TED ÜNİVERSİTESİ

CMPE361 Computer Organization

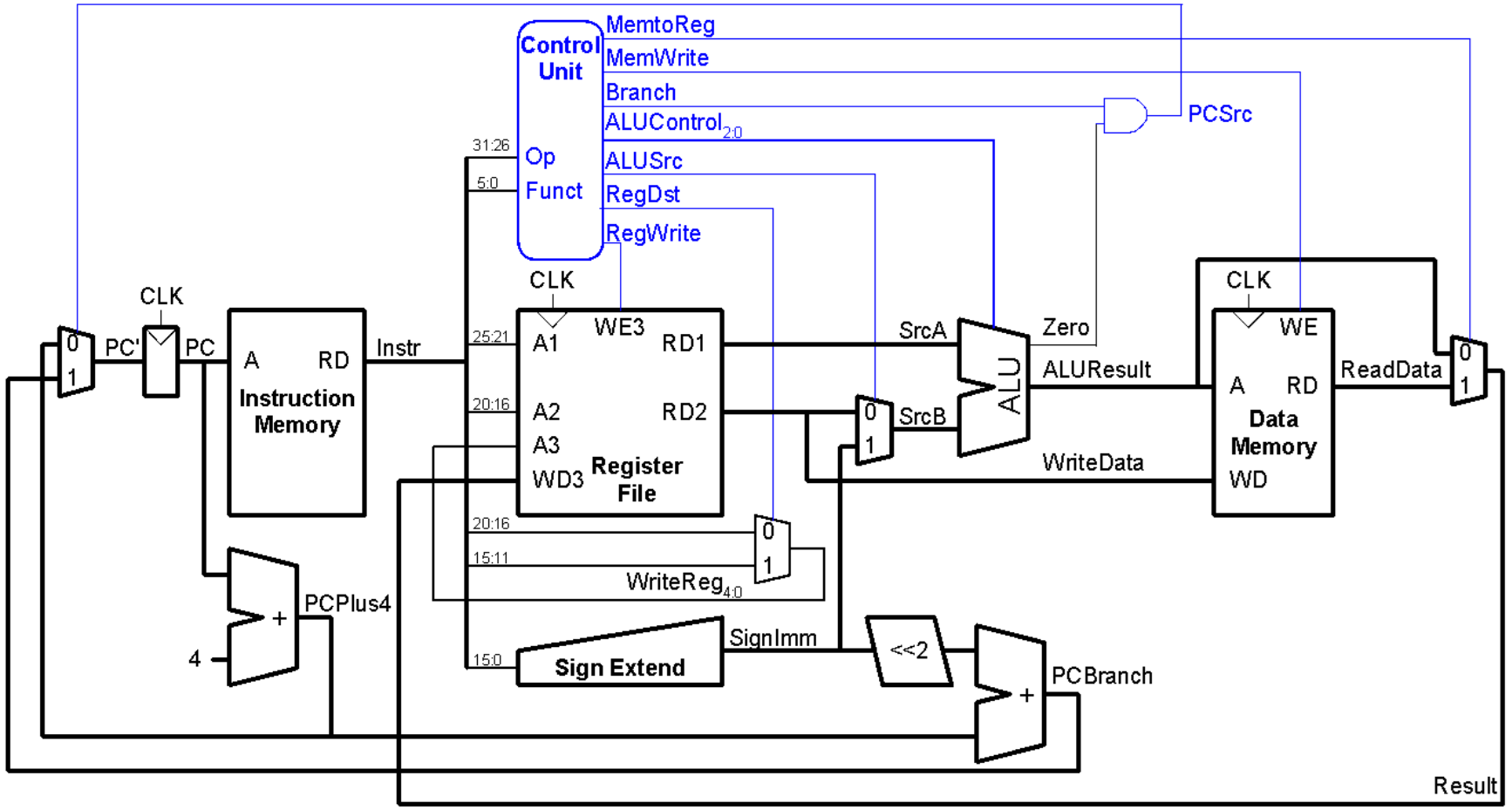
Department of Computer Engineering
TED University- Fall 2023

Single Cycle Processor - II

Single-Cycle Control

- In a single-cycle datapath design, the control signals are inserted whenever required.
- All control signals can be assembled in a dedicated circuit blocks.

