Hw 7 Solutions

5.5

5.5.1

The byte offset in the address is 5 bits. This means that the cache block size in bytes must be 2^5 = 32 bytes. Since typically 1 word = 4 bytes (for 32-bit architectures), this means that the block has $\frac{32\ bytes}{4\ bytes\ per\ word}$ = 8 words.

5.5.2

There are 5 index bits. This means that the cache has 2^5 = 32 entries.

5.5.3

Data Storage for Cache:

The cache has 32 entries. Since it is direct-mapped, each entry has 1 block, each of which contains 8 words. Since a word has 4 bytes, this means that each block stores a total of 8 words/block * 4 bytes/word = 32 bytes/word. Therefore, the total number of bits in the cache used for data storage is:

Cache data bits = 32 lines * 32 bytes/word = 1024 bytes = 8192 bits.

Total Storage for Cache

An address has 22 tag bits.

In addition to the data, each entry contains 22 tag bits and 1 valid bit. Thus, the total bits required = 8192 + 22*32 + 1*32 = 8928 bits.

Ratio of Total Cache Storage to Data Cache Storage = 8928 bits/8192 bits = 1.09

5.5.4

Note: the upper 20 bits of the binary addresses are all 0's.

Byte Address	Binary Address	Tag	Index	Offset	Hit/Miss	Bytes Replaced
0x00	0000 0000 0000	0x0	0x00	0x00	M	
0x04	0000 0000 0100	0x0	0x00	0x04	H	
0x10	0000 0001 0000	0x0	0x00	0x10	H	
0x84	0000 1000 0100	0x0	0x04	0x04	M	
0xe8	0000 1110 1000	0x0	0x07	0x08	M	
0xa0	0000 1010 0000	0x0	0x05	0x00	M	
0x400	0100 0000 0000	0x1	0x00	0x00	M	0x00-0x1F
0x1e	0000 0001 1110	0x0	0x00	0x1e	M	0x400-0x41F
0x8c	0000 1000 1100	0x0	0x04	0x0c	H	
0xc1c	1100 0001 1100	0x3	0x00	0x1c	M	0x00-0x1F
0xb4	0000 1011 0100	0x0	0x05	0x14	H	
0x884	1000 1000 0100	0x2	0x04	0x04	M	0x80-0x9f

5.5.5

Hit ratio = 4/12 = 33%

5.5.6

<index, tag, data>

<0, 3, Mem[0xC00] - Mem[0xC1F]>

<4, 2, Mem[0x880] - Mem[0x89F]>

<5, 0, Mem[0x0A0] - Mem[0x0BF]>

<7, 0, Mem[0x0E0] - Mem[0x0FF]>

5.10

5.10.1

P1 Clock Rate: $\frac{1}{0.66*10^{-9}\ s} = 1.515\ GHz$

P2 Clock Rate: $\frac{1}{0.90*10^{-9}\ s} = 1.11\ GHz$

5.10.2

AMAT = L1 Hit Time + L1 Miss rate * Main Memory access time

Note that we first need to find out the number of cycles needed for an L1 hit and miss for both processors.

By the problem definition, an L1 Hit will complete in 1 cycle for both processors.

Main memory cycles:

- P1:

70 ns / (0.66 ns/cycle) = 106.06 cycles => 107 cycles (since we need whole number of cycles to complete operation).

- P2:

70 ns / (0.90 ns/cycle) = 77.77 cycles => 78 cycles

P1 AMAT = 1 cycle + 0.08 * 107 cycles = 9.56 cycles 9.56 cycles*(0.66 ns/cycle) = 6.31 ns

P2 AMAT = 1 cycle + 0.06 * 5.68 cycles 5.68 cycles*(0.90 ns/cycle) = 5.112 ns

5.10.3

P1:

MCPI = (Instruction access) 1*0.08*107 + (Data access) 0.36*0.08*107 = 11.64

TCPI = BCPI + MCPI = 1 + 11.64 = 12.64

P2:

MCPI = (Instruction access) 1*0.06*78 + (Data access) 0.36*0.06*78 = 6.36

TCPI = BCPI + MCPI = 1 + 6.36 = 7.36

5.10.4

AMAT = L1 Hit Time + L1 Miss rate * (L2 Hit Time + L2 Miss rate * Main memory access time)

Note that we first need to find out the number of cycles needed for an L1 hit, an L2 hit, and a main memory access for P1

By the problem definition, an L1 Hit will complete in 1 cycle for both processors.

