BM

Brief Manual of AmpCore100 Family

Flash / ISP / IAP
8-bit Turbo Microcontrollers

V1.0

August 2010

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

Contents

1. Product Overview

- **CORERIVER's AmpCore100 Family is a group of fast 80C52 compatible microcontrollers.**
- **The instruction execution of AmpCore100 Family is max. 3 times faster than that of traditional 80C52.**
  - ✓ **1 machine cycle = 4 clocks vs. 12 clocks**
- **Additional peripherals of AmpCore100 Family:**
  - ✓ **I2C / SPI / UART / 10-bit ADC / 8-bit PWM / OP-Amp / WDT / LVD / POR.**
- **Two independent high gain operational amplifiers with internal frequency compensation**
- **Wide operating voltage range for OP-Amp**
- **Single Power Supply / Dual power supply for OP-Amp**
- **Power saving modes**
- **Noise tolerant scheme**
- **Provides User-Friendly MDS environment**
- **Provides Easy-to-Use training-kit system**

1. Product Overview (Cont'd)

A. AmpCore100 Family

Product	Flash (byte)	EEPROM (byte)	RAM (Byte)	Volt (V)	Freq. (MHz)	T/C (16bits)	COMM I/O	WDT	ADC (bit x ch)	PWM (bit x ch)	OP-Amp	Package	Others	Available Time
AmpCore100-ML40I	8K	(1K)	512	2.2~5.5 (MCU) 3.0~32.0 (OP-Amp)	6 (24)	3	1 UART 1 I2C 1 SPI	YES	10 X 24	8 X 14	2	40-MLF	ISP IAP EJTAG LVD POR RING	**Now**

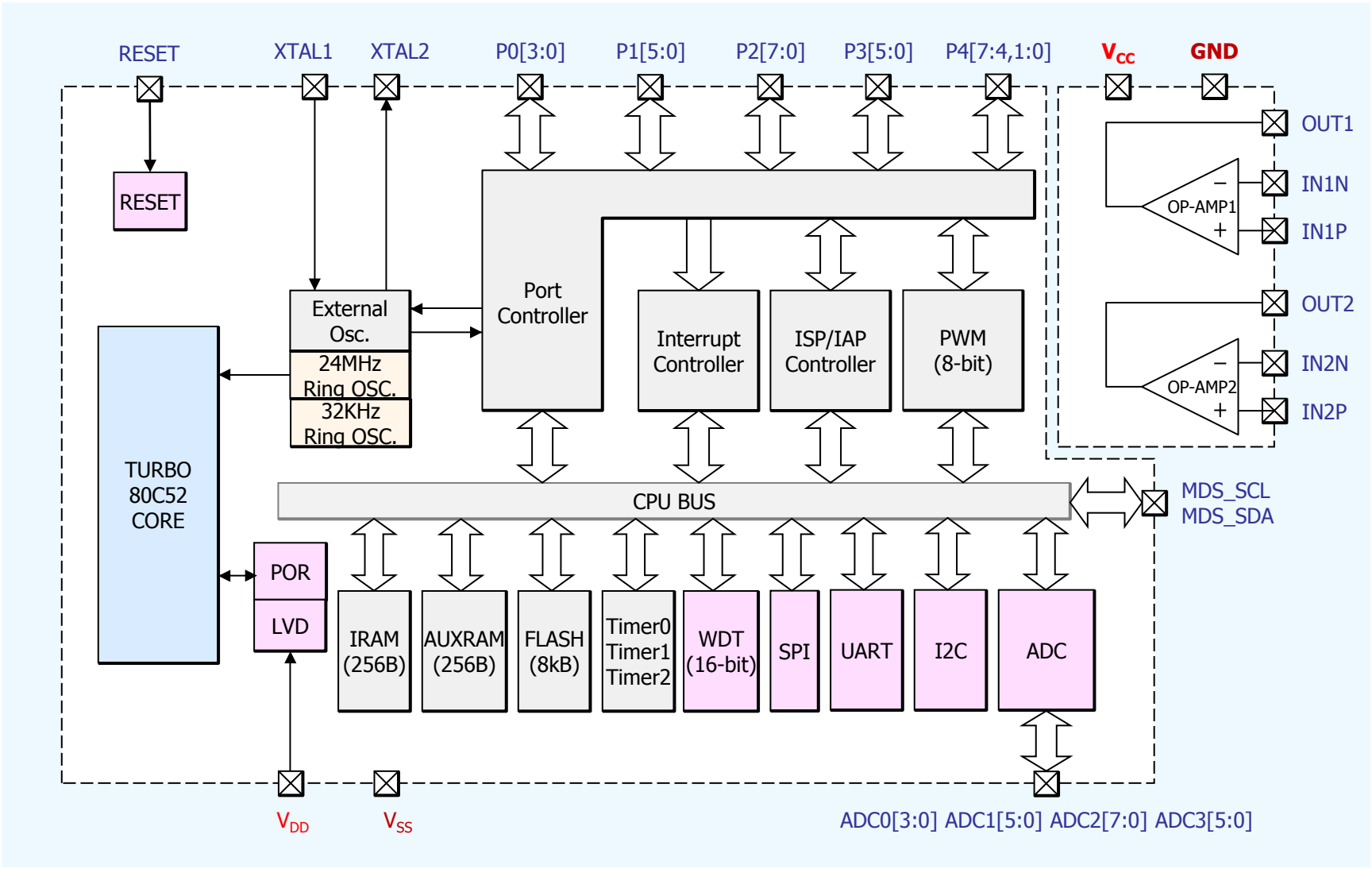
2. Features (MCU)

- CPU
  - ✓ 8-bit turbo 80C52 architecture
  - ✓ 4 cycles/1 machine cycle
  - ✓ instruction level compatible with Intel 80C52
- 8kB FLASH (Including 1kB User EEPROM)
- 256B Internal Aux. RAM
- 256B Internal RAM
- Operating Voltage : +2.2V ~ +5.5V
- Operating Temperature : -40°C ~ 85°C
- Operating Frequency
  - ✓ Max. 6MHz @ 3V
- Max. Programmable 30 I/O Pins
  - ✓ Pull-up control, Open drain, Push-Pull output
  - ✓ Pull-down control, Open drain, Push-Pull output (P4.7)
  - ✓ TTL and CMOS compatible logic levels
- Low Voltage Detector : +1.6V
- Internal Ring OSC with Calibration function
  - ✓ Max. 6MHz @ 5.0V
  - ✓ 4MHz @ 2.7V ~ 5.0V (+/- 3%)
  - ✓ 32KHz @ 2.7V (+/-10%) (Low power OSC)
- 24-channel 10-bit ADC
- 2-channel differential OP-AMP
- Supporting ISP/IAP/MDS
- 16-bit Programmable Watchdog Timer
- Three 16-bit Timer/Counters
- 1-channel I2C (Master/Slave)
- 1-channel SPI (Master/Slave)
- 1-channel UART
- 12-channel 8bit high speed PWM for DIMMING
- 12 Interrupt Sources
  - ✓ Timer0/1/2, ADC, WDT, I2C, SPI, UART
  - ✓ 4 External Source : both edge/level
  - ✓ Two-level interrupt priority
- Reset Sources
  - ✓ On-chip power-on-reset
  - ✓ External reset
  - ✓ Low voltage detector reset
  - ✓ Watchdog timer reset
- Power Down Wake-up Sources
  - ✓ Reset Sources + 4 External interrupt (Both Levels)
- Power Consumption
  - ✓ active current : Max 1mA @3.3V, 2MHz
  - ✓ idle current : Max 0.5mA @3.3V, 2MHz
  - ✓ stop current : Max 1uA @3.3V (all clock off)
- E.S.D. Protection Up to 2,000V
- Latch-up Protection Up to ±200mA
- Package
  - ✓ 40-MLF (5mm X 5mm)

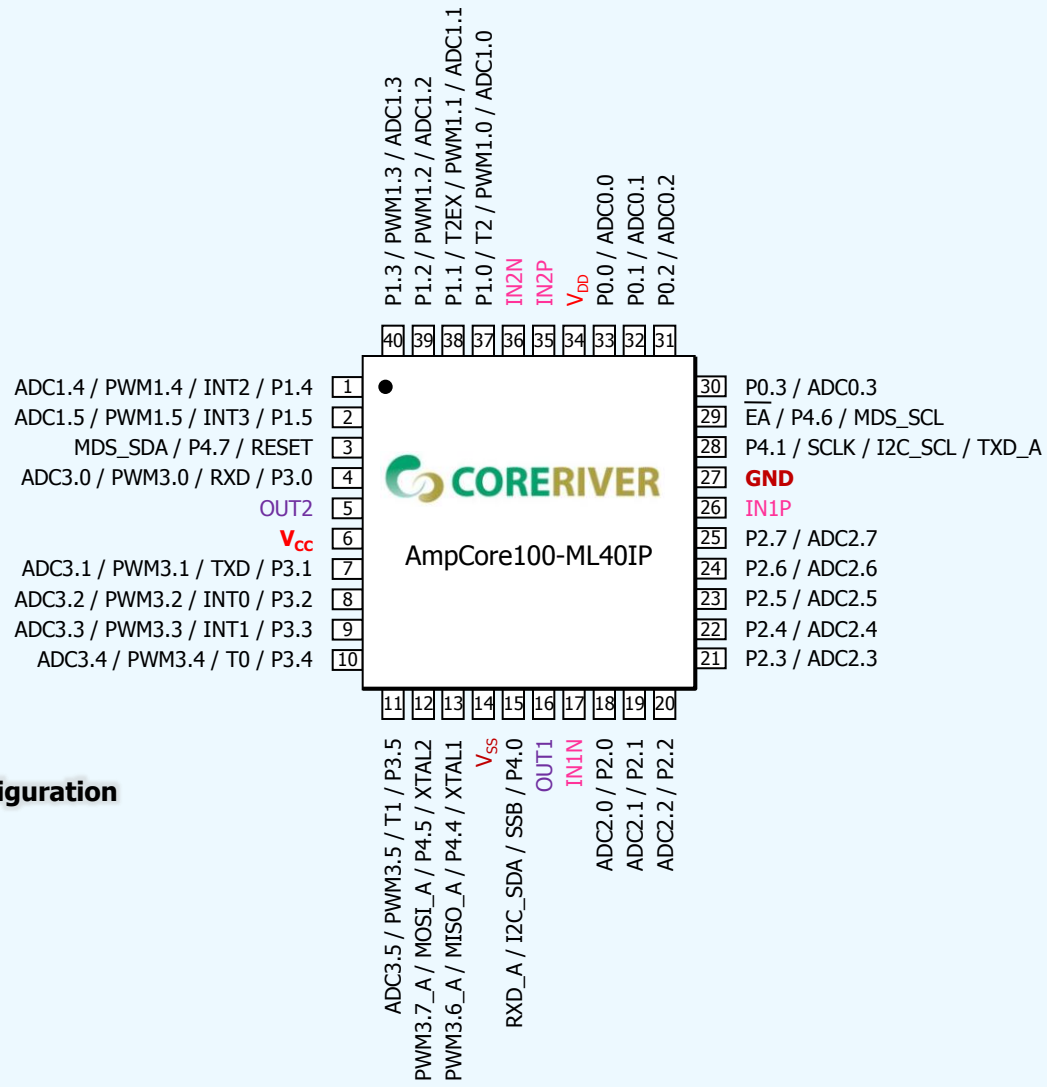
2. Features (OP-Amp)

- Internally Frequency Compensated for Unity Gain
- Large DC Voltage Gain : 100dB
- Wide Power Supply Rage :
  - ✓ +3.0V to +32.0V (or ±1.5V to ±16.0V)
- Input Common-mode Voltage Range included Ground
- Large Output Voltage Swing :
  - ✓ 0V DC to (V_{CC} – 1.5V) DC
- Power Drain Suitable for Battery Operation
- Low Input Offset Voltage and Offset Current
- Differential Input Voltage Range equals to the Power Supply Voltage.

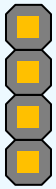
3. Block Diagram



4. Pin Configurations



ISP / MDS Pin Configuration



VDD (#34)

VSS (#14)

MDS_SCL (#29)

MDS_SDA (#3)

5. Pin Descriptions

Symbol	Direction	Description	Share Pins
V_{DD}	Input	Power Supply for MCU	-
V_{SS}	Input	Ground for MCU	-
V_{CC}	Input	Power Supply for OP-Amp (Operational Amplifier)	
GND	Input	Ground for OP-Amp	
RESET	Input	External Reset The pull-down resistor is turned on at power-on.	• P4.7 / MDS_SDA
XTAL1	Input	Input to the inverting oscillator amplifier	• P4.5 / MOSI_A / PWM3.7_A
XTAL2	Output	Output from the inverting oscillator amplifier	• P4.4 / MISO_A / PWM3.6_A
$\overline{EA}$	Input	External ROM Access Enable	• P4.6 / $\overline{MDS}$_SCL
MDS_SDA	Input/Output	Serial Data Pin for ISP/MDS.	• P4.7 / RESET
MDS_SCL	Input/Output	Serial Clock Pin for ISP/MDS.	• P4.6 / EA
P0[3:0]	Input/Output	◆ An 4-bit open-drain or push-pull I/O port or ADC Input. • P0.0 → ADC0.0 : A/D converter Input 0.0 • P0.1 → ADC0.1 : A/D converter Input 0.1 • P0.2 → ADC0.2 : A/D converter Input 0.2 • P0.3 → ADC0.3 : A/D converter Input 0.3	• P0.0 : ADC0.0 • P0.1 : ADC0.1 • P0.2 : ADC0.2 • P0.3 : ADC0.3

5. Pin Descriptions (Cont'd)

Symbol	Direction	Description	Share Pins
P1[5:0]	Input/Output	◆ An 6-bit open-Drain or push-pull I/O port or ADC Input. ◆ Note that the output is fully driven (push-pull) when P1 drives PWM1 output. • P1.0 → T2 : External Input for Timer/Counter 2 • P1.1 → T2EX : Timer/Counter 2 Capture/Reload Trigger • P1.4 → INT2 : External Interrupt 2 (Positive Edge) • P1.5 → INT3 : External Interrupt 3 (Negative Edge) • P1.0 → ADC1.0 : A/D converter Input 1.0 • P1.1 → ADC1.1 : A/D converter Input 1.1 • P1.2 → ADC1.2 : A/D converter Input 1.2 • P1.3 → ADC1.3 : A/D converter Input 1.3 • P1.4 → ADC1.4 : A/D converter Input 1.4 • P1.5 → ADC1.5 : A/D converter Input 1.5 • P1.0 → PWM1.0 : PWM output 1.0 • P1.1 → PWM1.1 : PWM output 1.1 • P1.2 → PWM1.2 : PWM output 1.2 • P1.3 → PWM1.3 : PWM output 1.3 • P1.4 → PWM1.4 : PWM output 1.4 • P1.5 → PWM1.5 : PWM output 1.5	• P1.0 : T2 / ADC1.0 / PWM1.0 • P1.1 : T2EX / ADC1.1 / PWM1.1 • P1.2 : ADC1.2 / PWM1.2 • P1.3 : ADC1.3 / PWM1.3 • P1.4 : INT2 / ADC1.4 / PWM1.4 • P1.5 : INT3 / ADC1.5 / PWM1.5
P2[7:0]	Input/Output	◆ An 8-bit open-drain or push-pull I/O port or ADC Input. • P2.0 → ADC2.0 : A/D converter Input 16 • P2.1 → ADC2.1 : A/D converter Input 17 • P2.2 → ADC2.2 : A/D converter Input 18 • P2.3 → ADC2.3 : A/D converter Input 19 • P2.4 → ADC2.4 : A/D converter Input 20 • P2.5 → ADC2.5 : A/D converter Input 21 • P2.6 → ADC2.6 : A/D converter Input 22 • P2.7 → ADC2.7 : A/D converter Input 23	• P2.0 : ADC2.0 • P2.1 : ADC2.1 • P2.2 : ADC2.2 • P2.3 : ADC2.3 • P2.4 : ADC2.4 • P2.5 : ADC2.5 • P2.6 : ADC2.6 • P2.7 : ADC2.7

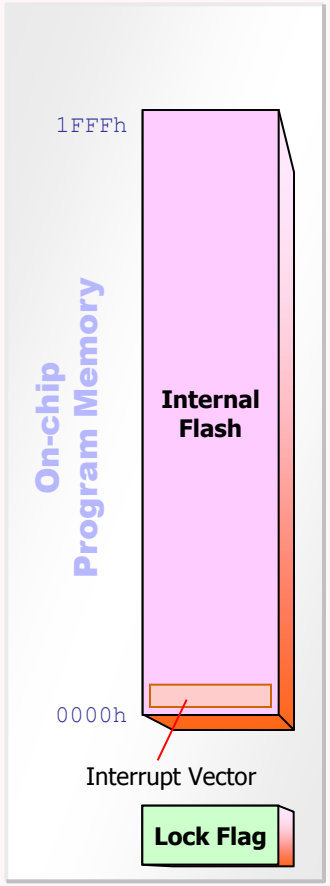
5. Pin Descriptions (Cont'd)

Symbol	Direction	Description	Share Pins
P3[5:0]	Input/Output	◆ An 6-bit open-drain or push-pull I/O port or ADC Input. • P3.0 → RXD : UART Serial Port Input • P3.1 → TXD : UART Serial Port Output • P3.2 → INT0 : External Interrupt Input 0 • P3.3 → INT1 : External Interrupt Input 1 • P3.4 → T0 : Timer 0 External Input • P3.5 → T1 : Timer 1 External Input • P3.0 → ADC3.0 : A/D converter Input 24 • P3.1 → ADC3.1 : A/D converter Input 25 • P3.2 → ADC3.2 : A/D converter Input 26 • P3.3 → ADC3.3 : A/D converter Input 27 • P3.4 → ADC3.4 : A/D converter Input 28 • P3.5 → ADC3.5 : A/D converter Input 29 • P3.0 → PWM3.0 : PWM output 3.0 • P3.1 → PWM3.1 : PWM output 3.1 • P3.2 → PWM3.2 : PWM output 3.2 • P3.3 → PWM3.3 : PWM output 3.3 • P3.4 → PWM3.4 : PWM output 3.4 • P3.5 → PWM3.5 : PWM output 3.5	• P3.0 : RXD / ADC3.0 / PWM3.0 • P3.1 : TXD / ADC3.1 / PWM3.1 • P3.2 : INT0 / ADC3.2 / PWM3.2 • P3.3 : INT1 / ADC3.3 / PWM3.3 • P3.4 : T0 / ADC3.4 / PWM3.4 • P3.5 : T1 / ADC3.5 / PWM3.5

5. Pin Descriptions (Cont'd)

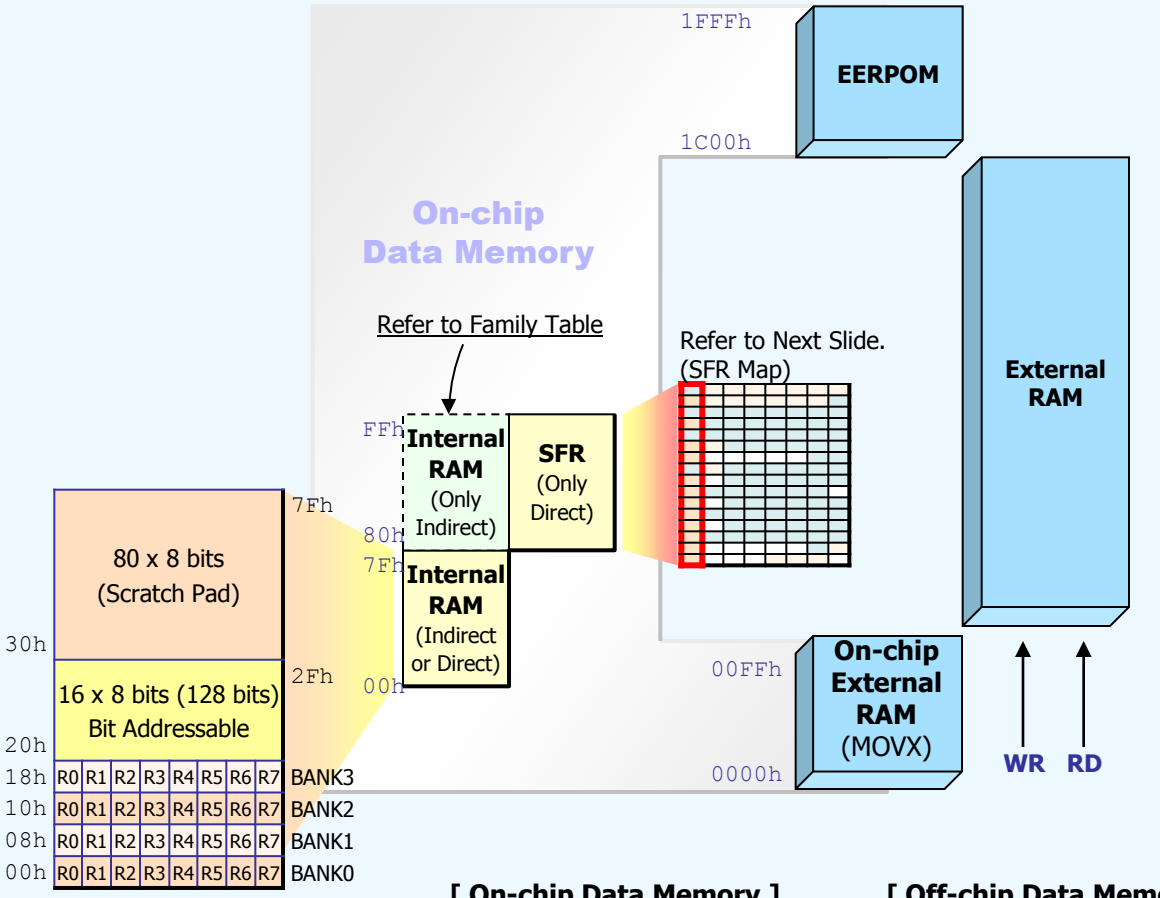
Symbol	Direction	Description	Share Pins
P4[7:4] P4[1:0]	Input/Output	◆ An 6-bit open-drain or push-pull I/O port. • P4.0 → I2C_SDA : I2C Serial Data • P4.1 → I2C_SCL : I2C Serial Clock • P4.0 → SSB : SPI Slave Select Bar • P4.1 → SCLK : SPI Serial Clock • P4.4 → MISO_A : SPI Master Input Slave Output Alternative • P4.5 → MOSI_A : SPI Master Output Slave Input Alternative • P4.0 → RXD_A : UART Serial Port Input Alternative • P4.1 → TXD_A : UART Serial Port Output Alternative • P4.4 → PWM3.6_A : PWM output 3.6 Alternative • P4.5 → PWM3.7_A : PWM output 3.7 Alternative	• P4.0 : SSB / I2C_SDA / RXD_A • P4.1 : SCLK / I2C_SCL / TXD_A • P4.4 : XTAL2 / MISO_A / PWM3.6_A • P4.5 : XTAL1 / MOSI_A / PWM3.7_A • P4.6 : EA / MDS_SCL • P4.7 : RESET / MDS_SDA
IN1N	Input	Negative Voltage Input of OP-Amp1	
IN1P	Input	Positive Voltage Input of OP-Amp1	
OUT1	Output	Output of OP-Amp1	
IN2N	Input	Negative Voltage Input of OP-Amp2	
IN2P	Input	Positive Voltage Input of OP-Amp2	
OUT2	Output	Output of OP-Amp2	

6.1. Memory Organization



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

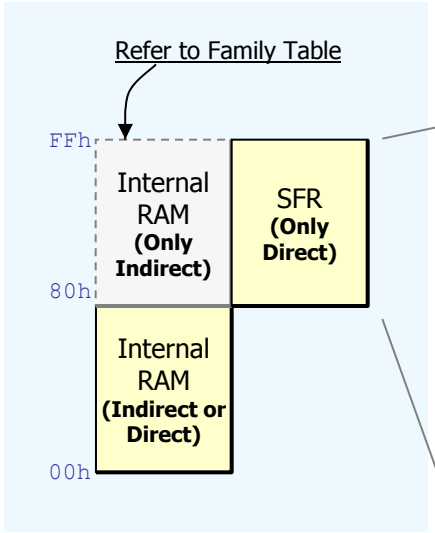
◆ User can write the data to FLASH or EEPROM with IAP (In-Application Programming).



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

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

6.2. SFR (Special Function Register) Map



Bit addressable

: Reserved for future use.

									FFh
F0h	B	EECNTLD	EECNTL	EECNTM			EECON	EEAEN	F7h
E8h	I2CST	I2CCON	I2CCFG	I2CSLA	I2CDAT	I2CSCL			EFh
E0h	ACC	P0DIR	P1DIR	P2DIR	P3DIR	P4DIR	ADCR		E7h
D8h	WDCON	P0TYP	P1TYP	P2TYP	P3TYP	P4TYP	ADCON	ADCSEL	DFh
D0h	PSW	P0SEL	P1SEL	P2SEL	P3SEL	P4SEL	ADCENB2	ADCENB3	D7h
C8h	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2	ADCENB0	ADCENB1	CFh
C0h	SPIST				PMR	STATUS	OSCICN	OSCICN2	C7h
B8h	IP	SADEN	ITSEL						BFh
B0h	P3	EIP	IT	P4	SPICON	SPICK	SPIDR	IPH	B7h
A8h	IE	SADDR		PWM1OEN	PWM1D4	PWM1D5	PWM1D6	PWM1D7	AFh
A0h	P2	EIE	PWM1CON	PWM1CNT	PWM1D0	PWM1D1	PWM1D2	PWM1D3	A7h
98h	SCON	SBUF		PWM0OEN	PWM0D4	PWM0D5	PWM0D6	PWM0D7	9Fh
90h	P1	EXIF	PWM0CON	PWM0CNT	PWM0D0	PWM0D1	PWM0D2	PWM0D3	97h
88h	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	RINGCON	8Fh
80h	P0	SP	DPL	DPH	ALTSEL2	ALTSEL	CKSEL	PCON	87h

6.2. SFR Brief Description

- 80C52 SFR Registers

Register	Name	Reset Value
ACC	Accumulator	00000000
B	B Register	00000000
PSW	Program Status Word	00000000
SP	Stack Pointer	00000111
DPTR	Data Pointer (2 bytes)	
DPL	Low byte	00000000
DPH	High byte	00000000
P0	Port 0	11111111
P1	Port 1	11111111
P2	Port 2	11111111
P3	Port 3	11111111
IP	Interrupt Priority Low	10000000
IPH	Interrupt Priority High	10000000
IE	Interrupt Enable Control	00000000
TCON	T/C 0/1 Control	00000000
TMOD	T/C 0/1 Mode Control	00000000
T2CON	T/C 2 Control	00000000
T2MOD	T/C 2 Mode Selection	******00
TH0	T/C 0 High byte	00000000
TL0	T/C 0 Low byte	00000000
TH1	T/C 1 High byte	00000000
TL1	T/C 1 Low byte	00000000
TH2	T/C 2 High byte	00000000
TL2	T/C 2 Low byte	00000000
RCAP2H	T/C 2 Capture Reg. High byte	00000000
RCAP2L	T/C 2 Capture Reg. Low byte	00000000
SCON	Serial Control	00000000
SBUF	Serial Buffer	00000000
SADEN	Slave Address Mask Enable	00000000
SADDR	Slave Address	00000000
PCON	Power Control	00*10000

- Newly added SFR Registers in AmpCore100 Family

Register	Name	Reset Value
P4	Port 4	11111111
P0SEL	Port 0 Pull-up Control	11111111
P1SEL	Port 1 Pull-up Control	00000000
P2SEL	Port 2 Pull-up Control	00000000
P3SEL	Port 3 Pull-up Control	00000000
P4SEL	Port 4 Pull-up Control	00110000
P0DIR	Port 0 Input/Output Control	00000000
P1DIR	Port 1 Input/Output Control	00000000
P2DIR	Port 2 Input/Output Control	00000000
P3DIR	Port 3 Input/Output Control	00000000
P4DIR	Port 4 Input/Output Control	00000000
P0TYP	Port 0 Type Control	11111111
P1TYP	Port 1 Type Control	11111111
P2TYP	Port 2 Type Control	11111111
P3TYP	Port 3 Type Control	11111111
P4TYP	Port 4 Type Control	11111111
IT	Interrupt Type	****1111
ITSEL	Interrupt Selection	**010100
ALTSEL	Alternative Pin Selection	*****010
ALTSEL2	Alternative Pin Selection2	*0000000
WDCON	Power Flag and Watchdog Timer Control	01010000
CKCON	Watchdog Timer and 4-cycle Switching Control	111000**
CKSEL	Clock Selection	***00*11
RINGCON	RING Calibration Control	01111011
PMR	Power Management Control	****00**
EXIF	Added External Interrupt and LVD Control	**000101
EIP	Extended Interrupt Priority	0*00**00
EIE	Extended Interrupt Enable	0*00**00
STATUS	Crystal Status	***0****
OSCICN	RING Oscillator Control	****1100
OSCICN2	RING Oscillator Control2	*****1**

CAUTION : Don't touch bit *. Updating these bits will cause the malfunctions.

6.2. SFR Brief Description

- Newly added SFR Registers in AmpCore100 Family

Register	Name	Reset Value
I2CST	I2C Status	00000000
I2CCON	I2C Control	*0100000
I2CCFG	I2C Configuration	****0000
I2CSLA	I2C Slave Address	00000000
I2CDAT	I2C Data	00000000
I2CSCL	I2C Clock Scaling	00000000
SPIST	SPI Status	****0000
SPICON	SPI Control	*0000000
SPICK	SPI SCLK Scaling	*****000
SPIDR	SPI Data	00000000
PWM0CON	PWM0 (P1) Control	*000**00
PWM0CNT	PWM0 (P1) Count	00000000
PWM0OEN	PWM0 (P1) Output Enable	**000000
PWM0D0	PWM0D0 (P1.0) Duty Data	00000000
PWM0D1	PWM0D1 (P1.1) Duty Data	00000000
PWM0D2	PWM0D2 (P1.2) Duty Data	00000000
PWM0D3	PWM0D3 (P1.3) Duty Data	00000000
PWM0D4	PWM0D4 (P1.4) Duty Data	00000000
PWM0D5	PWM0D5 (P1.5) Duty Data	00000000
PWM0D6	PWM0D6 (P1.6) Duty Data	00000000
PWM0D7	PWM0D7 (P1.7) Duty Data	00000000
PWM1CON	PWM1 (P3) Control	*000**00
PWM1CNT	PWM1 (P3) Count	00000000
PWM1OEN	PWM1 (P3) Output Enable	00000000
PWM1D0	PWM1D0 (P3.0) Duty Data	00000000
PWM1D1	PWM1D1 (P3.1) Duty Data	00000000
PWM1D2	PWM1D2 (P3.2) Duty Data	00000000
PWM1D3	PWM1D3 (P3.3) Duty Data	00000000
PWM1D4	PWM1D4 (P3.4) Duty Data	00000000
PWM1D5	PWM1D5 (P3.5) Duty Data	00000000
PWM1D6	PWM1D6 (P3.6) Duty Data	00000000
PWM1D7	PWM1D7 (P3.7) Duty Data	00000000

- Newly added SFR Registers in AmpCore100 Family

Register	Name	Reset Value
EECNTLD	EEPROM Program/Erase Count Load	0*******
EECNTL	EEPROM Program/Erase Count Low	00000000
EECNTM	EEPROM Program/Erase Count Low	00000000
EECON	EEPROM Control	*0*00000
EEAEN	EEPROM Access Enable	******00
ADCON	ADC Control & ADC Result Low	0010**00
ADCR	ADC Result High	00000000
ADCSEL	ADC Channel Selection Low and MUX Selection	00000000
ADCENB0	ADC Channel Enable Bar 0	****1111
ADCENB1	ADC Channel Enable Bar 1	**111111
ADCENB2	ADC Channel Enable Bar 2	11111111
ADCENB3	ADC Channel Enable Bar 3	**111111

CAUTION : Don't touch bit *. Updating these bits will cause the malfunctions.

6.3. Instruction Set Summary

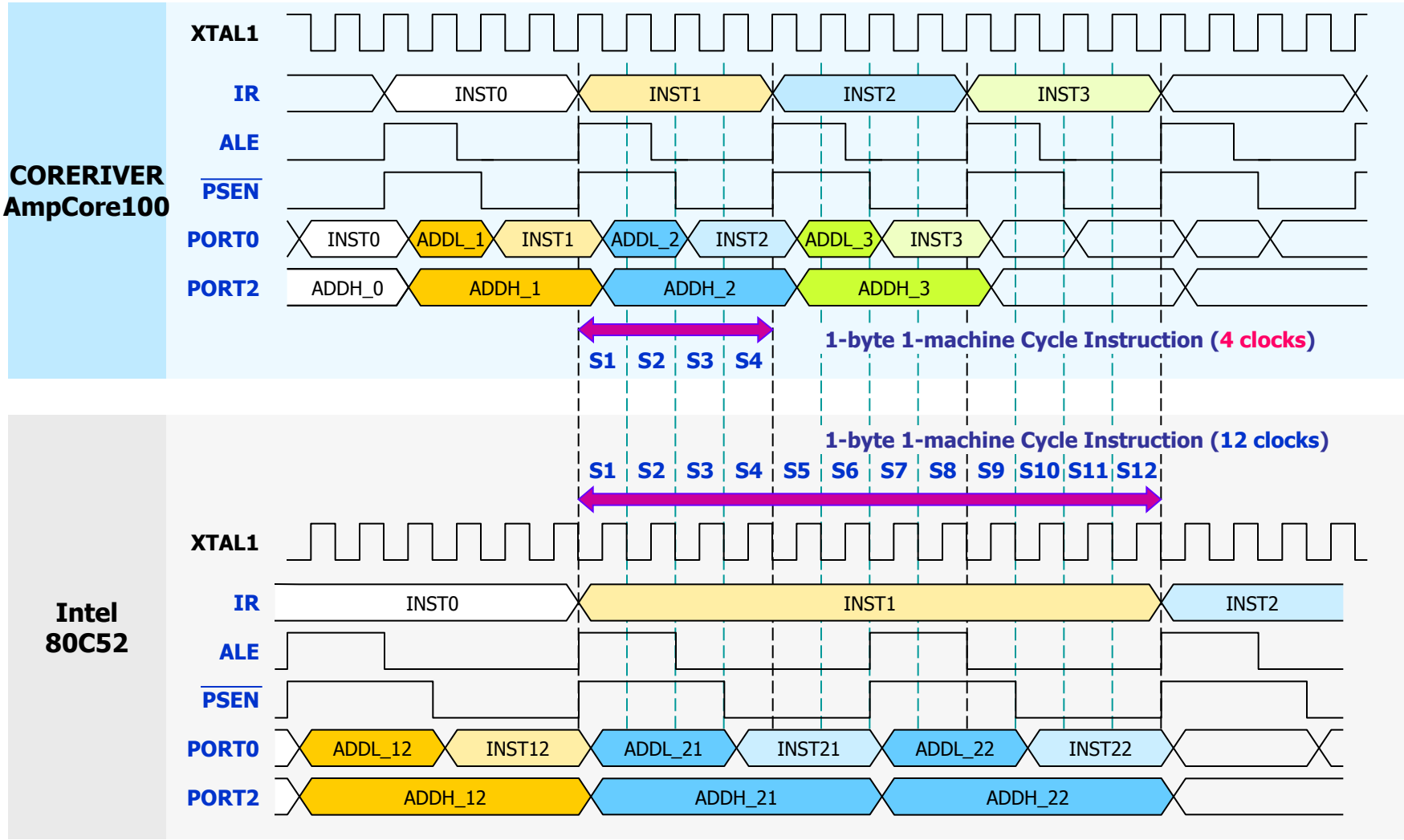
- Refer to Appendix A (Instruction Set) for more details.

