Computer Architecture – Branch Predictor, Yu Liu, Nov 19th, 2019.

- These files in the package are for problem 1 to problem4.
- The source codes are under c language, including head are <stdio.h> , “stdbool.h” and <math.h>, can run under linux , unix, window 7 or above.
- Unzip the file of HW4_YULIU, put these files in one folder:
  - ✓ branchprd_yu.c
  - ✓ stdbool.h
  - ✓ branch-trace-gcc.trace (by extracting data.zip)
  - ✓ branches_0_dhrystone.out (by extracting data.zip)
  - ✓ branches_0_linpack.out (by extracting data.zip)
  - ✓ README.pdf

  In the package files, below files are generated in the experiments for reference only:
  - ✓ generalInfo_0_dhrystone.out (this file is generated by pin tool)
  - ✓ generalInfo_0_linpack.out (this file is generated by pin tool)
  - ✓ branch_prediction.txt (this file is generated by the branchprd_yu.out)

- Below processes all run under linux as examples
- Any questions call my phone: +1 704 858 7806

This program will automatically detect the data type by checking the file postfix, the fetch the data according to the format and output the result by generating the file named branch_prediction.txt.

For all problems just follow the same steps as below:

1. Use file of “branchprd_yu.c”
2. Select the parameters and data file by revising the red word in “define” area:

```
#define FILENAME_INPUT    "branches_0_dhrystone.out"
#define FILENAME_OUTPUT "branch_prediction.txt"
#define PREDICTOR_TYPE    'S' //upper case.O:one level, G:Global, S: Gshare,    L: Two level local, H: Hybrid
#define PHT_SIZE 1024 //size of Local history register entry.Optionals: 256 (8 bits),128 (7 bits),64 (6 bits), 16(4 bits), 4(2 bits)
#define LHR_SIZE 128 //size of Local history register entry.Optionals: 1024(10 bits), 256 (8 bits),128 (7 bits),64 (6 bits), 16(4 bits), 4(2 bits)
#define GHR_BITS 10 //width of bit counter: optionals are 10,8,6,4,2
#define PC_SHIFT_BITS 5 // the last x bits of program counter to be ignored. options are 0,2,3,5
```

For example, we want to run branch-trace-gcc.trace under Two-Level Local predictor with 10 bits LHR and PC shit bits 3, then we can seting as below figure:

```
22 //                                                                      *
23 //Output file:branch_prediction.txt                                    *
24 //**********************************************************************
25 #include <stdio.h>
26 #include <math.h>
27 #include <string.h>
28 #include "stdbool.h"
29
30 #define FILENAME_INPUT  "branches_0_dhrystone.out" // if run dhrystone data, cha
31 #define FILENAME_OUTPUT "branch_prediction.txt" // if run dhrystone data, cHange
32
33 //constant declarations
34 #define PREDICTOR_TYPE  'L' //upper case.O:one level, G:Global, S: Gshare,  L: T
35 #define PHT_SIZE 1024
36 #define LHR_SIZE 1024 //size of Local history register entry.Optionals: 1024 (10
37 #define GHR_BITS 10 //width of bit counter: optionals are 10,8,6,4,2
38 #define PC_SHIFT_BITS 3// the last x bits of program counter to be ignored. opti
39
40
41 int PHT[PHT_SIZE]; //pattern history table
42 int choiceTable[PHT_SIZE]; //for tournament predictor
43 int LHR[LHR_SIZE]; //local history register
44 int hitTaken=0;
45 int missTaken=0;
46 int hitNotaken=0;
47 int missNotaken=0;
48 int branchNum=0;
49
50
51 void mainPredictor();
```

3. Compile file of branchprd_yu.c, use gcc command, add –lm for including <math.h>:

   $ gcc branchprd_yu.c –o branchprd_yu.out –lm

4. Run

   $ ./ branchprd_yu.out

5. Output data

   A file named “r branch_prediction.txt” will be generated automatically (samples already listed). Open and read the data inside, sample as below figure.

File Edit Format View Help

```
Input Data File: branches_0_dhrystone.out
Output Data File: branch_prediction.txt

Preditor type(G:Global, L: local, S: Gshare, O:one level): H
Pattern History Table size: 1024
Local History Register Size: 128
Global History Register Bits: 10
PC Shift Bits: 5

Branches            Total         Taken           Not Take
Misses               2365          1719               646
Hits               378189        371256              6933
Total              380554        372975              7579
Hit Rate           0.9938        0.9954            0.9148
Miss Rate          0.0062        0.0046            0.0852
```

6. Done