

# RAM과 캐시 메모리

RAM, 논리 주소와 물리 주소 그리고 캐시 메모리

OS-Challenge 4주 차 - 심수현

# INDEX

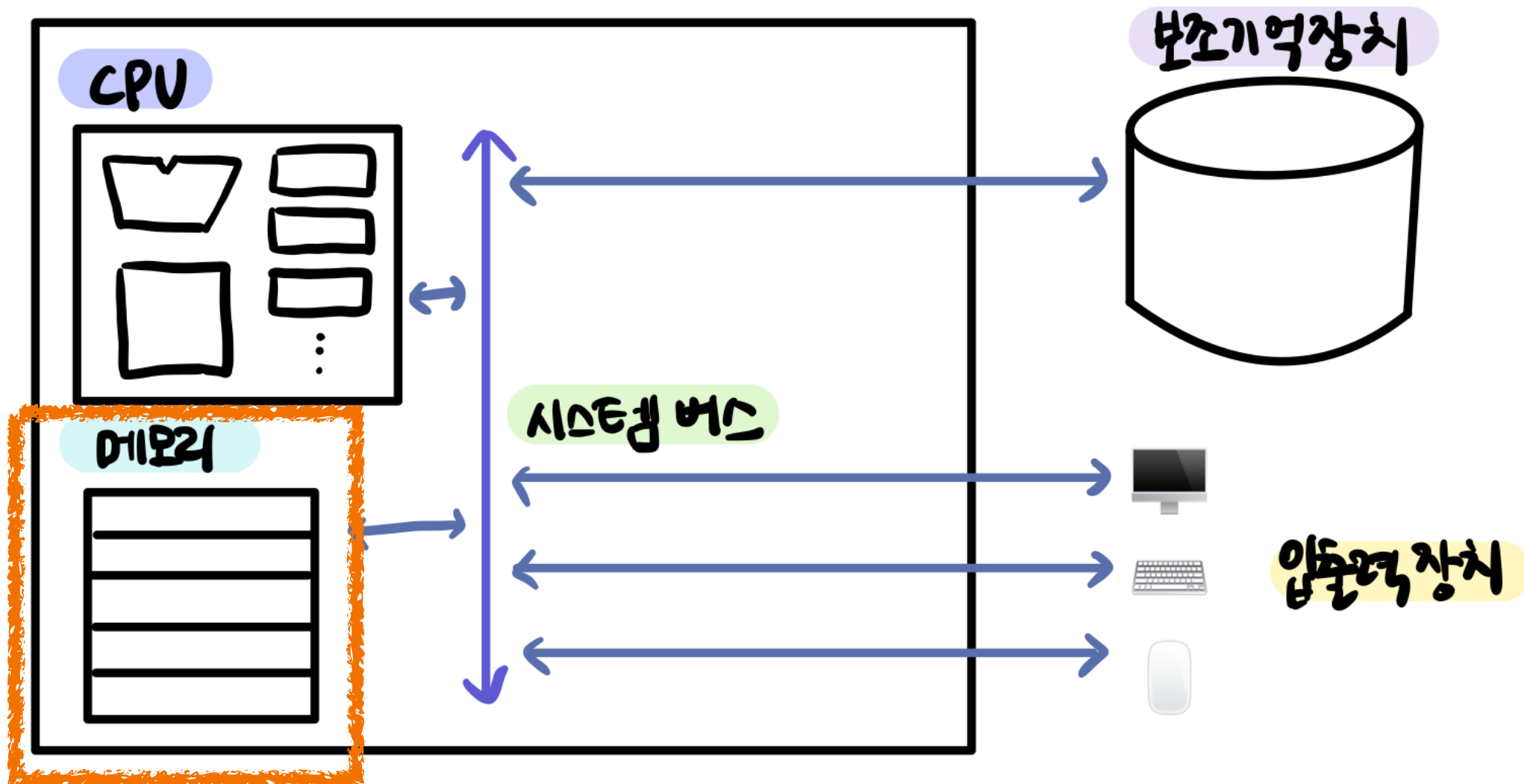
- 컴퓨터의 4가지 핵심
  - 중앙처리장치(CPU) / 주기억장치(메모리) / 보조기억장치 / 입출력장치
- 주기억장치
  - ROM/RAM
- RAM
  - 특징, 용량, 성능, 종류
- 물리 주소와 논리 주소
  - 주소 변환 예제
- 캐시 메모리
  - 저장 장치 계층 구조
  - 캐시 읽기, 쓰기, 더티 플래그
  - 참조 지역성 원리
- 퀴즈

# 컴퓨터의 4가지 핵심

중앙처리장치(CPU) / 주기억장치(메모리) / 보조기억장치 / 입출력장치

Q. 컴퓨터를 구성하는 핵심 부품을 그려보세요!

메인보드



- 현재 실행되는 프로그램의 명령어와 데이터를 저장하는 부품
  - 프로그램이 실행되기 위해서는 반드시 메모리에 저장되어 있어야 함
- 컴퓨터가 빠르게 작동하기 위해서는 저장된 메모리 속 명령어와 데이터 값에 빠르고 효율적으로 접근해야 함
  - => 주소의 개념 사용

