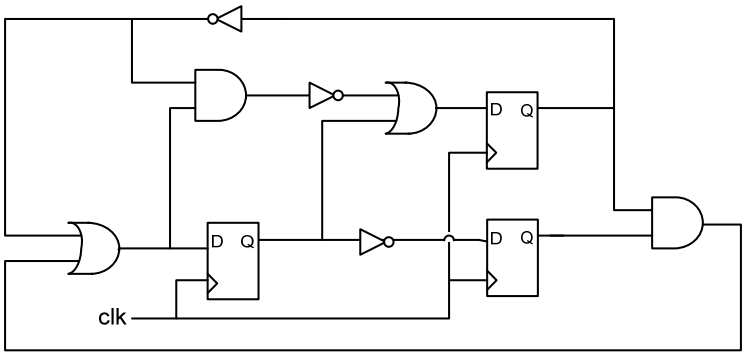
CPEN311 Practice Assignment 6

1. Consider the following circuit:



Assume the following:

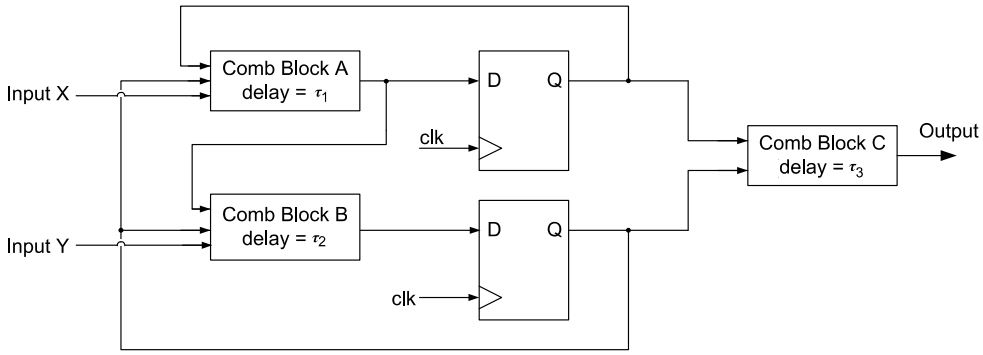
Hold time of each flip-flop: 0.15 ns
Set up time of each flip-flop: 0.2 ns
Maximum Clk-to-Q delay of each flip-flop: 0.4 ns
Minimum Clk-to-Q delay of each flip-flop: 0.1 ns
Delay of each inverter: 0.5 ns
Delay of each AND and OR gate: 0.8 ns

a) What is the maximum frequency of the clock in this circuit?
(if there is no maximum, write ∞ as your answer)

b) What is the minimum frequency of the clock in this circuit?
(if there is no minimum, write 0 as your answer)

c) Suppose the hold time is not 0.15ns. What is the largest value for the hold time for which this circuit will still function properly?

2. Consider the following circuit. Both flip-flops are clocked by the same clock.



Assume the following:

Delay of Combinational Block A: τ_1 = unknown!
Delay of Combinational Block B: τ_2 = 3 ns
Delay of Combinational Block C: τ_3 = 2 ns
Hold time of each flip-flop: 0.6 ns
Maximum Clk-to-Q delay of each flip-flop: 0.2 ns
Minimum Clk-to-Q delay of each flip-flop: 0.1 ns
Set up time of each flip-flop: 0.3 ns

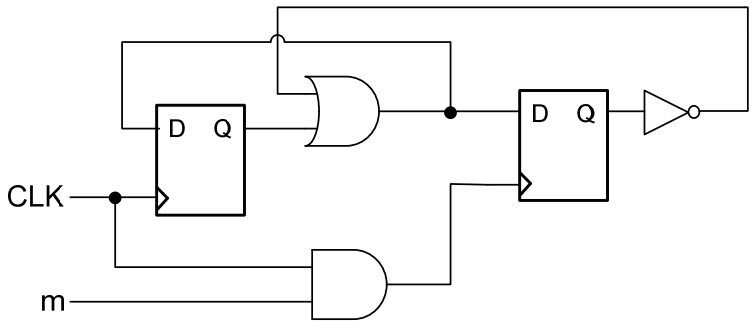
Also, suppose we want to run this circuit at 100Mhz (if you do not have a calculator, 1/100Mhz = 10 ns)

a) What is the *minimum* possible value of τ_1 for the circuit to work correctly?

b) What is the *maximum* possible value of τ_1 for the circuit to work correctly?

(continued on next page…)

3. Consider the following circuit:



Assume the following:

Set up time of each flip-flop:	0.2 ns
Hold time of each flip-flop:	0.5 ns
Maximum Clk-to-Q delay of each flip-flop:	0.8 ns
Minimum Clk-to-Q delay of each flip-flop:	0.0 ns
Delay of the OR gate:	1.2 ns
Delay of the NOT gate:	0.3 ns

The input ‘m’ does not change very often (so, you can assume it is usually ‘1’).

(a) Assume the *delay* of the AND gate is zero. What is the *maximum* frequency for CLK?

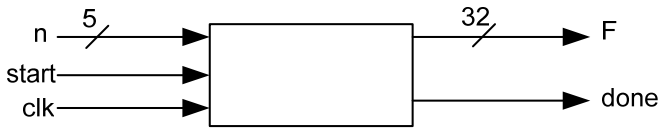
(b) Now assume the *delay* of the AND gate is 1.1 ns. What is the maximum frequency for CLK?

4. Design the *datapath* and *controller* for a circuit that calculates the *n*th element of the Fibonacci series (where *n* is an input to your circuit). Remember that the Fibonacci series is:

1, 1, 2, 3, 5, 8, 13, 21, 33, 54, 87 …

In the above series, the first element (which we will denote by F_1) is 1, and the second element (which we will denote by F_2) is 1, by definition. Each remaining element is equal to the sum of the previous two elements. So in other words, F_3 is equal to the sum of F_1 and F_2 . Similarly, F_5 is equal to the sum of F_4 and F_3 . Similarly, F_7 is equal to the sum of F_5 and F_6

The inputs and outputs of your circuit are as shown below.



The user would supply a 5-bit value on the input bus **n**, and assert **start** (in the same clock cycle). The circuit will take several clock cycles to calculate F_n. When the circuit has computed F_n , it will supply this value on the output bus **F**, and assert **done** for one clock cycle. This is the same timing behaviour as the datapath circuits we talked about in class.

*You can assume that the input **n** is greater than 2.*

You must design your circuit using a datapath controlled by a state machine. Give your answer in terms of **one or more schematics (for the datapath) and/or one or more state diagrams (for the controlling state machine)**.