



DIGITAL DESIGN AND COMPUTER ORGANIZATION

Multi-Cycle Processor - 4

Reetinder Sidhu

Department of Computer Science and Engineering

DIGITAL DESIGN AND COMPUTER ORGANIZATION

Multi-Cycle Processor - 4

Reetinder Sidhu

Department of Computer Science and
Engineering

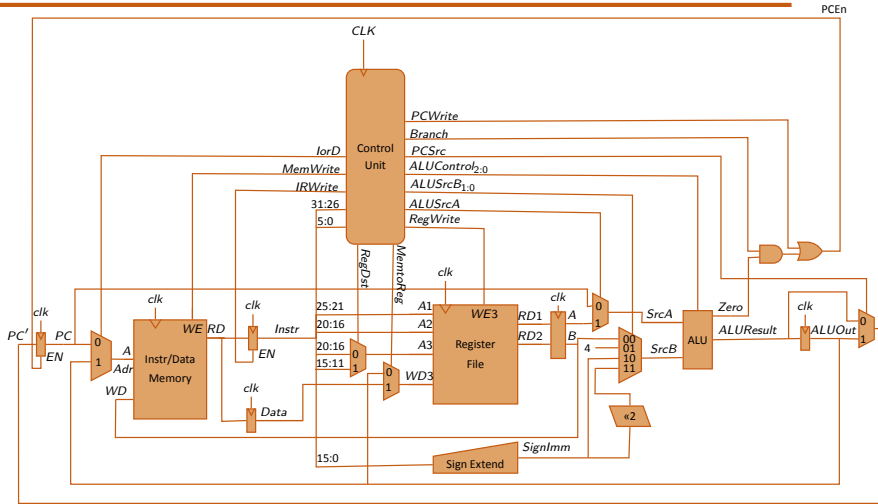
- Digital Design
 - ▶ Combinational logic design
 - ▶ Sequential logic design
- Computer Organization
 - ▶ Architecture (microprocessor instruction set)
 - ▶ Microarchitecture (microprocessor operation)
 - ★ **Multi-Cycle Processor - 4**

Concepts covered

- Datapath and Control Operation
 - ▶ lw instruction
 - ▶ sw instruction
 - ▶ R-type instructions
 - ▶ beq instruction

MULTI-CYCLE PROCESSOR - 4

MIPS Multi-Cycle Datapath



- 1w instruction steps
 - 0 Fetch
 - 1 Read register
 - 2 Read and sign-extend immediate
 - 3 Compute memory address
 - 4 Read data from memory
 - 5 Write data back to register file
 - 6 Increment PC

- 1w instruction steps

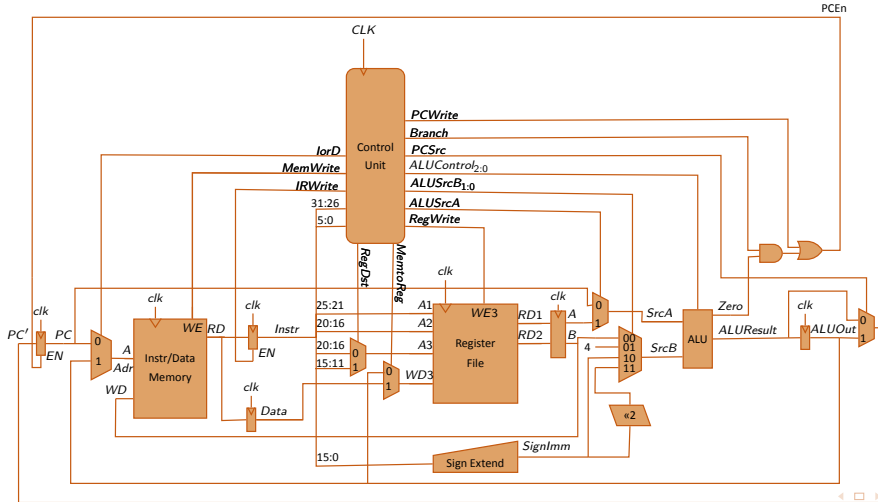
- 0 Fetch
- 1 Read register
- 2 Read and sign-extend immediate
- 3 Compute memory address
- 4 Read data from memory
- 5 Write data back to register file
- 6 Increment PC

- Execute 1w in five clock cycles

- 1 Fetch and increment PC
- 2 Read register and read/sign-extend immediate
- 3 Compute memory address
- 4 Read data from memory
- 5 Write data back to register file