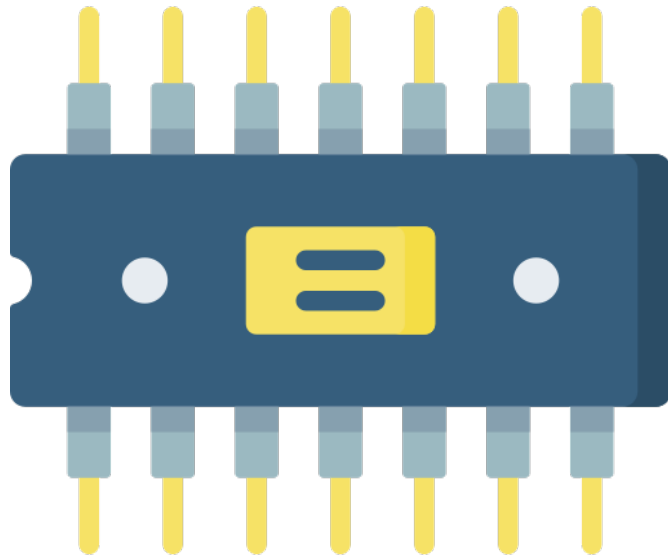
# 주 기억 장치

ROM / RAM

## 주 기억 장치

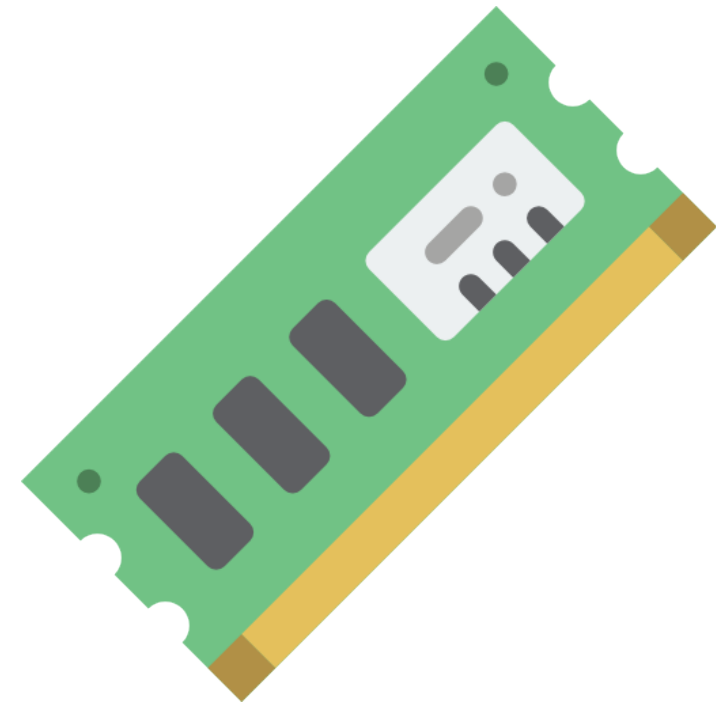
컴퓨터 내부에서 현재 CPU가 처리하고 있는 내용을 저장하고 있는 기억 장치

### ROM(Read Only Memory)



- 전원이 끊어져도 기록된 데이터가 소실되지 않음  
**비휘발성 메모리(Non-Volatile Memory)**
- 데이터를 저장한 후 반영구적으로 사용
- 시스템에 기억시키고 변화시키면 안되는 BIOS 같은 주요 데이터 저장
- *Mask ROM, OTP(One Time PROM), EPROM(Erasable PROM), EEPROM(Electrically Erasable PROM), UVEPROM...*

### RAM(Random Access Memory)

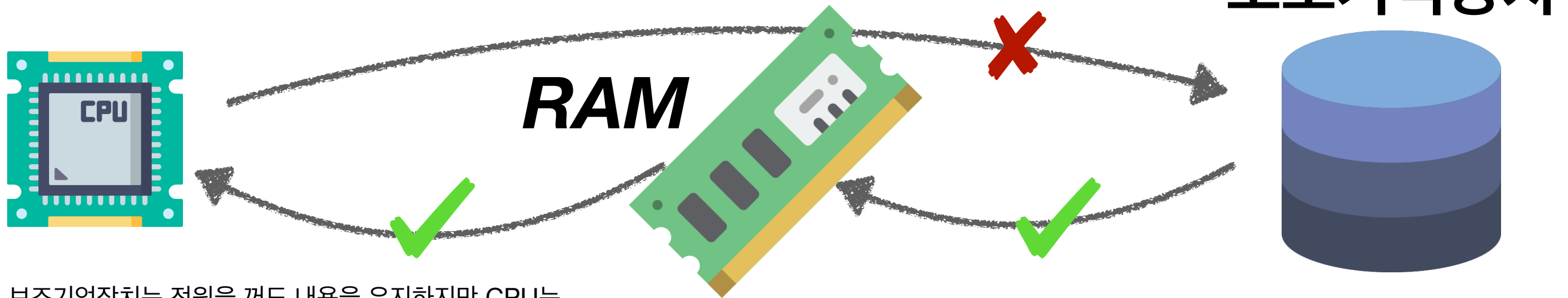


- 읽고 쓰기가 가능, 응용 프로그램, 운영체제 등을 불러와 CPU가 작업할 수 있게 하는 기억 장치
- 전원이 끊어지면 데이터가 전부 지워짐  
**휘발성 메모리(Volatile Memory)**
- 데이터를 읽는 속도와 기록하는 속도가 같고 프로그램을 로딩하거나 데이터를 임시 저장하는 곳에 사용
- *SRAM, DRAM, SDRAM, DDR SDRAM*

