

In this design assignment, you will implement and evaluate a number of direction predictors for conditional branches and target predictors for indirect control transfer using a portion of the PIN tool that you developed in the first homework. This assignment is not meant to observe the performance advantage that can come from a good direction predictor or target predictor. Instead, you will only evaluate the predictor accuracy and not use any of the predictions. Please feel free to use parts of first homework solution.

#### PART A: DIRECTION PREDICTORS FOR CONDITIONAL BRANCHES [70 POINTS]

-----

Implement the following conditional branch predictors and count the number of mispredictions.

- A. Static forward not-taken and backward taken (FNBT) [5 points]
- B. Bimodal [5 points]
- C. SAg [10 points]
- D. GAg [10 points]
- E. gshare [10 points]
- F. Hybrid of SAg and GAg [12 points]
- G. Hybrid of SAg, GAg, and gshare [ 7 points for majority vote meta-predictor implementation, 11 points for tournament meta-predictor implementation]

The configurations of the predictors are described below.

Bimodal predictor: 512x2-bit PHT

SAg: 1024x9-bit BHT, 512x2-bit PHT

GAg: 9-bit global history, 512x3-bit PHT

gshare: 9-bit global history, 512x3-bit PHT

Hybrid of SAg and GAg: Same configuration of SAg and GAg as above. The tournament meta-predictor is a table of 512 2-bit saturating counters indexed by 9-bit global history.

Hybrid of SAg, GAg, and gshare: Same configuration of SAg, GAg, and gshare as above. You will experiment with two meta-predictors. In the first experiment, the meta-predictor is simply a majority voter. In the second experiment, the meta-predictor has three tables of 2-bit saturating counters. Each of the tables has 512 entries and is indexed by 9-bit global history. The first of these tables keeps track of the relative number of correct predictions offered by SAg and GAg. The second of these tables keeps track of the relative number of correct predictions offered by GAg and gshare. The third table keeps track of the relative number of correct predictions offered by gshare and SAg. Use these three indications to pick the best prediction using the following algorithm. First, the winner among SAg and GAg is picked using the same tournament technique used in the hybrid of SAg and GAg. Let's refer to it as W. Next, the winner among W and gshare is identified as the provider of the final prediction. I will refer to this meta-predictor as the tournament meta-predictor.

Initialize all tables to zero. Do not shift out the least significant two bits of the PC before indexing because x86 instructions are not necessarily word-aligned. All index functions are simple modulo hashes. Do not update any tables speculatively i.e., always update table entries with the correct outcomes.

Following benchmark applications from the SPEC 2006 suite should be used for evaluation.

1. 400.perlbench
2. 401.bzip2
3. 403.gcc
4. 429.mcf
5. 450.soplex
6. 456.hmmer
7. 471.omnetpp
8. 483.xalancbmk

## PART B: TARGET PREDICTORS FOR INDIRECT CONTROL FLOW INSTRUCTIONS [30 POINTS]

The branch target buffer (BTB) can offer nearly 100% accurate target prediction for direct calls and unconditional jumps because the targets of these control transfer instructions do not change. This may not be true for indirect control transfer instructions. Particularly, the target of an indirect control transfer may have high correlation with the control flow path from which the branch is made. We will evaluate the following two target predictors for indirect control transfer instructions in terms of the number of mispredictions.

## 1. Branch target buffer (BTB) indexed with PC [10 points]

This is a cache that remembers the target of the last execution of an indirect control transfer. The cache is indexed by a modulo hash function of the PC of the indirect control transfer instruction.

Again, do not shift out any bits of PC before indexing. The BTB should be configured to have 128 sets each with four ways (i.e., a total of 512 entries) and use the LRU replacement policy. Each entry of the BTB has a valid bit, a tag, LRU states, and a target. Remember that a BTB miss implicitly provides a target prediction pointing to the next instruction.

## 2. Branch target buffer (BTB) indexed with a hash of PC and global path history [20 points]

Maintain an  $n$ -bit global history register storing the outcomes of the last  $n$  conditional branches. When an indirect control transfer instruction is encountered, the XOR of the global history register and the least significant  $n$  bits of the PC (do not shift out any bits) of the indirect control transfer instruction is used to index into the BTB. Each entry of the BTB has a valid bit, a tag, LRU states, and a target. The tag is the complete PC of the indirect control transfer instruction.

For a BTB with  $k$  sets,  $n$  is  $\log(k)$  with base two. Evaluate this predictor for a BTB with 128 sets, four ways, and LRU replacement policy. The Intel Prescott processor (Pentium 4 with 90 nm transistors) used a similar algorithm for target prediction of indirect control transfer instructions. Originally, the idea came from the Pentium M (the Pentium processor for the mobile devices) design team, which greatly influenced the design of the Intel Core microarchitecture.

This part should be evaluated on the same set of SPEC 2006 applications as in PART A.

## WHAT TO SUBMIT

Use the exactly same benchmark application setup (i.e., number of instructions to fast-forward and number of instructions to be used for evaluation) as in the first homework, except that your work should be put in HW2 directory (as opposed to HW1 directory). Feel free to use portions of the PIN tool from the solutions to the first homework. Write your PIN tool so that you can simulate all the predictors and collect all the results in one execution of an instrumented binary. This would greatly reduce the time to complete this homework because you will fast-forward only once for each application.

Prepare two tables showing the misprediction fraction (count of wrong predictions over total number of predictions made) for PART A and PART B.

In the table of PART A, for each prediction technique and for each application, separately show the misprediction fraction of forward conditional branches, misprediction fraction of backward conditional branches, and the overall misprediction fraction.

In the table of PART B, in addition to the misprediction fraction, show the BTB miss rate for each of the applications.

Prepare a PDF document with these two tables and brief explanations for the results. Put the document in your HW2 directory. Archive your HW2 directory after removing all .o and .so files:

```
zip -r HW2_MYROLLNO.zip HW2/
```

Please note that we will not accept anything other than a .zip file. Replace

10/4/23, 1:00 AM

cse.iitk.ac.in/users/mainakc/2023Autumn/lec422/HW2.txt

MYROLLNO by your roll number. Send HW2\_MYROLLNO.zip to cs422autumn2023@gmail.com with subject line [CS422 HW2].