

EECS 151/251 A Midterm Review Problems

Justin Kalloor (he/him)

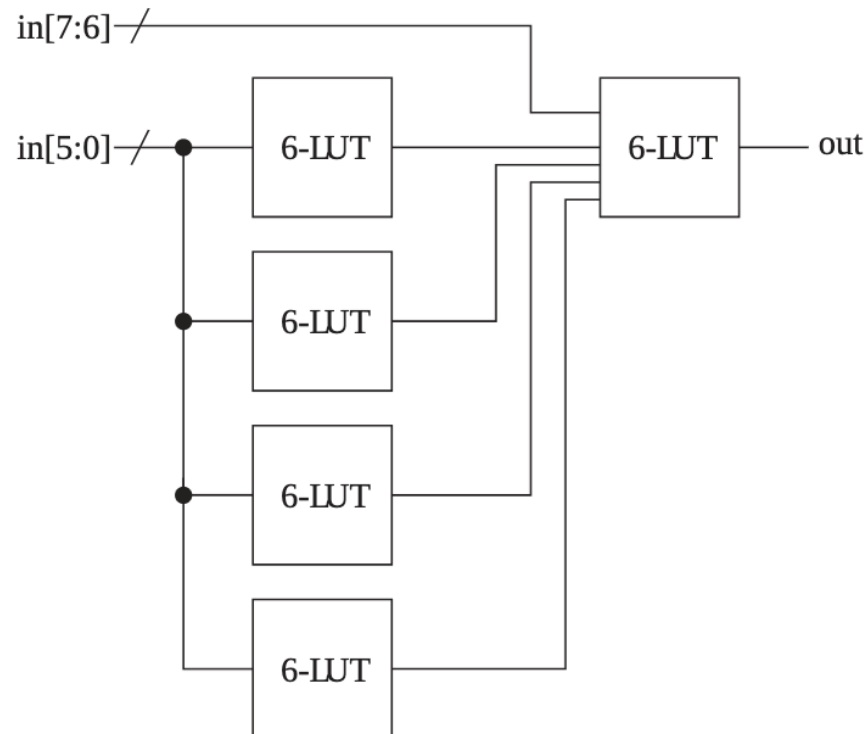
Problem 1 – FPGA LUTS

4 FPGAs [5 pts]

You are given a set of 6-input LUTs and asked to use them (with no other components or logic gates) to implement a 8-input LUT. Show how this can be done using the least number of 6-input LUTs as possible. How many 6-input LUTs does it require?

Solution:

First of all, an 8-input LUT has $2^8 = 256$ configuration bits. Since each 6-input LUT has $2^6 = 64$ configuration bits, we need four 6-input LUTs to store all of them. Each of those four 6-input LUTs takes the first six inputs and selects one bit to output. Then, we can implement a 4-to-1 multiplexer using a 6-input LUT, which selects one of those four outputs depending on the remaining two inputs. This design uses five 6-input LUTs.



Problem 2

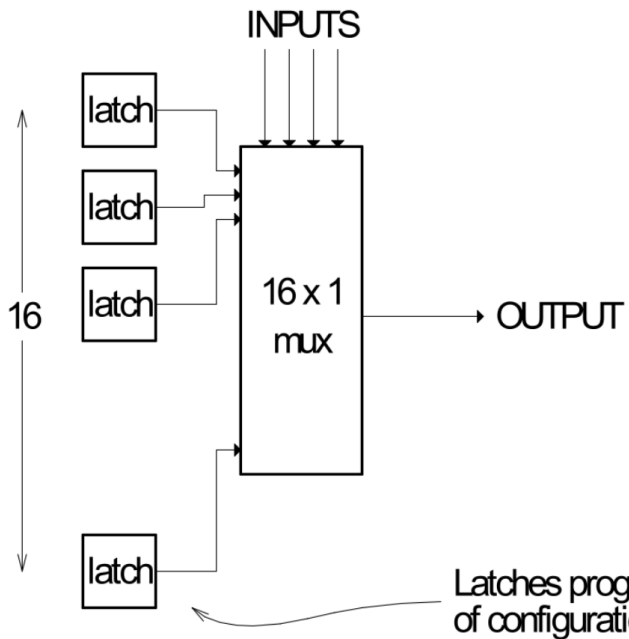
2. FPGAs [15 pts]

- (a) John is a theoretician at heart but works on FPGA architecture research. He comes up with the following two formulas for expressing LUT area and LUT delay as a function of N , the number of LUT inputs, for $N \geq 2$. In the formulas k_a and k_t are layout and process related constants.

$$Area = k_a \cdot 2^N$$

$$Delay = k_t \cdot N$$

Are John's formulas accurate? Explain. What are the constants, k_a and k_t , meant to represent?



