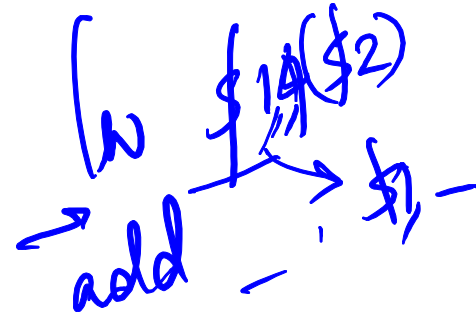


Control Hazard

Outline

- Pipeline, Pipelined datapath
- Dependences, Hazards
 - Structural, Data - Stalling, Forwarding
- Control Hazards
- Branch prediction



Control Hazard

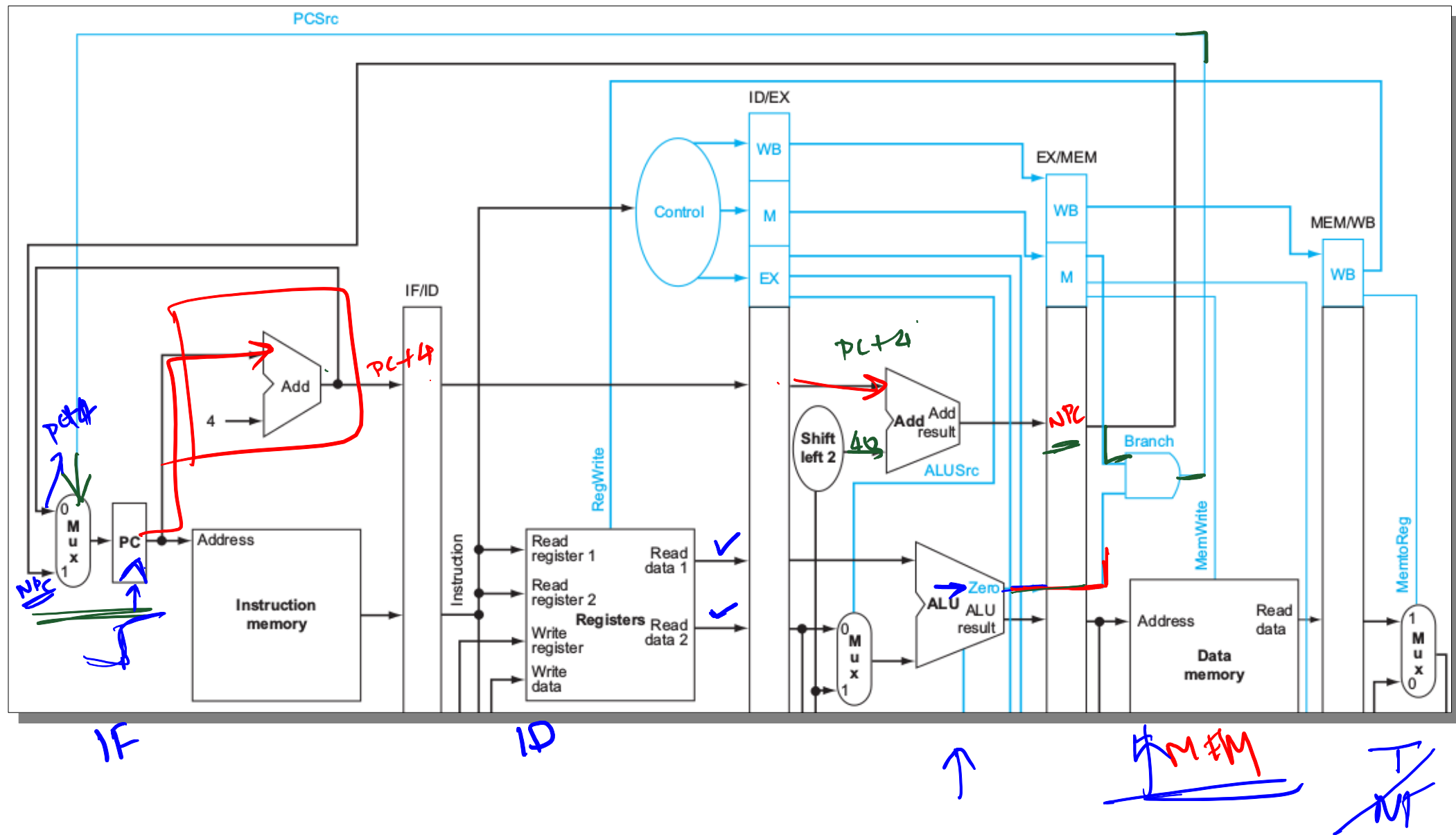
- Arise from the pipelining of branches and
other instructions that change the PC

Control Hazard

- Arise from the pipelining of branches and other instructions that change the PC
- Also called Branch Hazards

beg \$1, \$2, 10
 $NPC = PC + 4 + (10 \times 4)$
 $NPC = PC + 44$
 $NPC = PC + 4 + NT \Rightarrow NPC = PC + 44$

Branch Evaluation



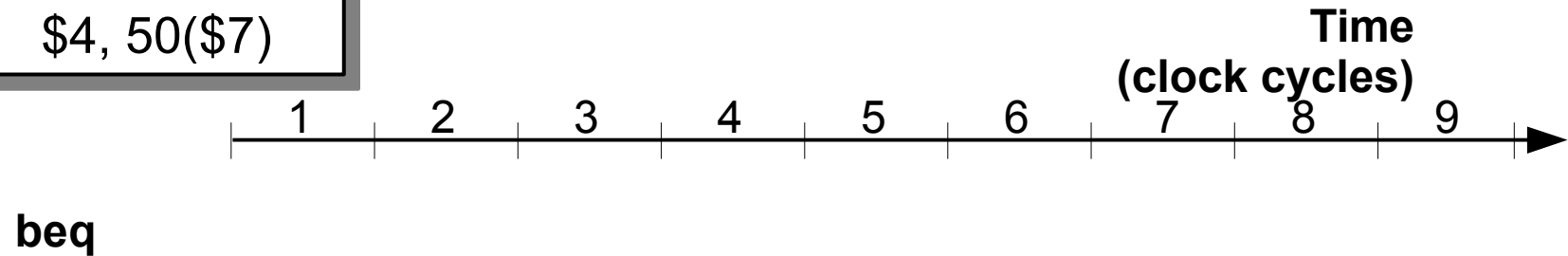
Control Hazard

beq \$1, \$3, 28
and \$12, \$2, \$5
or \$13, \$6, \$2
add \$14, \$2, \$2
...
lw \$4, 50(\$7)

184

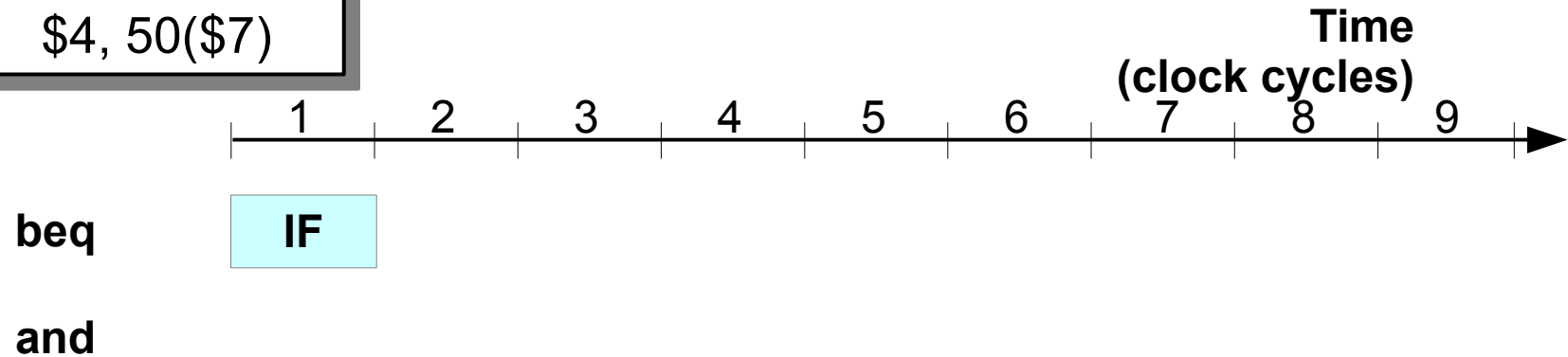
Control Hazard

beq	\$1, \$3, 28
and	\$12, \$2, \$5
or	\$13, \$6, \$2
add	\$14, \$2, \$2
...	
lw	\$4, 50(\$7)



Control Hazard

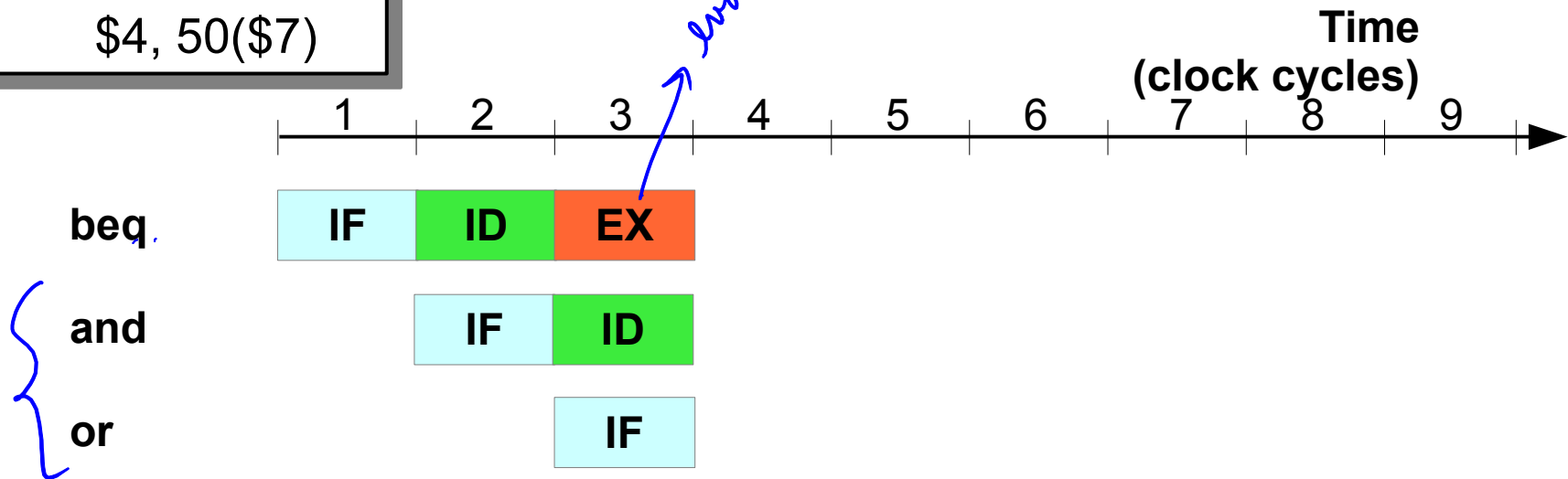
beq	\$1, \$3, 28
and	\$12, \$2, \$5
or	\$13, \$6, \$2
add	\$14, \$2, \$2
...	
lw	\$4, 50(\$7)



Control Hazard

beq	\$1, \$3, 28
and	\$12, \$2, \$5
or	\$13, \$6, \$2
add	\$14, \$2, \$2
...	
lw	\$4, 50(\$7)

~~$\$1 == \3~~
evaluation



Control Hazard

beq \$1, \$3, 28
and \$12, \$2, \$5
or \$13, \$6, \$2
add \$14, \$2, \$2
...
lw \$4, 50(\$7)

Branch Taken

Time
(clock cycles)

beq

IF

ID

EX

and

IF

ID

or

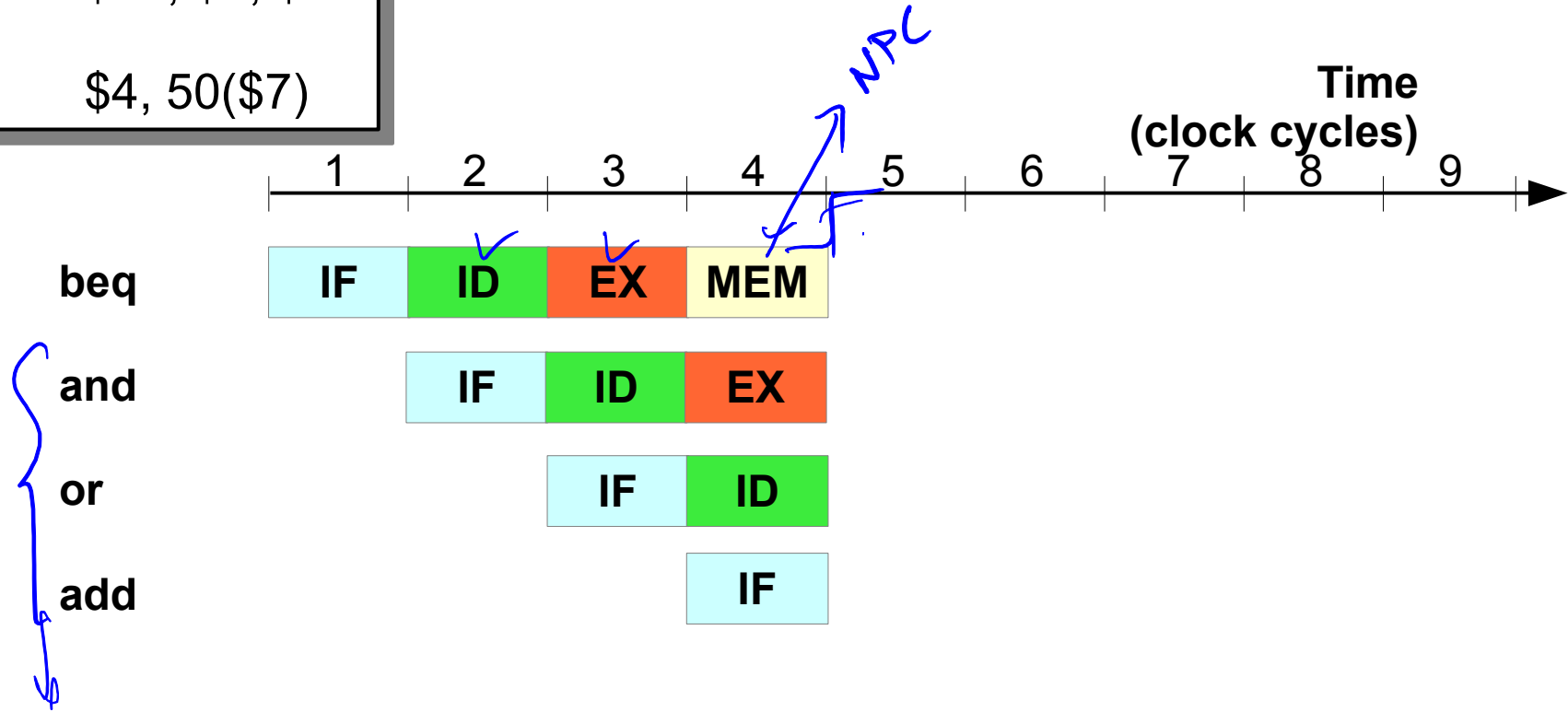
IF

TRUE

Control Hazard

beq	\$1, \$3, 28
and	\$12, \$2, \$5
or	\$13, \$6, \$2
add	\$14, \$2, \$2
...	
lw	\$4, 50(\$7)

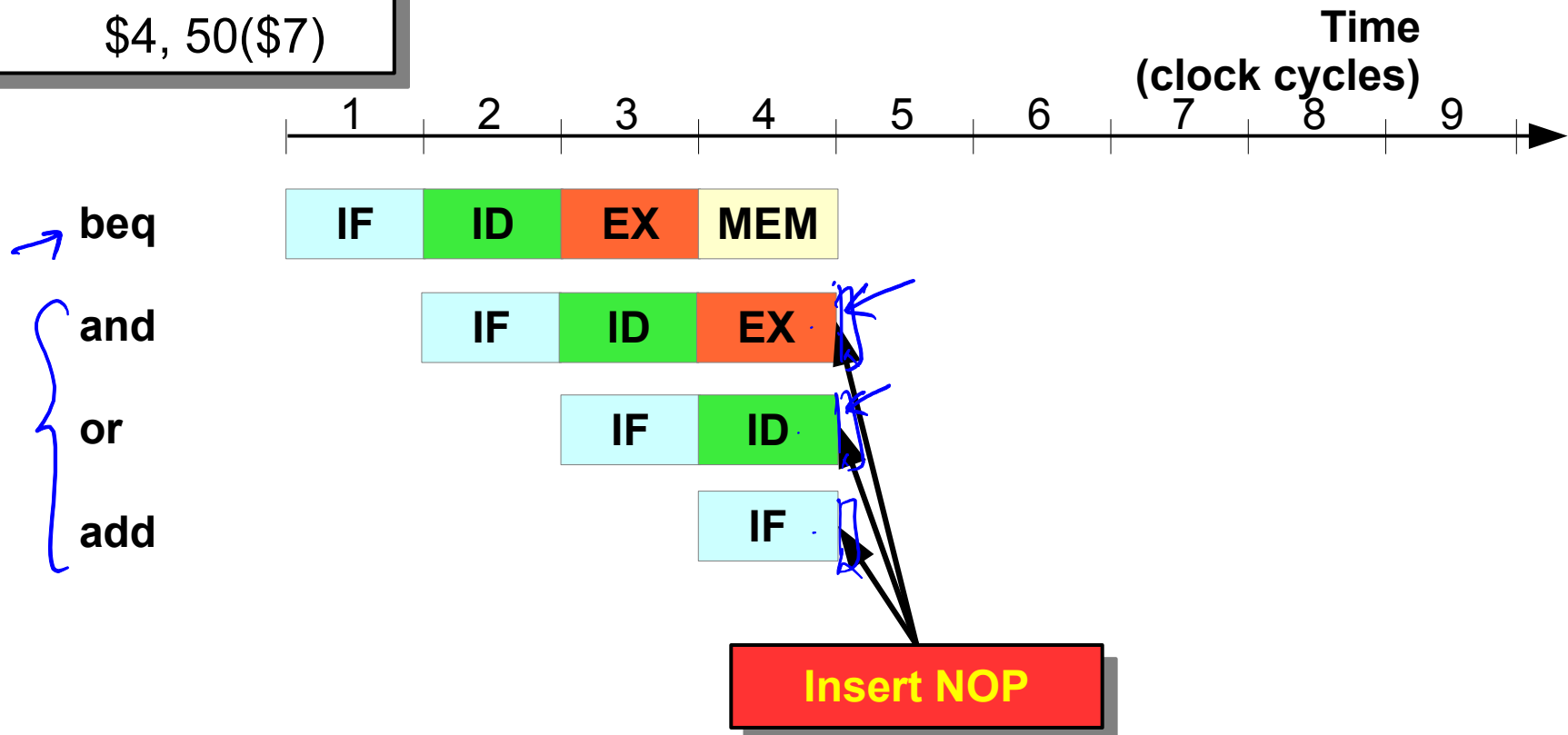
NPC Updates



Control Hazard

beq	\$1, \$3, 28
and	\$12, \$2, \$5
or	\$13, \$6, \$2
add	\$14, \$2, \$2
...	
lw	\$4, 50(\$7)

NPC Updates



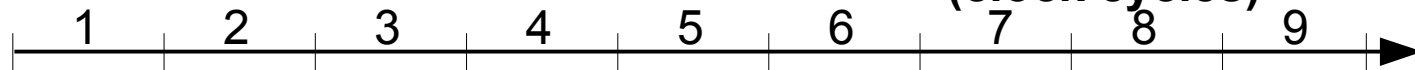
Control Hazard

basic block

beq	\$1, \$3, 28
and	\$12, \$2, \$5
or	\$13, \$6, \$2
add	\$14, \$2, \$2
...	
lw	\$4, 50(\$7)

Branch target
enters the pipeline

Time
(clock cycles)



beq



and



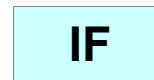
or



add



lw

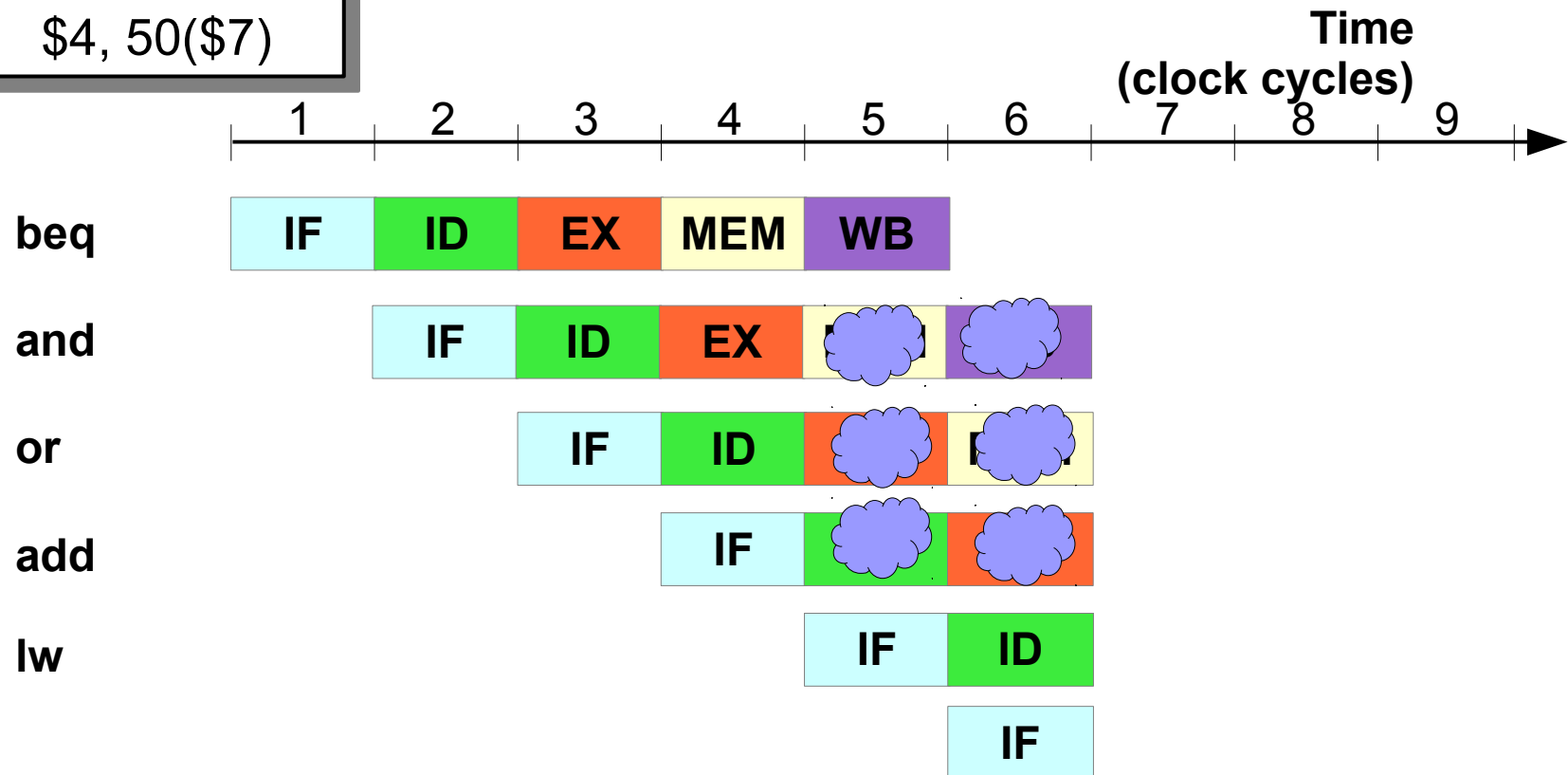


NPC
Target:

85.1 → 1.0
15.2 → 4.0

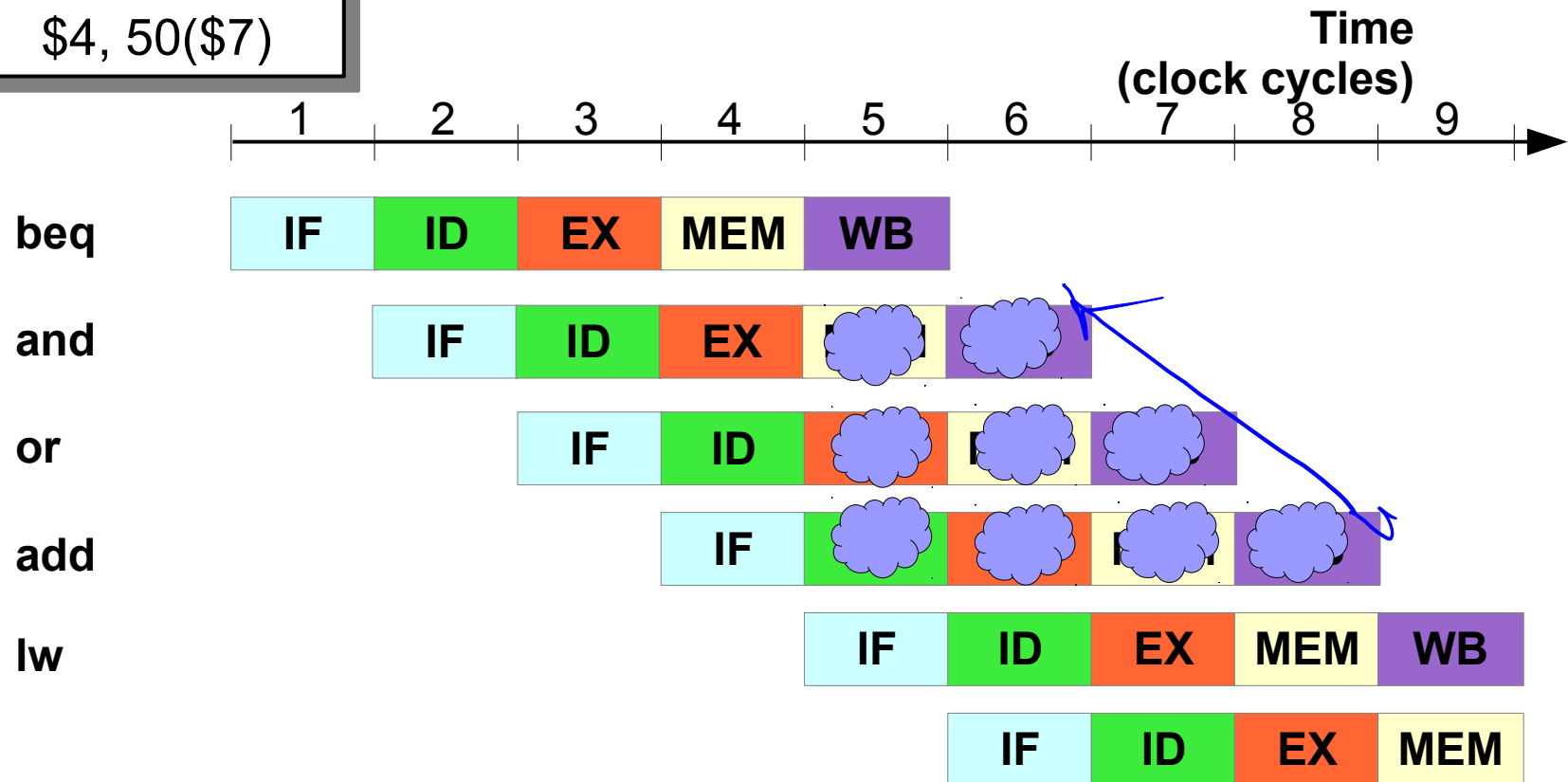
Control Hazard

beq	\$1, \$3, 28
and	\$12, \$2, \$5
or	\$13, \$6, \$2
add	\$14, \$2, \$2
...	
lw	\$4, 50(\$7)



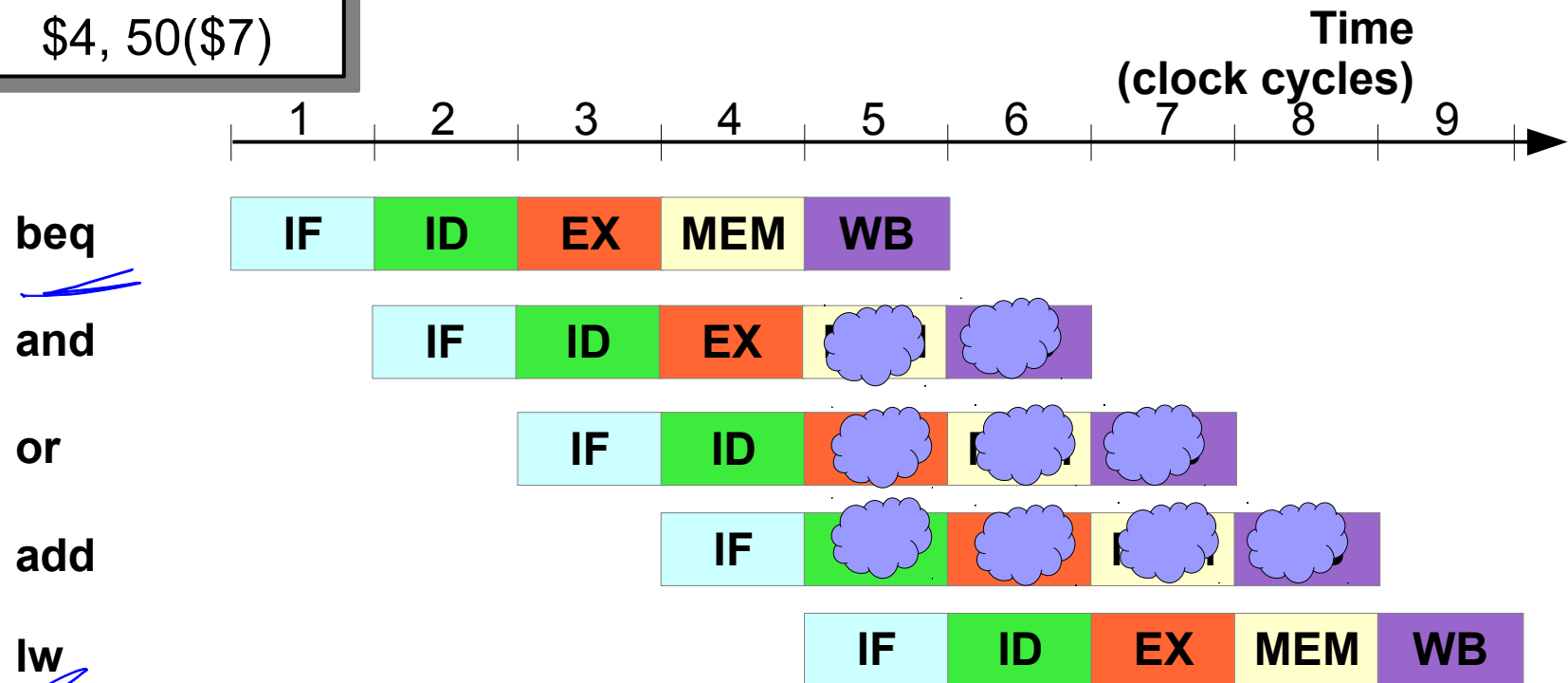
Control Hazard

beq	\$1, \$3, 28
and	\$12, \$2, \$5
or	\$13, \$6, \$2
add	\$14, \$2, \$2
...	
lw	\$4, 50(\$7)



Control Hazard

beq	\$1, \$3, 28
and	\$12, \$2, \$5
or	\$13, \$6, \$2
add	\$14, \$2, \$2
...	
lw	\$4, 50(\$7)

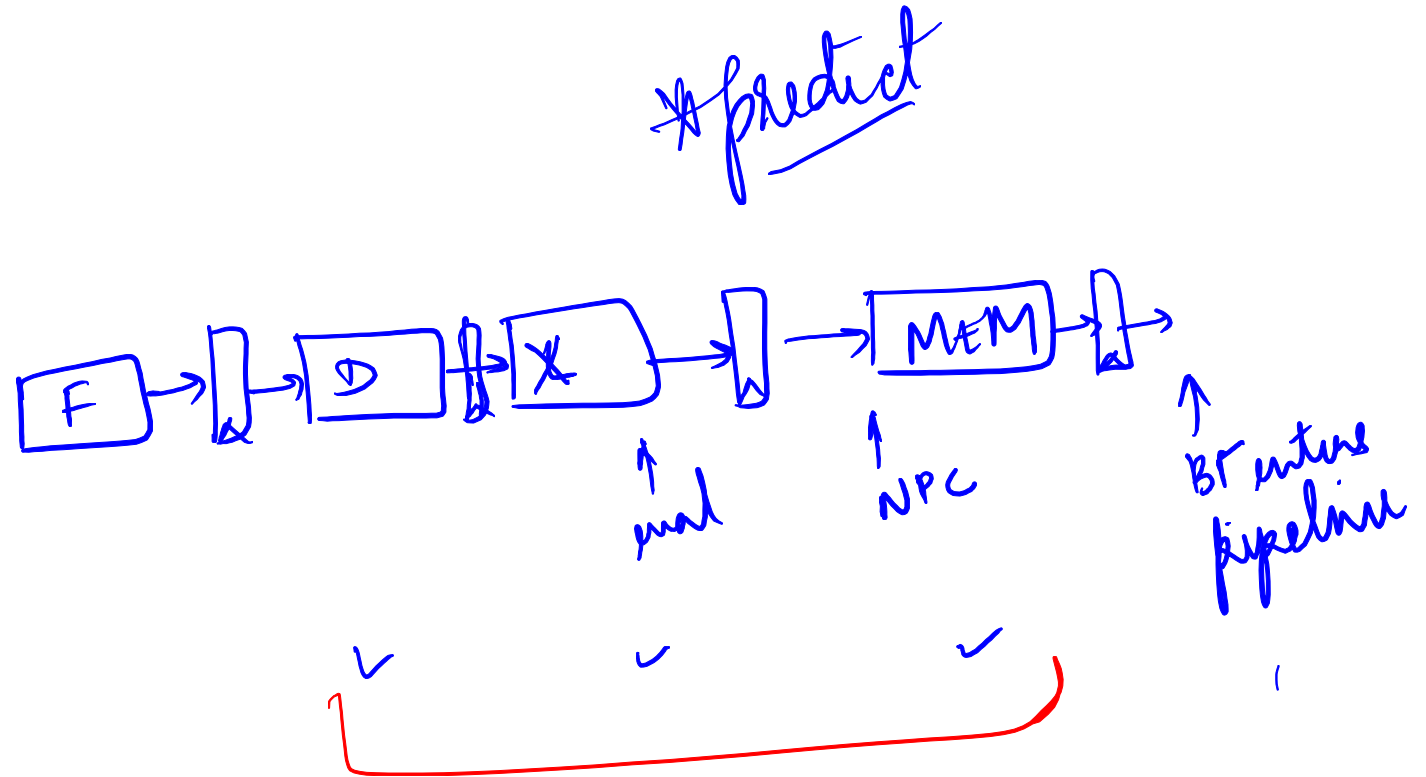


Branch Penalty = 3 cycles

Control Hazard

- Branch evaluation occurs in EX stage.
- PC updates in MEM stage.
- Branch target loads in the next cycle
- The delay in determining the proper instruction to fetch is called a Control Hazard or Branch Hazard.

Reducing Pipeline Branch Penalties



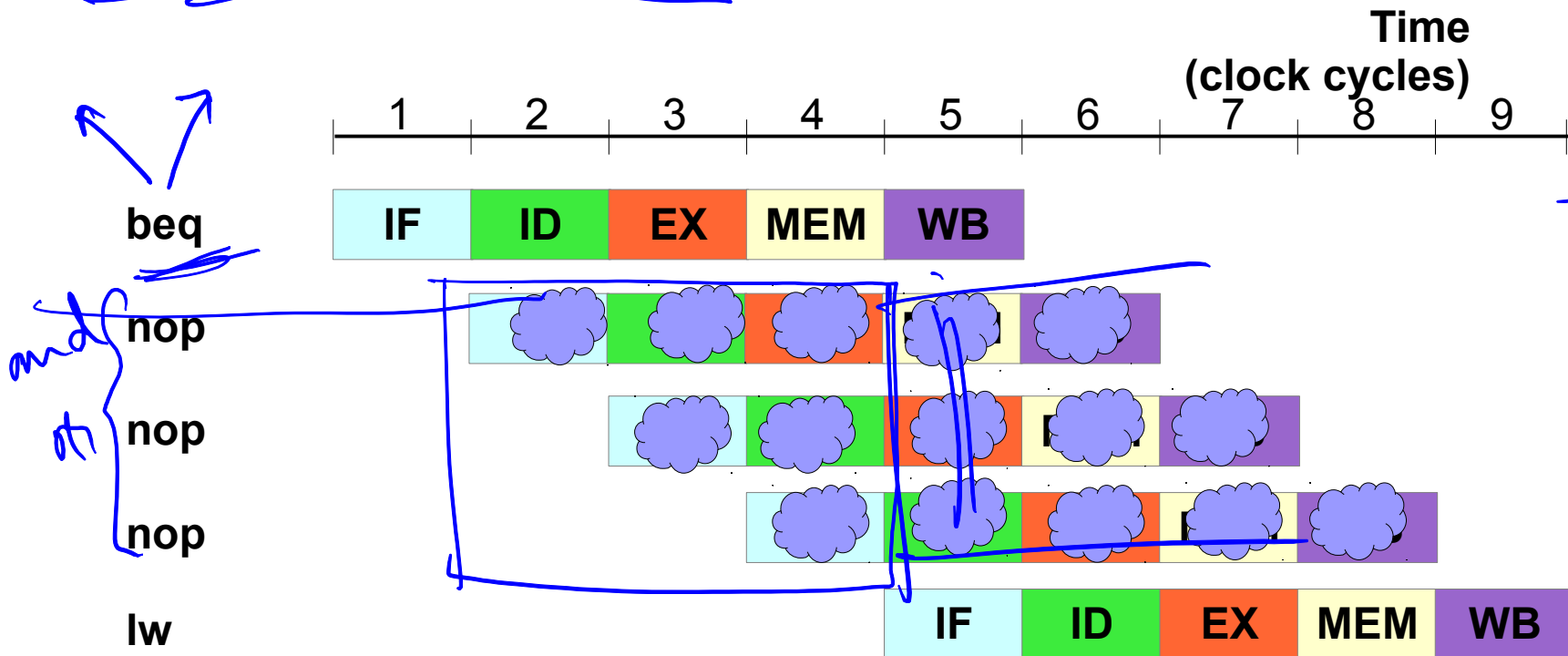
Reducing Pipeline Branch Penalties

- Freeze the pipeline

Reducing Pipeline Branch Penalties

- Freeze the pipeline

proceed with
shut down pipeline → energy
 $E = P \times t$



Reducing Branch Delay

* branch penalty = 3 cycles

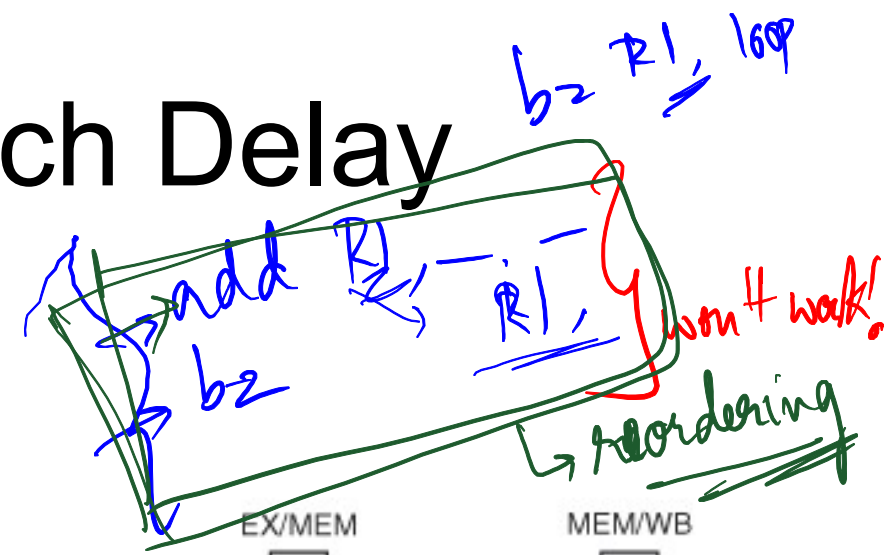
b2
b2v

Reducing Branch Delay

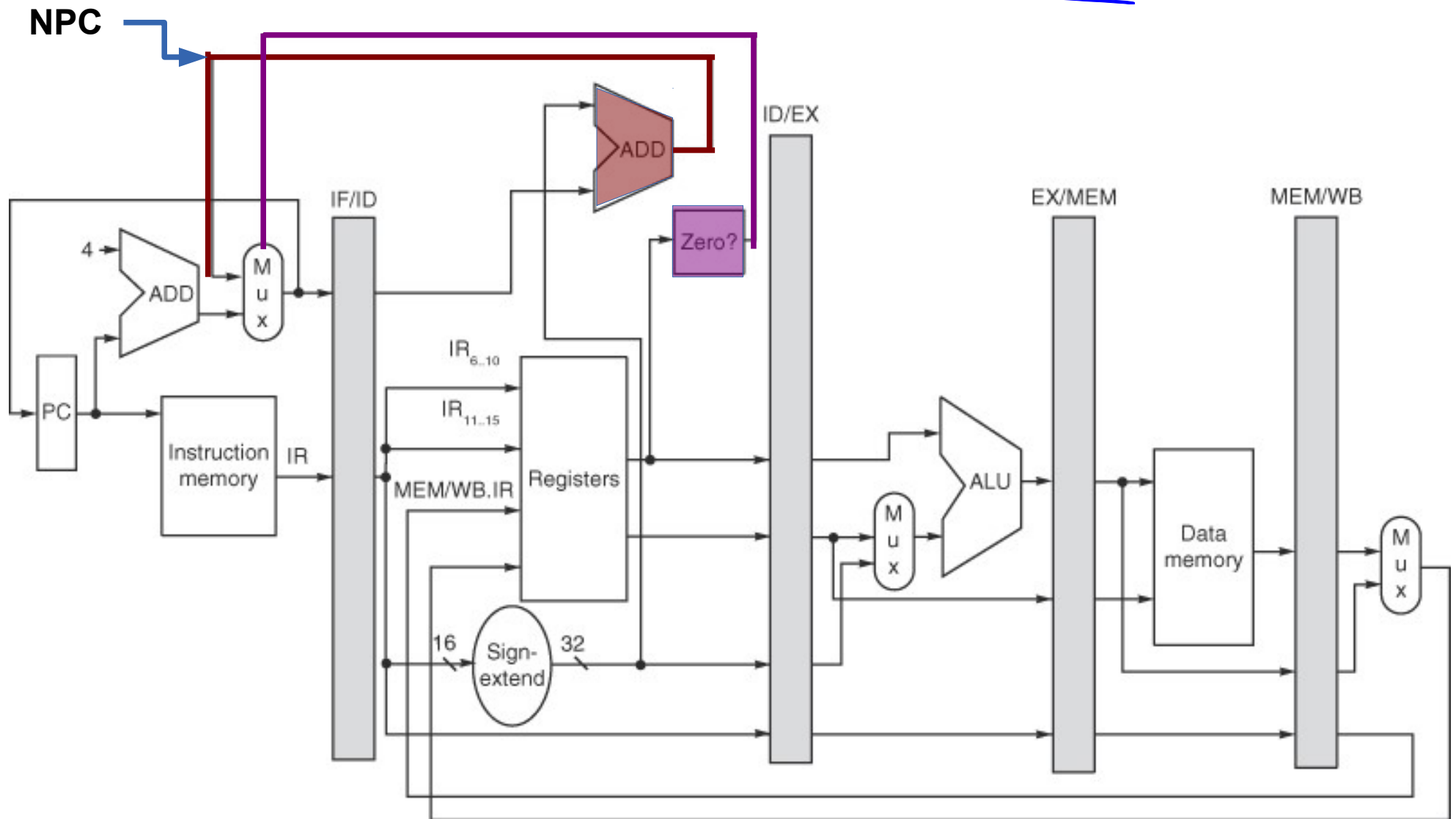
- Most branches rely on simple tests (equality or sign)
- Move the branch decision up
-

