1. (1) C = ABS

16K = 4 \* 64 \* S

S = 64

Associativity (A): 4

Cache Size (C): 16K

Block Size (B): 64 bytes

**Offset** = log(B) = log(64) = **6 bits**

**Index** = log(S) = log(64) = **6 bits**

**Tag** = 64 – Offset – Index = 64 – 6 – 6 = **52 bits**

(2)

Size of tag array = (tag + dirty bit + valid bit) \* number of blocks

Number of Blocks – cache size / block size = (16 \* 1024) bytes / 64 bytes = 256

Size of tag array = (52 + 1 + 1) \* 256 = 13824 bits

Data Array = 16 \* 1024 \* 8 = 131072 bits

Total bits needed for building cache = 13824 + 131072 = **144896 bits**

1. (1) C = ABS (from number 1)

Offset = 6 bits

Index = 6 bits

Tag = 52 bits

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Array | I | J | Address | Tag | Index | Offset | Hit/miss? |
| A | 0 | 0 | 20000 | 00100000 | 000000 | 000000 | Miss/com |
| B | 0 | 0 | 40000 | 01000000 | 000000 | 000000 | Miss |
| A | 0 | 1 | 20400 | 00100000 | 010000 | 000000 | Miss/com |
| B | 0 | 1 | 40008 | 01000000 | 000000 | 001000 | Hit |
| A | 0 | 2 | 20800 | 00100000 | 100000 | 000000 | Miss/com |
| B | 0 | 2 | 40010 | 01000000 | 000000 | 010000 | Hit |
| A | 0 | 3 | 20C00 | 00100000 | 110000 | 000000 | Miss/com |
| B | 0 | 3 | 40018 | 01000000 | 000000 | 011000 | Hit |
| A | 0 | 4 | 21000 | 00100001 | 000000 | 000000 | Miss/con |
| B | 0 | 4 | 40020 | 01000000 | 000000 | 100000 | Hit |
| A | 0 | 5 | 21400 | 00100001 | 010000 | 000000 | Miss/con |
| B | 0 | 5 | 40028 | 01000000 | 000000 | 101000 | Hit |
| A | 0 | 6 | 21800 | 00100001 | 100000 | 000000 | Miss/con |
| B | 0 | 6 | 40038 | 01000000 | 000000 | 111000 | Hit |
| A | 0 | 7 | 21C00 | 00100001 | 110000 | 000000 | Miss/con |
| B | 0 | 7 | 40040 | 01000000 | 000001 | 000000 | Miss |
| A | 0 | 8 | 22000 | 00100010 | 000000 | 000000 | Miss/con |
| B | 0 | 8 | 40048 | 01000000 | 000001 | 001000 | Hit |
| A | 0 | 9 | 22400 | 00100010 | 010000 | 000000 | Miss/con |
| B | 0 | 9 | 40050 | 01000000 | 000001 | 010000 | Hit |

A will always miss so the miss rate for A is 1

B misses when j % 8 = 1 so 1/8 of the time

Therefore, we get that the cache miss rate = (1 + 1/8) / 2 = **0.5625**

(2)

Array A and Array B are both accessed N \* N times which is 128 \* 128

For array A, it will have a compulsory miss: (128 / 8) \* 128 = 2048 misses

For array B, every miss is a compulsory miss, and they happen 1/8 of the time, so 1/8 \* (128\*128) = 2048 misses

Adding those two together, get 2048 + 2048 = 4096

Conflict misses are all the other misses that aren’t compulsory misses. Since all of array A’s are misses and none of B are conflict misses we just calculate A total misses – compulsory so: (128 \* 128) – 2048 = 14336

**Compulsory Misses: 4096; Conflict Misses: 14336**

1. (1)

25% load/store instructions

L1 I-Cache: 2% miss rate

L1 D-Cache: 5% miss rate (50% replaced are dirty)

L2 U-Cache: 12 ns access time; 20% miss rate (25% replaced are dirty)

Main memory access latency: 60 ns

Overall CPI = CPIbase + miss\_rate\*miss\_penalty

60 ns \* 2 GHz = 120 cycles if hit main memory

12 ns \* 2 GHz = 24 cycles if hit L2

CPI = 1 + 100%\*(2%\*(24 + 20%\*((1 + 25%) \* 120))) + 25%\*(5%\*(1 + 50%)\*(24 + 20%\*((1 + 25%) \* 120))) = **3.0925**

(2) 3 GHz instead of 2 GHz

60 ns \* 3 GHz = 180 cycles

12 ns \* 3 GHz = 36 cycles

CPI = 1 + 100%\*(2%\*(36 + 20%\*((1 + 25%) \* 180))) + 25%\*(5%\*(1 + 50%) \* (36 + 20%\*((1 + 25%) \* 180))) = 4.13875

Assume IC is the same for both

ET = IC \* CPI \* CT

ET2GHz = 3.0925 \* 2 GHz = 1.5 \* 109

ET3GHz = 4.13875 \* 3 GHz = 1.3 \* 109

Speedup: 1.5 \* 109 / 1.3 \* 109 = **1.12x**