Solution:

The formulas are correct.

An N-LUT consists of 2^N configuration bits (latches) and a 2^N -to-1 MUX. A 2^N -to-1 MUX can be constructed from a binary tree of $(2^N - 1)$ 2-to-1 MUXes with a depth of $\log_2(2^N) = N$.

Assume the area of a config latch is a_l , and the area of a 2-to-1 MUX is a_m , and the gate delay of a 2-to-1 MUX is d .

The area of an N-LUT is $Area = a_l \cdot 2^N + a_m \cdot (2^N - 1) \approx (a_l + a_m) \cdot 2^N$ (if N is large enough)

The delay of an N-LUT is $Delay = d \cdot N$ (the longest path delay – which is the propagation delay from the latches to N levels of MUXes to the LUT output)

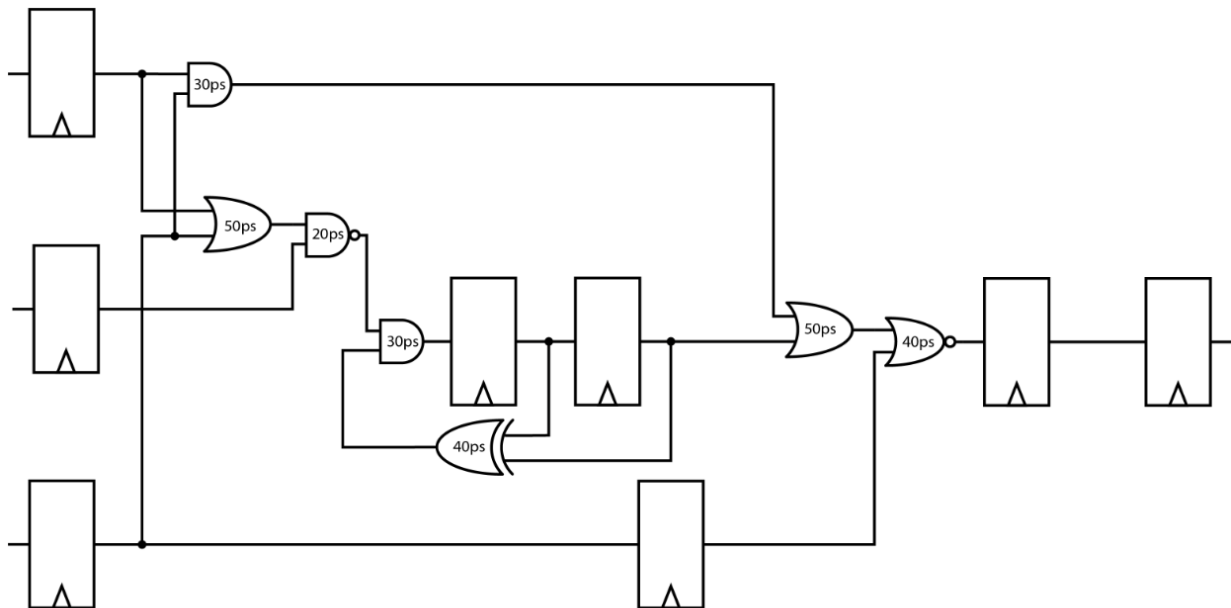
Therefore, the area scales proportionally to 2^N , and k_a represents the area of a config latch plus a 2-to-1 MUX. The delay scales proportionally to N , and k_t represents the gate delay of a 2-to-1 MUX.

