

Advanced Computer Architecture

Lecture 4

Pipeline Hazards



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.**

Pipelined RISC Data path

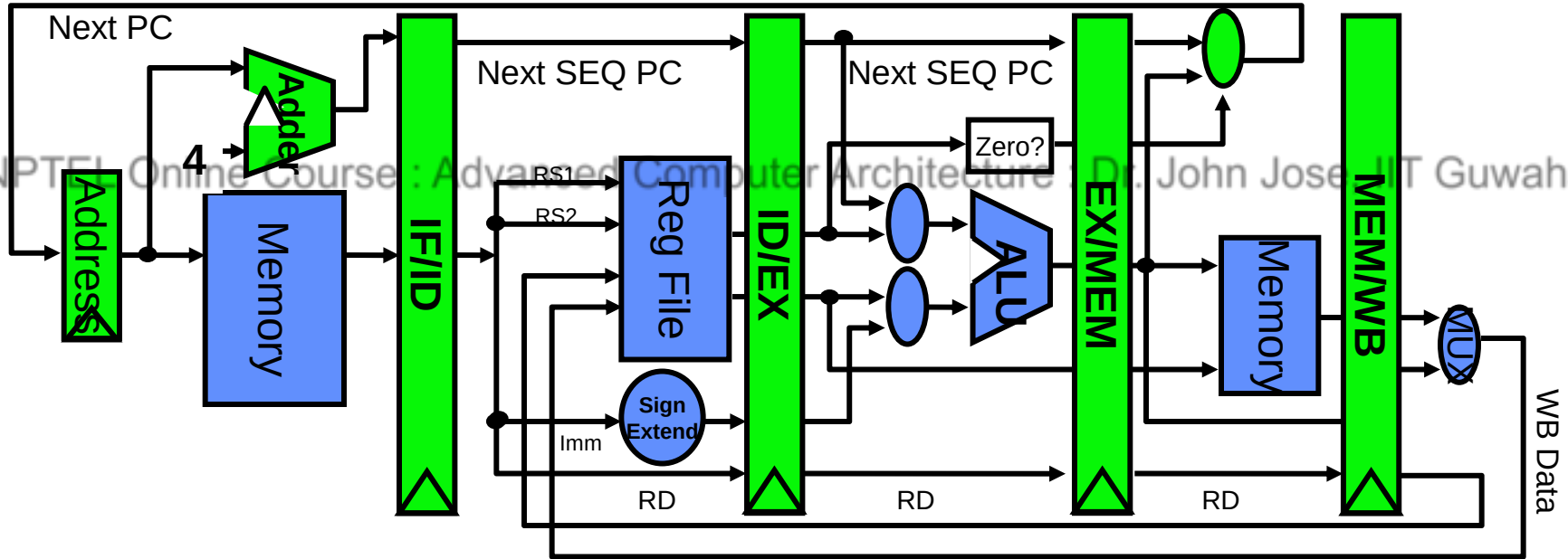
Instruction
Fetch

Instr. Decode
Reg. Fetch

Execute
Addr.
Calc

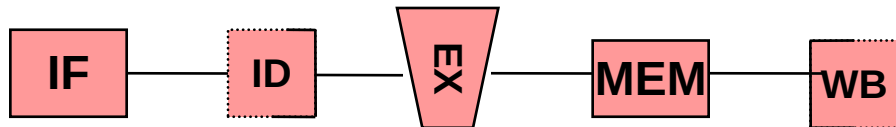
Memory
Access

Write
Back



5 Steps of RISC Data path

- ❖ Each instruction can take at most 5 clock cycles
- ❖ **Instruction fetch cycle (IF)**
- ❖ **Instruction decode/register fetch cycle (ID)**
- ❖ **Execution/Effective address cycle (EX)**
- ❖ **Memory access cycle (MEM)**
- ❖ **Write-back cycle (WB)**



