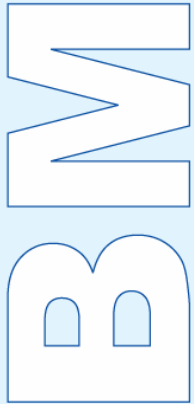




# MiDAS Family

BM-MiDAS1.0-V1.9



## Brief Manual of MiDAS1.0 Family

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

V1.9

November 2005

- ◆ CORERIVER Semiconductor reserves the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice.
- ◆ Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.
- ◆ The CORERIVER Semiconductor products listed in this document are intended for usage in general electronics applications. These CORERIVER Semiconductor 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 개관

2. 특징

3. 블록 도표

4. 핀 구성

5. 핀 기능

6. 기능 설명

✓ CPU 설명

- 메모리 구조
- SFR Map과 설명
- 명령어 셋 요약
- CPU 타이밍

✓ 주변회로 설명

- 입출력 단자
- LVD (저전압 검출기)
- WDT (Watchdog 타이머)
- 타이머 0/1/2
- UART (보편적 비동기 RX/TX)
- PWM (펄스 폭 변조기)
- ADC
- 인터럽트
- 리셋 회로
- 클럭 회로
- 전력 조절
- EPROM

7. Intel 80C52에 비하여 강점

8. Absolute Maximum Ratings

9. DC 특성

10. AC 특성

11. ADC 특성

12. Package Dimensions

13. 제품 번호 체계

14. 지원 툴

◆ 부록

- A. 명령어 셋
- B. SFR 설명
- C. Update History

## 1. 제품 개관

- ◆ **CORERIVER의 MiDAS1.0 family**는 고속의 **80C52** 호환성 **microcontroller** 군이다.
- ◆ 명령어 수행 시간은 **Intel 80C52**에 비하여 최대 **3배** 빠르다.
  - ✓ 1 기계어 주기 = 4 클럭 대 12 클럭
- ◆ **MiDAS1.0 family**의 추가적인 주변회로들:
  - ✓ 9bit ADC / 8bit PWM / WDT / LVD / POR.
- ◆ 전력소모를 줄이기 위한 전력 절약 모드들
- ◆ 잡음 저항성 구조
- ◆ 사용자 친근 **MDS** 환경 제공
- ◆ 사용하기 쉬운 **training-kit system**

# 1. 제품 개관 (계속)

## A. MiDAS1.0 Family - GC80C520G 계열 (일반 MCU)

제품	Mask-ROM (byte)	EPROM (byte)	RAM (Byte)	전압 (V)	주파수 (MHz)	T/C (16bits)	직렬 입출력	WDT	ADC (bit x ch)	PWM (bit x ch)	입출력 핀	Package	기타	사용가능 시기
GC87C520G0-PL44I GC87C520G0-LQ44I GC87C520G0-P40I GC87C520G0-SP28I GC87C520G0-SO28I	-	8K	256	2.7~5.5	40 (20)	3	1 UART	YES	-	-	36 36 32 22 22	44-PLCC 44-MQFP 40-PDIP 28-SPDIP 28-SOIC	LVD POR	<b>Now</b> <b>Now</b> <b>Now</b> <b>Now</b> <b>Now</b>
GC81C520G0-PL44I GC81C520G0-LQ44I GC81C520G0-P40I GC81C520G0-SP28I GC81C520G0-SO28I	8K	-	256	2.7~5.5	40 (20)	3	1 UART	YES	-	-	36 36 32 22 22	44-PLCC 44-MQFP 40-PDIP 28-SPDIP 28-SOIC	LVD POR	<b>Now</b> <b>Now</b> <b>Now</b> <b>Now</b> <b>Now</b>
GC80C520G0-PL44I GC80C520G0-LQ44I GC80C520G0-P40I	ROMless		256	2.7~5.5	40 (20)	3	1 UART	YES	-	-	36 36 32	44-PLCC 44-MQFP 40-PDIP	LVD POR	<b>Now</b> <b>Now</b> <b>Now</b>

\* MiDAS family의 동작 주파수는 5V에서 40MHz이다.

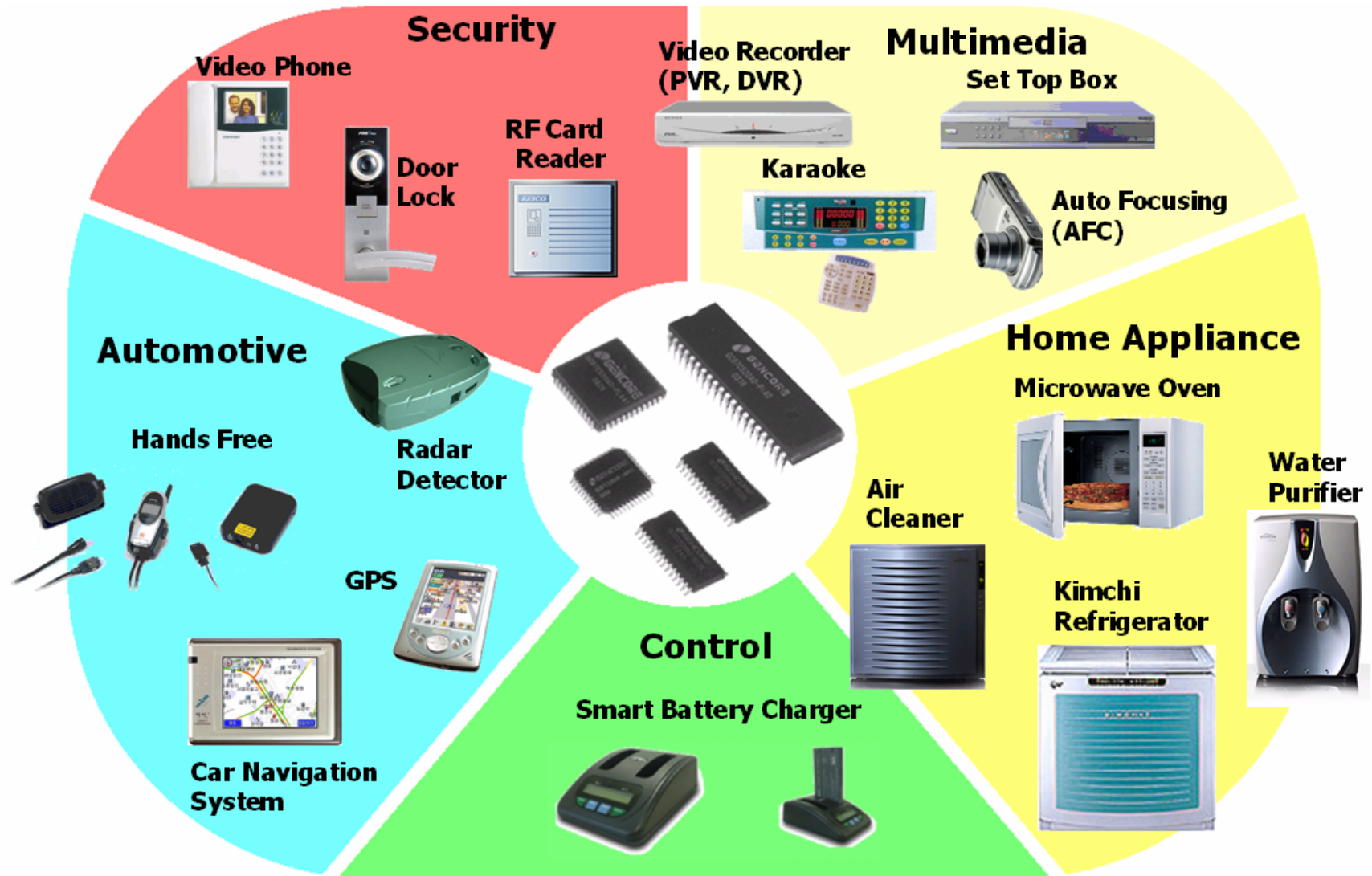
# 1. 제품 개관 (계속)

## B. MiDAS1.0 Family - GC80C520A 계열 (ADC 응용 MCU)

제품	Mask-ROM (byte)	EPROM (byte)	RAM (Byte)	전압 (V)	주파수 (MHz)	T/C (16bits)	직렬 입출력	WDT	ADC (bit x ch)	PWM (bit x ch)	입출력 핀	Package	기타	사용가능 시기
GC87C520A0-PL44I GC87C520A0-LQ44I GC87C520A0-P40I GC87C520A0-SP28I GC87C520A0-SO28I	-	8K	256	2.7~5.5	40 (20)	3	1 UART	YES	9x4	8x2	36 36 32 22 22	44-PLCC 44-MQFP 40-PDIP 28-SPDIP 28-SOIC	LVD POR	Now Now Now Now Now
GC81C520A0-PL44I GC81C520A0-LQ44I GC81C520A0-P40I GC81C520A0-SP28I GC81C520A0-SO28I	8K	-	256	2.7~5.5	40 (20)	3	1 UART	YES	9x4	8x2	36 36 32 22 22	44-PLCC 44-MQFP 40-PDIP 28-SPDIP 28-SOIC	LVD POR	Now Now Now Now Now
GC80C520A0-PL44I GC80C520A0-LQ44I GC80C520A0-P40I	ROMless		256	2.7~5.5	40 (20)	3	1 UART	YES	9x4	8x2	36 36 32	44-PLCC 44-MQFP 40-PDIP	LVD POR	Now Now Now

\* MiDAS 1.0 family의 동작 주파수는 5.0V에서 40MHz이다.

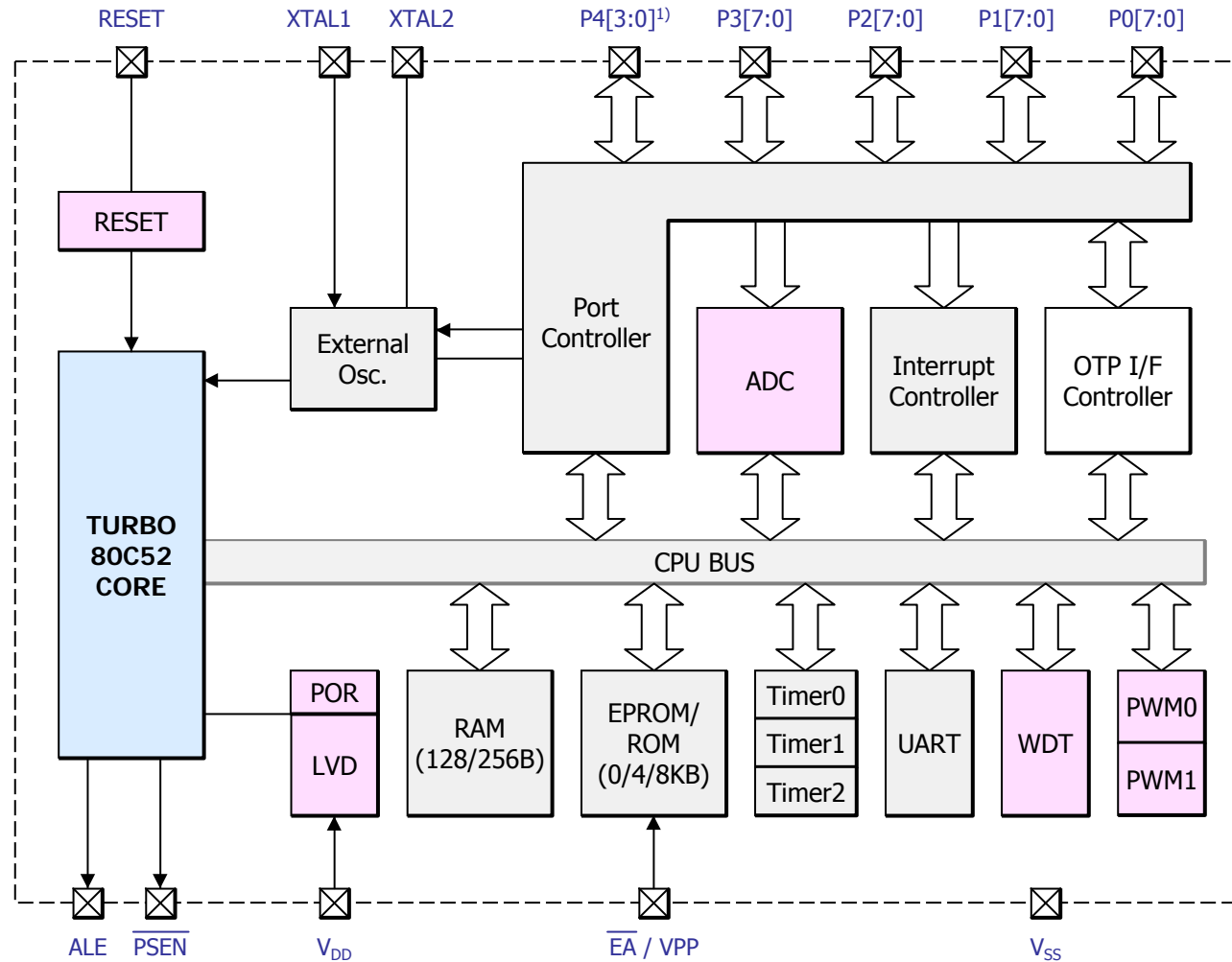
# CORERIVER MCU 군들의 응용



## 2. 특징

- ◆ CPU
  - ✓ 8-bit turbo 80C52 구조
  - ✓ 4 클럭 / 1 기계어 주기
  - ✓ Intel 80C52와 핀/명령어 수준 호환
- ◆ 0/4/8Kbytes 프로그램 가능한 ROM
- ◆ 128/256bytes RAM
- ◆ 넓은 범위의 2.7V에서 5.5V까지 전원 공급
- ◆ 동작 주파수
  - ✓ 최대 40MHz @4.5V ~ 5.5V
  - ✓ 최대 20MHz @2.7V ~ 3.3V
- ◆ 동작 온도 : -20 °C to 85 °C
- ◆ 32/36개의 입출력 핀을 완전히 program할 수 있음
  - ✓ 기본설정으로 Intel 8052와 호환
  - ✓ 입출력을 선택적으로 조정
  - ✓ TTL 입력 레벨과 CMOS 호환 logic 레벨
- ◆ EMI 삭감 모드 : 선택적인 ALE 금지
- ◆ 저전압 검출 회로
- ◆ 27-bit 프로그램이 가능한 Watchdog 타이머
- ◆ 16-bit 타이머/카운터 3개
- ◆ 전이중 UART
  - ✓ 자동 주소 인식: 멀티 프로세서 간의 통신
- ◆ 2-channel 8-bit 고속 PWM
- ◆ 4-channel 9-bit ADC
  - ✓ 초당 최대 100K sample (@8 MHz)
  - ✓ 입력 클럭 주파수를 프로그램으로 조정 가능
- ◆ 13 인터럽트 발생원 (6개의 외부 발생원)
  - ✓ 타이머 0/1/2, UART, ADC, WDT, LVD, 6 외부
  - ✓ 4 단계의 인터럽트 우선순위 & NMI
- ◆ 리셋 방법
  - ✓ 칩상의 파워 온 리셋
  - ✓ 외부 리셋
  - ✓ 저전압 검출 리셋
  - ✓ 선택적인 Watchdog 타이머 리셋
- ◆ 전력 소모
  - ✓ 활동 전류 : 최대 20mA @5V, 40MHz
  - ✓ 정지 전류 : 최대 1uA
- ◆ E.S.D. protection up to 2,000V
- ◆ Latch-up protection up to ±200mA
- ◆ Package :
  - ✓ 44-PLCC, 44-MQFP, 40-PDIP, and 28-SPDIP/SOIC

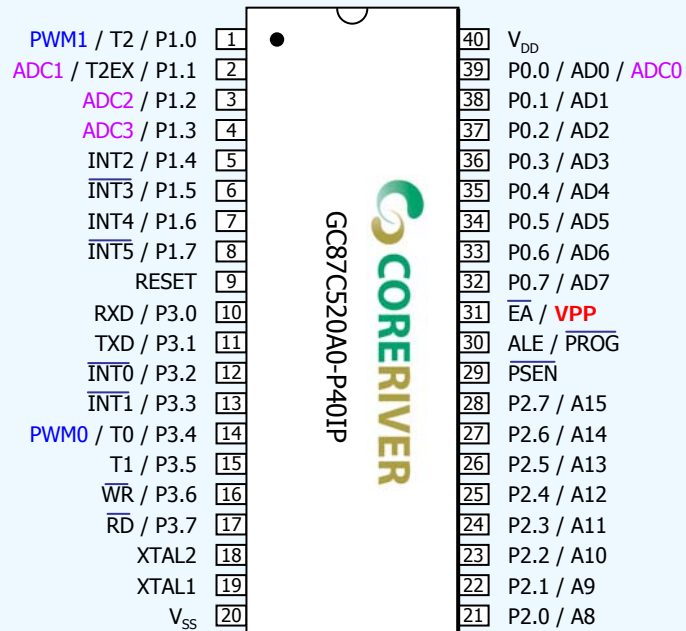
### 3. Block Diagram



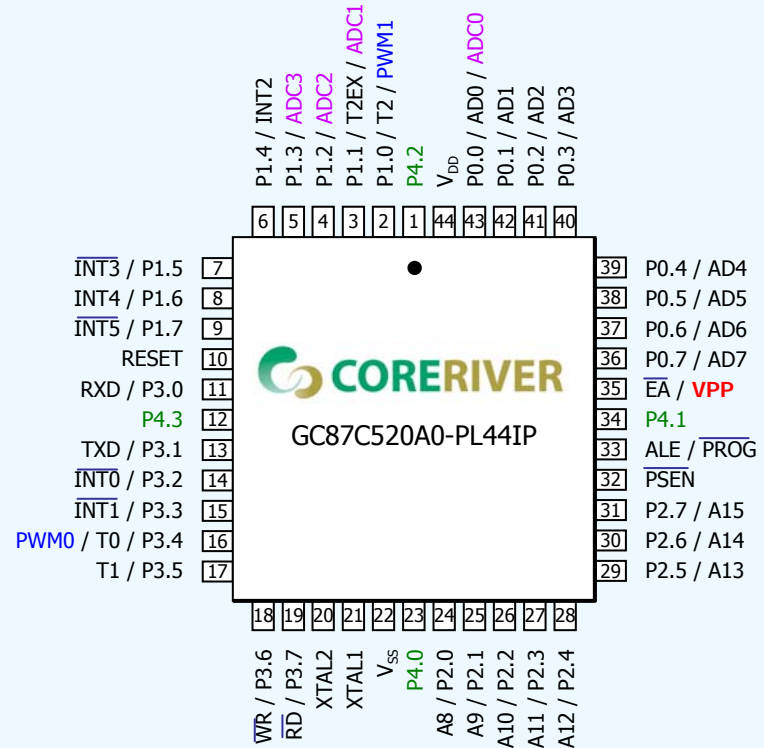
1) P4[3:0]는 오직 44-PLCC와 44-LQFP type Package의 경우에 사용할 수 있다.



## 4. Pin 구조

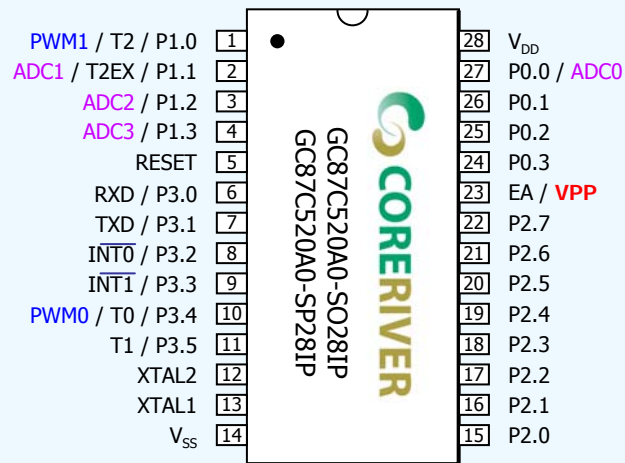


[ 40-PDIP ]

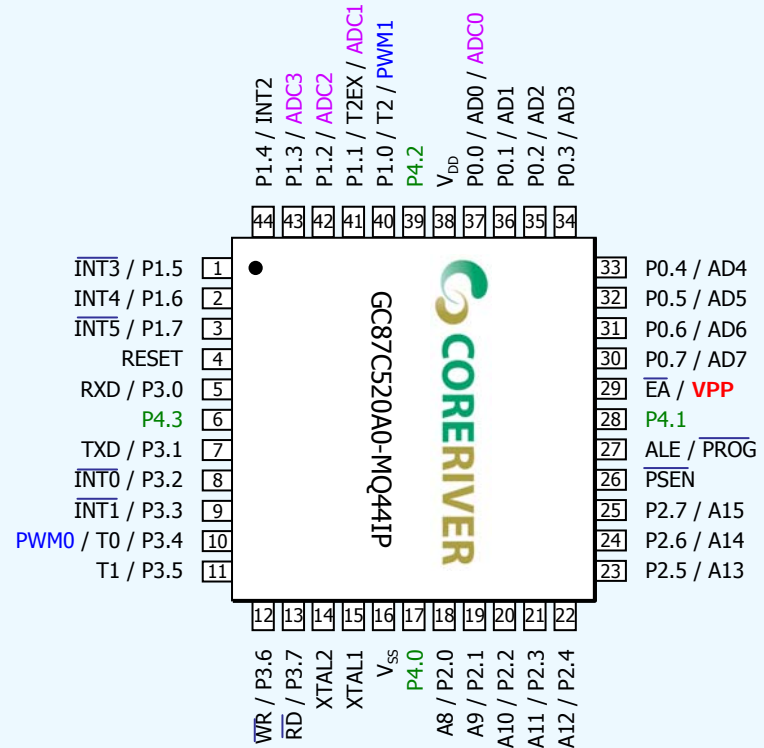


[ 44-PLCC ]

## 4. Pin 구조 (계속)



[ 28-SPDIP/SOIC ]

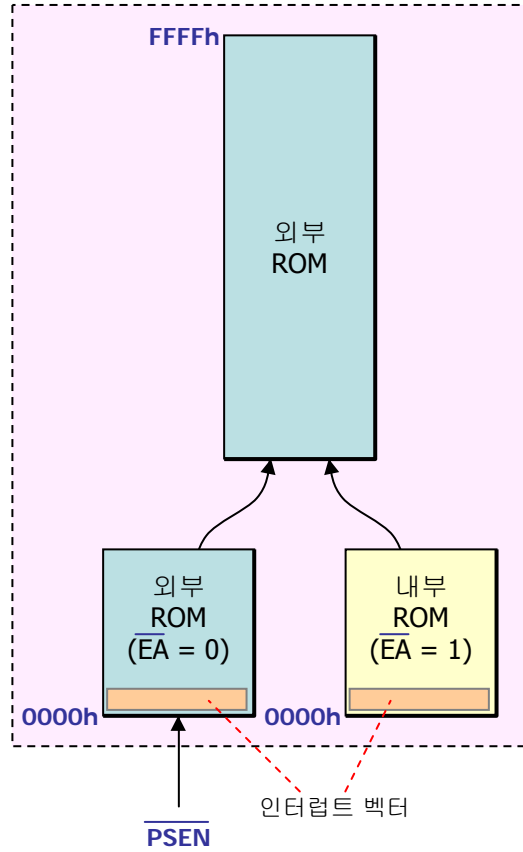


[ 44-MQFP ]

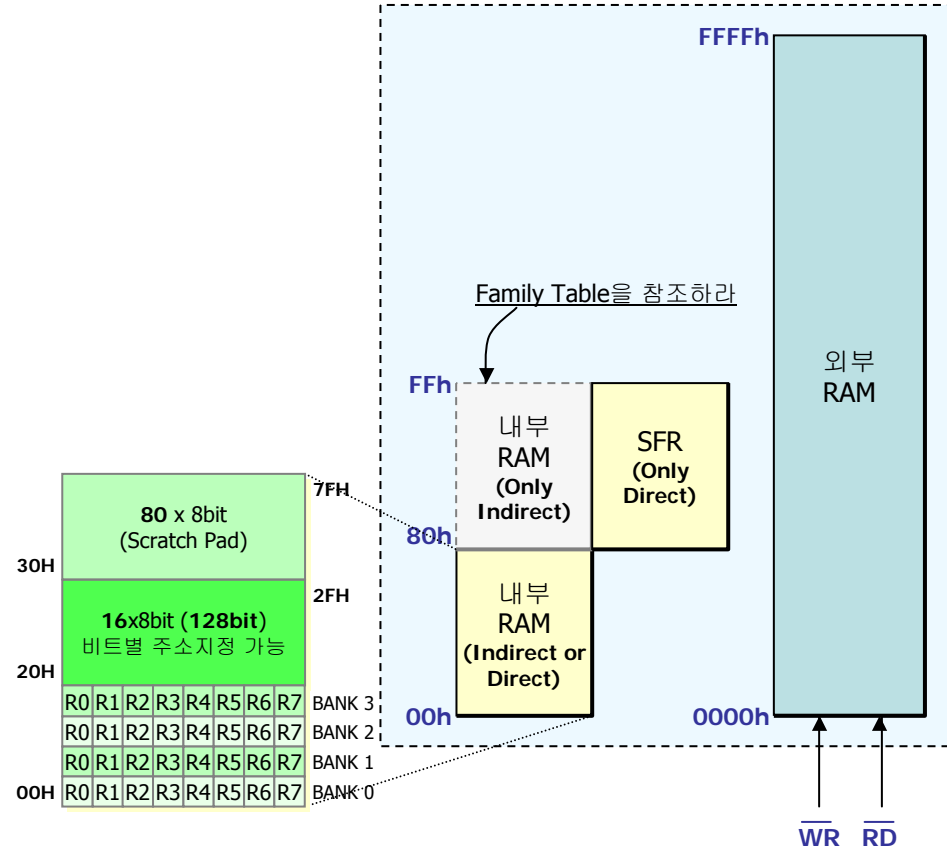
## 5. 핀 기능

부호	방향	설명	공유 핀
V <sub>DD</sub>	입력	전력 전압	-
V <sub>SS</sub>	입력	전력 그라운드	-
RESET	입력	외부 리셋	-
XTAL1	입력	외부 Crystal 입력	-
XTAL2	출력	외부 Crystal 출력	-
$\overline{EA}$	입력	외부 ROM Access Enable ( $\overline{EA}=0 \rightarrow$ 내부 Rom Access Disable, $\overline{EA}=1 \rightarrow$ 외부) 이 핀은 떠 있어서는 안된다.	VPP
$\overline{ALE}$	입출력	주소 래치 허용	$\overline{PROG}$
PSEN	입출력	Program Strobe 허용 (외부 ROM 출력 허용 신호)	-
P0[7:0]	입출력	Schmitt Trigger 입력 또는 Push-pull/Open-Drain 출력으로 프로그램 가능한 입출력 단자. Pull-up 저항은 소프트웨어로 제어할 수 있다.	P0.0 / AD0 / ADC0 P0.1~P0.7 / AD1~7
P1[7:0]	입출력	Schmitt Trigger 입력 or Push-pull 출력으로 프로그램이 가능한 입출력 단자. Pull-up 저항은 소프트웨어로 제어할 수 있다.	P1.0 / T2 / PWM1 P1.1 / T2EX / ADC1 P1.2~P1.3 / ADC2~ADC3 P1.4~P1.7 / INT2~INT5
P2[0:7]	입출력	Schmitt Trigger 입력 or Push-pull 출력으로 프로그램이 가능한 입출력 단자. Pull-up 저항은 소프트웨어로 제어할 수 있다.	P2.0~P2.7 / A8~A15
P3[7:0]	입출력	Schmitt Trigger 입력 or Push-pull 출력으로 프로그램이 가능한 입출력 단자. Pull-up 저항은 소프트웨어로 제어할 수 있다.	P3.0 / RXD P3.1 / TXD P3.2~P3.3 / INT0~INT1 P3.4 / T0 / PWM0 P3.5 / T1 P3.6 / WR P3.7 / RD
P4[3:0]	입출력	Schmitt Trigger 입력 or Push-pull 출력으로 프로그램이 가능한 입출력 단자. Pull-up 저항은 소프트웨어로 제어할 수 있다.	-

# 6.1. 메모리 구조

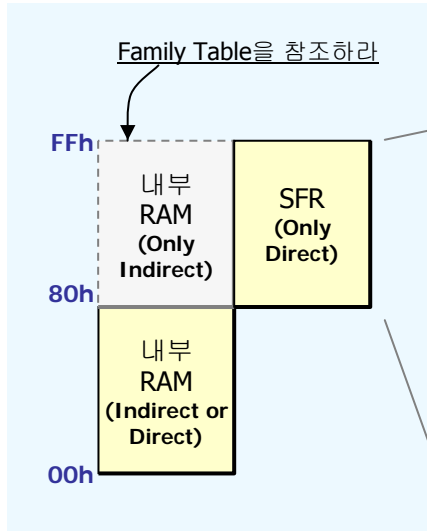


[ 프로그램 메모리 ]  
(Read Only)



[ 데이터 메모리 ]  
(Read and Write)

## 6.2. SFR (특수 기능 레지스터) Map



비트 주소지정 가능

□ : MiDAS1.0 Family에서 새로 첨가된 SFR  
□ : 미래 사용을 위해 예약됨.

F8h	EIP								FFh
F0h	B								F7h
E8h	EIE						ADCR	ADCON	EFh
E0h	ACC		ADCSEL	<sup>1)</sup> ALTSEL	POSEL	P1SEL	P2SEL	P3SEL	E7h
D8h	WDCON				PWM0CON	PWM1CON	PWMOD	PWM1D	DFh
D0h	PSW								D7h
C8h	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2			CFh
C0h					PMR	STATUS			C7h
B8h	IP	SADEN							BFh
B0h	P3							IPH	B7h
A8h	IE	SADDR							AFh
A0h	P2					P4	P4SEL		A7h
98h	SCON	SBUF							9Fh
90h	P1	EXIF							97h
88h	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON		8Fh
80h	P0	SP	DPL	DPH				PCON	87h

<sup>1)</sup> ALTSEL :

- 단지 8/10/20pin DIP/SOP package에 가능.

- 40DIP/44PLCC/44LQFP/28DIP/28SOP에서 이 레지스터를 사용하지 말라.

## 6.2. SFR에 대한 간략한 설명

### ◆ 80C52 SFR 레지스터

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

### ◆ MiDAS1.0 Family에 새로 추가된 SFR 레지스터

레지스터	이름	리셋 값
<b>POSEL</b>	Port 0 Pull-up Control	11111111
<b>P1SEL</b>	Port 1 Pull-up Control	00000000
<b>P2SEL</b>	Port 2 Pull-up Control	00000000
<b>P3SEL</b>	Port 3 Pull-up Control	00000000
<b>P4SEL</b>	Port 4 Pull-up Control	****0000
<b>P4</b>	Port 4. Only Available for 44PLCC	****1111
<b>ALTSEL</b>	Alternative Function Selection only for 8/10/20 DIP/SOP Package Type Don't use in 40DIP/44PLCC Package Type.	**000000
<b>PWM0CON</b>	PWM 0 Control	00000000
<b>PWM1CON</b>	PWM 1 Control	00000000
<b>PWM0D</b>	PWM 0 Duty Data	00000000
<b>PWM1D</b>	PWM 1 Duty Data	00000000
<b>ADCON</b>	ADC Control & ADC Result LSB	001000*0
<b>ADCR</b>	ADC Result Value Including MSB bit	00000000
<b>ADCSEL</b>	ADC Channel Selection and ADC Input Clock Divide Control	000*0000
<b>WDCON</b>	Power Flag and Watchdog Timer Control	*1010000
<b>CKCON</b>	Watchdog Timer and 4-cycle Switching Control	00000***
<b>PMR</b>	Power Management Control	*****0**
<b>EXIF</b>	Added External Interrupt and LVD Control	00001**1
<b>EIP</b>	Extended Interrupt Priority	*0000000
<b>EIE</b>	Extended Interrupt Enable	*0000000
<b>STATUS</b>	Crystal Status	***0****

주의 : 비트 \*를 건드리지 말라. 이 비트들이 수정되면 오작동을 일으킬 것이다.

