

HW#2 Branch Predictor Design



Chun-Jen Tsai
National Chiao Tung University
10/26/2020

Homework Goal

- ❑ The branch predictor removes the control hazard of a processor pipeline and improve its performance.
Modify the Branch Predictor of Aquila to see its impact on DMIPS/MHz
 - Change the parameters of the dynamic predictor in Aquila
 - Compare a static predictor to the dynamic predictor
 - Try other predictor ideas
- ❑ You must upload a report to E3 by 11/9, 17:00.

Types of Branches

- ❑ There are three types of branches
 - Conditional forward jumps – for if-then-else statements
 - Conditional backward – for looping
 - Unconditional jumps – for function calls, or from bad coding
- ❑ The problem of branches:

```
318:  li    a0, 48
31c:  jal   ra, 102c
320:  addi   s1, s1, -1
324:  bne   s1, s3, 318
328:  li     s1, -1
32c:  srli   a5, s0, 0x1f
330:  add    s3, s0, a5
```

The next PC should be 0x102c, but
the Fetch unit does not know that until
two clocks later (after the Execute stage)

The Fetch unit does not even know what the
next PC should be until two clocks later (after
the Execute stage)

Static Branch Prediction

- ❑ Static branch prediction always make the same decision (forward/backward × taken/not taken)
- ❑ Implementation can done by one of three methods
 - Hardwired into the processor pipeline
 - The predictor can do a quick decode of the target of the branch, and tells the program counter the next PC to fetch
 - Compilers generates the hint bit with ISA support
 - Cooperation between the processor and the compiler, by following some register usage convention. For example,
 - “bne s1, s3, 318” suggests taken
 - “bne s1, s4, 318” suggests not taken

Dynamic Branch Prediction

- ❑ The processor collects statistics of whether branches happen or not at every branch instructions
- ❑ The fetch unit fetches the predicted next instruction
- ❑ In the case of a misprediction, the pipeline has to be flushed to re-fetch the correct instruction
 - The penalty is high for a misprediction

Branch Prediction Schemes

❑ One-level Predictor

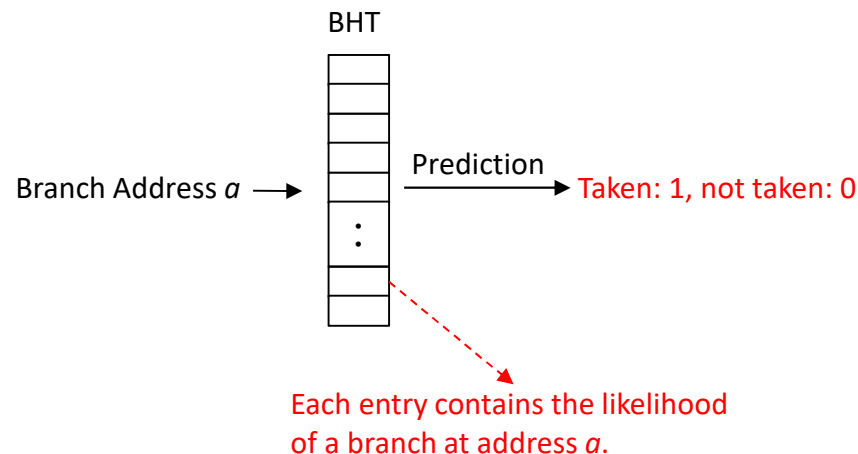
- Uses a Branch History Table (BHT) indexed by the recent branch addresses
- When the fetch unit reaches a branch location, it gets the PC for the next instruction to fetch based on the BHT

❑ Two-level Adaptive Branch Prediction

- MCFarling's Two-Level Prediction with index sharing (gshare, 1993).

