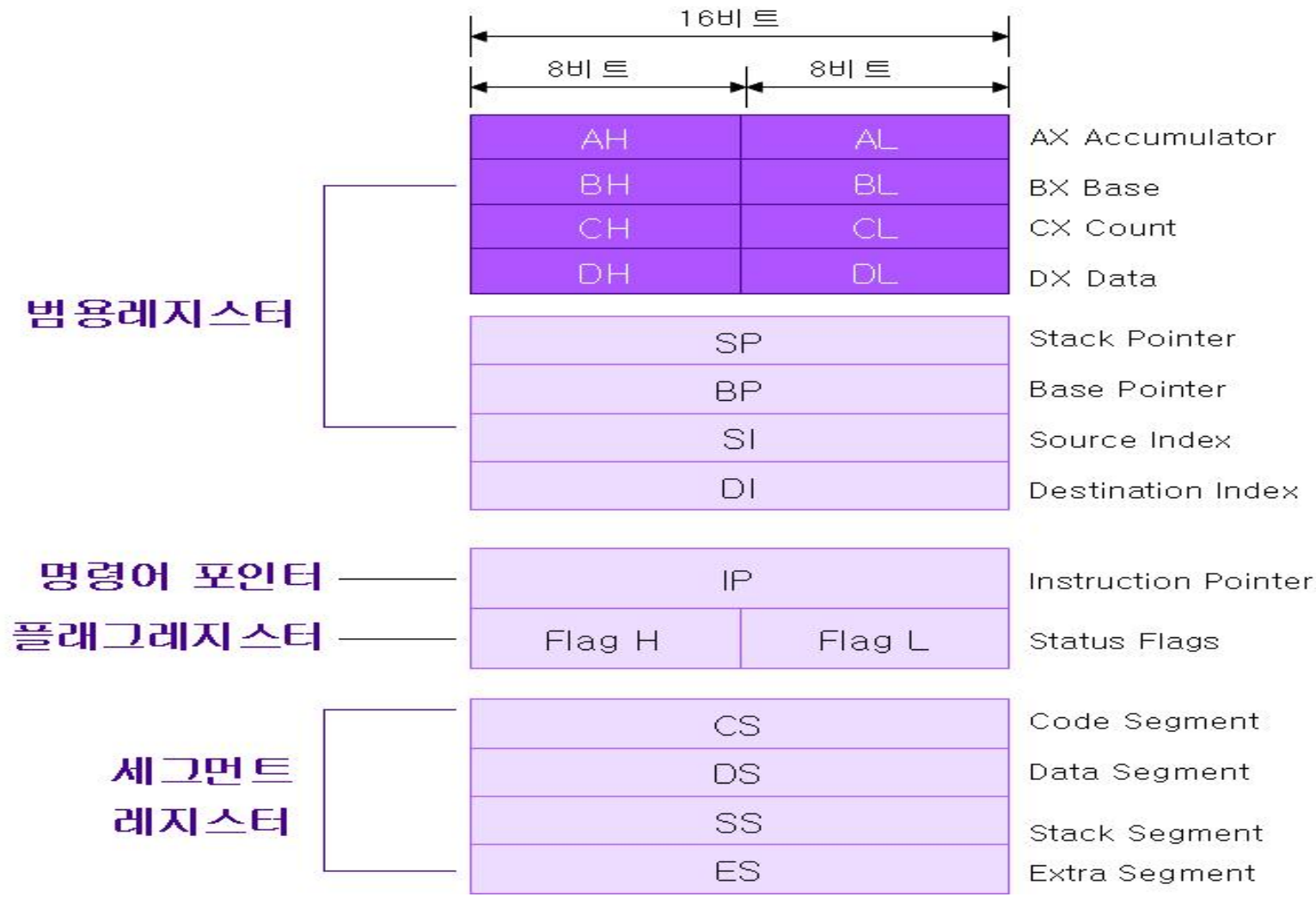
제 4 장

어셈블리어 3

학습내용

* 프로그래밍 언어
* 어셈블리어 개요
* 어셈블리 프로그래밍

레지스터 구성도


16비트
8비트
8비트
AH
AL
BH
BL
CH
CL
DH
DL
AX Accumulator
BX Base
CX Count
DX Data
범용레지스터
SP
BP
SI
DI
Stack Pointer
Base Pointer
Source Index
Destination Index
명령어 포인터
IP
Instruction Pointer
플래그레지스터
Flag H
Flag L
Status Flags
세그먼트
레지스터
CS
DS
SS
ES
Code Segment
Data Segment
Stack Segment
Extra Segment

범용 레지스터의 특별한 사용

- **AX(AL, AH):**산술 논리 연산항과 결과에 대한 저장, **Accumulator**
- **BX(BL, BH):** 간접주소 지정시 베이스주소를 가리키는 레지스터로 사용,**Base**
- **CX(CL, CH):** 문자열과 반복문(루프)에서 반복 횟수를 카운트하는 레지스터로 사용, **Counter**
- **DX(DL, DH) :** 입출력에 대한 포인터 , 입출력 주소 저장에 사용되며 상위 워드용 데이터 레지스터로 사용되는 보조 어큐물레이터, **Data**
- **SI :** 문자열 연산에 대한 출발 항 또는 DS 레지스터가 가리키고 있는
- 세그먼트의 데이터에 대한 인덱스 포인터, **Source index**
- **DI :** 문자열 연산에 대한 목적 항 또는 ES 레지스터가 가리키고 있는
- 세그먼트의 데이터에 대한 인덱스 포인터, **Destination index**
- **SP :** 스택 포인터 , 스택 조작을 위해서 사용하는 레지스터로 프로그램 실행 중에 데이터의 저장 주소를 기억하고 있는 레지스터, **Stack Pointer**
- **BP :** 스택에 있는 데이터에 대한 포인터 , 스택 세그먼트 SS영역 내에 배치한 데이터에 대한 베이스 주소, **Base Pointer**

주소 지정 방식

주소지정 방식

1) 개념

- 데이터들은 명령어의 연산항이 가리키는 곳에 저장됨
- 명령어 연산항이 데이터의 주소를 지정

  8086 데이터 전송의 예)

- MOV DX, CX
- DX : 목적항, CX : 출발항
- 오퍼랜드에서 어느 데이터 또는 어디에 데이터를 저장하느냐가 중요함
- 데이터 표기법을 주소 지정 방식이라 함

2) 주소 지정 방식(8086 어셈블러의 종류)

(1) 값 즉시 지정

• 연산항에 레지스터나 기억장소의 주소가 아닌 8 또는 16비트 값 직접 이용

(2) 직접 주소 지정

• 대상이 되는 레지스터 또는 기억장소의 주소가 그대로 연산항 으로 이용

지정되는 레지스터의 내용 또는 지정되는 기억 장소 번지의 내용 자체가 처리대상

① 레지스터 직접 : 레지스터를 이용하여 연산 (가장 빠르다)

② 메모리 직접 지정 방법 : 지정되는 기억 장소 주소를 직접 지정

(3) 간접 주소 지정

• 레지스터가 가리키는 주소에 있는 값 간접이용 ([ ] 기호 사용)

① 레지스터 간접주소지정 : 변위는 포함하지 않고 베이스레지스터(BR:BX)나 인덱스레지스터(INX:DI, SI) 이용

② 베이스 주소지정 : 변위값에 베이스레지스터 값을 더해서 실효번지 구함

③ 인덱스 주소지정 : 변위값에 인덱스레지스터 값을 더해서 실효번지 구함

④ 베이스 인덱스 주소지정 : 변위값에 BR, INX 값을 더해서 실효번지 구함.

구조체의 구조를 액세스 하는데 유용.

주소 지정 방식

값 즉시 지정방식

값 즉시 지정 방식

(1) 개념 : 상수를 연산항으로 사용

(2) 방식 : MOV AX, 3004H

[예제4-4 : a_i_1.asm] 값 즉시 지정 방식으로 1010H +203CH을 더하여 변수 RESULT에 저장하시오.

