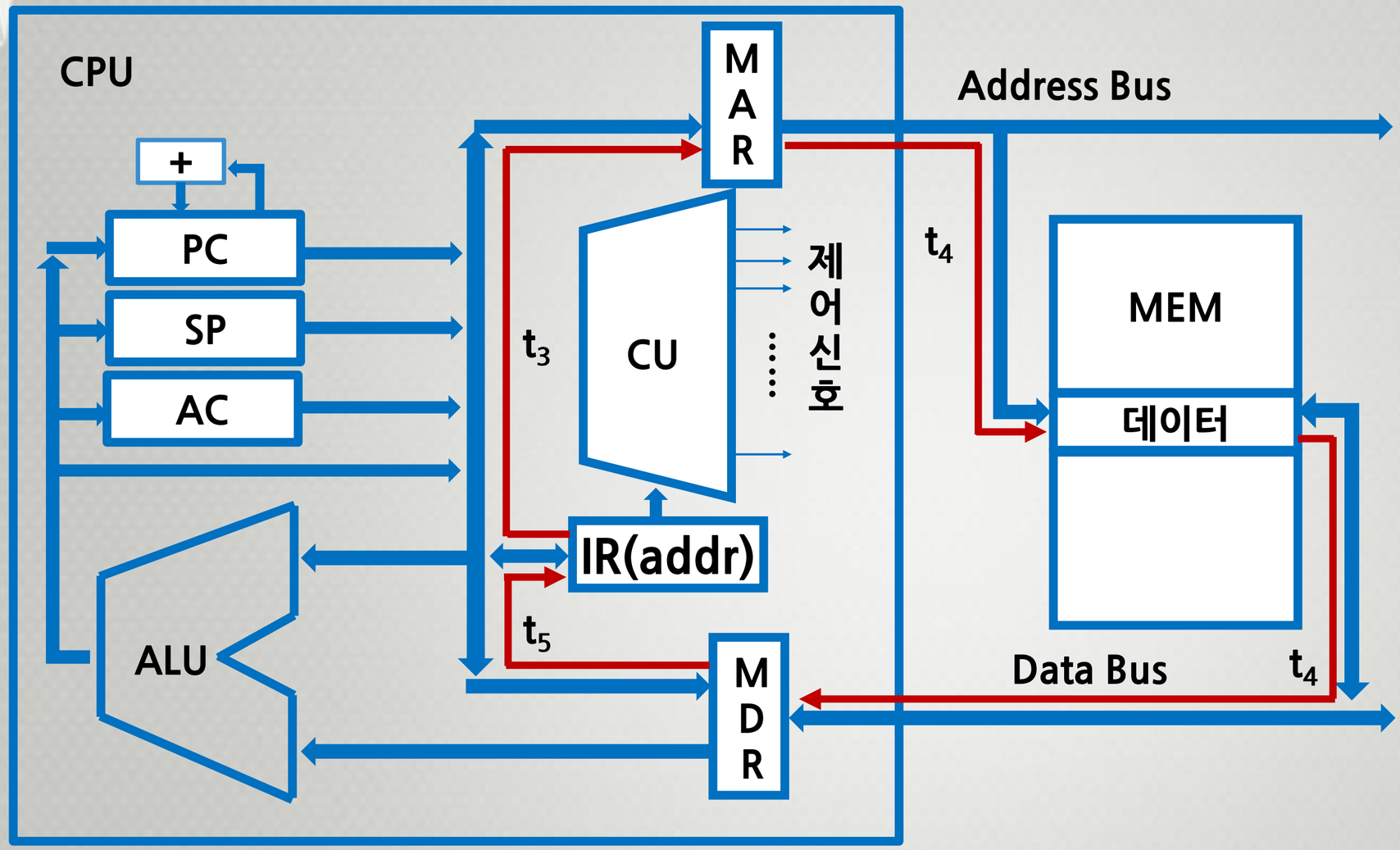
간접 사이클(Indirect Cycle)