MULTI-CYCLE PROCESSOR - 4

Load Word Instruction



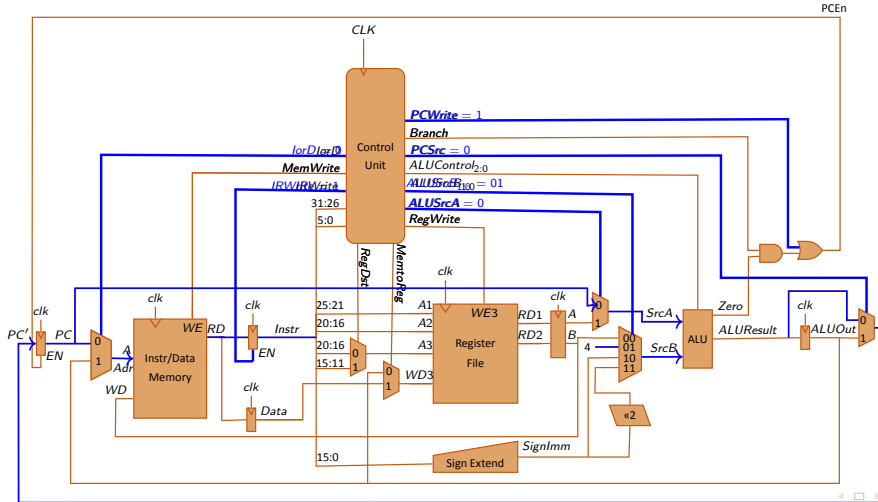
MULTI-CYCLE PROCESSOR - 4

Load Word Instruction



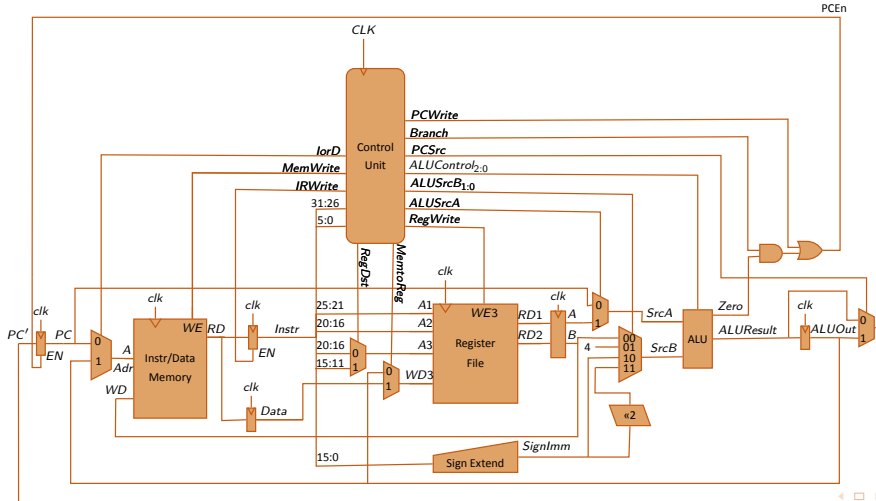
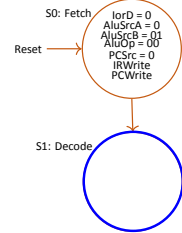
PES
UNIVERSITY
ONLINE

S0: Fetch
Reset
lorD = 0
AluSrcA = 0
AluSrcB = 01
AluOp = 00
PCSrc = 0
IRWrite
PCWrite



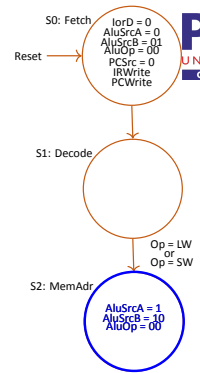
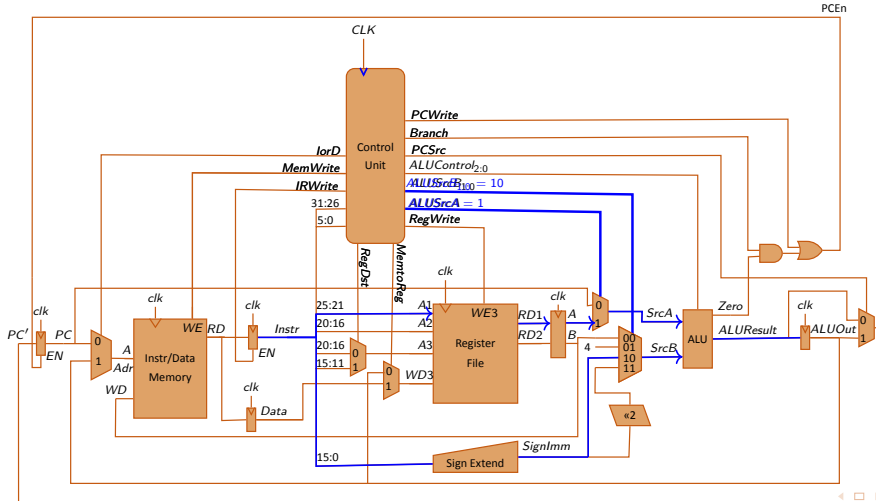
MULTI-CYCLE PROCESSOR - 4

Load Word Instruction



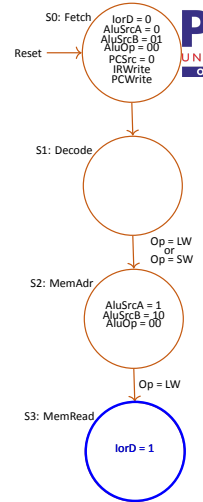
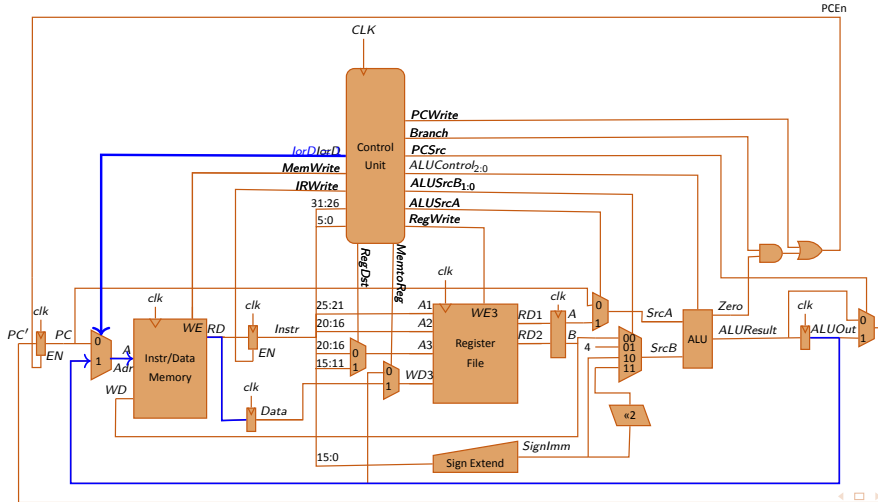
MULTI-CYCLE PROCESSOR - 4

