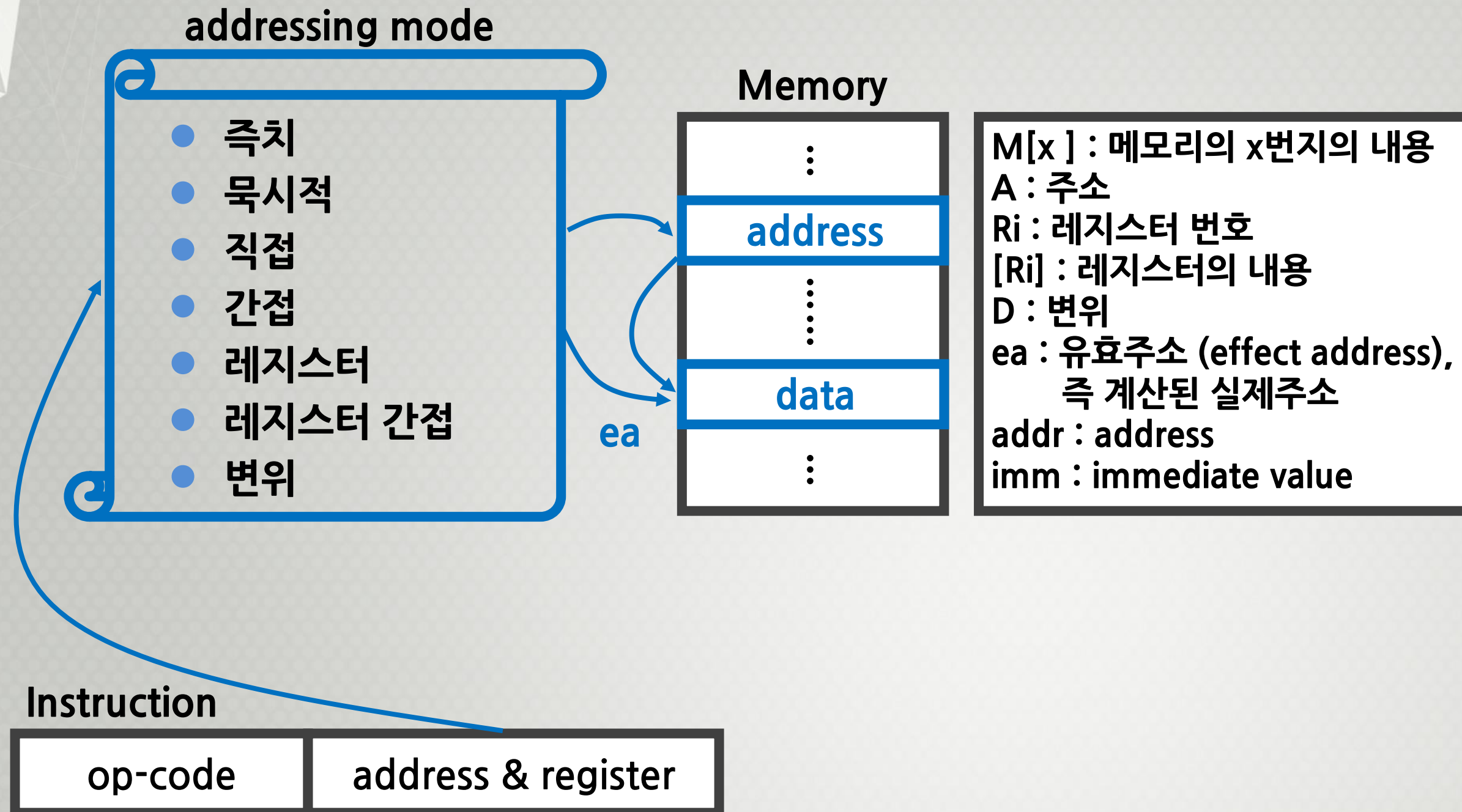


■ 명령어 주소지정(Addressing Mode) 방식

- 정해진 명령어의 비트들은 그 수에 있어서 매우 제한적이다.
- 그 안에 오퍼랜드들 만으로 메모리를 지정하는 것은 매우 제한적이다.
- 더 큰 용량의 메모리에 접근하기 위해서는 직접 또는 간접으로 다양한 방법이 요구된다.
- 이미 설계된 CPU의 종류마다 그 수가 매우 다양하고 다르게 제안되어 있다.

