

예제

| | | | | | |
|--|--|--|--|--|--|
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

번역

어셈블러 → 목적파일

① 저급언어 (기계어, 어셈블리어)

② 고급언어 (C, C++, ...)

번역 (컴파일러, 인터프리터) → 목적파일

* 연산 코드

① 산술 / 논리 연산 : ADD, AND, OR, NOT, COMPARE

② 데이터 전송 : MOVE, STORE, LOAD, PUSH, POP

③ 제어 흐름 변경 (분기문, 조건문) :

JUMP, CONDITIONAL JUMP, CALL, RETURN, HALT

돌아올 주소를 저장한 후 진행

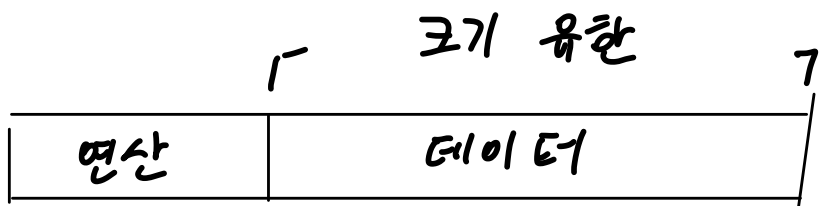
④ 입출력 제어 : READ, WRITE, START I/O, TEST I/O

