BM

Brief Manual of HallCore110 Family

An 8-bit MCU with Hall Sensor

V1.0

July 2010

Contents

1. Product Overview

- **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 HallCore110 Family:**
  - ✓ **10 bit ADC / 8-bit PWM / UART / WDT / LVD / POR / I2C.**
- **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)**

1. Product Overview (Cont'd)

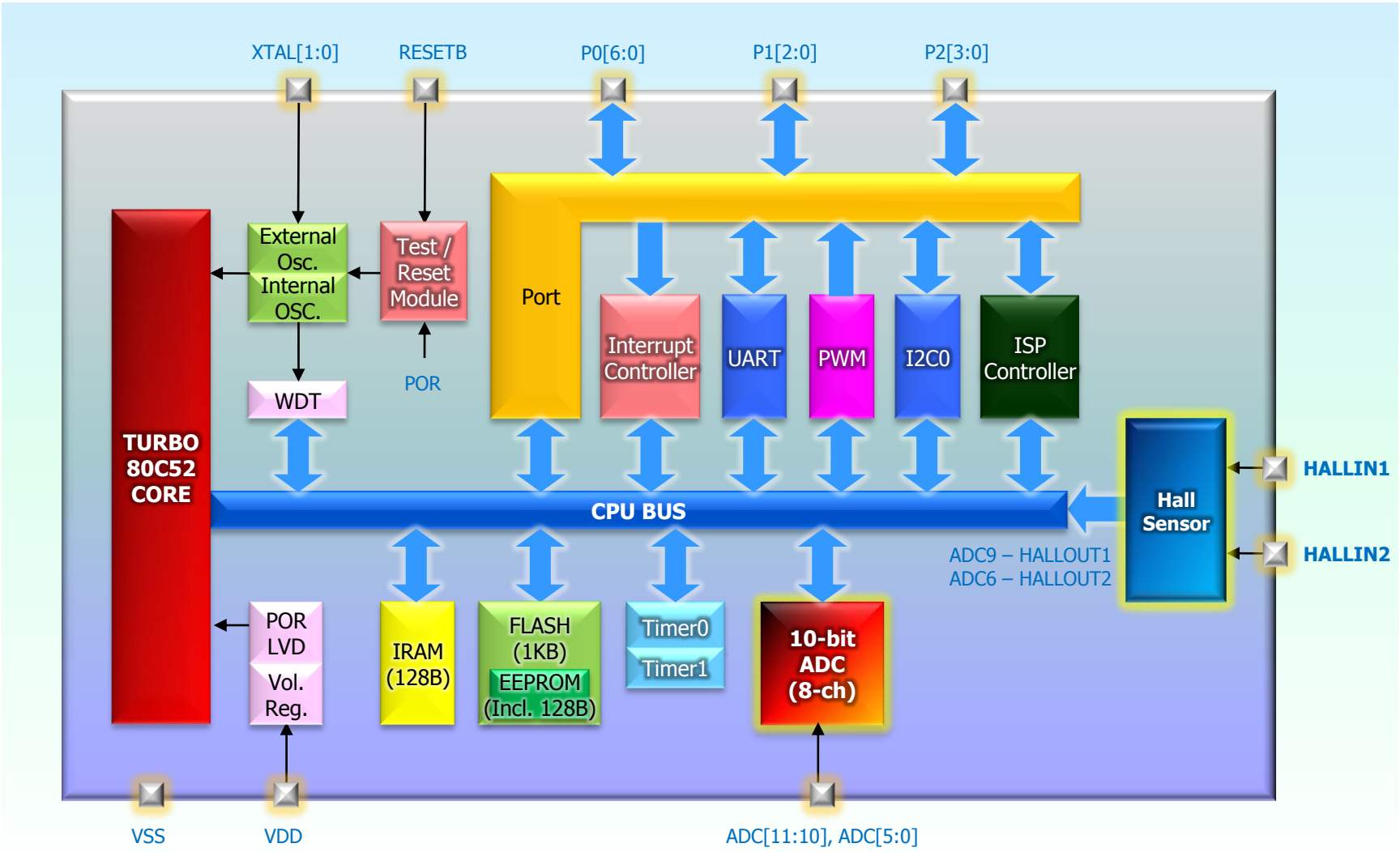
A. HallCore110 Family

Product	FLASH [Byte]	EEPROM [Byte]	RAM [Byte]	Volt [V]	Freq [MHz]	T/C [16 bits]	COM I/O	ADC (bit X Ch)	Hall Sensor	Touch Channel	Package	Others
HallCore110-ML20I	1k	(128)	128	2.2 ~ 3.6	6	2	1 UART 1 I2C	10 X 8	1-axis	None	20-MLF	IAP ISP EJTAG LVD POR RING

2. Features

- CPU
  - ✓ 8-bit turbo 80C52 architecture
  - ✓ 4 cycles/1 machine cycle
  - ✓ instruction level compatible with Intel 80C52
- 1kB FLASH (Including 128B User EEPROM)
- 128B Internal RAM
- Operating Voltage : +2.2V to +3.6V
- Operating Frequency : Max. 6MHz
- Max. Programmable 16 I/O Pins
  - ✓ Pull-up control, Open drain, & Push-Pull output
  - ✓ TTL and CMOS compatible logic levels
- Low Voltage Detector (LVD) : +2.2V
- Internal Ring OSC with Calibration function
  - ✓ 4MHz @+3.0V (+/- 3%)
  - ✓ 32kHz @+2.7V (+/- 10%) : Low Power OSC.
- **1-aixis Hall Sensor**
- **8-channel 10-bit ADC**
- Supporting ISP/IAP/MDS
- Two 16-bit Timer / Counter
- 16-bit Programmable Watchdog Timer
- **1-channel I2C Comm. (Master/Slave)**
- 1-channel UART Comm.
- **1-channel 8-bit High Speed PWM**
- 11 Interrupt Sources
  - ✓ Timer0/1, WDT, ADC, UART, PWM, I2C0
  - ✓ 4 External Interrupt Sources : Both Edge/Level
  - ✓ Two-level Interrupt Priority
- Reset Sources
  - ✓ On-chip Power-On-Reset (POR)
  - ✓ External Reset
  - ✓ Low Voltage Detector Reset (LVR)
  - ✓ Watchdog Timer Reset
- Power Down Wake-up Sources
  - ✓ Reset Sources + 4 External Interrupt (Both Levels)
  - ✓ WDT interrupt
- Power Consumption
  - ✓ Active Current : Max. 2mA @+3.0V, 2MHz
  - ✓ Idle Current : Max. 0.5mA @+3.0V, 2MHz
  - ✓ Stop Current : Max. 5uA @+3.0V (WDT ON)
- E.S.D. Protection up to 2,000 V
  - ✓ 2,000V for Normal I/O Pin
- Latch-up Protection Up to ±200mA
- Package
  - ✓ 20-MLF

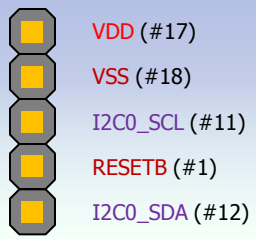
3. Block Diagram

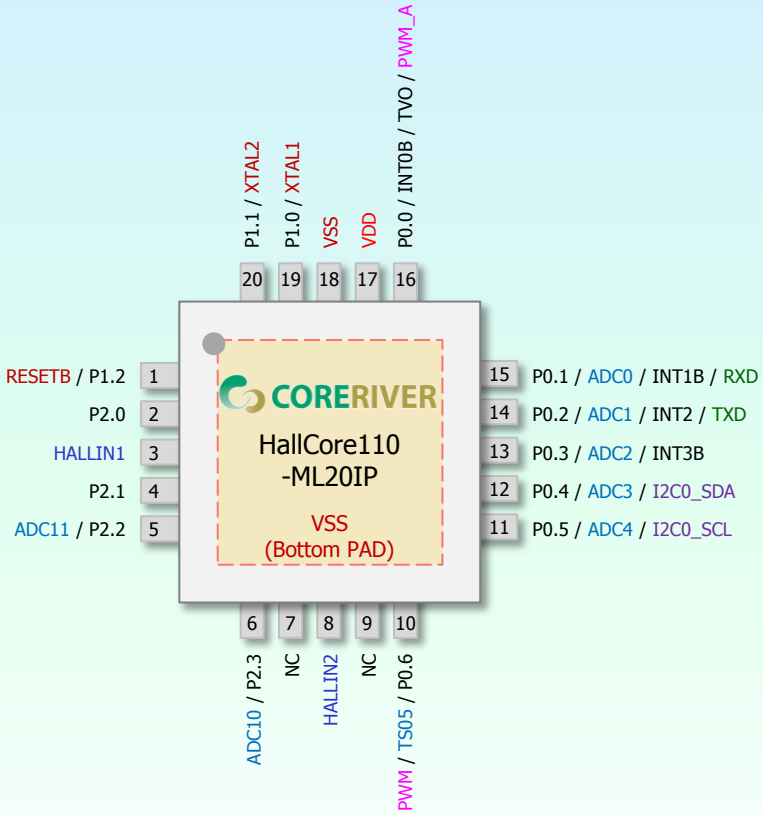


4. Pin Configurations

- Ordering Information : HallCore110-ML20IP
- Hall Sensor : 1-axis
- ADC : 8 Channels
- ISP / MDS : 1 Channel
- I2C Comm. : 1 Channel
- UART Comm. : 1 Channel
- PWM : 1 Channel
- I/O : 16 Pins

ISP / MDS Pin Configuration





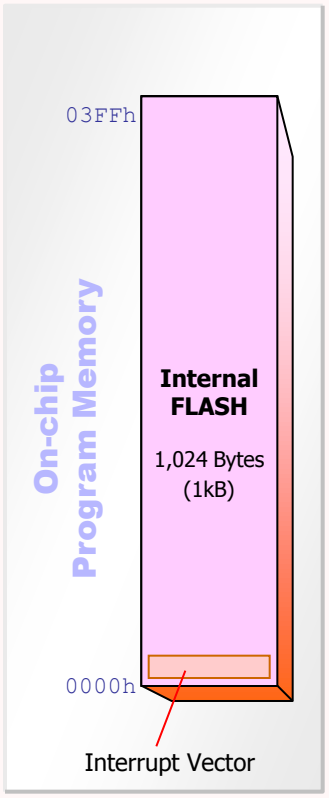
Two pins (#7 & #9) must be N/C (Not Connected).

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

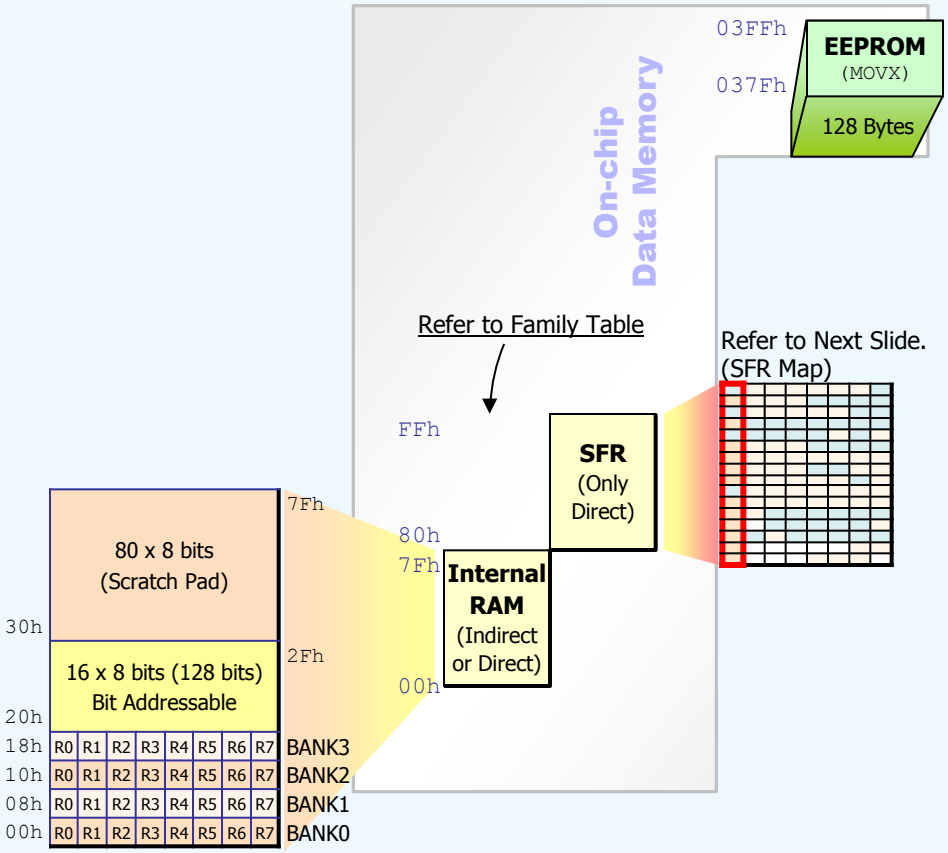
5. Pin Descriptions

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[6:0]	Input/Output	▪ Bit Programmable with Schmitt Trigger - Optional Pull-up Control Enable - Open-drain Output - Push-pull Output (Default)	P0.0 / INT0B / TVO / PWM_A P0.1 / ADC0 / INT1B / RXD P0.2 / ADC1 / INT2 / TXD P0.3 / ADC2 / INT3B P0.4 / ADC3 / I2C0_SDA P0.5 / ADC4 / I2C0_SCL P0.6 / ADC5 / PWM
P2[3:0]	Input/Output	▪ Bit Programmable with Schmitt Trigger - Optional Pull-up Control Enable - Open-drain Output - Push-pull Output (Default)	P2.0 P2.1 P2.2 / ADC11 P2.3 / ADC10
HALLIN[1:0]	Input	▪ Hall Sensor Input	

6.1. Memory Organization



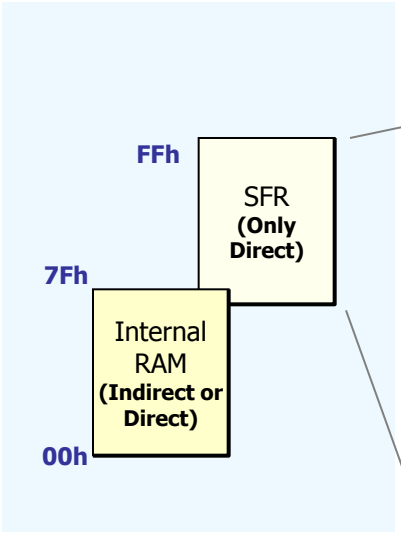
[On-chip Program Memory]
(Read/Write with IAP)



[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



: Newly added SFR at HallCore110 Family

: Reserved for future use.

Bit addressable

F8h	EIP	EECNTLD	EECNTL	EECNTM	EECNTH			EEAEN	FFh
F0h	B				P0DIR	P1DIR	P2DIR		F7h
E8h	EIE						ADCR	ADCON	EFh
E0h	ACC	ADCSELH	ADCSEL	ALTSEL	P0SEL	P1SEL	P2SEL		E7h
D8h	WDCON			ADCHSEL	PWMCON	PWMIF	PWMD		DFh
D0h	PSW	WDMOD			P0TYPE	P1TYPE	P2TYPE		D7h
C8h									CFh
C0h					PMR	STATUS			C7h
B8h	IP						OSCICN	OSC2ICN	BFh
B0h	I2CST0			I2CCON0	I2CCFG0	I2CSLA0	I2CDAT0	I2CI2C0_SCL	B7h
A8h	IE								AFh
A0h	P2								A7h
98h	SCON	SBUF							9Fh
90h	P1	EXIF			CLKOFF	RINGCON			97h
88h	TCON	TMOD	TL0	TL1	TH0	TH1			8Fh
80h	P0	SP	DPL	DPH			ITSEL	PCON	87h

6.2. SFR Brief Description

- 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	*1111111 *****111 ****1111
IP **IE**	Interrupt Priority Interrupt Enable Control	10*00000 00*00000
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
SCON **SBUF**	Serial Control Serial Buffer	***0**00 00000000
PCON	Power Control	0**10000

- Newly added SFR Registers at HallCore110 Family

Register	Name	Reset Value
P0SEL **P1SEL** **P2SEL**	Port 0 Pull-up Control Port 1 Pull-up Control Port 2 Pull-up Control	*0000000 *****011 ****0000
P0TYPE **P1TYPE** **P2TYPE**	Port 0 Type Control Port 1 Type Control Port 2 Type Control	*0000000 *****000 ****0000
P0DIR **P1DIR** **P2DIR**	Port 0 Input/Output Control Port 1 Input/Output Control Port 2 Input/Output Control	*1111111 *****111 ****1111
ALTSEL	Alternative Function Control	000000**
PWMCON **PWMIF** **PWMD**	PWM Control PWM Interrupt Flag PWM Duty Data	00000000 *******0 00000000
ADCON **ADCR** **ADCSEL** **ADCSELH** **ADCHSEL**	ADC Control & ADC Result Low ADC Result High ADC Channel Selection Low and MUX Selection ADC High Channel Selection ADC High Channel Selection Register	0010*000 00000000 11111111 111**111 0*******
WDCON **WDMOD** **PMR** **EXIF** **EIP** **EIE** **STATUS** **OSCICN** **OSC2ICN** **ITSEL**	Watchdog Timer Control Watchdog Mode register Power Management Control Added External Interrupt and LVD Control and Extended Interrupt Priority Extended Interrupt Enable Crystal Status Internal Ring Oscillator Control Internal Ring2 Oscillator Control Register External Interrupt Control	11010000 *******0 ****0*** 01000101 **000000 **000000 ***0**** ****0100 ******01 11**0100

*** : Don't touch bit.**

6.2. SFR Brief Description (Cont'd)

- Newly added SFR Registers at HallCore110 Family (Cont'd)

Register	Name	Reset Value
CLKOFF **RINGCON**	Clock Control for Minimizing Power Consumption Internal Ring Oscillator Tuning	**00**00 01100000
EEAEN	EEPROM Access Enable	*******0
I2CCON0 **I2CST0** **I2CCFG0** **I2CSLA0** **I2CDAT0** **I2CI2C0_SCL**	I^2C Control 0 I^2C Status 0 I^2C Configuration 0 I^2C Slave Address 0 I^2C Data 0 I^2C SCL Clock Scaler	*0100000 00000000 ****0000 00000000 00000000 00000000

*** : Don't touch bit.**

6.3. Instruction Set Summary

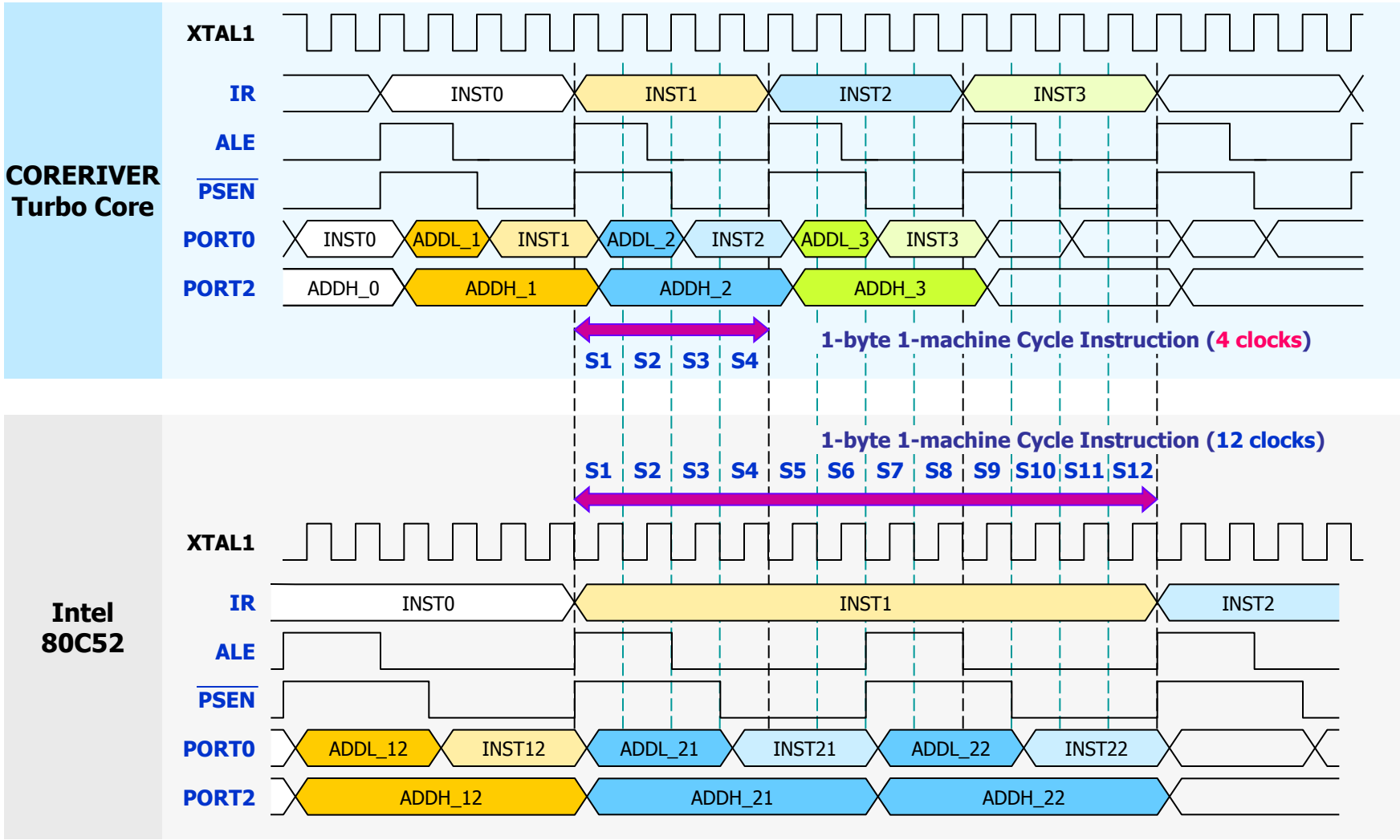
- Refer to Appendix A (Instruction Set) for more details.