1010H + 203CH의 예

```
1 ;값 즉시 지정방식의 예
2   MAIN   SEGMENT
3              ASSUME       CS:MAIN, DS:MAIN
4              MOV          AX, CS
5              MOV          DS, AX
6              MOV          AX, 1010H
7              ADD          AX, 203CH
8              MOV          RESULT, AX
9              MOV          AH, 4CH
10             INT           21H
11  RESULT      DW      ?
12  MAIN  ENDS
13             END
```

3 0 4 C

[예제4-4 : a_i_1.asm] 값 즉시 지정 방식으로 1010H+203CH을 더하여 변수 RESULT에 저장하시오.

=> 실행 전과 후의 기억장소 상태

실행전의 기억장소 상태

0A5C : 0000	MOV AX, CS
0A5C : 0002	MOV DS, AX
0A5C : 0004	MOV AX, 1010
0A5C : 0007	ADD AX, 203C
0A5C : 000A	MOV [0011] , AX
0A5C : 000D	MOV AH, 4C
0A5C : 000F	INT 21
0A5C : 0011	00 | 00
0A5C : 0013	

실행후의 기억장소 상태

0A5C : 0000	MOV AX, CS
0A5C : 0002	MOV DS, AX
0A5C : 0004	MOV AX, 1010
0A5C : 0007	ADD AX, 203C
0A5C : 000A	MOV [0011] , AX
0A5C : 000D	MOV AH, 4C
0A5C : 000F	INT 21
0A5C : 0011	4C (11번지) | 30 (12번지)
0A5C : 0013	

[예제4-4 : a_i_1.asm] 값 즉시 지정 방식으로 1010H+203CH을 더하여 변수 RESULT에 저장하시오.

=> 실행 전과 후의 기억장소 상태

```
C:\8086>debug a_i_1.exe
-u
076A:0000 8CC8          MOV     AX,CS
076A:0002 8ED8          MOV     DS,AX
076A:0004 B81010        MOV     AX,1010
076A:0007 053C20        ADD     AX,203C
076A:000A A31100        MOV     [0011],AX
076A:000D B44C          MOV     AH,4C
076A:000F CD21          INT     21
076A:0011 0000          ADD     [BX+SI],AL
076A:0013 50            PUSH    AX
076A:0014 8B46FC        MOV     AX,[BP-04]
076A:0017 8B56FE        MOV     DX,[BP-02]
076A:001A 050C00        ADD     AX,000C
076A:001D 52            PUSH    DX
076A:001E 50            PUSH    AX
076A:001F E80E49        CALL    4930
-g=0,15

Program terminated normally
```

```
-g=0,15

Program terminated normally
-u0
076A:0000 8CC8          MOV     AX,CS
076A:0002 8ED8          MOV     DS,AX
076A:0004 B81010        MOV     AX,1010
076A:0007 053C20        ADD     AX,203C
076A:000A A31100        MOV     [0011],AX
076A:000D B44C          MOV     AH,4C
076A:000F CD21          INT     21
076A:0011 4C            DEC     SP
076A:0012 30508B        XOR     [BX+SI-75],DL
076A:0015 CC            INT     3
076A:0016 FC            CLD
076A:0017 8B56FE        MOV     DX,[BP-02]
076A:001A 050C00        ADD     AX,000C
076A:001D 52            PUSH    DX
076A:001E 50            PUSH    AX
076A:001F E80E49        CALL    4930
-
```