| | |
|-------|------|
| 연산 코드 | 오퍼랜드 |
|-------|------|

명령어 주소 지정 방식

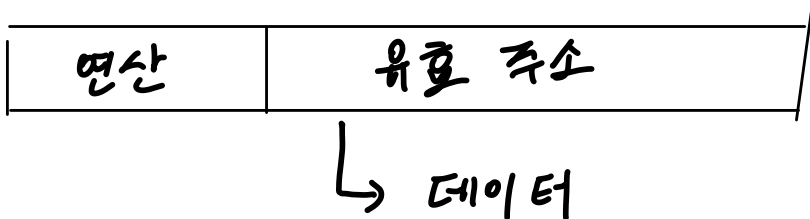
① 즉시 주소 지정 방식

- 간단 연산, 상수 처리



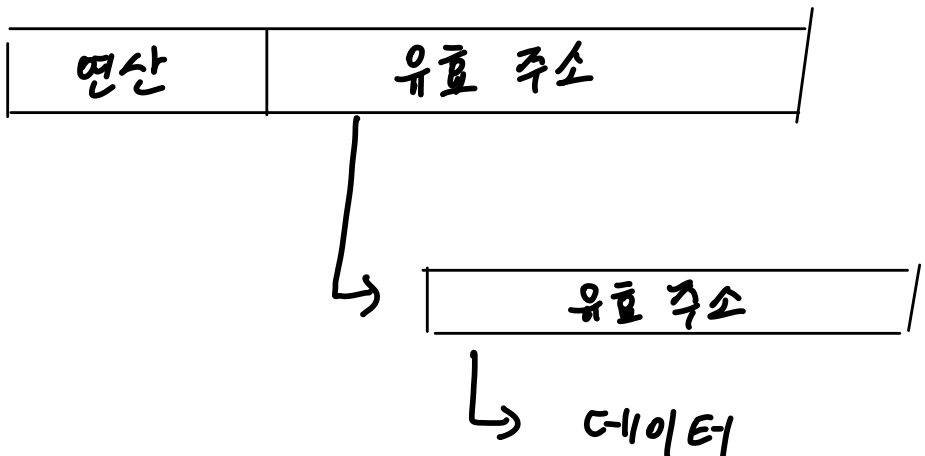
② 직접 주소 지정 방식

- 메모리의 데이터 사용



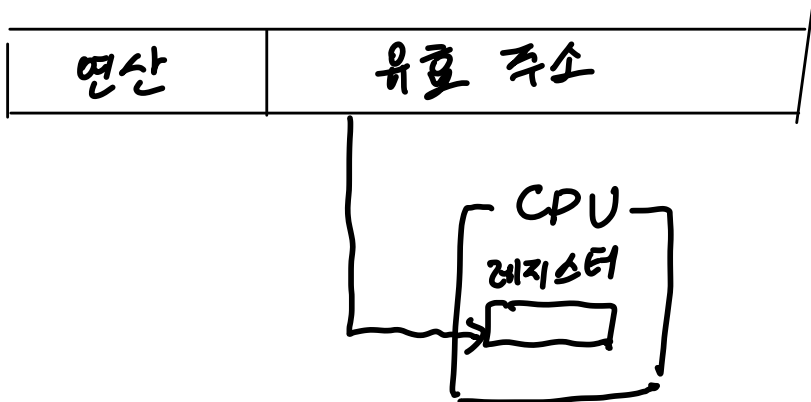
③ 간접 주소 지정 방식

- 포인터, 동적 할당

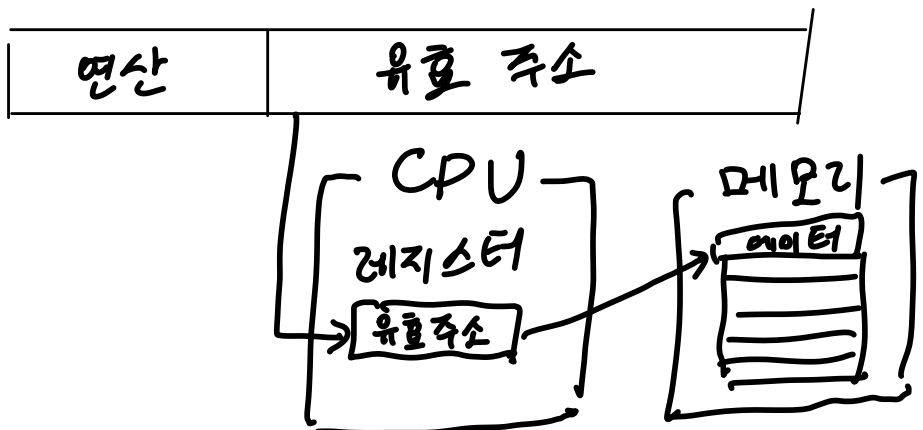


④ 레지스터 주소 지정 방식

- 속도는 빠르지만, 자원이 높음

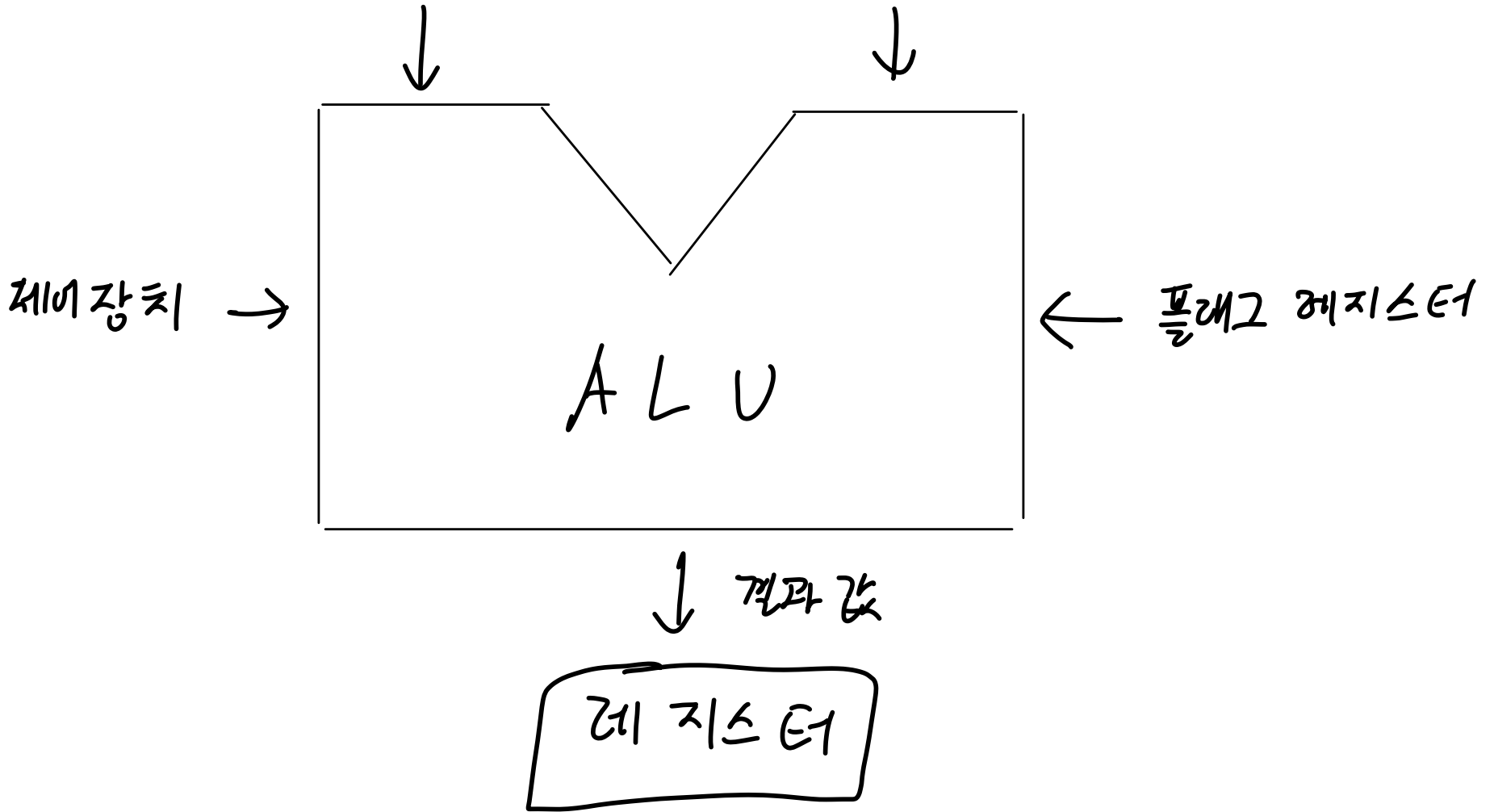