## 6.3. 명령어 셋 요약

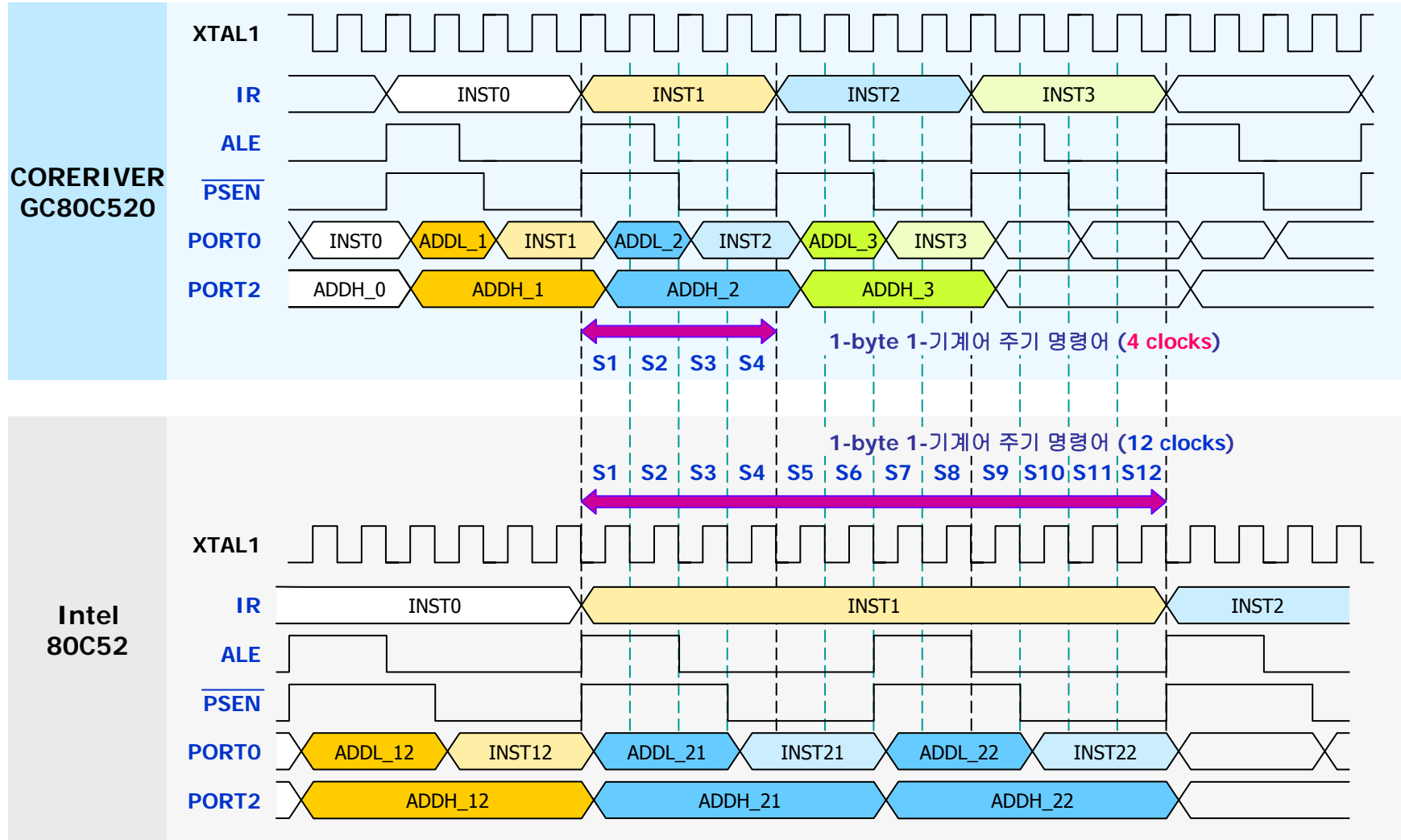
◆ 더 자세한 사항은 부록 A(명령어 셋)을 참조하라.

종류	명령어	설명
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

종류	명령어	설명
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 and clear bit	
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	

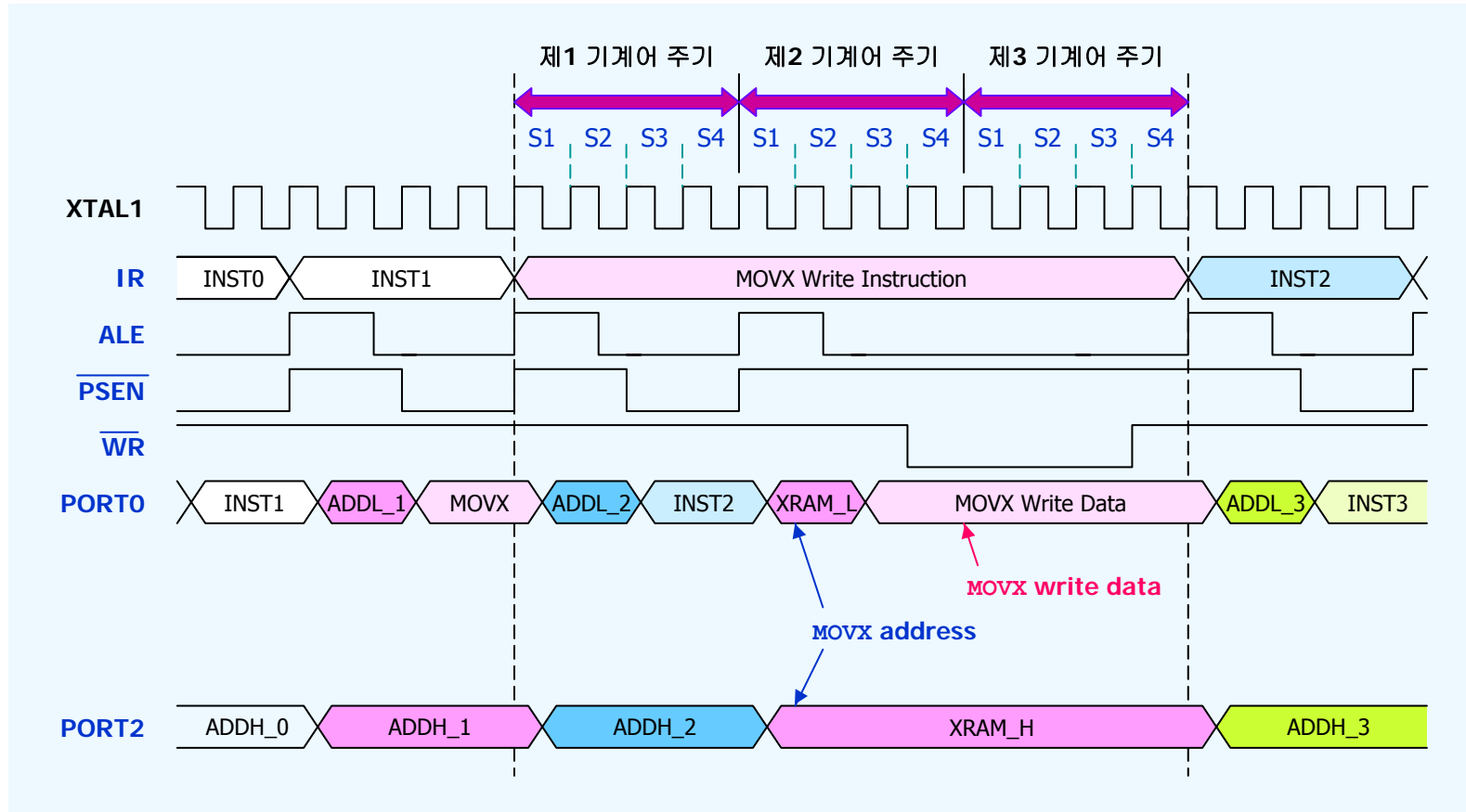
## 6.4. CPU 타이밍

- ◆ MiDAS1.0 family와 Intel 80C52의 타이밍 비교

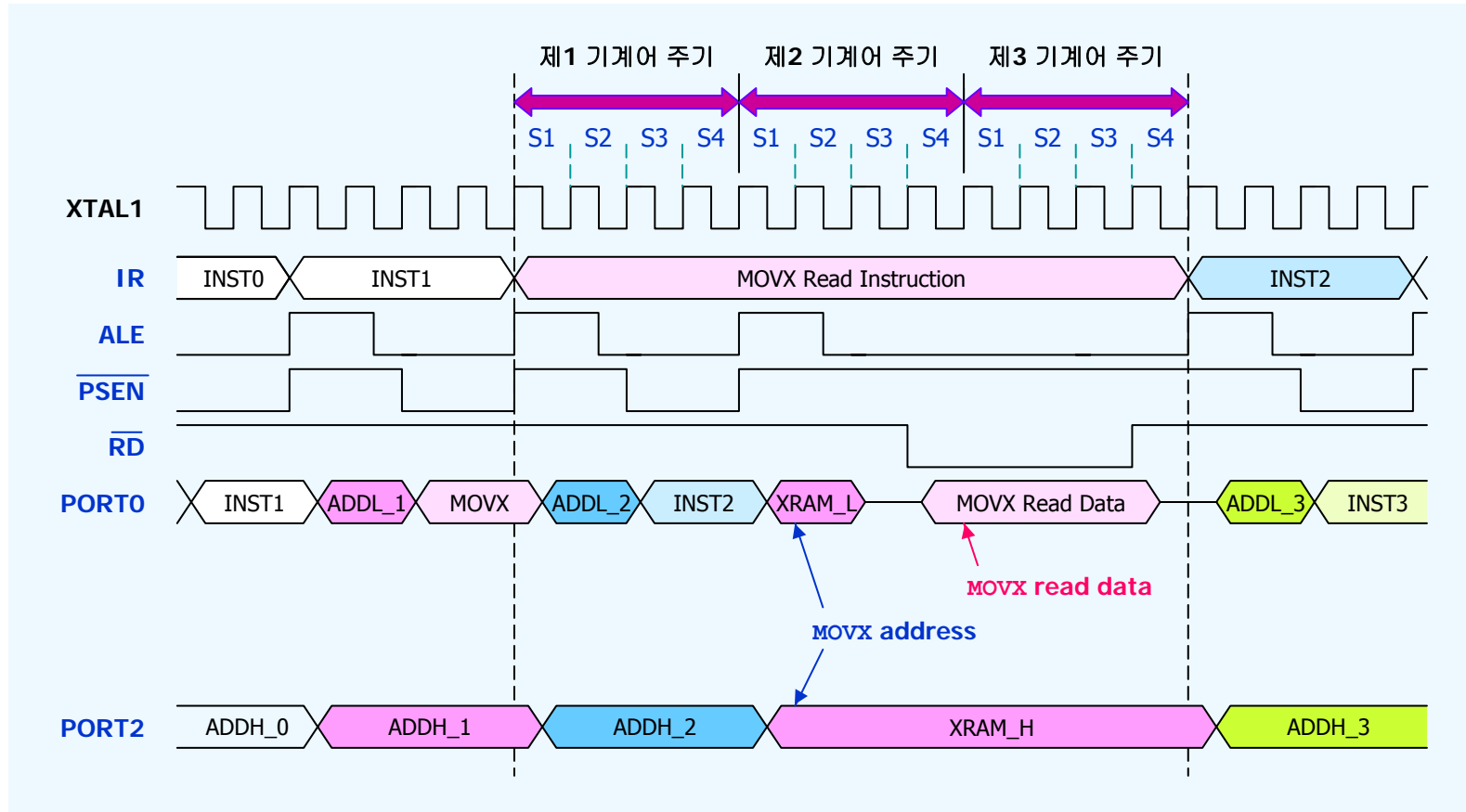




## 6.4. CPU Timing : MOVX Write Timing



## 6.4. CPU Timing : MOVX Read Timing



## 6.4. CPU 타이밍 : 수행 시간 표

- ◆ 세계에서 가장 빠른 명령어 수행 시간

명령어	GC80C520 (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 DEC DPTR	4 clocks 4 clocks	8 clocks 8 clocks	12 clocks Not exist	24 clocks Not exist
Others	Same	Same	Same	-

## 6.5. 입출력 단자 : PORT0[7:0]

- ◆ 유사 양방향 단자 (Intel 호환).
- ◆ 단자 P0.0만 아날로그 입력 핀으로 사용될 수 있다 (ADC 입력 채널 0).
- ◆ 외부에 접근할 때, P0 SFR는 자동적으로 "FFh"으로 설정된다.
- ◆ 선택적으로 Pull-up 저항 조정.
- ◆ 읽기-수정-쓰기 명령은 단자 핀이 아니라 SFR 레지스터를 읽는다.
  - ✓ ANL / ORL / XRL / JBC / CPL / INC / DEC / DJNZ / MOV PX.Y, C / CLR PX.Y / SETB PX.Y

### ◆ PORT0 설명

#### ✓ ADCSEL (E2h) : ADC 입력 선택 레지스터

ADIV2	ADIV1	ADIV0	-	ADC3	ADC2	ADC1	ADC0
R/W(0)	R/W(0)	R/W(0)		R/W(0)	R/W(0)	R/W(0)	R/W(0)

- ADC0 : '1'이면 P0.0에서 ADC0 입력 선택.  
'0'이면 P0.0에서 디지털 입력 선택
- ADIV2, ADIV1, ADIV0 : ACD 입력 클럭 분주비 결정.

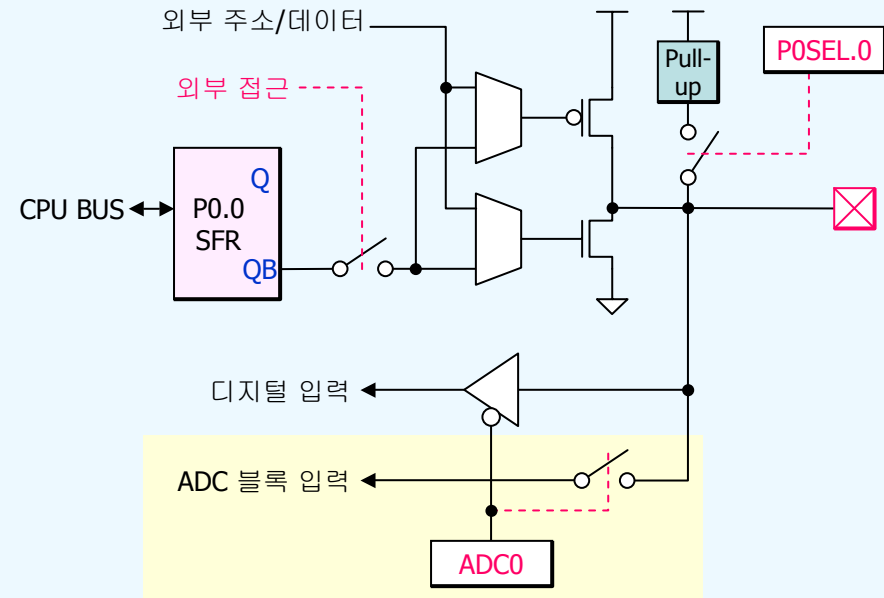
#### ✓ POSEL (E4h) : P0 Pull-up 제어 레지스터

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

- POSEL[7:0] : Pull-up 저항 ON/Off 선택  
0 = Pull-up 저항 ON / 1 = Pull-up 저항 Off  
리셋 후에 PORT0 Pull-up 저항은 OFF된다.

#### ✓ PO (80h) : PORT0 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)



## 6.5. 입출력 단자: PORT1[7:0]

- ◆ 유사 양방향 단자 (Intel 8052 호환).
- ◆ P1.1 / P1.2 / P1.3는 아날로그 입력 핀으로 사용될 수 있다(ADC 입력 채널 1/2/3).
- ◆ 선택적 Pull-up 저항 조정.
- ◆ SFR 비트가 "1"이면 다른 선택 기능으로 사용된다.
  - ✓ P1.0 = T2, PWM1 / P1.1 = T2EX, ADC1 / P1.2 = ADC2 / P1.3 = ADC3 / P1.4 = INT2 / P1.5 =  $\overline{\text{INT3}}$  / P1.6 = INT4 / P1.7 =  $\overline{\text{INT5}}$
- ◆ 읽기-수정-쓰기 명령어는 단자 핀이 아니라 SFR 레지스터를 읽는다.
  - ✓ ANL / ORL / XRL / JBC / CPL / INC / DEC / DJNZ / MOV PX.Y, C / CLR PX.Y / SETB PX.Y
- ◆ PORT1 설명

- ✓ ADCSEL (E2h) : ADC 입력 선택 레지스터

ADIV2	ADIV1	ADIV0	-	ADC3	ADC2	ADC1	ADC0
R/W(0)	R/W(0)	R/W(0)		R/W(0)	R/W(0)	R/W(0)	R/W(0)

- ADC1 : 1 = P1.1에서 ADC1 입력 선택, 0 = 디지털 입력 선택.
- ADC2 : 1 = P1.2에서 ADC2 입력 선택, 0 = 디지털 입력 선택.
- ADC3 : 1 = P1.3에서 ADC3 입력 선택, 0 = 디지털 입력 선택.

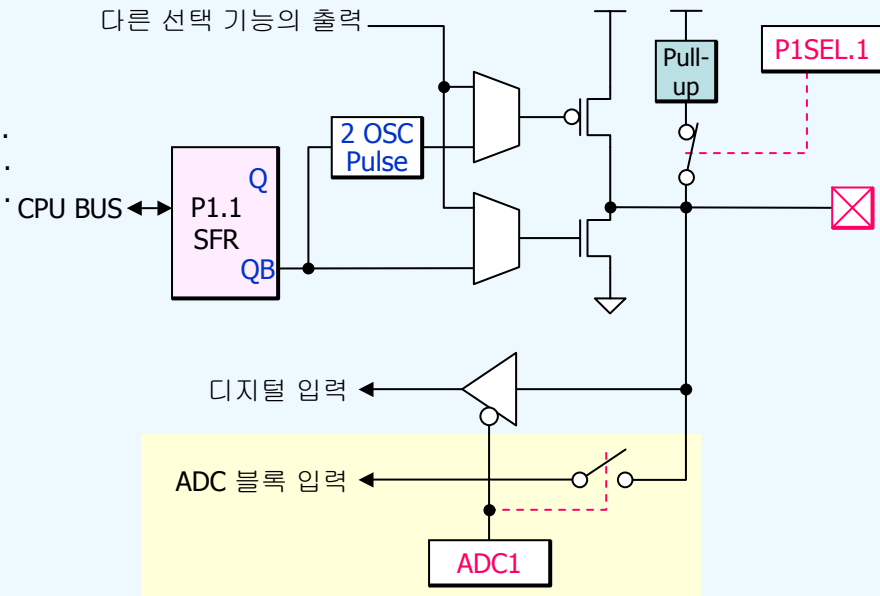
- ✓ P1SEL (E5h) : P1 Pull-up 제어 레지스터

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

- P1SEL[7:0] : Pull-up 저항 ON/Off 선택  
0 = Pull-up 저항 ON / 1 = Pull-up 저항 Off  
리셋 후에 PORT1 Pull-up 저항 ON.

- ✓ P1 (90h) : PORT1 레지스터

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. 입출력 단자 : PORT2[7:0]

- ◆ 유사 양방향 단자 (Intel 8052 호환).
- ◆ 선택적 Pull-up 저항 제어.
- ◆ 읽기-수정-쓰기 명령어는 단자 핀이 아니라 SFR 레지스터를 읽는다.
  - ✓ ANL / ORL / XRL / JBC / CPL / INC / DEC / DJNZ / MOV PX.Y, C / CLR PX.Y / SETB PX.Y

### ◆ PORT2 설명

- ✓ P2SEL (E6h) : P2 Pull-up 저항 제어 레지스터

P2SEL.7	P2SEL.6	P2SEL.5	P2SEL.4	P2SEL.3	P2SEL.2	P2SEL.1	P2SEL.0
---------	---------	---------	---------	---------	---------	---------	---------

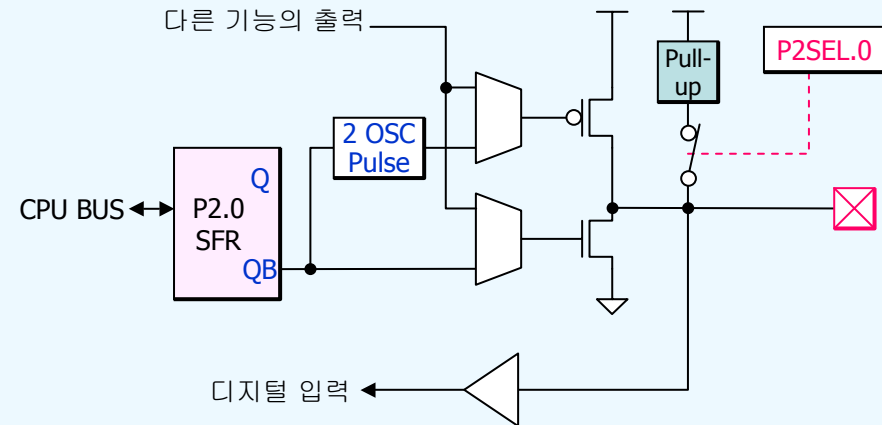
R/W(0) R/W(0) R/W(0) R/W(0) R/W(0) R/W(0) R/W(0) R/W(0)

- P2SEL[7:0] : Pull-up 저항 ON/Off 선택  
0 = Pull-up 저항 ON / 1 = Pull-up 저항 Off  
리셋 후에 PORT2 Pull-up 저항 ON.

- ✓ P2 (A0h) : PORT2 레지스터

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. 입출력 단자 : PORT3[7:0]

- ◆ 유사 양방향 단자 (Intel 8052 호환).
- ◆ 선택적으로 Pull-up 저항 사용여부 결정.
- ◆ SFR 비트가 "1"이면 다른 기능으로 사용된다.
  - ✓ P3.0 = RXD / P3.1 = TXD / P3.2 =  $\overline{\text{INT0}}$  / P3.3 =  $\overline{\text{INT1}}$  / P3.4 = T0, PWM0 / P3.5 = T1 / P3.6 =  $\overline{\text{WR}}$  / P3.7 =  $\overline{\text{RD}}$
- ◆ 읽기-수정-쓰기 명령어는 단자 핀이 아니라 SFR 레지스터를 읽는다.
  - ✓ ANL / ORL / XRL / JBC / CPL / INC / DEC / DJNZ / MOV PX.Y, C / CLR PX.Y / SETB PX.Y
- ◆ PORT3 설명

- ✓ P3SEL (E7h) : P3 Pull-up 제어 레지스터

P3SEL.7	P3SEL.6	P3SEL.5	P3SEL.4	P3SEL.3	P3SEL.2	P3SEL.1	P3SEL.0
---------	---------	---------	---------	---------	---------	---------	---------

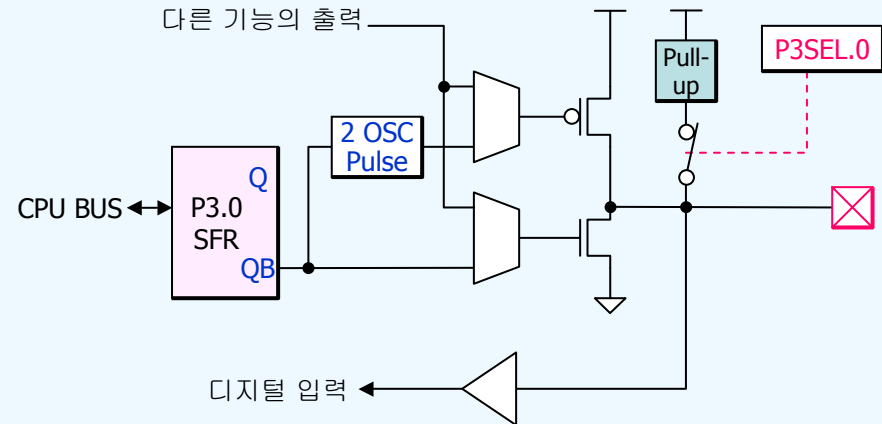
R/W(0) R/W(0) R/W(0) R/W(0) R/W(0) R/W(0) R/W(0) R/W(0)

- P3SEL[7:0] : Pull-up 저항 ON/Off 결정  
0 = Pull-up 저항 ON / 1 = Pull-up 저항 Off  
리셋 후에 PORT3 Pull-up 저항 ON.

- ✓ P3 (B0h) : PORT3 레지스터

P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
------	------	------	------	------	------	------	------

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



## 6.5. 입출력 단자 : PORT4[3:0]

- ◆ 유사 양방향 단자.
- ◆ 선택적으로 Pull-up 저항 사용여부 결정.
- ◆ 오직 44-pin PLCC package의 경우에 사용 가능.
- ◆ 읽기-수정-쓰기 명령어는 단자 핀이 아니라 SFR 레지스터를 읽는다.
  - ✓ ANL / ORL / XRL / JBC / CPL / INC / DEC / DJNZ

### ◆ PORT4 설명

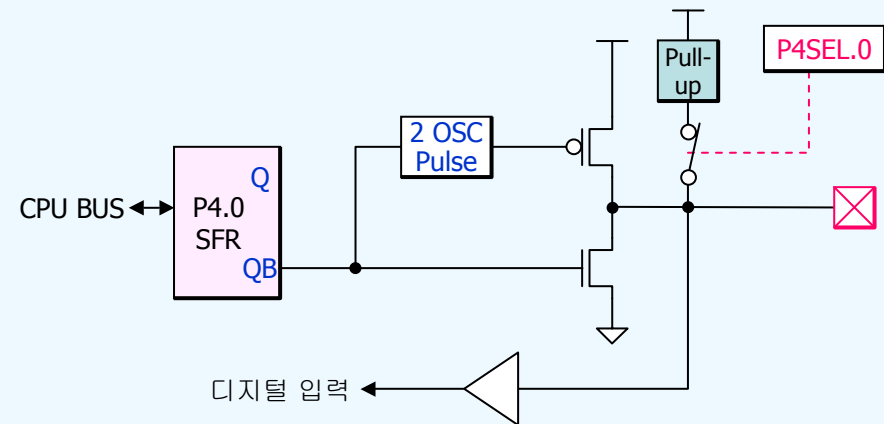
- ✓ P4SEL (A6h) : P4 Pull-up 제어 레지스터

-	-	-	-	P4SEL.3	P4SEL.2	P4SEL.1	P4SEL.0
				R/W(0)	R/W(0)	R/W(0)	R/W(0)

- P4SEL[3:0] : Pull-up 저항 ON/Off 결정  
0 = Pull-up 저항 ON / 1 = Pull-up 저항 Off  
리셋 후에 PORT4 Pull-up 저항 ON.

- ✓ P4 (A5h) : PORT4 레지스터

-	-	-	-	P4.3	P4.2	P4.1	P4.0
				R/W(1)	R/W(1)	R/W(1)	R/W(1)



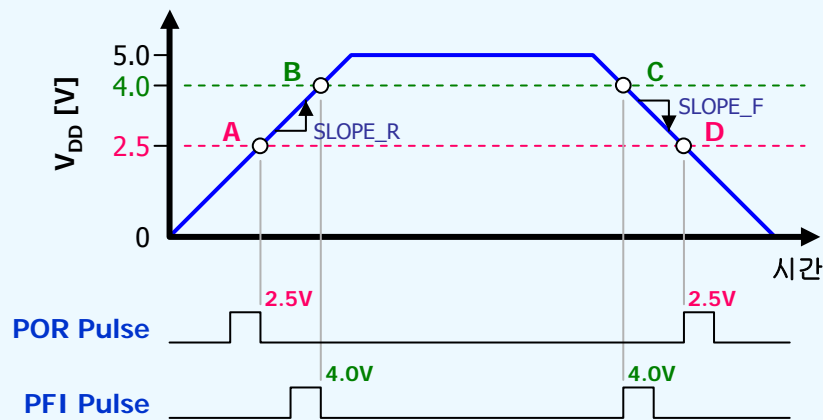


## 6.6. LVD (저전압 검출기)

- ◆ 칩 내부 파워 온 리셋 : 2.5V
- ◆ 칩 내부 파워 오류 리셋 : 2.5V
- ◆ Optional 파워 오류 인터럽트 : 4.0V
- ◆ 표시 변화

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

- POF is a mirror of POR.



