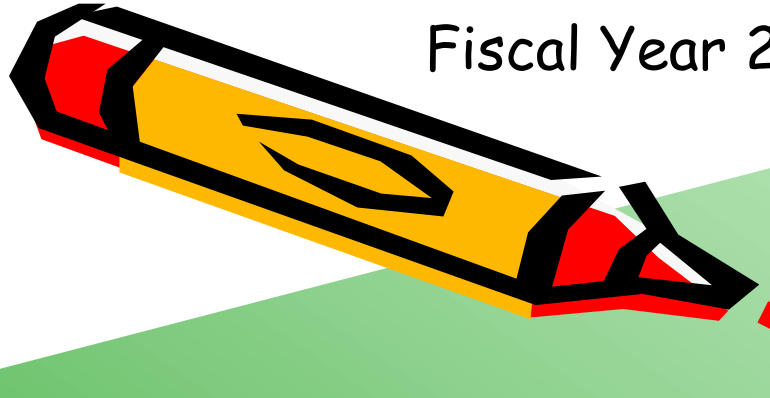
Fiscal Year 2022

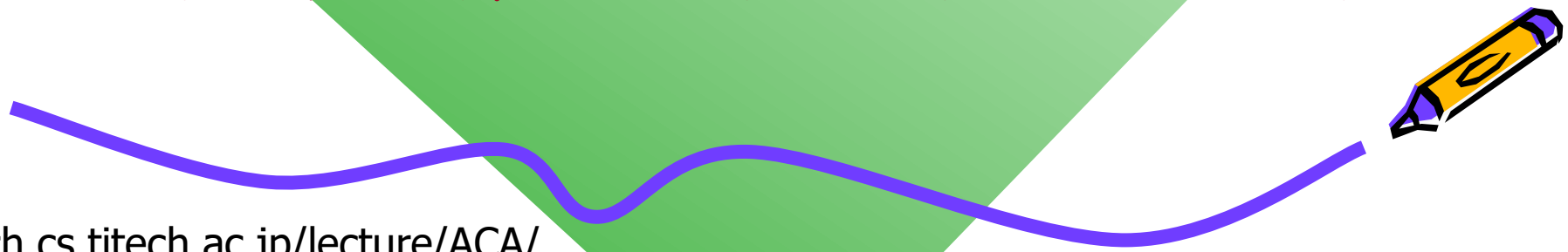
Ver. 2023-01-05a



Course number: CSC.T433
School of Computing,
Graduate major in Computer Science

Advanced Computer Architecture

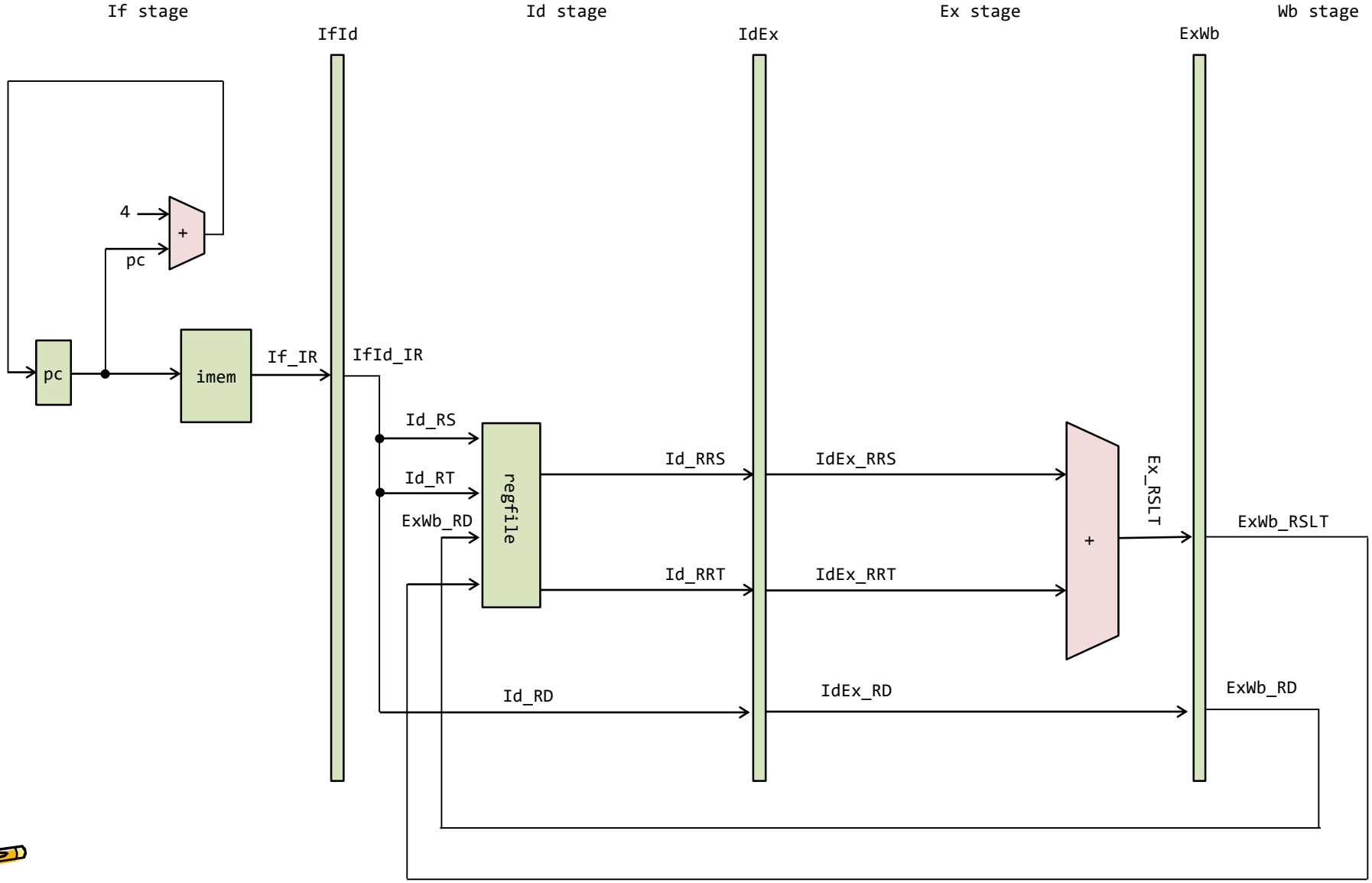
6. Instruction Level Parallelism: Instruction Fetch and Branch Prediction



www.arch.cs.titech.ac.jp/lecture/ACA/
Room No.W831, HyFlex
Mon 13:45-15:25, Thr 13:45-15:25

Kenji Kise, Department of Computer Science
kise _at_ c.titech.ac.jp

Four stage pipelined processor supporting ADD



Assignment 4 (2023-01-05)

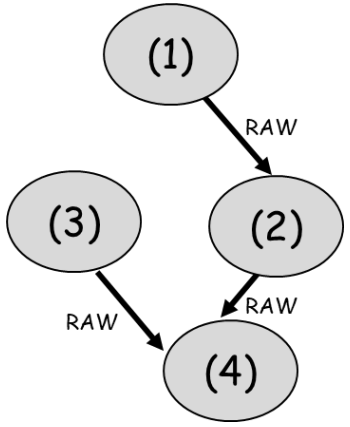
1. Design a four stage pipelined processor supporting MIPS add instructions in Verilog HDL. Please download proc01.v from the support page and refer it.
2. Verify the behavior of designed processor using following assembly code assuming initial values of r[1]=22, r[2]=33, r[3]=44, and r[4]=55
   - `add $0, $0, $0 # NOP {6'h0, 5'd0, 5'd0, 5'd0, 5'd0, 6'h20}`
   - `add $1, $1, $1 #`
   - `add $2, $2, $2 #`
   - `add $3, $3, $3 #`
   - `add $4, $4, $4 #`
3. Submit a report printed on A4 paper at the beginning of the next lecture on **Thursday**. Or,
   Submit your report in a PDF file via E-mail (kise [at] c.titech.ac.jp) by the beginning of the next lecture on **Thursday**.
   - The report should include a block diagram, a source code in Verilog HDL, and obtained waveforms of your design.
   - E-mail title should be "Assignment of Advanced Computer Architecture"

Exploiting Instruction Level parallelism (ILP)

- A superscalar has to handle some flows efficiently to exploit ILP
  - Control flow (control dependence)
    - To execute ***n*** instructions per clock cycle, the processor has to fetch at least ***n*** instructions per cycle.
    - The main obstacles are branch instruction (BNE, BEQ)
    - Prediction
    - Another obstacle is instruction cache
  - Register data flow (data dependence)
    - Out-of-order execution
      - Register renaming
      - Dynamic scheduling
  - Memory data flow
    - Out-of-order execution
    - Another obstacle is data cache

```
(1) add $5,$1,$2
(2) add $9,$5,$3
(3) lw  $4, 4($7)
(4) add $8,$9,$4
```

```
(3) lw  $4, 4($7)
(1) add $5,$1,$2
(2) add $9,$5,$3
(4) add $8,$9,$4
```

