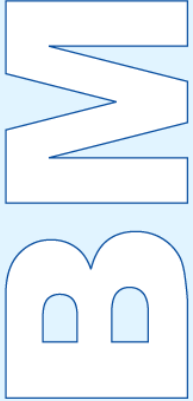




# ChargerCore Family

**BM-ChargerCore2.0-V1.3**



## Brief Manual of ChargerCore2.0 Family

### Optimized AS-MCU for Li-Ion Battery Charger

V1.3

October 2010

- ◆ CORERIVER Semiconductor reserves the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time.
- ◆ CORERIVER shall give customers at least a three month advance notice of intended discontinuation of a product or a service through its homepage.
- ◆ Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.
- ◆ The CORERIVER products listed in this document are intended for usage in general electronics applications. These CORERIVER products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury.

# Contents

1. Product Overview

2. Features

3. Block Diagram

4. Pin Configurations

5. Pin Descriptions

6. Function Descriptions

✓ CPU Descriptions

- Memory Organization
- SFR Map and Description
- Instruction Set Summary
- CPU Timing

✓ Peripheral Descriptions

- I/O Ports
- The ESD Structure of Pads
- LVD (Low Voltage Detector)
- WDT (Watchdog Timer)
- Timer0/1
- UART (Universal Async. RX/TX)
- PWM (Pulse Width Modulator)
- ADC
- C/C & C/V Controller
- Interrupt
- Reset Circuit
- Clock Circuit
- Power Management

7. Strong Points Compared to Conventional 80C52

8. Recommended Power Slope

9. Absolute Maximum Ratings

10. DC Characteristics

11. AC Characteristics

12. ADC Specifications

13. Package Dimensions

14. Supporting Tools

15. Typical Application Circuit

16. Appendix

A. Instruction Set

B. SFR Descriptions

C. Update History

# 1. Product Overview

- ◆ **CORERIVER's ChargerCore2.0 Family is a group of fast 80C52 compatible microcontrollers.**
- ◆ **The instruction execution of ChargerCore2.0 Family is max. 3 times faster than that of traditional 80C52.**
  - ✓ 1 machine cycle = 4 clocks vs. 12 clocks
- ◆ **Additional peripherals of ChargerCore2.0 Family:**
  - ✓ 10-bit ADC / 8-bit PWM / WDT / LVD / POR.
- ◆ **Power saving modes**
- ◆ **Noise tolerant scheme**
- ◆ **Provides User-Friendly MDS environment**
- ◆ **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>)**

# 1. Product Overview

## ◆ ChargerCore2.0 Family

Product	Mask-ROM (byte)	EPROM (byte)	RAM (Byte)	Volt (V)	Freq. (MHz)	T/C (16bits)	Serial I/O	WDT	ADC (bit x ch)	PWM (bit x ch)	Control	Package	Others	Available Time
ChargerCore20-ML20I ChargerCore20-TS20I	-	4k	128	2.4~5.5	20	2	1 UART	1	10x8	8x1	CC-CV	20-MLF 20-TSSOP	LVD POR Ring OSC	<b>Now</b>

\* Operating frequency of ChargerCore2.0 family is 20 MHz at 5.0 voltage.

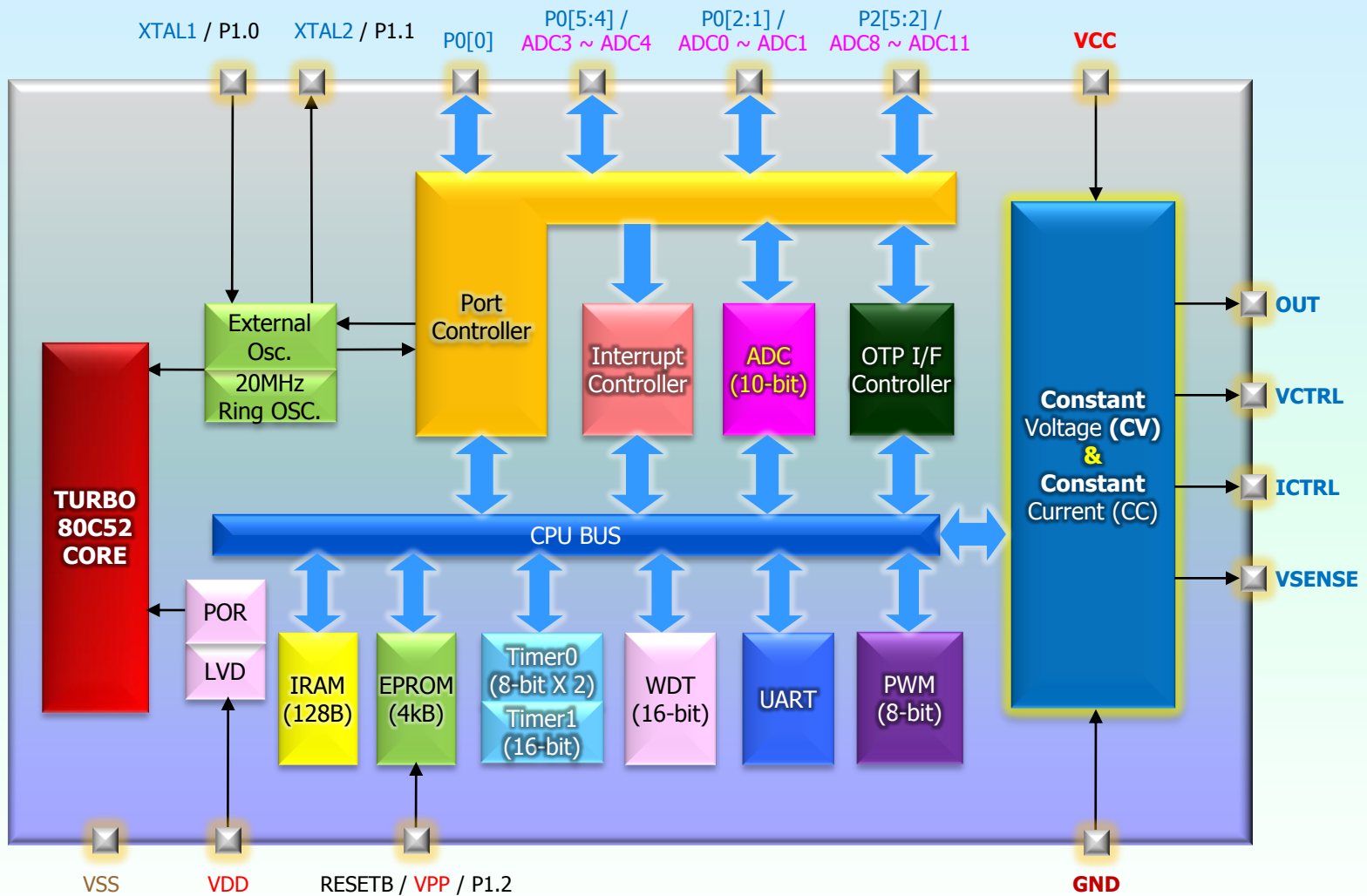
## 2. Features

- ◆ CPU
  - ✓ 8-bit turbo 80C52 architecture
  - ✓ 4 cycles/1 machine cycle
  - ✓ Instruction level compatible with Intel 80C52
- ◆ 4kBytes EPROM
- ◆ 128Bytes RAM
- ◆ Operating Voltage
  - ✓ MCU : +2.4V to +5.5V
  - ✓ CC/CV Controller : +2.5V to +12.0V
- ◆ Operating Frequency
  - ✓ Max. 20MHz @4.5V ~ 5.5V
  - ✓ Max. 10MHz @2.4V ~ 3.3V
- ◆ Operating Temperature
  - ✓ MCU : -40°C to 120°C
  - ✓ CC/CV Controller : 0°C to 85°C
- ◆ Max. Programmable 12 (20-MLF) I/O pins
  - ✓ Pull-up controlled by S/W
  - ✓ Push-pull output
- ◆ Low Voltage Detector : +2.3V
- ◆ Internal Ring OSC. : 3.6MHz@5V ( $\pm 15\%$ )
- ◆ 16-bit Programmable Watchdog Timer
- ◆ Two 16-bit Timer/Counters
- ◆ 8-channel 10-bit ADC
  - ✓ Max 100k SPS (@ $F_{ADC} = 8$  MHz)
  - ✓ Programmable input clock frequency
- ◆ 1-channel UART Comm.
- ◆ 9 interrupt sources
  - ✓ Timer0/1, WDT, ADC, PWM, UART
  - ✓ Three External Interrupt Sources : Both Edge/Level
  - ✓ Two-level interrupt priority
- ◆ Reset scheme
  - ✓ On-chip Power-On-Reset (POR)
  - ✓ External Reset
  - ✓ Low Voltage Detector Reset (LVR)
  - ✓ Watchdog Timer Reset
- ◆ Power Down Wake-up Sources
  - ✓ Reset Sources  
+ 2 External Interrupt INT0B/INT1B (Level Detect)
- ◆ Power Consumption
  - ✓ Active current : Max 10mA @+5.0V, 20MHz
  - ✓ Stop current : Max 1uA
- ◆ E.S.D. protection up to 2,000V
- ◆ Latch-up protection up to  $\pm 200$ mA
- ◆ Package
  - ✓ 20-MLF (5mm X 5mm)
  - ✓ 20-TSSOP



## 2. Features(cont'd)

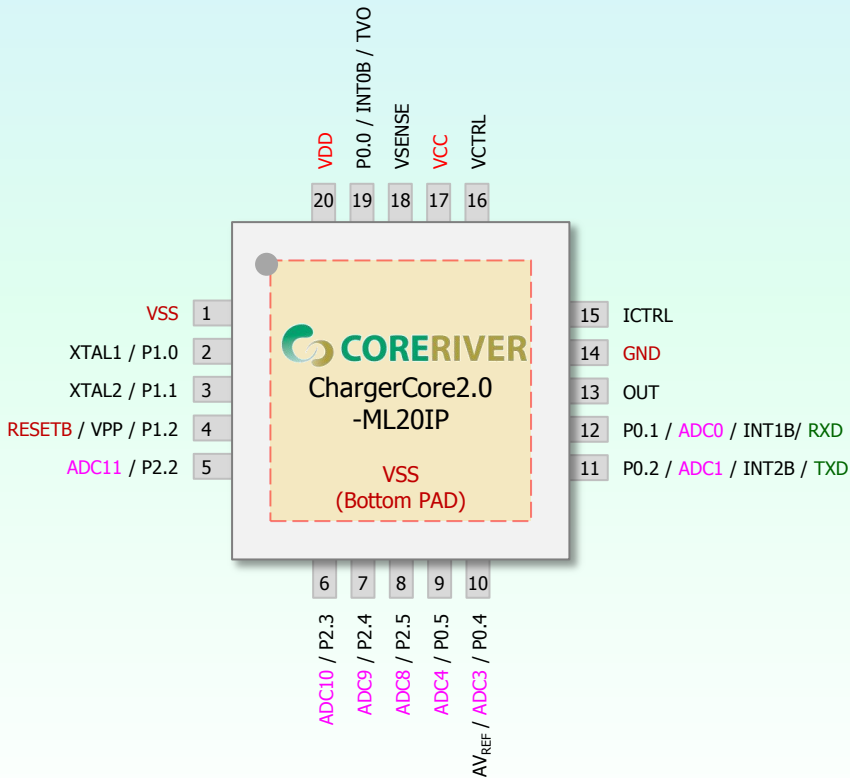
- ◆ Complete Charger for Li-Ion Batteries
- ◆ Very Low Thermal Dissipation
- ◆ CC (Constant Current) Profile
  - ✓ Charge Current : Dependent on Maker
  - ✓ ID
    - 1.5k $\Omega$  : 750mA  $\pm$  50mA
    - 4.7k $\Omega$  : 750mA  $\pm$  50mA
    - 27k $\Omega$  : 450mA  $\pm$  50mA
- ◆ CV (Constant Voltage) Profile
  - ✓ Charge Voltage : +4.2V (-0.05V / +0.03V)
- ◆ Wake-up
  - ✓ Wake-up Charging when the Initial Battery Voltage is below +2.5V
  - ✓ 30 Times Repeat  
(Charging during 5 sec, Standby during 2 sec)
  - ✓ Fast Charging at +2.5V above
- ◆ Current Cut-Off : 30mA to 100mA
- ◆ Recharge when below +4.07V  $\pm$ 0.05V
- ◆ Chargeable Ambient Temperature Range
  - ✓ -5 $^{\circ}$ C ( $\pm$ 3 $^{\circ}$ C) to 45 $^{\circ}$ C ( $\pm$ 3 $^{\circ}$ C)
- ◆ Charging Time
  - ✓ Fast Charge : 4 Hours
  - ✓ Recharge : 2 Hours
- ◆ Over-Voltage Protection
  - ✓ Charge Cut-off when +4.35V ( $\pm$ 0.05) above
- ◆ Error Mode
  - ✓ When Battery Voltage is greater than +4.35V
  - ✓ Short or Open in Charging
  - ✓ Detecting Abnormal Battery
- ◆ LED Display
  - ✓ Red : in Charging or in Wake-up Mode
  - ✓ Green : Charge Completion or in Recharging
  - ✓ Yellow : Stand-by due to Temperature Emergency
  - ✓ Yellow Flickering : in Error Mode
- ◆ Korea TTA Specification Certification

# 3. Block Diagram

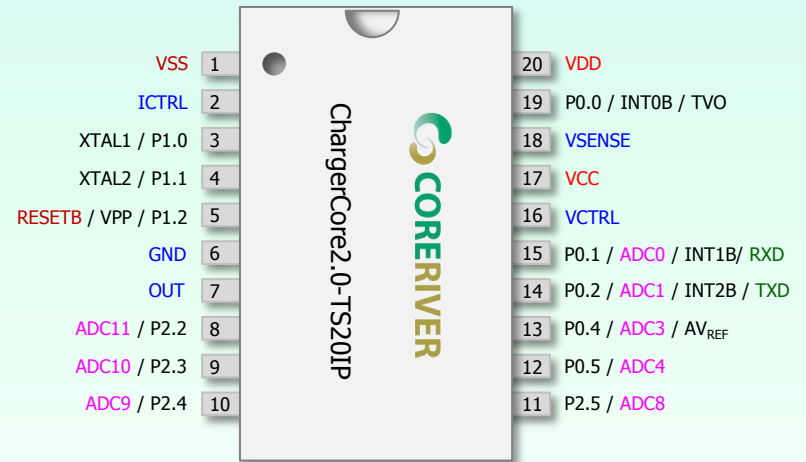


# 4. Pin Configurations

-  ADC channel : 8 Channels
-  UART Comm. : 1 Channel
-  I/O : 12 Pins



[ 20-pin MLF : 5mm X 5mm, 0.65mm Pin Pitch ]



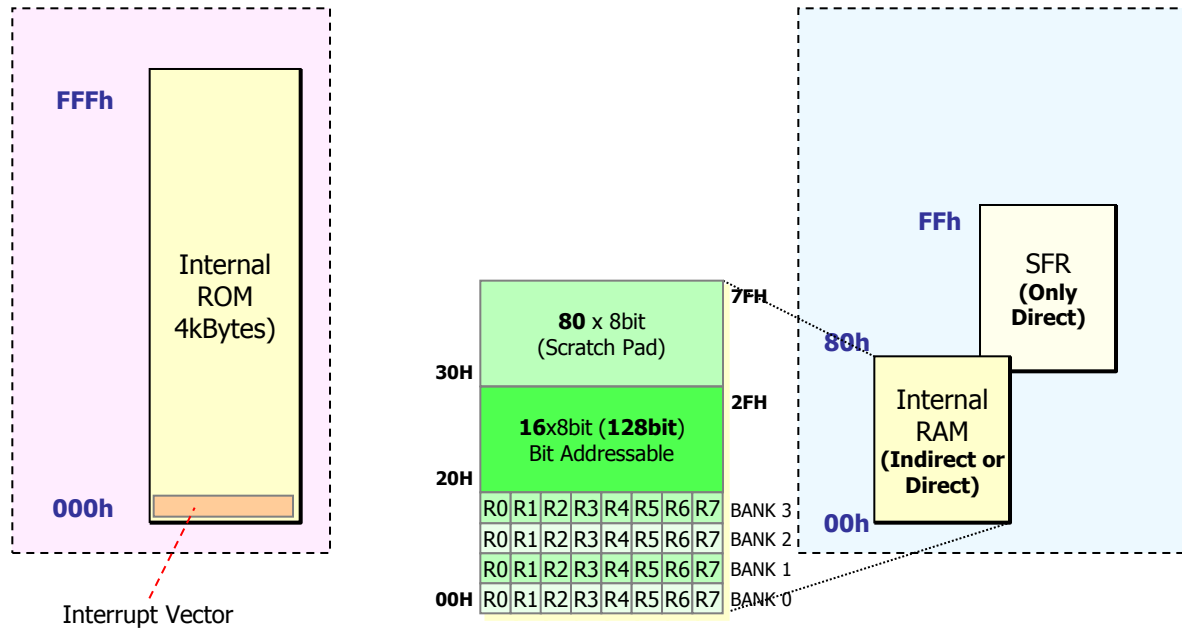
[ 20-pin TSSOP : JEDEC ]



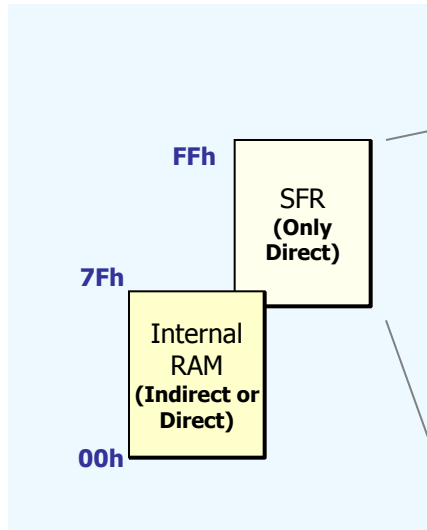
## 5. Pin Descriptions

Pin No.	Pin Name	Direction	Description	Share Pins
1	VSS	Power	Voltage Power Ground for MCU	
2	XTAL1 / P1.0	I/O	<ul style="list-style-type: none"> <li>Crystal Input/Output (Default)</li> <li>Bit Programmable with Schmitt Trigger</li> </ul>	Crystal Input
3	XTAL2 / P1.1	I/O	<ul style="list-style-type: none"> <li>- Optional Pull-up Control Enable</li> <li>- Push-pull Output</li> </ul>	Crystal Output
4	RESETB / VPP / P1.2	I/O	<ul style="list-style-type: none"> <li>External Reset Input Signal (Default)</li> <li>Bit Programmable</li> </ul>	VPP (+11.5V)
5	P2.2 / ADC11	I/O	<ul style="list-style-type: none"> <li>Bit Programmable with Schmitt Trigger</li> <li>- Optional Pull-up Control Enable</li> <li>- Push-pull Output (Default)</li> </ul>	
6	P2.3 / ADC10	I/O		
7	P2.4 / ADC9	I/O		
8	P2.4 / ADC8	I/O		
9	P0.5 / ADC4	I/O	<ul style="list-style-type: none"> <li>Bit Programmable with Schmitt Trigger</li> <li>- Optional Pull-up Control Enable</li> <li>- Push-pull Output (Default)</li> </ul>	
10	P0.4 / ADC3	I/O		AV <sub>REF</sub>
11	P0.2 / ADC1	I/O	<ul style="list-style-type: none"> <li>Bit Programmable with Schmitt Trigger</li> <li>- Optional Pull-up Control Enable</li> <li>- Push-pull Output (Default)</li> </ul>	INT2B / TXD
12	P0.1 / ADC0	I/O		INT1B / RXD
13	OUT	O	Output pin. Sinking Current Only	
14	GND	Power	Voltage Power Ground for CC / CV Control	
15	ICTRL	I	Input Pin of the Current Control Loop	
16	VCTRL	I	Input Pin of the Voltage Control Loop	
17	VCC	Power	Voltage Power Source for CC / CV Control	
18	VSENSE	I	Input Pin of the Current Control Loop	
19	P0.0	I/O	<ul style="list-style-type: none"> <li>Bit Programmable with Schmitt Trigger</li> <li>- Optional Pull-up Control Enable</li> <li>- Push-pull Output (Default)</li> </ul>	INT0B / TVO
20	VDD	Power	Voltage Power Source for MCU	

# 6.1. Memory Organization



## 6.2. SFR (Special Function Register) Map



Bit addressable

Legend:  
 : Newly added SFR at ChargerCore2.0 Family  
 : Reserved for future use.