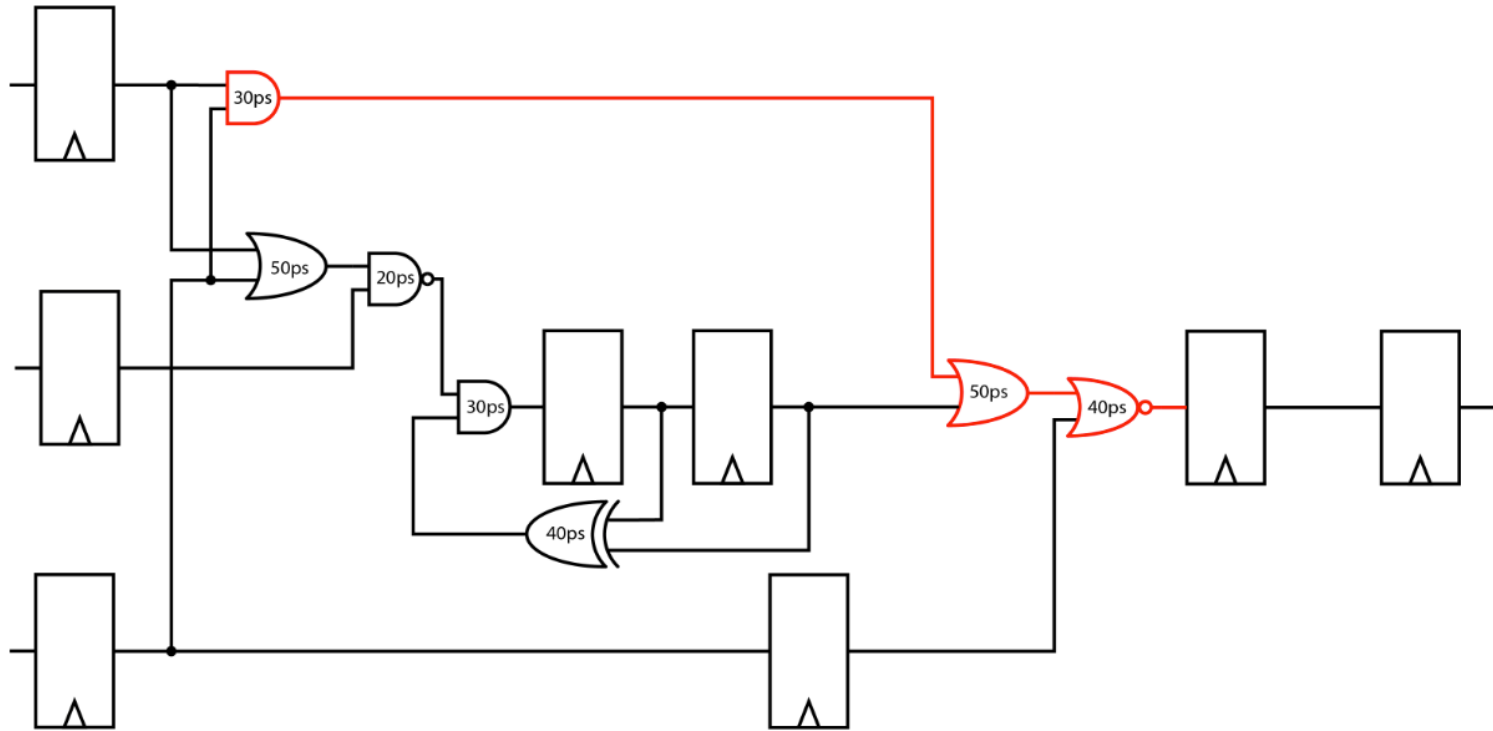
The following figure shows how an N-LUT is typically built

Circuit Timing

