BM

Brief Manual of MiDAS400 Family

Flash /ISP / IAP
8-bit Turbo Microcontrollers

V1.9

May 2011

Contents

1. Product Overview

- **CORERIVER's MiDAS400 Family is a group of fast 80C52 compatible microcontrollers.**
- **The instruction execution of MiDAS400 Family is max. 3 times faster than that of traditional 80C52.**
  - ✓ **1 machine cycle = 4 clocks vs. 12 clocks**
- **Additional peripherals of MiDAS400 Family:**
  - ✓ **I2C / WDT / LVD / POR.**
- **Power saving modes**
- **Noise tolerant scheme**
- **Provides Easy-to-Use training-kit system**

1. Product Overview (Cont'd)

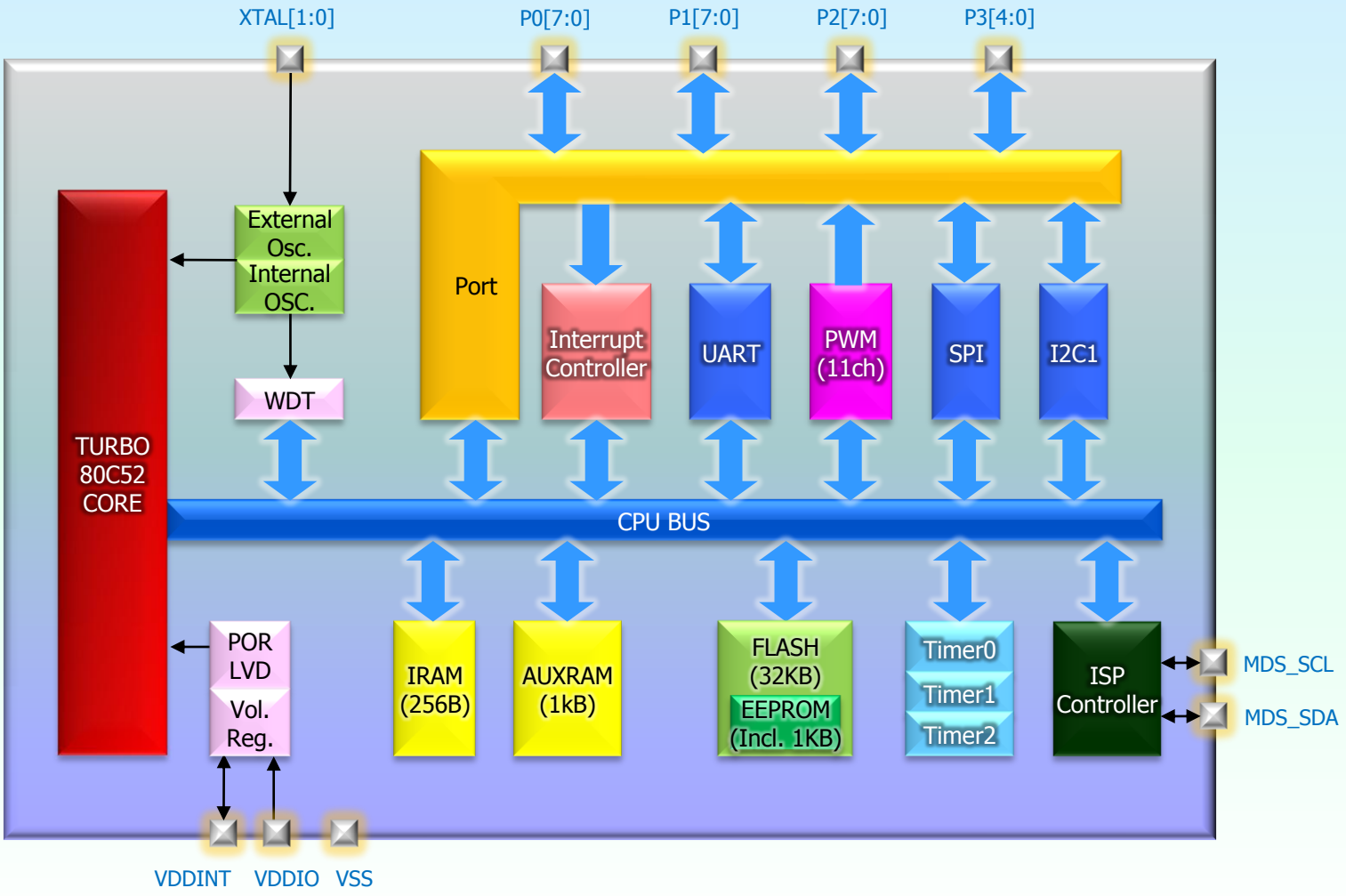
A. MiDAS400 Family

Product	FLASH [Byte]	EEPROM [Byte]	RAM [Byte]	Volt [V]	Freq [MHz]	T/C [16 bits]	COM I/O	WDT	ADC (bit X Ch)	PWM (bit X ch)	I/O Pins	Package	Others
GC400-ML32I	32k	(1k)	1k + 256	2.7 ~ 3.6	48	3	1 UART 1 SPI 2 I2C	1	None	8 X 12	29	32-MLF	IAP ISP EJTAG LVD POR RING
GC410-ML32I	16k												

2. Features

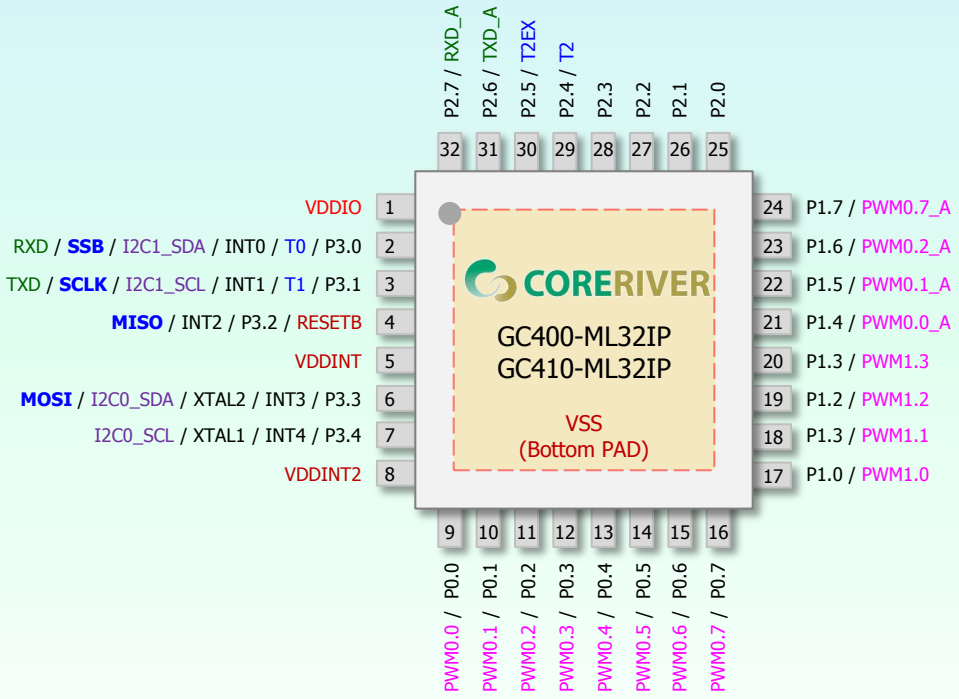
- CPU
  - ✓ 8-bit turbo 80C52 architecture
  - ✓ 4 cycles/1 machine cycle
  - ✓ instruction level compatible with Intel 80C52
- 32kB (GC400) / 16kB (GC410) FLASH (Including 1kB User EEPROM)
- 1kB Internal Aux. RAM
- 256B Internal RAM
- Operating Voltage : +2.7V to +3.6V
- Operating Frequency : Max. 48MHz
- Programmable 29 I/O Pins
  - ✓ Pull-up control(2 I/O Pins), Open drain, Push-Pull output
  - ✓ TTL and CMOS compatible logic levels
- Low Voltage Detector (LVD) : +1.6V
- Internal Ring OSC with Calibration function
  - ✓ Max. 48MHz @3.0V
  - ✓ 48MHz @ 3.0V (+/- 3%)
- Supporting ISP/IAP/MDS
- Three 16-bit Timer/Counters
- 26-bit Programmable Watchdog Timer(WDT)
- 1-channel UART Comm.
- **2-channel I2C Comm. (Master/Slave, Slave)**
- **1-channel SPI Comm.**
- **12-channel 8-bit High Speed PWM for LED Dimming**
- Max. 13 Interrupt Sources
  - ✓ Timer0/1/2, WDT, I2C0/1, SPI, UART
  - ✓ 5 External Interrupt Sources : Both Edge/Level
  - ✓ Four/Two-level Interrupt Priority
- Reset Sources
  - ✓ On-chip Power-On-Reset (POR)
  - ✓ External Reset
  - ✓ Low Voltage Detector Reset (LVR)
  - ✓ Watchdog Timer Reset
- Power Down Wake-up Sources
  - ✓ Reset Sources + 5 External Interrupt (Both Levels)
  - ✓ WDT interrupt
- Power Consumption
  - ✓ Active Current : Max. 1mA @+3.3V, 2MHz
  - ✓ Idle Current : Max. 0.5mA @+3.3V, 2MHz
  - ✓ Stop Current : Max. 60uA @+3.3V
- E.S.D. Protection up to 8,000V
- Latch-up Protection Up to ±200mA
- Package
  - ✓ 32-MLF (5mm X 5mm)

3. Block Diagram

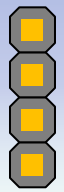


3. Pin Configuration

- Ordering Information : GC400-ML32IP / GC410-ML32IP
- ISP / MDS : 1 Channel
- I2C Comm. : 2 Channel
- SPI Comm. : 1 Channel
- UART Comm. : 1 Channel
- PWM : 12 Channels
- I/O : 29 Pins



ISP / MDS Pin Configuration



VDDIO (#1)
VSS (Bottom PAD)
I2C1_SCL (#3)
I2C1_SDA (#2)

[32-pin MLF : 5mm X 5mm, 0.5mm Pin Pitch **]**

5. Pin Descriptions

Symbol	Direction	Description	Share Pins
VDDIO	Input	Digital I/O Power	-
VDDINT	Output	Digital Power Input/Output (+1.8V)	-
VDDINT2	Output	Digital Power Input/Output (+1.8V)	-
RESETB	Input	External Reset (Active Low)	-
I2C1_SDA	Input/Output	Serial Data pin for ISP/MDS.	• P3.0 : INT0 / I2C1_SDA / RXD
I2C1_SCL	Input/Output	Serial Clock pin for ISP/MDS.	• P3.1 : INT1 / I2C0_SCL / SCLK / TXD
P3[4:0]	Input/Output	◆ A 5-bit open-drain or push-pull I/O port. - Optional Pull-up Control Enable (Only P3[1:0]) • P3.0 → I2C1_SDA : I2C1 Serial Data • P3.1 → I2C1_SCL : I2C1 Serial Clock • P3.0 → SSB : SPI Slave Select Bar • P3.1 → SCLK : SPI Serial Clock • P3.2 → MISO : SPI Master Input Slave Output • P3.3 → MOSI : SPI Master Output Slave Input • P3.3 → I2C0_SDA : I2C0 Serial Data • P3.4 → I2C0_SCL : I2C0 Serial Clock • P3.0 → RXD : UART Serial Port Input • P3.1 → TXD : UART Serial Port Output • P3.0 → INT0 : External Interrupt 0 (Positive/Negative Edge) • P3.1 → INT1 : External Interrupt 1 (Positive/Negative Edge) • P3.2 → INT2 : External Interrupt 2 (Positive/Negative Edge) • P3.3 → INT3 : External Interrupt 3 (Positive/Negative Edge) • P3.4 → INT4 : External Interrupt 4 (Positive/Negative Edge)	• P3.0 : INT0 / I2C1_SDA / SSB / RXD • P3.1 : INT1 / I2C1_SCL / SCLK / TXD • P3.2 : RESETB / INT2 / MISO I2C0_SDA • P3.3 : INT3 / XTAL2 / MOSI / I2C0_SDA • P3.4 : INT4 / XTAL1 / I2C0_SCL

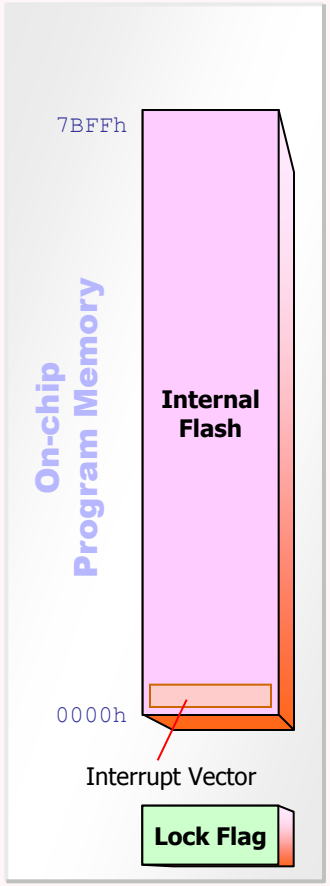
5. Pin Descriptions (Cont'd)

Symbol	Direction	Description	Share Pins
VSS	Input	Digital Ground	-
XTAL1	Input	Input to the inverting oscillator amplifier	• P3.4 / INT4 / XTAL1 / I2C0_SCL
XTAL2	Output	Output to the inverting oscillator amplifier	• P3.3 / INT3 / XTAL2 / I2C0_SDA / MOSI
P0[7:0]	Input/Output	◆ An 8-bit open-Drain or push-pull I/O port. ◆ Note that the output is fully driven (push-pull) when P0 drives PWM0 output. • P0.0 → PWM0.0 : PWM output 0.0 • P0.1 → PWM0.1 : PWM output 0.1 • P0.2 → PWM0.2 : PWM output 0.2 • P0.3 → PWM0.3 : PWM output 0.3 • P0.4 → PWM0.4 : PWM output 0.4 • P0.5 → PWM0.5 : PWM output 0.5 • P0.6 → PWM0.6 : PWM output 0.6 • P0.7 → PWM0.7 : PWM output 0.7	• P0.0 : PWM0.0 • P0.1 : PWM0.1 • P0.2 : PWM0.2 • P0.3 : PWM0.3 • P0.4 : PWM0.4 • P0.5 : PWM0.5 • P0.6 : PWM0.6 • P0.7 : PWM0.7
P1[7:0]	Input/Output	◆ An 8-bit open-Drain or push-pull I/O port. ◆ Note that the output is fully driven (push-pull) when P1[3:0] drives PWM1[3:0] output. • 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 → PWM0.0_A : Alternative PWM output 0.0_A • P1.5 → PWM0.1_A : Alternative PWM output 0.1_A • P1.6 → PWM0.2_A : Alternative PWM output 0.2_A • P1.7 → PWM0.7_A : Alternative PWM output 0.7_A	• P1.0 : PWM1.0 • P1.1 : PWM1.1 • P1.2 : PWM1.2 • P1.3 : PWM1.3 • P1.4 : PWM0.0_A • P1.5 : PWM0.1_A • P1.6 : PWM0.2_A • P1.7 : PWM0.7_A

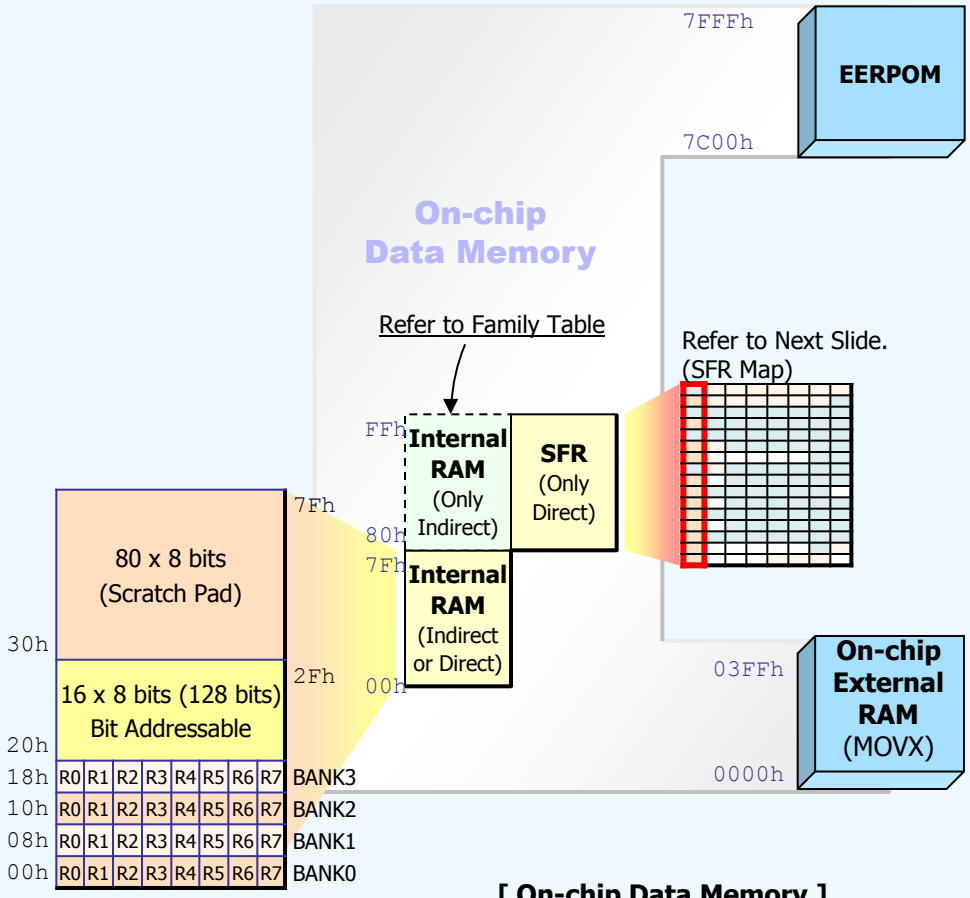
5. Pin Descriptions (Cont'd)

Symbol	Direction	Description	Share Pins
P2[7:0]	Input/Output	◆ An 8-bit open-Drain or push-pull I/O port. • P2.6 → TXD_A : Alternative UART Serial Port Output • P2.7 → RXD_A : Alternative UART Serial Port Input	• P2.6 : TXD_A • P2.7 : RXD_A

6.1. Memory Organization

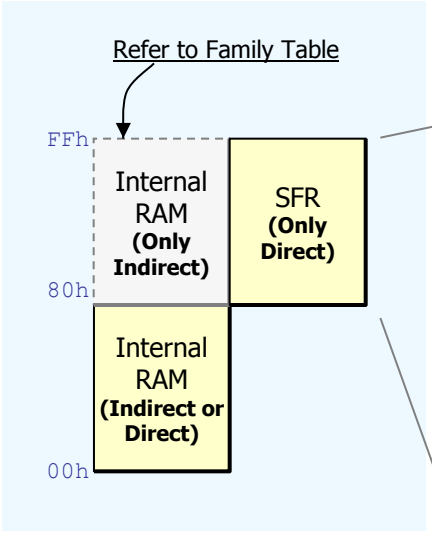


[On-chip Program Memory]
(Read/Write with IAP)



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

6.2. SFR (Special Function Register) Map



Bit addressable

: Reserved for future use.

F8h	I2C1ST	I2C1CON	I2C1CFG	I2C1SLA	I2C1DAT	I2C1SCL	STCON	STCFG	FFh
F0h	B	FCNTLD	FCNTL	FCNTM	FCNTH	SWRST	FCON	FAEN	F7h
E8h	I2C0CON	I2C0RGA	I2C0CFG	I2C0SLA	I2C0TDAT	I2C0TIDX	I2C0RBF0	I2C0RBF1	EFh
E0h	ACC	P0DIR	P1DIR	P2DIR	P3DIR	P2INEN		I2C0ST	E7h
D8h	WDCON	P0TYP	P1TYP	P2TYP	P3TYP	P1INEN			DFh
D0h	PSW	P0SEL	P1SEL	P2SEL	P3SEL	P0INEN	PCLK2EN		D7h
C8h	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2	PCLKEN		CFh
C0h	SPIST				PMR	STATUS	OSCICN	OSC2ICN	C7h
B8h	IP	SADEN	ITSEL						BFh
B0h	P3	EIP	IT		SPICON	SPICK	SPIDR	IPH	B7h
A8h	IE	SADDR		PWM1OEN	P0HD	P1HD	P2HD	P3HD	AFh
A0h	P2	EIE	PWM1CON	PWM1CNT	PWM1D0	PWM1D1	PWM1D2	PWM1D3	A7h
98h	SCON	SBUF	LDOCON	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	LVDCON	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	***11111
IP	Interrupt Priority Low	**000000
IPH	Interrupt Priority High	**000000
IE	Interrupt Enable Control	0*000000
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***0
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 MiDAS400 Family

Register	Name	Reset Value
P0SEL	Port 0 Pull-up Control	11111111
P1SEL	Port 1 Pull-up Control	11111111
P2SEL	Port 2 Pull-up Control	11111111
P3SEL	Port 3 Pull-up Control	***11000
P0HD	Port0 High Current Driving	00000000
P1HD	Port1 High Current Driving	00000000
P2HD	Port2 High Current Driving	00000000
P3HD	Port3 High Current Driving	*****000
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	***00000
P0TYPE	Port 0 Type Control	11111111
P1TYPE	Port 1 Type Control	11111111
P2TYPE	Port 2 Type Control	11111111
P3TYPE	Port 3 Type Control	***11111
IT	Interrupt Type	*****111
ITSEL	Interrupt Selection	***00000
ALTSEL	Alternative Pin Selection	****0010
LDOCON	LDO Control	*1001100
LVDCON	LVD Control	*100*100
SWRST	ISP Configuration	00000000
WDCON	Power Flag and Watchdog Timer Control	01010000
CKCON	Watchdog Timer and 4-cycle Switching Control	111000**
CKSEL	Clock Selection	*****0**
RINGCON	RING Calibration Control	01111011
PMR	Power Management Control	****1***
EXIF	Added External Interrupt and LVD Control	*0000101
EIP	Extended Interrupt Priority	0000*000
EIE	Extended Interrupt Enable	0000*000
STATUS	Crystal Status	***0****
OSCICN	Internal RING Oscillator Control	****0101
OSC2ICN	Internal RING2 Oscillator Control	*****1**

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

6.2. SFR Brief Description

- Newly added SFR Registers in MiDAS400 Family

Register	Name	Reset Value
I2C0ST	I2C0 Status	00000000
I2C0CON	I2C0 Control	*0100000
I2C0CFG	I2C0 Configuration	****0000
I2C0SLA	I2C0 Slave Address	00000000
I2C0RGA	I2C0 Register Address	00000000
I2C0TDAT	I2C0 TX Data	00000000
I2C0TIDX	I2C0 TX Data Index	****0000
I2C0RBF0	I2C0 RX Data Buffer 0	00000000
I2C0RBF1	I2C0 RX Data Buffer 1	00000000
I2C1ST	I2C1 Status	00000000
I2C1CON	I2C1 Control	*0100000
I2C1CFG	I2C1 Configuration	*****000
I2C1SLA	I2C1 Slave Address	00000000
I2C1DAT	I2C1 Data	00000000
I2C1SCL	I2C1 Clock Scaling	00000000
PWM0CON	PWM0 Control	*000**00
PWM0CNT	PWM0 Count	00000000
PWM0OEN	PWM0 Output Enable	00000000
PWM0D0	PWM0D0 Duty Data	00000000
PWM0D1	PWM0D1 Duty Data	00000000
PWM0D2	PWM0D2 Duty Data	00000000
PWM0D3	PWM0D3 Duty Data	00000000
PWM0D4	PWM0D4 Duty Data	00000000
PWM0D5	PWM0D5 Duty Data	00000000
PWM0D6	PWM0D6 Duty Data	00000000
PWM0D7	PWM0D7 Duty Data	00000000
PWM1CON	PWM1 Control	*000**00
PWM1CNT	PWM1 Count	00000000
PWM1OEN	PWM1 Output Enable	****0000
PWM1D0	PWM1D0 Duty Data	00000000
PWM1D1	PWM1D1 Duty Data	00000000
PWM1D2	PWM1D2 Duty Data	00000000
PWM1D3	PWM1D3 Duty Data	00000000

- Newly added SFR Registers in MiDAS400 Family

Register	Name	Reset Value
SPIST	SPI Status	****0000
SPICON	SPI Control	*0000000
SPICK	SPI SCLK Scaling	*****000
SPIDR	SPI Data	00000000
PCLKEN	Peripheral Clock Enable	11111111
PCLK2EN	Peripheral 2 Clock Enable	**11111*
STCON	Stop Timer Control	******00
STCFG	Stop Timer Configuration	***01111
P0INEN	P0 Port Input Enable	11111111
P1INEN	P1 Port Input Enable	11111111
P2INEN	P2 Port Input Enable	11111111
FCNTLD	FLASH Program/Erase Count Load	0*******
FCNTL	FLASH Program/Erase Count Low	00000000
FCNTM	FLASH Program/Erase Count Middle	00000000
FCNTH	FLASH Program/Erase Count High	00000000
FCON	FLASH Control	*0*00000
FAEN	FLASH Access Enable	******00

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

6.3. Instruction Set Summary

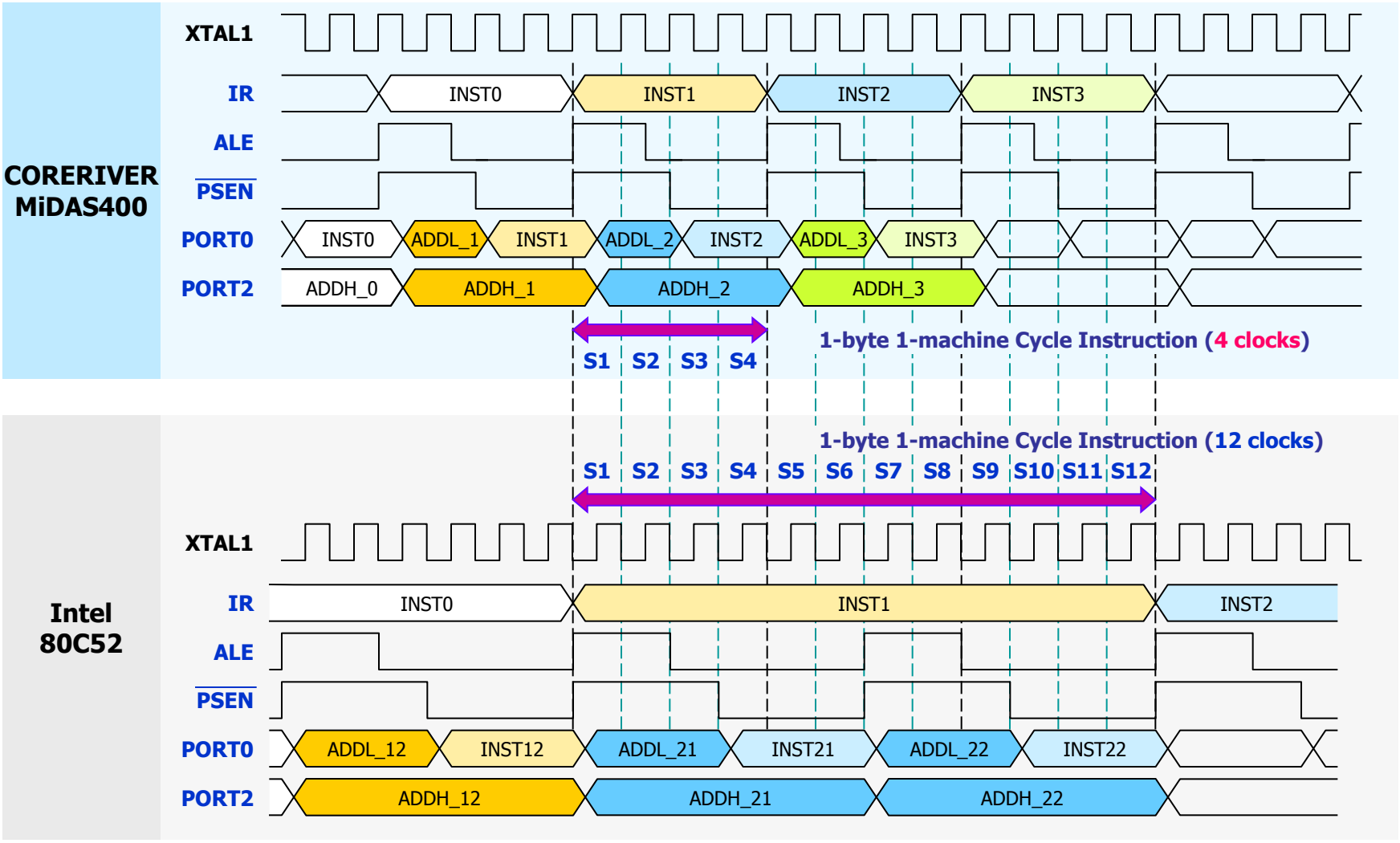
- Refer to Appendix A (Instruction Set) for more details.