- ✓ **EXIF (91h)** : 외부 인터럽트 플래그 레지스터

IE5	IE4	IE3	IE2	XT	-	-	BGS
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R(1)			R/W(1)

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

- ✓ **PCON (87h)** : 파워 제어 레지스터

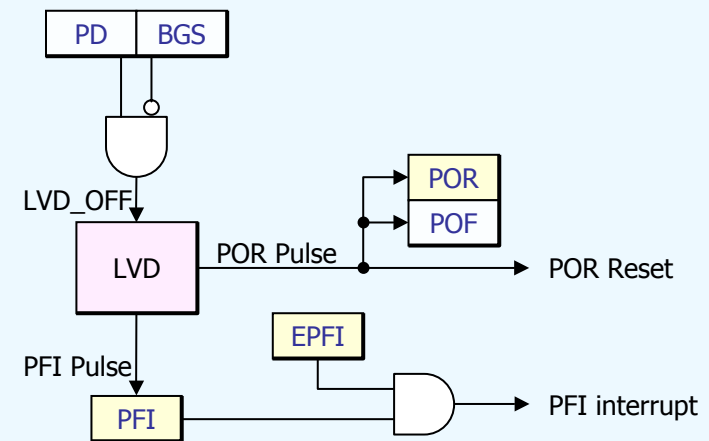
SMOD1	SMOD0	-	POF	GF1	GF0	PD	IDL
R/W(0)	R/W(0)		R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- POF : 파워 오프 표시
- PD : 파워 다운 비트

- ✓ **WDCON (D8h)** : Watchdog & 파워 상태 레지스터

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

- POR : 파워 온 리셋 표시
- EPFI : 파워 오류 인터럽트 허용
- PFI : 파워 오류 인터럽트 표시 (항상 1 @ 3V 동작)



## 6.7. WDT (Watchdog Timer)

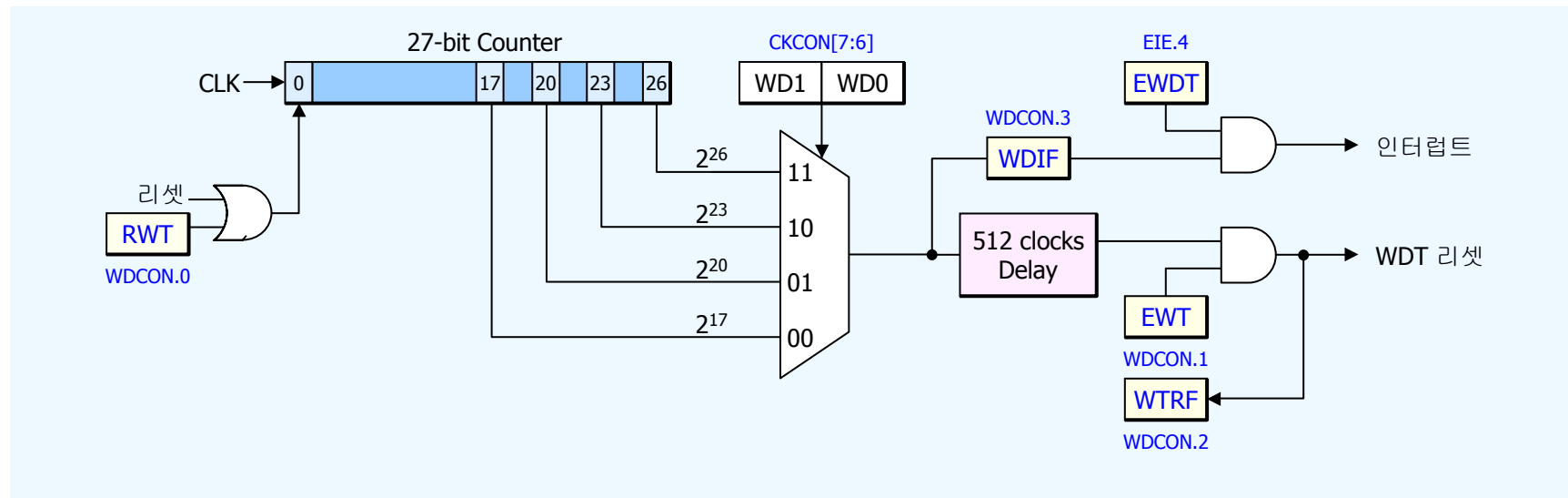
- ◆ 외부 잡음 등의 원인으로 인한 프로그램 오동작을 검출한다.
- ◆ WDT 인터럽트를 사용하여 동작을 정상적인 상태로 되돌린다.
- ◆ Watchdog 타임 아웃 값
  - ✓ 기본설정 :  $WD[1:0] = [0,0]$

WD1	WD0	인터럽트 타임 아웃 (@25MHz)		리셋 타임 아웃 (@25MHz)	
0	0	$2^{17}$ clocks	5.24 ms	$2^{17} + 512$ clocks	5.26 ms
0	1	$2^{20}$ clocks	41.94 ms	$2^{20} + 512$ clocks	41.96 ms
1	0	$2^{23}$ clocks	335.54 ms	$2^{23} + 512$ clocks	335.56 ms
1	1	$2^{26}$ clocks	2,684.35 ms	$2^{26} + 512$ clocks	2,684.38 ms

✓ **WDCON (D8h) : Watchdog & 파워 상태 레지스터**

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

- POR : 파워 온 리셋
- EPFI : 파워 오류 인터럽트 허용 (1: 허용, 0: 금지)
- PFI : 파워 오류 인터럽트 표시
- WDIF : Watchdog 타이머 인터럽트 표시
- WTRF : Watchdog 타이머 리셋 표시. 단지 S/W로 지워진다.
- EWT : Watchdog 타이머 리셋 허용(1: 허용, 0: 금지)
- RWT : Watchdog 타이머를 재출발시킴.



## 6.8. 타이머/카운터 : 타이머 0/1

- ◆ Intel 80C52 타이머/카운터와 기능이 호환
- ◆ 소프트웨어로 time base를 선택: 4 clocks 또는 12 clocks

모드 타이머	모드 0 (M1,M0=00)	모드 1 (M1,M0=01)	모드 2 (M1,M0=10)	모드 3 (M1,M0=11)
타이머0	13-비트 T/C	16-비트 T/C	8-비트 T/C 자동 재입력 (TL0 ← TH0)	8-비트 T/C (TL0) → 타이머0 인터럽트 8-비트 T/C (TH0) → 타이머1 인터럽트
타이머1	13-비트 T/C	16-비트 T/C	8-비트 T/C 자동 재입력 (TL1 ← TH1)	정지

- ✓ **TMOD (89h) : 타이머/카운터 0/1 모드 제어 레지스터**

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)

- GATE[7] : 타이머 1 제어 비트, '1'이면 INT1 핀으로 외부에서 제어  
INT1='0'이면 카운트 정지, '1'이면 카운트 시작
- C/T[6] : 타이머 1의 카운터/타이머 선택. 설정되면, T1 핀의 입력 사용.
- M1, M0[5:4] : 타이머 1의 모드 선택  
[0,0] : 모드 0. 13-비트 T/C.  
[0,1] : 모드 1. 16-비트 T/C.  
[1,0] : 모드 2. 8-비트 T/C, 자동 재입력  
[1,1] : 모드 3. 타이머 1 정지
- GATE[3] : 타이머 0 제어 비트, '1'이면 INTO 핀으로 외부에서 제어  
INT0='0'이면 카운트 정지, '1'이면 카운트 시작
- C/T[2] : 타이머 0의 카운터/타이머 선택. 설정되면, T0 핀 입력 사용.
- M1, M0[1:0] : 타이머 0의 모드 선택  
[0,0] : 모드 0. 13-비트 T/C.  
[0,1] : 모드 1. 16-비트 T/C.  
[1,0] : 모드 2. 8-비트 T/C, 자동 재입력  
[1,1] : 모드 3. 2개의 8-비트 T/C로 작동

- ✓ **CKCON (8Eh) : 클럭 제어 레지스터**

WD1	WD0	T2M	T1M	T0M	-	-	-
-----	-----	-----	-----	-----	---	---	---

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

- T1M : 타이머 1의 time base 선택  
T1M=1, time base가 12 클럭이 아니라 4 클럭.
- T0M : 타이머 0의 time base 선택  
T0M=1, time base가 12클럭이 아니라 4클럭.

- ✓ **TCON (88h) : 타이머/카운터 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 : 타이머 1 오버플로우 표시
- TR1 : 타이머 1 동작 제어 (1: 계수 시작, 0: 정지)
- TF0 : 타이머 0 오버플로우 표시
- TR0 : 타이머 0 동작 제어 (1: 계수 시작, 0: 정지)
- IE1 : 외부 인터럽트 1 edge 표시  
Edge에서 설정, 인터럽트 처리될 때 소거.
- IT1 : 외부 인터럽트 1의 종류 제어  
Edge 검출 (IT1=1). 레벨 검출 (IT1=0)
- IE0 : 외부 인터럽트 0 edge 표시
- IT0 : 외부 인터럽트 0의 종류 제어  
edge 검출 (IT0=1). 레벨 검출 (IT0=0)

- ✓ **TL0 (8Ah) : 타이머/카운터 0 하위 바이트 레지스터**

TL0.7	TL0.6	TL0.5	TL0.4	TL0.3	TL0.2	TL0.1	TL0.0
-------	-------	-------	-------	-------	-------	-------	-------

- ✓ **TH0 (8Ch) : 타이머/카운터 0 상위 바이트 레지스터**

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

- ✓ **TL1 (8Bh) : 타이머/카운터 1 하위 바이트 레지스터**

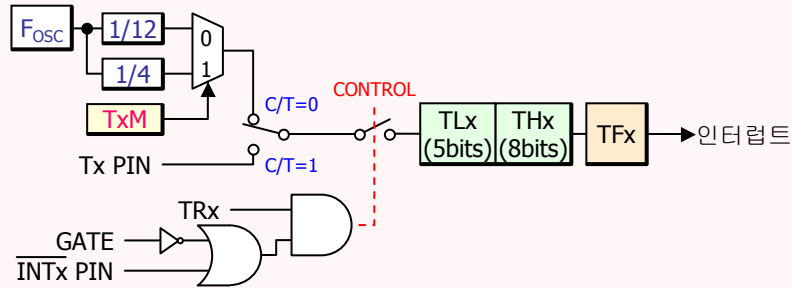
TL1.7	TL1.6	TL1.5	TL1.4	TL1.3	TL1.2	TL1.1	TL1.0
-------	-------	-------	-------	-------	-------	-------	-------

- ✓ **TH1 (8Dh) : 타이머/카운터 1 상위 바이트 레지스터**

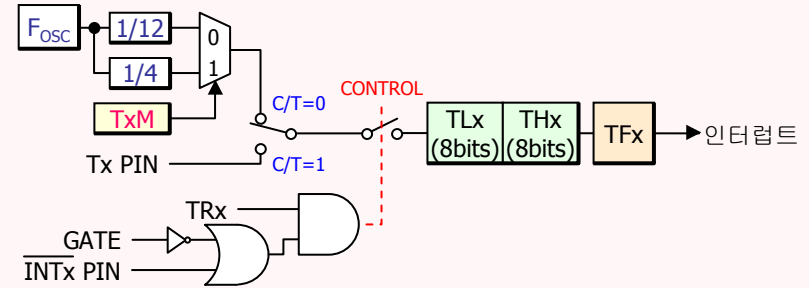
TH1.7	TH1.6	TH1.5	TH1.4	TH1.3	TH1.2	TH1.1	TH1.0
-------	-------	-------	-------	-------	-------	-------	-------

## 6.8. 타이머/카운터 : 타이머 0/1 모드 설명

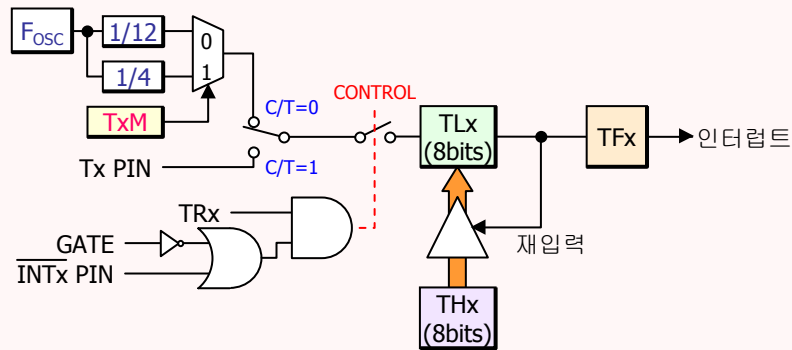
\* 기본설정 :  $F_{osc}/12$  (T0M와 T1m는 각각 0.)



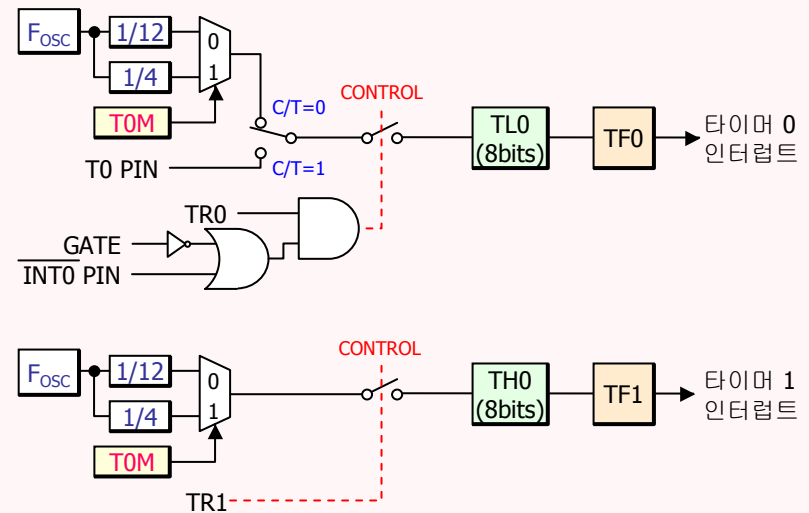
[모드 0]



[모드 1]



[모드 2]



[모드 3(타이머 0 만)]

## 6.8. 타이머/카운터 : 타이머 2

- ◆ Intel 80C52 타이머/카운터 2와 기능 호환.
- ◆ 소프트웨어로 up/down을 결정.
- ◆ 소프트웨어로 time base 선택 : 4 clocks 또는 12 clocks

1. 16-비트 자동 재입력 [RCLK+TCLK=0, CP/RL2=0, T2OE=0]	16-비트 타이머/카운터 자동 재입력 (TH2, TL2 ← RCAP2H, RCAP2L)
2. 16-비트 카운터 값 수집 [RCLK+TCLK=0, CP/RL2=1, T2OE=0]	16-비트 타이머/카운터 값 수집 (RCAP2H, RCAP2L ← TH2, TL2)
3. 보레이트 발생기 [RCLK+TCLK=1, CP/RL2=X, T2OE=X]	보레이트 발생 * 타이머 2 인터럽트 금지
4. 출력 클럭을 프로그램할 수 있음 [RCLK+TCLK=X, CP/RL2=0, T2OE=1]	P1.0으로 클럭 출력

### ✓ T2CON (C8h) : 타이머 2 제어 레지스터

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 : 타이머 2 오버플로우 표시
- EXF2 : 타이머 2 외부 edge, 소프트웨어에 의해 소거되어야 함  
EXEN2=1이고 T2EX에 하강 edge가 나타나서 재입력이나 값 수집이 일어날 때 설정됨,
- RCLK : 클럭 수신 표시, 모드 1,3에서 '1'이면 타이머 2 overflow가 수신 클럭, '0'이면 타이머 1 오버플로우가 수신 클럭.
- TCLK : 클럭 전송 표시, 송신 클럭으로 RCLK와 사용방법은 같음.
- EXEN2 : 타이머 2 외부 허용 표시. 설정되면 직렬 포트 클럭으로 사용되지 않을 때 T2EX 하강 결과로 값 수집이나 재입력이 일어나게 함. '0'이면 T2EX의 사건을 무시함.
- TR2 : 타이머 2의 시작/정지 제어
- C/T2 : 타이머나 카운터 선택. C/T2=0이면, 타이머 동작.
- CP/RL2 : 값 수집/재입력 표시.  
CP/RL2=0, 재입력. (TH2,TL2) ← (RCAP2H, RCAP2L)  
CP/RL2=1, 값 수집. (RCAP2H, RCAP2L) ← (TH2,TL2)

### ✓ CKCON (8Eh) : 클럭 제어 레지스터

WD1	WD0	T2M	T1M	T0M	-	-	-
-----	-----	-----	-----	-----	---	---	---

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

- T2M : 타이머 2 클럭 time base 선택  
T2M=1, time base가 12클럭이 아니라 4클럭이다.

### ✓ T2MOD (C9h) : 타이머 2 모드 레지스터

-	-	-	-	-	-	T2OE	DCEN
---	---	---	---	---	---	------	------

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

- T2OE : P1.0로 타이머 2 클럭 출력
- DCEN : 타이머 2 카운트 다운 허용. '1'일 때,  
T2EX 핀이 '0'이면 카운트 다운, '1'이면 카운트 업

### ✓ TL2 (CCh) : 타이머 2 하위 바이트 레지스터

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) : 타이머 2 상위 바이트 레지스터

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) : 타이머 2 수집/재입력 하위 바이트 레지스터

RCAP2L.7	RCAP2L.6	RCAP2L.5	RCAP2L.4	RCAP2L.3	RCAP2L.2	RCAP2L.1	RCAP2L.0
----------	----------	----------	----------	----------	----------	----------	----------

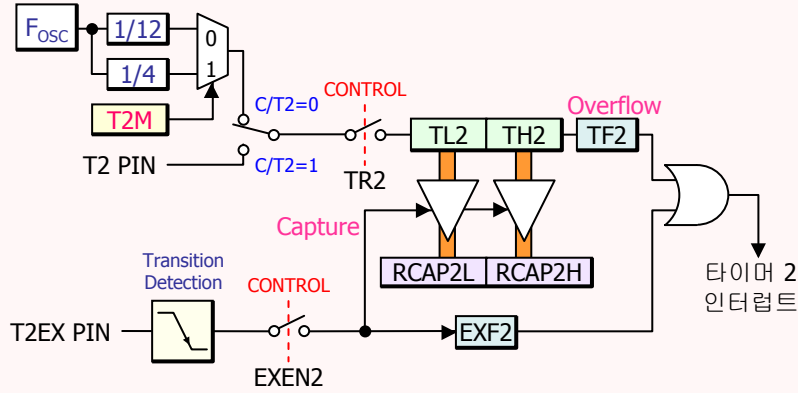
R/W(0) R/W(0) R/W(0) R/W(0) R/W(0) R/W(0) R/W(0) R/W(0)

### ✓ RCAP2H (CBh) : 타이머 2 수집/재입력 상위 바이트 레지스터

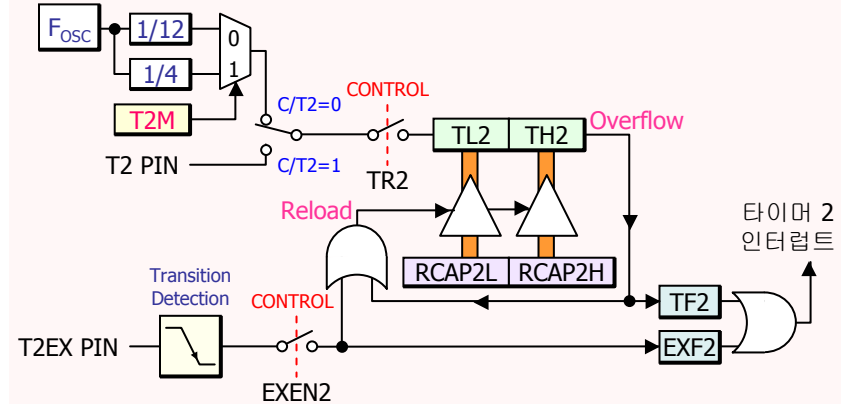
RCAP2H.7	RCAP2H.6	RCAP2H.5	RCAP2H.4	RCAP2H.3	RCAP2H.2	RCAP2H.1	RCAP2H.0
----------	----------	----------	----------	----------	----------	----------	----------

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

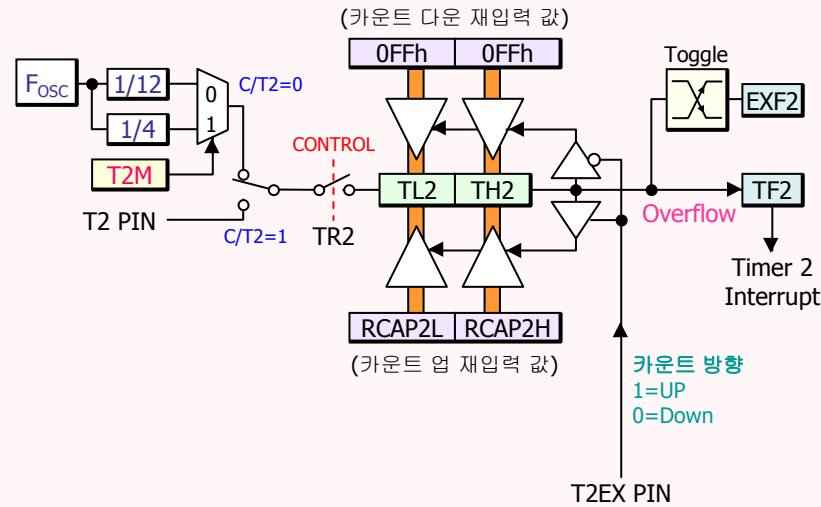
## 6.8. 타이머/카운터 : 타이머 2 모드 설명



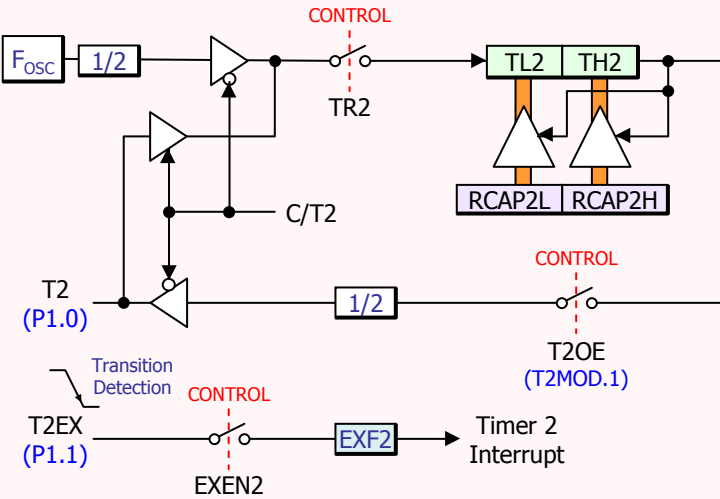
[값 수집 모드]



[자동 재입력 모드 (DCEN=0)]

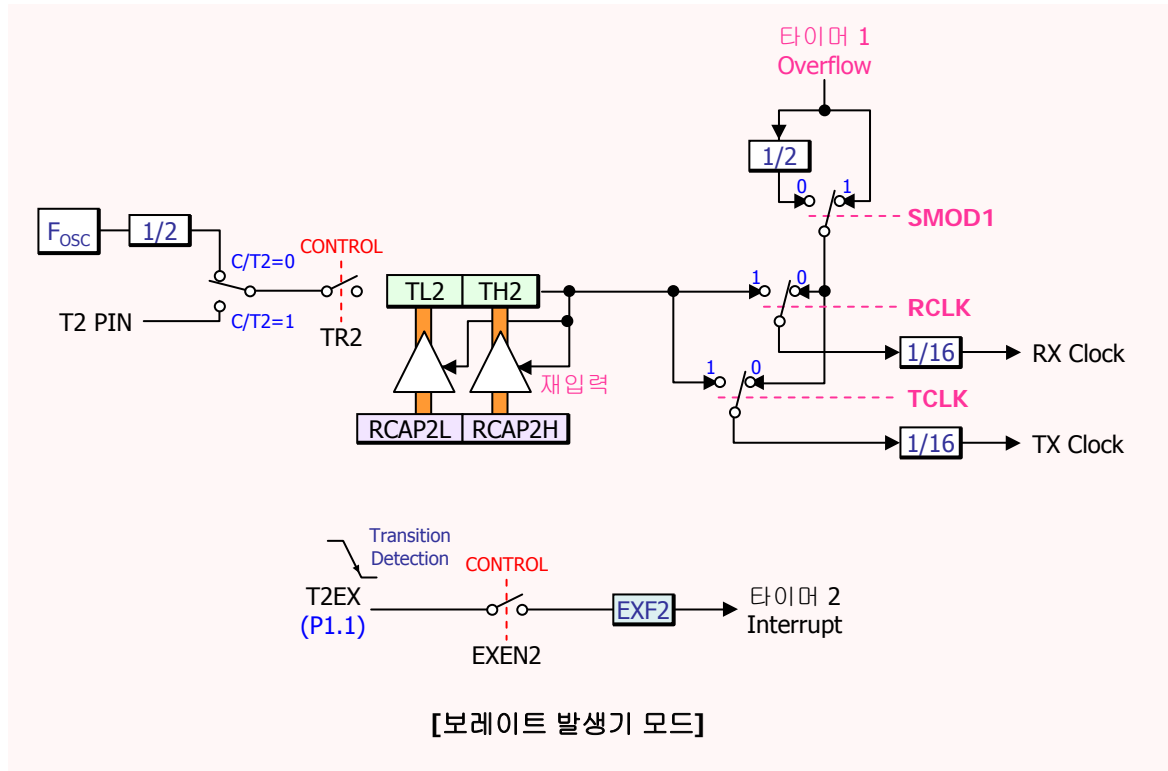


[자동 재입력 모드 (DCEN=1)]



[클럭 출력 모드]

## 6.8. 타이머/카운터 : 타이머 2 모드 설명



## 6.9. UART

- ◆ Intel 80C52 UART와 function-level 호환.
- ◆ 자동 주소 인식 : 다중 프로세서 통신.

	데이터 크기		보레이트
모드 0	8 비트	8 data bits	1/4 x Oscillator Clock
모드 1	10 비트	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
모드 2	11 비트	Start bit(0) 8 data bit Programmable bit Stop bit(1)	1/32 x Oscillator Clock (SMOD1=0) 1/16 x Oscillator Clock (SMOD1=1)
모드 3	11 비트	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

- ✓ 타이머 1 오버플로우는 CKCON 레지스터와 함께 변한다.  
→ 12 clocks time base 또는 4 clocks time base.

- ✓ PCON (87h) : 전력 제어 레지스터

SMOD1	SMOD0	-	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 : 보레이트 두 배 비트 (UART 모드 1, 2, 3, 타이머 1 사용)
- SMOD0 : '1'이면 SCON.7 접근이 FE 비트로 '0'이면 SM0 비트.

- ✓ SCON (98h) : 직렬 포트 제어 레지스터

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 : 직렬 포트 모드 선택  
[0,0] : 모드 0. 8-비트 Shift Register (OSC/4)  
[0,1] : 모드 1. 8-비트 UART (Variable)  
[1,0] : 모드 2. 9-비트 UART (OSC/32 또는 OSC/16)  
[1,1] : 모드 3. 9-비트 UART (Variable)
- SM2 : 모드 2와 3에서 자동 주소 인식 허용.  
주소 수신 후에 지움.  
모드 1에서, SM2=1이면 Stop Bit의 유효성 점검.  
모드 0에서, SM2는 0이어야 한다.
- REN : 직렬 수신 Enable, 소프트웨어로 설정하고 소거함.
- TB8 : 모드 2와 3에서 전송될 9번째 데이터 비트.
- RB8 : 모드 2와 3에서 수신된 9번째 데이터 비트.  
모드 1에서, SM2=0이면 RB8는 Stop Bit와 같다.  
모드 0에서, RB8은 사용되지 않는다.
- TI : 전송 인터럽트 플래그. 소프트웨어로 지워야 함.
- RI : 수신 인터럽트 플래그. 소프트웨어로 지워야 함.

- ✓ 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.
- 읽고 쓰는 주소는 같다.



## 6.9. UART : 보레이트 예제

### ◆ 직렬 포트 모드 0

$$\text{보레이트} = \frac{\text{오실레이터 주파수}}{4}$$

### ◆ 직렬 포트 모드 2

$$\text{보레이트} = \frac{2^{\text{SMOD1}}}{32} \times \text{PCON.7} \times \text{오실레이터 주파수}$$

### ◆ 직렬 포트 모드 1, 3

#### ✓ 타이머 1 Overflow 사용

$$\text{보레이트} = \frac{2^{\text{SMOD1}}}{32} \times \text{타이머 1 overflow}$$

#### ✓ 타이머 2 Overflow 사용

$$\text{보레이트} = \frac{\text{타이머 2 overflow}}{16}$$

#### EX) 타이머 1로 보통 사용되는 보레이트 발생

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

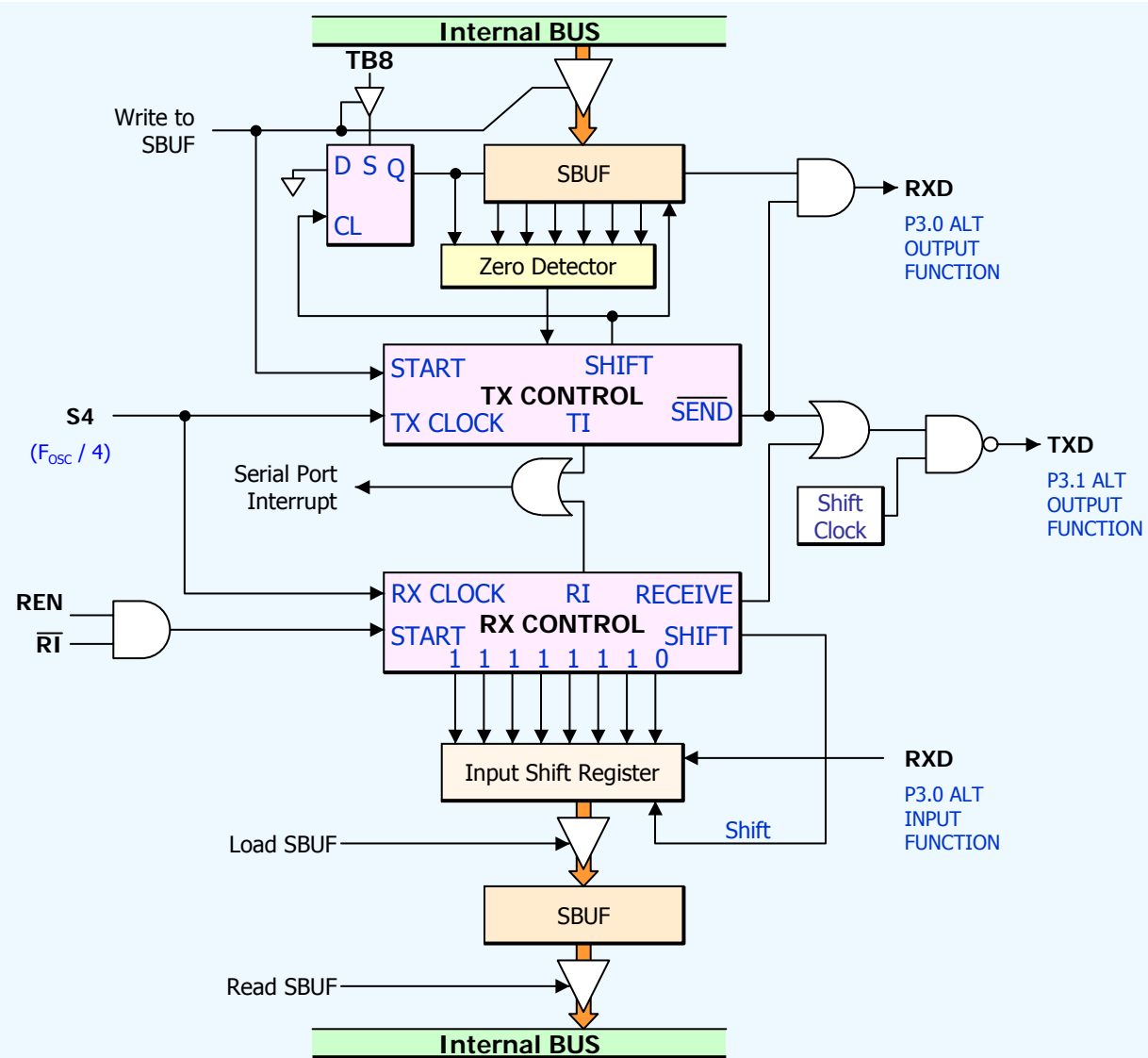
- If SMOD1(PCON.7) = 1 → 두 배 보레이트
- If T1M(CKCON.4) = 0 →  $F_{\text{OSC}} / 12$
- If T1M(CKCON.4) = 1 →  $F_{\text{OSC}} / 4$

