

○○○○

MERL
Accelerating Engineering Innovation

MERL
DSU

RISC-V
Pipeline Core

TABLE OF CONTENTS

- Overview
- Implementation of Fetch Cycle
- Implementation of Decode Cycle
- Implementation of Execute Cycle
- Implementation of Memory Cycle
- Implementation of Write Back Cycle
- Implementation of Pipeline Top
- Pipeline Hazards
- Implementation of Hazard Unit
- Implementation of Pipeline Top II

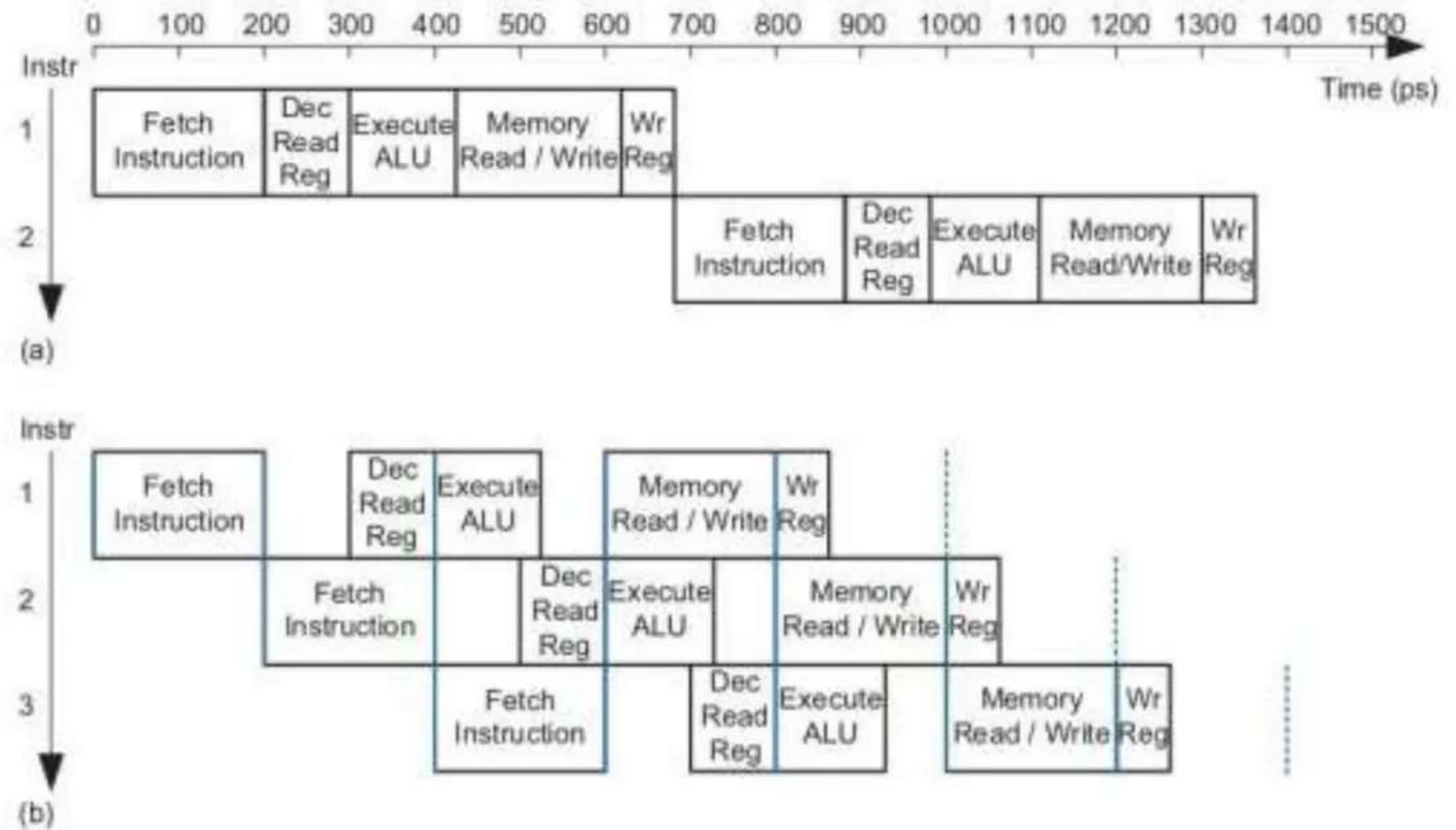


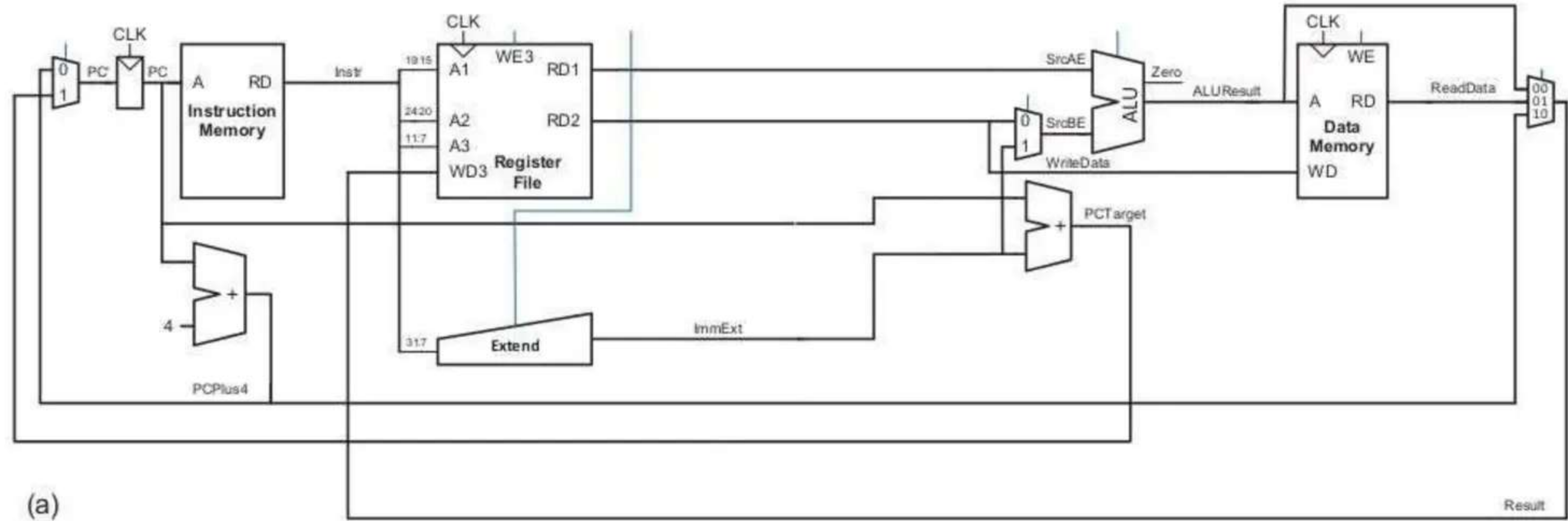
Overview of RISC-V Pipeline Architecture