명령어 주소지정(Addressing Mode) 방식

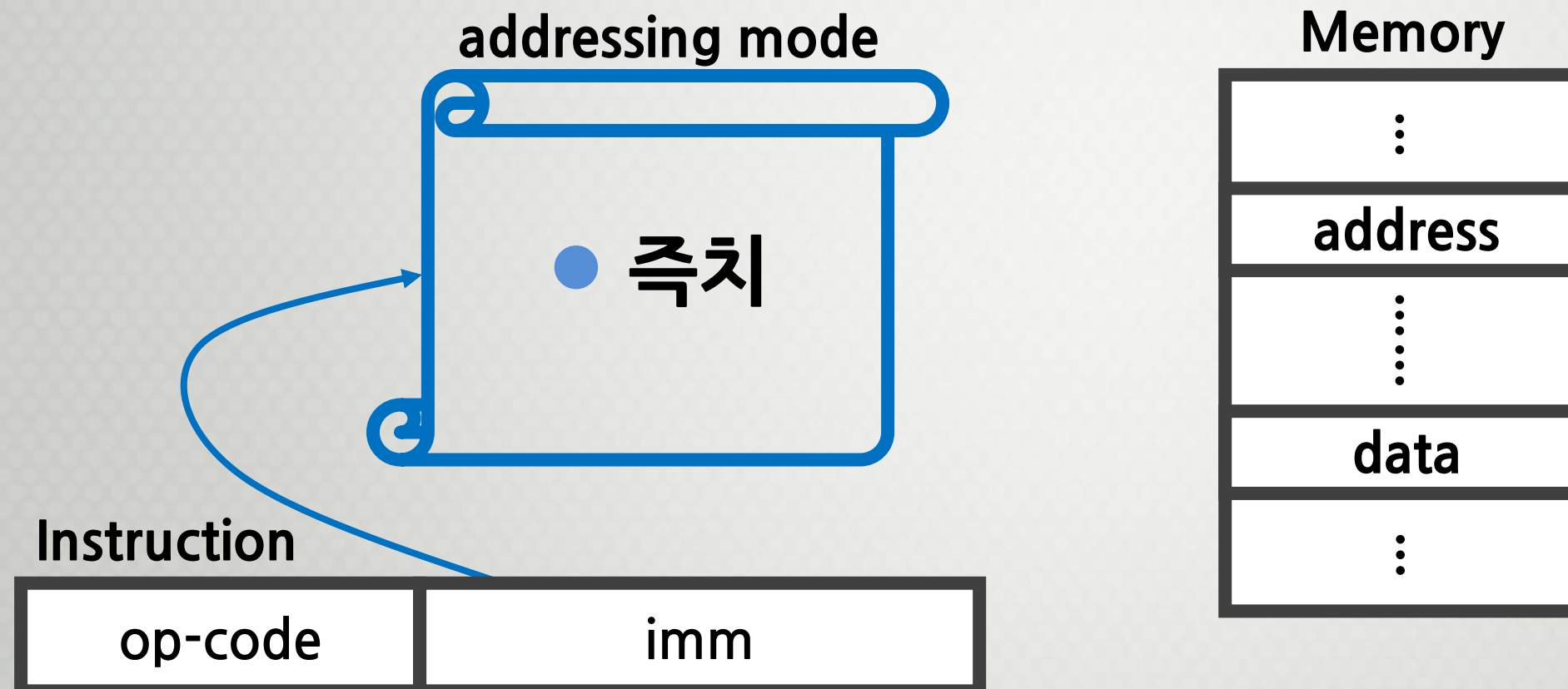


■ 명령어 주소지정(Addressing Mode) 방식

1) 즉치(immediate addressing mode)

