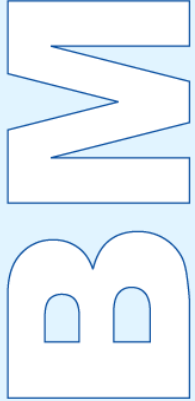




# MiDAS Family

BM-MiDAS1.1-Kor-V2.9



## Brief Manual of MiDAS1.1 Family

### EPROM / ROM based 8-bit Turbo Microcontrollers

V2.9

December 2011

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

# 목차

## 1. 제품 개요

## 2. 특징

## 3. 블록 도표

## 4. Pin 구성

## 5. Pin 설명

## 6. 기능 설명

### ✓ CPU 설명

- Memory 구조
- SFR Map과 설명
- 명령어 Set 요약
- CPU Timing

### ✓ 주변회로 설명

- 입출력 단자
- Pad의 ESD 보호 구조
- LVD (Low Voltage Detector)
- WDT (Watchdog Timer)
- Timer0/1
- UART (Universal Async. RX/TX)
- PWM (Pulse Width Modulator)
- ADC
- 인터럽트
- Reset 회로
- 클럭 회로
- 전력 관리

## 7. 전통적 80C52에 비하여 강점

## 8. 전원 전압 상승 기울기 추천

## 9. 절대 최대 동작 범위 한계

## 10. DC 특성

## 11. AC 특성

## 12. ADC 특성

## 13. Package Dimensions

## 14. Product Numbering System

## 15. 지원 툴

## 16. 부록

### A. 명령어 Set

### B. SFR 설명

### C. Update History

# 1. 제품 개요

- ◆ 코아리버 **MiDAS1.1** 제품군은 고속의 **80C52** 호환 마이크로컨트롤러이다.
- ◆ **MiDAS1.1** 제품군은 전통적인 **80C52**와 비교해서 최대 **3배**의 속도로 명령어를 실행한다.
  - ✓ **1** 기계어 주기 = **4** 클럭 주기 vs. **12** 클럭 주기
- ◆ **MiDAS1.1** 제품군의 추가적인 주변회로들:
  - ✓ **10-bit ADC / 8-bit PWM / WDT / LVD / POR.**
- ◆ 전력 절감 방안들을 갖는다.
- ◆ 잡음 저항성 방안이 있다.
- ◆ **사용자 편리 MDS** 환경을 제공한다.
- ◆ **사용하기 쉬운 training-kit**를 제공한다.
- ◆ 이 간략한 매뉴얼의 내용은 언제든지 갱신될 수 있다. **CORERIVER** 웹 사이트 (<http://www.coreriver.com>)에서 갱신된 내용을 확인해야 한다.

# 1. 제품군

## ◆ MiDAS1.1 제품군-GC80C510A 계열 (저비용 ADC 응용 MCU)

Product	Mask-ROM (byte)	EPROM (byte)	RAM (Byte)	Volt (V)	Freq. (MHz)	T/C (16bits)	Serial I/O	WDT	ADC (bit x ch)	PWM (bit x ch)	I/O Pins	Package	Others	Available Time
GC87C510A0-SP20I GC87C510A0-SO20I GC87C510A0-SO16I GC87C510A0-TS16I GC87C510A0-SO8I	-	4K	128	2.4~5.5	20 (10)	2	1 UART	1	10x12 10X12 10X8 10X8 10X2	8x1	18 18 14 14 6	20-SPDIP 20-SOP 16-SOP 16-TSSOP 8-SOP	LVD POR Ring OSC	<b>Now</b>
<a href="#">GC87C510A1-SO8I</a>	-	4K	128	2.4~5.5	20 (10)	2	1 UART	1	10X3	8x1	6	8-SOP	LVD POR Ring OSC	<b>Now</b>
GC81C510A0-SP20I GC81C510A0-SO20I GC81C510A0-SO16I GC81C510A0-TS16I GC81C510A0-SO8I	4K	-	128	2.4~5.5	20 (10)	2	1 UART	1	10x12 10X12 10X8 10X8 10X2	8x1	18 18 14 14 6	20-SPDIP 20-SOP 16-SOP 16-TSSOP 8-SOIC	LVD POR Ring OSC	<b>Now</b>
<a href="#">GC81C510A1-SO8I</a>	4K	-	128	2.4~5.5	20 (10)	2	1 UART	1	10X3	8x1	6	8-SOIC	LVD POR Ring OSC	<b>Now</b>

\* MiDAS1.1 제품군의 동작주파수는 5.0V로 동작할 때 20 MHz 이다.

# 1. 제품군 (계속)

## ◆ MiDAS1.1 제품군-GC80C510A 계열 (저비용 ADC 응용 MCU)

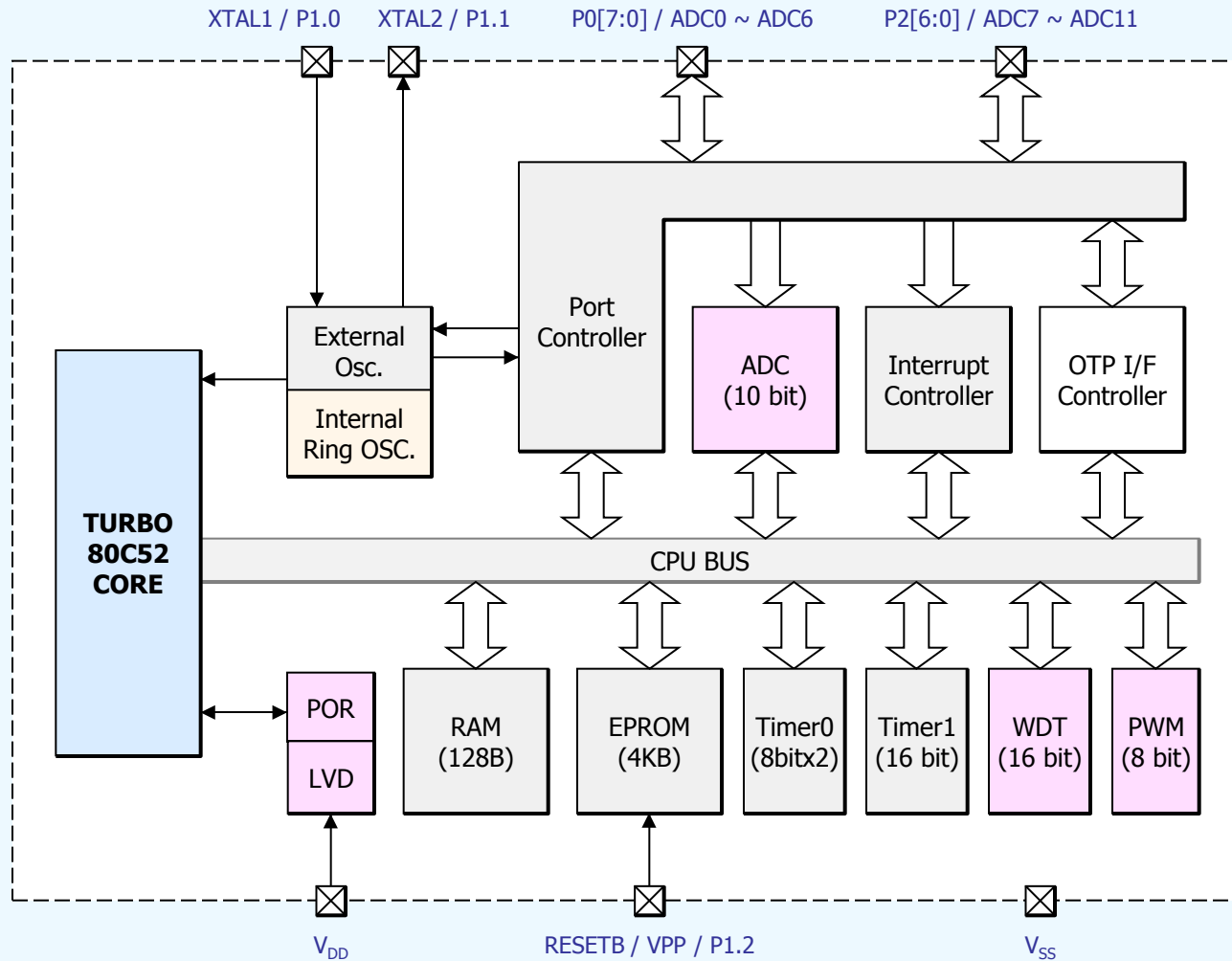
Product	Mask-ROM (byte)	EPROM (byte)	RAM (Byte)	Volt (V)	Freq. (MHz)	T/C (16bits)	Serial I/O	WDT	ADC (bit x ch)	PWM (bit x ch)	I/O Pins	Package	Others	Available Time
GC81C500A0-SP20I GC81C500A0-SO20I GC81C500A0-SO16I GC81C500A0-TS16I GC81C500A0-SO8I	2K	-	128	2.4~5.5	20 (10)	2	1 UART	1	10x12 10X12 10X8 10X8 10X2	8x1	18 18 14 14 6	20-SPDIP 20-SOP 16-SOP 16-TSSOP 8-SOIC	LVD POR Ring OSC	<b>Now</b>
<a href="#">GC81C500A1-SO8I</a>	2K	-	128	2.4~5.5	20 (10)	2	1 UART	1	10X3	8x1	6	8-SOIC	LVD POR Ring OSC	<b>Now</b>

\* Operating frequency of MiDAS1.1 family is 20 MHz at 5.0 voltage.

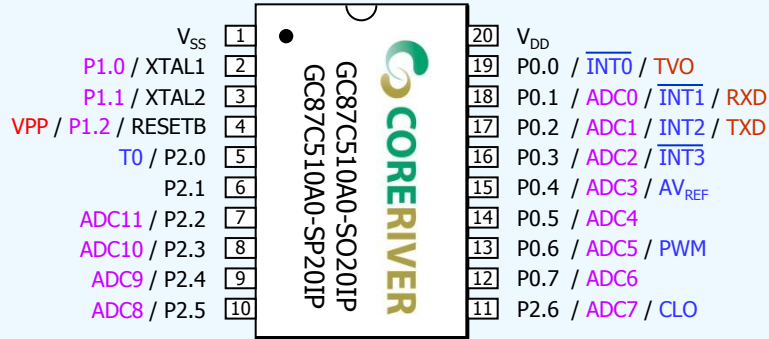
## 2. 특징

- ◆ CPU
  - ✓ 8-bit turbo 80C52 구조
  - ✓ 4 클럭 주기/1 기계어
  - ✓ 인텔 80C52와 명령어 수준 호환
- ◆ 4Kbytes EPROM
- ◆ 128bytes RAM
- ◆ 공급 전압: 2.4V ~ 5.5V
- ◆ 동작 주파수
  - ✓ 최대 20MHz @4.5V ~ 5.5V
  - ✓ 최대 10MHz @2.4V ~ 3.3V
- ◆ -40 °C 에서 120 °C 까지 동작
- ◆ 프로그램이 가능한 18개의 입출력 pins
  - ✓ 소프트웨어에 의해 pull-up 제어
  - ✓ Push-pull 출력
- ◆ 저전압 검출기(Low Voltage Detector)
- ◆ 내부 링 오실레이터: 3.6MHz@5V (±15%)
- ◆ 16-bit 프로그램 가능 Watchdog Timer
- ◆ 2개의 16-bit 타이머/카운터
- ◆ 전이중 UART
- ◆ 1-채널 8-bit 고속 PWM
- ◆ 12-채널 10-bit ADC
  - ✓ 최대 100KSPS (@Fadc = 8 MHz)
  - ✓ 입력 클럭 주파수를 프로그램 가능
- ◆ 10개의 인터럽트 발생원
  - ✓ Timer0/1, UART, ADC, PWM, WDT, 외부 4개
  - ✓ 2단계 인터럽트 우선순위
- ◆ Reset 방안
  - ✓ On-chip Power-On-Reset
  - ✓ 외부 reset
  - ✓ 저전압 검출기 reset
  - ✓ Watchdog timer reset
- ◆ 전력 소모
  - ✓ Active current : 최대 10mA @5V, 20MHz
  - ✓ Stop current : 최대 1uA
- ◆ E.S.D.는 2,000V까지 안전함
- ◆ Latch-up은 ±200mA까지 안전함
- ◆ Package
  - ✓ 20-PDIP, 20-SOP
  - ✓ 16-TSSOP, 16-SOP
  - ✓ 8-SOP

### 3. 블록도표

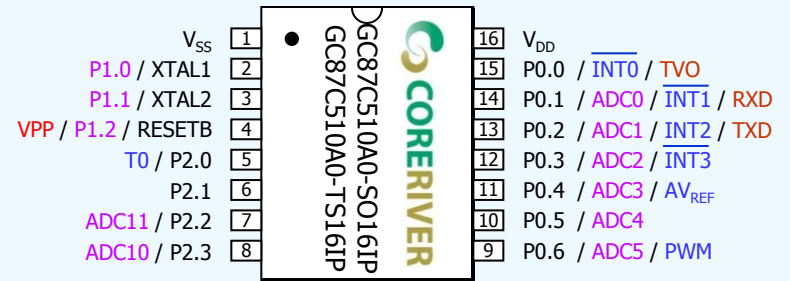


# 4. Pin 구조



[ 20-SOP / 20-PDIP ]

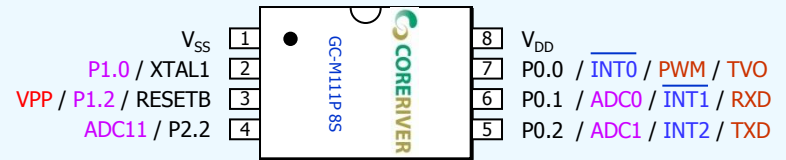
- ◆ 20-SOP : Wide
- ◆ 16/8-SOP : Narrow



[ 16-SOP / 16-TSSOP ]



[ 8-SOP (A0 version) ]

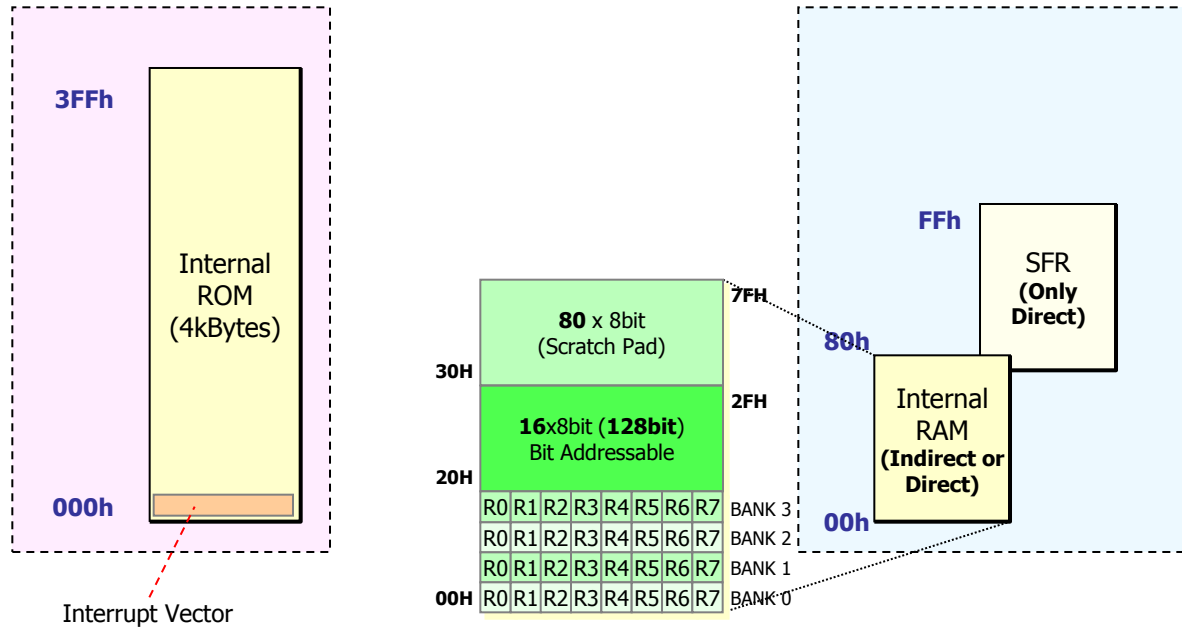


[ 8-SOP (A1 version) ]

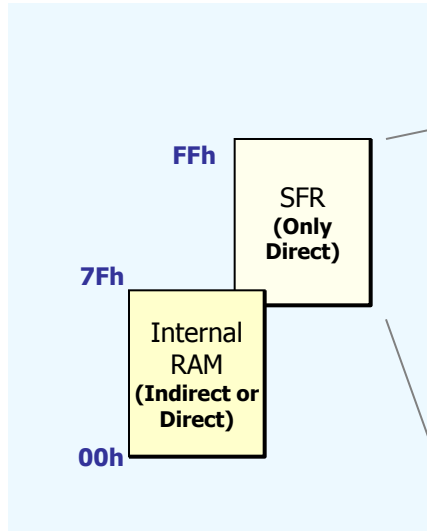
## 5. Pin 설명

기호	방향	설명	공유 Pins
$V_{DD}$	입력	전력 공급원	-
$V_{SS}$	입력	전력용 접지	-
RESETB / VPP / P1.2	입출력	<ul style="list-style-type: none"> <li>외부 Reset 입력 신호(기본지정)</li> <li>비트 단위 프로그램 가능</li> </ul>	VPP (11.5V)
XTAL1 / P1.0	입출력	<ul style="list-style-type: none"> <li>크리스탈 입출력(기본지정)</li> <li>Schmitt Trigger 입력과 비트 단위 프로그램 가능                             <ul style="list-style-type: none"> <li>- Pull-up On/OFF 제어 가능</li> <li>- Push-pull 출력</li> </ul> </li> </ul>	크리스탈 입력
XTAL2 / P1.1			크리스탈 출력 (오직 A0 종류의 경우)
P0[7:0]	입출력	<ul style="list-style-type: none"> <li>Schmitt Trigger 입력과 비트 단위 프로그램 가능                             <ul style="list-style-type: none"> <li>- Pull-up On/OFF 제어 가능</li> <li>- Push-pull 출력 (기본지정)</li> </ul> </li> </ul>	<u>RX</u> , <u>TX</u> , <u>TVO</u> INTO ~ INT3, ADC0 ~ ADC6, PWM AV <sub>REF</sub>
P2[6:0]	입출력	<ul style="list-style-type: none"> <li>Schmitt Trigger 입력과 비트 단위 프로그램 가능                             <ul style="list-style-type: none"> <li>- Pull-up On/OFF 제어 가능</li> <li>- Push-pull 출력 (기본지정)</li> </ul> </li> </ul>	ADC7 ~ ADC11 CLO T0

# 6.1. Memory 구조



## 6.2. SFR (Special Function Register) Map



Bit 주소지정 가능

: MiDAS1.1 제품군에 새로 추가된 SFR

: Reserved for future use.

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

## 6.2. SFR에 대한 간략한 설명

### ◆ 80C52 SFR 레지스터

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

### ◆ MiDAS1.1에 새로 추가된 SFR 레지스터

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

\* : 사용하지 않는 비트.

