***Assume we have a computer where the CPI is 1.0 when all memory accesses (including data and instruction accesses) hit in the cache. The cache is a unified (data + instruction) cache of size 256 KB, 4-way set associative, with a block size of 64 bytes. The data accesses (loads and stores) constitute 50% of the instructions. The unified cache has a miss penalty of 25 clock cycles and a miss rate of 2%. Assume 32-bit instruction and data addresses***.

1. ***What is the tag size for the cache?***

Cache size: 256 kb

Block size: 64 bytes

No. of blocks: (256x1024)/64 = 4096

No. of sets: 4096/4 = 1024

Offset bits: log2 64 = 6

Index bits: log2 1024 = 10

Tag bits: 32 – 10 – 6 = 16

***b. How much faster would the computer be if all memory accesses were cache hits?***

Instruction: 100

Loads/stores: 50

Original CPI = 1.0

Effective CPI = 1.0 + (1.5 x 0.02 x 25) = 1.75

Speedup = 1.75/1.0 = 1.75 faster

***You purchased an Acme computer with the following features:***

***95% of all memory accesses are found in the cache.***

***Each cache block is two words, and the whole block is read on any miss.***

***The processor sends references to its cache at the rate of 109 words per second.***

***25% of those references are writes.***

***Assume that the memory system can support 109 words per second, reads or writes.***

***The bus reads or writes a single word at a time (the memory system cannot read or***

***write two words at once).***

***Assume at any one time, 30% of the blocks in the cache have been modified.***

***The cache uses write allocate on a write miss.***

***You are considering adding a peripheral to the system, and you want to know how much of the memory system bandwidth is already used. Calculate the percentage of memory system bandwidth used on the average in the two cases below. Be sure to state your assumptions***.

***a. The cache is write through.***

Hit rate: 0.95

Miss rate: 0.05

Write rate: 0.25 x10^9 = 2.5 x 10^8

Read rate: 0.75 x 10^9 = 7.5 x 10^8

Read miss rate: 0.05 x 7.5 x 10^8 = 3.75 x 10^7 read 2 words

Write miss rate: 0.05 x 2.5 x 10^8 = 1.25 x 10^7 x 3 write 1 word load 2 words

Write hit rate: 0.95 x 2.5 x 10^8 = 2.375 x 10^8 write 1 word

Total = (3.75 x 10^7 x 2) + (1.25 x 10^7 x 3) + (2.375 x 10^8 x 1)

= 3.5 x 10^8

Percentage = (3.5 x 10^8)/10^9 x 100% = 35%

***b. The cache is write back.***

Write-back => 0.3 x (3.75 x 10^7 + 1.25 x 10^7) = 1.5 x 10^7

Total = (3.75 x 10^7 + 1.25 x 10^7 + 1.5 x 10^7) x 2 = 1.3 x 10^9

Percentage = (1.3 x 10^9)/10^9 x 100% = 13%

***One difference between a write-through cache and a write-back cache can be in the time it takes to write. During the first cycle, we detect whether a hit will occur, and during the second (assuming a hit) we actually write the data. Let’s assume that 50% of the blocks are dirty for a write-back cache. For this question, assume that the write buffer for the write through will never stall the CPU (no penalty). Assume a cache read hit takes 1 clock cycle, the cache miss penalty is 50 clock cycles, and a block write from the cache to main memory takes 50 clock cycles. Finally, assume the instruction cache miss rate is 0.5% and the data cache miss rate is 1%. Assuming that on average 26% and 9% of instructions in the workload are loads and stores, respectively, estimate the performance of a write-through cache with a two-cycle write versus a write-back cache with a two-cycle write***

|  |  |
| --- | --- |
| Cache read hit time | 1 cycle |
| Cache write time (on hit) | 2 cycles |
| Cache miss penalty (read or write miss) | 50 cycles |
| Block write-back to memory | 50 cycles |
| Instruction cache miss rate | 0.5% = 0.005 |
| Data cache miss rate | 1% = 0.01 |
| Load instructions | 26% = 0.26 |
| Store instructions | 9% = 0.09 |
| Dirty block probability (write-back only) | 50% = 0.5 |
| Write buffer (write-through only) | Always absorbs writes = no stall |

1. Write-through cache

Instruction miss penalty=> 0.005 x 50 = 0.25

Data loads=> 0.26 x 0.01 x 50 = 0.13

Data stores=> 0.09 x 0.01 x 50 = 0.045

Total => 0.425 cycle per instruction

1. Write-back cache

Instruction miss penalty=> 0.005 x 50 = 0.25

Load misses=> 0.26 x 0.01 x 50 = 0.13

Store misses=> 0.09 x 0.01 x (50 + (0.5 x 50) = 0.0675

Total => 0.4475 cycles per instruction