Load Word Instruction



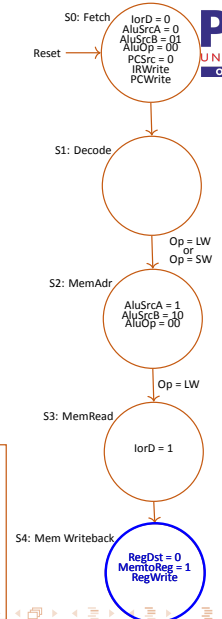
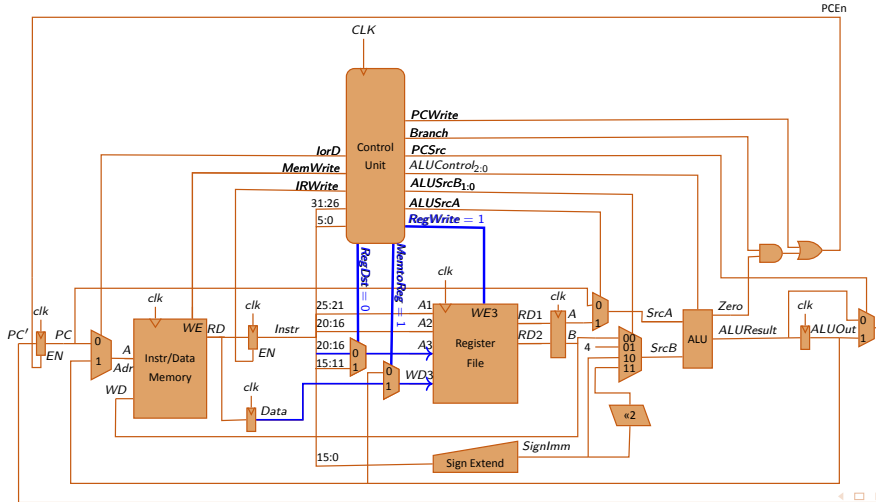
MULTI-CYCLE PROCESSOR - 4

Load Word Instruction



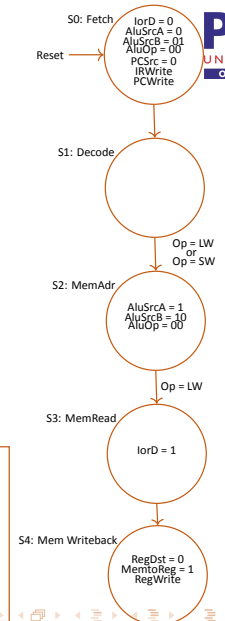
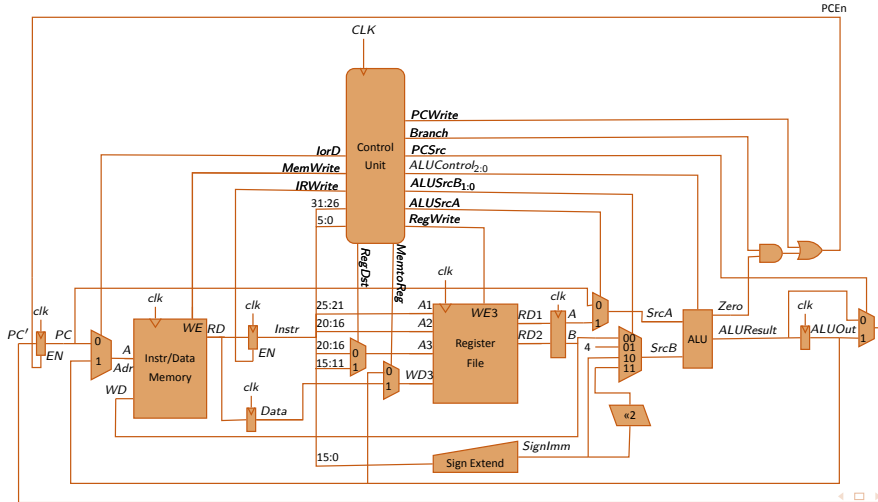
MULTI-CYCLE PROCESSOR - 4