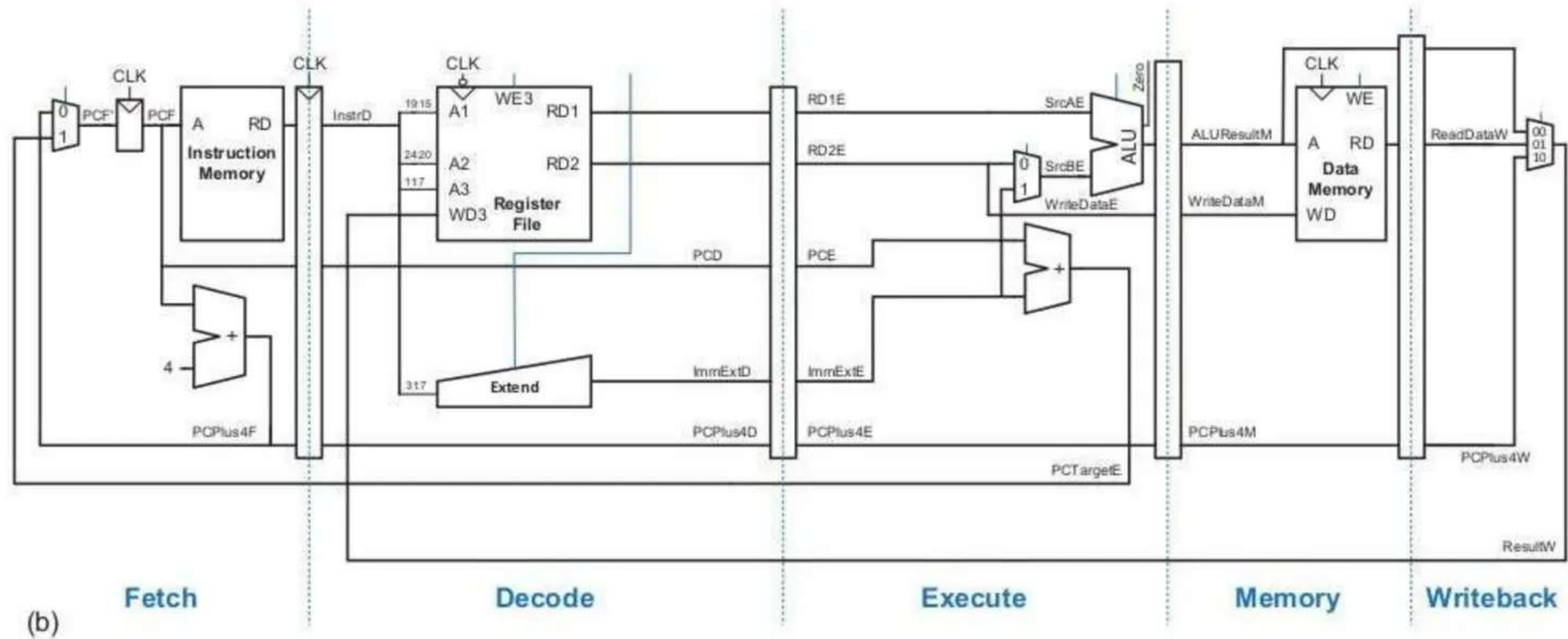
Pipelining

We design a pipelined processor by subdividing the single-cycle processor into five pipeline stages. Thus, five instructions can execute simultaneously, one in each stage. Because each stage has only one-fifth of the entire logic, the clock frequency is approximately five times faster.



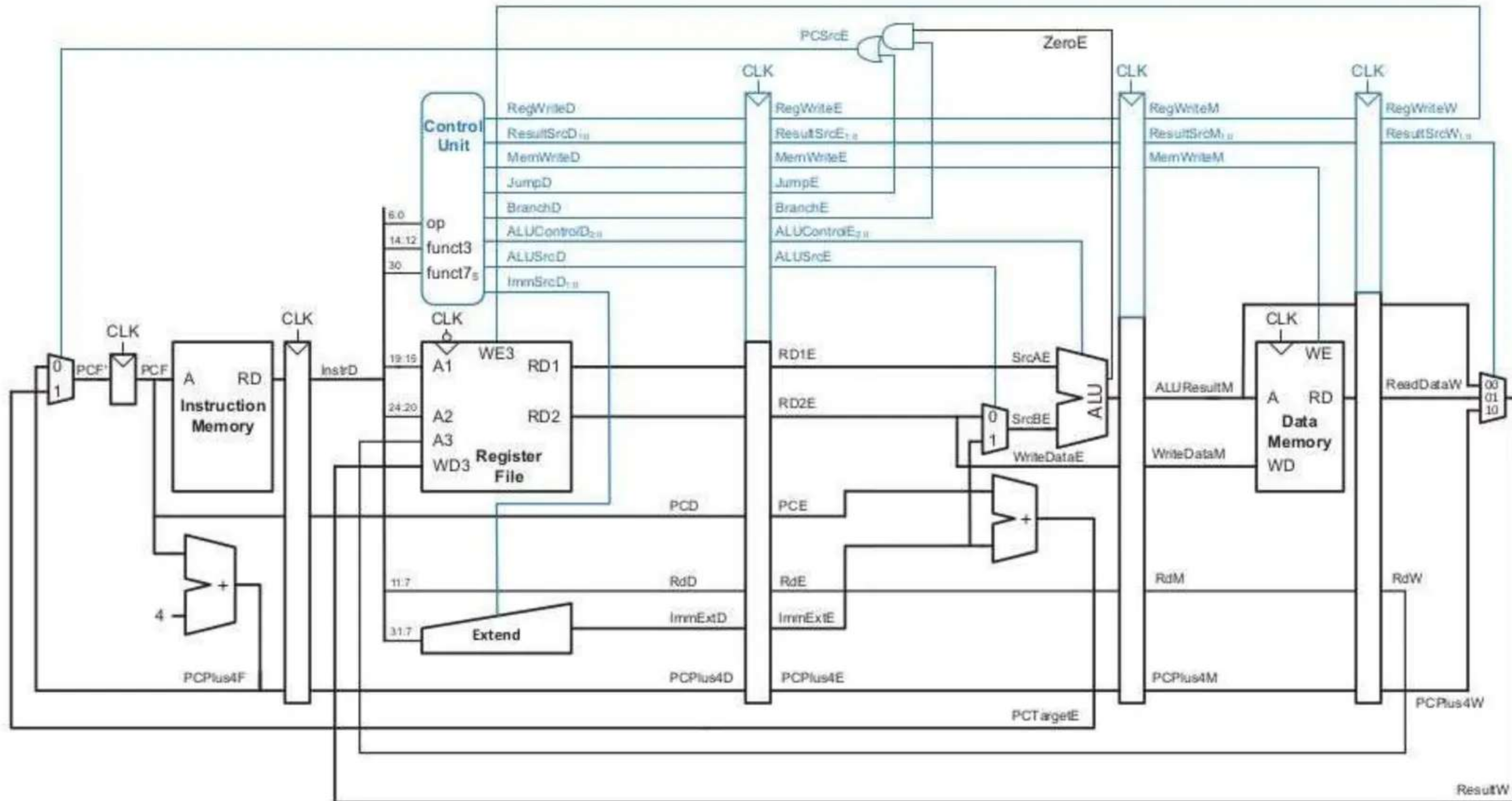


(a)



(b)