Type	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

Type	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

6.4. CPU Timing

- Comparative timing of the HallCore110 family and Intel 80C52



6.4. CPU Timing : Execution Time Table

- The fastest instruction execution in the world

Instruction	**Turbo Core (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[6:0]

- Push-pull output type with enabled pull-ups by default.
- P0[7:0] can be configured as ADC input : ADC6 (P0.7) ~ ADC0 (P0.1) and ADCH7 (P0.0).
- 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.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)

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

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

-	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)

- 0 = Output / 1 = Input (Default)

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

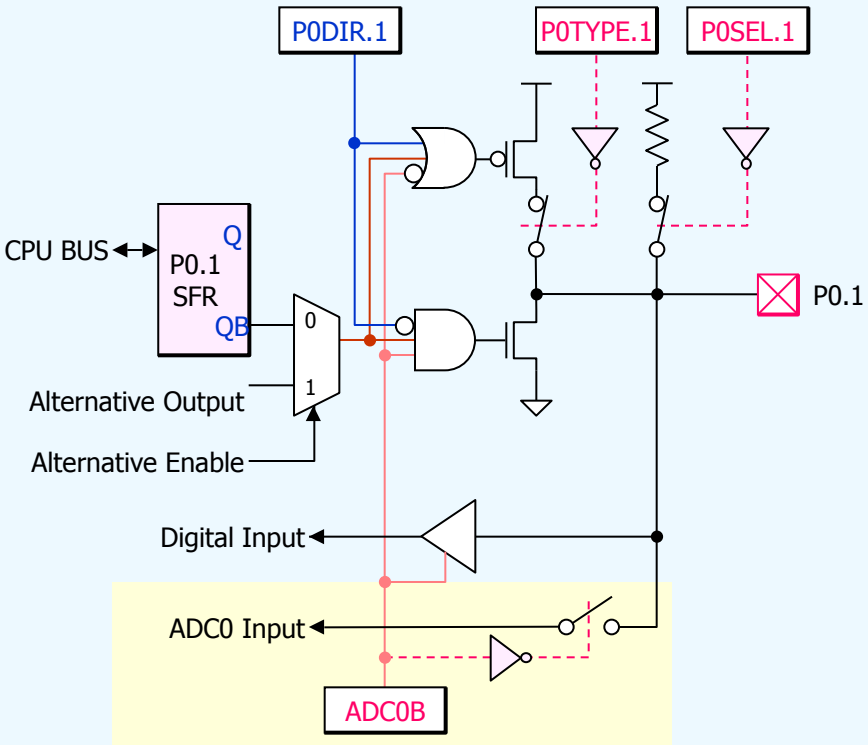
-	P0SEL.6	P0SEL.5	P0SEL.4	P0SEL.3	P0SEL.2	P0SEL.1	P0SEL.0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

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

✓ **P0** (80h) : Port 0 Register

-	P0.6	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)

✓ **ADCSEL** (E2h), **ADCSELH** (E1h), and **ADCHSEL**(DBh)
: Refer to Next Slide



6.5. I/O Ports : PORT0[6:0] (Cont'd)

✓ **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
- CH[3:0] : ADC MUX Selection.
  - 0000b = ADC0 Selection (0h)
  - 0001b = ADC1 Selection (1h)
  - 0010b = ADC2 Selection (2h)
  - 0011b = ADC3 Selection (3h)
  - 0100b = ADC4 Selection (4h)
  - 0101b = ADC5 Selection (5h)
  - 0110b = ADC6 Selection (6h)
  - 0111b = No ADC Input Select (7h)
  - 1000b = No ADC Input Select (8h)
  - 1001b = ADC9 Selection (9h)
  - 1010b = ADC10 Selection (Ah)
  - 1011b = ADC11 Selection (Bh)
  - 1100b = No ADC input select (Ch)
  - 1101b = No ADC input select (Dh)
  - 1110b = No ADC input select (Eh)
  - 1111b = No ADC input select (Fh, Default)

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

ADC11B	ADC10B	ADC9B	-	-	ADC6B	ADC5B	ADC4B
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

✓ **ADCHSEL** (DBh) : ADC High Channel Selection Register

CH_SEL	-	-	-	-	-	-	-
R/W(0)							

- CH_SEL : ADC MUX Selector with CH[3:0].

  0 = CH[3:0] → ADC[11:10, 5:0] Enable (Default)
  1 = Disable

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.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

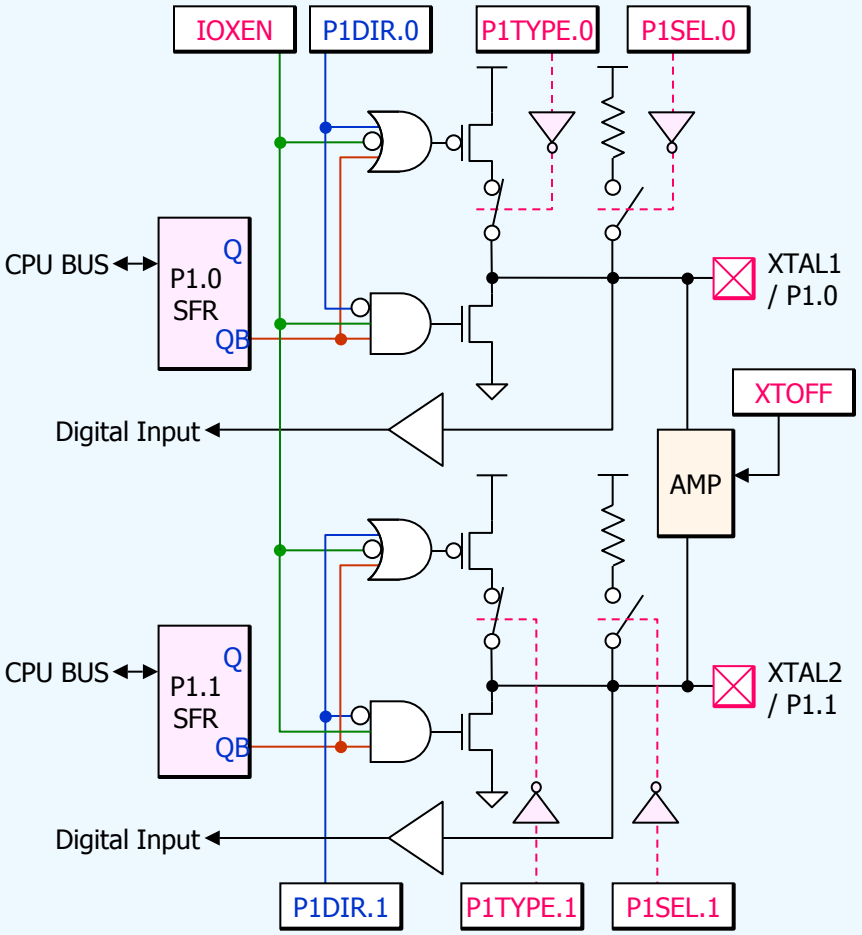
-	-	-	-	-	P1SEL.2	P1SEL.1	P1SEL.0
					R/W(0)	R/W(1)	R/W(1)

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

✓ **P1** (90h) : Port 1 Register

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

✓ **ALTSEL** (E3h), **PMR** (C4h), **STATUS** (C5h), and **EXIF** (91h) : Refer to Next Slide



6.5. I/O Ports : PORT1[1:0] (Cont'd)

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

IOXEN	IORSTEN	CLO	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)			

- 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 to 1 when XT/RG = 1.

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

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

- XTUP : Crystal oscillator warm-up status. (External crystal oscillator)
  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 (EXIF.3) = 1.
  Set by H/W after XTAL stabilization time.

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

-	RTRG	IE3	IE2	XT/RG	RGMD	RGSL	BGS
-	R/W(1)	R/W(0)	R/W(0)	R/W(0)	R(1)	R/W(0)	R/W(1)

- XT/RG : System clock selection.
  0 = Internal Ring oscillator is selected as system clock.
  1 = External clock is selected as system clock.

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

✓ **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)

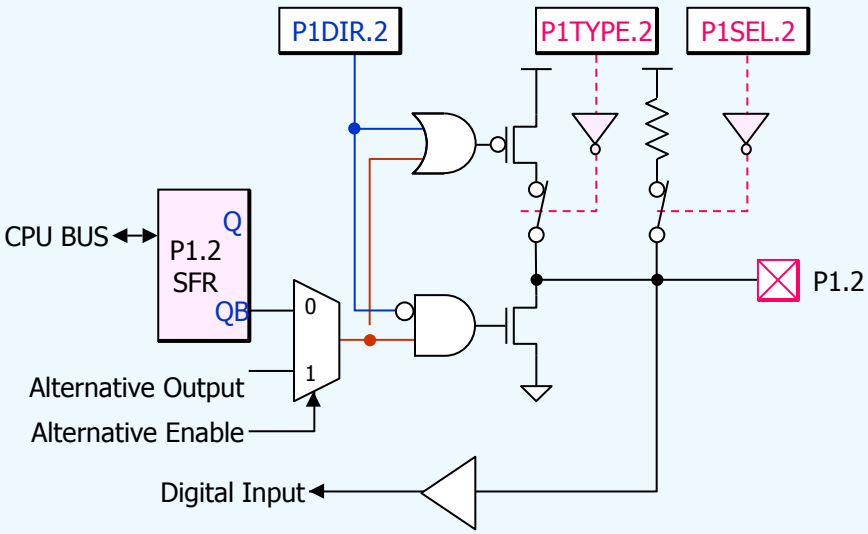
✓ **P1SEL** (E5h) : Port 1 Pull-up Control Register

-	-	-	-	-	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 (Default) when ADC_EN (ADCON[7]) = 1

✓ **P1** (90h) : Port 1 Register

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



6.5. I/O Ports : PORT2[3:0]

- Push-pull type with enabled pull-ups by default.
- P2[3:2] can be configured as ADC input : ADC10 (P2.3) ~ ADC11 (P2.2)
- 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

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

	-	-	-	P2TYPE.3	P2TYPE.2	P2TYPE.1	P2TYPE.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.3	P2DIR.2	P2DIR.1	P2DIR.0
				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

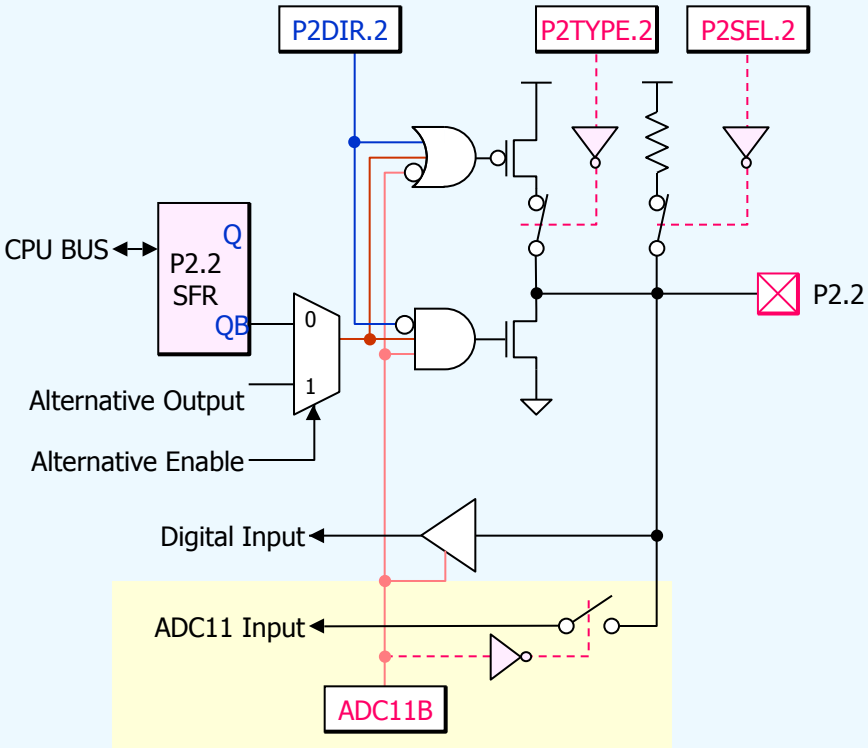
-	-	-	-	P2SEL.3	P2SEL.2	P2SEL.1	P2SEL.0
				R/W(0)	R/W(0)	R/W(0)	R/W(0)

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

✓ **P2** (A0h) : Port 2 Register

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

✓ **ADCSELH** (E1h) : Refer to Next Slide



6.5. I/O Ports : PORT2[7:0] (Cont'd)

✓ **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)

- CH[3:0] : ADC MUX Selection.
  0000b = ADC0 Selection (0h)
  0001b = ADC1 Selection (1h)
  0010b = ADC2 Selection (2h)
  0011b = ADC3 Selection (3h)
  0100b = ADC4 Selection (4h)
  0101b = ADC5 Selection (5h)
  0110b = ADC6 Selection (6h)
  0111b = No ADC Input Select (7h)
  1000b = No ADC Input Select (8h)
  1001b = ADC9 Selection (9h)
  1010b = ADC10 Selection (Ah)
  1011b = ADC11 Selection (Bh)
  1100b = No ADC input select (Ch)
  1101b = No ADC input select (Dh)
  1110b = No ADC input select (Eh)
  1111b = No ADC input select (Fh, Default)

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

ADC11B	ADC10B	ADC9B	-	-	ADC6B	ADC5B	ADC4B
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

✓ **ADCHSEL** (DBh) : ADC High Channel Selection Register

CH_SEL	-	-	-	-	-	-	-
R/W(0)							

- CH_SEL : ADC MUX Selector with CH[3:0].

  0 = CH[3:0] → ADC[11:10, 5:0] Enable (Default)
  1 = Disable

6.6. LVD (Low Voltage Detector)

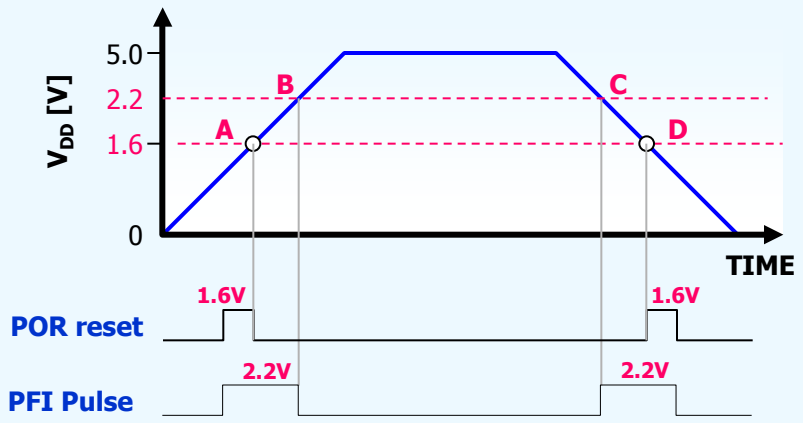
- On-chip power-on reset
  - ✓ 1.6V
- On-chip power-fail interrupt
  - ✓ 2.2V

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

-	RTRG	IE3	IE2	XT/RG	RGMD	RGSL	BGS
-	R/W(1)	R/W(0)	R/W(0)	R/W(0)	R(1)	R/W(0)	R/W(1)

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

  When BGS = 0,
  Band-gap block (LVD) will do not run in power-down mode, but function during normal mode.
  When BGS = 1,
  Band-gap block (LVD) will run in power-down mode.



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

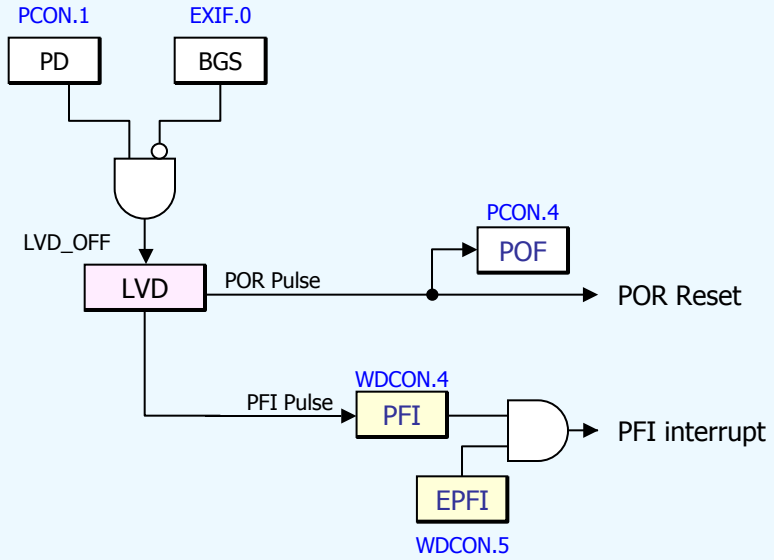
SMOD1	-	-	POF	GF1	GF0	PD	IDL
R/W(0)			R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

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

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

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

- EPFI : Power-fail interrupt enable.
- PFI : Power-fail interrupt flag.



6.7. WDT (Watch Dog Timer)

- 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 counter mode
  - ✓WDCON[7:6], WDMOD[3] : WD1, WD0, WDM

WDM	WD1	WD0	Interrupt Time-out		Reset Time-out
1 (@4MHz)	0	0	2^{19} clocks	131 ms	2^{17} + 256 clocks
	0	1	2^{20} clocks	262 ms	2^{20} + 256 clocks
	1	0	2^{21} clocks	524 ms	2^{23} + 256 clocks
	1	1	2^{22} clocks	1048 ms	2^{26} + 256 clocks
0 (@32KHz)	0	0	2^{9} clocks	16 ms	2^{9} + 256 clocks
	0	1	2^{10} clocks	32 ms	2^{10} + 256 clocks
	1	0	2^{11} clocks	64 ms	2^{11} + 256 clocks
	1	1	2^{12} clocks	128 ms	2^{12} + 256 clocks

✓ **WDMOD** (D1h) : Watchdog Mode register

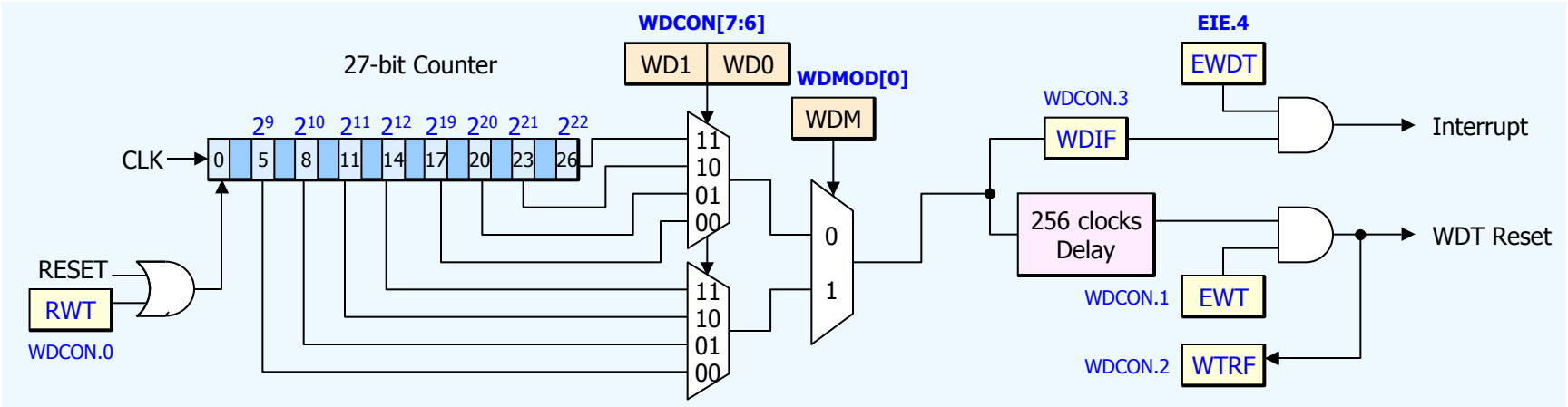
-	-	-	-	-	-	-	WDM
							R/W(0)

- WD2 : Watchdog clock divide mode

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

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

- WD[1:0] : WDT Clock Divide
- EPFI : Enable Power-fail Interrupt
- PFI : Power-Fail interrupt Flag
- WDIF : Watchdog Timer Interrupt Flag
- WTRF : Watchdog Timer Reset Flag
- EWT : Watchdog Timer Reset Enable
- RWT : Restart Watchdog Timer



6.8. 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 : 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 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).

✓ **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)

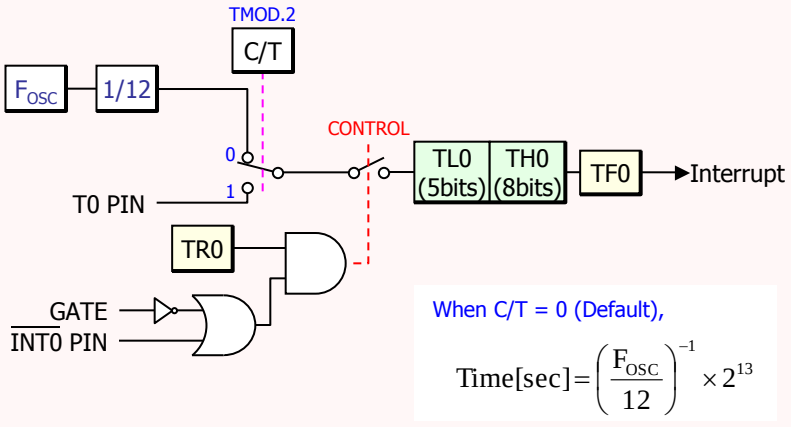
✓ **TL1** (8Bh) : Timer/Counter 1 Low Byte Register

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)

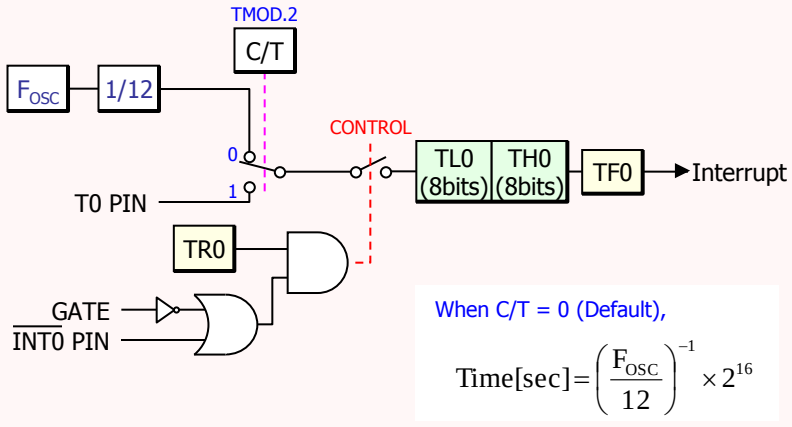
6.8. Timer/Counter : Timer 0 Mode Description



When C/T = 0 (Default),

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

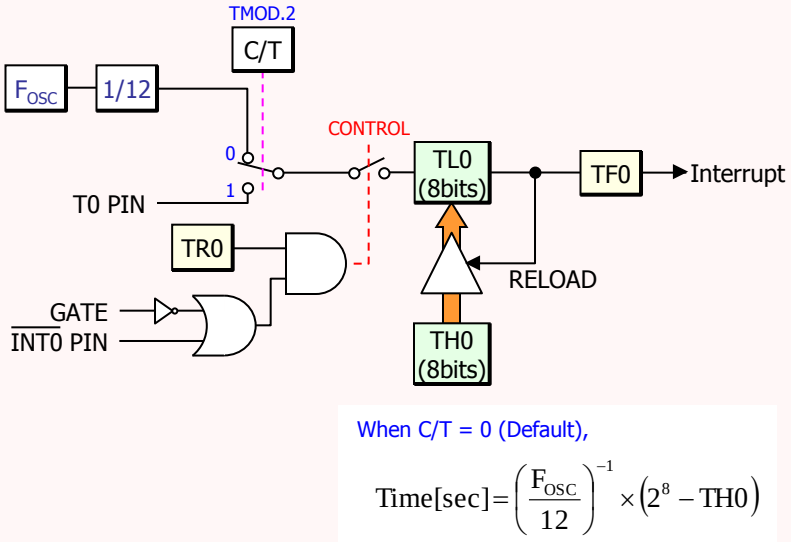
[Mode 0]



When C/T = 0 (Default),

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

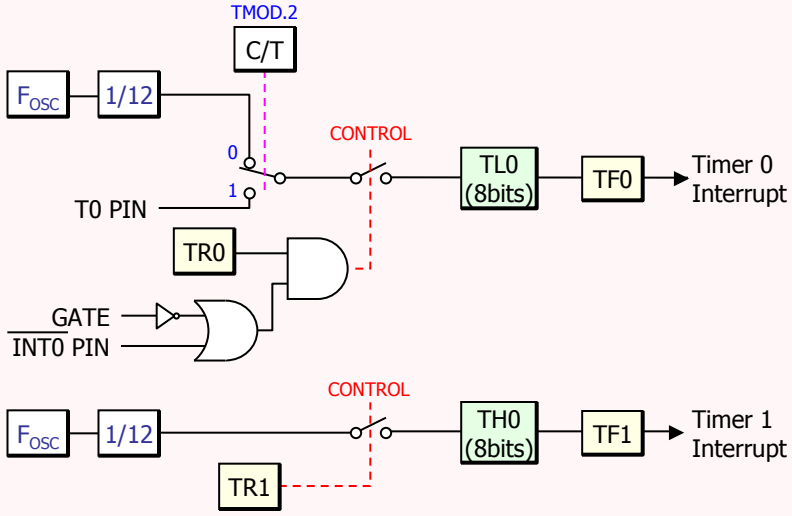
[Mode 1]



