18-640 Foundations of Computer Architecture

Recitation 2:

Project 1: Branch Prediction

Naman Jain
September 10, 2014

Electrical & Computer ENGINEERING

Announcements

- Lab group signup
  - We hope everybody signed-up
  - The groups will be locked by midnight and submission directories will be made
- Must have ECE account by now
  - If you don't have one, email ECE IT support at help@ece.cmu.edu

Outline

- Setup
- Part 1: Analysis
- Part 2: Implementation
- Part 3: Challenge
- Report
- Hand-in Details
- Other resources

Outline

- Setup
- Part 1: Analysis
- Part 2: Implementation
- Part 3: Challenge
- Report
- Hand-in Details
- Other resources

Setup

- Copy `lab1.patch` in your gem5 folder available at

```
/afs/ece.cmu.edu/class/ece640/project/project1/lab1.patch
```

- Execute

```
patch -p1 < lab1.patch
```

- Make sure you see a list of file changes in `stdout`.

Outline

- Setup
- Part 1: Analysis
- Part 2: Implementation
- Part 3: Challenge
- Report
- Hand-in Details
- Other resources

Analysis

- You will be analyzing
  - Tournament Predictor
    Kessler, Richard E. "The alpha 21264 microprocessor." *Micro, IEEE* 19.2 (1999): 24-36.
  - Bi-mode Predictor
    Lee, Chih-Chieh, I-CK Chen, and Trevor N. Mudge. "The bi-mode branch predictor." *Microarchitecture, 1997. Proceedings., Thirtieth Annual IEEE/ACM International Symposium on*. IEEE, 1997
- View source code at `src/cpu/pred/*`
  - Specifically, `tournament.{hh, cc}` and `bi_mode.{hh, cc}`

What did the patch do?

- To make it easier for you to change predictor type and their configurations, this patch adds various command line options.

  `--pred-type=tournament` – branch predictor to be implemented
  `--local-pred-size=` local predictor size
  `--global-pred-size=` global predictor size
  `--choice-pred-size=` choice predictor size
  `--local-ctr=` number of local counter bits
  `--global-ctr=` number of global counter bits
  `--choice-ctr=` number of choice counter bits

Example

We will be using only SE mode for this project:

```
build/X86/gem5.opt
   --stats-file=jpeg-encode.stat
   --dump-config=jpeg-encode.config

   configs/example/se.py
         -c mibench/consumer/jpeg/jpeg-6a/cjpeg
         -o "-dct int -progressive -opt -outfile mibench/consumer/jpeg/
output_small_encode.jpeg mibench/consumer/jpeg/input_small.ppm”
         --cpu-type=detailed
         --caches

         --pred-type=tournament
         --local-pred-size=2048
         --global-pred-size=8192…
```

Output

• Understand the `m5out/*.stat` file generated; following portion of the file is specifically generated for branch predictors:

```
system.cpu.branchPred.lookups                 18211747
system.cpu.branchPred.condPredicted           17199049
system.cpu.branchPred.condIncorrect            6188168
system.cpu.branchPred.BTBLookups              10359986
system.cpu.branchPred.BTBHits                  8157880
system.cpu.branchPred.BTBCorrect                     0
system.cpu.branchPred.BTBHitPct              78.744122
system.cpu.branchPred.usedRAS                   174617
system.cpu.branchPred.RASInCorrect                   0
```

Output

- Calculating mispredictions:

```
100 –
[(system.cpu.branchPred.condIncorrect/
system.cpu.branchPred.condPredicted)*100]
```

- Calculate IPC:

```
system.cpu.ipc
```

- Script

  `parse_branch_data.py` can be used to find important data, it looks for all `.stat` files in `m5out` folder.

Your Task

- **Analyze** predictors and **report** branch mispredictions and IPC for the following configurations only for *jpeg-encode* benchmark:

  Take number of counter bits as 2 for all data structures.

- ***Tournament Predictor:***

	Config – 1	Config – 2	Config – 3	Config – 4
Local Predictor Size	2048	4096	4096	4096
Global Predictor Size	8192	4096	8192	8192
Choice Predictor Size	8192	8192	4096	8192

- ***Bi-mode Predictor:***

	Config – 1	Config – 2	Config – 3	Config – 4
Global Predictor Size	2048	4096	8192	8192
Choice Predictor Size	4096	8192	4096	8192

Outline

- Setup
- Part 1: Analysis
- Part 2: Implementation
- Part 3: Challenge
- Report
- Hand-in Details
- Other resources

Implementation

- You will be implementing
  - gshare

    Scott McFarling, "Combining Branch Predictors" - Technical Note, http://www.hpl.hp.com/techreports/Compaq-DEC/WRL-TN-36.pdf
  - YAGS

    Eden, Avinoam N., and Trevor Mudge. "The YAGS branch prediction scheme."*Proceedings of the 31st annual ACM/IEEE international symposium on Microarchitecture*. IEEE Computer Society Press, 1998.

- We have added dummy files such as {`gshare.cc, gshare.hh`} and {`yags.cc, yags.hh`} for you. You may take help from the existing predictors implemented in the distribution.

Your Task

- **Analyze** predictors and **report** branch mispredictions and IPC for the following configurations only for *jpeg-encode* benchmark:

  Take number of counter bits as 2 for all data structures.

- ***gshare Predictor:***

	Config – 1	Config – 2	Config – 3	Config – 4
Local Predictor Size	2048	4096	8192	16384

- ***YAGS Predictor:***

	Config – 1	Config – 2	Config – 3	Config – 4
Global Predictor Size	2048	4096	8192	8192
Choice Predictor Size	4096	8192	4096	8192

Outline

- Setup
- Part 1: Analysis
- Part 2: Implementation
- Part 3: Challenge
- Report
- Hand-in Details
- Other resources

Challenge

- Improve your misprediction rate (gshare and YAGS) keeping the configuration:

Gshare Predictor:

	Config
Local Predictor Size	4096

YAGS Predictor:

	Config
Global Predictor Size	4096
Choice Predictor Size	8192

Higher the better!

