

Advanced Computer Architecture

Lecture 3

Introduction to RISC instruction pipeline



NPTEL Online Course : Advanced Computer Architecture : Dr. John Jose, IIT Guwahati

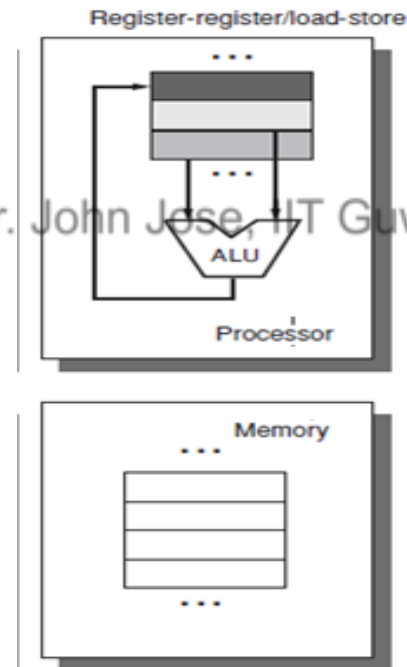
Dr. John Jose

Assistant Professor

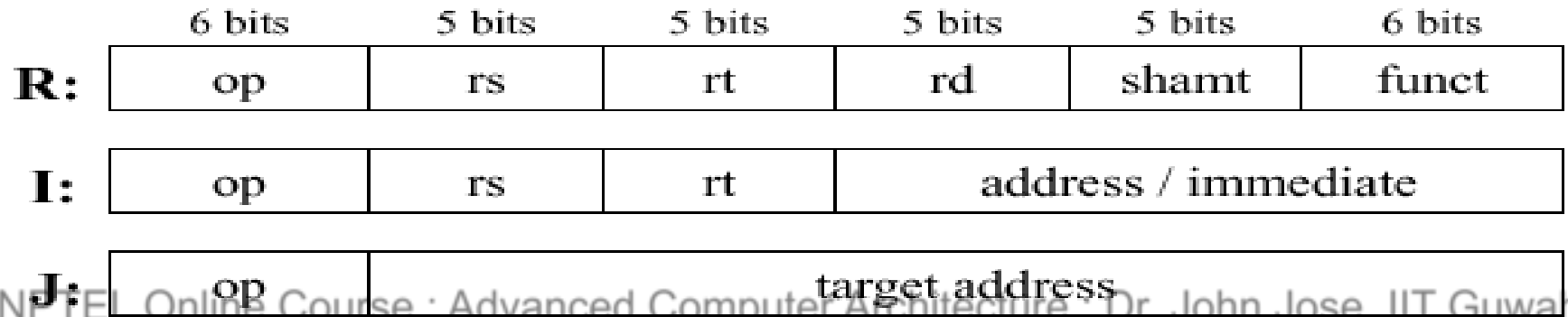
**Department of Computer Science & Engineering
Indian Institute of Technology Guwahati, Assam.**

Introduction to MIPS

- ❖ Microprocessor without Interlocked Pipelined Stages
- ❖ 32 registers (32 bit each)
- ❖ Uniform length instructions
- ❖ RISC- Load store architecture



Introduction to MIPS



op: basic operation of the instruction (opcode)