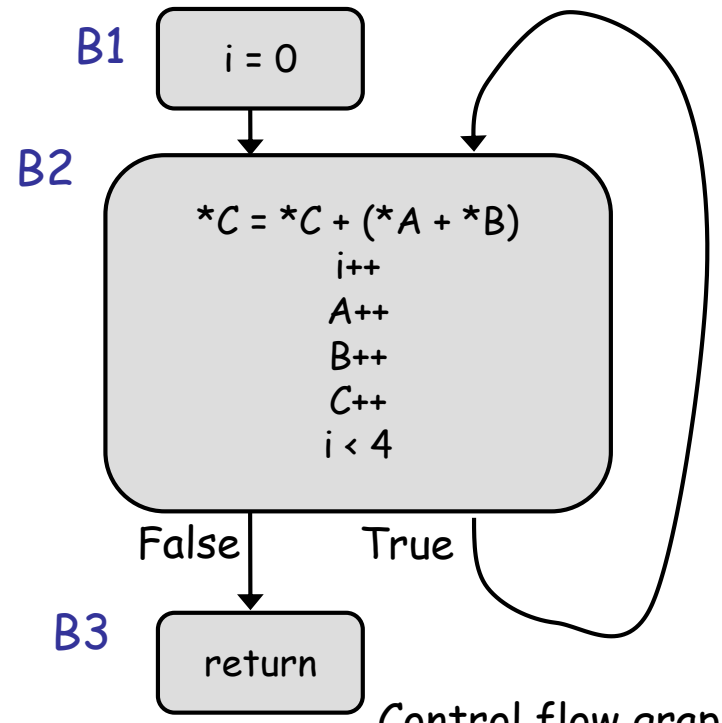


Branch predictor

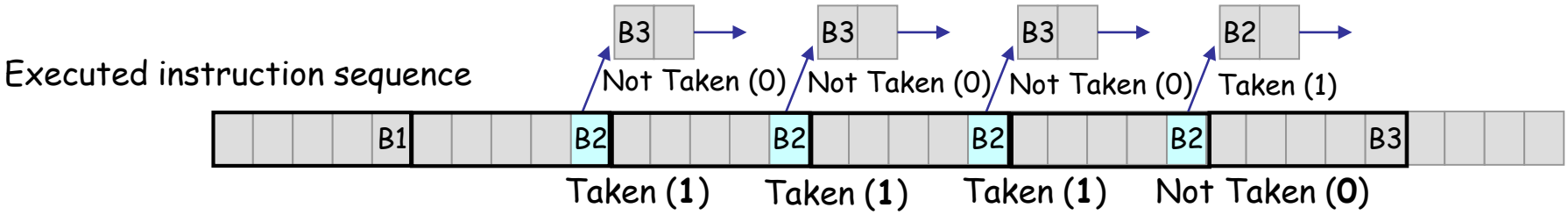
- A branch predictor is a digital circuit that tries to guess or predict which way (taken or untaken) a branch will go before this is known definitively.
  - A random predictor will achieve about a 50% hit rate because the prediction output is 1 or 0.
  - Let's guess the accuracy. What is the accuracy of typical branch predictors for high-performance commercial processors?

Sample program: vector add

```
#define VSIZE 4
void vadd(long *A, long *B, long *C){
  for(i=0; i<VSIZE; i++)
    C[i] += (A[i] + B[i]);
}
```



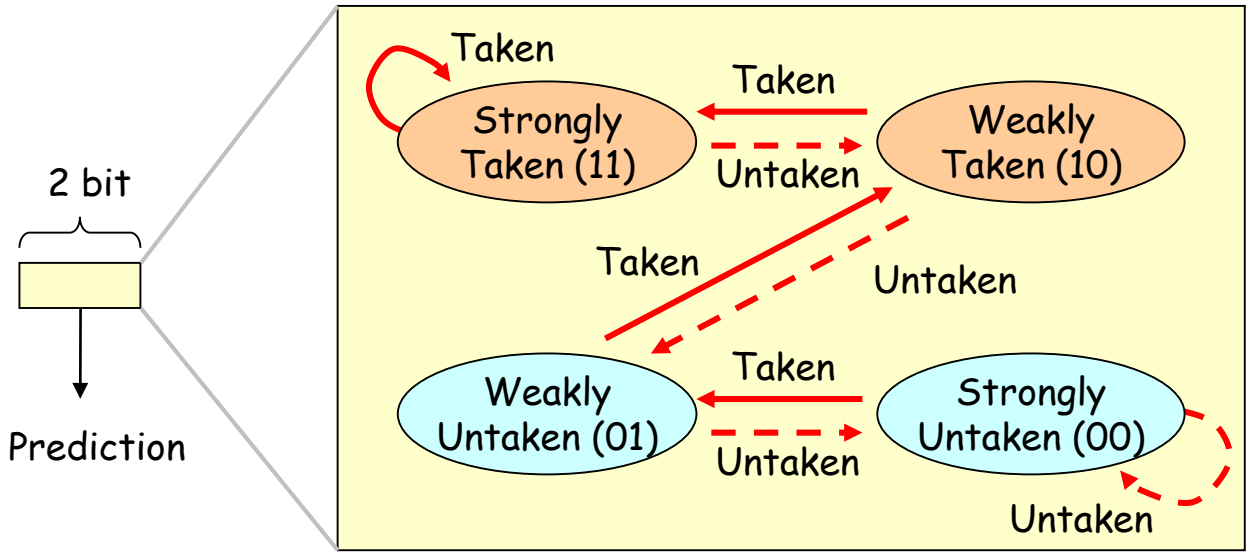
Control flow graph



Predicting the sequence of 1110 1110 1110 1110 1110 …

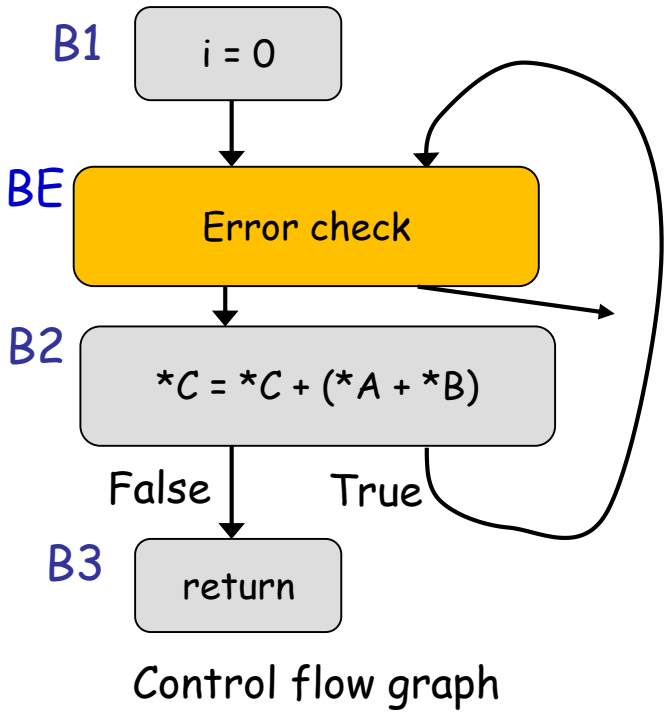
Simple branch predictor: 2bit counter

- It uses two bit register or a counter.
- How to update the register
  - If the branch outcome is taken and the value is not 3, then increment the register.
  - If the branch outcome is untaken and the value is not 0, then decrement the register.
- Hot to predict
  - It predicts as 1 if the MSB of the register is one, otherwise predicts as 0.

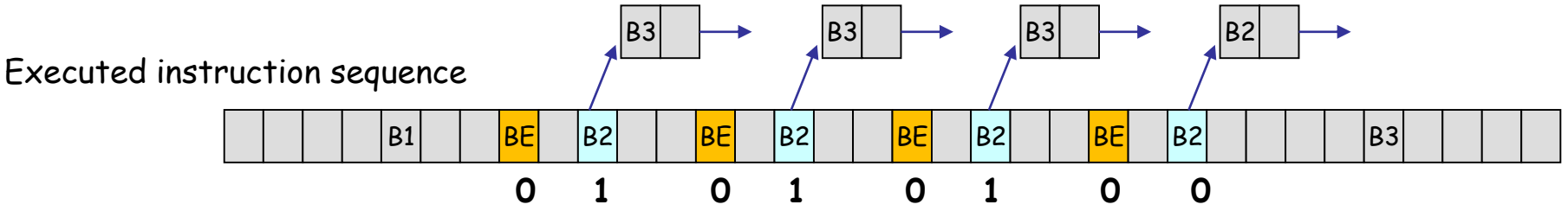


Sample program: vector add with two branches

```
#define VSIZE 4
void vadd(long *A, long *B, long *C){
  for(i=0; i<VSIZE; i++) {
    if(A[i]<0) error_routine();
    C[i] += (A[i] + B[i]);
  }
}
```