When C/T = 0 (Default),

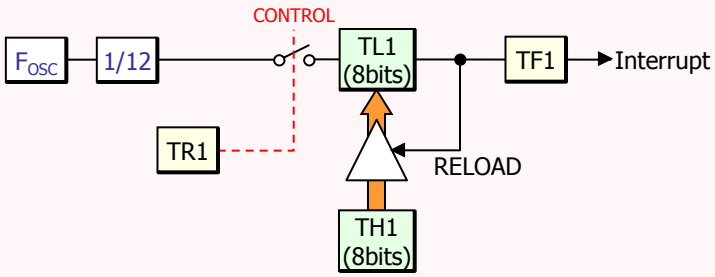
$$\text{Time[sec]} = \left(\frac{F_{OSC}}{12}\right)^{-1} \times \left(2^{8} - \text{TH0}\right)$$

[Mode 2]



[Mode 3]

6.8. Timer/Counter : Timer 1 Mode Description



[Mode 2]

6.9. 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)	1/32 x Timer 1 Overflow (SMOD1=0) 1/16 x Timer 1 Overflow (SMOD1=1)

UART Mode 1
(Using Timer 1 Overflow)

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

[Baudrate Examples]

Baudrate	UART Mode	F_{OSC} [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.
  If user want to receive the data with UART, the REN flag bit is set to 1.
- 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

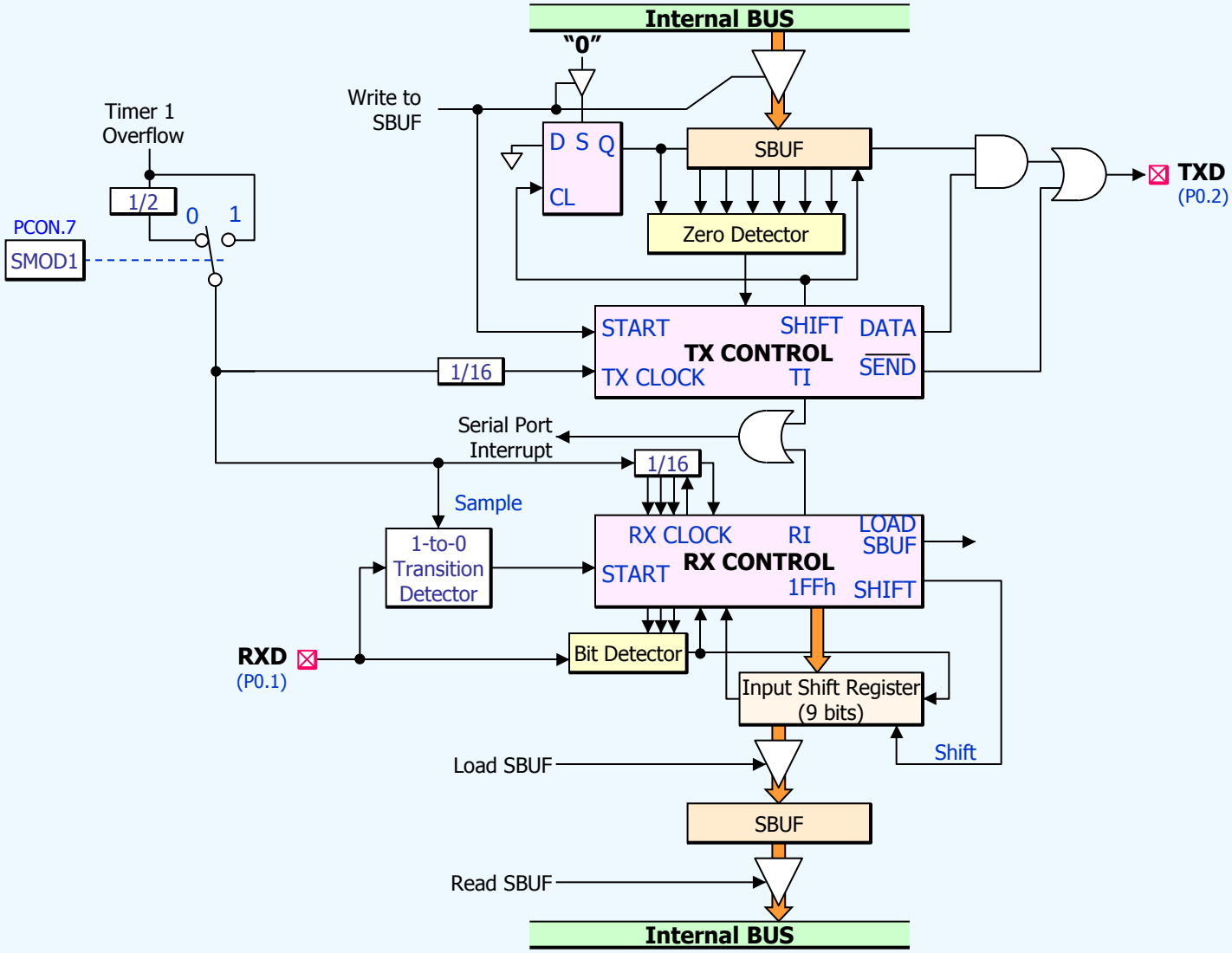
SBUF.7	SBUF.6	SBUF.5	SBUF.4	SBUF.3	SBUF.2	SBUF.1	SBUF.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

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

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

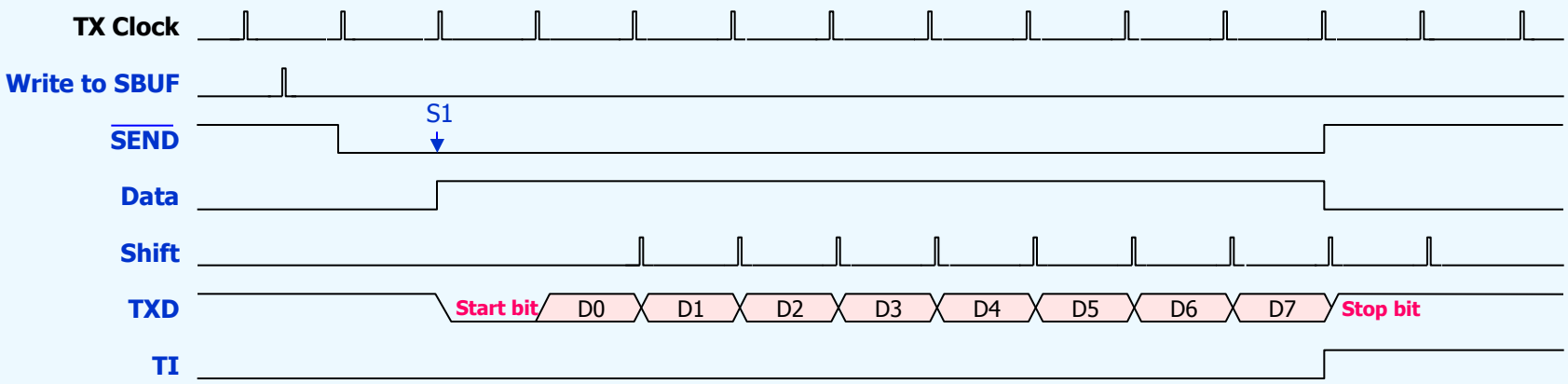
TH1.7	TH1.6	TH1.5	TH1.4	TH1.3	TH1.2	TH1.1	TH1.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

6.9. UART : Mode 1 Function

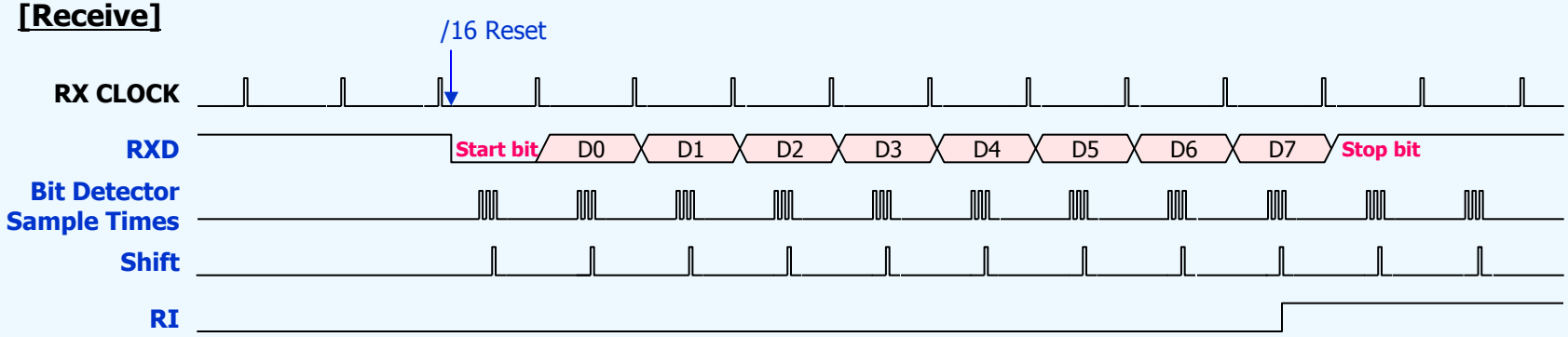


6.9. UART : Mode 1 Timing

[Transmit]



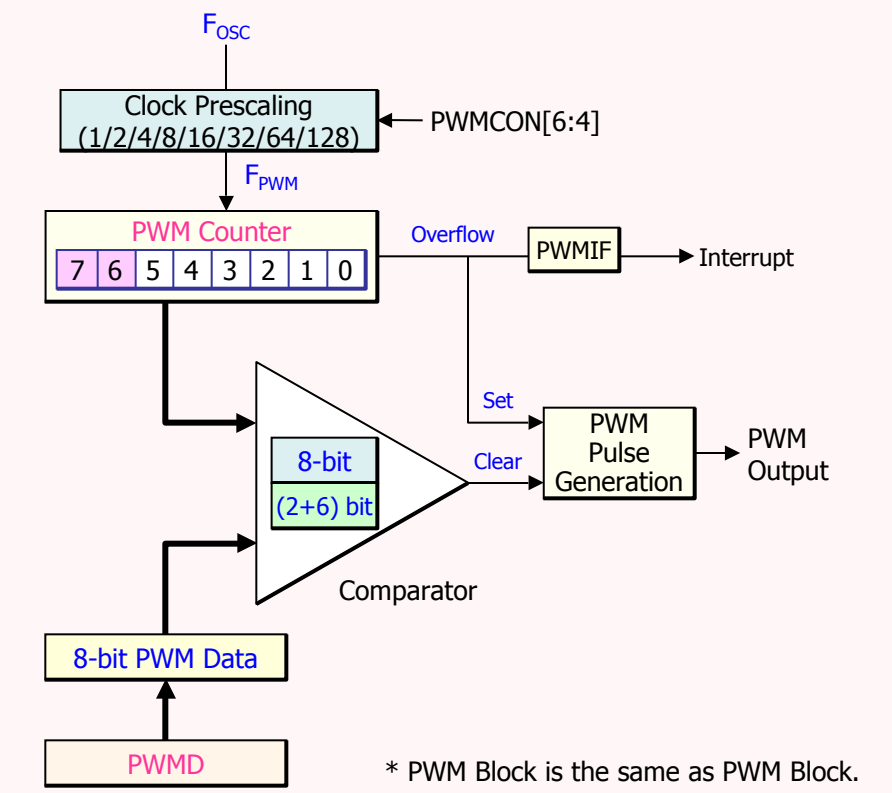
[Receive]



6.10. PWM (Pulse Width Modulator)

- Intelligent 2-channel 8-bit PWM
- PWM Data buffer Update (8-bit / 6-bit Counter Overflow Update)
- PWM Counter can be cleared by S/W.
- PWM is stopped or started (resumed) by S/W.

Mode	Description
8-bit Mode	8-bit Compare
(2+6)-bit Mode	2-bit Extension Compare & 6-bit Compare



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

P0SEL	PS2_P0	PS1_P0	PS0_P0	MODE_P0	RL_P0	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)	R/W(0)

- P0SEL : PWM Waveform Output to Port.
- PS2_P0, PS1_P0, and PS0_P0 : Clock prescale ratio Selection. F_{OSC}/1, /2, /4, /8, /16, /32, /64, /128.
- MODE_P0 : 8-bit / (2+6)-bit Counter Mode Selector. MODE_P0=0, (2+6)-bit Mode MODE_P0=1, 8-bit Mode
- 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.
- CLR_P0 : Counter Reset Enable. Clear by H/W.
- RUN_P0 : Counter Start Enable.

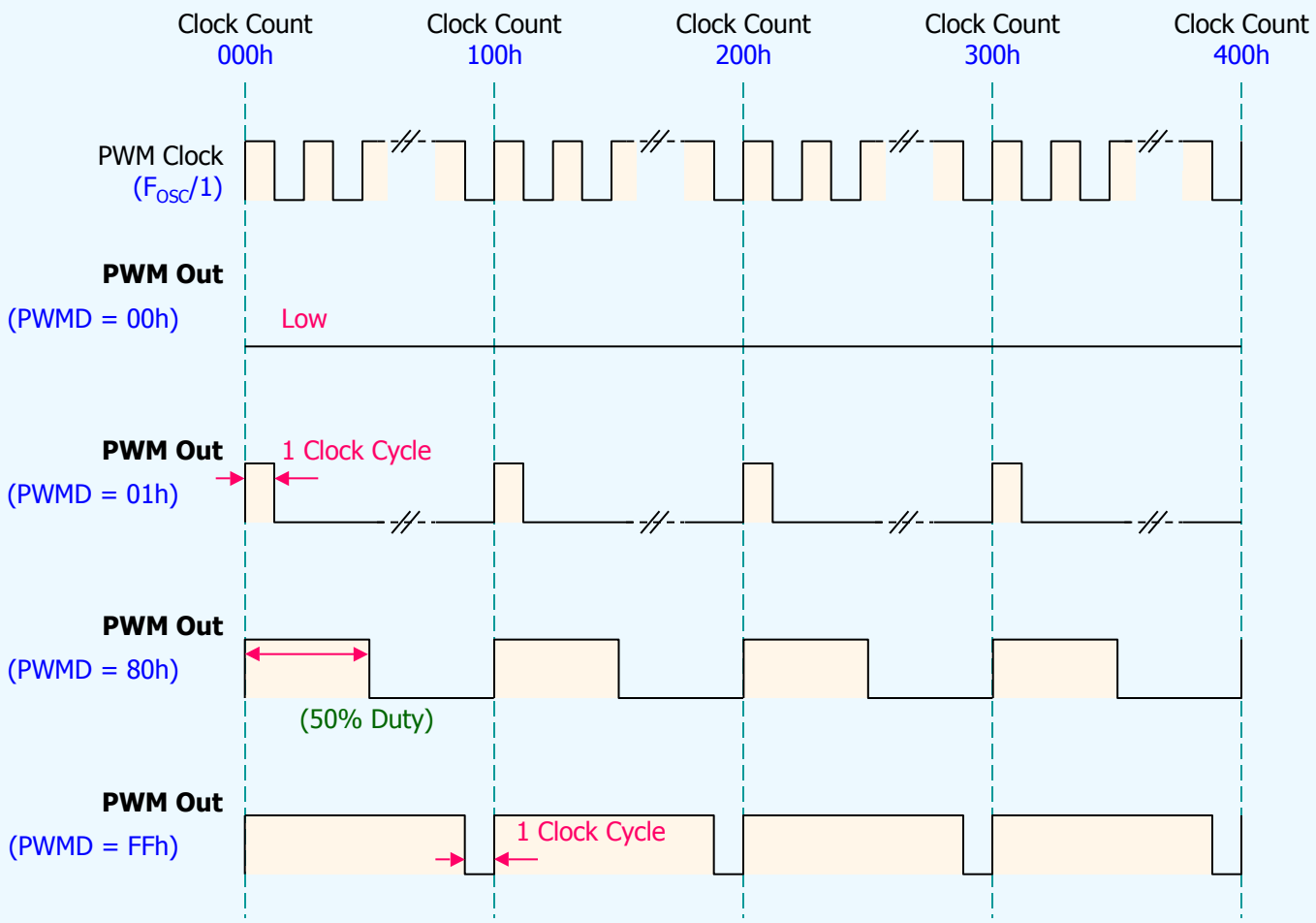
✓ **PWMIF** (DDh) : PWM Interrupt Register

-	-	-	-	-	-	-	PWMIF
							R/W(0)

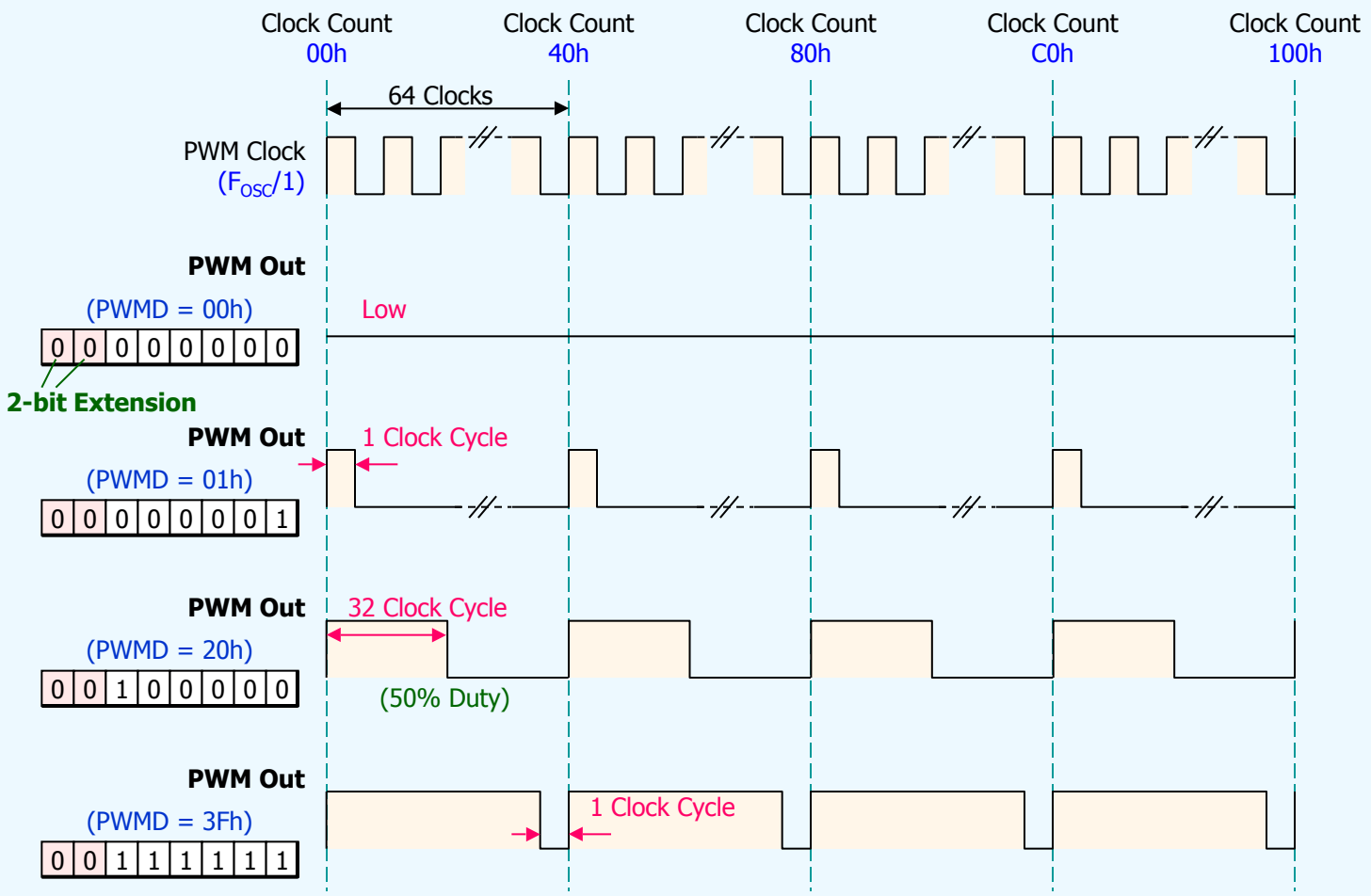
✓ **PWMD** (DEh) : PWM Duty Data Register

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)

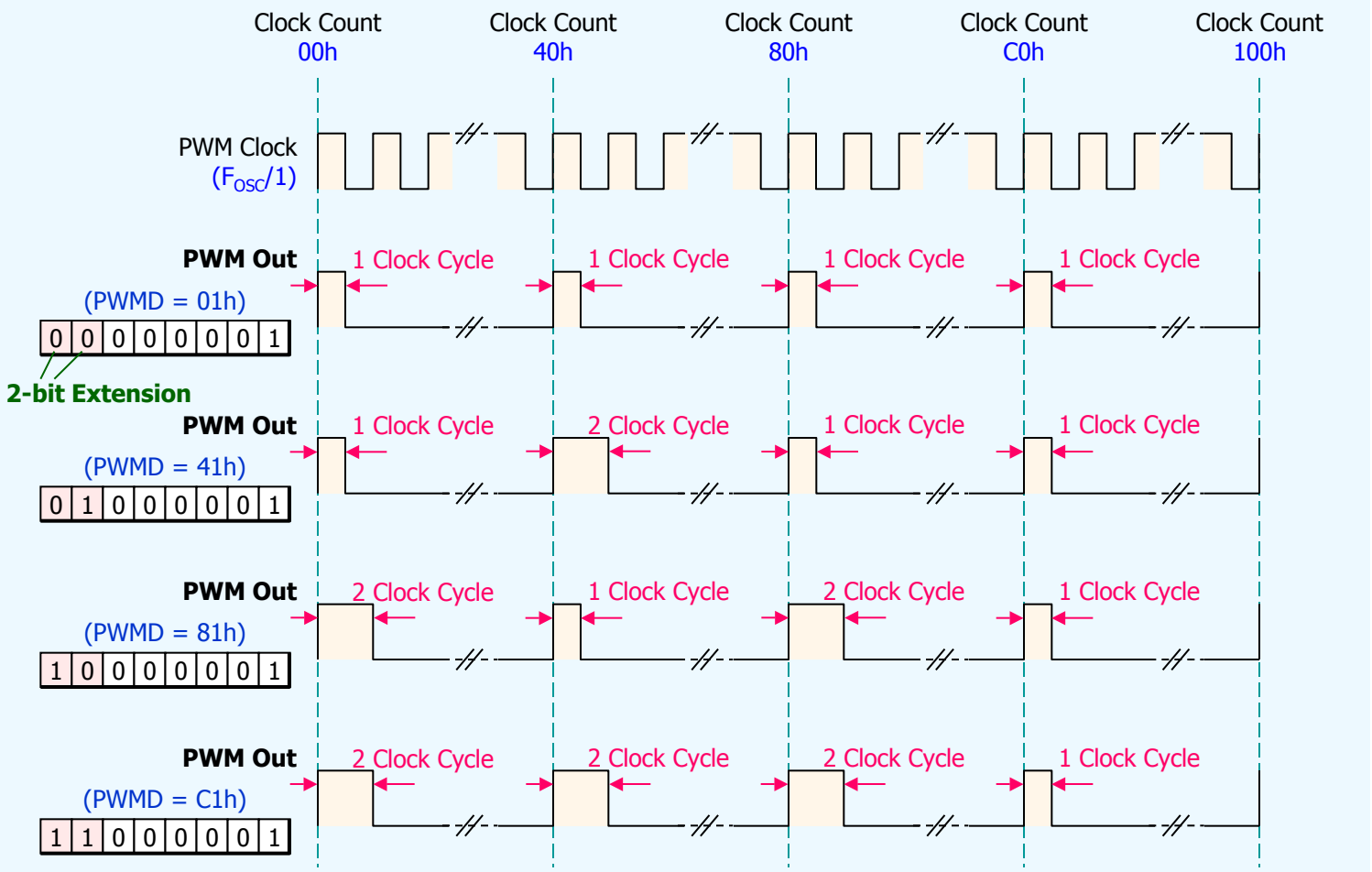
6.11. PWM : 8-bit Mode Pulse Generation



6.11. PWM : The (2+6)-bit Mode



6.11. PWM : The (2+6)-bit Mode (Cont'd)



6.11. ADC (Analog-to-Digital Converter)

- 8-channel 10-bit ADC (SAR Type)
- Max. 52ksps(samples per sec.) @ F_{ADC} = 5MHz & +3.0V

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

ADC11B	ADC10B	ADC9B	-	-	ADC6B	ADC5B	ADC4B
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).