rs: first source operand register

rt: second source operand register

rd: destination operand register

shamt: shift amount

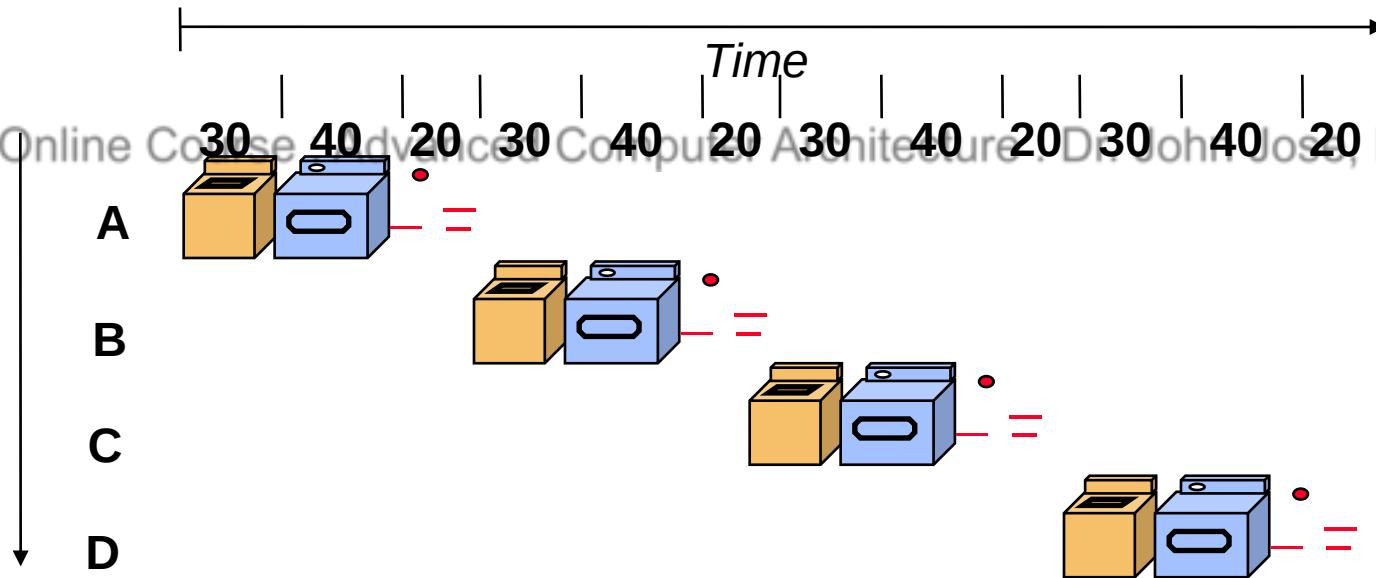
funct: selects the specific variant of the opcode (function code)

address: offset for load/store instructions ($\pm 2^{15}$)

immediate: constants for immediate instructions

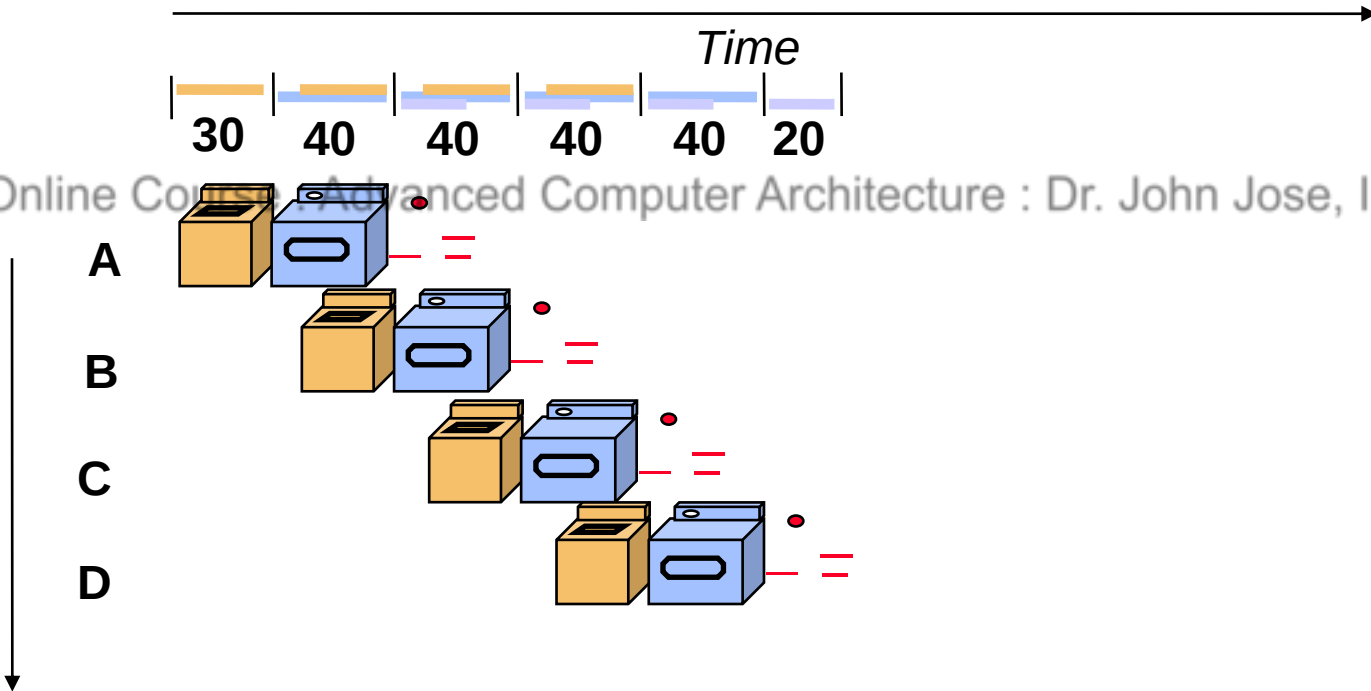
Unpipelined Work flow

- ❖ Start work when previous one is fully over
- ❖ Sequential laundry takes 6 hours for 4 loads



