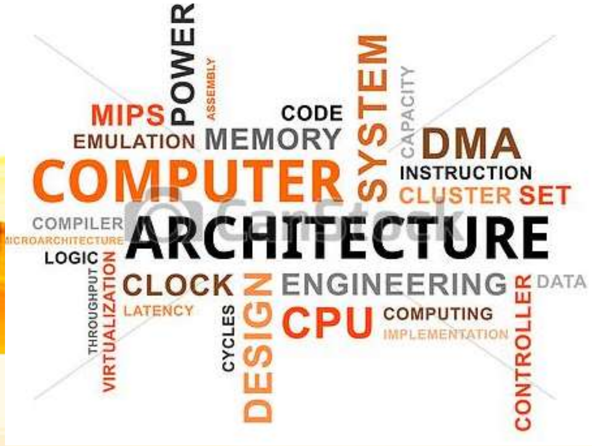
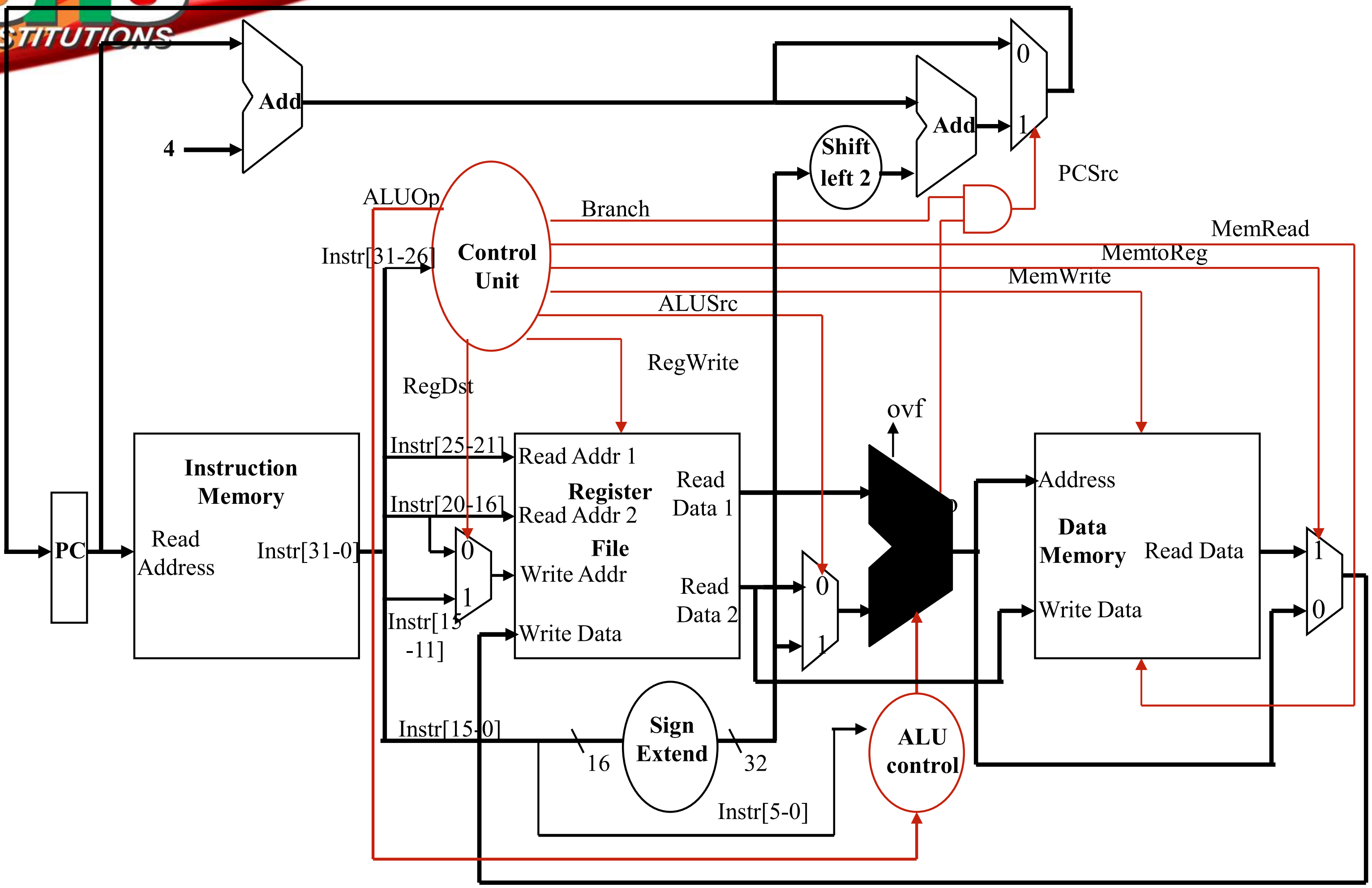
UNIT III
PROCESSOR AND PIPELINING

Fundamental concepts – Execution of a complete instruction – Multiple bus organization – Hardwired control – Micro programmed control – **Pipelining: Basic concepts –** Data hazards – Instruction hazards – Influence on Instruction sets – Data path and control consideration.



