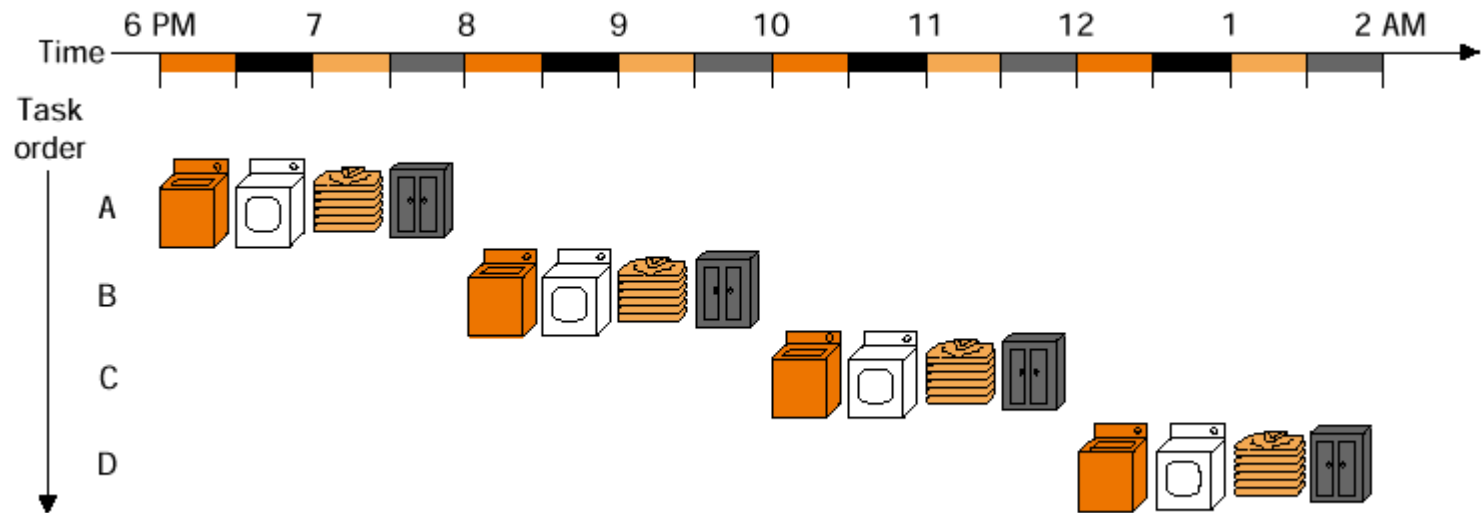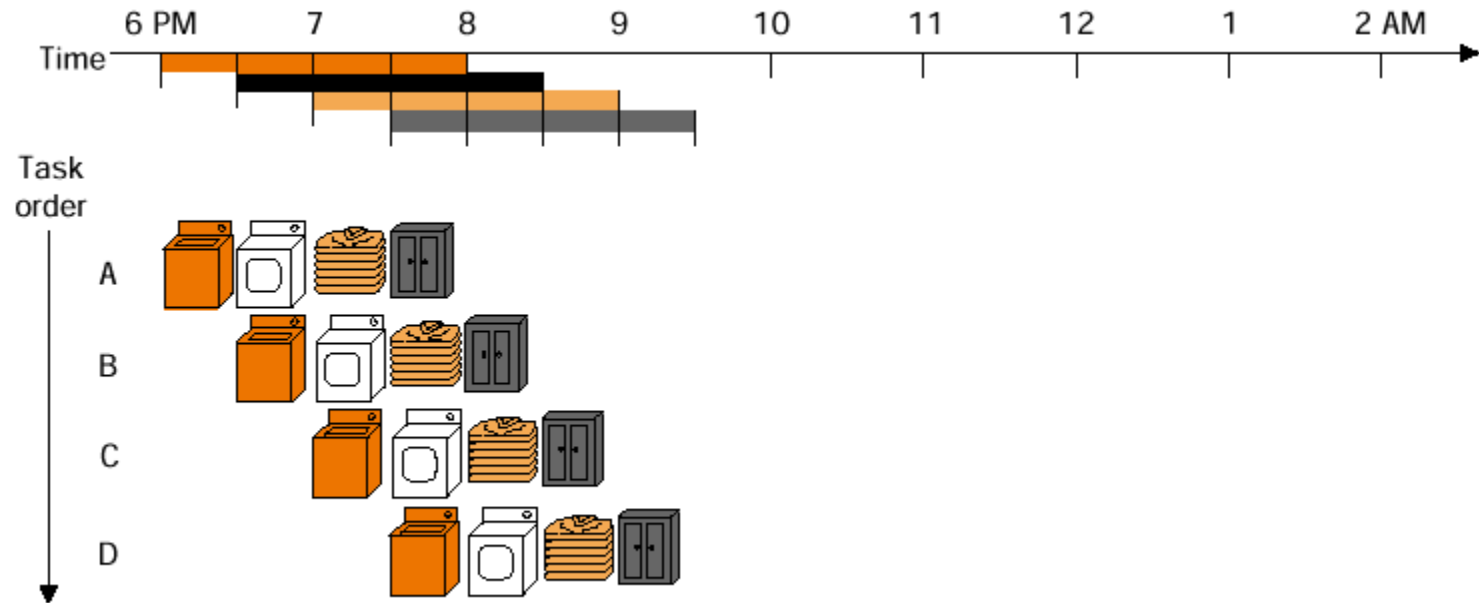# Computer Architecture