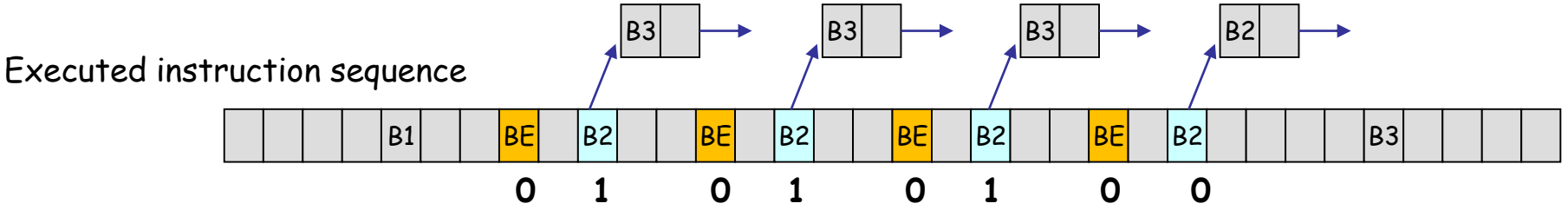


Control flow graph



Predicting the sequence of 01010100 01010100 01010100 …

Sample program: vector add with two branches



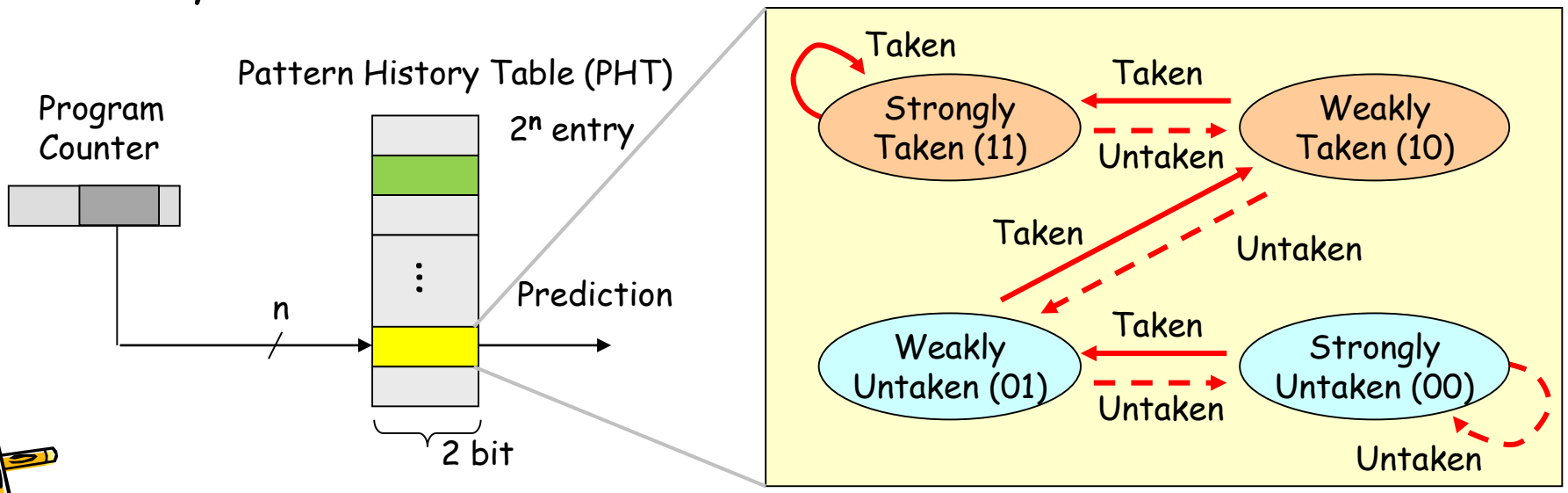
Predicting the sequence of 01010100 01010100 01010100 …

The BE's sequence of 01010100 01010100 01010100 …

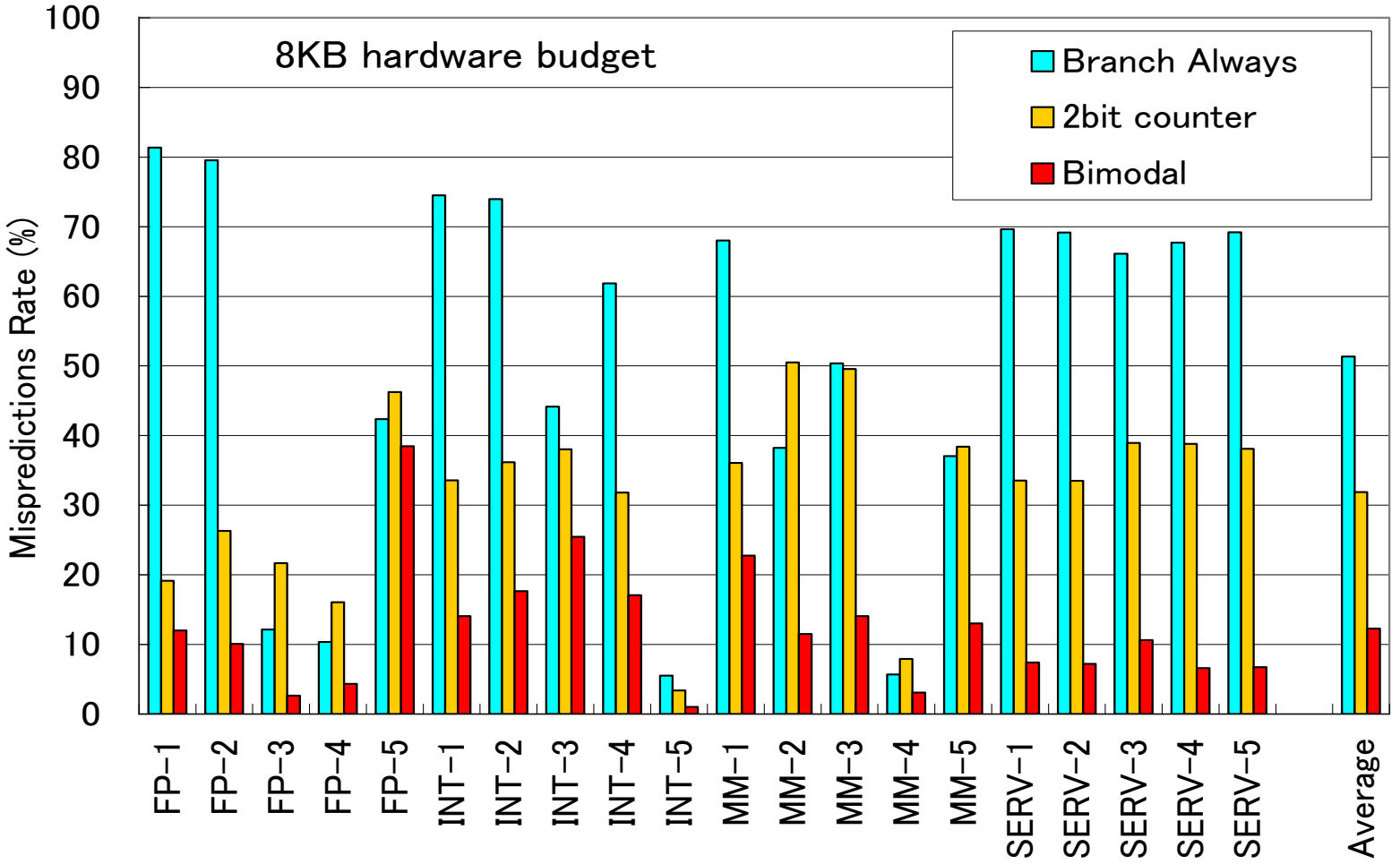
The B2's sequence of 01010100 01010100 01010100 …

Simple branch predictor: bimodal

- Program has many static branch instructions. The behavior may depend on each branch. Use one counter for one branch instruction
- How to predict
  - Select one counter using PC, then it predicts 1 if the MSB of the register is one, otherwise predicts 0.
- How to update
  - Select one counter using PC, then update the counter in the same way as 2bit counter.