- 프로그램에서 상수 값으로 사용된다.
- 명령어 내의 오퍼랜드가 실제 데이터가 되는 것을 의미한다.
- CPU는 메모리로 부터 데이터를 인출하는 과정이 필요 없다.
(ea가 필요 없음)

ex) Mov R1, #8H ; R1 ← 8H
 Add R2, #9H ; R2 ← R2 + 9H

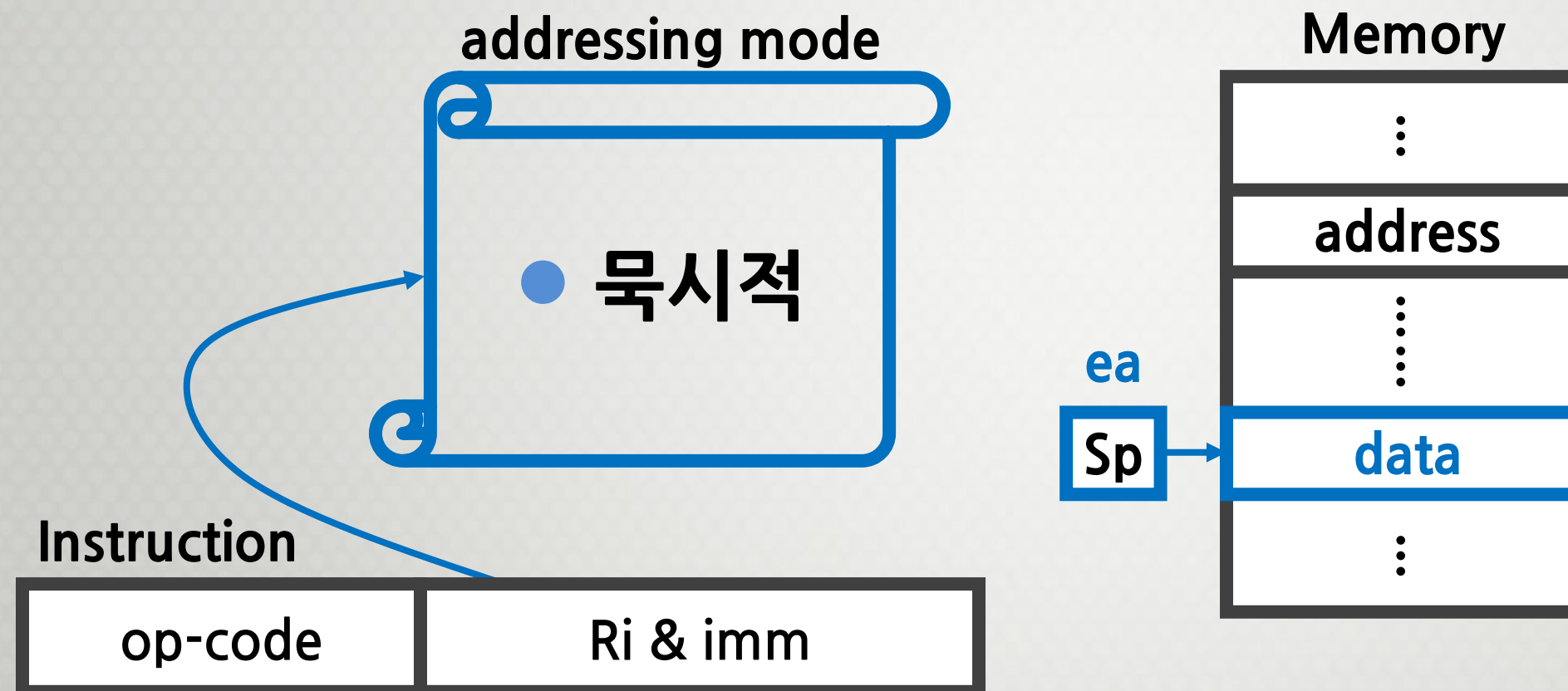


명령어 주소지정(Addressing Mode) 방식

2) 묵시적(implied addressing mode)

- 명령어 내에 필요한 데이터의 위치를 지정하지 않는다.
- 쉬프트 연산이나 스택에 관련된 연산 사용된다.
- Sp가 유효주소(ea)가 된다.

