E85: Digital Design and Computer Engineering
Problem Set 10

Hint: many students in the past have found it helpful to review the textbook for this problem set, including Section 7.4 about the multicycle processor, Section 7.5 about the pipelined processor, Appendix B about the instruction set, and Section 5.5.5 about Register files.

1) Modify the multicycle-cycle RISC-V processor to implement the `lui` instruction. Mark up copies of the controller, main decoder FSM, ALU decoder, Instruction Decoder, and datapath (attached) to handle the new instruction as simply as possible. Name any control signals you need to add.

2) Goliath Corp claims to have a patent on a three-ported register file. Rather than fighting Goliath in Court, you offer to design a new register file with only two ports. Port 1 can be read or written, and port 2 is read-only. Redesign the multicycle datapath and controller (attached) to use your new register file.

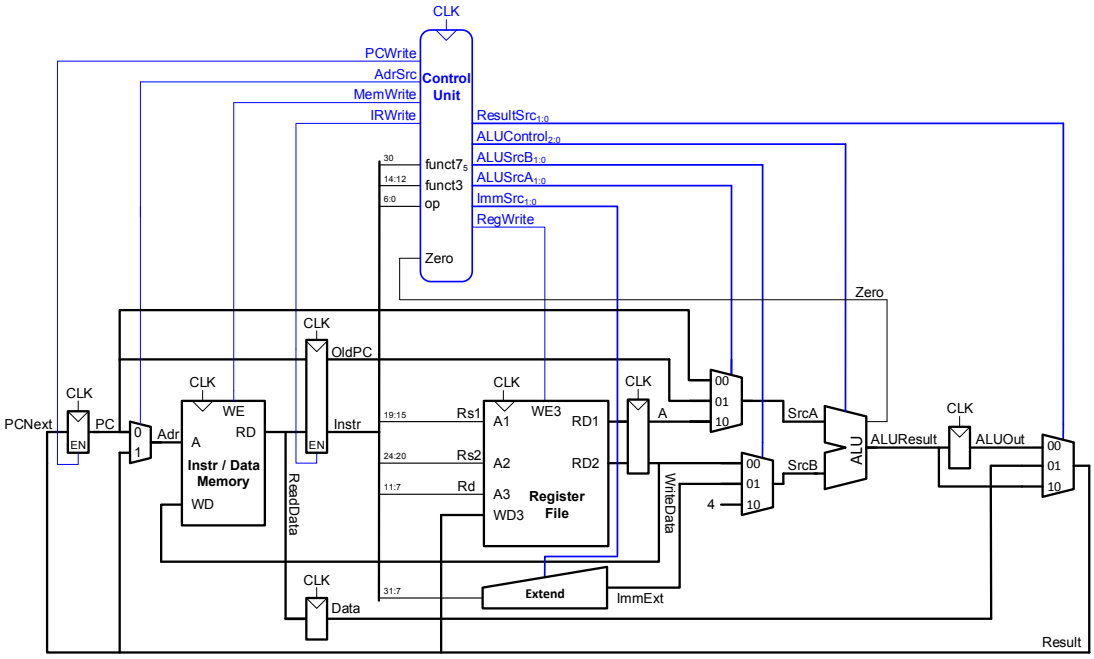
3) How long would your processor from the previous question take to execute the 100-billion instruction program from Example 7.8? Assume your new register file has 30% lower clk-to-Q and setup time because it has fewer ports but ordinary registers and other blocks have delay unchanged.

4) The pipelined RISC-V processor is running the following code snippet. Which registers are being written, and which are being read on the fifth cycle? Recall that the pipelined processor has a hazard unit.
```
addi s1, zero, 42
addi s0, s1, -5
lw   s3, 7(s0)
sw   s4, 66(s1)
or   s2, s0, s3
```

