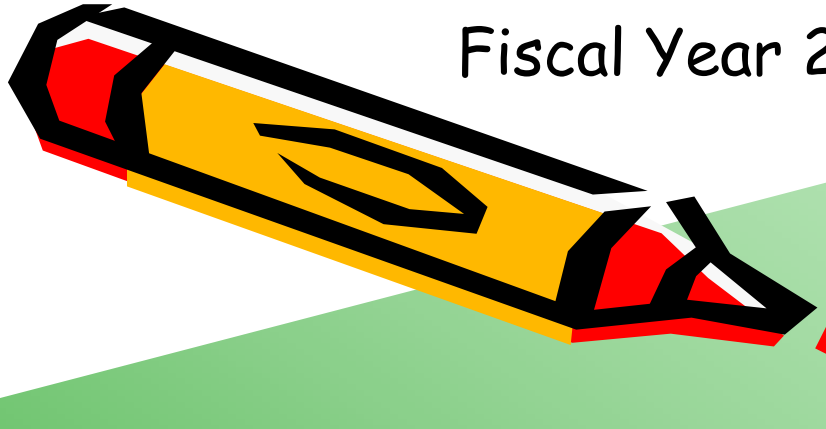
Fiscal Year 2018

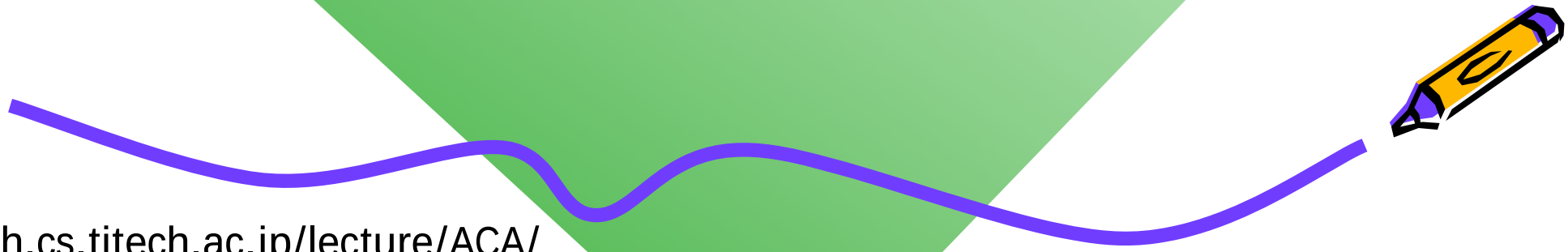
Ver. 2018-12-19a



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

Advanced Computer Architecture

6. Instruction Level Parallelism: Instruction Fetch and Branch Prediction

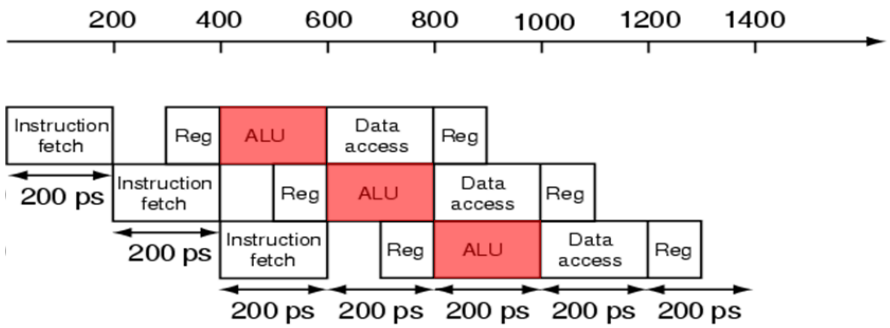


www.arch.cs.titech.ac.jp/lecture/ACA/
Room No.W936
Mon 13:20-14:50, Thr 13:20-14:50

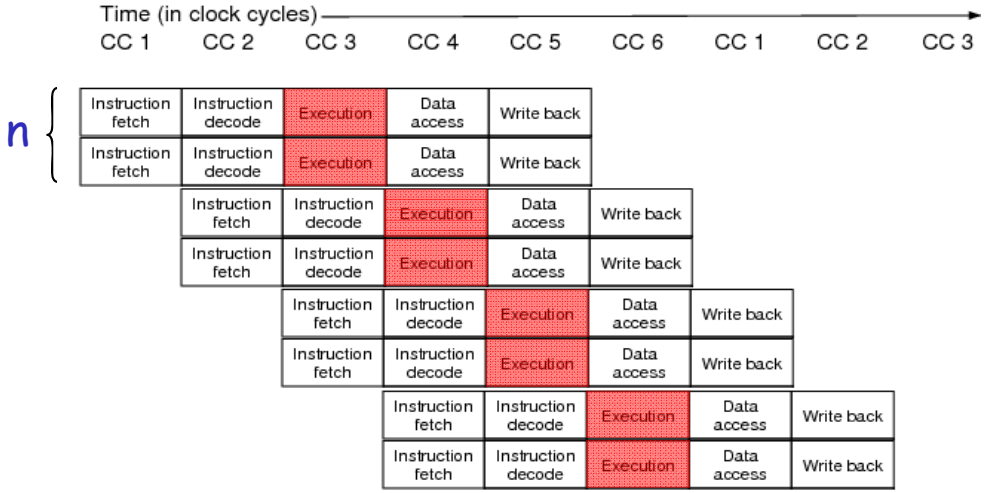
Kenji Kise, Department of Computer Science
kise _at_ c.titech.ac.jp

Scalar and Superscalar processors

- Scalar processor can execute at most one single instruction per clock cycle using one ALU.
  - IPC (Executed Instructions Per Cycle) is less than 1.
- Superscalar processor can execute more than one instruction per clock cycle by executing multiple instructions using multiple pipelines.
  - IPC (Executed Instructions Per Cycle) can be more than 1.
  - using n pipelines is called n-way superscalar