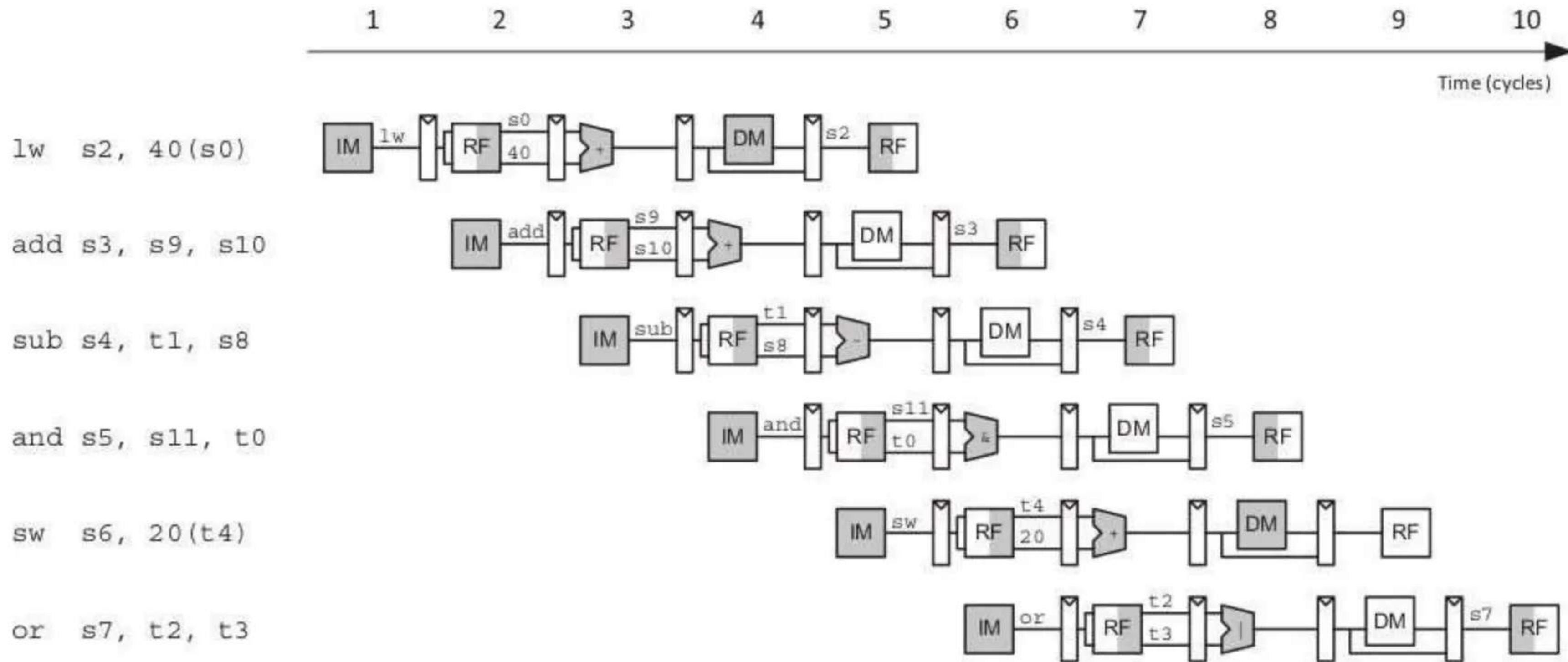
Pipeline Datapath



Implementation of Fetch Cycle



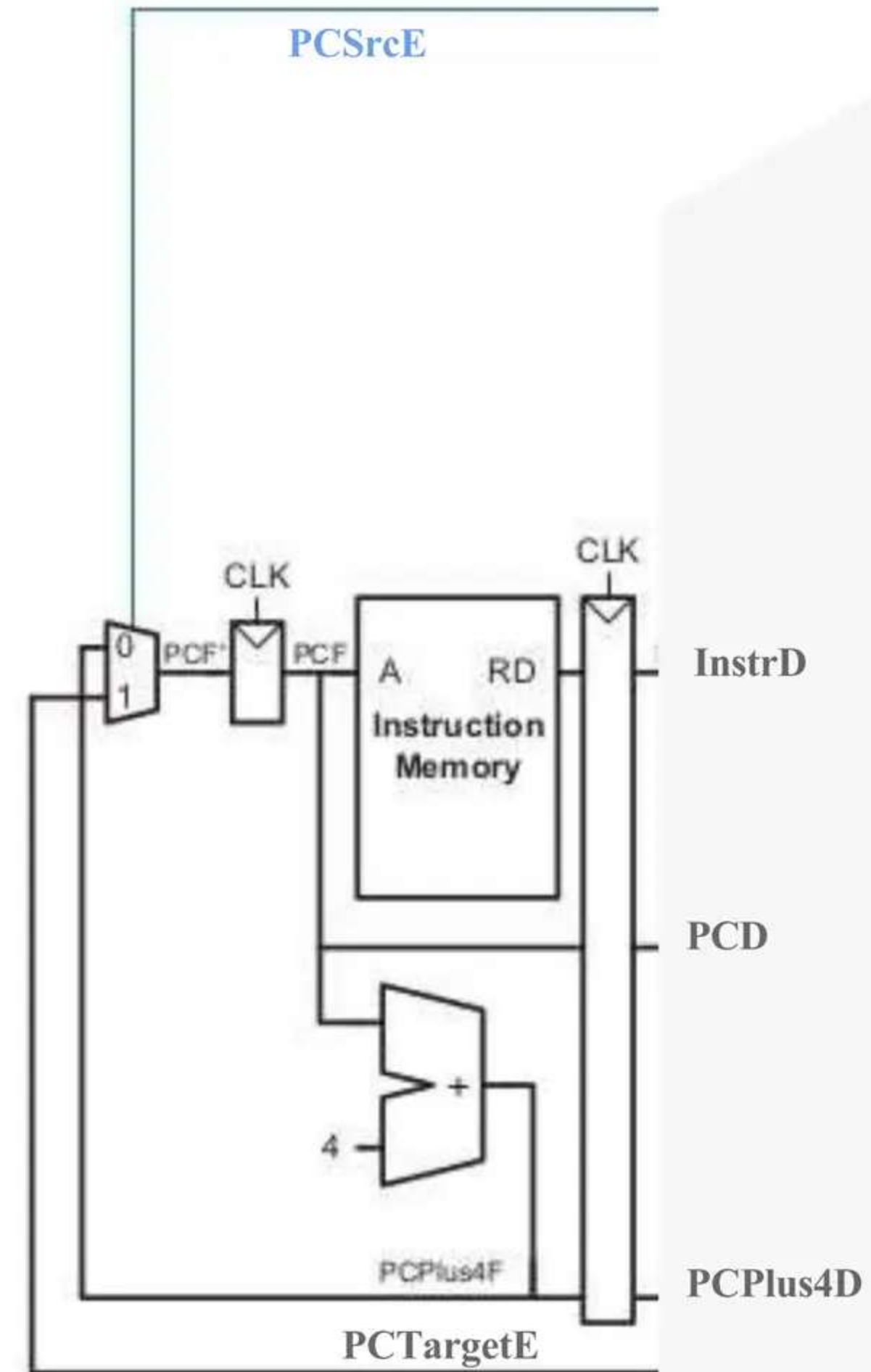
Abstract View of Pipelining



Fetch Cycle Datapath

Modules to be Integrated:

- 1) PC Mux
- 2) Program Counter
- 3) Adder
- 4) Instruction Memory
- 5) Fetch Stage Registers



