Computer Organization (0306-550) - Winter 2012

Homework Assignment #5 - Do Not Submit

1. **Consider an initially empty 16-word cache and this memory access trace of *word addresses*:**

2, 3, 11, 16, 21, 13, 64, 48, 19, 11, 3, 22, 4, 27, 6, and 11.

This cache has three configurations:

a) Direct-mapped cache with one-word blocks.
b) Direct-mapped cache with four-word blocks.
c) Four-way set associative cache with one-word blocks.

For each of the above three cache configurations:

- **Show cache operation (contents and the hit/miss status of each reference).**
- **Find the hit rate.**

2. **Consider the three processors/cache configurations described in the table below.**

Processor	Cycle Time (ps)	Cache Configuration		Miss Rate	
		Organization	Block Size (words)	Instruction (%)	Data (%)
1	420	Direct mapped	1	4	6
2	420	Direct mapped	4	2	4
3	310	Two-way set associative	4	2	3

For each processor, the cache miss penalty is six plus the block size in words. A cache hit incurs no stall cycles. Each processor runs the same program in which half the instructions are data memory access instructions. The program running on processor 1 with its cache has a CPI of 2.0.

a) Determine which processor spends the most cycles on cache misses.
b) Determine which processor is the fastest and which one is the slowest.

3. **Given the cache with 64KB nominal capacity shown in slide 27 of lecture notes # 8.**
Assume that the physical amount of memory in the system that's cacheable is 1GB.
Show the CPU address fields and calculate the total number of bits needed (including valid and tag bits) for the cache for the following cases with the same nominal capacity of 64KB and block size:

a) Cache is directly mapped (as shown).
b) Cache is 8-way set associative.
c) Cache is 32-way set associative.
d) Cache is fully associative.

4. **For the following MIPS code (the same code given in assignment #4):**

```
loop:       lw     $1,  0($2)
            lw     $1,  8($1)
            add    $1,  $1, $8
            lw     $3,  0($4)
            lw     $3,  12($3)
            add    $1,  $1,  $3
            addi   $2,  $2,  4
            addi   $4,  $4,  4
            sw     $1,  0($5)
            sub    $6,  $7,  $2
            addi   $5,  $5,  4
            bne    $6,  $0,  loop
```

Assuming that initially: $2 = $7 - 26000 The above code is run on the following two CPUs:

a) Multi-cycle CPU (as covered in class - 3rd Edition Figure 5.28 page 323) with clock cycle = 2ns.

a) MIPS pipelined CPU version 3 with forwarding and reduced branch delay (4th Edition Figure 4.65 page 384 - 3rd Edition Figure 6.41 page 427) and clock cycle = 2ns.

Assume each CPU uses non-ideal memory in conjunction with separate single-level caches for instructions and data with the following parameters:

- **A cache hit incurs no stall cycles.**
- **A cache miss incurs M = 25 stall cycles.**
- **Instruction cache miss rate = 2%**
- **Data cache miss rate = 9%**

For each CPU while running this code find:

- **Average Memory Access Time (AMAT) in cycles.**
- **Effective CPI.**
- **MIPS rating.**
- **Execution time.**
- **How much faster is each CPU with ideal memory.**
- **The miss rate of a unified single-level cache that gives the same performance.**