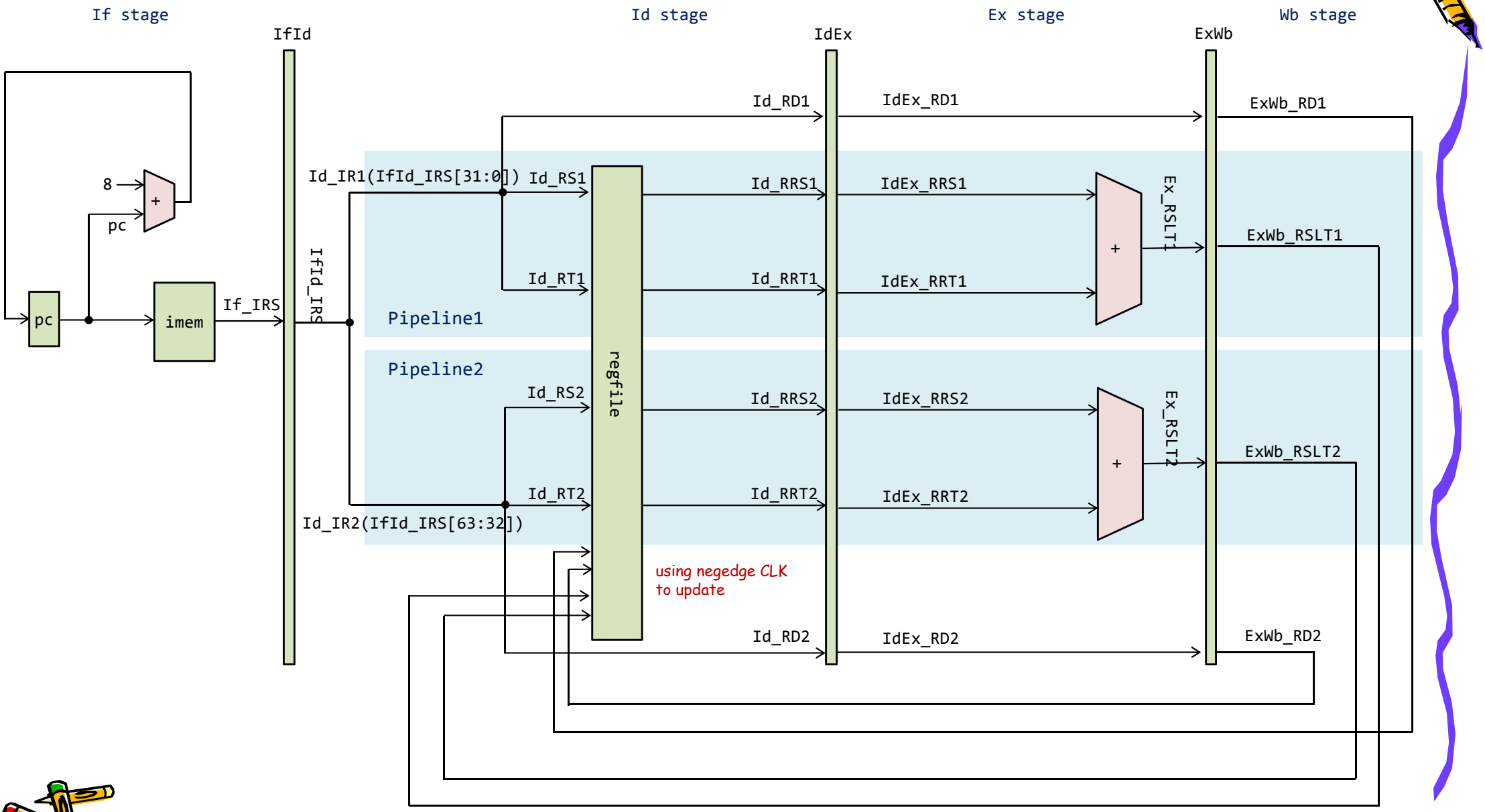


(a) pipeline diagram of scalar processor

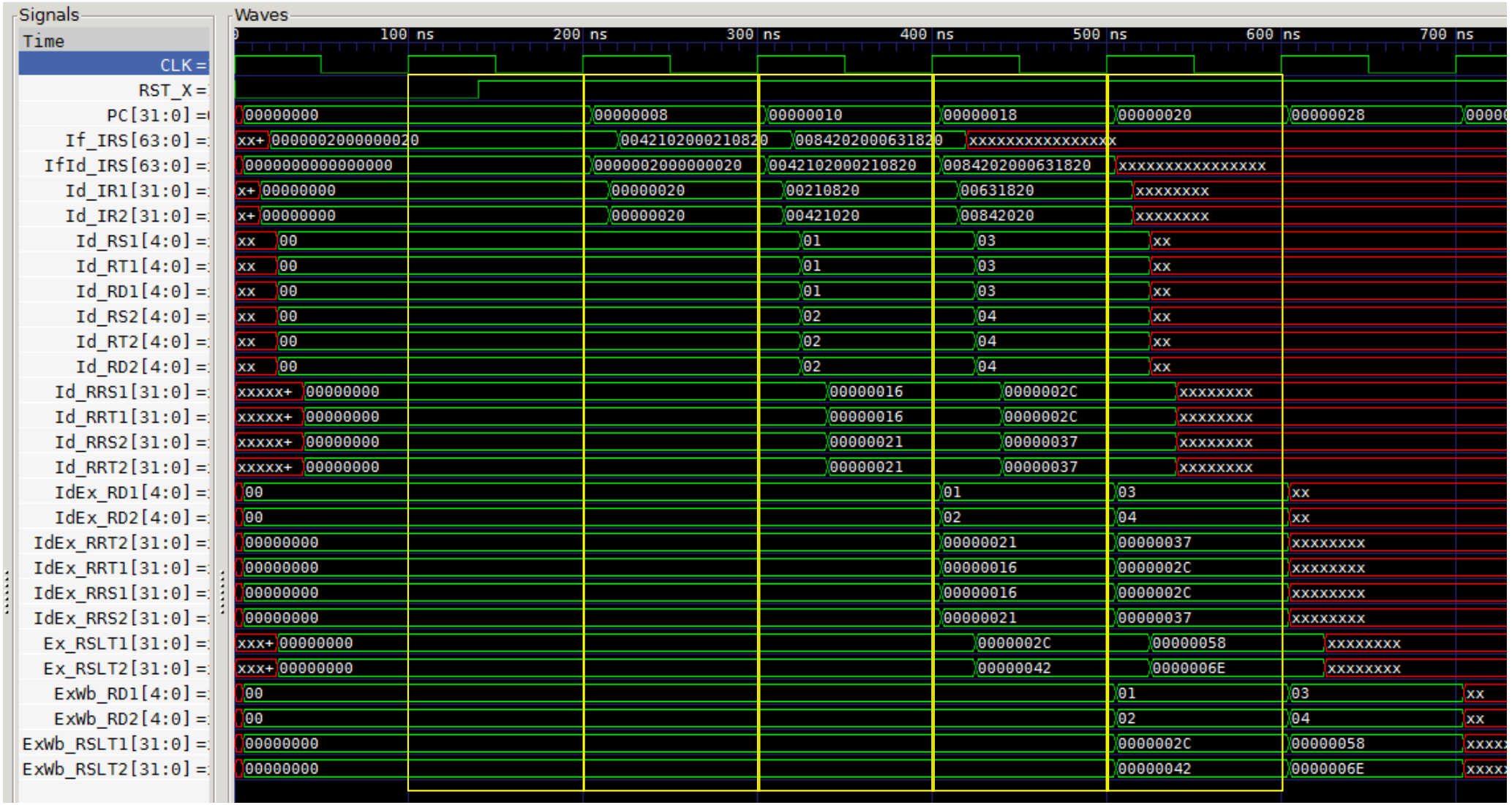


(b) pipeline diagram of 2-way superscalar processor

A four stage pipelined 2-way superscalar processor supporting ADD which does not adopt data forwarding (proc10, Homework 5)

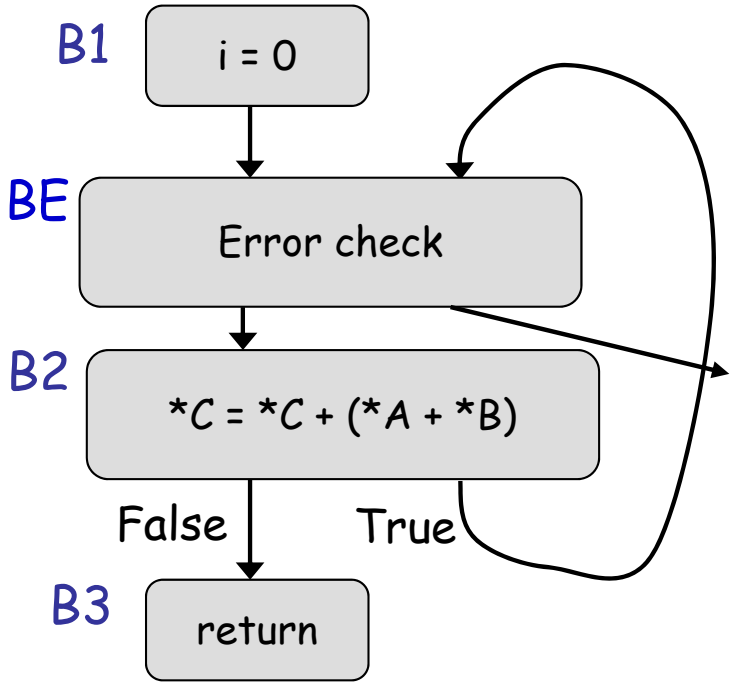


Waveform of Proc10



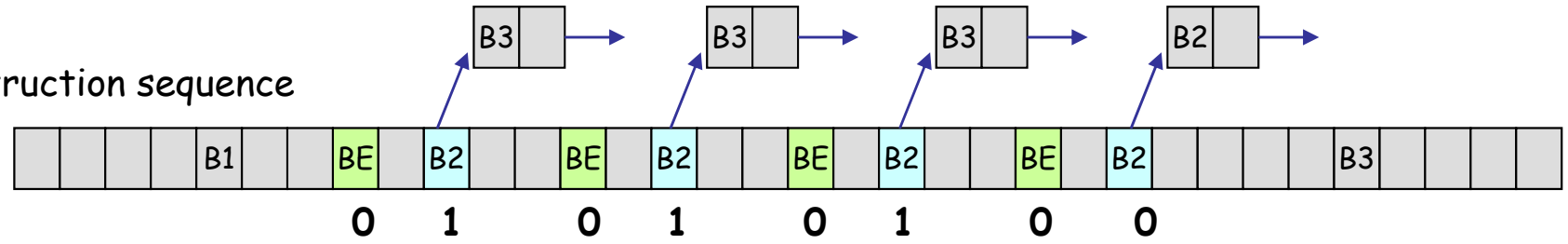
Sample program: vector add with two branches

```
#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]);
  }
}
```