Load Word Instruction



MULTI-CYCLE PROCESSOR - 4

Load Word Instruction

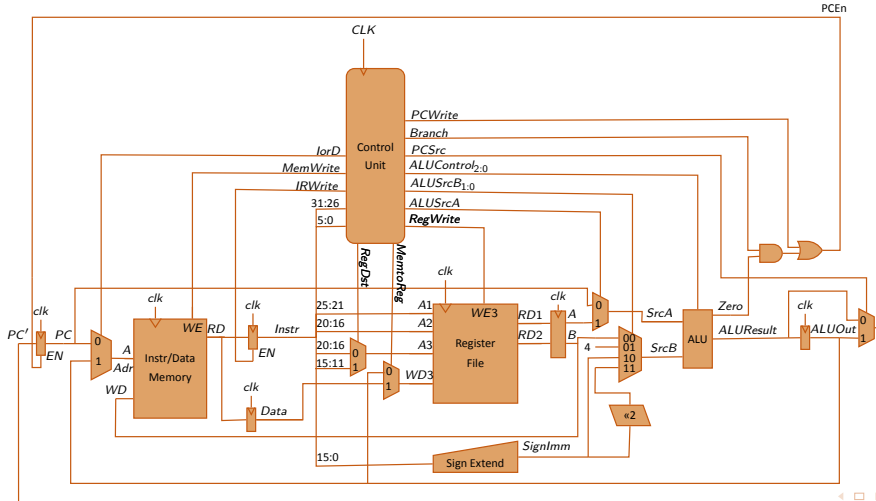


MULTI-CYCLE PROCESSOR - 4

Store Word Instruction

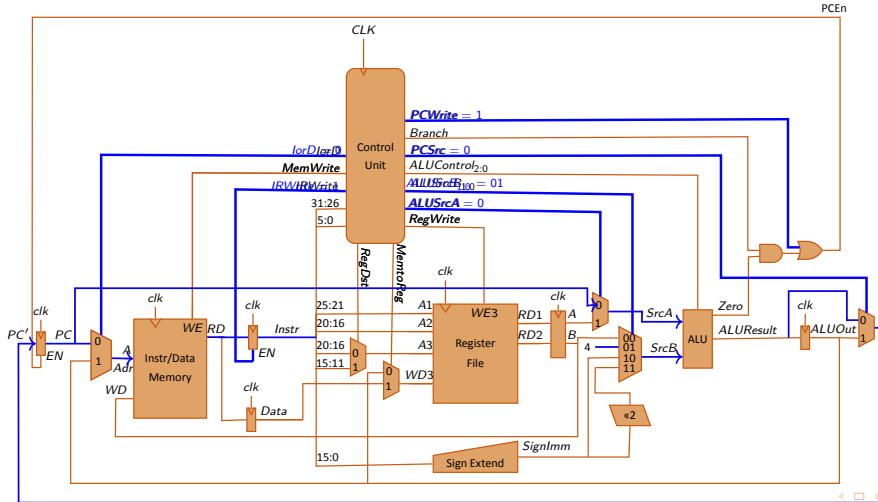


S0: Fetch
Reset →
lorD = 0
AluSrcA = 0
AluSrcB = 01
AluOp = 00
PCSrc = 0
IRWrite
PCWrite



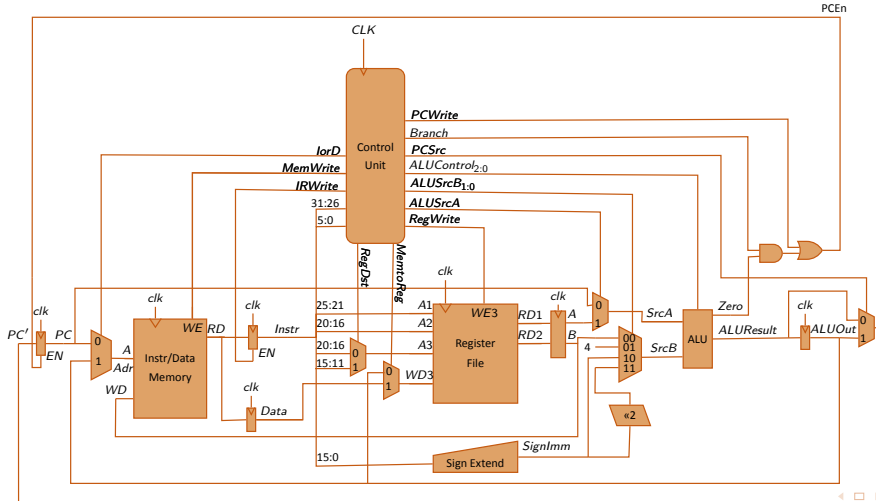
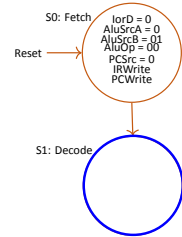
MULTI-CYCLE PROCESSOR - 4

Store Word Instruction



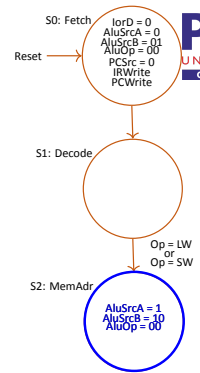
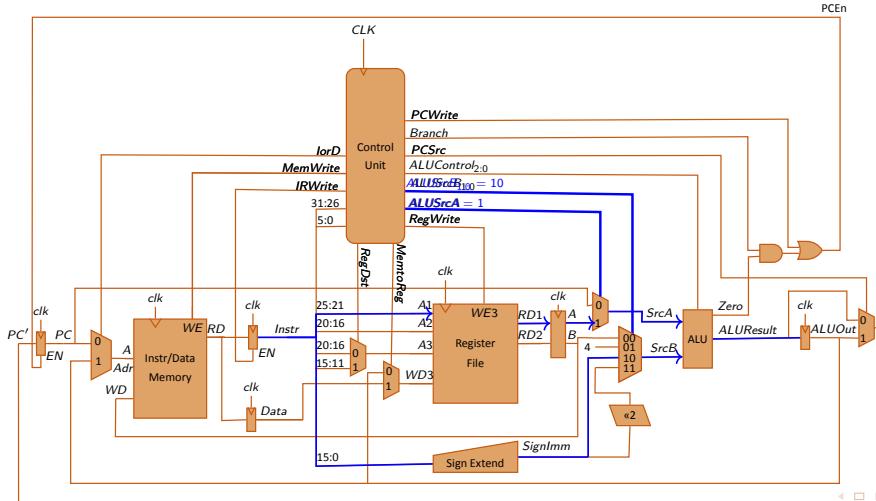
MULTI-CYCLE PROCESSOR - 4