## 6.3. 명령어 Set 요약

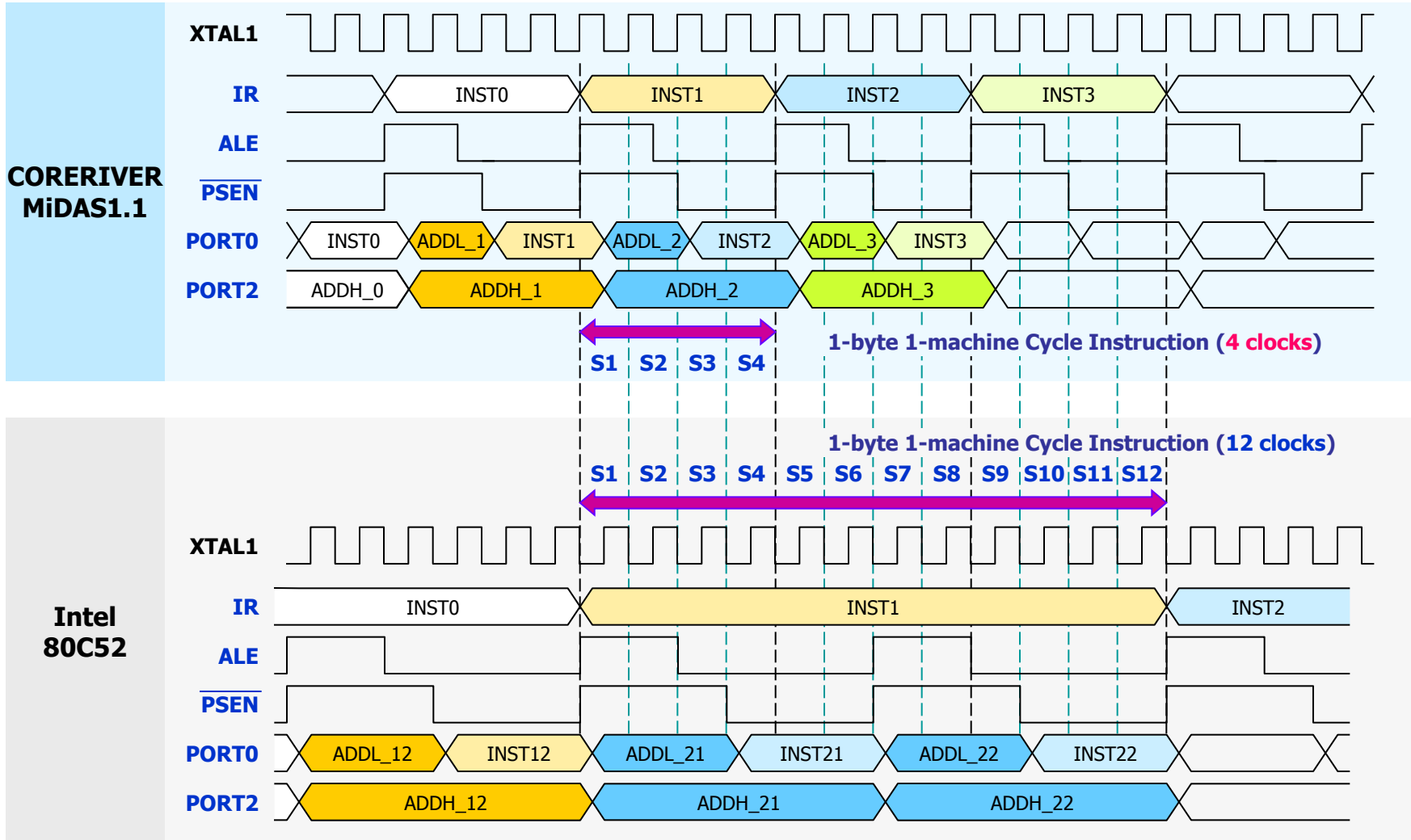
- ◆ 더 자세한 것들은 부록 A (명령어 Set)를 참조하라.

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
	PUSH	PUSH
	POP	POP
	XCH	Exchange
	XCHD	Exchange Low-digit

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

# 6.4. CPU Timing

◆ Comparative timing of the MiDAS1.1 family and Intel 80C52



## 6.4. CPU Timing : 수행 시간표

- ◆ 세계에서 가장 빠른 명령어 수행

Instruction	MIDAS1.1 (CORERIVER)	W77C32 (Winbond)	DS80C320 (Maxim)	87C52 (Intel)
MUL AB DIV AB	<b>12 clocks</b>	20 clocks	20 clocks	48 clocks
MOVC A, @A+PC MOVC A, @A+DPTR	<b>8 clocks</b>	8 clocks	12 clocks	24 clocks
JMP @A+DPTR	<b>8 clocks</b>	8 clocks	12 clocks	24 clocks
RET RETI	<b>8 clocks</b>	8 clocks	16 clocks	24 clocks
INC DPTR	<b>4 clocks</b>	8 clocks	12 clocks	24 clocks
Others	<b>Same</b>	Same	Same	-



## 6.5. 입출력 단자 : PORT1[1:0] (XTAL1/XTAL2)

- ◆ XTAL1/XTAL2은 입출력 단자로 사용될 수 있다.
- ◆ 읽기-수정-쓰지 명령어는 단자 pin이 아니라 SFR 레지스터를 읽는다.
  - ✓ ANL / ORL / XRL / JBC / CPL / INC / DEC / DJNZ / MOV PX.Y, C / CLR PX.Y / SETB PX.Y
- ◆ P1는 Asynchronous POR (Power-on-reset)에 의해 전압 인가와 동시에 Register가 초기화 된다.

✓ **P1TYPE** (D5h) : Port 1 출력방식 제어 레지스터

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

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

- 0 = Push-pull 출력 (기본지정)

✓ **P1DIR** (F5h) : Port 1 입출력 제어 레지스터

-	-	-	-	-	-	P1DIR.2	P1DIR.1	P1DIR.0
---	---	---	---	---	---	---------	---------	---------

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

- 0 = 출력 / 1 = 입력 (기본지정)

✓ **P1SEL** (E5h) : Port 1 Pull-up 제어 레지스터

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

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

- 0 = Pull-up 저항 ON
- 1 = Pull-up 저항 OFF (기본지정)

✓ **P1** (90h) : Port 1 레지스터

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

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

✓ **ALTSEL** (E3h) : Alternate 기능 제어 레지스터

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

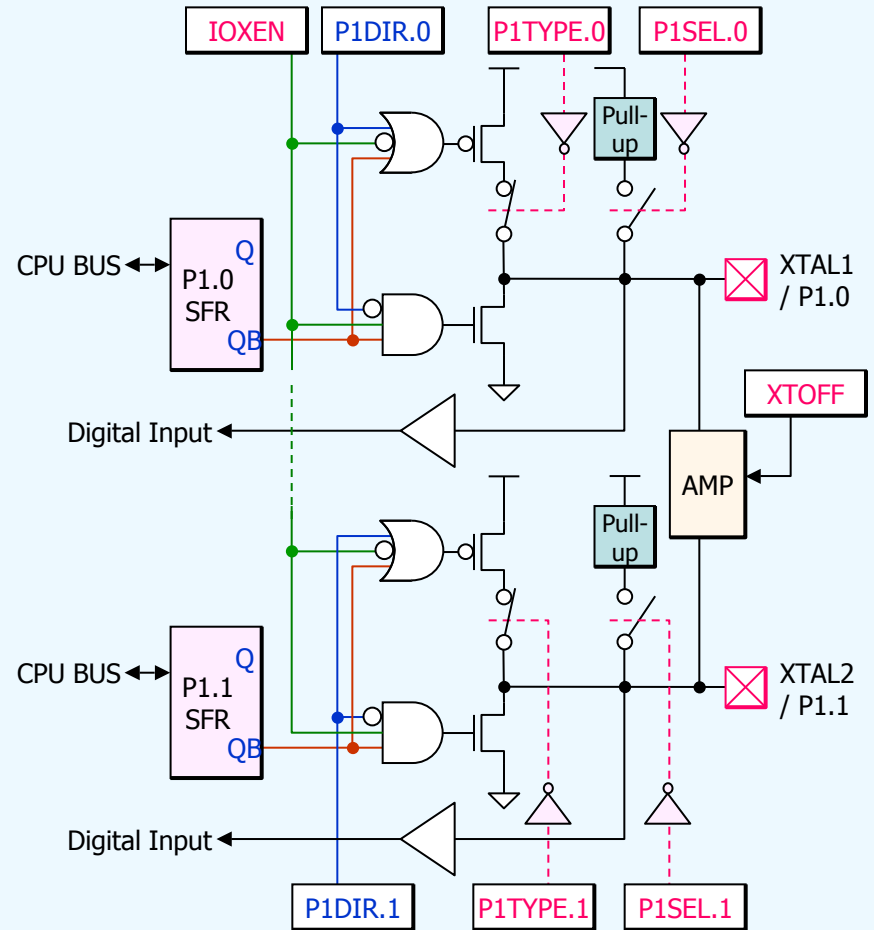
R/W(0) R/W(0) R/W(0) R/W(0) R/W(0) R/W(0)

- IOXEN = 1 : XTAL1 과 XTAL2은 입출력 단자로 사용된다.

✓ **PMR** (C4h) : 전력 관리 제어 레지스터

-	-	-	-	XTOFF	-	-	-
---	---	---	---	-------	---	---	---

R/W(0)



## 6.5. 입출력 단자: PORT1[2] (RESETB)

- ◆ RESETB은 입출력 단자로 사용될 수 있다.
- ◆ 읽기-수정-쓰기 명령어는 단자 pin이 아니라 SFR 레지스터를 읽는다.
  - ✓ ANL / ORL / XRL / JBC / CPL / INC / DEC / DJNZ / MOV PX.Y, C / CLR PX.Y / SETB PX.Y
- ◆ P1는 Asynchronous POR (Power-on-reset)에 의해 전압 인가와 동시에 Register가 초기화 된다.

✓ P1DIR (F5h) : Port 1 입출력 제어 레지스터

-	-	-	-	-	P1DIR.2	P1DIR.1	P1DIR.0
---	---	---	---	---	---------	---------	---------

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

- 0 = 출력 / 1 = 입력 (기본지정)

✓ P1 (90h) : Port 1 레지스터

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

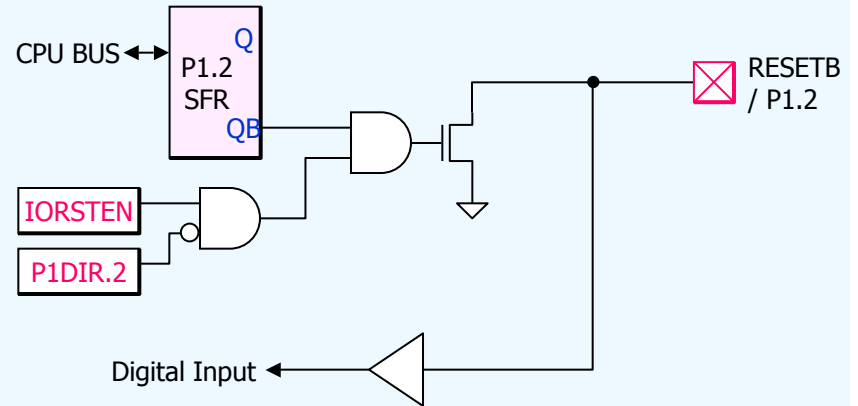
R/W(1) R/W(1) R/W(1)

✓ ALTSEL (E3h) : Alternate 기능 제어 레지스터

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

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

- IORSTEN = 1 : RESETB은 입출력 단자로 사용된다.



## 6.5. 입출력 단자 : PORT2[6:0]

- ◆ 기본지정으로 Pull-up은 ON되고 push-pull 출력 상태가 된다.
- ◆ P2[6:2]은 ADC11 ~ ADC7 입력으로 사용될 수 있다.
- ◆ 읽기-수정-쓰기 명령어는 단자 pin이 아니라 SFR 레지스터를 읽는다.
  - ✓ ANL / ORL / XRL / JBC / CPL / INC / DEC / DJNZ / MOV PX.Y, C / CLR PX.Y / SETB PX.Y
- ◆ P2는 전원이 인가된 후 **synchronous internal reset** 에 의해 초기화 되기 전까지 **Register** 상태가 정의되어 있지 않다. **만일 전원 인가 직후 Port의 상태가 일정해야 한다면 P1을 이용하십시오.**

✓ **ADCSELH (E1h) : ADC 채널 선택 상위 레지스터**

ADC11B	ADC10B	ADC9B	ADC8B	ADC7B	ADC6B	ADC5B	ADC4B
--------	--------	-------	-------	-------	-------	-------	-------

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

- ADCXB = 0 : ADCX 입력 인에이블 & 디지털 입력 디세이블

✓ **P2TYPE (D6h) : Port 2 출력방식 제어 레지스터**

-	P2TYPE.6	P2TYPE.5	P2TYPE.4	P2TYPE.3	P2TYPE.2	P2TYPE.1	P2TYPE.0
---	----------	----------	----------	----------	----------	----------	----------

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

- 0 = Push-pull 출력(기본지정)

✓ **P2DIR (F6h) : Port 2 입출력 제어 레지스터**

-	P2DIR.6	P2DIR.5	P2DIR.4	P2DIR.3	P2DIR.2	P2DIR.1	P2DIR.0
---	---------	---------	---------	---------	---------	---------	---------

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

- 0 = 출력 / 1 = 입력 (기본지정)

✓ **P2SEL (E6h) : Port 2 Pull-up 제어 레지스터**

-	P2SEL.6	P2SEL.5	P2SEL.4	P2SEL.3	P2SEL.2	P2SEL.1	P2SEL.0
---	---------	---------	---------	---------	---------	---------	---------

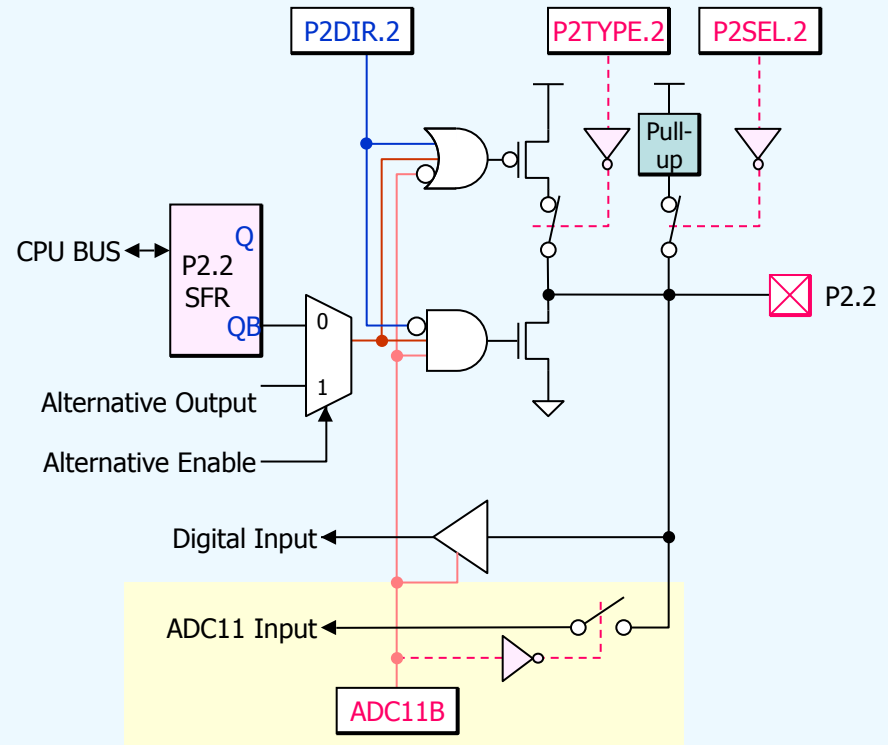
R/W(0) R/W(0) R/W(0) R/W(0) R/W(0) R/W(0) R/W(0) R/W(0)

- 0 = Pull-up 저항ON
- 1 = ADC\_EN(ADCON[3]) = 1일 때, pull-up 저항 OFF.

✓ **P2 (A0h) : Port 2 레지스터**

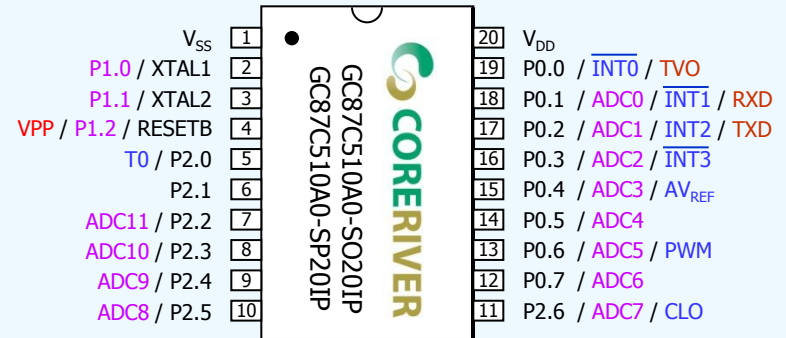
-	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
---	------	------	------	------	------	------	------

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. 입출력 단자 : Feedback Pull-up Issue

- ◆ 내부 pull-up 저항들은 PnSEL SFR의 값을 변경함에 따라 ON/OFF된다.  
ADC 기능으로도 사용하는 단자는 ADC\_EN를 설정해야 하며, 그러면 내부 pull-up 저항이 Off된다.
- ◆ XTAL1(P1.0), XTAL2(P1.1) 단자는 P1SEL SFR의 값에 따라 pull-up 저항이 on/off된다.
- ◆ 예제
  - ✓ XTAL1(P1.0), XTAL2(P1.1) 단자의 pull-up 저항 off  
P1SEL |= 0x03;  
(0 = Pull-up 저항 ON (기본지정) / 1 = Pull-up 저항 (OFF))
  - ✓ P0.1 단자 Pull-up 저항 off  
P0SEL |= 0x02;  
(0 = Pull-up 저항 ON (기본지정) / 1 = Pull-up 저항 (OFF))  
ADCSEL &= 0xEF;  
(ADC0B = 0, Port 0.1의 내부 pull up 저항 Off)



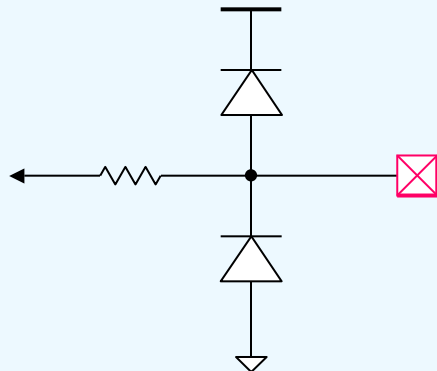
[ 20-SOIC / 20-SPDIP ]

- ◆ 20-SOIC : Wide
- ◆ 16/8-SOIC : Narrow

## 6.6. PAD의 ESD 보호 구조

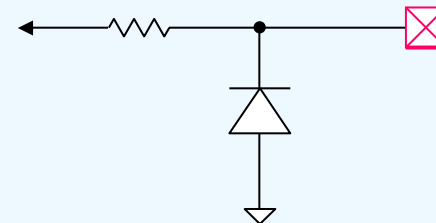
- ◆ 두 개의 ESD diode와 하나의 ESD 저항이 VPP/P1.2/RESETB를 제외한 모든 PAD에 포함되어 있다.
- ◆ 하나의 ESD diode와 하나의 ESD 저항이 VPP/P1.2/RESETB에 포함되어 있다.

