BM

Brief Manual of MiDAS230 Family

FLASH / ISP / IAP
8-bit Turbo Microcontrollers

V1.1

April 2011

Contents

Preliminary

1. Product Overview

Preliminary

- **CORERIVER's Family is a group of fast 80C52 compatible microcontrollers**
- **The instruction execution is max. 3 times faster than that of traditional 80C52.**
  - ✓ **1 Machine cycle = 4 clocks vs. 12 clocks**
- **Additional peripherals of MiDAS230 Family:**
  - ✓ **10 bit ADC / WDT / LVD / POR.**
- **Power saving modes**
- **Noise tolerant scheme**
- **Support ISP / IAP of FLASH memory**
- **Provides Easy-to-Use training-kit system**
- **The Brief Manual contents could be updated at any time. Please check update contents from CORERIVER Web Site (http://www.coreriver.com)**

A. MiDAS230 Family

Product	MASK [Byte]	FLASH [Byte]	EEPROM [Byte]	RAM [Byte]	Volt [V]	Freq [MHz]	T/C [16 bits]	COM I/O	WDT	ADC (bit X Ch)	PWM (bit X ch)	Package	Others
GC230-SO20I GC230-TS16I GC230-SO8I	-	2K	(128)	128	2.4 ~ 5.5	12 (24)	2	-	1	10 X 12 10 X 8 10 X 3	8 X 1	20-SOP 16-TSSOP 8-SOP	IAP ISP EJTAG LVD POR POSC

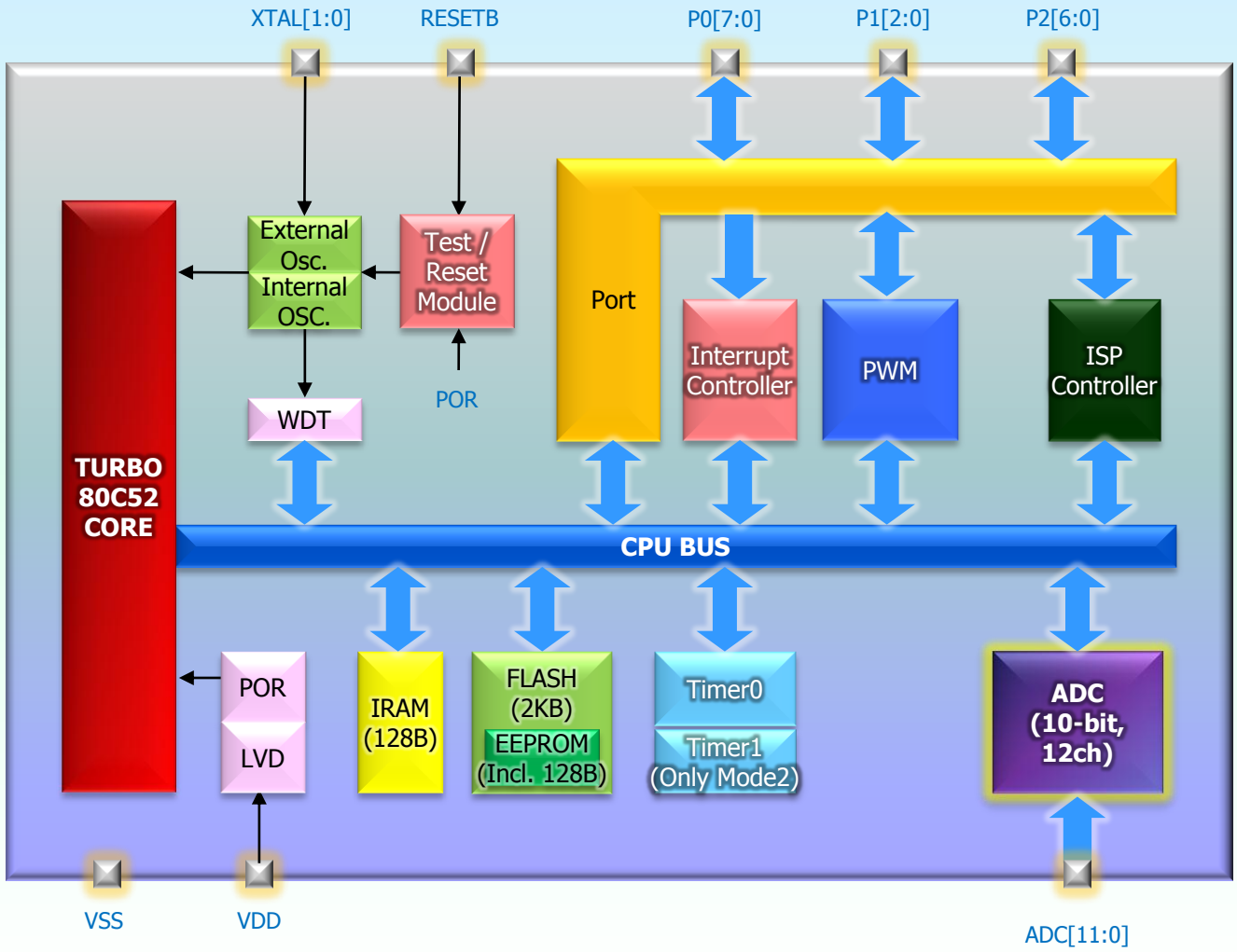
2. Features

Preliminary

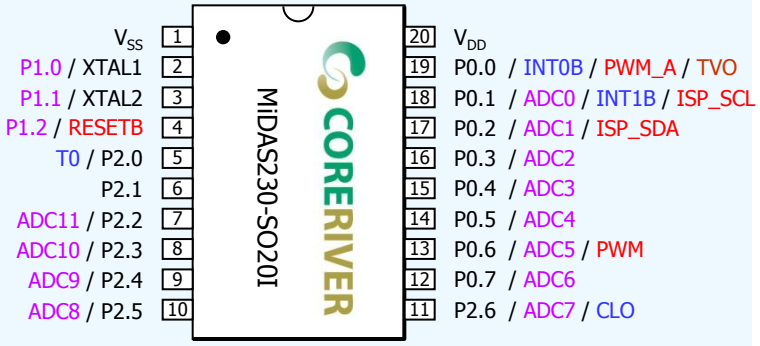
- CPU
  - ✓ 8-bit turbo 80C52 architecture
  - ✓ 4 cycles/1 machine cycle
  - ✓ instruction level compatible with Intel 80C52
- 2kB FLASH (Including 128B User EEPROM)
- Support ISP/IAP of FLASH
- 128B Internal RAM
- Operating Voltage : +2.4V to +5.5V
- Operating Frequency (F_{SYS})
  - ✓ 4MHz (Internal Clock, Default)
  - ✓ Max. 12MHz (Internal Clock)
  - ✓ Max. 24MHz @4.5V ~ 5.5V (External Clock)
- Operating temperature : -40 °C ~ 85 °C
- Max. Programmable 18 (20-SOP), 14 (16-TSSOP), 6 (8-SOP) I/O Pins
  - ✓ Pull-up control, Open drain, & Push-Pull output
  - ✓ TTL and CMOS compatible logic levels
- Configurable Low Voltage Detector (LVD)
- Internal Precision OSC with Calibration function
  - ✓ 4MHz @+2.4 V to +3.3 V (± 1%)
- 12-channel 10-bit ADC
  - ✓ Max. 120k SPS @F_{ADC} = 12MHz
  - ✓ Programmable Input Clock Frequency
- 1-channel 8-bit High Speed PWM
- 23-bit Programmable Watchdog Timer (WDT)
- Two 16-bit Timer/Counters
- Max. 7 Interrupt Sources
  - ✓ Timer0/1, WDT, ADC, PWM
  - ✓ 2 External Interrupt Sources : Both Edge/Level
  - ✓ Two-level Interrupt Priority
- Reset Sources
  - ✓ On-chip Power-On-Reset (POR)
  - ✓ External Reset
  - ✓ Configurable Low Voltage Detector Reset (LVR)
  - ✓ Watchdog Timer Reset
- Power Down Wake-up Sources
  - ✓ Reset Sources + 2 External Interrupt (Low Level)
  - ✓ WDT interrupt
- Power Consumption (TBD)
  - ✓ Active Current : Max. 2mA @+3.0V, 2MHz
  - ✓ Idle Current : Max. 0.5mA @+3.0V, 2MHz
  - ✓ Stop Current : Max. 1uA @+5.0V (All Clock OFF)
- E.S.D. Protection up to 2,000 V
- Latch-up Protection Up to ±200mA
- Package
  - ✓ 20-SOP
  - ✓ 16-TSSOP
  - ✓ 8-SOP

3. Block Diagram

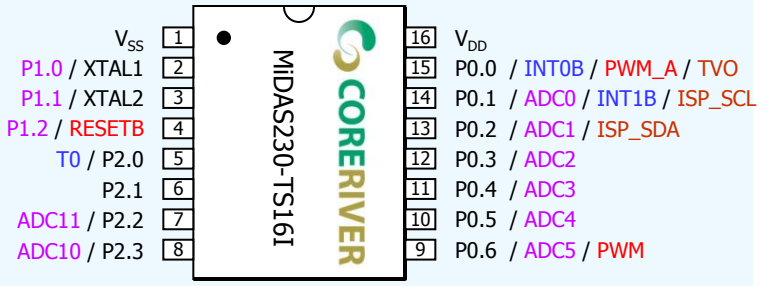
Preliminary



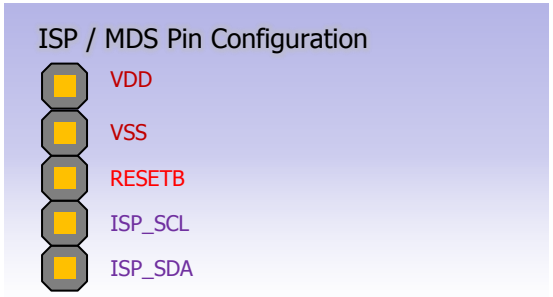
4. Pin Configurations

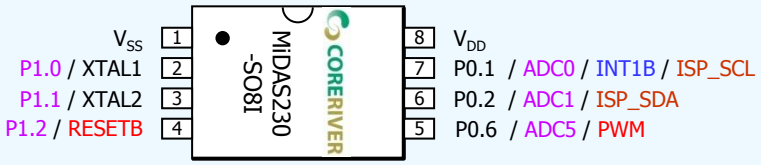


[20-SOIC]



[16-TSSOP]





[8-SOIC]

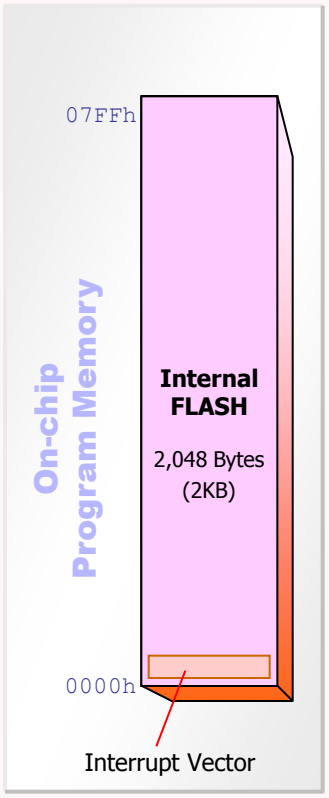
5. Pin Descriptions

Preliminary

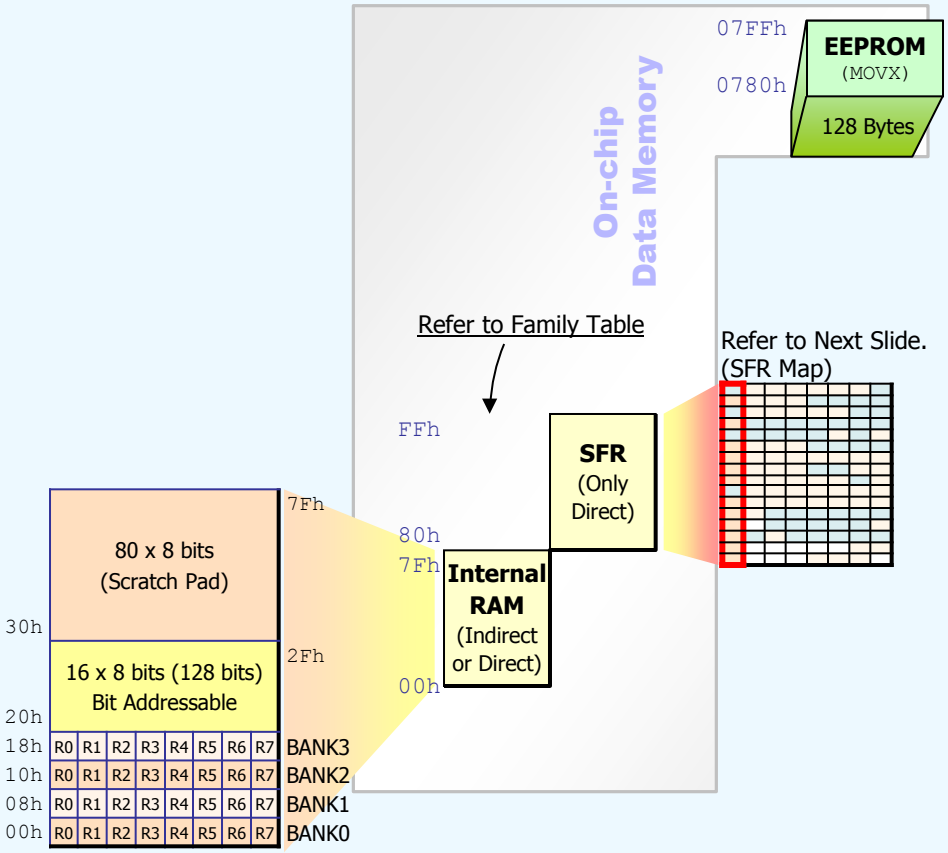
Symbol	Direction	Description	Share Pins
V_{DD}	Input	Power Supply	-
V_{SS}	Input	Ground	-
XTAL1 / P1.0	Input/Output	▪ Crystal Input/Output (Default) ▪ Bit Programmable with Schmitt Trigger - Optional Pull-up Control Enable - Open-drain Output - Push-pull Output	XTAL1 / P1.0 (Crystal Input)
XTAL2 / P1.1			XTAL2 / P1.1 (Crystal Output)
RESETB / P1.2	Input/Output	▪ External Reset Input Signal (Default) ▪ Bit Programmable - Optional Pull-up Control Enable - Open-drain Output - Push-pull Output	RESETB / P1.2
P0[7:0]	Input/Output	▪ Bit Programmable with Schmitt Trigger - Optional Pull-up Control Enable - Open-drain Output - Push-pull Output (Default)	P0.0 / INT0B / PWM_A / TVO P0.1 / ADC0 / INT1B / ISP_SCL P0.2 / ADC1 / ISP_SDA P0.3 / ADC2 P0.4 / ADC3 P0.5 / ADC4 P0.6 / ADC5 / PWM P0.7 / ADC6
P2[6:0]	Input/Output	▪ Bit Programmable with Schmitt Trigger - Optional Pull-up Control Enable - Open-drain Output - Push-pull Output (Default)	P2.0 / T0 P2.1 P2.2 / ADC11 P2.3 / ADC10 P2.4 / ADC9 P2.5 / ADC8 P2.6 / ADC7 / CLO

6.1. Memory Organization

Preliminary



[On-chip Program Memory]
(Read/Write with IAP)

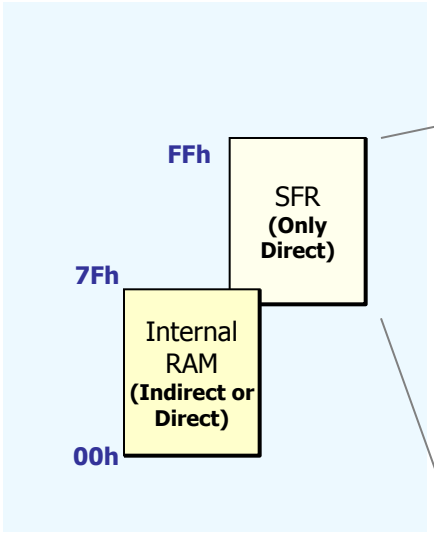


[On-chip Data Memory]
(Read and Write)

- User can write the data to FLASH or EEPROM with IAP (In-Application Programming).

6.2. SFR (Special Function Register) Map

Preliminary



: Newly added SFR at MiDAS230 Family

: Reserved for future use.

	Bit addressable								
F8h					**IAPWEN**	**IAPCON**	**IAPDAT**		FFh
F0h	B				**P0DIR**	**P1DIR**	**P2DIR**		F7h
E8h							**ADCR**	**ADCON**	EFh
E0h	ACC	**ADCSELH**	**ADCSEL**	**ALTSEL**	**P0SEL**	**P1SEL**	**P2SEL**		E7h
D8h	**WDCON**				**PWMCON**		**PWMD**		DFh
D0h	PSW				**P0TYPE**	**P1TYPE**	**P2TYPE**		D7h
C8h									CFh
C0h					**PMR**	**STATUS**	**LVDCFG**		C7h
B8h	IP						**OSCICN**		BFh
B0h									B7h
A8h	IE								AFh
A0h	P2								A7h
98h									9Fh
90h	P1	**EXIF**				**RINGCON**	**OSCBIAS**	**OSCREF**	97h
88h	TCON	TMOD	TL0	TL1	TH0	TH1			8Fh
80h	P0	SP	DPL	DPH				PCON	87h

6.2. SFR Brief Description

Preliminary

- 80C52 SFR Registers

Register	Name	Reset Value
ACC **B** **PSW** **SP**	Accumulator B Program Status Word Stack Pointer	00000000 00000000 00000000 00000111
DPTR **DPL** **DPH**	Data Pointer (2 bytes) Low Byte High Byte	 00000000 00000000
P0 **P1** **P2**	Port 0 Port 1 Port 2	11111111 *****111 *1111111
IP **IE**	Interrupt Priority Interrupt Enable Control	10000000 00000000
TCON **TMOD**	Timer/Counter 0/1 Control Timer/Counter 0 Mode Control	00000000 ****0000
TH0 **TL0** **TH1** **TL1**	Timer/Counter 0 High Byte Timer/Counter 0 Low Byte Timer/Counter 1 High Byte Timer/Counter 1 Low Byte	00000000 00000000 00000000 00000000
PCON	Power Control	***10000

*** : Don't touch bit. Read as '0'.**

- Newly added SFR Registers at MiDAS230 Family

Register	Name	Reset Value
P0SEL **P1SEL** **P2SEL**	Port 0 Pull-up Control Port 1 Pull-up Control Port 2 Pull-up Control	11111100 *****011 *1111111
P0TYPE **P1TYPE** **P2TYPE**	Port 0 Type Control Port 1 Type Control Port 2 Type Control	00000000 *****000 *0000000
P0DIR **P1DIR** **P2DIR**	Port 0 Input/Output Control Port 1 Input/Output Control Port 2 Input/Output Control	11111111 *****111 *1111111
ALTSEL	Alternative Function Control	00000***
ADCON **ADCR** **ADCSEL** **ADCSELH**	ADC Control & ADC Result Low ADC Result High ADC Channel Selection Low and MUX Selection ADC High Channel Selection	00100000 00000000 11111111 11111111
EXIF **WDCON** **PMR** **STATUS** **LVDCFG**	Added External Interrupt and POR Control Watchdog Timer Control Power Management Control Crystal Status LVD Configuration	****0111 110*0000 ****0*** ***0**** 00000101
OSCICN **RINGCON** **OSCBIAS** **OSCREF**	Internal Precision Oscillator Control Internal Precision Oscillator Frequency Tuning Internal Precision Oscillator Bias Current Tuning Internal Precision Oscillator Reference Tuning	****0100 10000000 01011111 ***10111
IAPWEN **IAPCON** **IAPDAT**	IAP Write Enable IAP Control IAP Data	00000000 00000000 00000000

6.3. Instruction Set Summary

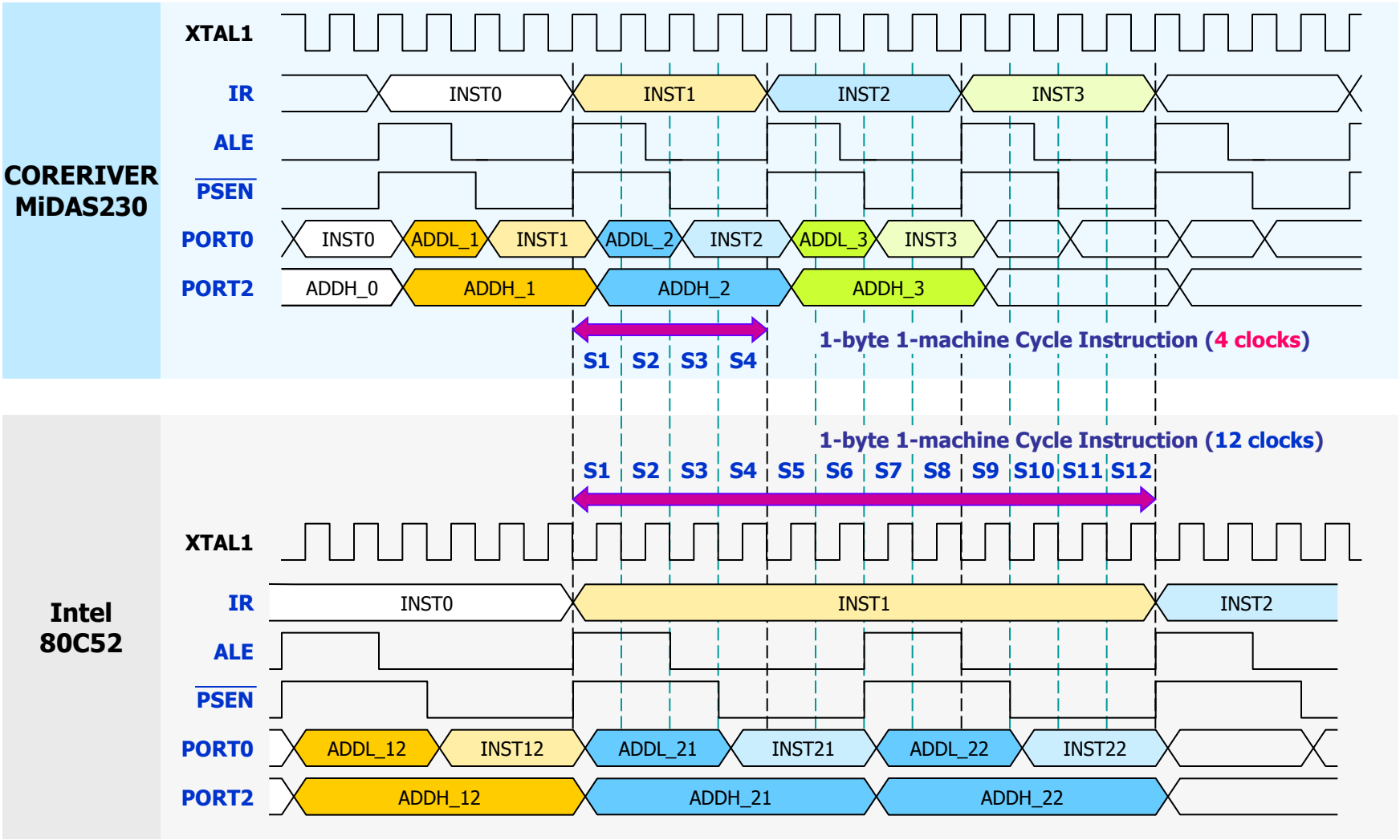
Preliminary

- Refer to Appendix A (Instruction Set) for more details.

