MEGAWIN
MAKE YOU WIN

8051-Based MCU

MG74PG1A08
Data Sheet

Version: 0.25

Features

- 1-T 80C51 Central Processing Unit
- **MG74PG1A08** with **8K** Bytes **OTP** ROM
  - Code protection for OTP memory access
  - OTP data retention: 10 years at 85°C
- On-chip 256 bytes scratch-pad RAM
- Interrupt controller
  - 6 sources, two-level-priority interrupt capability
  - Two external interrupt inputs, nINT0 and nINT1
  - All external interrupts support High/Low level or Rising/Falling edge trigger
- Two 16-bit timer/counters, Timer 0 and Timer 1
  - T0CKO on P1.4, T1CKO on P1.5
  - X12 mode enabled for T0/T1
- Programmable 16-bit counter/timer Array (PCA) with one capture/compare module.
  - Capture mode, 16-bit software timer mode and High speed output mode
  - 8-bit PWM mode, 16-bit PWM, double channel 8-bit PWM
- Keypad Interrupt function on all GPIO.
- Enhanced UART0 (S0)
  - Framing Error Detection
  - Speed improvement mechanism (X2 mode)
  - SPI master/Slave support in mode 4/6
- Two wire interface Start/Stop detection (STWI, IIC compatible)
- USB Device Controller
  - USB Full speed (12Mbps) operation and USB specification 2.0 compliant
  - Intel 8X931 like USB control flow
  - Built-in USB transceiver and 3.3V regulator
  - Integrated clock recovery, no external crystal required
  - 8-bytes FIFO for EP0 control In/Out
  - 8-bytes FIFO for EP1 INT/BULK In
  - 16-bytes FIFO for EP2 INT/BULK In/Out (default: In)
  - Software-controlled USB connection/disconnection mechanism
- Programmable Watchdog Timer, clock sourced from ILRCO (64KHz)
  - One time enabled by CPU or power-on
  - Interrupt CPU or Reset CPU on WDT overflow
  - Support WDT function in power down mode (watch mode)
- Maximum 11 GPIOs in SOP16 package
  - P3 can be configured to quasi-bidirectional, push-pull output, open-drain output and analog-input-only
  - P1 can be configured to open-drain with pull-up resistor, push-pull output, open-drain and analog-input-only
- Multiple power control modes: idle mode, power-down mode, slow mode, sub-clock mode, watch mode and monitor mode.
  - All interrupts can wake up IDLE mode
  - 8 sources to wake up Power-Down mode
  - Slow mode and sub-clock mode support low speed MCU operation
  - Watch mode supports WDT to resume CPU in power down
- One POR & Two Brown-Out Detectors
  - POR: detect 1.95V (2.3V Select by SFR)
  - BOD0: detect 2.1V (2.6V Select by SFR)

  - BOD1: detect 3.6V
  - Interrupt CPU or reset CPU
- Operating voltage range(with LDO): 2.3V – 5.5V (RMLS=0)
- Operating voltage range(without LDO, for battery): 2.0V – 3.6V (RMLS=0)
- Operation frequency range: 12MHz(max)
  - 0 – 6MHz @ 2.0V – 5.5V, 0 – 12MHz @ 2.4V – 5.5V, 0 – 24MHz @ 2.7V – 5.5V
  - CPU up to 3MHz @ 2.0V – 5.5V, up to 6MHz @ 2.4V – 5.5V and up to 12MHz @ 2.7V – 5.5V
- Clock Sources
  - Internal 12MHz oscillator (IHRCO) with USB clock recovery enabled: ±0.25% accuracy
  - Internal 12MHz oscillator (IHRCO) without USB clock recovery: factory calibrated to ±1%, typical
  - Internal Low frequency RC Oscillator (ILRCO) support: about 64KHz
  - External clock input (ECKI) on P1.7
  - Internal Oscillator output (ICKO) on p1.7
  - On-chip Clock Multiplier (CKM) to provide high speed clock source
- Operating Temperature:
  - Industrial (-40℃ to +85℃)*
- Package Types:
  - DICE: MG74PG1A08AH
  - SOP16: MG74PG1A08AS16
  - QFN16: MG74PG1A08AY16 (4X4)

*: Tested by sampling.

Content

1. General Description

The **MG74PG1A08** is a single-chip micro-controller based on a high performance 1-T architecture 80C51 CPU that executes instructions in 1~7 clock cycles (about 6~7 times the rate of a standard 8051 device), and has an 8051 compatible instruction set. Therefore at the same performance as the standard 8051, the **MG74PG1A08** can operate at a much lower speed and thereby greatly reduce the power consumption.

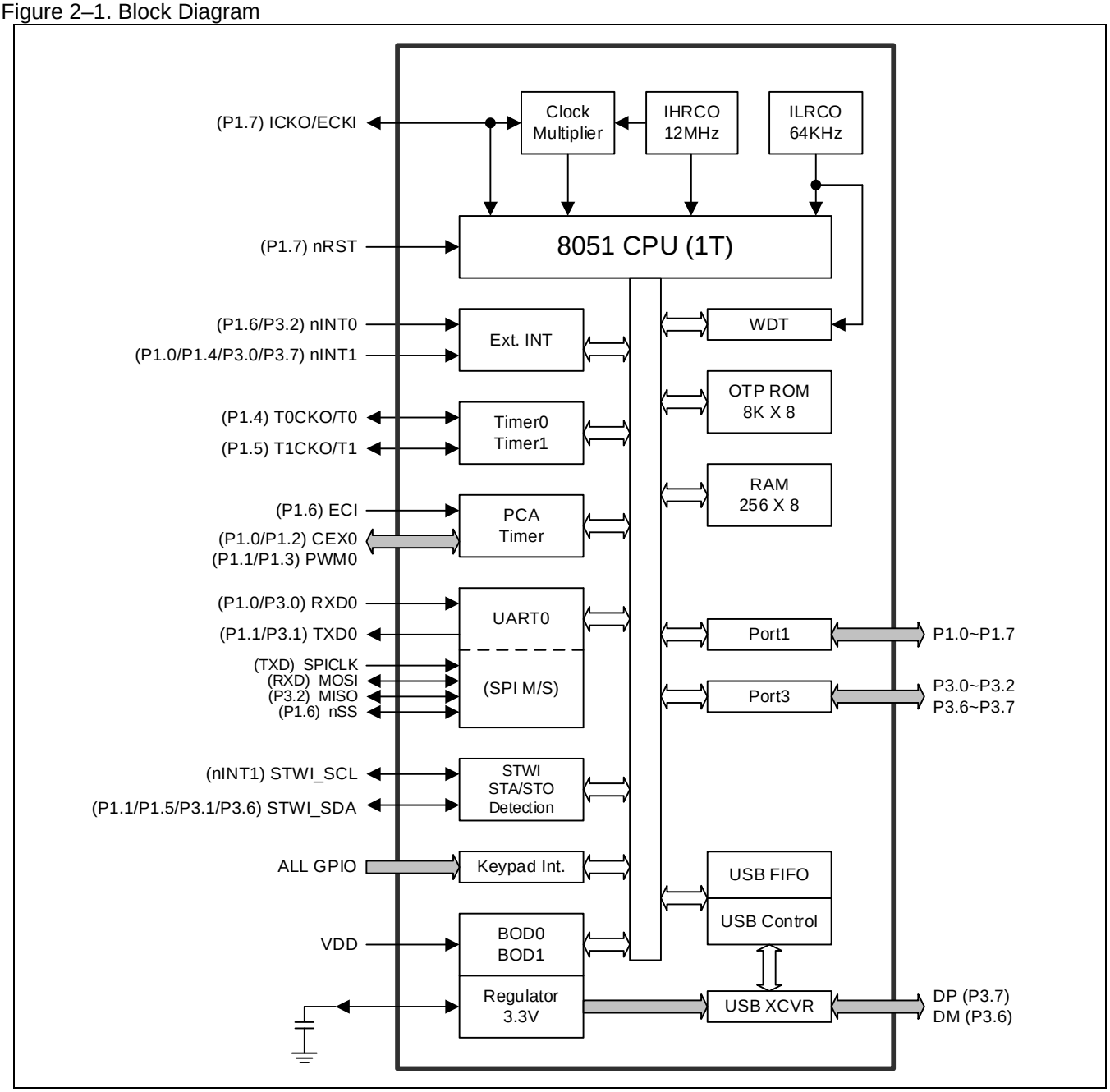
The **MG74PG1A08** has **8K** bytes of embedded **OTP** memory for CPU execution code. The **OTP** memory can be programmed in serial writer mode (via ICP, In-Circuit Programming). ICP allows the user to download new code without removing the microcontroller from the actual end product.

The **MG74PG1A08** retains all features of the standard 80C52 with 256 bytes of scratch-pad RAM, 11 Programmable I/O pins, two external interrupts, a multi-source 2-level interrupt controller, a serial port 0 (UART0) and two timer/counters. In addition, the **MG74PG1A08** has a full speed USB device function, an one-channel 16-bit PCA, SPI, STWI, keypad interrupt, an one-time enabled Watchdog Timer, two Brown-out Detectors, an internal high precision RC oscillator (IHRCO) to fit USB full speed application, an internal low speed RC oscillator (ILRCO) and an enhanced serial function in UART0 that facilitates SPI master/slave engine and a speed improvement mechanism (X2 mode).

The **MG74PG1A08** has multiple operating modes to reduce the power consumption: idle mode, power down mode, slow mode, sub-clock mode, watch mode and monitor mode. In the Idle mode the CPU is frozen while the peripherals and the interrupt system are still operating. In the Power-Down mode the RAM and SFRs' value are saved and all other functions are inoperative; most importantly, in the Power-down mode the device can be waked up by many interrupt or reset sources. In slow mode, the user can further reduce the power consumption by using the 8-bit system clock pre-scaler to slow down the operating speed. Or select sub-clock mode which clock source is derived from internal low speed oscillator (ILRCO) for CPU to perform an ultra-low speed operation. In watch mode, it keeps WDT running in power-down or idle mode and resumes CPU 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.

2. Block Diagram

Figure 2–1. Block Diagram



3. Special Function Register

3.1. SFR Map

Table 3–1. SFR Map

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
F8	--	**CH**	**CCAP0H**	--	--	--	--	--
F0	B	--	--	--	--	--	--	--
E8	--	**CL**	**CCAP0L**	--	--	--	--	--
E0	ACC	WDTCR	--	--	--	--	**SCMD**	--
D8	**CCON**	**CMOD**	**CCAPM0**	--	--	--	--	--
D0	PSW	--	--	--	--	--	**KBIEN0**	--
C8	--	--	--	--	--	--	**CLRL**	**CHRL**
C0	--	--	--	--	--	--	--	**CKCON0**
B8	IP0L	--	**PCON2**	**CKCON2**	**DCON0**	**SPCON0**	--	--
B0	P3	**P3M0**	**P3M1**	--	--	--	--	--
A8	IE	--	**USBDAT**	**USBADR**	--	**XPIE1**	--	--
A0	--	**AUXR0**	**AUXR1**	**AUXR2**	--	--	--	--
98	S0CON	S0BUF	--	--	--	--	--	--
90	P1	**P1M0**	**P1M1**	--	--	--	**BOREV**	**PCON1**
88	TCON	TMOD	TL0	TL1	TH0	TH1	**SFIE**	--
80	--	SP	DPL	DPH	--	--	--	**PCON0**
	0/8	**1/9**	**2/A**	**3/B**	**4/C**	**5/D**	**6/E**	**7/F**

3.2. SFR Bit Assignment

Table 3–2. SFR Bit Assignment

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
PCON0	Power Control 0	87H	SMOD1	SMOD0	GF	POF	GF1	GF0	PD	IDL	00010000
TCON	*Timer Control*	*88H*	*TF1*	*TR1*	*TF0*	*TR0*	*IE1*	*IT1*	*IE0*	*IT0*	*00000000*
TMOD	Timer Mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	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 Enable	8EH	SIDFIE	--	--	--	KBIFIE	BOF1IE	BOF0IE	WDTFIE	0xxx0000
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	00000000
BOREV	Bit Order Reversed	96H	BOREV.7	BOREV.6	BOREV.5	BOREV.4	BOREV.3	BOREV.2	BOREV.1	BOREV.0	00000000
PCON1	Power Control 1	97H	SWRF	EXRF	--	--	KBIF	BOF1	BOF0	WDTF	00xx0000
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
AUXR0	Auxiliary Register 0	A1H	P17OC1	P17OC0	--	T0XL	P1FS1	P1FS0	INT1H	INT0H	00000000
AUXR1	Auxiliary Register 1	A2H	INT1IS1	INT1IS0	INT0IS0	--	STAF	STOF	PTCKOE	--	00000000
AUXR2	Auxiliary Register 2	A3H	UTIE (CPHA)	BTI (SSIG)	URM0X3 (SMOD2)	SM30	T1X12	T0X12	T1CKOE	T0CKOE	00000000
IE	*Interrupt Enable*	*A8H*	*EA*	*--*	*EXPIE1*	*ES0*	*ET1*	*EX1*	*ET0*	*EX0*	*0x000000*
USBDAT	USB Data Register	AAH	UDAT7	UDAT6	UDAT5	UDAT4	UDAT3	UDAT2	UDAT1	UDAT0	xxxxxxxx
USBADR	USB Indirect Address	ABH	UBSY	--	USFRA5	USFRA4	USFRA3	USFRA2	USFRA1	USFRA0	0x111111
XPIE1	Expanded INT. 1 Enable	ADH	EUSB	--	--	--	--	--	EPCA	ESF	0xxxxx00
P3	*Port 3*	*B0H*	*P3.7*	*P3.6*	*--*	*--*	*--*	*P3.2*	*P3.1*	*P3.0*	*11xxx111*
P3M0	P3 Mode Register 0	B1H	P3M0.7	--	--	--	--	P3M0.2	P3M0.1	P3M0.0	0xxxx111
P3M1	P3 Mode Register 1	B2H	P3M1.7	--	--	--	--	P3M1.2	P3M1.1	P3M1.0	0xxxx000
IP0L	*Interrupt Priority Low*	*B8H*	*URXR*	*--*	*PXPI1L*	*PSL*	*PT1L*	*PX1L*	*PT0L*	*PX0L*	*0x000000*
PCON2	Power Control 2	BAH	AWBOD1	EBOD1	--	--	BO1RE	BO0RE	--	RMLS	01000000
CKCON2	Clock Control 2	BBH	--	--	--	IHRCOE	MCKS1	MCKS0	OSCS1	OSCS0	xxx10000
DCON0	Device Control 0	BCH	WCKS	USBR	ENUSB	ENCKM	CKMIS1	CKMIS0	RSTIO	SWRST	00000110
SPCON0	SFR Page Control 0	BDH	--	--	--	WRCTL	--	CKCTL0	PWCTL1	PWCTL0	0xx00000
CKCON0	Clock Control 0	C7H	--	--	CCKS1	CCKS0	--	SCKS2	SCKS1	SCKS0	xx10x001
CLRL	PCA base timer Low Reload register	CEH	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CHRL	PCA base timer High 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*
KBIEN0	KBI Enable 0	D6H	P3HKBI	P3LKBI	--	--	P1HKBI	P1LKBI	--	--	00xx00xx
CCON	*PCA Control Reg.*	*D8H*	*CF*	*CR*	*--*	*--*	*--*	*--*	*--*	*CCF0*	*00xxxx0*
CMOD	PCA Mode Reg.	D9H	CIDL	--	--	--	CPS2	CPS1	CPS0	ECF	0xxx0000
CCAPM0	PCA Module0 Mode	DAH	P0INV	ECOM0	CAPP0	CAPN0	MAT0	TOG0	PWM0	ECCF0	00000000
ACC	*Accumulator*	*E0H*	*ACC.7*	*ACC.6*	*ACC.5*	*ACC.4*	*ACC.3*	*ACC.2*	*ACC.1*	*ACC.0*	*00000000*
WDTCR	Watch-dog-timer Control register	E1H	WREN	NSW	ENW	CLRW	WIDL	PS2	PS1	PS0	00000000
SCMD	ISP Serial Command	E6H	.7	.6	.5	.4	.3	.2	.1	.0	xxxxxxxx
CL	PCA base timer Low	E9H	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP0L	PCA module0 capture Low	EAH	.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*
CH	PCA base timer High	F9H	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP0H	PCA Module0 capture High	FAH	.7	.6	.5	.4	.3	.2	.1	.0	00000000

4. Pin Configurations

4.1. Package Instruction

Figure 4–1. MG74PG1A08AH Top View

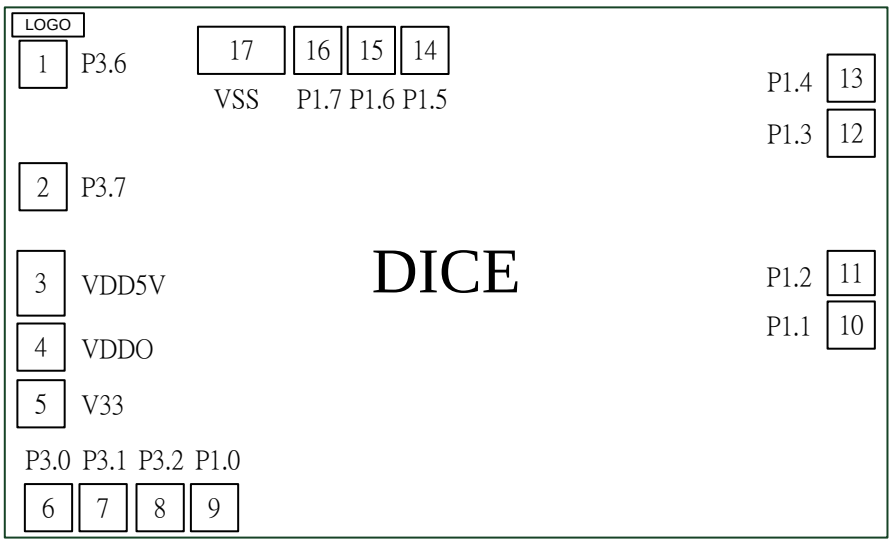


Figure 4–2. MG74PG1A08AS16 Top View

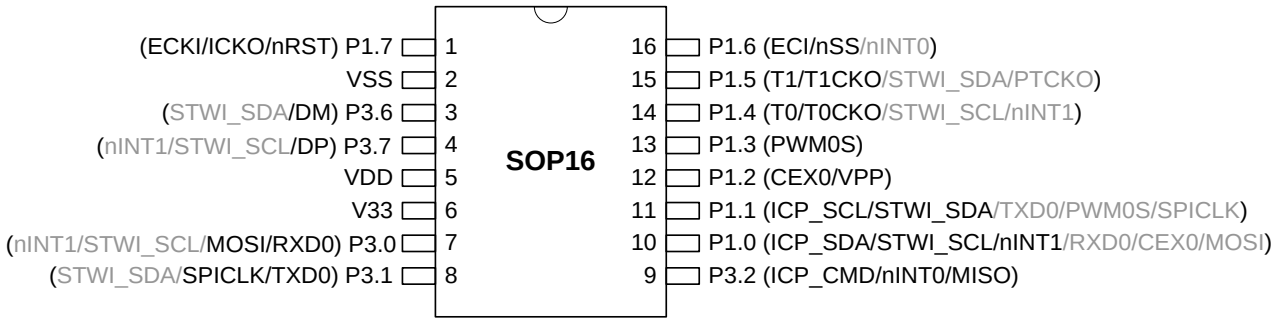
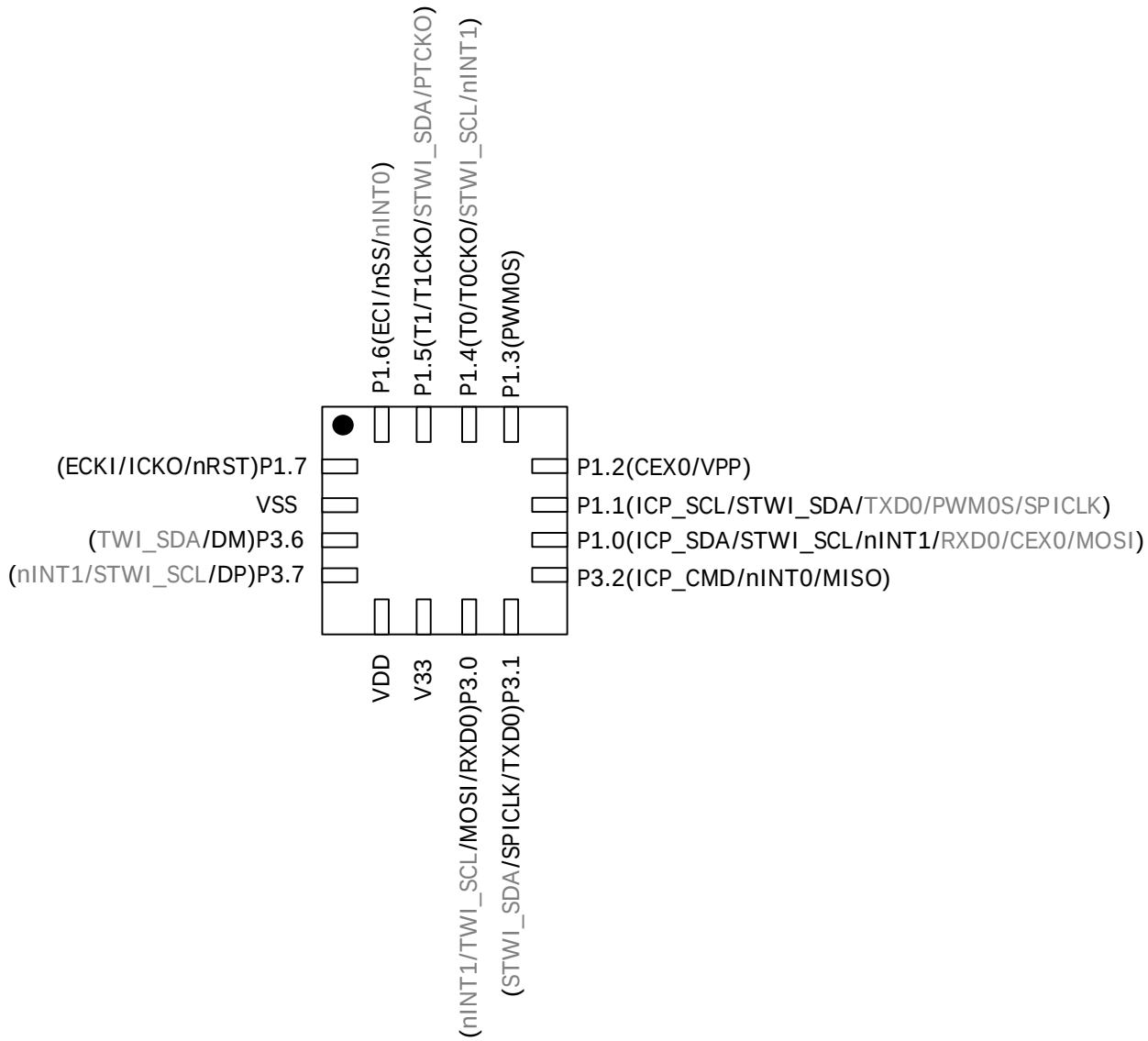


Figure 4–3. MG74PG1A08AY16 Top View



4.2. Pin Description

Table 4–1. Pin Description

MNEMONIC	PIN NUMBER			I/O TYPE	DESCRIPTION
	16-Pin SOP/ QFN	10-Pin SOP	dice		
P3.0 (RXD0) (STWI_SCL) (MOSI) (nINT1)	7	5	6	I/O	* Port 3.0. * RXD0: UART0 serial input port. * TWI_SCL: SCL input for STWI Start/Stop detection. * MOSI: SPI master out & slave in. * nINT1: external interrupt 1 input.
P3.1 (TXD0) (SPICLK) (STWI_SDA)	8	--	7	I/O	* Port 3.1. * TXD0: UART0 serial output port. * SPICLK: SPI clock, output for master and input for slave. * STWI_SDA: SDA input for STWI Start/Stop detection.
P3.2 (nINT0) (MISO) (ICP_CMD)	9	6	8	I/O	* Port 3.2. * nINT0: External interrupt 0 input. * MISO: SPI master in & slave out. * ICP_CMD: Serial command of ICP interface.
P3.6 (DM) (STWI_SDA)	3	1	1	I/O	* Port 3.6. * DM: USB DM (D-) pin. * STWI_SDA: SDA input for STWI Start/Stop detection.
P3.7 (DP) (nINT1) (STWI_SCL)	4	2	2	I/O	* Port 3.7. * DP: USB DP (D+) pin. * nINT1: External interrupt 1 input. * STWI_SCL: SCL input for STWI Start/Stop detection.
P1.0 (nINT1) (RXD0) (MOSI) (STWI_SCL) (CEX0) (ICP_SDA)	10	7	9	I/O	* Port 1.0. * nINT1: External interrupt 1 input. * RXD0: UART0 serial input port * MOSI: SPI master out & slave in. * STWI_SCL: SCL input for TWI Start/Stop detection. * CEX0: PCA module 0 input/output. * ICP_SDA: Serial data of ICP interface.
P1.1 (TXD0) (SPICLK) (STWI_SDA) (PWM0S) (ICP_SCL)	11	8	10	I/O	* Port 1.1. * TXD0: UART0 serial output port * SPICLK: SPI clock, output for master and input for slave. * STWI_SDA: SDA input for TWI Start/Stop detection. * PWM0S: PWM0 Secondary output. * ICP_SCL: Serial clock of ICP interface.
P1.2 (CEX0) (VPP)	12	9	11	I/O	* Port 1.2. * CEX0: PCA module 0 input/output. * VPP: VPP Pad for ICP interface.(attention: VPP voltage is 7.5V)
P1.3 (PWM0S)	13	--	12	I/O	* Port 1.3. * PWM0S: PWM0 Secondary output.
P1.4 (T0) (T0CKO) (nINT1) (STWI_SCL)	14	--	13	I/O	* Port 1.4. * T0: Timer/Counter 0 external input. * T0CKO: Programmable clock-out from Timer 0. * nINT1: External interrupt 1 input * STWI_SCL: SCL input for STWI Start/Stop detection.
P1.5 (T1) (T1CKO) (STWI_SDA) (PTCKO)	15	--	14	I/O	* Port 1.5. * T1: Timer/Counter 1 external input. * T1CKO: Programmable clock-out from Timer 1. * STWI_SDA: SDA input for STWI Start/Stop detection. * PTCKO: PCA Base timer clock output.
P1.6 (nINT0) (ECI) (nSS)	16	--	15	I/O	* Port 1.6. * nINT0: External interrupt 0 input. * ECI: External clock trigger source for PCA * nSS: SPI Slave select.
P1.7 (nRST) (ICKO) (ECKI)	1	--	16	I/O	* Port 1.7. * nRST: External reset input, low-active. * ICKO: Internal clock output. * ECKI: In external clock input mode, this is clock input pin.
V33	6	4	5	P	Core power supply. 3.3V input/output.
VDD	5	3	3	P	Power supply input. 5V input.
VSS	2	10	17	G	Ground, 0 V reference.
VDDO	--	--	4	P	Power supply for I/O pad.

4.3. Alternate Function Redirection

Many I/O pins, in addition to their normal I/O function, also serve the alternate function for internal peripherals. For the peripherals UART0, STWI detection, nINT0 and nINT1, Port 1 and Port 3 serve the alternate function in the default state. However, the user may select other Port to serve their alternate function by setting the corresponding control bits INT1IS1, INT1IS0 and INT0IS0 in AUXR1 register. P1F1~P1FS0 in AUXR0 register select the S0/PCA/Timer0/Timer1 function swapped to Port 1. It is especially useful by software programming.

AUXR0: Auxiliary Register 0

SFR Attribute = Normal Read/Write
SFR Address = 0xA1 RESET = 0000-0000

7	6	5	4	3	2	1	0
P17OC1	**P17OC0**	GF	T0XL	**P1FS1**	**P1FS0**	INT1H	INT0H
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~6: P1.7 function configured control bit 1 and 0. The two bits only act when internal RC oscillator (IHRCO or ILRCO) is selected for system clock source. In external clock input mode, P1.7 is the dedicated clock input pin. In internal oscillator condition, P1.7 provides the following selections for GPIO or clock source generator. When P17OC[1:0] index to non-P1.7 GPIO function, P1.7 will drive the on-chip RC oscillator output to provide the clock source for other devices.

P17OC[1:0]	P1.7 function	I/O mode
00	P1.7	By P1M0.7 & P1M1.7
01	MCK	By P1M0.7 & P1M1.7
10	MCK/2	By P1M0.7 & P1M1.7
11	MCK/4	By P1M0.7 & P1M1.7

Please refer Section "8 System Clock" to get the more detailed clock information. For clock-out on P1.7 function, it is recommended to set P1M0.7 and P1M1.7 to "11" which selects P1.7 as push-pull output mode.

Bit 2: P1FS1~0, P1.1 and P1.0 alternated function selection.

P1FS[1:0]	P1.1	P1.0
00	Reserved	Reserved
01	TXD0	RXD0
10	PWM0S	CEX0
11	T1/T1CKO	T0/T0CKO

AUXR1: Auxiliary Control Register 1

SFR Attribute = Normal Read/Write
SFR Address = 0xA2 RESET = 0000-0000

7	6	5	4	3	2	1	0
INT1IS1	**INT1IS0**	**INT0IS0**	GF	STAF	STOF	PTCKOE	GF
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~6: INT1IS1~0, input selection bits of nINT1, TWI_SCL and TWI_SDA, which function is defined as following table.

INT1IS.1~0	nINT1 & TWI_SCL	TWI_SDA
00	P1.0	P1.1
01	P1.4	P1.5
10	P3.0	P3.1
11	P3.7	P3.6

Bit 5: INT0IS0, nINT0 input selection bits which function is defined as following table.

INT0IS.0	nINT0
0	P3.2
1	P1.6

5. 8051 CPU Function Description

5.1. CPU Register

PSW: Program Status Word

SFR Attribute = Normal Read/Write
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 Attribute = Normal Read/Write
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 Attribute = Normal Read/Write
SFR Address = 0x82 RESET = 0000-0000

7	6	5	4	3	2	1	0
DPL.7	DPL.6	DPL.6	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 Attribute = Normal Read/Write

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 Attribute = Normal Read/Write

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 Attribute = Normal Read/Write

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 **MG74PG1A08** 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 6~7 times the rate of a standard 8051 device). It employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. The instruction timing is different than that of the standard 8051.

In many 8051 implementations, a distinction is made between machine cycles and clock cycles, with machine cycles varying from 2 to 12 clock cycles in length. However, the 1T-80C51 implementation is based solely on clock cycle timing. All instruction timings are specified in terms of clock cycles. For more detailed information about the 1T-80C51 instructions, please refer Section "24 Instruction Set" which includes the mnemonic, number of bytes, and number of clock cycles for each instruction.

5.3. CPU Addressing Mode

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

Indirect Addressing (IND)
In indirect addressing the instruction specified a register which contains the address of the operand. Both internal and external RAM can be indirectly addressed.

The address register for 8-bit addresses can be R0 or R1 of the selected bank, or the Stack Pointer.
The address register for 16-bit addresses can only be the 16-bit data pointer register – DPTR.

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

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

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

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

6. Memory Organization

Like all 80C51 devices, the **MG74PG1A08** has separate address spaces for program and data memory. The logical separation of program and data memory allows the data memory to be accessed by 8-bit addresses, which can be quickly stored and manipulated by the 8-bit CPU.

Program memory (ROM) can only be read, not written to. There can be up to **8K** bytes of program memory. In the **MG74PG1A08**, all the program memory are on-chip OTP memory, and without the capability of accessing external program memory because of no External Access Enable (/EA) and Program Store Enable (/PSEN) signals designed.

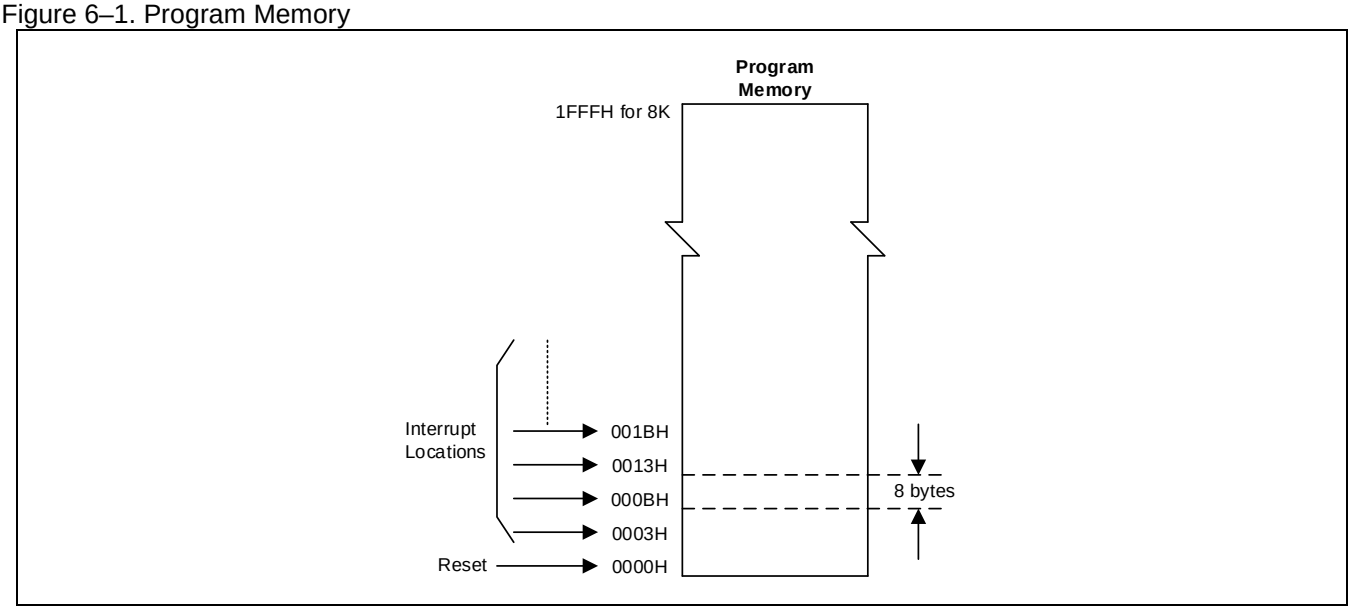
Data memory occupies a separate address space from program memory. In the **MG74PG1A08**, there is only a 256 bytes of internal scratch-pad RAM and has no any expanded RAM (XRAM).

6.1. On-Chip Program OTP

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

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

Figure 6–1. Program Memory



6.2. On-Chip Data RAM

Figure 6–2 shows the internal and external data memory spaces available to the **MG74PG1A08** user. Internal data memory can be divided into three blocks, which are generally referred to as the lower 128 bytes of RAM, the upper 128 bytes of RAM, and the 128 bytes of SFR space. Internal data memory addresses are always 8-bit wide, which implies an address space of only 256 bytes. Direct addresses higher than 7FH access the SFR space; and indirect addresses higher than 7FH access the upper 128 bytes of RAM. Thus the SFR space and the upper 128 bytes of RAM occupy the same block of addresses, 80H through FFH, although they are physically separate entities.

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

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

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

Figure 6–2. Data Memory

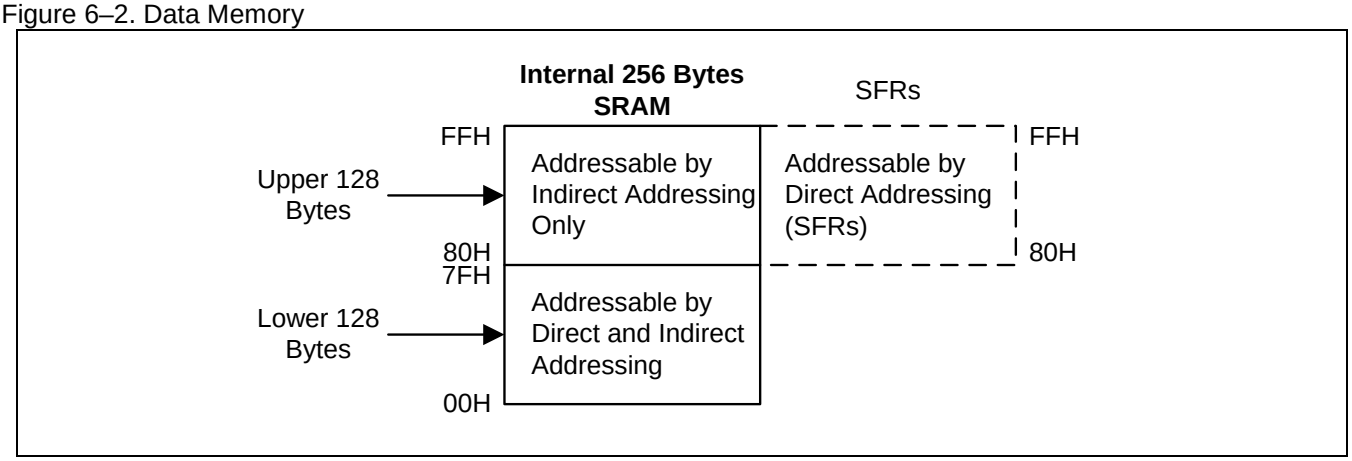


Figure 6–3. Lower 128 Bytes of Internal RAM

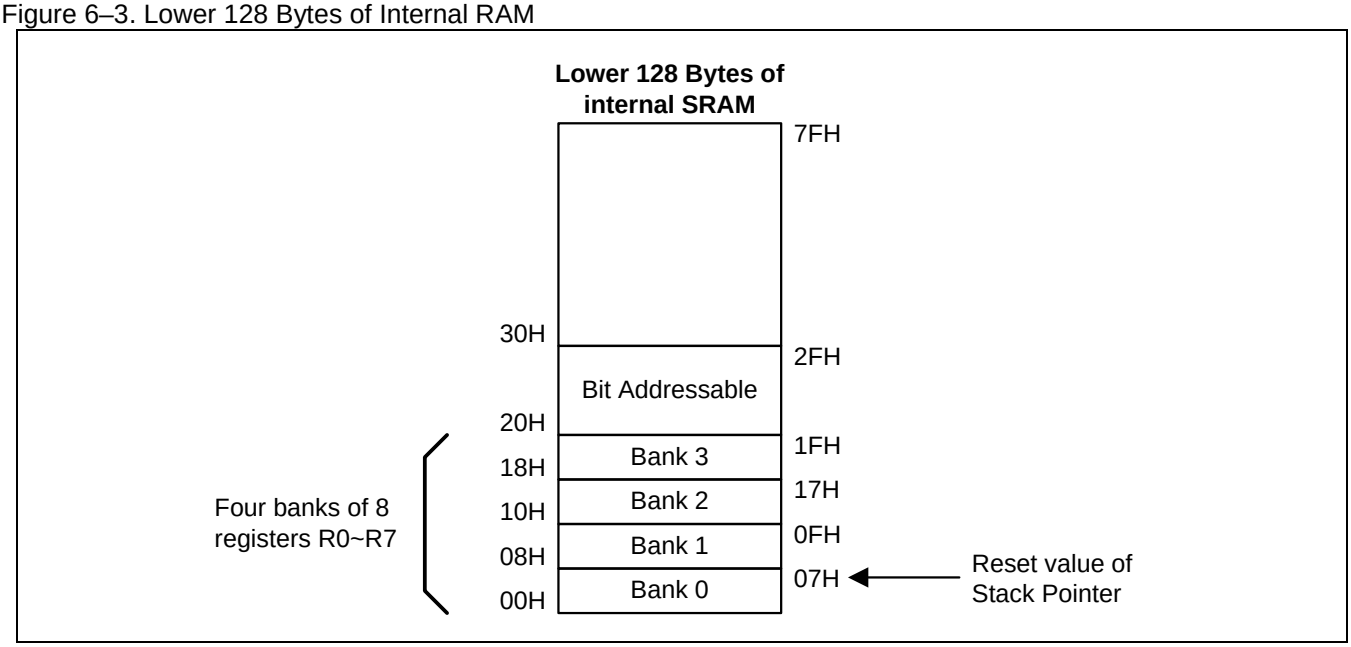
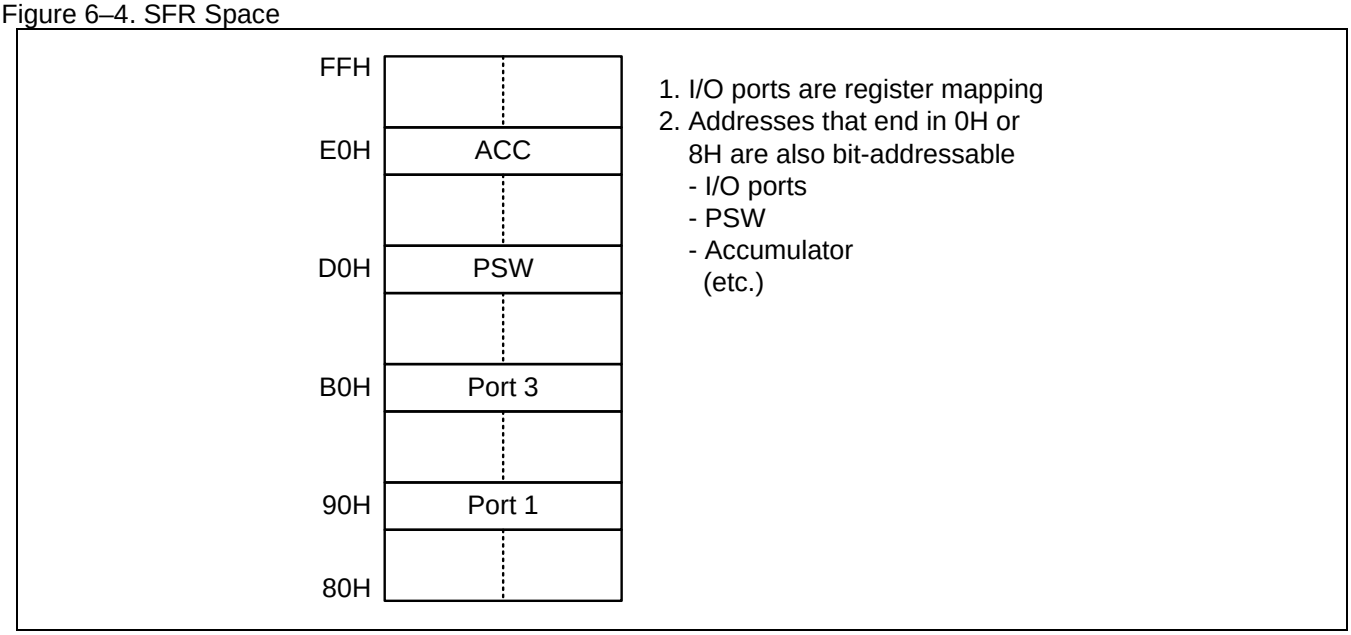


Figure 6–4. SFR Space



6.3. Declaration Identifiers in a C51-Compiler

The declaration identifiers in a C51-compiler for the various **MG74PG1A08** 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
There is no on-chip XRAM or XRAM interface for ***xdata*** access.

pdata
There is no on-chip XRAM or XRAM interface for ***pdata*** access.

code
8K bytes of program memory space; accessed as part of program execution and via the “MOVC @A+DTPR” instruction.

