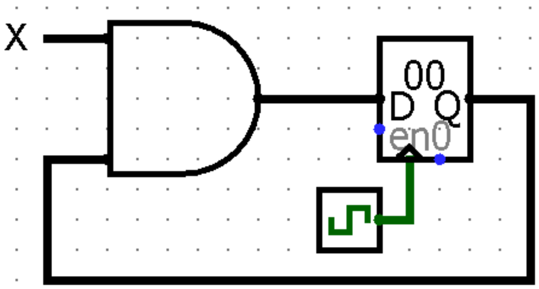
1. Draw the transition state diagram from a FSM that reads a binary string bit-by-bit and outputs whether the total number of 1s seen since the beginning is divisible by 3.

a. 2. For the circuit below, assume that the setup time is 15ns, hold time is 30ns, and the AND gate delay is 10ns. If the clock rate is 10 MHz and x updates 25ns after the rising edge of the clock, what are the minimum and maximum values for the clk-to-Q delay to ensure proper functionality?

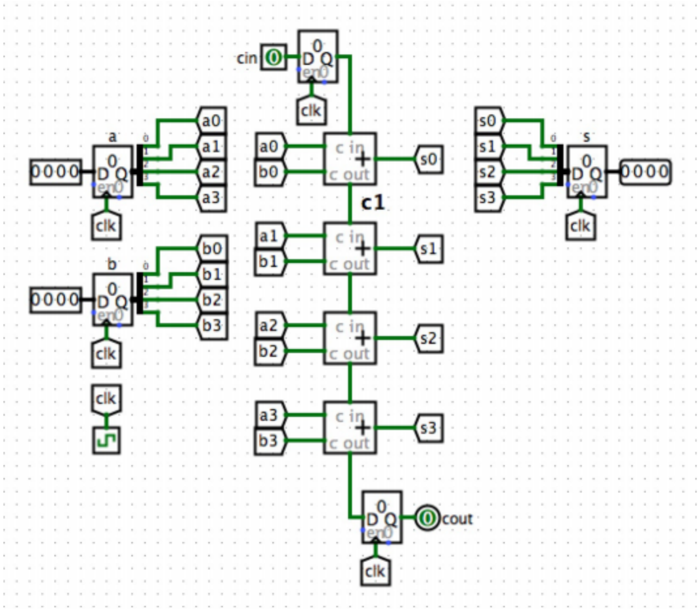


Min: ___________________ns

Max: ___________________ns

- **t_hold <= t_input <= t_clk_period - t_setup**
- **Clq_to_q + AND +setup <= clk_period**

3.



Consider the 4-bit adder shown to the right
It takes:

- a carry in (`cin`)
- two four-bit inputs:
  - *a* with bits `a0, a1, a2, a3`
  - *b* with bits `b0, b1, b2, b3`

Outputs:

- a carry out (`cout`)
- one four-bit output:
  - *s* with bits `s0, s1, s2, s3`

Assume each adder has a delay of 10ns, and any registers have a clk-to-q, hold time, and setup time of 5ns. Assume the inputs are driven by registers, and outputs are registers as well.

1. Write Boolean formulas for `s0` and `c1` in terms of the inputs `cin`, `a0`, and `b0`. You may use XOR as an operator in the Boolean formulas. Each formula should use as few operators as possible.

2. What is the critical path delay of the circuit?

3. What is the maximum clock frequency at which the circuit will function correctly? Please include proper units in your answer.

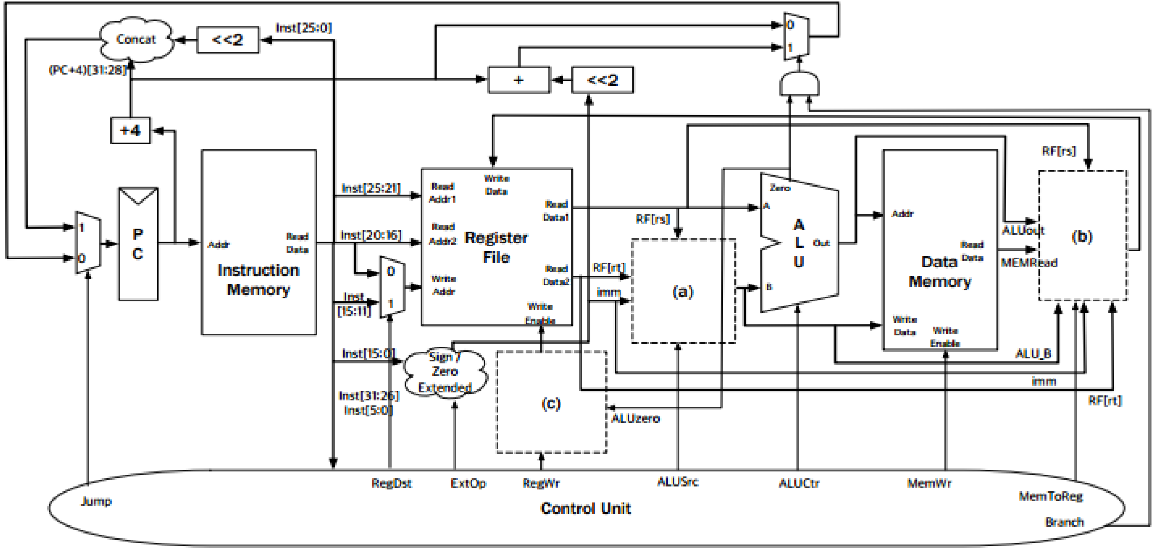
4. What is the maximum hold time the output registers could have at which the circuit would still function correctly? Please include proper units in your answer.

