MEGAWIN
MAKE YOU WIN

8051-Based MCU

MG82FE(L)308/316
Data Sheet

Version: A1.0

This document contains information on a new product under development by Megawin. Megawin reserves the right to change or discontinue this product without notice.

2013/11 version A1.0

Features

- 1-T 80C51 Central Processing Unit
- MG82FE(L)308/316 with 8K/13.5K Bytes AP flash ROM
  - ISP memory zone as 1.5KB (default)
  - IAP size as 1KB(MG82FE(L)316), 6.5KB(MG82FE(L)308) in default.
  - User could control IAPLB register to redefine IAP size. Flexible IAP size up to 15.5KB.
  - Code protection for flash memory access
- On-chip 256 bytes data RAM and on-chip 256 bytes expanded RAM.
- Dual data pointer.
- Three 16-bit timer/counters: Timer 0, Timer 1 and Timer 2.
  - T0CKO on P34, T1CKO on P35 and T2CKO on P10
  - X12 mode enabled for T0/T1/T2.
- PWM-Timer for PWM generator or normal 16-bit timer.
  - Maximum 16 channel PWM output on P2 and P5.
- Enhanced UART
  - Provides frame-error detection.
  - Hardware address-recognition.
  - Function swapped on P16/P17.
- Six external interrupt input: nINT0/nINT1/nINT2/nINT3/nINT4/nINT5.
  - nINT0/nINT1 trigger type: Low Level or Falling Edge.
  - nINT2/nINT3/nINT4/nINT5: Low Level, Falling Edge, High Level or Rising Edge.
- 13 sources, four-level-priority interrupt capability.
- Built-in analog comparator with on-chip VDD reference input.
  - 4 channel I/O selection on comparator Plus input.
  - Programmable comparator Minus input from I/O or 15 levels on-chip VDD reference.
- 15 bits Watch-Dog-Timer.
  - 8-bit pre-scalar
  - One-time enabled by CPU or power-on
- Maximum 57 GPIOs in LQFP-64 pin package.
  - P0, P1, P2, P3, P4, P5 can be configured to quasi-bidirectional, push-pull output, open-drain output and input only.
  - P6 and P7 serve quasi-bidirectional mode only.
  - P4.0 and P4.1 are configured to quasi-bidirectional in default.
- Power control: idle mode and power-down mode.
  - All interrupts and 16 GPIOs can wake up IDLE mode.
  - 6 external interrupt and 16 GPIOs can wake up Power-Down mode.
- Brown-Out Detector for VDD 4.0V(E) or 2.4V(L)
  - Option to reset CPU.
  - Option to interrupt CPU.
- Operating voltage:
  - MG82FE308/316: 4.5V~5.5V.
  - MG82FL308/316: 2.4V~3.6V, minimum 2.7V requirement in flash write operation(ISP/IAP/……)
- Operation frequency : 24MHz(max)
  - External crystal mode
  - Internal RC-oscillator (12MHz) , output on XTAL2/P6.0

    +/- 1% frequency drift @ 25℃,

    +/- 2% frequency drift @ -20 ~ 50℃,

    +/- 4% frequency drift @ -40 ~ 85℃,.

- Operating Temperature:
  - Industrial (-40℃ to +85℃)*
- Package Types:
  - LQFP64 (7mm X 7mm): MG82FE(L)316AD64
  - LQFP48: MG82FE(L)308AD48

*: Tested by sampling.

Content

1. General Description

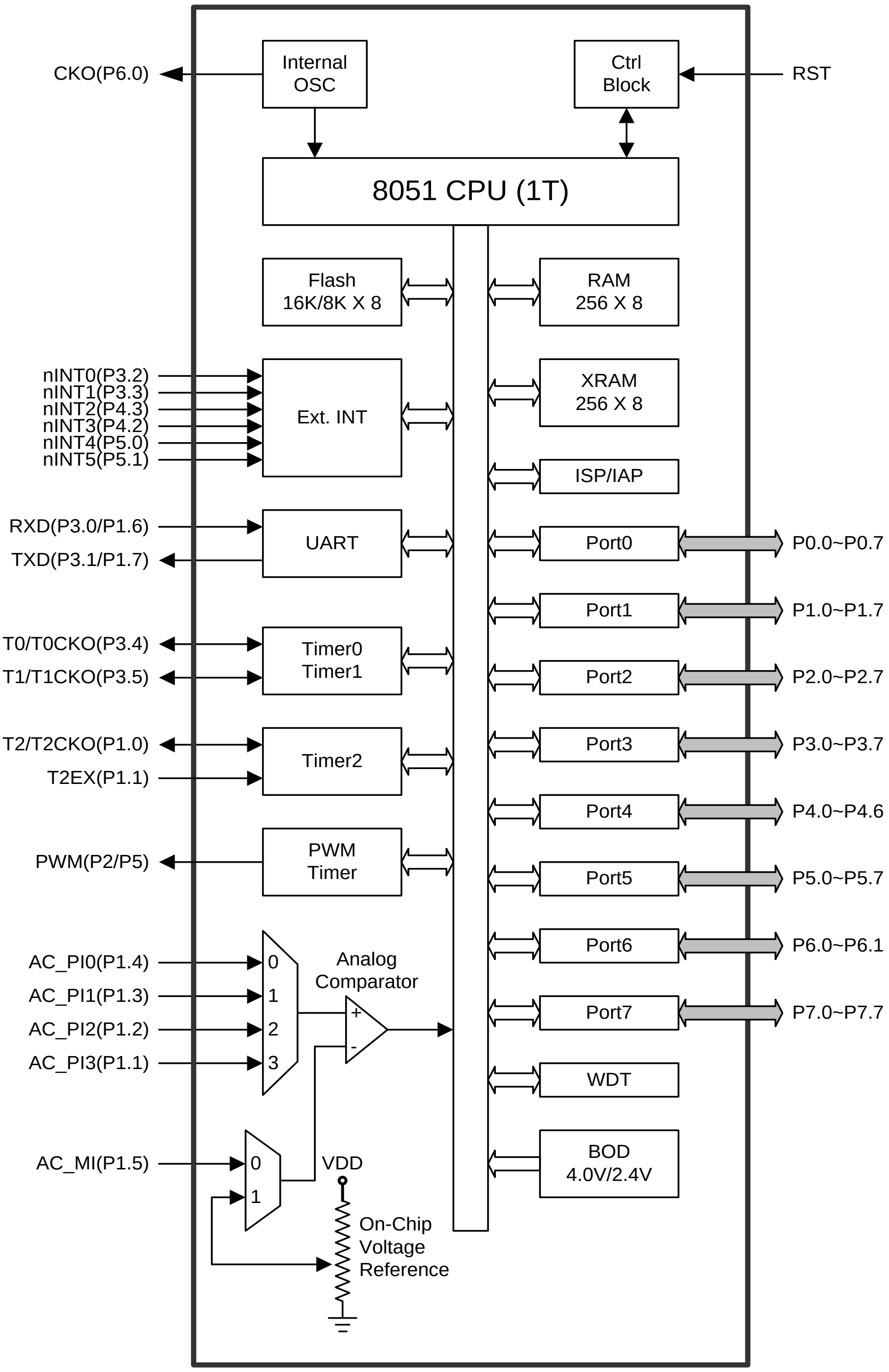
The MG82FE(L)308/316 is a single-chip microcontroller based on a high performance 1-T architecture 80C51 CPU that executes instructions in 1~7 clock cycles (about 6~7 times the rate of a standard 8051 device), and has an 8051 compatible instruction set. Therefore at the same performance as the standard 8051, the MG82FE(L) 308/316 can operate at a much lower speed and thereby greatly reduce the power consumption.

The MG82FE(L)308/316 has 16K bytes of embedded Flash memory for code and data. The Flash memory can be programmed either in writer mode or in ISP (In-System Programming) mode. And, it also provides the In-Application Programming (IAP) capability. ISP allows the user to download new code without removing the microcontroller from the actual end product; IAP means that the device can write non-volatile data in the Flash memory while the application program is running. There needs no external high voltage for programming due to its built-in charge-pumping circuitry.

The MG82FE(L)308/316 retains all features of the standard 80C52 with 256 bytes of scratch-pad RAM, four 8bit I/O ports, two external interrupts, a multi-source 4-level interrupt controller and three timer/counters. In addition, the MG82FE(L)308/316 has four extra I/O ports (P4[6:0], P5, P6[1:0], P7), an XRAM of 256 bytes, four extra external interrupts with High/Low trigger option, a PWM Timer, a one-time enabled Watchdog Timer, a precision analog comparator with multi-channel inputs and on-chip VDD reference, a Brown-out Detector, an on-chip crystal oscillator(shared with P6.0 and P6.1), a high precision internal oscillator, a more versatile serial channel that facilitates multiprocessor communication (EUART) and a speed improvement mechanism (extra X2/X4 mode).

The MG82FE(L)308/316 has two power-saving modes and an 8-bit system clock pre-scaler to reduce the power consumption. 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 the external interrupts. And, the user can further reduce the power consumption by using the 8-bit system clock pre-scaler to slow down the operating speed.

2. Block Diagram



3. Special Function Register

3.1. SFR Map

	SFR Page	**0/8**	**1/9**	**2/A**	**3/B**	**4/C**	**5/D**	**6/E**	**7/F**
F8	0 F	P5	--	CCAP0H	--	--	--	--	--
F0	0 F	B	--	--	--	--	--	--	--
E8	0 F	P4	--	CCAP0L	--	--	--	--	--
E0	0 F	ACC	WDTCR	IFD	IFADRH	IFADRL	IFMT	SCMD	ISPCR
D8	**0** **F**	**CCON** **P7**	CMOD	--	--	--	--	--	--
D0	0 F	PSW	--	--	--	--	--	GPWKPE	P1WKPE
C8	**0** **F**	**T2CON** **P6**	T2MOD	RCAP2L	RCAP2H	TL2	TH2	--	--
C0	0 F	XIFLG	XICON0	XICON1	--	--	--	--	CKCON0
B8	0 F	IP0L	SADEN	--	--	--	--	--	CKCON1
B0	0 F	P3	P3M0	P3M1	P4M0	P4M1	P5M0	P5M1	IP0H
A8	0 F	IE	SADDR	--	--	SFRPI	EIE1	EIP1L	EIP1H
A0	0 F	P2	--	AUXR1	AUXR2	--	--	--	--
98	0 F	SCON	SBUF	--	--	--	--	ACCON	ACMOD
90	0 F	P1	P1M0	P1M1	P0M0	P0M1	P2M0	P2M1	PCON1
88	0 F	TCON	TMOD	TL0	TL1	TH0	TH1	AUXR0	
80	0 F	P0	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

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	
P0	Port 0	80H	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	11111111B
SP	Stack Pointer	81H									00000111B
DPL	Data Pointer Low	82H									00000000B
DPH	Data Pointer High	83H									00000000B
PCON0	Power Control 0	87H	SMOD1	SMOD0	GF	POF	GF1	GF0	PD	IDL	00010000B
TCON	Timer Control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00000000B
TMOD	Timer Mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00000000B
TL0	Timer Low 0	8AH									00000000B
TL1	Timer Low 1	8BH									00000000B
TH0	Timer High 0	8CH									00000000B
TH1	Timer High 1	8DH									00000000B
AUXR0	Auxiliary Register 0	8EH	P60OC1	P60OC0	P60FD	P34FD	--	--	EXTRAM	--	0000xx0xB
P1	Port 1	90H	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	11111111B
P1M0	P1 Mode Register 0	91H	P1M0.7	P1M0.6	P1M0.5	P1M0.4	P1M0.3	P1M0.2	P1M0.1	P1M0.0	00000000B
P1M1	P1 Mode Register 1	92H	P1M1.7	P1M1.6	P1M1.5	P1M1.4	P1M1.3	P1M1.2	P1M1.1	P1M1.0	00000000B
P0M0	P0 Mode Register 0	93H	P0M0.7	P0M0.6	P0M0.5	P0M0.4	P0M0.3	P0M0.2	P0M0.1	P0M0.0	00000000B
P0M1	P0 Mode Register 1	94H	P0M1.7	P0M1.6	P0M1.5	P0M1.4	P0M1.3	P0M1.2	P0M1.1	P0M1.0	00000000B
P2M0	P2 Mode Register 0	95H	P2M0.7	P2M0.6	P2M0.5	P2M0.4	P2M0.3	P2M0.2	P2M0.1	P2M0.0	00000000B
P2M1	P2 Mode Register 1	96H	P2M1.7	P2M1.6	P2M1.5	P2M1.4	P2M1.3	P2M1.2	P2M1.1	P2M1.0	00000000B
PCON1	Power Control 1	97H	--	--	BORF	--	--	--	--	BOD	xxxxxxx0B
SCON	Serial Control	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	Ti	RI	00000000B
SBUF	Serial Buffer	99H									xxxxxxxxB
ACCON	Analog Comp. Control	9EH	ACIDX	ACPDX	ACOUT	ACF	ACEN	ACM2	ACM1	ACM0	00x00000B
ACMOD	Analog Comp. Mode	9FH	MVRS3	MVRS2	MVRS1	MVRS0	--	--	PIS1	PIS0	0000xx00B
P2	Port 2	A0H	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	11111111B
AUXR1	Auxiliary Register 1	A2H	GPWKS1	GPWKS0	P5PWM	P1S0	GF2	--	--	DPS	00000xx0B
AUXR2	Auxiliary Register 2	A3H	T0X12	T1X12	URM0X6	--	--	--	T1CKOE	T0CKOE	0000xx00B
IE	Interrupt Enable	A8H	EA	GF4	ET2	ES	ET1	EX1	ET0	EX0	00000000B
SADDR	Slave Address	A9H									00000000B
SFRPI	SFR Page Index	ACH	--	--	--	--	IDX3	IDX2	IDX1	IDX0	xxxx0000B
EIE1	Extended INT Enable 1	ADH	--	--	--	--	--	--	EACI	EBOI	xxxxxx00B
EIP1L	Ext. INT Priority 1 Low	AEH	--	--	--	PX5L	PX4L	PPTL	PACL	PBOL	xxx00000B
EIP1H	Ext. INT Priority 1 High	AFH	--	--	--	PX5H	PX4H	PPTH	PACH	PBOH	xxx00000B
P3	Port 3	B0H	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	11111111B
P3M0	P3 Mode Register 0	B1H	P3M0.7	P3M0.6	P3M0.5	P3M0.4	P3M0.3	P3M0.2	P3M0.1	P3M0.0	00000000B
P3M1	P3 Mode Register 1	B2H	P3M1.7	P3M1.6	P3M1.5	P3M1.4	P3M1.3	P3M1.2	P3M1.1	P3M1.0	00000000B
P4M0	P4 Mode Register 0	B3H	--	P4M0.6	P4M0.5	P4M0.4	P4M0.3	P4M0.2	P4M0.1	P4M0.0	x00000xxB
P4M1	P4 Mode Register 1	B4H	--	P4M1.6	P4M1.5	P4M1.4	P4M1.3	P4M1.2	P4M1.1	P4M1.0	x0000000B
P5M0	P5 Mode Register 0	B5H	P5M0.7	P5M0.6	P5M0.5	P5M0.4	P5M0.3	P5M0.2	P5M0.1	P5M0.0	00000000B
P5M1	P5 Mode Register 1	B6H	P5M1.7	P5M1.6	P5M1.5	P5M1.4	P5M1.3	P5M1.2	P5M1.1	P5M1.0	00000000B
IP0H	Interrupt Priority 0 High	B7H	PX3H	PX2H	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	00000000B
IP0L	Interrupt Priority Low	B8H	**PX3L**	**PX2L**	PT2L	PSL	PT1L	PX1L	PT0L	PX0L	00000000B
SADEN	Slave Address Mask	B9H									00000000B
CKCON1	Clock Control 1	BFH	OSCDR	--	--	XCKS4	XCKS3	XCKS2	XCKS1	XCKS0	xxx01010B
XIFLG	External INT Flag	C0H	--	--	--	--	IE5	IE4	IE3	IE2	xxxx0000B
XICON0	External INT Control 0	C1H	--	INT3H	IT3	EX3	--	INT2H	IT2	EX2	x000x000B
XICON1	External INT Control 1	C2H	--	INT5H	IT5	EX5	--	INT4H	IT4	EX4	x000x000B
CKCON0	Clock Control 0	C7H	--	--	--	--	--	SCKS2	SCKS1	SCKS0	xxxxx000B
T2CON	Timer 2 Control	C8H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL	00000000B
P6	Port 6	C8H	--	--	--	--	--	--	P6.1	P6.0	xxxxxx11B
T2MOD	Timer2 mode	C9H	--	--	--	T2X12	--	--	T2OE	DCEN	xxx0xx00B
RCAP2L	Timer2 Capture Low	CAH									00000000B
RCAP2H	Timer2 Capture High	CBH									00000000B
TL2	Timer Low 2	CCH									00000000B
TH2	Timer High 2	CDH									00000000B
PSW	Program Status Word	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	00000000B
GPWKPE	General Port WKPE	D6H	GP7WE	GP6WE	GP5WE	GP4WE	GP3WE	GP2WE	GP1WE	GP0WE	00000000B
P1WKPE	P1 Wakeup Enable	D7H	P17WE	P16WE	P15WE	P14WE	P13WE	P12WE	P11WE	P10WE	00000000B
CCON	Counter Control Reg.	D8H	CF	CR	-	-	PWMEN	-	-	-	00xx0xxxB
P7	Port 7	D8H	P7.7	P7.6	P7.5	P7.4	P7.3	P7.2	P7.1	P7.0	11111111B
CMOD	Counter Mode Reg.	D9H	CIDL	POS2	POS1	POS0	CPS2	CPS1	CPS0	ECF	00000000B
ACC	Accumulator	E0H	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	00000000B
WDTCR	Watch-dog-timer Control register	E1H	WRF	--	ENW	CLW	WIDL	PS2	PS1	PS0	0x000000B
IFD	ISP Flash data	E2H									11111111B
IFADRH	ISP Flash address	E3H									00000000B

	High										
IFADRL	ISP Flash Address Low	E4H									00000000B
IFMT	ISP Mode Table	E5H	--	--	--	--	MS3	MS2	MS1	MS0	xxxx0000B
IAPLB	IAP Low Boundary	Note 1	IAPLB6	IAPLB5	IAPLB4	IAPLB3	IAPLB2	IAPLB1	IAPLB0	--	11111111B
SCMD	ISP Serial Command	E6H									xxxxxxxxB
ISPCR	ISP Control Register	E7H	ISPEN	BS	SRST	CFAIL	--	--	--	--	0000xxxxB
P4	Port 4	E8H	--	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0	11111111B
CCAP0L	PWM Counter L-Duty	EAH									00000000B
B	B Register	F0H	F7H	F6H	F5H	F4H	F3H	F2H	F1H	F0H	00000000B
P5	Port 5	F8H	P5.7	P5.6	P5.5	P5.4	P5.3	P5.2	P5.1	P5.0	11111111B
CCAP0H	PWM Counter H-Duty	FAH									00000000B

Note1: The registers are addressed by IFMT and SCMD. Please refer the IFMT register description for more detail information.

3.3. SFR Paging

The MG82FE(L)308/316 features SFR paging, allowing the device to map many SFRs into the 0x80 to 0xFF memory address space. The SFR memory space has 16 pages. In this way, each memory location from 0x80 to 0xFF can access up to 128 SFRs. The MG82FE(L)308/316 utilizes two SFR pages: 0 and F. SFR pages are selected using the Special Function Register Page Index register, SFRPI. The procedure for reading and writing an SFR is as follows:
1. Select the appropriate SFR page number using the SFRPI register.
2. Use direct accessing mode to read or write the special function register (MOV instruction).

SFRPI: SFR Page Index Register
SFR Address = 0xAC
SFR Page = All   Reset Value = XXXX-0000

7	6	5	4	3	2	1	0
--	--	--	--	PIDX3	PIDX2	PIDX1	PIDX0
R	**R**	**R**	**R**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~4: Reserved for testing. Must write “0” on these bits.

Bit 3~0: SFR Page Index. The available pages are only page “0” and “F”.

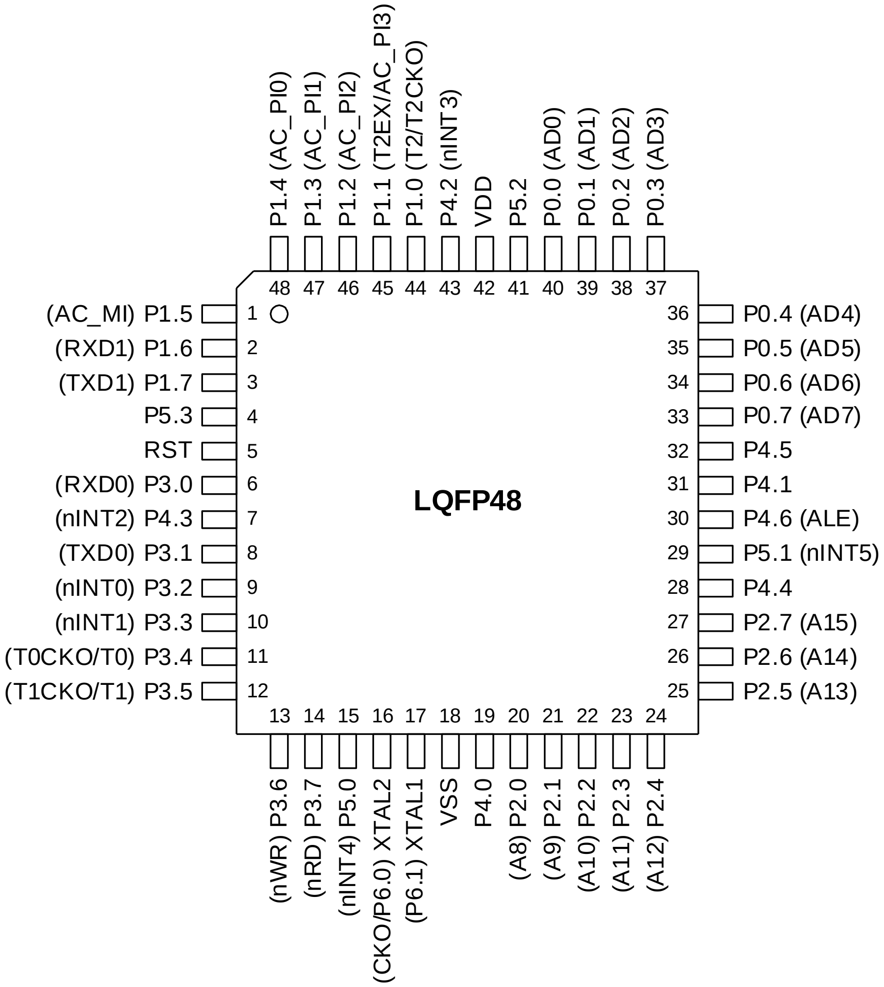
PIDX[3:0]	Selected Page
0000	Page 0
0001	Page 1
0010	Page 2
0011	Page 3
……	……
……	……
……	……
1111	Page F

There are two registers in Page 0 only, T2CON(C8H) and CCON(D8H), and two registers in Page F only, P6(C8H) and P7(D8H). Other registers are accessed in both page “0” and page “F”.