7. Data Pointer Register (DPTR)

There is only one set DPTR in **MG74PG1A08**. **MG74PG1A08** does not support external memory access and MOVX instruction.

8. System Clock

There are three clock sources for the system clock: Internal High-frequency RC Oscillator (IHRCO), Internal Low-frequency RC Oscillator (ILRCO) and External Clock Input. Figure 8–1 shows the structure of the system clock in **MG74PG1A08**.

The **MG74PG1A08** always boots from IHRCO on 12MHz with divided **2** on system clock. CPU clock divider is cascaded after system clock with default divided by 4. Software can select the one of the three clock sources by application required and switches them on the fly. But software needs to settle the clock source stably before clock switching. In external clock input mode (ECKI), the clock source comes from P1.7 input.

The built-in IHRCO provides the high precision frequency at **12MHz** for system clock source. It is the default clock source in **MG74PG1A08** after power-on. To find the detailed IHRCO performance, please refer Section "23.5 IHRCO Characteristics"). In IHRCO mode, P1.7 can be configured to internal *MCK* output or *MCK*/2 and *MCK*/4 for system application.

The **MG74PG1A08** device includes a Clock Multiplier to generate the high speed clock for system clock source. It generates 4/5.33/8 times frequency of CKM, CKM is shown in Figure 8–1 and its typical input is 6MHz. This function provides the high speed operation on MCU without external high-frequency clock input. To find the detailed CKM performance, please refer Section "23.7 CKM Characteristics").

The built-in ILRCO provides the low power and low speed frequency about **64KHz** 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 "23.6 ILRCO Characteristics"). In ILRCO mode, P1.7 can be configured to internal *MCK* output or *MCK*/2 and *MCK*/4 for system application.

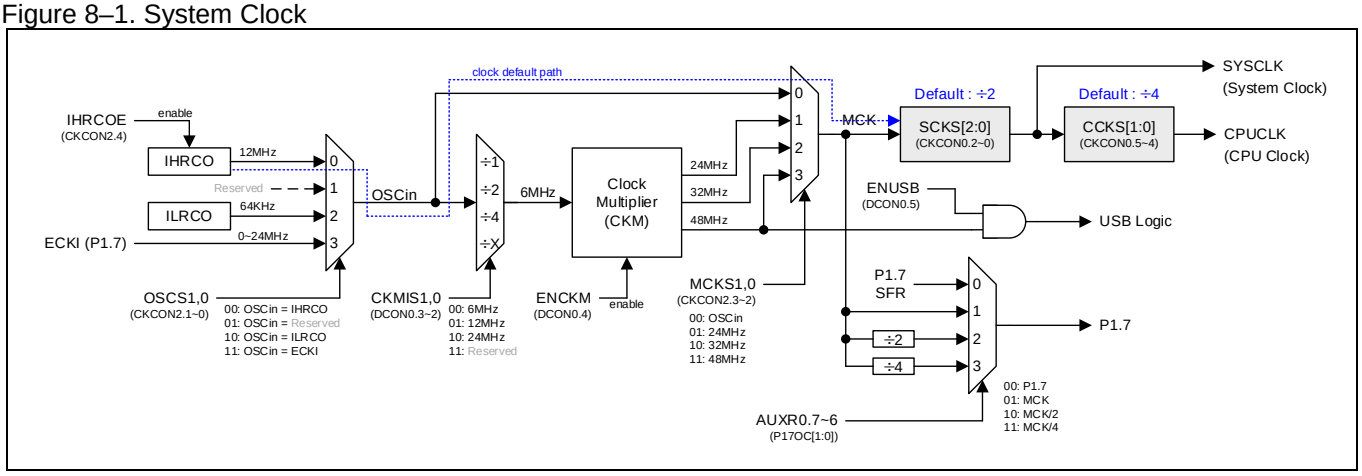
The system clock, *SYSCLK*, is obtained from one of these four clock sources through the clock divider, as shown in Figure 8–1. The user can program the divider control bits SCKS2~SCKS0 (in CKCON0 register) to get the desired system clock. The default system clock divider is set to "***/2***" in **MG74PG1A08** after power on or reset.

The CPU clock, *CPUCLK,* divide from system clock. The CPU clock divider, CCKS.1~CCKS.0 default is set to "***/4***" in **MG74PG1A08** after power on or reset.

8.1. Clock Structure

Figure 8–1 presents the principal clock systems in the **MG74PG1A08**. The system clock can be sourced by the external oscillator circuit or either internal oscillator.

Figure 8–1. System Clock



8.2. Clock Register

CKCON0: Clock Control Register 0

SFR Attribute = Normal Read/Write or Protected Write

SFR Address = 0xC7 RESET = xx10-x001

7	6	5	4	3	2	1	0
--	--	CCKS1	CCKS0	--	SCKS2	SCKS1	SCKS0
W	**W**	**R/W**	**R/W**	**W**	**R/W**	**R/W**	**R/W**

Bit 7~6: Reserved. Software must write “0” on these bits when CKCON0 is written.

Bit 5~4: CCKS1 ~ CCKS0, CPU Clock Selection.

CCKS[1:0]	CPU Clock Selection
0 0	SYSCLK /1
0 1	SYSCLK /2
1 0	SYSCLK /4 **(default)**
1 1	SYSCLK /8

Bit 3: Reserved. Software must write “0” on this bit when CKCON0 is written.

Bit 2~0: SCKS2 ~ SCKS0, programmable System Clock Selection. The default value of SCKS[2:0] is set to “001” to select system clock on OSCin/2.

SCKS[2:0]	System Clock Selection
0 0 0	OSCin /1
0 0 1	OSCin /2 **(default)**
0 1 0	OSCin /4
0 1 1	OSCin /8
1 0 0	OSCin /16
1 0 1	OSCin /32
1 1 0	OSCin /64
1 1 1	OSCin /128

CKCON2: Clock Control Register 2

SFR Attribute = Normal Read and Protected Write

SFR Address = 0xBB RESET =xxx1-0000

7	6	5	4	3	2	1	0
--	--	--	IHRCOE	MCKS1	MCKS0	OSCS1	OSCS0
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 CKCON2 is written.

Bit 4: IHRCOE, Internal High frequency RC Oscillator Enable. The default value is set for MCU clock source.
0: Disable internal high frequency RC oscillator.
1: Enable internal high frequency RC oscillator. If this bit is set by CPU software, it needs 32us to have stable output after IHRCOE enabled.

Bit 3~2: MCKS[1:0], MCK Source Selection.

MCKS[1:0]	MCK Source Selection
0 0	OSCin
0 1	24MHz (ENCKM must be enabled)
1 0	32MHz (ENCKM must be enabled)
1 1	48MHz (ENCKM must be enabled)

Bit 1~0: OSCS[1:0], OSCin source selection. The default selection of OSCin is IHRCO.

OSCS[1:0]	OSCin source Selection
0 0	IHRCO
0 1	Reserved
1 0	ILRCO
1 1	ECKI, External Clock Input (P1.7) as OSCin.

DCON0: Device Control 0

SFR Attribute = Normal Read and Protected Write

SFR Address = 0xBC RESET = 0000-0110

7	6	5	4	3	2	1	0
WCKS	**USBR**	**ENUSB**	**ENCKM**	**CKMIS1**	**CKMIS0**	RSTIO	SWRST
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: WCKS, WDT Clock selection.
0: Select ILRCO for WDT clock source.
1: Select SYSCLK/12 for WDT clock source.

Bit 6: USBR, Software trigger USB block reset
0: Software end the reset of USB block.
1: Software start the reset of USB block.

Bit 5: ENUSB, Enable USB clock and whole USB function.
0: Disable USB clock and USB function.
1: Enable USB clock and USB function.

Bit 4: ENCKM, Enable clock multiplier (X8)
0: Disable the X8 clock multiplier.
1: Enable the X8 clock multiplier.

Bit 3~2: CKMIS1 ~ CKMIS0, Clock Multiplier Input Selection.

CKMIS[1:0]	Clock Multiplier Input Selection
0 0	6MHz input
0 1	12MHz input (default)
1 0	24MHz input
1 1	Reserved.

AUXR0: Auxiliary Register 0

SFR Attribute = Normal Read/Write

SFR Address = 0xA1 RESET = 0000-0000

7	6	5	4	3	2	1	0
P17OC1	**P17OC0**	GF	T0XL	P1FS1	P1FS0	INT1H	INT0H
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~6: P1.7 function configured control bit 1 and 0. The two bits only act when internal RC oscillator (IHRCO or ILRCO) is selected for system clock source. In external clock input mode, P1.7 is the dedicated clock input pin. In internal oscillator condition, P1.7 provides the following selections for GPIO or clock source generator. When P17OC[1:0] index to non-P1.7 GPIO function, P1.7 will drive the on-chip RC oscillator output to provide the clock source for other devices.

P17OC[1:0]	P1.7 function	I/O mode
00	P1.7	By P1M0.7 & P1M1.7
01	MCK	By P1M0.7 & P1M1.7
10	MCK/2	By P1M0.7 & P1M1.7
11	MCK/4	By P1M0.7 & P1M1.7

Please refer Section "8 System Clock" to get the more detailed clock information. For clock-out on P1.7 function, it is recommended to set P1M0.7 and P1M1.7 to "11" which selects P1.7 as push-pull output mode.

9. Watch Dog Timer (WDT)

9.1. WDT Structure

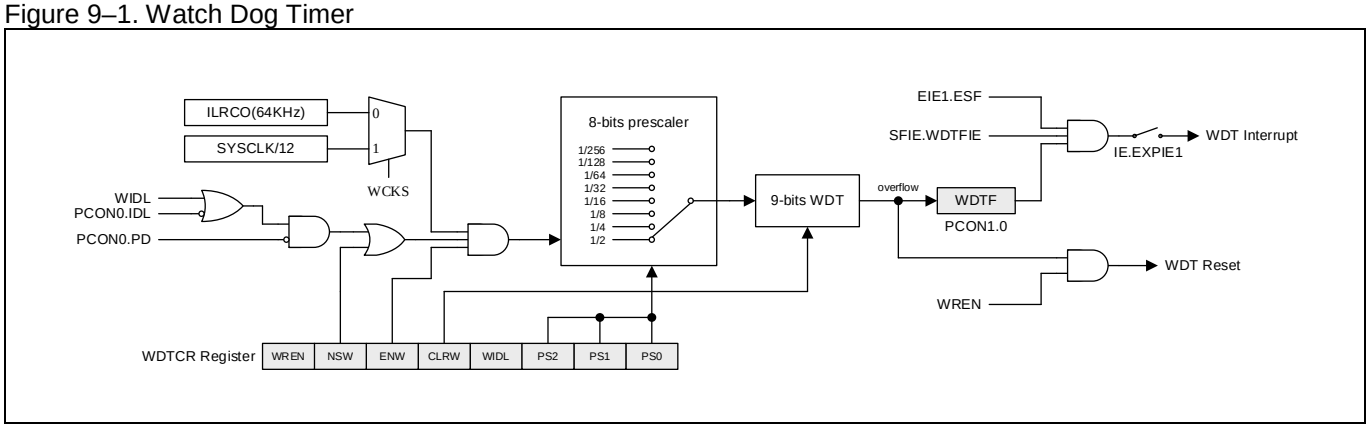
The Watch-dog Timer (WDT) is intended as a recovery method in situations where the CPU may be subjected to software upset. The WDT consists of a 9-bit free-running counter, an 8-bit pre-scaler and a control register (WDTCR). Figure 9–1 shows the WDT structure in **MG74PG1A08**.

When WDT is enabled, it derives its time base from the 64 KHz ILRCO or SYSCLK/12. The WDT overflow will set the WDTF on PCON1.0 which can be configured to generate an interrupt by enabled WDTFIE (SFIE.0) and enabled ESF (XPIE1.0). 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 Protected-Write SFR over-write on ENW, which will clear the ENW bit. The WDTCR register will keep the previous programmed value unchanged after external reset (nRST-pin), software reset and WDT reset.

WREN, NSW and ENW are implemented to one-time-enabled function, only writing "1" valid in general SFR page. Protected-Write SFR Access on WDTCR can disable WREN, NSW and ENW, writing "0" on WDTCR.7~5. Please refer Section "9.3 WDT Register" and Section "20 Protected-Write SFR Access" for more detail information.

Figure 9–1. Watch Dog Timer



9.2. WDT during Idle and Power Down

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.

In the Power down mode, the ILRCO won't stop if the NSW (WDTCR.6) is enabled. 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.

9.3. WDT Register

DCON0: Device Control 0

SFR Attribute = Normal Read and Protected Write
SFR Address = 0xBC RESET = 0000-0110

7	6	5	4	3	2	1	0
WCKS	USBR	ENUSB	ENCKM	CKMIS1	CKMIS0	RSTIO	SWRST
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: WCKS, WDT Clock selection.
0: Select ILRCO for WDT clock source.
1: Select SYSCLK/12 for WDT clock source.

WDTCR: Watch-Dog-Timer Control Register

SFR Attribute = Normal Read/Write or Protected Write

SFR Address = 0xE1 POR = 0000-0000 (xxx0_xxxx by Hardware Option)

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 cannot be cleared by software in normal page. **In protected-write access (protect-write key = 8C), software can modify it to "0" or "1".**

Bit 6: NSW. Non-Stopped WDT. The initial value can be changed by hardware option, NSWDT.

0: WDT stop counting while the MCU is in power-down mode.

1: WDT always keeps counting while the MCU is in power-down mode (Watch Mode) or idle mode. Once NSW has been set, it cannot be cleared by software in normal page. **In protected-write access (protect-write key = 8C), software can modify it to "0" or "1".**

Bit 5: ENW. Enable WDT. The initial value can be changed by hardware option, HWENW.

0: Disable WDT running.

1: Enable WDT while it is set. Once ENW has been set, it cannot be cleared by software in normal page. **In protected-write access (protect-write key = 8C), software can modify it as "0" or "1".**

Bit 4: CLRW. Clear WDT counter.

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

1: Writing "1" to this bit will clear the 9-bit WDT counter to 000H. Note this bit has no need to be cleared by writing "0".

Bit 3: WIDL. WDT idle control. The initial value can be changed by hardware option, HWWIDL.

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 pre-scaler output for WDT time base input.

PS[2:0]	Pre-scaler Value	WDT Period	
		ILRCO(64K)	**SYSCLK(12Mhz)/12**
		WCKS=0	WCKS=1
0 0 0	2	15 ms	1ms
0 0 1	4	30 ms	2ms
0 1 0	8	62 ms	4ms
0 1 1	16	124 ms	8ms
1 0 0	32	248 ms	16ms
1 0 1	64	496 ms	32ms
1 1 0	128	992 ms	64ms
1 1 1	256	1.984 S	128ms

PCON1: Power Control Register 1

SFR Attribute = Normal Read/Write or Protected Write

SFR Address = 0x97 POR = 00xx-0000

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

Bit 0: 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.

9.4. WDT Hardware Option

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

If HWENW is programmed to “enabled”, then hardware will automatically do the following initialization for the WDTCR register at power-up: (1) set ENW bit, (2) load WRENO into WREN bit, (3) load NSWDT into NSW bit, (4) load HWWIDL into WIDL bit, 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 Protected-Write 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, ENW, WIDL, PS2, PS1 and PS0 in WDTCR, will be write-protected.
☐: Disabled. The WDT SFRs, WREN, NSW, ENW, WIDL, PS2, PS1 and PS0 in WDTCR, are free for writing of software.

10. System Reset

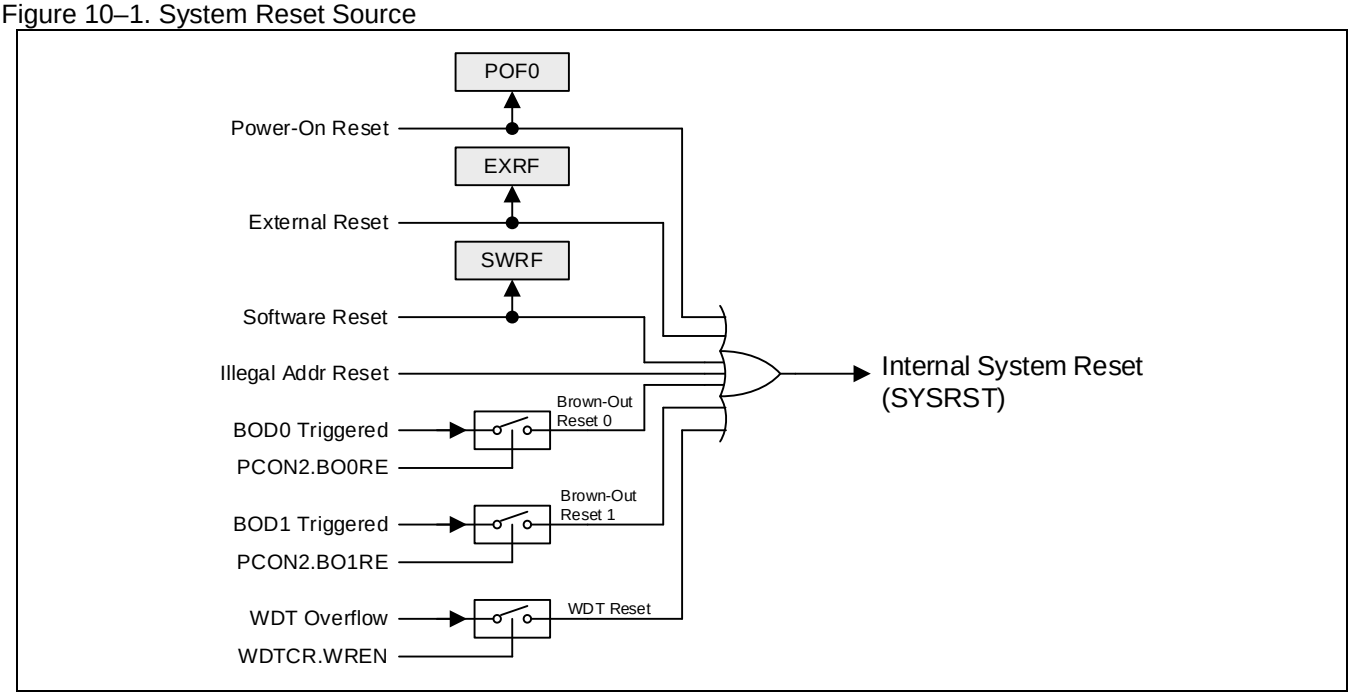
During reset, all I/O Registers are set to their initial values, and the program starts execution from the Reset Vector, 0000H. The **MG74PG1A08** has seven sources of reset: power-on reset, external reset, software reset, illegal address reset, WDT reset and brown-out 0/1 reset. Figure 10–1 shows the system reset source in **MG74PG1A08**.

The following sections describe the reset happened source and corresponding control registers and indicating flags. (IAR only acts reset on CPU)

10.1. Reset Source

Figure 10–1 presents the reset systems in the **MG74PG1A08** and all of its reset sources.

Figure 10–1. System Reset Source



10.2. 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 Attribute = Normal Read/Write or Write-protected

SFR Address = 0x87 POR = 0001-0000, RESET = 000X-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 4: POF. Power-On Flag.
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. POF can also be set by software.

The Power-on Flag, POF, 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 and WDT reset. It helps users to check if the running of the CPU begins from power up or not. Note that the POF must be cleared by firmware.

10.3. External Reset

PCON1: Power Control Register 1

SFR Attribute = Normal Read/Write or Protected Write
SFR Address = 0x97 POR = 00xx-0000

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

Bit 6: EXRF, External Reset Flag.
0: This bit must be cleared by software writing **“1”** on it. Software writing “:0” is no operation.
1: This bit is only set by hardware if an External Reset occurs. Writing **“1”** on this bit will clear EXRF.

DCON0: Device Control 0

SFR Attribute = Normal Read and Protected Write
SFR Address = 0xBC RESET = 0000-0110

7	6	5	4	3	2	1	0
WCKS	USBR	ENUSB	ENCKM	CKMIS1	CKMIS0	**RSTIO**	SWRST
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 1: RSTIO, nRST function on I/O,
1: Select I/O pad function for nRST.
0: Select I/O pad function for GPIO.

10.4. Software Reset

Software can trigger the CPU to restart by software reset, writing “1” on SWRST (DCON0.0), and set the SWRF flag (PCON1.7).

DCON0: Device Control 0

SFR Attribute = Normal Read and Protected Write
SFR Address = 0xBC RESET = 0000-0110

7	6	5	4	3	2	1	0
WCKS	USBR	ENUSB	ENCKM	CKMIS1	CKMIS0	RSTIO	**SWRST**
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 0: SWRST, software reset trigger control.
0: No operation
1: Generate software system reset. It will be cleared by hardware automatically.

PCON1: Power Control Register 1

SFR Attribute = Normal Read/Write or Protected Write
SFR Address = 0x97 POR = 00xx-0000

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

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

10.5. Brown-Out Reset

In **MG74PG1A08**, there are one Power on reset (POR) and two Brown-Out Detectors (BOD0 & BOD1) to monitor VDD power. POR detects the VDD level by software selecting 1.95V or 2.3V. BOD0 detects the VDD level by software selecting 2.1V or 2.6V. BOD1 services the fixed detection level at VDD=3.6V. If VDD power drops below POR, BOD0 or BOD1 monitor level. Associated flag, BOF0 and BOF1, is set. If BO0RE (PCON2.2) is enabled, BOF0 indicates a BOD0 Reset occurred. If BO1RE (PCON2.3) is enabled, BOF1 indicates a BOD1 Reset occurred.

PCON2: Power Control Register 2

SFR Attribute = Normal Read and Protected Write
SFR Address = 0xBA POR/RESET = 0100-0000

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

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

Bit 6: EBOD1, Enable BOD1 that monitors VDD power dropped below 3.6V.
0: Disable BOD1 to slow down the chip power consumption.
1: Enable BOD1 to monitor VDD power dropped.

Bit 5~4: Reserved. Software must write “0” on these bits when PCON2 is written.

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: 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 2.1V or 2.6V).

Bit 1: Reserved. Software must write “0” on these bits when PCON2 is written.

Bit 0: RMLS, Power on Reset (POR) and Brown-Out detector 0(BOD0) monitored level Selection. The initial values of this bit is loaded from RMLSO.

RMLS	POR detecting level	BOD0 detecting level
0	1.95V	2.1V
1	2.3V	2.6V

PCON1: Power Control Register 1

SFR Attribute = Normal Read/Write or Protected Write
SFR Address = 0x97 POR = 00xx-0000

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

Bit 2: BOF1, BOF1 (Reset) Flag.
0: This bit must be cleared by software writing **“1”** on it. Software writing “:0” is no operation.
1: This bit is only set by hardware when VDD meets BOD1 monitored level. Writing **“1”** on this bit will clear BOF1. If BO0RE (PCON2.3) is enabled, BOF0 indicates a BOD0 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 (PCON2.2) is enabled, BOF0 indicates a BOD0 Reset occurred.

10.6. 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 Attribute = Normal Read/Write or Protected Write
SFR Address = 0x97 POR = 00xx-0000

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

Bit 0: WDTF, WDT Overflow/Reset Flag.
0: This bit must be cleared by software writing **"1"** on it. Software writing ":0" is no operation.
1: This bit is only set by hardware when WDT overflows. Writing **"1"** on this bit will clear WDTF. If WREN (WDTCR.7) is set, WDTF indicates a WDT Reset occurred.

WDTCR: Watch-Dog-Timer Control Register

SFR Attribute = Normal Read/Write or Protected Write
SFR Address = 0xE1 POR = 0000-0000 (xxx0_xxxx by Hardware Option)

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 cannot be cleared by software in normal page. **In protected-write mode, software can modify it to "0" or "1".**

11. Power Management

The **MG74PG1A08** supports two power monitor modules, Brown-Out Detector 0 (BOD0) and Brown-Out Detector 1 (BOD1), and 6 power-reducing modes: Idle mode, Power-down mode, Slow mode, Sub-Clock mode, Watch mode and Monitor mode.

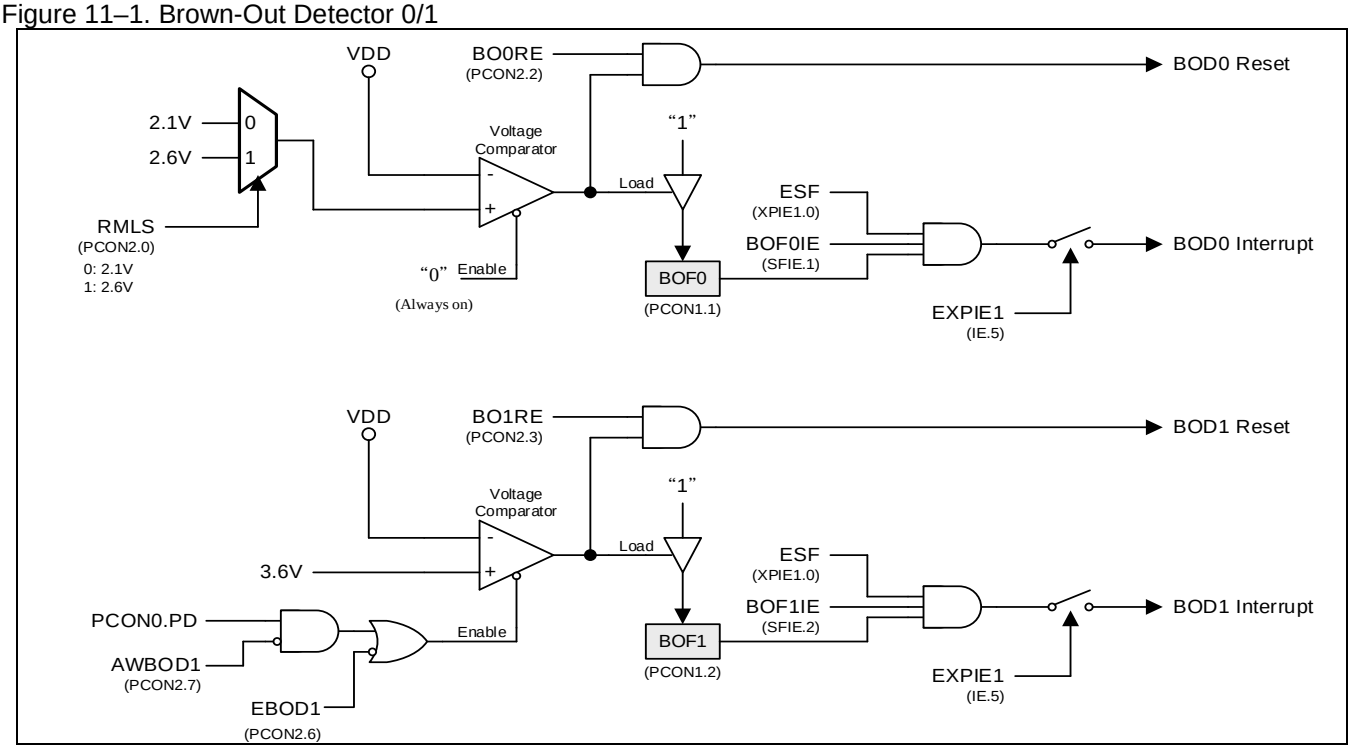
BOD0 and BOD1 report the chip power status on the flags, BOF0 and BOF1, which provide the capability to interrupt CPU or to reset CPU by software configured. The six power-reducing modes provide the different power-saving scheme for chip application. These modes are accessed through the CKCON0, CKCON2,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

11.1. Brown-Out Detector

In **MG74PG1A08**, there are two Brown-Out Detectors (BOD0 & BOD1) to monitor VDD power. Figure 11–1 shows the functional diagram of BOD0 and BOD1. BOD0 detects the software selection levels (2.1V/2.6V) and BOD1 services the fixed detection level at VDD=3.6V on Core Power. Associated flag, BOF0 (PCON1.1), is set when BOD0 meets the detection level. If both of EXPIE1 (IE5), ESF (XPIE1.0) 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 BOD0 and BOD1 interrupt also wake up CPU in power down mode if AWBOD1 (PCON2.7) is enabled.

If BO0RE (PCON2.2) 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 also restart CPU in power down mode if AWBOD1 (PCON2.7) is enabled.

Figure 11–1. Brown-Out Detector 0/1



11.2. Power Saving Mode

11.2.1. Slow Mode

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

11.2.2. Sub-Clock Mode

The alternative to slow down the MCU's operating speed by programming OSCS1~0 can select the ILRCO for system clock. The 64 KHz ILRCO provides the MCU to operate in an ultra-low speed and low power operation. Additional programming SCKS2~SCKS0 bits (in CKCON0 register, see Section "8 System Clock"), the user could put the MCU speed down to 500Hz slowest.

11.2.3. Watch Mode

If Watch-Dog-Timer is enabled and NSW is set, Watch-Dog-Timer will keep running in power down mode, which named Watch Mode in **MG74PG1A08**. 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 "13 Interrupt" for more detail information.

11.2.4. Monitor Mode

The BOD0 always keep VDD monitor in power down mode. It is the Monitor Mode in **MG74PG1A08**. When BOD0 meets the detection level, set BOF0 and wakeup CPU from interrupt or system reset by software configured. Please refer Section "11.1 Brown-Out Detector" and Section "13 Interrupt" for more detail information.

11.2.5. Idle Mode

Setting the IDL bit in PCON enters idle mode. Idle mode halts the internal CPU clock. The CPU state is preserved in its entirety, including the RAM, stack pointer, program counter, program status word, and accumulator. The Port pins hold the logical states they had at the time that Idle was activated. Idle mode leaves the peripherals running in order to allow them to wake up the CPU when an interrupt is generated. Timer 0, Timer 1, UART0, KBI and the BOD0 will continue to function during Idle mode. The PCA Timer and WDT is 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.

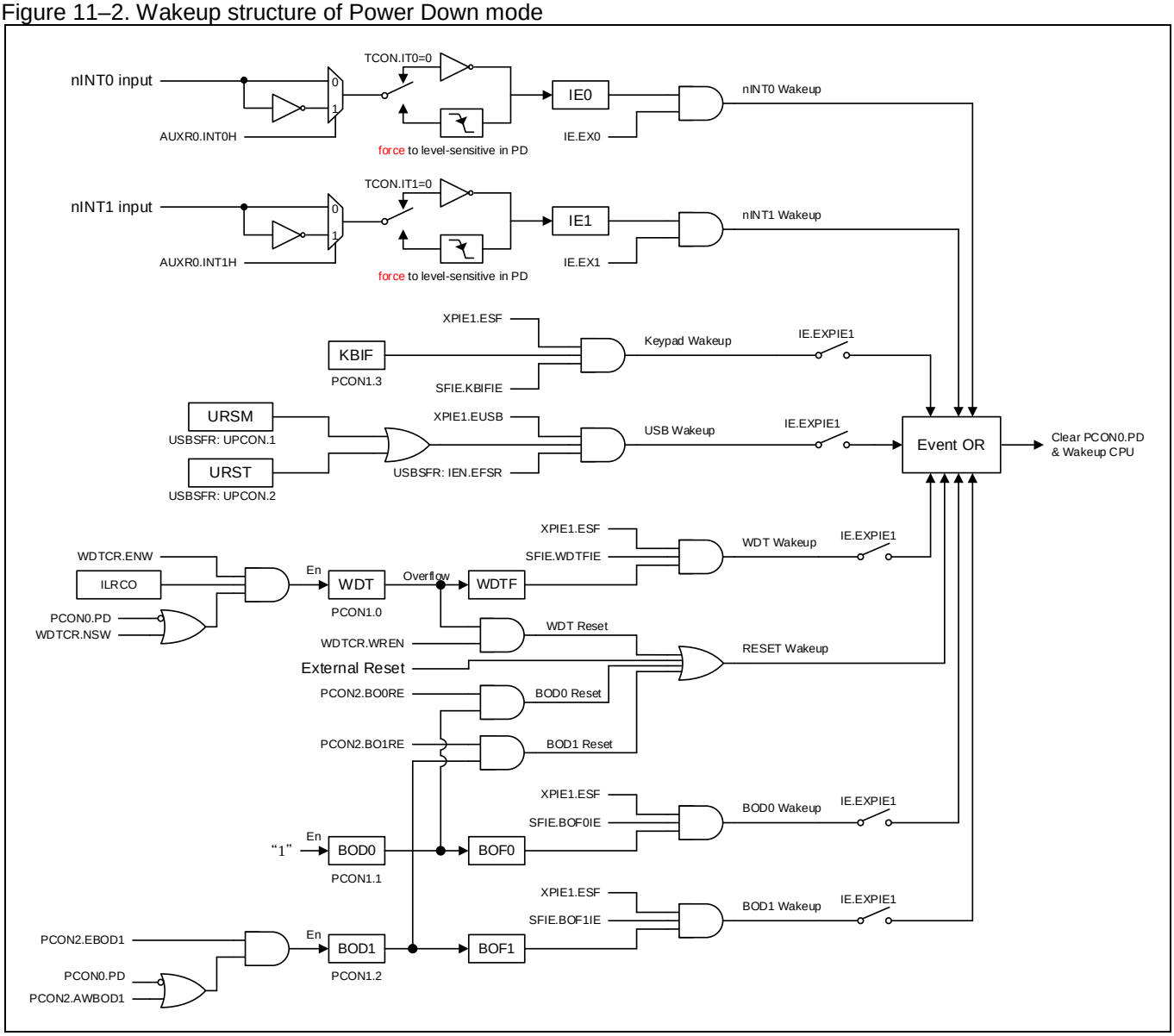
11.2.6. Power-down Mode

Setting the PD bit in PCON enters Power-down mode. Power-down mode stops the oscillator and power down the OTP 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 exit by external reset, enabled external interrupts, enabled KBI GPIOs, enabled USB, enabled BOD0/BOD1 (Monitor mode) or enabled Non-Stop WDT (Watch mode).

The user should not attempt to enter (or re-enter) the power-down mode for a minimum of 4 µs until after one of the following conditions has occurred: Start of code execution (after any type of reset), or Exit from power-down mode.

Figure 11–2 shows the wakeup mechanism of power-down mode in **MG74PG1A08**.

Figure 11–2. Wakeup structure of Power Down mode



11.2.7. Interrupt Recovery from Power-down

Two external interrupts may be configured to terminate Power-down mode. External interrupts nINT0 (P3.2/P1.6), nINT1 (P1.0/P1.4/P3.0/P3.7) may be used to exit Power-down. To wake up by external interrupt nINT0, nINT1, the interrupt must be enabled and configured for level-sensitive operation. If the enabled external interrupts are configured to edge-sensitive operation (Falling or Rising), they will be forced to level-sensitive operation (Low level or High level) by hardware in power-down mode.

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

11.2.8. Reset Recovery from Power-down

If P1.7 is configured for nRST input pin, wakeup from Power-down through an external reset is similar to the interrupt. At the rising edge of nRST, Power-down is exited, the oscillator is restarted, and an internal timer begins counting. The internal clock will not be allowed to propagate to the CPU until after the timer has reached internal counter full. The nRST pin must be held high for longer than the timeout period to ensure that the device is reset properly. The device will begin executing once nRST is brought low.

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

11.2.9. KBI wakeup Recovery from Power-down

All of the GPIOs of **MG74PG1A08**, P1.7 ~ P1.0, P3.7~P3.6 and P3.2~P3.0 have wakeup CPU capability that are nibble enabled by the control bit in KBIEN0 and the associated port pin is configured to digital input only mode. Please refer Section “12.1.5 Port 3 Digital-Input-Only (High Impedance Input) Structure” and Section “12.1.11 Port 3 Digital-Input-Only (High Impedance Input) Structure” for the input mode configuration.

Wakeup from Power-down through an enabled KBI GPIO is similar to the interrupt. At the low-level of enabled KBI GPIO, 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. Please refer Section **“**17 Keypad Interrupt (KBI)**”** for more detail information.

11.2.10. USB wakeup Recovery from Power-down

If USB function is enabled and USB host is connected to **MG74PG1A08**, USB host reset and resume event will wake up CPU from power down mode. Wakeup from Power-down through enabled USB function (DCON0.5) and enabled USB interrupt (IEN.2 in USB SFR) is same to the interrupt. At the active USB interrupt, 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.

11.3. Power Control Register

PCON0: Power Control Register 0

SFR Attribute = Normal Read/Write or Protected Write
SFR Address = 0x87 POR = 0001-0000, RESET = 000x-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 4: POF, Power-On Flag.
0: This bit must be cleared by software writing **"1"** 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 Attribute = Normal Read/Write or Protected Write
SFR Address = 0x97 POR = 00xx-0000

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

Bit 7: SWRF, Software Reset Flag.
0: This bit must be cleared by software writing **"1"** to it.
1: This bit is set by hardware if a Software Reset occurs.

Bit 6: EXRF, External Reset Flag.
0: This bit must be cleared by software writing **"1"** to it.
1: This bit is set by hardware if an External Reset occurs.

Bit 5~4: Reserved. Software must write "0" on these bits when PCON1 is written.

Bit 3: KBIF, KBI Flag.
0: This bit must be cleared by software writing **"1"** to it.
1: This bit is set by hardware if a KBI input event occurs.

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 (3.6V).

Bit 1: BOF0, Brown-Out Detector 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 (2.1V/2.6V).

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 Attribute = Normal Read and Protected Write

SFR Address = 0xBA POR/RESET = 0100-0000

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

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

Bit 6: EBOD1, Enable BOD1 that monitors VDD power dropped below 3.6V.
0: Disable BOD1 to slow down the chip power consumption.
1: Enable BOD1 to monitor VDD power dropped.

Bit 5~4: Reserved. Software must write “0” on these bits when PCON2 is written.

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: 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 2.1V or 2.6V).

Bit 1: Reserved. Software must write “0” on these bits when PCON2 is written.