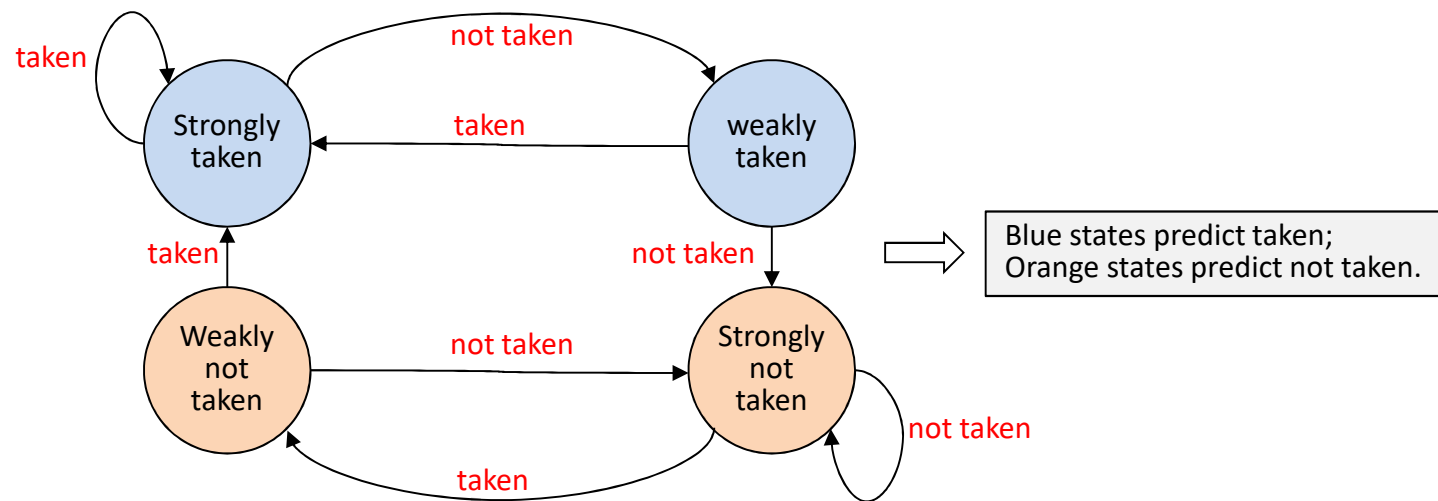
One-level Branch Predictor (1/2)

- ❑ For one-level branch predictor, we must determine:
 - How many address bits are used to index the BHT
 - How many bits are used to record the branch statistics
 - How many branch instructions are recorded in the BHT



One-level Branch Predictor (2/2)