Visualizing Pipelining

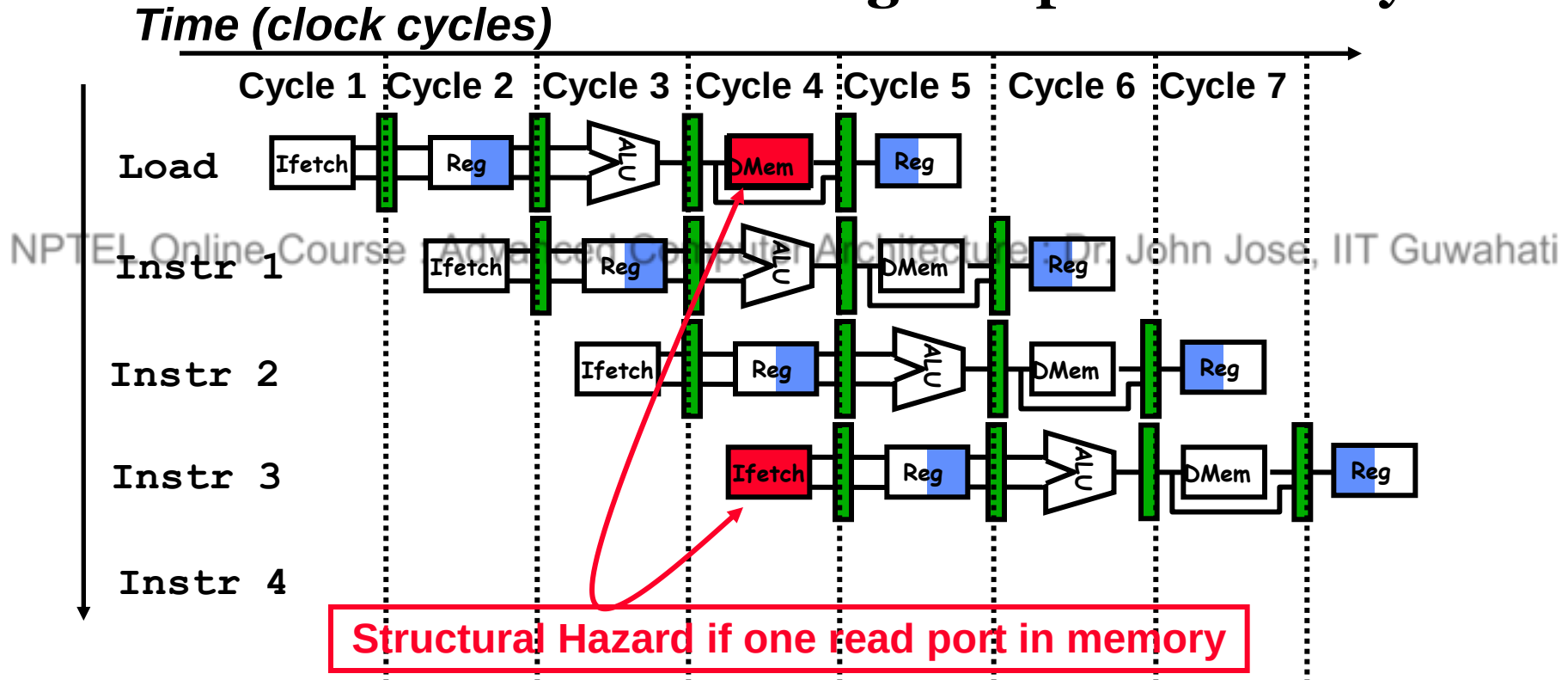
Clock number								
Instruction number	1	2	3	4	5	6	7	8
i	IF	ID	EX	MEM	WB			
$i+1$		IF	ID	EX	MEM	WB		
$i+2$			IF	ID	EX	MEM	WB	
$i+3$				IF	ID	EX	MEM	WB
$i+4$					IF	ID	EX	MEM

Limits to pipelining

- ❖ **Hazards:** circumstances that would cause incorrect execution if next instruction is fetched and executed
 - ❖ **Structural hazards:** Different instructions, at different stages, in the pipeline want to use the same hardware resource
 - ❖ **Data hazards:** An instruction in the pipeline requires data to be computed by a previous instruction still in the pipeline
 - ❖ **Control hazards:** Succeeding instruction, to put into pipeline, depends on the outcome of a previous branch instruction, already in pipeline

Structural Hazard

Eg: Uniport Memory



Resolving Structural Hazard

- ❖ Eliminate the use same hardware for two different things at the same time

- ❖ **Solution 1: Wait**

 - ❖ must detect the hazard

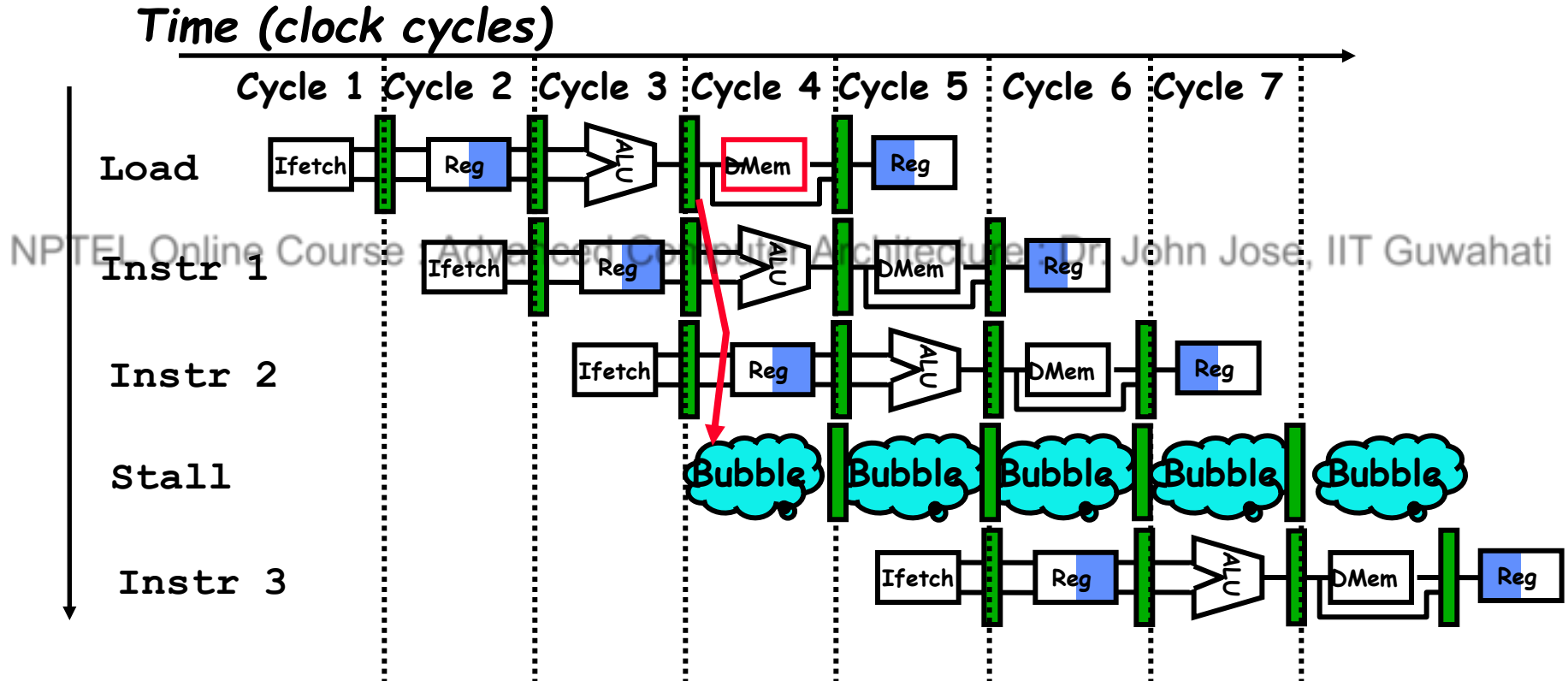
NPTEL Online Course : Advanced Computer Architecture : Dr. John Jose, IIT Guwahati