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

- Instruction timing comparison of the AmpCore100 family and Intel 80C52



6.4. CPU Timing : Comparison Table

- The Fastest CPU timing in the world

Instruction	**AmpCore100 (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[3:0]

- Open-drain (compatible with Intel 8052) or push-pull output, pull-up control, ADC input.
- The alternative functions are available only when the corresponding SFR bit is "1".
  - ✓ P0.0 = SSB_A, I2C_SDA_A, RXD_B, ADC0.0 / P0.1 = SCLK_A, I2C_SCL_A, TXD_A, ADC0.1 / P0.2 = ADC0.2 / P0.3 = ADC0.3
- Read-Modify-Write instructions do not read port pin but the port SFR.
  - ✓ ANL / ORL / XRL / JBC / CPL / INC / DEC / DJNZ / MOV PX.Y, C / CLR PX.Y / SETB PX.Y

✓ **P0TYPE**(B9h) : Port 0 Type Control Register

-	-	-	-	P0TYPE.3	P0TYPE.2	P0TYPE.1	P0TYPE.0
				R/W(1)	R/W(1)	R/W(1)	R/W(1)

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

✓ **P0DIR** (E1h) : Port 0 Input/Output Control Register

-	-	-	-	P0DIR.3	P0DIR.2	P0DIR.1	P0DIR.0
				R/W(0)	R/W(0)	R/W(0)	R/W(0)

- 0 = Output (Default) / 1 = Input

✓ **P0SEL** (D1h) : Port 0 Pull-up Control Register

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

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

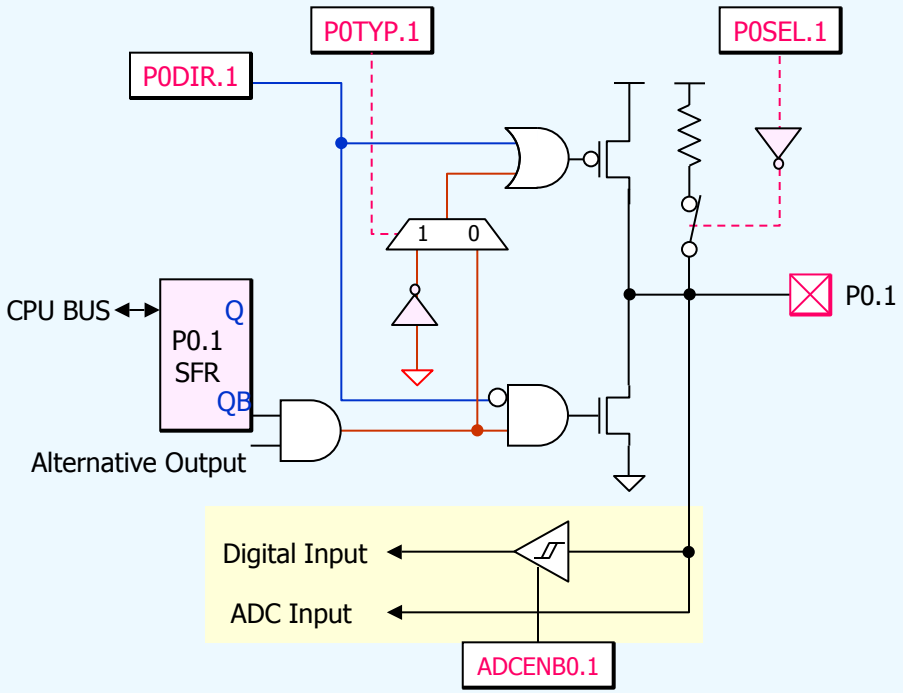
✓ **ADCENB0** (CEh) : ADC Channel Enable Bar Register (P0 port)

-	-	-	-	ADCENB0.3	ADCENB0.2	ADCENB0.1	ADCENB0.0
				R/W(1)	R/W(1)	R/W(1)	R/W(1)

- 0 = ADC0 channel ON / 1 = ADC0 channel OFF (Default)

✓ **P0** (80h) : Port 0 Register

-	-	-	-	P0.3	P0.2	P0.1	P0.0
				R/W(1)	R/W(1)	R/W(1)	R/W(1)



6.5. I/O Ports : PORT1[5:0]

- Open-drain or push-pull output, pull-up control, ADC input.
- The alternative functions are available only when the corresponding SFR bit is "1".
  - ✓ P1.0 = T2, PWM1.0, ADC1.0 / P1.1 = T2EX, PMW1.1, ADC1.1 / P1.2 = PWM1.2, ADC1.2 / P1.3 = PWM1.2, ADC1.3 / P1.4 = INT2, PWM1.4, ADC1.4 / P1.5 = INT3, PWM1.5, ADC1.5
- Read-Modify-Write instructions do not read port pin but the port SFR.
  - ✓ ANL / ORL / XRL / JBC / CPL / INC / DEC / DJNZ / MOV PX.Y, C / CLR PX.Y / SETB PX.Y

✓ **P1TYPE**(BAh) : Port 1 Type Control Register

-	-	P1TYPE.5	P1TYPE.4	P1TYPE.3	P1TYPE.2	P1TYPE.1	P1TYPE.0
		R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

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

✓ **P1DIR** (E2h) : Port 1 Input/Output Control Register

-	-	P1DIR.5	P1DIR.4	P1DIR.3	P1DIR.2	P1DIR.1	P1DIR.0
		R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- 0 = Output (Default) / 1 = Input

✓ **P1SEL** (D2h) : Port 1 Pull-up Control Register

-	-	P1SEL.5	P1SEL.4	P1SEL.3	P1SEL.2	P1SEL.1	P1SEL.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 = OFF

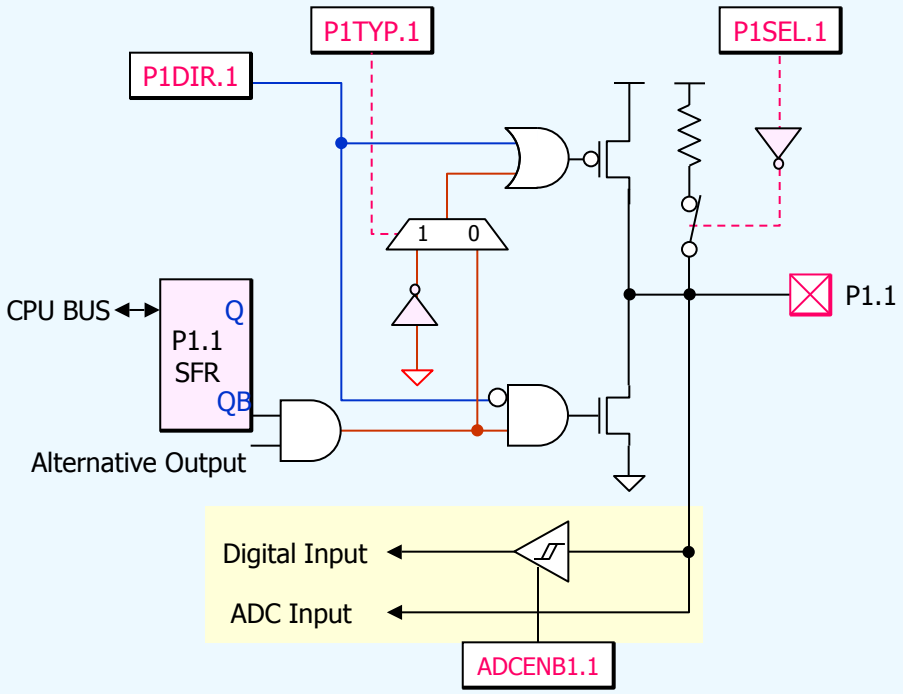
✓ **ADCENB1** (CFh) : ADC Channel Enable Bar Register (P1 port)

-	-	ADCENB1.5	ADCENB1.4	ADCENB1.3	ADCENB1.2	ADCENB1.1	ADCENB1.0
		R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

- 0 = ADC0 channel ON / 1 = ADC0 channel OFF (Default)

✓ **P1** (90h) : Port 1 Register

-	-	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
		R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)



6.5. I/O Ports : PORT2[7:0]

- Open-drain or push-pull output, pull-up control, ADC input.
- The alternative functions are available only when the corresponding SFR bit is "1".
  - ✓ P2.0 = ADC2.0 / P2.1 = ADC2.1 / P2.2 = ADC2.2 / P2.3 = ADC2.3 / P2.4 = ADC2.4 / P2.5 = ADC2.5 / P2.6 = ADC2.6 / P2.7 = ADC2.7
- Read-Modify-Write instructions do not read port pin but the port SFR.
  - ✓ ANL / ORL / XRL / JBC / CPL / INC / DEC / DJNZ / MOV PX.Y, C / CLR PX.Y / SETB PX.Y

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

P2TYPE.7	P2TYPE.6	P2TYPE.5	P2TYPE.4	P2TYPE.3	P2TYPE.2	P2TYPE.1	P2TYPE.0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

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

✓ **P2DIR** (E3h) : Port 2 Input/Output Control Register

P2DIR.7	P2DIR.6	P2DIR.5	P2DIR.4	P2DIR.3	P2DIR.2	P2DIR.1	P2DIR.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- 0 = Output (Default) / 1 = Input

✓ **P2SEL** (D3h) : Port 2 Pull-up Control Register

P2SEL.7	P2SEL.6	P2SEL.5	P2SEL.4	P2SEL.3	P2SEL.2	P2SEL.1	P2SEL.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

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

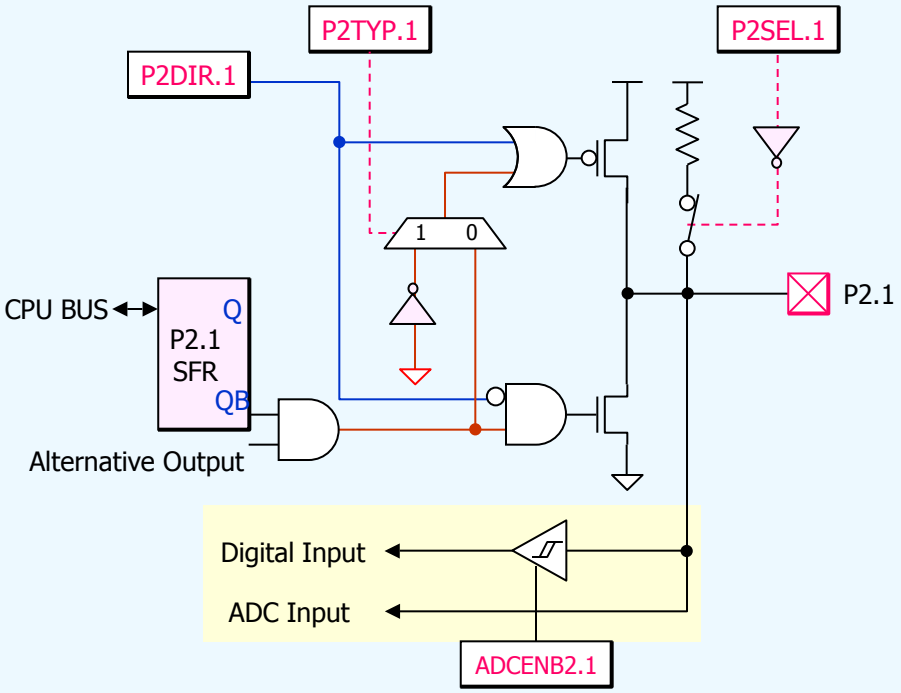
✓ **ADCENB2** (D6h) : ADC Channel Enable Bar Register (P2 port)

ADCENB2.7	ADCENB2.6	ADCENB2.5	ADCENB2.4	ADCENB2.3	ADCENB2.2	ADCENB2.1	ADCENB2.0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

- 0 = ADC0 channel ON / 1 = ADC0 channel OFF (Default)

✓ **P2** (A0h) : Port 2 Register

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



6.5. I/O Ports : PORT3[5:0]

- Open-drain or push-pull output, pull-up control, ADC input.
- The alternative functions are available only when the corresponding SFR bit is "1".
  - ✓ P3.0 = RXD, PWM3.0, ADC3.0 / P3.1 = TXD, PWM3.1, ADC3.1 / P3.2 = INT0, PWM3.2, ADC3.2 / P3.3 = INT1, PWM3.3, ADC3.3 / P3.4 = T0, PWM3.4, ADC3.4 / P3.5 = T1, PWM3.5, ADC3.5
- Read-Modify-Write instructions do not read port pin but the port SFR.
  - ✓ ANL / ORL / XRL / JBC / CPL / INC / DEC / DJNZ / MOV PX.Y, C / CLR PX.Y / SETB PX.Y

✓ **P3TYPE**(BCh) : Port 3 Type Control Register

-	-	P3TYPE.5	P3TYPE.4	P3TYPE.3	P3TYPE.2	P3TYPE.1	P3TYPE.0
		R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

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

✓ **P3DIR** (E4h) : Port 3 Input/Output Control Register

-	-	P3DIR.5	P3DIR.4	P3DIR.3	P3DIR.2	P3DIR.1	P3DIR.0
		R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- 0 = Output (Default) / 1 = Input

✓ **P3SEL** (D4h) : Port 3 Pull-up Control Register

-	-	P3SEL.5	P3SEL.4	P3SEL.3	P3SEL.2	P3SEL.1	P3SEL.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 = OFF

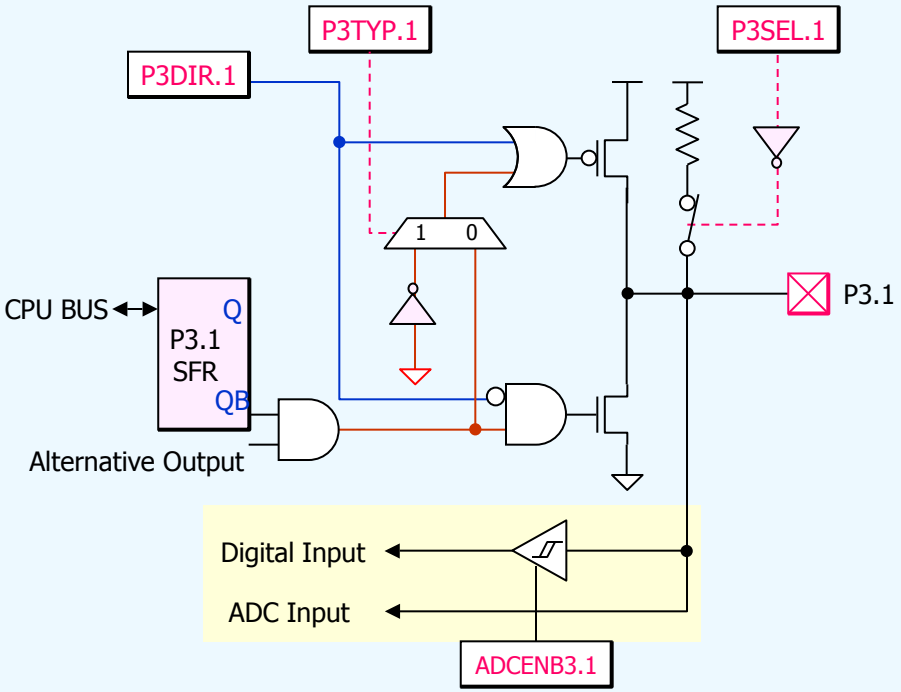
✓ **ADCENB3** (D7h) : ADC Channel Enable Bar Register (P3 port)

-	-	ADCENB3.5	ADCENB3.4	ADCENB3.3	ADCENB3.2	ADCENB3.1	ADCENB3.0
		R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

- 0 = ADC0 channel ON / 1 = ADC0 channel OFF (Default)

✓ **P3** (B0h) : Port 3 Register

-	-	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
		R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)



6.5. I/O Ports : PORT4[6:4,1:0]

- Open-drain or push-pull output, pull-up control, ADC input.
- The alternative functions are available only when the corresponding SFR bit is "1".
  - ✓ P4.0 = SSB, I2C_SDA, RXD_A / P4.1 = SCLK,I2C_SCL, TXD_A / P4.4 = MISO_A, PWM3.6_A / P4.5 = MOSI_A, PWM3.7_A / P4.6 = EA, MDS_SCL
- Read-Modify-Write instructions do not read port pin but the port SFR.
  - ✓ ANL / ORL / XRL / JBC / CPL / INC / DEC / DJNZ

✓ **P4TYPE**(BDh) : Port 4 Type Control Register

P4TYPE.7	P4TYPE.6	P4TYPE.5	P4TYPE.4	-	-	P4TYPE.1	P4TYPE.0
R/W(1)	R/W(1)	R/W(1)	R/W(1)			R/W(1)	R/W(1)

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

✓ **P4DIR** (E5h) : Port 4 Input/Output Control Register

P4DIR.7	P4DIR.6	P4DIR.5	P4DIR.4	-	-	P4DIR.1	P4DIR.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)			R/W(0)	R/W(0)

- 0 = Output / 1 = Input (Default)

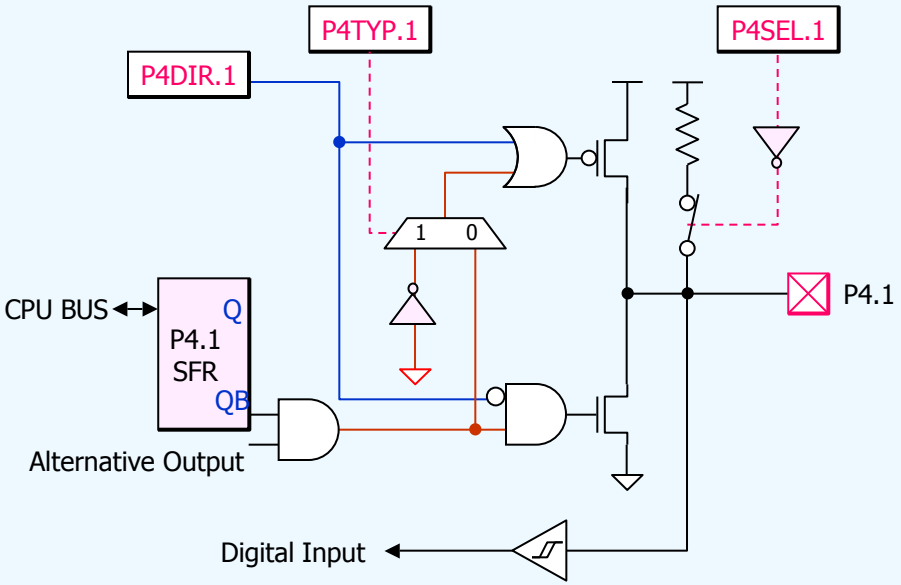
✓ **P4SEL** (D5h) : Port 4 Pull-up Control Register

P4SEL.7	P4SEL.6	P4SEL.5	P4SEL.4	-	-	P4SEL.1	P4SEL.0
R/W(0)	R/W(0)	R/W(1)	R/W(1)			R/W(0)	R/W(0)

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

✓ **P4** (B3h) : Port 4 Register

P4.7	P4.6	P4.5	P4.4	-	-	P4.1	P4.0
R/W(1)	R/W(1)	R/W(1)	R/W(1)			R/W(1)	R/W(1)



6.5. I/O Ports : PORT4[7]

- Open-drain or push-pull output, pull-up control, ADC input.
- The alternative functions are available only when the corresponding SFR bit is "1".
  - ✓ P4.7 = RESET, MDS_SDA
- Read-Modify-Write instructions do not read port pin but the port SFR.
  - ✓ ANL / ORL / XRL / JBC / CPL / INC / DEC / DJNZ

✓ **P4TYPE**(BDh) : Port 4 Type Control Register

P4TYPE.7	P4TYPE.6	P4TYPE.5	P4TYPE.4	-	-	P4TYPE.1	P4TYPE.0
R/W(1)	R/W(1)	R/W(1)	R/W(1)			R/W(1)	R/W(1)

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

✓ **P4DIR** (E5h) : Port 4 Input/Output Control Register

P4DIR.7	P4DIR.6	P4DIR.5	P4DIR.4	-	-	P4DIR.1	P4DIR.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)			R/W(0)	R/W(0)

- 0 = Output / 1 = Input (Default)

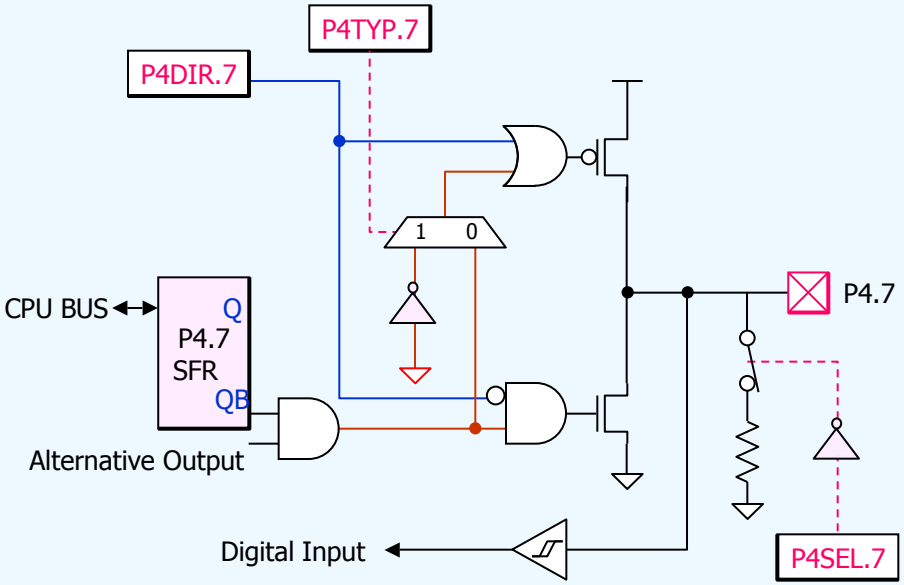
✓ **P4SEL** (D5h) : Port 4 Pull-up Control Register

P4SEL.7	P4SEL.6	P4SEL.5	P4SEL.4	-	-	P4SEL.1	P4SEL.0
R/W(0)	R/W(0)	R/W(1)	R/W(1)			R/W(0)	R/W(0)

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

✓ **P4** (B3h) : Port 4 Register

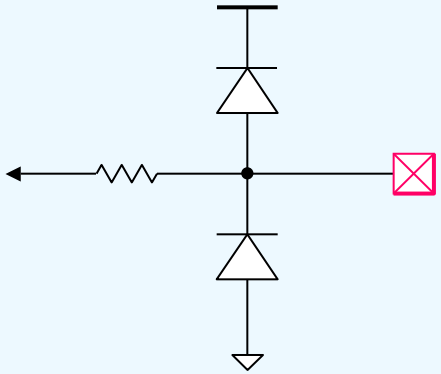
P4.7	P4.6	P4.5	P4.4	-	-	P4.1	P4.0
R/W(1)	R/W(1)	R/W(1)	R/W(1)			R/W(1)	R/W(1)



6.6. The ESD Structure of Pads

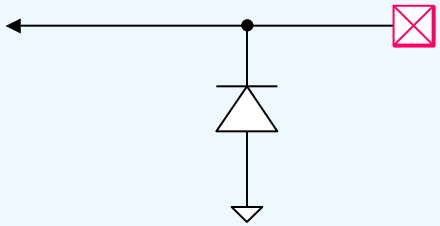
- Two ESD diodes and one ESD resister are contained in all pads except VDD.
- One ESD diode are contained in VDD.

[All pads except VDD]



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

[VDD]



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

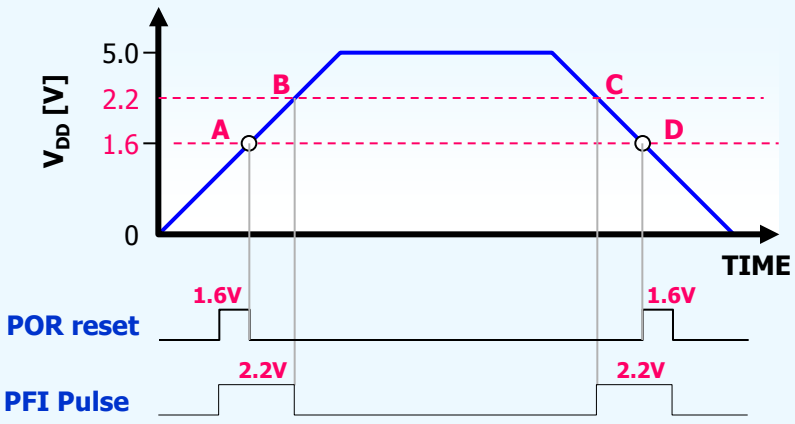
6.7. LVD (Low Voltage Detector)

- On-chip power-on reset : 1.6V
- On-chip power-fail reset : 1.6V
- Optional power-fail interrupt : 2.2V

- Flag Transition

	POF	POR	PFI
A	X → 1	X → 1	X
B	1	1	X → 1
C	X	X	X → 1
D	X → 1	X → 1	1

- POF is a mirror of POR.



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

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

- BGS : Band-gap Select
  0 = LVD Block Off / 1 = LVD Block ON

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

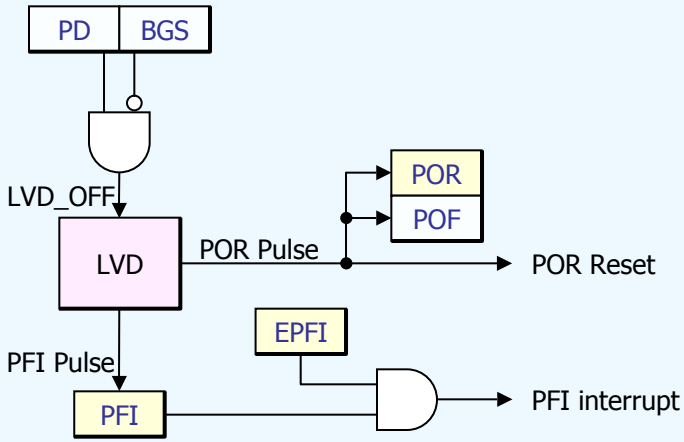
SMOD1	SMOD0	-	POF	GF1	GF0	PD	IDL
R/W(0)	R/W(0)		R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- POF : Power-off Flag
- PD : Power-down mode bit

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

WDMOD	POR	EPFI	PFI	WDIF	WTRF	EWT	RWT
R/W(0)	R/W(1)	R/W(0)	R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- POR : Power-on Reset Flag
- EPFI : Enable Power-fail Interrupt
- PFI : Power-Fail interrupt Flag (always 1 @ 3V operation)



6.8. WDT (Watchdog Timer)

- Detects software upset due to external noise or other causes
- Allows an automatic recovery using WDT interrupt

✓ **CKCON** (8Eh) : Clock Control Register

WD2	WD1	WD0	T2M	T1M	T0M	-	-
R/W(1)	R/W(1)	R/W(1)	R/W(0)	R/W(0)	R/W(0)		

- WD[2:0] : WDT Timer Count mode

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

WDMOD	POR	EPFI	PFI	WDIF	WTRF	EWT	RWT
R/W(0)	R/W(1)	R/W(0)	R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

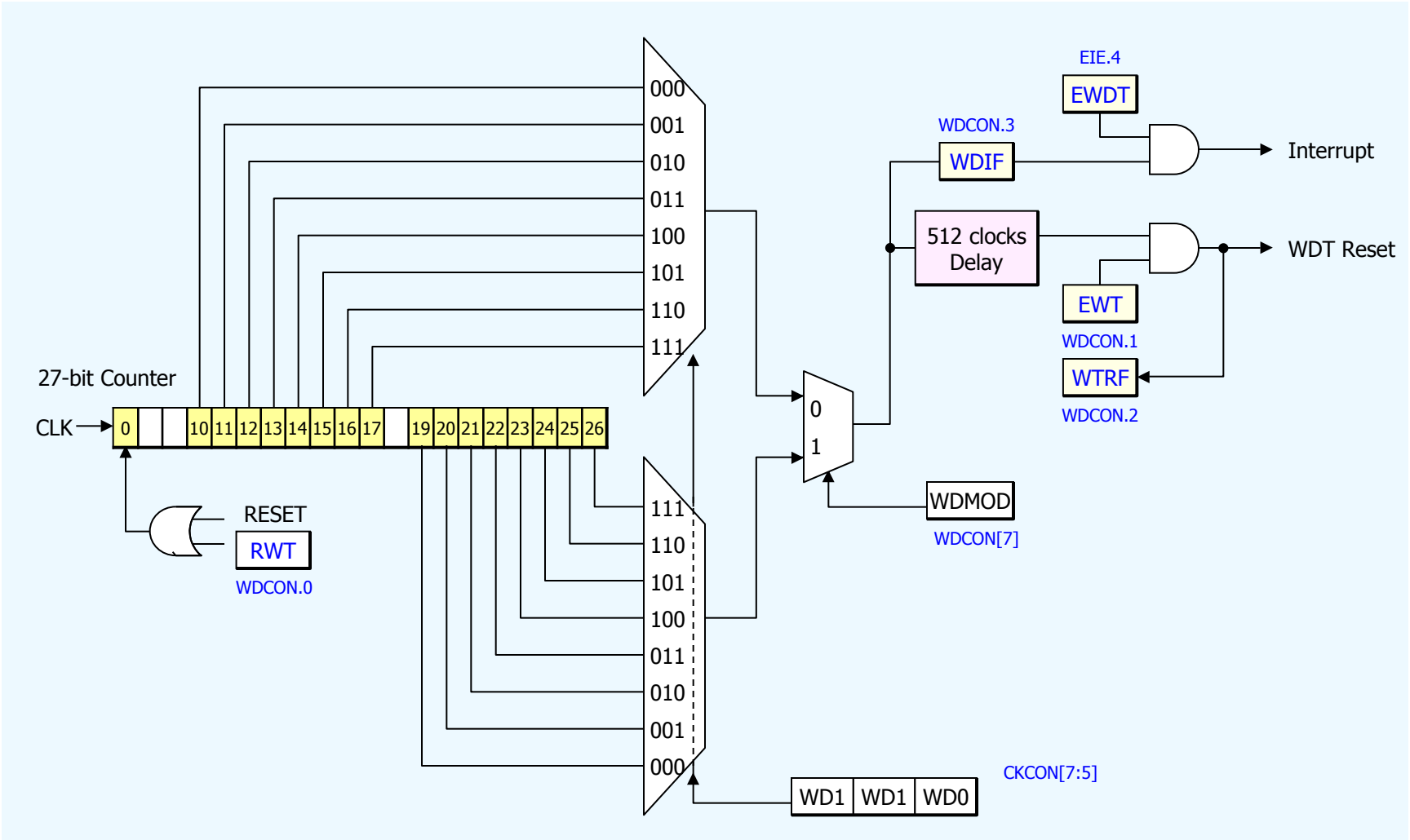
- WDMOD : WDT mode selection Flag
- POR : Power-on Reset Flag
- EPFI : Power-fail Interrupt Enable
- PFI : Power-Fail interrupt Flag
- WDIF : Watchdog Timer Interrupt Flag
- WTRF : Watchdog Timer Reset Flag. Only cleared by S/W.
- EWT : Watchdog Timer Reset Enable
- RWT : Restart Watchdog Timer

- Watchdog Time-out Values
  - ✓ Default : WD[2:0] = [1,1,1]

WDMOD=0			Interrupt Time-out (@12MHz)		Reset Time-out (@12MHz)	WDMOD=1			Interrupt Time-out (@12MHz)		Reset Time-out (@12MHz)
WD2	WD1	WD0				WD2	WD1	WD0			
0	0	0	2^{10} clocks	about 85 us	2^{10} + 512 clocks	0	0	0	2^{19} clocks	43.69 ms	2^{19} + 512 clocks
0	0	1	2^{11} clocks	170 us	2^{11} + 512 clocks	0	0	1	2^{20} clocks	87.38 ms	2^{20} + 512 clocks
0	1	0	2^{12} clocks	341 us	2^{12} + 512 clocks	0	1	0	2^{21} clocks	174.76 ms	2^{21} + 512 clocks
0	1	1	2^{13} clocks	682 us	2^{13} + 512 clocks	0	1	1	2^{22} clocks	349.52 ms	2^{22} + 512 clocks
1	0	0	2^{14} clocks	1.365 ms	2^{14} + 512 clocks	1	0	0	2^{23} clocks	699.05 ms	2^{23} + 512 clocks
1	0	1	2^{15} clocks	2.73 ms	2^{15} + 512 clocks	1	0	1	2^{24} clocks	1.398 s	2^{24} + 512 clocks
1	1	0	2^{16} clocks	5.46 ms	2^{16} + 512 clocks	1	1	0	2^{25} clocks	2.796 s	2^{25} + 512 clocks
1	1	1	2^{17} clocks	10.92 ms	2^{17} + 512 clocks	1	1	1	2^{26} clocks	5.592 s	2^{26} + 512 clocks

6.8. WDT (Watchdog Timer)

- Block Diagram



6.9. Timer/Counter : Timer 0/1

- Compatible with traditional 80C52 Timer/Counter
- Time base is selectable by S/W : 4 clocks or 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	13-bit T/C	16-bit T/C	8-bit T/C with automatic reload (TL1 ← TH1)	Halt

✓ **TMOD** (89h) : Timer/Counter 0/1 Mode Control Register

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

- Timer[1]: GATE[7], C/T[6], M1:M0[5:4]
- Timer[0]: GATE[3], C/T[2], M1:M0[1:0]
- GATE : 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 : Counter or Timer Selector. Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from Tx input pin).
- M1, M0 : Mode Selector bits
  - [0 0] Mode 0. 13-bit T/C.
  - [0 1] Mode 1. 16-bit T/C.
  - [1 0] Mode 2. 8-bit Auto-Reload T/C.
  - [1 1] Mode 3. (Timer 1) stopped, (Timer 0) TL0: 8-bit T/C controlled by the Timer 0 control bits. TH0: 8-bit T/C controlled by the Timer 1 control bits.

✓ **CKCON** (8Eh) : Clock Control Register

WD2	WD1	WD0	T2M	T1M	T0M	-	-
R/W(1)	R/W(1)	R/W(1)	R/W(0)	R/W(0)	R/W(0)		

- T1M : Timer 1 Clock Time-base Selection
  T1M=1, Time-base is 4 clocks not 12clocks.
- T0M : Timer 0 Clock Time-base Selection
  T0M=1, Time-base is 4 clocks not 12clocks.

✓ **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 Control
- TF0 : Timer 0 Overflow Flag
- TR0 : Timer 0 Run Control
- IE1 : External Interrupt 1 Flag
- IT1 : External Interrupt 1 Type Select
  Edge Detect (IT1=1). Level Detect (IT1=0)
- IE0 : External Interrupt 0 Flag
- IT0 : External Interrupt 0 Type Select
  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

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

TH0.7	TH0.6	TH0.5	TH0.4	TH0.3	TH0.2	TH0.1	TH0.0

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

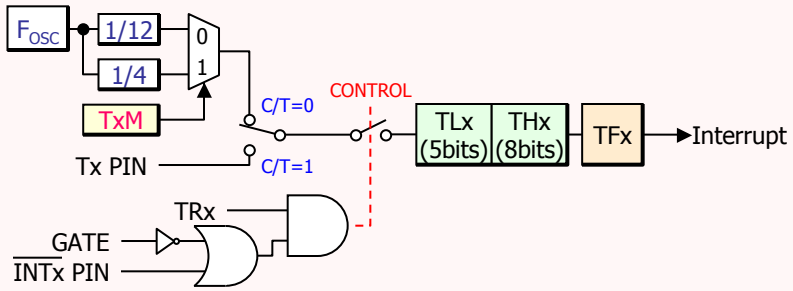
TL1.7	TL1.6	TL1.5	TL1.4	TL1.3	TL1.2	TL1.1	TL1.0

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

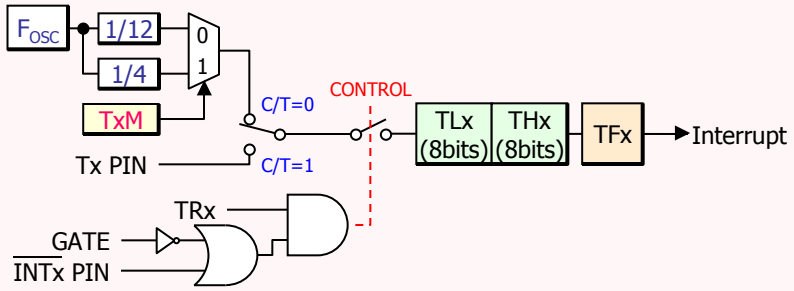
TH1.7	TH1.6	TH1.5	TH1.4	TH1.3	TH1.2	TH1.1	TH1.0

6.9. Timer/Counter : Timer 0/1 Mode Description

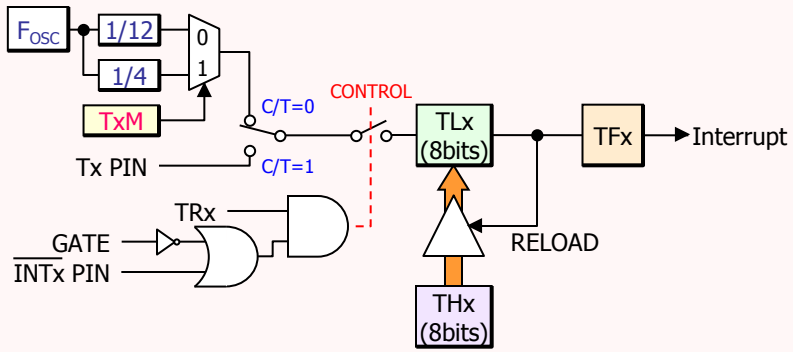
* Default : $F_{OSC}/12$ (T0M and T1m is each 0.)



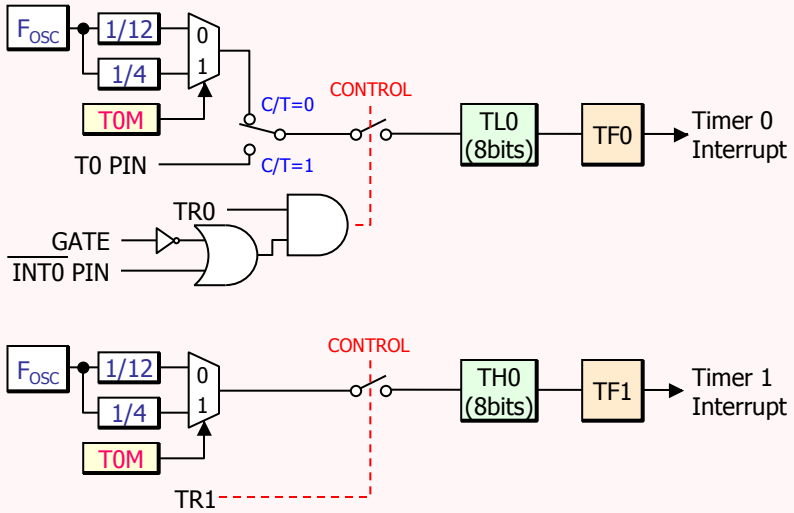
[Mode 0]



[Mode 1]



[Mode 2]



[Mode 3(Timer 0 only)]

6.9. Timer/Counter : Timer 2

- Compatible with traditional 80C52 Timer/Counter 2 function
- Up or down counting selectable by a software
- Time base is selectable by S/W : 4 clocks or 12 clocks

1. 16-bit Auto-reload [RCLK+TCLK=0, CP/RL2=0, T2OE=0]	16-bit Timer/Counter With Automatic Reload (TH2, TL2 ← RCAP2H, RCAP2L)
2. 16-bit Capture [RCLK+TCLK=0, CP/RL2=1, T2OE=0]	16-bit Timer/Counter with Capture (RCAP2H, RCAP2L ← TH2, TL2)
3. Baudrate Generator [RCLK+TCLK=1, CP/RL2=X, T2OE=X]	Baudrate Generation * Timer 2 Interrupt Disable
4. Programmable Clock Out [RCLK+TCLK=X, CP/RL2=0, T2OE=1]	Clock-out on P1.0

✓ **T2CON** (C8h) : Timer 2 Control Register

TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- TF2 : Timer 2 Overflow Flag
- EXF2 : Timer 2 External Flag
- RCLK : Receive Clock Flag
- TCLK : Transmit Clock Flag
- EXEN2 : Timer 2 External Enable Flag
- TR2 : Timer 2 Run Control
- C/T2 : Timer or Counter Selection. If C/T2=0, Timer Operation.
- CP/RL2 : Capture/Reload Flag.
  CP/RL2=0, Reload. (TH2,TL2) ← (RCAP2H, RCAP2L)
  CP/RL2=1, Capture. (RCAP2H, RCAP2L) ← (TH2,TL2)

✓ **CKCON** (8Eh) : Clock Control Register

WD2	WD1	WD0	T2M	T1M	T0M	-	-
R/W(1)	R/W(1)	R/W(1)	R/W(0)	R/W(0)	R/W(0)		

- T2M : Timer 2 Clock Time-base Selection
  T2M=1, Time-base is 4 clocks not 12clocks.