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

## 6.2. SFR Brief Description

### ◆ 80C52 SFR Registers

Register	Name	Reset Value
<b>ACC</b> <b>B</b> <b>PSW</b> <b>SP</b>	Accumulator B Program Status Word Stack Pointer	00000000 00000000 00000000 00000111
<b>DPTR</b> <b>DPL</b> <b>DPH</b>	Data Pointer (2 bytes) Low Byte High Byte	00000000 00000000
<b>P0</b> <b>P1</b> <b>P2</b>	Port 0 Port 1 Port 2	11111111 *****111 *1111111
<b>IP</b> <b>IE</b>	Interrupt Priority Interrupt Enable Control	10*00000 00*00000
<b>TCON</b> <b>TMOD</b>	Timer/Counter 0/1 Control Timer/Counter 0 Mode Control	00000000 ****0000
<b>TH0</b> <b>TLO</b> <b>TH1</b> <b>TL1</b>	Timer/Counter 0 High Byte Timer/Counter 0 Low Byte Timer/Counter 1 High Byte Timer/Counter 1 Low Byte	00000000 00000000 00000000 00000000
<b>SCON</b> <b>SBUF</b>	Serial Control Serial Buffer	***0**00 00000000
<b>PCON</b>	Power Control	0**10000

### ◆ Newly added SFR Registers in ChargerCore2.0 Family

Register	Name	Reset Value
<b>P0SEL</b> <b>P1SEL</b> <b>P2SEL</b>	Port 0 Pull-up Control Port 1 Pull-up Control Port 2 Pull-up Control	00000000 *****11 *0000000
<b>P0TYPE</b> <b>P1TYPE</b> <b>P2TYPE</b>	Port 0 Type Control Port 1 Type Control Port 2 Type Control	00000000 *****00 *0000000
<b>P0DIR</b> <b>P1DIR</b> <b>P2DIR</b>	Port 0 Input/Output Control Port 1 Input/Output Control Port 2 Input/Output Control	11111111 *****111 *1111111
<b>ALTSEL</b>	Alternative Function Control	000000**
<b>PWMCON</b> <b>PWMD</b>	PWM Control PWM Duty Data	0000*000 00000000
<b>ADCON</b> <b>ADCR</b> <b>ADCSEL</b> <b>ADCSELH</b>	ADC Control & ADC Result Low ADC Result High ADC Channel Selection Low and MUX Selection ADC Channel Selection High	00100000 00000000 11111111 11111111
<b>WDCON</b> <b>PMR</b> <b>EXIF</b> <b>EIP</b> <b>EIE</b> <b>STATUS</b> <b>OSCICN</b>	Watchdog Timer Control Power Management Control Added External Interrupt and LVD Control Extended Interrupt Priority Extended Interrupt Enable Crystal Status Internal Ring Oscillator Control	11010000 ****0*** **000101 **00**00 **00**00 ***0*** ****100

\* : Not Used Bit.

## 6.3. Instruction Set Summary

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

Type	Instruction	Description
<b>Arithmetic</b>	ADD	Addition
	ADDC	Addition with Carry
	SUBB	Subtraction with Borrow
	INC	Increment
	DEC	Decrement
	MUL	Multiply
	DIV	Divide
	DA	Decimal Adjust
<b>Logical</b>	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	
<b>Data Transfer</b>	MOV	Move Data
	MOVC	Move Code
	PUSH	PUSH
	POP	POP
	XCH	Exchange
	XCHD	Exchange Low-digit

Type	Instruction	Description
<b>Boolean</b>	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	
<b>Branch</b>	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	



## 6.4. CPU Timing : Execution Time Table

- ◆ Fastest instruction execution time in the world

Instruction	ChargerCore2.0 (CORERIVER)	W77C32 (Winbond)	DS80C320 (Maxim)	87C52 (Intel)
MUL AB DIV AB	<b>12 clocks</b>	20 clocks	20 clocks	48 clocks
MOVC A, @A+PC MOVC A, @A+DPTR	<b>8 clocks</b>	8 clocks	12 clocks	24 clocks
JMP @A+DPTR	<b>8 clocks</b>	8 clocks	12 clocks	24 clocks
RET RETI	<b>8 clocks</b>	8 clocks	16 clocks	24 clocks
INC DPTR	<b>4 clocks</b>	8 clocks	12 clocks	24 clocks
Others	<b>Same</b>	Same	Same	-

## 6.5. I/O Ports : PORT0[5:4, 2:0]

- ◆ Pull-up enable with push-pull output type by default.
- ◆ P0[5:4, 2:1] can be configured as ADC4 ~ ADC3, ADC1 ~ ADC0 input.
- ◆ 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

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

ADC3B	-	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

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

-	-	P0TYPE.5	P0TYPE.4	-	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)

### ✓ PODIR (F4h) : Port 0 Input/Output Control Register

-	-	PODIR.5	PODIR.4	-	PODIR.2	PODIR.1	PODIR.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)

### ✓ POSEL (E4h) : Port 0 Pull-up Control Register

-	-	POSEL.5	POSEL.4	-	POSEL.2	POSEL.1	POSEL.0
---	---	---------	---------	---	---------	---------	---------

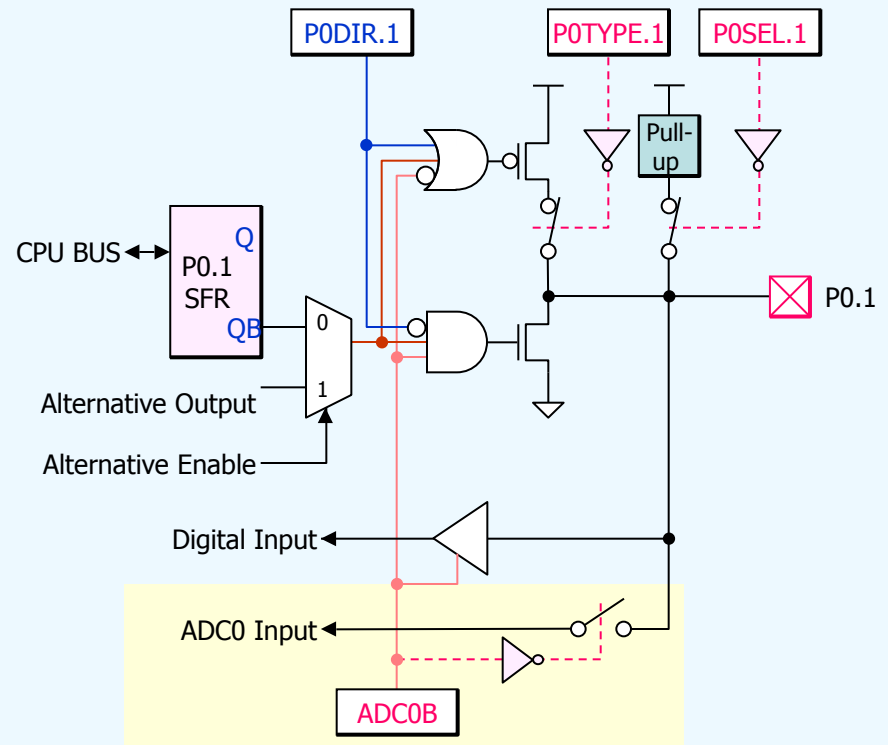
R/W(0) R/W(0) R/W(0) R/W(0) R/W(0) R/W(0) R/W(0) R/W(0)

- 0 = Pull-up resistor ON (Default)
- 1 = Pull-up resistor OFF when ADC\_EN (ADCON[7]) = 1

### ✓ P0 (80h) : Port 0 Register

-	-	P0.5	P0.4	-	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)

- ◆ 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.1	P1TYPE.0
---	---	---	---	---	---	----------	----------

R/W(0) R/W(0)

- 0 = Push-pull Output (Default)

### ✓ 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.1	P1SEL.0
---	---	---	---	---	---	---------	---------

R/W(1) R/W(1)

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

### ✓ P1 (90h) : Port 1 Register

-	-	-	-	-	P1.2	P1.1	P1.0
---	---	---	---	---	------	------	------

R/W(1) R/W(1) R/W(1)

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

IOXEN	IORSTEN	-	PWM00	TV0	TX	-	-
-------	---------	---	-------	-----	----	---	---

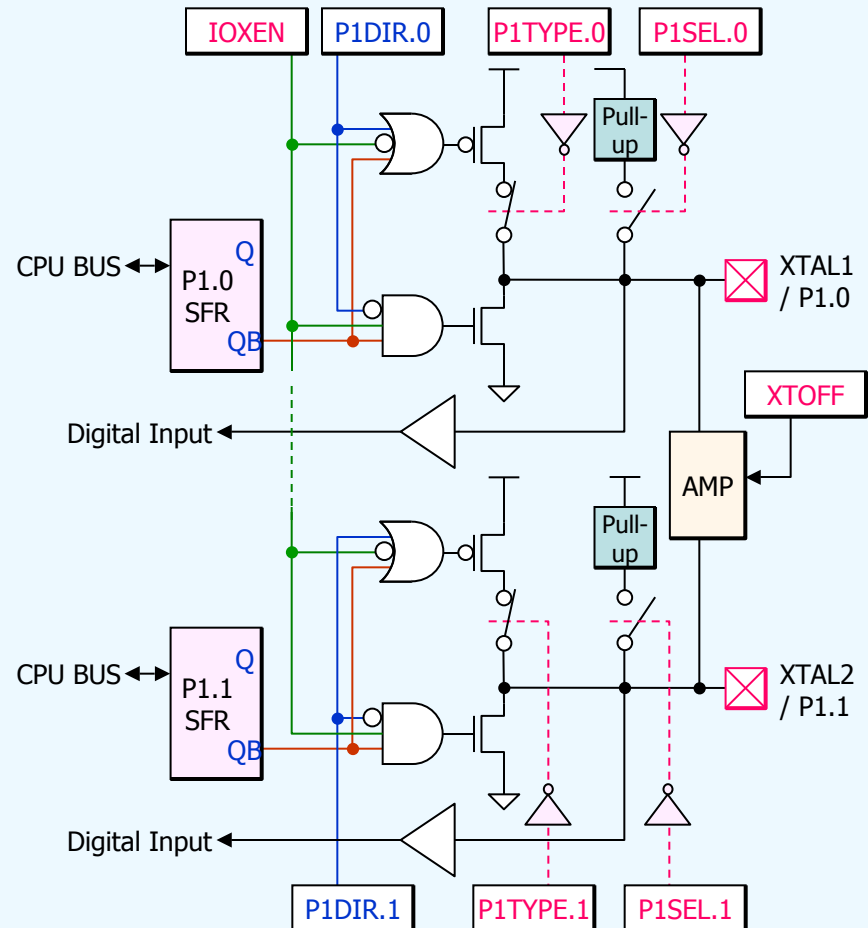
R/W(0) R/W(0) R/W(0) R/W(0) R/W(0) R/W(0)

- 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)

- ◆ 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

### ✓ 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)

### ✓ P1 (90h) : Port 1 Register

-	-	-	-	-	P1.2	P1.1	P1.0
---	---	---	---	---	------	------	------

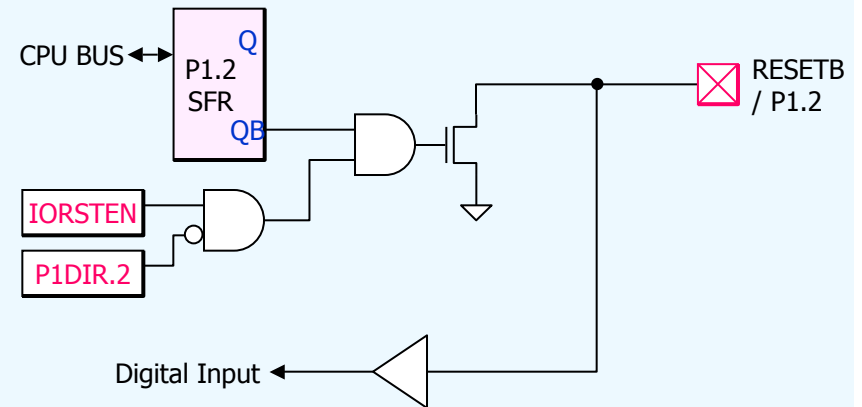
R/W(1) R/W(1) R/W(1)

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

IOXEN	IORSTEN	-	PWM00	TV0	TX	-	-
-------	---------	---	-------	-----	----	---	---

R/W(0) 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[5:2]

- ◆ Pull-up enable with push-pull type by default.
- ◆ P2[5:2] can be configured as ADC11 ~ ADC8 input.
- ◆ 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	-	-	-	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.5	P2TYPE.4	P2TYPE.3	P2TYPE.2	-	-
---	---	----------	----------	----------	----------	---	---

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)

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

-	-	P2DIR.5	P2DIR.4	P2DIR.3	P2DIR.2	-	-
---	---	---------	---------	---------	---------	---	---

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.5	P2SEL.4	P2SEL.3	P2SEL.2	-	-
---	---	---------	---------	---------	---------	---	---

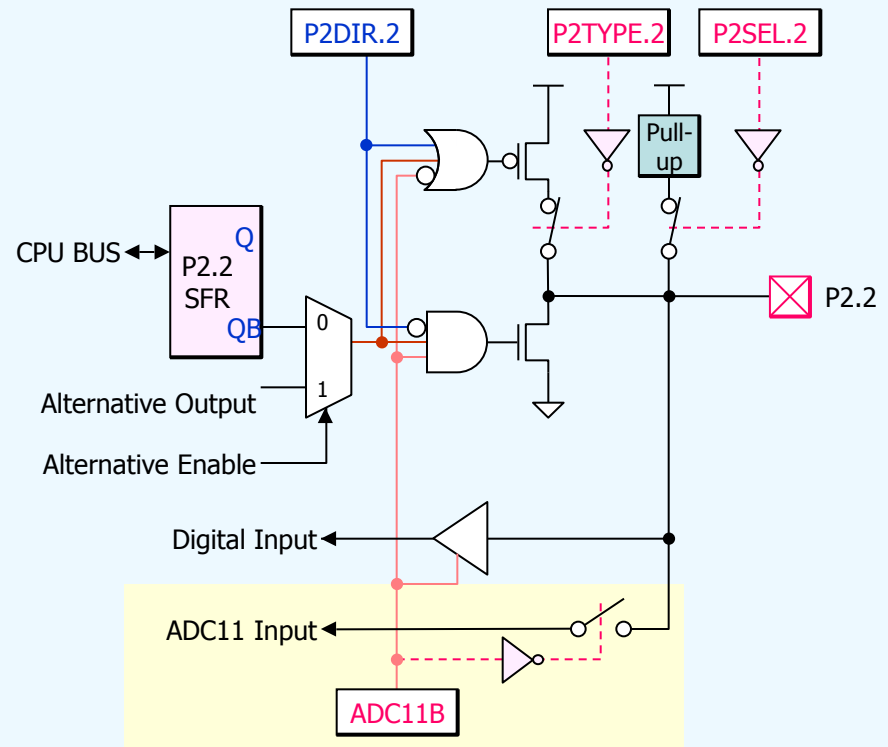
R/W(0) R/W(0) R/W(0) R/W(0) R/W(0) R/W(0) R/W(0) R/W(0)

- 0 = Pull-up resistor ON
- 1 = Pull-up resistor OFF when ADC\_EN(ADCON[3]) = 1.

### ✓ P2 (A0h) : Port 2 Register

-	-	P2.5	P2.4	P2.3	P2.2	-	-
---	---	------	------	------	------	---	---

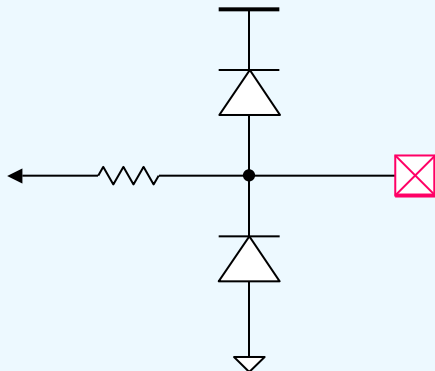
R/W(1) R/W(1) R/W(1) R/W(1) R/W(1) R/W(1) R/W(1) R/W(1)



## 6.6. The ESD Structure of Pads

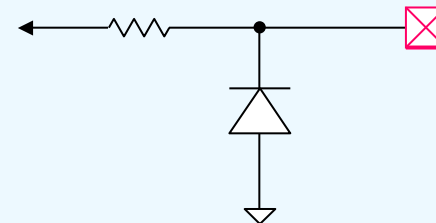
- ◆ Two ESD diodes and one ESD resistor are contained in all pads except VPP/P1.2/RESETB.
- ◆ One ESD diode and one ESD resistor are contained in VPP/P1.2/RESETB.

