Jonathan Poisson | Homework 3

1

a

Cache block = 2^offset-bits = 2^(4) bytes = 2^2 words = **4 words**

b

entries = 2^index-bits = 2^(6) lines = **64 entries**

c

Total bits for implementation: 2^index * (block size + tag size + Valid bit)

2^index = 2^6 = 64 bits
block size = 2^offset = 2^4 => 16bytes * 8 = 128 bits
Tag size = 31-10 = 22 bits

= 64bits * (128bits + 22bits + 1bit) = **9664 bits**

Data storage bits: 2^index * blocksize

= 64bits * 128bits = **8192bits**

2

a

The cache has a total of 64 * 1024/32 = 2048 cache lines. The only miss occurs is when the 0 byte is accessed, a *cold cache* miss. However, the entire 32 byte line is cached so we get 1 hit and 15 misses. This pattern continues for all *512* 1024/32 = 16384* blocks in this working set (512KB) which leads to a cache miss rate of *1/16 = 6.25%*. This miss rate is completely insensitive to the size of the working set and the size of the cache. Only the cache line size plays in role in the hit rate.

b

Cache block size = 16 bytes

8 entries and 1 miss, *1/8* = **12.5%**

Cache block size = 64 bytes

32 entries and 1 miss, *1/32* = **3.12%**

Cache block size = 128 bytes

64 entries and 1 miss, *1/64* = **1.56%**

c

If using two stream buffers are used and we can assume that the cache latency is such that a cache block can be loaded before the computation on the previous cache blocks completion, we can say the miss rate will be *0 (zero)*

3

a

Number of index bits in a cache = 2^n * (block size + tag size + valid field size)

Cache A

64 KiB = 2^16
1 word = 2^0
tag size = 32 - (16 + 0 + 2)
Number of bits = 2^16 * (1 * 32 + (32 - 16 - 2) + 1) = **3080192 bits**

Cache B

64 KiB = 2^16
2 word = 2^1
tag size = 32 - (16 + 1 + 2)
Number of bits = 2^16 * (2 * 32 + (32 - 16 - 1 - 2) + 1) = **5111808 bits**

b

I think tag size changes?

Cache A

64 KiB = 2^16
1 word = 2^0
tag size = 32 - (14 + 0 + 2)
Number of bits = 2^16 * (1 * 32 + (32 - 14 - 2) + 1) = **3211264 bits**

Cache B

64 KiB = 2^16
2 word = 2^1
tag size = 32 - (14 + 1 + 2)
Number of bits = 2^16 * (2 * 32 + (32 - 14 - 1 - 2) + 1) = **5242880 bits**

4

a

Cycle time = L1 hit time

Clock rate = 1 / cycle time

L1 hit time for P1 = .55 ns
P1 clock rate = 1/.55 ns = **1.81 GHz**

L1 hit time for P2 = 1.70 ns
P1 clock rate = 1/1.70 ns = **.59 GHz**

b

Miss time in cycles = Main memory access time / L1 hit time

*AMAT = (L1 hit time) + [(L1 Miss Rate) * (L1 Miss Time)]*

P1 miss time in cycles = 70/.55 = *127 cycles*
P1 miss time = 127 cycles * .55 ns = *69.85 ns*
AMAT for P1 = .55 + (.075 * 69.85) = **5.79 ns**

P2 miss time in cycles = 70/1.70 = *41 cycles*
P2 miss time = 41 cycles * 1.70 ns = *69.7 ns*
AMAT for P2 = 1.70 + (.07 * 69.7) = **6.58 ns**

c

Data miss cycle for P1 = I 7.5% * 36% * 100* = *2.7I*
Total memory stall cycles = *7.50I + 2.7I = 10.2I*
Total CPI for P1 = *1.00I + 10.2* = **11.2 clock cycles**

Data miss cycle for P2 = I 7.0% * 36% * 100* = *2.52I*
Total memory stall cycles = *7.0I + 2.52I = 9.52I*
Total CPI for P2 = *1.00I + 9.52* = **10.52 clock cycles**

d

AMAT for L2 for P2 = 5.62 + .50 * 100 = 55.62 clock cycles
AMAT for L1 for P2 = 1.70 + .07 * 100 = 8.7 clock cycles
AMAT for P2 = 55.62 + 8.7 = **64.32 clock cycles**

The addition of L2 results in a longer AMAT

e

L2 cache whole data miss cycle for P2 = I .50 * .36 * 100* = *18I*
Total memory stall cycles for L2 for P2 = *50I + 18I = 68I*
Total CPI for L2 for P2 = *1.00 + 68* = 69

Total CPI for P2 with L1 and L2 = *69 + 10.52* = **79.52 clock cycles**

f

When using the L2 cache with P2, P1 is faster because of L2's poor performance

5

a

0 Hits

b

1 hit

6

Page Table size = Number of page table entries * Size of a page table entry
Page table size = 2^4 * 2^20 = **16 MiB**

Number of memory references = Number of page table entries
= **2^22**

a

- The first reason for entry 2's valid bit to be set to 0 is if process context switch occurrs. In this case, the access will not be given to the valid ID 2
- The second reason for entry 2's valid bit to be set to 0 is if the page is not present in the memory (along with the valid bit, the dirty bit is 0) and a true page fault occurs

b

When instrucion writes to VA page 30, a TLB miss will occur. The steps that take place are ...

- Copy the page in TLB from memory because of the TLB miss for page 30
- Replace page in TLB
- Write instruction for that page and set the modified bit

A Software managed TLB is faster where there are not many TLB misses that occur and the instructions to resolve the miss live in the L1 cache.

c

When instruction tries to write to VA Page 200, nothing will happen because the location is protected by RO (Read-Only).