1. Exercise 6.25 on page 686 with the following table.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Cache | m | C | B | E | S | t | s | b |
| 1 | 48 | 64KB | 32 | 4 | 512 | 34 | 9 | 5 |
| 2 | 48 | 64KB | 32 | 8 | 256 | 35 | 8 | 5 |
| 3 | 48 | 64KB | 32 | 16 | 128 | 36 | 7 | 5 |
| 4 | 48 | 64KB | 64 | 1 | 1024 | 32 | 10 | 6 |
| 5 | 48 | 64KB | 64 | 2 | 512 | 33 | 9 | 6 |
| 6 | 48 | 64KB | 64 | 8 | 128 | 35 | 7 | 6 |

2. Exercise 6.29 on page 687 with the following assumption

Assumption:

- There are only two sets (S = 2).

- Set index 0 is 0 and set index 2 is 1. There are no entries for set 1 and 3.

set index=1bit block은 4byte이므로 block offset=2bit이다.

주소가 12bit이므로 tag bit는 9bit이다.

a) CT CT CT CT CT CT CT CT CT CI CO CO

b)

0x834=100000110100이다.

Set는 0=1인데 태그 값이 10000011이다.

Set1의 tag는 111111111이거나 100000001 이므로 miss. 10000011 이라는 tag의 값이 set1의 valid bit가 0인 부분에 load된다.

0x836=1000001100110이다. Set가 1이고 태그값은 10000011이다. 이는 아까 0x834에서 cache에 load한 값으로 hit가 발생함.

0xffd=111111111101이다. Set가 1이고 태그 값은 111111111인데 이는 valid bit가 1인 값으로 cache에 이미 load되어 있다. Block offset이 1이므로 1byte를 읽어오면 c0이다.

miss

hit

hit->C0

3. Exercise 6.35 on page 689 with the following assumption

Assumption:

- The cache 2-way set-associative, write-through, write-allocate, with a block size of 16 bytes,

- The total size of cache is 256.

Cache의 size는 256byte이고 전체 배열은 4\*4\*4=64byte이다. Dst array의 시작주소는 64이므로 src array와 set가 겹치지 않는다. 따라서 dst array는 첫번째 열에 접근할 때만 miss가 나고 나머지 열은 cache에 이미 load된 값이므로 miss가 나지 않는다. 이와 동일하게 src array도 행의 첫번쨰 element에 접근할 때만 miss가 나고 나머지 경우에는 다 hit된다.

Dst array>

M h h h

M h h h

M h h h

M h h h

Src array>

M h h h

M h h h

M h h h

M h h h

4. Exercise 6.42 on page 695

루프 돌 때마다 처음 buffer의 pixel의 member에 접근할 때 miss가 나고 char pointer로 순차적으로 접근하고 block이 8byte이므로, 그 후 7개에 대해선 hit가 발생한다. 따라서 miss rate는 1/8=12.5%이다.

따라서 hit rate는 87.5%이다.

5. Practice Problem 7.5 on page 731 when the linker relocates .text in m.o to address 0x4004d0 and swap to address 0x4003e8.

0x4004d0+0xa=0x4004da

0x4003e8-4-0x4004da=0xffffff0a

=>0xffffff0a

6. Exercise 7.8 on page 751

A.

a) REF (main.1)->DEF(main.1)

b)REF(main.2)->DEF(main.2)

=>static is local symbol

/weak와 strong은 global symbol의 문제를 다룰 때 이용

B.

a) REF (x.1)->DEF(unknown)

b)REF(x.2)->DEF(unknown)

=>두개의 weak symbol

C.

a) REF (x.1)->DEF(error)

b)REF(x.2)->DEF(error)

//두개의 강한 symbol

7. Exercise 7.10 on page 752

Symbol resolution을 할 때 앞서 참조 된 값이 나중에 꼭 정의 되어야 한다. 따라서 a->b인 경우 b가 더 뒤에 위치해야 한다.

A. gcc p.o libx.a

B gcc p.o libx.a liby.a libx.a

C gcc p.o libx.a liby.a libx.a libz.a