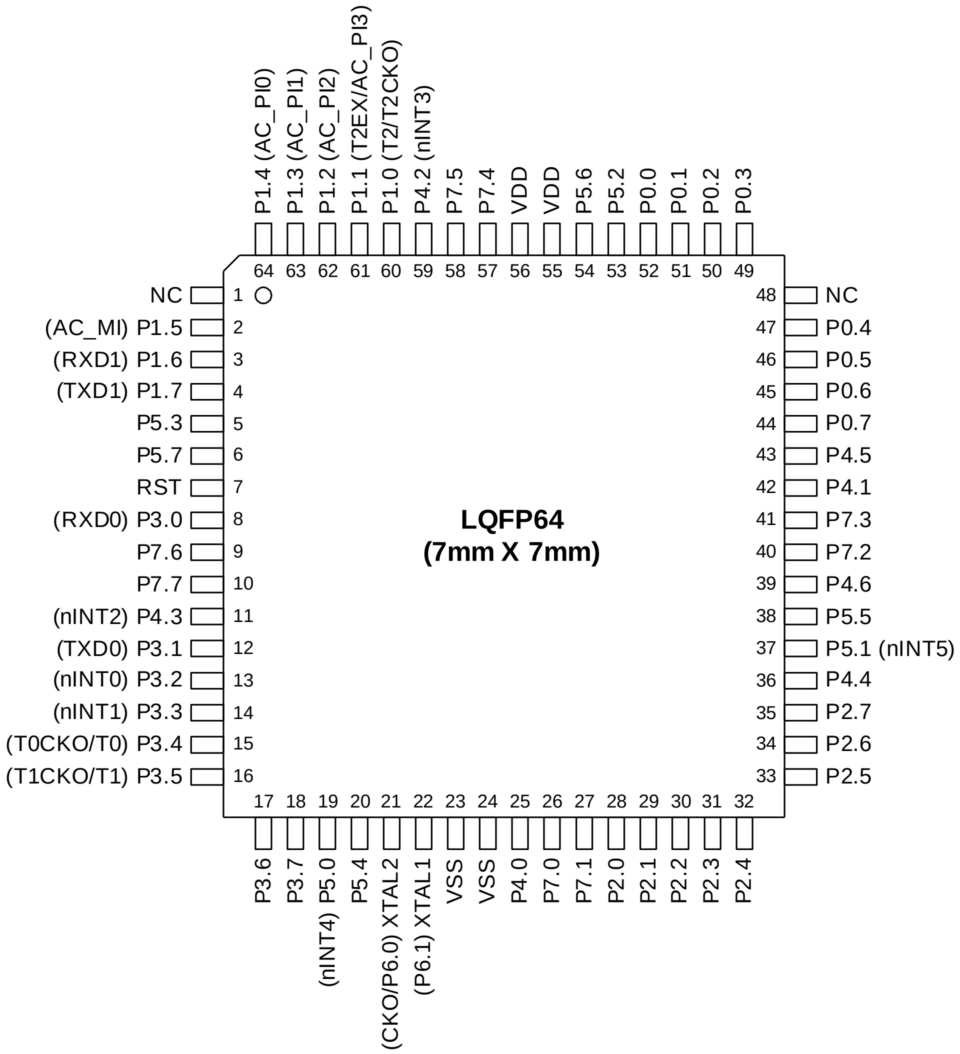
4. Pin Configurations

4.1. Package Instruction

MG82FE(L)308AD48



MG82E(L)316AD64

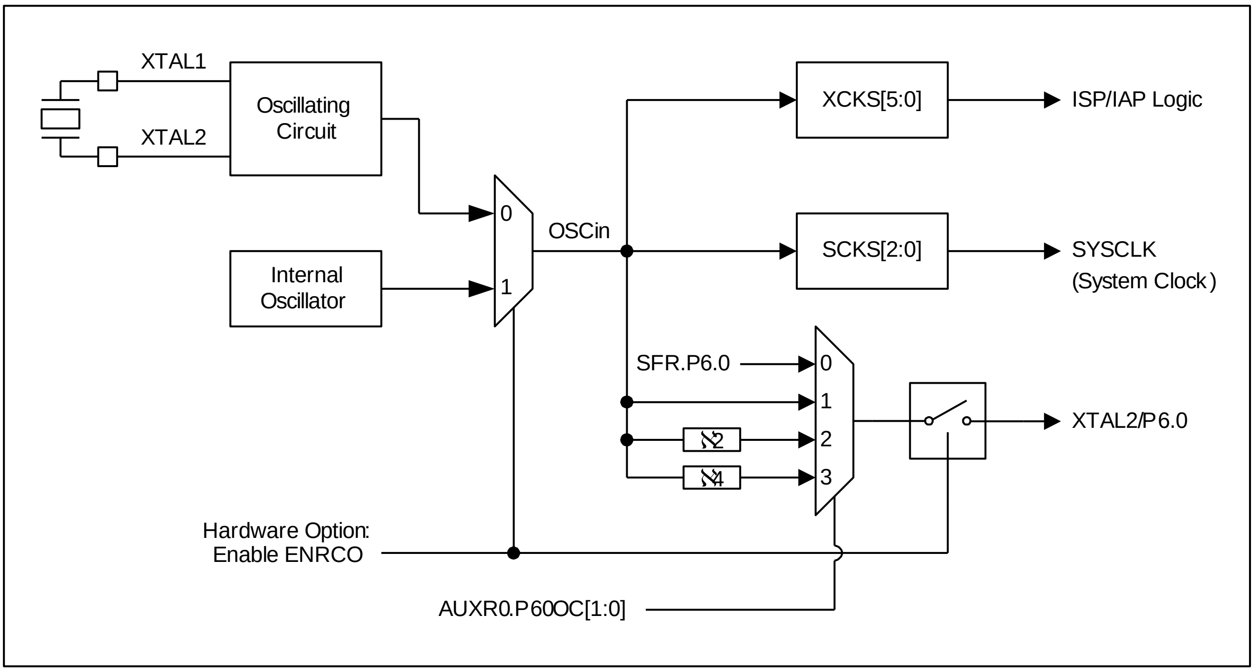


4.2. Pin Description

MNEMONIC	64-Pin LQFP	48-Pin LQFP	TYPE	DESCRIPTION
P1.0 ~ P1.4 P1.5 ~ P1.7	60~64 2~4	44~48 1~3	I/O I/O	**Port 1**: General-purposed I/O Port 1. P10 is T2 or T2CKO. P11 is T2EX. P11 ~ P14 are the programmable comparator positive inputs. Default input is P14. P15 is the comparator negative input. P16/P17 have an alternated function for RXD/TXD by firmware configured.
P3.0 P3.1 ~ P3.3 P3.4 P3.5 P3.6 ~ P3.7	8 12~14 15 16 17,18	6 8~10 11 12 13,14	I/O I/O I/O I/O I/O	**Port 3**: General-purposed I/O port 3. P30 is UART RXD. P31 is UART TXD. P32 is nINT0. P33 is nINT1. P34 is T0 or T0CKO. P35 is T1 or T1CKO.
P0.7 ~ P0.0	44~47, 49~52	33~40	I/O	**Port 0**: General-purposed I/O port 0.
P2.0 ~ P2.7	28~35	20~27	I/O	**Port 2**: General-purposed I/O port 2. It is also the PWM output from PWM Timer.
P4.0 P4.1 P4.2 P4.3 P4.4 P4.5 P4.6	25 42 59 11 36 43 39	19 31 43 7 28 32 30	I/O I/O I/O I/O I/O I/O I/O	**Port 4**: General-purposed I/O port 4. P40 and P41 can be configured to input only in default. P42 is nINT3. P43 is nINT2.
P5.0 ~ P5.1 P5.2 ~ P5.3 P5.4 ~ P5.5 P5.6 ~ P5.7	19,37 53,5 20,38 54,6	15,29 41,4 - -	I/O I/O I/O I/O	**Port 5**: General-purposed I/O port 5. It is also the PWM output from PWM Timer. P5.0 is nINT4. P5.1 is nINT5.
P7.0 ~ P7.1 P7.2 ~ P7.3 P7.4 ~ P7.5 P7.6 ~ P7.7	26,27 40,41 57,58 9,10	- - - -	I/O I/O I/O I/O	**Port 7:** General-purposed I/O port 7.
RST	7	5	I	**RST**: A high on this pin for at least two machine cycles will reset the device.
XTAL1 P6.1	 22	 17	I I/O	**Crystal1**: Input to the inverting oscillator amplifier. XTAL1 has an alternate function for P6.1.
XTAL2 P6.0	 21	 16	O I/O	**Crystal2**: Output from the inverting amplifier. XTAL2 has an alternate function for P6.0 and internal clock output.
VDD	55,56	42	P	**Power**
VSS	23,24	18	G	**Ground**

5. System Clock

5.1. Clock Structure



5.2. Clock Register

CKCON0: Clock Control Register 0

SFR Address = 0xC7

SFR Page = All Reset Value = xxxx-x000

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

Bit 7~3: Reserved.

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

SCKS[2:0]	System Clock (Fosc)
0 0 0	CLKin
0 0 1	CLKin /2
0 1 0	CLKin /4
0 1 1	CLKin /8
1 0 0	CLKin /16
1 0 1	CLKin /32
1 1 0	CLKin /64
1 1 1	CLKin /128

CKCON1: Clock Control Register 1

SFR Address = 0xBF

SFR Page = All Reset Value = xxx0-1010

7	6	5	4	3	2	1	0

OSCDR	--	--	XCKS4	XCKS3	XCKS2	XCKS1	XCKS0
R/W	**R**	**R**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: OSCDR, OSC Driving control Register. Reset value is load from OSCDN (in hardware option). If OSCDN in hardware option is enabled, the OSCDR is set to “1” after power on. Otherwise, it is set to “0”. And it could be read/written by CPU.
0: The driving of crystal oscillator is enough for oscillation up to 25MHz.
1: The driving of crystal oscillator is reduced. It will helpful in EMI reduction. Regarding application not needing high frequency clock, below 16MHz, it is recommended to do so.

Bit 6~5: Reserved.

Bit 4~0: This is set the crystal frequency value to define the time base of ISP/IAP programming. Fill with a proper value according to OSCin, as listed below.

[XCKS4~XCKS0] = OSCin – 1, w*here OSCin=1~25 (MHz).*

For examples,
(1) If OSCin=12MHz, then fill [XCKS4~XCKS0] with 11, i.e., 001011B.
(2) If OSCin=6MHz, then fill [XCKS4~XCKS0] with 5, i.e., 000101B.

OSCin	XCKS[4:0]
1MHz	5’b00000
2MHz	5’b00001
3MHz	5’b00010
4MHz	5’b00011
……	……
22MHz	5’b10101
23MHz	5’b10110
24MHz	5’b10111
25MHz	5’b11000

The default value of XCKS= 5’b01010 for OSCin= 11MHz.

AUXR0: Auxiliary Register 0
SFR Address = 0x8E
SFR Page = All Reset Value = 0000-0000

7	6	5	4	3	2	1	0
P60OC1	P60OC0	P60FD	P34FD	--	--	--	--
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~6: P60 output configured control bit 1 and 0. The two bits only act when internal RC oscillator is selected for system clock source. In this condition, XTAL2 and XTAL1 are the alternated function for P60 and P61. P60 provides the following selections for GPIO or clock source generator. When P60OC[1:0] index to non-P60 function, XTAL2 will drive the on-chip RC oscillator output to provide the clock source for other devices.

P60OC[1:0]	XTAL2 function	I/O mode
00	P60	Quasi-bidirectional I/O
01	INTOSC	Push-Pull Output
10	INTOSC/2	Push-Pull Output
11	INTOSC/4	Push-Pull Output

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

6. 8051 CPU Function Description

6.1. CPU Register

PSW: Program Status Word
SFR Address = 0xD0
SFR Page = All Reset Value = 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 the on-chip-data-RAM section. 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 Address = 0x81
SFR Page = All Reset Value = 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**

DPL: Data Pointer Low
SFR Address = 0x82
SFR Page = All Reset Value = 0000-0000

7	6	5	4	3	2	1	0
DPL[7]	DPL[6]	DPL[5]	DPL[4]	DPL[3]	DPL[2]	DPL[1]	DPL[0]
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

DPH: Data Pointer High
SFR Address = 0x83
SFR Page = All Reset Value = 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**

B: B Register
SFR Address = 0xF0
SFR Page = All Reset Value = 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**

6.2. CPU Timing

The MG82FE(L)308/316 is a single-chip microcontroller based on a high performance 1-T architecture 80C51 CPU that has an 8051 compatible instruction set, and executes instructions in 1~7 clock cycles (about 6~7 times the rate of a standard 8051 device). It employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. The instruction timing is different than that of the standard 8051.

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

6.3. 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.

7. Memory Organization

Like all 80C51 devices, the MG82FE(L)308/316 has separate address spaces for program and data memory. The logical separation of program and data memory allows the data memory to be accessed by 8-bit addresses, which can be quickly stored and manipulated by the 8-bit CPU.

Program memory (ROM) can only be read, not written to. There can be up to 16K bytes of program memory. In the MG82FE(L)308/316, all the program memory are on-chip Flash memory, and without the capability of accessing external program memory because of no External Access Enable (/EA) and Program Store Enable (/PSEN) signals designed.

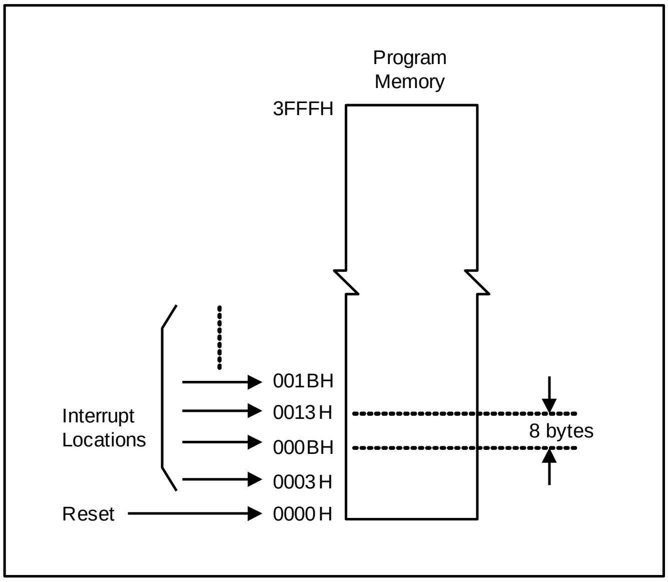
Data memory occupies a separate address space from program memory. In the MG82FE(L)308/316, there are 256 bytes of internal scratch-pad RAM and 256 bytes of on-chip expanded RAM(XRAM).

7.1. On-Chip Program Flash

Program memory is the memory which stores the program codes for the CPU to execute, as shown in Fig 8-1. 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.

Fig 8-1 Program Memory



7.2. On-Chip Data RAM

Fig 8-1 shows the internal and external data memory spaces available to the MG82FE(L)308/316 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.

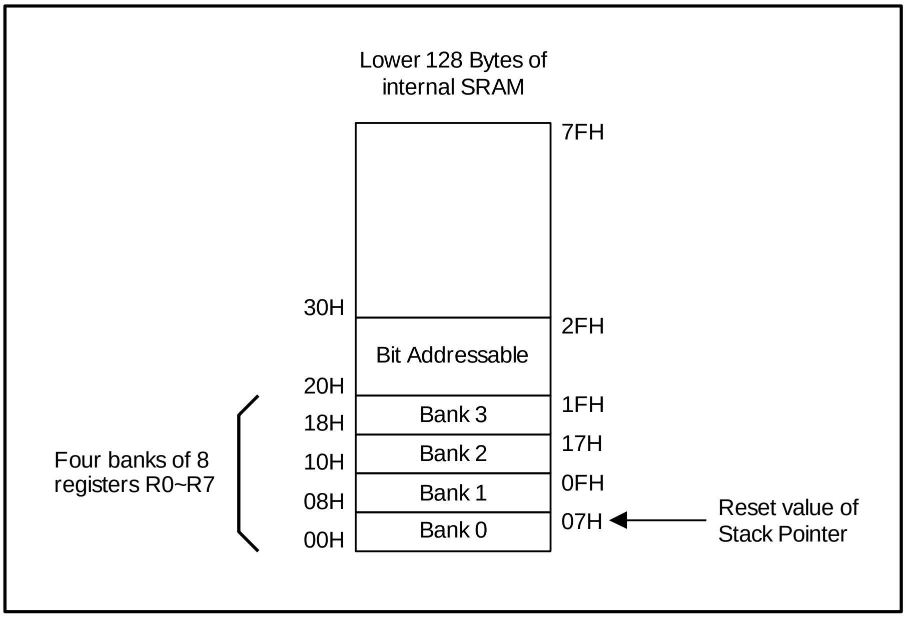
The lower 128 bytes of RAM are present in all 80C51 devices as mapped in Fig 8-2. The lowest 32 bytes are grouped into 4 banks of 8 registers. Program instructions call out these registers as R0 through R7. Two bits in the Program Status Word (PSW) select which register bank is in use. This allows more 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 8-1 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.

To access the on-chip expanded RAM (XRAM), refer to Figure 8-1, the 256 bytes of XRAM (0000H to 00FFH) are indirectly accessed by move external instruction, MOVX. An access to XRAM will have not any outputting of address, address latch enable and read/write strobe.

Fig 8-2 Lower 128 Bytes of Internal RAM


Lower 128 Bytes of internal SRAM
7FH
30H
2FH
Bit Addressable
20H
1FH
18H
Bank 3
17H
10H
Bank 2
0FH
08H
Bank 1
07H
00H
Bank 0
Four banks of 8 registers R0~R7
Reset value of Stack Pointer

7.3. On-chip expanded RAM (XRAM)

To access the on-chip expanded RAM (XRAM), refer to Fig 8-1, the 256 bytes of XRAM (0000H to 00FFH) are indirectly accessed by move external instruction, “MOVX @Ri” and “MOVX @DPTR”. For KEIL-C51 compiler, to assign the variables to be located at XRAM, the “pdata” or “xdata” definition should be used. After being compiled, the variables declared by “pdata” and “xdata” will become the memories accessed by “MOVX @Ri” and “MOVX @DPTR”, respectively. Thus the MG82FE(L)308/316 hardware can access them correctly.

7.4. Declaration Identifiers in a C51-Compiler

The declaration identifiers in a C51-compiler for the various MG82FE(L)308/316 memory spaces are as follows:

data
128 bytes of internal data memory space (00h~7Fh); accessed via direct or indirect addressing, using instructions other than MOVX and MOVC. All or part of the Stack may be in this area.

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

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

xdata
External data or on-chip eXpanded RAM (XRAM); duplicates the classic 80C51 64KB memory space addressed via the "MOVX @DPTR" instruction. The MG82FE(L)308/316 has 256 bytes of on-chip xdata memory.

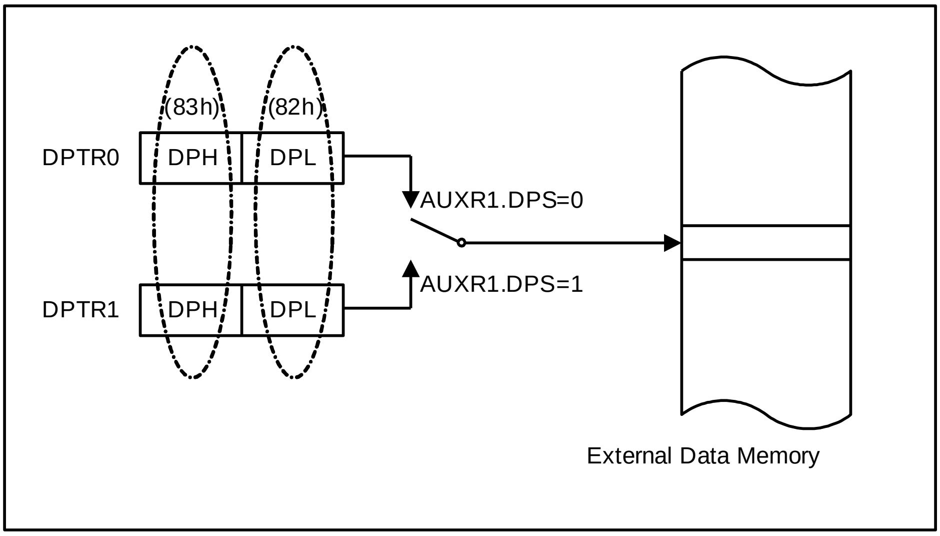
pdata
Paged (256 bytes) external data or on-chip eXpanded RAM; duplicates the classic 80C51 256 bytes memory space addressed via the "MOVX @Ri" instruction. The MG82FE(L)308/316 has 256 bytes of on-chip pdata memory which is shared with on-chip xdata memory.

code
16K bytes of program memory space; accessed as part of program execution and via the "MOVC @A+DTPR" instruction. The MG82FE(L)308/316 has 16K bytes of on-chip code memory.

8. Dual Data Pointer Register (DPTR)

The dual DPTR structure as shown in Fig9-1 is a way by which the chip can specify the address of an external data memory location. There are two 16-bit DPTR registers that address the external memory, and a single bit called DPS (AUXR1.0) that allows the program code to switch between them.

Fig9-1 Dual DPTR



DPTR Instructions

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

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

AUXR1: Auxiliary Control Register 1

SFR Address = 0xA2

SFR Page = All Reset Value = 0000-0xx0

7	6	5	4	3	2	1	0
GPWKS1	GPWKS0	P5PWM	P1S0	GF2	GF	GF	DPS
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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

DPS	Selected DPTR
0	DPTR0
1	DPTR1

9. Configurable I/O Ports

The MG82FE(L)308/316 has following I/O ports: P0.0~P0.7, P1.0~P1.7, P2.0~P2.7, P3.0~P3.7, P4.0~P4.6, P5.0~P5.7, P6.0~P6.1 and P7.0~P7.7. If select internal oscillator as system clock input, XTAL2 and XTAL1 are configured to Port 6.0 and Port 6.1. The exact number of I/O pins available depends upon the package types. See Table 10-1.

Table 10-1 Number of I/O Pins Available

Package Type	I/O Pins	Number of I/O ports
64-pin LQFP	P0.0~P0.7, P1.0~P1.7, P2.0~P2.7, P3.0~P3.7, P4.0~P4.6, P5.0~P5.7, P7.0~P7.7, XTAL2(P6.0), XTAL1(P6.1)	55 or 57 (INTOSC enabled)
48-pin LQFP	P0.0~P0.7, P1.0~P1.7, P2.0~P2.7, P3.0~P3.7, P4.0~P4.6, P5.0~P5.3, XTAL2(P6.0), XTAL1(P6.1)	43 or 45 (INTOSC enabled)

9.1. IO Structure

Except P6.0~P6.1 and P7.0~P7.7, all I/O port pins can be configured to one of four operating modes. These are: quasi-bidirectional (standard 8051 I/O port), push-pull output, input-only (high-impedance input) and open-drain output. P6.0 and P6.1 are only one I/O mode for quasi-bidirectional ports.

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

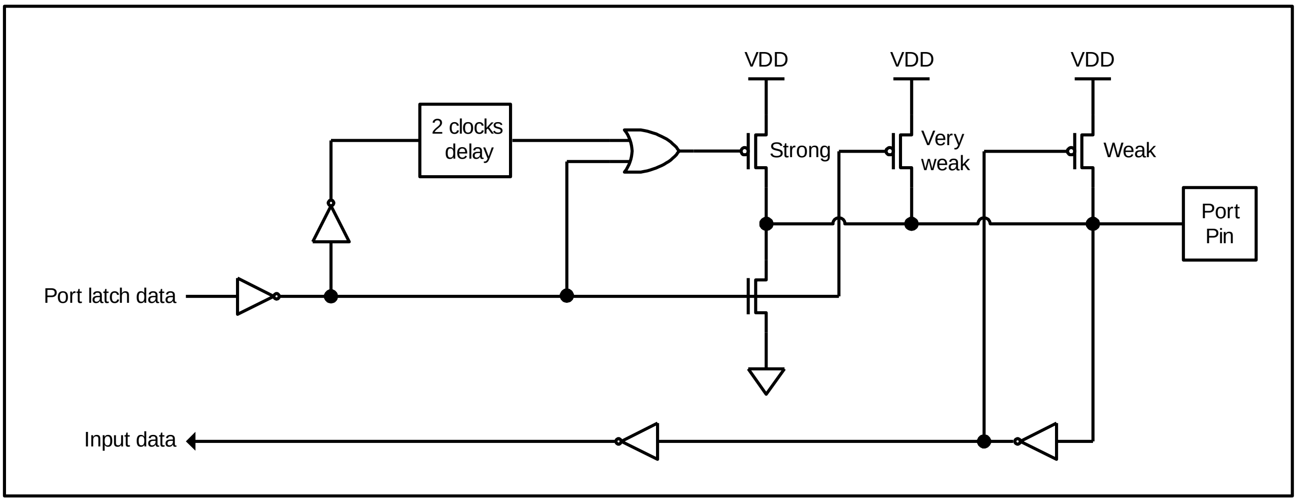
9.1.1. Quasi-Bidirectional IO Structure

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

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

The quasi-bidirectional port configuration is shown in Figure 10-1.

