Exercise 4: Given a pipelined processor with 5 stages and the following features:

- The Register File can be read and written in the same cycle.
- It has HW for detecting hazards and stall in Decode stage if necessary.
- It has the forwarding technique.
- Branches are resolved at Decode using stalls.
- Structural hazards at the MEM stage are detected and resolved via stalling in the last stage of execution.
- WAW hazards are resolved by avoiding the write of the first instruction.

The following program is executed in this processor:

```
LD      F10,0(R1)
MULD    F4,F0,F10
LD      F12,0(R2)
ADDD F2,F12,F4
LD      F4,8(R1)
MULD    F12,F4,F12
LD      F12,16(R1)
```

Functional Unit	Number	Latency	Pipelined
FP ADD	**1**	**2**	**YES**
FP MUL	**1**	**5**	**YES**
Int ALU	**1**	**1**	**NO**

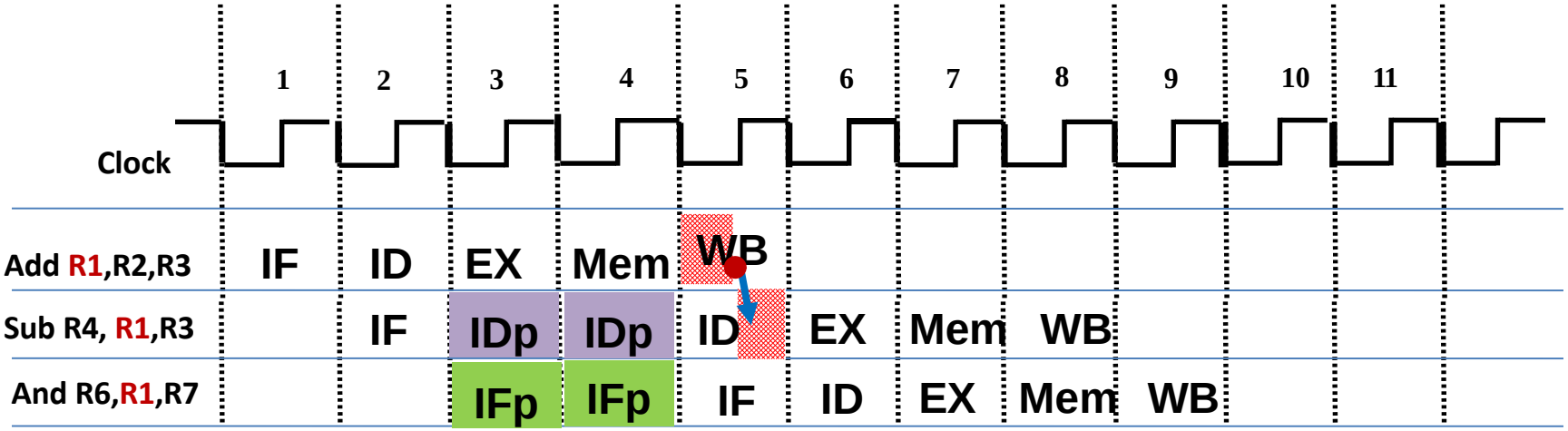
a)Represent the instruction/time diagram for the program and highlight the forwardings that take place.

b)Calculate the CPI.

4. Given a pipelined processor with 5 stages and the following features:

The Register File can be read and written in the same cycle.

We modify the Register Bank: writes are completed at mid cycle, on the falling edge of the clock, and half cycle is enough to read the updated value



4. Given a pipelined processor with 5 stages and the following features:

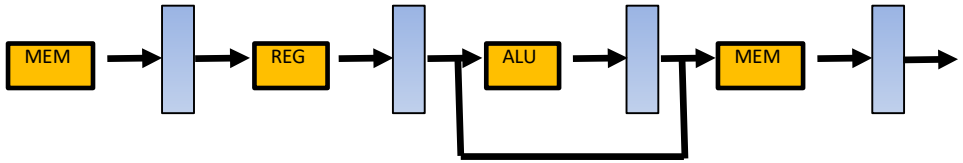
It has the forwarding technique.