Store Word Instruction



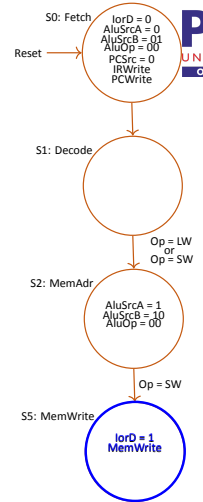
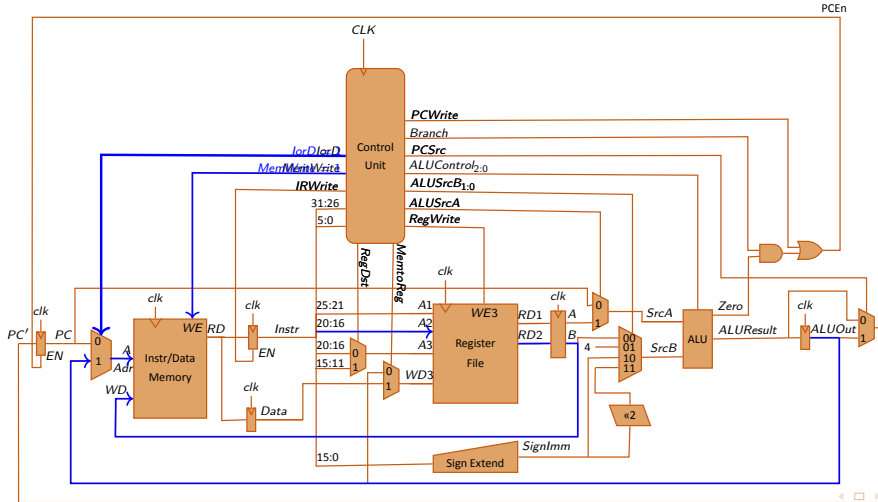
MULTI-CYCLE PROCESSOR - 4

Store Word Instruction



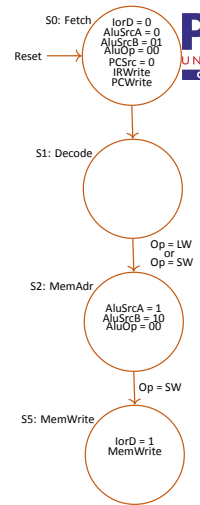
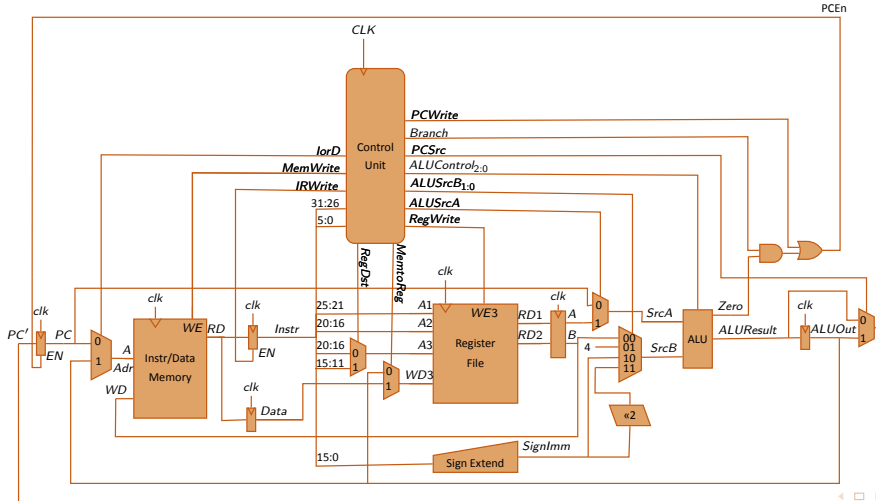
MULTI-CYCLE PROCESSOR - 4