TYP	Instruction	Description
Arithmetic	ADD	Addition
	ADDC	Addition with Carry
	SUBB	Subtraction with Borrow
	INC	Increment
	DEC	Decrement
	MUL	Multiply
	DIV	Divide
	DA	Decimal Adjust
Logical	ANL	AND
	ORL	OR
	XRL	Exclusive OR
	CLR	Clear
	CPL	Complement
	RL	Rotate Left
	RLC	Rotate Left with Carry
	RR	Rotate Right
	RRC	Rotate Right with Carry
	SWAP	Swap Nibbles
Data Transfer	MOV	Move Data
	MOVC	Move Code
	MOVX	Move Data to Ext. RAM
	PUSH	PUSH
	POP	POP
	XCH	Exchange
	XCHD	Exchange Low-digit

TYP	Instruction	Description
Boolean	CLR	Clear bit
	SETB	Set bit
	CPL	Complement bit
	ANL	AND bit
	ORL	OR bit
	MOV	Move bit
	JC	Jump if Carry is set
	JNC	Jump if Carry is not set
	JB	Jump if bit is set
	JNB	Jump if bit is not set
	JBC	Jump if bit is set & clear
Branch	ACALL	Absolute Call
	LCALL	Long Call
	RET	Return from Subroutine
	RETI	Return from Interrupt
	AJMP	Absolute Jump
	LJMP	Long Jump
	SJMP	Short Jump
	JMP	Jump with DPTR
	JZ	Jump if ACC is zero
	JNZ	Jump if ACC is not zero
	CJNE	Compare and Jump if not equal
	DJNZ	Decrement and Jump if not zero
	NOP	No Operation

- Comparative timing of the MiDAS230 family and Intel 80C52



6.4. CPU Timing : Execution Time Table

Preliminary

- The fastest instruction execution in the world

Instruction	**MiDAS230 (CORERIVER)**	W77C32 (Winbond)	DS80C320 (Maxim)	87C52 (Intel)
`MUL AB` `DIV AB`	**12 clocks**	20 clocks	20 clocks	48 clocks
`MOVC A, @A+PC` `MOVC A, @A+DPTR`	**8 clocks**	8 clocks	12 clocks	24 clocks
`JMP @A+DPTR`	**8 clocks**	8 clocks	12 clocks	24 clocks
`RET` `RETI`	**8 clocks**	8 clocks	16 clocks	24 clocks
`INC DPTR`	**4 clocks**	8 clocks	12 clocks	24 clocks
`Others`	**Same**	Same	Same	-

6.5. I/O Ports : PORT0[7:0]

Preliminary

- Open-drain or push-pull Output.
- The alternative functions are available only when the corresponding SFR bit is "1".
  - ✓ P0.0 = INT0B, PWM_A, TVO / PWM0.1 = ADC0, INT1B, ISP_SCL / P0.2 = ADC1, ISP_SDA / P0.3 = ADC2 / P0.4 = ADC3 / P0.5 = ADC4 / P0.6 = ADC5, PWM / P0.7 = ADC6
- Read-Modify-Write instructions do not read port pin but SFR register.
  - ✓ ANL / ORL / XRL / JBC / CPL / INC / DEC / DJNZ / MOV PX.Y, C / CLR PX.Y / SETB PX.Y

✓ **P0TYPE** (D4h) : Port 0 Type Control Register

P0TYPE.7	P0TYPE.6	P0TYPE.5	P0TYPE.4	P0TYPE.3	P0TYPE.2	P0TYPE.1	P0TYPE.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- 0 = Push-pull Output (Default) / 1 = Open-drain Output

✓ **P0DIR** (F4h) : Port 0 Input/Output Control Register

P0DIR.7	P0DIR.6	P0DIR.5	P0DIR.4	P0DIR.3	P0DIR.2	P0DIR.1	P0DIR.0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

- 0 = Output / 1 = Input (Default)

✓ **P0SEL** (E4h) : Port 0 Pull-up Control Register

P0SEL.7	P0SEL.6	P0SEL.5	P0SEL.4	P0SEL.3	P0SEL.2	P0SEL.1	P0SEL.0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(0)	R/W(0)

- 0 = Pull-up resistor ON / 1 = Pull-up resistor OFF

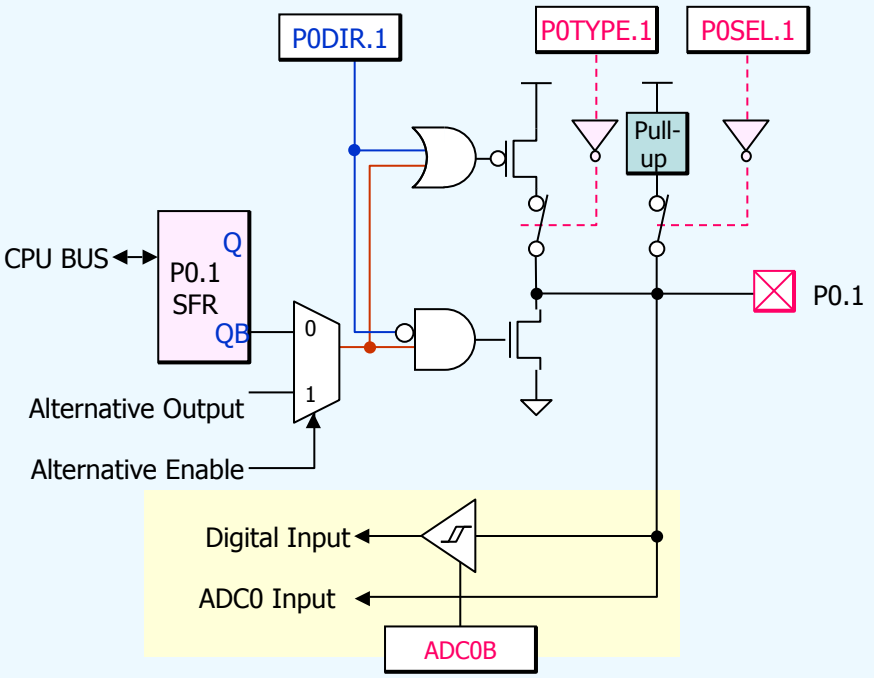
✓ **ADCSEL** (E2h) : ADC Channel Selection Low & MUX Selection

ADC3B	ADC2B	ADC1B	ADC0B	CH3	CH2	CH1	CH0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

- ADCXB = 0 : ADCX Input Enable & Digital Input Disable

✓ **P0** (80h) : Port 0 Register

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



6.5. I/O Ports : PORT1[1:0] (XTAL1/XTAL2)

Preliminary

- XTAL1/XTAL2 can be configured as I/O port.
- Read-Modify-Write instructions do not read port pin but SFR register.
  - ✓ ANL / ORL / XRL / JBC / CPL / INC / DEC / DJNZ / MOV PX.Y, C / CLR PX.Y / SETB PX.Y

✓ **P1TYPE** (D5h) : Port 1 TYPE Control Register

-	-	-	-	-	P1TYPE.2	P1TYPE.1	P1TYPE.0
					R/W(0)	R/W(0)	R/W(0)

- 0 = Push-pull Output (Default) / 1 = Open-drain Output

✓ **P1DIR** (F5h) : Port 1 Input/Output Control Register

-	-	-	-	-	P1DIR.2	P1DIR.1	P1DIR.0
					R/W(1)	R/W(1)	R/W(1)

- 0 = Output / 1 = Input (Default)

✓ **P1SEL** (E5h) : Port 1 Pull-up Control Register

-	-	-	-	-	P1SEL.2	P1SEL.1	P1SEL.0
					R/W(0)	R/W(1)	R/W(1)

- 0 = Pull-up resistor ON / 1 = Pull-up resistor OFF

✓ **P1** (90h) : Port 1 Register

-	-	-	-	-	P1.2	P1.1	P1.0
					R/W(1)	R/W(1)	R/W(1)

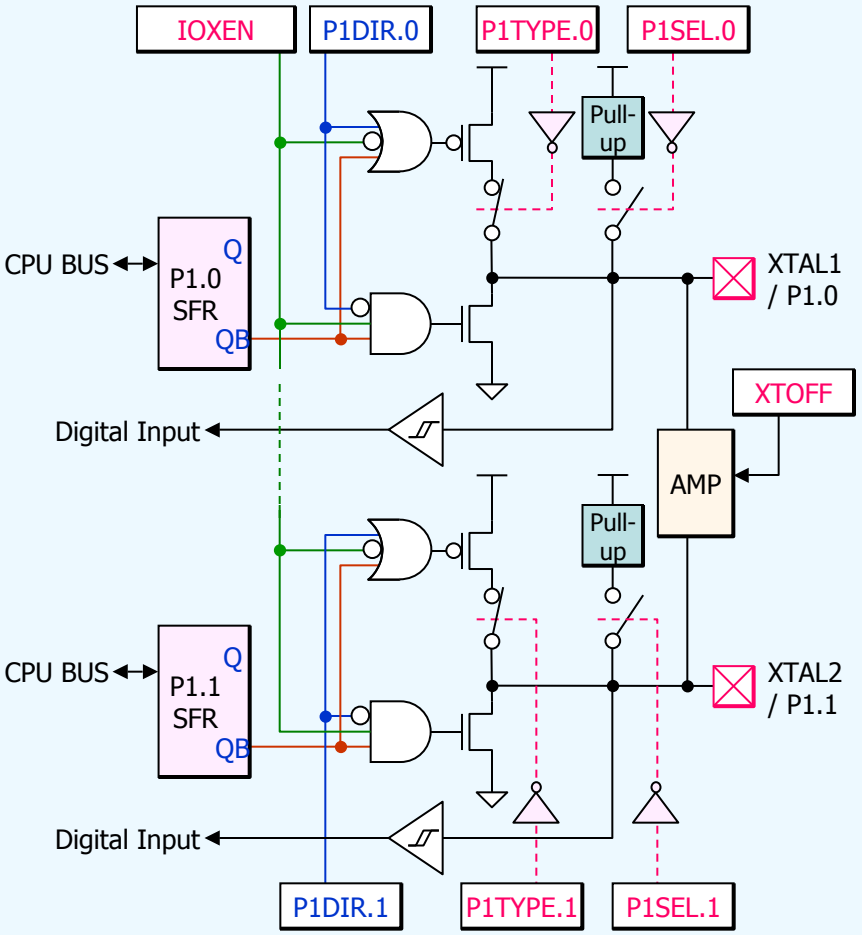
✓ **ALTSEL** (85h) : Alternative Function Control Register

IOXEN	IORSTEN	CLO	PWM00	TV0	-	-	-
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)			

- IOXEN = 1 : XTAL1 and XTAL2 are configured as I/O Ports

✓ **PMR** (C4h) : Power Management Control Register

-	-	-	-	XTOFF	-	-	-
				R/W(0)			



6.5. I/O Ports : PORT1[2] (RESETB)

Preliminary

- RESETB can be configured as I/O port.
- Read-Modify-Write instructions do not read port pin but SFR register.
  - ✓ ANL / ORL / XRL / JBC / CPL / INC / DEC / DJNZ / MOV PX.Y, C / CLR PX.Y / SETB PX.Y

✓ **P1TYPE** (D5h) : Port 1 TYPE Control Register

-	-	-	-	-	P1TYPE.2	P1TYPE.1	P1TYPE.0
					R/W(0)	R/W(0)	R/W(0)

- 0 = Push-pull Output (Default) / 1= Open-drain Output

✓ **P1DIR** (F5h) : Port 1 Input/Output Control Register

-	-	-	-	-	P1DIR.2	P1DIR.1	P1DIR.0
					R/W(1)	R/W(1)	R/W(1)

- 0 = Output / 1 = Input (Default)

✓ **P1SEL** (E5h) : Port 1 Pull-up Control Register

-	-	-	-	-	P1SEL.2	P1SEL.1	P1SEL.0
					R/W(0)	R/W(1)	R/W(1)

- 0 = Pull-up resistor ON / 1 = Pull-up resistor OFF

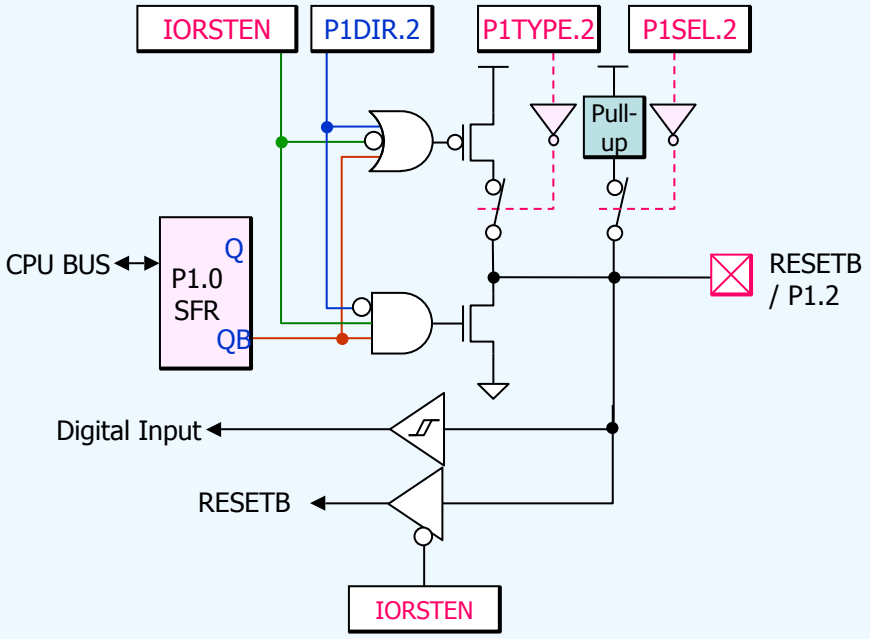
✓ **P1** (90h) : Port 1 Register

-	-	-	-	-	P1.2	P1.1	P1.0
					R/W(1)	R/W(1)	R/W(1)

✓ **ALTSEL** (85h) : Alternative Function Control Register

IOXEN	IORSTEN	CLO	PWM00	TV0	-	-	-
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)			

- IORSTEN = 1 : RESETB is configured as I/O port.



6.5. I/O Ports : PORT2[6:0]

Preliminary

- Open-drain or push-pull output.
- The alternative functions are available only when the corresponding SFR bit is "1".
  - ✓ P2.0 = T0 / P2.2 = ADC11 / P2.3 = ADC10 / P2.4 = ADC9 / P2.5 = ADC8 / P2.6 = ADC7, CLO
- Read-Modify-Write instructions do not read port pin but SFR register.
  - ✓ ANL / ORL / XRL / JBC / CPL / INC / DEC / DJNZ / MOV PX.Y, C / CLR PX.Y / SETB PX.Y

✓ **ADCSELH** (E1h) : ADC Channel Selection High Register

ADC11B	ADC10B	ADC9B	ADC8B	ADC7B	ADC6B	ADC5B	ADC4B
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

- ADCXB = 0 : ADCX Input Enable & Digital Input Disable

✓ **P2TYPE** (D6h) : Port 2 Type Control Register

-	P2TYPE.6	P2TYPE.5	P2TYPE.4	P2TYPE.3	P2TYPE.2	P2TYPE.1	P2TYPE.0
R/W(0))	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- 0 = Push-pull Output (Default) / 1= Open-drain Output

✓ **P2DIR** (F6h) : Port 2 Input/Output Control Register

-	P2DIR.6	P2DIR.5	P2DIR.4	P2DIR.3	P2DIR.2	P2DIR.1	P2DIR.0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

- 0 = Output / 1 = Input (Default)

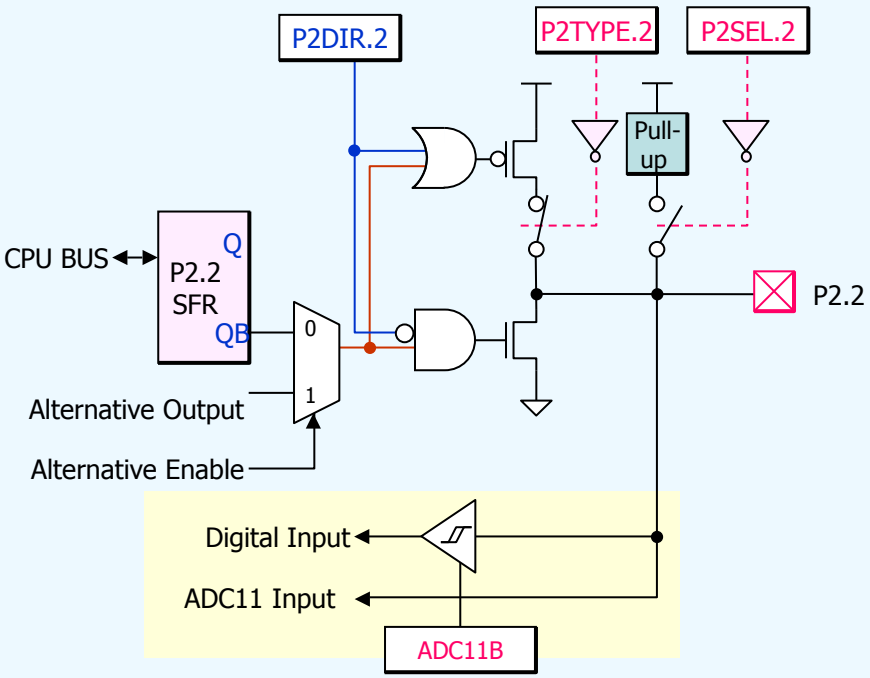
✓ **P2SEL** (E6h) : Port 2 Pull-up Control Register

-	P2SEL.6	P2SEL.5	P2SEL.4	P2SEL.3	P2SEL.2	P2SEL.1	P2SEL.0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

- 0 = Pull-up resistor ON / 1 = Pull-up resistor OFF (Default)

✓ **P2** (A0h) : Port 2 Register

-	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

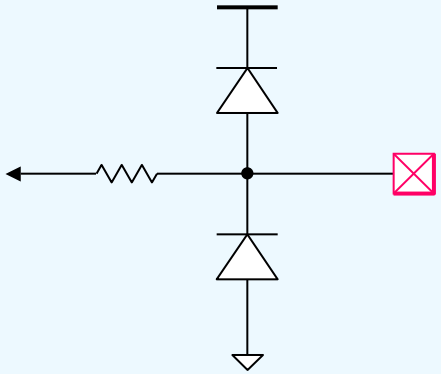


6.5. The ESD Structure of Pads

Preliminary

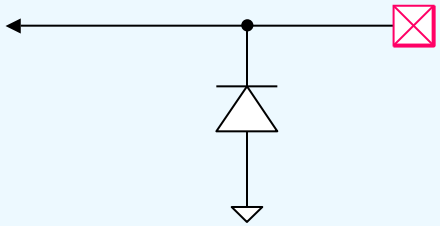
- Two ESD diodes and one ESD resister are contained in all pads except VDD.
- One ESD diode are contained in VDD.

[All pads except VDD]



- Two ESD Diodes (V_{DD} side, V_{SS} side)
- One ESD Resister

[VDD]



- One ESD Diode (GND side)
- One ESD Resister

6.6. POR (Power-On Reset)

Preliminary

- On-chip power-on reset
  - ✓ Generate reset when VDD is below 1.6V.
  - ✓ POR block is off during power-down mode if BGS is cleared.
- General Flags in PCON[3:2]
  - ✓ Cleared only by POR.
  - ✓ Not cleared by other resets.
  - ✓ Maybe used to distinguish cold start reset or warm start reset.

