Anthony Alayo 997487401
Freddy Chen 997363124

ECE552 Lab 2 – Dynamic Branch Prediction

Micro-Benchmark Explanation

Our micro-benchmark was created to confirm that our predictors were working correctly. Our micro-benchmark is a 4 loop with several if-conditions to simulate how the predictors will catch patterns. Looking at our loop:

```
for(i = 1; i <= max; i++) {
        a = i, b = i, c = i;

        if(a%10 == 0) {
                a = 0;
        }
        if(b%3 == 0) {
                b = 0;
        }

        if(c%5 == 0) {
                c = 0;
        }

        if(a != 0) {
                a = 0;
        }
}
```

You can see I created a situation such that there will be 5 branches per loop iteration (4 if statements and the branch to the top of loop) and a variable amount of branches taken per iteration. When i = 1, the first 3 branches will be taken, the 4th branch won't be taken, and the branch for the loop iteration will be taken. That looks like T T T N T. This pattern will vary depending on the value of I, which should test the 2 bit saturating predictor and 2 level predictor well.

The first 6 patterns will be… 1) T T T N T, 2) T T T N T, 3) T N T N T, 4) T T T N T, 5) T T N T T, 6) T N T N T, ….etc

We would expect for a small number of loop iterations that the 2 bit saturating counter would out-perform the 2-level as the 2-level requires more warm up time due to the amount of tables it has to fill. We see this pattern when running the micro-benchmark.

We are able to prove this by comparing runs of this benchmark with max = ~10 vs max = ~10000.

For max = 10,
```
sim_num_mispred_2bitsat        207
sim_br_2bitsat_ratio        0.1949
sim_num_mispred_2level        225
sim_br_2level_ratio        0.2119
```

For max = 10000,
```
sim_num_mispred_2bitsat        7536
sim_br_2bitsat_ratio        0.1474
sim_num_mispred_2level        7554
sim_br_2level_ratio        0.1477
```

Showing all the patterns would take on forms such as TTTN…TTTN…..TTNT…etc, but showing all would be tedious. To demonstrate this in a table with the current size of our tables would be very extensive.

Anthony Alayo 997487401
Freddy Chen 997363124

Collected Misprediction Statistics

Below are the collected statistics for each configuration.

Static Predict-Taken Predictor		
	Num Mispredicted	Misprediction Ratio
gcc.eio	21702643	0.5046
compress.eio	4898201	0.5455
go.eio	33840033	0.5449
mb.c (N=100000)	347481	0.6933

2-Bit Saturating Counter Predictor		
	Num Mispredicted	Misprediction Ratio
gcc.eio	5707779	0.1327
compress.eio	1447537	0.1612
go.eio	15529165	0.2501
mb.c (N=100000)	73536	0.1467

Two-Level Predictor using 2-bit sat., Pap Configuration		
	Num Mispredicted	Misprediction Ratio
gcc.eio	5839308	0.1358
compress.eio	1446624	0.1611
go.eio	13373801	0.2154
mb.c (N=100000)	73557	0.1468

Open Ended - Tournament Predictor		
	Num Mispredicted	Misprediction Ratio
gcc.eio	3637268	0.0846
compress.eio	907007	0.1010
go.eio	12420222	0.2000
mb.c (N=100000)	60246	0.1202

Anthony Alayo 997487401
Freddy Chen 997363124

Open-Ended Branch Predictor Implementation

For our open ended branch predictor, we decided to use a Tournament Predictor, which is a hybrid between a global predictor and a private 2 level predictor. The Tournament Predictor chooses which type of predictor to use based on a 2 bit saturating counter indexed by 14 bits in the PC. This chooser uses a total of 2^14 * 2 = 32768 bits.

The global predictor consists of a 8 bit history register and a global prediction table using 2 bit saturating counters. We index the global prediction table by combining 8 bits in the history register and 6 bits in the PC, for a total of 14 bits. This technique is known as gselect, and is known to yield better predictions. The global predictor uses a total of 8 + 2^14 * 2 = 32776 bits.

The private 2 level predictor uses 13 bits from the PC to index a private history table. The private history is used to index private prediction tables using 2 bit saturating counters. There are 8 private prediction tables to choose from, and we use 3 bits from the PC to index these. The bits used to index the private history table do not overlap with the bits used to choose the private prediction tables. The private 2 level predictor uses a total of 2^13 * 6 + 8 * 2^6 * 2 = 50176 bits.

In total, our open ended branch predictor uses 32768 + 32776 + 50176 = 115720 bits, or 14465 bytes.

CACTI - Area, Access Latency & Leakage Power

We created cacti configuration files for each unique table that existed in our scheme. This implies that for the two-level predictor, we had 2 unique tables. For the open-ended predictor, we had 3 unique tables. Our global history register was not able to be represented in CACTI as it is only a single register.

CACTI Configuration - Two-level Predictor using 2-bit sat. Pap				
2level-bpred-1.cfg				
Type: ram	Size: 512	Blk Size: 1	I/o bus width: 8	tag size: N/A
2level-bpred-2.cfg				
Type: cache	Size: 64	Blk Size: 1	I/o bus width: 8	tag size: 2

CACTI Configuration - Two-level Predictor using 2-bit sat. Pap				
open-ended-bpred-1.cfg				
Type: cache	Size: 16384	Blk Size: 1	I/o bus width: 8	tag size: 2
open-ended-bpred-2.cfg				
Type: cache	Size: 64	Blk Size: 1	I/o bus width: 8	tag size: 2
open-ended-bpred-3.cfg				
Type: ram	Size: 8192	Blk Size: 1	I/o bus width: 8	tag size: N/A

Cacti results…

Anthony Alayo 997487401
Freddy Chen 997363124

CACTI Results - Two Level Pap Predictor		
	2level-bpred-1	2level-bpred-2
Area (h x w) [mm]	0.0391054 x 0.0269215	0.0135995 x 0.00768072
Access Latency [ns]	0.163585	0.114907
Dynamic Read [nJ]	0.000420231	0.000102761
Total Leakage [mW]	0.195006	0.0205643

CACTI Results - Open Ended			
	open-ended-bpred-1	open-ended-bpred-2	open-ended-bpred-3
Area (h x w) [mm]	0.131933 x 0.0953235	0.0135995 x 0.00768072	0.133416 x 0.0953244
Access Latency [ns]	0.280927	0.114907	0.279886
Dynamic Read [nJ]	0.00211634	0.000102761	0.0022076
Total Leakage [mW]	2.87744	0.0205643	2.87418

Work Completed Summary

Freddy Chen – 2 bit saturating predictor, 2 level predictor, open predictor

Anthony Alayo – static predictor, micro-benchmark, cacti results, report compilation