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 MiDAS400 family and Intel 80C52



6.4. CPU Timing : Comparison Table

- The Fastest CPU timing in the world

Instruction	**MiDAS400 (CORERIVER)**	W77C32 (Winbond)	DS80C320 (Maxim)	87C52 (Intel)
`MUL AB` `DIV AB`	**12 clocks**	20 clocks	20 clocks	48 clocks
`MOVC A, @A+PC` `MOVC A, @A+DPTR`	**8 clocks**	8 clocks	12 clocks	24 clocks
`JMP @A+DPTR`	**8 clocks**	8 clocks	12 clocks	24 clocks
`RET` `RETI`	**8 clocks**	8 clocks	16 clocks	24 clocks
`INC DPTR`	**4 clocks**	8 clocks	12 clocks	24 clocks
`Others`	**Same**	Same	Same	-

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

- Open-drain or push-pull output.
- The alternative functions are available only when the corresponding SFR bit is "1".
  - ✓ P0.0 = PWM0.0 / P0.1 = PWM0.1 / P0.2 = PWM0.2 / P0.3 = PWM0.3 / P0.4 = PWM0.4 / P0.5 = PWM0.5 / P0.6 = PWM0.6 / P0.7 = PWM0.7
- 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

✓ **P0TYPE** (D9h) : Port 0 Type Control 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)

✓ **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)

- 0 = Output (Default) / 1 = Input

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

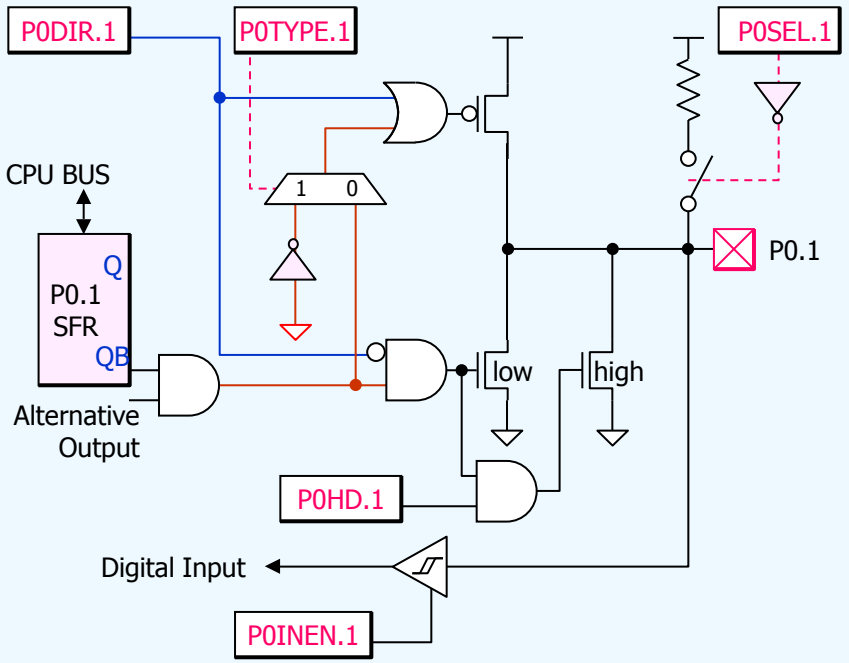
P0SEL.7	P0SEL.6	P0SEL.5	P0SEL.4	P0SEL.3	P0SEL.2	P0SEL.1	P0SEL.0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

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

✓ **P0HD** (ACh) : Port 0 High Current Driving Control Register

P0HD.7	P0HD.6	P0HD.5	P0HD.4	P0HD.3	P0HD.2	P0HD.1	P0HD.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 = High Current Driving OFF (Default) / 1 = ON



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

✓ **P0INEN** (D5h) : Port 0 Input Enable Register

P0INEN.7	P0INEN.6	P0INEN.5	P0INEN.4	P0INEN.3	P0INEN.2	P0INEN.1	P0INEN.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 = Input Disable / 1 = Input Enable (Default)

✓ **P0** (80h) : Port 0 Register

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

6.5. I/O Ports : PORT1[7:0]

- Open-drain or push-pull output.
- The alternative functions are available only when the corresponding SFR bit is "1".
  - ✓ P1.0 = PWM1.0 / P1.1 = PMW1.1 / P1.2 = PWM1.2 / P1.3 = PWM1.3 / P1.4 = PWM0.0_A / P1.5 = PMW0.1_A / P1.6 = PWM0.2_A / P1.7 = PWM0.7_A
- 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** (DAh) : Port 1 Type Control 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)

✓ **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)

- 0 = Output (Default) / 1 = Input

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

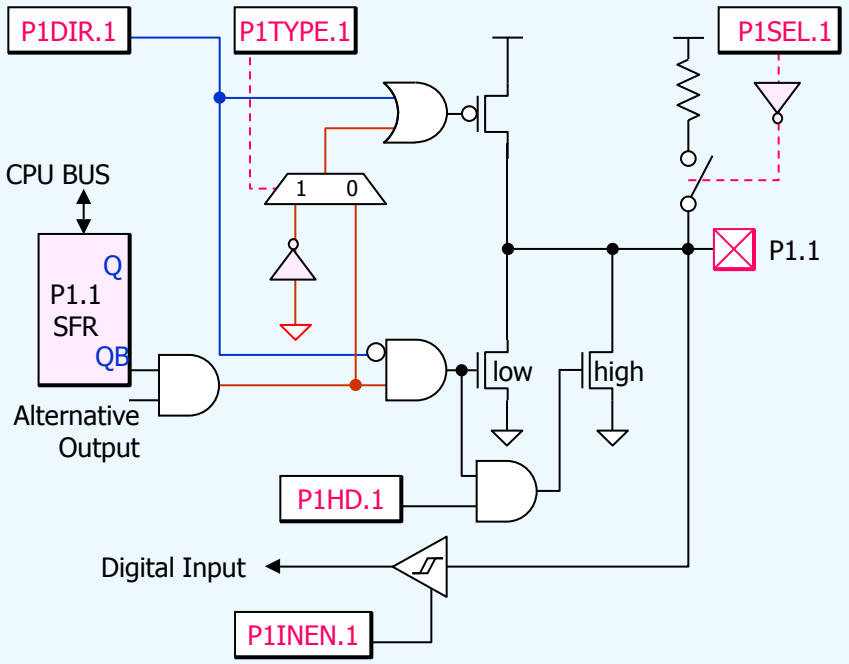
P1SEL.7	P1SEL.6	P1SEL.5	P1SEL.4	P1SEL.3	P1SEL.2	P1SEL.1	P1SEL.0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

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

✓ **P1HD** (ADh) : Port 1 High Current Driving Control Register

P1HD.7	P1HD.6	P1HD.5	P1HD.4	P1HD.3	P1HD.2	P1HD.1	P1HD.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 = High Current Driving OFF (Default) / 1 = ON



6.5. I/O Ports : PORT1[7:0]

✓ **P1INEN** (DDh) : Port 1 Input Enable Register

P1INEN.7	P1INEN.6	P1INEN.5	P1INEN.4	P1INEN.3	P1INEN.2	P1INEN.1	P1INEN.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 = Input Disable / 1 = Input Enable (Default)

✓ **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)

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

- Open-drain or push-pull output.
- The alternative functions are available only when the corresponding SFR bit is "1".
  - ✓ P2.6 = TXD_A / P2.7 = RXD_A
- 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** (DBh) : 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

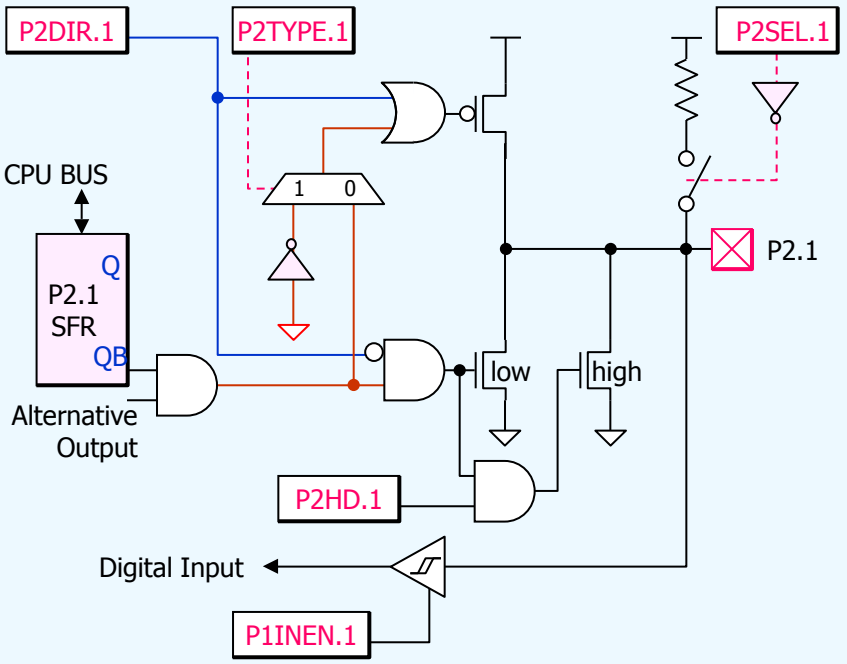
P2SEL.7	P2SEL.6	P2SEL.5	P2SEL.4	P2SEL.3	P2SEL.2	P2SEL.1	P2SEL.0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

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

✓ **P2HD** (AEh) : Port 2 High Current Driving Control Register

P2HD.7	P2HD.6	P2HD.5	P2HD.4	P2HD.3	P2HD.2	P2HD.1	P2HD.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 = High Current Driving OFF (Default) / 1 = ON



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

✓ **P2INEN** (E5h) : Port 2 Input Enable Register

P2INEN.7	P2INEN.6	P2INEN.5	P2INEN.4	P2INEN.3	P2INEN.2	P2INEN.1	P2INEN.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 = Input Disable / 1 = Input Enable (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[4:0]

- Open-drain or push-pull output, pull-up control (Only P3[1:0]).
- The alternative functions are available only when the corresponding SFR bit is "1".
  - ✓ P3.0 = SSB, I2C1_SDA, INT0, RXD / P3.1 = SCLK, I2C1_SCL, INT1, TXD / P3.2 = RESETB, MISO, INT2 / P3.3 = MOSI, I2C0_SDA, INT3, XTAL2 / P3.4 = I2C0_SCL, INT4, XTAL1
- 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** (DCh) : Port 3 Type Control Register

-	-	-	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)

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

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

-	-	-	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)

- 0 = Output (Default) / 1 = Input

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

-	-	-	P3SEL.4	P3SEL.3	P3SEL.2	P3SEL.1	P3SEL.0
			R/W(1)	R/W(1)	R/W(0)	R/W(0)	R/W(0)

- 0 = Pull-up Resistor ON (P3[[2:0] for Default)
- 1 = Pull-up Resistor OFF (P3[4:3] for Default))

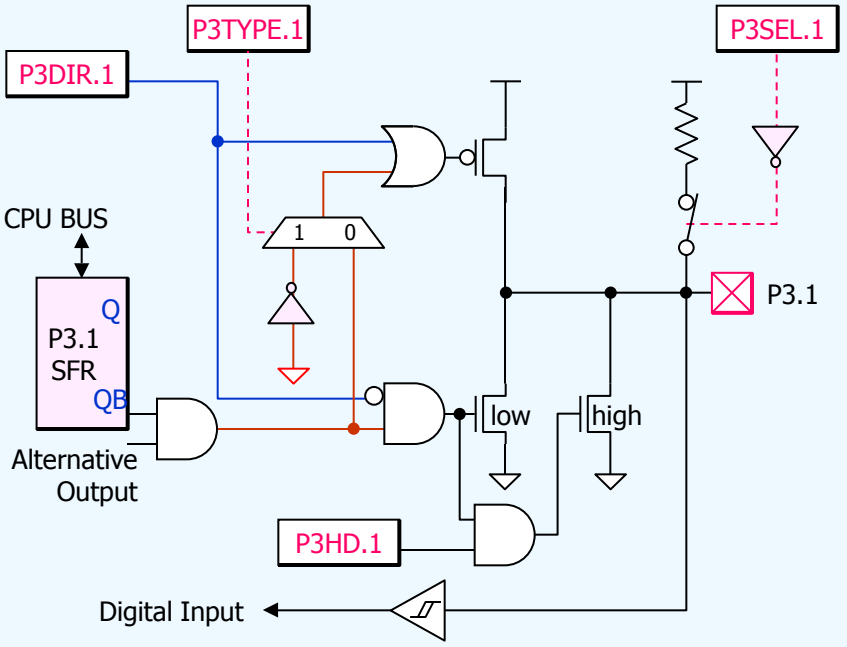
✓ **P3HD** (AFh) : Port 3 High Current Driving Control Register

-	-	-	-	-	P3HD.2	P3HD.1	P3HD.0
					R/W(0)	R/W(0)	R/W(0)

- 0 = High Current Driving OFF (Default) / 1 = ON

✓ **P3** (B0h) : Port 3 Register

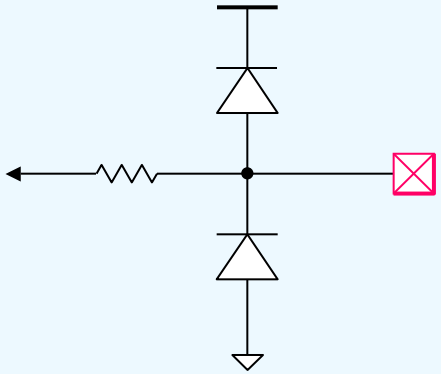
-	-	-	P3.4	P3.3	P3.2	P3.1	P3.0
			R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)



6.5. The ESD Structure of Pads

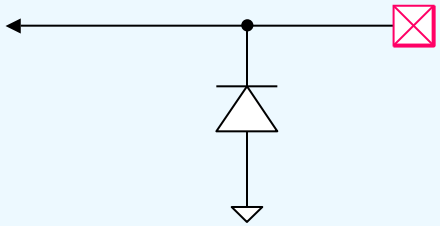
- Two ESD diodes and one ESD resister are contained in all pads except VDD.
- One ESD diode are contained in VDD.

[All pads except VDD]



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

[VDD]



- One ESD Diode (GND side)

6.6. LVD (Low Voltage Detector)

- On-chip power-on reset : 2.5V
- On-chip power-fail reset : 2.5V
- Flag Transition

	POF	POR
A	X → 1	X → 1
D	X → 1	X → 1

- POF is a mirror of POR.

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

-	IE4	IE3	IE2	XT/RG	RGMD	RGSL	BGS
	R/W(0)	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

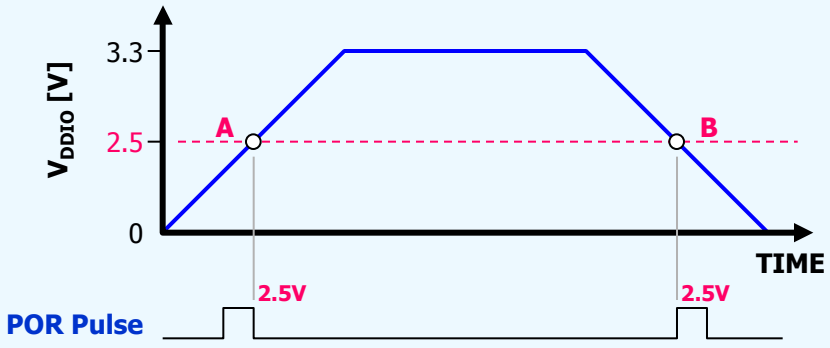
SMOD1	SMOD0	-	POF	GF1	GF0	PD	IDL
R/W(0)	R/W(0)		R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

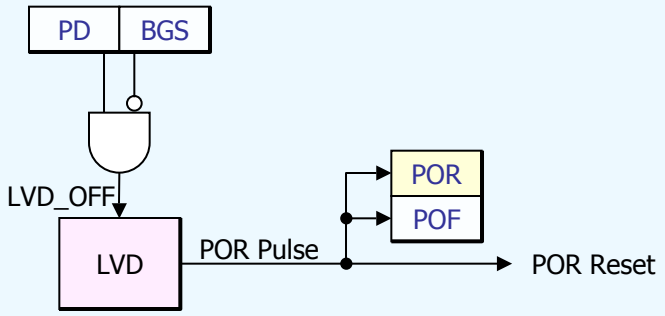
- POF : Power-off Flag
- PD : Power-down mode bit

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

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

- POR : Power-on Reset Flag





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

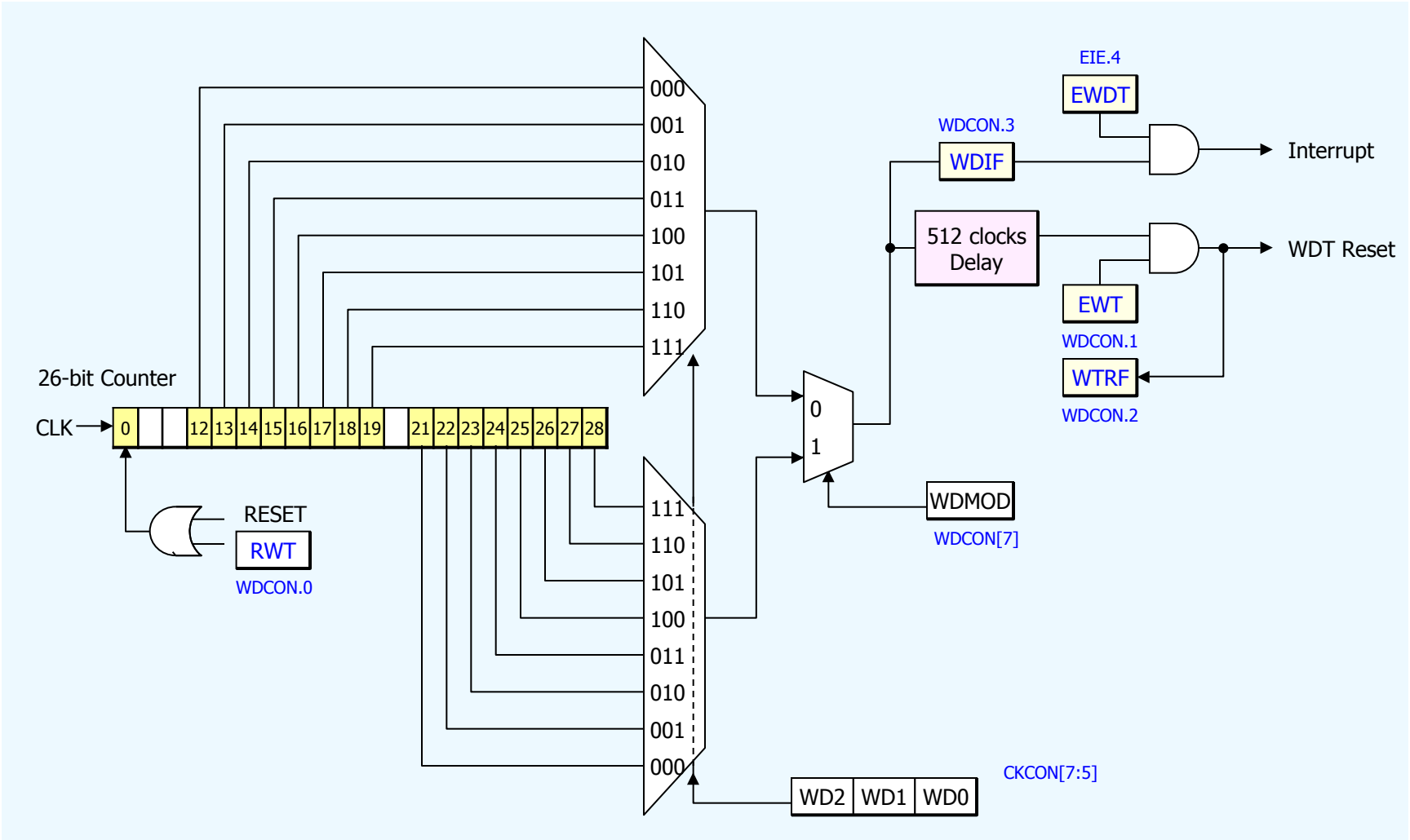
- WDMOD : WDT mode selection Flag
- POR : Power-on Reset Flag
- WDIF : Watchdog Timer Interrupt Flag
- WTRF : Watchdog Timer Reset Flag. Only cleared by S/W.
- EWT : Watchdog Timer Reset Enable
- RWT : Restart Watchdog Timer

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

WDMOD=0			Interrupt Time-out (@48MHz)		Reset Time-out (@48MHz)	WDMOD=1			Interrupt Time-out (@48MHz)		Reset Time-out (@48MHz)
WD2	WD1	WD0				WD2	WD1	WD0			
0	0	0	2^{12} clocks	about 85 us	2^{12} + 512 clocks	0	0	0	2^{21} clocks	43.69 ms	2^{21} + 512 clocks
0	0	1	2^{13} clocks	170 us	2^{13} + 512 clocks	0	0	1	2^{22} clocks	87.38 ms	2^{22} + 512 clocks
0	1	0	2^{14} clocks	341 us	2^{14} + 512 clocks	0	1	0	2^{23} clocks	174.76 ms	2^{23} + 512 clocks
0	1	1	2^{15} clocks	682 us	2^{15} + 512 clocks	0	1	1	2^{24} clocks	349.52 ms	2^{24} + 512 clocks
1	0	0	2^{16} clocks	1.365 ms	2^{16} + 512 clocks	1	0	0	2^{25} clocks	699.05 ms	2^{25} + 512 clocks
1	0	1	2^{17} clocks	2.73 ms	2^{17} + 512 clocks	1	0	1	2^{26} clocks	1.398 s	2^{26} + 512 clocks
1	1	0	2^{18} clocks	5.46 ms	2^{18} + 512 clocks	1	1	0	2^{27} clocks	2.796 s	2^{27} + 512 clocks
1	1	1	2^{19} clocks	10.92 ms	2^{19} + 512 clocks	1	1	1	2^{28} clocks	5.592 s	2^{28} + 512 clocks

6.7. WDT (Watchdog Timer)

- Block Diagram



6.8. ST (Stop Timer)

- Timer to wake up in stop mode
- Use the internal 32KHz oscillator for saving power consumption

✓ **STCON** (FEh) : Stop Timer Control Register

-	-	-	-	-	-	ST_CLR	ST_EN
						R/W(0)	R/W(0)

- ST_CLR : Clear Stop Timer Count & Stop Timer disable
- ST_EN : Stop Timer enable

✓ **STCFG** (FFh) : Stop Timer Configuration Register

-	-	-	ST_BUSY	ST_DR3	ST_DR2	ST_DR1	ST_DR0
			R(0)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

- ST_BUSY : Stop Timer Busy flag
- ST_DR[3:0] : Stop Timer Time-out Values

Stop Timer Time-out Values

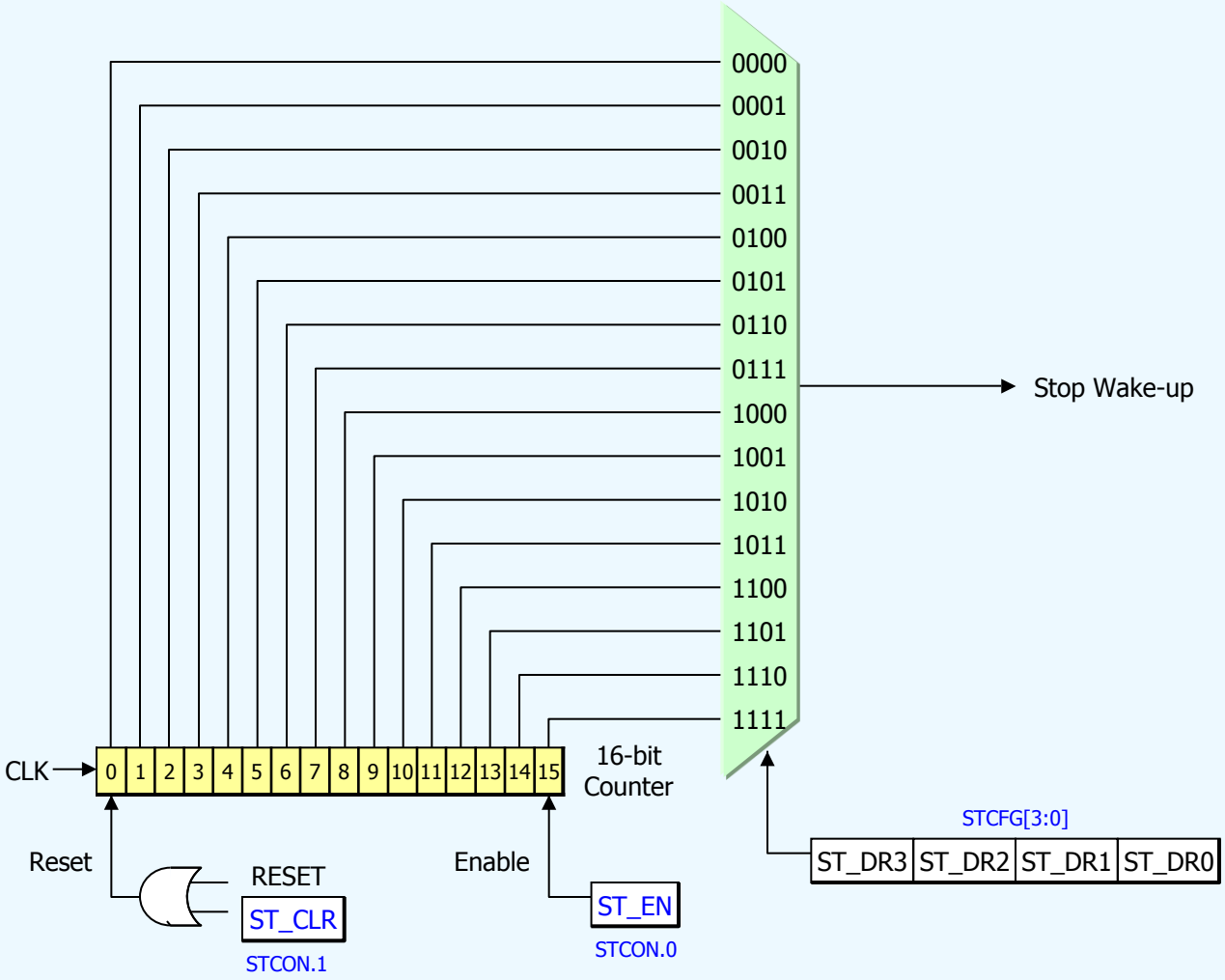
ST_DR3	ST_DR2	ST_DR1	ST_DR0	Interrupt Time-out (@32KHz)
0	0	0	0	2^0 clocks
0	0	0	1	2^1 clocks
0	0	1	0	2^2 clocks
0	0	1	1	2^3 clocks
0	1	0	0	2^4 clocks
0	1	0	1	2^5 clocks
0	1	1	0	2^6 clocks
0	1	1	1	2^7 clocks
1	0	0	0	2^8 clocks
1	0	0	1	2^9 clocks
1	0	1	0	2^{10} clocks
1	0	1	1	2^{11} clocks
1	1	0	0	2^{12} clocks
1	1	0	1	2^{13} clocks
1	1	1	0	2^{14} clocks
1	1	1	1	2^{15} clocks

Example source

```
//
// c - code
//

STCFG = 0x04;   // 2^5 clock time-out (32KHz)
STCON = 0x01;   // run ST

PCON |= 0x02;   // enter the stop mode
_nop_();
_nop_();

STCON = 0x02;   // clear ST counter & stop ST
```

6.8. ST (Stop 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

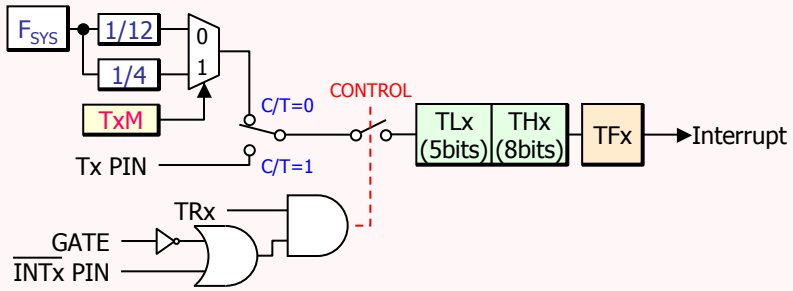
TL1.7	TL1.6	TL1.5	TL1.4	TL1.3	TL1.2	TL1.1	TL1.0

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

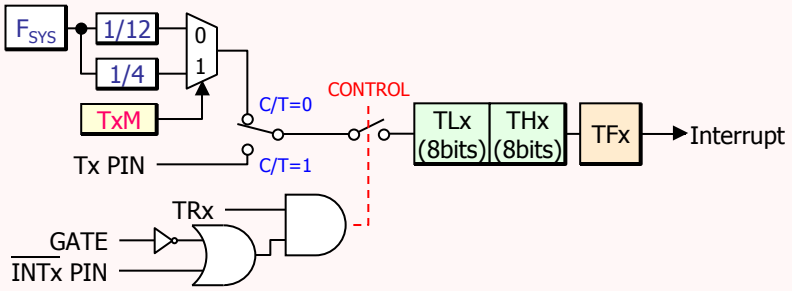
TH1.7	TH1.6	TH1.5	TH1.4	TH1.3	TH1.2	TH1.1	TH1.0

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

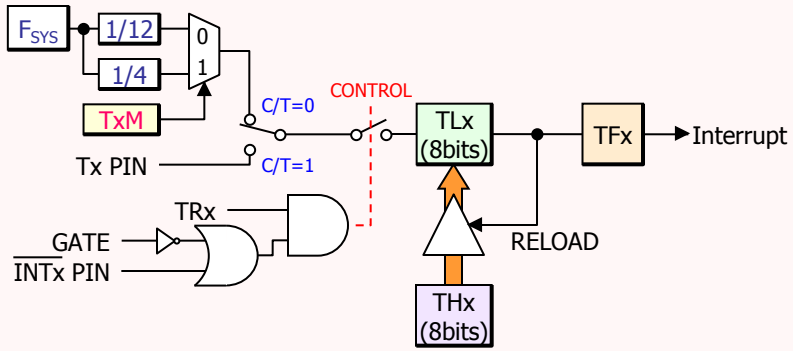
* Default : $F_{SYS}/12$ (T0M and T1m is each 0.)