Reducing Branch Delay

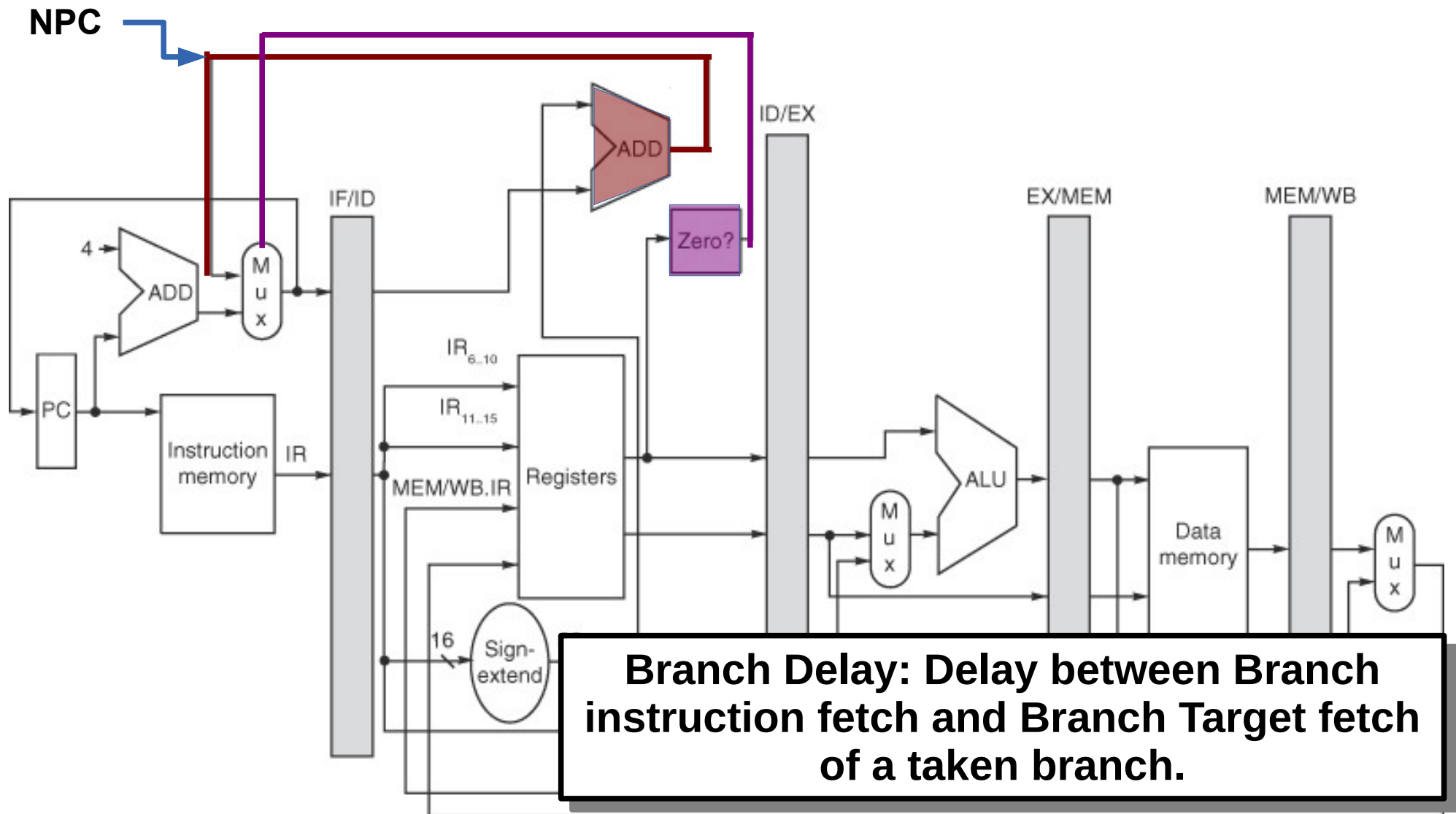
- Most branches rely on simple tests (equality or sign)
- Move the branch decision up
 - Compute NPC earlier
 - Evaluate branch at the earliest



Reducing Branch Delay

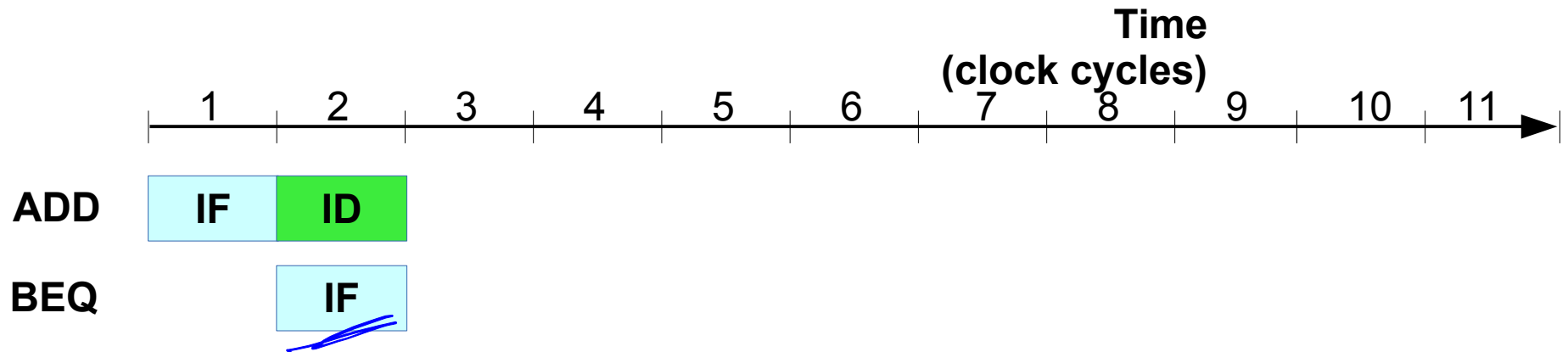


Reducing Branch Delay



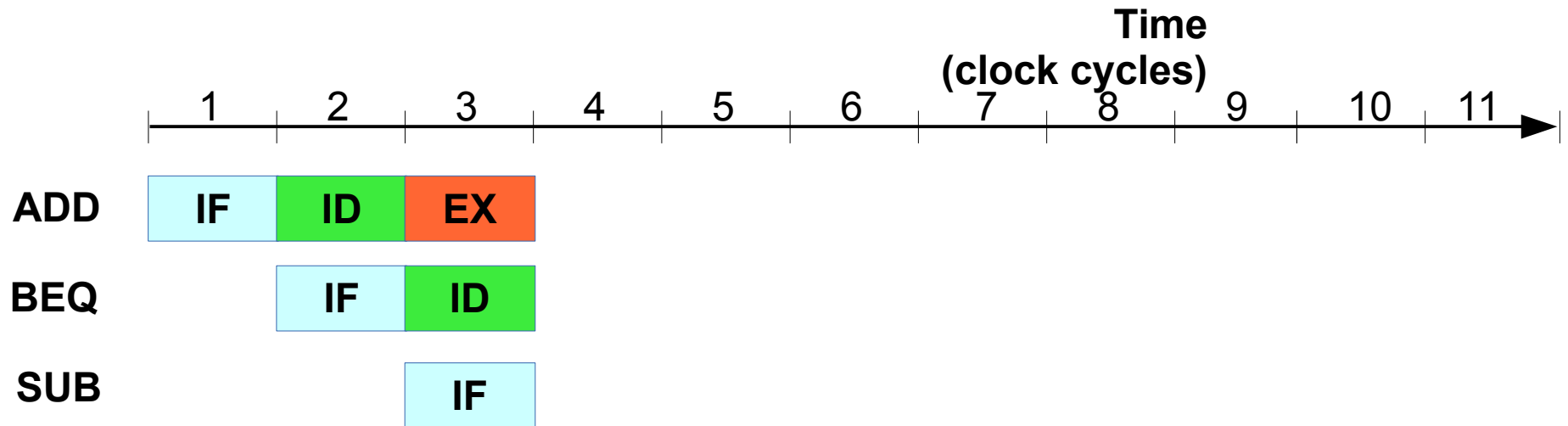
Branch Delay

```
ADD    R2, R3,R4
BEQ    R2, R4, loop
SUB    R5, R5,R4
...
```



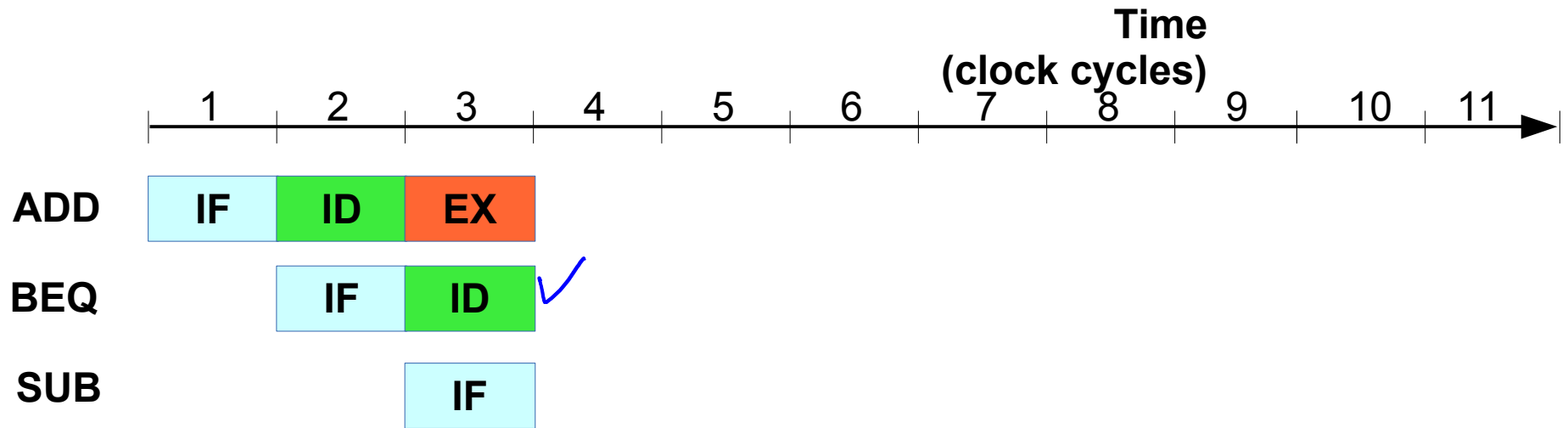
Branch Delay

```
ADD    R2, R3,R4
BEQ    R2, R4, loop
SUB    R5, R5,R4
...
```



Branch Delay

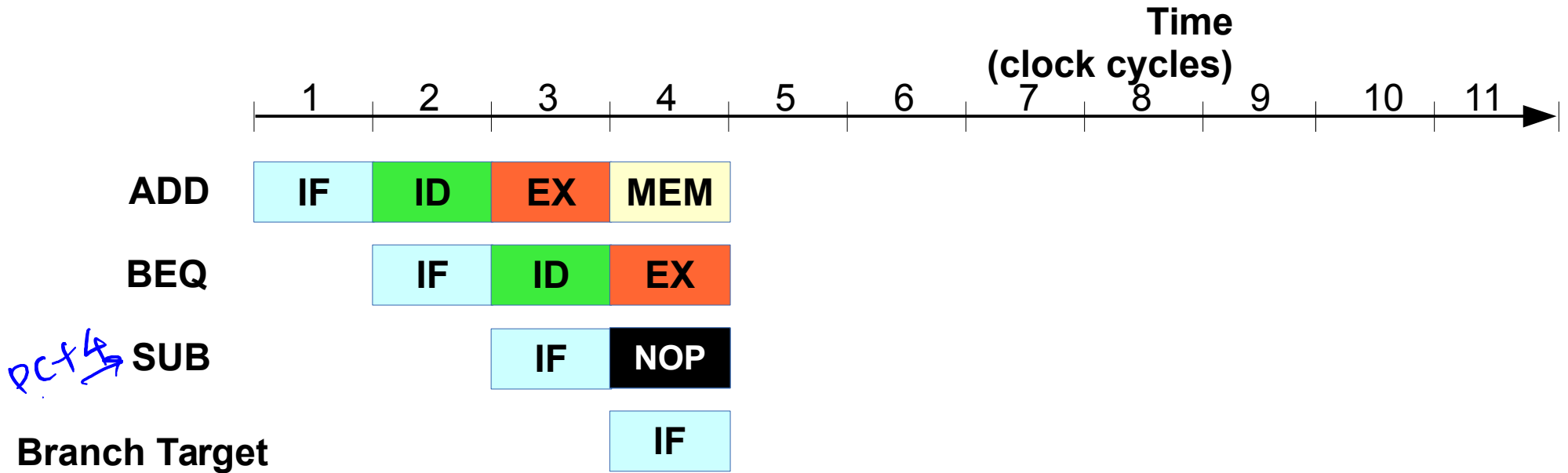
ADD R2, R3, R4
BEQ R2, R4, loop
SUB R5, R5, R4
...



PC = PC + Offset

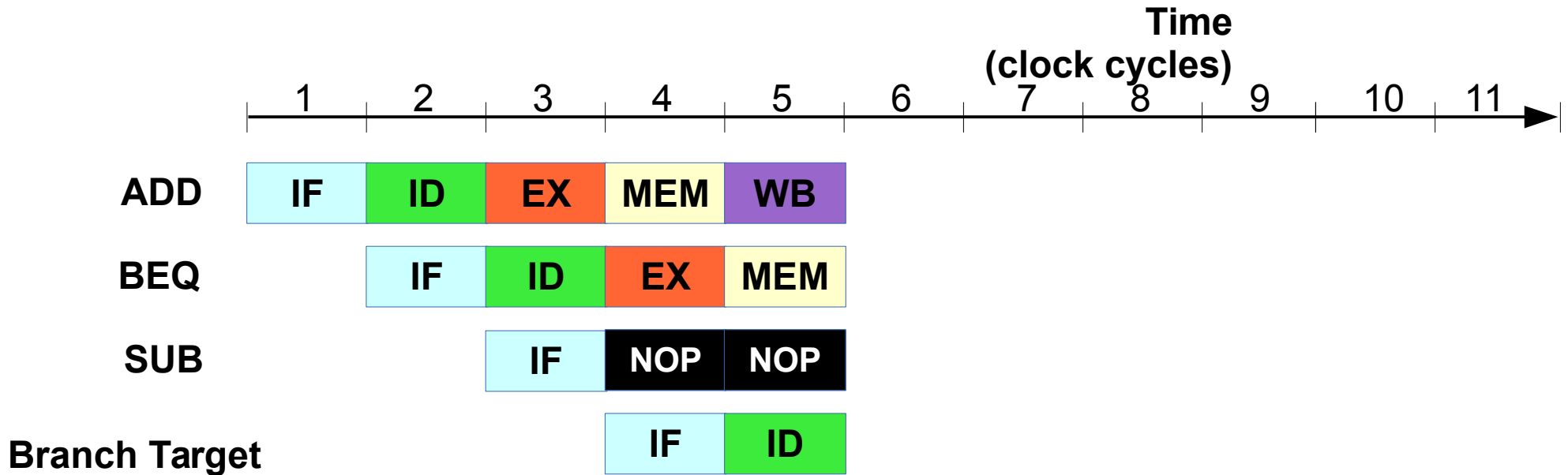
Branch Delay

```
ADD    R2, R3,R4
BEQ    R2, R4, loop
SUB    R5, R5,R4
...
```



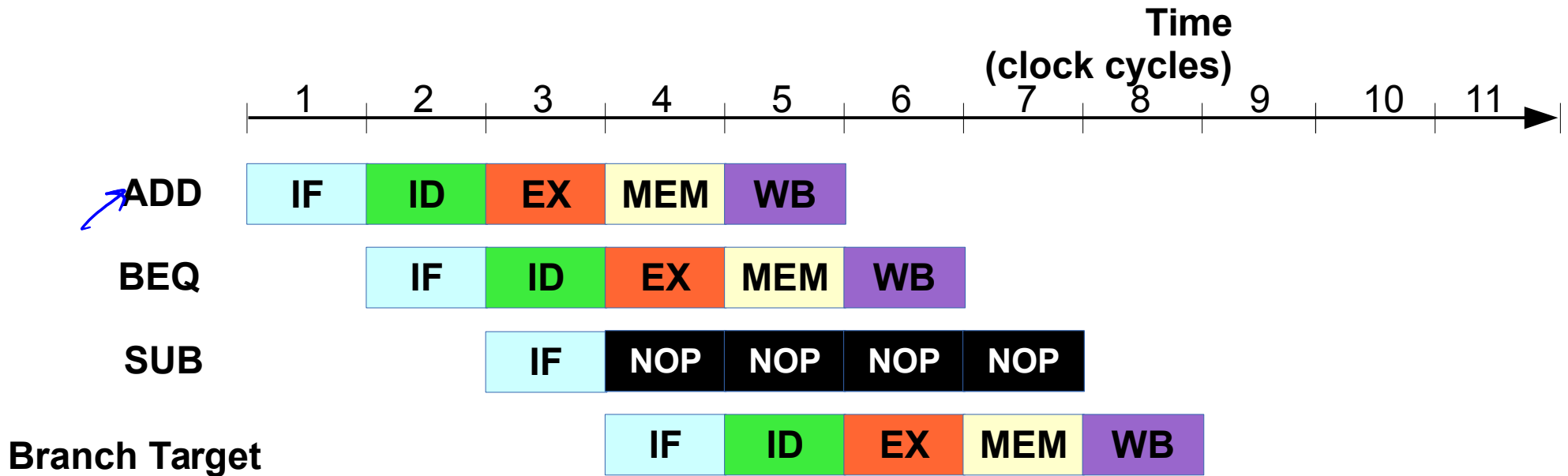
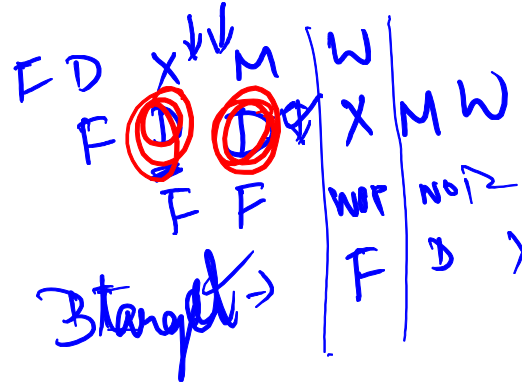
Branch Delay

ADD R2, R3,R4
BEQ R2, R4, loop
SUB R5, R5,R4
...



Branch Delay

or $n/1, n/2, n/3$
 ADD R2, R3, R4
 BEQ R2, R4, loop
 SUB R5, R5, R4
 ...

