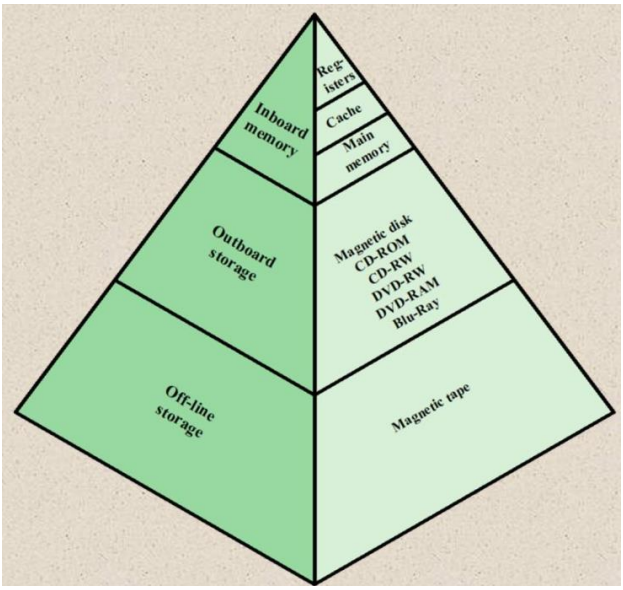
1. Explain the locality of memory reference and explain the hierarchical memory system.

Cpu가 메모리를 참조할 때는 spatial locality, temporal locality가 있다. Spatial locality는 실행 시 순차적으로 실행되는 프로그램의 특성을 이용해서 cpu가 100번지에 있는 프로그램을 접근할 때 그 다음에 실행될 가능성이 높은 101,102번지를 접근하는 순차적 접근 특성을 말한다. Temporal locality는 같은 데이터나 프로그램을 반복해서 참조하는 특성으로, 대표적으로 90/10 법칙이 있다. 전체 프로그램의 10%가 전체 수행시간의 90%를 차지한다는 법칙인데, 결과적으로 프로그램의 10%가 반복적으로 수행된다는 것을 뜻한다.




메모리는 속력이 빠를수록 가격은 비싸지고, 용량이 커질수록 느려지고 가격이 싸진다. 하지만 우리는 cpu가 요구하는만큼 빨라야하고, 용량은 필요한 프로그램을 저장할만큼 커야하며 구매할 수 있을만큼 가격이 저렴한 메모리를 원한다. 그래서 이러한 문제를 해결하기 위해 계층적 메모리를 사용한다. 계층적 메모리는 레지스터, 캐시, 메인 메모리, 하드 디스크를 층으로 메모리를 사용하며 계층적 메모리를 이용해서 속력이 빨라야하고 용량은 커야하고 가격은 싼 메모리 요구 조건을 충족시킬 수 있다.

2. Main memory:16GB(Ta=45ns), Block:16Bytes, Cache: 16MB(Ta=5ns)

1) What is the number of address bits connected to the main memory? (Assume byte addressable)

메모리 용량은 16GB=$2^{34}$Bytes 16Gb 메모리에 연결되는 address bits는 34개가 필요하다

2) Show the size of the Tag in case of Direct Mapping and Associative Mapping.

Block size = line size = 16bytes =$2^{w}$=$2^{4}$

Number of lines in cache =캐시 용량/블록 크기= 16Mb/16Bytes=1Mb= $2^{r} = 2^{20}$

Number of blocks in main memory = 메인 메모리 용량/블록 크기 = 16GB/16Bytes = 1Gb $=2^s=2^{30}$

Direct mapping의 태그 크기 = (s-r)bits = (30-20)bits=10bits

Associative mapping의 태그 크기= s bits=30bits

3) Show the advantage(s) of associative mapping over direct mapping.

mapping이란 물리적 주소가 캐시 메모리에 있는지 없는지, 만약 있다면 어디에 있는지를 알아내는 방법이다.

Direct mapping은 가장 간단한 방법으로, 블록들이 캐시에 들어갈 자리가 미리 정해져있고 캐시의 각 블록에는 하나의 블록만 들어올 수 있다. 하지만 자주 사용하는 다수의 블록이 하나의 같은 위치로 mapping된다면 계속 miss가 발생하고 성능이 저하된다.

Associative mapping은 메인 메모리의 어떤 부분이라도 캐시에 빈 곳만 있으면 저장할 수 있다. 메모리 블록 적재 시에 캐시 라인이 따로 정해지지 않기 때문에 캐시 라인 선택이 자유롭고 적중률이 direct mapping에 비해 매우 높다

4) Explain a mapping method which does not require replacement algorithm.

Replacement algorithm은 캐시가 가득 차 있을 때 새로운 블록을 캐시에 넣어야하는 상황에서 기존의 블록을 빼고 새로운 블록을 캐시에 넣는 알고리즘을 말한다. Direct mapping은 그 주소에 해당하는 내용이 1개밖에 없고, 해당 주소의 기존에 있는 내용을 그냥 빼기만 하면 되기 때문에 replacement algorithm이 따로 필요하지 않다

5) If the given main memory is implemented with 4G×1bit memory chips, how many memory chips are needed?