[All pads except VPP/P1.2/RESETB]



- Two ESD Diodes ( $V_{DD}$  side, GND side)
- One ESD Resistor

[VPP/P1.2/RESETB]



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

## 6.7. LVD (Low Voltage Detector)

- ◆ On-chip power-on reset : 2.3V
- ◆ On-chip power-fail reset : 2.3V

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

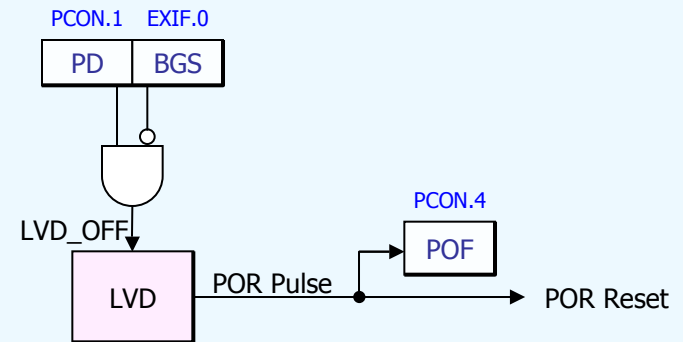
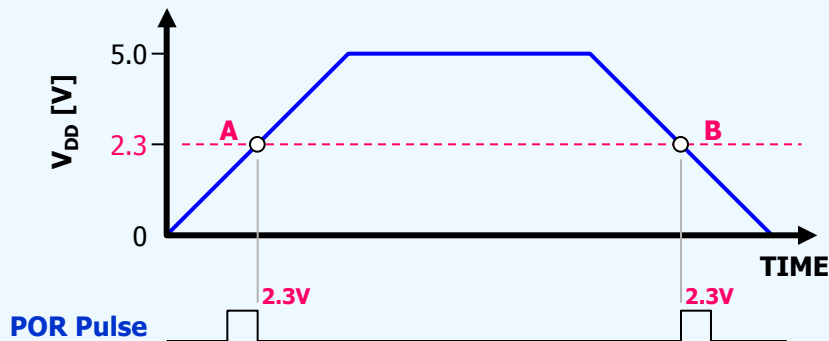
-	-	-	IE2	XT/RG	RGMD	RGSL	BGS
R/W(0)	R/W(0)	R/W(0)	R(1)	R/W(0)	R/W(1)		

- BGS : Band-gap Select  
0 = Band-gap block (LVD) Off in stop mode, but ON during normal mode.  
1 = Band-gap block (LVD) ON in power-down mode.

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

SMOD1	-	-	POF	GF1	GF0	PD	IDL
R/W(0)			R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- POF : Power Off flag. When power-on, POF = 1 by H/W.
- PD : Power-down mode bit



## 6.8. WDT (Watchdog Timer)

- ◆ Detect the malfunction of program due to external noise or other causes.
- ◆ Return the operation to the normal condition using WDT interrupt.
- ◆ If enabled, WDT interrupt or WDT reset makes MCU wake up from Stop Mode 2.
- ◆ Watchdog Time-out Values

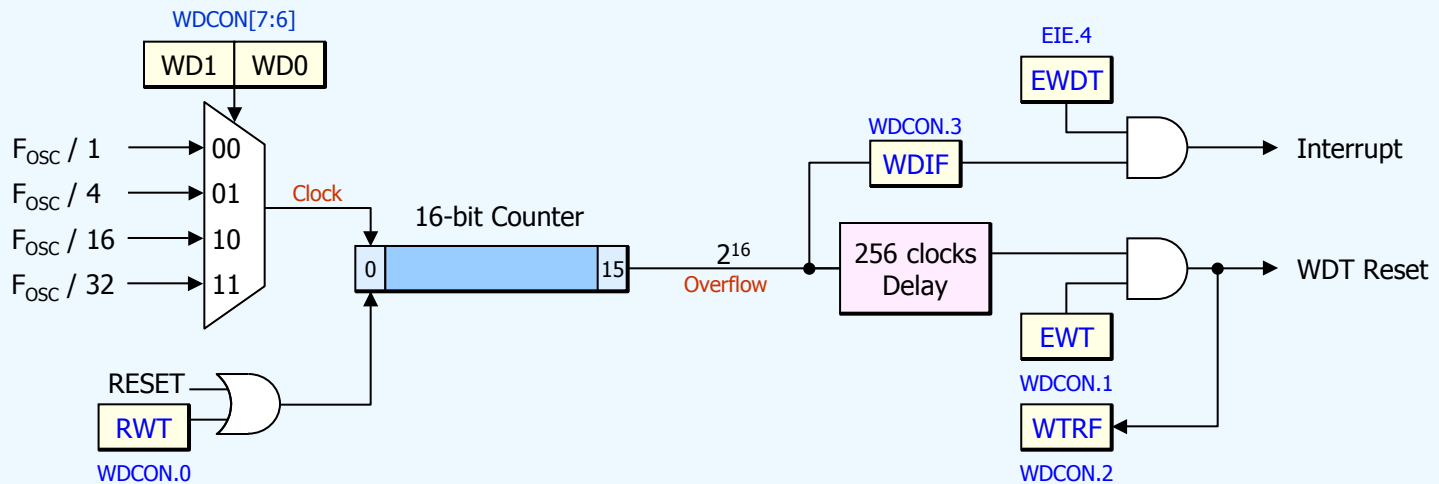
WD1	WD0	Interrupt Time-out (@4MHz)		Reset Time-out (@4MHz)	
0	0	1x2 <sup>16</sup> clocks	16.38 ms	1x2 <sup>16</sup> + 256 clocks	16.45 ms
0	1	4x2 <sup>16</sup> clocks	65.54 ms	4x2 <sup>16</sup> + 256 clocks	65.60 ms
1	0	16x2 <sup>16</sup> clocks	262.14 ms	16x2 <sup>16</sup> + 256 clocks	262.21 ms
1	1	32x2 <sup>16</sup> clocks	524.29 ms	32x2 <sup>16</sup> + 256 clocks	524.35 ms

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

WD1	WD0	-	-	WDIF	WTRF	EWT	RWT
-----	-----	---	---	------	------	-----	-----

R/W(1) R/W(1) R/W(0) R/W(1) R/W(0) R/W(0) R/W(0) R/W(0)

- WD[1:0] : WDT Clock Divide(1/4/8/32)
- WDIF : Watchdog Timer Interrupt Flag
- WTRF : Watchdog Timer Reset Flag. Only cleared by S/W.
- EWT : Watchdog Timer Reset Enable
- RWT : Restart Watchdog Timer



\* RWT is only used with WDT mode 0 (WD[1:0] = [0,0]) for MiDAS1.1 Family (Refer to Application Note #009 (AN009))

## 6.9. Timer/Counter : Timer 0/1

- ◆ 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[3] : Timer 0 Gate Control
- C/T[2] : Timer 0 Counter/Timer Select.  
0 = Timer by  $F_{OSC}/12$ . (Default)  
1 = not support
- 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 Type Select Flag  
Edge Detect (IT1=1). Level Detect (IT1=0)
- IE0 : External Interrupt 0 Flag
- IT0 : External Interrupt 0 Type Select Flag  
Edge Detect (IT0=1). Level Detect (IT0=0)

### ✓ TLO (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)

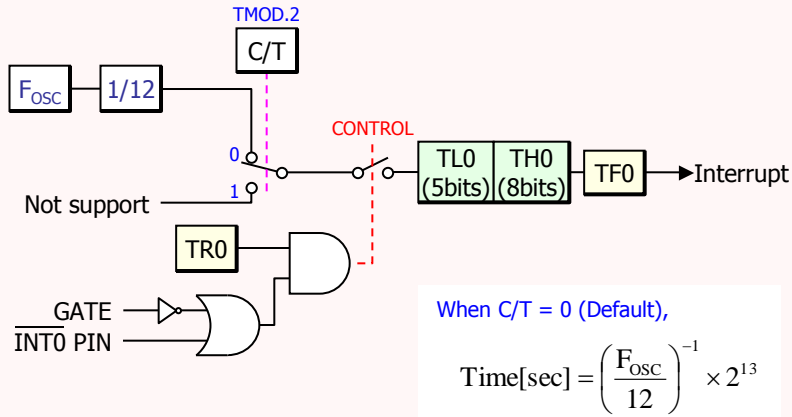
### ✓ TL1 (8Bh) : Timer/Counter 1 Low Byte Register

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)

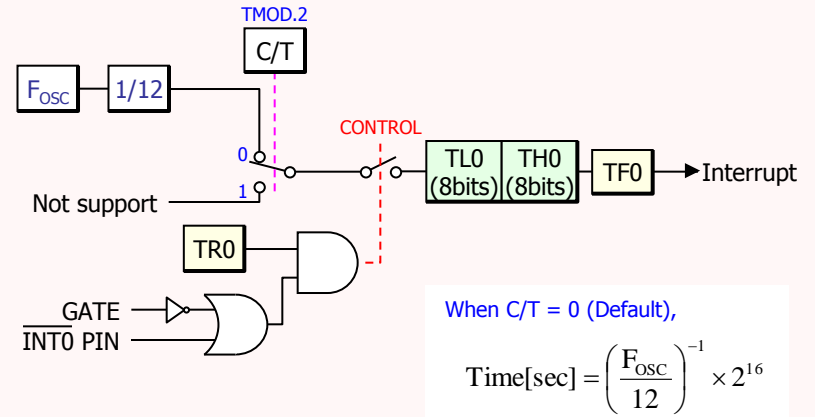
### ✓ TH1 (8Dh) : Timer/Counter 1 High Byte Register

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)

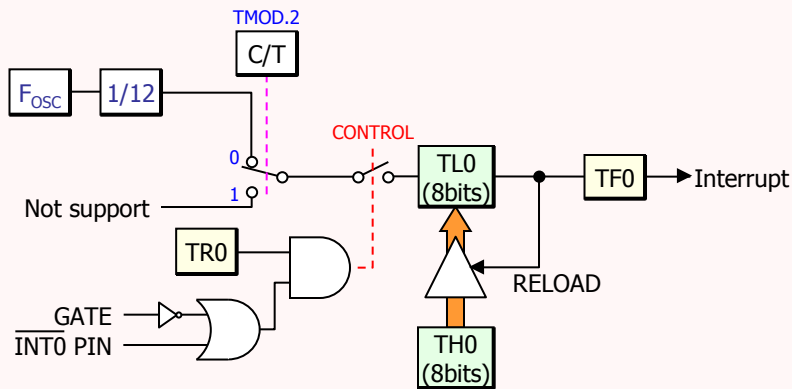
# 6.9. Timer/Counter : Timer 0 Mode Description



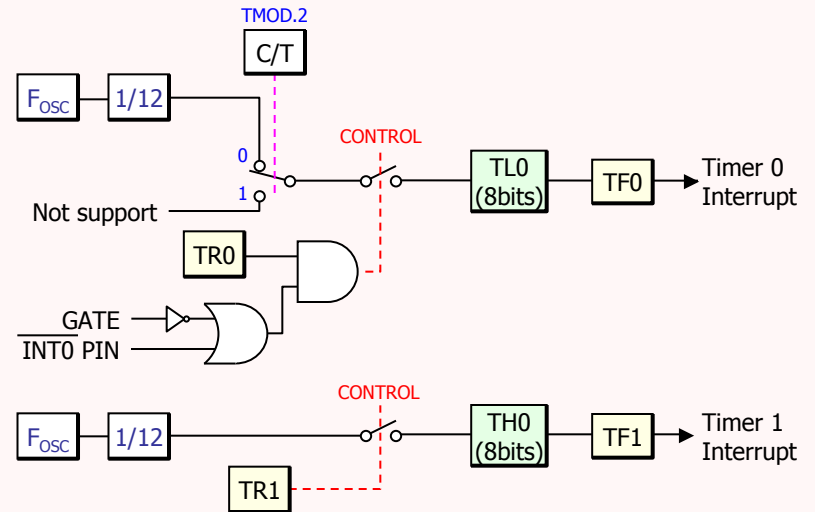
[Mode 0]



[Mode 1]



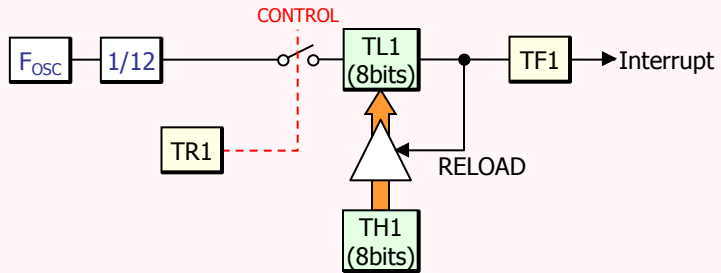
[Mode 2]



[Mode 3]



## 6.9. Timer/Counter : Timer 1 Mode Description



[Mode 2]

# 6.10. UART

◆ Simplified 8052 UART  
( only UART Mode 1 is supported.)

	Data Size		Baudrate
	Mode 1	10 bits	Start bit(0) 8 data bit Stop bit(1)

◆ UART Mode 1  
(Using Timer 1 Overflow)

$$\text{Baudrate} = \frac{2^{\text{SMOD1}}}{32} \times F_{\text{osc}} \times \frac{1}{12 \times [256 - (\text{TH1})]}$$

[Baudrate Examples]

Baudrate	UART Mode	F <sub>osc</sub> [MHz]	SMOD1	Timer 1		
				C/T	Mode	Reload Value (TH1)
62.5 KHz	Mode 1	12	1	0	Mode 2 8-bit Auto-reload	FFh
19.2 KHz		11.0592	1	0		FDh
9.6 KHz		11.0592	0	0		FDh
4.8 KHz		11.0592	0	0		FAh
2.4 KHz		11.0592	0	0		F4h
1.2 KHz		11.0592	0	0		E8h
137.5 Hz		11.0592	0	0		1Dh
110 Hz		6	0	0		72h

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

SMOD1	-	-	POF	GF1	GF0	PD	IDL
R/W(0)			R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- SMOD1 : Timer 1 baudrate double in UART mode.

✓ **SCON** (98h) : Serial Port Control Register

-	-	-	REN	-	-	TI	RI
			R/W(0)			R/W(0)	R/W(0)

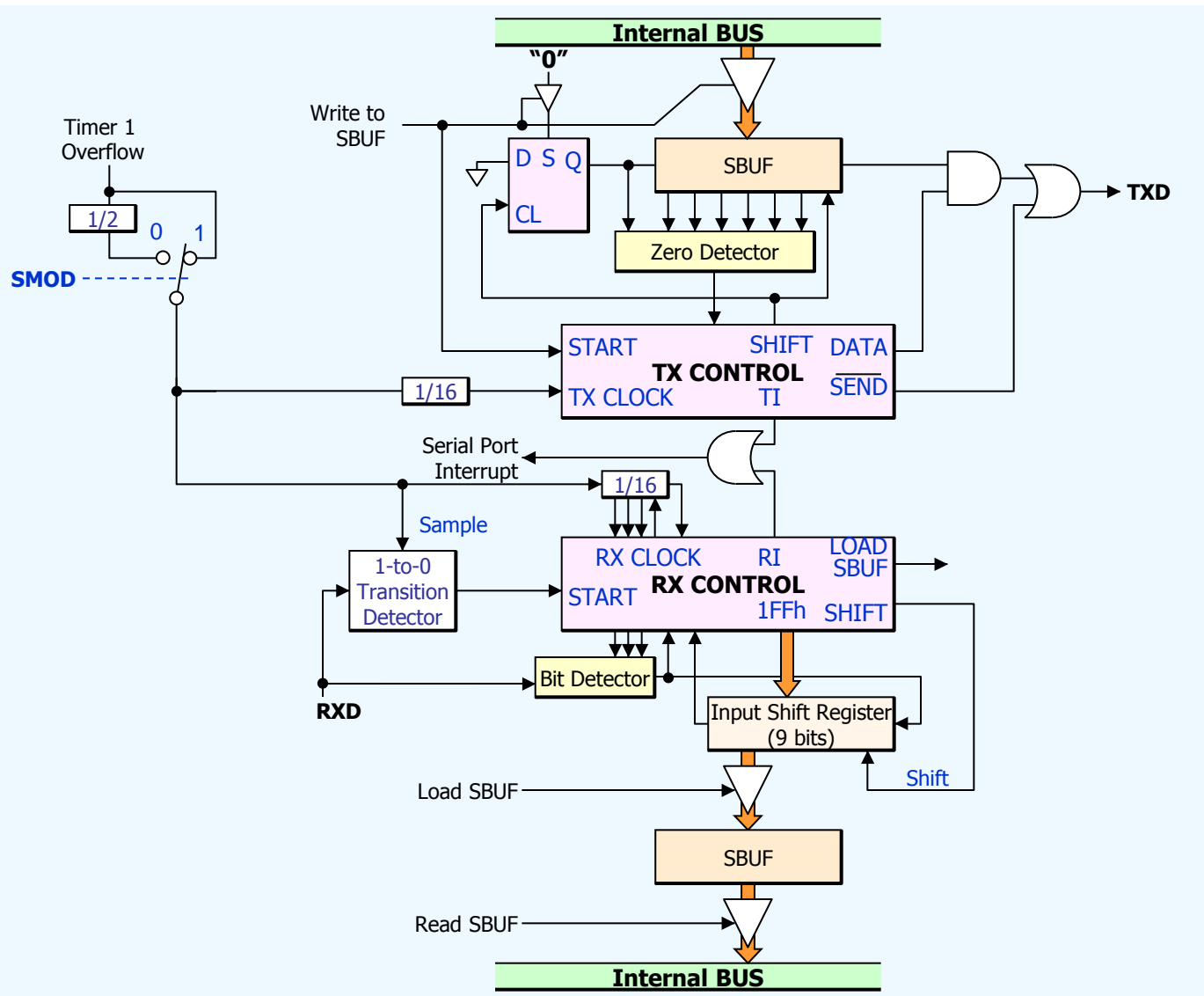
- REN : Serial Reception Enable.
- TI : Transmission Interrupt Flag. Must be cleared by S/W.
- RI : Reception Interrupt Flag. Must be cleared by S/W.