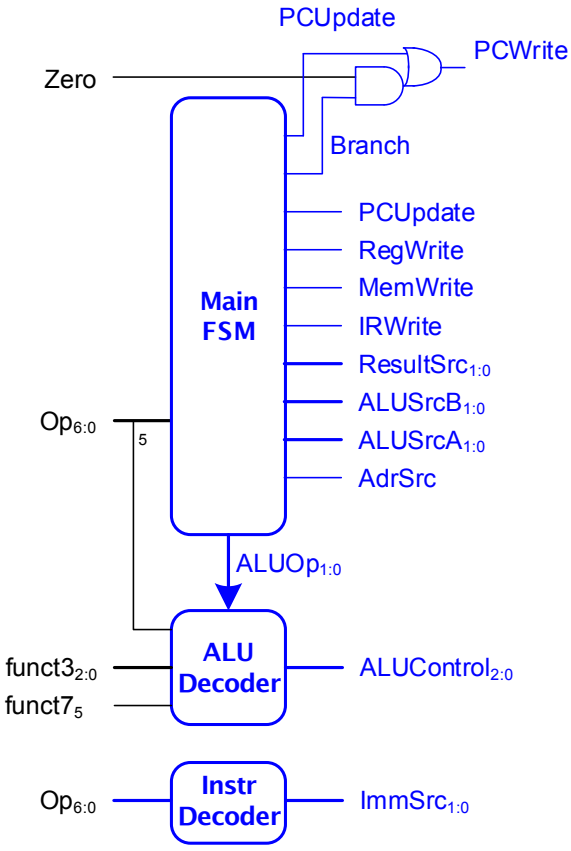
5) Using a diagram similar to Figure 7.66, show the forwarding and stalls needed to execute the following instructions on the pipelined RISC-V.
```
add s0, s4, s5
sub s0, s0, s2
lw  s1, 60(s0)
and s2, s1, s0
```

6) Impact on Society: Write a thoughtful paragraph describing a concrete example of how the skills you have developed in E85 may be applicable to another field of engineering in which you are interested in practicing.

How long did you spend on this problem set? This will not count toward your grade but will help calibrate the workload.



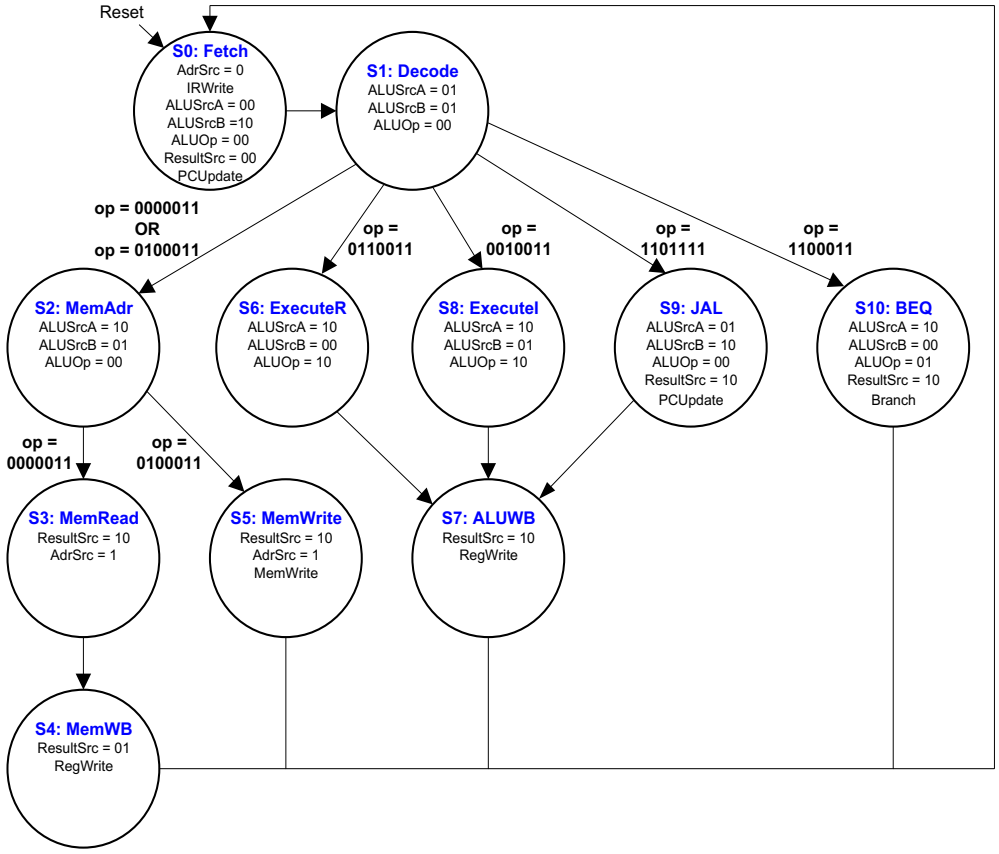
Problem 1: Multicycle Datapath



Problem 1: Multicycle Controller

ALUOp	op_5	$funct7_5$	funct3	Instruction	$ALUControl_{2:0}$
00	X	X	X	lw, sw	010 (add)
01	X	X	X	beq	110 (subtract)
10	00, 01, 10		000	add	010 (add)
	1	1	000	sub	110 (subtract)
	X	0	010	slt	111 (set less than)
	X	0	110	or	001 (or)
	X	0	111	and	000 (and)

