Hardware Design

Tutorial 4

Instructor: Dr. Haiyu Mao
TA: Zihao Pu

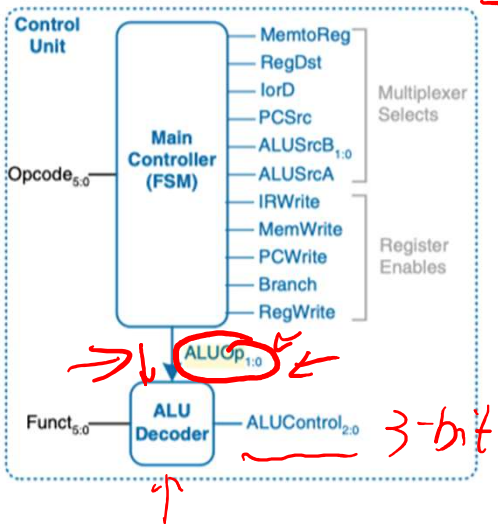
20.02.2026

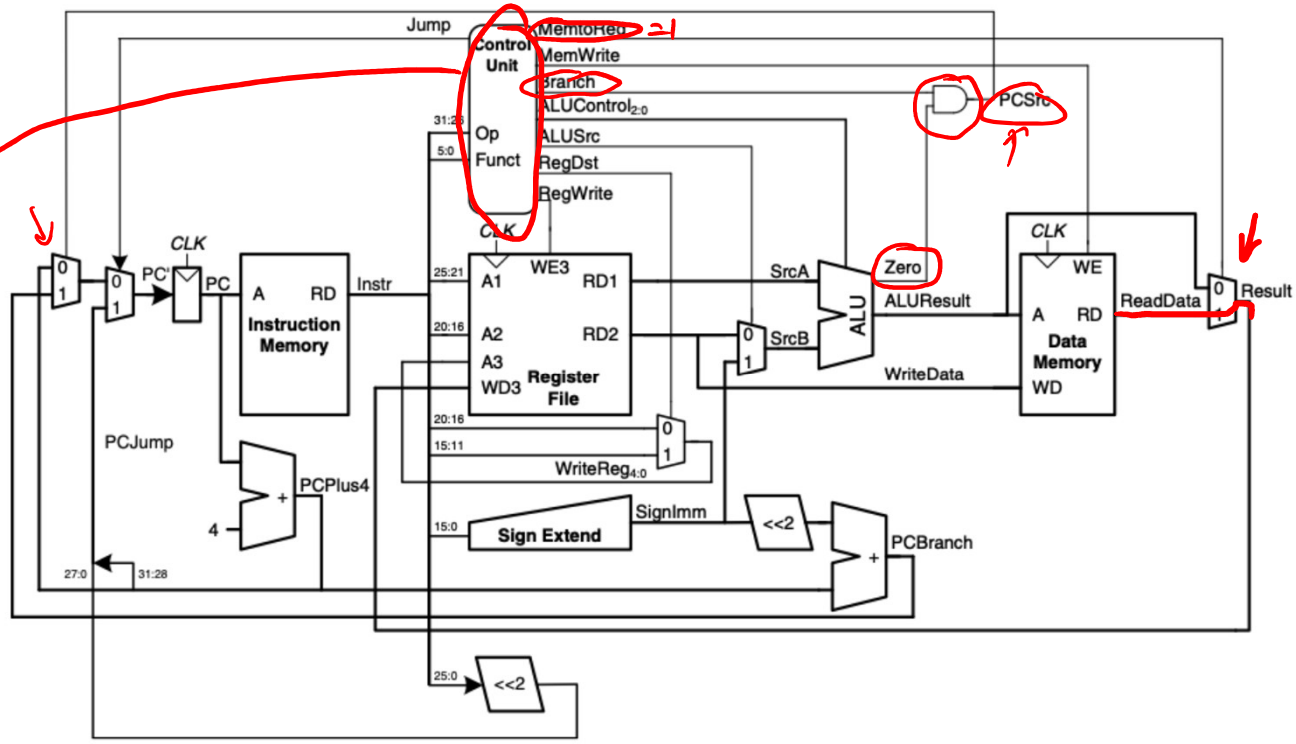
Question 1

MICROARCHITECTURE - MALFUNCTIONS

Microarchitecture - malfunctions

- Suppose one of the following control signals in the single-cycle MIPS processor has a stuck-at-1 fault, meaning that the signal is always 1, regardless of its intended value. Which of the instructions shown in **Table 2** would malfunction? Why?
- (a) MemtoReg
- (b) ALUOp0
- (c) PCSrc





Microarchitecture - malfunctions

a). R-type, addi

b) R-type, lw, sw, addi

c). R-type, lw, sw, addi, beq.

Table 1: ALUOp Encoding Table

***ALUOp* encoding**

ALUOp	Meaning
~~00~~	~~add~~
01	subtract
~~10~~	look at `funct` field
11	n/a

0 AND X = 0

MemtoReg = 1

ALUop 0 => 0 bit of ALUop

Table 2: Supported Instructions and Main decoder's decoding table

Instruction	Opcode	RegWrite	RegDst	ALUSrc	Branch	MemWrite	MemtoReg	ALUOp	Jump
R-type	000000	1	1	0	0	0	0 => 1	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
`addi`	001000	1	0	1	0	0	0	00	0
`j`	000010	0	X	X	X	0	X	XX	1

Answer

Answer:

a) R-type, addi. Calculated results cannot be writen back to register file.
b) R-type, lw, sw, addi
c) R-type, lw, sw, bew, addi

Question 2

MICROARCHITECTURE - IMPLEMENTATION

Question 2: Datapath Modification

Modify the single-cycle MIPS processor to implement one of the following instructions.

(1) Mark up a copy of Figure 1 and Figure 2 to indicate the changes to the datapath, and name any new control signals.

(2)Mark up a copy of Table 1 and Table 2 to show the changes.

(3)Describe any other changes that are required.

