

ch6 메모리와 캐시 메모리

06-1 RAM의 특징과 종류

RAM의 특징

- 실행할 프로그램의 명령어와 데이터가 저장됨
- 전원을 끄면 저장된 내용이 사라짐 : **휘발성 저장 장치**

cf) 보조기억장치인 비휘발성 저장 장치 ← 보관할 대상 저장

cf) 휘발성 저장 장치인 RAM ← 실행할 대상 저장

“CPU가 실행하고 싶은 프로그램이 보조기억장치에 있다면 이를 RAM으로 복사하여 저장한 뒤 실행”

RAM의 용량과 성능

- RAM 용량이 적으면
 - 보조기억장치에서 실행할 프로그램을 가져오는 일이 잦아 실행 시간이 길어짐
- RAM용량 크면 많은 프로그램들을 동시에 빠르게 실행하는 데 유리
 - But, 용량이 필요 이상으로 커졌을 때 속도가 그에 비례하여 증가하지는 않음.

RAM의 종류

DRAM(Dynamic RAM)

: 저장된 데이터가 동적으로 변하는(사라지는) RAM

- (시간이 지나면 저장되니 데이터가 점차 사라짐)
- 데이터 소멸 막기 위해 일정 주기로 데이터를 재활성화해야함

SRAM(Static RAM)

: 저장된 데이터가 변하지 않는 RAM

- 재활성화 필요x
- DRAM보다 속도 빠름

****근데 사용되는건 DRAM**

왜?? SRAM은 집적도 낮고, 소비전력 크고, 가격도 더 비싸기 때문

	DRAM	SRAM
재충전	필요함	필요 없음
속도	느림	빠름
가격	저렴함	비쌈
집적도	높음	낮음
소비 전력	적음	높음
사용 용도	주 기억장치(RAM)	캐시 메모리

SDRAM(Synchronous Dynamic RAM)

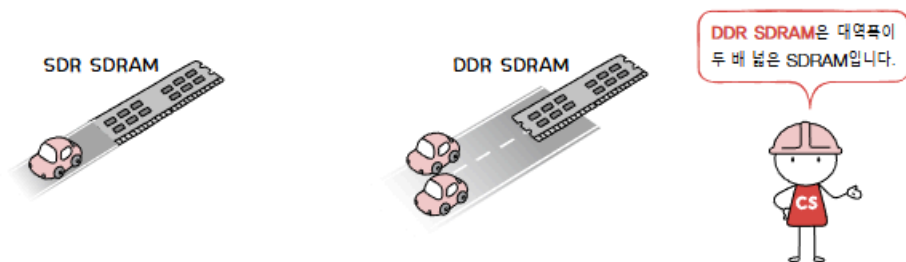
: 클럭 신호와 동기화된, 발전된 형태의 DRAM

(클럭타이밍에 맞춰 CPU와 정보를 주고받을 수 있음을 의미함.)

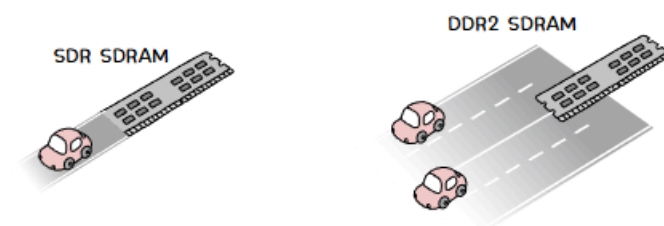
DDR SDRAM(Double Data Rate SDAM)

가장 흔히 사용되는 RAM

- 대역폭을 넓혀 속도를 빠르게 만든 SDRAM
 - DDR2 SDRAM은 DDR SDRAM보다 두 배 넓음



DDR2 SDRAM은 DDR SDRAM보다 대역폭이 두 배 넓은 SDRAM입니다. 다시 말해, DDR2 SDRAM은 SDR SDRAM보다 너비가 네 배 넓은 도로와도 같습니다.



06-2 메모리의 주소 공간

물리 주소와 논리 주소

물리 주소 : 하드웨어 상 주소

논리 주소 : 실행 중인 프로그램 각각에게 부여된 0번지부터 시작되는 주소를 의미

** CPU가 이해하는 주소가 논리 주소라고는 해도 CPU가 메모리와 상호작용하려면 논리주소와 물리 주소간의 변환이 이루어져야 함.(forCPU와 메모리간 상호작용)

논리주소와 물리 주소 간 변환 : MMU(Memory Management Unit)에 의해 수행됨

- MMU는 CPU가 발생시킨 논리 주소에 베이스 레지스터 값을 더하여 논리 주소를 물리 주소로 변환함
- **베이스 레지스터** : 첫 물리주소 저장(프로그램의 가장 작은 물리 주소)
- **논리 주소** : 프로그램의 시작점으로부터 떨어진 거리