L2 Hit cycles:

5.62 ns / (0.66 ns/cycle) = 8.52 cycles => 9 cycles

Main memory cycles:

- P1: 107 cycles

P1 AMAT = 1 cycle + 0.08 * (9 cycles + 0.95*107 cycles) = 9.852 cycles 9.852 cycles*(0.66 ns/cycle) = 6.502 ns

The AMAT for P1 is worse when using the L2 cache.

5.10.5

P1:

MCPI = (Instruction access) 1*0.08*(9 + 0.95 * 107) + (Data access) 0.36*0.08*(9 + 0.95*107) = 12.04

TCPI = BCPI + MCPI = 1 + 12.04 = 13.04

5.10.6

Because the clock cycle time and percentage of memory instructions is the same for both versions of P1, it is sufficient to focus on AMAT (in terms of cycles). We want

AMAT with L2 < AMAT with L1 only

Let m be L2 miss rate

1 + 0.08*(9 + m*107) < 9.56 cycles

Solving for m gives us

m < 0.916

5.10.7

We want P1's average time per instruction to be less than P2's average time per instruction.

Average time per instruction = Average CPI*clock time

P2's average time per instruction = 7.36*0.9 = 6.62 ns

This means that we want

Average P1 Time per instruction < 6.62 ns

CPI of P1 * 0.66 < 6.62 ns

CPI of P1 < 10.03

Let m be the miss rate of the L2 cache for P1.

CPI of P1 = Base CPI + MCPI = 1 + (Instruction access) 1*0.08*(9 + m * 107) + (Data access) 0.36*0.08*(9 + m*107) < 10.03

Solving for m gives us

m < 0.692

5.12

5.12.1

Standard memory time: Each cycle on a 2-Ghz machine takes 0.5 ns. Thus, a main memory access requires 100/0.5 = 200 cycles.

First level cache only: CPI = 1.5 + 0.07*200 = 15.5

Direct-Mapped L2: CPI = 1.5 + 0.07*(12 + 0.035*200) = 2.83

8-waySet-associative L2: CPI = 1.5 +0.07*(28 + 0.015*200) = 3.67

Double memory time: main memory access requires 200/0.5 = 400 cycles.

First level cache only: CPI = 1.5 + 0.07*400 = 29.5 (90% increase)

Direct-Mapped L2: CPI = 1.5 + 0.07*(12 + 0.035*400) = 3.32 (17% increase)

8-waySet-associative L2: CPI = 1.5 +0.07*(28 + 0.015*400) = 3.88 (5% increase)

5.12.2

CPI = 1.5 + 0.07 × (12 + 0.035 × (50 + 0.13 × 200)) = 2.53

Compare to Direct-Mapped L2 with standard memory CPI: 2.83

Adding the L3 cache does improve the overall performance, which is the main advantage of having an L3 cache. The disadvantage is that the L3 cache takes real estate away from having other types of resources, such as functional units.

5.12.3

Compare to Direct-Mapped L2 with standard memory CPI: 2.83

We want the CPI of the CPU with an external L2 cache to be at most 2.83.

Let x be the necessary miss rate.

New CPI < Original CPI

$1.5 + 0.07*(50 + x*200) < 2.83$

Solving for x gives that $x < -0.155$. Th is means that even if the miss rate of the L2 cache was 0, a 50-ns access time gives a CPI of $1.5 + 0.07*(50 + 0*200) = 5$, which is greater than the 2.83 given by the on-chip L2 caches. As such, no size will achieve the performance goal.