⑤ 레지스터 간접 주소 지정 방식



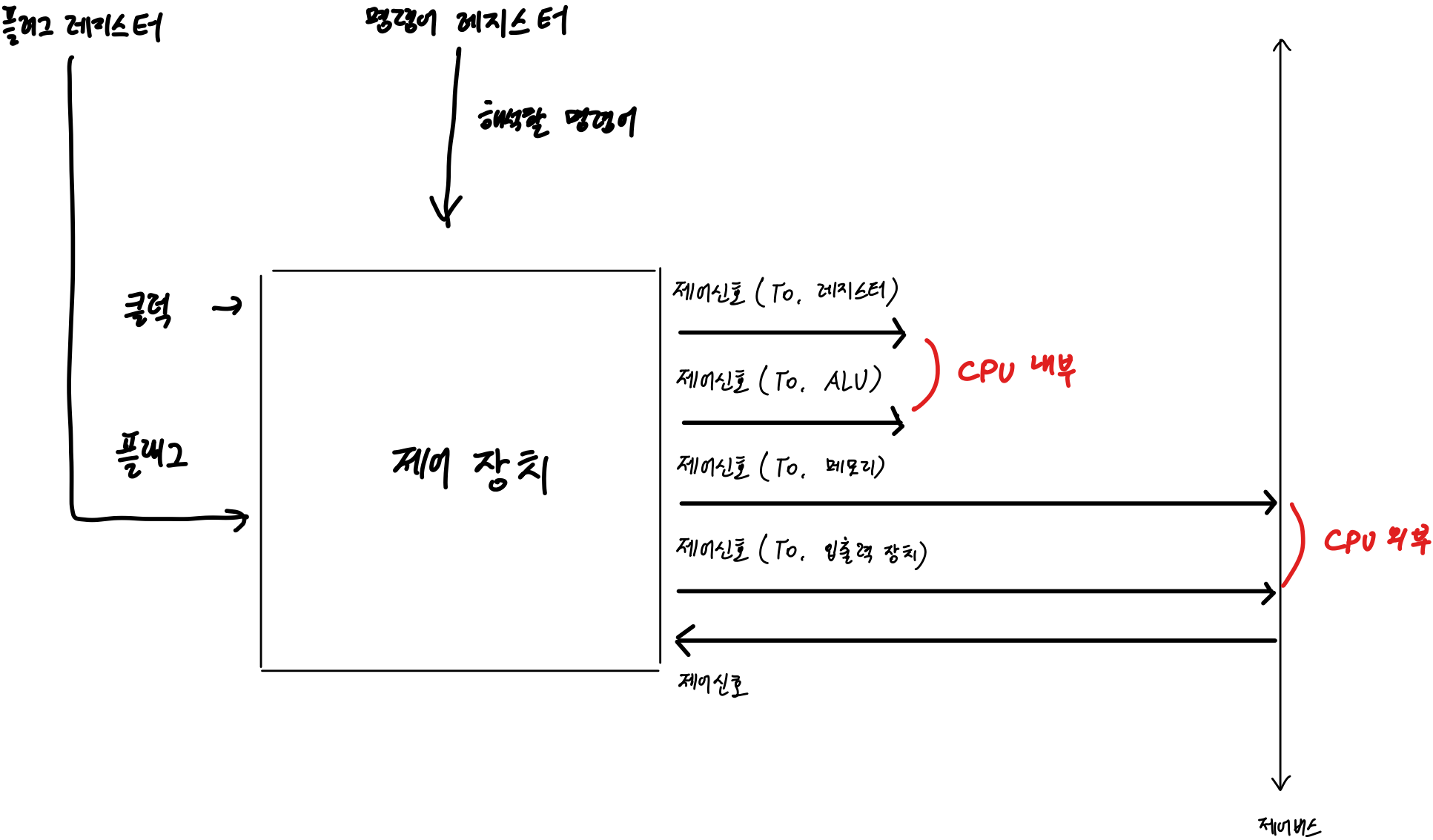
레지스터 피연산자

레지스터 피연산자



* 플래그 레지스터

- ① 부호 플래그
 - 0 양수
 - 1 음수
- ② 제로 플래그
 - 0 결과값이 0이 아니다
 - 1 " 0이다
- ③ 캐리 플래그
 - 0 올림수 발생 X
 - 1 올림수 발생 O
- ④ 오버플로우 플래그
 - 0 오버플로우 발생 X
 - 1 오버플로우 발생 O
- ⑤ 인터럽트 플래그
 - 0 인터럽트 가능 X
 - 1 인터럽트 가능 O
- ⑥ 슈퍼바이저 플래그
 - 0 현재 사용자 모드
 - 1 현재 커널 모드



① 프로그램 카운터 PC

