CS 51 Homework 7

Jessie Li

November 9, 2024

1.

	Hits	Misses
Direct-mapped	2	8
2-way set associative	2	8
4-way set associative	1	9
Fully associative	3	7

Table 1: Hits and misses on `sample.txt` for each type of cache.

2.

	Hits	Misses
Direct-mapped	6076728	269350
2-way set associative	6185842	160236
4-way set associative	6214876	131202
Fully associative	6220013	126065

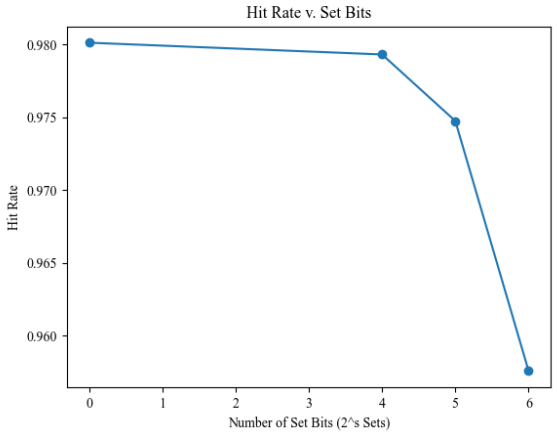

Figure 1: Hits and misses on `long-trace.txt` for each type of cache. Counted 6346078 touches in total. In the graph on the right, associativity increases from right to left as the number of set bits decreases.

The fully associative cache had the highest hit rate (97.9%), followed by 4-way associative (97.9%), 2-way associative (97.5%), and finally, directly mapped (95.8%).

These results support conventional wisdom, showing that increasing associativity improves performance but at a diminishing rate. Relative to the improvement from direct-mapped to 2-way associative (+1.7% per set bit), the improvement from 2-way to 4-way associative was much smaller (+0.2% per set bit), and even smaller from 4-way associative to fully associative (+0.1% per 4 set bits).

3.

delay = (combo delay) + (reg delay) = 270 ps

throughput = (1 instruction)/(270 ps) * 1000 ps/ns = 3.70 GIPS

4.

The register should be inserted between blocks B and C to minimize the clock cycle, which must be long enough to cover the slowest stage. Inserting here creates a 2-stage pipeline with A, B + register in the first half (80 + 30 + 20 = 130 ps) and C, D, E + register in the second half (50 + 70 + 20 + 20 = 160 ps). The second stage, the slower stage, requires a clock cycle of 160 ps.

delay = 2 stages · 160 ps/stage = 320 ps

throughput = (1 instruction)/(160 ps) * 1000 ps/ns = 6.25 GIPS

5.

```
addl %ecx, %eax      // W_valE
addl %ecx, %eax      // M_valE
addl %ecx, %eax      // e_valE
addl %ecx, %eax      // current value
```

6.

I tested my new `opli` instruction by loading `opli.hex` into ROM and slowly clocking through to confirm that the registers and condition codes were set appropriately:

	%eax	%ecx	CC (Z,S,O)
iaddl $0xdeadbeef, %eax	0xdeadbeef	0x00000000	010
iandl $0xbeef, %eax	0x0000beef	0x00000000	000
isubl $0xbeef, %eax	0x00000000	0x00000000	100
ixorl $0xfeedcafe, %eax	0xfeedcafe	0x00000000	010
iaddl $0x80000000, %eax	0x7eedcafe	0x80000000	001
iaddl $0xba11ade, %ecx	0x7eedcafe	0x0ba11ade	000