1. 메모리 참조 지역성의 특징은 먼저 참조가 일어난 메모리의 근처에서 참조가 일어나는 경향이 있다는 것과 한 번 참조된 메모리는 다시 참조될 확률이 높다는 것, 마지막으로 메모리 참조는

  순차적으 로접근한다는 것이다.

  또한 메모리가 계층구조로 되어 있는 이유는 메모리의 용량, 속도, 비용에는 trade-off라는 하나를 얻으면 하나는 포기해야 하는 특성이 있다. 하지만 우리는 용량은 크면서 빠르고 싼 메모리를 원한다.

  따라서 계층구조로 메모리를 만들어 이 모든것을 충족 시킬 수 있게 하는 것이다.

2 -1.  1GB는 2^30byte 이다. 즉, 16GB는 2^34 byte이고 문제에서 1byte 당 하나의 주소를 저장한다고 했으므로  2^34개의 주소를 갖는다.

2-2.  메인메모리=2^34 byte, block크기 = 2^4byte, 캐시크기= 2^24byte 이므로  w = 4, s = 30 ,r = 20 이다.

       따라서 Direct Mapping에서 Tag의 사이즈는  (s-r)bit로 10bit 이고 Associative Mapping 에서 Tag의 사이즈는 30bit 이다.

2-3. Direct Mapping은 특정한 장소에 특정한 데이터를 저장해야 한다. 하지만 만약 자주 사용하는 데이터들이 Direct Mapping 기법을 사용하여 같은 위치에 계속 저장된다면 히트율이 떨어질 것이다.

      반면에 Associative Mapping은 아무데나 데이터를 저장할 수 있기 때문에 히트에 크게 영향을 받지 않는다는 장점이 있다.

2-4. replacement 알고리즘이 필요없는 것은 Direct Mapping이다. 왜냐하면 Direct Mapping은 캐시의 지정된 장소에 메인메모리의 지정된 데이터가 들어가게 된다. 따라서 교체 시에도 어떤 데이터를 뺄지

     알고리즘을 계산할 필요가 없다.

2-5. 4Gx1bit chip은  2^32개의 주소를 갖고, 각 주소에는 1bit의 용량이 저장된다는 것이다. 따라서 16GB(2^37bit) 용량의 메인메모리를 구성하려면 chip은 적어도 2개가 필요하다.

2-6. effective memory access time = 캐시 접근 시간 +(1-캐시 hit ratio) x 메인메모리 접근 시간 이다. 따라서 10 = 5 +(1-캐시 hit ratio) x 45 라는 식을 구하면 캐시 hit ratio 는 약 0.88888889가 된다.

3. write through 의 겨우 캐시와 메인메모리가 같이 수정되면서 데이터가 일관성 있게 유지된다. write back은 일단 캐시만 수정하고 나중에 메모리로 수정 사항이 옵데이트 되는데  이경우 속도가 느린 I/O 나 사용중인 컴퓨터의 파워가 나가는 경우에는 수정사항이 메인메모리로 업데이트 되지 못할 수 있기 때문에 이런 경우엔 write through이 더 나은 방법일 수 있다.

4. unified 캐시의 경우 데이터와 프로그램을 모두 구별없이 저장한다. 하지만 프로그램과 데이터가 동시에 필요한 경우 하나씩 처리되기를 기다려야 하는데 split 캐시의 경우 프로그램과 데이터를 따로 공간을 지정해 저장하므로 처리 속도가 빠르다는 장점이 있다.

5.   1, 6, 8 이 휘발성이다.

6. programmed I/O는 I/O 장치가 준비되는 것을 확인하고 데이터를 전송하는 모든 일을 CPU가 한다, 그런데 I/O 장치는 상당히 느리기 때문에 상당히 비효율적일 수 있다. 하지만 interrupt-driven I/O를 사용하면 I/O 장치가 준비될 동안 CPU은 다른일을 할 수 있고 준비가 되면 그떄 interrupt를 걸어 데이터 전송에만 CPU가 관여하므로 더 효율적으로 CPU를 사용할 수 있다는 장점이 있다.

7. I/O 장치가 프로세스에게 interrupt를 알리고 프로세스가 어떤 장치가 어떤 interrupt를 걸었는지 구분하는 주요한 방식은 Daisy chain 방식이다.

   i/O장치는 interrupt를 걸 때 vector에 자신이 어떤 장치인지 그리고 자신이 거는 interrupt의 인덱스 주소를 함께 넣어 보낸다. 그러면 프로세스는 이 신호를 받아 interrupt     acknowledge를 보내는데 I/O 장치들이 체인처럼 연결되어 있어서 앞에 연결된 장치부터 순차적으로 자신이 보낸 interrupt에 대한 응답이 맞는지 확인한다.