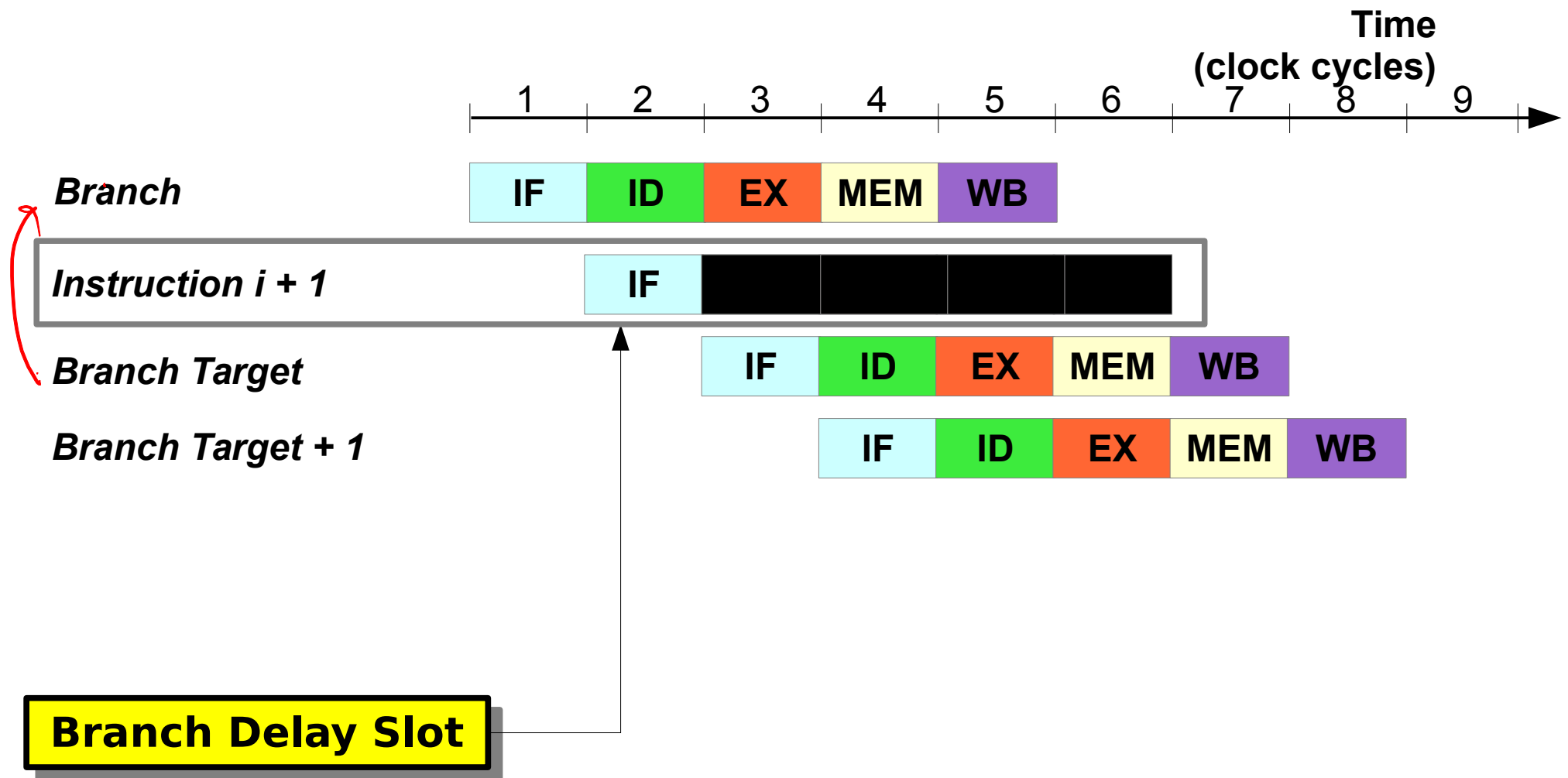


04 R11, R12, R13
add R2, — —

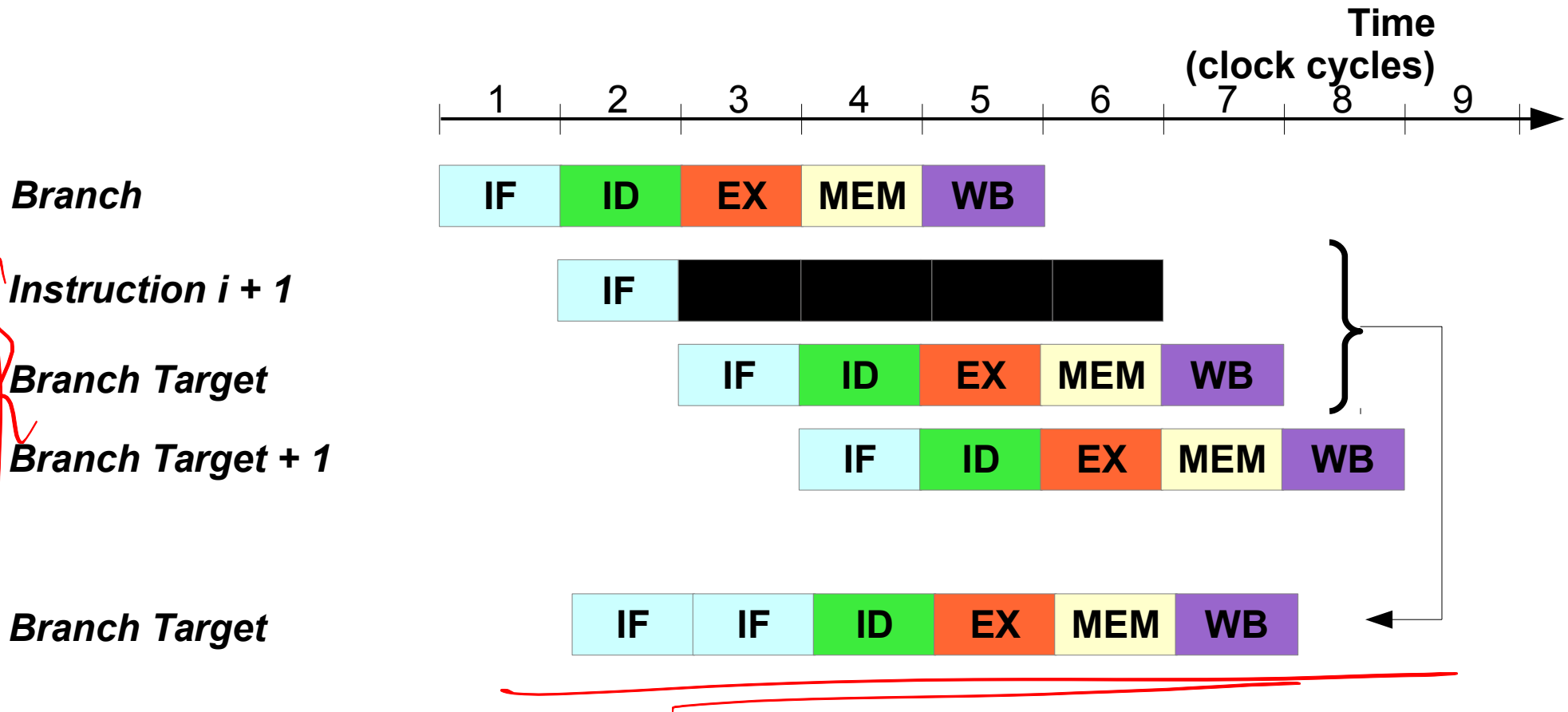
h2 R2, —

sub, R5, —

Branch Delay Slot




Branch Hazards



- 1 stall cycle for every branch yields a performance loss of 10% to 30%!

Example – Branch Taken

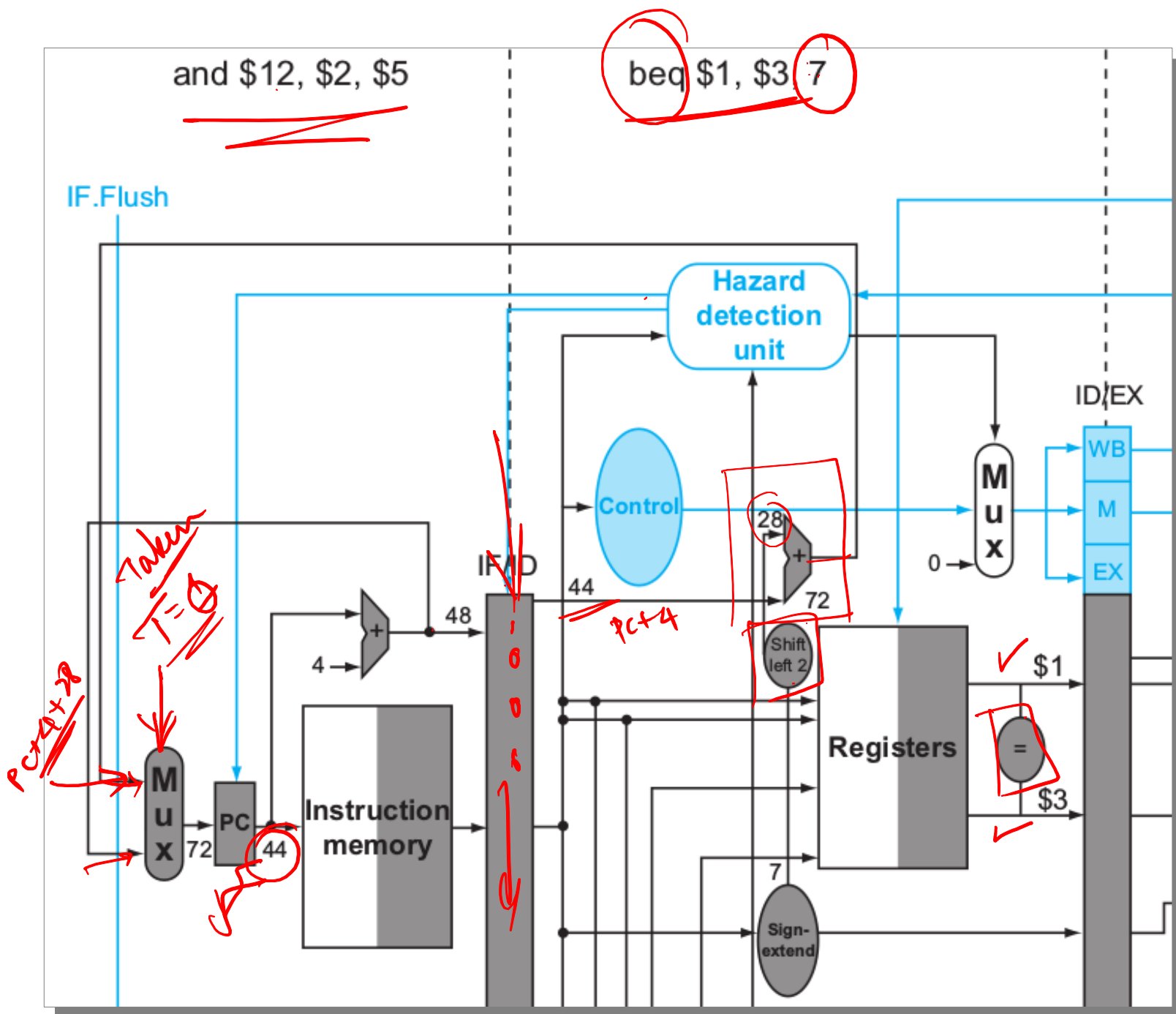


```
36 sub $10, $4, $8
40 beq $1, $3, 7
44 and $12, $2, $5
48 or  $13, $2, $6
52 add $14, $4, $2
56 slt $15, $6, $7
. . .
72 lw  $4, 50($7)
```

and \$12, \$2, \$5

beq \$1, \$3, 7

```
36 sub $10, $4, $8
40 beq $1, $3, 7 #
44 and $12, $2, $5
48 or $13, $2, $6
52 add $14, $4, $2
56 slt $15, $6, $7
. . .
72 lw $4, 50($7)
```



Branch Evaluation in ID – Steps

- Are beq inputs ready?

Branch Evaluation in ID – Steps

- Decode the instruction
- Decide whether to bypass/forward to the equality unit
 - Forwarding Logic has to be modified
- Complete the comparison
- If branch taken, set the PC to the branch target address.

Branch Evaluation – Forwarding

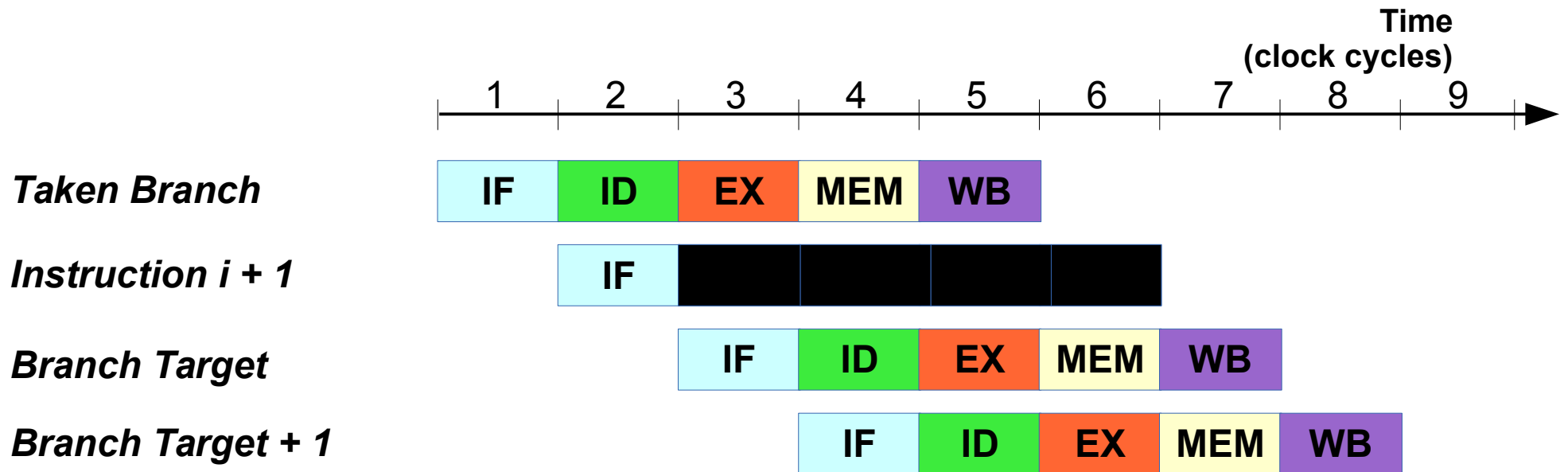
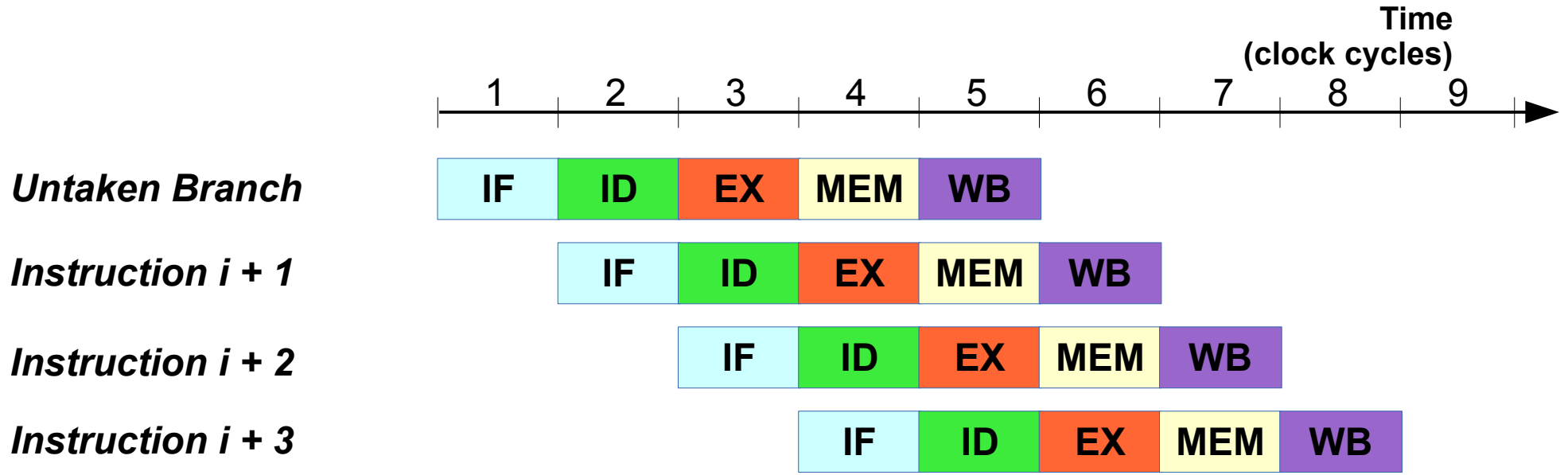
- New forwarding logic required
 - Source operands can come from ALU/MEM or MEM/WB
- If source operands are not ready, data hazard can occur and a stall will be needed.

```
...  
add    $1, $2, $2  
beq    $1, $3, 28  
...
```

Reducing Pipeline Branch Penalties

- Freeze the pipeline
- Static Prediction
 - Predict Untaken, Predict Taken
- Delayed Branch
 - Fill Branch Delay Slot

Predict Untaken Scheme



Reducing Pipeline Branch Penalties

- Predict Taken

beq	\$1, \$3, 28
and	\$12, \$2, \$5
or	\$13, \$6, \$2
add	\$14, \$2, \$2
lw	\$4, 50(\$7)

Reducing Pipeline Branch Penalties

- Predict Taken
 - Great if prediction is correct

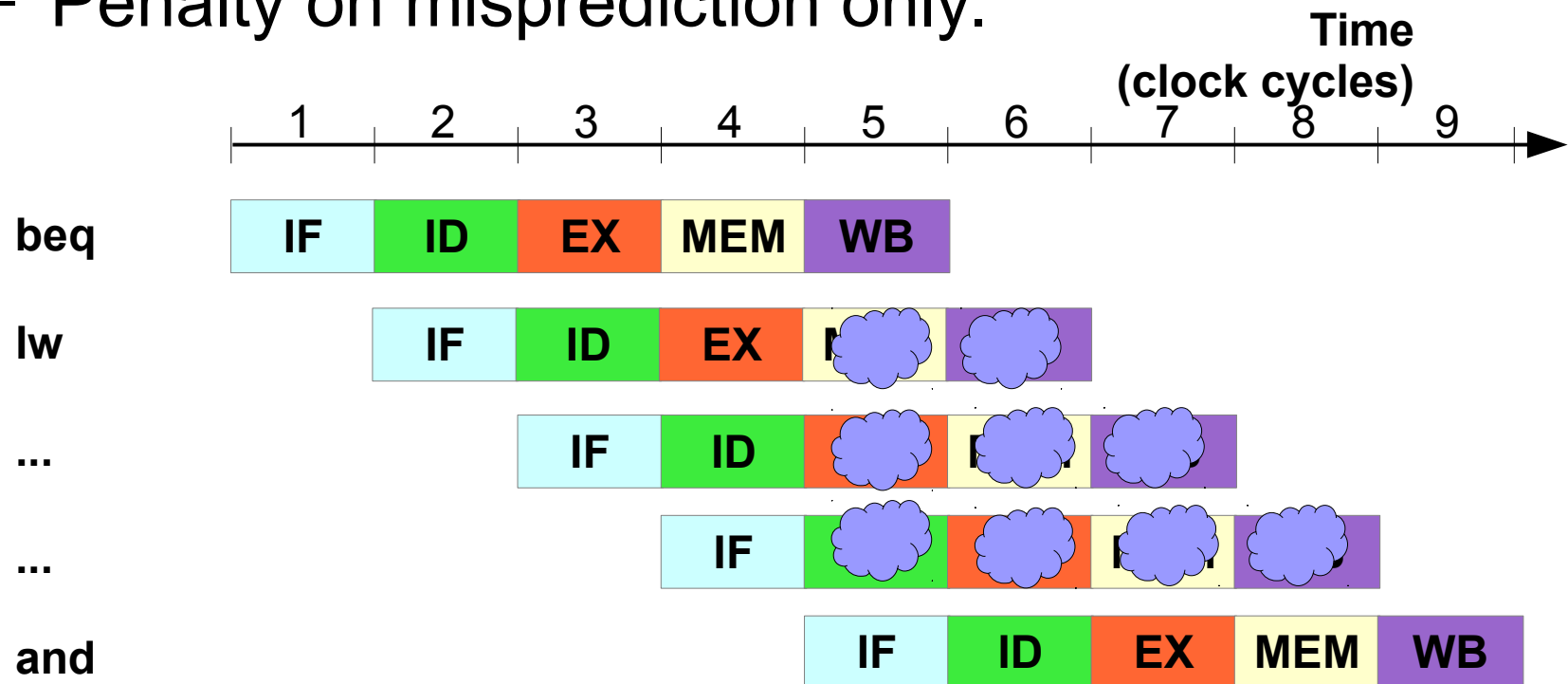
beq	\$1, \$3, 28
and	\$12, \$2, \$5
or	\$13, \$6, \$2
add	\$14, \$2, \$2
lw	\$4, 50(\$7)

Reducing Pipeline Branch Penalties

- Predict Taken

- Great if prediction is correct.
- Penalty on misprediction only.

beq	\$1, \$3, 28
and	\$12, \$2, \$5
or	\$13, \$6, \$2
add	\$14, \$2, \$2
lw	\$4, 50(\$7)



Reducing Pipeline Branch Penalties

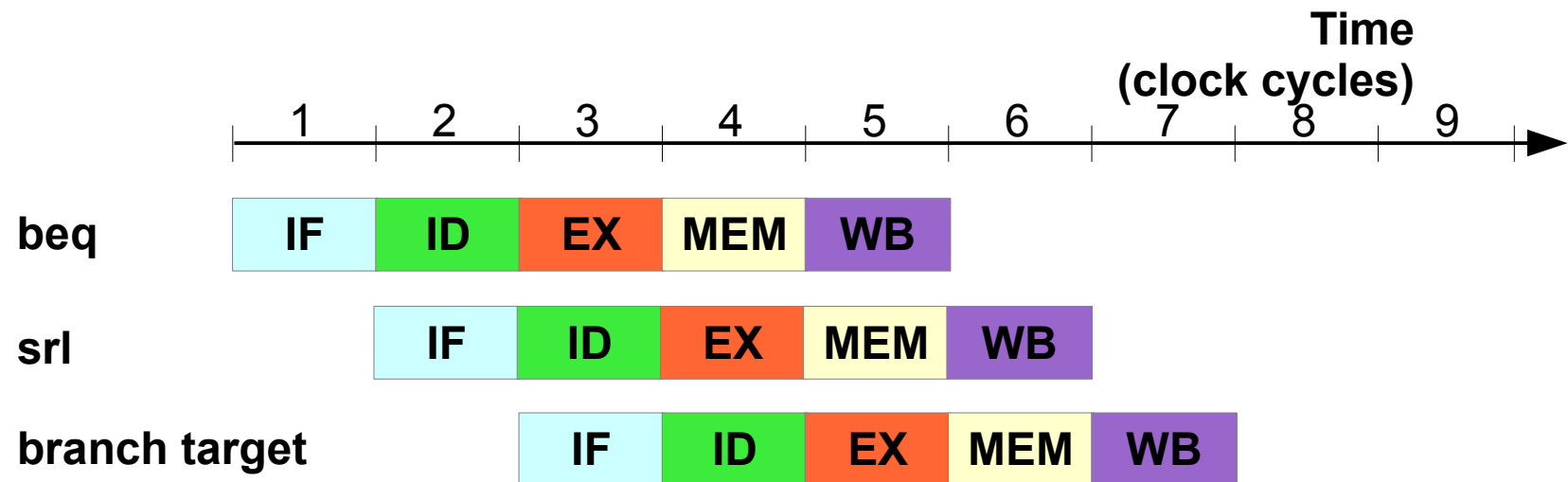
- Fill the branch delay slot

sll	...
srl	\$14, \$2, \$2
beq	\$1, \$3, 28
and	\$12, \$2, \$5
...	