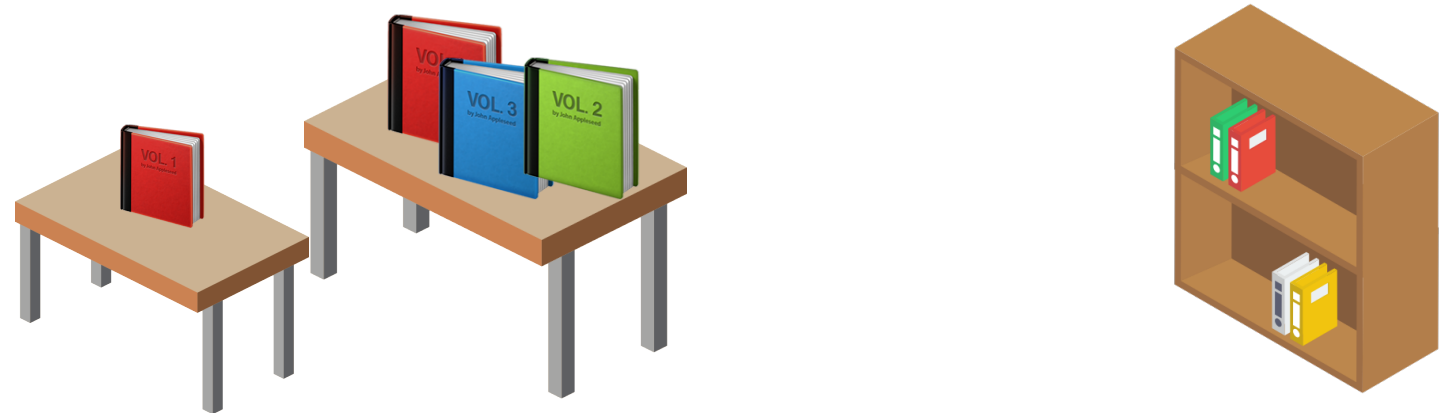
# RAM

특징, 용량, 성능, 종류

## RAM(Random Access Memory)



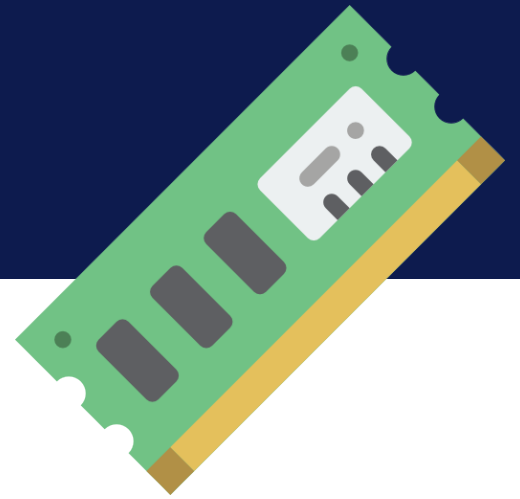
- 보조기억장치는 전원을 꺼도 내용을 유지하지만 CPU는 보조기억장치에 **직접 접근하지 못함**
- 보조기억장치(비휘발성 저장장치)에는 **보관할 대상**을 저장
- RAM(휘발성 저장장치)에는 **실행할 대상**을 저장
- CPU가 실행하고 싶은 프로그램이 보조기억장치에 있으면 RAM에 복사해서 저장 후 실행



- RAM 용량이 **적으면** 보조기억장치에서 실행할 프로그램을 가져 오는 일이 잦아 **실행 시간이 길어짐**
- RAM 용량이 충분히 **크다면** 보조기억장치에서 많은 데이터를 가져와 **미리 RAM에 저장**할 수 있어 여러 번 가져오지 않아도 됨

# RAM

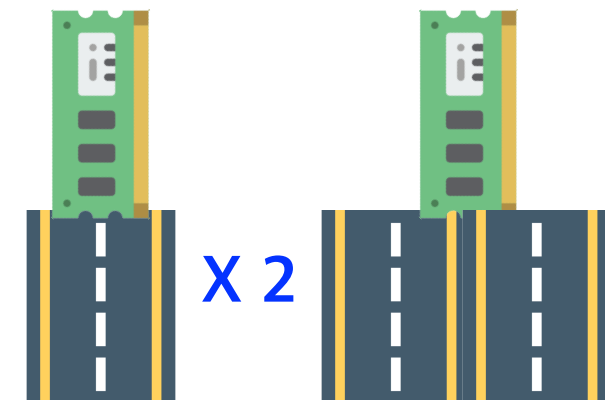
특징, 용량, 성능, 종류



## RAM(Random Access Memory)

	<b>DRAM</b> Dynamic RAM	<b>SRAM</b> Static RAM
재충전	필요함	필요 없음
속도	느림	빠름
가격	저렴함	비쌈
집적도	높음	낮음
소비 전력	적음	높음
사용 용도	주기억장치(RAM)	캐시 메모리