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

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).
- CH[3:0] : ADC MUX Selection.
  0000b = ADC0 Selection (0h)
  0001b = ADC1 Selection (1h)
  0010b = ADC2 Selection (2h)
  0011b = ADC3 Selection (3h)
  0100b = ADC4 Selection (4h)
  0101b = ADC5 Selection (5h)
  0110b = ADC6 Selection (6h)
  0111b = No ADC Input Select (7h)
  1000b = No ADC Input Select (8h)
  1001b = ADC9 Selection (9h)
  1010b = ADC10 Selection (Ah)
  1011b = ADC11 Selection (Bh)
  1100b = No ADC input select (Ch)
  1101b = No ADC input select (Dh)
  1110b = No ADC input select (Eh)
  1111b = No ADC input select (Fh, Default)

✓ **ADCHSEL** (DBh) : ADC High Channel Selection Register

CH_SEL	-	-	-	-	-	-	-
R/W(0)							

- CH_SEL : ADC MUX Selector with CH[3:0].
  0 = CH[3:0] → ADC[11:10, 5:0] Enable (Default)
  1 = Disable

6.11. ADC (Analog-to-Digital Converter) (Cont'd)

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

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

- AD_EN : AD Conversion Enable.
- AD_REQ : AD Conversion Request.
  Cleared by H/W when AD_END goes to 1 from 0.
- AD_END : Current ADC Status.
  0 = ADC is running now.
  User must check the ADCF instead of AD_END.
- ADCF : ADC Interrupt Flag.
  Must be cleared by S/W.
- ADIV : ADC Input Clock (F_{ADC}) Select.
  0 = System Clock (F_{OSC}) / 2. (Default)
  1 = PWM Input Clock (F_{PWM})
  PWM Clock for ADC should not be set to F_{OSC}/1.
- SAR[1:0] : Low Bits of ADC Result Value. (Total 10 bits)

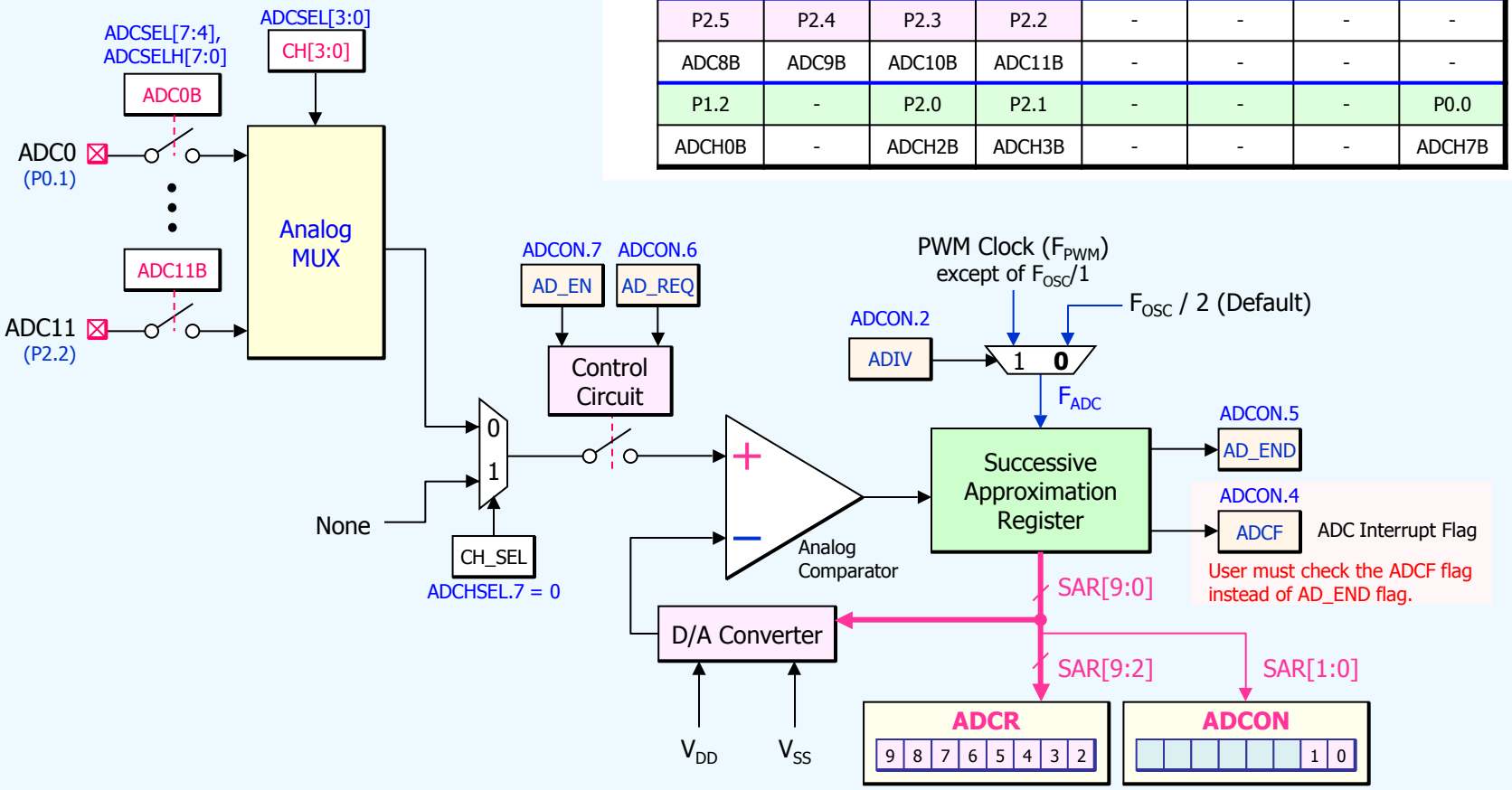
✓ **ADCR** (EEh) : ADC Result High Register

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)

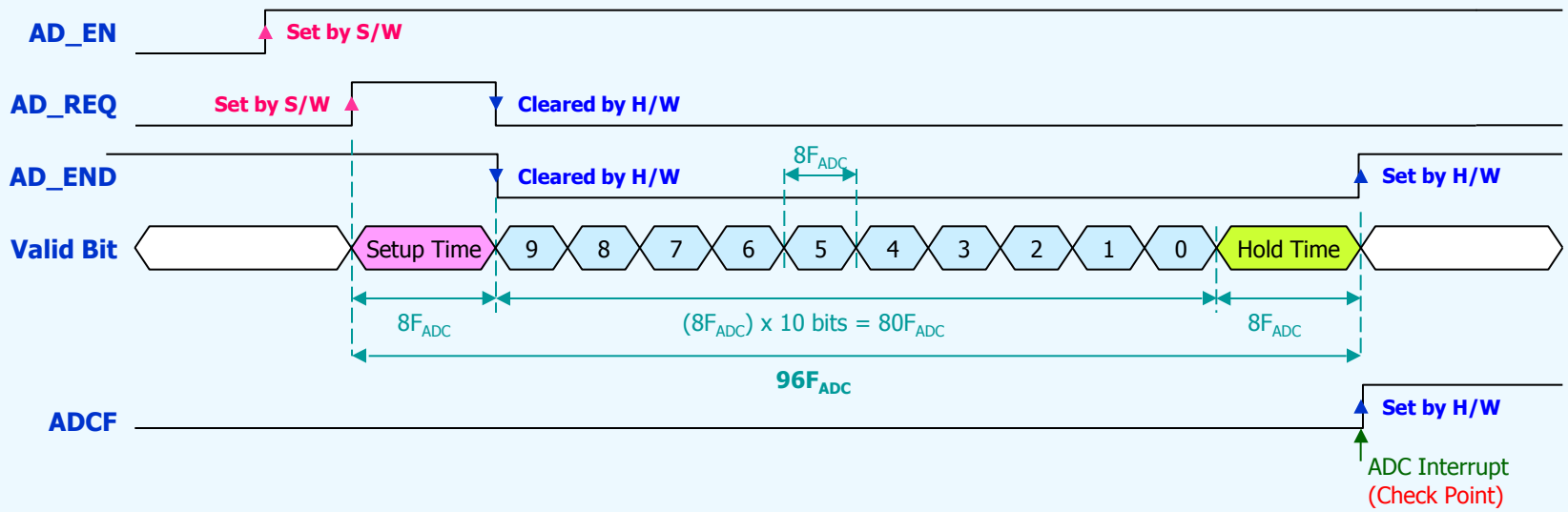
6.11. ADC (Analog-to-Digital Converter)

[ADC Input Channel Selectors versus Port Pins]

P0.1	P0.2	P0.3	P0.4	P0.5	P0.6	P0.7	P2.6
ADC0B	ADC1B	ADC2B	ADC3B	ADC4B	ADC5B	ADC6B	ADC7B
P2.5	P2.4	P2.3	P2.2	-	-	-	-
ADC8B	ADC9B	ADC10B	ADC11B	-	-	-	-
P1.2	-	P2.0	P2.1	-	-	-	P0.0
ADCH0B	-	ADCH2B	ADCH3B	-	-	-	ADCH7B



6.11. ADC : Conversion Timing



- ✓ **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_{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

6.12. I2C : SFR

- Two-wire Interface
- 1-channel I2C
  - ✓I2C0 : Master or Slave Operation
- Transmitter or Receiver Operation
- 100Kbps (Min. Fosc = 1MHz), 400Kbps (Min. Fosc = 4MHz)
- 7bits / 10bits (Extended 15bits) Address Mode
- Transfer Wait State
- Fully Programmable Slave Address
- SDA/SCL Schmitt-trigger input
- 256 Programmable Bit Rates
- Wake-up from IDLE mode
- Compatible with Phillips I2C protocol

✓ **EIE** (E8h) : Extended Interrupt Enable Register

-	-	EPWM	EWDT	-	EI2C0	EX3	EX2
R/W(0)	R/W(0)	R/W(0)	R/W(0)		R/W(0)	R/W(0)	R/W(0)

- EI2C0 : I2C 0 Interrupt Enable

✓ **I2CSLA0** (B5h) : I²C Slave Address Register

SLA1.7	SLA1.6	SLA1.5	SLA1.4	SLA1.3	SLA1.2	SLA1.1	SLA1.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- SLA[7:0] : I²C Slave Address Register.

In 7-bit address mode and in 10-bit address mode (1st SLA),
I2C_SLA[7:1] is used for matching address and I2C_SLA[0] is masked.
In 10-bit address mode (2nd SLA),
I2C_SLA[7:0] is used for matching address.

✓ **I2CDAT0** (B6h) : I²C Address / Data Register

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

✓ **I2CCFG0** (B4h) : I²C Configuration Register

-	-	-	-	MSSEL	ADSEL	SP_IE	GCE
-	-	-	-	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- MSSEL : I2C Master/Slave Mode Selection
  [0] : Slave mode [1] : Master mode
- ADSEL : 7-bit / 10-bit Address Mode Selection in Slave mode
  [0] : 7-bit mode [1] : 10-bit mode
- SP_IE : Start/Stop Interrupt Enable
  [0] : Start/Stop Interrupt Disable [1] : Start/Stop Interrupt Enable
- GCE : General Call Enable in Slave mode
  [1] : Respond to the general call address (0x00)

✓ **I2CI2C0_SCL** (B7h) : I²C SCL Clock Scaler

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

- MSCL[7:0] : Frequency scaler of I²C Master

$F_{I2C} = F_{OSC} / (2 * (MSCL[7:0] + 2))$

6.12. I2C : SFR (Cont'd)

✓ **I2CCON0** (B3h) : I²C Control Register

-	SLA2ME	SCLHD	LASTB	PGEN	SGEN	I2CIOEN	I2CEN
-	R/W(0)	R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

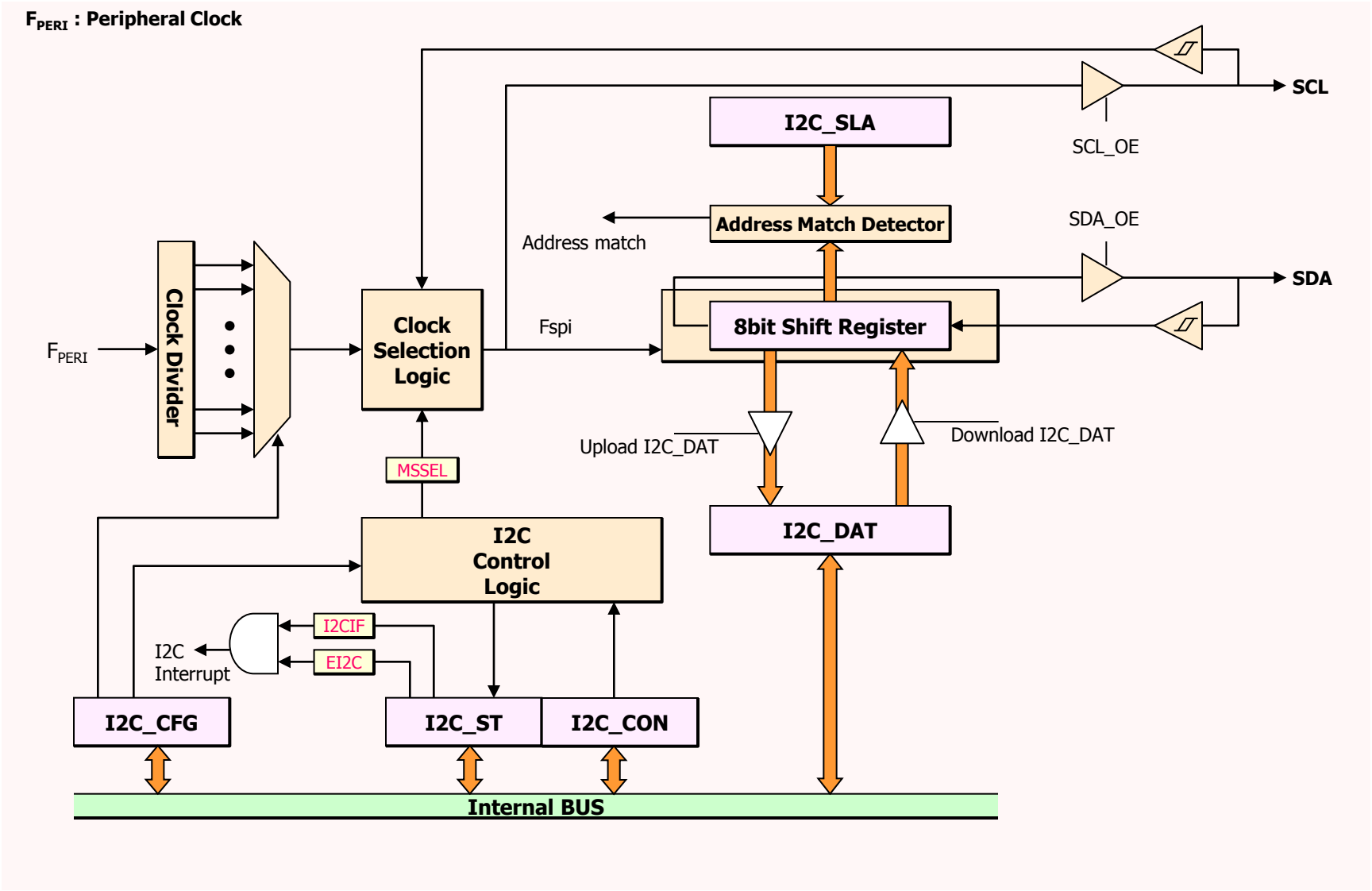
- SLA2ME : 2nd Byte Slave Address Match Enable in Slave mode
  [0] : 2nd Byte SLA Match Disable [1] : 2nd SLA Byte Match Enable
- SCLHD : Hold SCL 'low' for Wait State in Slave mode.
  [0] : Hold SCL 'low'. The flag is cleared automatically by H/W
  [1] : Release SCL 'float'. The flag is set by S/W
- LASTB : Indicate last byte in Master Receiver mode.
  [0] : Send Acknowledge after last byte
  [1] : Send Not Acknowledge after last byte
  In Master Receiver mode, before receiving last byte, the flag must be set.
- PGEN : Generate Stop bit.
  [0] : Start or Idle state. [1] : Generate Stop bit.
  The flag is cleared automatically after Stop bit in Master mode and when I2CEN is cleared.
- SGEN : Generate Start bit
  [0] : Stop or Idle state [1] : Generate Start bit
  If the bus is not free, it waits for Stop bit condition.
  The flag is cleared automatically after Start bit in Master mode and when I2CEN is cleared.
- I2CIOEN : Enable I2C IO
  [0] : Disable I2C IO [1] : Enable I2C IO
- I2CEN : Enable I2C module
  [0] : Disable I2C module [1] : Enable I2C module

✓ **I2CST0** (B0h) : I²C Status Register

I2CIF	I2COF	I2CACK	I2CRW	I2CDA	I2CP	I2CS	I2CBF
R/W(0)	R/W(0)	R (0)	R (0)	R (0)	R (0)	R (0)	R (0)

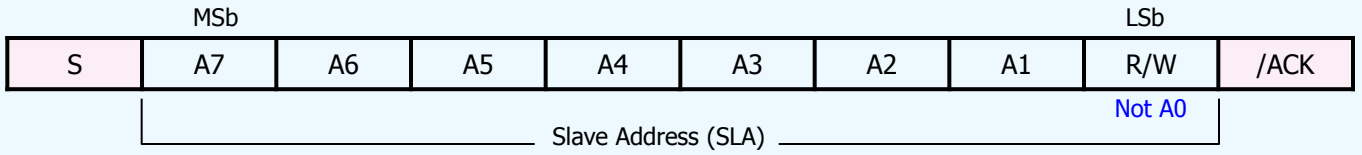
- I2CIF : I²C Master Interrupt Flag in slave & master mode.
  [0] : Idle [1] : Interrupt occurred.
  It is set each time a byte is received or transmitted.
  If SP_IE flag in I2C_CFG SFR is set, it is set at Start/Stop condition.
  The flag is set by H/W and cleared by S/W.
- I2COF : I2C Overflow Flag in slave & master mode
  [0] : Idle [1] : Overflow occurred.
  It is set when a byte is received while I2C_BUF SFR is still holding the previous byte.
  It is set by H/W and cleared by S/W
- I2CACK : I2C Acknowledge flag in slave & master mode.
  [0] : Indicate receiving Acknowledge bit.
  [1] : Indicate receiving Not Acknowledge bit.
- I2CRW : I2C Read/Write flag in slave mode
  [0] : Write state [1] : Read state
- I2CDA : Data / Address flag in slave mode
  [0] : Indicates the last byte received or transmitted was Data
  [1] : Indicates the last byte received or transmitted was Address
- I2CP : Stop flag in slave & master mode
  [0] : Indicates Stop bit was not detected.
  [1] : Indicates Stop bit was detected.
  This flag is cleared when I2CS is set or I2CEN is cleared.
- I2CS : Start flag in slave & master mode
  [0] : Indicates Start bit was not detected.
  [1] : Indicates Start bit was detected.
  This flag is cleared when I2CP is set or I2CEN is cleared.
- I2CBF : Busy flag in slave & master mode
  [0] : RX not complete (Receiver), TX not complete (Transmitter)
  [1] : RX complete (Receiver), TX complete (Transmitter)

6.12. I2C : Block Diagram

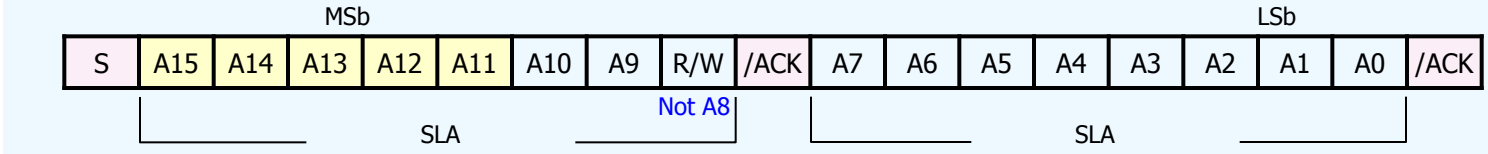


6.12. I2C : Overview

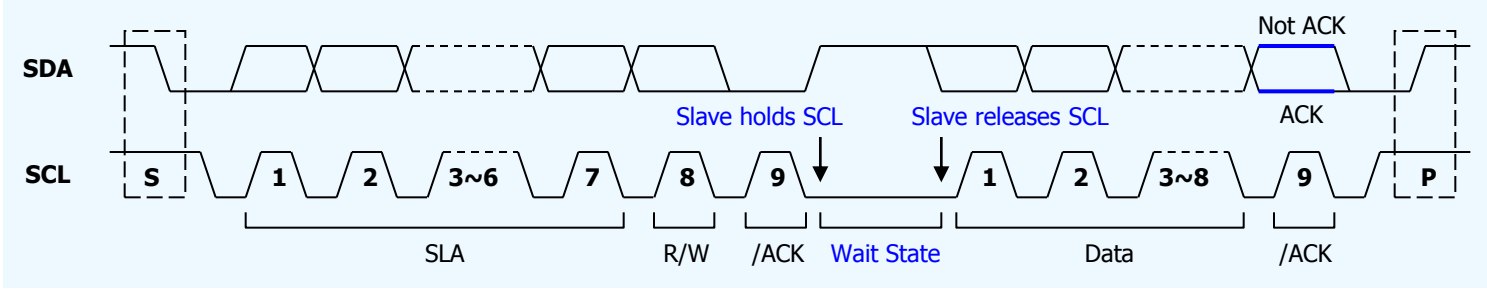
- Addressing I2C devices
  - ✓ 7-bit Address Format



  - ✓ 10-bit / Extended 15-bit Address Format



- Transfer Acknowledge
  - ✓ Slave-Receiver generates an acknowledge bit after Master transfers each byte. If not, Master aborts the transfer.
  - ✓ Master-Receiver generates an acknowledge bit after Slave transfers each byte except last byte.
  - ✓ Transfer Wait State
    1) If Slave needs to delay the transmission of the next byte, it can hold the SCL 'low'
    2) Master must enter the wait state, if the SCL is held 'low'.
    3) When Slave releases the SCL, Master starts the transfer again.