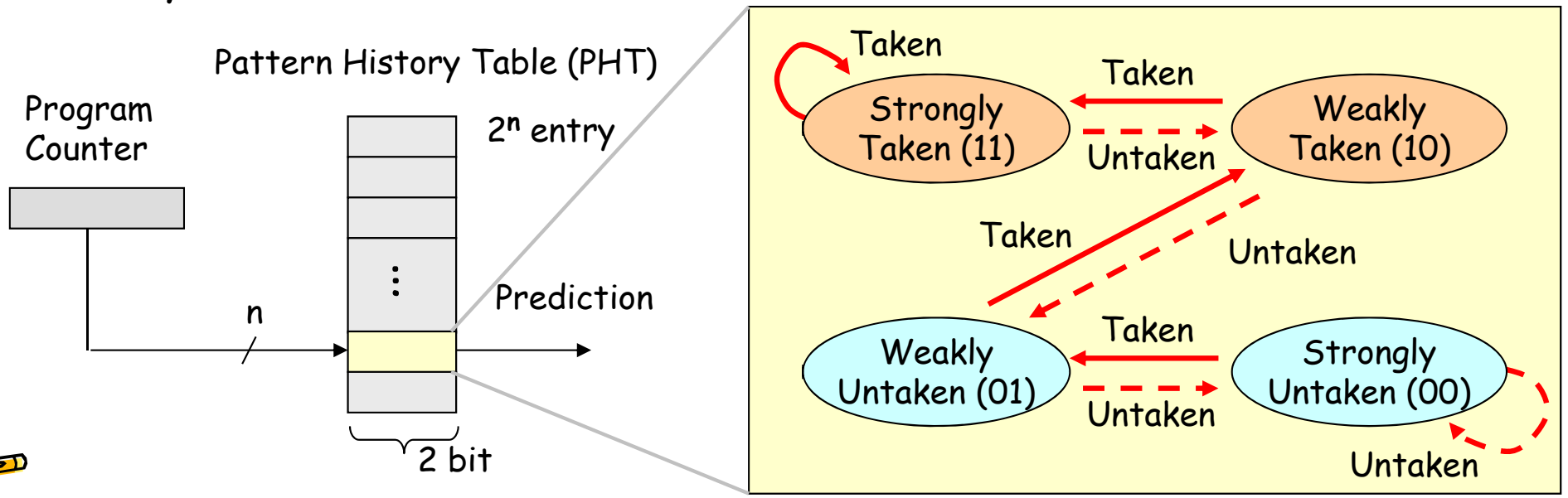
Control flow graph

Executed instruction sequence



Simple branch predictor: bimodal

- Program has many 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.



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 …

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

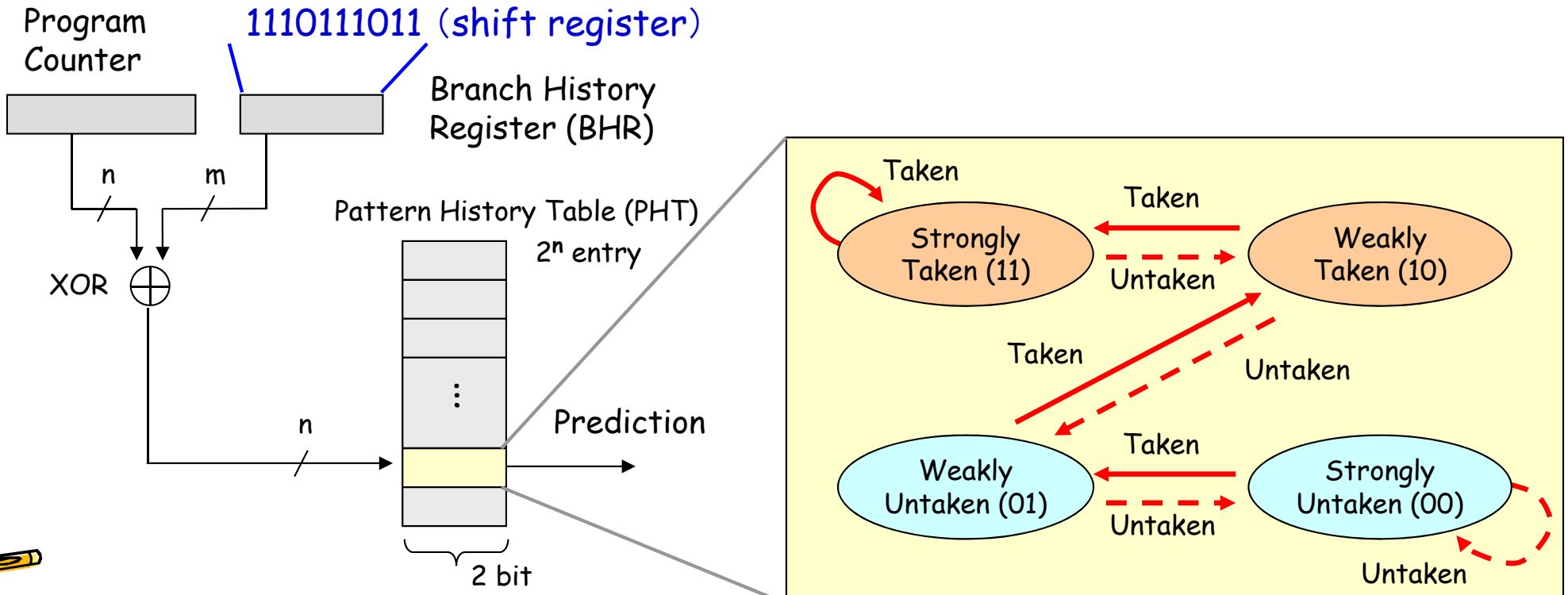
Recommended Reading

- 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."

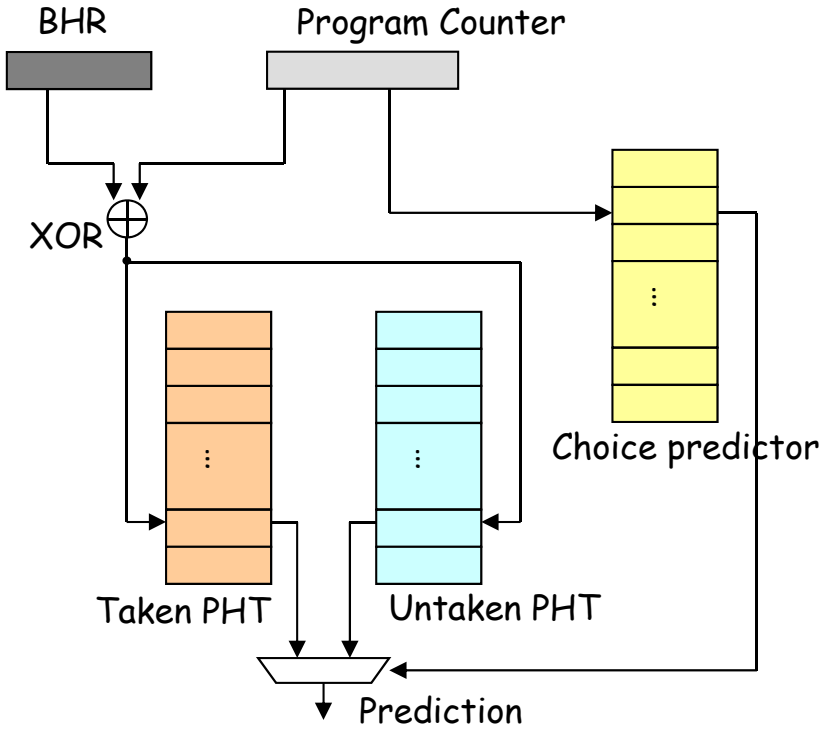
Gshare (TR-DEC 1993)

- 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.