✓ **PCON** (87h) : Power Control Register

-	-	-	POF	GF1	GF0	PD	IDL
			R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

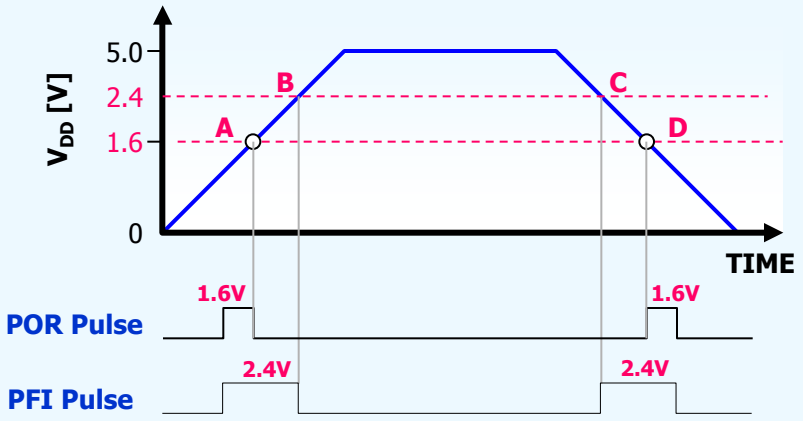
- POF : Power off flag.
  When power-on, this flag bit will be set by H/W.
- PD : Power-down (Stop) mode enable.

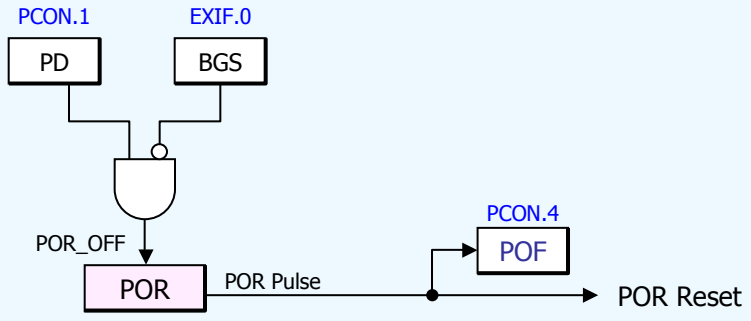
✓ **EXIF** (91h) : External Interrupt Flag Register

-				XT/RG	RGMD	RGSL	BGS
-				R/W(0)	R(1)	R(1)	R/W(1)

- BGS : Band-gap select (Default = 1).

  If 0, Band-gap block (POR) will be off in power-down mode
  If 1, Band-gap block (POR) will run in power-down mode

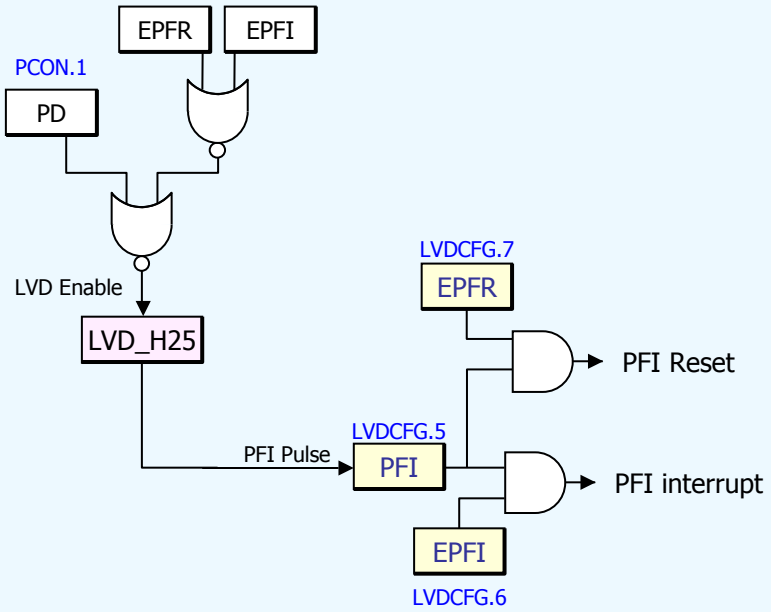




6.7. LVD (Configurable Low Voltage Detector)

Preliminary

- On-chip power-fail interrupt / Reset
  - ✓ PFI pulse is sampled at the end of an instruction (machine cycle if in IDLE mode).
  - ✓ LVD Reset or interrupt is generated if enabled and two continuous samples of PFI pulse are high.
- S/W Reset
  - ✓ User S/W may generate reset by setting PFI bit two times continuously when EFPR bit is set.
- LVD Configuration
  - ✓ Default setting is for 2.4V.
  - ✓ Factory tuning value should be updated by user S/W.
  - ✓ Refer to the LVD application note.



✓ **LVDCFG** (C6h) : LVD Configuration Register

EPFR	EPFI	PFI	CFG4	CFG3	CFG2	CFG1	CFG0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R(1)	R/W(0)	R/W(1)

- EPFR : Power-fail reset enable.
- EPFI : Power-fail interrupt enable.
- PFI : Power-fail interrupt flag.
- CFG[4:0] : LVD Voltage Selection

NOTE : This SFR is initialized only by power-on-reset.
That is, it holds user's setting for any other reset.

CFG[4:0]	V_{LVD} [V]	CFG[4:0]	V_{LVD} [V]
1 1111	5.0	0 1111	3.4
1 1110	4.9	0 1110	3.3
1 1101	4.8	0 1101	3.2
1 1100	4.7	0 1100	3.1
1 1011	4.6	0 1011	3.0
1 1010	4.5	0 1010	2.9
1 1001	4.4	0 1001	2.8
1 1000	4.3	0 1000	2.7
1 0111	4.2	0 0111	2.6
1 0110	4.1	0 0110	2.5
1 0101	4.0	**0 0101**	**2.4**
1 0100	3.9	0 0100	2.3
1 0011	3.8	0 0011	2.2
1 0010	3.7	0 0010	2.1
1 0001	3.6	0 0001	2.0
1 0000	3.5	0 0000	1.9

6.8. WDT (Watch Dog Timer)

Preliminary

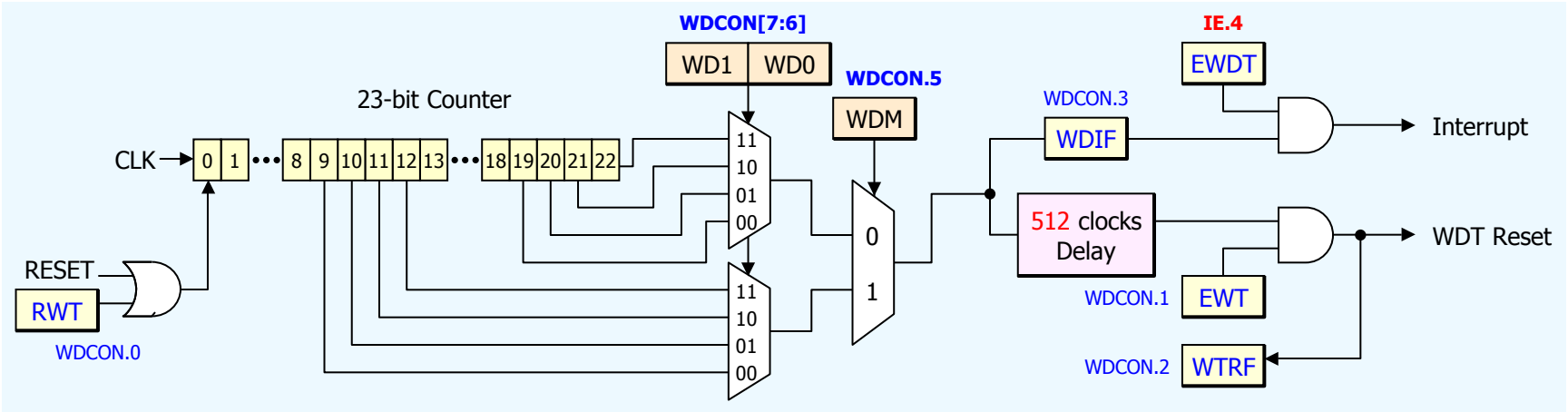
- Detects software upset due to external noise or other causes
- Allows an automatic recovery using WDT interrupt
- If enabled, WDT interrupt or WDT reset makes MCU wake up from stop mode.
- Watchdog time-out values

WDM	WD1	WD0	Interrupt Time-out (@ 4MHz)		Reset Time-out
0	0	0	2^{19} clocks	131 ms	2^{19} + 512 clocks
	0	1	2^{20} clocks	262 ms	2^{20} + 512 clocks
	1	0	2^{21} clocks	524 ms	2^{21} + 512 clocks
	1	1	2^{22} clocks	1048 ms	2^{22} + 512 clocks
1	0	0	2^{9} clocks	128 us	2^{9} + 512 clocks
	0	1	2^{10} clocks	256 us	2^{10} + 512 clocks
	1	0	2^{11} clocks	512 us	2^{11} + 512 clocks
	1	1	2^{12} clocks	1024 us	2^{12} + 512 clocks

✓ **WDCON** (D8h) : Watchdog Timer & Power Status Register

WD1	WD0	WDM	-	WDIF	WTRF	EWT	RWT
R/W(1)	R/W(1)	R/W(0)	R(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- WD[1:0] : WDT Clock Divide
- WDM : Watchdog clock divide mode
- WDIF : Watchdog Timer Interrupt Flag
- WTRF : Watchdog Timer Reset Flag
- EWT : Watchdog Timer Reset Enable
- RWT : Restart Watchdog Timer



6.9. Timer/Counter : Timer 0/1

Preliminary

- Compatible with traditional 80C52 Timer/Counter function
- Time base is 12 clocks.

Mode / Timer	Mode 0 (M1,M0=00)	Mode 1 (M1,M0=01)	Mode 2 (M1,M0=10)	Mode 3 (M1,M0=11)
Timer0	13-bit T/C	16-bit T/C	8-bit T/C with automatic reload (TL0 ← TH0)	8-bit T/C (TL0) ➔ Timer0 interrupt 8-bit T/C (TH0) ➔ Timer1 interrupt
Timer1	Not Supported		8-bit T/C with automatic reload (TL1 ← TH1)	Not Supported

✓ **TMOD** (89h) : Timer/Counter 0 Mode Control Register

-	-	-	-	GATE	C/T	M1	M0
				R/W(0)	R/W(0)	R/W(0)	R/W(0)

- GATE : Timer 0 Gate control. When TRx (in TCON) is set and GATE=1, Timer x will run only while INTx pin is high (hardware control). When GATE=0, Timer x will run only while TRx=1 (software control).
- C/T[2] : Timer 0 Counter/Timer Select. 0 = Timer by $F_{OSC}/12$. (Default) 1 = Counter by T0 pin.
- M1, M0 : Timer 0 Mode Select.
  [0,0] : Mode 0. 13-bit T/C.
  [0,1] : Mode 1. 16-bit T/C.
  [1,0] : Mode 2. 8-bit T/C with automatic reload
  [1,1] : Mode 3. Two 8-bit T/C

✓ **TCON** (88h) : Timer/Counter 0/1 Control Register

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- TF1 : Timer 1 Overflow Flag.
- TR1 : Timer 1 Run Enable.
- TF0 : Timer 0 Overflow Flag.
- TR0 : Timer 0 Run Enable.
- IE1 : External Interrupt 1 Flag.
- IT1 : External Interrupt 1 TYP Select Flag. Edge Detect (IT1=1). Level Detect (IT1=0).
- IE0 : External Interrupt 0 Flag.
- IT0 : External Interrupt 0 TYP Select Flag. Edge Detect (IT0=1). Level Detect (IT0=0).

✓ **TL0** (8Ah) : Timer/Counter 0 Low Byte Register

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

✓ **TH0** (8Ch) : Timer/Counter 0 High Byte Register

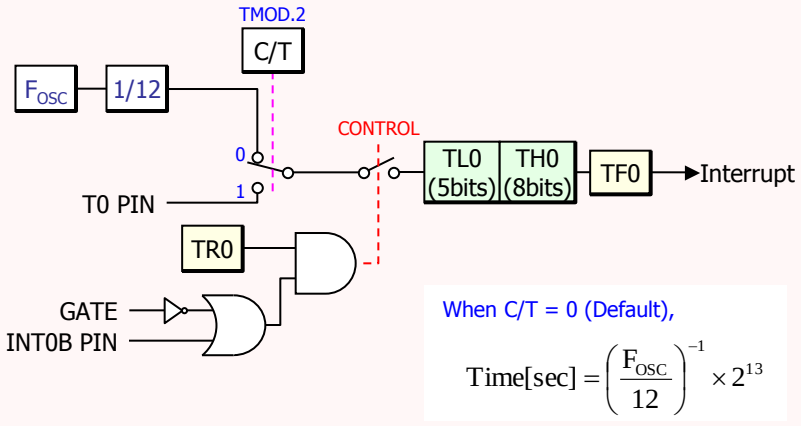
TH0.7	TH0.6	TH0.5	TH0.4	TH0.3	TH0.2	TH0.1	TH0.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

✓ **TL1** (8Bh) : Timer/Counter 1 Low Byte Register

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

✓ **TH1** (8Dh) : Timer/Counter 1 High Byte Register

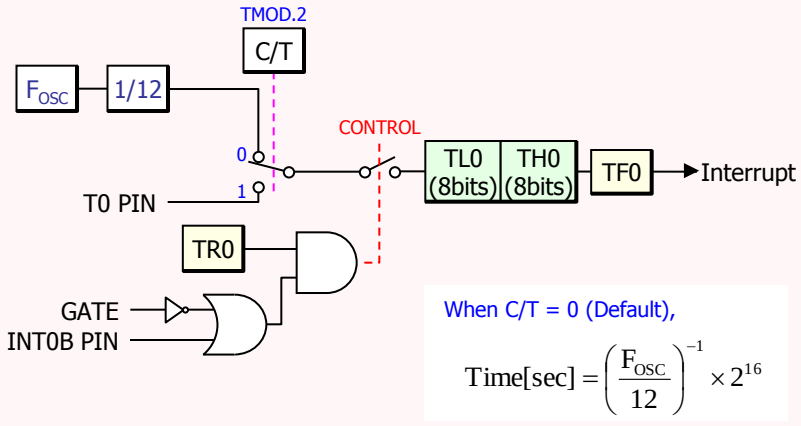
TH1.7	TH1.6	TH1.5	TH1.4	TH1.3	TH1.2	TH1.1	TH1.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)



When C/T = 0 (Default),

$$\text{Time[sec]} = \left(\frac{F_{OSC}}{12}\right)^{-1} \times 2^{13}$$

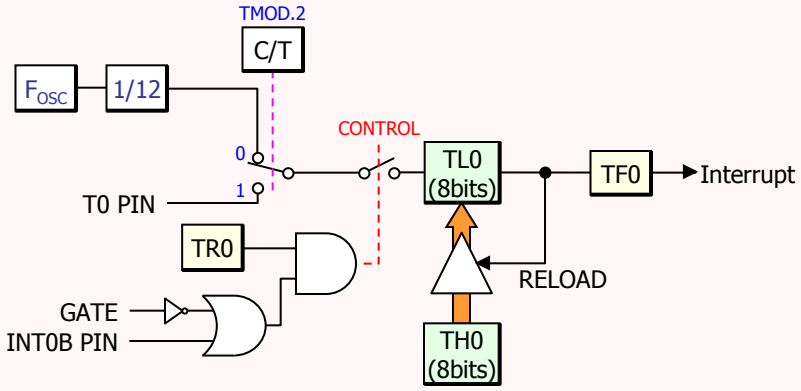
[Mode 0]



When C/T = 0 (Default),

$$\text{Time[sec]} = \left(\frac{F_{OSC}}{12}\right)^{-1} \times 2^{16}$$

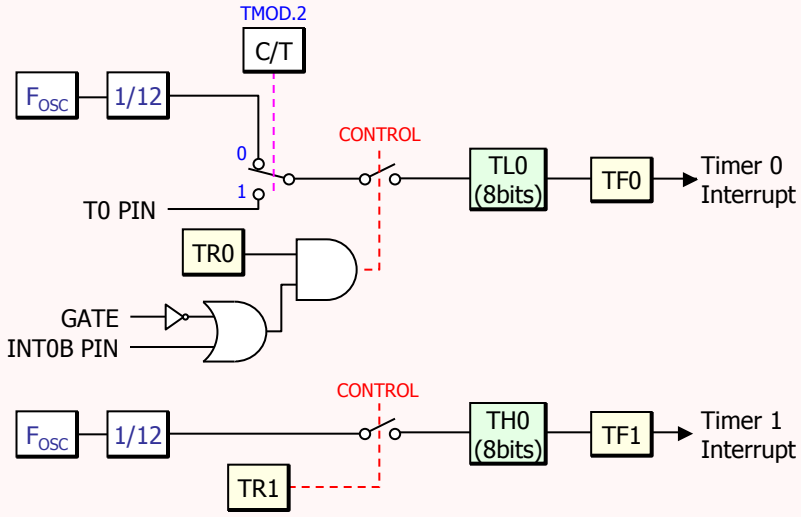
[Mode 1]



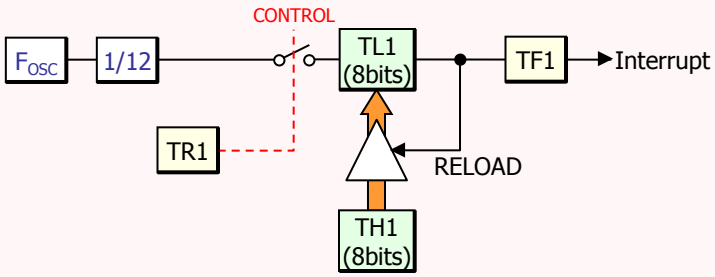
When C/T = 0 (Default),

$$\text{Time[sec]} = \left(\frac{F_{OSC}}{12}\right)^{-1} \times \left(2^{8} - \text{TH0}\right)$$

[Mode 2]



[Mode 3]

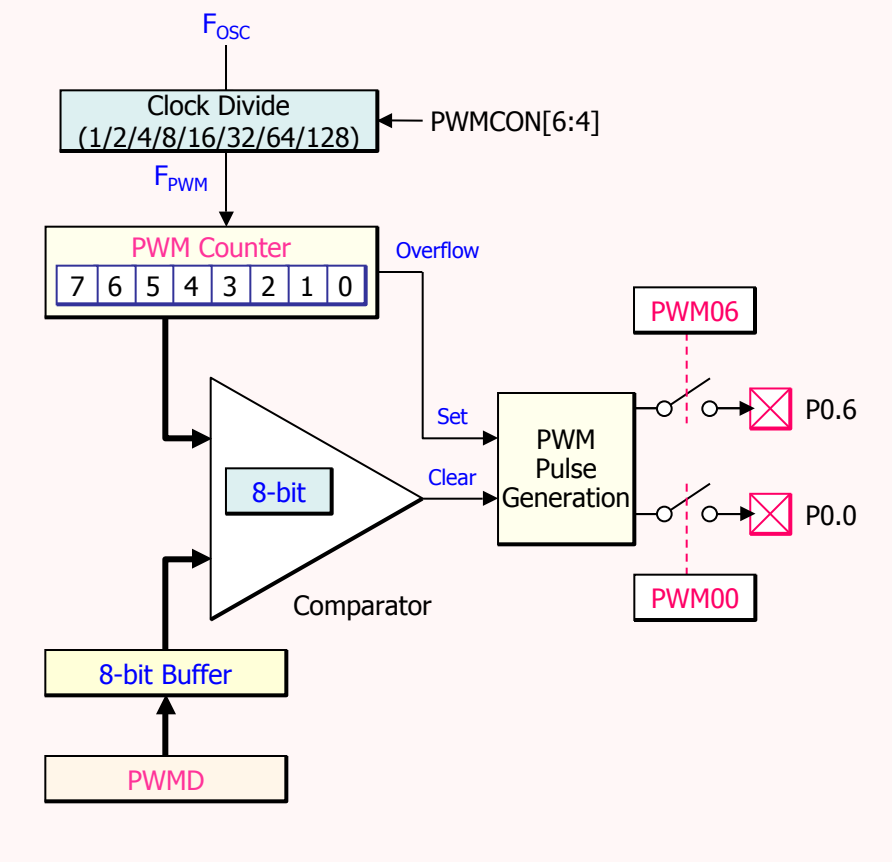

CONTROL
F_{OSC}
1/12
TL1
(8bits)
TF1
Interrupt
TR1
RELOAD
TH1
(8bits)

[Mode 2]

6.10. PWM (Pulse Width Modulator)

Preliminary

- Intelligent 1-channel 8-bit PWM
- PWM Counter Reload Mode (8-bit Counter Overflow Reload)
- PWM Counter can be cleared by S/W.
- PWM is stopped or started (resumed) by S/W.



✓ **PWMCON** (DCh) : PWM Control Register

PWM06	PS2_P0	PS1_P0	PS0_P0	-	PWMF	CLR_P0	RUN_P0
R/W(0)	R/W(0)	R/W(0)	R/W(0)		R/W(0)	R/W(0)	R/W(0)

- PWM06 : PWM Waveform Output Enable to P0.6
- PS2_P0, PS1_P0, PS0_P0 : Pre-scaled Clock Selection.
  [0,0,0] = $F_{OSC}/1$, [0,0,1] = $F_{OSC}/2$, [0,1,0] = $F_{OSC}/4$,
  [0,1,1] = $F_{OSC}/8$, [1,0,0] = $F_{OSC}/16$, [1,0,1] = $F_{OSC}/32$,
  [1,1,0] = $F_{OSC}/64$, [1,1,1] = $F_{OSC}/128$
- PWMF : PWM Interrupt Flag. Cleared by S/W
- CLR_P0 : Counter Reset Enable. Cleared by H/W.
- RUN_P0 : Counter Start Enable.