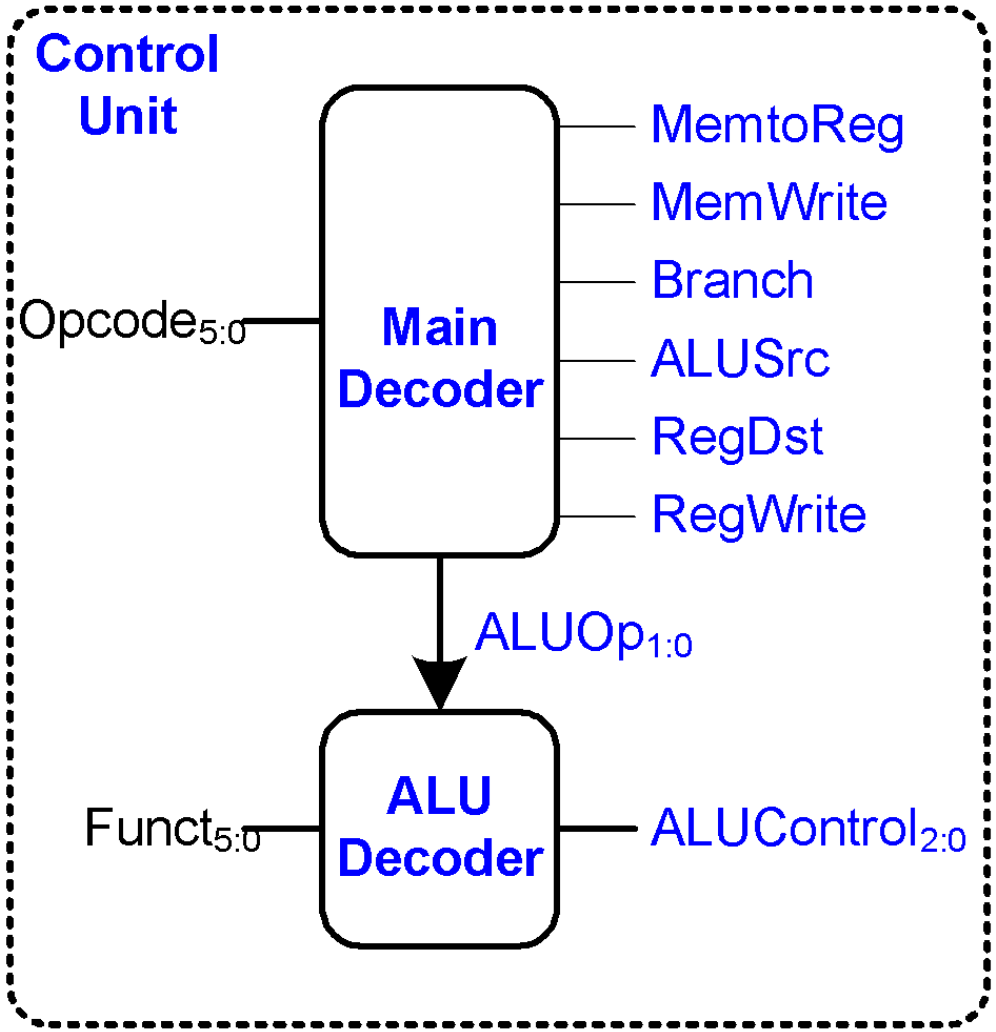
Single-Cycle Main Control Unit with multiple ALU



Single-Cycle Control

- The desired values of control signals can be generated from 6-bit operation code field and 6-bit function fields $Instr_{31:26}$ and $Instr_{5:0}$.
- The control can be divided into two decoding units.
  - ✓ The **main decoder** computes most of control signals related nonzero opcode.
  - ✓ It uses 2-bit ALUOp signal to let **ALU decoder block** to compute the control signals based on the function field.