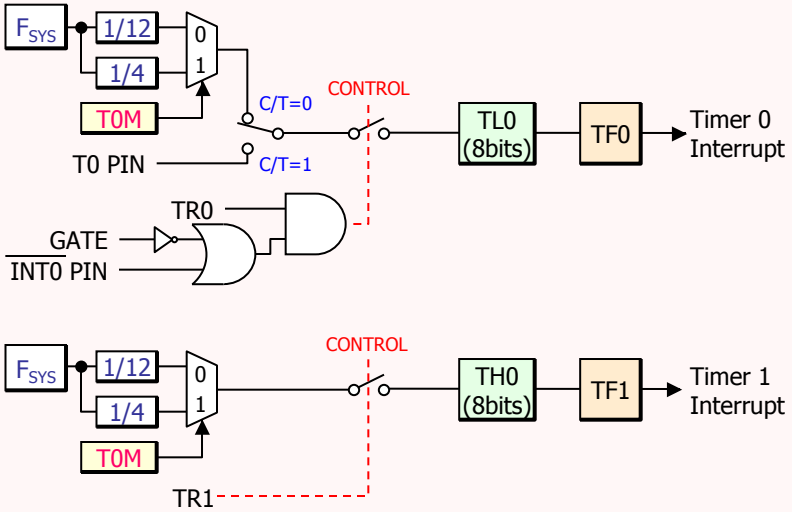
[Mode 0]



[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]	16-bit Timer/Counter With Automatic Reload (TH2, TL2 ← RCAP2H, RCAP2L)
2. 16-bit Capture [RCLK+TCLK=0, CP/RL2=1]	16-bit Timer/Counter with Capture (RCAP2H, RCAP2L ← TH2, TL2)
3. Baud Rate Generator [RCLK+TCLK=1, CP/RL2=X]	Baud Rate Generation * Timer 2 Interrupt Disable

✓ **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 Enable
- 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

-	-	LINBG	LINBD	-	-	-	DCEN
		R/W(0)	R/W(0)				R/W(0)

- LINBG : LIN Baud Rate Generation Enable
- LINBD : LIN Baud Rate Detection Enable, Cleared by H/W.
- 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)

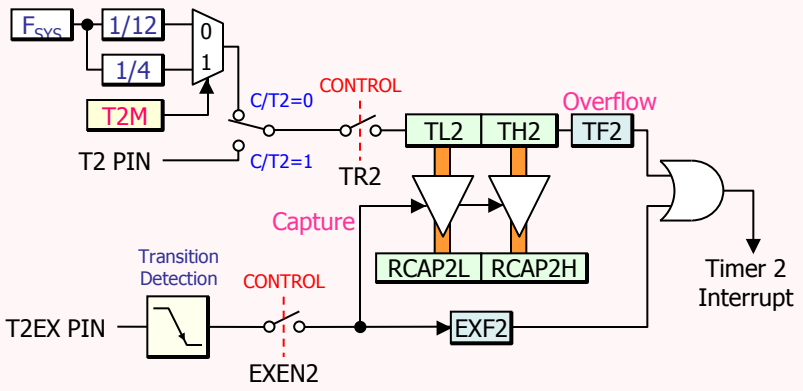
✓ **RCAP2L** (CAh) : Timer 2 Capture/Reload Low Byte Register

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)

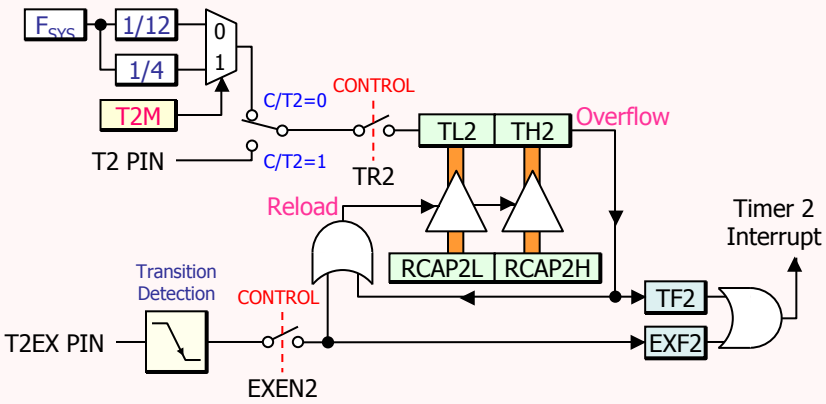
✓ **RCAP2H** (CBh) : Timer 2 Capture/Reload High Byte Register

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)

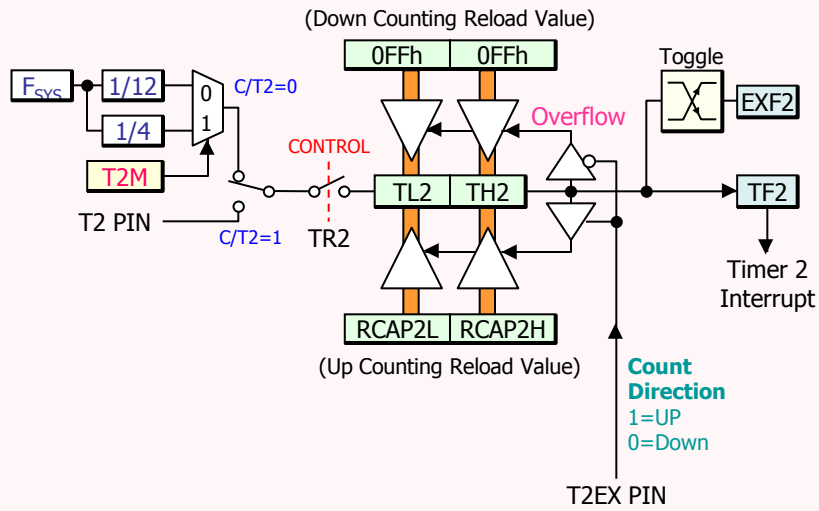
6.9. Timer/Counter : Timer 2 Mode Description



[Capture Mode]

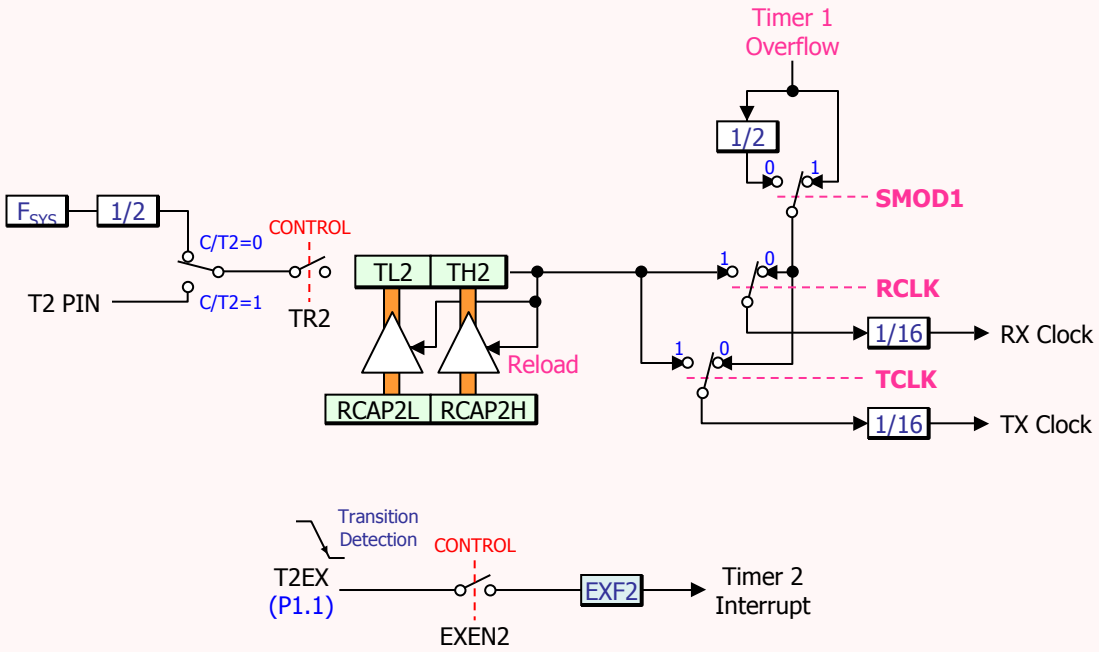


[Auto Reload Mode (DCEN=0)]



[Auto Reload Mode (DCEN=1)]

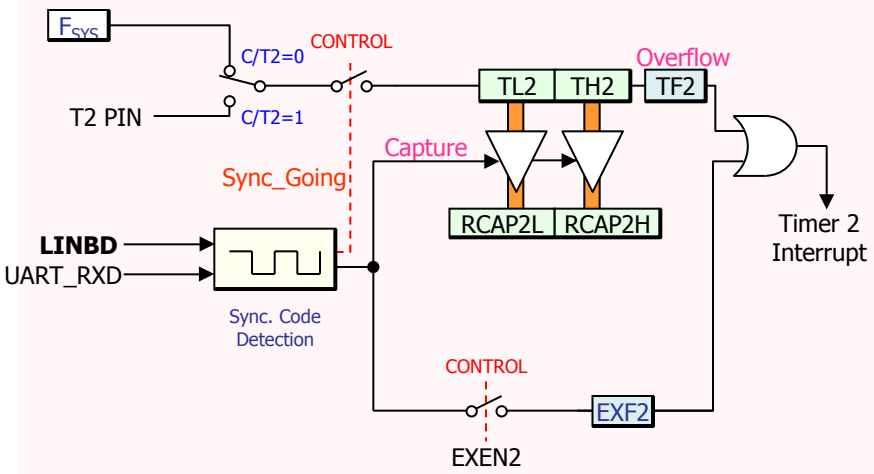
6.9. Timer/Counter : Timer 2 Mode Description



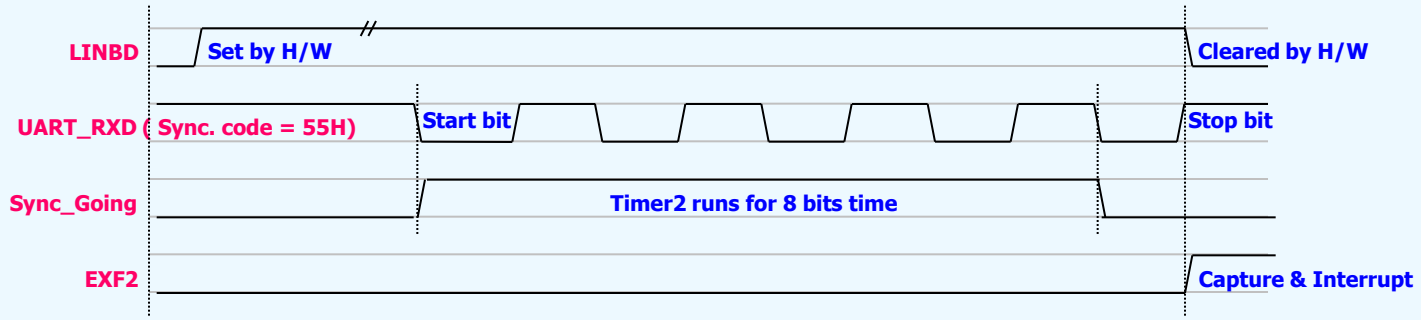
[Baud Rate Generator Mode]

6.9. Timer/Counter : Timer 2 Mode Description

- LIN Baud Rate Detector Mode (LINBD = 1)
  - ✓ Timer 2 counts the first 8 bit times of sync. code (55H) received by UART.
  - ✓ If enabled, EXF2 is set after the detection.
  - ✓ TF2 is set when Timer 2 overflows due to sync. failure.
- Operation Guide
  - ✓ Clear TL2 and TH2.
  - ✓ Set EXEN2 and LINBD. Do not set TR2.
  - ✓ After the sync. is detected (EXF2 interrupt), set TR2, LINBG and RCLK or TCLK for UART transmission with detected baud rate.

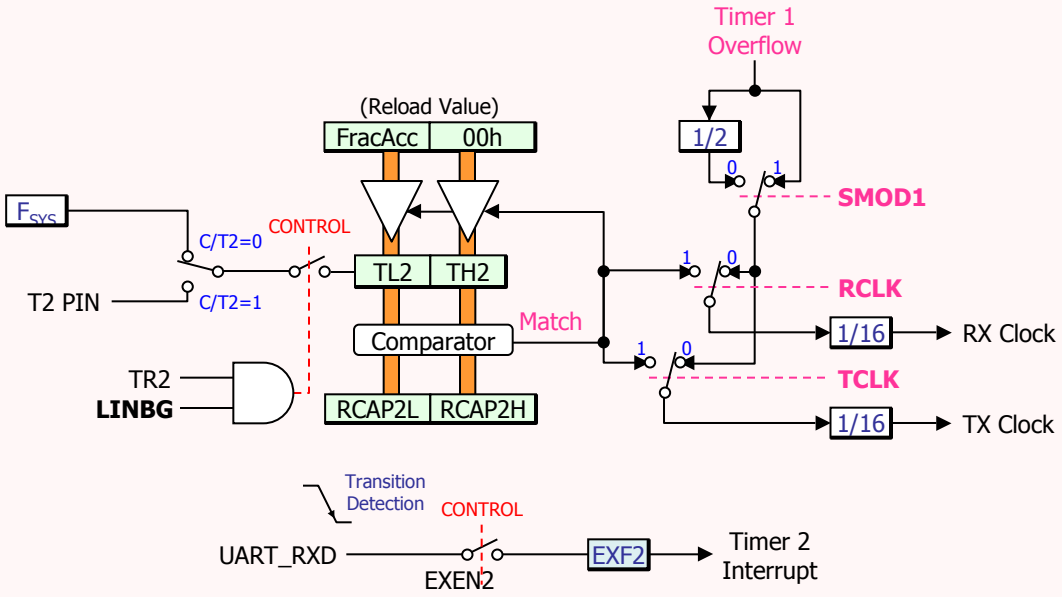


[LIN Baud Rate Detector Mode(LINBD = 1)]



6.9. Timer/Counter : Timer 2 Mode Description

- LIN Baud Rate Generator Mode (LINBG = 1)
  - ✓ Generate clock for UART according to the baud rate detected by LINBD.
  - ✓ In this mode, Timer 2 is divided into 9-bit digital part and 7-bit fraction part.



[LIN Baud Rate Generator Mode (LINBG = 1)]

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 System Clock (F_{SYS})
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 F_{SYS} (SMOD1=0) 1/16 x F_{SYS} (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 (F_{SYS}/4)
  [0,1] : Mode 1. 8-bit UART (Variable)
  [1,0] : Mode 2. 9-bit UART (F_{SYS}/32 or F_{SYS}/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

$$\text{Baudrate} = \frac{\text{System Frequency } (F_{SYS})}{4}$$

- Serial Port Operating Mode 2

$$\text{Baudrate} = \frac{2^{SMOD1}}{32} \times F_{SYS}$$

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

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

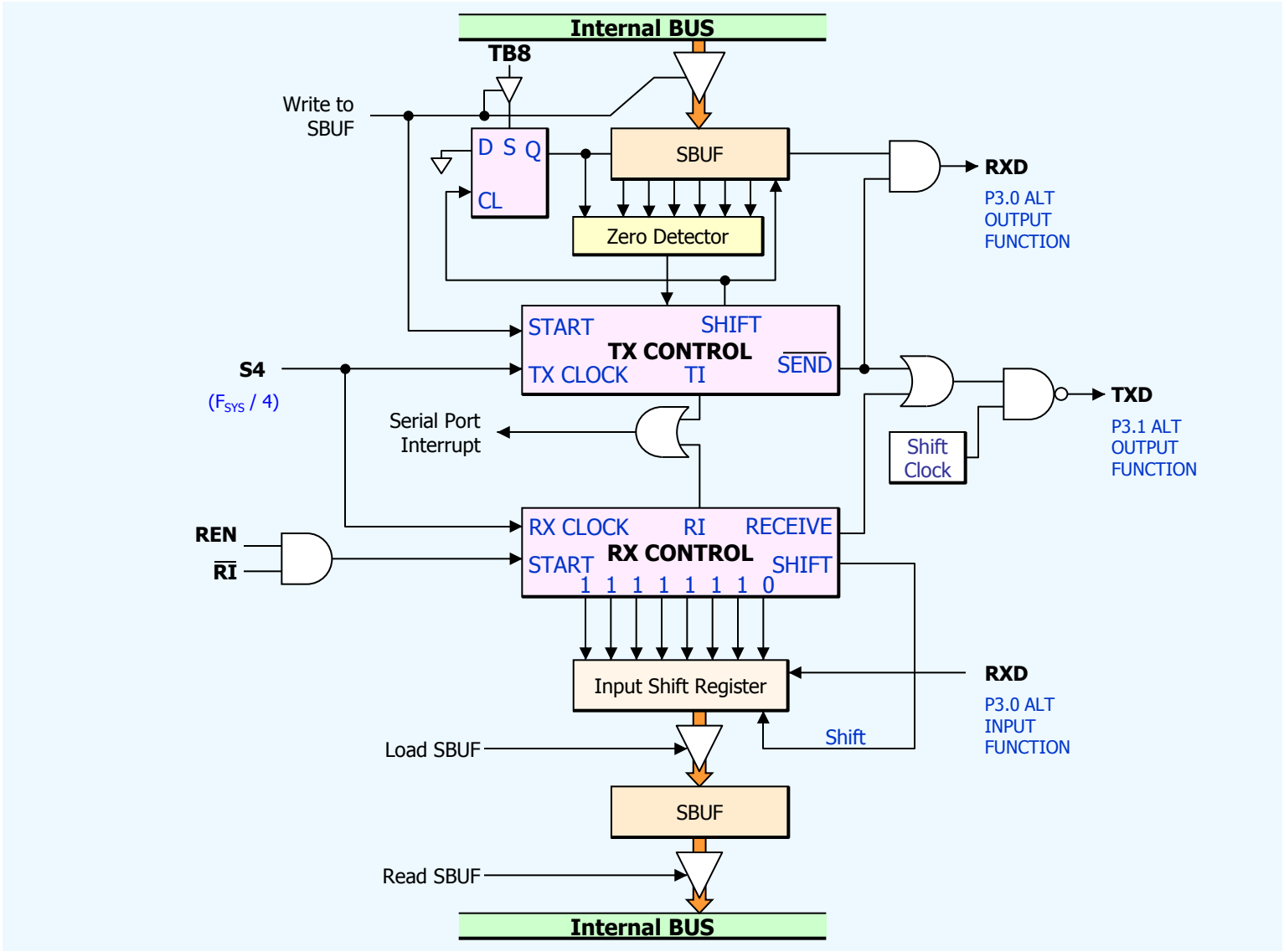
- If SMOD1(PCON.7) = 1 ➔ Double Baudrate
- If T1M(CKCON.4) = 0 ➔ F_{SYS} / 12
- If T1M(CKCON.4) = 1 ➔ F_{SYS} / 4

EX) Using Timer 2 to Generate Baudrates

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

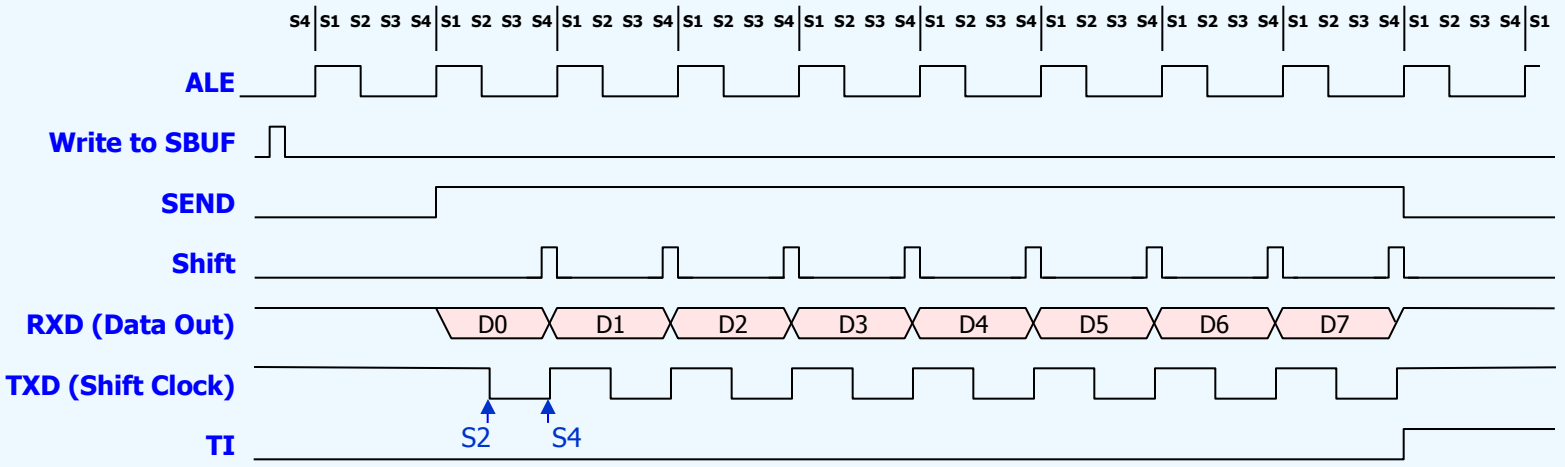
Baudrate		UART Mode	F_{SYS}	SMOD1	Timer 1		
T1M=0	T1M=1				C/T	Mode	Reload Value (TH1)
Max : 3 MHz	Max : 3 MHz	Mode 0	12 MHz	X	X	X	X
Max : 750 KHz	Max : 750 KHz	Mode 2	12 MHz	1	X	X	X
62.5 KHz	187.5 KHz	Mode 1 & 3	12 MHz	1	0	2	FFh
19.2 KHz	57.6 KHz		11.0592 MHz	1	0	2	FDh
9.6 KHz	28.8 KHz		11.0592 MHz	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

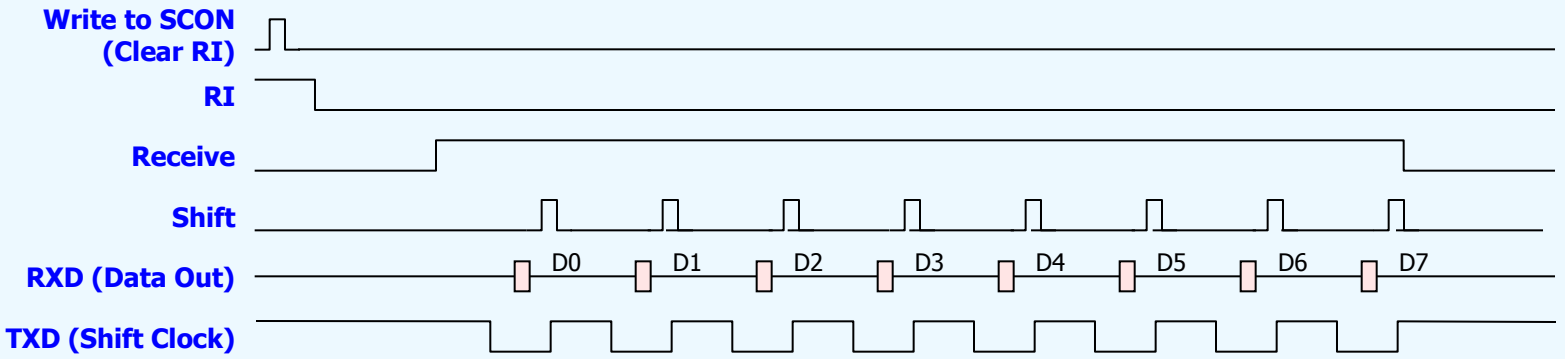


6.10. UART : Mode 0, Timing Diagram

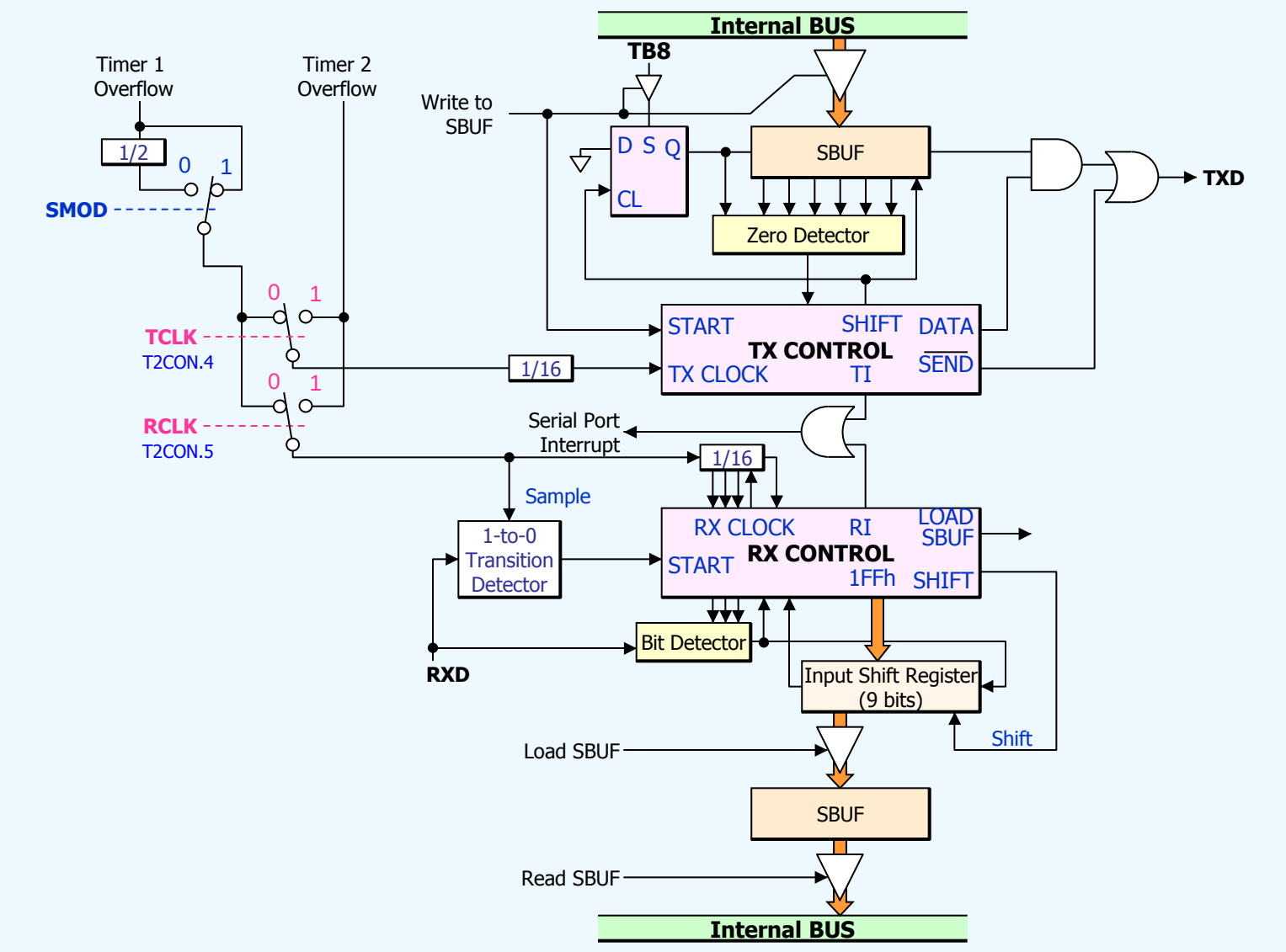
[Transmit]



[Receive]