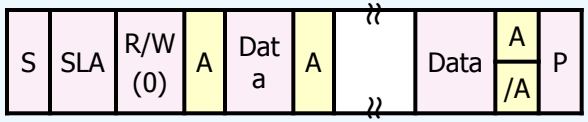


6.12. I2C : Overview

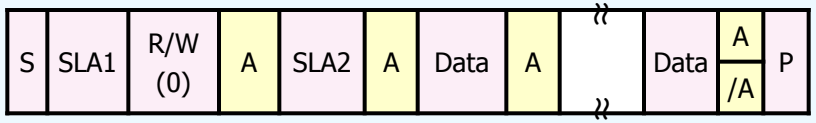
Master-Transmitter Sequence

From Master to Slave | From Slave to Master

[7-bit Address Mode]

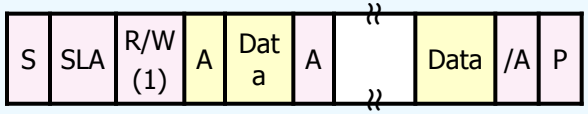


[10-bit / Extended 15-bit Address Mode]



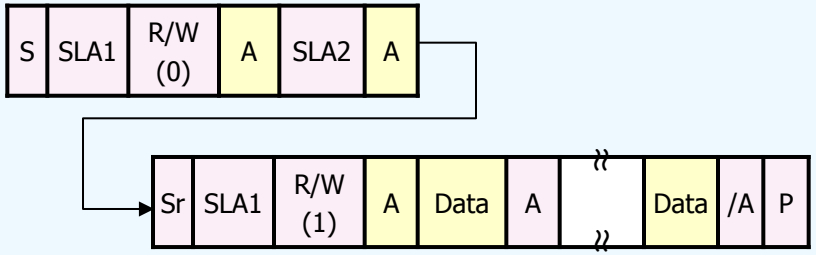
Master-Receiver Sequence

[7-bit Address Mode]



- A : Acknowledge
- /A : Nor Acknowledge
- S : Start
- Sr : Repeated Start
- P : Stop
- SLA : Slave Address

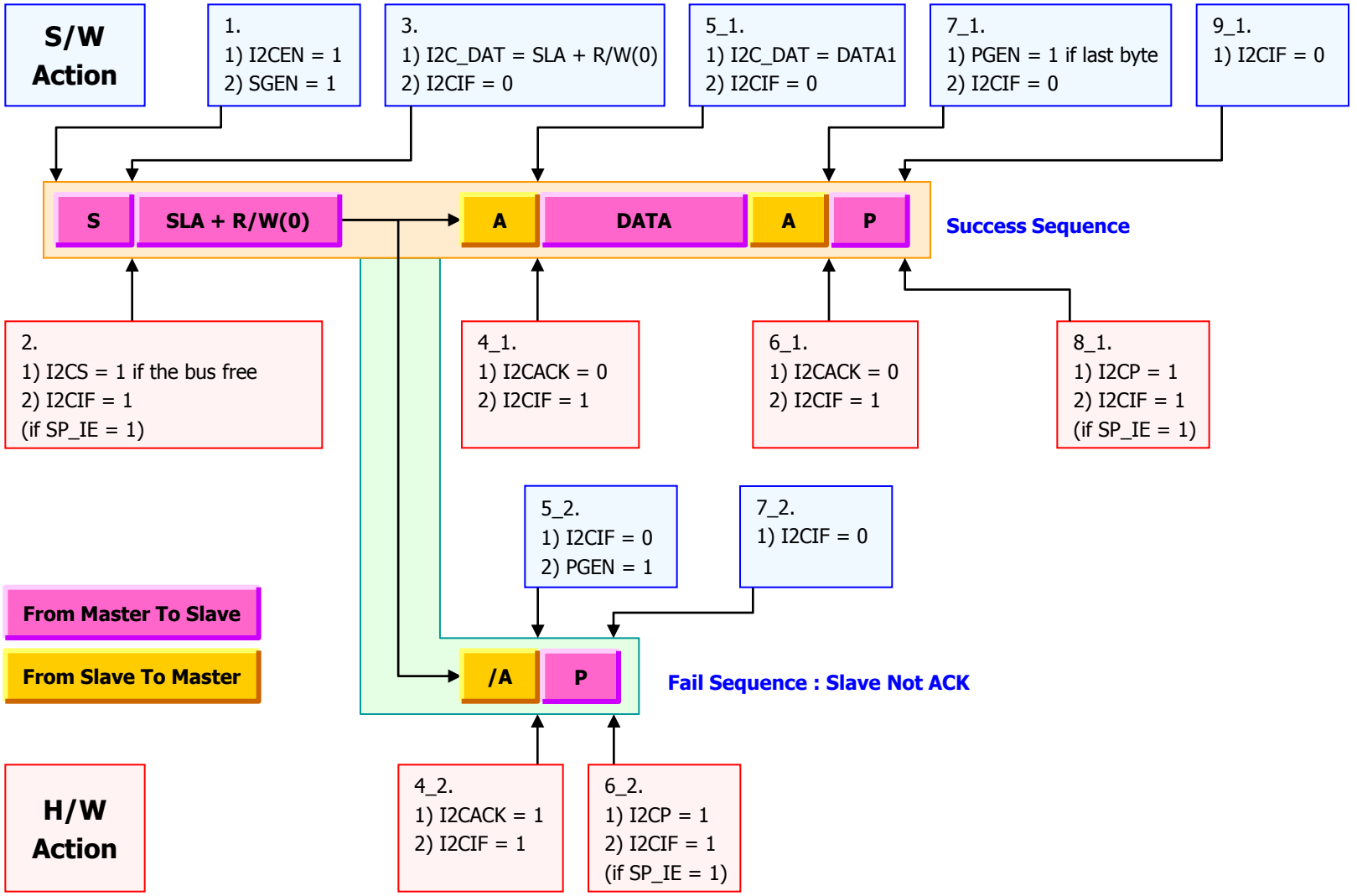
[10-bit / Extended 15-bit Address Mode]



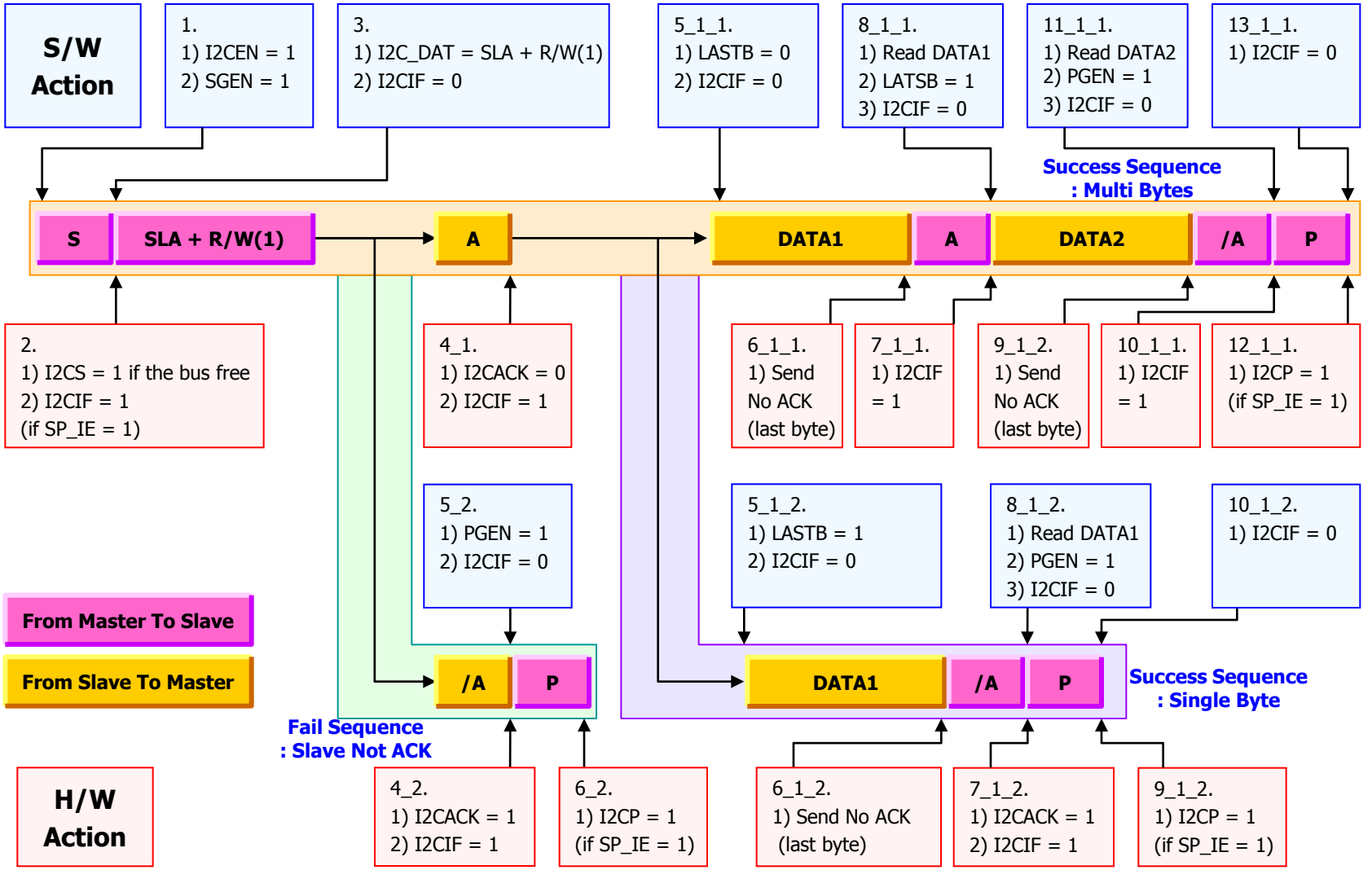
Combined Format

- ✓ When Master does not want to release the bus, a repeated start condition must be generated without a stop condition.
- ✓ The condition is identical to a start condition
- ✓ The condition must occurs after a data transfer acknowledge pulse.

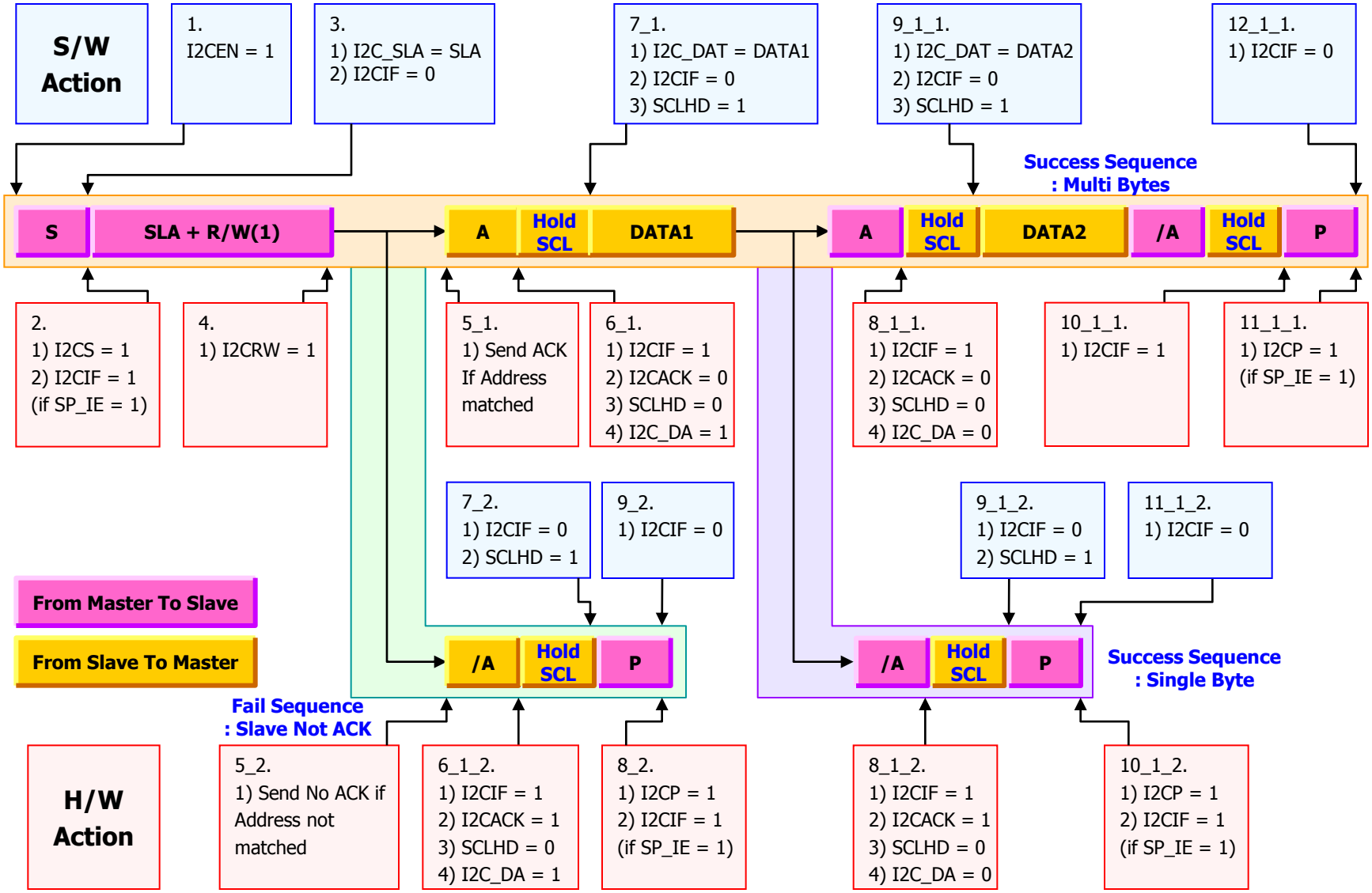
6.12. I2C : Master Transmitter Flow



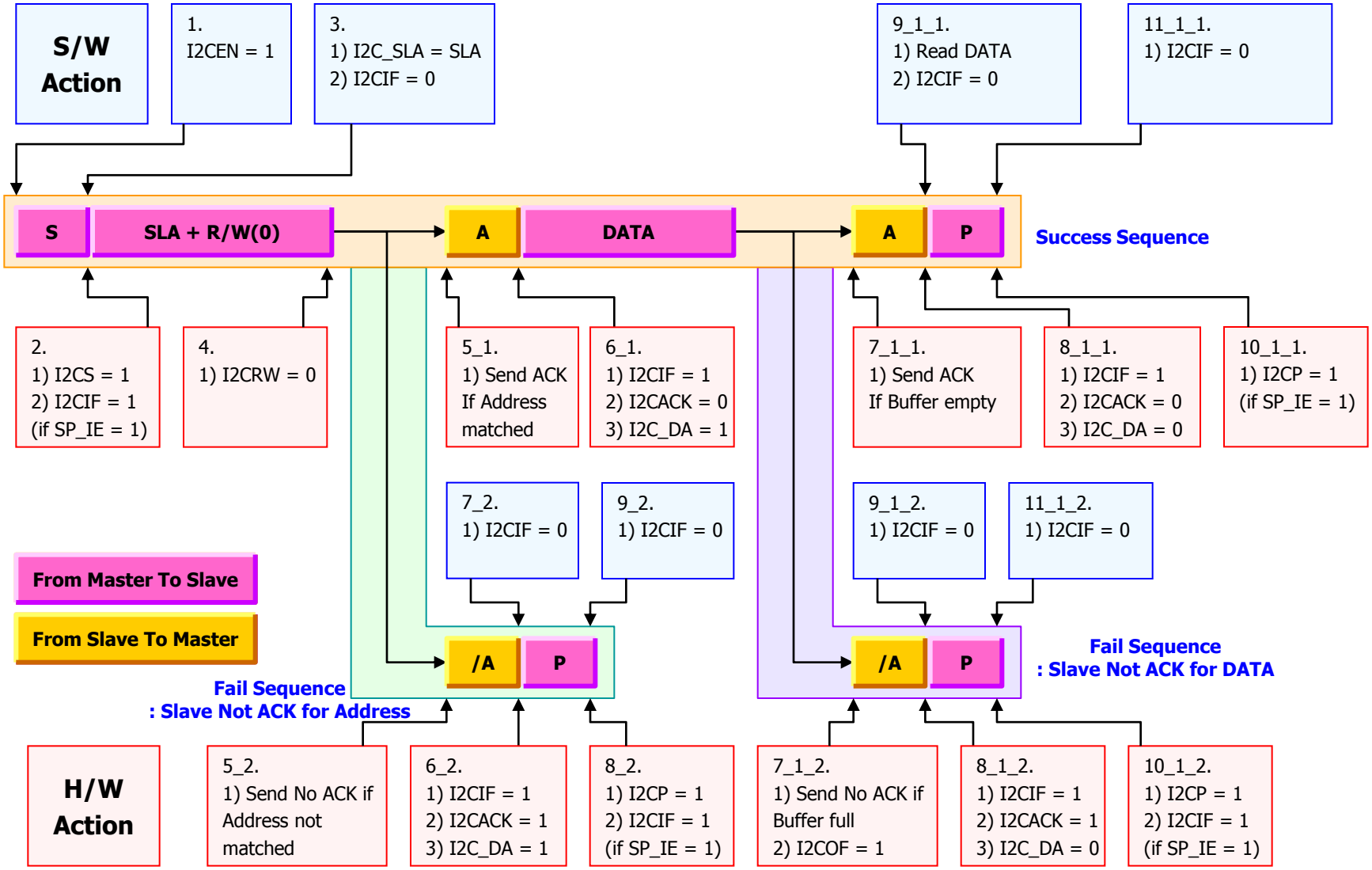
6.12. I2C : Master Receiver Flow



6.12. I2C : Slave Transmitter Flow



6.12. I2C : Slave Receiver Flow



6.12. I2C : Master Example

- I2C0 Master example code

```
I2CST    EQU  0B0H          ; I2CST SFR
I2CIF    EQU  0B7H          ; I2CST.7 Flag
I2COF    EQU  0B6H          ; I2CST.6 Flag
I2CACK   EQU  0B5H          ; I2CST.5 Flag
I2CRW    EQU  0B4H          ; I2CST.4 Flag
I2CDA    EQU  0B3H          ; I2CST.3 Flag
I2CP     EQU  0B2H          ; I2CST.2 Flag
I2CS     EQU  0B1H          ; I2CST.1 Flag
I2CBF    EQU  0B0H          ; I2CST.0 Flag

I2CCON   EQU  0B1H
I2CCFG   EQU  0B2H
I2CSLA   EQU  0B3H
I2CDAT   EQU  0B4H
I2CSCL   EQU  0B5H

ORG 000h
  LJMP   START

ORG 0100h
START:
  ORL    I2CCFG,   #08h      ; master mode
  ORL    I2CCFG,   #04h      ; 10bit address mode
  ANL    I2CCFG,   #0FDh     ; Start/Stop interrupt disable
  MOV    I2CSCL,   #10h      ; clock scaling
  ORL    I2CCON,   #02h      ; I2C IO enable
  ORL    I2CCON,   #01h      ; I2C enable
  MOV    I2CDAT,   #01h      ; TX : SLA1 = 10h, write mode
  ORL    I2CCON,   #04h      ; Send Start bit

WAIT_SLA1:                   ; Wait for end of SLA1 TX
  JNB    I2CIF,    WAIT_SLA1
  JB     I2CACK,   ACK_FAIL  ; check ack fail or not

  MOV    I2CDAT,   #50h      ; TX : SLA2 = 50h
  CLR    I2CIF               ; clear after setting I2CDAT

WAIT_SLA2:                   ; Wait for end of SLA2 TX
  JNB    I2CIF,    WAIT_SLA2
  JB     I2CACK,   ACK_FAIL  ; check ack fail or not

  MOV    I2CDAT,   #38h      ; TX : Data = 38h

WAIT_TXD:                    ; Wait for end of Data TX
  JNB    I2CIF,    WAIT_TXD
  JB     I2CACK,   ACK_FAIL  ; check ack fail or not
  ORL    I2CCON,   #08h      ; Send Stop bit
  CLR    I2CIF               ; clear after setting STOP

WAIT_STOP:
  MOV    A,        I2CCON    ; Wait end of STOP
```

6.12. I2C : Slave Example

- I2C1 Slave example code using interrupt

```
I2CST     EQU  0B0H          ; I2CST SFR
I2CIF     EQU  0B7H          ; I2CST.7 Flag
I2COF     EQU  0B6H          ; I2CST.6 Flag
I2CACK    EQU  0B5H          ; I2CST.5 Flag
I2CRW     EQU  0B4H          ; I2CST.4 Flag
I2CDA     EQU  0B3H          ; I2CST.3 Flag
I2CP      EQU  0B2H          ; I2CST.2 Flag
I2CS      EQU  0B1H          ; I2CST.1 Flag
I2CBF     EQU  0B0H          ; I2CST.0 Flag

I2CCON    EQU  0B1H
I2CCFG    EQU  0B2H
I2CSLA    EQU  0B3H
I2CDAT    EQU  0B4H
I2CSCL    EQU  0B5H

ORG 000h
  LJMP    START

ORG 06Bh
  LJMP    I2CS_ISR           ; JMP I2C interrupt routine

ORG 0100h
START:
  ANL     I2CCFG,   #0F7h    ; slave mode
  ORL     I2CCFG,   #04h     ; 10bit address mode
  ORL     I2CCFG,   #02h     ; Start/Stop interrupt enable
  MOV     I2CSLA,   #80h     ; 1st Slave address
  ANL     I2CCON,   #0BFh    ; 2nd Slave address not compare
  ORL     I2CCON,   #02h     ; I2C IO enable
  ORL     I2CCON,   #01h     ; I2C enable
  SETB    EIE.3              ; I2C interrupt enable
  SETB    IE.7               ; All interrupt enable
I2C_RX:
  JNB     I2CS,     .
  JNB     I2CP,     .
  SJMP    I2C_RX
```

```
I2CS_ISR:                    ;---- I2C Slave interrupt routine ----
  MOV     OSCICN,   #04h     ; clock speed-up
  CLR     EIE.3              ; I2C interrupt disable
  CLR     I2CIF              ; clear interrupt flag (START bit)

WAIT_BYTE:                   ;------- Wait Event -------
  JB      I2CP,     END_ISR  ; check STOP bit
  JNB     I2CIF,    WAIT_BYTE ; if I2CIF is set, go next process
  CLR     I2CIF              ; clear interrupt flag
  MOV     R1,       SLA2BUF  ; save 2nd SLA to R1
  JB      I2CDA,    SLA1_RX  ; check address or data field
  JB      I2CRW,    S_TX     ; check RX or TX operation
S_RX:                        ;------- RX operation -------
  MOV     @R1,      I2CDAT   ; save I2CDAT(RX data) to R1
  INC     SLA2BUF            ; increment 2nd SLA for burst
  SJMP    WAIT_BYTE          ; repeat loop
S_TX:                        ;------- TX operation -------
  JB      I2CACK,   END_TX   ; if no ack, finish TX
  MOV     I2CDAT,   @R1      ; TX data
END_TX:
  ORL     I2CCON,   #20h     ; release SCL hold from "low"
  INC     SLA2BUF            ; increment 2nd SLA for burst
  SJMP    WAIT_LOOP          ; repeat loop
SLA1_RX:                     ;------- SLA1 operation -------
  JB      I2CBF,    SLA2_RX  ; if I2CBF is set, RX 2nd SLA
  JB      I2CRW,    S_TX     ; if I2CRW is set, TX data
  SJMP    WAIT_LOOP          ; repeat loop
SLA2_RX:                     ;------- SLA2 operation -------
  MOV     SLA2PTR,  I2CDTA   ; save 2nd SLA to SLA2BUF
  SJMP    WAIT_LOOP          ; repeat loop

END_ISR:                     ;------- end of I2C Slave -------
  CLR     I2CIF              ; clear interrupt flag (STOP bit)
  SETB    EIE.5              ; enable I2C interrupt
  MOV     OSCICN,   #0Fh     ; restore clock speed
  RETI
```

