TED ÜNİVERSİTESİ

CMPE361 Computer Organization

Department of Computer Engineering

TED University- Fall 2023

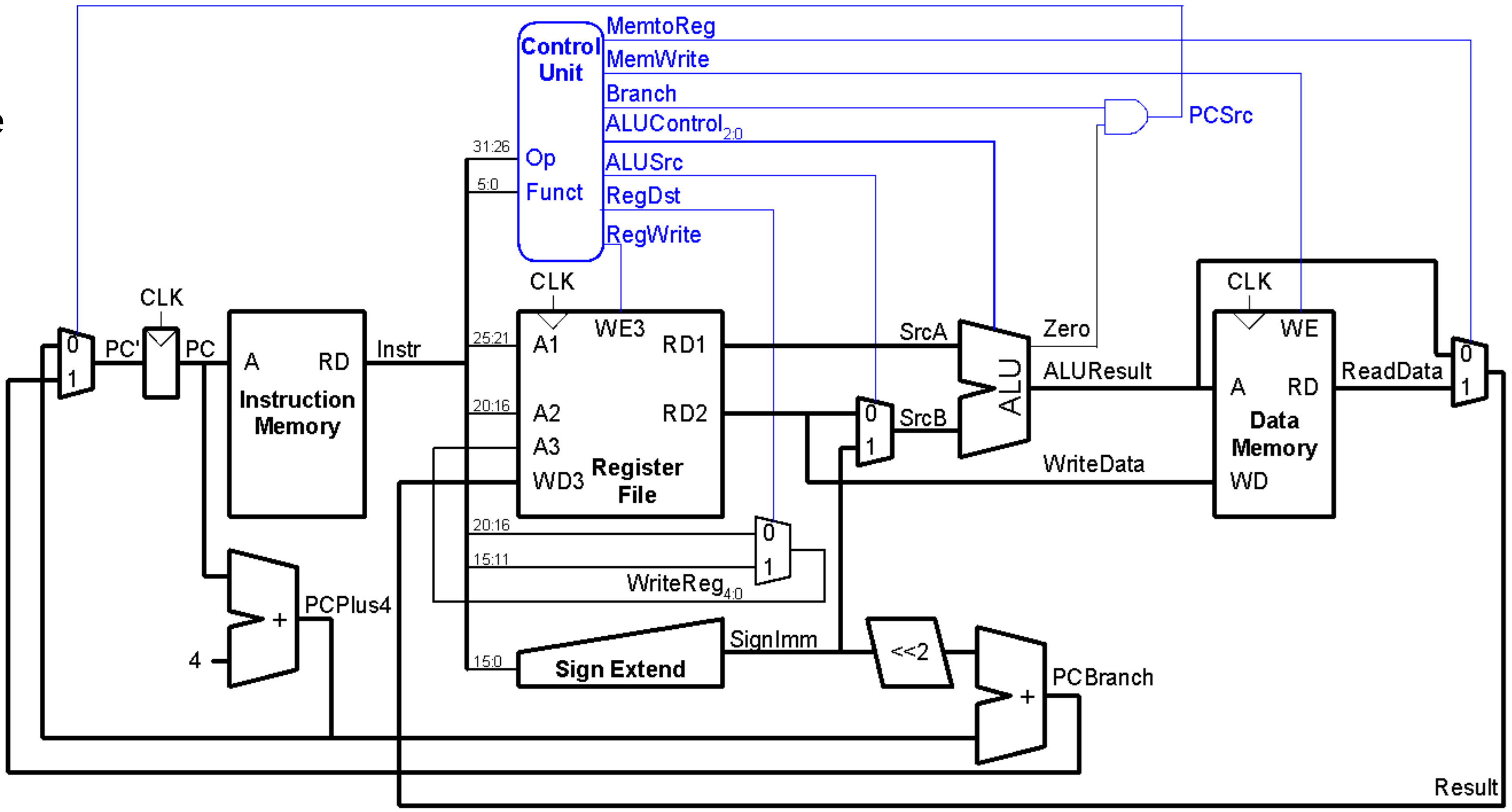
Extending the the sing cycle processor

These Slides are mainly based on slides of the text book (downloadable from the book’s website).

Extended Single-Cycle Processor

Extended Single-Cycle Processor Design with addi

An appropriate Signal combination is enough



No change to datapath!

