

# CSE 112: Computer Organization

---

Instructor: Sujay Deb

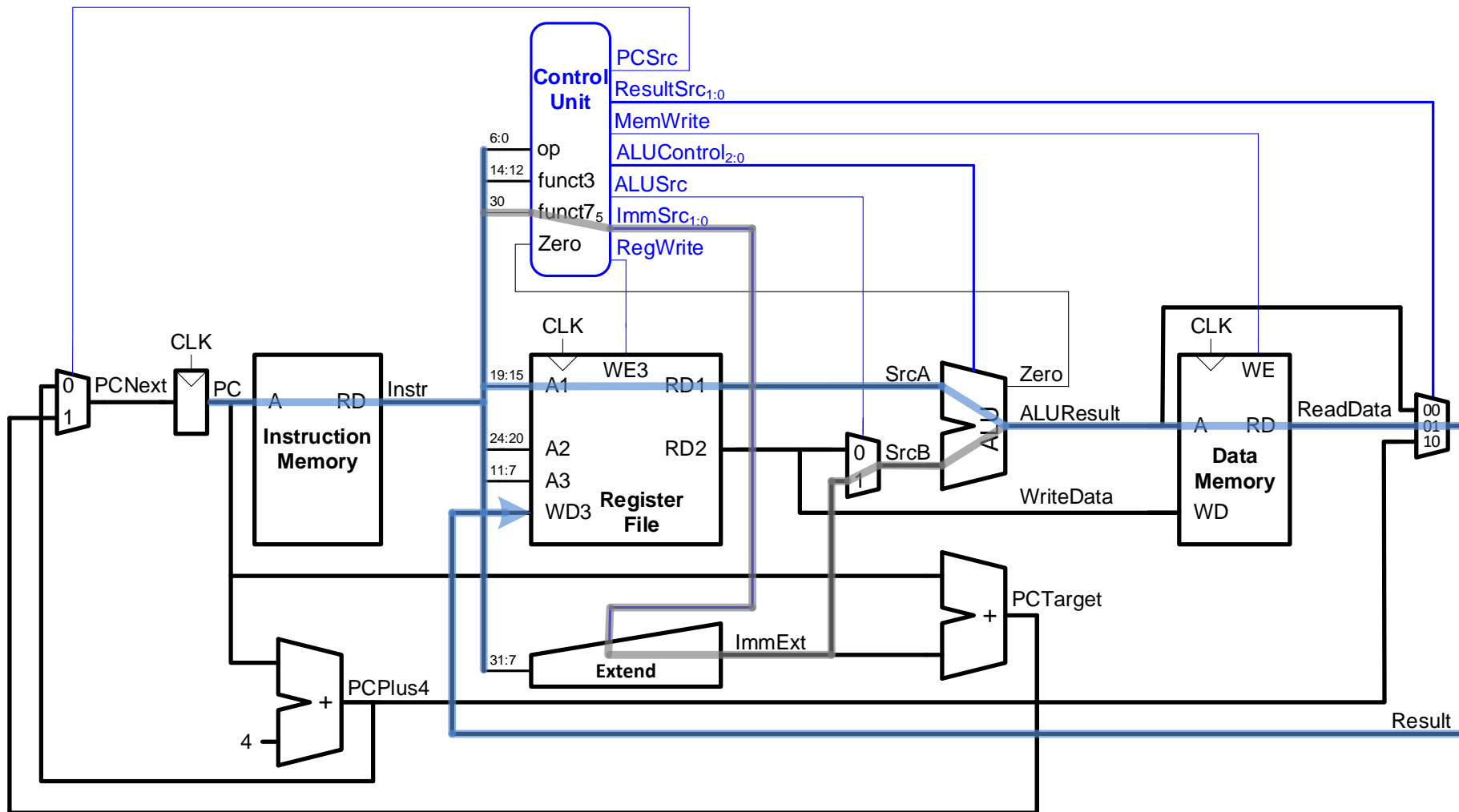
## Lecture 14 & 15



INDRAPRASTHA INSTITUTE *of*  
INFORMATION TECHNOLOGY  
DELHI



# Single-Cycle Processor Performance



$T_C$  limited by critical path (**1w**)