Recap the previous Class

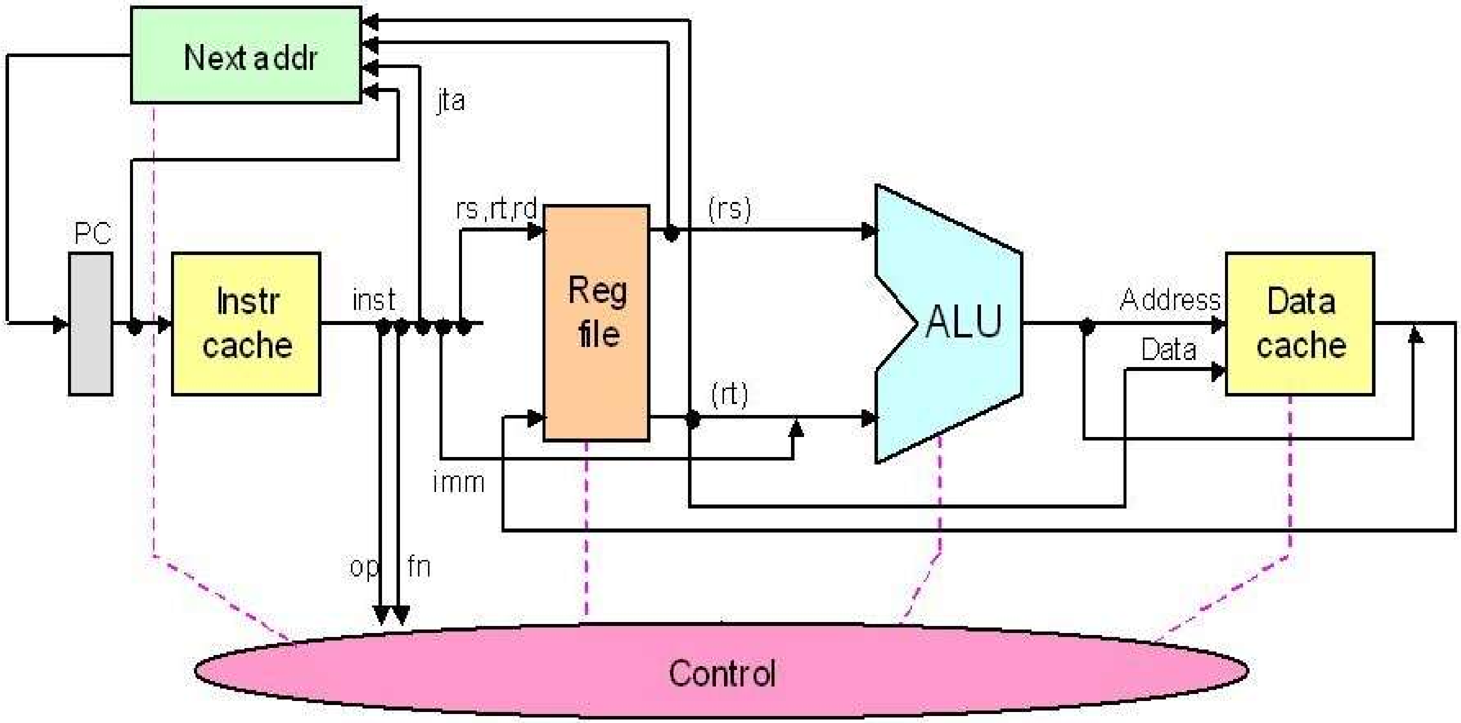
RECALL THE PRIOR KNOWLEDGE



Overview

- Widely used in modern processor
- Executing machine instructions concurrently
- Improve System performance in terms of throughput
- Requires sophisticated complication Technique

INSTRUCTION EXECUTION



Basic Concepts

- Faster circuit technology to build the processor and the main memory
- Arrange the hardware so that more than one operation can be performed at the same time
- Number of operations performed per second is increased

WHAT IS PIPELINING?

Overlapping of instruction
Improve Performance

REQUIREMENTS

Three Students – Act as Processor
Resources Ball

Calculate the Throughput and Response Time ?

Activity Set 1 : Rules
complete the task one by one

Activity Set 2 : Rules
complete the task in parallel

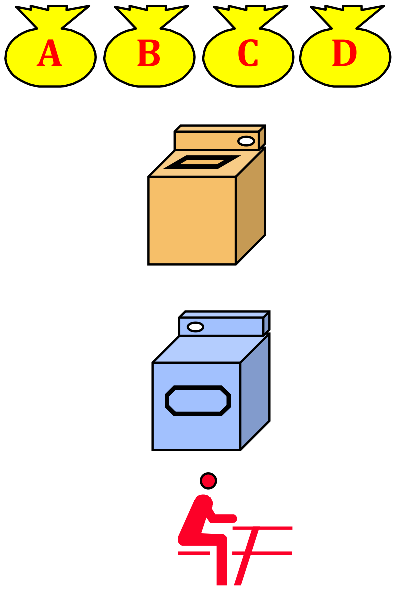
Pipelining

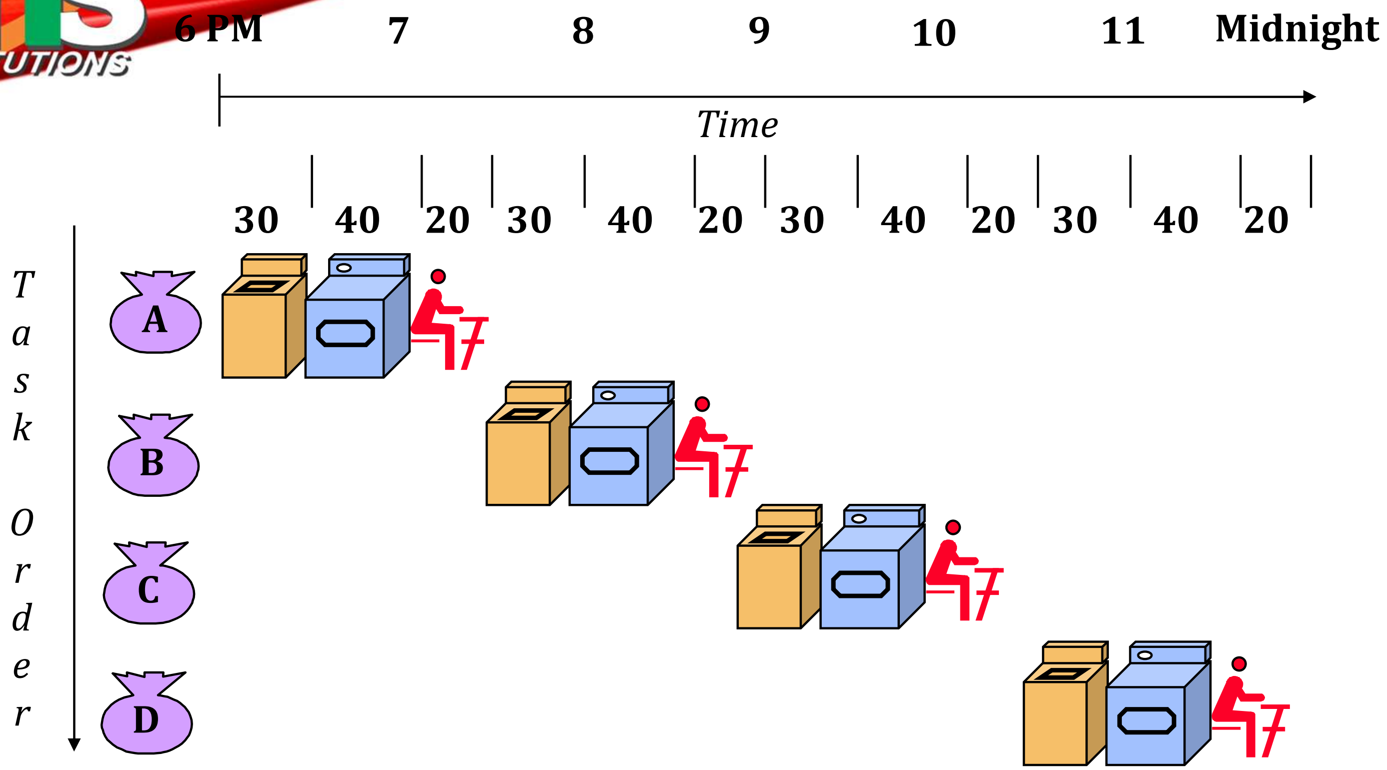
- Effective way of organizing concurrent activity in a computer system.
- Analogy - assembly-line operation in car manufacturing Sector.
  - welding
  - Painting
  - Polishing

