실험 3. LED에 생명을 부여하기

(실험 2. LED ON/OFF 해보기)

2019년 1학기
(월요일, 수요일)

담 당 : 이인수교수

실험 2. LED ON/OFF 해보기

I/O Port를 사용한 입출력

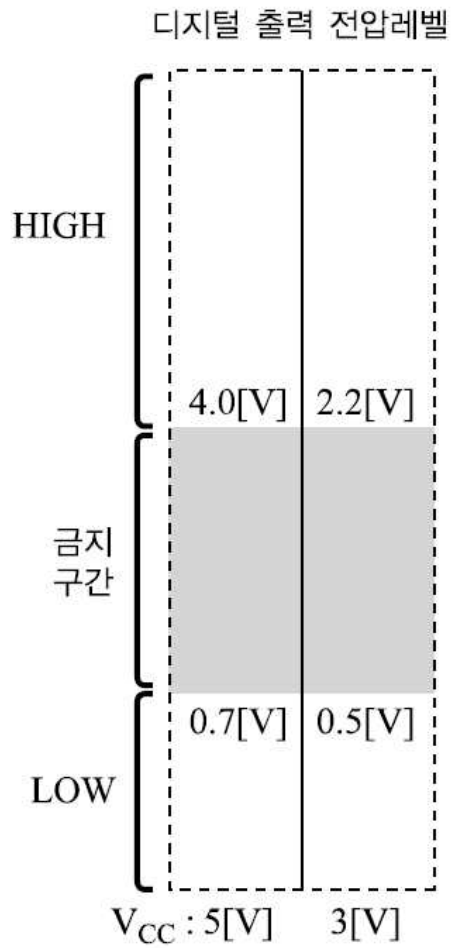
- PIC16F87x의 디지털 입출력 사용하기
- 입력 설정
  - TRISx Reg.의 해당 비트를 "1"로 설정
- 출력 설정
  - TRISx Reg.의 해당 비트를 "0"으로 설정

- 주의
  - PORTA의 경우 아날로그 입력과 디지털 입출력을 모두 지원함
  - 설정은 ADCON1 Reg. 참고 -→ ADCON1 레지스터 bit 0, 1, 2 설정, 00000111로 설정시에 디지탈
  - **RA4**의 기능에 유의
    - T0CKI : Counter로 사용
    - Open Drain 특성 이해

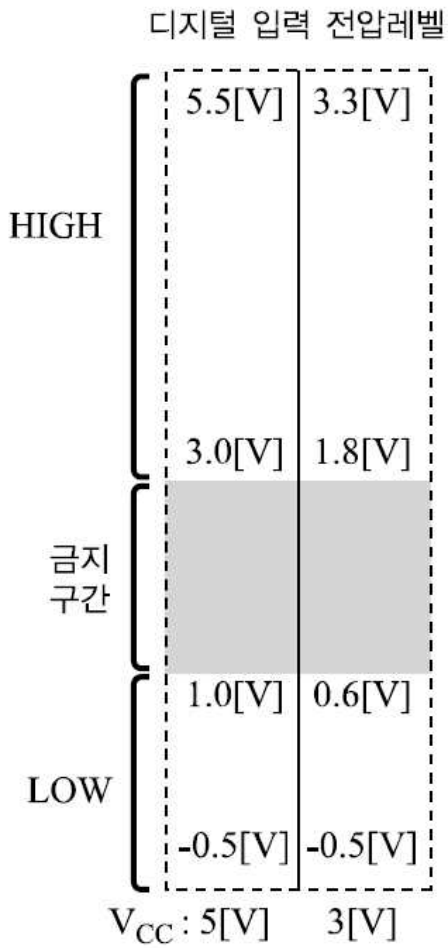
❖ 각 I/O Port의 특성은 I/O Port Description 부분 참고

PIC16F876A 핀을 이용한 논리값의 외부 입출력

➢ ATmega128의 논리값과 디지털 출력 및 입력전압



(a) 디지털 출력전압

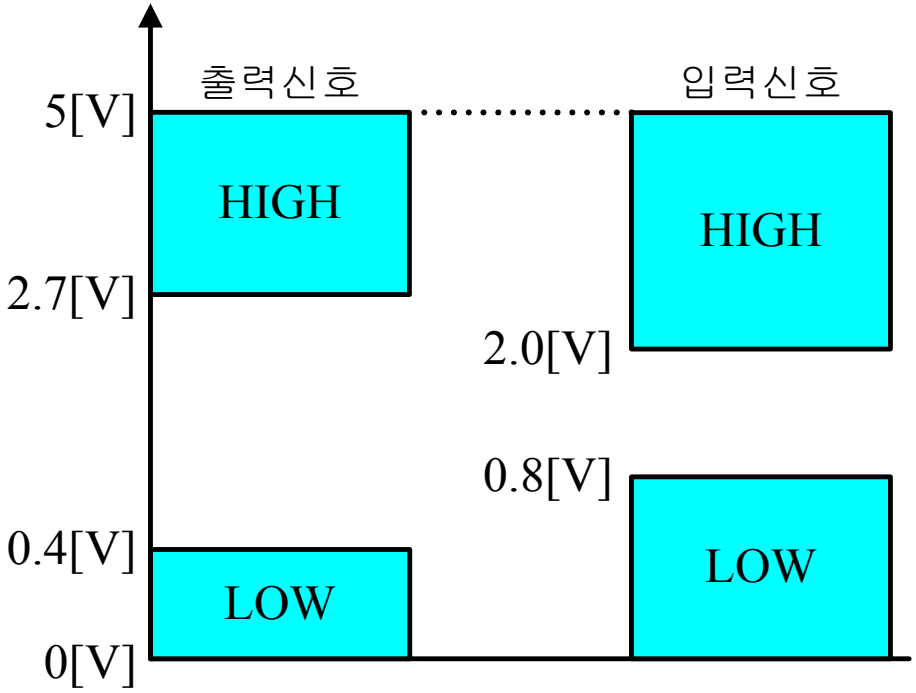


(b) 디지털 입력전압

Digital 정보의 표현

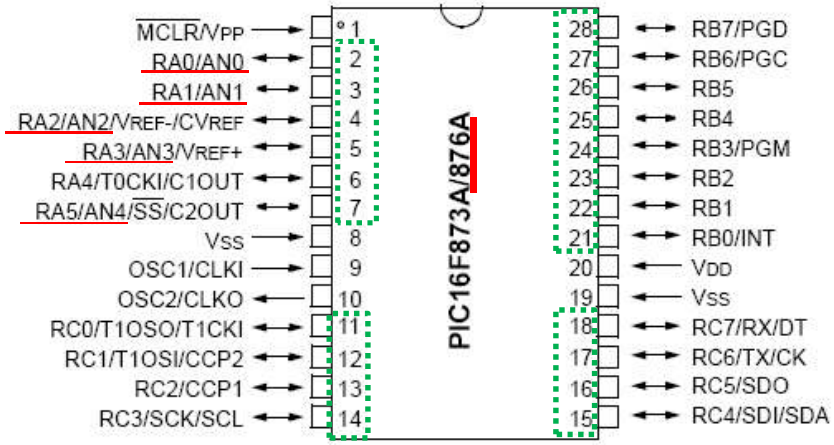
- 디지털 정보의 전압 범위
  - 디지털 정보를 표현하기 “0”과 “1”의 2진수 체계(binary system)사용

PIC16F876A



Architectural overview-General description

- General Description : 마이크로칩사(Microchip) PIC series
  : PIC가 상대적으로 저렴하여 소량생산이 가능, 내부 프로그램메모리를 플래시 타입으로 대체, 소수의 명령(35개)으로 사용가능



- Pin diagrams

Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I^2C			
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2

- Device features

I/O PORT DESCRIPTION

ADCON1 Register (9Fh)

Port A를 디지털 I/O로 사용하기 위해서는 반드시 ADCON1 reg. 를 설정해야 함.

```
BSF     STATUS,RP0
MOVLW   B'00000111'
MOVWF   ADCON1
BCF     STATUS,RP0
```

Here, x is don't care

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

bit 7 **ADFM:** A/D Result Format Select bit
1 = Right justified. 6 Most Significant bits of ADRESH are read as '0'.
0 = Left justified. 6 Least Significant bits of ADRESL are read as '0'.

bit 6-4 **Unimplemented:** Read as '0'

bit 3-0 **PCFG3:PCFG0**: A/D Port Configuration Control bits:

PCFG3: PCFG0	AN7(1) RE2	AN6(1) RE1	AN5(1) RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN/ Refs(2)
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

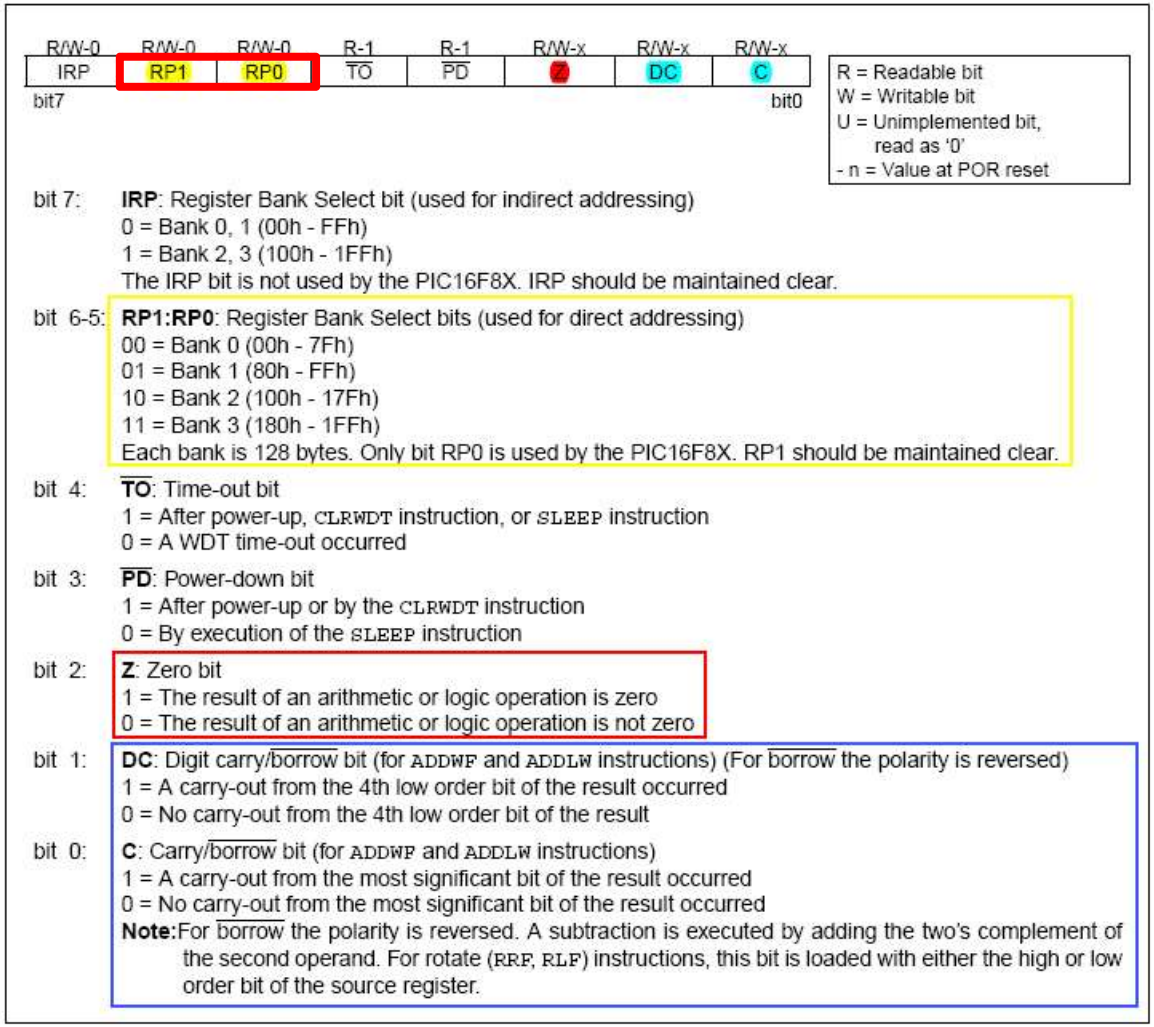
A = Analog input D = Digital I/O

Note 1: These channels are not available on PIC16F873/876 devices.
2: This column indicates the number of analog channels available as A/D inputs and the number of analog channels used as voltage reference inputs.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Register Overview-STATUS(1)