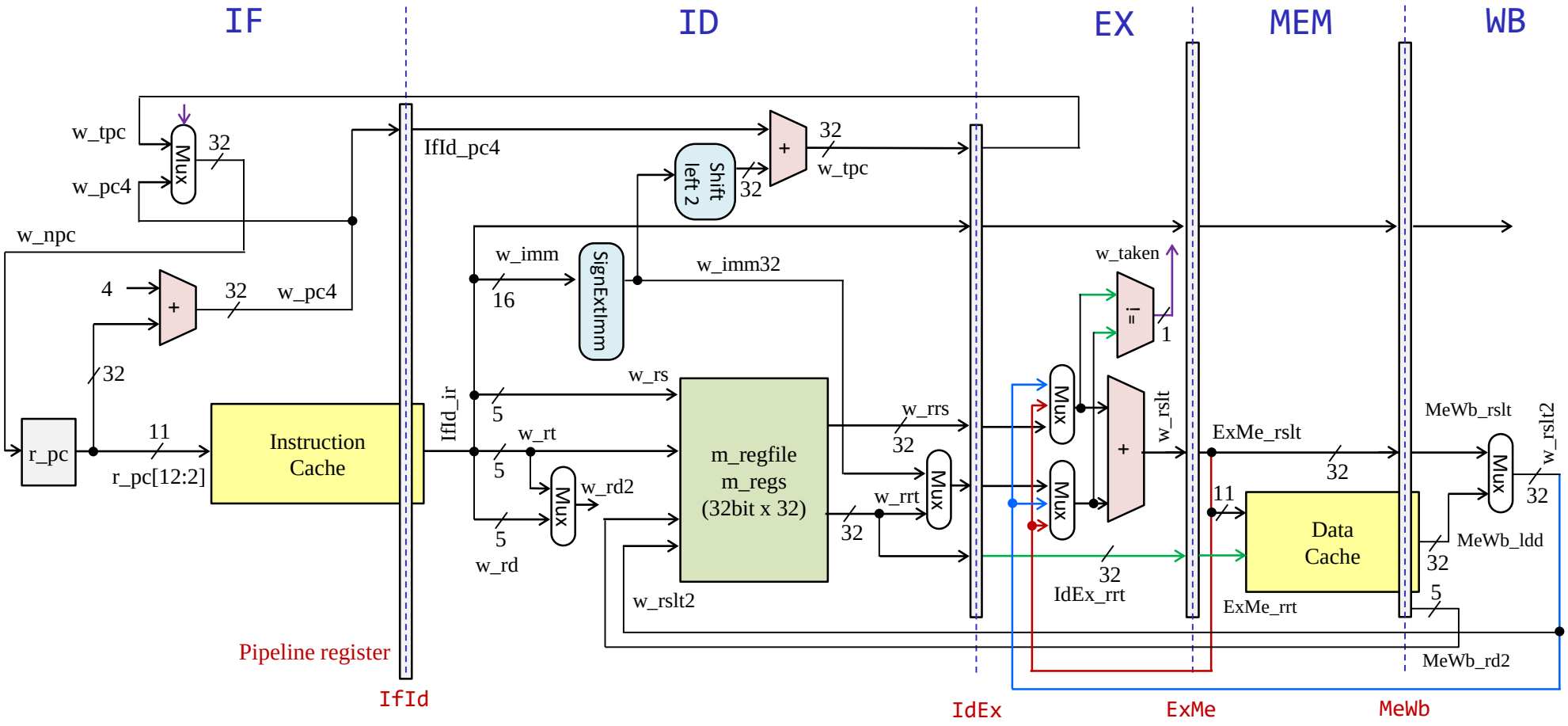


Accuracy of simple predictors with 8KB HW budget



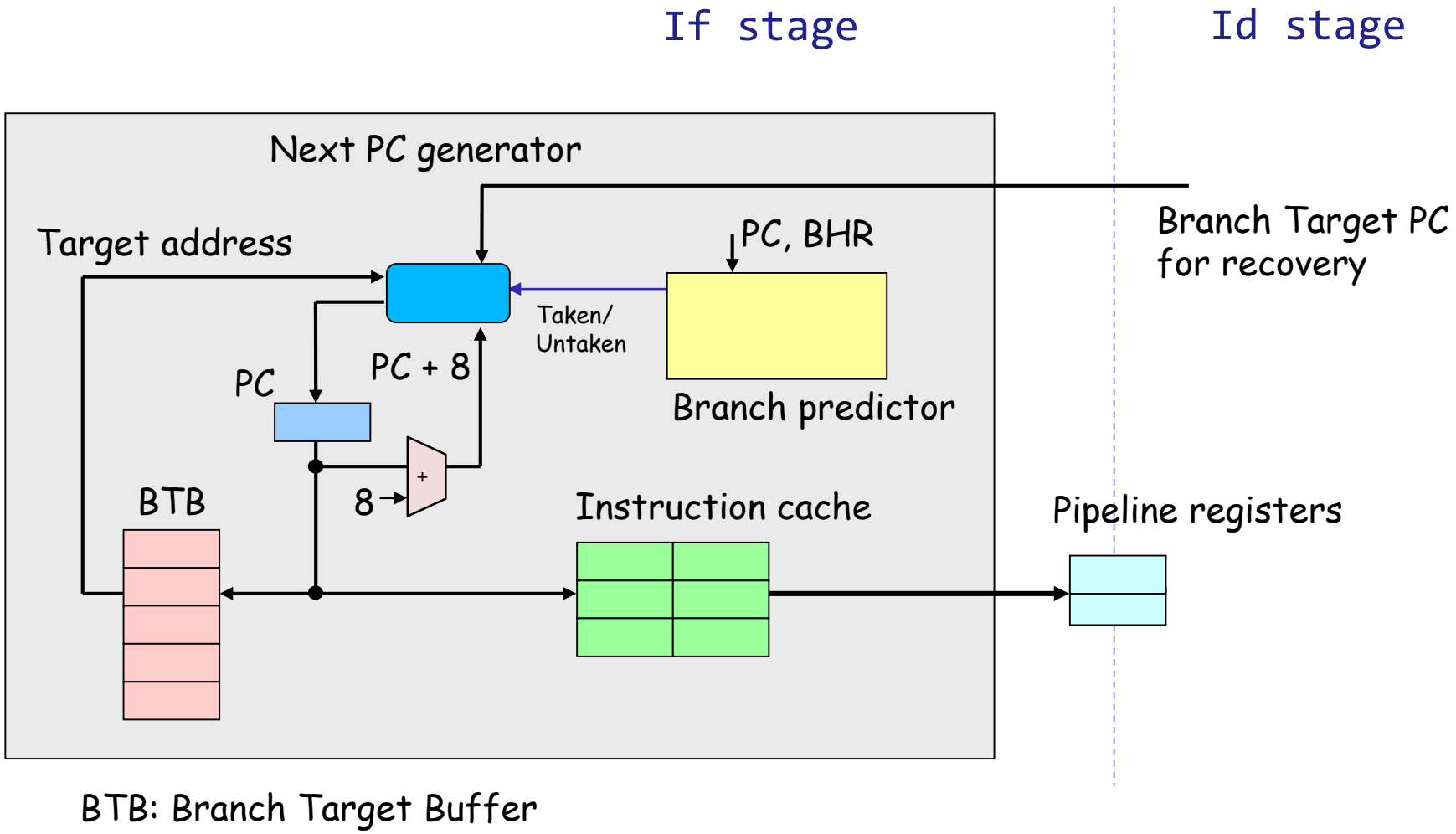
Benchmark for CBP(2004) by Intel MRL and IEEE TC uARCH.

Pipelined MIPS processor of five stages



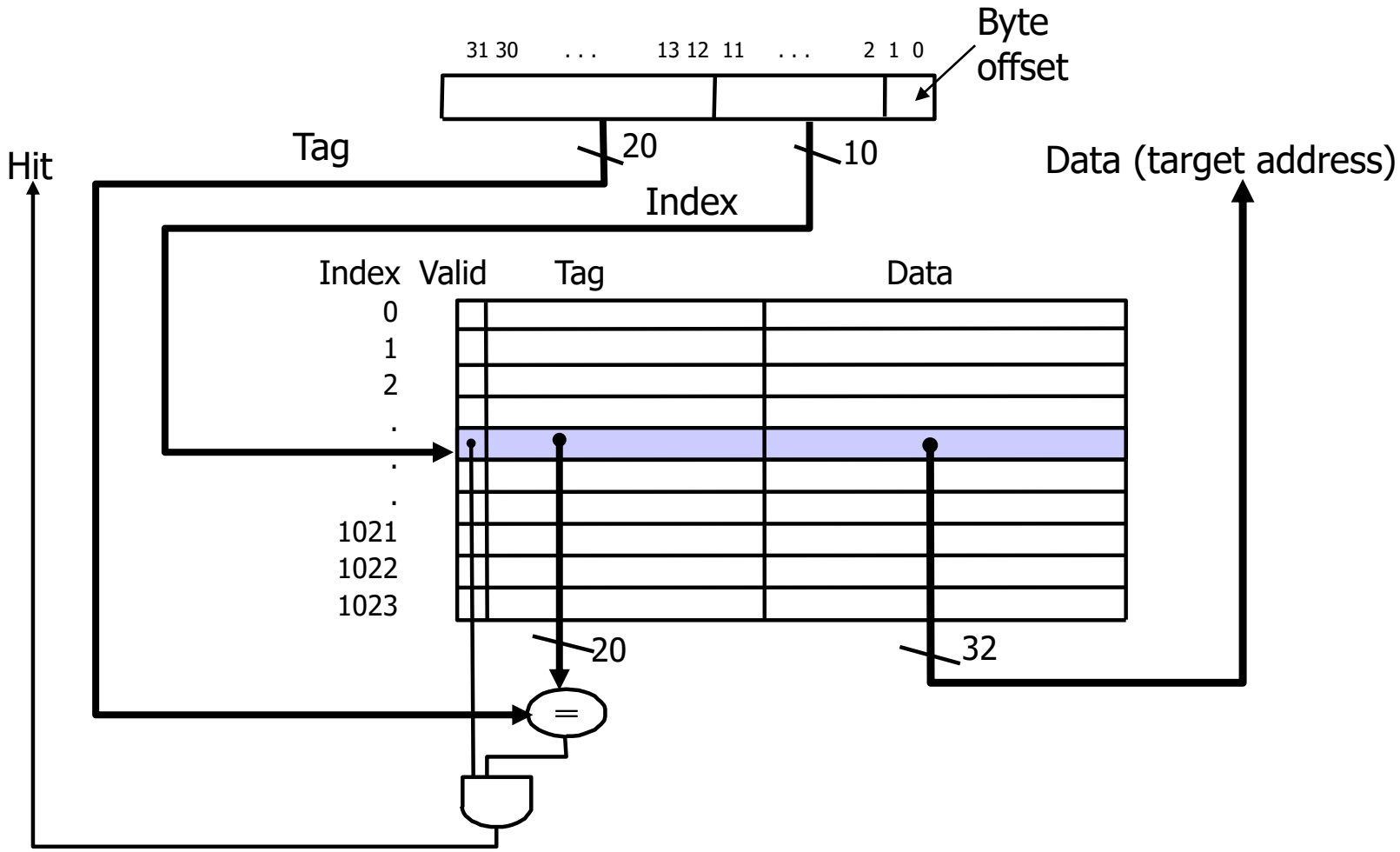
Instruction fetch unit of super-scalar in IF stage

- For high-bandwidth instruction delivery, prediction, and speculation



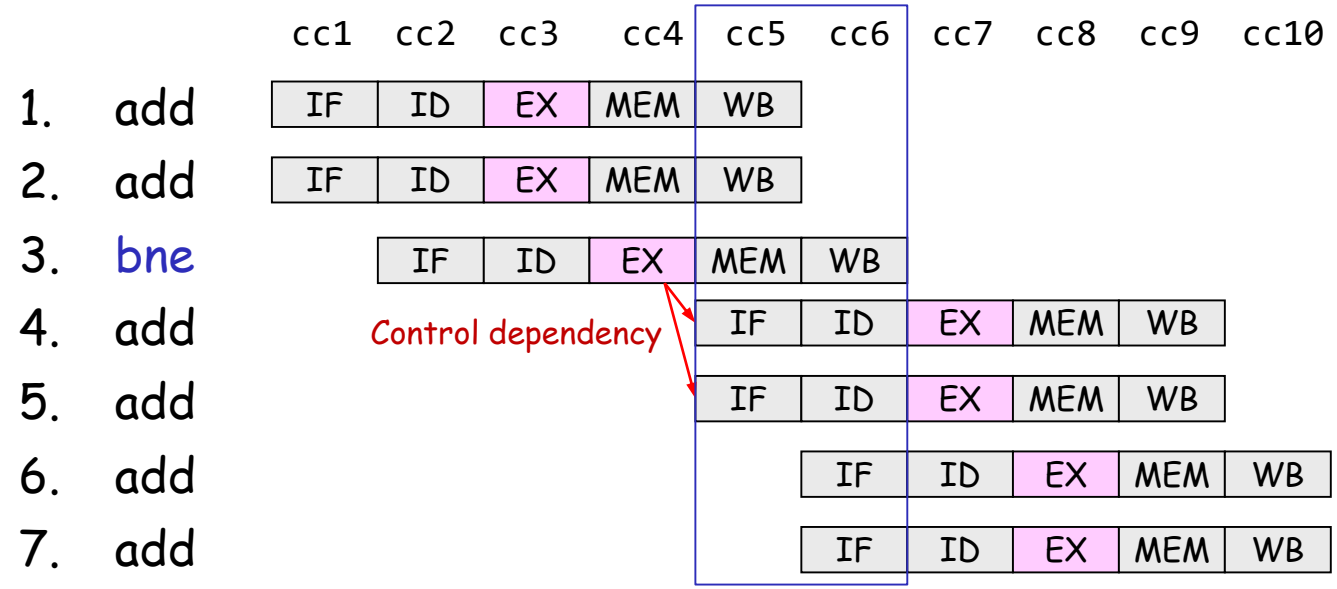
Simple Branch Target Buffer (BTB) Example

- memory size of 4KB



Why do branch instructions degrade IPC?

