Jong Park
10/17/2018
CS 472 – Computer Architecture
HW2

1. Suppose the existence of a direct-mapped cache design with a 32-bit address. Assume that each address maps to a single byte. Assume the cache scheme uses the following breakdown of the address bits:

Tag	**Index**	**Offset**
31-9 (23bits)	8-5 (4bits)	4-0 (5bits)

a) What is the cache block size (in words)?

$5\,offsets = \frac{2^5\,bytes}{block} * \frac{words}{4\,bytes} = \frac{32}{4}\,words$ = **8words/ block**

b) How many entries does the cache have?

$4\,Index = 2^4\,rows = 16\,rows\,of\,index$
$16\,rows * 32\,bytes$ = **512 bytes of entries**

c) Including space for the valid bits, tags, and actual block data, how many bits would be required to actually implement this hypothetical cache?

Total Cache size = (# of Rows) *(valid bit + tag bits + data bits)

$Cache = 16 * (1 + 23 + 32\,bytes * (\frac{8\,bits}{1\,byte}))$ = **4,480 bits**

2. Using the cache design from problem 1, assume the following byte-addressed cache references are recorded. Assume that the cache is initially empty.
Also assume that accesses occurred from left to right (e.g. address 0 was requested, then address 4, followed by address 16, etc).

Byte Address											
0	4	16	132	232	160	1024	30	140	3100	180	2180

a) What was the number of cache hits?

Address (binary)	**Address**	Block address	**Index (mod16)**	**Hit/Miss**
0000 00000	0	0	0	Miss
0000 00100	**4**	**0**	**0**	**Hit**
0000 10000	**16**	**0**	**0**	**Hit**
0100 00100	132	4	4	Miss
0111 01000	232	7	7	Miss
0101 00000	160	5	5	Miss
10 0000 00000	1024	32	0	Miss*
0000 11110	30	0	0	Miss*
0100 01100	**140**	**4**	**4**	**Hit**
110 0000 11100	3100	96	0	Miss*
0101 10100	**180**	**5**	**5**	**Hit**
100 0100 00100	2180	68	4	Miss*

* means replaced

4 Hits.

(Bold indicates final state of the index)

b) Create a table to show the final state of the cache including the value of each Index, Valid bit, and Tag.

Index	**Valid**	**Tag** (leading 0's not shown)	**Data** (address in decimal)
0000 (0)	**Y**	110	Mem[96]
0001 (1)	N		
0010 (2)	N		
0011 (3)	N		
0100 (4)	**Y**	100	Mem[68]
0101 (5)	**Y**	0	Mem[180]
0110 (6)	N		
0111 (7)	**Y**	0	Mem[232]
1000 (8)	N		
1001 (9)	N		
1010 (10)	N		
1011 (11)	N		
1100 (12)	N		
1101 (13)	N		
1110 (14)	N		
1111 (15)	N		

3. Suppose the existence of a different direct-mapped cache design with a 32-bit address. Assume that each address maps to a single byte. Assume this cache scheme uses the following breakdown of the address bits:

Tag	**Index**	**Offset**
31-6 (26bits)	5-3 (3bits)	2-0 (3bits)

a) What is the cache block size (in words)?

$3\, offsets = \frac{2^3\, bytes}{block} * \frac{words}{4\, bytes} = \frac{8}{4}\, words$ = **2 words**

b) How many entries does this cache have?

$3\, Index = 2^3\, rows = 8\, rows\, of\, index$

$8\, rows * 8\, bytes$ = **64 bytes**

c) Including space for the valid bits, tags, and actual block data, how many bits would be required to actually implement this hypothetical cache?
Total Cache size = (# of Rows) *(valid bit + tag bits + data bits)

$Cache = 8 * \left(1 + 26 + \frac{8\, bytes\ per\ block}{8\, bits}\right)$ **= 728bits**

4. Again assuming an initially empty cache, repeat the steps described in problem 2.

a) What was the number of cache hits?

Address (binary)	Address	Block #	Index (mod8)	H/M
000 000 000	0	0	0	Miss
000 000 100	4	0	0	Hit
000 010 000	**16**	**2**	**2**	**Miss**
010 000 100	132	16	0	Miss*
011 101 000	**232**	**29**	**5**	**Miss**
010 100 000	**160**	**20**	**4**	**Miss**
10000 000 000	1024	128	0	Miss*
000 011 110	30	3	3	Miss
010 001 100	**140**	**17**	**1**	**Miss**
110000 011 100	**3100**	**387**	**3**	**Miss***
010 110 100	**180**	**22**	**6**	**Miss**
100010 000 100	**2180**	**272**	**0**	**Miss***

* means replaced

1 Hit.

(Bold indicates final state of the index)

b) Create a table to show the final state of the cache including the value of each Index, Valid bit, and Tag.

Index	**Valid**	**Tag** (leading 0's not shown)	**Data** (address in decimal)
0000 (0)	**Y**	100010	Mem[272]
0001 (1)	N		
0010 (2)	**Y**	0	Mem[16]
0011 (3)	**Y**	110000	Mem[387]
0100 (4)	**Y**	10	Mem[160]
0101 (5)	**Y**	11	Mem[232]
0110 (6)	**Y**	10	Mem[180]
0111 (7)	N		
1000 (8)	N		