Bit 0: RMLS, Power on Reset (POR) and Brown-Out detector 0(BOD0) monitored level Selection. The initial values of these two bits are loaded from OR1.RMLSO.

RMLS	POR detecting level	BOD0 detecting level
0	1.95V	2.1V
1	2.3V	2.6V

12. Configurable I/O Ports

The **MG74PG1A08** has following I/O ports: P1.0~P1.7, P3.0~P3.2 and P3.6~P3.7. **nRST** pin has a swapped function on **P1.7**. The exact number of I/O pins available depends upon the package types. See Table 12–1.

Table 12–1. Number of I/O Pins Available

Package Type	I/O Pins	Number of I/O ports
16-pin SOP	P1.0~P1.7, P3.0~P3.2, P3.6, P3.7, P1.7 (nRST/ECKI/ICKO)	15 (nRST/ECKI selected) or 14 (USB DP/DM selected) or 13 (nRST/ECKI or USB selected)

12.1. IO Structure

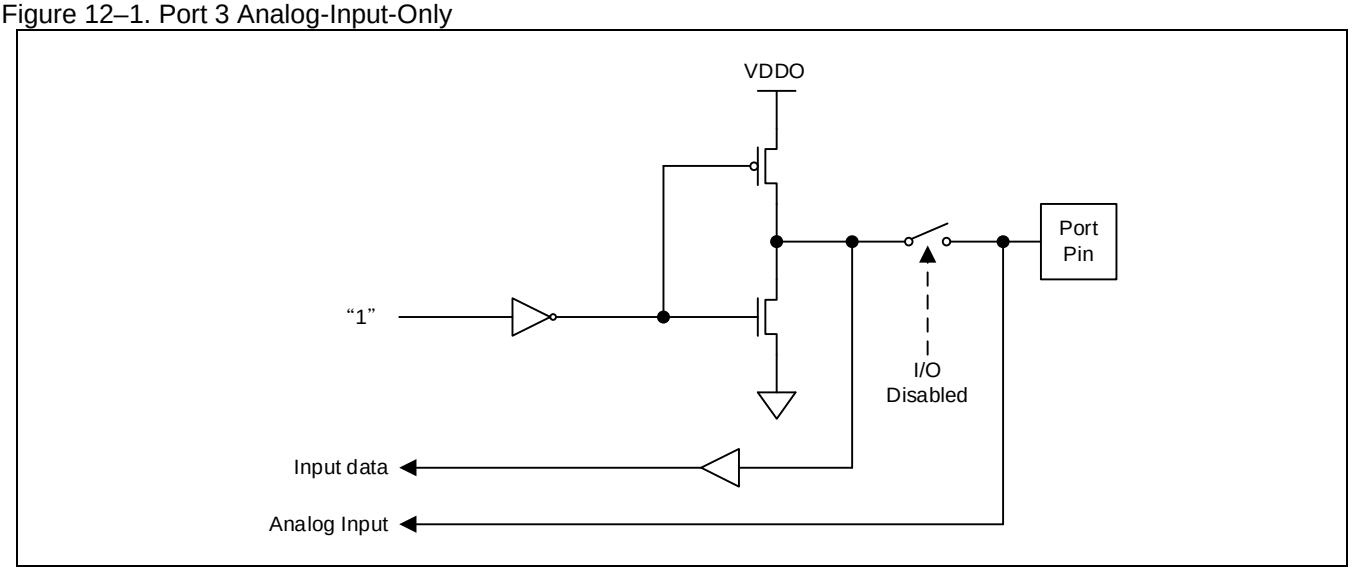
The I/O operating modes are distinguished two groups in **MG74PG1A08**. The first group is only for Port 3 to support four configurations on I/O operating. These are: analog-input-only (high-impedance), quasi-bidirectional (standard 8051 I/O port), open-drain output, and push-pull output. The default setting of port 3 is quasi-bidirectional mode. P3.7 and P3.6 only support open-drain output mode and are shared with USB DP/DM pins.

All other general port pins belong to the second group. They can also be programmed to four operating modes, analog-input-only (high-impedance), 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 mode, which means the port pins in high impedance state after power-on or any reset.

Followings describe the configuration of the all types I/O mode.

12.1.1. Port 3 Analog-Input-Only (High Impedance) Structure

Figure 12–1. Port 3 Analog-Input-Only



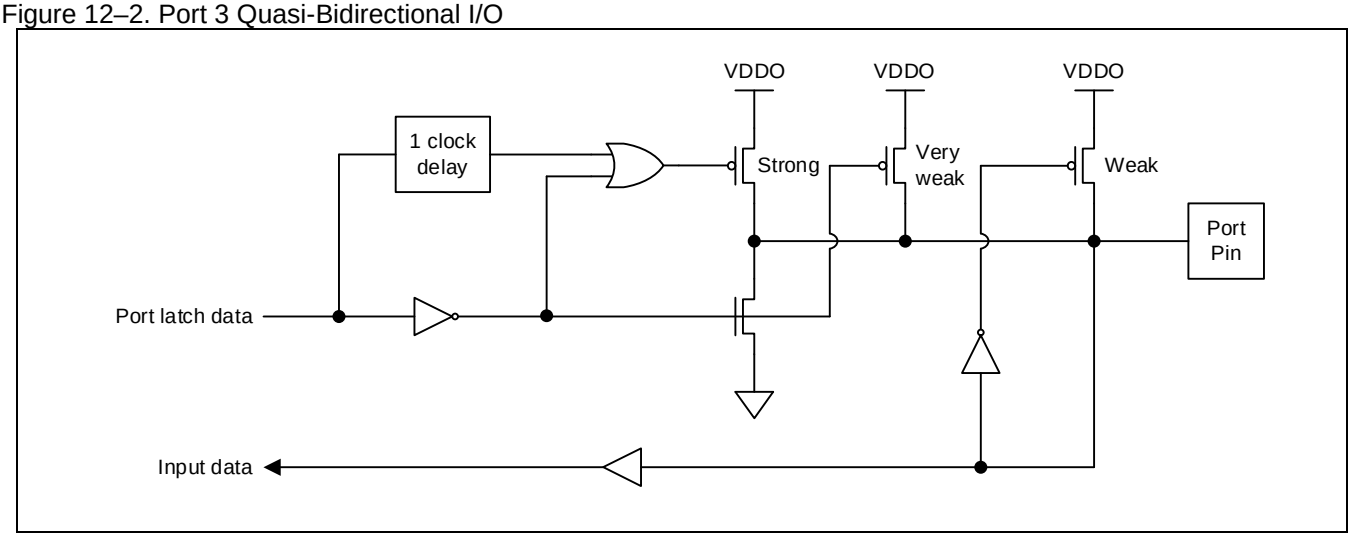
12.1.2. Port 3 Quasi-Bidirectional IO Structure (default)

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 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 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 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 logic “0” to 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 on Port 3 is shown in Figure 12–2.

Figure 12–2. Port 3 Quasi-Bidirectional I/O

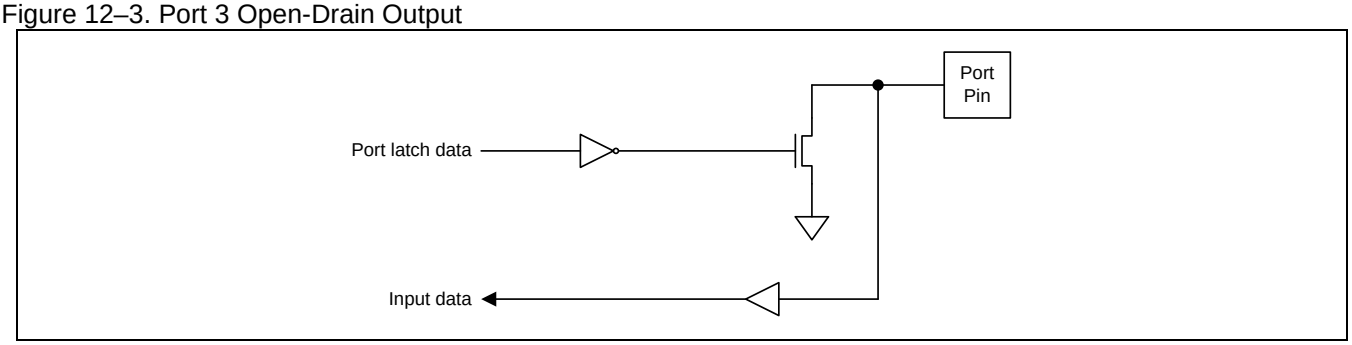


12.1.3. Port 3 Open-Drain Output Structure

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 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 Port 3 open-drain port configuration is shown in Figure 12–3.

Figure 12–3. Port 3 Open-Drain Output

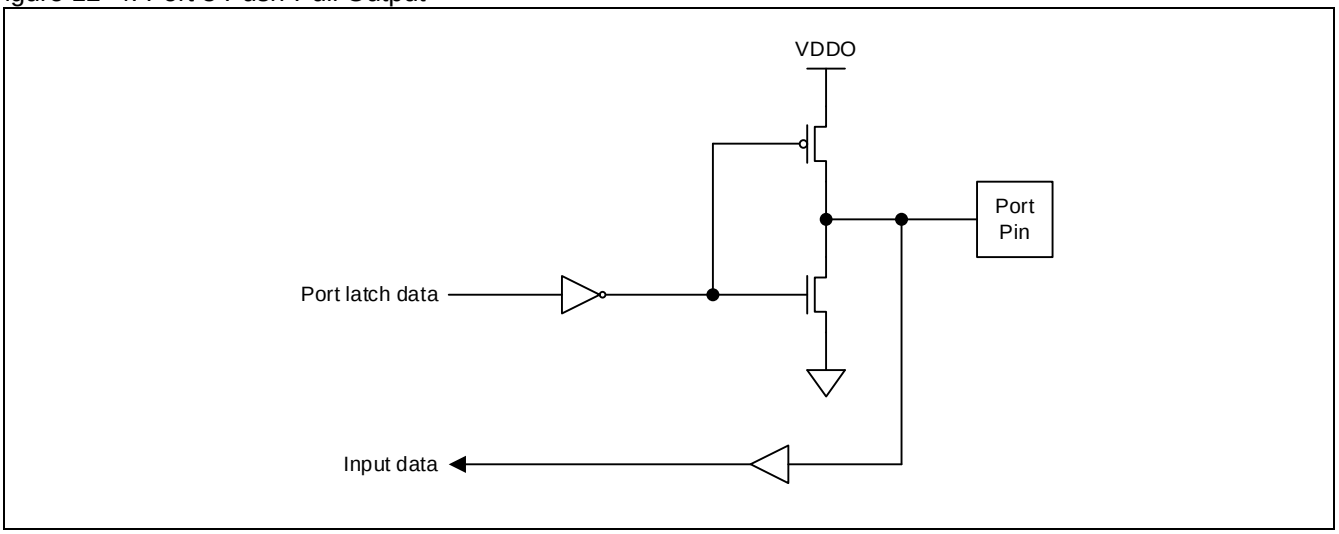


12.1.4. Port 3 Push-Pull Output Structure

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 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 on Port 3 is shown in Figure 12–4.

Figure 12–4. Port 3 Push-Pull Output



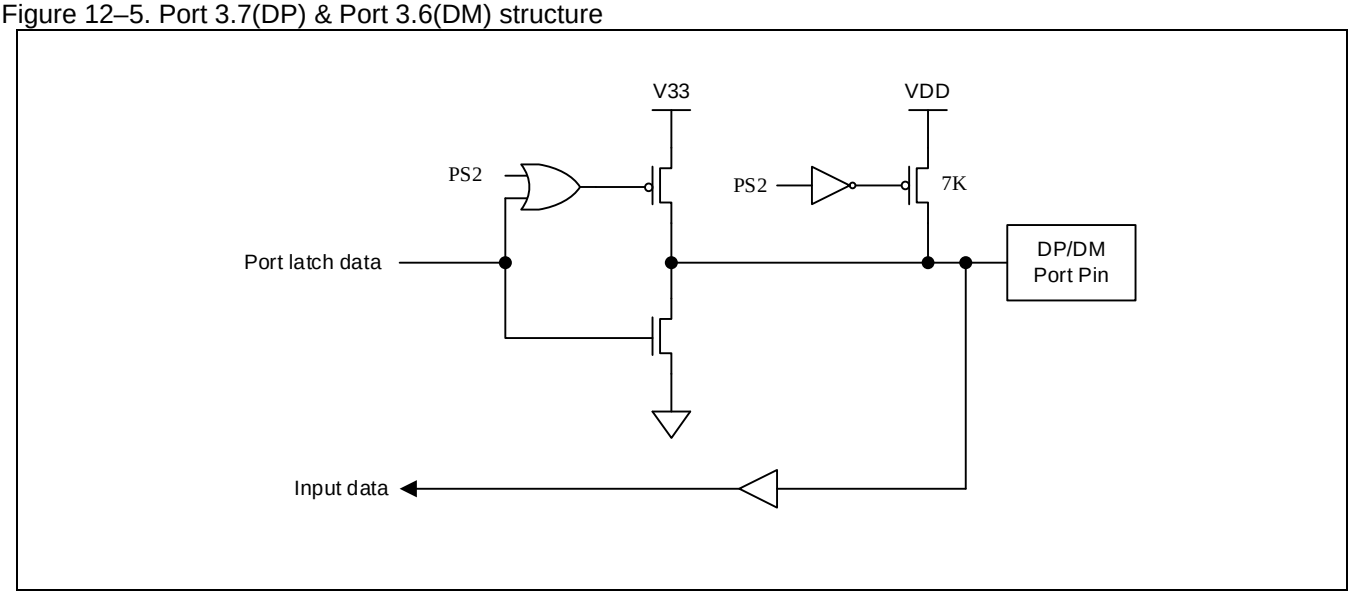
12.1.5. Port 3 Digital-Input-Only (High Impedance Input) Structure

A Port pin is configured as a digital input by setting its output mode to “Open-Drain” and writing logic “1” to the associated bit in the Port Latch Data register. For example, pin P3.2 is configured as a digital input by setting P3M0.2 and P3M1.2 to “01” as open-drain mode and setting P3.2 Port Latch Data to logic 1. Then, the digital-input-only configuration on P3.2 is an input without any pull-up resistors on the port pin. If a pull-up resistor is necessary in the application, software must configure the port pin to quasi-bidirectional I/O mode which enables the on-chip pull-up resistor.

12.1.6. DP/DM Structure

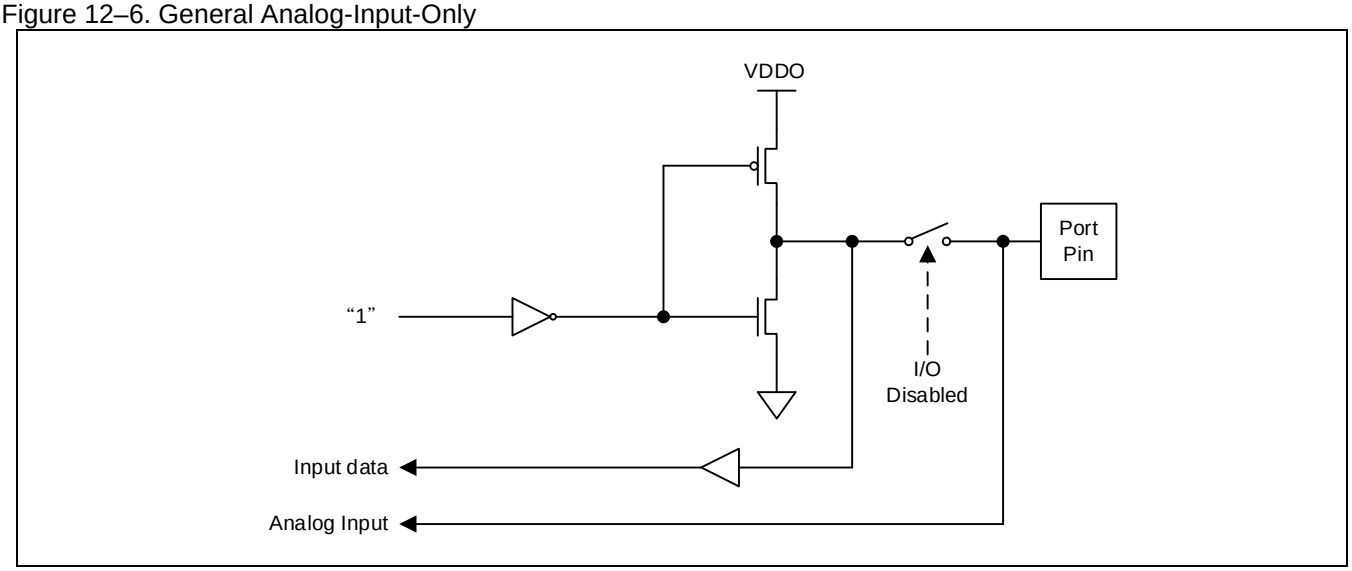
DP & DM input/output structure is shown in Figure 12–5.

Figure 12–5. Port 3.7(DP) & Port 3.6(DM) structure



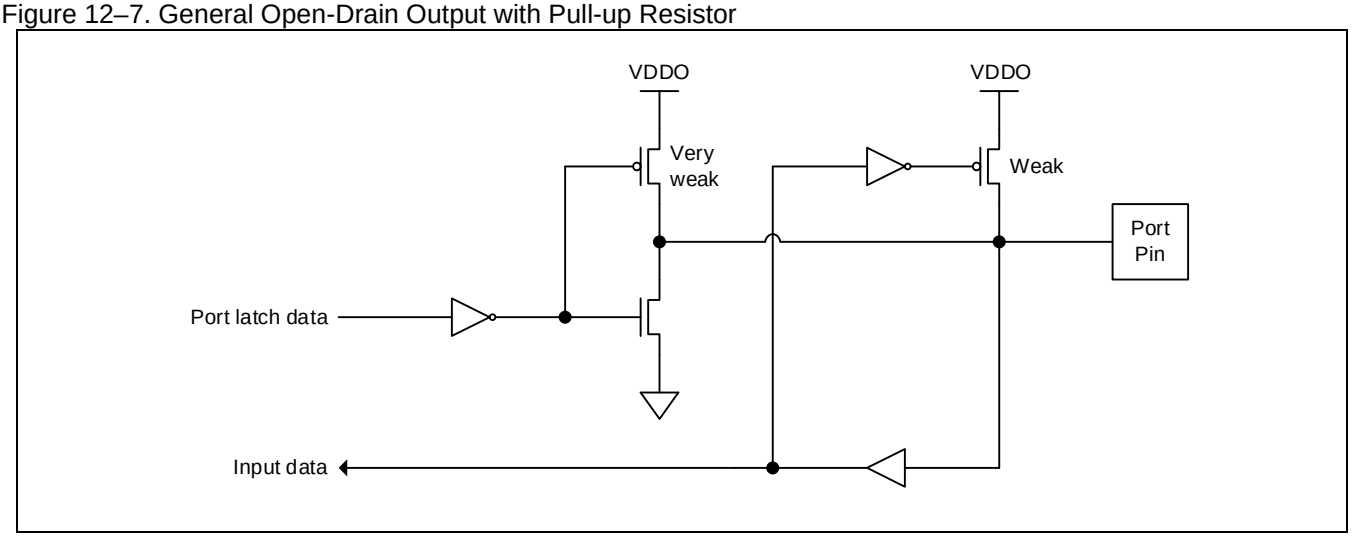
12.1.7. General Analog-Input-Only (High Impedance) Structure (default)

Figure 12–6. General Analog-Input-Only



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

Figure 12–7. General Open-Drain Output with Pull-up Resistor

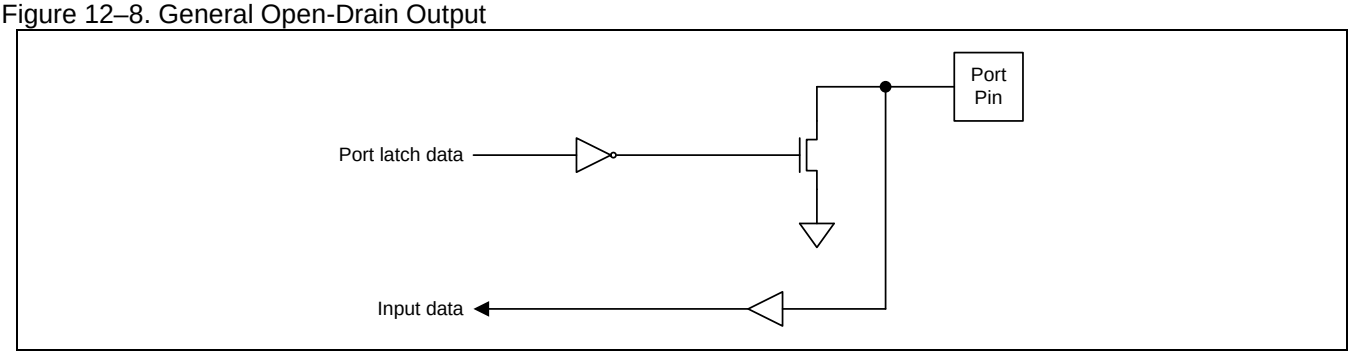


12.1.9. General Open-Drain Output Structure

The open-drain output configuration on general port pins only drives the pull-down transistor of the port pin when the Port Data register contains logic “0”.

The general open-drain port configuration is shown in Figure 12–8.

Figure 12–8. General Open-Drain Output

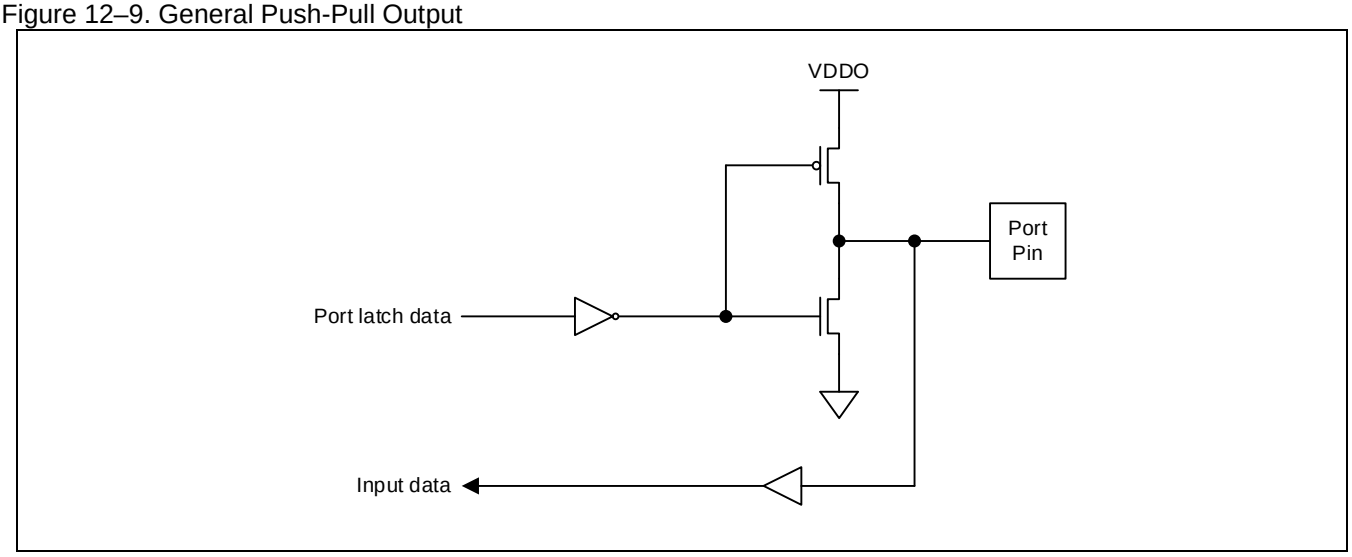


12.1.10. General Push-Pull Output Structure

The push-pull output configuration on general port pins has the same pull-down structure as the open-drain output modes, but provides a continuous strong pull-up when the port register contains 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 open-drain mode.

The push-pull port configuration is shown in Figure 12–9.

Figure 12–9. General Push-Pull Output



12.1.11. General Digital-Input-Only (High Impedance Input) Structure

A Port pin is configured as a digital input by setting its output mode to “Open-Drain” and writing logic “1” to the associated bit in the Port Latch Data register. For example, P1.2 is configured as a digital input by setting P1M0.2 and P1M1.2 to “01” as open-drain mode and setting P1.2 Port Latch Data to logic 1. Then, the digital-input-only configuration on P1.2 is an input without any pull-up resistors on the port pin. If a pull-up resistor is necessary in the application, software must configure the port pin to open-drain output with pull-up resistor mode, {P1M0.2, P1M1.2} = “10”, which enables the on-chip pull-up resistor.

12.2. I/O Port Register

All I/O port pins on the **MG74PG1A08** may be individually and independently configured by software to select its operating mode. Port 3 has four operating modes, as shown in Table 12–2. Two mode registers select the I/O type for each port 3 pin. Only port 3 supports quasi-bidirectional mode and setting them to quasi-bidirectional mode after power-on or any reset.

Table 12–2. Port 3 Configuration Settings

P3M0.y	P3M1.y	Port Mode
0	0	Analog Input Only
0	1	Open-Drain Output (support digital-input-only)
1	0	Quasi-Bidirectional **(default, with pull-up)**
1	1	Push-Pull Output

Where y=0~2 (port pin). The registers P3M0 and P3M1 are listed in each port description.

Table 12–3. Port 37 Configuration Settings

P3M0.**7**	P3M1.**7**	Port Mode
0	0	Input Only **(default)**
0	1	XCVR power-down
1	0	Open-Drain Output with pull-up (PS2 mode)
1	1	Reserve

Other general port pins also support four operating modes, as shown in Table 12–4. Two mode registers select the I/O type for each port pin and setting to analog-input-only on these port pins after power-on or any reset.

Table 12–4. General Port Configuration Settings

PxM0.y	PxM1.y	Port Mode
0	0	Analog-Input-Only **(default)**
0	1	Open-Drain Output (support digital-input-only)
1	0	Open-Drain Output with Pull-Up resistor
1	1	Push-Pull Output

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

12.2.1. Port 1 Register

P1: Port 1 Register

SFR Attribute = Normal Read/Write
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: P1.7~P1.0 could be only set/cleared by CPU.

P1M0: Port 1 Mode Register 0

SFR Attribute = Normal Read/Write
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**

P1M0: Port 1 Mode Register 1

SFR Attribute = Normal Read/Write

SFR Address = 0x92 RESET = 0000-0000

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**

12.2.2. Port 3 Register

P3: Port 3 Register

SFR Attribute = Normal Read/Write

SFR Address = 0xB0 RESET = 11xx-x111

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

Bit 7~0: P3.7~P3.0 could be only set/cleared by CPU.

P3M0: Port 3 Mode Register 0

SFR Attribute = Normal Read/Write

SFR Address = 0xB1 RESET = 0xxx-x111

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

P3M1: Port 3 Mode Register 1

SFR Attribute = Normal Read/Write

SFR Address = 0xB2 RESET = 0xxx-x000

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

13. Interrupt

The **MG74PG1A08** has 6 interrupt sources with a two-level interrupt structure. There are several SFRs associated with the two-level interrupt. They are the IE, IP0L and XPIE1. The IP0L (Interrupt Priority 0 Low) register makes the two-level interrupt structure possible. The two priority level interrupt structure allows great flexibility in handling these interrupt sources.

13.1. Interrupt Structure

Table 13–1 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 IE register) should have been set previously. The 'Request Bits' can be set or cleared by software, with the same result as though it had been set or cleared by hardware. That is, interrupts can be generated or pending interrupts can be cancelled in software. The 'Priority Bits' determine the priority level for each interrupt. The 'Priority within Level' is the polling sequence used to resolve simultaneous requests of the same priority level. The 'Vector Address' is the entry point of an interrupt service routine in the program memory.

Figure 13–1 shows the interrupt system. Each of these interrupts will be briefly described in the following sections.

Table 13–1. Interrupt Sources

No	Source Name	Enable Bit	Request Bits	Priority Bits	Polling Priority	Vector Address
#1	External Interrupt 0, nINT0	EX0	IE0	[PX0L]	(Highest)	0003H
#2	Timer 0	ET0	TF0	[PT0L]	...	000Bh
#3	External Interrupt 1, nINT1	EX1	IE1	[PX1L]	...	0013H
#4	Timer 1	ET1	TF1	[PT1L]	...	001BH
#5	Serial Port 0 (UART0)	ES0	RI0, TI0	[PS0L]	...	0023H
#6	Expanded Interrupt 1	EXPIE1 (ESF, EPCA, EUSB)	(Note 1) (Note 2) (Note 3)	[PXPI1L]	(Lowest)	002Bh

Note1: The System Flag interrupt flags include: KBIF, BOF1, BOF0 and WDTF in PCON1 register, and, STAF and STOF in AUXR1 register.

Note2: The PCA interrupt flags include: CF and CCF0 in PCA register, CCON.

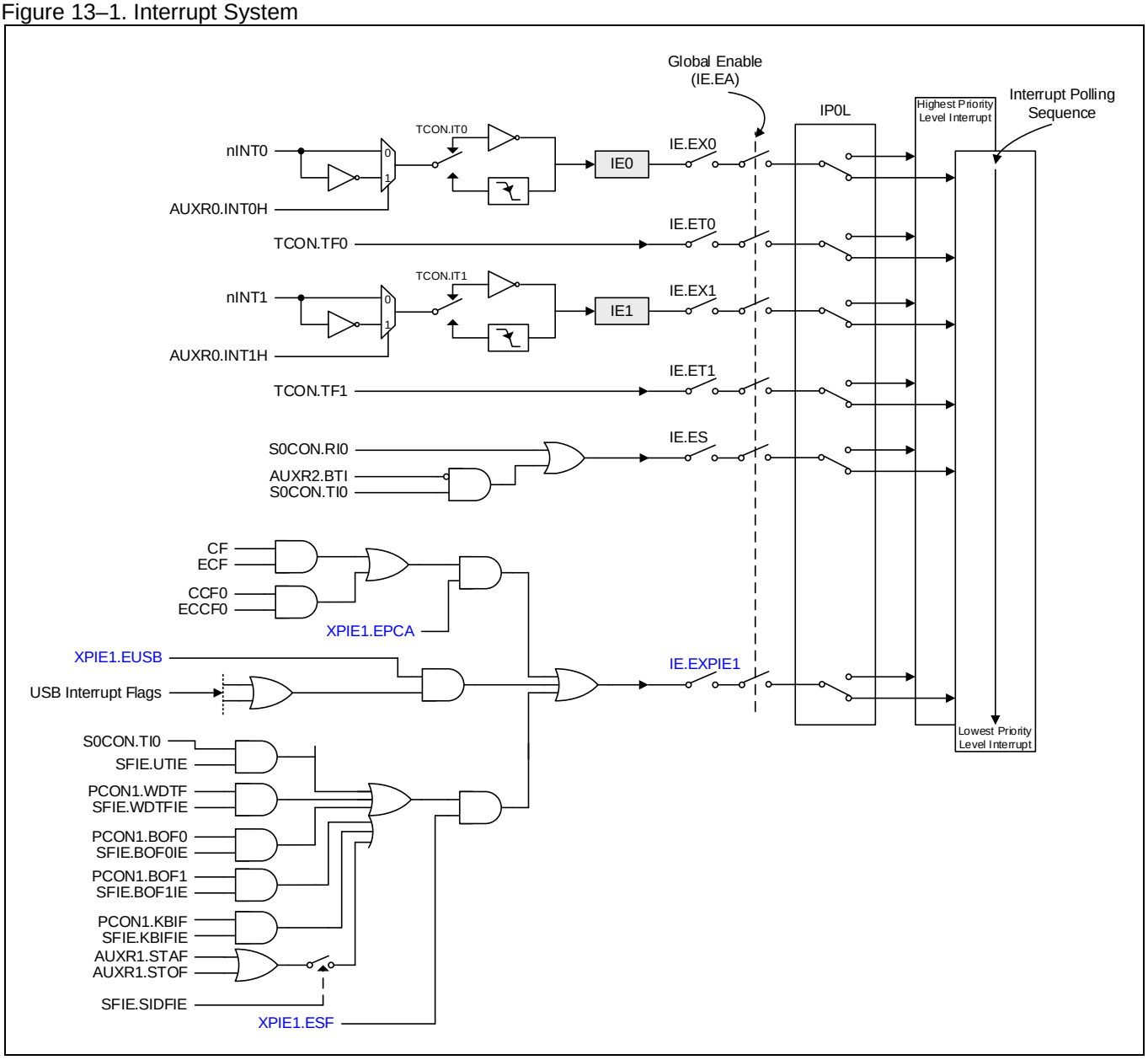
Note3: The USB interrupt flags include:

(1) URSTWKP, URST, URSM and USUS: contained in USB register UPCON.

(2) UTXD0, URXD0, UTXD1, UTXD2, URXD2 and SOFIF: contained in USB register UIFLG.

(3) TXNAK and RXNAK: contained in USB register UIFLG1

Figure 13–1. Interrupt System



13.2. Interrupt Source

Table 13–2. Interrupt flags

No	Source Name	Request Bits	Bit Location
#1	External Interrupt 0,nINT0	IE0	TCON.1
#2	Timer 0	TF0	TCON.5
#3	External Interrupt 1,nINT1	IE1	TCON.3
#4	Timer 1	TF1	TCON.7
#5	Serial Port 0(UART0)	RI0, TI0	S0CON.0 S0CON.1
#6	Expanded Interrupt 1	KBIF, BOF1, BOF0, WDTF, STAF, STOF, (TI0), CF, CCF0, (Note 1)	PCON1.3 PCON1.2 PCON1.1 PCON1.0 AUXR1.3 AUXR1.2 (S0CON.1) CCON.7 CCON.0 (Note 1)

Note1: The USB interrupt flags include:

(1) URSTWKP, URST, URSM and USUS: contained in USB register UPCON.
(2) UTXD0, URXD0, UTXD1, UTXD2, URXD2 and SOFIF: contained in USB register UIFLG.
(3) TXNAK, RXNAK: contained in USB register UIFLG1

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

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

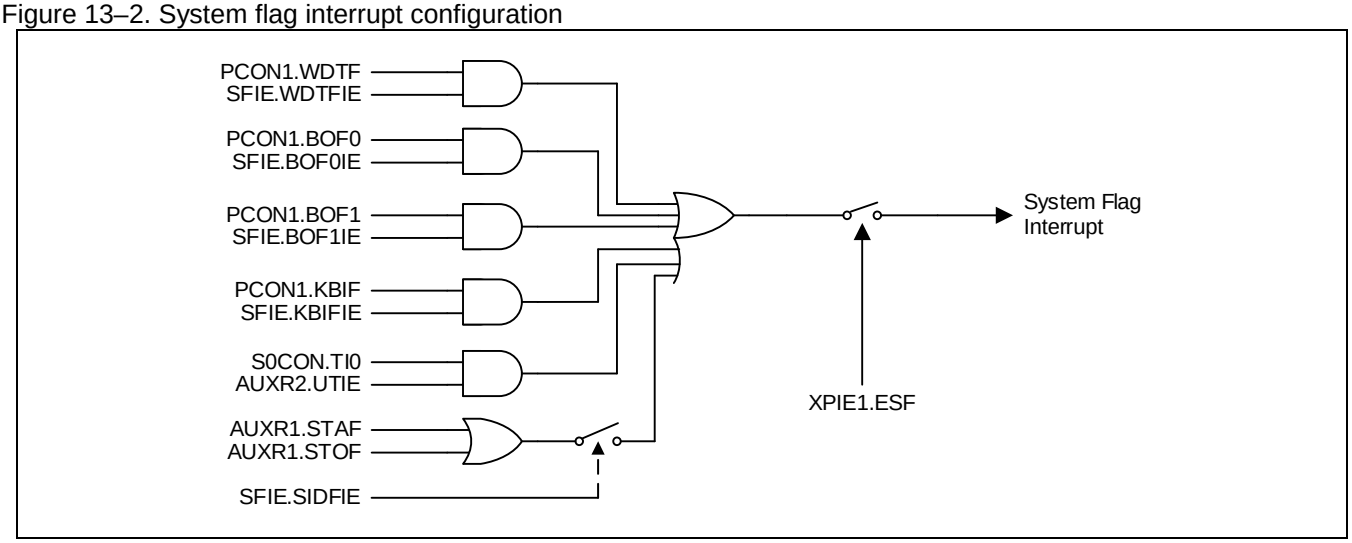
The serial port 0(UART0) interrupt is generated by the logical OR of RI0 and TI0. Neither of these flags is cleared by hardware when the service routine is vectored to. The service routine should poll RI0 and TI0 to determine which one to request service and it will be cleared by software.

The expanded interrupt 1 is grouping by 3 interrupt source: System Flag interrupt, PCA interrupt and USB interrupt. Following describes each interrupt source flags.

The System Flag interrupt is generated by STAF, STOF, KBIF, BOF1 BOF0 and WDTF in PCON1. STAF and STOF are set by serial interface detection. KBIF is set by KBI event. 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. They will not be cleared by hardware when the service routine is vectored to.

Figure 13–2 shows the system flag interrupt configuration.

Figure 13–2. System flag interrupt configuration



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

The USB interrupt is generate by a grouping of USB event flags in USB SFR, which are set by USB engine detecting a new bus state or USB function event happened. They will not be cleared by hardware when the service routine is vectored to.

13.3. Interrupt Enable

Table 13–3. Interrupt enable control

No	Source Name	Enable Bit	Bit Location
#1	External Interrupt 0,nINT0	EX0	IE.0
#2	Timer 0	ET0	IE.1
#3	External Interrupt 1,nINT1	EX1	IE.2
#4	Timer 1	ET1	IE.3
#5	Serial Port 0(UART0)	ES0	IE.4
#6	Expanded Interrupt 1	EXPIE1	IE.5

There are **6** interrupt sources available in **MG74PG1A08**. Each of these interrupt sources can be individually enabled or disabled by setting or clearing an interrupt enable bit in the register IE. 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.

13.4. Interrupt Priority

The priority scheme for servicing the interrupts is the same as that for the standard 80C51. The Priority Bits (see Table 13–1) determine the priority level of each interrupt. IP0L determines to two-level priority interrupt. Table 13–4 shows the bit values and priority levels associated with each combination.

Table 13–4. Interrupt priority level

{ IP0L.x}	Priority Level
1	1 (high)
0	2 (low)

Each interrupt source has one corresponding bit to represent its priority which is located in IP0L register. Higher-priority interrupt will be not interrupted by lower-priority interrupt request. If two interrupt requests of different priority levels are received simultaneously, the request of higher priority is serviced. If interrupt requests of the same priority level are received simultaneously, an internal polling sequence determine which request is serviced. Table 13–2 shows the internal polling sequence in the same priority level and the interrupt vector address.

13.5. 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 and IP0L 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 IP0L, then at least one or more instruction will be executed before any interrupt is vectored to.

13.6. Special Interrupt Vector for TI0

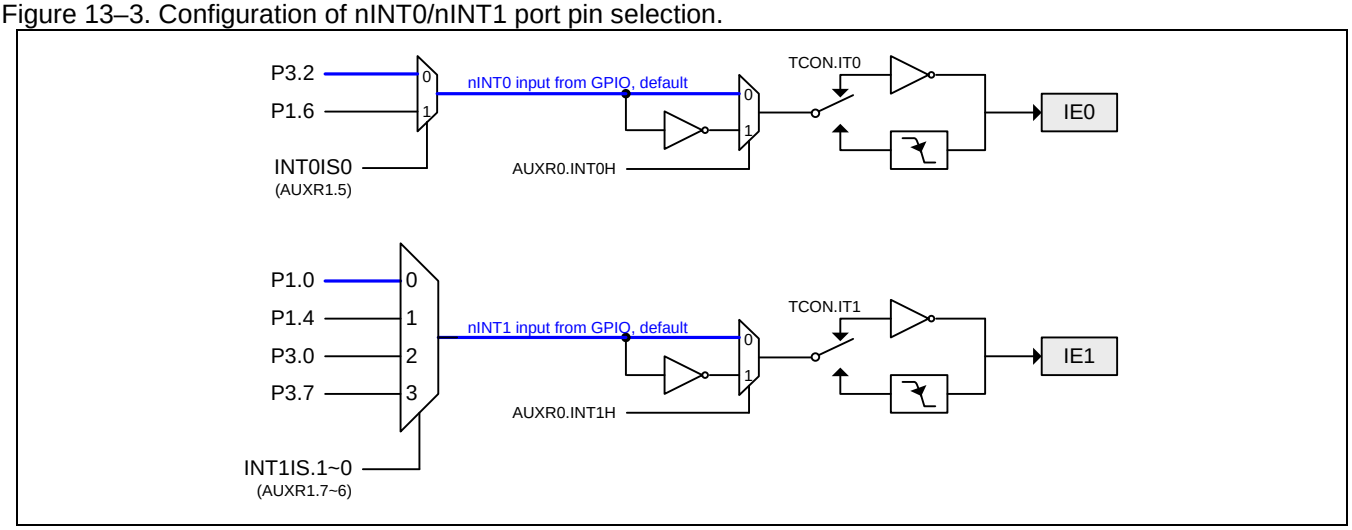
The serial port 0 interrupt from TI0 flag can be masked by BTI (AUXR2.6). If BTI is set, set TI0 flag will not generate a serial port 0 interrupt. The serial port 0 interrupt only reflects the RI0 flag.