# Single-Cycle Processor Performance

- Single-cycle critical path:

$$T_{c\_single} = t_{pcq\_PC} + t_{mem} + \max[t_{RFread}, t_{dec} + t_{ext} + t_{mux}] + t_{ALU} + t_{mem} + t_{mux} + t_{RFsetup}$$

- Typically, limiting paths are:

- memory, ALU, register file

- So, 
$$T_{c\_single} = t_{pcq\_PC} + t_{mem} + t_{RFread} + t_{ALU} + t_{mem} + t_{mux} + t_{RFsetup}$$
$$= t_{pcq\_PC} + 2t_{mem} + t_{RFread} + t_{ALU} + t_{mux} + t_{RFsetup}$$

# Single-Cycle Performance Example

Element	Parameter	Delay (ps)
Register clock-to-Q	$t_{pcq\_PC}$	40
Register setup	$t_{setup}$	50
Multiplexer	$t_{mux}$	30
AND-OR gate	$t_{AND-OR}$	20
ALU	$t_{ALU}$	120
Decoder (Control Unit)	$t_{dec}$	25
Extend unit	$t_{ext}$	35
Memory read	$t_{mem}$	200
Register file read	$t_{RFread}$	100
Register file setup	$t_{RFsetup}$	60

$$T_{c\_single} = t_{pcq\_PC} + 2t_{mem} + t_{RFread} + t_{ALU} + t_{mux} + t_{RFsetup}$$
$$=$$

# Single-Cycle Performance Example

Program with 100 billion instructions:

$$\text{Execution Time} = \# \text{ instructions} \times \text{CPI} \times T_c$$

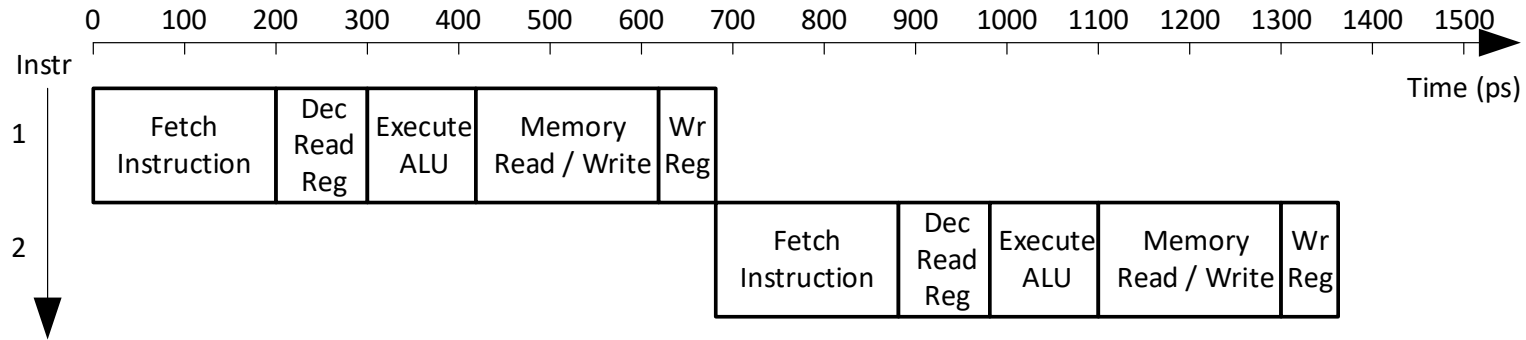
# **Pipelined RISC-V Processor**

# Pipelined RISC-V Processor

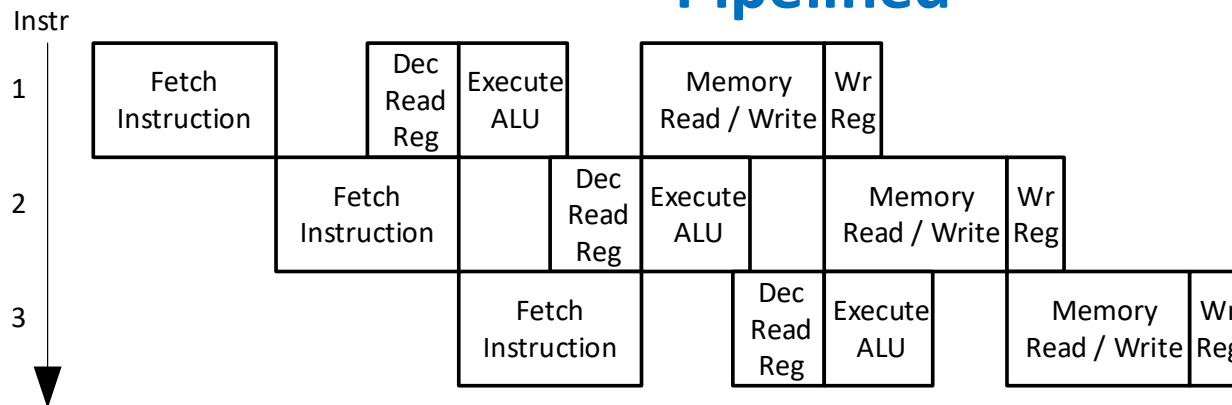
- Temporal parallelism
- Divide single-cycle processor into **5 stages**:
  - Fetch
  - Decode
  - Execute
  - Memory
  - Writeback
- Add **pipeline registers** between stages