#### EX) 타이머 2로 보통 사용되는 보레이트 발생

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

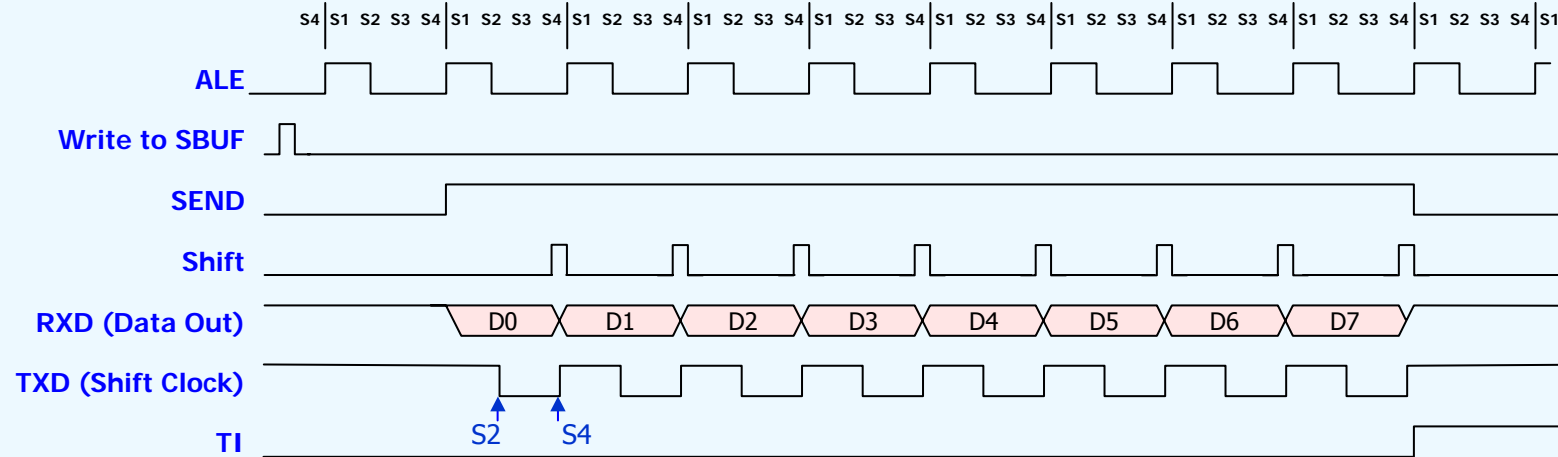
보레이트		UART 모드	F <sub>OSC</sub>	SMOD1	타이머 1		
T1M=0	T1M=1				C/T	모드	재입력 값 (TH1)
최대 : 3 MHz	최대 : 3 MHz	모드 0	12 MHz	X	X	X	X
최대 : 750 KHz	최대 : 750 KHz	모드 2	12 MHz	1	X	X	X
62.5 KHz	187.5 KHz	모드 1 & 3	12 MHz	1	0	2	FFh
19.2 KHz	57.6 KHz		11.0592 MHz	1	0	2	FDh
9.6 KHz	28.8 KHz		11.0592 MHz	0	0	2	FDh
4.8 KHz	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	FEbH

## 6.9. UART : 모드 0 기능

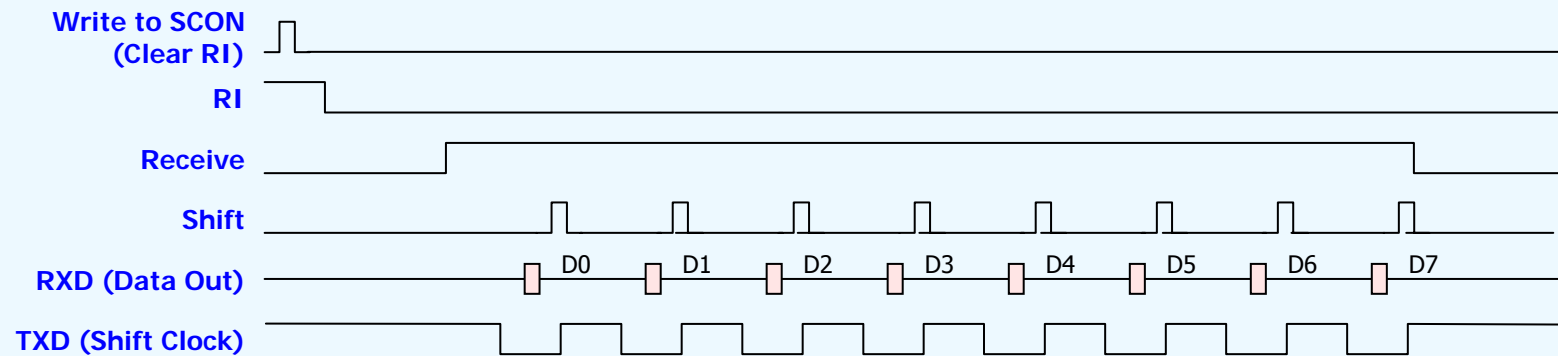


## 6.9. UART : 모드 0 타이밍

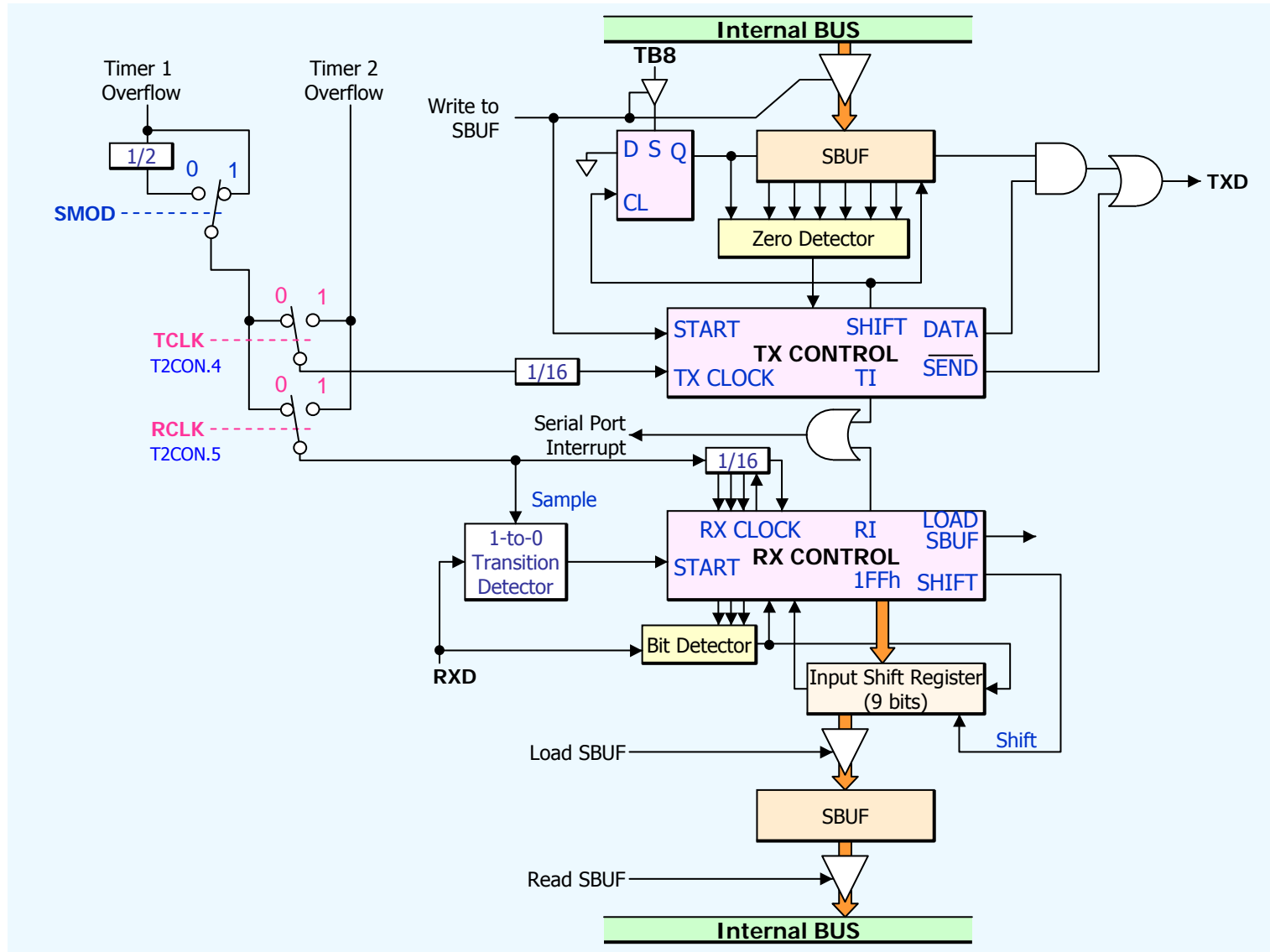
[송신]



[수신]

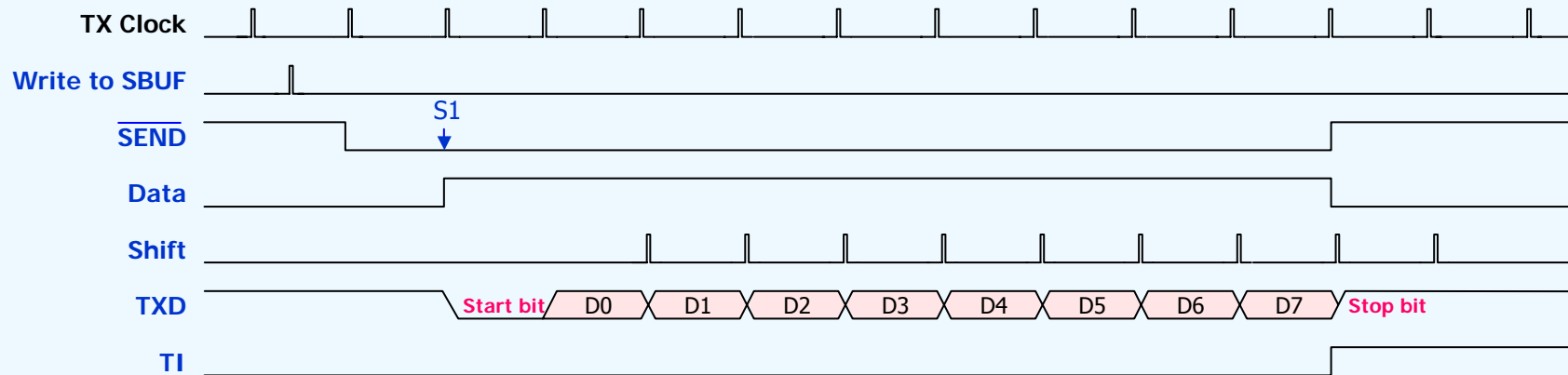


## 6.9. UART : 모드 1 기능

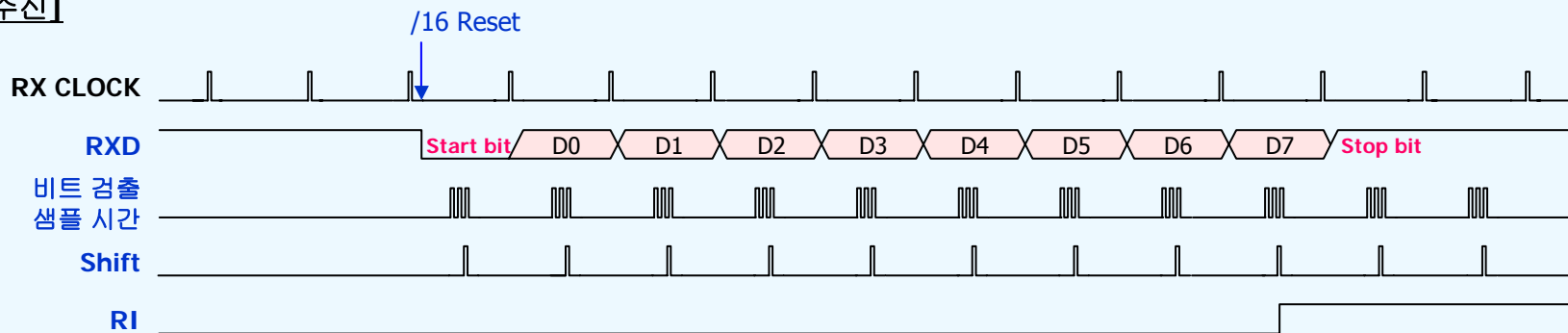


## 6.9. UART : 모드 1 타이밍

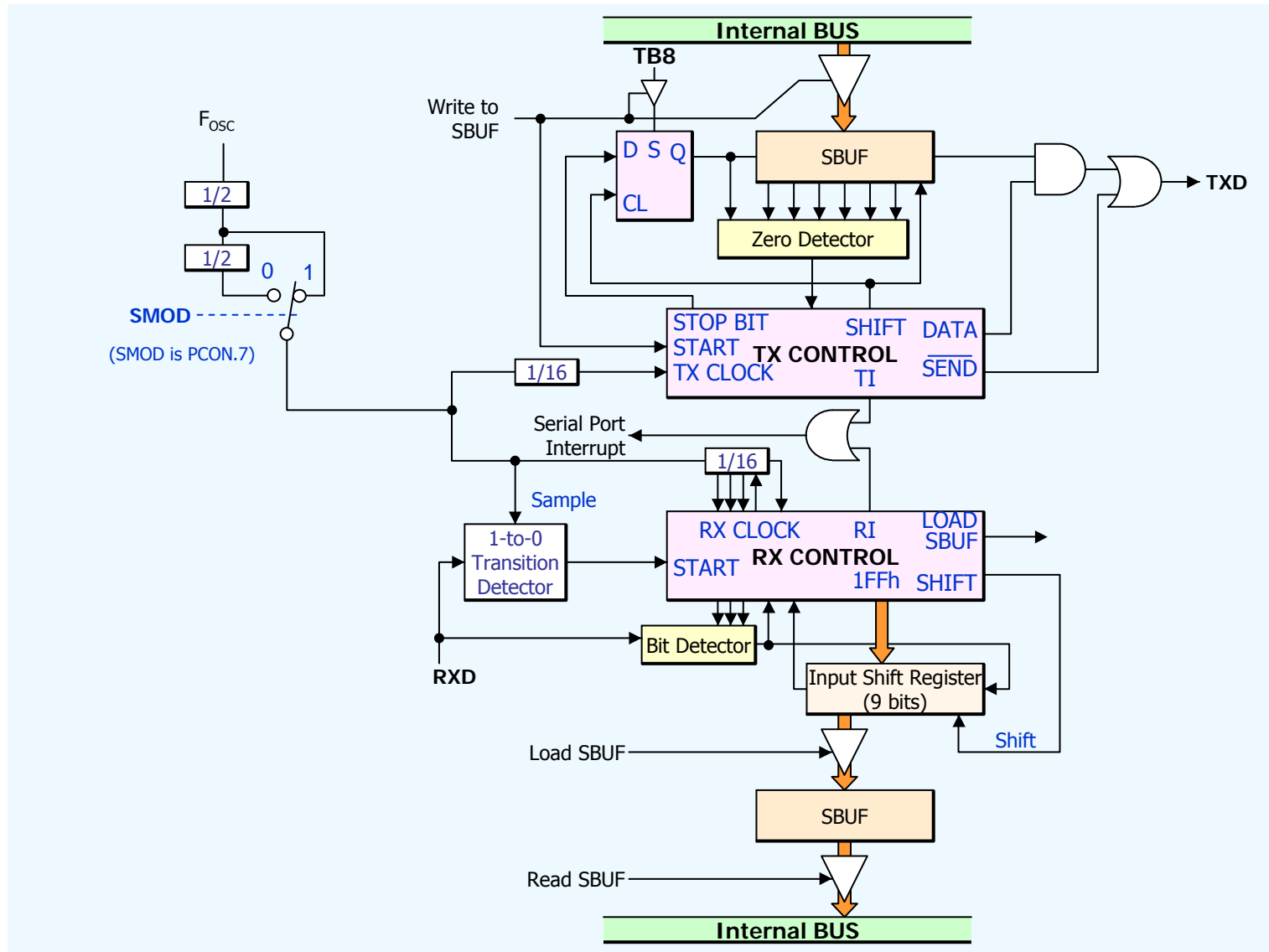
### [송신]



### [수신]

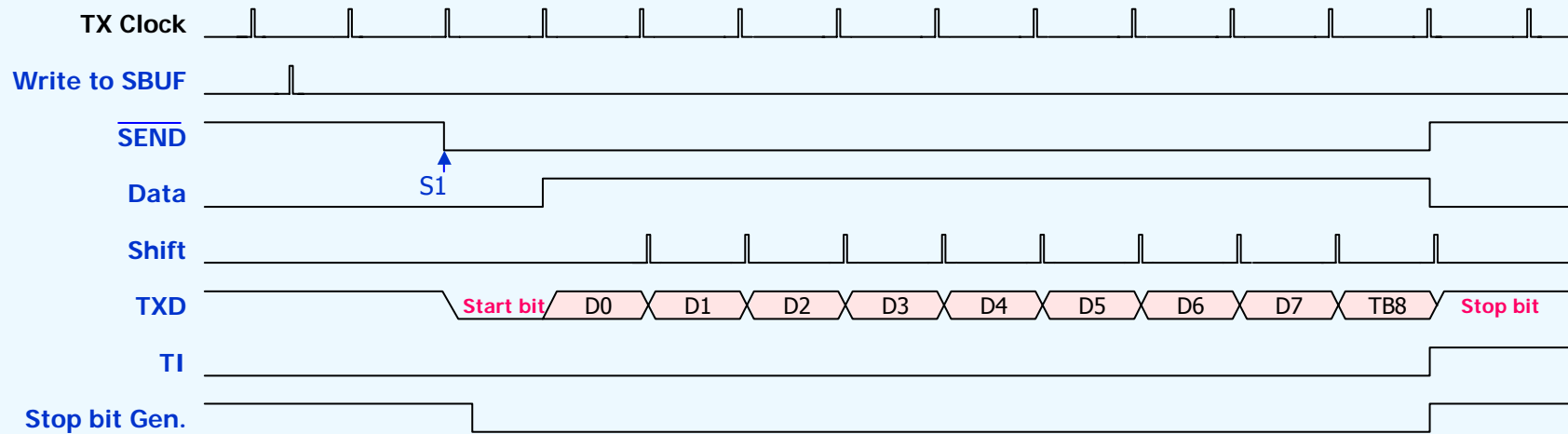


## 6.9. UART : 모드 2 기능

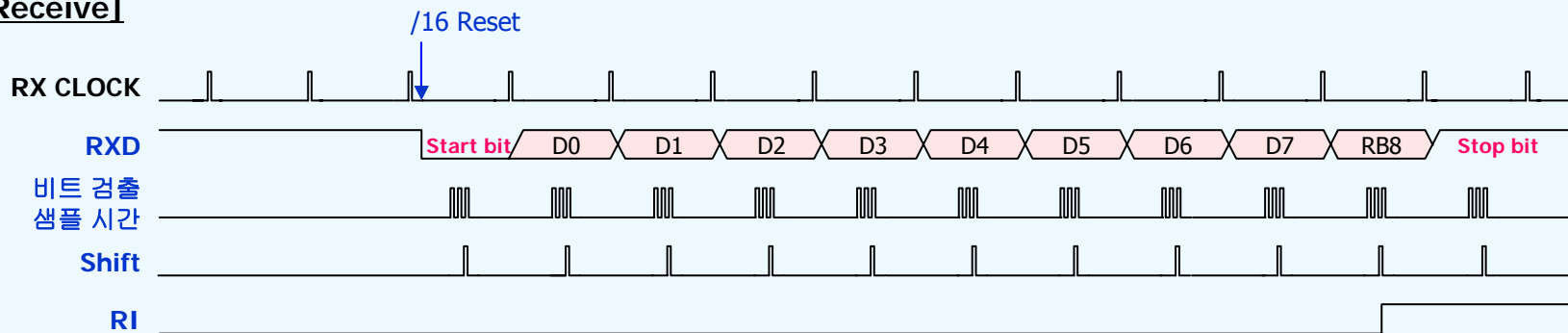


## 6.9. UART : 모드 2 타이밍

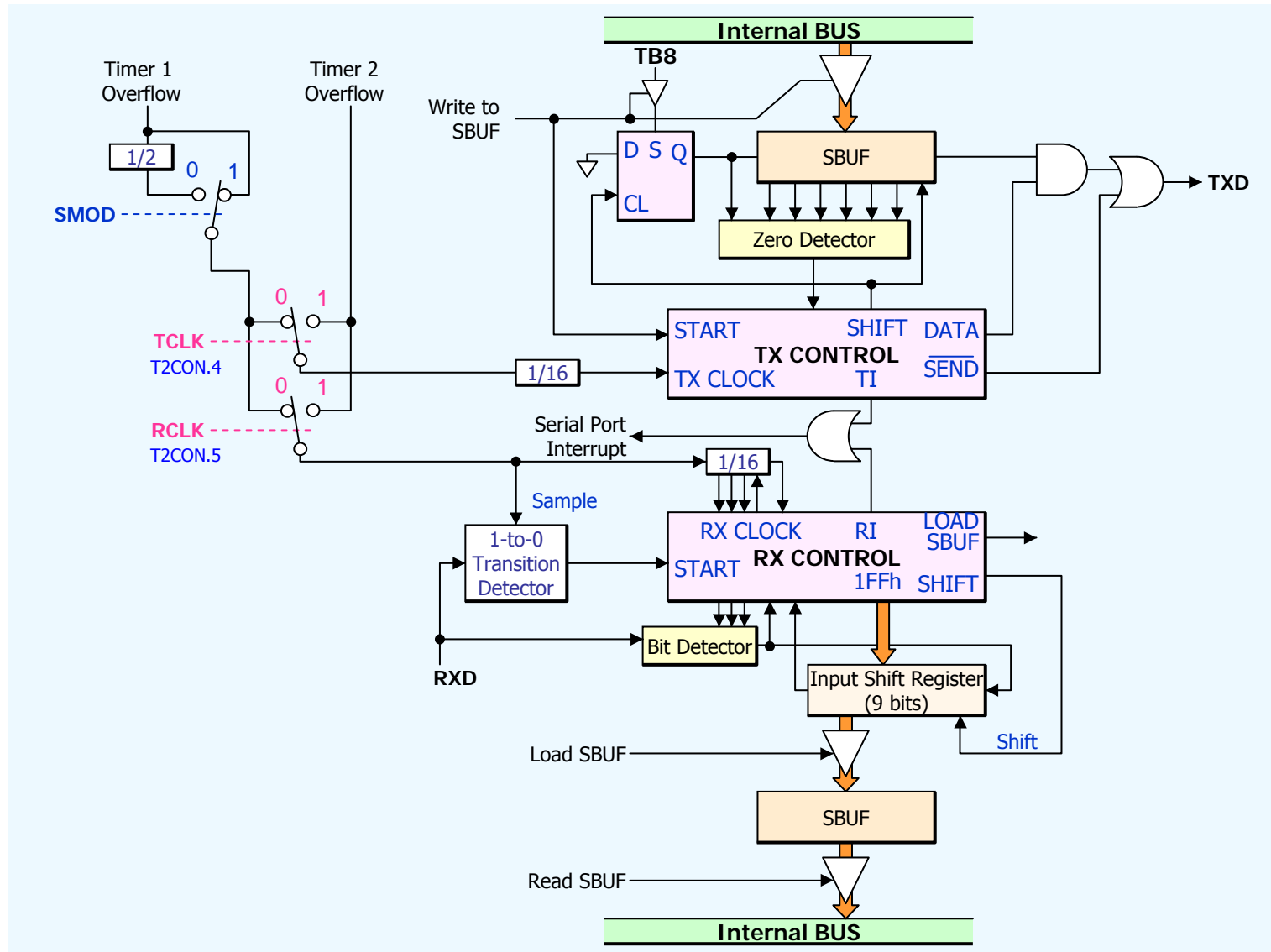
### [Transmit]



### [Receive]



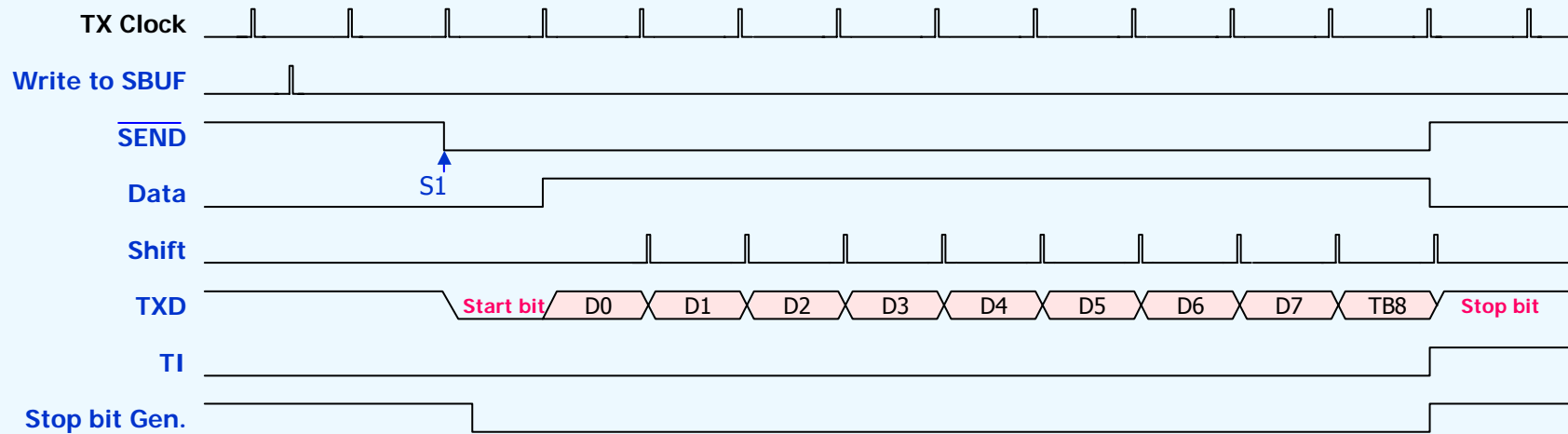
## 6.9. UART : 모드 3 기능



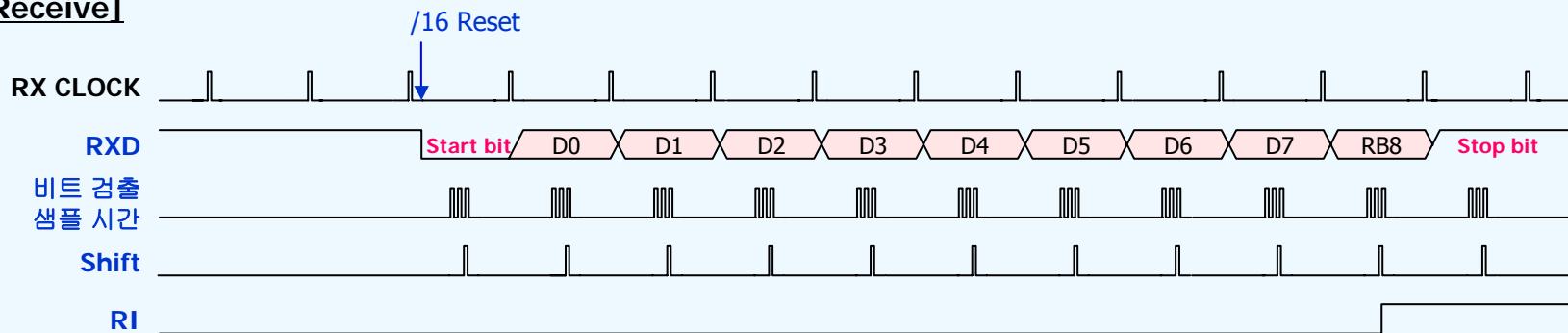


## 6.9. UART : 모드 3 타이밍

### [Transmit]

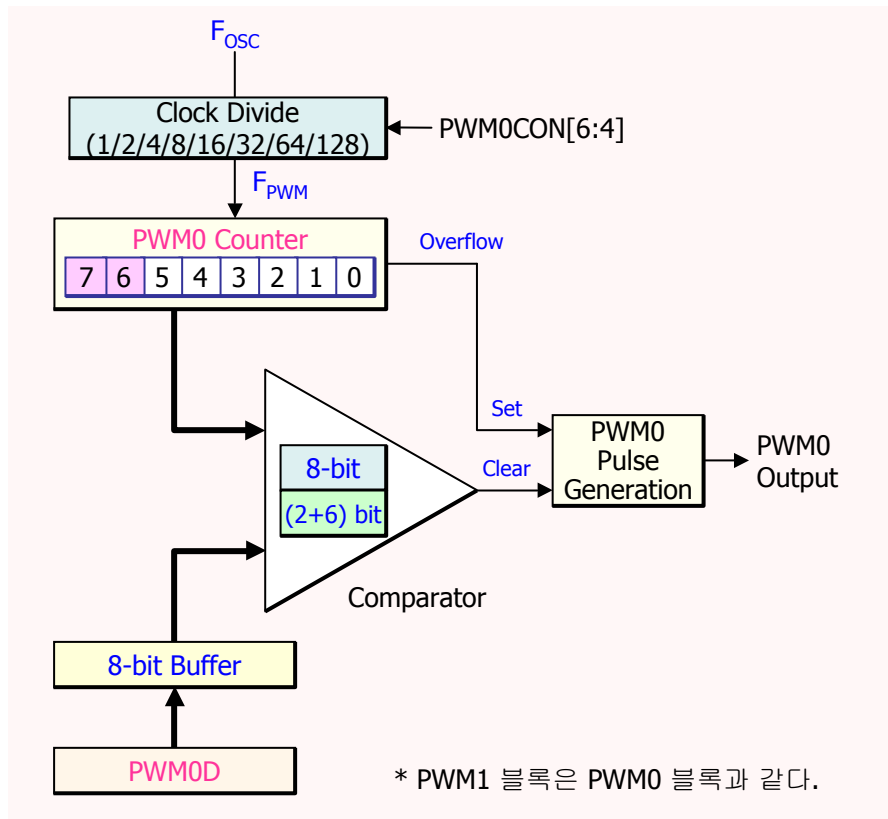


### [Receive]



## 6.10. PWM (펄스 폭 변조기)

- ◆ 지능형 2-채널 8-비트 PWM
- ◆ PWM 카운터 재입력 모드 (8-bit / 6-bit 카운터 Overflow 재입력)
- ◆ PWM 카운터는 소프트웨어에 의하여 지워질 수 있다.
- ◆ PWM는 소프트웨어에 의해 정지나 시작(재작동)될 수 있다.



모드	설명
8-bit 모드	8-bit 비교
(2+6)-bit 모드	2-bit Extension 비교 & 6-bit 비교

### ✓ PWM0CON (DCh) : PWM0 제어 레지스터

P0SEL	PS2_P0	PS1_P0	PS0_P0	MODE_P0	RL_P0	CLR_P0	RUN_P0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- P0SEL : PWM0 파형을 Port 3.4으로 출력.
- PS2\_P0, PS1\_P0, PS0\_P0 : 클럭 선 분주비 선택.  
 $F_{osc}/1, /2, /4, /8, /16, /32, /64, /128$  분주.
- MODE\_P0 : 8-bit / (2+6)-bit 카운터 모드 선택.  
MODE\_P0=0, (2+6)-bit 모드  
MODE\_P0=1, 8-bit 모드
- RL\_P0 : Duty 데이터 재입력 모드 선택.  
RL\_P0=0, 6-bit 카운터 Overflow후 재입력.  
RL\_P0=1, 8-bit 카운터 Overflow후 재입력.
- CLR\_P0 : 카운터 리셋 허용. H/W가 지움.
- RUN\_P0 : 카운터 시작.