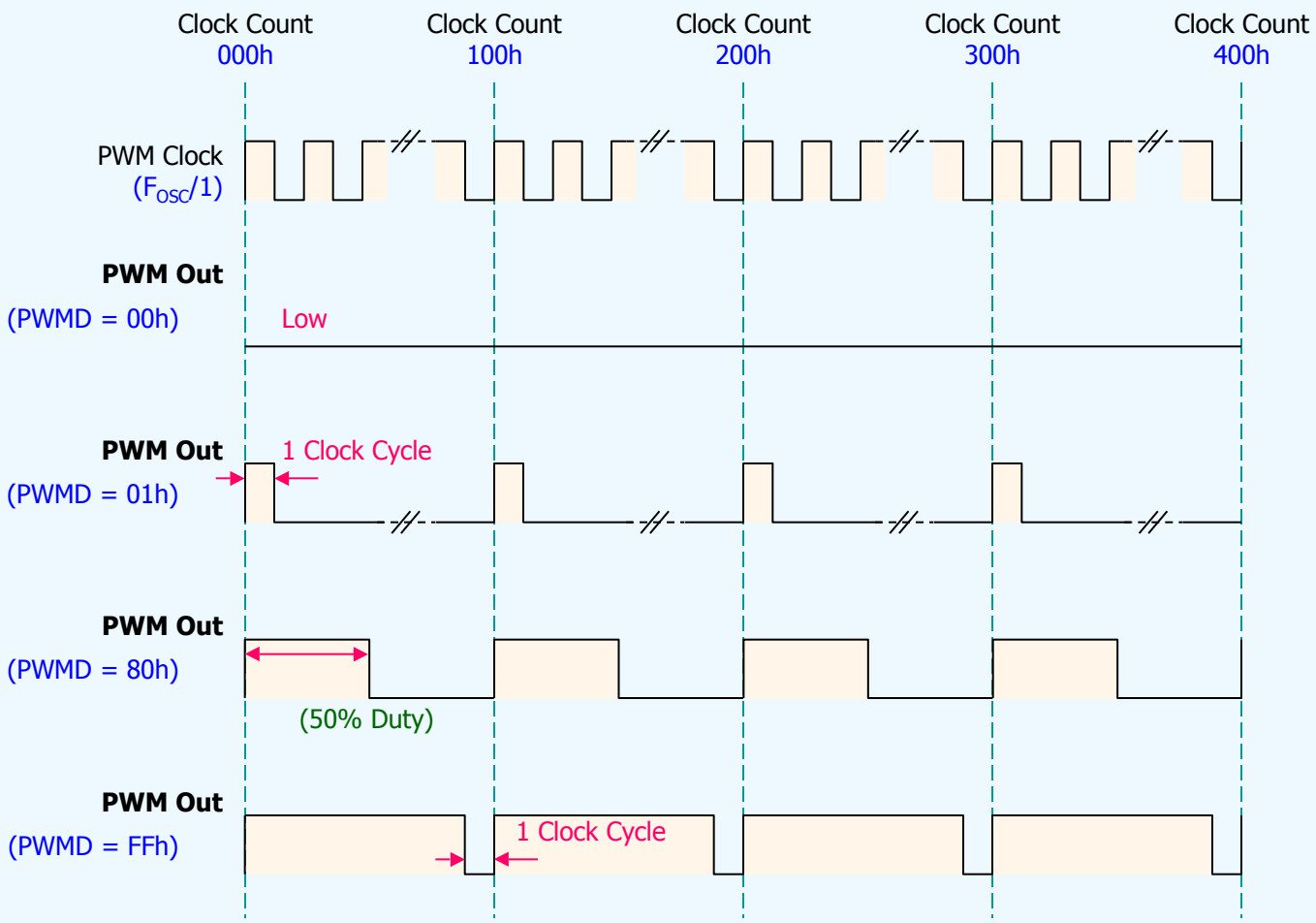
✓ **PWMD** (DEh) : PWM Duty Data Register

PWMD.7	PWMD.6	PWMD.5	PWMD.4	PWMD.3	PWMD.2	PWMD.1	PWMD.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

✓ **ALTSEL** (E3h) : Alternative Function Control Register

IOXEN	IORSTEN	CLO	PWM00	TV0	-	-	-
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)			

- PWM00 : PWM Waveform Output Enable to P0.0


Clock Count
000h
Clock Count
100h
Clock Count
200h
Clock Count
300h
Clock Count
400h
PWM Clock
(FOSC/1)
PWM Out
(PWMD = 00h)
Low
PWM Out
(PWMD = 01h)
1 Clock Cycle
PWM Out
(PWMD = 80h)
(50% Duty)
PWM Out
(PWMD = FFh)
1 Clock Cycle

6.11. ADC (Analog-to-Digital Converter)

Preliminary

- 12-channel 10-bit ADC (SAR Type)
- Max. 120k SPS @F_{ADC} = 12MHz (F_{SYS} = 12MHz @+3.3V)

✓**ADCSELH** (E1h) : ADC Channel Selection High Register

ADC11B	ADC10B	ADC9B	ADC8B	ADC7B	ADC6B	ADC5B	ADC4B
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

✓ **ADCSEL** (E2h) : ADC Channel Selection Low & MUX Selection Register

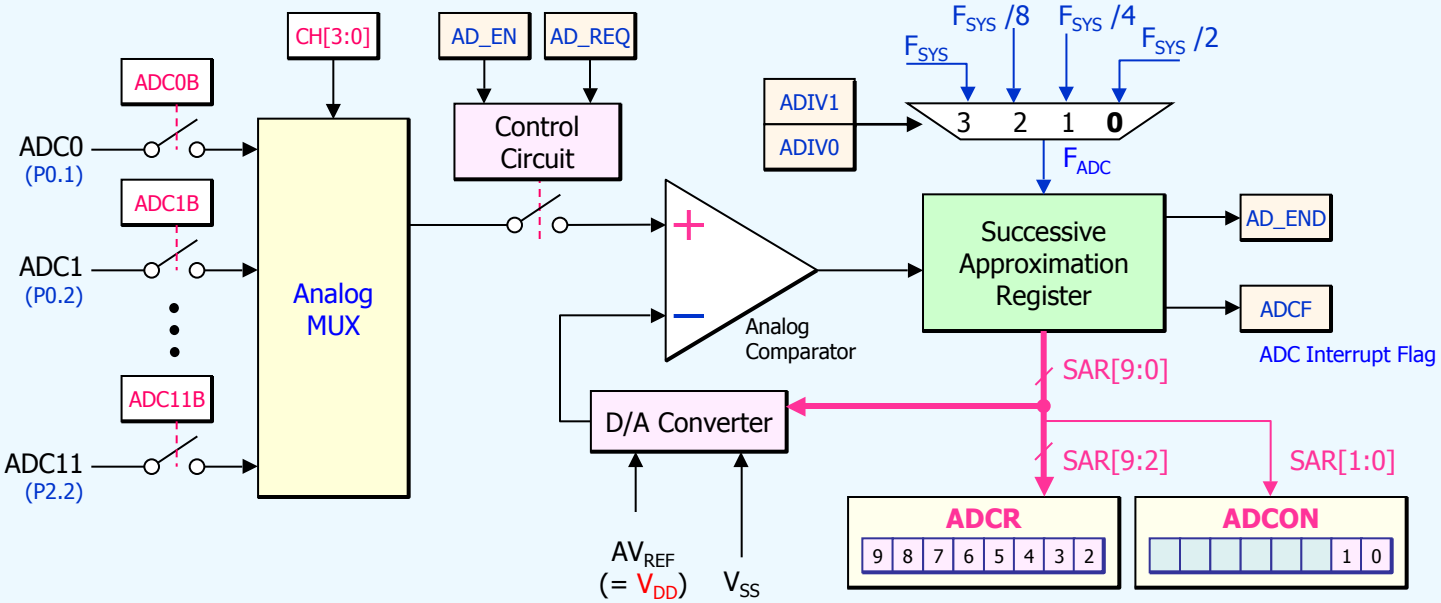
ADC3B	ADC2B	ADC1B	ADC0B	CH3	CH2	CH1	CH0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

✓**ADCON** (EFh) : ADC Control & ADC Result Low Register

AD_EN	AD_REQ	AD_END	ADCF	ADIV1	ADIV0	SAR1	SAR0
R/W(0)	R/W(0)	R(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

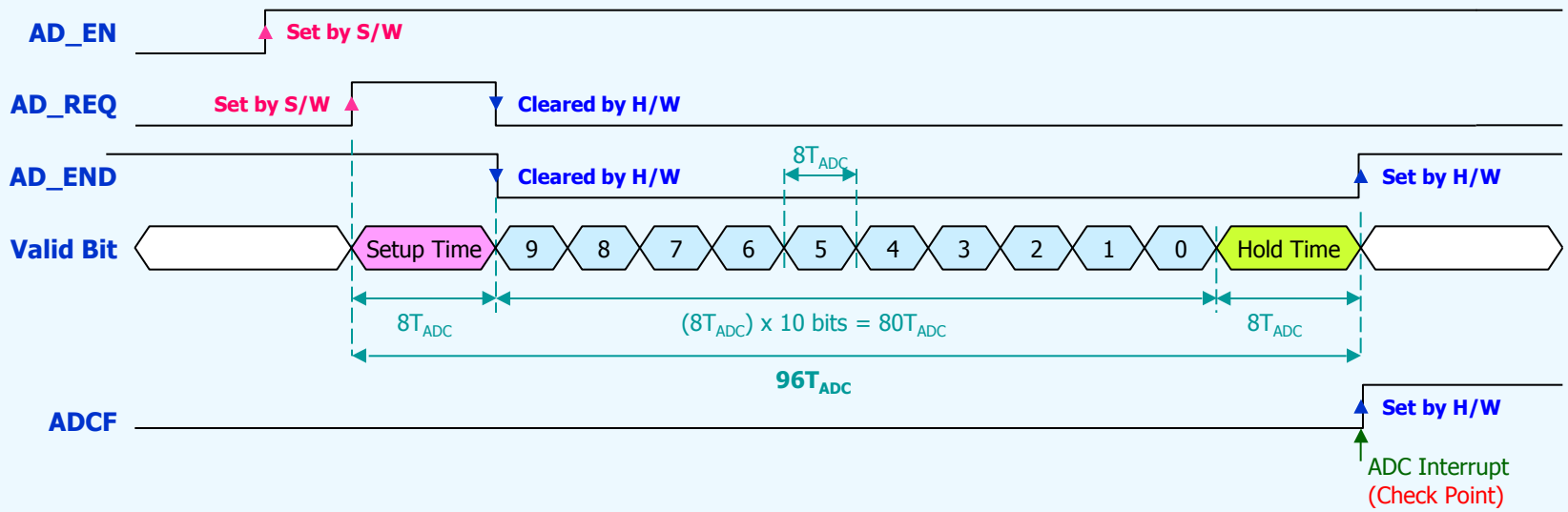
✓ **ADCR** (EEh) : ADC Result High Register

SAR9	SAR8	SAR7	SAR6	SAR5	SAR4	SAR3	SAR2
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)



6.11. ADC : Conversion Timing

Preliminary



- ✓ **AD_EN** : AD Conversion Enable Signal.
  Set or Cleared by S/W.
- ✓ **AD_REQ** : AD Conversion Request Bit.
  Set by S/W and Cleared by H/W.
  This bit must be set at each sample conversion.
- ✓ **AD_END** : Set or Cleared by H/W.
  Clear when Conversion started.
  Set when Conversion ended.
- ✓ **ADCF** : AD Conversion Interrupt Flag.
  Set by H/W and Cleared by S/W.
  A User should clear ADCF bit in ADC interrupt routine.
  A User must check the ADCF flag instead of AD_END.

[An Example of ADC Conversion Table]

System Clock (F_{SYS})	Divide (ADIV=0)	F_{ADC}	T_{ADC} ($1/F_{ADC}$)	1 Sample Conversion Time
24MHz @ 5V	$F_{SYS}/2$	12MHz	83.3 ns	8 us
12MHz	$F_{SYS}/2$	6MHz	166.6 ns	16 us
6MHz	$F_{SYS}/2$	3MHz	333.3 ns	32 us
4MHz	$F_{SYS}/2$	2MHz	500.0 ns	48 us

6.12. Interrupt : 7 Sources / 2-level Priority

Preliminary

- 7 Interrupt Sources
  - ✓ Timer 0/1, ADC, WDT, PWM, 2 External.
- 3-level Interrupt Priority

[Interrupt Vector Address]

HIGH

PRIORITY

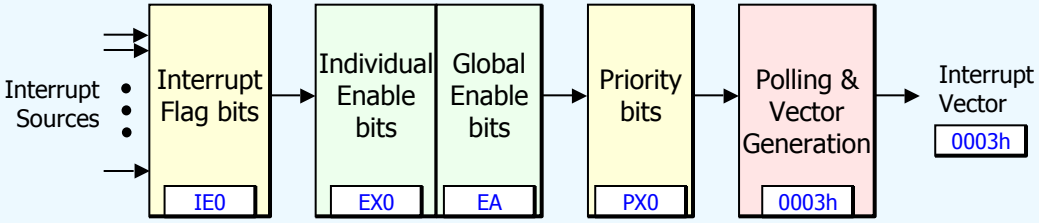
LOW

Interrupt Sources	Address	Priority Level
LVD	0033h	NMI
INT0B	0003h	2 Levels
TF0	000Bh	2 Levels
INT1B	0013h	2 Levels
TF1	001Bh	2 Levels
WDT	0023h	2 Levels
PWM	002Bh	2 Levels
ADC	003Bh	2 Levels

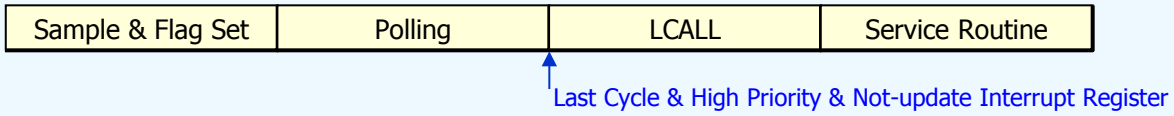
*** Interrupt related to SFR (refer to Appendix B : SFR Description)**

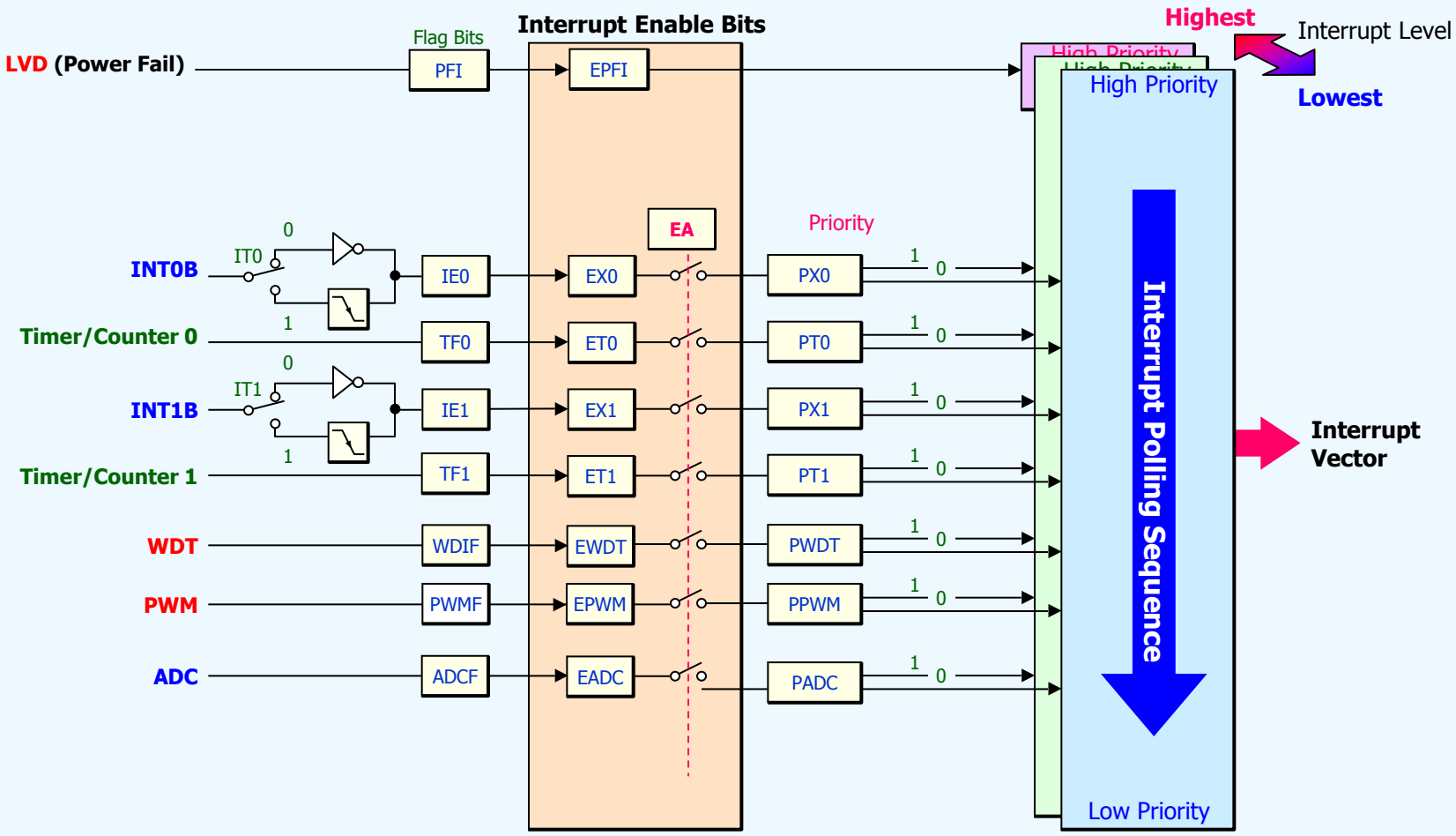
✓ **TCON** (88h)	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
✓ **IE** (A8h)	EA	EADC	EPWM	EWDT	ET1	EX1	ET0	EX0
✓ **IP** (B8h)	-	PADC	PPWM	PWDT	PT1	PX1	PT0	PX0
✓ **LVDCFG** (C6h)	EPFR	EPFI	PFI	CFG4	CFG3	CFG2	CFG1	CFG0
✓ **WDCON** (D8h)	WD1	WD0	WDM	-	WDIF	WTRF	EWT	RWT
✓ **PWMCON** (DCh)	PWM06	PS2_P0	PS1_P0	PS0_P0	-	PWMF	CLR_P0	RUN_P0
✓ **ADCON** (EFh)	AD_EN	AD_REQ	AD_END	ADCF	ADIV1	ADIV0	SAR1	SAR0

[Interrupt Vector Generation Flow]



[Response Sequence]




Flag Bits
Interrupt Enable Bits
Highest
Interrupt Level
Lowest
LVD (Power Fail)
PFI
EPFI
High Priority
EA
Priority
INT0B
IT0
IE0
EX0
PX0
Timer/Counter 0
TF0
ET0
PT0
INT1B
IT1
IE1
EX1
PX1
Timer/Counter 1
TF1
ET1
PT1
WDT
WDIF
EWDT
PWDT
PWM
PWMF
EPWM
PPWM
ADC
ADCF
EADC
PADC
Interrupt Polling Sequence
Interrupt Vector
Low Priority

6.13. Reset Circuit : Four Reset Sources

Preliminary

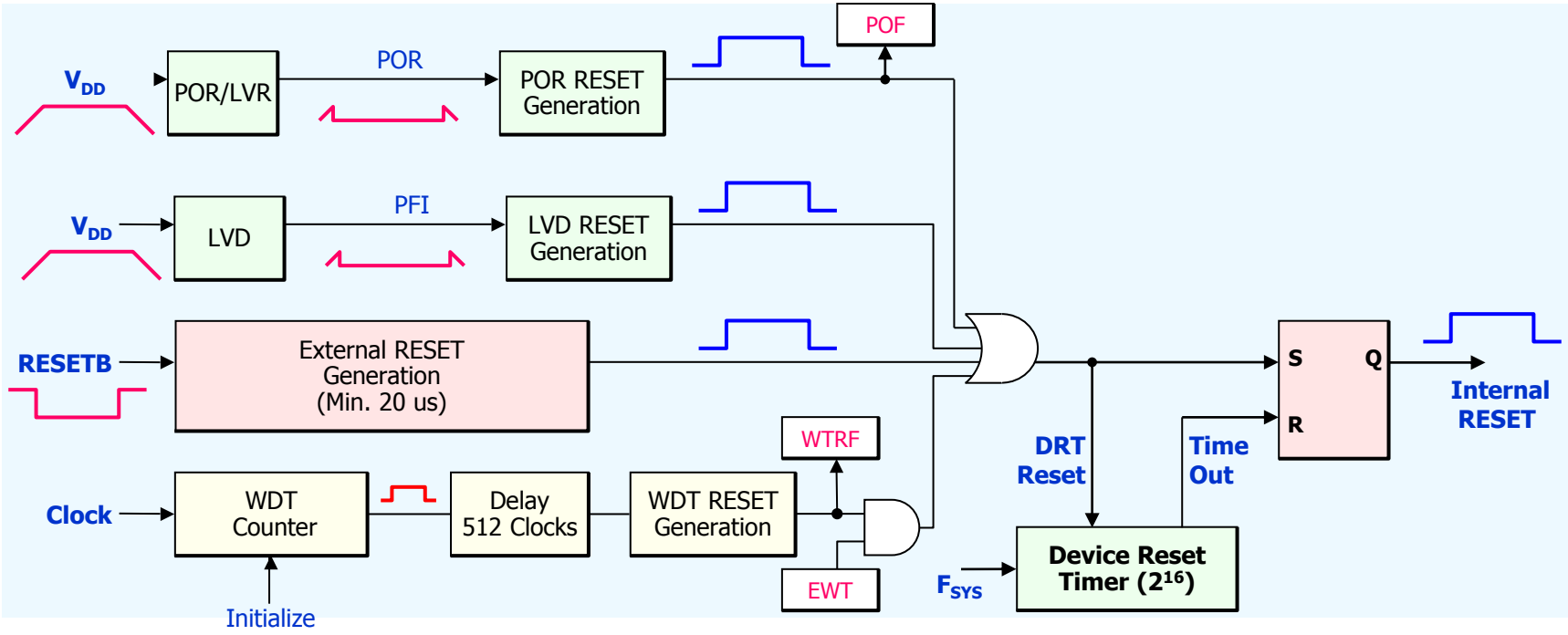
- Reset Sources
  - ✓ POR reset : Power-on Reset (POR) / Power-fail Reset
  - ✓ LVD reset : Configurable Reset Voltage
  - ✓ External Reset : RESETB must be low for 20 us or more.
  - ✓ WDT reset : Optional control by S/W.