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)

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

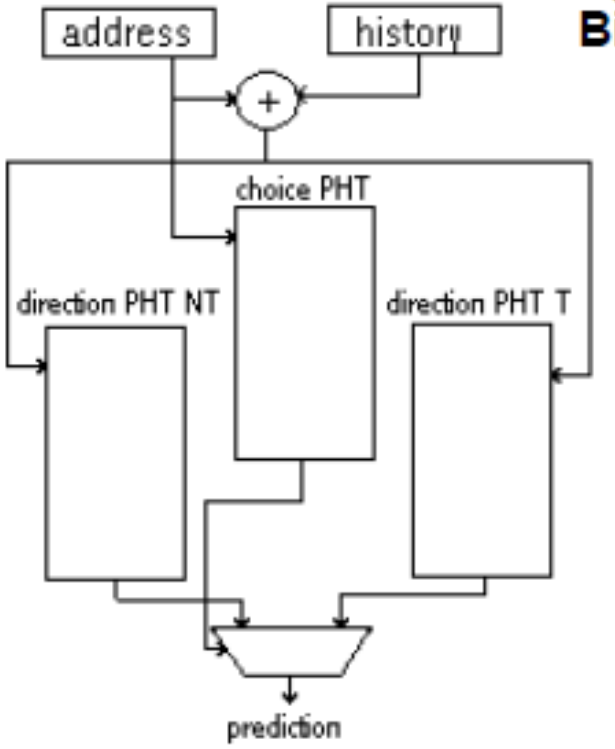


Figure 3. Bi-Mode

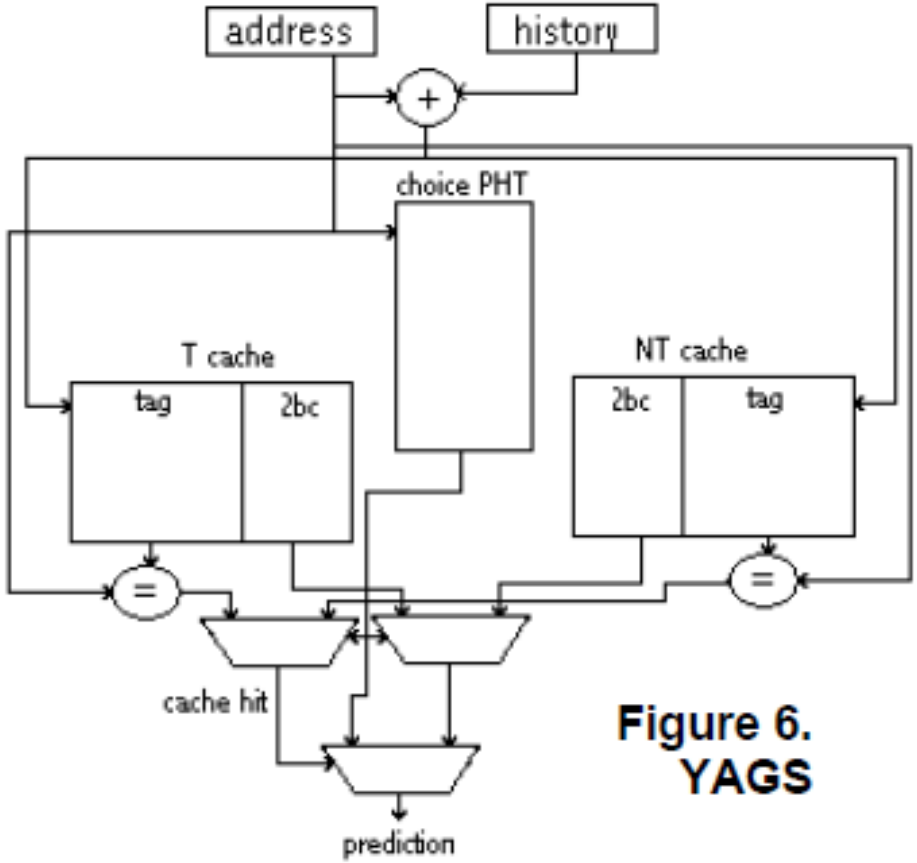
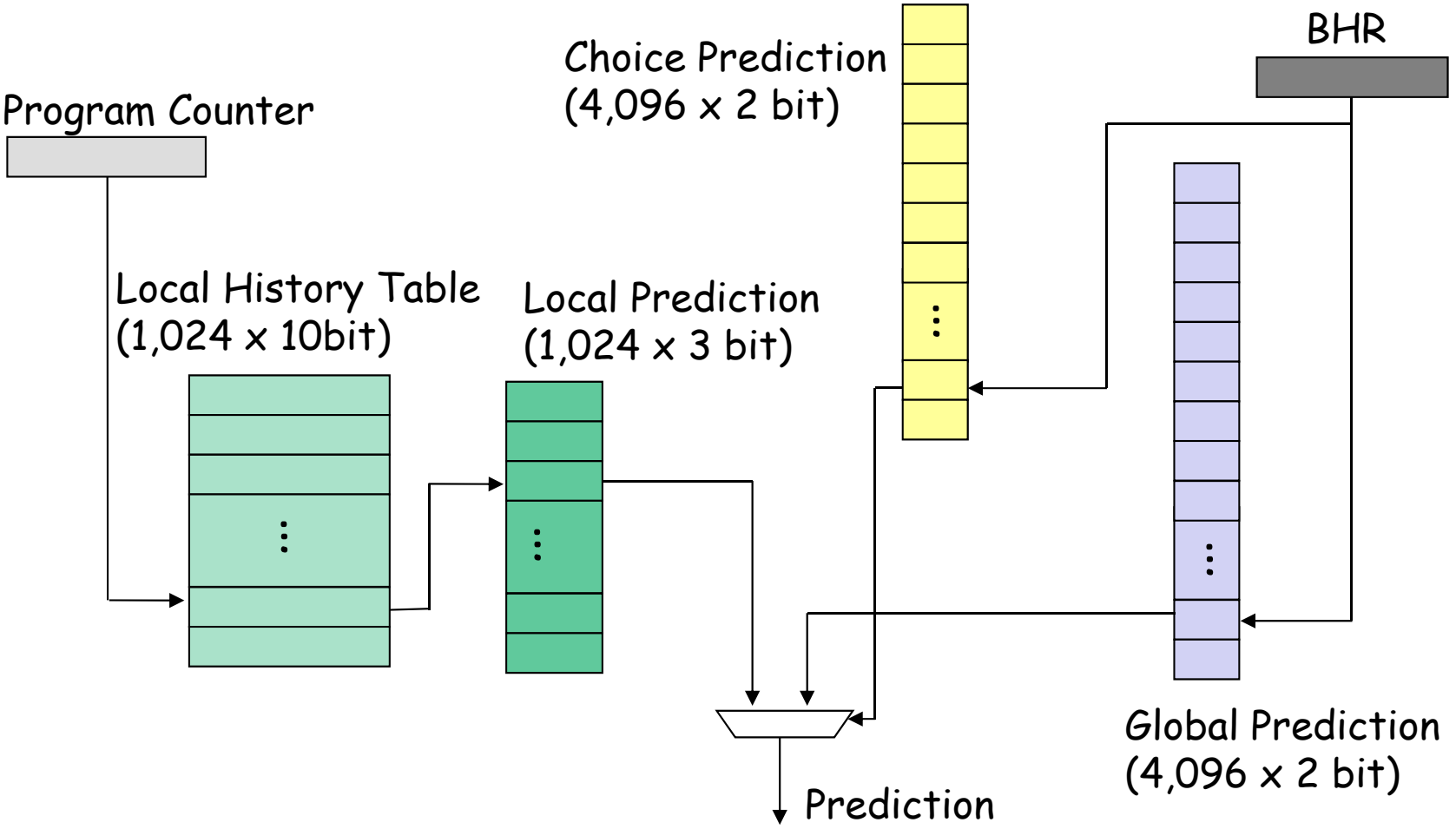