☐ Status Register(Address 03H, 83H)



R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as '0'
- n = Value at POR reset

bit 7: **IRP**: Register Bank Select bit (used for indirect addressing)
0 = Bank 0, 1 (00h - FFh)
1 = Bank 2, 3 (100h - 1FFh)
The IRP bit is not used by the PIC16F8X. IRP should be maintained clear.

bit 6-5: **RP1:RP0**: Register Bank Select bits (used for direct addressing)
00 = Bank 0 (00h - 7Fh)
01 = Bank 1 (80h - FFh)
10 = Bank 2 (100h - 17Fh)
11 = Bank 3 (180h - 1FFh)
Each bank is 128 bytes. Only bit RP0 is used by the PIC16F8X. RP1 should be maintained clear.

bit 4: **$\overline{TO}$**: Time-out bit
1 = After power-up, CLRWDT instruction, or SLEEP instruction
0 = A WDT time-out occurred

bit 3: **$\overline{PD}$**: Power-down bit
1 = After power-up or by the CLRWDT instruction
0 = By execution of the SLEEP instruction

bit 2: **Z**: Zero bit
1 = The result of an arithmetic or logic operation is zero
0 = The result of an arithmetic or logic operation is not zero

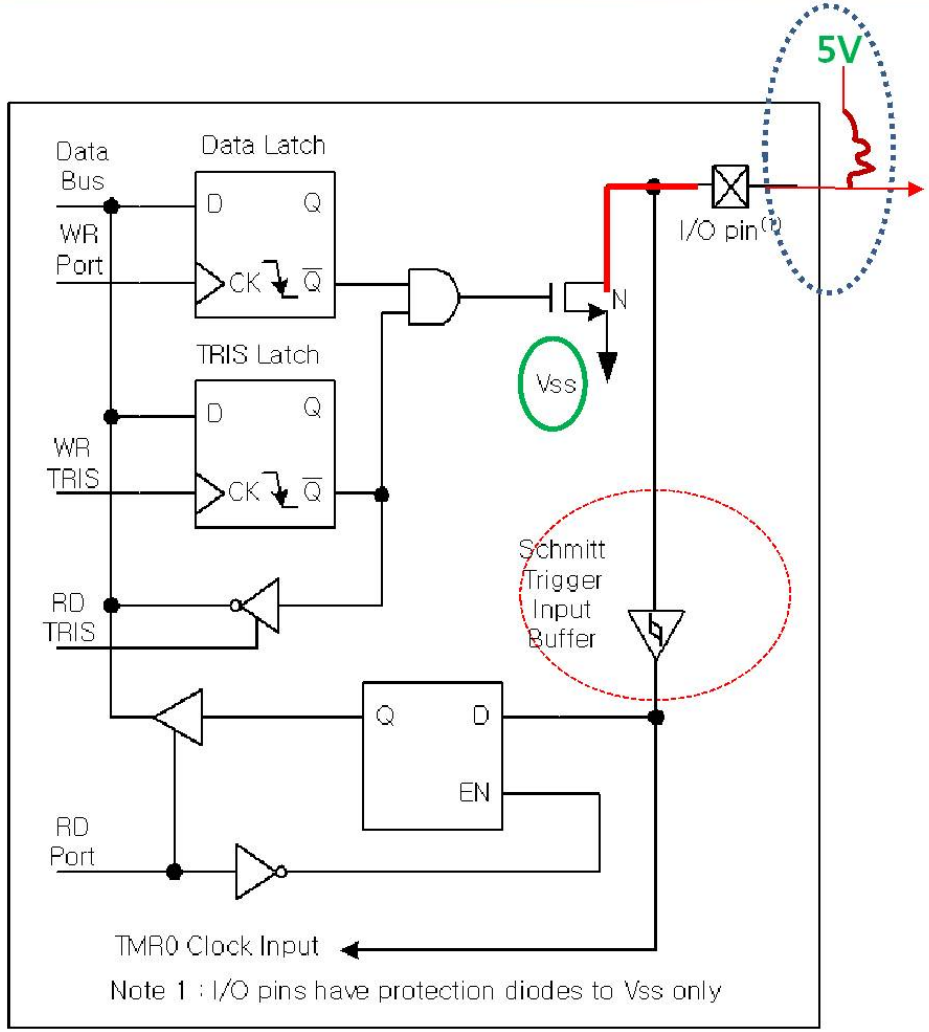
bit 1: **DC**: Digit carry/$\overline{borrow}$ bit (for ADDWF and ADDLW instructions) (For $\overline{borrow}$ the polarity is reversed)
1 = A carry-out from the 4th low order bit of the result occurred
0 = No carry-out from the 4th low order bit of the result

bit 0: **C**: Carry/$\overline{borrow}$ bit (for ADDWF and ADDLW instructions)
1 = A carry-out from the most significant bit of the result occurred
0 = No carry-out from the most significant bit of the result occurred
Note: For $\overline{borrow}$ the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high or low order bit of the source register.

- ✓ IRP, RP1 : not used PIC16F84
- ✓ RP0 : Bank select
  - •BCF STATUS,RP0 ; BANK0
  - •BSF STATUS,RP0; BANK1
- ✓ Z(F) : Zero flag
  - •BTFSC STATUS, Z(F)

I/O PORT DESCRIPTION

- [RA4]
  - TOCKI
    - **RTCC(Real Time Clock Counter)**
  - Open Drain 구조
    - 높은 전압 사용가능
    - **사용상 주의사항**
      - **출력 단자에 저항을 반드시 연결**
  - Schmitt Trigger Input
    - **Noise Margin 증가**
  - **Pull-up Resistor**
    - **2~6[Kohm] 정도의 낮은 저항 사용**



- MOS FET가 ON : 출력핀은 GND에 연결되므로 논리 "0"
  OFF : 출력핀은 Floating or high impedance상태

→ 강제로 논리값이 "1"이 되게 해주어야 함

→ 풀업저항의 연결로 MOS FET OFF시에 "1"을 출력가능
(우리 실험키트에는 RA4에 풀업저항 연결되어 있음)

→ 그러나 풀업저항연결시 Power On 시에는 RA4는 high 상태가 되어서 Buzzer On

I/O PORT DESCRIPTION

◘ [RA4]

POWER ON RESET과 관련되어서…

초기 TRISA상태는 "1"인가 "0"인가?

➔1"로 설정 PORTA는 입력으로 설정되어짐 그리고

PORTA의 상태는 HIGH인가 LOW인가?

➔X, U의 값을 가짐(X :UNKNOWN, U:UNCHANGED)

결국 초기에 POWER ON RESET상태에서 **RA4는 입력 포트로 설정 되어 있으므로 PULL UP 저항을 달아놓았다면 PIN 상태는 HIGH이며 BUZZ쪽에 연결되어 있는 TR이 ON이 되어 부저가 울리게 된다. (우리는 PULL UP저항있음)**

RA4와 BUZZ 연결하고 전원 ON하여 부저울림 확인!

- 어떻게 하면 이 문제를 해결 할 수 있을까?

① 트랜지스터의 베이스 쪽 저항 앞 즉 RA4 핀에 NOT 게이트를 달아 주면 초기에 BUZZ는 울리지 않을 것이다. 단, RA4번 동작은 반대로(BSF PORTA,4 일 경우 트랜지스터는 OFF되고, BCF PORTA,4의 경우 ON됨) 사용하면 된다.

② RA4를 사용하지 않으면 된다.

-→ 만일 RA4를 사용하려면 프로그램 로드후에 연결하면 됨 (프로그램 로드전에는 연결하지말것!)

소프트웨어적으로 해결 할 방법은 없다. 이것은 구조적 문제이기 때문이다.

PORT 와 TRIS 레지스터의 초기치

PORT 초기값

x = unknown, - = unimplemented

05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read	--0x 0000	43, 150
06h	PORTB	PORTB Data Latch when written: PORTB pins when read			xxxx xxxx	45, 150
07h	PORTC	PORTC Data Latch when written: PORTC pins when read			xxxx xxxx	47, 150

RA4는 입력 포트로 설정 되어 있으므로 PULL UP 저항을 달아놓았으며 PIN 상태는 "1"

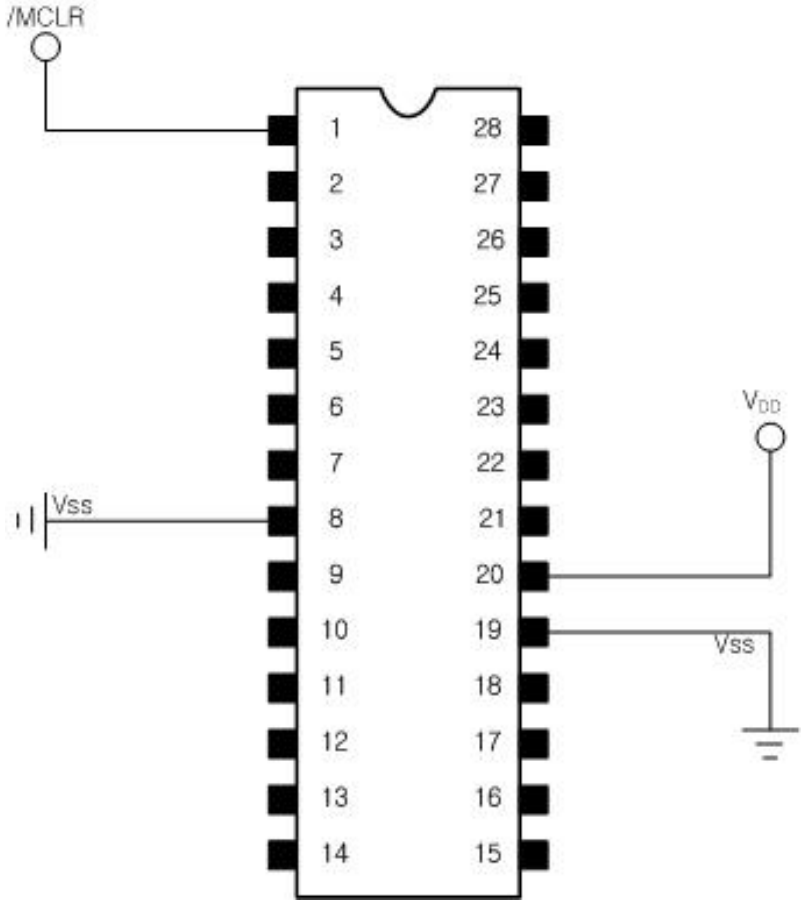
TRISA, B, C 초기값 ("1", 초기 입력으로 설정되어 있음)

85h	TRISA	—	—	PORTA Data Direction Register	--11 1111	43, 150
86h	TRISB	PORTB Data Direction Register			1111 1111	45, 150
87h	TRISC	PORTC Data Direction Register			1111 1111	47, 150