- Device Reset Timer
  - ✓ Once set, internal reset remains high until the DRT (Device Reset Timer) is expired.
  - ✓ The reset time is about 16 ms.

✓ **WDCON** (D8h) : Watchdog Timer Control Register

WD1	WD0	WDM	-	WDIF	WTRF	EWT	RWT
R/W(1)	R/W(1)			R/W(0)	R/W(0)	R/W(0)	R/W(0)

- WTRF : Watchdog Timer Reset Flag. Only cleared by S/W.
- EWT : Watchdog Timer Reset Enable.



6.14. Clock Circuit

Preliminary

- Two System Clock Sources
  - ✓ Internal Precision Oscillator
  - ✓ External Crystal Oscillator
- Default System Clock is internal Precision OSC.
- User need to check XTUP flag before set XT/RG bit.
- Fast Wake-up from Power-down Mode using internal OSC.

Control Flag			System Clock	Status Bit	
XT/RG	XTOFF	RINGON		RGMD	XTUP
1	0	X	Crystal OSC.	0	1
0	X	1	Precision OSC.	1	0/1

✓ **EXIF** (91h) : External Interrupt Flag Register

-	-			XT/RG	RGMD	RGSL	BGS
				R/W(0)	R(1)	R(1)	R/W(1)

✓ **OSCICN** (BEh) : Internal Oscillator Control Register

-	-	-	-	DIV2	RINGON	DIV1	DIV0
				R/W(0)	R/W(1)	R/W(0)	R/W(0)

✓ **STATUS** (C5h) : Crystal Status Register

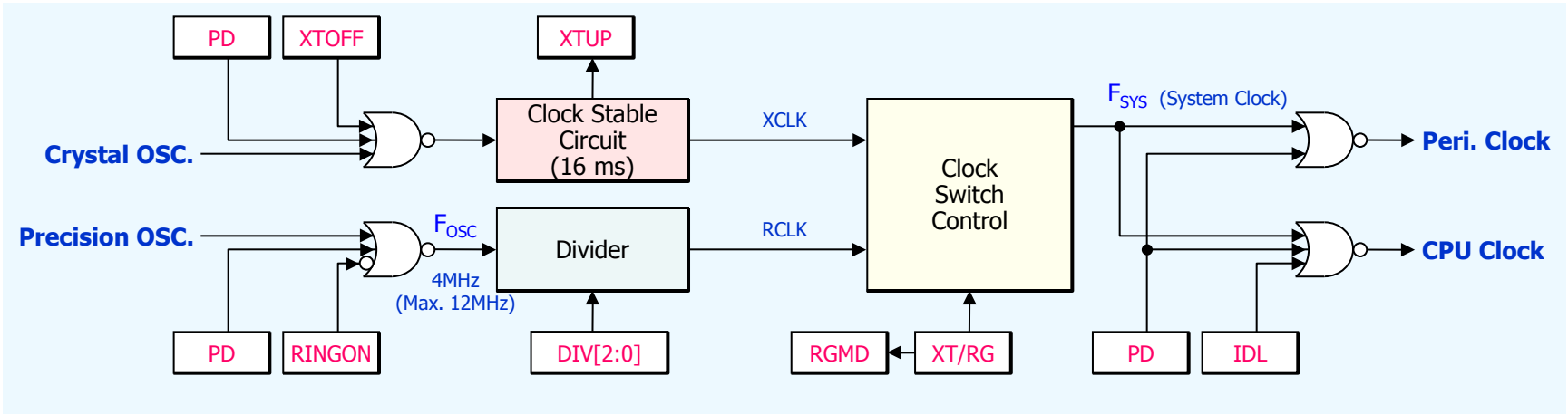
-	-	-	XTUP	-	-	-	-
			R(0)				

✓ **PMR** (C4h) : Power Management Control Register

-	-	-	-	XTOFF	-	-	-
				R/W(0)			

✓ **PCON** (87h) : Power Control Register

-	-	-	POF	GF1	GF0	PD	IDL
			R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)



- Internal Precision Oscillator with Calibration Function
  - ✓ The default internal clock (RCLK) is 4MHz.
  - ✓ The calibration values are saved in OTP area and should be copied to the tuning registers by user S/W (Refer to IAP example).

✓ **OSCICN** (BEh) : Internal Oscillator Control Register

-	-	-	-	DIV2	RINGON	DIV1	DIV0
				R/W(0)	R/W(1)	R/W(0)	R/W(0)

- RINGON : 1 = Internal oscillator(12MHz) is running.
  0 = Internal oscillator is off.
  Don't clear RINGON bit when XTRG = 0.
- DIV2, DIV1, DIV0 : Internal oscillator divider.
  [0,0,0] = 4MHz ; Default
  [0,0,1] = 2MHz
  [0,1,0] = 1MHz
  [0,1,1] = 0.5MHz
  [1,0,0] = reserved
  [1,0,1] = 12MHz
  [1,1,0] = 6MHz
  [1,1,1] = 3MHz

✓ **RINGCON**(95h) : Internal Oscillator Frequency Tuning

S7	S6	S5	S4	S3	S2	S1	S0
R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

✓ **OSCBIAS**(96h) : Internal Oscillator Bias Current Tuning

S7	S6	S5	S4	S3	S2	S1	S0
R/W(0)	R/W(1)	R/W(0)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

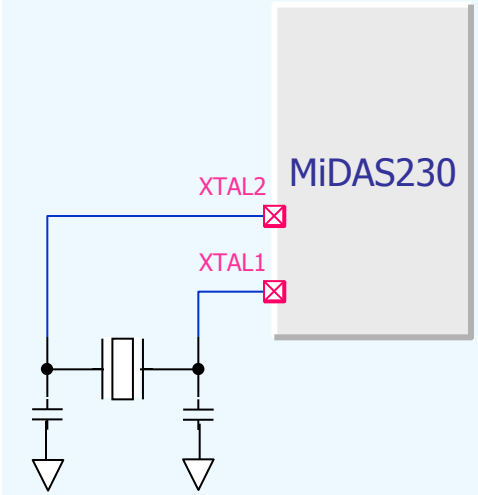
✓ **OSCREF**(97h) : Internal Oscillator Reference Tuning

-	-	-	S4	S3	S2	S1	S0
			R/W(1)	R/W(0)	R/W(1)	R/W(1)	R/W(1)

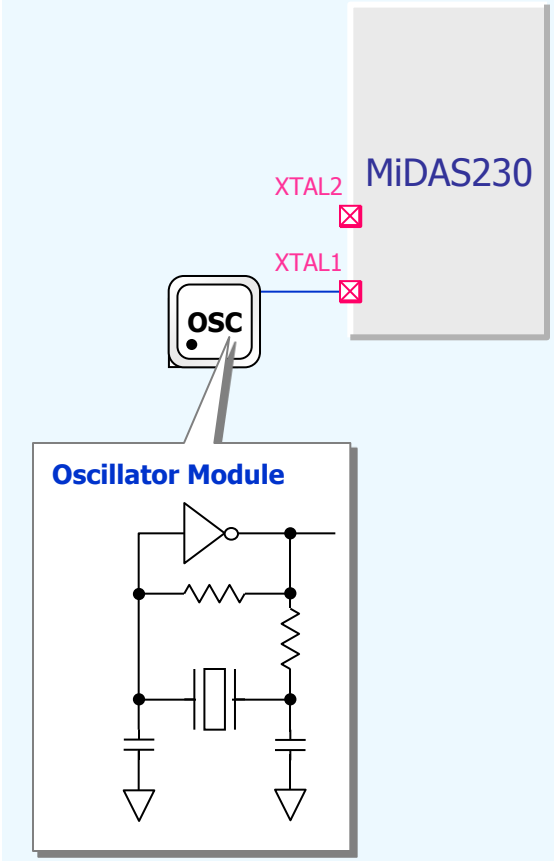
6.14. Clock Circuit : Guideline for Configuration

Preliminary

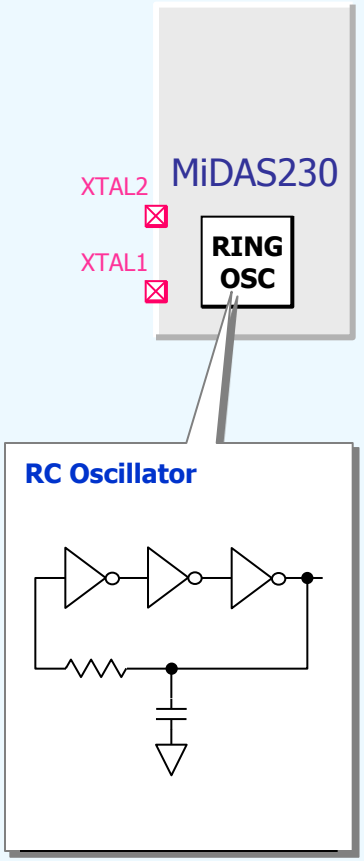
◆ **Crystal Oscillator**



◆ **Oscillator Module**



◆ **Internal Ring Oscillator**



6.15. Power Management : 3 Modes

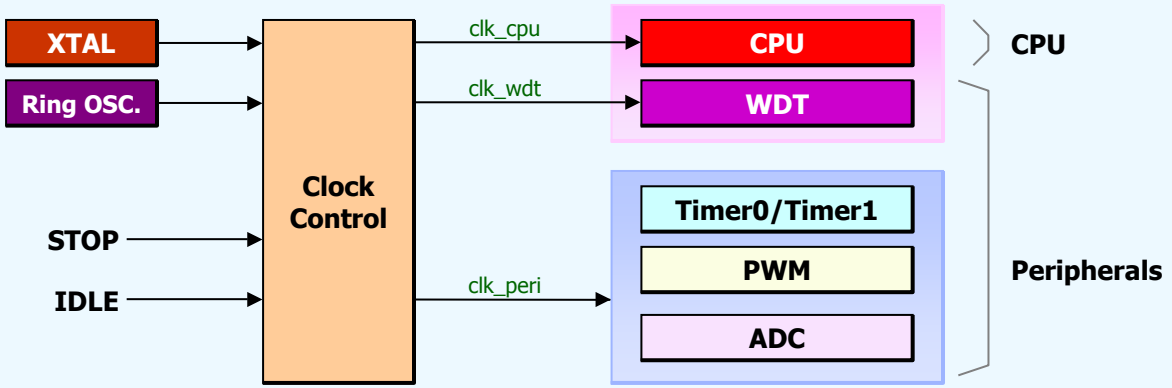
Preliminary

- **Active Mode** : CPU and Peripheral are running.
- **Idle Mode** : Only Peripherals are running.
  - ✓ Wake-up from all kinds of interrupts. CPU continues.
  - ✓ Wake-up from all kinds of resets. CPU restarts.
- **Stop Mode 1/2** : CPU and Peripheral will stop.
- **Stop Mode 1** : When WDT is off
  - ✓ Wake-up from all kinds of external interrupt (level detect).
    → MCU continues.
  - ✓ Wake-up external reset.
    → MCU restarts.
- **Stop Mode 2** : When WDT is on
  - ✓ Wake-up from all kinds of external interrupt (level detect), or WDT interrupt. → MCU continues.
  - ✓ Wake-up from all kinds of resets. (RESETB or WDT reset)
    → MCU restarts.

✓ **PCON** (87h) : Power Control Register

-	-	-	POF	GF1	GF0	PD	IDL
			R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

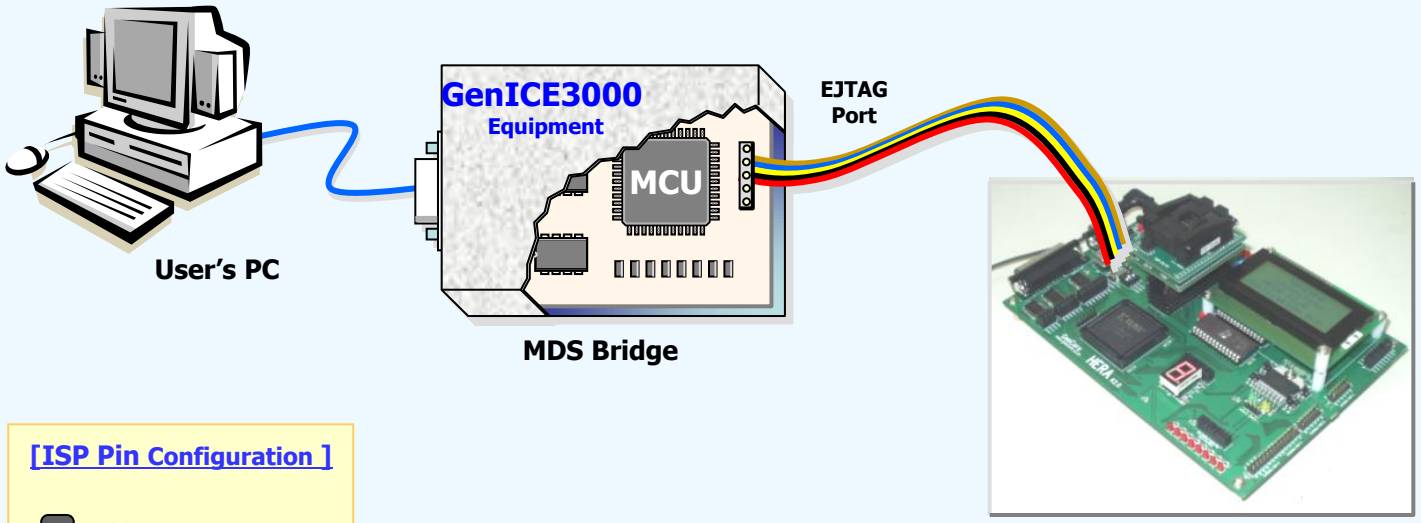
- PD : Stop Mode (Power-down) Enable.
- IDL : IDLE Mode Enable.



6.16. ISP (In System Programming)

Preliminary

- Code memory (2kBytes) can be programmed using EJTAG in target system.
  - ✓ FLASH : 0x0000 ~ 0x07FF (2,048 Bytes)
- EEPROM (128Bytes) can be programmed using EJTAG in target system.
  - ✓ EEPROM : 0x0780 ~ 0x07FF (128 Bytes)



[ISP Pin Configuration]

VCC
GND
RESETB
ISP_SCL (P0.1)
ISP_SDA (P0.2)

6.16. ISP : Command Set

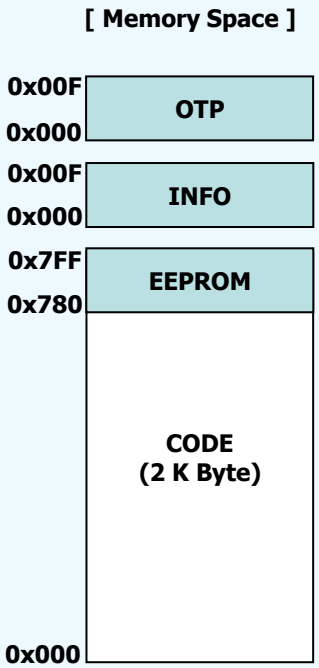
Command	Function
Blank	◆ Check the blank status of the device currently connected.
Erase Chip	◆ Performs an erase chip, the device's memory, both code and data. • Code : FLASH • User data : EEPROM • Information data : Lock bits, RING option, PGM/ERS time (ISP)
	◆ The device will be blank and in a programmable state.
Read Code/EEPROM	◆ Reads in the device's memory.
	◆ The results from the read are loaded into the CORERIVER ISP software's buffer and displayed on the screen.
Write Chip/EEPROM	◆ Writes all memory locations in the CORERIVER ISP software's buffer out to the device's memory.
Verify Chip	◆ Compares the CORERIVER ISP software buffer with the device's internal memory.
	◆ If the buffers are found to be exact replicas of the device's memory, a success result is returned.
	◆ If there are any differences, a failure result is returned along with the total number of mismatched bytes.

6.17. IAP (In Application Programming)

Preliminary

- In Application Programming
  - ✓ User S/W can read or modify specific regions of FLASH memory with IAP function during operation.
  - ✓ CPU is halt during IAP and continues execution after IAP from the next instruction which set IAPCON.
  - ✓ It takes 8 system clocks to read a byte with IAP.
  - ✓ It takes about 2 ms to write(erase) a byte with IAP, which may yields a delay of interrupt service.
  - ✓ It is recommended to disable any interrupt before IAP write or erase.

- Memory Address Space
  - ✓ CODE : Program memory
    - EEPROM is a part of CODE memory, which can be read as a data memory with MOVX.
  - ✓ INFO : Information memory.
    - INFO[0] contains the lock bits.
    - Erased only by the full chip erase of ISP.
  - ✓ OTP : Fabrication specific memory.
    - OTP contains the calibration data for Precision OSC. and others.
    - Not erased by the full chip erase of ISP.
    - User S/W should not modify the data in this memory.



6.17. IAP (In Application Programming)

IAP Related SFR

- ✓ DPH / DPL : Address for IAP.
- ✓ IAPDAT: 8-bit data buffer for read or write by IAP.
- ✓ IAPCON : IAP control SFR. Automatically cleared to zero after IAP is done.
- ✓ IAPWEN : A protection code (0xAC35) should be written to this address to enable IAP write(erase).

✓ **IAPWEN** (FCh) : IAP Write/Erase Enable Register

WEN.7	WEN.6	WEN.5	WEN.4	WEN.3	WEN.2	WEN.1	WEN.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

✓ **IAPDAT** (FEh) : IAP read/write Data Register

DAT.7	DAT.6	DAT.5	DAT.4	DAT.3	DAT.2	DAT.1	DAT.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

✓ **DPL** (82h) : Data Pointer Low Register

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

✓ **DPH** (83h) : Data Pointer High Register

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

✓ **IAPCON** (FDh) : IAP Control SFR

-	-	-	-	RGS1	RGS0	OPS1	OPS0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- **RGS[1:0] : Select IAP region**

RGS1	RGS0	IAP Region
0	0	CODE (0x000 ~ 0x7FF)
0	1	Reserved
1	0	INFO (0x0 ~ 0xF)
1	1	OTP (0x0 ~ 0xF)

- **OPS[1:0] : Select IAP function**

OPS1	OPS0	IAP Function
0	0	No operation
0	1	Byte Read : IAPDAT = F[DPTR]
1	0	Byte Erase : F[DPTR] = 0
1	1	Byte Write : F[DPTR] = IAPDAT

6.17. IAP (In Application Programming)

Preliminary

- Electrical Characteristic of IAP
  - ✓ Note that the program time depends on the frequency of CPU clock.
  - ✓ If the clock frequency is out of IAP range, user should use the internal RING clock for IAP write.

Parameter	Symbol	MIN	TYP	MAX	Unit
Power Supply Voltage	V_{DD}	2.7	**3.3**	5.5	V
CPU Clock Frequency	F_{SYS}	3	**4**	5	MHz
Write / Erase Time	Tp	1.5	**2.0**	3.3	ms

- Enable condition of IAP write(erase)
  - ✓ The protection code (0xAC35) should be written to 16-bit hidden write enable register through IAPWEN.
  - ✓ WDT provides program timing during IAP. WDM bit in WDCON should not be set.

- Lock Bits (The first byte of INFO memory)

-	-	-	-	-	-	-	LB0

  - ✓ LB0 : If set, disable ISP except the full chip erase.