What Is Pipelining?

Laundry Example

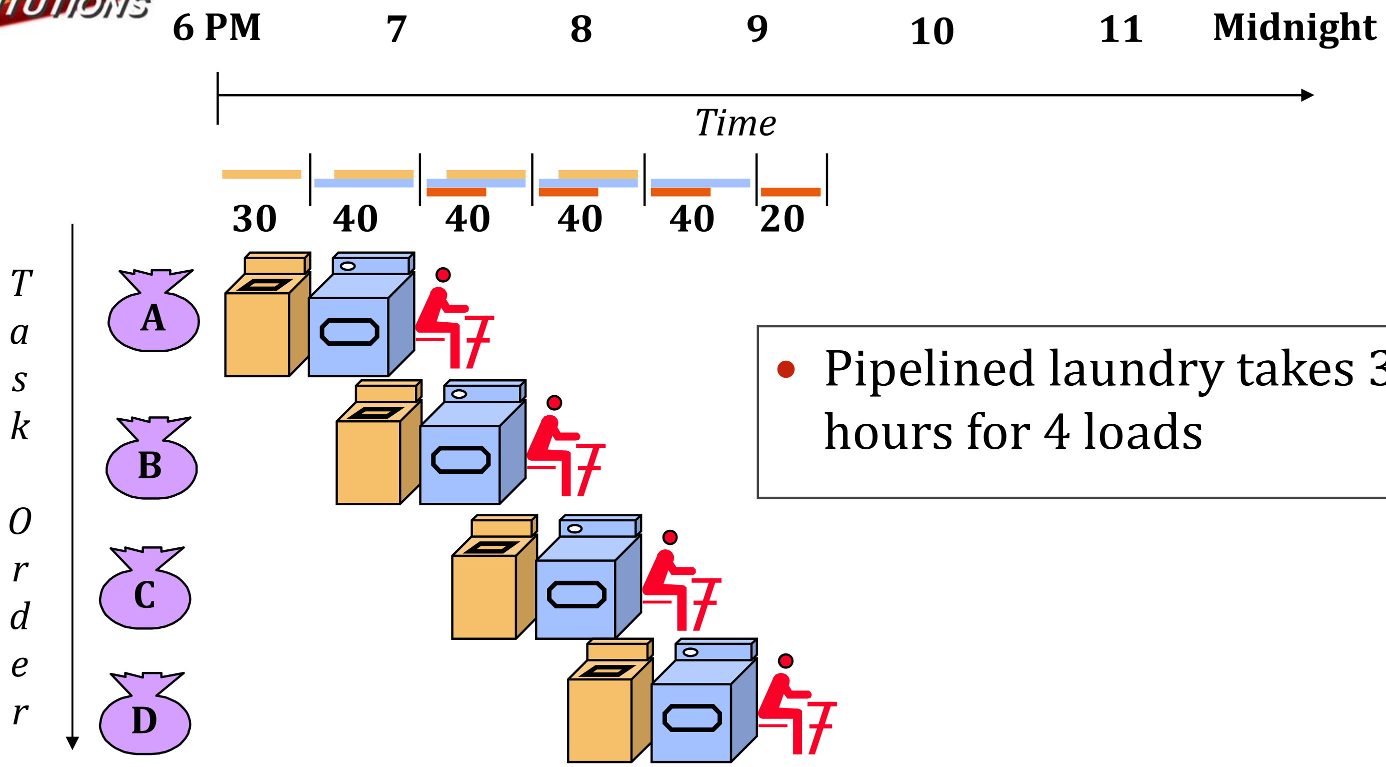
- Ann, Brian, Cathy, Dave , each have one load of clothes to wash, dry, and fold
  - Washer takes 30 minutes
  - Dryer takes 40 minutes
  - “Folder” takes 20 minutes





Sequential laundry takes (360 Minutes) 6 hours for 4 loads

If they learned pipelining, how long would laundry take?



- Pipelined laundry takes 3.5 hours for 4 loads

PIPELINE Characteristics

Balanced (all take the same time)

Time between instructions$_{\text{pipelined}}$ =

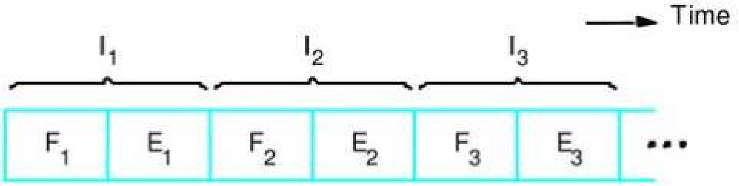
Time between instructions$_{\text{nonpipelined}}$ / Number of stages

Not balanced

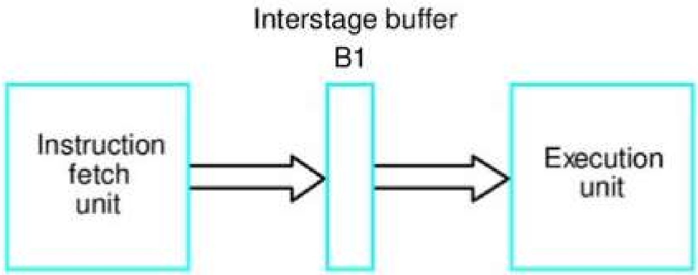
Speedup is less

Speedup due to increased throughput
Latency (Time for each instruction) does not decrease