 - ❖ must have mechanism to stall

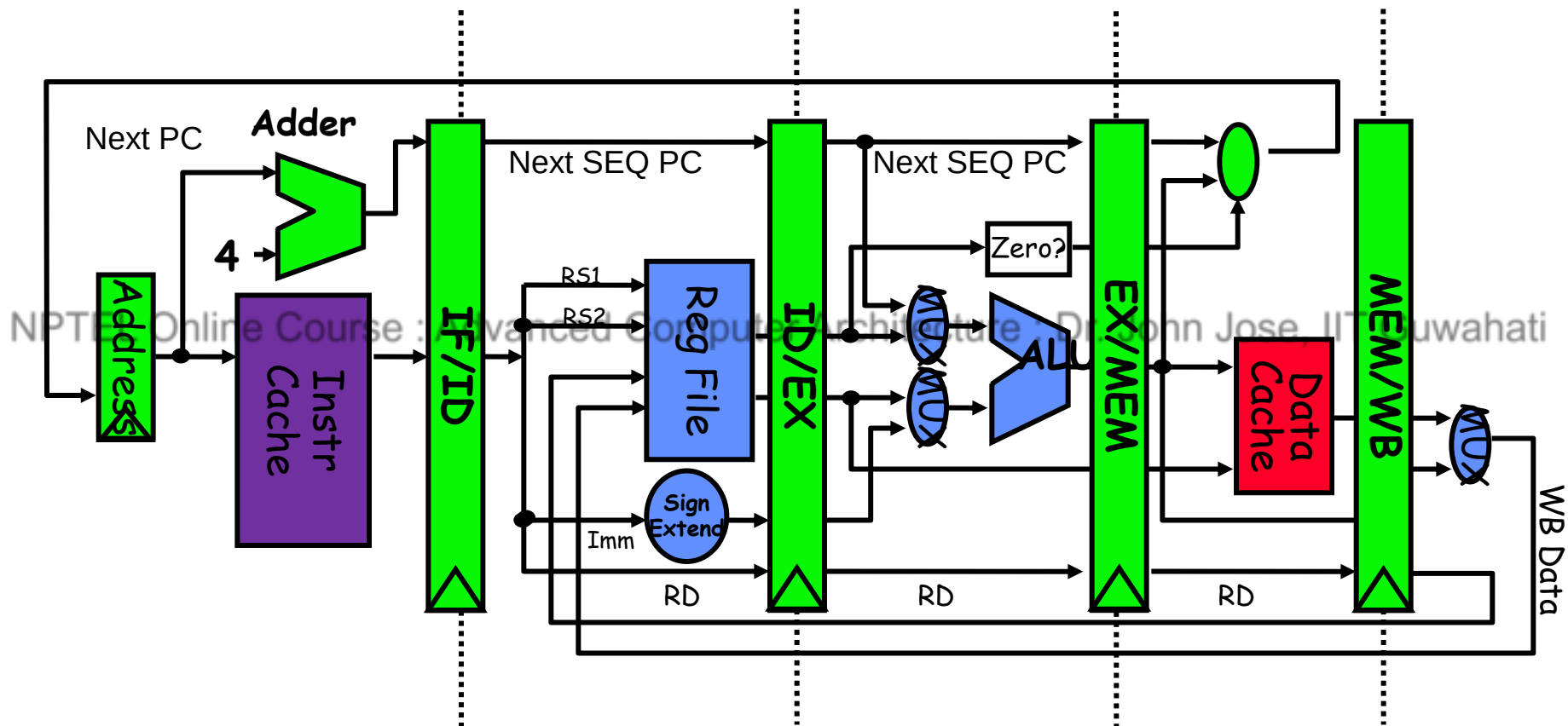
- ❖ **Solution 2: Duplicate hardware**

 - ❖ Multiple such units will help both instruction to progress

Detecting & Resolving Structural Hazard

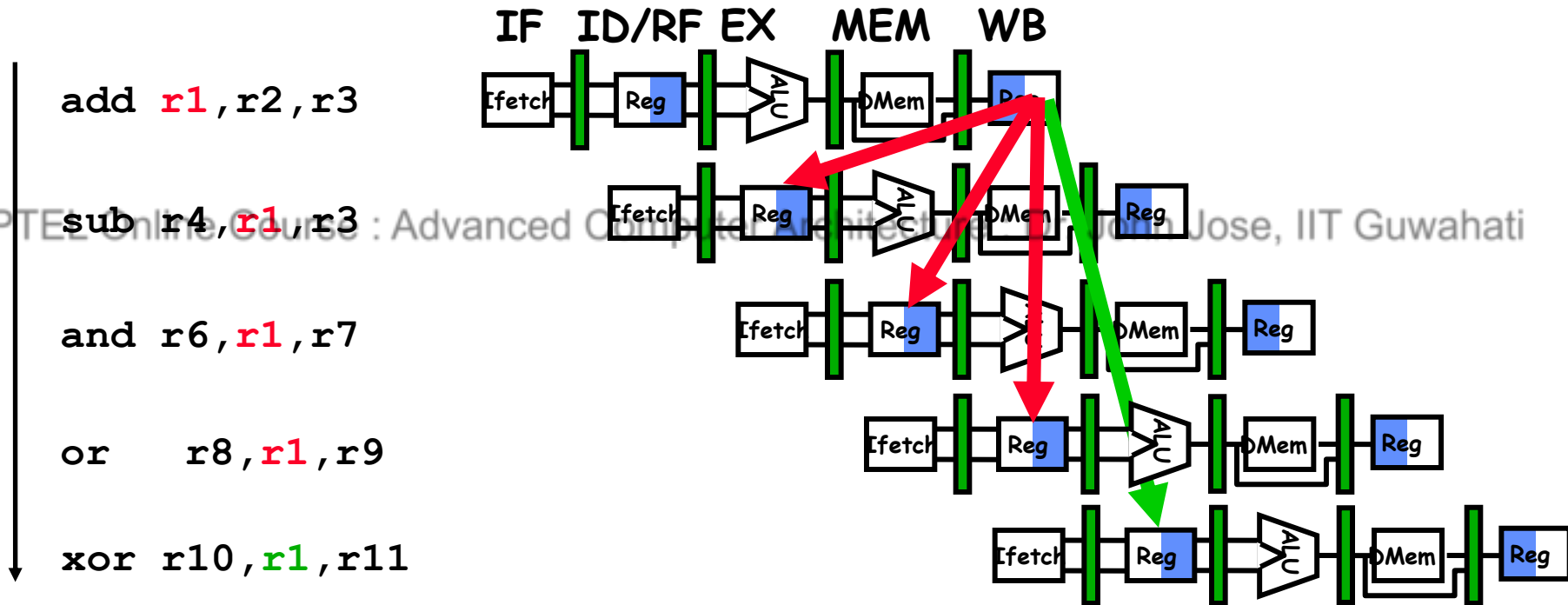


Eliminating Structural Hazards at Design



Data Hazard

Time (clock cycles) →

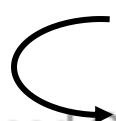


Three Generic Data Hazards

❖ Read After Write (RAW)

Instr_j tries to read operand **before** Instr_i writes it

I : add **r1**,r2,r3
J: sub r4,**r1**,r3



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

❖ Caused by a data dependence

❖ This hazard results from an actual need for communication.

Three Generic Data Hazards

❖ Write After Read (WAR)

Instr_j writes operand **before** Instr_i reads it

 I: sub r4, **r1**, r3

J: add **r1**, r2, r3

