12주 1강

주소 지정방식

주소 지정 방식

- 주기억장치에서 데이터가 저장된 위치를 주소(address)라 한다.
- 다양한 주소지정 방식(addressing mode)을 컴퓨터 시스템에 사용.
- 주소 지정 방식이 다양한 이유
  - 제한된 명령어 비트들을 적절하게 이용하여 효율적으로 오퍼랜드를 지정
  - 더 큰 용량의 기억장치를 사용할 수 있도록 하기 위한 것이다.

- 주소지정 방식의 표기 방법

정의 내용	표기 방법
유효 주소(기억장치의 실제 주소)	EA
기억장치 주소	A
레지스터 번호	R
기억장치 A번지의 내용	(A)
레지스터 R번지의 내용	(R)

주소 지정 방식의 분류

① 직접 주소지정 방식(direct addressing mode)

② 간접 주소지정 방식(indirect addressing mode)

③ 묵시적 주소지정 방식(implied addressing mode)

④ 즉치 주소지정 방식(immediate addressing mode)

⑤ 레지스터 주소지정 방식(register addressing mode)

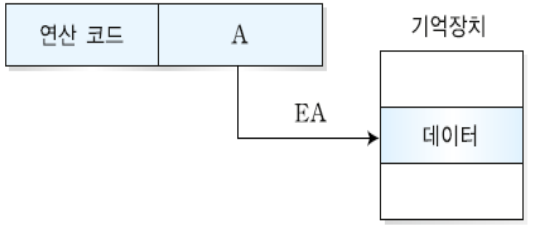
⑥ 레지스터 간접 주소지정 방식(register-indirect addressing mode)

⑦ 변위 주소지정 방식 (displacement addressing mode)

- 상대 주소지정 방식(relative addressing mode)

직접 주소지정 방식(Direct Addressing Mode)

- 오퍼랜드 필드의 내용이 유효 주소로 가장 일반적인 개념 방식
- 지적하려는 위치를 오퍼랜드에서 직접 표현하는 형식



EA = A

(유효 주소 = 기억장치 주소)

- 데이터 인출을 위해 오퍼랜드에 저장된 해당 주소의 기억장치에 한 번만 액세스하는 장점이 있다.

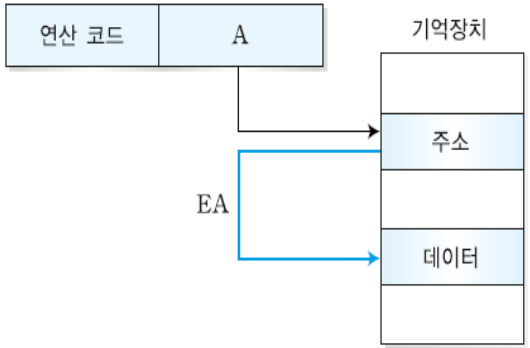
- 연산 코드를 제외하고 남은 비트들이 주소 비트로 사용
  - 지정할 수 있는 기억장소의 수가 제한
  - 많은 수의 주소를 지정할 수 없는 단점을 갖는다.

간접 주소지정 방식(Indirect Addressing Mode)

- 오퍼랜드에 유효 주소가 저장되어 있는 주소가 저장, 이 주소가 가리키는 기억 장소에서 유효 주소를 얻을 수 있다.

$$EA = (A)$$

(유효 주소 = 기억장치 A번지의 내용)



- 간접 주소지정 방식의 동작
  - 두 번의 기억장치 액세스가 필요
  - 첫 번째는 유효주소가 저장된 곳에 액세스하는 것
  - 두 번째는 유효주소에 액세스하여 실질적인 데이터를 얻는 것

간접 주소지정 방식의 특징

- 최대 기억장치용량이 중앙처리장치가 한 번에 액세스할 수 있는 단어의 길이에 의하여 결정된다는 것이 장점
  - 기억장치의 구조 변경 등을 통해 확장이 가능
  - 단어 길이가 n비트라면, 최대 2^n개의 기억장소를 주소 지정할 수 있다.
- 실행 사이클 동안 두 번의 기억장치 액세스가 필요하다는 단점
- 명령어 형식에서 주소지정 방식을 위한 간접비트(I)필드가 필요
- 간접 주소지정방식을 사용하는 컴퓨터에서의 명령어 형식
  - I = 0이면 직접 주소지정 방식, I = 1이면 간접 주소지정 방식이 된다.

연산 코드	I	오퍼랜드(기억장치 주소)

묵시적 주소지정 방식과 즉치 주소지정 방식

묵시적 주소지정 방식(Implied Addressing Mode)

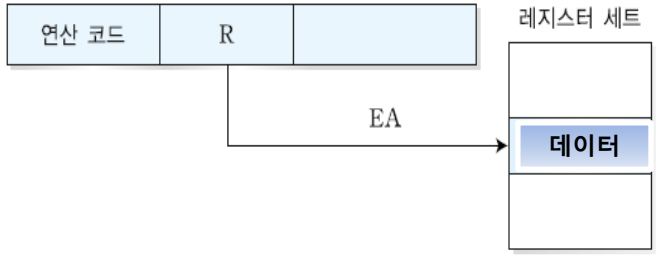
- 데이터 위치가 별도로 지정되지 않고, 명령어의 연산 코드가 내포
- 명령어 길이가 짧다는 장점을 갖지만 명령어의 종류가 제한
- SHL 명령어 :
  - shift left에서 온 것으로 AC(누산기)의 내용을 좌측으로 이동시키는 것
  - 데이터가 AC에 저장되어 있다는 것을 명령어에 표시하지 않는다.