Pipelining Basics

Sequential Processing

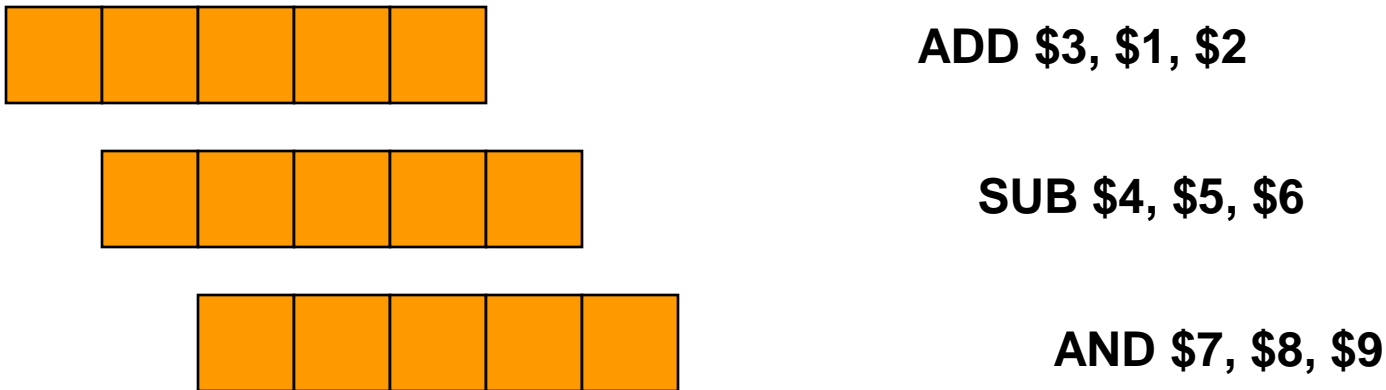# Pipelined Processing

Basic Steps of Execution

1. Instruction fetch step (IF)

2. Instruction decode/register fetch step (ID)

3. Execution/effective address step (EX)

4. Memory access (MEM)

5. Register write-back step (WB)

Pipelined Instruction Execution

❑ Sequential Execution

❑ Pipelined Execution

ADD $3, $1, $2

SUB $4, $5, $6

AND $7, $8, $9

Basic Pipeline

Instruction number	Clock number								
	1	2	3	4	5	6	7	8	9
Instruction i	IF	ID	EX	MEM	WB				
Instruction $i + 1$		IF	ID	EX	MEM	WB			
Instruction $i + 2$			IF	ID	EX	MEM	WB		
Instruction $i + 3$				IF	ID	EX	MEM	WB	
Instruction $i + 4$					IF	ID	EX	MEM	WB