[VPP/P1.2/RESETB를 제외한 모든 PAD]



- 두 개의 ESD Diode ( $V_{DD}$  쪽, GND 쪽)
- 하나의 ESD 저항

[VPP/P1.2/RESETB]



- 하나의 ESD Diode (GND 쪽)
- 하나의 ESD 저항

## 6.7. LVD (Low Voltage Detector)

- ◆ On-chip power-on reset : 2.3V
- ◆ On-chip 전원 오류 reset : 2.3V

✓ **EXIF** (91h) : 외부 인터럽트 확장 플래그

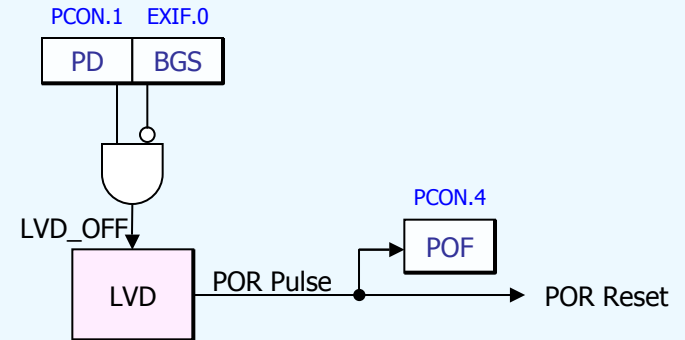
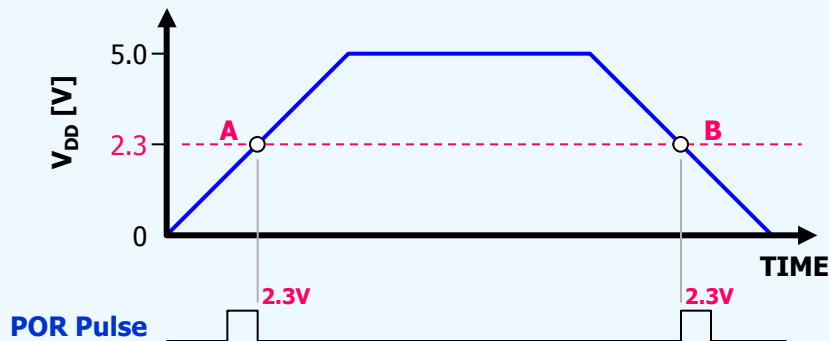
-	-	IE3	IE2	XT/RG	RGMD	RGSL	BGS
R/W(0)	R/W(0)	R/W(0)	R(1)	R/W(0)	R/W(1)		

- BGS : Band-gap 선택  
0 = 정지 모드에서 Band-gap block (LVD) 정지, 그러나 정상 모드에서 동작.  
1 = 정지 모드에서 Band-gap block (LVD) 동작.

✓ **PCON** (87h) : 전력 제어 레지스터

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

- POF : 전원 꺼짐 플래그. 전원이 켜지면, 하드웨어에 의해 POF가 1이 됨.
- PD : 정지 모드 비트



## 6.8. WDT (Watchdog Timer)

- ◆ 외부 잡음이나 다른 원인으로 인하여 프로그램이 오동작하는 것을 검출한다.
- ◆ WDT 인터럽트를 사용하여 동작을 정상 상태로 돌린다.
- ◆ 인에이블이 되어 있으면 WDT 인터럽트 또는 WDT reset이 정지모드 2에서 MCU를 깨운다.
- ◆ Watchdog Time-out 값

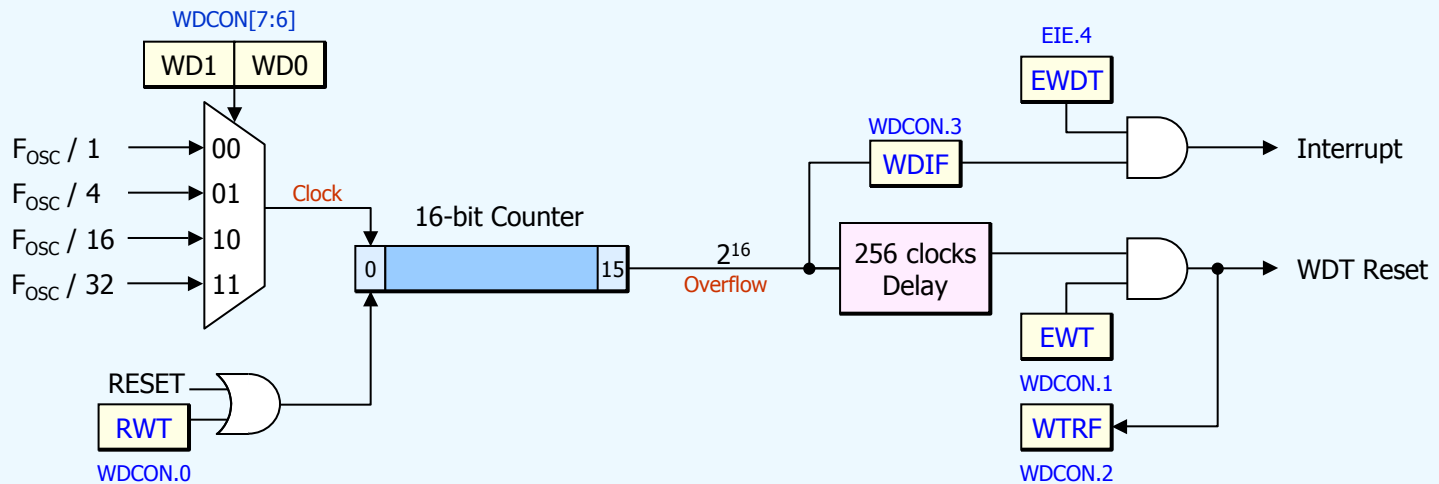
WD1	WD0	Interrupt Time-out (@4MHz)		Reset Time-out (@4MHz)	
0	0	1x2 <sup>16</sup> clocks	16.38 ms	1x2 <sup>16</sup> + 256 clocks	16.45 ms
0	1	4x2 <sup>16</sup> clocks	65.54 ms	4x2 <sup>16</sup> + 256 clocks	65.60 ms
1	0	16x2 <sup>16</sup> clocks	262.14 ms	16x2 <sup>16</sup> + 256 clocks	262.21 ms
1	1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

✓ **WDCON (D8h) : Watchdog Timer 제어 레지스터**

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

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

- WD[1:0] : WDT 클럭 분주(1/4/8/32)
- WDIF : Watchdog Timer 인터럽트 플래그
- WTRF : Watchdog Timer Reset 플래그.  
오직 소프트웨어에 의해 소거된다.
- EWT : Watchdog Timer Reset 인에이블
- RWT : Watchdog Timer 재출발



\* MiDAS1.1 제품군의 경우, RWT 는 오직 WDT 모드 0(WD[1:0] = [0,0])에서 사용된다. (Application Note #009 (AN009) 참조)

# 6.9. Timer/Counter : Timer 0/1

- ◆ 전통적인 80C52 타이머/카운터와 기능 호환
- ◆ Time base는 12 클럭 주기이다.

모드 Timer	모드 0 (M1,M0=00)	모드 1 (M1,M0=01)	모드 2 (M1,M0=10)	모드 3 (M1,M0=11)
Timer0	13-bit T/C	16-bit T/C	8-bit T/C 자동 재입력 (TL0 ← TH0)	8-bit T/C (TL0) → Timer0 interrupt 8-bit T/C (TH0) → Timer1 interrupt
Timer1	지원 안함		8-bit T/C 자동 재입력 (TL1 ← TH1)	지원 안함

### ✓ TMOD (89h) : Timer/Counter 0 모드 제어 레지스터

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

- GATE[3] : Timer 0 Gate 제어
- C/T[2] : Timer 0 선택.  
0 =  $F_{osc}/12$ 을 계수하는 타이머. (기본지정)  
1 = T0 pin 입력을 계수하는 카운터.
- M1, M0 : Timer 0 모드 선택  
[0,0] : 모드 0. 13-bit T/C.  
[0,1] : 모드 1. 16-bit T/C.  
[1,0] : 모드 2. 8-bit T/C 자동재입력  
[1,1] : 모드 3. Two 8-bit T/C

### ✓ TCON (88h) : Timer/Counter 0/1 제어 레지스터

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 오버플로우
- TR1 : Timer 1 동작
- TF0 : Timer 0 오버플로우
- TR0 : Timer 0 동작
- IE1 : External Interrupt 1 플래그
- IT1 : External Interrupt 1 촉발 방식 선택  
Edge 촉발 (IT1=1). Level 촉발 (IT1=0)
- IE0 : External Interrupt 0 플래그
- IT0 : External Interrupt 0 촉발 방식 선택  
Edge 촉발 (IT0=1). Level 촉발 (IT0=0)

### ✓ TLO (8Ah) : Timer/Counter 0 하위 바이트 레지스터

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

### ✓ TH0 (8Ch) : Timer/Counter 0 상위 바이트 레지스터

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

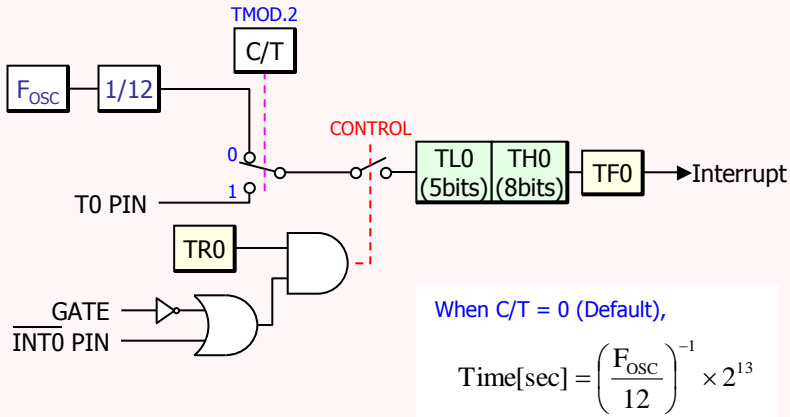
### ✓ TL1 (8Bh) : Timer/Counter 1 하위 바이트 레지스터

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

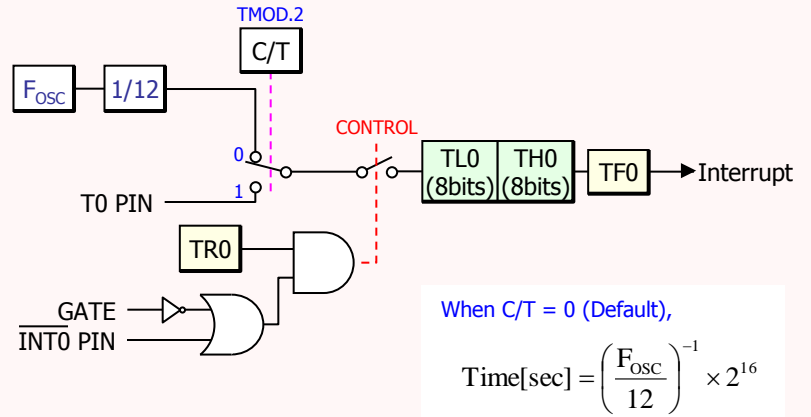
### ✓ TH1 (8Dh) : Timer/Counter 1 상위 바이트 레지스터

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

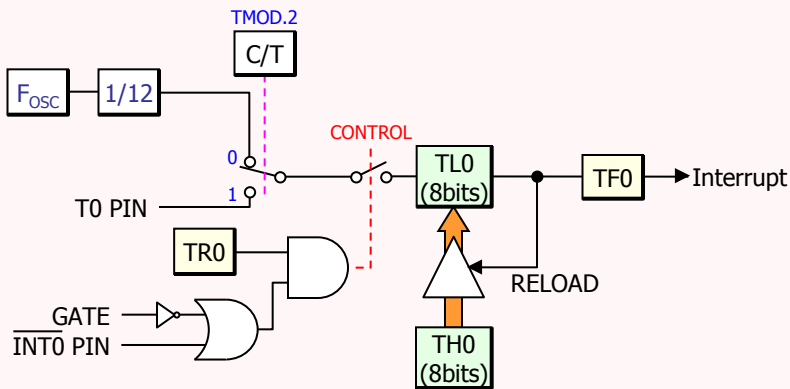
# 6.9. Timer/Counter : Timer 0 모드 설명



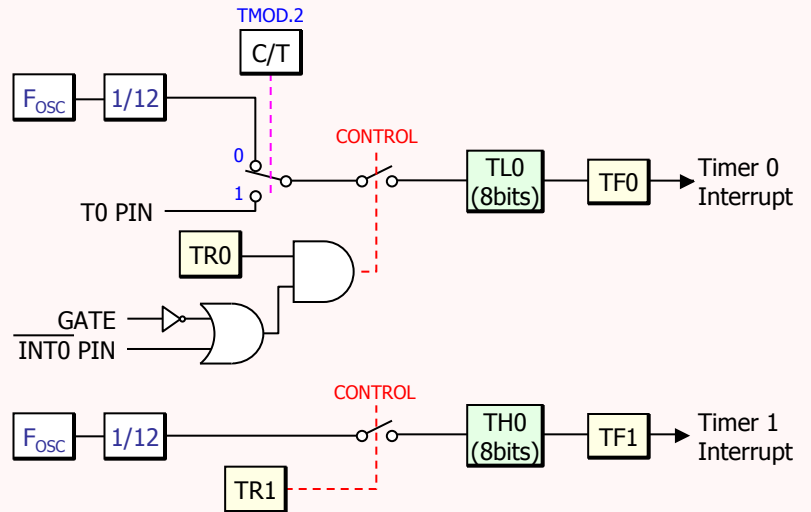
[모드 0]



[모드 1]

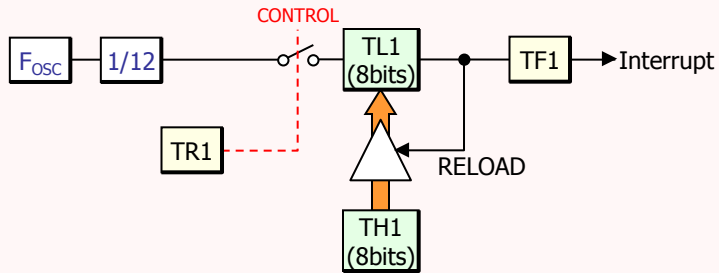


[모드 2]



[모드 3]

## 6.9. Timer/Counter : Timer 1 모드 설명



[모드 2]

# 6.10. UART

- ◆ 간소화된 8052 UART  
(오직 UART 모드 1만 지원한다.)

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

- ◆ UART 모드 1  
(Timer 1 오버플로우 사용)

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

### [Baudrate Examples]

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

- ✓ **PCON** (87h) : 전력 제어 레지스터

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

- SMOD1 : UART 모드에서 타이머 1 보레이트 2 배.

- ✓ **SCON** (98h) : 직렬 단자 제어 레지스터

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

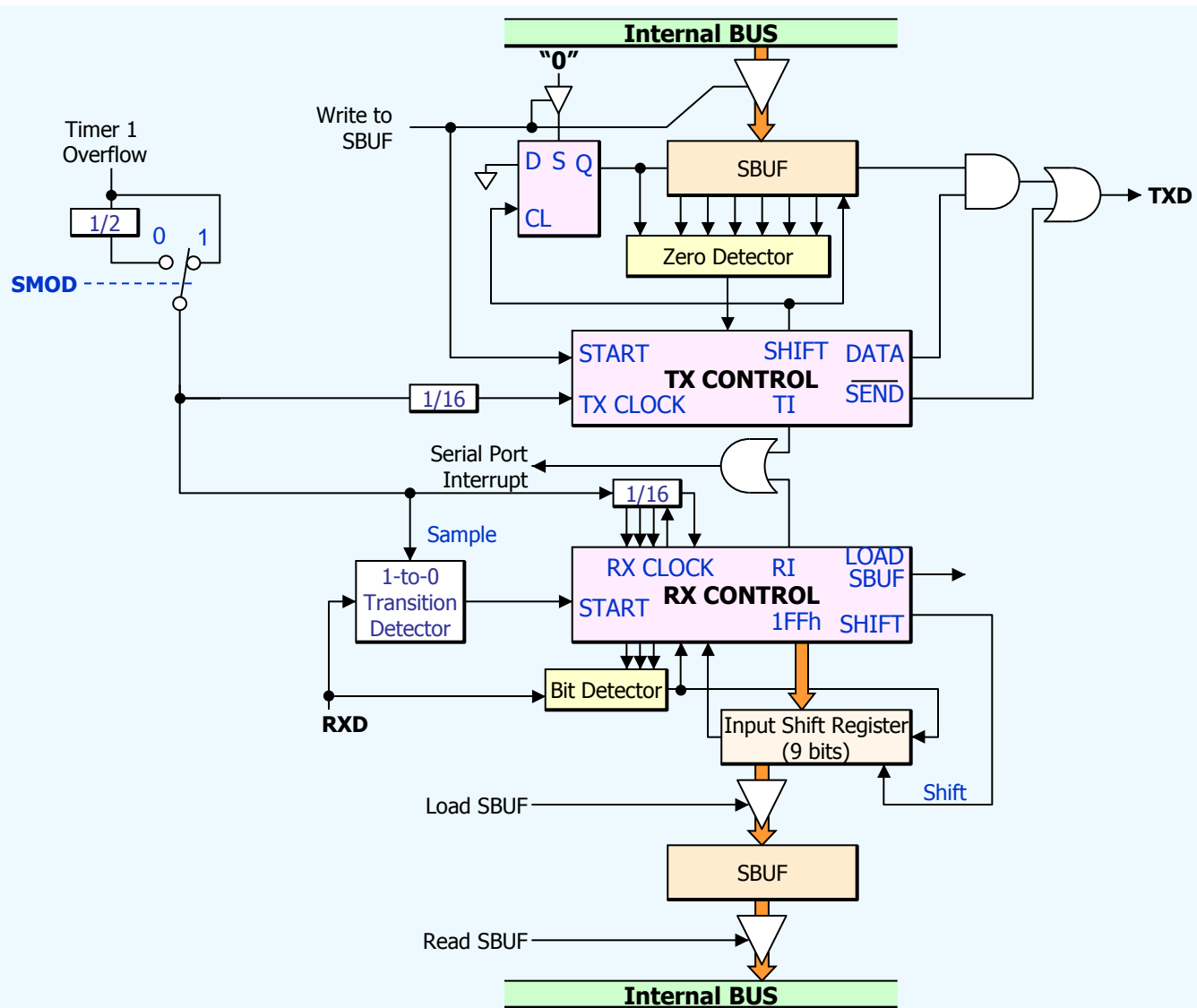
- REN : 직렬 수신 인에이블.
- TI : 송신 인터럽트 플래그. 소프트웨어가 소거해야 한다.
- RI : 수신 인터럽트 플래그. 소프트웨어가 소거해야 한다.

