Lecture 4

Speculative Instruction Execution

- **Dynamically scheduled pipelines (Ch. 3.4)**

→ ✓ Speculative execution (3.4.2)

→ ✓ Dynamic branch prediction (3.4.3)

Putting it Together:

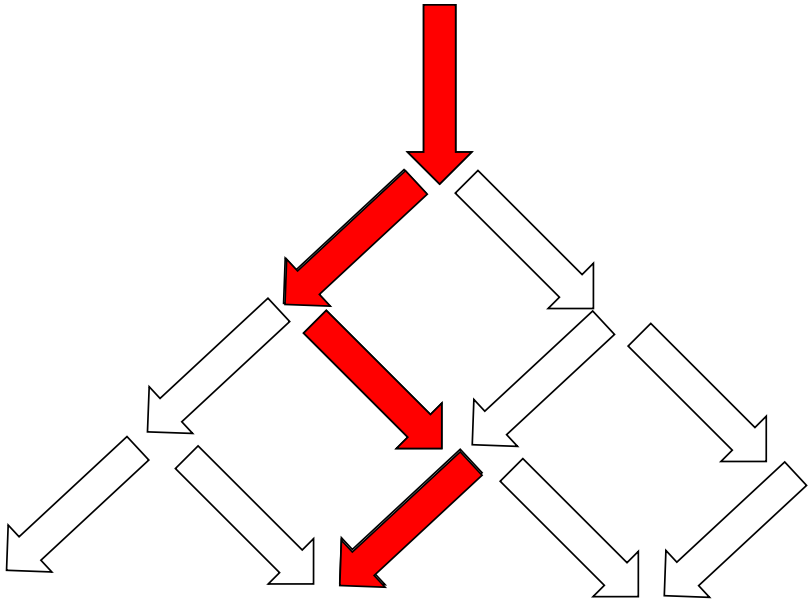
→ ✓ Tomasulo + Branch prediction + Speculation (3.4.4)

→ ✓ Dynamic memory disambiguation (3.4.5)

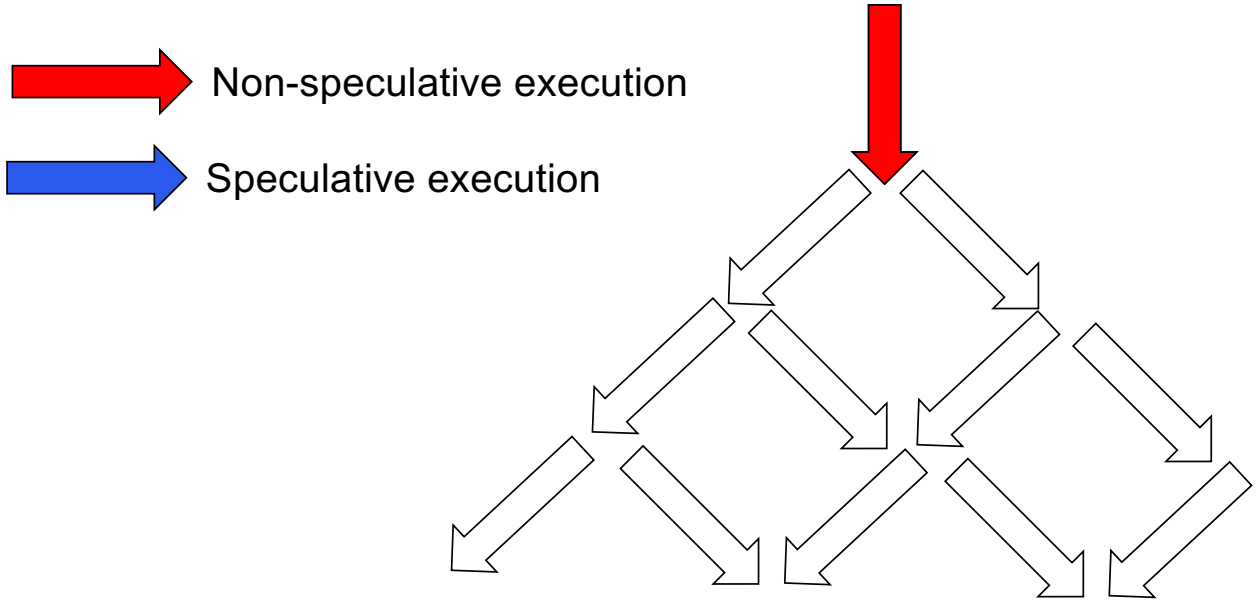
→ ✓ Register renaming techniques (3.4.6)