4G비트를 바이트로 환산하면 $2^{29}$Bytes이다. 따라서 용량이 16GB인 메모리 안에서 이러한 칩의 개수를 계산해보면 칩의 개수 = 메모리 용량/칩의 용량 = $2^{34}/2^{29}=2^5$이므로 메인 메모리에는 32개의 칩이 들어갈 수 있다.

6) Determine the cache hit ratio for the effective memory access time of 10ns.

$$Ts = Pc \times Tc + (1-Pc) \times (Tc + Tm) = Tc + (1-Pc) \times Tm$$

- Pc : Cache hit ratio
- Tc : access time of cache
- Tm : access time of main memory

Access time을 구하는 식은 다음과 같다. 주어진 값들을 식에 대입해보면,

10=5+(1-h)*45 =-45h+50 이므로 h=8/9이다.

3. Explain the case when write through policy is better than write back policy?

Write through는 캐시만 내용을 바꾸고 메인 메모리의 내용은 바꾸지 않으면 내용의 불일치가 일어나기 때문에 캐시와 메인 메모리의 내용을 동시에 둘 다 바꾸는 방법이다

Write back은 캐시에만 내용을 바꾸고 메인 메모리의 내용은 바꾸지 않고 필요할 때에만 바꾸는 방법이다.

Write through가 write back보다 더 좋은 경우는 다음과 같다. Write through는 위에서 언급한 것처럼 캐시와 메모리 내용을 둘 다 동시에 바꾸기 때문에 데이터 일관성을 유지할 수 있어서 안정적이다. 그렇기 때문에 데이터 손실이 발생하면 안되는 상황에서 사용하는 것이 좋다.

4. Explain advantages of a split cache compared to a unified cache.

Unified cache는 프로그램과 데이터를 구별없이 메모리에 저장하고 그 메모리에 대한 접근은 프로그램이나 데이터의 주소를 사용한다. 그런데 cpu에서 프로그램과 데이터가 동시에 필요하면, 메모리에서 프로그램을 가져올땐 데이터 가져오는 것을 기다려야하고 데이터를 가져올땐 프로그램 가져오는 것을 기다려야 한다. 따라서 성능 상에 문제가 생길 수 있다.

Split cache는 프로그램과 데이터를 저장하는 캐시를 따로 두어서 프로그램을 가져올 때 데이터를 동시에 가져올 수 있어서, unified cache보다 프로그램 성능을 향상시킬 수 있다.

5. Select all volatile memories. ① SDRAM ② Flash memory ③ SSD ④ CD-R ⑤ Hard disk ⑥ SRAM ⑦ EEPROM ⑧ DDR SDRAM

Volatile memory는 휘발성 메모리로, 저장된 정보를 유지하기 위해 전기를 요구하는 컴퓨터 메모리이다. 반대로 Nonvolatile memory는 비휘발성 메모리로, 전원이 공급되지 않아도 저장된 정보를 계속 유지하는 컴퓨터 메모리이다. Volatile memory로는 시간이 지나면 스스로 방전되기 때문에 일정 시간마다 기억장치의 내용을 저장시키는 동적인 DRAM의 발전 형태인 SDRAM, 전원 공급만 되면 내용이 소멸, 변하지 않는 정적인 SRAM, SDRAM의 차세대인 DDR SDRAM이 있다.

6. What is the major advantage of interrupt-driven IO over programmed IO?

I/O는 데이터를 직접 읽어올 수 없기 때문에 I/O의 상태를 체크해서 ready 상태가 됐을 때 I/O 장치로 데이터를 전송하도록 한다. 상태 준비와 전송 2가지를 CPU가 다 하는 경우를 programmed I/O라고 한다

interrupt-driven I/O는 상태 체크를 I/O 모듈 자체가 하고 준비가 됐을 때 데이터 전송만 CPU가 하는 방법이다. Interrupt-driven 방식은 programmed 방식에 비해 cpu가 I/O에 참여하는 정도가 작아서 CPU가 I/O에 참여하는 기여도가 작아진만큼의 시간동안, 메인 프로그램을 수행하는데 사용할 수 있어서 더 효율적이다.

7. Explain how an external device indicates an Interrupt event to the processor and how the processor identifies the external device

대표적인 device identification 방법으로는 Daisy chain이 있다. CPU에 interrupt request 선과 interrupt acknowledge 신호가 있다. Interrupt request 선에 어떤 I/O 디바이스가 요청하면, CPU는 현재 수행하고 있는 일이 끝나면 interrupt acknowledge 신호를 보낸다. 여러 장치들이 있으면 각 디바이스마다 거쳐서 daisy chain식으로 연결이 돼있다. Interrupt acknowlenge 신호를 받은 장치들이 있다면 그 장치 중에 제일 앞부분에 있는 장치가 interrupt acknowledge 신호를 받게 한다. 만약 앞부분에 있는 장치가 interrupt request를 보냈었다면 끝나지만, 보내지 않았다면 뒤의 디바이스에게 넘긴다. 신호를 받은 디바이스가 request했던 디바이스라면 cpu에게 자신이 누구인지를 알리는 벡터 값을 데이터 버스를 통해 보내서 device identification을 한다. 만약 2개의 디바이스에서 동시에 interrupt request를 보냈는데 앞의 디바이스에서 신호 처리가 완료되어 뒤의 디바이스로 전달되지 못할 경우가 존재한다.