- The branch taken / untaken is determined in execution stage of the branch.
- The conservative approach of stalling instruction fetch until the branch direction is determined.

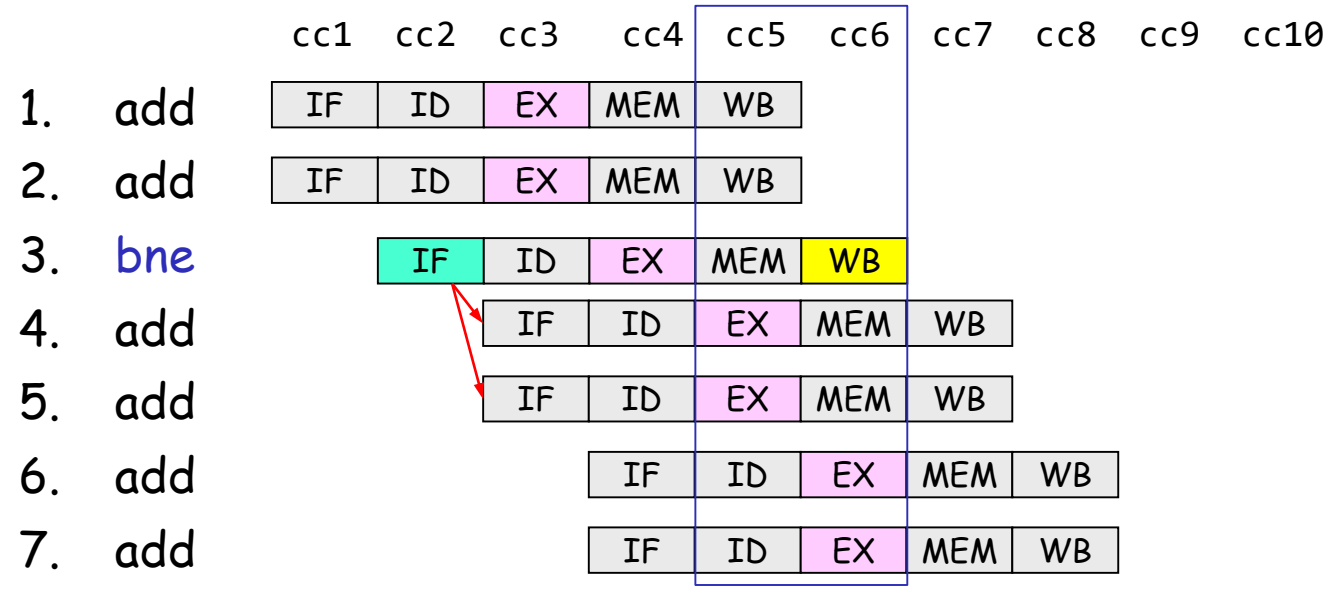


2-way superscalar processor executing instruction sequence with a branch

Note that because of a branch instruction, only one instruction is executed in cc4 and no instructions are executed in CC6 and CC7. This reduces the IPS.

Why do branch instructions degrade IPC?

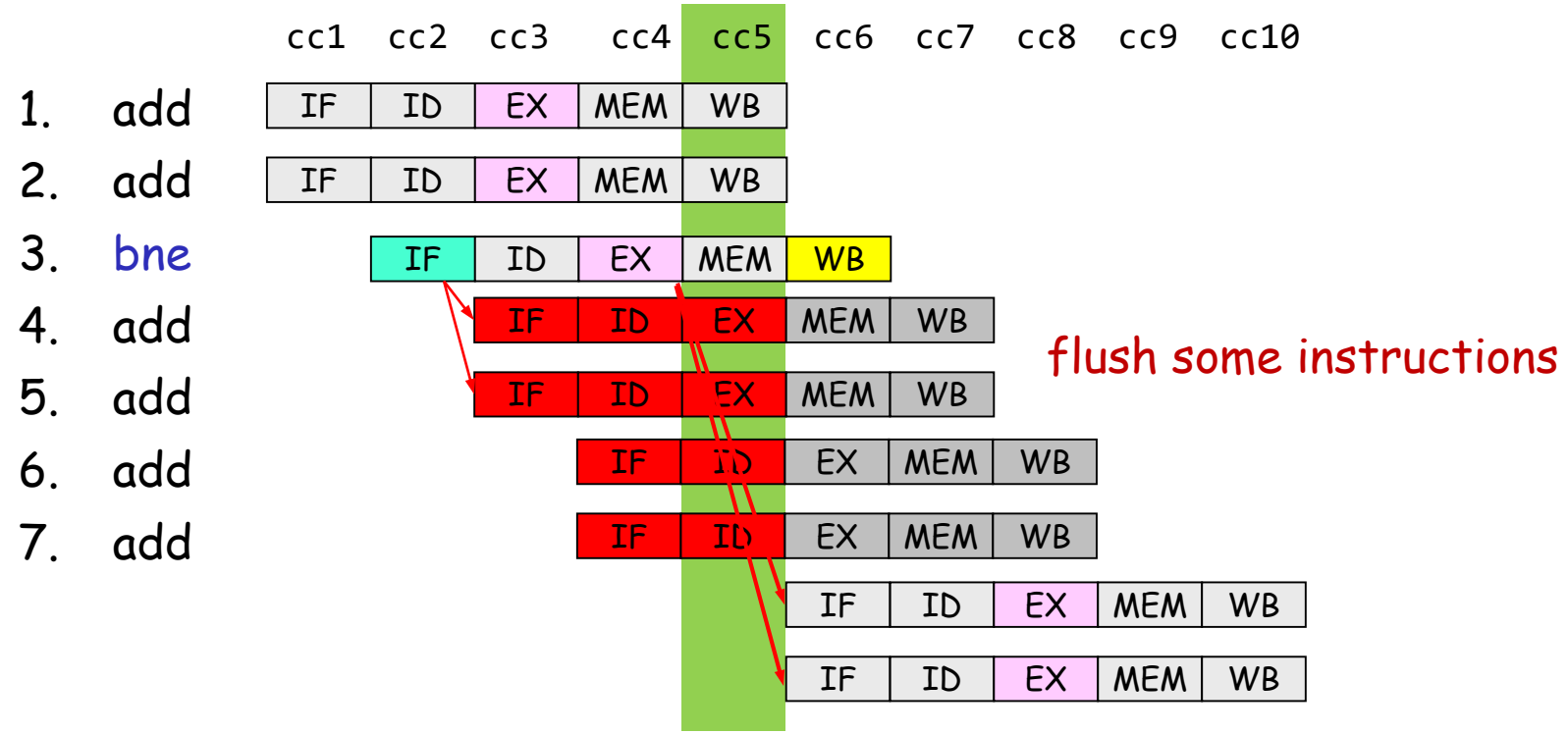
- The branch taken / untaken is determined in execution stage of the branch.
- Prediction and speculation, then training
- Recovery whan a prediction miss



2-way superscalar processor executing instruction sequence with a branch

Why do branch instructions degrade IPC?

- The branch taken / untaken is determined in execution stage of the branch.
- Prediction and speculation, then training
- Recovery whan a prediction miss



2-way superscalar processor executing instruction sequence with a branch

An innovation in branch predictors in 1993

- Using branch history
  - global branch history
  - local branch history
- 2-level branch predictor and Gshare
- Assume predicting the sequence 1110 1110 1110 1110 1110 …

```
1110111 0
11101110 ?
111011101 ?
1110111011 ?
11101110111 ?
111011101110 ?
```