✓ **T2MOD** (C9h) : Timer 2 Mode Register

-	-	-	-	-	-	T2OE	DCEN
						R/W(0)	R/W(0)

- T2OE : Timer 2 Clock Output to P1.0
- DCEN : Timer 2 Down Count Enable

✓ **TL2** (CCh) : Timer 2 Low Byte Register

TL2.7	TL2.6	TL2.5	TL2.4	TL2.3	TL2.2	TL2.1	TL2.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

✓ **TH2** (CDh) : Timer 2 High Byte Register

TH2.7	TH2.6	TH2.5	TH2.4	TH2.3	TH2.2	TH2.1	TH2.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

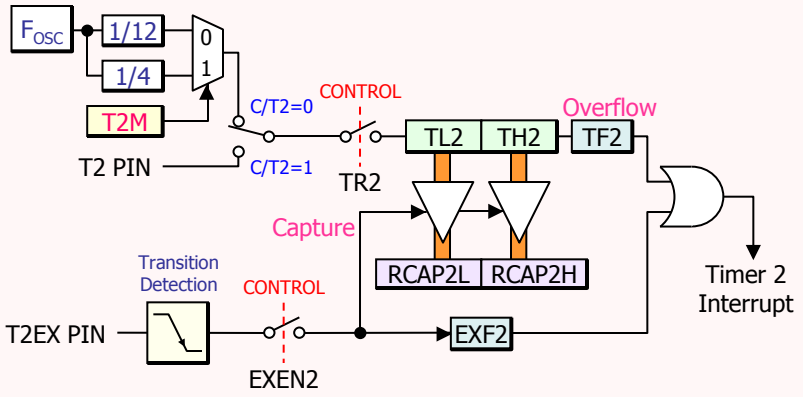
✓ **RCAP2L** (CAh) : Timer 2 Capture/Reload Low Byte Register

RCAP2L.7	RCAP2L.6	RCAP2L.5	RCAP2L.4	RCAP2L.3	RCAP2L.2	RCAP2L.1	RCAP2L.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)

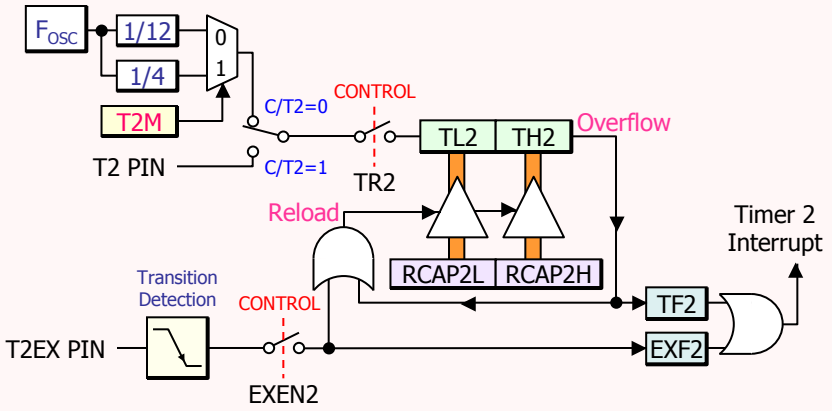
✓ **RCAP2H** (CBh) : Timer 2 Capture/Reload High Byte Register

RCAP2H.7	RCAP2H.6	RCAP2H.5	RCAP2H.4	RCAP2H.3	RCAP2H.2	RCAP2H.1	RCAP2H.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

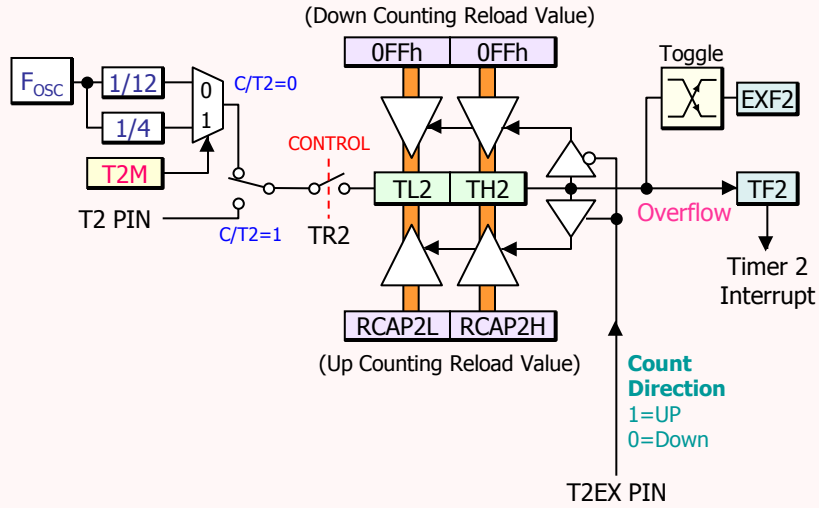
6.9. Timer/Counter : Timer 2 Mode Description



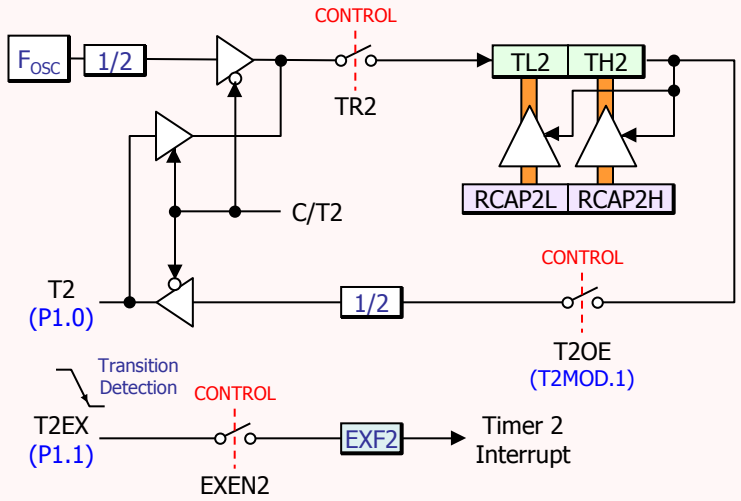
[Capture Mode]



[Auto Reload Mode (DCEN=0)]

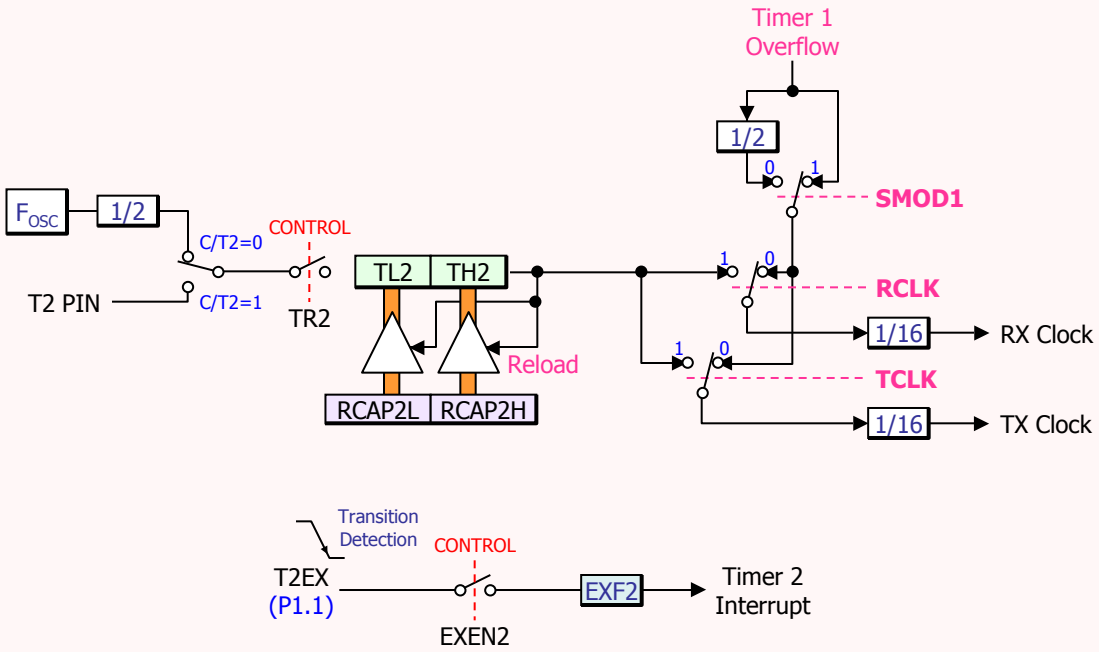


[Auto Reload Mode (DCEN=1)]



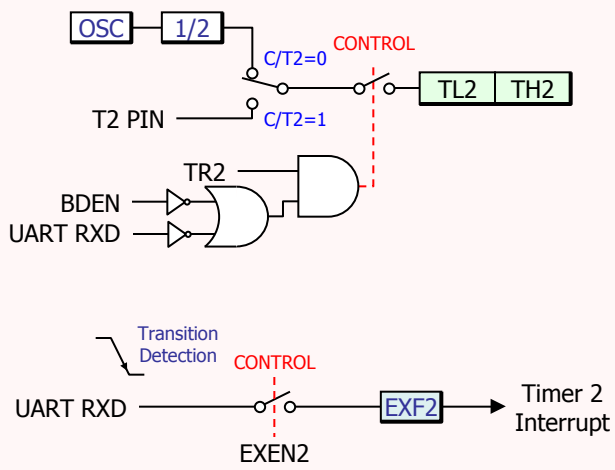
[Clock-Out Mode]

6.9. Timer/Counter : Timer 2 Mode Description



[Baudrate Generator Mode]

6.9. Timer/Counter : Timer 2 Mode Description



[Baud Rate Detector Mode (BDEN = 1)]

- Support for Baud Rate Detection (T2MOD_BDEN is set)
  - ✓ Counter runs when TR2 is set and the RXD input of UART is low.
  - ✓ Counter runs with OSC/2 like Baud Rate Generation.
  - ✓ EXF2 is set at falling edge of RXD if EXEN2 is set.

6.10. UART

- Function-level compatible with traditional 80C52 UART.
- Automatic address recognition : Multiprocessor communication.

	Data Size		Baudrate
Mode 0	8 bits	8 data bits	1/4 x Oscillator Clock
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) 1/16 x Timer 2 Overflow Rate
Mode 2	11 bits	Start bit(0) 8 data bit Programmable bit Stop bit(1)	1/32 x Oscillator Clock (SMOD1=0) 1/16 x Oscillator Clock (SMOD1=1)
Mode 3	11 bits	Start bit(0) 8 data bit Programmable bit Stop bit(1)	1/32 x Timer 1 Overflow (SMOD1=0) 1/16 x Timer 1 Overflow (SMOD1=1) 1/16 x Timer 2 Overflow Rate

✓ **The Timer 1 Overflow varies with CKCON register.**
➔ 12 clocks time-base or 4 clocks time-base.

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

SMOD1	SDMO0	-	POF	GF1	GF0	PD	IDL
R/W(0)	R(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, 2, and 3
- SMOD0 : Enable SM0 access. Don't modify this bit.

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

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- SM0, SM1 : Serial Port Operating Mode Selection
  [0,0] : Mode 0. 8-bit Shift Register (OSC/4)
  [0,1] : Mode 1. 8-bit UART (Variable)
  [1,0] : Mode 2. 9-bit UART (OSC/32 or OSC/16)
  [1,1] : Mode 3. 9-bit UART (Variable)
- SM2 : Enable the Automatic Address Recognition in Mode 2 and 3.
  Cleared after receiving the address.
  In Mode 1, the validity of a Stop Bit is checked if SM2=1.
  In Mode 0, SM2 should be 0.
- REN : Enable/Disable Reception.
- TB8 : 9th data bit that will be transmitted in Mode 2 and 3.
- RB8 : 9th data bit that was received in Mode 2 and 3.
  In Mode 1, RB8 is equal to Stop Bit if SM2=0.
  In Mode 0, RB8 is not used.
- 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)

- The transmission buffer and the reception buffer are separated.
- The transmission/reception buffers have the same address.

6.10. UART : Baudrate Example

- Serial Port Operating Mode 0

$$\text{Baudrate} = \frac{\text{Oscillator Frequency}}{4}$$

- Serial Port Operating Mode 2

$$\text{Baudrate} = \frac{2^{SMOD1}}{32} \times \text{Oscillator Frequency}$$

(PCON.7 ←)

- Serial Port Operating Mode 1, 3
  - ✓ Using Timer 1 Overflow

$$\text{Baudrate} = \frac{2^{SMOD1}}{32} \times \text{Timer 1 overflow}$$

  - ✓ Using Timer 2 Overflow

$$\text{Baudrate} = \frac{\text{Timer 2 overflow}}{16}$$

EX) Using Timer 1 to Generate Baudrates

$$\text{Mode 1 \& 3 Baudrate} = \frac{2^{SMOD1}}{32} \times F_{OSC} \times \frac{3^{T1M}}{12} \times \frac{1}{[\,256 - (TH1)\,]}$$

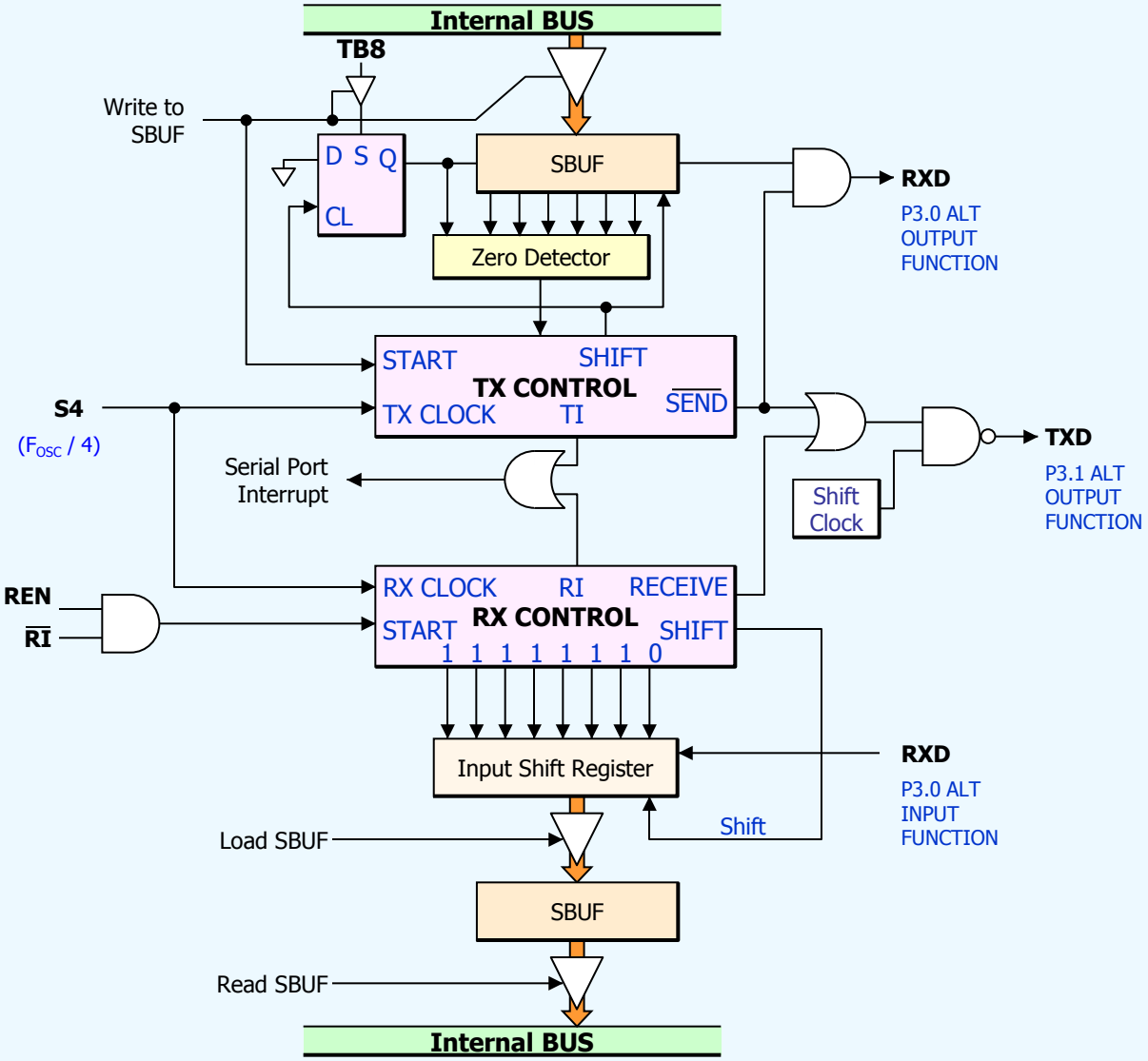
- If SMOD1(PCON.7) = 1 ➔ Double Buadrate
- If T1M(CKCON.3) = 0 ➔ F_{OSC} / 12
- If T1M(CKCON.3) = 1 ➔ F_{OSC} / 4

EX) Using Timer 2 to Generate Baudrates

$$\text{Mode 1 \& 3 Baudrate} = \frac{1}{32} \times F_{OSC} \times \frac{1}{[65536 - (RCAPH,RCAPL)\,]}$$

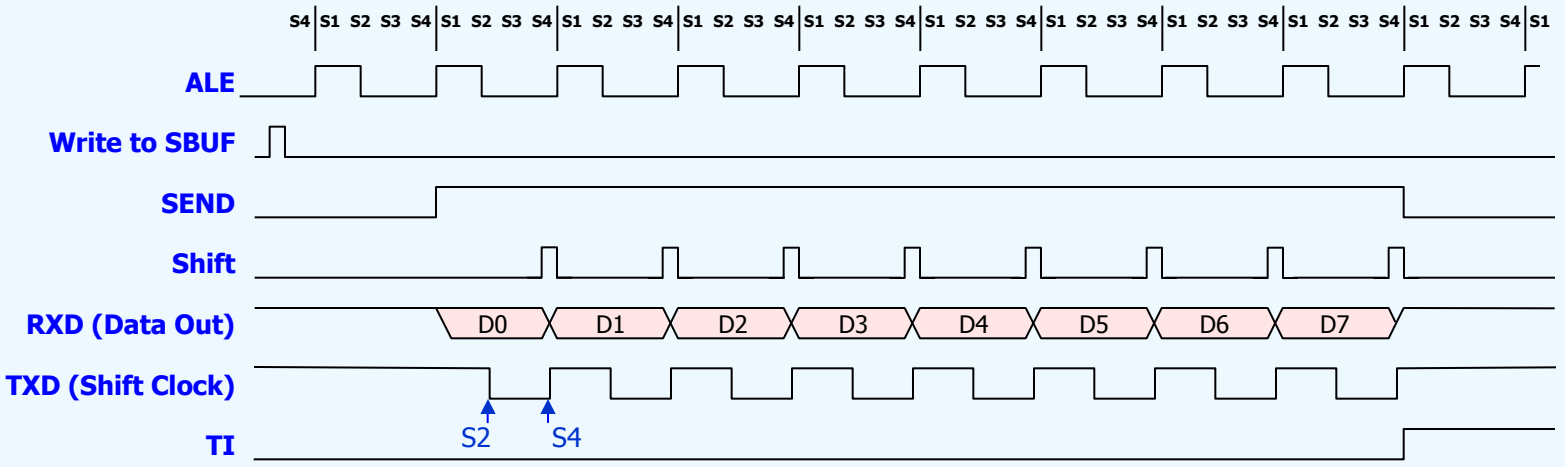
Baudrate		UART Mode	F_{OSC}	SMOD1	Timer 1		
T1M=0	T1M=1				C/T	Mode	Reload Value (TH1)
Max : 3 MHz	Max : 3 MHz	Mode 0	12 MHz	X	X	X	X
Max : 750 KHz	Max : 750 KHz	Mode 2	12 MHz	1	X	X	X
62.5 KHz	187.5 KHz	Mode 1 & 3	12 MHz	1	0	2	FFh
19.2 KHz	57.6 KHz		11.0592 MHz	1	0	2	FDh
9.6 KHz	28.8 KHz		11.0592 MHz	0	0	2	FDh
4.8 KHz	14.4 KHz		11.0592 MHz	0	0	2	FAh
2.4 KHz	7.2 KHz		11.0592 MHz	0	0	2	F4h
1.2 KHz	3.6 KHz		11.0592 MHz	0	0	2	E8h
137.5 Hz	412.5 Hz		11.0592 MHz	0	0	2	1Dh
110 Hz	330 Hz		6 MHz	0	0	2	72h
110 Hz	330 Hz		12 MHz	0	0	1	FEEBh

6.10. UART : Mode 0, Functional Diagram

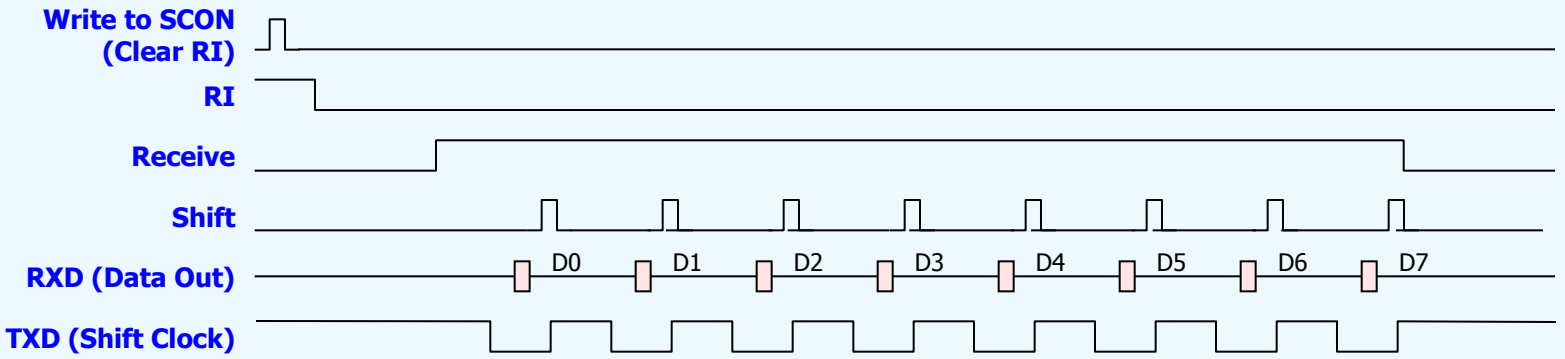


6.10. UART : Mode 0, Timing Diagram

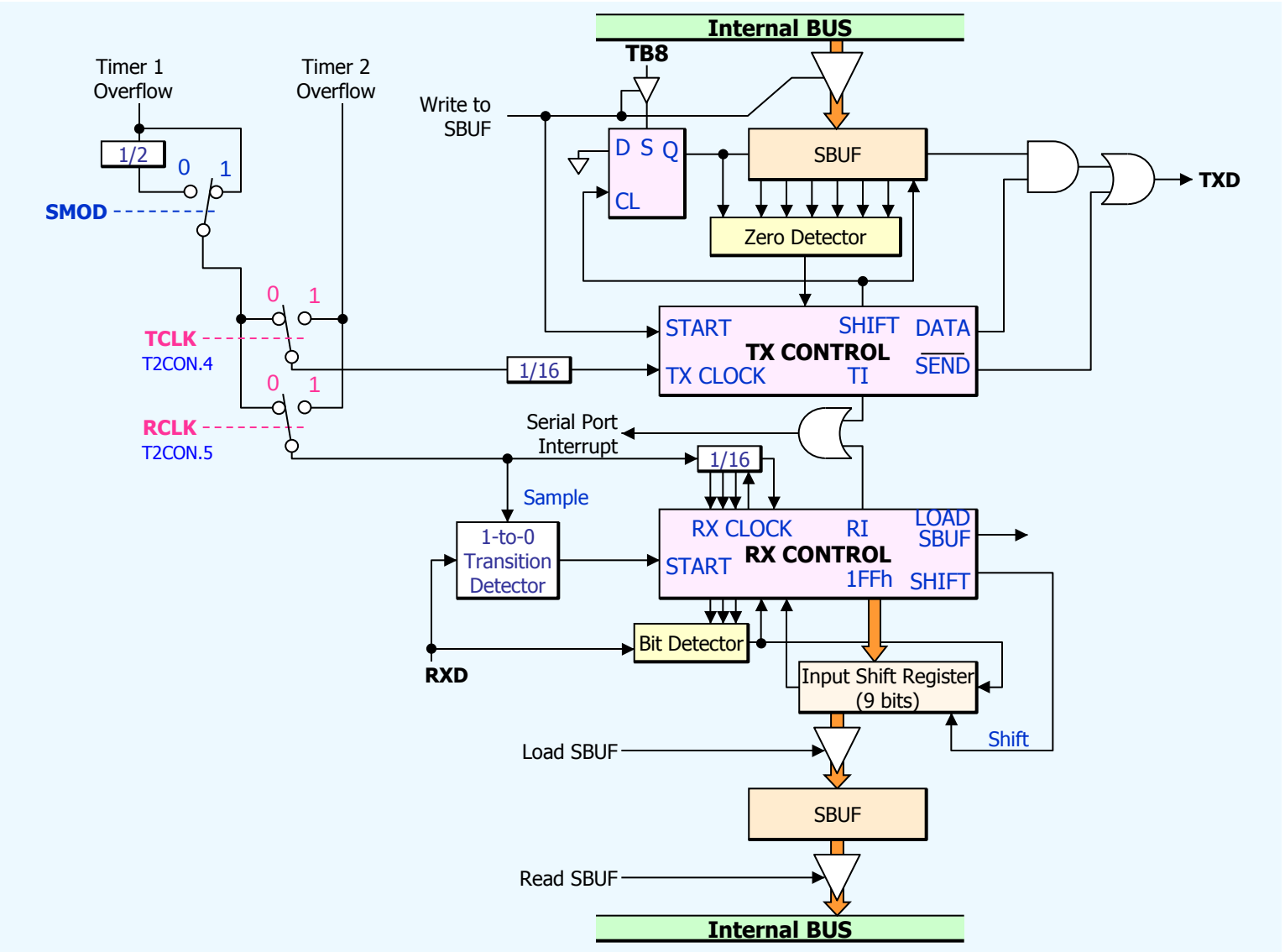
[Transmit]



[Receive]

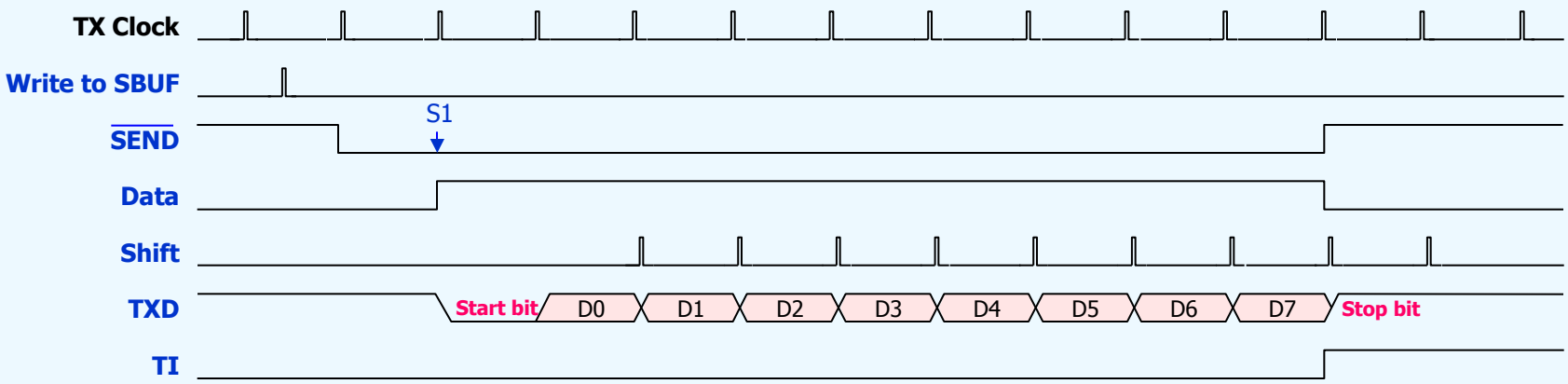


6.10. UART : Mode 1, Functional Diagram

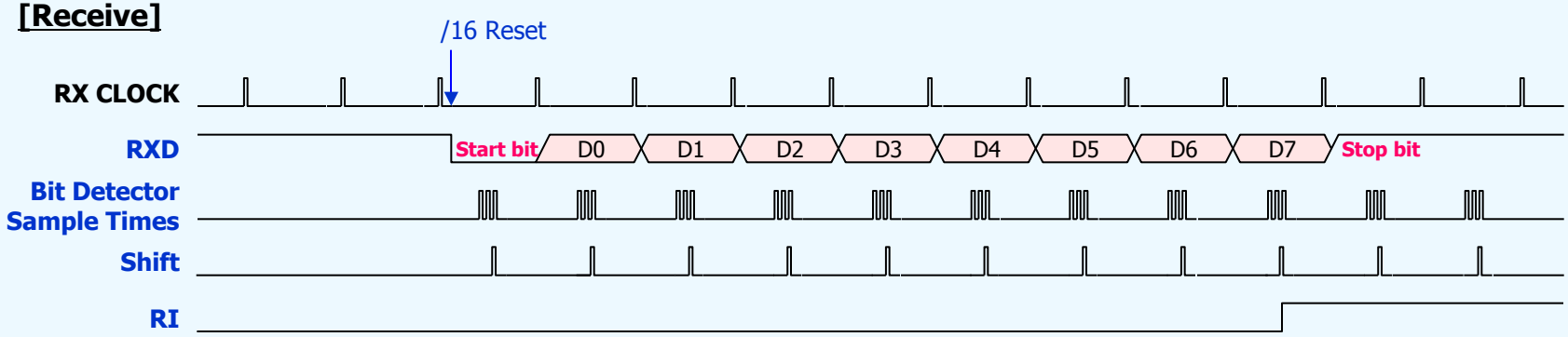


6.10. UART : Mode 1, Timing Diagram

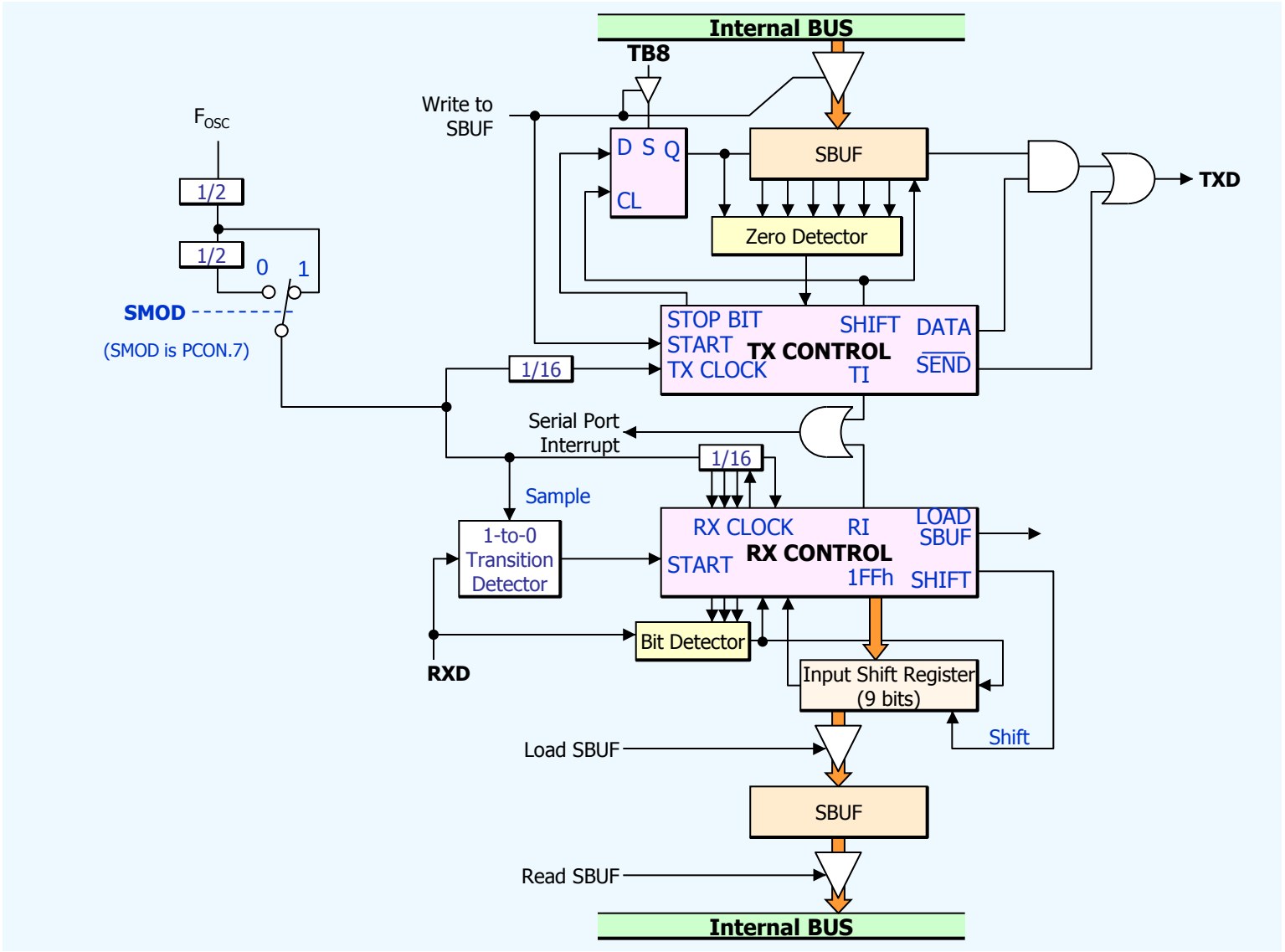
[Transmit]



[Receive]