그러므로 TRISA, B, C 을 출력으로 사용시는 반드시 "0"으로 설정함

Lab.1 I/O Port를 사용한 입출력

- LED & Push 관련 회로도
  - PIC16F87x의 I/O pin test
  - Chip 선택 확인 :16F876A
  - 장비이상 유무 파악
  - Reg. 내용 확인
  - Cable 확인
  - ME Type 확인
  - GND, VDD 연결 확인
  - WDT 설정확인
  - 입력 Clock확인
  - 외부/내부 전원 확인
  - 보드확인
  - Port A, B, C 체크하기(입력출력)



Lab.1 I/O Port를 사용한 입출력

* 실험 1. 4개의 LED(L1,2,3,4)를 조건대로 Turn ON 시키기 (test1, Delay 없으면 Step into 기능으로 보기)
  - Port A : 출력 4개➔ LED(L1-L4)
  - PORTA에 연결된 LED가 0000, 0001, 0010, 0011, 0100, …,1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111 순서로 TURN ON
  - Debugger 동작(mode)과 Programmer 동작 (mode) 차이점 발견해 보자!!! (Power ON/OFF실험으로 구분)

* 실험 2. 4개의 스위치와 4개의 LED를 사용하여 각 스위치로 각 LED를 제어 (test5)
  - Port A : 출력 4개➔ LED(L1-L4)
    Port C : 입력 4개➔ TOGGLE S/W (S2, S3, S4, S5)
  - S/W OFF시에 해당 LED ON, S/W ON시에 해당 LED OFF -→ PULL UP 저항 연결때문 (즉, S/W OFF시에 1입력, ON시에 0입력됨)

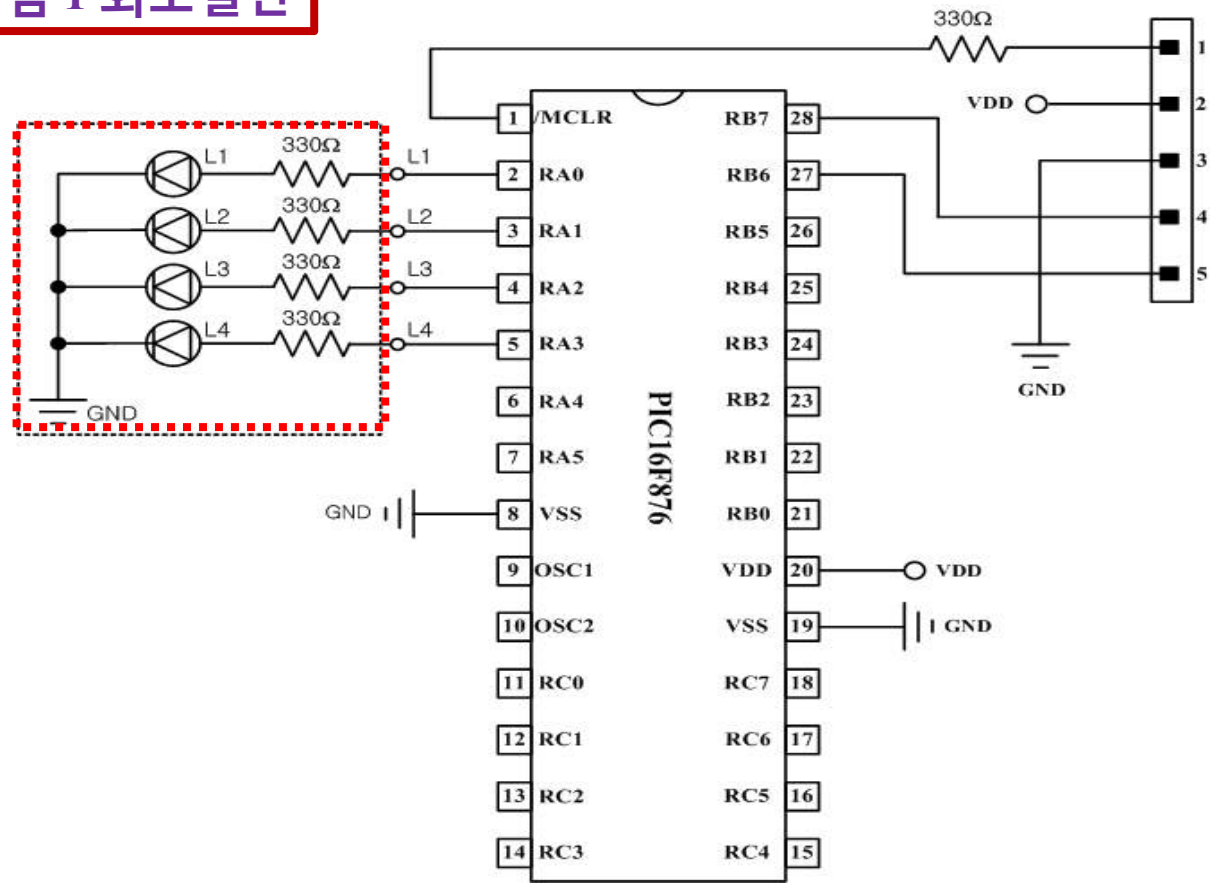
* 실험 3. 4개의 스위치와 4개의 LED를 사용하여 각 스위치로 각 LED를 제어(test4)
  - Port A : 출력 4개➔ LED(L1-L4)
    Port C : 입력 4개➔ TOGGLE S/W (S2, S3, S4, S5)
  - S/W ON시에 해당 LED ON, S/W OFF시에 해당 LED OFF 시키려면?

* 추가실험 : 실험 1에 지연시간(Delay time, 실험 3장내용) 추가해서 실험

I/O Port를 사용한 LED 구동(그림 1)

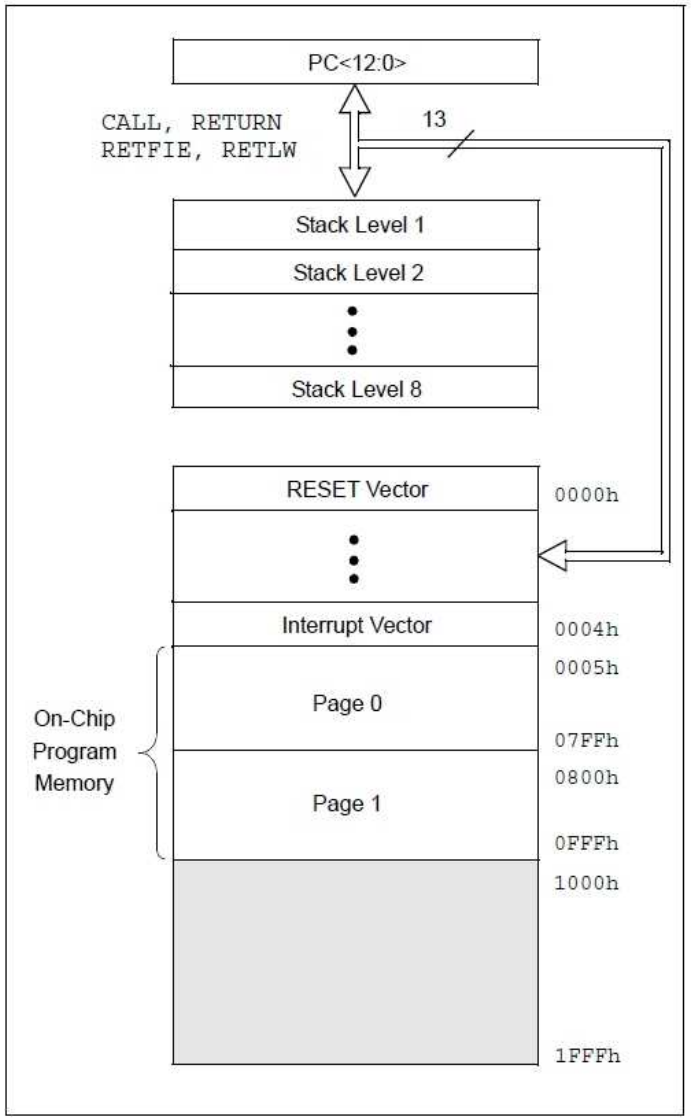
실험 1 회로결선

이 부분만 결선



Architectural overview-Memory & Stack

Program Memory & Stack & Data Memory



Data Memory Organization

RP1:RP0	Bank
00	0
01	1
10	2
11	3

bit 6-5: **RP1:RP0**: Register Bank Select bits (used for direct addressing)
00 = Bank 0 (00h - 7Fh)
01 = Bank 1 (80h - FFh)
10 = Bank 2 (100h - 17Fh)
11 = Bank 3 (180h - 1FFh)

(Status register)

0x000 ~ 0X7FF 번지 : 페이지 0
0X800 ~ 0XFFF 번지 : 페이지 1
0X1000 ~ 0X17FF 번지 : 페이지 2
0X1800 ~ 0X1FFF 번지 : 페이지 3

Register overview–FILE REGISTER

Register File Map

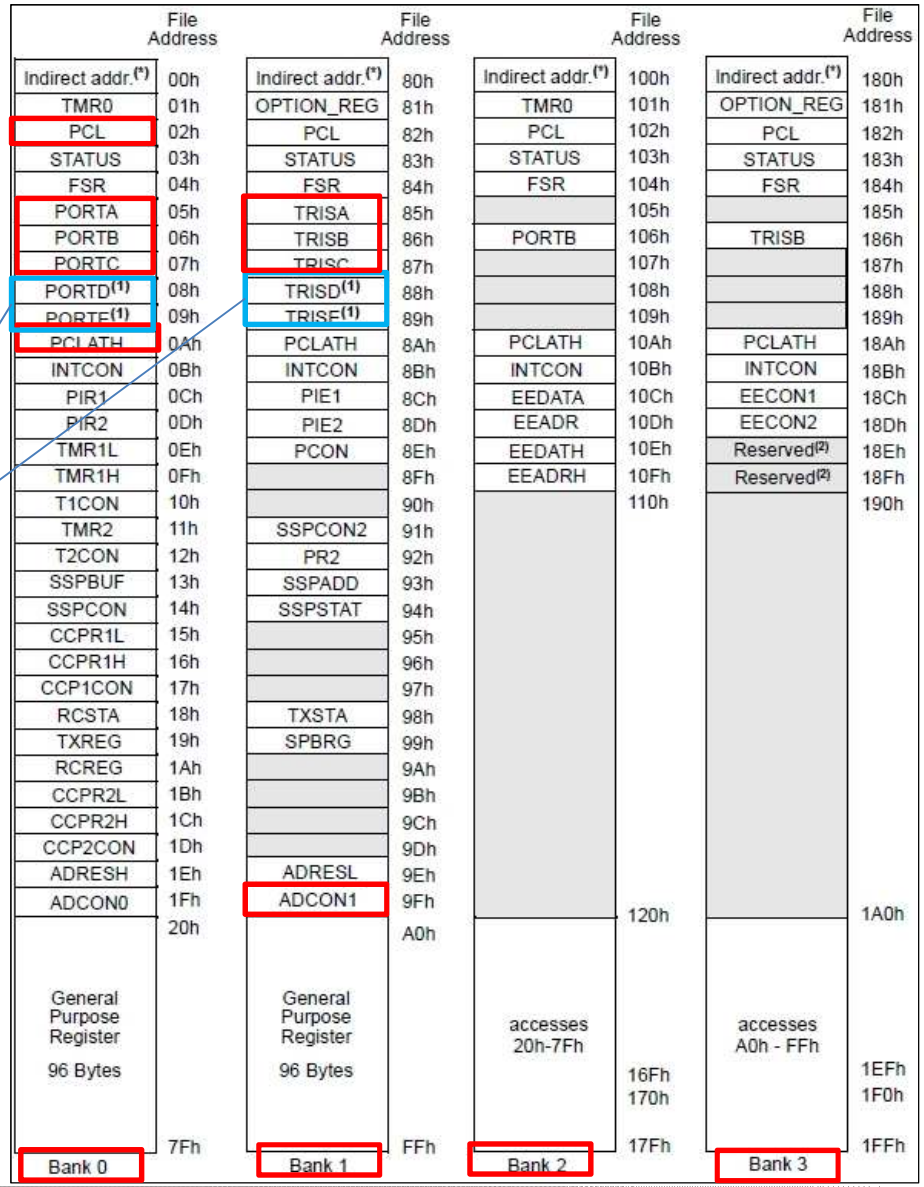
앞으로 사용가능

Unimplemented data memory locations, read as '0'.
* Not a physical register.
Note 1: These registers are not implemented on the PIC16F873.
2: These registers are reserved, maintain these registers clear.

