


06.

메모리와
캐시 메모리

전소은!

발표!!

06-1 RAM의 특징과 종류

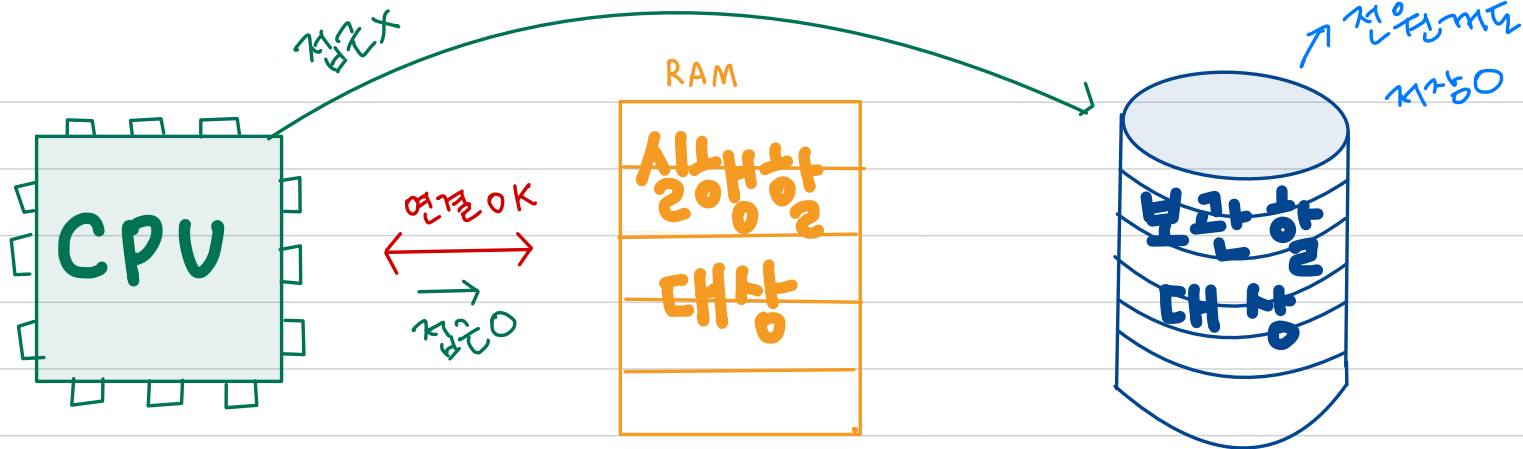
RAM의 특징

실시간 데이터의 명령어와 데이터가 저장됨

★ 전원을 끄면 RAM에 저장된 명령어와 데이터가 모두 날아감 ★ ⇒ 휘발성 저장장치

전원이 꺼져도 저장된 내용이 유지

↕
비휘발성 저장장치 ex) 하드디스크, SSD, CD-ROM,
USB 메모리



근데 이
과제

!!
자정자
명령1 데려와!

↑ 휘발성 저장 장치

!! 우리들...

누가 할래?
명령1

↑ 비휘발성 저장 장치

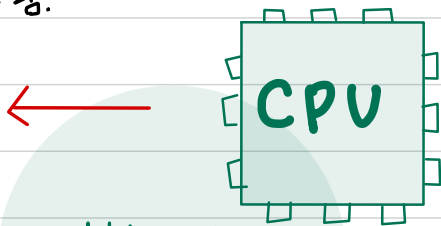
!! GPT?...

저장공간
명령1

RAM의 용량과 성능

RAM 용량 적음 = 시간 지체됨.

속도 ↓

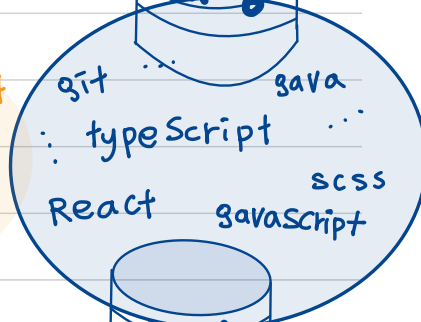


느리게 할게!

깃...?



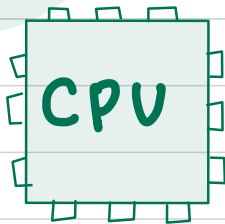
깃, typeScript
React, scss,
sava ...
RAM



깃 ... sava
typeScript ...
React savaScript



속도 ↑



↳ 그러나 한계 존재!