✓ **SBUF** (99h) : Serial Data Buffer Register

SBUF.7	SBUF.6	SBUF.5	SBUF.4	SBUF.3	SBUF.2	SBUF.1	SBUF.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)

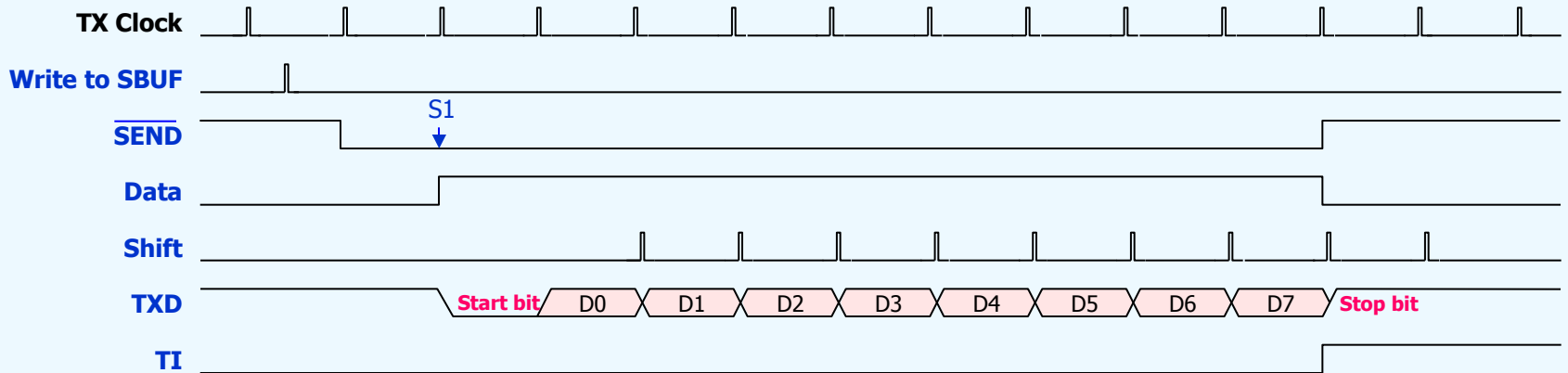
- Transmission buffer and reception buffer are separated.
- Read and Write address are same.

# 6.10. UART : Mode 1 Function

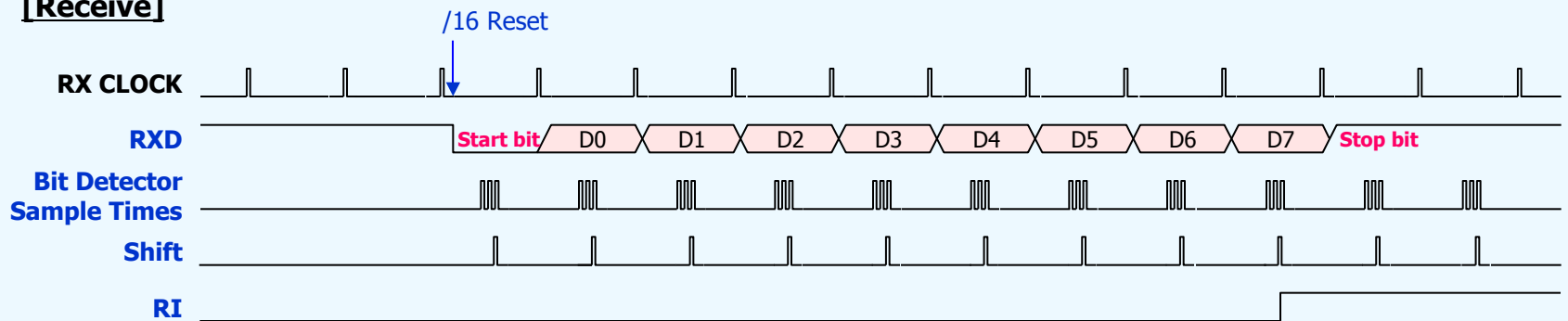


## 6.10. UART : Mode 1 Timing

### [Transmit]

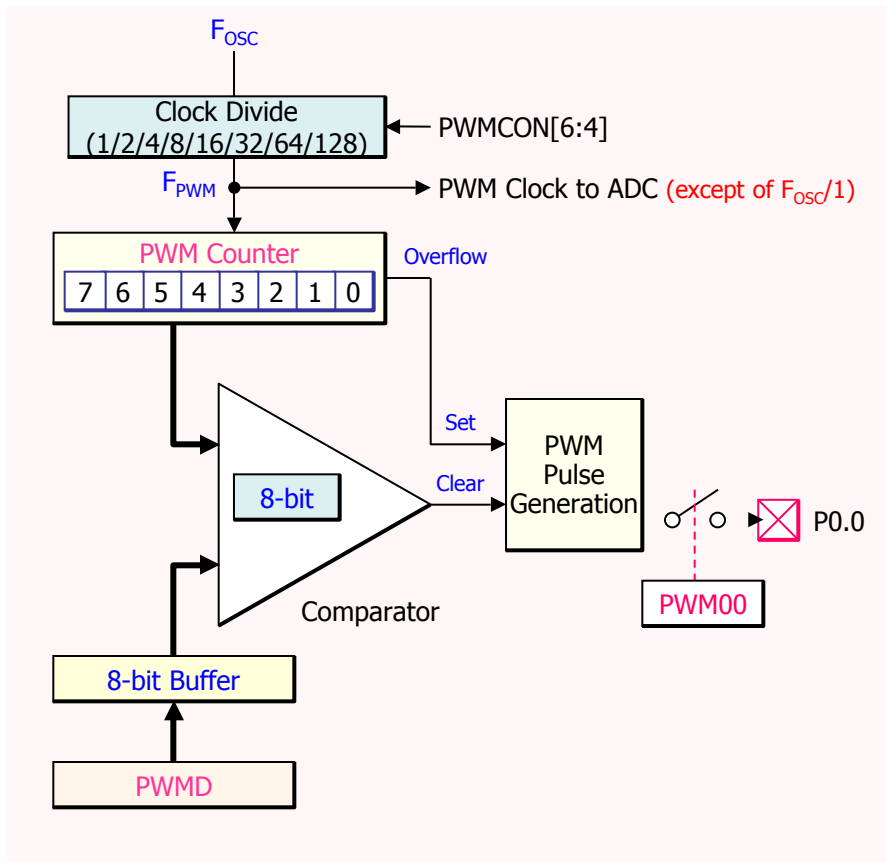


### [Receive]



# 6.11. PWM (Pulse Width Modulator)

- ◆ 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

-	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)

- 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] =  $F_{osc}/128$
- \* PWM Clock ( $F_{PWM}$ ) to ADC should not be set to  $F_{osc}/1$ .
- 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

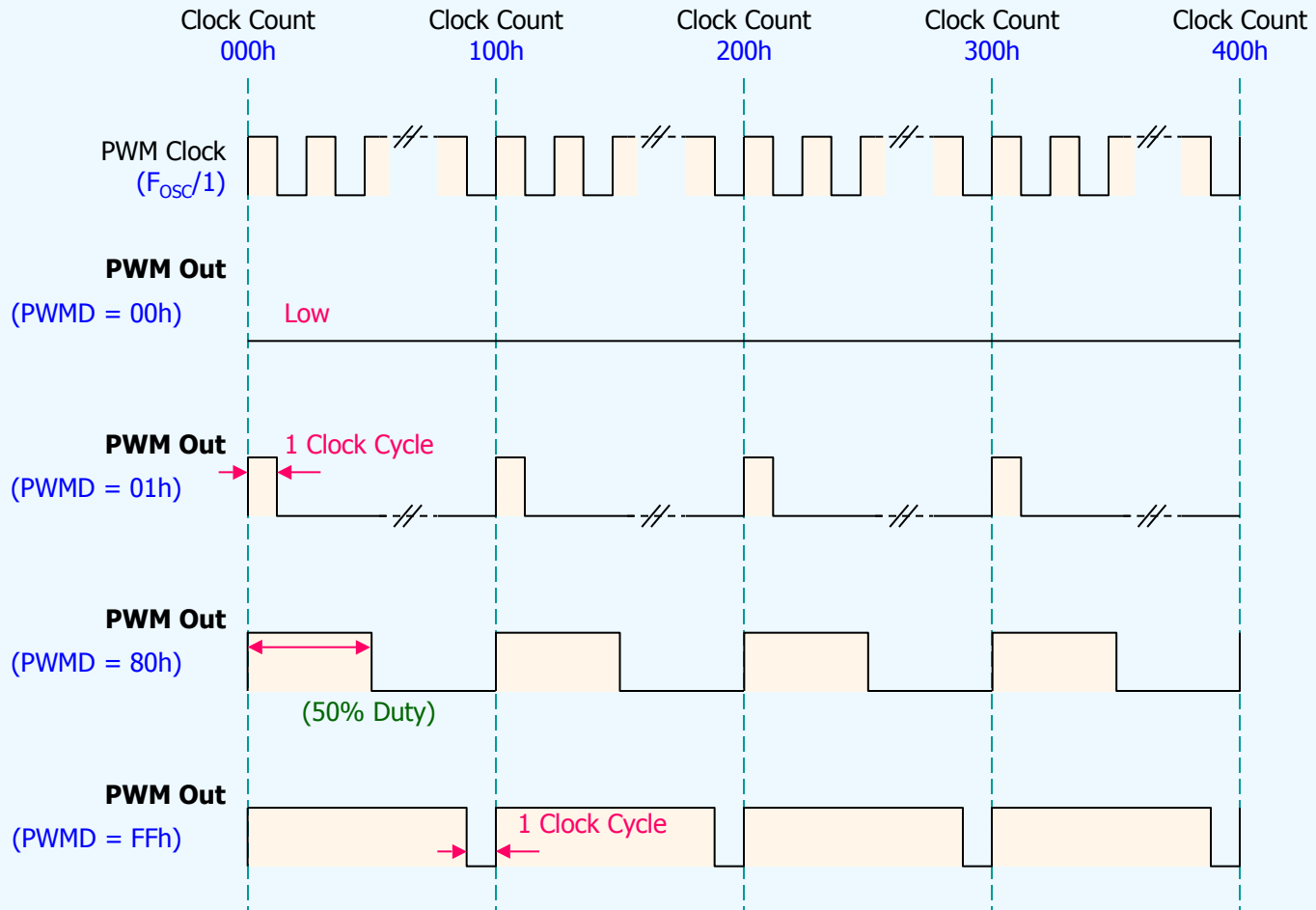
PWMD.7	PWMD.6	PWMD.5	PWMD.4	PWMD.3	PWMD.2	PWMD.1	PWMD.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

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

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

- PWM00 : PWM Waveform Output Enable to P0.0

# 6.11. PWM : Pulse Generation



# 6.12. ADC (Analog-to-Digital Converter)

- ◆ 8-channel 10-bit ADC (SAR Type)
- ◆ Max. 104ksps(samples per sec.) @  $F_{ADC} = 10\text{MHz} \ \& \ 5\text{V}$ . (Max. 52ksps @  $F_{ADC} = 5\text{MHz} \ \& \ 3\text{V}$ )

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

ADC11B	ADC10B	ADC9B	ADC8B	-	-	-	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**

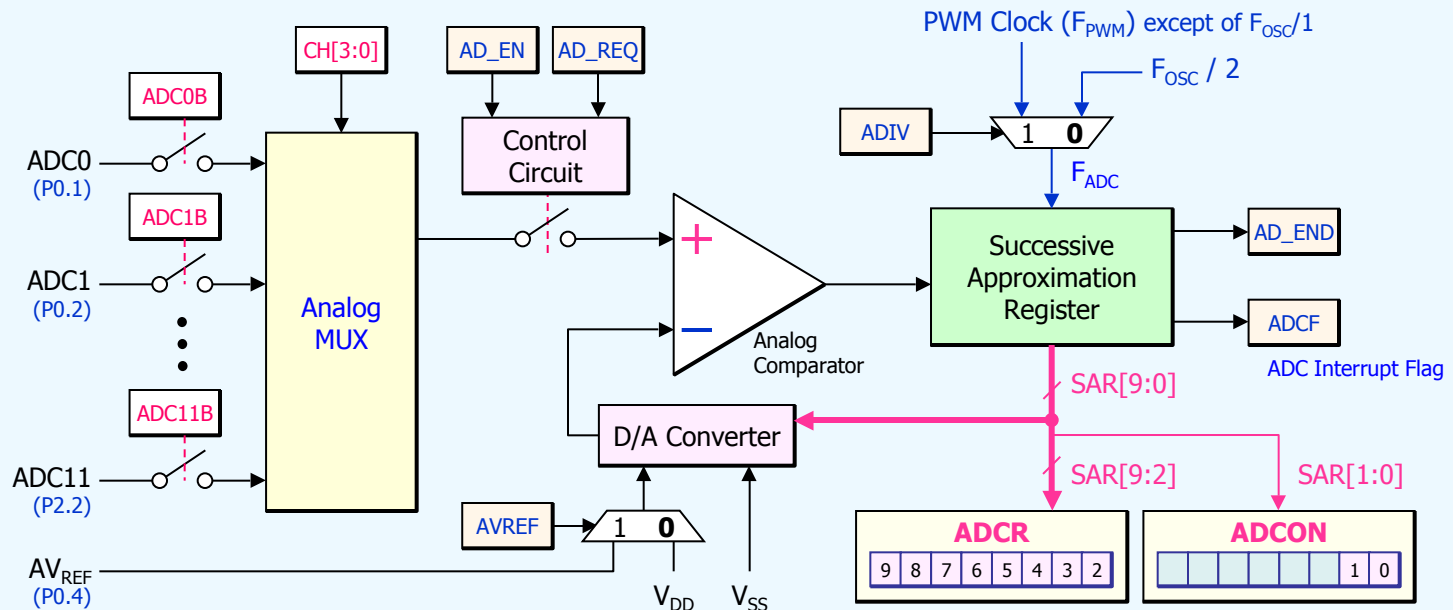
ADC3B	-	ADC1B	ADC0B	CH3	CH2	CH1	CH0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

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

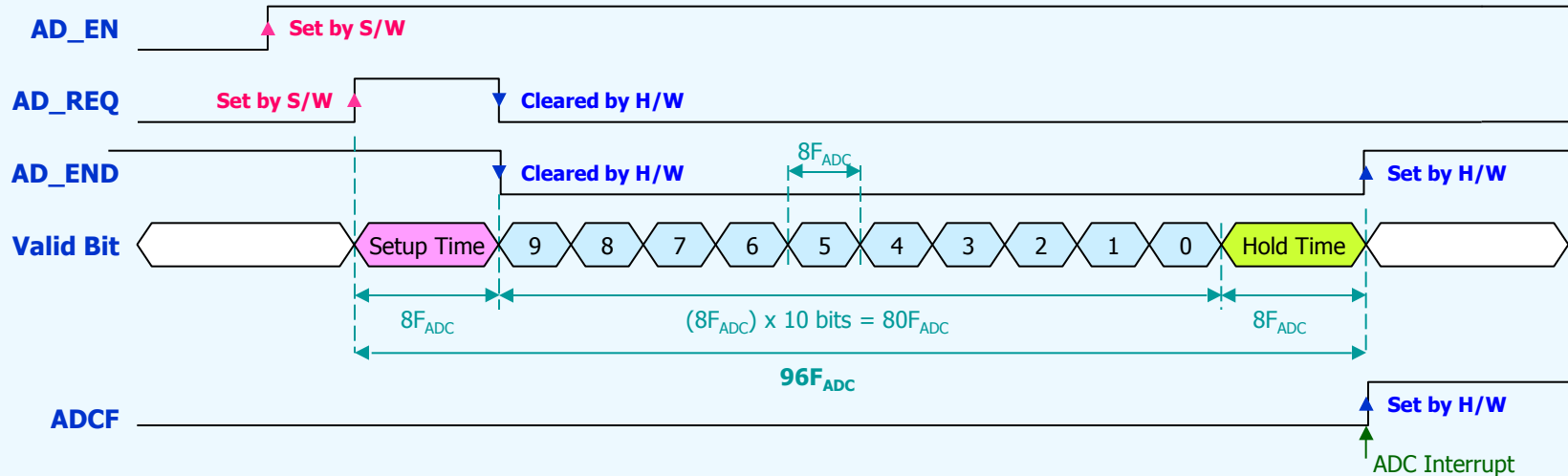
AD_EN	AD_REQ	AD_END	ADCF	AVREF	ADIV	SAR1	SAR0
R/W(0)	R/W(0)	R(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

✓ **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)



## 6.12. ADC : Conversion Timing



- ✓ **AD\_EN** : ADC Block Enable Signal.  
Set or Cleared by S/W.
- ✓ **AD\_REQ** : ADC Conversion Request Start 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** : ADC Interrupt Flag.  
Set by H/W and Cleared by S/W.  
You should clear ADCF bit in ADC interrupt routine.