Bank = SFR(특수기능 Reg.) + GPR (범용 Reg.)

각 Bank에는 128개의 레지스터존재
총 SPR(특수목적 Reg.) 72개와
GPR (범용 Reg.) 368개

PIC16F876A에는 없음



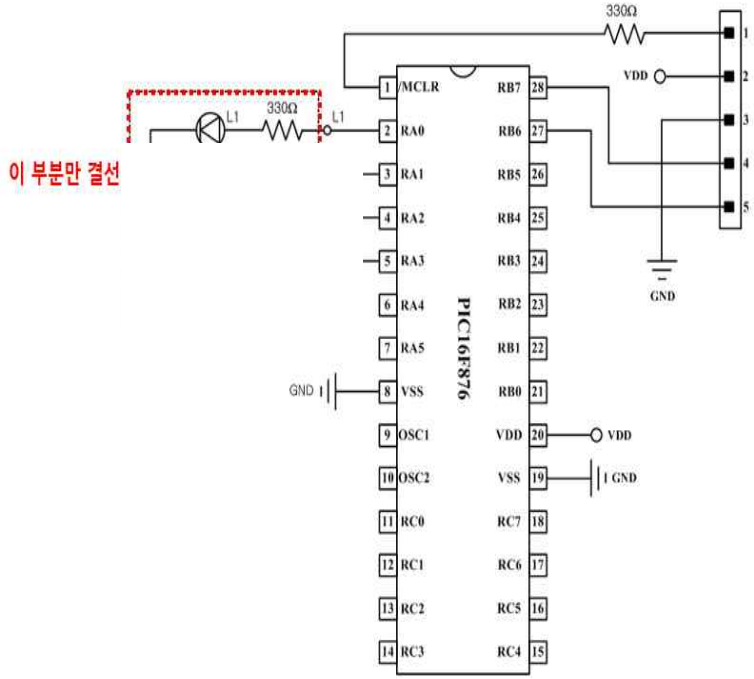
I/O Port를 사용한 LED 구동(예제 1) LED 1개 ON/OFF

- 준비 예제 실험 : 1개의 LED(L1)를 PORTA의 bit0에 연결하고 Turn ON/OFF 시키기

(Step into 기능으로 보기)

Port A : 출력 1개➔ LED(L1), TURN ON/OFF 동작 무한

```
; STANDARD HEADER FILE
                    PROCESSOR          16F876A
    ; --- REGISTER FILES 선언 -------
    ; BANK 0
    INDF                    EQU                 00H
    TMR0                    EQU                 01H
    PCL                     EQU                 02H
    STATUS      EQU                     03H
    FSR                     EQU                 04H
    PORTA       EQU                     05H
    EEDATA      EQU                     08H
    EEADR                   EQU                 09H
    PCLATH      EQU                     0AH
    INTCON      EQU                     0BH
    ; BANK 1
    OPTINOR     EQU                     81H
    TRISA                   EQU                 85H
    EECON1      EQU                     88H
    EECON2      EQU                     89H
    ADCON1      EQU                     9FH
  ; --- STATUS BITS 선언 -----
    IRP                     EQU                 7
    RP1                     EQU                 6
```



I/O Port를 사용한 LED 구동 (그림 1)

```
RP0                     EQU             5
NOT_TO      EQU                 4
NOT_PD      EQU                 3
ZF                      EQU             2   ; ZERO FLAG BIT
DC                      EQU             1   ; DIGIT CARRY/BORROW BIT
CF                      EQU             0   ; CARRY/BORROW FLAG BIT
; -- INTCON BITS 선언 -----
; -- OPTION BITS 선언 -----
W                       EQU             B'0'  ; W 변수를 0으로 선언
F                       EQU             .1    ; F 변수를 1로 선언
; MAIN PROGRAM
            ORG         0000
            BSF                     STATUS,RP0 ;Bank 1
            MOVLW       B ‘11111110'
            MOVWF       TRISA
            MOVLW       B'00000111 ‘ -> PORTA를 디지털로 사용하겠다
            MOVWF       ADCON1 -> PORTA를 디지털로 사용하겠다
            BCF         STATUS,RP0 ;Bank 0
            MOVLW       00H
            MOVWF       PORTA ; PORTA bit 0에 0출력 (초기화)
LOOP        MOVLW       01H
            MOVWF       PORTA ; PORTA bit 0에 1출력
            MOVLW       00H
            MOVWF       PORTA ; PORTA bit 0에 0출력
            GOTO        LOOP
            END
```

I/O Port를 사용한 LED 구동(그림 1) 실험 1

```
; STANDARD HEADER FILE
                    PROCESSOR       16F876A
                ; --- REGISTER FILES 선언 -------
                ; BANK 0
                INDF                EQU             00H
                TMR0                EQU             01H
                PCL                 EQU             02H
                STATUS      EQU             03H
                FSR                 EQU             04H
                PORTA       EQU             05H
                PORTB               EQU             06H
                EEDATA      EQU             08H
                EEADR               EQU             09H
                PCLATH      EQU             0AH
                INTCON      EQU             0BH
                ; BANK 1
                OPTINOR     EQU             81H
                TRISA               EQU             85H
                TRISB               EQU             86H
                EECON1      EQU             88H
                EECON2      EQU             89H
               ADCON1       EQU             9FH
              ; --- STATUS BITS 선언 -----
                IRP                 EQU             7
                RP1                 EQU             6
```

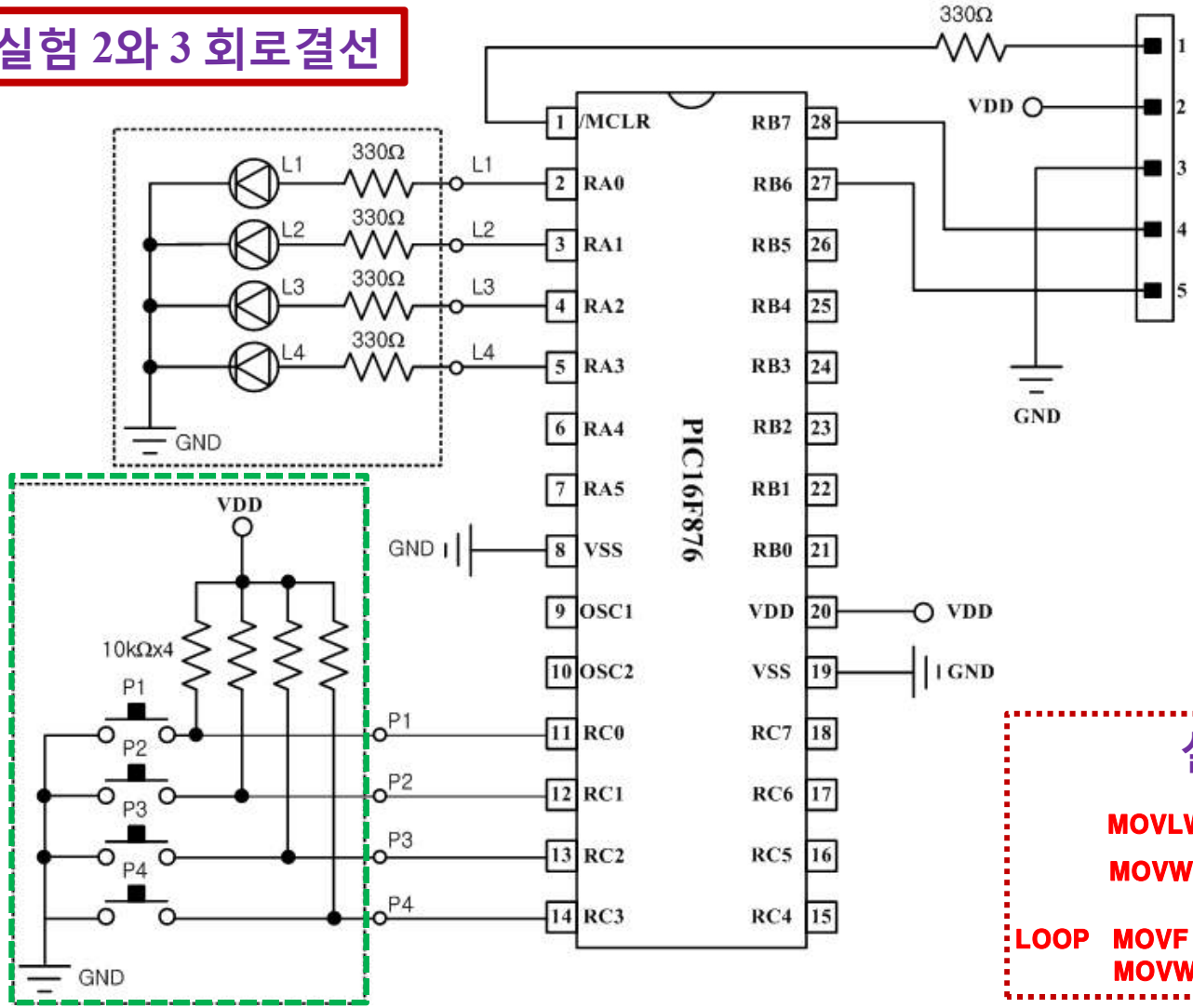
I/O Port를 사용한 LED 구동 (그림 1)

```
RP0                     EQU                     5
NOT_TO      EQU                 4
NOT_PD      EQU                 3
ZF                      EQU                     2   ; ZERO FLAG BIT
DC                      EQU                     1   ; DIGIT CARRY/BORROW BIT
CF                      EQU                     0   ; CARRY/BORROW FLAG BIT
; -- INTCON BITS 선언 -----
; -- OPTION BITS 선언 -----
W                       EQU                     B'0'  ; W 변수를 0으로 선언
F                       EQU                     .1    ; F 변수를 1로 선언
; MAIN PROGRAM
                ORG         0000
                BSF                     STATUS,RP0 ;Bank 1
                MOVLW       B'00000000'
                MOVWF       TRISA
                MOVLW       B'00001111 '        -> 여기서는 불필요 (PORTB는 사용하지 않음)
                MOVWF       TRISC   ; error 발생 -> TRISB 로 수정   -> 여기서는 불필요 (PORTB는 사용하지 않음)
                MOVLW       B'00000111  ' -> PORTA를 디지털로 사용하겠다
                MOVWF       ADCON1 -> PORTA를 디지털로 사용하겠다
                BCF                     STATUS,RP0 ;Bank 0
;
                MOVLW       00H
                MOVWF       PORTA
LOOP            INCF        PORTA,F
                GOTO        LOOP
                END
```

I/O Port를 사용한 LED 구동(그림 2)

◘ LED & Push 관련 회로도

실험 2와 3 회로결선



실험 2와 3 P/G hint

```
        MOVLW   B'00001111' ;PORTC 4비트 입력
        MOVWF   TRISC ;PORTC 4비트 입력

LOOP    MOVF    PORTC,W ; PORTC의 값 W에 저장
        MOVWF   PORTA   ; W값을 PORTA에 출력
```