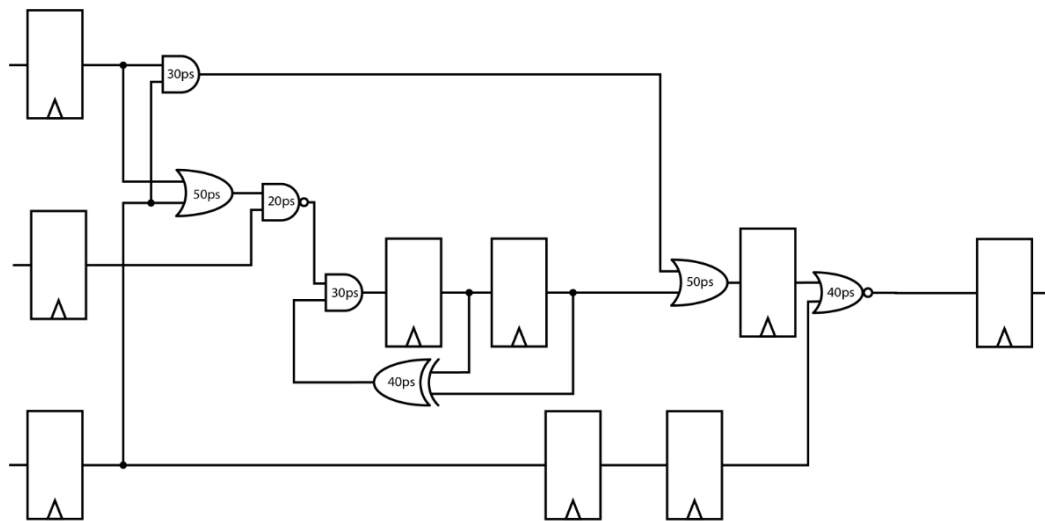
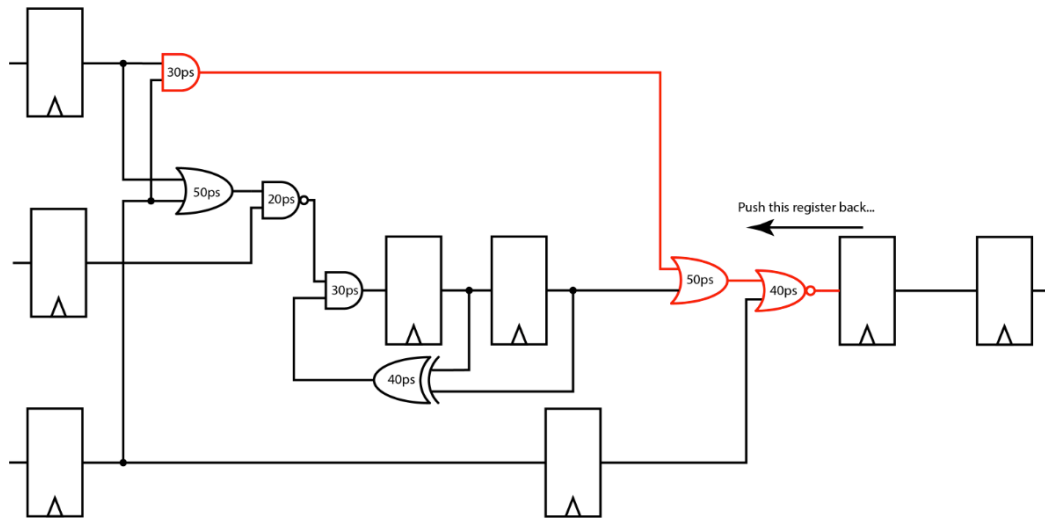
Problem 1: Retiming