Forwarding (or bypassing)

- **Send data as soon as it is computed to the stages that need it without waiting for it to be written on to the register bank**

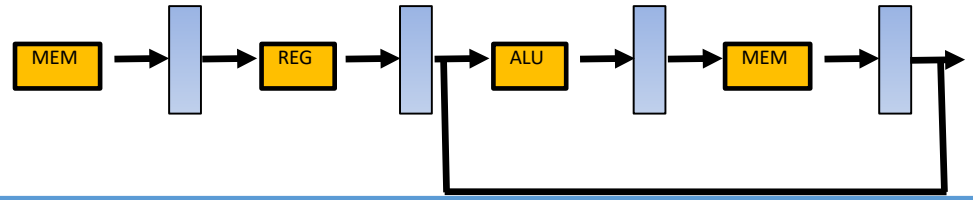
Case 1:

```
add r1,r2,r3
sub r4,r1,r3
```

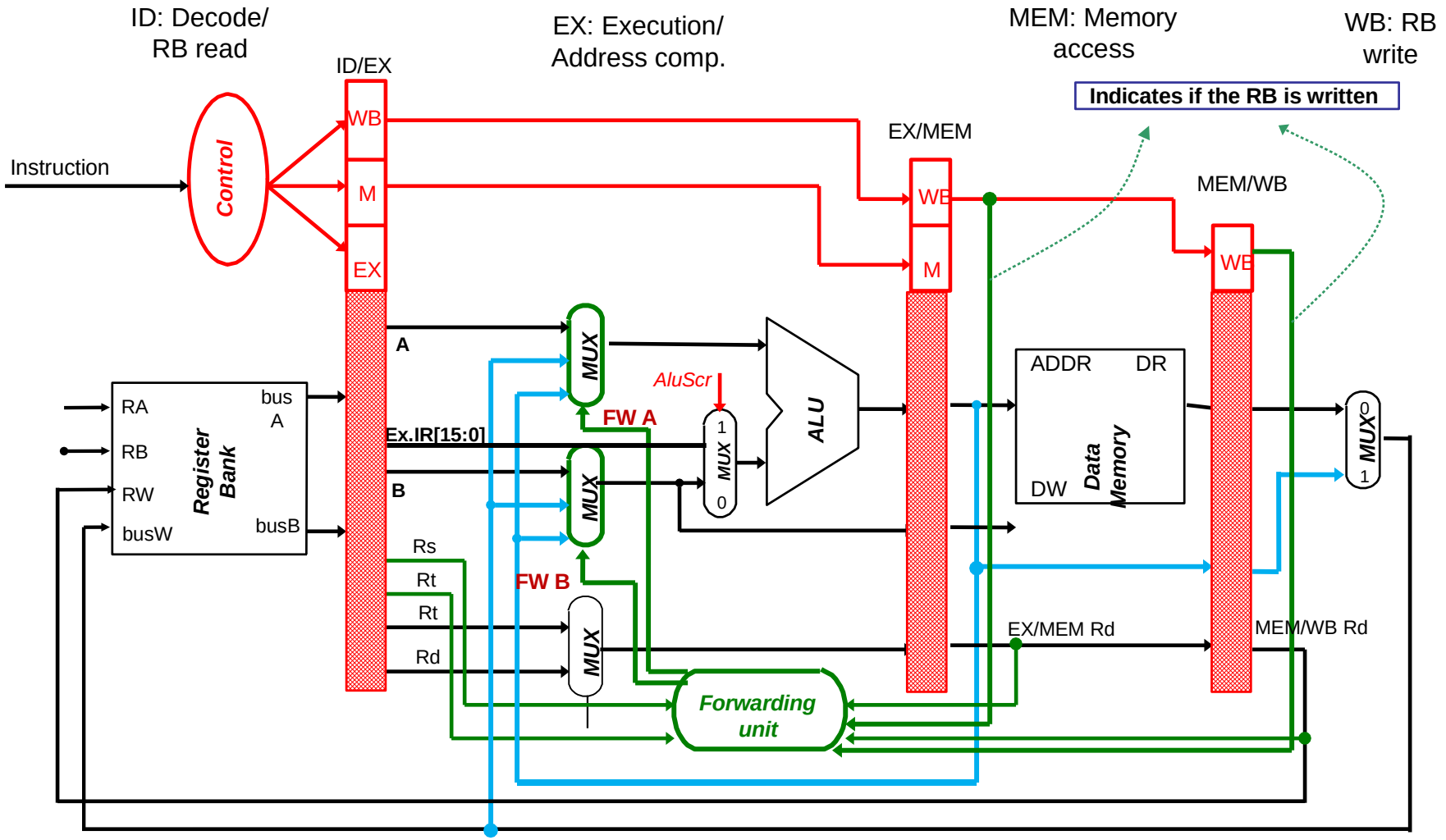


Case 2:

```
add r1,r2,r3
sub  r4,r5,r3
and r6,r1
```

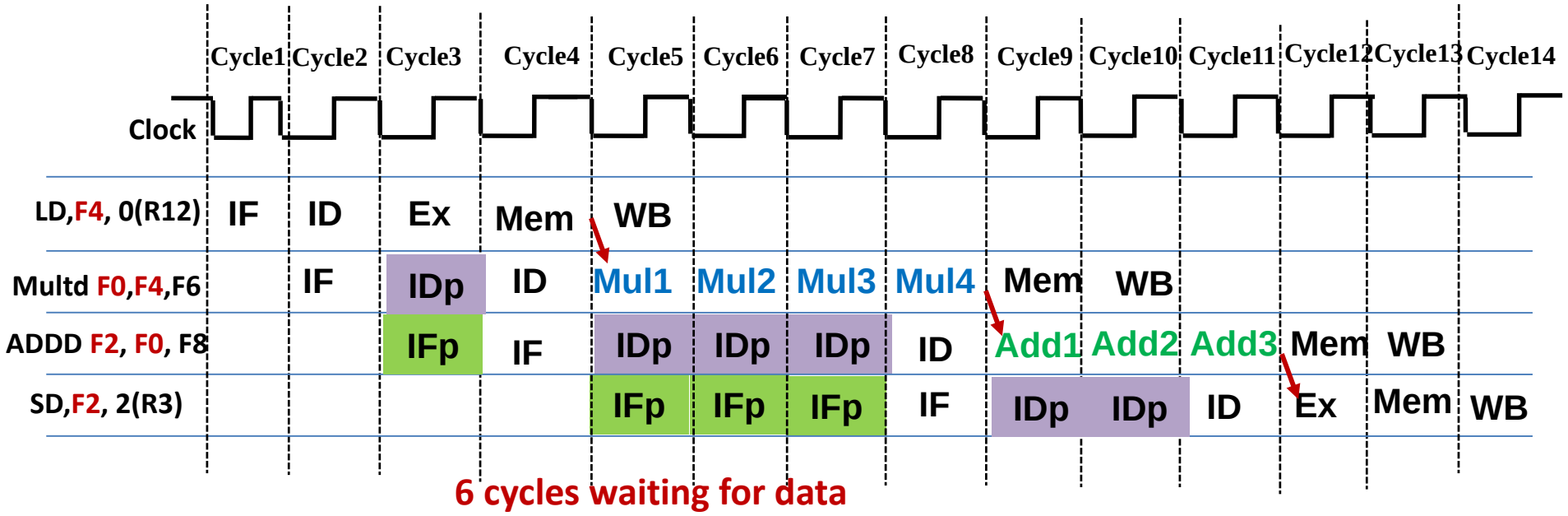