Speculative Execution

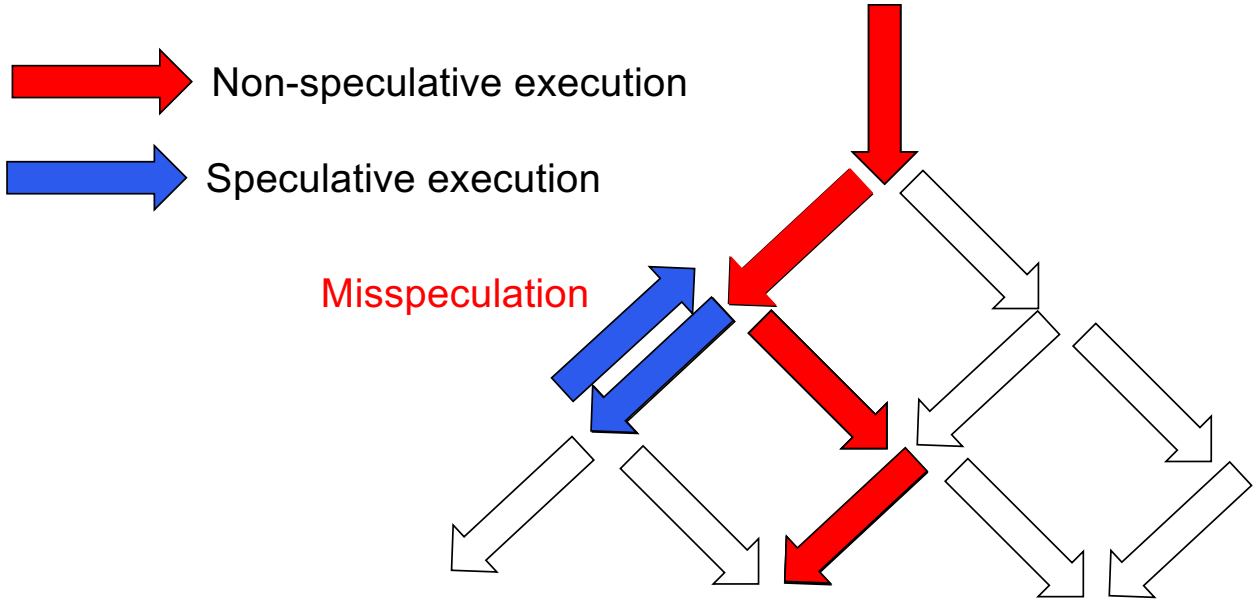
Non-speculative Execution



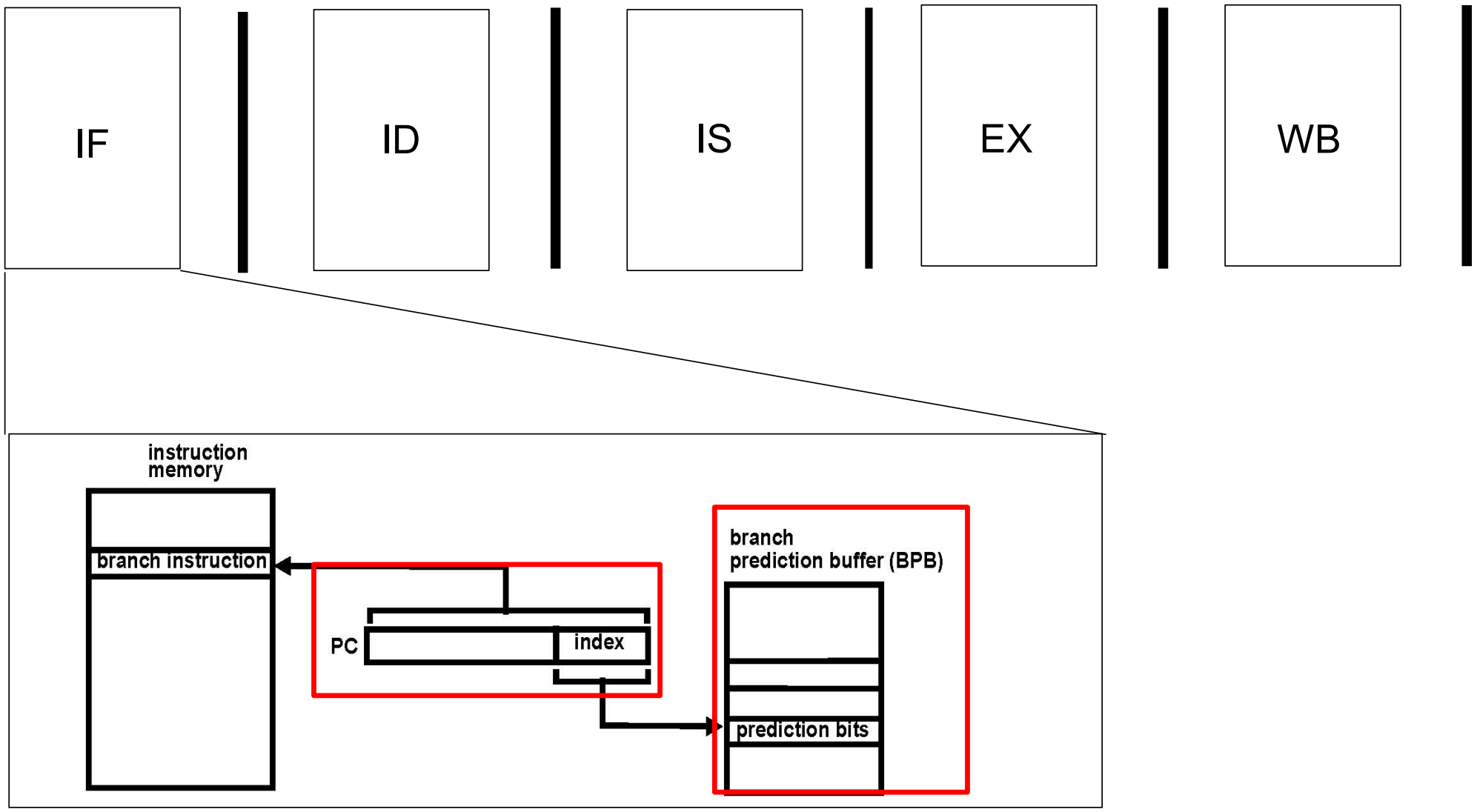
Speculative Execution



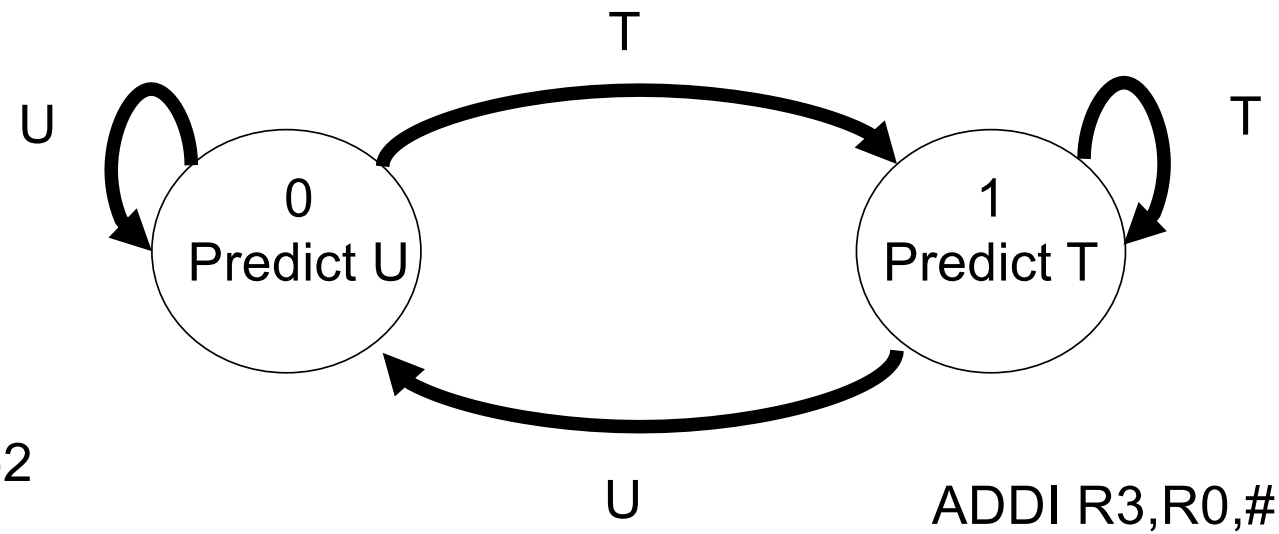
Speculative Execution



Dynamic Branch Prediction


IF
ID
IS
EX
WB
instruction
memory
branch instruction
PC
index
branch
prediction buffer (BPB)
prediction bits

1-Bit Predictor



BNEZ R2,Loop2

Prediction:	U	T	…	T
Outcome:	T	T	…	U

BNEZ R3,Loop1

Prediction:	U	T	…	T
Outcome:	T	T	…	U

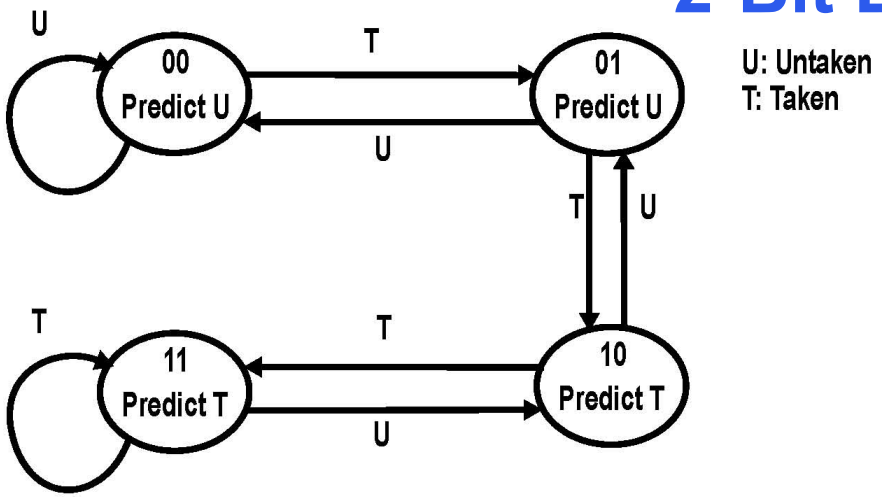
```
        ADDI R3,R0,#100
Loop 1: ADDI R2,R0,#100
        SUBI R3,R3,#1
Loop2:  SUBI R2,R2,#1
        BNEZ R2, Loop2
        BNEZ R3,Loop1
```

Question:
Consider the following history for the outcome of a branch where 1=Taken and 0=UnTaken:
01111101011110
Determine the branch prediction accuracy for a 1-bit predictor assuming it is initially in state=0

Answer:
Prediction: 00111110101111
Outcome: 01111101011110

Correct predictions: 8
Number of outcomes: 14
Prediction accracy: 8/14 = 57%

2-Bit Branch Predictor



U: Untaken
T: Taken

BNEZ R2,Loop1

Prediction:	U	U	T	…	T	T	T
Outcome:	T	T	T	…	U	T	T

BNEZ R3,Loop2

Prediction:	U	U	T	…	T	T	T
Outcome:	T	T	T	…	U	T	T

```
        ADDI R3,R0,#100
Loop 1: ADDI R2,R0,#100
        SUBI R3,R3,#1
Loop2:  SUBI R2,R2,#1
        BNEZ R2, Loop2
        BNEZ R3,Loop1
```

Question:
How many times does a 2-bit predictor mispredict for the following loop?

```
for (i=0; i<100; i++);
```

Answer:
Three times – twice at the entry and once at the exit of the loop. However, on the second invocation it will only mispredict at the exit.