K: mul r6, r1, r7

❖ Called an anti-dependence by compiler writers.

❖ This results from reuse of the name r1

❖ Can't happen in MIPS 5 stage pipeline because:

❖ All instructions take 5 stages, and

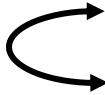
❖ Reads are always in stage 2, and

❖ Writes are always in stage 5

Three Generic Data Hazards

❖ Write After Write (WAW)

Instr_j writes operand **before** Instr_i writes it.

 I: sub **r1**,r4,r3
J: add **r1**,r2,r3

❖ Called an output dependence

❖ This also results from the reuse of name r1. K: mul r6,r1,r7

❖ Can't happen in MIPS 5 stage pipeline because:

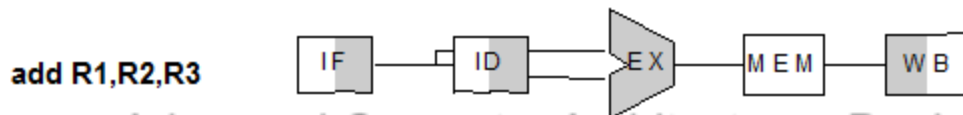
❖ All instructions take 5 stages, and

❖ Writes are always in stage 5

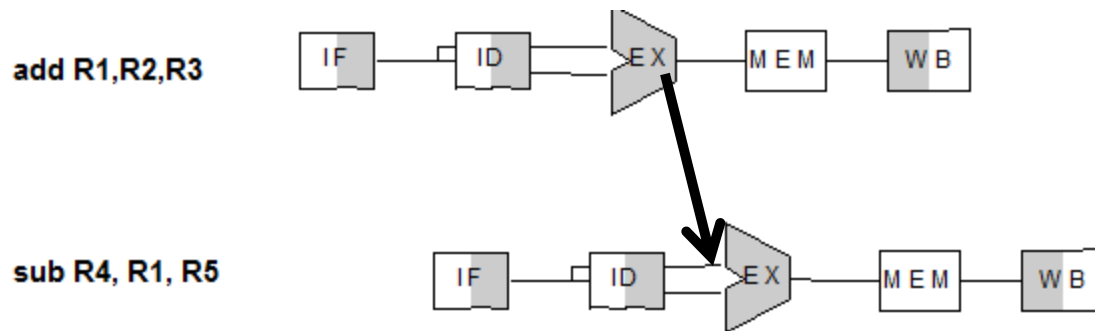
❖ WAR and WAW happens in out of order pipes

How to Handle Data Hazard ?

