# Cache Performance

## Question 1

1. Block Offset – log2 64 = 6

Cache set = 256000/(64\*4) =1000

Index bits = log2 1000 ~ 10

Tag bits = 32 – (10+6) + 16

1. CP I = CP I *execution* + StallCyclesPerInstruction

CP I = 1

For non-zero miss rate, let us compute StallCyclesP erInstruction StallCyclesPerInstruction = (Memory accesses per instruction) \* miss rate \* miss penalty Memory accesses per instruction = 1 + 0.5 (1 instruction access + 0.5 data access) StallCyclesPerInstruction = 1.5 \* 0.02 \* 25 = 0.75

CP I = 1.75

The computer with no cache misses is 1.75 times faster

## Question 2

Miss rate = 0.05 (5%)

Block size = 2 words (8 bytes)

Frequency of memory operations = 109

Frequency of writes from processor = 0.25 ∗ 109

So:

Fraction of read hits = 0.75 ∗ 0.95 = 0.7125

Fraction of read misses = 0.75 ∗ 0.05 = 0.0375

Fraction of write hits = 0.25 ∗ 0.95 = 0.2375

Fraction of write misses = 0.25 ∗ 0.05 = 0.0125

1. **Write through Cache**

Then:

No Memory access on read Hit

2 words sent to cache on read miss

A word sent to Memory on write hit

2 words sent to cache on write miss, one word sent to memory

Therefore;

Average words transferred = 0.7125 ∗ 0 + 0.0375 ∗ 2 + 0.2375 ∗ 1 + 0.0125 ∗ 3 = 0.35

Average bandwidth used = 0.35 ∗ 109

*Fractionofbandwidthused* = = 0.35

1. **Writeback cache**

No memory access on Read

On a read miss:

1. If replaced line is modified then cache must send two words to memory, and then

memory must send two words to the cache

2. If replaced line is clean then memory must send two words to the cache

No memory access on write hit

On a write miss:

1. If replaced line is modified then cache must send two words to memory, and then

memory must send two words to the cache

2. If replaced line is clean then memory must send two words to the cache

Thus:

Average words transferred = 0.7125 ∗ 0 + 0.0375 ∗ (0.7 ∗ 2 + 0.3 ∗ 4) + 0.2375 ∗ 0 + 0.0125 ∗ (0.7 ∗ 2 + 0.3 ∗ 4) = 0.13

Average bandwidth used = 0.13 ∗ 109*Fractionofbandwidthused* = = 0.13

## Question 3

CPU performance: CPU Time = IC ∗ CP I ∗ Clock Time

CP I = CPI*execution* + StallCyclesPerInstruction

Then:

CPI*execution* = 0.26 ∗ 1 + 0.09 ∗ 2 + 0.65 ∗ 1 = 1.09

**Write through**

StallCyclesPerInstruction = MRI ∗ss 50 + MRD ∗ (0.26 ∗ 50 + 0.09 ∗ 50) = 0.425

so:

CP I = 1.09 + 0.425 = 1.515

**Write back**

StallCyclesPerInstruction = MRI ∗ 50 + MRD ∗ (0.26 ∗ (0.5 ∗ 50 + 0.5 ∗ 100) + 0.09 ∗

(0.5 ∗ 50 + 0.5 ∗ 100)) = 0.5125

so:

CP I = 1.09 + 0.5125 = 1.6025