RAW Hazards in the basic pipeline: Forwarding implementation



Control information in green, data in light blue

RAW Hazards in the pipeline with multicycle functional units: Forwarding

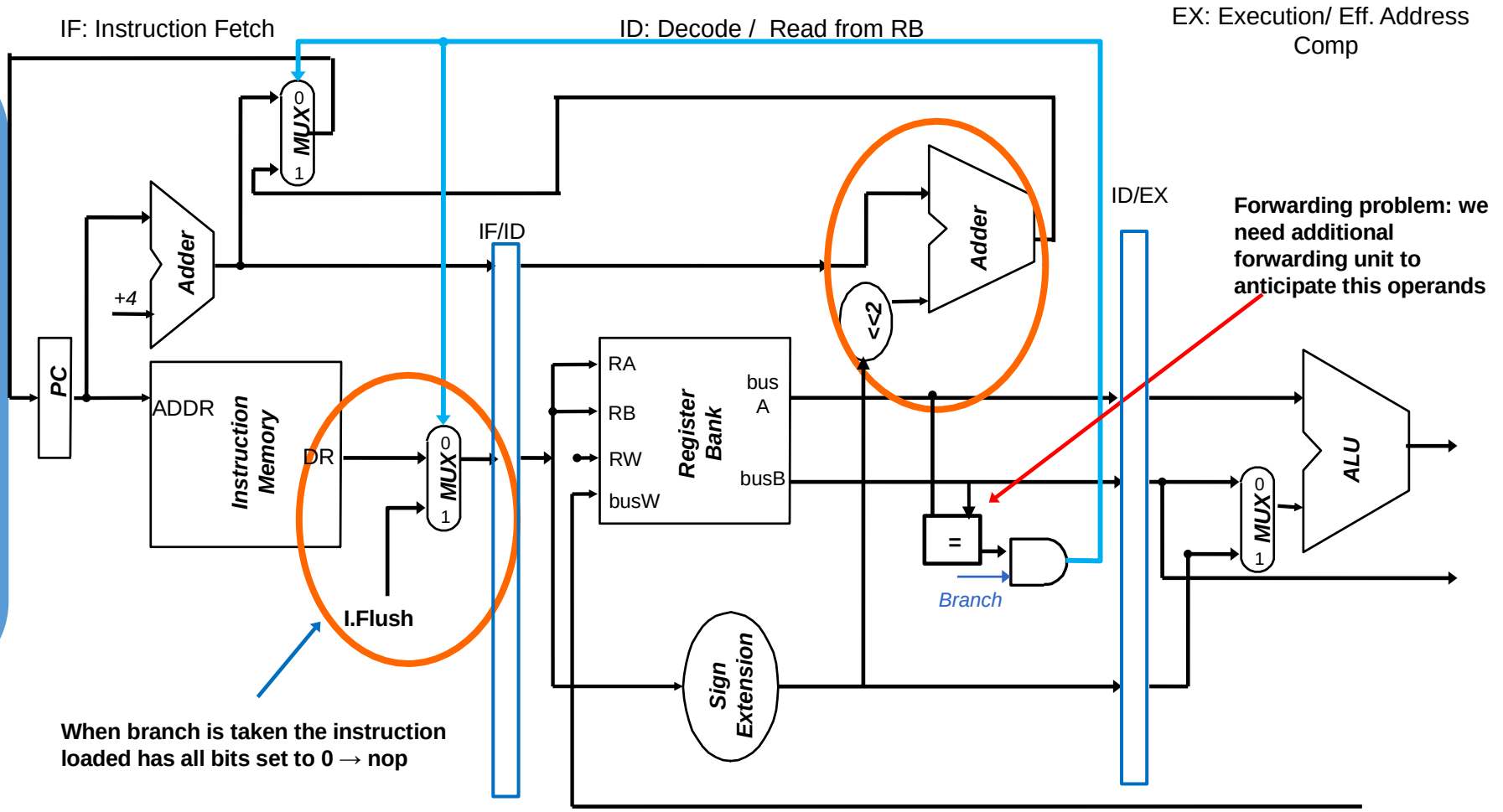


4. Given a pipelined processor with 5 stages and the following features:

Branches are resolved at Decode using stalls.