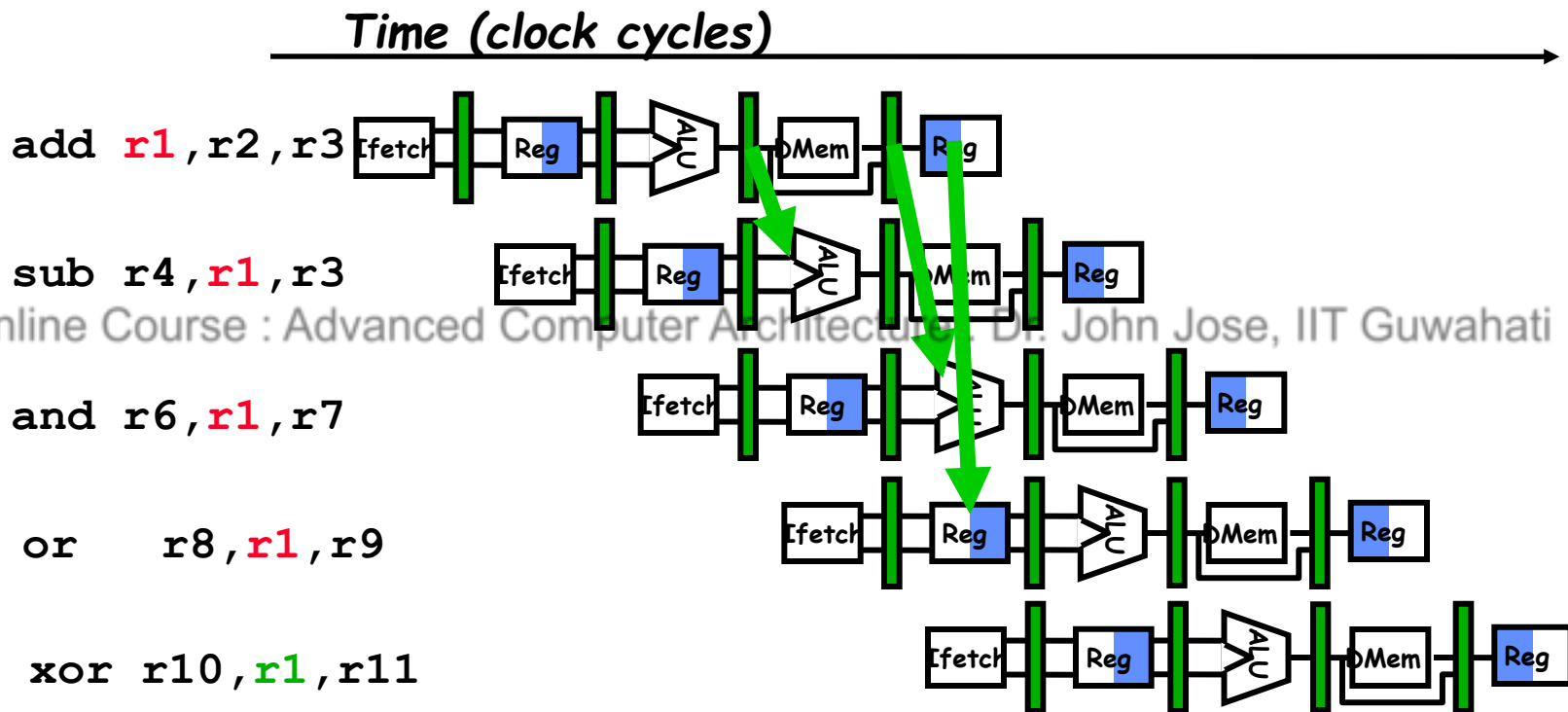
- ❖ Data hazard: instruction needs data from the result of a previous instruction still executing in pipeline
- ❖ **Solution:** Forward data if possible.



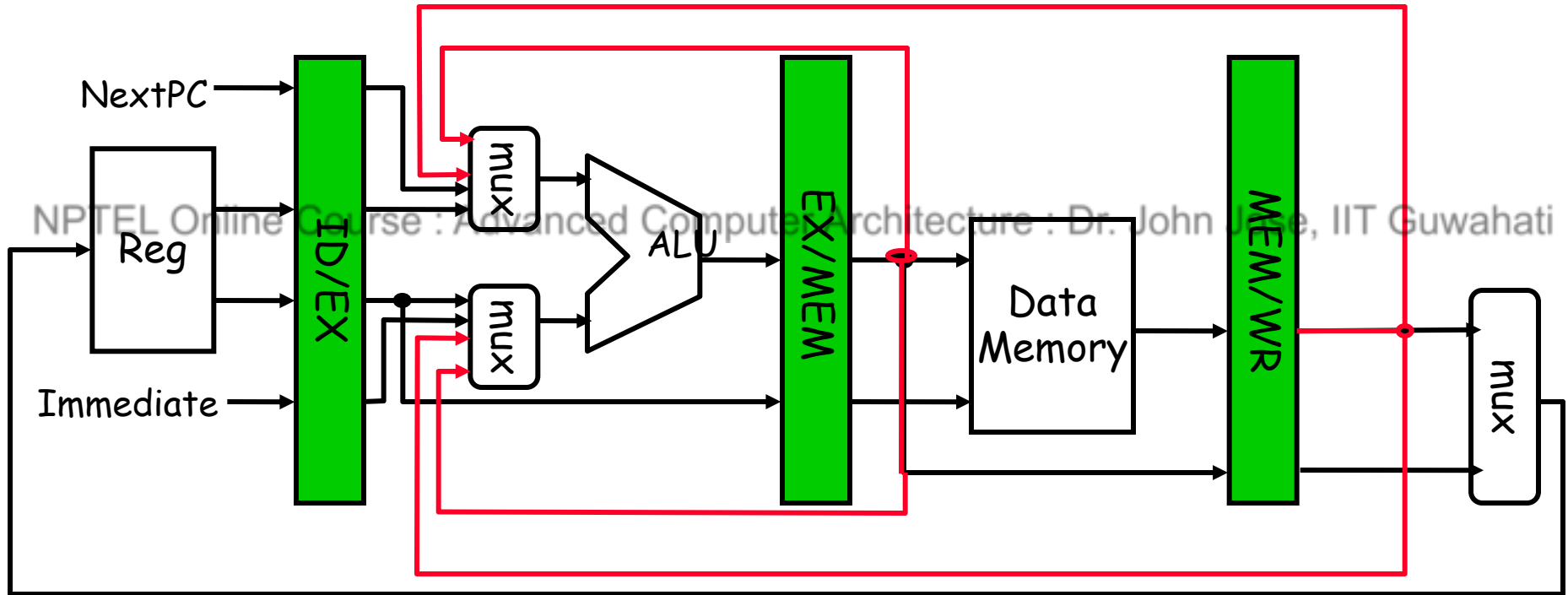
NPTEL Online Course : Advanced Computer Architecture : Dr. John Jose, IIT Guwahati