If UTIE (AUXR2.7) is set, TI0 flag will be combined into System Flag Interrupt. In this mode, TI0 interrupt shares the interrupt vector with BOF0 and WDTF in System Flag Interrupt.

13.7. nINT0/nINT1 Input Source Selection

The **MG74PG1A08** provides flexible nINT0 and nINT1 source selection to share the port pin input with on-chip serial interface. That will support the additional remote wakeup function for communication peripheral in power-down mode. The nINT0/nINT1 input can be routed to the interface pin to catch port change and set them as an interrupt input event to wake up MCU. INT0H (AUXR0.0) and INT1H (AUXR0.1) configure the port change detection level on low/falling or high/rising event. In MCU power-down mode, both of the falling edge or rising edge configurations of the external interrupts are forced to level-sensitive operation.

Figure 13–3. Configuration of nINT0/nINT1 port pin selection.



13.8. Interrupt Register

TCON: Timer/Counter Control Register

SFR Attribute = Normal Read/Write
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 Edge flag.
0: Cleared when interrupt processed on if transition-activated.
1: Set by hardware when external interrupt 1 edge is detected (transmitted or level-activated).

Bit 2: IT1: Interrupt 1 Type control bit.
0: Cleared by software to specify low level triggered external interrupt 1. 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. If INT1H (AUXR0.1) is set, this bit specifies rising edge triggered on nINT1.

Bit 1: IE0, Interrupt 0 Edge flag.
0: Cleared when interrupt processed on if transition-activated.
1: Set by hardware when external interrupt 0 edge is detected (transmitted or level-activated).

Bit 0: IT0: Interrupt 0 Type control bit.
0: Cleared by software to specify low level triggered external interrupt 0. 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. If INT0H (AUXR0.0) is set, this bit specifies rising edge triggered on nINT0.

IE: Interrupt Enable Register

SFR Attribute = Normal Read/Write

SFR Address = 0xA8 RESET = 0x00-0000

7	6	5	4	3	2	1	0
EA	--	EXPIE1	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: EXPIE1. Enable Expanded Interrupt 1.
0: Disable the interrupt which is grouping of System Flag, PCA and USB.
1: Enable the interrupt which is grouping of System Flag, PCA and USB.

Bit 4: ES0, Serial port 0 interrupt 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 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 enable register.
0: Disable external interrupt 0.
1: Enable external interrupt 1.

XPIE1: Expanded Interrupt 1 Enable Register

SFR Attribute = Normal Read/Write

SFR Address = 0xAD RESET = 0xxx-xx00

7	6	5	4	3	2	1	0
EUSB	--	--	--	--	--	EPCA	ESF
R/W	**W**	**W**	**W**	**W**	**W**	**R/W**	**R/W**

Bit 7: EUSB, Enable USB Interrupt.
0: Disable USB interrupt.
1: Enable USB interrupt.

Bit 6~2: Reserved. Software must write “0” on these bits when XPIE1 is written.

Bit 1: EPCA, Enable PCA interrupt.
0: Disable PCA interrupt.
1: Enable PCA interrupt.

Bit 0: ESF, Enable System Flag interrupt.
0: Disable the interrupt when the group of {KBIF, BOF1, BOF0, WDTF} in PCON1, {STAF, STOF} in AUXR1 or TI0 in S0CON is set
1: Enable the interrupt of the flags of {KBIF, BOF1, BOF0, WDTF} in PCON1, {STAF, STOF} in AUXR1 or TI0 in S0CON when the associated system flag interrupt is enabled in SFIE.

SFIE: System Flag Interrupt Enable Register

SFR Attribute = Normal Read/Write

SFR Address = 0x8E RESET = 0xxx-0000

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

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

Bit 3: KBIFIE, Enable KBIF (PCON1.3) Interrupt.
0: Disable KBIF interrupt.
1: Enable KBIF interrupt.

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

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

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

IP0L: Interrupt Priority 0 Low Register

SFR Attribute = Normal Read/Write

SFR Address = 0xB8 RESET = 0x00-0000

7	6	5	4	3	2	1	0
URXR	--	PXPI1L	PSL	PT1L	PX1L	PT0L	PX0L
R/W	**W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: URXR, Serial Port 0 eXtension Receive.
Bit 6: Reserved. Software must write “0” on this bit when IP0L is written.
Bit 5: PXPI1L, Expanded interrupt 1 priority-L register.
Bit 4: PSL, Serial port 0 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.

AUXR0: Auxiliary Register 0

SFR Attribute = Normal Read/Write

SFR Address = 0xA1 RESET = 0000-0000

7	6	5	4	3	2	1	0
P17OC1	P17OC0	--	T0XL	P1FS1	P1FS0	**INT1H**	**INT0H**
R/W	**R/W**	**W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 5: Reserved. Software must write “0” on this bit when AUXR0 is written.

Bit 1: INT1H, INT1 High/Rising trigger enable.
0: Remain INT1 triggered on low level or falling edge on nINT1 port pin.
1: Set INT1 triggered on high level or rising edge on nINT1 port pin.

Bit 0: INT0H, INT0 High/Rising trigger enable.
0: Remain INT0 triggered on low level or falling edge on nINT0 port pin.

1: Set INT0 triggered on high level or rising edge on nINT0 port pin.

AUXR1: Auxiliary Control Register 1

SFR Attribute = Normal Read/Write

SFR Address = 0xA2 RESET = 0000-0000

7	6	5	4	3	2	1	0
INT1IS1	**INT1IS0**	**INT0IS0**	--	STAF	STOF	PTCKOE	--
R/W	**R/W**	**R/W**	**W**	**R/W**	**R/W**	**R/W**	**W**

Bit 7~6: INT1IS1~0, nINT1 input selection bits which function is defined as following table.

INT1IS.1~0	nINT1
00	P1.0
01	P1.4
10	P3.0
11	P3.7

Bit 5: INT0IS0, nINT0 input selection bits which function is defined as following table.

INT0IS.0	nINT0
0	P3.2
1	P1.6

Bit 4: Reserved. Software must write “0” on this bit when AUXR0 is written.

Bit 0: Reserved. Software must write “0” on this bit when AUXR0 is written.

14. Timers/Counters

MG74PG1A08 has two Timers/Counters: Timer 0 and Timer 1. Timer0/1 can be configured as timers or event counters.

In the “timer” function, the timer rate is pre-scaled by 12 clock cycle to increment register value. In other words, it is to count the standard C51 machine cycle. AUXR2.T0X12 and AUXR2.T1X12 are the function for Timer 0/1 to set the timer rate on every clock cycle. The AUXR0.T0XL is combined with T0X12 to select additional pre-scaler value, SYSCLK/48 and SYSCLK/192, for Timer 0 clock input.

In the “counter” function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T0 or T1. 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.

14.1. Timer0 and Timer1

14.1.1. Mode 0 Structure

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 GATE=0 or nINTx = 1. Mode 0 operation is the same for Timer0 and Timer1. The PWM function of Timer 0/1 is shown in Figure 14–1 and Figure 14–2.

Figure 14–1. Timer 0 Mode 0 Structure

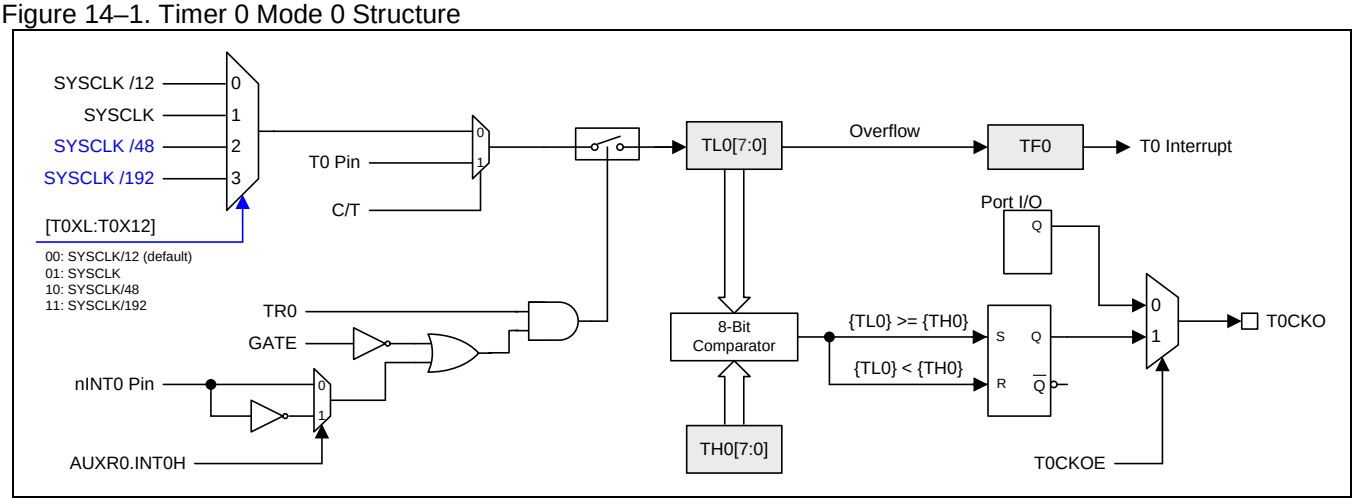
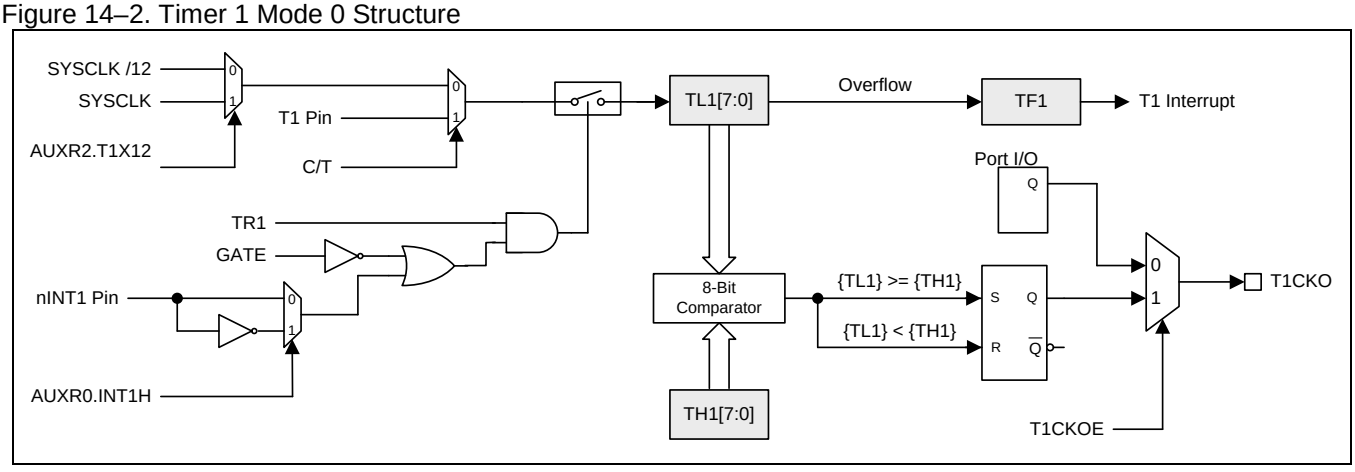


Figure 14–2. Timer 1 Mode 0 Structure



14.1.2. Mode 1 Structure

Timer 0/1 in Mode1 is configured as a 16 bit timer or counter. The function of GATE, nINTx and TRx is same as mode 0. Figure 14–3 and Figure 14–4 show the mode 1 structure of Timer 0 and Timer 1.

Figure 14–3. Timer 0 Mode 1 Structure

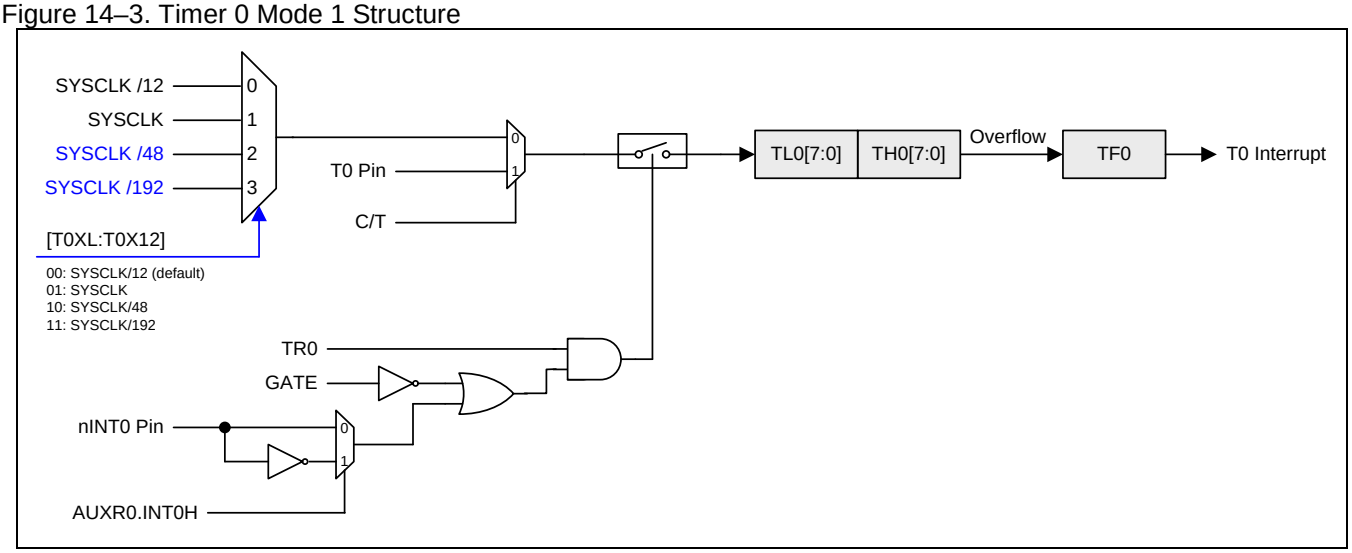
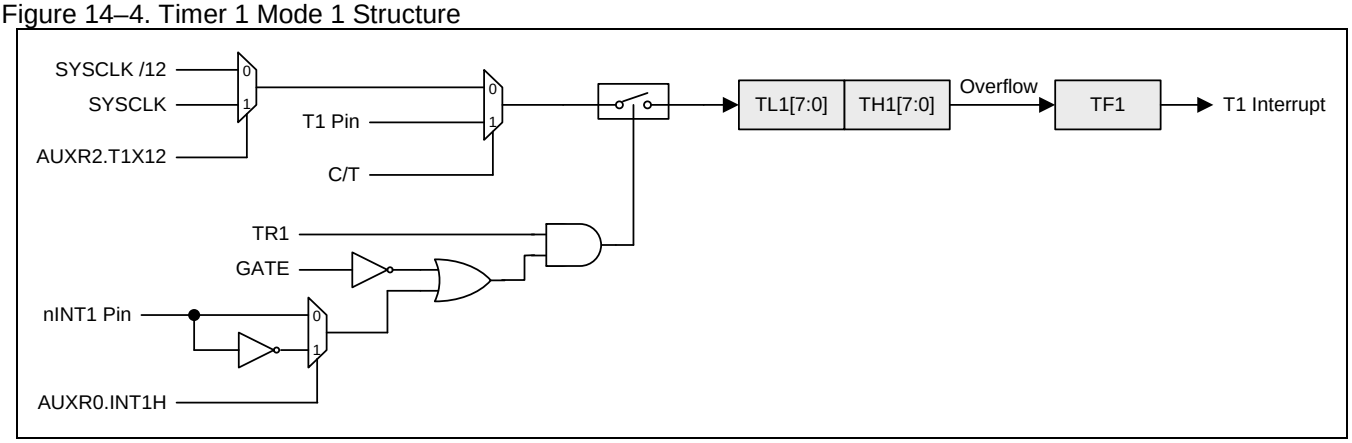


Figure 14–4. Timer 1 Mode 1 Structure



14.1.3. Mode 2 Structure

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 14–5. Timer 0 Mode 2 Structure

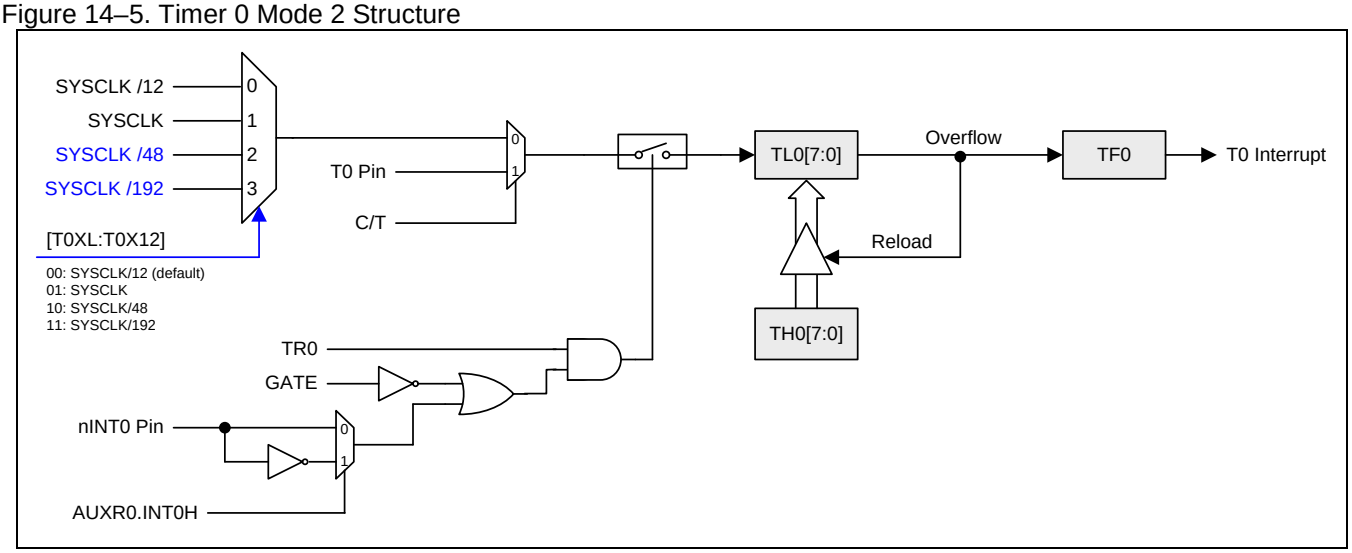
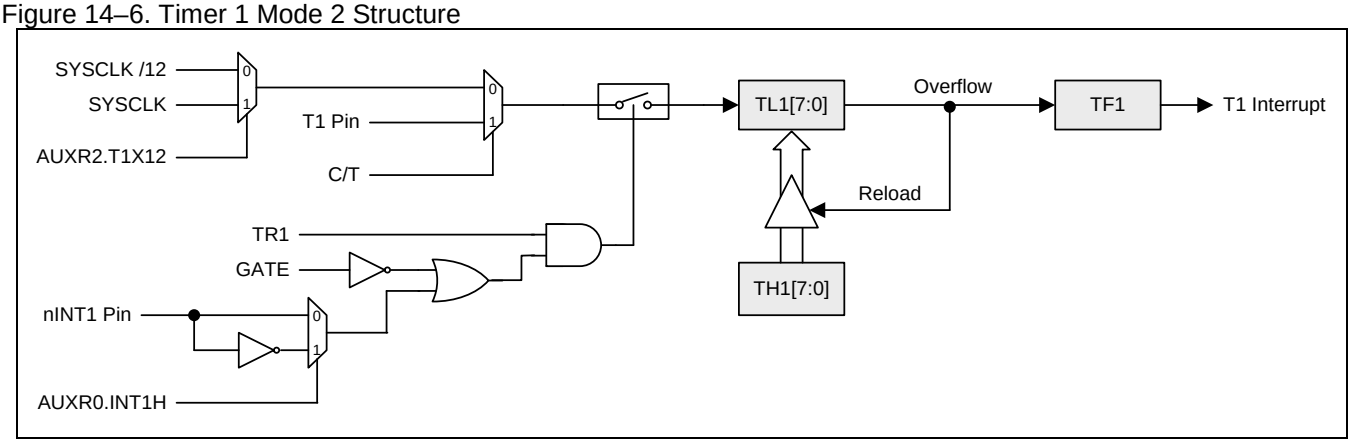


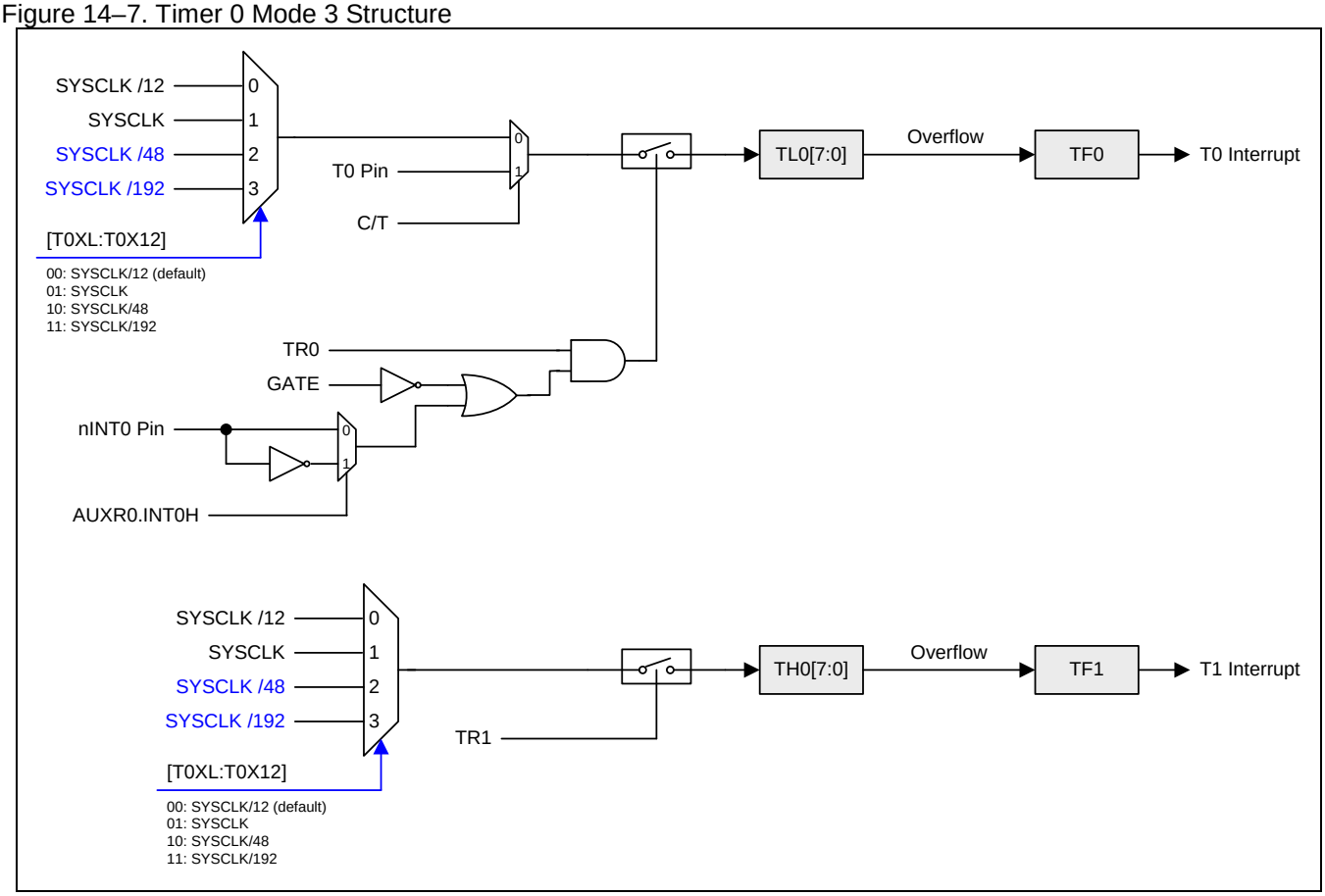
Figure 14–6. Timer 1 Mode 2 Structure



14.1.4. Mode 3 Structure

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, GATE, TR0, nINT0 and TF0. TH0 is locked into a timer function (cannot be external event counter) and take over the use of TR1, TF1 from Timer1. TH0 now controls the Timer1 interrupt.

Figure 14–7. Timer 0 Mode 3 Structure



14.1.5. Timer 0/1 Programmable Clock-Out

Timer 0 and Timer 1 have a Clock-Out Mode (while C/Tx=0 & 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 P1.4 (T0CKO) and P1.5 (T1CKO) individually. The input clock (SYSCLK/12, SYSCLK, SYSCLK/48 or SYSCLK/192) increments the 8-bit timer, TL0, in Timer 0 module. The input clock (SYSCLK/12 or SYSCLK) increments 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. The following formula gives the clock-out frequency:

Figure 14–8. Timer 0 clock out equation

$$\text{T0 Clock-out Frequency} = \frac{\text{SYSCLK Frequency}}{n \times (256 - \text{THx})}$$

; n=24, if {T0XL,T0X12}=00
; n=2, if {T0XL,T0X12}=01
; n=96, if {T0XL,T0X12}=10
; n=384, if {T0XL,T0X12}=11
; C/T = 0

Figure 14–9. Timer 1 clock out equation

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

; n=24, if T1X12=0
; n=2, if T1X12=1
; C/T = 0

Note:

(1) Timer 0/1 overflow flag, TF0/1, will be set when Timer 0/1 overflows but not generate interrupt.

(2) For SYSCLK=12MHz & TxX12=0, Timer 0/1 has a programmable output frequency range from 1.95KHz to 500KHz.

(3) For SYSCLK=12MHz & TxX12=1, Timer 0/1 has a programmable output frequency range from 23.43KHz to 6MHz.

(4) For SYSCLK=12MHz, T0X12=0 & T0XL=1, Timer 0 has a programmable output frequency range from 488Hz to 125KHz.

(5) For SYSCLK=12MHz, TxX12=1 & T0XL=1, Timer 0 has a programmable output frequency range from 122Hz to 31.25KHz.

Figure 14–10. Timer 0 in Clock Output Mode

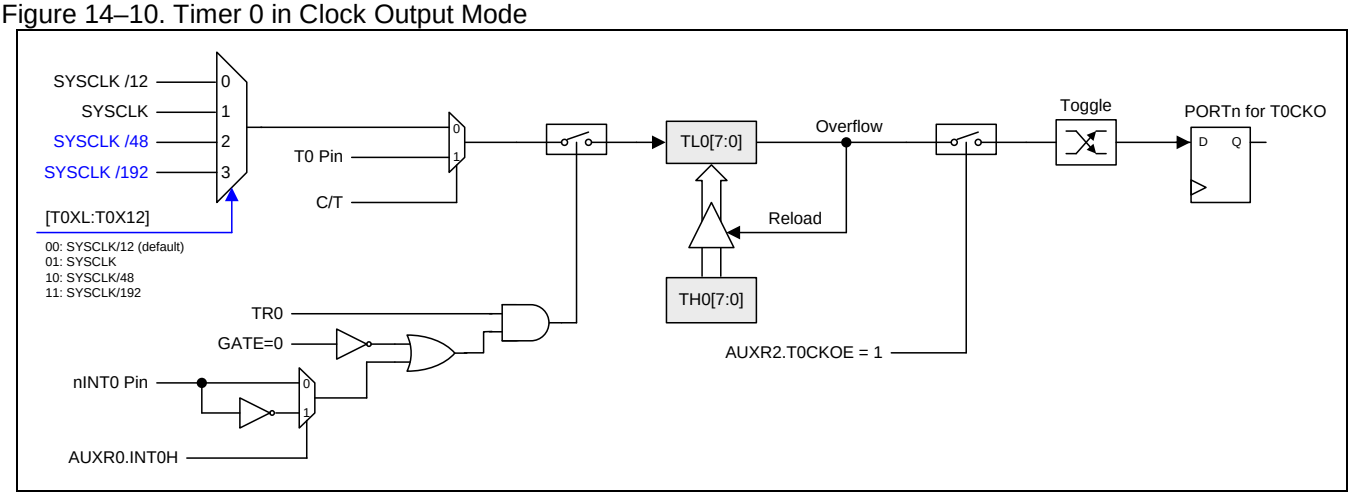
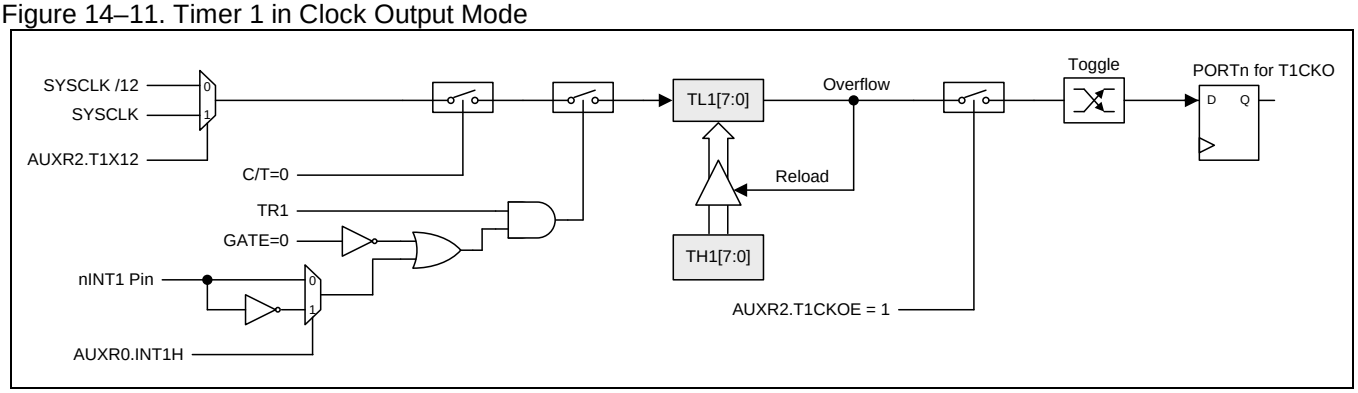


Figure 14–11. Timer 1 in Clock Output Mode



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

- Select AUXR2.T0X12 and AUXR0.T0XL bits decide the Timer 0 clock source. Or select T1X12 in AUXR2 register to decide the Timer 1 clock source.
- Set T0CKOE/T1CKOE bit in AUXR2 register.
- Clear C/T bit in TMOD register.
- 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 TR0/TR1 bit in TCON register to start the Timer 0/1.

In the Clock-Out mode, Timer 0/1 rollovers will not generate an interrupt. This is similar to when Timer 1 is used as a baud-rate generator. It is possible to use Timer 1 as a baud rate generator and a clock generator simultaneously. Note, however, that the baud-rate and the clock-out frequency depend on the same overflow rate of Timer 1.

14.1.6. Timer0/1 Register

TCON: Timer/Counter Control Register

SFR Attribute = Normal Read/Write
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: Cleared by software to turn Timer/Counter 1 off.
1: Set by software to turn Timer/Counter 1 on.

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: Cleared by software to turn Timer/Counter 0 off.
1: Set by software to turn Timer/Counter 0 on.

TMOD: Timer/Counter Mode Control Register

SFR Attribute = Normal Read/Write
SFR Address = 0x89 RESET = 0000-0000

7	6	5	4	3	2	1	0
GATE	C/T	M1	M0	GATE	C/T	M1	M0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**
|←---------- Timer1 ----------→|				|←----------Timer0 ----------→|		

Bit 7/3: Gate, Gating control for Timer1/0.
0: Disable gating control for Timer1/0.
1: Enable gating control for Timer1/0. When set, Timer1/0 or Counter1/0 is enabled only when /INT1 or /INT0 pin is high and TR1 or TR0 control bit is set.

Bit 6/2: C/T, Timer for Counter function selector.
0: Clear for Timer operation, input from internal system clock.
1: Set for Counter operation, input form T1 input pin.

Bit 5~4/1~0: Operating mode selection.

M1	**M0**	**Operating Mode**
0	0	8-bit PWM generator for Timer0 and Timer1
0	1	16-bit timer/counter for Timer0 and Timer1
1	0	8-bit timer/counter with automatic reload for Timer0 and Timer1
1	1 (Timer0)	TL0 is 8-bit timer/counter, TH0 is locked into 8-bit timer
1	1 (Timer1)	Timer/Counter1 Stopped

TL0: Timer Low 0 Register

SFR Attribute = Normal Read/Write
SFR Address = 0x8A RESET = 0000-0000

7	6	5	4	3	2	1	0
TL0.7	TL0.6	TL0.5	TL0.4	TL0.3	TL02	TL0.1	TL0.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

TH0: Timer High 0 Register

SFR Attribute = Normal Read/Write

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 Low 1 Register

SFR Attribute = Normal Read/Write

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 High 1 Register

SFR Attribute = Normal Read/Write

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 Attribute = Normal Read/Write

SFR Address = 0xA3 RESET = 0000-0000

7	6	5	4	3	2	1	0
UTIE	BTI	URM0X3	SM30	T1X12	T0X12	T1CKOE	T0CKOE
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 3: T1X12, Timer 1 clock source selector while C/T=0.
0: Clear to select SYSCLK/12.
1: Set to select SYSCLK as the clock source.

Bit 2: T0X12, Timer 0 clock source selector while C/T=0.
0: Clear to select SYSCLK/12.
1: Set to select SYSCLK as the clock source.

T0XL, T0X12	Timer 0 Clock Selection
0 0	SYSCLK/12
0 1	SYSCLK
1 0	SYSCLK/48
1 1	SYSCLK/192

Bit 1: T1CKOE, Timer 1 Clock Output Enable.
0: Disable Timer 1 clock output.
1: Enable Timer 1 clock output on P1.5.

Bit 0: T0CKOE, Timer 0 Clock Output Enable.
0: Disable Timer 0 clock output.
1: Enable Timer 0 clock output on P1.4.

15. Programmable Counter Array (PCA)

The **MG74PG1A08** is equipped with a Programmable Counter Array (PCA), which provides more timing capabilities with less CPU intervention than the standard timer/counters. Its advantages include reduced software overhead and improved accuracy.

15.1. PCA Overview

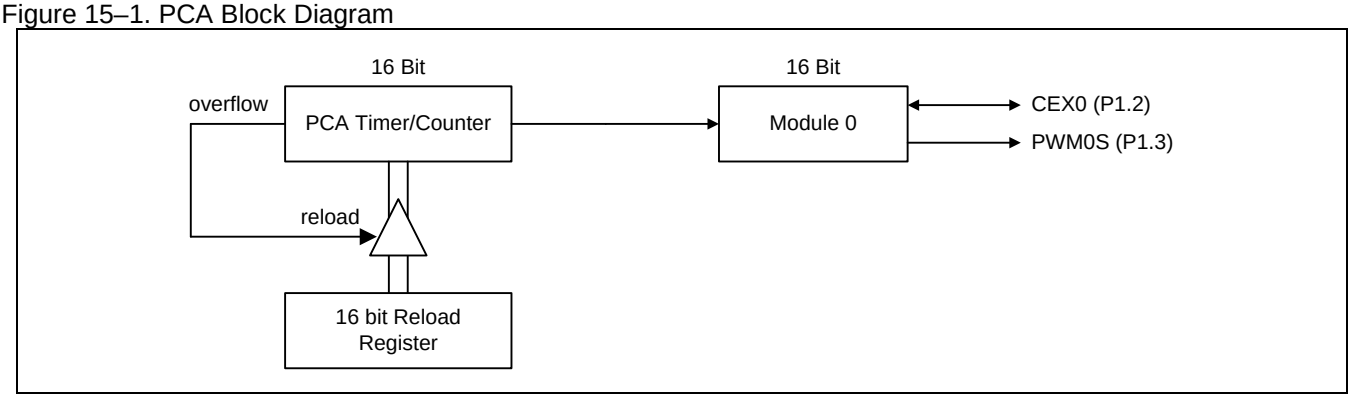
The PCA consists of a dedicated timer/counter which serves as the time base for an compare/capture modules. Figure 15–1 shows a block diagram of the PCA. Notice that the PCA timer and module are all 16-bits. If an external event is associated with a module, that function is shared with the corresponding Port 1 pin. If the module is not using the port pin, the pin can still be used for standard I/O.

The module can be programmed in any one of the following modes:

- Rising and/or Falling Edge Capture
- Software Timer
- High Speed Output
- Pulse Width Modulator (PWM) Output

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

Figure 15–1. PCA Block Diagram



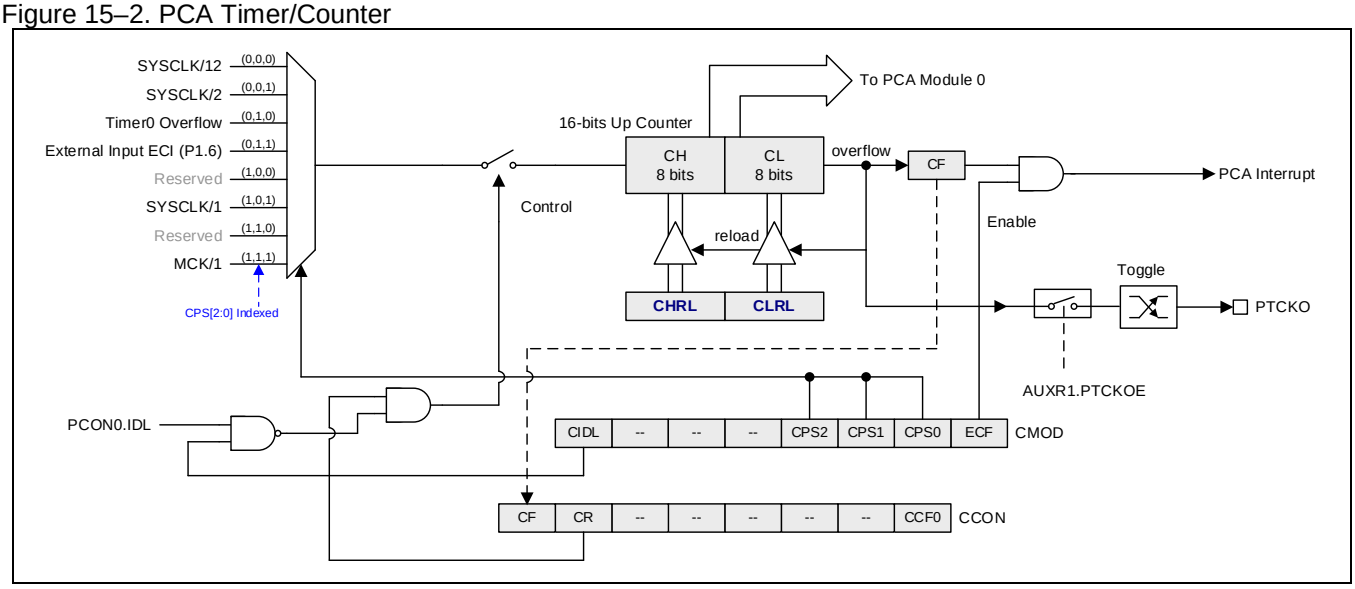
15.2. PCA Timer/Counter

The timer/counter for the PCA is an auto-reload 16-bit timer consisting of registers CH, CL (the high and low bytes of the count values), CHRL, CLRL (the high and low bytes reload registers), as shown in Figure 15–2. 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 16-bit PWM.

It 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 (P1.6).
- directly from the system clock (SYSCLK) frequency,
- directly from the MCK frequency,

Special Function Register CMOD contains the Count Pulse Select bits (CPS2, CPS1 and CPS0) to specify the PCA timer input. This register also contains the ECF bit which enables an interrupt when the counter overflows. In addition, the user has the option of turning off the PCA timer during Idle Mode by setting the Counter Idle bit (CIDL). This can further reduce power consumption during Idle mode.