I/O Port를 사용한 LED 구동(실험 2)

```
; STANDARD HEADER FILE
                PROCESSOR       16F876
        ; --- REGISTER FILES 선언 -------
        ; BANK 0
        INDF                    EQU             00H
        TMR0                    EQU             01H
        PCL                     EQU             02H
        STATUS  EQU                     03H
        FSR                     EQU             04H
        PORTA   EQU                     05H
        PORTB                   EQU             06H
        PORTC                   EQU             07H
        EEDATA  EQU                     08H
        EEADR                   EQU             09H
        PCLATH  EQU                     0AH
        INTCON  EQU                     0BH
        ; BANK 1
        OPTINOR EQU                     81H
        TRISA                   EQU             85H
        TRISB                   EQU             86H
        TRISC                   EQU             87H
        EECON1  EQU                     88H
        EECON2  EQU                     89H
      ; --- STATUS BITS 선언 -----
        IRP                     EQU             7
        RP1                     EQU             6
```

I/O Port를 사용한 LED 구동 (실험 2)

```
RP0                     EQU                     5
NOT_TO          EQU             4
NOT_PD          EQU             3
ZF                      EQU                     2   ; ZERO FLAG BIT
DC                      EQU                     1   ; DIGIT CARRY/BORROW BIT
CF                      EQU                     0   ; CARRY/BORROW FLAG BIT
; -- INTCON BITS 선언 -----
; -- OPTION BITS 선언 -----
W                       EQU                     B'0'  ; W 변수를 0으로 선언
F                       EQU                     .1    ; F 변수를 1로 선언
; MAIN PROGRAM
                ORG             0000
                BSF                     STATUS,RP0
                MOVLW           B'00000000'
                MOVWF           TRISA
                MOVLW           B'00001111'  -> 여기서는 불필요 (PORTB는 사용하지 않음)
                MOVWF           TRISB   -> 여기서는 불필요 (PORTB는 사용하지 않음)
                MOVLW           B'00001111' ;PORTC의 4비트를 입력으로 설정
                MOVWF           TRISC ;PORTC의 4비트를 입력으로 설정
                MOVLW           B'00000111 '
                MOVWF           ADCON1
                BCF             STATUS,RP0
;
                MOVLW           00H
                MOVWF           PORTA

    LOOP        MOVF            PORTC,W   ; PORTC의 값을 W에 저장
                MOVWF           PORTA     ; W값(PORTC의 값)을 PORTA에 출력
                GOTO            LOOP
                END
```

설정하지 않아도 PORTC 는 입력이됨

I/O Port를 사용한 LED 구동(그림 2)
– 스위치 누를때 LED 불 켜기 위한 P/G (실험 3)

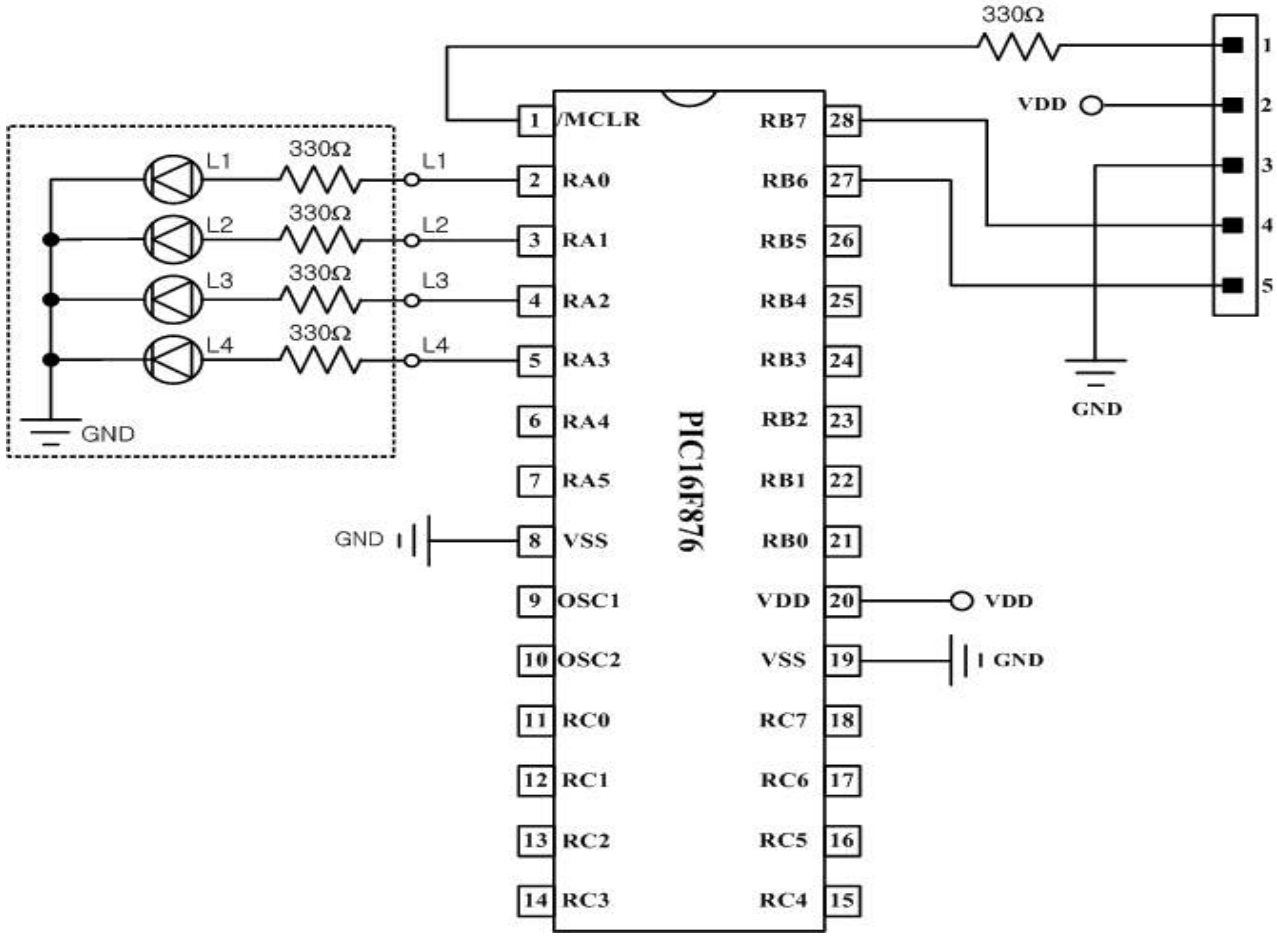
```
; STANDARD HEADER FILE
                    PROCESSOR           16F876
                ; --- REGISTER FILES 선언 -------
                ; BANK 0
                INDF                EQU             00H
                TMR0                EQU             01H
                PCL                 EQU             02H
                STATUS      EQU                 03H
                FSR                 EQU             04H
                PORTA       EQU                 05H
                PORTB               EQU             06H
                PORTC               EQU             07H
                EEDATA      EQU                 08H
                EEADR               EQU             09H
                PCLATH      EQU                 0AH
                INTCON      EQU                 0BH
                ; BANK 1
                OPTINOR     EQU                 81H
                TRISA               EQU             85H
                TRISB               EQU             86H
                TRISC               EQU             87H
                EECON1      EQU                 88H
                EECON2      EQU                 89H
            ; --- STATUS BITS 선언 -----
                IRP                 EQU             7
                RP1                 EQU             6
```

I/O Port를 사용한 LED 구동(그림 2)
– 스위치 누를때 LED 불 켜기 위한 P/G (실험 3)

```
RP0                     EQU                     5
NOT_TO          EQU             4
NOT_PD          EQU             3
ZF                      EQU                     2   ; ZERO FLAG BIT
DC                      EQU                     1   ; DIGIT CARRY/BORROW BIT
CF                      EQU                     0   ; CARRY/BORROW FLAG BIT
; -- INTCON BITS 선언 -----
; -- OPTION BITS 선언 -----
W                       EQU                     B'0'  ; W 변수를 0으로 선언
F                       EQU                     .1    ; F 변수를 1로 선언
; MAIN PROGRAM
                ORG             0000
                BSF                             STATUS,RP0
                MOVLW           B'00000000'
                MOVWF           TRISA
                MOVLW           B'00001111'
                MOVWF           TRISB
                MOVLW           B'00001111' ;PORTC의 4비트를 입력으로 설정
                MOVWF           TRISC ;PORTC의 4비트를 입력으로 설정
                MOVLW           B'00000111  '
                MOVWF   ADCON1
                BCF                             STATUS,RP0
;
                MOVLW           00H
                MOVWF           PORTA

        LOOP    COMF            PORTC,0   ; PORTC의 값을 반전시켜서 W에 저장
                MOVWF           PORTA     ; W값(PORTC의 값)을 PORTA에 출력
                 GOTO LOOP
                 END
```

I/O Port를 사용한 LED 구동(그림 1)
2장 실험1 에 지연함수추가 해보자



I/O Port를 사용한 LED 구동(그림 1)

```
; STANDARD HEADER FILE
                PROCESSOR       16F876A
        ; --- REGISTER FILES 선언 -------
        ; BANK 0
        INDF                    EQU             00H
        TMR0                    EQU             01H
        PCL                     EQU             02H
        STATUS  EQU                     03H
        FSR                     EQU             04H
        PORTA   EQU                     05H
        PORTB                   EQU             06H
        EEDATA  EQU                     08H
        EEADR                   EQU             09H
        PCLATH  EQU                     0AH
        INTCON  EQU                     0BH
        ; BANK 1
        OPTINOR EQU                     81H
        TRISA                   EQU             85H
        TRISB                   EQU             86H
        EECON1  EQU                     88H
        EECON2  EQU                     89H
       ADCON1   EQU                     9FH
      ; --- STATUS BITS 선언 -----
        IRP                     EQU             7
        RP1                     EQU             6
```

I/O Port를 사용한 LED 구동 (그림 1)