Extend the processor with addi

- addi, adds the value in a register to the immediate and writes the result to another register.
- So extend the main **decoder truth table,** showing the control signal values for addi (*see the related table*)

Control Unit Truth Table for addi

Instruction	$Op_{5:0}$	RegWrite	RegDst	AluSrc	Branch	MemWrite	MemtoReg	$ALUOp_{1:0}$
R-type	000000	1	1	0	0	0	0	10
lw	100011	1	0	1	0	0	1	00
sw	101011	0	X	1	0	1	X	00
beq	000100	0	X	0	1	0	X	01
addi	001000							

addi instruction control signals

- The result is written to the register file, so RegWrite = 1.
- The destination register is in the rt field of the instruction, so RegDst = 0.
- SrcB comes from the immediate, so ALUSrc = 1.
- The instruction is not a branch, so Branch = 0
- There is no memory write, so MemWrite = 0.
- The result is from ALU, not memory, so MemtoReg = 0.
- ALU should add, so ALUOp = 00.

Control Unit: addi

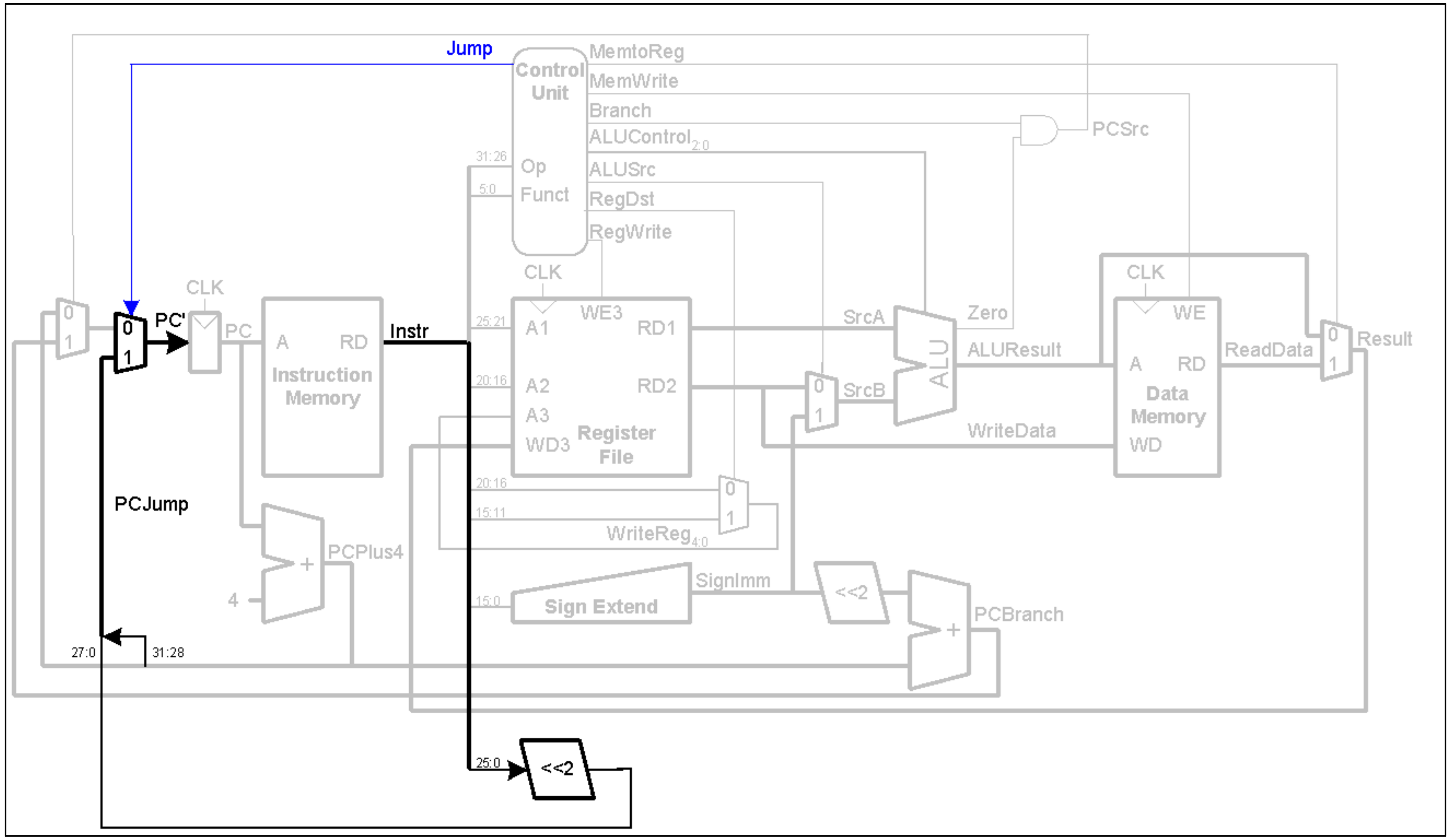
Instruction	$Op_{5:0}$	RegWrite	RegDst	AluSrc	Branch	MemWrite	MemtoReg	$ALUOp_{1:0}$
R-type	000000	1	1	0	0	0	0	10
lw	100011	1	0	1	0	0	1	00
sw	101011	0	X	1	0	1	X	00
beq	000100	0	X	0	1	0	X	01
addi	**001000**	**1**	**0**	**1**	**0**	**0**	**0**	**00**

