CHAPTER 5.3

1

a) 32-bit memory adress references/word addresses:

1, 134, 212, 1, 135, 213, 162, 161, 2, 44, 41, 221

The cache has 16 blocks so that 4 bits are required for index and the tax requires 32-4= 28 bits. Therefore the following table shows the binary address, tag, index, and hit/miss status of the above word address:

Word address	Binary address	Tag	Index	Hit/Miss
1	1	0	0001	Miss
134	10000110	1000	0110	Miss
212	11010100	1101	0100	Miss
1	1	0	0001	Hit
135	10000111	1000	0111	Miss
213	11010101	1101	0101	Miss
162	10100010	1010	0010	Miss
161	10100001	1010	0001	Miss
2	10	0	0010	Miss
44	101100	10	1100	Miss
41	101001	10	1001	Miss
221	11011101	1101	1101	Miss

STEP 2

b) 32 - bit memory address references / word addresses:
6, 214, 175, 214, 6, 84, 65, 174, 64, 105, 85, 215

The cache has 16 blocks so that 4bits are required for index and the tag requires 32-4= 28 bits. Therefore the following table shows the binary address, tag, index, and hit/miss status of the above word address.

Word address	Binary address	Tag	Index	Miss/Hit
6	110	11	0000	Miss
124	1111100	1111	0100	Miss
175	10101111	1010	1111	Miss
214	11010110	1101	0110	Miss
6	110	11	0000	Hit
84	1010100	1010	0100	Miss
65	10000001	1000	0001	Miss
174	10101110	1010	1110	Miss
64	1000000	1000	0000	Miss
105	1101001	1101	0001	Miss
85	1000	10	0000	Miss
215	110101111	1101	0111	Miss

5.3 chapter

2

step 1

a) Each block contain two words, a 1-bit offset is required. The cache has 8 blocks so that 3 bits are required for index and tag is requires 32-(3+1)=28 bi

Therefore the following table shows the binary address, tag, index, and hit/miss.

word address	Binary address	Tag	Index	Hit/miss
1	1	0	000	Miss
134	10000110	1000	011	Miss
212	11010100	1101	010	Miss
1	1	0	000	Hit
135	1000011	1000	011	Hit
213	11010101	1101	010	Hit
162	10100010	1010	001	Miss
161	10100001	1010	000	Miss
2	10	0	001	Miss
44	101100	10	110	Miss
41	101001	10	100	Miss
221	11011101	1101	110	Miss

step2

b) Each block contaiy two words an 1-bit offset is required. The cache has 8 blocks so that 3 bits are required for index and tag is requires 32-(3+1)=28 bits
Therefore the following table shows the binary address, tag, index and miss/hit.

Word address	Binary address	Tag	Index	Hit/Miss
6	110	11	000	Miss
124	1111100	1111	010	Miss
175	10101111	1010	111	Miss
214	11010110	1101	011	Miss
6	110	11	000	Hit
84	1010100	1010	010	Miss
65	10000001	1000	000	Miss
174	10101110	1010	111	Hit
64	1000000	1000	000	Miss
105	1101001	1101	000	Miss
85	1000	10	000	Hit
215	11010111	1101	011	Hit

4

Step 1:

a) The number of sets in the cache $= 64K/(1\times4)$

$= 16K$

$n = 14$ bits

Cache block size $m=0$ (because 1 word for a block)

Total cache size =

$$2^n\times(2^m\times32+(32-n-m-2)+1)$$

$$=2^{14}\times(2^0\times32+(32-14-0-2)+1)$$

$= 802816$ bits

step 2

16-word block:

if $n=10$ and $m=4$

Total cache size $= 2^n\times(2^m\times32+(32-n-m-2)+1)$

$$=2^{10}\times(2^4\times32+(32-10-4-2)+1)$$

$= 541696$ bits

step 3

if $n=11$ and $m=4$

Total cache $= 2^n\times(2^m\times32+(32-n-m-2)+1)$

$$=2^{11}\times(2^4 * 32+(32-11-4-2)+1)$$

$= 1$ Mbit

That is 128 Kb of data

Therefore the second cache might provide the slower performance because the miss penalty is larger, increase block size may also increase the conflict misses step 4 of 4E of Chepter 5 3

b) The number of sets in the cache n=13 bits and m=1 because block size is 2 word.

Total cache size= $2^{n} \times (2^{m} \times 32 + (32-n-m-2)+1)$

$= 2^{13} \times (2^{1} \times 32 + (32-13-1-2)+1)$

$= 663552$ bits

step 5

16-word block:

if n=10 and m=4

Total cache size= $2^{n} \times (2^{m} \times 32 + (32-n-m-2)+1)$

$= 2^{10} \times (2^{4} \times 32 + (32-10-4-2)+1)$

$= 541696$ bits

step 6

if n=11 and m=4

Total cache size = $2^{n} \times (2^{m} \times 32 + (32-n-m-2)+1)$

$= 2^{11} \times (2^{4} \times 32 + (32-11-4-2)+1)$

$=$ 1Mbit.

That is 64 Kb of data

The second cache might provide the slower performance because the miss penalty is larger, increase block size may also increase the conflict misses

6 of 5.3

step 1:

Yes It is possible to use this index a direct-mapped cacte because each blelk address still has a stable mapping between the address and the actual locatioy in cache. . However you neeed more tag bits to store in the cache array for determining the hit or miss.