6.13. Interrupt : 11 Sources / 2-level Priority

- 11 Interrupt Sources
  - ✓ Timer 0/1, UART, ADC, WDT, I²C, PWM, 4 External
- 2-level Interrupt Priority

[Interrupt Vector Address]

HIGH

PRIORITY

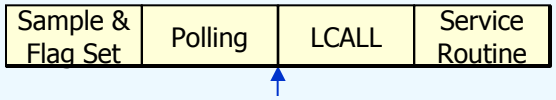
LOW

Interrupt Sources	Address	Priority Level	
LVD	0033h	Highest	NMI
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	
INT3	004Bh	2 Levels	
I2C0	0053h	2 Levels	
-	005Bh	2 Levels	
WDT	0063h	2 Levels	
PWM	006Bh	2 Levels	

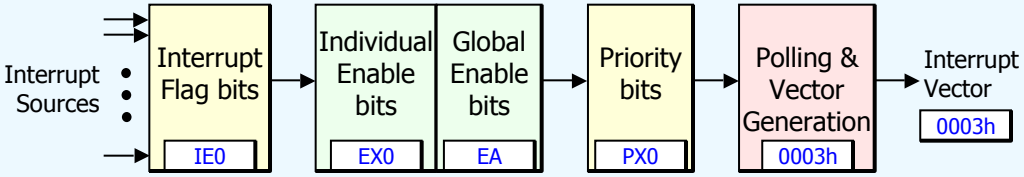
*** Interrupt related to SFR (refer to Appendix B : SFR Description)**

✓ **TCON**	(88h)	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
✓ **EXIF**	(91h)	BGSB	RTRG	IE3	IE2	XT/RG	RGMD	RGSL	BGS
✓ **SCON**	(98h)	-	-	-	REN	-	-	TI	RI
✓ **IE**	(A8h)	EA	EADC	-	ES	ET1	EX1	ET0	EX0
✓ **IP**	(B8h)	-	PADC	-	PS	PT1	PX1	PT0	PX0
✓ **EIE**	(E8h)	-	-	EPWM	EWDT	-	EI2C0	EX3	EX2
✓ **EIP**	(F8h)	-	-	PPWM	PWDT	PI2C1	PI2C0	PX3	PX2
✓ **WDCON**	(D8h)	WD1	WD0	EPFI	PFI	WDIF	WTRF	EWT	RWT
✓ **PWMIF**	(DDh)	-	-	-	-	-	-	-	PWMIF
✓ **I2CST0**	(B0h)	I2CIF	I2COF	I2CACK	I2CRW	I2CDA	I2CP	I2CS	I2CBF
✓ **ADCON**	(EFh)	AD_EN	AD_REQ	AD_END	ADCF	-	ADIV	SAR1	SAR0

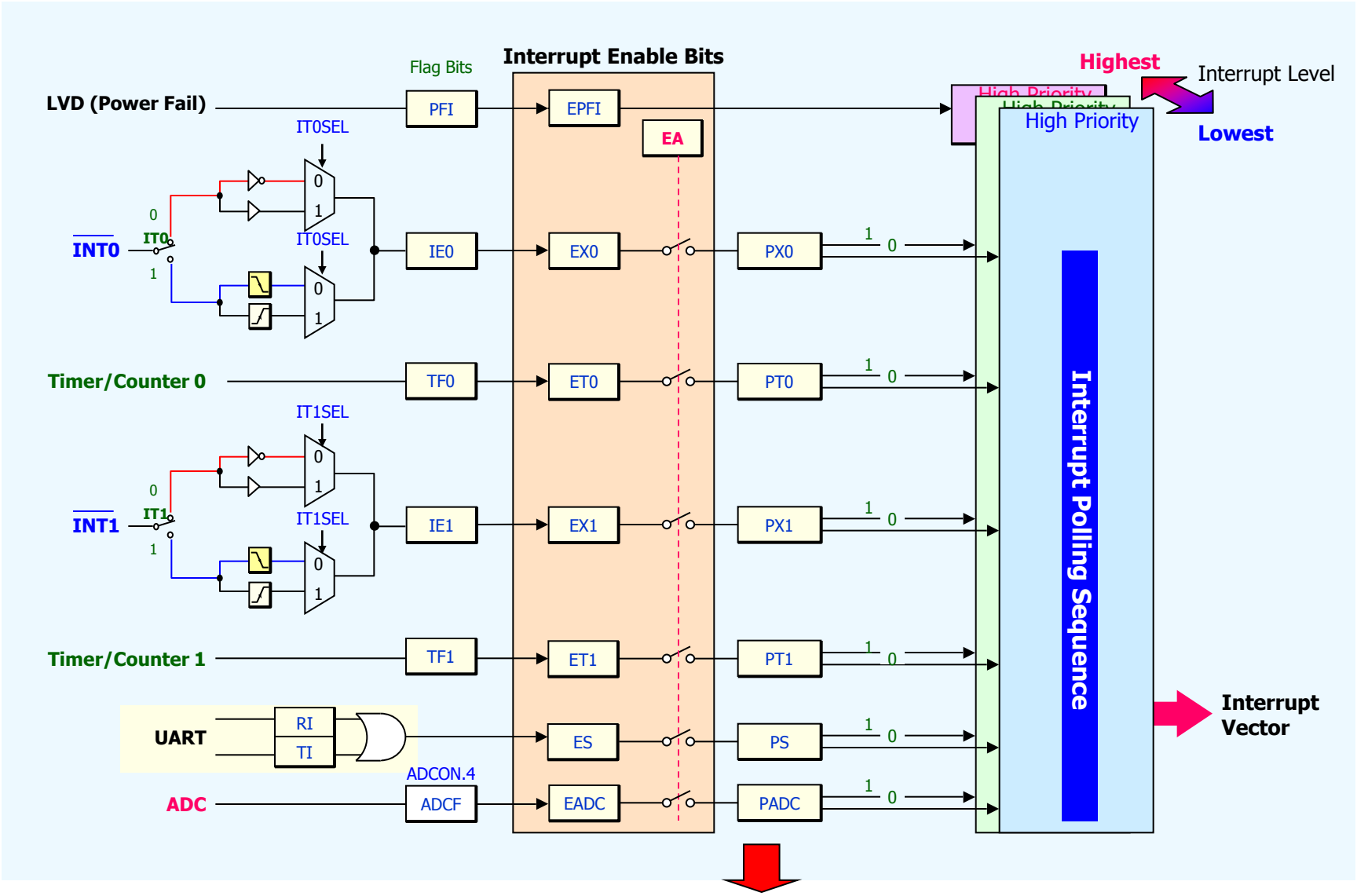
[Response Sequence]



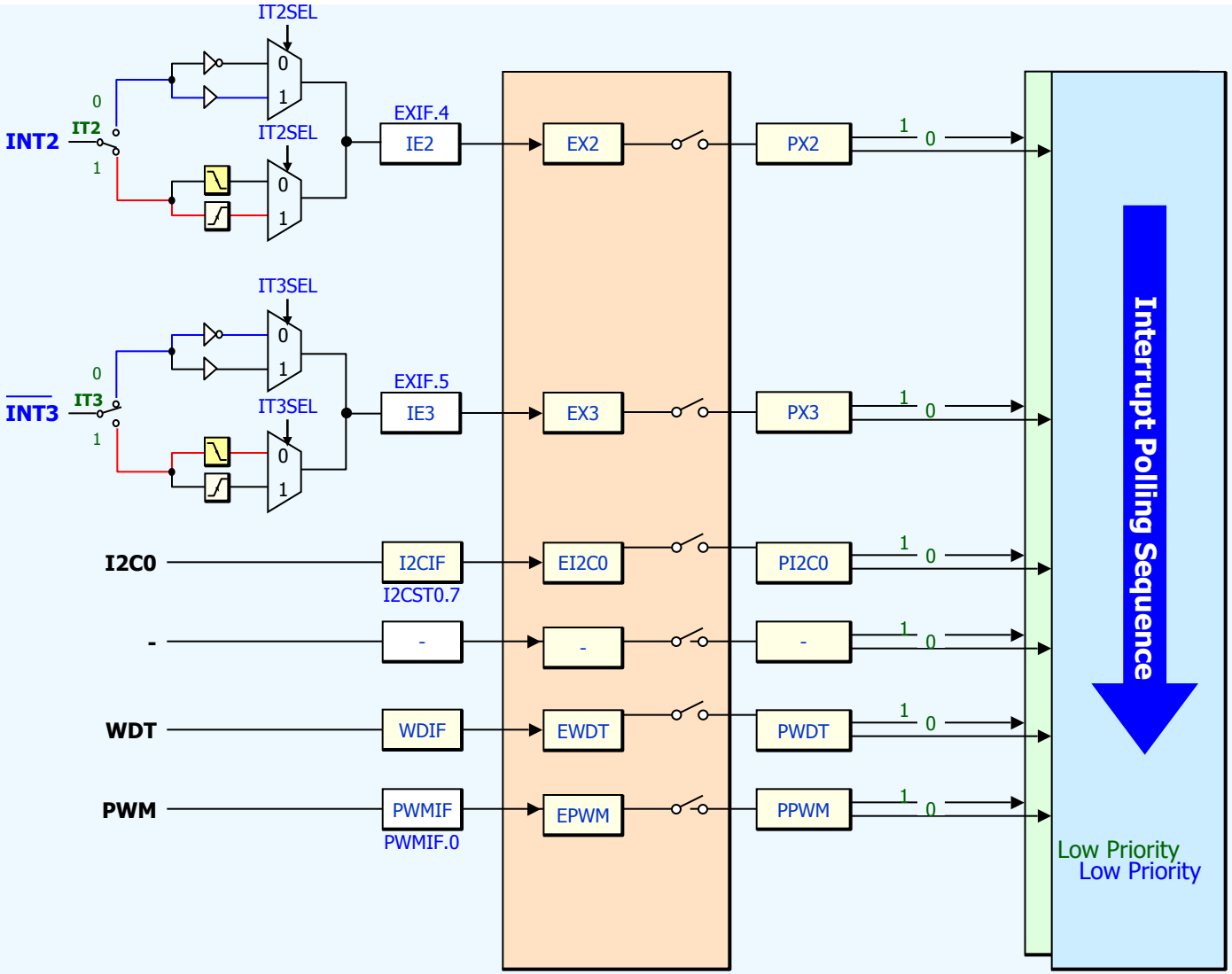
[Interrupt Vector Generation Flow]



6.13. Interrupt Functional Description



6.13. Interrupt Functional Description (Cont'd)



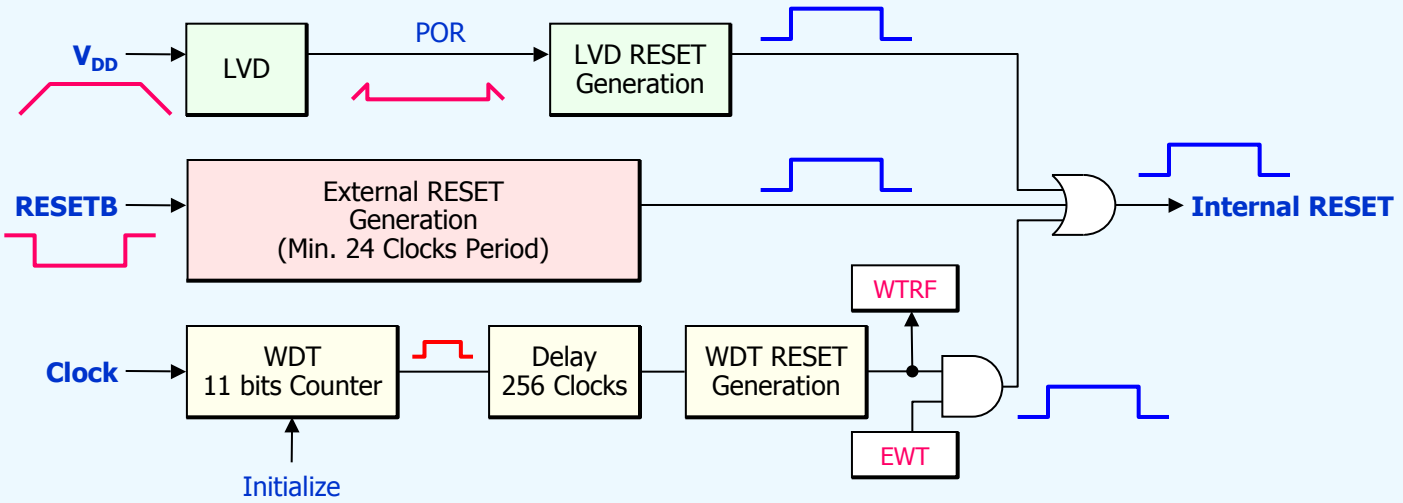
6.14. Reset Circuit : Three Reset Sources

- 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.
- WDT Reset : Enabled or disabled by S/W

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

WD1	WD0	EPFI	PFI	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

- 4 System Clock Sources : Ring OSC, Ring OSC(32KHz). 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		
RT/RG	XT/RG	RTOFF	XTOFF	RING2ON	RINGON	RGSL		RGMD	XTUP	RTUP
x	1	x	0	x	X	X	Crystal OSC.	0	1	0/1
1	0	x	X	x	1	X	Ring OSC.	1	0/1	0/1
0	0	x	x	1	X	x	Ring OSC(32KHz).	0	1	0/1
x	1	x	0	x	x	0	Crystal OSC. (during Power-down Wake-up)	0	0	0/1
1	0	x	x	x	1	1	Ring OSC. (during Power-down Wake-up)	1	0	0/1
0	0	x	x	1	X	1	Ring OSC(32KHz). (during Power-down Wake-up)	0	0	0

6.15. Clock Circuit : SFR

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

-	RTRG	IE3	IE2	XT/RG	RGMD	RGSL	BGS
-	R/W(1)	R/W(0)	R/W(0)	R/W(0)	R(1)	R/W(0)	R/W(1)

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

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

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

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

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

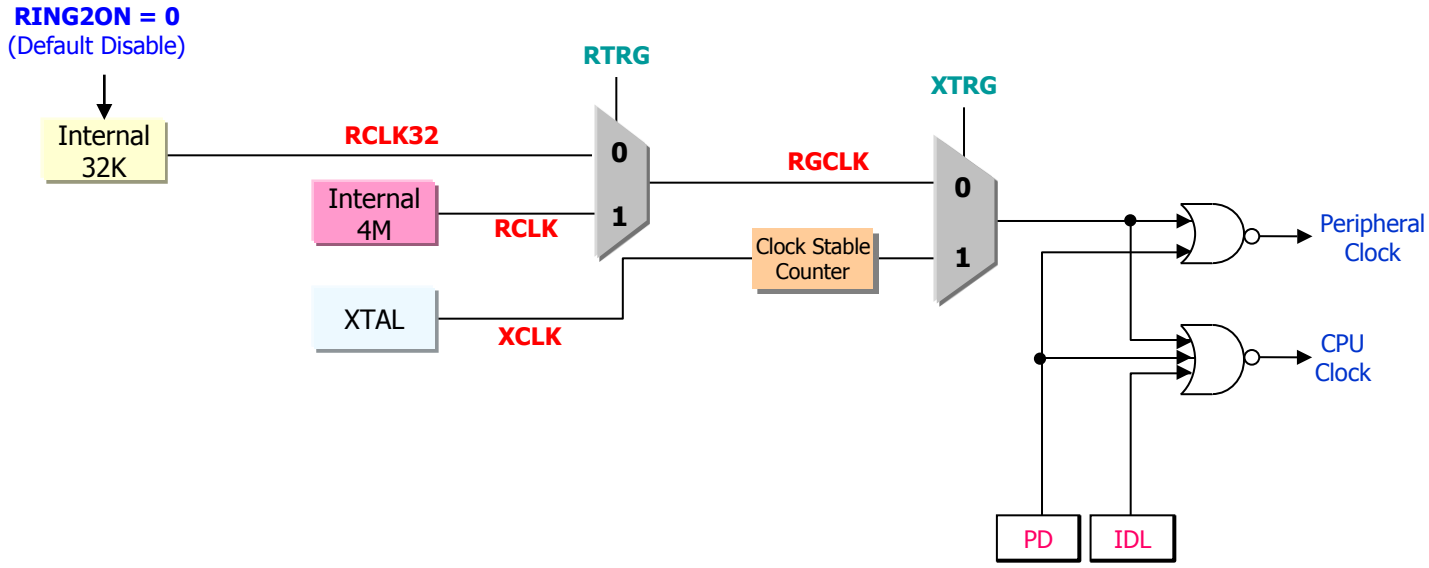
-	-	-	-	XTOFF	-	-	-
				R/W(0)			

✓ **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)

✓ **OSC2ICN** (BFh) : Internal Ring2 Oscillator Control Register

-	-	-	-	-	-	OSC_REF	RING2ON
						R/W(0)	R/W(1)



6.15. Clock Circuit : RING OSC. Calibration

- Internal RING Oscillator Calibration
  - ✓ RING Calibration value is saved at 0x103F address area.
  - ✓ The calibration value is set to fit the internal RING frequency to 4.00MHz.
  - ✓ User must move the calibration value into RINGCON SFR for using 4.00MHz RING OSC.

[Example Code : Update RING calibration value]

```
MOV  DPTR, #0x103F
CLR  A
MOVC A, @A+DPTR
MOV  RINGCON, A
```

```
unsigned char code ringcon_cal _at_ 0x103F;

void main (void) {
   RINGCON = ringcon_cal;
   …
   …
}
```

■ **RINGCON** (95h) : Ring Control Configuration Register

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

- RINGCON[7:0] : Internal Ring OSC. can be tuned.

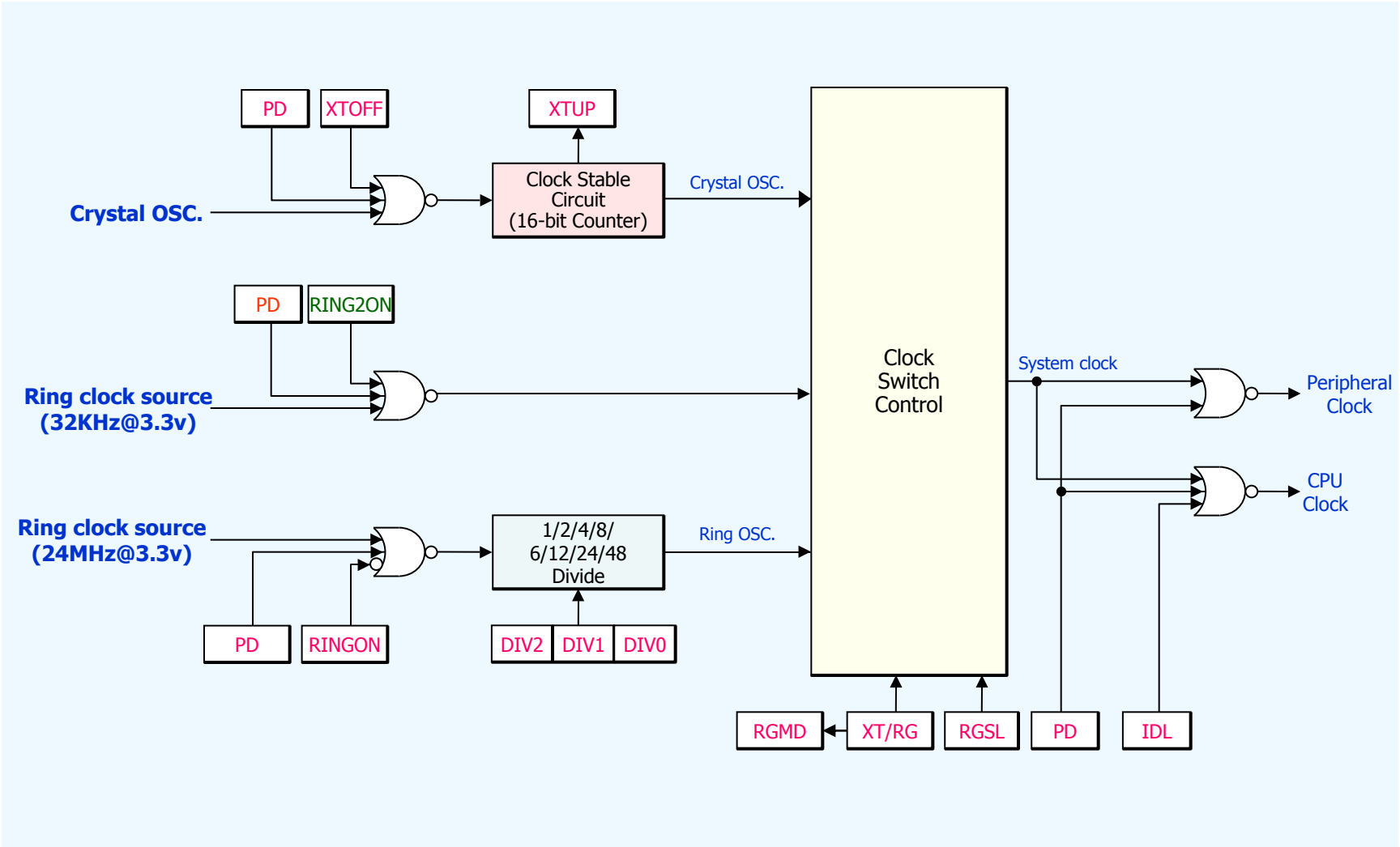
■ **OSCICN** (BEh) : Internal Ring Oscillator Control Register

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

- RINGON : 1 = Internal ring oscillator(24MHz) is running.
  0 = Internal ring oscillator is killed.
  Don't clear RINGON bit when XTRG = 0.

- DIV2, DIV1, DIV0 : Ring 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

6.15. Clock Circuit : Block Diagram



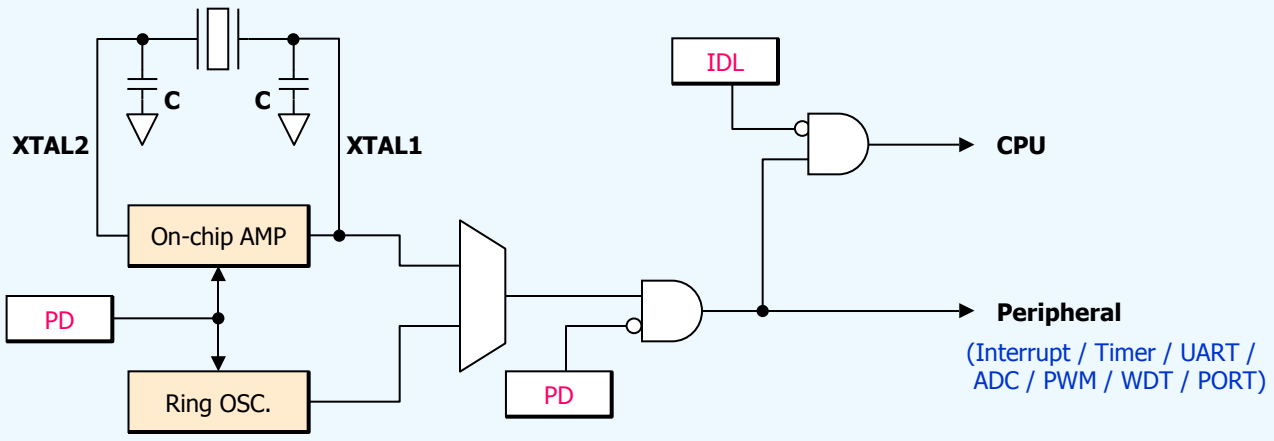
6.16. Power Management : 3 Modes

- **Active Mode** : CPU and Peripheral are running.
- **Idle Mode** : Only Peripheral is 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 from all kinds of resets. (ex: RESETB, LVD. etc)
    → 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. (ex: RESETB, LVD, .etc)
    → MCU restarts.

