Name: Mohammad Elkholy
ID:900202159

Q1)

Problem 1:

Translate the following RISC-V function F1 into machine code both in binary and hexadecimal:

```
F1:   addi x2, x2, -4
      sw x18, 0(x2)
      lui x18, 27
      ori x18, x18, 21
      blt x10, x18, L1
      add x10, x10, x11
      beq x0, x0, Ex
L1:   sub x10, x10, x11
Ex:   lw x18, 0(x2)
      addi x2, x2, 4
      jalr x0, 0(x1)
```

If a caller uses the instruction:
`jal x1, F1`
residing at address 4416 to call the F1 function residing at address 6200 and if the values of x2, x10, x11, and x18 were 8000, 2700, 300, and -1200 respectively at call time, what would be the final values of x0, x1, x2, 10, x11, and x18 at return time? Also what will be the value of x18 immediately after the execution of the ori instruction?

Inst	Format	Instruction Fields (Binary)						HexaDecimal
addi	I	Imm[11:0]		Rs1	funct3	Rd	Opcode	0xFFC10113
		1111 1111 1100		00010	000	00010	0010011	
sw	S	imm[11:5]	Rs2	Rs1	funct3	imm[4:0]	Opcode	0x01212023
		0000000	10010	00010	010	00000	0100011	
lui	U	imm[31:12]				Rd	Opcode	**0x0001B937**
		000 0000 0000 0001 1011				10010	0110111	
ori	I	Imm[11:0]		Rs1	funct3	Rd	Opcode	0x01596913
		0000 0001 0101		10010	110	10010	0010011	
blt	SB	imm[12|10:5]	Rs2	Rs1	funct3	imm[4:1|11]	Opcode	0x01254663
		0000000	10010	01010	100	01100	1100011	
add	R	funct7	Rs2	Rs1	funct3	rd	Opcode	0x00B50533
		000000	01011	01010	000	01010	0110011	

beq	SB	imm[12|10:5]	Rs2	Rs1	funct3	imm[4:1|11]	Opcode	0x00000463
		0000000	00000	00000	000	01000	1100011	
sub	R	funct7	Rs2	Rs1	funct3	rd	Opcode	0x40B50533
		0100000	01011	01010	000	01010	0110011	
lw	I	Imm[11:0]		Rs1	funct3	Rd	Opcode	0x00012903
		0000 0000 0000		00010	010	10010	0000011	
addi	I	Imm[11:0]		Rs1	funct3	Rd	Opcode	0x00410113
		0000 0000 0100		00010	000	00010	0010011	
jalr	I	Imm[11:0]		Rs1	funct3	Rd	Opcode	0x00008067
		0000 0000 0000		00001	000	00000	1100111	

Part 2:
Steps:

1. Jal x1, F1: x1 = 4416+4 = 4420
2. x2 -=4 -> x2 = 7996
3. Store -1200 at memory address 7996
4. x18 = 27 << 12 = 110592
5. ori x18 with 21 -> x18 = 110613
6. x10 = 2700 < x18 -> branch to L1
7. x10 -= x11 -> 2700-300 = 2400
8. Load Memory[7996] into x18 -> x18 = -1200
9. x2+= 4 -> x2 = 8000
10. Jalr x0, 0(x1) sets PC to x1+4, PC = 4424