- ✓ **SBUF** (99h) : 직렬 데이터 버퍼 레지스터

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)

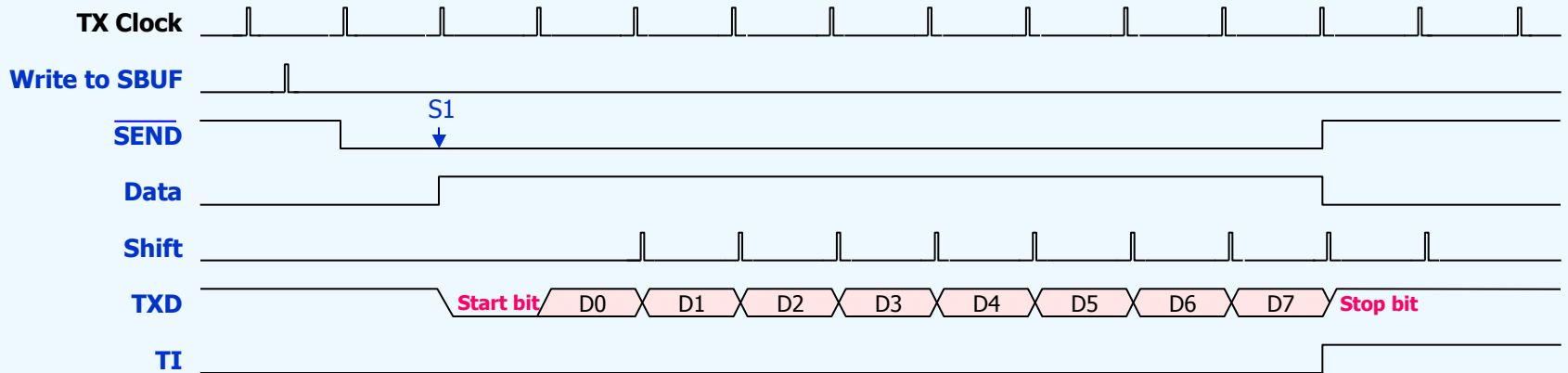
- 송신 버퍼와 수신 버퍼는 분리되어 있다.
- 읽기와 쓰기 주소는 같다.

# 6.10. UART : 모드 1 기능

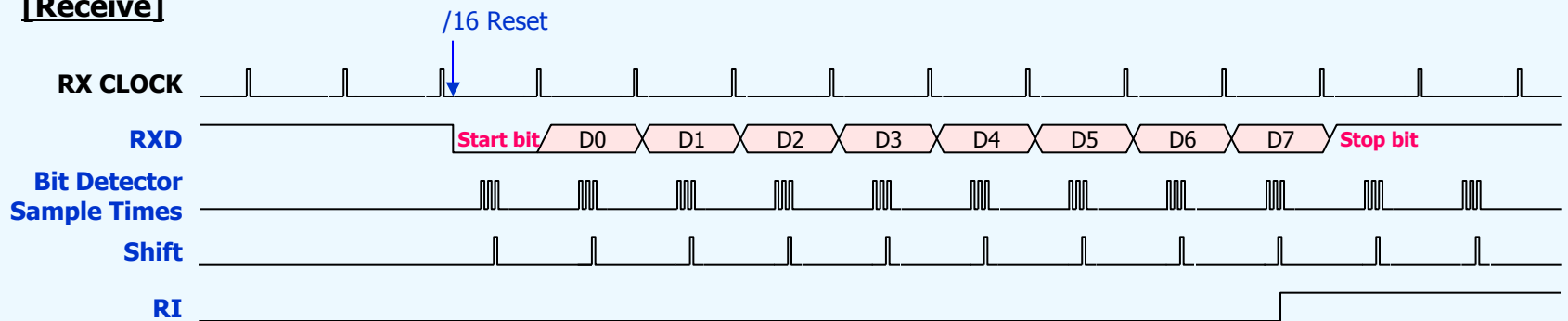


## 6.10. UART : 모드 1 타이밍

### [Transmit]

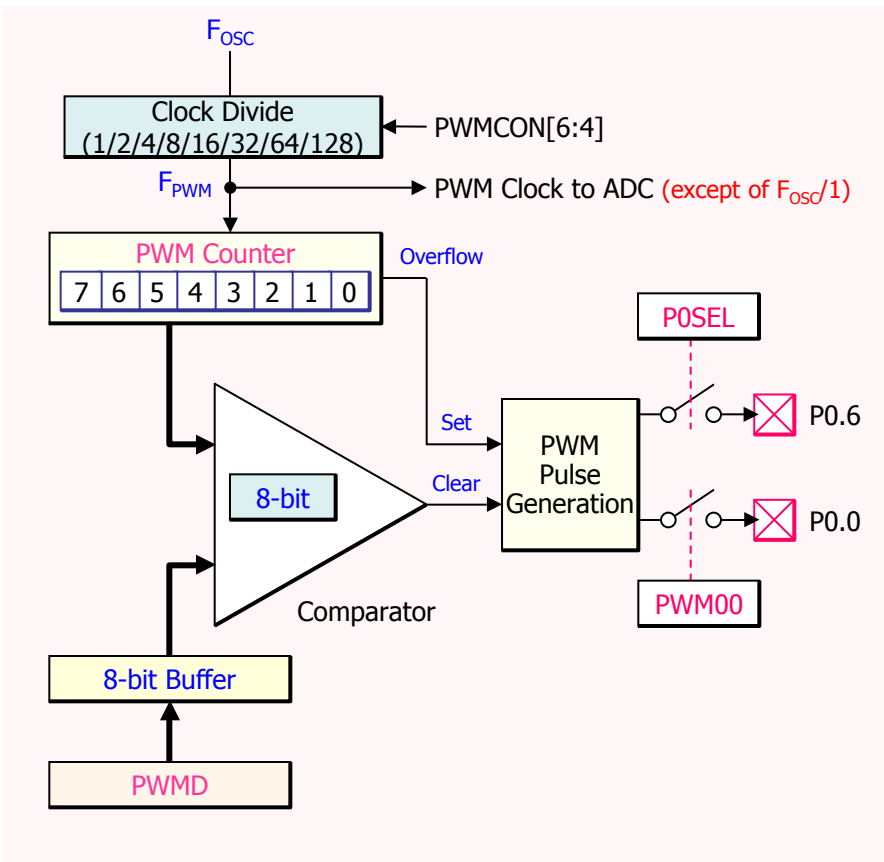


### [Receive]



# 6.11. PWM (Pulse Width Modulator)

- ◆ 지능형 1-채널 8-bit PWM
- ◆ PWM 카운터 갱신 모드 (8-bit 카운터 오버플로우 때 갱신)
- ◆ PWM 카운터는 소프트웨어에 의해 소거될 수 있다.
- ◆ PWM는 소프트웨어에 의해 정지하거나 출발(재개)된다.



### ✓ PWMCON (DCh) : PWM 제어 레지스터

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

- POSEL : P0.6으로 PWM 파형 출력 인에이블
- PS2\_P0, PS1\_P0, PS0\_P0 : 클럭 분주비 선택.  
 $[0,0,0] = F_{osc}/1$ ,  $[0,0,1] = F_{osc}/2$ ,  $[0,1,0] = F_{osc}/4$ ,  
 $[0,1,1] = F_{osc}/8$ ,  $[1,0,0] = F_{osc}/16$ ,  $[1,0,1] = F_{osc}/32$ ,  
 $[1,1,0] = F_{osc}/64$ ,  $[1,1,1] = F_{osc}/128$
- \* PWM 클럭( $F_{PWM}$ )이  $F_{osc}/10$ 이 아니면 ADC에 사용 가능.
- PWMF : PWM 인터럽트 플래그. 소프트웨어에 의해 소거
- CLR\_P0 : 카운터 계수 값 소거. 하드웨어에 의해 소거.
- RUN\_P0 : 카운터 동작.

### ✓ PWMD (DEh) : PWM Duty 데이터 레지스터

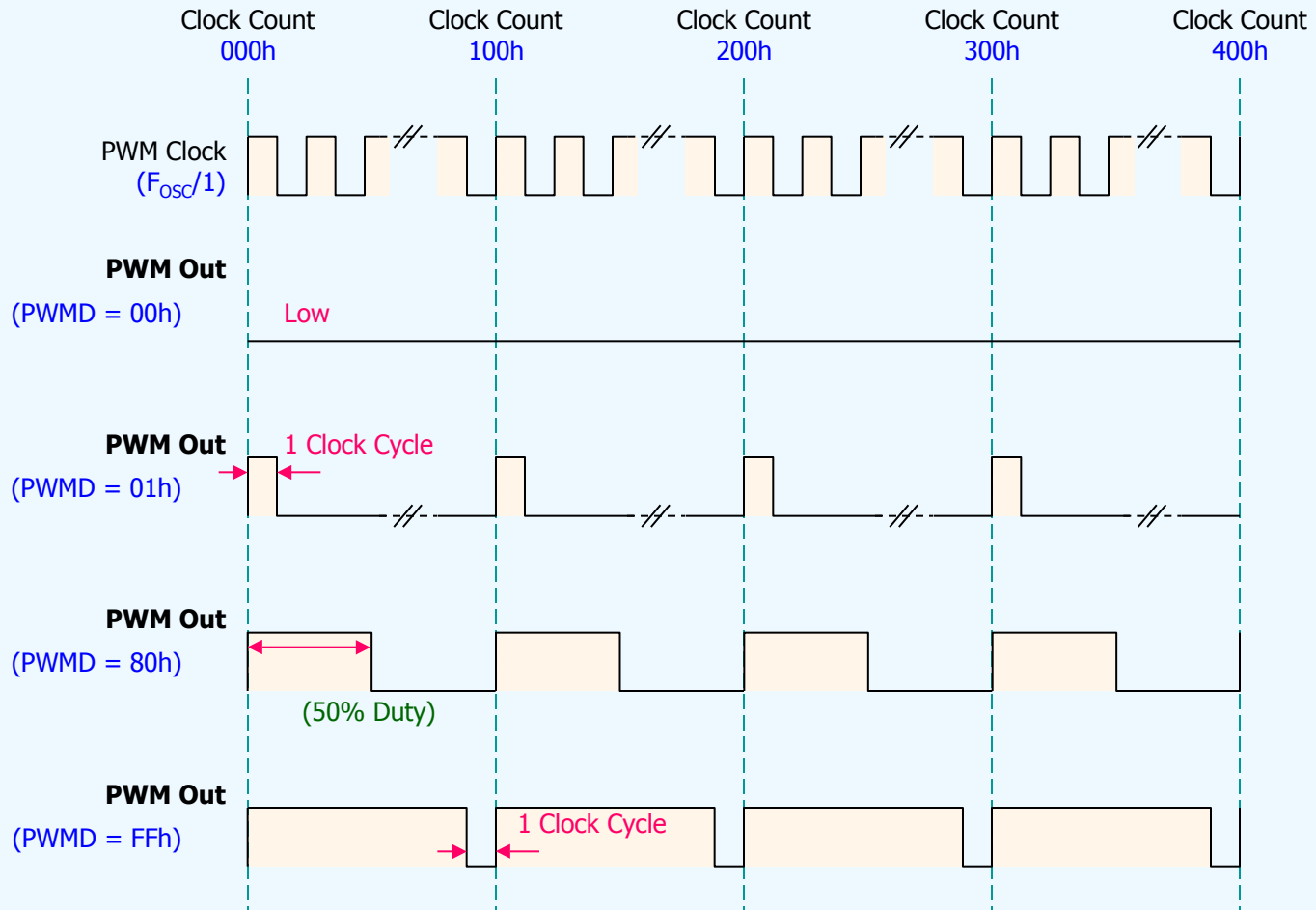
PWMD.7	PWMD.6	PWMD.5	PWMD.4	PWMD.3	PWMD.2	PWMD.1	PWMD.0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

### ✓ ALTSEL (E3h) : Alternate 기능 제어 레지스터

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

- PWM00 : P0.0으로 PWM 파형 출력 인에이블

# 6.11. PWM : Pulse Generation



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

- ◆ 12-채널 10-bit ADC (SAR Type)
- ◆ 최대 104ksps(초당 sample 수) @  $F_{ADC} = 10\text{MHz} \ \& \ 5\text{V}$ . (최대 52ksps @  $F_{ADC} = 5\text{MHz} \ \& \ 3\text{V}$ )

✓ **ADCSELH (E1h) : ADC 채널 선택 상위 레지스터**

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

✓ **ADCSEL (E2h) : ADC 채널 선택 하위 & MUX 선택 레지스터**

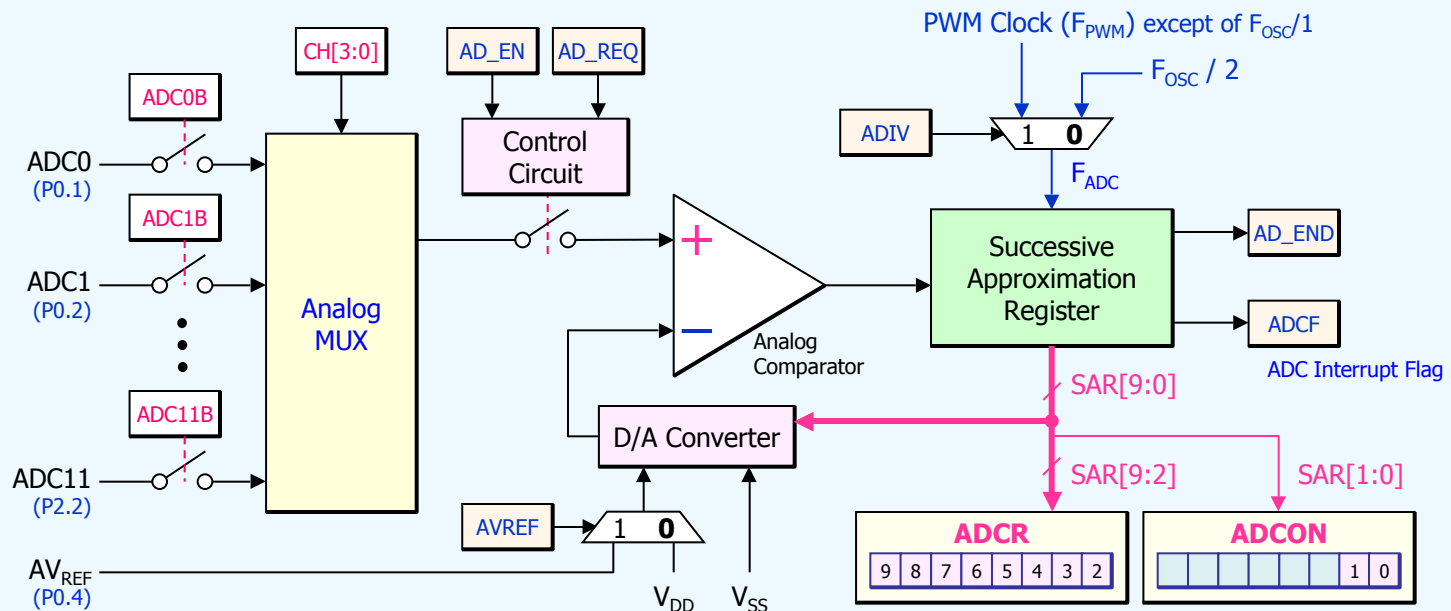
ADC3B	ADC2B	ADC1B	ADC0B	CH3	CH2	CH1	CH0
R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

✓ **ADCON (EFh) : ADC 제어 & ADC 결과 하위 레지스터**

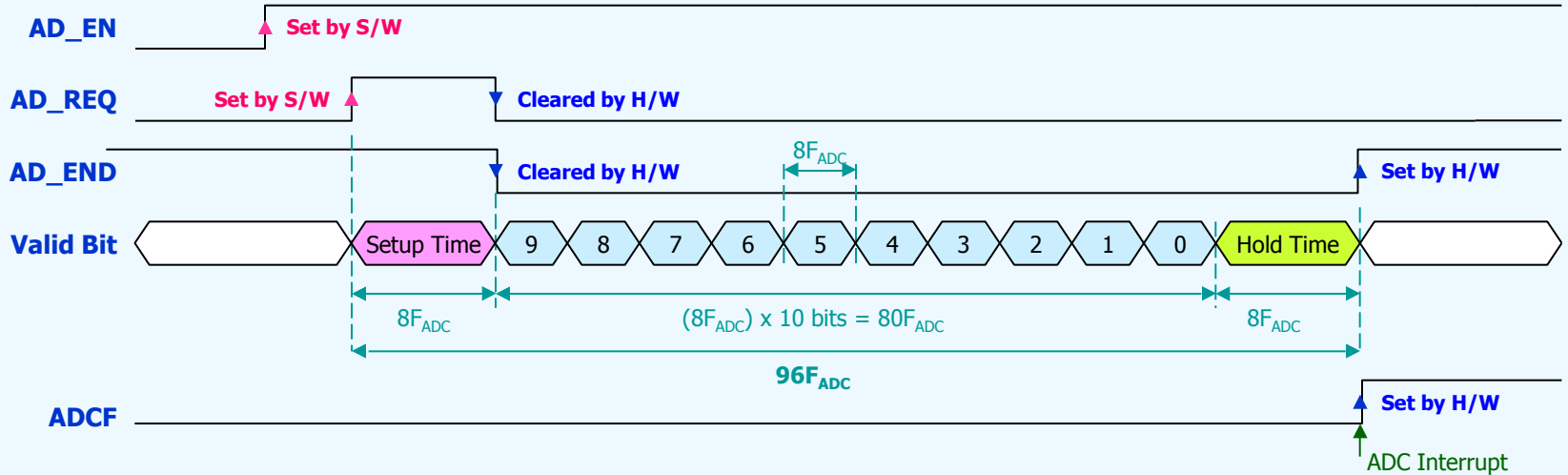
AD_EN	AD_REQ	AD_END	ADCF	AVREF	ADIV	SAR1	SAR0
R/W(0)	R/W(0)	R(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

✓ **ADCR (EEh) : ADC 결과 상위 레지스터**

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



## 6.12. ADC : 변환 타이밍



- ✓ **AD\_EN** : ADC Block 인에이블.  
소프트웨어에 의하여 설정되고 소거됨.
- ✓ **AD\_REQ** : ADC 변환 요구.  
소프트웨어가 설정하고 하드웨어가 소거함.  
각 샘플을 변환할 때마다 설정되어야 한다.
- ✓ **AD\_END** : 하드웨어에 의해 설정되고 소거됨.  
변환이 시작될 때 소거됨.  
변환이 끝날 때 설정됨.
- ✓ **ADCF** : ADC 인터럽트 플래그.  
하드웨어가 설정하고 소프트웨어가 소거함.  
ADC 인터럽트 처리 루틴에서 ADCF를 소거해야 함.