Computes the **branch target address** and evaluates the **condition** in the **ID stage.**

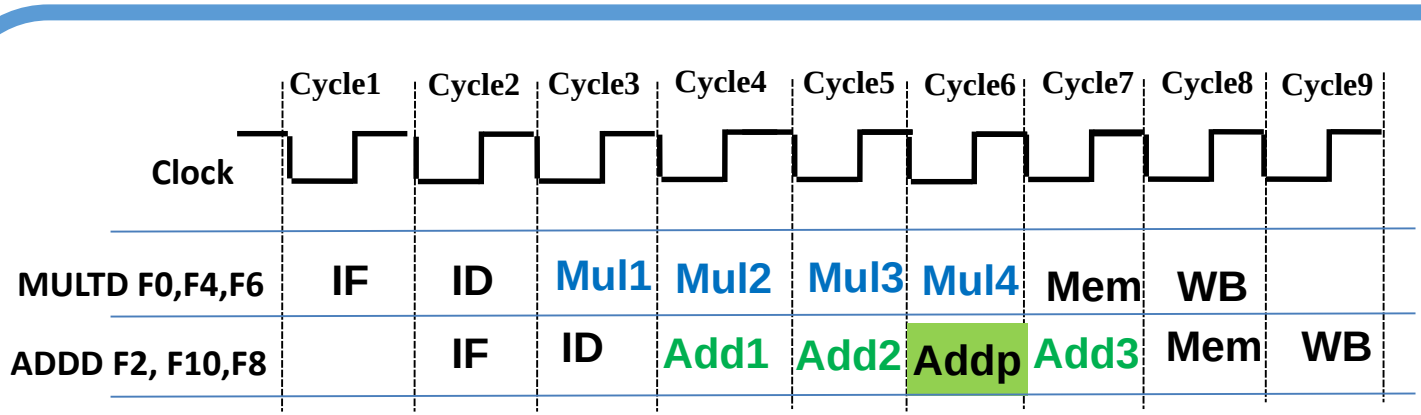
Transforms the fetched instruction into a nop (flush) if branch is taken



4. Given a pipelined processor with 5 stages and the following features:

Structural hazards at the MEM stage are detected and resolved via stalling in the last stage of execution.