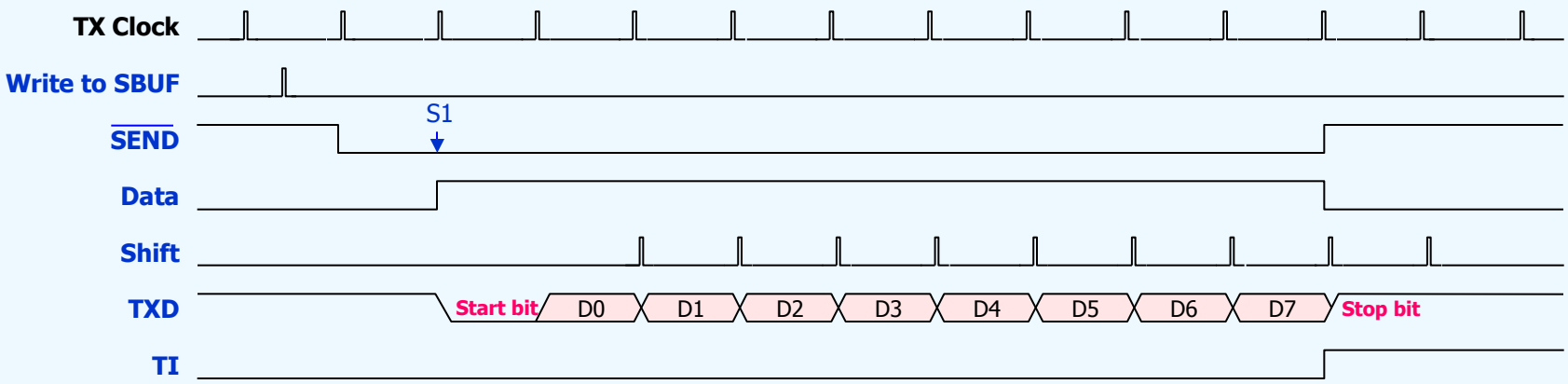


6.10. UART : Mode 1, Functional Diagram

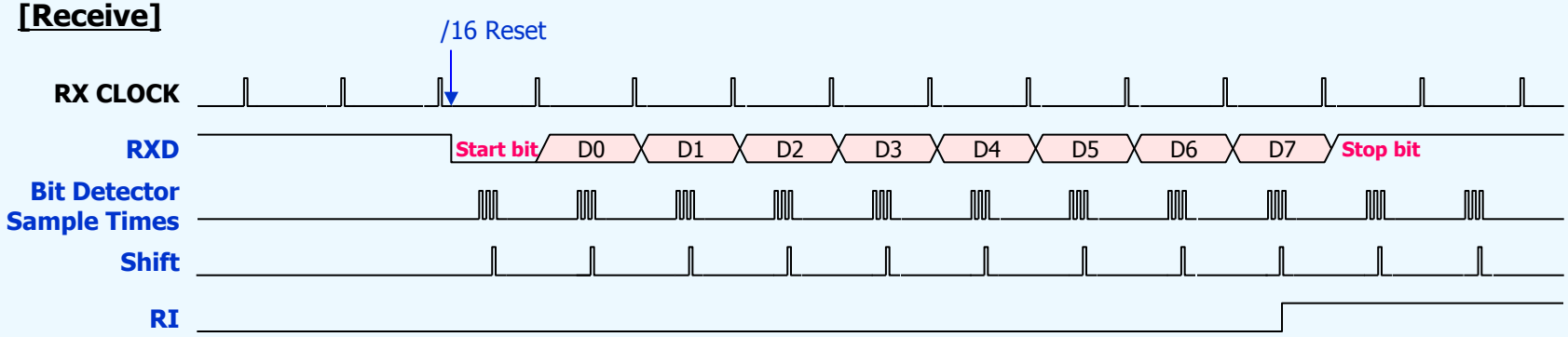


6.10. UART : Mode 1, Timing Diagram

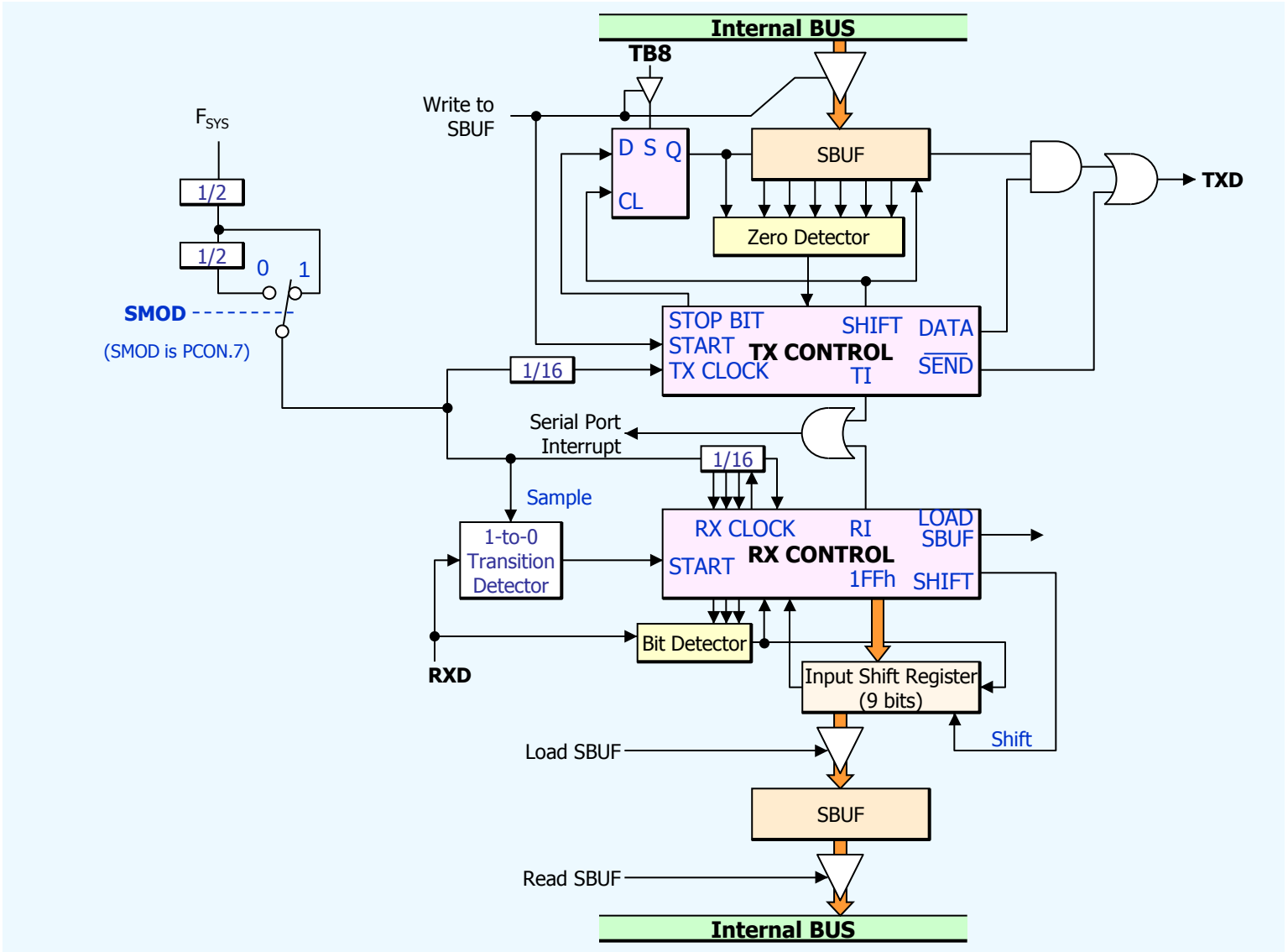
[Transmit]



[Receive]

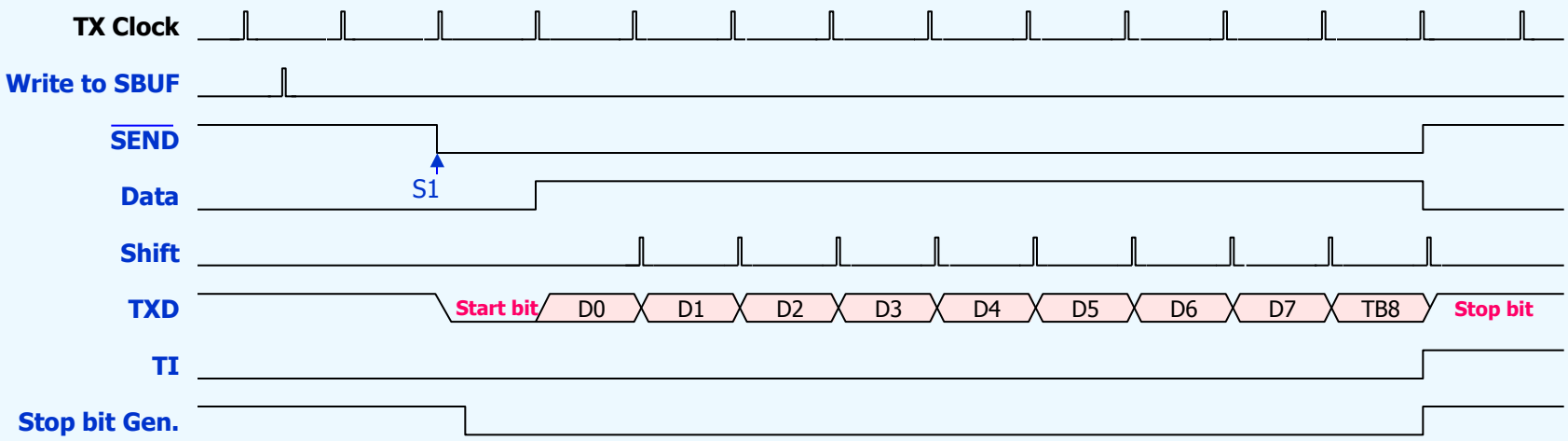


6.10. UART : Mode 2, Functional Diagram

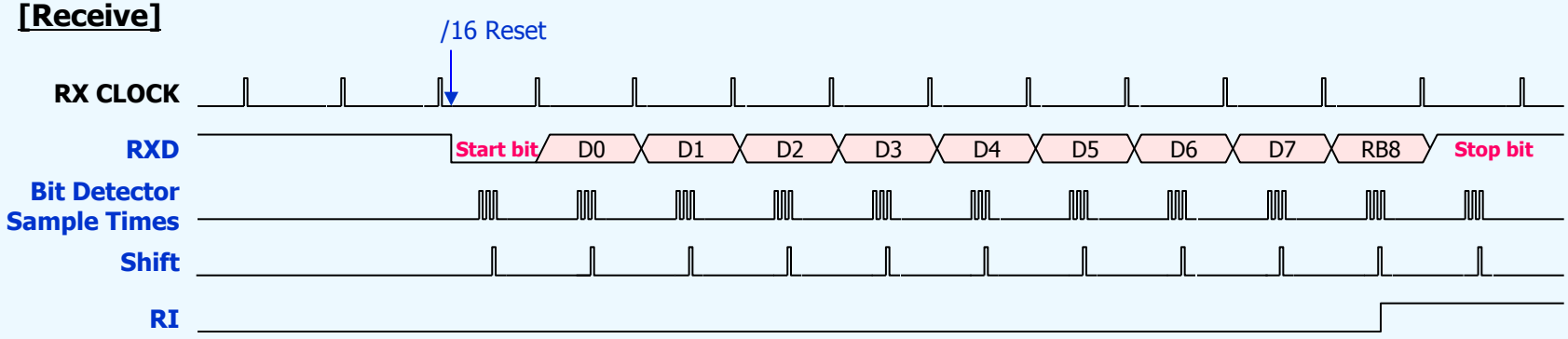


6.10. UART : Mode 2, Timing Diagram

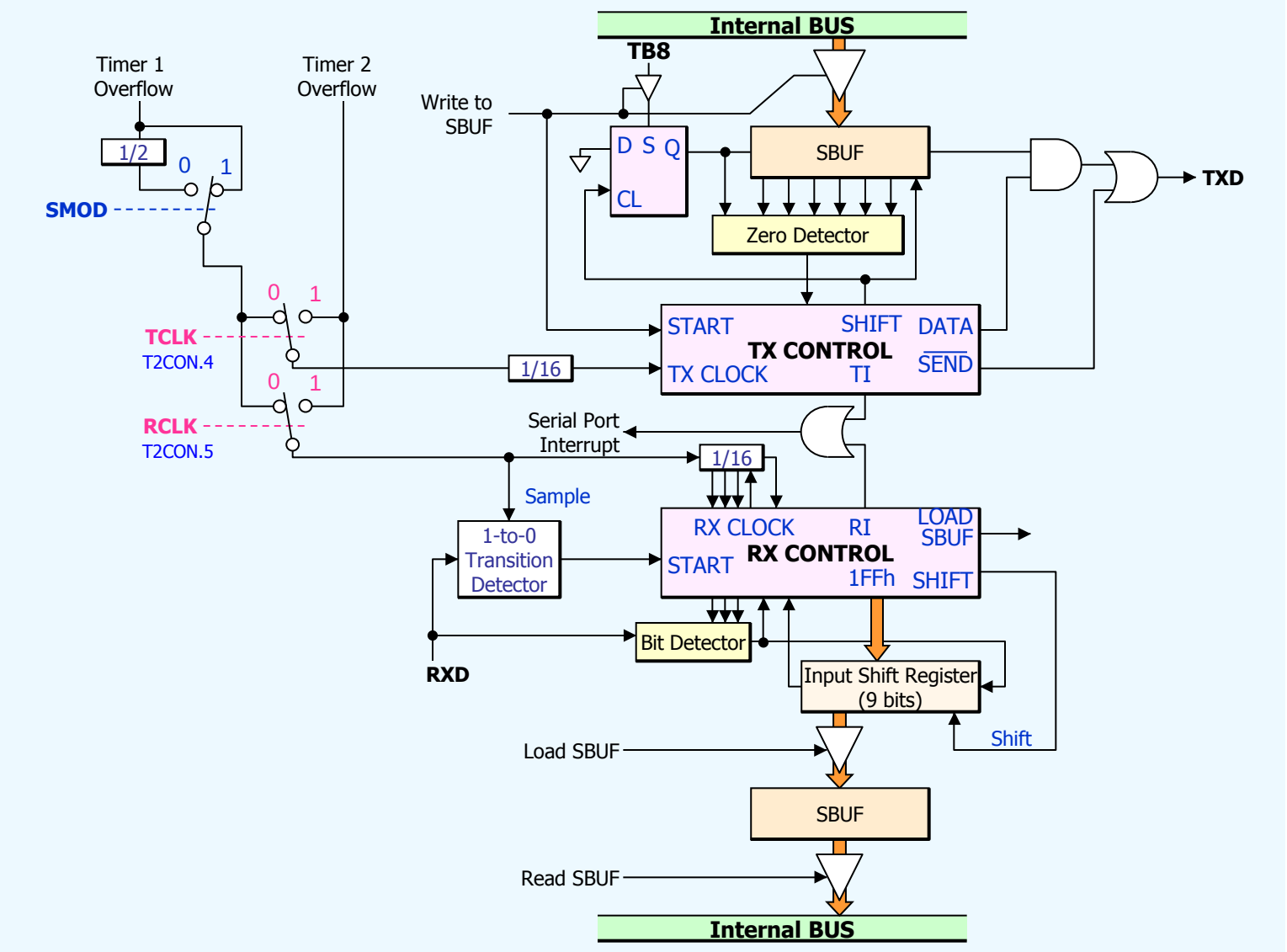
[Transmit]



[Receive]

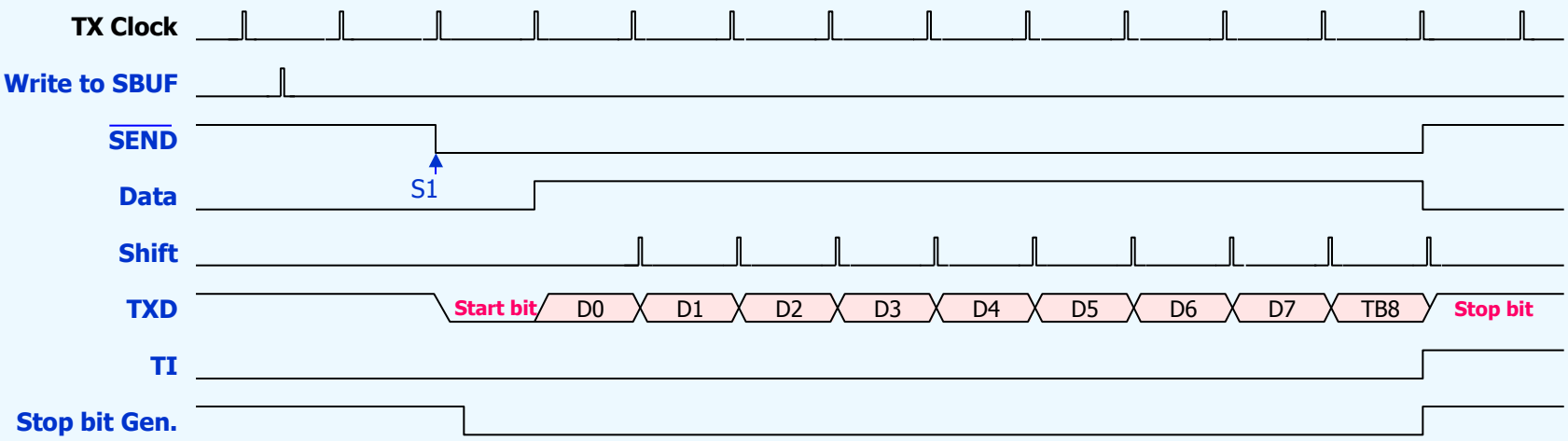


6.10. UART : Mode 3, Functional Diagram

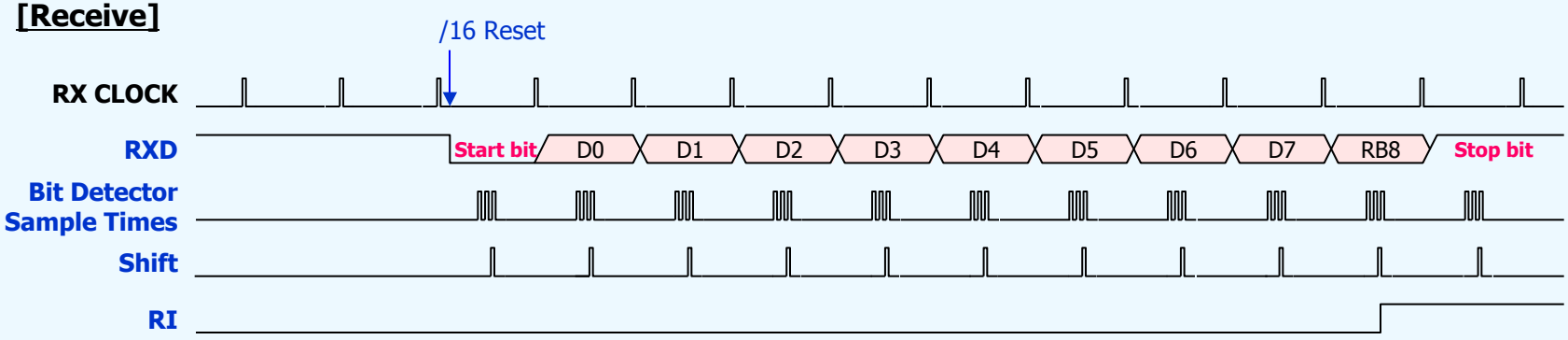


6.10. UART : Mode 3, Timing Diagram

[Transmit]



[Receive]



6.11. I2C0 : SFR

- Two-wire Interface
- Slave Operation
- Transmitter or Receiver Operation
- 100Kbps (Min. F_{SYS} = 1MHz), 400Kbps (Min. F_{SYS} = 4MHz)
- 7bits / 10bits (Extended 15bits) Address Mode
- Fully Programmable Slave Address
- Fully Programmable Register Address (10bit mode)
- SDA/SCL Schmitt-trigger input
- Support self-running mode without CPU
- 9bytes TX buffers
- 2bytes RX buffers
- Compatible with Phillips I2C protocol

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

ESPI	EI2C1	EI2C0	EWDT	-	EX4	EX3	EX2
R/W(0)	R/W(0)	R/W(0)	R/W(0)		R/W(0)	R/W(0)	R/W(0)

- EI2C1 : I2C1 Interrupt Enable
- EI2C0 : IC20 Interrupt Enable

✓ **I2C0SLA** (EBh) : I²C0 Slave Address Register

SLA1.7	SLA1.6	SLA1.5	SLA1.4	SLA1.3	SLA1.2	SLA1.1	-
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[7:1] : I²C Slave Address Register.

✓ **I2C0CFG** (EAh) : I²C0 Configuration Register

TXIEN	RXIEN	ADSEL	ADMIE	CKCFG	I2CDIV2	I2CDIV1	I2CDIV0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- I2CDIV : I2C0 clock divisor
  [000] : F_{SYS} / 2 [001] : F_{SYS} / 4
  [010] : F_{SYS} / 8 [011] : F_{SYS} / 16
  [100] : F_{SYS} / 32 [101] : F_{SYS} / 64
  [110] : F_{SYS} / 128 [111] : F_{SYS} / 256
- CKCFG : I2C Clock configuration
  [0] : In the wait state I2C0 clock is set as I2CDIV value
  [1] : In the wait state I2C0 clock is set as Fsys/256
- ADSEL : 7-bit / 10-bit Address Mode Selection in Slave mode
  [0] : 7-bit mode [1] : 10-bit mode
- ADMIE : SLA matching interrupt enable
  [0] : When SLA matched, interrupt occurred
  [1] : Regardless of SLA matching, interrupt occurred
- RXIEN : I2C Data RX Interrupt Enable
  [0] : disable [1] : enable
- TXIEN : I2C Data TX Interrupt Enable
  [0] : disable [1] : enable

✓ **I2C0RGA** (E9h) : I²C0 Slave Register

RGA1.7	RGA1.6	RGA1.5	RGA1.4	-	-	-	-
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R(0)	R(0)	R(0)	R(0)

- RGA[7:4] : I²C Register Address Register.
  The upper 4bit Register Address is fixed as RGA[7:4], the lower 4bit Register Address indicates RX/TX buffer address in the 10bit mode.

6.11. I2C0 : SFR (Cont'd)

✓ **I2C0CON** (E8h) : I²C0 Control Register

-	-	-	-	I2CRST	MV_BF	I2CIOEN	I2CEN
				R/W(0)	R/W(0)	R/W(0)	R/W(0)

- I2CRST : Reset I2C0 module
(Set by S/W, cleared by H/W automatically)
- MV_BF : Move Double Buffer's data into Output Buffer
[1] : Move the data (Set by S/W, cleared by H/W automatically)
During I2C0 is busy, moving action is held until I2C0 operation ends.
- I2CIOEN : Enable I2C IO
[0] : Disable I2C IO [1] : Enable I2C IO
- I2CEN : Enable I2C module
[0] : Disable I2C module [1] : Enable I2C module

✓ **ALTSEL** (85h) : Port Alternative Function Selection

-	-	I2C_A	PWM_A	-	SPI_A	XTAL_IOEN	RST_IOEN
		R/W(0)	R/W(0)		R/W(0)	R/W(1)	R/W(0)

- I2C_A : I2C0/1 port switching selection
[0] : I2C1 (SDA : P3.0, SCL : P3.1), I2C0 (SDA : P3.3, SCL : P3.4)
[1] : I2C0 (SDA : P3.0, SCL : P3.1), I2C1 (SDA : P3.3, SCL : P3.4)

✓ **I2C0ST** (E7h) : I²C0 Status Register

-	-	-	I2C0TIF	I2C0RIF	I2C0OF	I2C0COL	I2C0BF
			R/W(0)	R/W(0)	R/W(0)	R/W(0)	R (0)

- I2C0TIF : I2C0 TX Interrupt Flag in slave & master mode.
[0] : Idle [1] : Interrupt occurred.
It is set each time a byte is transmitted.
The flag is set by H/W and cleared by S/W.
- I2C0RIF : I2C0 RX Interrupt Flag in slave & master mode.
[0] : Idle [1] : Interrupt occurred.
It is set each time a byte is received.
The flag is set by H/W and cleared by S/W.
- I2C0OF : I2C0 Overflow Flag in write mode
[0] : Idle [1] : Overflow occurred.
It is set when the bytes are received while I2C buffers are still holding the previous bytes.
It is set by H/W and cleared by S/W
- I2C0COL : I2C0 Collision Flag in read mode
[0] : Idle [1] : Collision occurred.
It is set when the bytes are not transmitted before I2C buffers are updated.
It is set by H/W and cleared by S/W
- I2C0BF : Busy flag in slave & master mode
[0] : RX not complete (Receiver), TX not complete (Transmitter)
[1] : RX complete (Receiver), TX complete (Transmitter)

6.11. I2C0 : SFR (Cont'd)

✓ **I2C0TDAT** (ECh) : I²C0 TX 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)

- DATA : I2C0 Double Buffer Data

✓ **I2C0TIDX** (EDh) : I²C0 TX Data Index Register

-	-	-	-	INDX3	INDX2	INDX1	INDX0
				R/W(0)	R/W(0)	R/W(0)	R/W(0)

- IDNX : I2C0 I2C0DAT Buffer Index

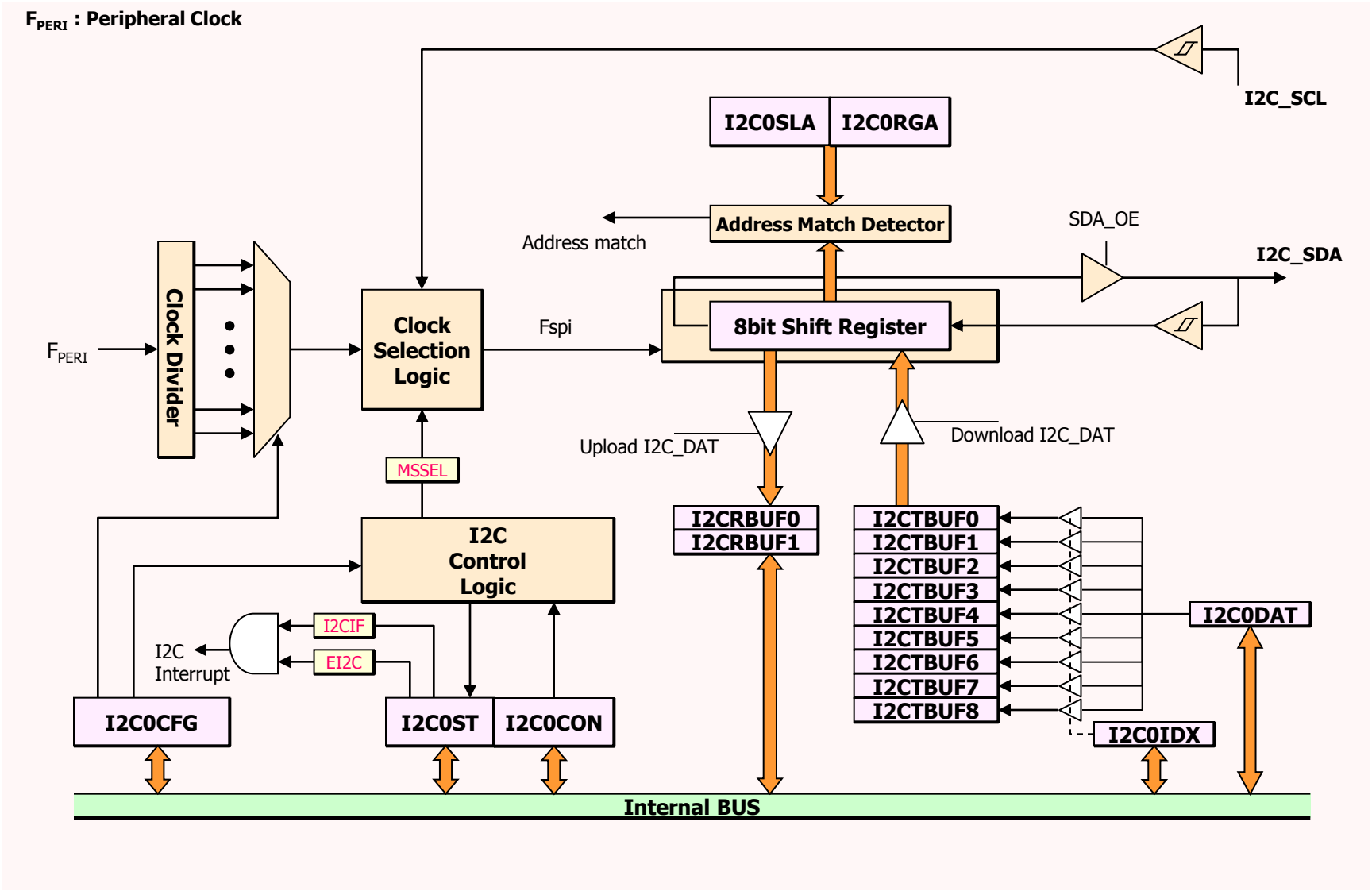
✓ **I2C0RBF0** (EEh) : I²C0 Data RX Buffer0 Register

RBUF07	RBUF06	RBUF05	RBUF04	RBUF03	RBUF02	RBUF01	RBUF00
R (0)	R (0)	R (0)	R (0)	R (0)	R (0)	R (0)	R (0)

✓ **I2C0RBF1** (EFh) : I²C0 Data RX Buffer1 Register

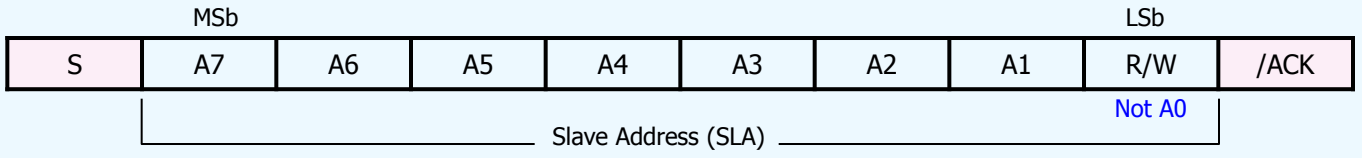
RBUF17	RBUF16	RBUF15	RBUF14	RBUF13	RBUF12	RBUF11	RBUF10
R (0)	R (0)	R (0)	R (0)	R (0)	R (0)	R (0)	R (0)

6.11. I2C0 : Block Diagram

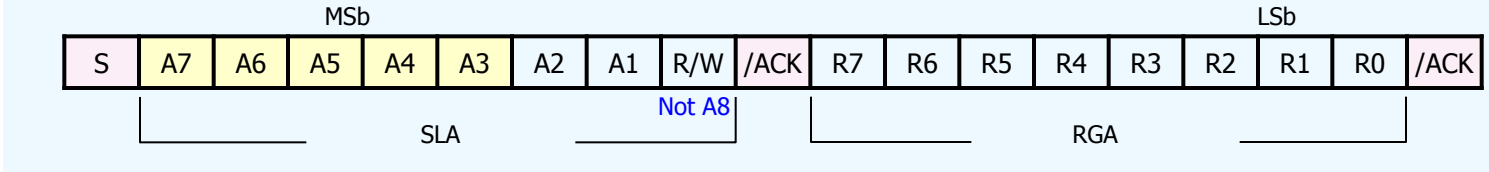


6.11. I2C0 : Overview

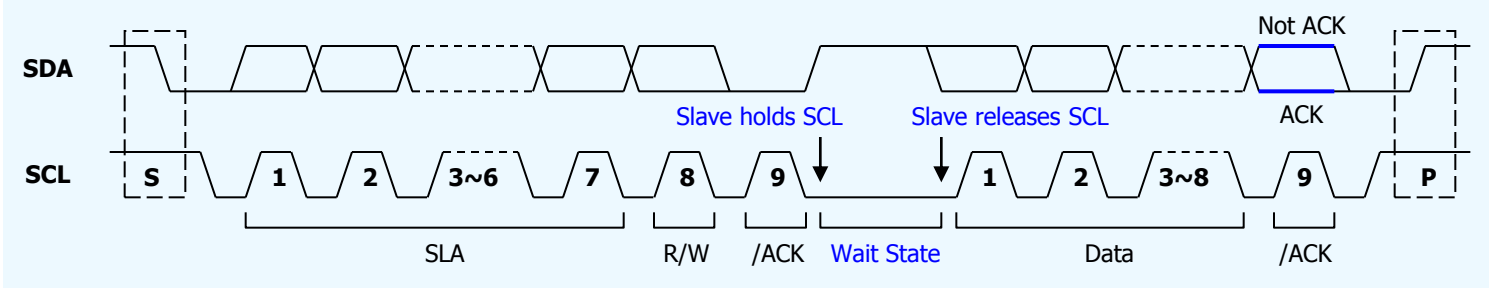
- Addressing I2C devices
  - ✓ 7-bit Address Format



  - ✓ 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.