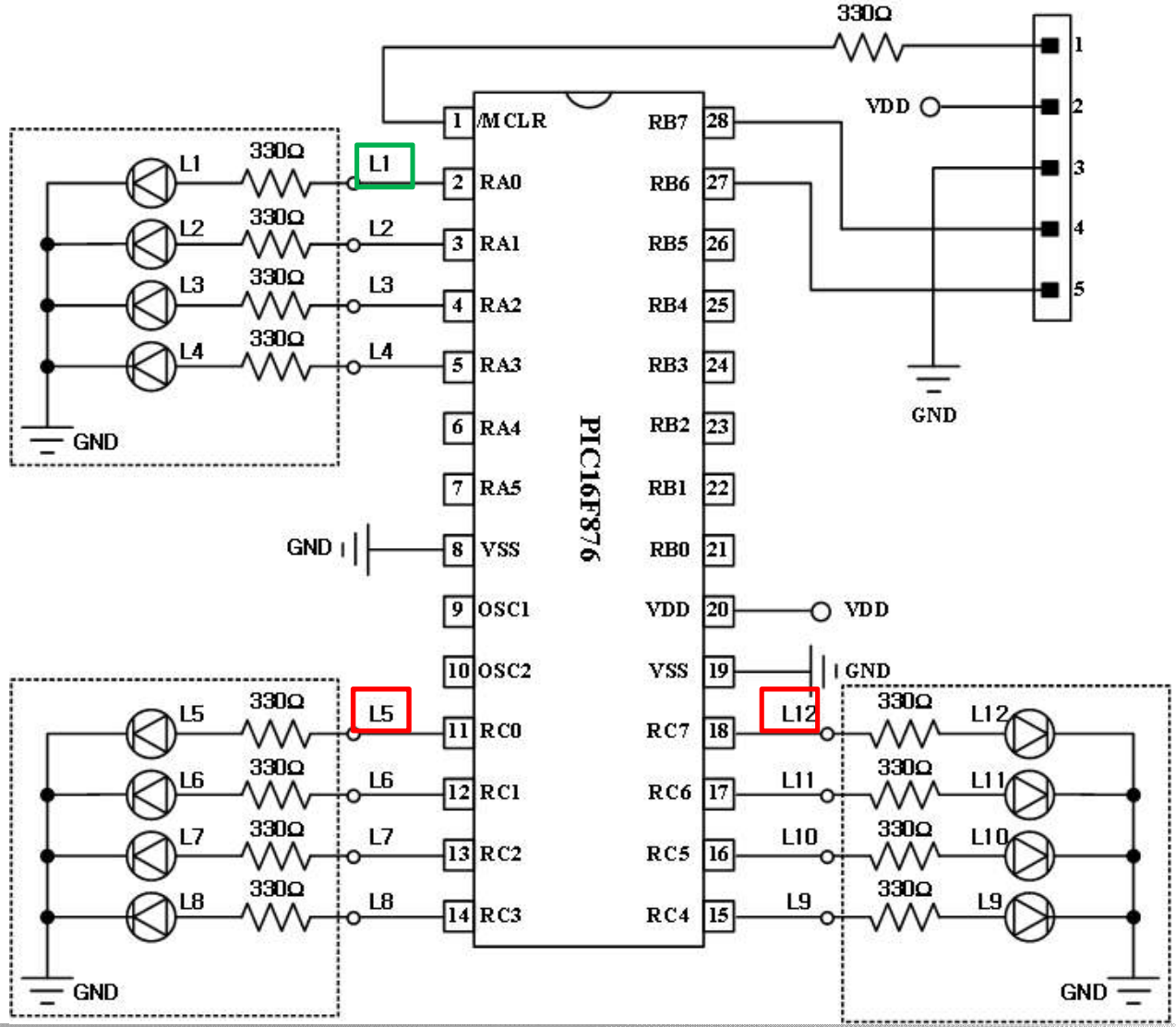
```
            RP0                 EQU                 5
            NOT_TO      EQU                 4
            NOT_PD      EQU                 3
            ZF                  EQU                 2   ; ZERO FLAG BIT
            DC                  EQU                 1   ; DIGIT CARRY/BORROW BIT
            CF                  EQU                 0   ; CARRY/BORROW FLAG BIT
            W                   EQU                 B'0'  ; W 변수를 0으로 선언
            F                   EQU                 .1    ; F 변수를 1로 선언
            DBUF1               EQU     24H  ;GPR register에 번지지정!!!!!
            DBUF2               EQU     25H  ;GPR register에 번지지정!!!!!
            ; MAIN PROGRAM
                        ORG     0000
                        BSF                 STATUS,RP0 ;Bank 1
                        MOVLW   B'00000000'
                        MOVWF   TRISA
                        MOVLW   B'00001111 '    ;여기서는 불필요 (PORTB는 사용하지 않음)
                        MOVWF   TRISB   ; 여기서는 불필요 (PORTB는 사용하지 않음)
                        MOVLW   B'00000111  ' -> PORTA를 디지털로 사용하겠다
                        MOVWF   ADCON1 -> PORTA를 디지털로 사용하겠다
                        BCF                 STATUS,RP0 ;Bank 0
            ;
                        MOVLW   00H
                        MOVWF   PORTA
            LOOP        INCF    PORTA,F ; 0000, 0001, 0010, 0011, 0100, ...,1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111 순서로 TURN ON
                        CALL    DELAY
                        GOTO    LOOP

; Subroutine
DELAY
            MOVLW       .125
            MOVWF       DBUF1       ; 긴 시간을 설정하기 위한 변수
LP1         MOVLW       .200
            MOVWF       DBUF2
LP2         NOP
            DECFSZ      DBUF2, F
            GOTO        LP2         ; ZERO가 아니면 GOTO LP2 수행
            DECFSZ      DBUF1, F    ;변수를 감소시켜 가면서 00이 되었나 확인
            GOTO        LP1         ; ZERO가 아니면 GOTO LP1수행
            RETURN                  ; 부 프로그램 종료
END
```

실험 3장. LED에 생명부여하기

Lab.1 I/O Port를 사용한 입출력

◘ LED & Push 관련 회로도



Lab.1 I/O Port를 사용한 입출력

L12 L5

□ 실험 1. LED 회전 (8개) ex3_1 (최하위(L5)부터 0이 이동), ex3_1_1 (최상위(L12)부터 0이 이동)

: PORTC 8개 LED출력(PORT C에 연결된 LED 8개에 대해 순차적으로 OFF가 이동)

--→ 11111111->11111110->11111101->11111011->11110111 -> 11101111->11011111 ->10111111 ->01111111->11111111 ->11111110...

□ 실험 2. LED 회전 시키기 (12개) ex3_1_2 (최하위(L1)부터 최상위(L12)까지 0이 이동)

- ■ H/W 설계 조건
  - ▪ Port A : 출력 4개➔ LED(L1-L4)
  - ▪ Port C : 출력 8개 ➔ LED(L5-P12)
- ■ S/W 설계 조건
  - ▪ PORTC+PORTA에 연결된 LED 가 순차적으로 1개씩 회전(OFF가)하게 한다.

□ 실험 3. LED 2개씩 회전 시키기 (8개) -→ Home Work 3 ex3_1_3

- ■ H/W 설계 조건
  - ▪ Port C : 출력 8개 ➔ LED(L1-P8)
- ■ S/W 설계 조건

: PORTC 8개 LED출력(PORT C에 연결된 LED 8개에 대해 2개씩 순차적으로 OFF가 이동

--→ 11111111->11111100->11111001->11110011->11100111 -> 11001111->10011111 ->00111111

● 부가 실험1 : 실험 1부터 3까지에서 LED ON이 이동하게 수정하시오

부가실험 2 : 8개의 LED(L1,2,3,4,5,6,7)를 Port C를 출력으로 하여 조건대로 Turn ON 시키기

PORTC에 연결된 LED가 00000000, 00000001, 00000010, 00000011, 00000100, …, 00001000, 00001001, 00001010, 00001011, 00001100, 00001101, 00001110, 00001111, 00010000, 00010001,…. 111111110, 11111111 순서로 TURN ON

Status register (address : 03H(bank 0), 83H(bank 1), 103H(bank 2), 183H(bank 3))

IRP(bit7)	RP1	RP0	TO	PD	Z	DC	C(bit0)

IRP(bit7) : Register Bank Select bit (indirect addressing)

RP1 RP0 : Register Bank Select bits (00, 01, 10, 11) (direct addressing)

C(carry /borrow), DC(digit carry/borrow), Z(zero), PD, TO로 5가지가 있다. 일반적으로 많이 사용되는 status는 C, Z이다.

C : STATUS REG.의 BIT0으로 ADD, SUB, rotate 명령어 사용 결과 CARRY가 발생하면 '1'로 SET되고 발생하지 않으면 '0'이 됨

DC : STATUS REG.의 BIT1로 명령어 사용 결과로 bit3에서 bit4로 자리올림이 발생하면 '1'로 SET되고, 발생하지 않으면 '0'이 됨
(BCD 연산에서 사용하며, 일반적인 경우는 거의 사용하지 않는다)

Z : STATUS REG.의 BIT2로 명령어 사용 결과 값이 ZERO 이면 '1'로 SET되고 아니면 '0'이됨

Status register (address : 03H(bank 0), 83H(bank 1), 103H(bank 2), 183H(bank 3))

- STATUS reg.의 C bit(bit 0)는 연산결과 CARRY의 발생 유무에 따라서 결정되며, 기본 명령어는 ALU와 연관된 명령어 들이다.
  : ADDWF F,d : ADDLW K --- 연산 결과 그대로 영향 받음
  : SUBWF F,d : SUBLW K --- 아래 주의 참고 (다음페이지 참조)
  : RLF F,d : RRF F,d --- 'C'를 만드는 것뿐만 아니라 'C'를 받아들임으로 주의 요망

- Zero flag에 영향을 주는 명령어는 아래와 같은 명령어로 가장 많으며, 명령의 수행 상태가 0이 만들어지는 경우에 설정된다.
  : ADD, AND, CLR, COMF, DECF, INCF, IOR, MOVF, SUB, XOR

SUBLW K 와 Status register

- PIC에서 Subtraction(뺄셈)은 2의 보수를 이용(음수로 표현)하여 덧셈을 행한다

Ex) SUBLW 07H 는?

1) W 에는 05H 가 저장되어 있는 경우

07H + (100H-05H) = 07H + FBH = 02H -→ C=1, DC=1, Z=0 이된다.

2) W 에는 08H 가 저장되어 있는 경우

07H + (100H-08H) = 07H + F8H = FFH -→ C=0, DC=0, Z=0 이된다.

* 2진수(binary number)의 보수 -→ 음의 수를 표현하기 위한 방법

: 주어진 이진수보다 한자리 높고 가장 높은 자리가 1인 2진수에서 주어진 수를 빼서 얻거나, 1의 보수에 1을 더하여 구할 수 있다. (1의 보수는 이진수의 0은 1로, 1은 0으로 대체한 수)

예) 0011(3)의 2의 보수는 다음과 같이 구할 수 있다.

* 10000 – 0011 = 1101
* 1100(1의 보수) + 1 = 1101

조건에 따라서 분기(branch) 하는 명령어

- **상태를 확인하여(Bit Test, Skip if Clear or Set) 분기 하는 명령어**

  : BTFSC f, b (f-file register의 bit b의 값이 0 이면 skip하라는 의미)

  : BTFSS f, b (f-file register의 bit b의 값이 1 이면 skip하라는 의미)

- **연산 결과로 바로 분기 하는 명령어**

  : DECFSZ f, d (f-file register를 1씩 감소시키면서 d에 저장하고, 그 값이 0이 되면 바로 아래 명령어를 SKIP, d=0이면 W, 1이면 f에 저장. 특히 **이 명령어는 STATUS에 영향을 전혀 주지 않는**다.)

  : INCFSZ f, d (f-file register를 1씩 증가시키면서 d에 저장하고, 그 값이 0이 되면 바로 아래 명령어를 SKIP)

시간 지연 프로그램

예9) 어떤 프로그램을 100번 반복할 경우의 프로그램

```
                MOVLW     .100
                MOVWF     COUNT       ; 100번을 확인하기 위한 변수
LP              .                     ; 반복되는 PROGRAM 영역
                .
                DECFSZ    COUNT,F     ; 변수를 감소시켜 가면서 00 ; 이 되었나 확인
                GOTO      LP          ; zero가 아니면 GOTO LP 수행
                .                     ; zero이면 NEXT PROGRAM 시작
```

☞ 당연히 COUNT는 FILE REGISTER 중 USER 영역 안의 번지로 선언

(Ex) CONNT EQU 23H)

●지연프로그램

NOP (한 사이클 지연)

Lab.1 I/O Port를 사용한 입출력

예 9-1) 지연시간 약 0.5msec

```
CALL     DELAY

; Subroutine -→ Loop 1개
DELAY
         MOVLW    .130  ; 8bit register이므로 .255까지 가능
         MOVWF    DBUF1      ; 긴 시간을 설정하기 위한 변수
LP1      NOP
         DECFSZ   DBUF1, F   ;변수를 감소시켜 가면서 00이 되었나 확인
         GOTO     LP1        ; ZERO가 아니면 GOTO LP1수행
         RETURN              ; 부 프로그램 종료
```

☞ 이 프로그램은 부프로그램으로 작성되었으므로 필요한 위치에서 불러서 사용 하면 되고, 프로그램으로는 'CALL DELAY' 라고 함

지연시간 : 130*(1+ 1+ 2)* 1usec= 520usec = 0.5 msec **-→ 1명령어 수행에** 1usec **소요 (부록** Data sheet 159-160p)

(4 clock* 1/4MHz = 1usec)

-→ 지연시간 부정확하며 정확한 시간 요구시에는 timer**와** interrupt **기능 사용**

-→ DBUF1를 주소지정하기!!!

-→ 지연시간이 짧다 어떻게 할까?

