Frame Error Detection

When used for framing error detection, the UART0 looks for missing stop bits in the communication. A missing stop bit will set the FE bit in the S0CON register. The FE bit shares the S0CON.7 bit with SM00 and the function of S0CON.7 is determined by SMOD0 bit (PCON.6). If SMOD0 is set then S0CON.7 functions as FE. S0CON.7 functions as SM00 when SMOD0 is cleared. When S0CON.7 functions as FE, it can only be cleared by firmware. Refer to Figure 17–7.

Figure 17-7. UART0 Frame Error Detection



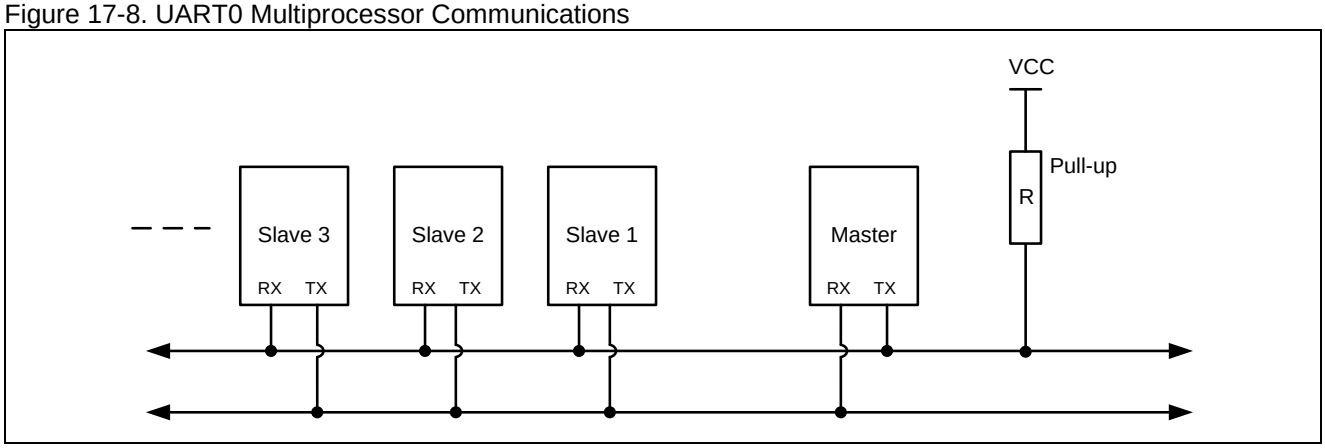
17.5. Multiprocessor Communications

Modes 2 and 3 have a special provision for multiprocessor communications as shown in Figure 17–8. In these two modes, 9 bits are received. After the 9th bit goes into RB80, and then end with a stop bit. 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 comes with “1” of the 9th bit and “0” of ~~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 slave which has been addressed will clear its SM20 bit and prepare to receive the data in the coming bytes. 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 17-8. UART0 Multiprocessor Communications



17.6. Automatic Address Recognition

Automatic Address Recognition is a feature which allows the UART0 to recognize certain addresses in the serial bit stream by using hardware to make the comparisons. This feature saves a great deal of firmware overhead by eliminating the need for the firmware to examine every serial address which passes by the serial port. This feature is enabled by setting the SM20 bit in S0CON.

In the 9 bit UART modes, mode 2 and mode 3, the Receive Interrupt flag (RI0) will be automatically set when the received byte contains either the “Given” address or the “Broadcast” address. The 9-bit mode requires that the 9th information bit is a 1 to indicate that the received information is an address and not data. Automatic address recognition is shown in Figure 17–9. The 8 bit mode is called Mode 1. In this mode the RI flag will be set if SM20 is enabled and the information received has a valid stop bit following the 8 address bits and the information is either a Given or Broadcast address. Mode 0 is the Shift Register mode and SM20 is ignored.

Using the Automatic Address Recognition feature allows a master to selectively communicate with one or more slaves by invoking the Given slave address or addresses. All of the slaves may be contacted by using the Broadcast address. Two special Function Registers are used to define the slave’s address, SADDR, and the address mask, SADEN.

SADEN is used to define which bits in the SADDR are to be used and which bits are “don’t care”. The SADEN mask can be logically ANDed with the SADDR to create the “Given” address which the master will use for addressing each of the slaves. Use of the Given address allows multiple slaves to be recognized while excluding others.

The following examples will help to show the versatility of this scheme:

Slave 0	Slave 1
SADDR = 1100 0000	SADDR = 1100 0000
SADEN = 1111 1101	SADEN = 1111 1110
Given = 1100 00X0	Given = 1100 000X

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:

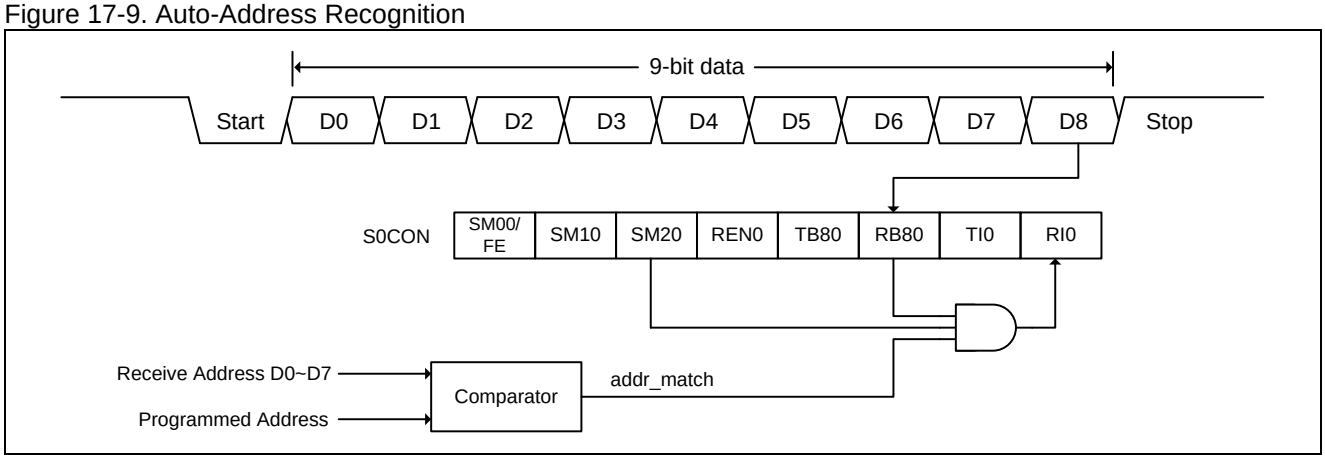
Slave 0	Slave 1	Slave 2
SADDR = 1100 0000	SADDR = 1110 0000	SADDR = 1110 0000
SADEN = 1111 1001	SADEN = 1111 1010	SADEN = 1111 1100
Given = 1100 0XX0	Given = 1110 0X0X	Given = 1110 00XX

In the above example the differentiation among the 3 slaves is in the lower 3 address bits. Slave 0 requires that bit 0 = 0 and it can be uniquely addressed by 1110 0110. Slave 1 requires that bit 1 = 0 and it can be uniquely addressed by 1110 0101. Slave 2 requires that bit 2 = 0 and its unique address is 1110 0011. To select Slaves 0 and 1 and exclude Slave 2 use address 1110 0100, since it is necessary to make bit 2 = 1 to exclude slave 2.

The Broadcast Address for each slave is created by taking the logical OR of SADDR and SADEN. Zeros in this result are treated as don't-cares. In most cases, interpreting the don't-cares as ones, the broadcast address will be FF hexadecimal.

Upon reset SADDR (SFR address 0xA9) and SADEN (SFR address 0xB9) are loaded with 0s. This produces a given address of all "don't cares" as well as a Broadcast address of all "don't cares". This effectively disables the Automatic Addressing mode and allows the micro-controller to use standard 80C51 type UART drivers which do not make use of this feature.

Figure 17-9. 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.

17.7. Baud Rate Setting

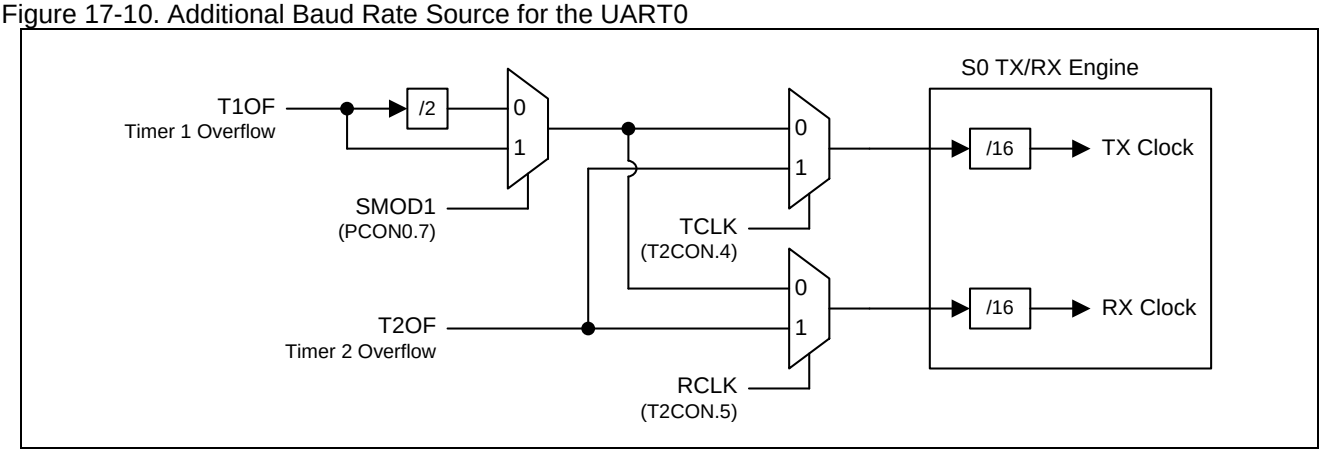
Bits T2X12 (T2MOD.4), T1X12 (AUXR2.3), URM0X3 (S0CFG.5) and SMOD2 (S0CFG.6) provide a new option for the baud rate setting, as listed below.

17.7.1. Baud Rate Selection in S0

In the Mode 1 and Mode 3 operation of the UART0, the software can select Timer 1 as the Baud Rate Generator by clearing bits TCLK and RCLK in T2CON register.

Users can also select Timer 2 as the alternated Baud Rate Generator for Mode 1 or Mode 3 of the UART0 as long as TCLK = 1, or RCLK=1. In this condition, Timer 1 is free for other application.

Figure 17-10. Additional Baud Rate Source for the UART0



17.7.2. Baud Rate in Mode 0

$$\text{Mode 0 Baud Rate} = \frac{F_{SYSCLK}}{n}$$; n=12, if URM0X3=0 ; n=4, if URM0X3=1

Note:

If URM0X6=0, the baud rate formula is as same as standard 8051.

17.7.3. 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 17–1 defines the Baud Rate setting with SMOD2 factor in Mode 2 baud rate generator.

Table 17-1. 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	Double Baud Rate **X4**	Enhanced function	± 1%

Note: When Timer 1 in Double Baud Rate x4 (SMOD1=1 & SMOD2=1) mode, the TH1 can not equal to 254 & 255.

17.7.4. Baud Rate in Mode 1 & 3

17.7.4.1 Using Timer 1 as the Baud Rate Generator

$$\text{Mode 1, 3 Baud Rate} = \frac{2^{SMOD1} \times 2^{(SMOD2 \times 2)}}{32} \times \frac{F_{SYSCLK}}{12 \times (256 - TH1)} \quad ; T1X12=0$$

$$\text{or} = \frac{2^{SMOD1} \times 2^{(SMOD2 \times 2)}}{32} \times \frac{F_{SYSCLK}}{1 \times (256 - TH1)} \quad ; T1X12=1$$

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 17–2 defines the Baud Rate setting with SMOD2 factor in Timer 1 baud rate generator.

Table 17-2. SMOD2 application criteria in Mode 1 & 3 using Timer 1

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	Double Baud Rate **X4**	Enhanced function	± 1%

Note: When Timer 1 in Double Baud Rate x4 (SMOD1=1 & SMOD2=1) mode, the TH1 can not equal to 254 & 255.

Table 17–3 ~ Table 17–18 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 17-3. Timer 1 Generated Commonly Used Baud Rates @ F_{SYSCLK}=11.0592MHz

Baud Rate	**TH1**, the Reload Value					
	T1X12=0 & SMOD2=0			**T1X12=1 & SMOD2=0**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	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 17-4. Timer 1 Generated High Baud Rates @ F_{SYSCLK} =11.0592MHz

Baud Rate	**TH1**, the Reload Value					
	T1X12=0 & SMOD2=1			**T1X12=1 & SMOD2=1**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
230.4K	--	255	0.0%	250	244	0.0%
460.8K	--	--	--	253	250	0.0%
691.2K	--	--	--	254	252	0.0%
921.6K	--	--	--	--	253	0.0%
1.3824M	--	--	--	255	~~254~~	0.0%
2.7648M	--	--	--	--	~~255~~	0.0%

Table 17-5. Timer 1 Generated Commonly Used Baud Rates @ F_{SYSCLK}=22.1184MHz

Baud Rate	**TH1**, the Reload Value					
	T1X12=0 & SMOD2=0			**T1X12=1 & SMOD2=0**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	208	160	0.0%	--	--	--
2400	232	208	0.0%	--	--	0.0%
4800	244	232	0.0%	112	--	0.0%
9600	250	244	0.0%	184	112	0.0%
14400	252	248	0.0%	208	160	0.0%
19200	253	250	0.0%	220	184	0.0%
28800	254	252	0.0%	232	208	0.0%
38400	--	253	0.0%	238	220	0.0%
57600	255	254	0.0%	244	232	0.0%
115200	--	255	0.0%	250	244	0.0%
230400	--	--	--	253	250	0.0%
460800	--	--	--	--	253	0.0%

Table 17-6. Timer 1 Generated High Baud Rates @ F_{SYSCLK}=22.1184MHz

Baud Rate	**TH1**, the Reload Value					
	T1X12=0 & SMOD2=1			**T1X12=1 & SMOD2=1**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
460.8K	--	255	0.0%	250	244	0.0%
691.2K	--	--	--	252	248	0.0%
921.6K	--	--	--	253	250	0.0%
1.3824M	--	--	--	254	252	0.0%
1.8432M				--	253	0.0%
2.7648M	--	--	--	255	~~254~~	0.0%
5.5296M	--	--	--	--	~~255~~	0.0%

Table 17-7. Timer 1 Generated Commonly Used Baud Rates @ F_{SYSCLK}=12.0MHz

Baud Rate	**TH1**, the Reload Value					
	T1X12=0 & SMOD2=0			**T1X12=1 & SMOD2=0**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	230	204	0.16%	--	--	--
2400	243	230	0.16%	100	--	0.16%
4800	--	243	0.16%	178	100	0.16%
9600	--	--	--	217	178	0.16%
14400	--	--	--	230	204	0.16%
19200	--	--	--	--	217	0.16%
28800	--	--	--	243	230	0.16%
38400	--	--	--	246	236	2.34%
57600	--	--	--	--	243	0.16%
115200	--	--	--	--	--	--

Table 17-8. Timer 1 Generated High Baud Rates @ F_{SYSCLK}=12.0MHz

Baud Rate	**TH1**, the Reload Value					
	T1X12=0 & SMOD2=1			**T1X12=1 & SMOD2=1**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
115.2K	--	--	--	243	230	0.16%
230.4K	--	--	--	--	243	0.16%
460.8K	--	--	--	--	--	--

Table 17-9. Timer 1 Generated Commonly Used Baud Rates @ F_{SYSCLK}=24.0MHz

Baud Rate	**TH1**, the Reload Value					
	T1X12=0 & SMOD2=0			**T1X12=1 & SMOD2=0**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	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 17-10. Timer 1 Generated High Baud Rates @ F_{SYSCLK}=24.0MHz

Baud Rate	**TH1**, the Reload Value					
	T1X12=0 & SMOD2=1			**T1X12=1 & SMOD2=1**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
230.4K	--	--	--	243	230	0.16%
460.8K	--	--	--	--	243	0.16%

Table 17-11. Timer 1 Generated Commonly Used Baud Rates @ F_{SYSCLK}=29.4912MHz

Baud Rate	**TH1**, the Reload Value					
	T1X12=0 & SMOD2=0			**T1X12=1 & SMOD2=0**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	192	128	0.0%	--	--	--
2400	224	192	0.0%	--	--	--
4800	240	224	0.0%	64	--	0.0%
9600	248	240	0.0%	160	64	0.0%
14400	--	--	--	192	128	0.0%
19200	252	248	0.0%	208	160	0.0%
28800	--	--	--	224	192	0.0%
38400	--	--	--	232	208	0.0%
57600	--	--	--	240	224	0.0%
115200	--	--	--	248	240	0.0%
230.4K	--	--	--	252	248	0.0%
460.8K	--	--	--	254	252	0.0%
921.6K	--	--	--	255	254	0.0%
1.8432M	--	--	--	--	255	0.0%

Table 17-12. Timer 1 Generated High Baud Rates @ F_{SYSCLK}=29.4912MHz

Baud Rate	**TH1**, the Reload Value					
	T1X12=0 & SMOD2=1			**T1X12=1 & SMOD2=1**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1.8432M	--	--	--	254	252	0.0%
2.7648M	--	--	--	--	--	--
3.6864M	--	--	--	--	~~254~~	--

Table 17-13. Timer 1 Generated Commonly Used Baud Rates @ F_{SYSCLK}=44.2368MHz

Baud Rate	**TH1**, the Reload Value					
	T1X12=0 & SMOD2=0			**T1X12=1 & SMOD2=0**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	160	64	0.0%	--	--	--
2400	208	160	0.0%	--	--	--
4800	232	208	0.0%	--	--	--
9600	244	232	0.0%	112		0.0%
14400	248	240	0.0%	160	64	0.0%
19200	250	244	0.0%	184	112	0.0%
28800	252	248	0.0%	208	160	0.0%
38400	253	250	0.0%	220	184	0.0%
57600	254	252	0.0%	232	208	0.0%
115200	255	254	0.0%	244	232	0.0%
230.4K	--	255	0.0%	250	244	0.0%
460.8K	--	--	--	253	250	0.0%
921.6K	--	--	--	--	253	0.0%
1.8432M	--	--	--	--	--	--
2.7648M	--	--	--	--	255	0.0%

Table 17-14. Timer 1 Generated High Baud Rates @ F_{SYSCLK}=44.2368MHz

Baud Rate	**TH1**, the Reload Value					
	T1X12=0 & SMOD2=1			**T1X12=1 & SMOD2=1**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
2.7648M	--	--	--	254	252	0.0%
3.6864M	--	--	--	--	253	0.0%
5.5296M	--	--	--	--	~~254~~	0.0%

Table 17-15. Timer 1 Generated Commonly Used Baud Rates @ F_{SYSCLK}=32MHz

Baud Rate	**TH1**, the Reload Value					
	T1X12=0 & SMOD2=0			**T1X12=1 & SMOD2=0**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	187	118	0.64%	--	--	--
2400	221	186	-0.79%	--	--	--
4800	239	222	2.12%	48	--	0.16%
9600	--	239	2.12%	152	48	0.16%
14400	--	--	--	187	118	0.64%
19200	--	--	--	204	152	0.16%
28800	--	--	--	221	186	-0.79%
38400	--	--	--	230	204	0.16%
57600	--	--	--	239	222	2.12%
115200	--	--	--	--	239	2.12%

Table 17-16. Timer 1 Generated High Baud Rates @ F_{SYSCLK}=32MHz

Baud Rate	**TH1**, the Reload Value					
	T1X12=0 & SMOD2=1			**T1X12=1 & SMOD2=1**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
38400	--	--	--	152	48	0.16%
57600	--	--	--	--	117	-0.08%
115200	--	--	--	--	187	0.64%
230.4K	--	--	--	--	221	-0.79%

Table 17-17. Timer 1 Generated Commonly Used Baud Rates @ F_{SYSCLK}=48.0MHz

Baud Rate	**TH1**, the Reload Value					
	T1X12=0 & SMOD2=0			**T1X12=1 & SMOD2=0**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	152	48	0.16%	--	--	--
2400	204	152	0.16%	--	--	--
4800	230	204	0.16%	--	--	--
9600	243	230	0.16%	100	--	0.16%
14400	--	239	2.12%	152	48	0.16%
19200	--	243	0.16%	178	100	0.16%
28800	--	--	--	204	152	0.16%
38400	--	--	--	217	178	0.16%
57600	--	--	--	230	204	0.16%
115200	--	--	--	243	230	0.16%
230.4K	--	--	--	--	243	0.16%

Table 17-18. Timer 1 Generated High Baud Rates @ F_{SYSCLK}=48.0MHz

Baud Rate	**TH1**, the Reload Value					
	T1X12=0 & SMOD2=1			**T1X12=1 & SMOD2=1**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
230.4K	--	--	--	230	204	0.16%
460.8K	--	--	--	243	230	0.16%
921.6K	--	--	--	--	243	0.16%
1.8432M	--	--	--	--	--	--
2.7648M	--	--	--	--	--	--
3.6864M	--	--	--	--	--	--
5.5296M	--	--	--	--	--	--

17.7.4.2 Using Timer 2 as the Baud Rate Generator

When Timer 2 is used as the baud rate generator (either TCLK or RCLK in T2CON is '1'), the baud rate is as follows.

$$\text{Mode 1, 3 Baud Rate} = \frac{2^{\text{SMOD2 X (SMOD1 + 1)}} \times F_{SYSCLK}}{32 \times (65536 - \text{(RCAP2H, RCAP2L)})} \quad ; \text{T2X12=0}$$

$$\text{or} = \frac{2^{\text{SMOD2 X (SMOD1 + 1)}} \times F_{SYSCLK}}{16 \times (65536 - \text{(RCAP2H, RCAP2L)})} \quad ; \text{T2X12=1}$$

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 17–19 defines the Baud Rate setting with SMOD2 factor in Timer 2 baud rate generator.

Table 17-19. 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	± 3%
1	1	Double Baud Rate **X2**	Enhanced function	± 2%

Note: When Timer 2 in Double Baud Rate x2 (SMOD1=1 & SMOD2=1) mode, the RCAP2H & RPAC2L can not equal to 65534 & 65535.

Table 17–20 ~ Table 17–35 list various commonly used baud rates and how they can be obtained from Timer 2 in its Baud-Rate Generator Mode. For the non-standard Baud Rate, the maximum frequency is 6MHz when F_{SYSCLK} = 48MHz).

Table 17-20. Timer 2 Generated Commonly Used Baud Rates @ F_{SYSCLK}=11.0592MHz

Baud Rate	**[RCAP2H, RCAP2L]**, the Reload Value					
	T2X12=0 & SMOD2=0			**T2X12=1 & SMOD2=0**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	65248	65248	0.0%	64960	64960	0.0%
2400	65392	65392	0.0%	65248	65248	0.0%
4800	65464	65464	0.0%	65392	65392	0.0%
9600	65500	65500	0.0%	65464	65464	0.0%
14400	65512	65512	0.0%	65488	65488	0.0%
19200	65518	65518	0.0%	65500	65500	0.0%
28800	65524	65524	0.0%	65512	65512	0.0%
38400	65527	65527	0.0%	65518	65518	0.0%
57600	65530	65530	0.0%	65524	65524	0.0%
115200	65533	65533	0.0%	65530	65530	0.0%
230400	--	--	--	65533	65533	0.0%

Table 17-21. Timer 2 Generated High Baud Rates @ F_{SYSCLK}=11.0592MHz

Baud Rate	**[RCAP2H, RCAP2L]**, the Reload Value					
	T2X12=0 & SMOD2=1			**T2X12=1 & SMOD2=1**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
230.4K	65533	65530	0.0%	65530	65524	0.0%
460.8K	--	65533	0.0%	65533	65530	0.0%
691.2K	65535	65534	0.0%	65534	65532	0.0%
921.6K	--	--	--	--	65533	0.0%
1.3824M	--	65535	0.0%	65535	~~65534~~	0.0%
2.7648M	--	--	--	--	~~65535~~	0.0%

Table 17-22. Timer 2 Generated Commonly Used Baud Rates @ F_{SYSCLK}=22.1184MHz

Baud Rate	**[RCAP2H, RCAP2L]**, the Reload Value					
	T2X12=0 & SMOD2=0			**T2X12=1 & SMOD2=0**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	64960	64960	0.0%	64384	64384	0.0%
2400	65248	65248	0.0%	64960	64960	0.0%
4800	65392	65392	0.0%	65248	65248	0.0%
9600	65464	65464	0.0%	65392	65392	0.0%
14400	65488	65488	0.0%	65440	65440	0.0%
19200	65500	65500	0.0%	65464	65464	0.0%
28800	65512	65512	0.0%	65488	65488	0.0%
38400	65518	65518	0.0%	65500	65500	0.0%
57600	65524	65524	0.0%	65512	65512	0.0%
115200	65530	65530	0.0%	65524	65524	0.0%
230400	65533	65533	0.0%	65530	65530	0.0%
460800	--	--	--	65533	65533	0.0%

Table 17-23. Timer 2 Generated High Baud Rates @ F_{SYSCLK}=22.1184MHz

Baud Rate	**[RCAP2H, RCAP2L]**, the Reload Value					
	T2X12=0 & SMOD2=1			**T2X12=1 & SMOD2=1**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
460.8K	65533	65530	0.0%	65530	65524	0.0%
691.2K	65534	65532	0.0%	65532	65528	0.0%
921.6K	--	65533	0.0%	65533	65530	0.0%
1.3824M	65535	65534	0.0%	65534	65532	0.0%
1.8432M	--	--	--	--	65533	0.0%
2.7648M	--	65535	0.0%	65535	~~65534~~	0.0%
5.5296M	--	--	--	--	~~65535~~	0.0%

Table 17-24. Timer 2 Generated Commonly Used Baud Rates @ F_{SYSCLK}=12.0MHz

Baud Rate	**[RCAP2H, RCAP2L]**, the Reload Value					
	T2X12=0 & SMOD2=0			**T2X12=1 & SMOD2=0**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	65224	65224	0.16%	64912	64912	0.16%
2400	65380	65380	0.16%	65224	65224	0.16%
4800	65458	65458	0.16%	65380	65380	0.16%
9600	65497	65497	0.16%	65458	65458	0.16%
14400	65510	65510	0.16%	65484	65484	0.16%
19200	65516	65516	2.34%	65497	65497	0.16%
28800	65523	65523	0.16%	65510	65510	0.16%
38400	--	--	--	65516	65516	2.34%
57600	--	--	--	65523	65523	0.16%
115200	--	--	--	--	--	--

Table 17-25. Timer 2 Generated High Baud Rates @ F_{SYSCLK}=12.0MHz

Baud Rate	**[RCAP2H, RCAP2L]**, the Reload Value					
	T2X12=0 & SMOD2=1			**T2X12=1 & SMOD2=1**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
115.2K	--	65523	0.16%	65523	65510	0.16%
230.4K	--	--	--	--	65523	0.16%

Table 17-26. Timer 2 Generated Commonly Used Baud Rates @ F_{SYSCLK}=24.0MHz

Baud Rate	**[RCAP2H, RCAP2L]**, the Reload Value					
	T2X12=0 & SMOD2=0			**T2X12=1 & SMOD2=0**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	64912	64912	0.16%	64288	64288	0.16%
2400	65224	65224	0.16%	64912	64912	0.16%
4800	65380	65380	0.16%	65224	65224	0.16%
9600	65458	65458	0.16%	65380	65380	0.16%
14400	65484	65484	0.16%	65432	65432	0.16%
19200	65497	65497	0.16%	65458	65458	0.16%
28800	65510	65510	0.16%	65484	65484	0.16%
38400	65516	65516	2.34%	65497	65497	0.16%
57600	65523	65523	0.16%	65510	65510	0.16%
115200	--	--	--	65523	65523	0.16%