Pipelined Work flow

- ❖ Start work as soon as possible
- ❖ Pipelined laundry takes 3.5 hours for 4 loads



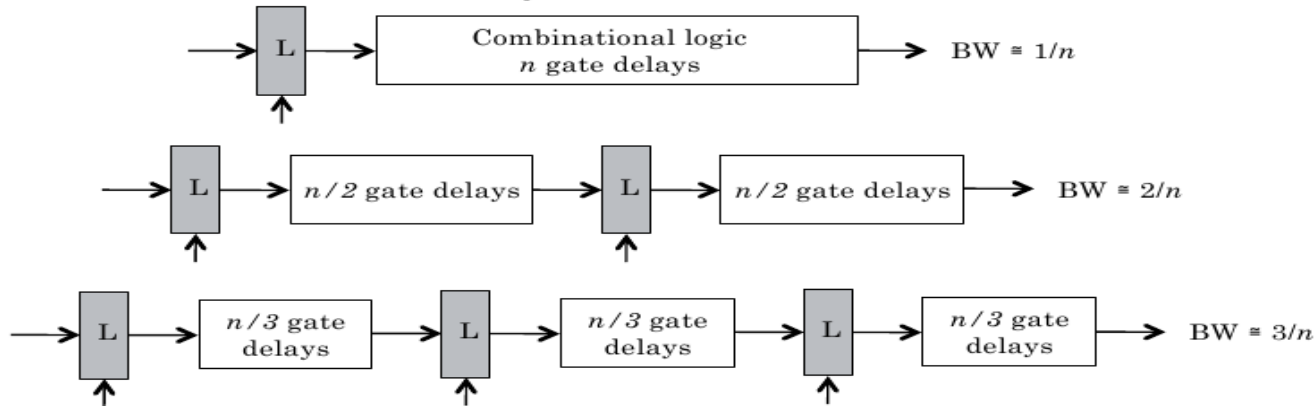
Pipelining Characteristics

- ❖ Pipelining doesn't reduce latency of single task, it improves throughput of entire workload
- ❖ Pipeline rate limited by slowest pipeline stage
- ❖ Potential speedup = Number of pipe stages
- ❖ Unbalanced lengths of pipe stages reduces speedup
- ❖ Time to fill pipeline and time to drain it reduces speedup

Pipelining in Circuits

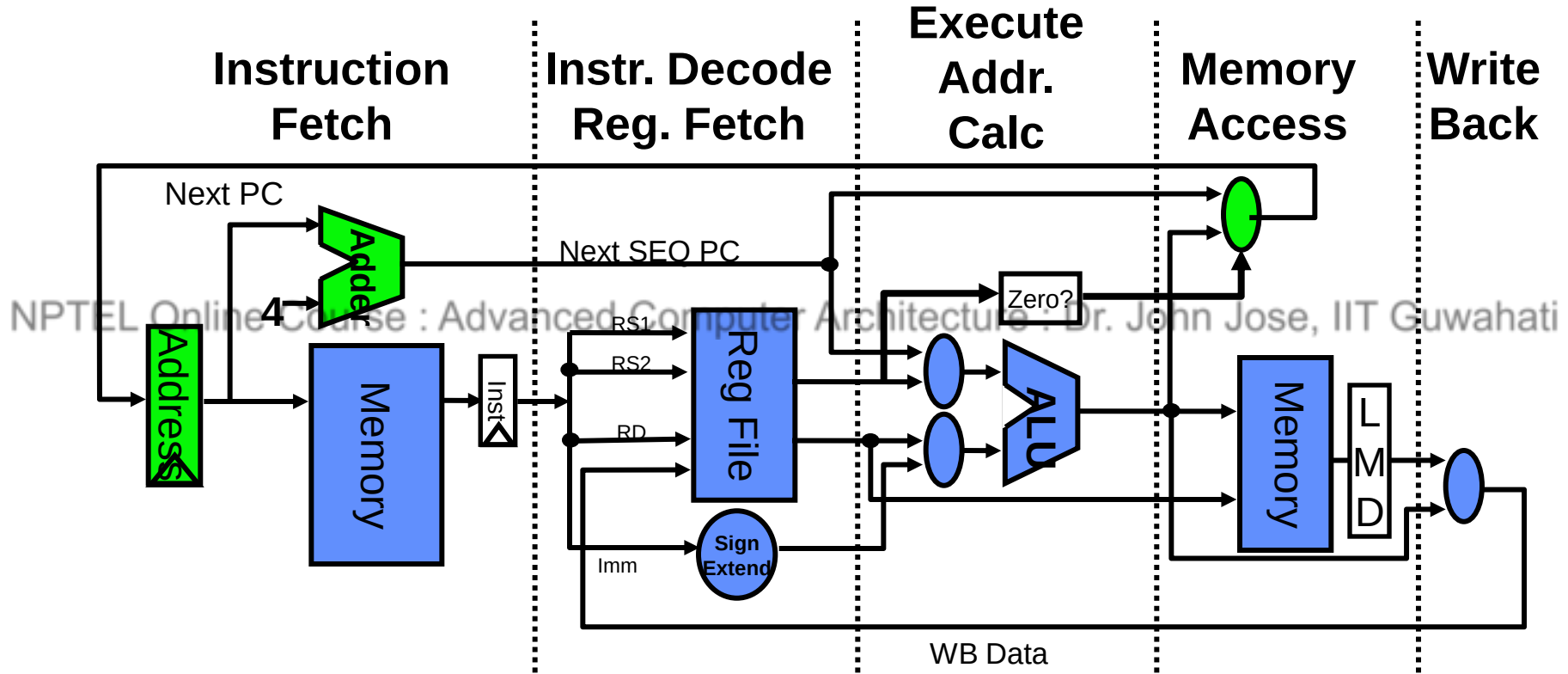
- ❖ Pipelining partitions the system into multiple independent stages with added buffers between the stages.
- ❖ Pipelining can increase the throughput of a system.