Operand Forwarding to Avoid Data Hazard



Hardware Change for Forwarding



Data Hazard even with Operand Forwarding

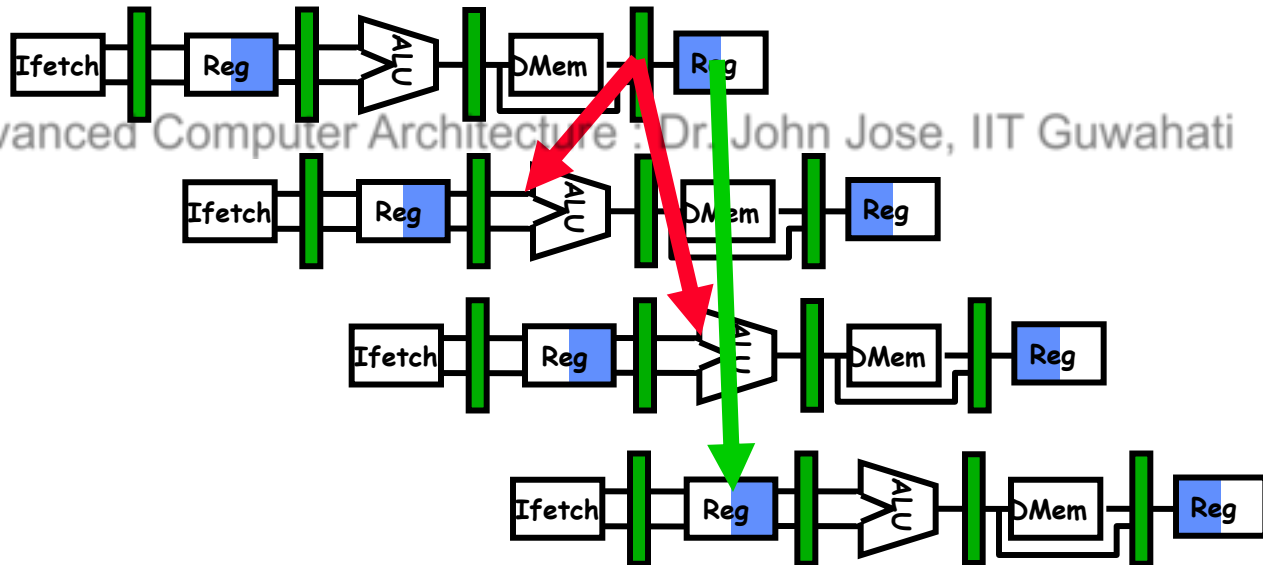
Time (clock cycles) →

lw r1, 0(r2)

sub r4, r1, r6

and r6, r1, r7

or r8, r1, r9



Resolving the Load-ALU Hazard

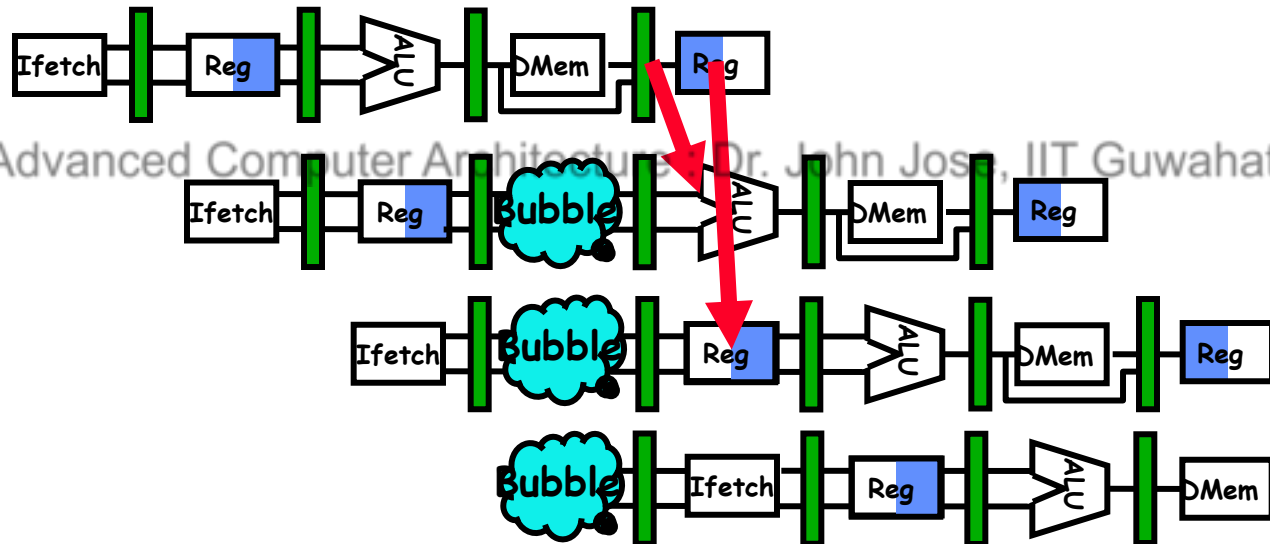
Time (clock cycles) →

lw r1, 0(r2)

sub r4, r1, r6

and r6, r1, r7

or r8, r1, r9



Software Scheduling for Load Hazards

Assume a, b, c, d, e, and f in memory.