Consider the circuit below. What is the critical path delay? Retime the circuit to minimize the critical path delay. What is the maximum clock frequency for the retimed circuit? The flip-flops have $t_{\text{setup}} = 20$ ps, $t_{\text{clk-q}} = 10$ ps, and $t_{\text{hold}} = 5$ ps.

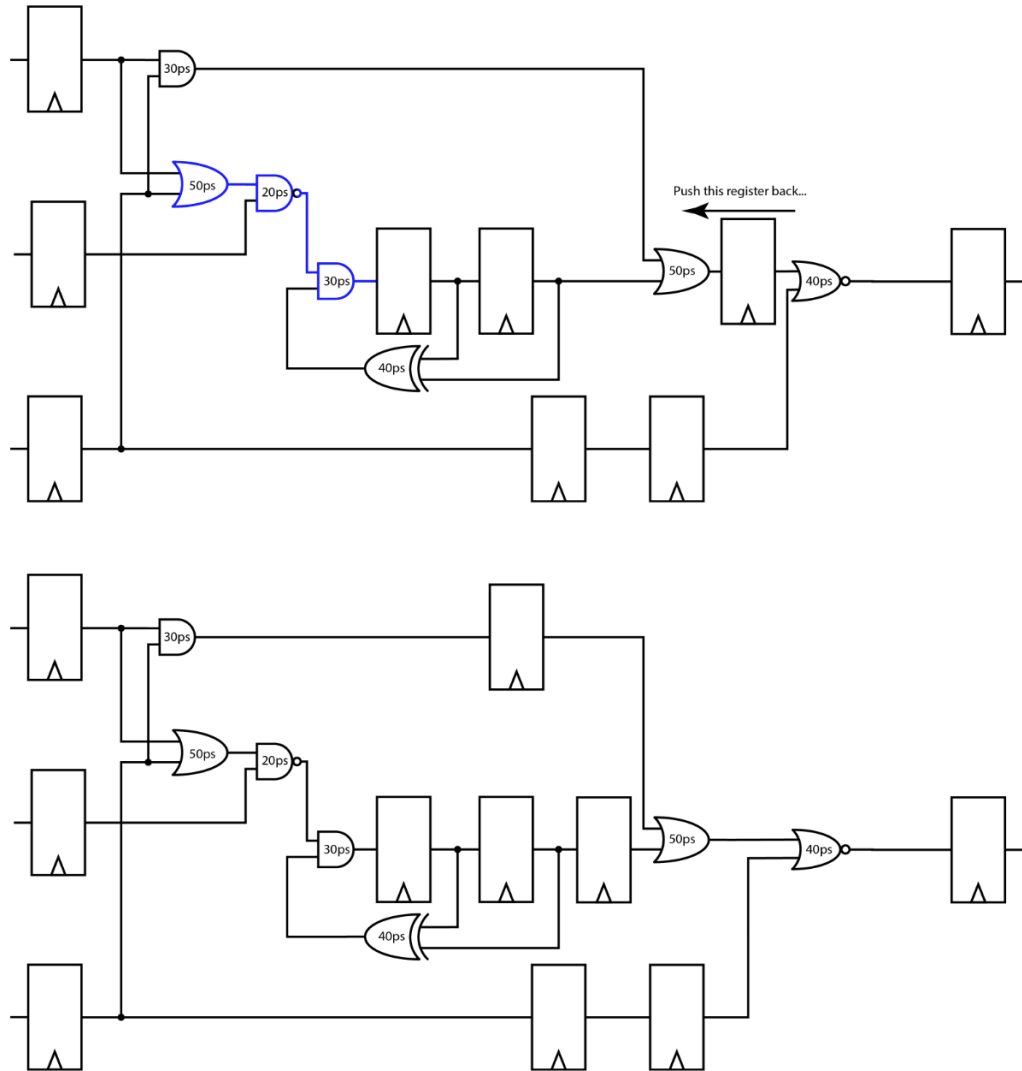




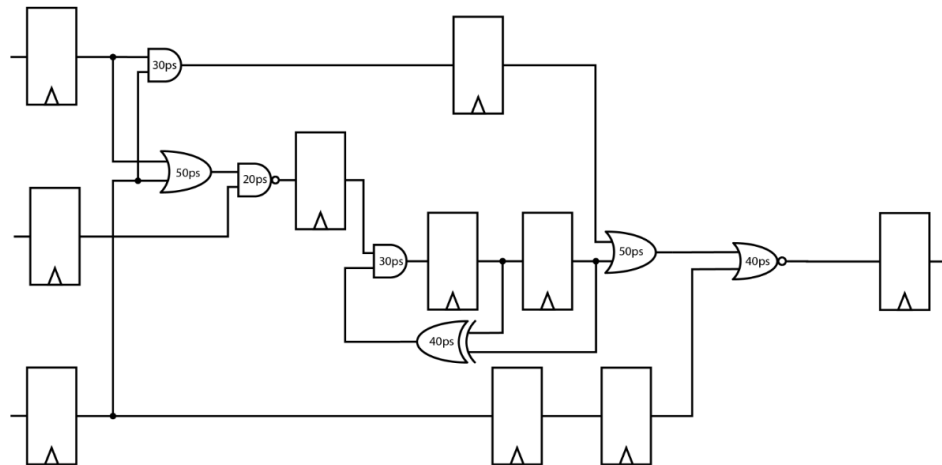
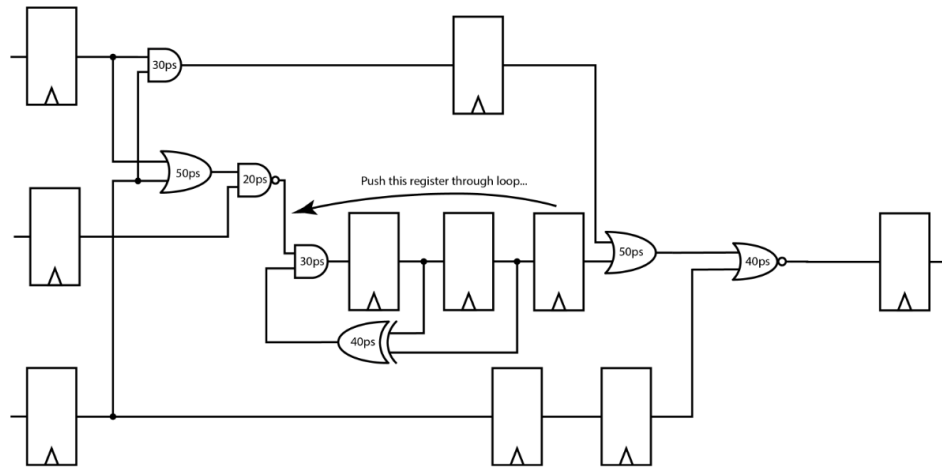
To break this path, we will need to push the right register back through the NOR.



To break this path, we will have to keep pushing the register through the design.



Since there is a loop involved, we will want to keep this loop intact during the retiming, so we treat this loop as a fixed unit and push the register directly to the input to the loop.

