제 20강. chap5. 메모리 계층구조, 5.1 서론, 5.2 메모리 기술

5.1 서론

디스크 > DRAM > SRAM

빠르고 클수록 비쌈

지역성의 원칙 Locality principle

- 시간적 지역성 (temporal locality) : 한번 참조된 항목은 곧바로 다시 참조되는 경향이 있다.

  프로세서에 한번 가져오면 캐시에 보관해둬라

  캐시 - 프로세서 옆이나 메모리 옆에 있는

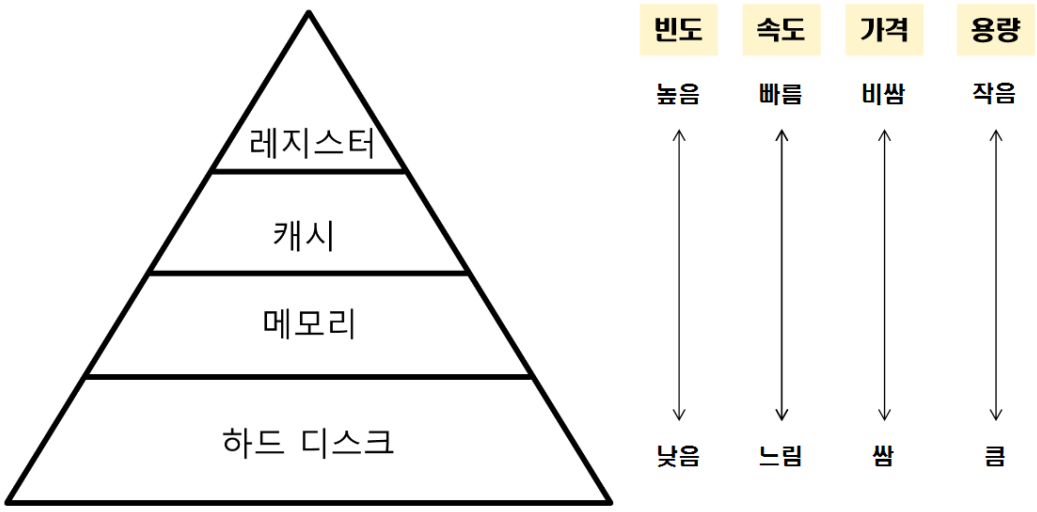
- 공간적 지역성 (spatial locality) : 어떤 항목이 참조되면 그 근처에 있는 다른 항목들이 곧바로 참조될 가능성이 높음.

  가져올 때 하나만 가져오는 게 아니라 한 블럭을 가져오는 게 좋음

  sequential block

지역성 원칙 활용

메모리 계층구조



- 디스크에는 모든 데이터가 다 저장
- 데이터는 인접한 두 계층 사이에서 한 번에 복사됨
- 상위 계층은 더 비싼 기술을 사용하므로 하위 계층보다 작고 빠름

- 최근에 접근했던 데이터를 프로세서 가까이에 적재함 → 시간적 지역성 활용
- 필요한 데이터뿐 아니라 인접한 다량의 데이터로 이루어진 블록을 메모리의 상위 계층으로 옮김 → 공간적 지역성 활용

