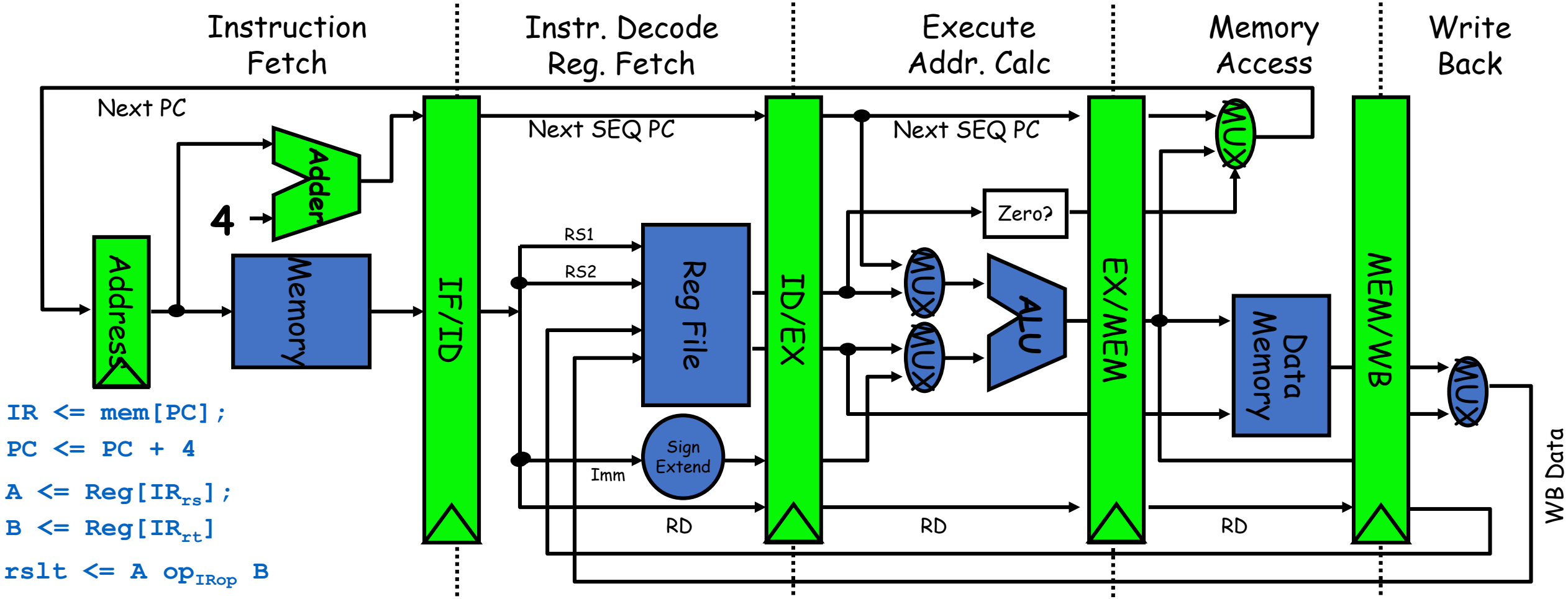
Lecture-6
(10K feet view: Processor Core)
CS422-Spring 2019

Biswa@CSE-IITK

Simple 5-stage Pipeline




```
IR <= mem[PC];
PC <= PC + 4
A <= Reg[IR_rs];
B <= Reg[IR_rt]
rslt <= A op_IRop B
WB <= rslt
Reg[IR_rd] <= WB
```

- Data stationary control
  - local decode for each instruction phase / pipeline stage

Hazards

- Limits to pipelining: Hazards prevent next instruction from executing during its designated clock cycle
  - Structural hazards: HW cannot support this combination of instructions (single person to fold and put clothes away)
  - Data hazards: Instruction depends on result of prior instruction still in the pipeline (missing sock)
  - Control hazards: Caused by delay between the fetching of instructions and decisions about changes in control flow (branches and jumps).