<b>SDRAM</b> Synchronous Dynamic RAM	<b>DDR SDRAM</b> Double Data Rate SDRAM
클럭신호와 동기화된 발 전된 형태의 DRAM (클럭 타이밍에 맞춰 동 작, 클럭마다 CPU와 정 보를 주고받는 DRAM)	가장 흔히 사용하는 RAM, 대역폭(데이터 주고받는 길 너비)을 넓혀 속도를 빠르게 만든 SDRAM



DDR2 SDRAM



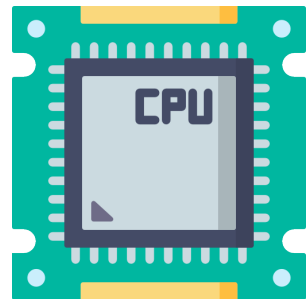
DDR3 SDRAM

...  
X 2 DDR4 SDRAM

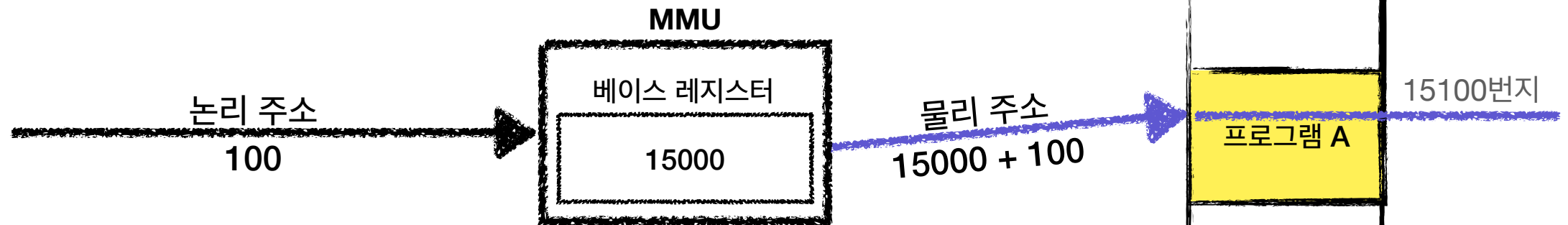
# 물리 주소와 논리 주소

주소 변환 예제

## 주소 변환

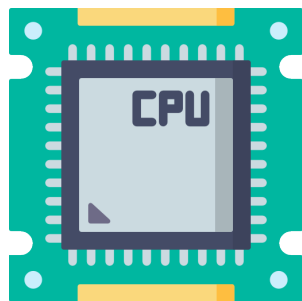


프로그램 A의  