Table 17-27. Timer 2 Generated High Baud Rates @ F_{SYSCLK}=24.0MHz

Baud Rate	**[RCAP2H, RCAP2L]**, the Reload Value					
	T2X12=0 & SMOD2=1			**T2X12=1 & SMOD2=1**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
230.4K	--	65523	0.16%	65523	65510	0.16%
460.8K	--	--	--	--	65523	0.16%

Table 17-28. Timer 2 Generated Commonly Used Baud Rates @ F_{SYSCLK}=29.4912MHz

Baud Rate	**[RCAP2H, RCAP2L]**, the Reload Value					
	T2X12=0 & SMOD2=0			**T2X12=1 & SMOD2=0**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	64768	64768	0.0%	64000	64000	0.0%
2400	65152	65152	0.0%	64768	64768	0.0%
4800	65344	65344	0.0%	65152	65152	0.0%
9600	65440	65440	0.0%	65344	65344	0.0%
14400	65472	65472	0.0%	65408	65408	0.0%
19200	65488	65488	0.0%	65440	65440	0.0%
28800	65504	65504	0.0%	65472	65472	0.0%
38400	65512	65512	0.0%	65488	65488	0.0%
57600	65520	65520	0.0%	65504	65504	0.0%
115200	65528	65528	0.0%	65520	65520	0.0%
230.4K	65532	65532	0.0%	65528	65528	0.0%
460.8K	65534	65534	0.0%	65532	65532	0.0%
691.2K	--	--	--	--	--	--
921.6K	65535	65535	0.0%	65534	65534	0.0%

Table 17-29. Timer 2 Generated High Baud Rates @ F_{SYSCLK}=29.4912MHz

Baud Rate	**[RCAP2H, RCAP2L]**, the Reload Value					
	T2X12=0 & SMOD2=1			**T2X12=1 & SMOD2=1**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
921.6K	65534	65532	0.0%	65532	65528	0.0%

Table 17-30. Timer 2 Generated Commonly Used Baud Rates @ F_{SYSCLK}=44.2368MHz

Baud Rate	**[RCAP2H, RCAP2L]**, the Reload Value					
	T2X12=0 & SMOD2=0			**T2X12=1 & SMOD2=0**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	64384	64384	0.0%	63232	63232	0.0%
2400	64960	64960	0.0%	64384	64384	0.0%
4800	65248	65248	0.0%	64960	64960	0.0%
9600	65392	65392	0.0%	65248	65248	0.0%
14400	65440	65440	0.0%	65344	65344	0.0%
19200	65464	65464	0.0%	65392	65392	0.0%
28800	65488	65488	0.0%	65440	65440	0.0%
38400	65500	65500	0.0%	65464	65464	0.0%
57600	65512	65512	0.0%	65488	65488	0.0%
115200	65524	65524	0.0%	65512	65512	0.0%
230.4K	65530	65530	0.0%	65524	65524	0.0%
460.8K	65533	65533	0.0%	65530	65530	0.0%
691.2K	65534	65534	0.0%	65532	65532	0.0%
921.6K	--	--	--	65533	65533	0.0%
1.3824M	65535	65535	0.0%	65534	65534	0.0%
2.7648M	--	--	--	65535	65535	0.0%

Table 17-31. Timer 2 Generated High Baud Rates @ F_{SYSCLK}=44.2368MHz

Baud Rate	**[RCAP2H, RCAP2L]**, the Reload Value					
	T2X12=0 & SMOD2=1			**T2X12=1 & SMOD2=1**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
2.7648M	65535	65534	0.0%	65534	65532	0.0%
5.5296M	--	65535	0.0%	65535	~~65534~~	0.0%
11.0592M	--	--	--	--	~~65535~~	0.0%

Table 17-32. Timer 2 Generated Commonly Used Baud Rates @ F_{SYSCLK}=32.0MHz

Baud Rate	**[RCAP2H, RCAP2L]**, the Reload Value					
	T2X12=0 & SMOD2=0			**T2X12=1 & SMOD2=0**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	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	--	--	--	65519	65519	2.12%

Table 17-33. Timer 2 Generated High Baud Rates @ F_{SYSCLK}=32.0MHz

Baud Rate	**[RCAP2H, RCAP2L]**, the Reload Value					
	T2X12=0 & SMOD2=1			**T2X12=1 & SMOD2=1**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
115200	65519	65502	2.12%	65501	65467	0.64%
230.4K	--	--	--	--	65501	-0.79%

Table 17-34. Timer 2 Generated Commonly Used Baud Rates @ F_{SYSCLK}=48.0MHz

Baud Rate	**[RCAP2H, RCAP2L]**, the Reload Value					
	T2X12=0 & SMOD2=0			**T2X12=1 & SMOD2=0**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
1200	64286	64286	0.00%	63036	63036	0.00%
2400	64911	64911	0.00%	64286	64286	0.00%
4800	65224	65224	0.16%	64911	64911	0.00%
9600	65380	65380	0.16%	65224	65224	0.16%
14400	65432	65432	0.16%	65328	65328	0.16%
19200	65458	65458	0.16%	65380	65380	0.16%
28800	65484	65484	0.16%	65432	65432	0.16%
38400	65497	65497	0.16%	65458	65458	0.16%
57600	65510	65510	0.16%	65484	65484	0.16%
115200	65523	65523	0.16%	65510	65510	0.16%
230.4K	--	--	--	65523	65523	0.16%

Table 17-35. Timer 2 Generated High Baud Rates @ F_{SYSCLK}=48.0MHz

Baud Rate	**[RCAP2H, RCAP2L]**, the Reload Value					
	T2X12=0 & SMOD2=1			**T2X12=1 & SMOD2=1**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
230.4K	65523	65510	0.16%	65510	65484	0.16%
460.8K	--	65522	0.16%	65523	65510	0.16%
691.2K	--	--	--	--	--	--
921.6K	--	--	--	--	65523	0.16%

17.8. Serial Port 0 Mode 4 (SPI Master)

The Serial Port of **MG82FG5D16** is embedded an additional Mode 4 to support SPI master engine. The Mode 4 is selected by SM30, SM00 and SM10. Table 17–36 shows the serial port mode definition in **MG82FG5D16**.

Table 17-36. Serial Port 0 Mode Selection

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
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

URM0X3 also controls the SPI transfer speed. If URM0X3 = 0, the SPI clock frequency is SYSCLK/12. If URM0X3 = 1, the SPI clock frequency is SYSCLK/4.

The SPI master in **MG82FG5D16** uses the TXD0 as SPICLK, RXD0 as MOSI, and S0MI as MISO. nSS is selected by MCU software on other port pin. Figure 17–11 shows the SPI connection. It also can support the configuration for multiple slaves communication in Figure 17–12.

Figure 17-11. Serial Port 0 Mode 4, Single Master and Single Slave configuration (n = 0)

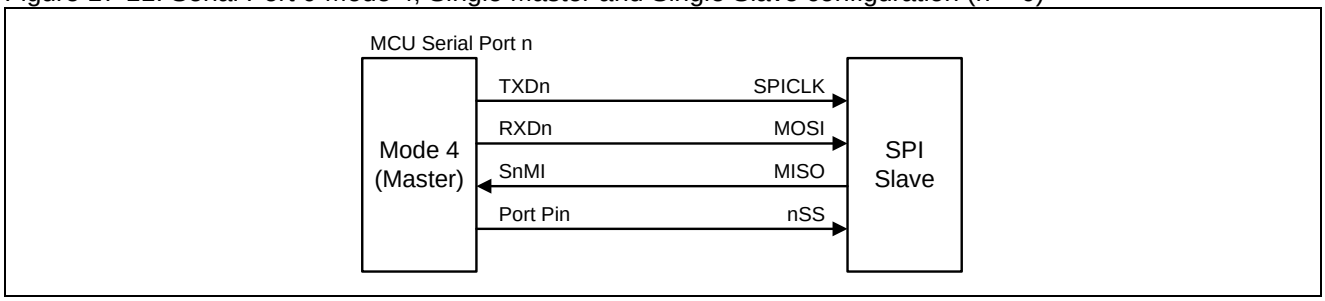
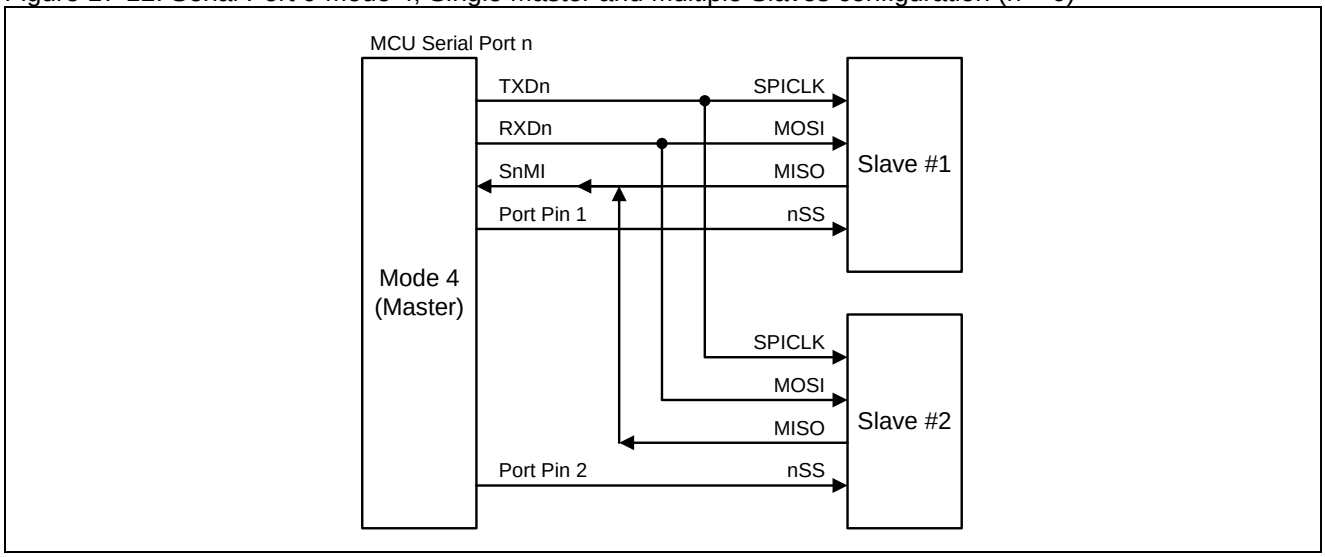


Figure 17-12. Serial Port 0 Mode 4, Single Master and Multiple Slaves configuration (n = 0)



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, **MG82FG5D16** uses an easy way by initialize SPI clock polarity to fit them. Table 17–37 shows the serial port Mode 4 mapping with the four SPI operating mode.

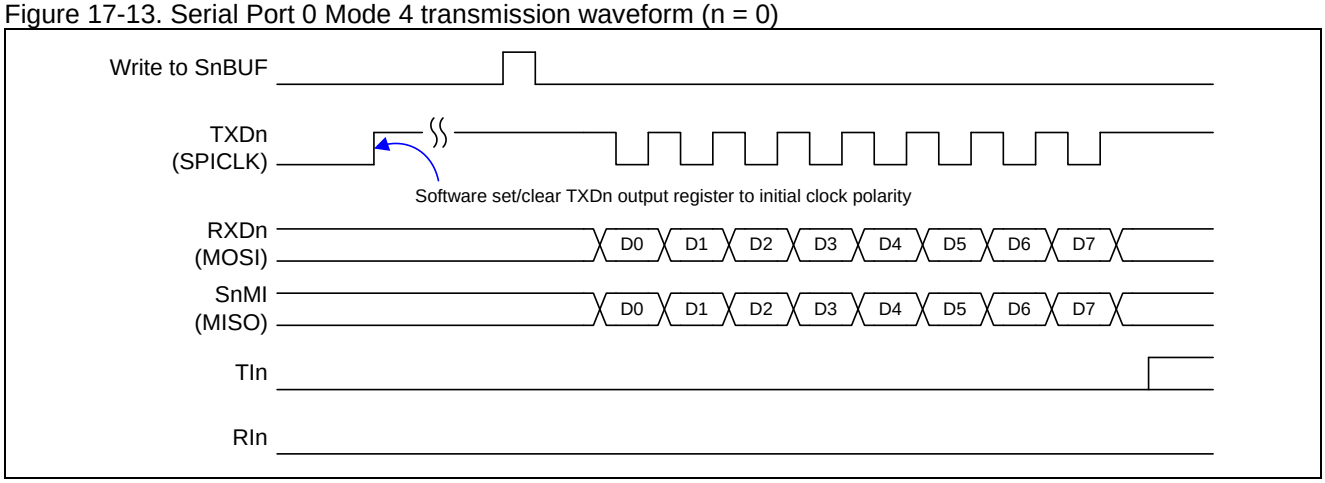
Table 17-37. SPI mode mapping with Serial Port Mode 4 configuration

SPI Mode	CPOL	CPHA	Configuration in TXD0
0	0	0	Clear TXD0 output register to “0”
1	0	1	Clear TXD0 output register to “0”
2	1	0	Set TXD0 output register to “1”
3	1	1	Set TXD0 output register to “1”

For bit order control (DORD) on SPI serial transfer, **MG82FG5D16** provides a bit S0DOR in S0CFG to control the bit order. Default value of S0DOR is 1 and control the bit order on LSB first. S0DOR is active in all S0 operating modes.

Transmission is initiated by any instruction that uses S0BUF as a destination register. The “write to S0BUF” signal triggers the UART engine to start the transmission. The data in the S0BUF would be shifted into the RXD0 pin as MOSI serial data. The SPI shift clock is built on the TXD0 pin for SPICLK output. After eight raising edge of shift clocks passing, TI0 would be asserted by hardware to indicate the end of transmission. And the contents on the S0MI pin would be sampled and shifted into shift register. Then, “read S0BUF” can get the SPI shift-in data. Figure 17–13 shows the transmission waveform in Mode 0. RI0 will not be asserted in Mode 4.

Figure 17-13. Serial Port 0 Mode 4 transmission waveform (n = 0)



17.9. Serial Port 0 Register

All the four operation modes of the serial port are the same as those of the standard 8051 except the baud rate setting. Two registers, PCON0 and S0CFG, are related to the baud rate setting:

S0CON: Serial port 0 Control Register

SFR Page = **0 ~ F**

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 or /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.
0: Disable SM20 function.
1: Enable the automatic address recognition feature in Modes 2 and 3. If SM20=1, RI0 will not be set unless the received 9th data bit is 1, indicating an address, and the received byte is a Given or Broadcast address. In mode1, if SM20=1 then RI0 will not be set unless a valid stop Bit was received, and the received byte is a Given or Broadcast address. In Mode 0, SM20 should be 0.

Bit 4: REN0, Enable serial reception.
0: Clear by software to disable reception.
1: Set by software to enable reception.

Bit 3: TB80, The 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired.

Bit 2: RB80, In Modes 2 and 3, the 9th data bit that was received. In Mode 1, if SM20 = 0, RB80 is the stop bit that was received. In Mode 0, RB80 is not used.

Bit 1: TI0. Transmit interrupt flag.
0: Must be cleared by software.
1: Set by hardware at the end of the 8th bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission.

Bit 0: RI0. Receive interrupt flag.
0: Must be cleared by software.
1: Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM20).

S0BUF: Serial port 0 Buffer Register

SFR Page = **0 ~ F**

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

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

SFR Address = 0x87 POR = 0001-0000, 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 ~ F**

SFR Address = 0x9C RESET = x000-100x

7	6	5	4	3	2	1	0
--	SMOD2	URM0X3	SM30	S0DOR	BTI	UTIE	--
W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**W**

Bit 7: Reserved. Software must write “0” on this bit when S0CFG is written.

Bit 6: SMOD2, UART0 extra double baud rate selector.
0: Disable extra double baud rate for UART0.
1: Enable extra double baud rate for UART0.

Bit 5: URM0X3, Serial Port mode 0 and mode 4 baud rate selector.
0: Clear to select SYSCLK/12 as the baud rate for UART Mode 0 and Mode 4.
1: Set to select SYSCLK/4 as the baud rate for UART Mode 0 and Mode 4.

Bit 4: SM30, Serial Port Mode control bit 3. This bit function is defined with SM00 and SM10.

Bit 3: S0DOR, Serial Port 0 data order control in all operating modes.
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.

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.

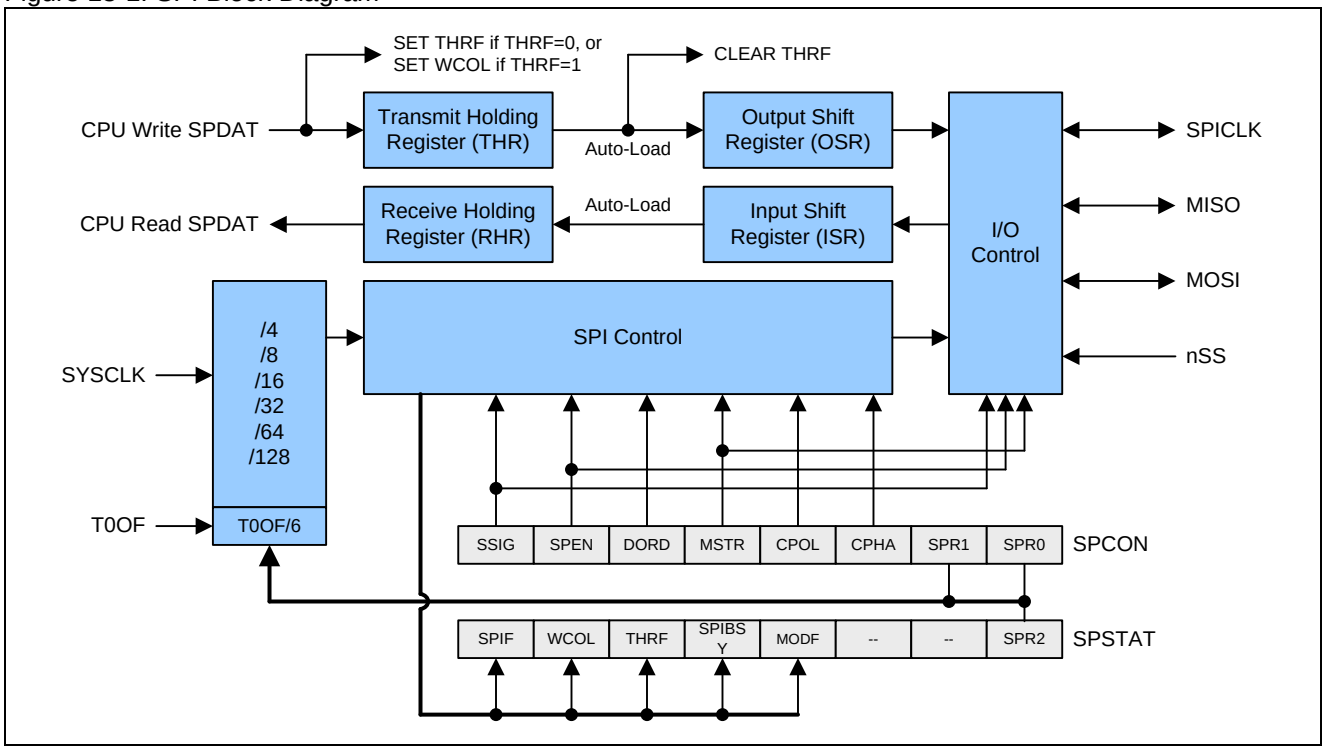
Bit 1: UTIE, S0 TI0 Enabled in system flag interrupt.
0: Disable the interrupt vector sharing for TI0 in system flag interrupt.
1: Set TI0 flag will share the interrupt vector with system flag interrupt.

Bit 0: Reserved. Software must write “0” on this bit when S0CFG is written.

18. Serial Peripheral Interface (SPI)

The **MG82FG5D16** 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 3 Mbps can be supported in Master mode under a 12MHz 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 18-1. 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 (SPCON.6) = 0, these pins function as normal I/O pins.

• /SS is the optional slave select pin. In a typical configuration, an SPI master asserts one of its port pins to select one SPI device as the current slave. An SPI slave device uses its nSS pin to determine whether it is selected. The /SS is ignored if any of the following conditions are true:

- If the SPI system is disabled, i.e. SPEN (SPCON.6) = 0 (reset value).
- If the SPI is configured as a master, i.e., MSTR (SPCON.4) = 1, and P1.4 (nSS) is configured as an output.
- If the /SS pin is ignored, i.e. SSIG (SPCON.7) bit = 1, this pin is configured for port functions.

*Note: See the **AUXR4** in Section "4.3 Alternate Function Redirection", for its alternate pin-out option.*

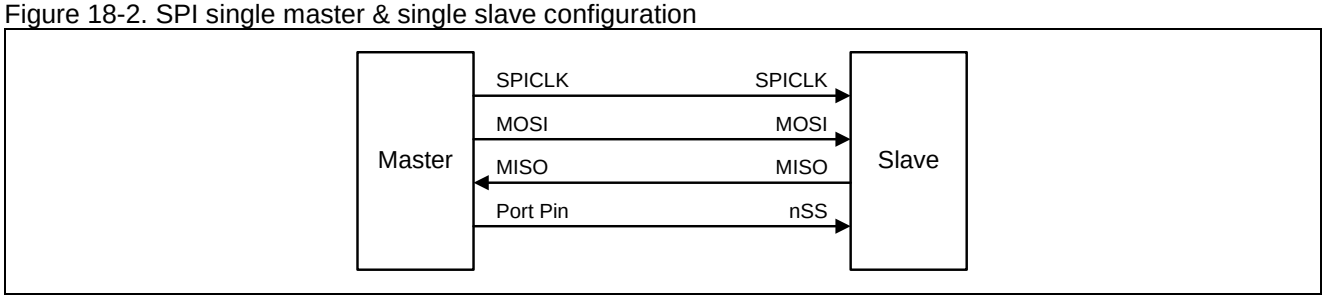
Note that even if the SPI is configured as a master (MSTR=1), it can still be **converted** to slave mode by the logic low of nSS pin input (if SSIG=0). Should this happen, the SPIF bit (SPSTAT.7) will be set **and SPEN will be cleared**. (See Section "18.2.3 Mode Change on nSS-pin")

18.1. Typical SPI Configurations

18.1.1. Single Master & Single Slave

For the master: any port pin, including P1.4 (nSS), can be used to drive the nSS pin of the slave.
For the slave: SSIG is '0', and n/SS pin is used to determine whether it is selected.

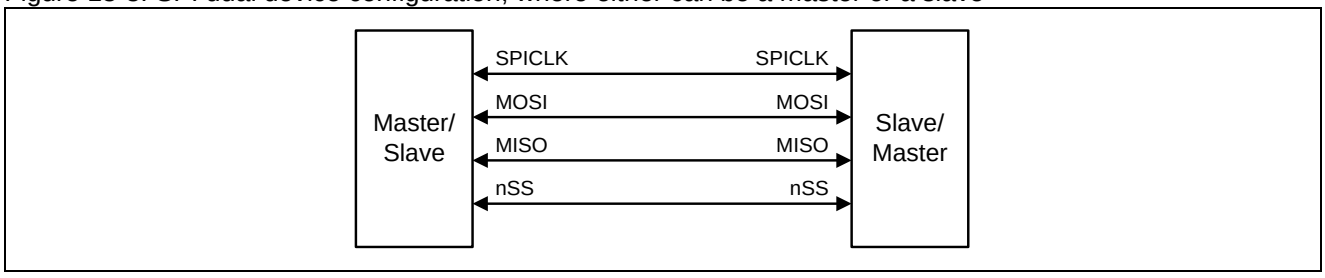
Figure 18-2. SPI single master & single slave configuration



18.1.2. 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 P1.4 (nSS) configured in quasi-bidirectional mode. When any device initiates a transfer, it can configure P1.4 as an output and drive it low to force a "mode change to slave" in the other device. (See Section "18.2.3 Mode Change on nSS-pin")

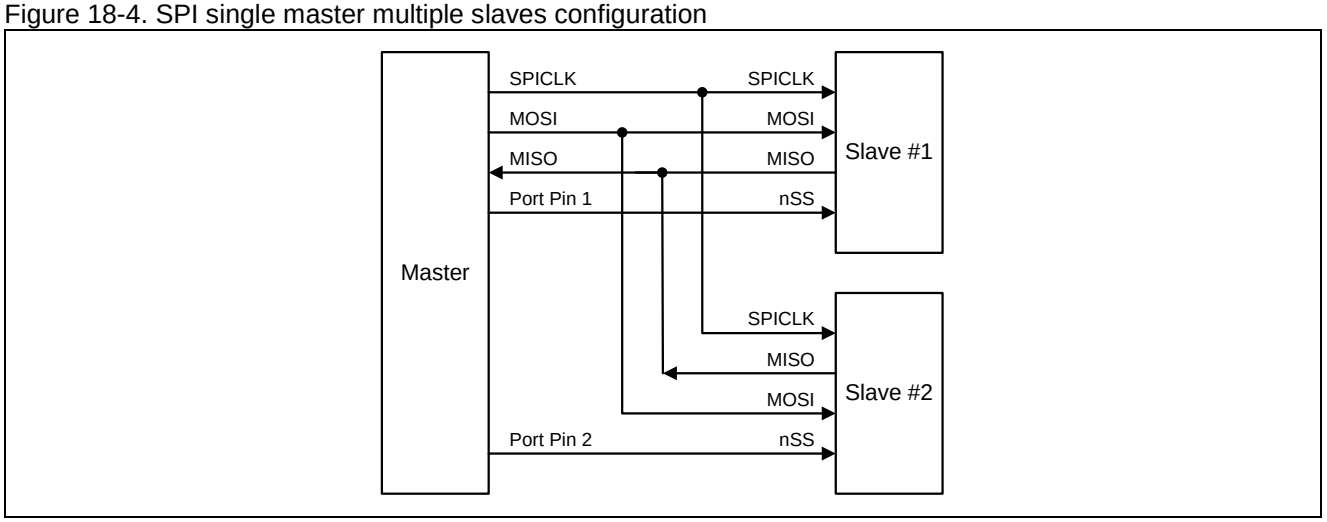
Figure 18-3. SPI dual device configuration, where either can be a master or a slave



18.1.3. Single Master & Multiple Slaves

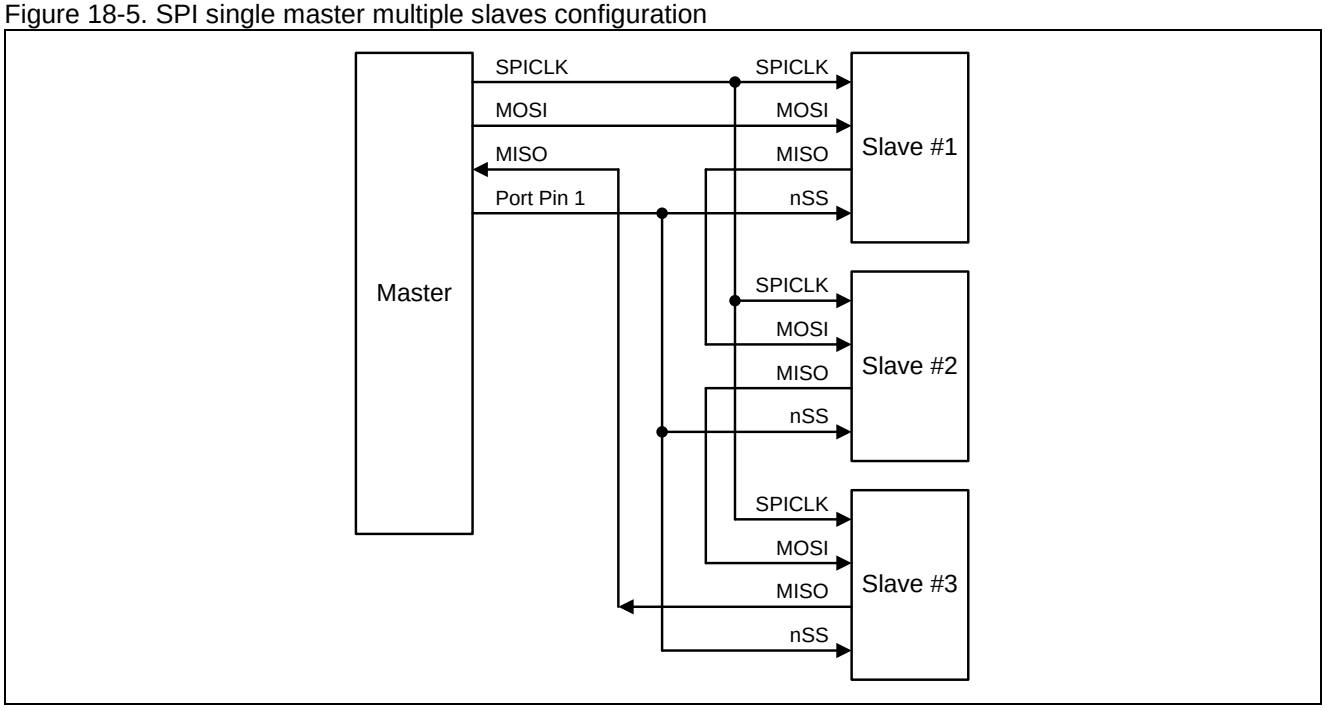
For the master: any port pin, including P1.4 (nSS), can be used to drive the nSS pins of the slaves. For all the slaves: SSIG is '0', and nSS pin are used to determine whether it is selected.