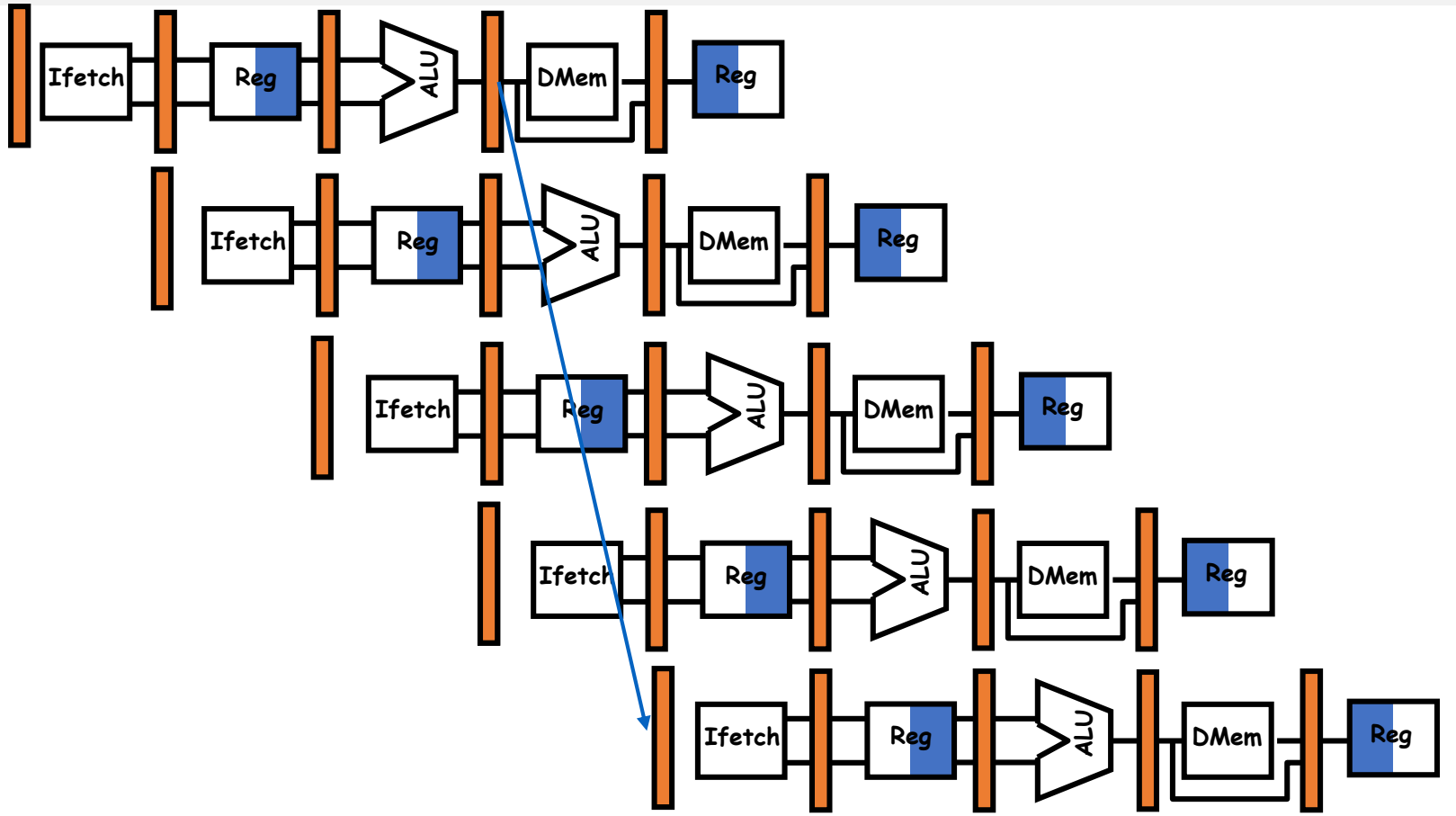
Data Hazards

- Read After Write (RAW)
  $Instr_J$ tries to read operand before $Instr_I$ writes it