- 간접 사이클에서 클럭주기(t_3, t_4, t_5)에 따른 읽혀올 유효주소의 흐름도



간접 사이클(Indirect Cycle)

간접 사이클의 마이크로 연산(Micro-operation)

- 명령어에 포함되어 있는 주소정보를 이용하여,
  실제 명령어 실행에 필요한 데이터를 인출하는
  사이클로서 간접 주소지정 방식에서 사용되며,
  이것은 인출 사이클과 실행 사이클 중간에 실행된다.

t_3 : MAR ← IR(addr)
t_4 : MDR ← M[MAR]
t_5 : IR(addr) ← MDR

↳ 여기서, t_3, t_4, 및 t_5는 CPU 클럭주기

간접 사이클(Indirect Cycle)

간접 사이클의 마이크로 연산(Micro-operation)

클럭 t_3	명령어 레지스터인 IR에 있는 명령어의 오퍼랜드 (addr) 값을 MAR로 전송한다.
클럭 t_4	그 주소 값이 지정하는 기억장치 주소로부터 읽혀진 데이터를 데이터 버스를 통하여 MDR에 저장한다.
클럭 t_5	전송된 MDR의 데이터는 유효주소 정보이기에 그 값을 다시 IR의 어드레스 필드로 전송한다.

예)
- CPU 클럭이 2GHz 인 경우 클럭 주기 및 ADD 명령어 내에 간접 사이클이 포함된 수행시간
  - 클럭 주기 = 1 sec ÷ 2×10^9 = 0.5ns
  - 인출 및 실행사이클 시간 = 0.5ns × (3+3+3) = 4.5ns

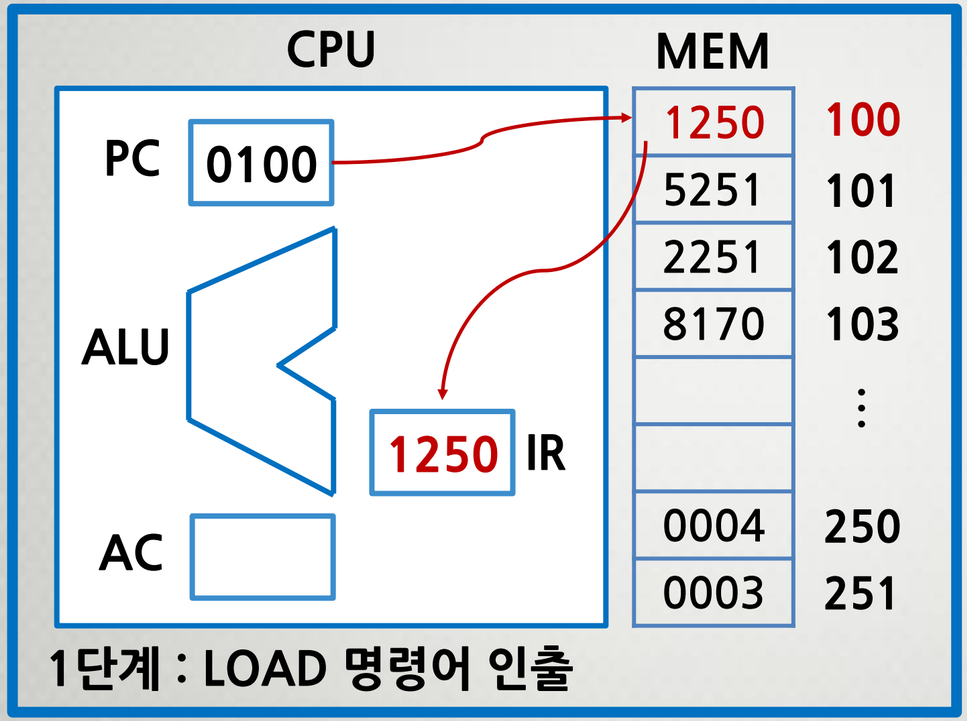
여러 가지 명령어의 종합적인 실행과정(예제)