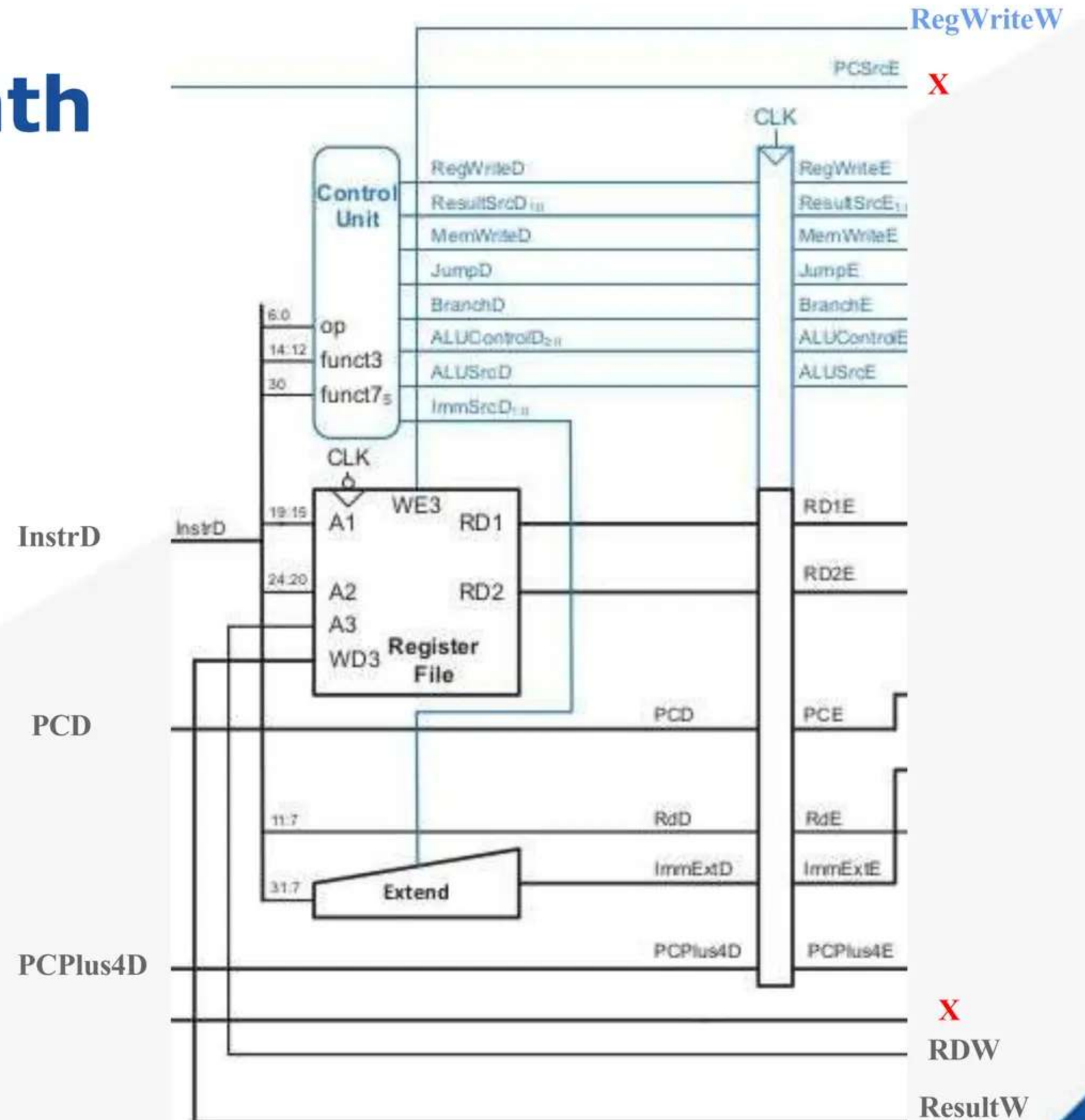
Implementation of Decode Cycle



Decode Cycle Datapath

Modules to be Integrated:

- 1) Control Unit
- 2) Register File
- 3) Extender
- 4) Decode Stage Registers



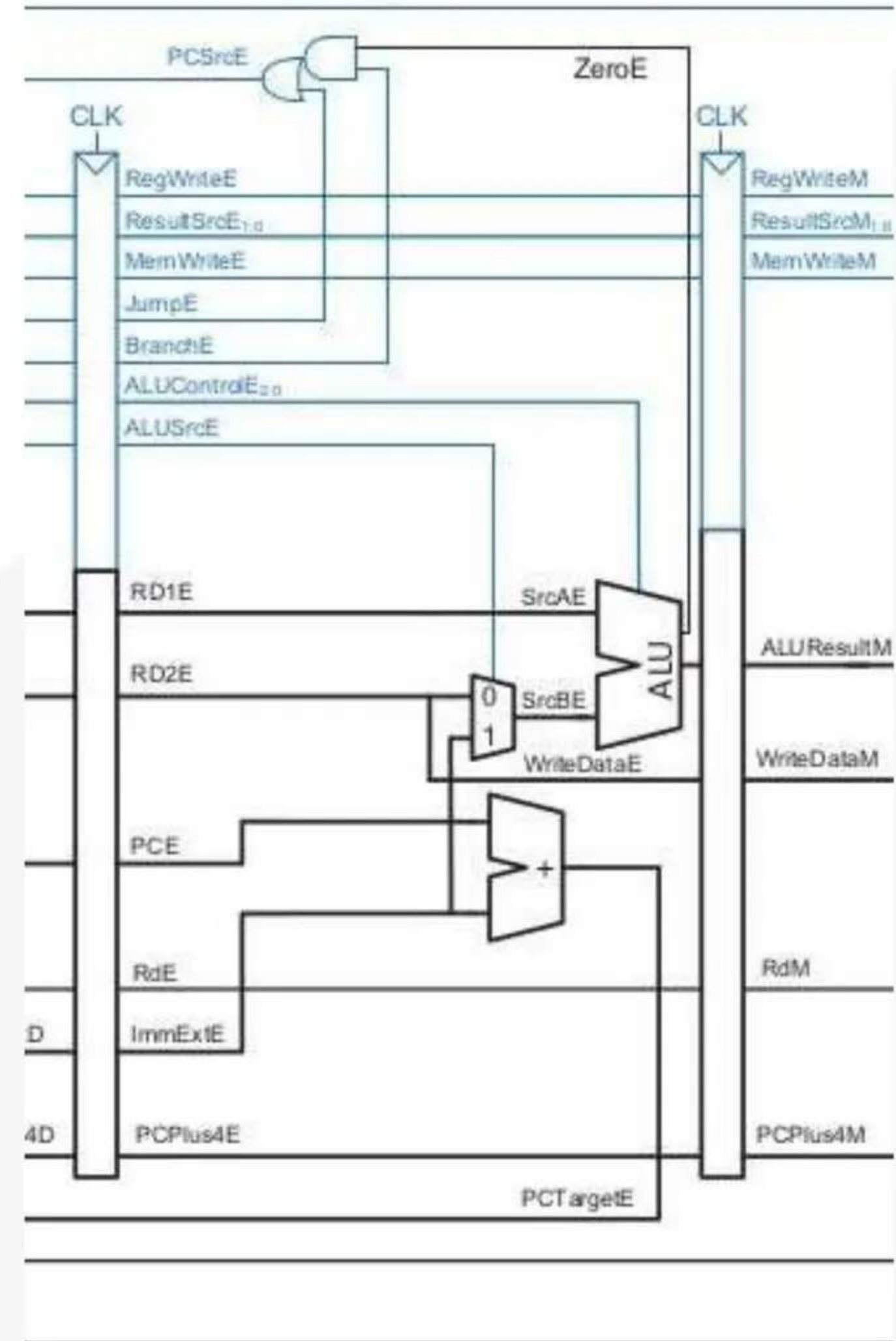
Implementation of Execute Cycle



Execute Cycle Datapath

Modules to be Integrated:

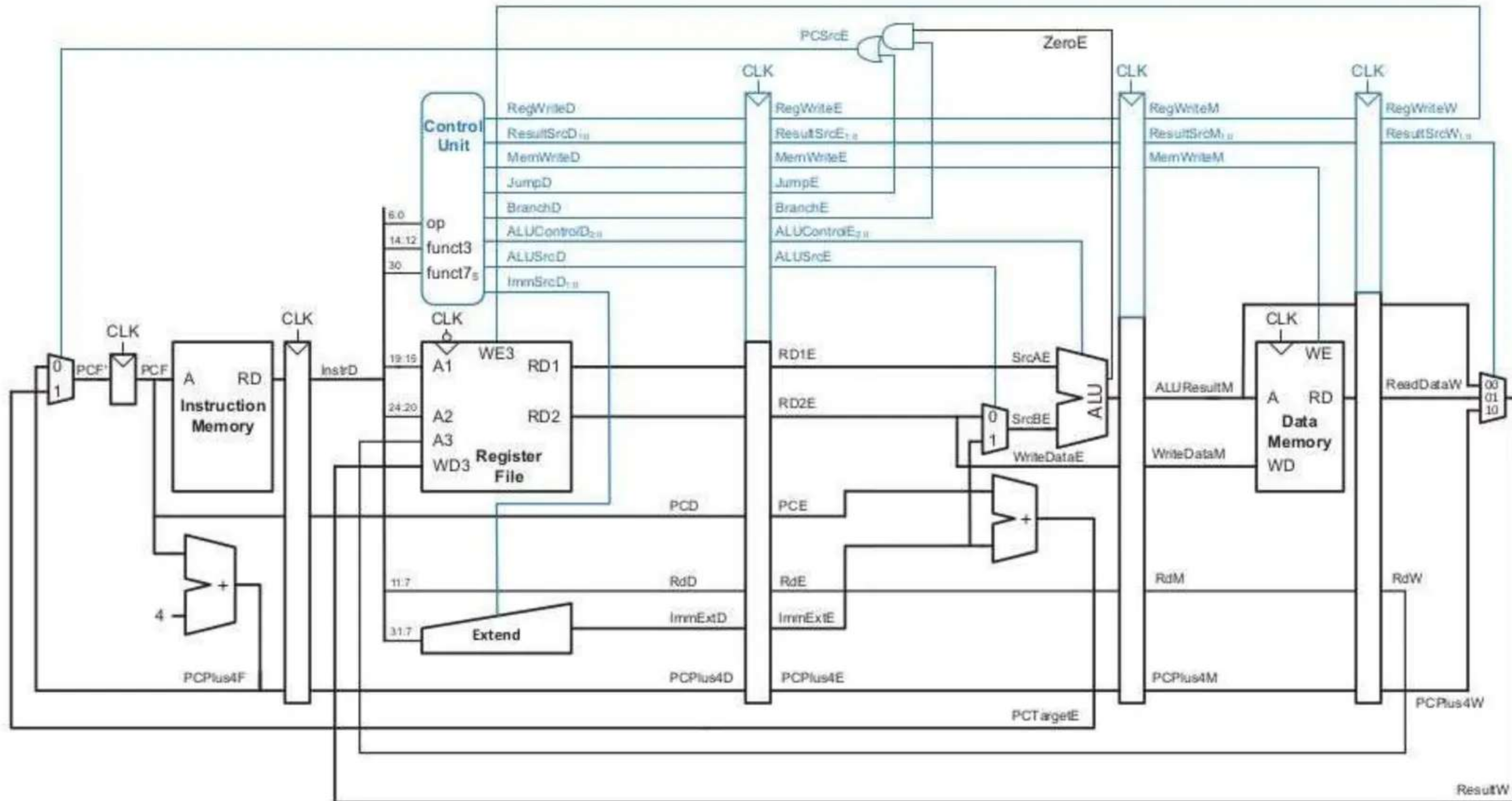
- 1) AND Gate
- 2) Mux
- 3) Adder
- 4) ALU
- 5) Execute Stage Registers



Implementation of Memory Cycle



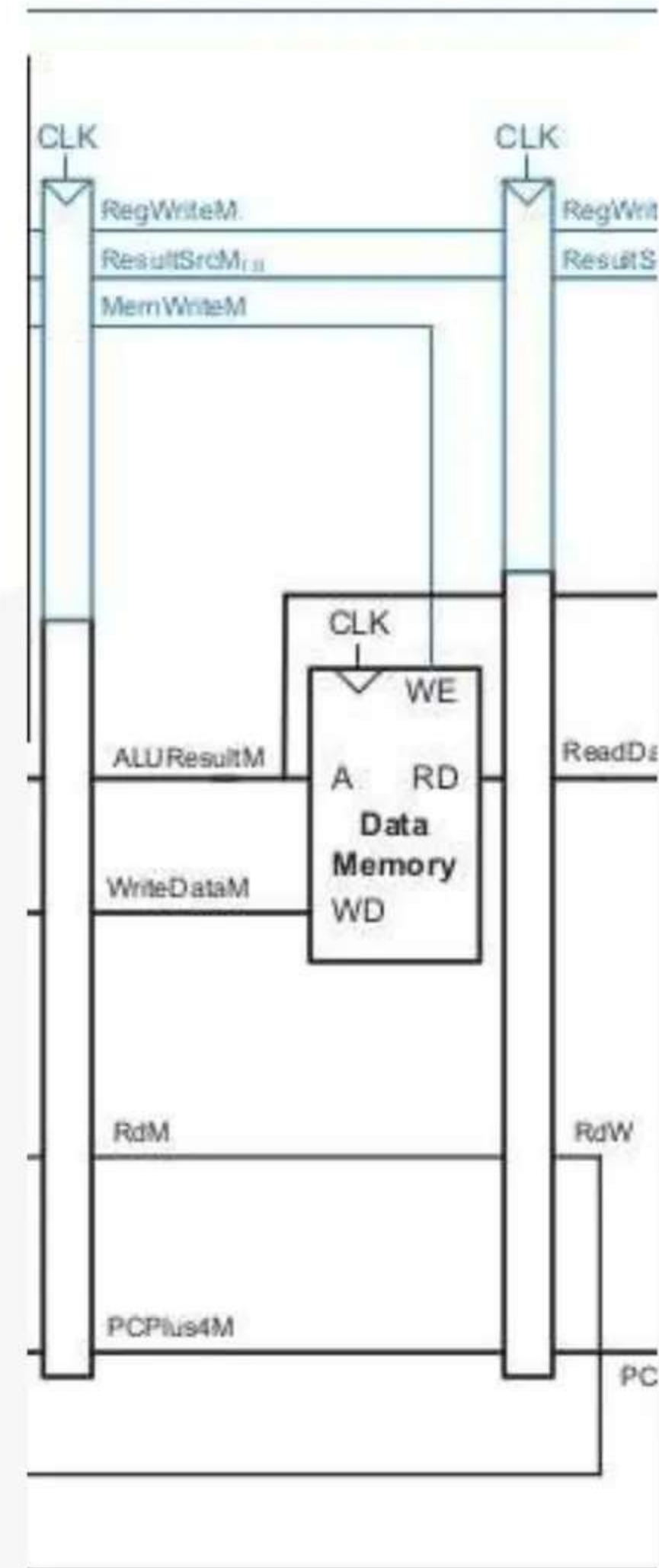
Pipeline Datapath



Memory Cycle Datapath

Modules to be Integrated:

- 1) Data Memory
- 2) Memory Stage Registers



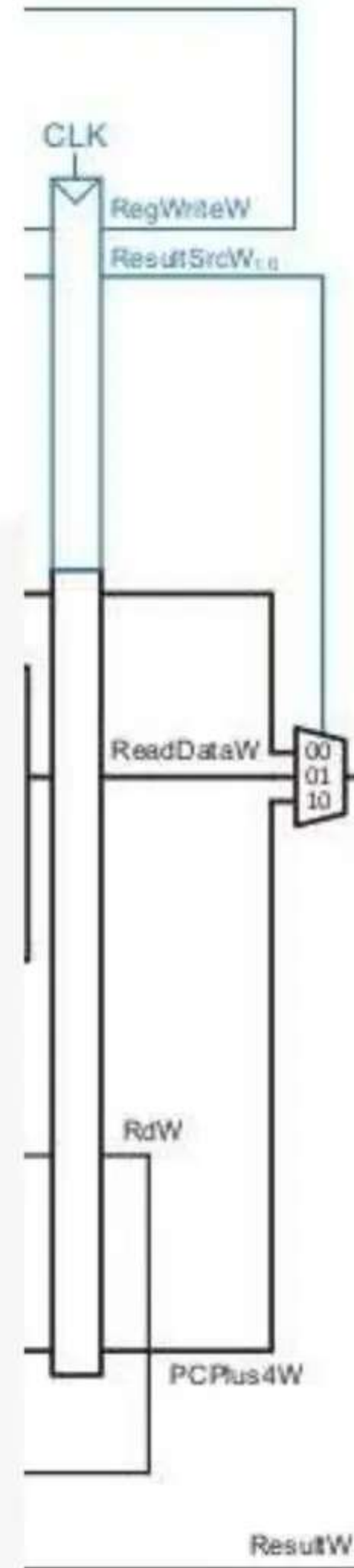
Implementation of Write Back Cycle



Write Back Cycle Datapath

Modules to be Integrated:

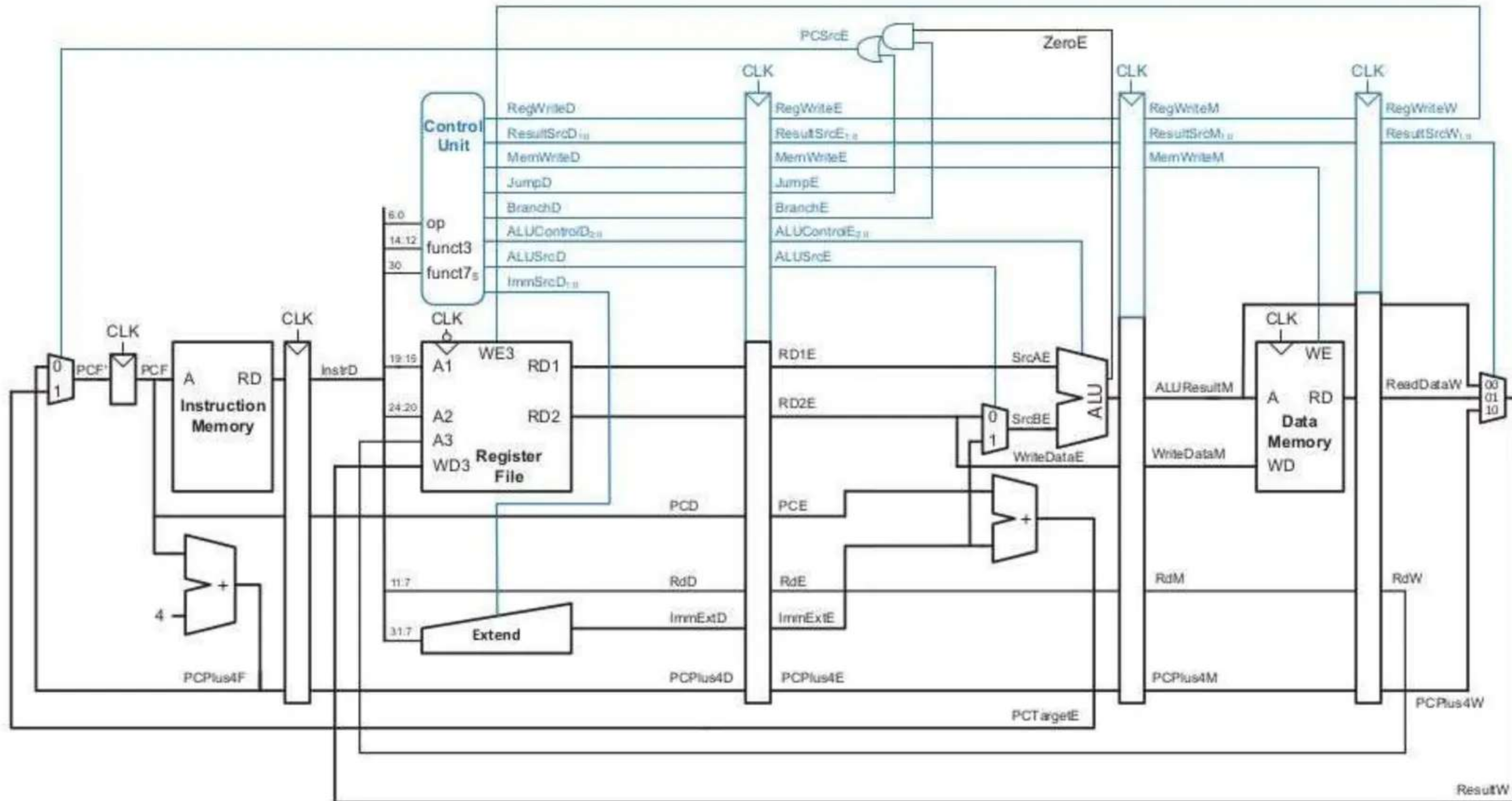
- 1) Mux



Implementation of Pipeline Top



Pipeline Datapath



Pipeline Hazards



Pipeline Hazard

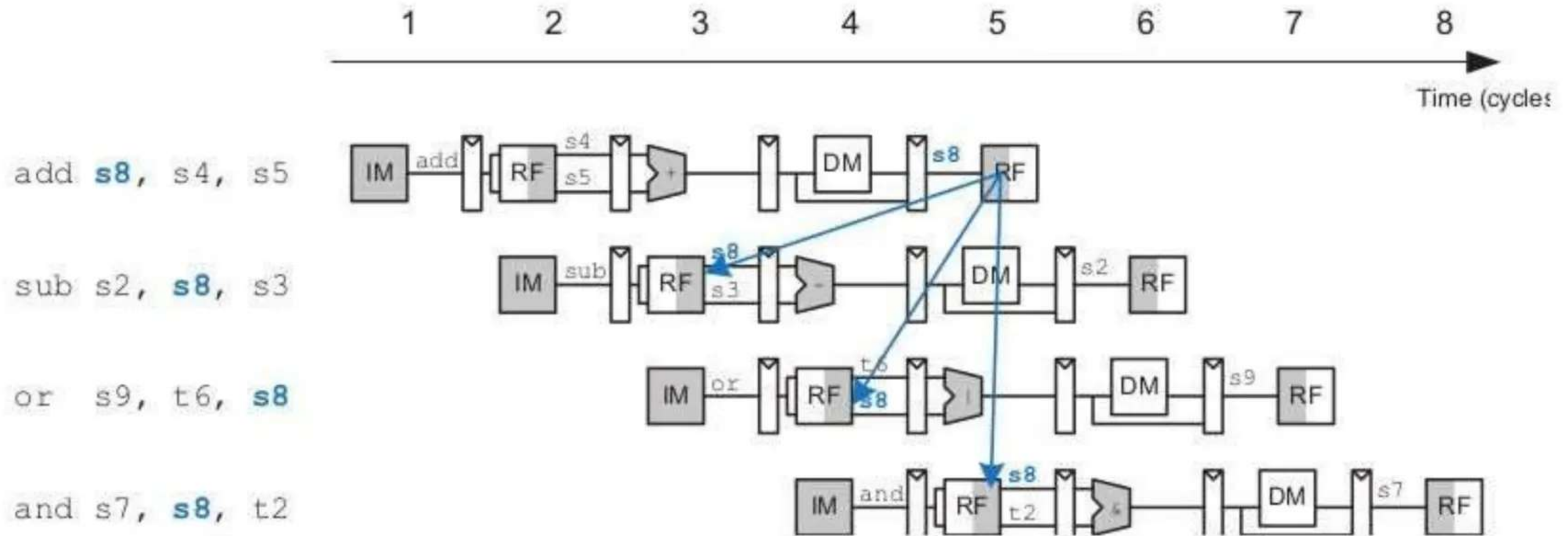
- ***Structural Hazard***

1. Hardware does not support the execution of instruction in same clock cycle.
2. Without having Two memories RISC-V pipelining architecture will have structural hazard.

- ***Data Hazard***

1. Data to be executed is not available.
2. May occur when pipeline is stalled.
3. Solve by using **forwarding** or **bypassing** technique.

