

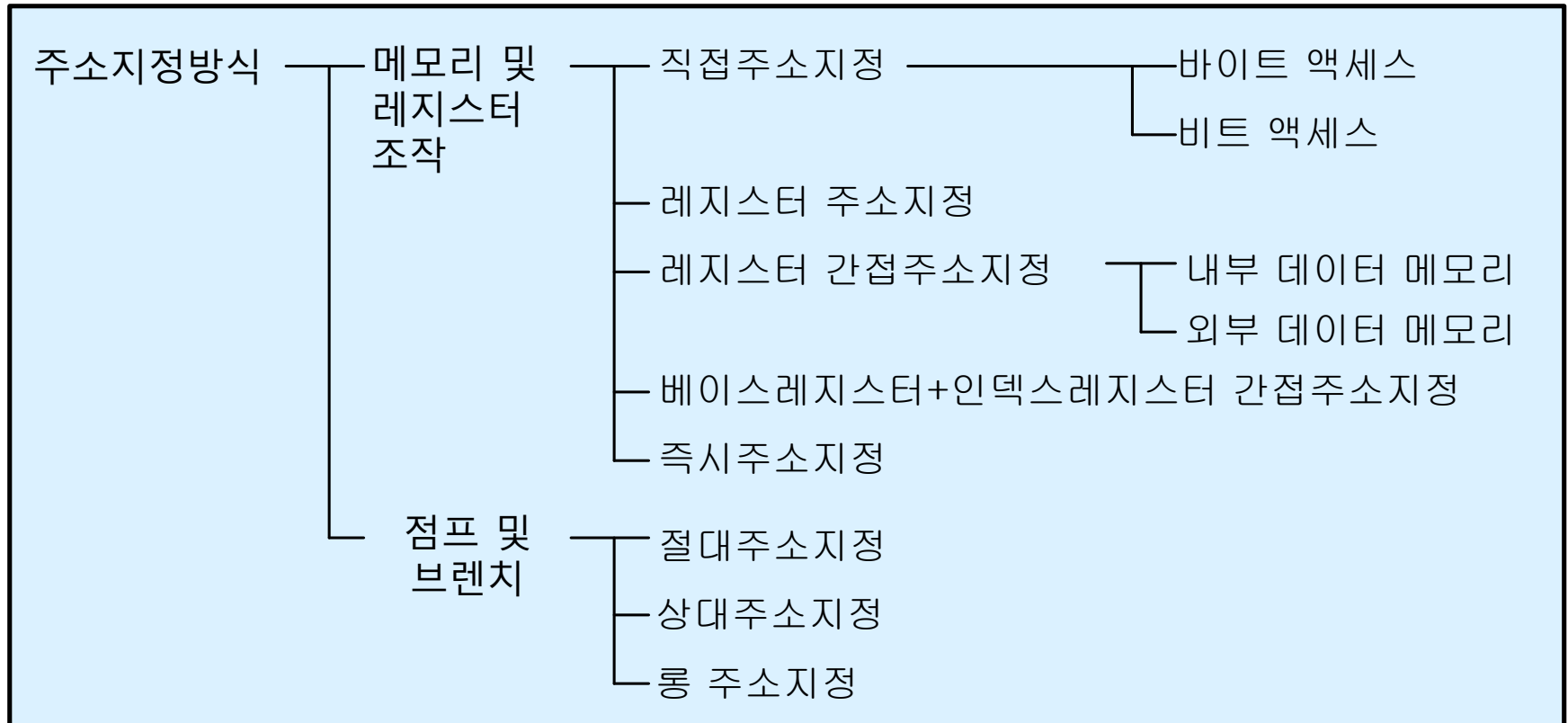
UNIT 6

INSTRUCTION SET

2012학년 2학기

마이크로 프로세서 실습

- Instruction은 addressing mode로 구분하며, 이것은 operand를 선택하는 방법을 나타낸다.