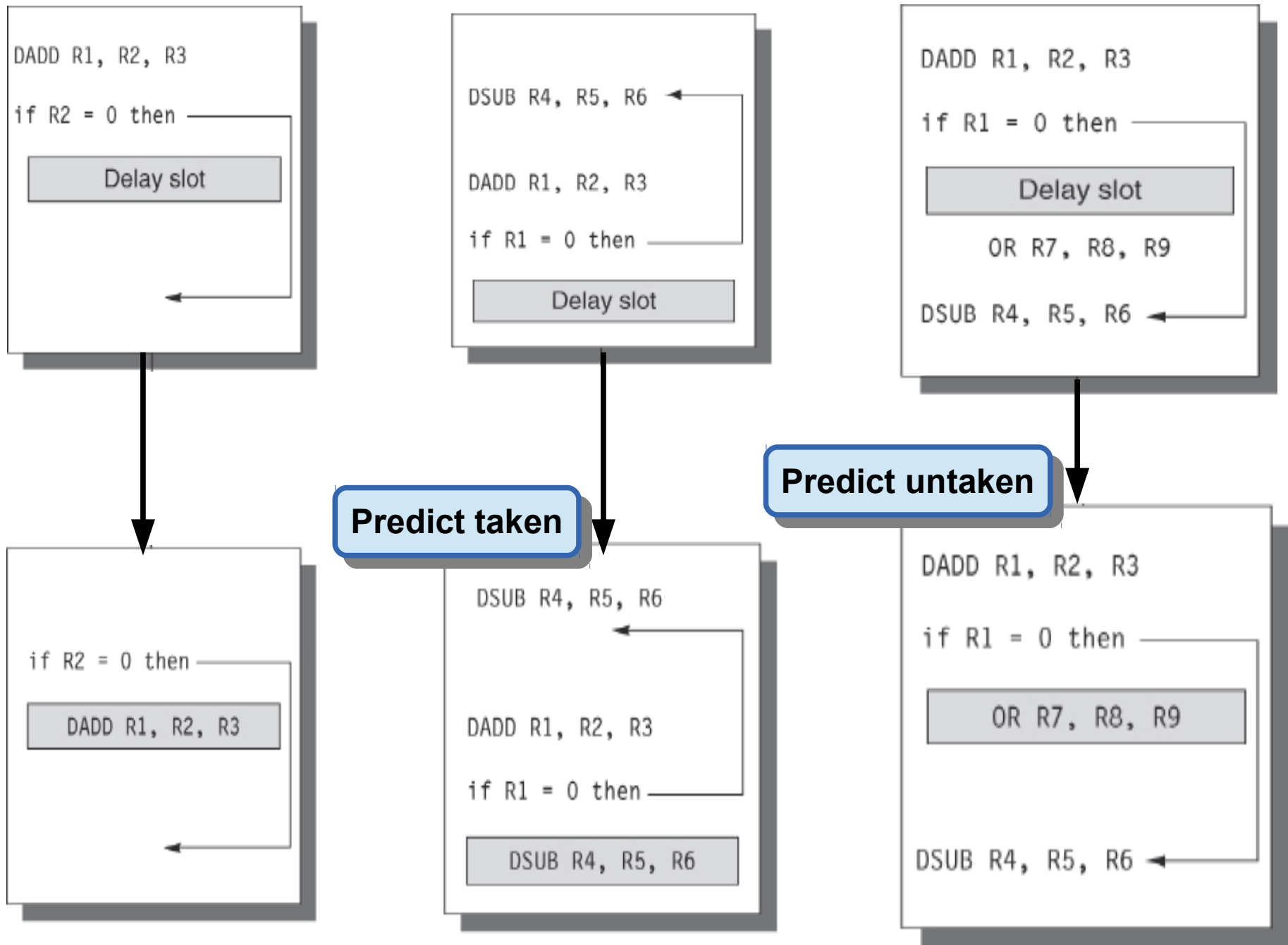
Reducing Pipeline Branch Penalties

- Fill the branch delay slot

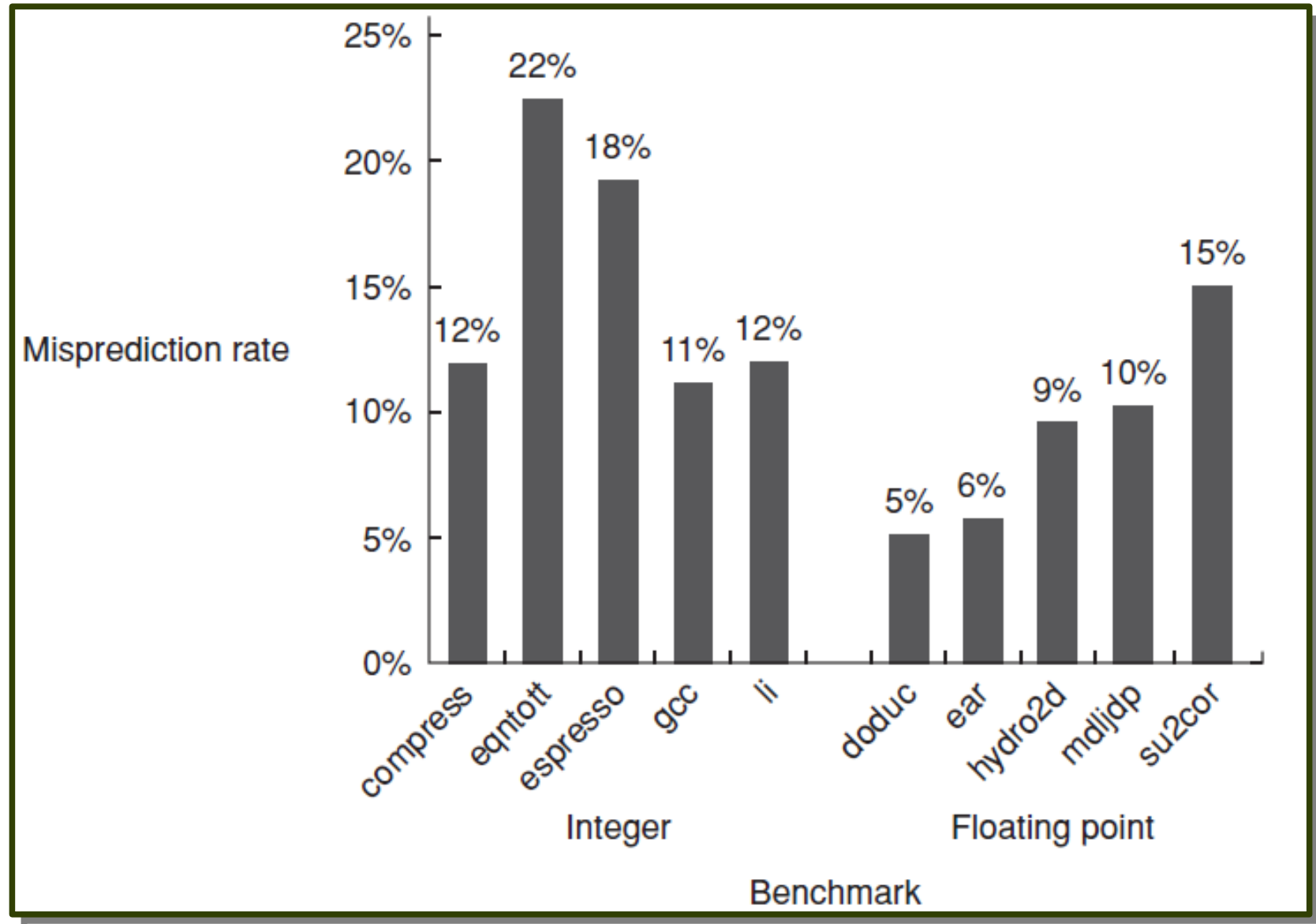
sll	...
srl	\$14, \$2, \$2
beq	\$1, \$3, 28
and	\$12, \$2, \$5
...	



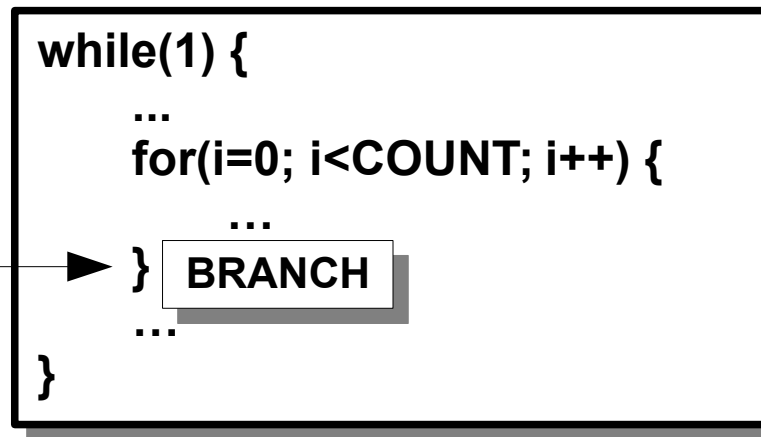
Branch Delay Slot



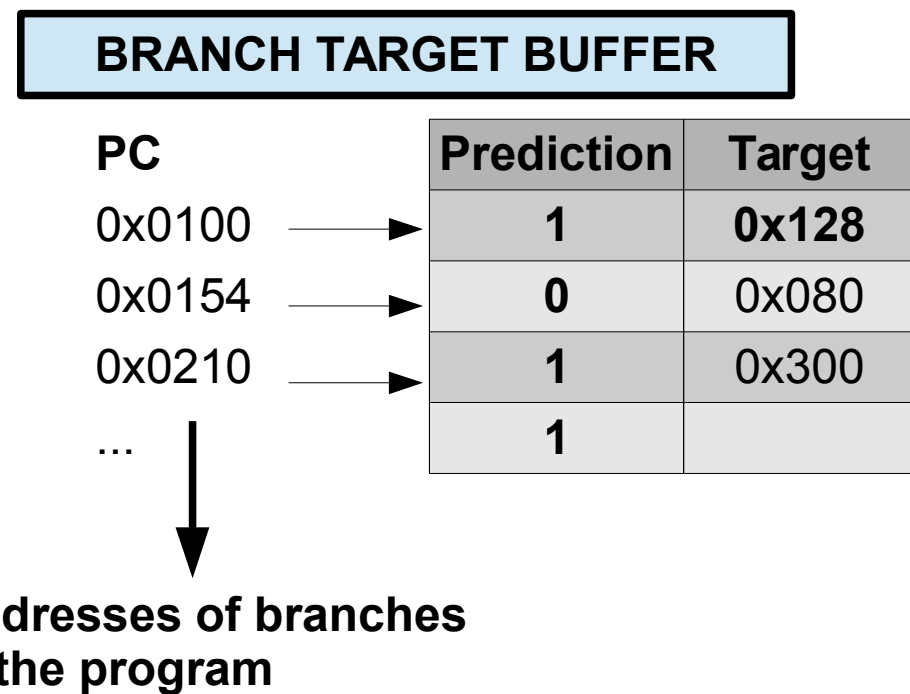
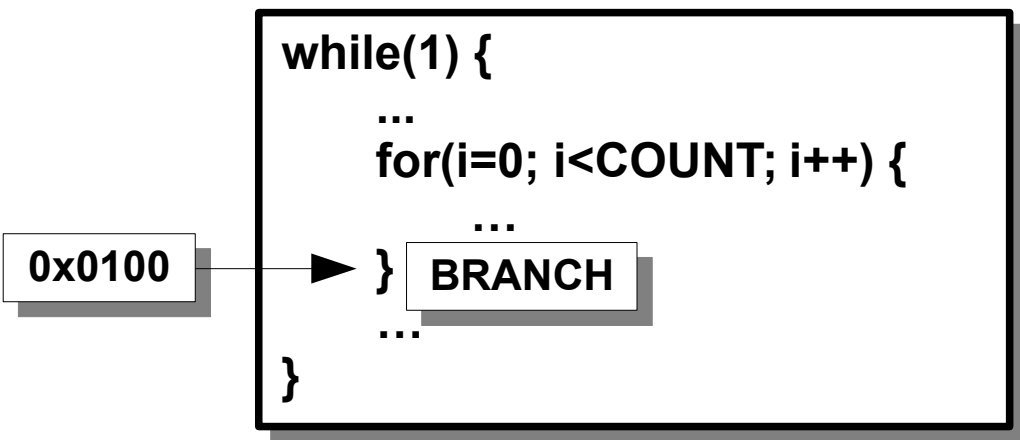
Static Branch Prediction



Dynamic Branch Prediction

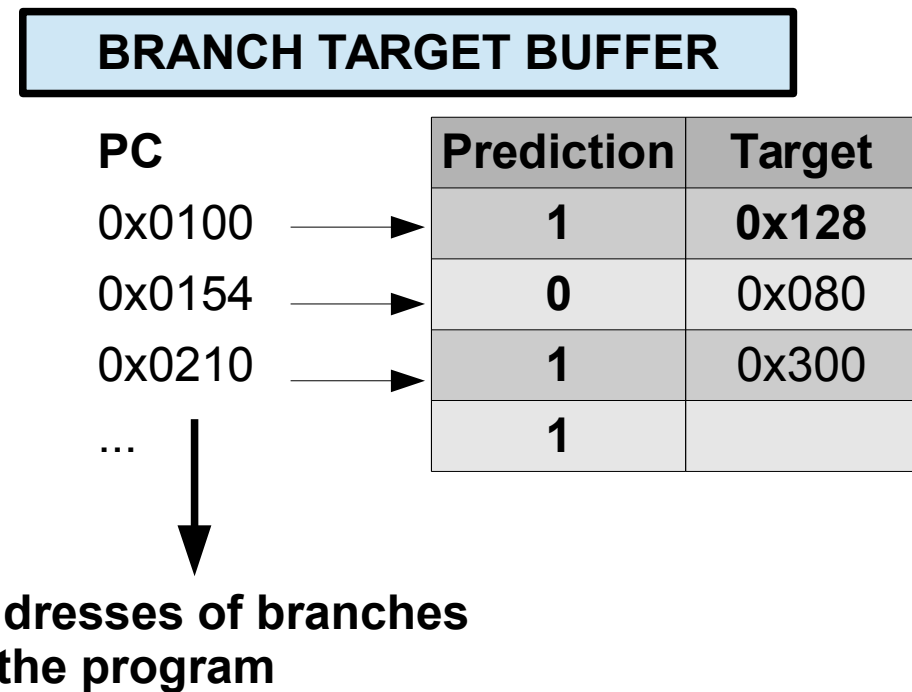
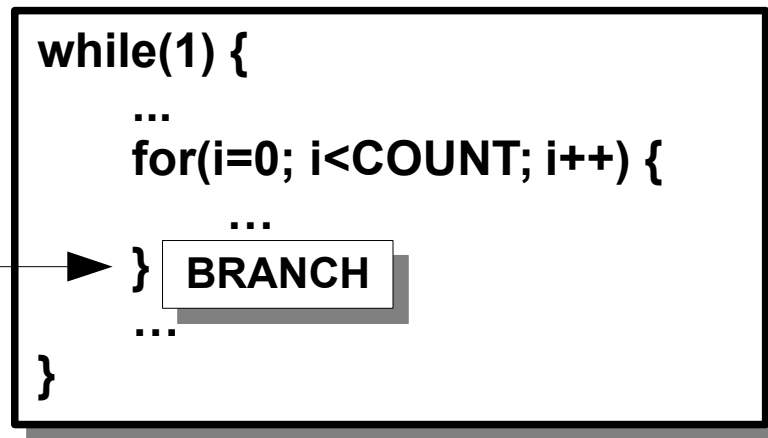


Dynamic Branch Prediction



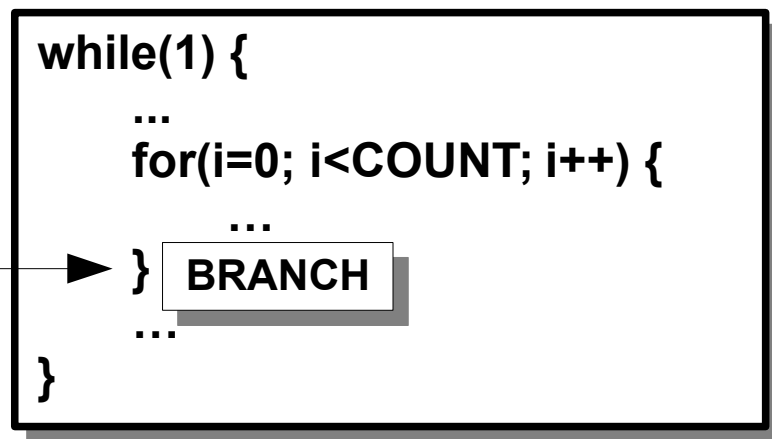
Dynamic Branch Prediction

- Branch Target Buffer
 - Single bit predictors (1-bit bimodal predictor)
 - Change prediction with branch behaviour
 - No. of wrong predictions?



Dynamic Branch Prediction

- Branch Target Buffer
 - Single bit predictors (1-bit bimodal predictor)
 - Change prediction with branch behaviour
 - No. of wrong predictions?



Branch instruction behaviour



Wrong Predictions

BRANCH TARGET BUFFER

PC

0x0100

0x0154

0x0210

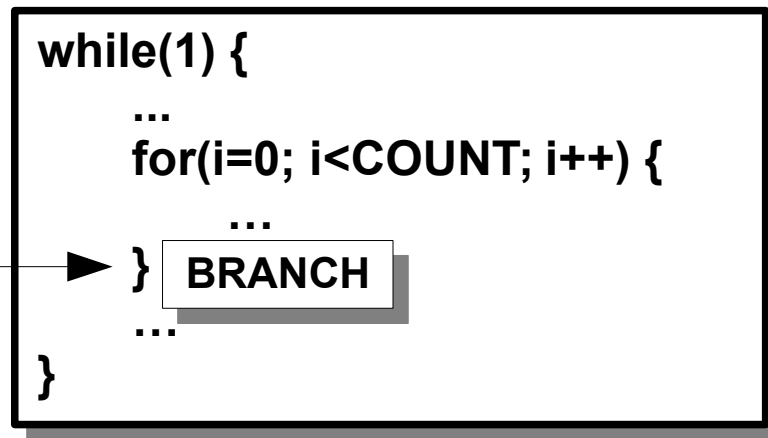
...

Prediction	Target
1	0x128
0	0x080
1	0x300
1	

Addresses of branches
in the program

Dynamic Branch Prediction

- Branch Target Buffer
 - Single bit predictors (1-bit bimodal predictor)
 - Change prediction with branch behaviour
 - No. of wrong predictions?



Branch instruction behaviour



Wrong Predictions

BRANCH TARGET BUFFER

PC

0x0100

0x0154

0x0210

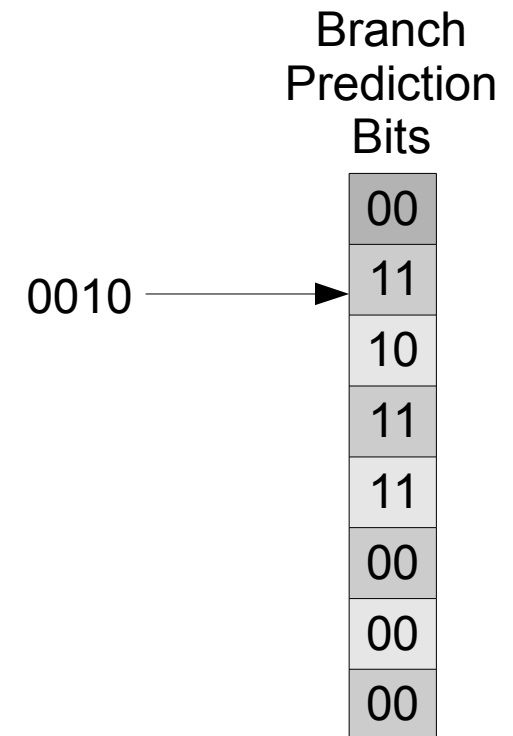
...

Prediction	Target
1	0x128
0	0x080
1	0x300
1	

Addresses of branches
in the program

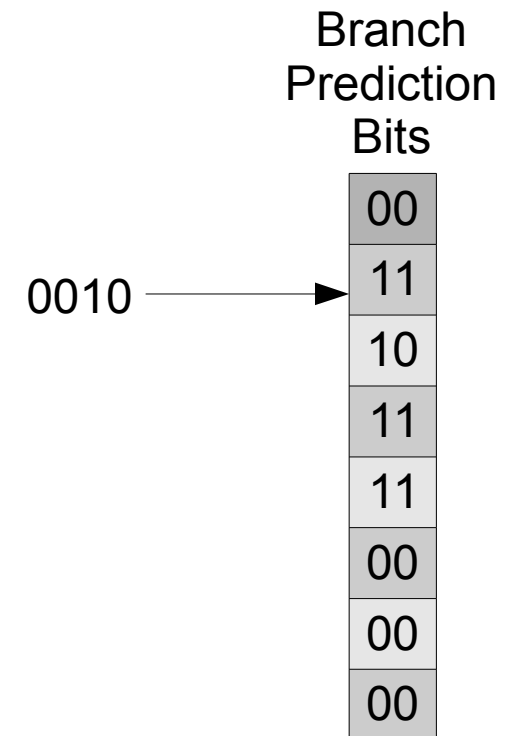
Can we do better?