Figure 6. YAGS

From YAGS paper

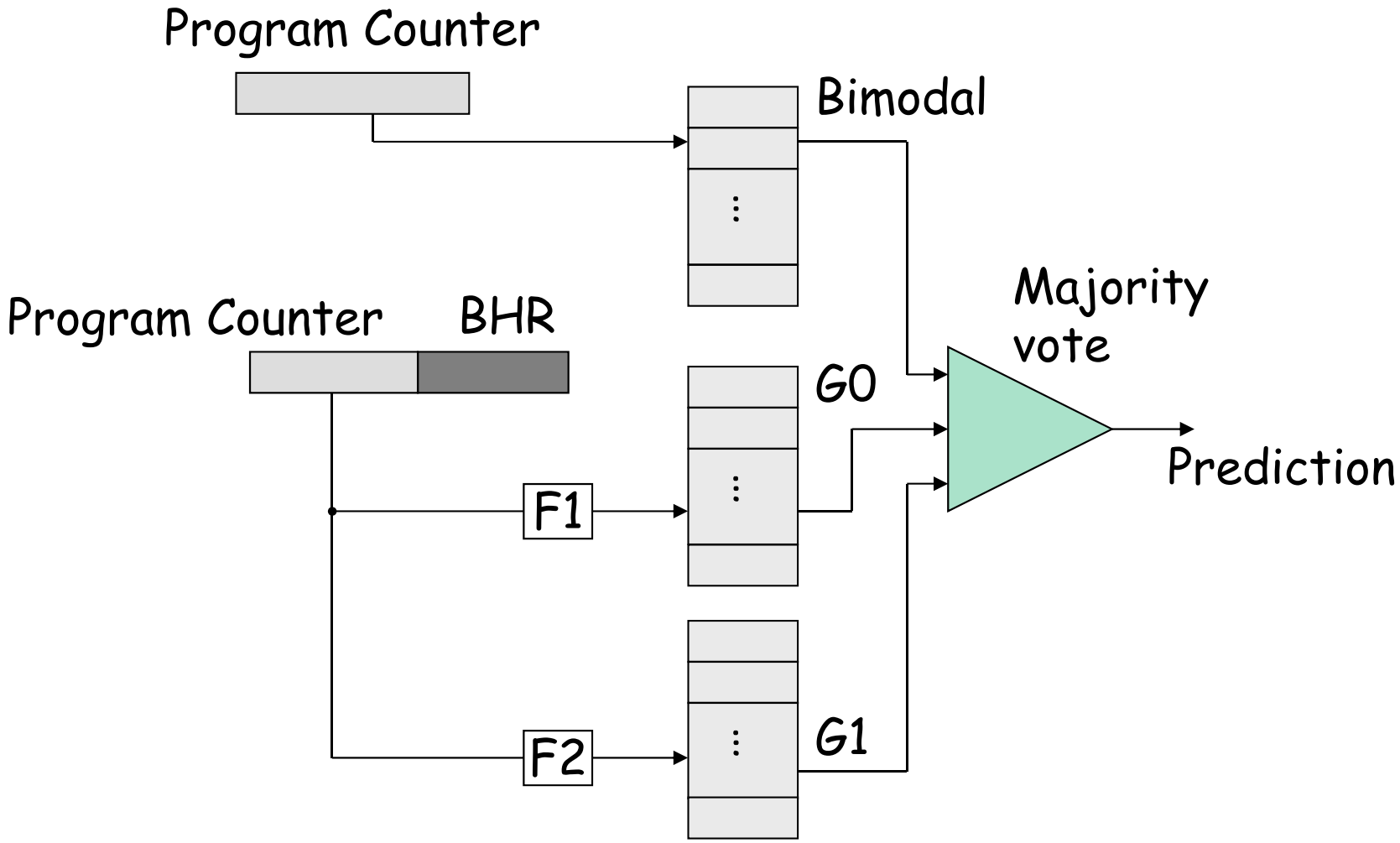
Alpha 21264's hybrid branch predictor

- A hybrid of local prediction and global prediction implemented in DEC Alpha 21264 which was the state-of-the art commercial processor.
- A choice predictor is used as a meta-predictor



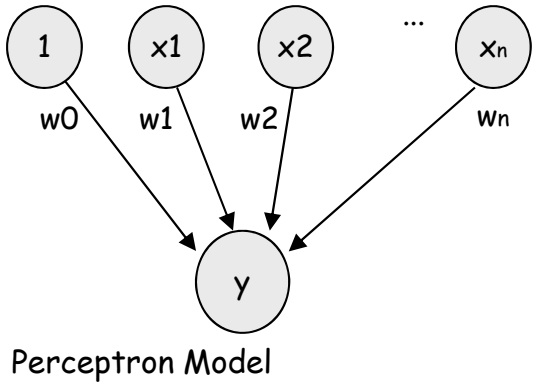
E-gskew (ISCA 1997)