Lab.1 I/O Port를 사용한 입출력

예 10) 약 100msec 지연

```
CALL    DELAY

; Subroutine
DELAY
        MOVLW   .125
        MOVWF   DBUF1       ; 긴 시간을 설정하기 위한 변수
LP1     MOVLW   .200
        MOVWF   DBUF2
LP2     NOP
        DECFSZ  DBUF2, F
        GOTO    LP2         ; ZERO가 아니면 GOTO LP2 수행
        DECFSZ  DBUF1, F    ;변수를 감소시켜 가면서 00이 되었나 확인
        GOTO    LP1         ; ZERO가 아니면 GOTO LP1수행
        RETURN              ; 부 프로그램 종료
```

☞ 이 프로그램은 부프로그램으로 작성되었으므로 필요한 위치에서 불러서 사용 하면 되고, 프로그램으로는 'CALL DELAY' 라고 함

지연시간 : 125*(200*(1+ 1+ 2)* 1usec)= 100,000usec = 100 msec **-→ 1명령어 수행에** 1usec **소요 (부록** Data sheet 159-160p)

-→ 지연시간 부정확하며 정확한 시간 요구시에는 timer**와** interrupt **기능 사용**

Lab.1 I/O Port를 사용한 입출력

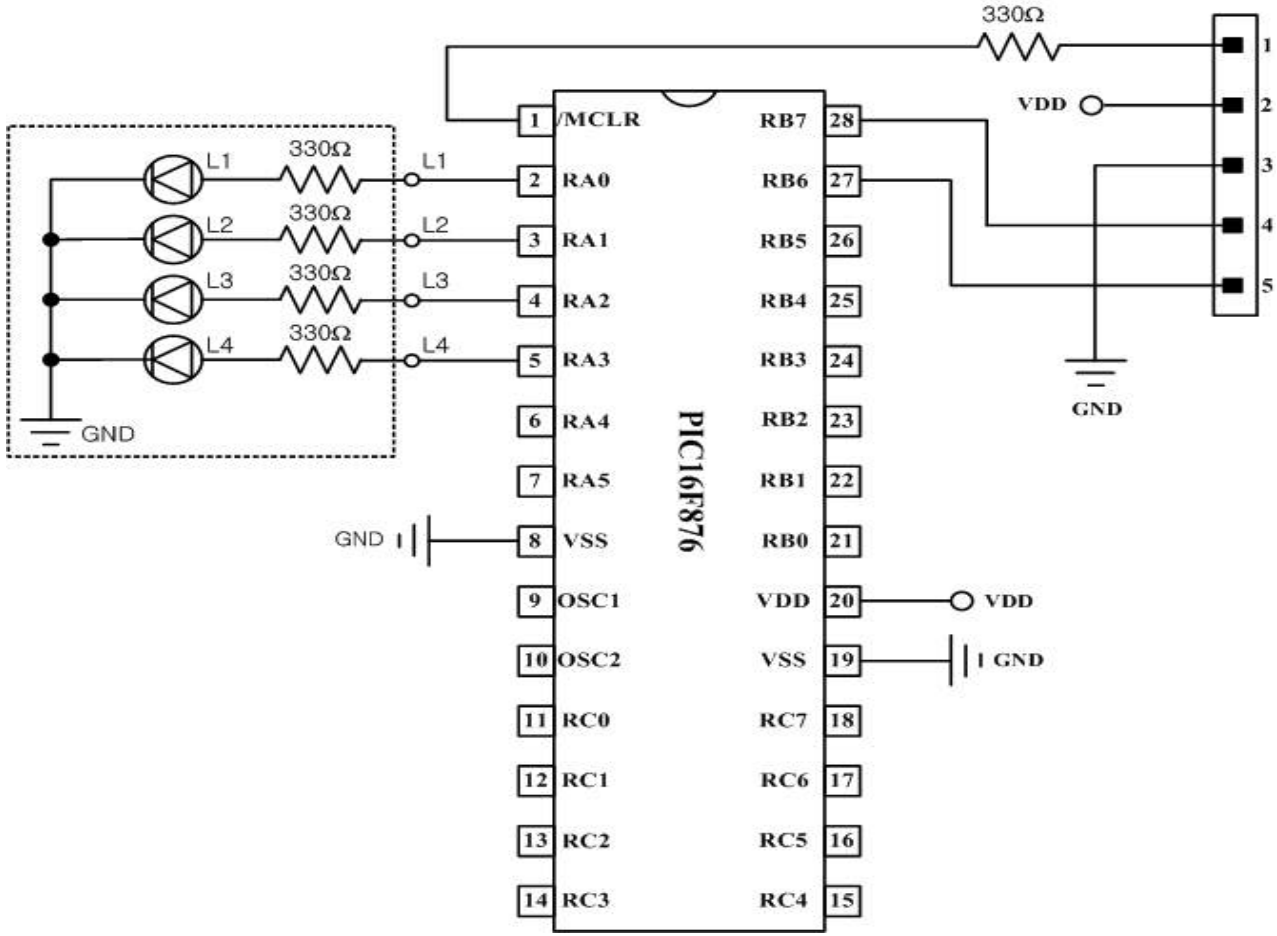
예 10-1) 약 100*200=20sec 지연

```
CALL    DELAY

; Subroutine
DELAY
        MOVLW   .125
        MOVWF   DBUF1       ; 긴 시간을 설정하기 위한 변수
LP1     MOVLW   .200
        MOVWF   DBUF2
LP2     MOVLW   .200
        MOVWF   DBUF3
LP3     NOP
        DECFSZ  DBUF3, F
        GOTO    LP3         ; ZERO가 아니면 GOTO LP2 수행
        DECFSZ  DBUF2, F    ;변수를 감소시켜 가면서 00이 되었나 확인
        GOTO    LP2         ; ZERO가 아니면 GOTO LP1수행
        DECFSZ  DBUF1, F    ;변수를 감소시켜 가면서 00이 되었나 확인
        GOTO    LP1         ; ZERO가 아니면 GOTO LP1수행
        RETURN              ; 부 프로그램 종료
```

지연시간 : 125*200*(200*(1+ 1+ 2)* 1usec)= 200*100 msec -→ **1명령어 수행에** 1usec **소요 (부록** Data sheet 159-160p)

I/O Port를 사용한 LED 구동(그림 1)
2장 실험1 에 지연함수추가 해보자



I/O Port를 사용한 LED 구동(그림 1)

```
; STANDARD HEADER FILE
                PROCESSOR       16F876A
        ; --- REGISTER FILES 선언 -------
        ; BANK 0
        INDF                    EQU             00H
        TMR0                    EQU             01H
        PCL                     EQU             02H
        STATUS  EQU                     03H
        FSR                     EQU             04H
        PORTA   EQU                     05H
        PORTB                   EQU             06H
        EEDATA  EQU                     08H
        EEADR                   EQU             09H
        PCLATH  EQU                     0AH
        INTCON  EQU                     0BH
        ; BANK 1
        OPTINOR EQU                     81H
        TRISA                   EQU             85H
        TRISB                   EQU             86H
        EECON1  EQU                     88H
        EECON2  EQU                     89H
       ADCON1   EQU                     9FH
      ; --- STATUS BITS 선언 -----
        IRP                     EQU             7
        RP1                     EQU             6
```

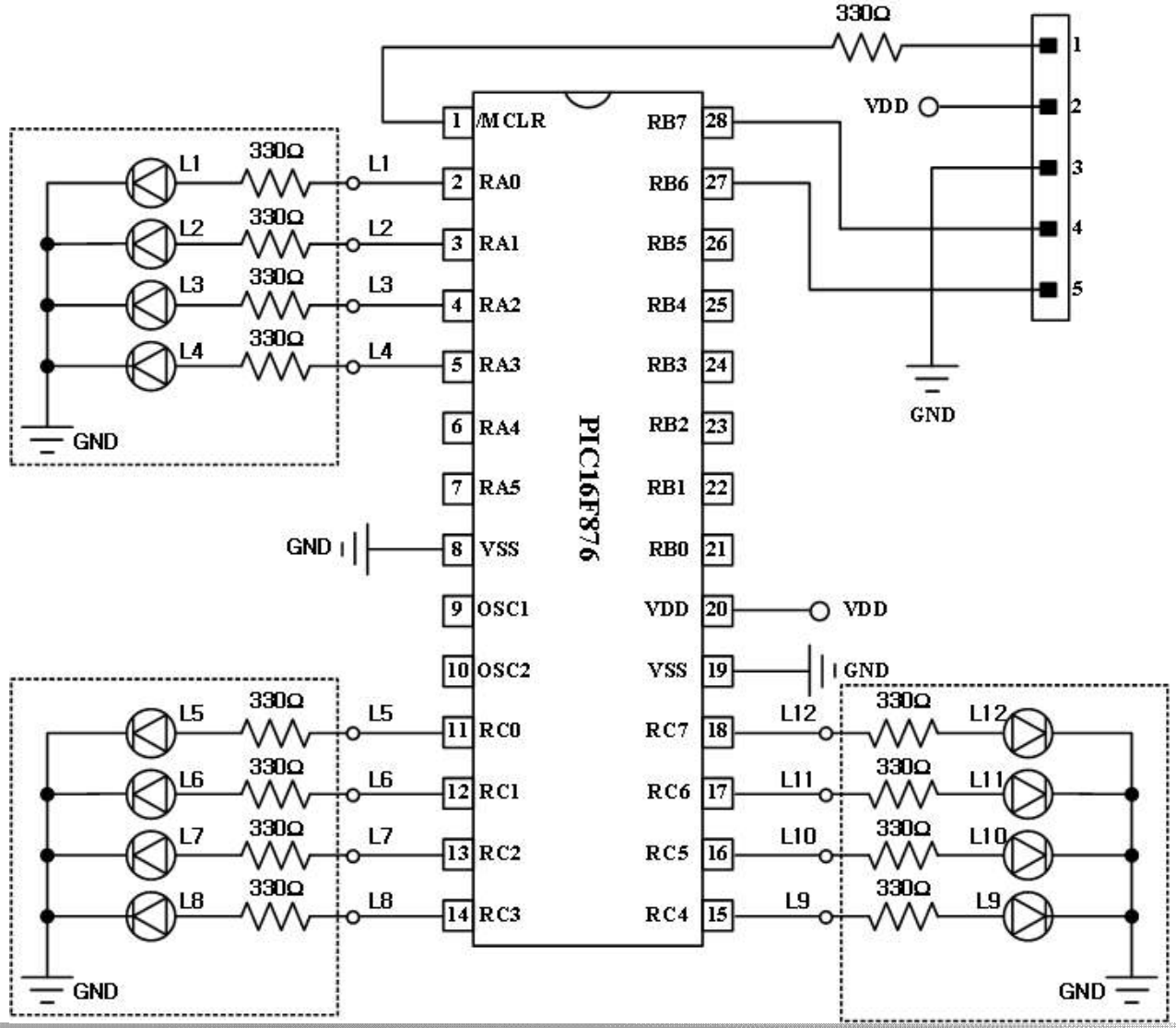
I/O Port를 사용한 LED 구동 (그림 1)

```
RP0                     EQU                     5
NOT_TO      EQU                 4
NOT_PD      EQU                 3
ZF                      EQU                     2   ; ZERO FLAG BIT
DC                      EQU                     1   ; DIGIT CARRY/BORROW BIT
CF                      EQU                     0   ; CARRY/BORROW FLAG BIT
W                       EQU                     B'0'  ; W 변수를 0으로 선언
F                       EQU                     .1    ; F 변수를 1로 선언
DBUF1                   EQU     24H  ;GPR register에 번지지정!!!!!
DBUF2                   EQU     25H  ;GPR register에 번지지정!!!!!
; MAIN PROGRAM
                ORG     0000
                BSF                 STATUS,RP0 ;Bank 1
                MOVLW   B'00000000'
                MOVWF   TRISA
                MOVLW   B'00001111 '    ;여기서는 불필요 (PORTB는 사용하지 않음)
                MOVWF   TRISB   ; 여기서는 불필요 (PORTB는 사용하지 않음)
                MOVLW   B'00000111  ' -> PORTA를 디지털로 사용하겠다
                MOVWF   ADCON1 -> PORTA를 디지털로 사용하겠다
                BCF                 STATUS,RP0 ;Bank 0
;
                MOVLW   00H
                MOVWF   PORTA
LOOP            INCF    PORTA,F ; 0000, 0001, 0010, 0011, 0100, ...,1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111 순서로 TURN ON
                CALL    DELAY
                GOTO    LOOP

; Subroutine
DELAY
        MOVLW   .125
        MOVWF   DBUF1       ; 긴 시간을 설정하기 위한 변수
LP1     MOVLW   .200
        MOVWF   DBUF2
LP2     NOP
        DECFSZ  DBUF2, F
        GOTO    LP2         ; ZERO가 아니면 GOTO LP2 수행
        DECFSZ  DBUF1, F    ;변수를 감소시켜 가면서 00이 되었나 확인
        GOTO    LP1         ; ZERO가 아니면 GOTO LP1수행
        RETURN              ; 부 프로그램 종료
END
```