- 메모리에서 읽어들이 명령어 주소 저장후 다음에 실행될 명령어 위치 가르킴

- 분기/조건 문

② 명령어 레지스터 IR

- 현재 실행 중인 명령어 저장

③ 메모리 주소 레지스터 MAR

- CPU가 메모리에 데이터로 형식 | 여기 저장된 주소를 주소버스로 메모리에 전달하여 가져옴

④ 메모리 버퍼 레지스터 MBR

- CPU가 메모리에서 읽기또는 쓰기 한 데이터가 저장됨

⑤ 플레그 레지스터

- ALU 연산 결과에 대한 상태를 저장

- 제어 결정에 이용

⑥ 범용 레지스터

- 프로그램이 수행하는 연산에 활용되는 임시 저장소

스택 주소 지정 방식

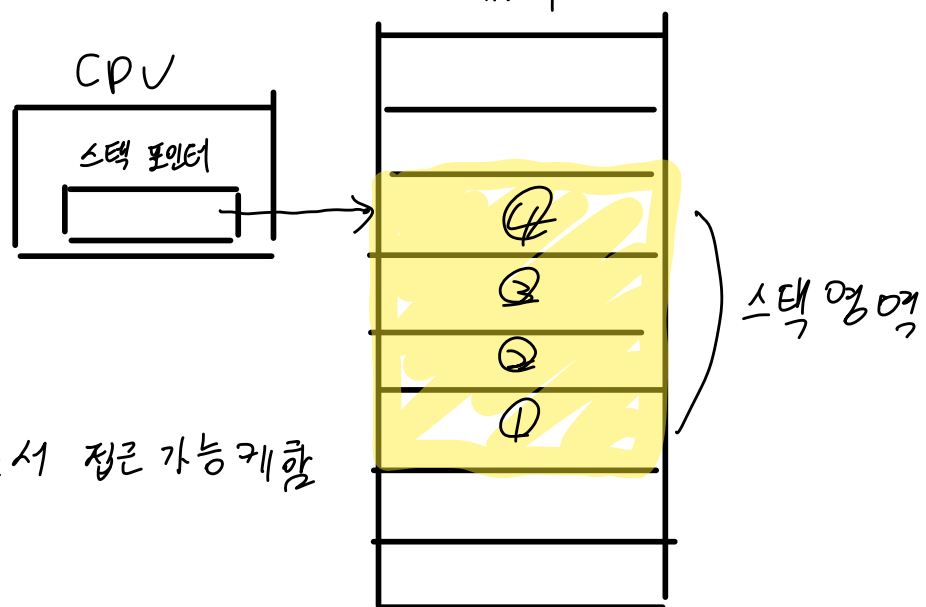
- 스택과 스택 포인터를 이용

함수의 로컬 변수, 복귀 주소 스택에 저장

(스택 포인터)