[클럭 주파수 별 ADC 변환 시간 표]

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

# 6.13. 인터럽트 : 11개 발생원 / 2-level 우선순위

- ◆ 10개의 인터럽트 발생원
  - ✓ Timer 0/1, UART, ADC, WDT, PWM, 4개의 외부.
- ◆ 2-level 인터럽트 우선순위

[Interrupt Vector Address]

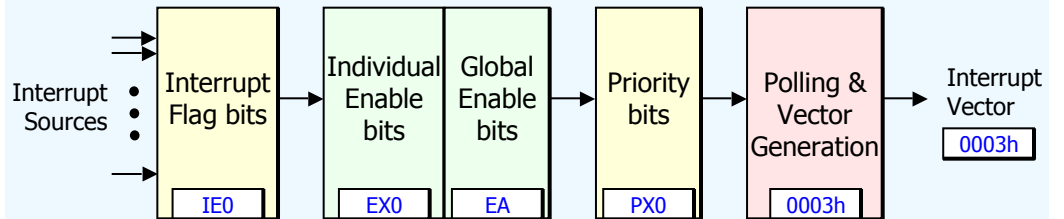
Interrupt Sources	Address	Priority Level
INT0	0003h	2 Levels
TF0	000Bh	2 Levels
INT1	0013h	2 Levels
TF1	001Bh	2 Levels
RI+TI	0023h	2 Levels
ADC	003Bh	2 Levels
INT2	0043h	2 Levels
INT3	004Bh	2 Levels
WDT	0063h	2 Levels
PWM	006Bh	2 Levels

↑ HIGH PRIORITY  
↓ LOW PRIORITY

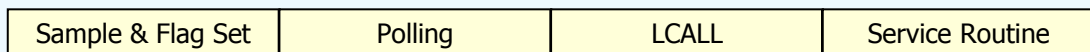
\* Interrupt related to SFR (refer to Appendix B : SFR Description)

✓ <b>TCON</b> (88h)	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
✓ <b>EXIF</b> (91h)	-	-	IE3	IE2	XT/RG	RGMD	RGSL	BGS
✓ <b>IE</b> (A8h)	EA	EADC	-	ES	ET1	EX1	ET0	EX0
✓ <b>IP</b> (B8h)	-	PADC	-	PS	PT1	PX1	PT0	PX0
✓ <b>EIE</b> (E8h)	-	-	EPWM	EWDT	-	-	EX3	EX2
✓ <b>EIP</b> (F8h)	-	-	PPWM	PWDT	-	-	PX3	PX2
✓ <b>WDCON</b> (D8h)	WD1	WD0	-	-	WDIF	WTRF	EWT	RWT
✓ <b>PWMCON</b> (DCh)	POSEL	PS2_P0	PS1_P0	PS0_P0	-	PWMF	CLR_P0	RUN_P0

[Interrupt Vector Generation Flow]

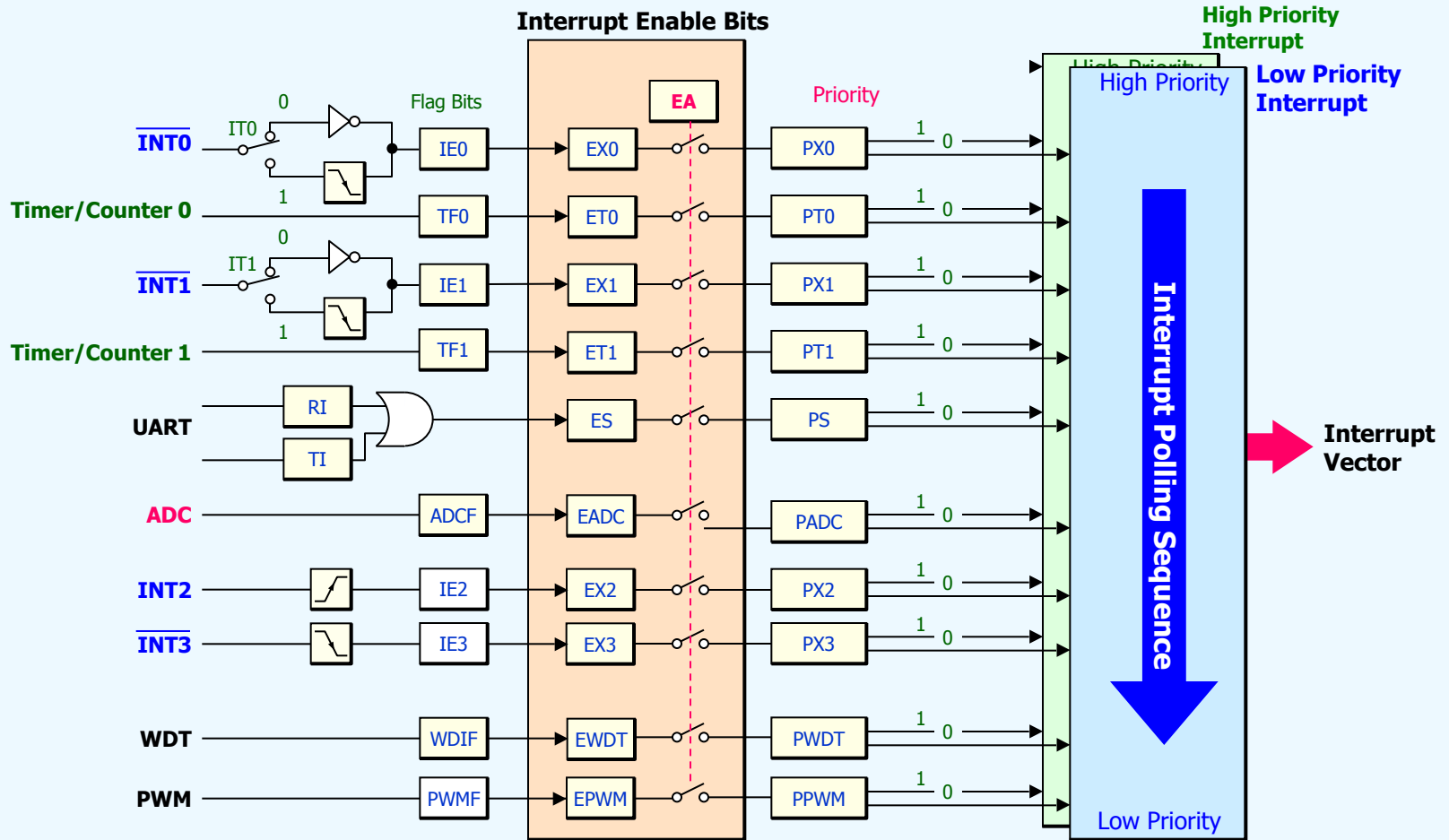


[Response Sequence]



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

## 6.13. 인터럽트 기능적 설명



## 6.14. Reset 회로 : 3개의 Reset 발생원

### ◆ LVD(POR) Reset

- ✓ 전원을 켤 때, Power-on Reset.
- ✓  $V_{RST}$  아래로 떨어질 때 전력 오류 리셋

### ◆ External RESET Pin

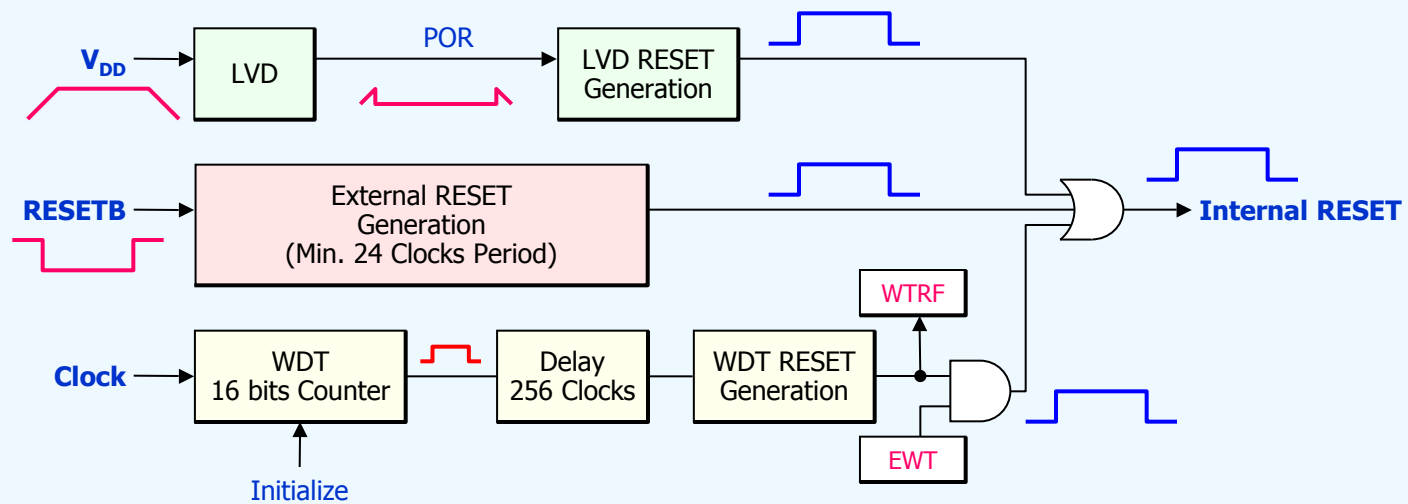
- ✓ RESETB Pin은 최소 24 클럭주기를 "L" 상태를 유지해야 한다.
- ✓ 링 오실레이터가 동작해야 한다.
- ✓ 주의, Reset 신호가 Glitch 잡음이 아니어야 한다.

### ◆ WDT Reset : 소프트웨어에 의해 동작여부 선택함

✓ **WDCON** (D8h) : Watchdog Timer 제어 레지스터

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

- WTRF : Watchdog Timer Reset 플래그. 소프트웨어만 소거함.
- EWT : Watchdog Timer Reset 인에이블.



# 6.15. 클럭 회로

- ◆ 2개의 시스템 클럭 발생원: 링 발진기나 외부 크리스탈
- ◆ 링 발진기로 시스템 클럭은 기본지정됨
- ◆ 링 발진기를 사용하여 정지 모드에서 신속하게 탈출

제어 플래그				시스템 클럭	상태 Bit	
XT/RG	XTOFF	RINGON	RGSL		RGMD	XTUP
1	0	X	X	크리스탈 발진기	0	1
0	X	1	X	링 발진기	1	0/1
1	0	X	0	크리스탈 발진기 (정지 모드 탈출하는 동안)	0	0
0	X	1	1	링 발진기 (정지 모드 탈출하는 동안)	1	0

✓ **EXIF (91h)** : 외부 인터럽트 플래그 레지스터

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

✓ **OSCICN (BEh)** : 내부 링 발진기 제어 레지스터

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

✓ **STATUS (C5h)** : 크리스탈 상태 레지스터

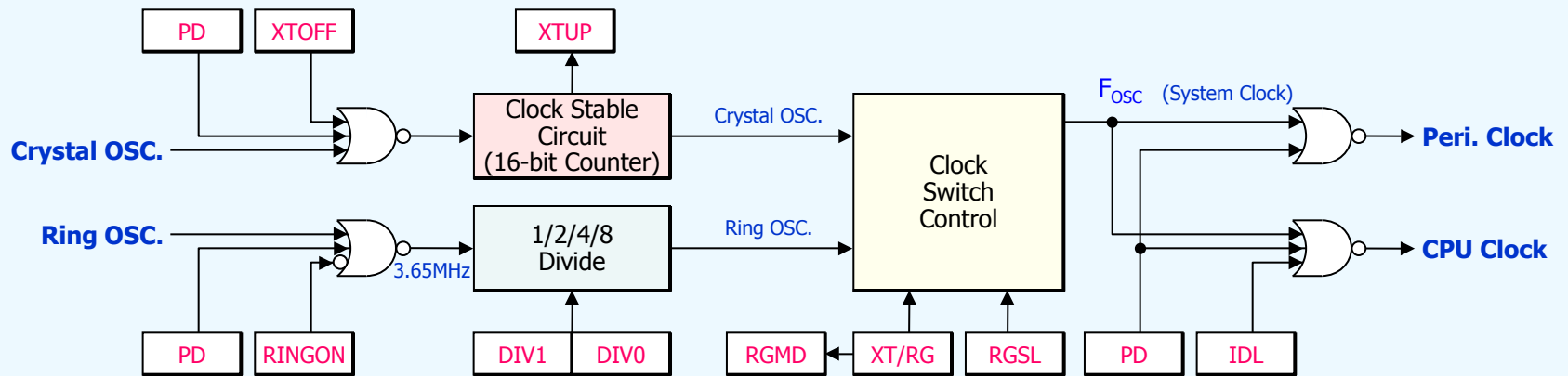
-	-	-	XTUP	-	-	-	-
R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

✓ **PMR (C4h)** : 전력 관리 제어 레지스터

-	-	-	-	XTOFF	-	-	-
R(0)	R(0)	R(0)	R(0)	R(0)	R(0)	R(0)	R(0)

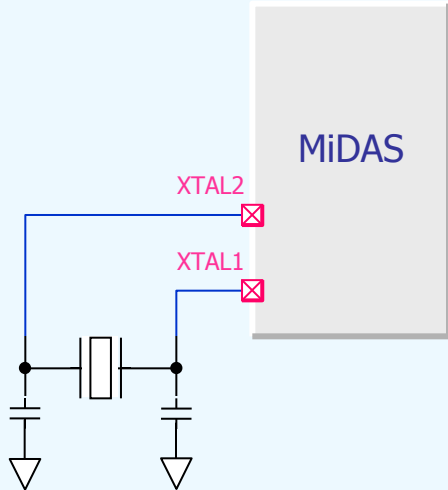
✓ **PCON (87h)** : 전력 제어 레지스터

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

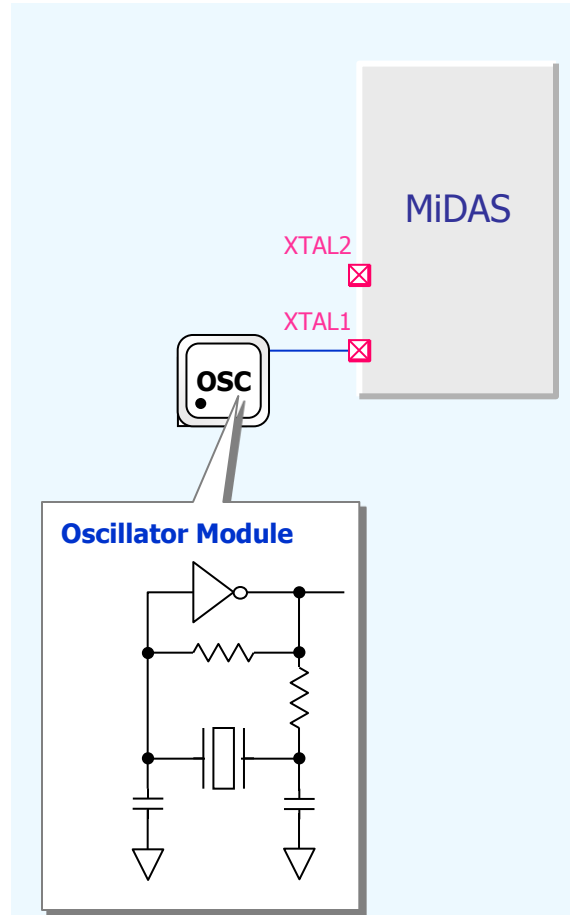


## 6.15. 클럭 회로: 구성을 위한 지침

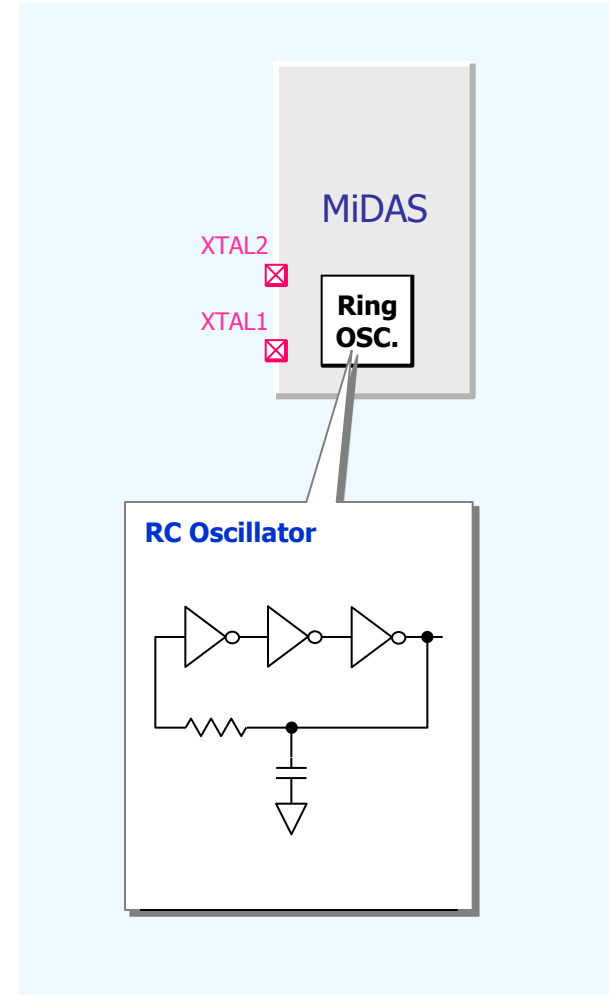
### ◆ 크리스탈 발진기 (오직 A0 계열)



### ◆ 발진기 모듈

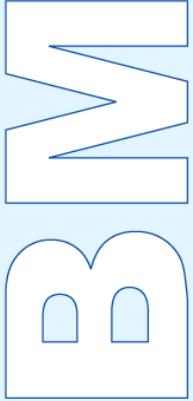


### ◆ 내부 링 발진기





# MiDAS Family



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

## 6.16. 전력 관리: 3가지 모드

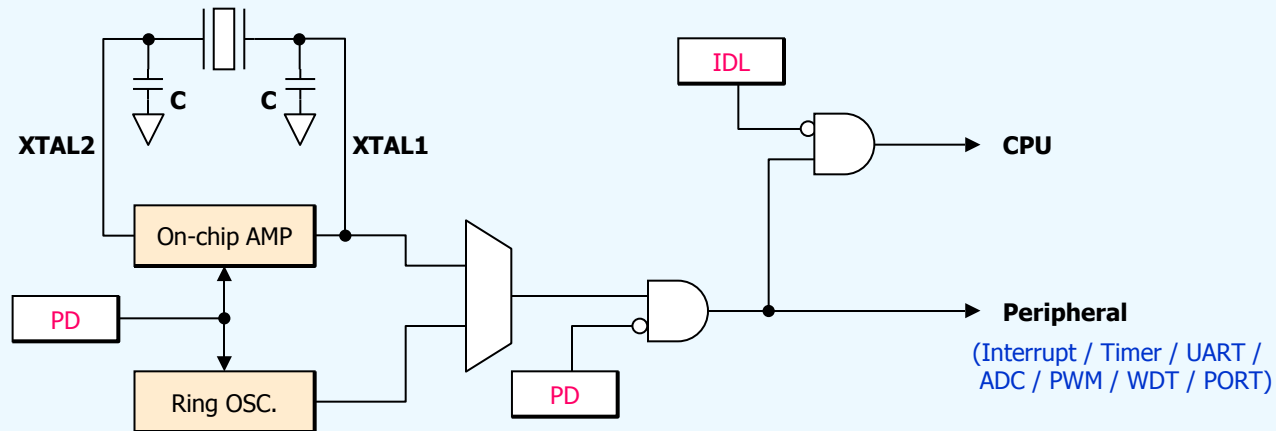
- ◆ **Active 모드** : CPU와 주변회로 동작.
- ◆ **Idle 모드** : 오직 주변회로만 동작.
  - ✓ 모든 종류의 인터럽트로 탈출. CPU는 동작 재개.  
그러나, WDT와 level 촉발 인터럽트는 제외된다.
  - ✓ 모든 종류의 reset으로 탈출. CPU는 재출발.
- ◆ **정지 모드 1/2** : CPU와 주변회로 정지.
- ◆ **정지 모드 1** : WDT가 정지
  - ✓ 외부 인터럽트 INT0/INT1 (level 촉발)에 의해 탈출.  
→ CPU 동작 재개.
  - ✓ 모든 종류 리셋으로 탈출. (ex: RESETB, LVD. etc)  
→ CPU는 재출발.

- ◆ **정지 모드 2** : WDT가 동작
  - ✓ 외부 인터럽트 INT0/INT1 (level 촉발) 또는 WDT 인터럽트로 탈출.  
→ CPU는 동작 재개.
  - ✓ 모든 종류의 reset으로 탈출. (ex: RESETB, LVD, .etc)  
→ CPU 재출발.