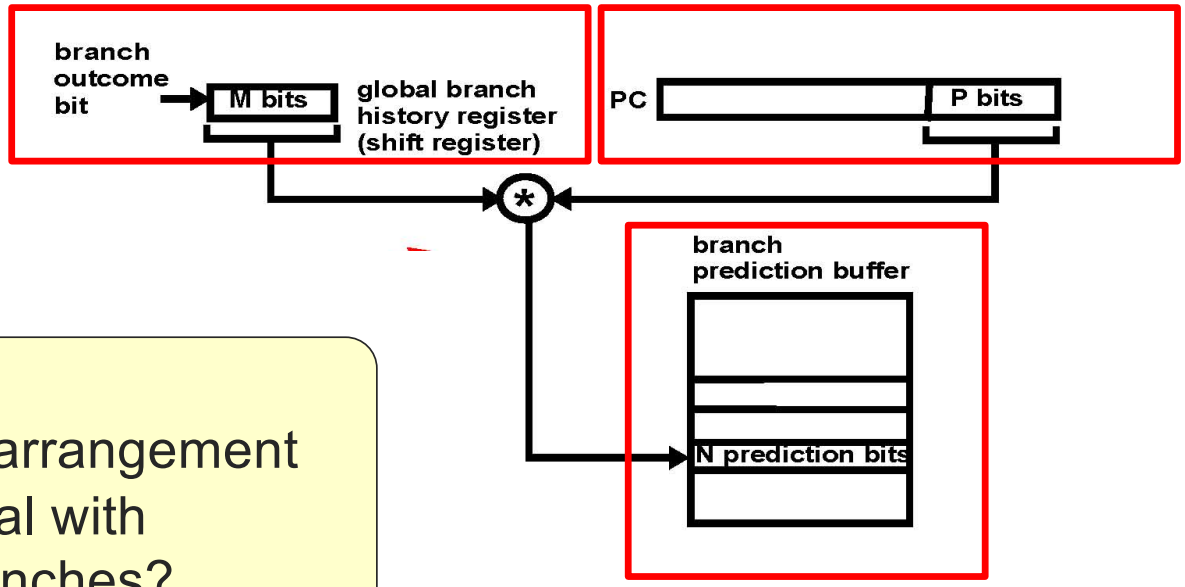
Correlating Branch Predictors

```
if (a==2) then a:=0;
if (b==2) then b:=0;
if (a!=b) then ---
```

Question:
What can we say about the outcome of branches if
a) a==3 and b==1?
b) a==b==2?

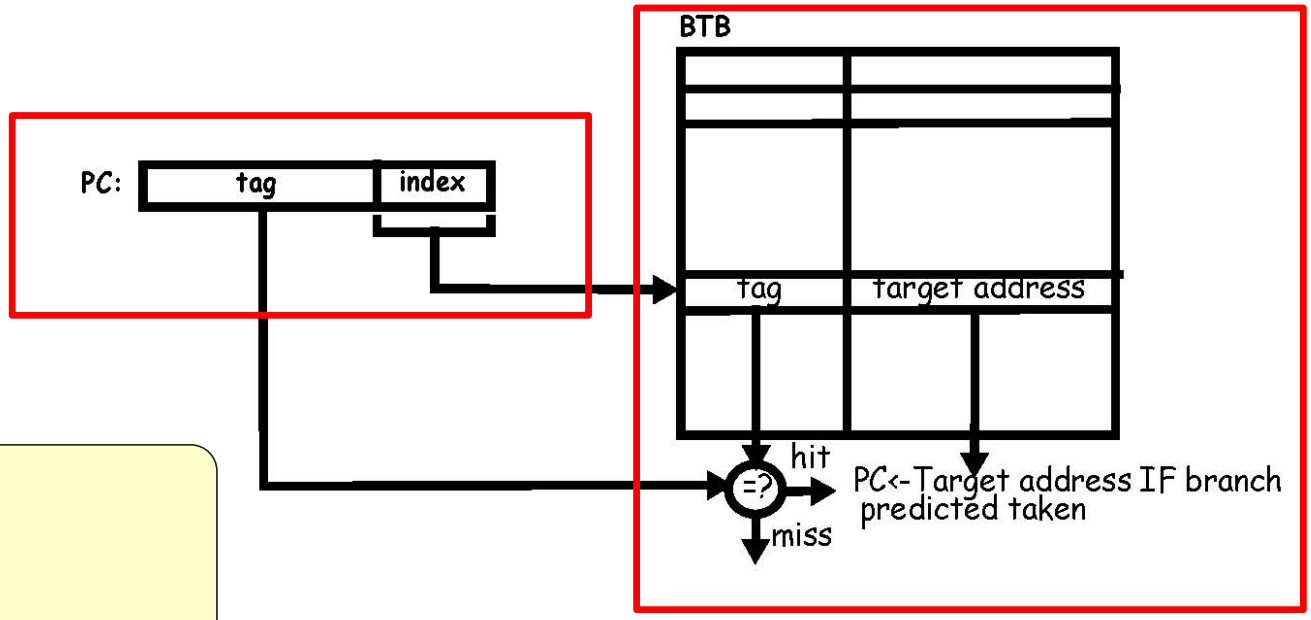
Answer:
a) All branches are uncorrelated
b) All branches are correlated

(M,N) Branch Prediction Buffer



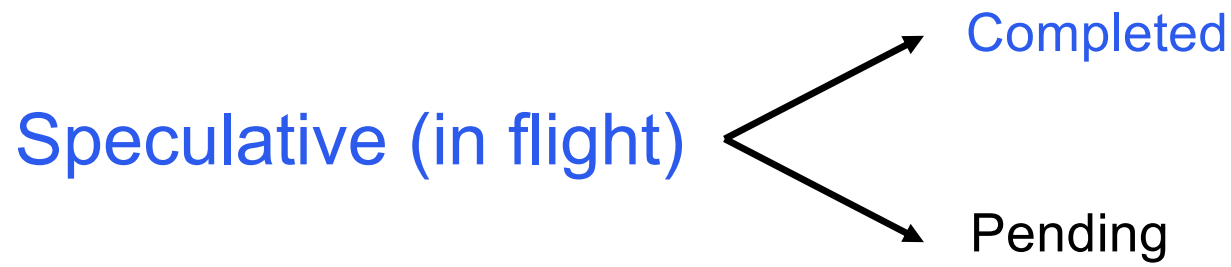
Discussion:
How can this arrangement be used to deal with correlated branches?