ex) Asl 2 ; $AC \leftarrow AC \ll 2$
Push R2 ; $M[Sp] \leftarrow R2$

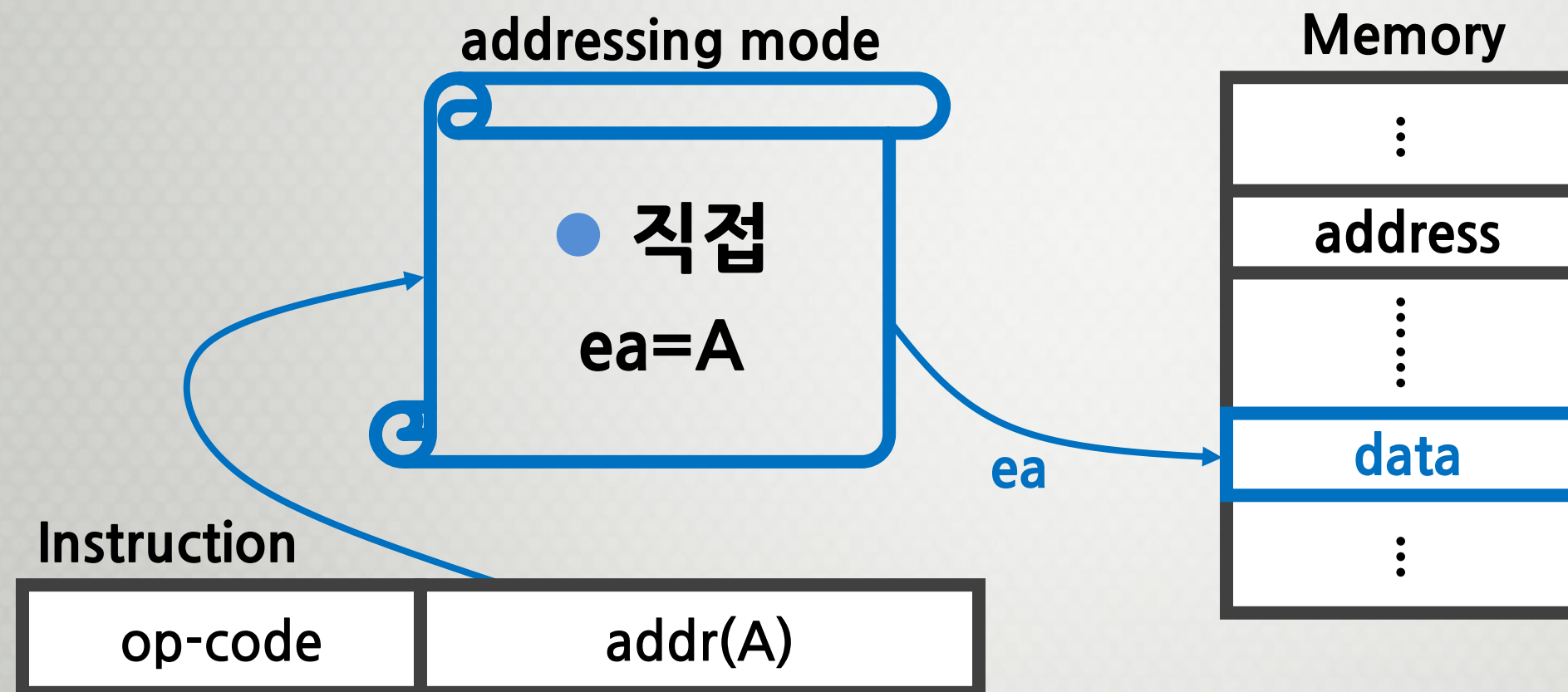


■ 명령어 주소지정(Addressing Mode) 방식

3) 직접(direct addressing mode)

- 절대 주소지정(absolute addressing mode)이라고도 한다.
- 명령어 내에 오퍼랜드가 주소 정보로 사용되어 메모리 내에 필요한 데이터의 위치를 지정한다.

ex) Mov R1, 5H ; R1 \leftarrow M[5H]
 Add R2, 5H ; R2 \leftarrow R2 + M[5H]

