Computer Architecture Theory + Lab (CS 305/341)

Assignment 8: Memory Hierarchy

(Theory Assignment 4)

Bhaskar Gupta - 180050022
Niraj Mahajan - 180050069
Shivam Goel - 180050098
Tathagat Verma - 180050111

1. A machine has two levels of cache. The miss rates of L1 and L2 are respectively 40 and 20 per 1000 memory references. (Of the 40 references missed by L1 and picked up by L2, 20 are missed). The hit times of L1 and L2 are respectively 1 and 10 cycles while the miss time of L2 is 200 cycles. Assume that 50% of all instructions executed are Load/Store.

AMAT1 -> average memory access time of L1
HT1 -> hit time of L1 = 1 cycle
h1 -> hit rate of L1 = 960/1000 = 0.96
AMAT2 -> average memory access time of L2
HT2 -> hit time of L2 = 10 cycles
h2 -> hit rate of L2 = 20/40 = 0.5
MP2 -> miss penalty of L2 = 200 cycles
SI -> average number of stalls due to instruction memory
SD -> average number of stalls due to data memory
SI2 -> average number of stalls due to instruction memory without L2
SD2 -> average number of stalls due to data memory without L2
LS -> fraction of load/store instructions = 0.5

(a) The average memory access time of L1 cache is ______5.4______ cycles .

AMAT1 = HT1 + (1 - h1) * AMAT2
= 1 + (1 - 0.96) * 110 (AMAT2 calculated in the next part)
= 5.4 cycles

(b) The average memory access time of L2 cache is ____110______ cycles .

AMAT2 = HT2 + (1 - h2) * MP2
= 10 + 0.5*200

= 110 cycles

(c) The average number of stalls per instruction is _____6.6______ cycles.

SI = (1 - h1) * AMAT2
= 0.04 * 110
= 4.4 cycles

SD = LS * ((1-h1) * AMAT2)
= 0.5 * 4.4
= 2.2 cycles

Average number of stalls = SI + SD
= 6.6 cycles

(d) The average number of stalls per instruction assuming the presence of only L1 but not L2 is ______12_______ cycles .

SI2 = (1 - h1) * MP2 (since miss penalty for L1 in this case will be MP2)
= 0.04 * 200
= 8 cycles

SD2 = LS * ((1-h1) * MP2)
= 0.5 * 8
= 4 cycles

Average number of stalls without L2 = SI2 + SD2
= 12 cycles

2. Consider a toy 32-byte D-cache accessed using physical addresses. Assume the replacement policy is LRU and the MM update policy is write back with write allocate on a write miss.

```
        li $2, 0
        addi $3, $0, 5
here:   lw $1, arr($2)
        addi $2, 4
        sw $1, arr($2)
stride: addi $2, 16
        addi $3, -1
        bnez $3, here
```

Assume that the start address (physical) of array, arr is 0x100 and that the width of the physical address is 16 bits on this toy machine. The questions below pertain to the execution of the above program on this machine.

(i) The number of D-cache misses assuming D-cache is direct-mapped with line size = 4 bytes is ___10___ .

Since the line size is 4 bytes and lw/sw access 4 bytes i.e an entire line(and $2 is a multiple of 4, word aligned) a cache hit will occur only when the word we access has been accessed before (this is a necessary condition as only 1 word will be loaded into the line and put into the cache). Since $2 is always incremented, different words are accessed every time. Hence a cache hit never occurs as no word is accessed more than once. Thus every data access results in a D-cache miss. Since there are 10 accesses, number of D-cache misses = number of accesses = 10

(ii) The number of D-cache misses assuming D-cache is direct-mapped but with line size = 8 bytes is ____7____ .