Branch Target Buffer (BTB)



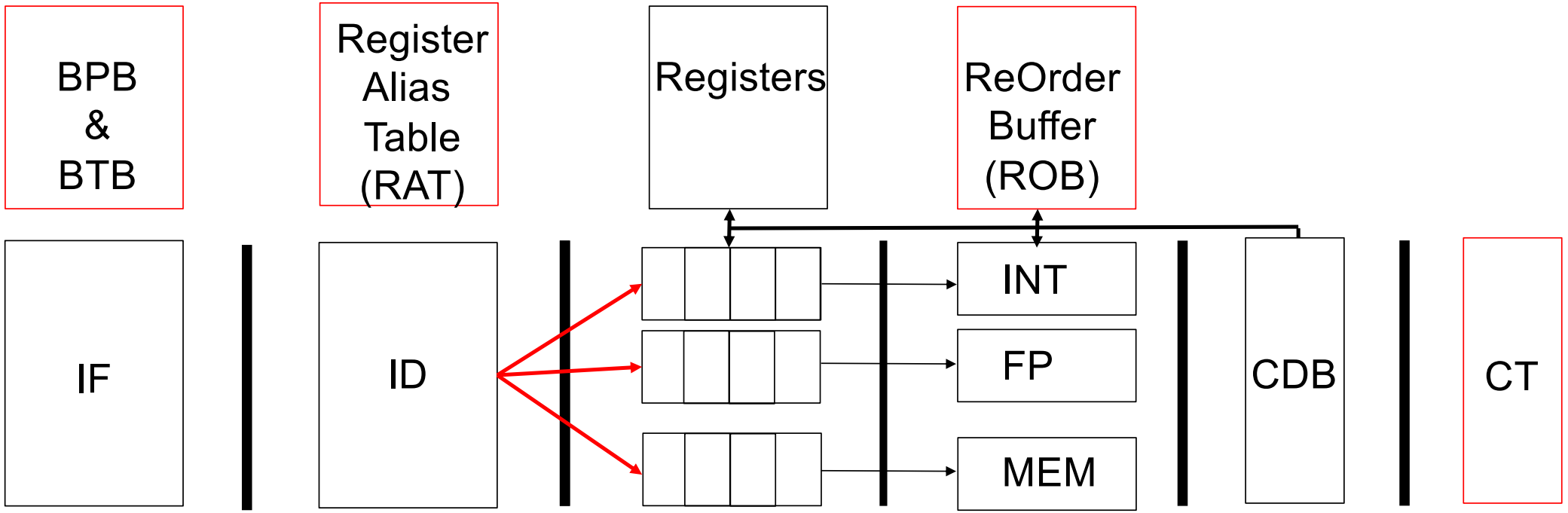
Discussion:
How is a miss handled?

Two Execution Modes



Committed (non-speculative)

Question:
What is the difference between a **completed** and a **committed** instruction?


BPB
&
BTB
Register
Alias
Table
(RAT)
Registers
ReOrder
Buffer
(ROB)
IF
ID
INT
FP
MEM
CDB
CT

Register Alias Table (RAT)

Register	Tag	Status
R0	--	Committed
R1	--	Committed
R2	ROB entry	Completed
…	…	…
R31	ROB entry	Completed
F0	ROB entry	Pending
F1	ROB entry	Pending
F2	--	Committed
...	...	…
F31	--	Committed

Register Alias Table (RAT)

Question:
Where can the values of registers R0, R1 and R2 be found given the content of the RAT?

Answer:
Registers R0 and R1 are committed so their values are found in the registerfile. Register R2 is completed, but not committed so the value is in ROB entry 0

Register	Tag	Status
R0	--	Committed
R1	--	Committed
R2	ROB 0	Completed
…	…	…
R31	ROB 1	Completed
F0	ROB 10	Pending
F1	ROB 11	Pending
F2	--	Committed
...	...	…
F31	--	Committed

ReOrder Buffer (ROB)

Entry	Instruction	Dest. reg	Value	Complete
15	BNEZ R3,Loop	--	--	YES
16	L.S F0,0(R1)	F0	1.2E-4	YES
17	L.S F1,0(R2)	F1	5.8E-3	YES
18	ADD.S F2,F1,F0	F2	2.0E-2	YES
19	S.S F2,0(R1)	--	--	NO
20	ADDI R1,R1,#4	R1	N/A	NO
21	ADDI R2,R2,#4	R2	N/A	NO
22	SUBI R3,R3,#1	R3	N/A	NO
23	BNEZ R3,Loop	--	--	--

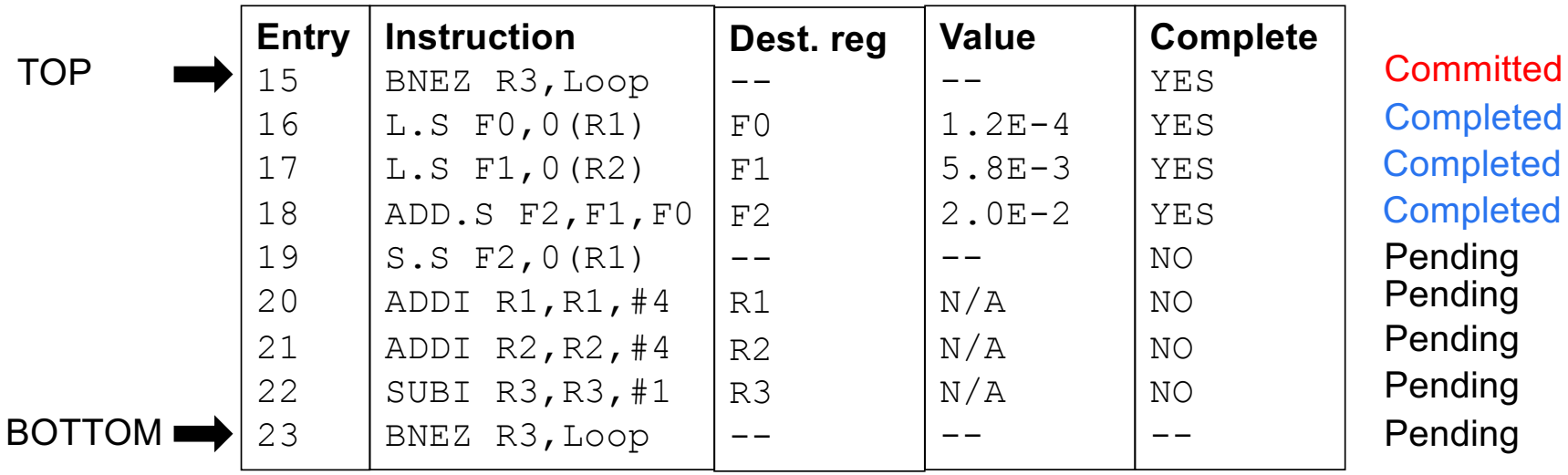
ReOrder Buffer (ROB)

Question:
Where can the values of F2 and R3 be found?

Answer:
F2 is in the reorder buffer but R3 is not available as an instruction is speculatively producing its value. This can be seen as the status is not completed.

Entry	Instruction	Dest. reg	Value	Complete
15	BNEZ R3,Loop	--	--	YES
16	L.S F0,0(R1)	F0	1.2E-4	YES
17	L.S F1,0(R2)	F1	5.8E-3	YES
18	ADD.S F2,F1,F0	F2	2.0E-2	YES
19	S.S F2,0(R1)	--	--	NO
20	ADDI R1,R1,#4	R1	N/A	NO
21	ADDI R2,R2,#4	R2	N/A	NO
22	SUBI R3,R3,#1	R3	N/A	NO
23	BNEZ R3,Loop	--	--	--

ReOrder Buffer (ROB)