- WAR and WAW too

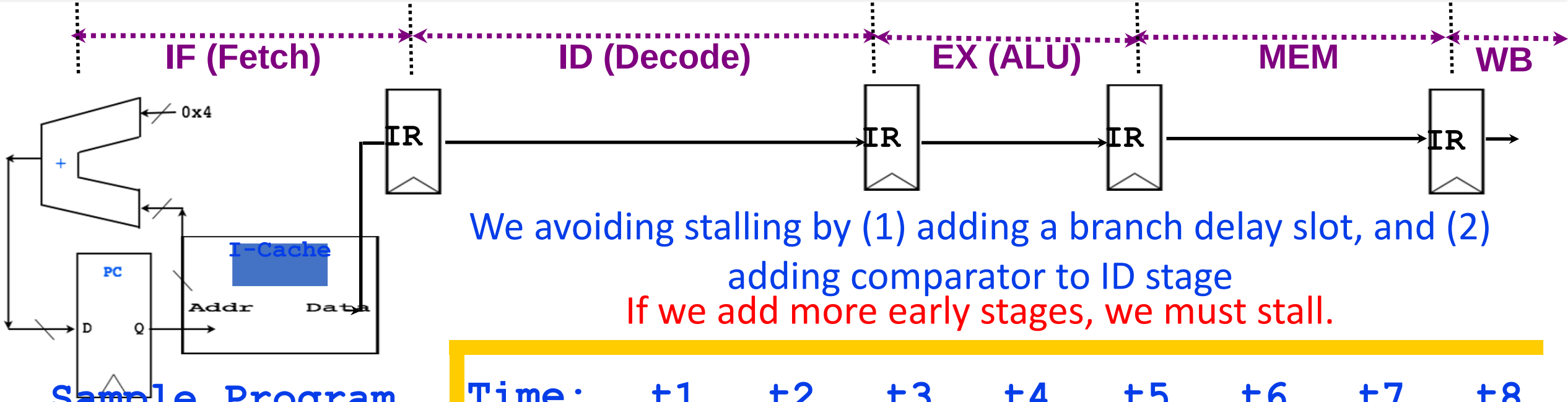
Control Hazard

```
10: beq r1,r3,36

14: and r2,r3,r5

18: or  r6,r1,r7

22: add r8,r1,r9

36: xor r10,r1,r11
```



What do you do with the 3 instructions in between?
How do you do it?
Where is the "commit"?

Remember This



We avoiding stalling by (1) adding a branch delay slot, and (2) adding comparator to ID stage

If we add more early stages, we must stall.

Sample Program (ISA w/o branch delay slot)