[An Example of ADC Conversion Table]

System Clock ( $F_{OSC}$ )	Divide (ADIV=0)	$F_{ADC}$	$T_{ADC}$ ( $1/F_{ADC}$ )	1 Sample Conversion Time
20MHz @ 5V	$F_{OSC}/2$	10MHz	100ns	9.6us
10MHz @ 5V	$F_{OSC}/2$	5MHz	200ns	19.2us
10MHz @ 3V	$F_{OSC}/2$	5MHz	200ns	19.2us
5MHz @ 3V	$F_{OSC}/2$	2.5MHz	400ns	38.4us



# 6.13. Interrupt : 9 Sources / 2-level Priority

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

[Interrupt Vector Address]

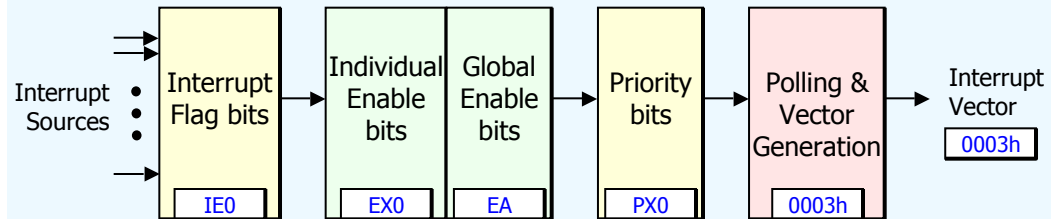
Interrupt Sources	Address	Priority Level
INT0	0003h	2 Levels
TF0	000Bh	2 Levels
INT1	0013h	2 Levels
TF1	001Bh	2 Levels
RI+TI	0023h	2 Levels
ADC	003Bh	2 Levels
INT2	0043h	2 Levels
	004Bh	2 Levels
WDT	0063h	2 Levels
PWM	006Bh	2 Levels

↑ HIGH  
 PRIORITY  
↑ LOW

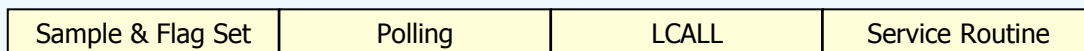
**\* Interrupt related to SFR (refer to Appendix B : SFR Description)**

✓ <b>TCON</b> (88h)	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
✓ <b>EXIF</b> (91h)	-	-	-	IE2	XT/RG	RGMD	RGSL	BGS
✓ <b>IE</b> (A8h)	EA	EADC	-	ES	ET1	EX1	ET0	EX0
✓ <b>IP</b> (B8h)	-	PADC	-	PS	PT1	PX1	PT0	PX0
✓ <b>EIE</b> (E8h)	-	-	EPWM	EWDT	-	-	-	EX2
✓ <b>EIP</b> (F8h)	-	-	PPWM	PWDT	-	-	-	PX2
✓ <b>WDCON</b> (D8h)	WD1	WD0	-	-	WDIF	WTRF	EWT	RWT
✓ <b>PWMCON</b> (DCh)	-	PS2_P0	PS1_P0	PS0_P0	-	PWMF	CLR_P0	RUN_P0

[Interrupt Vector Generation Flow]

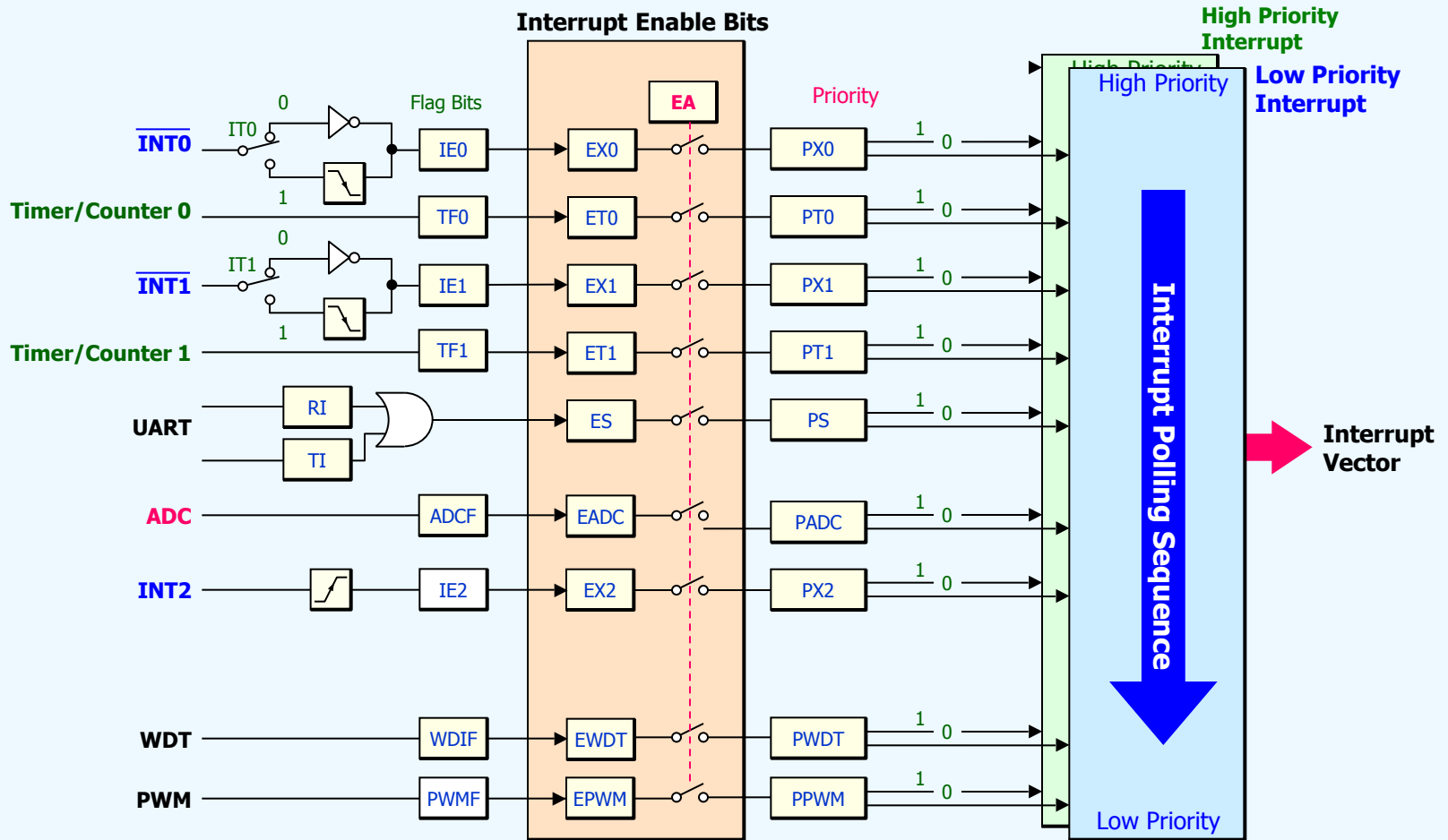


[Response Sequence]



↑ Last Cycle & High Priority & Not-update Interrupt Register

# 6.13. Interrupt Functional Description



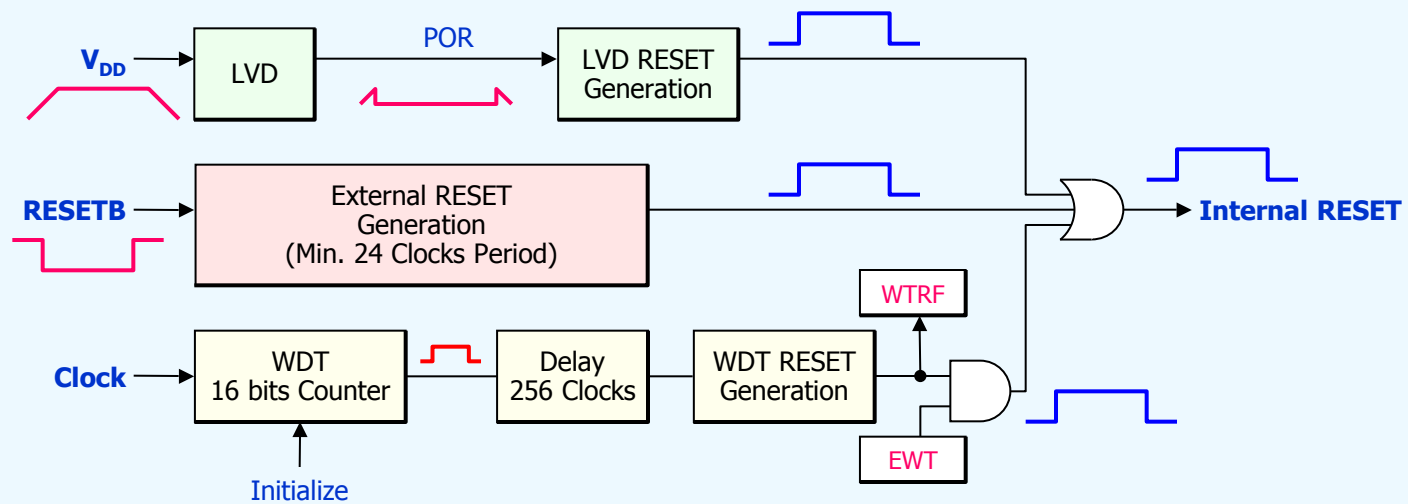
## 6.14. Reset Circuit : Three Reset Sources

- ◆ LVD(POR) Reset
  - ✓ Power-on Reset when Power-Up.
  - ✓ Power-fail Reset under  $V_{RST}$
- ◆ External RESET Pin
  - ✓ RESETB Pin must hold "L" for min. 24 clocks period.
  - ✓ Ring OSC. must be running.
  - ✓ **Caution, Reset Signal not must be Glitch noise.**
- ◆ WDT Reset : Optional Control by S/W

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

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



# 6.15. Clock Circuit

- ◆ 2 System Clock Sources : Ring OSC. or External Crystal
- ◆ Default System Clock is Ring OSC.
- ◆ Fast Wake-up from Power-down Mode using Ring OSC.

Control Flag				System Clock	Status Bit	
XT/RG	XTOFF	RINGON	RGSL		RGMD	XTUP
1	0	X	X	Crystal OSC.	0	1
0	X	1	X	Ring OSC.	1	0/1
1	0	X	0	Crystal OSC. (during Power-down Wake-up)	0	0
0	X	1	1	Ring OSC. (during Power-down Wake-up)	1	0

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

-	-	-	IE2	XT/RG	RGMD	RGSL	BGS
R/W(0)	R/W(0)	R/W(0)	R(1)	R/W(0)	R/W(1)		

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

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

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

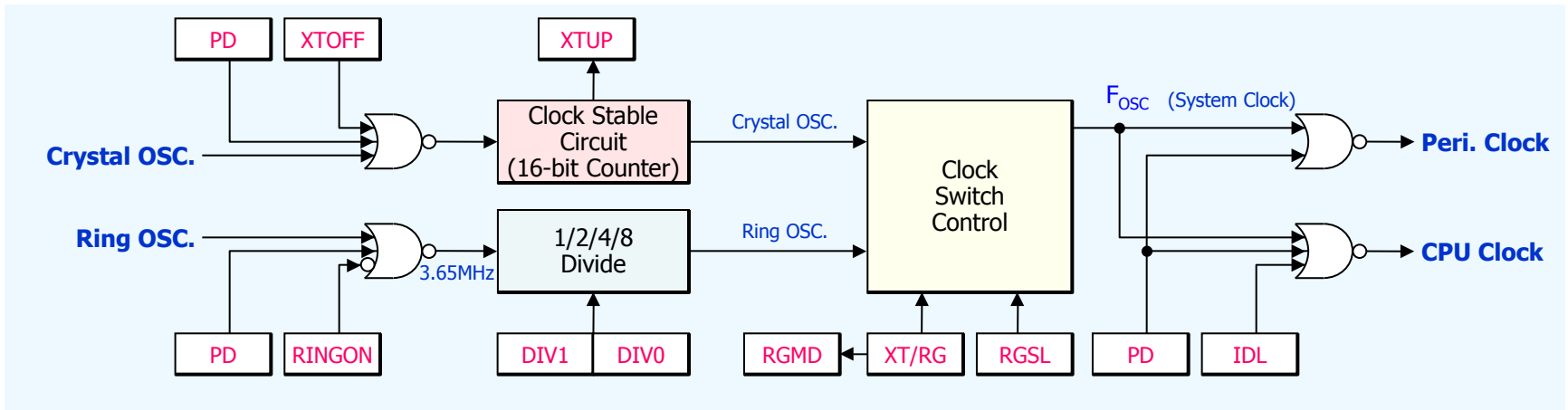
-	-	-	XTUP	-	-	-	-
			R(0)				

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

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

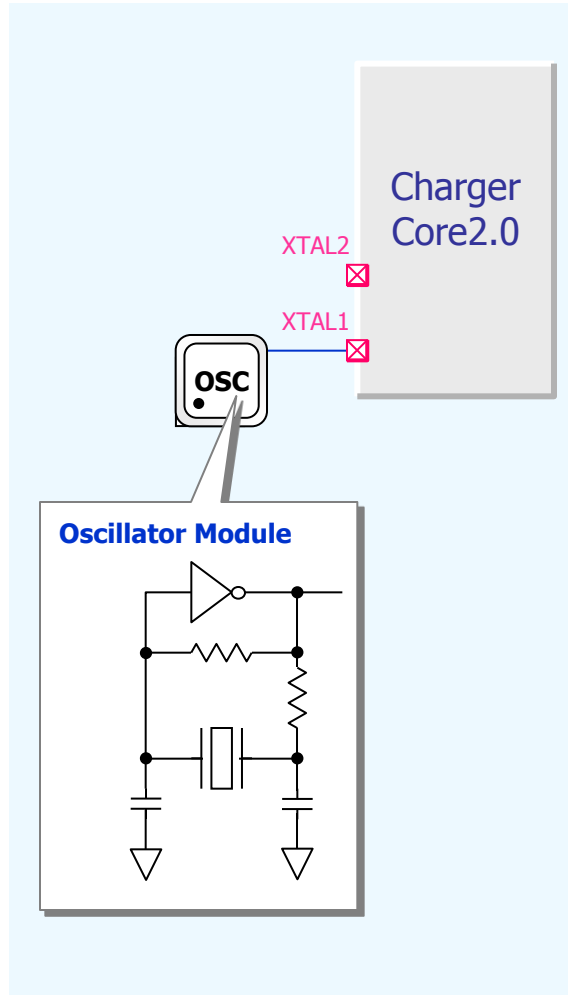
✓ **PCON** (87h) : Power Control Register

SMOD1	-	-	POF	GF1	GF0	PD	IDL
R/W(0)			R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

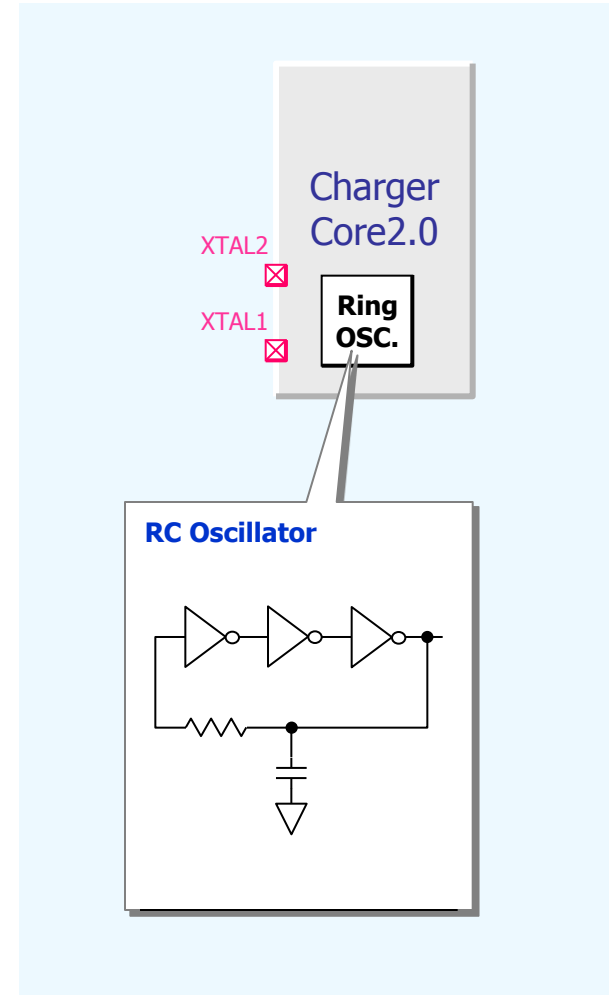


# 6.15. Clock Circuit : Guideline for Configuration

## ◆ Oscillator Module

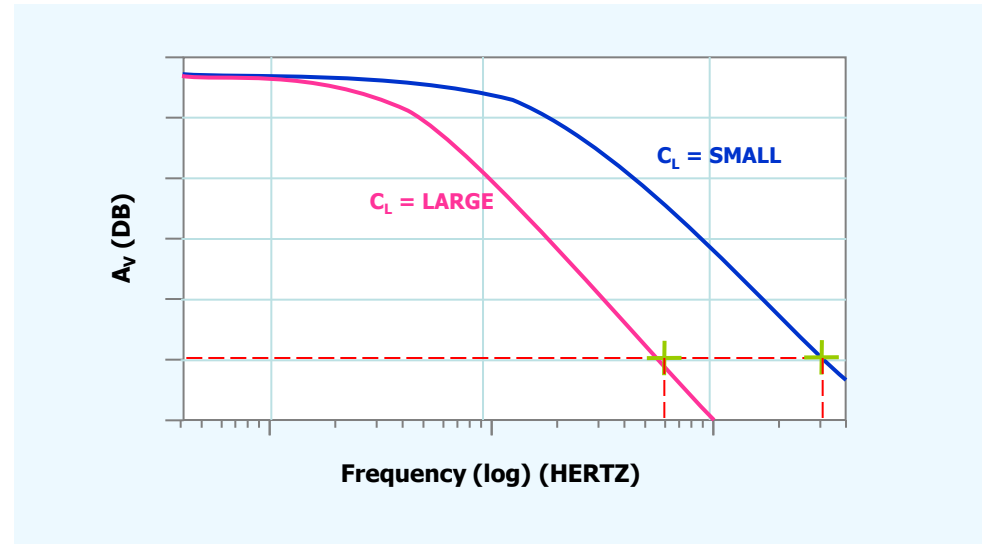


## ◆ Internal Ring Oscillator



## 6.15. Clock Circuit : Guideline for Crystal Usage

### ◆ Graph for Load Capacitor & Frequency



### ◆ Recommended $C_L$ (Load Capacitor)

$V_{DD} = 5\text{ V}$

	Crystal Oscillator [MHz]	
		~ 11.0592
Load Cap. $C_L$	47pF	20pF

## 6.16. Power Management : 3 Modes

◆ **Active Mode** : CPU and Peripheral are running.

◆ **Idle Mode** : Only Peripheral is running.

- ✓ Wake-up from any kinds of interrupts. CPU continues.  
But, WDT and Level Interrupt are not allowed.
- ✓ 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 external interrupt INT0/INT1 (level detect).  
→ MCU continues.
- ✓ Wake-up from all kinds of resets. (ex: RESETB, LVD. etc)  
→ MCU restarts.

