COL788: Advanced Topics in Embedded Computing

Lecture 11 – Pipelining (Cont.)

Vireshwar Kumar
CSE@IITD

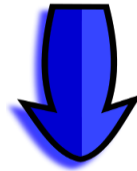September 1, 2022

Semester I
2022-2023

Last Lectures

- Basics of Pipelining

- Data and Control Hazards
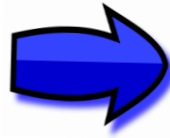
Software Solution to Data Hazard: *nop*

```
[1]: add r1, r2, r3
[2]: sub r3, r1, r4
```



```
[1]: add r1, r2, r3
[2]: nop
[3]: nop
[4]: nop
[5]: sub r3, r1, r4
```

Software Solution: Code Reordering

```
add r1, r2, r3
add r4, r1, 3
add r8, r5, r6
add r9, r8, r5
add r10, r11, r12
add r13, r10, 2
```



```
add r1, r2, r3
add r8, r5, r6
add r10, r11, r12
nop
add r4, r1, 3
add r9, r8, r5
add r13, r10, 2
```

Software Solution to Control Hazard

- Assume that the two instructions fetched after a branch are valid instructions

- These instructions are said to be in the delay slots as a delayed branch

- The compiler transfers instructions before the branchto the delay slots

```
add r1, r2, r3
add r4, r5, r6
b .foo
add r8, r9, r10
```
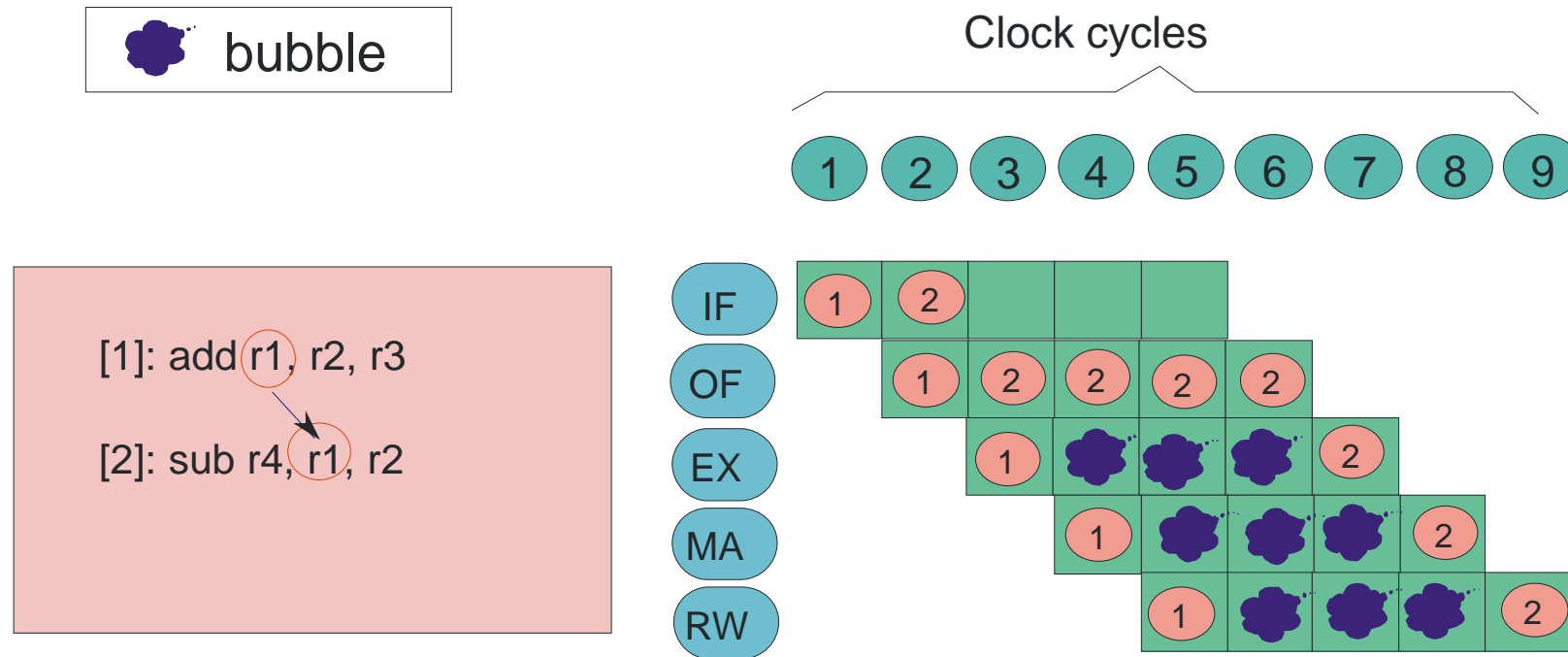
```
b .foo
add r1, r2, r3
add r4, r5, r6
add r8, r9, r10
```
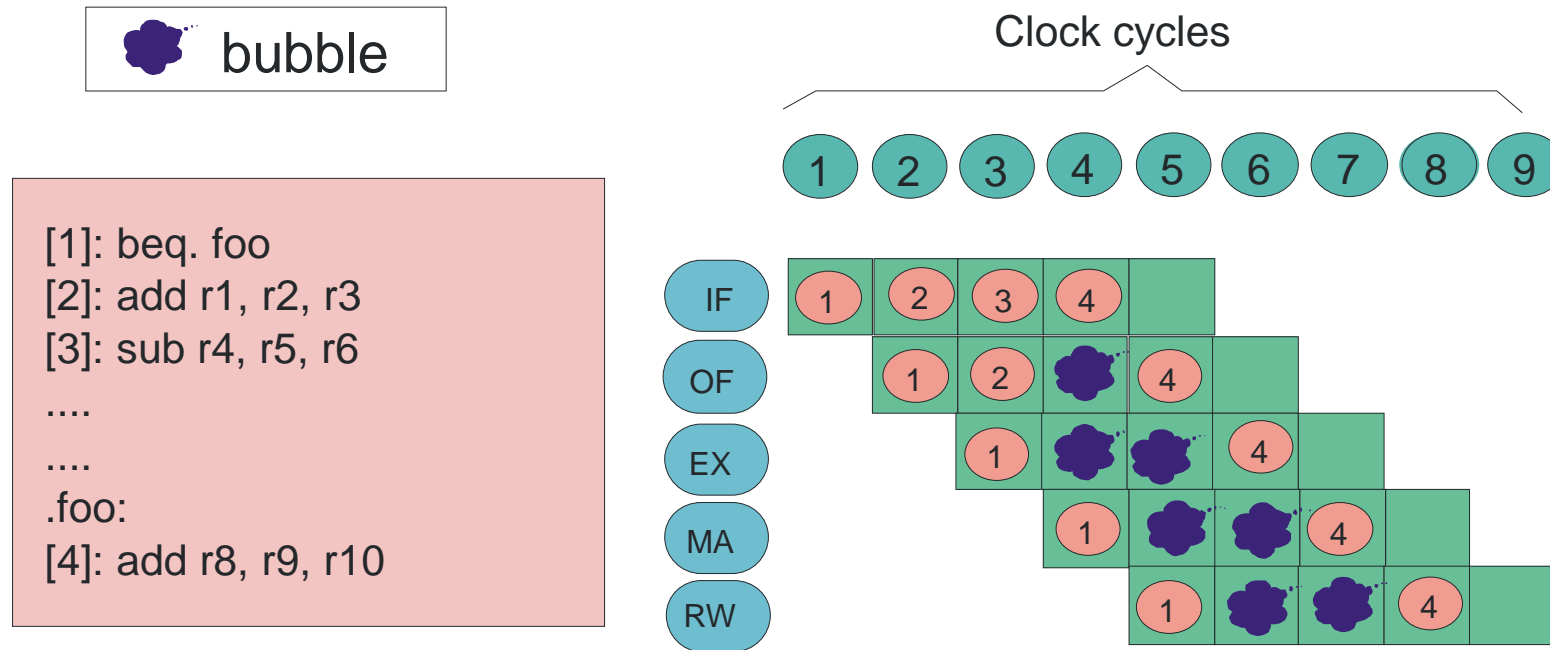
Hardware Solution: Interlock

- Hardware mechanism to enforce correctness → interlock

- Data-Lock
  - Do not allow a consumer instruction to move beyond the OF stage till it has read the correct values
  - Implication : Stall the IF and OF stages
- Branch-Lock
  - We never execute instructions in the wrong path

