

06. RAM

yongzzai

Summary of Chap.06 of book [1]

1 RAM의 특성과 종류

* 주기억장치의 종류에는 RAM(Random Access Memory)과 ROM(Read Only Memory)이 있음. RAM은 읽기와 쓰기가 모두 가능한 반면, ROM은 읽기만 가능함.

* RAM은 실행할 대상을 저장하지만 전원이 꺼지면 저장된 내용이 사라지는 휘발성 메모리임. ** 보관해야하는 정보는 보조기억장치에 저장하며, 비휘발성 저장장치라고함.

* RAM의 크기가 작은 경우엔 CPU가 여러 프로그램을 동시에 실행할 수 있는 능력이 저하됨.

1.1 RAM의 종류

1. Dynamic RAM (DRAM)

: 저장된 데이터가 동적으로 사라지는 RAM으로 전원이 연결되어 있어도 저장된 데이터가 점점 사라지는 RAM임. 데이터의 소멸을 막기위해 주기적으로 **재활성화** (Refresh)해줌. 일반적으로 DRAM이 가장 많이 사용됨. 상대적으로 소비전력이 낮고 집적도 (디테일한 설계가 가능한)가 높아 비용이 저렴함.

2. Static RAM (SRAM)

: 저장된 데이터가 사라지지않는 RAM으로 전원이 연결되어 있어도 데이터가 사라지지 않음. DRAM보다 빠르지만 비용이 높고 소비전력이 높음. 대용량으로 설계할 필요는 없으나 빨라야하는 장치에 사용됨.

표 2-1 DRAM과 SRAM의 차이점

	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

3. Synchronous DRAM (SDRAM)

: 발전된 형태의 DRAM으로 클럭신호와 동기화된 형태의 DRAM임. 클럭신호 한번에 데이터를 한번 주고 받음.

4. Double Data Rate SDRAM (DDR SDRAM)

: 발전된 형태의 SDRAM으로 최근 가장 대중적으로 사용되는 RAM임. 클럭신호 한번에 데이터를 2개 주고 받음. 참고로 DDR2는 4개, DDR3는 8개, DDR4는 16개의 데이터를 한번에 주고 받음.

2 메모리의 주소공간

* CPU와 실행 중인 프로그램은 메모리의 몇번지에 무엇이 저장되어 있는지 알지 못함.

* 새롭게 실행되는 프로그램은 새롭게 메모리에 적재되며, 실행이 끝난 프로그램은 메모리에서 삭제됨. 또한 같은 프로그램을 실행하더라도 매번 다른 메모리 주소에 적재됨.

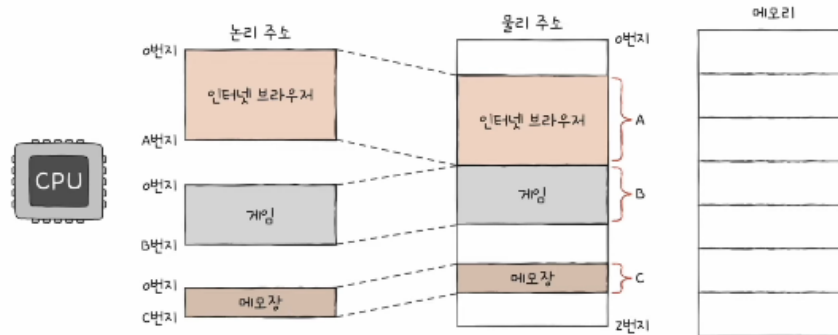
2.1 물리 주소와 논리 주소

1. 물리 주소

: 메모리의 입장에서 바라본 주소, 정보가 실제로 저장된 하드웨어상의 주소

2. 논리 주소

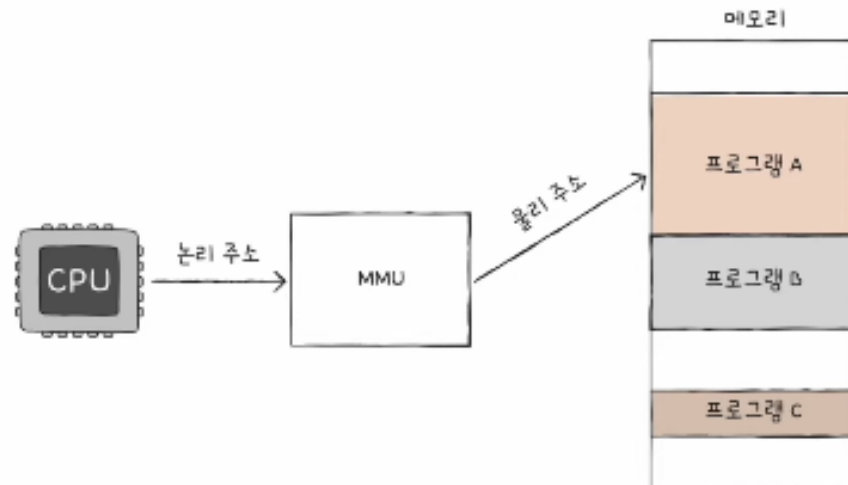
: CPU와 실행 중인 프로그램 입장에서 바라본 주소, 실행 중인 프로그램 각각에게 부여된 0번지부터 시작되는 주소



2.2 물리주소와 논리주소의 변환

* CPU는 논리주소를 사용하고, 메모리는 물리주소를 사용함. 따라서 CPU가 논리주소를 메모리에 전달할 때 논리주소를 물리주소로 변환해야함.

