Computer Organization and Architecture

Branch Prediction

Jeongseob Ahn

Department of Software & Computer Engineering
Ajou University

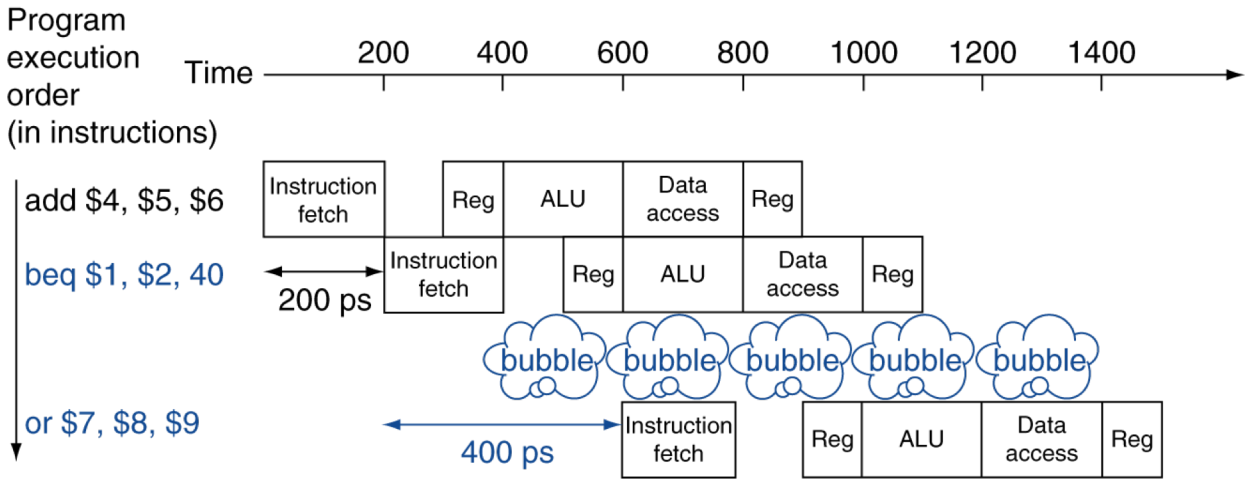
Branch Prediction

- Longer pipelines can't readily determine branch outcome early
  - Stall penalty becomes unacceptable
- Predict outcome of branch
  - Only stall if prediction is wrong
- In MIPS pipeline
  - Can *statically* predict branches **not taken**
  - Fetch instruction after branch, with no delay