- 어셈블리 프로그램이 아래 표와 같이 작성되었을 때 실행과정 살펴보기

주소 (Address)	명령어 (Instruction)	기계 코드 (Machine Code)
100	LOAD 250	1250
101	ADD 251	5251
102	STORE 251	2251
103	JUMP 170	8170

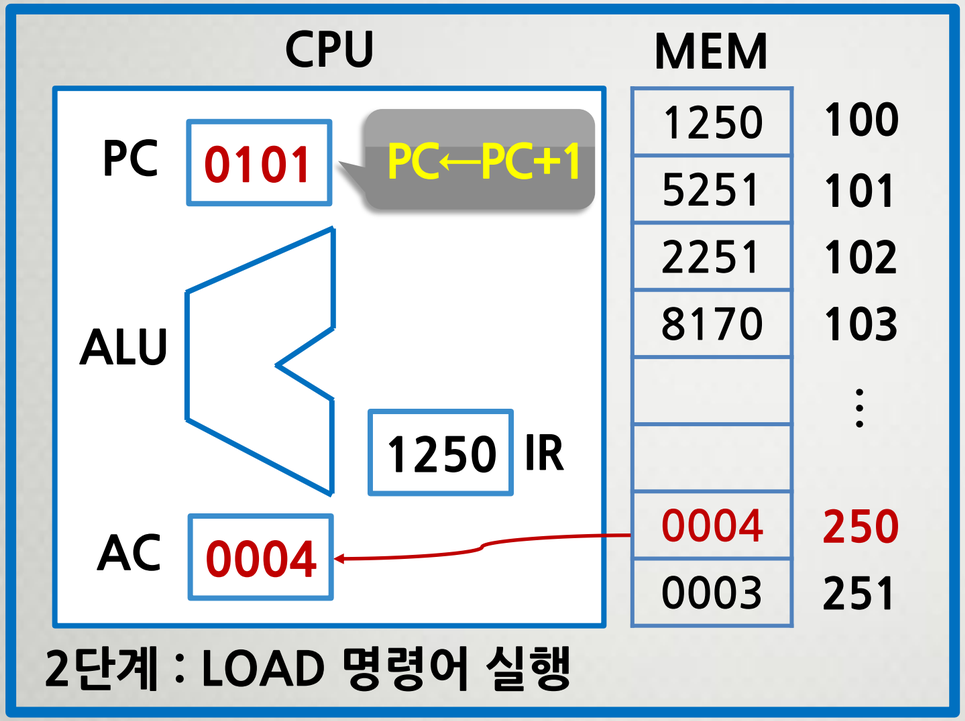
여러 가지 명령어의 종합적인 실행과정(예제)

주소(Address)	명령어(Instruction)	기계 코드(Machine Code)
100	LOAD 250	1250
101	ADD 251	5251
102	STORE 251	2251
103	JUMP 170	8170



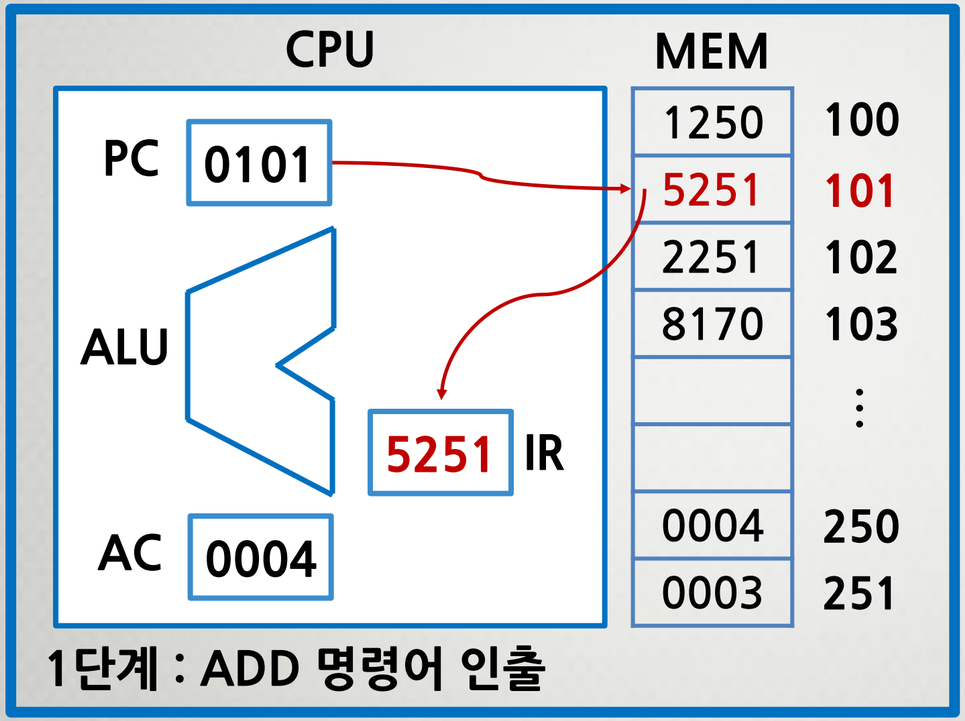
여러 가지 명령어의 종합적인 실행과정(예제)