RAM의 종류

DRAM

Dynamic RAM의 준말

저장된 데이터가 동적으로 사라지는, 변하는 RAM

= 데이터의 소멸을 막기 위해 일정 주기로 데이터 재활성화 (다시 저장)해야 함

일반적인 메모리로서 사용됨

비교적 낮은 소비전력, 저렴, 직접도 높음 → 대용량 설계 용이

SRAM

Static RAM

정적의

) 저장된 데이터가 변하지 않는 RAM

DRAM보다 속도 빠름

메모리로 사용 X

직접도 낮음, 소비전력 큼, 가격 비쌈

캐시 메모리에 사용

SDRAM

클럭 신호와 동기화된, 발전된 형태의 DRAM
↳ 타이밍에 맞춰서 CPU와 정보를 주고 받을 수 있음

DDR SDRAM

가장 흔히 사용되는 RAM

대역폭을 넓혀 속도를 빠르게 만든 SDRAM

↳ 데이터를 주고 받는 길의 너비

한 클럭에 두 번씩 CPU와 데이터 주고 받음

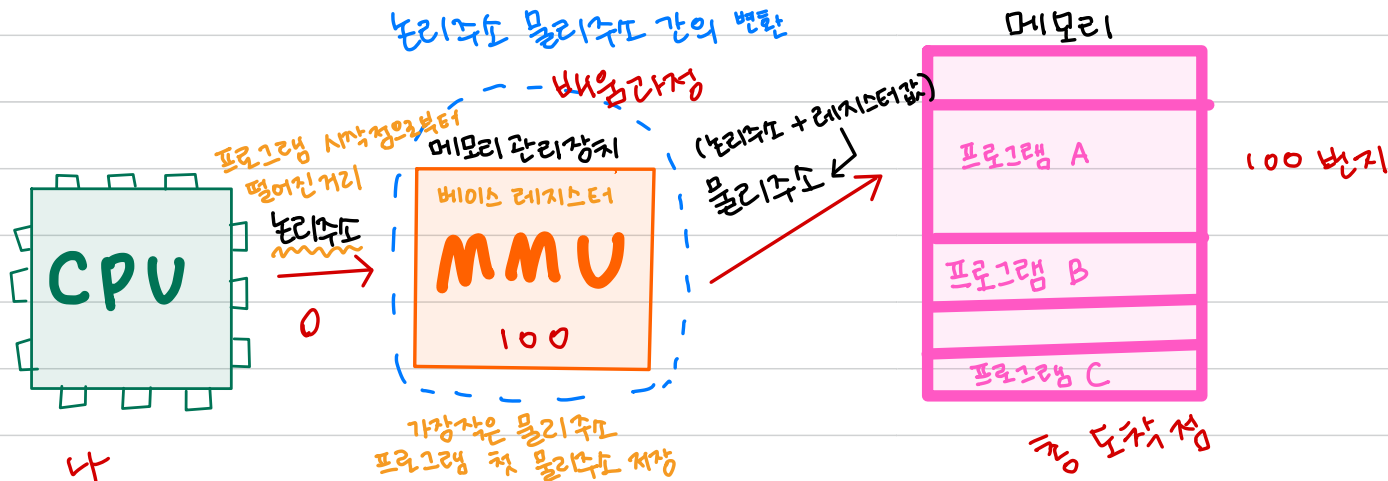
전용속도 2배 빠름

06-2 메모리의 주소공간

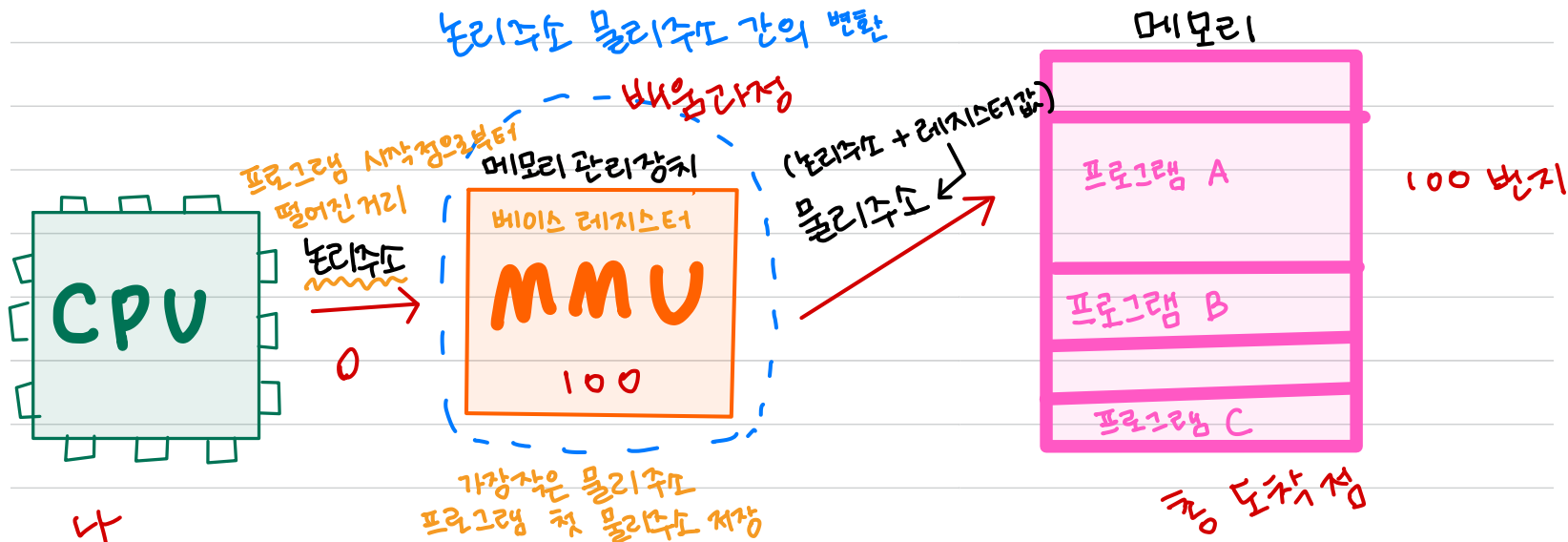
물리주소와 논리주소

물리주소: 정보가 실제로 저장된 하드웨어상의 주소 메모리가 사용하는 주소

논리주소: 실행 중인 프로그램 각각에 부여된 0번지 부터 시작되는 주소 CPU, 실행 중인 프로그램이 사용하는 주소

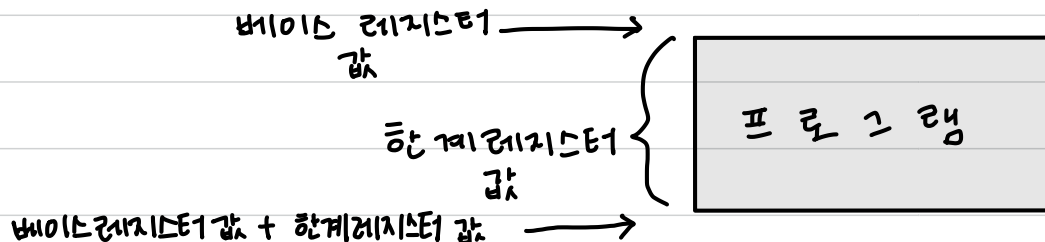


메모리 보호 기법



한계 레지스터

논리주소의 최대 크기 저장



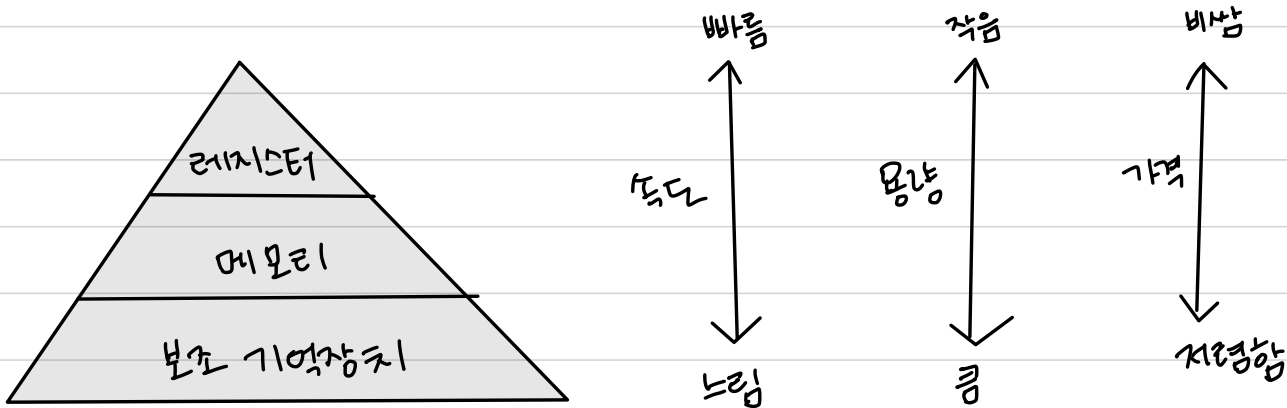
* CPU는 메모리 접근 전 접근하려는 논리주소가 한계 레지스터보다 작은지를 항상 검사