Figure 18-4. SPI single master multiple slaves configuration



18.1.4. Daisy-Chain Connection (MCU in SPI slave)

Figure 18-5. SPI single master multiple slaves configuration



18.2. Configuring the SPI

Table 18–1 shows configuration for the master/slave modes as well as usages and directions for the modes.

Table 18-1. 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	P1.4~P1.7 are used as general port pins.
1	0	0	0	Slave (selected)	output	input	input	Selected as slave.
1	0	1	0	Slave (not selected)	Hi-Z	input	input	Not selected.
1	0	0	1 ➔ 0	Slave (by mode change)	output	input	input	Mode change to slave if nSS pin is driven low, then MSTR will be cleared to '0' by 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".

18.2.1. 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.

18.2.2. 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.

18.2.3. Mode Change on nSS-pin

If SPEN=1, SSIG=0, MSTR=1 and /SS pin=1, the SPI is enabled in master mode. In this case, another master can drive this pin low to select this device as an SPI slave and start sending data to it. To avoid bus contention, the SPI becomes a slave. As a result of the SPI becoming a slave, the MOSI and SPICLK pins are forced to be an input and MISO becomes an output. The SPIF flag in SPSTAT is set, and if the SPI interrupt is enabled, an SPI interrupt will occur. User software should always check the MSTR bit. If this bit is cleared by a slave select and the user wants to continue to use the SPI as a master, the user must set the MSTR bit again, otherwise it will stay in slave mode.

18.2.4. 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).

18.2.5. Write Collision

The SPI in **MG82FG5D16** 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.

18.2.6. 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 18–2.

Table 18-2. 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/128	93.75 KHz	375 KHz
1	1	0	Reserved	--	--
1	1	1	T0OF/6	Variable	Variable

Note:
1. *SYSCLK is the system clock.*
2. *T0OF is Timer 0 Overflow.*

18.3. 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.

Figure 18-6. SPI Slave Transfer Format with CPHA=0

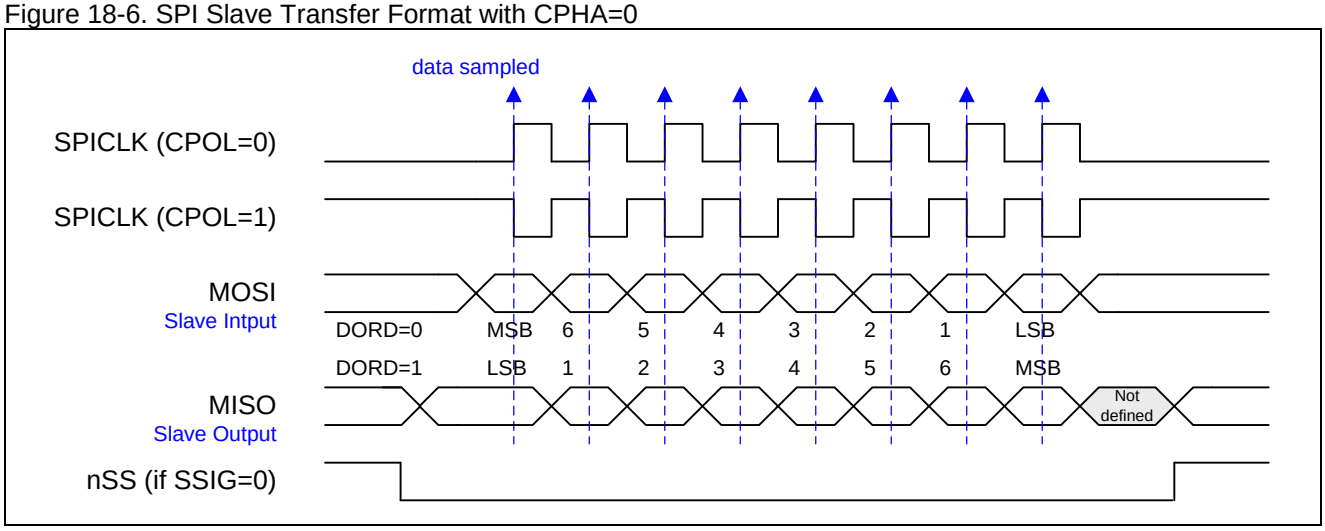


Figure 18-7. Slave Transfer Format with CPHA=1

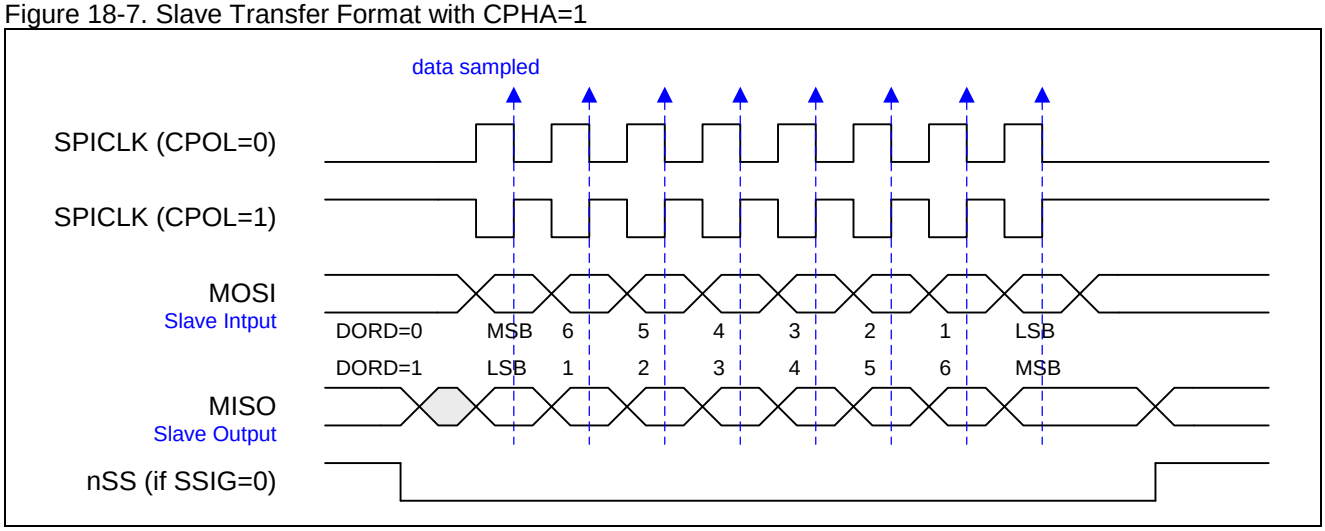


Figure 18-8. SPI Master Transfer Format with CPHA=0

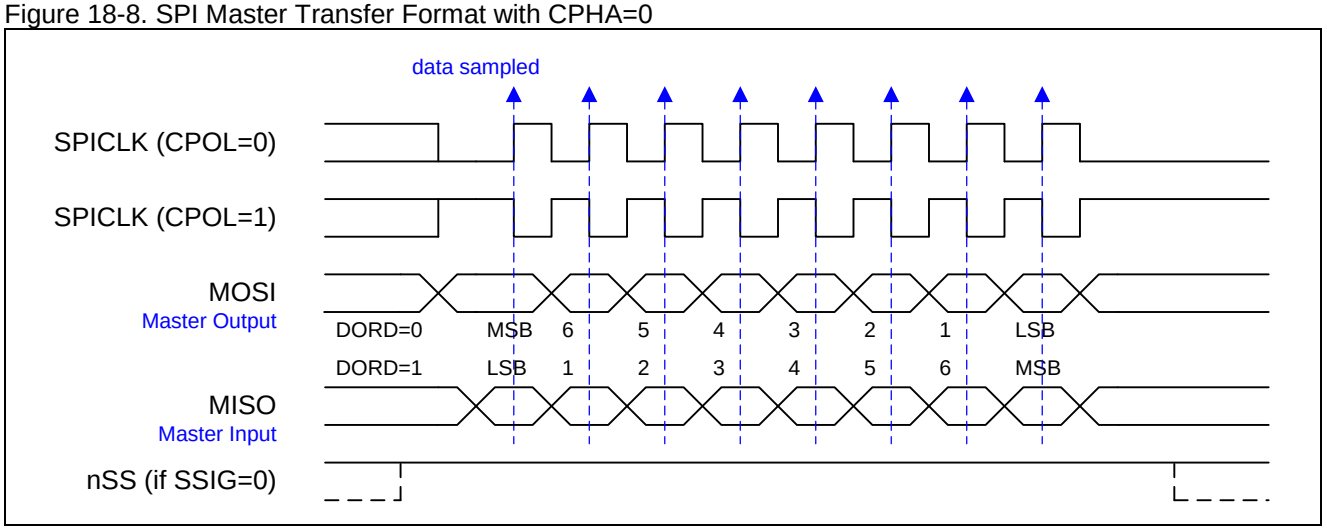
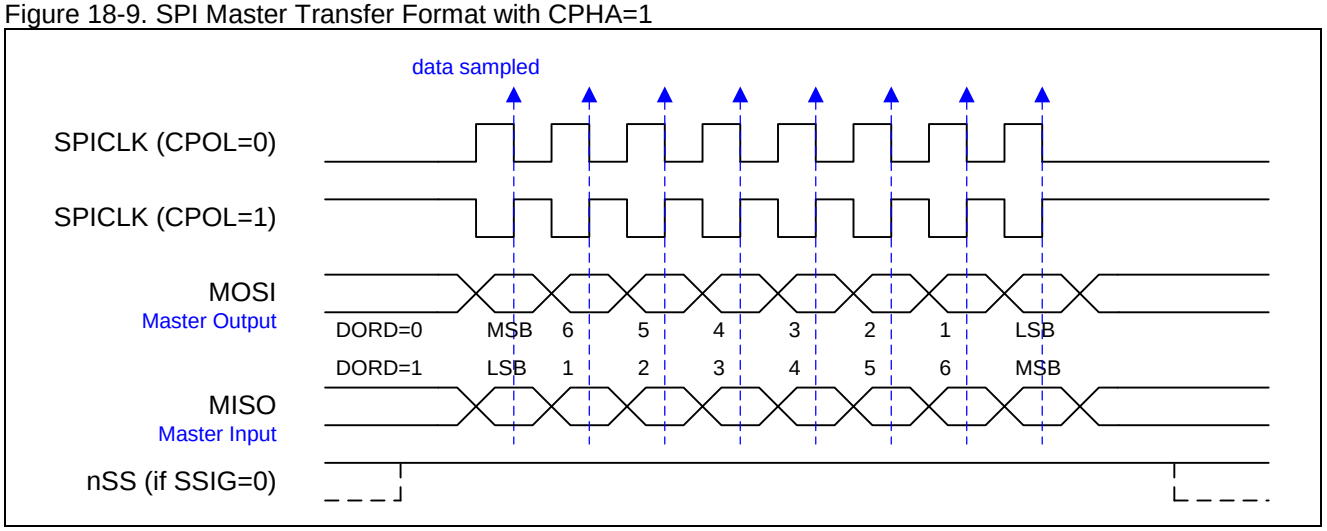


Figure 18-9. SPI Master Transfer Format with CPHA=1



18.4. SPI Register

The following special function registers are related to the SPI operation:

SPCON: SPI Control Register

SFR Page = 0~F

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 /SS pin is low (SSIG=0) and changes on the trailing edge of SPICLK. Data is sampled on the leading edge of SPICLK.
1: Data is driven on the leading edge of SPICLK, and is sampled on the trailing edge.
(Note: If SSIG=1, CPHA must not be 1, otherwise the operation is not defined.)

Bit 1~0: SPR1-SPR0, SPI clock rate select 0 & 1 (associated with SPR2, when in master mode)

SPR2	SPR1	SPR0	SPI Clock Selection	SPI Clock Rate @ SYSCLK=12MHz	SPI Clock Rate @ SYSCLK=48MHz
0	0	0	SYSCLK/4	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/128	93.75 KHz	375 KHz
1	1	0	Reserved	--	--
1	1	1	T0OF/6	Variable	Variable

Note:

1. SYSCLK is the system clock.
2. T0OF is Timer 0 Overflow.

SPSTAT: SPI Status Register

SFR Page = 0~F

SFR Address = 0x84 RESET= 0000-0xx0

7	6	5	4	3	2	1	0
SPIF	WCOL	THRF	SPIBSY	MODF	0	0	SPR2
R/W	R/W	R	R	R/W	W	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 "18.2.5 Write Collision").

Bit 5: THRF, Transmit Holding Register (THR) Full flag. Read only.

0: Means the THR is "empty". This bit is cleared by hardware when the THR is empty. That means the data in THR is loaded (by H/W) into the Output Shift Register to be transmitted, and now the user can write the next data byte to SPDAT for next transmission.

1: Means the THR is "full". This bit is set by hardware just when SPDAT is written by software.

Bit 4, SPIBSY, SPI Busy flag. Read only.

0: It indicates SPI engine is idle and all shift registers are empty.

1: It is set to logic 1 when a SPI transfer is in progress (Master or slave Mode).

Bit 3: Mode Fault Flag. This bit is set to logic 1 by hardware when a master mode collision is detected (nSS is low, MSTEN = 1, and SSIG = 0). If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software writing "1".

Bit 2~1: Reserved. Software must write "0" on these bits when SPSTAT is written.

Bit 0: SPR2, SPI clock rate select 2 (associated with SPR1 and SPR0)

SPDAT: SPI Data Register

SFR Page = 0~F

SFR Address = 0x86 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.

AUXR7: Auxiliary Register 7

SFR Page = **4 only**

SFR Address = 0xA4 RESET = 1100-xxxx

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

Bit 4: SPI0M0, SPI0 model control bit 0. It controls the SPI application with daisy-chain connection.

0: Disable the mode control.

1: Enable the mode control.

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	SPIPS0	--	AC0OE	AC0FLT1
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**W**	**R/W**	**R/W**

Bit 3: SPIPS0, SPI Port Selection 0.

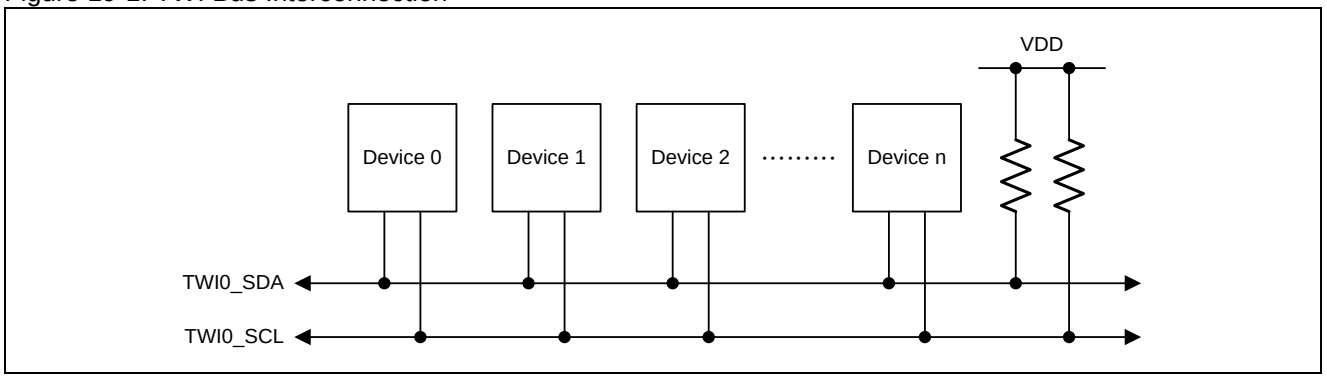
SPIPS0	nSS	MOSI	MISO	SPICLK
0	P1.4	P1.5	P1.6	P1.7
1	P3.4	P3.5	P4.1	P4.0

19. Two Wire serial Interface (TWI0)

The Two-Wire serial Interface is a two-wire, bi-directional serial bus. It is ideally suited for typical microcontroller applications.

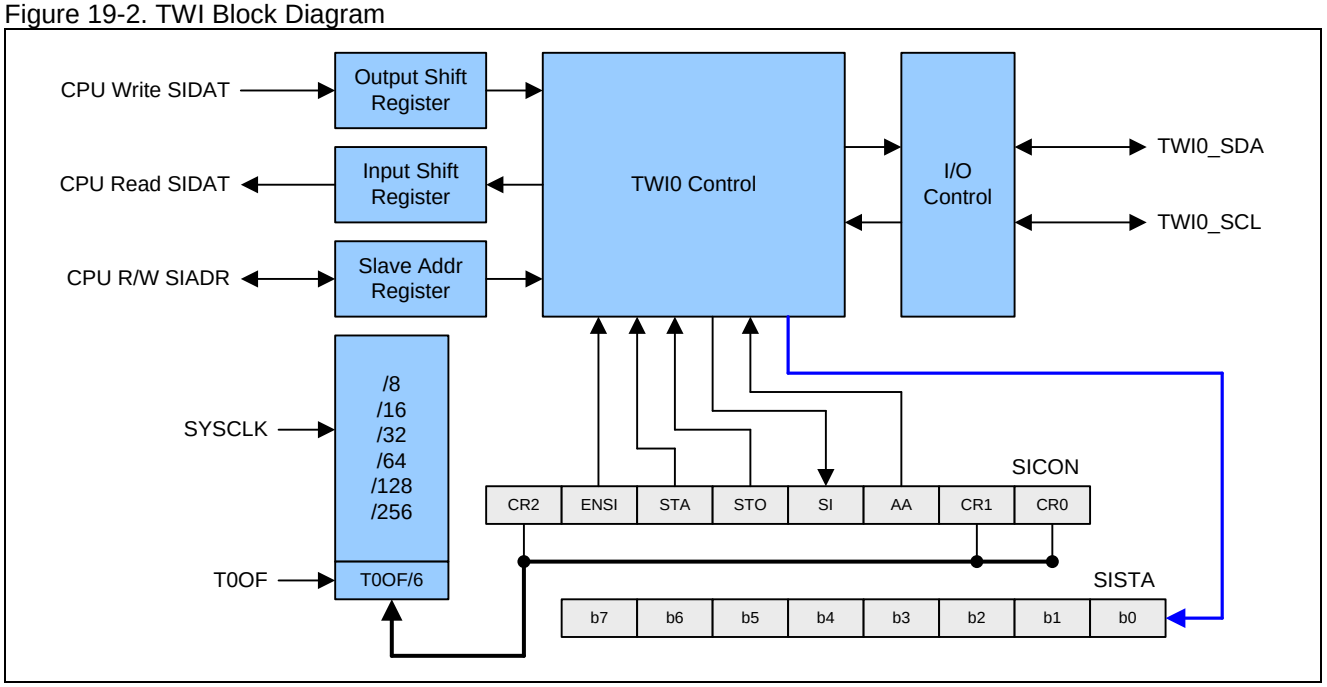
The TWI protocol allows the systems designer to interconnect up to 128 different devices using only two bi-directional bus lines, one for clock (TWI0_SCL) and one for data (TWI0_SDA). The TWI bus provides control of _TWI0_SDA (serial data), TWI0_SCL (serial clock) generation and synchronization, arbitration 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 protocol.

Figure 19-1. TWI Bus Interconnection



The TWI bus may operate as a master and/or slave, and may function on a bus with multiple masters. The CPU interfaces to the TWI through the following four special function registers: SICON configures the TWI bus; SISTA reports the status code of the TWI bus; and SIDAT is the data register, used for both transmitting and receiving TWI data. SIADR is the slave address register. And, the TWI hardware interfaces to the serial bus via two lines: SDA (serial data line) and SCL (serial clock line).

Figure 19-2. TWI Block Diagram



19.1. Operating Modes

There are four operating modes for the TWI: 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 interrupt is enabled), and the new status code can be used to determine which appropriate routine in the software is to branch to.

19.1.1. 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. If the AA bit is reset, TWI 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 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 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 as a 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 may switch to the master receiver mode by loading SIDAT with SLA+R.

19.1.2. 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 may switch to the master transmitter mode by loading SIDAT with SLA+W.

19.1.3. 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 will respond when addressed by a master. If the LSB (GC) is set, TWI 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 in the slave mode. ENSI must be set to “1” to enable TWI. The AA bit must be set to enable TWI 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 waits until it is addressed by its own slave address followed by the data direction bit which must be “1” (R) for TWI 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 is in the master mode (see state B0H).

If the AA bit is reset during a transfer, TWI will transmit the last byte of the transfer and enter state C0H or C8H. TWI 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 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 from the bus.

19.1.4. 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 waits until it is addressed by its own slave address followed by the data direction bit which must be “0” (W) for TWI 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 as a 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 is in the master mode (see status 68H and 78H).

If the AA bit is reset during a transfer, TWI will return a not acknowledge (logic 1) to SDA after the next received data byte. While AA is reset, TWI 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.

19.2. Miscellaneous States

There are two SISTA codes that do not correspond to a defined TWI 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 is not involved in a serial transfer.

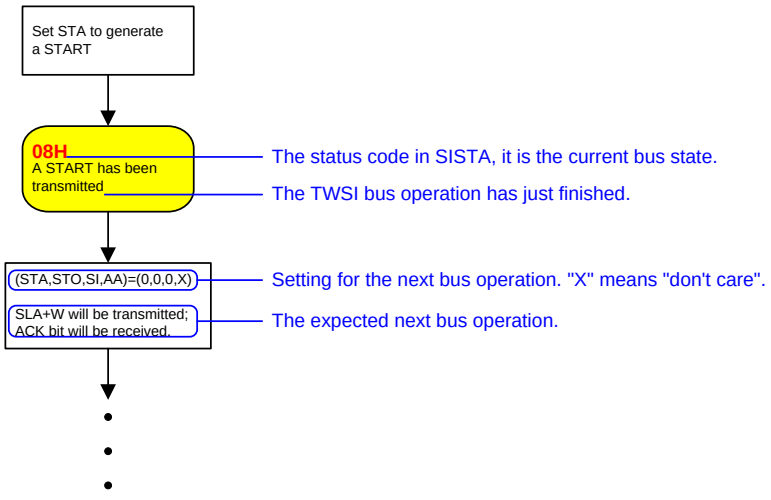
S1STA = 00H:
This status code indicates that a bus error has occurred during a TWI serial transfer. A bus error is caused when a START or STOP condition occurs at an illegal position in the format frame. Examples of such illegal positions are during the serial transfer of an address byte, a data byte, or an acknowledge bit. A bus error may also be caused when external interference disturbs the internal TWI 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 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).

19.3. Using the TWI

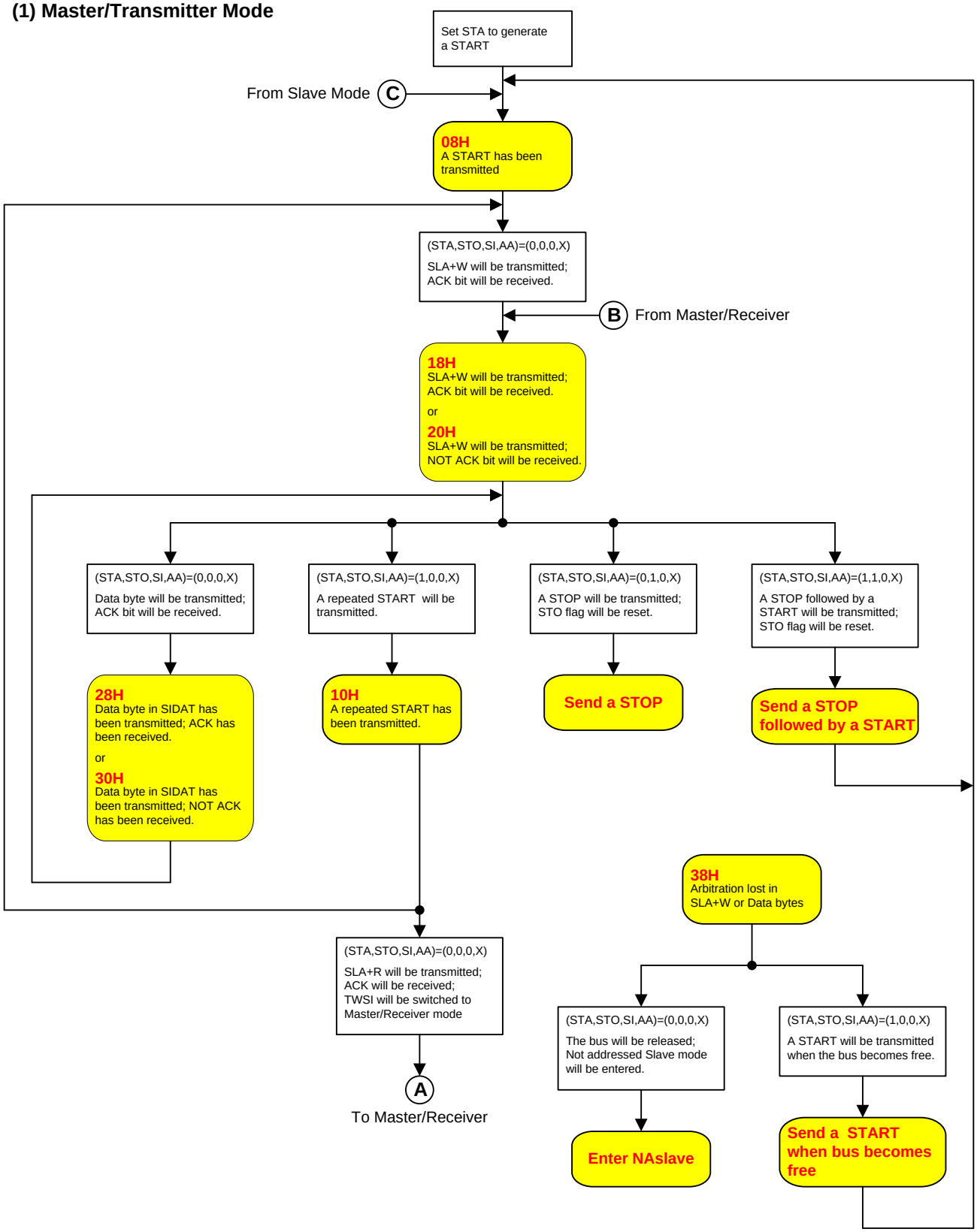
The TWI is byte-oriented and interrupt based. Interrupts are issued after all bus events, like reception of a byte or transmission of a START condition. Because the TWI is interrupt-based, the application software is free to carry on other operations during a TWI byte transfer. Note that the TWI0 interrupt enable bit ETWI0 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 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 bus. The application software can then decide how the TWI should behave in the next TWI 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 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 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 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”

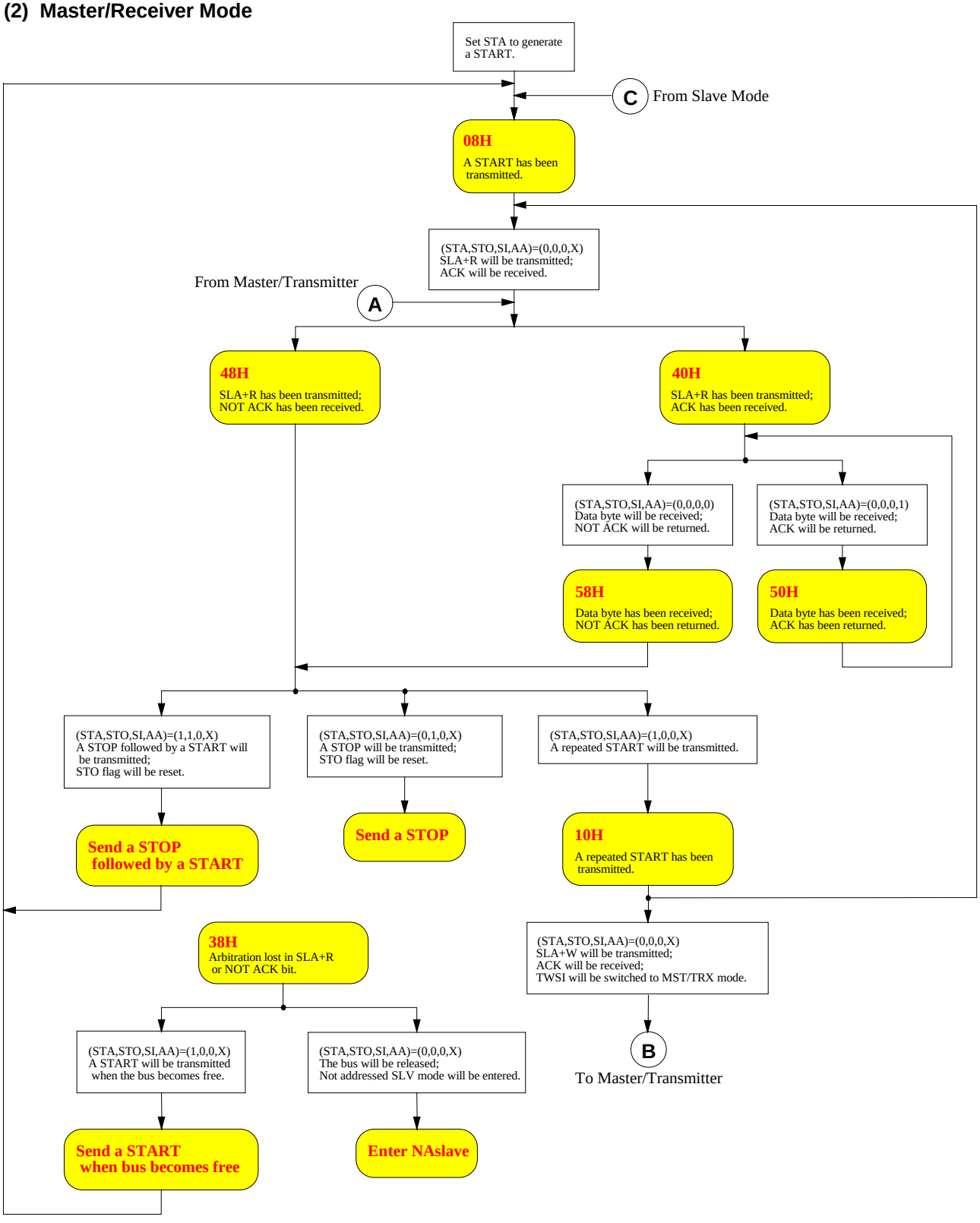
The figure below shows how to read the flow charts.