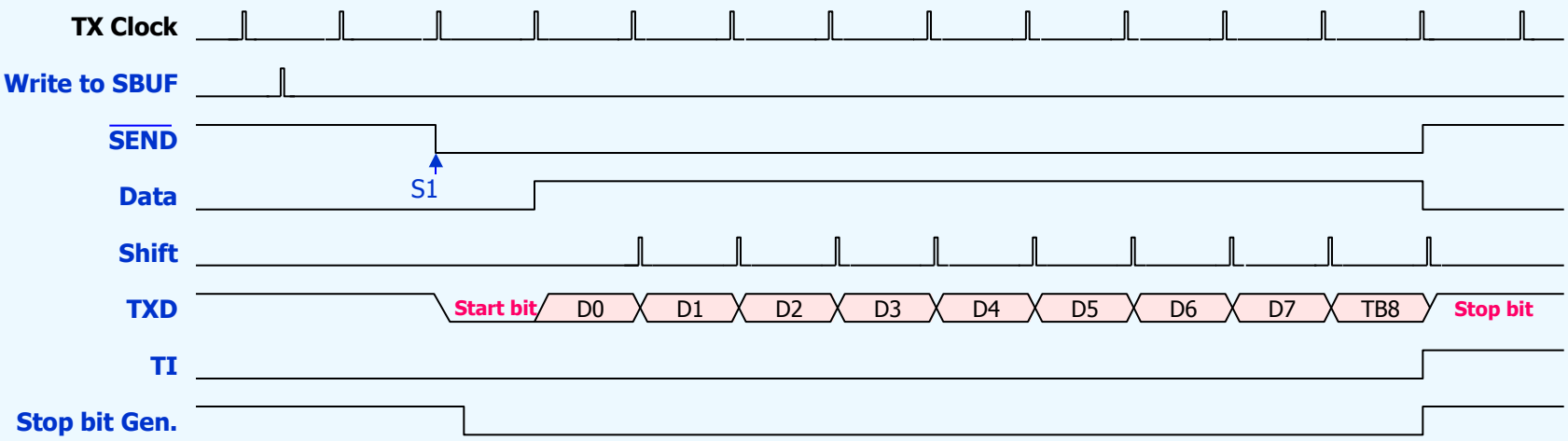


6.10. UART : Mode 2, Functional Diagram

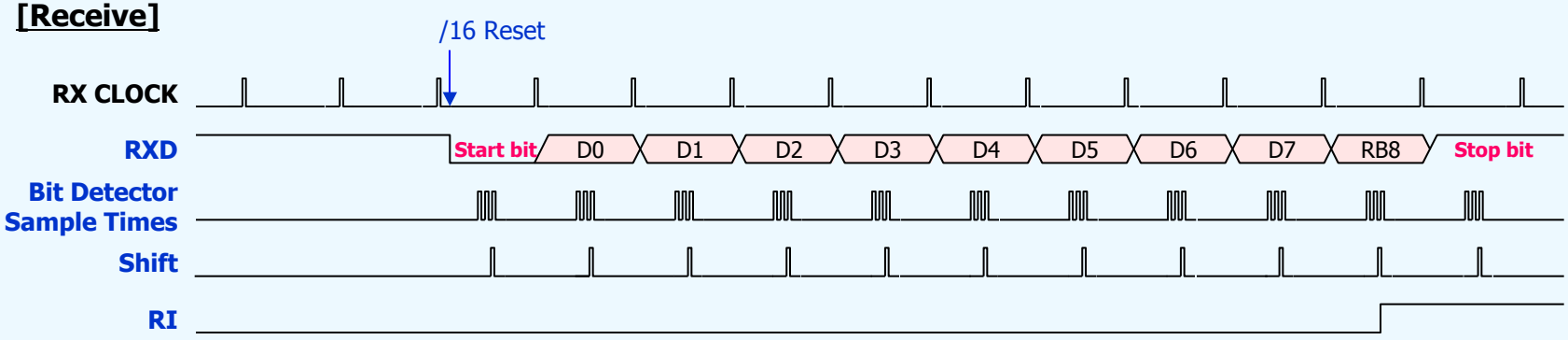


6.10. UART : Mode 2, Timing Diagram

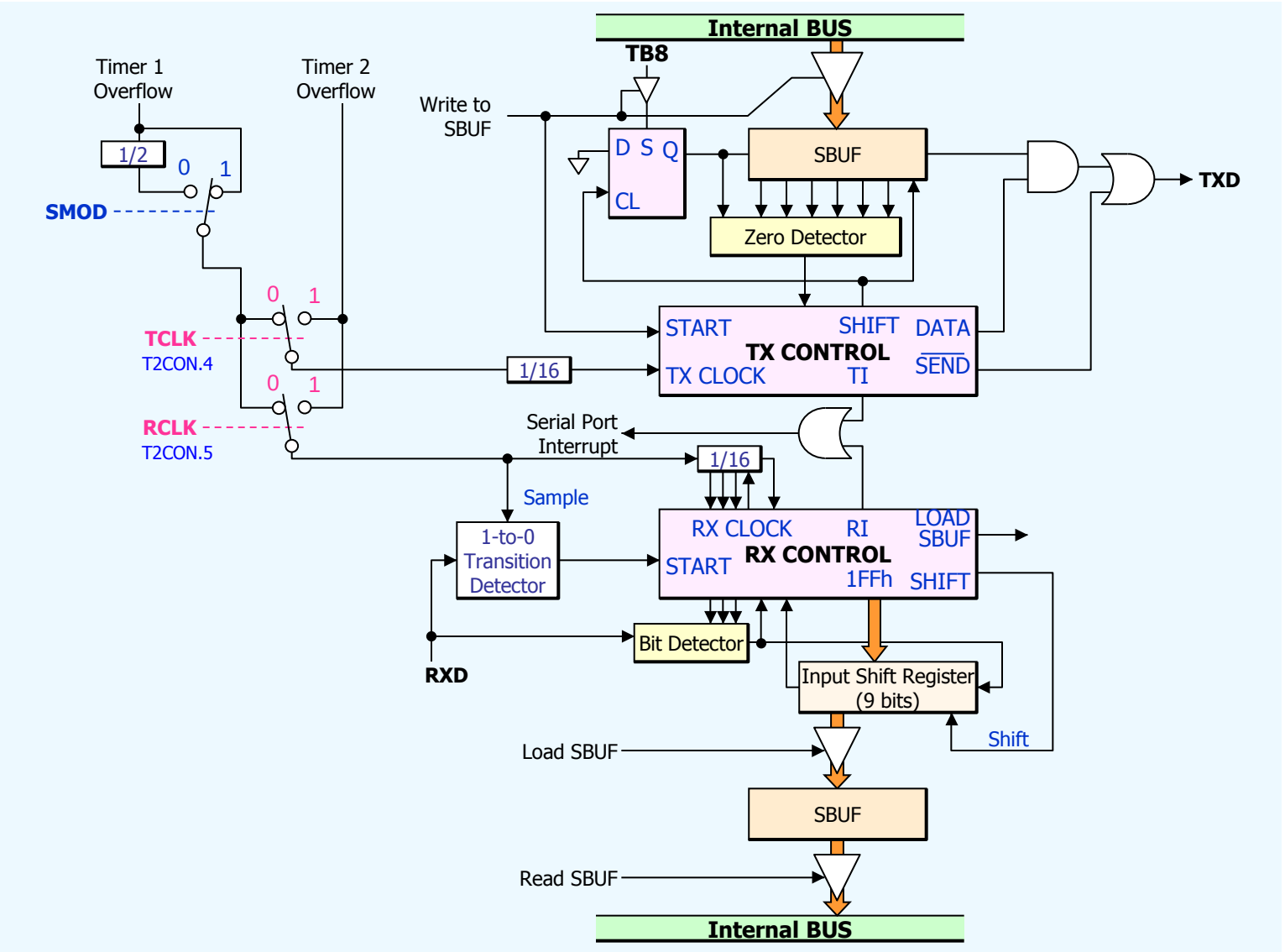
[Transmit]



[Receive]

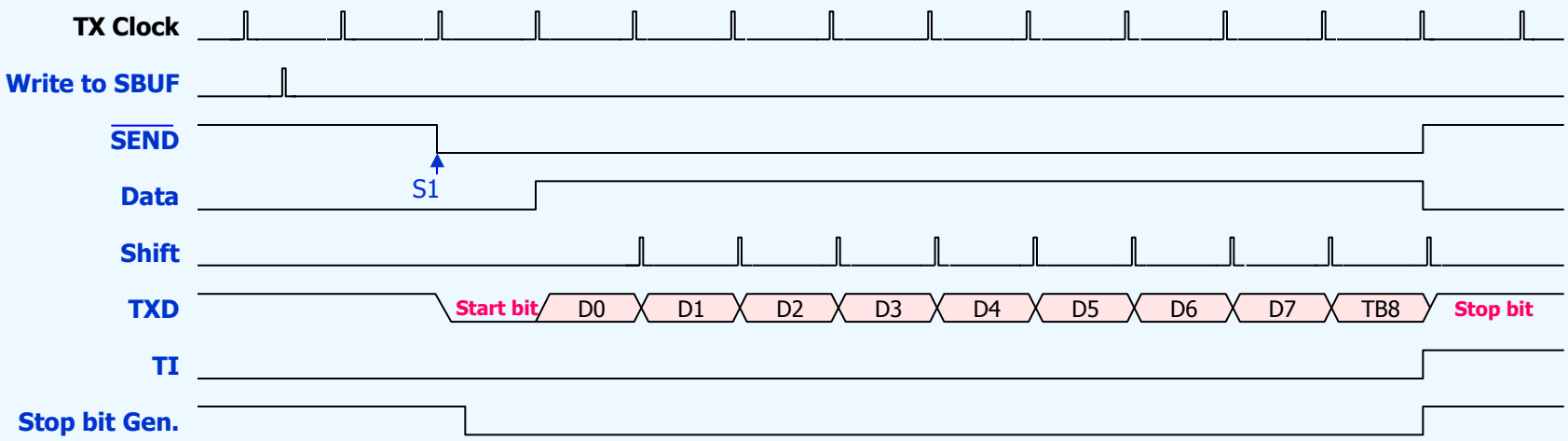


6.10. UART : Mode 3, Functional Diagram

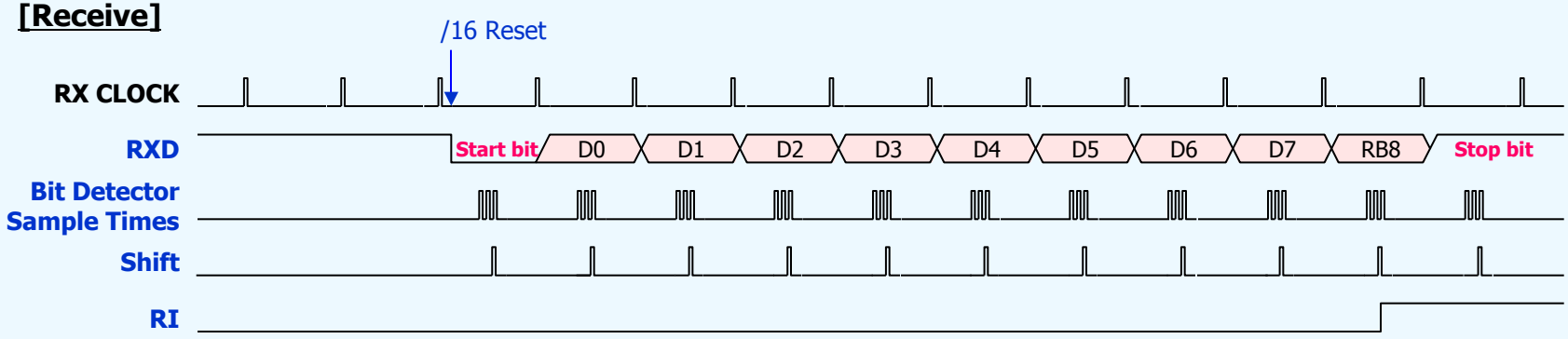


6.10. UART : Mode 3, Timing Diagram

[Transmit]

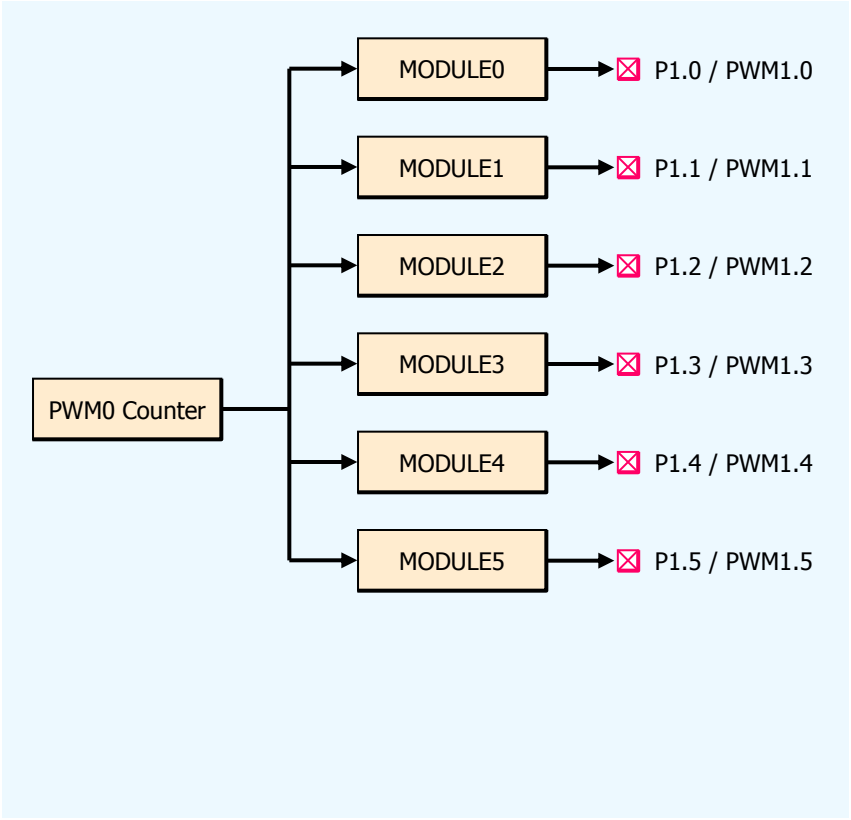


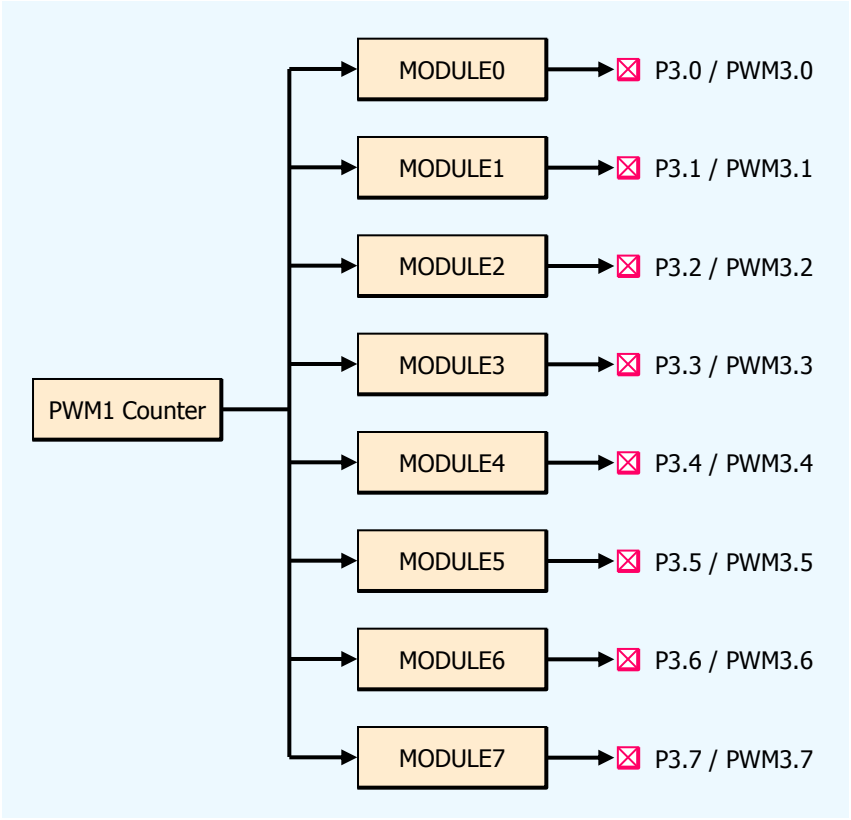
[Receive]



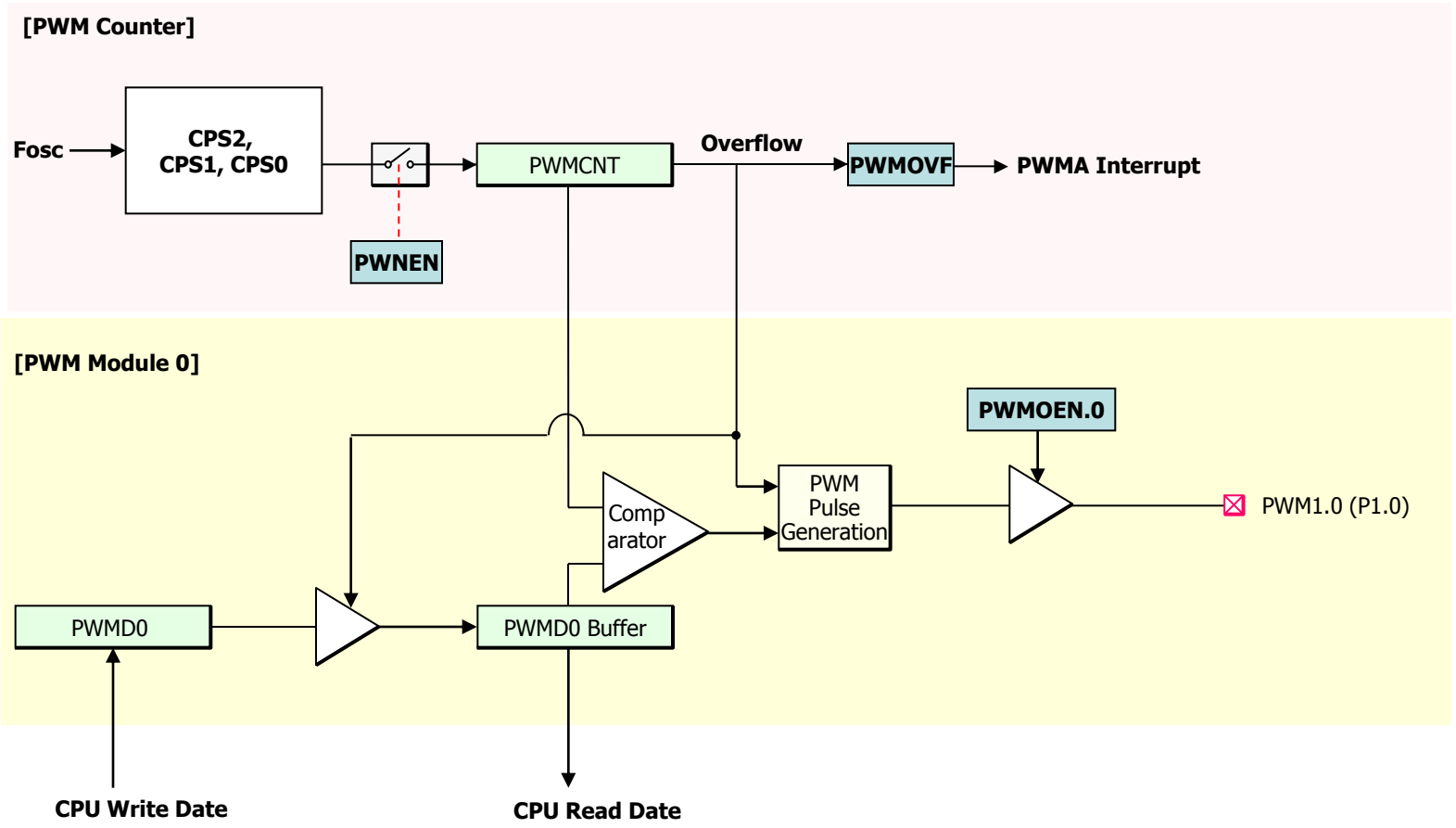
6.11. PWMA (PWM Arrays)

- PWMA
  - ✓ Two 8-bit PWM generation with 6 & 8 modules (Compatible to M1.0A 8-bit mode)
  - ✓ PWM Data buffer Update (8-bit Counter Overflow Update)
  - ✓ PWM Counter can be cleared by S/W.
  - ✓ PWM is stopped or started (resumed) by S/W.





6.11. PWMA : Block Diagram



6.11. PWMA : PWMA0 SFR

✓ **PWM0CON** (92h) : PWMA CH0 Control Register

-	CPS2	CPS1	CPS0	-	-	PWMOVF	PWMEN
	R/W(0)	R/W(0)	R/W(0)			R/W(0)	R/W(0)

- CPS2, CPS1, CPS0 : PWMA counter frequency selection.
  [0,0,0] = F_{OSC} / 1 ; Default
  [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
- PWMOVF : PWMA counter overflow flag.
  Set by hardware and cleared by software.
  PWMOVF flags an interrupt.
- PWMEN : PWMA counter run control bit.
  [0] = Stop the PWMA counter.
  [1] = Run the PWMA counter.

✓ **PWM0CNT** (93h) : PWMA CH0 Counter Register

CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- Software can write this register for the initialization of the counter.

✓ **PWM0OEN** (9Bh) : PWMA CH0 Module Output Enable

-	-	OE5	OE4	OE3	OE2	OE1	OE0
		R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- OE5 : Module 5 PWM output enable
- OE4 : Module 4 PWM output enable
- OE3 : Module 3 PWM output enable
- OE2 : Module 2 PWM output enable
- OE1 : Module 1 PWM output enable
- OE0 : Module 0 PWM output enable

6.11. PWMA : PWMA0 SFR (Cont'd)

✓ **PWM0D0** (94h) : PWMA CH0 Duty Data Register of Module 0

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)

- Each Module has a internal buffer register for the duty data register.
  The buffer register is updated with the new data
  whenever the PWMA counter rolls over.
  When user write, the data register is written.
  When user read, the contents of buffer register is read out.

✓ **PWM0D1** (95h) : PWMA CH0 Duty Data Register of Module 1
✓ **PWM0D2** (96h) : PWMA CH0 Duty Data Register of Module 2
✓ **PWM0D3** (97h) : PWMA CH0 Duty Data Register of Module 3
✓ **PWM0D4** (9Ch) : PWMA CH0 Duty Data Register of Module 4
✓ **PWM0D5** (9Dh) : PWMA CH0 Duty Data Register of Module 5
✓ **PWM0D6** (9Eh) : PWMA CH0 Duty Data Register of Module 6
✓ **PWM0D7** (9Fh) : PWMA CH0 Duty Data Register of Module 7

6.11. PWMA : PWMA1 SFR

✓ **PWM1CON** (A2h) : PWMA CH1 Control Register

-	CPS2	CPS1	CPS0	-	-	PWMOVF	PWMEN
	R/W(0)	R/W(0)	R/W(0)			R/W(0)	R/W(0)

- CPS2, CPS1, CPS0 : PWMA counter frequency selection.
  [0,0,0] = F_{OSC} / 1 ; Default
  [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
- PWMOVF : PWMA counter overflow flag.
  Set by hardware and cleared by software.
  PWMOVF flags an interrupt.
- PWMEN : PWMA counter run control bit.
  [0] = Stop the PWMA counter.
  [1] = Run the PWMA counter.

✓ **PWM1CNT** (A3h) : PWMA CH1 Counter Register

CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- Software can write this register for the initialization of the counter.

✓ **PWM1OEN** (ABh) : PWMA CH1 Module Output Enable

OE7	OE6	OE5	OE4	OE3	OE2	OE1	OE0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- OE7 : Module 7 PWM output enable
- OE6 : Module 6 PWM output enable
- OE5 : Module 5 PWM output enable
- OE4 : Module 4 PWM output enable
- OE3 : Module 3 PWM output enable
- OE2 : Module 2 PWM output enable
- OE1 : Module 1 PWM output enable
- OE0 : Module 0 PWM output enable

6.11. PWMA : PWMA1 SFR (Cont'd)

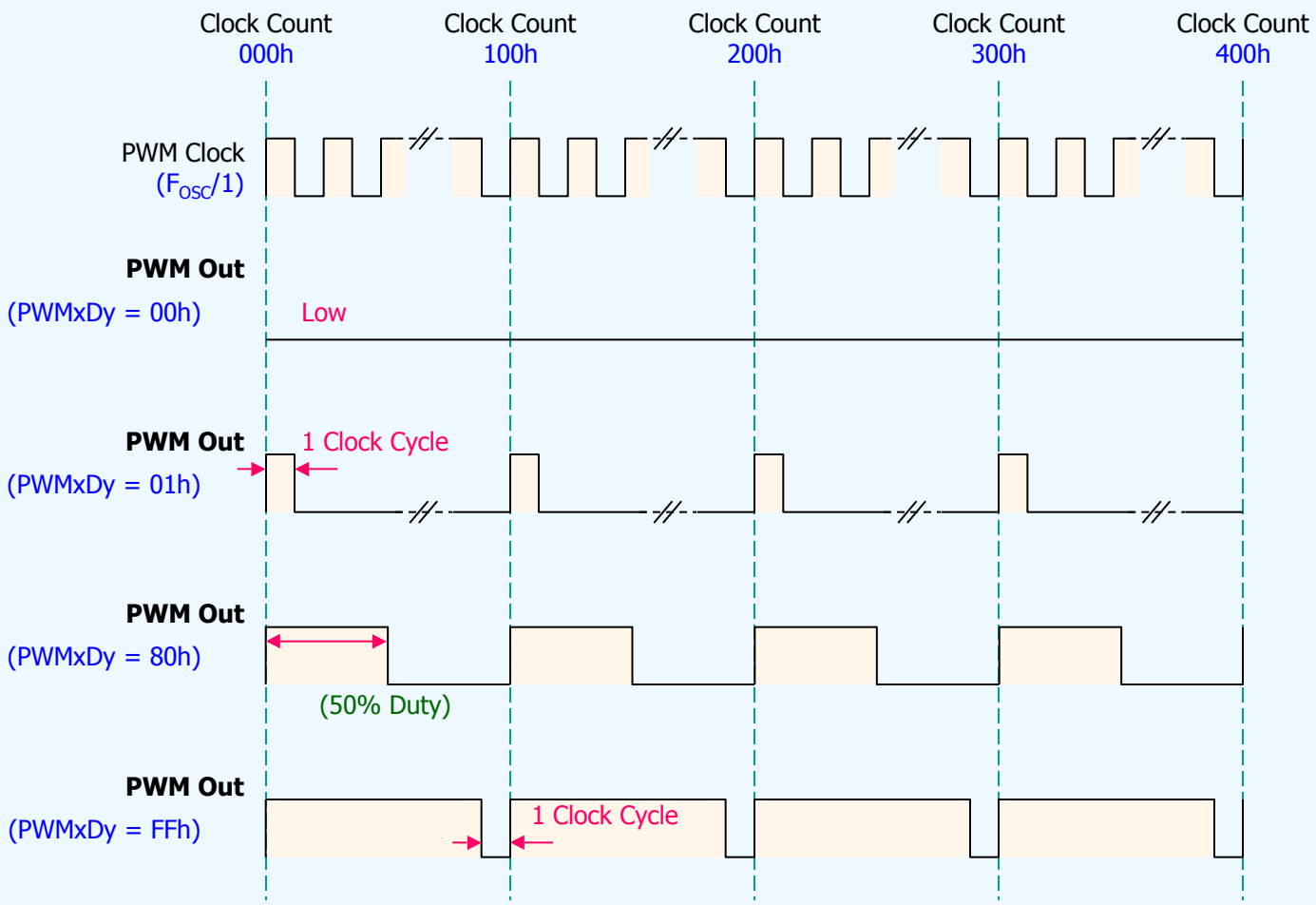
✓ **PWM1D0** (A4h) : PWMA CH1 Duty Data Register of Module 0

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)

- Each Module has a internal buffer register for the duty data register.
  The buffer register is updated with the new data
  whenever the PWMA counter rolls over.
  When user write, the data register is written.
  When user read, the contents of buffer register is read out.

✓ **PWM1D1** (A5h) : PWMA CH1 Duty Data Register of Module 1
✓ **PWM1D2** (A6h) : PWMA CH1 Duty Data Register of Module 2
✓ **PWM1D3** (A7h) : PWMA CH1 Duty Data Register of Module 3
✓ **PWM1D4** (ACh) : PWMA CH1 Duty Data Register of Module 4
✓ **PWM1D5** (ADh) : PWMA CH1 Duty Data Register of Module 5
✓ **PWM1D6** (AEh) : PWMA CH1 Duty Data Register of Module 6
✓ **PWM1D7** (AFh) : PWMA CH1 Duty Data Register of Module 7

6.11. PWMA : Pulse Generation Example



6.12. I2C : SFR

- Two-wire Interface
- 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** (A1h) : Extended Interrupt Enable Register

ESPI	-	EI2C	EWDT	-	-	EX3	EX2
R/W(0)		R/W(0)	R/W(0)			R/W(0)	R/W(0)

- EI2C : I2C Interrupt Enable

✓ **I2CSLA** (EBh) : 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.

✓ **I2CDAT** (ECh) : 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)

✓ **I2CCFG** (EAh) : 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)

✓ **I2CSCL** (EDh) : 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)

✓ **I2CCON** (E9h) : 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)

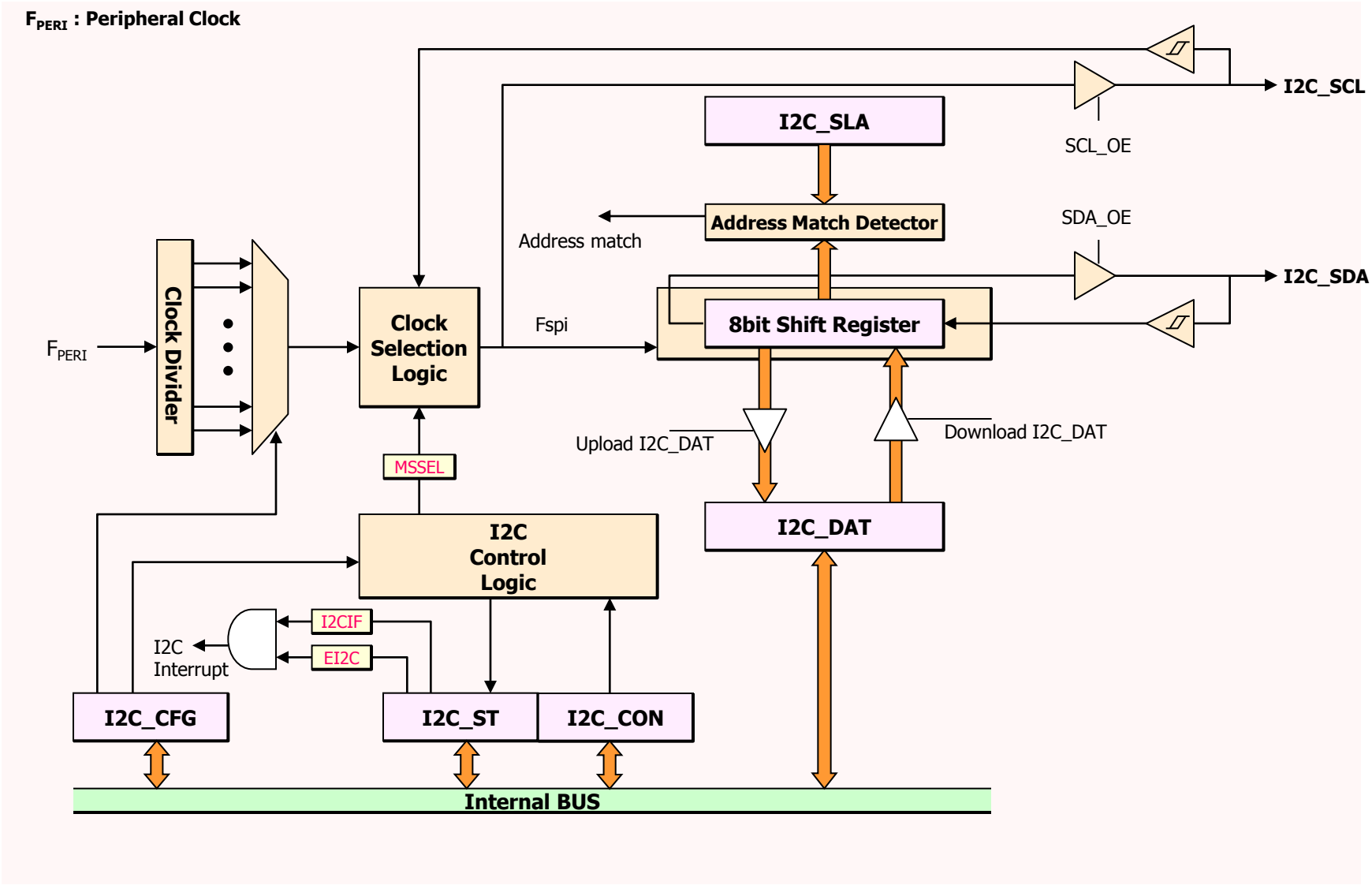
- SLA2ME : 2nd Byte Slave Address Match Enable in Slave mode
  [0] : 2nd Byte SLA Match Disable [1] : 2nd SLA Byte Match Enable
- SCLHD : Hold SCL 'low' for Wait State in Slave mode.
  [0] : Hold SCL 'low'. The flag is cleared automatically by H/W
  [1] : Release SCL 'float'. The flag is set by S/W
- 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

✓ **I2CST** (E8h) : 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)

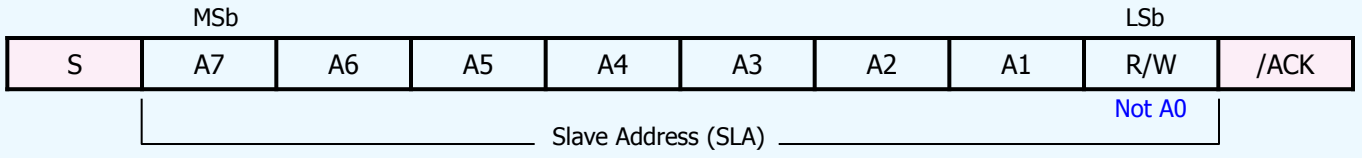
- I2CIF : I²C Master Interrupt Flag in slave & master mode.
  [0] : Idle [1] : Interrupt occurred.
  It is set each time a byte is received or transmitted.
  If SP_IE flag in I2C_CFG SFR is set, it is set at Start/Stop condition.
  The flag is set by H/W and cleared by S/W.
- I2COF : I2C Overflow Flag 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

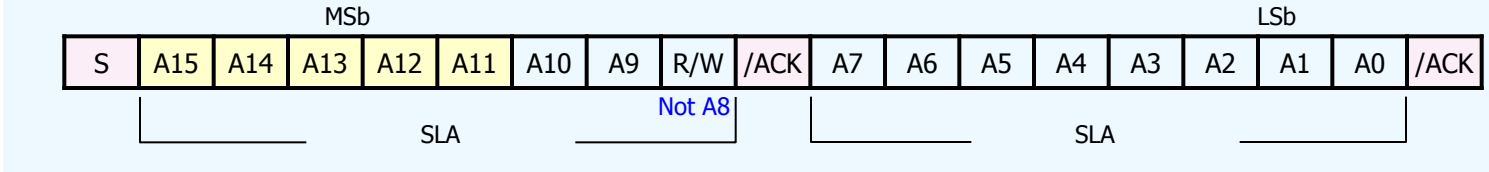


6.12. I2C : Overview

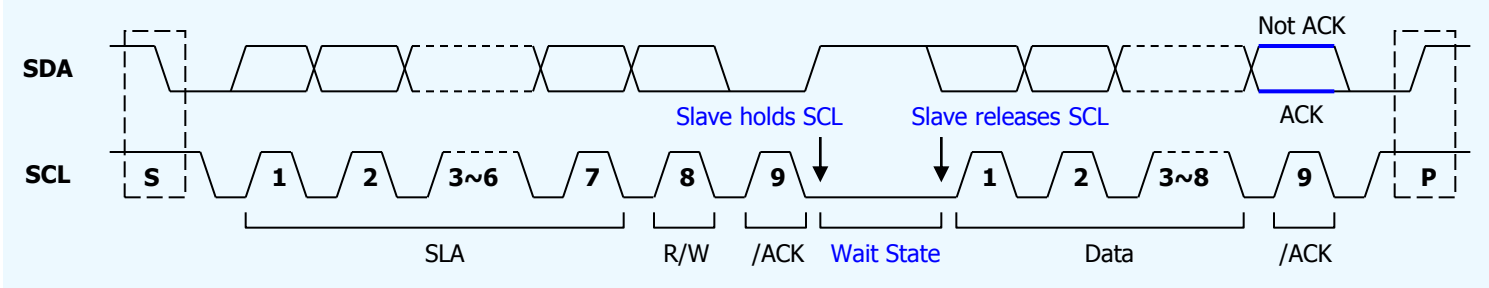
- Addressing I2C devices
  - ✓ 7-bit Address Format



  - ✓ 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.