■ Direct addressing mode

- + Internal data memory 와 accumulator 사이
- + Internal data memory 와 register 사이
- + Internal data memory 와 internal data memory 사이
 - Ex) MOV A, 50H

Register addressing mode

- + Opcode가 필요로 하는 data가 register내에 저장되어 있는 경우
 - Ex) MOV A, R1

Register indirect addressing mode

- + Register 내에 저장된 값이 명령 코드가 필요로 하는 data가 저장된 장소의 address를 가지고 있는 경우
 - Internal memory에서의 데이터 전송 (ex) MOV A, @R1)
 - External memory와의 데이터 전송 (ex) MOVX A, @DPTR)

Base register + Index register indirect addressing mode

- + Opcode 가 필요로 하는 data가 external program memory에 있을 경우
 - Ex) MOVC A, @A+DPTR

■ Immediate addressing mode

- + Opcode가 필요로 하는 데이터가 상수 값 으로 직접 주어지는 경우
 - Ex) MOV A, #1FH

Relative addressing mode

- + Branch에 관련된 addressing mode이며, 현재 위치로 부터 (offset +2) 만큼 떨어진 곳으
로 떨어진 곳으로 점프하는 addressing mode, PC 값이 변하게 된다.
 - Ex) SJMP 05H

Absolute addressing mode

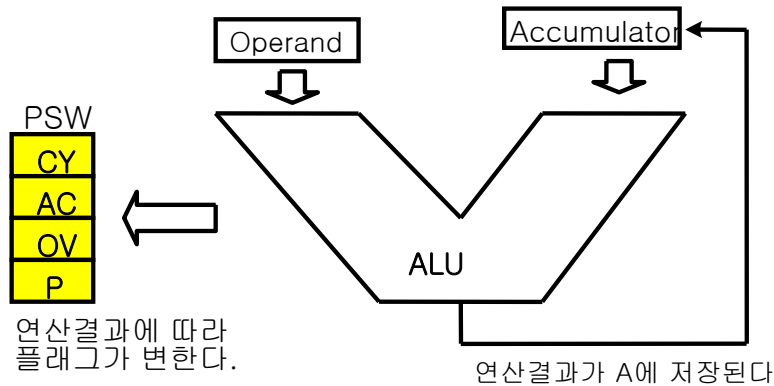
- + Absolute address 로 jump하는 방식으로 PC의 하위 11bit 값을 니모닉 다음에 위치하는
operand로 대치시킨다.
 - Ex) AJMP 01A5H

Long addressing mode

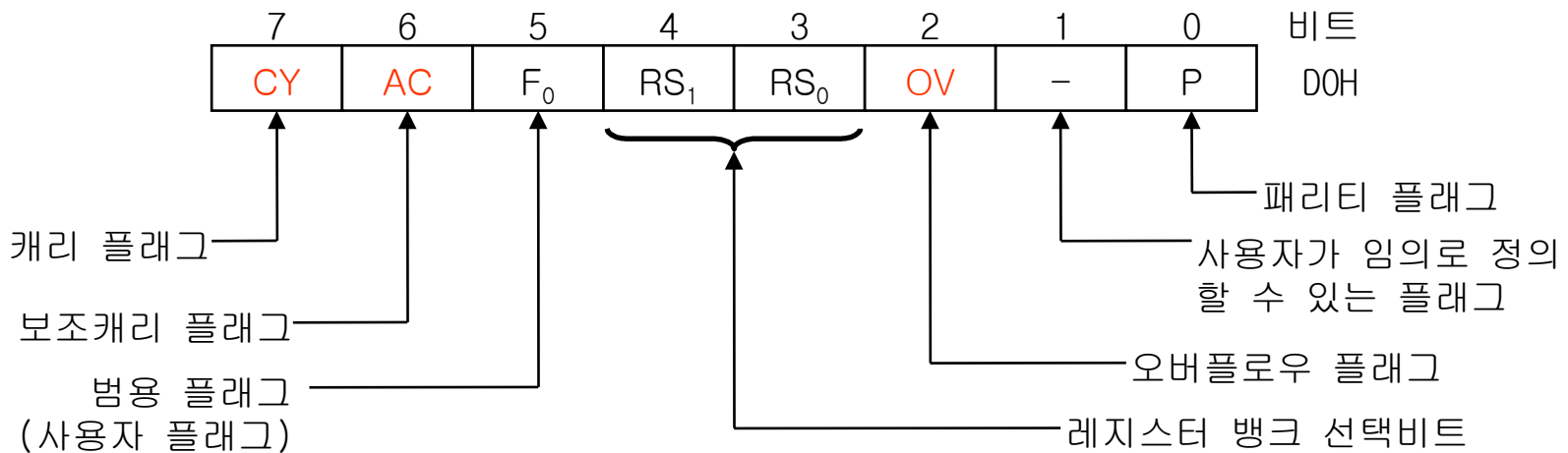
- + Absolute addressing의 일종으로 64k byte의 전 메모리 영역으로 점프를 가능하게 하는
명령어 이며 점프할 주소는 16bit로 표현한다.
 - Ex) LJMP 8000H