Dynamic Branch Prediction



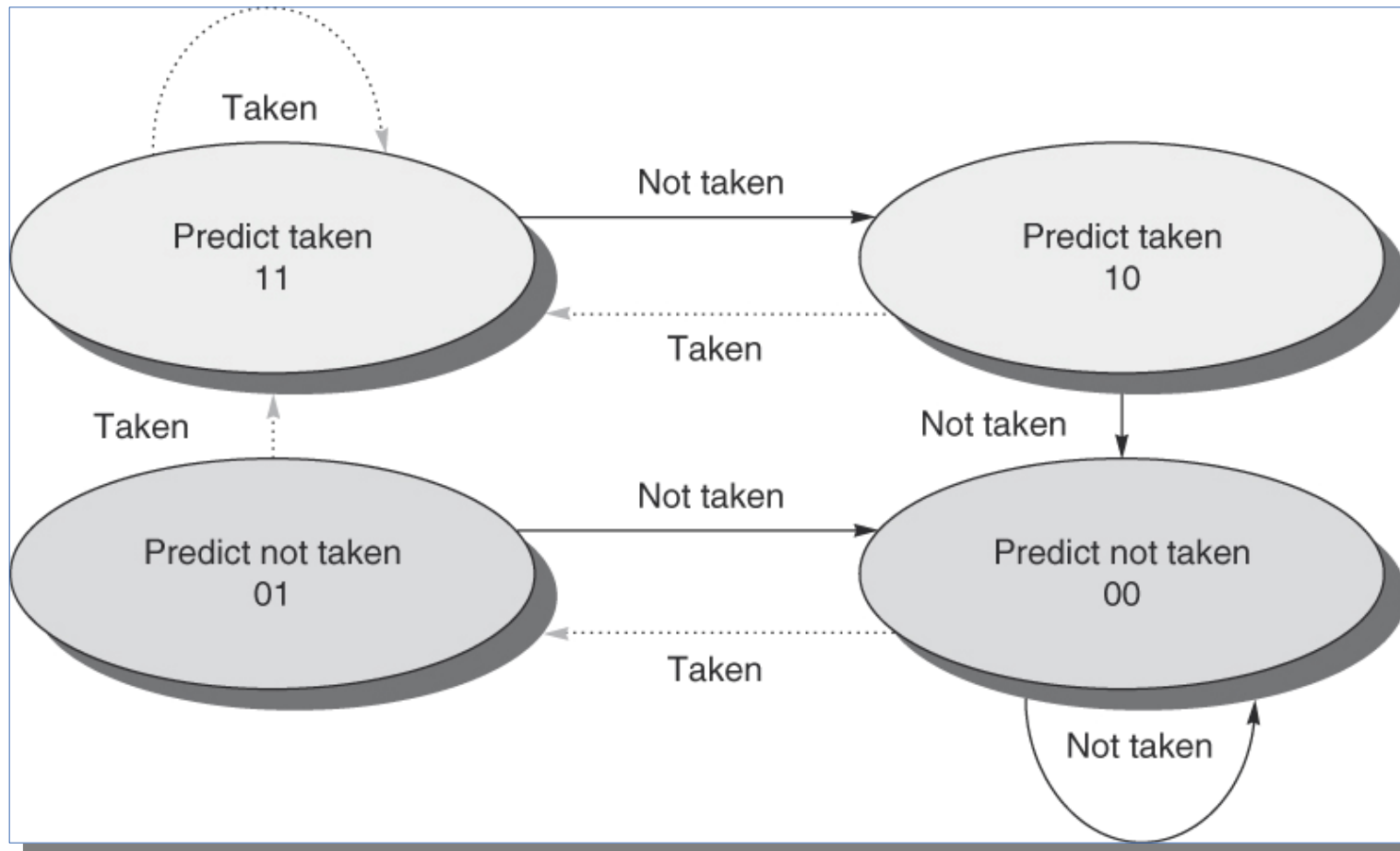
Dynamic Branch Prediction

- 2-bit predictors
 - 2-bit Bimodal Saturating Counter



Dynamic Branch Prediction

- 2-bit predictors
 - 2-bit Bimodal Saturating Counter



Branch
Prediction
Bits

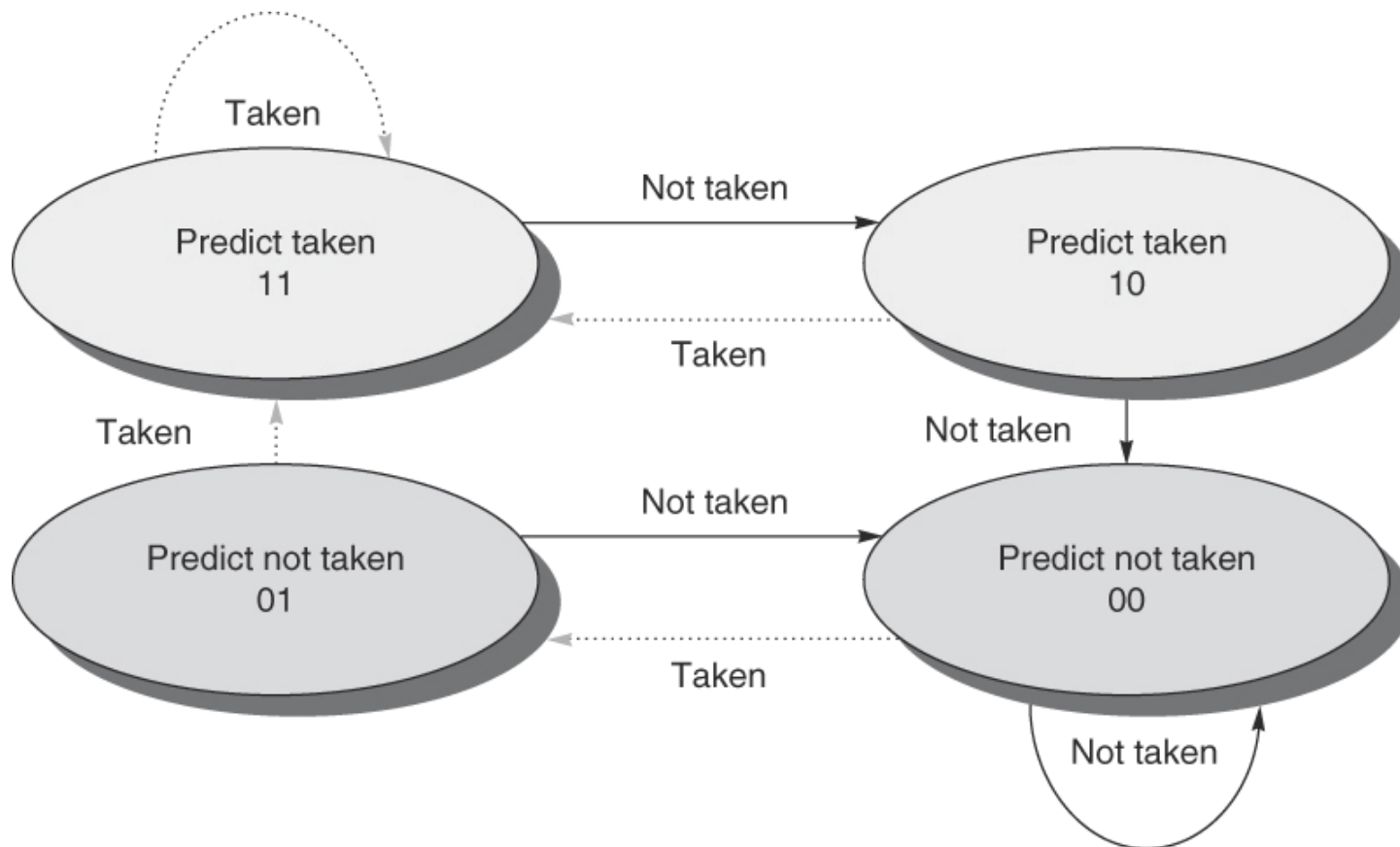
0010

00
11
10
11
11
00
00
00

Dynamic Branch Prediction

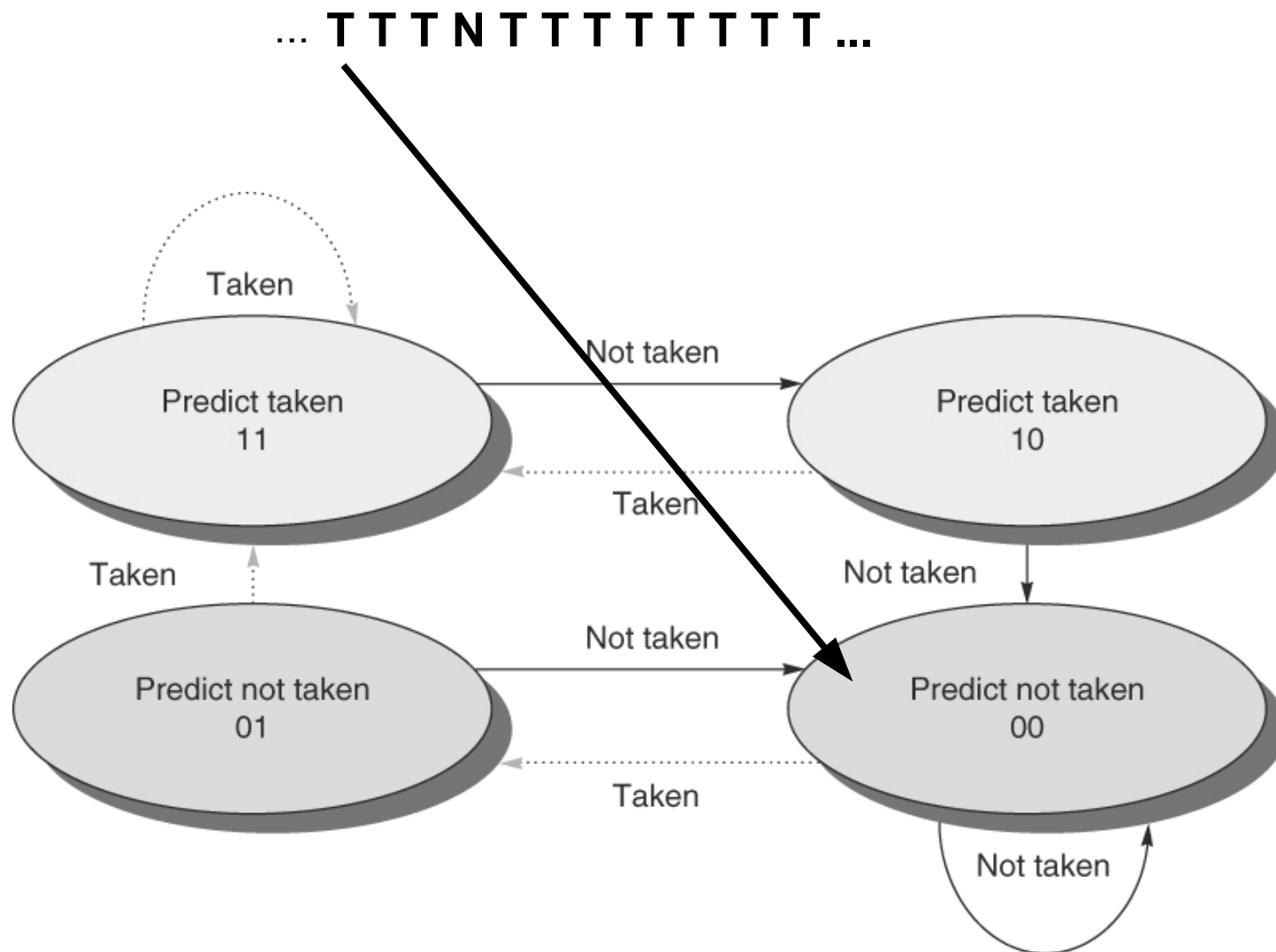
- 2-bit predictors

... TTTNTTTT TTTT ...