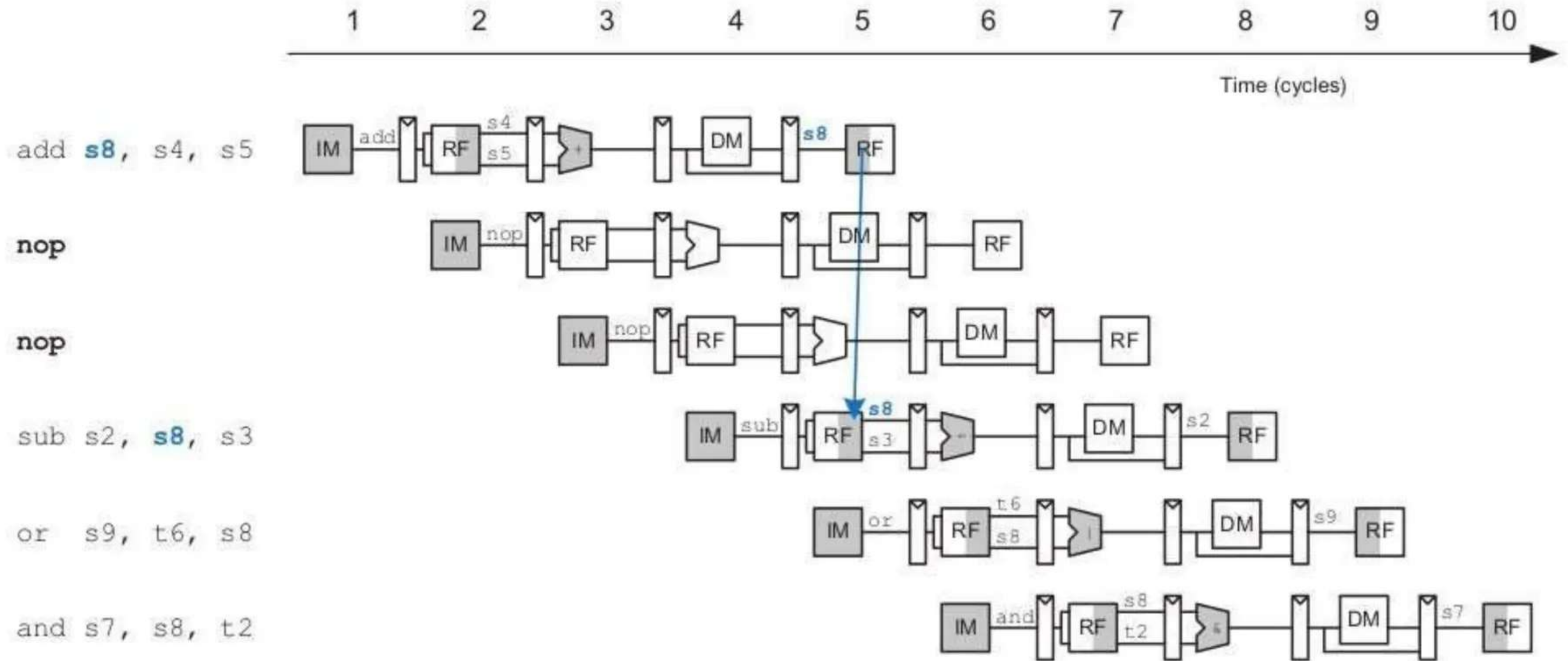
Data Hazard In Pipelining



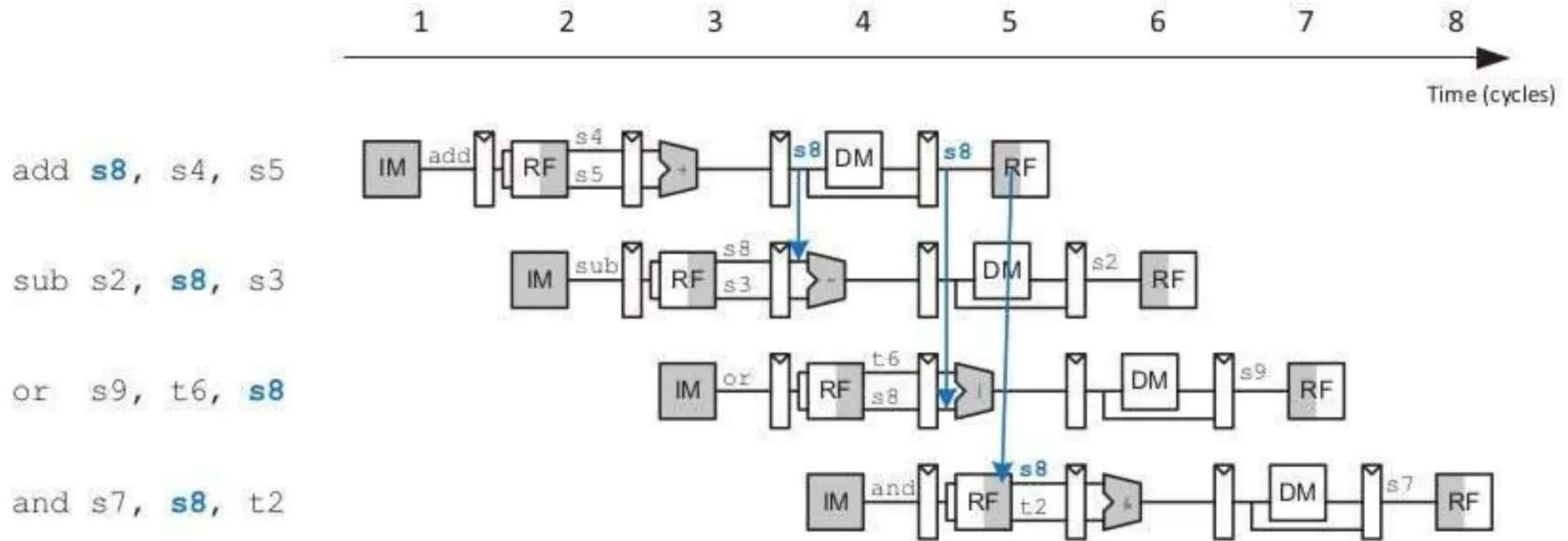
Solution of Data Hazards

- *Solving Data Hazards with nops*
- *Solving Data Hazard with Forwarding / Bypassing*