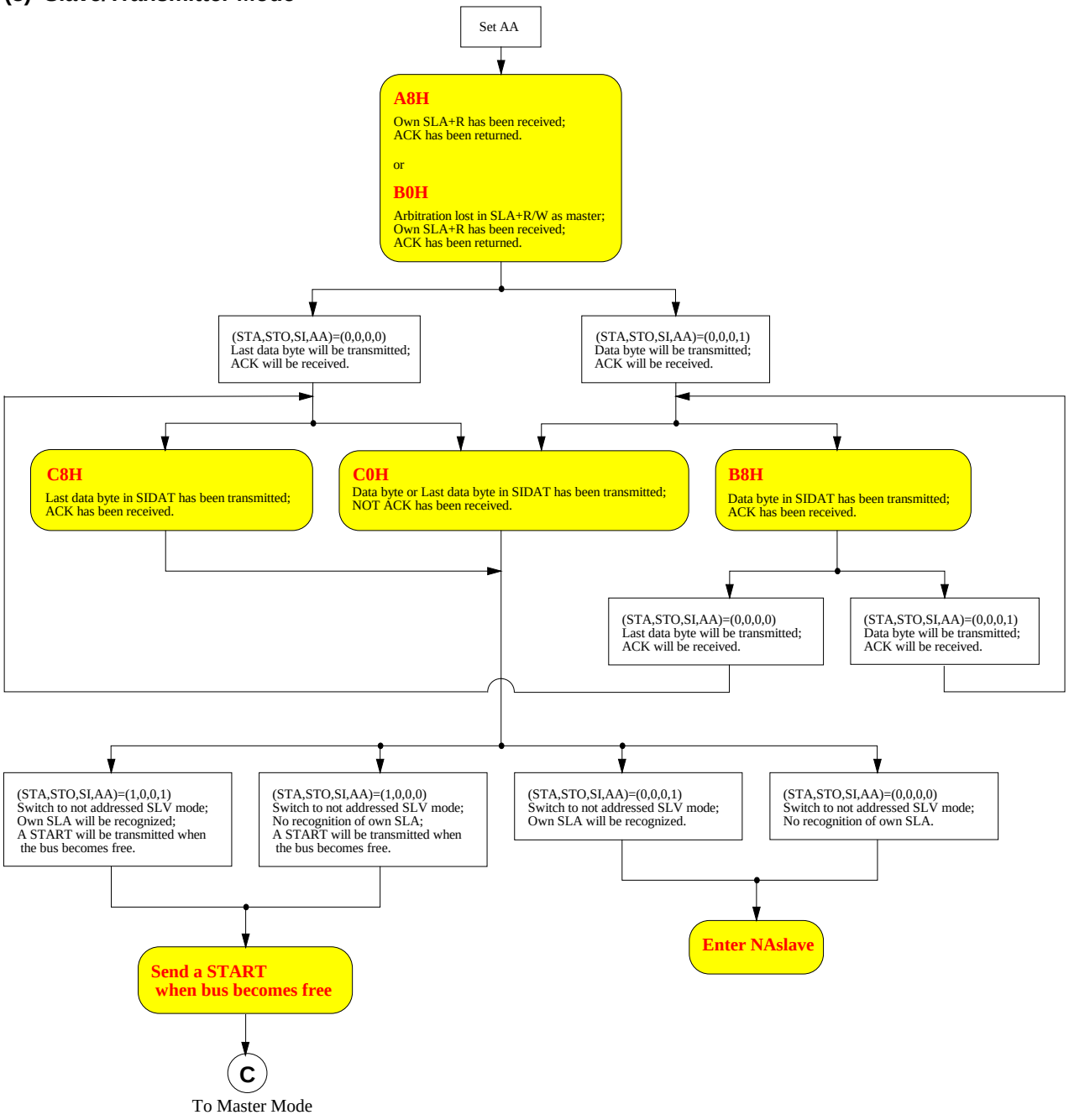
(1) Master/Transmitter Mode



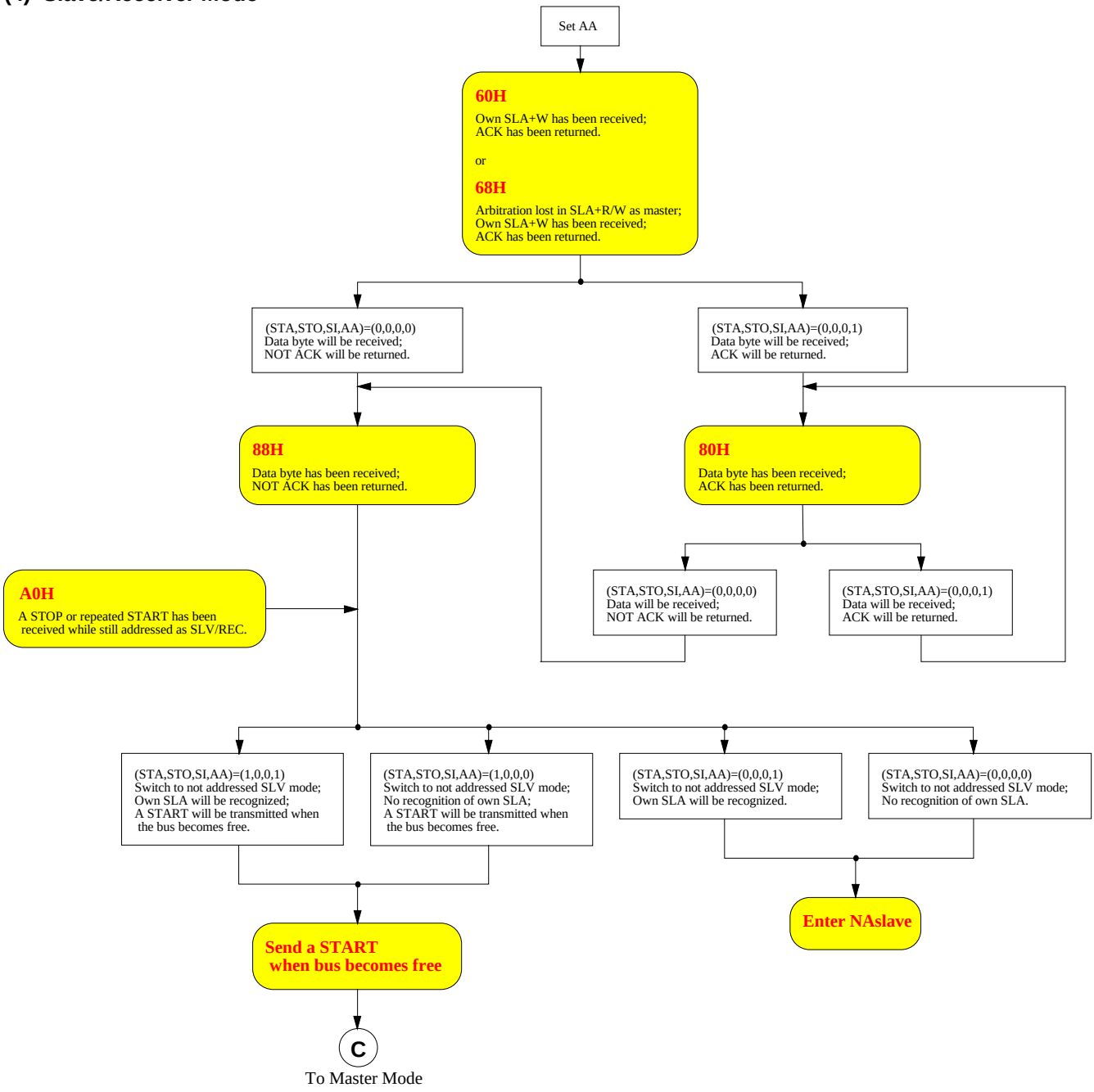
(2) Master/Receiver Mode



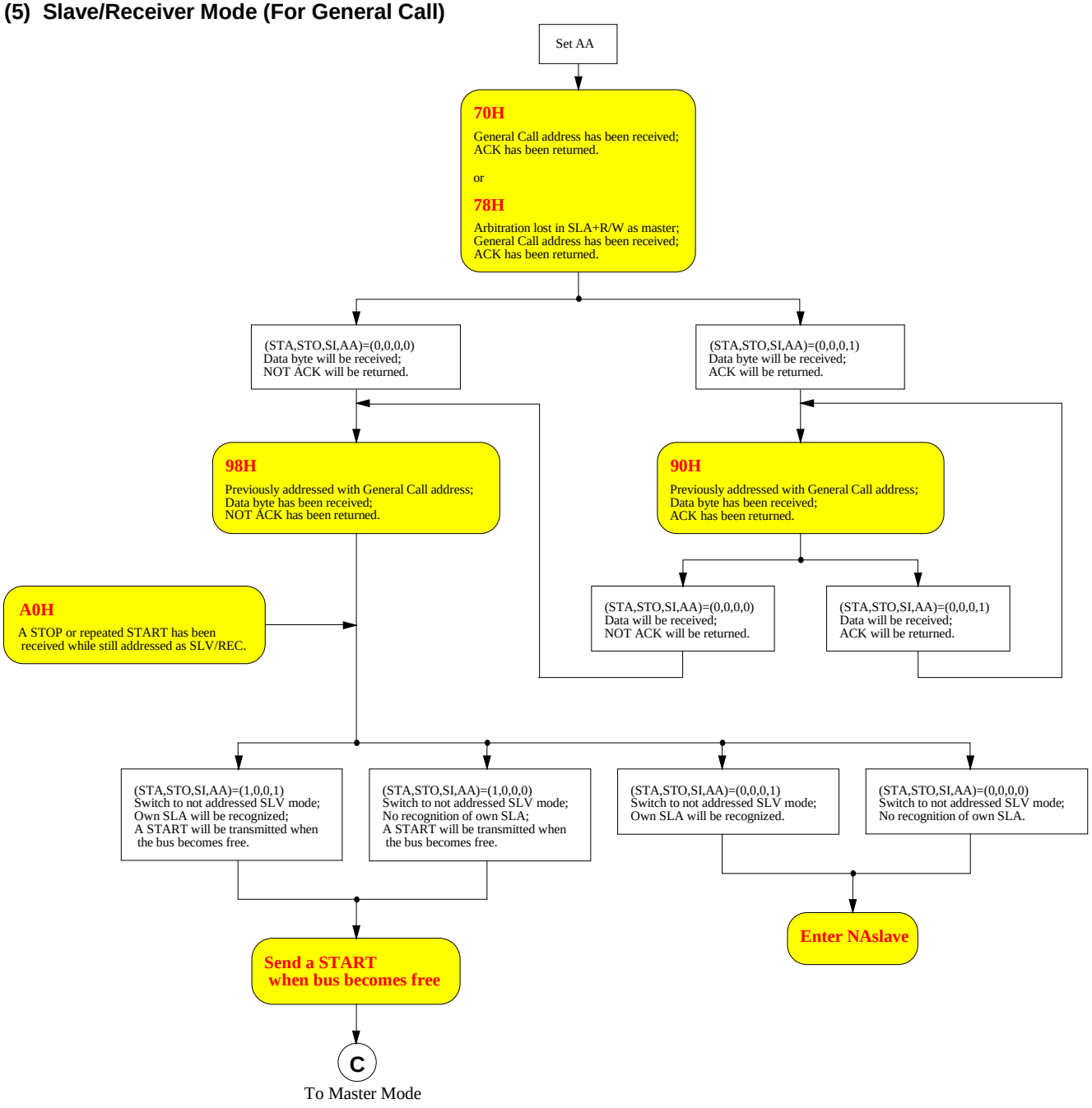
(3) Slave/Transmitter Mode



(4) Slave/Receiver Mode



(5) Slave/Receiver Mode (For General Call)



19.4. TWI0 Register

SIADR: TWI0 Address Register

SFR Page = 0~F
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

The CPU can read from and write to this register directly. SIADR is not affected by the TWI0 hardware. The contents of this register are irrelevant when TWI0 is in a master mode. In the slave mode, the seven most significant bits must be loaded with the microcontroller's own slave address, and, if the least significant bit (GC) is set, the general call address (00H) is recognized; otherwise it is ignored. The most significant bit corresponds to the first bit received from the TWI0 bus after a START condition.

SIDAT: TWI0 Data Register

SFR Page = 0~F
SFR Address = 0xD2 RESET= 0000-0000

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
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 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 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 Control Register

SFR Page = 0~F
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 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 Clock Rate select bit 2 (associated with CR1 and CR0).

Bit 6: ENSI, the TWI0 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 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 hardware will check the status of the serial bus. It will generate a START condition if the bus is free. Otherwise TWI0 will wait for a STOP condition and generates a START condition after a delay. If STA is set while TWI0 is already in a master mode and one or more bytes are transmitting or receiving, TWI0 will send a repeated START condition. STA may be set at any time. STA may also be set when TWI0 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 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 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 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 is in a master mode (in a slave mode, TWI0 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 state is present in the SISTA register, the SI flag is set by hardware. And, if the TWI0 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 is in the master/receiver mode.
3) A data byte has been received while TWI0 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 is in the master/receiver mode.
2) A data byte has been received while TWI0 is in the addressed slave/receiver mode.

Bit 7, 1~0: CR2, CR1 and CR0, the Clock Rate select Bits
These three bits determine the serial clock frequency when TWI0 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 will automatically synchronize with any clock frequency from master, which is up to 400KHz. The various serial clock rates are shown The various serial clock rates are shown in Table 19–1.

Table 19-1. TWI0 Serial Clock Rates

CR2	CR1	CR0	TWI0 Clock Selection	TWI0 Clock Rate @ SYSCLK=12MHz
0	0	0	SYSCLK/8	1.5 MHz Note1
0	0	1	SYSCLK/16	750 KHz
0	1	0	SYSCLK/32	375 KHz
0	1	1	SYSCLK/64	187.5 KHz
1	0	0	SYSCLK/128	93.75 KHz
1	0	1	SYSCLK/256	46.875 KHz
1	1	0	Reserved	--
1	1	1	T0OF/6	Variable

Note:

1. *The Maximum TWI0 clock Rate should under 1MHz, to set SYSCLK = 8MHz to generate 1MHz.*
2. *SYSCLK is the system clock.*
3. *T0OF is Timer 0 Overflow.*

SISTA: TWI0 Status Register

SFR Page = 0~F

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 states. When each of these states is entered, a status interrupt is requested (SI=1). A valid status code is present in SISTA when SI is set by hardware.

In addition, state 00H stands for a Bus Error. A Bus Error occurs when a START or STOP condition is present at an illegal position, such as inside an address/data byte or just on an acknowledge bit.

AUXR3: Auxiliary Register 3

SFR Page = **0 only**

SFR Address = 0xA4 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 Port Selection [1:0].

TWIPS1~0	TWI0_SCL	TWI0_SDA
00	P4.0	P4.1
01	P6.0	P6.1
10	P3.0	P3.1
11	P2.2	P2.4

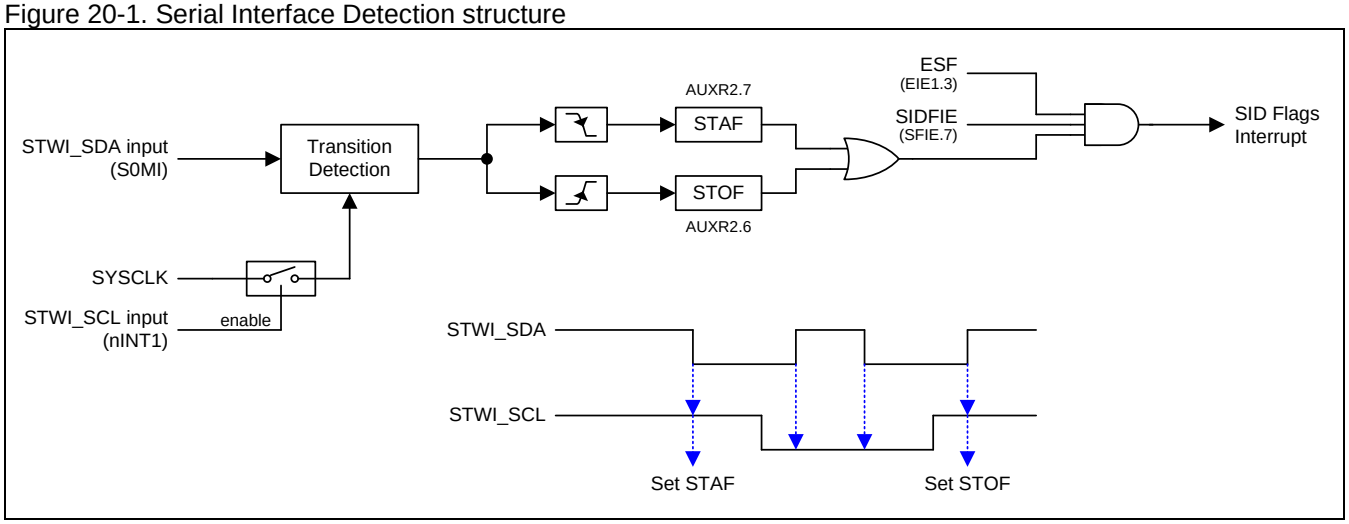
20. Serial Interface Detection (SID/STWI)

The serial interface detection module is always monitoring the “Start” and “Stop” condition on software two-wire-interface (STWI). STWI_SCL is the serial clock signal and STWI_SDA is the serial data signal. If any matched condition is detected, hardware set the flag on STAF and STOF. Software can poll these two flags or set SIDFIE (SFIE.7) to share the interrupt vector on System Flag. And STWI_SCL is located on nINT1 which helps MCU to strobe the serial data by nINT1 interrupt. Software can use these resources to implement a variable TWI slave device.

20.1. SID (STWI) Structure

Figure 20–1 shows the configuration of STAF and STOF detection, interrupt architecture and event detecting waveform.

Figure 20-1. Serial Interface Detection structure



20.2. SID/STWI Register

AUXR2: Auxiliary Register 2

SFR Page = 0~F

SFR Address = 0xA3 RESET = 00xx-0000

7	6	5	4	3	2	1	0
STAF	STOF	--	--	T1X12	T0X12	T1CKOE	T0CKOE
R/W	**R/W**	**W**	**W**	**W**	**W**	**W**	**W**

Bit 7: STAF, Start Flag detection of TWI2.
0: Clear by firmware by writing “0” on it.
1: Set by hardware to indicate the START condition occurred on STWI bus.

Bit 6: STOF, Stop Flag detection of TWI2.
0: Clear by firmware by writing “0” on it.
1: Set by hardware to indicate the START condition occurred on STWI bus.

SFIE: System Flag Interrupt Enable Register

SFR Page = 0~F

SFR Address = 0x8E RESET = 0110-x000

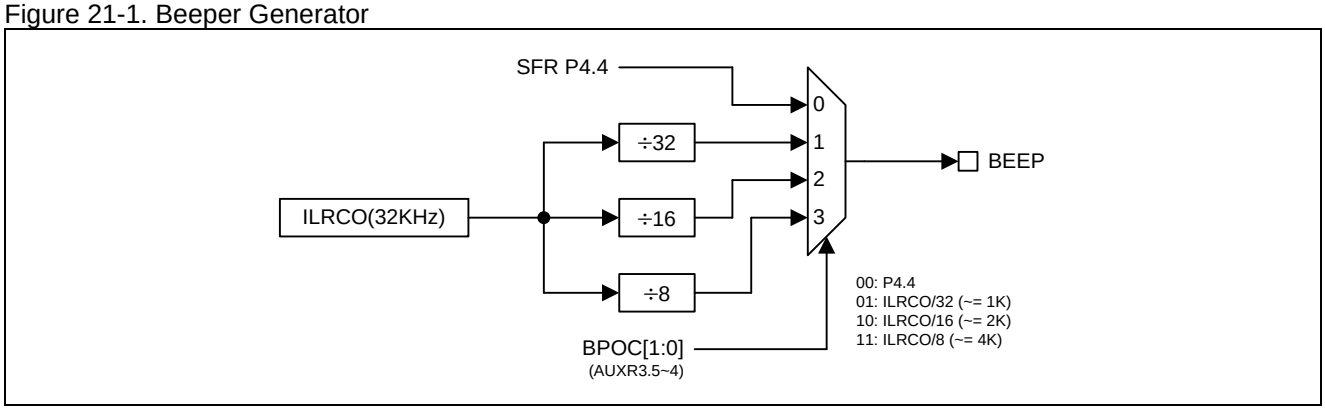
7	6	5	4	3	2	1	0
SIDFIE	MCDRE	MCDFIE	RTCFIE	--	BOF1IE	BOF0IE	WDTFIE
R/W	**R/W**	**R/W**	**R/W**	**W**	**R/W**	**R/W**	**R/W**

Bit 7: SIDFIE, Serial Interface (STWI) Detection Flag Interrupt Enabled.
0: Disable SID Flags (STAF or STOF) interrupt.
1: Enable SID Flags (STAF or STOF) interrupt.

21. 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 21–1 shows the beeper generator circuit. But ILRCO is not the precision clock source. Please refer Section "31.5 ILRCO Characteristics" for more detailed ILRCO frequency deviation range.

Figure 21-1. Beeper Generator



21.1. 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.

22. Keypad Interrupt (KBI)

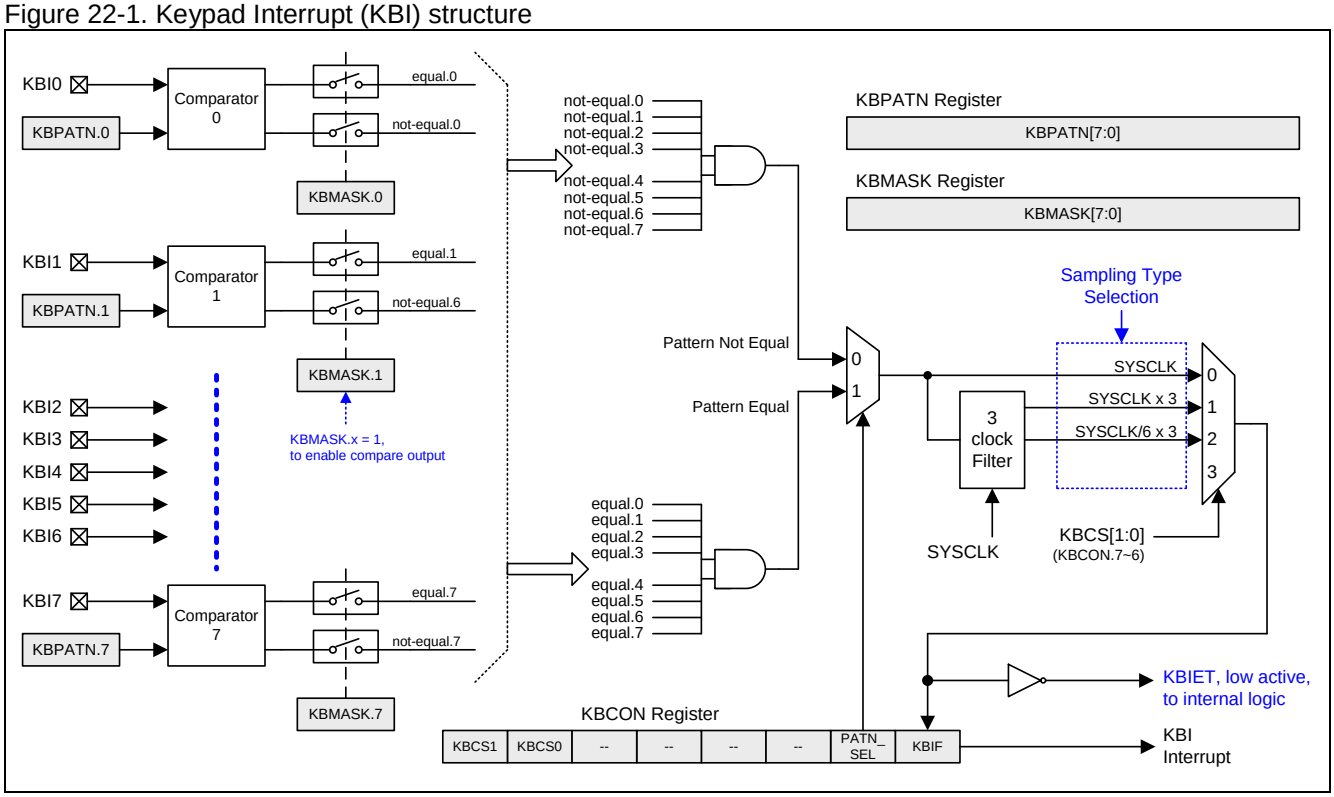
The Keypad Interrupt function is intended primarily to allow a single interrupt to be generated when KBI.7~0 is equal to or not equal to a certain pattern. This function can be used for bus address recognition or keypad recognition.