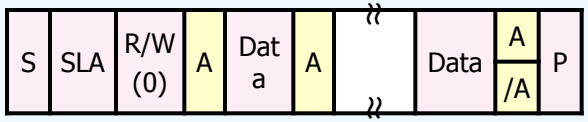


6.11. I2C0 : Overview

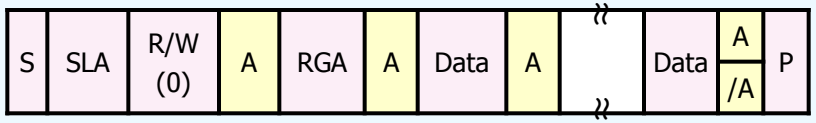
Master-Transmitter Sequence

From Master to Slave | From Slave to Master

[7-bit Address Mode]

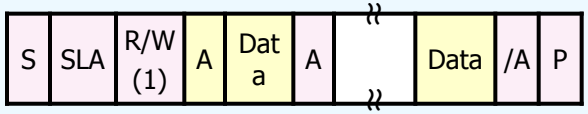


[10-bit / Extended 15-bit Address Mode]



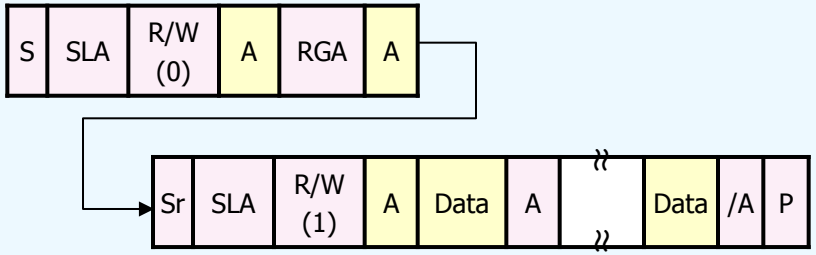
Master-Receiver Sequence

[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. I2C1 : SFR

- Two-wire Interface
- Master or Slave Operation
- Transmitter or Receiver Operation
- 100Kbps (Min. F_{SYS} = 1MHz), 400Kbps (Min. F_{SYS} = 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	EI2C1	EI2C0	EWDT	-	EX4	EX3	EX2
R/W(0)	R/W(0)	R/W(0)	R/W(0)		R/W(0)	R/W(0)	R/W(0)

- EI2C1 : I2C1 Interrupt Enable
- EI2C0 : IC20 Interrupt Enable

✓ **I2C1SLA** (FBh) : I²C1 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.

✓ **I2C1DAT** (FCh) : I²C1 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)

✓ **I2C1CFG** (FAh) : I²C1 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)

✓ **I2C1SCL** (FDh) : I²C1 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

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

✓ **ALTSEL** (85h) : Port Alternative Function Selection

-	-	I2C_A	PWM_A	-	SPI_A	XTAL_IOEN	RST_IOEN
		R/W(0)	R/W(0)		R/W(0)	R/W(1)	R/W(0)

- I2C_A : I2C0/1 port switching selection
  [0] : I2C1 (SDA : P3.0, SCL : P3.1), I2C0 (SDA : P3.3, SCL : P3.4)
  [1] : I2C0 (SDA : P3.0, SCL : P3.1), I2C1 (SDA : P3.3, SCL : P3.4)

6.12. I2C1 : SFR (Cont'd)

✓ **I2C1CON** (F9h) : I²C1 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)

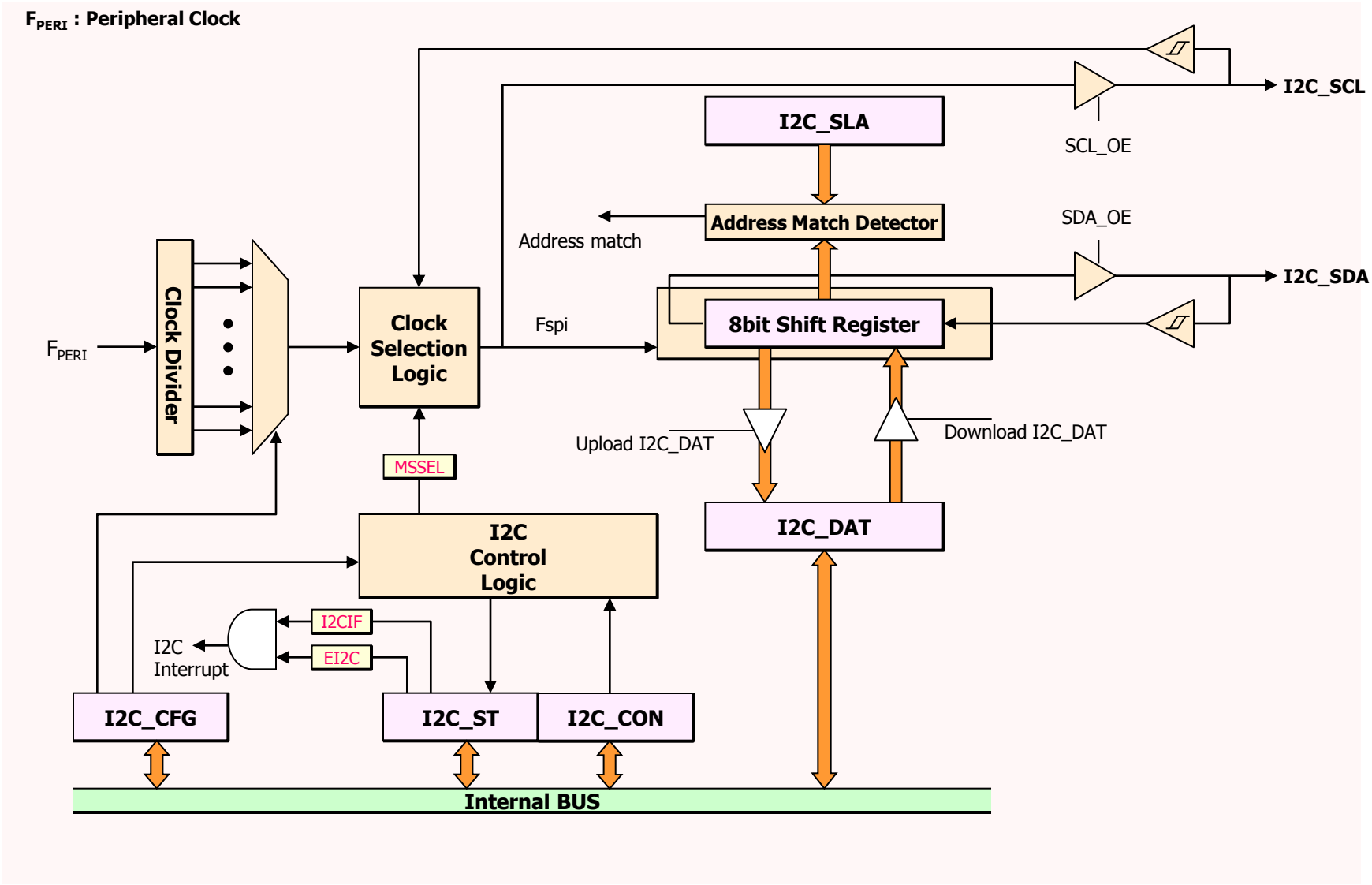
- SLA2ME : 2nd Byte Slave Address Match Enable in Slave mode
  [0] : 2nd Byte SLA Match Disable [1] : 2nd SLA Byte Match Enable
- SCLHD : Hold SCL 'low' for Wait State in Slave mode.
  [0] : Hold SCL 'low'. The flag is cleared automatically by H/W
  [1] : Release SCL 'float'. The flag is set by S/W
-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

✓ **I2C1ST** (F8h) : I²C1 Status Register

I2C1IF	I2C1OF	I2C1ACK	I2C1RW	I2C1DA	I2C1P	I2C1S	I2C1BF
R/W(0)	R/W(0)	R (0)	R (0)	R (0)	R (0)	R (0)	R (0)

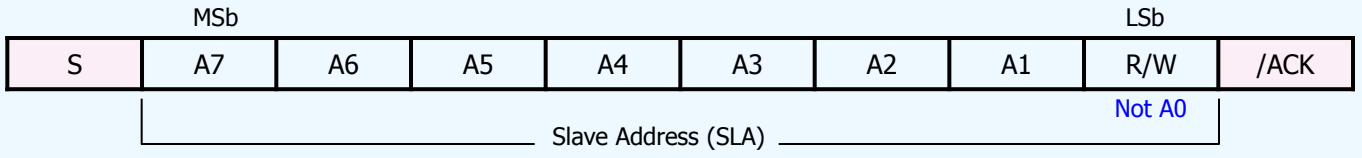
- I2C1IF : I2C1 Master Interrupt Flag in slave & master mode.
  [0] : Idle [1] : Interrupt occurred.
  It is set each time a byte is received or transmitted.
  If SP_IE flag in I2C_CFG SFR is set, it is set at Start/Stop condition.
  The flag is set by H/W and cleared by S/W.
- I2C1OF : I2C1 Overflow 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
- I2C1ACK : I2C1 Acknowledge flag in slave & master mode.
  [0] : Indicate receiving Acknowledge bit.
  [1] : Indicate receiving Not Acknowledge bit.
- I2C1RW : I2C1 Read/Write flag in slave mode
  [0] : Write state [1] : Read state
- I2C1DA : 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
- I2C1P : 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.
- I2C1S : 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.
- I2C1BF : Busy flag in slave & master mode
  [0] : RX not complete (Receiver), TX not complete (Transmitter)
  [1] : RX complete (Receiver), TX complete (Transmitter)

6.12. I2C1 : Block Diagram

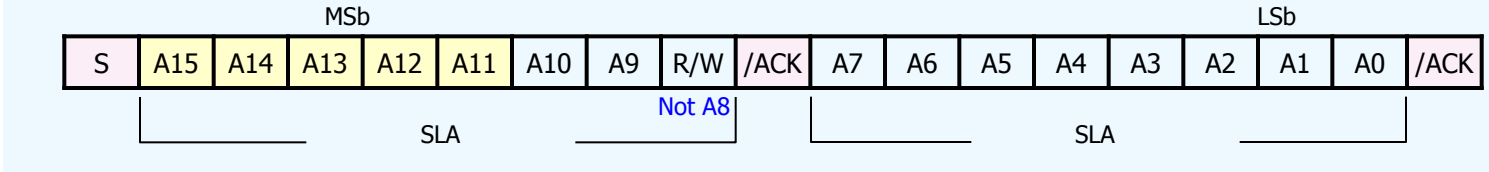


6.12. I2C1 : Overview

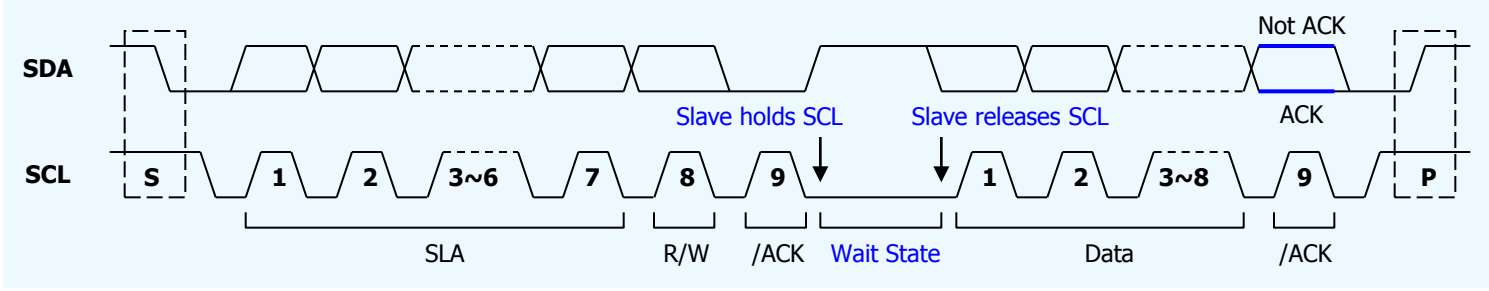
- Addressing I2C devices
  - ✓ 7-bit Address Format



  - ✓ 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.

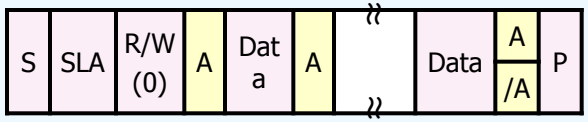


6.12. I2C1 : Overview

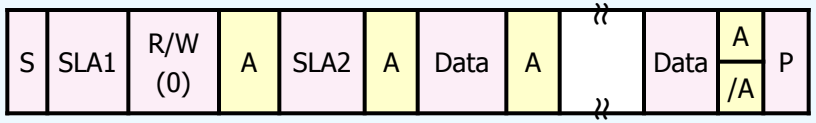
Master-Transmitter Sequence

From Master to Slave | From Slave to Master

[7-bit Address Mode]

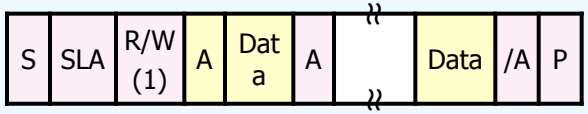


[10-bit / Extended 15-bit Address Mode]



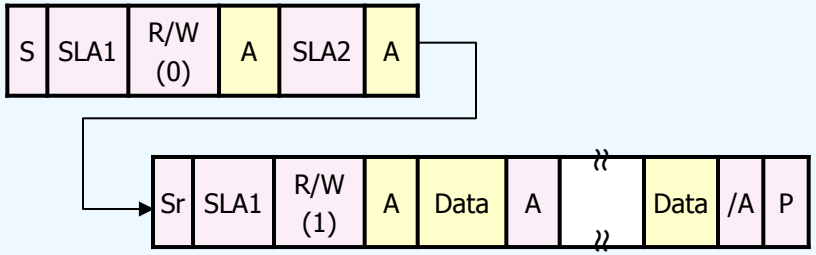
Master-Receiver Sequence

[7-bit Address Mode]



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

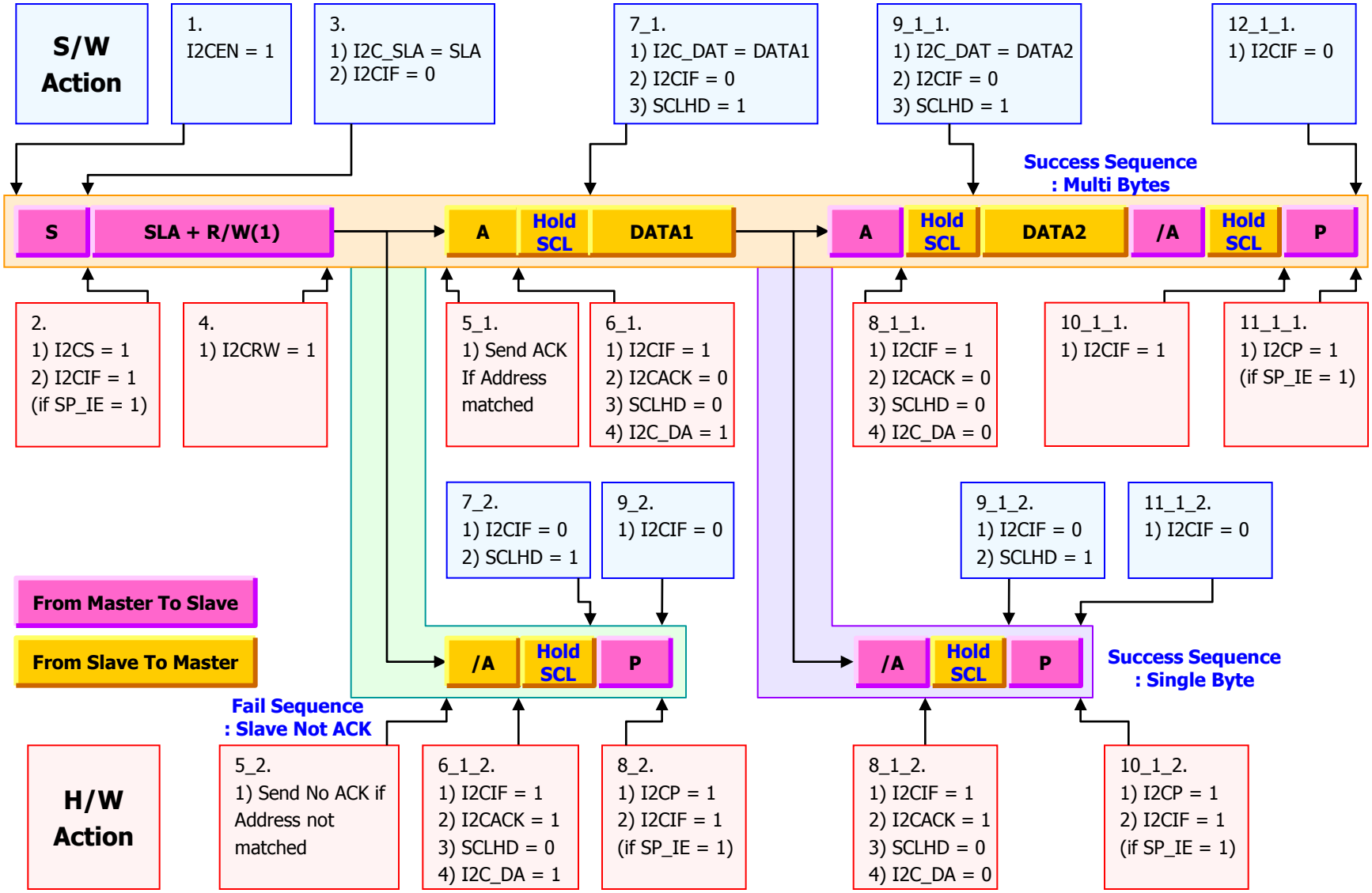
[10-bit / Extended 15-bit Address Mode]



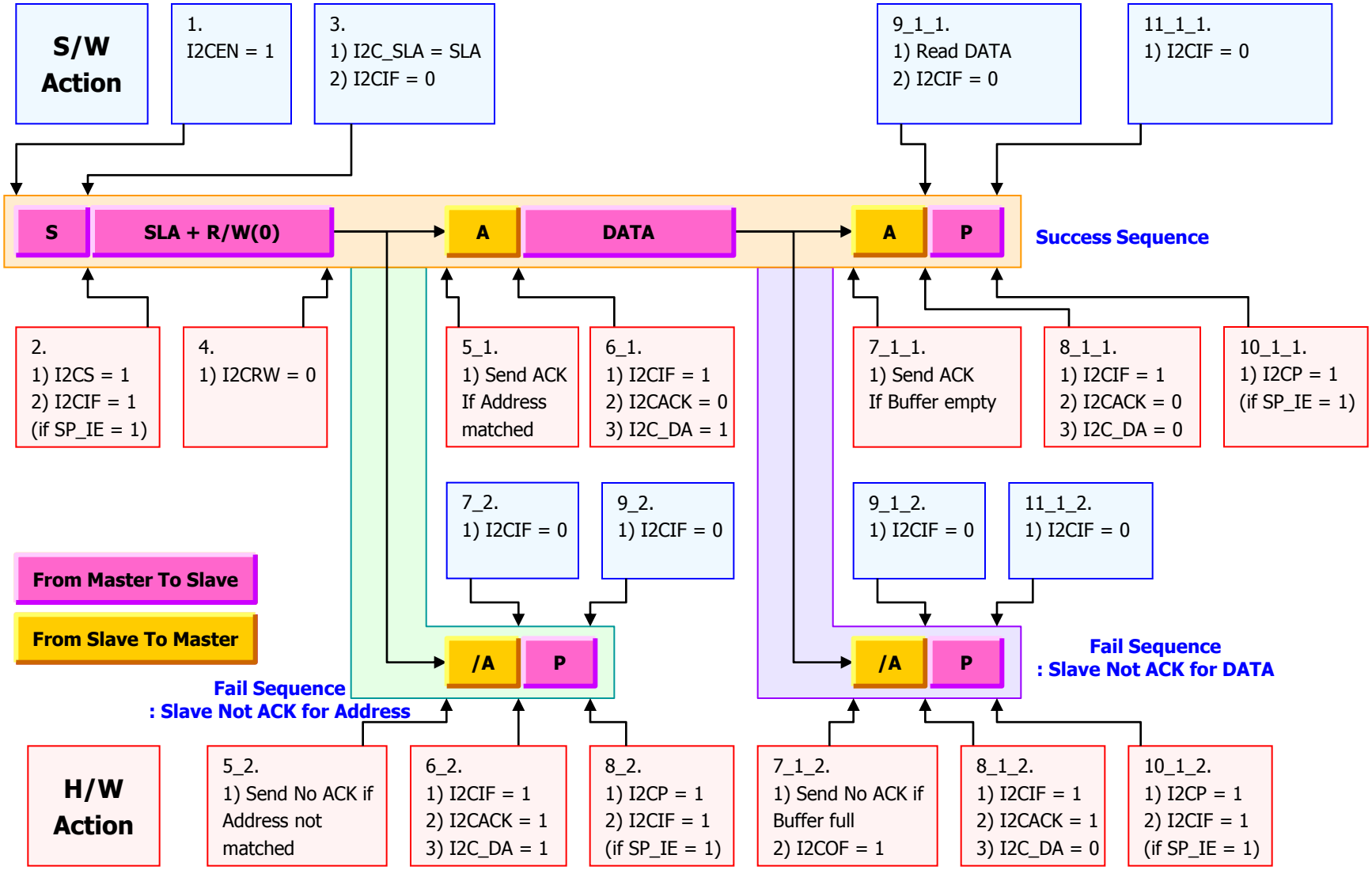
Combined Format

- ✓ 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. I2C1 : Slave Transmitter Flow



6.12. I2C1 : Slave Receiver Flow



6.12. I2C1 : Slave Example

- I2C1 Slave example code using interrupt

```
        I2CST     EQU  0F8H          ; I2CST SFR
        I2CIF     EQU  0FFH          ; I2CST.7 Flag
        I2COF     EQU  0FEH          ; I2CST.6 Flag
        I2CACK    EQU  0FDH          ; I2CST.5 Flag
        I2CRW     EQU  0FCH          ; I2CST.4 Flag
        I2CDA     EQU  0FBH          ; I2CST.3 Flag
        I2CP      EQU  0FAH          ; I2CST.2 Flag
        I2CS      EQU  0F9H          ; I2CST.1 Flag
        I2CBF     EQU  0F8H          ; I2CST.0 Flag

        I2CCON    EQU  0F9H
        I2CCFG    EQU  0FAH
        I2CSLA    EQU  0FBH
        I2CDAT    EQU  0FCH
        I2CSCL    EQU  0FDH

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
  ORL     EIE,        #20h         ; 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
  ANL     EIE,        #0DFh        ; 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)
  ORL     EIE,        #20h         ; enable I2C interrupt
  MOV     OSCICN,     #0Fh         ; restore clock speed
  RETI
```

6.13. SPI : SFR

- Full-duplex, Three-wire Synchronous Data Transfer
- 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
  [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)

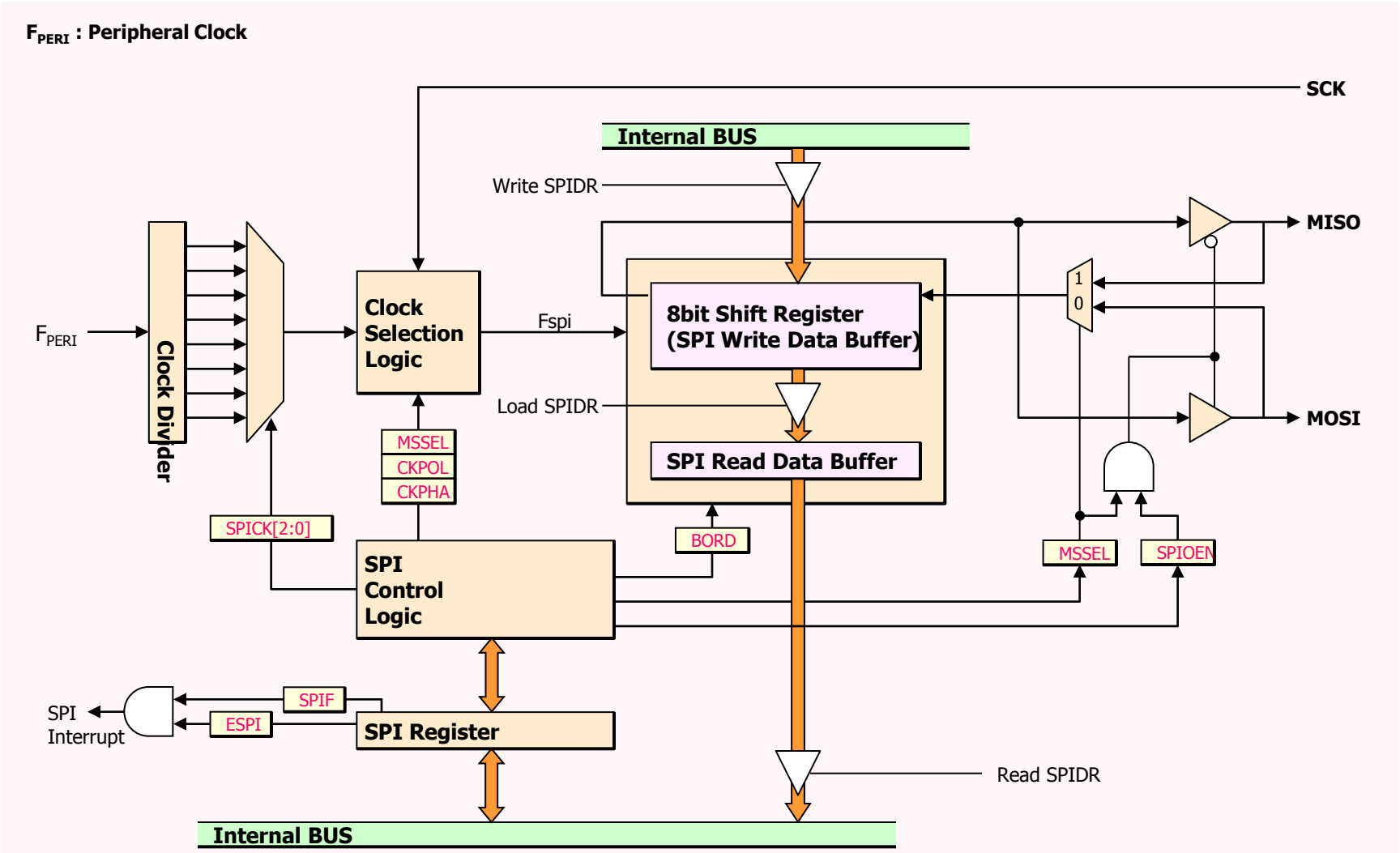
- SPICK[2:0] : SPI Master Clock Divider
  [0,0,0] : F_{SYS} / 2 [0,0,1] : F_{SYS} / 4
  [0,1,0] : F_{SYS} / 8 [0,1,1] : F_{SYS} / 16
  [1,0,0] : F_{SYS} / 32 [1,0,1] : F_{SYS} / 64
  [1,1,0] : F_{SYS} / 128 [1,1,1] : F_{SYS} / 256

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

ESPI	EI2C1	EI2C0	EWDT	-	EX4	EX3	EX2
R/W(0)	R/W(0)	R/W(0)	R/W(0)		R/W(0)	R/W(0)	R/W(0)