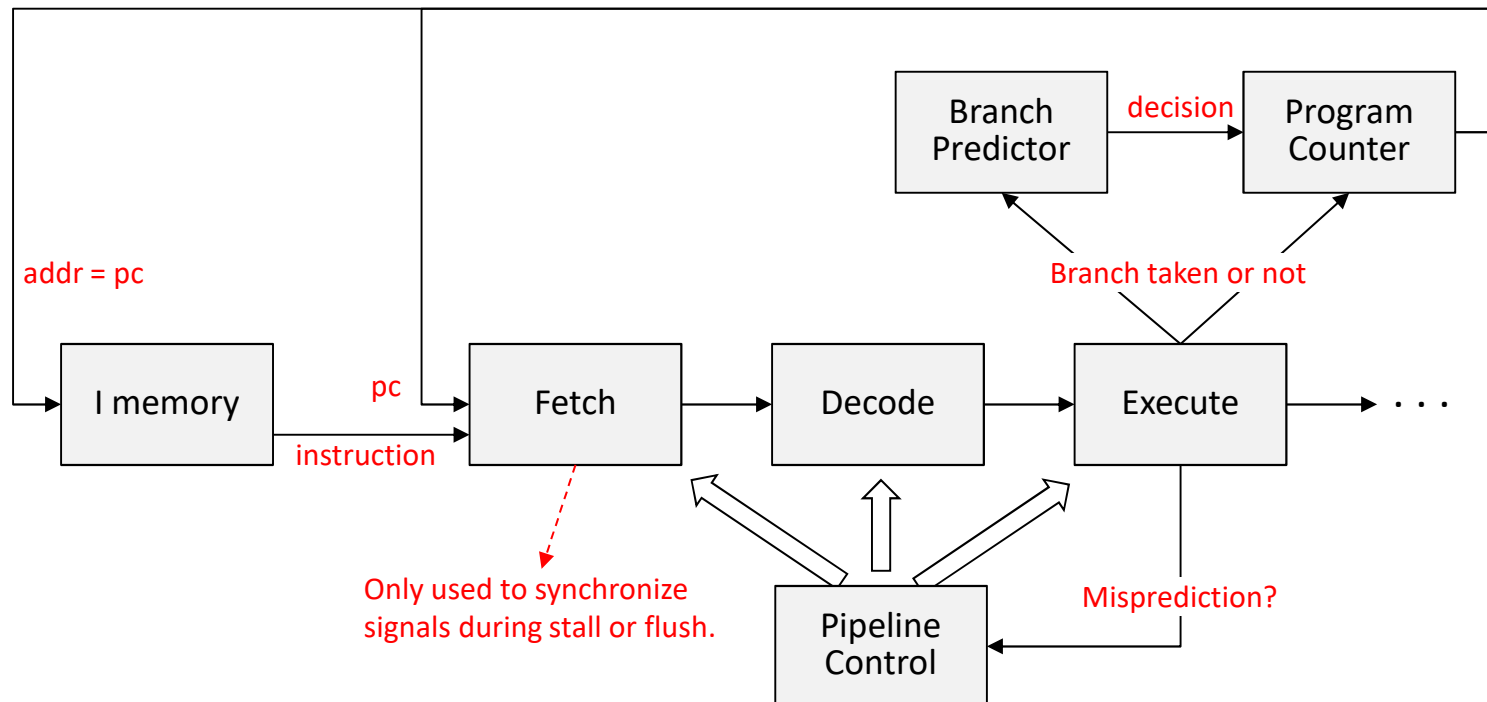
- ❑ Aquila implements the simple 2-bit predictor
 - For each branch instruction, we record its branch likelihood with one of four possible states:



- The state changes after the execute stage determines whether the branch is taken or not.