There are three SFRs used for this function. The Keypad Interrupt Mask Register (KBMASK) is used to define which input pins connected to Port 2 are enabled to trigger the interrupt. The Keypad Pattern Register (KBPATN) is used to define a pattern that is compared to the value of keypad input. The Keypad Interrupt Flag (KBIF) in the Keypad Interrupt Control Register (KBCON) is set by hardware when the condition is matched. An interrupt will be generated if it has been enabled by setting the EKBI bit in EIE1 register and EA=1. The PATN_SEL bit in the Keypad Interrupt Control Register (KBCON) is used to define “equal” or “not-equal” for the comparison. The keypad input can be assigned on the different port pins, please refer Section “4.3 Alternate Function Redirection” for more detailed information.

In order to use the Keypad Interrupt as the “Keyboard” Interrupt, the user needs to set KBPATN=0xFF and PATN_SEL=0 (not equal), then any key connected to keypad input which is enabled by KBMASK register will cause the hardware to set the interrupt flag KBIF and generate an interrupt if it has been enabled. The interrupt may wake up the CPU from Idle mode or Power-Down mode. This feature is particularly useful in handheld, battery powered systems that need to carefully manage power consumption but also need to be convenient to use.

22.1. KBI Structure

Figure 22-1. Keypad Interrupt (KBI) structure



22.2. 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= XXXX-XX01

7	6	5	4	3	2	1	0
KBCS1	KBCS0	0	0	0	0	PATN_SEL	KBIF
R/W	**R/W**	**W**	**W**	**W**	**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	Reserved

Bit 7~2: Reserved. Software must write “0” on these bits when KBCON is written.

Bit 1: PATN_SEL, Pattern Matching Polarity selection.
0: The keypad input has to be not equal to the 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 = 0000-0x00

7	6	5	4	3	2	1	0
KBI4PS1	KB4IPS0	KBI6PS0	KBI2PS0	KBI0PS0	--	S0MIPS	S0COPS
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**W**	**R/W**	**R/W**

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

KBI4PS1~0	KBI4	KBI5
00	P1.4	P1.5
01	P3.4	P3.5
10	P6.0	P6.1
11	P2.0	P2.1

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

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

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

KBI2PS0	KBI2	KBI3
0	P1.2	P1.3
1	P2.2	P2.4

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

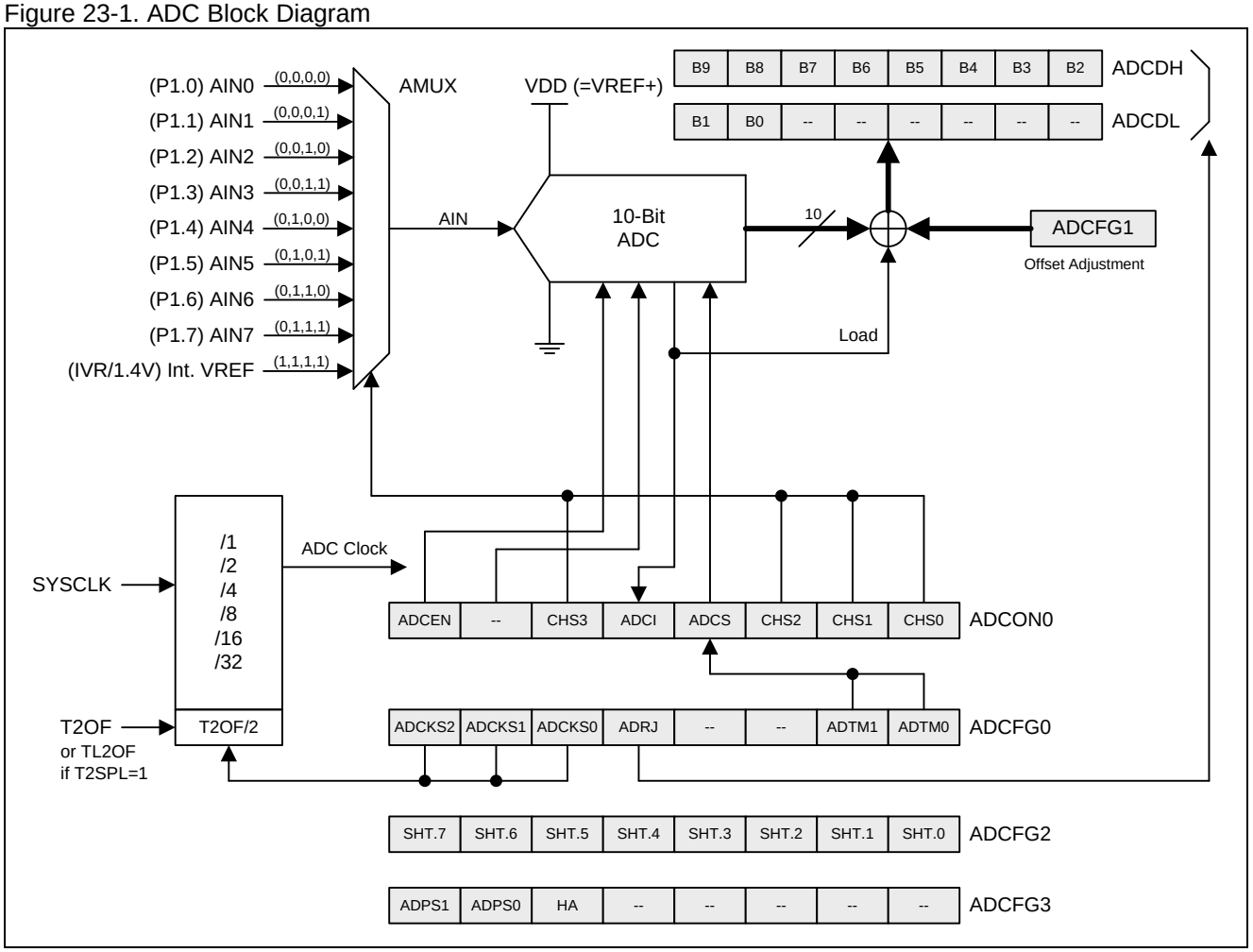
KBI0PS	KBI0	KBI1
0	P1.0	P1.1
1	P4.0	P4.1

23. 10-Bit ADC

The ADC subsystem for the **MG82FG5D16** consists of an analog multiplexer (AMUX), and a **1M** sps, 10-bit successive-approximation-register ADC. The AMUX can be configured via the Special Function Registers shown in Figure 23–1. ADC operates in Single-ended mode, and may be configured to measure any of the pins on Port 1 or internal reference. The ADC subsystem is enabled only when the ADEN bit in the ADC Control register (ADCON0) is set to logic 1. The ADC subsystem is in low power shutdown when this bit is logic 0.

23.1. ADC Structure

Figure 23-1. ADC Block Diagram



23.2. ADC Operation

ADC has a maximum conversion speed of **1M** sps. The ADC conversion clock is a divided version of the system clock 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

$$\text{ADC Result} = \frac{V_{IN} * 1024}{\text{VDD(=VREF+) Voltage}}$$

23.2.1. ADC Input Channels

The analog multiplexer (AMUX) selects the inputs to the ADC, allowing any of the pins on Port 1 to be measured in single-ended mode and one internal voltage reference (IVR, 1.4V). The ADC input channels are configured and selected by CHS3~0 in the ADCON0 register as shown in Figure 23–1. The selected pin is measured with respect to GND.

23.2.2. ADC Internal Voltage Reference

The default ADC reference is VDD. If the VDD is not fixed at a certain voltage, then use the following steps to read voltage:

1) To set the analog multiplexer (AMUX) to IVR.
2) Convert and store the IVR value by ADC. (Hint: Different VDD voltage will get different IVR read back value, but IVR is fixed at 1.4V. So this read back value can be tread as the reference value.)
3) To use the IVR read back reference value to calculate the VDD value. Now the VDD get a certain value, and can be treated as the reference voltage.
4) To use the reference voltage convert the input voltage.

23.2.3. Starting a Conversion

Prior to using the ADC function, the user should:

1) Turn on the ADC hardware by setting the ADCEN bit,
2) Configure the ADC input clock by bits ADCKS2, ADCKS1 and ADCKS0
3) Select the analog input channel by bits CHS3, CHS2, CHS1 and CHS0
4) Configure the selected input pin (shared with P1) to the Analog-Input-Only mode, and
5) Configure ADC result arrangement using ADRJ bit.

Now, user can set the ADCS bit to start the A-to-D conversion. The conversion time is controlled by the bits HA, SHT[7:0], 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 **10** 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 Timer2 over flow or free-run, then the ADC will keep conversion continuously unless ADCEN is cleared or configure ADC to manual mode.

As described above, the interrupt flag ADCI, when set by hardware, shows a completed conversion. Thus two ways may be used to check if the conversion is completed: (1) Always polling the interrupt flag ADCI by software; (2) Enable the ADC interrupt by setting bits EADC (in EIE1 register) and EA (in IE register), and then the CPU will jump into its Interrupt Service Routine when the conversion is completed. Regardless of (1) or (2), the ADCI flag should be cleared by software before next conversion.

23.2.4. ADC Conversion Rate

The user can select the appropriate conversion speed according to the frequency of the analog input signal. The maximum input clock of the ADC is **24MHz** and it operates a minimum conversion time with **24** ADC clocks. User can configure the ADCKS2~0 (ADCFG0.7~5), SHT (ADCFG2.7~0) and HA (ADCFG3.5) to specify the conversion rate. The following equation is the clock number of one ADC conversion:

$$\text{ADC Conversion Rate} = \frac{\text{ADC Clock Freq.}}{(24 + X + Y*6)}$$

; X = SHT, 0~255
; Y= HA, 0~1

For example,

1. To get 1MHz Sample Rate:
   If SYSCLK= 48MHz and the ADCKS = SYSCLK/2 is selected, SFT = 0, HA = 0, then the frequency of the analog input should be no more than 500KHz to maintain the conversion accuracy. (Conversion rate = 48MHz/2/(24+0+0) = 1M Hz.)
2. To get 666KHz Sample Rate:
   if SYSCLK= 48MHz and the ADCKS = SYSCLK/2 is selected, SFT = 0, HA = 2, then the frequency of the analog input should be no more than 333KHz to maintain the conversion accuracy. (Conversion rate = 48MHz/2/(24+0+12) = 666KHz.)

23.2.5. 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 disable its digital output. It is done by putting the port pin into the analog-input-only mode. The port pin configuration for analog input function is described in Table 13–3. General Port Configuration Settings and Section “13.2.1 Port 1 Register”.

23.2.6. 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.

23.3. ADC Register

ADCON0: ADC Control Register 0

SFR Page = 0~F

SFR Address = **0xC4** RESET = 0x00-0000

7	6	5	4	3	2	1	0
ADCEN	0	CHS3	ADCI	ADCS	CHS2	CHS1	CHS0
R/W	**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: Reserved. Software must write “0” on this bit when ADCON0 is written.

Bit 5: CHS3. Combined CH2~0 to select ADC input channel.

Bit 4: ADCI, ADC Interrupt Flag.
0: The flag must be cleared by software.
1: This flag is set when an A/D conversion is completed. An interrupt is invoked if it is enabled.

Bit 3: ADCS. ADC Start of conversion.
0: ADCS cannot be cleared by software.
1: Setting this bit by software starts an A/D conversion. On completion of the conversion, the ADC hardware will clear ADCS and set the ADCI. A new conversion may not be started while either ADCS or ADCI is high.

Bit 2~0: CHS2 ~ CHS1, Input Channel Selection for ADC analog multiplexer.

In Single-ended mode:

CHS3~0	Selected Channel
0 0 0 0	AIN0 (P1.0)
0 0 0 1	AIN1 (P1.1)
0 0 1 0	AIN2 (P1.2)
0 0 1 1	AIN3 (P1.3)
0 1 0 0	AIN4 (P1.4)
0 1 0 1	AIN5 (P1.5)
0 1 1 0	AIN6 (P1.6)
0 1 1 1	AIN7 (P1.7)
1 1 1 1	Int. VREF (IVR/1.4V)
Others	Reserved

ADCFG0: ADC Configuration Register 0

SFR Page = **0 Only**

SFR Address = **0xC3** RESET = 0000-xx00

7	6	5	4	3	2	1	0
ADCKS2	ADCKS1	ADCKS0	ADRJ	--	--	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[1:0]	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	Reserved
1 1 1	T2OF/2

Note:

1. *SYSCLK is the system clock.*
2. *T2OF is Timer2 Overflow.*

Bit 4: ADRJ, ADC result Right-Justified selection.

0: The most significant 8 bits of conversion result are saved in ADCDH[7:0], while the least significant 2 bits in ADCDL[7:6].

1: The most significant 2 bits of conversion result are saved in ADCDH[1:0], while the least significant 8 bits in ADCDL[7:0].

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
(B9)	(B8)	(B7)	(B6)	(B5)	(B4)	(B3)	(B2)
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
(B1)	(B0)	--	--	--	--	--	--
R	**R**	**R**	**R**	**R**	**R**	**R**	**R**

If ADRJ = 1

ADCDH

7	6	5	4	3	2	1	0
--	--	--	--	--	--	(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 10-bit unsigned integers. Inputs are measured from ‘0’ to VREF x 1023/1024. 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(or VREF+) x 1023/1024	0xFFC0	0x03FF
VDD(or VREF+) x 512/1024	0x8000	0x0200
VDD(or VREF+) x 256/1024	0x4000	0x0100
VDD(or VREF+) x 128/1024	0x2000	0x0080
0	0x0000	0x0000

Bit 3~2: Reserved. Software must write "0" on these bits when ADCFG0 is written.

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	Reserved

ADCFG1: ADC Configuration Register 1

SFR Page = **1 Only**

SFR Address = **0xC3** RESET = xxx0-0000

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

Bit 7~5: Reserved. Software must write "0" on these bits when ADCFG1 is written.

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 by setting ADCON0.AZEN and update the offset value to AD0ROC for an auto-cancellation on ADC transfer result. Software can also stores the value in **MG82FG5D16** IAP zone to use it as a constant parameter for ADC offset cancellation. The following table lists the AD0ROC adjustment value for ADC transfer result.

{Sign, AOS.[3:0]}	Value in {ADCDH, ADCDL}
0_1111	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 = 010x-xxxx

7	6	5	4	3	2	1	0
ADPS1	ADPS0	HA	0	0	0	0	0
R/W	**R/W**	**R/W**	**W**	**W**	**W**	**W**	**W**

Bit 7~6: ADC Power Saving Mode selection bit 1~0.

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: HA, extend 6 ADC clock for ADC conversion.

Bit 4~0: Reserved. Software must write “0” on these bits when ADCFG3 is written.

PCON3: Power Control Register 3

SFR Page = **P Only**

SFR Address = 0x45 POR = 0000-0000

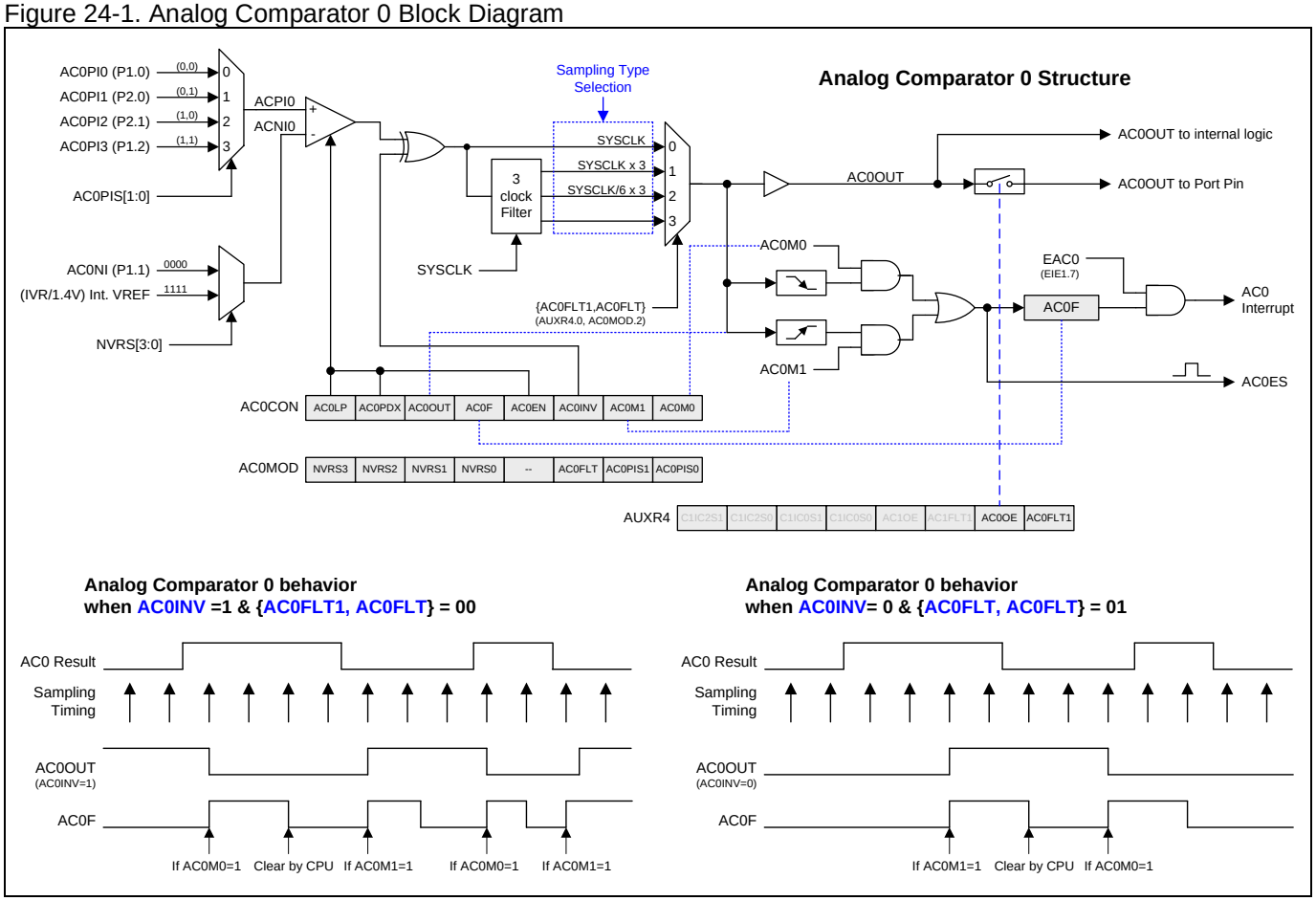
7	6	5	4	3	2	1	0
IVREN	--	--	--	LDOC1	LDOC0	FLDO	LDOL
R/W	**W**	**W**	**W**	**R/W**	**R/W**	**R/W**	**R/W**

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

24. Analog Comparator 0 (AC0)

24.1. AC0 Structure

Figure 24-1. Analog Comparator 0 Block Diagram



24.2. AC0 Register

AC0CON: Analog Comparator 0 Control & Status Register

SFR Page = 0~F

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 with level input (low level or high level).

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 Mode.

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~F

SFR Address = 0x9F RESET = 0000-x000

7	6	5	4	3	2	1	0
NVRS3	NVRS2	NVRS1	NVRS0	--	AC0FLT	AC0PIS1	AC0PIS0
W	**W**	**W**	**R/W**	**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:

NVRS[3:0]	(V-) Input
0000	AC0NI(P1.1)
1111	Int. VREF (1.4V)

Bit 2: AC0FLT, Analog Comparator 0 output Filter control. It selects AC0OUT filter mode with AC0FLT1 (AUXR4.0)

AC0FLT1, AC0FLT	AC0OUT filter mode
0 0	Disabled
0 1	SYSCLK x 3
1 0	SYSCLK/6 x 3
1 1	Reserved

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(P1.0)
0 1	AC0PI1(P2.0)
1 0	AC0PI2(P2.1)
1 1	AC0PI3(P1.2)

AUXR4: Auxiliary Register 4

SFR Page = 1 only

SFR Address = 0xA4 RESET = 0000-0x00

7	6	5	4	3	2	1	0
T2PS1	T2PS0	T1PS1	T1PS0	SPIPS0	--	AC0OE	AC0FLT1
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**W**	**R/W**	**R/W**

Bit 1: AC0OE, AC0OUT output enable on port pin.
0: Disable AC0OUT output on port pin.
1: Enable AC0OUT output on P6.1.

PCON3: Power Control Register 3

SFR Page = **P Only**

SFR Address = 0x45 POR = 0xxx-xxxx

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

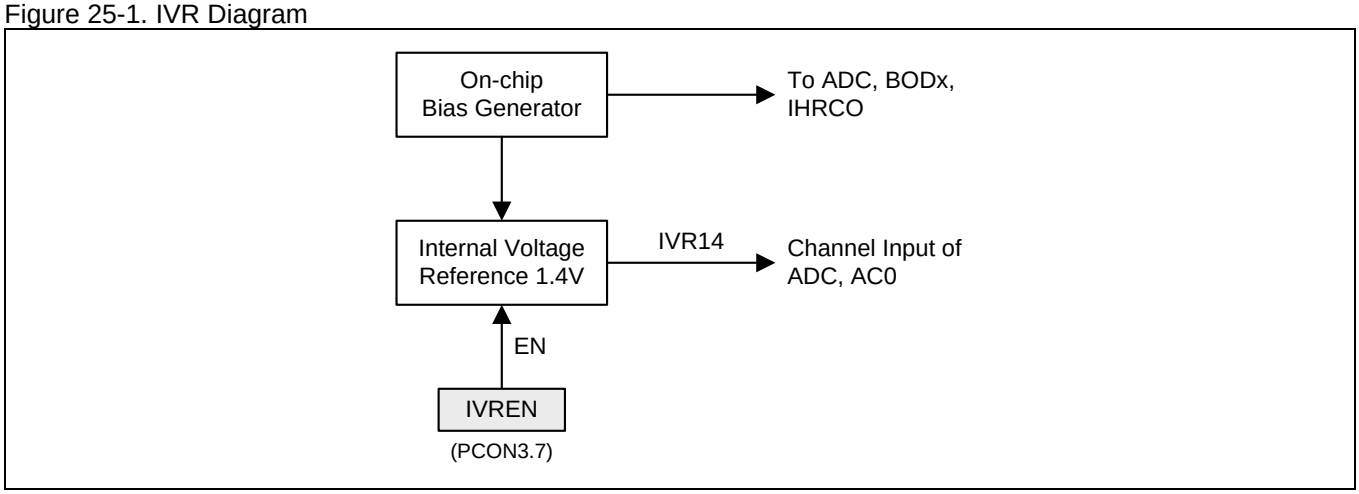
Bit 7: IVREN, Internal Voltage Reference Enable.
0: Disable on-chip Int. VREF (IVR/1.4V).
1: Enable on-chip Int. VREF (IVR/1.4V).

25. Internal Voltage Reference (IVR, 1.4V)

The IVR can be used as the reference voltage of the AC0 and ADC. The typical output is 1.4V. It can be disable by IVREN.

25.1. IVR (1.4V) Structure

Figure 25-1. IVR Diagram



25.2. IVR Register

PCON3: Power Control Register 3

SFR Page = **P Only**

SFR Address = 0x45 POR = 0xxx-xxxxx

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

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

26. ISP and IAP

The flash memory of **MG82FG5D16** 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.

26.1. MG82FG5D16 Flash Memory Configuration

There are total **16K** bytes of Flash Memory in **MG82FG5D16** and Figure 26–1 shows the device flash configuration of **MG82FG5D16**. The ISP-memory can be configured as disabled or up to 7.5K bytes space by hardware option setting with 0.5KB step. The flash size of IAP memory is located between the IAP low boundary and IAP high boundary. The IAP low boundary is defined by the value of IAPLB register. The IAP high boundary is associated with ISP start address which decides ISP memory size by hardware option. The IAPLB register value is configured by hardware option or AP software programming. All of the AP, IAP and ISP memory are shared the total **16K** bytes flash memory.

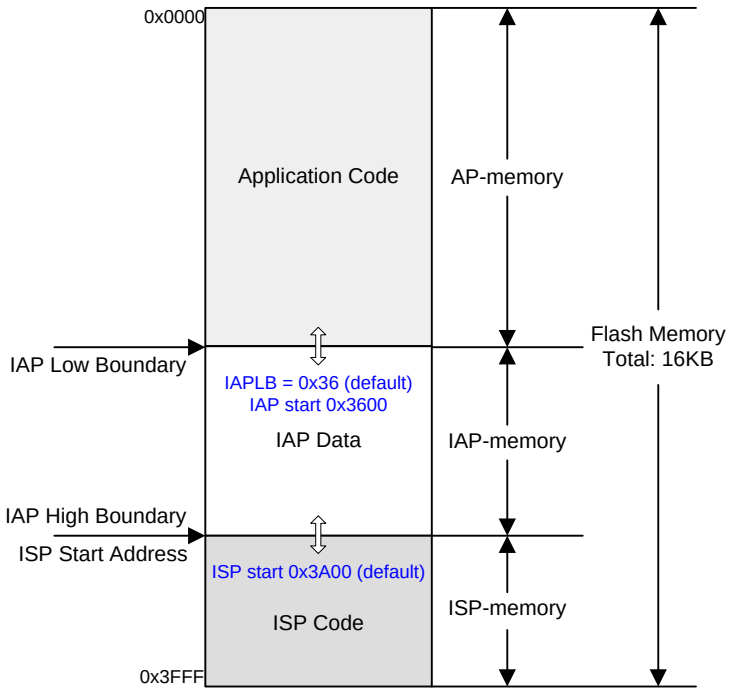
Figure 26-1. MG82FG5D16 Flash Memory Configuration

Note:
(1) ISP Start Address:
0x2200 if ISP Size = 7.5KB
0x2400 if ISP Size = 7KB
……
0x3000 if ISP Size = 4KB
……
0x3C00 if ISP Size = 1KB
0x3E00 if ISP Size = 0.5KB

(2) IAP Size :
IAPLB = IAP Low Boundary (ROM High-Byte address)
IAP Start Address = {IAPLB, 00H}
IAP Size = ISP Start Address – IAP Start Address
Set LAPLB = Change IAP Size

(3) If ISP is enabled:
IAP High Boundary = ISP Start Address – 1
IAP Low Boundary = ISP Start Address – IAP Size

(4) If ISP is disabled:
IAP High Boundary = 0x3FFF
IAP Low Boundary = 0x3FFF – IAP Size + 1



Note:

*In default, the **MG82FG5D16** that Megawin shipped had configured the flash memory for **1.5K ISP, 1.0K 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 **1.0K** IAP size can be re-configured by software for application required.*

26.2. MG82FG5D16 Flash Access in ISP/IAP

There are 3 flash access modes are provided in **MG82FG5D16** 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.

Before perform ISP/IAP operation, the user should fill the bits XCKS5~XCKS0 in CKCON1 register with a proper value. (Refer to Section "8.7 Clock Register")

To do Page Erase (512 Bytes per Page)

Step 1: Set MS[2:0]=[0,1,1] in ISPCR register to select Page Erase Mode.
Step 2: Fill page address in IFADRH & IFADRL registers.
Step 3: Sequentially write 0x46h then 0xB9h to SCMD register to trigger an ISP processing.

To do Byte Program

Step 1: Set MS[2:0]=[0,1,0] in ISPCR register to select Byte Program Mode.
Step 2: Fill byte address in IFADRH & IFADRL registers.
Step 3: Fill data to be programmed in IFD register.
Step 4: Sequentially write 0x46h then 0xB9h to SCMD register to trigger an ISP processing.