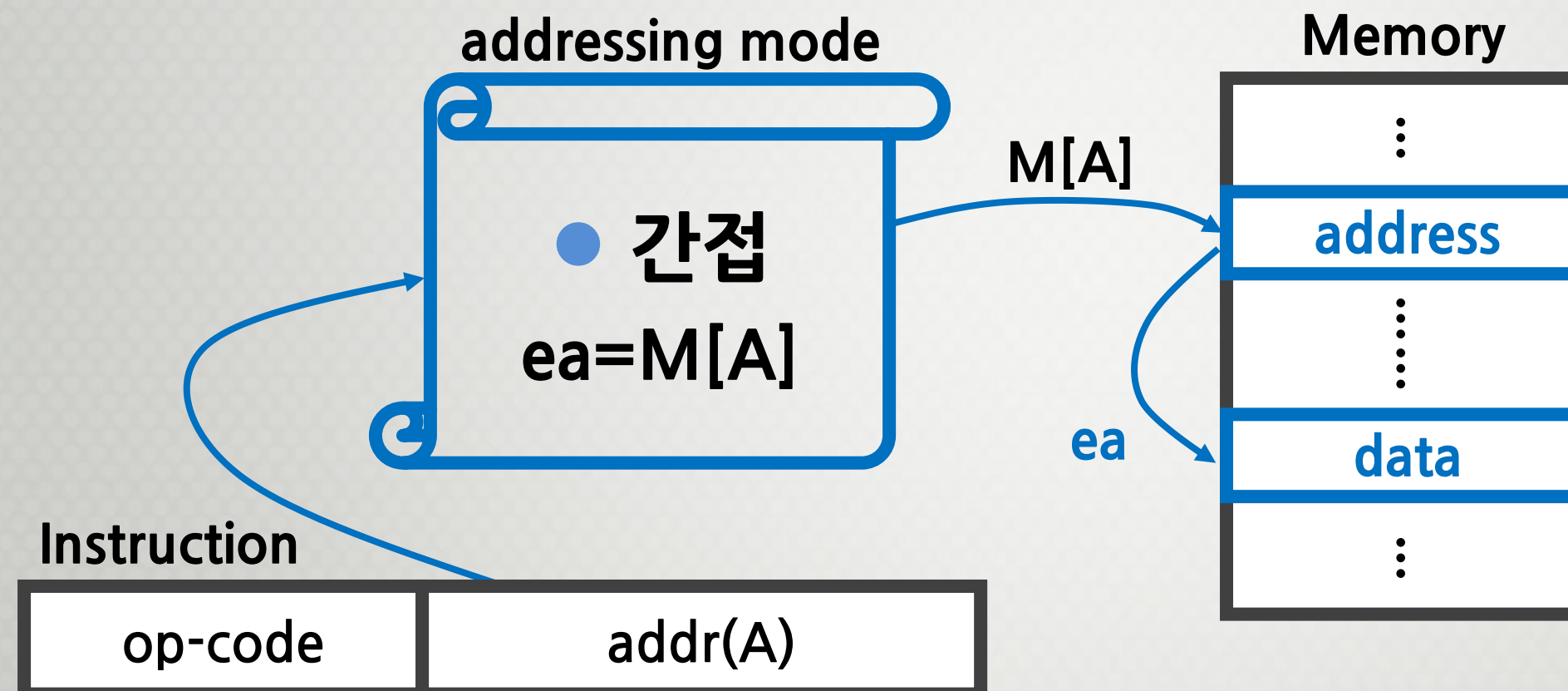


명령어 주소지정(Addressing Mode) 방식

4) 간접(indirect addressing mode)

- 직접주소지정 방식의 단점(주소 범위가 짧다)을 해결한다.
- 메모리 내의 데이터를 또 다른 주소 정보를 사용함으로써 더 큰 영역의 메모리 접근을 확보할 수 있다.
- 단점으로는 메모리접근의 지연시간이 발생한다는 것이다.