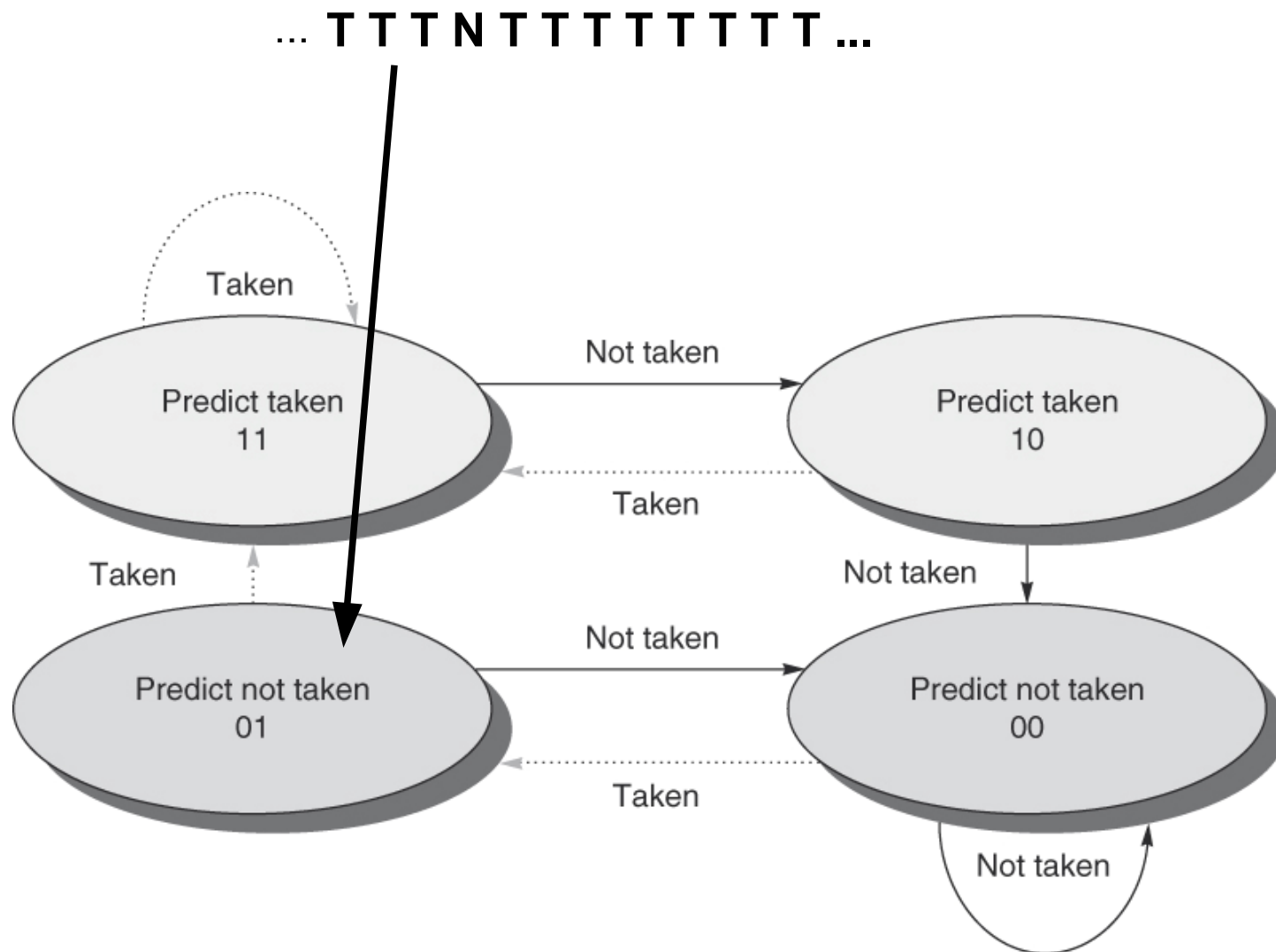
Dynamic Branch Prediction

- 2-bit predictors



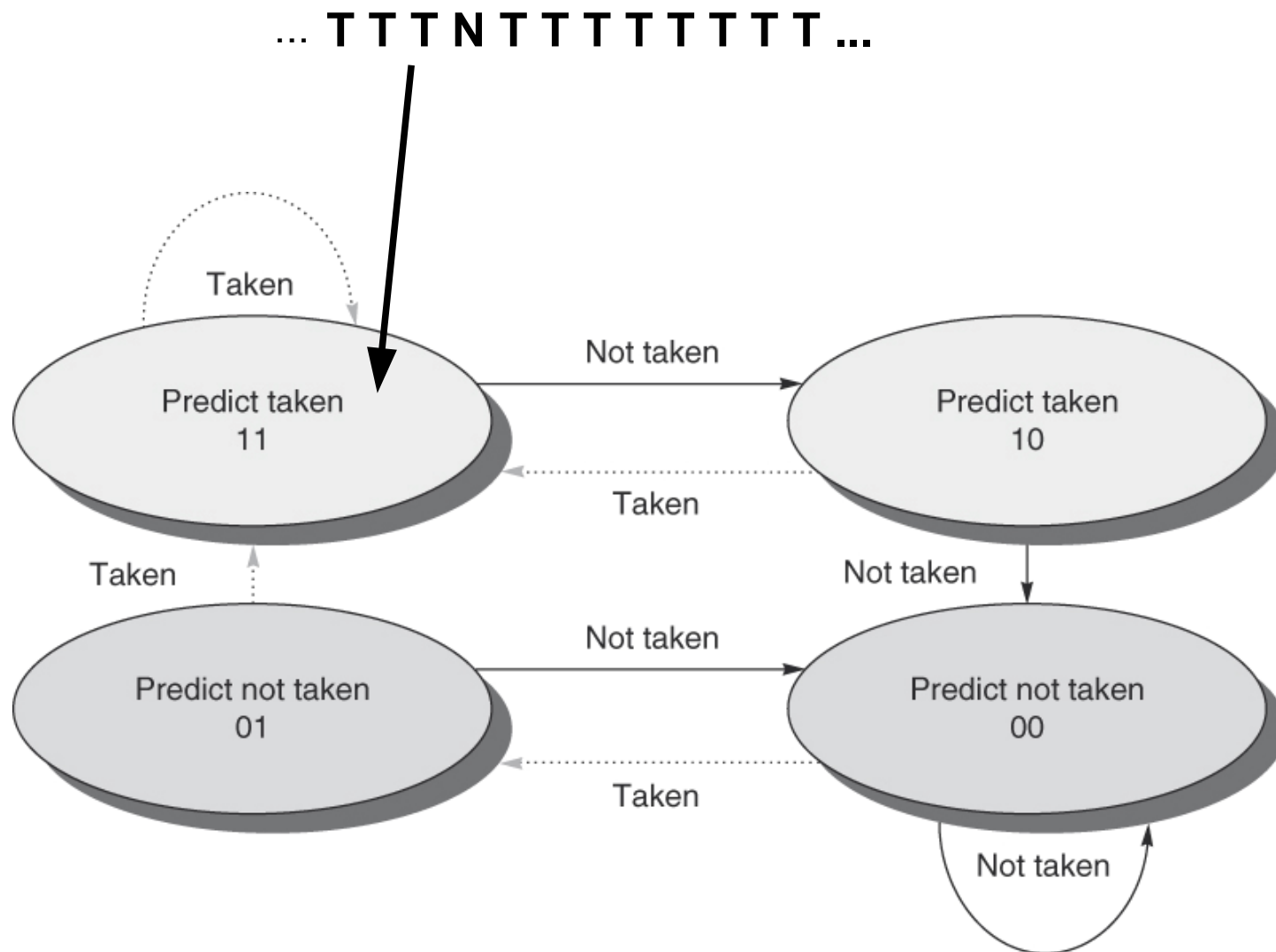
Dynamic Branch Prediction

- 2-bit predictors



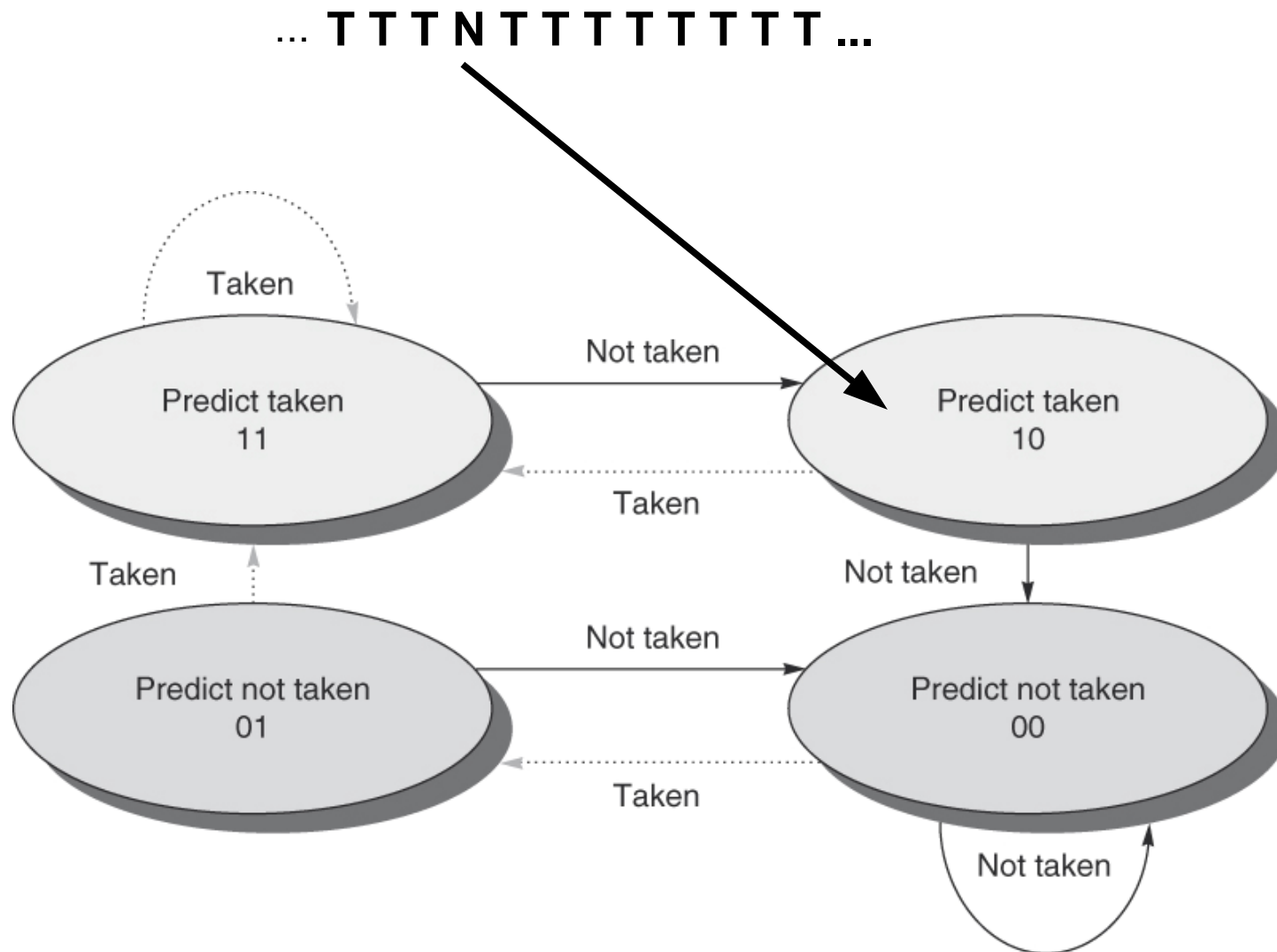
Dynamic Branch Prediction

- 2-bit predictors



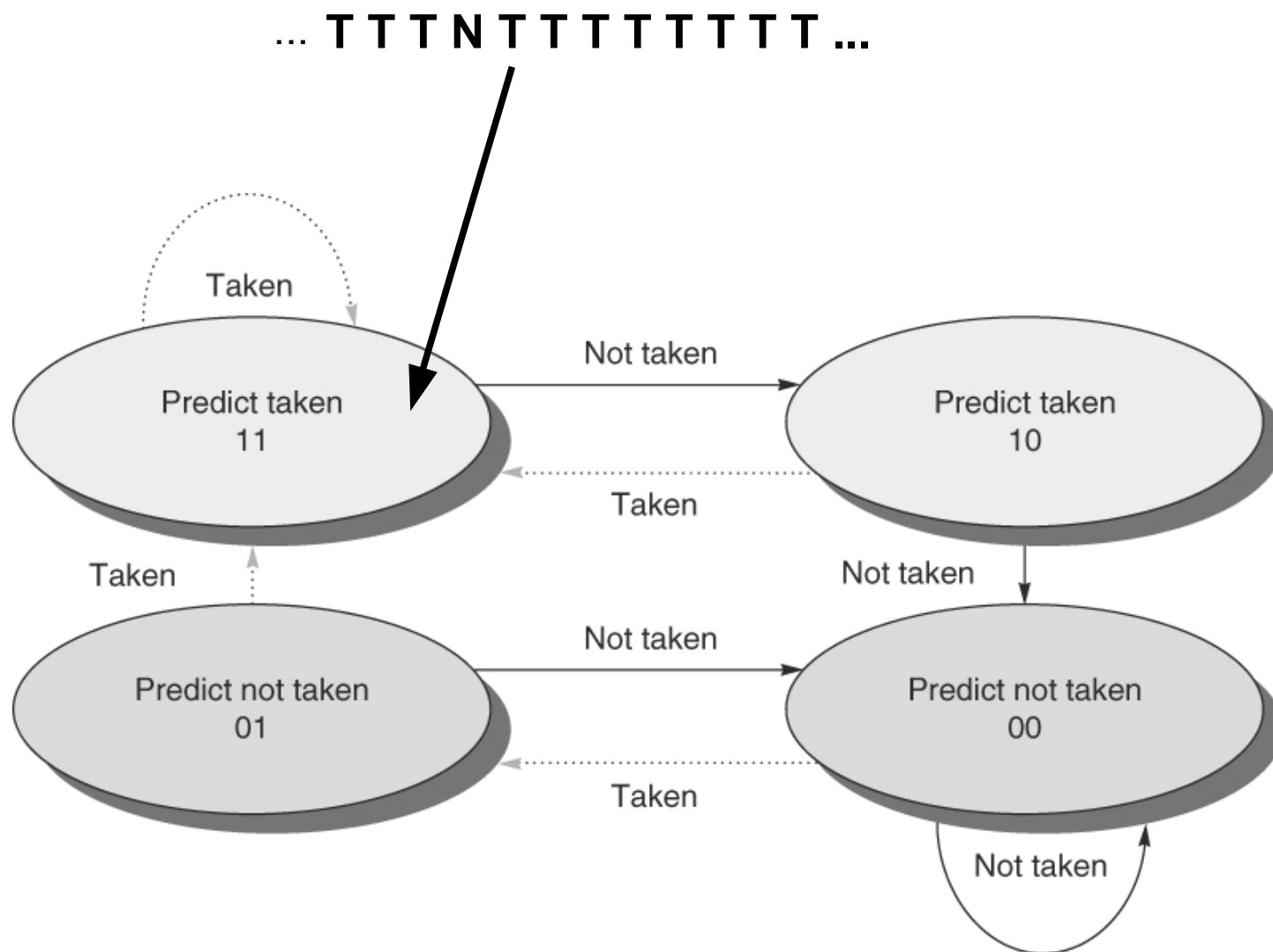
Dynamic Branch Prediction

- 2-bit predictors



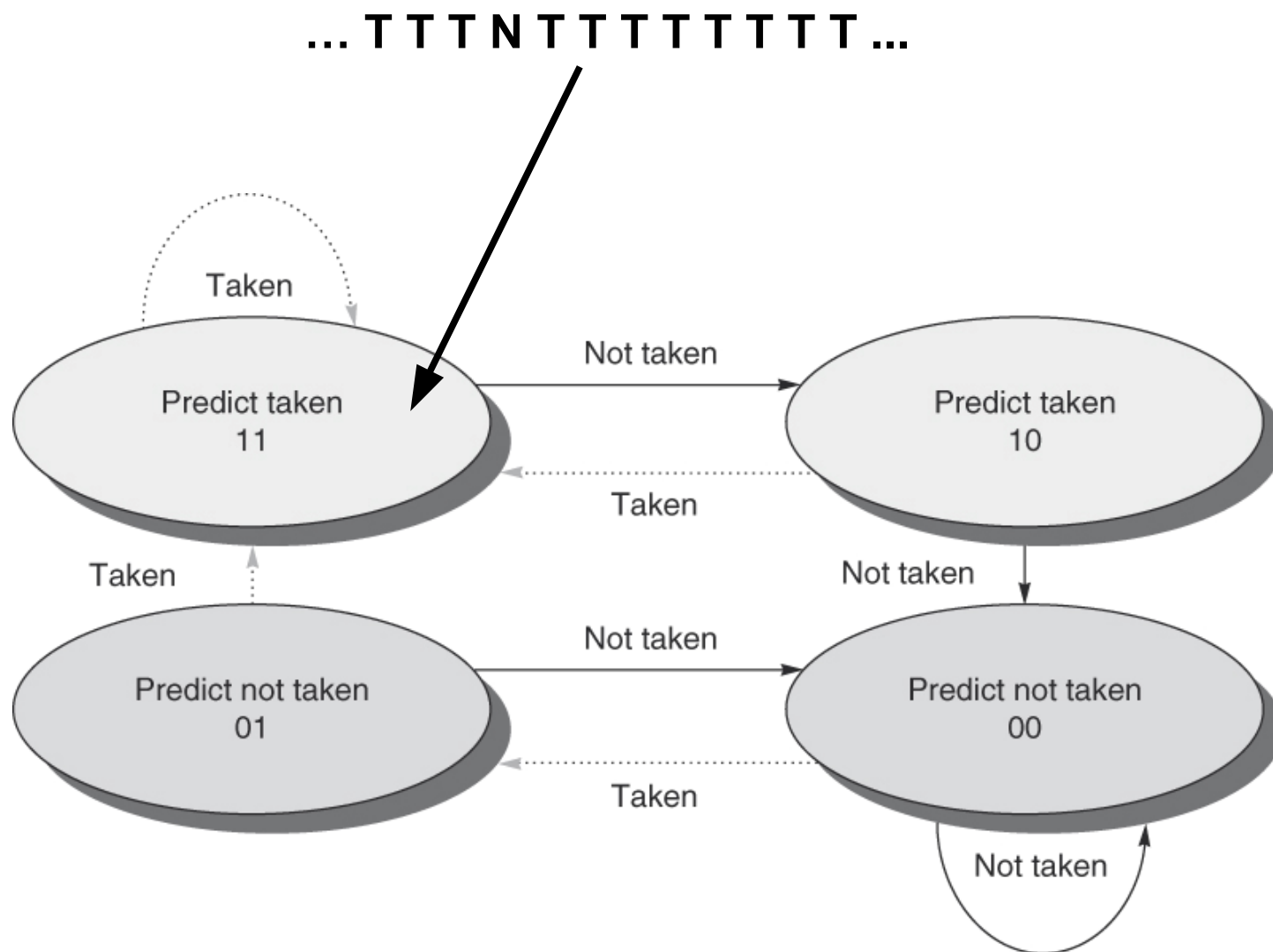
Dynamic Branch Prediction

- 2-bit predictors



Dynamic Branch Prediction

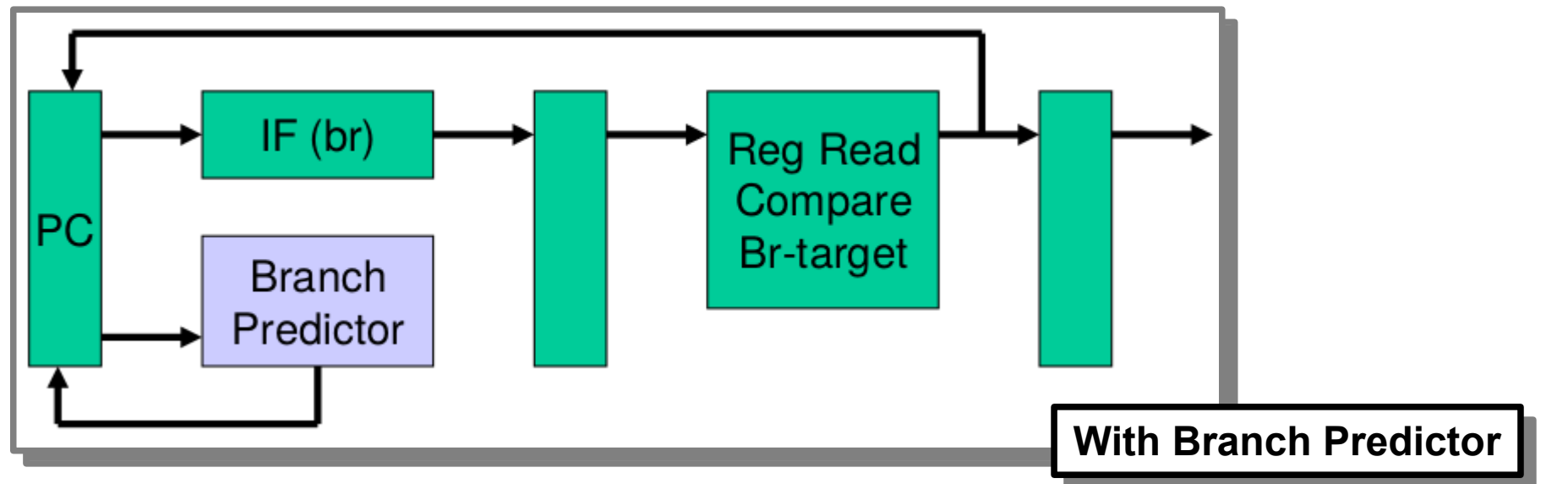
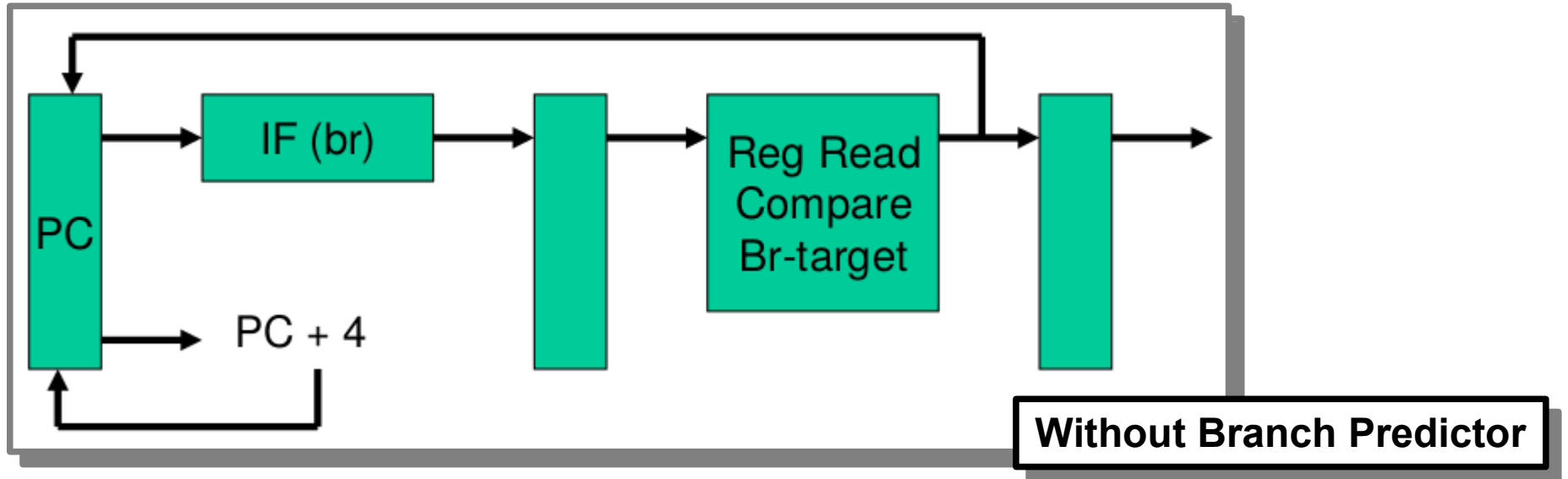
- 2-bit predictors



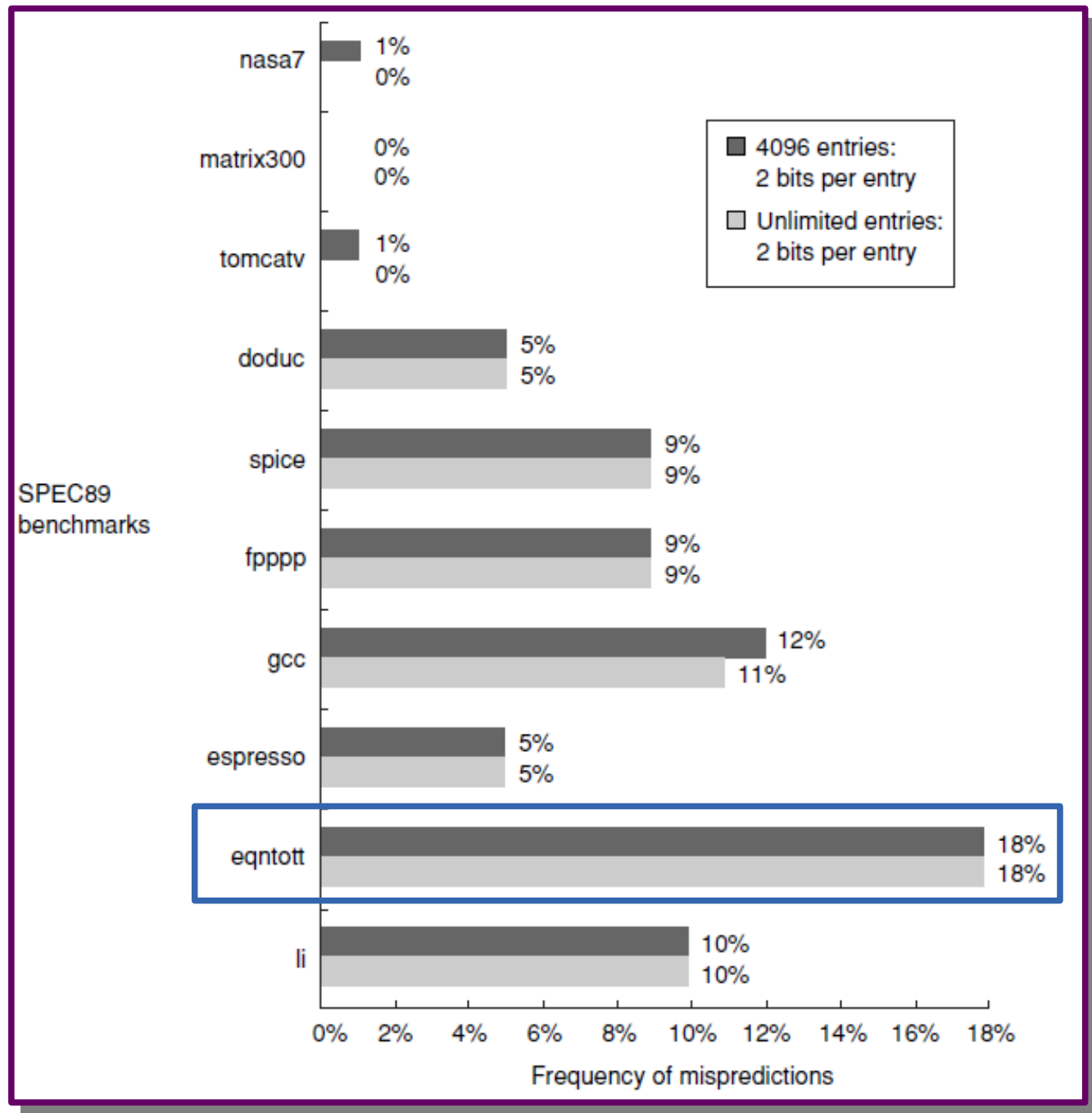
Dynamic Branch Prediction

- n-bit saturating counters

Branch Predictors



Dynamic Branch Prediction



Observations?

Correlating Branch Predictors

eqntott code

```
if (aa == 2)
```

```
    aa = 0;
```

```
if (bb == 2)
```

```
    bb = 0;
```

```
if (aa!=bb) {
```

Correlating Branch Predictors

eqntott code

```
if (aa == 2)
    aa = 0;
if (bb == 2)
    bb = 0;
if (aa!=bb) {
```

Outcome of the previous branch

X 0010

Branch Prediction
Buffer

00
11
10
11
11
00
00
01

Correlating Branch Predictors

eqntott code

```
if (aa == 2)
    aa = 0;
if (bb == 2)
    bb = 0;
if (aa!=bb) {
```

Outcome of the previous branch

X 0010

Branch Prediction
Buffer

00
11
10
11
11
00
00
01

- Two-level predictors
 - (1,2) predictor

Correlating Branch Predictors

eqntott code

```
if (aa == 2)
    aa = 0;
if (bb == 2)
    bb = 0;
if (aa!=bb) {
```

Outcome of the previous branch

X 0010

Branch Prediction
Buffer

00
11
10
11
11
00
00
01

- Two-level predictors
 - (1,2) predictor

Correlating Branch Predictors

if (aa == 2)

aa = 0;

if (bb == 2)

bb = 0;

if (aa!=bb) {

Outcomes of the
previous 2 branches

XX 0010

Branch Prediction
Buffer

00
11
10
11
11
00
00
01

- A 4096 bit, (2,2) buffer supports how many branch instructions?

Correlating Branch Predictors

```
if (aa == 2)
```

```
    aa = 0;
```

```
if (bb == 2)
```

```
    bb = 0;
```

```
if (aa!=bb) {
```

Outcomes of the
previous 2 branches

xx 0010

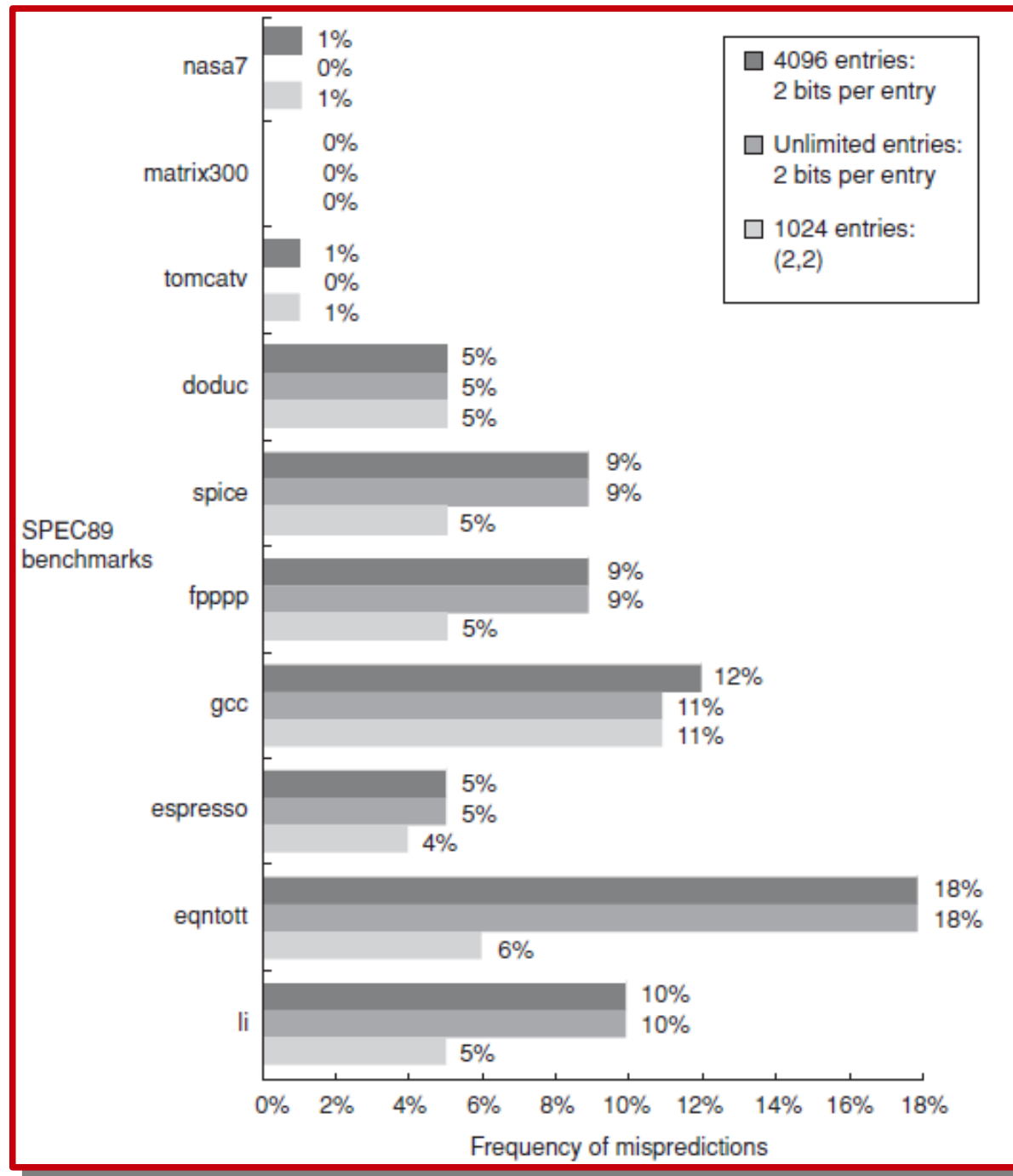
Branch Prediction
Buffer

00
11
10
11
11
00
00
01

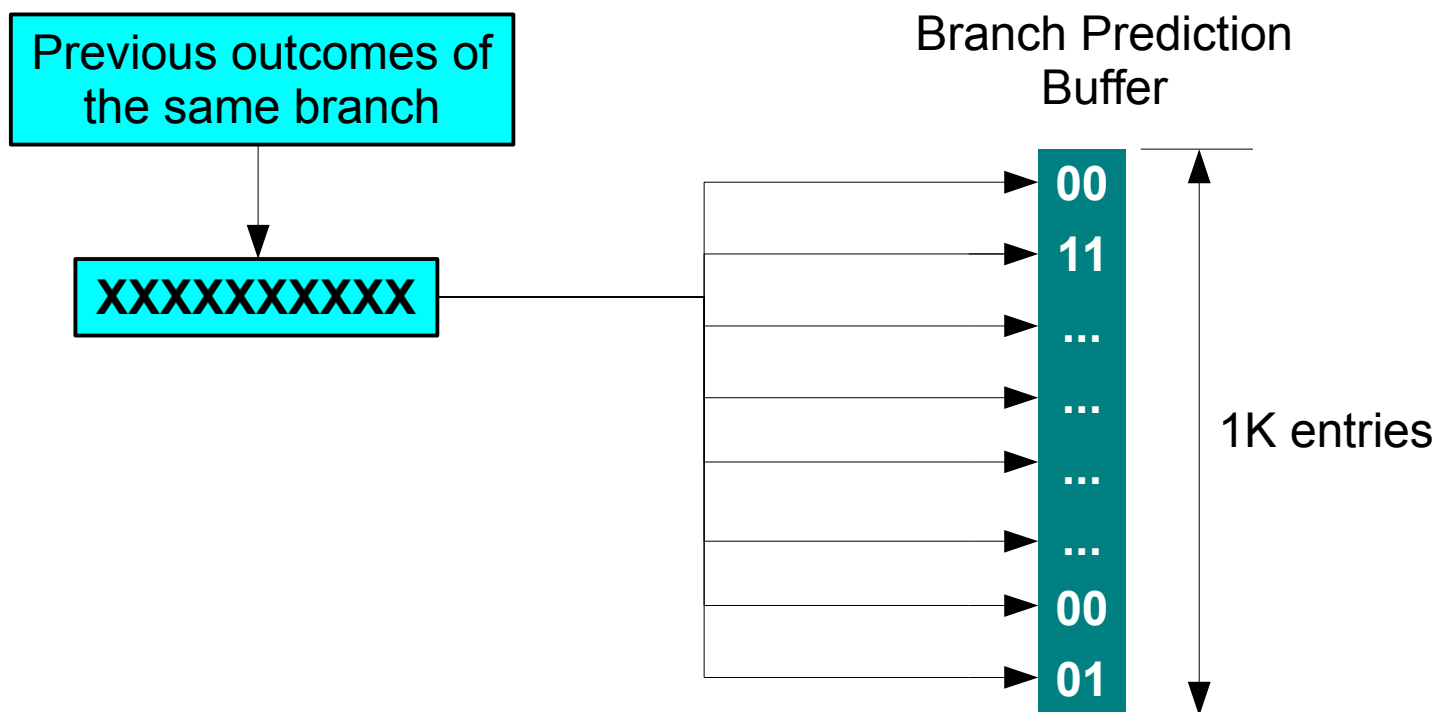
- A 4096 bit, (2,2) buffer supports how many branch instructions?