To do Read

Step 1: Set MS[2:0]=[0,0,1] in ISPCR register to select Read Mode.
Step 2: Fill byte address in IFADRH & IFADRL registers.
Step 3: Sequentially write 0x46h then 0xB9h to SCMD register to trigger an ISP processing.
Step 4: Now, the Flash data is in IFD register.

The detailed descriptions of flash page erase, byte program and flash read in **MG82FG5D16** is listed in the following sections:

26.2.1. ISP/IAP Flash Page Erase Mode

The any bit in flash data of **MG82FG5D16** 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 **MG82FG5D16** 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 **MG82FG5D16** 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 26–3 shows the flash page erase flow in ISPIAP operation.

Figure 26-2. ISP/IAP Page Erase Flow

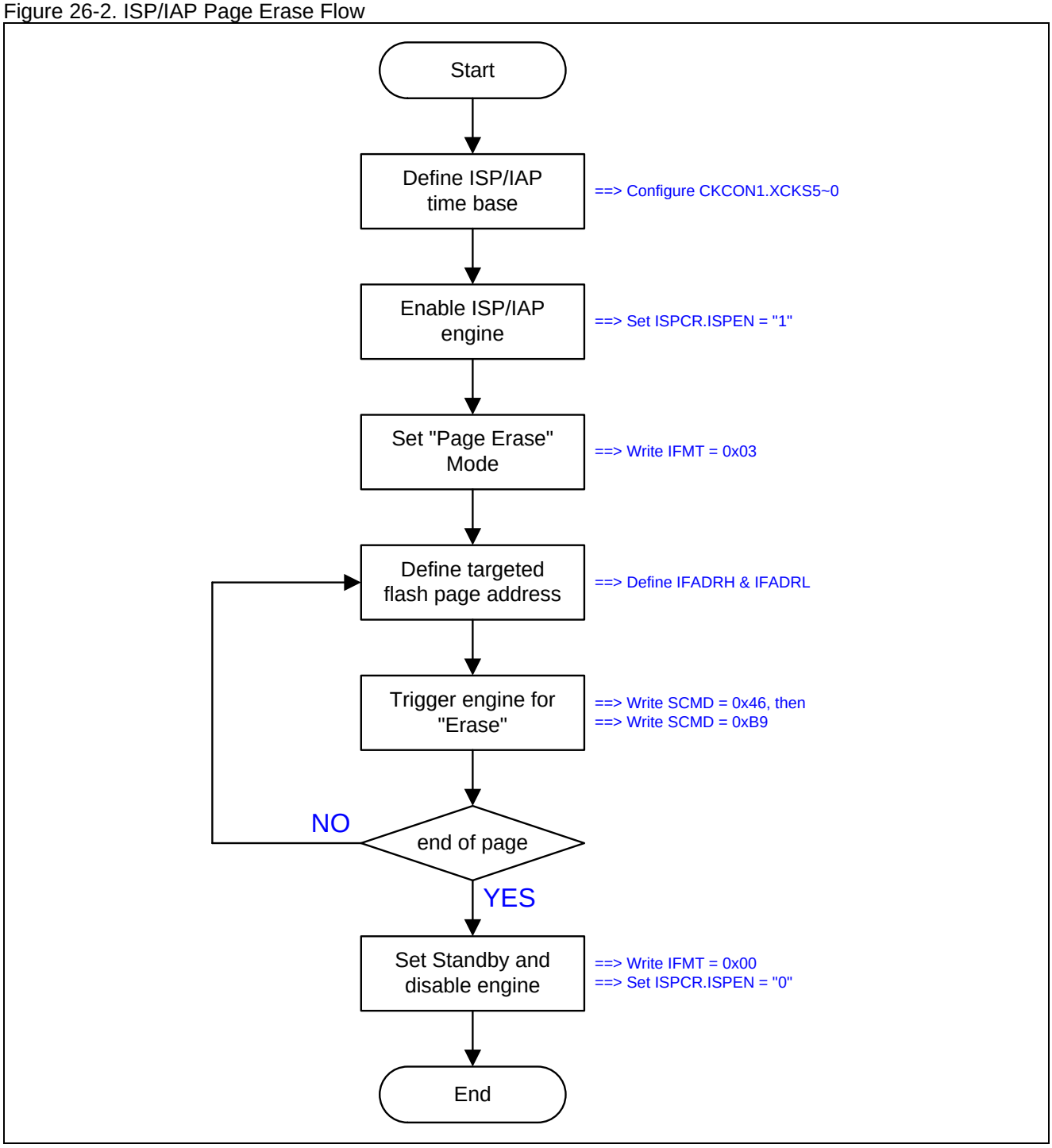


Figure 26–4 shows the demo code of the ISP/IAP page erase operation.

Figure 26-3. Demo Code for ISP/IAP Page Erase

```
MOV    ISPCR,#00010111b ; XCKS4~0 = decimal 23 when OSCin = 24MHz

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
```

26.2.2. ISP/IAP Flash Byte Program Mode

The “program” mode of **MG82FG5D16** 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 26–5 shows the flash byte program flow in ISP/IAP operation.

Figure 26-4. ISP/IAP byte Program Flow

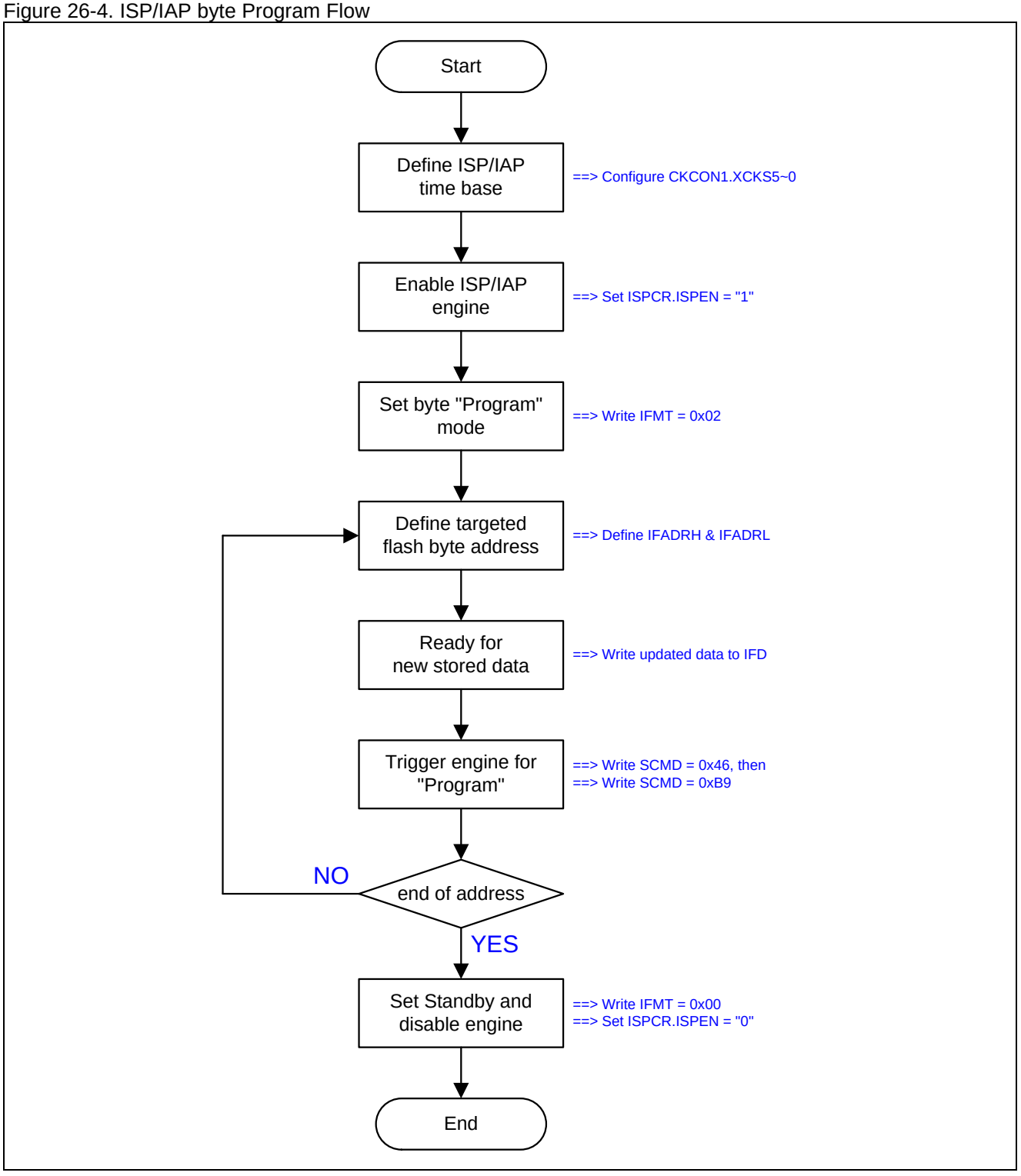


Figure 26–6 shows the demo code of the ISP/IAP byte program operation.

Figure 26-5. Demo Code for ISP/IAP byte Program

```
MOV    ISPCR,#00010111b ; XCKS4~0 = decimal 23 when OSCin = 24MHz

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
```

26.2.3. ISP/IAP Flash Read Mode

The “read” mode of **MG82FG5D16** 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 26–7 shows the flash byte read flow in ISP/IAP operation.

Figure 26-6. ISP/IAP byte Read Flow

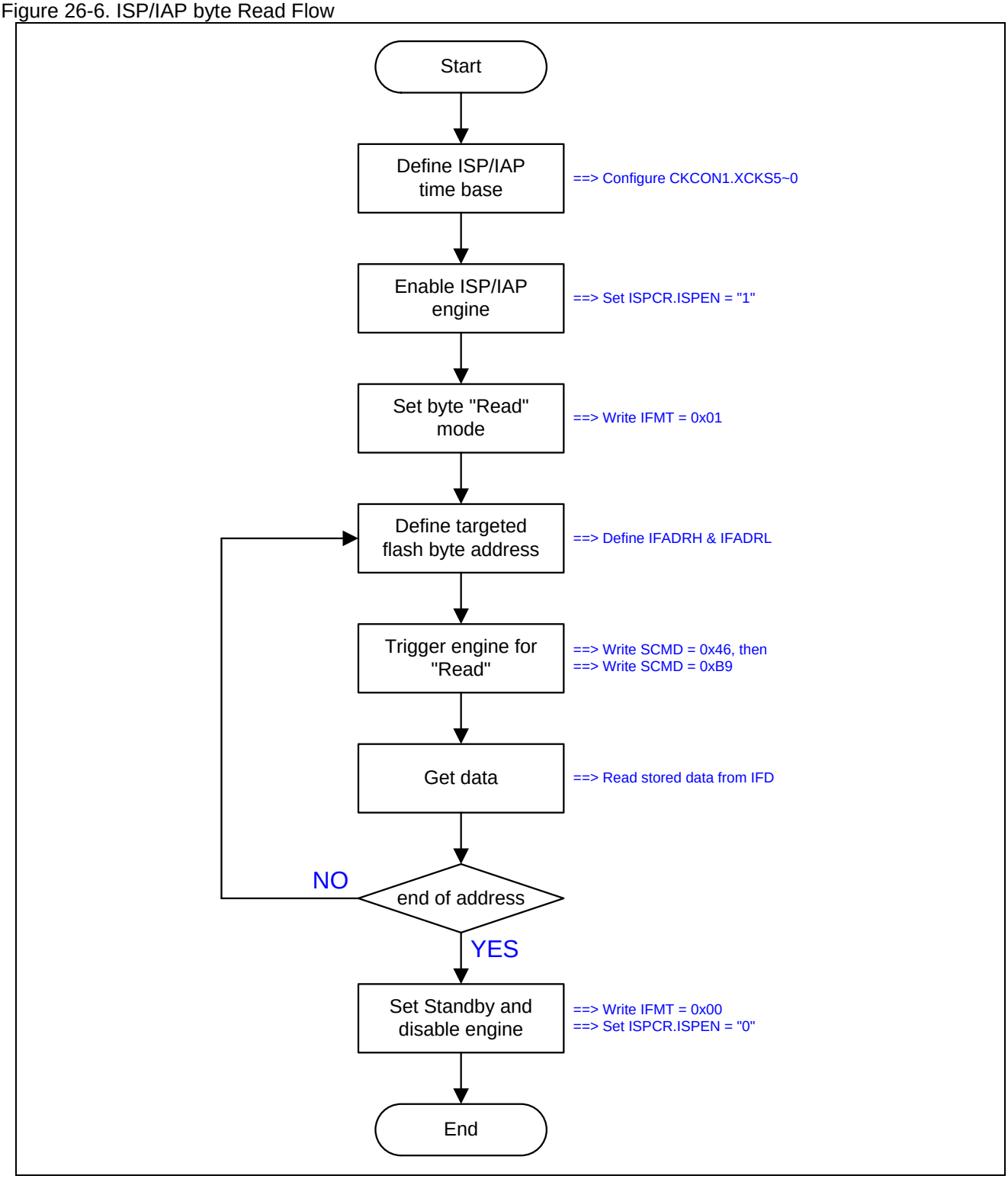


Figure 26–8 shows the demo code of the ISP/IAP byte read operation.

Figure 26-7. Demo Code for ISP/IAP byte Read

```
MOV    ISPCR,#00010111b ; XCKS4~0 = decimal 23 when OSCin = 24MHz

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
```

26.3. ISP Operation

ISP means In-System-Programming which makes it possible to update the user's application program (in AP-memory) and non-volatile application data (in IAP-memory) without removing the MCU chip from the actual end product. This useful capability makes a wide range of field-update applications possible. The ISP mode is used in the *loader program* to program both the AP-memory and IAP-memory.

Note:

(1) Before using the ISP feature, the user should configure an ISP-memory space and pre-program the ISP code (boot loader program) into the ISP-memory by a universal Writer/Programmer or Megawin proprietary Writer/Programmer.

(2) ISP code in the ISP-memory can only program the AP-memory and IAP-memory.

After ISP operation has been finished, software writes "001" on ISPCR.7 ~ ISPCR.5 which triggers an software RESET and makes CPU reboot into application program memory (AP-memory) on the address 0x0000.

As we have known, the purpose of the ISP code is to program both AP-memory and IAP-memory. Therefore, ***the MCU must boot from the ISP-memory in order to execute the ISP code***. There are two methods to implement In-System Programming according to how the MCU boots from the ISP-memory.

26.3.1. Hardware approached ISP

To make the MCU directly boot from the ISP-memory when it is just powered on, the MCU's hardware options *HWBS* and *ISP Memory* must be enabled. The ISP entrance method by hardware option is named hardware approached. Once *HWBS* and *ISP Memory* are enabled, the MCU will always boot from the ISP-memory to execute the ISP code (boot loader program) when it is just powered on. The first thing the ISP code should do is to check if there is an ISP request. If there is no ISP requested, the ISP code should trigger a software reset (setting ISPCR.7~5 to "101" simultaneously) to make the MCU re-boot from the AP-memory to run the user's application program..

If the additional hardware option, HWBS2, is enabled with HWBS and ISP Memory, the MCU will always boot from ISP memory after power-on or **external reset finished**. It provides another hardware approached to enter ISP mode by external reset signal. After first time power-on, **MG82FG5D16** can perform ISP operation by external reset trigger and doesn't wait for next time power-on, which suits the non-power-off system to apply the hardware approached ISP function.

26.3.2. 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.

26.3.3. 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 26–1 for **MG82FG5D16**), it doesn't mean you need to put this offset (= *ISP Start Address*) in your source code. The code offset is automatically manipulated by the hardware. User just needs to develop it like an application program in the AP-memory.

Interrupts during ISP

After triggering the ISP/IAP flash processing, the MCU will halt for a while for internal ISP processing until the processing is completed. At this time, the interrupt will queue up for being serviced if the interrupt is enabled previously. Once the processing is completed, the MCU continues running and the interrupts in the queue will be serviced immediately if the interrupt flag is still active. The user, however, should be aware of the following:

(1) Any interrupt can not be in-time serviced when the MCU halts for ISP processing.
(2) The low/high-level triggered external interrupts, nINTx, should keep activated until the ISP is completed, or they will be neglected.

ISP and Idle mode

MG82FG5D16 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 the last byte of the IAP-memory, the hardware will automatically neglect the triggering of ISP processing. That 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.

26.4. In-Application-Programming (IAP)

The **MG82FG5D16** 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 **MG82FG5D16** 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.*

26.4.1. IAP-memory Boundary/Range

If ISP-memory is specified, the range of the IAP-memory is determined by IAP and the ISP starts address as listed below.

IAP high boundary = ISP start address –1.
IAP low boundary = ISP start address - IAP.

If ISP-memory is not specified, the range of the IAP-memory is determined by the following formula.

*IAP high boundary = **0x3FFF**.*
*IAP low boundary = **0x3FFF** – IAP + 1.*

For example in **MG82FG5D16**, if ISP-memory is **1.5K**, so that ISP start address is **0x3A00**, and IAP-memory is **1K**, then the IAP-memory range is located at **0x3600** ~ **0x39FF**. The IAP low boundary in **MG82FG5D16** is defined by IAPLB register which can be modified by software to adjust the IAP size in user's AP program.

26.4.2. Update data in IAP-memory

The special function registers are related to ISP/IAP would be shown in Section "26.6 ISP/IAP Register".

Because of 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.

26.4.3. Notes for IAP

Interrupts during IAP

After triggering the ISP/IAP flash processing for In-Application Programming, the MCU will halt for a while for internal IAP processing until the processing is completed. At this time, the interrupt will queue up for being serviced if the interrupt is enabled previously. Once the processing is completed, the MCU continues running and the interrupts in the queue will be serviced immediately if the interrupt flag is still active. Users, however, should be aware of the following:

(1) Any interrupt can not be in-time serviced during the MCU halts for IAP processing.
(2) The low/high-level triggered external interrupts, nINTx, should keep activated until the IAP is completed, or they will be neglected.

IAP and Idle mode

MG82FG5D16 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 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.

26.5. 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 7~4: 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
Others	Reserved

IFMT is used to select the flash mode for performing numerous ISP/IAP function or to select page P SFR access.

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 RESET = 0000-xxxx

7	6	5	4	3	2	1	0
ISPEN	SWBS	SWRST	CFAIL	0	0	0	0
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 failed. 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.

CKCON1: Clock Control Register 1

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

SFR Address = 0xBF RESET = 0x00-1011

7	6	5	4	3	2	1	0
XTOR	0	XCKS5	XCKS4	XCKS3	XCKS2	XCKS1	XCKS0
R	**W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 5~0: This is set the OSCin frequency value to define the time base of ISP/IAP programming. Fill with a proper value according to OSCin, as listed below.

[XCKS5~XCKS0] = OSCin – 1, w*here OSCin=1~40 (MHz).*

For examples,
(1) If OSCin=12MHz, then fill [XCKS5~XCKS0] with 11, i.e., 00-1011B.
(2) If OSCin=6MHz, then fill [XCKS5~XCKS0] with 5, i.e., 00-0101B.

OSCin	XCKS[5:0]
1MHz	00-0000
2MHz	00-0001
3MHz	00-0010
4MHz	00-0011
……	……
……	……
38MHz	10-0101
39MHz	10-0110
40MHz	10-0111

The default value of XCKS= 00-1011 for OSCin= 12MHz.

IAPLB: IAP Low Boundary

SFR Page = **Page P Only**

SFR Address = 0x03 RESET = 0011-0110, 0001-0110

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 in **MG82FG5D16** , if IAPLB=**0x36** and ISP start address is **0x3A00**, then the IAP-memory range is located at **0x3600 ~ 0x39FF**.*

Additional attention point, the IAP low boundary address must not be higher than ISP start address.

26.6. ISP/IAP Sample Code

The following Figure 26–9 shows a sample code for ISP operation.