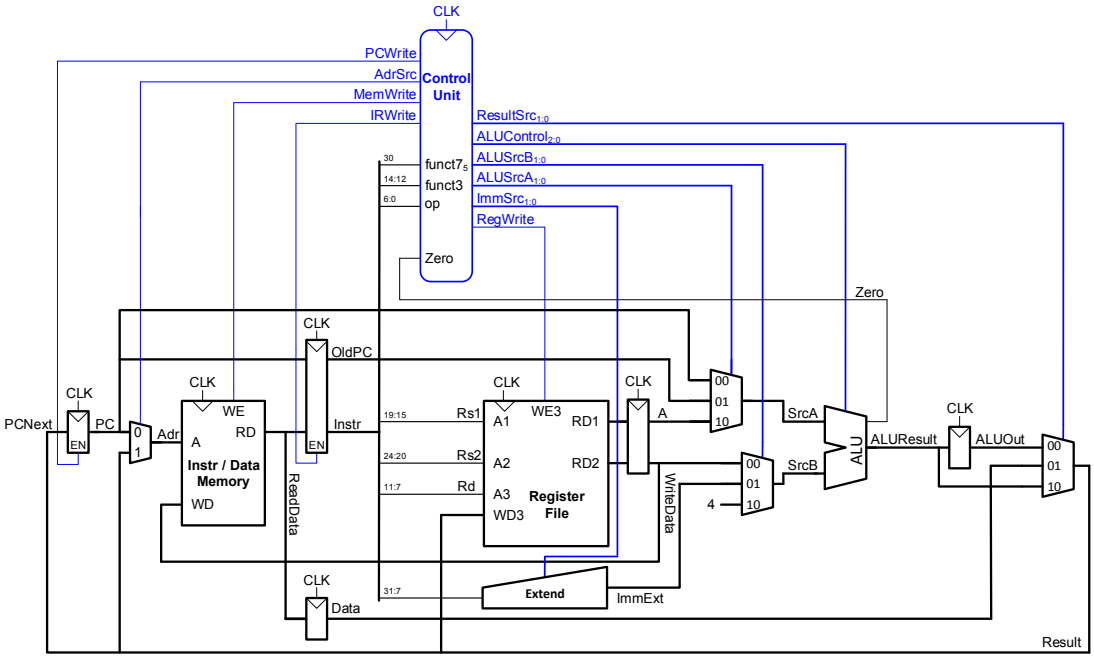
Problem 1: ALU Decoder



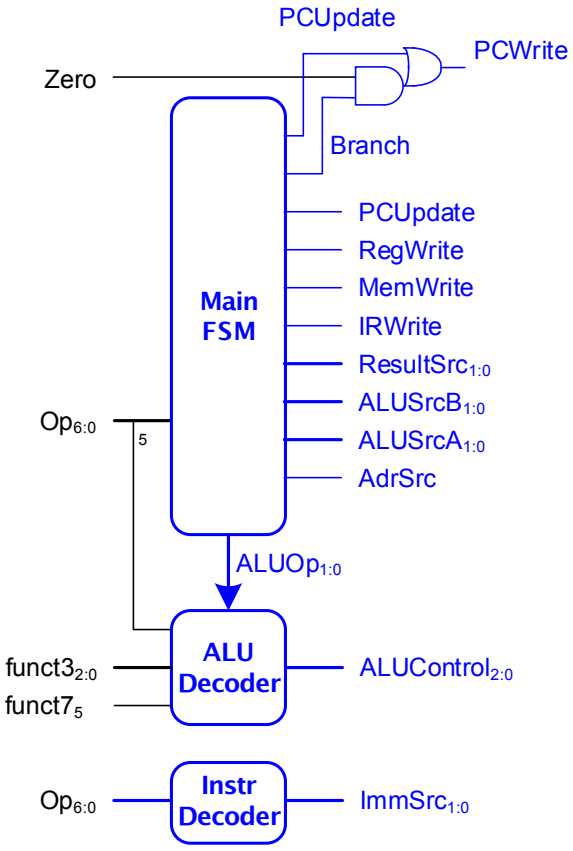
Problem 1: Multicycle Main Control FSM

Instruction	Opcode	ImmSrc	ImmExt	Description
lw	0000011	00	{{20{*Instr*[31]}}, *Instr*[31:20]}	12-bit signed immediate for I-type instructions
sw	0100011	01	{{20{*Instr*[31]}}, *Instr*[31:25], *Instr*[11:7]}	12-bit signed immediate for S-type instructions
beq	1100011	10	{{20{*Instr*[31]}}, *Instr*[7], *Instr*[30:25], *Instr*[11:8], 1'b0}	13-bit signed immediate for B-type instructions
jal	0111111	11	{{12{*Instr*[31]}}, *Instr*[19:12], *Instr*[20], *Instr*[30:21], 1'b0}	21-bit signed immediate for J-type instructions