Figure 10-1 Quasi-Bidirectional I/O

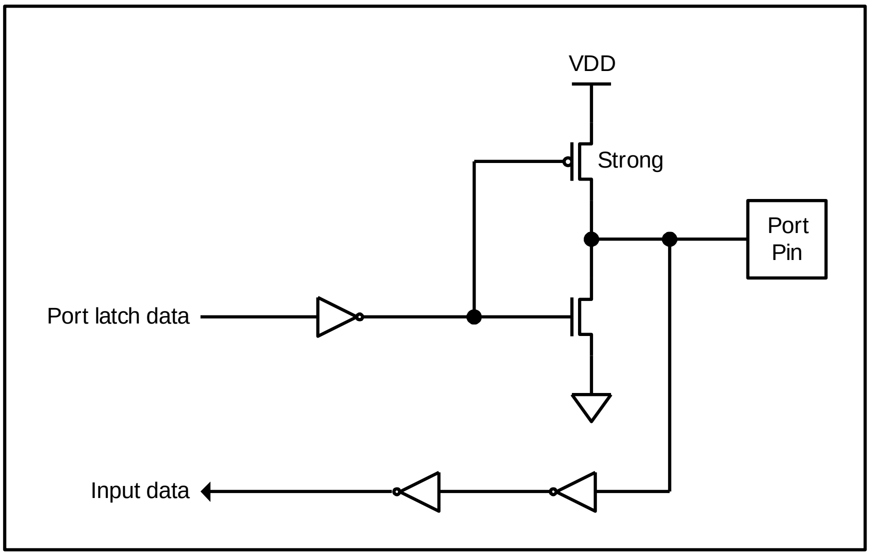


9.1.2. Push-Pull Output Structure

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

The push-pull port configuration is shown in Figure 10-2.

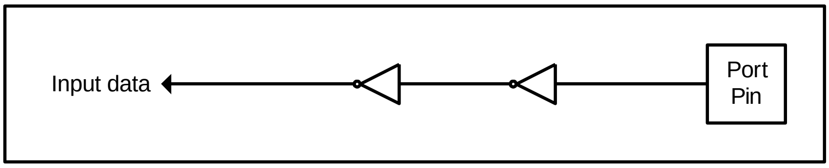
Figure 10-2 Push-Pull Output



9.1.3. Input-Only (High Impedance Input) Structure

The input-only configuration is a input without any pull-up resistors on the pin, as shown in Figure 10-3.

Figure 10-3 Input-Only

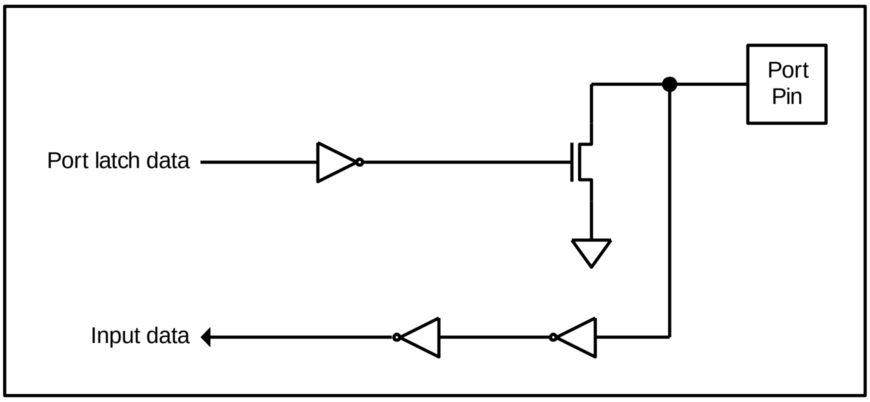


9.1.4. Open-Drain Output Structure

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

The open-drain port configuration is shown in Figure 10-4.

Figure 10-4 Open-Drain Output



9.2. I/O Port Register

All I/O port pins on the MG82FE(L)308/316 may be individually and independently configured by software to one of four types on a bit-by-bit basis, as shown in Table 10-2. Two mode registers for each port select the output type for each port pin.

Table 10-2 Port Configuration Settings

PxM0.y	PxM1.y	Port Mode
0	0	Quasi-Bidirectional
0	1	Push-Pull Output
1	0	Input Only (High Impedance Input)
1	1	Open-Drain Output

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

9.2.1. Port 0 Register

P0: Port 0 Register
SFR Address = 0x80
SFR Page = All Reset Value = 1111-1111

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

Bit 7~0: P0.7~P0.0 could be set/cleared by CPU.

P0M0: Port 0 Mode Register 0
SFR Address = 0x93
SFR Page = All Reset Value = 0000-0000

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

P0M1: Port 0 Mode Register 1
SFR Address = 0x94
SFR Page = All Reset Value = 0000-0000

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

9.2.2. Port 1 Register

P1: Port 1 Register
SFR Address = 0x90
SFR Page = All Reset Value = 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 Address = 0x91
SFR Page = All Reset Value = 0000-0000

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

P1M1: Port 1 Mode Register 1
SFR Address = 0x92
SFR Page = All Reset Value = 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**

If any one of P1.1 ~ P1.5 is set as comparator analog input, must configure the selected I/O mode to input only to get the analog signal for comparator.

9.2.3. Port 2 Register

P2: Port 2 Register
SFR Address = 0xA0
SFR Page = All Reset Value = 1111-1111

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

Bit 7~0: P2.7~P2.0 could be only set/cleared by CPU. Or it also can be toggled on addressed port channel by PWM Timer underflow event in PWM mode.

P2M0: Port 2 Mode Register 0

SFR Address = 0x95
SFR Page = All Reset Value = 0000-0000

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

P2M1: Port 2 Mode Register 1

SFR Address = 0x96
SFR Page = All Reset Value = 0000-0000

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

9.2.4. Port 3 Register

P3: Port 3 Register

SFR Address = 0xB0
SFR Page = All Reset Value = 1111-1111

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

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

P3M0: Port 3 Mode Register 0

SFR Address = 0xB1
SFR Page = All Reset Value = 0000-0000

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

P3M1: Port 3 Mode Register 1

SFR Address = 0xB2
SFR Page = All Reset Value = 0000-0000

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

9.2.5. Port 4 Register

P4: Port 4 Register

SFR Address = 0xE8
SFR Page = All Reset Value = x111-1111

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

Bit 6~0: P4.6~P4.0 could be only set/cleared by CPU.

P4M0: Port 4 Mode Register 0

SFR Address = 0xB3
SFR Page = All Reset Value = x000-00xx

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

The reset value of P4M0.1 and P4M0.0 are determined by hardware options, P41IOE and P40IOE. If the hardware options of P41IOE and P41IOE are turned on, the reset value of P4M0.1/P4M0.0 are set to “1” to select P4.1/P4.0 to “Input-Only” mode individually. Otherwise, the reset value of P4M0.1/P4M0.0 are cleared to “0” to select P4.1/P4.0 to “quasi-bidirectional” mode individually.

P4M1: Port 4 Mode Register 1

SFR Address = 0xB4
SFR Page = All Reset Value = x000-0000

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

9.2.6. Port 5 Register

P5: Port 5 Register

SFR Address = 0xF8
SFR Page = All Reset Value = 1111-1111

7	6	5	4	3	2	1	0
P5.7	P5.6	P5.5	P5.4	P5.3	P5.2	P5.1	P5.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~0: P5.7~P5.0 could be only set/cleared by CPU. P5 is also PWM output when PWM timer is enabled running and AUXR1.P5PWM is enabled to switch PWM output from P2 to P5.

P5M0: Port 5 Mode Register 0

SFR Address = 0xB5
SFR Page = All Reset Value = 0000-0000

7	6	5	4	3	2	1	0
P5M0.7	P5M0.6	P5M0.5	P5M0.4	P5M0.3	P5M0.2	P5M0.1	P5M0.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

P5M1: Port 5 Mode Register 1

SFR Address = 0xB6
SFR Page = All Reset Value = 0000-0000

7	6	5	4	3	2	1	0
P5M1.7	P5M1.6	P5M1.5	P5M1.4	P5M1.3	P5M1.2	P5M1.1	P5M1.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

9.2.7. Port 6 Register

P6: Port 6 Register

SFR Address = 0xC8
SFR Page = F Reset Value = xxxx-xx11

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

Bit 7~2: Reserved.

Bit 1~0: P6.1~P6.0 could be only set/cleared by CPU. These two I/Os are active when Internal Oscillator is enabled for system clock. Then, XTAL1 and XTAL2 behave P6.1 and P6.0. They only support one I/O mode, quasi-bidirectional mode. The register is only accessed in SFR page “F”.

9.2.8. Port 7 Register

P7: Port 7 Register

SFR Address = 0xD8

SFR Page = F Reset Value = 1111-1111

7	6	5	4	3	2	1	0
P7.7	P7.6	P7.5	P7.4	P7.3	P7.2	P7.1	P7.0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~0: P7.7~P7.0 could be only set/cleared by CPU. Port 7 only supports one I/O mode, quasi-bidirectional mode. The register is only accessed in SFR page “F”.

9.3. GPIO Sample Code

(1). Required Function: Set P1.0 to input-only mode

Assembly Code Example:

```
P1Mn0               EQU               01h

      ORL        P1M0, #P1Mn0
      ANL        P1M1, #(0FFh + P1Mn0)              ; Configure P1.0 to input only mode
      SETB       P1.0                                 ; Set P1.0 data latch to "1" to enable input mode
```

C Code Example:

```
#define       P1Mn0              0x01

      P1M0 |= P1Mn0;
      P1M1 &= ~P1Mn0;                                   // Configure P1.0 to input only mode
      P10 = 1;                                        // Set P1.0 data latch to "1" to enable input mode
```

(2). Required Function: Set P1.0 to push-pull output mode

Assembly Code Example:

```
P1Mn0               EQU               01h

      ANL        P1M0, #(0FFh - P1Mn0)
      ORL        P1M1, #P1Mn0                      ; Configure P1.0 to push pull mode
      SETB       P1.0
```

C Code Example:

```
#define       P1Mn0              0x01

      P1M0 &= ~P1Mn0;
      P1M1 |= P1Mn0;                                // Configure P1.0 to push pull mode
      P10 = 1;                                        // Set P1.0 data latch to "1" to enable push pull mode
```

(3). Required Function: Set P1.0 to open-drain output mode

Assembly Code Example:

```
P1Mn0               EQU               01h

      ORL        P1M0, #P1Mn0
      ORL        P1M1, #P1Mn0                         ; Configure P1.0 to open drain mode
      SETB       P1.0                                 ; Set P1.0 data latch to "1" to enable open drain mode
```

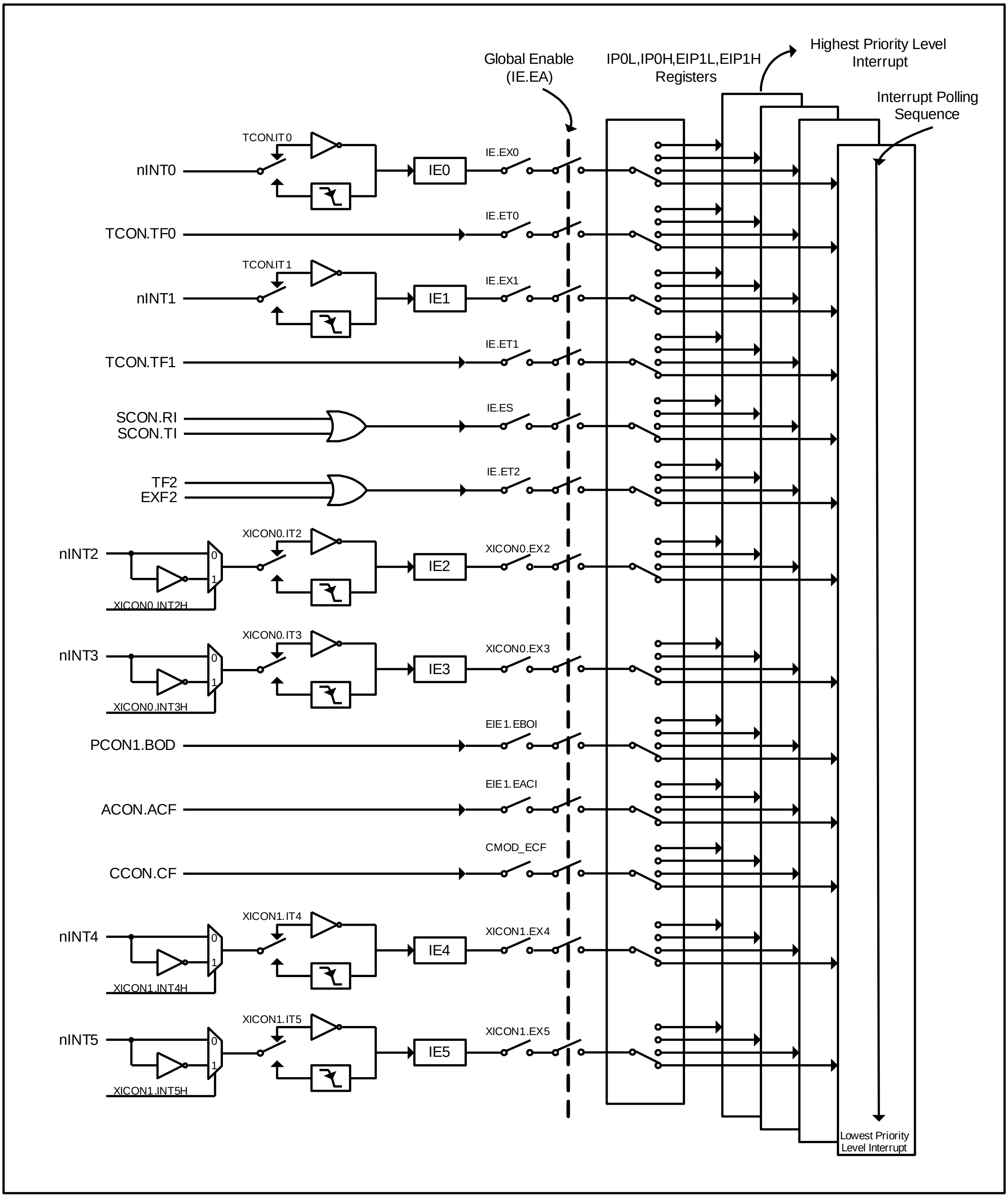
C Code Example:

```
#define       P1Mn0              0x01

      P1M0 |= P1Mn0;
      P1M1 |= P1Mn0;                                // Configure P1.0 to open drain mode
      P10 = 1;                                        // Set P1.0 data latch to "1" to enable open drain mode
```

10. Interrupt

10.1. Interrupt Structure



10.2. Interrupt Register

IE: Interrupt Enable Register
SFR Address = 0xE8
SFR Page = All Reset Value = 0X00-0000

7	6	5	4	3	2	1	0
EA	--	ET2	ES	ET1	EX1	ET0	EX0
R/W	**R**	**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.

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

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

XIFLG: External Interrupt Flag Register
SFR Address = 0xC0
SFR Page = All Reset Value = XXXX-0000

7	6	5	4	3	2	1	0
--	--	--	--	IE5	IE4	IE3	IE2
R	**R**	**R**	**R**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~4: Reserved.

Bit 3: IE5, External interrupt 5 Edge flag.
0: Cleared by hardware when the interrupt is starting to be serviced. It also could be cleared by CPU.
1: Set by hardware when external interrupt edge detected. It also could be set by CPU.

Bit 2: IE4, External interrupt 4 Edge flag.
0: Cleared by hardware when the interrupt is starting to be serviced. It also could be cleared by CPU.
1: Set by hardware when external interrupt edge detected. It also could be set by CPU.

Bit 1: IE3, External interrupt 3 Edge flag.
0: Cleared by hardware when the interrupt is starting to be serviced. It also could be cleared by CPU.
1: Set by hardware when external interrupt edge detected. It also could be set by CPU.

Bit 0: IE2, External interrupt 2 Edge flag.
0: Cleared by hardware when the interrupt is starting to be serviced. It also could be cleared by CPU.
1: Set by hardware when external interrupt edge detected. It also could be set by CPU.

XICON0: External Interrupt Control 0 Register
SFR Address = 0xC1
SFR Page = All Reset Value = X000-X000

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

Bit 7: Reserved.

Bit 6: INT3H, nINT3 High/Rising trigger enable.
0: Maintain nINT3 triggered on low level or falling edge on P4.2.
1: Set nINT3 triggered on high level or rising edge on P4.2.

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

Bit 4: EX3, external interrupt 3 enable register.
0: Disable external interrupt 3.
1: Enable external interrupt 3.

Bit 3: Reserved.

Bit 2: INT2H, nINT2 High/Rising trigger enable.
0: Maintain nINT2 triggered on low level or falling edge on P4.3.
1: Set nINT2 triggered on high level or rising edge on P4.3.

Bit 1: IT2, Interrupt 2 type control bit.
0: Cleared by CPU to specify low level triggered on /INT2. If INT2H is set, this bit specifies high level triggered on nINT2.
1: Set by CPU to specify falling edge triggered on /INT2. If INT2H is set, this bit specifies rising edge triggered on nINT2.

Bit 0: EX2, external interrupt 2 enable register.
0: Disable external interrupt 2.
1: Enable external interrupt 2.

XICON1: External Interrupt Control 1 Register
SFR Address = 0xC2
SFR Page = All Reset Value = X000-X000

7	6	5	4	3	2	1	0
--	INT5H	IT5	EX5	--	INT4H	IT4	EX4
R	**R/W**	**R/W**	**R/W**	**R**	**R/W**	**R/W**	**R/W**

Bit 7: Reserved.

Bit 6: INT5H, nINT5 High/Rising trigger enable.
0: Maintain nINT5 triggered on low level or falling edge on P5.1.
1: Set nINT5 triggered on high level or rising edge on P5.1.

Bit 5: IT5, Interrupt 5 type control bit.
0: Cleared by CPU to specify low level triggered on Interrupt 5. If INT5H is set, this bit specifies high level triggered on nINT5.

1: Set by CPU to specify falling edge triggered on Interrupt 5. If INT5H is set, this bit specifies rising edge triggered on nINT5.

Bit 4: EX5, external interrupt 5 enable register.
0: Disable external nINT5.
1: Enable external nINT5.

Bit 3: Reserved.

Bit 2: INT4H, nINT4 High/Rising trigger enable.
0: Maintain nINT4 triggered on low level or falling edge on P5.0.
1: Set INT4 triggered on high level or rising edge on P5.0.

Bit 1: IT4, Interrupt 4 type control bit.
0: Cleared by CPU to specify low level triggered on /INT4. If INT4H is set, this bit specifies high level triggered on nINT4.
1: Set by CPU to specify falling edge triggered on /INT4. If INT4H is set, this bit specifies rising edge triggered on nINT4.

Bit 0: EX4, external interrupt 4 enable register.
0: Disable external interrupt 4.
1: Enable external interrupt 4.

EIE1: Extended Interrupt Enable 1 Register

SFR Address = 0xAD
SFR Page = All Reset Value = XXXX-XX00

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

Bit 7~2: Reserved.

Bit 1: EACI, Enable Analog Comparator Interrupt.
0: Disable the interrupt when ACCON.ACF is set in Analog Comparator module.
1: Enable the interrupt when ACCON.ACF is set in Analog Comparator module.

Bit 0: EBOI, Enable BOD Interrupt.
0: Disable the interrupt when PCON1.BOD is set in power control module.
1: Enable the interrupt when PCON1.BOD is set in power control module.

IP0L: Interrupt Priority 0 Low Register

SFR Address = 0xB8
SFR Page = All Reset Value = 0000-0000

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

Bit 7: PX3L, external interrupt 3 priority-L register.
Bit 6: PX2L, external interrupt 2 priority-L register.
Bit 5: PT2L, Timer 2 interrupt priority-L register.
Bit 4: PSL, Serial port interrupt priority-L register.
Bit 3: PT1L, Timer 1 interrupt priority-L register.
Bit 2: PX1L, external interrupt 1 priority-L register.
Bit 1: PT0L, Timer 0 interrupt priority-L register.
Bit 0: PX0L, external interrupt 0 priority-L register.

IP0H: Interrupt Priority 0 High Register

SFR Address = 0xB7
SFR Page = All Reset Value = 0000-0000

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

Bit 7: PX3H, external interrupt 3 priority-H register.
Bit 6: PX2H, external interrupt 2 priority-H register.
Bit 5: PT2H, Timer 2 interrupt priority-H register.
Bit 4: PSH, Serial port interrupt priority-H register.
Bit 3: PT1H, Timer 1 interrupt priority-H register.
Bit 2: PX1H, external interrupt 1 priority-H register.
Bit 1: PT0H, Timer 0 interrupt priority-H register.
Bit 0: PX0H, external interrupt 0 priority-H register.

EIP1L: Extended Interrupt Priority 1 Low Register

SFR Address = 0xAE
SFR Page = All Reset Value = XXX0-0000

7	6	5	4	3	2	1	0
--	--	--	PX5L	PX4L	PPTL	PACL	PBOL
R	**R**	**R**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~5: Reserved.
Bit 4: PX5L, external interrupt 5 priority-L register.
Bit 3: PX4L, external interrupt 4 priority-L register.
Bit 2: PPTL, PWM-Timer interrupt priority-L register.
Bit 1: PACL, Analog Comparator interrupt priority-L register.
Bit 0: PBOL, BOD interrupt 0 priority-L register.

EIP1H: Extended Interrupt Priority 1 High Register

SFR Address = 0xAF
SFR Page = All Reset Value = XXX0-0000

7	6	5	4	3	2	1	0
--	--	--	PX5H	PX4H	PPTH	PACH	PBOH
R	**R**	**R**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~5: Reserved.
Bit 4: PX5H, external interrupt 5 priority-H register.
Bit 3: PX4H, external interrupt 4 priority-H register.
Bit 2: PPTH, PWM-Timer interrupt priority-H register.
Bit 1: PACH, Analog Comparator interrupt priority-H register.
Bit 0: PBOH, BOD interrupt 0 priority-H register.

IP0L, IP0H, EIP1L and EIP1H are combined to 4-level priority interrupt as the following table.

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

There are 13 interrupt sources available in MG82FE(L)308/316. Each interrupt source can be individually enabled or disabled by setting or clearing a bit in the SFRs named IE, EIE1, XICON0 and XICON1. This register also contains a global disable bit(EA), which can be cleared to disable all interrupts at once.

Each interrupt source has two corresponding bits to represent its priority. One is located in SFR named IPxH and the other in IPxL 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. The following table shows the internal polling sequence in the same priority level and the interrupt vector address.

Source	Vector address	Priority within level
External interrupt 0	0003H	1 (highest)
Timer 0	000BH	2
External interrupt 1	0013H	3
Timer1	001BH	4
Serial Port	0023H	5
Timer2	002BH	6
External interrupt 2	0033H	7
External interrupt 3	003BH	8
BOD	0043H	9
Analog Comparator	004BH	10
PWM Timer	0053H	11
External interrupt 4	005BH	12
External interrupt 5	0063H	13

The external interrupt /INT0, /INT1, /INT2, /INT3, /INT4 and INT5 can each be either level-activated or transition-activated, depending on bits IT0 and IT1 in register TCON, IT2 and IT3 in register XICON0, IT4 and IT5 in XICON1. The flags that actually generate these interrupts are bits IE0 and IE1 in TCON, IE2, IE3, IE4 and IE5 in XIFLG. 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 interrupt is generated by the logical OR of RI and TI. Neither of these flags is cleared by hardware when the service routine is vectored to. The service routine should poll RI and TI to determine which one to request service and it will be cleared by software.

The timer2 interrupt is generated by the logical OR of TF2 and EXF2. Just the same as serial port, neither of these flags is cleared by hardware when the service routine is vectored to.

BOD interrupt is generated by BOD in PCON1, which is set by on chip Brownout-Detector meets the low voltage event. It will not be cleared by hardware when the service routine is vectored to.

Analog comparator interrupt is generated by ACF in ACCON. It will not be cleared by hardware when the service routine is vectored to.

PWM-Timer interrupt is generated by CF in CCON. It will not be cleared by hardware when the service routine is vectored to.