Lab.1 I/O Port를 사용한 입출력

◘ LED & Push 관련 회로도



Lab.1 I/O Port를 사용한 입출력

□ 실험 1. LED 1개씩 회전 (8개)

: PORTC 8개 LED출력(PORT C에 연결된 LED 8개에대해 순차적으로 OFF가 이동 (최하위부터)

--→ 11111111->11111110->11111101->11111011->11110111 -> 11101111->11011111 ->10111111 ->01111111->11111111 ->11111110...

□ 실험 2. LED 회전 시키기 (12개) (최하위부터)

- ■ H/W 설계 조건
  - ▪ Port A : 출력 4개➔ LED(L1-L4)
  - ▪ Port C : 출력 8개 ➔ LED(L5-P12)
- ■ S/W 설계 조건
  - ▪ PORTC+PORTA에 연결된 LED 가 순차적으로 1개씩 회전(OFF가)하게 한다.

□ 실험 3. LED 2개씩 회전 시키기 (8개) → HW #3

- ■ H/W 설계 조건
  - ▪ Port C : 출력 8개 ➔ LED(L1-P8)
- ■ S/W 설계 조건

: PORTC 8개 LED출력(PORT C에 연결된 LED 8개에 대해 2개씩 순차적으로 OFF가 이동

--→ 11111111->11111100->11111001->11110011->11100111 -> 11001111->10011111 ->00111111

● 부가 실험1 : 실험 1부터 3까지에서 LED ON이 이동하게 수정하시오

부가실험 2 : 8개의 LED(L1,2,3,4,5,6,7)를 Port C를 출력으로 하여 조건대로 Turn ON 시키기

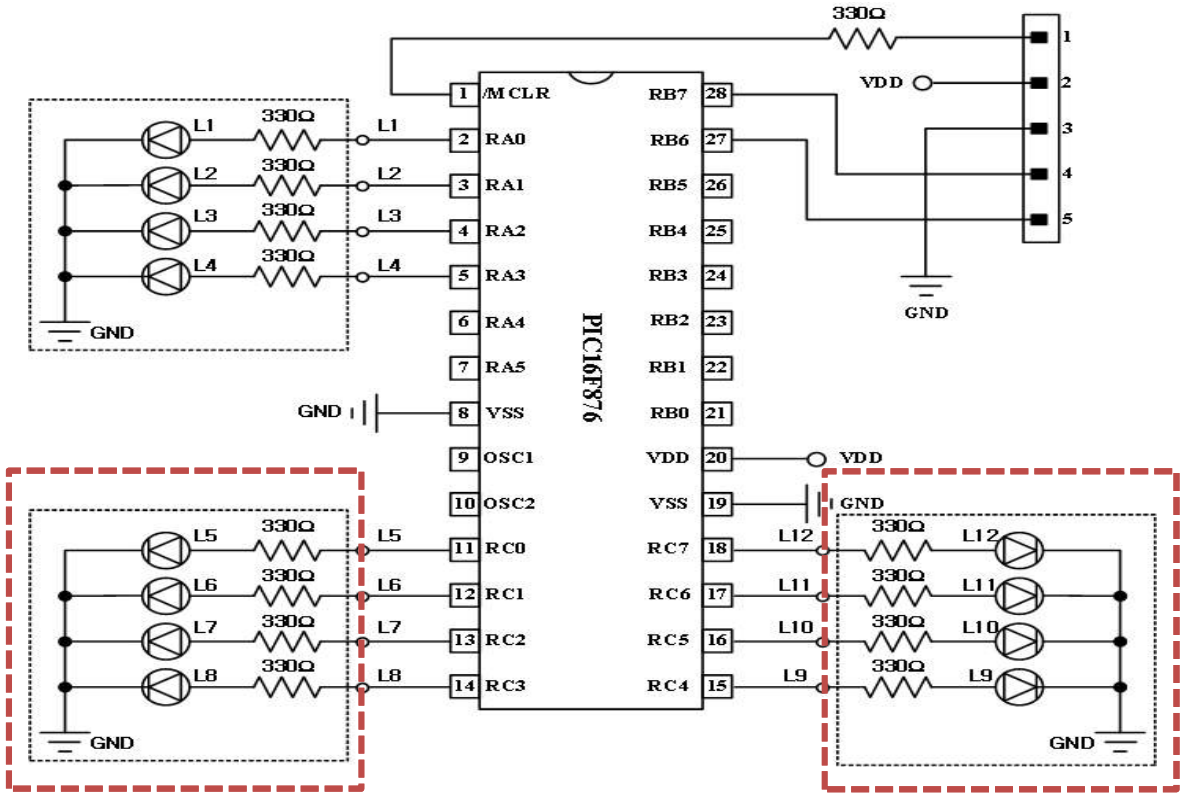
PORTC에 연결된 LED가 00000000, 00000001, 00000010, 00000011, 00000100, …, 00001000, 00001001, 00001010, 00001011, 00001100, 00001101, 00001110, 00001111, 00010000, 00010001,…. 111111110, 11111111 순서로 TURN ON

Lab.1 I/O Port를 사용한 입출력

◘ 실험 1. LED 회전 (8개)

: PORTC 8개 LED출력(PORT C에 연결된 LED 8개에대해 순차적으로 OFF가 이동

--→ 11111111->11111110->11111101->11111011->11110111 -> 11101111->11011111 ->10111111 ->01111111->11111111 ->11111110...



Lab.1 I/O Port를 사용한 입출력

```
;TEXT 실험 3 (91,PAGE)  -→ 실험과제 1
; PORTC 8개 LED출력(최하위비트부터 이동하며 불꺼짐 3.2와 동일), 11111111->11111110->11111101->11111011->11110111
;              ---> 11101111->11011111->10111111->01111111->11111111...
;  STANDARD HEADER FILE
                                PROCESSOR                       16F876
                    ; --- REGISTER FILES 선언 --------
                    ;  BANK 0
                    INDF        EQU                             00H
                    TMR0        EQU                             01H
                    PCL         EQU                             02H
                    STATUS      EQU                             03H
                    FSR         EQU                             04H
                    PORTA       EQU                             05H
                    PORTB       EQU                             06H
                    PORTC       EQU                             07H
                    EEDATA      EQU                             08H
                    EEADR       EQU                             09H
                    PCLATH      EQU                             0AH
                    INTCON      EQU                             0BH
                    ;  BANK 1
                    OPTINOR     EQU                             81H
                    TRISA       EQU                             85H
                    TRISB       EQU                             86H
                     TRISC      EQU                             87H
                    EECON1      EQU                             88H
                    EECON2      EQU                             89H
                    ADCON1      EQU                             9FH
                    ; --- STATUS BITS 선언 -----
                    IRP         EQU                             7
                    RP1         EQU                             6
                    RP0         EQU                             5
                    NOT_TO      EQU                             4
                    NOT_PD      EQU                             3
                    ZF          EQU                             2    ; ZERO FLAG BIT
                    DC          EQU                             1    ; DIGIT CARRY/BORROW BIT
                    CF          EQU                             0    ; CARRY/BORROW FLAG BIT
                    ; -- INTCON BITS 선언 -----
                    ; -- OPTION BITS 선언 -----

                    W           EQU                             B'0'  ; W 변수를 0으로 선언
                    F           EQU                             .1    ; F 변수를 1로 선언
   DBUF1   EQU    24H ;GPR레지스터에 저장 (중요)
   DBUF2   EQU    25H ;GPR레지스터에 저장 (중요)
   DBUF3   EQU    26H ;GPR레지스터에 저장 (중요)
   LED1    EQU    27H ;GPR레지스터에 저장 (중요)
  ;BUFFER   EQU    23H ;GPR레지스터에 저장 (중요)
```

Lab.1 I/O Port를 사용한 입출력

```
;  MAIN PROGRAM
                ORG             0000
                 BSF            STATUS,RP0
                 MOVLW          B'00000000'
                 MOVWF          TRISA
                 MOVLW          B'00000000'
                 MOVWF          TRISB
                 MOVLW          B'00000000' ;PORTC의 4비트를 입력으로 설정
                 MOVWF          TRISC ;PORTC의 4비트를 입력으로 설정
                 MOVLW          B'00000111'
                 MOVWF   ADCON1

                  BCF           STATUS, RP0 ;   PORT에 입출력시에는 반드시 Bank0로 해야함

                  CLRF          PORTA
                  MOVLW         B'00000001' ; 최하위부터 OFF  11111111 -> 11111110 -> 11111101 -> …. 과 RLF
                 ;  MOVLW       B'10000000' ; 최상위부터 OFF  11111111 -> 01111111 -> 10111111 -> …. 과 RRF

                  ADDLW         00                              ; Carry 를 0으로 만듦 (중요)
                  MOVWF         LED1                            ; 초기값 01를 넣음

                 MOVLW          B'11111111' ; 제일먼저 시작을 모두 ON하기 위해
                  MOVWF         PORTC

    LOOP
                  MOVF          LED1,W
                  XORLW         B'11111111'     ; off가 이동시에는 이부분 추가, ON 이동시에는 이부분을 코멘트처리
                 MOVWF          PORTC           ; W 을 PORTC에 출력
                  RLF           LED1,F          ; rotate 시킴
                  CALL          DELAY
                  GOTO          LOOP


  ;SUBROUTINE
DELAY
                  MOVLW         .255
                 MOVWF          DBUF1           ; 130번을 확인하기 위한 변수
LP1               MOVLW         .255
                  MOVWF         DBUF2
LP2               MOVLW         .10
                  MOVWF         DBUF3           ; 200번을 확인하기 위한 변수
LP3               NOP
                  DECFSZ        DBUF3,F
                  GOTO          LP3             ; ZERO가 아니면 GOTO
                  DECFSZ        DBUF2,F         ; 변수를 감소시켜 가면서 00이 되었나 확인
                  GOTO          LP2             ; ZERO가 아니면 GOTO
                 DECFSZ         DBUF1,F         ; 변수를 감소시켜 가면서 00이 되었나 확인
                  GOTO          LP1             ; ZERO가 아니면 GOTO
                  RETURN
```

Assembly Programming-Useful Program

- 비트반전

```
        BTFSC    PORTB, 0
        GOTO     BIT_COM
        BSF      PORTB, 0
        GOTO     LOOP
BIT_COM
        BCF      PORTB, 0
LOOP      ……
```

XOR 사용 –→ 비트반전

```
MOVLW     B'00000001'
XORWF     PORTB    ;PORTB에 저장
```