$$(m, n) \text{ BPB bits} = 2^m \times n \times \text{No. of prediction entries}$$

Correlating Branch Predictors

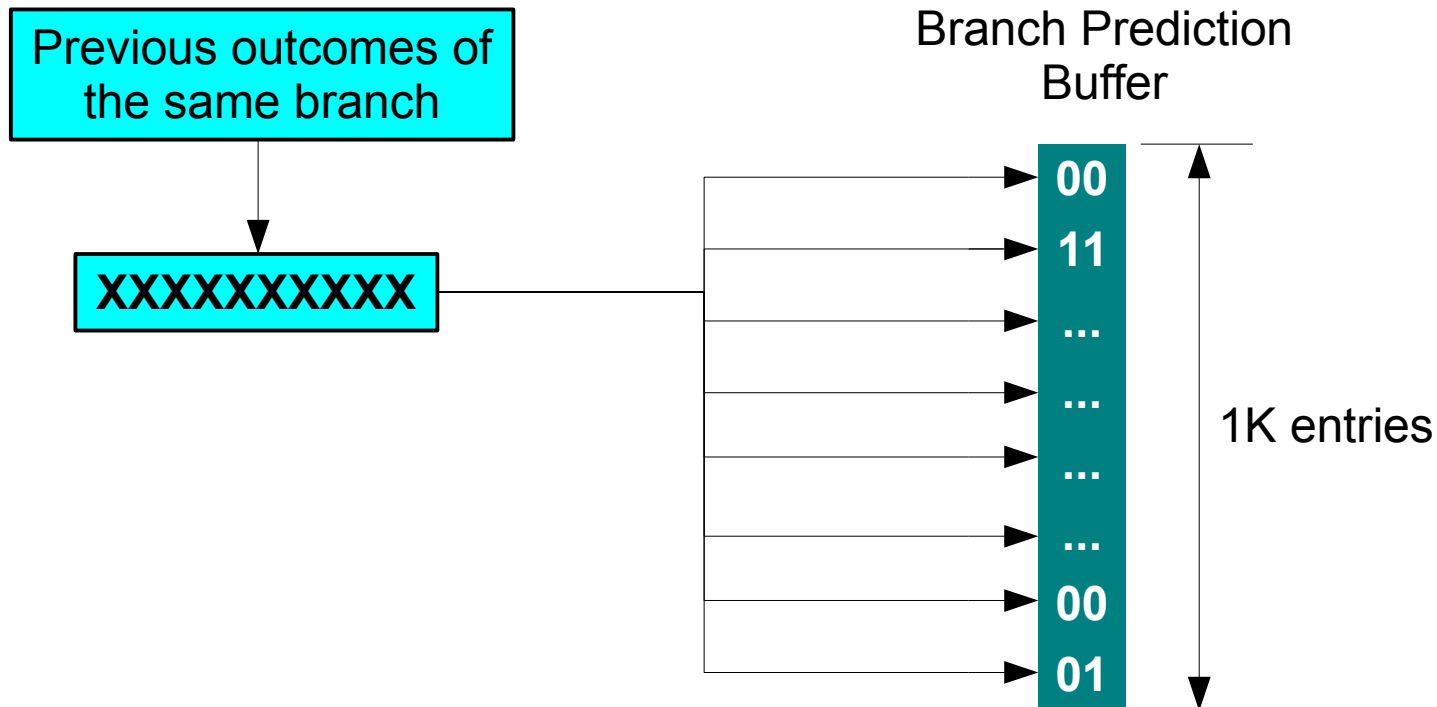


Local Predictors



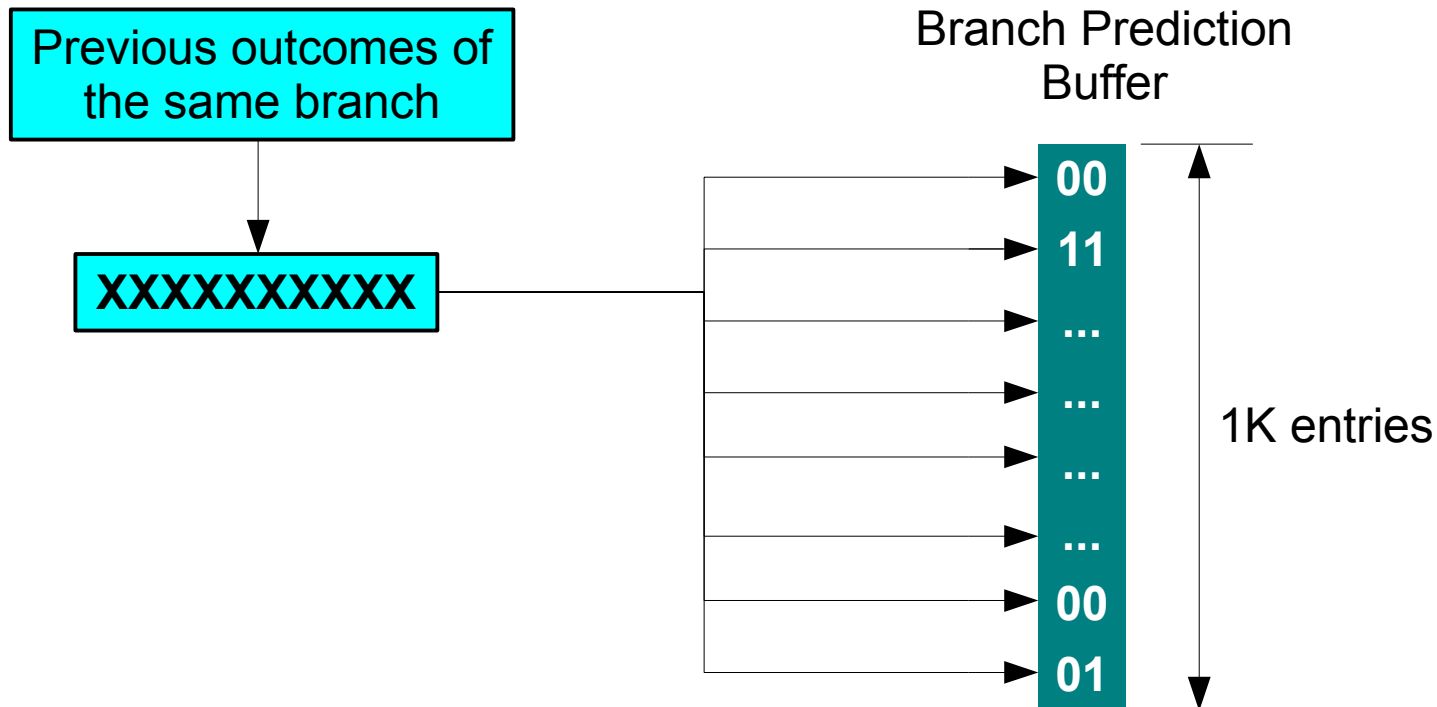
Local Predictors

- A history of branch behaviour is recorded

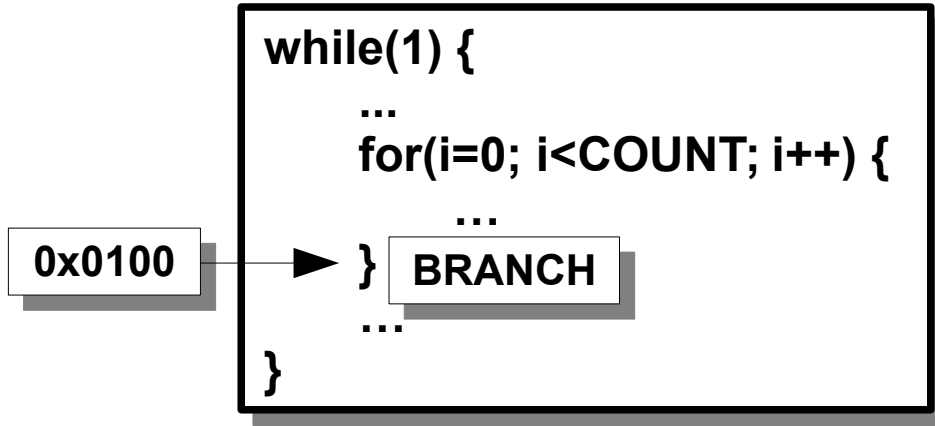


Local Predictors

- A history of branch behaviour is recorded
- One for each possible combination of outcomes for the last n occurrences of this branch



Local Predictors – Example

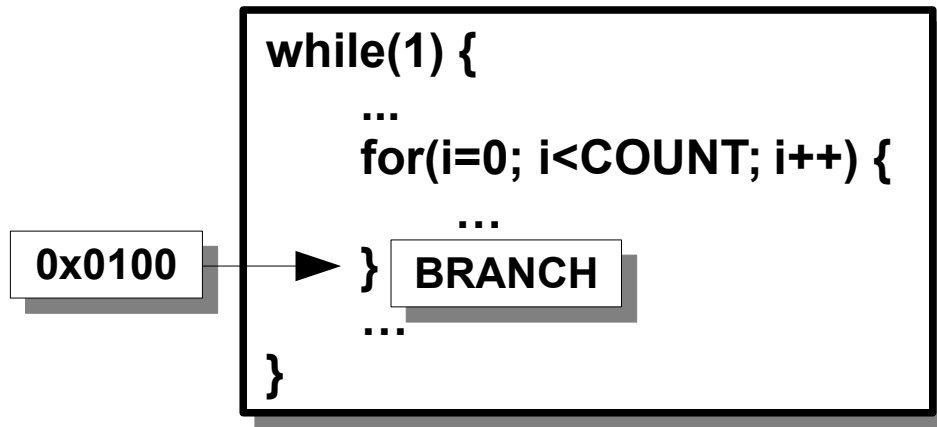


Branch instruction behaviour

... T T N T T T T N T T T T N T T ...

... 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 ...

Local Predictors – Example



Branch instruction behaviour

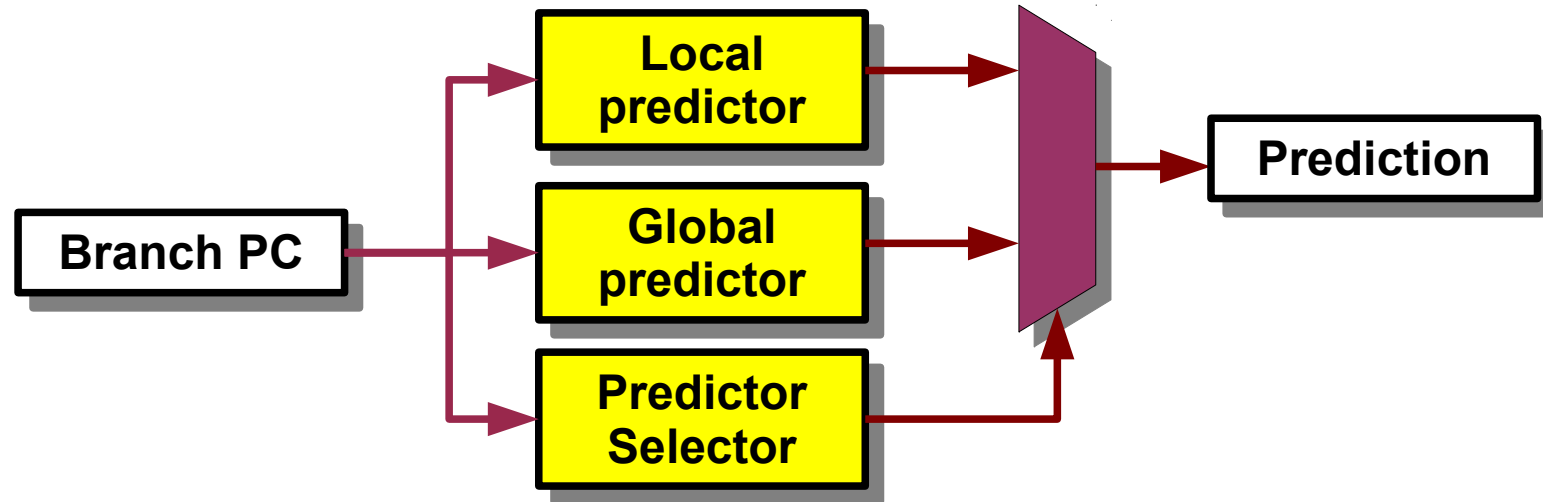
... T T N T T T T N T T T T N T T ...

... 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 ...

Local History	Prediction
0 1 1 1 1	NT
1 1 1 1 0	T
1 1 1 0 1	T
1 1 0 1 1	T
1 0 1 1 1	T
0 1 1 1 1	NT

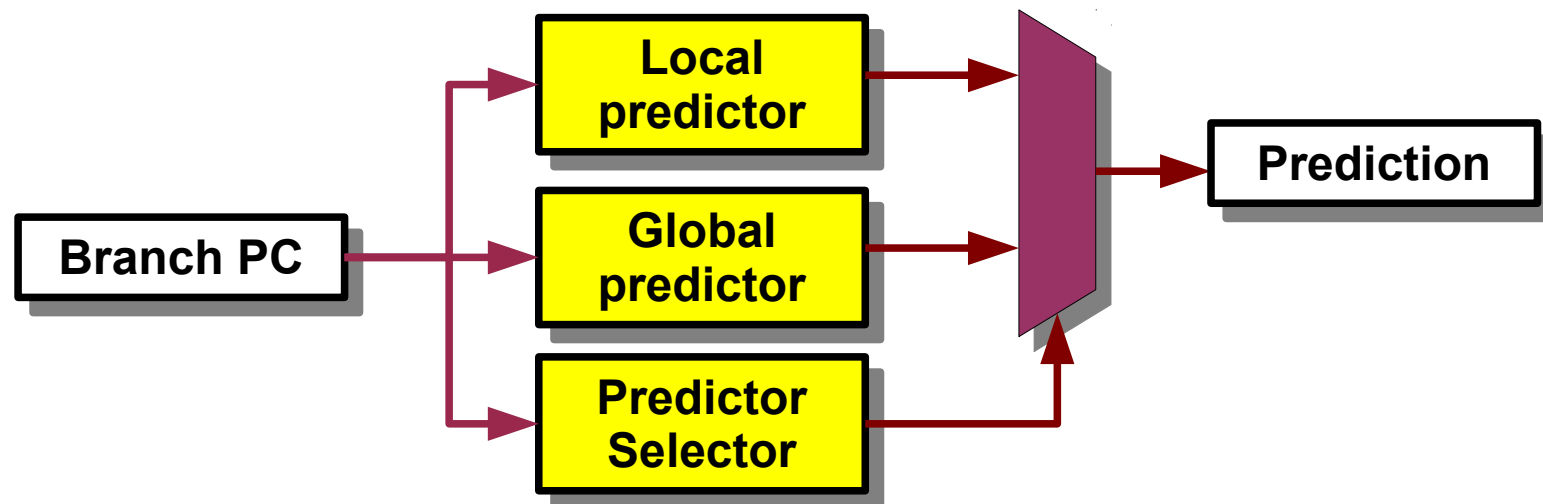
Tournament Predictors

Tournament Predictors



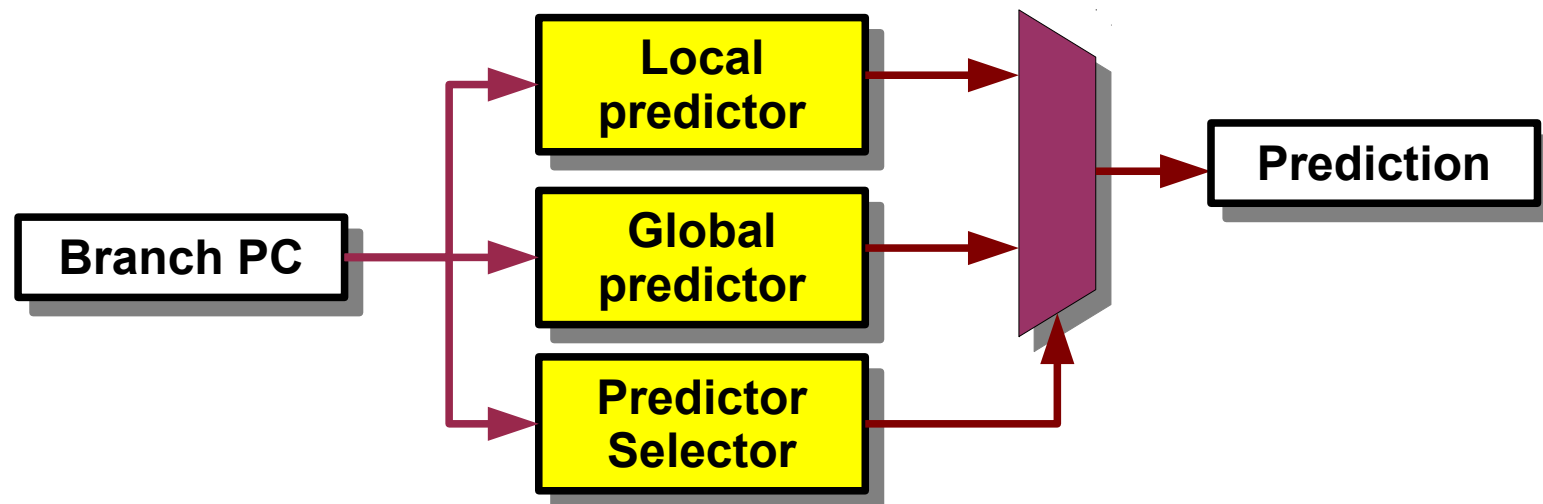
Tournament Predictors

- Use multiple predictors: Global, local or mix

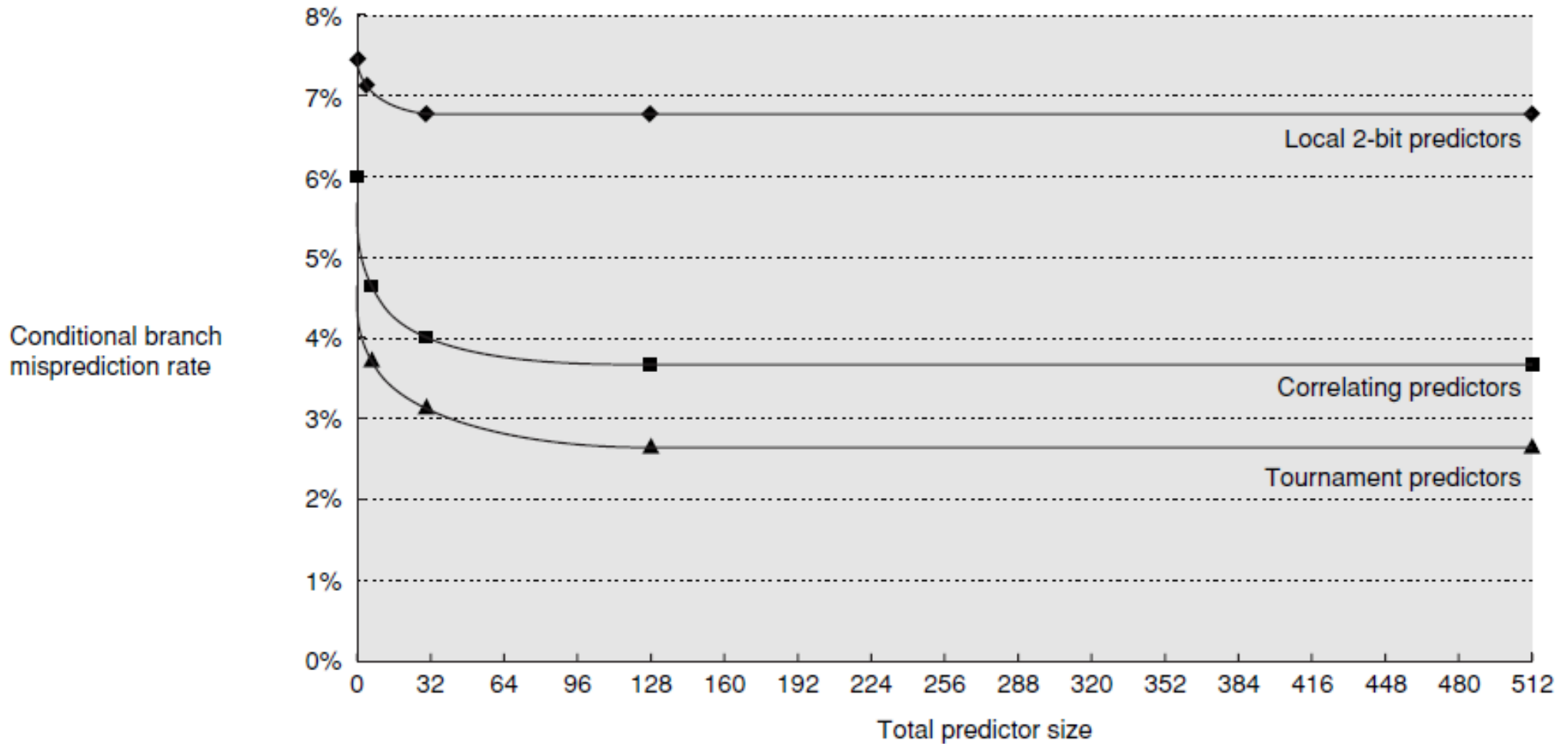


Tournament Predictors

- Use multiple predictors: Global, local or mix
- Combine them with a selector
 - 2 bit saturating counter to select the right predictor for the branch (global vs. local)



Tournament Predictors



Outline

- Pipeline, Pipelined datapath
- Dependences, Hazards
 - Structural, Data - Stalling, Forwarding
- Control Hazards
- Branch prediction