✓ **PCON (87h)** : 전력 제어 레지스터

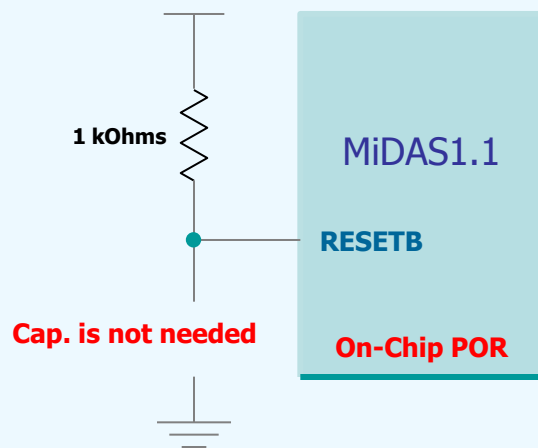
SMOD1	-	-	POF	GF1	GF0	PD	IDL
R/W(0)			R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- PD : 정지 모드 (Power-down).
- IDL : IDLE 모드.



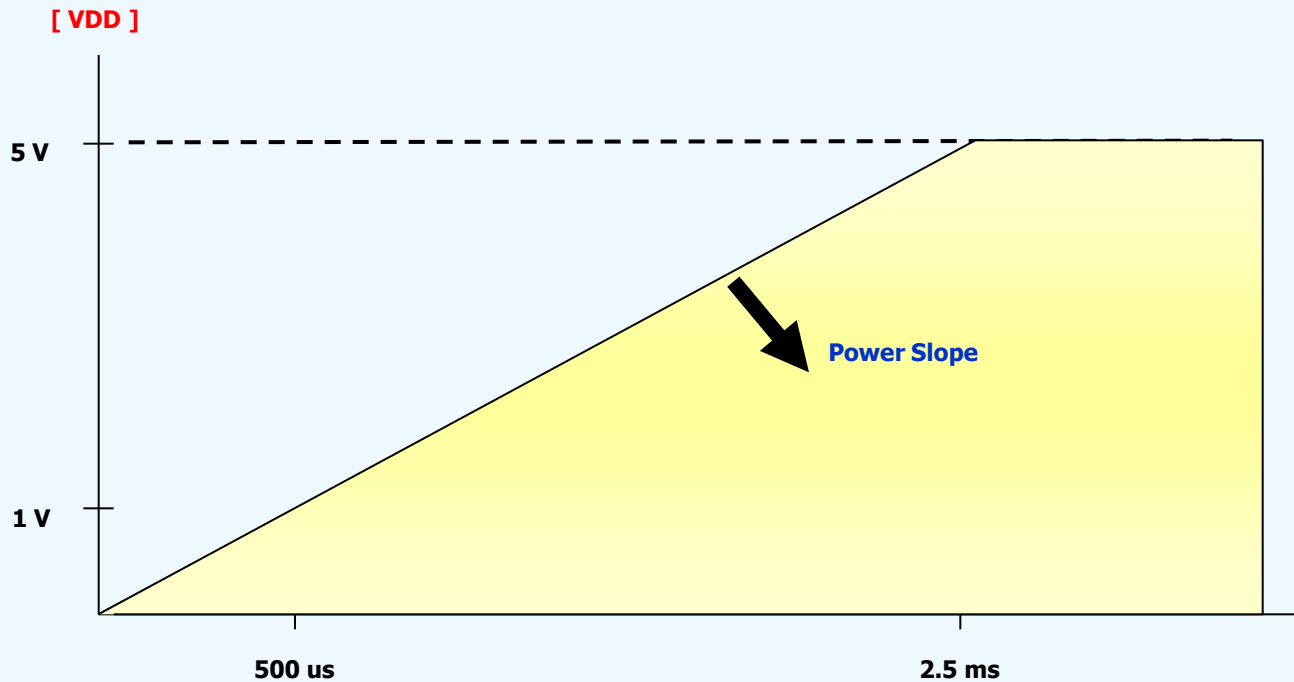
## 7. 강점: On-Chip POR

- ◆ On-Chip POR (Power On Reset)으로 필요 없는 커패시터를 제거하여 시스템 비용을 삭감할 수 있다.



## 8. 전력 전압 상승 기울기 지침

- ◆ 전력 전압의 상승 기울기는  $0.0V/us$ 에서  $1.0V/500us(5V/2.5ms)$ 의 범위에 있어야 한다.  
(이것은 전력 전압이 정상 상태에 도달할 때까지 단조 증가해야 한다는 것을 포함한다.)



## 9. 절대 최대 동작범위 한계

항목	조건	범위
접지에 대한 pin 전압	-	-0.5V에서 ( $V_{DD}+0.5V$ )
접지에 대한 $V_{DD}$ 전압	-	-0.5V에서 6.5V
출력 전압	-	-0.5V에서 ( $V_{DD}+0.5V$ )
Current High 상태 출력 전류	단일 입출력 pin 동작	-25mA
	모든 입출력 pin 동작	-100mA
Low 상태 출력 전류	단일 입출력 pin 동작	+30mA
	모든 입출력 pin 동작	+150mA
동작 온도	-	-40 °C에서 120 °C
저장 온도	-	-65 °C에서 +150 °C
무연 납땜 온도	-	250 또는 260 °C 피크, 피크 온도에서 5 °C 이내 20초

# 10. DC 특성

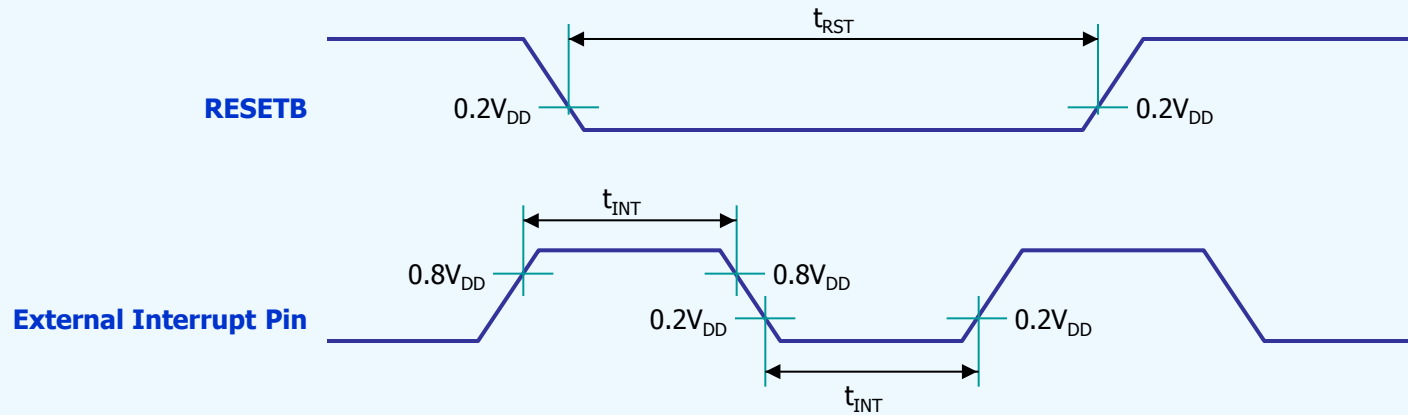
\* 별도로 기술되지 않았다면,  $TA = -20\text{ }^{\circ}\text{C} \sim +85\text{ }^{\circ}\text{C}$ ,  $V_{DD} = 2.4\text{V} \sim 5.5\text{V}$ .

파라미터	기호	Pin	조건	값			단위
				최소	보통	최대	
Low 입력 전압	$V_{IL1}$	P0, P2	$V_{DD} = 2.4\text{V} \sim 5.5\text{V}$	-0.5	-	$0.2V_{DD} - 0.1$	V
	$V_{IL2}$	XTAL1, XTAL2, RESETB		-0.5	-	$0.3V_{DD}$	
High 입력 전압	$V_{IH1}$	P0, P2	$V_{DD} = 2.4\text{V} \sim 5.5\text{V}$	$0.2V_{DD} + 1.0$	-	$V_{DD} + 0.5$	V
	$V_{IH2}$	XTAL1, XTAL2, RESETB		$0.7V_{DD}$	-	$V_{DD} + 0.5$	
Low 출력 전압	$V_{OL1}$	XTAL1, XTAL2, P0, P2	$I_{OL} = 20\text{mA} @V_{DD}=5\text{V}$ ( $I_{OL} = 5\text{mA} @V_{DD}=2.6\text{V}$ )	-	-	$0.3V_{DD}$	V
	$V_{OL2}$	RESETB	$I_{OL} = 10\text{mA} @V_{DD}=5\text{V}$ ( $I_{OL} = 2.5\text{mA} @V_{DD}=2.6\text{V}$ )	-	-	$0.3V_{DD}$	
High 출력 전압	$V_{OH}$	XTAL1, XTAL2, P0, P2	$I_{OH} = -15\text{mA} @V_{DD}=5\text{V}$ ( $I_{OH} = -2.5\text{mA} @V_{DD}=2.6\text{V}$ )	$0.7V_{DD}$	-	-	V
	$V_{OHP1}$	P0, P2 (pull-up resistor only)	$I_{OH} = -140\mu\text{A} @V_{DD}=5\text{V}$ ( $I_{OH} = -20\mu\text{A} @V_{DD}=2.6\text{V}$ )	$0.7V_{DD}$	-	-	V
	$V_{OHP2}$	XTAL1, XTAL2 (pull-up resistor only)	$I_{OH} = -10\mu\text{A} @V_{DD}=5\text{V}$ ( $I_{OH} = -1.5\mu\text{A} @V_{DD}=2.6\text{V}$ )	$0.7V_{DD}$	-	-	V
Input 누설 전류	$I_{IL}$	XTAL1, XTAL2 제외한 모든 pin	$V_{IN} = V_{IH}$ or $V_{IL}$	-	-	$\pm 1$	$\mu\text{A}$
Pin 캐패시턴스	$C_{I0}$	모두	$V_{DD} = 5\text{V}$	-	10	-	pF

# 11. AC 특성

\* 기재되지 않았으면  $TA = -20\text{ }^{\circ}\text{C} \sim +85\text{ }^{\circ}\text{C}$ .

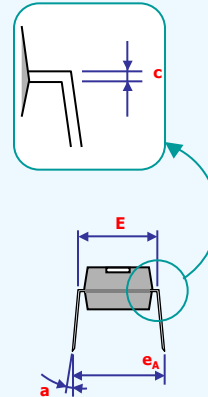
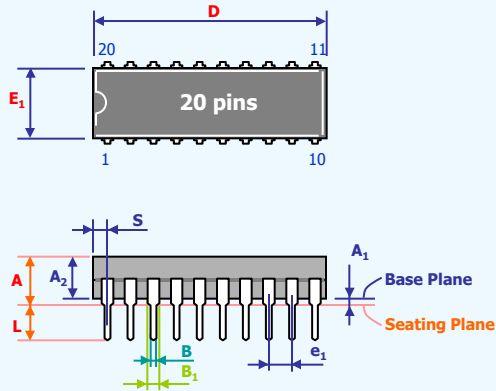
파라미터	기호	Pin	조건	값			단위
				Min.	Typ.	Max.	
동작 주파수	$F_{\text{OSC}}$	XTAL1, XTAL2	$V_{\text{DD}} = 5\text{V} \pm 10\%$	1	-	20	MHz
			$V_{\text{DD}} = 3\text{V} \pm 10\%$	1	-	10	
RESETB 입력 폭	$t_{\text{RST}}$	RESETB	$V_{\text{DD}} = 5\text{V} \pm 10\%$	24	-	-	$F_{\text{OSC}}$
			$V_{\text{DD}} = 3\text{V} \pm 10\%$	24	-	-	
외부 인터럽트 입력 폭	$t_{\text{INT}}$	External Interrupt	$V_{\text{DD}} = 5\text{V} \pm 10\%$	4	-	-	$F_{\text{OSC}}$
			$V_{\text{DD}} = 3\text{V} \pm 10\%$	4	-	-	



## 12. ADC 특성

파라메터	기호	조건	값			Unit	
			최소	보통	최대		
전력 전압	$V_{DDADC}$	-	2.4	-	5.5	V	
입력 전압	$V_{INADC}$	-	$V_{SS}$	-	$V_{DD}$	V	
해상도	$RES_{ADC}$	-	-	10	-	bit	
동작 주파수	$F_{ADC}$	$V_{DD} = 4.5V \sim 5.5V$ $V_{DD} = 2.4V \sim 3.3V$	-	-	10 5	MHz	
변환 시간	$t_{ADC}$	-	-	$96 / F_{ADC}$	-	s	
전체적 정확성	$OA_{ADC}$	$V_{DD} = 5V, F_{ADC} = 10MHz$ $V_{DD} = 3V, F_{ADC} = 5MHz$	-	$\pm 2$	$\pm 4$	LSB	
전체적 비선형성	$INL_{ADC}$	$V_{DD} = 5V, F_{ADC} = 10MHz$ $V_{DD} = 3V, F_{ADC} = 5MHz$	-	$\pm 2$	$\pm 4$	LSB	
Differential 비선형성	$DNL_{ADC}$	$V_{DD} = 5V, F_{ADC} = 10MHz$ $V_{DD} = 3V, F_{ADC} = 5MHz$	-	$\pm 0.5$	$\pm 1$	LSB	
0 입력 오류	$ZIE_{ADC}$	$V_{DD} = 5V, F_{ADC} = 10MHz$ $V_{DD} = 3V, F_{ADC} = 5MHz$	-	$\pm 2$	$\pm 4$	LSB	
최대 입력 오류	$FSE_{ADC}$	$V_{DD} = 5V, F_{ADC} = 10MHz$ $V_{DD} = 3V, F_{ADC} = 5MHz$	-	$\pm 2$	$\pm 4$	LSB	
아날로그 입력 커패시턴스	$C_{INADC}$	-	-	10	15	pF	
ADC 전류	Active	$I_{ADC}$	$V_{DD} = 5V, F_{ADC} = 10MHz$	-	1	2	mA
			$V_{DD} = 3V, F_{ADC} = 5MHz$	-	0.3	0.6	
	정지	$V_{DD} = 5V$	-	-	100	nA	

# 13. Package Dimensions : 20-PDIP



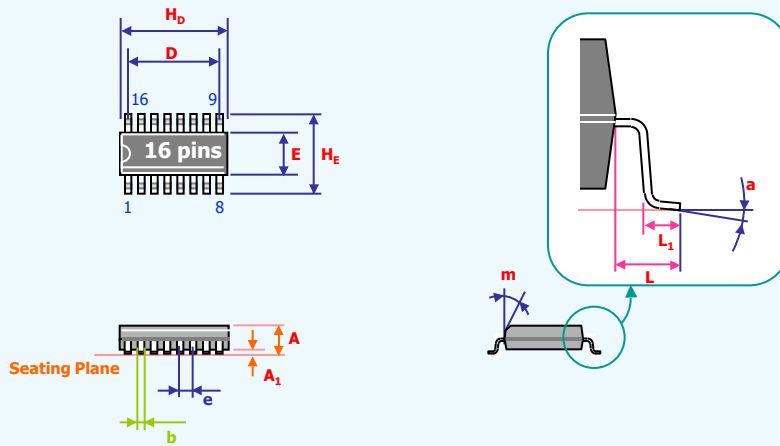
[20-SPDIP]

Symbol	Dimension in Inches			Dimension in mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
A	-	-	0.200	-	-	5.080
A <sub>1</sub>	0.015	-	-	0.381	-	-
A <sub>2</sub>	0.150	0.155	0.160	3.810	3.937	4.064
B	0.016	0.018	0.022	0.406	0.457	0.559
B <sub>1</sub>	0.045	0.055	0.065	1.143	1.397	1.651
c	0.008	0.010	0.012	0.203	0.254	0.356
D	1.045	1.055	1.075	26.543	26.797	27.305
E	0.290	0.300	0.310	7.366	7.62	7.874
E <sub>1</sub>	0.249	0.250	0.251	6.10	6.35	6.60
e <sub>1</sub>	0.090	0.100	0.110	2.286	2.540	2.794
L	0.120	0.130	0.140	3.048	3.302	3.556
a	0 <sup>ø</sup>	-	15 <sup>ø</sup>	0 <sup>ø</sup>	-	15 <sup>ø</sup>
e <sub>A</sub>	0.330	0.350	0.370	8.382	8.89	9.398
S	-	-	0.090	-	-	2.286

**Notes:**

1. Dimension D Max. & S include mold flash or tie bar Burns.
2. Dimension E<sub>1</sub> dose not include interlead flash.
3. Dimension D & E<sub>1</sub> include mold mismatch and are determined at the mold parting line.
4. Dimension B<sub>1</sub> does not include dambar protrusion/intrusion.
5. General appearance spec. should be based on final visual inspection spec.