# Single-Cycle vs. Pipelined Processor

## Single-Cycle



## Pipelined



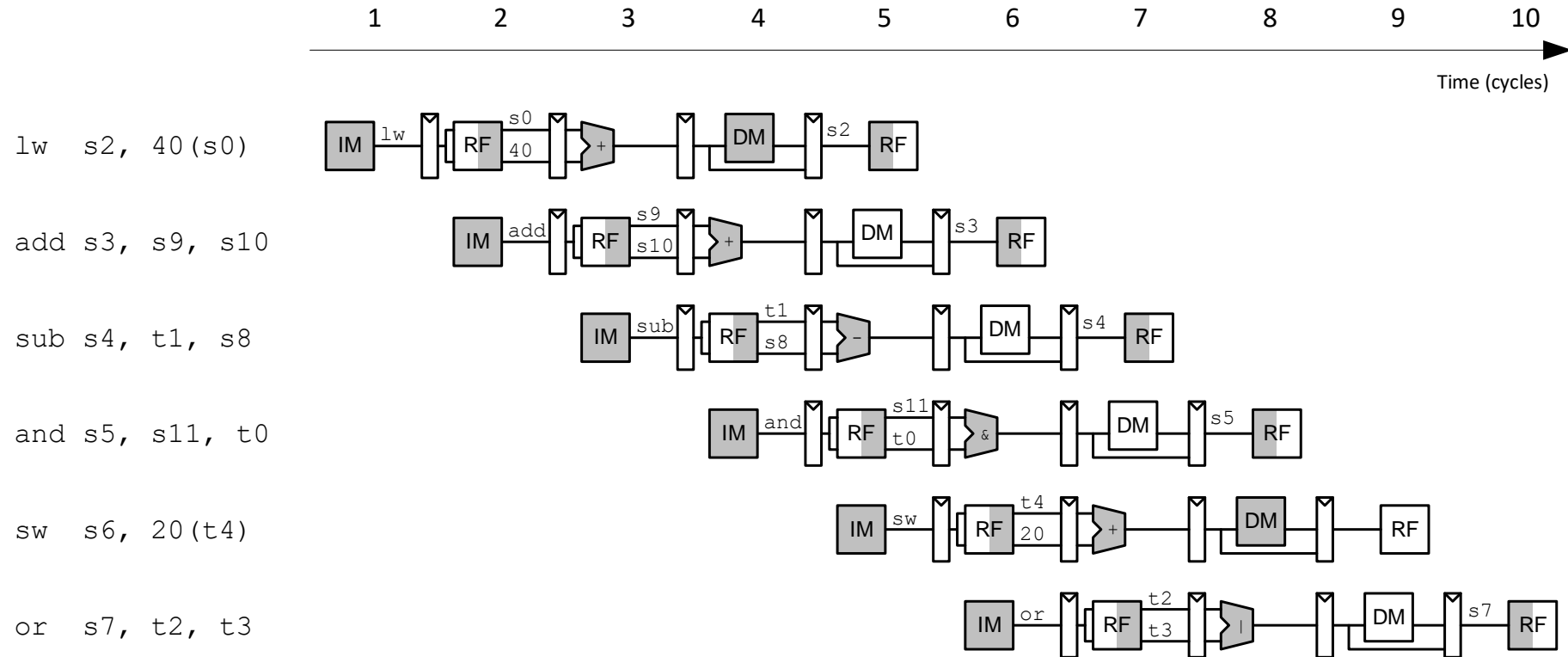


## Class Interaction # 16

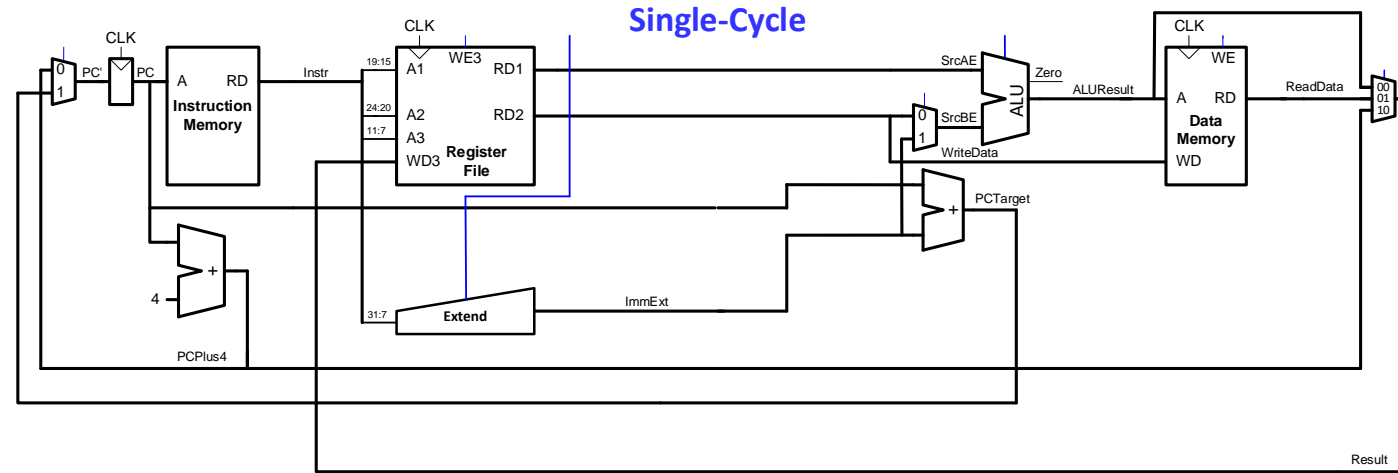
---



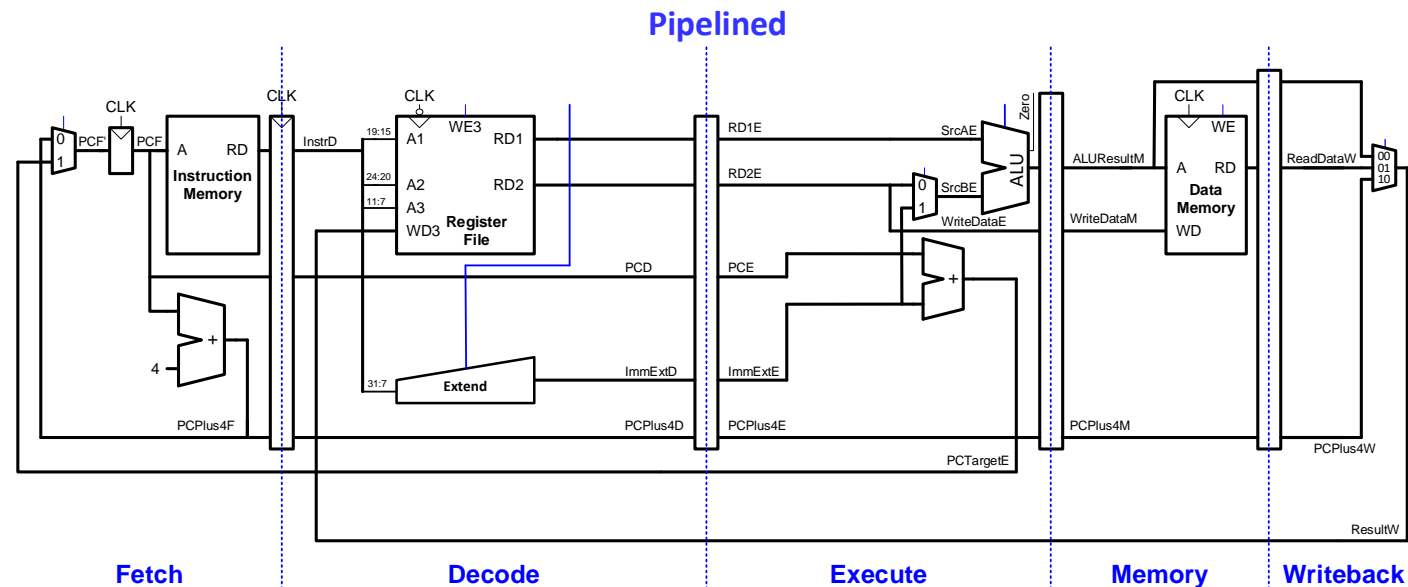