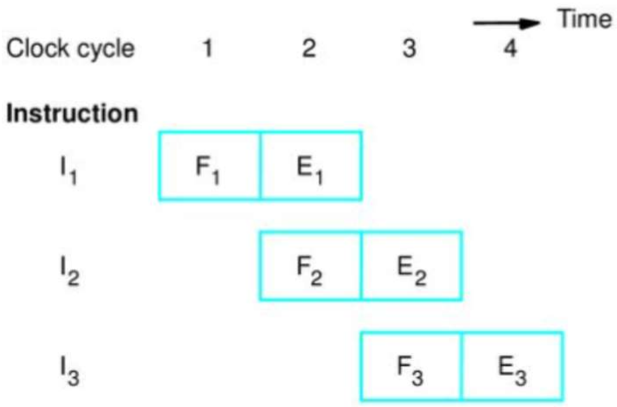
Idea of pipelining in computer



(a) Sequential execution



(b) Hardware organization



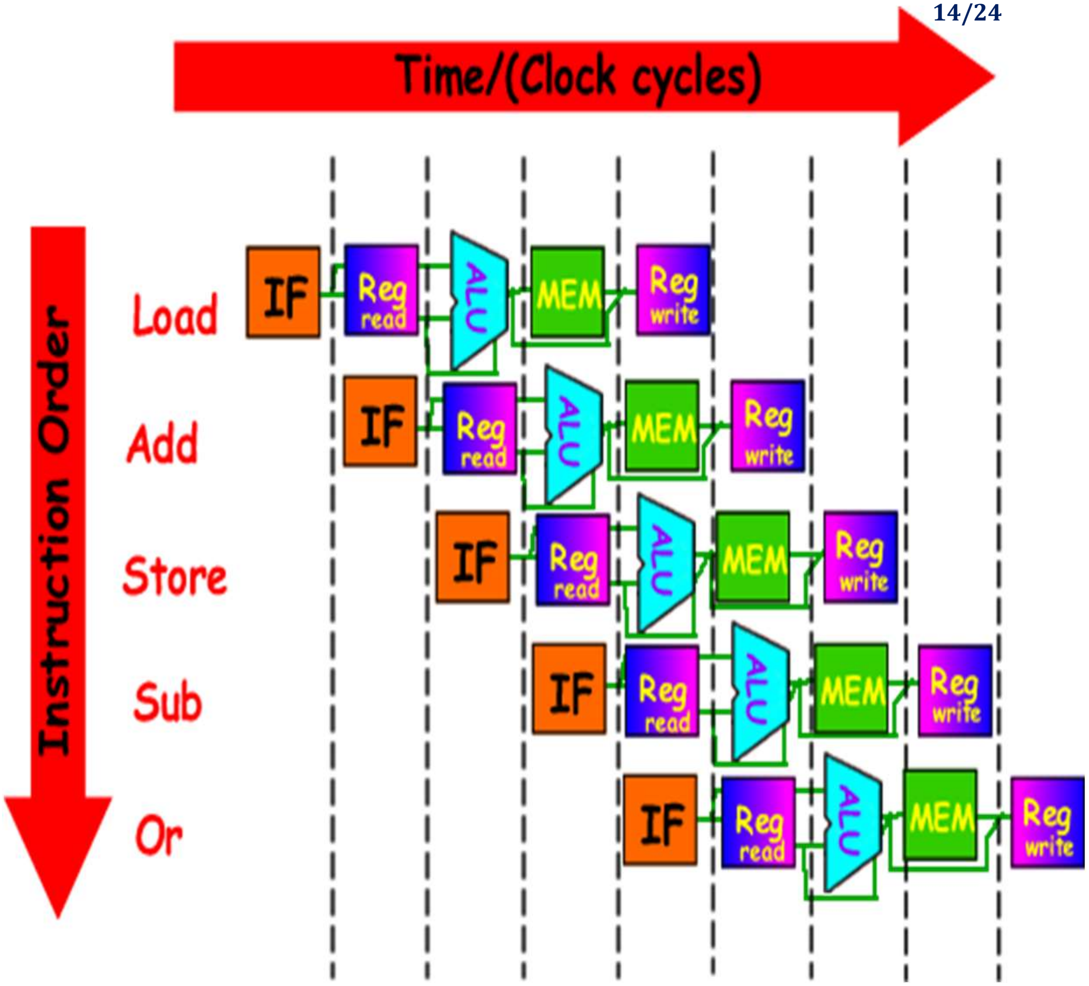
(c) Pipelined execution

F Fetch: read the instruction from the memory.

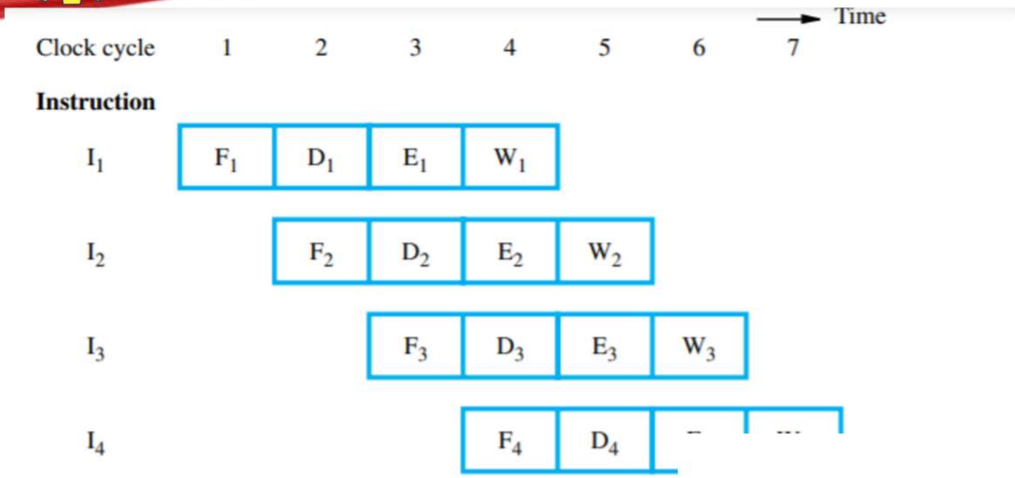
D Decode: decode the instruction and fetch the source operand(s).