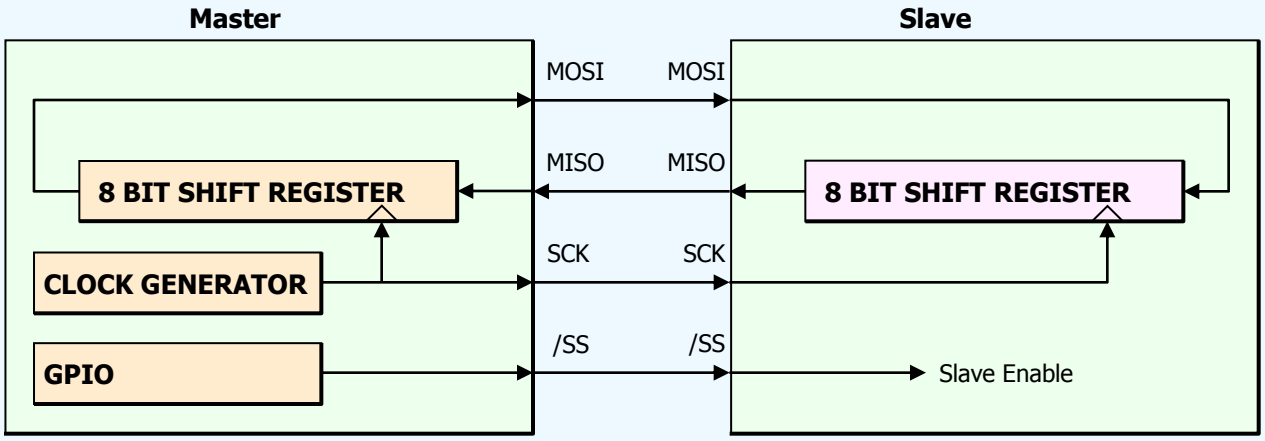
- ESPI : SPI Interrupt Enable

6.13. SPI : Block Diagram



6.13. SPI : Overview

- SPI Slave Interconnection

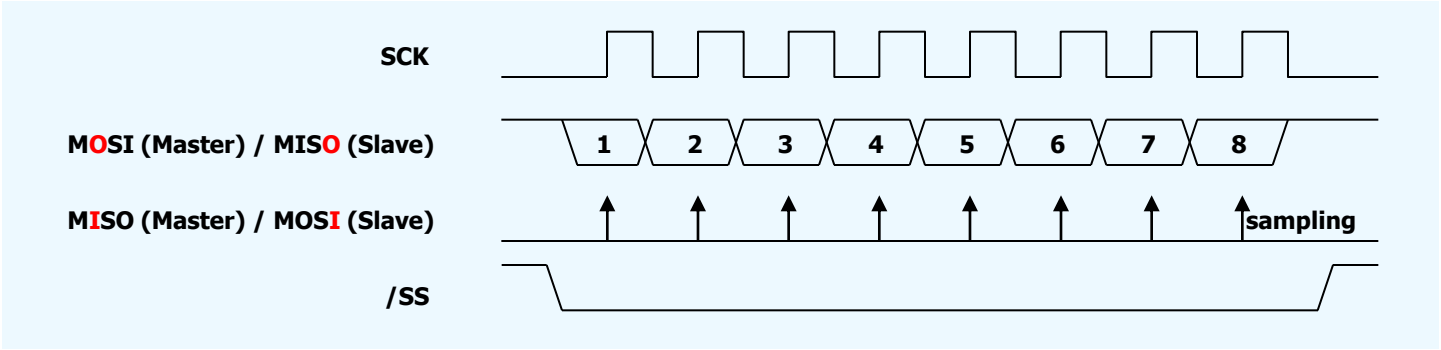


- SPI Pin Description

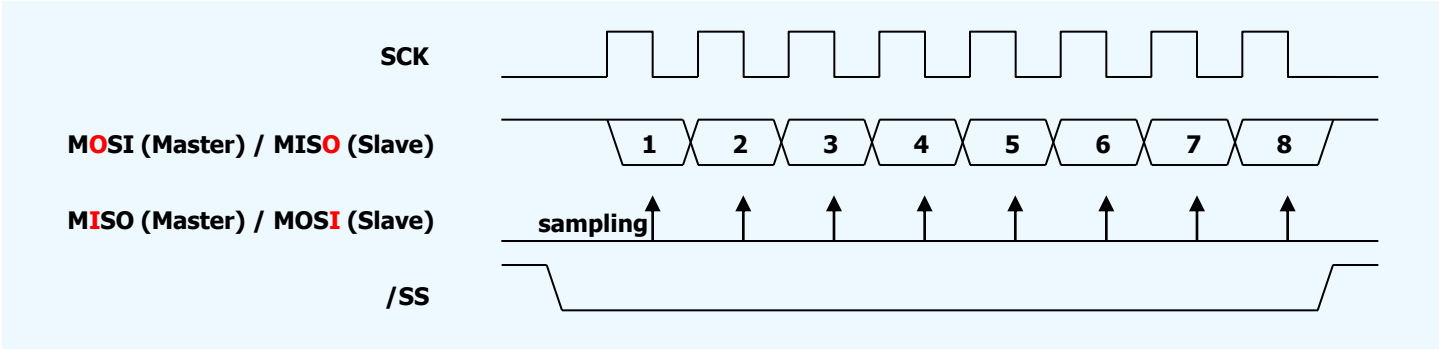
Pin	Description	Direction, Master	Direction, Slave
MOSI	Master Output Slave Input	User Defined	Input
MISO	Master Input Slave Output	Input	User Defined
SCK	SPI Clock	User Defined	Input
/SS	Slave Select Bar	User Defined	Input

6.13. SPI : Mode 0/1

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

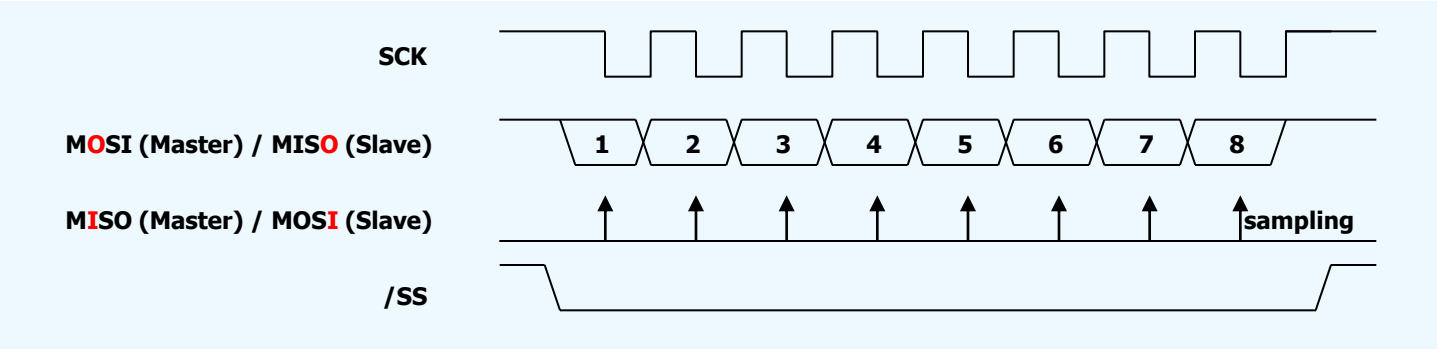


- SPI Mode 1
  - ✓ CKPOL = 0 : Leading Edge → Rising
  - ✓ CKPHA = 1 : Trailing Edge → Sampling

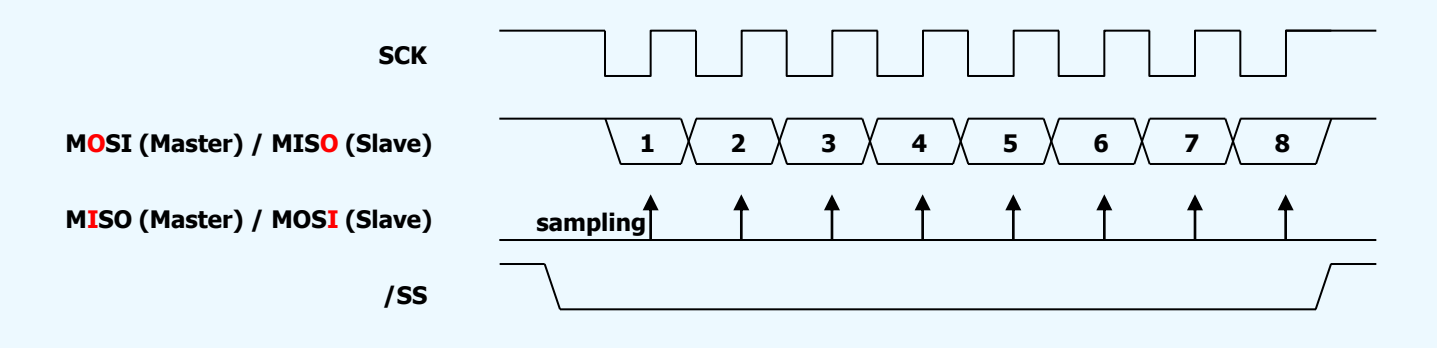


6.13. SPI : Mode 2/3

- 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



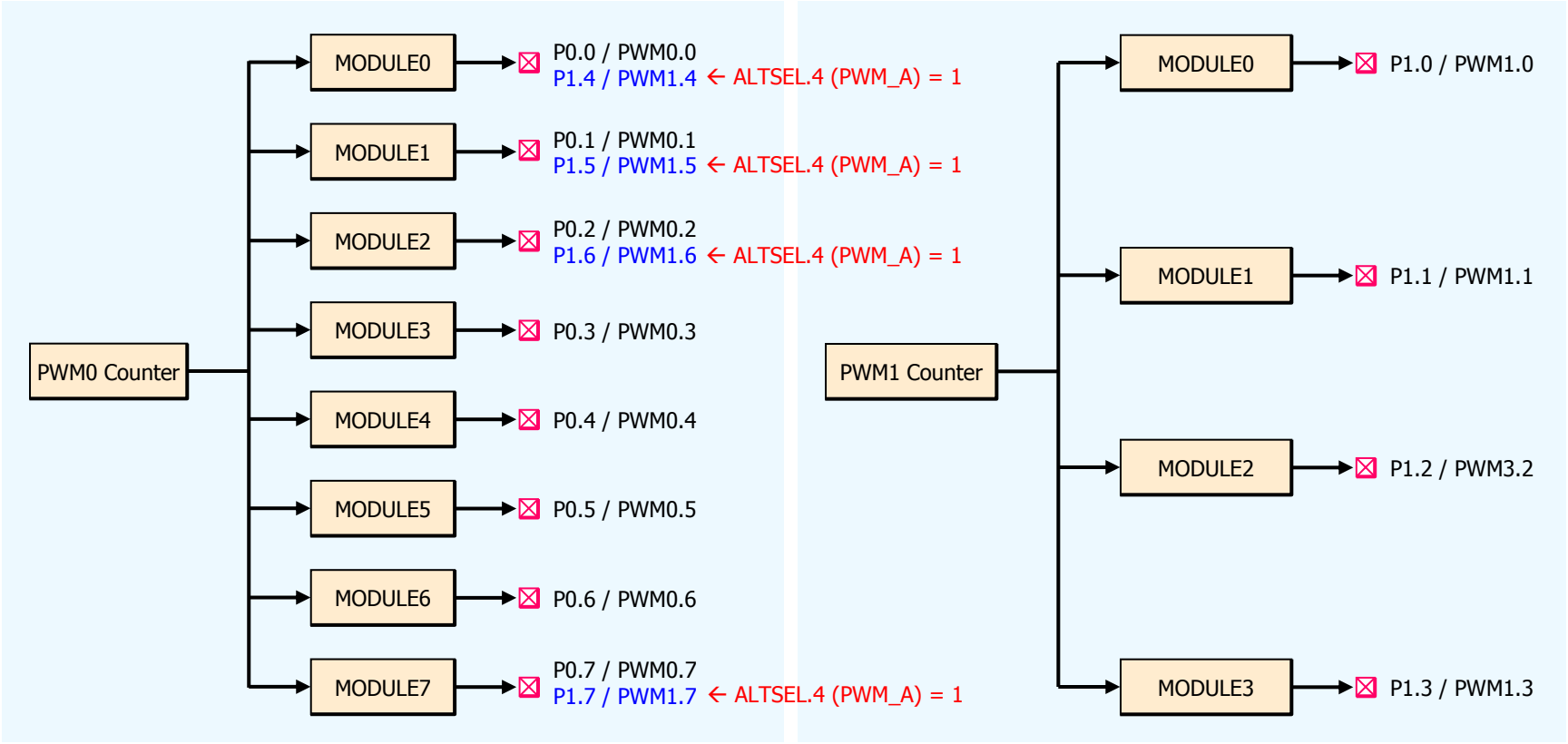
6.13. SPI : Example

- Slave example code

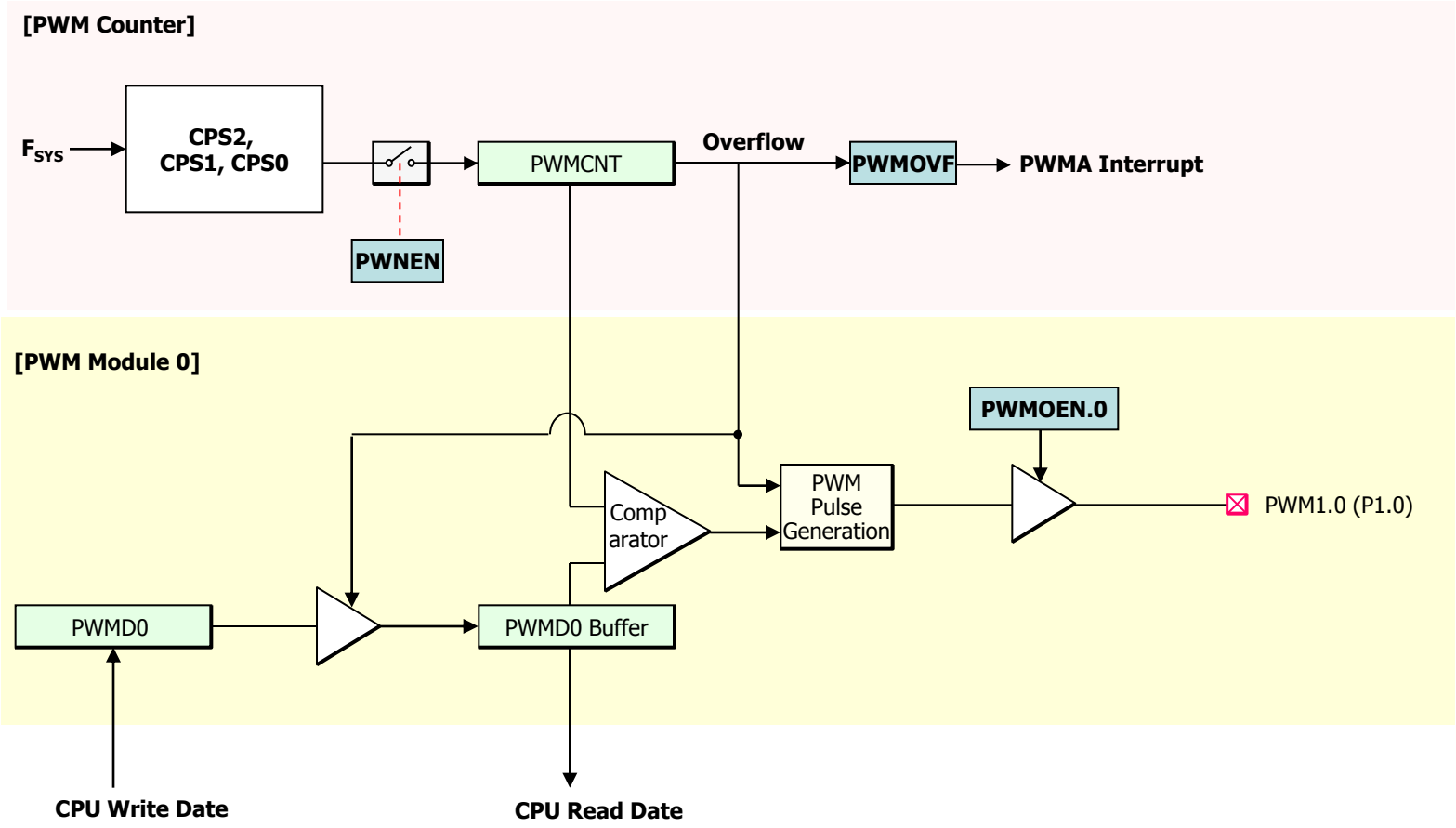
```
SPIST   EQU     0C0H
SPICON  EQU     0B4H
SPICK   EQU     0B5H
SPIDR   EQU     0B6H

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. PWMA (PWM Arrays)

- PWMA
  - ✓ Two 8-bit PWM generation with 8 modules (Compatible to M1.0B 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.



6.14. PWMA : Block Diagram



6.14. 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)	R/W(0)

- CPS2, CPS1, CPS0 : PWMA counter frequency selection.
  [0,0,0] = F_{SYS} / 1 ; Default
  [0,0,1] = F_{SYS} / 2
  [0,1,0] = F_{SYS} / 4
  [0,1,1] = F_{SYS} / 8
  [1,0,0] = F_{SYS} / 16
  [1,0,1] = F_{SYS} / 32
  [1,1,0] = F_{SYS} / 64
  [1,1,1] = F_{SYS} / 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

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

✓ **ALTSEL** (85h) : Port Alternative Function Selection

-	-	I2C_A	PWM_A	UART_A	SPI_A	XTAL_IOEN	RST_IOEN
		R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(1)	R/W(0)

- PWM_A : PWM alternative function selection
  [1] : PWM0[7,2,1,0] PWM1[7:4]A

6.14. 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

6.14. 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_{SYS} / 1 ; Default
  [0,0,1] = F_{SYS} / 2
  [0,1,0] = F_{SYS} / 4
  [0,1,1] = F_{SYS} / 8
  [1,0,0] = F_{SYS} / 16
  [1,0,1] = F_{SYS} / 32
  [1,1,0] = F_{SYS} / 64
  [1,1,1] = F_{SYS} / 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

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

- OE3 : Module 3 PWM output enable
- OE2 : Module 2 PWM output enable
- OE1 : Module 1 PWM output enable
- OE0 : Module 0 PWM output enable

6.14. 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)

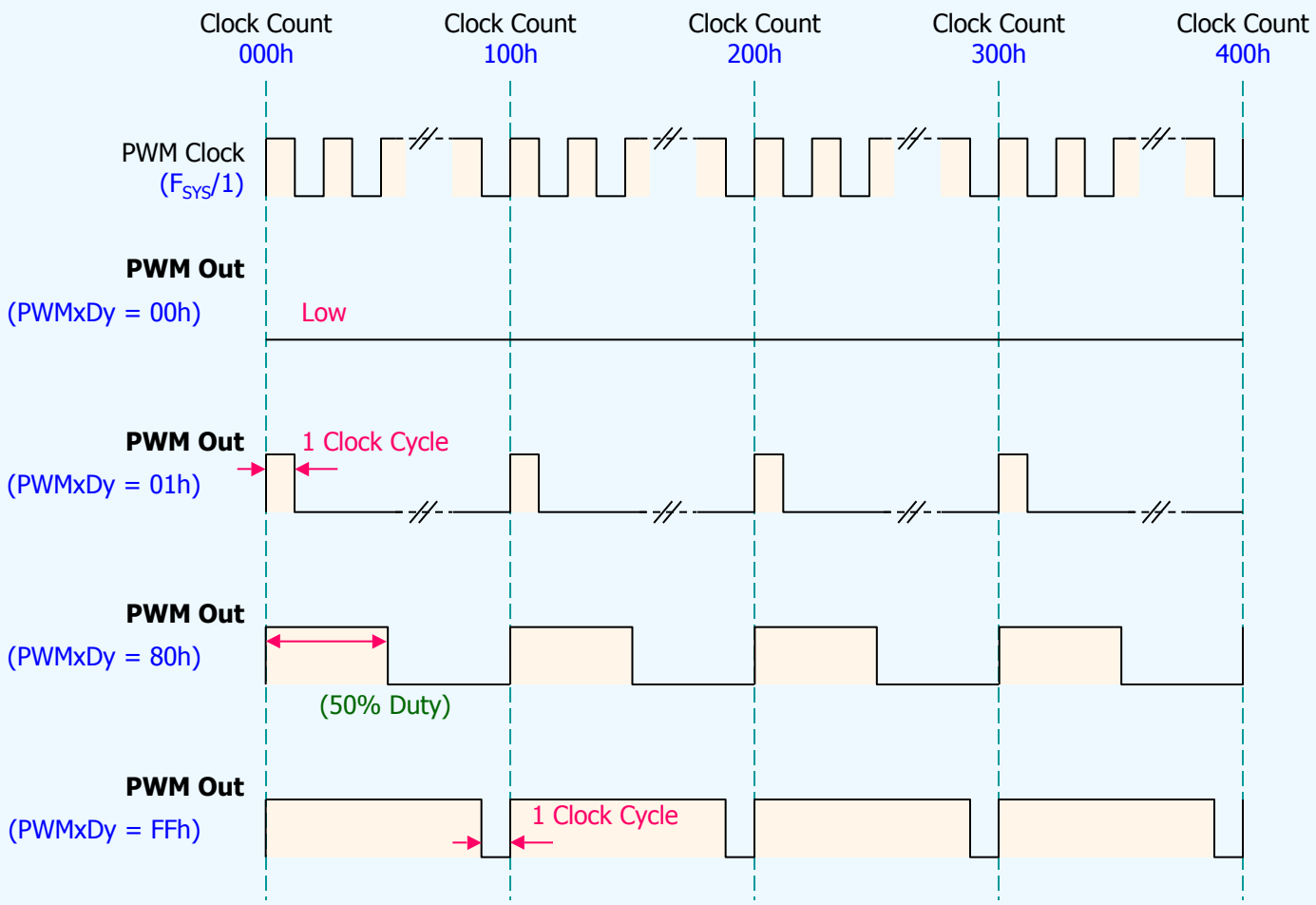
- Each Module has a internal buffer register for the duty data register.
  The buffer register is updated with the new data
  whenever the PWMA counter rolls over.
  When user write, the data register is written.
  When user read, the contents of buffer register is read out.

✓ **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

6.14. PWMA : Pulse Generation Example



6.15. Interrupt : 13 Sources / 4&2-level Priority

- Interrupt Sources : Timer0/1/2, WDT, I2C0/1, UART, SPI, & 5 External Interrupt Sources.
- 4-level Interrupt Priority
  - ✓ Timer 0/1/2, UART, INT0, INT1
- 2-level Interrupt Priority
  - ✓ WDT, I2C0/1, INT2, INT3, INT4, SPI, TS

[Interrupt Vector Address]

HIGH

PRIORITY

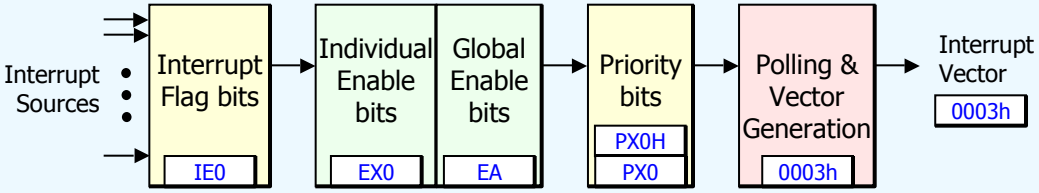
LOW

Interrupt Sources	Address	Priority Level	
LVD	0033h	NMI	8052
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	
-	003Bh	4 Levels	
INT2	0043h	2 Levels	
INT3	004Bh	2 Levels	
INT4	0053h	2 Levels	
-	005Bh	2 Levels	
WDT	0063h	2 Levels	
I2C0	006Bh	2 Levels	
I2C1	0073h	2 Levels	
SPI	007Bh	2 Levels	
-	0083h	2 Levels	

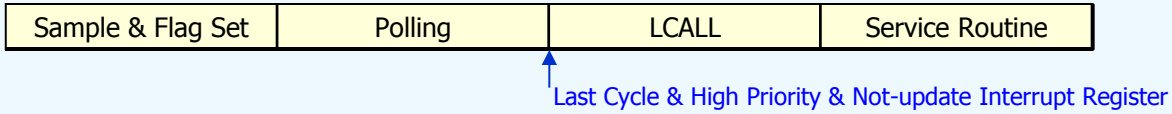
*** Interrupt SFR's (refer to Appendix B : SFR Description)**

✓ **TCON**	(88h)	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
✓ **IT**	(B2h)	-	-	-	-	-	IT4	IT3	IT2
✓ **ITSEL**	(BAh)	-	-	-	ITSEL4	ITSEL3	ITSEL2	ITSEL1	ITSEL0
✓ **EXIF**	(91h)	-	IE4	IE3	IE2	XT/RL	RGM0	RGSL	BGS
✓ **IE**	(A8h)	EA	-	ET2	ES	ET1	EX1	ET0	EX0
✓ **EIE**	(A1h)	ESPI	EI2C1	EI2C0	EWDT	-	EX4	EX3	EX2
✓ **IP**	(B8h)	-	-	PT2	PS	PT1	PX1	PT0	PX0
✓ **IPH**	(B7h)	-	-	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
✓ **EIP**	(B1h)	PSPI	PI2C1	PI2C0	PWDT	-	PX4	PX3	PX2
✓ **WDCON**	(D8h)	WDMOD	POR	-	-	WDIF	WTRF	EWT	RWT

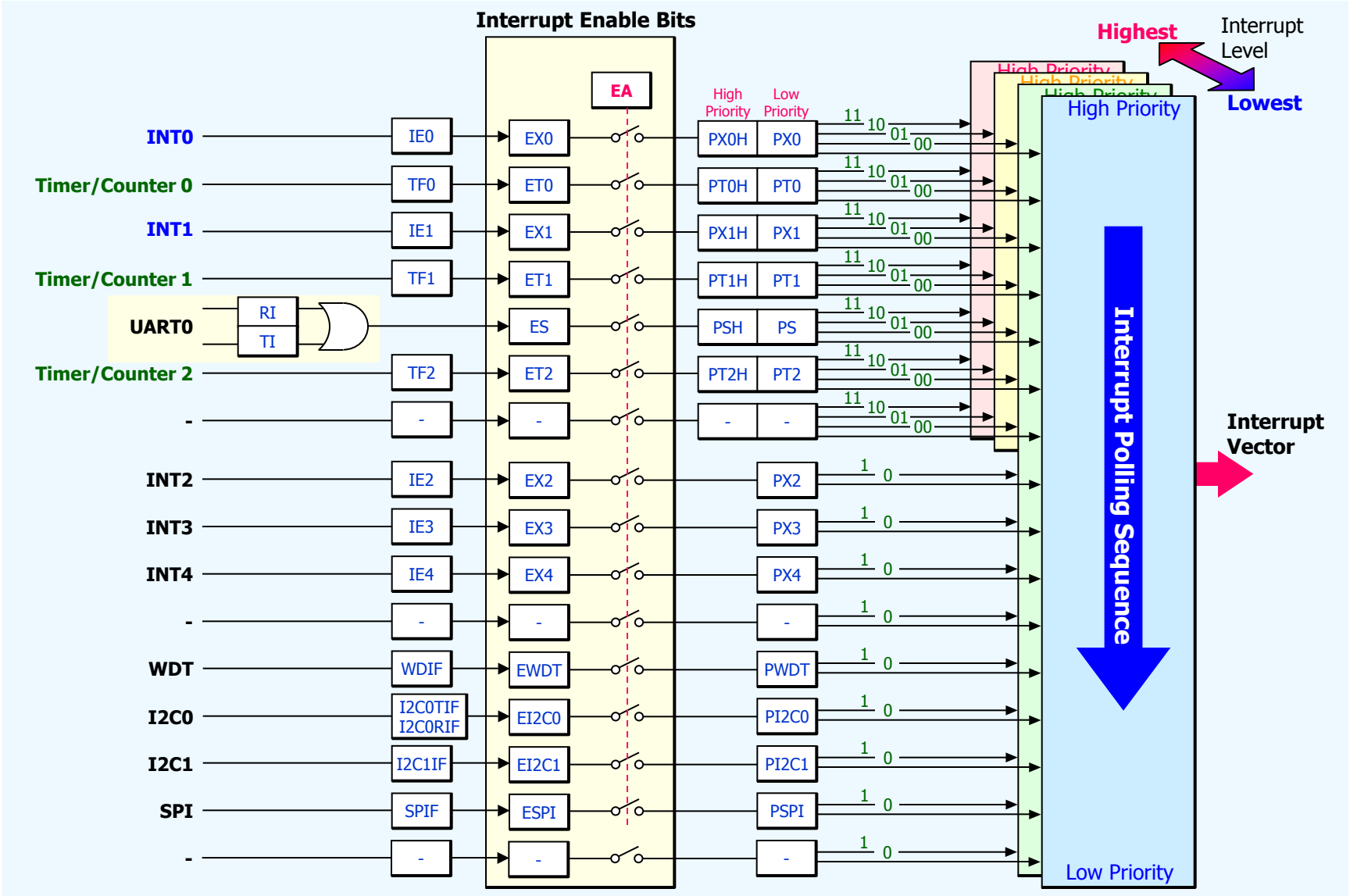
[Interrupt Vector Generation Flow]



[Response Sequence]



6.15. Interrupt Functional Description



6.15. Interrupt : External Interrupt

- External Interrupt Sources : INT4~0
- Support positive edge and negative edge detection
- Support high level and low level detection

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

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

- IT4 : External Interrupt 4 Type Select
  Edge Detect (IT4=1). Level Detect (IT4=0)
- IT3 : External Interrupt 3 Type Select
  Edge Detect (IT3=1). Level Detect (IT3=0)
- IT2 : External Interrupt 2 Type Select
  Edge Detect (IT2=1). Level Detect (IT2=0)

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

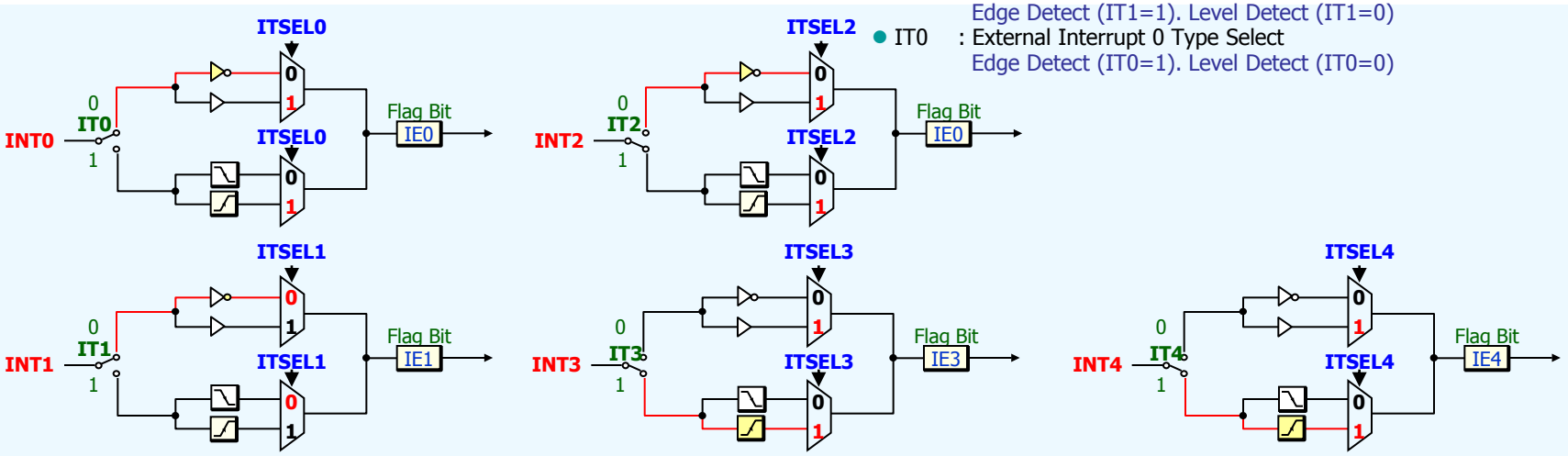
-	-	-	ITSEL4	ITSEL3	ITSEL2	ITSEL1	ITSEL0
			R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- 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

✓ **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)

- IT1 : External Interrupt 1 Type Select
  Edge Detect (IT1=1). Level Detect (IT1=0)
- IT0 : External Interrupt 0 Type Select
  Edge Detect (IT0=1). Level Detect (IT0=0)



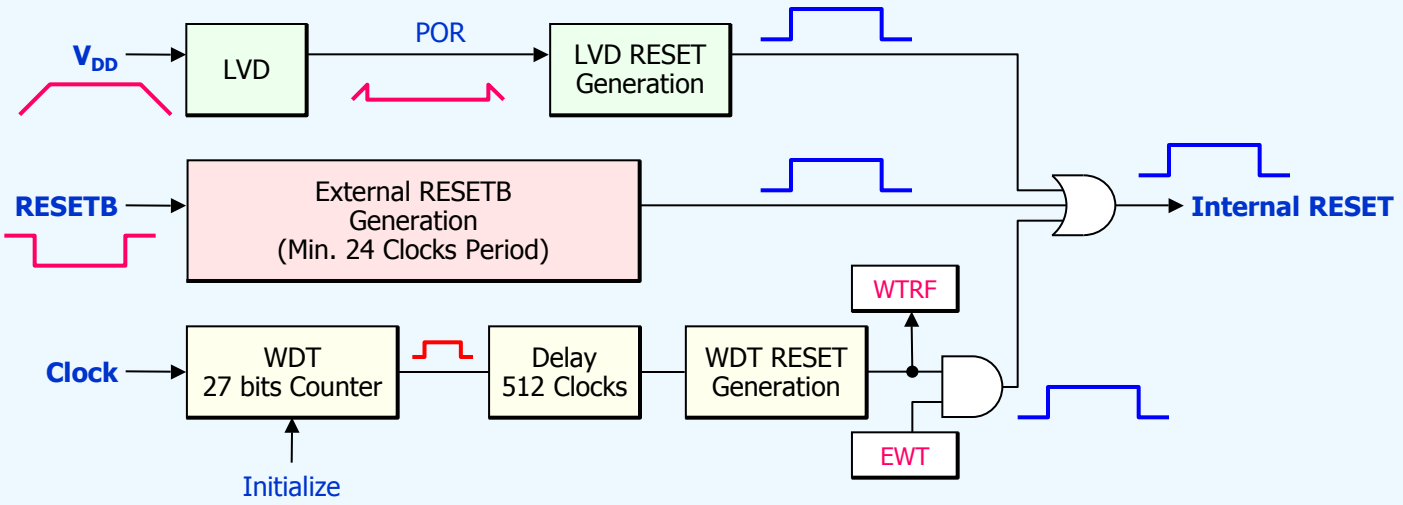
6.16. Reset Circuit : 3 Reset Sources

- 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 "LOW" for at least 24 clock cycles.
- WDT Reset : Enable or Disable by S/W

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

-	POR	-	-	WDIF	WTRF	EWT	RWT
	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 (F_{OSC})
  - ✓ 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

-	IE4	IE3	IE2	XT/RG	RGMD	RGSL	BGS
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R(1)	R/W(0)	R/W(1)

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

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

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

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

✓ **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.

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

-	-	-	-	-	R32KOE	-	-
					R/W(0)		

- R32KOE : RING 32kHz Port Output Enable (P3.3)

6.17. Clock Circuit : SFR (Cont'd)

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

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

- RINGON : 1 = Internal ring Oscillator is running.
  0 = Internal ring Oscillator is killed.
  Don't clear RINGON bit when XTRG = 0 & RINGON2=0.
- DIV[2:0] : Ring Oscillator divider. (F_{OSC} : 96MHz)
  [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
  [1,1,1] = Not supported

✓ **PCLKEN** (CEh) : Peripheral Clock Control Register

PCLKEN7	PCLKEN6	PCLKEN5	PCLKEN4	PCLKEN3	PCLKEN2	PCLKEN1	PCLKEN0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

- PCLKEN0 : IAP Clock Enable
- PCLKEN1 : Timer0/1 Clock Enable
- PCLKEN2 : Timer2 Clock Enable
- PCLKEN3 : UART Clock Enable
- PCLKEN4 : SPI Clock Enable
- PCLKEN5 : I2C0 Clock Enable
- PCLKEN6 : I2C1 Clock Enable
- PCLKEN7 : PWM Clock Enable

✓ **OSC2ICN** (C7h) : RING2 Oscillator Control Register

–	–	-	-	-	RING2ON	-	-
					R/W(1)		

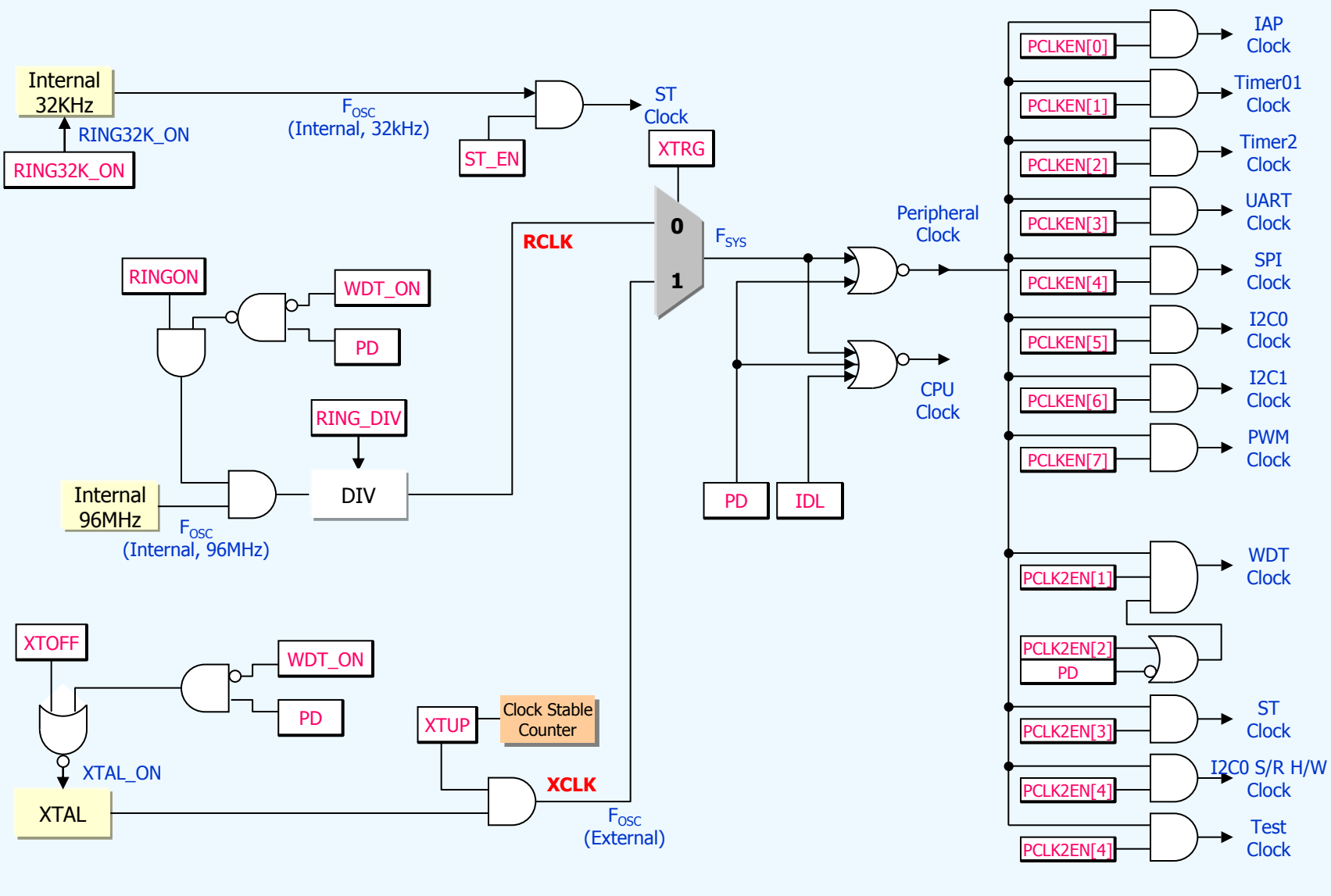
- RING2ON : 1 = Internal RING2 Oscillator is enabled.
  0 = Internal RING2 oscillator is killed.
  (This bit must be value 0 at any operation.)

✓ **PCLK2EN** (D6h) : Peripheral Clock2 Control Register

–	–	PCLK2EN5	PCLK2EN4	PCLK2EN3	PCLK2EN2	PCLK2EN1	-
		R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	

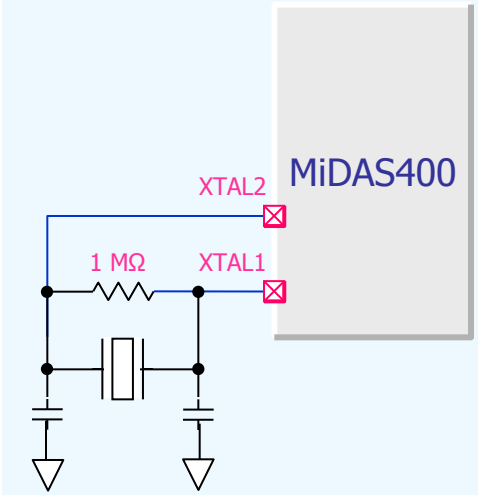
- PCLK2EN1 : WDT Clock Enable
- PCLK2EN2 : WDT Clock Enable ([0] : disabled in stop mode)
- PCLK2EN3 : ST Clock Enable
- PCLK2EN4 : U2C0 Self-running H/W Clock Enable
- PCLK2EN5 : Test Clock Enable

6.17. Clock Circuit : Circuit Diagram

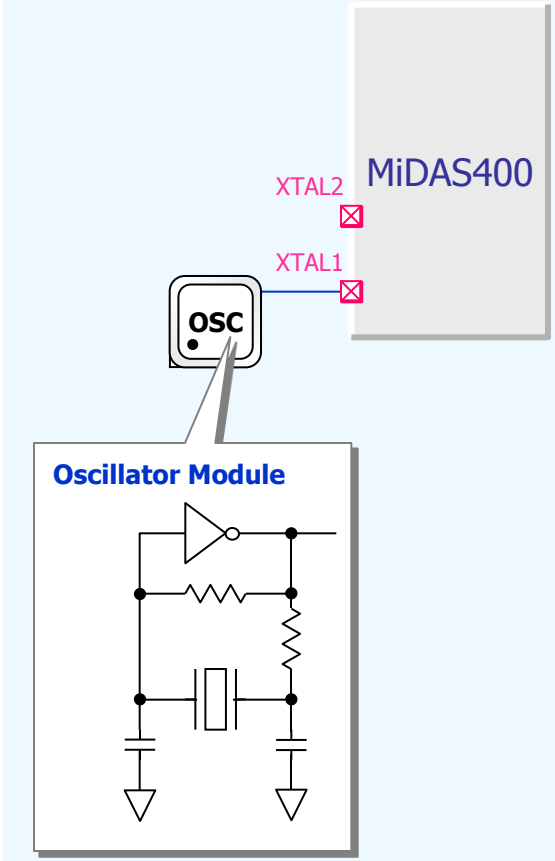


6.17. Clock Circuit : Guideline for Configuration

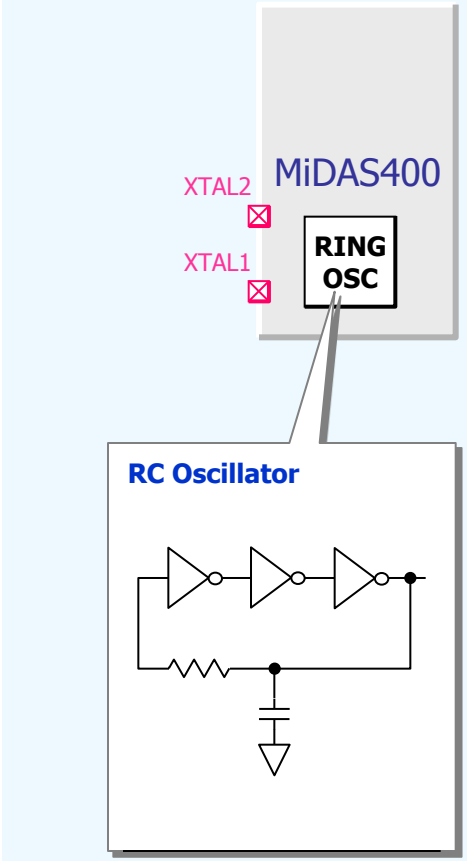
◆ **Crystal Oscillator**



◆ **Oscillator Module**



◆ **Internal Ring Oscillator**



6.17. Clock Circuit : RING OSC. Calibration

- Internal RING Oscillator Calibration
  - ✓ RING Calibration value is saved at 0xFFFF address area.
  - ✓ The calibration value is set to fit the internal RING frequency to 48.00MHz.
  - ✓ User must move the calibration value into RINGCON SFR for using 48.00MHz RING OSC.

[Example Code : Update RING calibration value]

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