# Pipelined Processor Abstraction



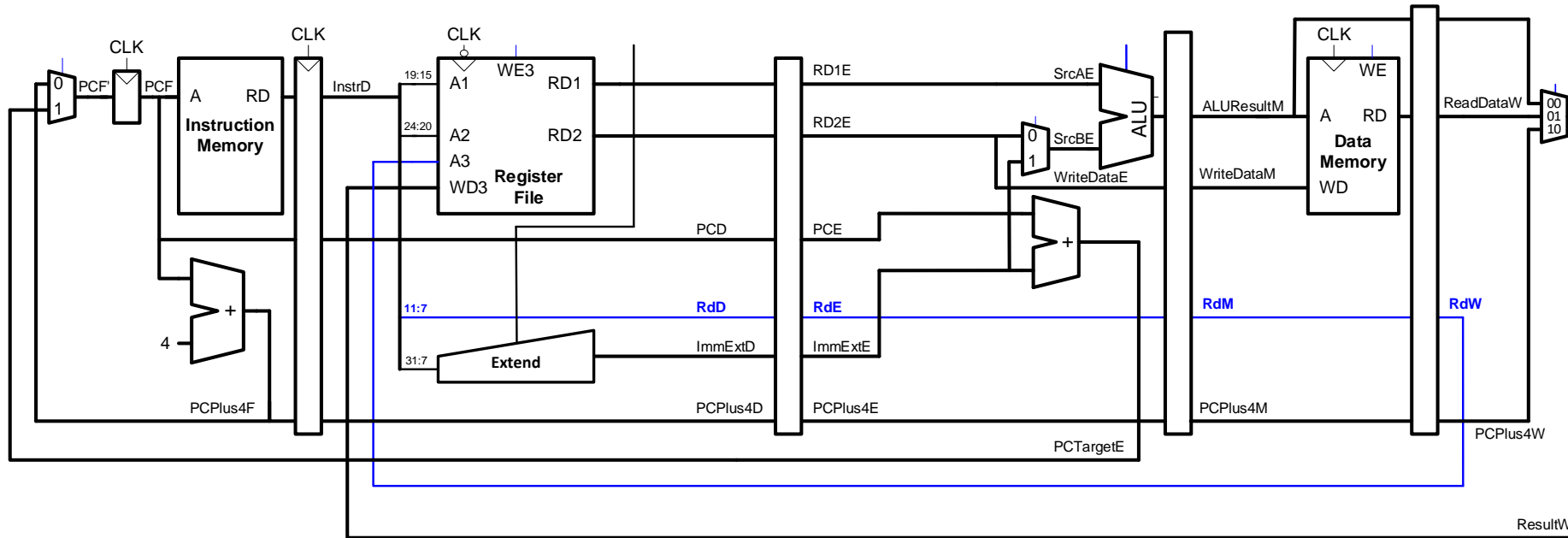
# Single-Cycle & Pipelined Datapaths



Signals in Pipelined Processor are appended with first letter of stage (i.e., PC<sub>F</sub>, PC<sub>D</sub>, PC<sub>E</sub>).