E Execute: perform the operation specified by the instruction.

W Write: store the result in the destination location

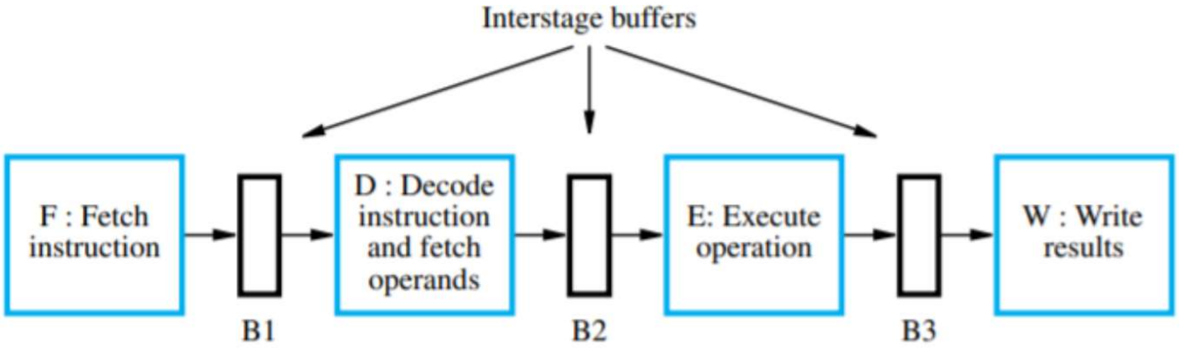


Role of cache memory



→ Time

Clock cycle	1	2	3	4	5	6	7
Instruction							
I_1	F_1	D_1	E_1	W_1			
I_2		F_2	D_2	E_2	W_2		
I_3			F_3	D_3	E_3	W_3	
I_4				F_4	D_4		

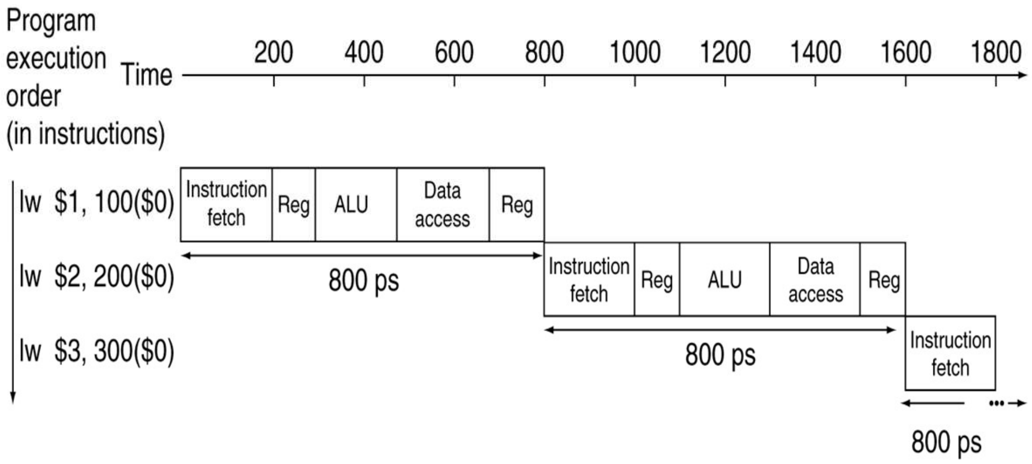


Pipeline Performance

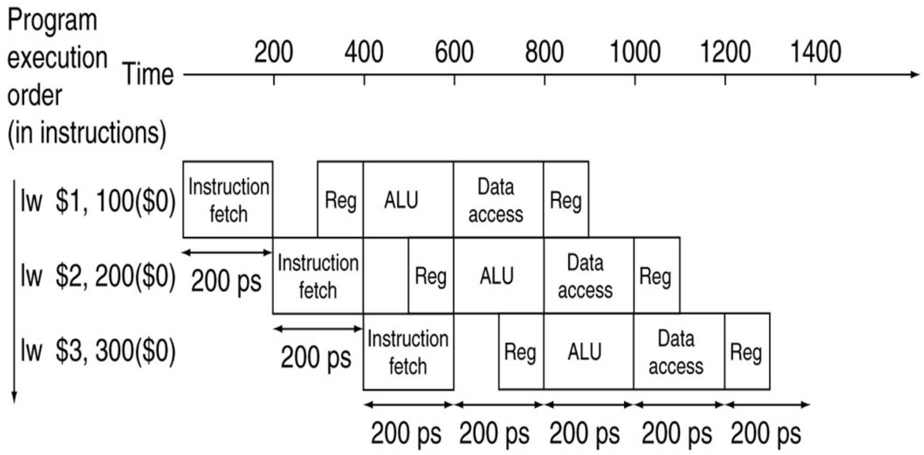
- Assume time for stages is
  - 100ps for register read or write
  - 200ps for other stages
- Compare pipelined datapath with single-cycle datapath