NPTEL Online Course : Advanced Computer Architecture : Dr. John Jose, IIT Guwahati



Potential k -fold increase of throughput in a k -stage pipelined system

Unpipelined RISC Data path



Pipelined RISC Data path

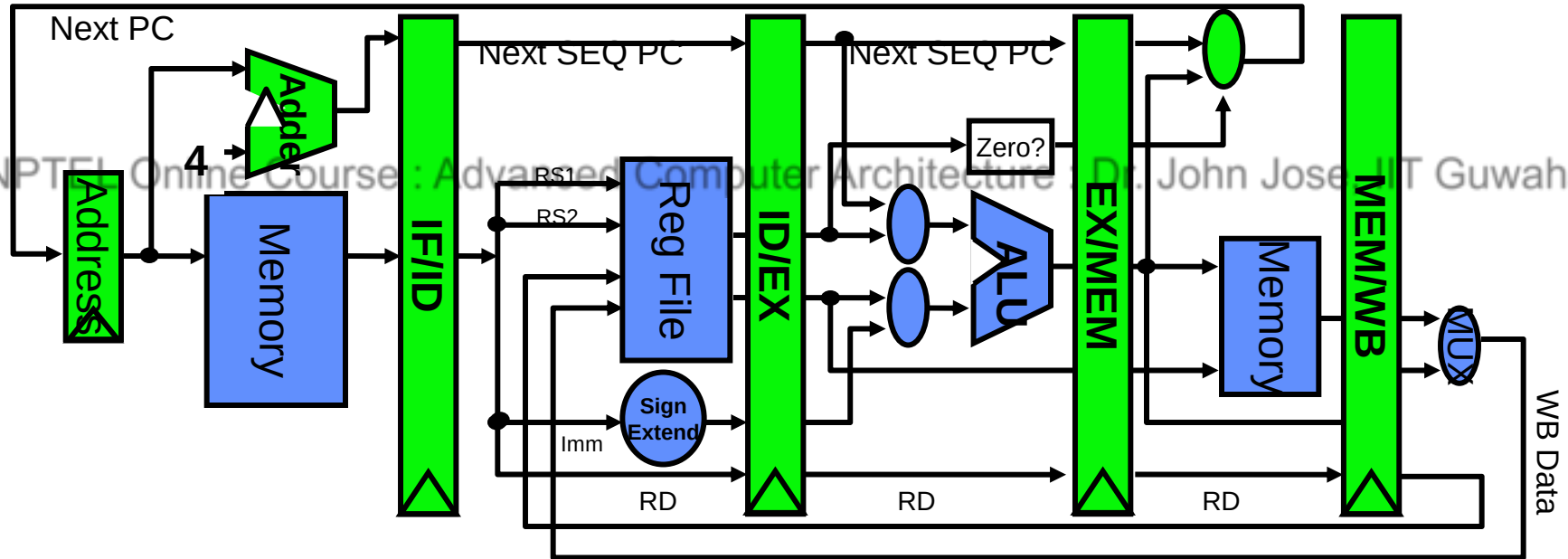
Instruction
Fetch

Instr. Decode
Reg. Fetch

Execute
Addr.
Calc

Memory
Access

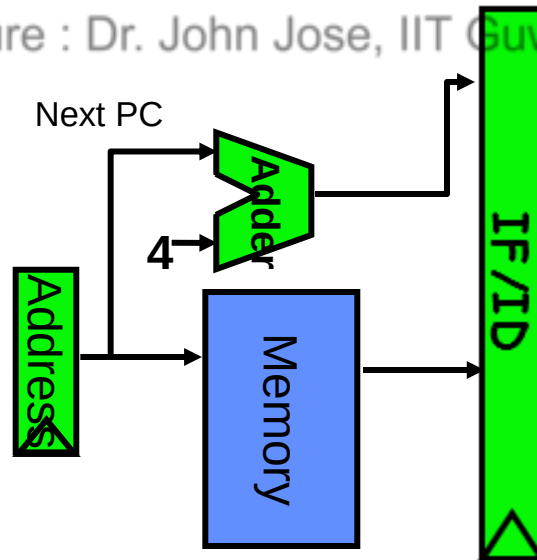
Write
Back



RISC MIPS Instruction Pipeline

- ❖ Each instruction can take at most 5 clock cycles
- ❖ **Instruction fetch cycle (IF)**
 - ❖ Based on PC, fetch the instruction from memory
 - ❖ Increment PC

NPTEL Online Course : Advanced Computer Architecture : Dr. John Jose, IIT Guwahati