The existing datapath is capable of addi with these control signals.

Extend the design with j Instruction

- The jump instruction j changes register PC.
- Computation of new PC value:
  - ✓ The two least significant bits of the PC are always 0, because the PC is word aligned (i.e., always a multiple of 4).
  - ✓ The next 26 bits are taken from the jump address field in Instr25:0.
  - ✓ The upper 4 bits are taken from the old value of the PC, to complete to 32 bits.

Extended Design for j



Extended Design for j

- A new multiplexer is used to be able to select the next instruction to be executed.
- The Main Decoder Truth table can also be updated to allow j instruction:
  - Table entries with X indicate that they have no affect on the outcome of j instruction.

Control Unit: Main Decoder

Instruction	$Op_{5:0}$	RegWrite	RegDst	AluSrc	Branch	MemWrite	MemtoReg	$ALUOp_{1:0}$	Jump
R-type	000000	1	1	0	0	0	0	10	0
lw	100011	1	0	1	0	0	1	00	0
sw	101011	0	X	1	0	1	X	00	0
beq	000100	0	X	0	1	0	X	01	0
j	000010	0	X	X	X	0	X	XX	1

Performance isue

Execution Time

Program Execution Time =
(#instructions)(cycles/instruction)(seconds/cycle)

- Definitions:
  - CPI: Cycles/instruction
  - clock period: seconds/cycle
  - IPC: instructions/cycle = IPC
- Measurement of the execution time of an instruction need to consider the entire execution path

Execution Time Critical path

- Time performance is normally function of the Critical Path (CP)
  - The longest path through the architecture.
- How to determine CP?
  - Add up the slowest actions along the path, from a PC load to a new PC load…