메모리 보호 기법

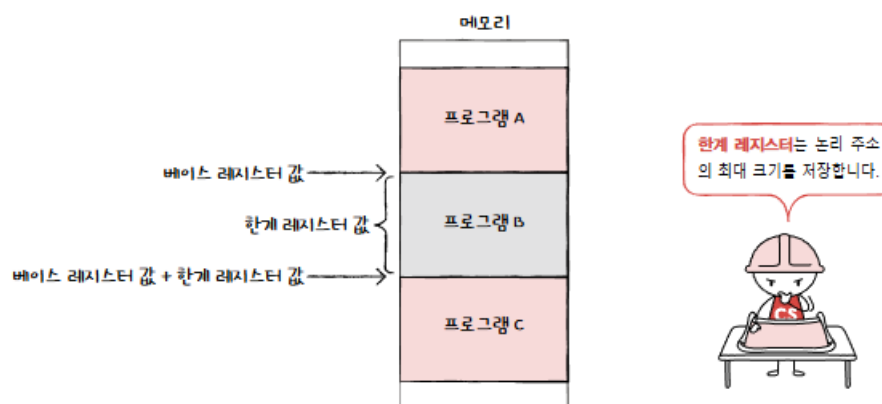
다른 프로그램의 영역을 침범할 수 있는 명령어 위험 : 논리 주소 범위를 벗어나는 명령어 실행 방지, 실행 중인 프로그램이 다른 프로그램에 영향 받지 않도록 보호할 방법이 필요

→ **한계 레지스터** : 논리 주소의 최대 크기를 저장

프로그램의 물리 주소 범위

:

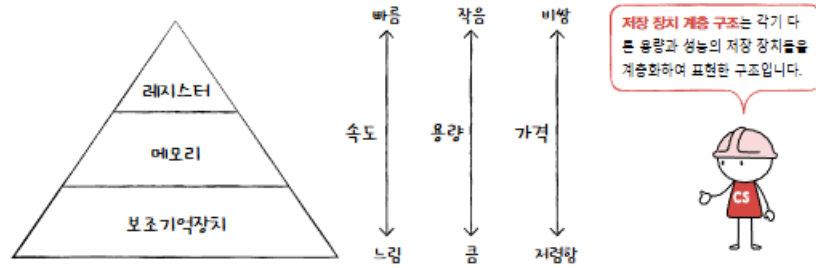
베이스 레지스터 값 이상 ~ 베이스 레지스터 값 + 한계 레지스터 값 미만 이 됨.



06-3 캐시 메모리

저장 장치 계층 구조 : CPU에 얼마나 가까운가를 기준으로

- CPU와 가까운 저장 장치는 빠르고, 멀리 있는 저장 장치는 느림
- 속도가 빠른 저장 장치는 저장 용량이 작고, 가격이 비쌈

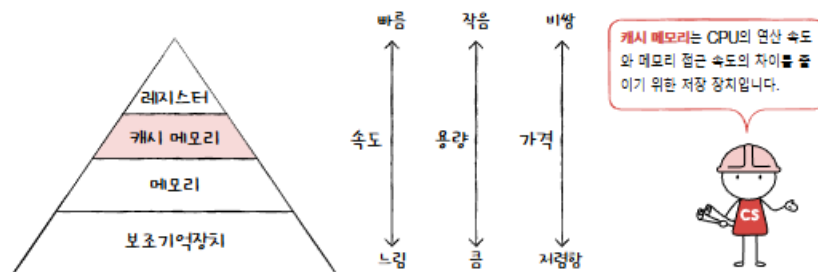


캐시 메모리

CPU와 메모리 사이에 위치, 레지스터보다 용량이 크고 메모리보다 빠른 SRAM 기반 저장장치

- CPU의 연산 속도와 메모리 접근 속도 차이를 줄이기 위해 나옴

캐시 메모리까지 반영한 저장 장치 계층 구조는 아래와 같이 그릴 수 있습니다.



- 코어(CPU)에 가까운 순서대로 계층 구성

- 가까운 캐시 메모리 L1(level 1) 캐시
- 그 다음 L2 캐시, L3 캐시..

** 일반적으로 L1, L2 캐시는 L3 캐시는 코어 외부에 위치해 있음

캐시 메모리 용량 : $L1 > L2 > L3$

캐시 메모리 속도 : $L3 > L2 > L1$

캐시 메모리 가격 : $L3 > L2 > L1$

- CPU 메모리 내 데이터가 필요하다고 판단하면 L1캐시부터 찾아보고, 없으면 L2,L3 캐시 순으로 데이터 검색

참조 지역성 원리

- **캐시 히트** : 자주 사용될 것으로 예측한 데이터가 실제로 들어맞아 캐시 메모리 내 데이터가 CPU에서 활용될 경우
- **캐시 미스** : 자주 사용될 것으로 예측하여 캐시 메모리에 저장했지만 예측이 틀려 메모리에서 필요한 데이터를 직접 가져와야 하는 경우
- **캐시 적중률** : 캐시 히트 횟수 / (캐시 히트 횟수 + 캐시 미스 횟수)
 - 우리가 사용하는 컴퓨터는 이게 85~95%정도..

“캐시 메모리는 참조 지역성의 원리에 따라 메모리로부터 가져올 데이터 결정”

- **참조 지역성의 원리?** CPU가 메모리에 접근할 때의 주된 경향을 바탕으로 만들어진 원리
 - **시간 지역성** : cpu는 최근에 접근했던 메모리 공간에 다시 접근하려는 경향
 - **공간 지역성** : CPU는 접근한 메모리 공간 근처를 접근하려는 경향