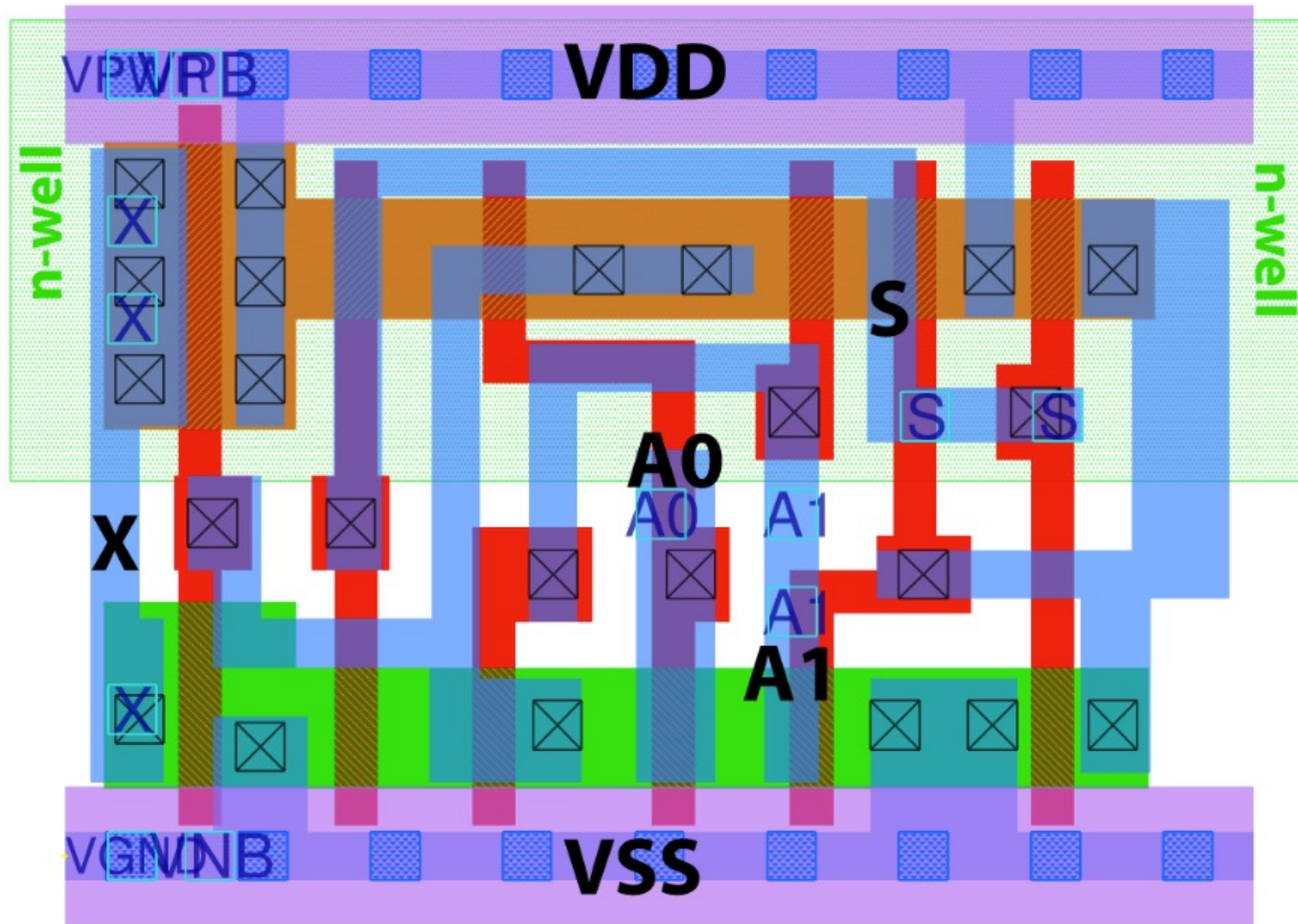


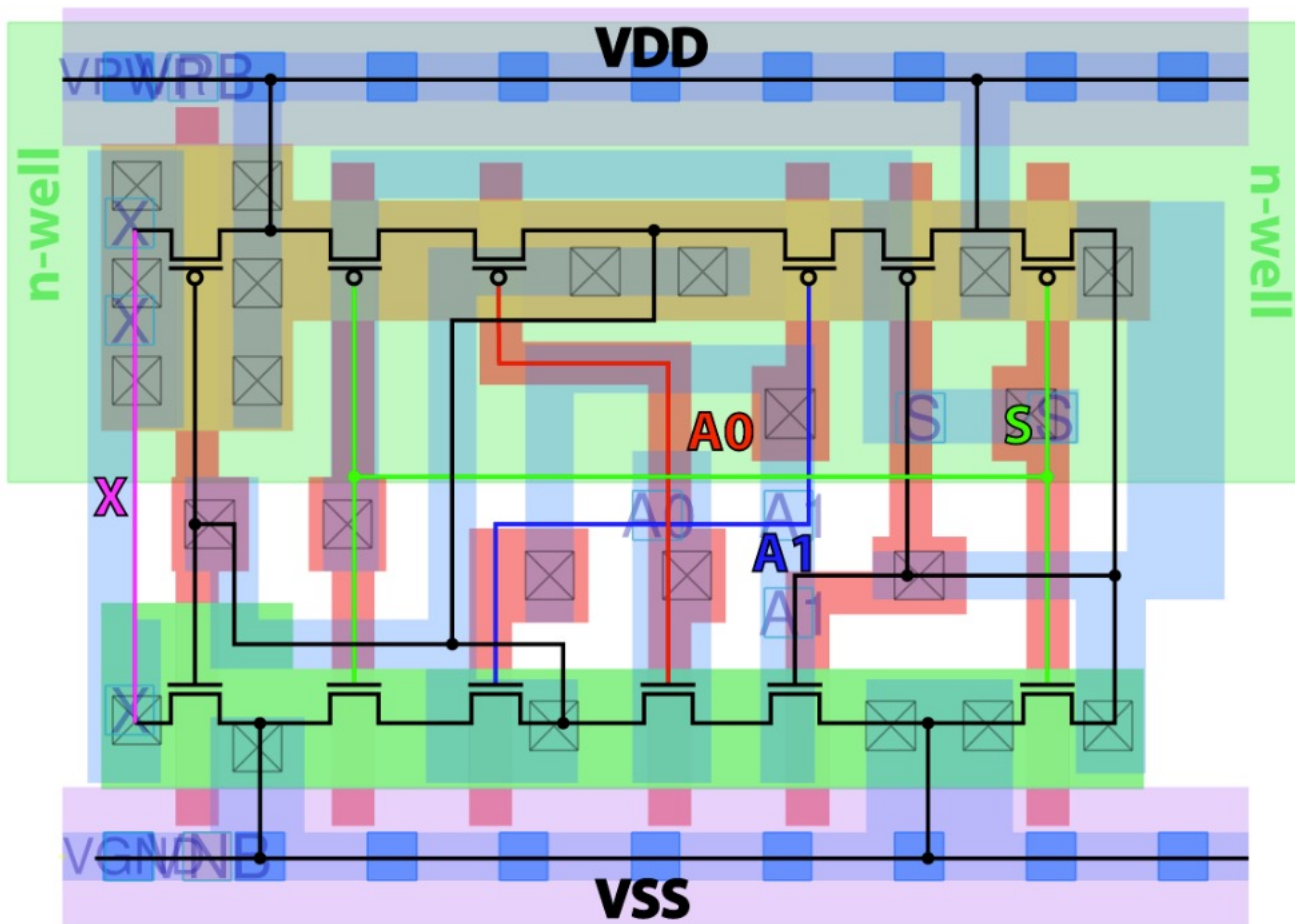
The final critical path of this network is thus 90 ps, and so the minimum clock period we can tolerate is,

$$T_{min} \geq t_{clk-q} + t_{p,crit} + t_{setup} = 10 \text{ ps} + 90 \text{ ps} + 20 \text{ ps} = 120 \text{ ps}$$

which leads to a maximum clock frequency of 8.3 GHz.