Figure 15–2. PCA Timer/Counter



CMOD: PCA Counter Mode Register

SFR Attribute = Normal Read/Write

SFR Address = 0xD9 RESET = 0xxx-0000

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

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

Bit 6~4: Reserved. Software must write “0” on these bits when CMOD is written.

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

CPS2	CPS1	CPS0	PCA Clock Source
0	0	0	Internal clock, SYSCLK/12
0	0	1	Internal clock, SYSCLK/2
0	1	0	Timer 0 overflow
0	1	1	External clock at the ECI pin
1	0	0	Reserved
1	0	1	Internal clock, SYSCLK/1
1	1	0	Reserved
1	1	1	Internal clock, MCK/1

Bit 0: ECF, Enable PCA counter overflow interrupt.
0: Disables an interrupt when CF bit (in CCON register) is set.
1: Enables an interrupt when CF bit (in CCON register) is set.

The CCON register shown below contains the run control bit for the PCA and the flags for the PCA timer and each module. To run the PCA the CR bit (CCON.6) must be set by software. The PCA is shut off by clearing this bit. The CF bit (CCON.7) is set when the PCA counter overflows and an interrupt will be generated if the ECF bit in the CMOD register is set. The CF bit can only be cleared by software. CCF0 is the interrupt flags for module 0, and it set by hardware when either a match or a capture occurs. These flags also can only be cleared by software. The PCA interrupt system is shown Figure 15–3.

CCON: PCA Counter Control Register

SFR Attribute = Normal Read/Write

SFR Address = 0xD8 RESET = 00xx-xxx0

7	6	5	4	3	2	1	0
CF	CR	--	--	--	--	--	CCF0
R/W	**R/W**	**W**	**W**	**W**	**W**	**W**	**R/W**

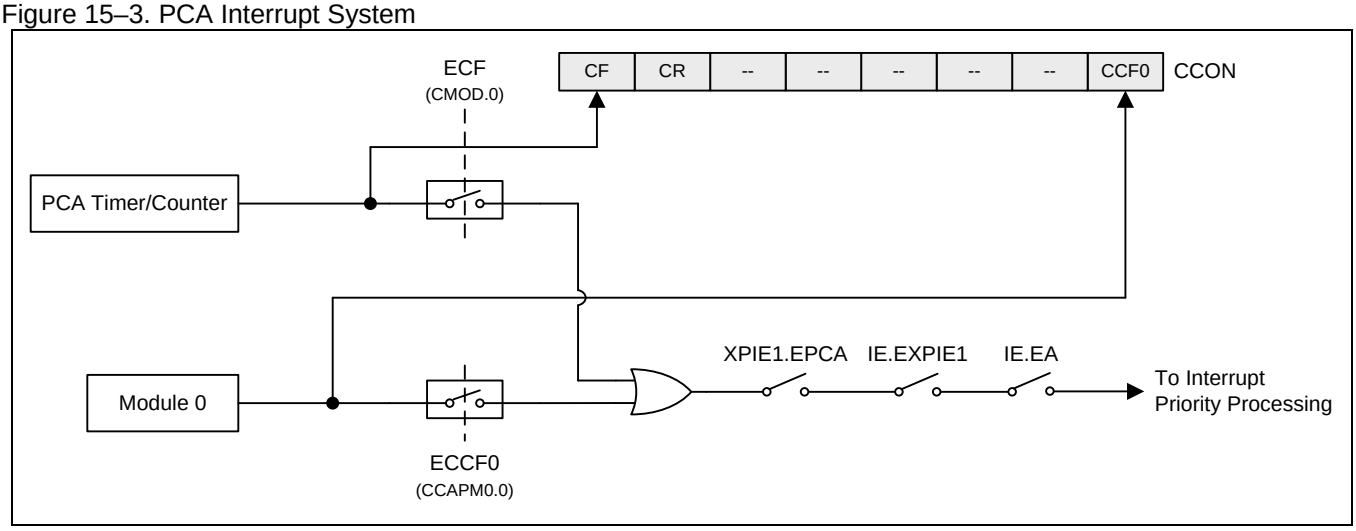
Bit 7: CF, PCA Counter Overflow flag.
0: Only be cleared by software.
1: Set by hardware when the counter rolls over. CF flag can generate an interrupt if bit ECF in CMOD is set. CF may be set by either hardware or software.

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

Bit 5~1: Reserved. Software must write “0” on these bits when CCON is written.

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

Figure 15–3. PCA Interrupt System



CL: PCA Counter Low byte Register

SFR Attribute = Normal Read/Write

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**

CH: PCA Counter High byte Register

SFR Attribute = Normal Read/Write

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**

CLRL: PCA Counter Low byte Reload Register

SFR Attribute = Normal Read/Write

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**

CHRL: PCA Counter High byte Reload Register

SFR Attribute = Normal Read/Write

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**

15.3. Compare/Capture Modules

The compare/capture module has a mode register called CCAPM0 to select which function it will perform. Note the ECCF0 bit which enables an interrupt to occur when a module's interrupt flag is set.

CCAPM0: PCA Module 0 Compare/Capture Register

SFR Attribute = Normal Read/Write

SFR Address = 0xDA RESET = 0000-0000

7	6	5	4	3	2	1	0
P0INV	ECOM0	CAPP0	CAPN0	MAT0	TOG0	PWM0	ECCF0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: Invert PWM output on CEX0.
0: Non-inverted PWM output.
1: Inverted PWM output.

Bit 6: ECOM0, Enable Comparator
0: Disable the digital comparator function.
1: Enables the digital comparator function.

Bit 5: CAPP0, Capture Positive enabled.
0: Disable the PCA capture function on CEX0 positive edge detected.
1: Enable the PCA capture function on CEX0 positive edge detected.

Bit 4: CAPN0, Capture Negative enabled.
0: Disable the PCA capture function on CEX0 positive edge detected.
1: Enable the PCA capture function on CEX0 negative edge detected.

Bit 3: MAT0, Match control.
0: Disable the digital comparator match event to set CCF0.
1: A match of the PCA counter with this module's compare/capture register causes the CCF0 bit in CCON to be set. **If this bit is set with PWM0, it will select the module to 16-bit PWM mode.**

Bit 2: TOG0, Toggle control.
0: Disable the digital comparator match event to toggle CEX0.
1: A match of the PCA counter with this module's compare/capture register causes the CEX0 pin to toggle. **If this bit is set with PWM0, it will select the module to double channel 8-bit PWM mode.**

Bit 1: PWM0, PWM control.
0: Disable the PWM mode in PCA module.
1: Enable the PWM function and cause CEX0 pin to be used as a pulse width modulated output.

Bit 0: ECCF0, Enable CCF0 interrupt.
0: Disable compare/capture flag CCF0 in the CCON register to generate an interrupt.
1: Enable compare/capture flag CCF0 in the CCON register to generate an interrupt.

Note: The bits CAPN0 (CCAPM0.4) and CAPP0 (CCAPM0.5) 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.

The module also has a pair of 8-bit compare/capture registers (CCAP0H, CCAP0L) associated with it. These registers are used to store the time when a capture event occurred or when a compare event should occur.

15.4. Operation Modes of the PCA

Table 15–1 shows the CCAPM0 register settings for the various PCA functions.

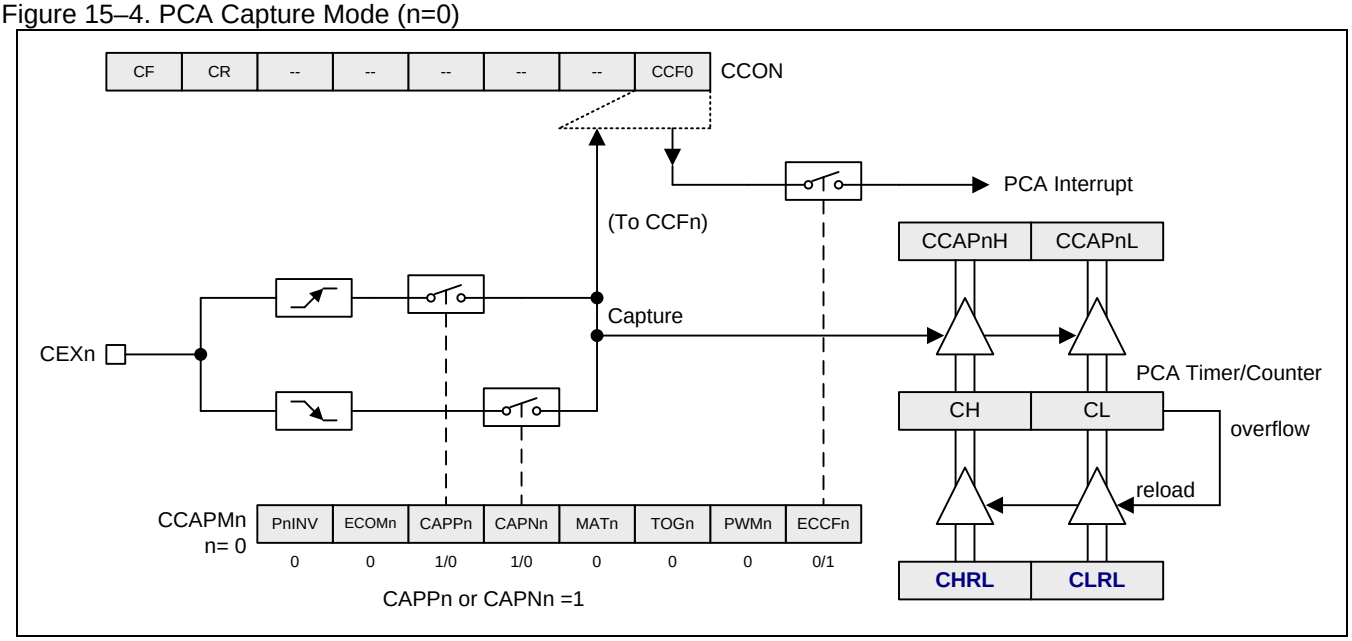
Table 15–1. PCA Module 0 Modes

ECOM0	CAPP0	CAPN0	MAT0	TOG0	PWM0	ECCF0	Module Function
0	0	0	0	0	0	0	No operation
0	1	0	0	0	0	X	16-bit capture by a positive-edge trigger on CEX0
0	0	1	0	0	0	X	16-bit capture by a negative-edge trigger on CEX0
0	1	1	0	0	0	X	16-bit capture by a transition on CEX0
1	0	0	1	0	0	X	16-bit Software Timer
1	0	0	1	1	0	X	16-bit High Speed Output
1	0	0	0	0	1	X	8-bit Pulse Width Modulator (8-bit PWM)
1	0	0	1	0	1	X	16-bit PWM, un-buffered
1	0	0	0	1	1	X	Double Channel 8-bit PWM, un-buffered

15.4.1. Capture Mode

To use one of the PCA modules in the capture mode, either one or both of the bits CAPN0 and CAPP0 for that module must be set. The external CEX0 input for the module is sampled for a transition. When a valid transition occurs the PCA hardware loads the value of the PCA counter registers (CH and CL) into the module's capture registers (CCAP0L and CCAP0H). If the CCF0 and the ECCF0 bits for the module are both set, an interrupt will be generated.

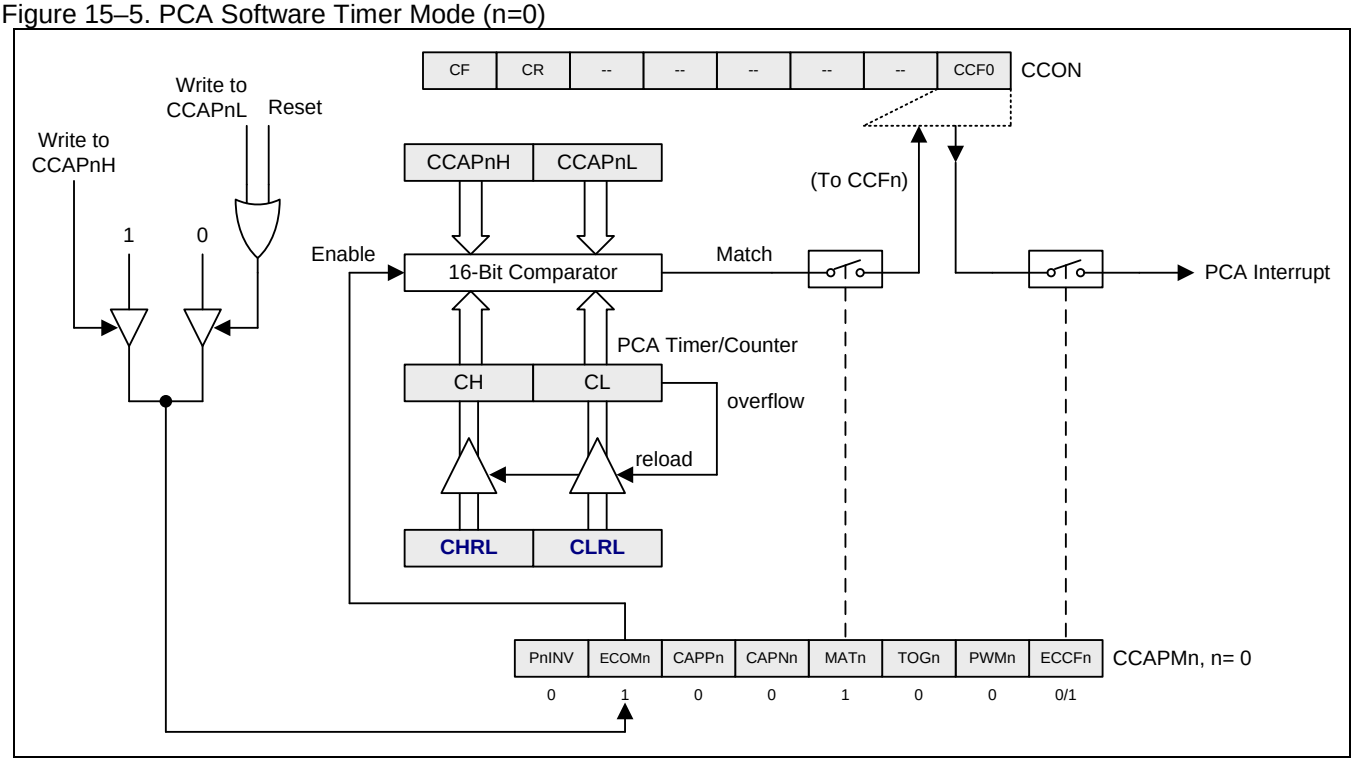
Figure 15–4. PCA Capture Mode (n=0)



15.4.2. 16-bit Software Timer Mode

The PCA modules can be used as software timers by setting both the ECOM0 and MAT0 bits in the module's CCAPM0 register. The PCA timer will be compared to the module's capture registers, and when a match occurs an interrupt will occur if the CCF0 and the ECCF0 bits for the module are both set.

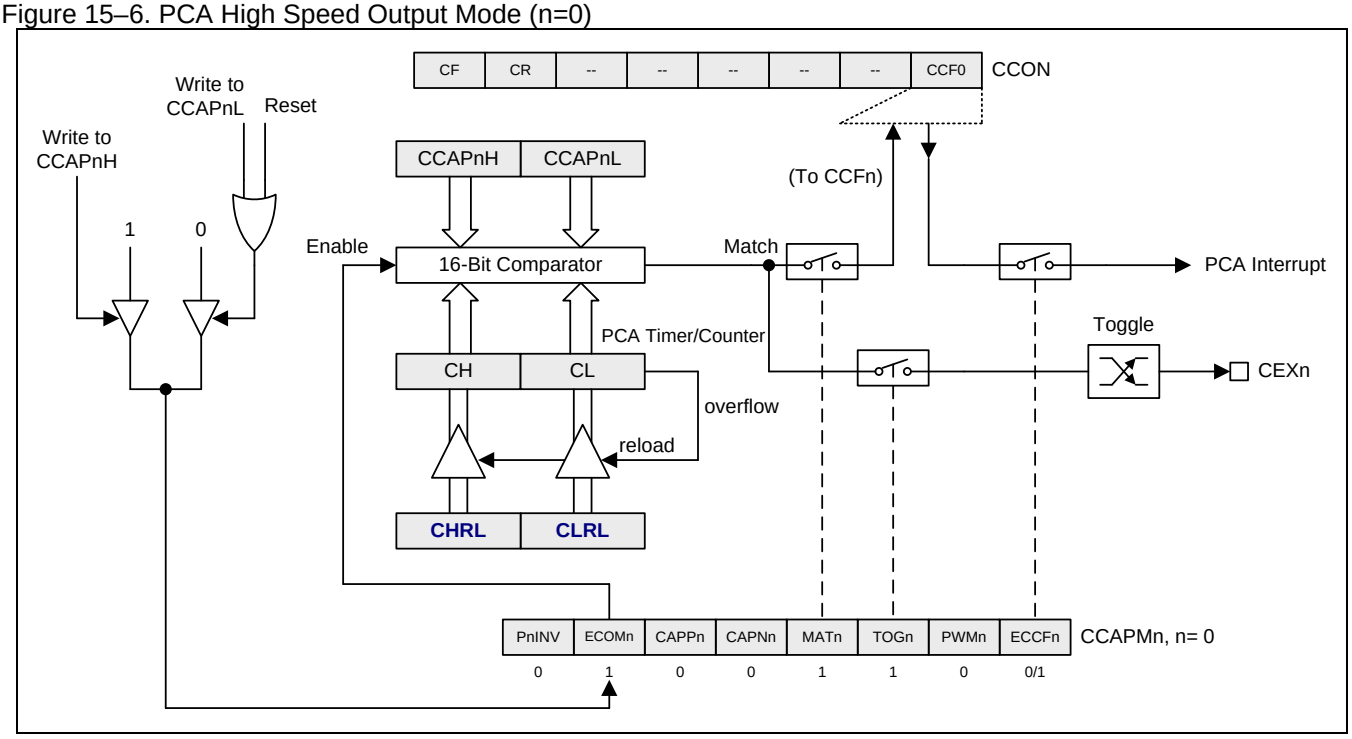
Figure 15–5. PCA Software Timer Mode (n=0)



15.4.3. High Speed Output Mode

In this mode the CEX0 output associated with the PCA module will toggle each time a match occurs between the PCA counter and the module's capture registers. To activate this mode, the TOG0, MAT0 and ECOM0 bits in the module's CCAPM0 register must be set.

Figure 15–6. PCA High Speed Output Mode (n=0)



15.4.4. 8-bit PWM Mode (Buffered 8-bit PWM)

The PCA module can be used as 8-bit buffered PWM output. The frequency of the PWM output depends on the clock source for the PCA timer and the PWM resolution setting. Software may modify the value of CHRL and CLRL to decrease the PWM resolution, such as 7-bit PWM. To activate this mode, the PWM0 and ECOM0 bits in the module's CCAPM0 register must be set.

In this PWM mode, the duty cycle of the module is determined by the module's capture register CCAP0L. When the 8-bit value of { CL } is *less than* the 8-bit value of { CCAP0L } the output will be low, and if *equal to or greater than* the output will be high.

When CL overflows from 0xFF to 0x00, { CCAP0L } is reloaded with the value of { CCAP0H }. This allows updating the PWM without glitches. The PWM0 and ECOM0 bits in the module's CCAPM0 register must be set to enable the 8-bit PWM mode.

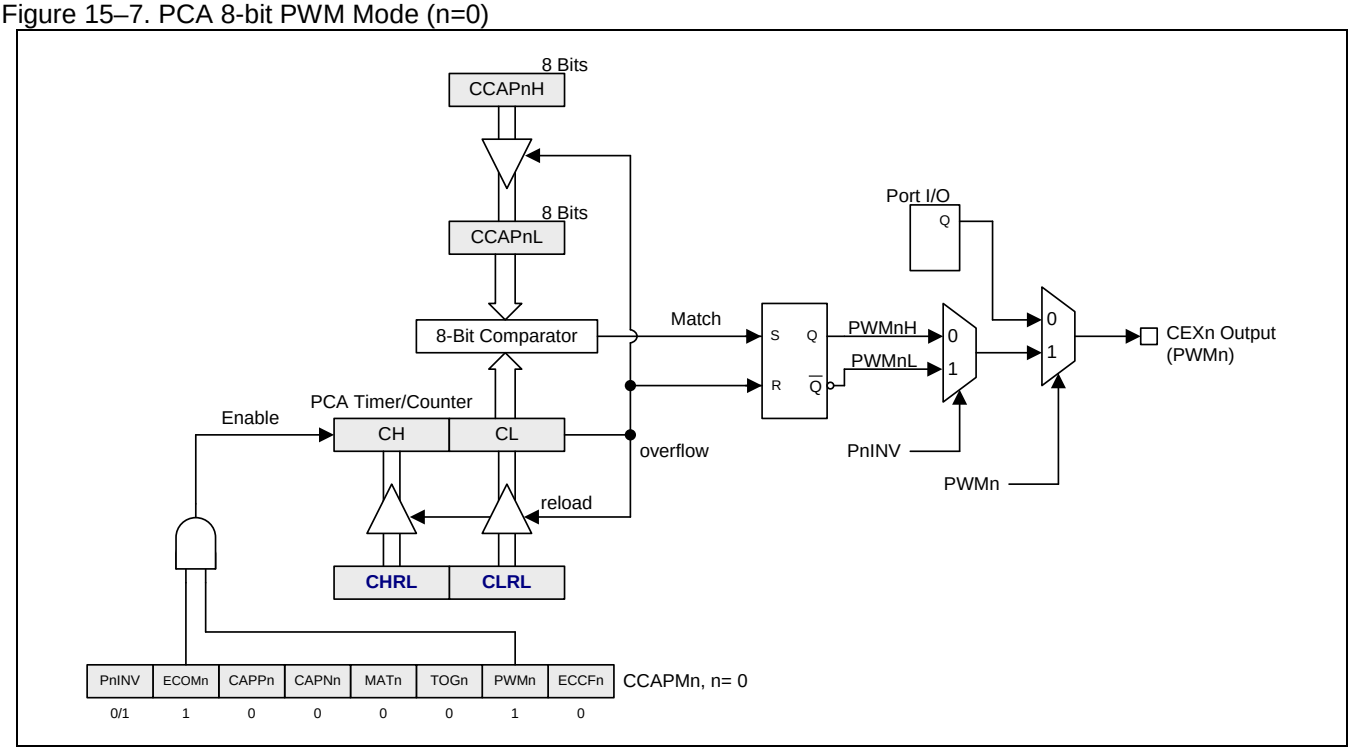
Using the 8-bit comparison, the duty cycle of the output can be improved to really start from 1/256, and up to 100%. The formula for the duty cycle is:

Duty Cycle = 1 – { CCAP0H } / 256.

For examples,
a. If CCAP0H= 0x00, the duty cycle is 100%.
b. If CCAP0H= 0x40, the duty cycle is 75%.
c. If CCAP0H= 0xC0, the duty cycle is 25%.
d. If CCAP0H= 0xFF, the duty cycle is 1/256.

Software can program the value of CHRL and CLRL, reload registers of CH and CL, to modify the PWM resolution less than 256. Please refer Figure 15–7 to find the PWM structure.

Figure 15–7. PCA 8-bit PWM Mode (n=0)



15.4.5. 16-bit PWM Mode (Un-Buffered)

The PCA module can be used as 16-bit un-buffered PWM output. The frequency of the PWM output depends on the clock source for the PCA timer and the PWM resolution setting. Software may modify the value of CHRL and CLRL to decrease the PWM resolution, such as 10-bit PWM. To activate this mode, the MAT0, PWM0 and ECOM0 bits in the module's CCAPM0 register must be set.

In this PWM mode, the duty cycle of the module is determined by the module's capture register CCAP0H and CCAP0L. When the 16-bit value of { CH, CL] } is *less than* the 16-bit value of { CCAP0H, CCAP0L } the output will be low, and if *equal to or greater than* the output will be high.

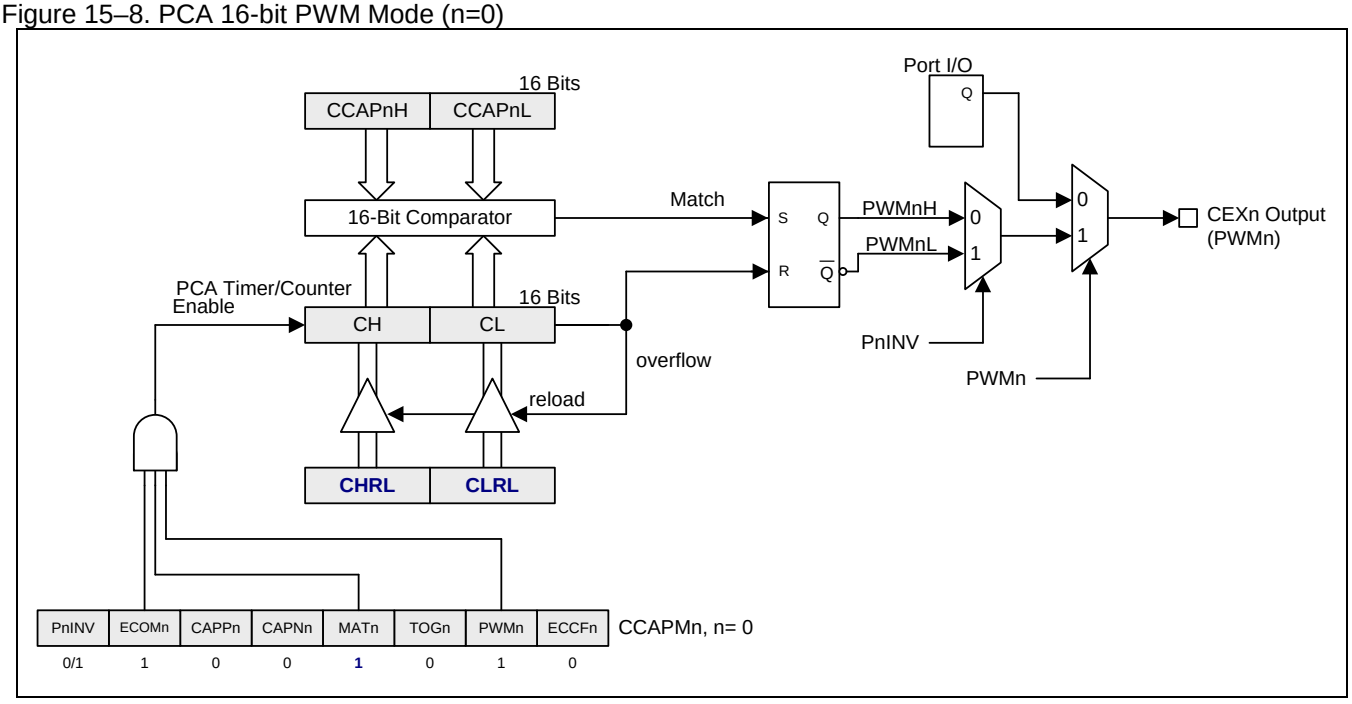
Using the 16-bit comparison, the duty cycle of the output can be improved to really start from 1/65536, and up to 100%. The formula for the duty cycle is:

Duty Cycle = 1 – { CCAP0H, CCAP0L } / 65536.

For examples,
a. If { CCAP0H, CCAP0L }= 0x0000, the duty cycle is 100%.
b. If { CCAP0H, CCAP0L }=0x4000, the duty cycle is 75%.
c. If { CCAP0H, CCAP0L }=0xC000, the duty cycle is 25%.
d. If { CCAP0H, CCAP0L }=0xFFFF, the duty cycle is 1/65536.

Software can program the value of CHRL and CLRL, reload registers of CH and CL, to modify the PWM resolution less than 65536. Please refer Figure 15–8 to find the PWM structure.

Figure 15–8. PCA 16-bit PWM Mode (n=0)



15.4.6. Double Channel PWM Mode (Un-Buffered 8-bit PWM)

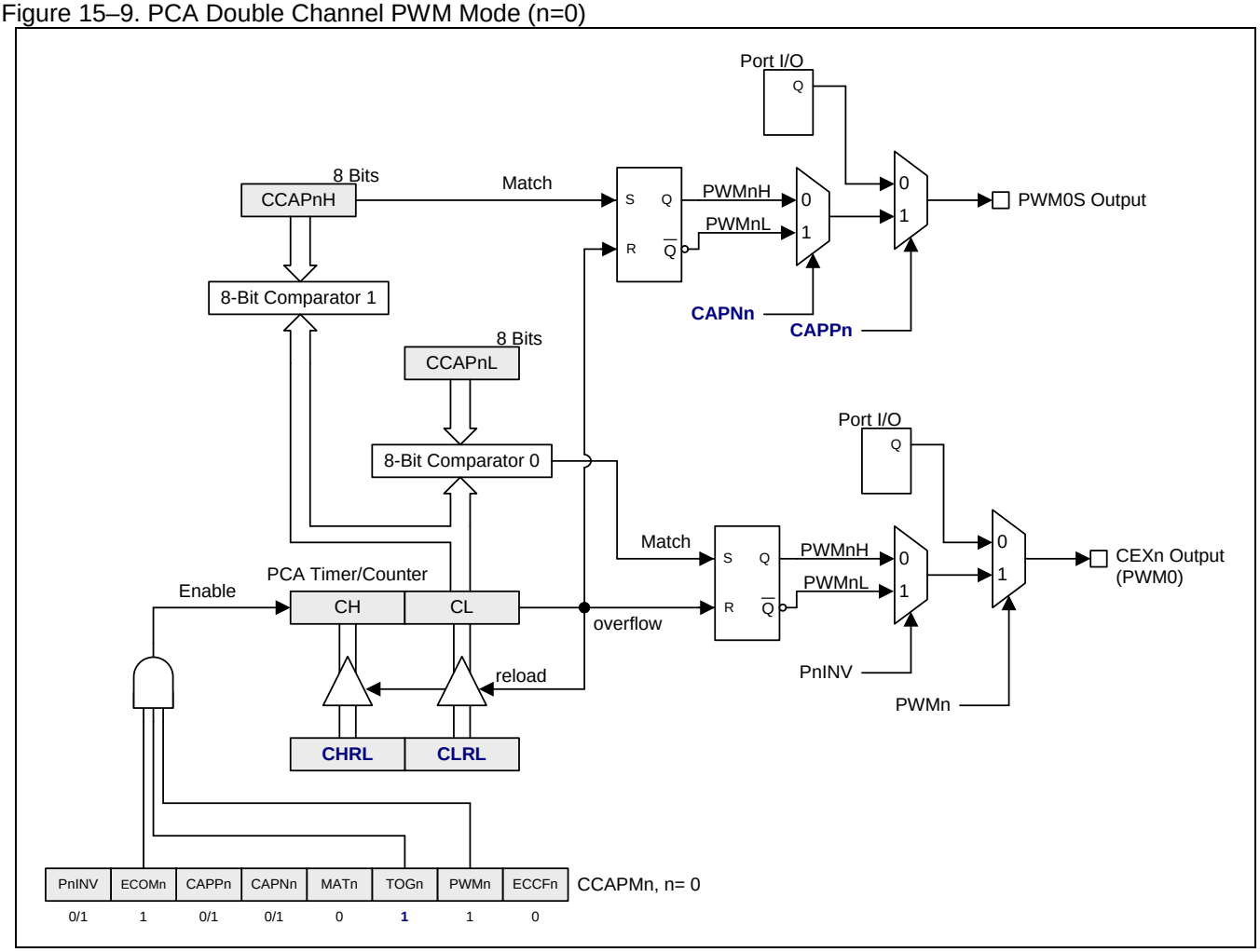
The PCA module can be used as double channel un-buffered 8-bit PWM output, PWM0 and PWM0S. The frequency of the PWM output depends on the clock source for the PCA timer and the PWM resolution setting. Software may modify the value of CHRL and CLRL to decrease the PWM resolution, such as 8-bit PWM. To activate this mode, the TOG0, PWM0 and ECOM0 bits in the module's CCAPM0 register must be set.

In this PWM mode, there are two 8-bit comparators to execute the PWM outputs. The primary PWM output channel, PWM0, is the comparison result of CL and CCAP0L. The secondary PWM output channel, PWM0S, is the comparison result of CL and CCAP0H. Both of the PWM generators are similar to 8-bit buffered PWM mode except the un-buffered PWM structure. Please refer Figure 15–9 to find the PWM structure.

Using the 8-bit comparison, the duty cycle of the output can be improved to really start from 1/256, and up to 100%. The formula for the duty cycle is:

PWN0 Duty Cycle = 1 – { [CCAP0L] } / 256.
PWN0S Duty Cycle = 1 – { [CCAP0H] } / 256.

Figure 15–9. PCA Double Channel PWM Mode (n=0)



16. Serial Port 0 (UART0)

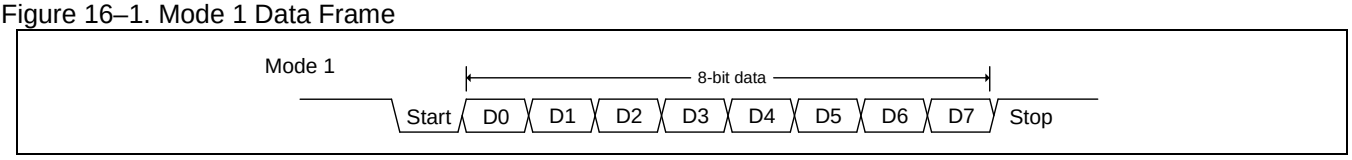
The serial port 0 of **MG74PG1A08** support full-duplex transmission, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the register. However, if the first byte still hasn't been read by the time reception of the second byte is complete, one of the bytes will be lost. The serial port 0 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 0 can operate in 4 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 0: 8 data bits (LSB first) are transmitted or received through RXD0 (P3.0). TXD0 (P3.1) 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 AUXR2 register. In **MG74PG1A08**, 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 16–4 and Figure 16–5 show the clock polarity waveform in Mode 0.

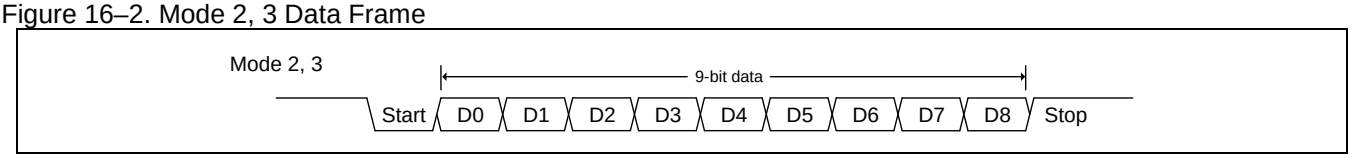
Mode 1: 10 bits are transmitted through TXD0 or received through RXD0. The frame data includes a start bit (0), 8 data bits (LSB first), and a stop bit (1), as shown in Figure 16–1. On receive, the stop bit would be loaded into RB80 in S0CON register. The baud rate is variable.

Figure 16–1. 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 16–2. 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 16–2. 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.

16.1. 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 AUXR2 register. Figure 16–3 shows a simplified functional diagram of the serial port 0 in Mode 0.

Transmission is initiated by any instruction that uses S0BUF as a destination register. The “write to S0BUF” signal triggers the UART0 engine to start the transmission. The data in the S0BUF would be shifted into the RXD0 (P3.0) pin by each raising edge shift clock on the TXD0 (P3.1) pin. After eight raising edge of shift clocks passing, TI0 would be asserted by hardware to indicate the end of transmission. Figure 16–4 shows the transmission waveform in Mode 0.

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

Receive enables Shift Clock which directly comes from RX Clock to the alternate output function of P3.1 pin. When Receive is active, the contents on the RXD0 (P3.0) 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 16–5 shows the reception waveform in Mode 0. The clock polarity can be selected by software setting on P3.1 data latch before serial transfer shifted. If P3.1 is set to logic high, the clock polarity is same as standard 8051. If P3.1 data latch is cleared to logic low, the clock polarity is inverted to standard 8051 UART Mode 0.