[Example Code : IAP Program]

```
// Erase and write a byte in CODE/EEPROM area.
// This example is a generalized version.
// Some of the back-up/restore procedure may be omitted according to users condition.
char code_erase_write (unsigned int addr, unsigned char w_data)
{
            unsigned char b_exif;      // Back up of EXIF
            unsigned char b_osc;       // Back up of OSCICN
            unsigned char b_ie;        // Back up of IE
            unsigned char b_wdc;       // Back up of WDCON

            b_ie   = IE;               // Save IE
            EA     = 0;                // Disable interrupt for atomic access
            b_osc  = OSCICN;           // Save OSCICN
            OSCICN = 0x04;             // RCLK is 4 MHz.
            b_exif = EXIF;             // Save EXIF
            EXIF  &= 0xF7;             // XTRG = 0. Select RCLK.
            b_wdc  = WDCON;            // Save WDCON
            WDCON &= 0xDF;             // Clear WDM

            DPH    = (addr >> 8);      // High byte of the address
            DPL    = (addr & 0xFF);    // Low  byte of the address
            IAPWEN = 0xAC;             // High byte of the write enable
            IAPWEN = 0x35;             // Low  byte of the write enable
            IAPCON = 0x02;             // Erase byte. About 2ms
            IAPDAT = w_data;           // Data to be written
            IAPCON = 0x03;             // Write byte. About 2ms
            IAPCON = 0x01;             // Read byte
            IAPWEN = 0;                // Clear the write enable register

            WDCON  = b_wdc;            // Restore WDCON
            OSCICN = b_osc;            // Restore RCLK divider
            EXIF   = b_exif;           // Restore clock selection
            IE     = b_ie;             // Restore interrupt
            if (IAPDAT != w_data)
                        return (1);    // Write fail
            else        return (0);    // Write OK
}
```

[Example Code : IAP OTP Read]

```
// Read the calibration data for Precision Oscillator
void init_posc(void)
{
            DPH     = 0;
            DPL     = 2;            // Low  byte of the address
            IAPCON  = 0x0D;         // Read byte in OTP
            RINGCON = IAPDAT;       // Read result

            DPL    += 2;            // Increment the address
            IAPCON  = 0x0D;         // Read byte in OTP
            OSCBIAS = IAPDAT;       // Read result

            DPL    += 2;            // Increment the address
            IAPCON  = 0x0D;         // Read byte in OTP
            OSCREF  = IAPDAT;       // Read result
}
```

7. Absolute Maximum Ratings

Preliminary

Items	Conditions	Ranges
Voltage on any pin relative to Ground	-	-0.5V to (V_{DD}+0.5V)
Voltage in V_{DD} relative to Ground	-	-0.5V to 6.5V
Output Voltage	-	-0.5V to (V_{DD}+0.5V)
Output Current High	One I/O pin active	-25mA
	All I/O pin active	-100mA
Output Current Low	One I/O pin active	+30mA
	All I/O pin active	+150mA
Storage Temperature	-	-65 °C to +150 °C
Soldering Temperature	-	260 °C for 10 seconds

8. DC Characteristics

Preliminary

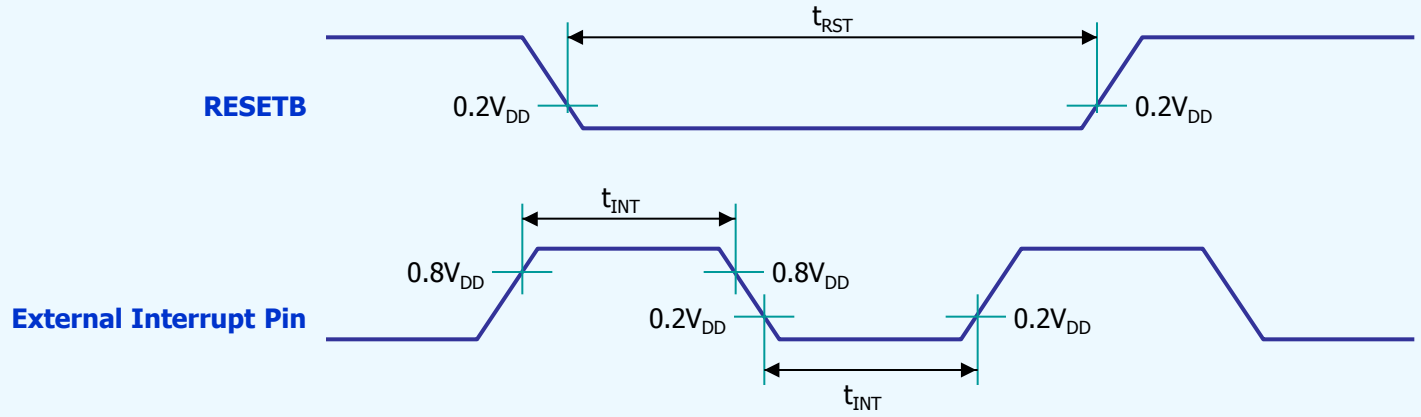
*** TA = = -40 oC ~ +85 oC, VDD = +2.4V to + 5.5V unless otherwise specified.**

Parameter	Symbol	Pin	Conditions	Value			Unit
				Min.	Typ.	Max.	
Input Low Voltage	V_{IL1}	RESETB ,P0, P1, P2	V_{DD} = 2.4V~5.5V	-0.5	-	$0.2V_{DD}$-0.1	V
	V_{IL2}	XTAL1, XTAL2		-0.5	-	$0.3V_{DD}$	
Input high Voltage	V_{IH1}	RESETB, P0, P1 ,P2	V_{DD} = 2.4V~5.5V	$0.2V_{DD}$+1.0	-	V_{DD}+0.5	V
	V_{IH2}	XTAL1, XTAL2		$0.7V_{DD}$	-	V_{DD}+0.5	
Output Low Voltage	V_{OL}	All Pins	V_{DD}= +4.4V to +5.5V (I_{OL} = -16.12mA) V_{DD}= +3.3V to +4.4V (I_{OL} = -9.23mA) V_{DD}= +2.4V to +3.3V (I_{OL} = -3.83mA)	-	-	$0.3V_{DD}$	V
Output High Voltage	V_{OH}	All Pins	V_{DD}= +4.4V to +5.5V (I_{OH} = -13.61mA) V_{DD}= +3.3V to +4.4V (I_{OH} = -7.61mA) V_{DD}= +2.4V to +3.3V (I_{OH} = -2.88mA)	$0.7V_{DD}$	-	-	V
	V_{OHP}	ALL Pins (Pull-up Resistor Only)	V_{DD}= +4.4V to +5.5V (I_{OHP} = -43.2uA) V_{DD}= +3.3V to +4.4V (I_{OHP} = -31.94uA) V_{DD}= +2.4V to +3.3V (I_{OHP} = -20.53uA)	$0.7V_{DD}$	-	-	V
Input Leakage Current	I_{IL}	All Pins Except of XTAL1, XTAL2	V_{IN} = V_{IH} or V_{IL}	-	-	±1	μA
Stop Current	I_{STOP}	-	V_{DD} = +5.0V (All Clock OFF)	-	0.05 (50nA)	1	μA
Pin Capacitance	C_{IO}	All Pins	V_{DD} = 5V	-	10	-	pF

9. AC Characteristics

*** TA = -40 °C ~ +85 °C unless otherwise specified.**

Parameter	Symbol	Pin	Conditions	Value			Unit
				Min.	Typ.	Max.	
Operating Frequency	F_{SYS}	Internal Oscillator XTAL1, XTAL2	V_{DD} = 5V ± 10%	1	-	12	MHz
			V_{DD} = 3V ± 10%	1	-	12	
RESETB Input Width	t_{RST}	RESETB	V_{DD} = 5V ± 10%	24	-	-	F_{SYS}
			V_{DD} = 3V ± 10%	24	-	-	
External Interrupt Input Width	t_{INT}	External Interrupt	V_{DD} = 5V ± 10%	4	-	-	F_{SYS}
			V_{DD} = 3V ± 10%	4	-	-	



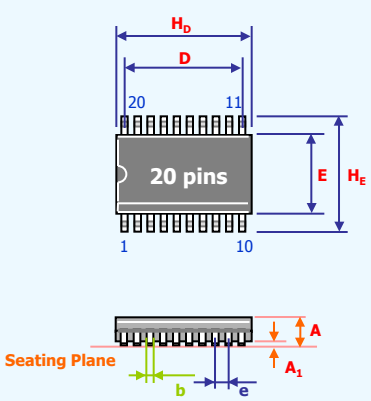
10. ADC Specifications

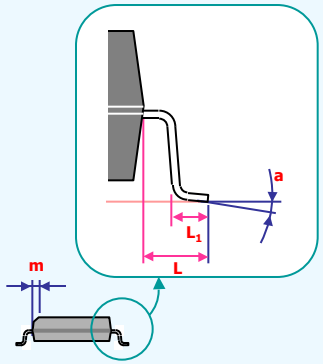
Preliminary

Parameter		Symbol	Conditions	Value			Unit
				Min.	Typ.	Max.	
Supply Voltage		V_{DDADC}	-	1.6	-	5.5	V
Input Voltage		V_{INADC}	-	V_{SS}	-	V_{DD}	V
Resolution		RES_{ADC}	-	-	10	-	bit
Operating Frequency		F_{ADC}	V_{DD} = 4.5V ~ 5.5V V_{DD} = 2.4V ~ 3.3V	-	-	10 5	MHz
Conversion Time		t_{ADC}	-	-	96 / F_{ADC}	-	s
Overall Accuracy		OA_{ADC}	V_{DD} =5V, F_{ADC}=10MHz V_{DD} =3V, F_{ADC}=5MHz	-	±2	±4	LSB
Integral Nonlinearity		INL_{ADC}	V_{DD} =5V, F_{ADC}=10MHz V_{DD} =3V, F_{ADC}=5MHz	-	±2	±4	LSB
Differential Nonlinearity		DNL_{ADC}	V_{DD} =5V, F_{ADC}=10MHz V_{DD} =3V, F_{ADC}=5MHz	-	±0.5	±1	LSB
Zero Input Error		ZIE_{ADC}	V_{DD} =5V, F_{ADC}=10MHz V_{DD} =3V, F_{ADC}=5MHz	-	±2	±4	LSB
Full Scale Error		FSE_{ADC}	V_{DD} =5V, F_{ADC}=10MHz V_{DD} =3V, F_{ADC}=5MHz	-	±2	±4	LSB
Analog Input Capacitance		C_{INADC}	-	-	10	15	pF
ADC Current	Active	I_{ADC}	V_{DD} = 5V, F_{ADC}=10MHz	-	1	2	mA
			V_{DD} = 3V, F_{ADC}=5MHz	-	0.3	0.6	
	Power-down		V_{DD} = 5V	-	-	100	nA

11. Package Dimensions : 20-SOP/16-TSSOP

Preliminary



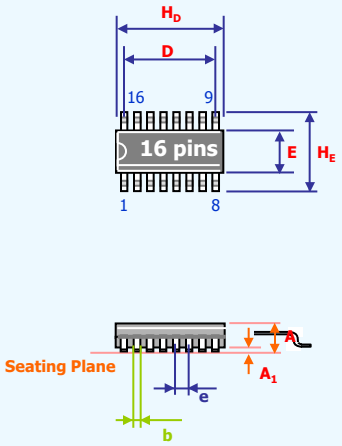


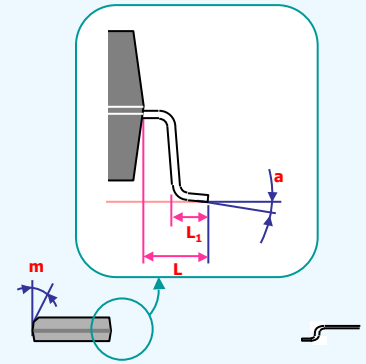
[20-SOP]

Symbol	Dimension in Inches			Dimension in mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
A	0.093	0.099	0.104	2.35	2.45	2.65
A_1	0.004	0.008	0.012	0.10	0.20	0.30
b	0.014	0.016	0.019	0.35	0.42	0.49
D	-	0.450	-	-	11.43	-
E	0.291	0.295	0.299	7.40	7.50	7.60
H_D	0.496	0.504	0.512	12.60	12.80	13.00
H_E	0.404	0.411	0.419	10.26	10.45	10.65
L	0.057	0.058	0.060	1.43	1.48	1.53
L_1	0.034	0.038	0.042	0.86	0.96	1.07
a	0°	-	8°	0°	-	8°
e	0.050 BSC			1.27 BSC		
m	0.020	0.025	0.030	0.50	0.62	0.75

Notes:

1. Dimension D & E include mold mismatch and are determined at the mold parting line.
2. General appearance spec. should be based on final visual inspection spec.





[16-TSSOP]

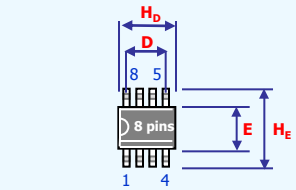
Symbol	Dimension in Inches			Dimension in mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
A	0.037	0.039	0.041	0.95	1.00	1.05
A_1	0.015	0.017	0.019	0.3865	0.4365	0.4865
b	0.008	0.009	0.009	0.20	0.22	0.24
D	0.176	0.179	0.182	4.47	4.55	4.63
E	0.171	0.173	0.175	4.35	4.4	4.45
H_D	0.200	0.202	0.204	5.077	5.127	5.177
H_E	0.248	0.252	0.248	6.30	6.40	6.30
L	0.033	0.037	0.041	0.85	0.95	1.05
L_1	0.020	0.024	0.028	0.50	0.60	0.70
a	1°	3°	5°	1°	3°	5°
e	0.026 BSC			0.65 BSC		
m	10°	12°	14°	10°	12°	14°

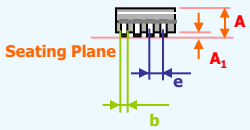
Notes:

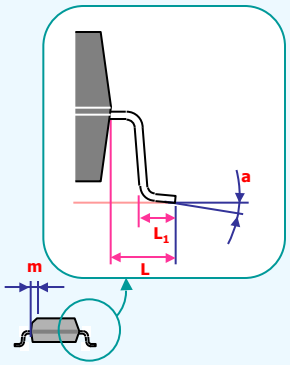
1. Dimension D & E include mold mismatch and are determined at the mold parting line.
2. General appearance spec. should be based on final visual inspection spec.

11. Package Dimensions : 8-SOP

Preliminary







[8-SOP]

Symbol	Dimension in Inches			Dimension in mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
A	0.093	0.099	0.104	2.35	2.45	2.65
A_1	0.004	0.008	0.012	0.10	0.20	0.30
b	0.014	0.016	0.019	0.35	0.42	0.49
D	-	0.150	-	-	3.81	-
E	0.150	0.153	0.157	3.80	3.90	4.00
H_D	0.189	0.193	0.197	4.80	4.90	5.00
H_E	0.234	0.239	0.244	5.95	6.07	6.20
L	0.038	0.043	0.048	0.97	1.08	1.2
L_1	0.022	0.027	0.032	0.58	0.70	0.82
a	0°	-	8°	0°	-	8°
e	0.050 BSC			1.27 BSC		
m	0.010	0.015	0.020	0.25	0.37	0.50

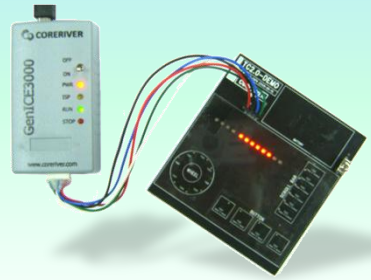
Notes:

1. Dimension D & E include mold mismatch and are determined at the mold parting line.
2. General appearance spec. should be based on final visual inspection spec.

12. Supporting tools

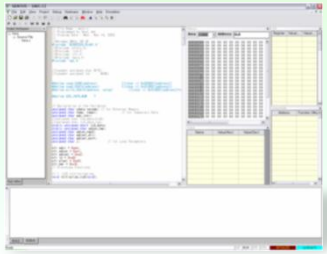
Preliminary

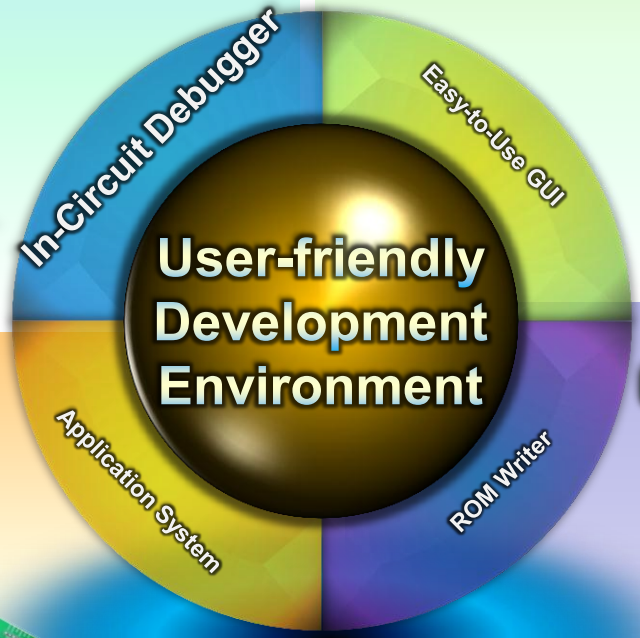
In-Circuit Debugger (GENSYS & GenICE)



Easy-to-Use GUI (GENTOS)

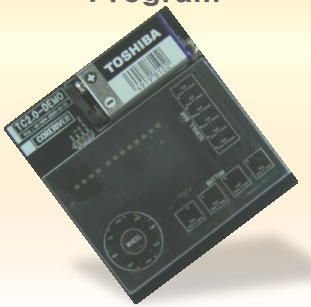
- Assembler & Linker for Windows
- Optimized Cross-C Compiler

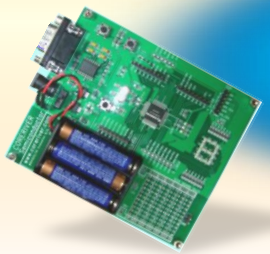




Application System

- On-board Application (with MCU Demo)
- Various Sample Test Program





ROM Writer

- World Wide Programmable in Anywhere (Hi-Lo Systems, ADVANTECH, TOPMAX, CORERIVER)
- Support Parallel / Serial Programming

Appendix A : Instruction Set (1/19)

- Note on Instruction Set and Addressing Modes

Notation	Descriptions
Rn	Register R0 ~ R7 of the currently selected Register Bank (RB0 ~ RB3).
direct	The address of 8-bit internal data location. This could be an IRAM location (0x00 ~ 0x7F; 128 bytes) or a SFR (0x80 ~ 0xFF).
@Ri	8-bit IRAM location (0x00 ~ 0xFF; 256 bytes) addressed indirectly through register **R0** or **R1**.
#data	8-bit constant included in instruction.
#data16	16-bit constant included in instruction.
addr16	16-bit destination address. Used by **LCALL** & **LJMP**. The branch can be anywhere within the 64kbytes program memory address space. (MiDAS230 Family : 2kbytes program memory)
addr11	11-bit destination address. Used by **ACALL** & **AJMP**. The branch will be within the same 2kbytes page of program memory as the first byte of the following instruction.
rel	Signed (2's complement number) 8-bit offset byte. Used by **SJMP** and all conditional jumps. Range is -128 to +127 byte relative to first byte of the following instruction.
Bit	Direct addressed bit n IRAM of SFR.

