RISC Design:

Multi-Cycle Implementation

Virendra Singh

Associate Professor

Computer Architecture and Dependable Systems Lab

Department of Electrical Engineering

Indian Institute of Technology Bombay

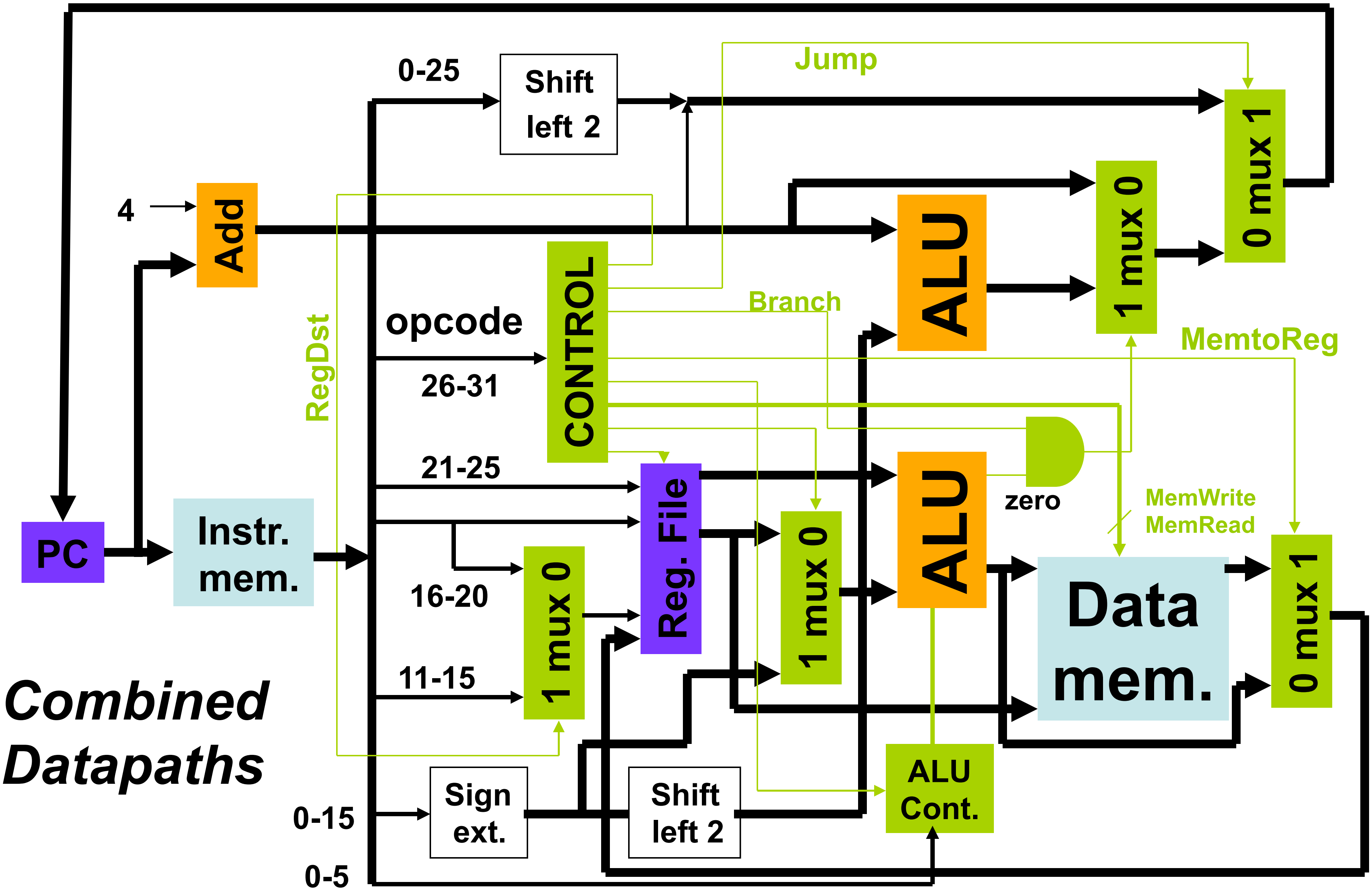
http://www.ee.iitb.ac.in/~viren/

E-mail: viren@ee.iitb.ac.in

CP-226: Computer Architecture

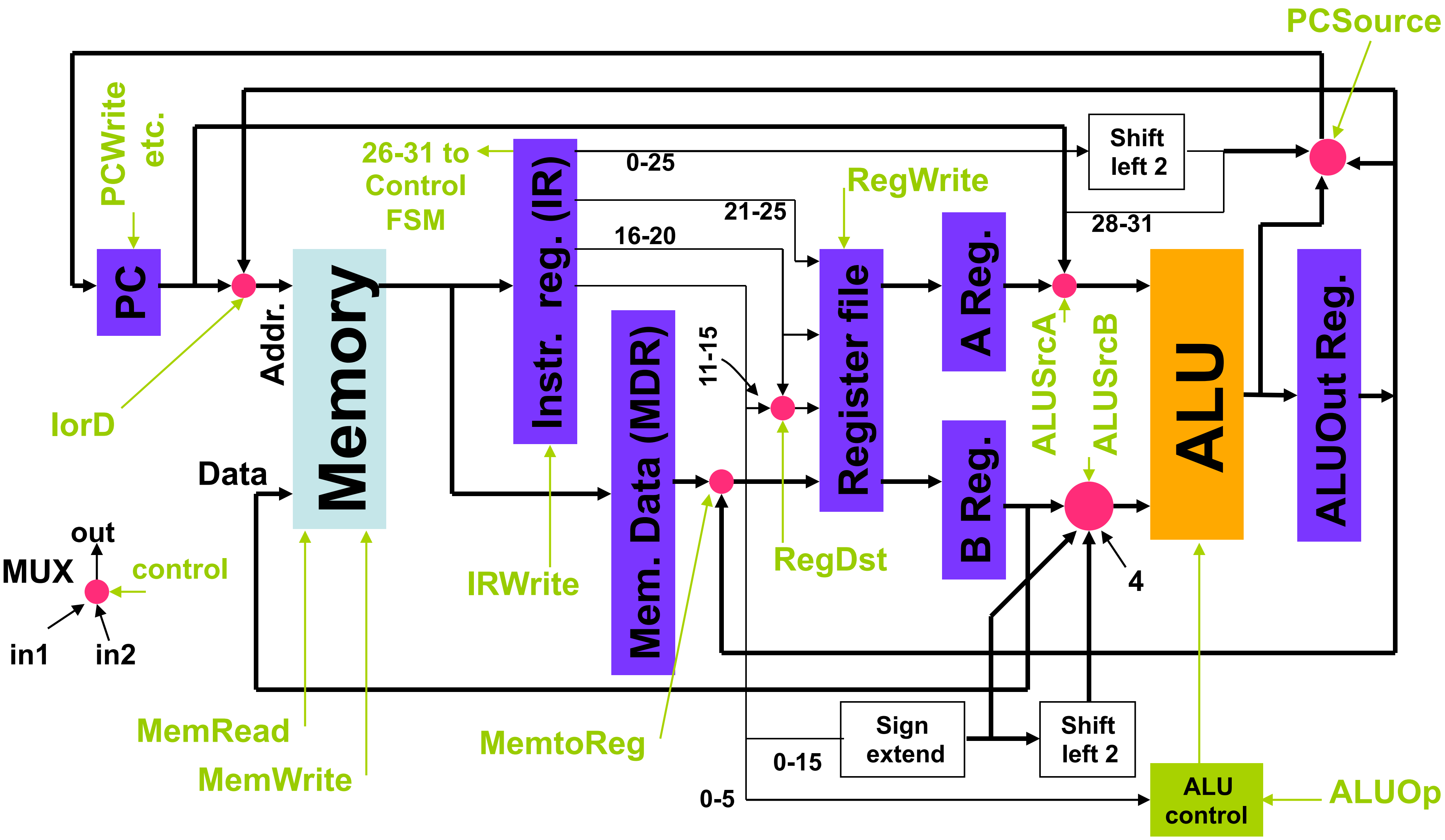
Lecture 9 (19 Feb 2013)

CADSL



Combined Datapaths

Multicycle Datapath



3 to 5 Cycles for an Instruction

Step	R-type (4 cycles)	Mem. Ref. (4 or 5 cycles)	Branch type (3 cycles)	J-type (3 cycles)
Instruction fetch	IR ← Memory[PC]; PC ← PC+4			

Instr. decode/ Reg. fetch	A ← Reg(IR[21-25]); B ← Reg(IR[16-20])
	ALUOut ← PC + (sign extend IR[0-15]) << 2			

Execution, addr. Comp., branch & jump completion	**ALUOut ← A op B**	**ALUOut ← A+sign extend (IR[0-15])**	**If (A= =B) then PC←ALUOut**	**PC←PC[28-31] || (IR[0-25]<<2)**
Mem. Access or R-type completion	**Reg(IR[11-15]) ← ALUOut**	**MDR←M[ALUout] or M[ALUOut]←B**		
Memory read completion		**Reg(IR[16-20]) ← MDR**		

ILP: Instruction Level Parallelism

- Single-cycle and multi-cycle datapaths execute one instruction at a time.
- How can we get better performance?
- Answer: Execute multiple instruction at a time:
  - Pipelining – Enhance a multi-cycle datapath to fetch one instruction every cycle.
  - Parallelism – Fetch multiple instructions every cycle.

