# Computer Architecture
# Project 2

## Understanding gem5 branch predictor structure

Source:
http://www.gem5.org/docs/html/classBPredUnit.html#af22b5b56f3d739a9c4d1f7a0ad415ced

You can find the different functions for branch predictor from the above website. There are five main function called in each branch predictor namely lookup(), update(), uncondBranch(), btbUpdate(), squash().

1. lookup(): This function looks up a given PC in the BP to see if it is taken or not taken. It parameters passed into the function are inst_PC (the PC to look up) and bp_history (pointer that will be set to an object predictor state associated with the lookup) and the methods returns whether the branch is taken or not.
2. update(): This function updates the BP with taken/not taken information. The input parameters are:
    i. inst_PC: The branch's PC that will be updated.
    ii. taken: whether the branch is taken or not
    iii. bp_history: pointer to the branch predictpr state that is associated with the branch lookup that is being updated
    iv. squashed: Set to true when this function is called during squash operation.
3. uncondBranch(): This method is called when the branch instruction is an unconditional branch. Branch predictor will not do anything but update the global history with taken.
4. btbupdate(): This function is called only when there is a branch miss in BTB. This can happen when the branch prediction is accurate but it does not know where to jump. In this case you will predict the branch as not taken so that BTA is not required.
5. squash(): This function squashes all outstanding updates until a given sequence number. Typically, the input parameter is a bp_history (Pointer to the history object. The predictor will need to update any state and delete the object).

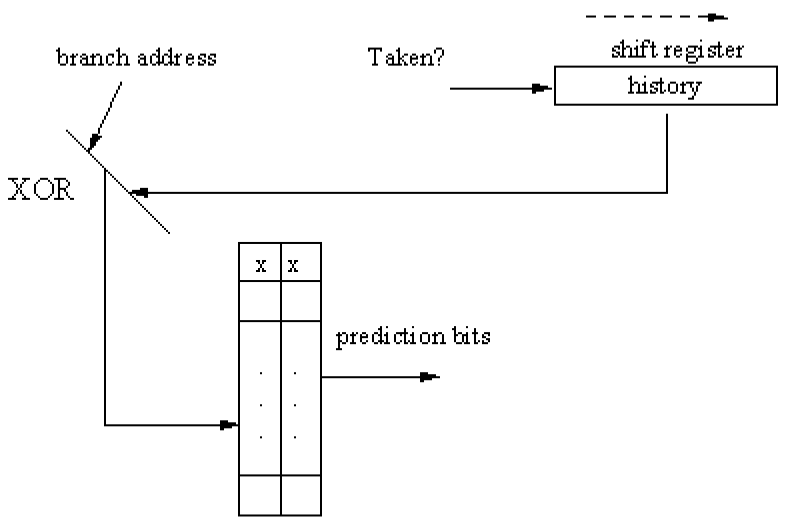
## Understanding 2-bit local and Tournament BP

2-bit local branch predictor is a simple branch predictor which has 4 states, namely Strongly taken, weakly taken, weakly not taken, strongly taken denoted as (11, 10, 01, 00). You can find the implementation is the following directory -- <path to gem5-cse-ca>/src/cpu/pred/ 2bit_local.cc, 2-bit_local.hh, tournament.cc and tournament.hh.

You can find a good explanation of 2-bit local BP in https://courses.cs.washington.edu/courses/csep548/06au/lectures/branchPred.pdf and tournament BP in http://courses.cs.tamu.edu/ejkim/614/tournament_predictors.pdf.

**Understanding Gshare branch predictor**

You can find a good explanation of gshare branch predictor in http://web.engr.oregonstate.edu/~benl/Projects/branch_pred/ under "Gshare Scheme" second diagram.



"Gshare scheme is similar to bimodal predictor or branch history table. However, like correlation-based prediction, this method records history of branches into the shift register. Then, it uses the address of the branch instruction and branch history (shift register) to XOR's them together. The result is used to index the table of prediction bits. Implementation can use either tagged or tagless tables. In the tagged table, we compare the indexed entry with a tag and, if they don't match, we just predict as not taken. If they match, we use the corresponding prediction bits to predict the branch outcome. In the tagless implementation, the table is just directly indexed and the corresponding prediction bits are used to predict the branch outcome. We have tried both these methods, and found that tagless table performs better. We just index an entry and use its contents (prediction bits) to predict the branch outcome. "

**Problem 1 [7 points]** – Implementing Gshare BP

You have to create two files namely gshare.cc and gshare.hh in the – <path to gem5>/src/cpu/pred/ directory. Have the five main functions namely lookup(), update(), uncondBranch(), btbUpdate(), squash() methods and other functions that you seem fit to implement the gshare BP.