adr	pred
000	
001	
010	
011	1
100	
101	1
110	1
111	0

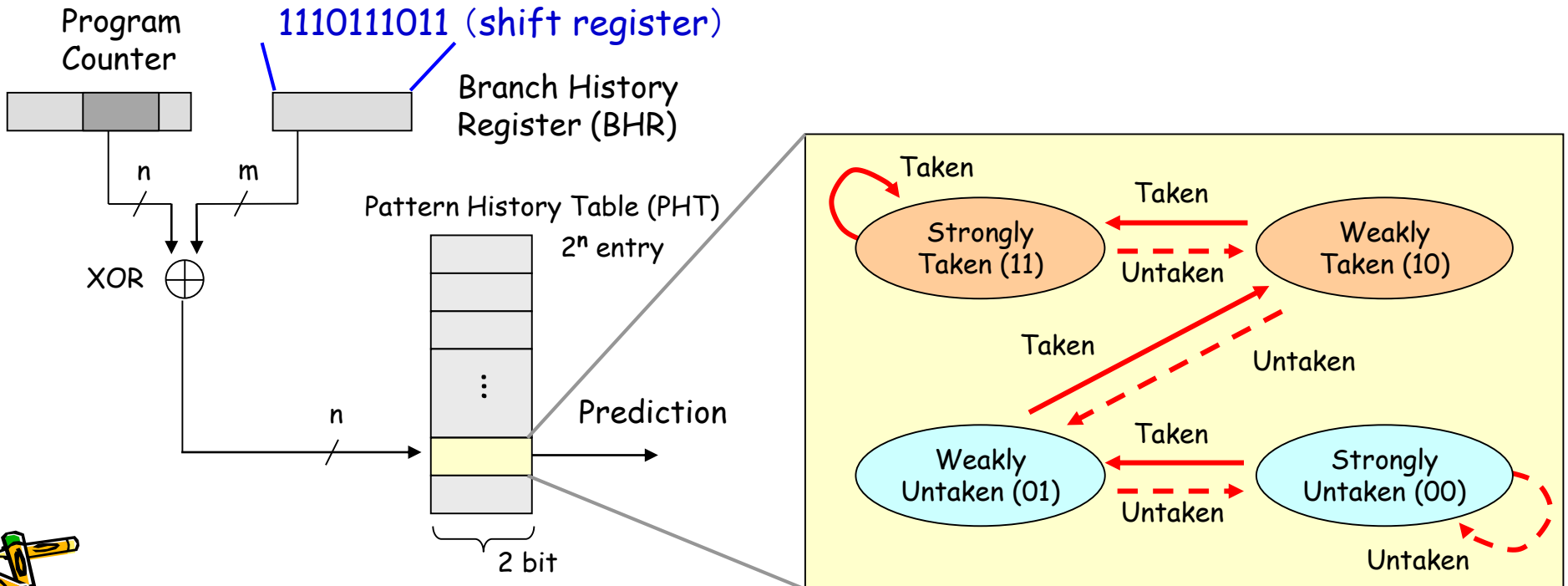
Recommended Reading

- Combining Branch Predictors
  - Scott McFarling, Digital Western Research Laboratory
  - WRL Technical Note TN-36, 1993

- *A quote:*
  "In this paper, we have presented two new methods for improving branch prediction performance. First, we showed that using the bit-wise exclusive OR of the global branch history and the branch address to access predictor counters results in better performance for a given counter array size."

Gshare (TR-DEC 1993)

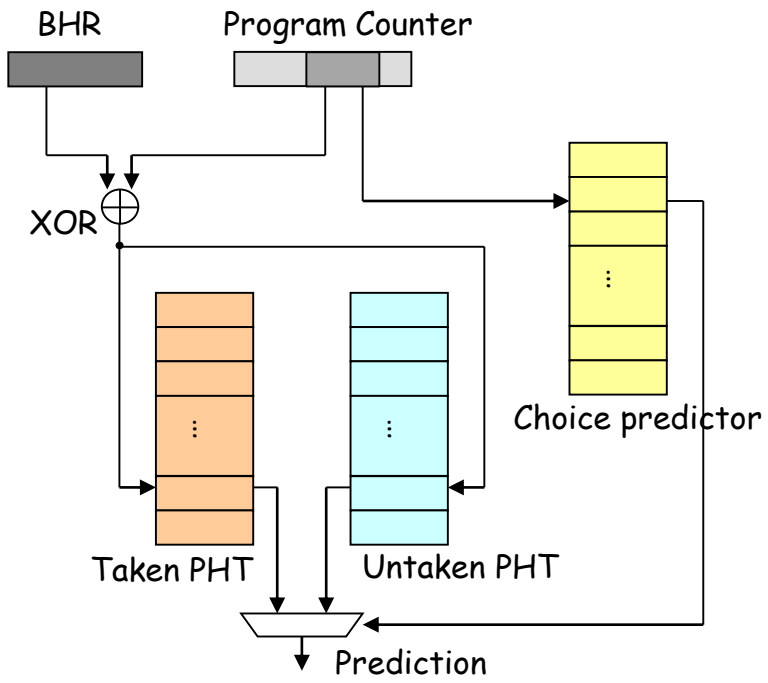
- How to predict
  - Using the exclusive OR of the global branch history and PC to access PHT, then MSB of the selected counter is the prediction.
- How to update
  - Shifting BHR one bit left and update LSB by branch outcome in IF stage.
  - Update the used counter in the same way as 2BC in WB stage.



this slide is to be used as a whiteboard

Bi-Mode (MICRO 1997)

- A choice predictor (bimodal) is used as a meta-predictor
- How to predict
  - Like Gshare, both of Taken PHT and Untaken PHT make two predictions.
  - Select one among them by the choice predictor which tracks the global bias of a branch.
- How to update
  - The used PHT is updated in the same way as 2BC.
  - Choice predictor is update in the same way as bimodal



YAGS, Yet Another Global Scheme (MICRO 1998)

- Using two tagged PHTs
- When a PHT miss, choice PHT makes a prediction.

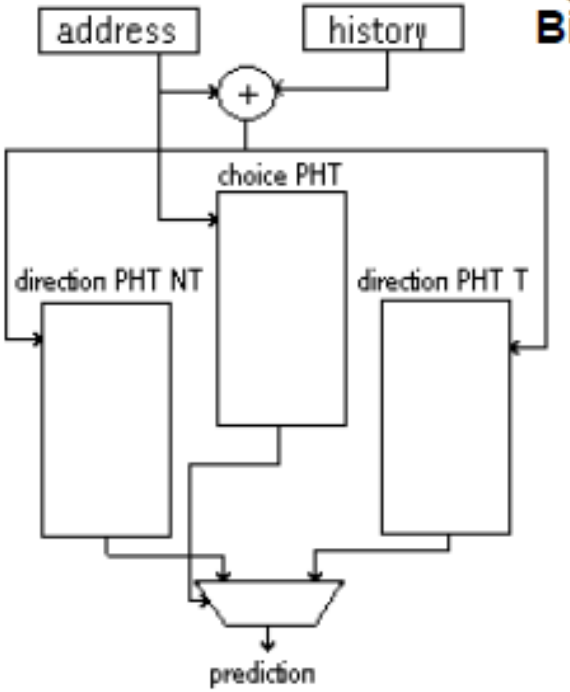


Figure 3. Bi-Mode

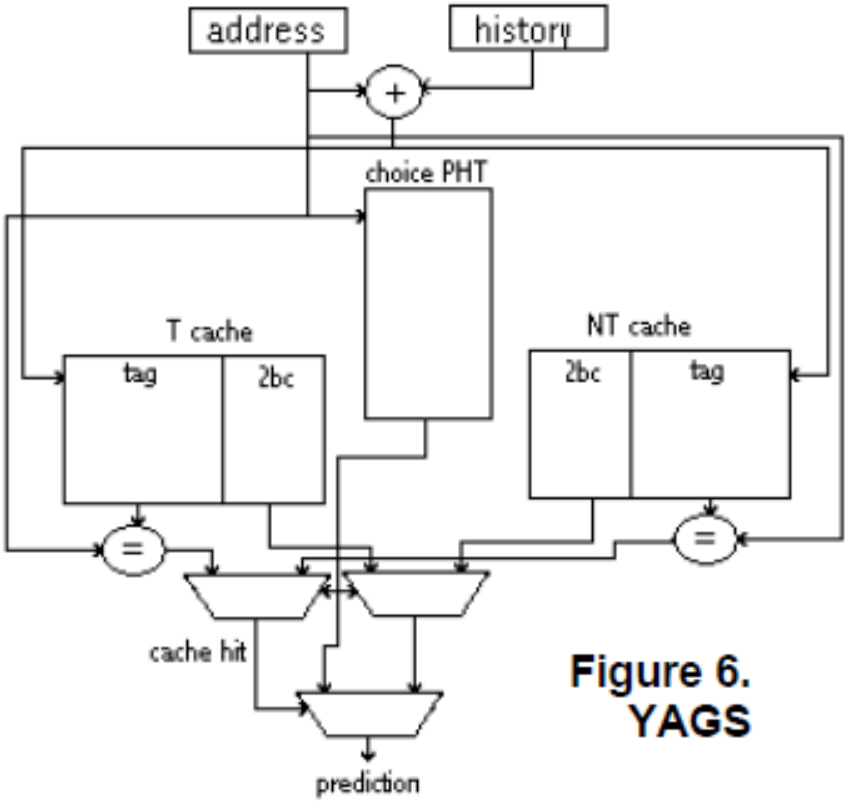


Figure 6. YAGS

From YAGS paper

this slide is to be used as a whiteboard

An innovation in branch predictors in 1993 (again)

- Using branch history
  - global branch history
  - local branch history
- 2-level branch predictor and Gshare

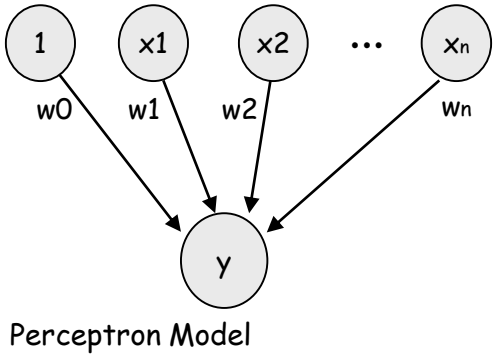
- Assume predicting the sequence 1110 1110 1110 1110 1110 …

```
11101110 ?
111011101 ?
1110111011 ?
11101110111 ?
111011101110 ?
```

```
11101110 ?
111011101 ?
1110111011 ?
11101110111 ?
111011101110 ?
```

Perceptron (HPCA 2001)

- How to predict
  - Select one perceptron by PC
  - Compute y using the equation. It predicts 1 if y>=0, predicts 0 if y<0
- How to update
  - Train the weights of used perceptron when the prediction miss or |y| < T