주소(Address)	명령어(Instruction)	기계 코드(Machine Code)
100	LOAD 250	1250
101	ADD 251	5251
102	STORE 251	2251
103	JUMP 170	8170



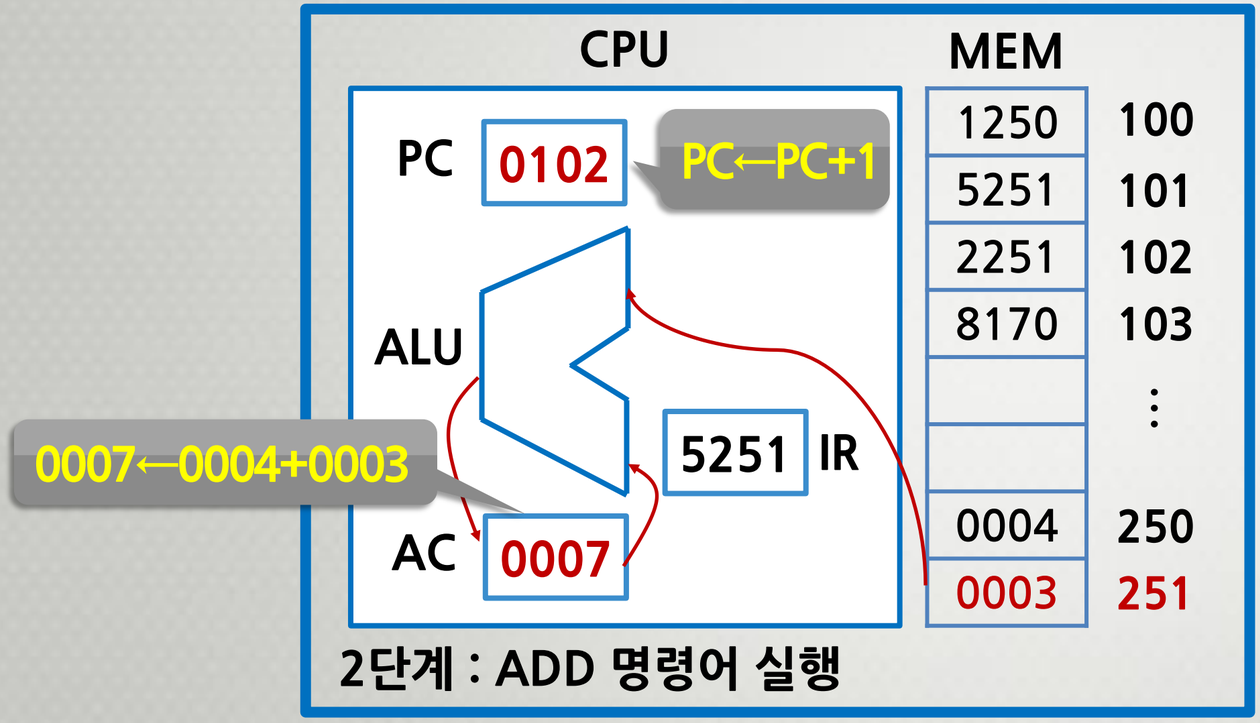
여러 가지 명령어의 종합적인 실행과정(예제)