Generic Jump

■ ALU (Arithmetic Logic Unit)



PSW (Program Status Word)





■ CY (Carry Flag)

- + ADD/SUB instruction 에서 최상위 bit에서 Carry/Borrow가 생기면 set
- + Rotate instruction 에서 8bit으로 사용
- + BOOLEAN instruction 에서 bit accumulator로 사용됨

AC (Auxiliary flag)

- + ADD/SUB instruction에서 3번째 bit에서 Carry/Borrow가 생기면 set, BCD연산에 사용

OV (Overflow Flag)

- + 부호있는 ADD/SUB 에서 2의 보수 수치(-128~+127)을 넘으면 set
- + Multiply instruction에서 결과가 FFH를 넘으면 set
- + Divide instruction에서 제수가 0이면 set

P (Parity Flag)

- + Accumulator의 내용에 따라서 변경되며 , Accumulator의 내용 중 1의 개수가 홀수이면 set



■ RS1, RS0 (Register Bank Select)

RS ₁	RS ₀	선택된 बैं크
0	0	Bank 0 (리셋 후 초기값)
0	1	Bank 1
1	0	Bank 2
1	1	Bank 3

[Example] MOV PSW, #00001000B
 MOV R1, A
 MOV PSW, #18H
 MOV R1, A

F0 (General Purpose Flag)

+ User의 needs에 맞게 software로 set/reset 가능



■ ADD

- + ADD A, Rn
 - Add register to accumulator
- + ADD A, direct
 - Add direct byte to accumulator
- + ADD A, @Ri
 - Add indirect RAM to accumulator
- + ADD A, #data
 - Add immediate data to accumulator

Example : ADD A, #32H

	Acc	CY	AC	OV	P
Before	76	–	–	–	X
After	A8	0	0	1	1

Example : ADD A, @R1

	Acc	@R1	CY	AC	OV	P
Before	86	62	–	–	–	X
After	E8	62	0	0	0	0

ADDC

- + ADDC A, Rn
 - Add register to accumulator with carry flag
- + ADDC A, direct
 - Add direct byte to A with carry flag
- + ADDC A, @Ri
 - Add indirect RAM to A with carry flag
- + ADDC A, #data
 - Add immediate data to A with carry flag

Example : ADDC A, 50H

	Acc	(50H)	CY	AC	OV	P
Before	55	4E	1	–	–	–
After	A4	4E	0	1	1	1

Example : ADDC A, R5

	Acc	R5	CY	AC	OV	P
Before	E9	58	0	–	–	–
After	41	58	1	1	0	0

SUBB

- + SUBB A, Rn
 - Subtract register from A with borrow
- + SUBB A, direct
 - Subtract direct byte from A with borrow
- + SUBB A, @Ri
 - Subtract indirect byte from A with borrow
- + SUBB A, #data
 - Subtract immediate data from A with borrow