메모리 계층구조

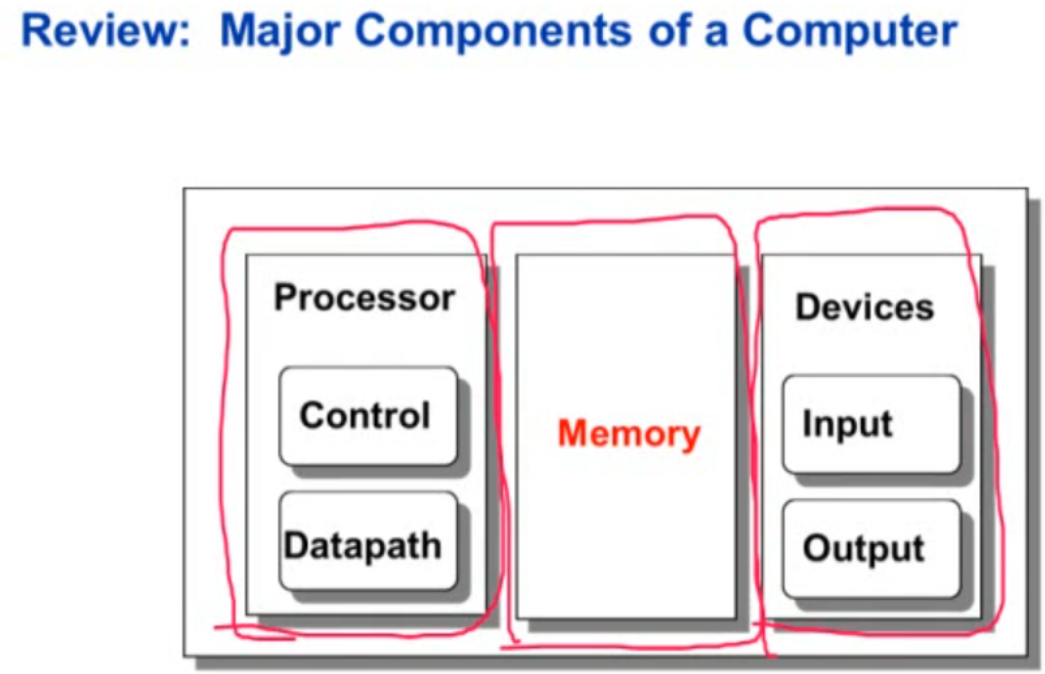
1. 블록 or 라인 : 두 계층 간 정보 전송의 최소 단위
   캐시에 있을 수도 없을 수도 있는 저옵의 최소 단위
2. 적중(hit) : 프로세서가 요구한 데이터가 상위 계층의 어떤 블록에 있을 때
3. 실패(miss) : 상위 계층에 없을 때 아래 계층에서 데이터를 가져와야 함
4. hit rate/hit ratio (적중률) : 메모리 접근 중 상위 계층에서 찾을 수 있는 것의 비율
   메모리 계층 성능을 평가하는 척도
   hits / accesses
5. miss rate(실패율) : 특정계층에서 찾을 수 없는 메모리 접근의 비율, 1 - hit ratio
6. hit time (적중 시간) : 메모리 계층구조의 상위 계층에 접근하는 데 걸리는 시간

7. **실패 손실** (miss penalty) : 하위 계층에서 해당 블록을 가져와서 상위 계층 블록과 교체하는 시간에다 그 블록을 프로세서에 보내는 데 걸리는 시간을 더한 값

>> 모든 프로그램은 메모리 접근에 많은 시간을 쓰기 때문에 메모리 시스템은 성능을 결정하는 중요한 요소

>> 메모리 계층구조는 행렬 곱셈의 성능을 2배로 향상시킬 수 있음

5.2 메모리 기술



캐시 메인 메모리 Secondary Memory(Disk)

5장 6장

메모리 기술

ROM vs RAM

- ROM : Read Only Memory, 컴퓨터에 지시사항을 영구히 저장하는 비휘발성 메모리
- RAM : Random Access Memory, 한시적으로 저장하는 휘발성 메모리

메모리 종류

- DRAM (Dynamic RAM) : 메인 메모리 구현, SRAM보다 느리고 비트당 가격이 덜 비쌈

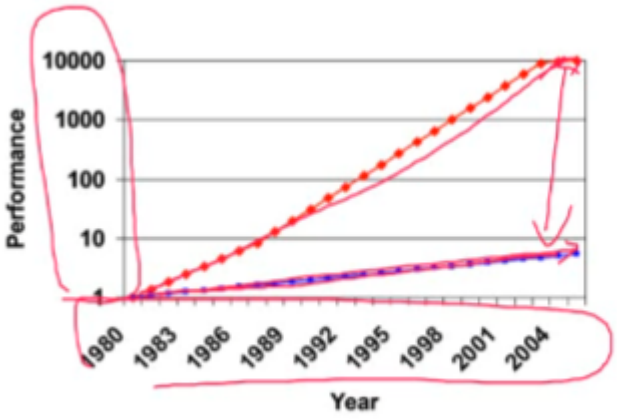
  메모리 비트당 면적이 작기 때문에 가격 차이가 생기고 같은 양의 실리콘으로 더 큰 용량을 만들 수 있음