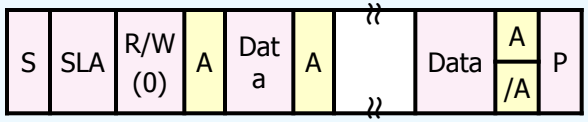


6.12. I2C : Overview

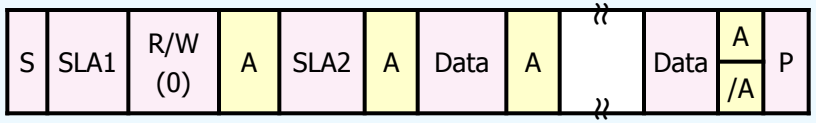
Master-Transmitter Sequence

From Master to Slave | From Slave to Master

[7-bit Address Mode]

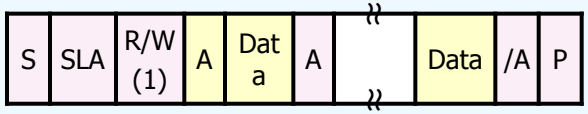


[10-bit / Extended 15-bit Address Mode]



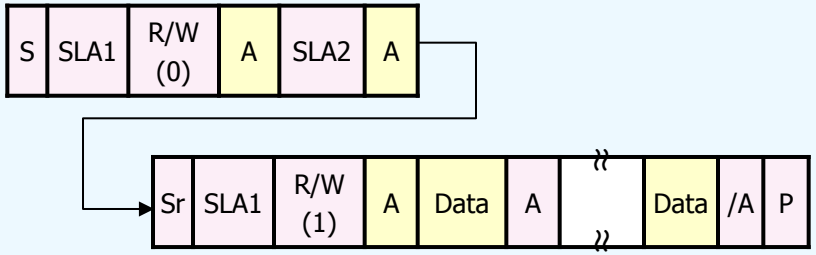
Master-Receiver Sequence

[7-bit Address Mode]



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

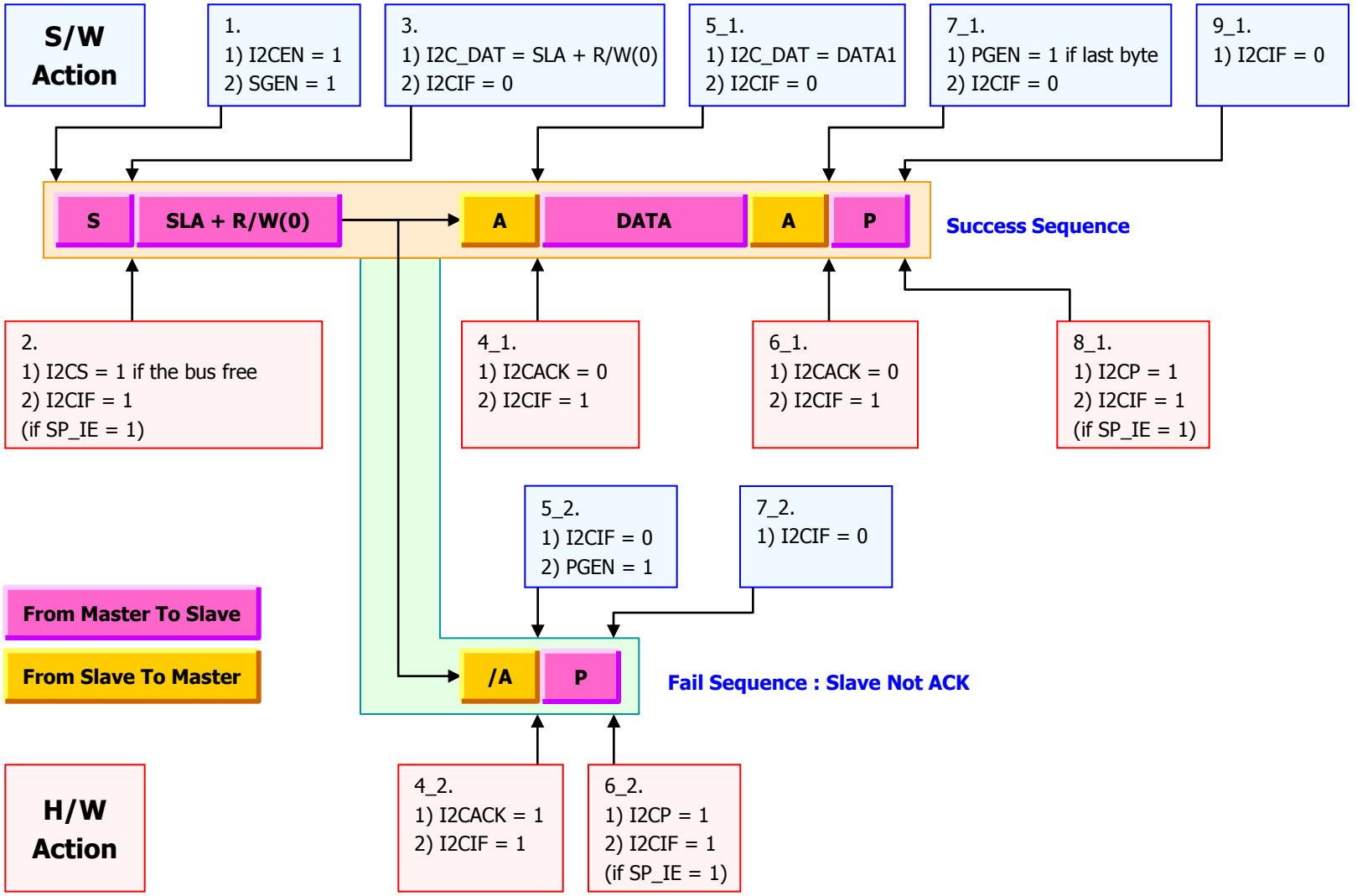
[10-bit / Extended 15-bit Address Mode]



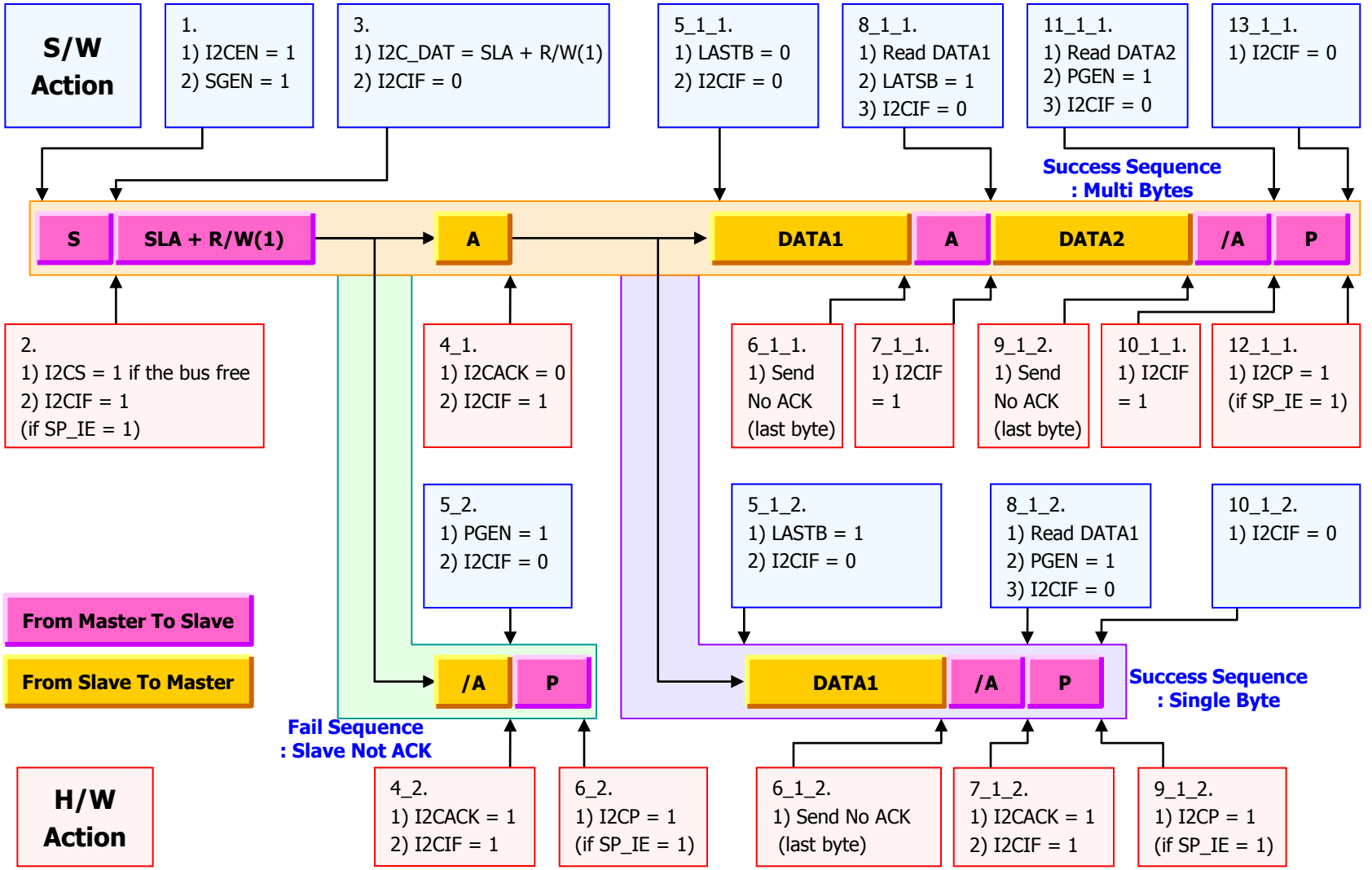
Combined Format

- ✓ 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.

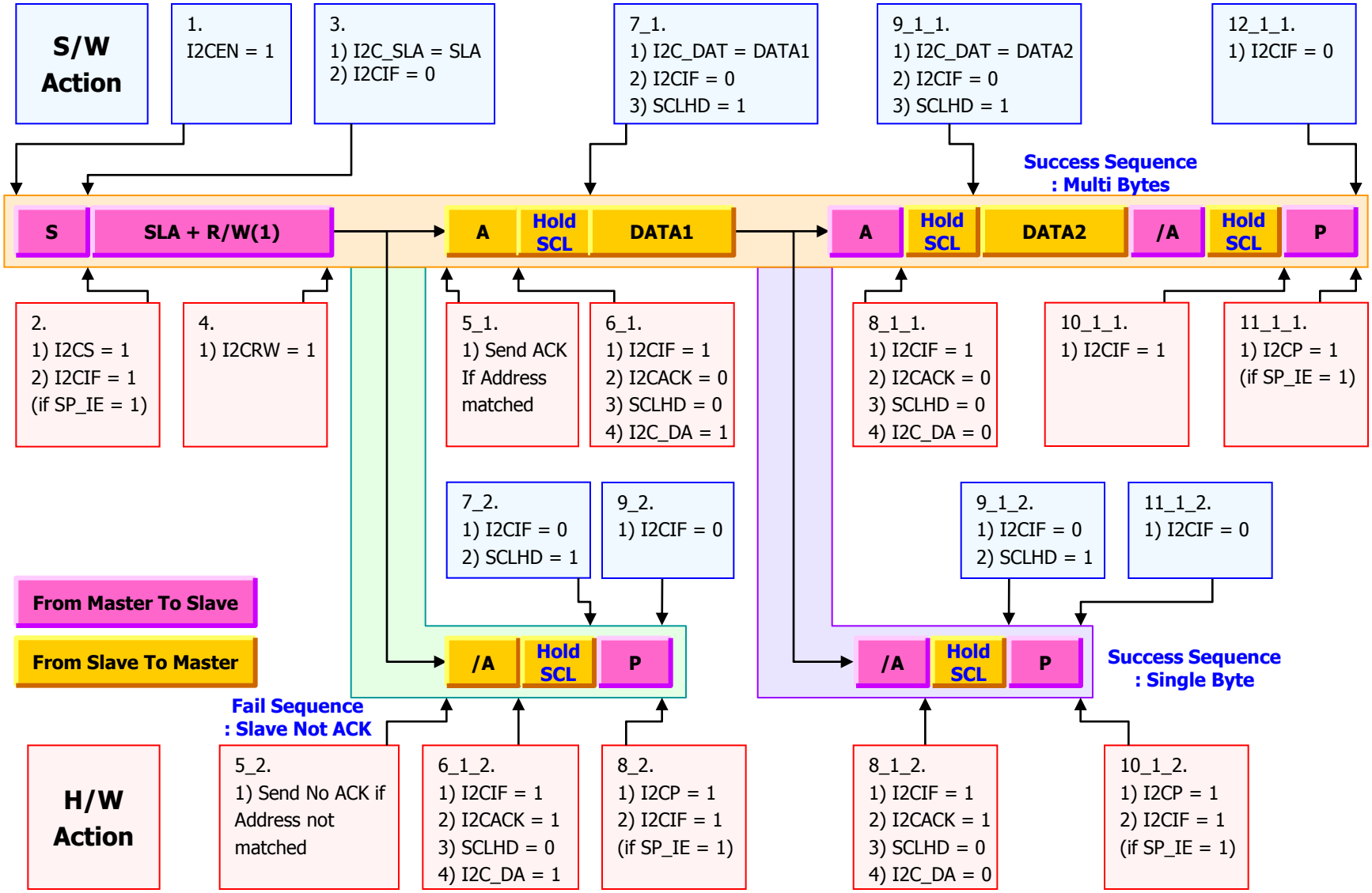
6.12. I2C : Master Transmitter Flow



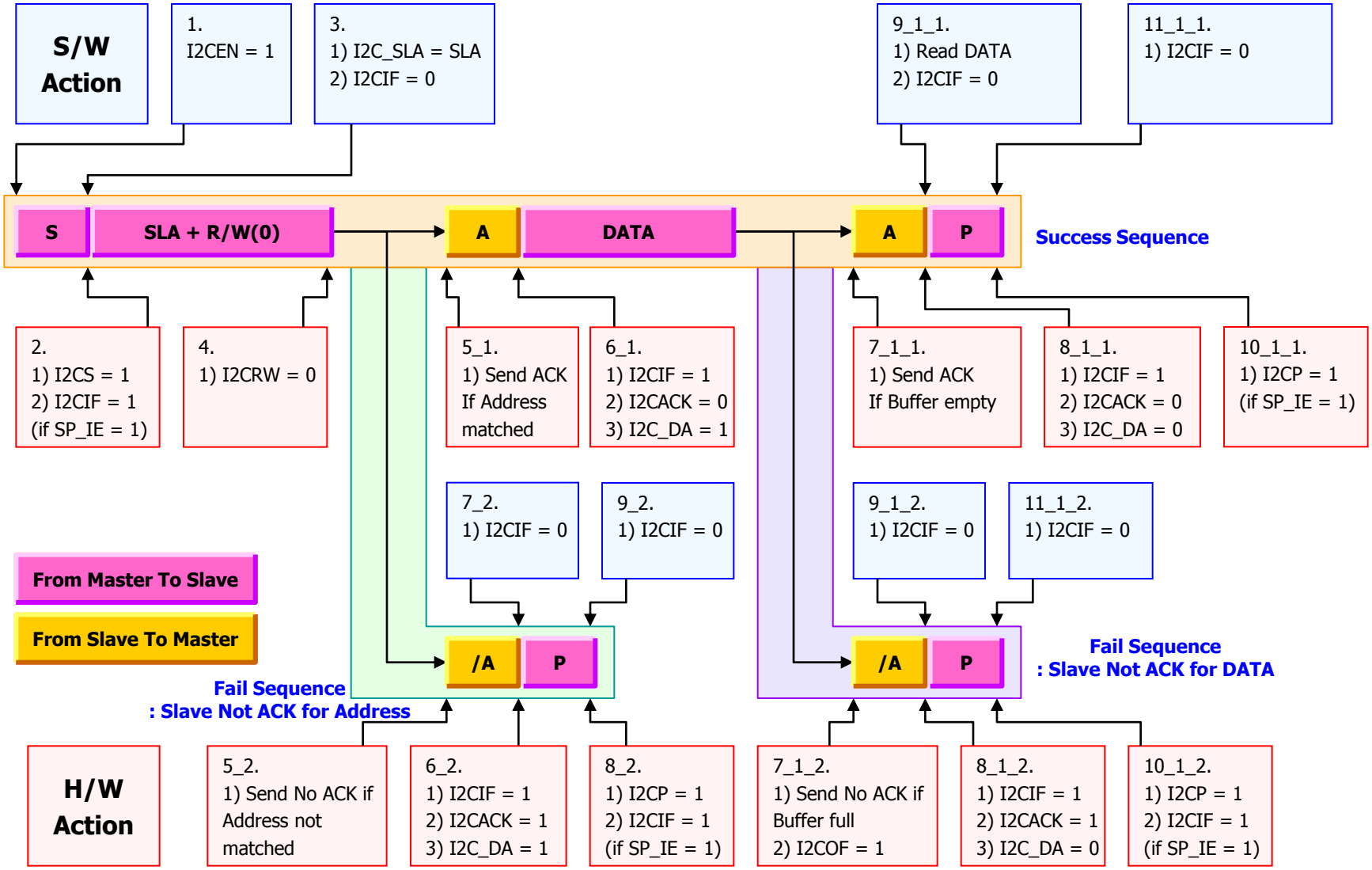
6.12. I2C : Master Receiver Flow



6.12. I2C : Slave Transmitter Flow



6.12. I2C : Slave Receiver Flow



6.12. I2C : Master Example

- I2C Master example code

```
I2CST    EQU  0E8H          ; I2CST SFR
I2CIF    EQU  0EFH          ; I2CST.7 Flag
I2COF    EQU  0EEH          ; I2CST.6 Flag
I2CACK   EQU  0EDH          ; I2CST.5 Flag
I2CRW    EQU  0ECH          ; I2CST.4 Flag
I2CDA    EQU  0EBH          ; I2CST.3 Flag
I2CP     EQU  0EAH          ; I2CST.2 Flag
I2CS     EQU  0E9H          ; I2CST.1 Flag
I2CBF    EQU  0E8H          ; I2CST.0 Flag

I2CCON   EQU  0E9H
I2CCFG   EQU  0EAH
I2CSLA   EQU  0EBH
I2CDAT   EQU  0ECH
I2CSCL   EQU  0EDH

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

- I2C Slave example code using interrupt

```
        I2CST     EQU  0E8H        ; I2CST SFR
        I2CIF     EQU  0EFH        ; I2CST.7 Flag
        I2COF     EQU  0EEH        ; I2CST.6 Flag
        I2CACK    EQU  0EDH        ; I2CST.5 Flag
        I2CRW     EQU  0ECH        ; I2CST.4 Flag
        I2CDA     EQU  0EBH        ; I2CST.3 Flag
        I2CP      EQU  0EAH        ; I2CST.2 Flag
        I2CS      EQU  0E9H        ; I2CST.1 Flag
        I2CBF     EQU  0E8H        ; I2CST.0 Flag

        I2CCON    EQU  0E9H
        I2CCFG    EQU  0EAH
        I2CSLA    EQU  0EBH
        I2CDAT    EQU  0ECH
        I2CSCL    EQU  0EDH

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. SPI : SFR

- Full-duplex, Three-wire Synchronous Data Transfer
- Master or Slave Operation
- LSB First or MSB First Data Transfer
- Eight Programmable Bit Rates
- Clock Polarity & Phase Selection
- Support Write Collision Protection
- Wake-up from IDLE mode

✓ **SPIDR** (B6h) : SPI TX / RX Data Register

DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

✓ **SPICON** (B4h) : SPI Control Register

-	MODE	BORD	MSSEL	CKPOL	CKPHA	SPIOEN	SPIEN
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- MODE : SPI mode selection
  [0] : 4-wire mode [1] : 3-wire mode
- BORD : SPI Transfer Bit Order
  [1] : First LSB, Last MSB [0] : First MSB, Last LSB
- MSSEL : SPI Master / Slave Selection Flag
  [1] : SPI Master Mode [0] : SPI Slave Mode
- CKPOL, CKPHA : SPI clock Polarity & Phase
  [0,0] : Leading edge Rising, Leading edge Sampling
  [0,1] : Leading edge Rising, Trailing edge Sampling
  [1,0] : Leading edge Falling, Leading edge Sampling
  [1,1] : Leading edge Falling, Trailing edge Sampling
- SPIOEN : SPI Output Enable
  [1] : SPI Output Enable [0] : SPI Output Disable
- SPIEN : SPI Enable Flag
  [1] : SPI Enable [0] : SPI Disable

✓ **SPIST** (C0h) : SPI Status Register

-	-	-	-	TXBV	SPIF	SPICOL	SPIOF
				R(0)	R/W(0)	R/W(0)	R/W(0)

- TXBV : TX buffer of SPIDR holds valid data.
  [1] : Set by H/W when user write SPIDR while SPI is enabled.
  [0] : Cleared by H/W when the data is moved to TX shift register or SPI is disabled.
- SPIF : SPI Interrupt Flag
  [1] : Serial transfer is complete. If SPIE is set and EA is set, SPI interrupt is generated.
- SPICOL : SPI Write Collision Flag
  [1] : SPIDR is written when TXBV is set. The previous data is lost.
- SPIOF : SPI Read Overflow Flag
  [1] : If a new data is received while SPIDR is still holding the previous data, the flag is set.
  SPIF must be cleared before receiving a data again.

✓ **SPICK** (B5h) : SPI Clock Control Register

-	-	-	-	-	SPICK2	SPICK1	SPICK0
					R/W(0)	R/W(0)	R/W(0)

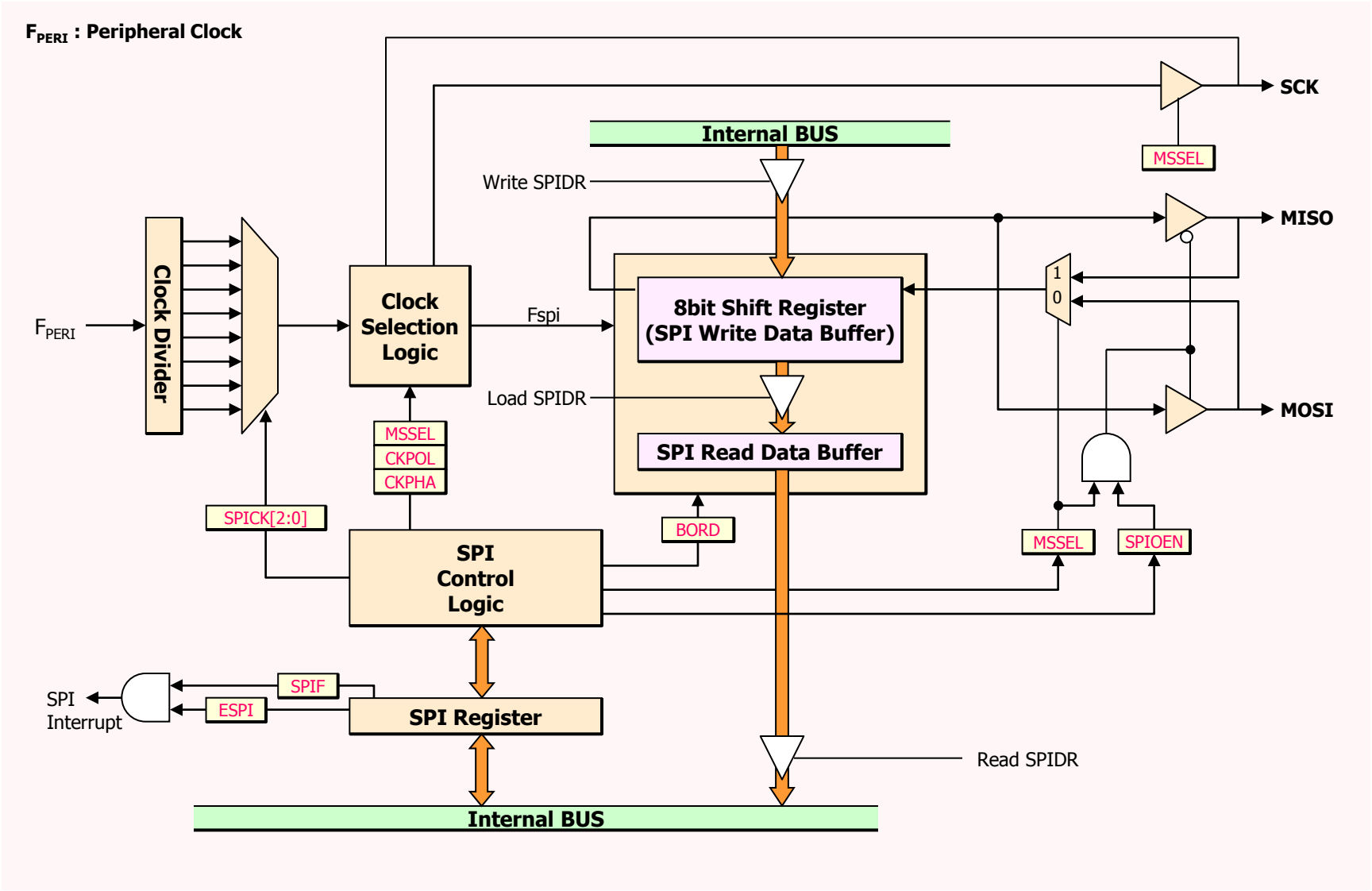
- SPICK[2:0] : SPI Master Clock Divider
  [0,0,0] : Fosc / 2 [0,0,1] : Fosc / 4
  [0,1,0] : Fosc / 8 [0,1,1] : Fosc / 16
  [1,0,0] : Fosc / 32 [1,0,1] : Fosc / 64
  [1,1,0] : Fosc / 128 [1,1,1] : Fosc / 256

✓ **EIE** (A1h) : Extended Interrupt Enable Register

ESPI	-	EI2C	EWDT	-	-	EX3	EX2
R/W(0)		R/W(0)	R/W(0)			R/W(0)	R/W(0)

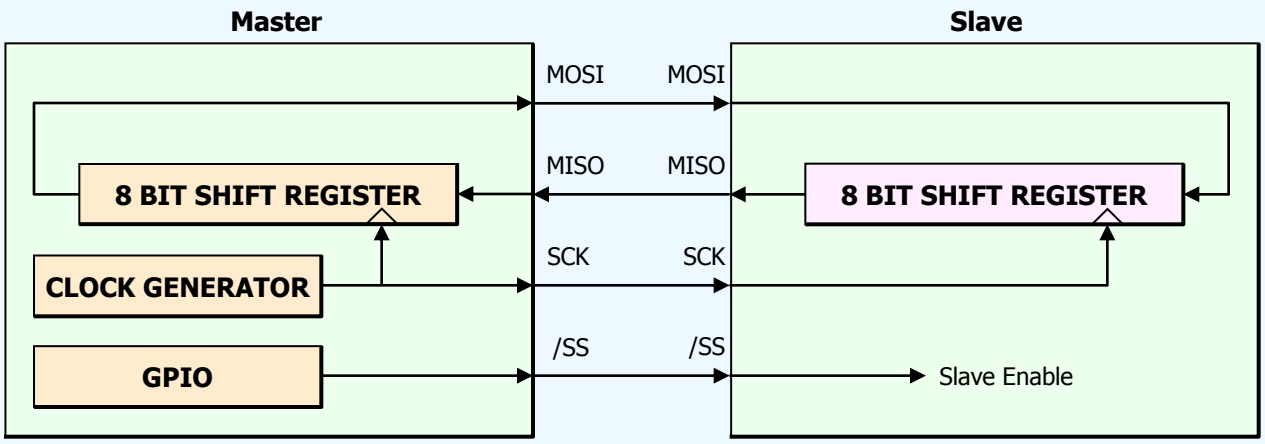
- ESPI : SPI Interrupt Enable

6.13. SPI : Block Diagram



6.13. SPI : Overview

- SPI Master-Slave Interconnection

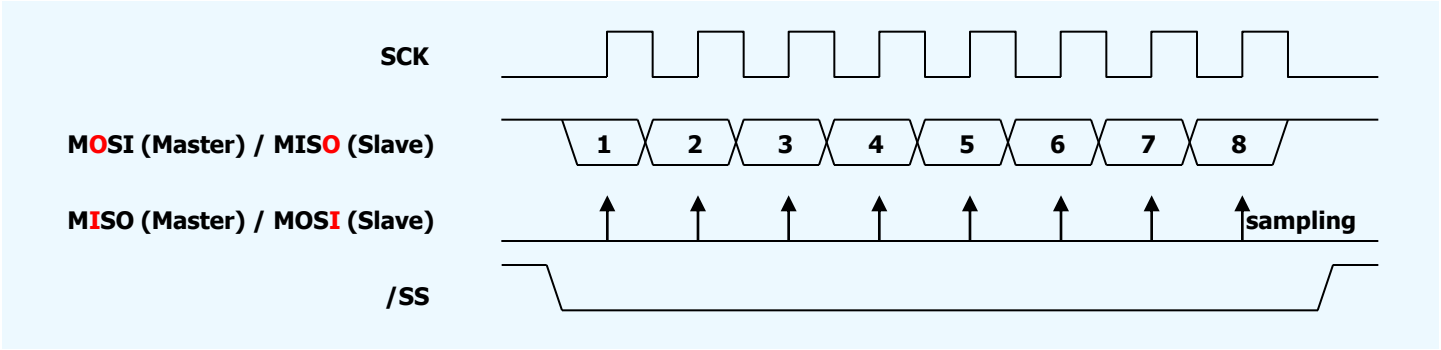


- SPI Pin Description

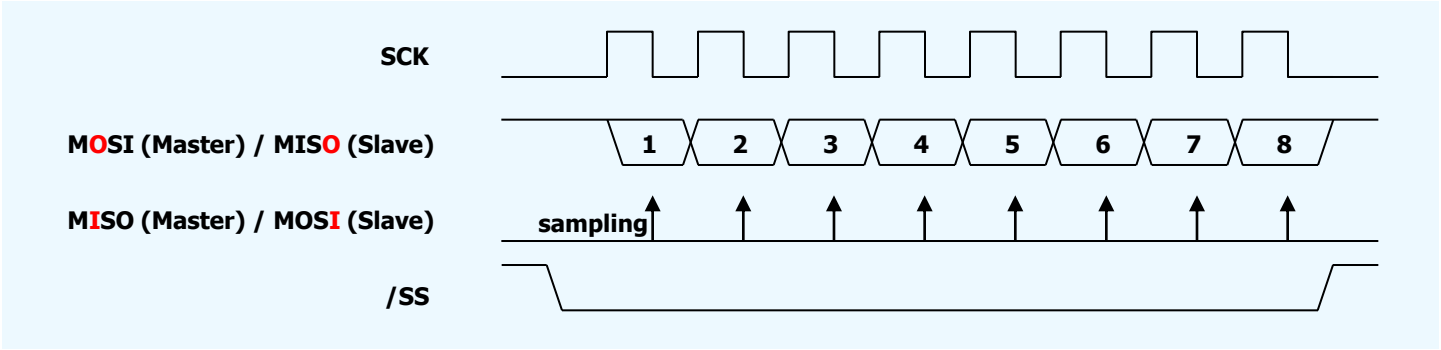
Pin	Description	Direction, Master	Direction, Slave
MOSI	Master Output Slave Input	User Defined	Input
MISO	Master Input Slave Output	Input	User Defined
SCK	SPI Clock	User Defined	Input
/SS	Slave Select Bar	User Defined	Input

6.13. SPI : Mode 0/1

- SPI Mode 0
  - ✓ CKPOL = 0 : Leading Edge → Rising
  - ✓ CKPHA = 0 : Leading Edge → Sampling

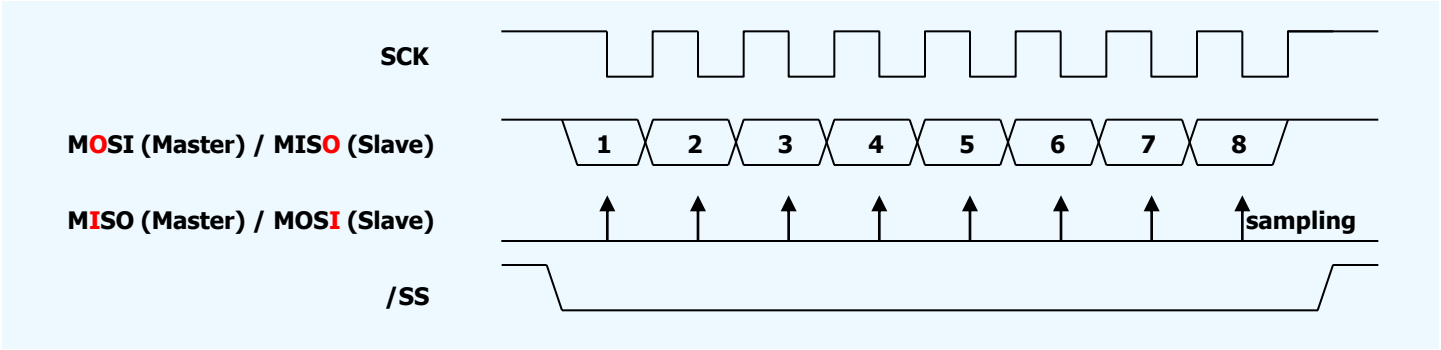


- SPI Mode 1
  - ✓ CKPOL = 0 : Leading Edge → Rising
  - ✓ CKPHA = 1 : Trailing Edge → Sampling

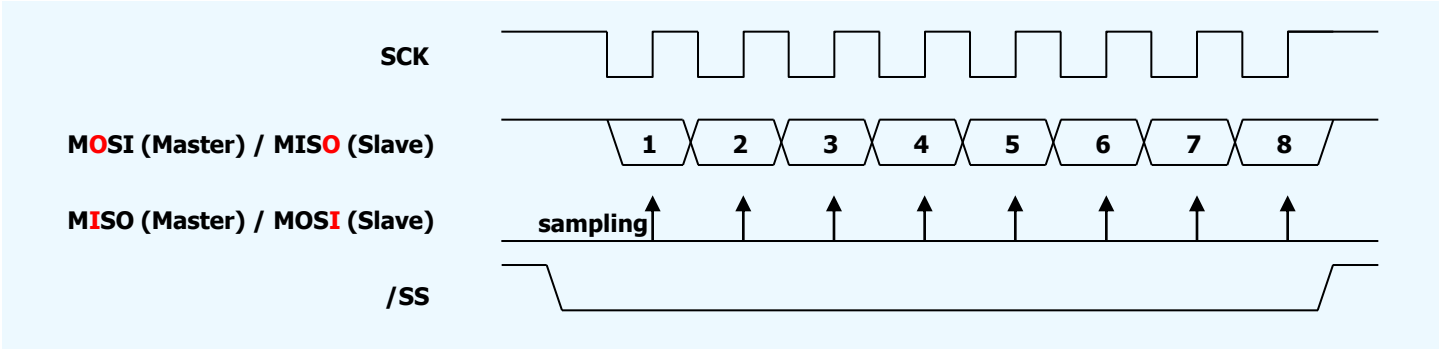


6.13. SPI : Mode 2/3

- SPI Mode 2
  - ✓ CKPOL = 1 : Leading Edge → Falling
  - ✓ CKPHA = 0 : Leading Edge → Sampling



- SPI Mode 3
  - ✓ CKPOL = 1 : Leading Edge → Falling
  - ✓ CKPHA = 1 : Trailing Edge → Sampling



6.13. SPI : Example

- Master example code

```
SPIST   EQU     0C0H
SPICON  EQU     0B4H
SPICK   EQU     0B5H
SPIDR   EQU     0B6H

ORG 000h
  LJMP  START

ORG 073h                        ; SPI interrupt routine
  ANL   SPIST,  #0FBh           ; clear interrupt flag
  MOV   R2,     #01h
  RETI

ORG 0100h
START:
  MOV   R2,     #00h            ; clear interrupt indicator
  ORL   SPICON, #10h            ; master mode
  ANL   SPICON, #0FBh           ; sampling clock leading edge
  ANL   SPICON, #0F7h           ; sampling clock first edge
  ANL   SPICON, #0DFh           ; first bit : MSB
  MOV   SPICK,  #00h            ; clock scaling Fperi / 2
  ORL   SPICON, #02h            ; SPI IO enable
  ORL   SPICON, #01h            ; SPI enable

  MOV   SPIDR,  #55h            ; write TX data
  CJNE  R2,     #01h,   .       ; wait TX interrupt
  MOV   R2,     #00h            ; clear interrupt indicator
```

- Slave example code

```
SPIST   EQU     0C0H
SPICON  EQU     0B4H
SPICK   EQU     0B5H
SPIDR   EQU     0B6H

ORG 000h
  LJMP  START

ORG 073h
  ANL   SPIST,  #0FBh           ; clear interrupt flag
  MOV   R2,     #01h
  RETI

ORG 0100h
START:
  MOV   R2,     #00h            ; clear interrupt indicator
  ANL   SPICON, #0EFh           ; slave mode
  ORL   SPICON, #04h            ; sampling clock trailing edge
  ORL   SPICON, #08h            ; sampling clock second edge
  ORL   SPICON, #20h            ; first bit : LSB
  MOV   SPICK,  #01h            ; clock scaling Fperi / 4
  ORL   SPICON, #02h            ; SPI IO enable
  ORL   SPICON, #01h            ; SPI enable

  CJNE  R2,     #01h,   .       ; wait RX interrupt
  MOV   R2,     #00h            ; clear interrupt indicator
  MOV   A,      SPIDR           ; read  RX data
```

6.14. ADC (Analog-to-Digital Converter)

- 8-channel 10-bit ADC (SAR Type)
- Max. 104ksps(samples per sec.) @ FADC = 10MHz & 3V. (Max. 52ksps @ FADC = 5MHz & 3V)

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

AD_EN	AD_REQ	AD_END	ADCF	-	-	SAR1	SAR0
R/W(0)	R/W(0)	R(1)	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.
  User must check the ADCF instead of AD_END.
- ADCF : ADC Interrupt Flag.
  Must be cleared by S/W.
- SAR[1:0] : Low Bits of ADC Result Value. (Total 10 bits)

✓ **ADCR** (E6h) : 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)

✓**ADCSEL** (DFh) : ADC Clock and MUX Selection Register

ADIV2	ADIV1	ADIV0	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- ADIV[2:0] : ADC clock selection.
  [000] : FSYS / 2.
  [001] : FSYS/ 4.
  [010] : FSYS / 8.
  [011] : FSYS / 16.
  [100] : FSYS / 32.
  [101] : FSYS / 64.
  [110] : FSYS / 128.
  [111] : FSYS / 256.
- ADCS[4:0] : ADC channel selection
  [00000] : ADC0.0 channel selection.
  [00001] : ADC0.1 channel selection.
  [00010] : ADC0.2 channel selection.
  [00011] : ADC0.3 channel selection.
  [00100] : ADC0.4 channel selection.
  [00101] : ADC0.5 channel selection.
  [00110] : ADC0.6 channel selection.
  [00111] : ADC0.7 channel selection.
  [01000] : ADC1.0 channel selection.
  .....
  [10111] : ADC2.7 channel selection.
  [11000] : ADC3.0 channel selection.
  [11001] : ADC3.1 channel selection.
  [11010] : ADC3.2 channel selection.
  [11011] : ADC3.3 channel selection.
  [11100] : ADC3.4 channel selection.
  [11101] : ADC3.5 channel selection.
  [11110] : ADC3.6 channel selection.
  [11111] : ADC3.7 channel selection.