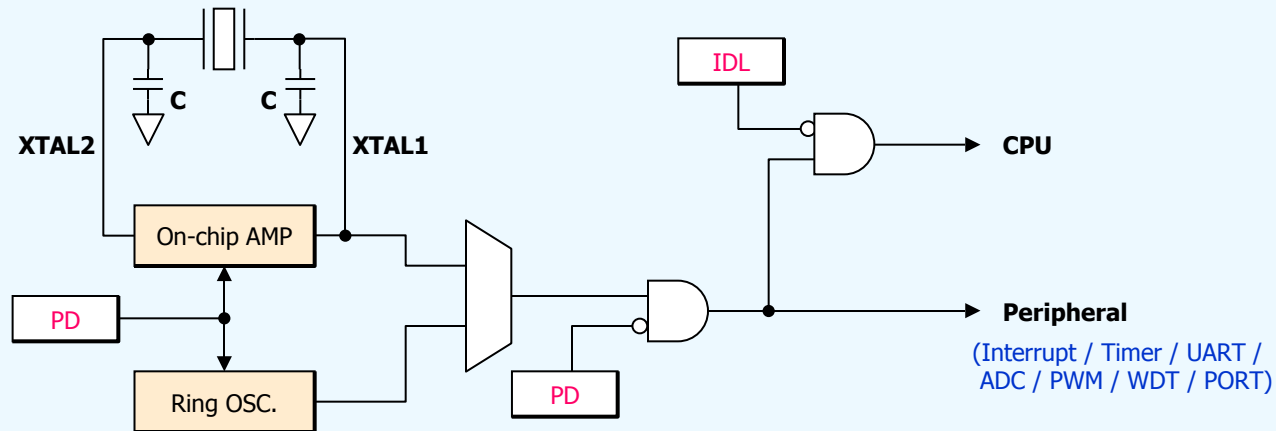
◆ **Stop Mode 2** : When WDT is on

- ✓ Wake-up from external interrupt INT0/INT1 (level detect), or WDT interrupt. → MCU continues.
- ✓ Wake-up from all kinds of resets. (ex: RESETB, LVD, .etc)  
→ MCU restarts.

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

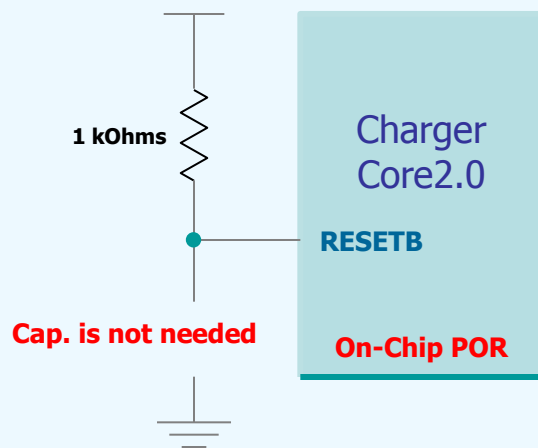
SMOD1	-	-	POF	GF1	GF0	PD	IDL
R/W(0)			R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

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



## 7. Strong Point : On-Chip POR

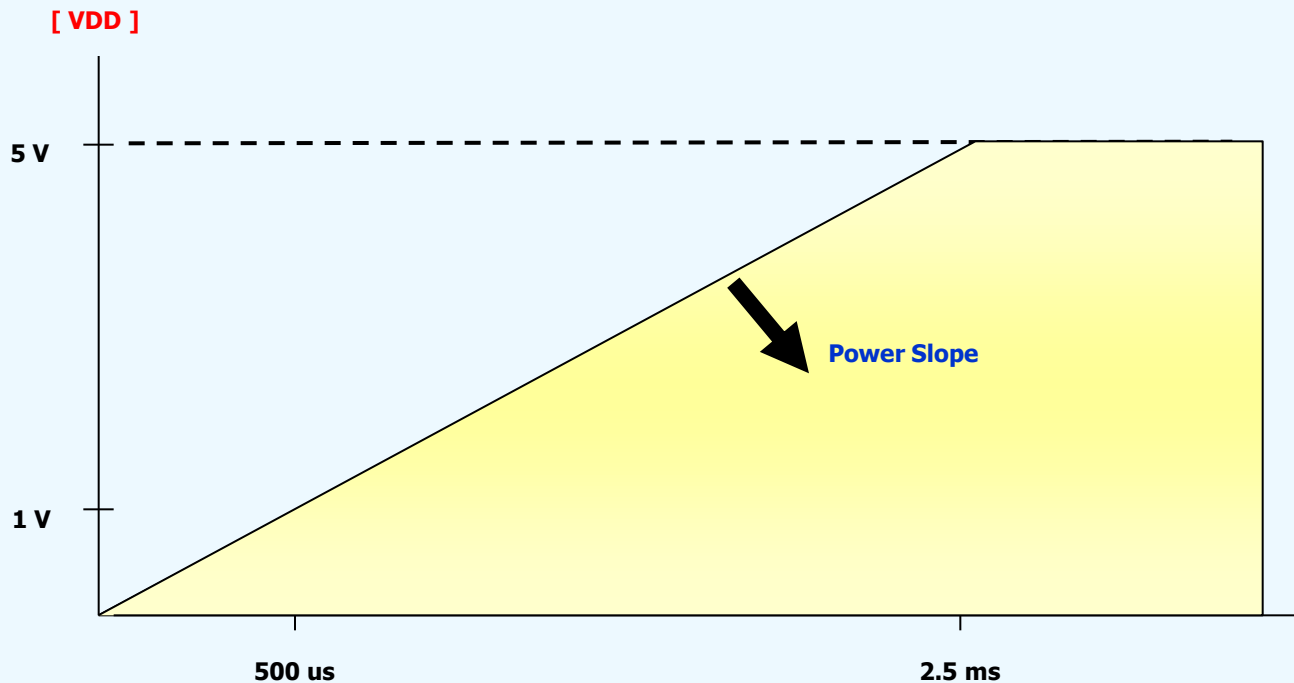
- ◆ On-Chip POR (Power On Reset) can reduce system cost by removing needless capacitor.





## 8. Recommended Power Slope

- ◆ The supply voltage slope must be in the range from 0.0V/us to 1.0V/500us. (5V/2.5ms)  
(That is, the supply voltage should be increasing monotonically until it reaches to the normal range.)



## 9. Absolute Maximum Ratings

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
Operating Temperature	-	-40 °C to 120 °C
Storage Temperature	-	-65 °C to +150 °C
Soldering Temperature	-	260 °C for 10 seconds

# 10. DC Characteristics

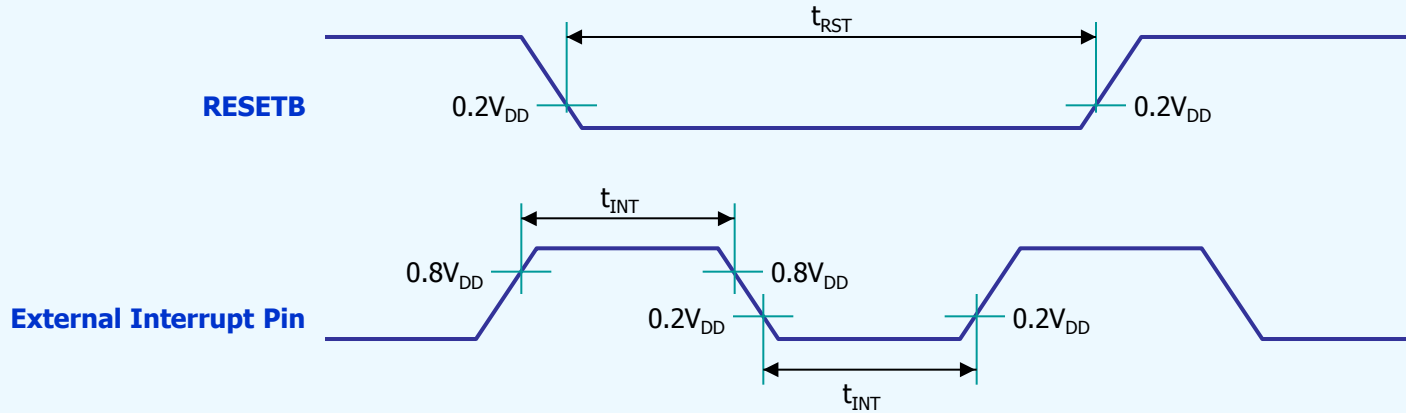
\*  $T_A = -20\text{ }^{\circ}\text{C} \sim +85\text{ }^{\circ}\text{C}$ ,  $V_{DD} = 2.4\text{V} \sim 5.5\text{V}$  unless otherwise specified.

Parameter	Symbol	Pin	Conditions	Value			Unit
				Min.	Typ.	Max.	
Input Low Voltage	$V_{IL1}$	P0, P2	$V_{DD} = 2.4\text{V} \sim 5.5\text{V}$	-0.5	-	$0.2V_{DD} - 0.1$	V
	$V_{IL2}$	XTAL1, XTAL2, RESETB		-0.5	-	$0.3V_{DD}$	
Input high Voltage	$V_{IH1}$	P0, P2	$V_{DD} = 2.4\text{V} \sim 5.5\text{V}$	$0.2V_{DD} + 1.0$	-	$V_{DD} + 0.5$	V
	$V_{IH2}$	XTAL1, XTAL2, RESETB		$0.7V_{DD}$	-	$V_{DD} + 0.5$	
Output Low Voltage	$V_{OL1}$	XTAL1, XTAL2, P0, P2	$I_{OL} = 20\text{mA} @V_{DD}=5\text{V}$ ( $I_{OL} = 5\text{mA} @V_{DD}=2.6\text{V}$ )	-	-	$0.3V_{DD}$	V
	$V_{OL2}$	RESETB	$I_{OL} = 10\text{mA} @V_{DD}=5\text{V}$ ( $I_{OL} = 2.5\text{mA} @V_{DD}=2.6\text{V}$ )	-	-	$0.3V_{DD}$	
Output High Voltage	$V_{OH}$	XTAL1, XTAL2, P0, P2	$I_{OH} = -15\text{mA} @V_{DD}=5\text{V}$ ( $I_{OH} = -2.5\text{mA} @V_{DD}=2.6\text{V}$ )	$0.7V_{DD}$	-	-	V
	$V_{OHP1}$	P0, P2 (pull-up resistor only)	$I_{OH} = -140\mu\text{A} @V_{DD}=5\text{V}$ ( $I_{OH} = -20\mu\text{A} @V_{DD}=2.6\text{V}$ )	$0.7V_{DD}$	-	-	V
	$V_{OHP2}$	XTAL1, XTAL2 (pull-up resistor only)	$I_{OH} = -10\mu\text{A} @V_{DD}=5\text{V}$ ( $I_{OH} = -1.5\mu\text{A} @V_{DD}=2.6\text{V}$ )	$0.7V_{DD}$	-	-	V
Input Leakage Current	$I_{IL}$	All pins except XTAL1, XTAL2	$V_{IN} = V_{IH}$ or $V_{IL}$	-	-	$\pm 1$	$\mu\text{A}$
Pin Capacitance	$C_{I0}$	All	$V_{DD} = 5\text{V}$	-	10	-	pF

# 11. AC Characteristics

\* TA = -20 °C ~ +85 °C unless otherwise specified.

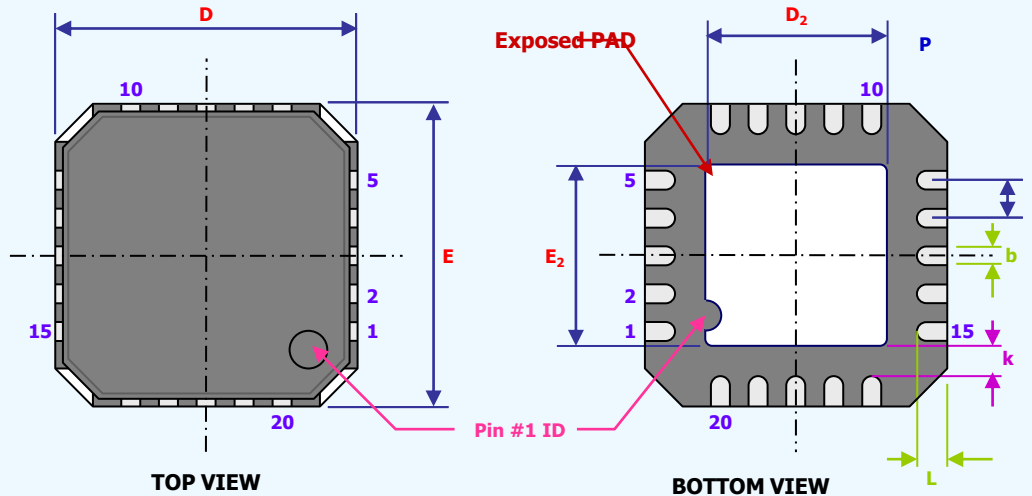
Parameter	Symbol	Pin	Conditions	Value			Unit
				Min.	Typ.	Max.	
Operating Frequency	F <sub>OSC</sub>	XTAL1, XTAL2	V <sub>DD</sub> = 5V ± 10%	1	-	20	MHz
			V <sub>DD</sub> = 3V ± 10%	1	-	10	
RESETB Input Width	t <sub>RST</sub>	RESETB	V <sub>DD</sub> = 5V ± 10%	24	-	-	F <sub>OSC</sub>
			V <sub>DD</sub> = 3V ± 10%	24	-	-	
External Interrupt Input Width	t <sub>INT</sub>	External Interrupt	V <sub>DD</sub> = 5V ± 10%	4	-	-	F <sub>OSC</sub>
			V <sub>DD</sub> = 3V ± 10%	4	-	-	



## 12. ADC Specifications

Parameter	Symbol	Conditions	Value			Unit	
			Min.	Typ.	Max.		
Supply Voltage	$V_{DDADC}$	-	2.4	-	5.5	V	
Input Voltage	$V_{INADC}$	-	$V_{SS}$	-	$V_{DD}$	V	
Resolution	$RES_{ADC}$	-	-	10	-	bit	
Operating Frequency	$F_{ADC}$	$V_{DD} = 4.5V \sim 5.5V$ $V_{DD} = 2.4V \sim 3.3V$	-	-	10 5	MHz	
Conversion Time	$t_{ADC}$	-	-	$96 / F_{ADC}$	-	s	
Overall Accuracy	$OA_{ADC}$	$V_{DD} = 5V, F_{ADC} = 10MHz$ $V_{DD} = 3V, F_{ADC} = 5MHz$	-	$\pm 2$	$\pm 4$	LSB	
Integral Nonlinearity	$INL_{ADC}$	$V_{DD} = 5V, F_{ADC} = 10MHz$ $V_{DD} = 3V, F_{ADC} = 5MHz$	-	$\pm 2$	$\pm 4$	LSB	
Differential Nonlinearity	$DNL_{ADC}$	$V_{DD} = 5V, F_{ADC} = 10MHz$ $V_{DD} = 3V, F_{ADC} = 5MHz$	-	$\pm 0.5$	$\pm 1$	LSB	
Zero Input Error	$ZIE_{ADC}$	$V_{DD} = 5V, F_{ADC} = 10MHz$ $V_{DD} = 3V, F_{ADC} = 5MHz$	-	$\pm 2$	$\pm 4$	LSB	
Full Scale Error	$FSE_{ADC}$	$V_{DD} = 5V, F_{ADC} = 10MHz$ $V_{DD} = 3V, F_{ADC} = 5MHz$	-	$\pm 2$	$\pm 4$	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	

# 13.1. Package Dimensions : 20-MLF

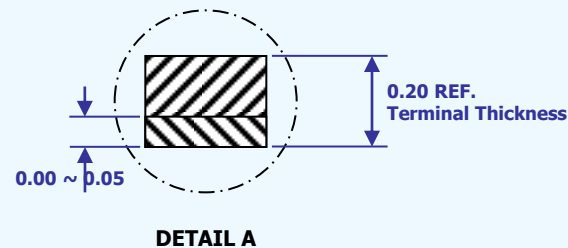
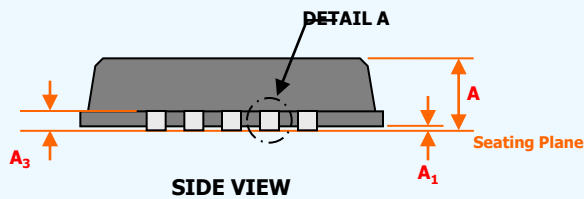


[20-MLF]

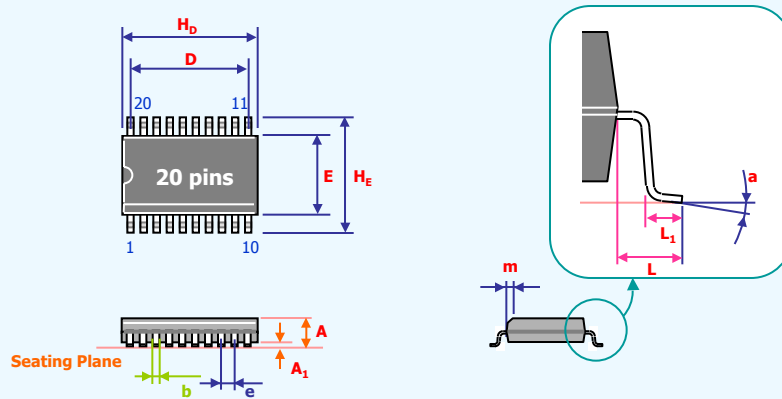
Symbol	Dimensions [mm]		
	Min.	Nom.	Max.
A	0.80	0.85	0.90
A <sub>1</sub>	0.00	0.01	0.05
A <sub>3</sub>	0.20 REF		
D	5.00 BSC		
E	5.00 BSC		
D <sub>2</sub>	3.00	3.10	3.20
E <sub>2</sub>	3.00	3.10	3.20
b	0.23	0.28	0.35
e	0.65 BSC		
L	0.50	0.60	0.75
k	0.20	-	-
P	0.24	0.42	0.60

**Notes:**

1. All Dimension are in mm. Angles in Degrees.
  2. Dimension b applies to Plated Terminal & is measured.
  3. BSC : Basic Dimension. Theoretically exact value shown without tolerances.
- REF : Reference Dimension, Usually without tolerance, for information purpose only.