Figure 26-8. 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:
       ...
;
```

27. Page P SFR Access

MG82FG5D16 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 = 0011-0110, 0001-0110

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 in **MG82FG5D16** , if IAPLB=**0x36** and ISP start address is **0x3A00**, then the IAP-memory range is located at **0x3600 ~ 0x39FF**.*

Additional attention point, the IAP low boundary address must not be higher than ISP start address.

CKCON2: Clock Control Register 2

SFR Page = **P**

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, OSC Driving control Register.

XTGS1, XTGS0	Gain Define
0, 0	Gain for 32.768K
0, 1	Gain for 2MHz ~ 25MHz
Others	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 x 4 (ENCKM =1)	24MHz	22.118MHz
1 0	CKMI x 5.33 (ENCKM =1)	32MHz	29.491MHz
1 1	CKMI x 8 (ENCKM =1)	48MHz	44.236MHz

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**

SFR Address = 0x41 RESET = 0000-0010

7	6	5	4	3	2	1	0
WDTCS1	WDTCS0	--	WDTFS	MCKD1	MCKD0	**1**	**0**
R/W	**R/W**	**W**	**R/W**	**R/W**	**R/W**	**W**	**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	Reserved

Bit 5: Reserved. Software must write “0” on this bit when CKCON3 is written.

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: Reserved. Software must write “**10**” on these bits when PCON1 is written.

CKCON4: Clock Control Register 4

SFR Page = **P only**

SFR Address = 0x42 POR = 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: RCSS2~0, RTC module Clock Source Selection bits 2~0.

RCSS[2:0]	RTC module Clock Source
0 0 0	XTAL2/ECKI (P6.0)
0 0 1	ILRCO
0 1 0	WDTPS
0 1 1	WDTOF
1 0 0	SYSCLK
Others	Reserved.

Bit 4: RPSC2, RTC PreScaler Control 2.
Bit 3: RPSC1, RTC PreScaler Control 1.
Bit 2: RPSC1, RTC PreScaler Control 0.

Bit 1~0: RTCCS3~2, RTC Counter Clock Selection. The function is active with RTCCS1~0.

PCON2: Power Control Register 2

SFR Page = **P Only**

SFR Address = 0x44 POR = 0011-0101

7	6	5	4	3	2	1	0
AWBOD1	**0**	BO1S1	BO1S0	BO1RE	EBOD1	BO0RE	**1**
R/W	W	R/W	R/W	R/W	R/W	R/W	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 0: 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	3.7V
1 1	4.2V

Bit 3: BO1RE, BOD1 Reset Enabled.
0: Disable BOD1 to trigger a system reset when BOF1 is set.
1: Enable BOD1 to trigger a system reset when BOF1 is set.

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

Bit 1: BO0RE, BOD0 Reset Enabled.
0: Disable BOD0 to trigger a system reset when BOF0 is set.
1: Enable BOD0 to trigger a system reset when BOF0 is set (VDD meets 2.2V).

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

PCON3: Power Control Register 3

SFR Page = **P Only**

SFR Address = 0x45 POR = 0xxx-xxxx

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

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

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

SPCON0: SFR Page Control 0

SFR Page = **P Only**

SFR Address = 0x48 POR = 0000-0000

7	6	5	4	3	2	1	0
RTCCTL	P6CTL	P4CTL	WRCTL	CKCTL1	CKCTL0	PWCTL1	PWCTL0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: RTCCTL. RTCCR SFR access Control.
If RTCCTL is set, it will disable the RTCCR SFR modified in general Page. RTCCR in general Page only keeps the SFR read function. But software always owns the modification capability in SFR Page P.

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 3: CKCTL1. CKCON1 SFR access Control.
If CKCTL1 is set, it will disable the CKCON1 SFR modified in Page 0~F. CKCON1 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 = Page P Only

SFR Address = 0x4C RESET = 10x-x011

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

Bit 7: HSE, High Speed operation Enable.
0: Select CPU running in lower speed mode (F_{CPUCLK} ≤ 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 SYSCLK, 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)

Bit 4~3: Reserved. Software must write "**0**" on these bits when DCON0 is written.

Bit 2: IORCTL, GPIO Reset Control.
0: Port 6 keeps reset condition for all reset events.
1: If this bit is set, Port 6 is only reset by POR/LVR/Ext_Reset/BOR0/BOR1 (if BOR0/1 is enabled).

Bit 1: RSTIO, RST function on I/O,
0: Select I/O pad function for P47.
1: Select I/O pad function for external reset input, RST.

Bit 0: OCDE, OCD enable.
0: Disable OCD interface on P4.4 and P4.5
1: Enable OCD interface on P4.4 and P4.5.

28. Auxiliary SFRs

AUXR0: Auxiliary Register 0

SFR Page = 0~F
SFR Address = 0xA1 RESET = 000x-xx00

7	6	5	4	3	2	1	0
P60OC1	P60OC0	P60FD	--	--	--	INT1H	INT0H
R/W	**R/W**	**R/W**	**W**	**W**	**W**	**R/W**	**R/W**

Bit 7~6: P6.0 function configured control bit 1 and 0. The two bits only act when internal RC oscillator (IHRCO or ILRCO) is selected for system clock source. In 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 P6M0.0
01	MCK	By P6M0.0
10	MCK/2	By P6M0.0
11	MCK/4	By P6M0.0

Please refer Section "8 System Clock" to get the more detailed clock information. For clock-out on P6.0 function, it is recommended to set P6M0.0 to "1" which selects P6.0 as push-push output mode.

Bit 5: P60FD, P6.0 Fast Driving.
0: P6.0 output with default driving.
1: P6.0 output with fast driving enabled. If P6.0 is configured to clock output, enable this bit when P6.0 output frequency is more than 12MHz at 5V application or more than 6MHz at 3V application.

Bit 4~2: Reserved. Software must write "**0**" on these bits 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
0	0	0	0	0	0	0	DPS
W	**W**	**W**	**W**	**W**	**W**	**W**	**R/W**

Bit 7~1: Reserved. Software must write "**0**" on these bits when AUXR1 is written.

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 = 00xx-0000

7	6	5	4	3	2	1	0
STAF	STOF	0	0	T1X12	T0X12	T1CKOE	T0CKOE
R/W	**R/W**	**W**	**W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: STAF, Start Flag detection of TWI2.
0: Clear by firmware by writing "0" on it.
1: Set by hardware to indicate the START condition occurred on STWI bus.

Bit 6: STOF, Stop Flag detection of TWI2.
0: Clear by firmware by writing "0" on it.
1: Set by hardware to indicate the START condition occurred on STWI bus.

Bit 5~4: Reserved. Software must write "**0**" on these bits when AUXR2 is written.

When selecting the asynchronous clock (external clock from I/O pin or ILRCO) as the timer clock source, it's frequency needs to lower then 1/2 of the system clock to ensure the edge event can be latched to trigger the counter.

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

Bit 1: T1CKOE, Timer 1 Clock Output Enable.
0: Disable Timer 1 clock output.
1: Enable Timer 1 clock output on T1CKO port pin.

Bit 0: T0CKOE, Timer 0 Clock Output Enable.
0: Disable Timer 0 clock output.
1: Enable Timer 0 clock output on T0CKO port pin.

AUXR3: Auxiliary Register 3

SFR Page = **0 only**

SFR Address = 0xA4 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
00	P3.4
01	P4.4
10	P2.2
11	P2.6

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.

Bit 3: S0PS0, Serial Port 0 pin Selection 0.

S0PS0	RXD0	TXD0
0	P3.0	P3.1
1	P4.4	P4.5

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

TWIPS1~0	TWI0_SCL	TWI0_SDA
00	P4.0	P4.1
01	P6.0	P6.1
10	P3.0	P3.1
11	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 = 0000-0x00

7	6	5	4	3	2	1	0
T2PS1	T2PS0	T1PS1	T1PS0	SPIPS0	--	AC0OE	AC0FLT1
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**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	P3.0	P3.1
10	P4.0	P4.1
11	P4.5	P4.4

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

T1PS1~0	T1/T1CKO
00	P3.5
01	P4.5
10	Reserved
11	P2.6

Bit 3: SPIPS0, SPI Port pin Selection 0.

SPIPS0	nSS	MOSI	MISO	SPICLK
0	P1.4	P1.5	P1.6	P1.7
1	P3.4	P3.5	P4.1	P4.0

Bit 2: Reserved. Software must write "**0**" on this bit 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 P6.1.

Bit 0: AC0FLT1 is the Analog Comparator 0 (AC0) input filter control bit. Please refer AC0FLT for AC0FLT1 function definition.

AUXR5: Auxiliary Register 5

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

7	6	5	4	3	2	1	0
C0IC4S0	C0IC2S0	C0PPS1	C0PPS0	C0PS1	C0PS0	ECIPS0	C0COPS
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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

C0IC4S0	CEX4 input
0	CEX4 Port Pin
1	AC0OUT

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

C0IC2S0	CEX2 input
0	CEX2 Port Pin
1	ILRCO

Bit 5: C0PPS1, {PWM2A, PWM2B} Port pin Selection 0.

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	P2.0	P2.1
1	P6.0	P6.1

Bit 3: C0PS1, PCA0 Port pin Selection 1.

C0PS1	CEX3	CEX5
0	P3.4	P3.5
1	P2.0	P2.1

Bit 2: C0PS0, PCA0 Port pin Selection0.

C0PS0	CEX0	CEX2	CEX4
0	P2.2	P2.4	P2.6
1	P3.0	P3.1	P3.3

Notice: When CEX1 and CEX4 both have been selected output to port pin, please note CEX1 will output through P3.3, therefore the CEX4 only can use P2.6 to output the signal.

Bit 1: ECIPS0, PCA0 ECI Port pin Selection0.

ECIPS0	ECI
0	P1.3
1	P1.6

Bit 0: C0COPS, PCA0 Clock Output (C0CKO) port pin Selection.

C0COPS	C0CKO
0	P4.7
1	P3.3

AUXR6: Auxiliary Register 6

SFR Page = **3 only**

SFR Address = 0xA4 RESET = 0000-0x00

7	6	5	4	3	2	1	0
KBI4PS1	KB4IPS0	KBI6PS0	KBI2PS0	KBI0PS0	0	S0MIPS	S0COPS
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**W**	**R/W**	**R/W**

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

KBI4PS1~0	KBI4	KBI5
00	P1.4	P1.5
01	P3.4	P3.5
10	P6.0	P6.1
11	P2.0	P2.1

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

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

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

KBI2PS0	KBI2	KBI3
0	P1.2	P1.3
1	P2.2	P2.4

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

KBI0PS	KBI0	KBI1
0	P1.0	P1.1
1	P4.0	P4.1

Bit 2: Reserved. Software must write "**0**" on this bit when AUXR6 is written.

Bit 1: S0MIPS, S0MI Port pin Selection.

S0MIPS	S0MI
0	P1.6
1	P3.3

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

S0COPS	S0CKO
0	P4.7
1	P3.3

AUXR7: Auxiliary Register 7

SFR Page = **4 only**

SFR Address = 0xA4 RESET = 1100-xxxx

7	6	5	4	3	2	1	0
POE5	POE4	C0CKOE	SPI0M0	0	0	0	0
R/W	**R/W**	**R/W**	**R/W**	**W**	**W**	**W**	**W**

Bit 7: POE5, PCA0 PWM5 main channel (PWM5O) output control.
0: Disable PWM5O output on port pin.
1: Enable PWM5O output on port pin. **Default is enabled.**

Bit 6: POE4, PCA0 PWM4 main channel (PWM4O) output control.
0: Disable PWM4O output on port pin.
1: Enable PWM4O output on port pin. **Default is enabled.**

Bit 5: C0CKOE, PCA0 clock output enable.
0: Disable PCA0 clock output.
1: Enable PCA0 clock output with PCA0 base timer overflow rate/2.

Bit 4: SPI0M0, SPI0 model control bit 0. It controls the SPI application with daisy-chain connection.
0: Disable the mode control.
1: Enable the mode control.

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

SFRPI: SFR Page Index Register

SFR Page = 0~F

SFR Address = 0xAC RESET = xxxx-0000

7	6	5	4	3	2	1	0
0	0	0	0	PIDX3	PIDX2	PIDX1	PIDX0
W	**W**	**W**	**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. The available pages are only page **"0" to "F"**.

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

29. Hardware Option

The MCU's Hardware Option defines the device behavior which cannot be programmed or controlled by software. The hardware options can only be programmed by a Universal Programmer, the "Megawin 8051 Writer U1" or the "Megawin 8051 ICE Adapter" (The ICE adapter also supports ICP programming function. Refer Section "30.5 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 **MG82FG5D16** 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, **MG82FG5D16** 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	**MG82FG5D16** ISP Start Address
7.5K bytes	2200
7.0K bytes	2400
6.5K bytes	2600
6.0K bytes	2800
5.5K bytes	2A00
5.0K bytes	2C00
4.5K bytes	2E00
4.0K bytes	3000
3.5K bytes	3200
3.0K bytes	3400
2.5K bytes	3600
2.0K bytes	3800
1.5K bytes	3A00
1.0K bytes	3C00
0.5K bytes	3E00
No ISP Space	--

HWBS:

☑: Enabled. When powered up, MCU will boot from ISP-memory if ISP-memory is configured.
☐: Disabled. MCU always boots from AP-memory.

HWBS2:

☑: Enabled. Not only power-up but also any reset will cause 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 **2.5K** bytes.

BO1S1O, BO1S0O:
☑,☑: Select BOD1 to detect 2.0V.
☑,☐: Select BOD1 to detect 2.4V.
☐,☑: Select BOD1 to detect 3.7V.
☐,☐: Select BOD1 to detect 4.2V.

BO0REO:
☑: Enabled. BOD0 will trigger a RESET event to CPU on AP program start address. (2.2V)
☐: Disabled. BOD0 can not trigger a RESET to CPU.

BO1REO:
☑: Enabled. BOD1 will trigger a RESET event to CPU on AP program start address. (4.2V, 3.7V, 2.4V or 2.0V)
☐: Disabled. BOD1 can not 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.

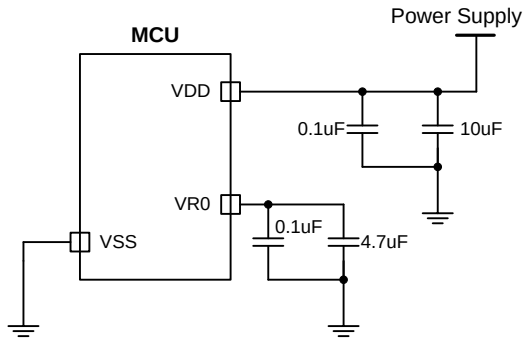
WDSFWP:
☑: Enabled. The WDT SFRs, WREN, NSW, WIDL, PS2, PS1 and PS0 in WDTCR, will be write-protected.
☐: Disabled. The WDT SFRs, WREN, NSW, WIDL, PS2, PS1 and PS0 in WDTCR, are free for writing of software.

30. Application Notes

30.1. Power Supply Circuit

To have the **MG82FG5D16** work with power supply varying from 1.8V to 5.5V, adding some external decoupling and bypass capacitors is necessary, as shown in Figure 30–1.

Figure 30-1. Power Supplied Circuit



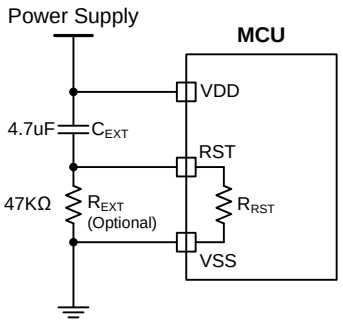
30.2. Reset Circuit

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 30–2 shows the external reset circuit, which consists of a capacitor C_{EXT} connected to VDD (power supply) and a resistor R_{EXT} connected to VSS (ground).

In general, R_{EXT} is optional because the RST pin has an internal pull-down resistor (R_{RST}). This internal diffused resistor to VSS permits a power-up reset using only an external capacitor C_{EXT} to VDD.

See Section "31.2 DC Characteristics" for R_{RST} value.

Figure 30-2. Reset Circuit



30.3. XTAL Oscillating Circuit

To achieve successful and exact oscillating (up to 24MHz), the capacitors C1 and C2 are necessary, as shown in Figure 30–3. Normally, C1 and C2 have the same value. Table 30–1 lists the C1 & C2 value for the different frequency crystal application.

Figure 30-3. XTAL Oscillating Circuit

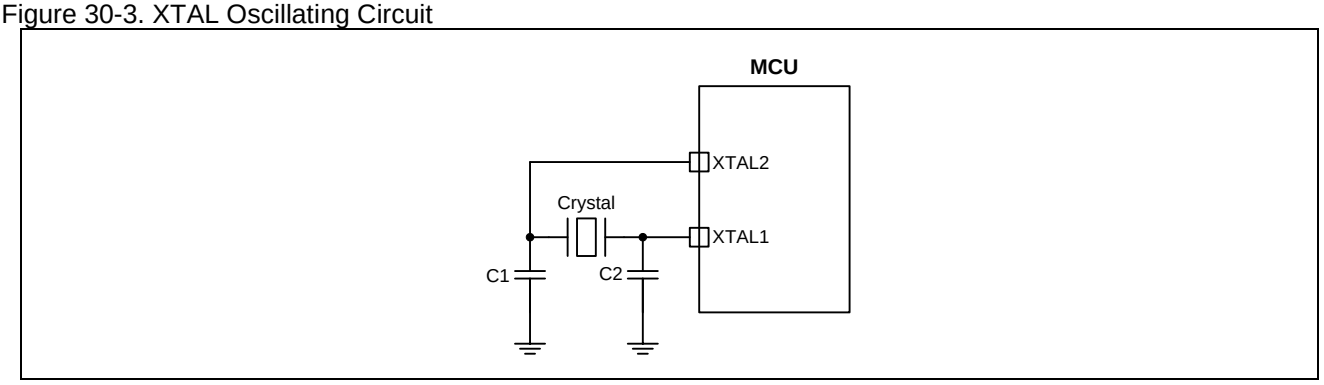


Table 30-1. Reference Capacitance of C1 & C2 for crystal oscillating circuit

Crystal	C1, C2 Capacitance
16MHz ~ 25MHz	10pF
6MHz ~ 16MHz	15pF
2MHz ~ 6MHz	33pF
32768Hz	7pF

30.4. ICP and OCD Interface Circuit

MG82FG5D16 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.

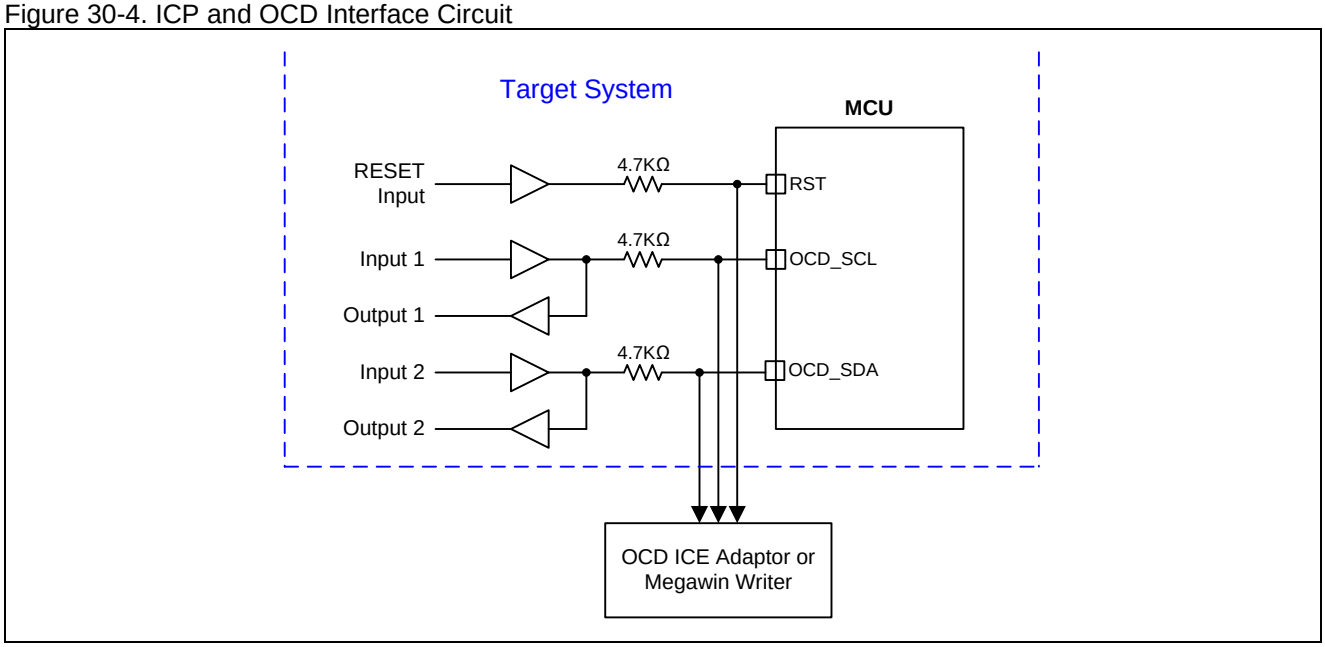
The ICP interface allows the ICP_SCL/ICP_SDA pins to be shared with user functions so that In-Chip Flash Programming function could be performed. This is practicable because ICP communication is performed when the device is in the halt state, where the on-chip peripherals and user software are stalled. 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 30–4.