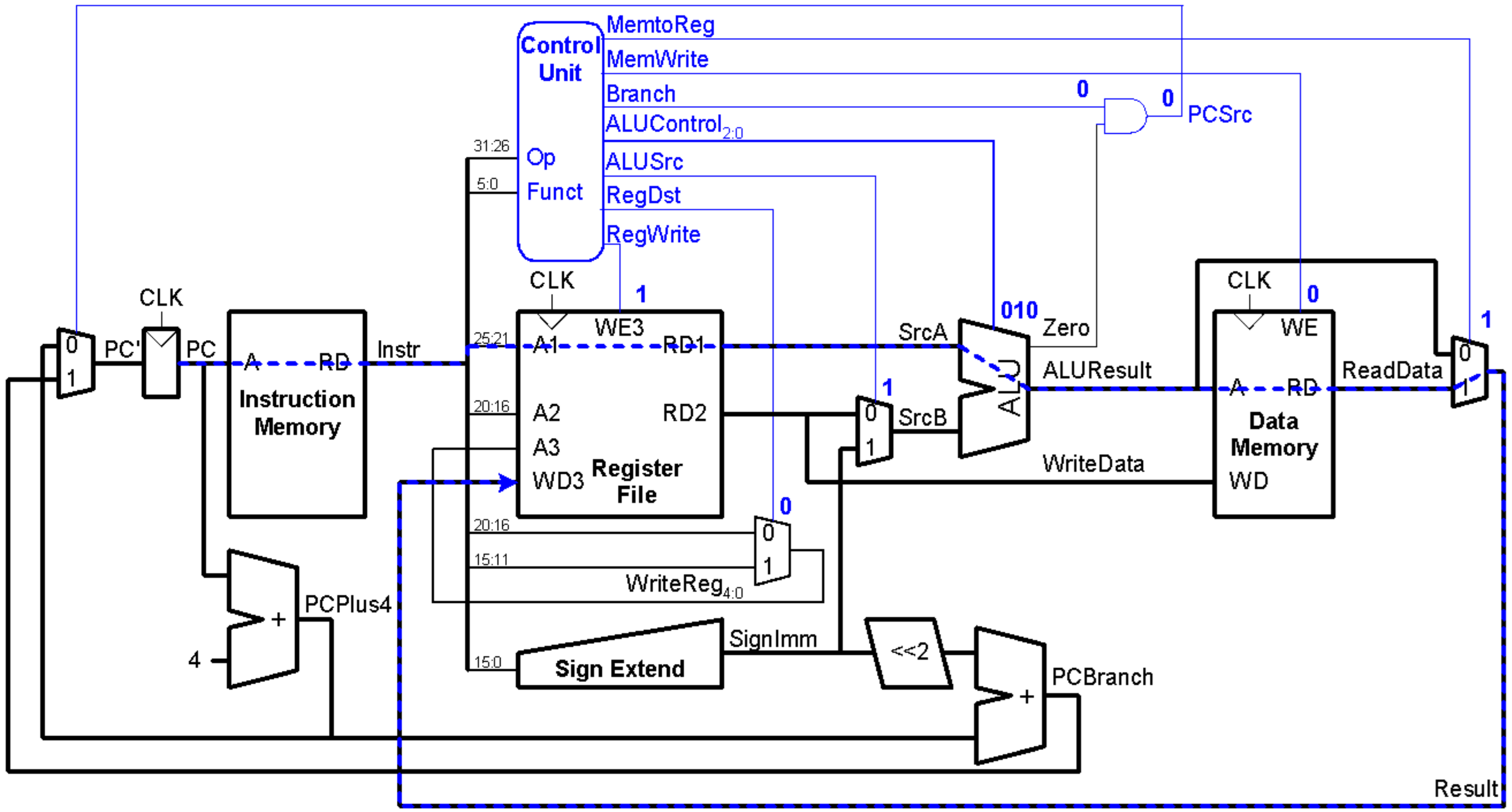
lw CP (see figure) 1

- A PC is loaded on the rising edge of the clock.
- The instruction memory reads the next instruction.
- The register file reads SrcA, the base register
- At the same time the immediate field is sign-extended to be selected at the related multiplexer to determine SrcB.

lw CP 2

- The ALU adds SrcA and SrcB to find the effective address.
- The data memory reads from this address.
- The MemtoReg multiplexer selects ReadData.
- Finally, Result must setup at the register file before the next rising clock edge, be be written.

Single-Cycle lw Performance



T_C limited by critical path (`lw`)

Performance: Tc is length of CP

- Single-cycle critical path:

$$T_c = t_{pcq\_PC} + t_{\text{mem}} + \max(t_{RF\text{read}}, t_{sext} + t_{\text{mux}}) + t_{\text{ALU}} + t_{\text{mem}} + t_{\text{mux}} + t_{RF\text{setup}}$$

$$T_c = t_{pcq\_PC} + 2t_{\text{mem}} + t_{RF\text{read}} + t_{\text{mux}} + t_{\text{ALU}} + t_{RF\text{setup}}$$

- Limiting paths are the ones related to memory, ALU and register file operation
- T_{pcq} stands for the time of propagation from clock to output (Q).

Given the given timing details in the table compute T_c

Element	Parameter	Delay (ps)
Register clock-to-Q	t_{pcq_PC}	30
Register setup	t_{setup}	20
Multiplexer	t_{mux}	25
ALU	t_{ALU}	200
Memory read	t_{mem}	250
Register file read	$t_{RF\text{read}}$	150
Register file setup	$t_{RF\text{setup}}$	20

$$\begin{aligned} T_c &= t_{pcq\_PC} + 2t_{\text{mem}} + t_{RF\text{read}} + t_{\text{mux}} + t_{\text{ALU}} + t_{RF\text{setup}} \\ &= [30 + 2(250) + 150 + 25 + 200 + 20]\text{ ps} \\ &= 925\text{ ps} \end{aligned}$$

Compute Tc for 100 billion instructions:

Execution Time = # instructions x CPI x T_C

$= (100 \times 10^{9})(1)(925 \times 10^{-12}\text{ s})$

= **92.5 seconds**