Line size = 8 bytes → 3 bits for offset
Number of lines in cache = 32/8 = 4 lines → 2 bits for indexing into the cache
Rest 16-3-2 = 11 bits are tag bits

8 Least significant bits of physical addresses accessed shown (8 higher significant bits are same for all: 0000 0001)

Tag mod 8 shown as the higher bits of tag are common for all

If we number the bits of physical address 0, 1, 2, … with 0 being the least significant bit, then bits 0, 1, 2 correspond to offset, bits 3, 4 correspond to the cache line index, bits 5,6,7 correspond to tag.

Physical addr (least sig. 8 bits)	Cache line index	Tag mod 8	Hit/ Miss	Tags in cache lines(decimal)			
				line0	line1	line2	line3
0000 0000	0	0	M	-	-	-	-
0000 0100	0	0	H	0	-	-	-
0001 0100	2	0	M	0	-	-	-

0001 1000	3	0	M	0	-	0	-
0010 1000	1	1	M	0	-	0	0
0010 1100	1	1	H	0	1	0	0
0011 1100	3	1	M	0	1	0	0
0100 0000	0	2	M	0	1	0	1
0101 0000	2	2	M	2	1	0	1
0101 0100	2	2	H	2	1	0	1

The table above shows how the cache will keep updating with each access to data and corresponding misses and hits

(iii) In the instruction at label "stride", if 16 were replaced by 12, the answer to (ii) above would be _____5_____.

Modifying the table made in previous part:

Physical addr (least sig. 8 bits)	Cache line index	Tag mod 8	Hit/ Miss	Tags in cache lines			
				line0	line1	line2	line3
0000 0000	0	0	M	-	-	-	-
0000 0100	0	0	H	0	-	-	-
0001 0000	2	0	M	0	-	-	-
0001 0100	2	0	H	0	-	0	-
0010 0000	0	1	M	0	-	0	-
0010 0100	0	1	H	1	-	0	-
0011 0000	2	1	M	1	-	0	-
0011 0100	2	1	H	1	-	1	-
0100 0000	0	2	M	1	-	1	-
0100 0100	0	2	H	2	-	1	-

The tag value increases on every alternate data access, that causes misses on lw instructions. On sw, the word accessed is the 2nd half of the line that was updated in cache, resulting in a hit. Hence half the number of data accesses result in cache misses, 5 in total.

(iv) The number of D-cache misses assuming D-cache is 3-way set associative with line size = 4 bytes is _____10_____ .

Line size = 4 bytes
Number of lines in cache = 32/4 = 8 lines.
Number of sets = ceil(8/3) = 3
Size of sets = 3,3,2 for set0, set1, set2 respectively

Physical addresses accessed (in decimal):
256 + (0, 4, 20, 24, 40, 44, 60, 64, 80, 84) {256 added to each element, 0x100}
Removing the least significant 2 bits from above, corresponding to offset (divide by 4 i.e line size)
64 + (0, 1, 5, 6, 10, 11, 15, 16, 20, 21) = (64, 65, 69, 70, 74, 75, 79, 80, 84, 85)
Set indices : (y mod s , s=3) = 1, 2, 0, 1, 2, 0, 1, 2, 0, 1
Tag (y/s, s=3) = 21, 21, 23, 23, 24, 25, 26, 26, 28, 28

Phy addr	Set index	Tag	Set 0 tags	Set 1 tags	Set 2 tags	Hit/Miss
64	1	21	-	-	-	M
65	2	21	-	21	-	M
69	0	23	-	21	21	M
70	1	23	23	21	21	M
74	2	24	23	21, 23	21	M
75	0	25	23	21, 23	21, 24	M
79	1	26	23, 25	21, 23	21, 24	M
80	2	26	23, 25	21,23,26	21, 24	M
84	0	28	23, 25	21,23,26	24,26	M
85	1	28	23,25,28	21,23,26	24,26	M
Final states $\rightarrow$			23,25,28	23,26,28	24,26	

As shown in the table above, updating the caches using LRU, we observe there to be 10 misses

(v) On program termination, the values of the tags in the set bearing index value = 1 in case (iv) above is/are
________23, 26, 28______ . 23 is least recently used, 28 is most recently used

Shown in the table made for part (iv)