Major Hurdles of Pipelining

- Structural Hazard
- Data Hazard
- Control Hazard

Structural Hazard

IF	ID	EX	MEM	WB

IF	ID	EX	MEM	WB

IF	ID	EX	MEM	WB
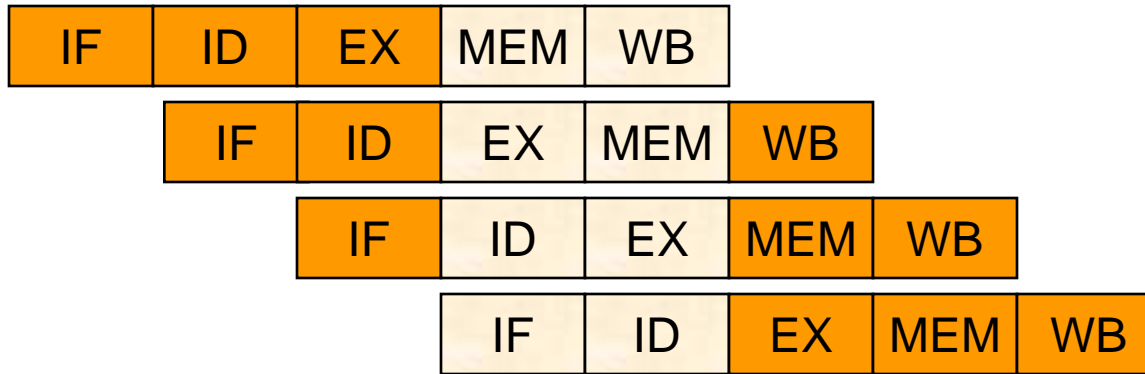
IF	ID	EX	MEM	WB

Instruction number	Clock number								
	1	2	3	4	5	6	7	8	9
Load Instruction	IF	ID	EX	MEM	WB				
Instruction i + 1		IF	ID	EX	MEM	WB			
Instruction i + 2			IF	ID	EX	MEM	WB	WB	
Instruction i + 3					IF	ID	EX	MEM	WB
Instruction i + 4						IF	ID	EX	MEM

Solutions to Structural Hazard

❑ Resource Duplication

- example
  - Separate I and D caches for memory access conflict
  - Time-multiplexed or multi-port register file for register file access conflict

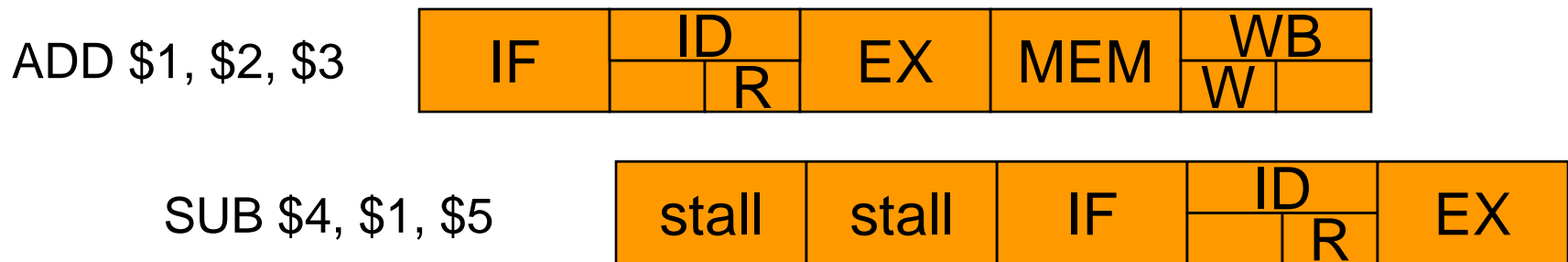# Data Hazard (RAW hazard)