# 13. Package Dimensions : 16-TSSOP



[16-TSSOP]

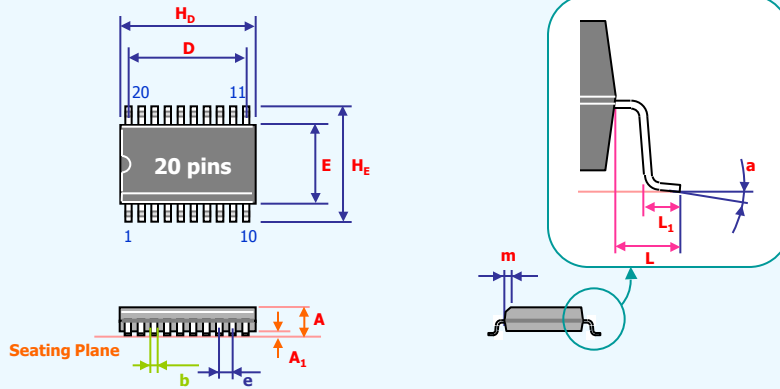
Symbol	Dimension in Inches			Dimension in mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
A	0.037	0.039	0.041	0.95	1.00	1.05
$A_1$	0.015	0.017	0.019	0.3865	0.4365	0.4865
b	0.008	0.009	0.009	0.20	0.22	0.24
D	0.176	0.179	0.182	4.47	4.55	4.63
E	0.171	0.173	0.175	4.35	4.4	4.45
$H_b$	0.200	0.202	0.204	5.077	5.127	5.177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

**Notes:**

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

# 13. Package Dimensions : 20/16-SOP

## [20-SOP]

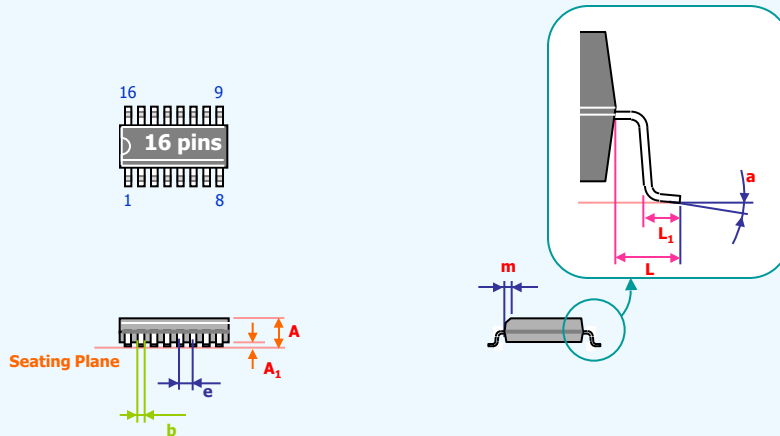


Symbol	Dimension in Inches			Dimension in mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
A	0.093	0.099	0.104	2.35	2.45	2.65
A <sub>1</sub>	0.004	0.008	0.012	0.10	0.20	0.30
b	0.014	0.016	0.019	0.35	0.42	0.49
D	-	0.450	-	-	11.43	-
E	0.291	0.295	0.299	7.40	7.50	7.60
H <sub>b</sub>	0.496	0.504	0.512	12.60	12.80	13.00
H <sub>E</sub>	0.404	0.411	0.419	10.26	10.45	10.65
L	0.057	0.058	0.060	1.43	1.48	1.53
L <sub>1</sub>	0.034	0.038	0.042	0.86	0.96	1.07
a	0°	-	8°	0°	-	8°
e	0.050 BSC			1.27 BSC		
m	0.020	0.025	0.030	0.50	0.62	0.75

**Notes:**

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

## [16-SOP]

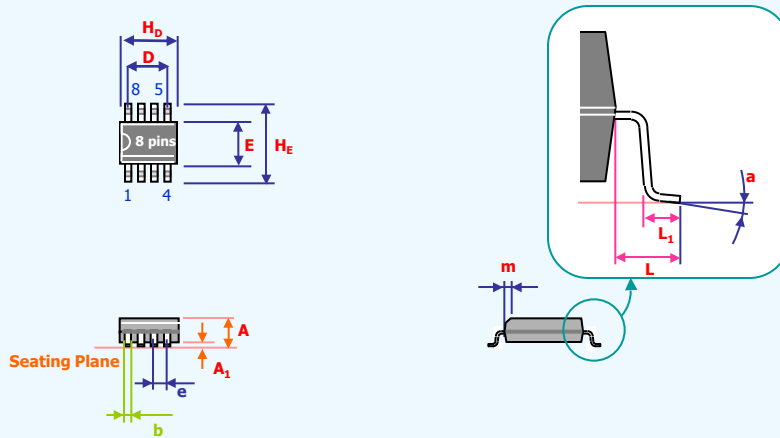


Symbol	Dimension in Inches			Dimension in mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
A	0.093	0.099	0.104	2.35	2.45	2.65
A <sub>1</sub>	0.004	0.008	0.012	0.10	0.20	0.30
b	0.014	0.016	0.019	0.35	0.42	0.49
D	-	.350	-	-	8.89	-
E	0.150	0.153	0.157	3.80	3.90	4.00
H <sub>b</sub>	0.398	0.405	0.413	10.10	10.29	10.50
H <sub>E</sub>	0.234	0.239	0.244	5.95	6.07	6.20
L	0.038	0.043	0.048	0.97	1.08	1.2
L <sub>1</sub>	0.022	0.027	0.032	0.58	0.70	0.82
a	0°	-	8°	0°	-	8°
e	0.050 BSC			1.27 BSC		
m	0.010	0.015	0.020	0.25	0.37	0.50

**Notes:**

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

# 13. Package Dimensions : 8-SOP



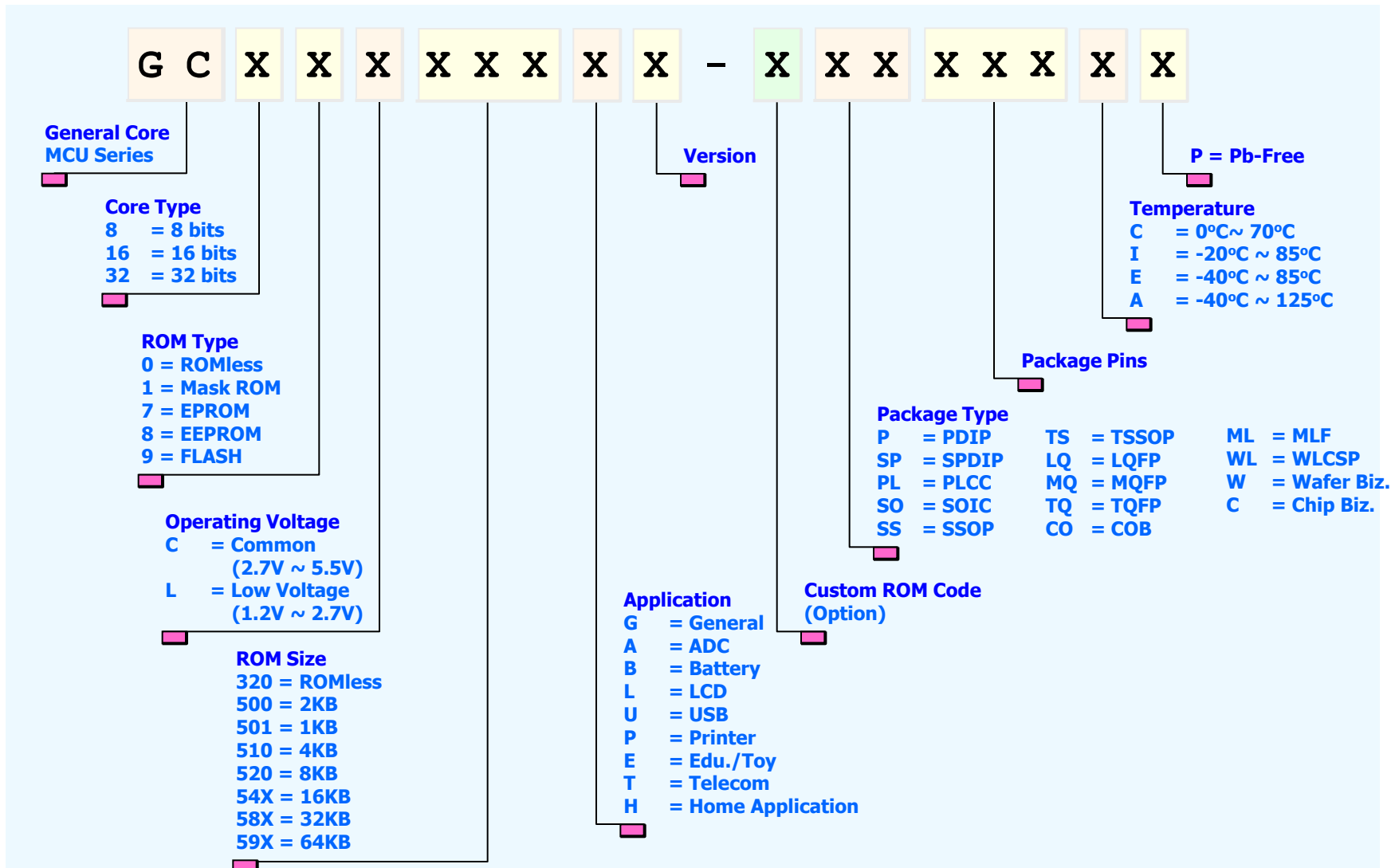
[8-SOP]

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

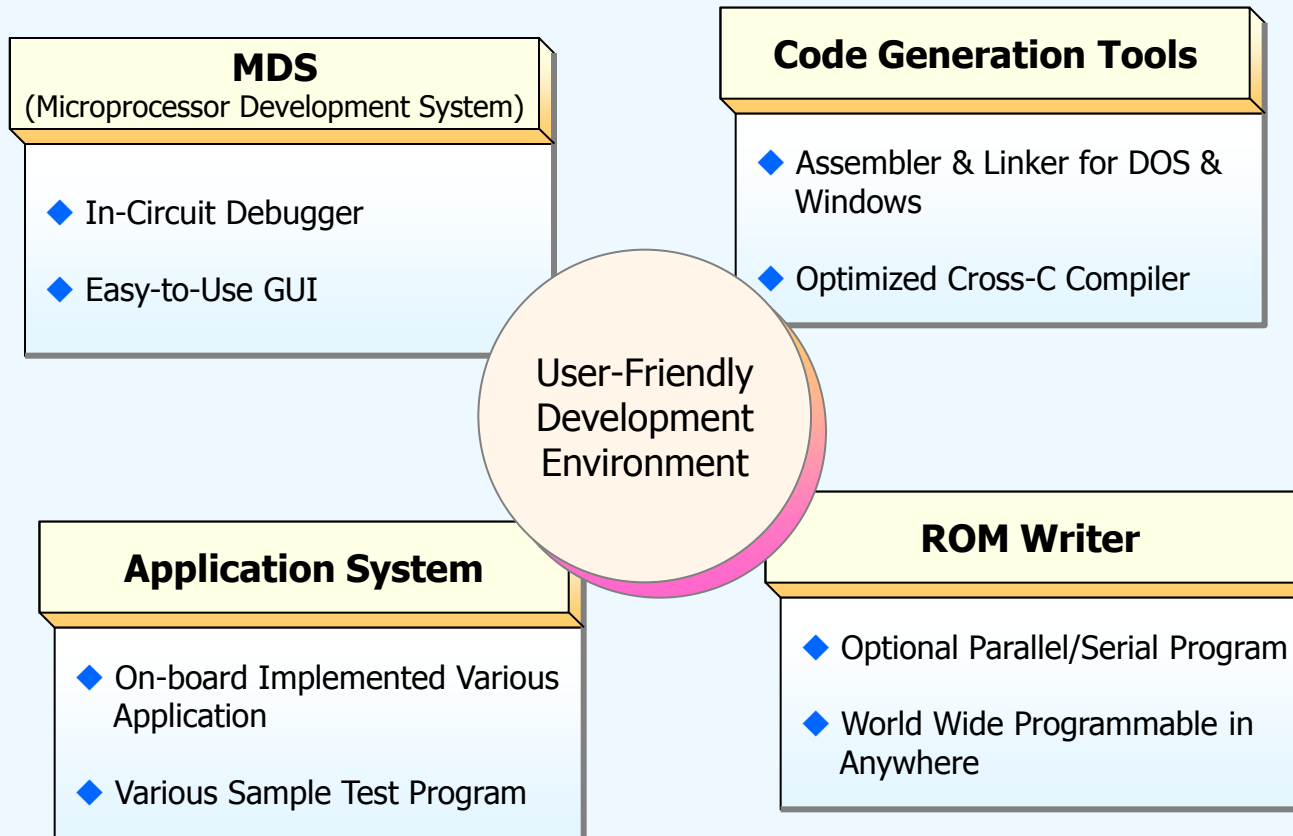
**Notes:**

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

# 14. Product Numbering System



## 15. 지원 툴



## 부록 A : 명령어 Set (1/19)

### ◆ Note on Instruction Set and Addressing Modes

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

# 부록 A : 명령어 Set (2/19)

## ADD A, <src-byte>

### Add

ADD A, Rn

Operation : (A) ← (A) + (Rn)

ADD A, @Ri

Operation : (A) ← (A) + ((Ri))

ADD A, direct

Operation : (A) ← (A) + (direct)

ADD A, #date

Operation : (A) ← (A) + data

## ADDC A, <src-byte>

### Add with Carry

ADDC A, Rn

Operation : (A) ← (A) + (C) + (Rn)

ADDC A, @Ri

Operation : (A) ← (A) + (C) + ((Ri))

ADDC A, direct

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

ADDC A, #date

Operation : (A) ← (A) + (C) + data

## 1 Machine Cycle = 4 Clocks

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

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

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

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

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

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

direct addr

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

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

immediate data

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

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

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

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

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

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

direct addr

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

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

immediate data

# 부록 A : 명령어 Set (3/19)

## SUBB A, <src-byte>

### Subtract with Borrow

SUBB A, Rn

Operation : (A) ← (A) - (C) - (Rn)

SUBB A, @Ri

Operation : (A) ← (A) - (C) - ((Ri))