Example : SUBB A, @R0

	Acc	@R0	CY	AC	OV	P
Before	53	37	1	–	–	–
After	1B	37	0	1	0	0

Example : SUBB A, #35H

	Acc	@R0	CY	AC	OV	P
Before	22	35	0	–	–	–
After	ED	35	1	1	0	0

INC

- + INC A (A <- A+1)
 - Increment accumulator
- + INC Rn (Rn <- Rn+1)
 - Increment register
- + INC Direct (A <- A+1)
 - Increment direct byte
- + INC @Ri (Ri <- Ri+1)
 - Increment Indirect RAM
- + INC DPTR (DPTR <- DPTR+1)
 - Increment data pointer

Example : INC A

	Acc	CY	AC	OV	P
Before	FF	–	–	–	X
After	00	–	–	–	0

Example : INC 30H

	(30H)	CY	AC	OV	P
Before	7F				
After	80	–	–	–	–



DEC

- + DEC A
 - Decrement accumulator
- + DEC Rn
 - Decrement register
- + DEC Direct
 - Decrement direct RAM
- + DEC @Ri
 - Decrement indirect pointer

Example : DEC A

	Acc	CY	AC	OV	P
Before	00	–	–	–	X
After	FF	–	–	–	0

Example : DEC @R0

	@R0	CY	AC	OV	P
Before	70				
After	6F	–	–	–	–

DA

- + DA A
 - Decimal adjust accumulator

Example : BCD 숫자 12+19인 경우

1. 결과가 9보다 큰 A~F인 경우 DA A 명령시 결과 +6을 함
2. AC = 1 이면 Accumulator + 06H
3. CY = 1 이면 Accumulator + 60H

Example : ADD A, R0 DA A,

ADD A,R0	A	11	19	91	99	09
Before	R0	22	18	81	88	05
ADD A,R0	A	33	31	12	21	0E
After	AC	0	1	0	1	0
	CY	0	0	1	1	0
DA A	A	33	37	72	87	14
After	AC	0	1	0	1	0
	CY	0	0	1	1	0

MUL/DIV instruction

- + 8bit data 연산을 수행하여 결과는 16bit data가 됨
- + 계산 결과가 FFH이상이면 OV가 set
- + 제수 일 때 OV set

MUL

- + MUL AB
 - Multiply A and B

```
MUL  AB    ; A × B = B A
DIV  AB    ; A ÷ B = A(몫)... B(나머지)
```

Example : MOV A, #31H
MOV B, #10H
MUL AB

B	Acc	OV
03	10	1

DIV

- + DIV AB
 - Divide A by B

Example : MOV A, #118
MOV B, #5
DIV AB

B	Acc	OV
3	23	0



■ ANL

- + ANL Rn
 - AND register to accumulator
- + ANL Direct
 - AND direct byte to accumulator
- + ANL @Ri
 - AND indirect RAM to accumulator
- + ANL A,#data
 - AND immediate data to accumulator
- + ANL direct, A
 - AND accumulator to direct byte
- + ANL direct, #data
 - AND immediate data to direct byte

Example : ANL A, 00001000B

	Acc	P
Before	01110111B 00001000B	X
After	00000000B	0

Example : ANL 37H, 11110000B

Before	(37H)	01110111B 11110000B
After	(37H)	01110000B



■ ORL

- + ORL Rn
 - OR register to accumulator
- + ORL Direct
 - OR direct byte to accumulator
- + ORL @Ri
 - OR indirect RAM to accumulator
- + ORL A,#data
 - OR immediate data to accumulator
- + ORL direct, A
 - OR accumulator to direct byte
- + ORL direct, #data
 - OR immediate data to direct byte

Example : ORL A, R4

Before	Acc	10010001B	P
	R4	10001000B	X
After	Acc	10011001B	0

Example : ORL 37H, A

Before	(37H)	10000000B
	Acc	01110111B
After	(37H)	11110111B



■ XRL

- + XRL Rn
 - Exclusive OR register to accumulator
- + XRL Direct
 - Exclusive OR direct byte to accumulator
- + XRL @Ri
 - Exclusive OR indirect RAM to accumulator
- + XRL A,#data
 - Exclusive OR immediate data to accumulator
- + XRL direct, A
 - Exclusive OR accumulator to direct byte
- + XRL direct, #data
 - Exclusive OR immediate data to direct byte