```
unsigned char code ringcon_cal _at_ 0xFFFF;

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

■ **RINGCON** (8Fh) : RING Control Configuration Register

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

◆ RINGCON[7:0] : Internal RING OSC. Can be tuned.

■ **OSCICN** (C6h) : Internal Ring Oscillator Control Register

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

◆ RINGON : 1 = Internal ring oscillator(96MHz) is running.
0 = Internal ring oscillator is killed.
Don't clear RINGON bit when XTRG = 0.

◆ DIV2, DIV1, DIV0 : Ring oscillator divider.
[0,0,0] = 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

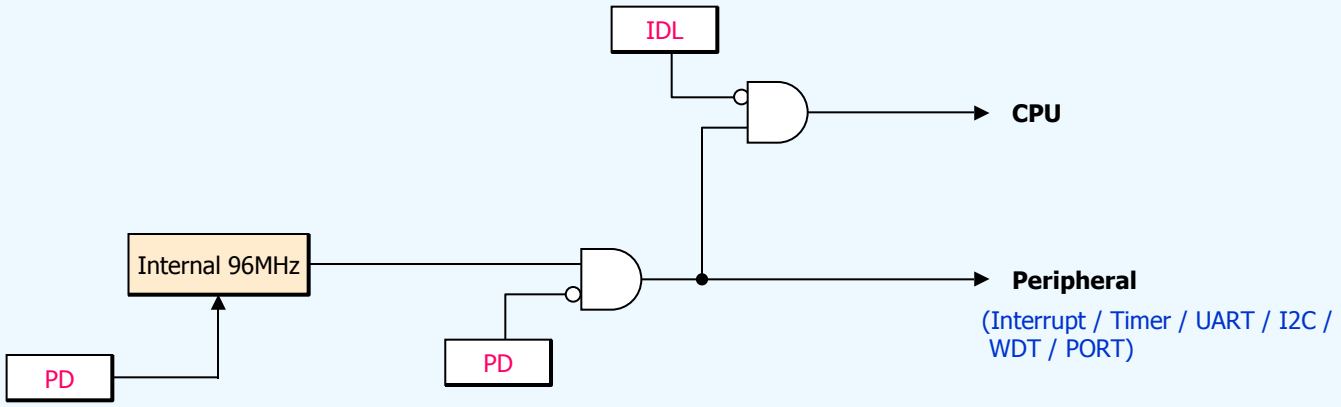
6.18. Power Management : 3 Modes

- **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.



6.19. IAP (In Application Programming)

- Code memory(32kB) & EEPROM(1kB) can be programmed during the operation of MCU.
- Program time : approximately 2 ms / Erase time : approximately 2 ms
- Program unit : 1 Byte / Erase unit : 1kByte
- IAP SFR

✓ **FAEN** (F7h) : IAP Routine Access Enable Register

FAEN7	FAEN6	FAEN5	FAEN4	FAEN3	FAEN2	FAEN1	FAEN0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- FAEN : IAP Routine Access Enable Code Pattern
  FAEN enable pattern flow (1st : 0xC1, 2nd : 0x1E, 3rd : 0xEC, 4th : 0x81)

✓ **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)

✓ **FCNTLD** (F1h) : FLASH Erase/Program Time Count Loading

FCNTLD	-	-	-	-	-	-	-
R/W(0)							

- FCNTLD : FLASH 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)

✓ **FCNTH** (F4h) : FLASH Erase/Program Time Count High

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

✓ **FCNTM** (F3h) : FLASH Erase/Program Time Count Middle

FCNT.15	FCNT.14	FCNT.13	FCNT.12	FCNT.11	FCNT.10	FCNT.9	FCNT.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)

✓ **FCNTL** (F2h) : FLASH Erase/Program Time Count Low

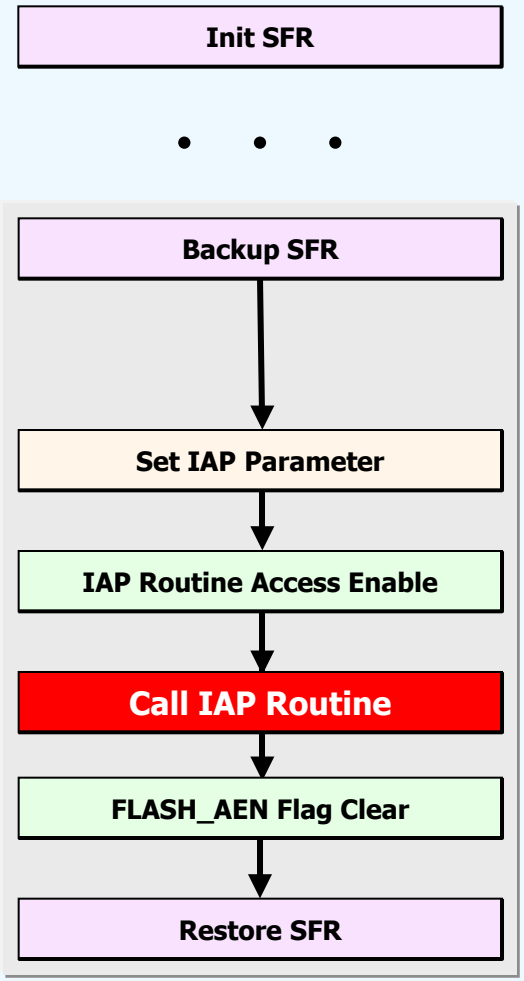
FCNT.7	FCNT.6	FCNT.5	FCNT.4	FCNT.3	FCNT.2	FCNT.1	FCNT.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

6.19. IAP : Function Set

- IAP call function
  - ✓ iap_main_program : call address (FF10h)
  - ✓ iap_eeprom_erase : call address (FF00h)

- Before calling IAP function, any interrupt must be disabled.
- Before calling IAP function, FAEN flag in FAEN SFR must be set.
  - ✓ **Only use ORL/ANL assembly instruction to set or reset FAEN 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).

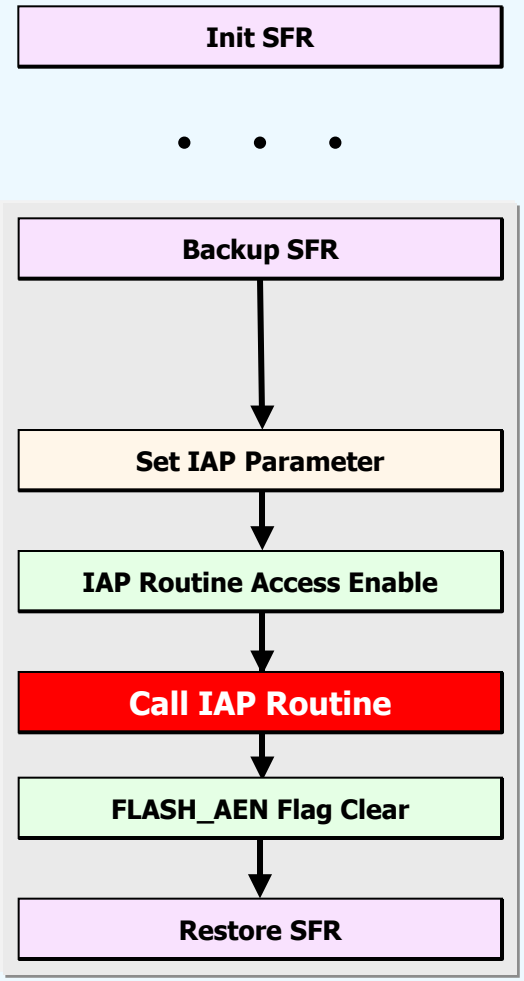
6.19. IAP : Program Flow



[Example Code : IAP Program for FLASH/EEPROM]

```
ORL FAEN, #01h          ; IAP routine access enable
MOV FCNTH, #00h         ; Program/Erase Time Count High
MOV FCNTM, #27h         ; Program/Erase Time Count Mid
MOV FCNTL, #10h         ; Program/Erase Time Count Low
MOV FAEN, #0C1h         ; IAP access enable 1st pattern
MOV FAEN, #01Eh         ; IAP access enable 2nd pattern
MOV FAEN, #0ECh         ; IAP access enable 3rd pattern
MOV FAEN, #081h         ; IAP access enable 4th pattern
                        ; 0x001027 @ RING Freq. == 4MHz

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

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

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

CALL iap_main_program   ; Call IAP routine

MOV FAEN, #000h         ; IAP routine access disable

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

6.19. IAP : Erase Flow



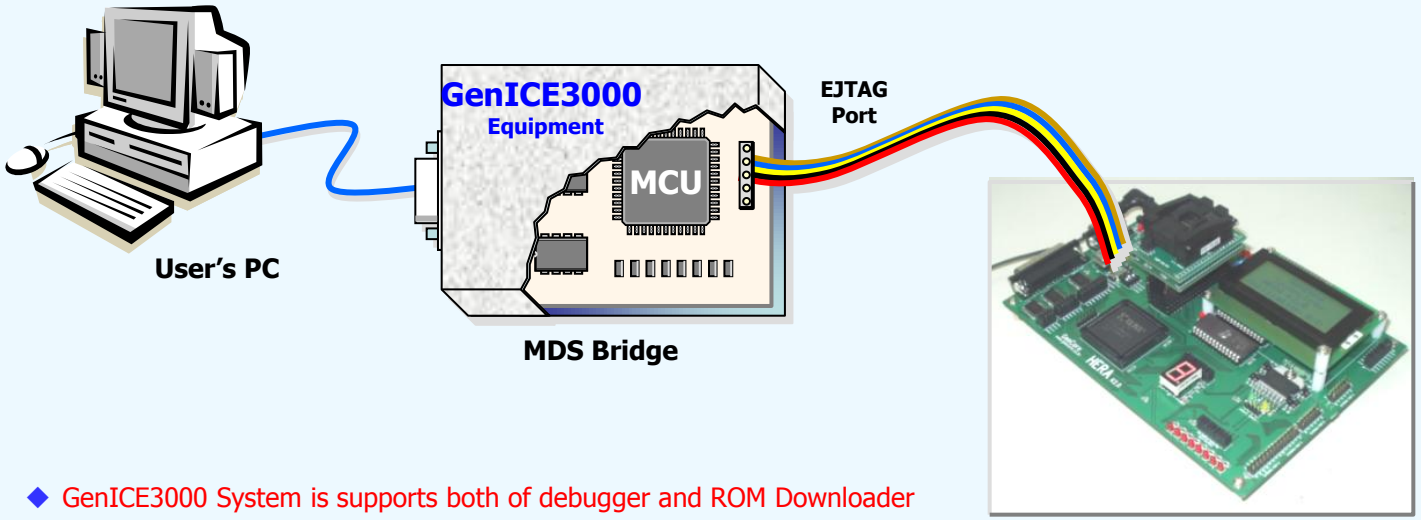
[Example Code : IAP Program for FLASH/EEPROM]

```
ORL FAEN, #01h          ; IAP routine access enable
MOV FCNTH, #00h         ; Program/Erase Time Count High
MOV FCNTM, #27h         ; Program/Erase Time Count Mid
MOV FCNTL, #10h         ; Program/Erase Time Count Low
MOV FAEN, #0C1h         ; IAP access enable 1st pattern
MOV FAEN, #01Eh         ; IAP access enable 2nd pattern
MOV FAEN, #0ECh         ; IAP access enable 3rd pattern
MOV FAEN, #081h         ; IAP access enable 4th pattern
                        ; 0x001027 @ RING Freq. == 4MHz

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

MOV DPTR, #ADDR         ; Erasing Address

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

CALL iap_main_erase     ; Call IAP routine

ANL FAEN, #0FEh         ; IAP routine access disable

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

6.20. ISP & Debugging

- Code memory (32kBytes) can be programmed using EJTAG in target system.
  - ✓ FLASH : 0x0000 ~ 0x7BFF + 0xFC00 ~ 0xFFFF (31KBytes + 1kByte)
- EEPROM (1kByte) can be programmed using EJTAG in target system.
  - ✓ EEPROM : 0xFC00 ~ 0xFFFF (1kByte)
- Debugging using GENICE
- I2C1_SDA pin connection for ISP
- I2C1_SCL pin connection for ISP

[ISP Pin Configuration]



- GenICE3000 System is supports both of debugger and ROM Downloader
- Refer to http://www.coreriver.com → Support → Supporting Tools → 1.Brief System Manual

7. Absolute Maximum Ratings

Items	Conditions	Ranges
Voltage on any pin relative to Ground	-	-0.5V to (V_{DDIO}+0.5V)
Voltage in V_{DDIO} relative to Ground	-	-0.5V to 3.6V
Output Voltage	-	-0.5V to (V_{DDIO}+0.5V)
Output Current High	One I/O pin active	-25mA
	All I/O pin active	-100mA
Output Current Low	One I/O pin active	+30mA
	All I/O pin active	+150mA
Storage Temperature	-	-65 °C to +150 °C
Soldering Temperature	-	260 °C for 10 seconds

8. DC Characteristics

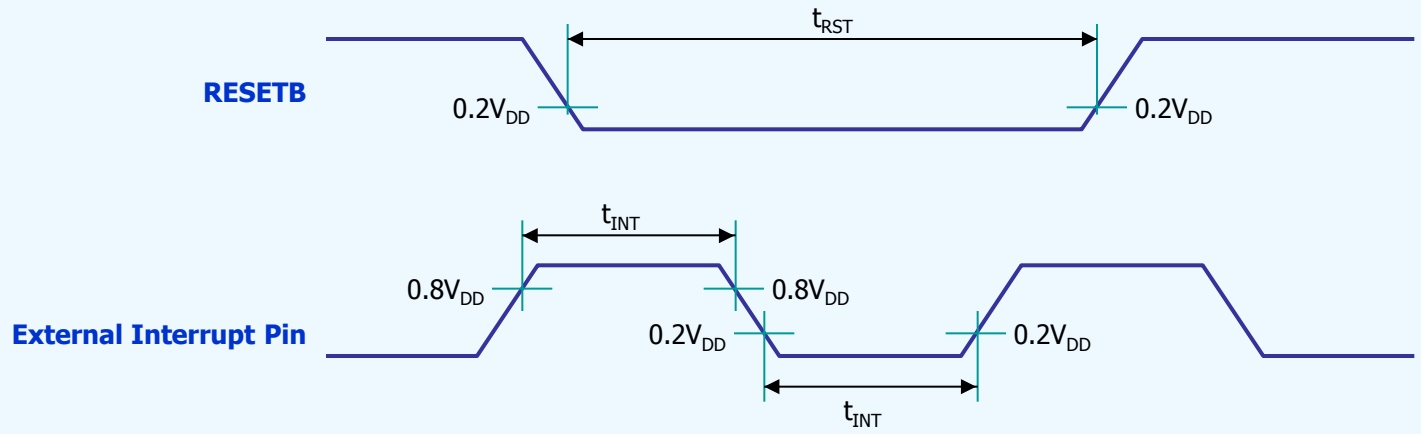
*** TA = -40 °C to +125 °C, V_{DDIO} = +2.7V to +3.6V unless otherwise specified.**

Parameter	Symbol	Pin	Conditions	Value			Unit
				Min.	Typ.	Max.	
Input Low Voltage	V_{IL}	P0, P1, P2, P3[4:3,1:0]	V_{DDIO} = +2.7V to +3.6V	-0.5	-	$0.2V_{DDIO}$-0.1	V
Input high Voltage	V_{IH}	P0, P1, P2, P3[4:3,1:0]	V_{DDIO} = +2.7V to +3.6V	$0.2V_{DDIO}$+1.0	-	V_{DDIO}+0.5	V
Output Low Voltage	V_{OL}	P0, P1, P2, P3[4:3,1:0]	V_{DDIO} = +3.0V to +3.6V (I_{OL} = 4.35mA) V_{DDIO} = +2.7V to +3.0V (I_{OL} = 3.35mA)	-	-	$0.3V_{DDIO}$	V
		P0, P1, P2, P3[4:3,1:0] (High Drive)	V_{DDIO} = +3.0V to +3.6V (I_{OL} = 34.79mA) V_{DDIO} = +2.7V to +3.0V (I_{OL} = 28.41mA)	-	-	$0.3V_{DDIO}$	V
Output High Voltage	V_{OH}	P0, P1, P2, P3[4:3,1:0]	V_{DDIO} = +3.0V to +3.6V (I_{OL} = -8.04mA) V_{DDIO} = +2.7V to +3.0V (I_{OL} = -6.62mA)	$0.7V_{DDIO}$	-	-	V
	V_{OHP}	P3[1:0] (Pull-up Resistor Only)	V_{DDIO} = +3.0V to +3.6V (I_{OL} = -30.30uA) V_{DDIO} = +2.7V to +3.0V (I_{OL} = -24.26uA)	$0.7V_{DDIO}$	-	-	V
Logical 1 to 0 Transition Current	I_{TL}	P0, P1, P2, P3[4:3,1:0]	V_{DDIO} = 3.0V±10% (V_{IN} = +2.0V)	-	-	-650	µA
Input Leakage Current	I_{IL}	P0, P1, P2, P3[4:3,1:0]	V_{IN} = V_{IH} or V_{IL}	-	-	±1	µA
Pin Capacitance	C_{IO}	All	V_{DDIO} = +3.0V	-	10	-	pF