100번지 데이터

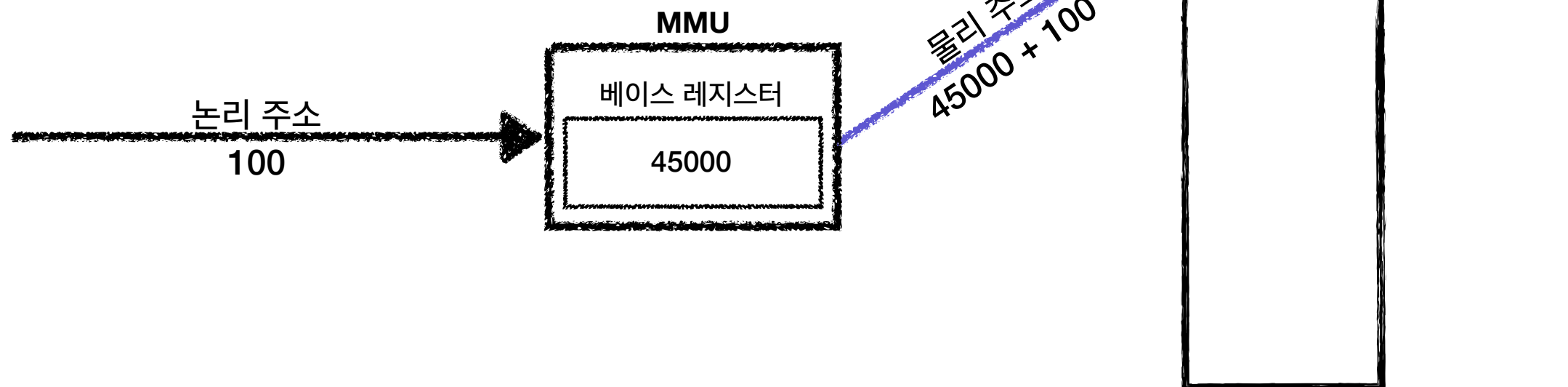


### 베이스 레지스터

- 프로그램의 가장 작은 물리 주소
- 프로그램의 첫 물리 주소를 저장



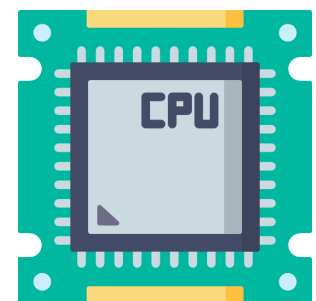
프로그램 C의  
100번지 데이터



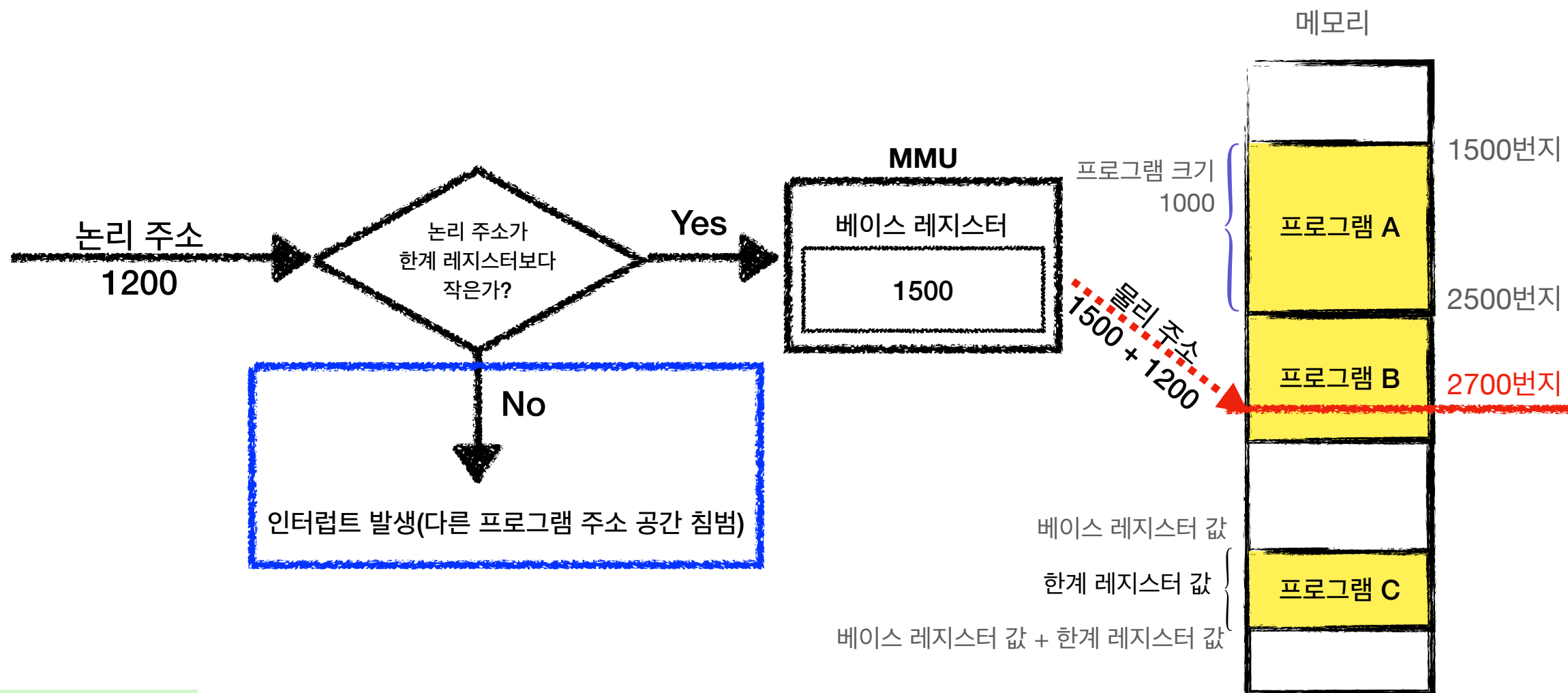
# 물리 주소와 논리 주소

주소 변환 예제

## 주소 변환



프로그램 A의  
1200번지 데이터를  
삭제해라



## 한계 레지스터

- 논리 주소의 최대 크기를 저장
- 다른 프로그램의 영역을 침범할 수 있는 명령어는 위험
- 논리 주소 범위를 벗어나는 명령어 실행 방지
- 실행 중인 프로그램이 다른 프로그램에 영향을 받지 않도록 보호하는 방법