Security Issue

MIPS with Predict Not Taken

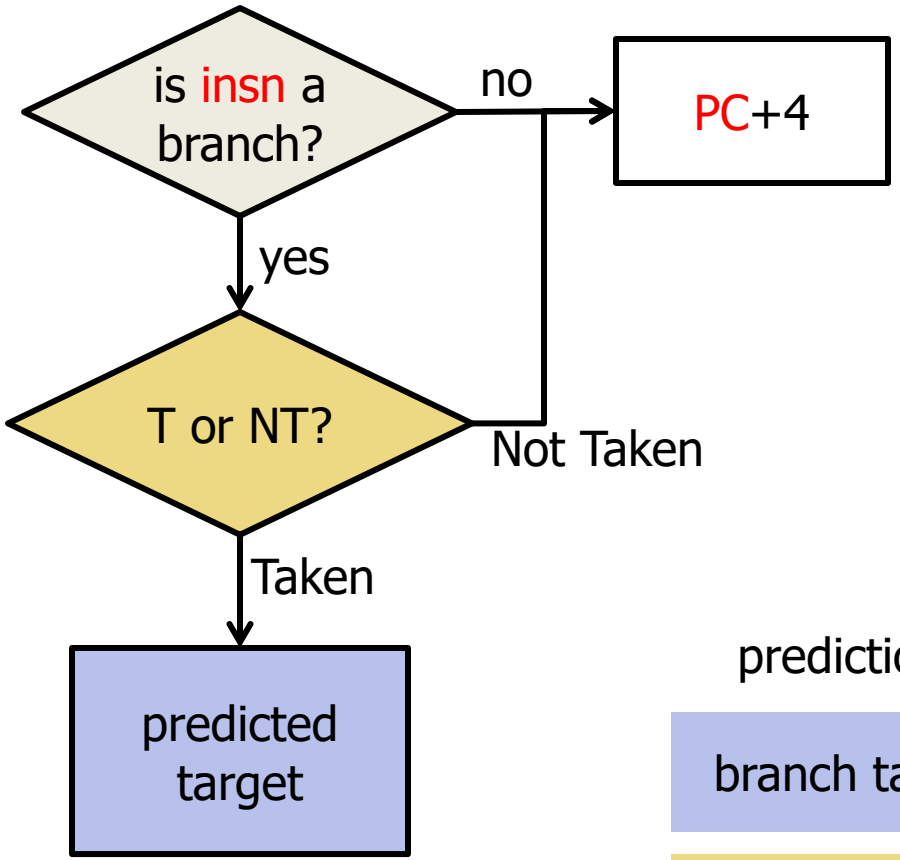
Prediction incorrect



Dynamic Branch Prediction

- Hardware measures actual branch behavior
  - Record recent history of each branch
- Assume future behavior will continue the trend
  - When wrong, stall while re-fetching, and update history

Branch Prediction Steps



- Which instructions behavior are we trying to predict?
- Where does PC come from?

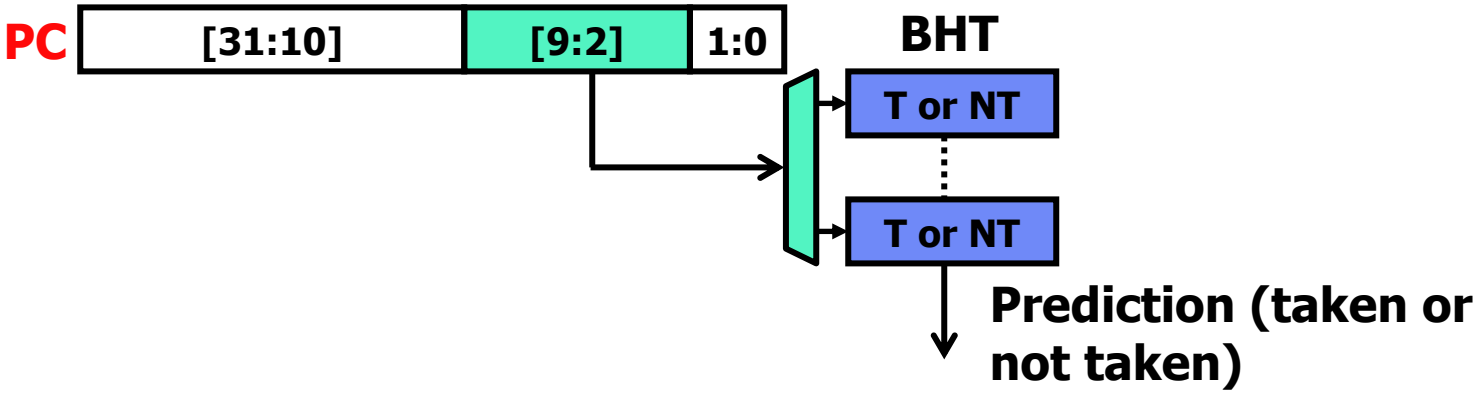
prediction source:

branch target buffer

direction predictor

Branch Direction Prediction

- **Direction predictor (DIRP)**
  - Map conditional-branch PC to taken/not-taken (T/N) decision
- **Branch history table (BHT)**: simplest predictor
  - PC indexes table of bits (0 = N, 1 = T), no tags
  - Essentially: branch will go same way it went last time