6.14. ADC (Block Diagram)

✓ **ADCENB0** (CEh) : ADC Channel Enable Bar Register (P0 port)

-	-	-		ADCENB0.3	ADCENB0.2	ADCENB0.1	ADCENB0.0
				R/W(1)	R/W(1)	R/W(1)	R/W(1)

- 0 = ADC0 channel ON / 1 = ADC0 channel OFF (Default)

✓ **ADCENB1** (CFh) : ADC Channel Enable Bar Register (P1 port)

-	-	ADCENB1.5	ADCENB1.4	ADCENB1.3	ADCENB1.2	ADCENB1.1	ADCENB1.0
		R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

- 0 = ADC1 channel ON / 1 = ADC1 channel OFF (Default)

✓ **ADCENB2** (D6h) : ADC Channel Enable Bar Register (P2 port)

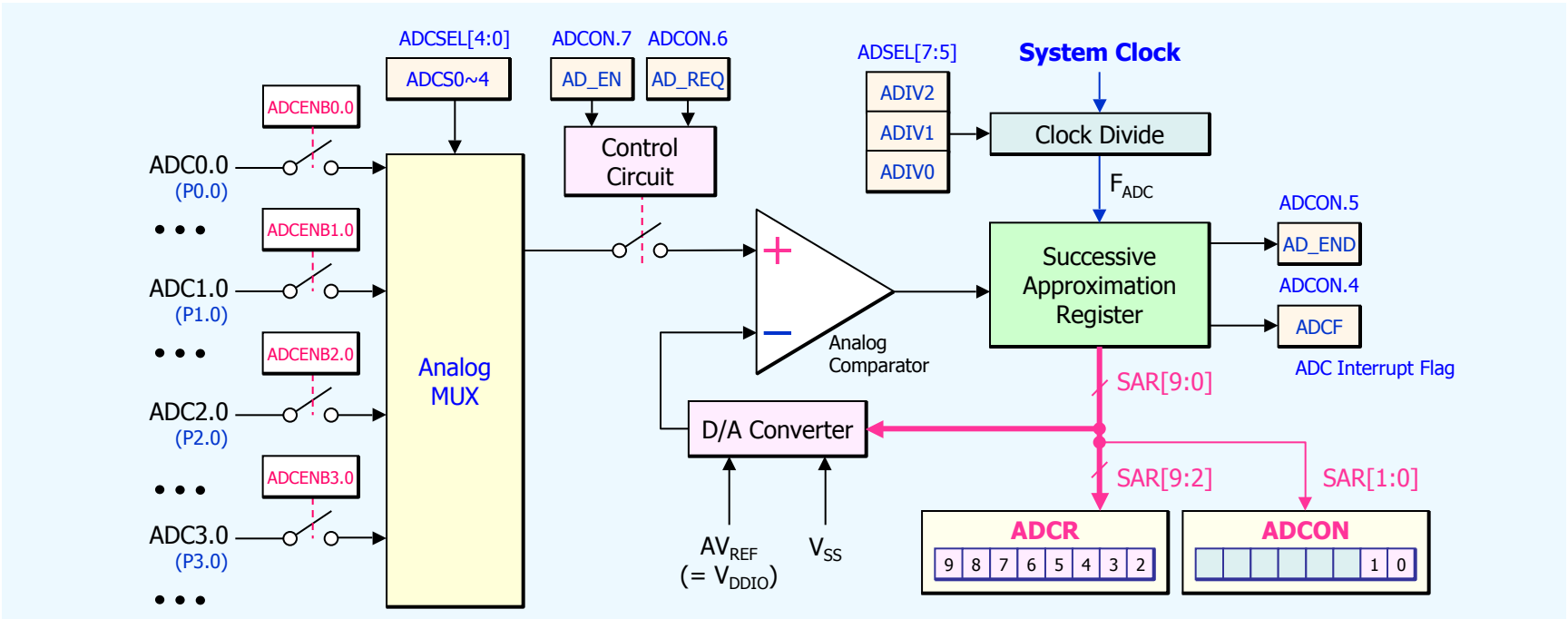
ADCENB2.7	ADCENB2.6	ADCENB2.5	ADCENB2.4	ADCENB2.3	ADCENB2.2	ADCENB2.1	ADCENB2.0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

- 0 = ADC2 channel ON / 1 = ADC2 channel OFF (Default)

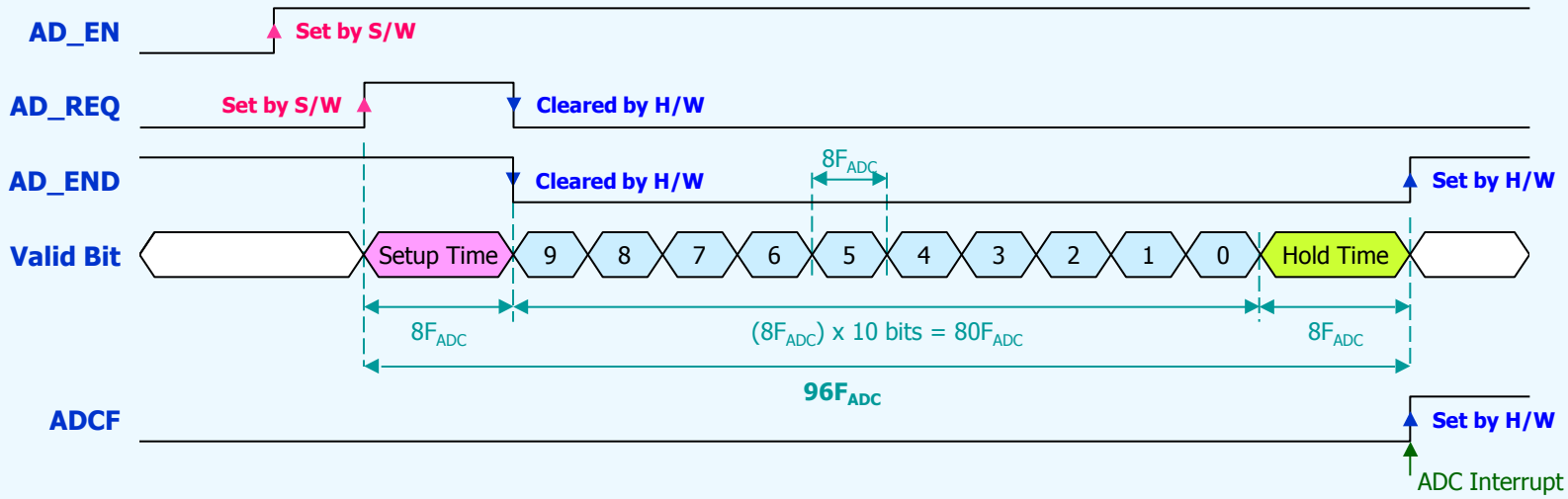
✓ **ADCENB3** (D7h) : ADC Channel Enable Bar Register (P3 port)

-	-	ADCENB3.5	ADCENB3.4	ADCENB3.3	ADCENB3.2	ADCENB3.1	ADCENB3.0
		R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

- 0 = ADC3 channel ON / 1 = ADC3 channel OFF (Default)



6.14. ADC : Conversion Timing



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

[An Example of ADC Conversion Table]

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

6.15. Interrupt : 12 Sources / 4-level Priority

- Interrupt Sources : Timer 0/1/2, ADC, WDT, I2C, SPI, UART 4 External.
- 4-level Interrupt Priority
  - ✓ Timer 0/1/2, UART, ADC, INT0, INT1
- 2-level Interrupt Priority
  - ✓ WDT, I2C, SPI, INT2, INT3

[Interrupt Vector Address]

HIGH

PRIORITY

LOW

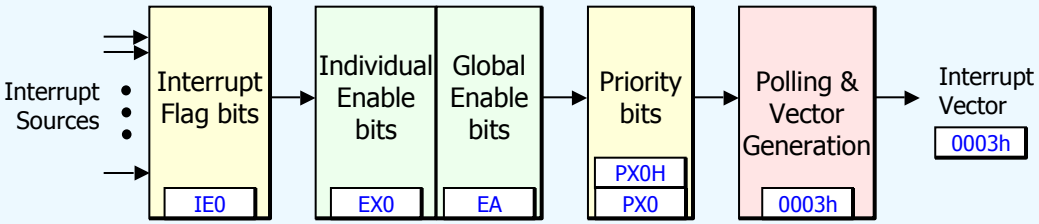
Interrupt Sources	Address	Priority Level
INT0	0003h	4 Levels
TF0	000Bh	4 Levels
INT1	0013h	4 Levels
TF1	001Bh	4 Levels
RI+TI	0023h	4 Levels
TF2	002Bh	4 Levels
ADC	003Bh	4 Levels
INT2	0043h	2 Levels
INT3	004Bh	2 Levels
-	0053h	2 Levels
-	005Bh	2 Levels
WDT	0063h	2 Levels
I2C	006Bh	2 Levels
Reserved	0073h	2 Levels
SPI	007Bh	2 Levels
Reserved	0083h	2 Levels

8052

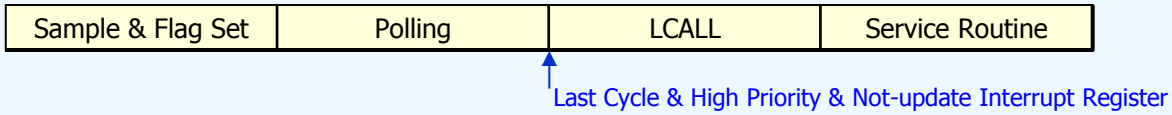
*** Interrupt SFR's (refer to Appendix B : SFR Description)**

✓ **TCON**	(88h)	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
✓ **IT**	(B2h)	-	-	-	-	IT5	IT4	IT3	IT2
✓ **ITSEL**	(BAh)	-	-	ITSEL5	ITSEL4	ITSEL3	ITSEL2	ITSEL1	ITSEL0
✓ **EXIF**	(91h)	-	-	IE3	IE2	XT/RL	RGM0	RGSL	BGS
✓ **IE**	(A8h)	EA	EADC	ET2	ES	ET1	EX1	ET0	EX0
✓ **EIE**	(A1h)	ESPI	-	EI2C	EWDT	-	-	EX3	EX2
✓ **IP**	(B8h)	-	PADC	PT2	PS	PT1	PX1	PT0	PX0
✓ **IPH**	(B7h)	-	PADCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
✓ **EIP**	(B1h)	PSPI	-	PI2C	PWDT	-	-	PX3	PX2
✓ **WDCON**	(D8h)	WDMOD	POR	EPFI	PFI	WDIF	WTRF	EWT	RWT

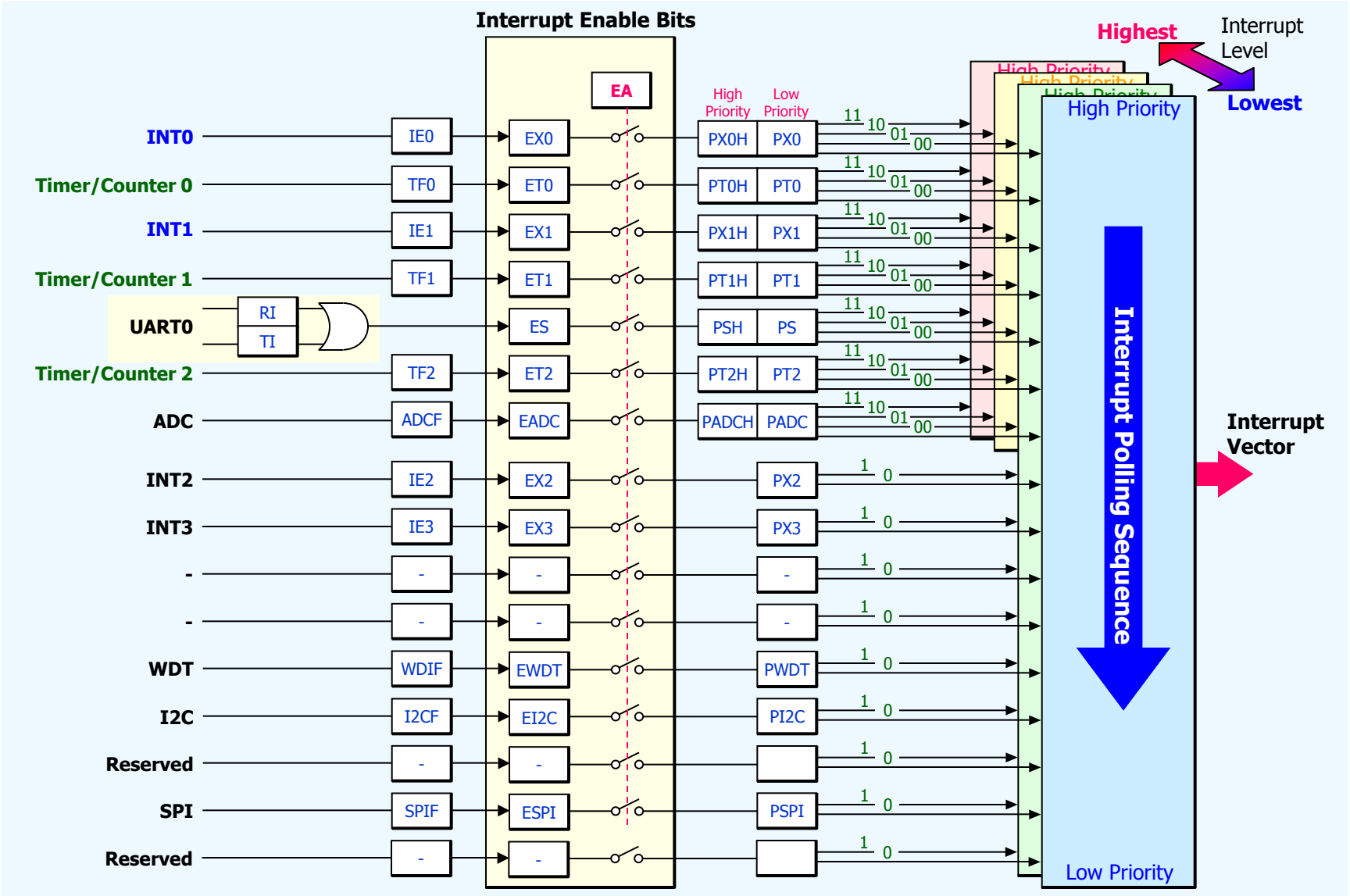
[Interrupt Vector Generation Flow]



[Response Sequence]



6.15. Interrupt Functional Description



6.15. Interrupt : External Interrupt

- External Interrupt Sources : INT5~0
- Support positive edge and negative edge detection
- Support high level and low level detection

✓ **IT** (B2h) : Interrupt Type Selection Register

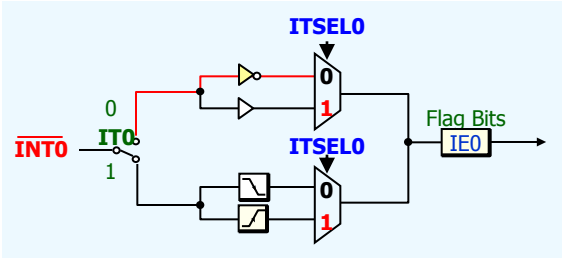
-	-	-	-	IT5	IT4	IT3	IT2
				R/W(1)	R/W(1)	R/W(1)	R/W(1)

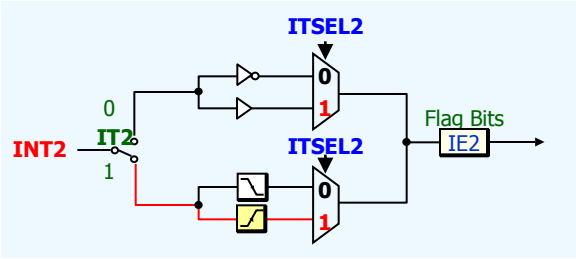
- IT5 : Interrupt5 Type Selection Flag
  [0] : Level detect, [1] : Edge detect
- IT4 : Interrupt4 Type Selection Flag
  [0] : Level detect, [1] : Edge detect
- IT3 : Interrupt3 Type Selection Flag
  [0] : Level detect, [1] : Edge detect
- IT2 : Interrupt2 Type Selection Flag
  [0] : Level detect, [1] : Edge detect

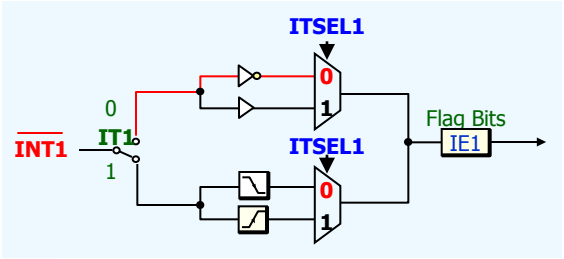
✓ **ITSEL** (BAh) : Interrupt Polarity Selection Register

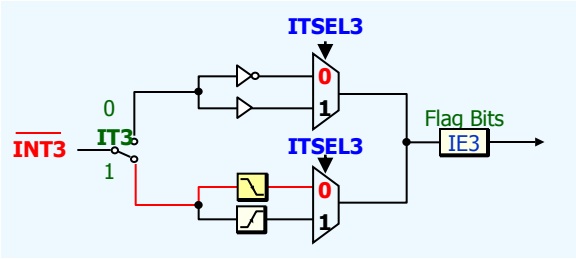
-	-	ITSEL5	ITSEL4	ITSEL3	ITSEL2	ITSEL1	ITSEL0
		R/W(0)	R/W(1)	R/W(0)	R/W(1)	R/W(0)	R/W(0)

- ITSEL5 : Interrupt5 Polarity Selection Flag
  [0] : low level or negative edge, [1] : high / positive
- ITSEL4 : Interrupt4 Polarity Selection Flag
  [0] : low level or negative edge, [1] : high / positive
- ITSEL3 : Interrupt3 Polarity Selection Flag
  [0] : low level or negative edge, [1] : high / positive
- ITSEL2 : Interrupt2 Polarity Selection Flag
  [0] : low level or negative edge, [1] : high / positive
- ITSEL1 : Interrupt1 Polarity Selection Flag
  [0] : low level or negative edge, [1] : high / positive
- ITSEL0 : Interrupt0 Polarity Selection Flag
  [0] : low level or negative edge, [1] : high / positive









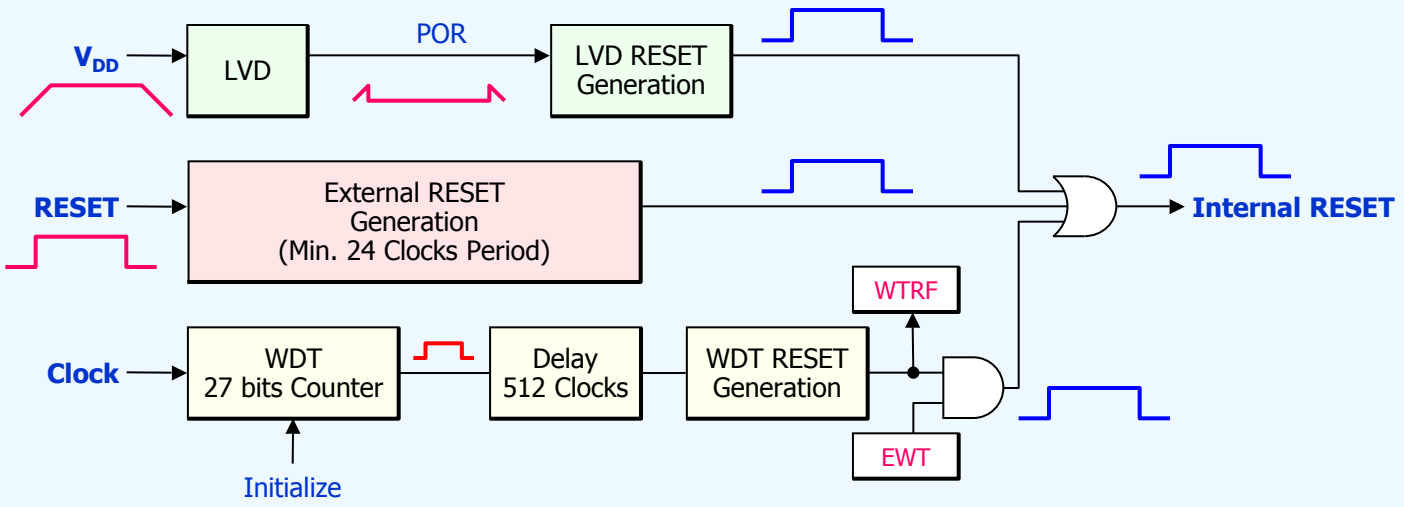
6.16. Reset Circuit : 3 Reset Sources

- LVD(POR) Reset
  - ✓ Power-on Reset when power is turned on.
  - ✓ Power-fail Reset when the supply voltage is below the threshold voltage (V_{RST}).
- External RESET Pin
  - ✓ RESET Pin must be held "High" for at least 24 clock cycles.
- WDT Reset : Enable or Disable by S/W

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

-	POR	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)

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



6.17. Clock Circuit : SFR

- System Clock Sources
  - ✓ Crystal OSC
  - ✓ Oscillator
  - ✓ Internal RING OSC
- Disable of External Clock (Crystal or External Oscillator)
  - ✓ If XTOFF is set.
  - ✓ When MCU is in stop mode and WDT is not active.
- Disable of the Internal RING Oscillator
  - ✓ If RINGON is cleared.
  - ✓ When MCU is in stop mode and WDT is not active.
- Wake-up from stop by WDT
  - ✓ WDT is active in stop mode if EWT is set or WDT interrupt is enabled.
  - ✓ In this case, the clock of WDT is alive during stop mode.

✓ **PCON** (87h) : Extended Interrupt Enable Register

SMOD1	SMOD0	-	POF	GF1	GF0	PD	IDL
R/W(0)	R/W(0)		R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- PD : Power-down (Stop) mode enable.
- IDL : IDL mode enable

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

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

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

-	-	-	-	XTOFF	ALEOFF	-	-
				R/W(1)	R/W(0)		

- XTOFF : 1 = External crystal Oscillator disable (Default).
  0 = External crystal will restart.

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

-	-	-	XTUP	-	-		-
			R(0)				

- XTUP : Crystal Oscillator warm-up status.
  It represents if the crystal clock is stable(1) or not(0).
  Cleared by H/W if XTOFF is set or if PD is set and WDT is not enabled.
  Set by H/W after crystal stabilization time.

✓ **OSCICN** (C6h) : Internal RING Oscillator Control Register

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

- RINGON : 1 = Internal ring Oscillator is running.
  0 = Internal ring Oscillator is killed.
  Don't clear RINGON bit when XTRG = 0.

✓ **OSCICN2** (C7h) : Test Internal RING Oscillator Control Register

-	-	-	-	-	RINGON2	-	-
					R/W(1)		

- RINGON2 : 1 = Test Internal ring Oscillator is running.
  0 = Test Internal ring Oscillator is killed.
  Must clear the flag for saving power at STOP mode

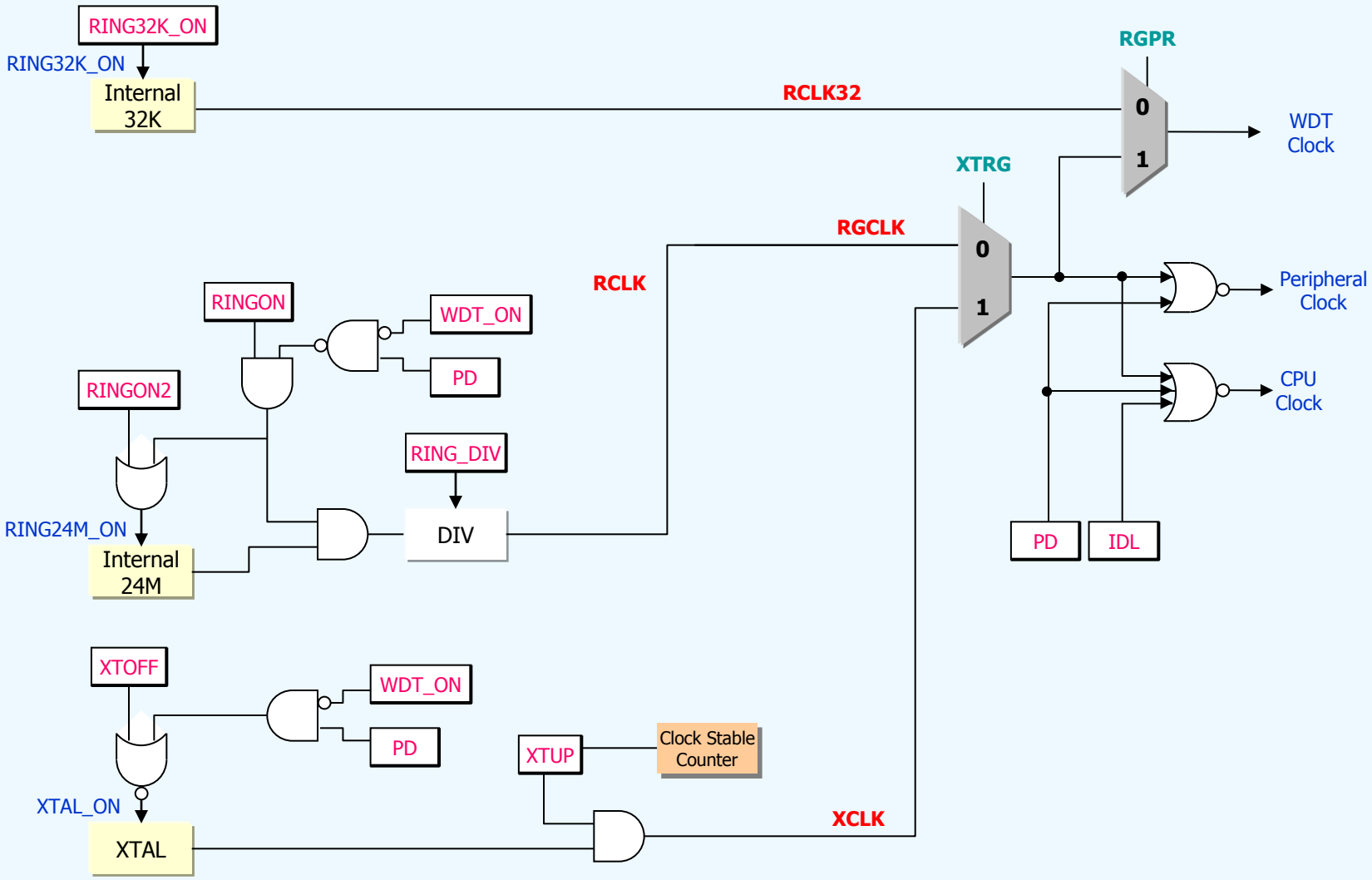
6.18. Clock Circuit : SFR (Cont'd)

✓ **CKSEL** (86h) : Clock Selection Register

-	-	-	R32KOE	R24MOE	-	RGPR	R32KEN
			R/W(0)	R/W(0)		R/W(1)	R/W(1)

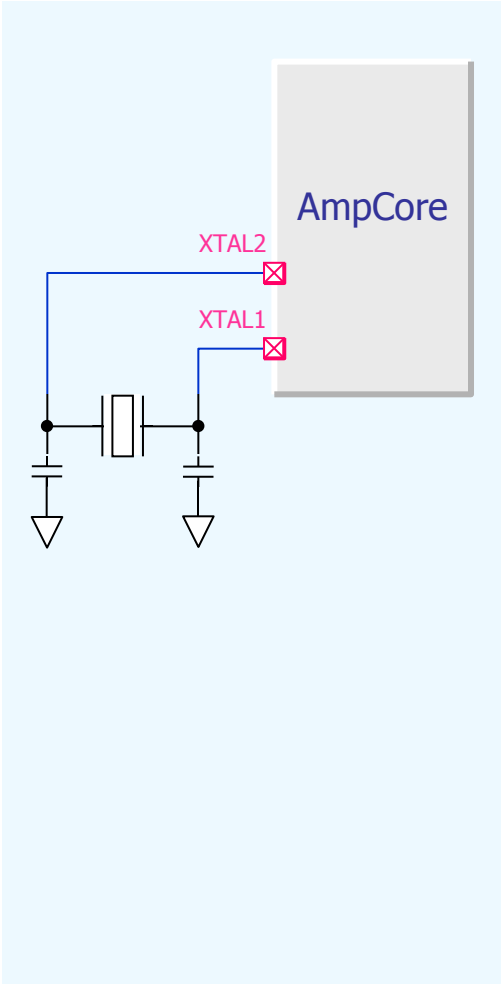
- R32KOE : RING 32K port output enable (P0.3)
- R32MOE : RING 24M port output enable (P0.2)
- RGPR : WDT clock selection
  0 = RING 32K is selected as WDT clock.
  1 = divided clock of RING 24M is selected as WDT clock.
- R32KEN : RING 32K clock enable flag.

6.17. Clock Circuit : Circuit Diagram

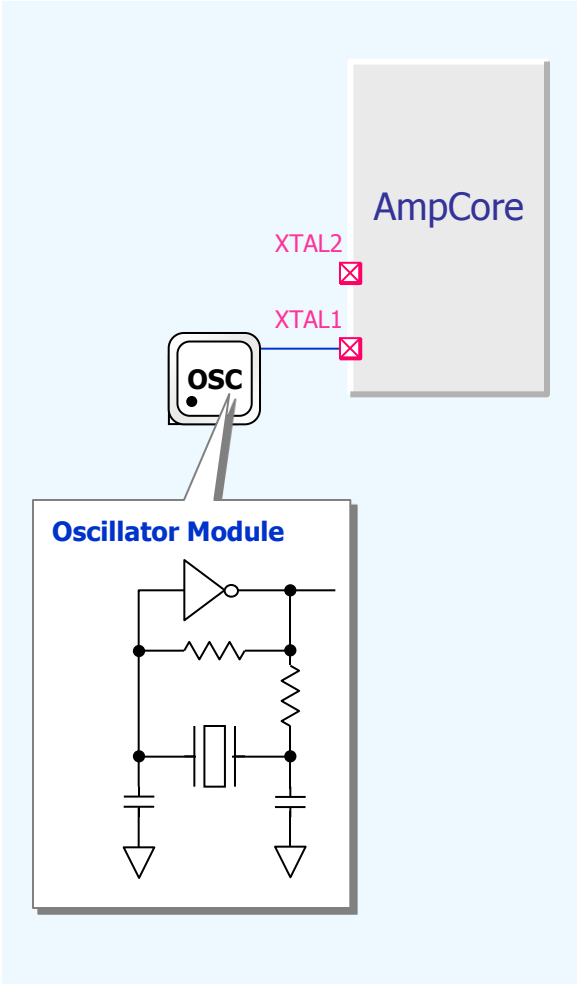


6.17. Clock Circuit : Guideline for Configuration

◆ **Crystal Oscillator**

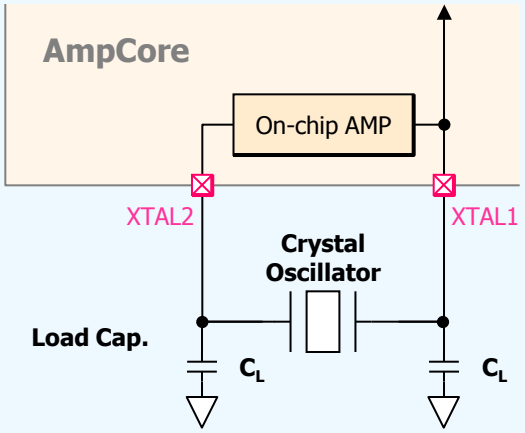


◆ **Oscillator Module**

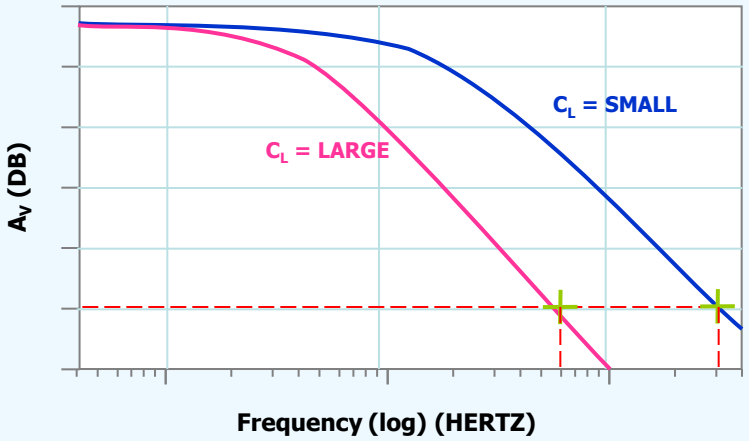


6.17. Clock Circuit: Guideline for Using a Crystal

◆ Crystal Oscillator & Load Capacitors



◆ Graph for Load Capacitor & Frequency



◆ Recommended C_L (Load Capacitor)

V_{DD} = 5 V

	Crystal Oscillator [MHz]		
	~ 11.0592	22.1184	30.0000
Load Cap. C_L	47pF	20pF	10pF

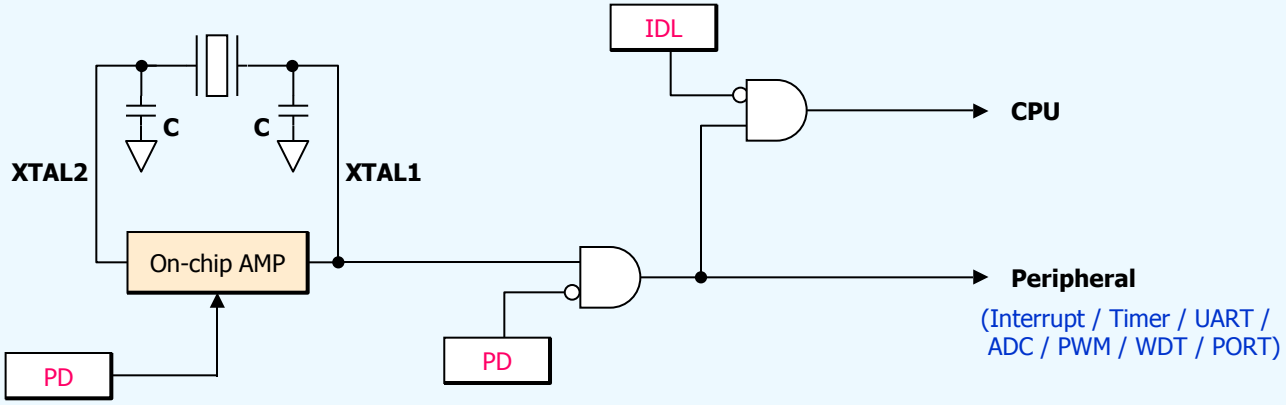
6.18. Power Management : 3 Modes

- **Active Mode** : The CPU and The Peripherals operate.
- **Idle Mode** : The CPU is gated off from the clock signal.
  Only the Peripherals operate.
  - ✓ Exited by activating any interrupt. The CPU resumes.
  - ✓ Exited by activating any reset. The CPU restarts.
- **Stop Mode** : All clocks are stopped.
  All activity is completely stopped.
  - ✓ Exited by activating external interrupt 0 or 1 (level detect) The CPU resumes.
    External pins must hold '0' during at least crystal stabilization time.
  - ✓ Exited by activating any reset. The CPU restarts.

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

SMOD1	SMOD0	-	POF	GF1	GF0	PD	IDL
R/W(0)	R/W(0)		R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

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



6.20. ISP & Debugging

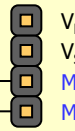
- Code memory (8KBytes) can be programmed using EJTAG in target system.
  - ✓ FLASH : 0x0000 ~ 0x1FFF (8,192 Bytes)
- EEPROM (1k Byte) can be programmed using EJTAG in target system.
  - ✓ EEPROM : 0x1C00 ~ 0x1FFF (1,024 Bytes)
- Debugging using GENICE
- MDS_SDA (RESET) pin connection for ISP
  - ✓ Case 1 : Not connection
  - ✓ Case 2 : Pull-down using resistor > 500KΩ
- MDS_SCL pin connection for ISP
  - ✓ Case 1 : Not connection
  - ✓ Case 2 : Pull-up using resistor > 10KΩ

[ISP Pin Configuration 0]

V_{DD} (+3.3V or +5V)
V_{SS} (GND)
MDS_SCL
MDS_SDA (RESET)

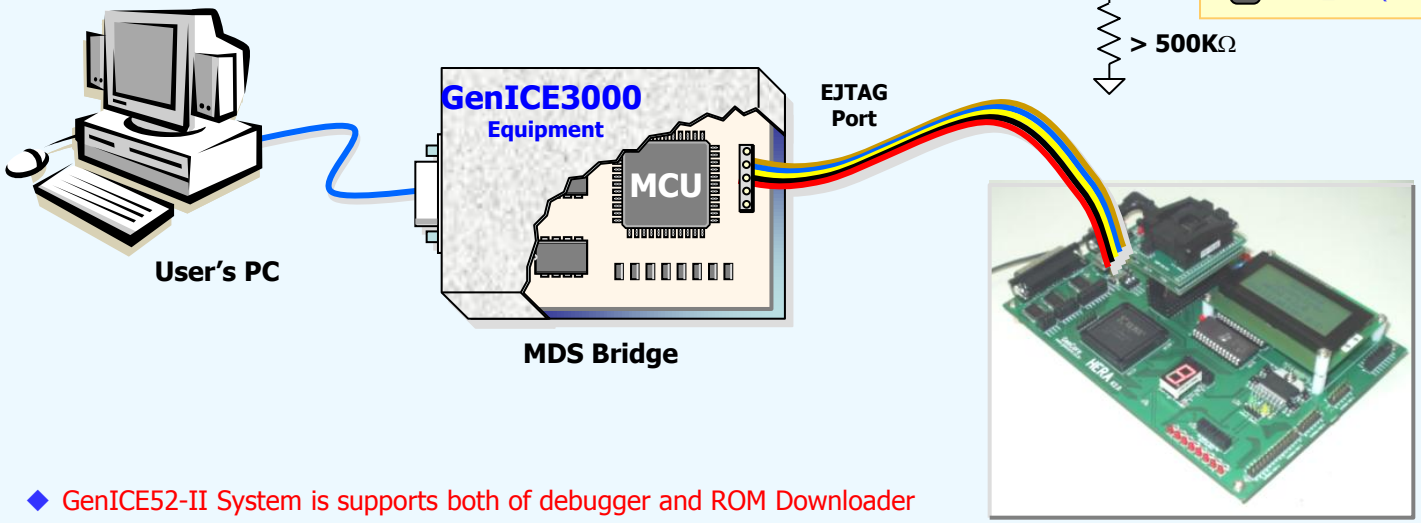
[ISP Pin Configuration 1]

> 10KΩ



V_{DD} (+3.3V or +5V)
V_{SS} (GND)
MDS_SCL
MDS_SDA (RESET)

> 500KΩ



- GenICE52-II System is supports both of debugger and ROM Downloader
- Refer to http://www.coreriver.com → Support → Supporting Tools → 1.Brief System Manual

6.20. 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.