즉치 주소지정 방식(Immediate Addressing Mode)

- 직접 데이터 형식 : 데이터가 명령어에 포함되어 있는 방식
  - 오퍼랜드 필드의 내용이 연산에 사용할 실제 데이터가 된다.
  - 레지스터나 변수의 초기 값을 어떤 상수값으로 설정하는 데 유용하게 사용
- 기억장치에 액세스할 필요가 없다는 장점을 가진다.
- 상수 값의 크기가 오퍼랜드 필드의 비트 수에 의하여 제한된다.

레지스터 주소지정 방식(Register Addressing Mode)

- 연산에 사용할 데이터가 레지스터에 저장되어 있는 방식이다.
- 오퍼랜드 부분이 레지스터 번호를 나타내며, 유효주소는 레지스터 번호



$$EA = R$$

(유효 주소 = 레지스터 번호)

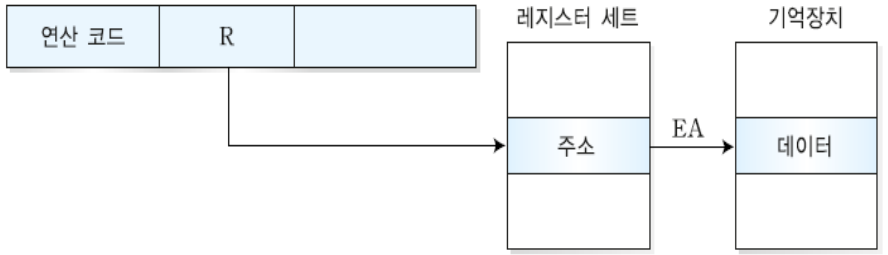
- 오퍼랜드의 비트수가 k비트이면, 주소지정에 사용할 수 있는 레지스터들의 수는 2^k개가 된다.
- 비트 수가 적어도 되며, 데이터 인출을 위하여 기억장치에 액세스할 필요가 없다.
- 데이터를 저장할 수 있는 공간이 중앙처리장치 내부의 레지스터들로 제한된다.

레지스터 간접 주소지정 방식

- 오퍼랜드 필드가 레지스터 번호를 가리키고, 그 레지스터에 저장된 내용이 유효주소다.

$$EA = (R)$$

(유효 주소 = 레지스터의 저장내용)

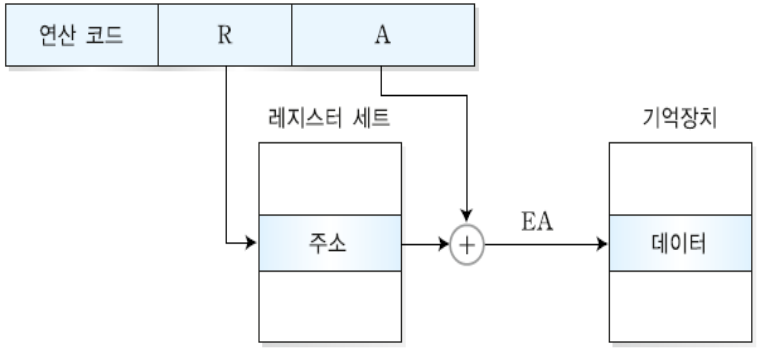


- 이 방식의 장점은 주소를 지정할 수 있는 기억장치 영역이 확장된다.
  - 레지스터의 길이에 따라 주소지정 영역이 결정된다.
  - 레지스터의 길이가 16비트라면, 주소지정 영역은 2^{16}비트(64K 바이트)가 된다.

- 간접 주소지정 방식에서는 두 번의 기억장치 액세스 과정이 있지만, 이 방법에서는 한 번의 기억장치 액세스만 있다.

변위 주소지정 방식

- 직접 주소지정 방식과 레지스터 간접 주소지정 방식을 조합한 것
- 오퍼랜드 필드는 레지스터 번호필드와 변위 값 필드로 2개가 존재한다.
- 두 오퍼랜드의 조합으로 유효 주소가 생성 된다.



- R이 가리키는 레지스터의 내용과 변위 값 A를 더하여 유효 주소를 결정

상대 주소지정 방식

- PC를 레지스터로 사용하기 때문에 주로 분기 명령어에서 사용

$$EA = A + (PC)$$

(유효 주소 = 변위 값 + 프로그램 카운터의 내용)

  - A ≥ 0이면, 앞 방향으로 분기, A < 0이면 후 방향으로 분기

- 450번지에 저장된 JUMP 명령이 인출된 후, PC는 451이 될때
  - A가 +21이면 분기 목적지 주소는 472(= 451 + 21)번지
  - A가 -50이면 분기 목적지 주소는 401(= 451 - 50)번지

- 기억장치 주소가 명령어에 포함되어야 하는 일반적인 분기 명령어보다 적은 수의 비트만 있으면 된다

- 분기 범위가 오퍼랜드 변위 필드의 길이에 의하여 제한을 받는다.

다음 시간

12주 2강 . 인터럽트 처리