- Using not a meta-predictor but a majority vote



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

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

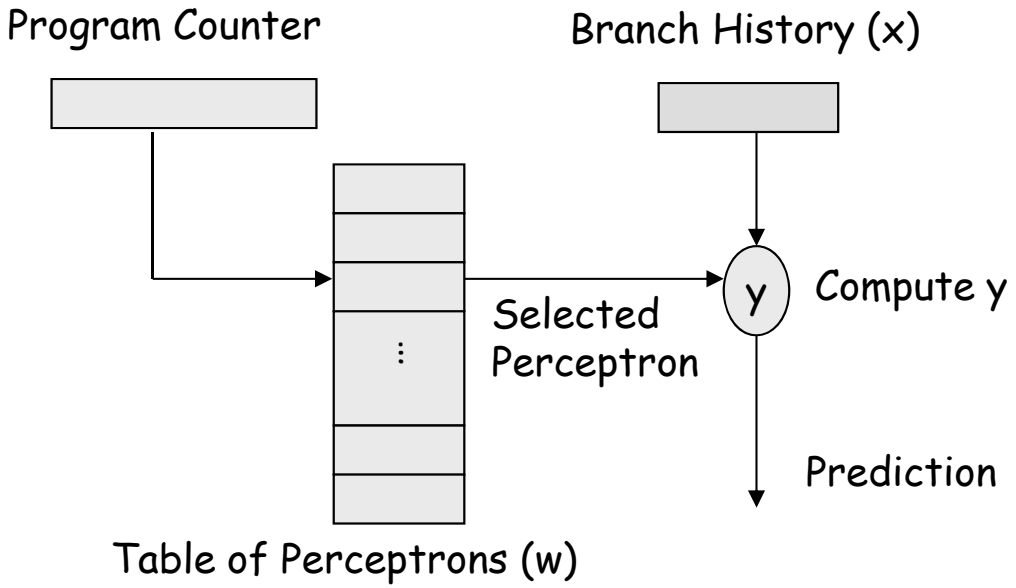
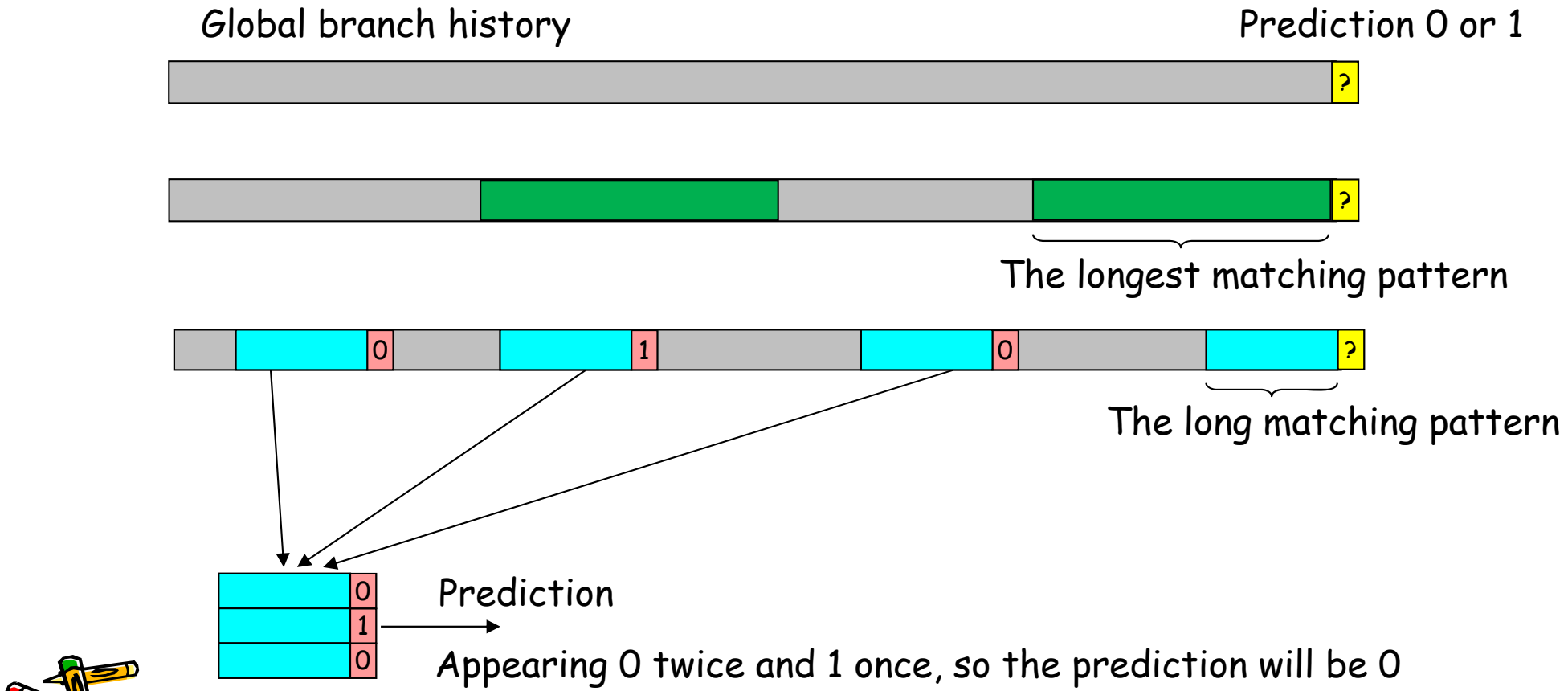


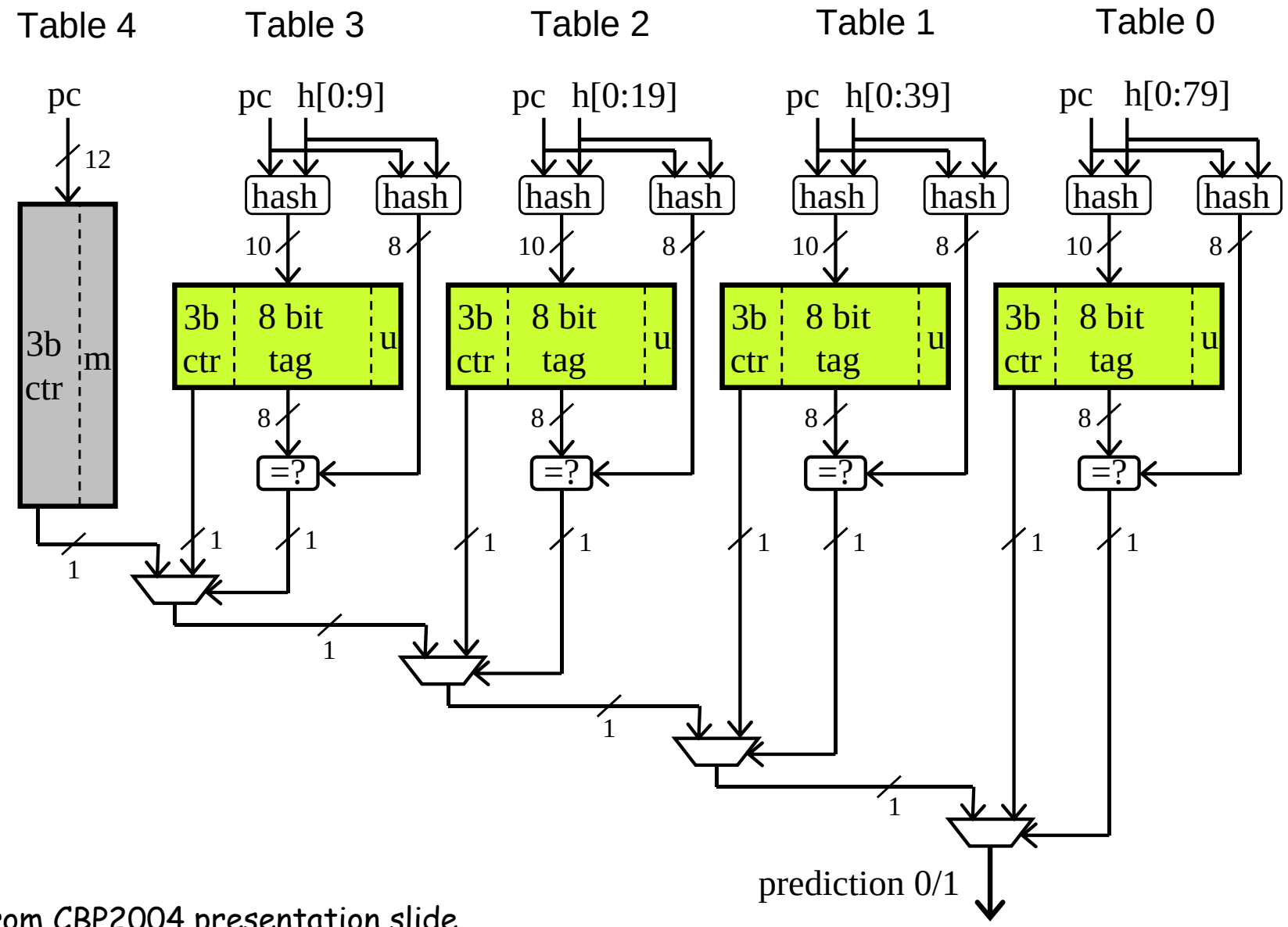
Table of Perceptrons (w)

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.