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

How hardware see the interrupts

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

Block conditions:

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

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

The polling cycle is repeated with each clock cycle. Note that if an interrupt flag is active but not being responded to for one of the above conditions, if the flag is not still active when the blocking condition is removed, the denied interrupt will not be serviced. In other words, the fact that the interrupt flag was once active but not being responded to for one of the above conditions, if the flag is not still active when the blocking condition is removed, the denied interrupt will not be serviced. The interrupt flag was once active but not serviced is not kept in memory. Each polling cycle is new.

10.3. Interrupt Sample Code

(1). Required Function: Set INT0 wake-up MCU in power-down mode

Assembly Code Example:

```
PX0                 EQU                 01h
PX0H                      EQU                 01h
PD                  EQU                 02h

     ORG       0000h
     JMP      main

     ORG       00003h
ext_int0_isr:
     to do.....
     RETI

main:

     SETB      P3.2                                  ;

     ORL       IP,#PX0                              ; Select INT0 interrupt priority
     ORL       IPH,#PX0H                            ;

     JB        P3.2, $                              ; Confirm P3.2 input low?????

     SETB      EX0                                   ; Enable INT0 interrupt
     CLR       IE0                                   ; Clear INT0 flag
     SETB      EA                                    ; Enable global interrupt

     ORL       PCON,#PD                             ; Set MCU into Power Down mode

     JMP       $
```

C Code Example:

```
#define      PX0                   0x01
#define      PX0H                         0x01
#define      PD                    0x02

void ext_int0_isr(void) interrupt 0
{
     To do……
}

void main(void)
{
     P32 = 1;

     IP |= PX0;                                      // Select INT0 interrupt priority
     IPH |= PX0H;

     while(P32);                                     // Confirm P3.2 input low??????

     EX0 = 1;                                        // Enable INT0 interrupt
     IE0 = 0;                                        // Clear INT0 flag
     EA = 1;                                         // Enable global interrupt

     PCON |= PD;                                    // Set MCU into Power Down mode

     while(1);
}
```

11. Timers/Counters

MG82FE(L)308/316 has four Timers/Counters: Timer 0, Timer 1, Timer 2 and PWM Timer. Timer0/1/2 can be configured as timers or event counters. PWM Timer can be configured as timer or PWM generator.

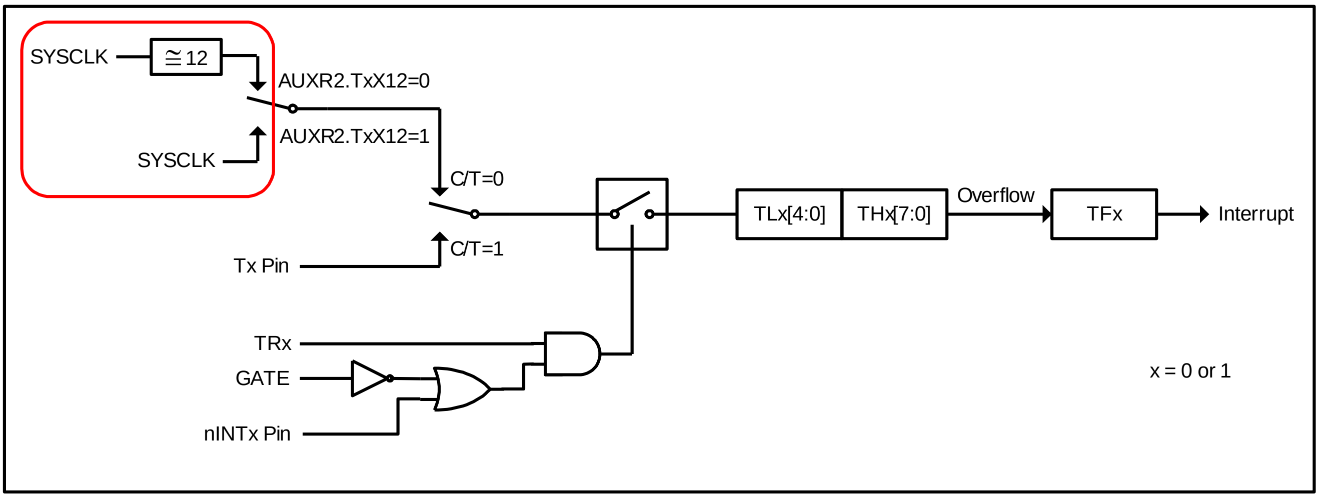
In the “timer” function, the timer rate is prescaled by 12 clock cycle to increment register value. In other words, it is to count the standard C51 machine cycle. AUXR2.T0X12, AUXR2.T1X12 and T2MOD.T2X12 are the function for Timer 0/1/2 to set the timer rate on every clock cycle.

In the “counter” function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T0, T1 or T2. In this function, the external input is sampled by every timer rate cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register at the end of the cycle following the one in which the transition was detected.

11.1. Timer0 and Timer1

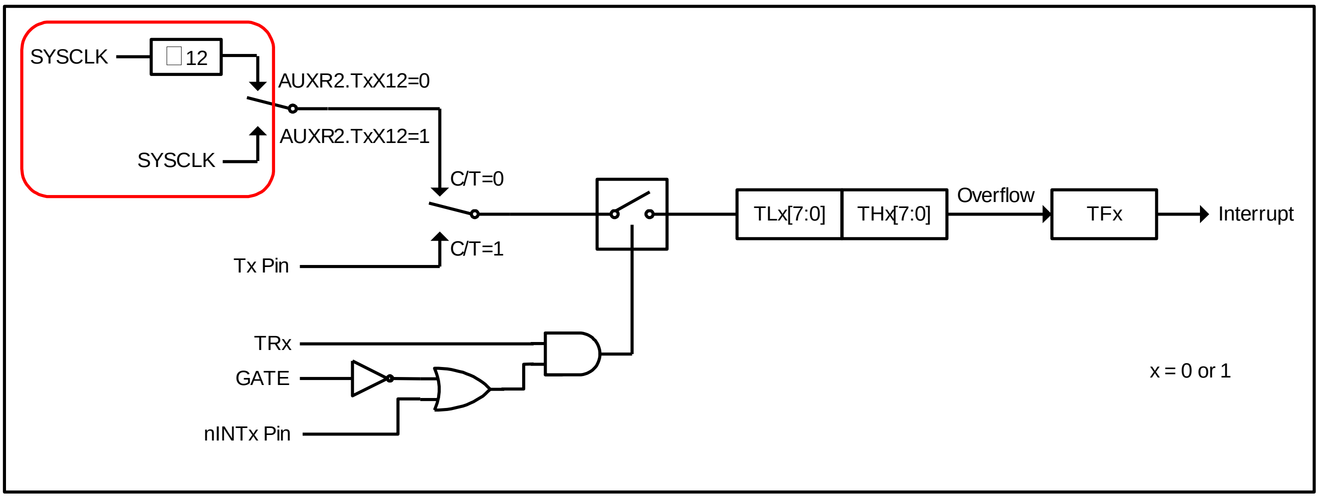
11.1.1. Mode 0 Structure

The timer register is configured as a 13-bit register. 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 INTx = 1. Mode 0 operation is the same for Timer0 and Timer1.



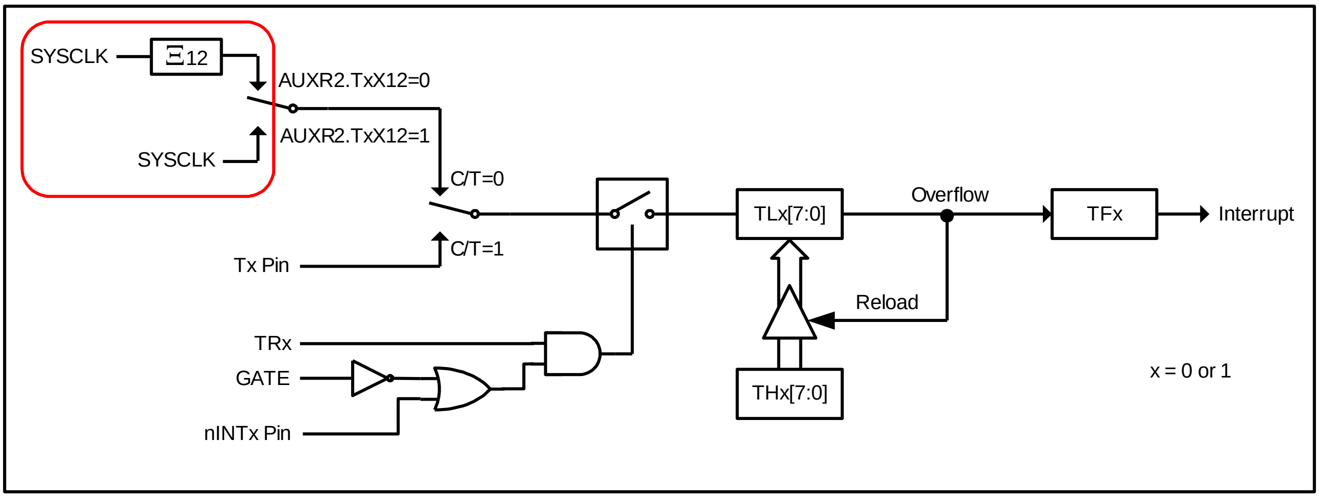
11.1.2. Mode 1 Structure

Mode1 is the same as Mode0, except that the timer register is being run with all 16 bits.



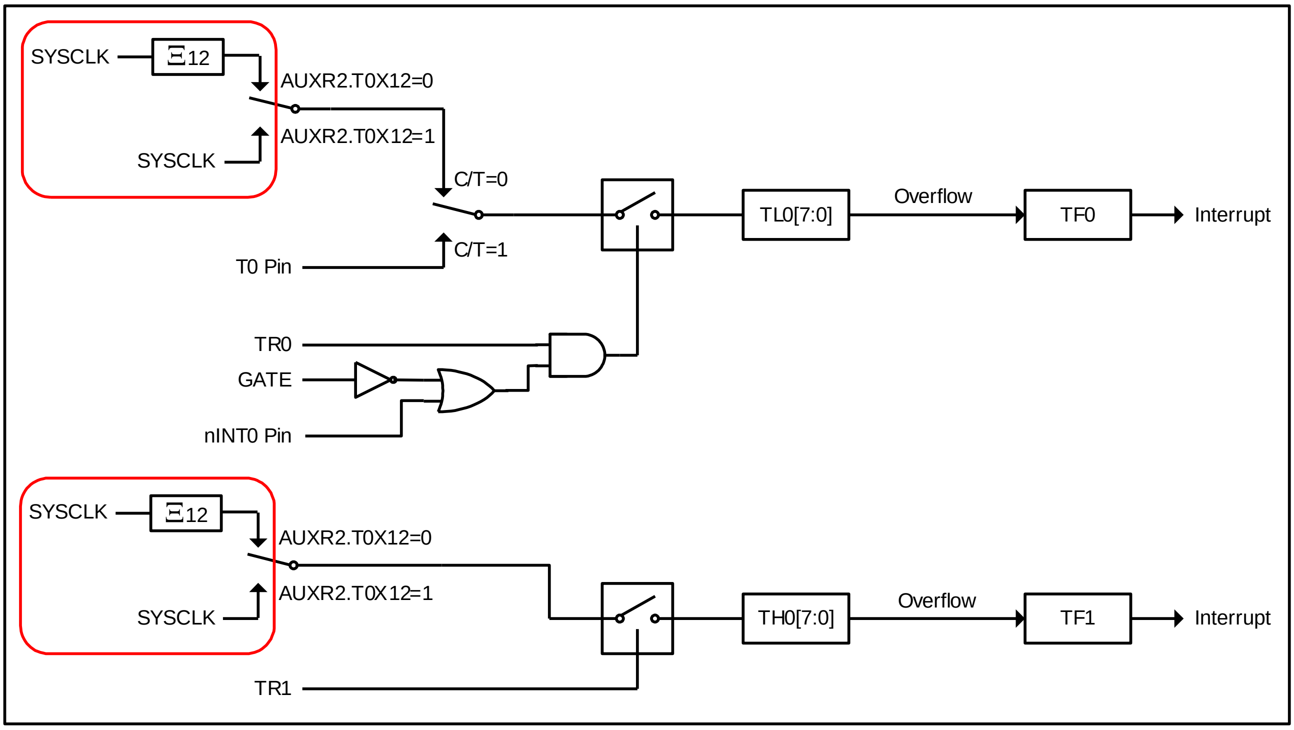
11.1.3. Mode 2 Structure

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.

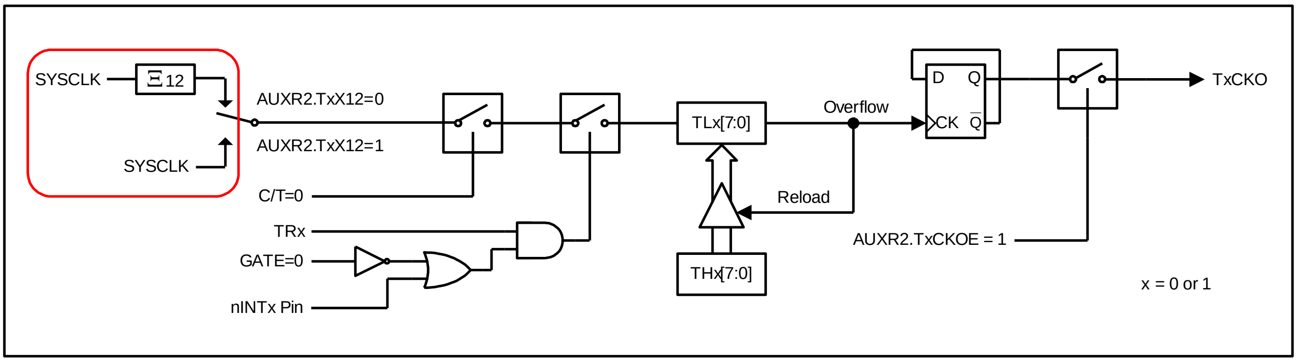


11.1.4. Mode 3 Structure

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, INT0 and TF0. TH0 is locked into a timer function (can not be external event counter) and take over the use of TR1, TF1 from Timer1. TH0 now controls the Timer1 interrupt.



11.1.5. Timer Clock-Out Structure



$$\text{T0/T1 Clock-out Frequency} = \frac{\text{SYSCLK Frequency}}{\text{n X (256 – THx)}}$$

; n=24, if TxX12=0
; n=2, if TxX12=1
; x = 0 or 1 & C/T = 0

11.1.6. Timer0/1 Register

TMOD: Timer/Counter Mode Control Register

SFR Address = 0x89
SFR Page = All Reset Value = 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	13-bit timer/counter 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

TCON: Timer/Counter Control Register

SFR Address = 0x88
SFR Page = All Reset Value = 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.

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.
1: Set by software to specify falling edge triggered external interrupt 1.

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.
1: Set by software to specify falling edge triggered external interrupt 0.

TL0: Timer Low 0 Register
SFR Address = 0x8A
SFR Page = All Reset Value = 0000-0000

7	6	5	4	3	2	1	0
TL0[7]	TL0[6]	TL0[5]	TL0[4]	TL0[3]	TL0[2]	TL0[1]	TL0[0]
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

TH0: Timer High 0 Register
SFR Address = 0x8C
SFR Page = All Reset Value = 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 Address = 0x8B
SFR Page = All Reset Value = 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 Address = 0x8D
SFR Page = All Reset Value = 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 Address = 0x87
SFR Page = All Reset Value = 000X-XX00

7	6	5	4	3	2	1	0
T0X12	T1X12	URM0X6	--	--	--	T1CKOE	T0CKOE
R/W	**R/W**	**R/W**	**R**	**R**	**R**	**R/W**	**R/W**

Bit 7: T0X12, Timer 1 clock source selector while C/T=0.
0: Clear to select SYSCLK/12.
1: Set to select SYSCLK as the clock source.

Bit 6: 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 1: T1CKOE, Timer 1 Clock Output Enable.
0: Disable Timer 1 clock output.
1: Enable Timer 1 clock output on P3.5.

Bit 0: T0CKOE, Timer 0 Clock Output Enable.
0: Disable Timer 0 clock output.
1: Enable Timer 0 clock output on P3.4.

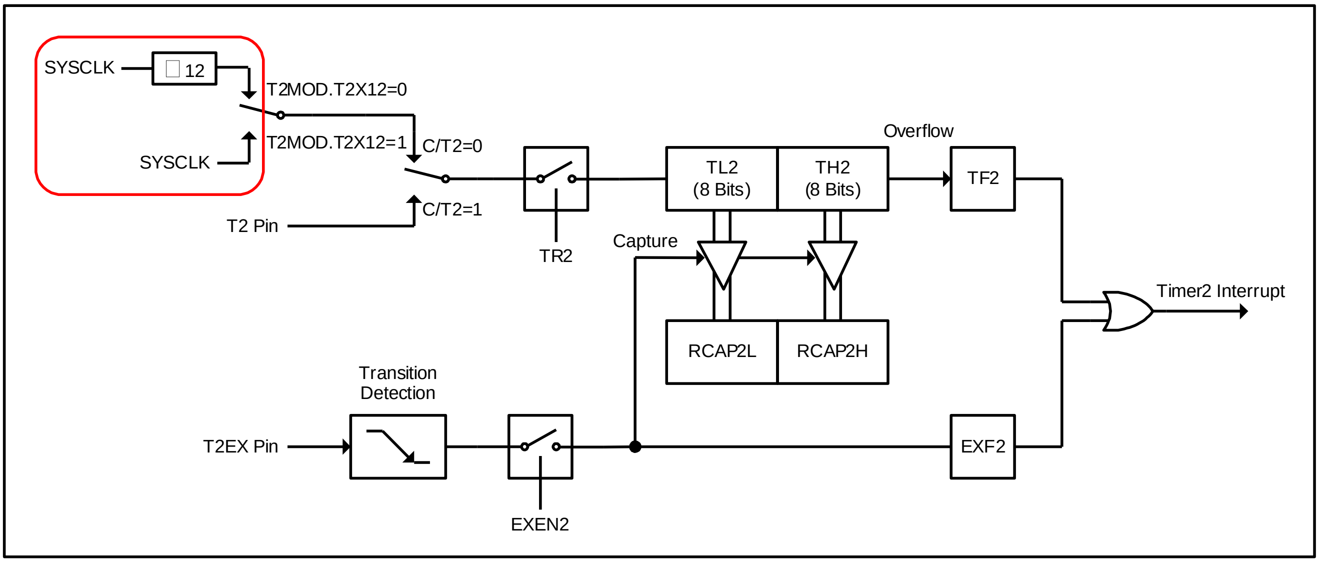
11.2. Timer2

Timer 2 is a 16-bit Timer/Counter which can operate either as a timer or an event counter, as selected by C/T2 in T2CON register. Timer 2 has four operating modes: Capture, Auto-Reload (up or down counting), Baud Rate Generator and Programmable Clock-Out, which are selected by bits in the T2CON and T2MOD registers.

11.2.1. Capture Mode (CP) Structure

In the capture mode there are two options selected by bit EXEN2 in T2CON. If EXEN2=0, Timer 2 is a 16-bit timer or counter which, upon overflow, sets bit TF2 (Timer 2 overflow flag). This bit can then be used to generate an interrupt (by enabling the Timer 2 interrupt bit in the IE register). If EXEN2=1, Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input T2EX causes the current value in the Timer 2 registers, TH2 and TL2, to be captured into registers RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set, and the EXF2 bit (like TF2) can generate an interrupt which vectors to the same location as Timer 2 overflow interrupt. The capture mode is illustrated in Figure 12-5.

Figure 12-5 Timer 2 in Capture Mode



11.2.2. Auto-Reload Mode (AR) Structure

Figure 12-6 shows DCEN=0, which enables Timer 2 to count up automatically. In this mode there are two options selected by bit EXEN2 in T2CON register. If EXEN2=0, then Timer 2 counts up to 0FFFFH and sets the TF2 (Overflow Flag) bit upon overflow. This causes the Timer 2 registers to be reloaded with the 16-bit value in RCAP2L and RCAP2H. The values in RCAP2L and RCAP2H are preset by firmware. If EXEN2=1, then a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at input T2EX. This transition also sets the EXF2 bit. The Timer 2 interrupt, if enabled, can be generated when either TF2 or EXF2 are 1.

Figure 12-6 Timer 2 in Auto-Reload Mode (DCEN=0)

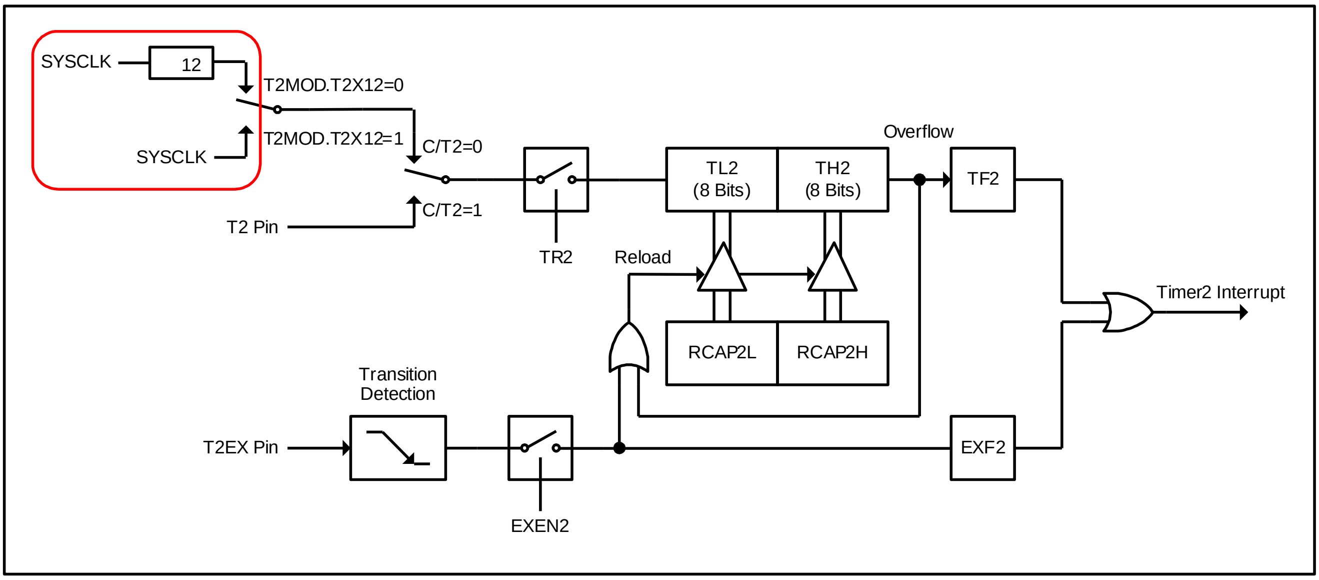
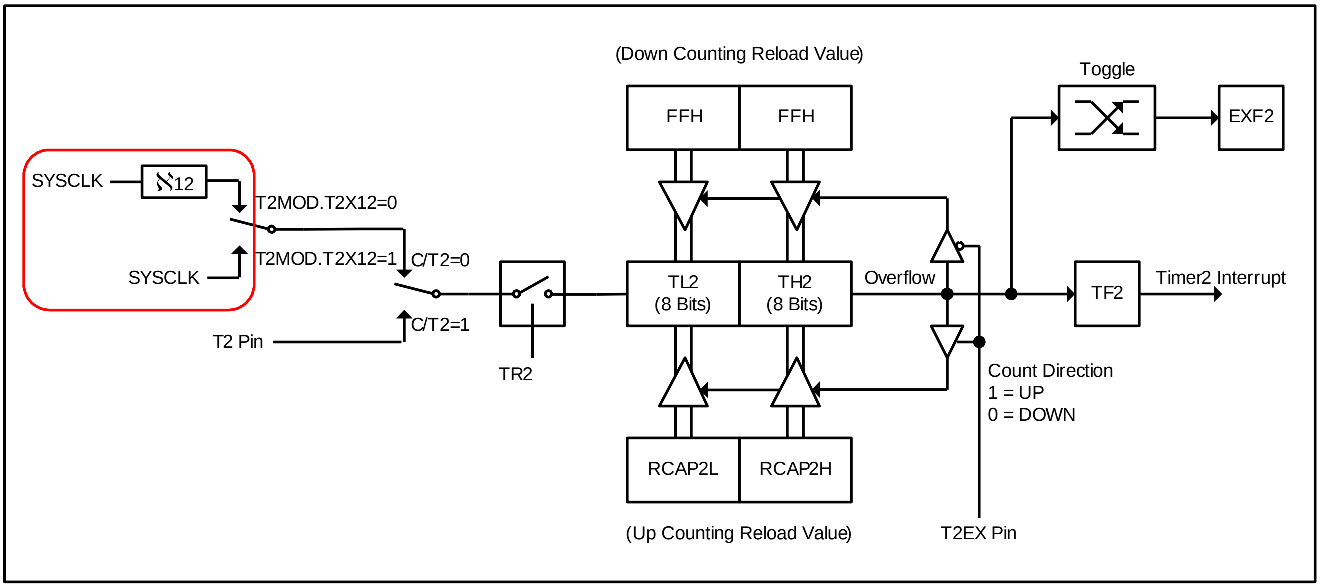


Fig 12-7 shows DCEN=1, which enables Timer 2 to count up or down. This mode allows pin T2EX to control the counting direction. When a logic 1 is applied at pin T2EX, Timer 2 will count up. Timer 2 will overflow at 0FFFFH and set the TF2 flag, which can then generate an interrupt if the interrupt is enabled. This overflow also causes the 16-bit value in RCAP2L and RCAP2H to be reloaded into the timer registers TL2 and TH2. A logic 0 applied to pin T2EX causes Timer 2 to count down. The timer will underflow when TL2 and TH2 become equal to the value stored in RCAP2L and RCAP2H. This underflow sets the TF2 flag and causes 0FFFFH to be reloaded into the timer registers TL2 and TH2.

