5.2

5.2.1

4 bits are required to represent the block address since there are 16 blocks, no bits are needed for the offset as it is only one word per block. The addresses are supposed to be 32 bits but that would take up a lot of space so I just used the minimum # of bits needed to represent the largest number in the example, which was 253.

|  |  |  |  |
| --- | --- | --- | --- |
| Ref/Address | Tag | Index | Hit or miss |
| 3  0000 0011 | 0000 | 0011 | Miss |
| 180  10110100 | 1011 | 0100 | Miss |
| 43  00101011 | 0010 | 1011 | Miss |
| 2  00000010 | 0000 | 0010 | Miss |
| 191  10111111 | 1011 | 1111 | Miss |
| 88  01011000 | 0101 | 1000 | Miss |
| 190  10111110 | 1011 | 1110 | Miss |
| 14  00001110 | 0000 | 1110 | Miss |
| 181  10110101 | 1011 | 0101 | Miss |
| 44  00101100 | 0010 | 1100 | Miss |
| 186  10111010 | 1011 | 1010 | Miss |
| 253  11111101 | 1111 | 1101 | Miss |

5.2.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Address | Tag | Index | Offset | Hit or miss |
| 3  00000011 | 0000 | 001 | 1 | Miss |
| 180  10110100 | 1011 | 010 | 0 | Miss |
| 43  00101011 | 0010 | 101 | 1 | Miss |
| 2  00000010 | 0000 | 001 | 0 | Hit |
| 191  10111111 | 1011 | 111 | 1 | Miss |
| 88  01011000 | 0101 | 001 | 0 | Miss |
| 190  10111110 | 1011 | 111 | 0 | Hit |
| 14  00001110 | 0000 | 111 | 0 | Miss |
| 181  10110101 | 1011 | 010 | 1 | Hit |
| 44  00101100 | 0010 | 110 | 0 | Miss |
| 186  10111010 | 1011 | 101 | 0 | Miss |
| 253  11111101 | 1111 | 110 | 1 | Miss |

5.6

5.6.1

Clock rate is the inverse of cycle time so:

P1: 1/.66ns =1.51GHz

P2: 1/.90ns = 1.11GHz

5.6.2

AMAT = Time for a hit + miss rate x miss penalty

P1: .66ns + .08 x 70ns = 6.26ns

P2: .9ns +.06 X 70ns = 5.1ns

5.6.3

CPI = BCPI + MCPI

MCPI = access/instruction X miss rate X miss penalty

Miss penalty for p1: 70/.66 =106.06 = 107 clock cycles

Miss penalty for p2: 70/.9 = 77.77 = 78 clock cycles

P1: CPI = 1 + (.36 X .08 X 107) = 4.0816

P2: CPI = 1 + (.36 X .06 X 78) = 2.6848

P1: (1.51GHz / 4.0816) =.369

P2: (1.11GHz / 2.6848) = .413

P2 is faster

5.6.4

AMAT= L1 hit time + L1 miss rate \* L1 miss penalty

Since there is now a L2 cache the penalty for missing in L2 is now the AMAT for L2

Hit time for L2 = 5.62/.66 =8.5 = 9 clock cycles

P1: 1 +.08 X (9 + .95 \*107) = 9.852 cycles

9.852 X .66ns = 6.502ns

The AMAT got worse with the addition of an L2 cache.

5.6.5

1 + .36 X .08 X (9 + .95 X 107) =4.1867 clock cycles

4.1867 X .66ns = 2.7632

5.6.6

P1: (1.51GHz / 4.1867) = .3606

P2: (1.11GHz / 2.6848) = .4134

P2 is still faster than p1 with the addition of L2.

|  |  |  |  |
| --- | --- | --- | --- |
| address | Tag | Index | Hit or miss |
| 3  00000011 | 00000011 | 0 | Miss |
| 180  10110100 | 10110100 | 1 | Miss |
| 43  00101011 | 00101011 | 2 | Miss |
| 2  00000010 | 00000010 | 3 | Miss |
| 191  10111111 | 10111111 | 4 | Miss |
| 88  01011000 | 01011000 | 5 | Miss |
| 190  10111110 | 10111110 | 6 | Miss |
| 14  00001110 | 00001110 | 7 | Miss |
| 181  10110101 | 10110101 | 0 | Miss |
| 44  00101100 | 00101100 | 1 | Miss |
| 186  10111010 | 10111010 | 2 | Miss |
| 253  11111101 | 11111101 | 3 | Miss |

5.7

|  |
| --- |
| M[181] |
| M[44] |
| M[186] |
| M[253] |
| M[191] |
| M[88] |
| M[190] |
| M[14] |

5.7.2

3, 180, 43, 2, 191, 88, 190, 14, 181, 44, 186, 253

|  |  |
| --- | --- |
| Word 0 | Word 1 |
| M[186] | M[187] |
| M[252] | M[253] |
| M[180] | M[181] |
| M[44] | M[45] |

5.7.3

3, 180, 43, 2, 191, 88, 190, 14, 181, 44, 186, 253

LRU

2 hits : M[2] and M[190]

Hit rate = 1/6

|  |  |
| --- | --- |
| Word 0 | Word 1 |
| M[2] | M[3] |
| M[180] | M[181] |
| M[42] | M[43] |
| M[252] | M[253] |

MRU

2 hits m[2] and m[181]

Hit rate = 1/6

They have the same hit rate for fully associative cache. Either replacement policy would be the best for this cache.

5.7.4

MR = .07

CR = 1/2 =.5ns

L1 miss penalty = 12 cycles

L2 miss penalty = 100/.5 =200 cycles

L1 cache: 1.5+(.07 X 200) =15.5

L2 cache direct map: 1.5 + (.07 X 12) + (.035 X 200) =9.34

L2 eight-way set: 1.5 + (.07 X 28) + (.015 X 200) =6.46

If the time to access main memory is cut in half then that means the penalty for an L2 miss is lower so the CPI would be lower, and if it were double the penalty for an L2 miss would be higher so the CPI would be higher.

5.7.5

Since adding this L3 would reduce the global miss rate and in turn reduce the chance at having to access main memory. The miss penalty for L2 would become 50ns and the miss penalty for L3 would be 200ns. One of the advantages of the L3 cache is that it is a larger cache so there would be less capacity misses. Since this type of cache is larger and farther away from the core it is much slower than L1 and L2 caches but still way faster than main memory.