## 13.2. Package Dimensions : 20-TSSOP



[20-TSSOP]

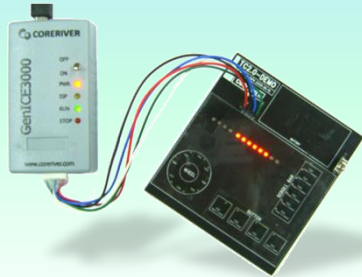
Symbol	Dimension in Inches			Dimension in mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
A			0.043			1.1
A <sub>1</sub>	0.001		0.006	0.020		0.150
b	0.007		0.012	0.190		0.300
D		0.234			5.850	
E	0.169	0.174	0.177	4.300	4.400	4.500
H <sub>b</sub>	0.252	0.254	0.259	6.400	6.500	6.600
H <sub>e</sub>	0.246	0.252	0.258	6.250	6.400	6.550
L	0.038	0.039	0.040	0.975	1.000	1.025
L <sub>1</sub>	0.020	0.024	0.028	0.500	0.600	0.700
a	1°	-	7°	1°	-	7°
e		0.026 BSC			0.65 BSC	

**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.

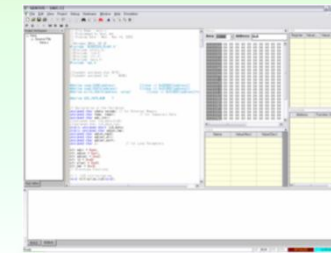
# 14. Supporting tools

## In-Circuit Debugger (GENSYS & GenICE)



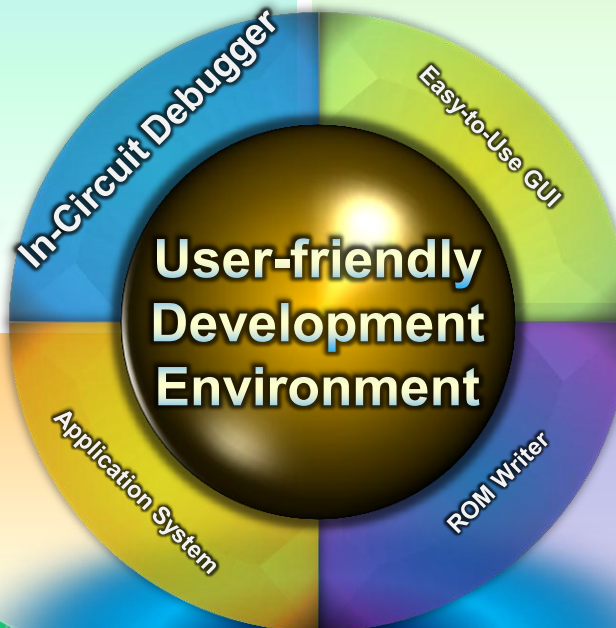
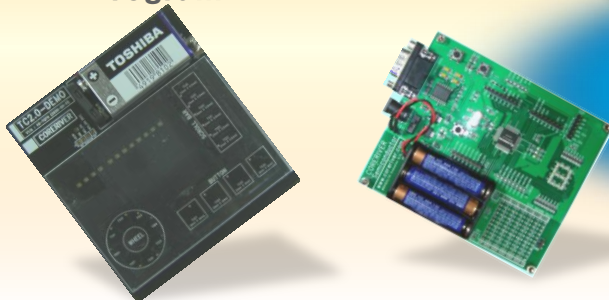
## Easy-to-Use GUI (GENTOS)

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



## Application System

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



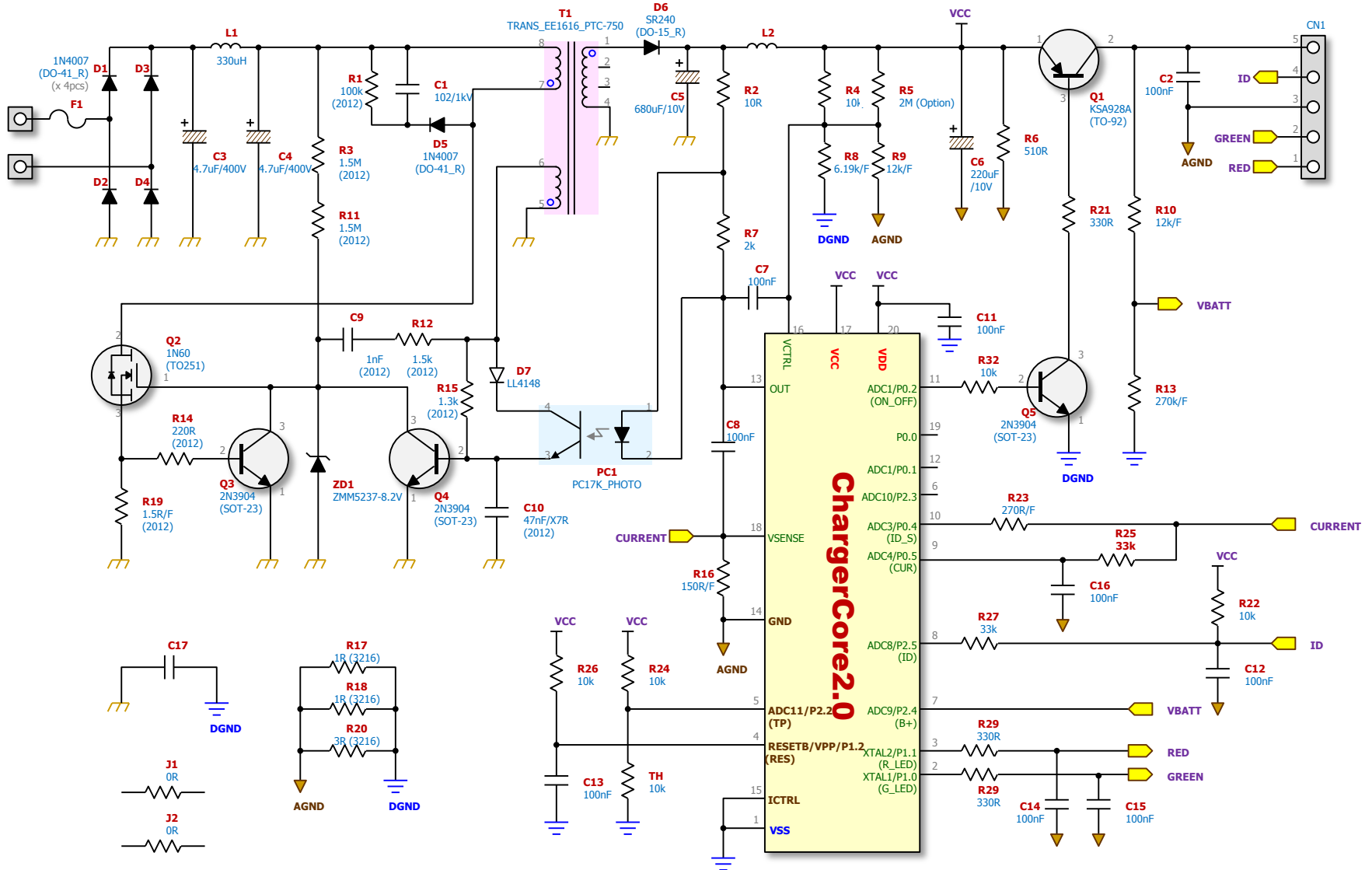
## ROM Writer

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





# 15. Typical Application Circuit



# Appendix A : Instruction Set (1/19)

◆ Note on Instruction Set and Addressing Modes

Notation	Descriptions
<b>Rn</b>	Register R0 ~ R7 of the currently selected Register Bank (RB0 ~ RB3).
<b>direct</b>	The address of 8-bit internal data location. This could be an IRAM location (0x00 ~ 0x7F; 128 bytes) or a SFR (0x80 ~ 0xFF).
<b>@Ri</b>	8-bit IRAM location (0x00 ~ 0xFF; 256 bytes) addressed indirectly through register <b>R0</b> or <b>R1</b> .
<b>#data</b>	8-bit constant included in instruction.
<b>#data16</b>	16-bit constant included in instruction.
<b>addr16</b>	16-bit destination address. Used by <b>LCALL</b> & <b>LJMP</b> . The branch can be anywhere within the 64kbytes program memory address space. (MiDAS1.1 Family : 4kbytes program memory)
<b>addr11</b>	11-bit destination address. Used by <b>ACALL</b> & <b>AJMP</b> . The branch will be within the same 2kbytes page of program memory as the first byte of the following instruction.
<b>rel</b>	Signed (2's complement number) 8-bit offset byte. Used by <b>SJMP</b> and all conditional jumps. Range is -128 to +127 byte relative to first byte of the following instruction.
<b>bit</b>	Direct addressed bit n IRAM of SFR.

# Appendix A : Instruction Set (2/19)

**ADD A, <src-byte>**

## Add

**ADD A, Rn**

**Operation :** (A) ← (A) + (Rn)

**ADD A, @Ri**

**Operation :** (A) ← (A) + ((Ri))

**ADD A, direct**

**Operation :** (A) ← (A) + (direct)

**ADD A, #date**

**Operation :** (A) ← (A) + data

**ADDC A, <src-byte>**

## Add with Carry

**ADDC A, Rn**

**Operation :** (A) ← (A) + (C) + (Rn)

**ADDC A, @Ri**

**Operation :** (A) ← (A) + (C) + ((Ri))

**ADDC A, direct**

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

**ADDC A, #date**

**Operation :** (A) ← (A) + (C) + data

**1 Machine Cycle = 4 Clocks**

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

0	0	1	0	1	r	r	r
---	---	---	---	---	---	---	---

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

0	0	1	0	0	1	1	i
---	---	---	---	---	---	---	---

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

0	0	1	0	0	1	0	1
---	---	---	---	---	---	---	---

direct addr

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

0	0	1	0	0	1	0	0
---	---	---	---	---	---	---	---

immediate data

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

0	0	1	1	1	r	r	r
---	---	---	---	---	---	---	---

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

0	0	1	1	0	1	1	i
---	---	---	---	---	---	---	---

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

0	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---

direct addr

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

0	0	1	1	0	1	0	0
---	---	---	---	---	---	---	---

immediate data

# Appendix A : Instruction Set (3/19)

**SUBB A, <src-byte>**

## Subtract with Borrow

SUBB A, Rn

**Operation :** (A)  $\leftarrow$  (A) - (C) - (Rn)

SUBB A, @Ri

**Operation :** (A)  $\leftarrow$  (A) - (C) - ((Ri))

SUBB A, direct

**Operation :** (A)  $\leftarrow$  (A) - (C) - (direct)

SUBB A, #data

**Operation :** (A)  $\leftarrow$  (A) - (C) - data

**INC <byte>**

## Increment

INC A

**Operation :** (A)  $\leftarrow$  (A) + 1

INC Rn

**Operation :** (Rn)  $\leftarrow$  (Rn) + 1

INC @Ri

**Operation :** ((Ri))  $\leftarrow$  ((Ri)) + 1

INC direct

**Operation :** (direct)  $\leftarrow$  (direct) + 1

INC DPTR

**Operation :** (DPTR)  $\leftarrow$  (DPTR) + 1

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

1	0	0	1	1	r	r	r
---	---	---	---	---	---	---	---

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

1	0	0	1	0	1	1	i
---	---	---	---	---	---	---	---

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

1	0	0	1	0	1	0	1
---	---	---	---	---	---	---	---

direct addr

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

1	0	0	1	0	1	0	0
---	---	---	---	---	---	---	---

immediate data

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

0	0	0	0	0	1	0	0
---	---	---	---	---	---	---	---

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

0	0	0	0	1	r	r	r
---	---	---	---	---	---	---	---

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

0	0	0	0	0	1	1	i
---	---	---	---	---	---	---	---

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

0	0	0	0	0	1	0	1
---	---	---	---	---	---	---	---

direct addr

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

1	0	1	0	0	0	1	1
---	---	---	---	---	---	---	---

# Appendix A : Instruction Set (4/19)

DEC <byte>

## Decrement

DEC A

Operation : (A) ← (A) - 1

DEC Rn

Operation : (Rn) ← (Rn) - 1

DEC @Ri

Operation : ((Ri)) ← ((Ri)) - 1

DEC direct

Operation : (direct) ← (direct) - 1

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

0	0	0	1	0	1	0	0
---	---	---	---	---	---	---	---

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

0	0	0	1	1	r	r	r
---	---	---	---	---	---	---	---

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

0	0	0	1	0	1	1	i
---	---	---	---	---	---	---	---

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

0	0	0	1	0	1	0	1
---	---	---	---	---	---	---	---

direct addr

MUL AB

## Multiply

Operation : (A)<sub>7-0</sub> ← (A) × (B)  
(B)<sub>15-8</sub>

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

1	0	1	0	0	1	0	0
---	---	---	---	---	---	---	---

DIV AB

## Divide

Operation : (A)<sub>15-8</sub> ← (A) / (B)  
(B)<sub>7-0</sub>

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)

ANL A, @Ri

**Operation :** (A) ← (A) ^ ((Ri))

ANL A, direct

**Operation :** (A) ← (A) ^ (direct)

ANL A, #data

**Operation :** (A) ← (A) ^ data

ANL direct, A

**Operation :** (direct) ← (direct) ^ (A)

ANL direct, #data

**Operation :** (direct) ← (direct) ^ data

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

0	1	0	1	1	r	r	r
---	---	---	---	---	---	---	---

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

0	1	0	1	0	1	1	i
---	---	---	---	---	---	---	---

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

0	1	0	1	0	1	0	1	direct addr
---	---	---	---	---	---	---	---	-------------

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

0	1	0	1	0	1	0	0	immediate data
---	---	---	---	---	---	---	---	----------------

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

0	1	0	1	0	0	1	0	direct addr
---	---	---	---	---	---	---	---	-------------

**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)  $\leftarrow$  (C)  $\wedge$  (bit)

ANL C, /bit

**Operation :** (C)  $\leftarrow$  (C)  $\wedge$   $\sim$ (bit)

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

1 0 0 0 0 0 1 0

bit addr

**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)  $\leftarrow$  (A)  $\vee$  (Rn)

ORL A, @Ri

**Operation :** (A)  $\leftarrow$  (A)  $\vee$  ((Ri))

ORL A, direct

**Operation :** (A)  $\leftarrow$  (A)  $\vee$  (direct)

ORL A, #data

**Operation :** (A)  $\leftarrow$  (A)  $\vee$  data

ORL direct, A

**Operation :** (direct)  $\leftarrow$  (direct)  $\vee$  (A)

ORL direct, #data

**Operation :** (direct)  $\leftarrow$  (direct)  $\vee$  data

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

0 1 0 0 1 r r r

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

0 1 0 0 0 1 1 i

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

0 1 0 0 0 1 0 1

direct addr

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

0 1 0 0 0 1 0 0

immediate data

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

0 1 0 0 0 0 1 0

direct addr

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

0 1 0 0 0 0 1 1

direct addr

immediate data

# Appendix A : Instruction Set (7/19)

**ORL C, <src-byte>**

## Logical OR for byte variables

ORL C, bit

**Operation :** (C)  $\leftarrow$  (C)  $\vee$  (bit)

ORL C, /bit

**Operation :** (C)  $\leftarrow$  (C)  $\vee$   $\sim$ (bit)

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

## Logical Exclusive-OR for byte variables

XRL A, Rn

**Operation :** (A)  $\leftarrow$  (A)  $\oplus$  (Rn)

XRL A, @Ri

**Operation :** (A)  $\leftarrow$  (A)  $\oplus$  ((Ri))

XRL A, direct

**Operation :** (A)  $\leftarrow$  (A)  $\oplus$  (direct)

XRL A, #data

**Operation :** (A)  $\leftarrow$  (A)  $\oplus$  data

XRL direct, A

**Operation :** (direct)  $\leftarrow$  (direct)  $\oplus$  (A)

XRL direct, #data

**Operation :** (direct)  $\leftarrow$  (direct)  $\oplus$  data

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

0 1 1 1 0 0 1 0

bit addr

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

1 0 1 0 0 0 0 0

bit addr

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

0 1 1 0 1 r r r

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

0 1 1 0 0 1 1 i

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

0 1 1 0 0 1 0 1

direct addr

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

0 1 1 0 0 1 0 0

immediate data

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

0 1 1 0 0 0 1 0

direct addr

**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)     $\leftarrow$  0

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

1	1	1	0	0	1	0	0
---	---	---	---	---	---	---	---

**CLR    <bit>**

**Clear bit**

CLR    C

**Operation :**    (C)     $\leftarrow$  0

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

1	1	0	0	0	0	1	1
---	---	---	---	---	---	---	---

CLR    bit

**Operation :**    (bit)     $\leftarrow$  0

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

1	1	0	0	0	0	1	0
---	---	---	---	---	---	---	---

bit addr
----------

**CPL    A**

**Complement Accumulator**

**Operation :**    (A)     $\leftarrow$   $\sim$ (A)

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

1	1	1	1	0	1	0	0
---	---	---	---	---	---	---	---

**CPL    <bit>**

**Complement bit**

CPL    C

**Operation :**    (C)     $\leftarrow$   $\sim$ (C)

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

1	0	1	1	0	0	1	1
---	---	---	---	---	---	---	---

CPL    bit

**Operation :**    (bit)     $\leftarrow$   $\sim$ (bit)

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

1	0	1	1	0	0	1	0
---	---	---	---	---	---	---	---

bit addr
----------

# Appendix A : Instruction Set (9/19)

**RL      A**

## Rotate Accumulator Left

**Operation :**       $(A_{n+1}) \leftarrow (A_n)$        $n=0\sim6$   
                           $(A_0) \leftarrow (A_7)$

**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 :**       $(A_{n+1}) \leftarrow (A_n)$        $n=0\sim6$   
                           $(A_0) \leftarrow (C)$   
                           $(C) \leftarrow (A_7)$

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

0	0	1	1	0	0	1	1
---	---	---	---	---	---	---	---

**RR      A**

## Rotate Accumulator Right

**Operation :**       $(A_n) \leftarrow (A_{n+1})$        $n=0\sim6$   
                           $(A_7) \leftarrow (A_0)$

**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 :**       $(A_n) \leftarrow (A_{n+1})$        $n=0\sim6$   
                           $(A_7) \leftarrow (C)$   
                           $(C) \leftarrow (A_0)$

**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
---	---	---	---	---	---	---	---

# Appendix A : Instruction Set (10/19)

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

## Move byte variable

MOV	A, Rn
<b>Operation :</b>	(A) ← (Rn)
MOV	A, @Ri
<b>Operation :</b>	(A) ← ((Ri))
MOV	A, direct
<b>Operation :</b>	(A) ← (direct)
MOV	A, #date
<b>Operation :</b>	(A) ← data
MOV	Rn, A
<b>Operation :</b>	(Rn) ← (A)
MOV	Rn, direct
<b>Operation :</b>	(Rn) ← (direct)
MOV	Rn, #date
<b>Operation :</b>	(Rn) ← data
MOV	direct, A
<b>Operation :</b>	(direct) ← (A)
MOV	direct, Rn
<b>Operation :</b>	(direct) ← (Rn)