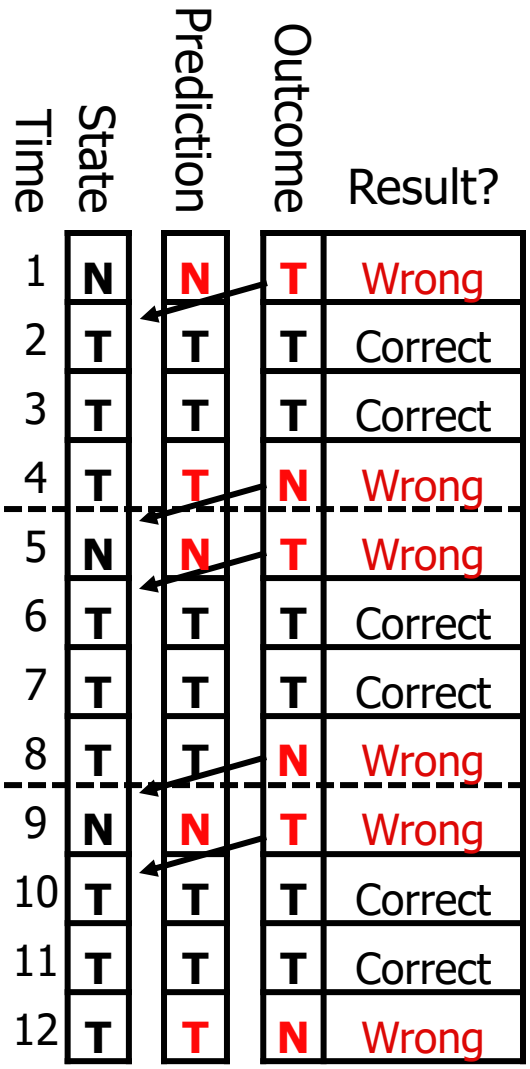


Branch History Table (BHT)

- **Branch history table (BHT)**: simplest direction predictor
  - PC indexes table of bits (0 = N, 1 = T), no tags
  - Essentially: branch will go same way it went last time
  - Problem: **inner loop branch** below

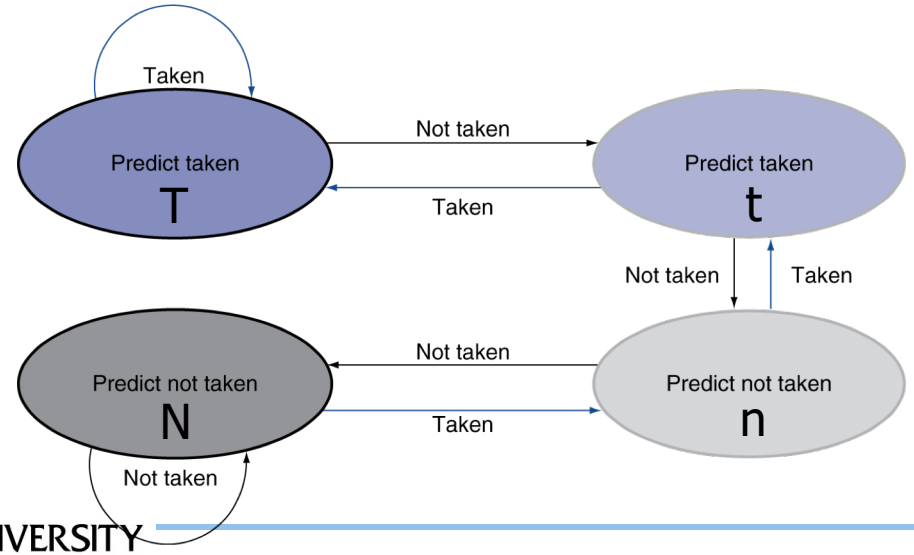
```
for (i=0;i<100;i++)
    for (j=0;j<3;j++)
        // loop body
```

    - Two "built-in" mis-predictions per inner loop iteration
    - Branch predictor "changes its mind too quickly"



Two-Bit Saturating Counters (2bc)

- **Two-bit saturating counters**
  - Replace each single-bit prediction
    - (0,1,2,3) = (N,n,t,T)
  - Adds "hysteresis"
    - Force predictor to mis-predict twice before "changing its mind"
  - One misprediction each loop execution (rather than two)



Time	State	Prediction	Outcome	Result?
1	N	N	T	Wrong
2	n	N	T	Wrong
3	t	T	T	Correct
4	T	T	N	Wrong
5	t	T	T	Correct
6	T	T	T	Correct
7	T	T	T	Correct
8	T	T	N	Wrong
9	t	T	T	Correct
10	T	T	T	Correct
11	T	T	T	Correct
12	T	T	N	Wrong