Assuming return time is before function returns immediately

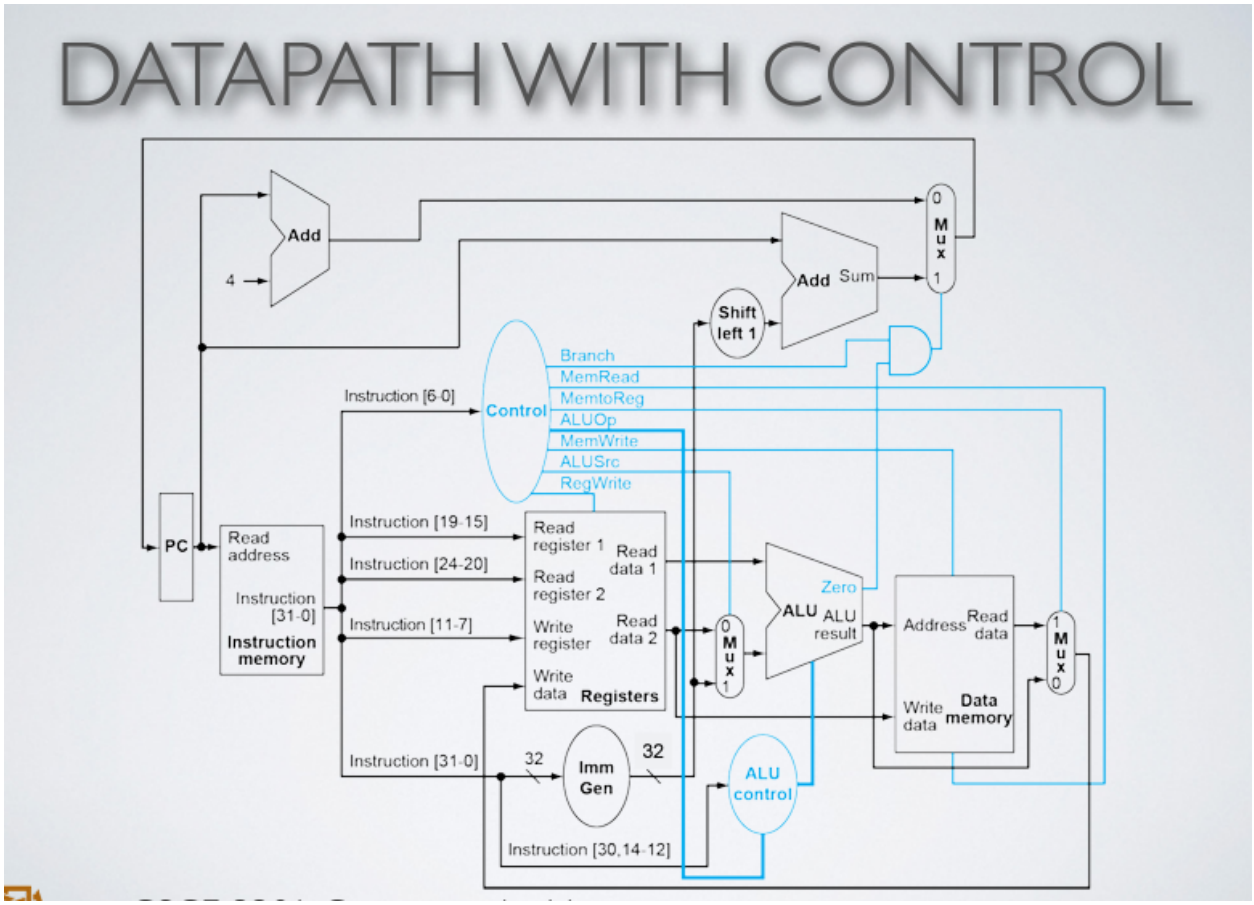
Register	Before	After
x0	0	0
x1	N/A	4420
x2	8000	8000
x10	2700	2400
x11	300	300
x18	-1200	-1200

Q2)

Problem 2:

Calculate the maximum clock frequency for the single cycle RISC-V processor you studied given that:

- Any MUX delay: 5 ns
- Any gate delay: 2.5 ns
- Decoding (control unit and ALU Control unit) delay: 5 ns (2-level logic)
- Adder delay: 11 ns
- ALU delay: 16 ns
- Memory read delay: 21 ns
- Data must be present @ the memory input 0.5 ns before the clock edge
- Register File (RF) read delay: 3 ns
- Data must be present @ the RF input 1 ns before the clock edge
- Immediate Generator delay: 0.4 ns
- Shift left 1 delay: 0.25 ns
- PC t_{su} and t_{CQ} = 0.75 ns



Data Memory read > adder delay + shift left + gate delay so we pick memory read and don't count those.

delay = PC_t_cq + Instruction memory read + Setup time for Rf + max(RegisterFile read, max(immGen, Control)+MUX, Control+ALUControl) + ALU + Data memory read + MUX