주소(Address)	명령어(Instruction)	기계 코드(Machine Code)
100	LOAD 250	1250
101	ADD 251	5251
102	STORE 251	2251
103	JUMP 170	8170



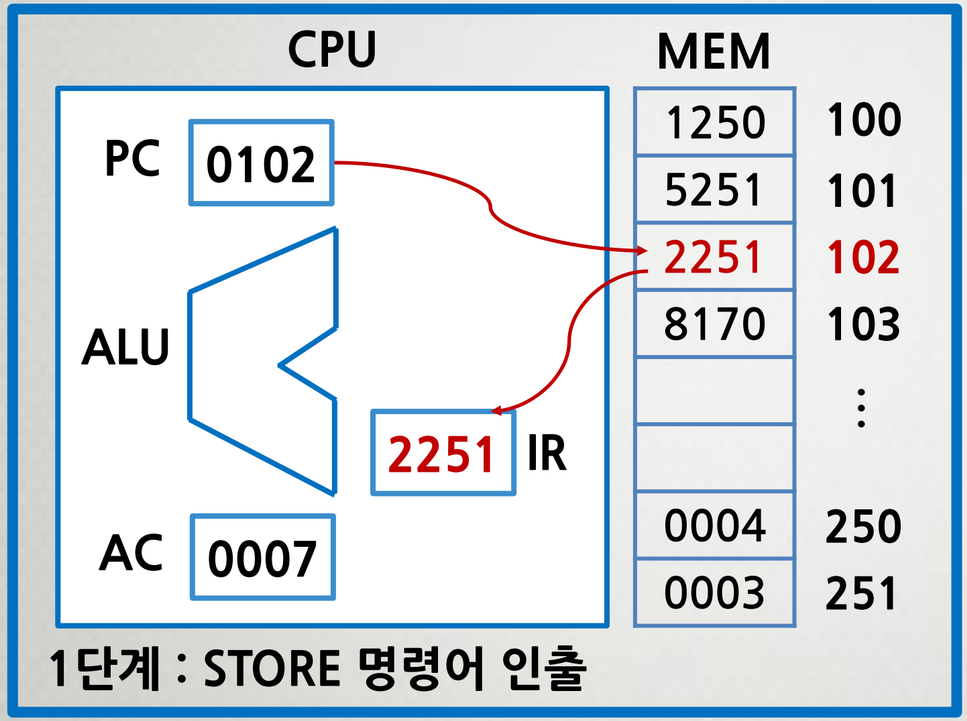
여러 가지 명령어의 종합적인 실행과정(예제)

주소(Address)	명령어(Instruction)	기계 코드(Machine Code)
100	LOAD 250	1250
101	ADD 251	5251
102	STORE 251	2251
103	JUMP 170	8170