ADD $1, $2, $3
SUB $4, $1, $5

Time

Solutions to Data Hazard

- ❑ Freezing the pipeline

- ❑ (Internal) Forwarding

- ❑ Compiler scheduling

Freezing The Pipeline

❑ **ALU result to next instruction**

ADD $1, $2, $3

IF	ID R	EX	MEM	WB W

SUB $4, $1, $5

stall	stall	IF	ID R	EX

❑ **Load result to next instruction**

LW $1, 32($6)

IF	ID R	EX	MEM	WB W

SUB $4, $1, $5

stall	stall	IF	ID R	EX
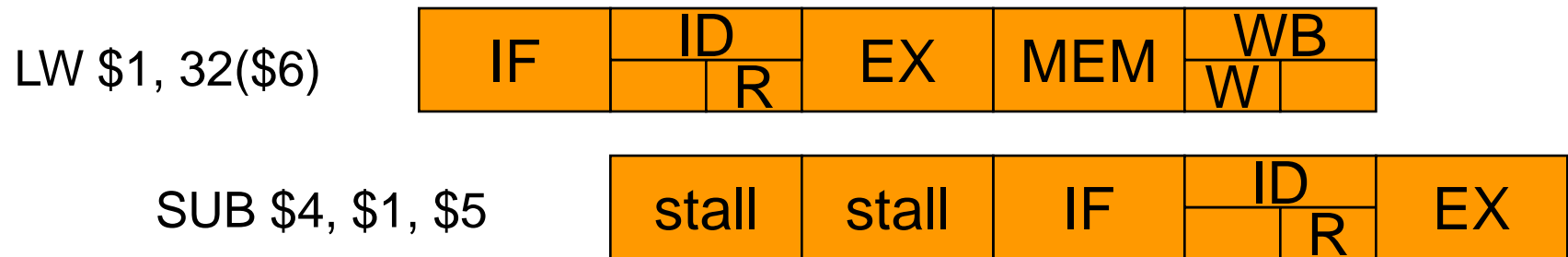
(Internal) Forwarding

❑ ALU result to next instruction (Stall X)

ADD $1, $2, $3

IF	ID R	EX	MEM	WB W

SUB $4, $1, $5

IF	ID R	EX	MEM	WB W

❑ Load result to next instruction (Stall 1)

LW $1, 32($6)