Figure 16–3. Serial Port 0 Mode 0

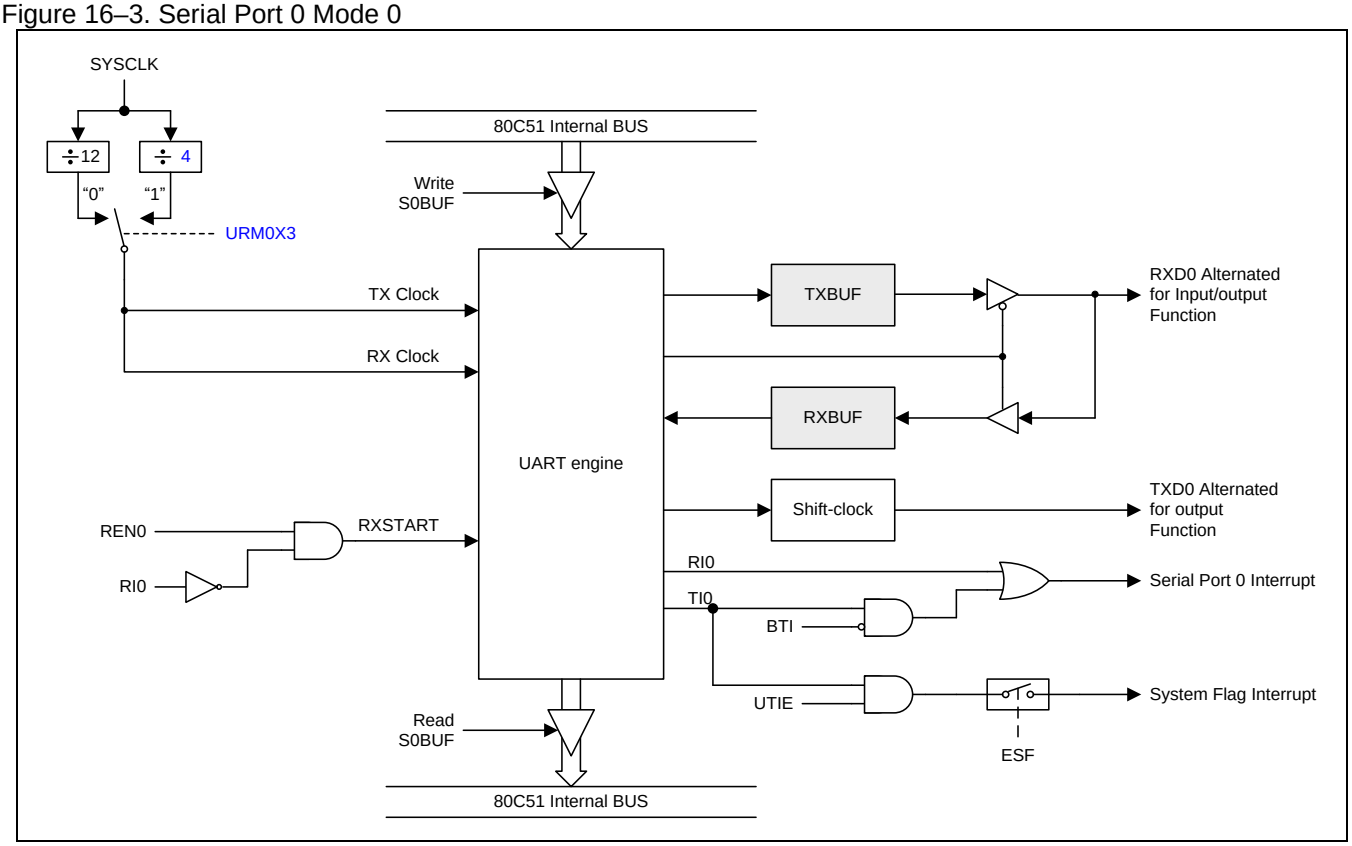


Figure 16–4. Mode 0 Transmission Waveform

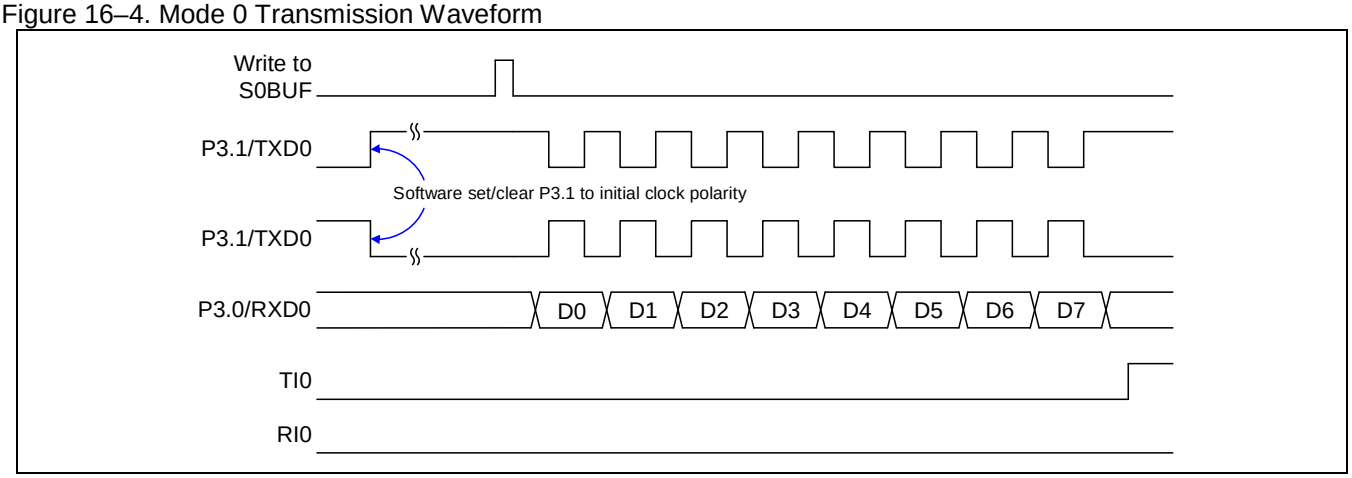
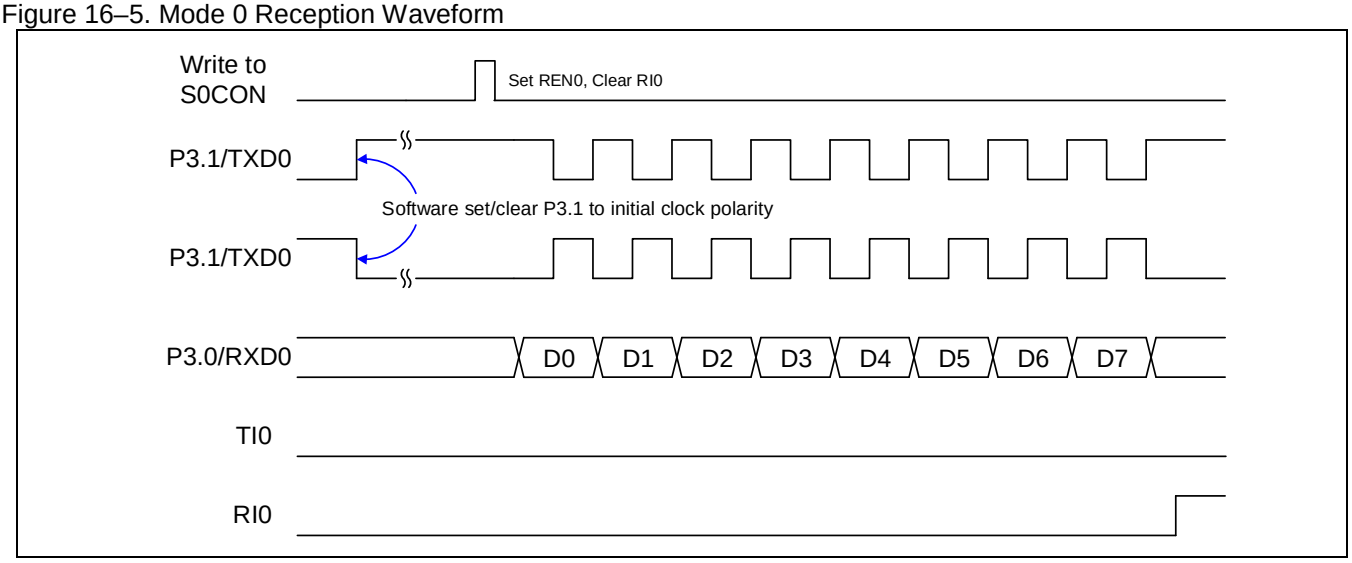


Figure 16–5. Mode 0 Reception Waveform



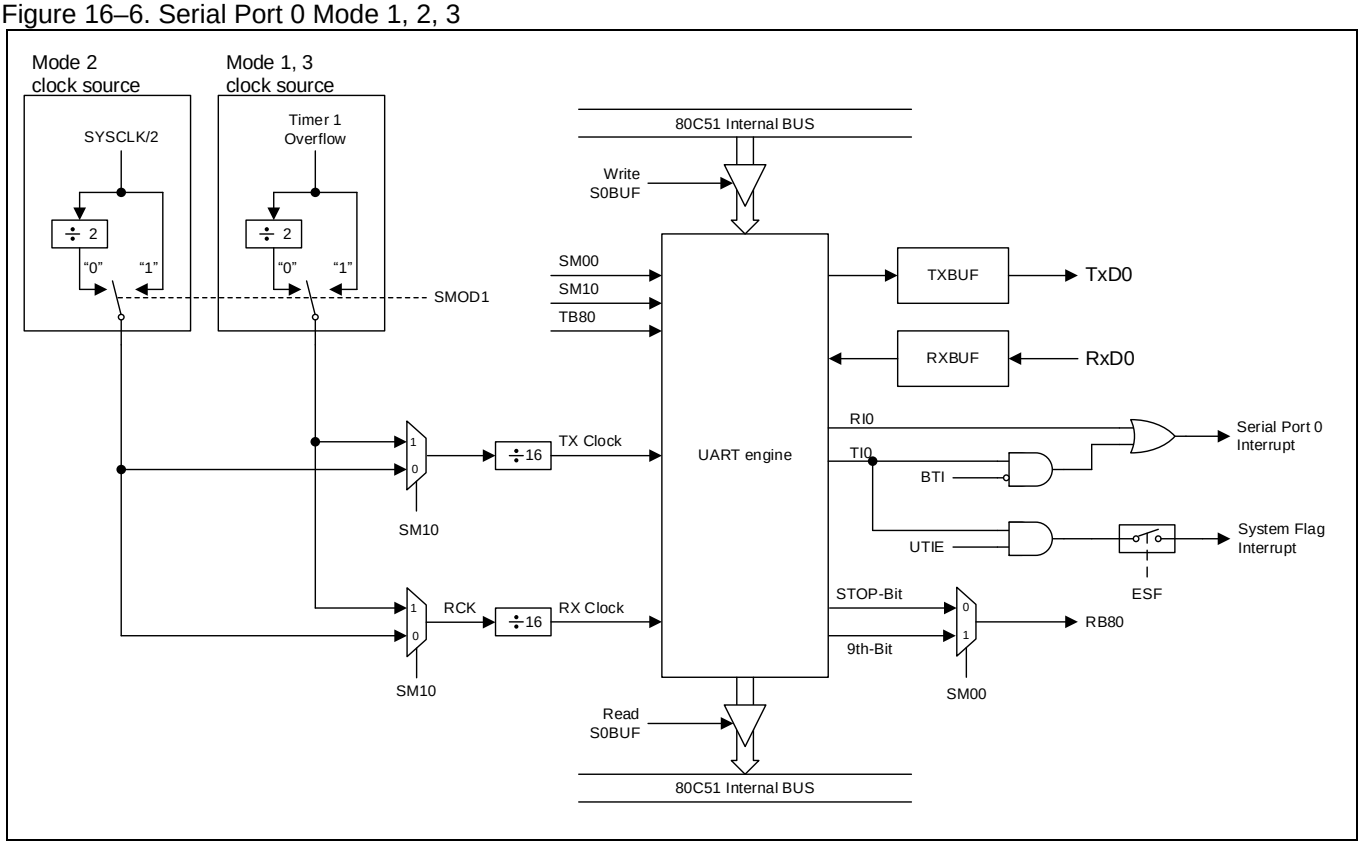
16.2. Serial Port 0 Mode 1

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 overflow rate. Figure 16–1 shows the data frame in Mode 1 and Figure 16–6 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 16–1 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.

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 16–6. Serial Port 0 Mode 1, 2, 3



16.3. Serial Port 0 Mode 2 and Mode 3

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

Figure 16–2 shows the data frame in Mode 2 and Mode 3. Figure 16–6 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 16–2 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.

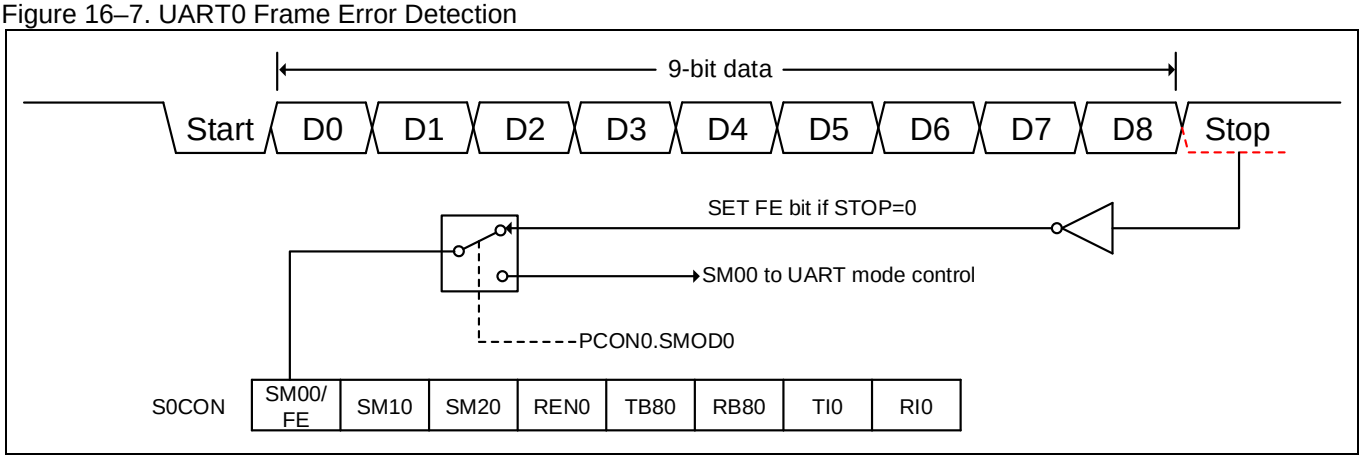
Reception is initiated when the UART0 engine detected 1-to-0 transition at RXD0 sampled by RCK. The data on the RXD0 pin would be sampled by Bit Detector in UART0 engine. After the end of 9th data bit reception, RI0 would be asserted by hardware to indicate the end of data reception and load the 9th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

16.4. 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 (PCON0.6). 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 16–7.

Figure 16–7. UART0 Frame Error Detection



16.5. Baud Rate Setting

Bits T1X12 and URM0X3 in AUXR2 register provide a new option for the baud rate setting, as listed below.

16.5.1. Baud Rate in Mode 0

Figure 16–8. Mode 0 baud rate equation

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

Note:

If URM0X3=0, the baud rate formula is as same as standard 8051.

Table 16–1. Serial Port Mode 0 baud rate example

SYSCLK	URM0X3	Mode 0 Baud Rate
12MHz	0	1M bps
12MHz	1	3M bps
24MHz	0	2M bps
24MHz	1	6M bps

16.5.2. Baud Rate in Mode 2

Figure 16–9. Mode 2 baud rate equation

$$\text{Mode 2 Baud Rate} = \frac{2^{SMOD1} \times 2^{(SMOD2 \times 2)}}{64} \times F_{SYSCLK}$$

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 16–2 defines the Baud Rate setting with SMOD2 factor in Mode 2 baud rate generator.

Table 16–2. Serial Port Mode 2 baud rate example

SYSCLK	SMOD2	SMOD1	Mode 2 Baud Rate	Note
12MHz	0	0	187.5K bps	Default Baud Rate, standard
12MHz	0	1	375K bps	Double Baud Rate, standard
12MHz	1	0	750K bps	Double Baud Rate **X2**, enhanced
12MHz	1	1	--	Reserved

16.5.3. Baud Rate in Mode 1 & 3

Using Timer 1 as the Baud Rate Generator

Figure 16–10. Mode 1/3 baud rate equation

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

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

Note:
If SMOD2=0, T1X12=0, the baud rate formula is as same as standard 8051. If SMOD2=1, there is an enhanced function for baud rate setting. Table 16–3 defines the Baud Rate setting with SMOD2 factor in Timer 1 baud rate generator.

Table 16–3. SMOD2 application criteria in Mode 1 & Mode 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	--	Reserved.	--

Table 16–4 ~ Table 16–7 list various commonly used baud rates and how they can be obtained from Timer 1 in its 8-Bit Auto-Reload Mode.

Table 16–4. Timer 1 Generated Commonly Used Baud Rates @ F_{SYSCLK}=12.0MHz and SMOD2 = 0

Baud Rate	**TH1**, the Reload Value					
	T1X12=0			**T1X12=1**		
	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 16–5. Timer 1 Generated Commonly Used Baud Rates @ F_{SYSCLK}=12.0MHz and SMOD2 = 1

Baud Rate	**TH1**, the Reload Value					
	T1X12=0			**T1X12=1**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
115200	--	--	--	243	--	0.16%
230400	--	--	--	--	--	--

Table 16–6. Timer 1 Generated Commonly Used Baud Rates @ F_{SYSCLK}=24.0MHz and SMOD2 = 0

Baud Rate	**TH1**, the Reload Value					
	T1X12=0			**T1X12=1**		
	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 16–7. Timer 1 Generated Commonly Used Baud Rates @ F_{SYSCLK}=24.0MHz and SMOD2 = 1

Baud Rate	**TH1**, the Reload Value					
	T1X12=0			**T1X12=1**		
	SMOD1=0	SMOD1=1	Error	SMOD1=0	SMOD1=1	Error
230400	--	--	--	243	--	0.16%
460800	--	--	--	--	--	--

16.6. Serial Port 0 Mode 4 (SPI Master)

The Serial Port 0 of **MG74PG1A08** is embedded an additional Mode 4 to support SPI master engine. The Mode 4 is selected by SM30, SM00 and SM10. Table 16–8 shows the serial port 0 mode definition in **MG74PG1A08**.

Table 16–8. 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	SPI Slave	Up to SYSCLK/8
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 **MG74PG1A08** uses the TXD0 as SPICLK, RXD0 as MOSI, and SOMI as MISO. nSS is selected by MCU software on other port pin. Figure 16–11 shows the SPI connection. It also can support the configuration for multiple slaves communication in Figure 16–12.

Figure 16–11. Serial Port 0 Mode 4, Single Master and Single Slave configuration

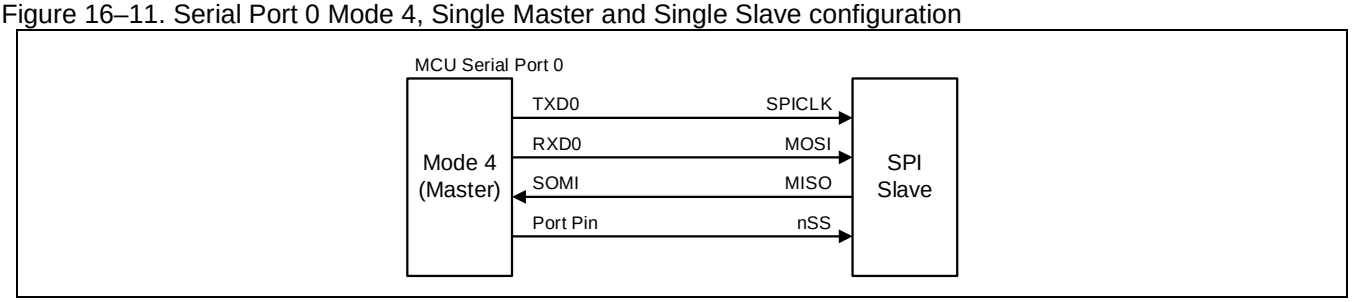
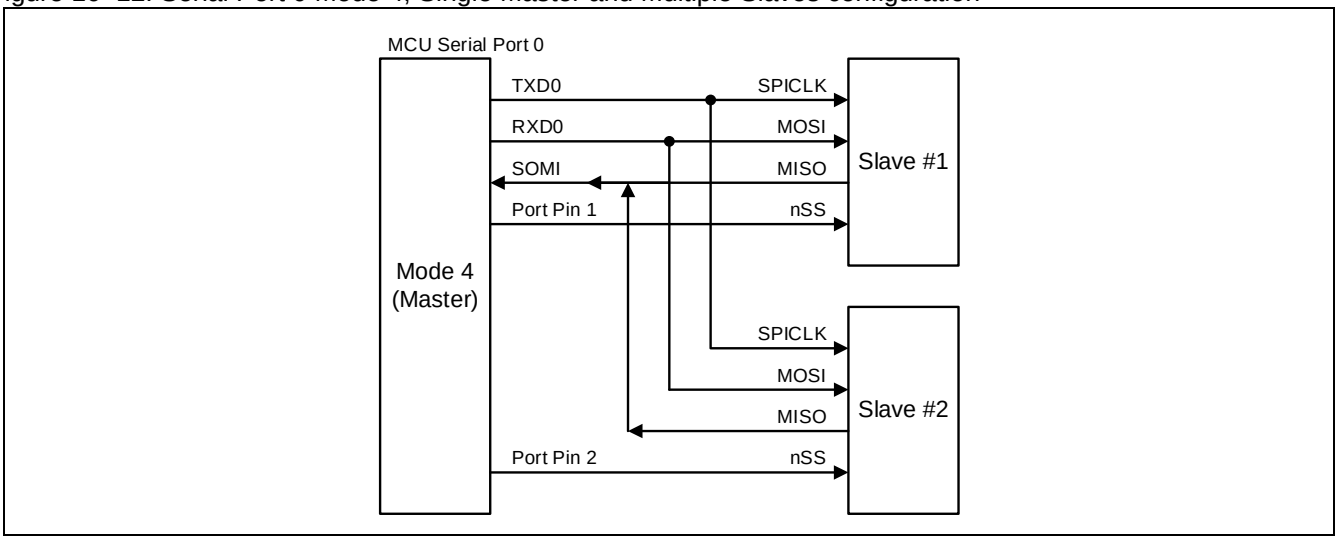


Figure 16–12. Serial Port 0 Mode 4, Single Master and Multiple Slaves configuration



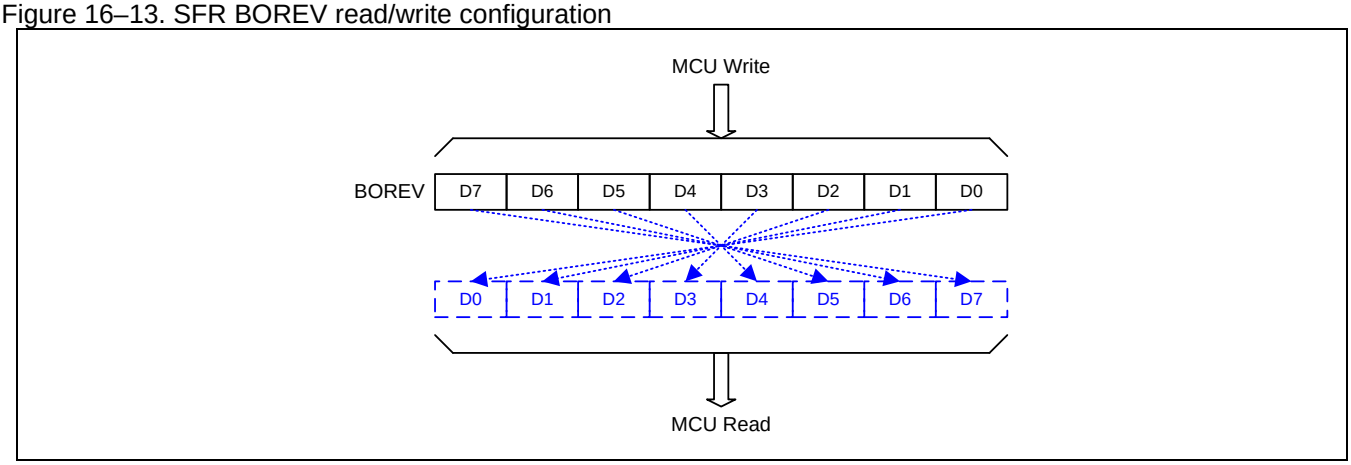
The SPI master satisfies the transfer with the full function SPI module of Megawin MG82/84 series MCU with CPOL, CPHA and DORD selection. For CPOL and CPHA condition, **MG74PG1A08** uses an easy way by initialize SPI clock (TXD0/P3.1) polarity to fit them. Table 16–9 shows the serial port 0 Mode 4 mapping with the four SPI operating mode.

Table 16–9. SPI mode mapping with Serial Port 0 Mode 4 configuration

SPI Mode	CPOL	CPHA	Configuration in **MG74PG1A08**
0	0	0	Clear P3.1 to “0”
1	0	1	Clear P3.1 to “0”
2	1	0	Set P3.1 to “0”
3	1	1	Set P3.1 to “0”

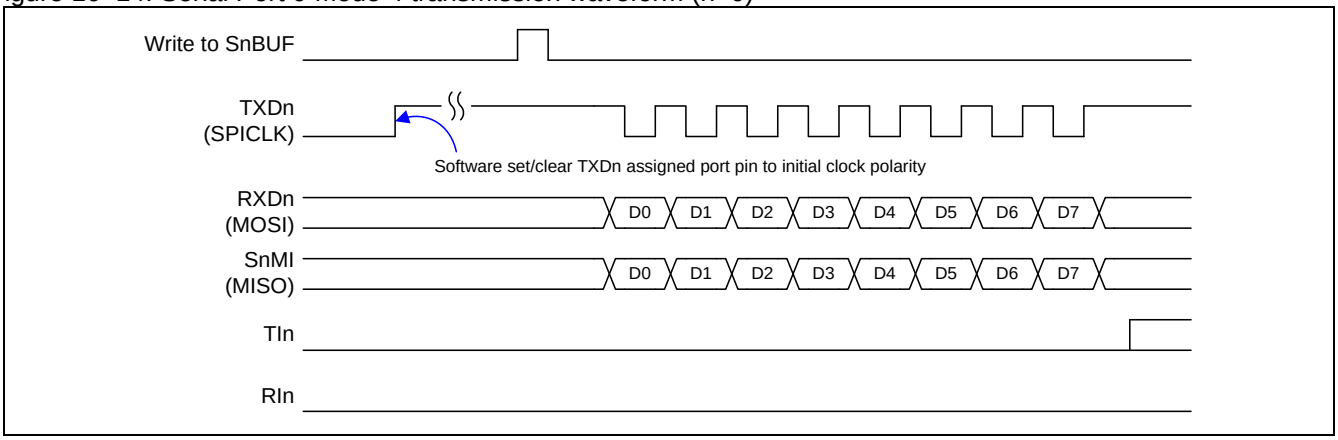
For bit order control (DORD) on SPI serial transfer, **MG74PG1A08** provides a SFR, BOREV, to reverse the bit order by software program. After MCU writing a MSB first data format to BOREV, MCU will get the LSB first data by reading BOREV back. The SPI master engine in serial port 0 Mode 4 is the LSB first transferred which is same as serial port 0 Mode 0. To support SPI MSB first shift, MCU must use the BOREV write/read operation to reverse the data bit order for SPI IN/OUT transmission. Figure 16–13 shows the BOREV configuration.

Figure 16–13. SFR BOREV read/write configuration



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 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 SOMI pin would be sampled and shifted into shift register. Then, “read S0BUF” can get the SPI shift-in data. Figure 16–14 shows the transmission waveform in Mode 0. RI0 will not be asserted in Mode 4.

Figure 16–14. Serial Port 0 Mode 4 transmission waveform (n=0)



16.7. Serial Port 0 Mode 6 (SPI Slave)

The serial port 0 mode 6 in **MG74PG1A08** supports SPI slave mode. The Mode 6 is selected by SM30, SM00 and SM10. Table 16–10 shows the serial port 0 mode definition in **MG74PG1A08**.

Table 16–10. 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**	**SPI Slave**	**Up to SYSCLK/8**
1	1	1	7	Reserved	Reserved

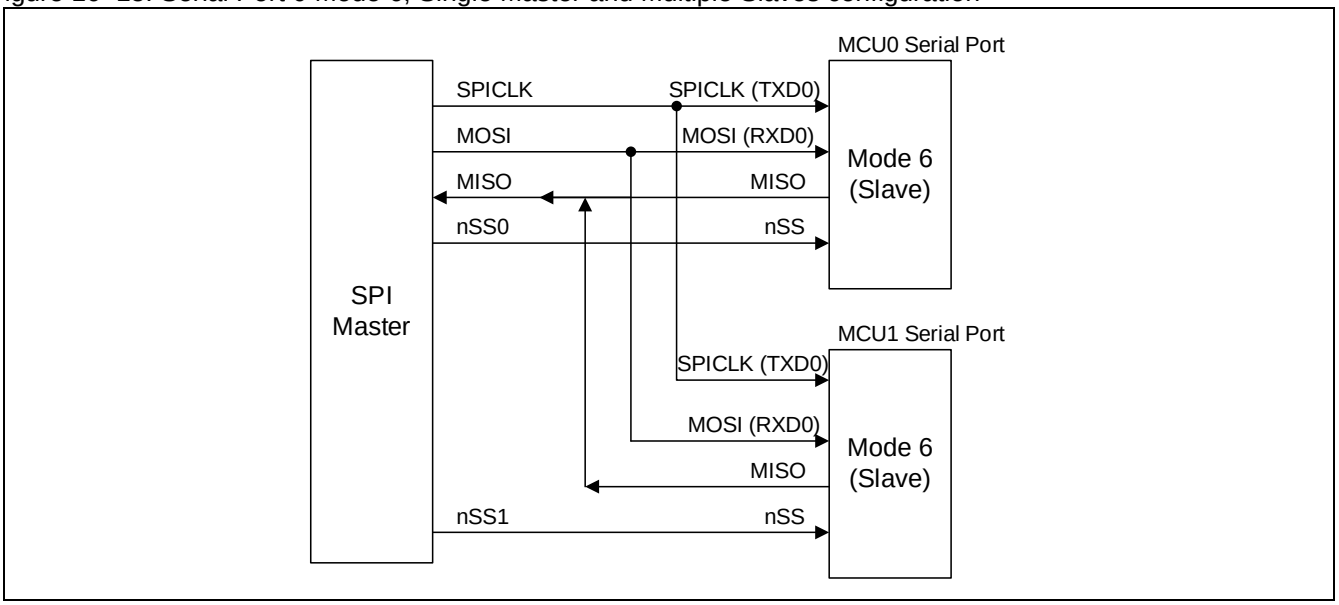
The SPI slave in **MG74PG1A08** uses the TXD0 as SPICLK, RXD0 as MOSI, and a dedicated MISO and nSS. Figure 16–15 shows the SPI connection for multiple slave MCU communication.

The SPI slave engine of serial port 0 mode 6 serves the maximum SPI clock rate up to SYSCLK/8. If SYSCLK = 12MHz, **MG74PG1A08** can receive the maximum frequency of SPICLK is 1.5MHz. This mode also supports the CPHA and SSIG options and the control bits are located on SADEN.1 and SADEN.0. But, there is no CPOL option in this chip. **MG74PG1A08** builds an automatic detection scheme on SPICLK clock polarity in the SPI slave engine.

When CPHA is 0, SSIG must be 0 and nSS pin must be negated and reasserted between each successive serial byte transfer. Note the S0BUF 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.

Figure 16–15. Serial Port 0 Mode 6, Single Master and Multiple Slaves configuration



16.8. Serial Port 0 Register

All the four operation modes of the serial port 0 are the same as those of the standard 8051 except the baud rate setting. Two registers, PCON0 and AUXR2, are related to the baud rate setting:

S0CON: Serial port 0 Control Register

SFR Attribute = Normal Read/Write
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
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	SPI Slave	Up to SYSCLK/8
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 0 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. This bit is also set in mode 4 (SPI master) and mode 6 (SPI slave) after a SPI transfer finished.

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 Attribute = Normal Read/Write

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.

PCON0: Power Control Register 0

SFR Attribute = Normal Read/Write or Protected Write

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 UART0.
1: Enable double Baud rate of the UART0 in mode 1, 2, or 3.

Bit 6: SMOD0, Frame Error select.
0: S0CON.7 is SM00 function.
1: S0CON.7 is FE function. Note that FE will be set after a frame error regardless of the state of SMOD0.

AUXR2: Auxiliary Register 2

SFR Attribute = Normal Read/Write

SFR Address = 0xA3 RESET = 0000-0000

7	6	5	4	3	2	1	0
UTIE/ CPHA	**BTI/ SSIG**	**URM0X3/ SMOD2**	**SM30**	**T1X12**	T0X12	T1CKOE	T0CKOE
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: UART0 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.

In mode 6, SPI slave mode:

Bit 7: CPHA function in serial port 0 mode 6, 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 6: 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.

In mode 6, SPI slave mode:

Bit 6: SSIG function in serial port 0 mode 6.
0: SPI slave engine is enabled by nSS input.
1: SPI slave ignores the nSS input and the engine is controlled by REN0 (S0CON.4). If REN0 = 0, SPI slave is pending. If REN0 = 1, SPI slave is active and shift data on SPICLK edge transition.

Bit 5: URM0X3, Serial Port 0 mode 0 and mode 4 baud rate selector.
0: Clear to select SYSCLK/12 as the baud rate for UART0 Mode 0 and Mode 4.
1: Set to select SYSCLK/4 as the baud rate for UART0 Mode 0 and Mode 4.

In mode 1, 2, 3 UART0 mode:

Bit 5: SMOD2, extra double baud rate selector.
0: Disable extra double baud rate for UART0.
1: Enable extra double baud rate for UART0.

Bit 4: SM30, Serial Port 0 Mode control bit 3.
0: Disable Serial Port 0 Mode 4 & 6.
1: Enable SM30 to control Serial Port 0 Mode 4 & 6, SPI Master & Slave.

Bit 3: T1X12, Timer 1 clock source selector while C/T=0.
0: Clear to select SYSCLK/12.
1: Set to select SYSCLK as the clock source.

BOREV: Bit Order Reversed Register

SFR Attribute = Normal Read/Write
SFR Address = 0x96 RESET = 0000-0000

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

This register serves data read as a Bit-Order **Reversed** function with data written into. Because the serial port 0 engine is always LSB first on transmit/receive. This SFR is used by software to transfer bit order for different SPI format. If the SPI transfer is MSB first, software must write transmitted data to BOREV and read back to get the reversed bit order data. Then software writes the read back data to S0BUF to perform the SPI MSB first transmitting.

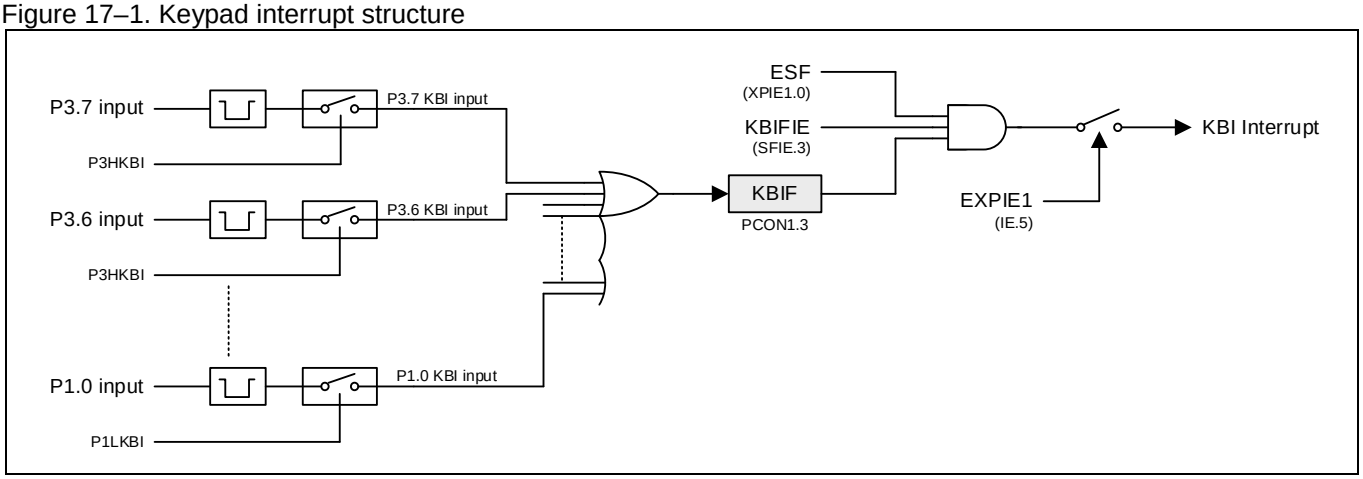
17. Keypad Interrupt (KBI)

The Keypad Interrupt function is intended primarily to allow a single interrupt to be generated when enabled port pin is a low level occurred. This function can be used keypad recognition. Figure 17–1 shows the structure of the keypad interrupt function.

There are two bit registers used for one port on this function. One bit enables the low nibble port pins for keypad interrupt (KBI) function. Another one enables the high nibble port pins. And the port pin operating mode must be configured to open-drain mode, open-drain mode with pull-up or quasi-bidirectional mode and the associated port pin latch is set to “1”. If port pin is configured to push-pull output mode, analog-input-only or output low in open-drain and quasi mode, the KBI on the port pin will be inhibited. Any recognized KBI event will cause the hardware to set the interrupt flag KBIF and generate an interrupt if it has been enabled. Not necessary to enable the KBI interrupt, enabled KBI port pin can wakeup CPU from idle mode (low level) and power-down mode (low level).

17.1. Keypad Interrupt Structure

Figure 17–1. Keypad interrupt structure



17.2. Keypad Interrupt Register

KBIEN0: KBI Enable Control Register 0

SFR Attribute = Normal Read/Write
SFR Address = 0xD6 RESET = 00xx-00xx

7	6	5	4	3	2	1	0
P3HKBI	P3LKBI	--	--	P1HKBI	P1LKBI	--	--
R/W	**R/W**	**W**	**W**	**R/W**	**R/W**	**W**	**W**

Bit 7: P3HKBI, Keypad Input function enable for P3.7 and P3.6.
0: Disable P3.7 and P3.6 KBI function.
1: Enable P3.7 and P3.6 KBI function.

Bit 6: P3LKBI, Keypad Input function enable for P3.2 ~ P3.0.
0: Disable P3.2 ~ P3.0 KBI function.
1: Enable P3.2 ~ P3.0 KBI function.

Bit 5~4: Reserved. Software must write “0” on these bits when KBIEN0 is written.

Bit 3: P1HKBI, Keypad Input function enable for P1.7 ~ P1.4.
0: Disable P1.7 ~ P1.4 KBI function.
1: Enable P1.7 ~ P1.4 KBI function.

Bit 2: P1LKBI, Keypad Input function enable for P1.3 ~ P1.0.
0: Disable P1.3 ~ P1.0 KBI function.
1: Enable P1.3 ~ P1.0 KBI function.

Bit 1~0: Reserved. Software must write “0” on these bits when KBIEN0 is written.

PCON1: Power Control Register 1

SFR Attribute = Normal Read/Write or Protected Write
SFR Address = 0x97 POR = 00xx-0000

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

Bit 3: KBIF, Keypad Interrupt 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 low level on enabled KBI port pin. Writing **“1”** on this bit will clear KBIF.

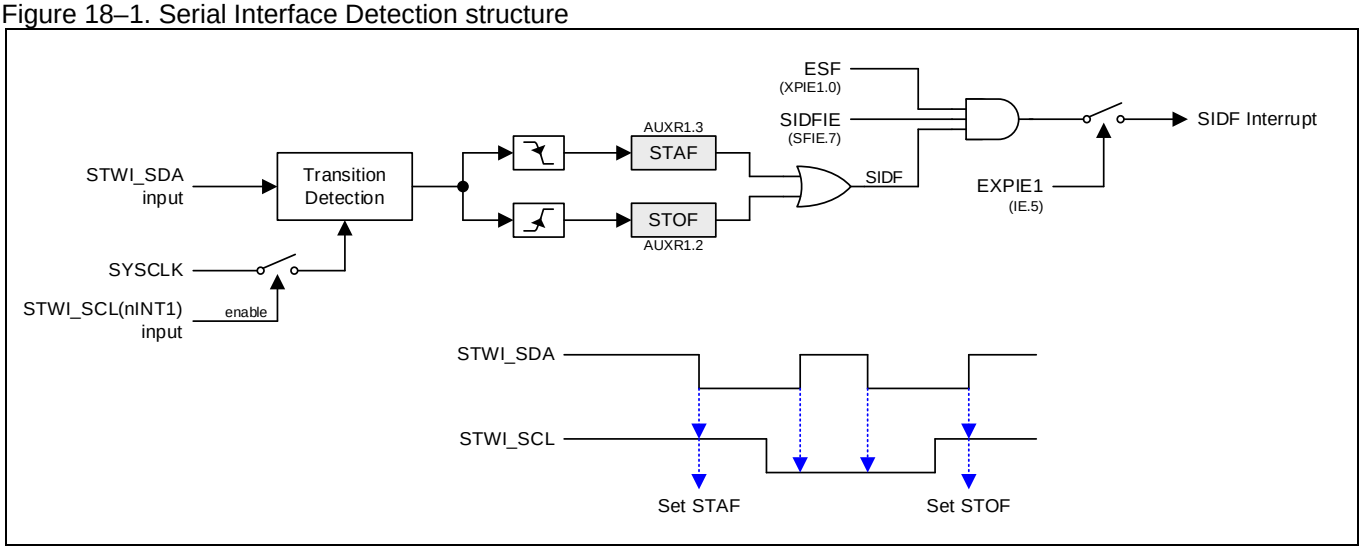
18. Serial Interface Detection (SID/STWI)

The serial interface detection module is always monitoring the “Start” and “Stop” condition on two-wire-interface. 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.

18.1. Serial Interface Detection Structure

Figure 18–1 shows the configuration of STAF and STOF detection, interrupt architecture and event detecting waveform.

Figure 18–1. Serial Interface Detection structure



18.2. Serial Interface Detection Register

AUXR1: Auxiliary Control Register 1

SFR Attribute = Normal Read/Write
SFR Address = 0xA2 RESET = 0000-0000

7	6	5	4	3	2	1	0
INT1IS1	INT1IS0	INT0IS0	GF	**STAF**	**STOF**	GF	GF
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 3: STAF, Start Flag detection of TWI.
0: Clear by firmware by writing “0” on it.
1: Set by hardware to indicate the START condition occurred on TWI bus.

Bit 2: STOF, Stop Flag detection of TWI.
0: Clear by firmware by writing “0” on it.
1: Set by hardware to indicate the START condition occurred on TWI bus.

SFIE: System Flag Interrupt Enable Register

SFR Attribute = Normal Read/Write
SFR Address = 0x8E RESET = 0xxx-0000

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

19. Universal Serial Bus (USB)

MG74PG1A08 implements a USB full-speed function which is fully compliable with USB specification 2.0 and 1.1 to support various USB applications. The USB block contains a on-chip 3.3V regulator, a USB transceiver which transmits and receives differential USB signal, a 32 bytes FIFO which is a temporary store data unit, and a USB Core to perform NRZI encoding and decoding, bit stuffing, CRC generation and checking, serial-parallel data transforming, data flow between 32 bytes FIFO and CPU, USB special function register and setting, and communication with CPU by accessing USBADR/USBDAT in CPU SFRs directly.

Before using **MG74PG1A08** USB function, we assume that user has a comprehensive understanding on USB protocol and application. So, the following descriptions in this chapter would not focus on the detail of USB specification. If user is interesting in USB specification, user can download the latest version of USB specification document from the USB official website http://www.usb.org/home.

Megawin Inc. also offer a development kit which contain sample code, C language library and application note on the website http://www.megawin.com.tw/ to help user to implement design more quickly and easily.