Internal structure of SCD: main decoder, ALU decoder



Single-Cycle Control

- ALUOp signal can take values 00, 01, 10, 11.
  - ✓ ALUOp of 10 examines to function field to determine the control signal, 00 is add, 01 is subract.
- Both lw and sw require add operation to compute the target address,
- beq requires sub operation
- A truth table for decoding logic can be used to help creating the proper signals

Control Unit: Truth Table for the Main ALU Decoder 1

Extend the Truth table for 2 bit $ALUOp_{1-0}$

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

Control Unit: Truth Table for the Main ALU Decoder 2

R-Type instruction depends on function code

Instruction	$Op_{5:0}$	RegWrite	RegDst	AluSrc	Branch	MemWrite	MemtoReg	$ALUOp_{1:0}$
R-type	000000	1	1	0	0	0	0	Check funct code
`lw`	100011	1	0	1	0	0	1	add
`sw`	101011	0	X	1	0	1	X	add
`beq`	000100	0	X	0	1	0	X	sub

Control Unit: Truth Table for the Main ALU Decoder 3

$ALUOP_{1-0}$ is determined to control ALU datapath

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

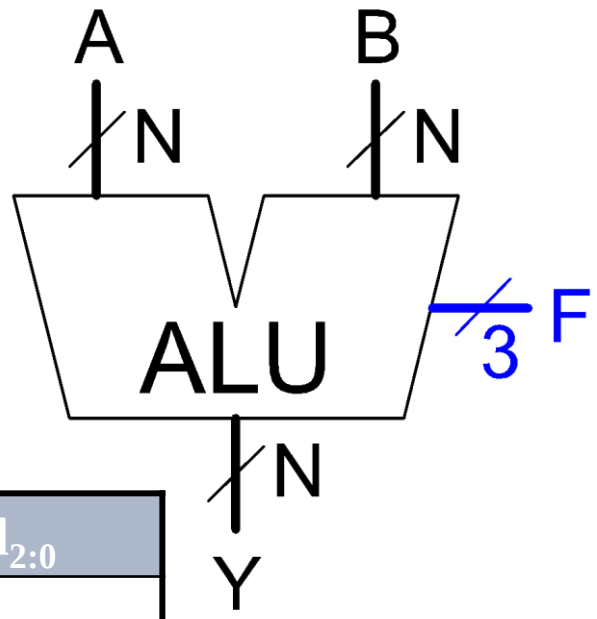
ALU decoder output

- For the instructions that do not write to the register file (e.g., sw and beq), the RegDst and MemtoReg control signals are don't cares (X);
- The address and data arriving at the register write port do not matter, because RegWrite is not asserted.
- When ALUOp is 10, the ALU decoder examines the funct field to determine the $ALUControl_{2\text{-}0}$ signals.

Summary of control signals

$ALUOp_{1:0}$	Meaning
00	Add
01	Subtract
10	Look at Funct
11	Not Used

$ALUOp_{1:0}$	Funct	$ALUControl_{2:0}$
00	X	010 (Add)
X1	X	110 (Subtract)
1X	100000 (`add`)	010 (Add)
1X	100010 (`sub`)	110 (Subtract)
1X	100100 (`and`)	000 (And)
1X	100101 (`or`)	001 (Or)
1X	101010 (`slt`)	111 (SLT)



$F_{2:0}$	Function
000	A & B
001	A | B
010	A + B
011	not used
100	A & ~B
101	A | ~B
110	A - B
111	SLT