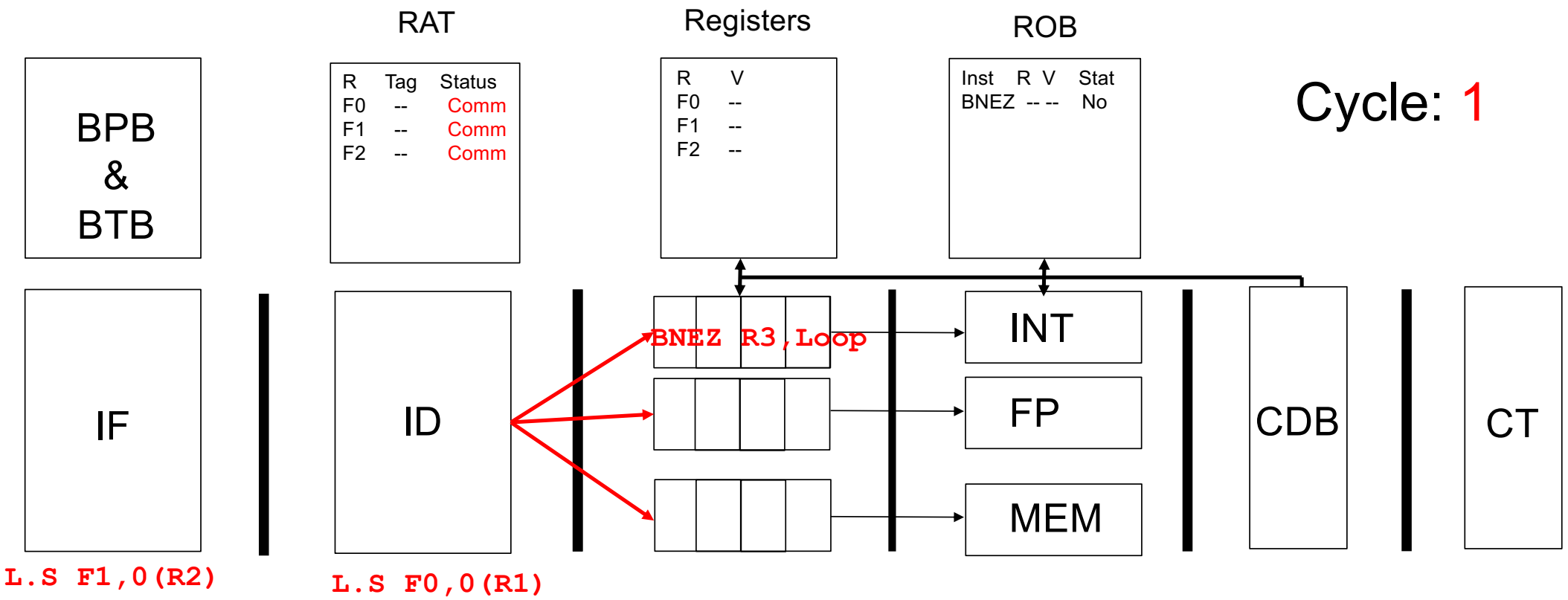
Entry	Instruction	Dest. reg	Value	Complete	
15	BNEZ R3,Loop	--	--	YES	Committed
16	L.S F0,0(R1)	F0	1.2E-4	YES	Completed
17	L.S F1,0(R2)	F1	5.8E-3	YES	Completed
18	ADD.S F2,F1,F0	F2	2.0E-2	YES	Completed
19	S.S F2,0(R1)	--	--	NO	Pending
20	ADDI R1,R1,#4	R1	N/A	NO	Pending
21	ADDI R2,R2,#4	R2	N/A	NO	Pending
22	SUBI R3,R3,#1	R3	N/A	NO	Pending
23	BNEZ R3,Loop	--	--	--	Pending

Tag = Reorder buffer entry

ReOrder Buffer (ROB)

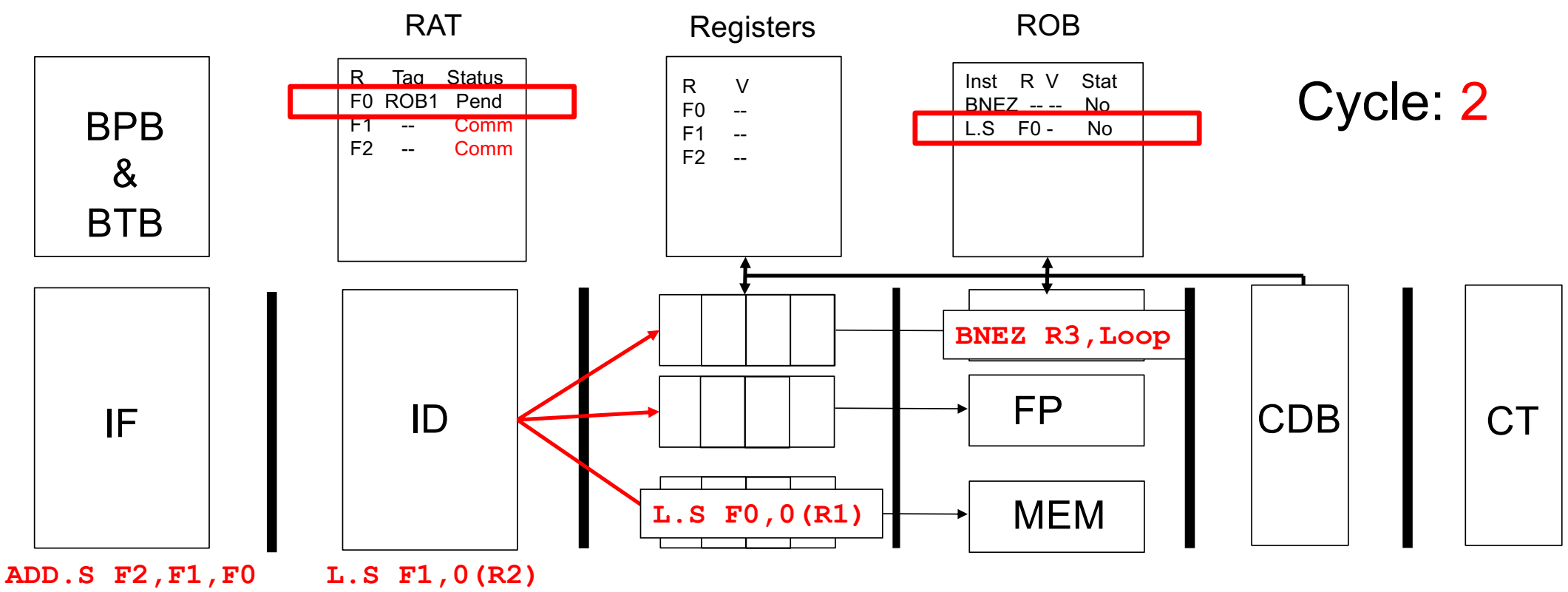
	Entry	Instruction	Dest. reg	Value	Complete	
TOP →	16	L.S F0,0(R1)	F0	1.2E-4	YES	Committed
	17	L.S F1,0(R2)	F1	5.8E-3	YES	Completed
	18	ADD.S F2,F1,F0	F2	2.0E-2	YES	Completed
	19	S.S F2,0(R1)	--	--	YES	Completed
	20	ADDI R1,R1,#4	R1	N/A	NO	Pending
	21	ADDI R2,R2,#4	R2	N/A	NO	Pending
	22	SUBI R3,R3,#1	R3	N/A	NO	Pending
	23	BNEZ R3,Loop	--	N/A	NO	Pending
BOTTOM →	24	L.S F0,0(R1)	F0	N/A	NO	Pending