RISC MIPS Instruction Pipeline

- ❖ **Instruction decode/register fetch cycle (ID)**
- ❖ Decode the instruction + register read operation
- ❖ Fixed field decoding

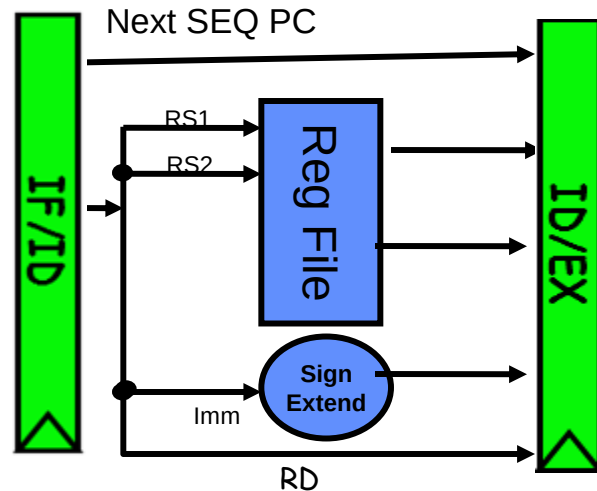
NPTEL Online Course: Advanced Computer Architecture : Dr. John Jose, IIT Guwahati

Ex: [ADD R1,R2,R3] : A3.01.02.03

10100011 00000001 00000010 00000011

Ex: [LW R1,8(R2)] : 86.01.02.03

10000110 00000001 00001000 00000010



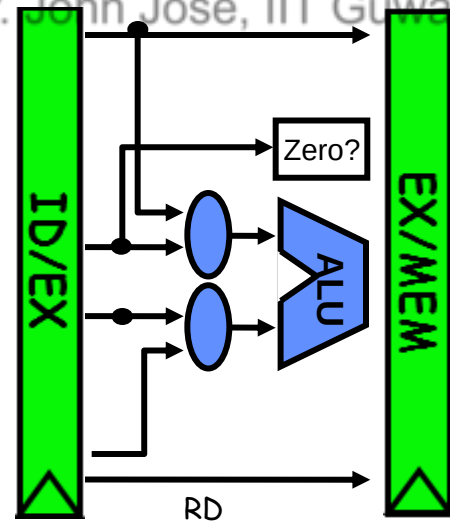
RISC MIPS Instruction Pipeline

- ❖ **Execution/Effective address cycle (EX)**
- ❖ Memory reference: Calculate the effective address

[LW R1,8(R2)] EFF ADDR= [R2] +8

- ❖ Register-register ALU instruction

[ADD R1,R2,R2]