The external flag EXF2 toggles when Timer 2 underflows or overflows. This EXF2 bit can be used as a 17th bit of resolution if needed. The EXF2 flag does not generate an interrupt in this mode.

Fig 12-7 Timer 2 in Auto-Reload Mode (DCEN=1)



11.2.3. Baud-Rate Generator Mode (BRG) Structure

Bits TCLK and/or RCLK in T2CON register allow the serial port transmit and receive baud rates to be derived from either Timer 1 or Timer 2. When TCLK=0, Timer 1 is used as the serial port transmit baud rate generator. When TCLK= 1, Timer 2 is used as the serial port transmit baud rate generator. RCLK has the same effect for the serial port receive baud rate. With these two bits, the serial port can have different receive and transmit baud rates – one generated by Timer 1, the other by Timer 2.

Fig 12-8 shows the Timer 2 in baud rate generation mode to generate RX Clock and TX Clock into UART engine (See Fig 12 6). The baud rate generation mode is like the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by firmware.

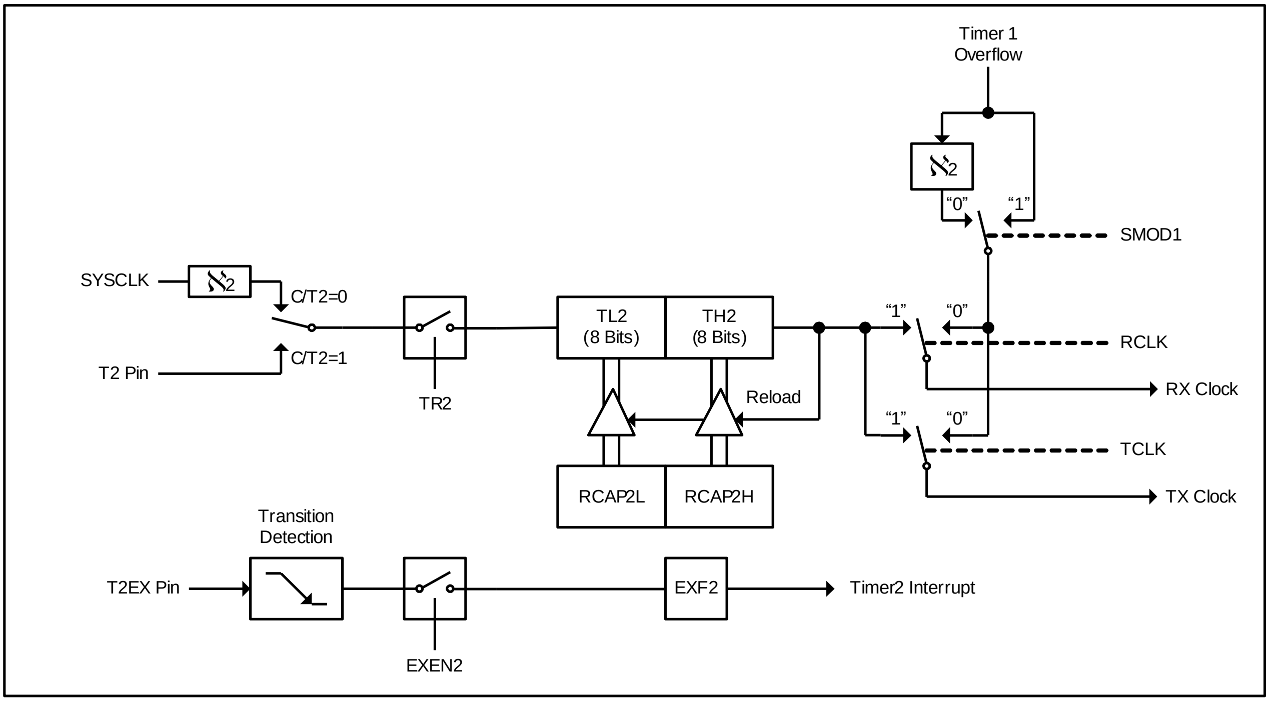
The Timer 2 as a baud rate generator mode is valid only if RCLK and/or TCLK=1 in T2CON register. Note that a rollover in TH2 does not set TF2, and will not generate an interrupt. Thus, the Timer 2 interrupt does not have to be disabled when Timer 2 is in the baud rate generator mode. Also if the EXEN2 (T2 external enable bit) is set, a 1-to-0 transition in T2EX (Timer/counter 2 trigger input) will set EXF2 (T2 external flag) but will not cause a reload from (RCAP2H, RCAP2L) to (TH2,TL2). Therefore when Timer 2 is in use as a baud rate generator, T2EX can be used as an additional external interrupt, if needed.

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

Note:

Refer to 13.7.3 Baud Rate in Mode 1 & 3 to get baud rate setting value when using Timer 2 as the baud rate generator.

Fig 12-8 Timer 2 in Baud-Rate Generator Mode



11.2.4. Programmable Clock Output from Timer 2 Structure

Timer 2 has a Clock-Out Mode (while CP/RL2=0 & T2OE=1). In this mode, Timer 2 operates as a programmable clock generator with 50% duty-cycle. The generated clocks come out on P1.0. The input clock, SYSCLK/2, increments the 16-bit timer (TH2, TL2). The timer repeatedly counts to overflow from a loaded value. Once overflows occur, the contents of (RCAP2H, RCAP2L) are loaded into (TH2, TL2) for the consecutive counting. The following formula gives the clock-out frequency:

$$\text{T2 Clock-out Frequency} = \frac{\text{SYSCLK Frequency}}{4 \times (65536 - (\text{RCAP2H, RCAP2L}))}$$

Note:
(1) Timer 2 overflow flag, TF2, will always not be set in this mode.
(2) For SYSCLK=12MHz, Timer 2 has a programmable output frequency range from 45.7Hz to 3MHz.

How to Program Timer 2 in Clock-out Mode

- Set T2OE bit in T2MOD register.
- Clear C/T2 bit in T2CON register.
- Determine the 16-bit reload value from the formula and enter it in the RCAP2H and RCAP2L registers.
- Enter the same reload value as the initial value in the TH2 and TL2 registers.
- Set TR2 bit in T2CON register to start the Timer 2.

In the Clock-Out mode, Timer 2 rollovers will not generate an interrupt. This is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud rate generator and a clock generator simultaneously. Note, however, that the baud-rate and the clock-out frequency depend on the same overflow rate of Timer 2.

11.2.5. Timer2 Register

T2MOD: Timer/Counter 2 Mode Control Register
SFR Address = 0xC9
SFR Page = All Reset Value= XXX0-XX00

7	6	5	4	3	2	1	0
--	--	--	T2X12	--	--	T2OE	DCEN
R	**R**	**R**	**R/W**	**R**	**R**	**R/W**	**R/W**

Bit 7~5: Reserved.

Bit 4: T2X12, Timer 2 clock source selector.
0: Select SYSCLK/12 as Timer 2 clock source while T2CON.C/T2 = 0 in Capture Mode and Auto-Reload Mode.
1: Select SYSCLK as Timer 2 clock source while T2CON.C/T2 = 0 in Capture Mode and Auto-Reload

Bit 3~2: Reserved.

Bit 1: T2OE, Timer 2 clock-out enable bit.
0: Disable Timer 2 clock output.
1: Enable Timer 2 clock output.

Bit 0: DCEN, Timer 2 down-counting enable bit.
0: Timer 2 always keeps up-counting.
1: Enable Timer 2 down-counting ability.

T2CON: Timer/Counter 2 Mode Control Register
SFR Address = 0xC8
SFR Page = **0** Reset Value = 0000-0000

7	6	5	4	3	2	1	0

TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: TF2, Timer 2 overflow flag.
0: TF2 must be cleared by software.
1: TF2 is set by a Timer 2 overflow happens. TF2 will not be set when either RCLK=1 or TCLK=1.

Bit 6: EXF2, Timer 2 external flag.
0: EXF2 must be cleared by software.
1: Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX pin and EXEN2=1. When Timer 2 interrupt is enabled, EXF2=1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 does not cause an interrupt in up/down mode (DCEN = 1).

Bit 5: RCLK, Receive clock flag.
0: Causes Timer 1 overflow to be used for the receive clock.
1: Causes the serial port to use Timer 2 overflow pulses for its receive clock in modes 1 and 3.

Bit 4: TCLK, Transmit clock flag.
0: Causes Timer 1 overflows to be used for the transmit clock.
1: Causes the serial port to use Timer 2 overflow pulses for its transmit clock in modes 1 and 3.

Bit 3: EXEN2, Timer 2 external enable flag.
0: Cause Timer 2 to ignore events at T2EX pin.
1: Allows a capture or reload to occur as a result of a negative transition on T2EX pin if Timer 2 is not being used to clock the serial port.

Bit 2: TR2, Timer 2 Run control bit.
0: Stop the Timer 2.
1: Start the Timer 2.

Bit 1: C/T2, Timer or counter selector.
0: Select Timer 2 as internal timer function.
1: Select Timer 2 as external event counter (falling edge triggered).

Bit 0: CP/-RL2, Capture/Reload flag.
0: Auto-reloads will occur either with Timer 2 overflows or negative transitions at T2EX pin when EXEN2=1.
1: Captures will occur on negative transitions at T2EX pin if EXEN2=1.
When either RCLK=1 or TCLK=1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

When the DCEN is cleared, which makes the function of Timer 2 as the same as the standard 8052 (always counts up). When DCEN is set, Timer 2 can count up or count down according to the logic level of the T2EX pin (P1.1). The following Table shows the operation modes of Timer 2.

RCLK + TCLK	CP/-RL2	TR2	DCEN	T2OE	Mode
x	x	0	x	0	(off)
1	x	1	0	0	Baud-rate generator
0	1	1	0	0	16-bit capture
0	0	1	0	0	16-bit auto-reload (counting-up only)
0	0	1	**1**	0	16-bit auto-reload (counting-up or counting-down)
0	0	1	0	**1**	Clock output

TL2: Timer Low 2 Register
SFR Address = 0xCC
SFR Page = All Reset Value = 0000-0000

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

TH2: Timer High 2 Register

SFR Address = 0xCD
SFR Page = All Reset Value = 0000-0000

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

RCAP2L: Timer 2 Capture Low Register

SFR Address = 0xCA
SFR Page = All Reset Value = 0000-0000

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

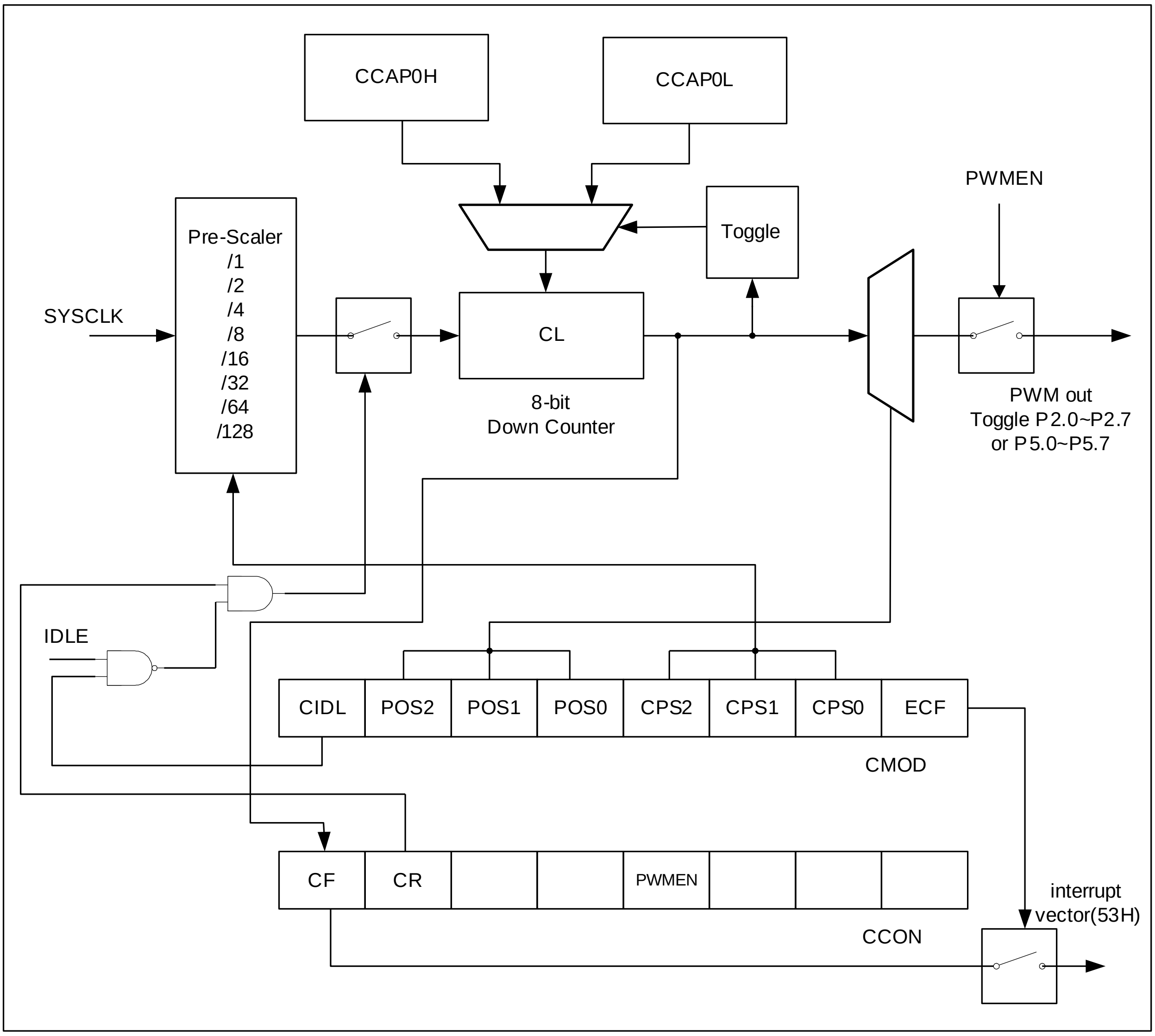
RCAP2H: Timer 2 Capture High Register

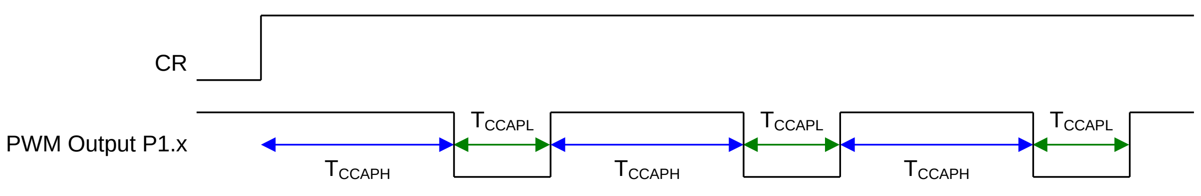
SFR Address = 0xCB
SFR Page = All Reset Value = 0000-0000

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

11.3. PWM Timer

11.3.1. PWM Timer Structure





T_{CCAPH} = (CCAP0H + 1) x T_{SYSTEM_CLOCK} x Pre-Scaler

T_{CCAPL} = (CCAP0L + 1) x T_{SYSTEM_CLOCK} x Pre-Scaler

11.3.2. PWM Timer Register

CMOD: PWM timer Mode Register

SFR Address = 0xD9

SFR Page = All Reset Value = 0000-0000

7	6	5	4	3	2	1	0

CIDL	POS2	POS1	POS0	CPS2	CPS1	CPS0	ECF
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: CIDL, Counter Idle Control.
0: Program the PWM Timer to continue functioning during IDLE mode.
1: Program the PWM Timer to be gated off during IDLE mode.

Bit 6~4: POS[2:0], PWM output port select.

POS[2:0]	PWMEN	PWM Output Port (AUXR1.P5PWM=0)	PWM Output Port (AUXR1.P5PWM=1)
0 0 0	1	P2.0	P5.0
0 0 1	1	P2.1	P5.1
0 1 0	1	P2.2	P5.2
0 1 1	1	P2.3	P5.3
1 0 0	1	P2.4	P5.4
1 0 1	1	P2.5	P5.5
1 1 0	1	P2.6	P5.6
1 1 1	1	P2.7	P5.7
X X X	0	Disabled	Disabled

Bit 3~1: CPS[2:0], Counter Prescaler Select.

CPS[2:0]	Prescaler
0 0 0	1
0 0 1	2
0 1 0	4
0 1 1	8
1 0 0	16
1 0 1	32
1 1 0	64
1 1 1	128

Bit 0: ECF, Enable PWM Timer underflow interrupt.
0: Disables CF bit in CCON to generate an interrupt.
1: Enables CF bit in CCON to generate an interrupt.

CCON: PWM timer Control Register
SFR Address = 0xD8
SFR Page = **0** Reset Value = 00XX-0XXX

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

Bit 7: CF, PWM timer underflow Flag.
0: This flag can only be cleared by software.
1: Set by hardware when the counter rolls under. CF flags an interrupt if bit ECF in CMOD is set. CF may be set by either hardware or software.

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

Bit 5~4: Reserved.

Bit 3: PWMEN, PWM output Enable.
0: Disable PWM timer under flow to toggle the I/O port.
1: Enable PWM timer under flow to toggle the port which is indexed by CMOD.POS[2:0] and AUXR1.P5PWM.

Bit 2~0: Reserved.

CACP0L: PWM Timer L-Duty Register

SFR Address = 0xEA
SFR Page = All Reset Value = 0000-0000

7	6	5	4	3	2	1	0
CACP0L[7]	CACP0L[6]	CACP0L[5]	CACP0L[4]	CACP0L[4]	CACP0L[2]	CACP0L[1]	CACP0L[0]
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

CACP0H: PWM Timer H-Duty Register

SFR Address = 0xFA
SFR Page = All Reset Value = 0000-0000

7	6	5	4	3	2	1	0
CACP0H[7]	CACP0H[6]	CACP0H[5]	CACP0H[4]	CACP0H[4]	CACP0H[2]	CACP0H[1]	CACP0H[0]
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

11.4. Timer0/1 Sample Code

(1). Required Function: IDLE mode with T0 wake-up frequency 10KHz, SYSCLK = 12MHz Crystal

Assembly Code Example:

```
T0M0            EQU             01h
T0M1            EQU             02h
PT0        EQU             02h
PT0H            EQU             02h
IDL        EQU             01h

    ORG     0000h
    JMP     main

    ORG     0000Bh
time0_isr:
    to do…
    RETI

main:                                          ; (unsigned short value)

    MOV      TH0,#(256-100)                    ; Set Timer 0 overflow rate = SYSCLK x 100
    MOV      TL0,#(256-100)                    ;
    ANL     TMOD,#0F0h                         ; Set Timer 0 to Mode 2
    ORL     TMOD,#T0M1                         ;
    CLR     TF0                                ; Clear Timer 0 Flag

    ORL     IP,#PT0                            ; Select Timer 0 interrupt priority
    ORL     IPH,#PT0H                          ;

    SETB    ET0                                ; Enable Timer 0 interrupt
    SETB    EA                                 ; Enable global interrupt

    SETB    TR0                                ; Start Timer 0 running

    ORL     PCON,#IDL                          ; Set MCU into IDLE mode
    JMP     $
```

C Code Example:

```
#define     T0M0                0x01
#define     T0M1                0x02
#define     PT0            0x02
#define     PT0H                0x02
#define     IDL            0x01

void time0_isr(void) interrupt 1
{
    To do…
}

void main(void)
{
    TH0 = TL0 = (256-100);                  // Set Timer 0 overflow rate = SYSCLK x 100
    TMOD &= 0xF0;                           // Set Timer 0 to Mode 2
    TMOD |= T0M1;
    TF0 = 0;                                // Clear Timer 0 Flag

    IP |= PT0;                              // Select Timer 0 interrupt priority
    IPH |= PT0H;

    ET0 = 1;                                // Enable Timer 0 interrupt
    EA = 1;                                 // Enable global interrupt

    TR0 = 1;                                // Start Timer 0 running
    PCON=IDL;                               // Set MCU into IDLE mode
    while(1);
}
```

(2). Required Function: Select Timer 0 clock source from SYSCLK (enable T0X12)

Assembly Code Example:

```
T0M0            EQU             01h
T0M1            EQU             02h
PT0         EQU             02h
PT0H            EQU             02h
T0X12           EQU             80h

    ORG      0000h
    JMP     main

    ORG      0000Bh
time0_isr:
    to do…
    RETI

main:
    ORL       AUXR, #T0X12                  ; Select SYSCLK/1 for Timer 0 clock input
    CLR       TF0                           ; Clear Timer 0 Flag

    ORL       IP,#PT0                      ; Select Timer 0 interrupt priority
    ORL       IPH,#PT0H                    ;

    SETB      ET0                          ; Enable Timer 0 interrupt
    SETB      EA                           ; Enable global interrupt

    MOV        TH0, #(256 - 240)           ;interrupt interval 20us
    MOV        TL0, #(256 - 240)           ;

    ANL       TMOD,#0F0h                    ; Set Timer 0 to Mode 2
    ORL       TMOD,#T0M1                    ;


    SETB      TR0                           ; Start Timer 0 running
    JMP       $
```

C Code Example:

```
#define       T0M0              0x01
#define       T0M1              0x02
#define       PT0           0x02
#define       PT0H              0x02
#define       T0X12             0x80

    AUXR |= T0X12
    TF0 = 0;

    IP |= PT0;                              // Select Timer 0 interrupt priority
    IPH |= PT0H;

    ET0 = 1;                                // Enable Timer 0 interrupt
    EA = 1;                                 // Enable global interrupt

    TH0 = TL0 = (256 - 240);

    TMOD &= 0xF0;                             // Set Timer 0 to Mode 2
    TMOD |= T0M1;

    TR0 = 1;                                 // Start Timer 0 running
```

12. Serial Port (UART)

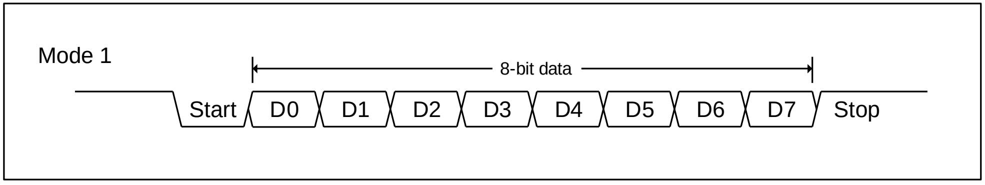
The serial port of MG82FE(L)308/316 support full-duplex transmission, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the register. However, if the first byte still hasn't been read by the time reception of the second byte is complete, one of the bytes will be lost. The serial port receive and transmit registers are both accessed at special function register SBUF. Writing to SBUF loads the transmit register, and reading from SBUF accesses a physically separate receive register.