$ALUControl_{2-0}$ Signal

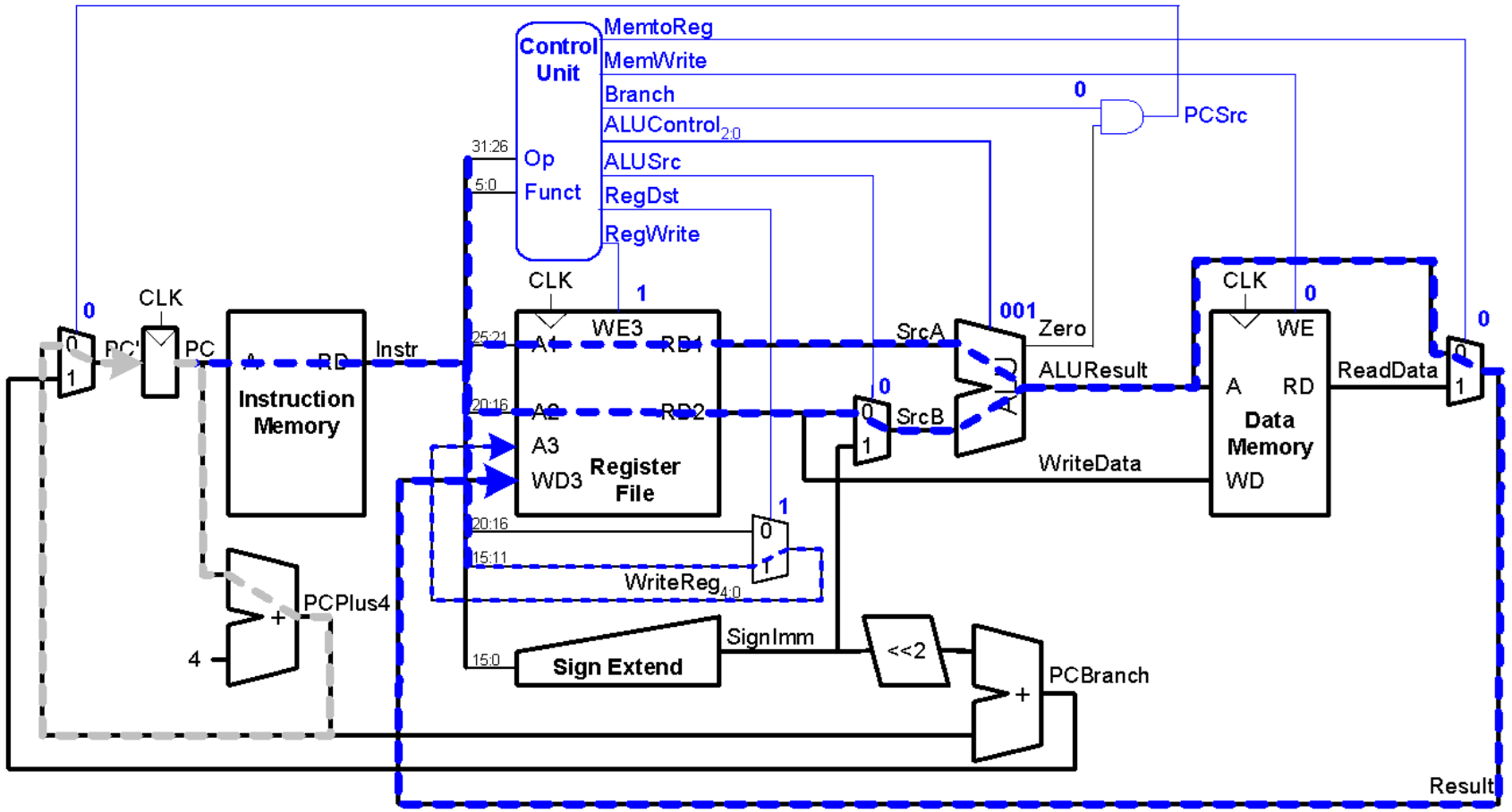
Extend SCP with or instruction (Example 7.1)

- Determine the values of the control signals and the portions of the datapath that are used when executing an or instruction.
- or is an R-type instruction.

R-Type

- In the following figure, the main flow of data through the register file and ALU is represented with a dashed blue line.
- The control signals are shown by the light blue color

Control and data flow for or instruction



Instruction or control signals

- The register block reads the two source registers specified by Instr25:21 and Instr20:16 bits.
- SrcB come from the second port (A1) of the register file (not SignImm), so ALUSrc = 0, ALUOp = 10
- Result is taken from the ALU, so MemtoReg signal is 0
- RegWrite is 1, MemWrite = 0
- The destination register is specified in the rd field, Instr15:11, so RegDst = 1
- Branch = 0, PCSrc = 0 to get PC next value from PCPlus4.