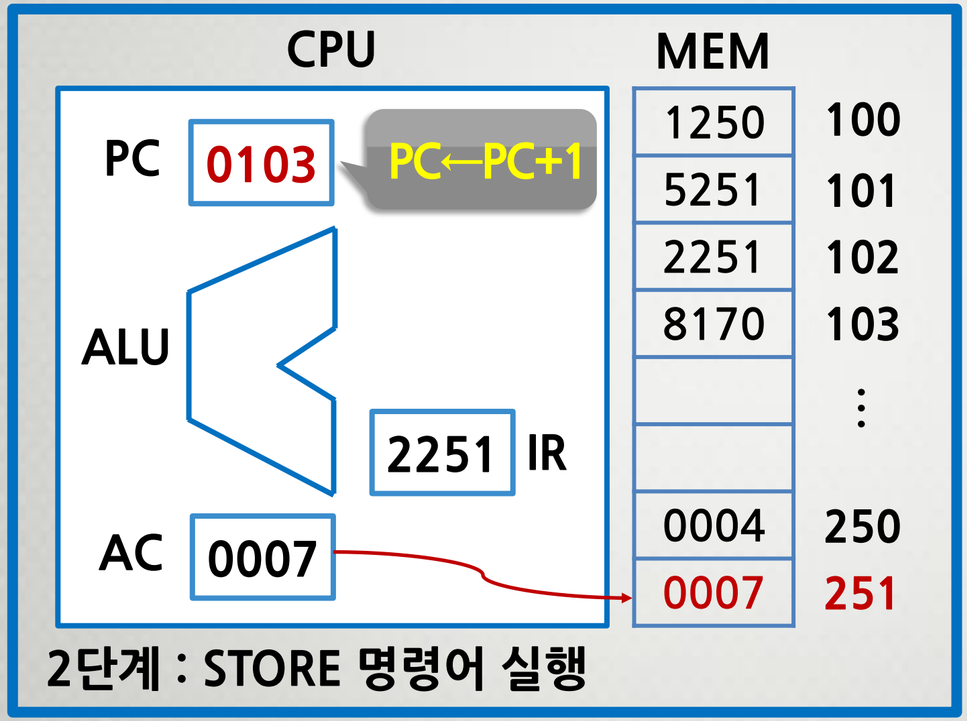
여러 가지 명령어의 종합적인 실행과정(예제)

주소(Address)	명령어(Instruction)	기계 코드(Machine Code)
100	LOAD 250	1250
101	ADD 251	5251
102	STORE 251	2251
103	JUMP 170	8170



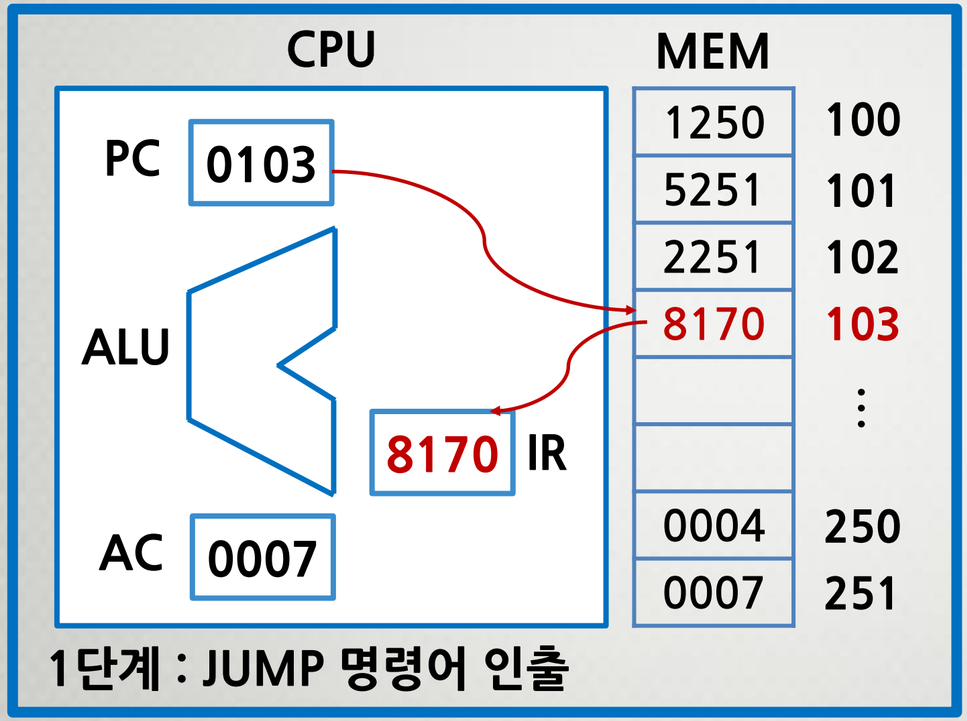
여러 가지 명령어의 종합적인 실행과정(예제)