The serial port 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 RXD(P3.0). TXD(P3.1) always outputs the shift clock. The baud rate can be selected to 1/12 or 1/2 the system clock frequency by URM0X6 setting in AUXR2 register.

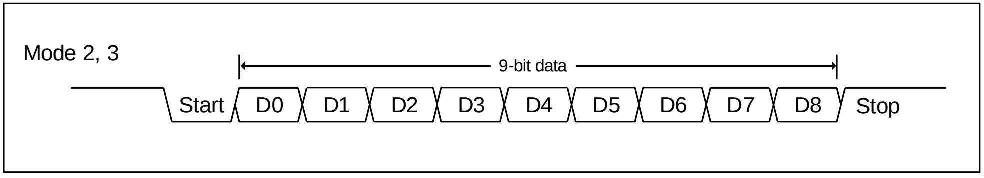
Mode 1: 10 bits are transmitted through TXD or received through RXD. The frame data includes a start bit (0), 8 data bits (LSB first), and a stop bit (1), as shown in Figure 12-1. On receive, the stop bit would be loaded into RB8 in SCON register. The baud rate is variable.

Figure 12-1 Mode 1 Data Frame



Mode 2: 11 bits are transmitted through TXD or received through RXD. 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 12-2. On Transmit, the 9th data bit comes from TB8 in SCON register can be assigned the value of 0 or 1. On receive, the 9th data bit would be loaded into RB8 in SCON register, while the stop bit is ignored. The baud rate can be configured to 1/32 or 1/64 the system clock frequency.

Figure 12-2 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 SBUF as a destination register. In Mode 0, reception is initiated by the condition RI=0 and REN=1. In the other modes, reception is initiated by the incoming start bit with 1-to-0 transition if REN=1.

In addition to the standard operation, the UART can perform framing error detection by looking for missing stop bits, and automatic address recognition.

12.1. Serial Port Mode 0

Serial data enters and exits through RXD. TXD 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 1/2 the system clock frequency by URM0X6 setting in AUXR2 register. Figure 12-3 shows a simplified functional diagram of the serial port in Mode 0.

Transmission is initiated by any instruction that uses SBUF as a destination register. The “write to SBUF” signal triggers the UART engine to start the transmission. The data in the SBUF would be shifted into the RXD(P3.0) pin by each raising edge shift clock on the TXD(P3.1) pin. After eight raising edge of shift clocks passing, TI would be asserted by hardware to indicate the end of transmission. Figure 12-4 shows the transmission waveform in Mode 0.

Reception is initiated by the condition REN=1 and RI=0. At the next instruction cycle, the Serial Port 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 RXD(P3.0) pin would be sampled and shifted into shift register by falling edge of shift clock. After eight falling edge of shift clock, RI would be asserted by hardware to indicate the end of reception. Figure 12-5 shows the reception waveform in Mode 0.

Figure 12-3 Serial Port Mode 0

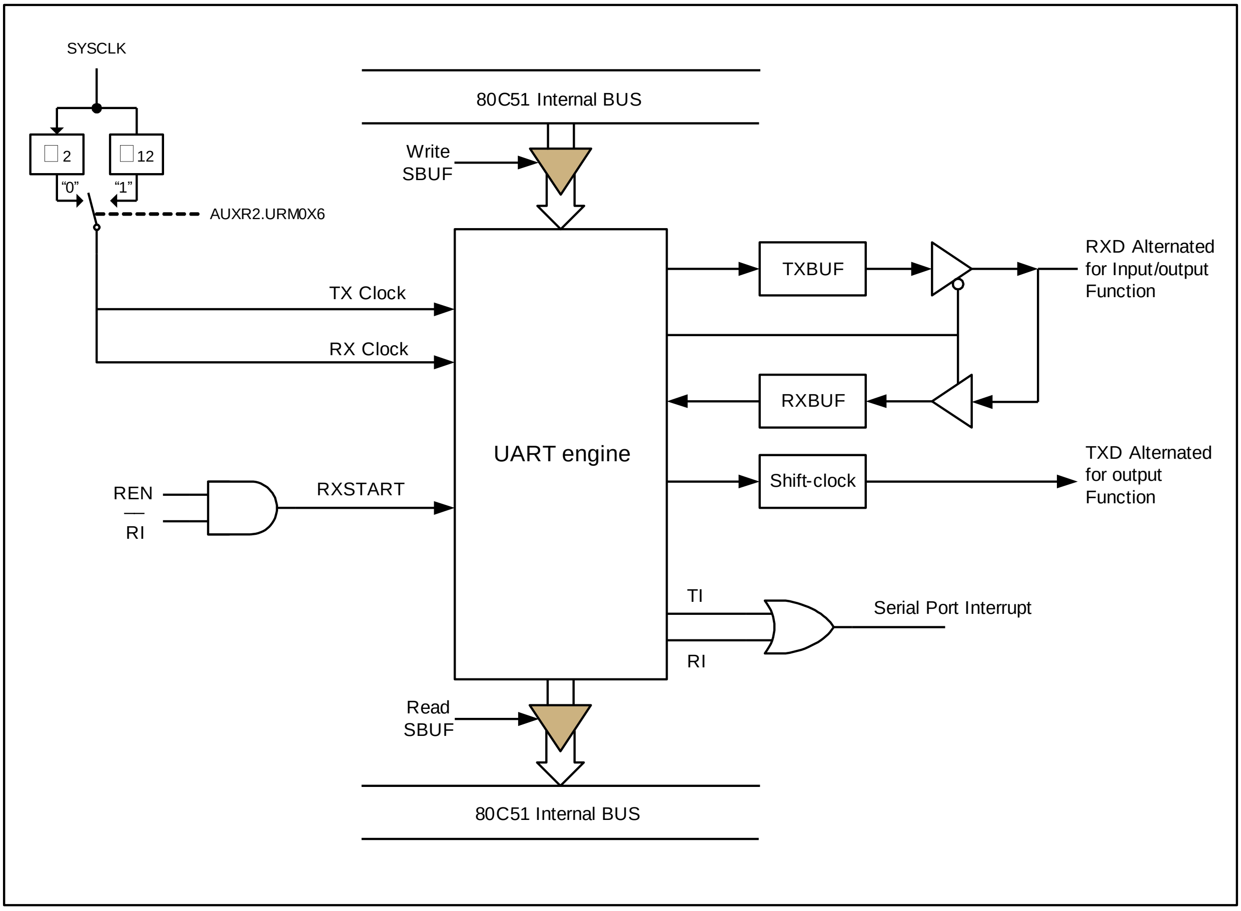


Figure 12-4 Mode 0 Transmission Waveform

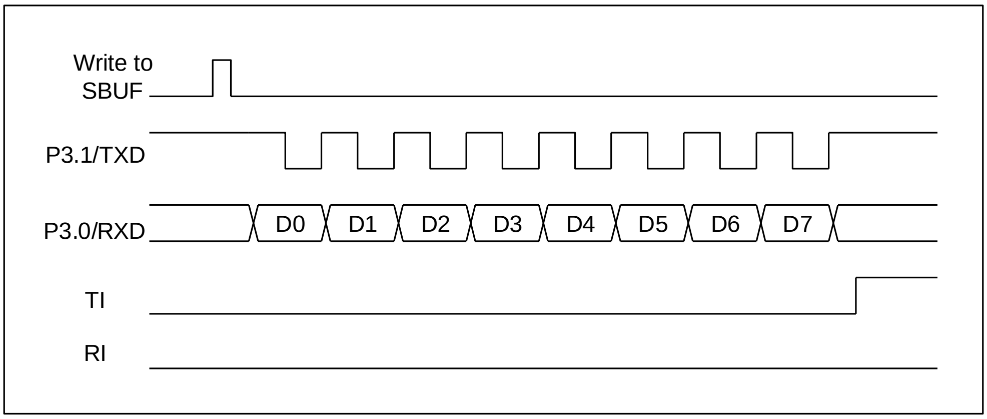
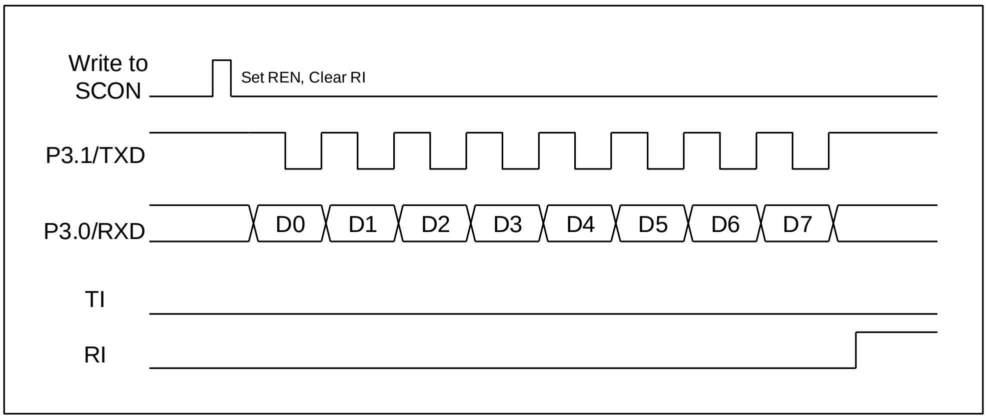


Figure 12-5 Mode 0 Reception Waveform



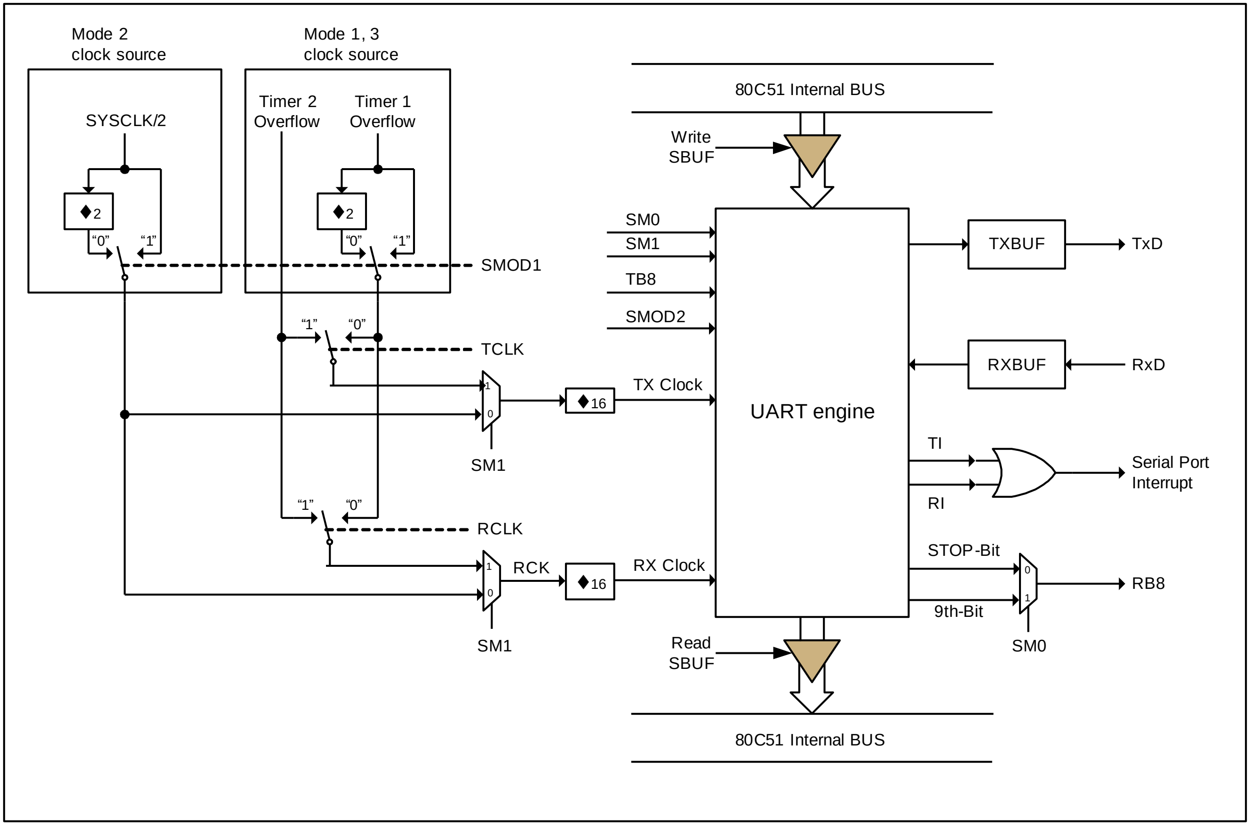
12.2. Serial Port Mode 1

10 bits are transmitted through TXD, or received through RXD: a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in SCON. The baud rate is determined by the Timer 1 or Timer 2 overflow rate. Figure 12-1 shows the data frame in Mode 1 and Figure 12-6 shows a simplified functional diagram of the serial port in Mode 1.

Transmission is initiated by any instruction that uses SBUF as a destination register. The “write to SBUF” signal requests the UART engine to start the transmission. After receiving a transmission request, the UART engine would start the transmission at the raising edge of TX Clock. The data in the SBUF would be serial output on the TXD pin with the data frame as shown in Figure 12-1 and data width depend on TX Clock. After the end of 8th data transmission, TI would be asserted by hardware to indicate the end of data transmission.

Reception is initiated when Serial Port Controller detected 1-to-0 transition at RXD sampled by RCK. The data on the RXD pin would be sampled by Bit Detector in Serial Port Controller. After the end of STOP-bit reception, RI would be asserted by hardware to indicate the end of data reception and load STOP-bit into RB8 in SCON register.

Figure 12-6 Serial Port Mode 1, 2, 3



12.3. Serial Port Mode 2 and Mode 3

11 bits are transmitted through TXD, or received through RXD: 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 (TB8) can be assigned the value of 0 or 1. On receive, the 9th data bit goes into RB8 in SCON. The baud rate is programmable to select one of 1/16, 1/32 or 1/64 the system clock frequency in Mode 2. Mode 3 may have a variable baud rate generated from Timer 1 or Timer 2.

Figure 12-2 shows the data frame in Mode 2 and Mode 3. Figure 12-6 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 SBUF” signal requests the Serial Port Controller to load TB8 into the 9th bit position of the transmit shit register and starts the transmission. After receiving a transmission request, the UART engine would start the transmission at the raising edge of TX Clock. The data in the SBUF would be serial output on the TXD pin with the data frame as shown in Figure 12-2 and data width depend on TX Clock. After the end of 9th data transmission, TI would be asserted by hardware to indicate the end of data transmission.

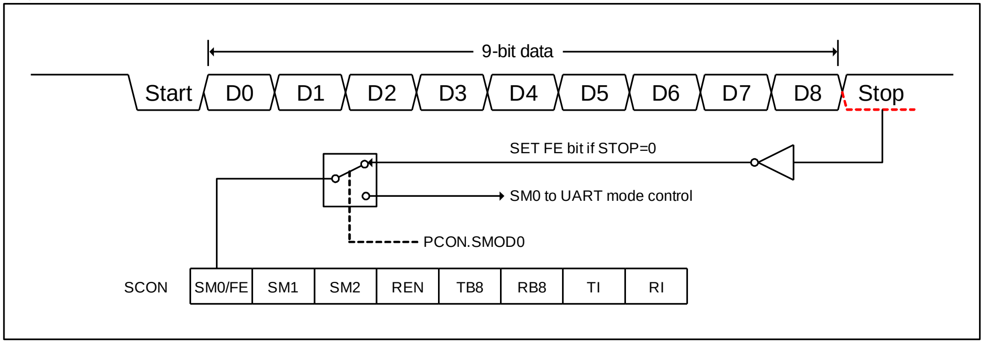
Reception is initiated when the UART engine detected 1-to-0 transition at RXD sampled by RCK. The data on the RXD pin would be sampled by Bit Detector in UART engine. After the end of 9th data bit reception, RI would be asserted by hardware to indicate the end of data reception and load the 9th data bit into RB8 in SCON register.

In all four modes, transmission is initiated by any instruction that use SBUF as a destination register. Reception is initiated in mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit with 1-to-0 transition if REN=1.

12.4. Frame Error Detection

When used for framing error detection, the UART looks for missing stop bits in the communication. A missing stop bit will set the FE bit in the SCON register. The FE bit shares the SCON.7 bit with SM0 and the function of SCON.7 is determined by SMOD0 bit (PCON.6). If SMOD0 is set then SCON.7 functions as FE. SCON.7 functions as SM0 when SMOD0 is cleared. When SCON.7 functions as FE, it can only be cleared by firmware. Refer to Figure 12-7.

Figure 12-7 UART Frame Error Detection



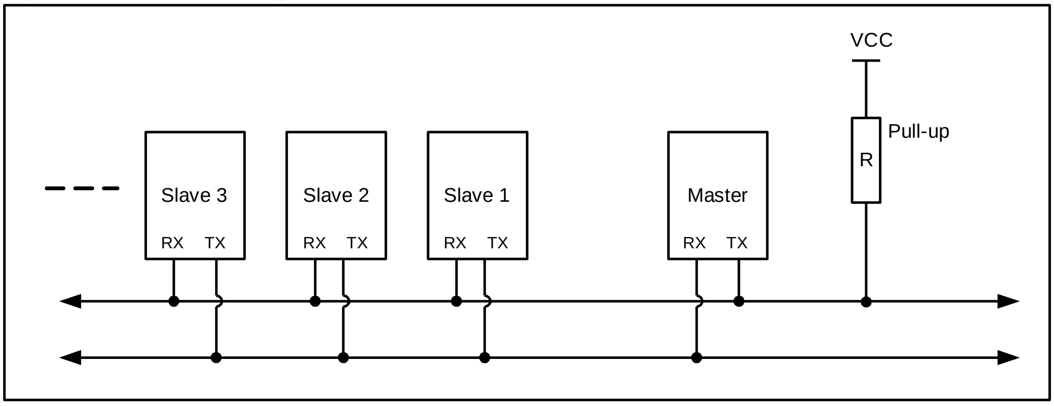
12.5. Multiprocessor Communications

Modes 2 and 3 have a special provision for multiprocessor communications as shown in Figure 12-8. In these two modes, 9 data bits are received. The 9th bit goes into RB8. Then comes a stop bit. The port can be programmed such that when the stop bit is received, the serial port interrupt will be activated only if RB8=1. This feature is enabled by setting bit SM2 (in SCON register). A way to use this feature in multiprocessor systems is as follows:

When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. With SM2=1, no slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and check if it is being addressed. The addressed slave will clear its SM2 bit and prepare to receive the data bytes that will be coming. The slaves that weren't being addressed leave their SM2 set and go on about their business, ignoring the coming data bytes.

SM2 has no effect in Mode 0, and in Mode 1 can be used to check the validity of the stop bit. In a Mode 1 reception, if SM2=1, the receive interrupt will not be activated unless a valid stop bit is received.

Figure 12-8 UART Multiprocessor Communications



12.6. Automatic Address Recognition

Automatic Address Recognition is a feature which allows the UART to recognize certain addresses in the serial bit stream by using hardware to make the comparisons. This feature saves a great deal of firmware overhead by eliminating the need for the firmware to examine every serial address which passes by the serial port. This feature is enabled by setting the SM2 bit in SCON.

In the 9 bit UART modes, mode 2 and mode 3, the Receive Interrupt flag (RI) will be automatically set when the received byte contains either the "Given" address or the "Broadcast" address. The 9-bit mode requires that the 9th information bit is a 1 to indicate that the received information is an address and not data. Automatic address recognition is shown in Figure 12-9. The 8 bit mode is called Mode 1. In this mode the RI flag will be set if SM2 is enabled and the information received has a valid stop bit following the 8 address bits and the information is either a Given or Broadcast address. Mode 0 is the Shift Register mode and SM2 is ignored.

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

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

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

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

In the above example SADDR is the same and the SADEN data is used to differentiate between the two slaves. Slave 0 requires a 0 in bit 0 and it ignores bit 1. Slave 1 requires a 0 in bit 1 and bit 0 is ignored. A unique address for Slave 0 would be 1100 0010 since slave 1 requires a 0 in bit 1. A unique address for slave 1 would be 1100 0001 since a 1 in bit 0 will exclude slave 0. Both slaves can be selected at the same time by an address which has bit 0 = 0 (for slave 0) and bit 1 = 0 (for slave 1). Thus, both could be addressed with 1100 0000.

In a more complex system the following could be used to select slaves 1 and 2 while excluding slave 0:

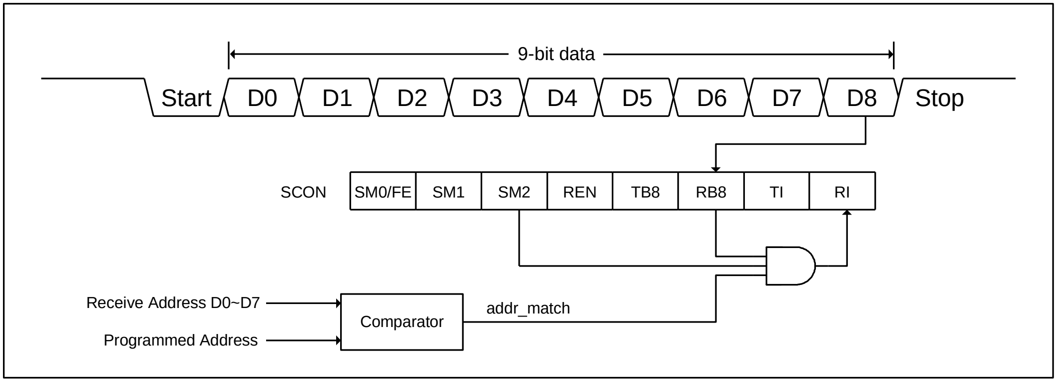
Slave 0	Slave 1	Slave 2
SADDR = 1110 0000	SADDR = 1110 0000	SADDR = 1110 0000
SADEN = 1111 1001	SADEN = 1111 1010	SADEN = 1111 1100
Given = 1110 0XX0	Given = 1110 0X0X	Given = 1110 00XX

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

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

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

Figure 12-9 Auto-Address Recognition



Note: (1) After address matching(addr_match=1), Clear SM2 to receive data bytes
(2) After all data bytes have been received, Set SM2 to wait for next address.

12.7. Baud Rate Setting

12.7.1. Baud Rate in Mode 0

$$\text{Mode 0 Baud Rate} = \frac{\text{SYSCLK Frequency}}{n} \quad \begin{array}{l}\text{; n=12, if URM0X6=0} \\ \text{; n=2, if URM0X6=1}\end{array}$$

Note:

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

12.7.2. Baud Rate in Mode 2

12.7.3. Baud Rate in Mode 1 & 3

Using Timer 1 as the Baud Rate Generator

Using Timer 2 as the Baud Rate Generator

12.8. Serial Port Register

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

SCON: Serial port Control Register

SFR Address = 0x98
SFR Page = All   Reset Value = 0000-0000

7	6	5	4	3	2	1	0
SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI
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 mode bit 0, (SMOD0 must = 0 to access bit SM0)
Bit 6: Serial port mode bit 1.

SM0	SM1	Mode	Description	Baud Rate
0	0	0	shift register	SYSCLK/12 or /2
0	1	1	8-bit UART	variable
1	0	2	9-bit UART	SYSCLK/64, /32, /16 or /8
1	1	3	9-bit UART	variable

Bit 5: Serial port mode bit 2.
0: Disable SM2 function.
1: Enable the automatic address recognition feature in Modes 2 and 3. If SM2=1, RI 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 SM2=1 then RI will not be set unless a valid stop Bit was received, and the received byte is a Given or Broadcast address. In Mode 0, SM2 should be 0.

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

Bit 3: TB8, The 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired.

Bit 2: RB8, In Modes 2 and 3, the 9th data bit that was received. In Mode 1, if SM2 = 0, RB8 is the stop bit that was received. In Mode 0, RB8 is not used.