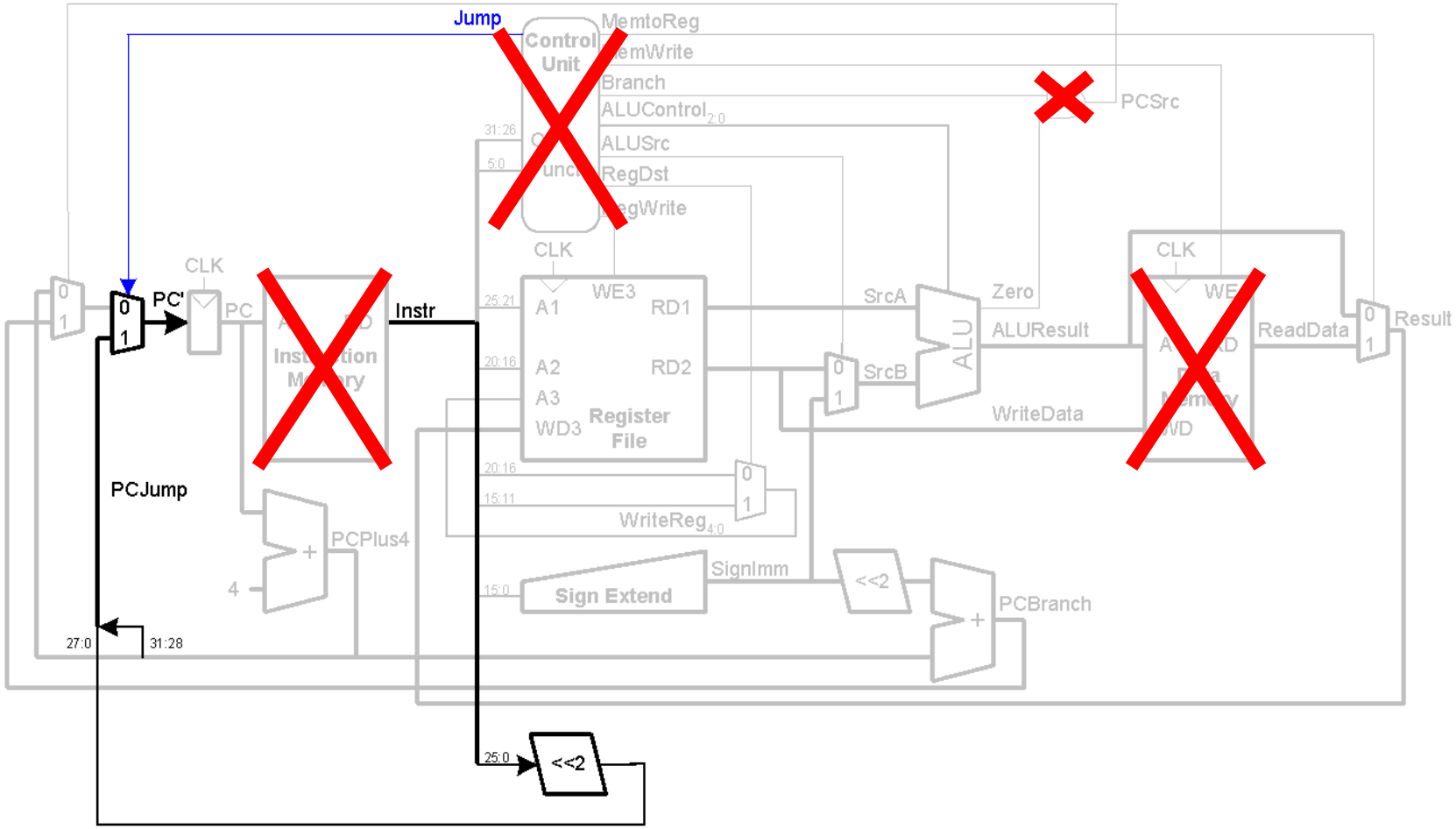
Or instruction design

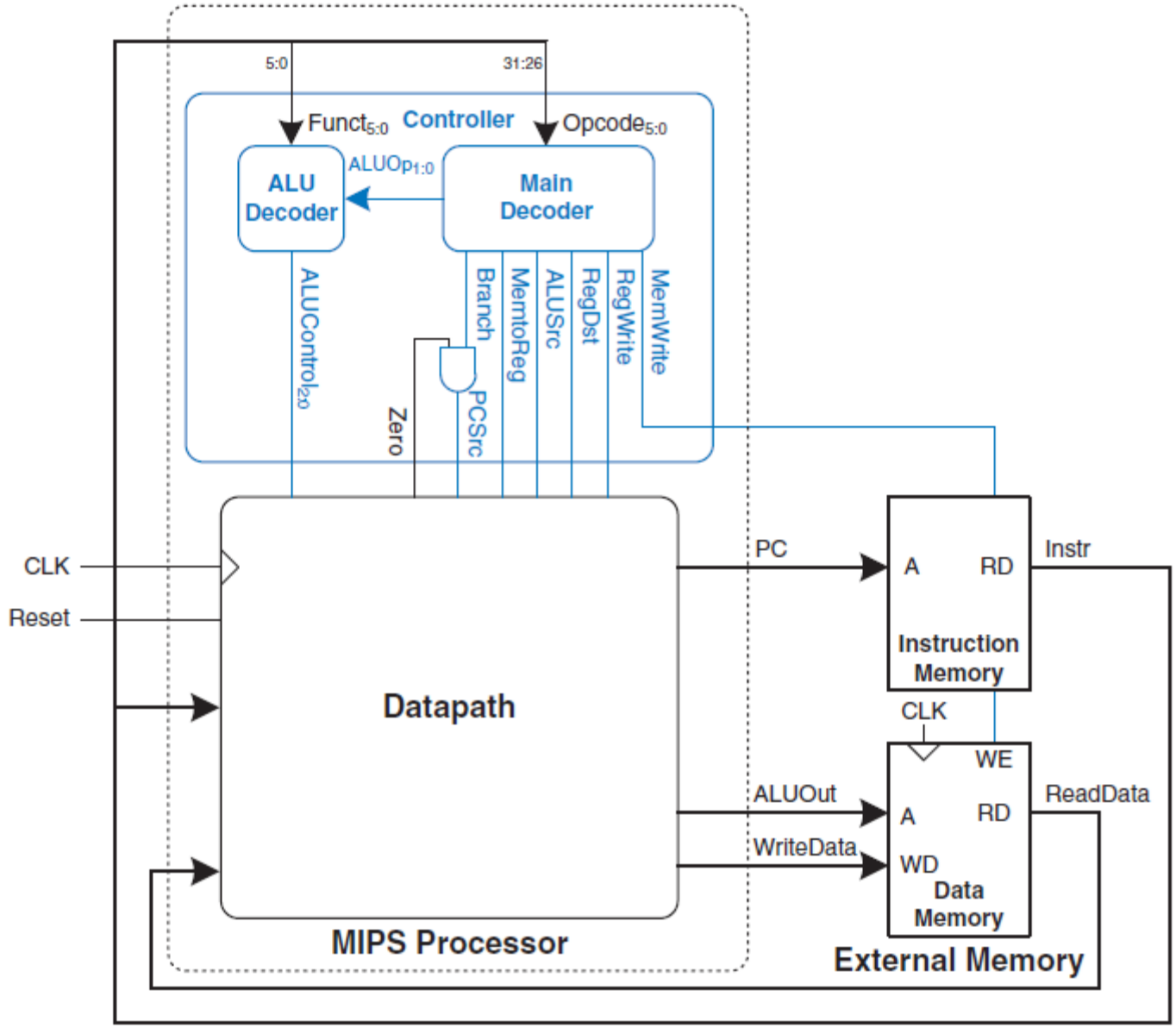
Note that data can always flow through the available paths; however, the control signals configured for each instruction will allow only the right ones to to be effective...

Re-arranging the control and datapath

- The separation of instruction and data memories from rest of the processor seems a better design.
  - The memory is connected with address and data busses to the processor.
- The controller can also be configured into 2 parts: main decoder, ALU decoder.

Re-arranging the modules




5:0
31:26
Funct5:0
Controller
Opcode5:0
ALU Decoder
ALUOp1:0
Main Decoder
ALUControl2:0
Branch
MemtoReg
ALUSrc
RegDst
RegWrite
MemWrite
Zero
PCSrc
CLK
Reset
Datapath
PC
A
RD
Instr
Instruction Memory
CLK
WE
ALUOut
A
RD
ReadData
WriteData
WD
Data Memory
MIPS Processor
External Memory