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

1	1	1	0	1	r	r	r
---	---	---	---	---	---	---	---

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

1	1	1	0	0	1	1	i
---	---	---	---	---	---	---	---

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

1	1	1	0	0	1	0	1	direct addr
---	---	---	---	---	---	---	---	-------------

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

0	1	1	1	0	1	0	0	immediate data
---	---	---	---	---	---	---	---	----------------

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

1	1	1	1	1	r	r	r
---	---	---	---	---	---	---	---

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

1	0	1	0	1	r	r	r	direct addr
---	---	---	---	---	---	---	---	-------------

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

0	1	1	1	1	r	r	r	immediate data
---	---	---	---	---	---	---	---	----------------

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

1	1	1	1	0	1	0	1	direct addr
---	---	---	---	---	---	---	---	-------------

**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
<b>Operation :</b>	(direct) ← ((Ri))
MOV	direct, direct
<b>Operation :</b>	(direct) ← (direct)
MOV	direct, #data
<b>Operation :</b>	(direct) ← data
MOV	@Ri, A
<b>Operation :</b>	((Ri)) ← (A)
MOV	@Ri, direct
<b>Operation :</b>	((Ri)) ← (direct)
MOV	@Ri, #data
<b>Operation :</b>	((Ri)) ← data

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

## Move bit data

MOV	C, bit
<b>Operation :</b>	(C) ← (bit)
MOV	bit, C
<b>Operation :</b>	(bit) ← (C)

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

1	0	0	0	0	1	1	i	direct addr
---	---	---	---	---	---	---	---	-------------

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

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

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

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

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

1	1	1	1	0	1	1	i
---	---	---	---	---	---	---	---

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

1	0	1	0	0	1	1	i	direct addr
---	---	---	---	---	---	---	---	-------------

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

0	1	1	1	0	1	1	i	immediate Data
---	---	---	---	---	---	---	---	----------------

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

1	0	1	0	0	0	1	0	bit addr
---	---	---	---	---	---	---	---	----------

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

1	0	0	1	0	0	1	0	bit addr
---	---	---	---	---	---	---	---	----------

# Appendix A : Instruction Set (12/19)

**MOV DPTR, #data16**

**Load Data Pointer with a 16-bit constant**

**Operation :** (DPTR) ← data<sub>15-0</sub>  
 (DPH, DPL) ← (data<sub>15-8</sub>, data<sub>7-0</sub>)

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



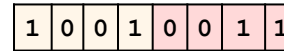
**MOVC A, @A + <base-reg>**

**Move Code byte**

**MOVC A, @A + DPTR**

**Operation :** (A) ← ((A) + (DPTR))

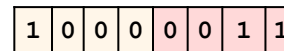
**Encoding :** HEX: 93h, #bytes: 1, Cycles: 2



**MOVC A, @A + PC**

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

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



# Appendix A : Instruction Set (13/19)

**XCH**    **A, <src-byte>**

**Exchange Accumulator with byte variable**

**XCH**    **A, Rn**

**Operation :**    (A) ↔ (Rn)

**XCH**    **A, @Ri**

**Operation :**    (A) ↔ ((Ri))

**XCH**    **A, direct**

**Operation :**    (A) ↔ (direct)

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

1	1	0	0	1	r	r	r
---	---	---	---	---	---	---	---

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

1	1	0	0	0	1	1	i
---	---	---	---	---	---	---	---

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

1	1	0	0	0	1	0	1	direct addr
---	---	---	---	---	---	---	---	-------------

**XCHD**    **A, @Ri**

**Exchange Digit**

**Operation :**    (A<sub>3-0</sub>) ↔ ((Ri))<sub>3-0</sub>

**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)  $\leftarrow$  1

**SETB** bit

**Operation :** (bit)  $\leftarrow$  1

**JC** rel

**Jump if Carry is set**

**Operation :** (PC)  $\leftarrow$  (PC) + 2  
If (C) = 1, then (PC)  $\leftarrow$  (PC) + rel

**JNC** rel

**Jump if Carry is not set**

**Operation :** (PC)  $\leftarrow$  (PC) + 2  
If (C) = 0, then (PC)  $\leftarrow$  (PC) + rel

**JB** bit, rel

**Jump if Bit is set**

**Operation :** (PC)  $\leftarrow$  (PC) + 3  
If (bit) = 1, then (PC)  $\leftarrow$  (PC)+rel

**JNB** bit, rel

**Jump if Bit is not set**

**Operation :** (PC)  $\leftarrow$  (PC) + 3  
If (bit) = 0, then (PC)  $\leftarrow$  (PC)+rel

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

1 1 0 1 0 0 1 1

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

1 1 0 1 0 0 1 0

bit addr

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

0 1 0 0 0 0 0 0

relative addr

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

0 1 0 1 0 0 0 0

relative addr

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

0 0 1 0 0 0 0 0

bit addr

relative addr

**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**

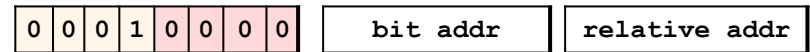
**Jump if Bit is set and Clear bit**

**Operation :**

$$(PC) \leftarrow (PC) + 3$$

If (bit) = 1,  
then (bit)  $\leftarrow$  0, (PC)  $\leftarrow$  (PC) + rel

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



**ACALL addr11**

**Absolute Subroutine Call**

**Operation :**

$$(PC) \leftarrow (PC) + 2$$

$$(SP) \leftarrow (SP) + 1$$

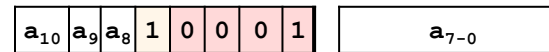
$$((SP)) \leftarrow (PC_{7-0})$$

$$(SP) \leftarrow (SP) + 1$$

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

$$(PC_{10-0}) \leftarrow \text{page address}$$

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



**LCALL addr16**

**Long Subroutine Call**

**Operation :**

$$(PC) \leftarrow (PC) + 3$$

$$(SP) \leftarrow (SP) + 1$$

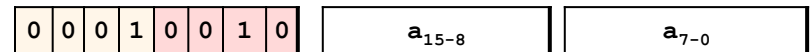
$$((SP)) \leftarrow (PC_{7-0})$$

$$(SP) \leftarrow (SP) + 1$$

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

$$(PC) \leftarrow \text{addr}_{15-0}$$

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





# Appendix A : Instruction Set (16/19)

## RET

### Return from Subroutine

**Operation :**

$$\begin{aligned} (PC_{15-8}) &\leftarrow ((SP)) \\ (SP) &\leftarrow (SP) - 1 \\ (PC_{7-0}) &\leftarrow ((SP)) \\ (SP) &\leftarrow (SP) - 1 \end{aligned}$$

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

0	0	1	0	0	0	1	0
---	---	---	---	---	---	---	---

## RETI

### Return from Interrupt

**Operation :**

$$\begin{aligned} (PC_{15-8}) &\leftarrow ((SP)) \\ (SP) &\leftarrow (SP) - 1 \\ (PC_{7-0}) &\leftarrow ((SP)) \\ (SP) &\leftarrow (SP) - 1 \end{aligned}$$

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

0	0	1	1	0	0	1	0
---	---	---	---	---	---	---	---

## AJMP addr11

### Absolute Jump

**Operation :**

$$\begin{aligned} (PC) &\leftarrow (PC) + 2 \\ (PC_{10-0}) &\leftarrow \text{page address} \end{aligned}$$

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

a <sub>10</sub>	a <sub>9</sub>	a <sub>8</sub>	0	0	0	0	1	a <sub>7-0</sub>
-----------------	----------------	----------------	---	---	---	---	---	------------------

## SJMP rel

### Short Jump (Relative address)

**Operation :**

$$\begin{aligned} (PC) &\leftarrow (PC) + 2 \\ (PC_{10-0}) &\leftarrow (PC) + \text{rel} \end{aligned}$$

**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<sub>15-0</sub>

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

0	0	0	0	0	0	1	0	a <sub>15-8</sub>	a <sub>7-0</sub>
---	---	---	---	---	---	---	---	-------------------	------------------

# Appendix A : Instruction Set (17/19)

**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**

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

**CJNE A, #data, rel**

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

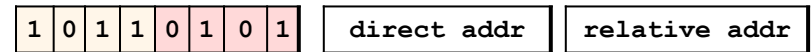
**CJNE Rn, #data, rel**

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

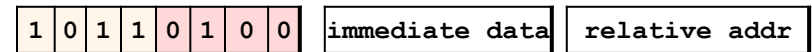
**CJNE @Ri, #data, rel**

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

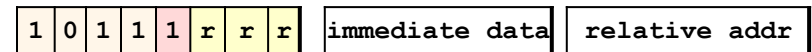
**Encoding :** HEX: B5h, #bytes: 3, Cycles: 4



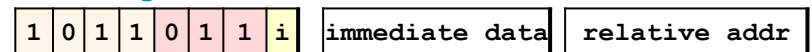
**Encoding :** HEX: B4h, #bytes: 3, Cycles: 4



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



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



# Appendix A : Instruction Set (19/19)

DJNZ <byte>, rel

## Decrement and Jump if Not Zero

DJNZ Rn, rel

**Operation :**  
 $(PC) \leftarrow (PC) + 2$   
 $(Rn) \leftarrow (Rn) - 1$   
 If  $(Rn) \neq 0$ , then  $(PC) \leftarrow (PC) + rel$

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

1	1	0	1	1	r	r	r	relative addr
---	---	---	---	---	---	---	---	---------------

DJNZ direct, rel

**Operation :**  
 $(PC) \leftarrow (PC) + 3$   
 $(direct) \leftarrow (direct) - 1$   
 If  $(direct) \neq 0$ ,  
 then  $(PC) \leftarrow (PC) + rel$

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

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

NOP

## No Operation

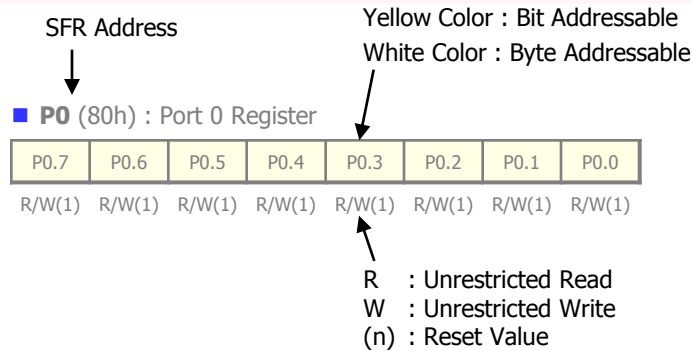
**Operation :**  $(PC) \leftarrow (PC) + 1$

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

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

# Appendix B : SFR Description [80h ~ 87h] (1/7)

## [How to Read a SFR Descriptions]



### ■ **PO** (80h) : Port 0 Register

-	-	P0.5	P0.4	-	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

SMOD1	-	-	POF	GF1	GF0	PD	IDL
-------	---	---	-----	-----	-----	----	-----

R/W(0) R/W(1) R/W(0) R/W(0) R/W(0) R/W(0)

- ◆ SMOD1 : Timer 1 baudrate double in UART mode 1.
- ◆ POF : Power off flag.  
When power-on, this bit will be set by H/W.
- ◆ GF1, GF0 : General purpose flag bit.
- ◆ PD : Power-down (Stop) mode enable.
- ◆ IDL : IDLE mode enable.

# Appendix B : SFR Description [88h ~ 90h] (2/7)

## ■ 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 type 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 type 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

## ■ TLO (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
- ◆ P1.1 : XTAL 2 alternative
- ◆ P1.2 : RESETB alternative

# Appendix B : SFR Description [91h ~ A8h] (3/7)

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

-	-	-	IE2	XT/RG	RGMD	RGSL	BGS
R/W(0)	R/W(0)	R/W(0)	R(1)	R/W(0)	R/W(0)	R/W(1)	

- ◆ IE2 : External interrupt 2 flag. Cleared by S/W.
- ◆ XT/RG : System clock selection  
0 = Internal Ring oscillator is selected as system clock.  
1 = External clock is selected as system clock.
- ◆ RGMD : Ring mode. Now system clock is Ring or XTAL.  
Generally RGMD is the invert of XT/RG.
- ◆ RGSL : Ring select bit when power-down wake-up.  
1 = When wake-up from power-down mode in XTAL clock, use Ring oscillator as system clock during 65,536 XTAL clocks.
- ◆ BGS : Band-gap select. (Default = 1)  
0 = Band-gap block (LVD) will do not run in power-down mode, but function during normal mode.  
It will support the significant power savings in power-down mode.  
1 = Band-gap block (LVD) will run in power-down mode.

## ■ SCON (98h) : Serial Port Control Register of UART0

-	-	-	REN	-	-	TI	RI
			R/W(0)			R/W(0)	R/W(0)

- ◆ REN : Serial reception enable.
- ◆ TI : Transmission interrupt flag. Must be cleared by S/W.
- ◆ RI : Reception interrupt flag. Must be cleared by S/W.

## ■ SBUF (99h) : Serial Data Buffer Register

SBUF.7	SBUF.6	SBUF.5	SBUF.4	SBUF.3	SBUF.2	SBUF.1	SBUF.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)

- ◆ Transmission buffer and reception buffer are separated.
- ◆ Read and write address are same.

## ■ P2 (A0h) : Port 2 Register

-	-	P2.5	P2.4	P2.3	P2.2	-	-
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 2 Register

## ■ IE (A8h) : Interrupt Enable Register

EA	EADC	-	ES	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.
- ◆ ES : Serial port interrupt enable.
- ◆ ET1 : Timer 1 interrupt enable.
- ◆ EX1 : External interrupt 1 enable.
- ◆ ET0 : Timer0 interrupt enable.
- ◆ EX0 : External interrupt 0 enable.

# Appendix B : SFR Description [B8h ~ D4h] (4/7)

## ■ IP (B8h) : Interrupt Priority Register

-	PADC	-	PS	PT1	PX1	PT0	PX0
R(1)	R/W(0)		R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- ◆ PADC : ADC interrupt priority.
- ◆ PS : Serial port 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 Ring Oscillator Control Register

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

- ◆ RINGON : 1 = Internal ring oscillator is running.  
0 = Internal ring oscillator is killed.  
Don't clear RINGON bit when XTRG = 0.
- ◆ DIV1, DIV0 : Ring oscillator divider.  
[0,0] = 3.65MHz/1  
[0,1] = 3.65MHz/2  
[1,0] = 3.65MHz/4  
[1,1] = 3.65MHz/8

## ■ 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.

## ■ 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.5	P0TYPE.4	-	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)



# Appendix B : SFR Description [D5h ~ E0h] (5/7)

## ■ P1TYPE (D5h) : Port 1 Type Control Register

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

- ◆ 0 = Push-pull output (Default)

## ■ P2TYPE (D6h) : Port 2 Type Control Register

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

- ◆ 0 = Push-pull Output (Default)

## ■ WDCON (D8h) : Watchdog Timer Control Register

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

- ◆ WD1. WD0 : Watchdog timer mode select
  - [0,0] :  $1 \times 2^{16}$  clocks (interrupt) + 256 clocks (reset)
  - [0,1] :  $4 \times 2^{16}$  clocks (interrupt) + 256 clocks (reset)
  - [1,0] :  $16 \times 2^{16}$  clocks (interrupt) + 256 clocks (reset)
  - [1,1] :  $32 \times 2^{16}$  clocks (interrupt) + 256 clocks (reset)
- ◆ WDIF : Watchdog timer interrupt flag.
- ◆ WTRF : Watchdog timer reset flag. Cleared by S/W.
- ◆ EWT : Watchdog timer reset enable.
- ◆ RWT : Restart watchdog tier.

## ■ PWMCON (DCh) : PWM Control Register

-	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)

- ◆ 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] =  $F_{Osc}/128$
- \* **PWM Clock ( $F_{PWM}$ ) to ADC should not be set to  $F_{osc}/1$ .**
- ◆ 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)

# Appendix B : SFR Description [E1h ~ E6h] (6/7)

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

ADC11B	ADC10B	ADC9B	ADC8B	-	-	-	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.
- ◆ ADC4B : 0 = ADC4 input enable & digital input disable at P0.5.

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

ADC3B	-	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 /  $AV_{REF}$  input enable & digital input disable at P0.4
- ◆ 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)
  - ⋮
  - [1,0,1,1] = ADC11 Selection (=Bh)
- \* Ch, Dh, Eh, Fh : All ADC input disable

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

IOXEN	IORSTEN	-	PWM00	TVO	TX	-	-
-------	---------	---	-------	-----	----	---	---

R/W(0) 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.
- ◆ PWMD0 : 1 = PWM waveform output enable to P0.0.
- ◆ TVO : 1 = Timer 0 overflow clock to P0.0.
- ◆ TX : 1 = UART TX data output to P0.2.  
User must set TX bit to use UART.

## ■ POSEL (E4h) : Port 0 Pull-up Control Register

-	-	POSEL.5	POSEL.4	-	POSEL.2	POSEL.1	POSEL.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 = Internal Pull-up resistor is ON (Default)
- ◆ 1 = Internal Pull-up resistor is OFF when ADC\_EN (ADCON[7]) = 1

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

-	-	-	-	-	-	P1SEL.1	P1SEL.0
---	---	---	---	---	---	---------	---------

R/W(1) R/W(1)

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

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

-	-	P2SEL.5	P2SEL.4	P2SEL.3	P2SEL.2	-	-
---	---	---------	---------	---------	---------	---	---

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 = Internal Pull-up resistor is ON (Default)
- ◆ 1 = Internal Pull-up resistor is OFF when ADC\_EN (ADCON[7]) = 1

# Appendix B : SFR Description [E8h ~ F8h] (7/7)

## ■ EIE (E8h) : Extended Interrupt Enable Register

-	-	EPWM	EWDT	-	-	-	EX2
		R/W(0)	R/W(0)			R/W(0)	R/W(0)

- ◆ EPWM : PWM interrupt enable.
- ◆ EWDT : Watchdog interrupt enable.
- ◆ EX2 : External 2 interrupt enable.

## ■ 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	AVREF	ADIV	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.
- ◆ AVREF : 1 = ADC reference voltage enable from P0.4.
- ◆ ADIV : ADC input clock select  
0 = System clock ( $F_{OSC}$ ) / 2. (Default)  
1 = PWM input clock ( $F_{PWM}$ )
- ◆ 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)

## ■ PODIR (F4h) : Port 0 Input/Output Control Register

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

- ◆ 1 = Input (Default) / 0 = Output

## ■ EIP (F8h) : Extended Interrupt Priority Register

-	-	PPWM	PWDT	-	-	-	PX2
		R/W(0)	R/W(0)			R/W(0)	R/W(0)

- ◆ PPWM : PWM interrupt priority bit.
- ◆ PWDT : Watchdog timer interrupt priority bit.
- ◆ PX2 : External interrupt 2 priority bit.

## Appendix C : Update History

- ◆ V1.0
  - ✓ First Release
- ◆ V1.2
  - ✓ EPROM 2kB → 4kB