New instruction: sll

Meaning: Shift Logical Left, R-type

MIPS Single-cycle datapath

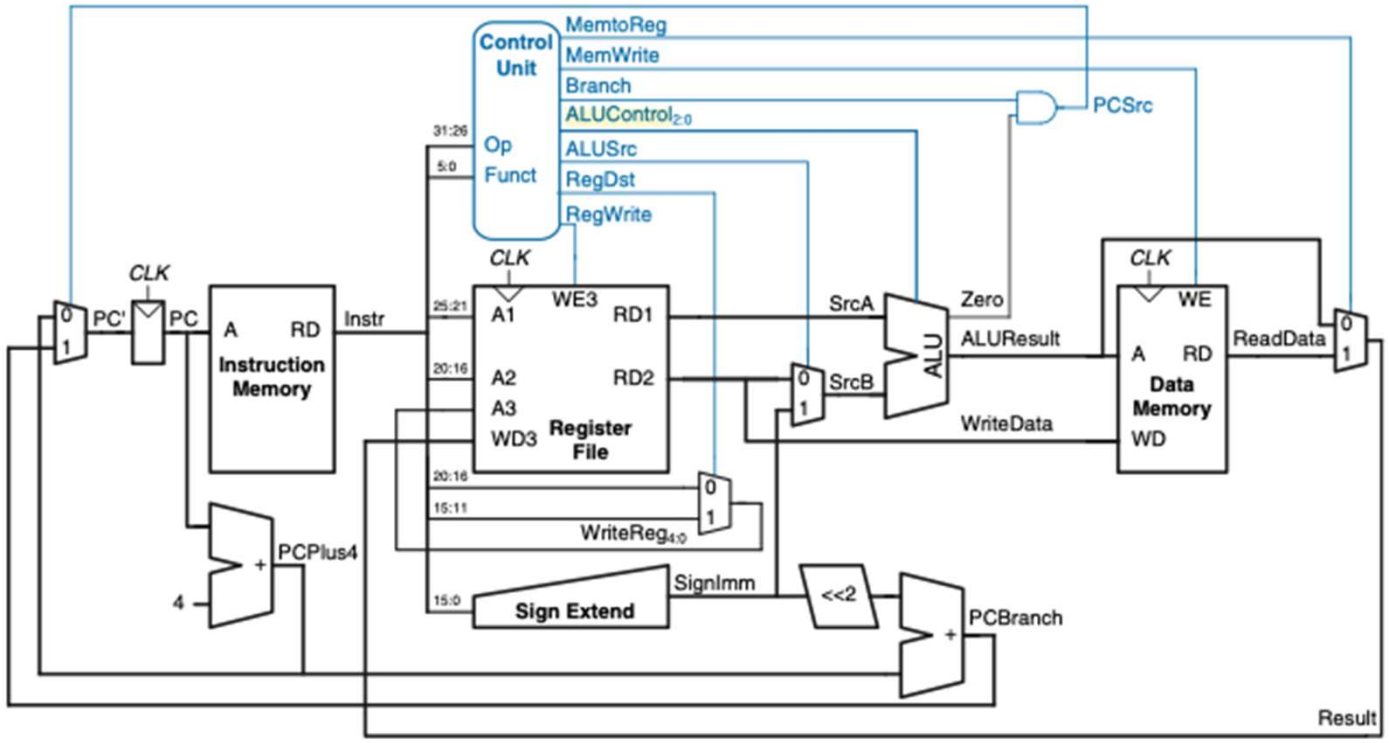


Figure 1: Current Datapath

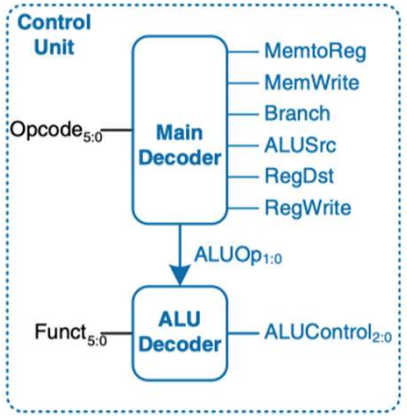


Figure 2: Control Unit Internal Structure

Main Decoder and ALU Decoder

Table 1: Main Decoder