SUBB A, direct

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

SUBB A, #data

Operation : (A) ← (A) - (C) - data

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

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

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

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

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

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

direct addr

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

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

immediate data

## INC <byte>

### Increment

INC A

Operation : (A) ← (A) + 1

INC Rn

Operation : (Rn) ← (Rn) + 1

INC @Ri

Operation : ((Ri)) ← ((Ri)) + 1

INC direct

Operation : (direct) ← (direct) + 1

INC DPTR

Operation : (DPTR) ← (DPTR) + 1

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

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

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

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

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

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

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

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

direct addr

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

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

# 부록 A : 명령어 Set (4/19)

DEC <byte>

## Decrement

DEC A

Operation : (A) ← (A) - 1

DEC Rn

Operation : (Rn) ← (Rn) - 1

DEC @Ri

Operation : ((Ri)) ← ((Ri)) - 1

DEC direct

Operation : (direct) ← (direct) - 1

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

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

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

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

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

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

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

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

direct addr

MUL AB

## Multiply

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

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

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

DIV AB

## Divide

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

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

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

# 부록 A : 명령어 Set (5/19)

DA A

## Decimal-adjust Accumulator for Addition

**Operation :**

```

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

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

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

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

## Logical AND for byte variables

ANL A, Rn

**Operation :** (A) ← (A) ^ (Rn)

ANL A, @Ri

**Operation :** (A) ← (A) ^ ((Ri))

ANL A, direct

**Operation :** (A) ← (A) ^ (direct)

ANL A, #data

**Operation :** (A) ← (A) ^ data

ANL direct, A

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

ANL direct, #data

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

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

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

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

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

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

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

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

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

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

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

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

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

# 부록 A : 명령어 Set (6/19)

**ANL C, <src-bit>**

## Logical AND for bit variables

ANL C, bit

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

ANL C, /bit

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

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

1 0 0 0 0 0 1 0

bit addr

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

1 0 1 1 0 0 0 0

bit addr

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

## Logical OR for byte variables

ORL A, Rn

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

ORL A, @Ri

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

ORL A, direct

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

ORL A, #data

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

ORL direct, A

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

ORL direct, #data

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

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

0 1 0 0 1 r r r

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

0 1 0 0 0 1 1 i

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

0 1 0 0 0 1 0 1

direct addr

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

0 1 0 0 0 1 0 0

immediate data

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

0 1 0 0 0 0 1 0

direct addr

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

0 1 0 0 0 0 1 1

direct addr

immediate data

# 부록 A : 명령어 Set (7/19)

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

## Logical OR for byte variables

ORL C, bit

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

ORL C, /bit

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

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

## Logical Exclusive-OR for byte variables

XRL A, Rn

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

XRL A, @Ri

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

XRL A, direct

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

XRL A, #data

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

XRL direct, A

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

XRL direct, #data

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

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

0 1 1 1 0 0 1 0

bit addr

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

1 0 1 0 0 0 0 0

bit addr

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

0 1 1 0 1 r r r

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

0 1 1 0 0 1 1 i

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

0 1 1 0 0 1 0 1

direct addr

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

0 1 1 0 0 1 0 0

immediate data

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

0 1 1 0 0 0 1 0

direct addr

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

0 1 1 0 0 0 1 1

direct addr

immediate Data

# 부록 A : 명령어 Set (8/19)

CLR A

## Clear Accumulator

Operation : (A)  $\leftarrow$  0

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

1 1 1 0 0 1 0 0

CLR <bit>

## Clear bit

CLR C

Operation : (C)  $\leftarrow$  0

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

1 1 0 0 0 0 1 1

CLR bit

Operation : (bit)  $\leftarrow$  0

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

1 1 0 0 0 0 1 0

bit addr

CPL A

## Complement Accumulator

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

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

1 1 1 1 0 1 0 0

CPL <bit>

## Complement bit

CPL C

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

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

1 0 1 1 0 0 1 1

CPL bit

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

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

1 0 1 1 0 0 1 0

bit addr

# 부록 A : 명령어 Set (9/19)

RL A

## Rotate Accumulator Left

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

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

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

RLC A

## Rotate Accumulator Left through the Carry flag

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

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

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

RR A

## Rotate Accumulator Right

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

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

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

RRC A

## Rotate Accumulator Right through the Carry flag

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

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

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

SWAP A

## Swap nibbles within the Accumulator

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

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

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

# 부록 A : 명령어 Set (10/19)

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

## Move byte variable

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

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

1 1 1 0 1 r r r

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

1 1 1 0 0 1 1 i

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

1 1 1 0 0 1 0 1

direct addr

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

0 1 1 1 0 1 0 0

immediate data

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

1 1 1 1 1 r r r

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

1 0 1 0 1 r r r

direct addr

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

0 1 1 1 1 r r r

immediate data

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

1 1 1 1 0 1 0 1

direct addr

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

1 0 0 0 1 r r r

direct addr

# 부록 A : 명령어 Set (11/19)

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

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

## Move bit data

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

# 부록 A : 명령어 Set (12/19)

**MOV DPTR, #data16**

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

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

**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)  $\leftarrow$  ((A) + (DPTR))

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

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

**MOVC A, @A + PC**

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

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

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

# 부록 A : 명령어 Set (13/19)

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

**Exchange Accumulator with byte variable**

XCH A, Rn

**Operation :** (A) ↔ (Rn)

XCH A, @Ri

**Operation :** (A) ↔ ((Ri))

XCH A, direct

**Operation :** (A) ↔ (direct)

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

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

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

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

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

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

**XCHD A, @Ri**

**Exchange Digit**

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

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

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

**PUSH direct**

**Push onto stack**

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

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

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

**POP direct**

**Pop onto stack**

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

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

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

# 부록 A : 명령어 Set (14/19)

**SETB** <bit>

**Set bit**

**SETB** C

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

**SETB** bit

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

**JC** rel

**Jump if Carry is set**

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

**JNC** rel

**Jump if Carry is not set**

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

**JB** bit, rel

**Jump if Bit is set**

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

**JNB** bit, rel

**Jump if Bit is not set**

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

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

1 1 0 1 0 0 1 1

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

1 1 0 1 0 0 1 0

bit addr

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

0 1 0 0 0 0 0 0

relative addr

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

0 1 0 1 0 0 0 0

relative addr

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

0 0 1 0 0 0 0 0

bit addr

relative addr

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

0 0 1 1 0 0 0 0

bit addr

relative addr

# 부록 A : 명령어 Set (15/19)

**JBC bit, rel**

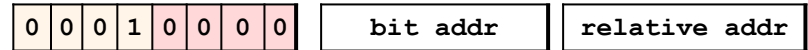
**Jump if Bit is set and Clear bit**

**Operation :**

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

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

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



**ACALL addr11**

**Absolute Subroutine Call**

**Operation :**

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

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

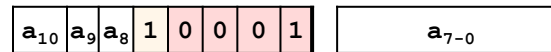
$$((SP)) \leftarrow (PC_{7-0})$$

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

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

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

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



**LCALL addr16**

**Long Subroutine Call**

**Operation :**

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

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

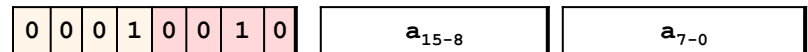
$$((SP)) \leftarrow (PC_{7-0})$$

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

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

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

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



# 부록 A : 명령어 Set (16/19)

## RET

### Return from Subroutine

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

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

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

## RETI

### Return from Interrupt

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

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

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

## AJMP addr11

### Absolute Jump

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

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

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

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

## LJMP addr16

### Long Jump

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

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

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

# 부록 A : 명령어 Set (17/19)

**JMP @A + DPTR**

**Jump Indirect Relative to the DPTR**

**Operation :** (PC) ← (A) + (DPTR)

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

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

**JZ rel**

**Jump if Accumulator is Zero**

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

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

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

relative addr
---------------

**JNZ rel**

**Jump if Accumulator is Not Zero**

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

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

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

relative addr
---------------

# 부록 A : 명령어 Set (18/19)

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

## Compare and Jump if Not Equal

**CJNE A, direct, rel**

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

**CJNE A, #data, rel**

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

**CJNE Rn, #data, rel**

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

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

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

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

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

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

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

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

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

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

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

# 부록 A : 명령어 Set (19/19)

DJNZ <byte>, rel

## Decrement and Jump if Not Zero

DJNZ Rn, rel

**Operation :**  
 (PC) ← (PC) + 2  
 (Rn) ← (Rn) - 1  
 If (Rn) ≠ 0, then (PC) ← (PC) + rel

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

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
---	---	---	---	---	---	---	---

## [SFR 설명을 읽는 방법]



### ■ **P0** (80h) : Port 0 레지스터

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

- ◆ Port 0 레지스터

### ■ **SP** (81h) : 스택 포인터 레지스터

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)

- ◆ 스택이 지장하는 곳을 지시함
- ◆ PUSH에 의해 증가하고 POP에 의해 감소함.

### ■ **DPL** (82h) : 데이터 포인터 하위 레지스터

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) : 데이터 포인터 상위 레지스터

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

### ■ **PCON** (87h) : 전력 제어 레지스터

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

- ◆ SMOD1 : UART 모드 1에서 타이머 1 보레이트 두 배.
- ◆ POF : 전원 꺼짐 플래그.  
전원이 켜질 때, 이 비트는 하드웨어에 의해 설정됨.
- ◆ GF1, GF0: 범용 플래그.
- ◆ PD : 정지 모드.
- ◆ IDL : IDLE 모드.

## ■ TCON (88h) : Timer/Counter 0/1 제어 레지스터

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 오버플로우.
- ◆ TR1 : Timer 1 동작.
- ◆ TF0 : Timer 0 오버플로우.
- ◆ TR0 : Timer 0 동작.
- ◆ IE1 : 외부 인터럽트 1 플래그.  
IT1 = 0이면, 소프트웨어에 의해 소거됨.  
IT1 = 1이면, 인터럽트 처리 루틴을 갈 때 자동소거.
- ◆ IT1 : 외부 인터럽트 1 촉발 방식 선택.  
Edge 촉발(IT1=1) / Level 촉발(IT1=0; 기본지정)
- ◆ IE0 : 외부 인터럽트 0 플래그.  
IT0 = 0이면, 소프트웨어에 의하여 소거됨.  
IT0 = 1이면, 인터럽트 처리 루틴으로 갈 때 자동소거.
- ◆ IT0 : 외부 인터럽트 0 촉발 방식 선택.  
Edge 촉발 (IT0=1) / Level 촉발(IT0=0; 기본지정)

## ■ TMOD (89h) : Timer/Counter 0 모드 제어 레지스터

-	-	-	-	GATE	C/T	M1	M0
---	---	---	---	------	-----	----	----

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

- ◆ GATE[3] : Timer 0 gate 제어.
- ◆ C/T[2] : Timer 0 Counter/Timer 선택  
0 =  $F_{osc}/12$ 을 계수하는 타이머. (기본지정)  
1 = T0 pin의 신호를 계수하는 카운터.
- ◆ M1, M0 : Timer 0 모드 선택.  
[0,0] : 모드0, 13-bit T/C  
[0,1] : 모드1, 16-bit T/C  
[1,0] : 모드2, 8-bit T/C, 자동재입력  
[1,1] : 모드3, Two 8-bit T/C

## ■ TLO (8Ah) : Timer/Counter 0 하위 Byte 레지스터

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 하위 Byte 레지스터

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 상위 Byte 레지스터

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 상위 Byte 레지스터

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

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

## ■ P1 (90h) : Port 1 레지스터

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

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

- ◆ P1.0 : XTAL 1 pin의 alternate 기능
- ◆ P1.1 : XTAL 2 pin의 alternate 기능
- ◆ P1.2 : RESETB pin의 alternate 기능

## ■ EXIF (91h) : 외부 인터럽트 플래그 레지스터

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

- ◆ IE3 : 외부 인터럽트 3 플래그. 소프트웨어에 의해 소거됨.
- ◆ IE2 : 외부 인터럽트 2 플래그. 소프트웨어에 의해 소거됨.
- ◆ XT/RG : 시스템 클럭 선택  
0 = 시스템 클럭으로 내부 링 발진기 선택.  
1 = 시스템 클럭으로 외부 클럭 선택.
- ◆ RGMD : 링 모드. 시스템 클럭은 링 또는 XTAL이다.  
일반적으로 RGMD은 XT/RG의 반대다.
- ◆ RGSL : 정지 모드를 탈출할 때 링 발진기 선택 비트.  
1 = 정지 모드를 탈출할 때, 65,536 XTAL 클럭 주기 동안 시스템 클럭으로 링 발진기를 사용한다.
- ◆ BGS : Band-gap 선택. (기본지정 = 1)  
0 = Band-gap block(LVD)은 정상 모드에서는 동작하지만 정지 모드에서는 동작하지 않는다.  
정지 모드에서 크게 전력소모를 줄인다.  
1 = Band-gap block(LVD)이 정지 모드에서 동작한다.

## ■ SCON (98h) : UART0의 직렬 단자 레지스터

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

- ◆ REN : 직렬 수신 인에이블.
- ◆ TI : 송신 인터럽트 플래그. 소프트웨어에 의해 소거.
- ◆ RI : 수신 인터럽트 플래그. 소프트웨어에 의해 소거.

## ■ SBUF (99h) : 직렬 데이터 버퍼 레지스터

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)

- ◆ 송신 버퍼와 수신 버퍼는 분리되어 있다.
- ◆ 읽기와 쓰기 주소는 같다.

## ■ P2 (A0h) : Port 2 레지스터

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

- ◆ Port 2 레지스터

## ■ IE (A8h) : 인터럽트 인에이블 레지스터

EA	EADC	-	ES	ET1	EX1	ET0	EX0
R/W(0)	R/W(0)		R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- ◆ EA : 전체 인터럽트 인에이블.
- ◆ EADC : ADC 인터럽트 인에이블.
- ◆ ES : 직렬 단자 인터럽트 인에이블.
- ◆ ET1 : Timer 1 인터럽트 인에이블.
- ◆ EX1 : 외부 인터럽트 1 인에이블.
- ◆ ET0 : Timer0 i인터럽트 인에이블.
- ◆ EX0 : 외부 인터럽트 0 인에이블.

## ■ IP (B8h) : 인터럽트 우선순위 레지스터

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

- ◆ PADC : ADC 인터럽트 우선순위.
- ◆ PS : Serial port 인터럽트 우선순위.
- ◆ PT1 : Timer 1 인터럽트 우선순위.
- ◆ PX1 : 외부 인터럽트 1 우선순위.
- ◆ PT0 : Timer 0 인터럽트 우선순위.
- ◆ PX0 : 외부 인터럽트 0 우선순위.

## ■ OSCICN (BEh) : 내부 링 발진기 제어 레지스터

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

- ◆ RINGON : 1 = 내부 링 발진기 동작  
0 = 내부 링 발진기 정지.  
XTRG = 0일 때 RINGON bit를 소거하지 말아야 한다.
- ◆ DIV1, DIV0 : 링 발진기 분주.  
[0,0] = 3.65MHz/1  
[0,1] = 3.65MHz/2  
[1,0] = 3.65MHz/4  
[1,1] = 3.65Mhz/8

## ■ PMR (C4h) : 전력 관리 제어 레지스터

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

- ◆ XTOFF : 외부 크리스탈 발진기를 위해서 내부 증폭기 디세이블.  
1 = 외부 크리스탈 정지.  
0 = 외부 크리스탈 동작 (기본지정).  
XT/RG = 1일 때 XTOFF bit를 설정하지 말아야 한다.

## ■ STATUS (C5h) : 크리스탈 상태 레지스터

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

- ◆ XTUP : 크리스탈 발진기 정상 동작 상태.  
크리스탈 클럭이 안정화되었거나(1) 아닌 상태(0)를 나타냄.  
모든 종류의 reset때 하드웨어에 의해 설정됨.  
XTOFF bit가 설정될 때, 하드웨어에 의해 소거됨.  
XT/RG = 1일 때, 정지 모드를 탈출하는 동안 소거됨.  
크리스탈이 안정화되는 시간이 지나면 하드웨어가 설정함.

## ■ PSW (D0h) : Program Status Word 레지스터

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 : 캐리 플래그.
- ◆ AC : 보조 캐리 플래그.
- ◆ F0 : 사용자 플래그 0.
- ◆ RS1, RS0 : Register bank 선택  
[0,0] : Bank 0 [1,0] : Bank 2  
[0,1] : Bank 1 [1,1] : Bank 3
- ◆ OV : 오버플로우 플래그.
- ◆ F1 : 사용자 플래그 1.
- ◆ P : 패리티 bit. ACC 기수 패리티를 따라 하드웨어가 설정/소거함.

## ■ P0TYPE (D4h) : Port 0 출력 방식 제어 레지스터

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

- ◆ 0 = Push-pull 출력(기본지정)

## ■ P1TYPE (D5h) : Port 1 출력 방식 제어 레지스터

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

◆ 0 = Push-pull 출력 (기본지정)

## ■ P2TYPE (D6h) : Port 2 출력 방식 제어 레지스터

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

◆ 0 = Push-pull 출력 (기본지정)

## ■ WDCON (D8h) : Watchdog Timer 제어 레지스터

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

- ◆ WD1, WD0 : Watchdog timer 모드선택
  - [0,0] :  $1 \times 2^{16}$  clocks (interrupt) + 256 clocks (reset)
  - [0,1] :  $4 \times 2^{16}$  clocks (interrupt) + 256 clocks (reset)
  - [1,0] :  $16 \times 2^{16}$  clocks (interrupt) + 256 clocks (reset)
  - [1,1] :  $32 \times 2^{16}$  clocks (interrupt) + 256 clocks (reset)
- ◆ WDIF : Watchdog timer 인터럽트 플래그.
- ◆ WTRF : Watchdog timer reset 플래그. 소프트웨어에 의해 소거됨.
- ◆ EWT : Watchdog timer reset 인에이블.
- ◆ RWT : Watchdog 타이머를 재출발.

## ■ PWMCON (DCh) : PWM 제어 레지스터

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

- ◆ POSEL : PWM 파형을 P0.6으로 출력 인에이블.
- ◆ PS2\_P0, PS1\_P0, PS0\_P0 : 선분주 클럭 선택.
  - [0,0,0] =  $F_{osc}/1$ , [0,0,1] =  $F_{osc}/2$ , [0,1,0] =  $F_{osc}/4$ ,
  - [0,1,1] =  $F_{osc}/8$ , [1,0,0] =  $F_{osc}/16$ , [1,0,1] =  $F_{osc}/32$ ,
  - [1,1,0] =  $F_{osc}/64$ , [1,1,1] =  $F_{osc}/128$

\*  $F_{osc}/10$ 이 아니면 PWM Clock ( $F_{PWM}$ )을 ADC에 사용 가능.
- ◆ PWMF : PWM 인터럽트 플래그. 소프트웨어에 의해 소거됨.
- ◆ CLR\_P0 : PWM counter reset. 하드웨어에 의해 소거됨.
- ◆ RUN\_P0 : PWM counter 동작.

## ■ PWMD (DEh) : PWM Duty 데이터 레지스터

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

## ■ ACC/A (E0h) : Accumulator

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

## ■ ADCSELH (E1h) : ADC 채널 선택 상위 레지스터

ADC11B	ADC10B	ADC9B	ADC8B	ADC7B	ADC6B	ADC5B	ADC4B
--------	--------	-------	-------	-------	-------	-------	-------

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

- ◆ ADC11B : 0 = P2.2로 ADC11 입력 인에이블 & 디지털 입력 디세이블.
- ◆ ADC10B : 0 = P2.3으로 ADC10 입력 인에이블 & 디지털 입력 디세이블.
- ◆ ADC9B : 0 = P2.4로 ADC9 입력 인에이블 & 디지털 입력 디세이블.
- ◆ ADC8B : 0 = P2.5로 ADC8 입력 인에이블 & 디지털 입력 디세이블.
- ◆ ADC7B : 0 = P2.6으로 ADC7 입력 인에이블 & 디지털 입력 디세이블.
- ◆ ADC6B : 0 = P0.7로 ADC6 입력 인에이블 & 디지털 입력 디세이블.
- ◆ ADC5B : 0 = P0.6으로 ADC5 입력 인에이블 & 디지털 입력 디세이블.
- ◆ ADC4B : 0 = P0.5로 ADC4 입력 인에이블 & 디지털 입력 디세이블.

## ■ ADCSEL (E2h) : ADC 채널 선택하위 & MUX 선택 레지스터

ADC3B	ADC2B	ADC1B	ADC0B	CH3	Ch2	CH1	CH0
-------	-------	-------	-------	-----	-----	-----	-----

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

- ◆ ADC3B : 0 = P0.4로 ADC3 / AV<sub>REF</sub> 입력 인에이블 & 디지털 입력 디세이블
  - ◆ ADC2B : 0 = P0.3로 ADC2 입력 인에이블 & 디지털 입력 디세이블
  - ◆ ADC1B : 0 = P0.2로 ADC1 입력 인에이블 & 디지털 입력 디세이블
  - ◆ ADC0B : 0 = P0.1으로 ADC0 입력 인에이블 & 디지털 입력 디세이블
  
  - ◆ CH[3:0] : ADC MUX 선택
    - [0,0,0,0] = ADC0 선택 (=0h)
    - [0,0,0,1] = ADC1 선택 (=1h)
    - [0,0,1,0] = ADC2 선택 (=2h)
    - :
    - [1,0,1,1] = ADC11 선택 (=Bh)
- \* Ch, Dh, Eh, Fh : 모든 ADC 입력 디세이블

## ■ ALTSEL (E3h) : Alternate 기능 제어레지스터

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

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

- ◆ IOXEN : 1 = XTAL과 XTAL2을 범용 입출력으로 사용함.  
XTOFF (PMR.3) = 1이어야 함 (발진기 증폭기 꺼짐)
- ◆ IORSTEN : 1 = RESETB을 범용 입출력으로 사용함.
- ◆ CLO : 1 = P2.6으로 시스템 클럭 출력.
- ◆ PWMD0 : 1 = P0.0로 PWM 파형 출력 인에이블.
- ◆ TVO : 1 = P0.0로 Timer 0 오버플로우 클럭 출력.
- ◆ TX : 1 = P0.2로 UART TX 데이터 출력.  
UART를 사용하기 위해서 TX.bit를 설정해야 한다.

## ■ POSEL (E4h) : Port 0 Pull-up 제어 레지스터

POSEL.7	POSEL.6	POSEL.5	POSEL.4	POSEL.3	POSEL.2	POSEL.1	POSEL.0
---------	---------	---------	---------	---------	---------	---------	---------

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

- ◆ 0 = 내부 Pull-up 저항 ON (기본지정)
- ◆ 1 = ADC\_EN (ADCON[7]) = 1일 때, 내부 Pull-up 저항 OFF

## ■ P1SEL (E5h) : Port 1 Pull-up 제어 레지스터

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

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

- ◆ 0 = 내부 Pull-up 저항 ON / 1 = OFF (기본지정)

## ■ P2SEL (E6h) : Port 2 Pull-up 제어 레지스터

	P2SEL.6	P2SEL.5	P2SEL.4	P2SEL.3	P2SEL.2	P2SEL.1	P2SEL.0
--	---------	---------	---------	---------	---------	---------	---------

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

- ◆ 0 = 내부 Pull-up 저항 ON (기본지정)
- ◆ 1 = ADC\_EN (ADCON[7]) = 1일 때, 내부 Pull-up 저항 OFF

## ■ EIE (E8h) : 확장된 인터럽트 인에이블 레지스터

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

- ◆ EPWM : PWM 인터럽트 인에이블.
- ◆ EWDT : Watchdog 인터럽트 인에이블.
- ◆ EX3 : 외부 인터럽트 3 인에이블.
- ◆ EX2 : 외부 인터럽트 2 인에이블.

## ■ ADCR (EEh) : ADC 결과 상위 레지스터 : Value[9:2]

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

## ■ ADCON (EFh) : ADC 제어 & ADC 결과 하위 레지스터 : Value[1:0]

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

- ◆ AD\_EN : ADC 변환 인에이블.
- ◆ AD\_REQ : ADC 변환 요구.  
AD\_END가 0에서 1로 변할 때 하드웨어가 소거함.
- ◆ AD\_END : 현재 ADC 상태.  
0 = ADC 동작 중.
- ◆ ADCF : ADC 인터럽트 플래그. 소프트웨어에 의해 소거됨.
- ◆ AVREF : 1 = P0.4로 ADC 기준전압 입력 인에이블.
- ◆ ADIV : ADC 입력 클럭 선택  
0 = 시스템 클럭( $F_{osc}$ ) / 2. (기본지정)  
1 = PWM 입력 클럭( $F_{pwm}$ )
- ◆ SAR1, SAR0 : ADC 결과값의 하위 비트 2개.

## ■ B (F0h) : B 레지스터

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

## ■ PODIR (F4h) : Port 0 입출력 제어 레지스터

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

- ◆ 1 = 입력(기본지정) / 0 = 출력

## ■ P1DIR (F5h) : Port 1 입출력 제어 레지스터

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

- ◆ 1 = 입력(기본지정) / 0 = 출력

## ■ P2DIR (F6h) : Port 2 입출력 제어 레지스터

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

- ◆ 1 = 입력(기본지정) / 0 = 출력

## ■ EIP (F8h) : 확장된 인터럽트 우선순위 레지스터

-	-	PPWM	PWDT	-	-	PX3	PX2
		R/W(0)	R/W(0)			R/W(0)	R/W(0)

- ◆ PPWM : PWM 인터럽트 우선순위 bit.
- ◆ PWDT : Watchdog timer 인터럽트 우선순위 bit.
- ◆ PX3 : 외부 인터럽트 3 우선순위 bit.
- ◆ PX2 : 외부 인터럽트 2 우선순위 bit.

## 부록 C : Update History

- ◆ V2.0
  - ✓ Updating the Marking of Package
- ◆ V2.1
  - ✓ Remove open-drain output
  - ✓ Change RING frequency specification
    - 4MHz @ 5V → 3.65MHz @ 5V
- ◆ V2.2
  - ✓ Updating Power Management
    - IDLE Mode, Change wakeup attribute.
- ◆ V2.3
  - ✓ Change RING frequency specification
    - 3.65MHz @ 5V → 3.65MHz (15%) @ 5V
  - ✓ Add on the ESD Structure of Pads slide
  - ✓ Update the External Reset slide.
  - ✓ Update the 'On-Chip POR' slide.
  - ✓ Update the Package Dimensions.
  - ✓ Update the Product Numbering System.
- ◆ V2.4
  - ✓ Modify the Supply voltage.
    - 2.4V ~ 5.5V
  - ✓ Modify the Operating Temperature.
    - -40 °C to 120 °C
- ◆ V2.5
  - ✓ Modify the 8-SOIC Package Dimension.
- ◆ V2.6
  - ✓ Add the 16-TSSOP Package.
  - ✓ Remove the 14-SPDIP/SOIC Package.
- ◆ V2.7
  - ✓ Add on the Power Slope slide.
  - ✓ Feedback Pull-up Issue : Update PnSEL Control
  - ✓ Update 'On-Chip POR' slide
- ◆ V2.8
  - ✓ Describes the initial port state.
- ◆ V2.9
  - ✓ Modify the package types.