Bit 1: TI. Transmit interrupt flag.
0: Must be cleared by software.

1: Set by hardware at the end of the 8th bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission.

Bit 0: RI. 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 SM2).

SBUF: Serial Buffer Register
SFR Address = 0x99
SFR Page = All Reset Value = XXXX-XXXX

7	6	5	4	3	2	1	0
SBUF[7]	SBUF[6]	SBUF[5]	SBUF[4]	SBUF[3]	SBUF[2]	SBUF[1]	SBUF[0]
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~0: It is used as the buffer register in transmission and reception.

SADDR: Slave Address Register
SFR Address = 0xA9
SFR Page = All Reset Value = 0000-0000

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

SADEN: Slave Address Mask Register
SFR Address = 0xB9
SFR Page = All Reset Value = 0000-0000

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

SADDR register is combined with SADEN register to form Given/Broadcast Address for automatic address recognition. In fact, SADEN functions as the "mask" register for SADDR register. The following is the example for it.

SADDR = 1100 0000
SADEN = 1111 1101
Given = 1100 00x0 ⟶ The Given slave address will be checked except bit 1 is treated as "don't care"

The Broadcast Address for each slave is created by taking the logical OR of SADDR and SADEN. Zero in this result is considered as "don't care". Upon reset, SADDR and SADEN are loaded with all 0s. This produces a Given Address of all "don't care" and a Broadcast Address of all "don't care". This disables the automatic address detection feature.

PCON0: Power Control Register 0
SFR Address = 0x87
SFR Page = All Reset Value = 00X1-0000

7	6	5	4	3	2	1	0
SMOD1	SMOD0	--	POF	GF1	GF0	PD	IDL
R/W	**R/W**	**R**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7: SMOD1, double Baud rate control bit.
0: Disable double Baud rate of the UART.
1: Enable double Baud rate of the UART in mode 1, 2, or 3.

Bit 6: SMOD0, Frame Error select.
0: SCON.7 is SM0 function.
1: SCON.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 Address = 0x87

SFR Page = All Reset Value = 0000-XX00

7	6	5	4	3	2	1	0
T0X12	T1X12	URM0X6	--	--	--	T1CKOE	T0CKOE
R/W	**R/W**	**R/W**	**R**	**R**	**R**	**R/W**	**R/W**

Bit 6: 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 5: URM0X6, Serial Port mode 0 baud rate selector.
0: Clear to select SYSCLK/12 as the baud rate for UART Mode 0.
1: Set to select SYSCLK/2 as the baud rate for UART Mode 0.

12.9. Serial Port Sample Code

(1). Required Function: IDLE mode with RI wake-up capability

Assembly Code Example:

```
PS                  EQU                 10h
PSH                 EQU                 10h


     ORG      00023h
uart_ri_idle_isr:
     JB         RI,RI_ISR                              ;
     JB         TI,TI_ISR                               ;
     RETI                                                   ;

RI_ISR:
;     Process
     CLR        RI                                        ;
     RETI                                                   ;

TI_ISR:
;     Process
     CLR        TI                                         ;
     RETI                                                   ;

main:
     CLR        TI                                        ;
     CLR        RI                                       ;
     SETB       SM1                                   ;
     SETB       REN                                   ; 8bit Mode2, Receive Enable

     CALL       UART_Baud_Rate_Setting          ;refer to “錯誤! 找不到參照來源。 ~ 錯誤! 找不到參照來源。“ for detailed information

     MOV         IP,#PSL                             ; Select UART interrupt priority
     MOV         IPH,#PSH                           ;

     SETB       ES                                      ; Enable S0 interrupt
     SETB       EA                                      ; Enable global interrupt

     ORL        PCON,#IDL;                         ; Set MCU into IDLE mode
```

C Code Example:

```
#define        PS                              0x10
#define        PSH                            0x10

void uart_ri_idle_isr(void) interrupt 4
{      if(RI)
       {
              RI=0;
              // to do ...
       }

       if(TI)
       {
              TI=0;
              // to do ...
       }
}

void main(void)
{
       TI = RI = 0;
```

```
    SM1 = REN = 1;                                      // 8bit Mode2, Receive Enable

    UART_Baud_Rate_Setting()                    // refer to “錯誤! 找不到參照來源。 ~ 錯誤! 找不到參照來
源。“ for detailed information

    IP = PSL;                                               // Select S0 interrupt priority
    IPH = PSH;                                  //

    ES = 1;                                         // Enable S0 interrupt
    EA = 1;                                         // Enable global interrupt

    PCON |= IDL;                                // Set MCU into IDLE mode
}
```

13. Analog Comparator

A single analog comparator is provided in the MG82FE(L)308/316. The comparator operation is such that the output is a logical “1” when the plus input AIN0 is greater than the minus input AIN1. Otherwise the output is a zero. There are four analog plus inputs to be selected in on AIN0. They are AC_PI0, AC_PI1, AC_PI2 and AC_PI3 which are multiplexed by PIS[1:0]. MVRS[3:0] select the 16 inputs on AIN1 that include AC_MI and 15 segments VDD reference. The comparator output is read out by CPU on ACCON.ACOUT.

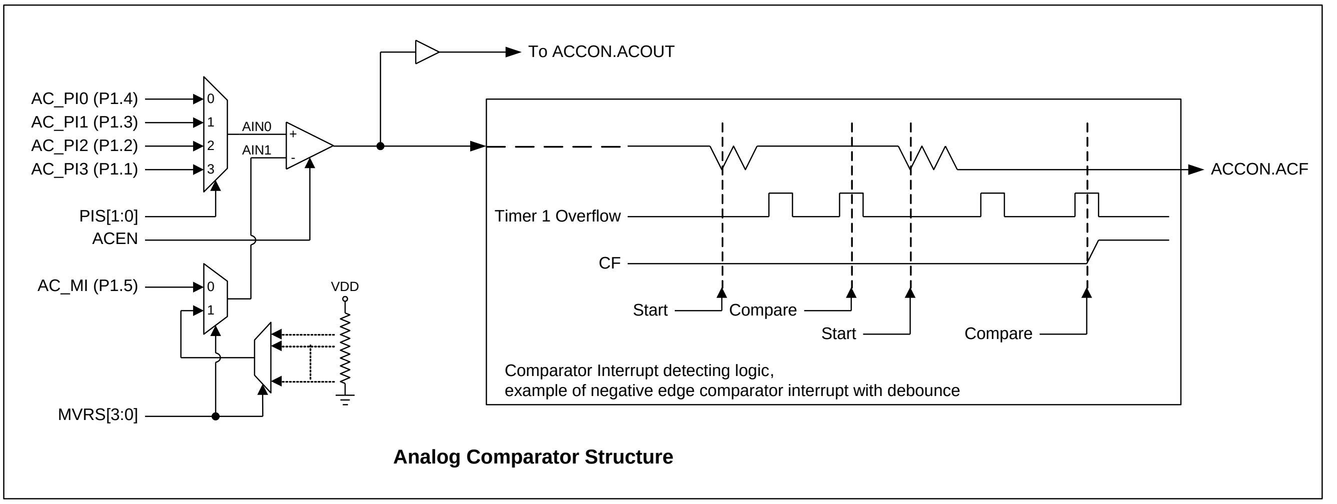
The default I/O states of AC_MI(P1.5), AC_PI0(P1.4), AC_PI1(P1.3), AC_PI2(P1.2) and AC_PI3(P1.1) are set to quasi-bidirectional I/O with on-chip pull-up resistor. To use the analog comparator properly, CPU must configure the selected comparator input channel to Input-Only mode to disable pull-up resistor effect. When enter device power down mode without comparator operating, must set the I/O ports to quasi-bidirectional, push-pull output or open-drain output low to fix I/O state for saving current consumption.

Setting the ACEN bit in ACCON enables the comparator. When the comparator is first enabled, the comparator output and interrupt flag are not guaranteed to be stable for 10 microseconds. The corresponding comparator interrupt should not be enabled during that time, and the comparator interrupt flag must be cleared before the interrupt is enabled in order to prevent an immediate interrupt service.

The comparator may be configured to cause an interrupt under a variety of output value conditions by setting the ACM bits in ACCON. The comparator interrupt flag ACF in ACCON is set whenever the comparator output matches the condition specified by ACM[2:0]. The flag may be polled by soft-ware or may be used to generate an interrupt and must be cleared by software.

The analog comparator in MG82FE(L)308/316 supports the wakeup function in Idle or Power-down mode. For this application, the comparator is conditional enabled during Idle or Power-down modes by ACIDX and ACPDX in ACCON register. If the comparator and its interrupt are enabled, the comparator can wake up CPU in Idle or Power-down mode. The detailed function is described in ACCON register description.

13.1. Analog Comparator Structure



Analog Comparator Structure

The comparator output is sampled at each clock cycle. The conditions on the analog inputs may be such that the comparator output will toggle excessively. This is especially true if applying slow moving analog inputs. Three debouncing modes are provided to filter out this noise. In debouncing mode, the comparator uses Timer 1 to modulate its sampling time. When a relevant transition occurs, the comparator waits until two Timer 1 overflows have occurred before re-sampling the output. If the new sample agrees with the expected value, ACF is set. Otherwise, the event is ignored. The filter may be tuned by adjusting the timeout period of Timer 1. Because Timer 1 is free running, the debouncer must wait for two overflows to guarantee that the sampling delay is at least 1 timeout period. Therefore after the initial edge event, the interrupt may occur between 1 and 2 timeout periods later.

13.2. Analog Comparator Register

ACCON: Analog Comparator Control Register

SFR Address = 0x9E

SFR Page = All Reset Value = 00X0-0000

7	6	5	4	3	2	1	0
ACIDX	ACPDX	ACOUT	ACF	ACEN	ACM2	ACM1	ACM0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Bit 7: ACIDX, Analog Comparator running control in IDLE mode.
0: Program the Analog Comparator to be gated off during IDLE mode.
1: Program the Analog Comparator to continue its function during IDLE mode.

Bit 6: ACPDX, Analog Comparator control in PD mode.
0: Program the Analog Comparator to be gated off during PD mode.
1: Program the Analog Comparator to continue its function during PD mode.
If ACEN, ACPDX and EACI have been set, the comparator in PD function can only wake up CPU in low level or high level mode.

Bit 5: ACOUT, this is a read only bit from comparator output.

Bit 4: ACF. Analog Comparator Interrupt Flag.
0: The flag must be cleared by software.
1: Set when the comparator output meets the conditions specified by the ACM [2:0] bits and ACEN is set. The interrupt may be enabled/disabled by setting/clearing bit 6 of IE.

Bit 3: ACEN. Analog Comparator Enable.
0: Clearing this bit will force the comparator output low and prevent further events from setting ACF.
1: Set this bit to enable the comparator.

Bit 2~0: ACM2 ~ ACM1, Analog Comparator Interrupt Mode.

ACM[2:0]	Interrupt Mode
0 0 0	Negative (Low) level
0 0 1	Positive edge
0 1 0	Toggle with debounce
0 1 1	Positive edge with debounce
1 0 0	Negative edge
1 0 1	Toggle
1 1 0	Negative edge with debounce
1 1 1	Positive (High) level

ACMOD: Analog Comparator Mode Register

SFR Address = 0x9F
SFR Page = All Reset Value = 0000-XX00

7	6	5	4	3	2	1	0
MVRS3	MVRS2	MVRS1	MVRS0	--	--	PIS1	PIS0
R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bit 7~4: MVRS[3:0], Minus Voltage Reference selector of analog comparator. The four bits determine the analog comparator (V-) input source as following:

MVRS[3:0]	(V-) Input	MVRS[3:0]	(V-) Input
0000	AC_MI(P1.5)	1000	8/16 VDD
0001	1/16 VDD	1001	9/16 VDD
0010	2/16 VDD	1010	10/16 VDD
0011	3/16 VDD	1011	11/16 VDD
0100	4/16 VDD	1100	12/16 VDD
0101	5/16 VDD	1101	13/16 VDD
0110	6/16 VDD	1110	14/16 VDD
0111	7/16 VDD	1111	15/16 VDD

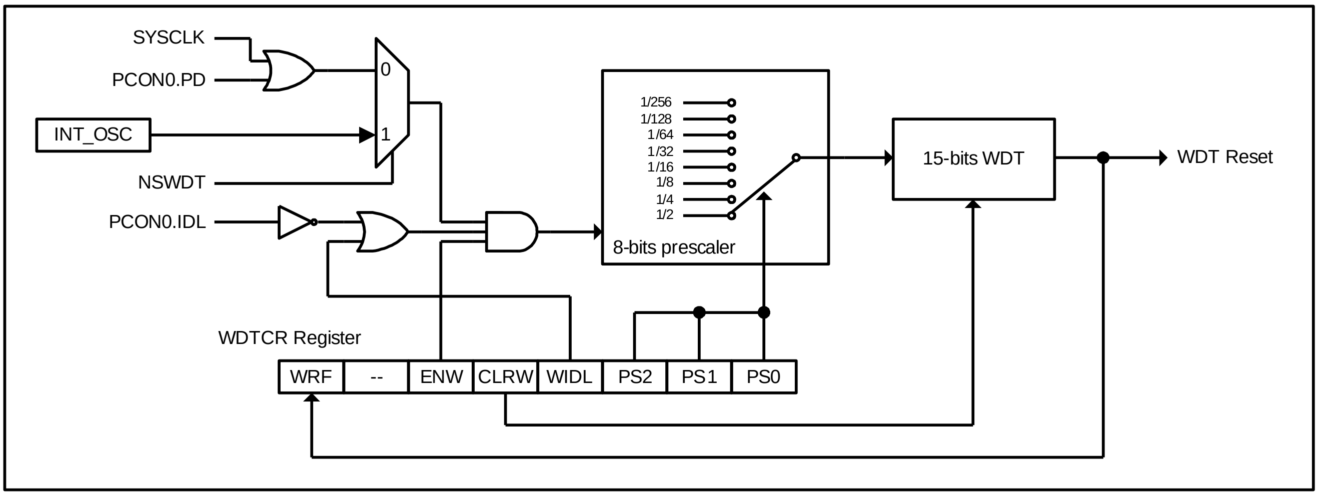
Bit 3~2: Reserved.

Bit 1~0: PIS[3:0], Plus Voltage Reference selector of analog comparator. The two bits determine the analog comparator (V+) input source as following:

PIS[2:0]	(V+) Input Select
00	AC_PI0(P1.4)
01	AC_PI1(P1.3)
10	AC_PI2(P1.2)
11	AC_PI3(P1.1)

14. Watch Dog Timer (WDT)

14.1. WDT Structure



14.2. WDT Register

WDTCR: Watch-Dog-Timer Control Register

SFR Address = 0xE1

SFR Page = All Reset Value = 0X00-XXXX

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

Bit 7: WRF, WDT reset flag.
0: This bit should be cleared by software.
1: When WDT overflows, this bit is set by hardware.

Bit 6: Reserved.

Bit 5: ENW. Enable WDT.
0: ENW can not be cleared by software. It is only cleared by POR.
1: Enable WDT while it is set.

Bit 4: CLRW. Clear WDT counter.
0: Hardware will automatically clear this bit.
1: Clear WDT to recount while it is set.

Bit 3: WIDL. WDT idle control.
0: WDT stops counting while the MCU is in idle mode.
1: WDT keeps counting while the MCU is in idle mode.

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

PS[2:0]	Prescaler Value
0 0 0	2
0 0 1	4
0 1 0	8
0 1 1	16
1 0 0	32
1 0 1	64
1 1 0	128

1 1 1	256

14.3. WDT Hardware Option

14.4. WDT Sample Code

(1) Required function: Enable WDT and select WDT prescalar to 1/32

Assembly Code Example:

```
PS0             EQU        01h
PS1             EQU        02h
PS2             EQU        04h
WIDL            EQU        08h
CLRW            EQU        10h
ENW             EQU        20h
WRF             EQU        80h

    ANL     WDTCR,#(0FFh - WRF)             ; Clear WRF flag (write “0”)
    MOV      WDTCR,#(ENW + CLRW + PS2)      ; Enable WDT counter and set WDT prescaler to 1/32
```

C Code Example:

```
#define     PS0        0x01
#define     PS1        0x02
#define     PS2        0x04
#define     WIDL       0x08
#define     CLRW       0x10
#define     ENW        0x20
#define     WRF        0x80

    WDTCR &= ~WRF;                            // Clear WRF flag (write “0”)
    WDTCR = (ENW | CLRW | PS2);             // Enable WDT counter and set WDT prescaler to 1/32
                                              // PS[2:0]    | WDT prescaler selection
                                              //     0      | 1/2
                                              //     1      | 1/4
                                              //     2      | 1/8
                                              //     3      | 1/16
                                              //     4      | 1/32
                                              //     5      | 1/64
                                              //     6      | 1/128
                                              //     7      | 1/256
```

15. Power Management

The MG82FE(L)308/316 supports one power monitor module, Brown-Out Detector, and two power-reducing modes: Idle and Power-down. These modes are accessed through the PCON0 and PCON1 register.

15.1. Power Saving Mode

15.1.1. 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, Timer 2, PWM Timer and the UART will continue to function during Idle mode. The analog comparator and WDT are conditional enabled during Idle mode to wake up CPU. Any enabled interrupt source or reset may terminate Idle mode. When exiting Idle mode with an interrupt, the interrupt will immediately be serviced, and following RETI, the next instruction to be executed will be the one following the instruction that put the device into Idle. There is another Idle existing mechanism by enabled wakeup GPIOs that don’t builds interrupt capability.

The channel inputs of analog comparator should be set to “output 0” or “quasi-bidirectional” when comparator is disabled in idle mode or power-down mode.

15.1.2. Power-down Mode

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

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.

15.1.3. Interrupt Recovery from Power-down

Six external interrupts may be configured to terminate Power-down mode. External interrupts nINT0 (P3.2), nINT1 (P3.3), nINT2 (P4.3), nINT3 (P4.2), nINT4 (P5.0) and nINT5 (P5.1) may be used to exit Power-down. To wake up by external interrupt nINT0, nINT1, nINT2, nINT3, nINT4 or nINT5, the interrupt must be enabled and configured for level-sensitive operation.

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.

15.1.4. Reset Recovery from Power-down

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

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

15.1.5. GPIO wakeup Recovery from Power-down

The GPIOs of MG82FE(L)308/316, P1.7 ~ P1.0 and general ports (indexed by GPWKS[1:0]) have wakeup CPU capability that are enabled by individual control bit in P1WKPE and GPWKPE. If the interrupt is disabled on P3.2/nINT0, P3.3/nINT1, P5.0/nINT4 or P5.1/nINT5, P3.2, P3.3, P5.0 or P5.1 still have the wakeup function from the GPWKPE control. But P4.2/nINT3 and P4.3/nINT2 can wakeup CPU only when the respective interrupt is enabled.

Wakeup from Power-down through an enabled wakeup GPIO is similar to the interrupt. At the falling edge of enabled wakeup 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, there is no any interrupt and CPU will execute the following command after last power-down instruction. That is, the enabled wakeup GPIOs will only have the capability to wakeup CPU without any interrupt function.

The enabled wakeup GPIOs also have the wakeup function form Idle mode. It will resume the CPU execute the instruction following last halt CPU command, set IDLE instruction.

15.2. Brown-Out Detector

15.3. Power Control Register

PCON0: Power Control Register 0

SFR Address = 0x87
SFR Page = All Reset Value = 0001-0000

7	6	5	4	3	2	1	0
SMOD1	SMOD0	--	POF	GF1	GF0	PD	IDL
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

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 Address = 0x97
SFR Page = All Reset Value = XXXX-XXX0

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

Bit 7~1: Reserved.

Bit 0: BOD, Brown-Out Detection flag.
0: This bit must be cleared by software.
1: This bit is set if the operating voltage matches the detection level of Brown-Out Detector.

P1WKPE: Port 1 Wakeup Enable Control Register

SFR Address = 0xD7
SFR Page = All Reset Value = 0000-0000

7	6	5	4	3	2	1	0

P17WKP	P16WKP	P15WKP	P14WKP	P13WKP	P12WKP	P11WKP	P10WKP
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~0: Wakeup enable bit for each Port 1 pins.
0: Disable port pin wakeup function.
1: Enable port pin wakeup function when port input at falling edge in power-down mode and idle mode.

GPWKPE: General Port Wakeup Enable Control Register

SFR Address = 0xD6
SFR Page = All Reset Value = 0000-0000

7	6	5	4	3	2	1	0
GP7WKP	GP6WKP	GP5WKP	GP4WKP	GP3WKP	GP2WKP	GP1WKP	GP0WKP
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~0: Wakeup enable bit for selected port pins. The port pins are selected by AUXR1.GPWKS[1:0].
0: Disable port pin wakeup function.
1: Enable port pin wakeup function when port input at falling edge in power-down mode and idle mode.

AUXR1: Auxiliary Control Register 1

SFR Address = 0xA2
SFR Page = All Reset Value = 0000-0XX0

7	6	5	4	3	2	1	0
GPWKS1	GPWKS0	P5PWM	P1S0	GF2	--	--	DPS
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R**	**R**	**R/W**

Bit 7~6: GPWKS[1:0], Index a port with wakeup function with GPWKPE control.

GPWKS[1:0]	Selected Port
0 0	Port 0
0 1	Port 2
1 0	Port 3
1 1	Port 5

15.4. Power Control Sample Code

(1) Required function: Select Slow mode with OSCin/128 (default is OSCin/1)

Assembly Code Example:

```
CKS0            EQU       01h
CKS1            EQU       02h
CKS2            EQU       04h

    ORL     PCON2,#( CKS2 + CKS1 + CKS0)    ; Set CKS[2:0] = “111” to select OSCin/128
```

C Code Example:

```
#define     CKS0                0x01
#define     CKS1                0x02
#define     CKS2                0x04

    PCON2 |= (CKS2 | CKS1 | CKS0);          // System clock divider /128
                                               // CKS[2:0], system clock divider
                                               //     0     | OSCin/1
                                               //     1     | OSCin/2
                                               //     2     | OSCin/4
                                               //     3     | OSCin/8
                                               //     4     | OSCin/16
                                               //     5     | OSCin/32
                                               //     6     | OSCin/64
                                               //     7     | OSCin/128
```

16. In System Programming (ISP)

IFD: ISP/IAP Flash Data Register

SFR Address = 0xE2
SFR Page = All Reset Value = 1111-1111

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

IFD is the data port register for ISP/IAP operation. The data in IFD will be written into the desired address in operating ISP/IAP write and it is the data window of readout in operating ISP/IAP read.

If IMFT is indexed on IAPLB or AUXRA access, read/write IFD through SCMD flow will access the register content of IAPLB.

IFADRH: ISP/IAP Address for High-byte addressing

SFR Address = 0xE3
SFR Page = All Reset Value = 0000-0000

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

IFADRH is the high-byte address port for all ISP/IAP modes.

IFADRL: ISP/IAP Address for Low-byte addressing

SFR Address = 0xE4
SFR Page = All Reset Value = 0000-0000

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

IFADRL is the low byte address port for all ISP/IAP modes. In page erase operation, it is ignored.

IFMT: ISP/IAP Flash Mode Table

SFR Address = 0xE5
SFR Page = All Reset Value = XXXX-0000

7	6	5	4	3	2	1	0
--	--	--	--	MDS[3]	MDS[2]	MDS[1]	MDS[0]
R	R	R	R	R/W	R/W	R/W	R/W

Bit 7~4: Reserved

Bit 3~0: ISP/IAP operating mode selection

Bit[3:0]				Mode
0	0	0	0	Standby
0	0	0	1	AP-memory read
0	0	1	0	AP-memory program
0	0	1	1	AP-memory page erase
0	1	0	0	IAPLB write
0	1	0	1	IAPLB read
Others				Reserved for test function.

IFMT is used to select the flash mode for performing numerous ISP/IAP function.

IAPLB: IAP Low Boundary

SFR Address=indirect Reset Value = 1111-111x

7	6	5	4	3	2	1	0

IAPLB							0
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~0: The IAPLB determines the IAP-memory lower boundary. Since a Flash page has 512 bytes, the IAPLB must be an even number.
To read IAPLB, CPU need to define the IMFT for mode selection on IAPLB Read and set ISPCR.ISPEN. And then write 0x46h & 0xB9h sequentially into SCMD. The IAPLB content is available in IFD. If write IAPLB, mcu will put new IAPLB setting value in IFD firstly. And then select IMFT, enable ISPCR.ISPEN and then set SCMD. The IAPLB content has already finished the updated sequence.