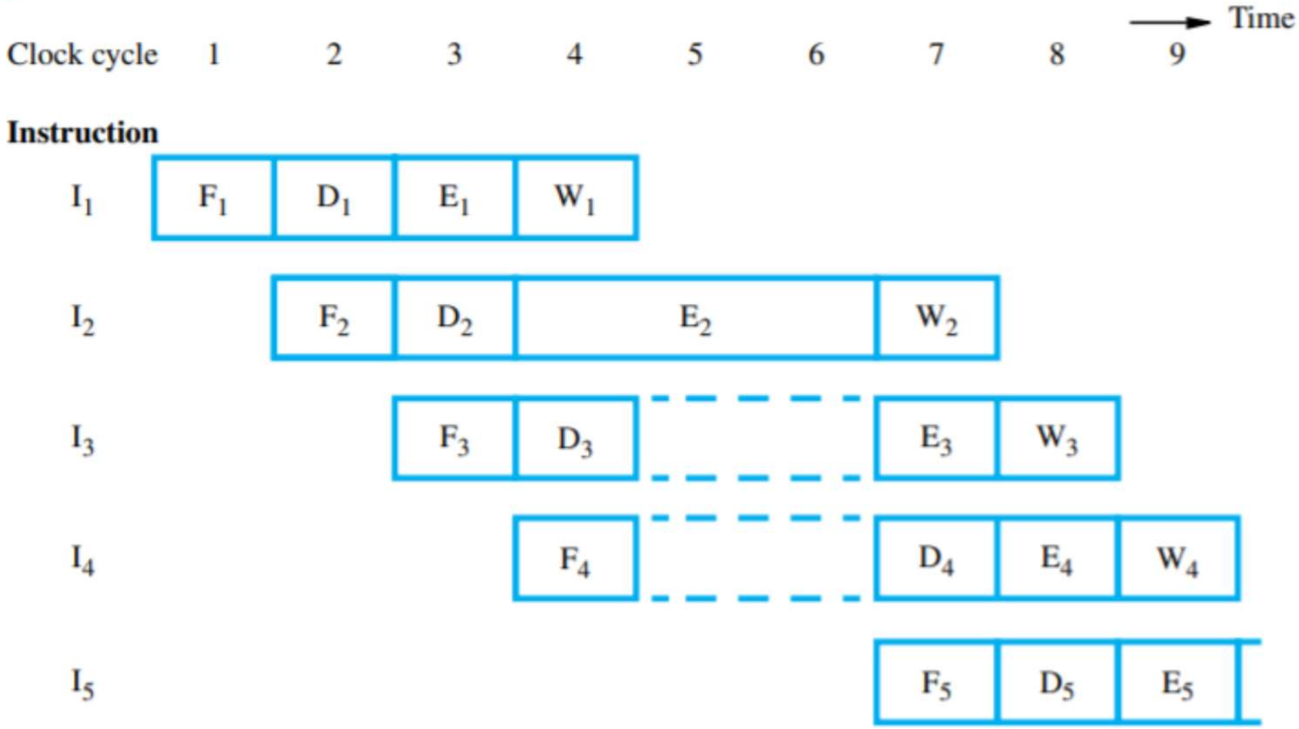
Instruction	Instruction fetch	Register read	ALU op	Memory access	Register write	Total time
Load	200ps	100 ps	200ps	200ps	100 ps	**800ps**
Store	200ps	100 ps	200ps	200ps		**700ps**
R-format	200ps	100 ps	200ps		100 ps	**600ps**
Branch	200ps	100 ps	200ps			**500ps**

Single-cycle (T_c= 800ps)



Pipelined (T_c= 200ps)

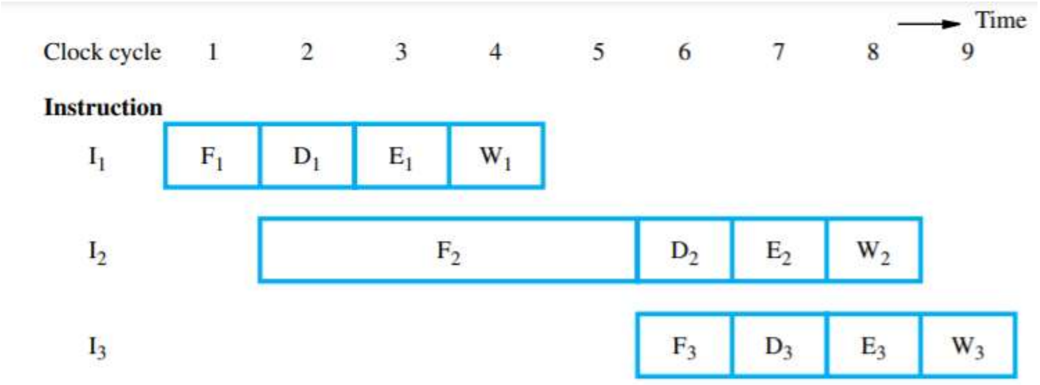




Clock cycle	1	2	3	4	5	6	7	8	9
Instruction									
I_1	F_1	D_1	E_1	W_1					
I_2		F_2	D_2	E_2			W_2		
I_3			F_3	D_3			E_3	W_3	
I_4				F_4			D_4	E_4	W_4
I_5							F_5	D_5	E_5

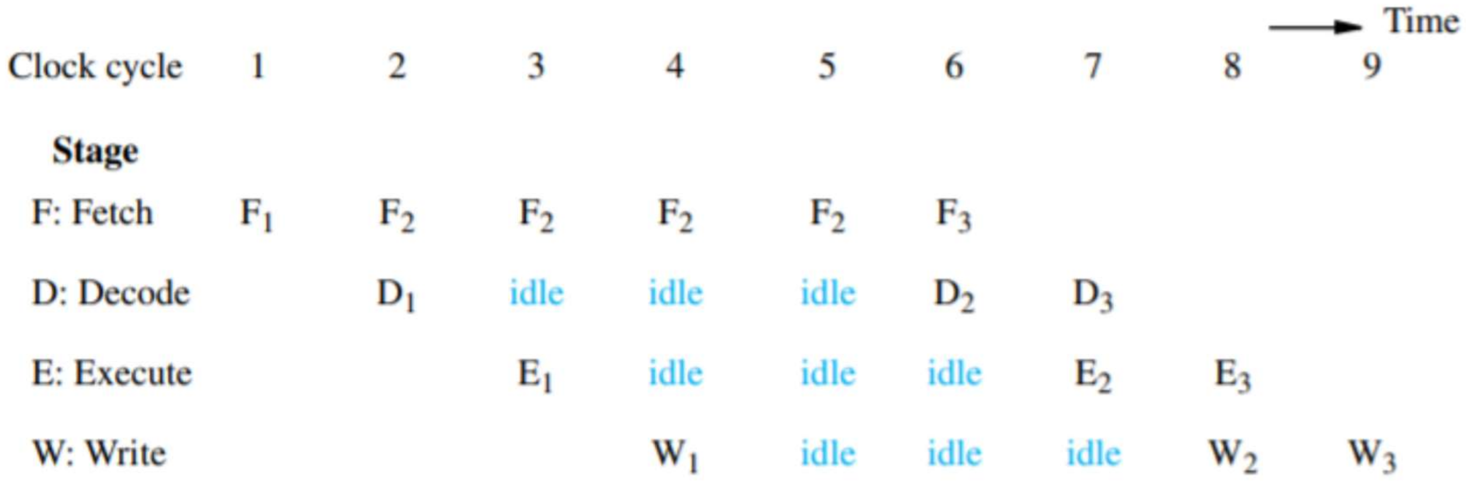
Effect of an execution operation taking more than one clock cycle