ex) Mov R1, (5H) ; $R1 \leftarrow M[M[5H]]$

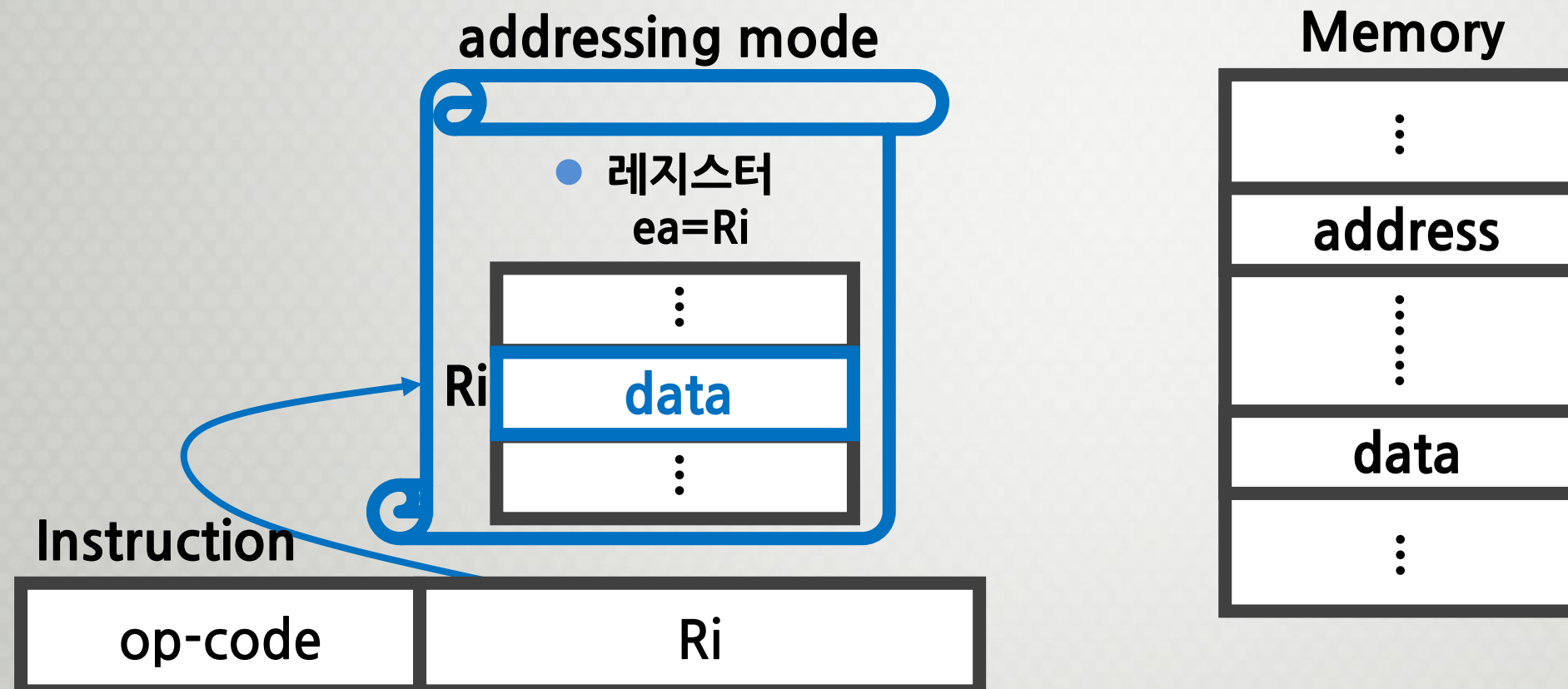


■ 명령어 주소지정(Addressing Mode) 방식

5) 레지스터(register addressing mode)

- 연산에 사용될 데이터가 레지스터에 저장되어 있다.
- 오퍼랜드의 내용은 레지스터 번호로 사용된다.
- 메모리 내의 데이터에 접근할 필요가 없다.
- 메모리 접근의 지연시간이 없다.