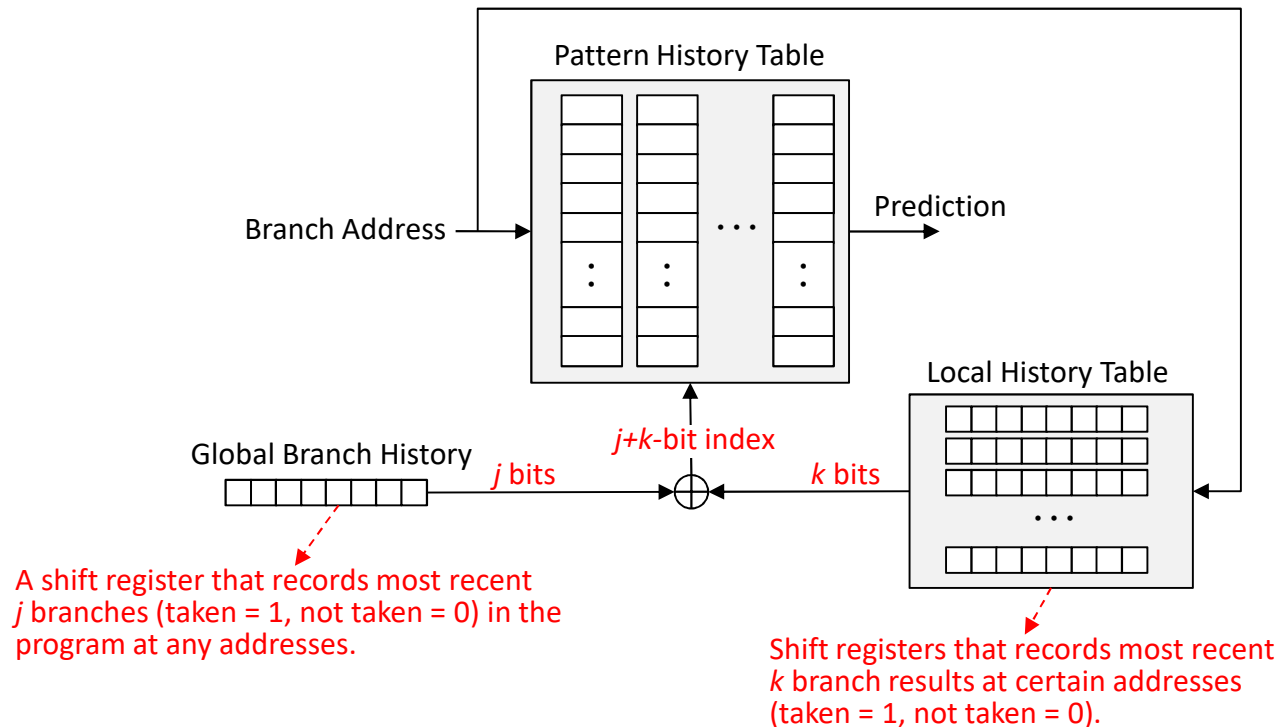
Branch Prediction Flow in Aquila

- ❑ A branch predictor tells the fetch unit which instruction to fetch before the branch has been executed



Two-level Branch Predictor (1/2)

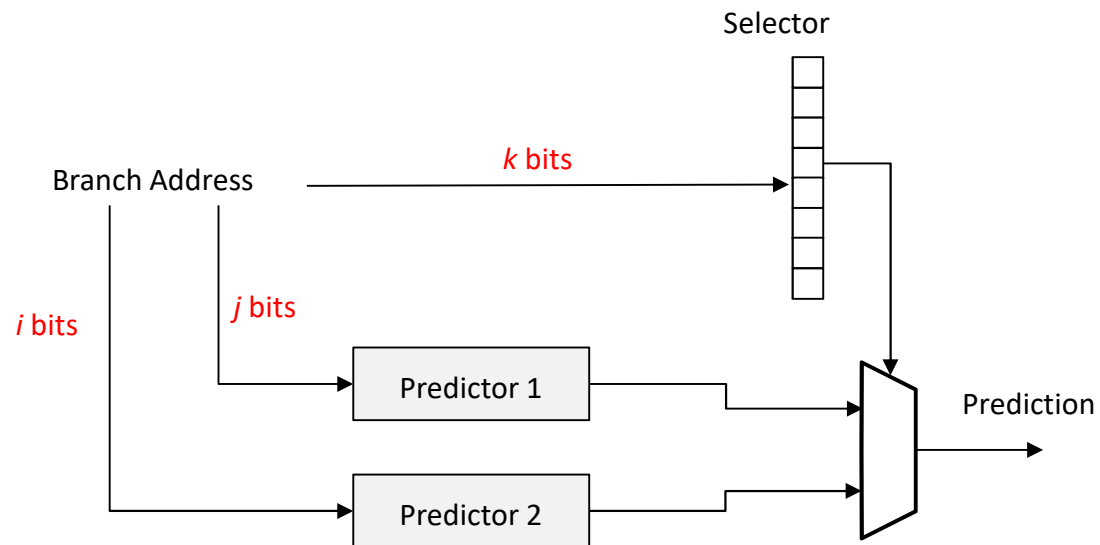
- ❑ A branch depends on the branch address as well as:
 - Nearby branches – recorded using Global Branch History
 - Longer history of the same branch – Local Branch History



T.-Y. Yeh and Y.N. Patt, "Two-level Training Branch Prediction," 24th ACM/IEEE Int. Symp. on Microarchitecture, Nov. 1991.

Two-level Branch Predictor (2/2)

- ❑ Different predictors work for different code patterns
 - Multiple predictors can be used to adapt to different code sections in the program:



Your Homework

- ❑ Study the branch predictor in Aquila
- ❑ Modify the branch predictor and see if the DMIPS is affected
 - Change the BHT parameters to see the performance
 - Try the static predictor (easy) or the two-level predictor (hard)
- ❑ Write a 4-page double-column report:
 - Your survey of branch predictors
 - Describe how BHT is implemented and how pipeline flush is done upon misprediction in Aquila
 - Analyze the branch statistics of Dhrystone and discuss whether current branch predictor can be further improved
 - Discuss your modifications to the branch predictor