6.21. 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** (F7h) : 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** (F1h) : 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)

✓ **EECNTM** (F3h) : EEPROM Erase/Program Time Count MSB

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** (F2h) : EEPROM Erase/Program Time Count LSB

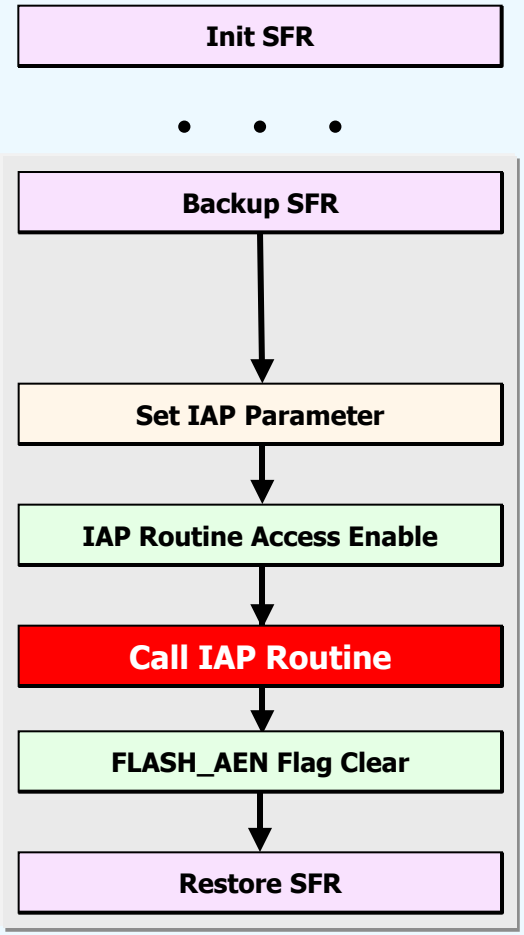
EECNT.7	EECNT.6	EECNT.5	EECNT.4	EECNT.3	EECNT.2	EECNT.1	EECNT.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

6.21. IAP : Function Set

- IAP call function
  - ✓ iap_eeprom_program : call address (FF0Ah)
  - ✓ iap_eeprom_erase : call address (FF00h)

- 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).

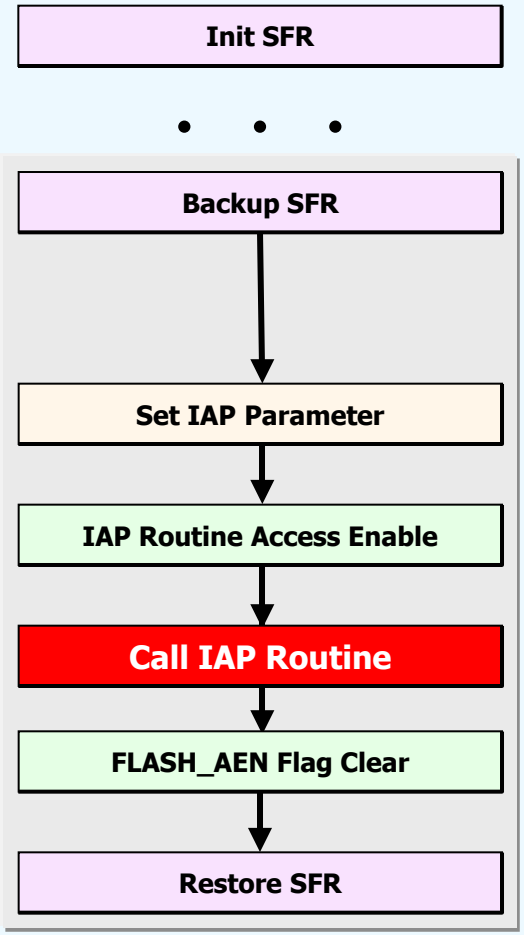
6.21. IAP : Program Flow



[Example Code : IAP Program for EEPROM]

```
ORL EEAEN, #01h          ; IAP routine access enable
MOV EECNTM, #75h         ; Program/Erase Time Count Mid
MOV EECNTL, #30h         ; Program/Erase Time Count Low
ANL EEAEN, #0FEh         ; IAP routine access disable
                         ; 0x007530 @ RING Freq. == 12MHz

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
```

6.21. IAP : Erase Flow

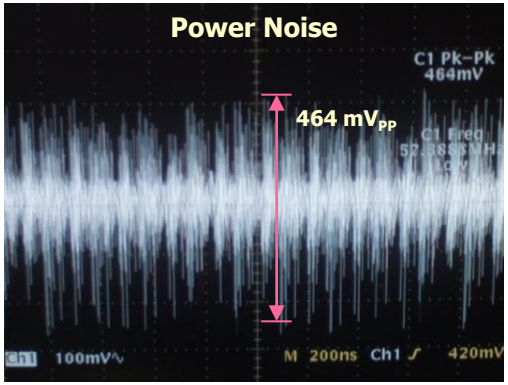
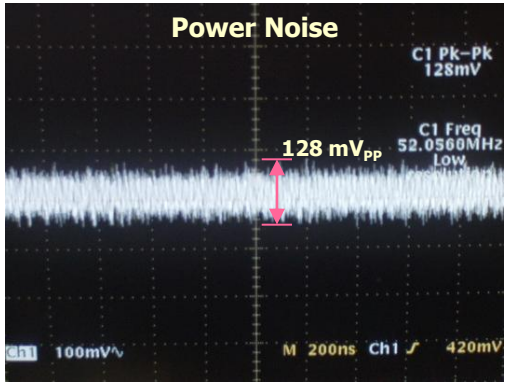
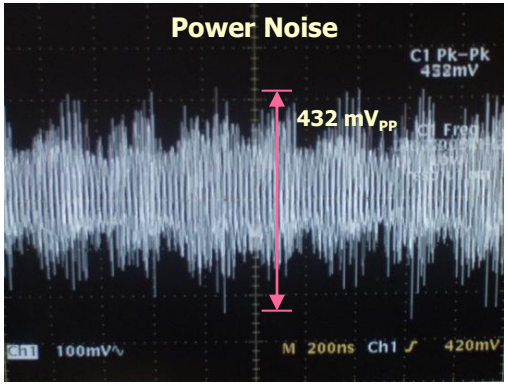
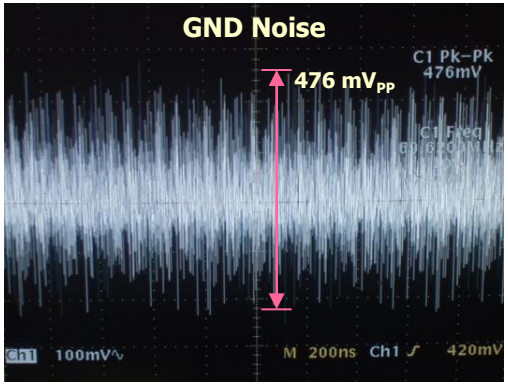
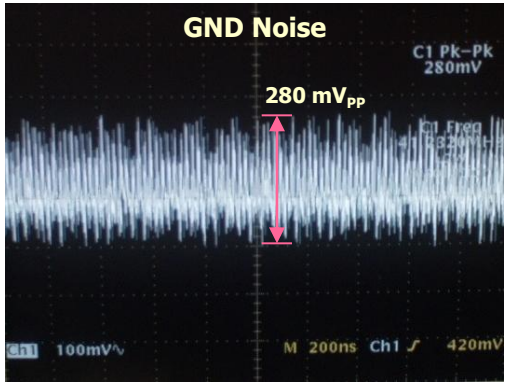
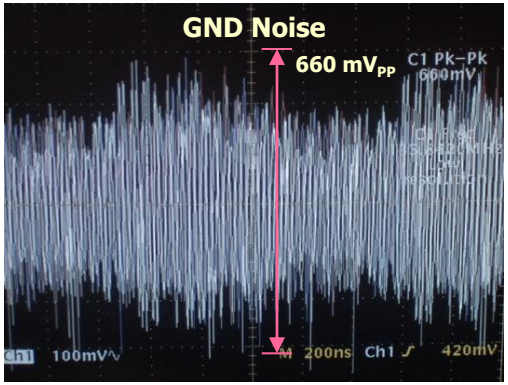


[Example Code : IAP Erase for EEPROM]

```
ORL EEAEN, #01h          ; IAP routine access enable
MOV EECNTM, #75h         ; Program/Erase Time Count Mid
MOV EECNTL, #30h         ; Program/Erase Time Count Low
ANL EEAEN, #0FEh         ; IAP routine access disable
                         ; 0x007530 @ RING Freq. == 12MHz

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, dptr
POP DPL
POP ACC
```

7. Strong Point I : Noise Reduction

Clock : 22.1184 MHz

AmpCore1.0 : ALE "ON"	AmpCore1.0 : ALE "OFF"	Company A's 80C52
Power Noise	Power Noise	Power Noise
		
GND Noise	GND Noise	GND Noise
		

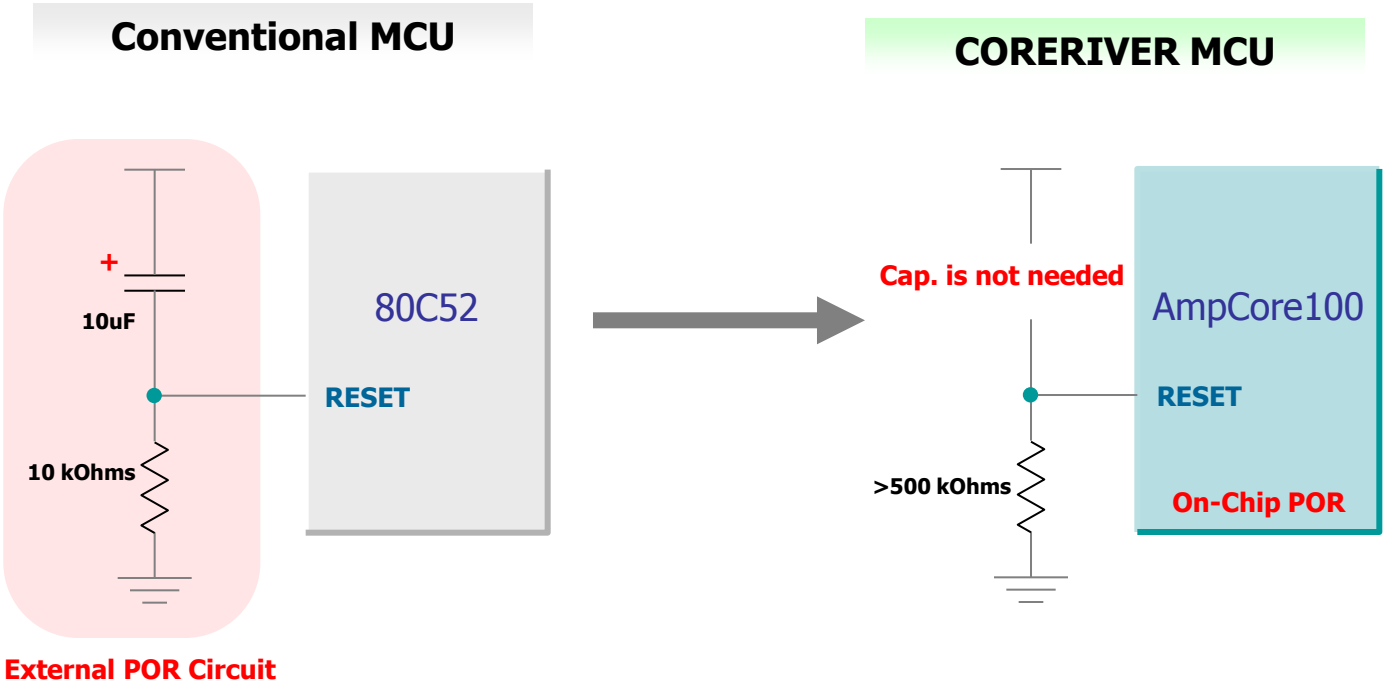
7. Strong Point I : Noise Reduction (Cont'd)

System Clock [MHz]	Noise	AmpCore100		Company A's 80C52
		ALE "ON"	ALE "OFF"	ALE always "ON"
11.0592	Power	410 mV_{PP}	**170 mV_{PP}**	360 mV_{PP}
	Ground	550 mV_{PP}	**330 mV_{PP}**	500 mV_{PP}
22.1184	Power	464 mV_{PP}	**128 mV_{PP}**	432 mV_{PP}
	Ground	476 mV_{PP}	**280 mV_{PP}**	640 mV_{PP}
6	Power	360 mV_{PP}	**170 mV_{PP}**	380 mV_{PP}
	Ground	500 mV_{PP}	**330 mV_{PP}**	480 mV_{PP}

- AmpCore100 can reduce EMI by removing the needless swing of ALE signal.
- You can enable/disable ALE signal by changing the value of ALEOFF bit (SFR PMR.2).
- Or can reduce system cost by removing needless decoupling capacitors while maintaining the EMI.

* This experimental results can vary according to design style of an application system.

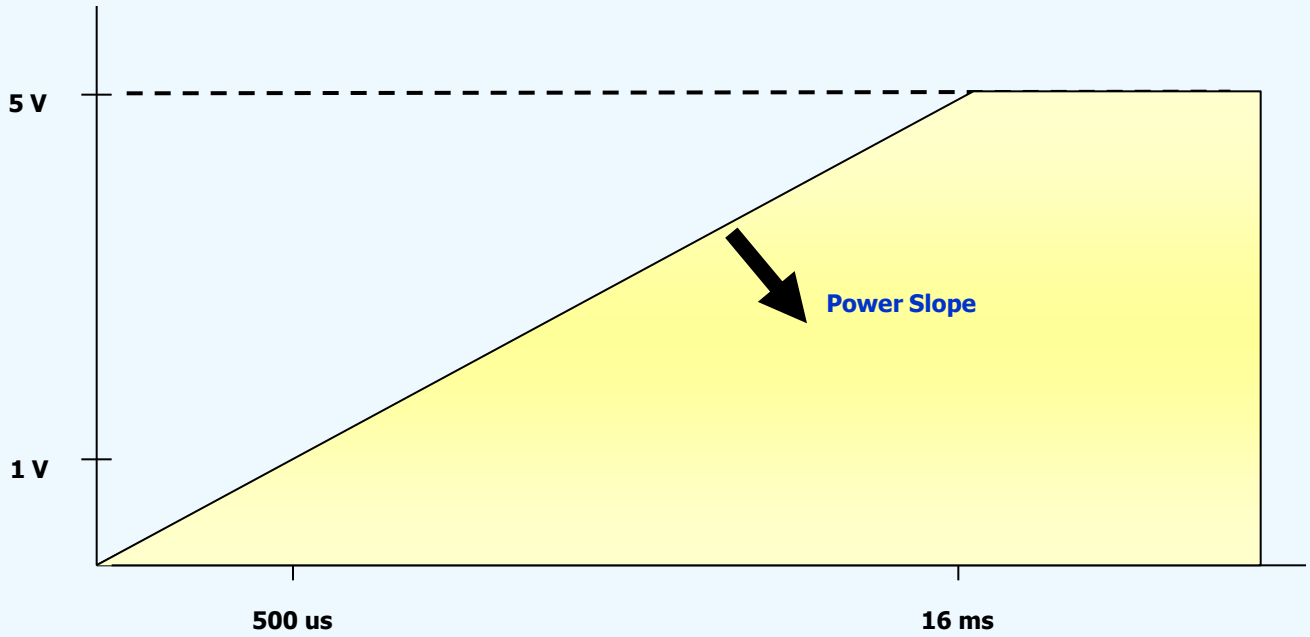
7. Strong Point II : On-Chip POR



- On-Chip POR (Power On Reset) can reduce system cost by removing a needless capacitor.
- The states of all ports will not be determined until the POR procedure ends.

8. Recommended Power Slope

- The supply voltage slope must be in the range from 0.0V/us to 1.0V/3.2ms. (5V/16ms)
  (That is, the supply voltage should be increasing monotonically until it reaches to the normal range.)



9.1. Absolute Maximum Ratings (MCU)

Items	Conditions	Ranges
Voltage on any pin relative to Ground	-	-0.5V to (V_{DD}+0.5V)
Voltage in V_{DD} relative to Ground	-	-0.5V to 6.5V
Output Voltage	-	-0.5V to (V_{DD}+0.5V)
Output Current High	One I/O pin active	-25mA
	All I/O pin active	-100mA
Output Current Low	One I/O pin active	+30mA
	All I/O pin active	+150mA
Storage Temperature (T_{STG})	-	-55 °C to +125 °C
Soldering Temperature (T_L)	-	Peak 260 °C, 10 seconds

9.2. Absolute Maximum Ratings (AMP)

Items	Conditions	Ranges
Power Supply Voltages (Single Supply) in V_{CC}	-	32V
Power Supply Voltages (Dual Supplies) in V_{CC}	-	±16
Input Differential Voltage Range (@Dual Power Supplies; V_{IDR})	-	±32
Input Common Mode Voltage Range (V_{ICR})	-	-0.3V to +32V
Output Short Circuit Duration (I_{SC})	-	-Continuous
Storage Temperature	-	-55 °C to +125 °C
Soldering Temperature	-	Peak 260 °C, 10 seconds
Junction Temperature Plastic Packages (T_J)	-	150 °C
Input Current per Pin (I_{IN})	-	50mA

10.1. DC Characteristics

*** TA = = -40 °C ~ +85 °C, V_{DD} = 2.2V ~ 5.5V unless otherwise specified.**

Parameter	Symbol	Pin	Conditions	Value			Unit
				Min.	Typ.	Max.	
Input Low Voltage	V_{IL1}	RESET ,P0, P1, P2, P3, P4	V_{DD} = 2.2V~5.5V	-0.5	-	$0.2V_{DD}$-0.1	V
	V_{IL2}	XTAL1, XTAL2		-0.5	-	$0.3V_{DD}$	
Input high Voltage	V_{IH1}	RESET, P0, P1 ,P2, P3, P4	V_{DD} = 2.2V~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}	XTAL1, XTAL2, RESET, EA, P0[3:2], P1, P2, P3[5:4,1:0], P4[7:4]	I_{OL} = 20mA @V_{DD}=5V (I_{OL} = 3mA @V_{DD}=2.2V)	-	-	$0.3V_{DD}$	V
	V_{OL1}	P0[1:0], P4[3:0]	I_{OL1} = 4mA @V_{DD}=5V	-	-	$0.3V_{DD}$	V
	V_{OLP}	RESET	I_{OLP} = 3.5uA @V_{DD}=5V	-	-	$0.3V_{DD}$	V
Output High Voltage	V_{OH}	XTAL1, XTAL2, RESET, EA, P0[3:2], P1, P2, P3[5:4,1:0], P4[7:4]	I_{OH} = -15mA @V_{DD}=5V (I_{OH}= -2mA @V_{DD}=2.2V)	$0.7V_{DD}$	-	-	V
	V_{OH1}	P0[1:0], P4[3:0]	I_{OH1} = -4mA @V_{DD}=5V	$0.7V_{DD}$	-	-	V
	V_{OHP}	ALL pin (Pull-up)	I_{OHP}= -40uA @V_{DD}=5V (I_{OHP} = -15uA @V_{DD}=2.2V)	$0.7V_{DD}$	-	-	V
Input Leakage Current	I_{IL}	All pins except XTAL1,XTAL2	V_{IN} = V_{IH} or V_{IL}	-	-	±1	μA
Pin Capacitance	C_{IO}	All	V_{DD} = 5V	-	10	-	pF

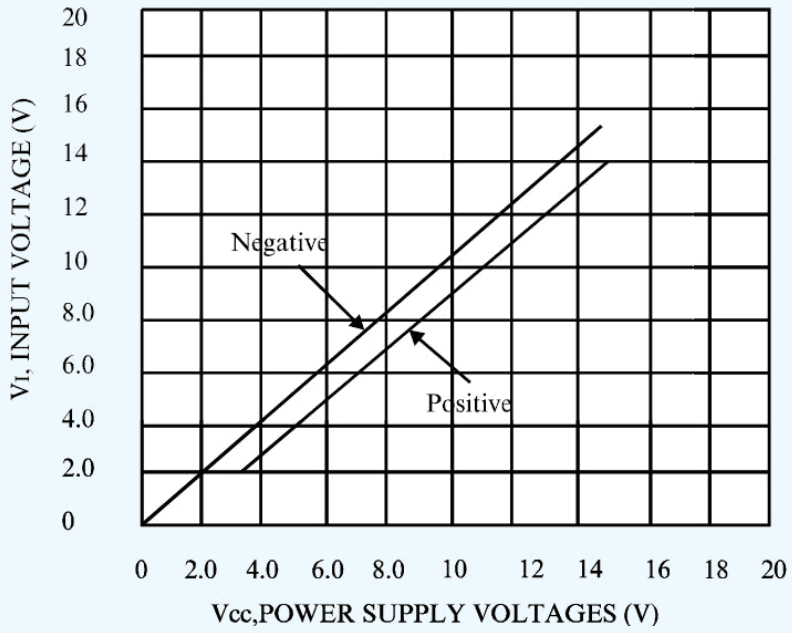
10.2. DC Characteristics (AMP)

T_A = = -40 °C ~ +85 °C (*=@25 °C)

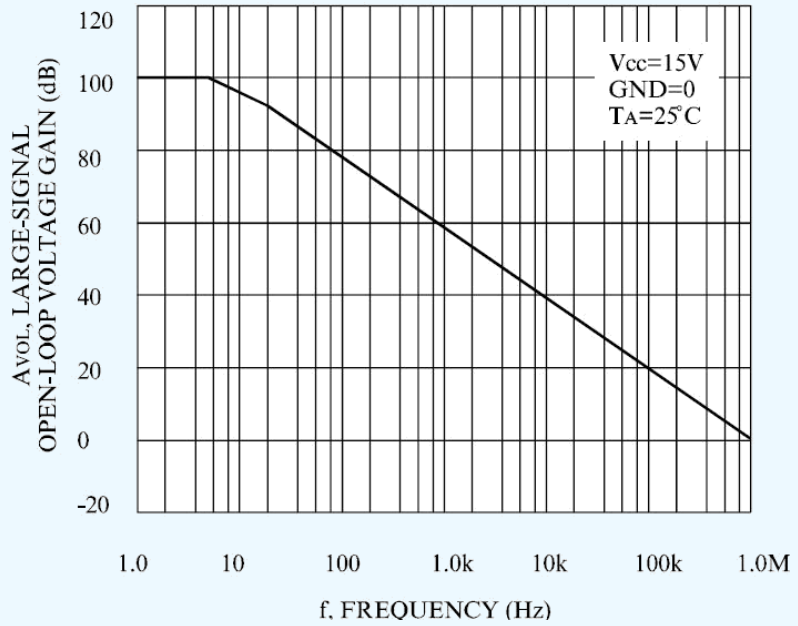
Parameter	Symbol	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
Maximum Input Offset Voltage	V_{IO}	V_O=1.4V, V_{CC}= 5 ~ 30V; R_S=0Ω, V_{ICM}=0V ~ V_{CC}-1.7V			9.0 5.0*	mV
Input Offset Voltage Drift	$\Delta V_{IO}/\Delta T$	R_S=0Ω, V_{CC}=30V		7.0		μV/°C
Maximum Input Offset Current	I_{IO}	V_{CC}=5.0V			150 50*	nA
Input Offset Current Drift	$\Delta I_{IO}/\Delta T$	R_S=0Ω, V_{CC}=30V		10		pA/°C
Maximum Input Bias Current	I_{IB}	V_{CC}=5.0V			500 250*	nA
Input Common Mode Voltage Range	V_{ICR}	V_{CC}=30V	0		28	V
Maximum Power Supply Current	I_{CC}	R_L=∞, V_{CC}=30V, V_0=0V R_L=∞, V_{CC}=5V, V_0=0V			3.0 1.2	mA
Minimum Large Signal Open-Loop Voltage Gain	A_{VOL}	V_{CC}=15V, R_L≥2KΩ V_{CC}=5.0V, R_L≥2KΩ	15(25*) 0.28			V/mV
Minimum Output High-Level Voltage Swing	V_{OH}	V_{CC}=30V, R_L=2KΩ V_{CC}=30V, R_L=10KΩ V_{CC}=5.0V, R_L=10KΩ	26 27 3.0			V
Maximum Output Low-Level Voltage Swing	V_{OL}	V_{CC}=5V, R_L=10KΩ			20	mV
Common Mode Rejection	CMR	V_{CC}=30V, R_S=10KΩ	65*			dB
Power Supply Rejection	PSR	V_{CC}=30V	65*			dB
Channel Separation	CS	f=1KHz to 20KHz, V_{CC}=30V	-120*			dB
Maximum Output Short Circuit to GND	I_{SC}	V_{CC}=5.0V, V_0=0V			60*	mA
Minimum Source Output Current	I_{SOURCE}	V_{IN+}=1V, V_{IN-}=0V, V_{CC}=15V, V_0=2V V_{IN+}=1V, V_{IN-}=0V, V_{CC}=5.0V, V_0=2V	10			mA
Minimum Output Sink Current	I_{SINK}	V_{IN+}=0V, V_{IN-}=1V, V_{CC}=15V, V_0=2V V_{IN+}=0V, V_{IN-}=1V, V_{CC}=5.0V, V_0=2V V_{IN+}=0V, V_{IN-}=1V, V_{CC}=15V, V_0=0.2V	5 5 12*			mA mA μA
Differential Input Voltage Range	V_{IDR}	All V_{IN}≥GND or V-Supply (if used)			V_{CC}*	V

10.3. Typical Performance Characteristics (AMP)

Input Voltage Range

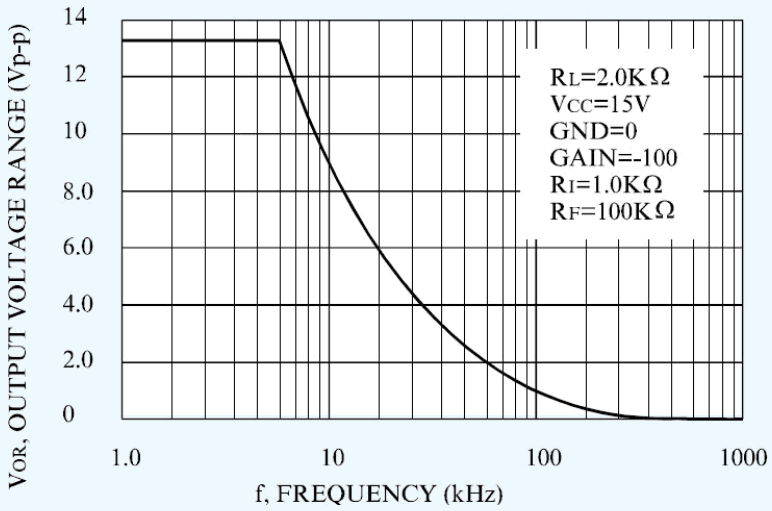


Open-Loop Frequency

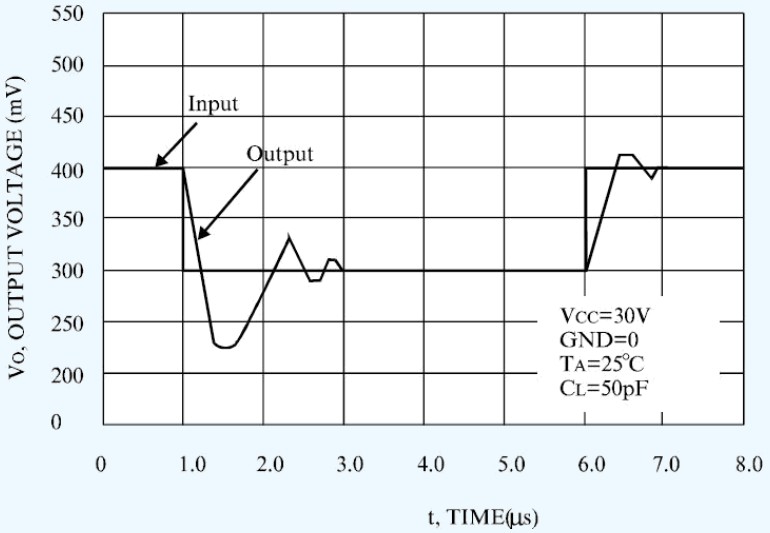


10.3. Typical Performance Characteristics (AMP)

Large-Single Frequency Response

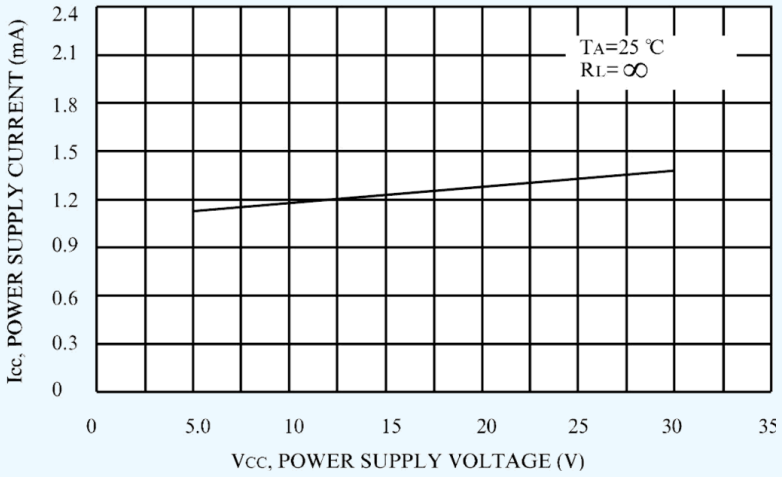


Small-Signal Voltage Follower Pulse Response (Non-Inverting)

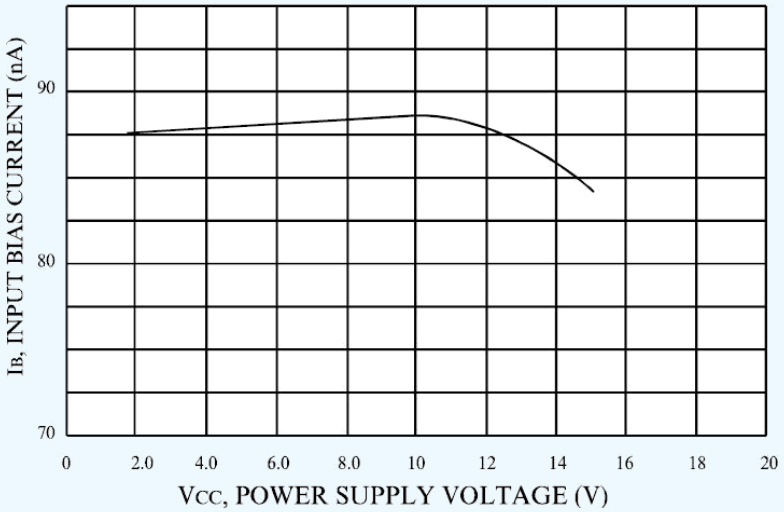


10.3. Typical Performance Characteristics (AMP)

Power Supply Current versus Power Supply Voltage



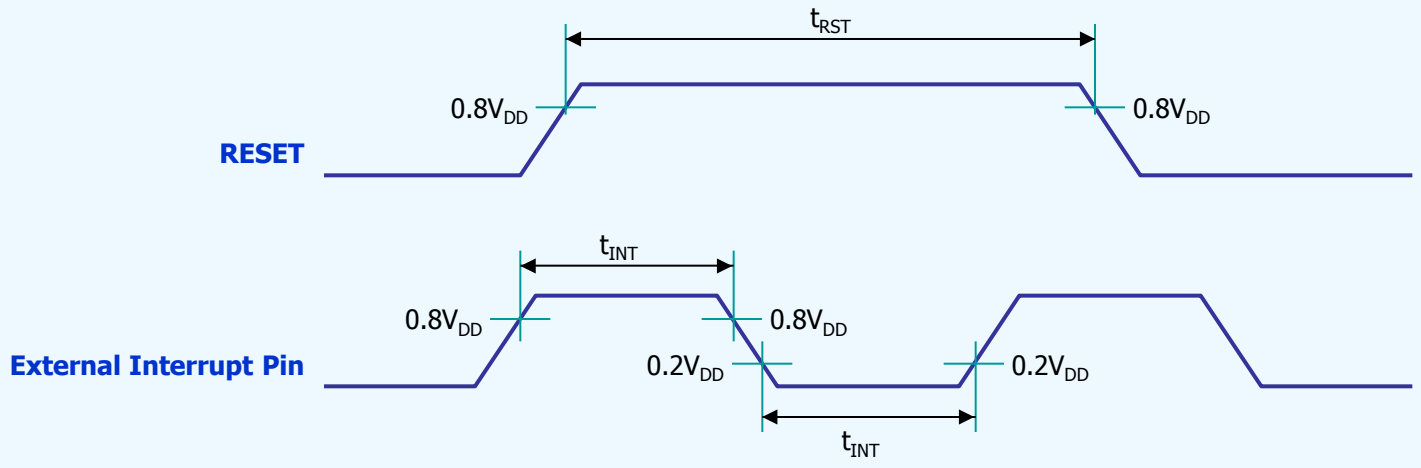
Input Bias Current versus Power Supply Voltage



11. AC Characteristics

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

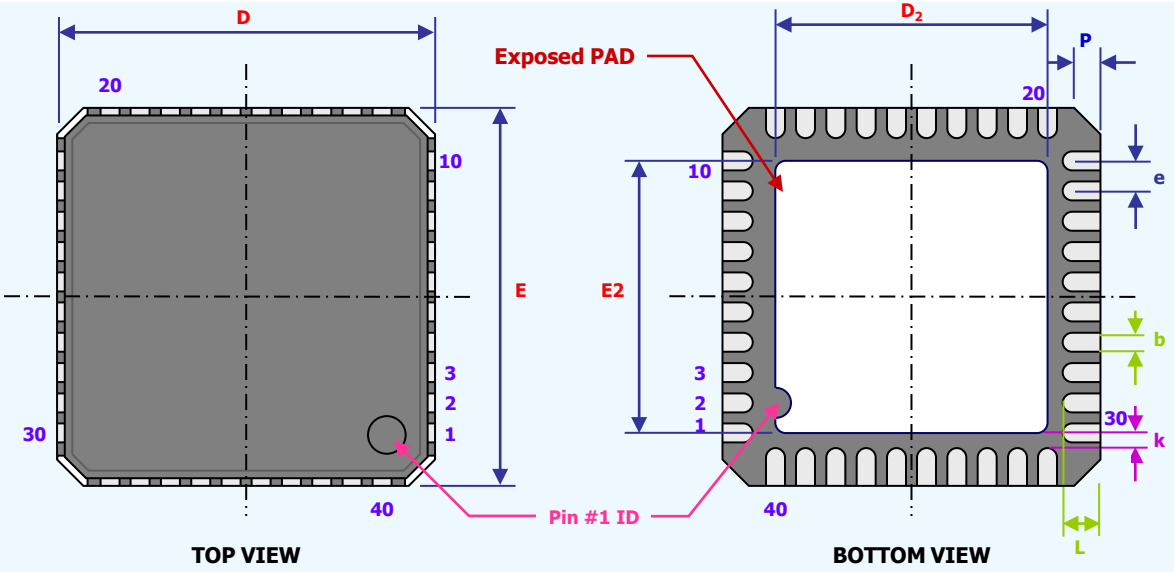
Parameter	Symbol	Pin	Conditions	Value			Unit
				Min.	Typ.	Max.	
Operating Frequency	F_{OSC}	XTAL1, XTAL2	V_{DD} = 5V ± 10%	1	-	24	MHz
			V_{DD} = 3V ± 10%	1	-	12	
RESET Input Width	t_{RST}	RESET	V_{DD} = 5V ± 10%	24	-	-	F_{OSC}
			V_{DD} = 3V ± 10%	24	-	-	
External Interrupt Input Width	t_{INT}	External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



12. 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

13. Package Dimensions : 40-MLF



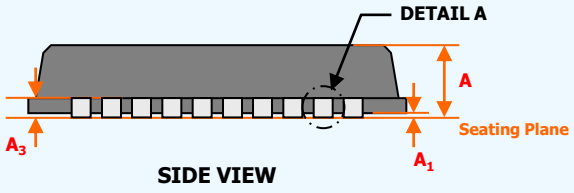
[40-MLF]

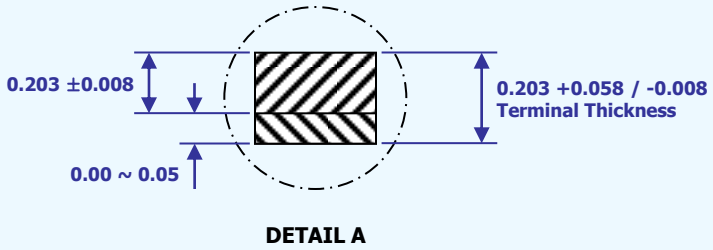
Symbol	Dimension in mm		
	M0in.	Nom.	Max.
A	0.80	0.85	0.90
A_1	0.00	0.01	0.05
A_3		0.20 REF	
D		5.00 BSC	
E		5.00 BSC	
D_2	3.50	3.60	3.70
E_2	3.50	3.60	3.70
b	0.15	0.20	0.25
e		0.40 BSC	
L	0.25	0.35	0.45
k	-	0.35	-

Notes:

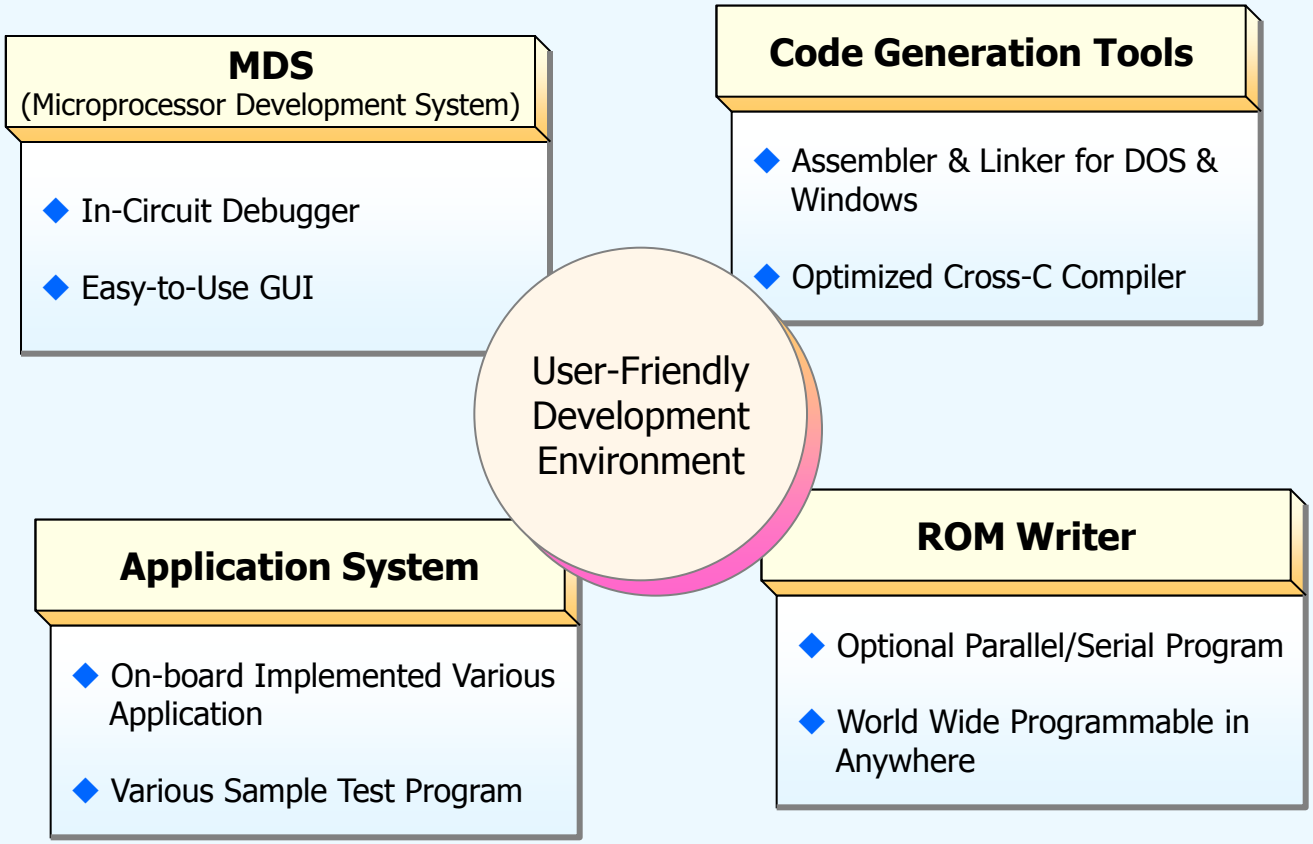
1. All Dimension are in mm. Angles in Degrees.
2. Pin 1 visual index feature may vary, but must be located within the hatched area.
4. Package is saw singulated.
5. Refer JEDEC MO-220.
6. BSC : Basic Dimension. Theoretically exact value shown without tolerances.