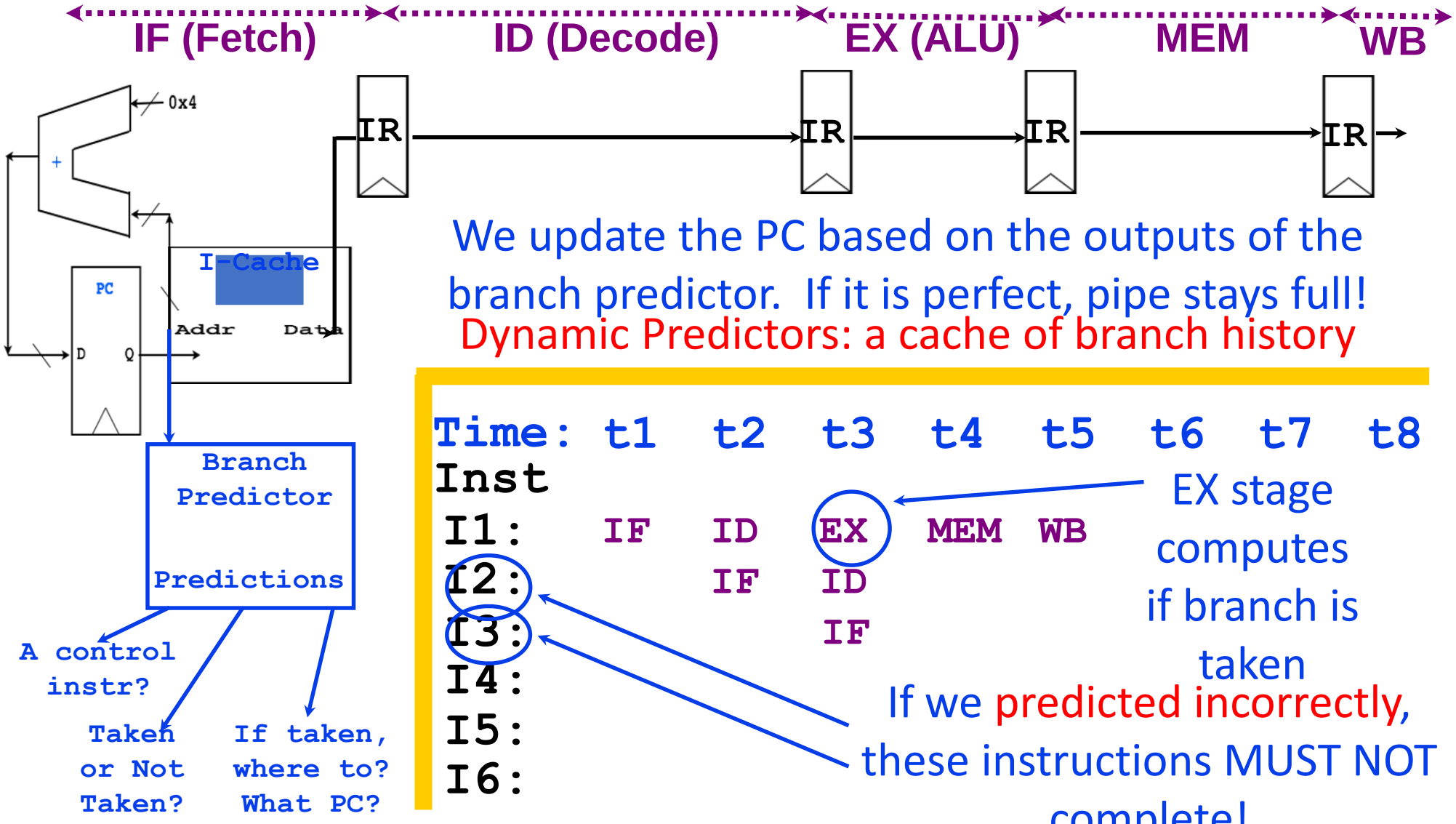
```
I1:  BEQ R4,R3,25
I2:  AND R6,R5,R4
I3:  SUB R1,R9,R8
```

Time:	t1	t2	t3	t4	t5	t6	t7	t8
Inst								
I1:	IF	ID	EX	MEM	WB			
I2:		IF	ID					
I3:			IF					
I4:								
I5:								
I6:								

EX stage computes if branch is taken

If branch is taken, these instructions MUST NOT complete!

Welcome to Branch Prediction



We update the PC based on the outputs of the branch predictor. If it is perfect, pipe stays full!