- A pipeline bubble is inserted into a stage, when the previous stage needs to be stalled
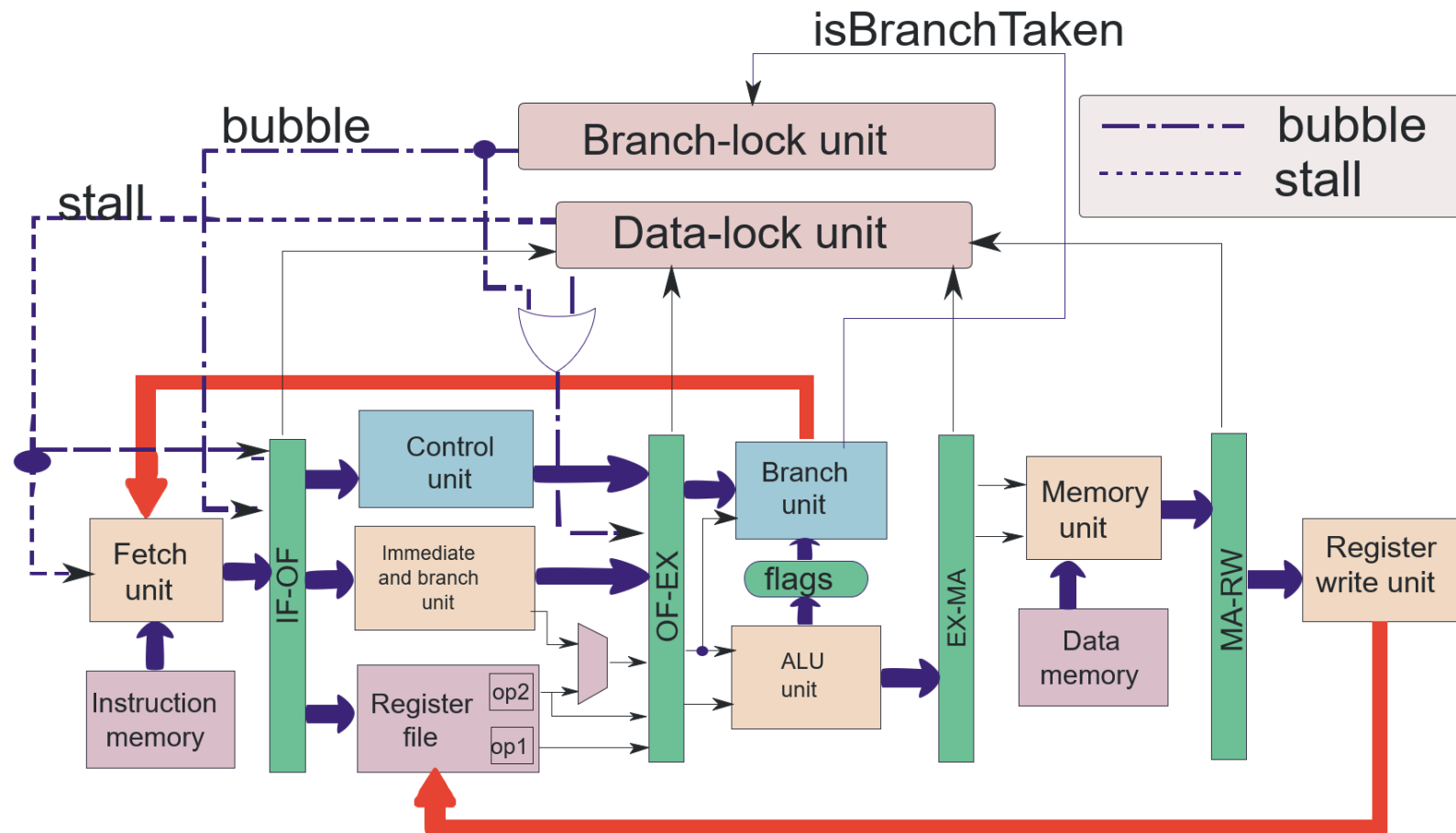
Data Hazard Mitigation

bubble

Clock cycles

[1]: add r1, r2, r3

[2]: sub r4, r1, r2

Control Hazard Mitigation

bubble

[1]: beq. foo
[2]: add r1, r2, r3
[3]: sub r4, r5, r6
....
....
.foo:
[4]: add r8, r9, r10

Clock cycles

1 2 3 4 5 6 7 8 9

	1	2	3	4	5	6	7	8	9
IF	1	2	3	4					
OF		1	2	🟣	4				
EX			1	🟣	🟣	4			
MA				1	🟣	🟣	4		
RW					1	🟣	🟣	4	

Data Path with Interlocks

Hardware vs Software Solutions

Attribute	Software	Hardware(withinterlocks)
Portability	Limited to a specific processor	Programs can be run on any processor irrespective of the nature of the pipeline
Branches	Possible to have no performance penalty, by using delay slots	Need to stall the pipeline for 2 cycles in our design
RAW hazards	Possible to eliminate them through code scheduling	Need to stall the pipeline
Performance	Highly dependent on the nature of the program	The basic version of a pipeline with interlocks is expected to be slower than the version that relies on software

What's Next?

- Next Lecture (September 5, Monday, 11 am – 12 pm)
  - Lecture 12