Speculative Execution

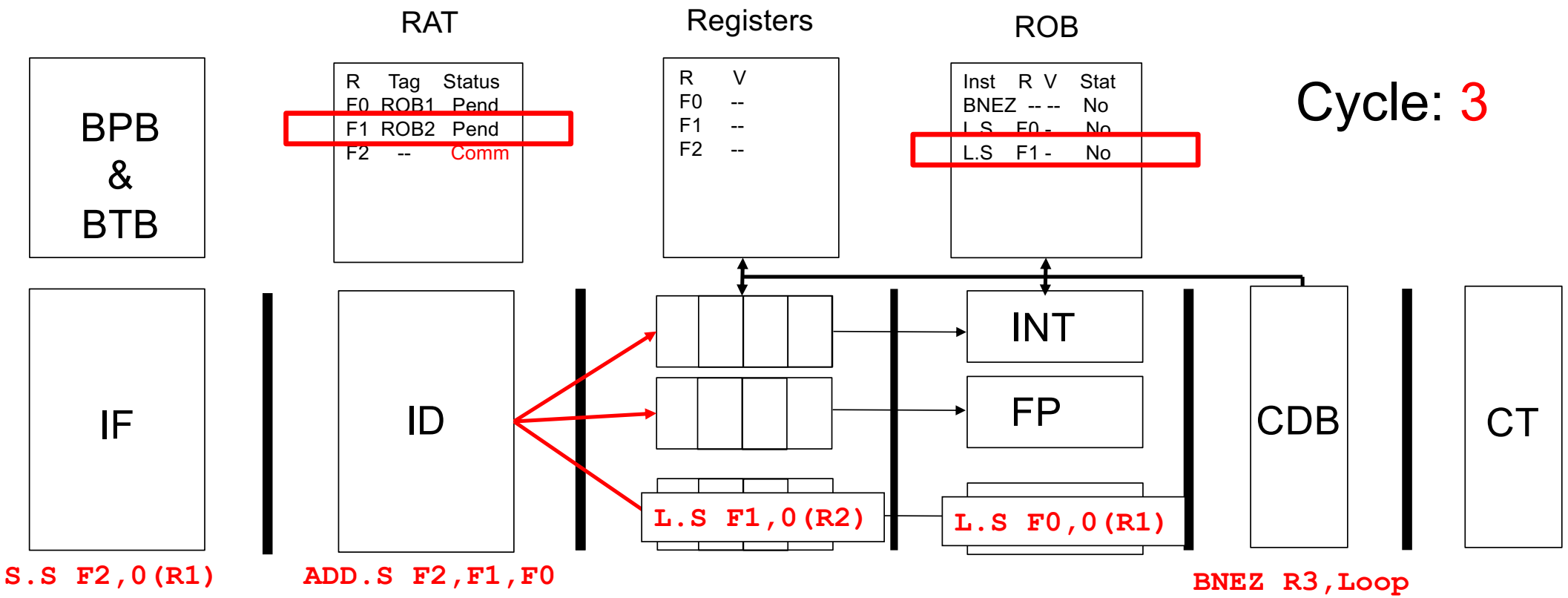


Here, the branch instruction is speculatively executed

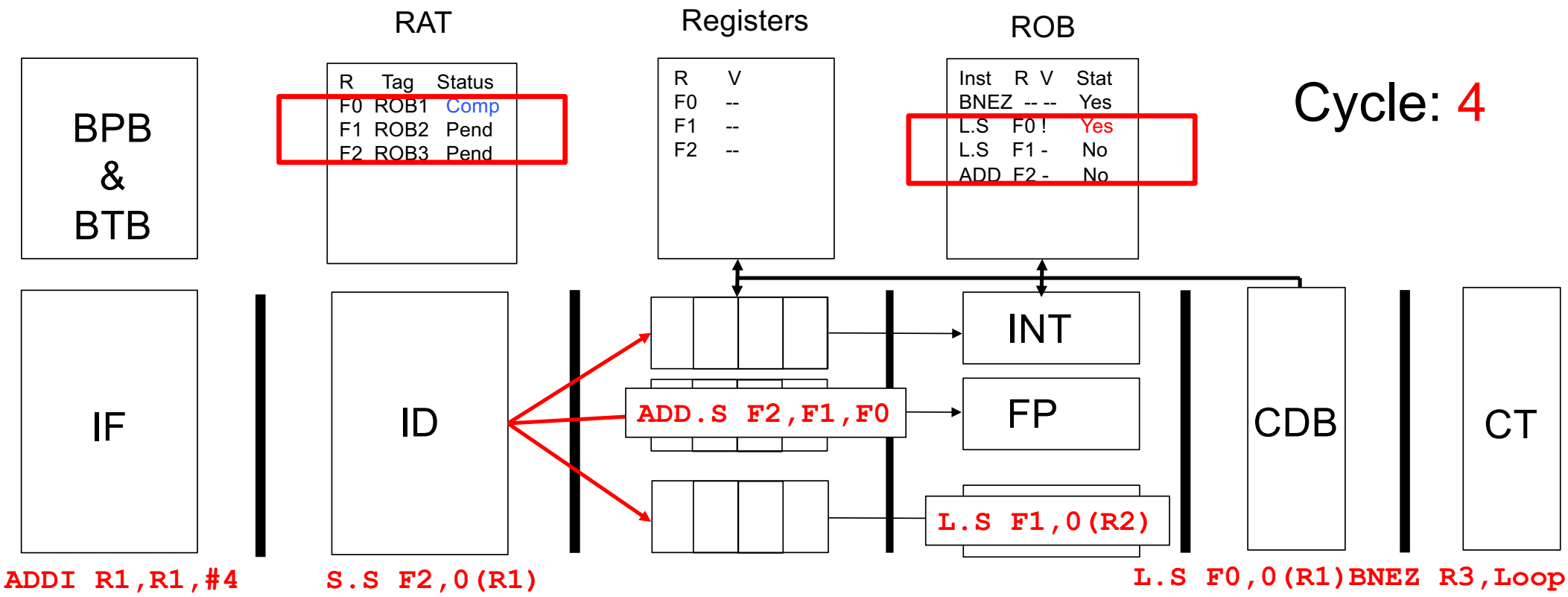
```
L.S F0,0(R1)
L.S F1,0(R2)
ADD.S F2,F1,F0
S.S F2,0(R1)
ADDI R1,R1,#4
ADDI R2,R2,#4
SUBI R3,R3,#1
BNEZ R3,Loop
```



L.S F0,0(R1)
L.S F1,0(R2)
ADD.S F2,F1,F0
S.S F2,0(R1)
ADDI R1,R1,#4
ADDI R2,R2,#4
SUBI R3,R3,#1
BNEZ R3,Loop

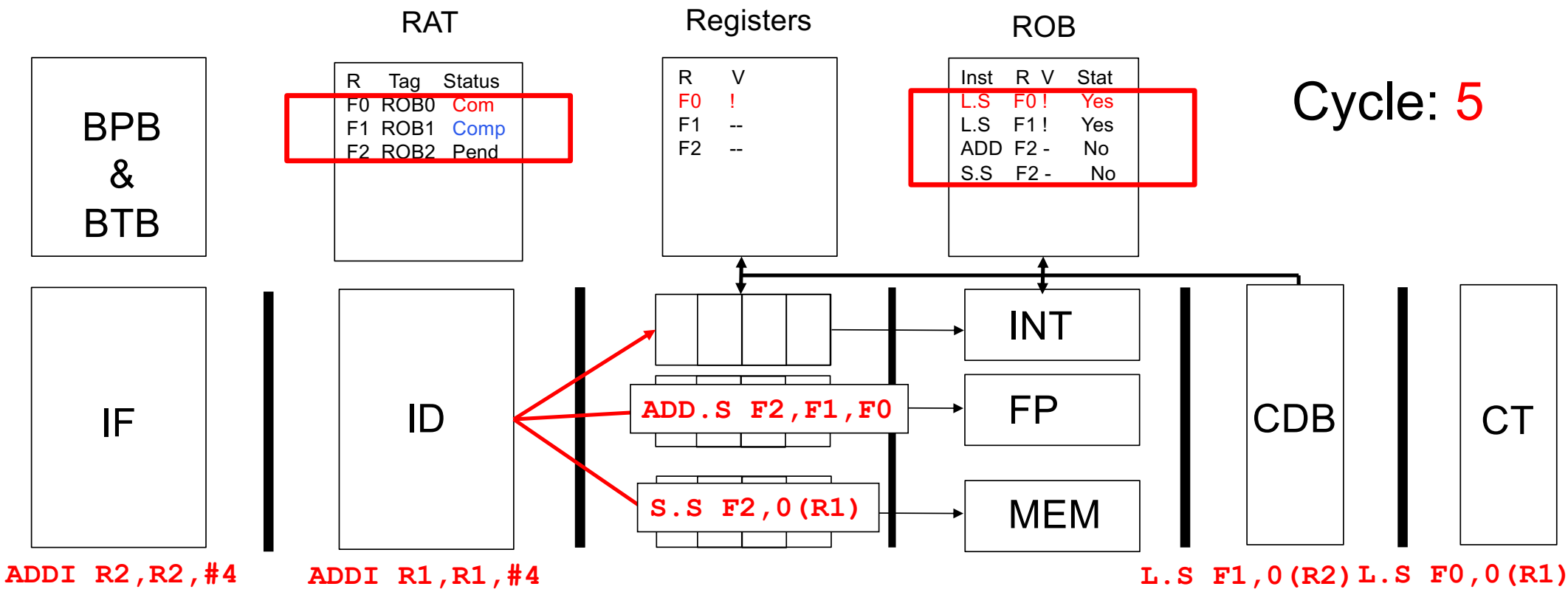



```
L.S F0,0(R1)
L.S F1,0(R2)
ADD.S F2,F1,F0
S.S F2,0(R1)
ADDI R1,R1,#4
ADDI R2,R2,#4
SUBI R3,R3,#1
BNEZ R3,Loop
```