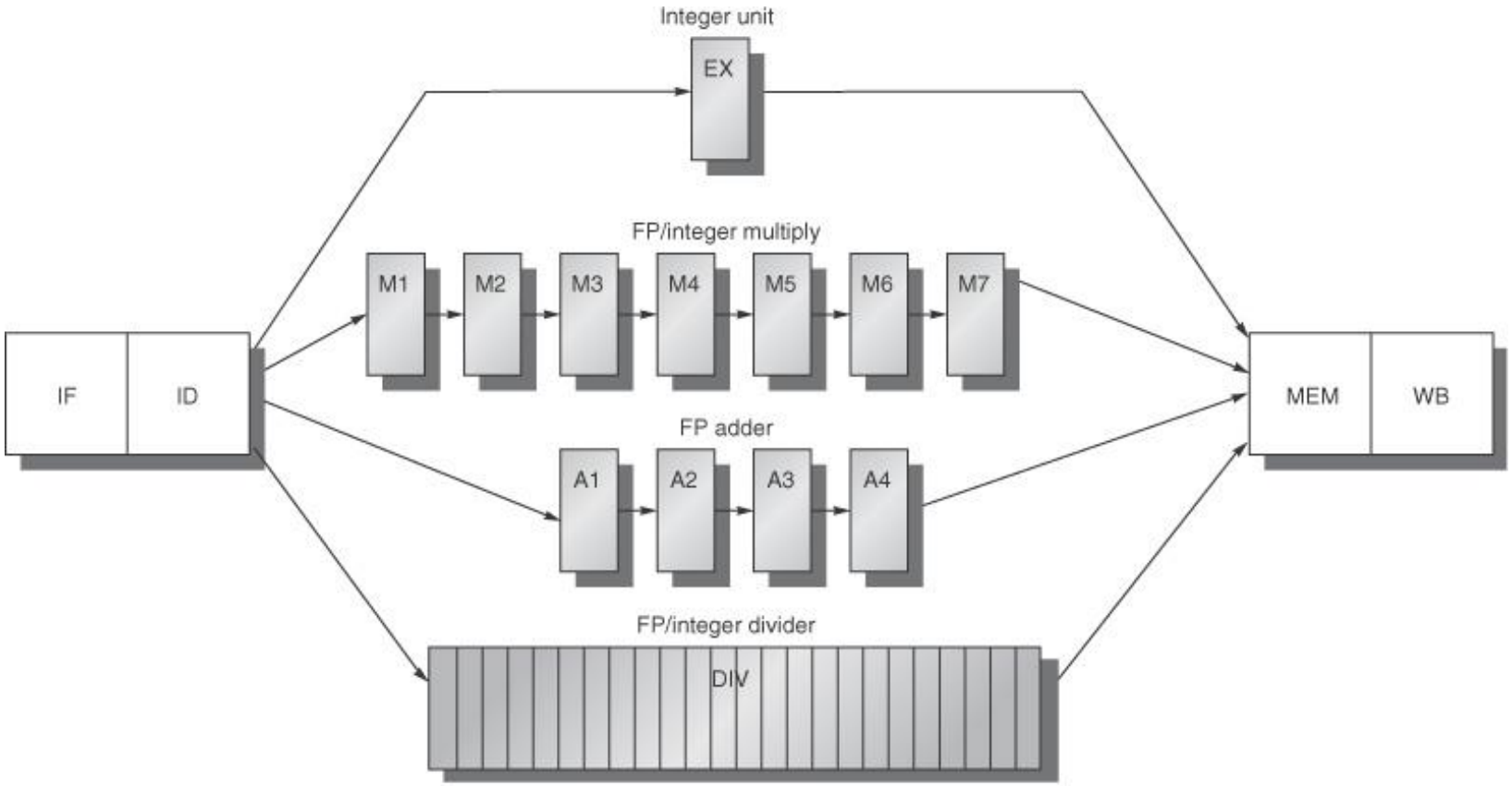
Dynamic Predictors: a cache of branch history

Time:	t1	t2	t3	t4	t5	t6	t7	t8
Inst								
I1:	IF	ID	EX	MEM	WB			
I2:		IF	ID					
I3:			IF					
I4:								
I5:								
I6:								