a = b + c;

d = e - f;

LW Rb,b

LW Rc,c

ADD Ra,Rb,Rc

SW a,Ra

LW Re,e

LW Rf,f

SUB Rd,Re,Rf

SW d,Rd

LW Rb,b

LW Rc,c

LW Re,e

ADD Ra,Rb,Rc

LW Rf,f

SW a,Ra

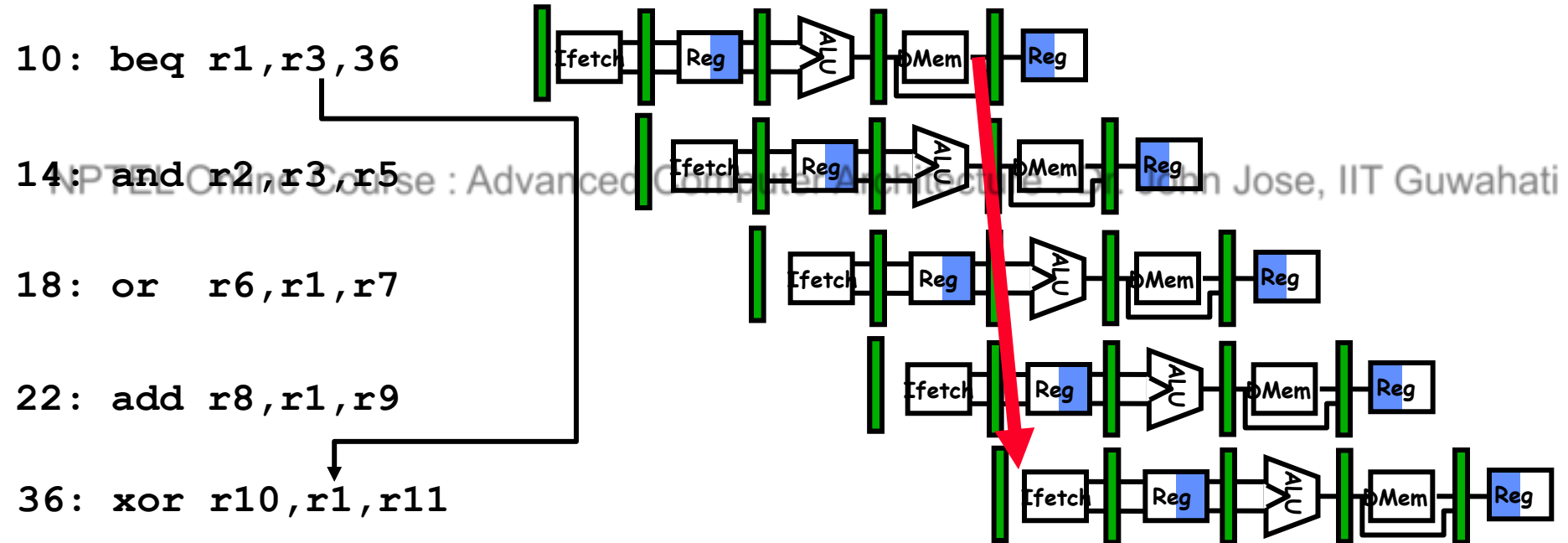
SUB Rd,Re,Rf

SW d,Rd

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