### ✓ PWM1CON (DDh) : PWM1 제어 레지스터

P1SEL	PS2_P1	PS1_P1	PS0_P1	MODE_P1	RL_P1	CLR_P1	RUN_P1
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

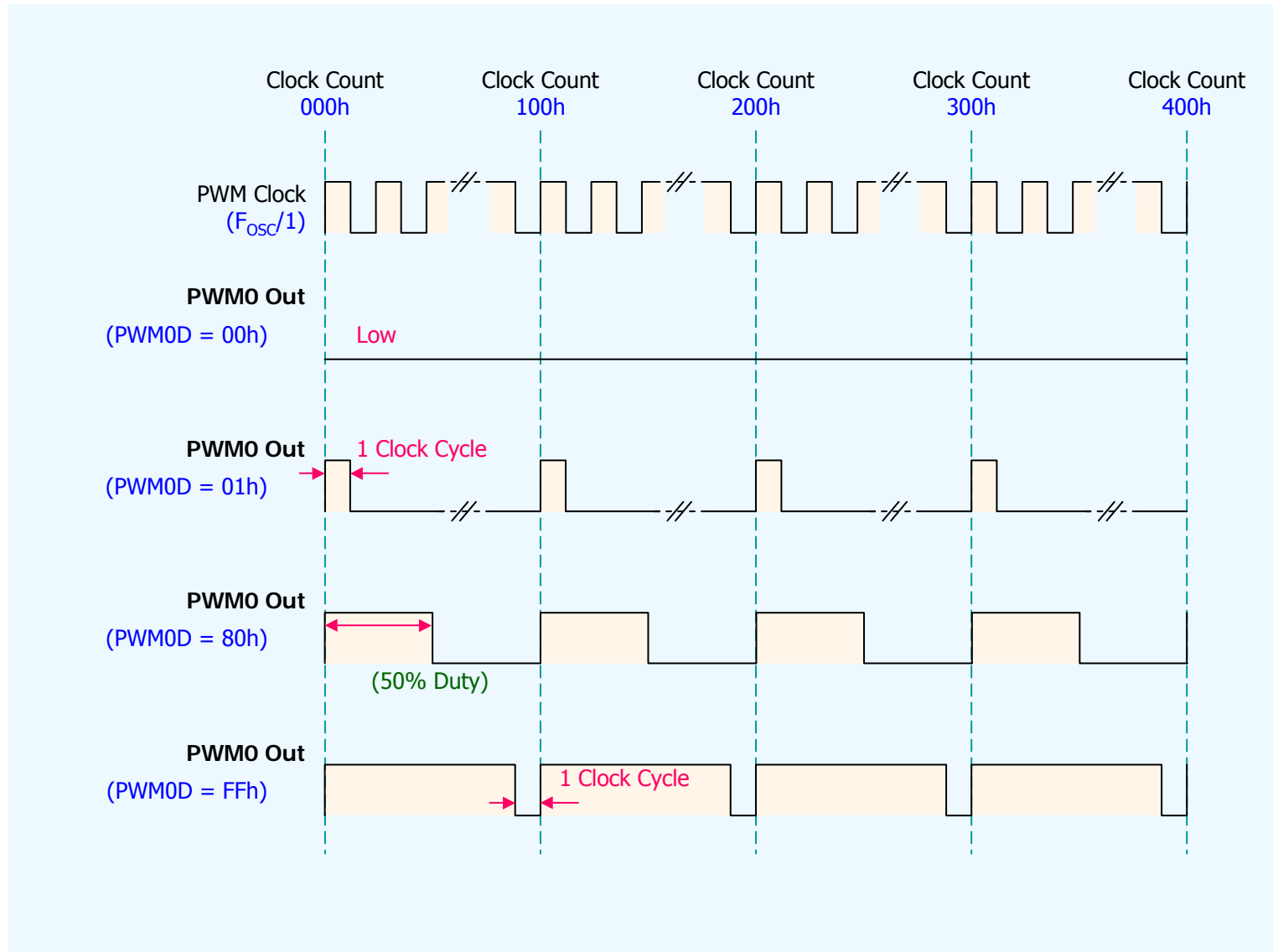
### ✓ PWM0D (DEh) : PWM0 Duty 데이터 레지스터

PWM0D.7	PWM0D.6	PWM0D.5	PWM0D.4	PWM0D.3	PWM0D.2	PWM0D.1	PWM0D.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)

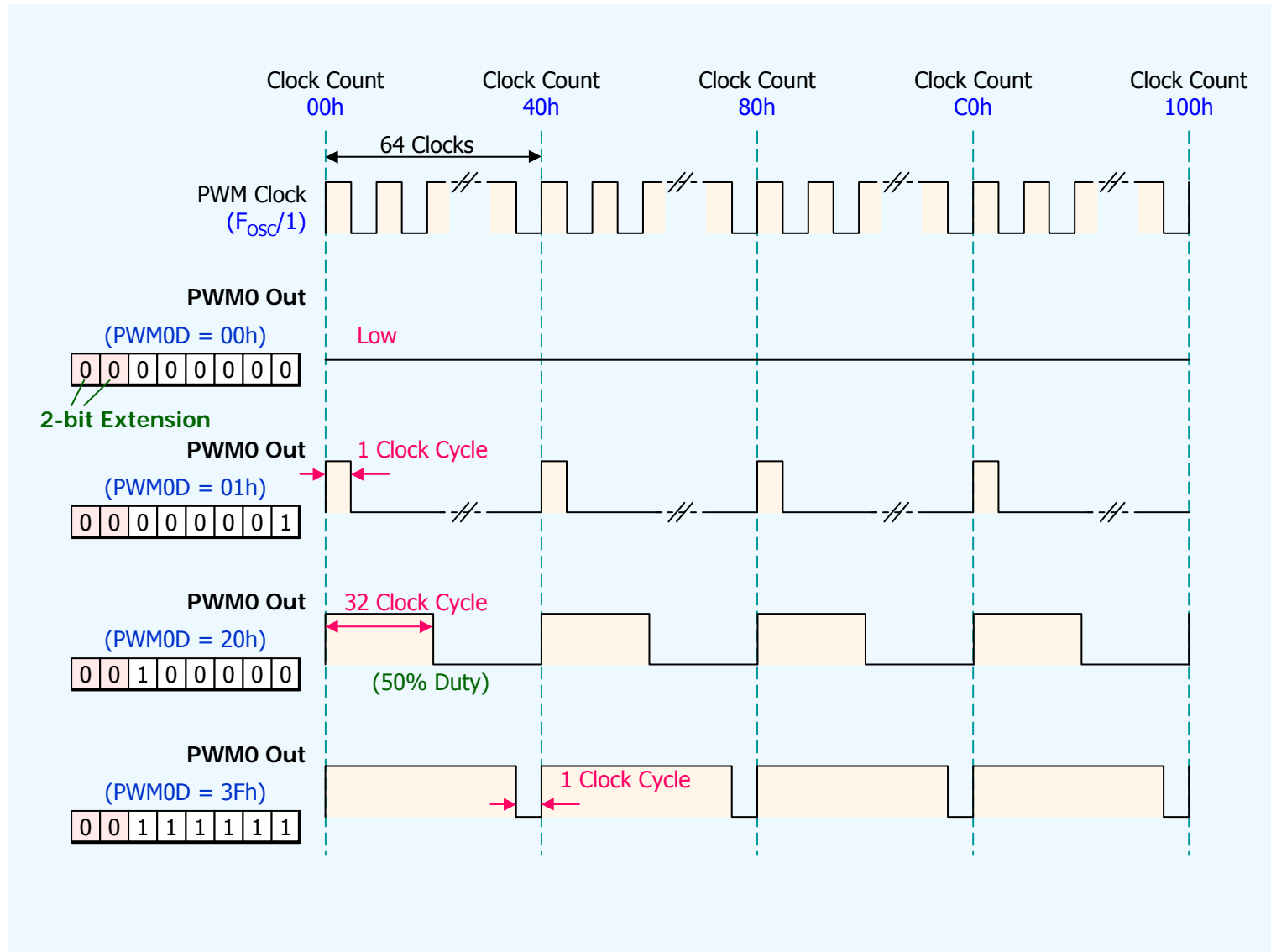
### ✓ PWM1D (DFh) : PWM1 Duty 데이터 레지스터

PWM1D.7	PWM1D.6	PWM1D.5	PWM1D.4	PWM1D.3	PWM1D.2	PWM1D.1	PWM1D.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)

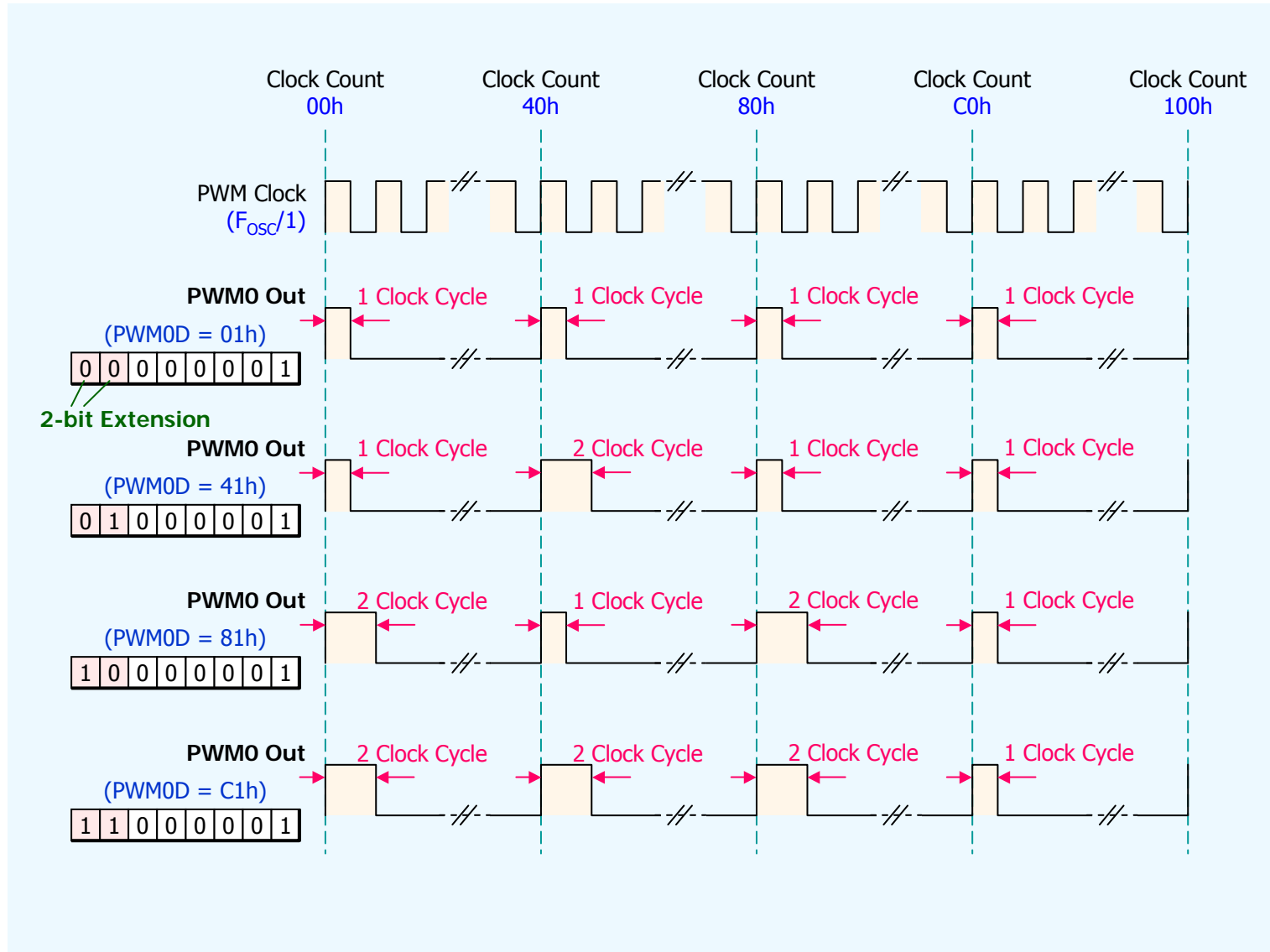
## 6.10. PWM : 8-bit 모드 펄스 생성



## 6.10. PWM : (2+6)-bit 모드



## 6.10. PWM : (2+6)-bit 모드의 확장 비트



## 6.11. ADC (아날로그 디지털 변환기)

- ◆ 4-channel 9-bit ADC (SAR Type)
- ◆ 최대 114ksps(초당 샘플 수) @  $F_{ADC} = 10\text{MHz} \ \& \ 5\text{V}$
- ◆ 최대 57ksps @  $F_{ADC} = 5\text{MHz} \ \& \ 3\text{V}$ .

✓ **ADCSEL (E2h) : ADC 클럭과 포트 제어 레지스터**

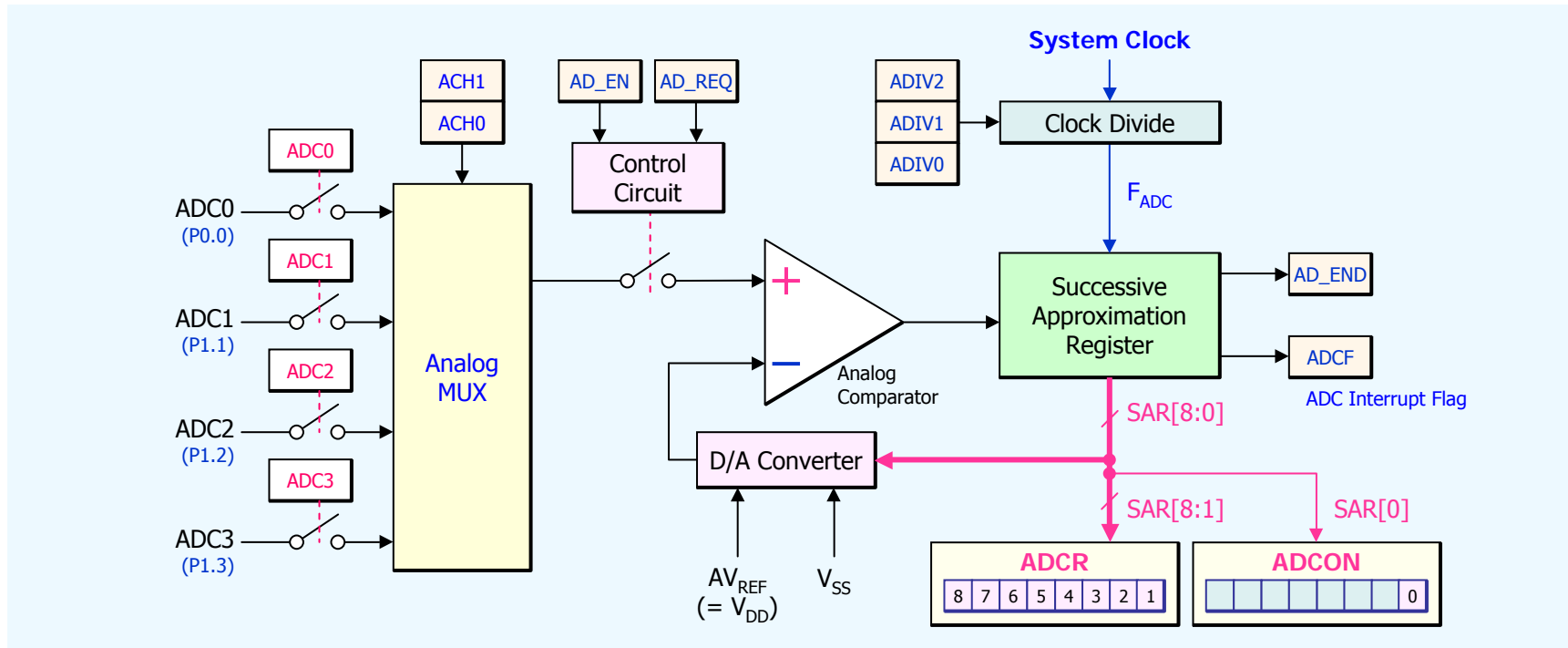
ADIV2	ADIV1	ADIV0	-	ADC3	ADC2	ADC1	ADC0
R/W(0)	R/W(0)	R/W(0)		R/W(0)	R/W(0)	R/W(0)	R/W(0)

✓ **ADCON (EFh) : ADC 제어 & ADC 결과값[0] 레지스터**

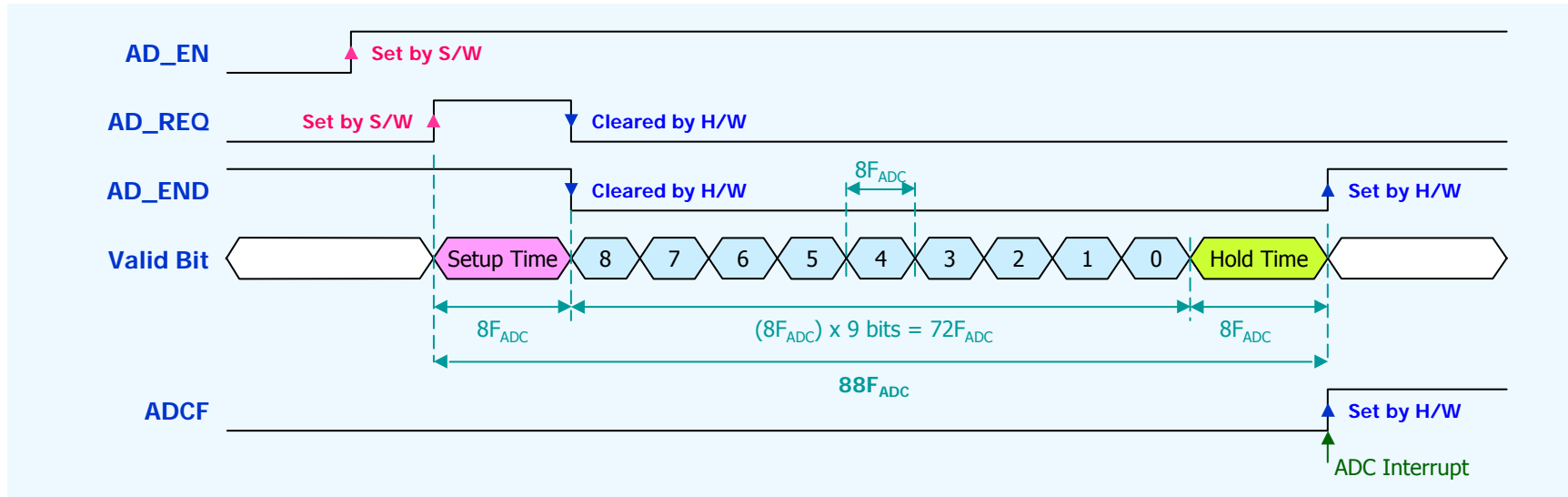
AD_EN	AD_REQ	AD_END	ADCF	ACH1	ACH0	-	SAR0
R/W(0)	R/W(0)	R(1)	R/W(0)	R/W(0)	R/W(0)		R/W(0)

✓ **ADCR (EEh) : ADC 결과값[8:1] 레지스터**

SAR8	SAR7	SAR6	SAR5	SAR4	SAR3	SAR2	SAR1
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)



## 6.11. ADC : 변환 타이밍



- ✓ **AD\_EN** : ADC 동작 허용 신호.  
소프트웨어에 의해 설정되거나 지워짐.
- ✓ **AD\_REQ** : ADC 변환 요청 시작 비트.  
S/W에 의해 설정되고 H/W에 의해 지워짐.  
이 비트는 각 샘플의 변환마다 설정되어야 한다.
- ✓ **AD\_END** : H/W에 의해 설정되거나 지워짐.  
변환이 시작될 때 지워짐  
변환이 끝나면 설정됨.
- ✓ **ADCF** : ADC 인터럽트 표시.  
H/W에 의해 설정되고 S/W에 의해 지워짐.  
ADC 인터럽트 루틴에서 지워야 함.

[ADC 변환표의 예]

OSC	분주 (ADCSEL[7:5])	$F_{ADC}$	$T (1/F_{ADC})$	1 샘플 변환 시간
20MHz @5V	000 (OSC/2)	10MHz	100ns	8.8us
	001 (OSC/4)	5MHz	200ns	17.6us
	010 (OSC/8)	2.5MHz	400ns	35.2us
	011 (OSC/16)	1.25MHz	800ns	70.4us
	100 (OSC/32)	0.625MHz	1.6us	140.8us
10MHz @3V/5V	000 (OSC/2)	5MHz	200ns	17.6us
	001 (OSC/4)	2.5MHz	400ns	35.2us
	010 (OSC/8)	1.25MHz	800ns	70.4us
	011 (OSC/16)	0.625MHz	1.6us	140.8us
	100 (OSC/32)	0.312MHz	3.2us	281.6us

## 6.12. 인터럽트: 13 발생원 / 4-단계 우선순위

- ◆ 인터럽트 발생원 : 타이머 0/1/2, UART, ADC, WDT, LVD, 외부 6개.
- ◆ 4-단계 인터럽트 우선순위

**[Interrupt Vector Address]**

Interrupt Sources	Address	Priority Level
LVD	0033h	-
INT0	0003h	4 Levels
TF0	000Bh	4 Levels
INT1	0013h	4 Levels
TF1	001Bh	4 Levels
RI+TI	0023h	4 Levels
TF2	002Bh	4 Levels
ADC	003Bh	4 Levels
INT2	0043h	2 Levels
INT3	004Bh	2 Levels
INT4	0053h	2 Levels
INT5	005Bh	2 Levels
WDT	0063h	2 Levels

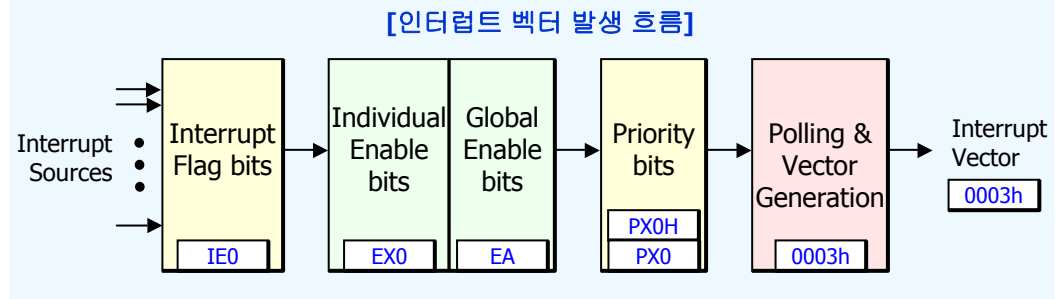
↑ HIGH PRIORITY  
↓ LOW

NMI (LVD, INT0, TF0, INT1, TF1, RI+TI, TF2, ADC)

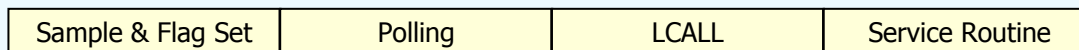
8052 (INT2, INT3, INT4, INT5, WDT)

\* SFR에 관련된 인터럽트 (Appendix B를 참조하라 : 설명)

✓ TCON (88h)	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
✓ EXIF (91h)	IE5	IE4	IE3	IE2	XT	-	-	BGS
✓ IE (A8h)	EA	EADC	ET2	ES	ET1	EX1	ET0	EX0
✓ EIE (E8h)	-	-	-	EWDT	EX5	EX4	EX3	EX2
✓ IP (B8h)	-	PADC	PT2	PS	PT1	PX1	PT0	PX0
✓ IPH (B7h)	-	PADCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
✓ EIP (F8h)	-	-	-	PWDT	PX5	PX4	PX3	PX2
✓ WDCON (D8h)	-	POR	EPFI	PRI	WDIF	WTRF	EWT	RWT



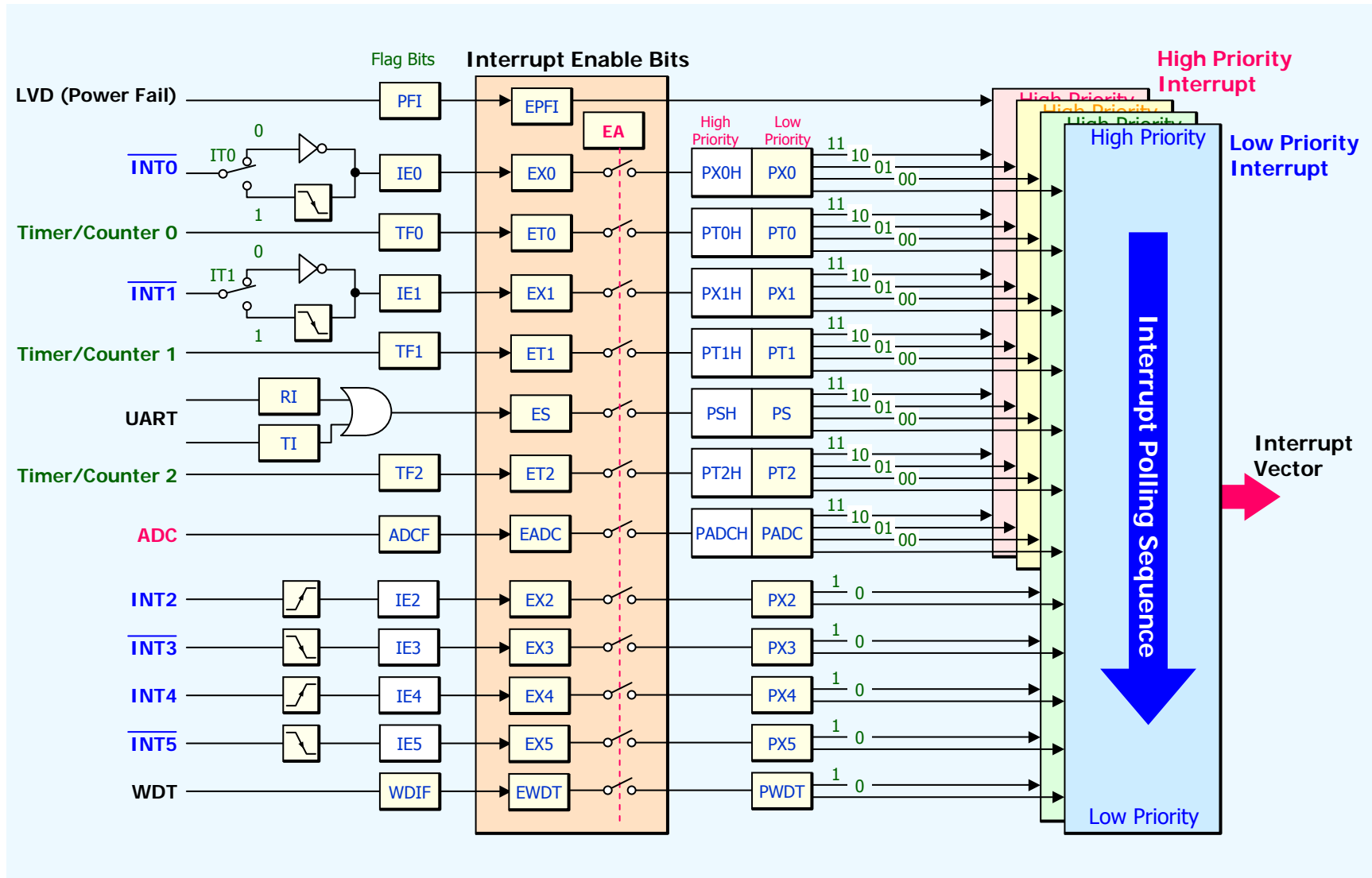
**[반응 순서]**



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



## 6.12. 기능적인 인터럽트 설명



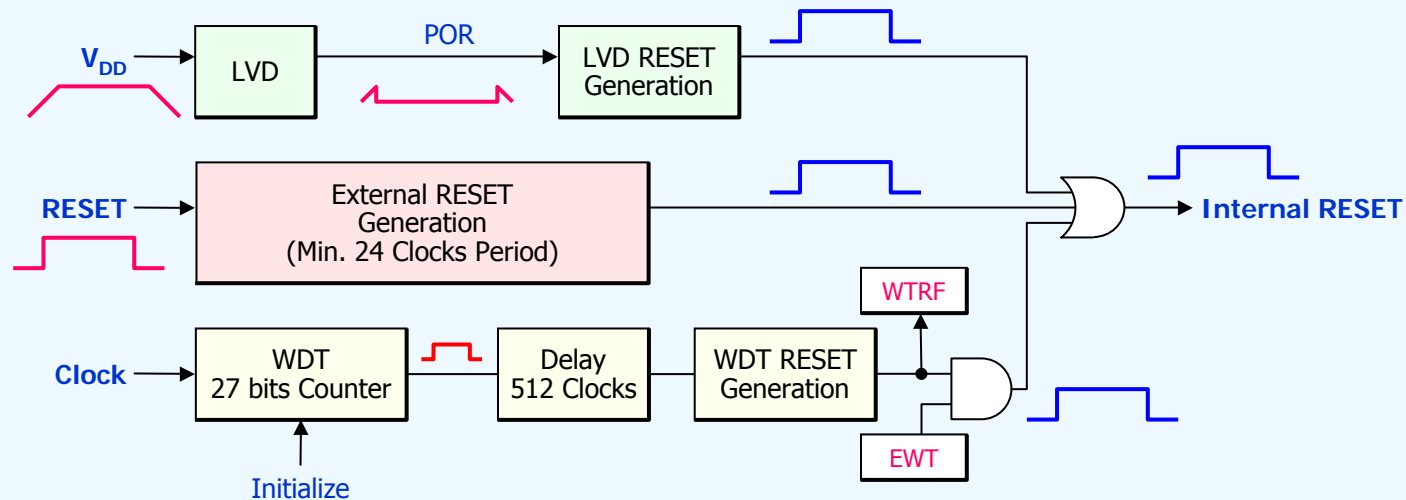
## 6.13. 리셋 회로 : 3개의 발생원

- ◆ LVD(POR) 리셋
  - ✓ 파워를 켤 때 파워 온 리셋.
  - ✓ 어떤 전압 아래일 때 파워 오류 리셋.
- ◆ 외부 리셋 핀
  - ✓ 리셋 핀은 24 클럭 주기 동안 "H"를 유지해야 한다.
- ◆ WDT 리셋 : 소프트웨어에 의해서 선택적으로 제어

✓ WDCON (D8h) : Watchdog & 전력 상태 레지스터

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

- WTRF : Watchdog 타이머 리셋 표시. 오직 S/W에 의해 소거.
- EWT : Watchdog 타이머 리셋 허용.