Automobile Team Assembly

1 hour
1 hour
1 hour
1 hour

1 car assembled every four hours

6 cars per day

180 cars per month

2,040 cars per year

Automobile Assembly Line

Task 1
1 hour

Mecahnical

Task 2
1 hour

Electrical

Task 3
1 hour

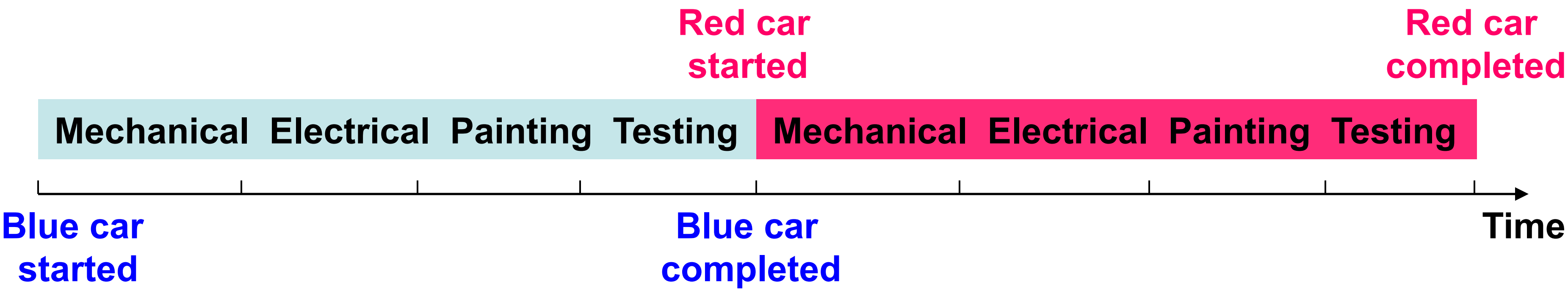
Painting

Task 4
1 hour

Testing

First car assembled in 4 hours (pipeline latency)
thereafter 1 car per hour
21 cars on first day, thereafter 24 cars per day
717 cars per month
8,637 cars per year

Throughput: Team Assembly

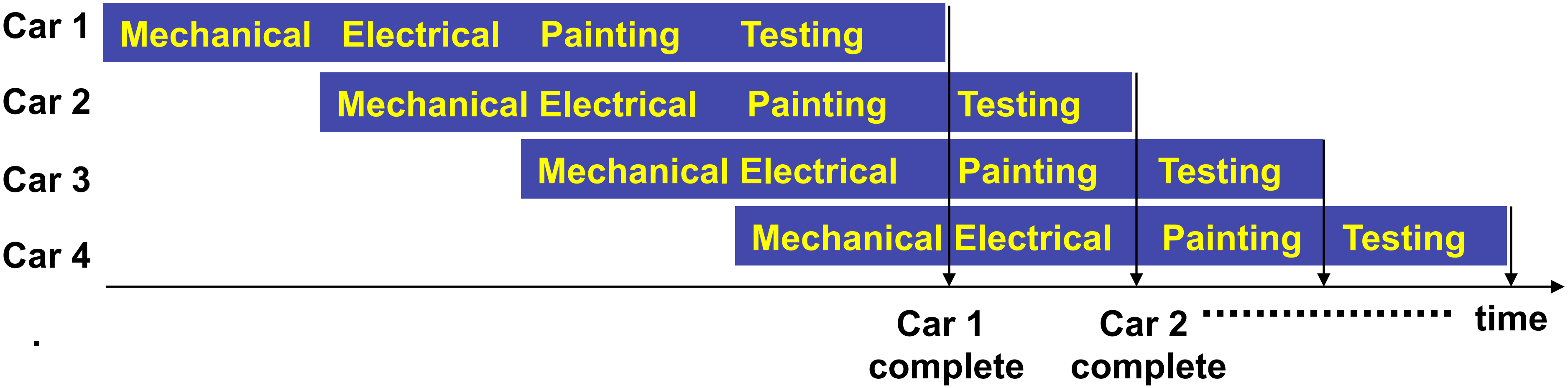

Red car started
Red car completed
Mechanical Electrical Painting Testing
Mechanical Electrical Painting Testing
Blue car started
Blue car completed
Time

Time of assembling one car = n hours

where n is the number of nearly equal subtasks, each requiring 1 unit of time

Throughput = $1/n$ cars per unit time

Throughput: Assembly Line



Car 1 Mechanical Electrical Painting Testing

Car 2 Mechanical Electrical Painting Testing

Car 3 Mechanical Electrical Painting Testing

Car 4 Mechanical Electrical Painting Testing

Car 1 complete

Car 2 complete ………………… time

Time to complete first car = n time units (latency)

Cars completed in time T = $T - n + 1$

Throughput = $1 - (n - 1)/T$ car per unit time

$$\frac{\text{Throughput (assembly line)}}{\text{Throughput (team assembly)}} = \frac{1-(n-1)/T}{1/n} = n - \frac{n(n-1)}{T} \to n \quad \text{as } T\to\infty$$

Some Features of Assembly Line

Electrical parts delivered (JIT)

Task 1
1 hour

Task 2
1 hour

Task 3
1 hour

Task 4
1 hour

Mechanical

Electrical

Painting

Testing

Defect found

3 cars in the assembly line are suspects, to be removed (flush pipeline)

Stall assembly line to fix the cause of defect

Pipelining in a Computer

- Divide datapath into nearly equal tasks, to be performed serially and requiring non-overlapping resources.
- Insert registers at task boundaries in the datapath; registers pass the output data from one task as input data to the next task.
- Synchronize tasks with a clock having a cycle time that just exceeds the time required by the longest task.
- Break each instruction down into a fixed number of tasks so that instructions can be executed in a staggered fashion.

Thank You