Problem 1: Instruction Decoder



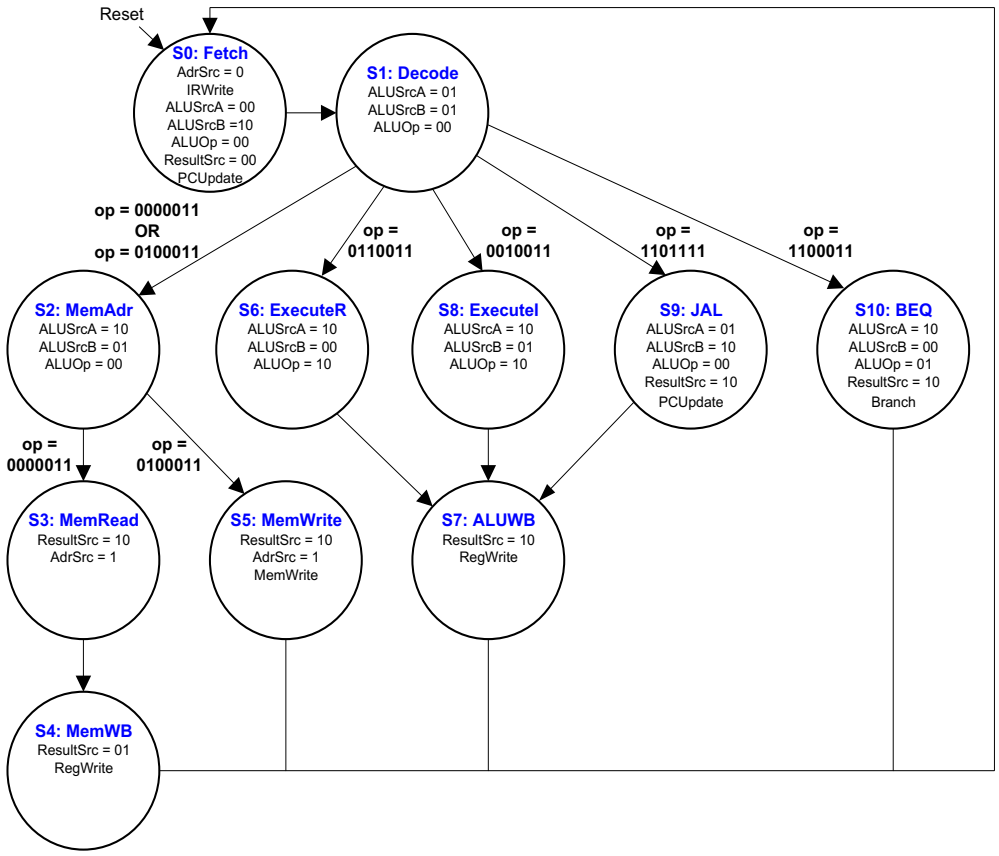
Problem 2: Multicycle Datapath



Problem 2: Multicycle Controller

ALUOp	op_5	$funct7_5$	funct3	Instruction	$ALUControl_{2:0}$
00	X	X	X	lw, sw	010 (add)
01	X	X	X	beq	110 (subtract)
10	00, 01, 10		000	add	010 (add)
	1	1	000	sub	110 (subtract)
	X	0	010	slt	111 (set less than)
	X	0	110	or	001 (or)
	X	0	111	and	000 (and)

Problem 2: ALU Decoder



Problem 2: Multicycle Main Control FSM

Instruction	Opcode	ImmSrc	ImmExt	Description
lw	0000011	00	{{20{*Instr*[31]}}, *Instr*[31:20]}	12-bit signed immediate for I-type instructions
sw	0100011	01	{{20{*Instr*[31]}}, *Instr*[31:25], *Instr*[11:7]}	12-bit signed immediate for S-type instructions
beq	1100011	10	{{20{*Instr*[31]}}, *Instr*[7], *Instr*[30:25], *Instr*[11:8], 1'b0}	13-bit signed immediate for B-type instructions
jal	0111111	11	{{12{*Instr*[31]}}, *Instr*[19:12], *Instr*[20], *Instr*[30:21], 1'b0}	21-bit signed immediate for J-type instructions

Problem 2: Instruction Decoder