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

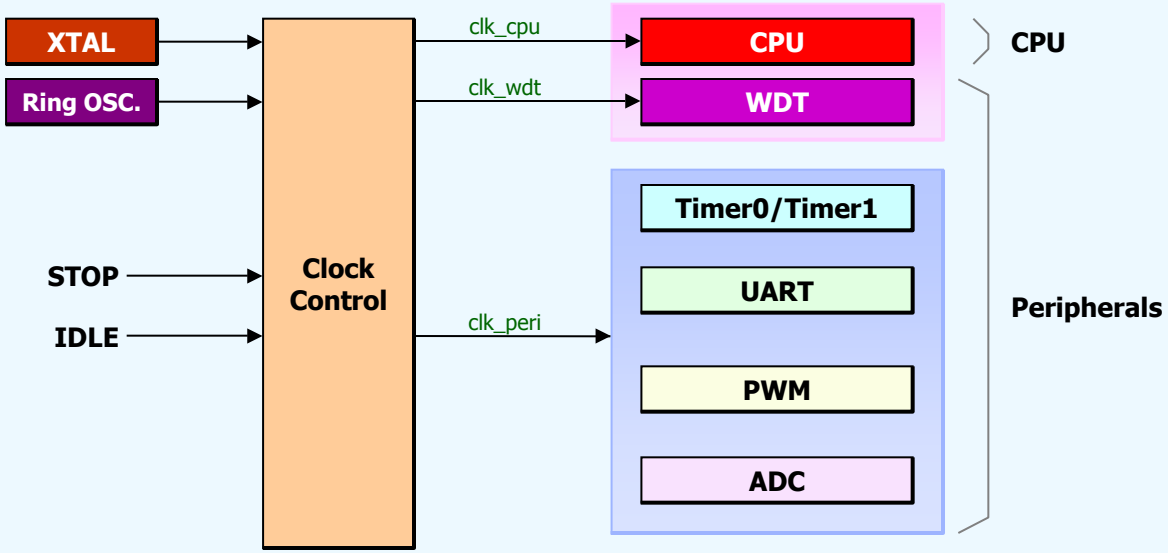
SMOD1	-	-	POF	GF1	GF0	PD	IDL
R/W(0)			R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

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



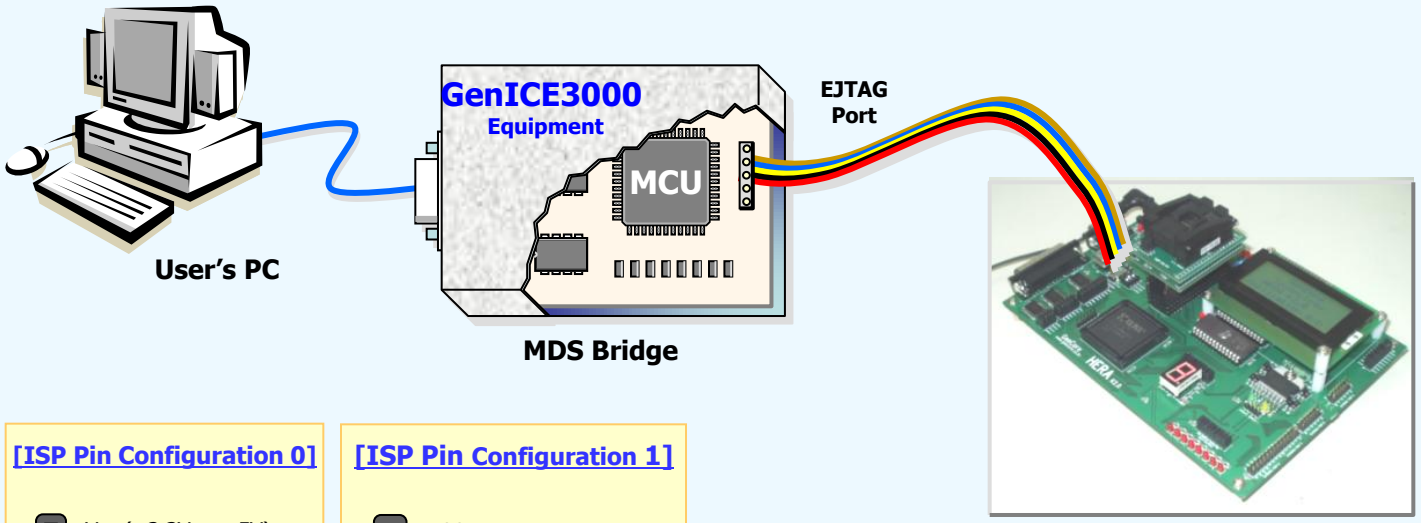
6.16. Power Management : Clock Circuit

- When MCU is in IDLE mode, CPU will stop and all the peripherals will run as below.
  - ✓ WDT, Timer, UART, PWM, & ADC
- When MCU is in STOP mode, CPU & all the peripherals will stop.



6.17. ISP & Debugging

- Code memory (1kBytes) can be programmed using EJTAG in target system.
  - ✓ FLASH : 0x0000 ~ 0x03FF (1,024 Bytes)
- EEPROM (128Bytes) can be programmed using EJTAG in target system.
  - ✓ EEPROM : 0x037F ~ 0x03FF (128 Bytes)
- Debugging using GENICE



[ISP Pin Configuration 0]

- V_{DD} (+3.3V or +5V)
- V_{SS} (GND)
- I2C0_SCL (P0.5)
- RESETB (P1.2)
- I2C0_SDA (P0.4)

[ISP Pin Configuration 1]

- VCC
- GND
- I2C1_SCL (P0.1)
- RESETB (P1.2)
- I2C1_SDA (P0.2)

- GenICE3000 System supports both of debugger and ROM Downloader

6.17. 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.

6.18. IAP (In Application Programming)

- Code memory(7KB) & EEPROM(1KB) can be programmed during the operation of MCU.
- Program time : approximately 3.0 ms
- Program unit : 1 Byte
- IAP SFR

✓ **EEAEN** (FFh) : IAP Routine Access Enable Register

-	-	-	-	-	-	-	EAEN
							R/W(0)

- EAEN : IAP Routine Access Enable

✓ **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)

✓ **EECNTLD** (F9h) : EEPROM Erase/Program Time Count Loading

EECNTLD	-	-	-	-	-	-	-
R/W(0)							

- EECNTLD : EEPROM Erase/Program Time Count Loading
  Set by S/W, cleared by H/W automatically.

✓ **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)

✓ **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)

✓ **EECNTH** (FCh) : EEPROM Erase/Program Time Count High

EECNT.23	EECNT.22	EECNT.21	EECNT.20	EECNT.19	EECNT.18	EECNT.17	EECNT.16
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

✓ **EECNTM** (FBh) : EEPROM Erase/Program Time Count Middle

EECNT.15	EECNT.14	EECNT.13	EECNT.12	EECNT.11	EECNT.10	EECNT.9	EECNT.8
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

✓ **EECNTL** (FAh) : EEPROM Erase/Program Time Count Low

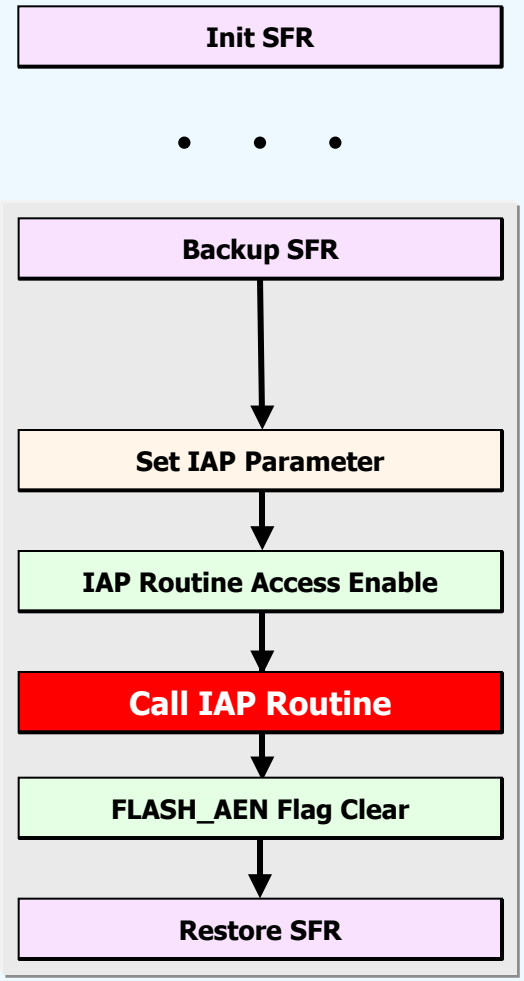
EECNT.7	EECNT.6	EECNT.5	EECNT.4	EECNT.3	EECNT.2	EECNT.1	EECNT.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

6.18. IAP : Function Set

- IAP call function
  - ✓ iap_eeprom_program : call address (FF3Eh)
  - ✓ iap_eeprom_erase : call address (FF29h)

- Before calling IAP function, any interrupt must be disabled.
- Before calling IAP function, EAEN flag in EEAEN SFR must be set.
  - ✓ **Only use ORL/ANL assembly instruction to set or reset EAEN flag.**
- After executing IAP function, the value of PSW SFR can be changed.
- Any interrupt service routine will not be executed timely since the CPU is suspended for tens of milliseconds during executing an IAP function (Program/Erase).

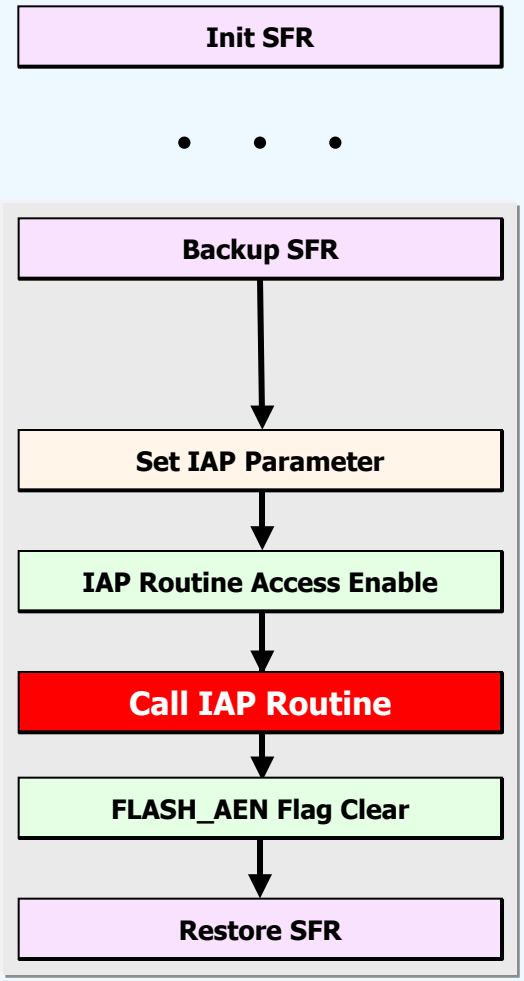
6.18. IAP : Program Flow



[Example Code : IAP Program for EEPROM]

```
ORL EEAEN, #01h          ; IAP routine access enable
MOV EECNTH, #00h         ; Program/Erase Time Count High
MOV EECNTM, #27h         ; Program/Erase Time Count Mid
MOV EECNTL, #10h         ; Program/Erase Time Count Low
ANL EEAEN, #0FEh         ; IAP routine access disable
                         ; 0x001027 @ RING Freq. == 4MHz

PUSH ACC                 ; backup acc
PUSH DPL                 ; backup dptr
PUSH DPH
MOV  R1, IE              ; backup IE SFR
CLR  IE.7                ; Interrupt disable

MOV DPTR, #ADDR          ; Programming Address
MOV A, #DATA             ; Programming Data

ORL EEAEN, #01h          ; IAP routine access enable
MOV EECNTLD, #80h        ; Program/Erase Time Count Loading

CALL iap_eeprom_program ; Call IAP routine

ANL EEAEN, #0FEh         ; IAP routine access disable

MOV IE, r1               ; restore IE SFR
POP DPH                  ; restore acc, dptr
POP DPL
POP ACC
```

6.18. IAP : Erase Flow

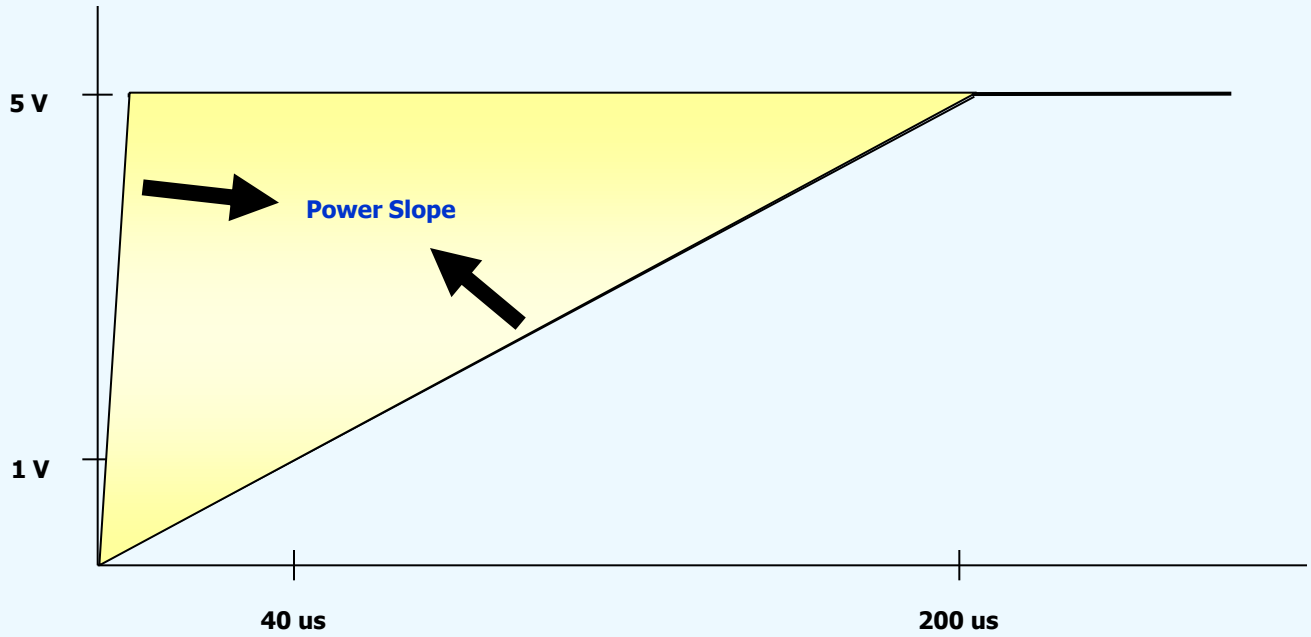


[Example Code : IAP Program for EEPROM]

```
ORL EEAEN, #01h          ; IAP routine access enable
MOV EECNTH, #00h         ; Program/Erase Time Count High
MOV EECNTM, #27h         ; Program/Erase Time Count Mid
MOV EECNTL, #10h         ; Program/Erase Time Count Low
ANL EEAEN, #0FEh         ; IAP routine access disable
                         ; 0x001027 @ RING Freq. == 4MHz

PUSH ACC                 ; backup acc
PUSH DPL                 ; backup dptr
PUSH DPH
MOV  R1, IE              ; backup IE SFR
CLR  IE.7                ; Interrupt disable

MOV DPTR, #ADDR          ; Erasing Address

ORL EEAEN, #01h          ; IAP routine access enable
MOV EECNTLD, #80h        ; Program/Erase Time Count Loading

CALL iap_eeprom_erase    ; Call IAP routine

ANL EEAEN, #0FEh         ; IAP routine access disable

MOV IE, r1               ; restore IE SFR
POP DPH                  ; restore acc, b, dptr
POP DPL
POP ACC
```

7. Recommended Power Slope

- The supply voltage slope must be rise faster than 1.0V/40us. (5V/200us)
  (That is, the supply voltage should be increasing monotonically until it reaches to the normal range.)



8. 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 Pins active	-100mA
Output Current Low	One I/O Pin active	+30mA
	All I/O Pins active	+150mA
Storage Temperature	-	-65 °C to +150 °C
Soldering Temperature	-	260 °C for 10 seconds
Maximum Input Current (Hall Sensor)	-	20mA @+25 °C
Maximum Power Dissipation (Hall Sensor)	-	150mW @+25 °C
Operating Temperature (Hall Sensor)	-	-40 °C + +120 °C
Storage Temperature (Hall Sensor)	-	-40 °C + +150 °C

9. DC Characteristics (MCU)

*** TA = = -40 ºC ~ +85 ºC, V_{DD} = +2.2V to +3.6V unless otherwise specified.**

Parameter	Symbol	Pin	Conditions	Value			Unit
				Min.	Typ.	Max.	
Input Low Voltage	V_{IL1}	RESETB ,P0, P1, P2	V_{DD} = +2.2V to +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} = +2.2V to +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}= +3.2V to +3.6V (I_{OL} = -8.67mA) V_{DD}= +2.7V to +3.2V (I_{OL} = -6.09mA) V_{DD}= +2.2V to +2.7V (I_{OL} = -3.83mA)	-	-	$0.3V_{DD}$	V
Output High Voltage	V_{OH}	All Pins	V_{DD}= +3.2V to +3.6V (I_{OH} = -7.12mA) V_{DD}= +2.7V to +3.2V (I_{OH} = -4.85mA) V_{DD}= +2.2V to +3.3V (I_{OH} = -2.88mA)	$0.7V_{DD}$	-	-	V
	V_{OHP}	ALL Pins (Pull-up Resistor Only)	V_{DD}= +3.2V to +3.6V (I_{OHP} = -30.91uA) V_{DD}= +2.7V to +3.2V (I_{OHP} = -25.75uA) V_{DD}= +2.2V to +2.7V (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
Pin Capacitance	C_{IO}	All Pins	V_{DD} = 5V	-	10	-	pF

9. DC Characteristics (Hall Sensor)

*** TA = = -40 ºC ~ +85 ºC, V_{DD} = +2.2V to +3.6V unless otherwise specified.**

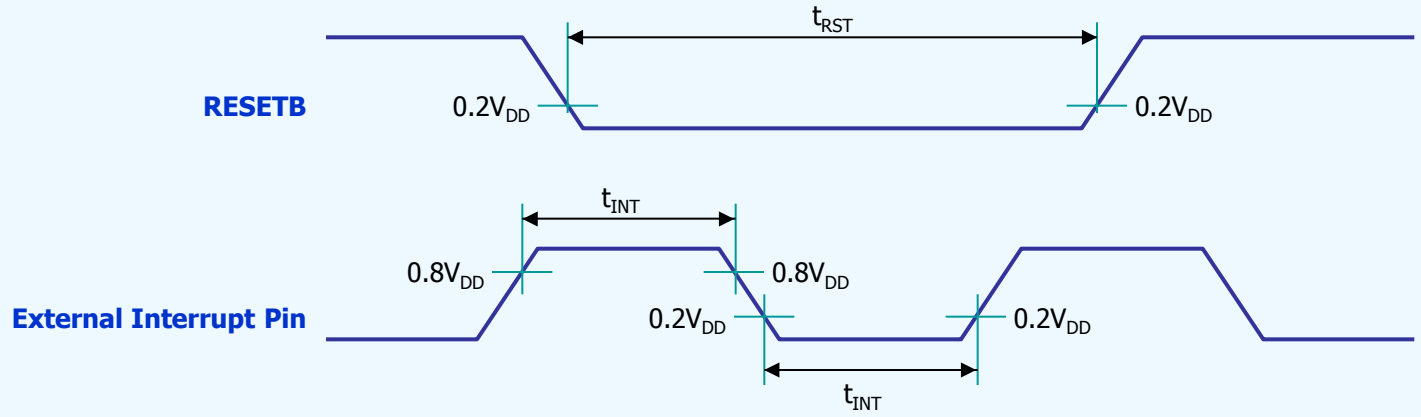
Parameter	Symbol	Pin	Conditions	Value			Unit
				Min.	Typ.	Max.	
Output Hall Voltage	V_H	-	VIN = +1V, B = 500G	196	-	320	mV
Input Resistance	R_{IN}	HALLIN [2:1]	I = 0.1mA	240	-	550	Ω
Output Resistance	R_{OUT}	-	I = 0.1mA	240	-	550	Ω
Offset Voltage	V_0	-	V_{IN} = +1V, B = 0G	-7	-	+7	mV
Temperature Coefficient of V_H	α	-	TA = 0℃ to +40℃	-	-	-1.8	%/℃
Temperature Coefficient Of R_{IN}, R_{OUT}	β	-	TA = 0℃ to +40℃	-	-	-1.8	%/℃

$V_H = V_{HM} - V_0$ (V_{HM} : The output voltage is measured at 500G)

10. AC Characteristics

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

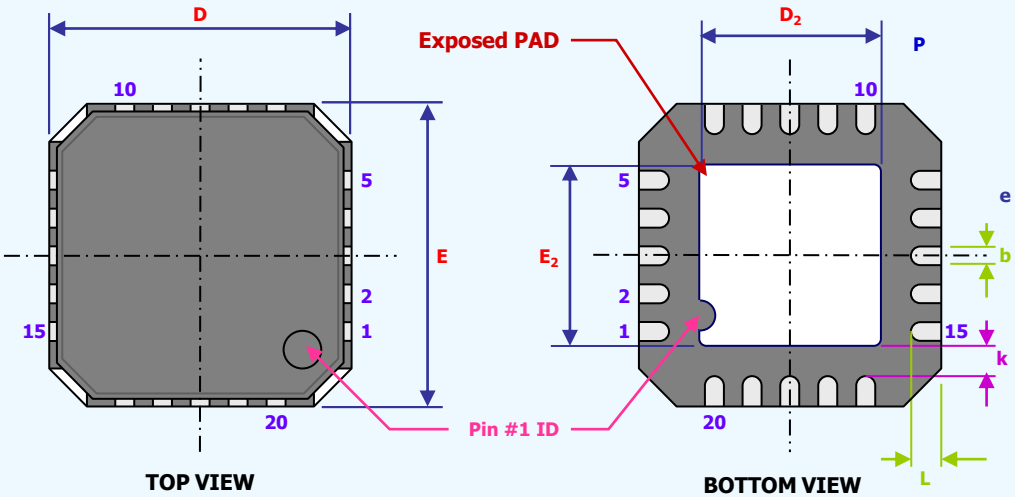
Parameter	Symbol	Pin	Conditions	Value			Unit
				Min.	Typ.	Max.	
Operating Frequency	F_{OSC}	XTAL1, XTAL2	V_{DD} = 5V ± 10%	1	-	23	MHz
			V_{DD} = 3V ± 10%	1	-	12	
RESETB Input Width	t_{RST}	RESETB	V_{DD} = 5V ± 10%	24	-	-	F_{OSC}
			V_{DD} = 3V ± 10%	24	-	-	
External Interrupt Input Width	t_{INT}	External Interrupt	V_{DD} = 5V ± 10%	4	-	-	F_{OSC}
			V_{DD} = 3V ± 10%	4	-	-	