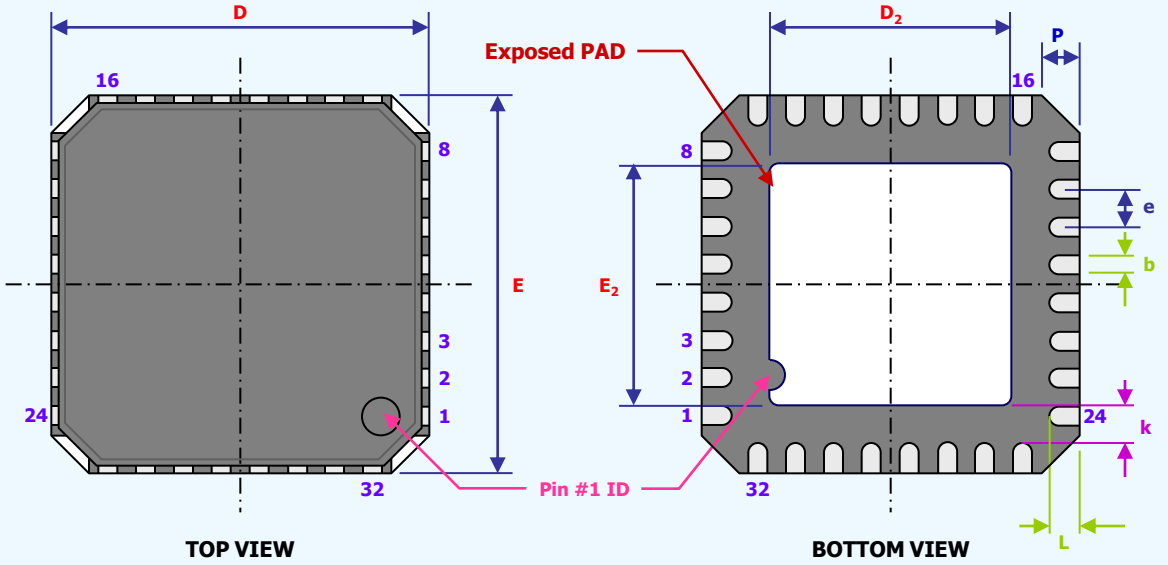
9. AC Characteristics

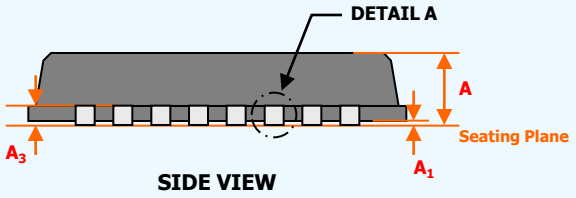
*** TA = -40 °C to +125 °C, V_{DDIO} = +2.7V to +3.6V unless otherwise specified.**

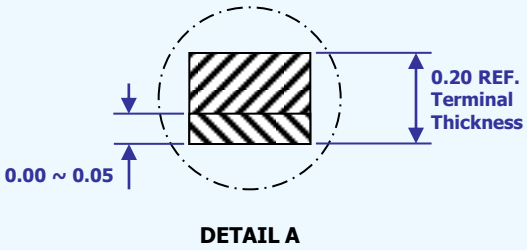
Parameter	Symbol	Pin	Conditions	Value			Unit
				Min.	Typ.	Max.	
RESETB Input Width	t_{RST}	RESETB	V_{DDIO} = 3V ± 10%	24	-	-	F_{SYS}
External Interrupt Input Width	t_{INT}	External Interrupt	V_{DDIO} = 3V ± 10%	4	-	-	F_{SYS}



10. Package Dimensions : 32-MLF







[32-MLF]

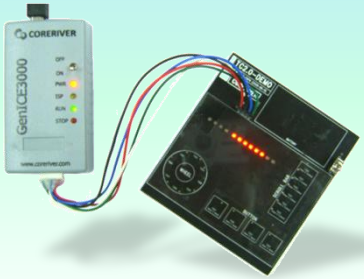
Symbol	Dimensions [mm]		
	Min.	Nom.	Max.
A	0.80	0.85	0.90
A_1	0.00	0.01	0.05
A_3		0.20 REF	
D		5.00 BSC	
E		5.00 BSC	
D_2	2.60	2.70	2.80
E_2	2.60	2.70	2.80
B	0.18	0.23	0.30
e		0.50 BSC	
L	0.30	0.40	0.50
k	0.20	-	-
P	0.24	0.42	0.60

Notes:

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

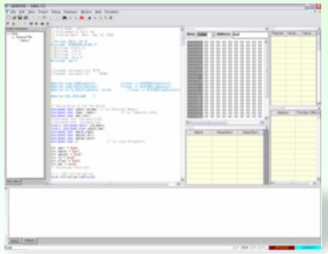
11. Supporting tools

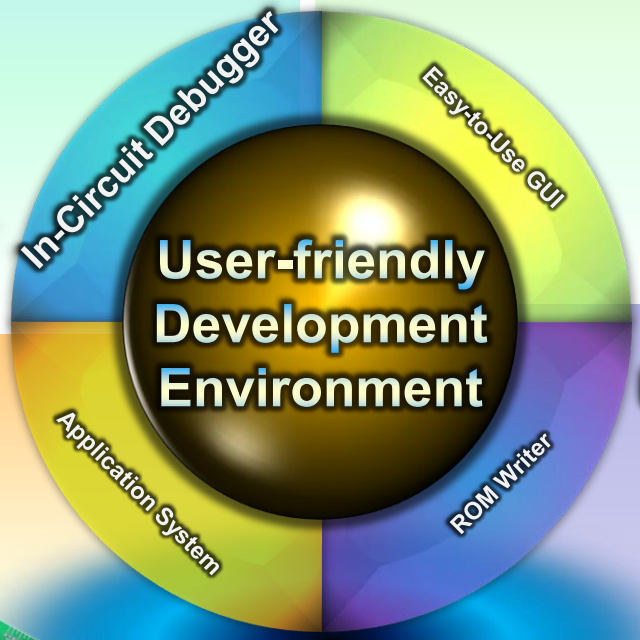
In-Circuit Debugger (GENSYS & GenICE)



Easy-to-Use GUI (GENTOS)

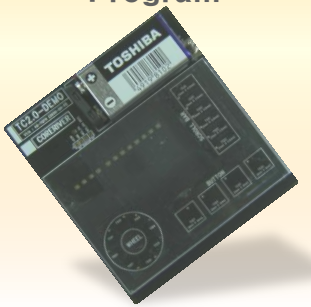
- Assembler & Linker for Windows
- Optimized Cross-C Compiler

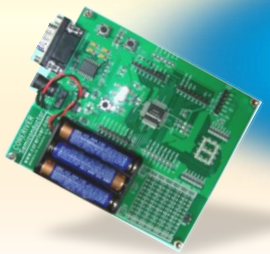




Application System

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





ROM Writer

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

Appendix A : instruction set (1/18)

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

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

1	1	0	1	0	1	0	0

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

Logical AND for byte variables

ANL A, Rn

Operation : (A) ← (A) ^ (Rn)

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

0	1	0	1	1	r	r	r

ANL A, @Ri

Operation : (A) ← (A) ^ ((Ri))

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

0	1	0	1	0	1	1	i

ANL A, direct

Operation : (A) ← (A) ^ (direct)

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

0	1	0	1	0	1	0	1	direct addr

ANL A, #date

Operation : (A) ← (A) ^ data

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

0	1	0	1	0	1	0	0	immediate data

ANL direct, A

Operation : (direct) ← (direct) ^ (A)

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

0	1	0	1	0	0	1	0	direct addr

ANL direct, #data

Operation : (direct) ← (direct) ^ data

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

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

Appendix A : instruction set (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)

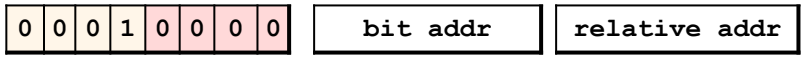
JBC bit, rel

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



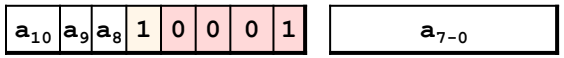
ACALL addr11

Absolute Subroutine Call

Operation :

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

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



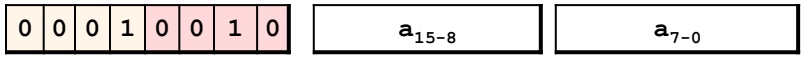
LCALL addr16

Long Subroutine Call

Operation :

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

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



Appendix A : instruction set (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

1	1	0	1	1	r	r	r	relative addr

DJNZ direct, rel

Operation :

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

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

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

NOP

No Operation

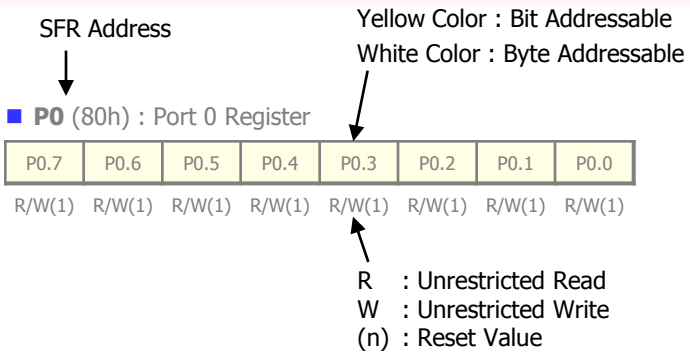
Operation : (PC) ← (PC) + 1

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

0	0	0	0	0	0	0	0

Appendix B : SFR Description [80h ~ 85h] (1/22)

[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)

■ **ALTSEL** (85h) : Port Alternative Function Selection

-	-	I2C_A	PWM_A	UART_A	SPI_A	XTAL_IOEN	RST_IOEN
		R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(1)	R/W(0)

- I2C_A : I2C0/1 port switching selection
  [0] : I2C1 (SDA : P3.0, SCL : P3.1), I2C0 (SDA : P3.3, SCL : P3.4)
  [1] : I2C0 (SDA : P3.0, SCL : P3.1), I2C1 (SDA : P3.3, SCL : P3.4)
- PWM_A : PWM alternative function selection
  [1] : PWM0[7,2,1,0] → PWM1[7:4]A
- UART_A : UART alternative function selection
  [0] : RXD (P3.0), TXD (P3.1)
  [1] : RXD_A (P2.7), TXD_A (P2.6)
- XTAL_IOEN : XTAL IO function enable
- RST_IOEN : RESETB IO function enable

Appendix B : SFR Description [86h ~ 88h] (2/22)

■ CKSEL (86h) : Clock Selection Register

-	-	-	-	-	R32KOE	-	-
					R/W(0)		

- ◆ R32KOE : RING 32kHz Port Output Enable (P3.3)

■ 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] (3/22)

■ **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 12 clocks.
  1: time base is 4 clocks.

Appendix B : SFR Description [8Fh ~ 93h] (4/22)

■ **RINGCON** (8Fh) : RING Control Configuration Register

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

◆ RINGCON[7:0] : Internal RING OSC. Can be tuned.

■ **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

-	IE4	IE3	IE2	XT/RG	RGMD	RGSL	BGS
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(1)	R/W(0)	R/W(1)

◆ IE4~3 : External interrupt 4~3 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] = F_{SYS} / 1 ; Default
[0,0,1] = F_{SYS} / 2
[0,1,0] = F_{SYS} / 4
[0,1,1] = F_{SYS} / 8
[1,0,0] = F_{SYS} / 16
[1,0,1] = F_{SYS} / 32
[1,1,0] = F_{SYS} / 64
[1,1,1] = F_{SYS} / 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_{SYS}/4)
[0,1] : Mode1, 8-bit UART (Variable)
[1,0] : Mode2, 9-bit UART (F_{SYS}/32 or F_{SYS}/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

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.

■ **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)

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

ESPI	EI2C1	EI2C0	EWDT	-	EX4	EX3	EX2
R/W(0)	R/W(0)	R/W(0)	R/W(0)		R/W(0)	R/W(0)	R/W(0)

- ESPI : SPI Interrupt Enable
- EI2C1 : I2C1 Interrupt Enable
- EI2C0 : I2C0 Interrupt Enable
- EWDT : Watchdog Timer Interrupt Enable
- EX4 : External interrupt 4 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] = F_{SYS} / 1 ; Default
[0,0,1] = F_{SYS} / 2
[0,1,0] = F_{SYS} / 4
[0,1,1] = F_{SYS} / 8
[1,0,0] = F_{SYS} / 16
[1,0,1] = F_{SYS} / 32
[1,1,0] = F_{SYS} / 64
[1,1,1] = F_{SYS} / 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	-	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)

- EA : Enable/Disable all interrupts.
- 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

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

- OE[3:0] : PWM1 module 0~3 output enable.

■ **P0HD** (ACh) : Port 0 High Current Driving Register

P0HD7	P0HD6	P0HD5	P0HD4	P0HD3	P0HD2	P0HD1	-
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	

- 0 = High Current Driving OFF (Default) / 1 = High Current Driving ON

■ **P1HD** (ADh) : Port 1 High Current Driving Register

P1HD7	P1HD6	P1HD15	P1HD4	P1HD13	P1HD2	P1HD1	P1HD0
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 = High Current Driving OFF (Default) / 1 = High Current Driving ON

■ **P2HD** (AEh) : Port 2 High Current Driving Register

P2HD7	P2HD6	P2HD5	P2HD4	P2HD3	P2HD2	P2HD1	P2HD0
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 = High Current Driving OFF (Default) / 1 = High Current Driving ON

■ **P3HD** (AFh) : Port 3 High Current Driving Register

-	-	-	-	-	P3HD2	P3HD1	P3HD0
					R/W(0)	R/W(0)	R/W(0)

- 0 = High Current Driving OFF (Default) / 1 = High Current Driving ON

■ **P3** (B0h) : Port 3 Register

-	-	-	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)

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

PSPI	PI2C1	PI2C0	PWDT	-	PX4	PX3	PX2
R/W(0)	R/W(0)	R/W(0)	R/W(0)		R/W(0)	R/W(0)	R/W(0)

- ◆ PSPI : SPI interrupt priority bit.
- ◆ PI2C1 : I2C1 interrupt priority bit.
- ◆ PI2C0 : I2C0 interrupt priority bit.
- ◆ PWDT : Watchdog timer interrupt priority bit.
- ◆ PX4 : External interrupt 4 priority bit.
- ◆ PX3 : External interrupt 3 priority bit.
- ◆ PX2 : External interrupt 2 priority bit.

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

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

- ◆ 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

■ **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

■ **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] : F_{SYS} / 2 [0,0,1] : F_{SYS} / 4
  [0,1,0] : F_{SYS} / 8 [0,1,1] : F_{SYS} / 16
  [1,0,0] : F_{SYS} / 32 [1,0,1] : F_{SYS} / 64
  [1,1,0] : F_{SYS} / 128 [1,1,1] : F_{SYS} / 256

Appendix B : SFR Description [B6h ~ BAh] (10/22)

■ **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

-	-	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
R(1)		R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- 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

-	-	PT2	PS	PT1	PX1	PT0	PX0
R(1)		R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- 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)

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

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

- 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	-	-	-
				R/W(1)			

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

■ **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(1)

- ◆ 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} : 96MHz)
  [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
  [1,1,1] = Not supported

■ OSC2ICN (C7h) : Internal Ring2 Oscillator Control Register

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

- ◆ RING2ON : 1 = Internal Ring2 Oscillator is running.
  0 = Internal Ring2 Oscillator is killed.
  (This bit must be value 0 at any operation.)

■ 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

-	-	LINBG	LINBD	-	-	-	DCEN
		R/W(0)	R/W(0)				R/W(0)

- ◆ LINBG : LIN Baud Rate Generation Enable
- ◆ LINBD : LIN Baud Rate Detection Enable, Cleared by H/W.
- ◆ 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)

■ **PCLKEN** (CEh) : Peripheral Clock Control Register

PCLKEN7	PCLKEN6	PCLKEN5	PCLKEN4	PCLKEN3	PCLKEN2	PCLKEN1	PCLKEN0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

- PCLKEN0 : IAP Clock Enable
- PCLKEN1 : Timer0/1 Clock Enable
- PCLKEN2 : Timer2 Clock Enable
- PCLKEN3 : UART Clock Enable
- PCLKEN4 : SPI Clock Enable
- PCLKEN0 : I2C0 Clock Enable
- PCLKEN0 : I2C1 Clock Enable
- PCLKEN0 : PWM Clock Enable

Appendix B : SFR Description [D0h ~ D4h] (14/19)

■ **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(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)

■ **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(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)

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

-	-	-	P3SEL.4	P3SEL.3	P3SEL.2	P3SEL.1	P3SEL.0
			R/W(1)	R/W(1)	R/W(0)	R/W(0)	R/W(0)

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

Appendix B : SFR Description [D4h ~ D8h] (15/22)

■ **P0INEN** (D5h) : Port 0 Input Enable Register

P0INEN.7	P0INEN.6	P0INEN.5	P0INEN.4	P0INEN.3	P0INEN.2	P0INEN.1	P0INEN.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 = Input Disable / 1 = Input Enable (Default)

■ **PCLK2EN** (D6h) :Peripheral Clock2 Control Register

-	-	PCLK2EN5	PCLK2EN4	PCLK2EN3	PCLK2EN2	PCLK2EN1	-
		R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	

- PCLK2EN1 : WDT Clock Enable
- PCLK2EN2 : WDT Clock Enable ([0] : Disabled in Stop Mode)
- PCLK2EN3 : ST Clock Enable
- PCLK2EN4 : U2C0 Self-running H/W Clock Enable
- PCLK2EN5 : Test Clock Enable

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

WDMOD	POR	-	-	WDIF	WTRF	EWT	RWT
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.
- WDIF : Watchdog timer interrupt flag.
- WTRF : Watchdog timer reset flag.
- EWT : Watchdog timer reset enable.
- RWT : Restart watchdog timer.

■ **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.4	P3TYPE.3	P3TYPE.2	P3TYPE.1	P3TYPE.0
			R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

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

■ **P1INEN** (DDh) : Port 1 Input Enable Register

P1INEN.7	P1INEN.6	P1INEN.5	P1INEN.4	P1INEN.3	P1INEN.2	P1INEN.1	P1INEN.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 = Input Disable / 1 = Input Enable (Default)

Appendix B : SFR Description [E0h ~ E6h] (17/22)

■ **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)

◆ 0 = Output (Default) / 1 = Input

■ **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)

◆ 0 = Output (Default) / 1 = Input

■ **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

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

-	-	-	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)

◆ 0 = Output (Default) / 1 = Input

■ **P2INEN** (E5h) : Port 2 Input/Output Control Register

P2INEN.7	P2INEN.6	P2INEN.5	P2INEN.4	P2INEN.3	P2INEN.2	P2INEN.1	P2INEN.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 = Input Disable / 1 = Input Enable (Default)

■ **I2C0ST** (E7h) : I2C0 Status Register

-	-	-	I2C0TIF	I2C0RIF	I2C0OF	I2C0COL	I2C0BF
			R/W(0)	R/W(0)	R/W(0)	R/W(0)	R(0)

◆ I2C0TIF : I2C0 TX Interrupt Flag in Slave & Master Mode.
[0] : Idle (Default) / [1] : Interrupt occurred
It is set each time a byte is transmitted.
The flag is set by H/W and cleared by S/W.

◆ I2C0RIF : I2C0 RX Interrupt Flag in Slave & Master Mode.
[0] : Idle (Default) / [1] : Interrupt occurred
It is set each time a byte is transmitted.
The flag is set by H/W and cleared by S/W.

◆ I2C0OF : I2C0 Overflow Flag in Write Mode
[0] : Idle (Default) / [1] : Overflow occurred.
It is set when the bytes are received while I2C buffers are still holding the previous bytes.
It is set by H/W and cleared by S/W.

◆ I2C0COL : I2C0 Collision Flag in Read Mode
[0] : Idle (Default) / [1] : Collision occurred.
It is set when the bytes are not transmitted before I2C buffers are updated.
It is set by H/W and cleared by S/W.

◆ I2C0BF : I2C0 Busy Flag in Slave Mode & Master Mode
[0] : RX not complete (Receiver), TX not complete (Transmitter)
[1] : RX complete (Receiver), TX complete (Transmitter)

■ **I2C0CON** (E8h) : I2C0 Control Register

-	-	-	-	I2CRST	MV_BF	I2CIOEN	I2CEN
				R/W(0)	R/W(0)	R/W(0)	R/W(0)

◆ I2CRST : Reset I2C0 Module
(Set by S/W, cleared by H/W automatically)

◆ MV_BF : Move Double Buffer's Data into Output Buffer
[1] : Move the Data
(Set by S/W, cleared by H/W automatically)

◆ I2CIOEN : Enable I2C IO
[0] : Disable I2C IO (Default) / [1] : Enable I2C IO

◆ I2CEN : Enable I2C Module
[0] : Disable I2C Module (Default) / [1] : Enable I2C Module

■ **I2C0RGA** (E9h) : I2C0 Slave Register

RGA1.7	RGA1.6	RGA1.5	RGA1.4	-	-	-	-
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R(0)	R(0)	R(0)	R(0)

◆ RGA[7:4] : I2C Register Address Register
The upper 4-bit Register Address is fixed as RGA[7:4], the lower 4-bit Register Address indicates RX/TX buffer address in the 10-bit mode.

■ **I2C0CFG** (EAh) : I2C0 Configuration Register

TXIEN	RXIEN	ADSEL	ADMIE	CKCFG	I2CDIV2	I2CDIV1	I2CDIV0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

◆ I2CDIV[2:0] : I2C0 Clock Divisor
[000] : F_{SYS} / 2 [001] : F_{SYS} / 4
[010] : F_{SYS} / 8 [011] : F_{SYS} / 16
[100] : F_{SYS} / 32 [101] : F_{SYS} / 64
[110] : F_{SYS} / 128 [111] : F_{SYS} / 256

◆ CKCFG : I2C Clock Configuration
[0] : In the wait state I2C0 clock is set as I2CDIV value.
[1] : In the wait state I2C0 clock is set as F_{SYS} / 256.

◆ ADMIE : SLA Matching Interrupt Enable
[0] : When SLA matched, interrupt occurred.
[1] : Regardless of SLA matching, interrupt occurred.

◆ ADSEL : 7-bit / 10-bit Address Mode Selection in Slave Mode
[0] : 7-bit Mode (Default) / [1] : 10-bit Mode

◆ RXIEN : I2C Data RX Interrupt Enable
[0] : Disable (Default) / [1] : Enable

◆ TXIEN : I2C Data TX Interrupt Enable
[0] : Disable (Default) / [1] : Enable

■ **I2C0SLA** (EBh) : I2C0 Slave Address Register

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

◆ SLA[7:1] : I2C Slave Address Register.

■ **I2C0TDAT** (ECh) : I2C0 TX 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)

◆ DAT[7:0] : I2C0 Double Buffer Data

■ **I2C0TIDX** (EDh) : I2C0 TX Data Index Register

-	-	-	-	INDX3	INDX2	INDX1	INDX0
				R/W(0)	R/W(0)	R/W(0)	R/W(0)

◆ INDX[3:0] : I2C0DATA Buffer Index

■ **I2C0RBF0** (EEh) : I2C0 Data RX Buffer0 Register

RBUF0.7	RBUF0.6	RBUF0.5	RBUF0.4	RBUF0.3	RBUF0.2	RBUF0.1	RBUF0.0
R(0)	R(0)	R(0)	R(0)	R(0)	R(0)	R(0)	R(0)

■ **I2C0RBF1** (EFh) : I2C0 Data RX Buffer1 Register

RBUF1.7	RBUF1.6	RBUF1.5	RBUF1.4	RBUF1.3	RBUF1.2	RBUF1.1	RBUF1.0
R(0)	R(0)	R(0)	R(0)	R(0)	R(0)	R(0)	R(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)

■ **FCNTLD** (F1h) : Flash Erase/Program Time Count Loading

FCNTLD	-	-	-	-	-	-	-
R/W(0)							

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

■ **FCNTL** (F2h) : Flash Erase/Program Time Count Low Byte

FCNT7	FCNT6	FCNT5	FCNT4	FCNT3	FCNT2	FCNT1	FCNT0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

■ **FCNTM** (F3h) : Flash Erase/Program Time Count Middle Byte

FCNT15	FCNT14	FCNT13	FCNT12	FCNT11	FCNT10	FCNT9	FCNT8
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

■ **FCNTH** (F4h) : Flash Erase/Program Time Count High Byte

FCNT23	FCNT22	FCNT21	FCNT20	FCNT19	FCNT18	FCNT17	FCNT15
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

■ **SWRST** (F5h) : ISP Configuration Register

ISP_DIS	SWRST.6	SWRST.5	SWRST.4	SWRST.3	SWRST.2	SWRST.1	SWRST.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 = ISP Mode Disable (Default), 1 = ISP Mode Enable
◆ SWRST[6:0]
0x5A : Software Reset (Chip Reset)
Others : Not Reset

■ **FCON** (F6h) : FLASH Control Register

-	-	-	IFLAG	FFLAG	FMASE	FSERA	FPGM
			R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

◆ IFLAG : Information Access Enable flag.
◆ FFLAG : Flash Access Enable flag.
◆ FMASE : Flash Mass Erase start.
◆ FSERA : Flash Sector Erase start.
◆ FPGM : Flash Program start

■ **FAEN** (F7h) : Flash Access Enable Register

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

◆ MDSF : MDS mode flag.
◆ FAEN : FLASH access enable flag.

Appendix B : SFR Description [F8h ~ F9h] (21/22)

■ I2C1ST (F8h) : I2C1 Status Register

I2C1IF	I2C1OF	I2C1ACK	I2C1RW	I2C1DA	I2C1P	I2C1S	I2C1BF
R/W(0)	R/W(0)	R (0)	R (0)	R (0)	R (0)	R (0)	R (0)

◆ I2C1IF : I2C1 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 I2CCFG SFR is set, it is set at Start/Stop condition.
The flag is set by H/W and cleared by S/W.

◆ I2C1OF : I2C1 Overflow Flag in slave & master mode
[0] : Idle [1] : Overflow occurred.
It is set when a byte is received while I2CDAT SFR is still holding the previous byte.
It is set by H/W and cleared by S/W

◆ I2C1ACK : I2C1 Acknowledge flag in slave & master mode.
[0] : Indicate receiving Acknowledge bit.
[1] : Indicate receiving Not Acknowledge bit.

◆ I2C1RW : I2C1 Read/Write flag in slave mode
[0] : Write state [1] : Read state

◆ I2C1DA : 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

◆ I2C1P : 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.

◆ I2C1S : 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.

◆ I2C1BF : Busy flag in slave & master mode
[0] : RX not complete (Receiver), TX not complete (Transmitter)
[1] : RX complete (Receiver), TX complete (Transmitter)

■ I2C1CON (F9h) : I2C1 Control Register

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

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

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

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

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

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

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

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

Appendix B : SFR Description [FAh ~ FDh] (22/22)

■ **I2C1CFG** (FAh) : I2C1 Configuration Register

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

◆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)

■ **I2C1SLA** (FBh) : I2C1 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),
I2CSLA[7:1] is used for matching address and I2CSLA[0] is masked.
In 10-bit address mode (2nd SLA),
I2CSLA[7:0] is used for matching address.

■ **I2C1DAT** (FCh) : I2C1 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)

■ **I2C1SCL** (FDh) : I2C1 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
FI2C = F_{SYS} / (2 * (MSCL[7:0] + 2))

■ **STCON** (FEh) : Stop Timer Control Register

-	-	-	-	-	-	ST_CLR	ST_EN
						R/W(0)	R/W(0)

◆ ST_CLR : Clear Stop Timer Count & Stop Timer Disable
◆ ST_EN : Stop Timer Enable

■ **STCFG** (FFh) : Stop Timer Configuration Register

-	-	-	ST_BUSY	ST_DR3	ST_DR2	ST_DR1	ST_DR0
			R/W(0)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

◆ ST_BUSY : Stop Timer Busy Flag
◆ ST_DR[3:0] : Stop Timer Time-out Values

Appendix C : Update History

- V1.0
  - ✓ First Release.
- V1.3
  - ✓ Device Spec. modified.
- V1.4
  - ✓ P0SEL, P1SEL, P2SEL inserted.
  - ✓ Slide 71, 117 : PWM0D0 inserted.
- V1.5
  - ✓ F_{OSC} → F_{SYS} for Peripherals (UART, I2C, SPI, PWM, Timer0/1/2, Clock Circuit, AC Characteristics)
- V1.6
  - ✓ P2HD modified at Appendix.
  - ✓ FCNTH modified at Appendix.
  - ✓ OSC2ICN inserted.
  - ✓ PCLKEN inserted at Appendix.
  - ✓ ADCSEL removed.
  - ✓ Slide 79, 114 : P3.4. is changed to P3.3 for clock output pin (R32KOE) at CKSEL.
- V1.7
  - ✓ P0INEN, P1INEN, & P2INEN are modified.
    0 : Input Enable → Input Disable
    1 : Input Disable (Default) → Input Enable (Default)
  - ✓ ALTSEL modified. (Description of I2C_A)
- V1.8
  - ✓ Port Description modified.
  - ✓ ADC function (including ADCDIV) removed.
- V1.9
  - ✓ Pin Configuration modified. (T0, T1, T2, & T2EX inserted)