만약 CPU가 한계 레지스터보다 높은 논리주소에 접근하려고 하면 인티드랩 발생시켜 실행 중단

캐시 메모리

저장 장치 계층구조

컴퓨터가 사용하는 저장장치들은 'CPU에 얼마나 가까운가'를 기준으로 계층적으로 나타낼 수 있음.

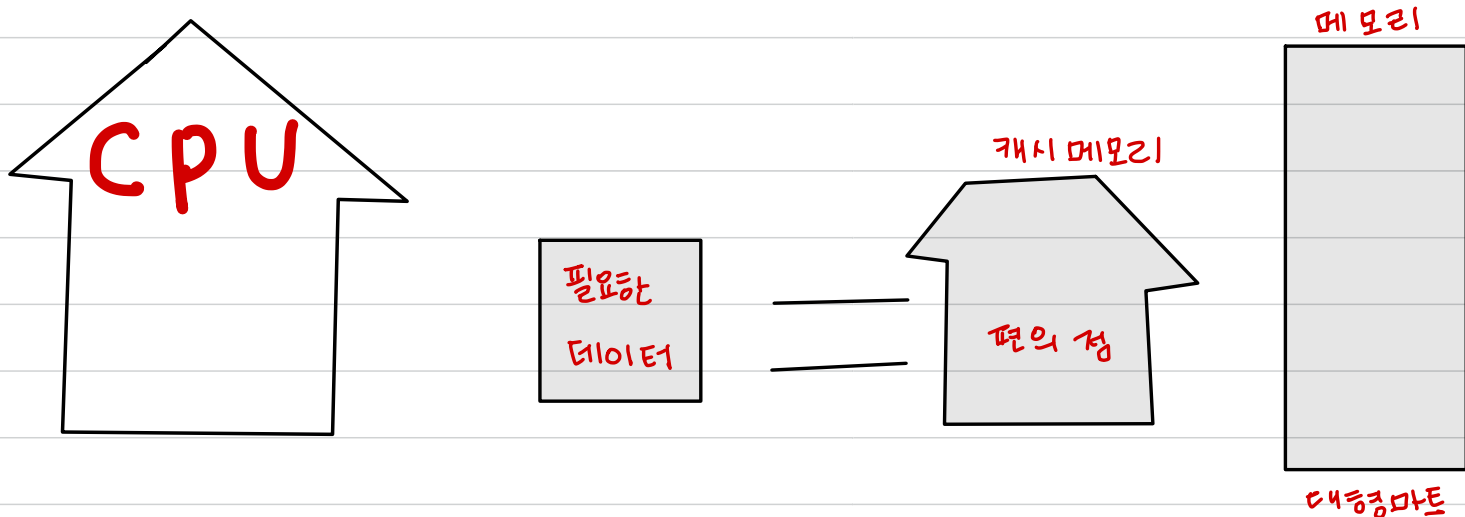


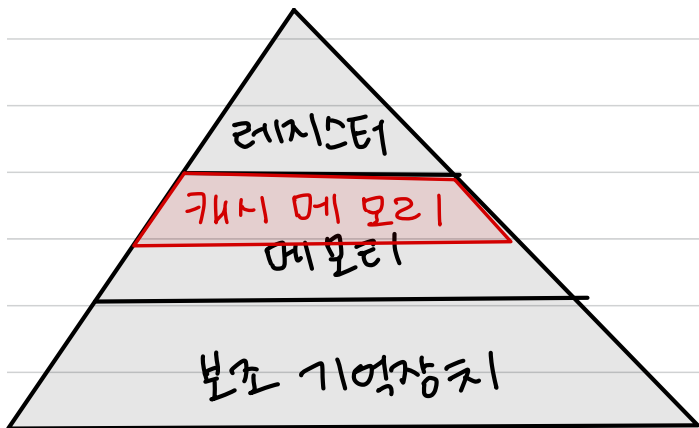
캐시 메모리

CPU와 메모리 사이 위치

레지스터보다 용량이 크고 메모리보다 빠른 SRAM 기반의 저장장치

존재이유: CPU의 연산속도와 메모리 접근 속도 차이를 줄이기 위해



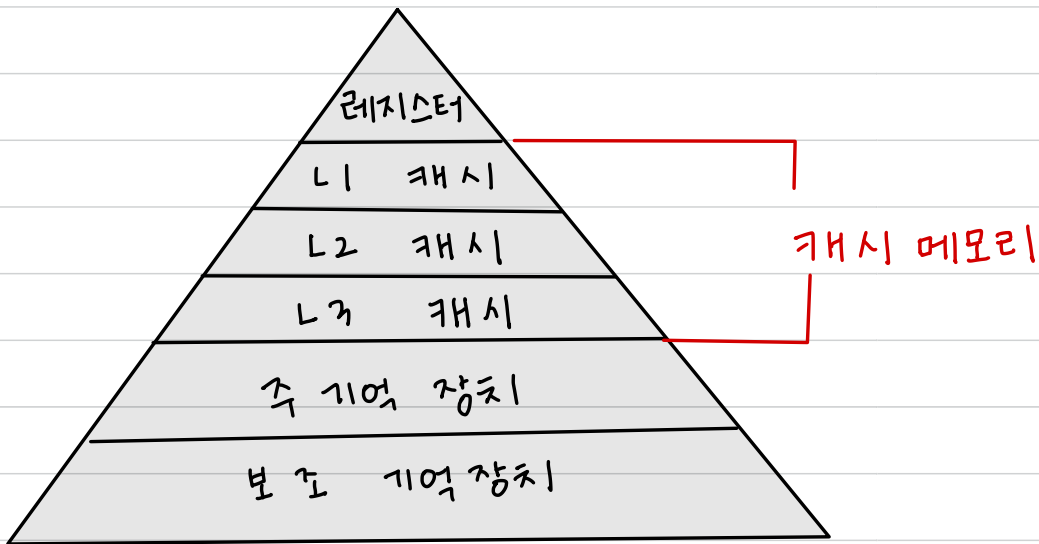


빠름
↑
속도
↓
느림

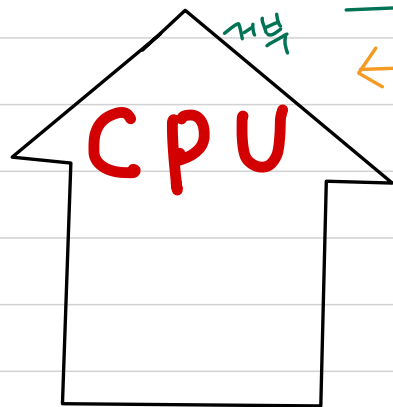
작음
↑
용량
↓
큼

비쌌음
↑
가격
↓
저렴함

캐시 메모리 계층 구조



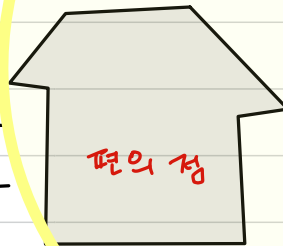
참조 지역성 원리



재시미스①

재시히트
적중률: 재시적중률

필요한
데이터



CPU가 사용할 대상
예측하여 저장
↑

캐시 메모리

메모리



대형마트

② // 캐시메모리 미점 활용 X
∴ 성능 저하

캐시메모리 용량 < 메모리 용량
메모리의 모든 내용 저장 X
일부만 복사하여 저장 O

참조 지역성의 원리

CPU가 메모리에 접근할 때의 주된 경향을 바탕으로 만들어진 원리

- 1 CPU는 최근에 접근했던 메모리 공간에 다시 접근하려는 경향 - 시간 지역성
- 2 CPU는 접근한 메모리 공간 근처를 접근하려는 경향 - 공간 지역성