주소 지정 방식

직접 주소 지정방식

주소지정 방식

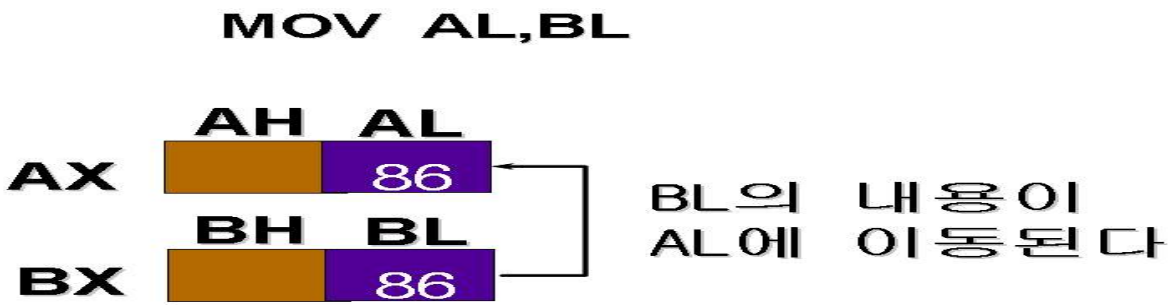
1) 직접 주소 지정 방식

(1) 개념

- 대상이 되는 레지스터나 기억장소의 번지를 그대로 연산항으로 사용하는 방식

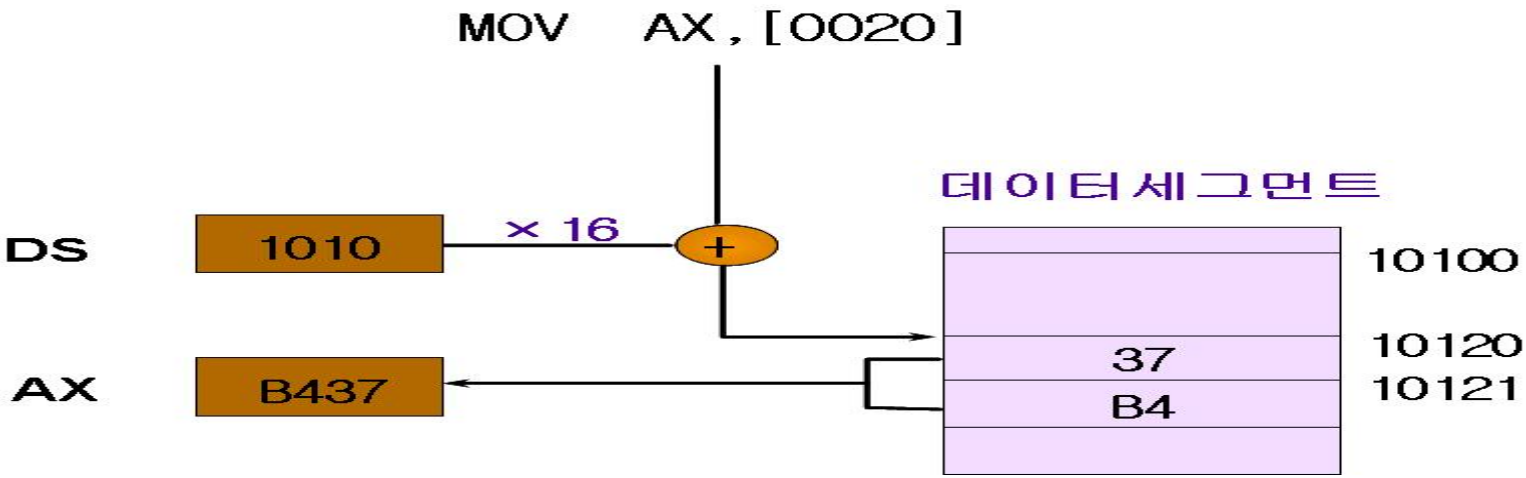
- 레지스터 직접 지정 방식과 메모리 직접 지정 방식

(2) 레지스터 직접 지정 방식



- 연산항을 8086의 레지스터를 이용하여 연산함
- 지정에 필요한 비트수가 적음
- 연산항의 접근이 주소지정 방식 중 가장 빠름

(3) 메모리 직접 지정 방식



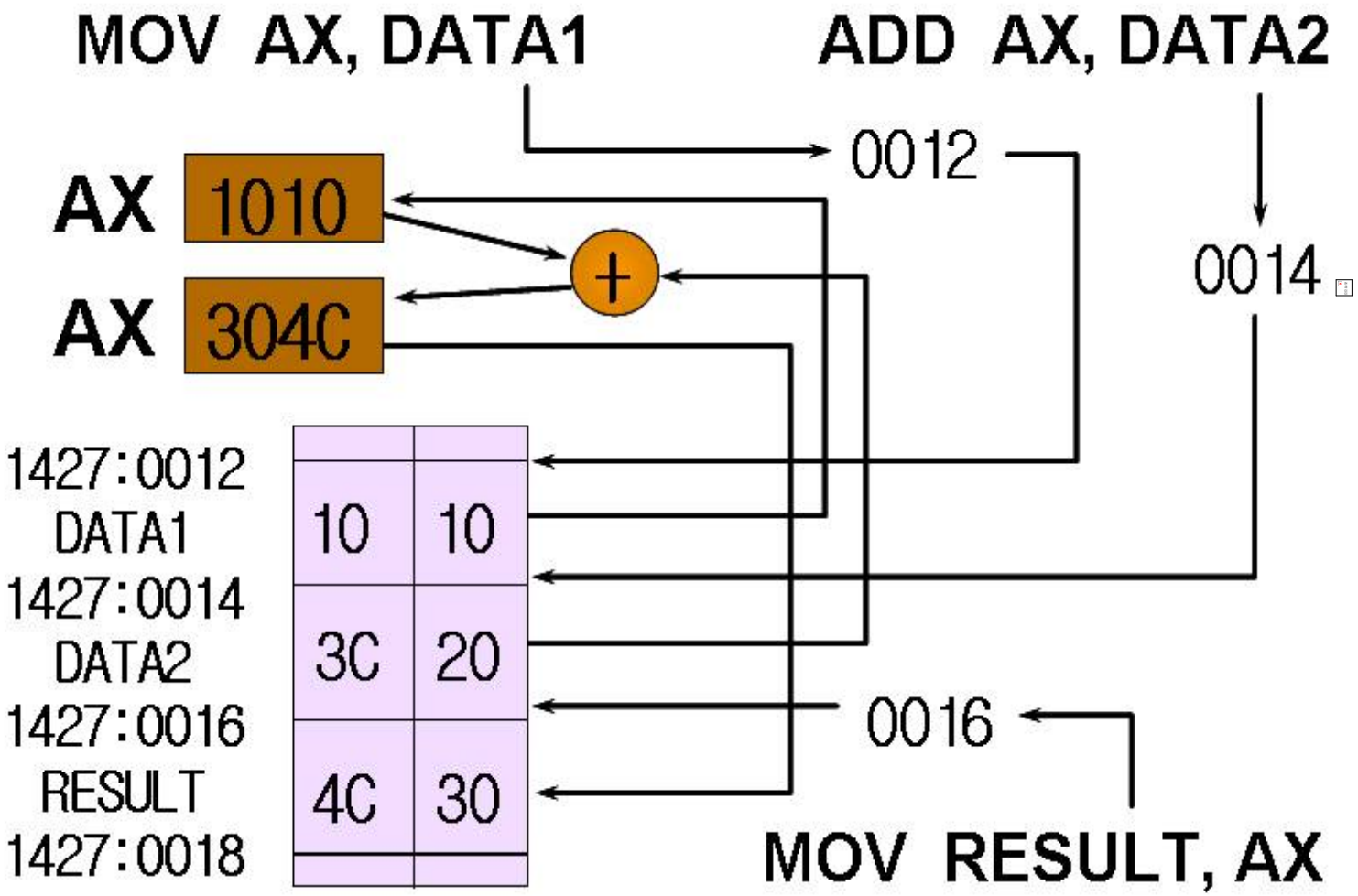
- 레지스터와 상관없이 주소를 직접 지정
- 명령 바이트 수는 길지만 세그먼트 내의 어떤 주소도 접근이 가능
- 연산 항으로 직접 주소 값을 가지며, 처리되는 대상은 그 값이 가리키는 곳의 내용
- 두 개의 연산 항 모두 메모리 직접 주소 지정 방식을 취하면 안되고 반드시 메모리 값 중에 하나는 레지스터에 넣고 수행해야 한다.
- 간단한 레이블(변수) 접근을 위해서도 이용

[예제4-5 : a_d_2.asm] 직접 주소 지정 방식으로 DATA1과 DATA2에 저장된 데이터를 더하여 변수 RESULT에 저장하시오.

```
1 ; 직접 주소지정방식 예제
2  MAIN     SEGMENT
3            ASSUME CS:MAIN, DS:MAIN
4            MOV      AX, CS        ; 레지스터 직접 주소 지정 방식
5            MOV      DS, AX
6            MOV      AX, DATA1 ;메모리 직접 주소 지정 방식
7            ADD      AX, DATA2
8            MOV      RESULT, AX
9            MOV      AH, 4CH
10           INT      21H
11 DATA1   DW       1010H
12 DATA2   DW       203CH
13 RESULT  DW        ?
14 MAIN     ENDS
15 END
```

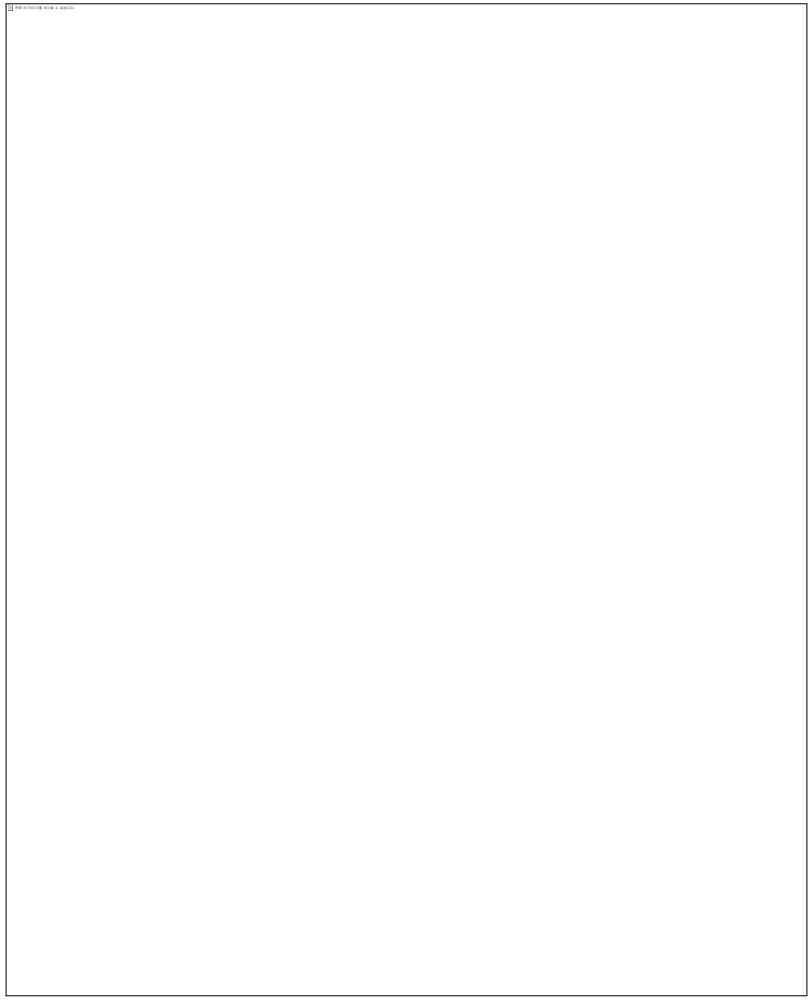
[예제4-5 : a_d_2.asm] 직접 주소 지정 방식으로 DATA1과 DATA2에 저장된 데이터를 더하여 변수 RESULT에 저장하시오.