ex) Mov R1, R2 ; R1 ← R2
Add R3, R4 ; R3 ← R3 + R4

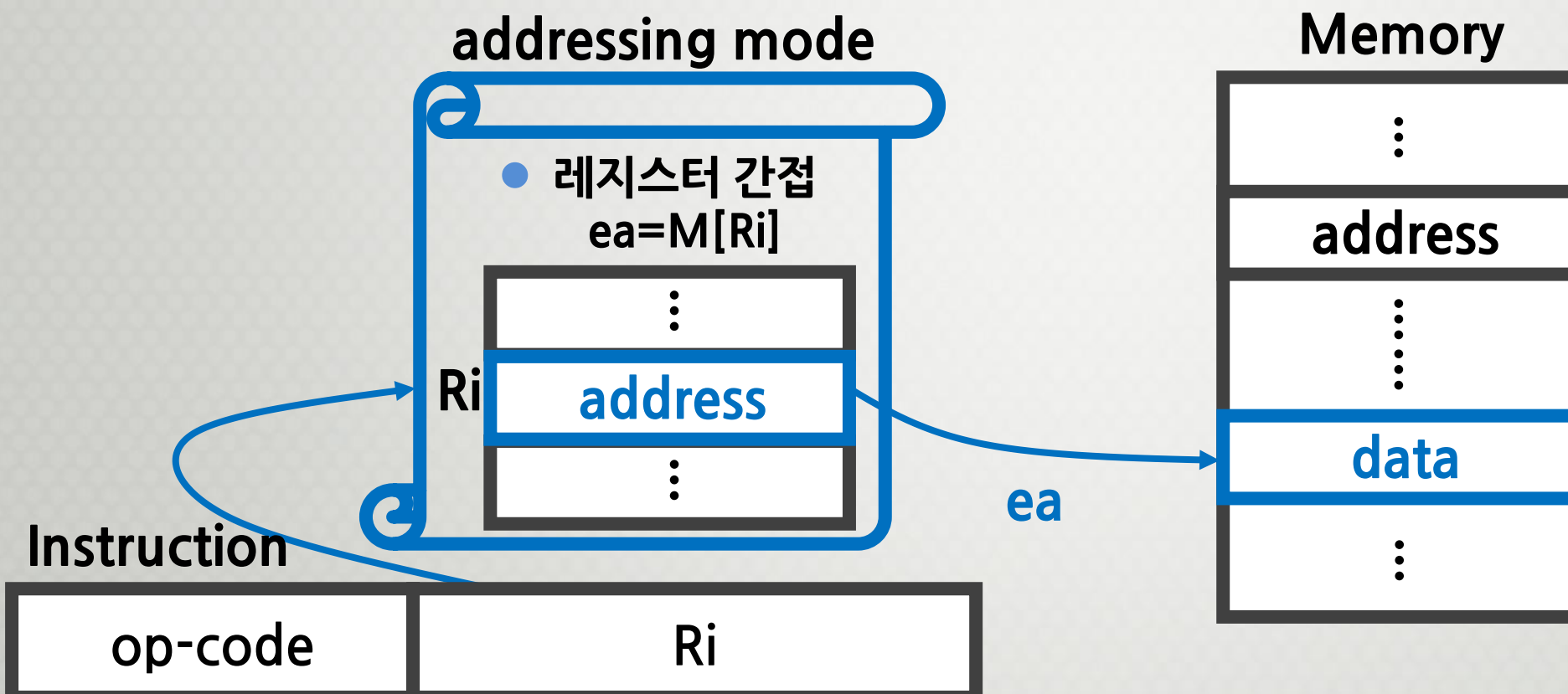


■ 명령어 주소지정(Addressing Mode) 방식

6) 레지스터 간접(register indirect addressing mode)

- 오퍼랜드의 내용은 레지스터 번호로 사용된다.
- 오퍼랜드에서 지정한 레지스터의 내용이 메모리의 주소정보이다.
- 레지스터의 비트 수에 따라 지정할 수 있는 메모리 영역 결정된다.

ex) Mov R1, (R2) ; R1 ← M[R2]
 Add R3, (R4) ; R3 ← R3 + M[R4]



명령어 주소지정(Addressing Mode) 방식

7)변위(displacement addressing mode)

- 두 개의 오퍼랜드로 구성되고, 하나는 레지스터이고 다른 하나는 변위(D)로 사용된다.
- 유효주소는 지정된 레지스터의 내용에 변위를 합한 것이다.
$$ea = Ri(pc, ix, br) + D$$
- 레지스터의 종류에 따라 상대(pc), 인덱스(index), 베이스(base) 레지스터 주소지정 방식이라고 한다.

ex) Mov R1, 9(R2) ; R1 \leftarrow M[9 + [R2]]
 Mov R3, TABLE(PC) ; R3 \leftarrow PC - TABLE