Perceptron Model

$$y = w_0 + \sum_{i=1}^{n} x_i w_i.$$

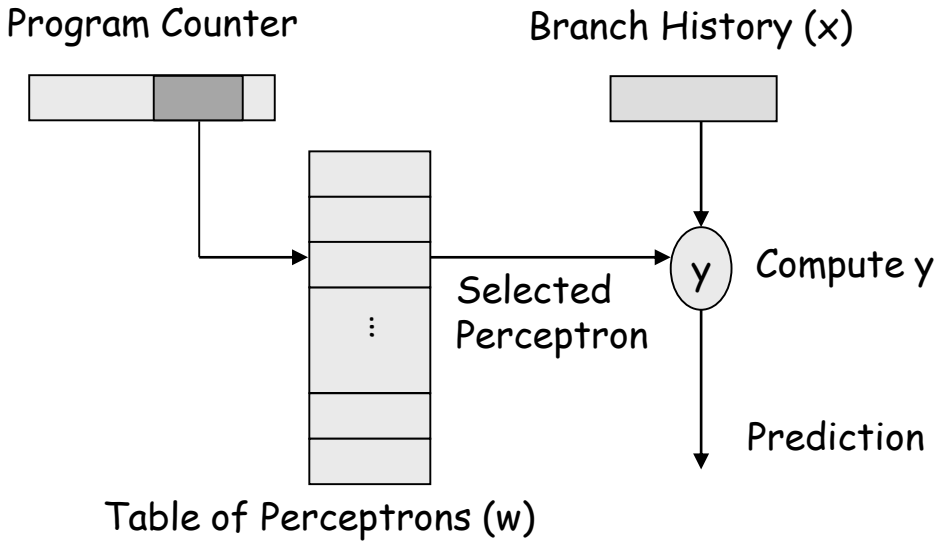
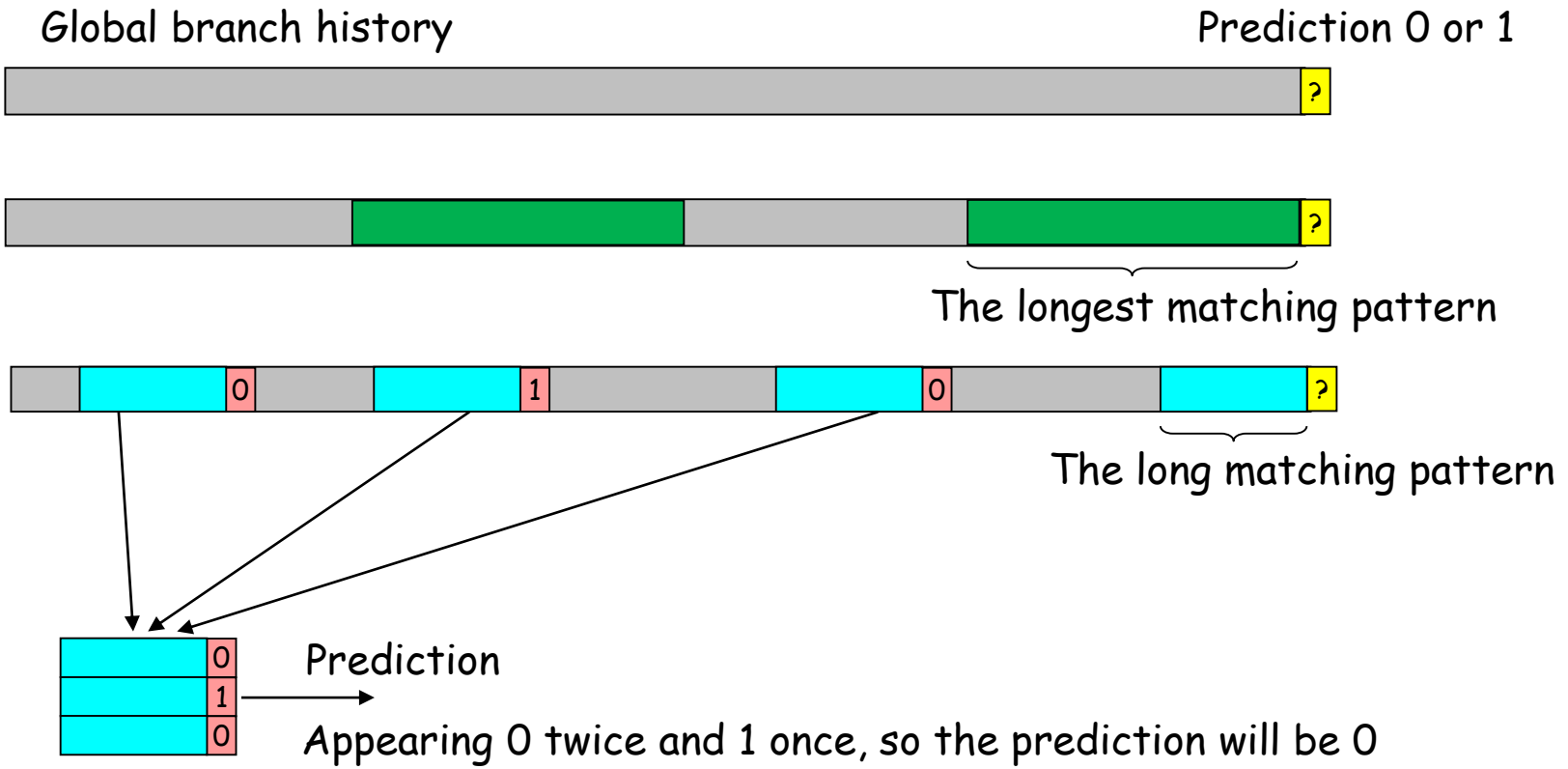


Table of Perceptrons (w)

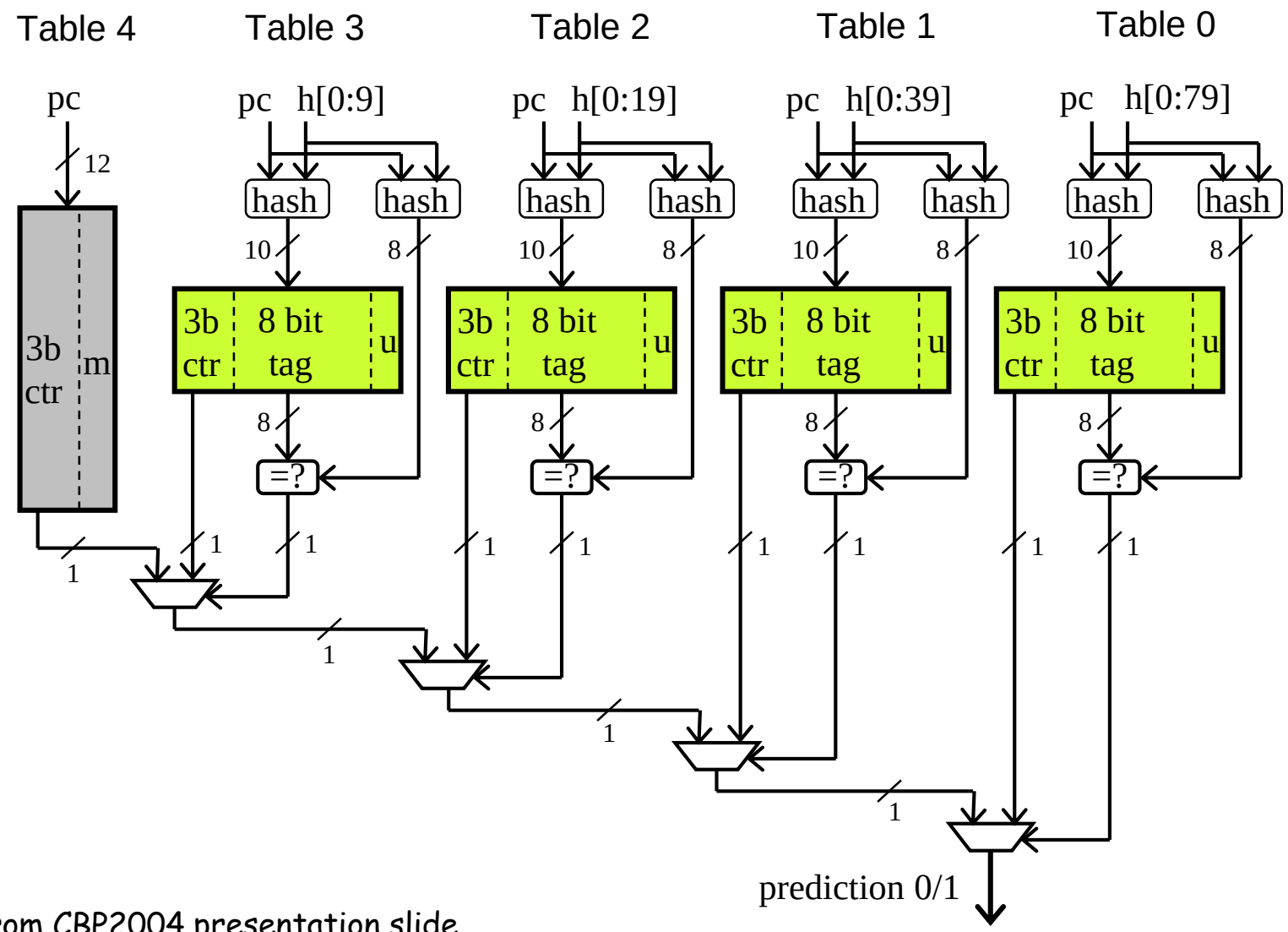
this slide is to be used as a whiteboard

Branch predictors based on pattern matching

- Find the longest matching pattern (green rectangle)
- Select the proper matching length or long matching pattern (blue rectangle)
- Count the number of 0 and the number of 1 after the pattern (red rectangle), then predict.