Instruction	Opcode	RegWrite	RegDst	ALUSrc	Branch	MemWrite	MemtoReg	ALUOp
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

Table 2: ALU Decoder

ALUOp	Funct	ALUControl
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 (set less than)

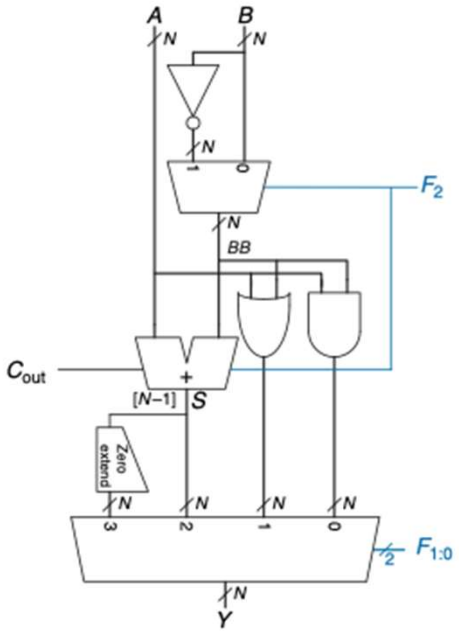
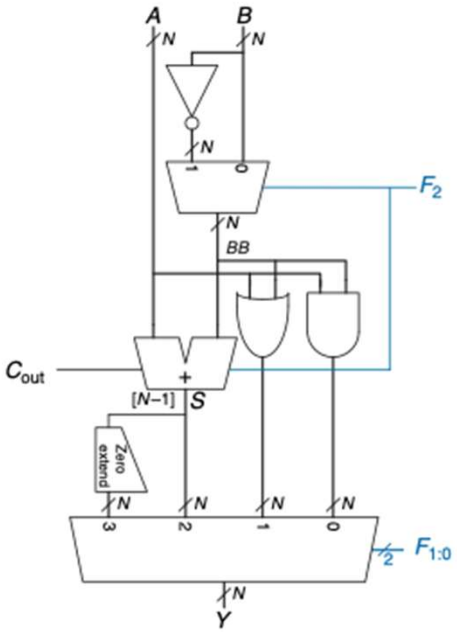