Store Word Instruction



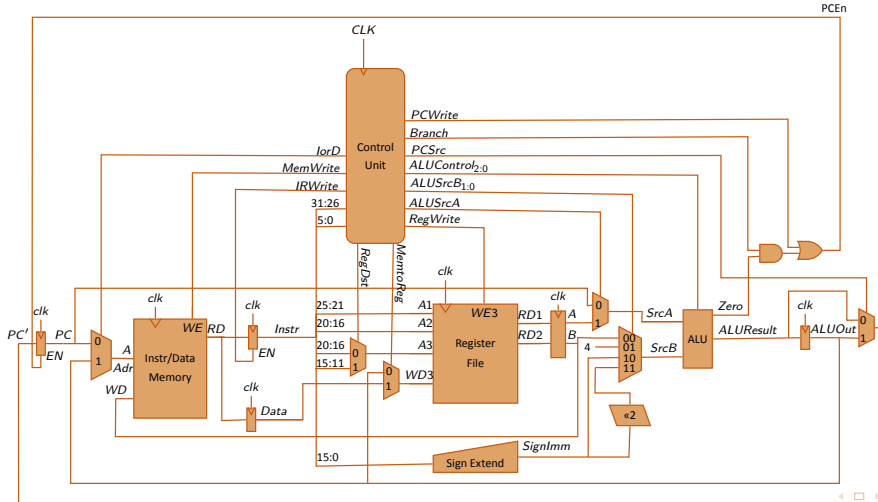
MULTI-CYCLE PROCESSOR - 4

Store Word Instruction



MULTI-CYCLE PROCESSOR - 4

R-Type Instructions

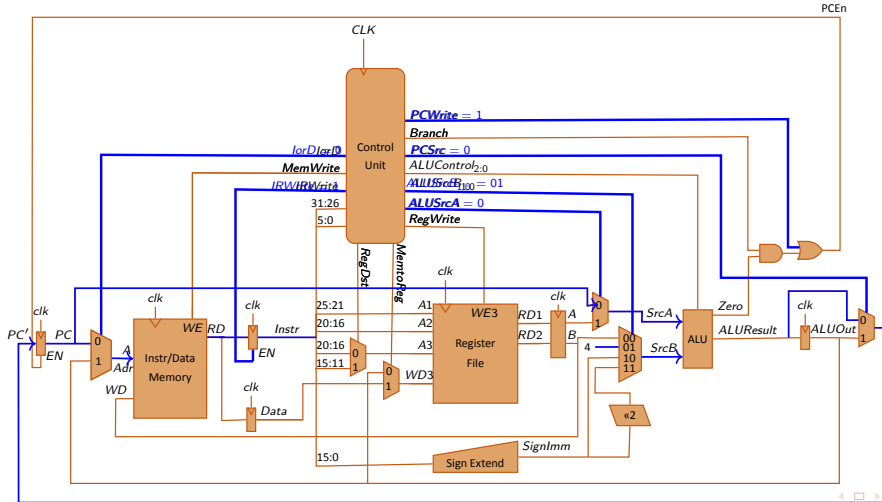


MULTI-CYCLE PROCESSOR - 4

R-Type Instructions

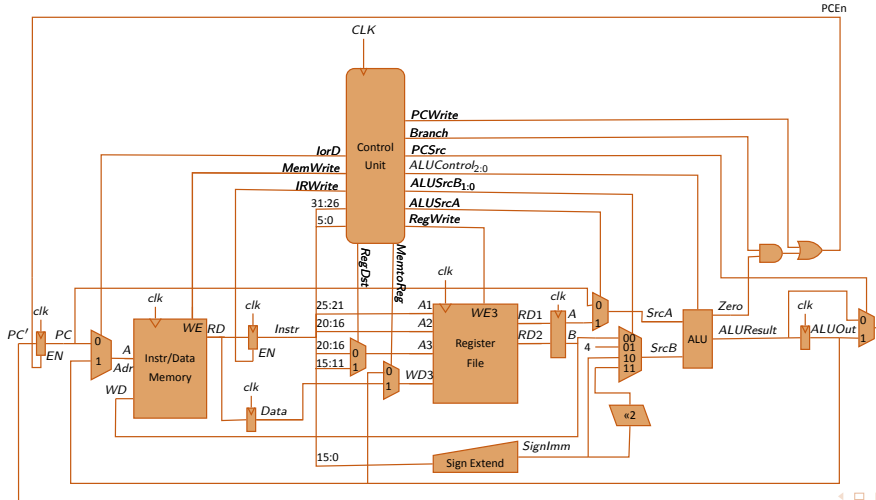
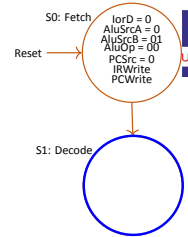


S0: Fetch
Reset
lorD = 0
AluSrcA = 0
AluSrcB = 00
AluOp = 00
PCSrc = 0
IRWrite
PCWrite



MULTI-CYCLE PROCESSOR - 4

R-Type Instructions

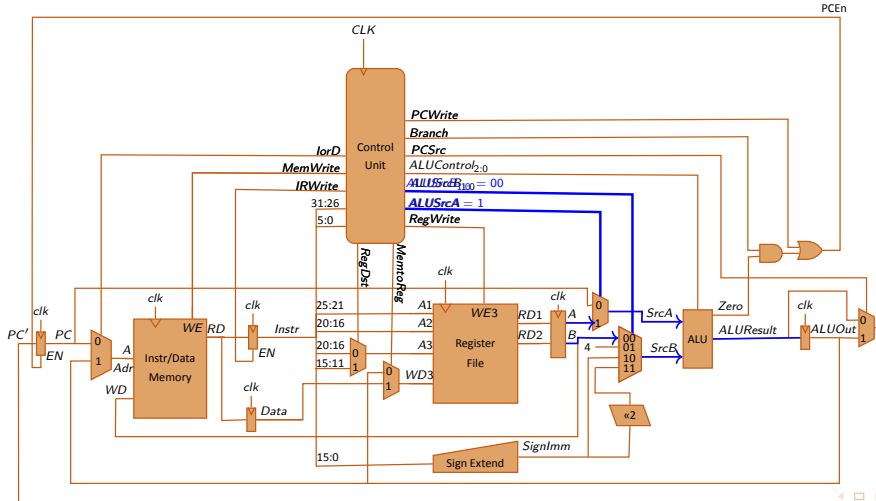
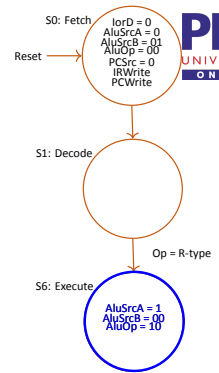


MULTI-CYCLE PROCESSOR - 4

R-Type Instructions



PES
UNIVERSITY
ONLINE

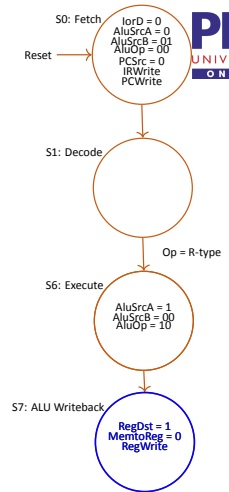
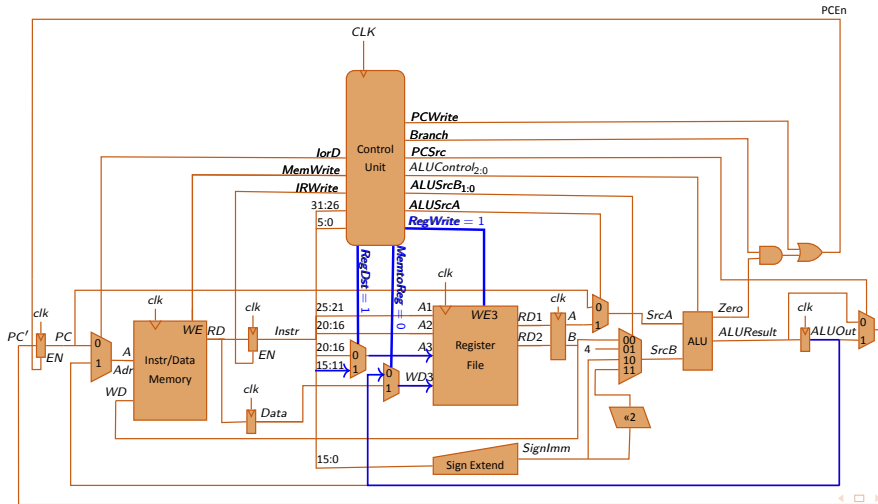