11. ADC Specifications

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

12. Package Dimensions : 20-MLF

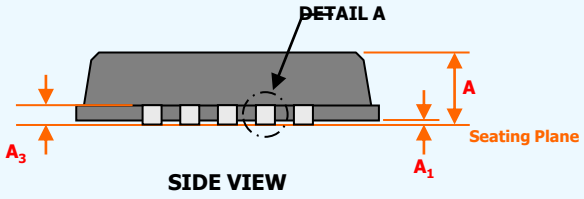


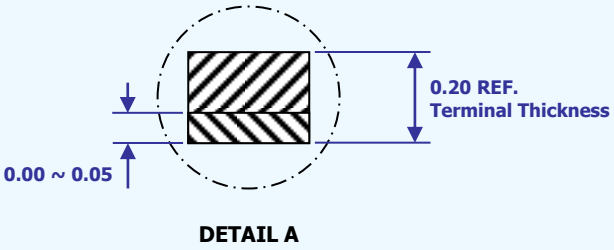
[20-MLF]

Symbol	Dimensions [mm]		
	Min.	Nom.	Max.
A	0.80	0.85	0.90
A_1	0.00	0.01	0.05
A_3		0.20 REF	
D		5.00 BSC	
E		5.00 BSC	
D_2	3.00	3.10	3.20
E_2	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

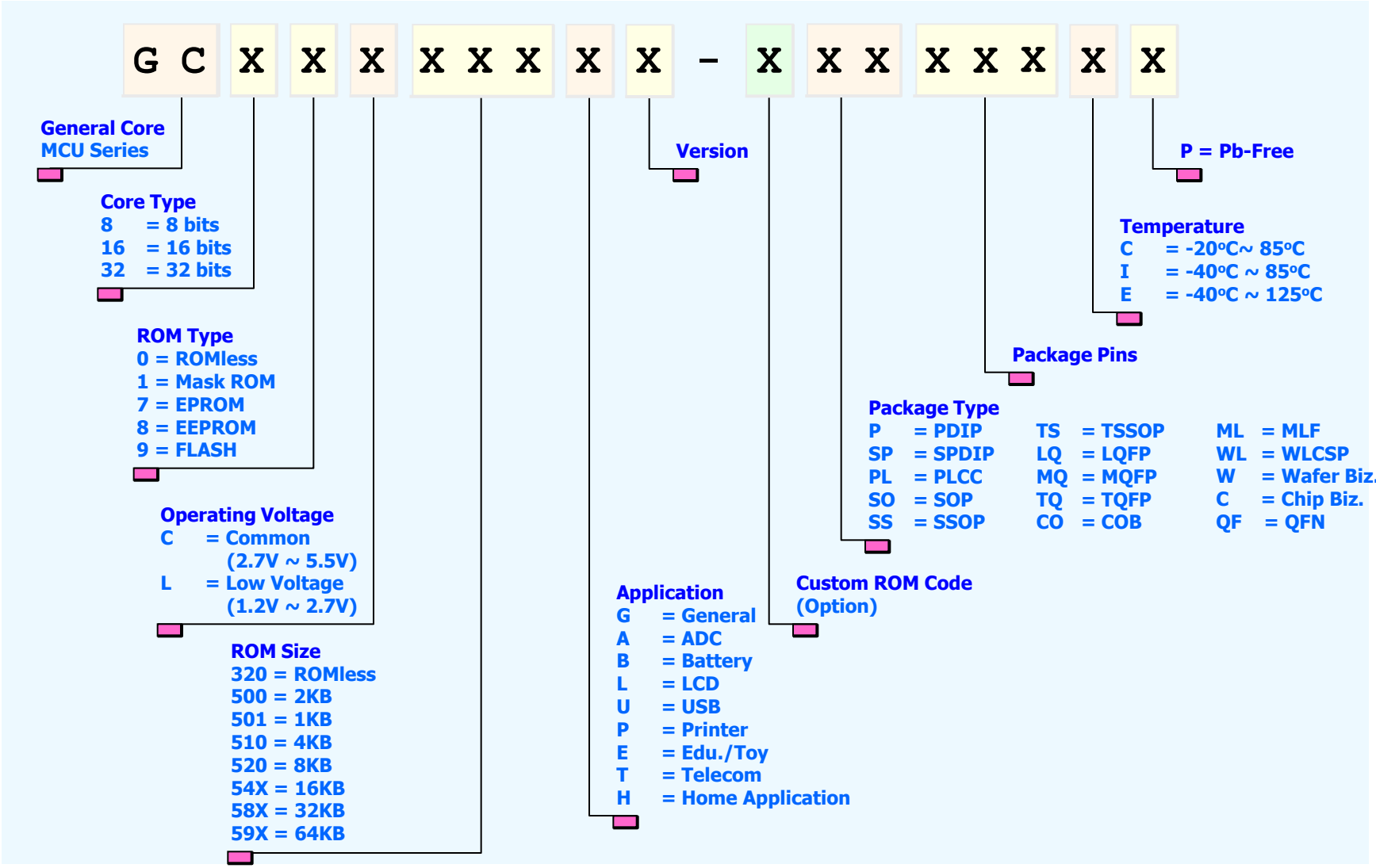
Notes:

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.



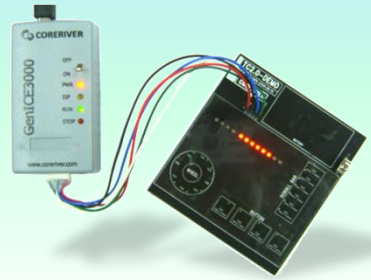


13. Product Numbering System



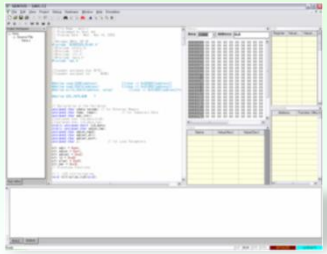
14. Supporting tools

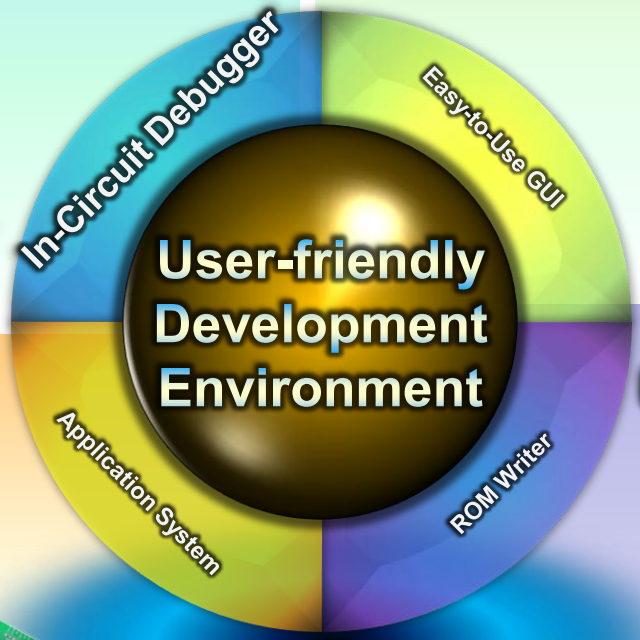
In-Circuit Debugger (GENSYS & GenICE)



Easy-to-Use GUI (GENTOS)

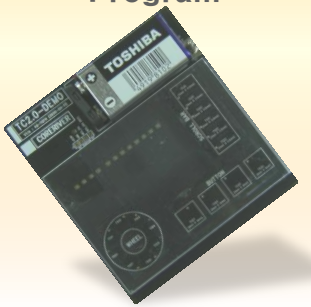
- Assembler & Linker for Windows
- Optimized Cross-C Compiler

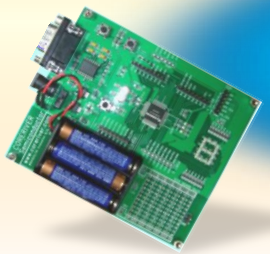




Application System

- On-board Application (with 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. (HallCore110 Family : 4kbytes 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>

1 cycle = 4 clocks

Add

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

Appendix A : Instruction Set (3/19)

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

Appendix A : Instruction Set (4/19)

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 [[(A_{3-0})>9] ∨ [(AC)=1]]
THEN (A_{3-0}) ← (A_{3-0})+6
IF [[(A_{7-4})>9] ∨ [(C)=1]]
THEN (A_{7-4}) ← (A_{7-4})+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

Appendix A : Instruction Set (7/19)

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

Appendix A : Instruction Set (9/19)

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

Appendix A : Instruction Set (10/19)

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

Appendix A : Instruction Set (12/19)

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)

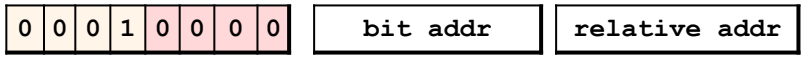
JBC bit, rel

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



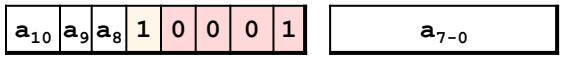
ACALL addr11

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



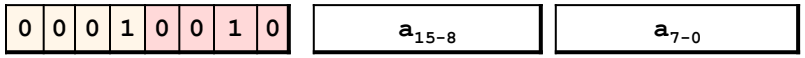
LCALL addr16

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}

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

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

Appendix A : Instruction Set (19/19)

DJNZ <byte>, rel

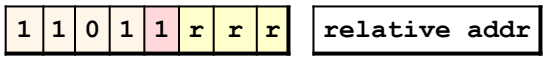
Decrement and Jump if Not Zero

DJNZ Rn, rel

Operation :

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

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



DJNZ direct, rel

Operation :

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

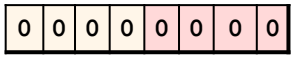
Encoding : HEX: D5h, #bytes: 3, Cycles: 4

NOP

No Operation

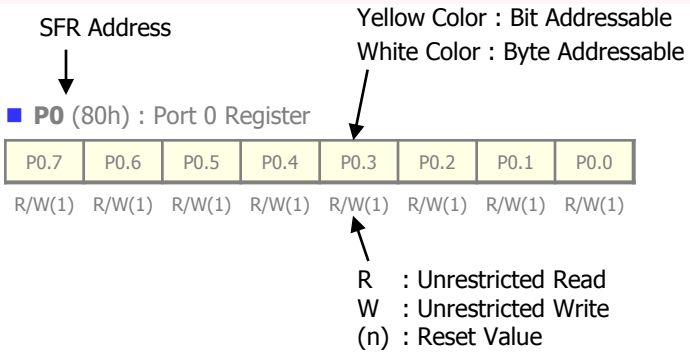
Operation : (PC) ← (PC) + 1

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



Appendix B : SFR Description [80h ~ 86h] (1/12)

[How to Read a SFR Descriptions]



■ **P0** (80h) : Port 0 Register

-	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)

◆ 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)

■ **ITSEL** (86h) : External Interrupt Control Register

IT3	IT2	-	-	IT3SEL	IT2SEL	IT1SEL	IT0SEL
R/W(1)	R/W(1)			R/W(0)	R/W(1)	R/W(0)	R/W(0)

◆ IT3 : External interrupt 3 type select flag.
Edge detect (IT3=1; Default) / Level detect (IT3=0)

◆ IT2 : External interrupt 2 type select flag.
Edge detect (IT2=1; Default) / Level detect (IT2=0)

◆IT3SEL : External Interrupt 3 type level and edge select register
0 = low level and falling edge Detect :Default
1 = high level and rising edge Detect

◆IT2SEL : External Interrupt 2 type level and edge select register
0 = low level and falling edge Detect
1 = high level and rising edge Detect :Default

◆IT1SEL : External Interrupt 1 type level and edge select register
0 = low level and falling edge Detect :Default
1 = high level and rising edge Detect

◆ IT0SEL : External Interrupt 0 type level and edge select register:
0 = low level and falling edge Detect :Default
1 = high level and rising edge Detect

Appendix B : SFR Description [87h ~ 8Dh] (2/12)

■ 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.

■ 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

■ 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)

Appendix B : SFR Description [90h ~ 99h] (3/12)

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

-	RTRG	IE3	IE2	XT/RG	RGMD	RGSL	BGS
	R/W(1)	R/W(0)	R/W(0)	R/W(0)	R(1)	R/W(0)	R/W(1)

- RT/RG : Internal RING clock selection [1] : 4MHz [0] : 32KHz
  If the flag is switched,
  RING 4MHz & RING 32KHz are switched into each other.
- IE3 : External interrupt 3 flag. Cleared by S/W.
- 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.

RINGCON (95h) : Ring Control Configuration Register

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

- RINGCON[7:0] : Internal Ring OSC. can be tuned.

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.

Appendix B : SFR Description [A0h ~ A8h] (4/12)

■ **P2** (A0h) : Port 2 Register

-	-	-	-	P2.3	P2.2	P2.1	P2.0
				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 [B0h ~ B3h] (5/12)

■ **I2CST0** (B0h) : I2C Status Register

I2CIF	I2COF	I2CACK	I2CRW	I2CDA	I2CP	I2CS	I2CBF
R/W(0)	R/W(0)	R (0)	R (0)	R (0)	R (0)	R (0)	R (0)

◆ I2CIF : I2C Master Interrupt Flag in slave & master mode.
[0] : Idle [1] : Interrupt occurred.
It is set each time a byte is received or transmitted.
If SP_IE flag in I2C_CFG SFR is set, it is set at Start/Stop condition.
The flag is set by H/W and cleared by S/W.

◆ I2COF : I2C Overflow Flag in slave & master mode
[0] : Idle [1] : Overflow occurred.
It is set when a byte is received while I2C_BUF SFR is still holding the previous byte.
It is set by H/W and cleared by S/W

◆ I2CACK : I2C Acknowledge flag in slave & master mode.
[0] : Indicate receiving Acknowledge bit.
[1] : Indicate receiving Not Acknowledge bit.

◆ I2CRW : I2C Read/Write flag in slave mode
[0] : Write state [1] : Read state

◆ I2CDA : Data / Address flag in slave mode
[0] : Indicates the last byte received or transmitted was Data
[1] : Indicates the last byte received or transmitted was Address

◆ I2CP : Stop flag in slave & master mode
[0] : Indicates Stop bit was not detected.
[1] : Indicates Stop bit was detected.
This flag is cleared when I2CS is set or I2CEN is cleared.

◆ I2CS : Start flag in slave & master mode
[0] : Indicates Start bit was not detected.
[1] : Indicates Start bit was detected.
This flag is cleared when I2CP is set or I2CEN is cleared.

◆ I2CBF : Busy flag in slave & master mode
[0] : RX not complete (Receiver), TX not complete (Transmitter)
[1] : RX complete (Receiver), TX complete (Transmitter)

■ **I2CCON0** (B3h) : I2C Control Register

-	SLA2ME	SCLHD	LASTB	PGEN	SGEN	I2CIOEN	I2CEN
-	R/W(0)	R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

◆ SLA2ME : 2nd Byte Slave Address Match Enable in Slave mode
[0] : 2nd Byte SLA Match Disable [1] : 2nd SLA Byte Match Enable

◆ SCLHD : Hold SCL 'low' for Wait State in Slave mode.
[0] : Hold SCL 'low'. The flag is cleared automatically by H/W
[1] : Release SCL 'float'. The flag is set by S/W

◆ LASTB : Indicate last byte in Master Receiver mode.
[0] : Send Acknowledge after last byte
[1] : Send Not Acknowledge after last byte
In Master Receiver mode, before receiving last byte, the flag must be set.

◆ PGEN : Generate Stop bit.
[0] : Start or Idle state. [1] : Generate Stop bit.
The flag is cleared automatically after Stop bit in Master mode and when I2CEN is cleared.

◆ SGEN : Generate Start bit
[0] : Stop or Idle state [1] : Generate Start bit
If the bus is not free, it waits for Stop bit condition.
The flag is cleared automatically after Start bit in Master mode and when I2CEN is cleared.

◆ I2CIOEN : Enable I2C IO
[0] : Disable I2C IO [1] : Enable I2C IO

◆ I2CEN : Enable I2C module
[0] : Disable I2C module [1] : Enable I2C module

Appendix B : SFR Description [B4h ~ B8h] (6/12)

I2CCFG0 (B4h) : I2C Configuration Register

-	-	-	-	MSSEL	ADSEL	SP_IE	GCE
-	-	-	-	R/W(0)	R/W(0)	R/W(0)	R/W(0)

◆MSSEL : I2C Master/Slave Mode Selection
[0] : Slave mode [1] : Master mode
◆ ADSEL : 7-bit / 10-bit Address Mode Selection in Slave mode
[0] : 7-bit mode [1] : 10-bit mode
◆ SP_IE : Start/Stop Interrupt Enable
[0] : Start/Stop Interrupt Disable [1] : Start/Stop Interrupt Enable
◆ GCE : General Call Enable in Slave mode
[1] : Respond to the general call address (0x00)

I2CSLA0 (B5h) : I2C Slave Address Register

SLA1.7	SLA1.6	SLA1.5	SLA1.4	SLA1.3	SLA1.2	SLA1.1	SLA1.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

◆ SLA[7:0] : I2C Slave Address Register.
In 7-bit address mode and in 10-bit address mode (1st SLA),
I2C_SLA[7:1] is used for matching address and I2C_SLA[0] is masked.
In 10-bit address mode (2nd SLA),
I2C_SLA[7:0] is used for matching address.

I2CDAT0 (B6h) : I2C Address / Data Register

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

I2CI2C0_SCL (B7h) : I2C SCL Clock Scaler

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

◆ MSCL[7:0] : Frequency scaler of I2C Master
F_{I2C} = FOSC / (2 * (MSCL[7:0] + 2))

IP (B8h) : Interrupt Priority Register

-	PADC	-	PS	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.
◆ 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.

Appendix B : SFR Description [BEh ~ C5h] (7/12)

■ OSCICN (BEh) : Internal Ring Oscillator Control Register

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

◆ RINGON : 1 = Internal ring oscillator(24MHz) is running.
0 = Internal ring oscillator is killed.
Don't clear RINGON bit when XTRG = 0.

◆ DIV2, DIV1, DIV0 : Ring 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

■ OSC2ICN (BFh) : Internal Ring2 Oscillator Control Register

-	-	-	-	-	-	OSC_REF	RING2ON
						R/W(0)	R/W(1)

◆ RING2ON: 1 = Internal ring oscillator(32KHz) is running.
0 = Internal ring oscillator is killed.
Don't clear RINGON bit when XTRG = 0.

◆ OSC_REF_SEL : RING reference selection

■ 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.

Appendix B : SFR Description [D0h ~ DBh] (8/12)

■ **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.

■ **WDMOD** (D1h) : Watchdog Mode register

-	-	-	-	-	-	-	WDM
							R/W(0)

- WD2 : Watchdog clock divide mode

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

-	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)

- 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.3	P2TYPE.2	P2TYPE.1	P2TYPE.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	EPFI	PFI	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 x 2^{16} clocks (interrupt)
  [0,1] : 4 x 2^{16} clocks (interrupt)
  [1,0] : 16 x 2^{16} clocks (interrupt)
  [1,1] : 32 x 2^{16} clocks (interrupt)
- EPFI : Power-fail interrupt enable.
- PFI : Power-fail interrupt flag.
- WDIF : Watchdog timer interrupt flag.
- WTRF : Watchdog timer Reset flag.
- EWT : Watchdog timer Reset Enable.
- RWT : Restart watchdog timer.

■ **ADCHSEL** (DBh) : ADC High Channel Selection Register

CH_SEL	-	-	-	-	-	-	-
R/W(0)							

- CH_SEL : ADC MUX Selector with CH[3:0].
  0 = CH[3:0] → ADC[11:10, 5:0] Enable (Default)
  1 = Disable

Appendix B : SFR Description [D9h ~ DEh] (9/12)

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

P0SEL	PS2_P0	PS1_P0	PS0_P0	MODE_P0	RL_P0	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)	R/W(0)

- P0SEL : 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] = $F_{OSC}/128$
  * PWM Clock (F_{PWM}) to ADC should not be set to Fosc/1.
- MODE_P0 : 8-bit / (2+6)-bit Counter Mode Selector.
  MODE_P0=0, (2+6)-bit Mode
  MODE_P0=1, 8-bit Mode
- 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.
- CLR_P0 : Counter reset enable. Cleared by H/W.
- RUN_P0 : Counter start enable. PWM clock (F_{PWM}) output enable.

■ **PWMIF** (DDh) : PWM Interrupt Register

-	-	-	-	-	-	-	PWMIF
							R/W(0)

- PWMF : PWM interrupt flag. Cleared by S/W.

■ **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)

Appendix B : SFR Description [E0h ~ E2h] (10/12)

■ **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	-	-	ADC6B	ADC5B	ADC4B
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 = ADC10 input enable & digital input disable at P2.2.
- ◆ ADC6B : 0 = ADC10 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] = No ADC Input Select (6h)
[0,1,1,1] = No ADC Input Select (7h)
[1,0,0,0] = No ADC Input Select (8h)
[1,0,0,1] = No ADC Input Select (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)

Appendix B : SFR Description [E3h ~ EFh] (11/12)

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

IOXEN	IORSTEN	CLO	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.
- CLO : 1 = System clock output to P2.6.
- 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.

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

-	P0SEL.6	P0SEL.5	P0SEL.4	P0SEL.3	P0SEL.2	P0SEL.1	P0SEL.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 is ON (Default)
- 1 = Pull-up resistor is OFF when ADC_EN (ADCON[7]) = 1

■ **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 (Default)
- 1 = Pull-up resistor is OFF when ADC_EN (ADCON[7]) = 1

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

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

- 0 = Pull-up resistor is ON (Default)
- 1 = Pull-up resistor is OFF when ADC_EN (ADCON[7]) = 1

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

-	-	EPWM	EWDT	EI2C1	EI2C0	EX3	EX2
		R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- EPWM : PWM interrupt enable.
- EWDT : Watchdog interrupt enable.
- EI2C1 : I2C 1 interrupt enable.
- EI2C0 : I2C 0 interrupt enable.
- EX3 : External 3 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	-	ADIV	SAR1	SAR0
R/W(0)	R/W(0)	R(1)	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.
- 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.

Appendix B : SFR Description [F0h ~ F0h] (12/12)

■ **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.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)

◆ 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.3	P2DIR.2	P2DIR.1	P2DIR.0
				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	PI2C1	PI2C0	PX3	PX2
		R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

◆ PPWM : PWM interrupt priority bit.
◆ PWDT : Watchdog timer interrupt priority bit.
◆ PI2C1 : I2C 1 interrupt priority bit.
◆ PI2C0 : I2C 0 interrupt priority bit.
◆ PX3 : External interrupt 3 priority bit.
◆ PX2 : External interrupt 2 priority bit.

■ **EECNTLD** (F9h) : EEPROM Erase/Program Time Count Loading

EECNTLD	-	-	-	-	-	-	-
R/W(0)							

◆ EECNTLD : EEPROM Erase/Program Time Count Loading
Set by S/W, cleared by H/W automatically.

■ **EECNT** (FCh ~ FAh) : EEPROM Erase/Program Time Count

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

■ **EEAEN** (FFh) : EEPROM Access Enable

-	-	-	-	-	-	-	EAEN
							R/W(0)

◆ EAEN = 1, EEPROM access enable.

Appendix C : Update History

- V1.0
  - ✓ First Release.