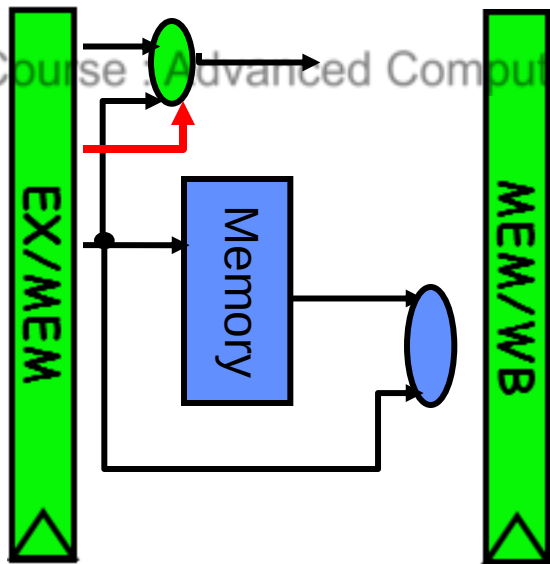


RISC MIPS Instruction Pipeline

❖ Memory access cycle (MEM)

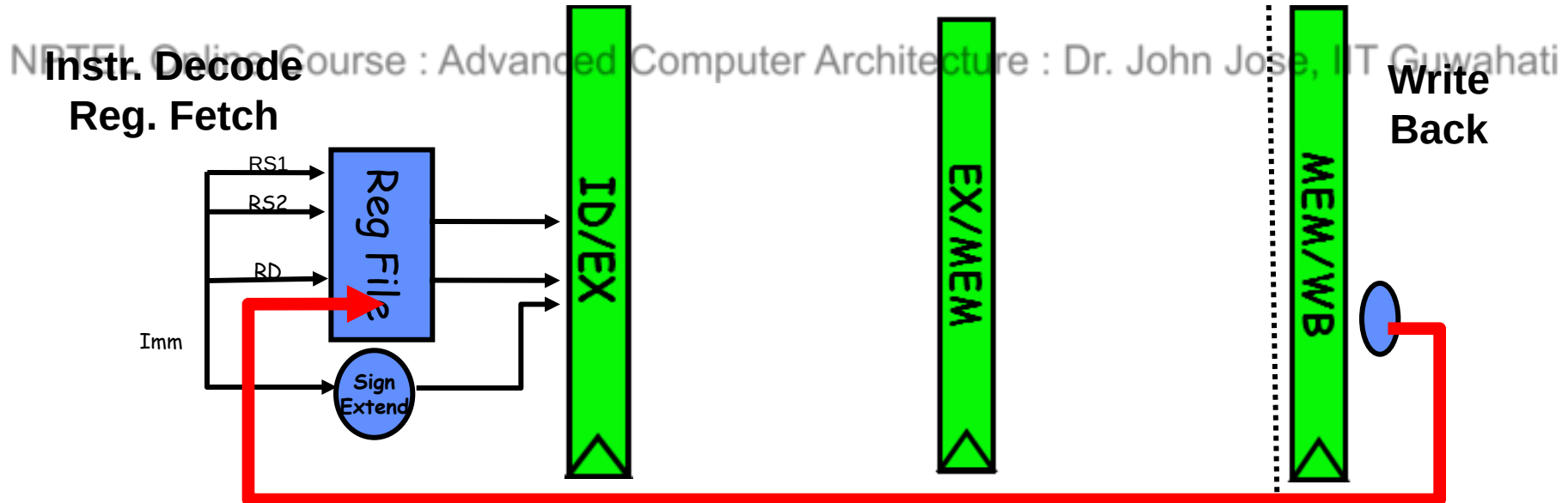
❖ Load from memory and store in register [LW R1,8(R2)]

❖ Store the data from the register to memory [SW R3,16(R4)]



RISC Instruction Pipeline

- ❖ **Write-back cycle (WB)**
- ❖ Register-register ALU instruction or load instruction
- ❖ Write to register file **[LW R1,8(R2)]** , **[ADD R1,R2,R3]**