MULTI-CYCLE PROCESSOR - 4

R-Type Instructions

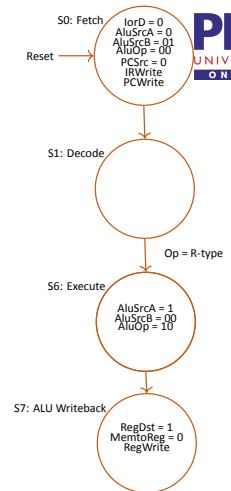
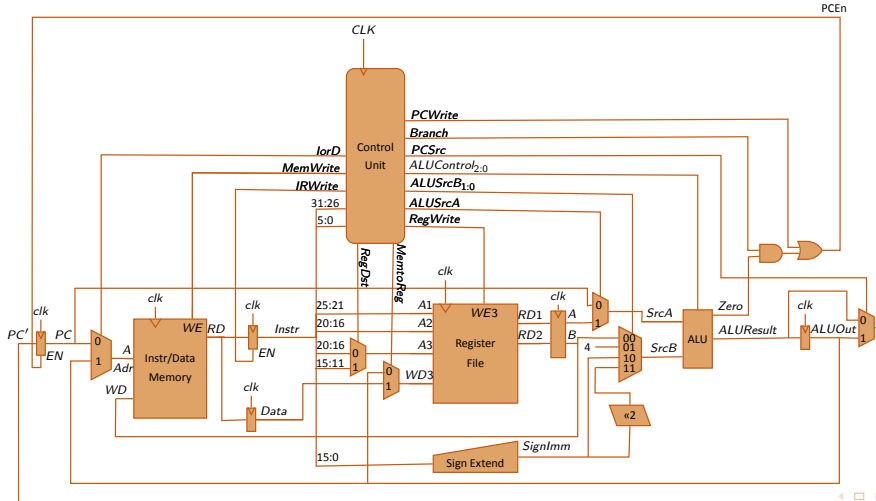