Example : XRL A, @R0

Before	Acc	10010001B	P
	@R0	11100011B	X
After	Acc	01110010B	0

CLR

+ CLR A

- Clear accumulator

CPL

+ CPL A

- Complement accumulator

Rotate

+ RL A

- Rotate accumulator left

+ RLC A

- Rotate accumulator left through carry

+ RR A

- Rotate accumulator right

+ RRC A

- Rotate accumulator right through carry

SWAP

+ SWAP A

- Swap nibbles within the accumulator

Example : CLR A

	Acc
Before	7FH
After	00H

Example : CPL A

	Acc
Before	10110101B
After	01001010B

Example : RL A

Before	#11010000B
After	#10100001B

Example : RR A

	CY	Acc	P
Before	1	00011001B	X
After	0	00110011B	0

Example : SWAP A

	Acc
Before	37H
After	73H



CLR

- + CLR C
 - Clear carry flag
- + CLR bit
 - Clear direct bit

Example : CLR 28H.0

Before	(28H)	0 0 0 0 0 1 0 1
After	(28H)	0 0 0 0 0 1 0 0

Example : CLR C

	CY
Before	1
After	0

SETB

- + SETB C
 - Set carry bit
- + SETB bit
 - Set direct bit

Example : SETB 21H.5

Before	(21H)	0 1 0 0 0 1 1 0
After	(21H)	0 1 1 0 0 1 1 0

CPL

- + CPL A
 - Complement carry flag
- + CPL bit
 - Complement direct bit

Example : CPL 23H.7

Before	(23H)	1 0 1 0 0 1 0 0
After	(23H)	0 0 1 0 0 1 0 0

ANL

- + CLR C
 - Clear carry flag
- + CLR bit
 - Clear direct bit

Example : ANL C, 27H, 3

Before	CY	1
	(27H)	0 0 1 0 1 1 1 0
After	CY	1

Example : ANL C, /20H.5

Before	CY	1
	(20H)	0 1 0 1 1 0 0 0
After	CY	1

ORL

- + ORL C
 - OR direct bit to carry flag
- + ORL bit
 - OR complement of direct bit to carry

Example : ORL C, Acc.7

Before	CY	1
	Acc	0 0 1 0 0 1 1 0
After	CY	1

MOV

- + MOV C
 - Move direct bit to carry flag
- + MOV bit
 - Move carry flag to direct bit

Example : MOV C, 20H, 5

Before	CY	0
	(20H)	1 0 1 0 1 1 0 0
After	CY	1

Example : MOV 30H, C

	CY=1	
Before	(30H) = 0	
After	(30H) = 1	

Call

- + ACALL addr11
 - Absolute subroutine call, 2K byte 이내의 subroutine 만을 call
- + LCALL addr16
 - Long subroutine call, 0~65535 번지

Example : ORG 0000H
 MOV R0, #00H
 CLR A
 MOV DPTR, #2100H
 LCALL EX_CRL

EX_CLR : MOVX @DPTR, A
 INC DPTR
 DJNZ R0, EX_CLR
 RET

Return

- + RET
 - Return from subroutine
- + RETI
 - Return from interrupt

Jump

- + AJMP addr11
 - Absolute jump
- + LJMP addr16
 - Long jump
- + SJMP rel
 - Short jump (relative addr)
- + JMP @A + DPTR
 - Jump indirect relative to the DPTR

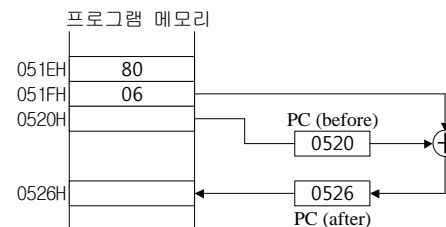
Example : MOV C,P1,0
 MOV P1,1,C
 AJMP NEAR
 ~

NEAR :

Example : MOV C,P1,0
 MOV P1,1,C
 LJMP FAR
 ~

FAR :