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 MG82FG5D16 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 30–4, 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 PCON3. When user would like to regain the OCD function, user can predict an event that triggers the software to switch the P4.4 and P4.5 back to OCD_SCL and OCD_SDA by setting OCED as "1". Or "Erase" the on-chip flash by ICP which cleans the user software to stop the port pins switching.

Figure 30-4. ICP and OCD Interface Circuit



30.5. 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.

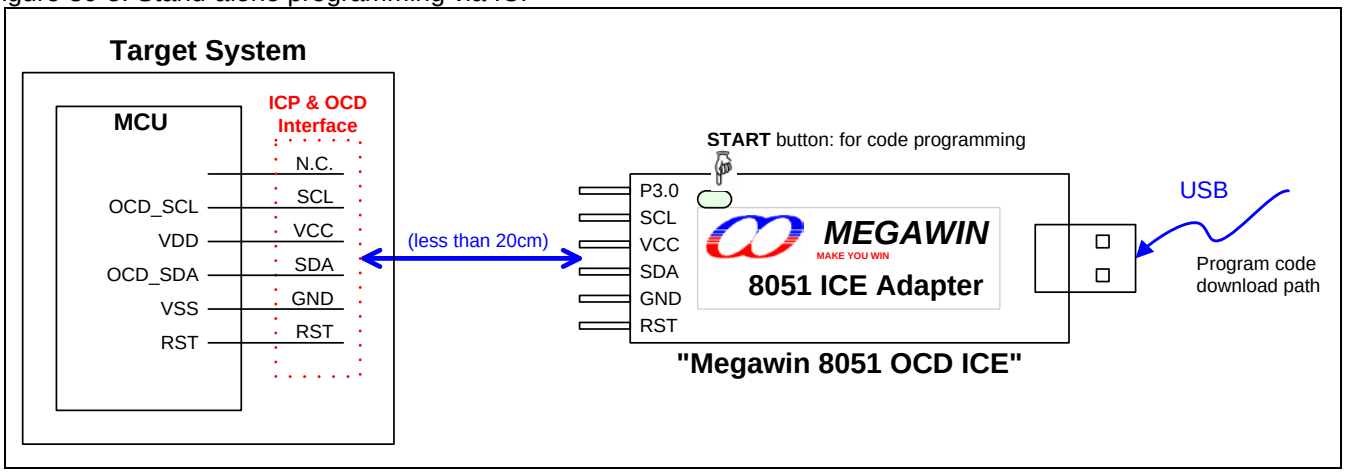
The proprietary 6-pin "Megawin 8051 ICE Adapter" can support the In-Circuit Programming of MG82FG5D16. "Megawin 8051 ICE Adapter" has the in-system storage to store the user program code and device options. So, the tools can perform a portable and stand-alone programming without a host on-line, such as connecting the tool to PC. Following lists the features of the ICP function:

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 30–5. 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 "Megawin 8051 ICE Adapter" to the target MCU; the RST line to halt the MCU, and the VCC & GND are the power supply entry of the 6-pin "Megawin 8051 ICE Adapter" for portable programming application. The USB connector can be directly plugged into the PC's USB port to download the programming data from PC to the 6-pin "Megawin 8051 ICE Adapter".

Figure 30-5. Stand-alone programming via ICP



30.6. On-Chip-Debug Function

The **MG82FG5D16** is equipped with a Megawin proprietary On-Chip Debug (OCD) interface for In-Circuit Emulator (ICE). The OCD interface provides on-chip and in-system non-intrusive debugging without any target resource occupied. Several operations necessary for an ICE are supported, such as Reset, Run, Stop, Step, Run to Cursor and Breakpoint Setting.

Using the OCD technology, Megawin provides the “Megawin 8051 OCD ICE” for the user, as shown in Figure 30–6. 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 30–6.

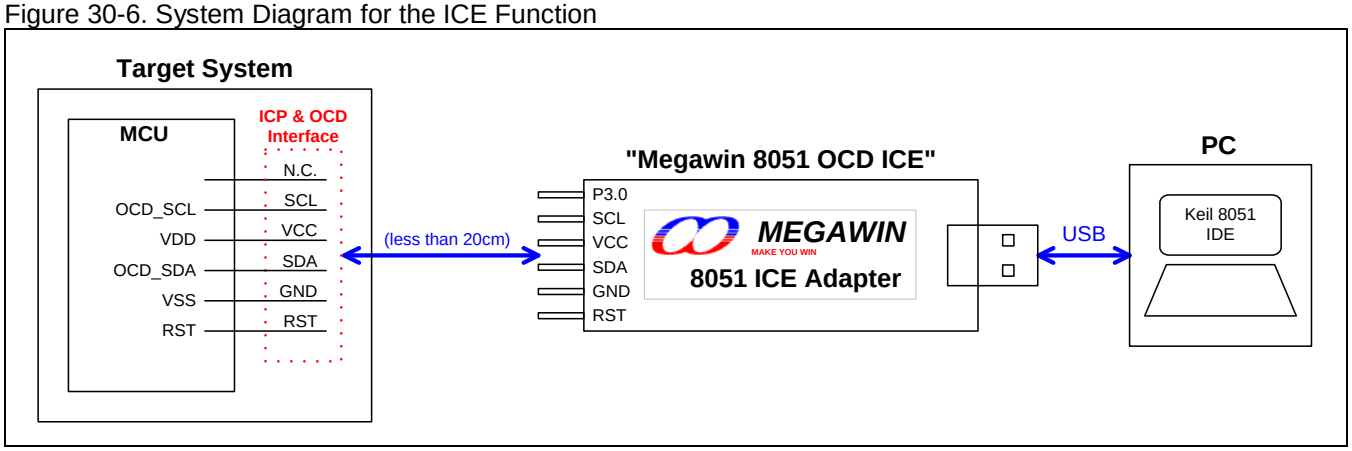
In addition, the most powerful feature is that it can directly connect the user’s target system to the Keil 8051 IDE software for debugging, which directly utilizes the Keil IDE’s dScope-Debugger function. Of course, all the advantages are based on your using Keil 8051 IDE software.

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 30-6. System Diagram for the ICE Function



Note: For more detailed information about the OCD ICE, please feel free to contact Megawin.

31. Electrical Characteristics

31.1. 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 RST 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.

31.2. 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 (RST, 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 (RST, P6.0, P6.1)				0.2	VDD
I_{IH}	Input High Leakage current (All I/O Ports)	V_{PIN} = VDD		0	10	uA
I_{IL1}	Logic 0 input current (P3 in quasi-mode)	V_{PIN} = 0.4V		20	50	uA
I_{IL2}	Logic 0 input current (All Input only or open-drain Ports)	V_{PIN} = 0.4V		0	10	uA
I_{H2L}	Logic 1 to 0 input transition current (P3 in quasi-mode)	V_{PIN} =1.8V		320	500	uA
I_{OH1}	Output High current (P3 in quasi-Mode)	V_{PIN} =2.4V	150	200		uA
I_{OH2}	Output High current (All push-pull output ports)	V_{PIN} =2.4V	12			mA
I_{OL1}	Output Low current (All I/O Ports)	V_{PIN} =0.4V	12			mA
I_{OH2}	Output High current (All push-pull output ports on low driving strength)	V_{PIN} =2.4V, except P6.0, P6.1, P4.7	2			mA
I_{OL2}	Output Low current (All I/O Ports on low driving strength)	V_{PIN} =0.4V, except P6.0, P6.1, P4.7	2			mA
R_{RST}	Internal reset pull-down resistance			110		Kohm
Power Consumption						
I_{OP1}	Normal mode operating current	SYSCLK = 32MHz @ IHRCO with PLL		6.7		mA
I_{OP2}		SYSCLK = 24MHz @ IHRCO with PLL		5.5		mA
I_{OP3}		SYSCLK = 12MHz @ IHRCO		2.9		mA
I_{OP4}		SYSCLK = 12MHz @ IHRCO with ADC				mA
I_{OP5}		SYSCLK = 24MHz @ XTAL		6.2		mA
I_{OP6}		SYSCLK = 12MHz @ XTAL		3.9		mA
I_{OP7}		SYSCLK = 6MHz @ XTAL		2.7		mA
I_{OP8}		SYSCLK = 2MHz @ XTAL		1.7		mA
I_{OPS1}	Slow mode operating current	SYSCLK = 12MHz/128 @ IHRCO		1		mA
I_{OPS2}		SYSCLK = 12MHz/128 @ XTAL		1.6		mA
I_{IDLE1}	Idle mode operating current	SYSCLK = 12MHz @ IHRCO		1.25		mA
I_{IDLE2}		SYSCLK = 12MHz @ XTAL		2.3		mA
I_{IDLE3}		SYSCLK = 12MHz/128 @ IHRCO		0.55		mA
I_{IDLE4}		SYSCLK = 12MHz/128 @ XTAL		1.6		mA
I_{IDLE5}		SYSCLK = 32KHz @ ILRCO		60		uA

Symbol	Parameter	Test Condition	Limits			Unit
			Min.	Typ.	Max.	
I_{SUB1}	Sub-clock mode operating current	SYSCLK = 32KHz @ ILRCO, BOD1 disabled		65		uA
I_{SUB2}		SYSCLK = 32KHz/128 @ ILRCO, BOD1 disabled		60		uA
I_{WAT}	Watch mode operating current	WDT = 32KHz @ ILRCO in PD mode		5		uA
I_{MON1}	Monitor Mode operating current	BOD1 enabled in PD mode		10		uA
I_{RTC1}	RTC Mode operating current	RTC operating in PD mode, VDD = 5.0V		8		uA
		RTC operating in PD mode, VDD = 3.0V		4.2		
I_{PD1}	Power down mode current			2.5		uA
BOD0/BOD1 Characteristics						
V_{BOD0}	BOD0 detection level	T_A = -40°C to +85°C		1.7		V
V_{BOD10}	BOD1 detection level for 2.0V	T_A = -40°C to +85°C		2.0		V
V_{BOD10}	BOD1 detection level for 2.4V	T_A = -40°C to +85°C		2.4		V
V_{BOD11}	BOD1 detection level for 3.7V	T_A = -40°C to +85°C		3.7		V
V_{BOD11}	BOD1 detection level for 4.2V	T_A = -40°C to +85°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}	XTAL Operating Speed 0–24MHz	T_A = -40°C to +85°C	2.7		5.5	V
V_{OP2}	XTAL Operating Speed 0-12MHz	T_A = -40°C to +85°C	2.0		5.5	V
V_{OP3}	CPU Operating Speed 0-32MHz	T_A = -40°C to +85°C	2.7		5.5	V
V_{OP4}	CPU Operating Speed 0-24MHz	T_A = -40°C to +85°C	2.2		5.5	V
V_{OP5}	CPU Operating Speed 0-12MHz	T_A = -40°C to +85°C	1.8		5.5	V

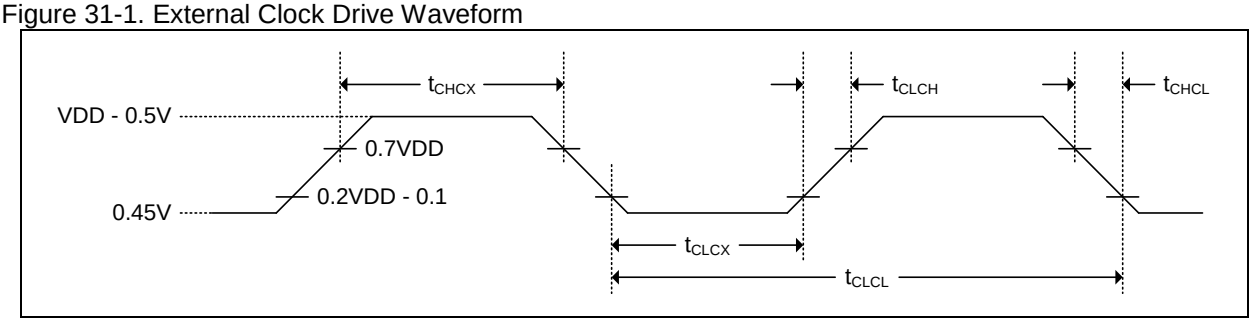
(1) Data based on characterization results, not tested in production.

31.3. External Clock Characteristics

VDD = 2.0V ~ 5.5V, VSS = 0V, T_A = -40°C to +85°C, unless otherwise specified

Symbol	Parameter	Oscillator				Unit
		Crystal Mode		ECKI Mode		
		Min.	Max.	Min.	Max.	
$1/t_{CLCL}$	Oscillator Frequency (VDD = 2.7V ~ 5.5V)	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		41.6		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 31-1. External Clock Drive Waveform



31.4. 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)	T_A = +25°C	-1.0		+1.0	%
	T_A = -40°C to +85°C	-2.0(1)		+2.0(1)	%
IHRCO Start-up Time	T_A = -40°C to +85°C			32(1)	us
IHRCO Power Consumption	T_A = +25°C, VDD=5.0V		350(1)		uA

(1) Data based on characterization results, not tested in production.

31.5. ILRCO Characteristics

Parameter	Test Condition	Limits			Unit
		Min.	Typ.	Max.	
Supply Voltage		1.8		5.5	V
ILRCO Frequency	T_A = +25°C		32		KHz
ILRCO Frequency Deviation	T_A = +25°C	-15(1)		+15(1)	%
	T_A = -40°C to +85°C	-40(1)		+40(1)	%

(1) Data based on characterization results, not tested in production.

31.6. CKM Characteristics

Parameter	Test Condition	Limits			Unit
		Min.	Typ.	Max.	
Supply Voltage	T_A = -40°C to +85°C	2.4		5.5	V
Clock Input Range	T_A = -40°C to +85°C	4.5[1]		6.5[1]	MHz
CKM Start-up Time	T_A = -40°C to +85°C	30[2]		100[2]	us
CKM Power Consumption	T_A = +25°C, VDD=5.0V		480		uA

[1] Data guaranteed by design, not tested in production.
[2] Data based on characterization results, not tested in production.

31.7. Flash Characteristics

Parameter	Test Condition	Limits			Unit
		min	typ	max	
Supply Voltage	T_A = -40°C to +85°C	1.7		5.5	V
Flash Write (Erase/Program) Voltage	T_A = -40°C to +85°C	1.8		5.5	V
Flash Erase/Program Cycle	T_A = -40°C to +85°C	20,000			times
Flash Data Retention	T_A = +25°C	100			year

31.8. ADC Characteristics

VDD=5.0V±10%, VSS=0V, T_A = -40°C to +85°C, C_{LOAD}=10pF unless otherwise specified

Parameter	Test Condition	Limits			Unit
		Min.	Typ.	Max.	
Supply Range					
Supply Voltage		2.7	5.0	5.5	V
DC Accuracy					
Resolution			10		bits
Integral Nonlinearity	VDD> 4V, 1MHz SPS	-4		+4	LSB
Differential Nonlinearity	VDD> 4V, 1MHz ksps	-1		+2	LSB
	VDD>= 2.7V, 666KHz	-1		+2	LSB
Conversion Rate					
SAR Conversion Clock			12	24	MHz
Conversion Time in SAR Clocks			24		clocks
Throughput Rate				1000	ksps
Power Consumption					
Power Supply Current	Operating Mode, 1M sps		1.48		mA
Other Characters and Definitions					
ADC enable time		5			uS

31.9. Analog Comparator AC0 Characteristics

VDD=5.0V±10%, VSS=0V, T_A = -40°C to +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		10		uA
	Low Power State		1		uA
DC Accuracy					
Input Voltage Range	Rail to Rail	50		VDD-50	mV
Input Offset Voltage	VDD= 5.0V		10	50	mV
Input Common Mode Voltage		50		VDD-50	mV
Comparator Hysteresis			9		mV
Response Time (Normal mode)	Rising(V_{OD}=100mV,V_{CM}=1/2 V_{DD})		200		ns
	Falling(V_{OD}=100mV,V_{CM}=1/2 V_{DD})		200		ns
Response Time	Rising(V_{OD}=100mV,V_{CM}=1/2 V_{DD})			10	us
	Falling(V_{OD}=100mV,V_{CM}=1/2 V_{DD})			10	us
Power on Time (from Power-down)	Normal mode	0.5		0.75	us
	Low power mode		TBD		us

31.10. IVR Characteristics

VDD=5.0V±10%, VSS=0V, T_A =-40°C to +85°C, C_{LOAD}=4.7upF/0.1ohm-ESR unless otherwise specified

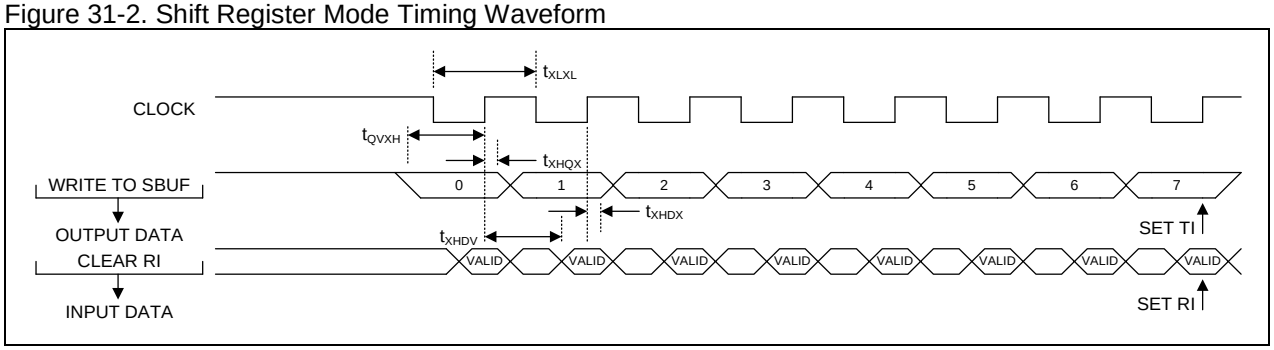
Parameter	Test Condition	Limits			Unit
		Min.	Typ.	Max.	
Supply Range					
Supply Voltage		2.4	5.0	5.5	V
Operation Current	Normal Power State	36.4		67	uA
	Low Power State		0.1		uA
DC Accuracy					
Output Supply Voltage	-40°C ~ +85°C	1.306	1.4	1.497	V

31.11. Serial Port Timing Characteristics

VDD = 5.0V±10%, VSS = 0V, T_A = -40°C to +85°C, unless otherwise specified

Symbol	Parameter	URM0X3 = 0		URM0X3 = 1		Unit
		Min.	Max.	Min.	Max.	
t_{XLXL}	Serial Port Clock Cycle Time	12T		4T		T_{SYSCLK}
t_{QVXH}	Output Data Setup to Clock Rising Edge	10T-20		T-20		ns
t_{XHQX}	Output Data Hold after Clock Rising Edge	T-10		T-10		ns
t_{XHDX}	Input Data Hold after Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		10T-20		2T-20	ns

Figure 31-2. Shift Register Mode Timing Waveform



31.12. SPI Timing Characteristics

VDD = 5.0V±10%, VSS = 0V, T_A = -40°C to +85°C, unless otherwise specified

Symbol	Parameter	Min.	Max.	Units
Master Mode Timing				
t_{MCKH}	SPICLK High Time	2T		T_{SYSCLK}
t_{MCKL}	SPICLK Low Time	2T		T_{SYSCLK}
t_{MIS}	MISO Valid to SPICLK Shift Edge	2T+20		ns
t_{MIH}	SPICLK Shift Edge to MISO Change	0		ns
t_{MOH}	SPICLK Shift Edge to MOSI Change		10	ns
Slave Mode Timing				
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	4T		T_{SYSCLK}
t_{CKL}	SPICLK Low Time	4T		T_{SYSCLK}
t_{SIS}	MOSI Valid to SPICLK Sample Edge	2T		T_{SYSCLK}
t_{SIH}	SPICLK Sample Edge to MOSI Change	2T		T_{SYSCLK}
t_{SOH}	SPICLK Shift Edge to MISO Change		4T	T_{SYSCLK}
t_{SLH}	Last SPICLK Edge to MISO Change (CPHA = 1 ONLY)	1T	2T	T_{SYSCLK}

Figure 31-3. SPI Master Transfer Waveform with CPHA=0

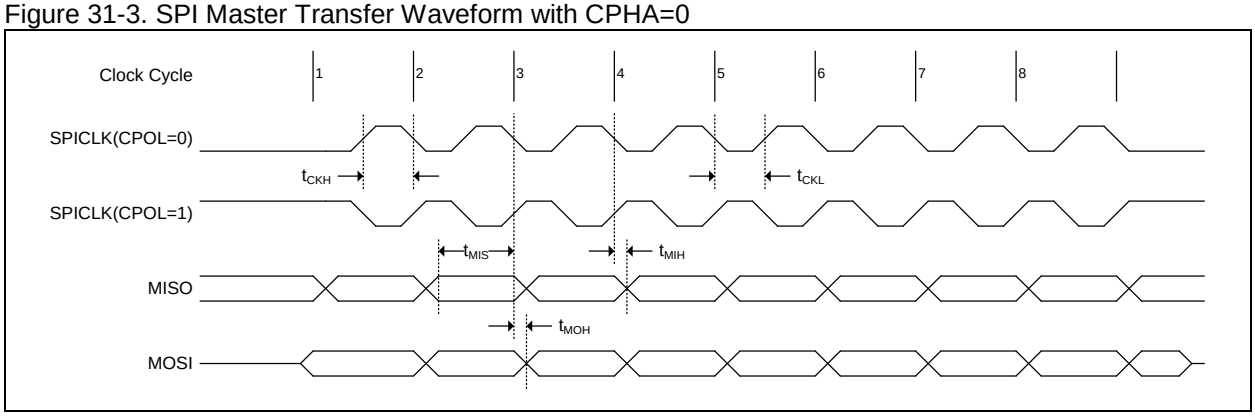


Figure 31-4. SPI Master Transfer Waveform with CPHA=1

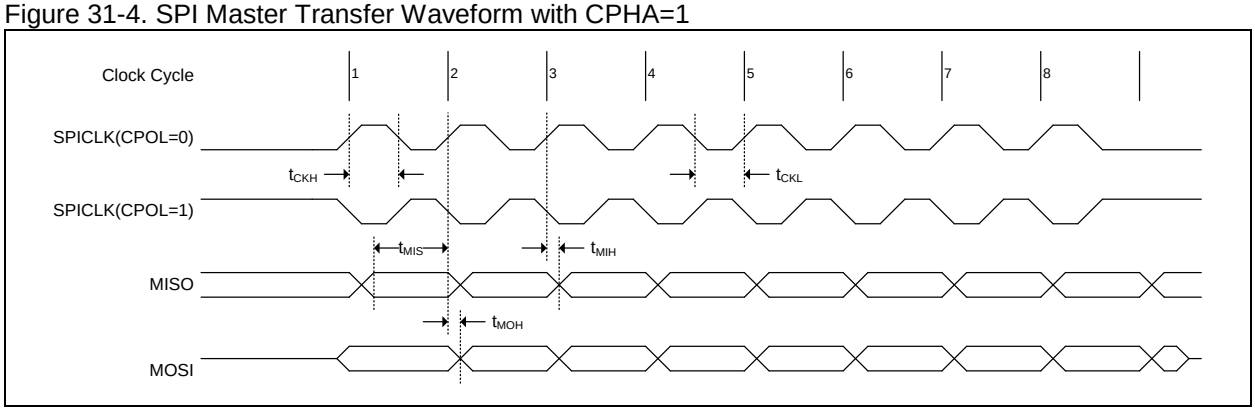


Figure 31-5. SPI Slave Transfer Waveform with CPHA=0

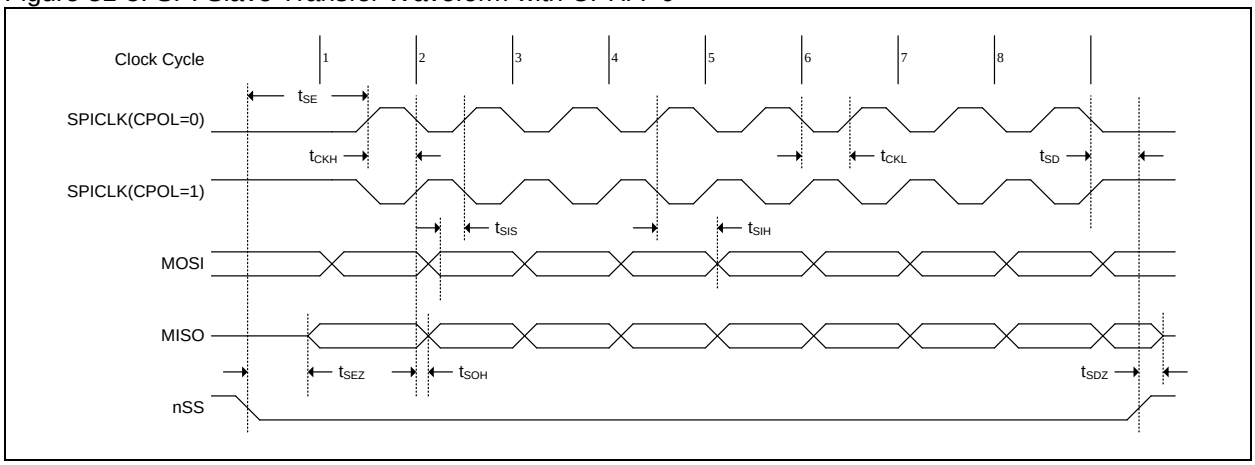
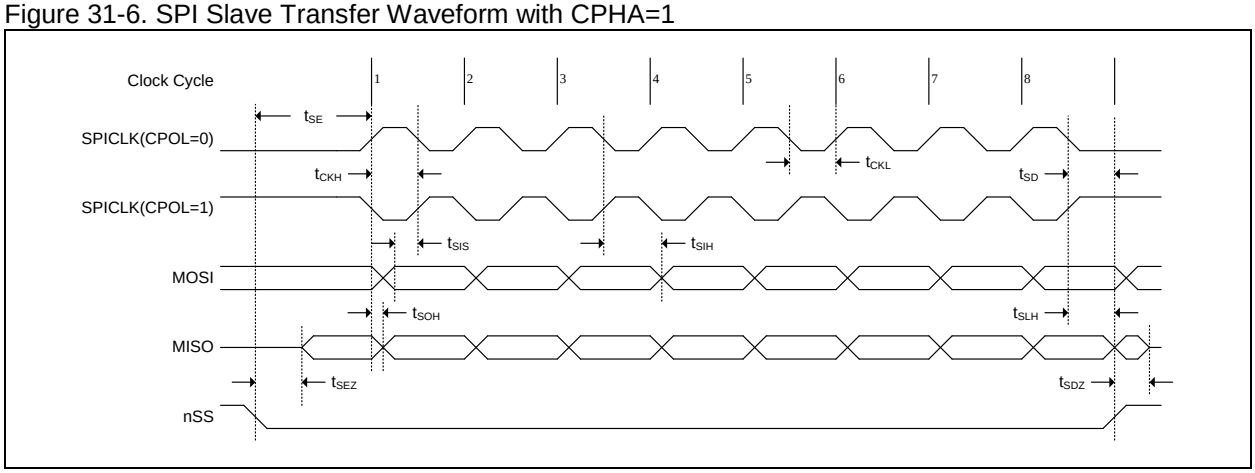


Figure 31-6. SPI Slave Transfer Waveform with CPHA=1



32. Instruction Set

Table 32-1. 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	1	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 ACC	1	3
MOVX @Ri,A	Move ACC to on-chip auxiliary RAM(8-bit address)	1	3
MOVX @DPTR,A	Move ACC to on-chip auxiliary RAM(16-bit address)	1	3
MOVX A,@Ri	Move external RAM(8-bit address) to ACC	1	not support
MOVX A,@DPTR	Move external RAM(16-bit address) to ACC	1	not support
MOVX @Ri,A	Move ACC to external RAM(8-bit address)	1	not support
MOVX @DPTR,A	Move ACC to external RAM(16-bit address)	1	not support
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

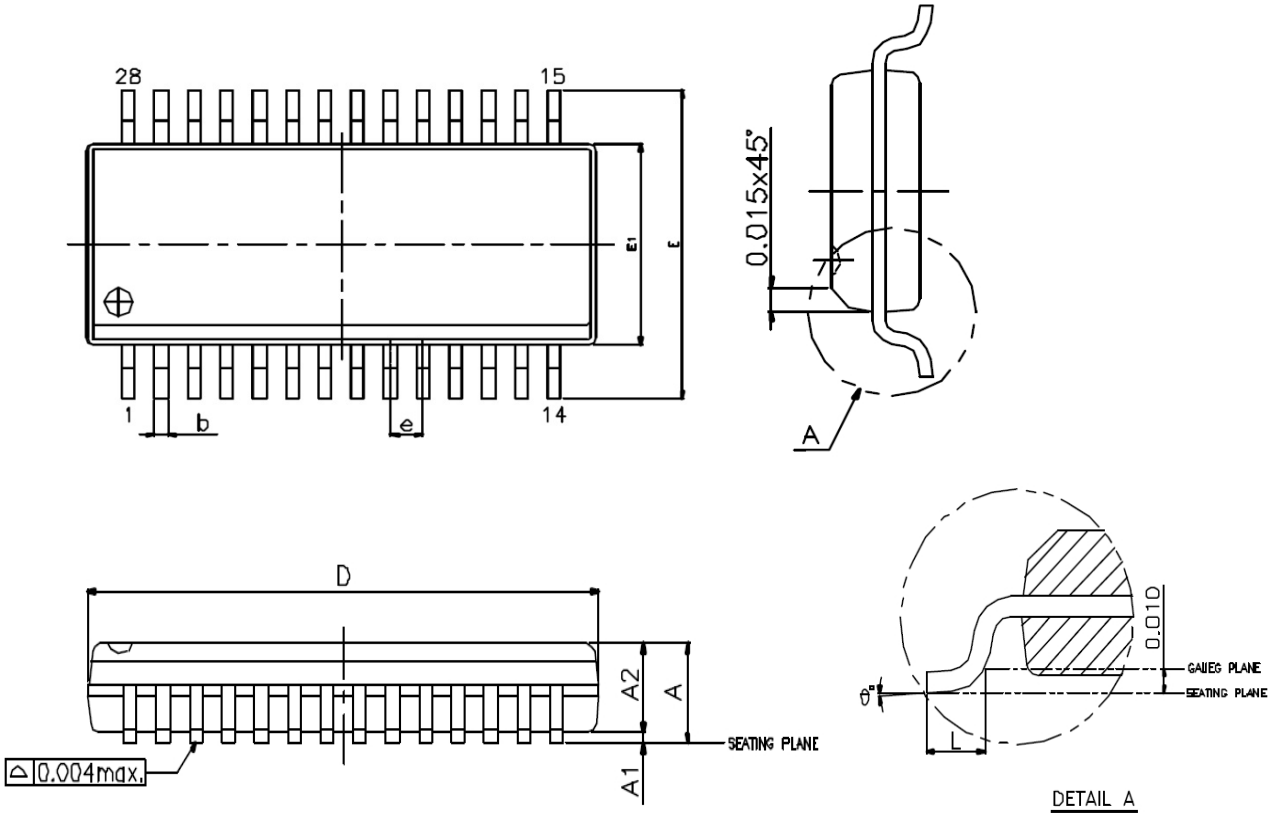
MNEMONIC	DESCRIPTION	BYTE	EXECUTION Cycles
SUBB A,direct	Subtract direct byte from ACC with borrow	2	3
SUBB A,@Ri	Subtract indirect RAM from ACC with borrow	1	3
SUBB A,#data	Subtract immediate data from ACC with borrow	2	2
INC A	Increment ACC	1	2
INC Rn	Increment register	1	3
INC direct	Increment direct 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

MNEMONIC	DESCRIPTION	BYTE	EXECUTION Cycles
CPL bit	Complement direct bit	2	4
ANL C,bit	AND direct bit to Carry	2	3
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 equal	3	4
CJNE Rn,#data,rel	Compare immediate data to register and jump if not equal	3	4
CJNE @Ri,#data,rel	Compare immediate data to indirect RAM and jump if not equal	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

33. Package Dimension

33.1. SSOP28 (150 mil) Package dimension

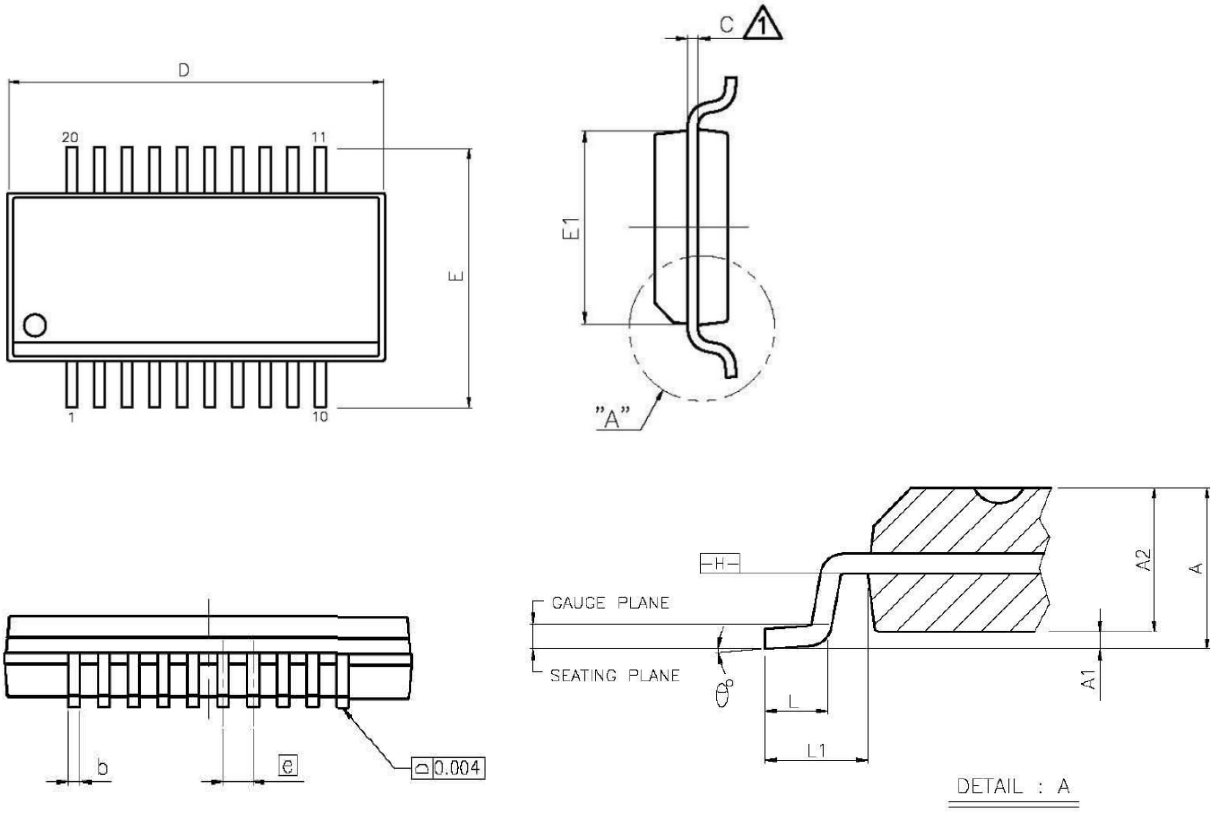
Figure 33-1. SSOP-28 (150 mil) Package dimension



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

33.2. SSOP20 (150 mil) Package dimension

Figure 33-2. SSOP-20 (150 mil) Package dimension

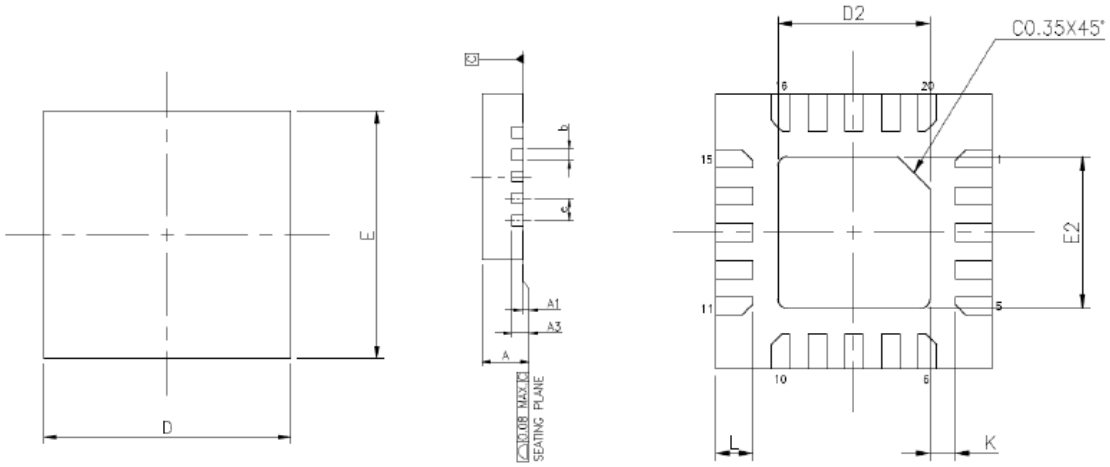


Symbols	Dimensions in mm		
	Min.	Nom.	Max.
A	1.346	1.62	1.752
A1	0.101	0.152	0.254
A2	----	----	1.498
b	0.203	----	0.304
C	0.177	----	0.254
D	8.559	8.661	8.737
E	5.791	5.994	6.197
E1	3.81	3.911	3.987
e	0.635 BASIC		
L	0.406	0.635	1.27
L1	1.0414 BASIC		
θ	0°	----	8°

33.3. QFN-20 (3x3mm)

Figure 33-3. QFN-20

QFN 20P(3x3mm) package dimension

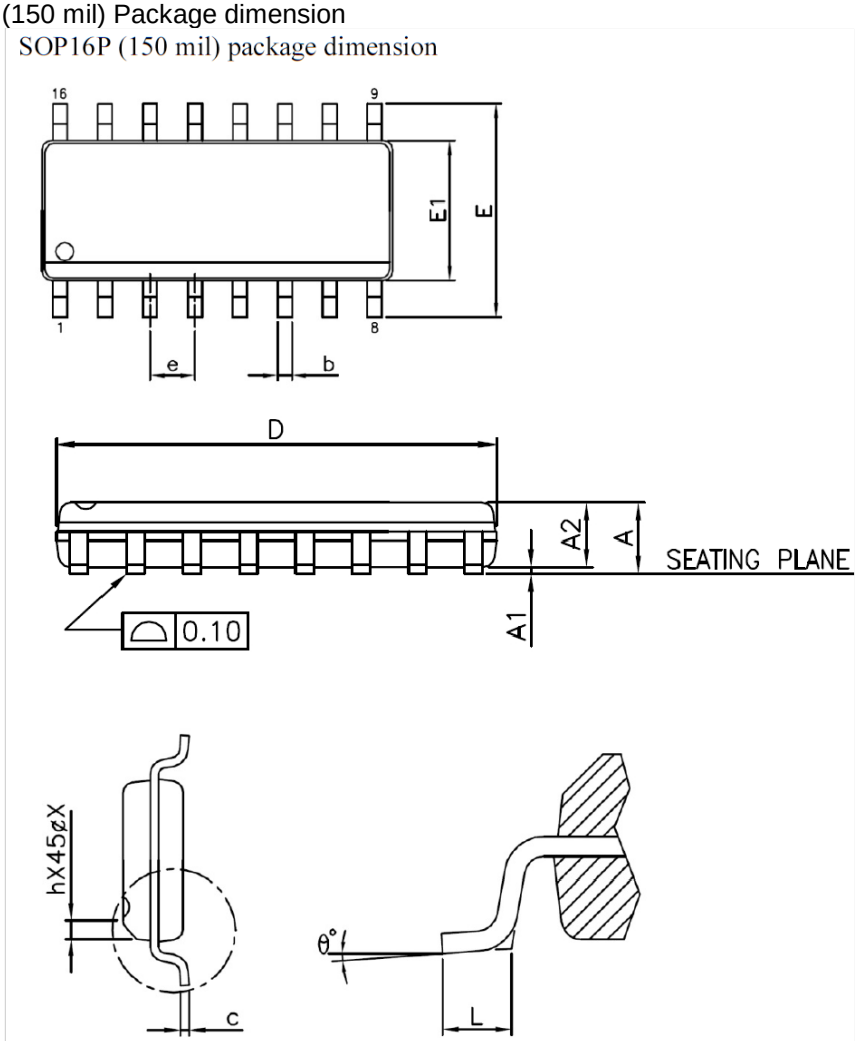


Dimensions in mm			
JEDEC	MO-220		
PKG	WQFN(X319)		
Symbol	Min.	Nom.	Max.
A	0.70	0.75	0.80
A1	0.00	0.02	0.05
A3	0.203 REF.		
b	0.15	0.20	0.25
D	3.00 BSC		
E	3.00 BSC		
e	0.40 BSC		
K	0.20	---	---

PAD SIZE	D2			E2			L			LEAD FINISH		JEDEC CIDE
	Min.	Nom.	Max.	Min.	Nom.	Max.	Min.	Nom.	Max.	Pure Tin	PPF	
75x75MIL	1.60	1.65	1.70	1.60	1.65	1.70	0.30	0.40	0.50	V	X	N/A

33.4. SOP-16 (150 mil) Package dimension

Figure 33-4. SOP-16 (150 mil) Package dimension



Symbols mm	Min.	Max.
A	----	1.75
A1	0.10	0.25
A2	1.25	----
b	0.31	0.51
c	0.10	0.25
D	9.90 BSC	
E	6.00 BSC	
E1	3.90 BSC	
e	1.27 BSC	
L	0.40	1.27
h	0.25	0.50
θ	0	8

34. Revision History

Table 34-1. Revision History

Rev	Descriptions	Date
v0.73	Preliminary version release.	2017/04/20
v0.74	1. Added SOP16, SSOP20, QFN20, SSOP28 Package Information 2. Corrected errors 3. Added CKM 100MHz function 4. Added Timer Global control	2018/01/22
V0.75	1. Removed T1/T1CKO P1.7 Port pin option 2. Added notice on CEX1 and CEX4. Don't use the same Port pin P3.3. 3. Changed AC0PI3 from P6.0 to P1.2 4. Change SSOP28 and SSOP20 pin pitch from 209mil to 150mil 5. Removed NVRS[3:0] = 1xxx to select Int. VREF (1.4V) 6. Modified Interrupt # in the Table 14-1, 14-2 and 14-3 7. Change SFR SPCTL to SPCON	2018/07/20

35. 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).