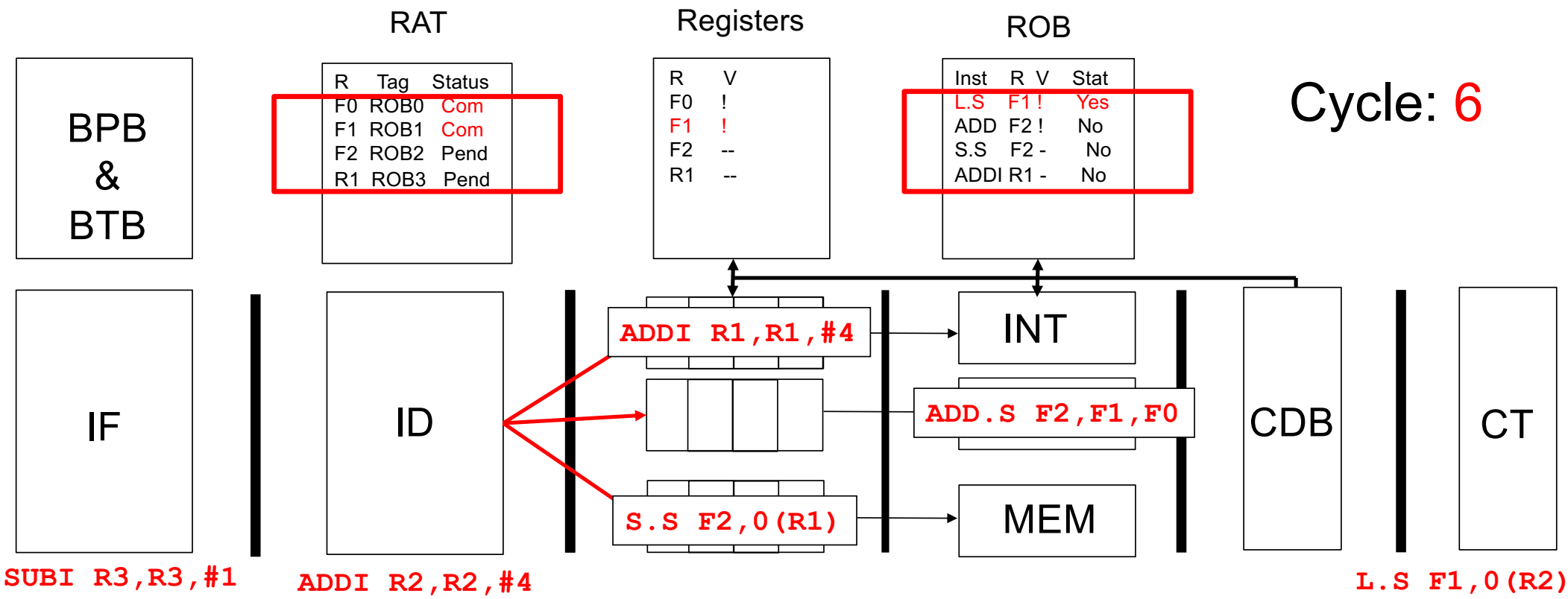


Here, the branch instruction commits

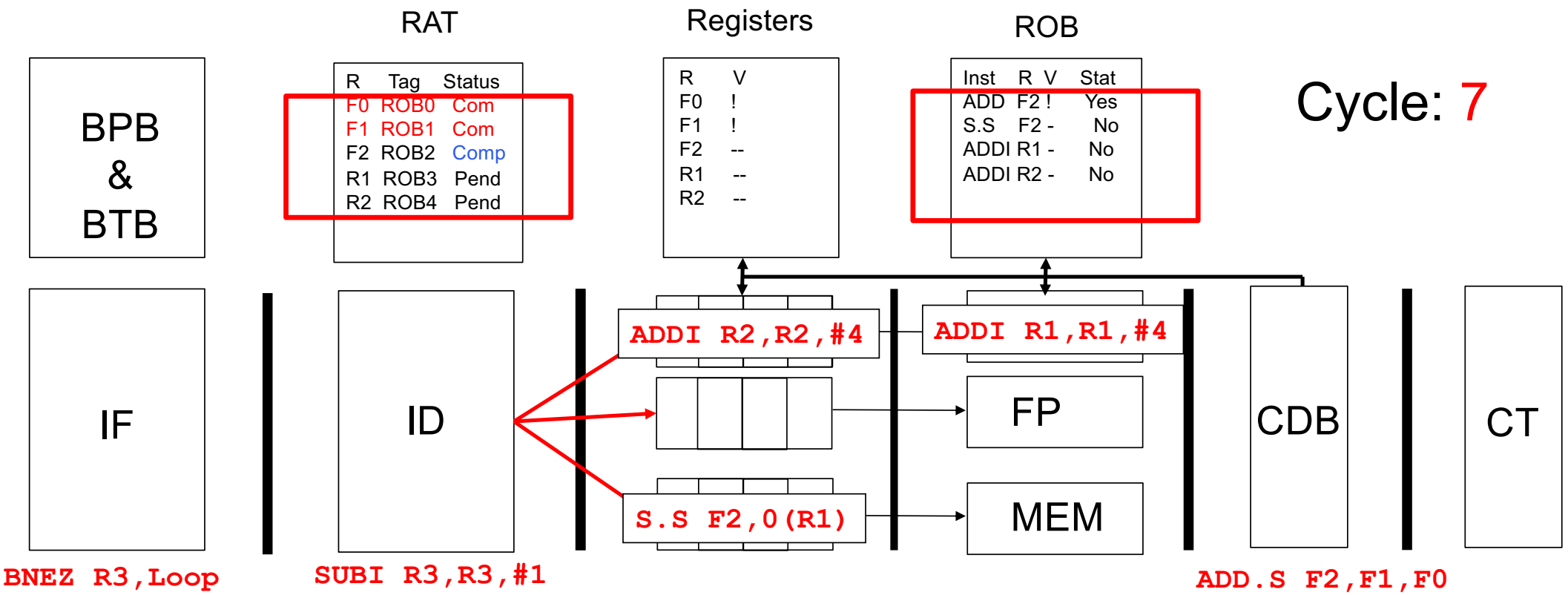
```
L.S  F0,0(R1)
L.S  F1,0(R2)
ADD.S F2,F1,F0
S.S  F2,0(R1)
ADDI R1,R1,#4
ADDI R2,R2,#4
SUBI R3,R3,#1
BNEZ R3,Loop
```