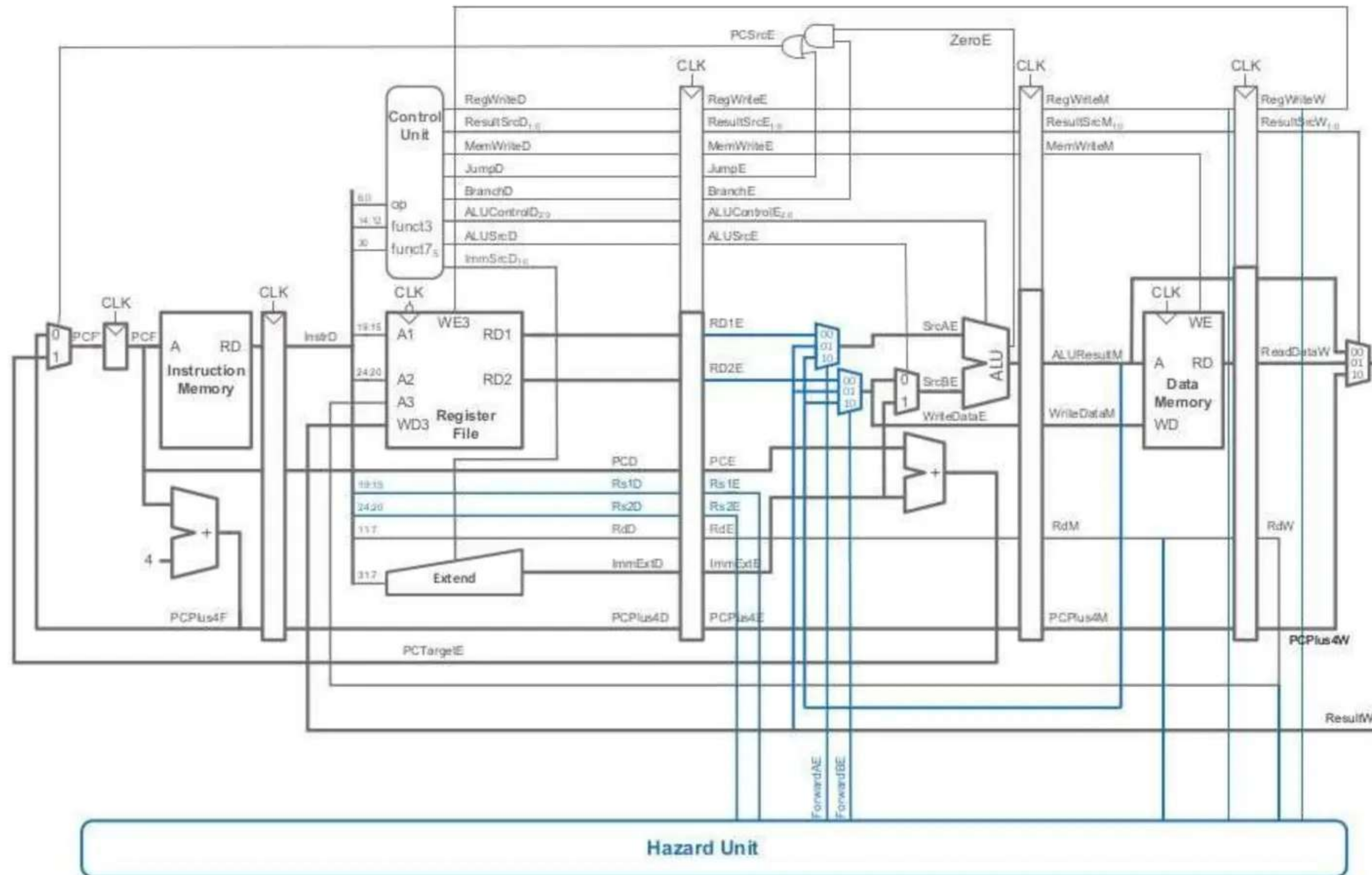
Using Nops



Using Forwarding / Bypassing



Updated Pipeline Top Architecture



Implementation of Hazard Unit



Condition Table

Mux control	Source	Explanation
ForwardA = 00	ID/EX	The first ALU operand comes from the register file.
ForwardA = 10	EX/MEM	The first ALU operand is forwarded from the prior ALU result.
ForwardA = 01	MEM/WB	The first ALU operand is forwarded from data memory or an earlier ALU result.
ForwardB = 00	ID/EX	The second ALU operand comes from the register file.
ForwardB = 10	EX/MEM	The second ALU operand is forwarded from the prior ALU result.
ForwardB = 01	MEM/WB	The second ALU operand is forwarded from data memory or an earlier ALU result.

Condition for Data Hazard

Memory Stage

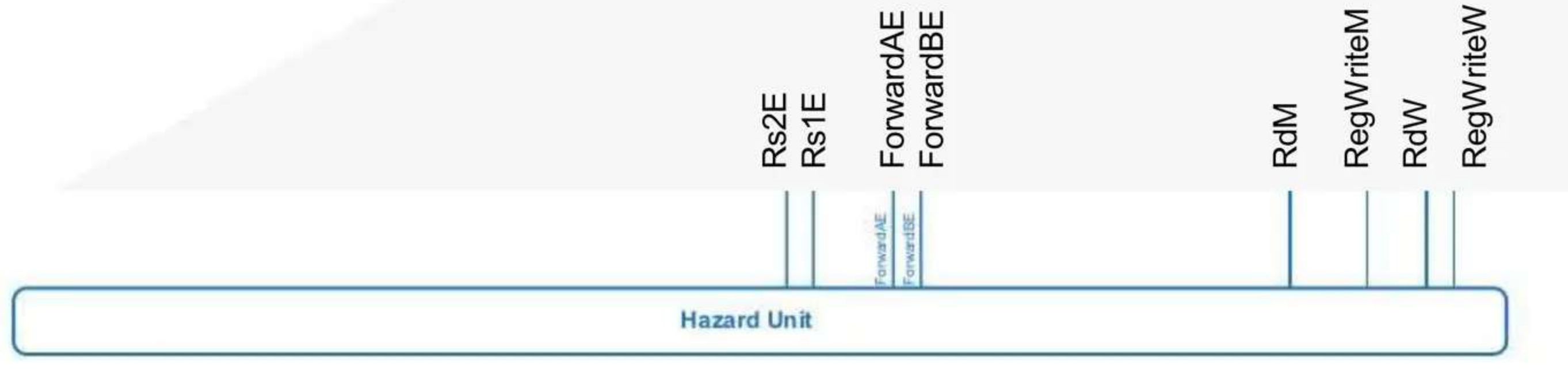
if (RegWriteM and (RdM != 0) and (RdM == Rs1E))
ForwardAE = 10

if (RegWriteM and (RdM != 0) and (RdM == Rs2E))
ForwardBE = 10

WriteBack Stage

if (RegWriteW and (RdW != 0) and (RdW == Rs1E))
ForwardAE = 01

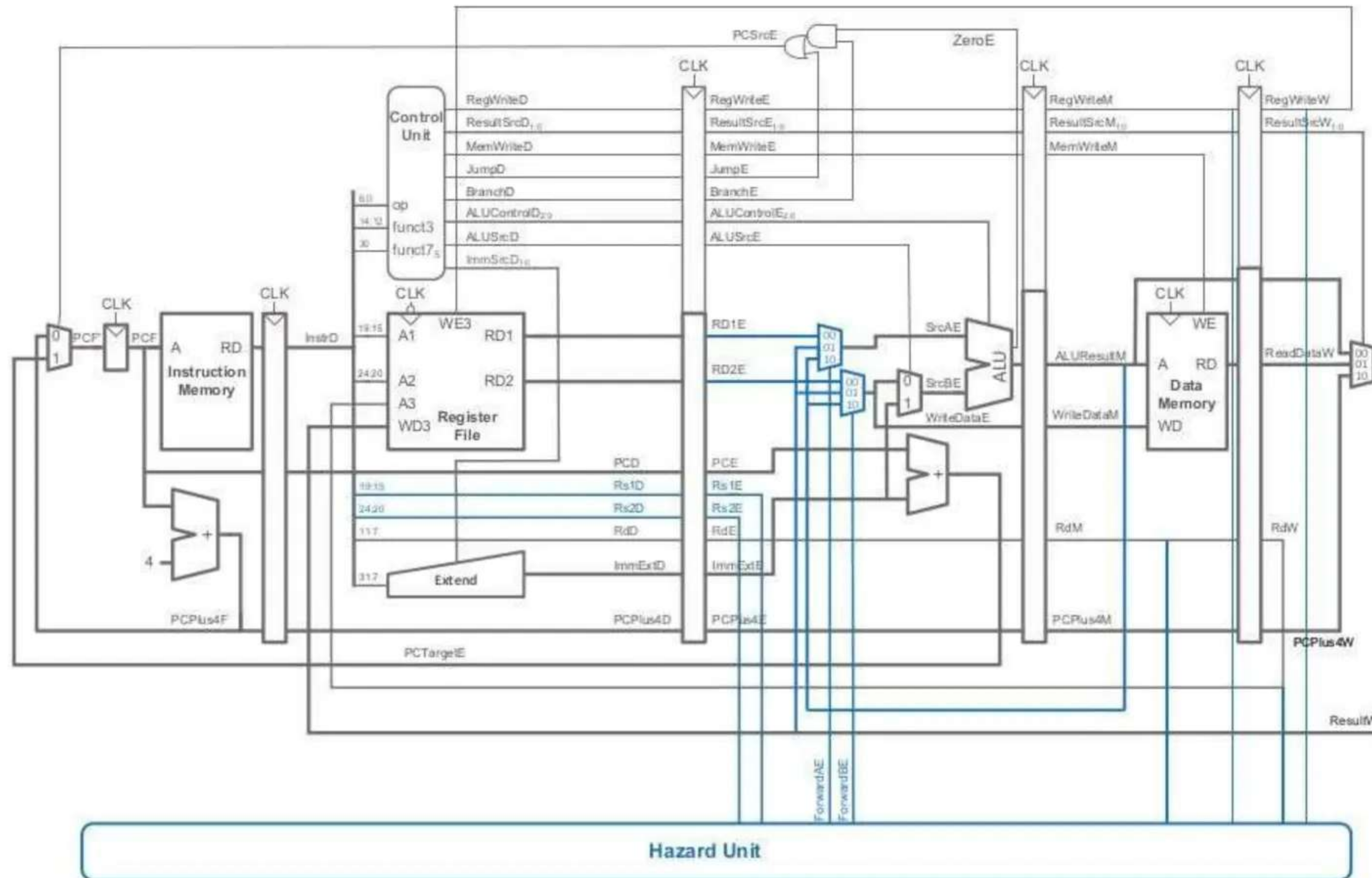
if (RegWriteW and (RdW != 0) and (RdW == Rs2E))
ForwardBE = 01



Implementation of Pipeline Top II



Updated Pipeline Top Architecture



Thnnfi You