② 실제 동작 과정

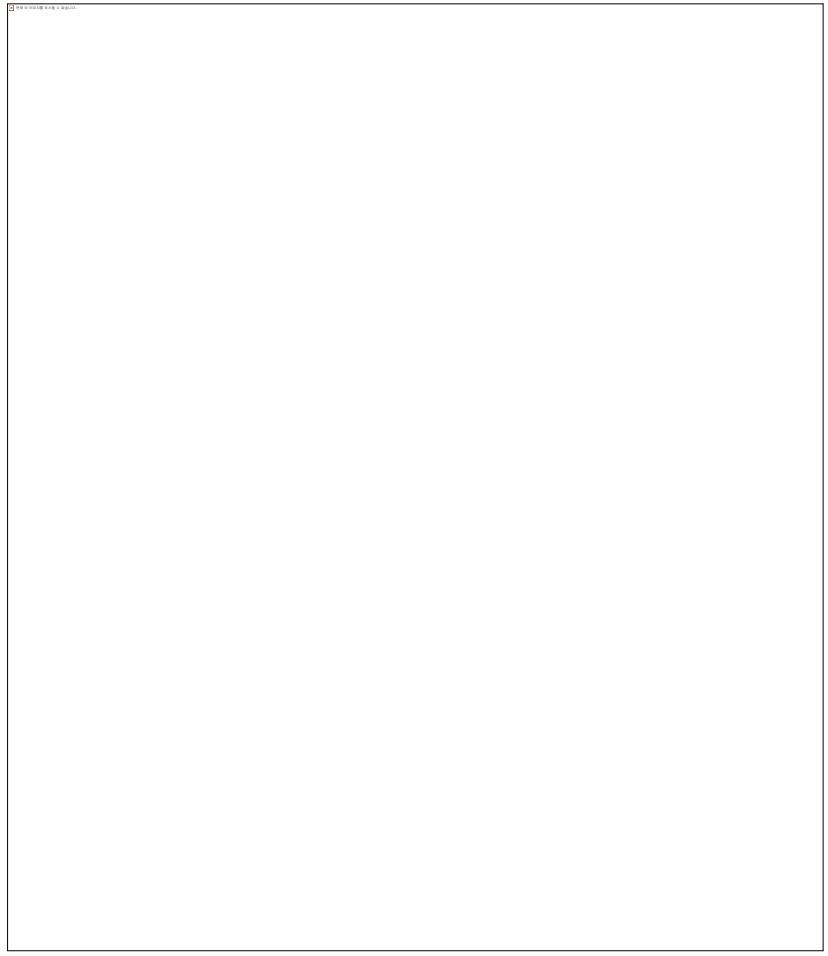


[예제4-5 : a_d_2.asm] 직접 주소 지정 방식으로 DATA1과 DATA2에 저장된 데이터를 더하여 변수 RESULT에 저장하시오.

③ 실행 전 기억장소 상태



④ 실행 후 기억장소 상태



[예제4-5 : a_d_2.asm] 직접 주소 지정 방식으로 DATA1과 DATA2에 저장된 데이터를 더하여 변수 RESULT에 저장하시오.

③ 실행 전 기억장소 상태

```
C:\8086>debug a_d_2.exe
-u0
076A:0000 8CC8          MOV     AX,CS
076A:0002 8ED8          MOV     DS,AX
076A:0004 A11200        MOV     AX,[0012]
076A:0007 03061400      ADD     AX,[0014]
076A:000B A31600        MOV     [0016],AX
076A:000E B44C          MOV     AH,4C
076A:0010 CD21          INT     21
076A:0012 1010          ADC     [BX+SI],DL
076A:0014 3C20          CMP     AL,20
076A:0016 0000          ADD     [BX+SI],AL
076A:0018 56            PUSH    SI
076A:0019 FE05          INC     BYTE PTR [DI]
076A:001B 0C00          OR      AL,00
076A:001D 52            PUSH    DX
076A:001E 50            PUSH    AX
076A:001F E80E49        CALL    4930
```

④ 실행 후 기억장소 상태

```
-g

Program terminated normally
-u0
076A:0000 8CC8          MOV     AX,CS
076A:0002 8ED8          MOV     DS,AX
076A:0004 A11200        MOV     AX,[0012]
076A:0007 03061400      ADD     AX,[0014]
076A:000B A31600        MOV     [0016],AX
076A:000E B44C          MOV     AH,4C
076A:0010 CD21          INT     21
076A:0012 1010          ADC     [BX+SI],DL
076A:0014 3C20          CMP     AL,20
076A:0016 4C            DEC     SP
076A:0017 3056FE        XOR     [BP-02],DL
076A:001A 050C00        ADD     AX,000C
076A:001D 52            PUSH    DX
076A:001E 50            PUSH    AX
076A:001F E80E49        CALL    4930
```

주소 지정 방식

간접지정방식

주소 지정 방식

1) 간접 주소 지정 방식

(1) 개념

- 기억장치 액세스 제어는 BIU에 의해 이루어짐
- IP 레지스터와 코드 세그먼트로부터 명령코드의 실제주소를 계산 하여 그 명령어 코드를 읽음
- 물리주소는 EU속에서 계산되는 오프셋과 세그먼트 시작주소의 합으로 계산
- 이 오프셋을 유효주소(EA)라 함

(2) 유효주소 계산 방법

유효주소(EA) = BR + INX + D

- BR: 베이스 레지스터(BX, BP)
- INX: 인덱스 레지스터(SI, DI)
- D: 변위(8비트 또는 16비트 상수 값)

– 레지스터 간접지정: BR

– 베이스 주소 지정: BR, D

– 인덱스 주소 지정: INX, D

– 베이스 인덱스 주소 지정: BR, INX, D

2) 레지스터 간접 주소지정 방식(BR: 베이스 레지스터(BX, BP))

- 명령어 속에 변위를 포함하지 않고 베이스 레지스터와 인덱스 레지스터의
- 레지스터는 [ ] 기호를 사용하여 나타낸다.

[예제4-6 : a_idr_31.asm] 레지스터 간접 주소 지정 방식으로 DATA1과 DATA2에 저장된 데이터를 더하여 변수 RESULT에 저장하시오.