The range of the IAP-memory is determined by IAPLB and the ISP start address as listed below.

IAP lower boundary = IAPLBx256, and
IAP higher boundary = ISP start address – 1.

For example, if IAPLB=0x12 and ISP start address is 0x1C00, then the IAP-memory range is located at 0x1200 ~ 0x1BFF.

Additional attention point, the IAP low boundary address must not be higher than ISP start address.

SCMD: Sequential Command Data register / RDID (Read DID register)

SFR Address = 0xE6
SFR Page = All Reset Value = xxxx-xxxx

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

SCMD is the command port for triggering ISP/IAP/IAPLB activity. If SCMD is filled with sequential 0x46h, 0xB9h and if ISPCR.7 = 1, ISP/IAP activity will be triggered.

ISPCR: ISP Control Register

SFR Address = 0xE5
SFR Page = All Reset Value = 0000-xxxx

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

Bit 7: ISPEN, ISP/IAP operation enable.
0: Global disable all ISP/IAP program/erase/read function.
1: Enable ISP/IAP program/erase/read function.

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

Bit 5: SWRST, software reset trigger control.
0: No operation
1: Generate software system reset. It will be cleared by hardware automatically.

Bit 4: CFAIL, Command Fail indication for ISP/IAP operation.
0: The last ISP/IAP command has finished successfully.
1: The last ISP/IAP command fails. It could be caused since the access of flash memory was inhibited.

Bit 3~0 : Reserved

ISP Description in more detail

MG82FE(L)308/316 does not make use of idle-mode to perform ISP operation. Instead, it creates CPU wait-state to release flash memory for ISP control circuit use. Once ISP run over, CPU will be waken-up and advanced to the instruction which follows the previous instruction that invokes ISP activity. During ISP operation, interrupt service is also blocked until ISP run over.

ISP control circuit has a built-in timer for timing sequence control. It is referred from OSC frequency and defined by CKCON2.XCKS[4:0] to get the accuracy erase/program timing.

17. In Application Programming (IAP)

MG82FE(L)308/316 available program memory size (AP-memory) is restricted to 8K/13.5K. The flash memory between IAPLB and ISP start address could be defined as data flash memory and can be accessed by the ISP operation in field application. The size of IAP flash memory is variable. It is defined by IAPLB.

17.1. ISP/IAP Sample Code

(1). Required Function: General function call for ISP/IAP flash read

Assembly Code Example:

```
IxP_Flash_Read     EQU                01h
ISPEN              EQU                80h

_ixp_read:
ixp_read:

        MOV        ISPCR,#ISPEN                        ; Enable Function
        MOV        IFMT,# IxP_Flash_Read              ; ixp_read=0x01

        MOV        IFADRH,??                            ; fill [IFADRH,IFADRL] with byte address
        MOV        IFADRL,??

        MOV        SCMD,#046h                          ;
        MOV        SCMD,#0B9h                          ;

        MOV        A,IFD                                ; now, the read data exists in IFD

        MOV        IFMT,#000h                          ; Flash_Standby=0x00
        ANL       ISPCR,#(0FFh – ISPEN)               ; Disable Function

        RET
```

C Code Example:

```
#define       Flash_Standby               0x00
#define       IxP_Flash_Read              0x01
#define       ISPEN                       0x80


unsigned char ixp_read (void)
{
        unsigned char arg;
        ISPCR = ISPEN;                              // Enable Function
        IFMT = IxP_Flash_Read;                      // IxP_Read=0x01

        IFADRH = ??
        IFADRL = ??

        SCMD = 0x46;                                //
        SCMD = 0xB9;                                //

        arg = IFD;

        IFMT = Flash_Standby;                       // Flash_Standby=0x00
        ISPCR &= ~ISPEN;

        return arg;
}
```

(2). Required Function: General function call for ISP/IAP flash Erase

Assembly Code Example:

```
IxP_Flash_ Erase    EQU                 03h
ISPEN               EQU                 80h

_ixp_erase:
ixp_erase:

        MOV       ISPCR,#ISPEN                          ; Enable Function
        MOV       IFMT,# IxP_Flash_Erase                ; ixp_erase=0x03

        MOV       IFADRH,??                             ; fill [IFADRH,IFADRL] with byte address
        MOV       IFADRL,??

        MOV       SCMD,#046h                            ;
        MOV       SCMD,#0B9h                            ;

        MOV       IFMT,#000h                            ; Flash_Standby=0x00
        ANL      ISPCR,#(0FFh – ISPEN)                  ; Disable Function

        RET
```

C Code Example:

```
#define       Flash_Standby                 0x00
#define       IxP_Flash_Erase               0x03
#define       ISPEN                         0x80


void ixp_erase (unsigned char Addr_H, unsigned char Addr_L)
{
        ISPCR = ISPEN;                                  // Enable Function
        IFMT = IxP_Flash_Erase;                         // IxP_Erase=0x03

        IFADRH = Addr_H;
        IFADRL = Addr_L;

        SCMD = 0x46;                                    //
        SCMD = 0xB9;                                    //

        IFMT = Flash_Standby;                           // Flash_Standby=0x00
        ISPCR &= ~ISPEN;
}
```

(3). Required Function: General function call for ISP/IAP flash program

Assembly Code Example:

```
IxP_Flash_ProgramEQU            02h
ISPEN           EQU             80h

_ixp_program:
ixp_program:

       MOV       ISPCR,#ISPEN                   ; Enable Function
       MOV       IFMT,# IxP_Flash_Program       ; ixp_program=0x03

       MOV       IFADRH,??                      ; fill [IFADRH,IFADRL] with byte address
       MOV       IFADRL,??
       MOV       IFD, A                         ; now, the program data exists in Accumulator

       MOV       SCMD,#046h                     ;
       MOV       SCMD,#0B9h                     ;

       MOV       IFMT,#000h                     ; Flash_Standby=0x00
       ANL       ISPCR,#(0FFh – ISPEN)          ; Disable Function

       RET
```

C Code Example:

```
#define     Flash_Standby           0x00
#define     IxP_Flash_Program       0x02
#define     ISPEN                   0x80

void ixp_program(unsigned char Addr_H, unsigned char Addr_L, unsigned char dta)
{
       ISPCR = ISPEN;                           // Enable Function
       IFMT = IxP_Flash_Program;                // IxP_Program=0x02

       IFADRH = Addr_H;
       IFADRL = Addr_L;
       IFD = dta;

       SCMD = 0x46;                             //
       SCMD = 0xB9;                             //

       IFMT = Flash_Standby;                    // Flash_Standby=0x00
       ISPCR &= ~ISPEN;


}
```

18. Auxiliary SFRs

AUXR0: Auxiliary Register 0

SFR Address = 0x8E
SFR Page = All Reset Value = 0000-0000

7	6	5	4	3	2	1	0
P60OC1	P60OC0	P60FD	P34FD	--	--	EXTRAM	GF
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~6: P60 output configured control bit 1 and 0. The two bits only act when internal RC oscillator is selected for system clock source. In this condition, XTAL2 and XTAL1 are the alternated function for P60 and P61. P60 provides the following selections for GPIO or clock source generator. When P60OC[1:0] index to non-P60 function, XTAL2 will drive the on-chip precision RC oscillator output to provide the clock source for other devices.

P60OC[1:0]	XTAL2 function	I/O mode
00	P60	Quasi-bidirectional I/O
01	INTOSC	Push-Pull Output
10	INTOSC/2	Push-Pull Output
11	INTOSC/4	Push-Pull Output

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

Bit 4: P34FD, P3.4 Fast Driving.
0: P3.4 output with default driving.
1: P3.4 output with fast driving enabled. If P3.4 is configured to T0CKO, enable this bit when P3.4 output frequency is more than 12MHz at 5V application or more than 6MHz at 3V application.

Bit 1: EXTRAM, External data RAM enable.
0: Enable on-chip expanded data RAM (XRAM 256 bytes)
1: Disable on-chip expanded data RAM.

Bit 0: Reserved.

AUXR1: Auxiliary Control Register 1

SFR Address = 0xA2
SFR Page = All Reset Value = 0000-0000

7	6	5	4	3	2	1	0
GPWKS1	GPWKS0	P5PWM	P1S0	GF2	--	--	DPS
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**	**R/W**

Bit 7~6: GPWKS[1:0], Index a port with wakeup function among P0, P2, P3 and P5.

GPWKS[1:0]	Selected Port
00	Port 0
01	Port 2
10	Port 3
11	Port 5

Bit 5: P5PWM, set PWM output on P5.
0: PWM output from PWM Timer module on P2.
1: Set PWM output on P5 to extend PWM output channel.

Bit 4: P1S0, Serial Port on P1.7/P1.6.
0: Keep serial port RXD/TXD on P3.0/P3.1.
1: Switch serial port RXD/TXD on P1.6/P1.7.

Bit 3: GF2, General purpose Flag 2.

Bit 2~1: Reserved.

Bit 0: DPS, dual DPTR Selector.
0: Select DPTR0.
1: Select DPTR1.

AUXR2: Auxiliary Register 2
SFR Address = 0x87
SFR Page = All Reset Value = 000X-XX00

7	6	5	4	3	2	1	0
T0X12	T1X12	URM0X6	--	--	--	T1CKOE	T0CKOE
R/W	**R/W**	**R/W**	**R**	**R**	**R**	**R/W**	**R/W**

Bit 7: T0X12, Timer 1 clock source selector while C/T=0.
0: Clear to select SYSCLK/12.
1: Set to select SYSCLK as the clock source.

Bit 6: 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 5: URM0X6, Serial Port mode 0 baud rate selector.
0: Clear to select SYSCLK/12 as the baud rate for UART Mode 0.
1: Set to select SYSCLK/2 as the baud rate for UART Mode 0.

Bit 3~2: Reserved.

Bit 1: T1CKOE, Timer 1 Clock Output Enable.
0: Disable Timer 1 clock output.
1: Enable Timer 1 clock output on P3.5.

Bit 0: T0CKOE, Timer 0 Clock Output Enable.
0: Disable Timer 0 clock output.
1: Enable Timer 0 clock output on P3.4.

19. 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 RESET with respect to Ground	-0.5 ~ VDD + 0.5	V
Voltage on VDD with respect to Ground	-0.5 ~ +6.0	V
Maximum total current through VDD and Ground	400	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.

20. Electrical Characteristics

20.1. DC Characteristics

VSS = 0V, TA = 25 ℃, VDD = 5.0V, unless otherwise specified

Symbol	Parameter	Test Condition	Limits			Unit
			min	typ	max	
V_{IH1}	Input High voltage, P0/P1/P2/P3/P4/P5/P6/P7 (Quasi, Input-only or Open-drain)		2.0			V
V_{IH2}	Input High voltage, RST		3.5			V
V_{IL1}	Input Low voltage, P0/P1/P2/P3/P4/P5/P6/P7 (Quasi, Input-only or Open-drain)				0.8	V
V_{IL2}	Input Low voltage, RST				1.6	V
I_{IH1}	Input High Leakage current P0/P1/P2/P3/P4/P5/P6/P7 (Quasi, Input-only or Open-drain)	V_{PIN} = VDD		0	10	uA
I_{IL1}	Logic 0 input current P0/P1/P2/P3/P4/P5/P6/P7 (Quasi-bidirectional)	V_{PIN} = 0.4V		20	50	uA
I_{IL2}	Logic 0 input current P0/P1/P2/P3/P4/P5 (Input-only or Open-drain)	V_{PIN} = 0.4V		0	10	uA
I_{H2L}	Logic 1 to 0 input transition current, P0/P1/P2/P3/P4/P5/P6/P7 (Quasi-bidirectional)	V_{PIN} =1.8V		250	500	uA
I_{OH1}	Output High current, P0/P1/P2/P3/P4/P5/P6/P7 (Quasi-bidirectional)	V_{PIN} =2.4V	-150	-220		uA
I_{OH2}	Output High current, P0/P1/P2/P3/P4/P5 (Push-pull output)	V_{PIN} =2.4V	-12			mA
I_{OL1}	Output Low current P0/P1/P2/P3/P4/P5/P6/P7 (Quasi, Open-drain or Push-pull output)	V_{PIN} =0.4V	12			mA
I_{OP}	Operating current	F_{OSC} = 12MHz		14	20	mA
I_{IDLE}	Idle mode current	F_{OSC} = 12MHz		6	10	mA
I_{PD}	Power down current			0.1	10	uA
R_{RST}	Internal reset pull-down resistance			100		Kohm
V_{CM}	Comparator input common mode voltage		0		VDD	V
V_{OS}	Comparator input offset voltage			10	30	mV
V_{BOD}	Brown-out detection voltage	F_{OSC} = 12MHz	3.95	4.0	4.05	V

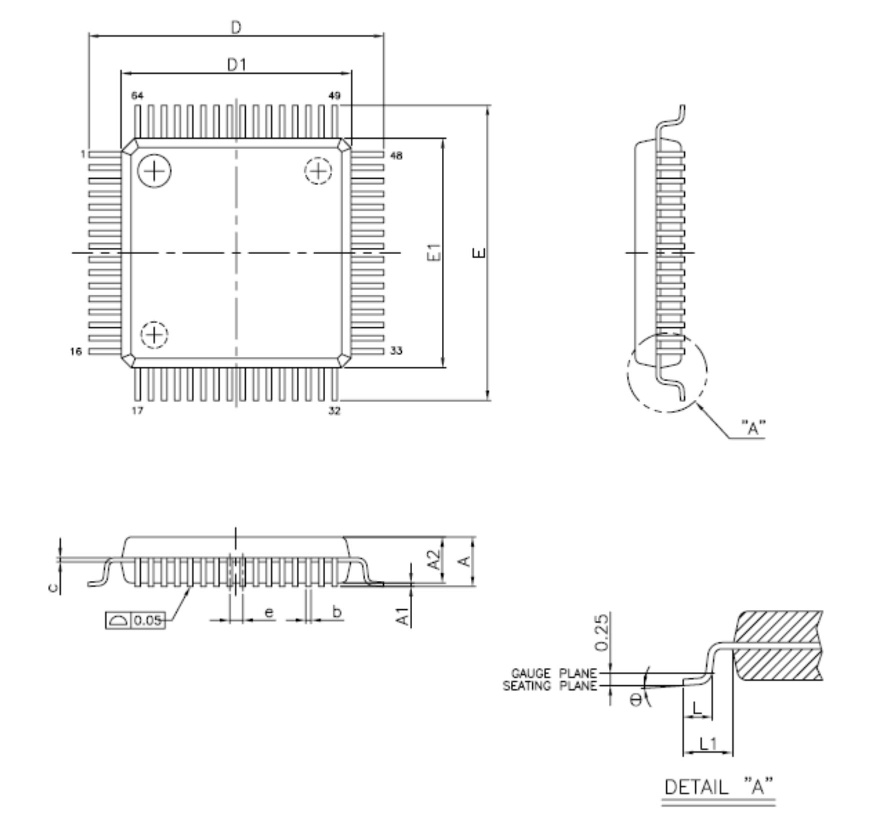
VSS = 0V, TA = 25 ℃, VDD = 3.3V, unless otherwise specified

Symbol	Parameter	Test Condition	Limits			Unit
			min	typ	max	
V_{IH1}	Input High voltage, P0/P1/P2/P3/P4/P5/P6/P7 (Quasi, Input-only or Open-drain)		2.0			V
V_{IH2}	Input High voltage, RST		2.8			V
V_{IL1}	Input Low voltage, P0/P1/P2/P3/P4/P5/P6/P7 (Quasi, Input-only or Open-drain)				0.8	V
V_{IL2}	Input Low voltage, RST				1.5	V

I_{IH1}	Input High Leakage current P0/P1/P2/P3/P4/P5/P6/P7 (Quasi, Input-only or Open-drain)	V_{PIN} = VDD		0	10	uA
I_{IL1}	Logic 0 input current P0/P1/P2/P3/P4/P5/P6/P7 (Quasi-bidirectional)	V_{PIN} = 0.4V		7	30	uA
I_{IL2}	Logic 0 input current P0/P1/P2/P3/P4/P5 (Input-only or Open-drain)	V_{PIN} = 0.4V		0	10	uA
I_{H2L}	Logic 1 to 0 input transition current, P0/P1/P2/P3/P4/P5/P6/P7 (Quasi-bidirectional)	V_{PIN} =1.8V		100	250	uA
I_{OH1}	Output High current, P0/P1/P2/P3/P4/P5/P6/P7 (Quasi-bidirectional)	V_{PIN} =2.4V	-40	-70		uA
I_{OH2}	Output High current, P0/P1/P2/P3/P4/P5 (Push-pull output)	V_{PIN} =2.4V	-4			mA
I_{OL1}	Output Low current P0/P1/P2/P3/P4/P5/P6/P7 (Quasi, Open-drain or Push-pull output)	V_{PIN} =0.4V	8			mA
I_{OP}	Operating current	F_{OSC} = 12MHz		11	15	mA
		F_{OSC} = 24MHz		17	22	
I_{IDLE}	Idle mode current	F_{OSC} = 12MHz		4	6	mA
		F_{OSC} = 24MHz		6	9	
I_{PD}	Power down current			0.1	10	uA
R_{RST}	Internal reset pull-down resistance			200		Kohm
V_{CM}	Comparator input common mode voltage		0		VDD	V
V_{OS}	Comparator input offset voltage			10	30	mV
V_{BOD}	Brown-out detection voltage	F_{OSC} = 12MHz	2.35	2.4	2.45	V

21. Package Dimension

LQFP-64 (7mm X 7mm)



VARIATIONS (ALL DIMENSIONS SHOWN IN MM)

SYMBOLS	MIN.	NOM.	MAX.
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
b	0.13	0.18	0.23
c	0.09	–	0.20
D	9.00 BSC		
D1	7.00 BSC		
e	0.40 BSC		
E	9.00 BSC		
E1	7.00 BSC		
L	0.45	0.60	0.75
L1	1.00 REF		
θ	0°	3.5°	7°

NOTES:

1.JEDEC OUTLINE : MS-026 BBD

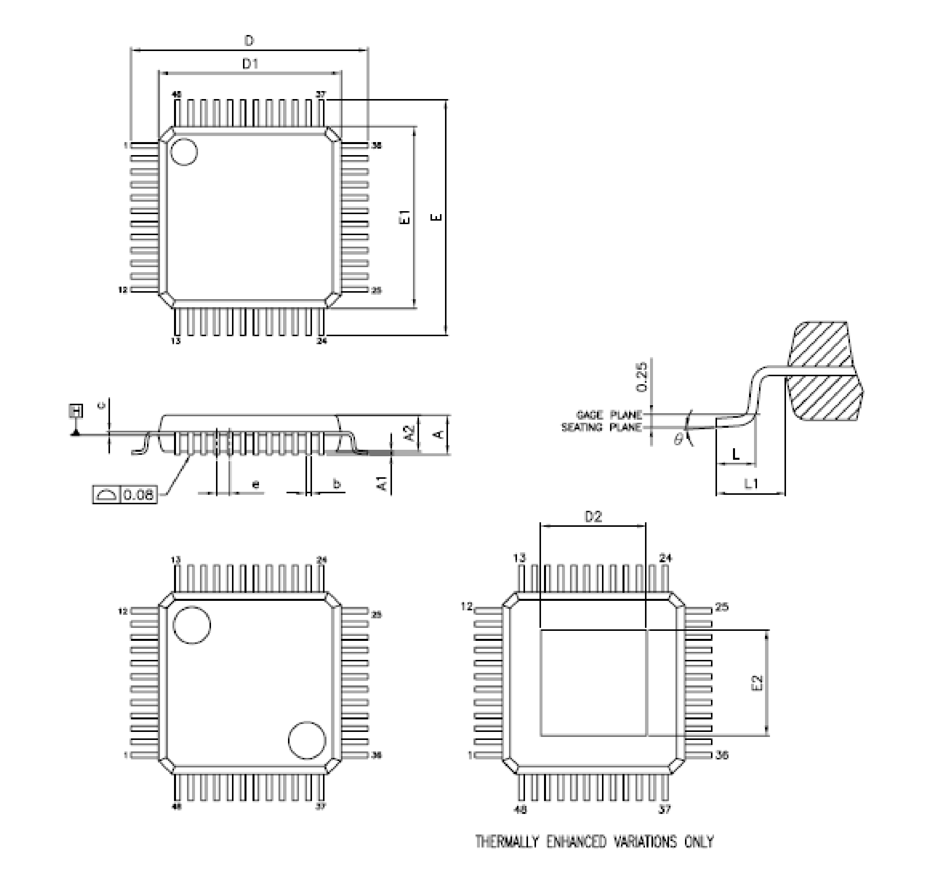
2.DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25mm PER SIDE. D1 AND E1 ARE MAXIMUM PLASTIC BODY SIZE DIMENSIONS INCLUDING MOLD MISMATCH.

3.DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.ALLOWABLE DAMBAR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED THE MAXIMUM b DIMENSION BY MORE THAN 0.08mm.

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

3RD ANGLE SYS.	比例 SCALE:	材質 MTRL:	製程 FINISH:	數量 Q'TY:
繪圖: DWN: 施佩杏 日期: DATE: 8/15/06'	圖名: TITLE: LOW PROFILE PLASTIC QUAD FLAT PACKAGE DATA SHEET 64 LEADS			
審核: CHK: 林俊隆 日期: DATE: 8/15/06'	圖號: DWG#: MW-D264-002			
核准: APPL: Rex 日期: DATE: 09/10/09'	圖檔: FILE: MW-D264-002-02		版別: REV: 02	張數: Q'TY: 1

LQFP-48



VARIATIONS (ALL DIMENSIONS SHOWN IN MM)

SYMBOLS	MIN.	NOM.	MAX.
A	--	--	1.60
A1	0.05	--	0.15
A2	1.35	1.40	1.45
b	0.17	0.22	0.27
c	0.09	--	0.20
D	9.00 BSC		
D1	7.00 BSC		
E	9.00 BSC		
E1	7.00 BSC		
e	0.50 BSC		
L	0.45	0.60	0.75
L1	1.00 REF		
θ	0°	3.5°	7°

⚠ THERMALLY ENHANCED DIMENSIONS(SHOWN IN MM)

PAD SIZE	E2		D2	
	MIN.	MAX.	MIN.	MAX.
205X20E	4.31	5.21	4.31	5.21

NOTES:

1.JEDEC OUTLINE :
MS-026 BBC
MS-026 BBC-HD(THERMALLY ENHANCED VARIATIONS ONLY)

2.DATUM PLANE H IS LOCATED AT THE BOTTOM OF THE MOLD PARTING LINE COINCIDENT WITH WHERE THE LEAD EXITS THE BODY.

3.DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 mm PER SIDE. DIMENSIONS D1 AND E1 DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE H.

4.DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.

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

3RD ANGLE SYS.	比例 SCALE:	材質 MTRL:	製程 FINISH:	數量 QTY:

繪圖 DWN: 施佩杏 日期 DATE: 5/14/08'	圖名 TITLE: LOW PROFILE PLASTIC QUAD FLAT PACKAGE DATA SHEET 48 LEADS (7X7X1.4mm)
審核 CHK: 林俊隆 日期 DATE: 5/14/08'	圖號 DWG#: MW-D148-001
核准 APPL: Rex 日期 DATE: 7/1/08'	圖檔 FILE: MW-D148-001-03 版次 REV: 03 張數 QTY: 1

22. Revision History

Version	Date	Page	Description
V0.03	2009/Nov./16		Initial release
V0.04	2010/Mar./24		Modify error description
		10	256B SFR→ 128B
		9	Delete P6
		24	Add LQFP-48 table
V0.05	2010/Jul./06	79	Modify Ambient temperature under bias -40~+85
		7	Add Int. RC-OSC draft range.
V0.06	2010/Sep./07	14	**Pin Description** LQFP-64: P4.5 & P4.6 Pin number error P4.5: Pin 43, P4.6: Pin 39
A1.0	2013/Dec./02		

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).