4

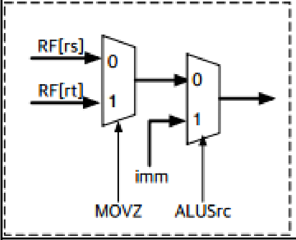
Consider adding the following instruction to MIPS (disregard any existing definitions you may see on the green sheet):

Instruction	Operation
movz rd, rs, rt	if (RF[rs] == 0) RF[rd] ← RF[rt]
movnz rd, rs, rt	if (RF[rs] != 0) RF[rd] ← RF[rt]

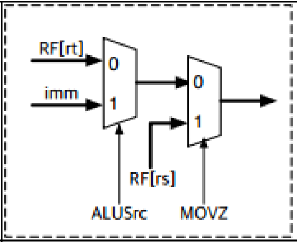
2. Implement **movz** (but not movnz) in the datapath. Choose the correct implementation for (a), (b), and (c). Note that you do not need to use all the signals provided to each box, and the control signal MOVZ is 1 if and only if the instruction is **movz**.



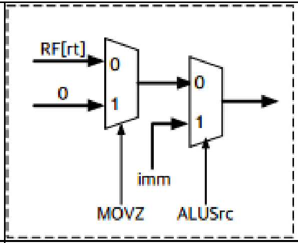
(a)



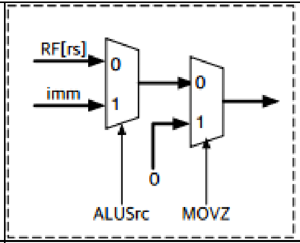
1



2

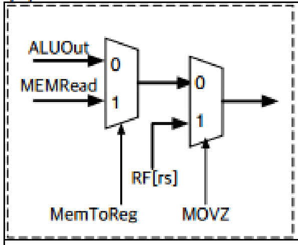


3

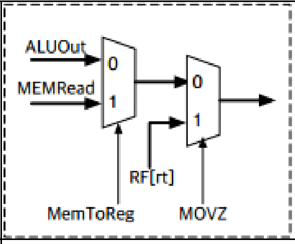


4

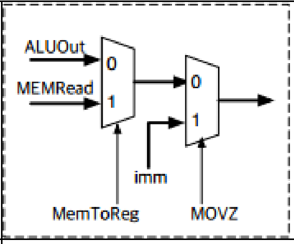
(b)



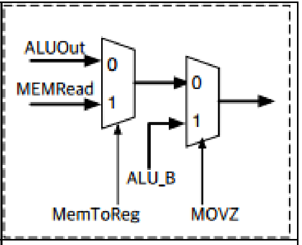
1



2

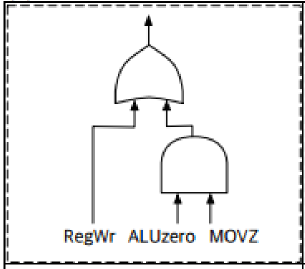


3

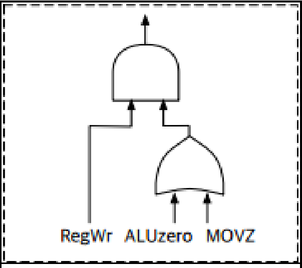


4

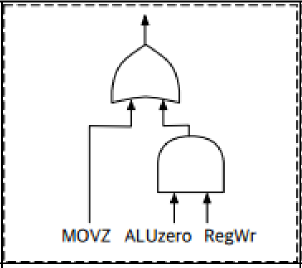
(c)



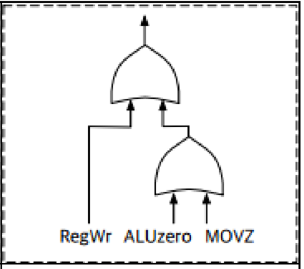
1



2



3



4

3. Generate the control signals for **movz**. The values should be 0, 1, or X (don't care) terms. You must use don't care terms where possible.

MOVZ	RegDst	ExpOp	RegWr	ALUSrc	ALUCtr	MEMWr	MemToReg	Jump	Branch
1									

This table shows the ALUCtr values for each operation of the ALU:

Operation	AND	OR	ADD	SUB	SLT	NOR
ALUCtr	0000	0001	0010	0110	0111	1100