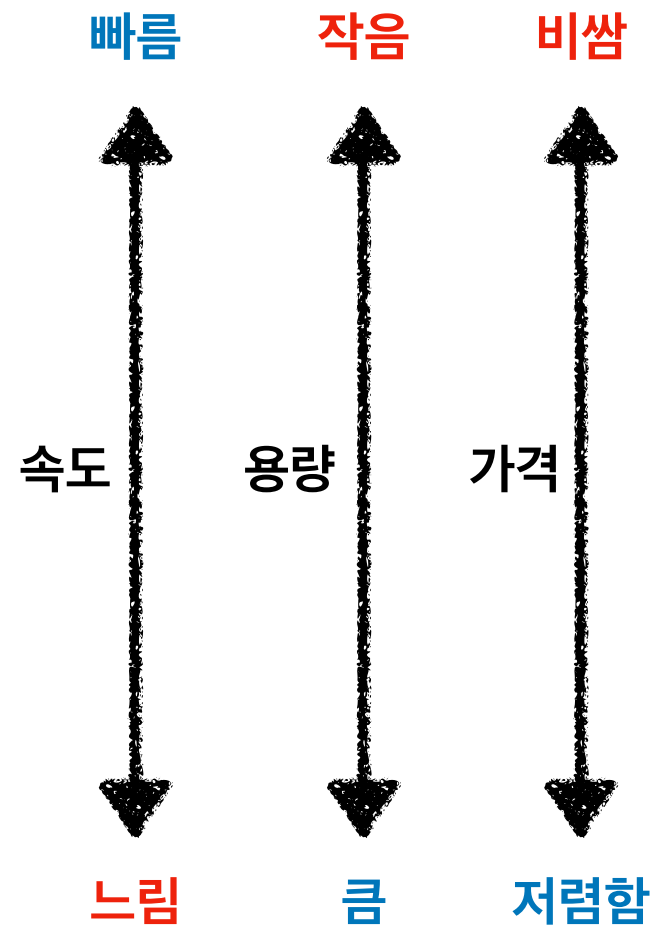
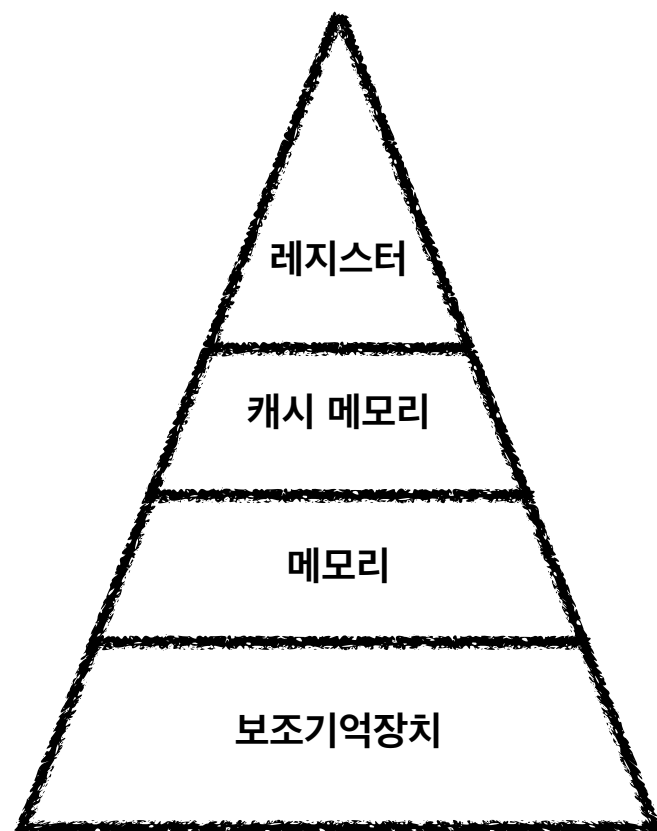
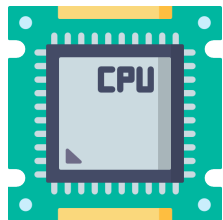


# 캐시 메모리

저장 장치 계층 구조

저장 장치 계층 구조:

“CPU에 얼마나 가까운가”를 기준으로 계층적으로 나타내기

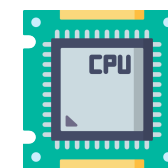


처리 (1): 명령어를 바탕으로 메모리에서 레지스터로 데이터를 읽음

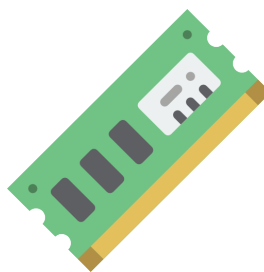
처리 (2): 레지스터에 있는 데이터를 바탕으로 계산

처리 (3): 계산 결과를 메모리에 씀

! 처리 (1) & (3) 병목

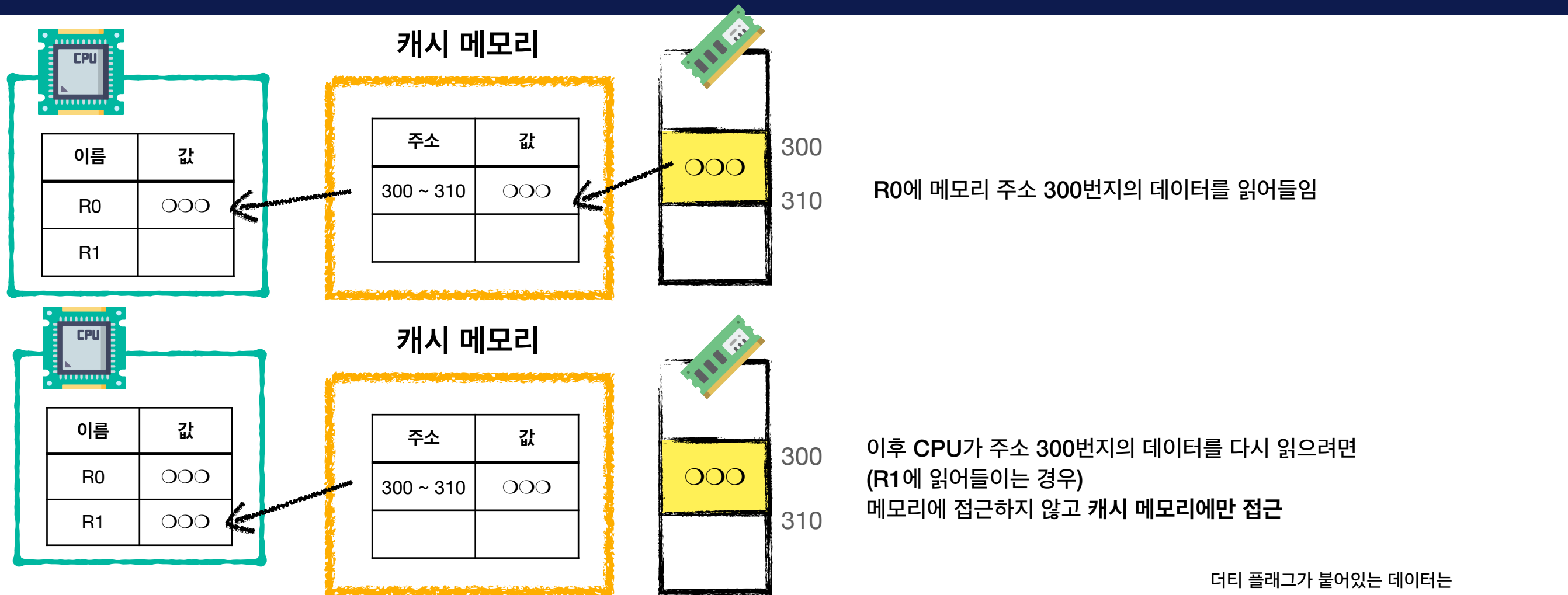


✓  
캐시

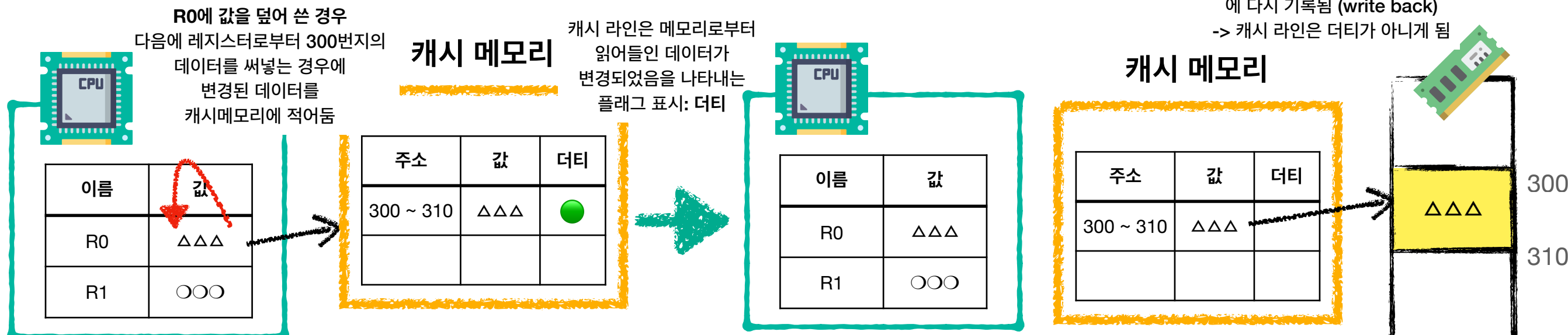


# 캐시 메모리

캐시 읽기, 쓰기, 더티



더티 플래그가 붙어있는 데이터는 나중에 백그라운드 처리로 메모리에 다시 기록됨 (write back)  
-> 캐시 라인은 더티가 아니게 됨



# 캐시 메모리

참조 지역성 원리



## 캐시 히트(Cache Hit)

- 자주 사용될 것으로 예측한 데이터가 실제로 들어맞아 캐시 메모리 내 데이터가 CPU에서 활용될 경우

## 캐시 미스(Cache Miss)

- 자주 사용될 것으로 예측하여 캐시 메모리에 저장했지만 예측이 틀려 메모리에서 필요한 데이터를 직접 가져와야 하는 경우
- 캐시 미스가 자주 발생하면 성능이 떨어짐

## 캐시 적중률(Cache Hit Ratio)

- 캐시가 히트되는 비율
- 캐시 히트 횟수 / (캐시 히트 횟수 + 캐시 미스 횟수)
- 캐시 적중률이 높으면 CPU의 메모리 접근 횟수를 줄일 수 있음

캐시 적중률을 높이기 위해 CPU가 사용할 법한 데이터를 알아내려면?

## 참조 지역성 원리

CPU가 메모리에 접근할 때 주된 경향을 바탕으로 만들어진 원리

### 시간 국소성(시간 지역성; temporal locality)

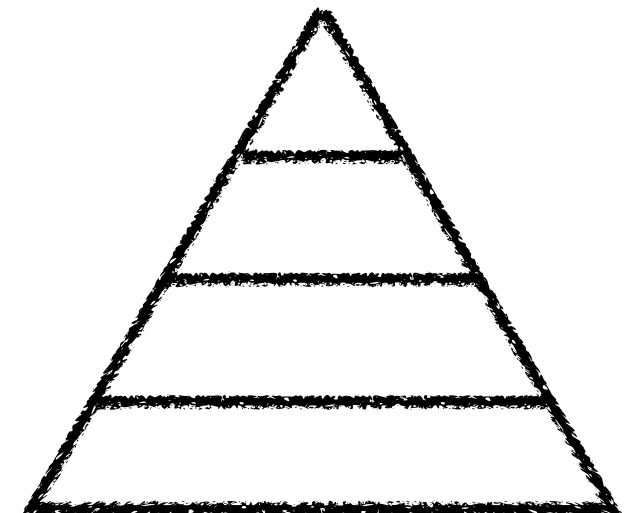
- 최근에 접근했던 메모리 공간에 다시 접근하려는 경향
- CPU는 변수가 저장된 메모리 공간을 언제든지 다시 참조할 수 있음
- CPU는 최근에 접근했던 (변수가 저장된) 메모리 공간을 여러 번 다시 접근할 수 있음