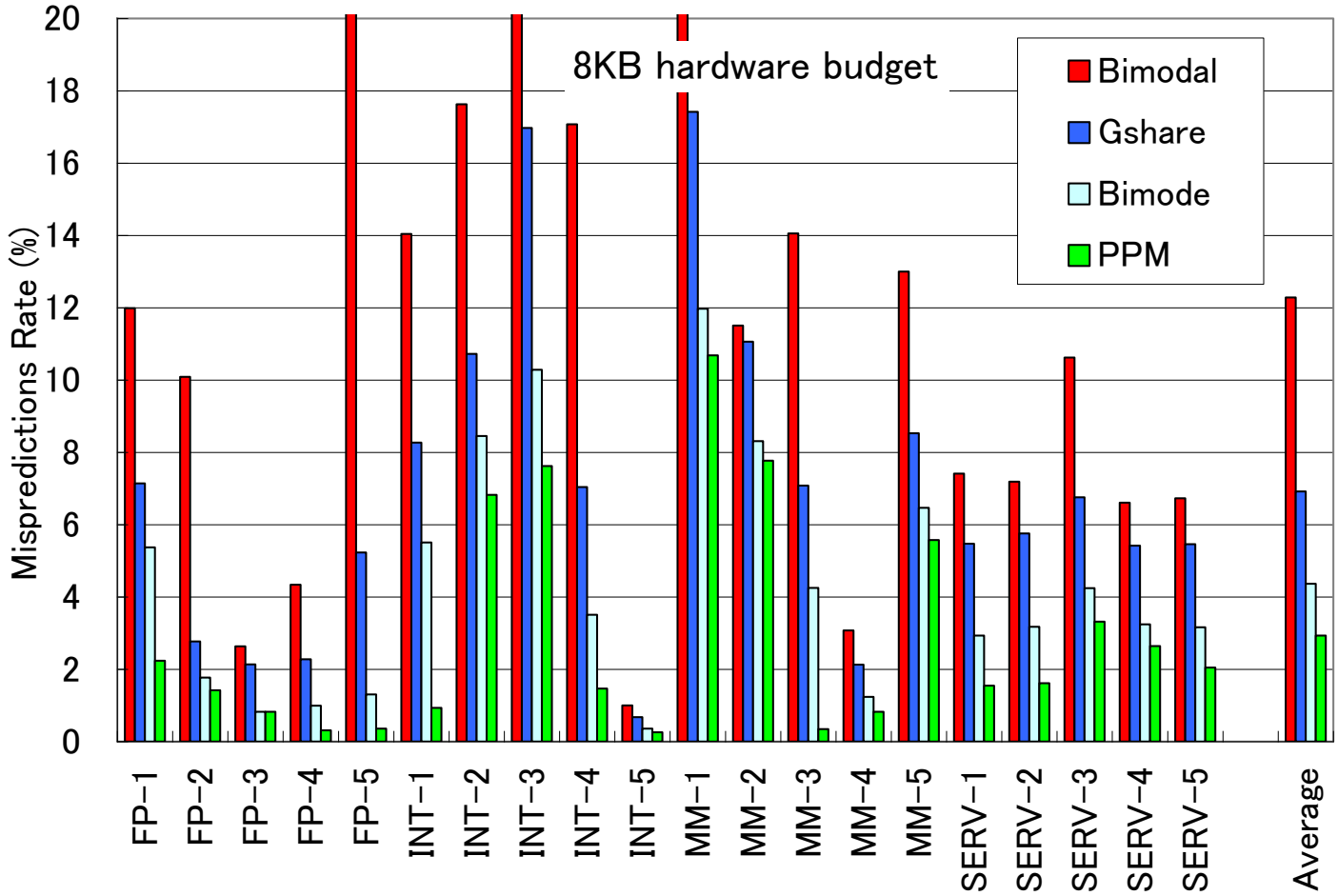
Partial Pattern Matching (CBP 2004)



From CBP2004 presentation slide

Prediction accuracy

- The accuracy of 4KB Gshare is about 93%.
- The accuracy of 4KB PPM is about 97%.



Recommended Reading

- Prophet-Critic Hybrid Branch Prediction
  - Ayose Falcon, UPC, Jared Stark, Intel, Alex Ramirez, UPC, Konrad Lai, Intel, Mateo Valero
  - ISCA-31 pp. 250-261 (2004)

A quote from Introduction (1/2)

Conventional predictors are analogous to a taxi with just one driver.

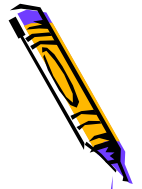
He gets the passenger to the destination using knowledge of the roads acquired from previous trips; i. e., using history information stored in the predictor's memory structures.

When he reaches an intersection, he uses this knowledge to decide which way to turn.

The driver accesses this knowledge in the context of his current location.

Modern branch predictors access it in the context of the current location (the program counter) plus a history of the most recent decisions that led to the current location.

A quote from Introduction (2/2)

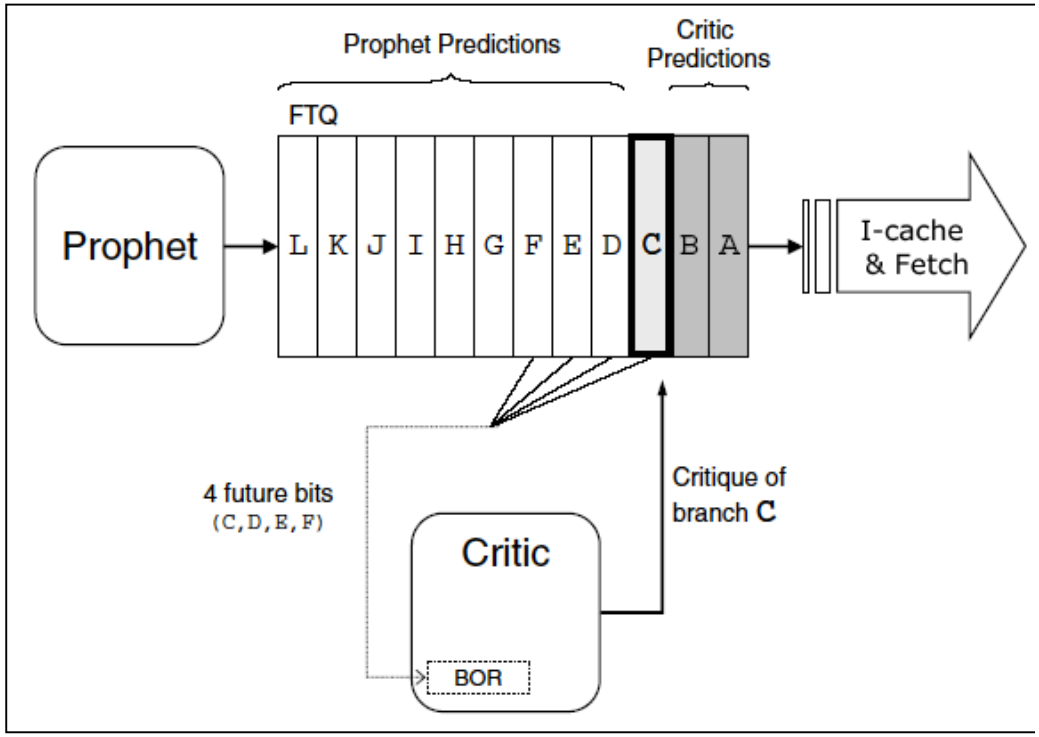


Prophet/critic hybrids are analogous to a taxi with two drivers: the front-seat and the back-seat. The front-seat driver has the same role as the driver in the single-driver taxi. This role is called the prophet.

The back-seat driver has the role of critic. She watches the turns the prophet makes at intersections. She doesn't say anything unless she thinks he's made a wrong turn. When she thinks he's made a wrong turn, she waits until he's made a few more turns to be certain they are lost. (Sometimes the prophet makes turns that initially look questionable, but, after he makes a few more turns, in hindsight appear to be correct.) Only when she's certain does she point out the mistake.

To recover, they backtrack to the intersection where she believes the wrong-turn was made and try a different direction.

Prophet-Critic Hybrid Branch Prediction



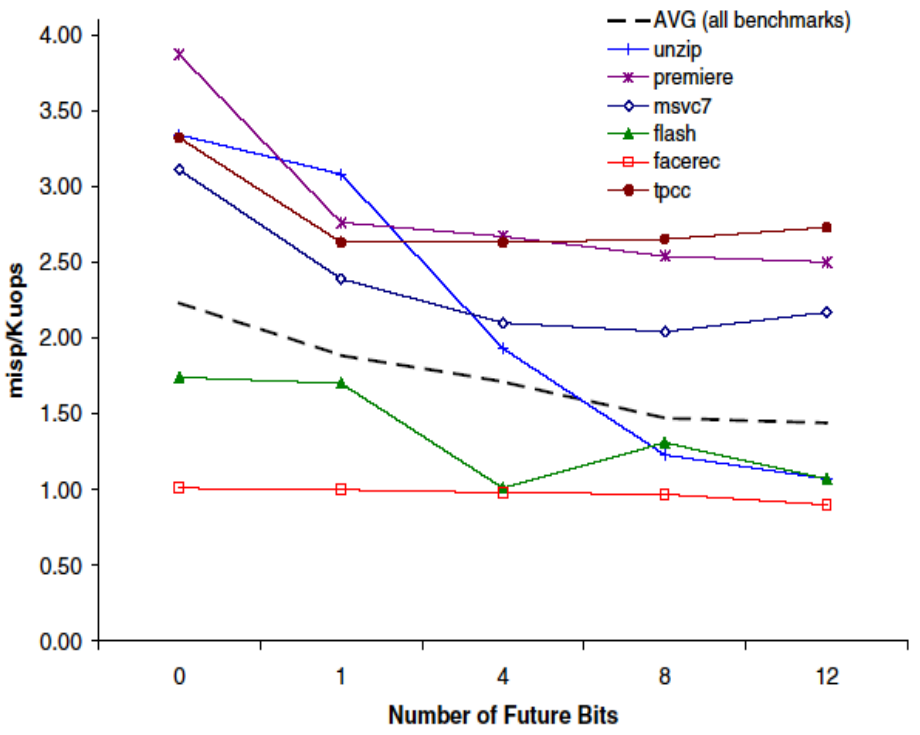


Figure 5. Effect of varying the number of future bits used by the critic on prediction accuracy for selected benchmarks. (prophet: 8KB perceptron; critic: 8KB tagged gshare)

this slide is to be used as a whiteboard