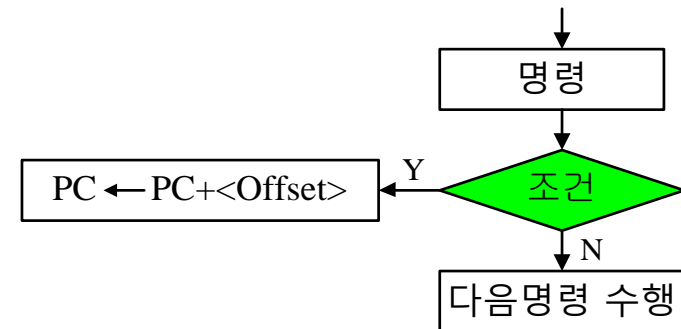
Example : SJMP 06H



Example : ORG 0300H
 SJMP BOT
 ~
 BOT : INC A

■ Jump (Conditional Jump)

- + JZ rel
 - Jump if accumulator is zero
- + JNZ rel
 - Jump if accumulator is not zero
- + JC rel
 - Jump if carry flag is zero
- + JNC rel
 - Jump if carry flag is not zero
- + JB bit, rel
 - Jump if indirect bit is zero
- + JNB bit, rel
 - Jump if indirect bit is not zero
- + JBC bit, rel
 - Jump if direct bit is set and clear bit

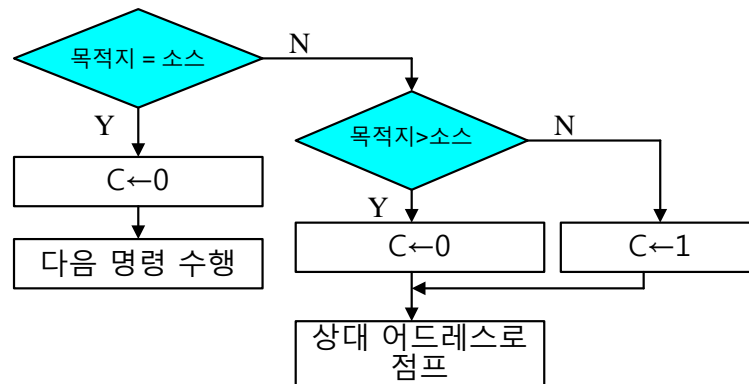


```
Example : ORG 1000H
           MOV A, #80H
           MOV R5, #20H
           ADD A, R5
           JNC NO ; CY = 0이므로 NO로
jump
           ~~~
           ORG 1030H
           ~~~
NO :
```

■ Jump

+ CJNE

- CJNE A, direct, rel = Compare direct byte to A and jump if not equal
- CJNE A, #data, rel = Compare immediate byte to A and jump if not equal
- CJNE Rn, #data, rel = Compare immediate to reg. and jump if not equal
- CJNE @Ri, #data, rel = Compare immediate to indirect and jump if not equal



Example : CJNE @R1, #01, SCAB

Example : CJNE A, #10H, NEXT

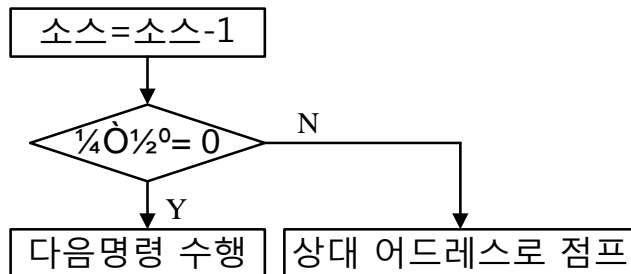
Example : CJNE A, 37H, TEST

Example : CJNE R5, #32H, SKIP

■ Jump

+ DJNZ

- DJNZ Rn, rel = Decrement register and jump if not zero
- DJNZ direct, rel = Decrement direct byte and jump if not zero



Example : MOV R7, #2

LOOP: ADD A, R7

~~~

DJNZ R7, LOOP1

INC A

### + NOP

- No operation and PC + 1, use delay routine

# THANK YOU