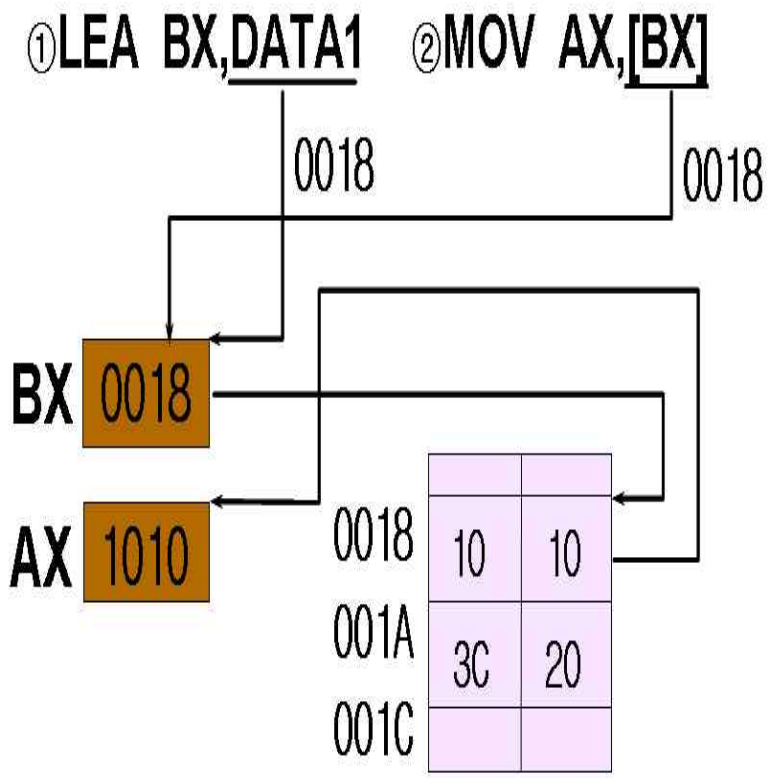
(1) 프로그램

LEA : 주소를 적재

DATA1 + DATA2의 계산

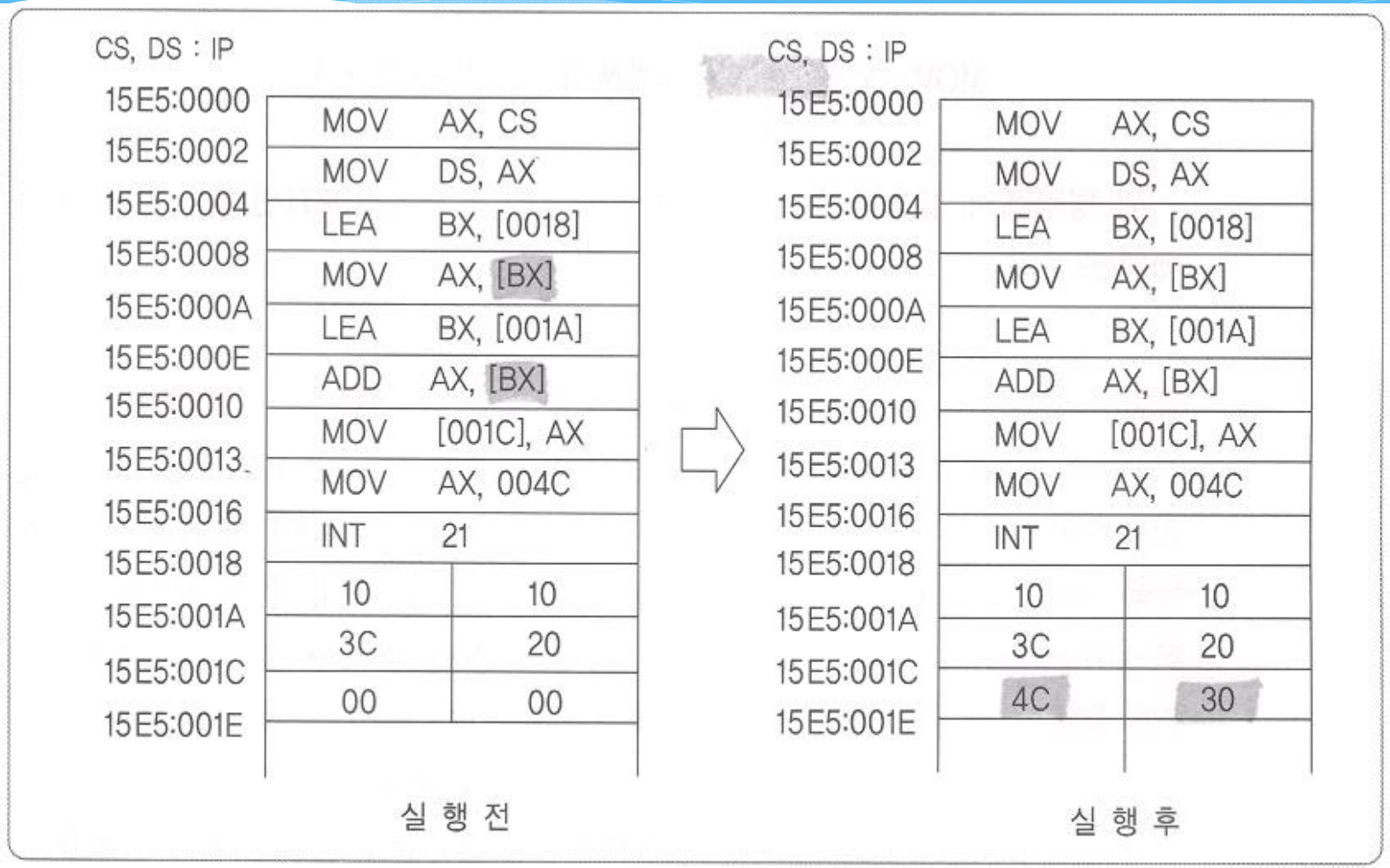
```
1 ;레지스터 간접 주소지정 예
2    MAIN      SEGMENT
3              ASSUME      CS:MAIN, DS:MAIN
4              MOV          AX, CS
5              MOV          DS, AX
6              LEA          BX, DATA1
7              MOV          AX, [BX]
8              LEA          BX, DATA2
9              ADD          AX, [BX]
10             MOV          RESULT, AX
11             MOV          AX, 4CH
12             INT          21H
13   DATA1     DW           1010H
14   DATA2     DW           203CH
15   RESULT    DW             ?
16   MAIN      ENDS
17             END
```

(2) 주소지정 방법



[예제4-6 : a_idr_31.asm] 레지스터 간접 주소 지정 방식으로 DATA1과 DATA2에 저장된 데이터를 더하여 변수 RESULT에 저장하시오.

(3) 실행 전/후의 기억장소 상태



[예제4-6 : a_idr_31.asm] 레지스터 간접 주소 지정 방식으로 DATA1과 DATA2에 저장된 데이터를 더하여 변수 RESULT에 저장하시오.

(3) 실행 전/후의 기억장소 상태

```
C:\8086>debug a_idr_31.exe
-u
076A:0000 8CC8          MOV     AX,CS
076A:0002 8ED8          MOV     DS,AX
076A:0004 8D1E1700      LEA     BX,[0017]
076A:0008 8B07          MOV     AX,[BX]
076A:000A 8D1E1900      LEA     BX,[0019]
076A:000E 0307          ADD     AX,[BX]
076A:0010 A31B00        MOV     [001B],AX
076A:0013 B44C          MOV     AH,4C
076A:0015 CD21          INT     21
076A:0017 1010          ADC     [BX+SI],DL
076A:0019 3C20          CMP     AL,20
076A:001B 0000          ADD     [BX+SI],AL
076A:001D 52            PUSH    DX
076A:001E 50            PUSH    AX
076A:001F E80E49        CALL    4930
```

```
-g

Program terminated normally
-u0
076A:0000 8CC8          MOV     AX,CS
076A:0002 8ED8          MOV     DS,AX
076A:0004 8D1E1700      LEA     BX,[0017]
076A:0008 8B07          MOV     AX,[BX]
076A:000A 8D1E1900      LEA     BX,[0019]
076A:000E 0307          ADD     AX,[BX]
076A:0010 A31B00        MOV     [001B],AX
076A:0013 B44C          MOV     AH,4C
076A:0015 CD21          INT     21
076A:0017 1010          ADC     [BX+SI],DL
076A:0019 3C20          CMP     AL,20
076A:001B 4C            DEC     SP
076A:001C 305250        XOR     [BP+SI+50],DL
076A:001F E80E49        CALL    4930
```

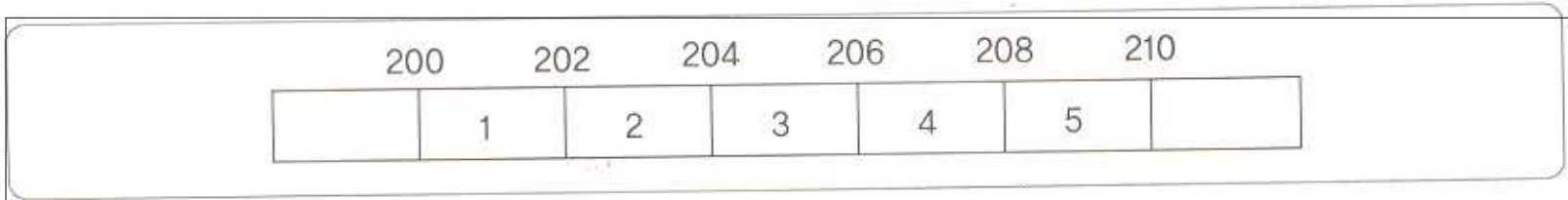
3) 베이스 주소지정 방식

(1) 개념

- 지정된 베이스 레지스터의 값과 변위 값을 합하여 유효주소를 계산

  [BR + D]