IF	ID R	EX	MEM	WB W

SUB $4, $1, $5
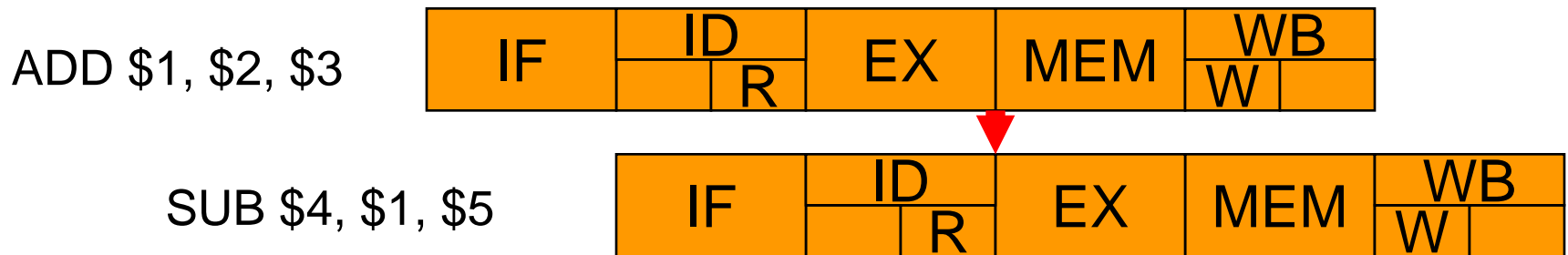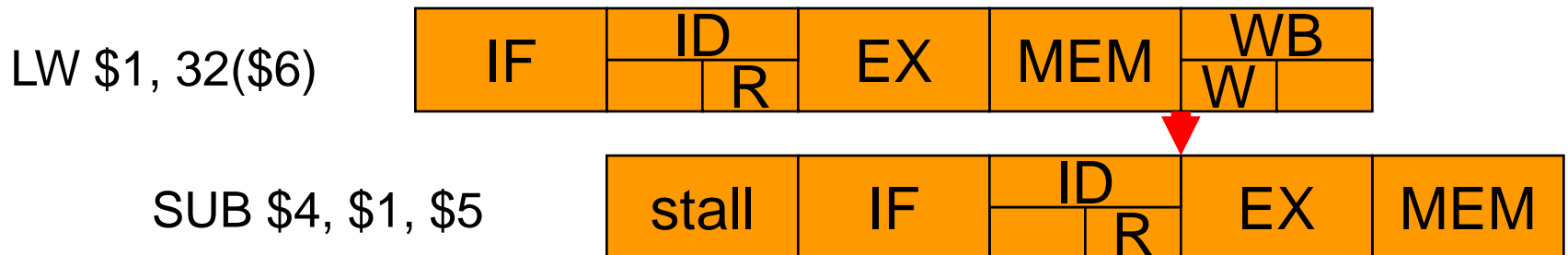
stall	IF	ID R	EX	MEM

Control Hazard

❑ Caused by PC-changing instructions
(Branch, Jump, Call/Return)

Branch instruction	IF	ID	EX	MEM	WB				
Branch successor		IF	stall	stall	IF	ID	EX	MEM	WB
Branch successor + 1						IF	ID	EX	MEM	WB
Branch successor + 2							IF	ID	EX	MEM
Branch successor + 3								IF	ID	EX
Branch successor + 4									IF	ID
Branch successor + 5										IF

For 5-stage pipeline, 3 cycle penalty
15% branch frequency. CPI = 1.45

Solutions to Control Hazard

❑ Optimized branch processing

❑ Branch prediction

❑ Delayed branch

Optimized Branch Processing

1. Find out branch <u>taken or not</u> early

    → simplified branch condition

2. Compute branch <u>target address</u> early

    → extra hardware

Branch Prediction

❑ **Predict-not-taken**

Untaken branch instruction	IF	ID	EX	MEM	WB				
Instruction $i + 1$		IF	ID	EX	MEM	WB			
Instruction $i + 2$			IF	ID	EX	MEM	WB		
Instruction $i + 3$				IF	ID	EX	MEM	WB	
Instruction $i + 4$					IF	ID	EX	MEM	WB

Taken branch instruction	IF	ID	EX	MEM	WB				
Instruction $i + 1$		IF	**idle**	**idle**	**idle**	**idle**			
Branch target			IF	ID	EX	MEM	WB		
Branch target + 1				IF	ID	EX	MEM	WB	
Branch target + 2					IF	ID	EX	MEM	WB

Delayed Branch

❑ **Semantics of delayed branch**

Untaken branch instruction	IF	ID	EX	MEM	WB				
Branch delay instruction ($i + 1$)		IF	ID	EX	MEM	WB			
Instruction $i + 2$			IF	ID	EX	MEM	WB		
Instruction $i + 3$				IF	ID	EX	MEM	WB	
Instruction $i + 4$					IF	ID	EX	MEM	WB

Taken branch instruction	IF	ID	EX	MEM	WB				
Branch delay instruction ($i + 1$)		IF	ID	EX	MEM	WB			
Branch target			IF	ID	EX	MEM	WB		
Branch target + 1				IF	ID	EX	MEM	WB	
Branch target + 2					IF	ID	EX	MEM	WB

Delayed Branch