**Change 1: Change to BranchPredictor.py**

Name your class as `GshareBP()` and add the following lines and classes in your **`gem5-cse-ca/src/cpu/pred/BranchPredictor.py`**

```
class GshareBP(BranchPredictor):

    type = ‘GshareBP’

    cxx_class = ‘GshareBP’

    cxx_header  = “cpu/pred/gshare.hh”

    PHTCtrBits  = Param.Unsigned('2', "Size of counter bits")

    PHTPredictorSize = Param.Unsigned('16384', "Size of local Predictor")

    globalPredictorSize = Param.Unsigned('16', "Size of global Predictor")
```

```
class Gshare8KBP(GshareBP):

   PHTPredictorSize = Param.Unsigned(4096, "Size of local predictor")

   PHTCtrBits = Param.Unsigned(2, "Bits per counter")

   globalPredictorSize = Param.Unsigned(12, "Size of global history bits")
```

```
class Gshare32KBP(GshareBP):

    PHTPredictorSize = Param.Unsigned(16384, "Size of local predictor")

    PHTCtrBits = Param.Unsigned(2, "Bits per counter")

    globalPredictorSize = Param.Unsigned(16, "Size of global history bits")
```

**Change 2: Change to SConscript**

Add the following line at the end of src/cpu/pred/SConscript

```
Source(‘gshare.cc’)
```

Note: After all the above modifications recompile gem5 by running the following command:

```
$> scons build/ALPHA/gem5.opt
```

**Problem 2 [3 points]** – Running your implementation

The gem5 provided has an in-build commandline --bp_type flag to call the branch predictor. For this project you will run FFT and qsort for 2 configurations of GshareBP given in the table.

| Config | PHT enteries | PHT bits | Global History Register | Benchmarks |
|---|---|---|---|---|
| 1 | 4096 | 2 | 12 | • FFT<br>• qsort |
| 2 | 16384 | 2 | 14 | • FFT<br>• qsort |

The gem5 commandline options you will be using for the assignment is the following

```
build/ALPHA/gem5.opt --outdir=FFT-Conf1 configs/example/se.py --cpu-type=DerivO3CPU --caches --l2cache --bp-type=Gshare32KBP -c  <path to fft binary> -o "4 4096"
```

**Project Deliverables**

1. For this project, you will create a patch file, which is portable and easy to apply your gem5 modifications and remove them by the graders. You will have to follow the following instructions to create a correct patch.
    a. If you have followed the getting started with gem5 document then you should have two copies of gem5. One with name gem5 and one with gem5-cse-ca. You should have modified the gem5-cse-ca directory.
    b. To create a correct patch run the following command;

```
diff -ruN gem5/src/cpu/pred/ <path_gem5-cse-ca>/src/cpu/pred/ > project2.patch
```

Note: Make sure the "N" in -ruN is uppercase. You can open project2.patch and locate all your modifications to ReplacementPolicies.py, SConscript, lru_ipv.hh and lru_ipv.cc with a "+" sign at the beginning of each line modified.

2. Like project 1, you will submit the config.ini, config.json and stats.txt along with the patch file created. The file directory should look like the following:

```
\---Project2-Submission
    +---Problem1
    |   +---project2.patch
    |
    \---Problem2
        +---FFT-Conf1
        |       config.ini
        |       config.json
        |       stats.txt
        |
        +---FFT-Conf2
        |       config.ini
        |       config.json
        |       stats.txt
        |
        +---qsort-Conf1
        |       config.ini
        |       config.json
        |       stats.txt
        |
        \---qsort-Conf2
                config.ini
                config.json
                stats.txt
```

## Submission Instructions

1) Submit single zip file containing all the files. Please follow the naming conventions correctly.

2) In case where you are doing project in a group, only one group member should make submission through blackboard. All the team members of the group will receive the same score.

## Rubric

1. **Problem 1 [7 points]**
   a. Correct patch file, grader should have no problem in applying the patch and running gem5 with your implementation [2 points]
   b. Correct implementation of GshareBP according to the explanation, with comments in the cc and hh files. [3 points]
   c. Correct class names, according to the Project deliverables [2 points]
2. **Problem 2 [3 points]**
   a. Correct configuration from the table from config.ini file. [1 points]
   b. Correct stats in the stats.txt file [2 points]