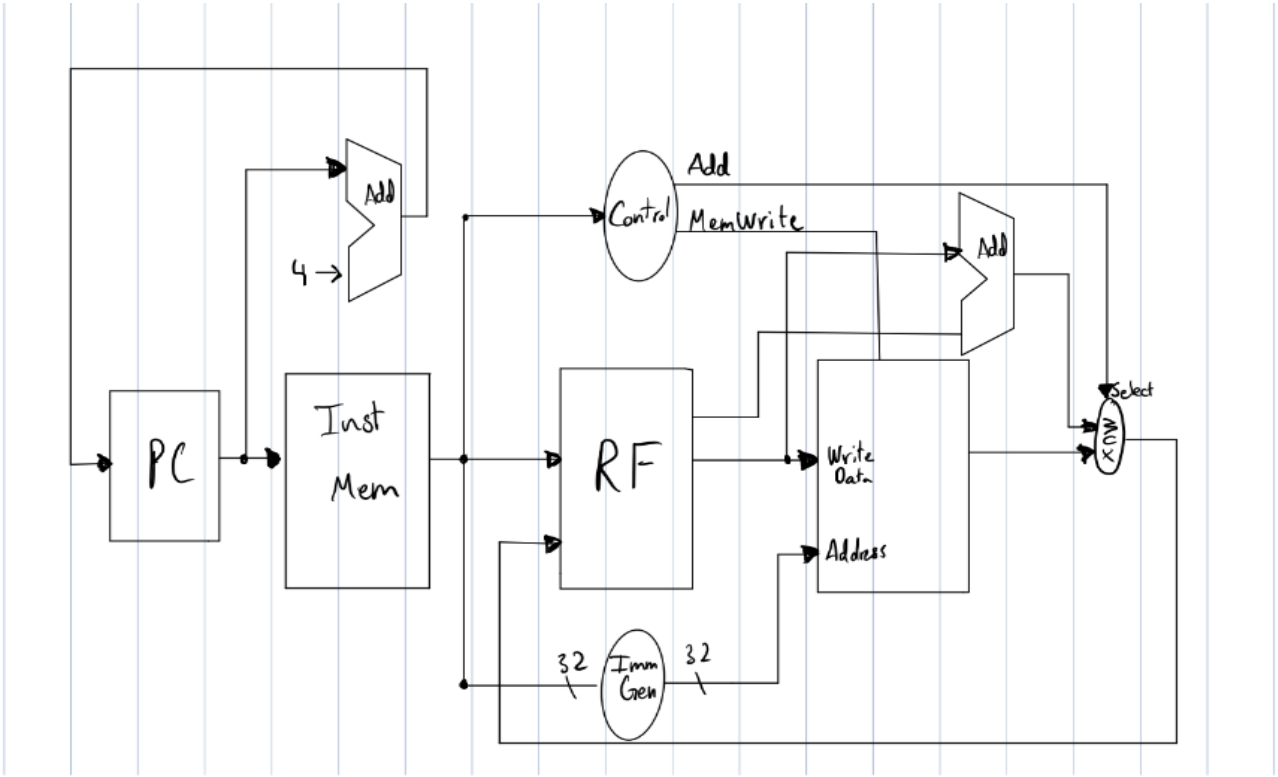
Delay = 0.75 + 21 + 1 + max(3, max(0.4, 5)+5, 2 * 5) + 16 + 21 + 5
= 0.75 + 21 + 1 + 10 + 16 + 21 + 5 = 74.75 ns

Max Freq = 1/74.75 = 13.38 MHz

Problem 3:

Problem 3:

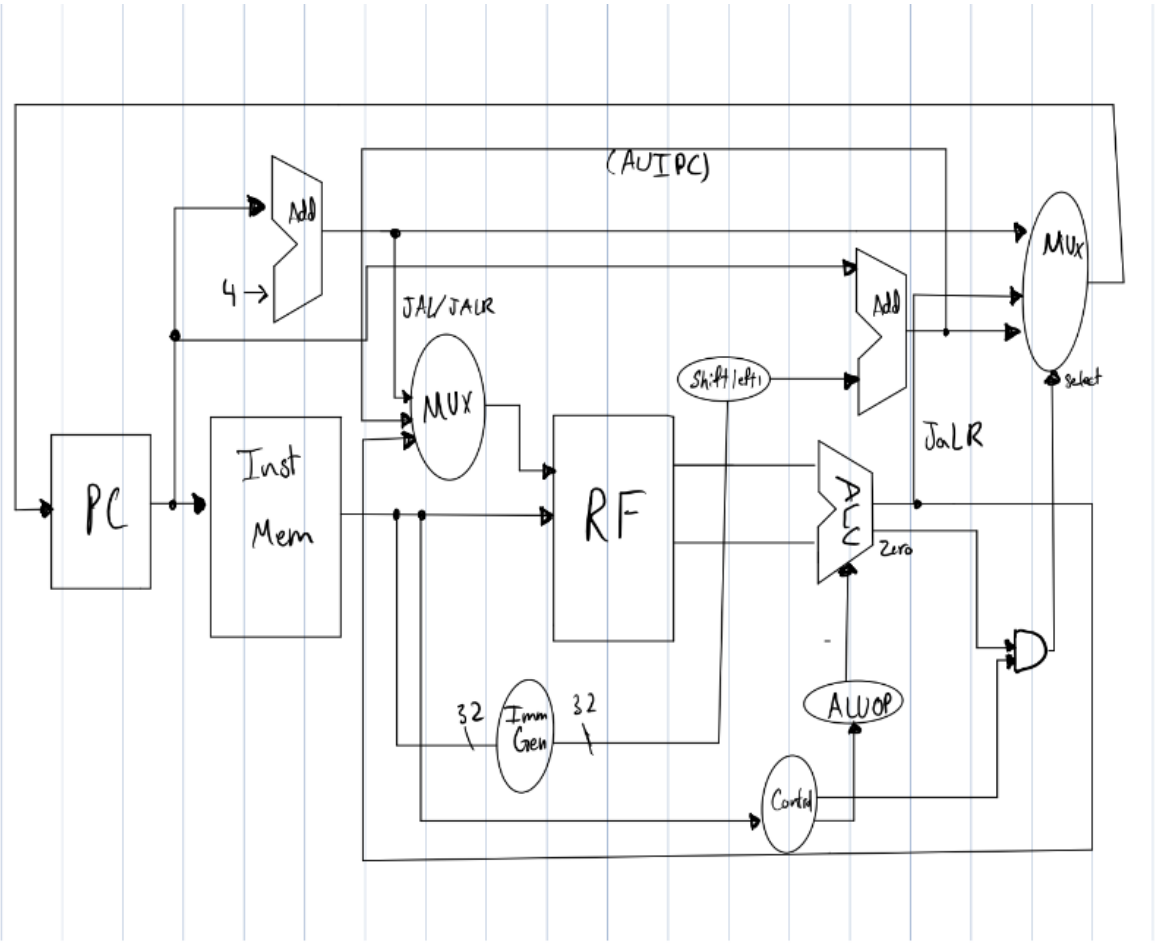
Draw a simplified datapath for the single cycle RISC-V processor that supports the ADD and SW instructions only. Using the delays given in problem 2, what would be the maximum clock frequency in this case?



Max delay = Pc_t_cq + Inst mem + setup for rf + max(Immgen + Memory , RF + Memory, RF + Adder, control) + mux
= 0.75 + 21 + 1 + max(0.4+21, 3+21, 3+11, 5) +5
= 0.75 + 21 + 1 + 24 + 5 = 51.75 ns
Max Freq = 1/51.75 = 19.32 MHz

Problem 4:
I used those to wire the different parts.

JALR: RF[rd] = PC + 4, PC = rs1 + Imm
JAL: RF[rd] = PC + 4; PC = PC + Imm * 2
LUI: RF[rd] = (Imm[32: 12] << 12)
AUIPC: RF[rd] = PC + (Imm[32: 12] << 12)



Max delay = pc_t_cq + memory+rf setup+rf+alu+gate+mux = 0.75+21+1+3+16+2.5+5 = 49.25
MAx freq = 1/49.25 = 20.30MHz

Problem 5:

Fill the table below with the values of the control signals corresponding to each instruction listed:

	Branch	**MemRead**	**MemtoReg**	**MemWrite**	**ALUSrc**	**RegWrite**	**ALUOp**
add	0	0	0	0	0	1	10
lw	0	1	1	0	1	1	00
sw	0	0	X	1	1	0	00
beq	1	0	X	0	0	0	01