Pipelined RISC Data path

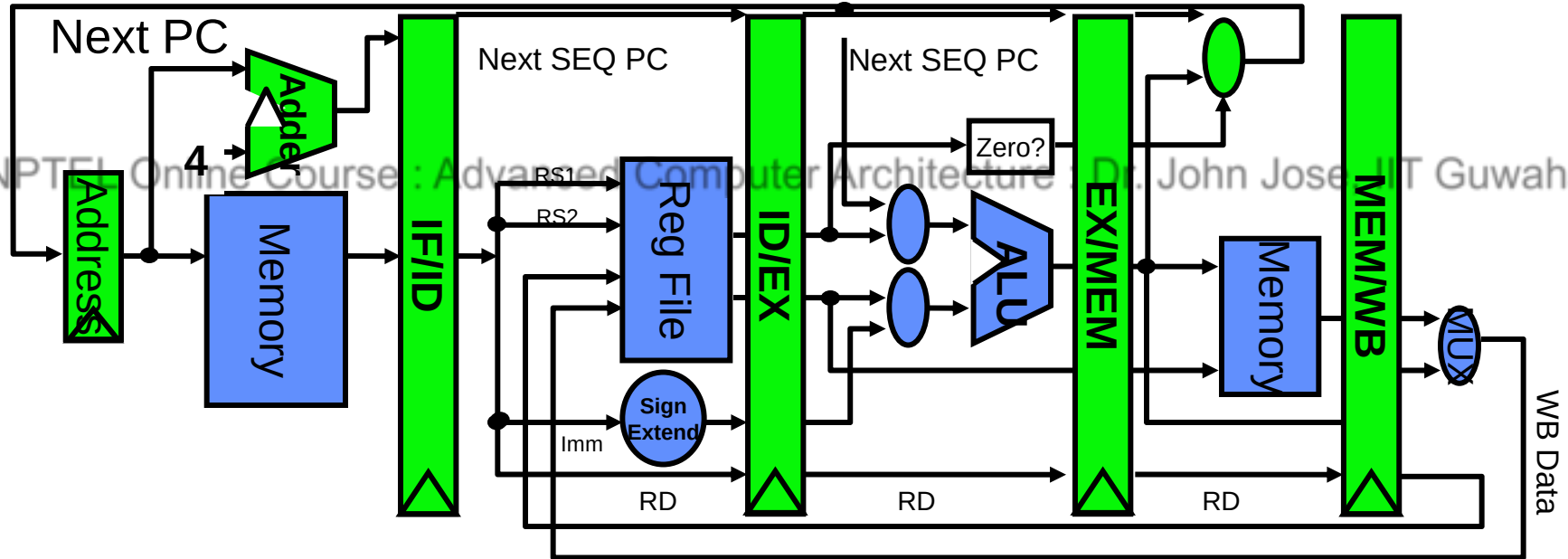
Instruction
Fetch

Instr. Decode
Reg. Fetch

Execute
Addr.
Calc

Memory
Access

Write
Back



5 Steps of RISC Data path

- ❖ Each instruction can take at most 5 clock cycles
- ❖ **Instruction fetch cycle (IF)**
 - ❖ Based on PC value, fetch the instruction from memory

- ❖ Update PC

NPTEL Online Course : Advanced Computer Architecture : Dr. John Jose, IIT Guwahati

- ❖ **Instruction decode/register fetch cycle (ID)**
 - ❖ Decode the instruction + register read operation
 - ❖ Fixed field decoding
 - ❖ Equality check of registers
 - ❖ Computation of branch target address if any

5 Steps of MIPS Data path

❖ Execution/Effective address cycle (EX)

- ❖ Memory reference: Calculate the effective address
- ❖ Register-register ALU instruction
- ❖ Register-immediate ALU instruction

❖ Memory access cycle (MEM)

- ❖ Load instruction: Read from memory using effective address
- ❖ Store instruction: Write the data in the register to memory using effective address

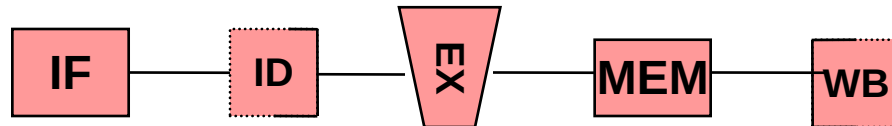
5 Steps of MIPS Data path

❖ Write-back cycle (WB)

- ❖ Register-register ALU instruction or load instruction
- ❖ Write the result into the register file