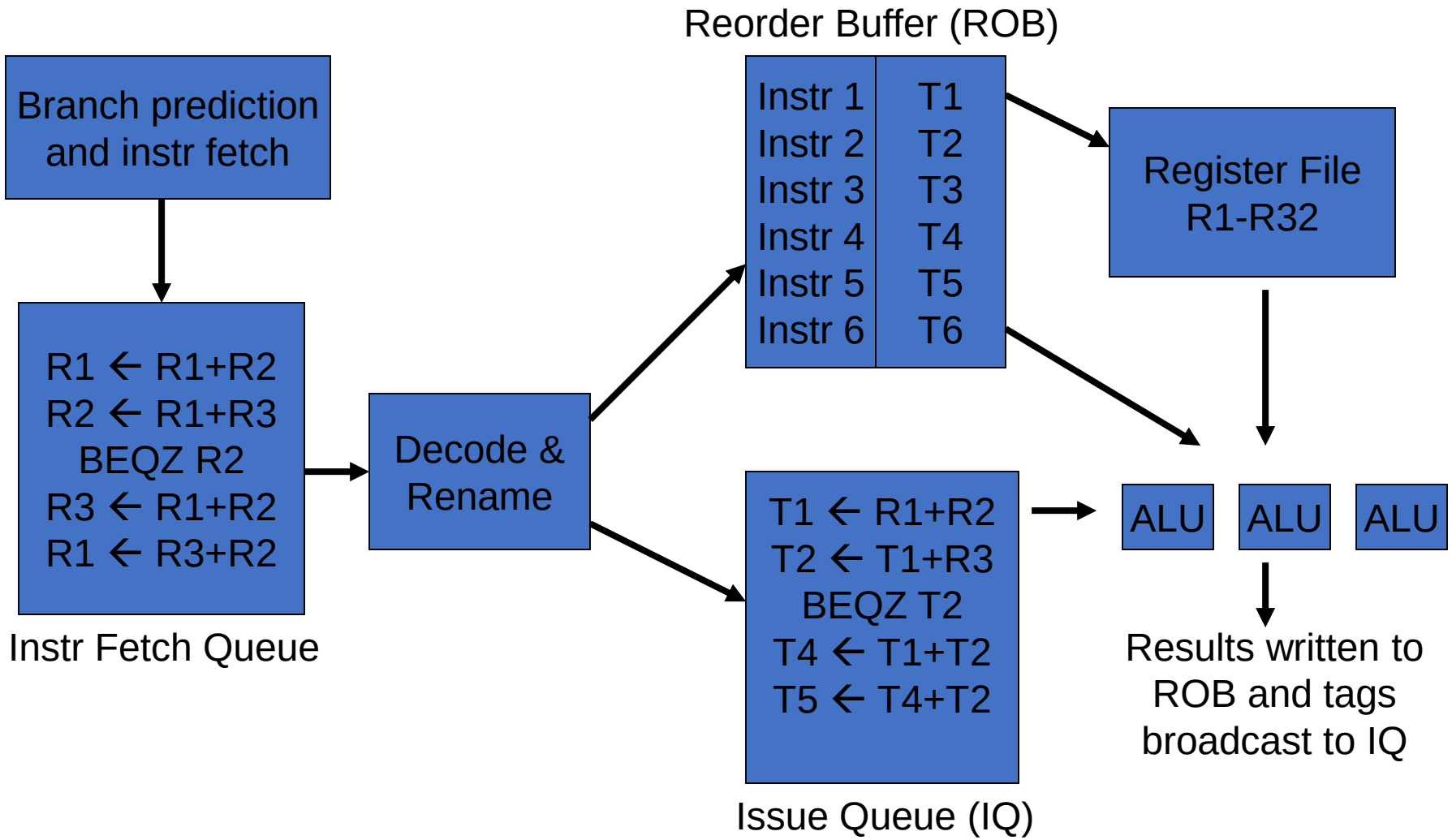
EX stage computes if branch is taken

If we predicted incorrectly, these instructions MUST NOT complete!

Multi-Cycle Instructions



Out-of-Order (O3) World



O3 + LSQ + Multi Issue (SuperScalar)