## 6.14. 클럭 회로

- ◆ 시스템 클럭 발생원
  - ✓ Crystal OSC
  - ✓ Oscillator

- ✓ **EXIF (91h)** : 외부 인터럽트 플래그 레지스터

IE5	IE4	IE3	IE2	XT	-	-	BGS
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R(1)			R/W(1)

- ✓ **STATUS (C5h)** : Crystal 상태 레지스터

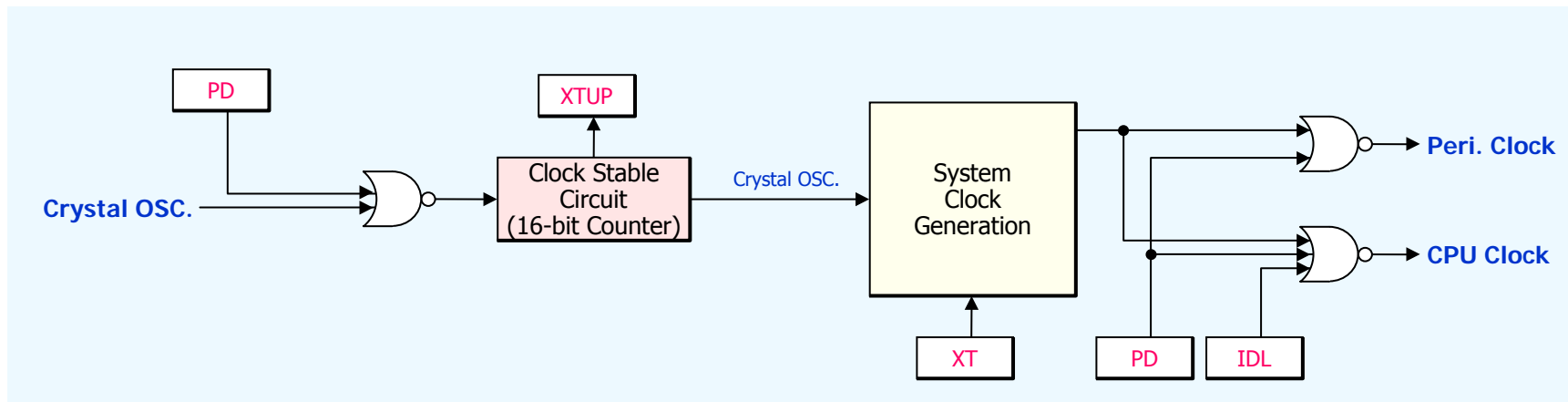
-	-	-	XTUP	-	-	-	-
			R(0)				

- ✓ **PMR (C4h)** : 파워 조절 제어 레지스터

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

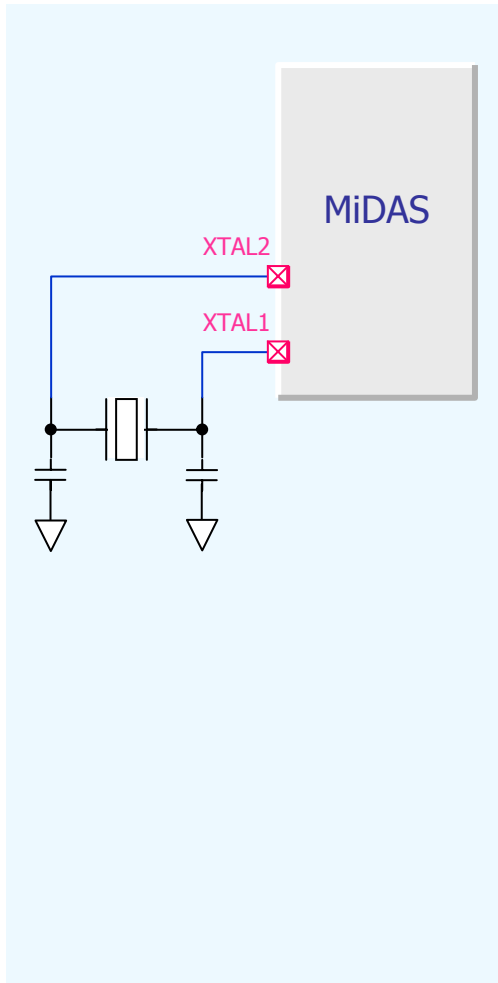
- ✓ **PCON (87h)** : 전원 제어 레지스터

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)

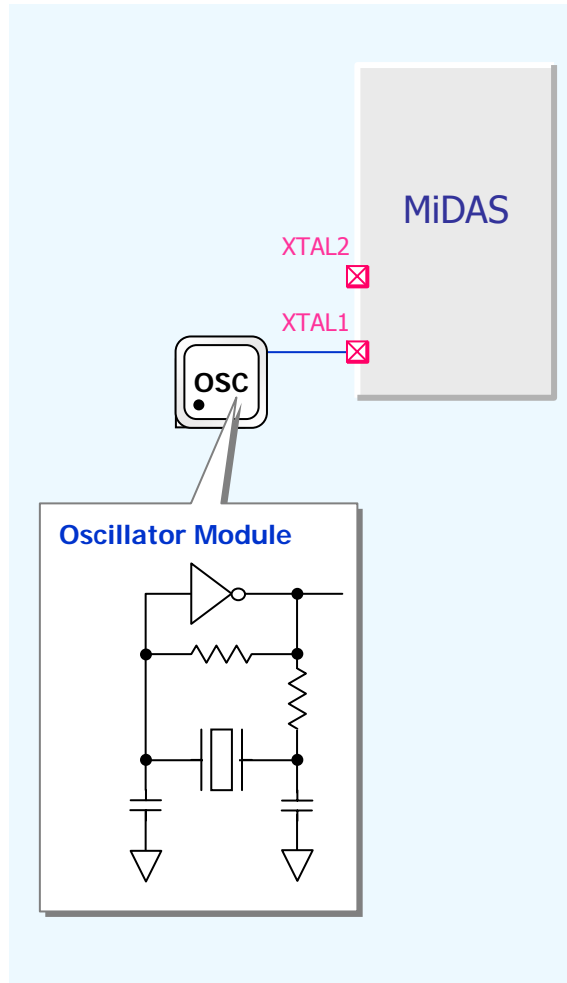


## 6.14. 클럭 회로 : 구조에 대한 가이드라인

### ◆ Crystal Oscillator

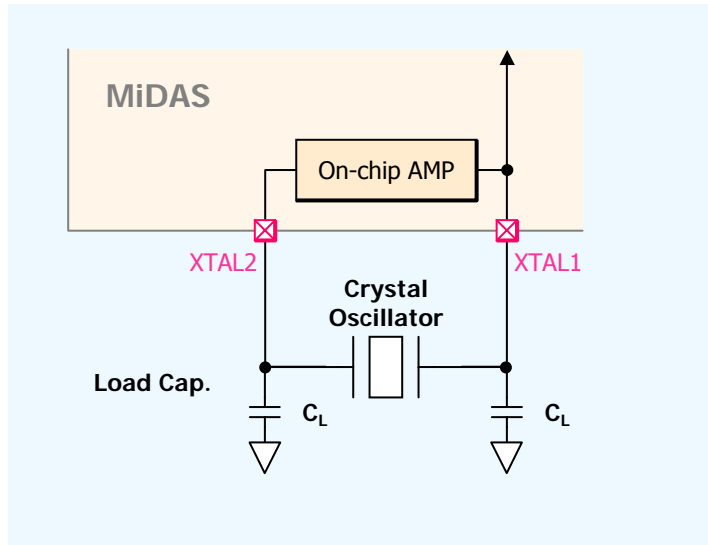


### ◆ Oscillator Module

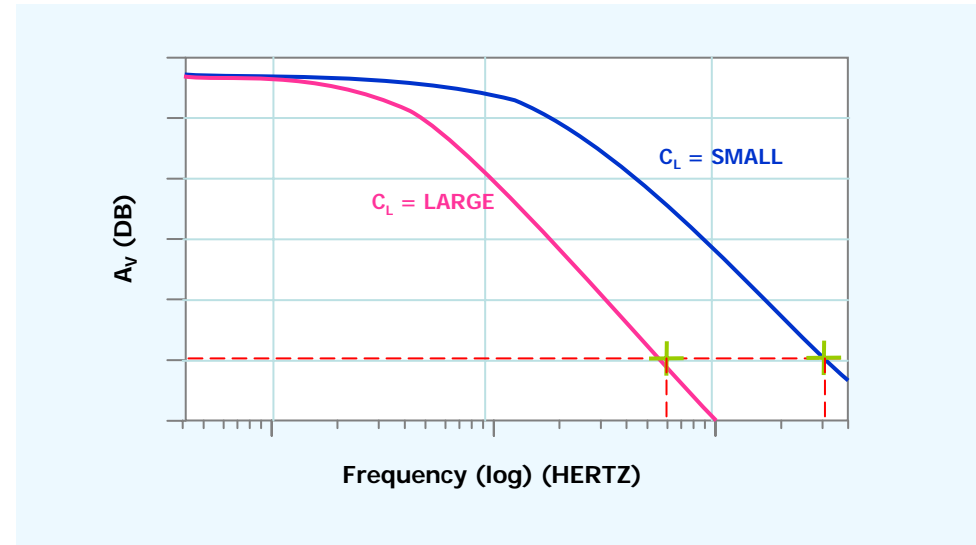


## 6.14. 클럭 회로 : Crystal 사용을 위한 가이드라인

### ◆ Crystal Oscillator & Load Capacitors



### ◆ Graph for Load Capacitor & Frequency



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

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

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

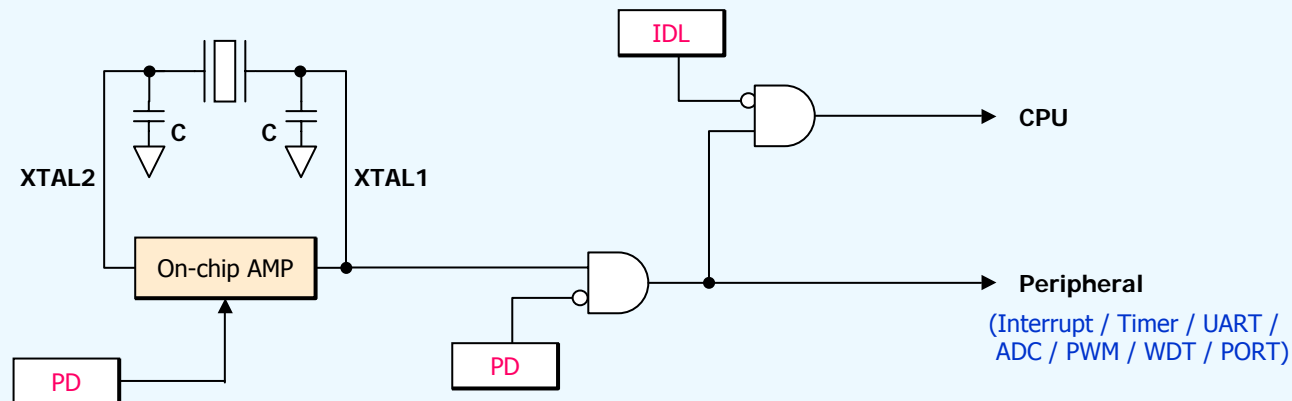
## 6.15. 전력 조절 : 3개의 모드

- ◆ **활동 모드** : CPU와 주변회로 동작중.
- ◆ **아이들 모드** : 오직 주변회로만 동작중.
  - ✓ 모든 종류의 인터럽트로 깨어난다. CPU는 활동을 재개한다.
  - ✓ 모든 종류의 리셋으로 깨어난다. CPU는 다시 시작한다.
- ◆ **정지 모드** : 모두 활동 정지.
  - ✓ 외부 인터럽트 0 또는 1로 깨어난다. (레벨 검출). CPU는 활동을 재개한다.  
외부 핀들은 최소한도 crystal이 안정화되는 시간 동안 '0'이 유지되어야 한다.
  - ✓ 모든 종류의 리셋으로 깨어난다. CPU는 다시 시작한다.

✓ **PCON (87h)** : 전력 제어 레지스터

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 : 정지 모드 (Power-down) 비트, 설정되면 정지 모드.
- IDL : 아이들 모드 비트, 설정되면 아이들 모드.



## 6.16. EPROM

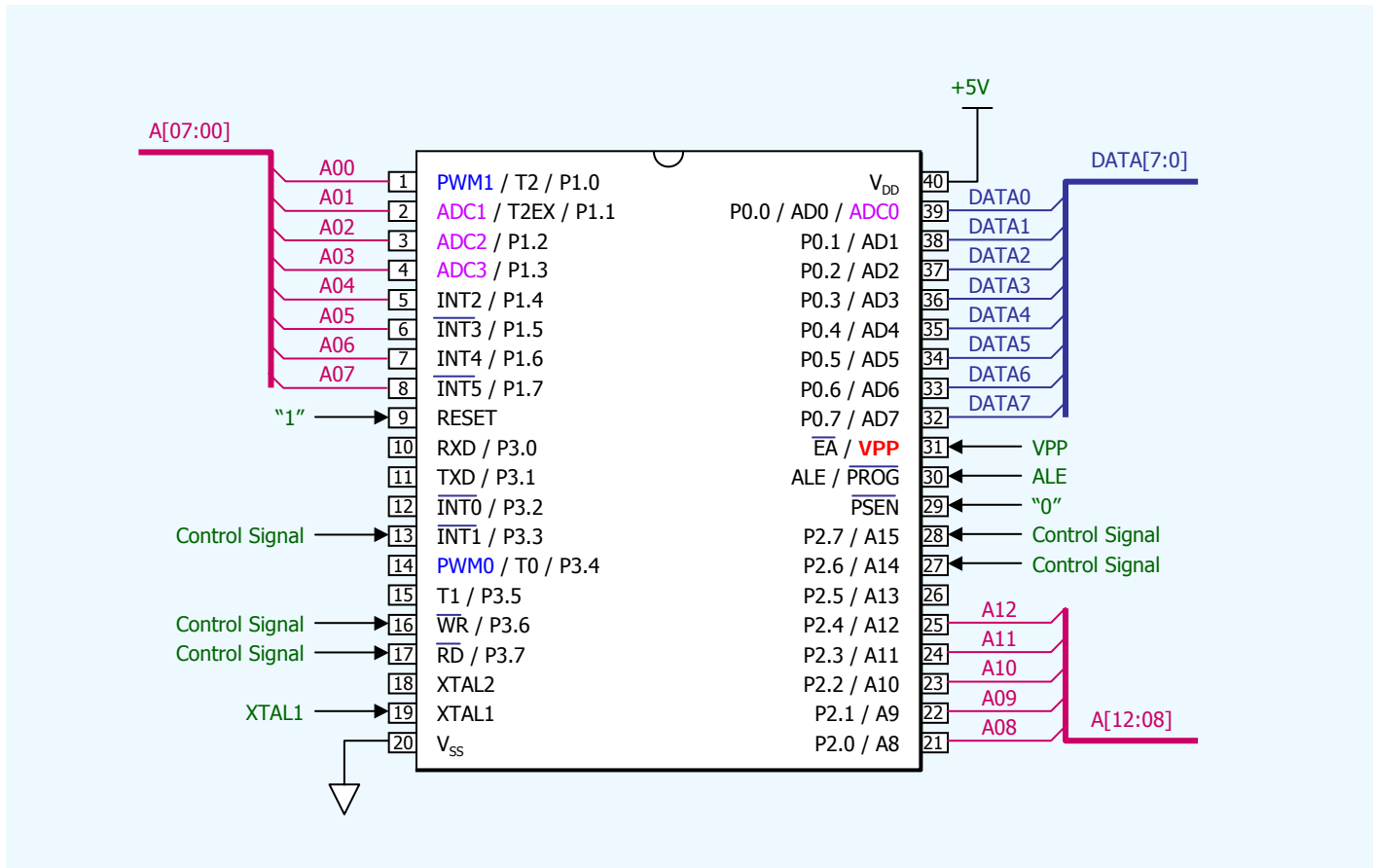
- ◆ 프로그래밍하고 검증하는 타이밍은 87C52와 매우 유사하다.
- ◆ 쓰기 전압 :  $11.5V \pm 0.5V$
- ◆ 쓰기 펄스 폭 : 최소 100us ~ 최대 500us
- ◆ 소거 시간 : 자외선 속에서 20분 ( $15mW/cm^2$ )
- ◆ 44-pin PLCC는 추가적인 adaptor가 필요하다.

Physical Layout의 EPROM 셀 구조]

Main Cell (4K / 8Kbytes)
Encryption Cell (64 bytes)
Lock Bit Cell (3 bits)
Signature Cell (3 bytes)

ROM Writer Interface										기능적인 설명
모드	RESET	PSEN	ALE	VPP	P2.6	P2.7	P3.3	P3.6	P3.7	
Program Code Data	H	L		11.5V	L	H	H	H	H	VPP = 11.5V and ALE Low Pulse
Verify Code Data	H	L	H	5V	L	L	L	H	H	Code = (Main byte) XNOR (Encryption byte)
Program Encryption Array	H	L		11.5V	L	H	H	L	H	VPP = 11.5V and ALE Low Pulse
Program Lock Bits	Bit 1	H	L		11.5V	H	H	H	H	LOCK1 = 1 : MOVC Instruction Disable. Further Programming Disable.
	Bit 2	H	L		11.5V	H	H	H	L	LOCK2 = 1 : Same as LOCK1 and Verify Disable.
	Bit 3	H	L		11.5V	H	L	H	L	LOCK3 = 1 : Same as LOCK1 and LOCK2. External Code Execution Disable.
Read Signature	H	L	H	5V	L	L	L	L	L	30h = C0h (CORERIVER) 31h = 87h (Product) 60h = 52h (EPROM Size)

## 6.16. EPROM : Pin configuration



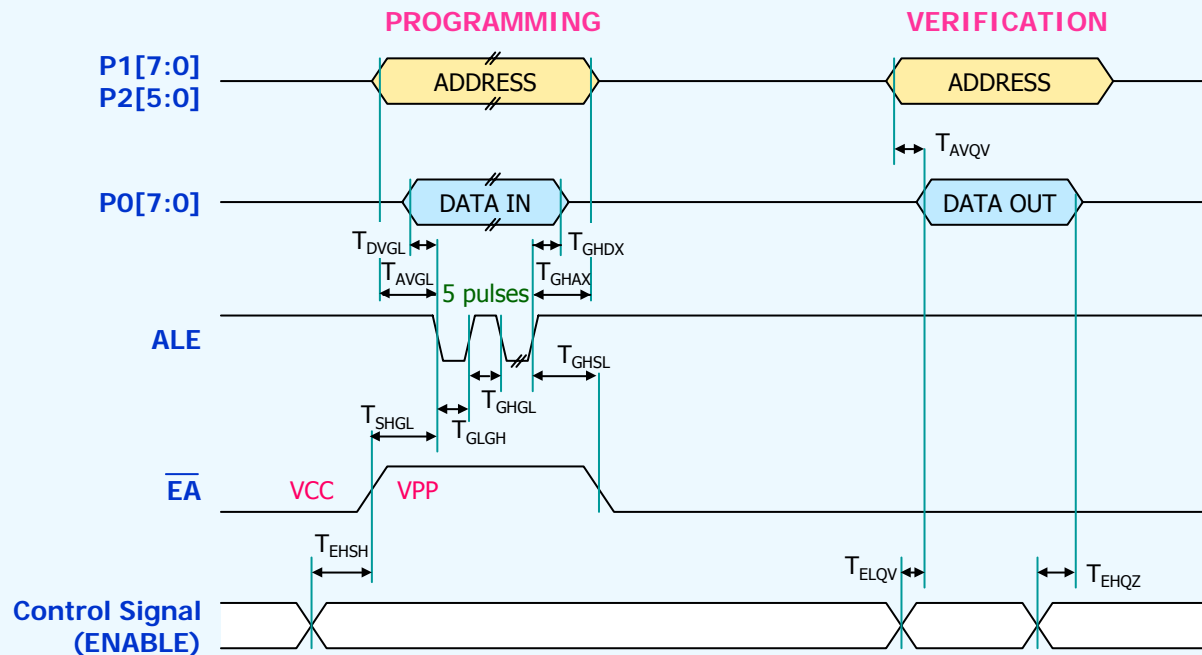


## 6.16. EPROM : 프로그램과 검증 타이밍

\* Oscillator Frequency : Frequency = 4 ~ 6 MHz ( $T_{CLCL} = 1/\text{Frequency}$ )

Parameter	Symbol	min.	Max.	Unit
Programming Supply Voltage	VPP	11.0	12.0	V
Programming Supply Current	IPP		75	mA
Address Setup to ALE Low	$T_{AVGL}$	$48 T_{CLCL}$		
Address hold after ALE High	$T_{GHAX}$	$48 T_{CLCL}$		
Data Setup to ALE Low	$T_{DVGL}$	$48 T_{CLCL}$		
Data Hold after ALE High	$T_{GHDX}$	$48 T_{CLCL}$		
ENABLE High to VPP	$T_{EHS}$	$48 T_{CLCL}$		

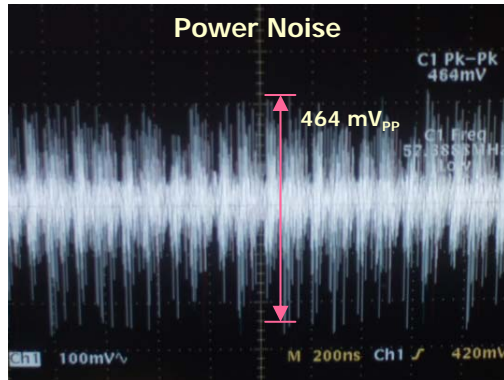
Parameter	Symbol	min.	Max.	Unit
VPP Setup to ALE Low	$T_{SHGL}$	10		$\mu\text{s}$
VPP Hold after ALE High	$T_{GHSL}$	10		$\mu\text{s}$
ALE Low Width	$T_{GLGH}$	90	110	$\mu\text{s}$
Address to Data Valid	$T_{AVQV}$		$48 T_{CLCL}$	
ENABLE Low to Data Valid	$T_{ELQV}$		$48 T_{CLCL}$	
Data Float after ENABLE	$T_{EHQZ}$	0	$48 T_{CLCL}$	
ALE High to ALE Low	$T_{GHGL}$	10		$\mu\text{s}$



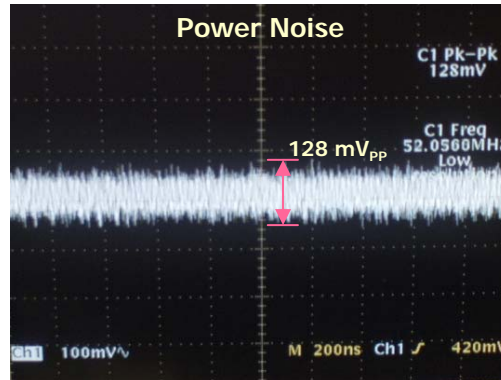
# 7. Strong Point I : 잡음 감소

Clock : 22.1184 MHz

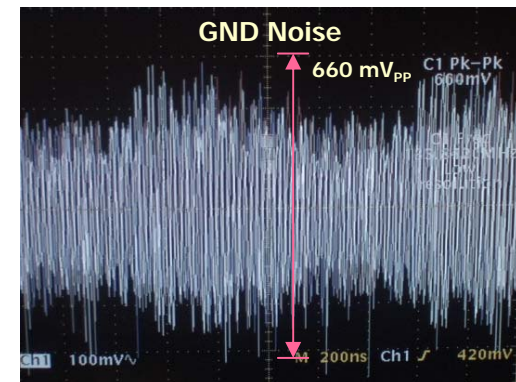
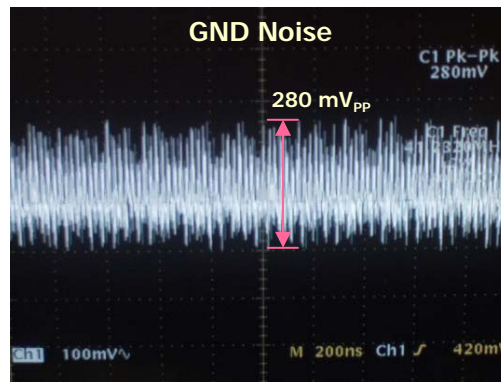
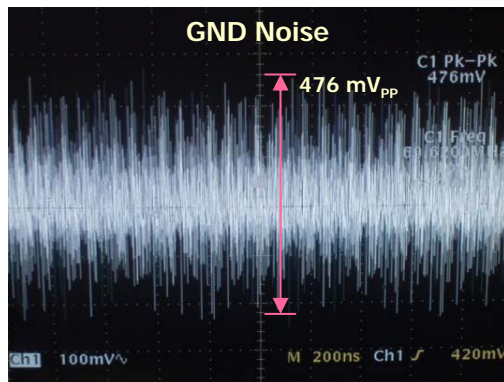
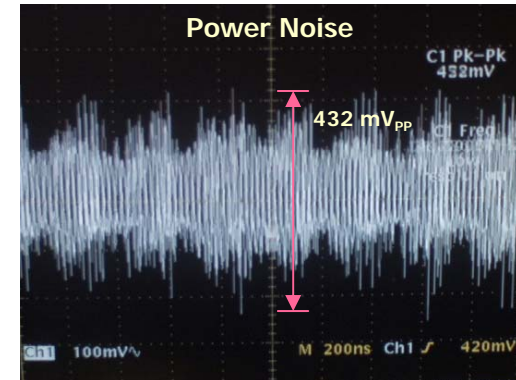
GC87C520G : ALE "ON"



GC87C520G : ALE "OFF"



Company A's 80C52



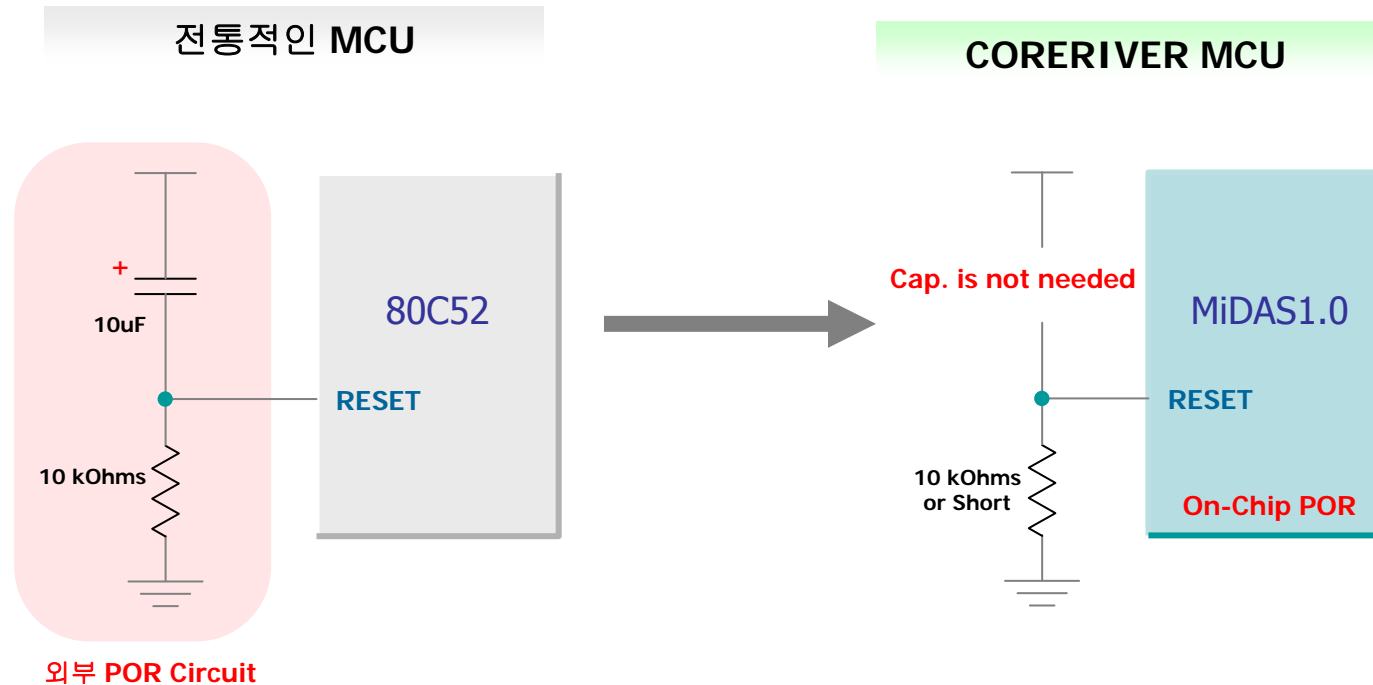
## 7. Strong Point I : 잡음 감소 (계속)

System Clock [MHz]	Noise	GC87C520G		Company A's 80C52
		ALE "ON"	ALE "OFF"	ALE always "ON"
11.0592	Power	410 mV <sub>pp</sub>	170 mV <sub>pp</sub>	360 mV <sub>pp</sub>
	Ground	550 mV <sub>pp</sub>	330 mV <sub>pp</sub>	500 mV <sub>pp</sub>
22.1184	Power	464 mV <sub>pp</sub>	128 mV <sub>pp</sub>	432 mV <sub>pp</sub>
	Ground	476 mV <sub>pp</sub>	280 mV <sub>pp</sub>	640 mV <sub>pp</sub>
6	Power	360 mV <sub>pp</sub>	170 mV <sub>pp</sub>	380 mV <sub>pp</sub>
	Ground	500 mV <sub>pp</sub>	330 mV <sub>pp</sub>	480 mV <sub>pp</sub>

- GC87C520G는 필요없는 ALE 신호의 진동을 삭제하여 EMI를 향상시킬 수 있다.
- ALEOFF bit (SFR PMR.2)를 사용하여 ALE 신호를 허용/금지시킬 수 있다.
- 또는 EMI를 유지시키면서 필요 없는 decoupling capacitor를 제거하여 시스템 비용을 줄일 수 있다.

\* 이 실험 결과는 응용 시스템의 설계 형태에 따라 변할 수 있다.

## 7. Strong Point II : 칩상 POR



- 칩상 POR(파워 온 리셋)은 불필요한 커패시터나 저항까지도 제거하여 시스템 비용을 줄일 수 있다.
- 시스템 전력의 기울기는 " $0.0V/us < V_{DD} \text{ slope} \leq 0.5V/us$ "이어야 한다.  
(즉, 전압이 상승할 때 마이너스 기울기가 되는 기간이 없어야 한다.)

## 8. Absolute Maximum Ratings

Items	Ranges
Ground에 대한 핀 전압	-0.3V 에서 ( $V_{DD}+0.5V$ )
Ground에 대한 $V_{DD}$ 전압	-0.3V 에서 6.0V
출력 전압	-0.3V 에서 ( $V_{DD}+0.3V$ )
동작 온도	-20 °C 에서 85 °C
데이터 기억 온도	-55 °C 에서 +125 °C
Soldering 온도	10초 동안 160 °C

## 9. DC 특성

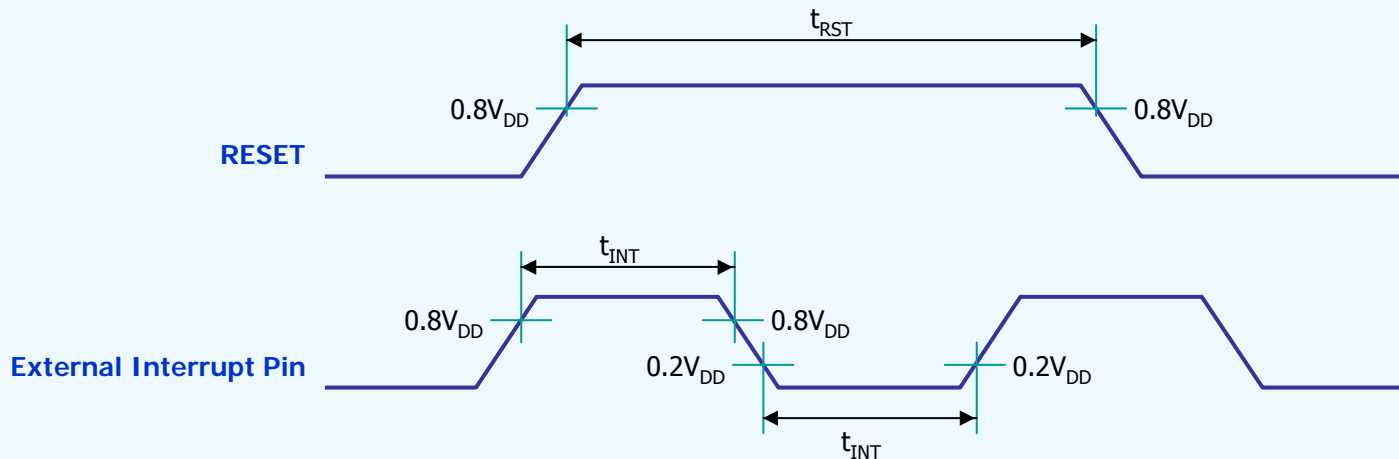
\* 별도의 표시가 없으면  $TA = -20\text{ }^{\circ}\text{C} \sim +85\text{ }^{\circ}\text{C}$ ,  $V_{DD} = 2.7\text{V} \sim 5.5\text{V}$ .