REF : Reference Dimension, Usually without tolerance, for information purpose only.





14. Supporting tools



Appendix A : instruction set (1/18)

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 (2/18)

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 (3/18)

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

DEC DPTR

Operation : (DPTR) ← (DPTR) - 1

Encoding : HEX: A5h, #bytes: 1, Cycles: 1

1	0	1	0	0	1	0	1

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 (4/18)

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 (5/18)

ANL C, <src-bit>

Logical AND for bit variables

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

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

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

1	0	0	0	0	0	1	0	bit addr

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

1	0	1	1	0	0	0	0	bit addr

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

Logical OR for byte variables

```
ORL     A, Rn
Operation :   (A)  ← (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 (6/18)

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 (7/18)

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 (8/18)

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 (9/18)

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 (10/18)

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 (11/18)

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 (12/18)

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 (13/18)

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 (14/18)

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

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

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

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

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

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

Appendix A : instruction set (15/18)

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 (16/18)

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 (17/18)

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 (18/18)

DJNZ <byte>, rel

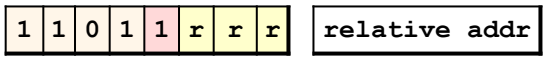
Decrement and Jump if Not Zero

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
```

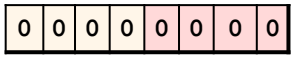
Encoding : HEX: D5h, #bytes: 3, Cycles: 4

NOP

No Operation

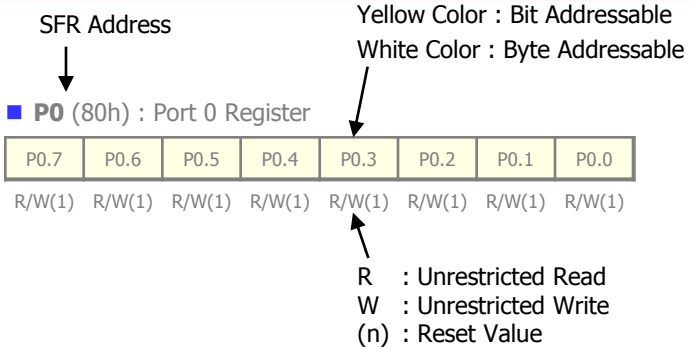
Operation : (PC) ← (PC) + 1

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



Appendix B : SFR Description [80h ~ 84h] (1/19)

[How to Read a SFR Descriptions]



■ **P0** (80h) : Port 0 Register

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

■ **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)

■ **ALTSEL2** (84h) : Port Alternative Function Selection 2

-	PWM_A	UART_B	UART_A	SPID_A	SPIC_A	I2C_A	INT_A
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- PWM_A : PWM alternative A function selection
  PWM3.6_A (P4.4), PWM3.7_A (P4.5)
- UART_B : UART alternative B function selection
  RXD_B (P0.0), TXD_B (P0.1)
- UART_A : UART alternative A function selection
  RXD_A (P4.0), TXD_A (P4.1)
- SPID_A : SPI Data pin alternative A function selection
  MISO_A (P4.4), MOSI_A (P4.5)
- SPIC_A : SPI Clock, Enable pin alternative A function selection
  SSB_A (P0.0), SCLK_A (P0.1)
- I2C_A : I2C alternative A function selection
  I2C_SDA_A (P0.0), I2C_SCL_A (P0.1)
- INT_A : INT alternative A function selection
  INT4_A (P4.2), INT5_A (P4.3)

Appendix B : SFR Description [85h ~ 88h] (2/20)

■ ALTSEL (85h) : Port Alternative Function Selection

-	-	-	-	-	EA_IOEN	XTAL_IOEN	RST_IOEN
					R/W(0)	R/W(1)	R/W(0)

- ◆ EA_IOEN : EA IO function enable
- ◆ XTAL_IOEN : XTAL IO function enable
- ◆ RST_IOEN : RESET IO function enable

■ CKSEL (86h) : Clock Selection Register

-	-	-	R32KOE	R24MOE	-	RGPR	R32KEN
			R/W(0)	R/W(0)		R/W(1)	R/W(1)

- ◆ R32KOE : RING 32K port output enable (P0.3)
- ◆ R32MOE : RING 24M port output enable (P0.2)
- ◆ RGPR : WDT clock selection
  0 = RING 32K is selected as WDT clock.
  1 = RING 24M is selected as WDT clock.
- ◆ R32KEN : RING 32K clock enable flag.

■ PCON (87h) : Power Control Register

SMOD1	SMOD0	-	POF	GF1	GF0	PD	IDL
R/W(0)	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, 2, 3.
- ◆ SMOD0 : Enable SM0 access. Don't modify this bit.
- ◆ POF : Power off flag.
  When power-on, this bit will be set by H/W.
- ◆ GF1, GF0 : General purpose flag.
- ◆ PD : Power-down (Stop) mode bit.
- ◆ IDL : IDL mode bit.

■ 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 control.
- ◆ TF0 : Timer 0 overflow flag.
- ◆ TR0 : Timer 0 run control.
- ◆ IE1 : External interrupt 1 flag.
  If IT1 = 0, cleared by S/W (software).
  If IT1 = 1, cleared by H/W when the interrupt is serviced.
- ◆ IT1 : External interrupt 1 level/edge trigger control.
  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 by H/W when the interrupt is serviced.
- ◆ IT0 : External interrupt 0 level/edge trigger control.
  Edge detect (IT0=1) / Level detect (IT0=0; Default)

Appendix B : SFR Description [89h ~ 8Eh] (3/19)

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

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

- Timer[1]: GATE[7], C/T[6], M1:M0[5:4]
- Timer[0]: GATE[3], C/T[2], M1:M0[1:0]
- GATE : 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 : Counter or Timer Selector. Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from Tx input pin).
- M1, M0 : Mode Selector bits
  - [0 0] Mode 0. 13-bit T/C.
  - [0 1] Mode 1. 16-bit T/C.
  - [1 0] Mode 2. 8-bit Auto-Reload T/C.
  - [1 1] Mode 3.
    (Timer 1) stopped, (Timer 0)
    TL0: 8-bit T/C controlled by the Timer 0 control bits.
    TH0: 8-bit T/C controlled by the Timer 1 control bits.

■ **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)

■ **CKCON** (8Eh) : Clock Control Register

WD2	WD1	WD0	T2M	T1M	T0M	-	-
R/W(1)	R/W(1)	R/W(1)	R/W(0)	R/W(0)	R/W(0)		

- WD2, WD1, WD0 : Watchdog timer mode select
  Refer to WDT section for detailed descriptions.
- T2M, T1M, T0M : Timer 2/1/0 time base selection
  0: time base is 4 clocks.
  1: time base is 12 clocks.

Appendix B : SFR Description [8Fh ~ 93h] (4/19)

■ **RINGCON** (8Fh) : Internal Ring Calibration Control Register

RINGC7	RINGC6	RINGC5	RINGC4	RINGC3	RINGC2	RINGC1	RINGC0
R/W(0)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(0)	R/W(1)	R/W(1)

■ **P1** (90h) : Port 1 Register

P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

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

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

- ◆ IE5~2 : External interrupt 3~2 flag.
- ◆ 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 except when the ring Oscillator provides clock during wake-up from power-down .
- ◆ RGSL : 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. When set, LVD will run in power-down mode.

■ **PWM0CON** (92h) : PWMA CH0 Control Register

-	CPS2	CPS1	CPS0	-	-	PWMOVF	PWMEN
	R/W(0)	R/W(0)	R/W(0)			R/W(0)	R/W(0)

- ◆ CPS2, CPS1, CPS0 : PWMA counter frequency selection.
  [0,0,0] = FOSC / 1 ; Default
  [0,0,1] = FOSC / 2
  [0,1,0] = FOSC / 4
  [0,1,1] = FOSC / 8
  [1,0,0] = FOSC / 16
  [1,0,1] = FOSC / 32
  [1,1,0] = FOSC / 64
  [1,1,1] = FOSC / 128
- ◆ PWMOVF : PWMA counter overflow flag.
  Set by hardware and cleared by software.
  PWMOVF flags an interrupt.
- ◆ PWMEN : PWMA counter run control bit.
  [0] = Stop the PWMA counter.
  [1] = Run the PWMA counter.

■ **PWM0CNT**(93h) : PWMA CH0 Counter Register

CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- ◆ Software can write this register for the initialization of the counter.

■ **PWM0D0** (94h) : PWMA CH0 Duty Data Register of Module 0

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)

◆ Each Module has a internal buffer register for the duty data register.
The buffer register is updated with the new data whenever the PWMA counter rolls over.
When user write, the data register is written.
When user read, the contents of buffer register is read out.

■ **PWM0D1** (95h) : PWMA CH0 Duty Data Register of Module 1

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)

■ **PWM0D2** (96h) : PWMA CH0 Duty Data Register of Module 2

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)

■ **PWM0D3** (97h) : PWMA CH0 Duty Data Register of Module 3

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)

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

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

◆ SM0, SM1 : Serial Port mode selector.
[0,0] : Mode0, 8-bit shift register (F_{OSC}/4)
[0,1] : Mode1, 8-bit UART (Variable)
[1,0] : Mode2, 9-bit UART (F_{OSC}/32 or F_{OSC}/16)
[1,1] : Mode3, 9-bit UART (Variable)
◆ SM2 : Enables the Automatic Address Recognition in Mode2 and 3.
In Mode1, the validity of a Stop Bit is checked if SM2=1
In Mode0, SM2 should be "0".
◆ REN : Enable/Disable reception.
◆ TB8 : 9th data bit that will be transmitted in Mode2 and 3.
◆ RB8 : 9th data bit that was received in Mode 2 and 3.
In Mode1, RB8 is equal to stop bit if SM2 is "0".
In Mode0, RB8 is not used.
◆ 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)

◆ The transmission buffer and the reception buffer are separated.
◆ The transmission/reception buffers have the same address.

■ **PWM0OEN** (9Bh) : PWMA CH0 Module Output Enable

-	-	OE5	OE4	OE3	OE2	OE1	OE0
		R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- OE5 : Module 5 PWM output enable.
- OE4 : Module 4 PWM output enable.
- OE3 : Module 3 PWM output enable.
- OE2 : Module 2 PWM output enable.
- OE1 : Module 1 PWM output enable.
- OE0 : Module 0 PWM output enable.

■ **PWM0D4** (9Ch) : PWMA CH0 Duty Data Register of Module 4

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)

■ **PWM0D5** (9Dh) : PWMA CH0 Duty Data Register of Module 5

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)

■ **PWM0D6** (9Eh) : PWMA CH0 Duty Data Register of Module 6

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)

■ **PWM0D7** (9Fh) : PWMA CH0 Duty Data Register of Module 7

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)

■ **P2** (A0h) : Port 2 Register

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

■ **EIE** (A1h) : Extended Interrupt Enable Register

ESPI	-	EI2C	EWDT	-	-	EX3	EX2
R/W(0)		R/W(0)	R/W(0)			R/W(0)	R/W(0)

- ESPI : SPI interrupt enable.
- EI2C : I2C interrupt enable.
- EWDT : Watchdog timer interrupt enable.
- EX3 : External interrupt 3 enable.
- EX2 : External interrupt 2 enable.

■ **PWM1CON** (A2h) : PWMA CH1 Control Register

-	CPS2	CPS1	CPS0	-	-	PWMOVF	PWMEN
	R/W(0)	R/W(0)	R/W(0)			R/W(0)	R/W(0)

◆ CPS2, CPS1, CPS0 : PWMA counter frequency selection.
[0,0,0] = FOSC / 1 ; Default
[0,0,1] = FOSC / 2
[0,1,0] = FOSC / 4
[0,1,1] = FOSC / 8
[1,0,0] = FOSC / 16
[1,0,1] = FOSC / 32
[1,1,0] = FOSC / 64
[1,1,1] = FOSC / 128

◆ PWMOVF : PWMA counter overflow flag.
Set by hardware and cleared by software.
PWMOVF flags an interrupt.

◆ PWMEN : PWMA counter run control bit.
[0] = Stop the PWMA counter.
[1] = Run the PWMA counter.

■ **PWM1CNT**(A3h) : PWMA CH1 Counter Register

CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

◆ Software can write this register for the initialization of the counter.

■ **PWM1D0** (A4h) : PWMA CH1 Duty Data Register of Module 0

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)

◆ Each Module has a internal buffer register for the duty data register.
The buffer register is updated with the new data whenever the PWMA counter rolls over.
When user write, the data register is written.
When user read, the contents of buffer register is read out.

■ **PWM1D1** (A5h) : PWMA CH1 Duty Data Register of Module 1

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)

■ **PWM1D2** (A6h) : PWMA CH1 Duty Data Register of Module 2

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)

■ **PWM1D3** (A7h) : PWMA CH1 Duty Data Register of Module 3

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)

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

EA	EADC	ET2	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)	R/W(0)

- EA : Enable/Disable all interrupts.
- EADC : ADC interrupt enable.
- ET2 : Timer 2 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.

■ **SADDR** (A9h) : Slave Address Register

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

- Programmed with the given or broadcast address assigned to serial port

■ **PWM1OEN** (ABh) : PWMA CH1 Module Output Enable

OE7	OE6	OE5	OE4	OE3	OE2	OE1	OE0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- OE7 : Module 7 PWM output enable.
- OE6 : Module 6 PWM output enable.
- OE5 : Module 5 PWM output enable.
- OE4 : Module 4 PWM output enable.
- OE3 : Module 3 PWM output enable.
- OE2 : Module 2 PWM output enable.
- OE1 : Module 1 PWM output enable.
- OE0 : Module 0 PWM output enable.

■ **PWM1D4** (ACh) : PWMA CH1 Duty Data Register of Module 4

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)

■ **PWM1D5** (ADh) : PWMA CH1 Duty Data Register of Module 5

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)

■ **PWM1D6** (AEh) : PWMA CH1 Duty Data Register of Module 6

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)

■ **PWM1D7** (AFh) : PWMA CH1 Duty Data Register of Module 7

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 [B0h ~ B4h] (9/19)

■ **P3** (B0h) : Port 3 Register

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

■ **EIP** (B1h) : Extended Interrupt Priority Register

PSPI	-	PI2C	PWDT	PX5	-	-	PX2
R/W(0)		R/W(0)	R/W(0)	R/W(0)			R/W(0)

- ◆ PSPI : SPI interrupt priority bit.
- ◆ PI2C : I2C interrupt priority bit.
- ◆ PWDT : Watchdog timer interrupt priority bit.
- ◆ PX3 : External interrupt 3 priority bit.
- ◆ PX2 : External interrupt 2 priority bit.

■ **IT** (B2h) : Interrupt Type Selection Register

-	-	-	-	IT5	IT4	IT3	IT2
				R/W(1)	R/W(1)	R/W(1)	R/W(1)

- ◆ IT5 : Interrupt5 Type Selection Flag
  [0] : Level detect [1] : Edge detect
- ◆ IT4 : Interrupt4 Type Selection Flag
  [0] : Level detect [1] : Edge detect
- ◆ IT3 : Interrupt3 Type Selection Flag
  [0] : Level detect [1] : Edge detect
- ◆ IT2 : Interrupt2 Type Selection Flag
  [0] : Level detect [1] : Edge detect

■ **P4** (B3h) : Port 4 Register

P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

■ **SPICON** (B4h) : SPI Control Register

-	MODE	BORD	MSSEL	CKPOL	CKPHA	SPIOEN	SPIEN
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- ◆ MODE : SPI mode selection
  [0] : 4-wire mode [1] : 3-wire mode
- ◆ BORD : SPI Transfer Bit Order
  [1] : First LSB, Last MSB [0] : First MSB, Last LSB
- ◆ MSSEL : SPI Master / Slave Selection Flag
  [1] : SPI Master Mode [0] : SPI Slave Mode
- ◆ CKPOL, CKPHA : SPI clock Polarity & Phase
  [0,0] : Leading edge Rising, Leading edge Sampling
  [0,1] : Leading edge Rising, Trailing edge Sampling
  [1,0] : Leading edge Falling, Leading edge Sampling
  [1,1] : Leading edge Falling, Trailing edge Sampling
- ◆ SPIOEN : SPI Output Enable
  [1] : SPI Output Enable [0] : SPI Output Disable
- ◆ SPIEN : SPI Enable Flag
  [1] : SPI Enable [0] : SPI Disable

Appendix B : SFR Description [B5h ~ B9h] (10/19)

■ **SPICK** (B5h) : SPI Clock Control Register

-	-	-	-	-	SPICK2	SPICK1	SPICK0
					R/W(0)	R/W(0)	R/W(0)

◆ SPICK[2:0] : SPI Master Clock Divider

[0,0,0] : Fosc / 2 [0,0,1] : Fosc / 4
[0,1,0] : Fosc / 8 [0,1,1] : Fosc / 16
[1,0,0] : Fosc / 32 [1,0,1] : Fosc / 64
[1,1,0] : Fosc / 128 [1,1,1] : Fosc / 256

■ **SPIDR** (B6h) : SPI TX / RX Data Register

DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

■ **IPH** (B7h) : Interrupt Priority High Register

-	PADCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

◆ PADCH : ADC interrupt priority high.
◆ PT2H : Timer 2 interrupt priority high.
◆ PSH : Serial Port (UART) interrupt priority high.
◆ PT1H : Timer 1 interrupt priority high.
◆ PX1H : External interrupt 1 priority high.
◆ PT0H : Timer 0 interrupt priority high.
◆ PX0H : External interrupt 0 priority high.

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

-	PADC	PT2	PS	PT1	PX1	PT0	PX0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

◆ PADC : ADC interrupt priority low.
◆ PT2 : Timer 2 interrupt priority low.
◆ PS : Serial port (UART) interrupt priority low.
◆ PT1 : Timer 1 interrupt priority low.
◆ PX1 : External interrupt 1 priority low.
◆ PT0 : Timer 0 interrupt priority low.
◆ PX0 : External interrupt 0 priority low.

■ **SADEN** (B9h) : Slave Address Mask Enable Register

SADEN.7	SADEN.6	SADEN.5	SADEN.4	SADEN.3	SADEN.2	SADEN.1	SADEN.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 [BAh ~ C4h] (11/19)

■ **ITSEL** (BAh) : Interrupt Polarity Selection Register

-	-	ITSEL5	ITSEL4	ITSEL3	ITSEL2	ITSEL1	ITSEL0
-	-	R/W(0)	R/W(1)	R/W(0)	R/W(1)	R/W(0)	R/W(0)

◆ ITSEL5 : Interrupt5 Polarity Selection Flag
[0] : low level or negative edge detect
[1] : high level or positive edge detect

◆ ITSEL4 : Interrupt4 Polarity Selection Flag
[0] : low level or negative edge detect
[1] : high level or positive edge detect

◆ ITSEL3 : Interrupt3 Polarity Selection Flag
[0] : low level or negative edge detect
[1] : high level or positive edge detect

◆ ITSEL2 : Interrupt2 Polarity Selection Flag
[0] : low level or negative edge detect
[1] : high level or positive edge detect

◆ ITSEL1 : Interrupt1 Polarity Selection Flag
[0] : low level or negative edge detect
[1] : high level or positive edge detect

◆ ITSEL0 : Interrupt0 Polarity Selection Flag
[0] : low level or negative edge detect
[1] : high level or positive edge detect

■ **SPIST** (C0h) : SPI Status Register

-	-	-	-	TXBV	SPIF	SPICOL	SPIOF
				R(0)	R/W(0)	R/W(0)	R/W(0)

◆ TXBV : TX buffer of SPIDR holds valid data.
[1] : Set by H/W when user write SPIDR while SPI is enabled.
[0] : Cleared by H/W when the data is moved to TX shift register or SPI is disabled.

◆ SPIF : SPI Interrupt Flag
[1] : Serial transfer is complete. If SPIE is set and EA is set, SPI interrupt is generated.

◆ SPICOL : SPI Write Collision Flag
[1] : SPIDR is written when TXBV is set. The previous data is lost.

◆ SPIOF : SPI Read Overflow Flag
[1] : If a new data is received while SPIDR is still holding the previous data, the flag is set.
SPIF must be cleared before receiving a data again.

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

-	-	-	-	XTOFF	ALEOFF	-	-
				R/W(1)	R/W(0)		

◆ XTOFF : [1] : External crystal Oscillator disable (Default).
[0] : External crystal Oscillator enable.

◆ ALEOFF : [1] : ALE toggling disable.
[0] : ALE toggling enable (Default).

Appendix B : SFR Description [C5h ~ CBh] (12/19)

■ STATUS (C5h) : Crystal Status Register

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

◆ XTUP : Crystal oscillator warm-up status.
This bit is cleared by H/W during executing Power-on reset or during exiting from the power-down mode.
It is set by H/W after XTAL stabilization.

■ OSCICN (C6h) : Internal Ring Oscillator Control Register

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

◆ RINGON : 1 = Internal ring Oscillator is running.
0 = Internal ring Oscillator is killed.
Don't clear RINGON bit when XTRG = 0.

◆ DIV[2:0] : Ring Oscillator divider. (F_{OSC} : 24MHz)
[0,0,0] = $F_{OSC}/48$
[0,0,1] = $F_{OSC}/24$
[0,1,0] = $F_{OSC}/12$
[0,1,1] = $F_{OSC}/8$
[1,0,0] = $F_{OSC}/6$
[1,0,1] = $F_{OSC}/4$
[1,1,0] = $F_{OSC}/2$

■ OSCICN2 (C7h) : Internal Ring Oscillator Control Register

-	-	-	-	-	RINGON2	-	-
					R/W(1)		

◆ RINGON2 : 1 = Test Internal ring Oscillator is running.
0 = Test Internal ring Oscillator is killed.
Must clear the flag for saving power at STOP mode

■ T2CON (C8h) : Timer/Counter 2 Control Register

TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

◆ TF2 : Timer 2 overflow flag.
◆ EXF2 : Timer 2 external flag.
◆ RCLK : Receive clock flag.
◆ TCLK : Transmit clock flag.
◆ EXEN2 : Timer 2 external enable flag.
◆ TR2 : Timer 2 run flag.
◆ C/T2 : Timer 2 Timer/Counter select. When set, counter by T2.
◆ CP/RL2 : Capture/Reload flag.
CP/RL2 = 0, Reload. (TH2,TL2) ← (RCAP2H,RCAP2L)
CP/RL2 = 1, Capture. (RCAP2H,RCAP2L) ← (TH2,TL2)

■ T2MOD (C9h) : Timer/Counter 2 Mode Control Register

-	-	-	-	-	-	T2OE	DCEN
						R/W(0)	R/W(0)

◆ T2OE : Timer 2 clock output enable. When set, clock output to P1.0.
◆ DCEN : Timer 2 down count enable. When set, count down.

■ RCAP2L (CAh) : Timer/Counter 2 Capture/Reload Low Byte Register

RCAP2L.7	RCAP2L.6	RCAP2L.5	RCAP2L.4	RCAP2L.3	RCAP2L.2	RCAP2L.1	RCAP2L.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)

■ RCAP2H (CBh) : Timer/Counter 2 Capture/Reload High Byte Register

RCAP2H.7	RCAP2H.6	RCAP2H.5	RCAP2H.4	RCAP2H.3	RCAP2H.2	RCAP2H.1	RCAP2H.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)

■ **TL2** (CCh) : Timer/Counter 2 Low Byte Register

TL2.7	TL2.6	TL2.5	TL2.4	TL2.3	TL2.2	TL2.1	TL2.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

■ **TH2** (CDh) : Timer/Counter 2 High Byte Register

TH2.7	TH2.6	TH2.5	TH2.4	TH2.3	TH2.2	TH2.1	TH2.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

■ **ADCENB0** (CEh) : ADC Channel Enable Bar Register (P0 port)

-	-	-		ADCENB0.3	ADCENB0.2	ADCENB0.1	ADCENB0.0
				R/W(1)	R/W(1)	R/W(1)	R/W(1)

◆ 0 = ADC0 channel ON / 1 = ADC0 channel OFF (Default)

■ **ADCENB1** (CFh) : ADC Channel Enable Bar Register (P1 port)

-	-	ADCENB1.5	ADCENB1.4	ADCENB1.3	ADCENB1.2	ADCENB1.1	ADCENB1.0
		R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

◆ 0 = ADC1 channel ON / 1 = ADC0 channel OFF (Default)

■ **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
[0,1] : Bank 1
[1,0] : Bank 2
[1,1] : Bank 3
◆ OV : Overflow flag.
◆ F1 : User flag 1.
◆ P : Parity bit. Set/clear by H/W according to ACC odd parity.

■ **P0SEL** (D1h) : Port 0 Pull-up Control Register

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

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

■ **P1SEL** (D2h) : Port 1 Pull-up Control Register

P1SEL.7	P1SEL.6	P1SEL.5	P1SEL.4	P1SEL.3	P1SEL.2	P1SEL.1	P1SEL.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

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

Appendix B : SFR Description [D3h ~ D8h] (14/19)

■ **P2SEL** (D3h) : Port 2 Pull-up Control Register

P2SEL.7	P2SEL.6	P2SEL.5	P2SEL.4	P2SEL.3	P2SEL.2	P2SEL.1	P2SEL.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

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

■ **P3SEL** (D4h) : Port 3 Pull-up Control Register

P3SEL.7	P3SEL.6	P3SEL.5	P3SEL.4	P3SEL.3	P3SEL.2	P3SEL.1	P3SEL.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

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

■ **P4SEL** (D5h) : Port 4 Pull-up/down Control Register

P4SEL.7	P4SEL.6	P4SEL.5	P4SEL.4	P4SEL.3	P4SEL.2	P4SEL.1	P4SEL.0
R/W(0)	R/W(0)	R/W(1)	R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- P4SEL.7 : 0 = Pull-down resistor ON (Default)
  1 = Pull-down resistor OFF.
- P4SEL.5 : 0 = Pull-up resistor ON.
  1 = Pull-up resistor OFF (Default).
- P4SEL.4 : 0 = Pull-up resistor ON.
  1 = Pull-up resistor OFF (Default).
- P4SEL.X : 0 = Pull-up resistor ON (Default).
  1 = Pull-up resistor OFF.

■ **ADCENB2** (D6h) : ADC Channel Enable Bar Register (P2 port)

ADCENB2.7	ADCENB2.6	ADCENB2.5	ADCENB2.4	ADCENB2.3	ADCENB2.2	ADCENB2.1	ADCENB2.0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

- 0 = ADC2 channel ON / 1 = ADC0 channel OFF (Default)

■ **ADCENB3** (D7h) : ADC Channel Enable Bar Register (P3 port)

-	-	ADCENB3.5	ADCENB3.4	ADCENB3.3	ADCENB3.2	ADCENB3.1	ADCENB3.0
		R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

- 0 = ADC3 channel ON / 1 = ADC0 channel OFF (Default)

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

WDMOD	POR	EPFI	PFI	WDIF	WTRF	EWT	RWT
R/W(0)	R/W(1)	R/W(0)	R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- WDMOD : WDT mode selection flag.
- POR : Power-on reset flag.
- 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.

Appendix B : SFR Description [D9h ~ DFh] (15/19)

■ **P0TYPE**(D9h) : Port 0 Type Register

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

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

■ **P1TYPE**(DAh) : Port 1 Type Register

P1TYPE.7	P1TYPE.6	P1TYPE.5	P1TYPE.4	P1TYPE.3	P1TYPE.2	P1TYPE.1	P1TYPE.0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

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

■ **P2TYPE**(DBh) : Port 2 Type Register

P2TYPE.7	P2TYPE.6	P2TYPE.5	P2TYPE.4	P2TYPE.3	P2TYPE.2	P2TYPE.1	P2TYPE.0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

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

■ **P3TYPE**(DCh) : Port 3 Type Register

P3TYPE.7	P3TYPE.6	P3TYPE.5	P3TYPE.4	P3TYPE.3	P3TYPE.2	P3TYPE.1	P3TYPE.0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

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

■ **P4TYPE**(DDh) : Port 4 Type Register

P4TYPE.7	P4TYPE.6	P4TYPE.5	P4TYPE.4	P4TYPE.3	P4TYPE.2	P4TYPE.1	P4TYPE.0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

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

■ **ADCON** (DEh) : ADC Control & ADC Result Low Register

AD_EN	AD_REQ	AD_END	ADCF	-	-	SAR1	SAR0
R/W(0)	R/W(0)	R(1)	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.
User must check the ADCF instead of AD_END.

◆ ADCF : ADC Interrupt Flag.
Must be cleared by S/W.

◆ SAR[1:0] : Low Bits of ADC Result Value. (Total 10 bits)

■ **ADCSEL** (DFh) : ADC Clock and MUX Selection Register

ADIV2	ADIV1	ADIV0	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

◆ ADIV[2:0] : ADC clock selection.
[000] : FSYS / 2.
[001] : FSYS/ 4.
[010] : FSYS / 8.
[011] : FSYS / 16.
[100] : FSYS / 32.
[101] : FSYS / 64.
[110] : FSYS / 128.
[111] : FSYS / 256.

◆ ADCS[4:0] : ADC channel selection
[00000] : ADC0.0 channel selection.
[00001] : ADC0.1 channel selection. (continued)

◆ ADCS[4:0] : ADC channel selection

[00010] : ADC0.2 channel selection.
[00011] : ADC0.3 channel selection.
[00100] : ADC0.4 channel selection.
[00101] : ADC0.5 channel selection.
[00110] : ADC0.6 channel selection.
[00111] : ADC0.7 channel selection.
[01000] : ADC1.0 channel selection.
[01001] : ADC1.1 channel selection.
[01010] : ADC1.2 channel selection.
[01011] : ADC1.3 channel selection.
[01100] : ADC1.4 channel selection.
[01101] : ADC1.5 channel selection.
[01110] : ADC1.6 channel selection.
[01111] : ADC1.7 channel selection.
[10000] : ADC2.0 channel selection.
[10001] : ADC2.1 channel selection.
[10010] : ADC2.2 channel selection.
[10011] : ADC2.3 channel selection.
[10100] : ADC2.4 channel selection.
[10101] : ADC2.5 channel selection.
[10110] : ADC2.6 channel selection.
[10111] : ADC2.7 channel selection.
[11000] : ADC3.0 channel selection.
[11001] : ADC3.1 channel selection.
[11010] : ADC3.2 channel selection.
[11011] : ADC3.3 channel selection.
[11100] : ADC3.4 channel selection.
[11101] : ADC3.5 channel selection.
[11110] : ADC3.6 channel selection.
[11111] : ADC3.7 channel selection.

■ **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)

■ **P0DIR** (E1h) : Port 0 Input/Output Control Register

P0DIR.7	P0DIR.6	P0DIR.5	P0DIR.4	P0DIR.3	P0DIR.2	P0DIR.1	P0DIR.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)

◆ 1 = Input / 0 = Output (Default).

■ **P1DIR** (E2h) : Port 1 Input/Output Control Register

P1DIR.7	P1DIR.6	P1DIR.5	P1DIR.4	P1DIR.3	P1DIR.2	P1DIR.1	P1DIR.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)

◆ 1 = Input / 0 = Output (Default).

■ **P2DIR** (E3h) : Port 2 Input/Output Control Register

P2DIR.7	P2DIR.6	P2DIR.5	P2DIR.4	P2DIR.3	P2DIR.2	P2DIR.1	P2DIR.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)

◆ 1 = Input / 0 = Output (Default).

■ **P3DIR** (E4h) : Port 3 Input/Output Control Register

P3DIR.7	P3DIR.6	P3DIR.5	P3DIR.4	P3DIR.3	P3DIR.2	P3DIR.1	P3DIR.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)

◆ 1 = Input / 0 = Output (Default).

■ **P4DIR** (E5h) : Port 4 Input/Output Control Register

P4DIR.7	P4DIR.6	P4DIR.5	P4DIR.4	P4DIR.3	P4DIR.2	P4DIR.1	P4DIR.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)

◆ 1 = Input / 0 = Output (Default).

■ **ADCR** (E6h) : 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)

■ **I2CST** (E8h) : 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)

Appendix B : SFR Description [E9h ~ ECh] (18/19)

I2CCON (E9h) : 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

I2CCFG (EAh) : 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)

I2CSLA (EBh) : 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.

I2CDAT (ECh) : 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)

I2CSCL (EDh) : 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)

Appendix B : SFR Description [F0h ~ F7h] (19/19)

■ **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)

■ **EECNTLD** (F1h) : 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.

■ **EECNTL** (F2h) : EEPROM Erase/Program Time Count LSB

EECNT7	EECNT6	EECNT5	EECNT4	EECNT3	EECNT2	EECNT1	EECNT0
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** (F3h) : EEPROM Erase/Program Time Count MSB

EECNT15	EECNT14	EECNT13	EECNT12	EECNT11	EECNT10	EECNT9	EECNT8
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

■ **EECON** (F6h) : EEPROM Control Register

-	EFLAG	-	EDONE	EWF	ECOMM	EWST	EINIT
	R/W(0)		R(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

◆ EFLAG : EEPROM mode flag.
◆ EDONE : EEPROM erase / program done.
◆ EWF : EEPROM erase / program mode selection flag.
◆ ECOMM : EEPROM common operation start.
◆ EWST : EEPROM erase / program start.
◆ EINIT : EEPROM initialization operation start.

■ **EEAEN** (F7h) : EEPROM Access Enable Register

-	-	-	-	-	-	MDSF	EEAEN
						R(0)	R/W(0)

◆ MDSF : MDS mode flag.
◆ EEAEN : EEPROM access enable flag.

Appendix C : Update History

- V1.0
  - ✓ Initial Release