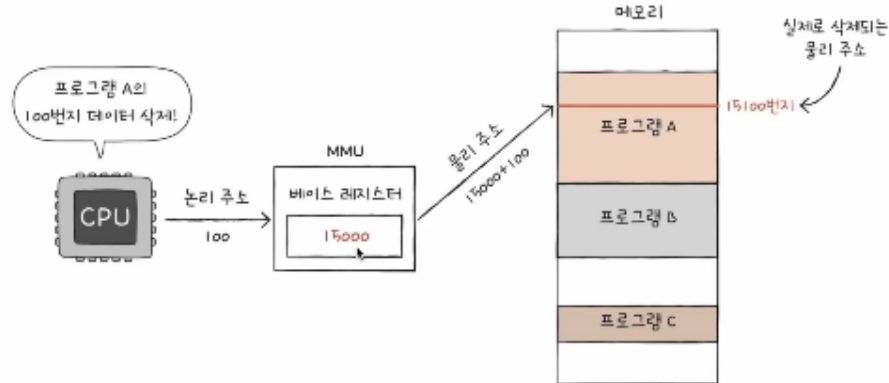
* MMU (메모리 관리 장치)는 논리주소를 물리주소로, 물리주소를 논리주소로 변환해주는 장치임.



* MMU는 논리주소와 베이스 레지스터 (기준 주소 역할을 하는 레지스터) 값을

더하여 논리 주소를 물리주소로 변환함.

* 베이스 레지스터는 프로그램의 시작주소를 나타내며 논리주소는 실제로 저장된 프로그램으로부터 얼마나 떨어져있느냐



* CPU가 논리주소 100번지에 있는 데이터를 삭제하라고 명령하면 MMU는 프로그램A의 기준주소 (프로그램 A가 메모리에서 시작하는 주소)를 15000번지에 100을 더한 15100번지에 있는 데이터를 삭제하라고 명령함.

* **한계 레지스터**는 램에서 사용할 수 있는 논리주소의 최대값을 나타내며 한계 레지스터를 넘어가면 (현재 명령어가 다른 프로그램을 침범하려고 할 때) 인터럽트를 발생시킴.

* 따라서 CPU는 메모리에 접근하기 전에 접근하고자 하는 논리 주소가 한계 레지스터보다 작은지를 항상 검사함.

2.3 캐시 메모리

* CPU가 메모리에 접근하는 시간은 CPU 연산 속도보다 훨씬 느림.

* **캐시 메모리**는 CPU와 메모리 사이에 위치한 레지스터보다 용량이 크고 메모리보다 빠른 SRAM기반의 저장장치이다.

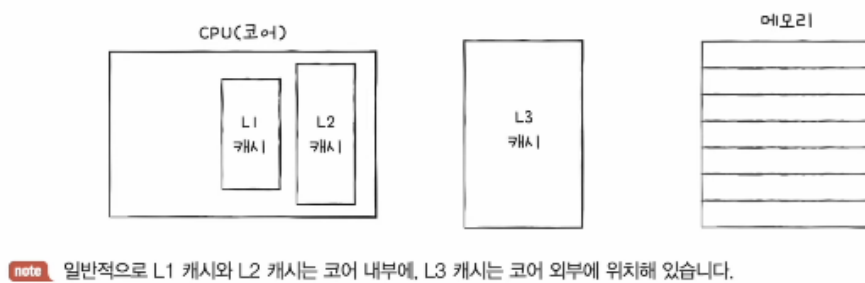
2.3.1 저장 장치 계층 구조 (Memory Hierarchy)

* CPU와 가까운 저장 장치는 빠르고, 멀리있는 저장 장치는 느림.

* 속도가 빠른 저장 장치는 저장 용량이 작고, 가격이 비쌈.

* 레지스터 > 메모리(RAM) > USB메모리 -> 레지스터가 가장 비싸지만 빠름.

- * CPU가 매번 메모리에 접근하는 것은 비효율적이므로 캐시 메모리를 사용함. 메모리에서 CPU가 사용할 일부 데이터를 미리 캐시 메모리로 가져다두고 쓰는 것이 핵심 아이디어임.
- * 메모리를 집에서 멀리 떨어진 대형 마트로 비유하면 캐시 메모리는 집 근처의 작은 편의점이라고 할 수 있음.
- * 캐시 메모리는 하나가 아닌 여러 계층적으로 구성이 됨.



- * 참고로 여러 코어가 여러개 있을 땐, 각 코어마다 L1, L2 캐시가 있고, 모든 코어가 공유하는 L3 캐시가 있음.

2.3.2 참조 지역성의 원리

- * 캐시 메모리는 용량이 작기 때문에 CPU가 자주 사용할 법한 내용을 예측하여 저장함.
- * **캐시 히트**는 예측이 들어맞는 경우를 말함.
- * **캐시 미스**는 예측이 틀린 경우를 말함.
- * **캐시 적중률**은 캐시 히트/(캐시 히트 횟수 + 캐시 미스 횟수)임.
- * **참조 지역성**은 CPU가 한번 참조한 데이터를 다시 참조할 확률이 높다는 원리임.
 1. CPU는 최근 접근했던 메모리 공간에 다시 접근하려는 경향이 있다 (시간 지역성).
 2. CPU는 접근한 메모리 공간 근처를 접근하려는 경향이 있다 (공간 지역성).

References

- [1] Minchul Kang. 혼자 공부하는 컴퓨터 구조+운영체제. Hanbit Media, 2022.