MULTD F0,F4,F6	IF	ID	Mul1	Mul2	Mul3	Mul4	Mem	WB
ADDD F2, F10,F8		IF	ID	Add1	Add2	Add3	Mem	WB



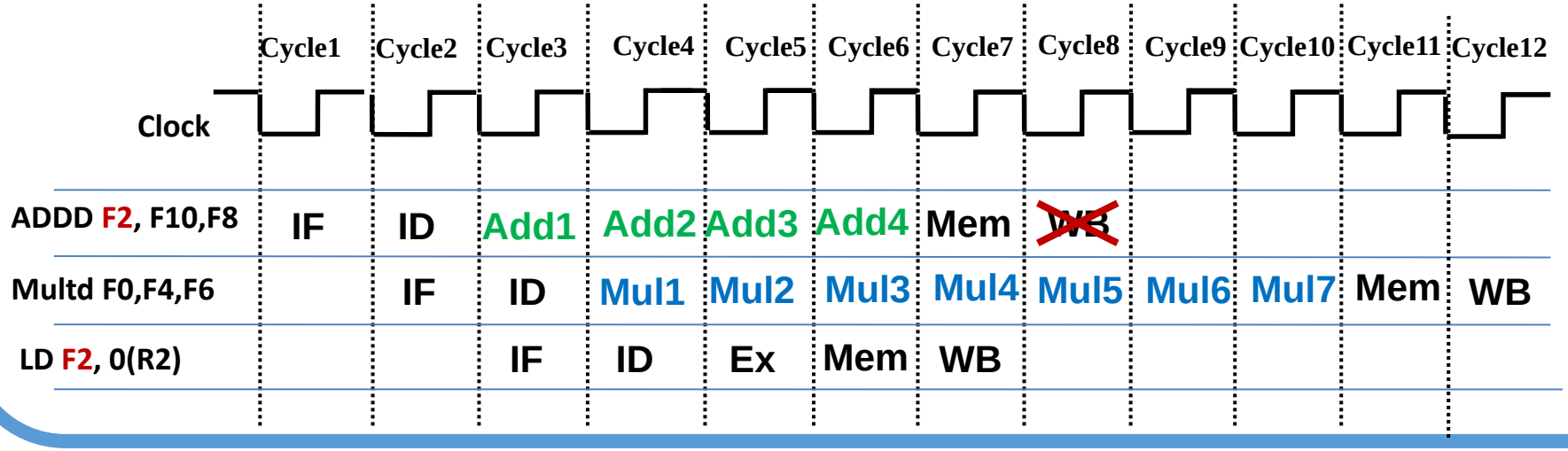
FU	Lat
FP add	3
FP multiply	4

Addp **Stall due to a structural hazard, two instructions reaching the Mem stage in cycle 7**

4. Given a pipelined processor with 5 stages and the following features:

WAW hazards are resolved by avoiding the write of the first instruction.

Inhibit the RB write for the first instruction (clearing its write enable signal on the pipelining register)



4. Given a pipelined processor with 5 stages and the following features:

Functional units in the processor:

Functional Unit	Number	Latency	Pipelined
FP ADD	1	2	YES
FP MUL	1	5	YES
Int ALU	1	1	NO

No structural hazards if two instructions need to use the same functional unit

4. Given a pipelined processor with 5 stages and the following features:

The following program is executed in this processor:

```
LD      F10,0(R1)
MULD    F4,F0,F10
LD      F12,0(R2)
ADDD    F2,F12,F4
LD      F4,8(R1)
MULD    F12,F4,F12
LD      F12,16(R1)
```

a) Represent the instruction/time diagram for the program and highlight the forwardings that take place.

b) Calculate the CPI.

4. Given a pipelined processor with 5 stages and the following features:

First step: detect hazards

```
        LD      F10,0(R1)
        MULD    F4,F0,F10
        LD      F12,0(R2)
Structural
hazard  ADDD    F2,F12,F4
MEM     LD      F4,8(R1)
        MULD    F12,F4,F12
        LD      F12,16(R1)
```

RAW

WAW

Second step: remember how to solve each problem (past slides)

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
LD	F10,0(R1)	IF	ID	EX	MEM	WB															
MULD	F4,F0,F10		IF	IDp	ID	X1	X2	X3	X4	X5	MEM	WB									
LD	F12,0(R2)				IF	ID	EX	MEM	WB												
ADDD	F2,F12,F4					IF	IDp	IDp	IDp	ID	A1	A2	MEM	WB							
LD	F4,8(R1)									IF	ID	EXP	EX	MEM	WB						
MULD	F12,F4,F12										IF	IDp	IDp	ID	X1	X2	X3	X4	X5	MEM	~~WB~~
LD	F12,16(R1)													IF	ID	EX	MEM	WB			