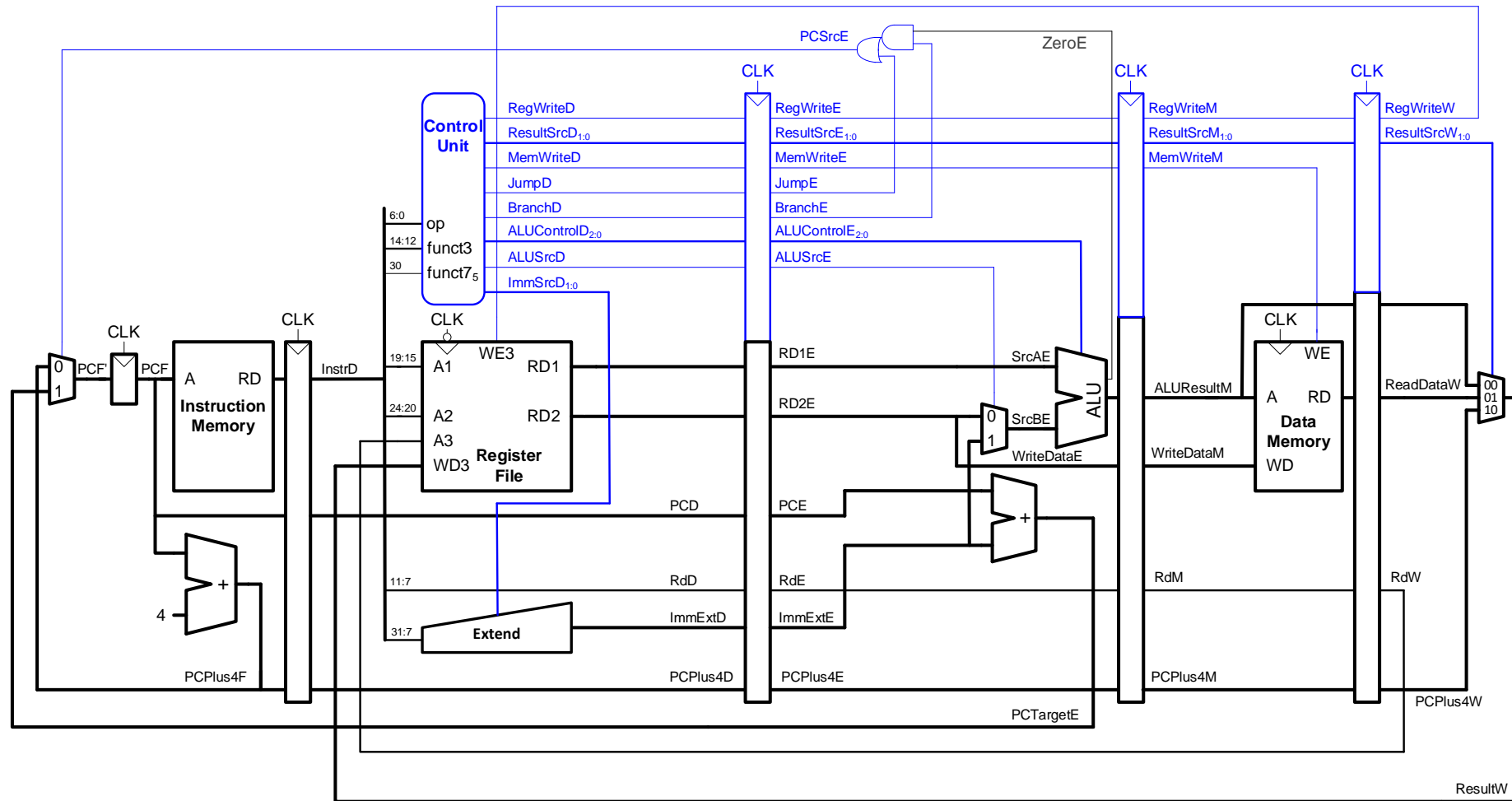


# Corrected Pipelined Datapath



- *Rd* must arrive at same time as *Result*
- Register file written on **falling edge** of *CLK*

# Pipelined Processor with Control



- **Same control unit** as single-cycle processor
- **Control signals travel with** the instruction (drop off when used)

# **Pipelined Processor Hazards**

# Pipelined Hazards

- When an instruction depends on result from instruction that hasn't completed
- Types:
  - **Data hazard:** register value not yet written back to register file
  - **Control hazard:** next instruction not decided yet (caused by branch)

# Data Hazard