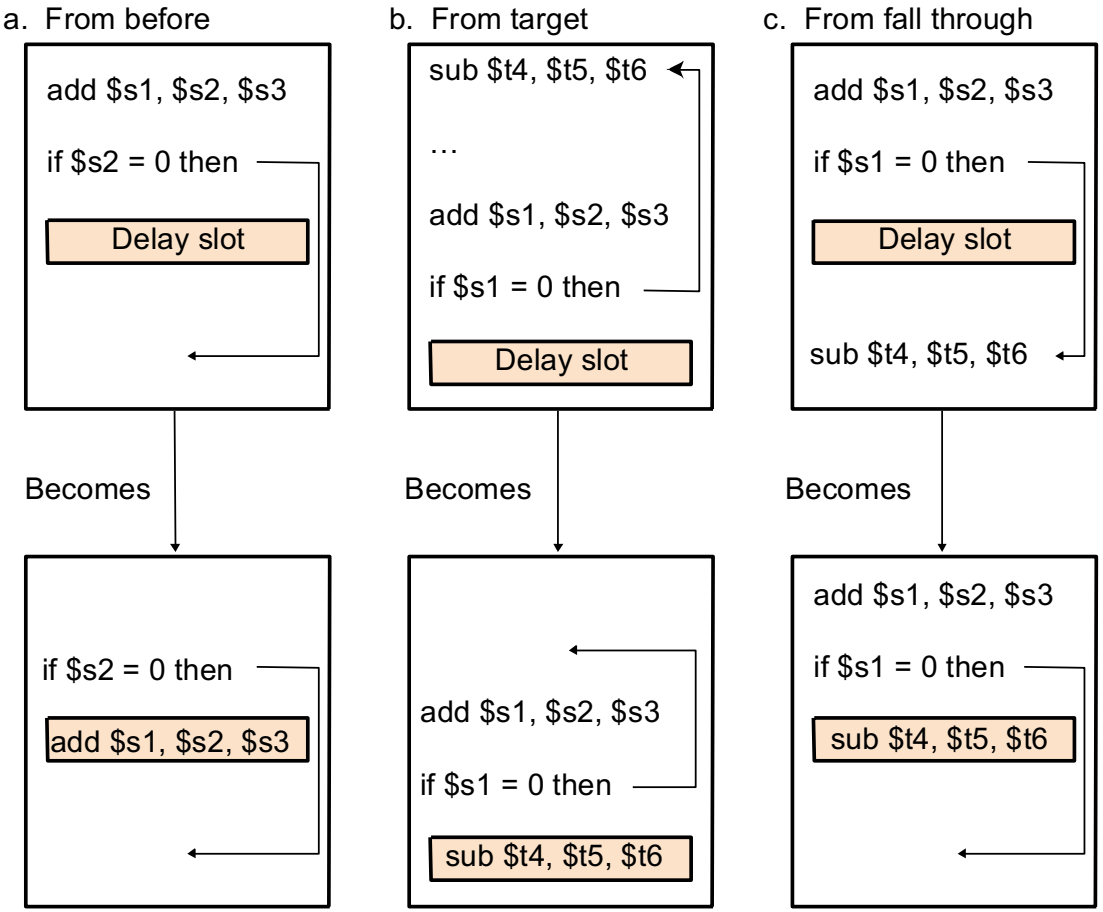
Branch Target Address Prediction

- **Branch target buffer (BTB)**:
  - A small hardware table
  - is-a-branch = (BTB[hash(PC)].**tag** == PC) ? 1 : 0
  - predicted-target = (BTB[hash(PC)].**tag** == PC) ? BTB[PC].**target** : 0

index	tag	target
0	0	0
1	0x4e3745	0x4738da
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0

Compiler Approach: Delayed Branching

- Reordering data independent instructions does not change program semantics



Precise Exception with Pipeline

- Traps (invoked by instruction)
  - Squash all subsequent instructions
  - Set PC to exception handling routine

- External interrupts
  - Processor states updated in MEM and WB
  - Must finish the instructions in the middle of MEM and WB

Advanced Topics

- Superscalar
  - Fetching two instructions at the same time (2-way Superscalar)
- Out-of-order Execution
  - Processors dynamically change the order of instruction to maximize ILPs (Instruction Level Parallelism)
- Hardware Multithreading (a.k.a Intel's Hyper Threading)
  - Sharing the hardware resources except for PC and Register File to maximize TLPs (Thread Level Parallelism)
  - Processor is responsible for scheduling the hardware threads
    - It is not software multithreading technique