Note:

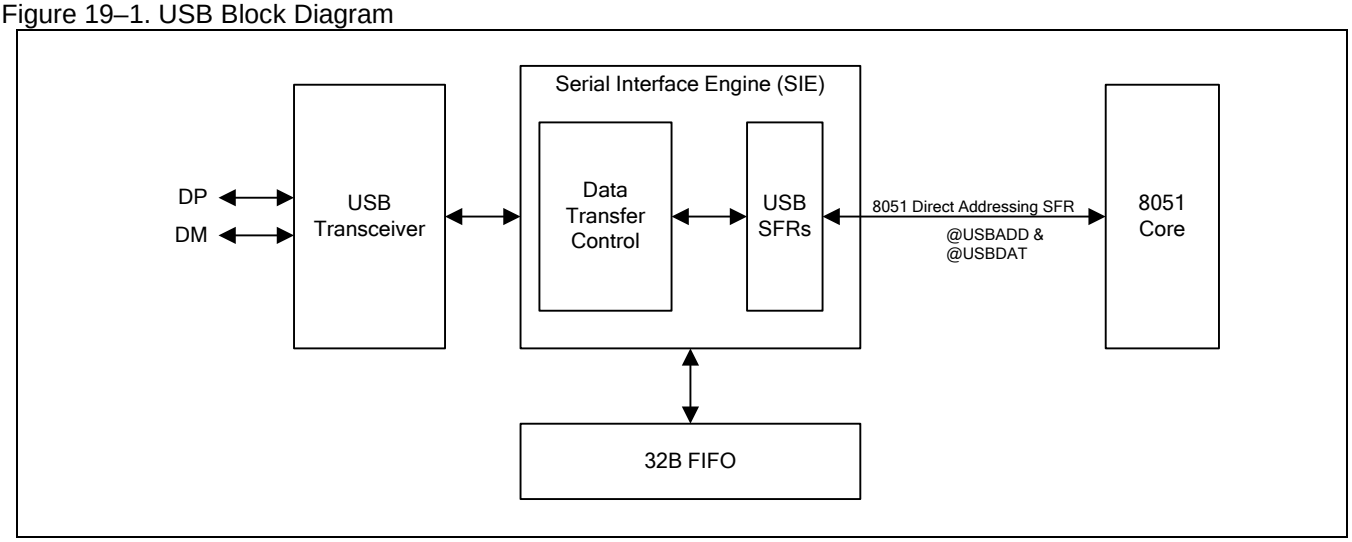
***MG74PG1A08** can't be used as a USB HOST device or USB OTG device.*

19.1. Features

- Compliant with USB specification v1.1/v2.0.
- Supports USB full speed 12M bps serial data transmission
- Supports USB suspend/resume and remote wake-up
- 32 bytes FIFO for USB endpoint-shared buffer
- 8 bytes FIFO for EP0 Control In/Out buffer
- 8 bytes FIFO for EP1 Interrupt/Bulk IN buffer
- 16 bytes FIFO for EP2 Interrupt/Bulk IN/OUT buffer (default is IN)

19.2. Block Diagram

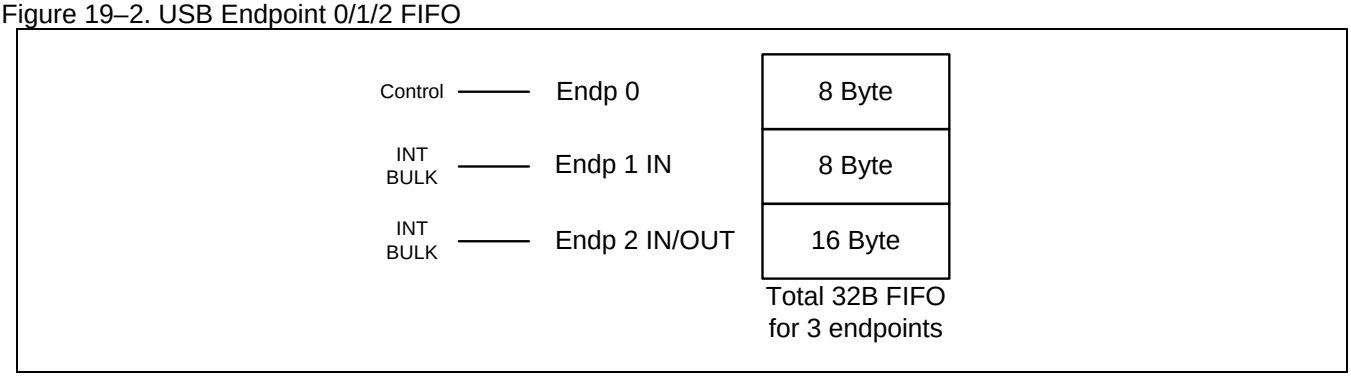
Figure 19–1. USB Block Diagram



19.3. FIFO Management

There are 32 bytes FIFO for temporary USB data store unit accessed by USB core and a total of 3 endpoints are available in **MG74PG1A08** as shown in Figure 19–2 Endpoint 0 supports a bi-direction control transfer. Endpoint 1/2 supports Interrupt/Bulk IN transaction. The maximum data packet size can be up to 8 bytes for endpoint 0 Control function, 8 bytes for endpoint 1 IN function, 16 bytes for endpoint 2 IN/OUT function.

Figure 19–2. USB Endpoint 0/1/2 FIFO



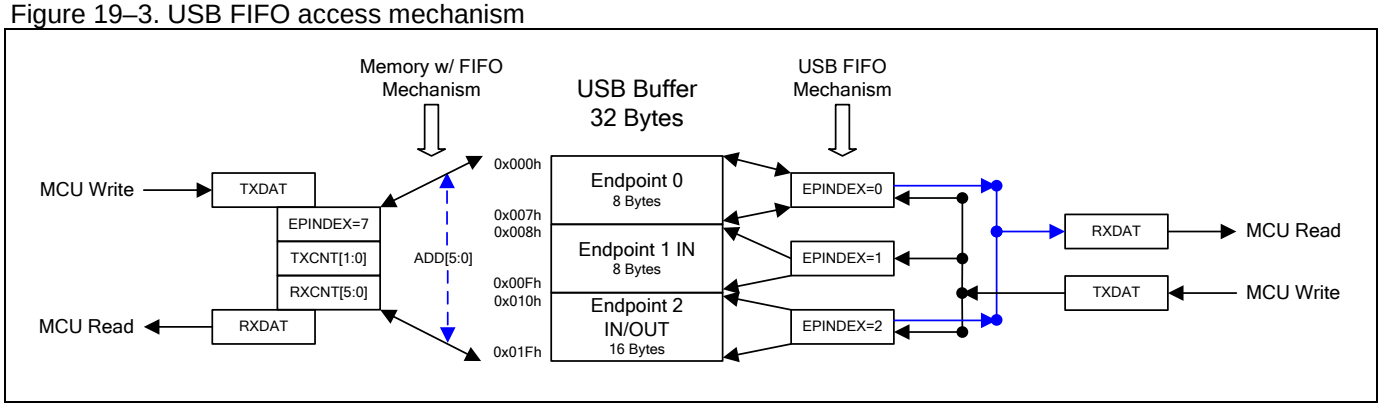
19.4. Access USB 32 bytes FIFO

If ENUSB is set to enable USB function, a 32 bytes FIFO would be a dedicated buffer for USB application. But in most application case, it is not necessary and wasted that all 32 bytes FIFO would reserve for USB buffer, the unused USB buffer can provide MCU additional data RAM which is access through USB re-directed mechanism. Figure 19–4 shows the USB FIFO access mechanism to share the USB FIFO for MCU application.

Access USB buffer through USB SFR re-direct flow:
Write data into USB buffer
1. Write EPINDEX=7
2. Write RXCNT (directly mapping RXCNT[5:0]} to 32B buffer Address space)
3. Write data into TXDAT
4. Repeat to step 2 (step 2 & 3 can be skipped, if the next buffer address is increased)

Read data from USB buffer
1. Write EPINDEX=7
2. Write RXCNT (directly mapping {RXCNT[5:0]} to 32B buffer Address space)
3. Read data from RXDAT
4. Repeat to step 2 (step 2 & 3 can be skipped, if the next buffer address is increased)

Figure 19–3. USB FIFO access mechanism



19.5. USB Initial

To activate the USB operation, the user should enable Clock Multiplier (CKM) unit by setting ENCKM bit, wait 100us for CKM ready to work, and then enable USB function by setting ENUSB bit. Clearing bit 'ENUSB' will deactivate the USB operation and let the USB function enter its power-down mode. These relevant control bits are contained in the CKCON0 register, as follows.

DCON0: Device Control 0

SFR Attribute = Normal Read and Protected Write
SFR Address = 0xBC RESET = x100-0110

7	6	5	4	3	2	1	0
WCKS	**USBR**	ENUSB	ENCKM	CKMIS1	CKMIS0	RSTIO	SWRST
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 6: USBR, Software trigger to reset USB function
0: Software end the reset of USB function.
1: Software start the reset of USB function.

Bit 5: ENUSB, Enable USB clock and whole USB function.
0: Disable USB clock and USB function.
1: Enable USB clock and USB function.

Bit 4: ENCKM, Enable clock multiplier (X8)
0: Disable the X8 clock multiplier.
1: Enable the X8 clock multiplier.

Bit 3~2: CKMIS1 ~ CKMIS0, Multiplier Input Clock Selection.

CKMIS[1:0]	Multiplier Input Clock Selection
0 0	6MHz input
0 1	12MHz input
1 0	24MHz input
1 1	36MHz input

Note:

Before using USB function, the setting of CKM must be correct to provide proper clock for USB communication. Please refer Section "8 System Clock" to get the information about the setting of CKM.

19.6. Access USB SFR

Table 19–1 Table 19–1shows the USB SFR and their indirect address from C0H~FFH. USB SFR can be indirectly accessed according to a 6-bit address hold in USBADR. Read/Write USBDAT will target the register indicated by USBADR.

USB Write SFR procedure

1. Wait for UBSY==0
2. Write USB SFR address into USBADR
3. Write data into USBDAT
4. Repeat to step1 to write next data(step 2 can be skipped, when writing to the same USB SFR address)

USB Read SFR procedure

1. Wait for UBSY==0
2. Write USB SFR address into USBADR
3. Read data from USBDAT
4. Repeat to step 1 to read next data(step 2 can be skipped, when reading from the same USB SFR address)

Table 3–1

USBADR: USB indirect Address Register

SFR Page = 0~F
SFR Address = 0xAB POR+RESET = 0x00-0000

7	6	5	4	3	2	1	0
UBSY	--	USFRA5	USFRA4	USFRA3	USFRA2	USFRA1	USFRA0
R	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: UBSY, USB core BUSY flag.
0: The data access request in USB core is finished.
1: USB core is **BUSY** on accessing software read/write request.

Bit 5~0: USFRA[5:0], USB SFR indirect address.

USBDAT: USB Data Register

SFR Page = 0~F
SFR Address = 0xAA POR+RESET = xxxx-xxxx

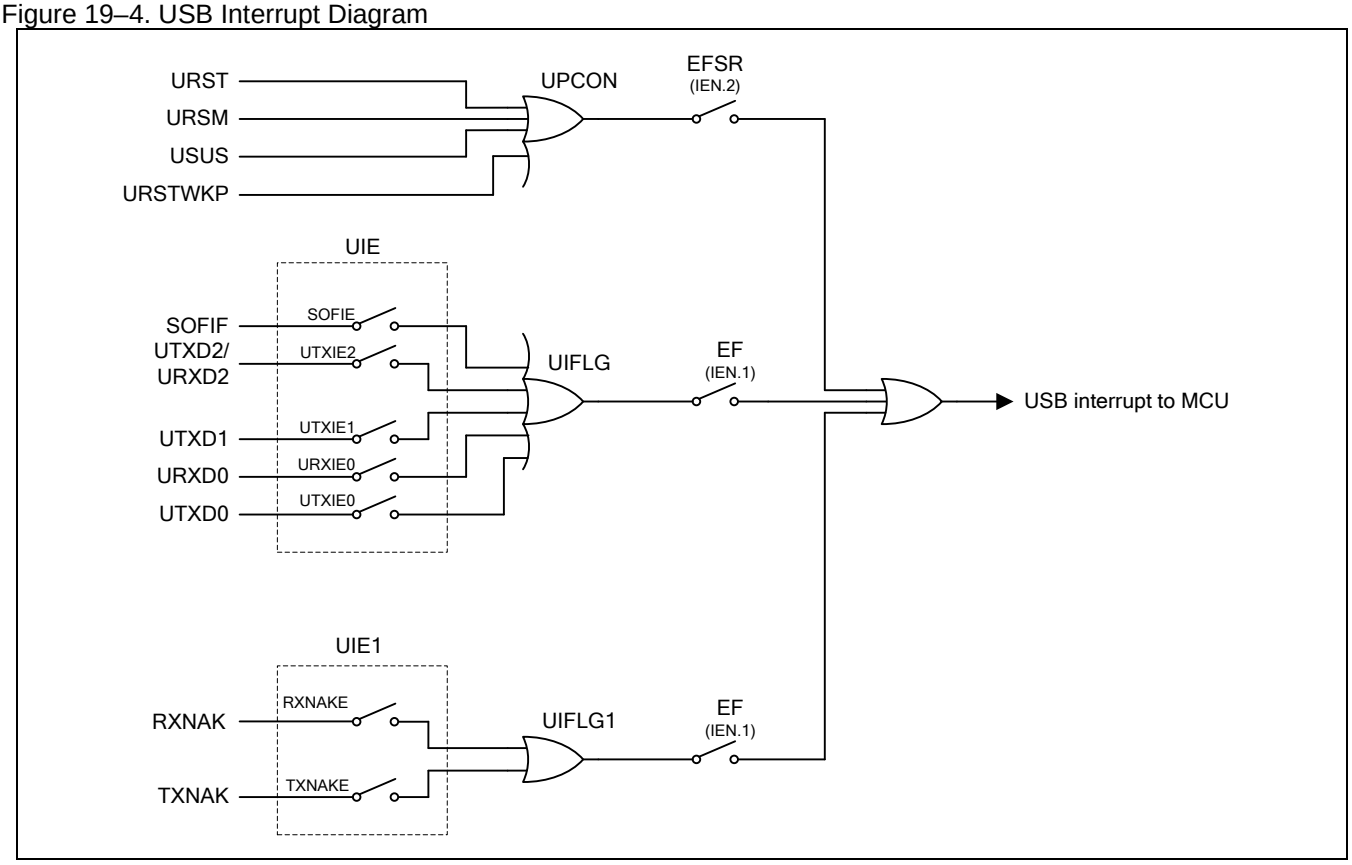
7	6	5	4	3	2	1	0
UDAT7	UDAT6	UDAT5	UDAT4	UDAT3	UDAT2	UDAT1	UDAT0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~0: UDAT[7:0], USB SFR Data.

19.7. USB Interrupt

Figure 19–5 shows the USB interrupt structure and there are 11 interrupt flags which are located in USB SFRs shown in Section "19.9.1 USB Function SFR Bit Assignment". The USB interrupt is generated on the combination of USB event flags and USB endpoint flags contained in USB SFRs. The USB event flags include USB reset flag (URST), USB resume flag (URSM), USB suspend flag (USUS) and USB reset wakeup flag (URSTWKP) can indicate that the upstream host has sent the USB reset, resume or suspend event on USB bus to device. The USB endpoint flags, as UTXDx and URXDx (x=0~2), show the USB data transmission or reception of respective endpoint had been done by USB transceiver. The associated interrupt enable bits are located in UIE, UIE1 and IEN registers.

Figure 19–4. USB Interrupt Diagram



19.8. USB Special Function Registers

All USB SFRs would be reset by the reset sources as listed in Section "10 System Reset" (SYSRST) and the most of USB SFRs would be reset when device receives the USB reset event (USBRST) except DCON0, DCON1, IEN, SIOCTL registers and CONEN bit in UPCON register.

19.9. USB Function SFR Mapping

Table 19–1. USB Function SFR Mapping

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
xxF8H	--	--	--	--	--	--	--	--	xxFFH
xxF0H	--	**EPINDEX**	**TXSTAT**	**TXDAT**	**TXCON**	--	**TXCNT**	--	xxF7H
xxE8H	--	--	--	--	--	--	--	--	xxEFH
xxE0H	--	**EPCON**	**RXSTAT**	**RXDAT**	**RXCON**	--	**RXCNT**	--	xxE7H
xxD8H	**UADDR**	**IEN**	**UIE**	**UIFLG**	**UIE1**	**UIFLG1**			xxDFH
xxD0H	--	--	--	--	--	--	--	--	xxD7H
xxC8H	--	**UPCON**	--	--	--	--	--	--	xxCFH
xxC0H	**UDCON0**	**UDCON1**	**SIOCTL**	--	--	--	--	--	xxC7H
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

19.9.1. USB Function SFR Bit Assignment

Table 19–2. USB Function SFR Bit Assignment

SYMBOL	DESCRIPTION	ADDR	BIT SYMBOL								RESET VALUE
			Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	
UDCON0	USB Device Control Register 0	C0H	EP2DIR	--	SCWKP	FRST	--	--	--	--	0x00xxxxB
UDCON1	USB Device Control Register 1	C1H	--	--	SETNO	STLDEN	NAKEP1	NAKEP0	RPDEN	--	xx00000xB
UADDR	USB Address Register	D8H	--	UADD6	UADD5	UADD4	UADD3	UADD2	UADD1	UADD0	x0000000B
UPCON	USB Power Control Register	C9H	CONEN	--	URWU	--	URSTWKP	URST	URSM	USUS	0x0x0000B
IEN	Interrupt Enable Register	D9H	--	--	--	--	--	EFSR	EF	--	xxxxx00xB
UIE	USB Interrupt Enable Register	DAH	SOFIE	--	--	UTXIE2/ URXIE2	--	UTXIE1	URXIE0	UTXIE0	0xx0x000B
UIFLG	USB Interrupt Flag Register	DBH	SOFIF	--	--	UTXD2/ URXD2	--	UTXD1	URXD0	UTXD0	0xx0x000B
UIE1	USB Interrupt Enable Register 1	DCH	RXNAKE	TXNAKE	--	--	--	--	--	--	00xxxxx0B
UIFLG1	USB Interrupt Flag Register 1	DDH	RXNAK	TXNAK	--	--	--	--	--	--	00xxxxx0B
EPINDEX	Endpoint Index Register	F1H	--	--	--	--	--	EPINX2	EPINX1	EPINX0	xxxxx000B
EPCON	Endpoint Control Register	E1H	RXSTL	TXSTL	--	--	--	RXEPEN	--	TXEPEN	00x0x101B
RXSTAT	Endpoint Receive Status Register	E2H	RXSEQ	RXSETUP	STOVW	EDOVW	RXSOVW	--	--	--	000000xxB
RXDAT	FIFO Receive Data Register	E3H	RXD7	RXD6	RXD5	RXD4	RXD3	RXD2	RXD1	RXD0	xxxxxxxxB
RXCON	FIFO Receive Control Register	E4H	RXCLR	--	--	RXFFRC	--	--	--	--	0xx0xxxxB
RXCNT	FIFO Receive Byte Count Register	E6H	RXBC7	RXBC6	RXBC5	RXBC4	RXBC3	RXBC2	RXBC1	RXBC0	00000000B
TXSTAT	Endpoint Transmit Status Register	F2H	TXSEQ	--	--	--	TXSOVW	--	--	--	0xxx0xxxB
TXDAT	FIFO Transmit Data Register	F3H	TXD7	TXD6	TXD5	TXD4	TXD3	TXD2	TXD1	TXD0	xxxxxxxxB
TXCON	FIFO Transmit Control Register	F4H	TXCLR	--	--	TXFFRC	--	--	--	--	0xx0xxxxB
TXCNT	FIFO Transmit Byte Count Register	F6H	TXBC7	TXBC6	TXBC5	TXBC4	TXBC3	TXBC2	TXBC1	TXBC0	xxxxxxxxB
SIOCTL	Serial I/O Control Register	C2H	DPI	DMI	--	--	--	--	--	--	xxxxxxxxB

UDCON0: USB Device Control Register 0

SFR Address = 0xC0H SYSRST= 0000-0000

7	6	5	4	3	2	1	0
EP2DIR	--	SCWKP	FRST	--	--	--	--
R/W	W	R/W	R/W	W	W	W	W

Bit 7: EP2DIR-- Endpoint 2 Direction select
0: Endpoint 2 will be configured for IN function.
1: Endpoint 2 will be configured for OUT function.

Bit 6: Reserved. Software must write “0” on this bit when UDCON0 is written.

Bit 5: SCWKP, Software Control Remote-wakeup.
0: Remote-wakeup length will be decided by hardware setting.
1: Remote-wakeup length will be decided by URWU value.

Bit 4: FRST-- Function interface Reset Flag of USB device.
Set by hardware when the **MG74PG1A08** detects the USB device interface in USB reset duration. If this bit is set, the chip will not generate an interrupt to uC. It would be cleared by firmware writing '1' to it.

Bit 3~0: Reserved. Software must write “0” on this bit when UDCON0 is written.

UDCON1: USB Device Control Register 1

SFR Address = 0xC1H SYSRST= 0000-000X

7	6	5	4	3	2	1	0
--	--	SETNO	STLDEN	NAKEP1	NAKEP0	RPDEN	--
W	W	R/W	R/W	R/W	R/W	R/W	W

Bit 7~6: Reserved. Software must write “0” on this bit when UDCON1 is written.

Bit 5: SETNO-- Set No-response in EP0 IN/OUT transaction.
0: Device will send ACK/NAK/STALL packet in IN/OUT transaction.
1: Device will be just only response ACK packet with SETUP transaction but no response with EP0 IN/OUT transaction.

Note:
This bit will be clear by HW when Device receive an SETUP token.

Bit 4: STLDEN-- STALL Done enable.
0: Disable IN/OUT STALL transaction flag setting.
1: IN/OUT STALL transaction will set TXD0/RXD0 in FIFLG.

Bit 3~2: NAKEP[1:0]-- Endpoint NAK done select.
00: RXNAK and TXNAK dedicated for endpoint 0
01:TXNAK dedicated for endpoint 1
10:TXNAK dedicated for endpoint 2
11: Reserved.

Bit 1: RPDEN, two individual pull-down Resistor enabled on DP and DM.
0: Disable the pull-down resistors on DP and DM.
1: Enable the pull-down resistors on DP and DM. The resistance is about 500K Ohm.

Bit 0: Reserved. Software must write “0” on this bit when UDCON1 is written.

UADDR: USB Function Address Register

SFR Address = 0xD8H SYSRST/USBRST= X000-0000

7	6	5	4	3	2	1	0
--	UADD6	UADD5	UADD4	UADD3	UADD2	UADD1	UADD0
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 UADDR is written.

Bit 6~0: UADD[6:0]-- USB Function Address.
This register holds the address for the USB function. During bus enumeration, it is written with a unique value assigned by the host.

UPCON: USB Power Control Register

SFR Address = 0xC9H SYSRST/USBRST= 0X0X-X000

7	6	5	4	3	2	1	0
CONEN	--	URWU	--	URSTWKP	URST	URSM	USUS
R/W	**W**	**R/W**	**W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: CONEN-- USB Connect Enable.
Default is cleared to '0' after reset. FW should set '1' to enable connection to upper host/hub.

Bit 6: Reserved. Software must write “0” on this bit when UPCON is written.

Bit 5: URWU-- USB Remote Wake-Up Trigger.
0: If SCWKP=0, this bit will be cleared by hardware when remote-wakeup is completed. If SCWKP=1, this bit will be cleared by firmware to stop device driving a remote wake-up on the USB bus. Don't set this bit unless the function is suspended
1: This bit is set by the firmware to initiate a remote wake-up on the USB bus when CPU is wake-up by external trigger.

Note:
Set by firmware to make driving resume signaling to the host. Don't set this bit unless the function is suspended (USUS=1 and URSM=0).

SCWKP	Device resume length
0	The resume signal period which device drive is about 5~6T, **T=1.172ms.**
1	The resume signal period is which start by FW set URWU and end by FW clear URWU.

Bit 4: Reserved. Software must write “0” on this bit when UPCON is written.

Bit 3: URSTWKP—USB Reset wakeup Flag.
During suspend, set by hardware when the function detects the USB bus reset. If this bit is set, the chip will generate an URSTWKP interrupt to uC. It would be cleared by firmware when serving the USB reset wakeup interrupt. This bit is cleared when firmware writes '1' to it.

Bit 2: URST-- USB Reset Flag.
Set by hardware when the function detects the USB bus reset. If this bit is set, the chip will generate an USRT interrupt to uC. It would be cleared by firmware when serving the USB reset interrupt. This bit is cleared when firmware writes '1' to it.

Bit 1: URSM-- USB Resume Flag.
Set by hardware when the function detects the resume state on the USB bus. If this bit is set, the chip will generate an interrupt to uC. It would be cleared by firmware when serving the function resume interrupt. This bit is cleared when firmware writes '1' to it.

Bit 0: USUS-- USB Suspend Flag.
Set by hardware when the function detects the suspend state on the USB bus. If this bit is set, the chip will generate an interrupt to uC. During the function suspend interrupt-service routine, firmware should clear this bit before enter the suspend mode. This bit is cleared when firmware writes '1' to it.

IEN: Interrupt Enable Register

SFR Address = 0xD9H SYSRST= XXXX-X00X

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

Bit 7~3: Reserved. Software must write “0” on these bits when IEN is written.

Bit 2: EFSR-- Enable USB Function’s Suspend/Resume interrupt.
If this bit is set, enables function’s interrupt of FPCON events. Function suspend/resume/remote-wakeup/USB-reset interrupt enable bit. This bit doesn't be reset USB_RESET. Default is cleared.

Bit 1: EF-- Enable USB Function’s interrupt Flag.
If this bit is set, enables function’s interrupt of UIFLG. Transmit/receive done interrupt enable bit for USB function endpoints. This bit doesn't be reset by USB_RESET. Default is cleared.

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

UIE: USB Interrupt Enable Register

SFR Address = 0xDAH SYSRST/USBRST= 00X0-X000

7	6	5	4	3	2	1	0
SOFIE	--	--	UTXIE2/ URXIE2	--	UTXIE1	URXIE0	UTXIE0
R/W	**W**	**W**	**R/W**	**W**	**R/W**	**R/W**	**R/W**

Bit 7: SOFIE-- Host SOF received Interrupt Enable.
If this bit is set, enables the Host SOF received interrupt. Default is cleared.

Bit 6~5: Reserved. Software must write “0” on this bit when UIE is written.

Bit 4: UTXIE2/URXIE2-- USB Function Transmit Interrupt Enable 2.
If this bit is set, enables the transmit and receive done interrupt for USB endpoint 2 (UTXD2/URXD2). Default is cleared.

UTXIE2 - Enable UIFLG.UTXD2 Interrupt. (Default DCON.EP2DIR=0)

URXIE2 - Enable UIFLG.URXD2 Interrupt. (DCON.EP2DIR=1)

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

Bit 2: UTXIE1-- USB Function Transmit Interrupt Enable 1.
If this bit is set, enables the transmit done interrupt for USB endpoint 1 (UTXD1). Default is cleared.

Bit 1: URXIE0-- USB Function Receive Interrupt Enable 0.
If this bit is set, enables the receive done interrupt for USB endpoint 0 (URXD0). Default is cleared.

Bit 0: UTXIE0-- USB Function Transmit Interrupt Enable 0.
If this bit is set, enables the transmit done interrupt for USB endpoint 0 (UTXD0). Default is cleared.

UIFLG: USB Interrupt Flag Register

SFR Address = 0xDBH SYSRST/USBRST= 00X0-X000

7	6	5	4	3	2	1	0
SOFIF	--	--	UTXD2/ URXD2	--	UTXD1	URXD0	UTXD0
R/W	**W**	**W**	**R/W**	**W**	**R/W**	**R/W**	**R/W**

Bit 7: SOFIF-- Host SOF received Interrupt Flag.
This bit is set by hardware when detected a host SOF. UC can read/write-clear on this bit. This bit is cleared when firmware writes '1' to it.

Bit 6~5: Reserved. Software must write “0” on this bit when UIFLG is written.

Bit 4: UTXD2/URXD2-- USB Transmit and Receive Done Flag for endpoint 2.
This bit is set by hardware when detected a transmit and receive done on endpoint 2. UC can read/write-clear on this bit. This bit is cleared when firmware writes '1' to it.

UTXD2 -- Endpoint 2 Transmit done flag. (Default DCON.EP2DIR=0)

URXD2 -- Endpoint 2 Receive done flag. (DCON.EP2DIR=1)

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

Bit 2: UTXD1-- USB Transmit Done Flag for endpoint 1.
This bit is set by hardware when detected a transmit done on endpoint 1. UC can read/write-clear on this bit. This bit is cleared when firmware writes '1' to it.

Bit 1: URXD0-- USB Receive Done Flag for endpoint 0.
This bit is set by hardware when detected a receive done on endpoint 0. UC can read/write-clear on this bit. This bit is cleared when firmware writes '1' to it.

Bit 0: UTXD0-- USB Transmit Done Flag for endpoint 0.
This bit is set by hardware when detected a transmit done on endpoint 0. UC can read/write-clear on this bit. This bit is cleared when firmware writes '1' to it.

UIE1: USB Interrupt Enable Register 1

SFR Address = 0xDCH SYSRST/USBRST= 00XX-XXX0

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

Bit 7: RXNAKE-- Enable RX NAK interrupt on NAKEP[1:0] indexed.
If this bit is set, enables the RXNAK interrupt for NAKEP[1:0] indexed endpoint. Default is cleared.

Bit 6: TXNAKE-- Enable TX NAK interrupt on NAKEP[1:0] indexed.
If this bit is set, enables the TXNAK interrupt for NAKEP[1:0] indexed endpoint. Default is cleared.

Bit 5~0: Reserved. Software must write “0” on this bit when UIE1 is written.

UIFLG1: USB Interrupt Flag Register 1

SFR Address = 0xDDH SYSRST/USBRST= 00XX-XXX0

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

Bit 7: RXNAK-- RX NAK Flag on NAKEP[1:0] indexed.
This bit is set by hardware when detected a receive done on the NAK packet for OUT transaction of the NAKEP[1:0] indexed endpoint. This bit is clear when firmware write "1" to it.

Bit 6: TXNAK-- TX NAK Flag on NAKEP[1:0] indexed.
This bit is set by hardware when detected a transmit done on the NAK packet for IN transaction of the NAKEP[1:0] indexed endpoint. This bit is clear when firmware write "1" to it.

Bit 5~0: Reserved. Software must write "0" on this bit when UIFLG1 is written.

EPINDEX: Endpoint Index Register

SFR Address = 0xF1H SYSRST/USBRST= XXXX-X000

7	6	5	4	3	2	1	0
--	--	--	--	--	EPINX2	EPINX1	EPINX0
W	**W**	**W**	**W**	**W**	**R/W**	**R/W**	**R/W**

Bit 7~3: Reserved. Software must write "0" on these bits when EPINDEX is written.

Bit 2~0: EPINX[2:0]-- Endpoint Index Bits [2:0]
3'b000: USB Function Endpoint 0.
3'b001: USB Function Endpoint 1.
3'b010: USB Function Endpoint 2.
3'b011~110: Reserved.
3'b111: uC access 32 byte buffer enable.

EPCON: Endpoint Control Register *(endpoint-indexed)*

SFR Address = 0xE1H SYSRST/USBRST= 00X0-X101

7	6	5	4	3	2	1	0
RXSTL	TXSTL	--	--	--	RXEPEN	--	TXEPEN
R/W	**R/W**	**W**	**W**	**W**	**R/W**	**W**	**R/W**

Bit 7: RXSTL-- Receive Endpoint Stall.
Set this bit to stall the receive endpoint.

Note:
Clear this bit only when the host has intervened through commands sent down endpoint 0. When this bit is set and RXSETUP is clear, the receive endpoint will respond with a STALL handshake to a valid OUT token. When this bit is set and RXSETUP is set, the receive endpoint will NAK. This bit does not affect the reception of SETUP tokens by a control endpoint.

Bit 6: TXSTL-- Transmit Endpoint Stall.
Set this bit to stall the transmit endpoint.

Note:
Clear this bit only when the host has intervened through commands sent down endpoint 0. When this bit is set and RXSETUP is clear, the transmit endpoint will respond with a STALL handshake to a valid IN token. When this bit is set and RXSETUP is set, the transmit endpoint will NAK.

Bit 5~3: Reserved. Software must write "0" on this bit when EPCON is written.

Bit 2: RXEPEN-- Receive Endpoint Enable.
Set this bit to enable the receive endpoint. When disabled, the endpoint does not respond to a valid OUT or SETUP token. This bit in endpoint 0 is enabled after reset. This bit has the highest priority than RXSTL.

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

Bit 0: TXEPEN-- Transmit Endpoint Enable.
Set this bit to enable the transmit endpoint. When disabled, the endpoint does not respond to a valid IN token. This bit in endpoint 0 is enabled after reset. This bit has the highest priority than TXSTL.

RXSTAT: Endpoint Receive Status Register *(endpoint-indexed)*
SFR Address = 0xE2H SYSRST/USBRST= 0000-00XX

7	6	5	4	3	2	1	0
RXSEQ	RXSETUP	STOVW	EDOVW	RXSOVW	--	--	--
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**W**	**W**	**W**

Bit 7: RXSEQ-- Receive Endpoint Sequence Bit (read, conditional write).
The bit will be toggled on completion of an ACK handshake in response to an OUT token. This bit can be written by firmware if the RXOVW bit is set when written along with the new RXSEQ value.

Bit 6: RXSETUP-- Received Setup Transaction.
This bit is set by hardware when a valid SETUP transaction has been received. Clear this bit upon detection of a SETUP transaction or the firmware is ready to handle the data/status stage of control transfer.

Bit 5: STOVW-- Start Overwrite Flag (read-only).
Set by hardware upon receipt of a SETUP token for the control endpoint to indicate that the receive FIFO is being overwritten with new SETUP data. This bit is used only for control endpoints.

Bit 4: EDOVW-- End Overwrite Flag.
This flag is set by hardware during the handshake phase of a SETUP transaction. This bit is cleared by firmware to read the FIFO data. This bit is only used for control endpoints.

Bit 3: RXSOVW-- Receive Data Sequence Overwrite Bit.
Write '1' to this bit to allow the value of the RXSEQ bit to be overwritten. Writing a '0' to this bit has no effect on RXSEQ. This bit always returns '0' when read.

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

RXDAT: Receive FIFO Data Register *(endpoint-indexed)*
SFR Address = 0xE3H SYSRST/USBRST= XXXX-XXXX

7	6	5	4	3	2	1	0
RXD7	RXD6	RXD5	RXD4	RXD3	RXD2	RXD1	RXD0
R	**R**	**R**	**R**	**R**	**R**	**R**	**R**

Bit 7~0: RXD[7:0]-- Receive FIFO Data.
Receive FIFO data specified by EPINDEX is stored and read from this register.

RXCON: Receive FIFO Control Register *(endpoint-indexed)*

SFR Address = 0xE4H SYSRST/USBRST= 0XX0-XXXX

7	6	5	4	3	2	1	0
RXCLR	--	--	RXFFRC	--	--	--	--
W	**W**	**W**	**W**	**W**	**W**	**W**	**W**

Bit 7: RXCLR-- Receive FIFO Clear.
Set this bit to flush the entire receive FIFO. All FIFO statuses are reverted to their reset states. Hardware clears this bit when the flush operation is completed.

Bit 6~5: Reserved. Software must write “0” on these bits when RXCON is written.

Bit 4: RXFFRC-- Receive FIFO Read Complete.
Set this bit to release the receive FIFO when data set read is complete. Hardware clears this bit after the FIFO release operation has been finished.

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

RXCNT: Receive FIFO Byte Count Register *(endpoint-indexed)*

SFR Address = 0xE6H SYSRST/USBRST= 0000-0000

7	6	5	4	3	2	1	0
RXBC7	RXBC6	RXBC5	RXBC4	RXBC3	RXBC2	RXBC1	RXBC0
R	**R**	**R**	**R**	**R**	**R**	**R**	**R**

Bit 7~0: RXBC[7:0]-- Receive Byte Count.
For endpoint 0~2, This register is used to store the byte count for the data packet received in the receive FIFO specified by EPINDEX. If EPINDEX =3’b111, this register is designed to be a low byte address for read/write 32B SRAM function.

TXSTAT: Endpoint Transmit Status Register *(endpoint-indexed)*

SFR Address = 0xF2H SYSRST/USBRST= 0XXX-0XXX

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

Bit 7: TXSEQ-- Transmit Endpoint Sequence Bit (read, conditional write).
The bit will be transmitted in the next PID and toggled on a valid ACK handshake of an IN transaction. This bit can be written by firmware if the TXOVW bit is set when written along with the new TXSEQ value.

Bit 6~4: Reserved. Software must write “0” on these bits when TXSTAT is written.

Bit 3: TXSOVW-- Transmit Data Sequence Overwrite Bit.
Write '1' to this bit to allow the value of the TXSEQ bit to be overwritten. Writing a '0' to this bit has no effect on TXSEQ. This bit always returns '0' when read.

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

TXDAT: Transmit FIFO Data Register *(endpoint-indexed)*

SFR Address = 0xF3H SYSRST/USBRST= XXXX-XXXX

7	6	5	4	3	2	1	0
TXD7	TXD6	TXD5	TXD4	TXD3	TXD2	TXD1	TXD0
W	**W**	**W**	**W**	**W**	**W**	**W**	**W**

Bit 7~0: TXD[7:0]-- Transmit FIFO Data.
Data to be transmitted in the FIFO specified by EPINDEX is written to this register.

TXCON: Transmit FIFO Control Register *(endpoint-indexed)*

SFR Address = 0xF4H SYSRST/USBRST= 0XXX-0XXX

7	6	5	4	3	2	1	0
TXCLR	--	--	TXFFRC	--	--	--	--
W	**W**	**W**	**W**	**W**	**W**	**W**	**W**

Bit 7: TXCLR-- Transmit FIFO Clear.
Set this bit to flush the entire transmit FIFO. All FIFO statuses are reverted to their reset states. Hardware clears this bit when the flush operation is completed.

Bit 6~5: Reserved. Software must write “0” on these bits when TXCON is written.

Bit 4: TXFFRC-- Transmit FIFO Write Complete.
Set this bit to release the transmit FIFO when data set write is complete. Hardware clears this bit after the FIFO release operation has been finished. Firmware should write this bit only after firmware finished writing TXCNT register.

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

TXCNT: Transmit FIFO Byte Count Register *(endpoint-indexed)*

SFR Address = 0xF6H SYSRST/USBRST= XXXX-XXXX

7	6	5	4	3	2	1	0
TXBC7	TXBC6	TXBC5	TXBC4	TXBC3	TXBC2	TXBC1	TXBC0
W	**W**	**W**	**W**	**W**	**W**	**W**	**W**

Bit 7~0: TXBC[7:0]-- Transmit Byte Count.
For endpoint 0~2, this register is used to stored the byte count for the data packet in the transmit FIFO specified by EPINDEX. If EPINDEX = 3’b111, this register is designed to be a high byte address for read/write 32B SRAM function.

SIOCTL: Serial I/O Control Register High

SFR Address = 0x2FH SYSRST = XXXX-XXXX

7	6	5	4	3	2	1	0
DPI	DMI	--	--	--	--	--	--
R	**R**	**R**	**R**	**R**	**R**	**R**	**R**

Bit 7: DPI-- USB DP port state, read only
Read the port status on USB DP.

Bit 6: DMI-- USB DM port state, read only
Read the port status on USB DM.

Bit 5~0: Reserved. Read only.

20. Protected-Write SFR Access

MG74PG1A08 builds a special SFR with Protected-Write function to store the control registers for MCU operation. These SFRs can be read by the normal SFR operation. In Write operation, SCMD must set to "**8C**" and then followed the target SFR write.

Following descriptions are the SFR function definition with Protected-Write Key for "8C" on SCMD.

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	
PCON2	Power Control 2	BAH	AWBOD1	EBOD1	--	--	BO1RE	BO0RE	--	RMLS	01000000
CKCON2	Clock Control 2	BBH	--	--	--	IHRCOE	MCKS1	MCKS0	OSCS1	OSCS0	xxx10000
DCON0	Device Control 0	BCH	WCKS	USBR	ENUSB	ENCKM	CKMIS1	CKMIS0	RSTIO	SWRST	00000110
SPCON0	SFR Page Control 0	BDH	--	--	--	WRCTL	--	CKCTL0	PWCTL1	PWCTL0	0xx00000

SPCON0: SFR Page Control 0

SFR Attribute = Normal Read and Protected Write

SFR Address = 0xBD POR+LVR = xxx0-0000

7	6	5	4	3	2	1	0
--	--	--	WRCTL	--	CKCTL0	PWCTL1	PWCTL0
W	**W**	**W**	**R/W**	**W**	**R/W**	**R/W**	**R/W**

Bit 7~5: Reserved. Software must write "0" on these bits when SPCON0 is written.

Bit 4: WRCTL. WDTCR SFR access Control.
If WRCTL is set, it will enable the protected-write function on WDTCR SFR modified with the protected key = 0x8C. Read WDTCR still keeps the normal SFR read function.

Bit 3: Reserved. Software must write "0" on these bits when SPCON0 is written.