Circuit Layout

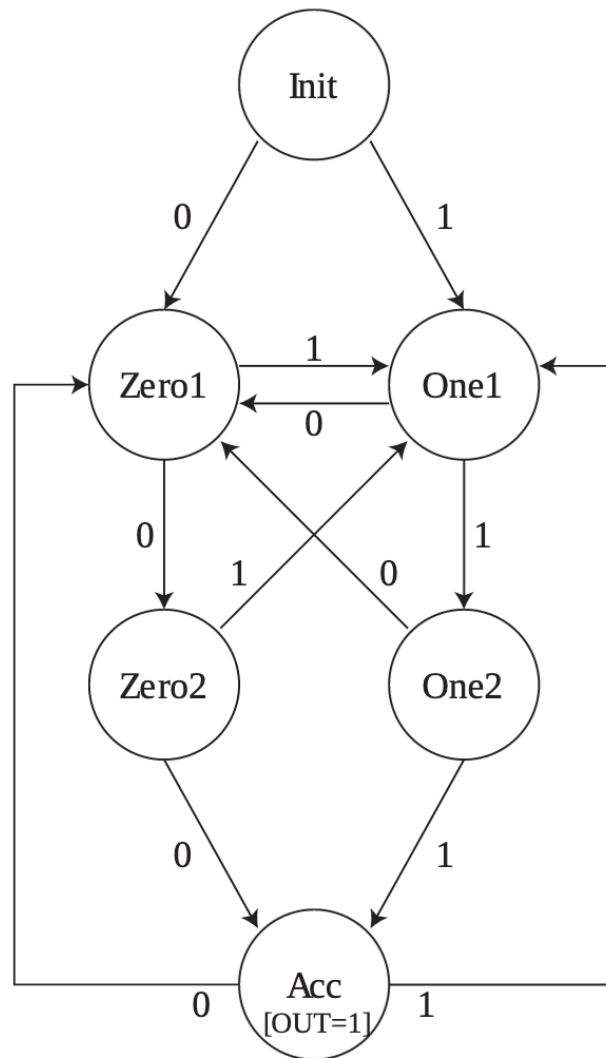




FSMs

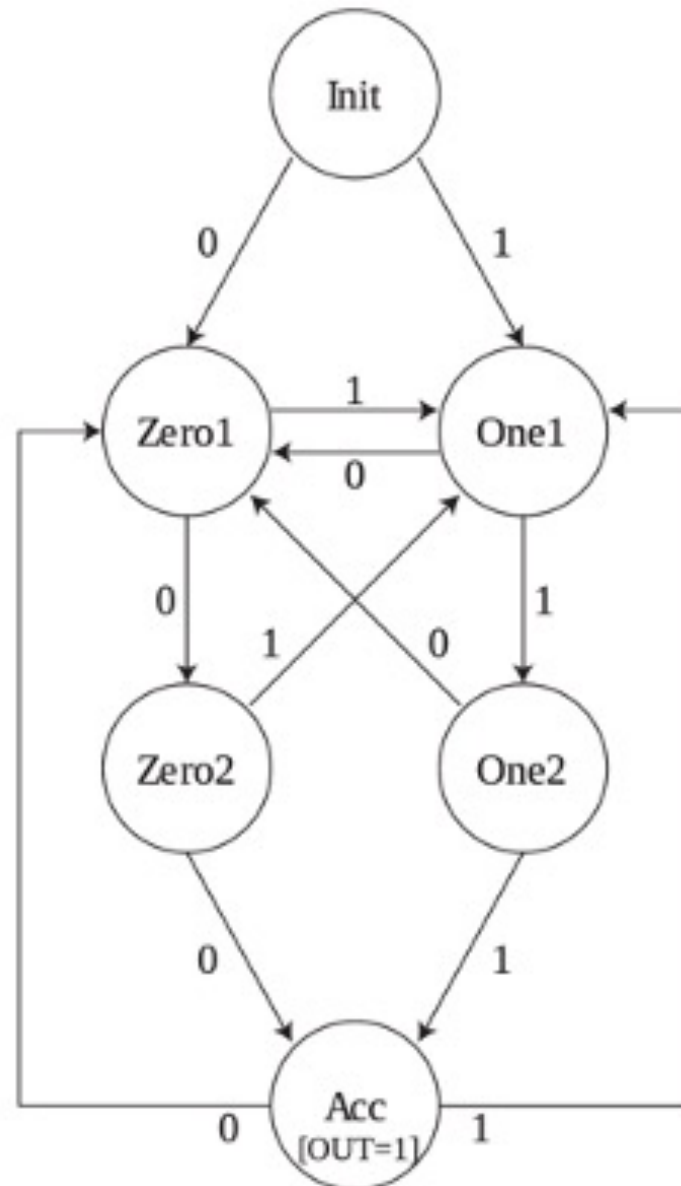
In the space below, neatly draw the state transition diagram for a Moore finite state machine with the following specification. The FSM has a single input, *IN*, and a single output, *OUT*. After reset it outputs a 0. When it sees a consecutive sequence of three 1's or three 0's at its input, it outputs a 1, then starts looking again. (For example, for the input sequence 0111_1111_0001, it outputs a sequence 0000_1001_0001, assuming that it resets at the beginning and outputs a 0 in the first cycle.)

Solution:



Labels on the edges represent the value of IN , and OUT is 0 unless specified.

- Draw a circuit for that FSM



Solution:

$$OUT_1 = A \oplus B \oplus C$$

$$OUT_2 = AB + AC + BC$$

Solution

(a)

a	b	c	d	e
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1

(b)

cycle	0	1	2	3	4
<i>a</i>	0	1	1	0	1
<i>b</i>	1	1	0	1	1
<i>c</i>	0	0	1	0	0
<i>d</i>	1	0	1	1	0
<i>e</i>	0	1	0	0	1