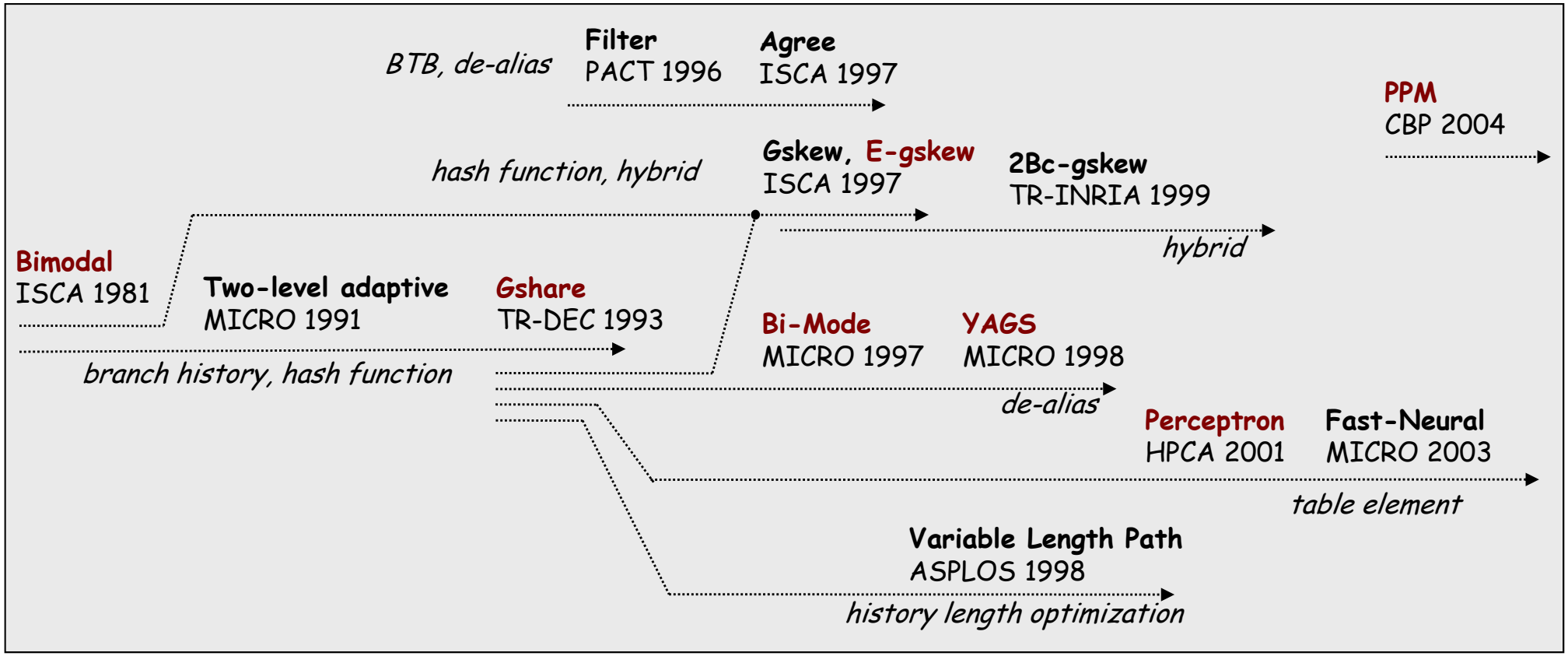


Partial Pattern Matching (CBP 2004)



From CBP2004 presentation slide

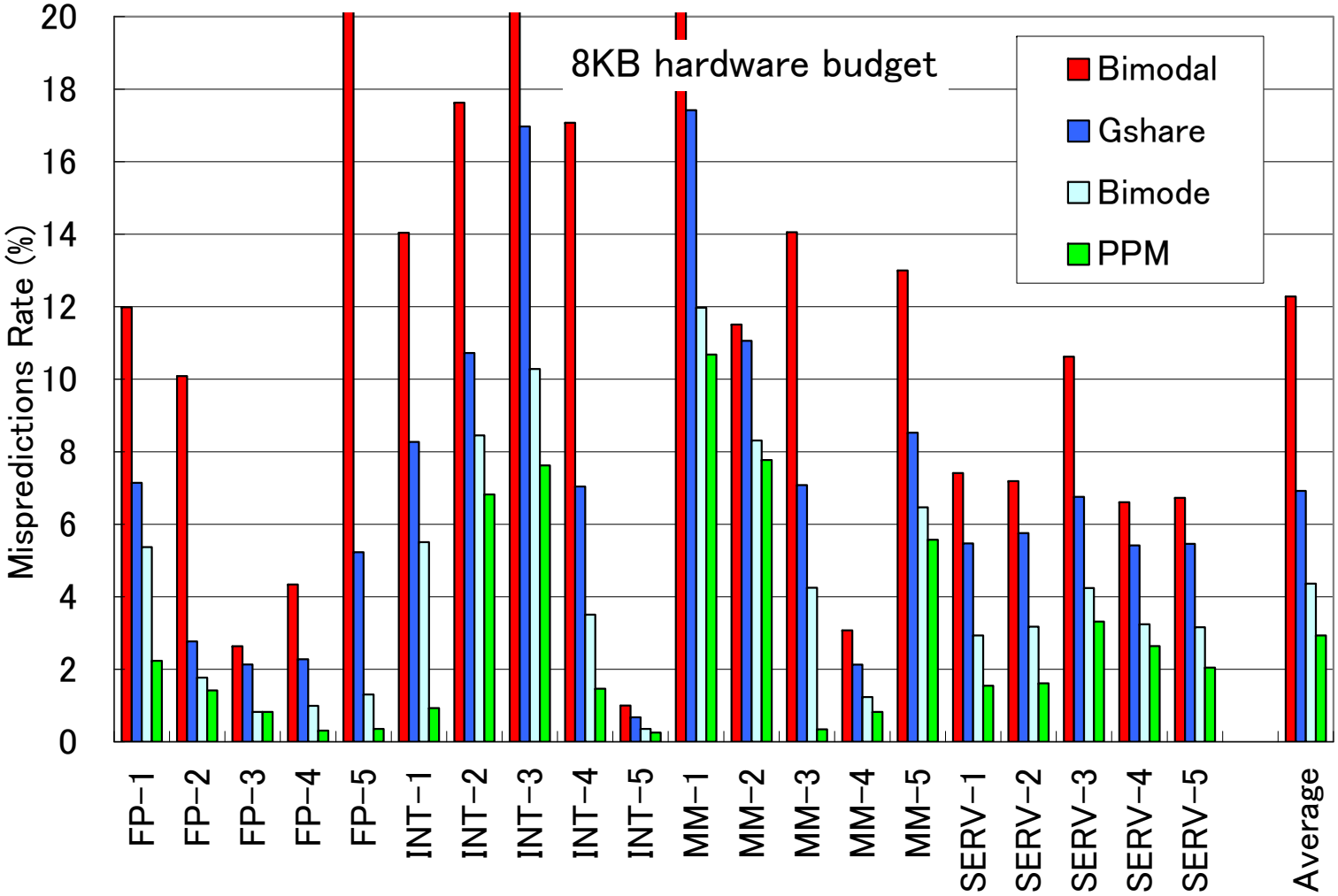
Some typical branch predictors until 2004



- ISCA (International Symposium on Computer Architecture)
- MICRO (International Symposium on Microarchitecture)
- PACT (International Conference on Parallel Architectures and Compilation Techniques)
- ASPLOS (International Conference on Architectural Support for Programming Languages and Operating Systems)