Bit 2: CKCTL0. CKCON0 SFR access Control.
If CKCTL0 is set, it will enable the protected-write function on CKCON0 SFR modified with the protected key = 0x8C. Read CKCON0 still keeps the normal SFR read function.

Bit 1: PWCTL1. PCON1 SFR access Control.
If PWCTL1 is set, it will enable the write-protected function on PCON1 SFR modified with the protected key = 0x8C. Read PCON1 still keeps the normal SFR read function.

Bit 0: PWCTL0. PCON0 SFR access Control.
If PWCTL0 is set, it will enable the write-protected function on PCON0 SFR modified with the protected key = 0x8C. PCON0 still keeps the general SFR read function.

21. 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 U3" or the "Megawin 8051 ICE Adapter" (The ICE adapter also supports ICP programming function.). After whole-chip erased, all the hardware options are left in "disabled" state. The **MG74PG1A08** has the following Hardware Options:

LOCK:
☑: Enabled. Code dumped on a universal Writer or Programmer is locked to 0x00 for security.
☐: Disabled. Not locked.

OSCS1O: MCU OSC type selection bit from power-on.
☑: Enabled. MCU clock source select IHRCO.
☐: Disabled. MCU clock source select ILRCO.

RMLSO:
☑: Enabled. POR level on 1.95V
☐: Disabled. POR level on 2.3V

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

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.

WRENO: WDT Reset Enable Option
☑: Enabled. Set WDTCR.WREN to enable a system reset function by WDTF.
☐: Disabled. Clear WDTCR.WREN to disable the system reset function by WDTF.

HWWIDL, HWPS2, HWPS1, HWPS0:
When HWENW is enabled, the content on these four fused bits will be loaded to WDTCR SFR after power-on.

P17EN:
☑: Enabled. RSTIO (DCON0.1) will be cleared and P1.7 function behaves on nRST pin.
☐: Disabled. RSTIO (DCON0.1) will be set and reserve nRST pin function.

FPUT: Fast Power-Up Timer enable.
☑: Enabled. Enable fast power-up timer less than 16ms.
☐: Disabled. Keep default power-up timer. It is about more than 16ms.

22. Application Notes

22.1. Power Supply Circuit

To have the **MG74PG1A08** work with power supply varying from 2.3V to 5.5V adding some external decoupling and bypass capacitors is necessary, as shown in Figure 22–1. There caps have to be close to the chip power and ground. **MG74PG1A08** work with battery power supply varying from 2.0V to 3.6V, as shown in Figure 22–2.

Figure 22–1. Power Supplied Circuit

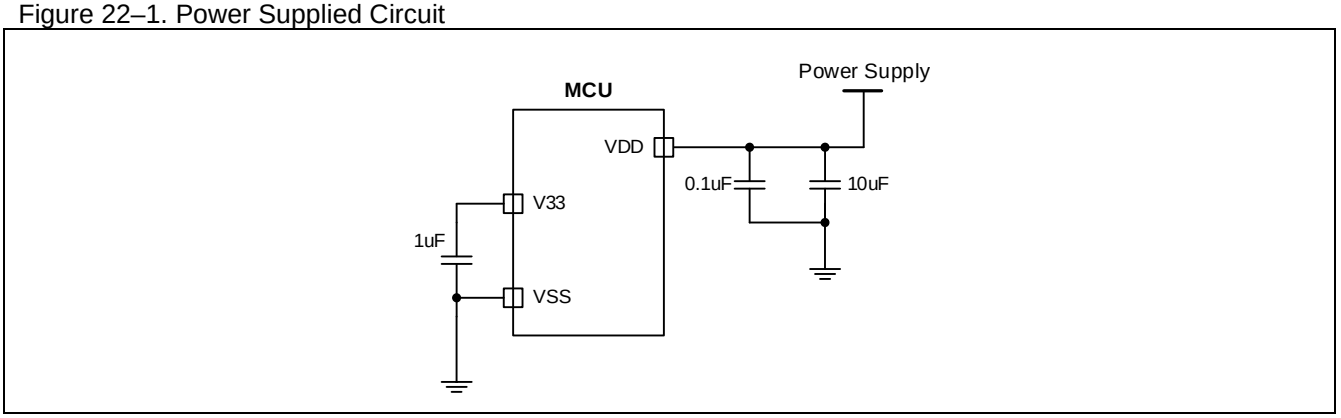
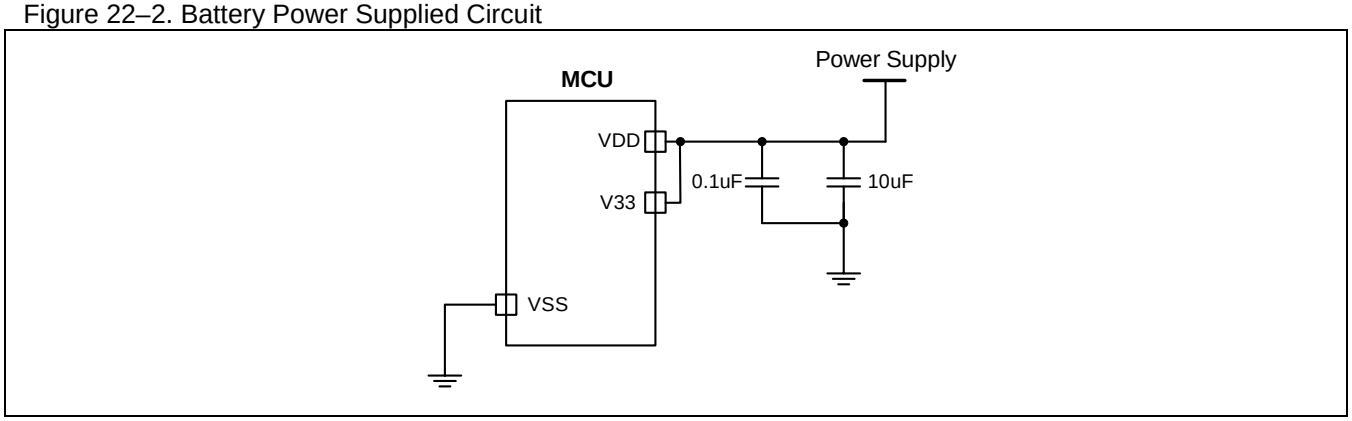


Figure 22–2. Battery Power Supplied Circuit



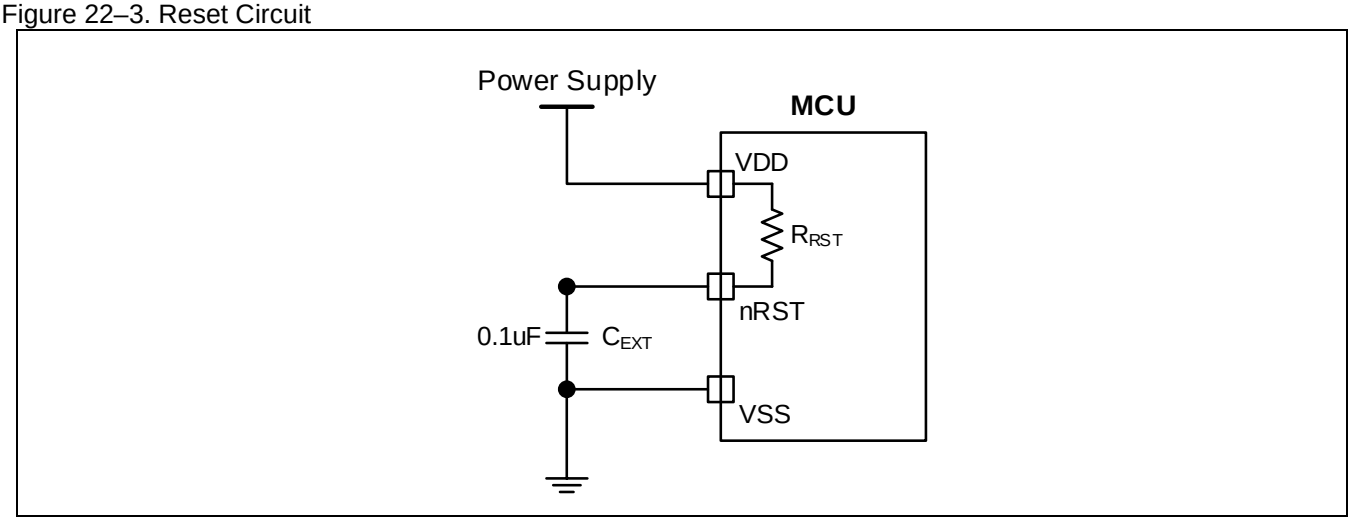
22.2. Reset Circuit

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 (RSTIO must set to 1). Figure 22–3 shows the external reset circuit, which consists of a capacitor C_{EXT} connected to VSS (ground).

In general, the nRST pin has an internal pull-high resistor (R_{RST}). This internal MOS resistor to VDD permits a power-up reset using only an external capacitor C_{EXT} to VSS.

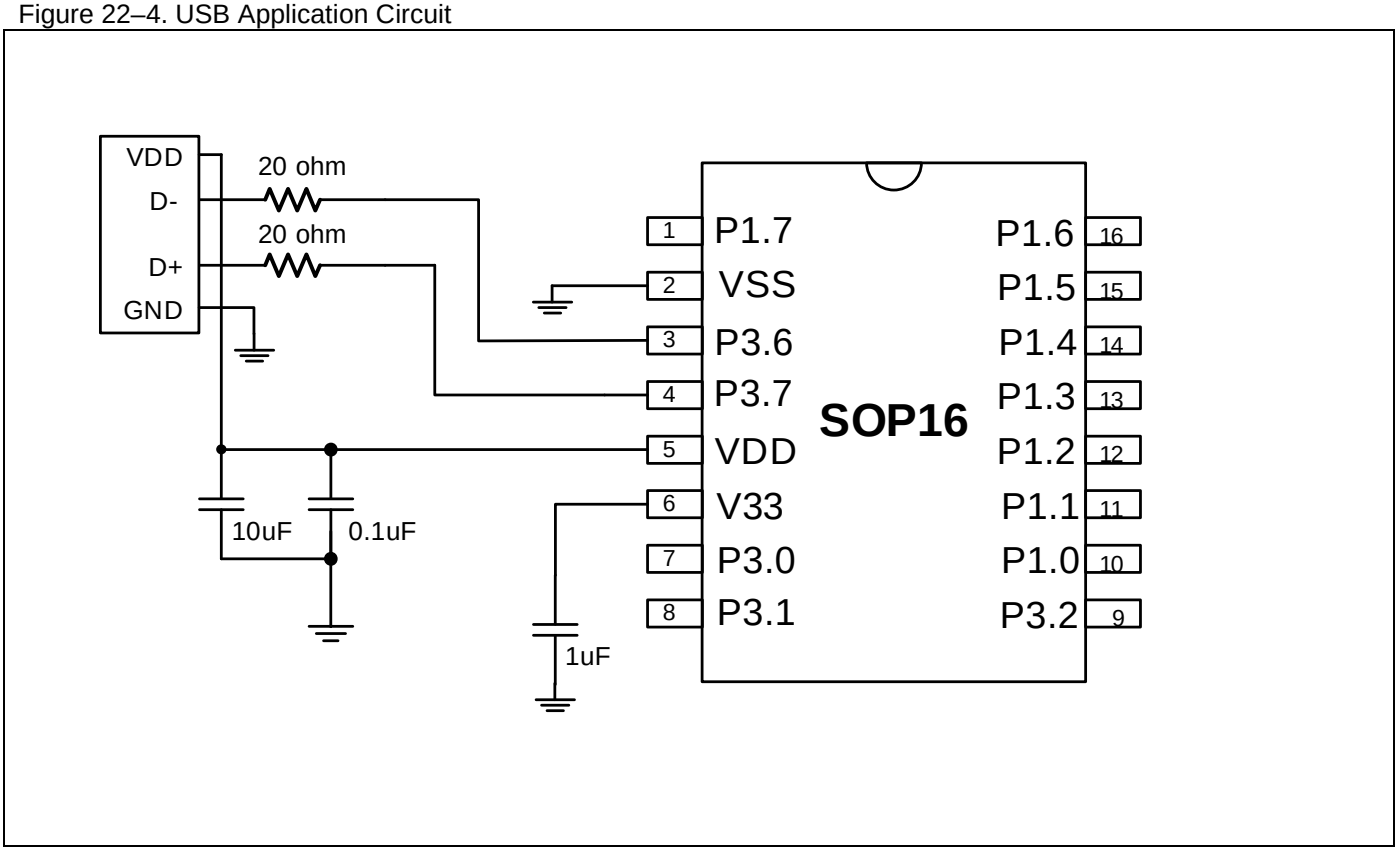
See Section "23.2 DC Characteristics" for R_{RST} value.

Figure 22–3. Reset Circuit



22.3. With USB application Circuit

Figure 22–4. USB Application Circuit



22.4. ICP Interface Circuit

MG74PG1A08 devices include an on-chip Megawin proprietary programming interface to allow In-Chip-Programming (ICP) with the production part installed in the end application. The ICP interface uses a clock signal (ICP_SCL) and a bi-directional data signal (ICP_SDA) to perform the device programming from a host instruction.

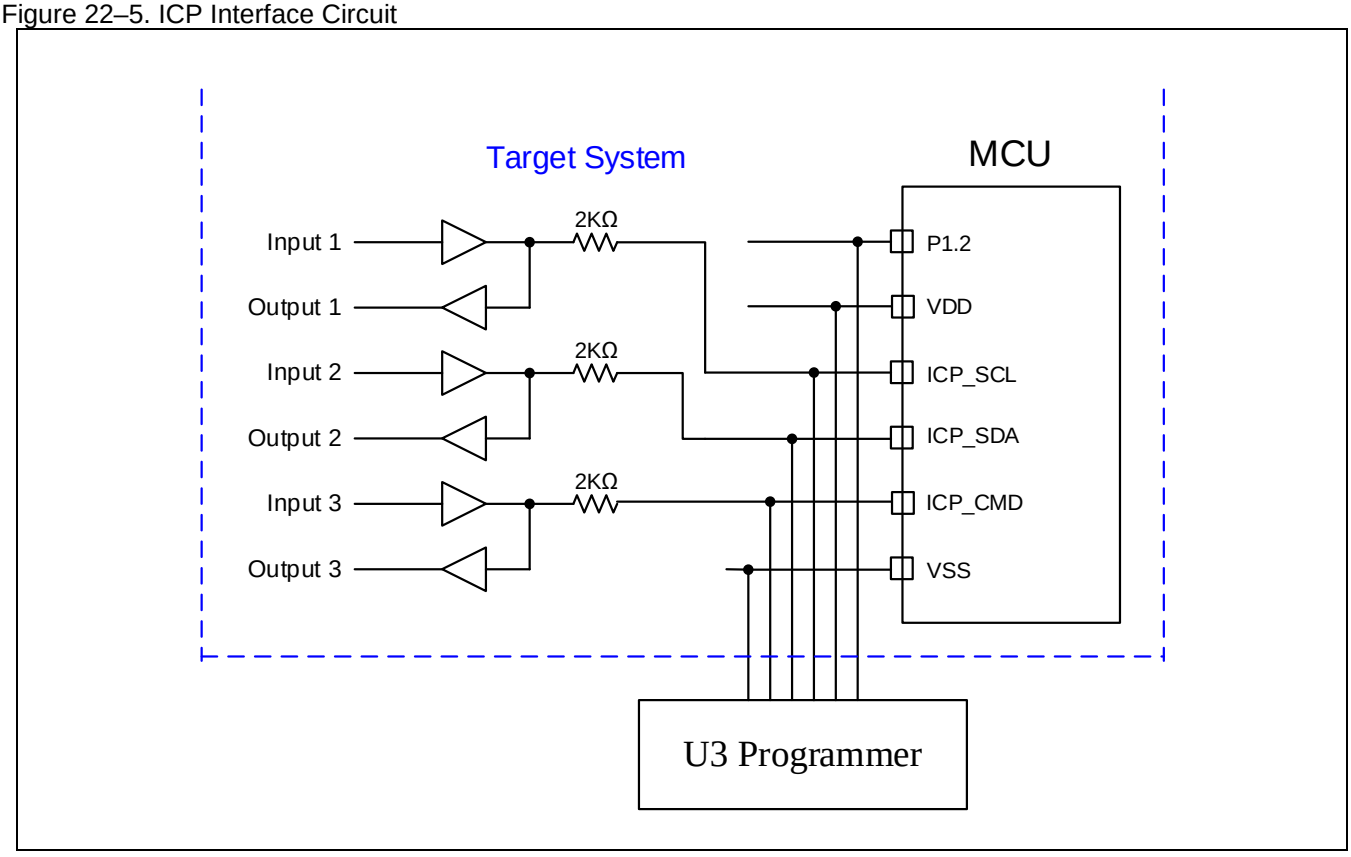
The ICP interface allows the ICP_SCL/ICP_SDA pins to be shared with user functions so that In-Chip Programming function could be performed. This is practicable because ICP communication is performed when the device is in the start-up state (in power-on period less than 8ms), where the on-chip peripherals and user software are stalled. In this state, the ICP interface can safely 'borrow' the ICP_SCL (P1.1) and ICP_SDA (P1.0) pins. The typical configuration is shown in Figure 22–5.

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.

Attention:
P1.2 (VPP) pin will have 7.5V high voltage form U3 programmer within OTP programming period. It may damage other components which connect to pin P1.2 (VPP).

The pins of ICP interface not allow any additional capacitance.

Figure 22–5. ICP Interface Circuit



23. Electrical Characteristics

23.1. Absolute Maximum Rating

Parameter	Rating	Unit
Ambient temperature under bias	-40 ~ +85	°C
Storage temperature	-65 ~ + 150	°C
Voltage on any Port I/O Pin or nRST with respect to VSS	-0.5 ~ VDD + 0.5	V
Voltage on VDD with respect to VSS	-0.5 ~ +6.0	V
Maximum total current through VDD and VSS	200	mA
Maximum output current sunk by any Port pin	40	mA

*Note: stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the devices at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

23.2. 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 P1.7, P3.6, P3.7	0.6			VDD
V_{IH2}	Input High voltage (P1.7,P3.6, P3.7)		0.75			VDD
V_{IL1}	Input Low voltage (All I/O Ports)	Except P1.7, P3.6, P3.7			0.15	VDD
V_{IL2}	Input Low voltage (P1.7,P3.6, P3.7)				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 or other Input port with on-chip pull-up resistor)	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 or other Input port with on-chip pull-up resistor)	V_{PIN} =V_{H2L}		330	500	uA
I_{OH}	Output High current (All push-pull output ports)	V_{OH} =2.4V		30		mA
I_{OL}	Output Low current (All I/O Ports)	V_{OL} =0.4V		20		mA
R_{RST}	Internal reset pull-up resistance	V_{P17}=0V		30		Kohm
R_{ph1}	Weak pull-high resistor (All I/O Ports)	V_{PIN}=2V		10		Kohm
R_{ph2}	Very weak pull-high resistor (All I/O Ports)	V_{PIN}=0V		260		Kohm
Power Consumption						
I_{OP1}	Normal mode operating current	SYSCLK = 6MHz @ IHRCO		2.7		mA
I_{OP2}	Normal mode operating current	SYSCLK = 12MHz @ IHRCO		4		mA
I_{OP3}	Normal mode operating current	SYSCLK = 24MHz @ IHRCO with PLL		4.5		mA
I_{OP4}	Normal mode operating current	SYSCLK = 12MHz @ IHRCO with USB+PLL		5.3		mA
I_{OPS1}	Slow mode operating current	SYSCLK = 12MHz/8 @ IHRCO		1.9		mA
I_{IDLE1}	Idle mode operating current	SYSCLK = 12MHz @ IHRCO		2		mA
I_{IDLE2}	Idle mode operating current	SYSCLK = 64KHz @ ILRCO		0.6		mA
I_{SUB1}	Sub-clock mode operating current	SYSCLK = 64KHz @ ILRCO		0.6		mA
I_{SUB2}	Sub-clock mode operating current	SYSCLK = 64KHz/8 @ ILRCO		0.6		mA
I_{WAT}	Watch mode operating current	WDT = 64KHz @ ILRCO in PD mode		4		uA
I_{MON1}	Monitor Mode operating current	BOD1 enabled in PD mode		45		uA
I_{PD1}	Power down mode current			2	3	uA
POR/BOD0/BOD1 Characteristics						
V_{PORL}	POR detection level for 1.95V	T_A = -40°C to +85°C	1.85(1)	2.0	2.15(1)	V
V_{PORH}	POR detection level for 2.3V	T_A = -40°C to +85°C	2.1(1)	2.3	2.5(1)	V

Symbol	Parameter	Test Condition	Limits			Unit
			Min.	Typ.	Max.	
V_{BOD0L}	BOD0 detection level for 2.1V	T_A = -40°C to +85°C	1.95(1)	2.1	2.25(1)	V
V_{BOD0H}	BOD0 detection level for 2.6V	T_A = -40°C to +85°C	2.5(1)	2.6	2.7(1)	V
V_{BOD1}	BOD1 detection level for 3.6V	T_A = -40°C to +85°C	3.3(1)	3.6	3.9(1)	V
I_{BOD1}	BOD1 Power Consumption	T_A = +25°C, VDD=5.0V		40		uA
Operating Condition						
V_{PSR}	Power-on Slop Rate	T_A = 25°C	0.05			V/ms
V_{POR1}	Power-on Reset Valid Voltage	T_A = -40°C to +85°C			0.1	V
V_{OP1}	IHRCO Operating Speed 0–6MHz	T_A = 25°C	2.0		5.5	V
V_{OP2}	IHRCO Operating Speed 0-12MHz	T_A = 25°C	2.4		5.5	V
V_{OP3}	IHRCO Operating Speed 0-24MHz	T_A = 25°C	2.7		5.5	V
V_{OP4}	CPU Operating Speed 0-3MHz	T_A = 25°C	2.0		5.5	V
V_{OP5}	CPU Operating Speed 0-6MHz	T_A = 25°C	2.4		5.5	V
V_{OP6}	CPU Operating Speed 0-12MHz	T_A = 25°C	2.7		5.5	V

(1) Data based on characterization results, not tested in production.

23.3. USB Transceiver Electrical Characteristics

VDD = 4.0V ~ 5.5V, VSS = 0V, T_A = 25°C, unless otherwise specified

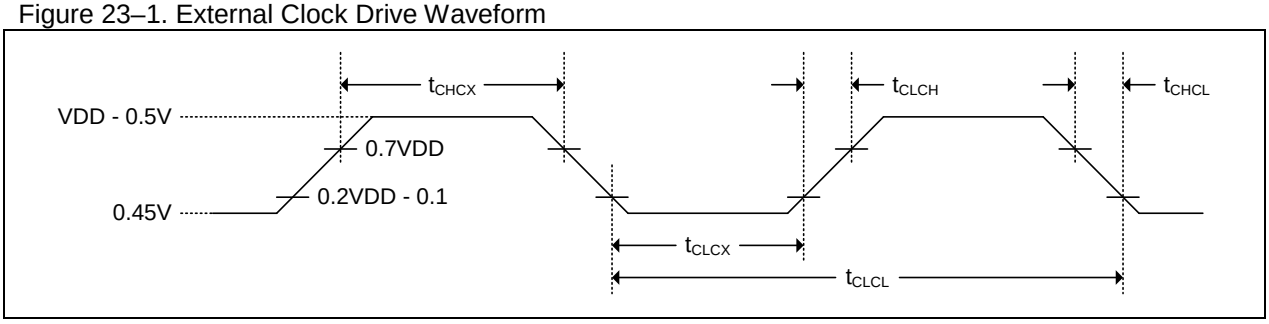
Symbol	Parameter	Test Condition	Limits			Unit
			min	typ	max	
V_{V33}	3.3V regulator output voltage	T_A = 25°C	3.0	3.3	3.6	V
I_{V33}	Regulator Output drive current	T_A = 25°C			35	mA
R_{PU}	Pull-Up Resistance	On DP	0.95	1.1	1.3	Kohm
R_{PD}	Pull-Down Resistance	On DP & DM		500		Kohm
R_{PU2}	Pull-Up Resistance for PS/2 mode	On DP & DM		7		Kohm
Transmitter						
V_{OH}	Output High Voltage		2.8			V
V_{OL}	Output Low Voltage				0.8	V
V_{CRS}	Output Cross Over point		1.3		2.0	V
Z_{DRVH}	Output Impedance on Driving High		28		44	Ohm
Z_{DRVL}	Output Impedance on Driving Low		28		44	Ohm
T_R	Output Rise Time		4		20	ns
T_F	Output Fall Time		4		20	ns
Receiver						
V_{DI}	Differential Input Sensitivity	| DP – DM |	0.2			V
V_{CM}	Differential Input Common Mode Range		0.8		2.5	V
I_L	Input Leakage current	Pull-up Disabled		<1.0		uA

23.4. External Clock Characteristics

VDD = 2.7V ~ 5.5V, VSS = 0V, T_A = -40°C to +85°C, unless otherwise specified

Symbol	Parameter	Oscillator ECKI Mode		Unit
		Min.	Max.	
$1/t_{CLCL}$	Frequency (VDD = 2.0V ~ 5.5V)	0	12	MHz
t_{CLCL}	Clock Period	80		ns
t_{CHCX}	High Time	0.4T	0.6T	t_{CLCL}
t_{CLCX}	Low Time	0.4T	0.6T	t_{CLCL}
t_{CLCH}	Rise Time		5	ns
t_{CHCL}	Fall Time		5	ns

Figure 23–1. External Clock Drive Waveform



23.5. IHRCO Characteristics

Parameter	Test Condition	Limits			Unit
		Min.	Typ.	Max.	
Supply Voltage		2.0		5.5	V
IHRCO Frequency	TA = +25°C		12		MHz
IHRCO Frequency Deviation (factory calibrated)	TA = +25°C	-1.0		+1.0	%
	TA = -40°C to +85°C	-4(1)		+4(1)	%
IHRCO Start-up Time	TA = -40°C to +85°C			32(1)	us
IHRCO Power Consumption	TA = +25°C, VDD=5.0V		500(1)		uA

(1) Data based on characterization results, not tested in production.

23.6. ILRCO Characteristics

Parameter	Test Condition	Limits			Unit
		Min.	Typ.	Max.	
Supply Voltage		2.0		5.5	V
ILRCO Frequency	TA = +25°C		64		KHz
ILRCO Frequency Deviation	TA = +25°C	-5(1)		+5(1)	%
	TA = -40°C to +85°C	-30(1)		+30(1)	%

(1) Data based on characterization results, not tested in production.

23.7. CKM Characteristics

Parameter	Test Condition	Limits			Unit
		Min.	Typ.	Max.	
Supply Voltage	TA = -40°C to +85°C	2.4		5.5	V
Clock Input Range	TA = -40°C to +85°C	5(1)	6	6.5(1)	MHz
CKM Start-up Time	TA = -40°C to +85°C		10	20	us
CKM Power Consumption	TA = +25°C, VDD=5.0V		650		uA

(1) Data guaranteed by design, not tested in production.
(2) Data based on characterization results, not tested in production.

23.8. OTP Characteristics

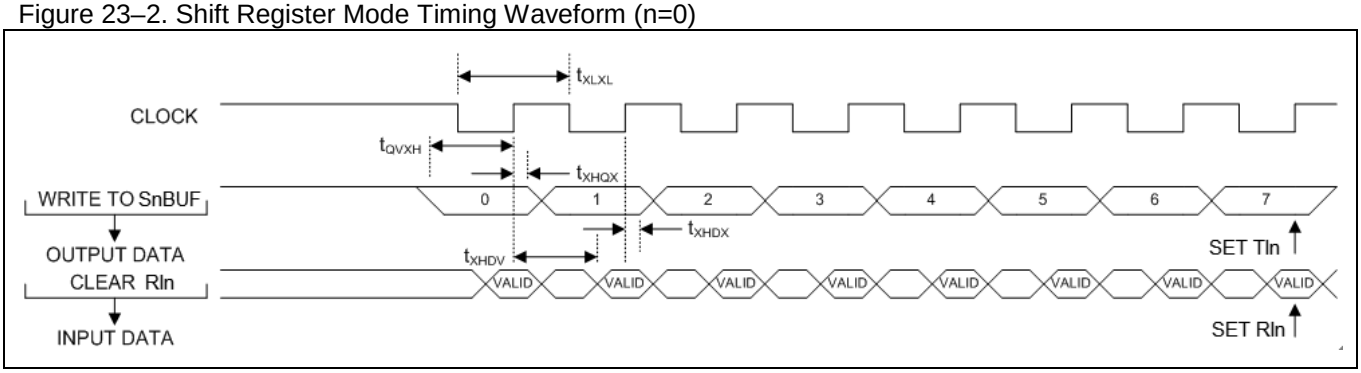
Parameter	Test Condition	Limits			Unit
		Min.	Typ.	Max.	
Supply Voltage	TA = -40°C to +85°C	3		5	V
Program Voltage (VPP)	TA = -40°C to +85°C	7.25	7.5	7.75	V
Access Time	TA = -40°C to +85°C			200	ns
Byte program time	TA = -40°C to +85°C		100		us
Data Retention	TA = +85°C	10			year

23.9. Serial Port 0 Timing Characteristics

VDD = 5.0V±10%, VSS = 0V, T_A = -40℃ to +85℃, unless otherwise specified

Symbol	Parameter	URM0X3 = 0		URM0X3 = 1		Unit
		Min.	Max.	Min.	Max.	
t_{XLXL}	Serial Port 0 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		4T-20	ns

Figure 23–2. Shift Register Mode Timing Waveform (n=0)



24. Instruction Set

Table 24–1. Instruction Set

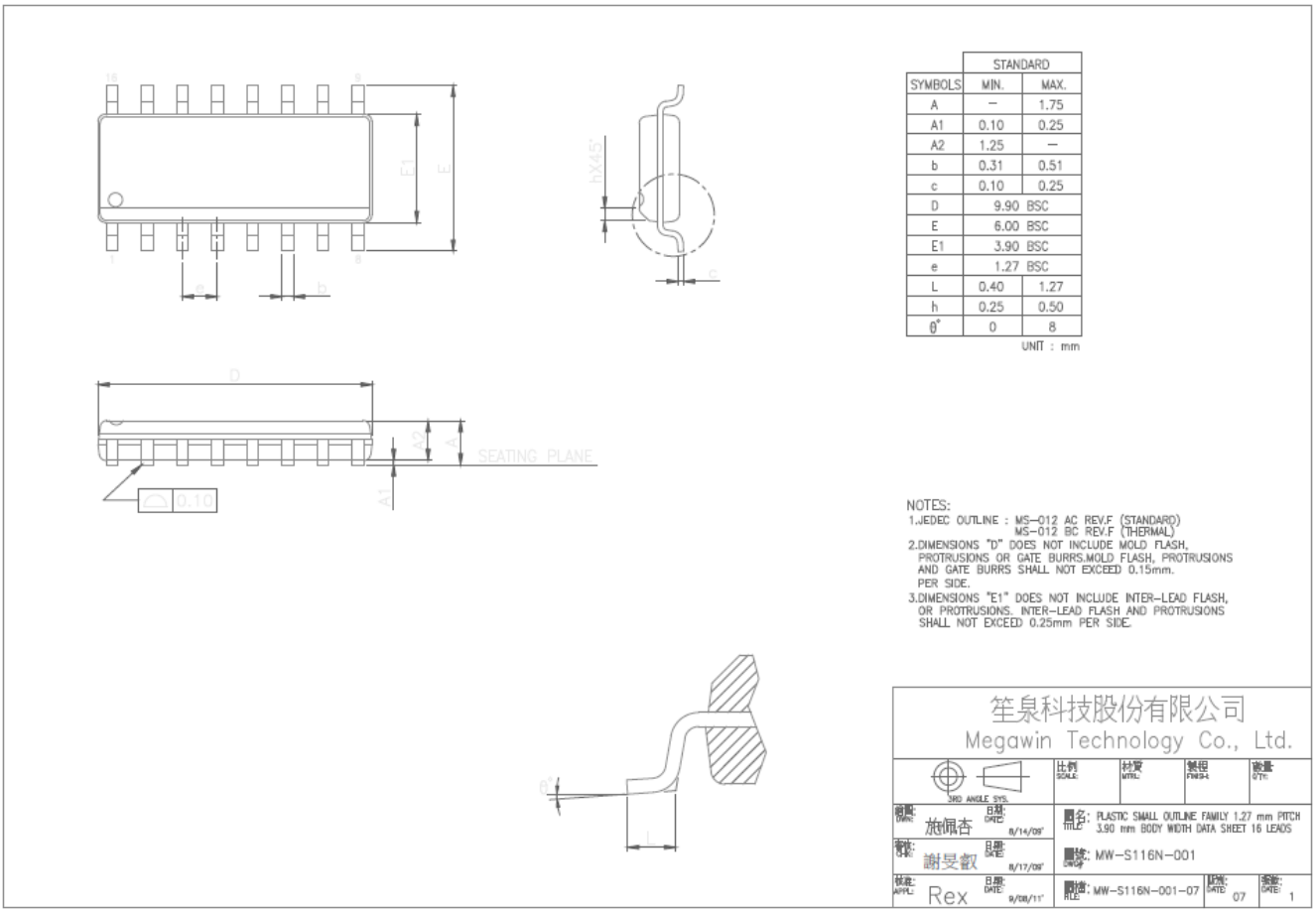
MNEMONIC	DESCRIPTION	BYTE	EXECUTION Cycles
DATA TRASFER			
MOV A,Rn	Move register to Acc	1	1
MOV A,direct	Move direct byte o Acc	2	2
MOV A,@Ri	Move indirect RAM to Acc	1	2
MOV A,#data	Move immediate data to Acc	2	2
MOV Rn,A	Move Acc to register	1	2
MOV Rn,direct	Move direct byte to register	2	4
MOV Rn,#data	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	Not Support
MOVX A,@DPTR	Move on-chip auxiliary RAM(16-bit address) to Acc	1	Not Support
MOVX @Ri,A	Move Acc to on-chip auxiliary RAM(8-bit address)	1	Not Support
MOVX @DPTR,A	Move Acc to on-chip auxiliary RAM(16-bit address)	1	Not Support
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
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
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

25. Package Dimension

25.1. SOP16



SYMBOLS	STANDARD MIN.	STANDARD 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

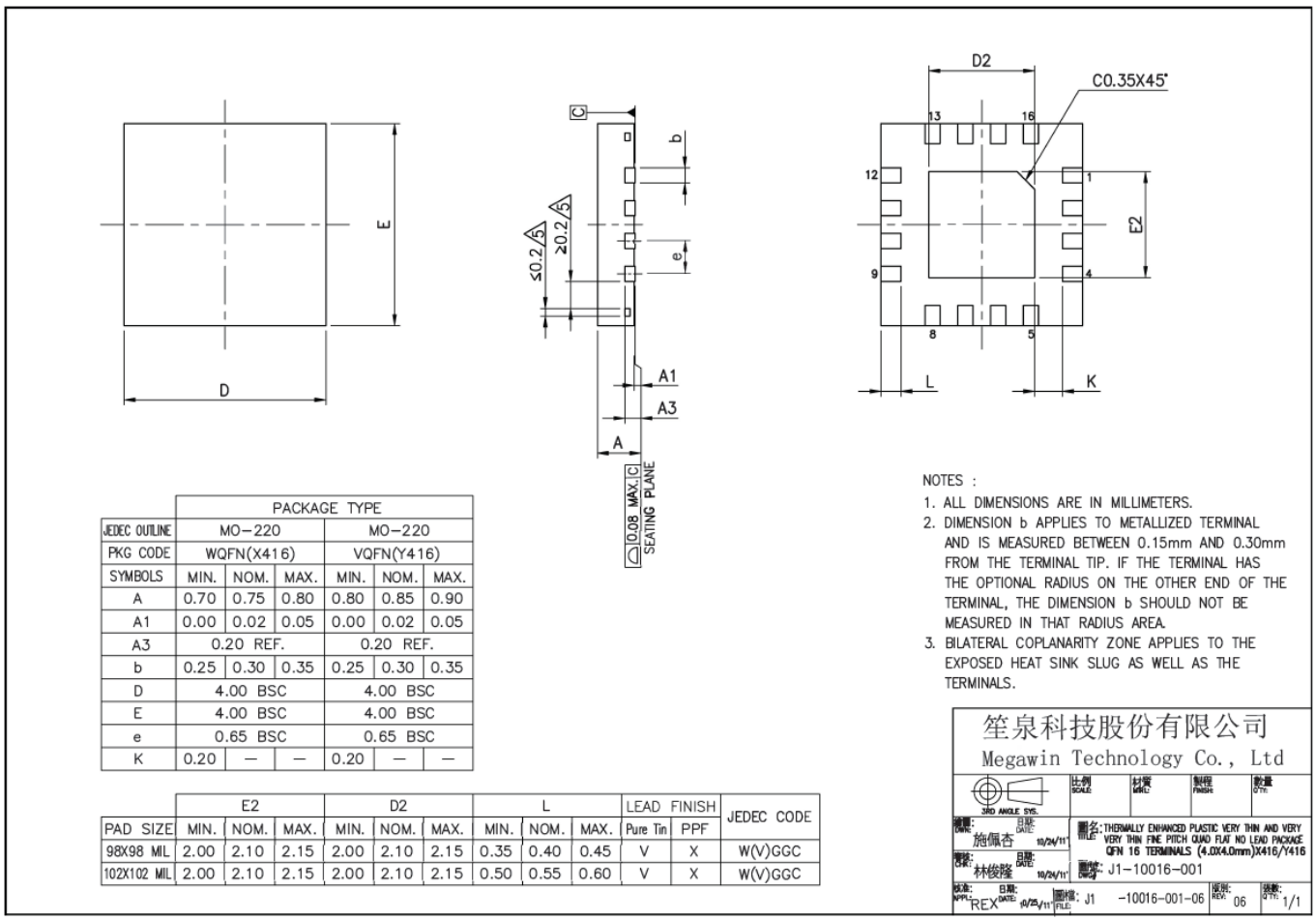
UNIT : mm

NOTES:
1.JEDEC OUTLINE : MS-012 AC REV.F (STANDARD)
MS-012 BC REV.F (THERMAL)
2.DIMENSIONS "D" DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS.MOLD FLASH, PROTRUSIONS AND GATE BURRS SHALL NOT EXCEED 0.15mm. PER SIDE.
3.DIMENSIONS "E1" DOES NOT INCLUDE INTER-LEAD FLASH, OR PROTRUSIONS. INTER-LEAD FLASH AND PROTRUSIONS SHALL NOT EXCEED 0.25mm PER SIDE.

笙泉科技股份有限公司
Megawin Technology Co., Ltd.

3RD ANGLE SYS.	比例 SCALE:	材質 MTRL:	數量 QTY:
製圖: 施佩杏	日期: 8/14/09'	圖名: PLASTIC SMALL OUTLINE FAMILY 1.27 mm PITCH 3.90 mm BODY WIDTH DATA SHEET 16 LEADS	
審核: 謝旻叡	日期: 8/17/09'	圖號: MW-S116N-001	
核准: Rex	日期: 9/08/11'	檔案: MW-S116N-001-07	版次: 07

25.2. QFN16



	PACKAGE TYPE					
JEDEC OUTLINE	MO-220			MO-220		
PKG CODE	WQFN(X416)			VQFN(Y416)		
SYMBOLS	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	0.70	0.75	0.80	0.80	0.85	0.90
A1	0.00	0.02	0.05	0.00	0.02	0.05
A3	0.20 REF.			0.20 REF.		
b	0.25	0.30	0.35	0.25	0.30	0.35
D	4.00 BSC			4.00 BSC		
E	4.00 BSC			4.00 BSC		
e	0.65 BSC			0.65 BSC		
K	0.20	—	—	0.20	—	—

	E2			D2			L			LEAD FINISH		JEDEC CODE
PAD SIZE	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.	Pure Tin	PPF	
98X98 MIL	2.00	2.10	2.15	2.00	2.10	2.15	0.35	0.40	0.45	V	X	W(V)GGC
102X102 MIL	2.00	2.10	2.15	2.00	2.10	2.15	0.50	0.55	0.60	V	X	W(V)GGC

26. Revision History

Table 26–1. Revision History

Rev	page	Descriptions	Date
V0.22		1. Preliminary version release.	2016/02/01
V0.23		1. Added QFN16 package type information.	2016/06/20
V0.24	112	1. Added battery power supply application.	2016/08/24
V0.25	3,11,125	1. Remove SOP10 package type.	2016/11/16

27. 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).