Pipeline stall caused by a cache miss in F2



→ Time

Clock cycle	1	2	3	4	5	6	7	8	9
Instruction									
I_1	F_1	D_1	E_1	W_1					
I_2		F_2				D_2	E_2	W_2	
I_3						F_3	D_3	E_3	W_3



→ Time

Clock cycle	1	2	3	4	5	6	7	8	9
Stage									
F: Fetch	F_1	F_2	F_2	F_2	F_2	F_3			
D: Decode		D_1	idle	idle	idle	D_2	D_3		
E: Execute			E_1	idle	idle	idle	E_2	E_3	
W: Write				W_1	idle	idle	idle	W_2	W_3

HAZARDS

DATA HAZARDS

Need to wait for previous instruction

STRUCTURAL HAZARDS

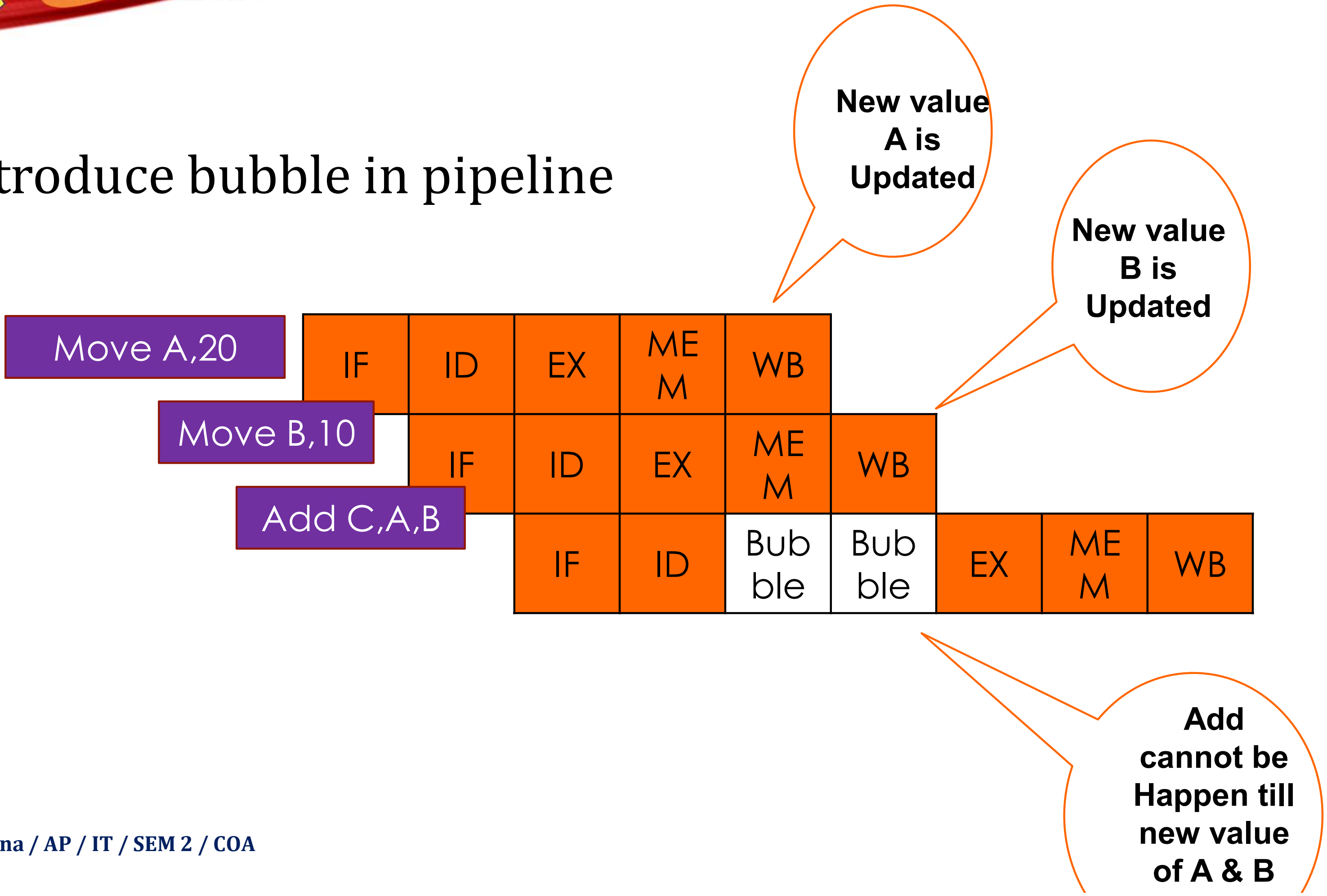
required resource is busy

CONTROL HAZARDS

Deciding on control action

HOW DO WE SOLVE?