Parameter	Symbol	Pin	Conditions	Value			Unit
				Min.	Typ.	Max.	
Input Low Voltage	$V_{IL}$	Except EA	$V_{DD} = 2.7\text{V} \sim 5.5\text{V}$	-0.5	-	$0.2V_{DD} - 0.1$	V
	$V_{IL1}$	EA		-0.5	-	$0.2V_{DD} - 0.3$	
Input high Voltage	$V_{IH}$	ALE, PSEN, P0, P1, P2, P3, P4	$V_{DD} = 2.7\text{V} \sim 5.5\text{V}$	2.0	-	$V_{DD} + 0.5$	V
	$V_{IH1}$	EA, RESET, XTAL1		$0.7V_{DD}$	-	$V_{DD} + 0.5$	
Output Low Voltage	$V_{OL}$	P1, P2, P3, P4	$V_{DD} = 2.7\text{V} \sim 5.5\text{V} (I_{OL} = 1.6\text{mA})$	-	-	0.45	V
	$V_{OL1}$	P0, ALE, PSEN	$V_{DD} = 2.7\text{V} \sim 5.5\text{V} (I_{OL} = 3.2\text{mA})$	-	-	0.45	
Output High Voltage	$V_{OH}$	P1, P2, P3, P4	$V_{DD} = 2.7\text{V} \sim 5.5\text{V} (I_{OH} = -60\mu\text{A})$	$0.75V_{DD}$	-	-	V
			$V_{DD} = 2.7\text{V} \sim 5.5\text{V} (I_{OH} = -25\mu\text{A})$	$0.76V_{DD}$	-	-	
			$V_{DD} = 2.7\text{V} \sim 5.5\text{V} (I_{OH} = -10\mu\text{A})$	$0.78V_{DD}$	-	-	
	$V_{OH1}$	ALE, PSEN, P0 (External Access)	$V_{DD} = 2.7\text{V} \sim 5.5\text{V} (I_{OH} = -800\mu\text{A})$	$0.75V_{DD}$	-	-	V
			$V_{DD} = 2.7\text{V} \sim 5.5\text{V} (I_{OH} = -300\mu\text{A})$	$0.78V_{DD}$	-	-	
			$V_{DD} = 2.7\text{V} \sim 5.5\text{V} (I_{OH} = -80\mu\text{A})$	$0.8V_{DD}$	-	-	
Logical 0 Input Current	$I_{IL}$	P0, P1, P2, P3, P4	$V_{DD} = 5.5\text{V} (V_{IN} = 0.45\text{V})$	-	-	-1	$\mu\text{A}$
Logical 1 to 0 Transition Current	$I_{TL}$	P0, P1, P2, P3, P4	$V_{DD} = 5\text{V} \pm 10\% (V_{IN} = 2\text{V})$	-	-	-650	$\mu\text{A}$
Input Leakage Current	$I_{IL}$	All pin except XTAL1, XTAL2	$V_{IN} = V_{IH}$ or $V_{IL}$	-	-	$\pm 1$	$\mu\text{A}$
Reset Pull-down Resistor	$R_{RST}$	RESET	-	25	-	100	$\text{k}\Omega$
Pin Capacitance	$C_{IO}$	All	$V_{DD} = 5\text{V}$	-	10	-	pF

# 10. AC 특성

\* 별도의 표시가 없으면,  $T_A = -20\text{ }^\circ\text{C} \sim +85\text{ }^\circ\text{C}$ .

Parameter	Symbol	Pin	Conditions	Value			Unit
				Min.	Typ.	Max.	
동작 주파수	$F_{OSC}$	XTAL1, XTAL2	$V_{DD} = 5V \pm 10\%$	1	-	40	MHz
			$V_{DD} = 3V \pm 10\%$	1	-	20	
리셋 입력의 펄스 폭	$t_{RST}$	RESET	$V_{DD} = 5V \pm 10\%$	24	-	-	$F_{OSC}$
			$V_{DD} = 3V \pm 10\%$	24	-	-	
외부 인터럽트 입력의 펄스 폭	$t_{INT}$	External Interrupt	$V_{DD} = 5V \pm 10\%$	4	-	-	$F_{OSC}$
			$V_{DD} = 3V \pm 10\%$	4	-	-	

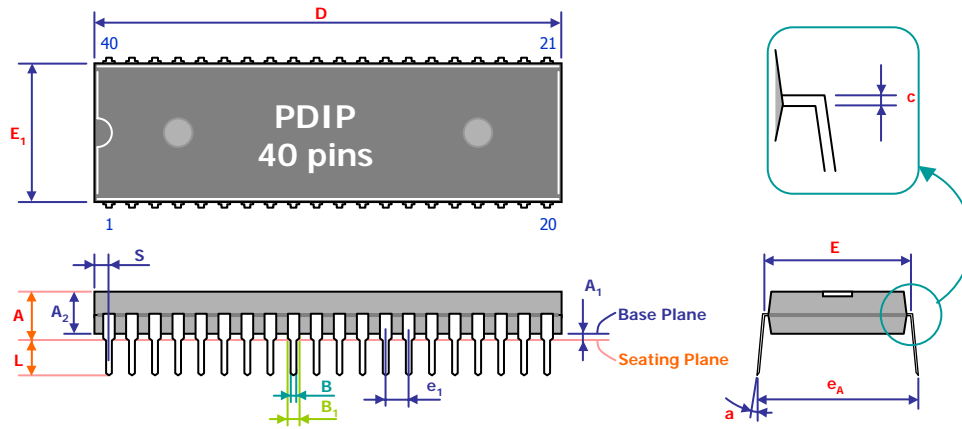


## 11. ADC 특성

Parameter	Symbol	Conditions	Value			Unit	
			Min.	Typ.	Max.		
Supply Voltage	$V_{DDADC}$	-	2.7	-	5.5	V	
Input Voltage	$V_{INADC}$	-	$V_{SS}$	-	$V_{DD}$	V	
Resolution	$RES_{ADC}$	-	-	9	-	bit	
Operating Frequency	$F_{ADC}$	$V_{DD} = 4.5V \sim 5.5V$ $V_{DD} = 2.7V \sim 3.3V$	-	-	10 5	MHz	
Conversion Time	$t_{ADC}$	-	-	$88 / F_{ADC}$	-	s	
Overall Accuracy	$OA_{ADC}$	$V_{DD} = 5V, F_{ADC} = 10MHz$ $V_{DD} = 3V, F_{ADC} = 5MHz$	-	$\pm 2$	$\pm 4$	LSB	
Integral Nonlinearity	$INL_{ADC}$	$V_{DD} = 5V, F_{ADC} = 10MHz$ $V_{DD} = 3V, F_{ADC} = 5MHz$	-	$\pm 2$	$\pm 4$	LSB	
Differential Nonlinearity	$DNL_{ADC}$	$V_{DD} = 5V, F_{ADC} = 10MHz$ $V_{DD} = 3V, F_{ADC} = 5MHz$	-	$\pm 0.5$	$\pm 1$	LSB	
Zero Input Error	$ZIE_{ADC}$	$V_{DD} = 5V, F_{ADC} = 10MHz$ $V_{DD} = 3V, F_{ADC} = 5MHz$	-	$\pm 2$	$\pm 4$	LSB	
Full Scale Error	$FSE_{ADC}$	$V_{DD} = 5V, F_{ADC} = 10MHz$ $V_{DD} = 3V, F_{ADC} = 5MHz$	-	$\pm 2$	$\pm 4$	LSB	
Analog Input Capacitance		$C_{INADC}$	-	10	15	pF	
ADC Current	Active	$I_{ADC}$	$V_{DD} = 5V, F_{ADC} = 10MHz$	-	1	2	mA
			$V_{DD} = 3V, F_{ADC} = 5MHz$	-	0.3	0.6	
	Power-down		$V_{DD} = 5V$	-	-	500	nA



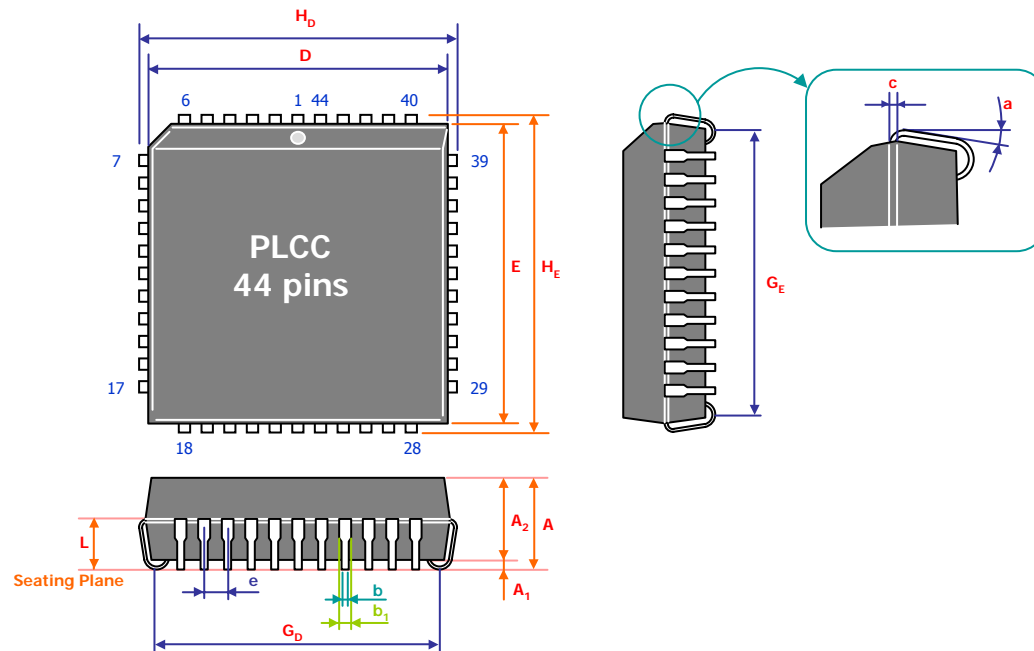
# 12. Package Dimensions : 40-PDIP & 44-PLCC



Symbol	Dimension in Inches			Dimension in mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
A	-	-	0.200	-	-	5.080
$A_1$	0.015	-	-	0.381	-	-
$A_2$	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
$B_1$	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	2.045	2.055	2.075	51.943	52.197	52.705
E	0.590	0.600	0.610	14.986	15.240	15.494
$E_1$	0.530	0.545	0.550	13.720	13.840	13.970
$e_1$	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°	-	15°	0°	-	15°
$e_A$	0.630	0.650	0.670	16.000	16.510	17.010
S	-	-	0.090	-	-	2.286

**Notes:**

1. Dimension D Max. & S include mold flash or tie bar Burns.
2. Dimension  $E_1$  dose not include interlead flash.
3. Dimension D &  $E_1$  include mold mismatch and are determined at the mold parting line.
4. Dimension  $B_1$  does not include dambar protrusion/intrusion.
5. General appearance spec. should be based on final visual inspection spec.

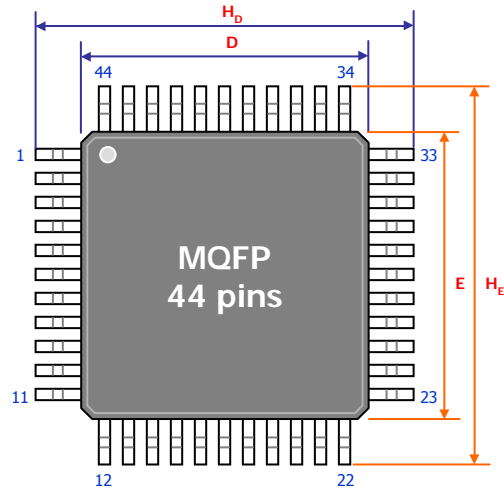


Symbol	Dimension in Inches			Dimension in mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
A	0.165	-	0.180	4.191	-	4.572
$A_1$	0.020	-	-	0.508	-	-
$A_2$	0.145	0.150	0.155	3.683	3.810	3.937
$b_1$	0.026	0.028	0.032	0.660	0.711	0.813
b	0.013	0.017	0.021	0.330	0.432	0.533
c	0.008	0.010	0.014	0.203	0.254	0.356
D	0.648	0.650	0.658	16.460	16.510	16.710
E	0.648	0.650	0.658	16.460	16.510	16.710
e	0.050 BSC			1.27 BSC		
$G_D$	0.590	0.610	0.630	14.986	15.494	16.002
$G_E$	0.590	0.610	0.630	14.986	15.494	16.002
$H_D$	0.680	0.690	0.700	17.272	17.526	17.780
$H_E$	0.680	0.690	0.700	17.272	17.526	17.780
L	0.090	0.100	0.120	2.296	2.540	3.048

**Notes:**

1. Dimension D \* E do not include interlead flash.
2. Dimension  $b_1$  dose not include dambar protrusion/intrusion.
3. Controlling dimension: Inches
4. General appearance spec. should be based on final visual inspection spec.

## 12. Package Dimensions : 44-MQFP

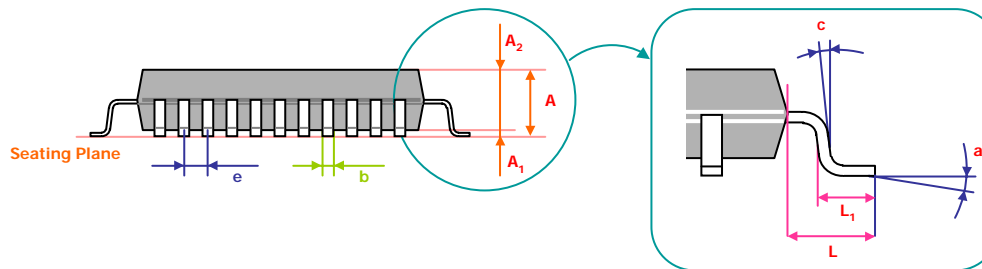


[44-MQFP]

Symbol	Dimension in Inches			Dimension in mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
A	-	-	0.091	-	-	2.30
A <sub>1</sub>	0.002	-	0.006	0.05	-	0.15
A <sub>2</sub>	0.077	0.081	0.085	1.95	2.05	2.15
b	0.012	0.015	0.018	0.30	0.37	0.45
D	0.394 BSC			10.00 BSC		
E	0.394 BSC			10.00 BSC		
e	0.031 BSC			0.80 BSC		
H <sub>b</sub>	0.520 BSC			13.20 BSC		
H <sub>e</sub>	0.520 BSC			13.20 BSC		
L	-	0.063	-	-	1.60	-
L <sub>1</sub>	0.024	0.031	0.039	0.60	0.80	1.00
a	0°	-	8°	0°	-	8°
c	0°	-	-	0°	-	-

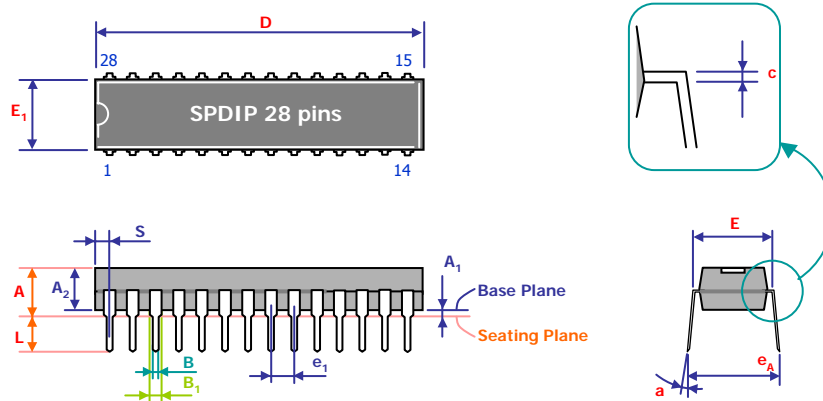
**Notes:**

1. Dimension D \* E do not include interlead flash.
2. Controlling dimension: Inches
3. General appearance spec. should be based on final visual inspection spec.



# 12. Package Dimensions : 28-SPDIP/SOIC

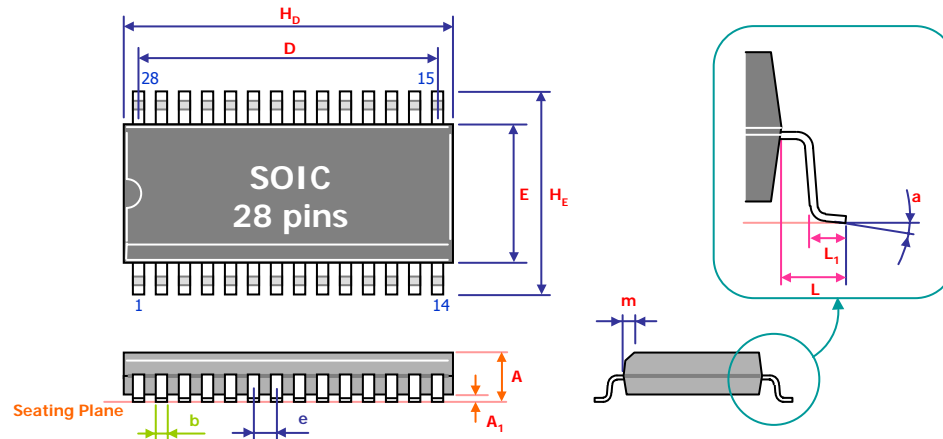
## [28-SPDIP]



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	1.445	1.455	1.475	36.703	36.907	37.465
E	0.290	0.300	0.310	7.366	7.62	7.874
E <sub>1</sub>	0.530	0.545	0.550	13.720	13.840	13.970
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°	-	15°	0°	-	15°
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.

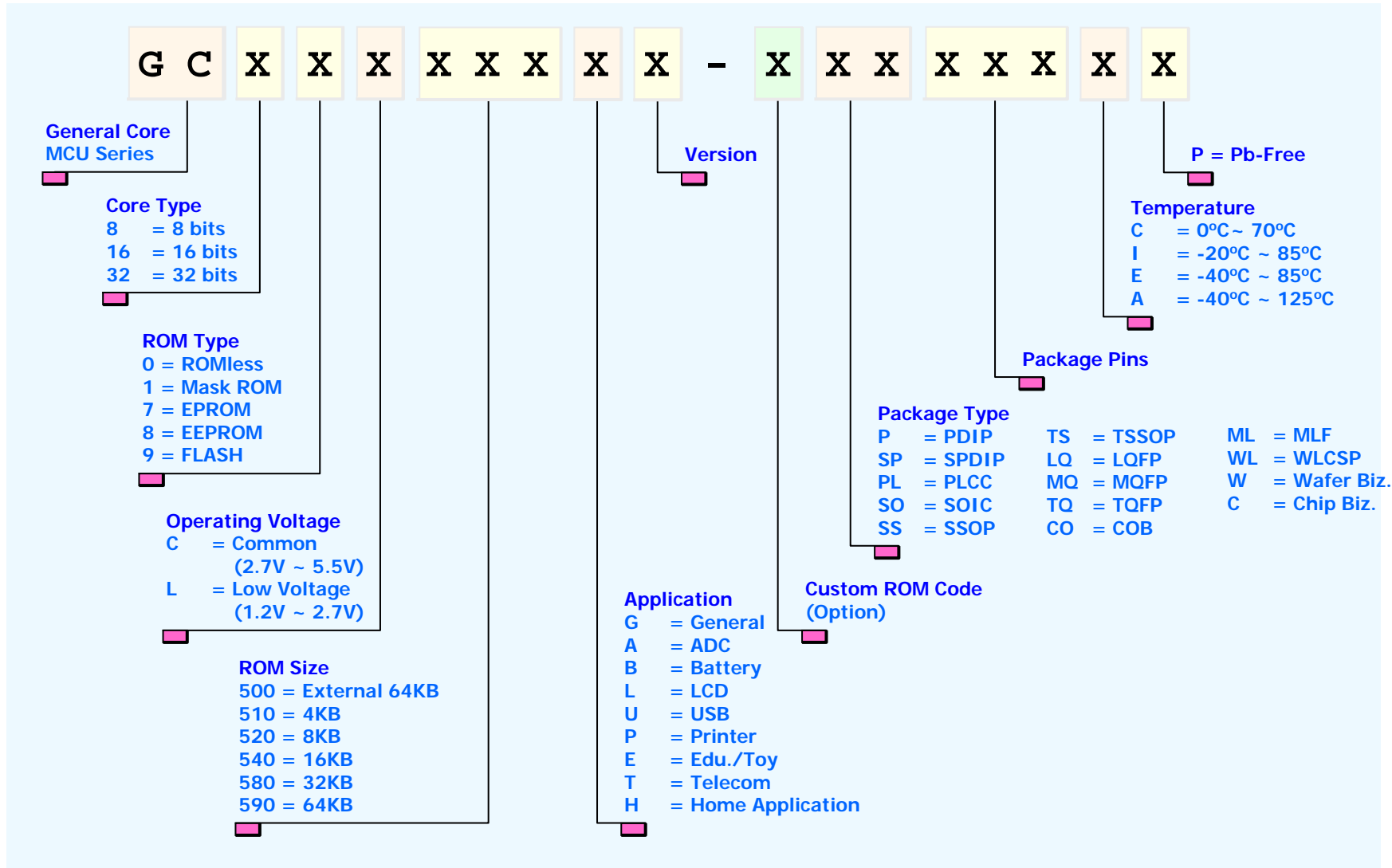


Symbol	Dimension in Inches			Dimension in mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
A	0.093	0.099	0.104	2.35	2.45	2.65
A <sub>1</sub>	0.004	0.008	0.012	0.10	0.20	0.30
b	0.014	0.016	0.019	0.35	0.42	0.49
D	-	0.65	-	-	16.51	-
E	0.291	0.295	0.299	7.40	7.50	7.60
H <sub>b</sub>	0.697	0.705	0.713	17.70	17.90	18.10
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

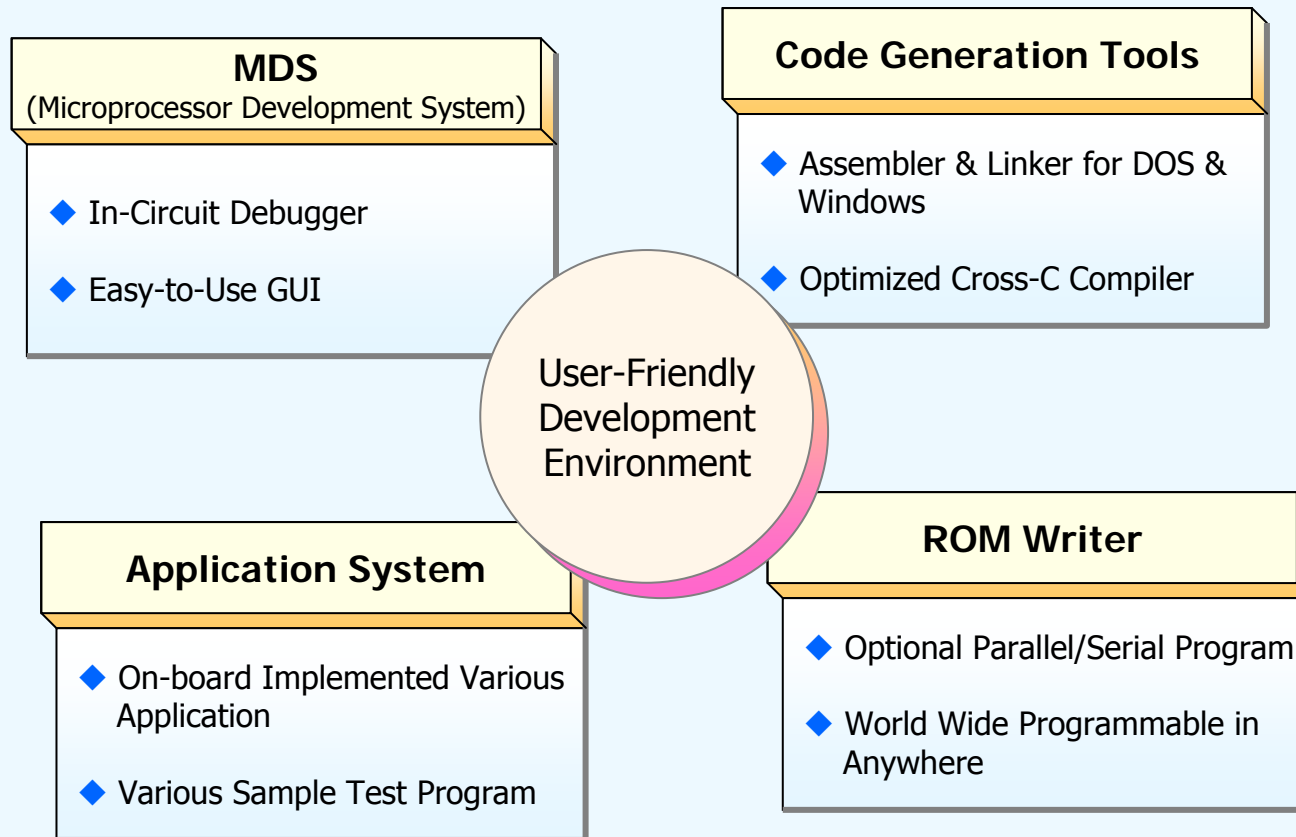
**Notes:**

1. Dimension D 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. Product Numbering System



## 14. 지원 툴



# Appendix A : instruction set (1/18)

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

1 cycle = 4 clocks

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

0 0 1 0 1 r r r

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

0 0 1 0 0 1 1 i

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

0 0 1 0 0 1 0 1 direct addr

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

0 0 1 0 0 1 0 0 immediate data

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

0 0 1 1 1 r r r

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

0 0 1 1 0 1 1 i

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

0 0 1 1 0 1 0 1 direct addr

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

0 0 1 1 0 1 0 0 immediate data

# Appendix A : instruction set (2/18)

## 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

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

1 0 0 1 1 r r r

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

1 0 0 1 0 1 1 i

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

1 0 0 1 0 1 0 1 direct addr

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

1 0 0 1 0 1 0 0 immediate data

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

0 0 0 0 0 1 0 0

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

0 0 0 0 1 r r r

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

0 0 0 0 0 1 1 i

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

0 0 0 0 0 1 0 1 direct addr

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

1 0 1 0 0 0 1 1

# Appendix A : instruction set (3/18)

## 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
DEC	DPTR
Operation :	(DPTR) ← (DPTR) - 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

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

1 0 1 0 0 1 0 1

## MUL AB

### Multiply

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

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

1 0 1 0 0 1 0 0

## DIV AB

### Divide

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

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

1 0 0 0 0 1 0 0



# Appendix A : instruction set (4/18)

DA A

## Decimal-adjust Accumulator for Addition

Operation : IF  $[(A_{3-0}) > 9] \vee [(AC)=1]$   
 THEN  $(A_{3-0}) \leftarrow (A_{3-0}) + 6$   
 IF  $[(A_{7-4}) > 9] \vee [(C)=1]$   
 THEN  $(A_{7-4}) \leftarrow (A_{7-4}) + 6$

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

1 1 0 1 0 1 0 0

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

## Logical AND for byte variables

ANL A, Rn

Operation :  $(A) \leftarrow (A) \wedge (Rn)$

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

0 1 0 1 1 r r r

ANL A, @Ri

Operation :  $(A) \leftarrow (A) \wedge ((Ri))$

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

0 1 0 1 0 1 1 i

ANL A, direct

Operation :  $(A) \leftarrow (A) \wedge (\text{direct})$

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

0 1 0 1 0 1 0 1 direct addr

ANL A, #date

Operation :  $(A) \leftarrow (A) \wedge \text{data}$

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

0 1 0 1 0 1 0 0 immediate data

ANL direct, A

Operation :  $(\text{direct}) \leftarrow (\text{direct}) \wedge (A)$

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

0 1 0 1 0 0 1 0 direct addr

ANL direct, #data

Operation :  $(\text{direct}) \leftarrow (\text{direct}) \wedge \text{data}$

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

ANL C, /bit

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

**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, #date

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: 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

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

0 1 0 0 1 r r r

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

0 1 0 0 0 1 1 i

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

0 1 0 0 0 1 0 1 direct addr

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

0 1 0 0 0 1 0 0 immediate data

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

0 1 0 0 0 0 1 0 direct addr

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

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

# Appendix A : instruction set (6/18)

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

**Logical OR for byte variables**

ORL	C, bit
<b>Operation :</b>	(C) $\leftarrow$ (C) $\vee$ (bit)
ORL	C, /bit
<b>Operation :</b>	(C) $\leftarrow$ (C) $\vee$ $\sim$ (bit)

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

**Logical Exclusive-OR for byte variables**

XRL	A, Rn
<b>Operation :</b>	(A) $\leftarrow$ (A) $\oplus$ (Rn)
XRL	A, @Ri
<b>Operation :</b>	(A) $\leftarrow$ (A) $\oplus$ ((Ri))
XRL	A, direct
<b>Operation :</b>	(A) $\leftarrow$ (A) $\oplus$ (direct)
XRL	A, #date
<b>Operation :</b>	(A) $\leftarrow$ (A) $\oplus$ data
XRL	direct, A
<b>Operation :</b>	(direct) $\leftarrow$ (direct) $\oplus$ (A)
XRL	direct, #data
<b>Operation :</b>	(direct) $\leftarrow$ (direct) $\oplus$ data

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

0 1 1 1 0 0 1 0 bit addr

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

1 0 1 0 0 0 0 0 bit addr

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

0 1 1 0 1 r r r

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

0 1 1 0 0 1 1 i

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

0 1 1 0 0 1 0 1 direct addr

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

0 1 1 0 0 1 0 0 immediate data

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

0 1 1 0 0 0 1 0 direct addr

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

0 1 1 0 0 0 1 1 direct addr immediate Data

# Appendix A : instruction set (7/18)

**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)     $\leftarrow$  ~(A)

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

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

**CPL    <bit>**

**Complement bit**

CPL    C

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

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

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

CPL    bit

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

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

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

**RLC     A**

**Rotate Accumulator Left through the Carry flag**

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

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

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

**RR      A**

**Rotate Accumulator Right**

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

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

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

**RRC     A**

**Rotate Accumulator Right through the Carry flag**

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

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

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

**SWAP    A**

**Swap nibbles within the Accumulator**

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

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

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

# Appendix A : instruction set (9/18)

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

## Move byte variable

MOV	A, Rn
Operation :	(A) ← (Rn)
MOV	A, @Ri
Operation :	(A) ← ((Ri))
MOV	A, direct
Operation :	(A) ← (direct)
MOV	A, #date
Operation :	(A) ← data
MOV	Rn, A
Operation :	(Rn) ← (A)
MOV	Rn, direct
Operation :	(Rn) ← (direct)
MOV	Rn, #date
Operation :	(Rn) ← data
MOV	direct, A
Operation :	(direct) ← (A)
MOV	direct, Rn
Operation :	(direct) ← (Rn)