주소(Address)	명령어(Instruction)	기계 코드(Machine Code)
100	LOAD 250	1250
101	ADD 251	5251
102	STORE 251	2251
103	JUMP 170	8170



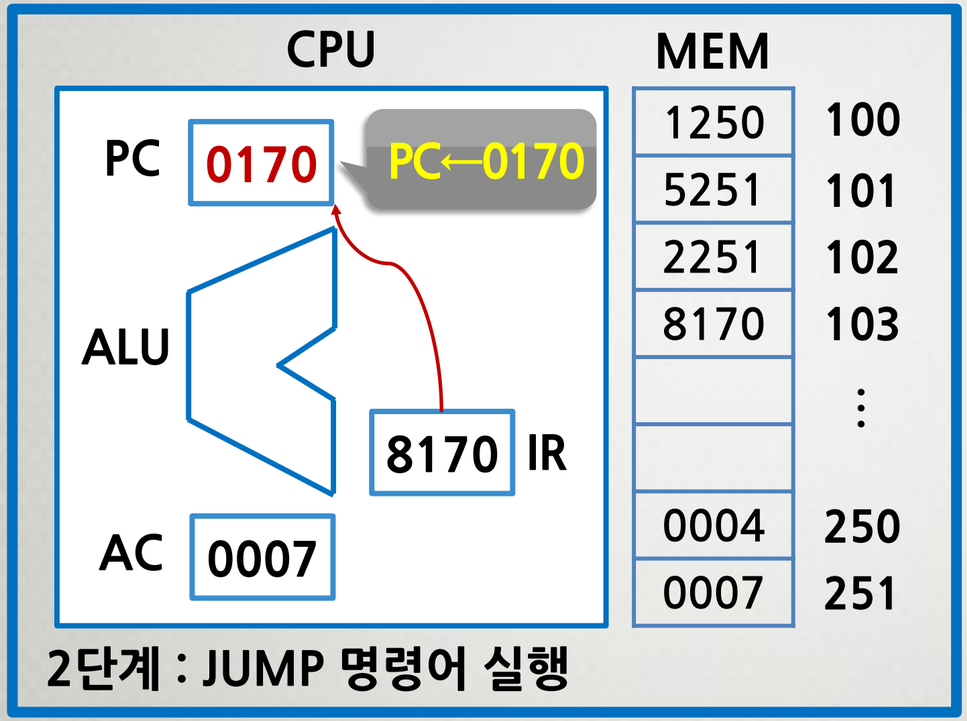
여러 가지 명령어의 종합적인 실행과정(예제)

주소(Address)	명령어(Instruction)	기계 코드(Machine Code)
100	LOAD 250	1250
101	ADD 251	5251
102	STORE 251	2251
103	JUMP 170	8170



여러 가지 명령어의 종합적인 실행과정(예제)

주소(Address)	명령어(Instruction)	기계 코드(Machine Code)
100	LOAD 250	1250
101	ADD 251	5251
102	STORE 251	2251
103	JUMP 170	8170



여러 가지 명령어의 종합적인 실행과정(예제)

- (예제) 프로그램이 아래 표와 같이 작성되었을 때, 실행 시간을 계산하시오.
  (단, CPU 클럭은 2GHz이고, 메모리 지연시간은 없다고 가정)

주소	명령어	간접 사이클(I)	기계 코드
100	LOAD 250	1	1250
101	ADD 251	1	5251
102	STORE 251	0	2251
103	JUMP 170	0	8170

  - 클럭 주기 = 1 sec ÷ 2×10^9 = 0.5ns
  - 전체 소요 클럭 수 : 28 클럭
    - LOAD : 인출(3) + 간접(3) + 실행(3) = 9 클럭
    - ADD : 인출(3) + 간접(3) + 실행(3) = 9 클럭
    - STORE : 인출(3) + 간접(0) + 실행(3) = 6 클럭
    - JUMP : 인출(3) + 간접(0) + 실행(1) = 4 클럭
  - 전체 실행 시간 : 0.5 ns × 28 클럭 = 14.0 ns