Implied 모드
피연산자가 묵시적으로 명령어의 정의에 따라 정해져있음
Ex) '누군가의 보수를 취하라' -> 피연산자가 누산기에 있기 때문에 implied 모드가 되며, 누산기를 사용하는 모든 명령어는 Implied 명령어다
스택 구조일 경우, 피연산자가 스택의 top 이기 때문에 implied 명령어다.

Immediate 모드
이 모드에서는 피연산자가 명령어 그 자체 내에 있다.
주소 필드라기보다도 피연산자 필드를 가지게 된다.
상수 레지스터에 초기 값으로 줄 때 편리하다
*명령어의 주소 필드는 메모리의 주소나 레지스터를 지정하는것은 레지스터 모드이다.

레지스터 모드
레지스터 모드는 cpu 내의 레지스터에 피연산자가 있다.
2^k 레지스터 중 하나를 선택할 수 있다.

레지스터 간접 모드
명령어가 피연산자의 주소를 가지고있는 레지스터를 지정
선택된 레지스터는 피연산자가 아닌 피연산자의 주소이다.
적은 비트가 든다는 장점이 있지만 피연산자 access time이 증가한다.

자동증가 또는 자동감소 모드
레지스터 값이 메모리를 엑세스하고 난 직후 자동적으로 하나 증가하거나 감소한다는 것을 제외하면 레지스터. 간접 모드와 같다
메모리가 있는 데이터가 어떤 표라면, 편하다

유효 주소(EA, Effective Address)
주어진 주소 지정 모드에 따라 계산된 주소
연산 유형 명령어에서 피연산자의 주소
분기 유형 명령어에서 제어가 분기되는 주소

직접 주소 모드
명령어의 주소 부분이 그대로 유효 명령어의 주소 필드에 의해 직접적으로 주어진다
분기(branch) 형식의 명령어에서는 실제 분기할 주소를 나타낸다

간접 주소 모드
간접 주소 모드에서는 명령어의 주소 필드가 가리키는 주소에는 유효주소가 있다.
이 명령어를 수행할 때에는 메모리로부터 명령어를 fetch하고 그것의 주소 부분으로부터 다시 유효 주소를 메모리에서 가져와서 연산한다.

유효 주소 = 명령어에서의 주소 부분 + CPU 내의 현재 레지스터의 값
이때 CPU의 레지스터는 프로그램 카운터나 인덱스 레지스터나 베이스 레지스터가 될 수 있다.

상대 주소 모드
프로그램 카운터가 명령어의 주소 부분과 더해져서 유효 주소가 결정된다.
명령어의 주소 부분은 보통 기호를 포함한 수(2의 보수 표현)이며 음수나 양수 둘 다 될 수 있다.
이 숫자가 프로그램 카운터에 더해져서 유효 주소가 된다.

베이스 레지스터 어드레싱 모드
베이스 레지스터의 내용이 명령어의 주소 부분에 더해져서 유효 주소가 결정된다.
인덱스드 어드레싱 모드와의 차이점: 인덱스 레지스터 대신 베이스 레지스터가 사용되었다.
인덱스 레지스와 쓰임새에서 차이점이 있다.
인덱스 레지스터는 주소 부분에 대한 상대적인 위치를 가지고 있음에 반해, 베이스 레지스터는 베이스 주소를 가지고 있고, 명령어의 주소 부분은 이 베이스 주소로부터의 상대적인 displacement가 된다.
프로그램이나 데이터가 메모리의 한 세그먼트로부터 다른 세그먼트로 옮겨질 때, 명령어의 주소는 이러한 위치의 변경을 반영해야한다.
베이스 레지스터가 있다면 명령어의 displacement는 변경될 필요가 없다
베이스 레지스터 값이 다른 메모리 세그먼트의 시작 부분을 참고로 해서 변경하면 된다.