Table 5.1 ALU operations

$F_{2:0}$	Function
000	A AND B
001	A OR B
010	A + B
011	not used
100	A AND $\overline{B}$
101	A OR $\overline{B}$
110	A – B
111	SLT

ALU Modify

Earlier ALU



F[2]

Table 5.1 ALU operations

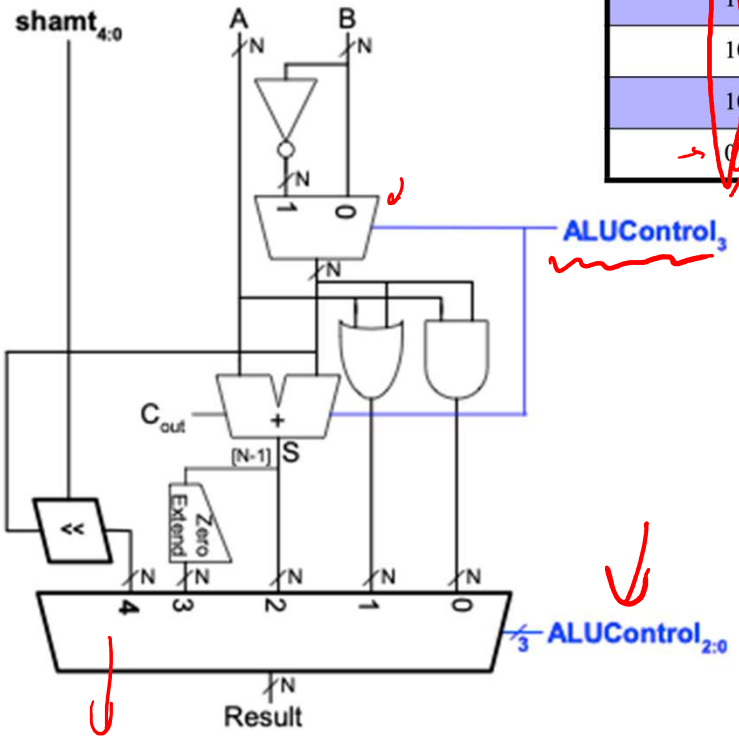
$F_{2:0}$	Function
000	A AND B
001	A OR B
010	A + B
011	not used
100	A AND $\overline{B}$
101	A OR $\overline{B}$
110	A – B
111	SLT

[3]