- SRAM (Static RAM) : 캐시 구현
- 플래시 메모리 : 비휘발성 메모리, 개인 휴대용 기기에서 2차 메모리로 사용됨
- 자기 디스크 (Magnetic Disk) : 서버에서 가장 크고 가장 느린 계층을 구현하는 데 사용

Memory technology	Typical access time	$ per GiB in 2020
SRAM	0.5 -2.5 ns	$500 - $1000
DRAM	50 - 70 ns	$3 - $6
플래시 메모리	5,000 - 50,000 ns	$0.06 - $0.12
자기 메모리	5,000,000 - 20,000,000ns	$0.01 - $0.02

Memory Wall

Processor vs DRAM speed disparity continues to grow



빨간색 - 프로세서 CPU

파란색 - 메모리

Moore's Law

메모리는 늦게 증가하기 때문에 전체 성능 향상은 CPU가 증가한 만큼 느껴지지 않는다.

cf) Power Wall

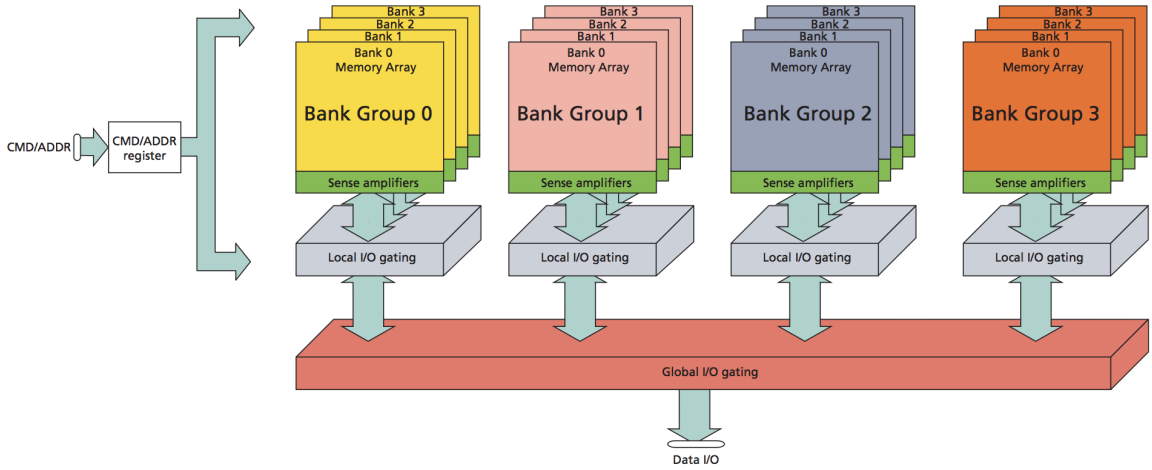
SRAM 기술

- SRAM이란 ? 읽기나 쓰기를 제공할 수 있는 접근 포트가 (일반적으로) 하나 있는 메모리 배열로 구성된 단순한 집적 회로
  읽기 접근시간과 쓰기 접근시간은 다를 수 있지만, 어떤 데이터든지 접근시간은 같다.
- SRAM은 리프레쉬가 필요 없으므로 접근시간 = 사이클 시간과 거의 같음
- 사이클 시간 = 메모리 접근 사이의 시간 간격
- SRAM은 읽을 때 정보가 바뀌지 않게 하기 위해 비트당 6개~8개의 트랜지스터를 사용
- 대기 모드에서 데이터 값을 유지하기 위해 최소한의 전력만을 사용
- 과거에는 1,2,3차 캐시에 별도의 SRAM을 사용하였으나
  현재는 모든 계층의 캐시가 프로세서 칩에 집적되어서 별도의 SRAM을 사용할 필요 X