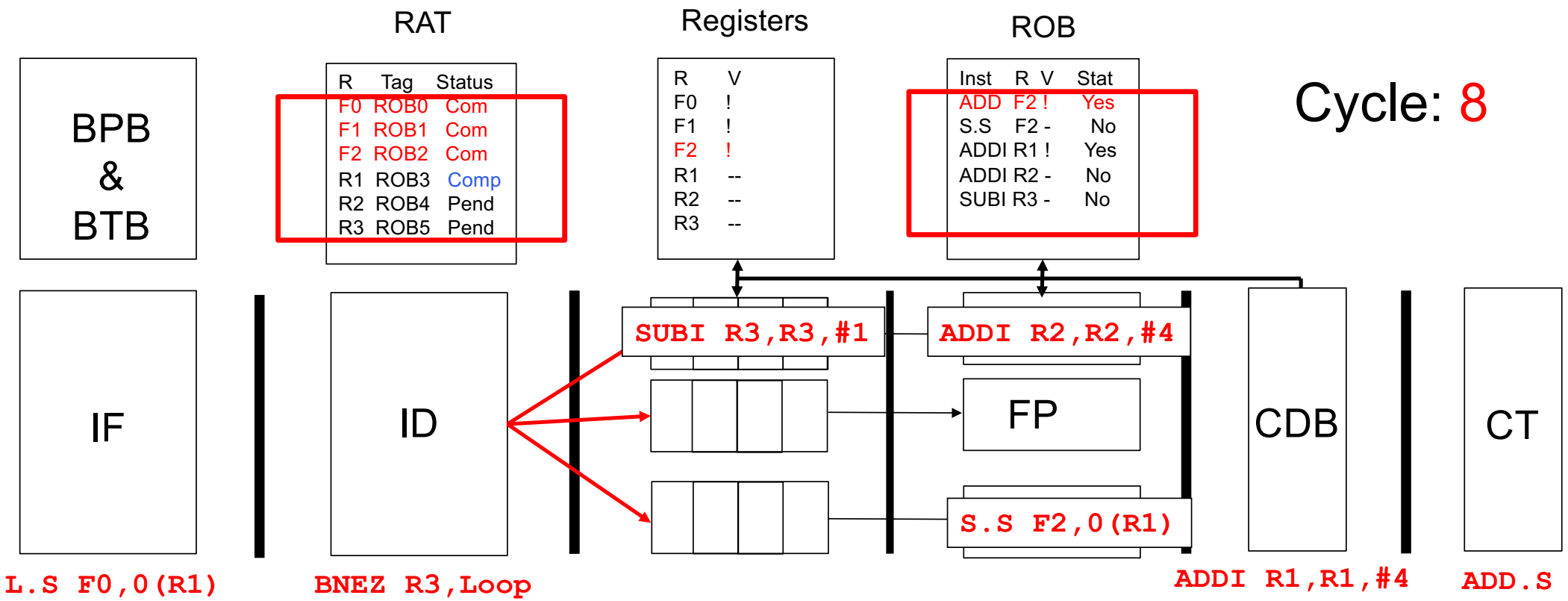

```
L.S F0,0(R1)
L.S F1,0(R2)
ADD.S F2,F1,F0
S.S F2,0(R1)
ADDI R1,R1,#4
ADDI R2,R2,#4
SUBI R3,R3,#1
BNEZ R3,Loop
```



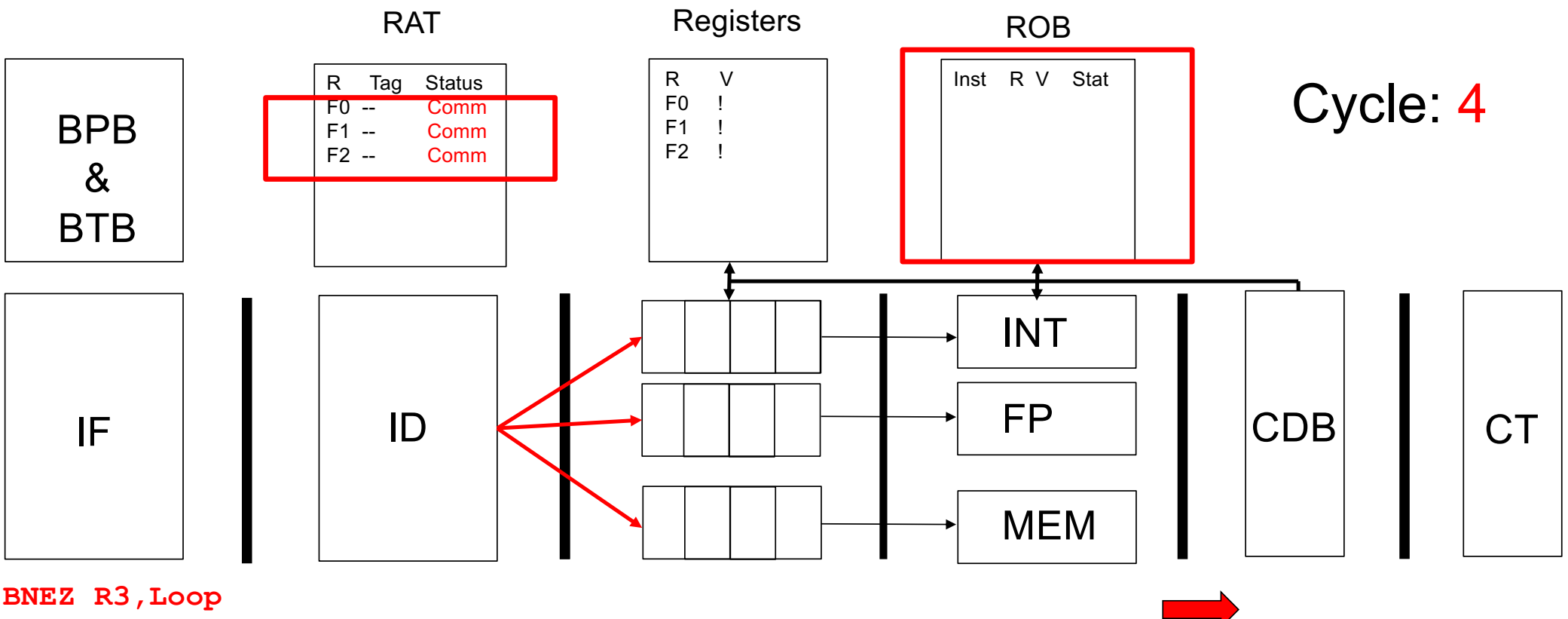

```
L.S F0,0(R1)
L.S F1,0(R2)
ADD.S F2,F1,F0
S.S F2,0(R1)
ADDI R1,R1,#4
ADDI R2,R2,#4
SUBI R3,R3,#1
BNEZ R3,Loop
```




```
L.S F0,0(R1)
L.S F1,0(R2)
ADD.S F2,F1,F0
S.S F2,0(R1)
ADDI R1,R1,#4
ADDI R2,R2,#4
SUBI R3,R3,#1
BNEZ R3,Loop
```




```
L.S F0,0(R1)
L.S F1,0(R2)
ADD.S F2,F1,F0
S.S F2,0(R1)
ADDI R1,R1,#4
ADDI R2,R2,#4
SUBI R3,R3,#1
BNEZ R3,Loop
```




```
L.S F0,0(R1)
L.S F1,0(R2)
ADD.S F2,F1,F0
S.S F2,0(R1)
ADDI R1,R1,#4
ADDI R2,R2,#4
SUBI R3,R3,#1
BNEZ R3,Loop
```

Mispredicted Branch

Question:
Consider the program below and determine how many cycles it takes to execute two iterations on the speculative pipeline assuming that the branch is correctly predicted.

```
Loop    L.S F0,0(R1)
        ADD.S F2,F1,F0
        S.S F2,0(R1)
        ADDI R1,R1,#4
        SUBI R3,R3,#1
        BNEZ R3,Loop
```

Answer:
18 cycles

```
Loop I1:L.S F0,0(R1)
     I2:ADD.S F2,F1,F0
     I3:S.S F2,0(R1)
     I4:ADDI R1,R1,#4
     I5:SUBI R3,R3,#1
     I6:BNEZ R3,Loop
```

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24
I1,1	IF	ID	IS	EX	WB																			
I2,1		IF	ID	IS	EX	EX	EX	EX	EX	WB														
I3,1			IF	ID	IS	IS	IS	IS	IS	EX	WB													
I4,1				IF	ID	IS	EX	WB																
I5,1					IF	ID	IS	EX	WB															
I6,1						IF	ID	IS	EX	WB	WB	WB												
I1,2							IF	ID	IS	EX	WB	WB	WB											
I2,2								IF	ID	IS	EX	EX	EX	EX	EX	WB								
I3,2									IF	ID	IS	IS	IS	IS	IS	EX	WB							
I4,2										IF	ID	IS	EX	WB										
I5,2											IF	ID	IS	EX	WB									
I6,2												IF	ID	IS	EX	WB	WB	WB						