- [레코드의 시작위치 + 특정 필드까지의 거리]



▶▶▶ 데이터가 워드형으로 저장된 상태

BX:0200

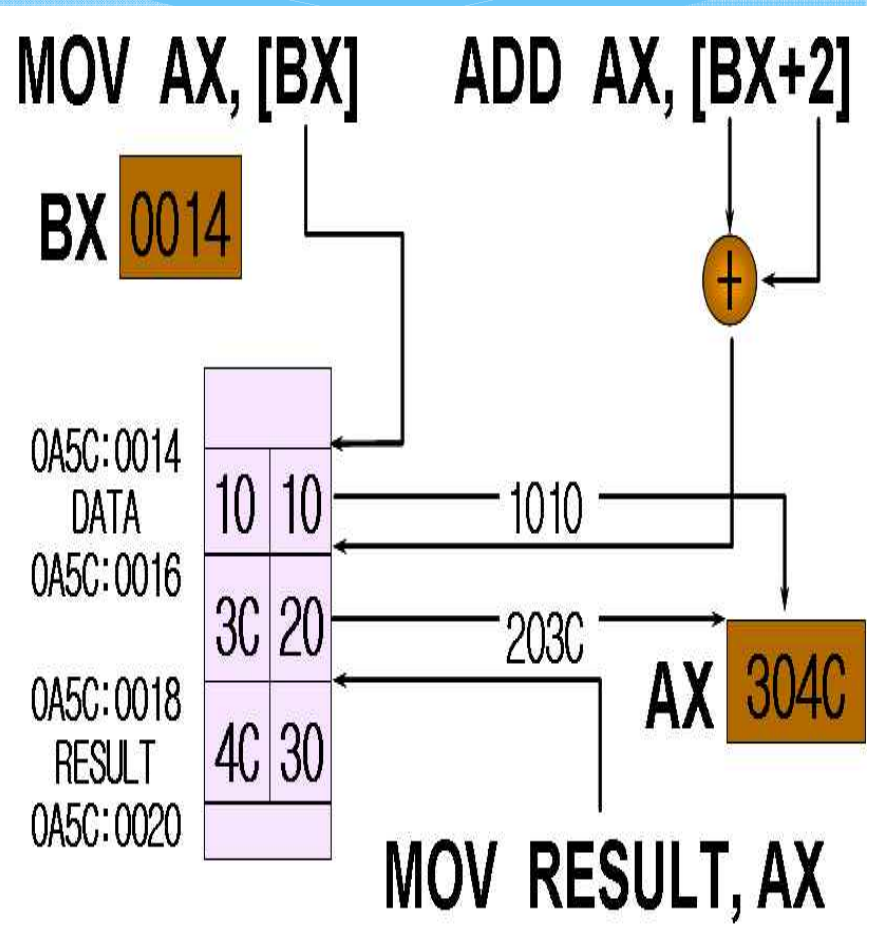
MOV AX, [BX + 6] ; 4번째 데이터를 AX에 저장

위 명령에서 실효번지는 BX + 6 = 200H + 6 = 206H번지가 된다.

[예제4-7 : a_idB_32.asm] 베이스 주소 지정 방식으로 DATA에 저장된 두 개의 데이터를 더하여 변수 RESULT에 저장하시오.

```
1  ;베이스 주소지정의 예
2     MAIN    SEGMENT
3     ASSUME     CS:MAIN, DS:MAIN
4                MOV          AX, CS
5                MOV          DS, AX
6   ;
7                LEA          BX, DATA
8                MOV          AX, [BX]
9                ADD          AX, [BX+2]
10               MOV          RESULT, AX
11               MOV          AH, 4CH
12               INT          21H
13 ;
14   DATA        DW       1010H, 203CH
15   RESULT      DW            ?
16   MAIN         ENDS
17                END
```

(3) 주소지정 방법



[예제4-7 : a_idB_32.asm] 베이스 주소 지정 방식으로 DATA에 저장된 두 개의 데이터를 더하여 변수 RESULT에 저장하시오.

(4) 실행 전/후의 기억장소 상태

0A5C :	0000	MOV	AX, CS
0A5C :	0002	MOV	DS, AX
0A5C :	0004	LEA	BX, [0014]
0A5C :	0008	MOV	AX, [BX]
0A5C :	000A	ADD	AX, [BX+02]
0A5C :	000D	MOV	[0018], AX
0A5C :	0010	MOV	AH, 4C
0A5C :	0012	INT	21
0A5C :	0014	10	10
0A5C :	0016	3C	20
0A5C :	0018	00	00
0A5C :	0020		

실행 전

0A5C :	0000	MOV	AX, CS
0A5C :	0002	MOV	DS, AX
0A5C :	0004	LEA	BX, [0014]
0A5C :	0008	MOV	AX, [BX]
0A5C :	000A	ADD	AX,[BX+02]
0A5C :	000D	MOV	[0018], AX
0A5C :	0010	MOV	AH, 4C
0A5C :	0012	INT	21
0A5C :	0014	10	10
0A5C :	0016	3C	20
0A5C :	0018	4C	30
0A5C :	0020		

실행 후

[예제4-7 : a_idB_32.asm] 베이스 주소 지정 방식으로 DATA에 저장된 두 개의 데이터를 더하여 변수 RESULT에 저장하시오.

(4) 실행 전/후의 기억장소 상태

```
C:\8086>debug a_idB_32.exe
-u
076A:0000 8CC8          MOV     AX,CS
076A:0002 8ED8          MOV     DS,AX
076A:0004 8D1E1400      LEA     BX,[0014]
076A:0008 8B07          MOV     AX,[BX]
076A:000A 034702        ADD     AX,[BX+02]
076A:000D A31800        MOV     [0018],AX
076A:0010 B44C          MOV     AH,4C
076A:0012 CD21          INT     21
076A:0014 1010          ADC     [BX+SI],DL
076A:0016 3020          XOR     [BX+SI],AH
076A:0018 0000          ADD     [BX+SI],AL
076A:001A 050C00        ADD     AX,000C
076A:001D 52            PUSH    DX
076A:001E 50            PUSH    AX
076A:001F E80E49        CALL    4930
```

```
-g

Program terminated normally
-u0
076A:0000 8CC8          MOV     AX,CS
076A:0002 8ED8          MOV     DS,AX
076A:0004 8D1E1400      LEA     BX,[0014]
076A:0008 8B07          MOV     AX,[BX]
076A:000A 034702        ADD     AX,[BX+02]
076A:000D A31800        MOV     [0018],AX
076A:0010 B44C          MOV     AH,4C
076A:0012 CD21          INT     21
076A:0014 1010          ADC     [BX+SI],DL
076A:0016 3020          XOR     [BX+SI],AH
076A:0018 40            INC     AX
076A:0019 3005          XOR     [DI],AL
076A:001B 0C00          OR      AL,00
076A:001D 52            PUSH    DX
076A:001E 50            PUSH    AX
076A:001F E80E49        CALL    4930
```