### 공간 국소성(공간 지역성; spatial locality)

- 접근한 메모리 공간 근처를 접근하려는 경향
- CPU가 실행하려는 프로그램은 보통 관련 데이터들끼리 한데 모여있음
- 하나의 프로그램 내에서도 관련있는 데이터들은 모여서 저장됨

# Quiz

- **Q1. 각각 어떤 RAM을 설명한 것인지 맞춰보세요.**  
(DRAM, SRAM, SDRAM, DDR SDRAM)
  - 1) 대역폭을 2배 넓힌 SDRAM:
  - 2) 시간이 지나도 저장된 데이터가 사라지지 않는 RAM:
  - 3) 데이터의 소멸을 막기 위해 일정 주기로 데이터를 재활성화해야 하는 RAM
  - 4) 클럭과 동기화된 DRAM:
  - 5) 주로 캐시 메모리로 활용:
  - 6) 주로 주기억장치로 활용:
  - 7) 대용량화하기 유리:
  - 8) 집적도가 상대적으로 낮음:
- **Q2. 캐시 메모리를 사용하는 이유는 무엇인가요?**
- **Q3. 저장 장치 계층 구조를 완성해보세요.**  
(메모리, 보조기억장치, 캐시 메모리, 레지스터)



# Quiz

- **Q1. 각각 어떤 RAM을 설명한 것인지 맞춰보세요.**

(DRAM, SRAM, SDRAM, DDR SDRAM)

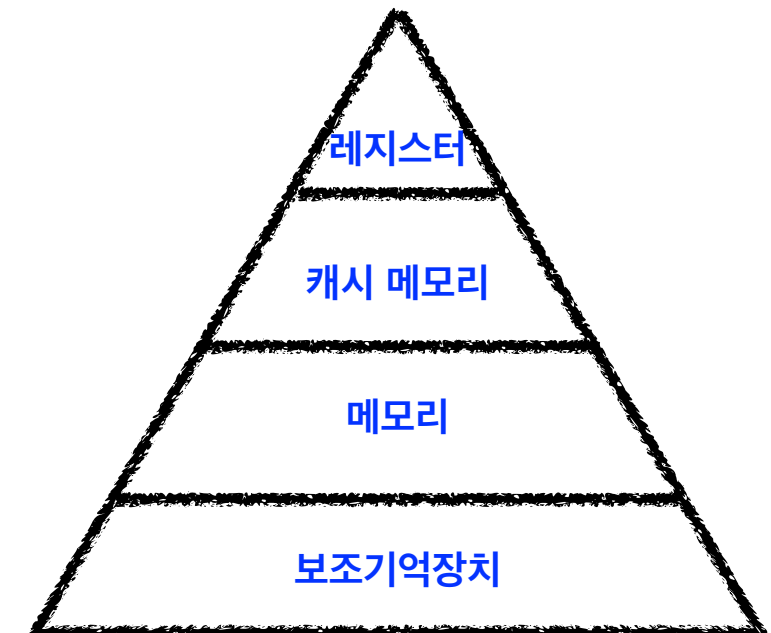
- 1) 대역폭을 2배 넓힌 SDRAM: **DDR SDRAM**
- 2) 시간이 지나도 저장된 데이터가 사라지지 않는 RAM: **SRAM**
- 3) 데이터의 소멸을 막기 위해 일정 주기로 데이터를 재활성화해야 하는 RAM: **DRAM**
- 4) 클럭과 동기화된 DRAM: **SDRAM**
- 5) 주로 캐시 메모리로 활용: **SRAM**
- 6) 주로 주기억장치로 활용: **DRAM**
- 7) 대용량화하기 유리: **DRAM**
- 8) 집적도가 상대적으로 낮음: **SRAM**

- **Q2. 캐시 메모리를 사용하는 이유는 무엇인가요?**

CPU가 메모리에 접근하는 시간이 CPU의 연산 속도보다 느리기 때문에 이러한 병목 지점의 처리 시간 차이를 메우기 위해 사용

- **Q3. 저장 장치 계층 구조를 완성해보세요.**

(메모리, 보조기억장치, 캐시 메모리, 레지스터)



# References

## Books

- 강민철(2022). 혼자 공부하는 컴퓨터 구조 + 운영체제. 한빛미디어
- 실습과 그림으로 배우는 리눅스 구조

## Youtube Videos

- [10분 테코톡] 🐻큰곰의 Cache
- RAM Vs. ROM | Animation