Introduce bubble in pipeline



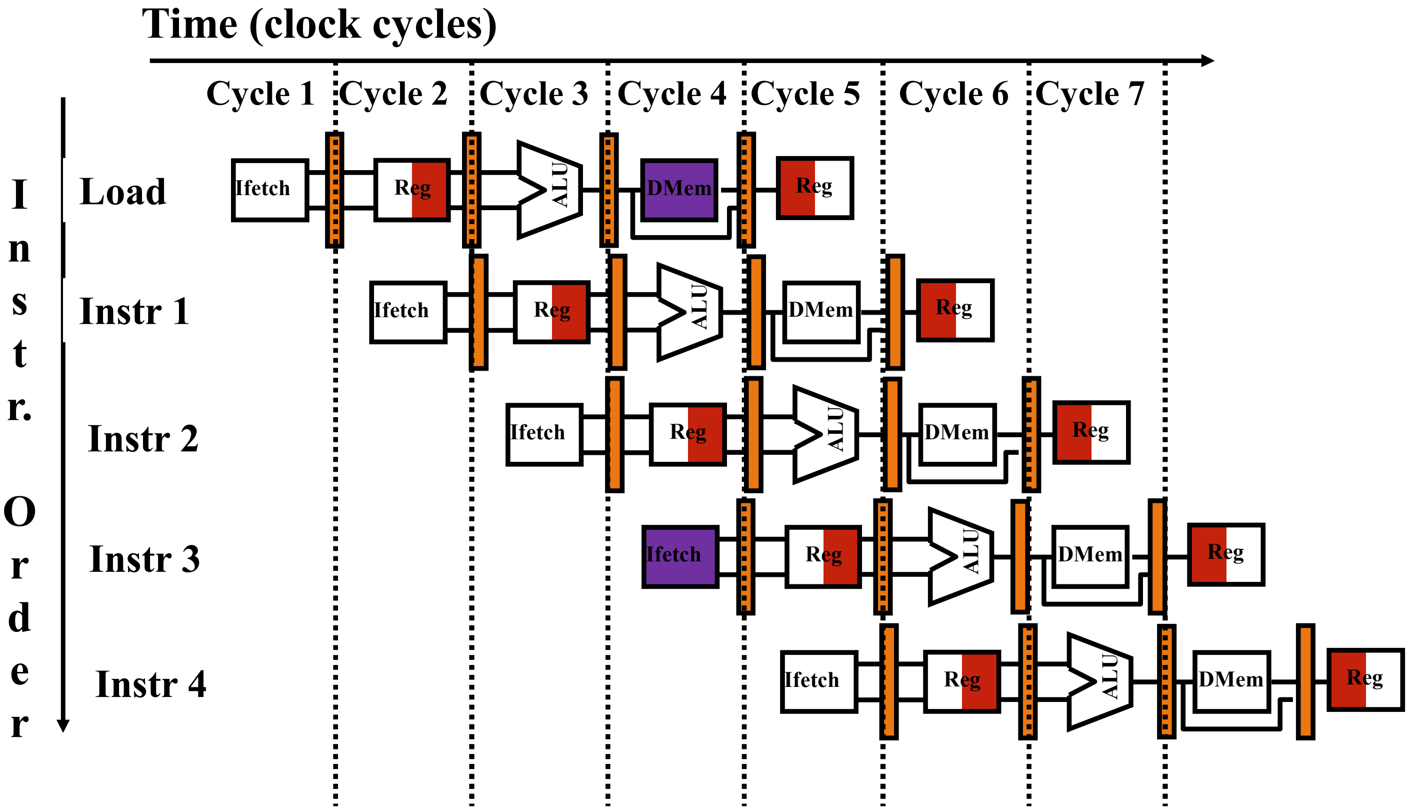
Move A,20	IF	ID	EX	MEM	WB				
Move B,10		IF	ID	EX	MEM	WB			
Add C,A,B			IF	ID	Bubble	Bubble	EX	MEM	WB

New value A is Updated

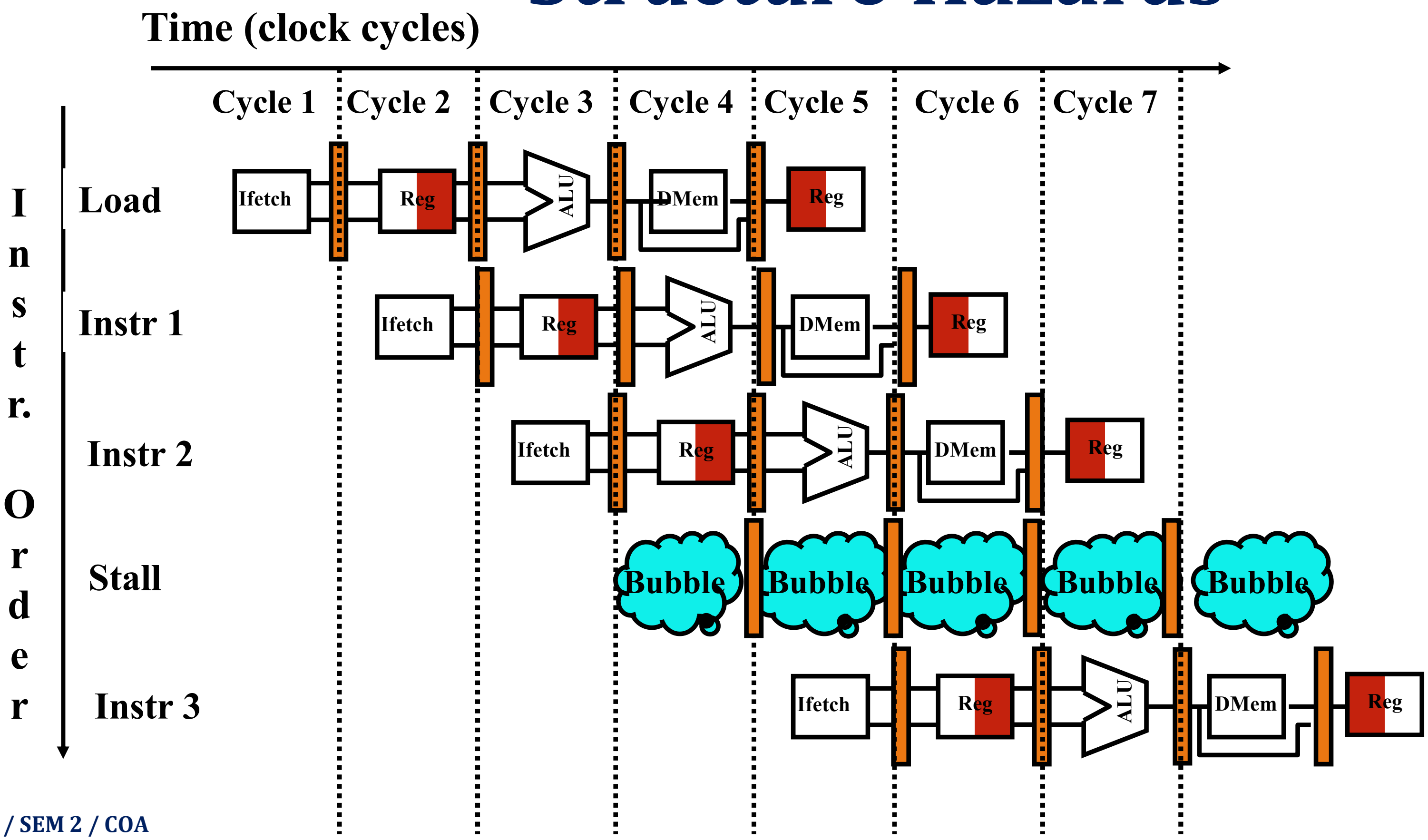
New value B is Updated

Add cannot be Happen till new value of A & B

Structure Hazards



Structure Hazards



Thank You