함수의 로컬 변수, 복귀 주소 스택에 저장 이를 가르킴으로서 접근 가능케 함

증감 → 데이터 후속어 포함

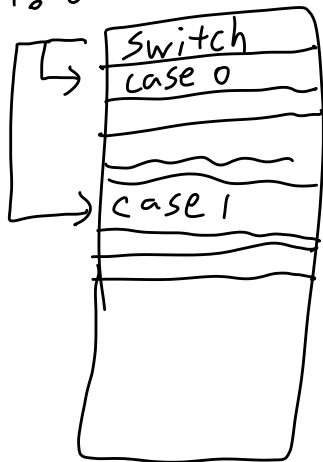


변위 주소 지정 방식

□ 상대 주소 지정 방식

- 오퍼랜드와 프로그램 카운터 (현재 실행 중인 명령어의 주소)의 값을 더하여 유효 주소 얻는 방식

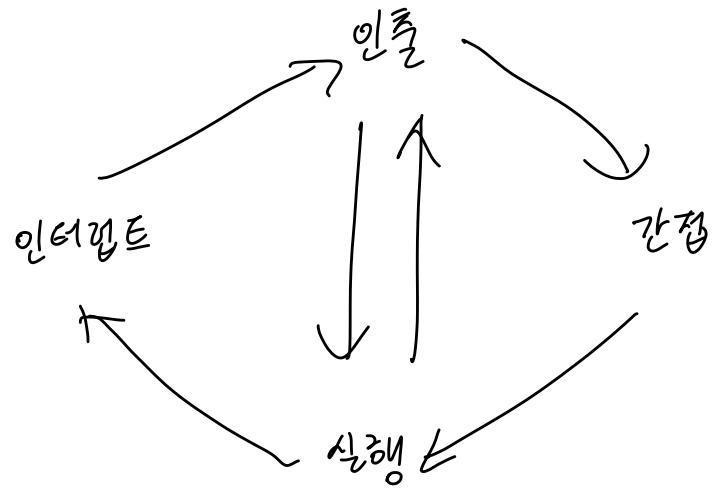
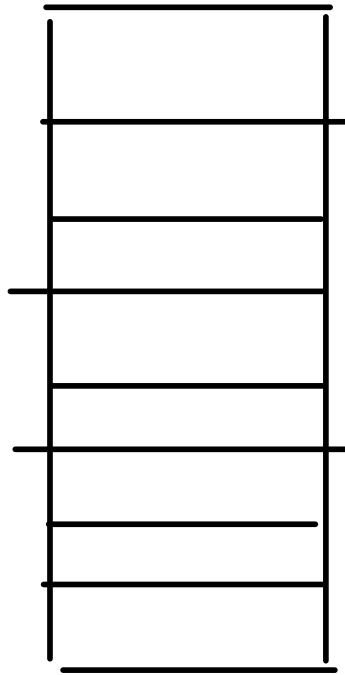
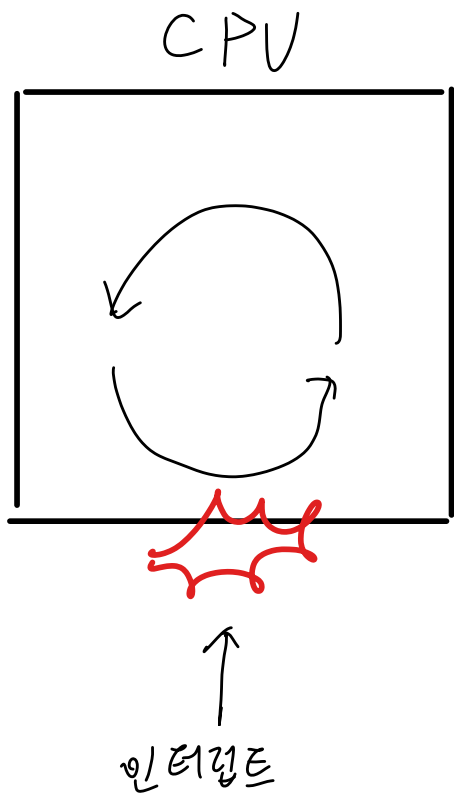
ex) 이 이동정도를 switch 기준으로 오퍼랜드 값으로 상대적으로 지정



□ 베이스 레지스터 주소 지정 방식

- 오퍼랜드와 베이스 레지스터 값을 더하여 유효주소 얻는 방식

메모리



* 명령어 사이클

① 인출 사이클

메모리에 있는 명령어를 CPU로

② 실행 사이클

CPU로 가져온 명령어 실행

제어 장치가 레지스터 값 해석 후 제어 신호 발생

③ 간접 사이클

인출과 실행 사이클 간 동작 중

메모리 접근이 더 필요할 경우

* 인터럽트

① 동기 인터럽트

- 오류, 예외 처리 상황

- 해결 후 작업 이어 나가야 하는 경우

② 비동기 인터럽트

- 실행 하고 계속 보고 있지 않아도 될 때

ex) 프린터로 인쇄 실행

→ 완료 시점에 알림 받음

→ 후 처리