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

1 1 1 0 1 r r r

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

1 1 1 0 0 1 1 i

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

1 1 1 0 0 1 0 1 direct addr

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

0 1 1 1 0 1 0 0 immediate data

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

1 1 1 1 1 r r r

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

1 0 1 0 1 r r r direct addr

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

0 1 1 1 1 r r r immediate data

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

1 1 1 1 0 1 0 1 direct addr

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

1 0 0 0 1 r r r direct addr

# Appendix A : instruction set (10/18)

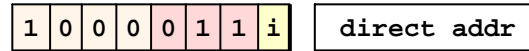
MOV	direct, @Ri
Operation :	(direct) ← ((Ri))
MOV	direct, direct
Operation :	(direct) ← (direct)
MOV	direct, #data
Operation :	(direct) ← data
MOV	@Ri, A
Operation :	((Ri)) ← (A)
MOV	@Ri, direct
Operation :	((Ri)) ← (direct)
MOV	@Ri, #data
Operation :	((Ri)) ← data

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

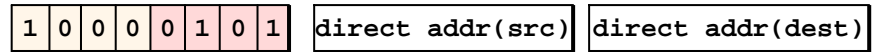
### Move bit data

MOV	C, bit
Operation :	(C) ← (bit)
MOV	bit, C
Operation :	(bit) ← (C)

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



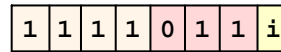
Encoding :      HEX: 85h, #bytes: 3, Cycles: 3



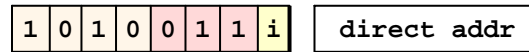
Encoding :      HEX: 75h, #bytes: 3, Cycles: 3



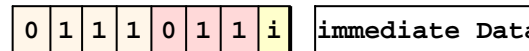
Encoding :      HEX: F6h, #bytes: 1, Cycles: 1



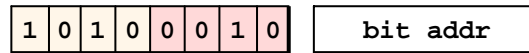
Encoding :      HEX: A6h, #bytes: 2, Cycles: 2



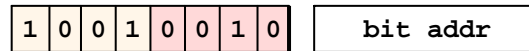
Encoding :      HEX: 76h, #bytes: 2, Cycles: 2



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



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



# Appendix A : instruction set (11/18)

**MOV DPTR, #data16**

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

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

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

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)  $\leftrightarrow$  (Rn)

XCH A, @Ri

Operation : (A)  $\leftrightarrow$  ((Ri))

XCH A, direct

Operation : (A)  $\leftrightarrow$  (direct)

**XCHD A, @Ri**

Exchange Digit

Operation : (A<sub>3-0</sub>)  $\leftrightarrow$  ((Ri))<sub>3-0</sub>

**PUSH direct**

Push onto stack

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

**POP direct**

Pop onto stack

Operation : (direct)  $\leftarrow$  ((SP))  
(SP)  $\leftarrow$  (SP) - 1

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

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

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

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

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

direct addr

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)  $\leftarrow$  1

**SETB bit**

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

**JC rel**

**Jump if Carry is set**

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

**JNC rel**

**Jump if Carry is not set**

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

**JB bit, rel**

**Jump if Bit is set**

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

**JNB bit, rel**

**Jump if Bit is not set**

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

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

1 1 0 1 0 0 1 1

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

1 1 0 1 0 0 1 0

bit addr

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

0 1 0 0 0 0 0 0

relative addr

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

0 1 0 1 0 0 0 0

relative addr

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

0 0 1 0 0 0 0 0

bit addr

relative addr

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

0 0 1 1 0 0 0 0

bit addr

relative addr

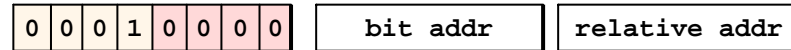
# Appendix A : instruction set (14/18)

## 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

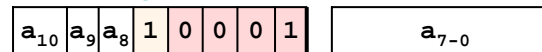


## ACALL addr11

**Absolute Subroutine Call**

**Operation :**  $(PC) \leftarrow (PC) + 2$   
 $(SP) \leftarrow (SP) + 1$   
 $((SP)) \leftarrow (PC_{7-0})$   
 $(SP) \leftarrow (SP) + 1$   
 $((SP)) \leftarrow (PC_{15-8})$   
 $(PC_{10-0}) \leftarrow$  page address

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

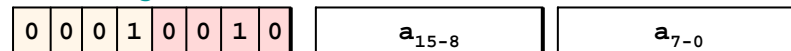


## LCALL addr16

**Long Subroutine Call**

**Operation :**  $(PC) \leftarrow (PC) + 3$   
 $(SP) \leftarrow (SP) + 1$   
 $((SP)) \leftarrow (PC_{7-0})$   
 $(SP) \leftarrow (SP) + 1$   
 $((SP)) \leftarrow (PC_{15-8})$   
 $(PC) \leftarrow addr_{15-0}$

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

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

## Appendix A : instruction set (16/18)

**JMP @A + DPTR**

**Jump Indirect Relative to the DPTR**

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

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

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

**JZ rel**

**Jump if Accumulator is Zero**

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

**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 :**  $(PC) \leftarrow (PC) + 2$   
If (A)≠0, then  $(PC) \leftarrow (PC) + rel$

**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**

(PC)  $\leftarrow$  (PC) + 3  
 If (A)  $\neq$  (direct),  
**Operation :** then (PC)  $\leftarrow$  (PC) + rel  
 If (A) < (direct), then (C)  $\leftarrow$  1  
 Else (C)  $\leftarrow$  0

**CJNE A, #data, rel**

(PC)  $\leftarrow$  (PC) + 3  
 If (A)  $\neq$  data,  
**Operation :** then (PC)  $\leftarrow$  (PC) + rel  
 If (A) < data, then (C)  $\leftarrow$  1  
 Else (C)  $\leftarrow$  0

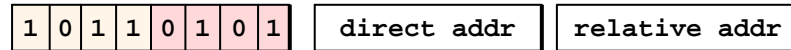
**CJNE Rn, #data, rel**

(PC)  $\leftarrow$  (PC) + 3  
 If (Rn)  $\neq$  data,  
**Operation :** then (PC)  $\leftarrow$  (PC) + rel  
 If (Rn) < data, then (C)  $\leftarrow$  1  
 Else (C)  $\leftarrow$  0

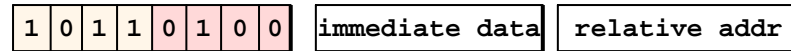
**CJNE @Ri, #data, rel**

(PC)  $\leftarrow$  (PC) + 3  
 If ((Ri))  $\neq$  data,  
**Operation :** then (PC)  $\leftarrow$  (PC) + rel  
 If ((Ri)) < data, then (C)  $\leftarrow$  1  
 Else (C)  $\leftarrow$  0

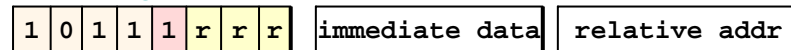
**Encoding :** HEX: B5h, #bytes: 3, Cycles: 4



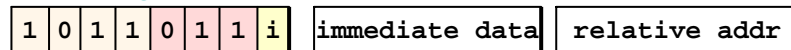
**Encoding :** HEX: B4h, #bytes: 3, Cycles: 4



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



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



## Appendix A : instruction set (18/18)

DJNZ <byte>, rel

Decrement and Jump if Not Zero

DJNZ Rn, rel

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

DJNZ direct, rel

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

NOP

No Operation

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

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

1 1 0 1 1 r r r relative addr

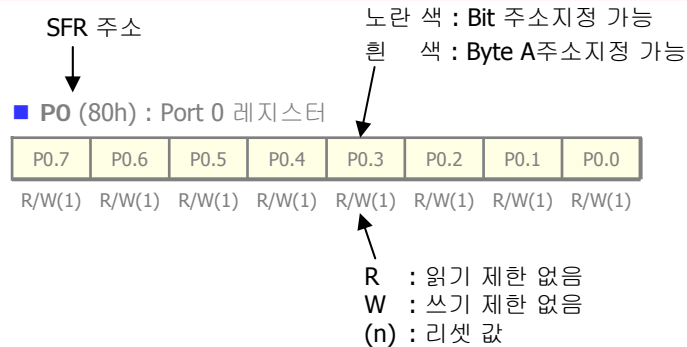
Encoding : HEX: D5h, #bytes: 3, Cycles: 4

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

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

0 0 0 0 0 0 0 0

### [설명을 읽는 방법]



### ■ PO (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)

- ◆ Open-drain 양방향 단자.
- ◆ 외부 메모리를 접근하는 동안, 하위 바이트 주소/데이터 버스의 다중화.

### ■ SP (81h) : Stack Pointer 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 시작할 곳을 지시한다.
- ◆ PUSH에 의하여 증가하고 POP에 의하여 감소한다.

### ■ DPL (82h) : Data Pointer Low Byte 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) : Data Pointer High Byte 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	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 : UART 모드 1, 2, 3에서 타이머 1 보레이트 두 배.
- ◆ SMOD0 : SM0 접근 허용. 이 비트를 수정하지 말라.
- ◆ POF : 파워 오프 표시.  
파워가 켜지면, 이 비트는 H/W에 의하여 설정될 것이다.
- ◆ GF1, GF0: 범용의 표시 비트.
- ◆ PD : 파워 다운 (정지) 모드 비트.
- ◆ IDL : 아이들 모드 비트.



■ **TCON (88h)** : 타이머/카운터 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 : 타이머 1 overflow 표시.
- ◆ TR1 : 타이머 1 동작 제어.
- ◆ TF0 : 타이머 0 overflow 표시.
- ◆ TR0 : 타이머 0 동작 제어.
- ◆ IE1 : 외부 인터럽트 1 표시.  
IT1 = 0이면, 소프트웨어에 의하여 지워진다.  
IT1 = 1이면, 인터럽트가 처리될 때 자동으로 지워진다.
- ◆ IT1 : 외부 인터럽트 1 타입 제어.  
Edge 검출 (IT1=1) / 레벨 검출 (IT1=0; 기본 지정)
- ◆ IE0 : 외부 인터럽트 0 표시.  
IT0 = 0이면, 소프트웨어에 의하여 지워진다.  
IT0 = 1이면, 인터럽트가 처리될 때 자동으로 지워진다.
- ◆ IT0 : 외부 인터럽트 0 타입 제어.  
Edge 검출 (IT0=1) / 레벨 검출 (IT0=0; 기본 지정)

■ **TMOD (89h)** : 타이머/카운터 0 모드 제어 레지스터

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)

- ◆ GATE[7] : 타이머/카운터 1 기능 제어 비트.  
'1'이면 INT1 핀으로 외부에서 제어, '0'이면 내부에서 제어.
- ◆ C/T[6] : 타이머/카운터 1의 입력 클럭 결정. '1'이면 T1 핀 입력 계수.
- ◆ M1, M0 [5:4] : 타이머/카운터 1의 모드 설정 0.  
[0,0] : 모드 0, 13-bit T/C  
[0,1] : 모드 1, 16-bit T/C  
[1,0] : 모드 2, 8-bit T/C, 자동 재입력  
[1,1] : 사용 안 함

◆ GATE[3] : 타이머/카운터 0 제어 비트.

'1'이면 INT1 핀으로 외부에서 제어, '0'이면 내부에서 제어.

◆ C/T[2] : 타이머/카운터 0의 입력 클럭 결정. '1'이면 T0 핀 입력 계수.

◆ M1, M0 : 타이머/카운터 0의 모드 설정 0.

[0,0] : 모드 0, 13-bit T/C

[0,1] : 모드 1, 16-bit T/C

[1,0] : 모드 2, 8-bit T/C, 자동 재입력

[1,1] : 모드 3, 두개의 8-bit T/C

■ **TLO (8Ah)** : 타이머/카운터 0의 Low 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)** : 타이머/카운터 1의 Low 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)** : 타이머/카운터 0의 High 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)** : 타이머/카운터 1의 High 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)

## Appendix B : SFR Description [8Eh ~ 99h] (3/9)

### ■ CKCON (8Eh) : 클럭 제어 레지스터

WD1	WD0	T2M	T1M	T0M	-	-	-
-----	-----	-----	-----	-----	---	---	---

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

- ◆ WD1, WD0 : Watchdog 타이머 모드 선택
  - [0,0] :  $2^{17}$  clocks (interrupt),  $2^{17} + 512$  clocks (reset)
  - [0,1] :  $2^{20}$  clocks (interrupt),  $2^{20} + 512$  clocks (reset)
  - [1,0] :  $2^{23}$  clocks (interrupt),  $2^{23} + 512$  clocks (reset)
  - [1,1] :  $2^{26}$  clocks (interrupt),  $2^{26} + 512$  clocks (reset)
- ◆ T2M : T/C 2의 기본 시간 선택. '1'이면, time base가 4 클럭.
- ◆ T1M : T/C 1의 기본 시간 선택. '1'이면, time base가 4 클럭.
- ◆ T0M : T/C 0의 기본 시간 선택. '1'이면, time base가 4 클럭.

### ■ P1 (90h) : Port 1 레지스터

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)

- ◆ 내부 pull-up 저항을 갖는 준 양방향 단자.
- ◆ 단자를 다른 선택 기능을 위하여 사용하려면, P1.X이 "1"이어야 한다.

### ■ EXIF (91h) : 외부 인터럽트 플래그 레지스터

IE5	IE4	IE3	IE2	XT	-	-	BGS
-----	-----	-----	-----	----	---	---	-----

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

- ◆ IE5, IE4, IE3, IE2 : 외부 인터럽트 5, 4, 3, 2 표시.  
소프트웨어에 의해 지워짐.
- ◆ XT : Crystal 선택. (Read only)  
설정되면, Crystal을 시스템 클럭으로 사용한다.
- ◆ BGS : Band-gap select. 설정되면, LVD는 파워 다운 모드에서 동작할 것이다.

### ■ SCON (98h) : UART0의 직렬 포트 제어 레지스터

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 : 시리얼 단자 모드 선택.
  - [0,0] : 모드 0, 8-bit shift register ( $F_{osc}/4$ )
  - [0,1] : 모드 1, 8-bit UART (Variable)
  - [1,0] : 모드 2, 9-bit UART ( $F_{osc}/32$  or  $F_{osc}/16$ )
  - [1,1] : 모드 3, 9-bit UART (Variable)
- ◆ SM2 : 모드 2, 3에서 자동 주소 인식을 허용한다.  
모드 1에서, SM2가 "1"이면 stop bit의 유효성 검사.  
모드 0에서, SM2은 "0"이어야 한다.
- ◆ REN : 직렬 수신 허용.
- ◆ TB8 : 모드 2, 3에서 전송되는 9번째 데이터 비트.
- ◆ RB8 : 모드 2, 3에서 수신되는 9번째 데이터 비트.  
모드 1에서, SM2 is "0"이면 RB8는 stop bit와 같다.  
모드 0에서, RB8은 사용되지 않는다.
- ◆ TI : 전송 인터럽트 표시. 소프트웨어에 의해 지워진다.
- ◆ RI : 수신 인터럽트 표시. 소프트웨어에 의해 지워진다.

### ■ 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)

- ◆ 송신과 수신 버퍼는 분리되어 있다.
- ◆ 읽기와 쓰기의 주소는 같다.

## ■ P2 (A0h) : Port 2 레지스터

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)

- ◆ 내부 pull-up 저항을 갖는 유사 양방향 포트.
- ◆ 외부 메모리를 접근할 때 어드레스 출력과 일반 I/O.

## ■ P4 (A5h) : Port 4 Register

-	-	-	-	P4.3	P4.2	P4.1	P4.0
---	---	---	---	------	------	------	------

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

- ◆ Quasi bi-directional port with internal pull-up resistors.

## ■ P4SEL (A6h) : Port 4 Pull-up 제어 레지스터

-	-	-	-	P4SEL.3	P4SEL.2	P4SEL.1	P4SEL.0
---	---	---	---	---------	---------	---------	---------

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

- ◆ 0 = 내부 Pull-up 저항이 ON (기본 설정) / 1 = OFF

## ■ IE (A8h) : 인터럽트 허용 레지스터

EA	EADC	ET2	ES	ET1	EX1	ET0	EX0
----	------	-----	----	-----	-----	-----	-----

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

- ◆ EA : 전체 인터럽트 허용.
- ◆ EADC : ADC 인터럽트 허용.
- ◆ ET2 : 타이머 2 인터럽트 허용.
- ◆ ES : 직렬 단자 인터럽트 허용.
- ◆ ET1 : 타이머 1 인터럽트 허용.
- ◆ EX1 : 외부 인터럽트 1 허용.
- ◆ ET0 : 타이머 0 인터럽트 허용.
- ◆ EX0 : 외부 인터럽트 0 허용.

## ■ SADDR (A9h) : Slave 어드레스 레지스터

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)

- ◆ 직렬포트에 할당된 주어지거나 전체에 전송되는 주소로 프로그램됨

## ■ P3 (B0h) : Port 3 레지스터

P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
------	------	------	------	------	------	------	------

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

- ◆ 내부 pull-up을 갖는 유사 양방향 단자.
- ◆ 다른 선택 기능을 사용하려면, P3.X가 "1"이어야 한다.

## ■ IPH (B7h) : 인터럽트 우선순위 상위 비트 레지스터

-	PADCH	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) R/W(0)

- ◆ PADCH : ADC 인터럽트 우선순위 상위 비트.
- ◆ PT2H : 타이머 2 인터럽트 우선순위 상위 비트.
- ◆ PSH : 직렬 포트 (UART) 인터럽트 우선순위 상위 비트.
- ◆ PT1H : 타이머 1 인터럽트 우선순위 상위 비트.
- ◆ PX1CH : 외부 인터럽트 1 우선순위 상위 비트.
- ◆ PT0H : 타이머 0 인터럽트 우선순위 상위 비트.
- ◆ PX0H : 외부 인터럽트 0 우선순위 상위 비트.

# Appendix B : SFR Description [B8h ~ CBh] (5/9)

## ■ IP (B8h) : 인터럽트 우선순위 하위 비트 레지스터

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

- ◆ PADC : ADC 인터럽트 우선순위 하위 비트.
- ◆ PT2 : 타이머 2 인터럽트 우선순위 하위 비트.
- ◆ PS : 직렬 포트 (UART) 인터럽트 우선순위 하위 비트.
- ◆ PT1 : 타이머 1 인터럽트 우선순위 하위 비트.
- ◆ PX1 : 외부 인터럽트 1 우선순위 하위 비트.
- ◆ PT0 : 타이머 0 인터럽트 우선순위 하위 비트.
- ◆ PX0 : 외부 인터럽트 0 우선순위 하위 비트.

## ■ SADEN (B9h) : Slave 주소 마스크 레지스터

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)

## ■ PMR (C4h) : 전력 조절 제어 레지스터

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

- ◆ ALEOFF : 1 = ALE 토글링 금지.  
0 = ALE 토글링 허용 (기본지정).

## ■ STATUS (C5h) : Crystal 상태 레지스터

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

- ◆ XTUP : Crystal oscillator warm-up status.  
Cleared by H/W when Power-on reset.  
Cleared by H/W during Power-down wake-up.  
Set by H/W after XTAL stabilization.

## ■ T2CON (C8h) : 타이머/카운터 2 제어 레지스터

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 : 타이머 2 overflow 표시.
- ◆ EXF2 : 타이머 2 외부 표시.
- ◆ RCLK : 클럭 수신 표시.
- ◆ TCLK : 클럭 전송 표시.
- ◆ EXEN2 : 타이머 2 제어 비트.  
'1'이면 T2EX의 하강 에지에서 재입력이나 데이터 수집 발생,  
'0'이면 T2EX 신호 무시.
- ◆ TR2 : 타이머 2의 시작/정지 제어.
- ◆ C/T2 : 타이머 2 타이머/카운터 기능 선택. '0'이면 타이머, '1' 카운터.
- ◆ CP/RL2 : 데이터 수집/재입력 선택.  
CP/RL2 = 0, 재입력. (TH2,TL2) ← (RCAP2H,RCAP2L)  
CP/RL2 = 1, 수집. (RCAP2H,RCAP2L) ← (TH2,TL2)

## ■ T2MOD (C9h) : 타이머/카운터 2 모드 제어 레지스터

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

- ◆ T2OE : 타이머 2 출력 허용. '1'이면, P1.0으로 출력됨.
- ◆ DCEN : 타이머 2 up/down 제어. '1'이면, 카운트 다운.

## ■ RCAP2L (CAh) : 타이머/카운터 2 수집/재입력 하위 바이트 레지스터

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) : 타이머/카운터 2 수집/재입력 상위 바이트 레지스터

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) : 타이머/카운터 2 하위 바이트 레지스터

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) : 타이머/카운터 2 상위 바이트 레지스터

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)

### ■ PSW (D0h) : 프로그램 상태 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 표시.
- ◆ AC : 보조 캐리 표시.
- ◆ F0 : 사용자 표시 0.
- ◆ RS1, RS0 : 레지스터 뱅크 선택
  - [0,0] : 뱅크 0
  - [0,1] : 뱅크 1
  - [1,0] : 뱅크 2
  - [1,1] : 뱅크 3
- ◆ OV : Overflow 표시, 대수 연산에 의하여 설정됨.
- ◆ F1 : 사용자 표시 1.
- ◆ P : 패리티 비트. ACC 홀수 패리티에 따라 H/W에 의해 set/clear.

### ■ WDCON (D8h) : Watchdog 타이머 & 파워 상태 레지스터

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

- ◆ POR : 파워 온 리셋 표시.
- ◆ EPFI : 파워 오류 인터럽트 가능.
- ◆ PFI : 파워 오류 인터럽트 표시.
- ◆ WDIF : Watchdog 타이머 인터럽트 표시.
- ◆ WTRF : Watchdog 타이머 리셋 표시.
- ◆ EWT : Watchdog 타이머 리셋 허용.
- ◆ RWT : Watchdog 타이머 재출발.

### ■ PWMOCON (DCh) : PWM 제어 레지스터

POSEL	PS2_P0	PS1_P0	PS0_P0	MODE_P0	RL_P0	CLR_P0	RUN_P0
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- ◆ POSEL : PWM 파형을 P3.4로 출력 허용.
- ◆ PS2\_P0, PS1\_P0, PS0\_P0 : 선분주 클럭 선택.
  - [0,0,0] = FOSC/1, [0,0,1] = FOSC/2, [0,1,0] = FOSC/4,
  - [0,1,1] = FOSC/8, [1,0,0] = FOSC/16, [1,0,1] = FOSC/32,
  - [1,1,0] = FOSC/64, [1,1,1] = FOSC/128
- ◆ MODE\_P0 : 8 bits / (2+6) bits 카운터 모드 선택.
  - 0 = (2+6) bits 모드 (기본 지정)
  - 1 = 8 bits 모드
- ◆ RL\_P0 : Duty 데이터 재입력 선택
  - 0 = 6-bit 카운터 overflow 후 재입력 (기본 지정)
  - 1 = 8-bit 카운터 overflow 후 재입력
- ◆ CLR\_P0 : 카운터 리셋 가능. 하드웨어에 의하여 소거.
- ◆ RUN\_P0 : 카운터 시작/정지 제어.

## Appendix B : SFR Description [DDh ~ E3h] (7/9)

### ■ PWM1CON (DDh) : PWM Control Register

P1SEL	PS2_P1	PS1_P1	PS0_P1	MODE_P1	RL_P1	CLR_P1	RUN_P1
R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- ◆ P1SEL : PWM 파형을 P1.0로 출력 가능.
- ◆ PS2\_P1, PS1\_P1, PS0\_P1 : 선분주 클럭 선택.  
 [0,0,0] = FOSC/1, [0,0,1] = FOSC/2, [0,1,0] = FOSC/4,  
 [0,1,1] = FOSC/8, [1,0,0] = FOSC/16, [1,0,1] = FOSC/32,  
 [1,1,0] = FOSC/64, [1,1,1] = FOSC/128
- ◆ MODE\_P1 : 8 bits / (2+6) bits 카운터 모드 선택.  
 0 = (2+6) bits 모드 (기본지정)  
 1 = 8 bits 모드
- ◆ RL\_P1 : Duty 데이터 재입력 선택  
 0 = 6-bit 카운터 overflow 후 재입력 (기본 지정)  
 1 = 8-bit 카운터 overflow 후 재입력
- ◆ CLR\_P1 : 카운터 리셋 가능. 하드웨어에 의하여 소거.
- ◆ RUN\_P1 : 카운터 시작/정지 제어.

### ■ PWM0D (DEh) : PWM 0 펄스 폭 데이터 레지스터

PWM0D.7	PWM0D.6	PWM0D.5	PWM0D.4	PWM0D.3	PWM0D.2	PWM0D.1	PWM0D.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)

### ■ PWM1D (DFh) : PWM 1 펄스 폭 데이터 레지스터

PWM1D.7	PWM1D.6	PWM1D.5	PWM1D.4	PWM1D.3	PWM1D.2	PWM1D.1	PWM1D.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) : 어큐물레이터

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)

### ■ ADCSEL (E2h) : ADC 클럭과 단자 제어 레지스터

ADIV2	ADIV1	ADIV0	-	ADC3	ADCS2	ADCS1	ADCS0
R/W(0)	R/W(0)	R/W(0)		R/W(0)	R/W(0)	R/W(0)	R/W(0)

- ◆ ADIV2, ADIV1, ADIV0 : ADC 입력 클럭 분주.  
 [0,0,0] : 1-분주 ( $F_{osc}$ )  
 [0,0,1] : 2-분주 ( $F_{osc}/2$ )  
 [0,1,0] : 4-분주 ( $F_{osc}/4$ )  
 [0,1,1] : 8-분주 ( $F_{osc}/8$ )  
 [1,0,0] : 16-분주 ( $F_{osc}/16$ )
- ◆ ADC3 : 1 = P1.3으로 ADC3 입력 허용 & 디지털 입력 금지.
- ◆ ADC2 : 1 = P1.2으로 ADC2 입력 허용 & 디지털 입력 금지.
- ◆ ADC1 : 1 = P1.1으로 ADC1 입력 허용 & 디지털 입력 금지.
- ◆ ADC0 : 1 = P0.0으로 ADC0 입력 허용 & 디지털 입력 금지.

### ■ ALTSEL (E3h) : 다른 선택 기능 제어 레지스터

-	-	EAINTO	POINT1	POINT0	TX	RX	PWM1
		R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

- ◆ 오직 I 8/10/20-pin SPDIP/SOIC Package에서 사용가능.  
 40-PDIP, 44-PLCC/LQFP, 28-SPDP/SOIC Package에서 사용하지 말라.
- ◆ EAINTO : 1 = INTO 입력을 위하여 EA 핀 사용.
- ◆ POINT1 : 1 = INT1 입력을 위하여 P0.2 핀 사용.
- ◆ POINT0 : 1 = INTO 입력을 위하여 P0.1 핀 사용.
- ◆ TX : 1 = TX 입출력을 위하여 P0.1 핀 사용.
- ◆ RX : 1 = RX 출력을 위하여 P0.0 핀 사용.
- ◆ PWM1 : 1 = PWM1 출력을 위하여 P0.1 핀 사용.

## Appendix B : SFR Description [E4h ~ EFh] (8/9)

### ■ 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(1) R/W(1) R/W(1) R/W(1) R/W(1) R/W(1) R/W(1) R/W(1)

◆ 0 = 내부 Pull-up 저항이 ON / 1= OFF (기본 지정)

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

P1SEL.7	P1SEL.6	P1SEL.5	P1SEL.4	P1SEL.3	P1SEL.2	P1SEL.1	P1SEL.0
---------	---------	---------	---------	---------	---------	---------	---------

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

◆ 0 = 내부 Pull-up 저항이 ON (기본 지정) / 1= OFF

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

P2SEL.7	P2SEL.6	P2SEL.5	P2SEL.4	P2SEL.3	P2SEL.2	P2SEL.1	P2SEL.0
---------	---------	---------	---------	---------	---------	---------	---------

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

◆ 0 = 내부 Pull-up 저항이 ON (기본 지정) / 1= OFF

### ■ P3SEL (E6h) : Port 3 Pull-up 제어 레지스터

P3SEL.7	P3SEL.6	P3SEL.5	P3SEL.4	P3SEL.3	P3SEL.2	P3SEL.1	P3SEL.0
---------	---------	---------	---------	---------	---------	---------	---------

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

◆ 0 = 내부 Pull-up 저항이 ON (기본 지정) / 1= OFF

### ■ EIE (E8h) : 확장 인터럽트 인에이블 레지스터

-	-	-	EWDT	EX5	EX4	EX3	EX2
---	---	---	------	-----	-----	-----	-----

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

- ◆ EWDT : Watchdog 타이머 인터럽트 허용
- ◆ EX5 : 외부 인터럽트 5 허용.
- ◆ EX4 : 외부 인터럽트 4 허용.
- ◆ EX3 : 외부 인터럽트 3 허용.
- ◆ EX2 : 외부 인터럽트 2 허용.

### ■ ADCR (EEh) : ADC 결과 상위 데이터 레지스터 : Value[8:1]

SAR8	SAR7	SAR6	SAR5	SAR4	SAR3	SAR2	SAR1
------	------	------	------	------	------	------	------

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 LSB 레지스터 : Value[0]

AD_EN	AD_REQ	AD_END	ADCF	ACH1	ACH0	-	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 허용, 소프트웨어에 의해 설정.
- ◆ AD\_REQ : 현 채널에 AD 변환 요청.  
AD\_END가 1에서 0으로 떨어질 때 하드웨어에 의해 소거.
- ◆ AD\_END : ADC 현상태.  
0 = ADC가 작동중임.
- ◆ ADCF : ADC 인터럽트 표시. 소프트웨어에 의해서 소거되어야 함.
- ◆ ACH1, ACH0 : ADC 채널 선택  
[0,0] = ADC0 입력 선택 (P0.0)  
[0,1] = ADC1 입력 선택 (P1.1)  
[1,0] = ADC2 입력 선택 (P1.2)  
[1,1] = ADC3 입력 선택 (P1.3)
- ◆ SAR0 : ADC 결과 값의 LSB.

## Appendix B : SFR Description [F0h ~ F8h] (9/9)

### ■ 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)

### ■ EIP (F8h) : 확장된 인터럽트 우선순위 레지스터

-	-	-	PWDT	RX5	PX4	PX3	PX2
---	---	---	------	-----	-----	-----	-----

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

- ◆ PWDT : Watchdog 타이머 인터럽트 우선순위 비트.
- ◆ PX5 : 외부 인터럽트 5 우선순위 비트.
- ◆ PX4 : 외부 인터럽트 4 우선순위 비트.
- ◆ PX3 : 외부 인터럽트 3 우선순위 비트.
- ◆ PX2 : 외부 인터럽트 2 우선순위 비트.



## Appendix C : Update History

- ◆ V1.8
  - ✓ 44-LQFP → 44-MQFP
- ◆ V1.9
  - ✓ EA : This pin must not be floating.
  - ✓ Describe the constraint of power slope
- ◆ V1.9 korea
  - ✓ 한글판 Brief Manual