Prediction accuracy

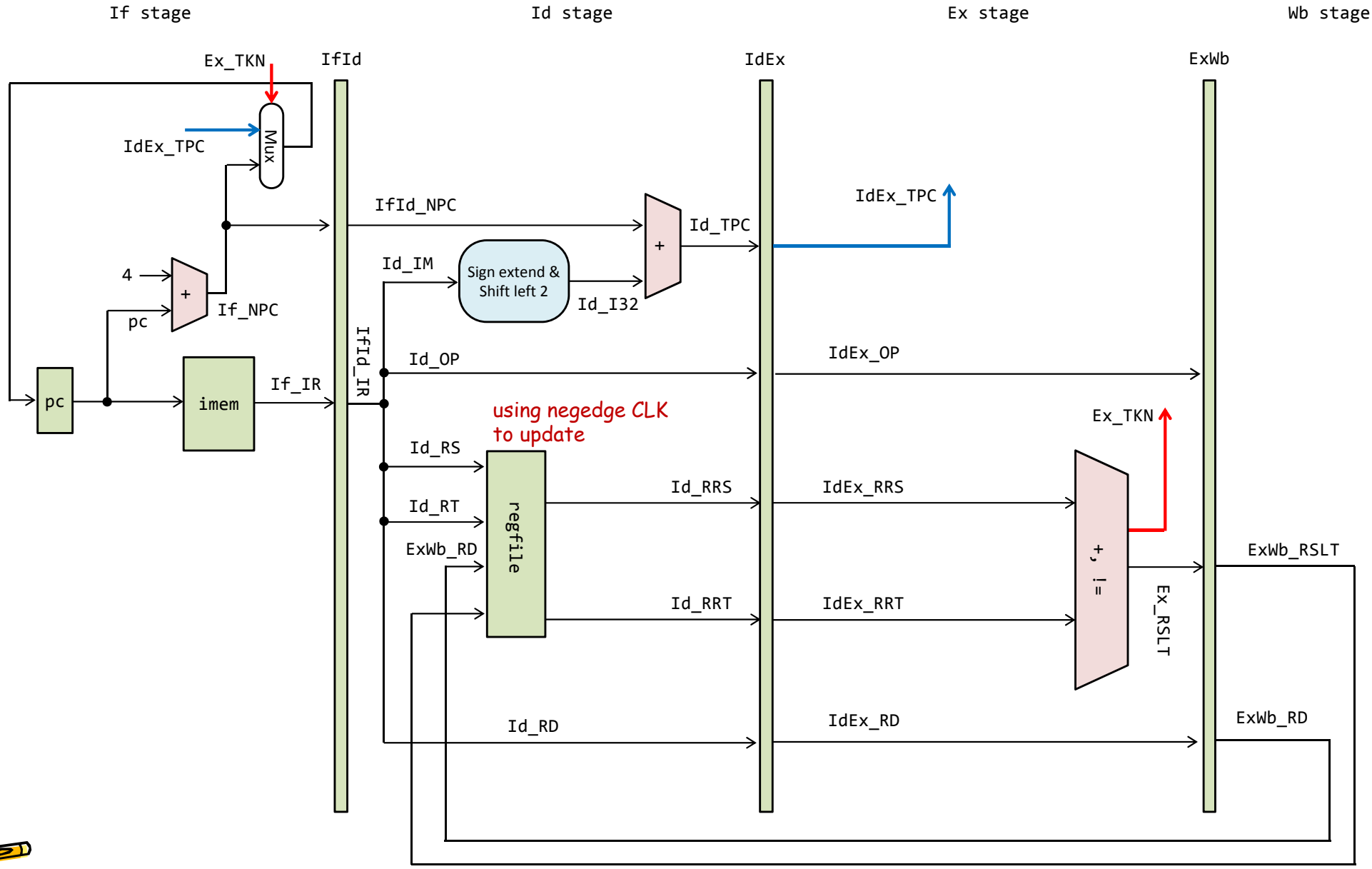
- The accuracy of 4KB Gshare is about 93%.
- The accuracy of 4KB PPM is about 97%.



Mid-term report

1. For details of the assignment, please visit the lecture support page. http://www.arch.cs.titech.ac.jp/lecture/ACA/
2. Submit your report printed on A4 paper at the beginning of the next lecture on January 7, 2019

Four stage pipelined processor supporting ADD and BNE, which does not adopt data forwarding (proc07.v)



Four stage pipelined processor supporting ADD and BNE, which does not adopt data forwarding (proc07.v)

```
initial begin /* initialize the instruction & data memory  & regfile */
  p.imem.mem[0] = {6'h0, 5'd0, 5'd0, 5'd0, 5'd0, 6'h20};  //     NOP
  p.imem.mem[1] = {6'h0, 5'd5, 5'd1, 5'd5, 5'd0, 6'h20};  // L1: add  $5, $5, $1
  p.imem.mem[2] = {6'h0, 5'd0, 5'd0, 5'd0, 5'd0, 6'h20};  //     NOP
  p.imem.mem[3] = {6'h0, 5'd0, 5'd0, 5'd0, 5'd0, 6'h20};  //     NOP
  p.imem.mem[4] = {6'h0, 5'd0, 5'd0, 5'd0, 5'd0, 6'h20};  //     NOP
  p.imem.mem[5] = {6'h5, 5'd4, 5'd5, 16'hfffb};           //     bne  $4, $5, L1
  p.imem.mem[6] = {6'h0, 5'd0, 5'd0, 5'd0, 5'd0, 6'h20};  //     NOP
  p.imem.mem[7] = {6'h0, 5'd0, 5'd0, 5'd0, 5'd0, 6'h20};  //     NOP
  p.imem.mem[8] = {6'h0, 5'd0, 5'd0, 5'd5, 5'd0, 6'h20};  //     add  $5, $0, $0
  p.imem.mem[9] = {6'h0, 5'd0, 5'd0, 5'd0, 5'd0, 6'h20};  //     NOP
  p.imem.mem[10]= {6'h0, 5'd0, 5'd0, 5'd0, 5'd0, 6'h20};  //     NOP
  p.imem.mem[11]= {6'h5, 5'd2, 5'd0, 16'hfff5};           //     bne  $2, $0, L1
  p.imem.mem[12]= {6'h0, 5'd0, 5'd0, 5'd0, 5'd0, 6'h20};  //     NOP
  p.imem.mem[13]= {6'h0, 5'd0, 5'd0, 5'd0, 5'd0, 6'h20};  //     NOP

  p.regfile.r[1] = 1;
  p.regfile.r[2] = 22;
  p.regfile.r[3] = 0;
  p.regfile.r[4] = 4;
  p.regfile.r[5] = 0;
end
```

```
while(1){
  for(int i=1; i!=4; i++){

  }
}
```

```
RRS, RRT, TKN:    4     1     1
RRS, RRT, TKN:    4     2     1
RRS, RRT, TKN:    4     3     1
RRS, RRT, TKN:    4     4     0
RRS, RRT, TKN:   22     0     1
RRS, RRT, TKN:    4     1     1
RRS, RRT, TKN:    4     2     1
RRS, RRT, TKN:    4     3     1
RRS, RRT, TKN:    4     4     0
RRS, RRT, TKN:   22     0     1
RRS, RRT, TKN:    4     1     1
RRS, RRT, TKN:    4     2     1
RRS, RRT, TKN:    4     3     1
RRS, RRT, TKN:    4     4     0
RRS, RRT, TKN:   22     0     1
RRS, RRT, TKN:    4     1     1
RRS, RRT, TKN:    4     2     1
RRS, RRT, TKN:    4     3     1
RRS, RRT, TKN:    4     4     0
RRS, RRT, TKN:   22     0     1
RRS, RRT, TKN:    4     1     1
RRS, RRT, TKN:    4     2     1
RRS, RRT, TKN:    4     3     1
RRS, RRT, TKN:    4     4     0
RRS, RRT, TKN:   22     0     1
RRS, RRT, TKN:    4     1     1
RRS, RRT, TKN:    4     2     1
RRS, RRT, TKN:    4     3     1
RRS, RRT, TKN:    4     4     0
RRS, RRT, TKN:   22     0     1
RRS, RRT, TKN:    4     1     1
RRS, RRT, TKN:    4     2     1
RRS, RRT, TKN:    4     3     1
RRS, RRT, TKN:    4     4     0
RRS, RRT, TKN:   22     0     1
RRS, RRT, TKN:    4     1     1
RRS, RRT, TKN:    4     2     1
RRS, RRT, TKN:    4     3     1
RRS, RRT, TKN:    4     4     0
RRS, RRT, TKN:   22     0     1
RRS, RRT, TKN:    4     1     1
```

Exercise: how to update PHT and BHR of Gshare