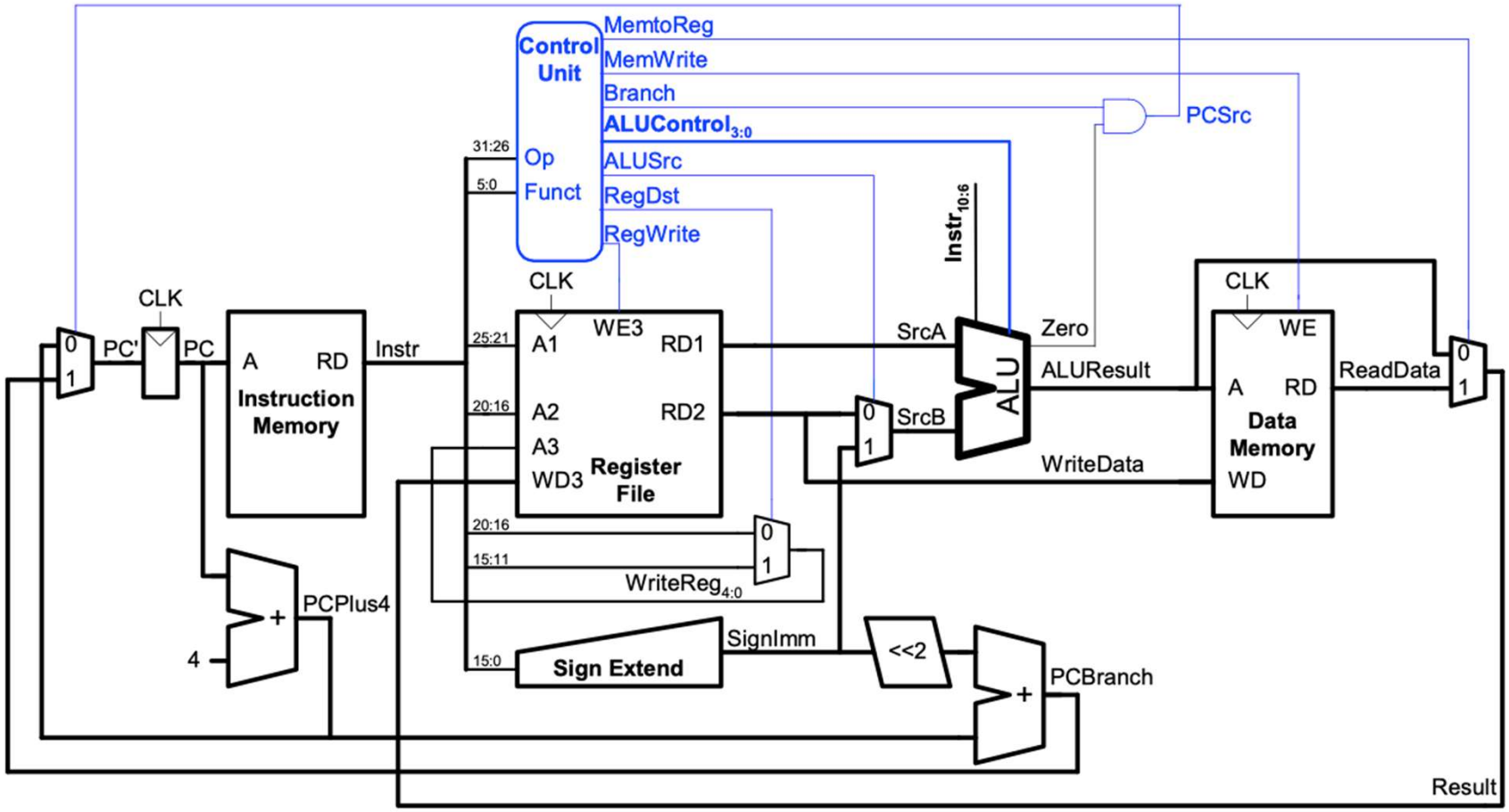
$ALUControl_{3:0}$	Function
0000	A AND B
0001	A OR B
0010	A + B
0011	not used
1000	A AND $\overline{B}$
1001	A OR $\overline{B}$
1010	A - B
1011	SLT
1100	SLL

4

New ALU



Modified Datapath



ALU Decoder Update

Original Decoder

ALUOp	Funct	ALUControl
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 (set less than)

New Decoder

ALUOp	Funct	ALUControl
00	X	0010 (add)
X1	X	1010 (subtract)
1X	100000 (add)	0010 (add)
1X	100010 (sub)	1010 (subtract)
1X	100100 (and)	0000 (and)
1X	100101 (or)	0001 (or)
1X	101010 (slt)	1011 (set less than)
1X	000000 (sll)	0100 (shift left logical)