

VE370 Homework 3 Lan Wang 519370910084

1. (30 points) Given RISC-V assembly instruction sequence:

```

bne x22, x23, Else          0x1000F400
add x19, x20, x21          0x1000F404
beq x0, x0, Exit           0x1000F408
Else: lw x19, 0(x20)        0x1000F40C
Exit: ...

```

Assuming the memory location of the first instruction (`bne`) is 0x1000F400, what are the values of the following control signals for each of the instructions?

Ctrl Signals Instruction	Branch	MemRead	MemtoReg	ALUOp	MemWrite	ALUSrc	RegWrite	Zero	ImmGen Output
bne bne	1	0	0	01	0	0	0	1	0x00000004
add	0	0	0	10	0	0	1	0	0x00000000
lw	0	1	1	00	0	1	1	0	0x00000000

2. (10 points) Given following assembly instruction:

and rd, rs1, rs2

(1) Which resources (blocks) perform a useful function for this instruction? (3 points)

(2) Which resources (blocks) produce no output for this instruction? Which resources produce output that is not used? (7 points)

(1) PC, instruction memory, control, registers, ALU control, ALU
 ↳ No output: Data memory, Imm Gen
 ↳ Output not used: The adder that adds pc & 2* immediate.

3. (10 points) Consider the following instruction mix:

R-type	I-type (non-lw)	Load	Store	Branch	Jump
24%	28%	25%	10%	11%	2%

(1) What fraction of all instructions use data memory? (3 points)

(2) What fraction of all instructions use instruction memory? (2 points)

(3) What fraction of all instructions use the sign extend? (5 points)

(1) 25% + 10% = 35%

↳ 100%.

(3) 28% + 25% + 10% + 11% + 2% = 76%

4. (10 points) When silicon chips are fabricated, defects in materials (e.g., silicon) and manufacturing errors can result in defective circuits. A very common defect is for one signal wire to get “broken” and always read a logical 0. This is often called a “stuck-at-0” fault.
- (1) Which instructions fail to operate correctly if the MemToReg wire is stuck at 0? (5 points)
 - (2) Which instructions fail to operate correctly if the ALUSrc wire is stuck at 0? (5 points)

(1) load instructions such as lw
 (2) All the instructions which need immediate

5. (1) What is the latency of an R-type instruction (i.e., how long must the clock period be to ensure that this instruction works correctly)? (5 points)
 (2) What is the latency of lw? (5 points)
 (3) What is the latency of sw? (5 points)
 (4) What is the latency of beq? (5 points)
 (5) What is the latency of an arithmetic, logical, or shift I-type (non-load) instruction? (5 points)
 (6) What is the minimum clock period for this CPU? (5 points)

(1) register read + I-mem + register setup = $30 + 250 = 280 \text{ ps}$

(before register file read data)

register file + mux + ALU + mux + register setup = $150 + 25 + 200 + 25 + 20 = 420 \text{ ps}$

(before register file write data) $\Rightarrow 700 \text{ ps}$

(2) 280 ps (before register file read data)

register file + ALU = $150 + 200 = 350 \text{ ps}$ $\Rightarrow 925 \text{ ps}$

(before read data memory)

D-mem + mux + register setup = $250 + 25 + 20 = 295 \text{ ps}$

(3) 280 ps (before register file read data)

350 ps (before read data memory) $\Rightarrow 905 \text{ ps}$

D-mem + mux = $250 + 25 = 275 \text{ ps}$

(4) 280 ps (before register file read data)

register file + mux + ALU + gate + mux + register setup

$= 150 + 25 + 200 + 5 + 25 + 20 = 425 \text{ ps}$ $\Rightarrow 705 \text{ ps}$

(5) 280 ps (before register file read data)

register file + ALU + mux + register setup = $150 + 200 + 25 + 20 = 395 \text{ ps}$

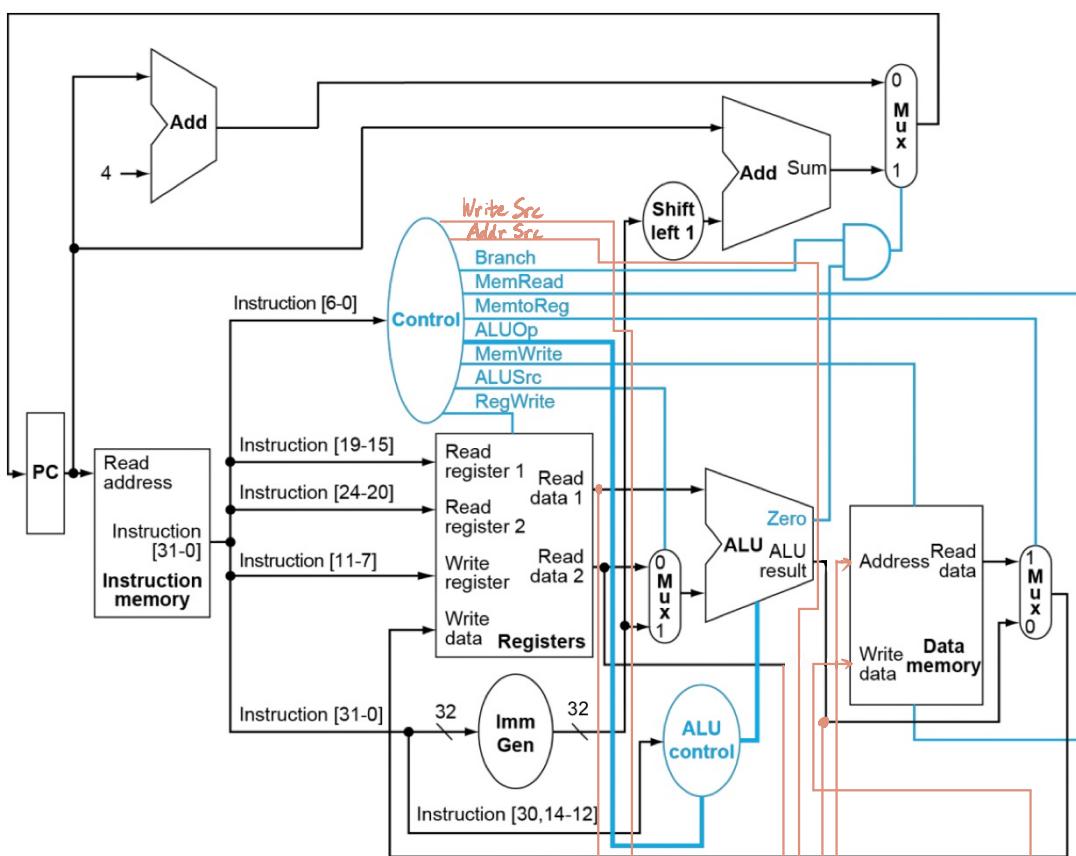
(before register file write data) $\Rightarrow 675 \text{ ps}$

(6) ps

6. (10 points) Modify the single-cycle processor datapath to add a proposed new assembly instruction:

```
ss rs1, rs2, immediate #Store Sum
```

Operation: $\text{Mem}[\text{Reg}[rs1]] = \text{Reg}[rs2] + \text{immediate}$



ss will make $\text{AddrSrc} = 1$
 $\text{WriteSrc} = 1$
 $\text{Branch} = 0$
 $\text{MemRead} = 0$
 $\text{MemtoReg} = 0$
 $\text{ALUOp} = 10$
 $\text{MemWrite} = 1$
 $\text{ALUSrc} = 1$
 $\text{RegWrite} = 0$