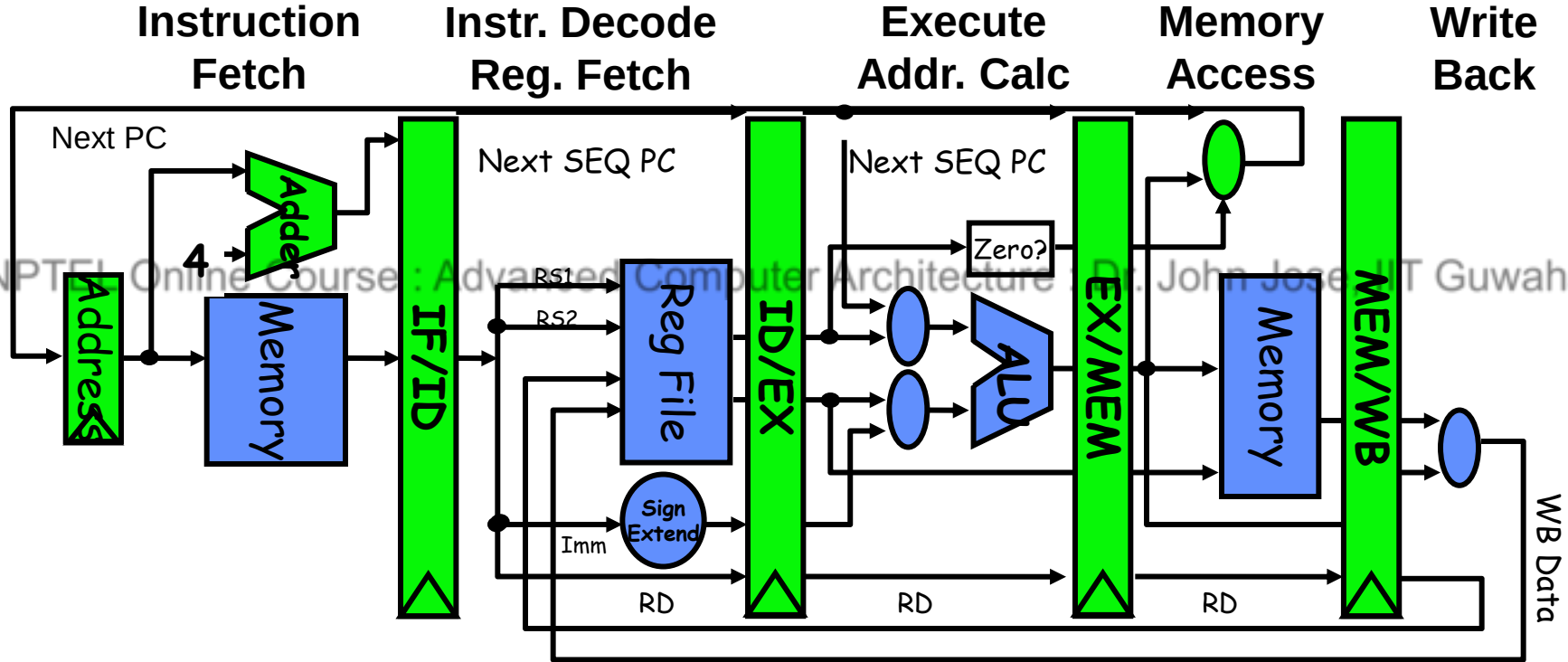
Control Hazard on Branches

=> Three Stage Stall



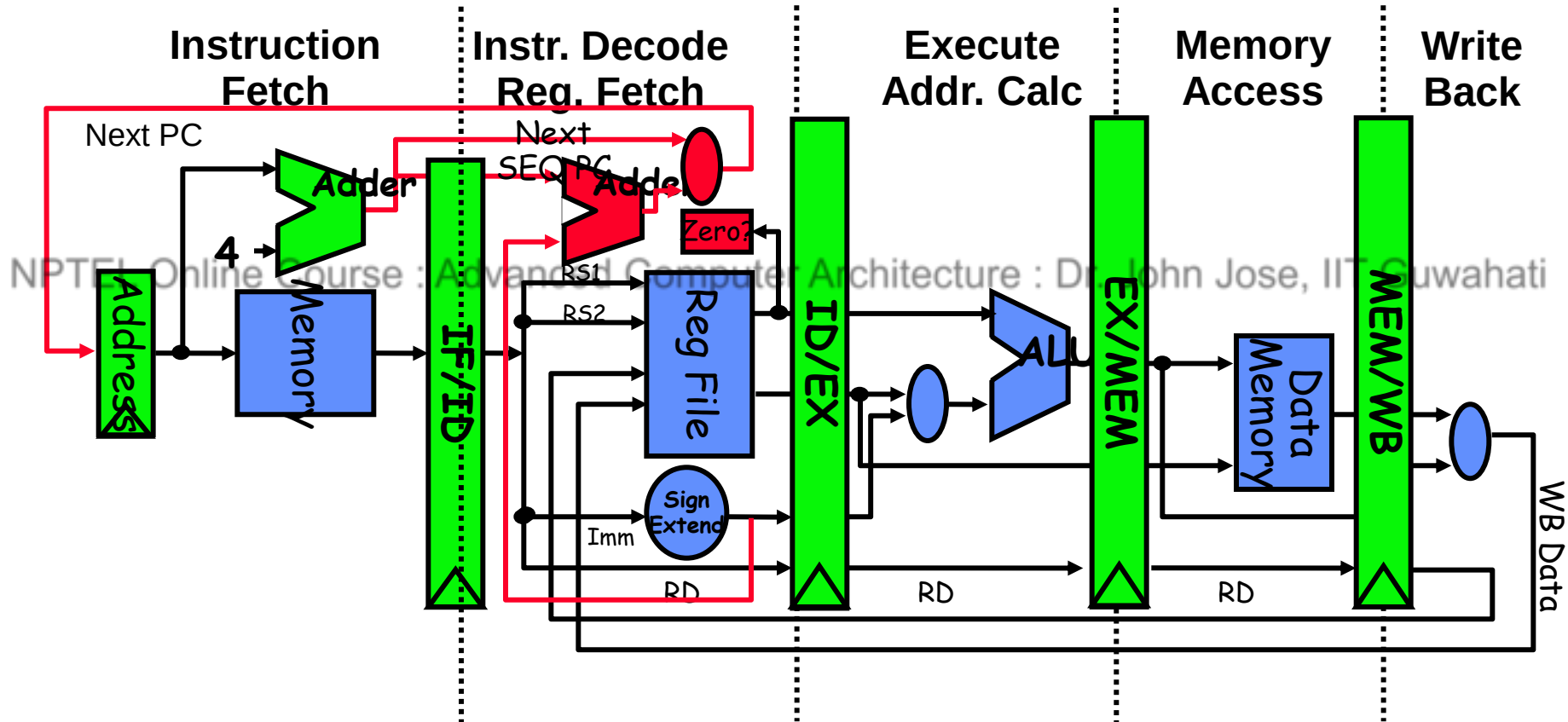
Conventional MIPS Pipeline

Branching at 4th stage



Branch Optimized MIPS Pipeline

Branching at 2nd stage



**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/>