DRAM 기술

- VS SRAM : SRAM은 전력이 공급되는 한 그 값이 무한히 유지
  DRAM은 셀에 기억되는 값이 전하 형태로 커패시터에 저장
- 저장된 값을 읽거나 새로운 값을 쓰기 위하여 저장된 전하를 접근하는 데 트랜지스터가 하나 필요
- 비트 하나당 트랜지스터 하나만 있으면 되므로 SRAM에 비해 훨씬 집적도가 높고 값도 쌈
- 2단계 디코딩 구조를 가져 전체 행을 한꺼번에 읽는 읽기 사이클 후 바로 쓰기 사이클을 실행하여 한 행을 통째로 리프레시할 수 있음

DRAM 내부 구조



최신 DRAM은 뱅크로 구성

DDR4는 일반적으로 뱅크가 4개

뱅크 = 일련의 행으로 구성, Pre 신호는 뱅크를 열거나 닫는 데 사용됨

행 주소는 Act 신호와 함께 보내지는데 이 신호는 행을 버퍼로 보냄

버퍼에 있는 행은 DRAM의 폭이 얼마든지 간에 연속된 열 주소를 사용해 전송하거나

블록 전송과 시작 주소를 지정해 전송 가능. 각 신호도 클럭과 동기화

SDRAM (synchronous DRAM)

DRAM에 클럭을 추가한 것

→ 클럭을 사용하므로 메모리와 프로세서를 동기화하는 시간이 필요 없음

- DDR(double data rate) SDRAM : 주소를 일일이 지정하는 대신 클럭이 연속적인 비트들을 버스트 모드로 전송
- 클럭의 상승 에지에서도 데이터가 전송되고 하강 에지에서도 데이터가 전송되어 대역폭이 2배가 됨
- 주소 인터리빙 (address interleaving) : 한 주소를 여러 뱅크에 보내서 모든 뱅크가 동시에 읽고 쓰게 하는 것 → n배의 대역폭을 제공 가능

miss penalty, bandwidth 계산

ex) cache block read example

- 1 bus cycle for address transfer
- 15 bus cycles per DRAM access
- 1 bus cycle per data transfer

for 4-word block 1-word-wide DRAM

- Miss penalty = 1 + 4*15 + 4*1 = 65 bus cycles
- bandwidth = 16bytes/65bytes = 0.25B/cycle

플래시 메모리

전기적으로 지울 수 있고 프로그래밍이 가능한 ROM 의 한 종류

디스크 메모리

디스크 메모리의 구성

- 원판의 집합으로 구성

  ㄴ분당 5400번에서 15,000번의 속도로 회전
- 양측 면은 카세트나 비디오테이프와 같이 자성체로 코팅
- 읽기/쓰기 헤드라고 불리는 작은 전자기 코일을 가지고 있는 암(arm)이 각 표면 위에 있음
- track : 디스크 표면을 나누는 동심원
- sector : 각 트랙은 정보를 저장하는 섹터로 나누어짐, 512bytes~4096bytes
- 실린더 : 헤드 아래에 있는 모든 면의 트랙

디스크 메모리에서 데이터에 접근하기

1. 탐색 : 적절한 트랙 위에 디스크 암을 갖다 놓음

   탐색 시간 : 디스크 헤드를 원하는 트랙까지 이동하는 데 걸리는 시간
2. 읽기/쓰기 헤드 밑에 원하는 섹터가 올때까지 기다림

   ㄴ 회전 지연 시간 (rotational latency)

   평균 회전 지연 시간 = 디스크가 1/2회전하는 데 걸리는 시간

   5400RPM일 때 평균 회전 지연 시간

   = 0.5 / 5400RPM = 0.0056sec = 5.6ms
3. 전송시간 : 블록 하나를 전송하는 시간