Outline

- Setup
- Part 1: Analysis
- Part 2: Implementation
- Part 3: Challenge
- Report
- Hand-in Details
- Other resources

Report

- The report should present your branch predictor design and document the results of all branch predictors you either implemented or analyzed.
- Pay special attention to highlight the changes you tried and how effective they were.
- The report must also address the questions asked in handout.

Grading

- Part 1: Analysis (15)
- Part 2: Implementation - gshare & YAGS (25 + 25)
- Part 3: Challenge (15)
  - You will be graded based on your *Branch Misprediction* (average of gshare and YAGS) relative to the rest of the class.
  - It will be tested on a subset of MiBench benchmarks.
  - Specifically, (exception if within 0.2% of highest)
    - If you are in the best quartile, you will receive the full 15 of 15 points.
    - If you are in the second best quartile, you will receive 10 of 15 points.
    - If you are in the third best quartile, you will receive 5 of 15 points.
    - If you are in the lowest quartile, you will receive 0 of 15 points.
- Report (15)
- Feedback (5)

Outline

- Setup
- Part 1: Analysis
- Part 2: Implementation
- Part 3: Challenge
- Report
- Hand-in Details
- Other resources

Hand-in Details

- You will need to hand-in `src/cpu/pred/gshare.*` and `src/cpu/pred/yags.*`

to

`/afs/ece/class/ece640/submission/<andrew_id>/<groupid>/project1/`

- Your group directories will be made and linked from your `submission/<andrew_id>/`

Outline

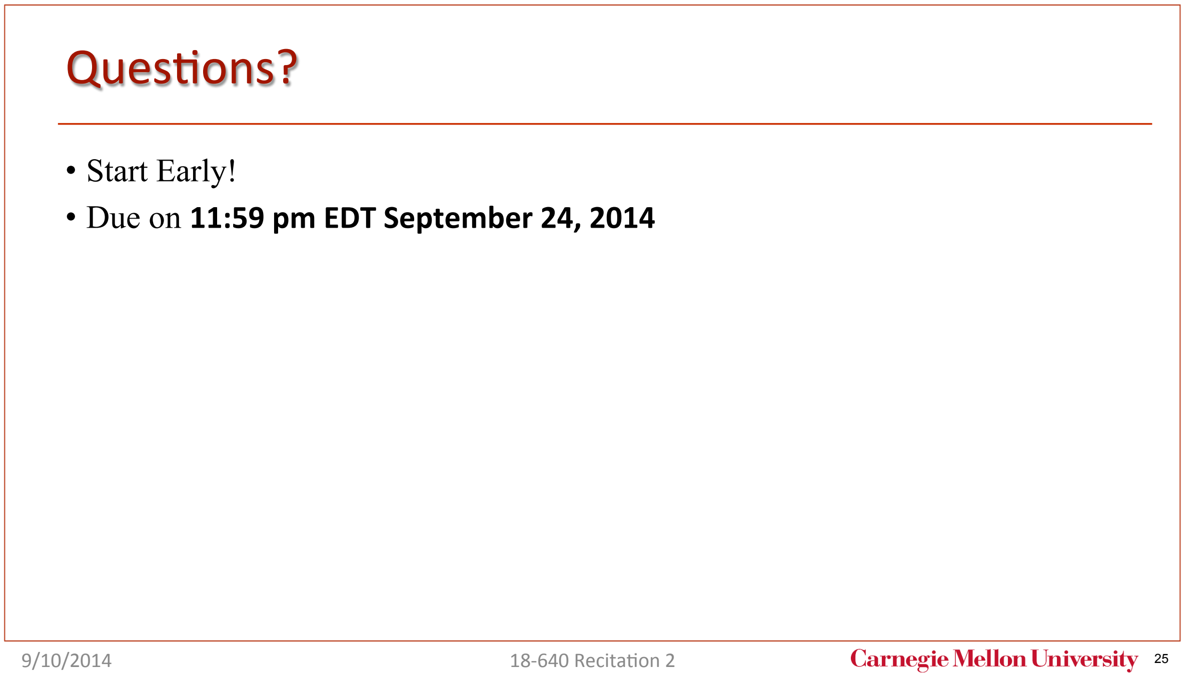
- Setup
- Part 1: Analysis
- Part 2: Implementation
- Part 3: Challenge
- Report
- Hand-in Details
- Other resources

Other Resources

- Appendix at the end of handout is available to get you started.
- Append your branch predictor configurations to the commands mentioned in the Appendix.

```
#consumer/jpeg-encode
build/X86/gem5.opt --stats-file=jpeg-encode.stat --dump-config=jpeg-encode.config
configs/example/se.py -c mibench/consumer/jpeg/jpeg-6a/cjpeg -o "-dct int -progressive
-opt -outfile mibench/consumer/jpeg/output_small_encode.jpeg mibench/consumer/jpeg/
input_small.ppm" --cpu-type=detailed --caches

#consumer/jpeg-decode
build/X86/gem5.opt --stats-file=jpeg_decode.stat --dump-config=jpeg_decode.config
configs/example/se.py -c mibench/consumer/jpeg/jpeg-6a/djpeg -o "-dct int -ppm -outfile
mibench/consumer/jpeg/output_small_decode.ppm mibench/consumer/jpeg/input_small.jpg" --
cpu-type=detailed --caches …
```



Questions?

- Start Early!
- Due on **11:59 pm EDT September 24, 2014**