❖ Cycles required to implement different instructions

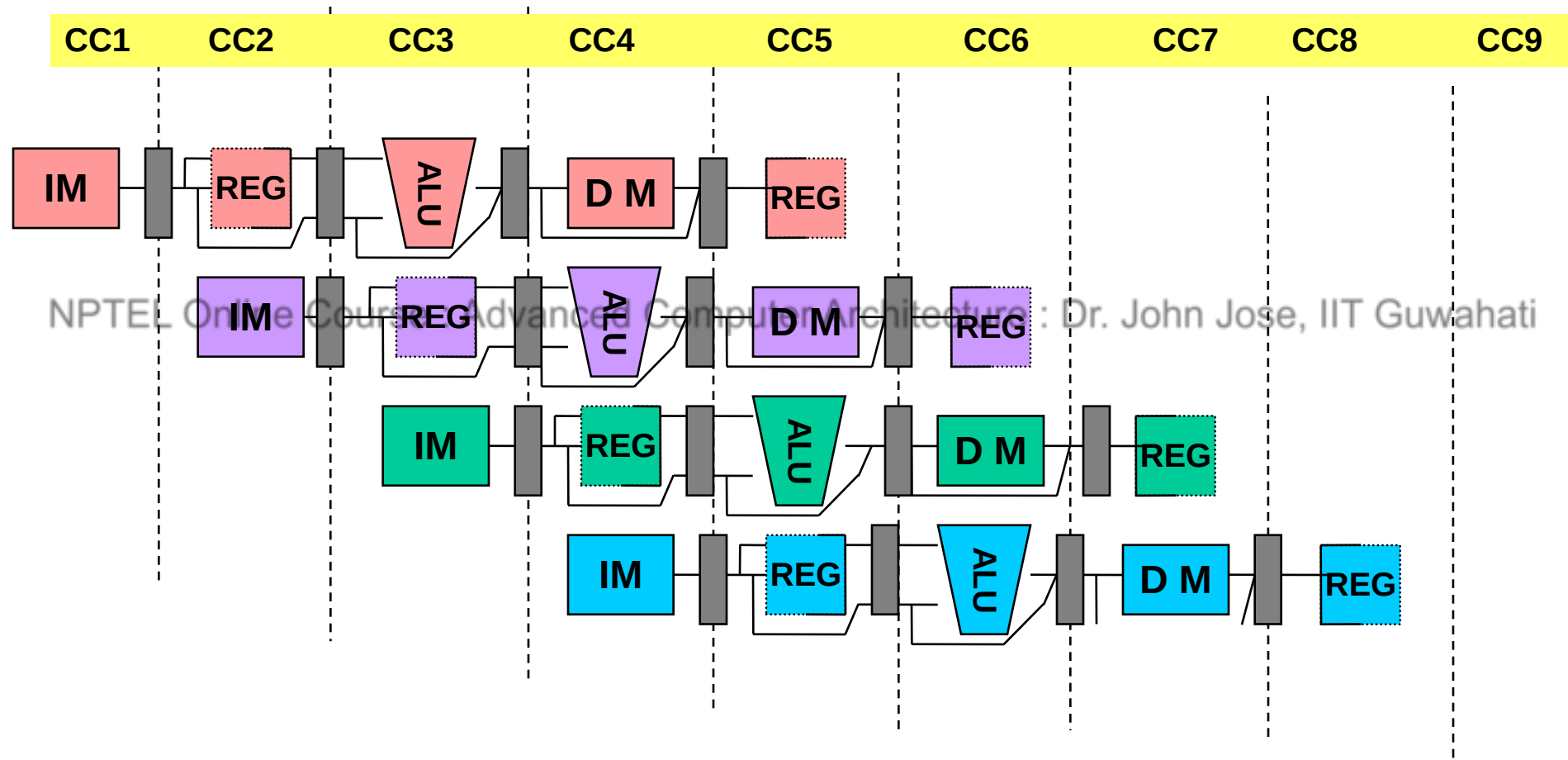
- ❖ Branch instructions – 4 cycles
- ❖ Store instructions – 4 cycles
- ❖ All other instructions – 5 cycles



Visualizing Pipelining

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

Visualizing Pipelining



Pipelining Issues

❖ Ideal Case: Uniform sub-computations

- ❖ The computation to be performed can be evenly partitioned into uniform-latency sub-computations

❖ Reality: Internal fragmentation

- ❖ Not all pipeline stages may have the uniform latencies

❖ Impact of ISA

- ❖ Memory access is a critical sub-computation
- ❖ Memory addressing modes should be minimized
- ❖ Fast cache memories should be employed

Pipelining Issues

❖ Ideal Case : Identical computations

- ❖ The same computation is to be performed repeatedly on a large number of input data sets

❖ Reality: External fragmentation

- ❖ Some pipeline stages may not be used

❖ Impact of ISA

- ❖ Reduce the complexity and diversity of the instruction types
- ❖ RISC architectures use uniform stage simple instructions

Pipelining Issues

- ❖ **Ideal Case : Independent computations**

- ❖ All the instructions are mutually independent

- ❖ **Reality: Pipeline stalls – cannot proceed.**

- ❖ A later computation may require the result of an earlier computation

- ❖ **Impact of ISA**

- ❖ Reduce Memory addressing modes - dependency detection

- ❖ Use register addressing mode - easy dependencies check

**THANK
YOU!**

FOR

YOUR

PRESENCE

NPTEL Online Course : Advanced Computer Architecture : Dr. John Jose, IIT Guwahati

johnjose@iitg.ac.in

<http://www.iitg.ac.in/johnjose/>