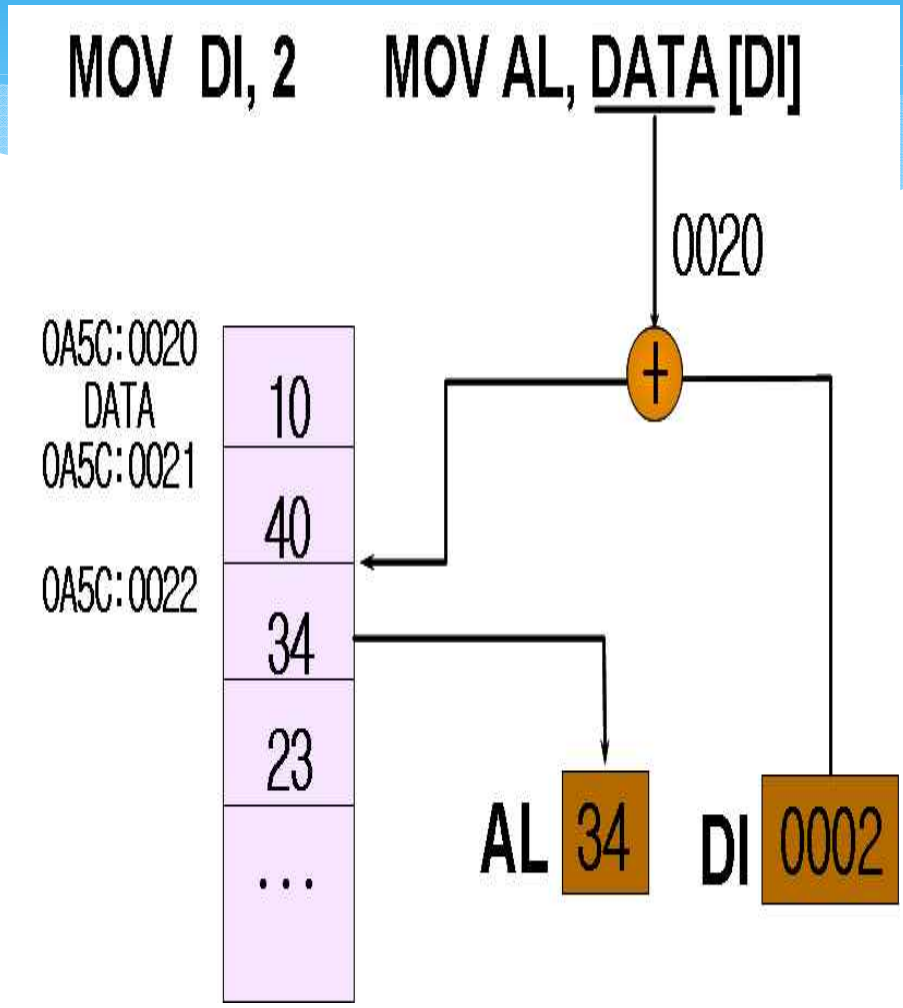
4) 인덱스 주소지정 방식

(1) 개념

- 변위값에 인덱스 레지스터의 값을 합하여 유효주소를 계산
- D[INX] : 유효주소 = D + INX
- 배열의 시작위치[특정 요소까지의 거리]
- 프로그램 실행 중에 기본이 되는 주소 값이 변하지 않는 데이터
- 예 : 고정 배열 데이터 내의 요소에 접근하는데 주로 사용

[예제4-8 : a_idx_33.asm] 인덱스 주소 지정 방식으로 DATA에 1월부터 6월까지의 임금이 있다고 한다. 3월분과 6월분의 임금을 각각 DATA1, DATA2에 저장하시오.

```
3월분→DATA1, 6월분→DATA2
1 ;인덱스 주소지정의 예
2   MAIN     SEGMENT
3   ASSUME     CS:MAIN, DS:MAIN
4              MOV     AX, CS
5              MOV     DS, AX
6              MOV     DI, 2
7              MOV     AL, DATA[DI]
8              MOV     DATA1, AL
9              MOV     DI, 5
10             MOV     AL, DATA[DI]
11             MOV     DATA2, AL
12             MOV     AH, 4CH
13             INT      21H
14  DATA     DB       10 ; 1월
15             DB       40 ; 2월
16             DB       34 ; 3월
               DB       23 ; 4월
               DB       34 ; 5월
               DB       56 ; 6월
   DATA1     DB          ?
   DATA2     DB          ?
28 MAIN       ENDS
29             END
```



[예제4-8 : a_idx_33.asm] 인덱스 주소 지정 방식으로 DATA에 1월부터 6월까지의 임금이 있다고 한다. 3월분과 6월분의 임금을 각각 DATA1, DATA2에 저장하시오.

```
076A:0004 BF0200        MOV     DI,0002
076A:0007 8A851C00      MOV     AL,[DI+001C]
076A:000B A22200        MOV     [0022],AL
076A:000E BF0500        MOV     DI,0005
076A:0011 8A851C00      MOV     AL,[DI+001C]
076A:0015 A22300        MOV     [0023],AL
076A:0018 B44C          MOV     AH,4C
076A:001A CD21          INT     21
076A:001C 0A28          OR      CH,[BX+SI]
076A:001E 2217          AND     DL,[BX]
-u
076A:0020 2238          AND     BH,[BX+SI]
076A:0022 0000          ADD     [BX+SI],AL
076A:0024 0450          ADD     AL,50
076A:0026 E89F0E        CALL    0EC8
076A:0029 83C404        ADD     SP,+04
076A:002C 3DFFFF        CMP     AX,FFFF
076A:002F 7403          JZ      0034
076A:0031 E91101        JMP     0145
076A:0034 B82F00        MOV     AX,002F
076A:0037 50            PUSH    AX
076A:0038 8B46FC        MOV     AX,[BP-04]
076A:003B 8B56FE        MOV     DX,[BP-02]
076A:003E 050C00        ADD     AX,000C
```

```
076A:0004 BF0200        MOV     DI,0002
076A:0007 8A851C00      MOV     AL,[DI+001C]
076A:000B A22200        MOV     [0022],AL
076A:000E BF0500        MOV     DI,0005
076A:0011 8A851C00      MOV     AL,[DI+001C]
076A:0015 A22300        MOV     [0023],AL
076A:0018 B44C          MOV     AH,4C
076A:001A CD21          INT     21
076A:001C 0A28          OR      CH,[BX+SI]
076A:001E 2217          AND     DL,[BX]
-u
076A:0020 2238          AND     BH,[BX+SI]
076A:0022 2238          AND     BH,[BX+SI]
076A:0024 0450          ADD     AL,50
076A:0026 E89F0E        CALL    0EC8
076A:0029 83C404        ADD     SP,+04
076A:002C 3DFFFF        CMP     AX,FFFF
076A:002F 7403          JZ      0034
076A:0031 E91101        JMP     0145
076A:0034 B82F00        MOV     AX,002F
076A:0037 50            PUSH    AX
076A:0038 8B46FC        MOV     AX,[BP-04]
076A:003B 8B56FE        MOV     DX,[BP-02]
076A:003E 050C00        ADD     AX,000C
```