Appendix A : Instruction Set (2/19)

ADD A, <src-byte>

Add

1 cycle = 4 clocks

ADD A, Rn

Operation : (A) ← (A) + (Rn)

Encoding : HEX: 28h, #bytes: 1, Cycles: 1

0	0	1	0	1	r	r	r

ADD A, @Ri

Operation : (A) ← (A) + ((Ri))

Encoding : HEX: 26h, #bytes: 1, Cycles: 1

0	0	1	0	0	1	1	i

ADD A, direct

Operation : (A) ← (A) + (direct)

Encoding : HEX: 25h, #bytes: 2, Cycles: 2

0	0	1	0	0	1	0	1	direct addr

ADD A, #date

Operation : (A) ← (A) + data

Encoding : HEX: 24h, #bytes: 2, Cycles: 2

0	0	1	0	0	1	0	0	immediate data

ADDC A, <src-byte>

Add with Carry

ADDC A, Rn

Operation : (A) ← (A) + (C) + (Rn)

Encoding : HEX: 38h, #bytes: 1, Cycles: 1

0	0	1	1	1	r	r	r

ADDC A, @Ri

Operation : (A) ← (A) + (C) + ((Ri))

Encoding : HEX: 36h, #bytes: 1, Cycles: 1

0	0	1	1	0	1	1	i

ADDC A, direct

Operation : (A) ← (A) + (C) + (direct)

Encoding : HEX: 35h, #bytes: 2, Cycles: 2

0	0	1	1	0	1	0	1	direct addr

ADDC A, #date

Operation : (A) ← (A) + (C) + data

Encoding : HEX: 34h, #bytes: 2, Cycles: 2

0	0	1	1	0	1	0	0	immediate data

SUBB A, <src-byte>

Subtract with Borrow

SUBB A, Rn

Operation : (A) ← (A) - (C) - (Rn)

Encoding : HEX: 98h, #bytes: 1, Cycles: 1

1	0	0	1	1	r	r	r

SUBB A, @Ri

Operation : (A) ← (A) - (C) - ((Ri))

Encoding : HEX: 96h, #bytes: 1, Cycles: 1

1	0	0	1	0	1	1	i

SUBB A, direct

Operation : (A) ← (A) - (C) - (direct)

Encoding : HEX: 95h, #bytes: 2, Cycles: 2

1	0	0	1	0	1	0	1	direct addr

SUBB A, #date

Operation : (A) ← (A) - (C) - data

Encoding : HEX: 94h, #bytes: 2, Cycles: 2

1	0	0	1	0	1	0	0	immediate data

INC <byte>

Increment

INC A

Operation : (A) ← (A) + 1

Encoding : HEX: 04h, #bytes: 1, Cycles: 1

0	0	0	0	0	1	0	0

INC Rn

Operation : (Rn) ← (Rn) + 1

Encoding : HEX: 08h, #bytes: 1, Cycles: 1

0	0	0	0	1	r	r	r

INC @Ri

Operation : ((Ri)) ← ((Ri)) + 1

Encoding : HEX: 06h, #bytes: 1, Cycles: 1

0	0	0	0	0	1	1	i

INC direct

Operation : (direct) ← (direct) + 1

Encoding : HEX: 05h, #bytes: 2, Cycles: 2

0	0	0	0	0	1	0	1	direct addr

INC DPTR

Operation : (DPTR) ← (DPTR) + 1

Encoding : HEX: A3h, #bytes: 1, Cycles: 1

1	0	1	0	0	0	1	1

DEC <byte>

Decrement

DEC A

Operation : (A) ← (A) - 1

Encoding : HEX: 14h, #bytes: 1, Cycles: 1

0	0	0	1	0	1	0	0

DEC Rn

Operation : (Rn) ← (Rn) - 1

Encoding : HEX: 18h, #bytes: 1, Cycles: 1

0	0	0	1	1	r	r	r

DEC @Ri

Operation : ((Ri)) ← ((Ri)) - 1

Encoding : HEX: 16h, #bytes: 1, Cycles: 1

0	0	0	1	0	1	1	i

DEC direct

Operation : (direct) ← (direct) - 1

Encoding : HEX: 15h, #bytes: 1, Cycles: 1

0	0	0	1	0	1	0	1	direct addr

MUL AB

Multiply

Operation : $(A)_{7-0}$, $(B)_{15-8}$ ← (A) x (B)

Encoding : HEX: A4h, #bytes: 1, Cycles: 3

1	0	1	0	0	1	0	0

DIV AB

Divide

Operation : $(A)_{15-8}$, $(B)_{7-0}$ ← (A) / (B)

Encoding : HEX: 84h, #bytes: 1, Cycles: 3

1	0	0	0	0	1	0	0

Appendix A : Instruction Set (5/19)

DA A

Decimal-adjust Accumulator for Addition

Operation :

```
IF [[(A3-0)>9] ∨ [(AC)=1]]
        THEN (A3-0) ← (A3-0)+6
IF [[(A7-4)>9] ∨ [(C)=1]]
        THEN (A7-4) ← (A7-4)+6
```

Encoding : HEX: D4h, #bytes: 1, Cycles: 1

1	1	0	1	0	1	0	0

ANL <dest-byte>, <src-byte>

Logical AND for byte variables

ANL A, Rn

Operation : (A) ← (A) ^ (Rn)

Encoding : HEX: 58h, #bytes: 1, Cycles: 1

0	1	0	1	1	r	r	r

ANL A, @Ri

Operation : (A) ← (A) ^ ((Ri))

Encoding : HEX: 56h, #bytes: 1, Cycles: 1

0	1	0	1	0	1	1	i

ANL A, direct

Operation : (A) ← (A) ^ (direct)

Encoding : HEX: 55h, #bytes: 2, Cycles: 2

0	1	0	1	0	1	0	1	direct addr

ANL A, #date

Operation : (A) ← (A) ^ data

Encoding : HEX: 54h, #bytes: 2, Cycles: 2

0	1	0	1	0	1	0	0	immediate data

ANL direct, A

Operation : (direct) ← (direct) ^ (A)

Encoding : HEX: 52h, #bytes: 2, Cycles: 2

0	1	0	1	0	0	1	0	direct addr

ANL direct, #data

Operation : (direct) ← (direct) ^ data

Encoding : HEX: 53h, #bytes: 3, Cycles: 3

0	1	0	1	0	0	1	1	direct addr	immediate data

Appendix A : Instruction Set (6/19)

ANL C, <src-bit>

Logical AND for bit variables

```
ANL     C, bit
Operation :   (C)  ← (C) ^ (bit)
```

Encoding : HEX: 82h, #bytes: 2, Cycles: 2

1	0	0	0	0	0	1	0	bit addr

```
ANL     C, /bit
Operation :   (C)  ← (C) ^ ~(bit)
```

Encoding : HEX: B0h, #bytes: 2, Cycles: 2

1	0	1	1	0	0	0	0	bit addr

ORL <dest-byte>, <src-byte>

Logical OR for byte variables

```
ORL     A, Rn
Operation :   (A)  ← (A) ∨ (Rn)
```

Encoding : HEX: 48h, #bytes: 1, Cycles: 1

0	1	0	0	1	r	r	r

```
ORL     A, @Ri
Operation :   (A)  ← (A) ∨ ((Ri))
```

Encoding : HEX: 46h, #bytes: 1, Cycles: 1

0	1	0	0	0	1	1	i

```
ORL     A, direct
Operation :   (A)  ← (A) ∨ (direct)
```

Encoding : HEX: 45h, #bytes: 2, Cycles: 2

0	1	0	0	0	1	0	1	direct addr

```
ORL     A, #date
Operation :   (A)  ← (A) ∨ data
```

Encoding : HEX: 44h, #bytes: 2, Cycles: 2

0	1	0	0	0	1	0	0	immediate data

```
ORL     direct, A
Operation :   (direct)  ← (direct) ∨ (A)
```

Encoding : HEX: 42h, #bytes: 2, Cycles: 2

0	1	0	0	0	0	1	0	direct addr

```
ORL     direct, #data
Operation :   (direct)  ← (direct) ∨ data
```

Encoding : HEX: 43h, #bytes: 3, Cycles: 3

0	1	0	0	0	0	1	1	direct addr	immediate data

ORL C, <src-byte>

Logical OR for byte variables

```
ORL     C, bit
Operation :   (C)  ← (C) ∨ (bit)
```

Encoding : HEX: 72h, #bytes: 2, Cycles: 2

0	1	1	1	0	0	1	0	bit addr

```
ORL     C, /bit
Operation :   (C)  ← (C) ∨ ~(bit)
```

Encoding : HEX: A0h, #bytes: 2, Cycles: 2

1	0	1	0	0	0	0	0	bit addr

XRL <dest-byte>, <src-byte>

Logical Exclusive-OR for byte variables

```
XRL     A, Rn
Operation :   (A)  ← (A) ⊕ (Rn)
```

Encoding : HEX: 68h, #bytes: 1, Cycles: 1

0	1	1	0	1	r	r	r

```
XRL     A, @Ri
Operation :   (A)  ← (A) ⊕ ((Ri))
```

Encoding : HEX: 66h, #bytes: 1, Cycles: 1

0	1	1	0	0	1	1	i

```
XRL     A, direct
Operation :   (A)  ← (A) ⊕ (direct)
```

Encoding : HEX: 65h, #bytes: 2, Cycles: 2

0	1	1	0	0	1	0	1	direct addr

```
XRL     A, #date
Operation :   (A)  ← (A) ⊕ data
```

Encoding : HEX: 64h, #bytes: 2, Cycles: 2

0	1	1	0	0	1	0	0	immediate data

```
XRL     direct, A
Operation :   (direct) ← (direct) ⊕ (A)
```

Encoding : HEX: 62h, #bytes: 2, Cycles: 2

0	1	1	0	0	0	1	0	direct addr

```
XRL     direct, #data
Operation :   (direct) ← (direct) ⊕ data
```

Encoding : HEX: 63h, #bytes: 3, Cycles: 3

0	1	1	0	0	0	1	1	direct addr	immediate Data

Appendix A : Instruction Set (8/19)

CLR A

Clear Accumulator

Operation : (A) ← 0

Encoding : HEX: E4h, #bytes: 1, Cycles: 1

1	1	1	0	0	1	0	0

CLR <bit>

Clear bit

CLR C

Operation : (C) ← 0

Encoding : HEX: C3h, #bytes: 1, Cycles: 1

1	1	0	0	0	0	1	1

CLR bit

Operation : (bit) ← 0

Encoding : HEX: C2h, #bytes: 2, Cycles: 2

1	1	0	0	0	0	1	0	bit addr

CPL A

Complement Accumulator

Operation : (A) ← ~(A)

Encoding : HEX: F4h, #bytes: 1, Cycles: 1

1	1	1	1	0	1	0	0

CPL <bit>

Complement bit

CPL C

Operation : (C) ← ~(C)

Encoding : HEX: B3h, #bytes: 1, Cycles: 1

1	0	1	1	0	0	1	1

CPL bit

Operation : (bit) ← ~(bit)

Encoding : HEX: B2h, #bytes: 2, Cycles: 2

1	0	1	1	0	0	1	0	bit addr

RL A

Rotate Accumulator Left

Operation : $(A_{n+1}) \leftarrow (A_n)$ n=0~6
$(A_0) \leftarrow (A_7)$

Encoding : HEX: 23h, #bytes: 1, Cycles: 1

0	0	1	0	0	0	1	1

RLC A

Rotate Accumulator Left through the Carry flag

Operation : $(A_{n+1}) \leftarrow (A_n)$ n=0~6
$(A_0) \leftarrow (C)$
$(C) \leftarrow (A_7)$

Encoding : HEX: 33h, #bytes: 1, Cycles: 1

0	0	1	1	0	0	1	1

RR A

Rotate Accumulator Right

Operation : $(A_n) \leftarrow (A_{n+1})$ n=0~6
$(A_7) \leftarrow (A_0)$

Encoding : HEX: 03h, #bytes: 1, Cycles: 1

0	0	0	0	0	0	1	1

RRC A

Rotate Accumulator Right through the Carry flag

Operation : $(A_n) \leftarrow (A_{n+1})$ n=0~6
$(A_7) \leftarrow (C)$
$(C) \leftarrow (A_0)$

Encoding : HEX: 13h, #bytes: 1, Cycles: 1

0	0	0	1	0	0	1	1

SWAP A

Swap nibbles within the Accumulator

Operation : $(A_{3-0}) \leftrightarrow (A_{7-4})$

Encoding : HEX: C4h, #bytes: 1, Cycles: 1

1	1	0	0	0	1	0	0

MOV <dest-byte>, <src-byte>

Move byte variable

MOV A, Rn

Operation : (A) ← (Rn)

Encoding : HEX: E8h, #bytes: 1, Cycles: 1

1	1	1	0	1	r	r	r

MOV A, @Ri

Operation : (A) ← ((Ri))

Encoding : HEX: E6h, #bytes: 1, Cycles: 1

1	1	1	0	0	1	1	i

MOV A, direct

Operation : (A) ← (direct)

Encoding : HEX: E5h, #bytes: 2, Cycles: 2

1	1	1	0	0	1	0	1	direct addr

MOV A, #date

Operation : (A) ← data

Encoding : HEX: 74h, #bytes: 2, Cycles: 2

0	1	1	1	0	1	0	0	immediate data

MOV Rn, A

Operation : (Rn) ← (A)

Encoding : HEX: F8h, #bytes: 1, Cycles: 1

1	1	1	1	1	r	r	r

MOV Rn, direct

Operation : (Rn) ← (direct)

Encoding : HEX: A8h, #bytes: 2, Cycles: 2

1	0	1	0	1	r	r	r	direct addr

MOV Rn, #date

Operation : (Rn) ← data

Encoding : HEX: 78h, #bytes: 2, Cycles: 2

0	1	1	1	1	r	r	r	immediate data

MOV direct, A

Operation : (direct) ← (A)

Encoding : HEX: F5h, #bytes: 2, Cycles: 2

1	1	1	1	0	1	0	1	direct addr

MOV direct, Rn

Operation : (direct) ← (Rn)

Encoding : HEX: 88h, #bytes: 2, Cycles: 2

1	0	0	0	1	r	r	r	direct addr

Appendix A : Instruction Set (11/19)

MOV direct, @Ri

Operation : (direct) ← ((Ri))

Encoding : HEX: 86h, #bytes: 2, Cycles: 2

1	0	0	0	0	1	1	i	direct addr

MOV direct, direct

Operation : (direct) ← (direct)

Encoding : HEX: 85h, #bytes: 3, Cycles: 3

1	0	0	0	0	1	0	1	direct addr(src)	direct addr(dest)

MOV direct, #data

Operation : (direct) ← data

Encoding : HEX: 75h, #bytes: 3, Cycles: 3

0	1	1	1	0	1	0	1	direct addr	immediate data

MOV @Ri, A

Operation : ((Ri)) ← (A)

Encoding : HEX: F6h, #bytes: 1, Cycles: 1

1	1	1	1	0	1	1	i

MOV @Ri, direct

Operation : ((Ri)) ← (direct)

Encoding : HEX: A6h, #bytes: 2, Cycles: 2

1	0	1	0	0	1	1	i	direct addr

MOV @Ri, #data

Operation : ((Ri)) ← data

Encoding : HEX: 76h, #bytes: 2, Cycles: 2

0	1	1	1	0	1	1	i	immediate Data

MOV <dest-bit>, <src-bit>

Move bit data

MOV C, bit

Operation : (C) ← (bit)

Encoding : HEX: A2h, #bytes: 2, Cycles: 2

1	0	1	0	0	0	1	0	bit addr

MOV bit, C

Operation : (bit) ← (C)

Encoding : HEX: 92h, #bytes: 2, Cycles: 2

1	0	0	1	0	0	1	0	bit addr

MOV DPTR, #data16

Load Data Pointer with a 16-bit constant

Operation :
(DPTR) ← $data_{15-0}$
(DPH,DPL) ← ($data_{15-8}$,$data_{7-0}$)

Encoding : HEX: 90h, #bytes: 3, Cycles: 3

1	0	0	1	0	0	0	0	immed. data 15-8	immed. data 7-0

MOVC A, @A + <base-reg>

Move Code byte

MOVC A, @A + DPTR

Operation : (A) ← ((A) + (DPTR))

Encoding : HEX: 93h, #bytes: 1, Cycles: 2

1	0	0	1	0	0	1	1

MOVC A, @A + PC

Operation :
(PC) ← (PC) + 1
(A) ← ((A) + (PC))

Encoding : HEX: 83h, #bytes: 1, Cycles: 2

1	0	0	0	0	0	1	1

MOVX <dest-byte>, <src-byte>

Move External

MOVX A, @Ri

Operation : (A) ← ((Ri))

Encoding : HEX: E2h, #bytes: 1, Cycles: 3

1	1	1	0	0	0	1	i

MOVX A, @DPTR

Operation : (A) ← ((DPTR))

Encoding : HEX: E0h, #bytes: 1, Cycles: 3

1	1	1	0	0	0	0	0

MOVX @Ri, A

Operation : ((Ri)) ← (A)

Encoding : HEX: F2h, #bytes: 1, Cycles: 3

1	1	1	1	0	0	1	i

MOVX @DPTR, A

Operation : ((DPTR)) ← (A)

Encoding : HEX: F0h, #bytes: 1, Cycles: 3

1	1	1	1	0	0	0	0

Appendix A : Instruction Set (13/19)

XCH A, <src-byte>

Exchange Accumulator with byte variable

XCH A, Rn

Operation : (A) ↔ (Rn)

Encoding : HEX: C8h, #bytes: 1, Cycles: 1

1	1	0	0	1	r	r	r

XCH A, @Ri

Operation : (A) ↔ ((Ri))

Encoding : HEX: C6h, #bytes: 1, Cycles: 1

1	1	0	0	0	1	1	i

XCH A, direct

Operation : (A) ↔ (direct)

Encoding : HEX: C5h, #bytes: 2, Cycles: 2

1	1	0	0	0	1	0	1	direct addr

XCHD A, @Ri

Exchange Digit

Operation : $(A_{3-0}) \leftrightarrow ((Ri))_{3-0}$

Encoding : HEX: D6h, #bytes: 1, Cycles: 1

1	1	0	1	0	1	1	i

PUSH direct

Push onto stack

Operation : (SP) ← (SP) + 1
((SP)) ← (direct)

Encoding : HEX: C0h, #bytes: 2, Cycles: 2

1	1	0	0	0	0	0	0	direct addr

POP direct

Pop onto stack

Operation : (direct) ← ((SP))
(SP) ← (SP) - 1

Encoding : HEX: D0h, #bytes: 2, Cycles: 2

1	1	0	1	0	0	0	0	direct addr

Appendix A : Instruction Set (14/19)

SETB <bit>

Set bit

SETB C

Operation : (C) ← 1

Encoding : HEX: D3h, #bytes: 1, Cycles: 1

1	1	0	1	0	0	1	1

SETB bit

Operation : (bit) ← 1

Encoding : HEX: D2h, #bytes: 2, Cycles: 2

1	1	0	1	0	0	1	0	bit addr

JC rel

Jump if Carry is set

Operation :
(PC) ← (PC) + 2
If (C) = 1, then (PC) ← (PC) + rel

Encoding : HEX: 40h, #bytes: 2, Cycles: 3

0	1	0	0	0	0	0	0	relative addr

JNC rel

Jump if Carry is not set

Operation :
(PC) ← (PC) + 2
If (C) = 0, then (PC) ← (PC) + rel

Encoding : HEX: 50h, #bytes: 2, Cycles: 3

0	1	0	1	0	0	0	0	relative addr

JB bit, rel

Jump if Bit is set

Operation :
(PC) ← (PC) + 3
If (bit) = 1, then (PC) ← (PC)+rel

Encoding : HEX: 20h, #bytes: 3, Cycles: 4

0	0	1	0	0	0	0	0	bit addr	relative addr

JNB bit, rel

Jump if Bit is not set

Operation :
(PC) ← (PC) + 3
If (bit) = 0, then (PC) ← (PC)+rel

Encoding : HEX: 30h, #bytes: 3, Cycles: 4

0	0	1	1	0	0	0	0	bit addr	relative addr

Appendix A : Instruction Set (15/19)

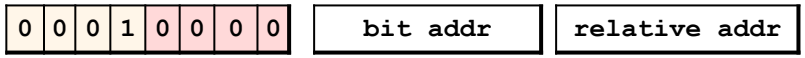
JBC bit, rel

Jump if Bit is set and Clear bit

Operation :

```
(PC)  ← (PC) + 3
If (bit) = 1,
then (bit) ← 0, (PC) ← (PC) + rel
```

Encoding : HEX: 10h, #bytes: 3, Cycles: 4



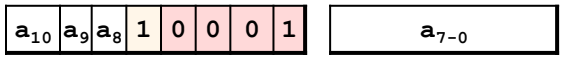
ACALL addr11

Absolute Subroutine Call

Operation :

```
(PC)    ← (PC) + 2
(SP)    ← (SP) + 1
((SP))  ← (PC7-0)
(SP)    ← (SP) + 1
((SP))  ← (PC15-8)
(PC10-0) ← page address
```

Encoding : HEX: 11h, #bytes: 2, Cycles: 3



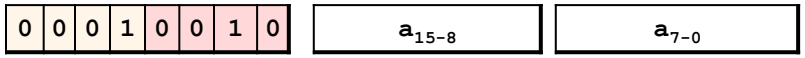
LCALL addr16

Long Subroutine Call

Operation :

```
(PC)    ← (PC) + 3
(SP)    ← (SP) + 1
((SP))  ← (PC7-0)
(SP)    ← (SP) + 1
((SP))  ← (PC15-8)
(PC)    ← addr15-0
```

Encoding : HEX: 12h, #bytes: 3, Cycles: 4



Appendix A : Instruction Set (16/19)

RET

Return from Subroutine

Operation :

$(PC_{15-8}) \leftarrow ((SP))$
$(SP) \leftarrow (SP) - 1$
$(PC_{7-0}) \leftarrow ((SP))$
$(SP) \leftarrow (SP) - 1$

Encoding : HEX: 22h, #bytes: 1, Cycles: 2

0	0	1	0	0	0	1	0

RETI

Return from Interrupt

Operation :

$(PC_{15-8}) \leftarrow ((SP))$
$(SP) \leftarrow (SP) - 1$
$(PC_{7-0}) \leftarrow ((SP))$
$(SP) \leftarrow (SP) - 1$

Encoding : HEX: 32h, #bytes: 1, Cycles: 2

0	0	1	1	0	0	1	0

AJMP addr11

Absolute Jump

Operation :

$(PC) \leftarrow (PC) + 2$
$(PC_{10-0}) \leftarrow$ page address

Encoding : HEX: 01h, #bytes: 2, Cycles: 3

a_{10}	a_9	a_8	0	0	0	0	1	a_{7-0}

SJMP rel

Short Jump (Relative address)

Operation :

$(PC) \leftarrow (PC) + 2$
$(PC_{10-0}) \leftarrow (PC) + rel$

Encoding : HEX: 80h, #bytes: 2, Cycles: 3

1	0	0	0	0	0	0	0	relative addr

LJMP addr16

Long Jump

Operation : $(PC) \leftarrow addr_{15-0}$

Encoding : HEX: 02h, #bytes: 3, Cycles: 4

0	0	0	0	0	0	1	0	a_{15-8}	a_{7-0}

JMP @A + DPTR

Jump Indirect Relative to the DPTR

Operation : (PC) ← (A) + (DPTR)

Encoding : HEX: 73h, #bytes: 1, Cycles: 2

0	1	1	1	0	0	1	1

JZ rel

Jump if Accumulator is Zero

Operation :
(PC) ← (PC) + 2
If (A)=0, then (PC) ← (PC) + rel

Encoding : HEX: 60h, #bytes: 2, Cycles: 3

0	1	1	0	0	0	0	0	relative addr

JNZ rel

Jump if Accumulator is Not Zero

Operation :
(PC) ← (PC) + 2
If (A)≠0, then (PC) ← (PC) + rel

Encoding : HEX: 70h, #bytes: 2, Cycles: 3

0	1	1	1	0	0	0	0	relative addr

Appendix A : Instruction Set (18/19)

CJNE <dest-byte>, <src-byte>, rel

Compare and Jump if Not Equal

CJNE A, direct, rel

Operation :

```
(PC)  ← (PC) + 3
If (A) ≠ (direct),
      then (PC) ← (PC) + rel
If (A) < (direct), then (C) ← 1
Else                (C) ← 0
```

Encoding : HEX: B5h, #bytes: 3, Cycles: 4

1	0	1	1	0	1	0	1	direct addr	relative addr

CJNE A, #data, rel

Operation :

```
(PC)  ← (PC) + 3
If (A) ≠ data,
      then (PC) ← (PC) + rel
If (A) < data, then (C) ← 1
Else               (C) ← 0
```

Encoding : HEX: B4h, #bytes: 3, Cycles: 4

1	0	1	1	0	1	0	0	immediate data	relative addr

CJNE Rn, #data, rel

Operation :

```
(PC)  ← (PC) + 3
If (Rn) ≠ data,
      then (PC) ← (PC) + rel
If (Rn) < data, then (C) ← 1
Else               (C) ← 0
```

Encoding : HEX: B8h, #bytes: 3, Cycles: 4

1	0	1	1	1	r	r	r	immediate data	relative addr

CJNE @Ri, #data, rel

Operation :

```
(PC)  ← (PC) + 3
If ((Ri)) ≠ data,
      then (PC) ← (PC) + rel
If ((Ri)) < data, then (C) ← 1
Else               (C) ← 0
```

Encoding : HEX: B6h, #bytes: 3, Cycles: 4

1	0	1	1	0	1	1	i	immediate data	relative addr

DJNZ <byte>, rel

Decrement and Jump if Not Zero

DJNZ Rn, rel

Operation :

```
(PC)  ← (PC) + 2
(Rn)  ← (Rn) - 1
If (Rn)≠0, then (PC) ← (PC) + rel
```

Encoding : HEX: D8h, #bytes: 2, Cycles: 3

1	1	0	1	1	r	r	r	relative addr

DJNZ direct, rel

Operation :

```
(PC)     ← (PC) + 3
(direct) ← (direct) - 1
If (direct)≠0,
        then (PC) ← (PC) + rel
```

Encoding : HEX: D5h, #bytes: 3, Cycles: 4

1	1	0	1	0	1	0	1	direct addr	relative addr

NOP

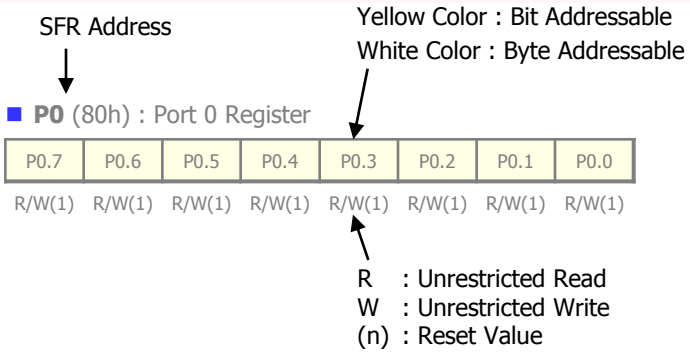
No Operation

Operation : (PC) ← (PC) + 1

Encoding : HEX: 00h, #bytes: 1, Cycles: 1

0	0	0	0	0	0	0	0

[How to Read a SFR Descriptions]



■ **P0** (80h) : Port 0 Register

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

◆ Port 0 Register

■ **SP** (81h) : Stack Pointer Register

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

◆ Indicate where stack will start.
◆ Increment by PUSH and decrement by POP.