PES
UNIVERSITY
ONLINE



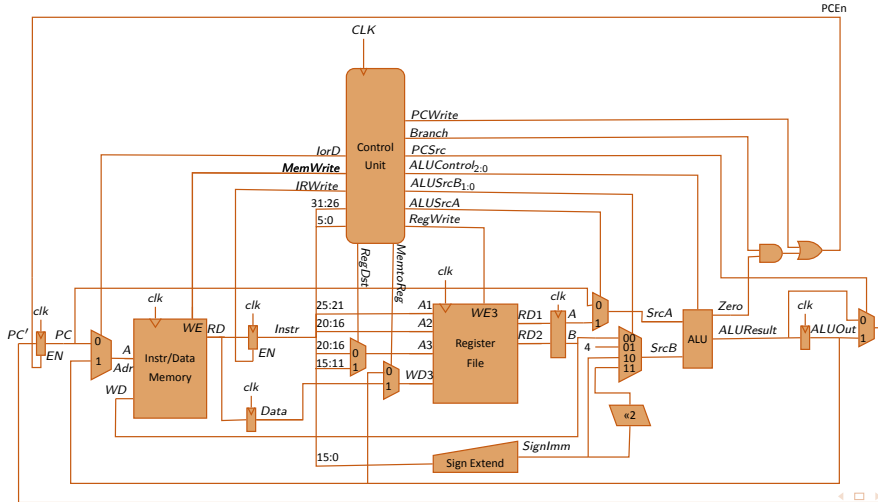
MULTI-CYCLE PROCESSOR - 4

R-Type Instructions



MULTI-CYCLE PROCESSOR - 4

Branch If Equal Instruction

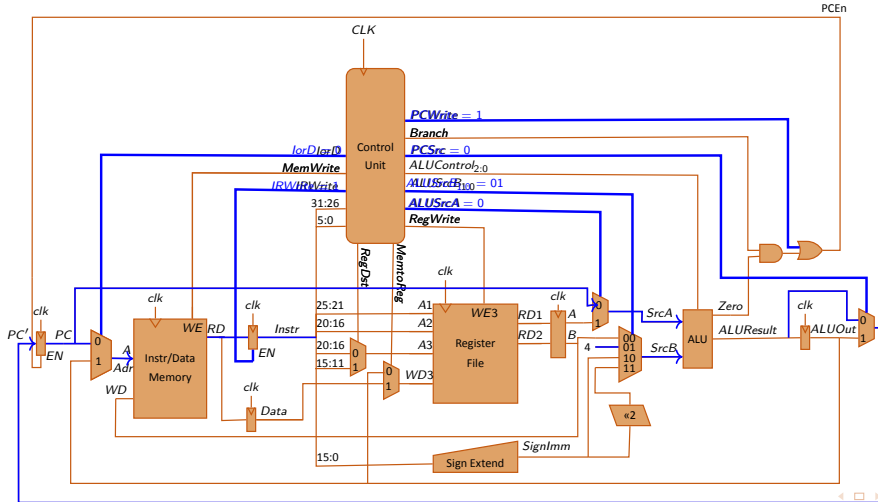


MULTI-CYCLE PROCESSOR - 4

Branch If Equal Instruction

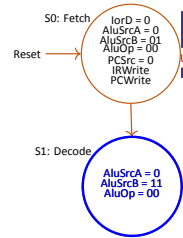
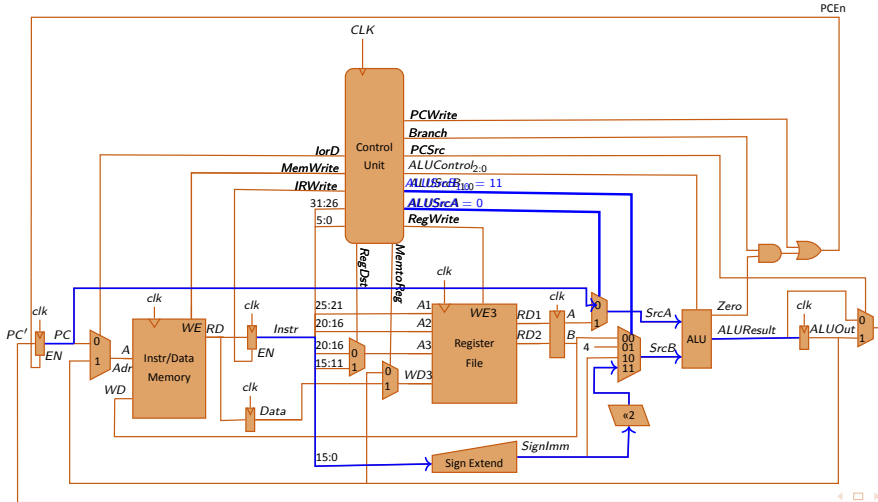


S0: Fetch
Reset
lorD = 0
AluSrcA = 0
AluSrcB = 00
AluOp = 00
PCSrc = 0
IRWrite
PCWrite



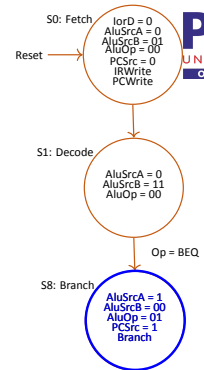
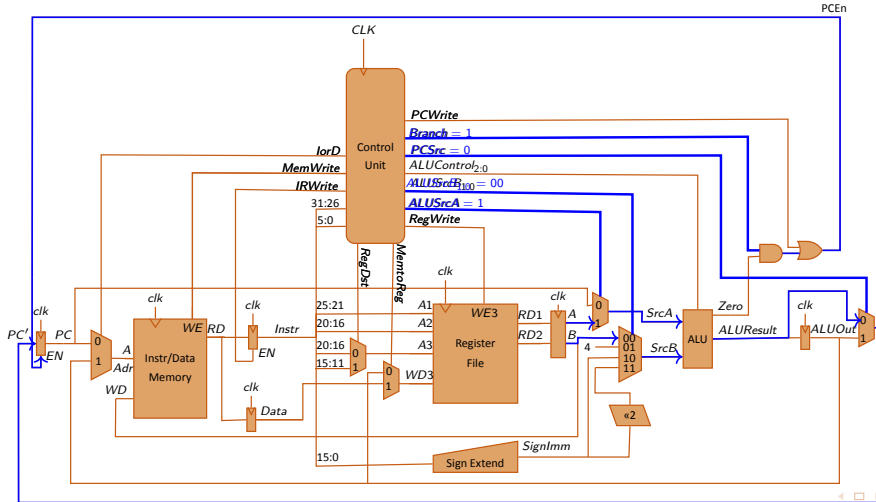
MULTI-CYCLE PROCESSOR - 4

Branch If Equal Instruction



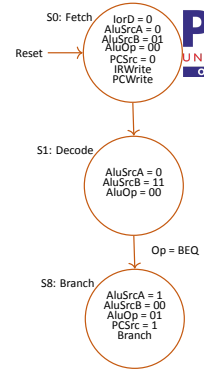
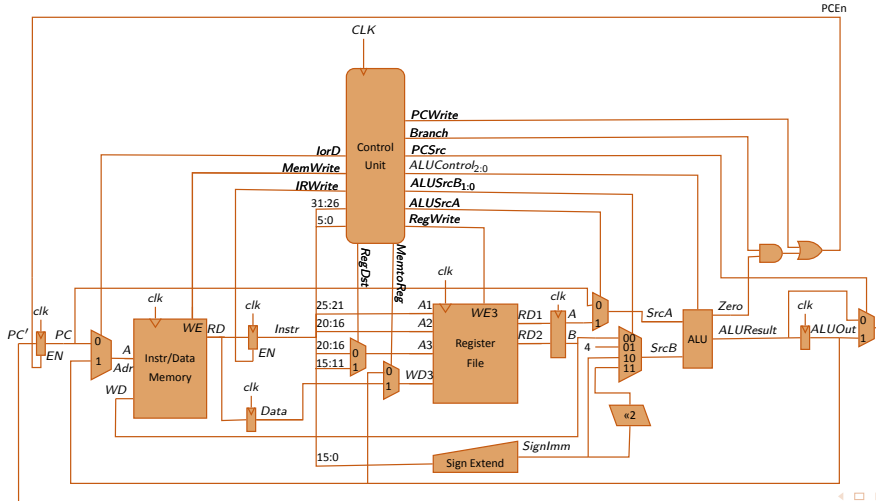
MULTI-CYCLE PROCESSOR - 4

Branch If Equal Instruction



MULTI-CYCLE PROCESSOR - 4

Branch If Equal Instruction



MULTI-CYCLE PROCESSOR - 4

Think About It

- How can the `addi` (add immediate) instruction be supported?
 - ▶ What changes, if any, are required to the datapath?
 - ▶ What states, if any, need to be added to the control logic?