5) 베이스 인덱스 주소지정 방식

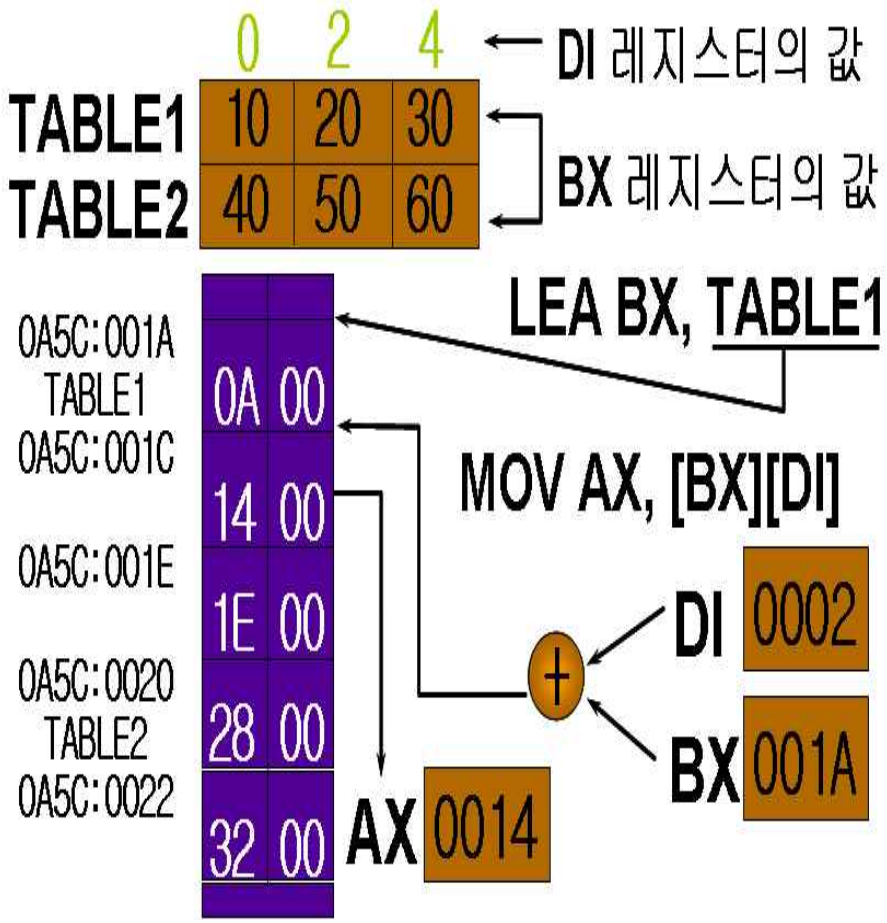
(1) 개념

- 지정된 베이스 레지스터의 값과 변위 값을 합한 값에 인덱스 레지스터의 값을 합하여 유효주소를 계산
- [BR+D] [INX] : 유효주소 = BR + D + INX
- 베이스 주소 지정 방식과 인덱스 주소 지정 방식의 장점으로
- 구조체와 같은 복잡한 자료의 표현을 위해 이용

[예제4-9 : a_iBx_34.asm] 베이스 인덱스 주소 지정 방식으로 TABLE1과 TABLE2에 저장되어 있는 데이터에서 두 번째 데이터만을 더해서 변수 RESULT에 저장하시오.

```
;베이스 인덱스 주소지정의 예
1 2   MAIN   SEGMENT
3   ASSUME      CS:MAIN, DS:MAIN
4             MOV     AX, CS
5             MOV     DS, AX
6             MOV     DI, 2
7             LEA     BX, TABLE1
8             MOV     AX, [BX] [DI]
9             LEA     BX, TABLE2
10            ADD     AX, [BX] [DI]
11            MOV     RESULT, AX
12            MOV     AH, 4CH
13            INT     21H
14   TABLE1   DW    10, 20, 30
15   TABLE2   DW    40, 50, 60
16   RESULT   DW       ?
17   MAIN     ENDS
18            END
```

(3) 주소지정 방법



[예제4-9 : a_iBx_34.asm] 베이스 인덱스 주소 지정 방식으로 TABLE1과 TABLE2에 저장되어 있는 데이터에서 두 번째 데이터만을 더해서 변수 RESULT에 저장하시오.

```
-u0
076A:0000 8CC8          MOV     AX,CS
076A:0002 8ED8          MOV     DS,AX
076A:0004 BF0200        MOV     DI,0002
076A:0007 8D1E1A00      LEA     BX,[001A]
076A:000B 8B01          MOV     AX,[BX+DI]
076A:000D 8D1E2000      LEA     BX,[0020]
076A:0011 0301          ADD     AX,[BX+DI]
076A:0013 A32600        MOV     [0026],AX
076A:0016 B44C          MOV     AH,4C
076A:0018 CD21          INT     21
076A:001A 0A00          OR      AL,[BX+SI]
076A:001C 1400          ADC     AL,00
076A:001E 1E            PUSH    DS
076A:001F 0028          ADD     [BX+SI],CH
-u
076A:0021 0032          ADD     [BP+SI],DH
076A:0023 003C          ADD     [SI],BH
076A:0025 0000          ADD     [BX+SI],AL
076A:0027 000E83C4      ADD     [C483],CL
076A:002B 043D          ADD     AL,3D
076A:002D FFFF          ???     DI
076A:002F 7403          JZ      0034
076A:0031 E91101        JMP     0145
076A:0034 B82F00        MOV     AX,002F
076A:0037 50            PUSH    AX
076A:0038 8B46FC        MOV     AX,[BP-04]
076A:003B 8B56FE        MOV     DX,[BP-02]
076A:003E 050C00        ADD     AX,000C
```

```
-u0
076A:0000 8CC8          MOV     AX,CS
076A:0002 8ED8          MOV     DS,AX
076A:0004 BF0200        MOV     DI,0002
076A:0007 8D1E1A00      LEA     BX,[001A]
076A:000B 8B01          MOV     AX,[BX+DI]
076A:000D 8D1E2000      LEA     BX,[0020]
076A:0011 0301          ADD     AX,[BX+DI]
076A:0013 A32600        MOV     [0026],AX
076A:0016 B44C          MOV     AH,4C
076A:0018 CD21          INT     21
076A:001A 0A00          OR      AL,[BX+SI]
076A:001C 1400          ADC     AL,00
076A:001E 1E            PUSH    DS
076A:001F 0028          ADD     [BX+SI],CH
-u
076A:0021 0032          ADD     [BP+SI],DH
076A:0023 003C          ADD     [SI],BH
076A:0025 004600        ADD     [BP+00],AL
076A:0028 0E            PUSH    CS
076A:0029 83C404        ADD     SP,+04
076A:002C 3DFFFF        CMP     AX,FFFF
076A:002F 7403          JZ      0034
076A:0031 E91101        JMP     0145
076A:0034 B82F00        MOV     AX,002F
076A:0037 50            PUSH    AX
076A:0038 8B46FC        MOV     AX,[BP-04]
076A:003B 8B56FE        MOV     DX,[BP-02]
076A:003E 050C00        ADD     AX,000C
```

[ex4.9] FOUR와 FIVE에 각각 값을 저장하고, 그 합을 구하는 프로그램을 작성하라. 합산한 값에 오버플로우가 발생하면 오류를 표시하는 부분(ERR)으로 분기하고 그렇지 않으면 합산한 값을 SAVE에 저장한다.

=>직접 주소 지정 방식

[ex4.10_i.asm] SUM이라는 기억공간에 TABLE에 저장된 데이터 10개를 누적 합산할 수 있는 프로그램을 작성하라.

=>인덱스 주소 지정 방식

[주소 지정 방식 요약]

1) 값 즉시 지정방식

연산항에 레지스터나 기억장소의 주소가 아닌 8BIT,16BIT 숫자가 직접 오게 하는 방법

ex) MOV ax,3004h

2) 직접 주소 지정방식

대상이 되는 레지스터나 메모리 번지를 그대로 연산항으로 하는 것

- 레지스터 직접 주소 지정방식

  ex) mov al,bl

- 메모리 직접 주소 지정방식 : 레지스터와 관계없이 주소를 직접 지정

  ex) mov ax,[0020]

3) 간접 주소 지정방식

- 레지스터 간접지정방식

  :변위는 포함하지 않고 베이스 레지스터(BR, BX) 또는 인덱스 레지스터(DI,SI)의 값이 주소가 된다.

  ex) lea bx, data1
  mov ax, [bx]

- 베이스 주소 지정방식 : 변위값에 베이스레지스터 값을 더해서 실효번지 구함

  : 변위값 + 베이스 레지스터 bx, bp값

  ex) mov ax,[bx+2]

- 인덱스 주소 지정방식 : 변위값에 인덱스레지스터 값을 더해서 실효번지 구함

  : 변위값+인덱스레지스터 SI,DI로 실효번지 구하기

  ex) mov dl 2
  mov al, data[di]

- 베이스 인덱스 주소 지정방식

  : 변위값에 BR, INX 값을 더해서 실효번지 구함. 구조체의 구조를 액세스 하는데 유용

  : 베이스번지 지정방식에 인덱스 레지스터 이용하는 방식

  ex) lea bx,table2
  mov ax,[bx] [di]