■ **DPL** (82h) : Data Pointer Low Register

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

■ **DPH** (83h) : Data Pointer High Register

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

■ **PCON** (87h) : Power Control Register

-	-	-	POF	GF1	GF0	PD	IDL
			R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

◆ POF : Power off flag.
When power-on, this bit will be set by H/W.
◆ GF1, GF0 : General purpose flag bit. Initialized only by POR.
◆ PD : Power-down (Stop) mode enable.
◆ IDL : IDLE mode enable.

■ **TCON** (88h) : Timer/Counter 0/1 Control Register

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- TF1 : Timer 1 overflow flag.
- TR1 : Timer 1 run enable.
- TF0 : Timer 0 overflow flag.
- TR0 : Timer 0 run enable.
- IE1 : External interrupt 1 flag.
  If IT1 = 0, cleared by S/W (software).
  If IT1 = 1, cleared automatically when go to routine.
- IT1 : External interrupt 1 TYP select flag.
  Edge detect (IT1=1) / Level detect (IT1=0; Default)
- IE0 : External interrupt 0 flag.
  If IT0 = 0, cleared by S/W (software).
  If IT0 = 1, cleared automatically when go to routine.
- IT0 : External interrupt 0 TYP select flag.
  Edge detect (IT0=1) / Level detect (IT0=0; Default)

■ **TMOD** (89h) : Timer/Counter 0 Mode Control Register

-	-	-	-	GATE	C/T	M1	M0
				R/W(0)	R/W(0)	R/W(0)	R/W(0)

- GATE[3] : Timer 0 gate control.
- C/T[2] : Timer 0 Counter/Timer select.
  0 = Timer by $F_{OSC}/12$. (Default)
  1 = Counter by T0 pin.
- M1, M0 : Timer 0 mode selection.
  [0,0] : Mode0, 13-bit T/C
  [0,1] : Mode1, 16-bit T/C
  [1,0] : Mode2, 8-bit T/C with auto-reload
  [1,1] : Mode3, Two 8-bit T/C

■ **TL0** (8Ah) : Timer/Counter 0 Low Byte Register

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

■ **TL1** (8Bh) : Timer/Counter 1 Low Byte Register

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

■ **TH0** (8Ch) : Timer/Counter 0 High Byte Register

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

■ **TH1** (8Dh) : Timer/Counter 1 High Byte Register

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

■ **P1** (90h) : Port 1 Register

-	-	-	-	-	P1.2	P1.1	P1.0
					R/W(1)	R/W(1)	R/W(1)

- P1.0 : XTAL 1 alternative. (Default = XTAL1).
- P1.1 : XTAL 2 alternative. (Default = XTAL2).
  Refer to ALTSEL & PMR SFR for I/O pins.
- P1.2 : RESETB alternative. (Default = RESETB)
  Refer to ALTSEL for I/O Pins.

■ **EXIF** (91h) : External Interrupt Flag Register

-				XT/RG	RGMD	RGSL	BGS
-				R/W(0)	R(1)	R(1)	R/W(1)

- XT/RG : System clock selection.
  0 = Internal Precision oscillator is selected as system clock.
  1 = External clock is selected as system clock.
- RGMD : RCLK mode. Now system clock is POSC or XTAL.
  Generally RGMD is the invert of XT/RG.
- RGSL : RCLK select bit when power-down wake-up.
  If set, when wake-up from power-down in XTAL clock,
  use RCLK as system clock during the first 65,536 cycles.
  This bit is always 1.
- BGS : Band-gap select. (Default = 1)
  If 0, Band-gap block (POR) will be off in power-down mode.
  If 1, Band-gap block (POR) will run in power-down mode.

■ **RINGCON** (95h: Internal Oscillator Frequency Tuning

S7	S6	S5	S4	S3	S2	S1	S0
R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

■ **OSCBIAS** (96h) : Internal Oscillator Bias Current Tuning

S7	S6	S5	S4	S3	S2	S1	S0
R/W(0)	R/W(1)	R/W(0)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

■ **OSCREF**(97h) : Internal Oscillator Reference Tuning

-	-	-	S4	S3	S2	S1	S0
			R/W(1)	R/W(0)	R/W(1)	R/W(1)	R/W(1)

■ **P2** (A0h) : Port 2 Register

-	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

◆ Port 2 Register

■ **IE** (A8h) : Interrupt Enable Register

EA	EADC	EPWM	EWDT	ET1	EX1	ET0	EX0
R/W(0)	R/W(0)		R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

◆ EA : Global interrupt enable.
◆ EADC : ADC interrupt enable.
◆ EPWM : PWM interrupt enable.
◆ EWDT : WDT interrupt enable.
◆ ET1 : Timer 1 interrupt enable.
◆ EX1 : External interrupt 1 enable.
◆ ET0 : Timer0 interrupt enable.
◆ EX0 : External interrupt 0 enable.

■ **IP** (B8h) : Interrupt Priority Register

-	PADC	PPWM	PWDT	PT1	PX1	PT0	PX0
	R/W(0)		R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

◆ PADC : ADC interrupt priority.
◆ PPWM : PWM interrupt priority.
◆ PWDT : WDT interrupt priority.
◆ PT1 : Timer 1 interrupt priority.
◆ PX1 : External interrupt 1 priority.
◆ PT0 : Timer 0 interrupt priority.
◆ PX0 : External interrupt 0 priority.

■ **OSCICN** (BEh) : Internal Oscillator Control Register

-	-	-	-	DIV2	RINGON	DIV1	DIV0
				R/W(0)	R/W(1)	R/W(0)	R/W(0)

◆ RINGON : 1 = Internal oscillator(12MHz) is running.
0 = Internal oscillator is off.
Don't clear RINGON bit when XTRG = 0.

◆ DIV2, DIV1, DIV0 : Internal oscillator divider.
[0,0,0] = 4MHz ; Default
[0,0,1] = 2MHz
[0,1,0] = 1MHz
[0,1,1] = 0.5MHz
[1,0,0] = reserved
[1,0,1] = 12MHz
[1,1,0] = 6MHz
[1,1,1] = 3MHz

■ **PMR** (C4h) : Power Management Control Register

-	-	-	-	XTOFF	-	-	-
R/W(0)							

◆ XTOFF : Internal amplifier disable for external crystal oscillator.
1 = External crystal will be killed.
0 = External crystal will run (Default).
Don't set XTOFF bit when XT/RG = 1.

■ **STATUS** (C5h) : Crystal Status Register

-	-	-	XTUP	-	-	-	-
R(0)							

◆ XTUP : Crystal oscillator warm-up status.
It represents the crystal clock is stable (1) or not (0).
Cleared by H/W when Power-on reset and all kinds of reset.
Cleared by H/W when XTOFF bit is set.
Cleared by during Power-down wake-up when XT/RG = 1.
Set by H/W after XTAL stabilization time.

■ **LVDCFG** (C6h) : LVD Configuration Register

EPFR	EPFI	PFI	CFG4	CFG3	CFG2	CFG1	CFG0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R(1)	R/W(0)	R/W(1)

◆ EPFR : Power-fail reset enable.
◆ EPFI : Power-fail interrupt enable.
◆ PFI : Power-fail interrupt flag.
◆ CFG[4:0] : LVD Voltage Selection

NOTE : This SFR is initialized only by power-on-reset.
That is, it holds user's setting for any other reset.

CFG[4:0]	V_{LVD} [V]	CFG[4:0]	V_{LVD} [V]
1 1111	5.0	0 1111	3.4
1 1110	4.9	0 1110	3.3
1 1101	4.8	0 1101	3.2
1 1100	4.7	0 1100	3.1
1 1011	4.6	0 1011	3.0
1 1010	4.5	0 1010	2.9
1 1001	4.4	0 1001	2.8
1 1000	4.3	0 1000	2.7
1 0111	4.2	0 0111	2.6
1 0110	4.1	0 0110	2.5
1 0101	4.0	**0 0101**	**2.4**
1 0100	3.9	0 0100	2.3
1 0011	3.8	0 0011	2.2
1 0010	3.7	0 0010	2.1
1 0001	3.6	0 0001	2.0
1 0000	3.5	0 0000	1.9

■ **PSW** (D0h) : Program Status Word Register

CY	AC	F0	RS1	RS0	OV	F1	P
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R(0)

- CY : Carry flag.
- AC : Auxiliary carry flag.
- F0 : User flag 0.
- RS1, RS0 : Register bank select.
  [0,0] : Bank 0 [1,0] : Bank 2
  [0,1] : Bank 1 [1,1] : Bank 3
- OV : Overflow flag.
- F1 : User flag 1.
- P : Parity bit. Set/clear by H/W according to ACC odd parity.

■ **P0TYPE** (D4h) : Port 0 TYPE Control Register

P0TYPE.7	P0TYPE.6	P0TYPE.5	P0TYPE.4	P0TYPE.3	P0TYPE.2	P0TYPE.1	P0TYPE.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- 0 = Push-pull Output (Default) / 1 = Open-drain Output

■ **P1TYPE** (D5h) : Port 1 TYPE Control Register

					P1TYPE.2	P1TYPE.1	P1TYPE.0
					R/W(0)	R/W(0)	R/W(0)

- 0 = Push-pull Output (Default) / 1 = Open-drain output

■ **P2TYPE** (D6h) : Port 2 TYPE Control Register

	P2TYPE.6	P2TYPE.5	P2TYPE.4	P2TYPE.3	P2TYPE.2	P2TYPE.1	P2TYPE.0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- 0 = Push-pull Output (Default) / 1 = Open-drain Output

■ WDCON (D8h) : Watchdog Timer Control Register

WD1	WD0	WDM	-	WDIF	WTRF	EWT	RWT
R/W(1)	R/W(1)	R/W(0)	R(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- {WDM,WD1,WD0} : Watchdog timer mode select.
  - [0,0,0] : 8 x 2^{16} clocks (interrupt)
  - [0,0,1] : 16 x 2^{16} clocks (interrupt)
  - [0,1,0] : 32 x 2^{16} clocks (interrupt)
  - [0,1,1] : 64 x 2^{16} clocks (interrupt)
  - [1,0,0] : 8 x 2^{6} clocks (interrupt)
  - [1,0,1] : 16 x 2^{6} clocks (interrupt)
  - [1,1,0] : 32 x 2^{6} clocks (interrupt)
  - [1,1,1] : 64 x 2^{6} clocks (interrupt)
- WDIF : Watchdog timer interrupt flag.
- WTRF : Watchdog timer Reset flag.
- EWT : Watchdog timer Reset Enable.
- RWT : Restart watchdog timer.

■ PWMCON (DCh) : PWM Control Register

PWM06	PS2_P0	PS1_P0	PS0_P0	-	PWMF	CLR_P0	RUN_P0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R(0)	R/W(0)	R/W(0)	R/W(0)

- PWM06 : PWM waveform output enable to P0.6.
- PS2_P0, PS1_P0, PS0_P0 : Pre-scaled Clock Selection.
  - [0,0,0] = $F_{OSC}/1$, [0,0,1] = $F_{OSC}/2$, [0,1,0] = $F_{OSC}/4$,
  - [0,1,1] = $F_{OSC}/8$, [1,0,0] = $F_{OSC}/16$, [1,0,1] = $F_{OSC}/32$,
  - [1,1,0] = $F_{OSC}/64$, [1,1,1] = $F_{OSC}/128$
- RL_P0 : PWM data update mode selector.
  - RL_P0=0, update at 6-bit Counter Overflow.
  - RL_P0=1, update at 8-bit Counter Overflow.
- PWMF : PWM Interrupt Flag. Cleared by S/W
- CLR_P0 : Counter reset enable. Cleared by H/W.
- RUN_P0 : Counter start enable.

■ PWMD (DEh) : PWM Duty Data Register

PWMD.7	PWMD.6	PWMD.5	PWMD.4	PWMD.3	PWMD.2	PWMD.1	PWMD.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

■ **ACC/A** (E0h) : Accumulator

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

■ **ADCSELH** (E1h) : ADC Channel Selection High Register

ADC11B	ADC10B	ADC9B	ADC8B	ADC7B	ADC6B	ADC5B	ADC4B
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

- ADC11B : 0 = ADC11 input enable & digital input disable at P2.2.
- ADC10B : 0 = ADC10 input enable & digital input disable at P2.3.
- ADC9B : 0 = ADC9 input enable & digital input disable at P2.4.
- ADC8B : 0 = ADC8 input enable & digital input disable at P2.5.
- ADC7B : 0 = ADC7 input enable & digital input disable at P2.6.
- ADC6B : 0 = ADC6 input enable & digital input disable at P0.7.
- ADC5B : 0 = ADC5 input enable & digital input disable at P0.6.
- ADC4B : 0 = ADC4 input enable & digital input disable at P0.5.

■ **ADCSEL** (E2h) : ADC Channel Selection Low & MUX Selection Register

ADC3B	ADC2B	ADC1B	ADC0B	CH3	Ch2	CH1	CH0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

- ADC3B : 0 = ADC3 input enable & digital input disable at P0.4.
- ADC2B : 0 = ADC2 input enable & digital input disable at P0.3.
- ADC1B : 0 = ADC1 input enable & digital input disable at P0.2.
- ADC0B : 0 = ADC0 input enable & digital input disable at P0.1.

- CH[3:0] : ADC MUX Selection.
  - [0,0,0,0] = ADC0 selection (0h)
  - [0,0,0,1] = ADC1 selection (1h)
  - [0,0,1,0] = ADC2 selection (2h)
  - [0,0,1,1] = ADC3 selection (3h)
  - [0,1,0,0] = ADC4 selection (4h)
  - [0,1,0,1] = ADC5 selection (5h)
  - [0,1,1,0] = ADC6 selection (6h)
  - [0,1,1,1] = ADC7 selection (7h)
  - [1,0,0,0] = ADC8 selection (8h)
  - [1,0,0,1] = ADC9 selection (9h)
  - [1,0,1,0] = ADC10 selection (Ah)
  - [1,0,1,1] = ADC11 selection (Bh)

  - [1,1,0,0] = No ADC input select (Ch)
  - [1,1,0,1] = No ADC input select (Dh)
  - [1,1,1,0] = No ADC input select (Eh)
  - [1,1,1,1] = No ADC input select (Fh, Default)

■ **ALTSEL** (E3h) : Alternative Function Control Register

IOXEN	IORSTEN	CLO	PWM00	TV0	-	-	-
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)			

- IOXEN : 1 = XTAL and XTAL2 is configured as I/O.
  Must be XTOFF (PMR.3) = 1 (Oscillator Amp. Off)
- IORSTEN : 1 = RESETB is configured as I/O.
- CLO : 1 = System clock output to P2.6.
- PWM00 : 1 = PWM waveform output enable to P0.0.
- TVO : 1 = Timer 0 overflow clock to P0.0.

■ **P0SEL** (E4h) : Port 0 Pull-up Control Register

P0SEL.7	P0SEL.6	P0SEL.5	P0SEL.4	P0SEL.3	P0SEL.2	P0SEL.1	P0SEL.0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(0)	R/W(0)

- 0 = Pull-up resistor is ON
- 1 = Pull-up resistor is OFF

■ **P1SEL** (E5h) : Port 1 Pull-up Control Register

-	-	-	-	-	P1SEL.2	P1SEL.1	P1SEL.0
					R/W(0)	R/W(1)	R/W(1)

- 0 = Pull-up resistor is ON
- 1 = Pull-up resistor is OFF

■ **P2SEL** (E6h) : Port 2 Pull-up Control Register

-	P2SEL.6	P2SEL.5	P2SEL.4	P2SEL.3	P2SEL.2	P2SEL.1	P2SEL.0
	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

- 0 = Pull-up resistor is ON
- 1 = Pull-up resistor is OFF (Default)

■ **ADCR** (EEh) : ADC Result High Register : Value[9:2]

SAR9	SAR8	SAR7	SAR6	SAR5	SAR4	SAR3	SAR2
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

■ **ADCON** (EFh) : ADC Control & ADC Result Low Register : Value[1:0]

AD_EN	AD_REQ	AD_END	ADCF	ADIV1	ADIV0	SAR1	SAR0
R/W(0)	R/W(0)	R(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- AD_EN : ADC ready enable.
- AD_REQ : ADC start.
  Cleared by H/W when AD_END goes to 1 from 0.
- AD_END : Current ADC status.
  0 = ADC is running now.
- ADCF : ADC interrupt flag. Must be cleared by S/W.
- ADIV1, ADIV0 : ADC input clock select.
  [0,0] = System clock (F_{SYS}) / 2 (Default)
  [0,1] = System clock (F_{SYS}) / 4
  [1,0] = System clock (F_{SYS}) / 8
  [1,1] = System clock (F_{SYS})
- SAR1, SAR0 : Low bits of ADC result value.

■ **B** (F0h) : B Register

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

■ **P0DIR** (F4h) : Port 0 Input/Output Control Register

P0DIR.7	P0DIR.6	P0DIR.5	P0DIR.4	P0DIR.3	P0DIR.2	P0DIR.1	P0DIR.0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

◆ 1 = Input (Default) / 0 = Output

■ **P1DIR** (F5h) : Port 1 Input/Output Control Register

-	-	-	-	-	P1DIR.2	P1DIR.1	P1DIR.0
					R/W(1)	R/W(1)	R/W(1)

◆ 1 = Input (Default) / 0 = Output

■ **P2DIR** (F6h) : Port 2 Input/Output Control Register

-	P2DIR.6	P2DIR.5	P2DIR.4	P2DIR.3	P2DIR.2	P2DIR.1	P2DIR.0
	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

◆ 1 = Input (Default) / 0 = Output

■ **IAPWEN** (FCh) : IAP Write/Erase Enable Register

WEN.7	WEN.6	WEN.5	WEN.4	WEN.3	WEN.2	WEN.1	WEN.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

■ **IAPCON** (FDh) : IAP Control SFR

-	-	-	-	RGS1	RGS0	OPS1	OPS0
				R/W(0)	R/W(0)	R/W(0)	R/W(0)

◆ RGS[1:0] : Select IAP region
- 00 : CODE (0x000 ~ 0x7FF)
- 01 : Reserved
- 10 : INFO (0x0 ~ 0xF)
- 11 : OTP (0x0 ~ 0xF)

◆ OPS[1:0] : Select IAP function
- 00 : No operation
- 01 : Byte Read (IAPDAT ← F[DPTR])
- 10 : Byte Erase (F[DPTR] ← 0)
- 11 : Byte Write (F[DPTR] ← IAPDAT)

■ **IAPDAT** (FEh) : IAP read/write Data Register

DAT.7	DAT.6	DAT.5	DAT.4	DAT.3	DAT.2	DAT.1	DAT.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

Appendix C : Update History

- V1.0
  - ✓ First Release.
- V1.1
  - ✓ Fixed ISP pin connections and IAP example.