Problem Set 9

Justin Ely

605.411 Foundations of Computer Architecture

08 November, 2016

1a)

For direct mapping, $log_2(128) = 7$ is the number of bits in the offset field and $log_2(131072) = 17$ is the number of bits in the line field.

	Line	Offset
1000 0010	0101 0001 1100 1010 1	100 1100

1b)

For a 2-way set associative, there will be 2 lines per set. For this, the offset stays at 7 bits, and the set becomes $log_2(\frac{131072}{2}) = 16$.

	Set	Offset
1000 0010 0	101 0001 1100 1010 1	100 1100

1c)

For 4-way set associative, there will be 4 lines per set. With a 64-bit line, the offset will be $log_2(64) = 6$. The set will be $log_2(\frac{131072}{4}) = 15$.

	Set	Offset
1000 0010 010	1 0001 1100 1010 11	00 1100

1d)

A 1024-byte line size results in an offset which is $log_2(1024) = 10$ bits. For fully associative, the rest of the address becomes the tag:

Tag	Offset
1000 0010 0101 0001 1100 10	10 1100 1100

2a)

According to the table in lecture: if the bits are currently 001, a miss will result in replacing way 2.

2b)

Startin with 001, a hit on way 2 will cause:

- Bit 2 to be set to 1
- Bit 1 to remain unchanged at 0
- Bit 0 to be set to 0

Which will result in bits 100.

3a)

$Ave = .9 \times .5ns + .1(.5 \times (.5ns + 50ns)) = 5.5ns.$

3b)

$Ave = .9 \times .5ns + .1(.6 \times (.5ns + 5ns) + .4 \times (.5ns + 5ns + 50ns)) = 3ns.$

4a)

$$\begin{aligned} 10 &= h \times 8 + (1-h) \times 60 && (1)\\ 10 &= 8h + 60 - 60h && (2)\\ -50 &= 8h - 60h && (3)\\ -50 &= -52h && (4)\\ .9625 &= h && (5) \end{aligned}$$

4b)

$$\begin{aligned} 10 &= h \times 8 + (1-h) \times (60+8) && (6)\\ 10 &= 8h + 68 - 68h && (7)\\ -58 &= 8h - 68h && (8)\\ -58 &= -60h && (9)\\ .9666 &= h && (10) \end{aligned}$$

4)

c) access time/address space

5a)

The virtual address will be 4 bits for the page number and 128 bits (64 bytes) for the offset, which totals 132 bits.

5b)

The physical address will be 3 bites for the frame number and 128 bits (64 bytes) for the offset, which totals 131 bits.

5c)

The value 0x97 doesn't seem like enough bits to give an address for this problem, so i may not be completely understanding this, however:

The leftmost 4 bits (1001) will be used to index the page table. If the frame at that index does exist, then the bits (001) will be used to find the correct frame.

6a)

$Ave = .9 \times .6ns + .1(.6 + 70ns) = 7.6ns.$

6b)

$Ave = .9 \times .6ns + .1(.1 \times (.6 + 6) + .9 \times (.6 + 6 + 70)) = 7.5ns.$