CSM 61C Spring 2020

FSM

Exam Question Compilation

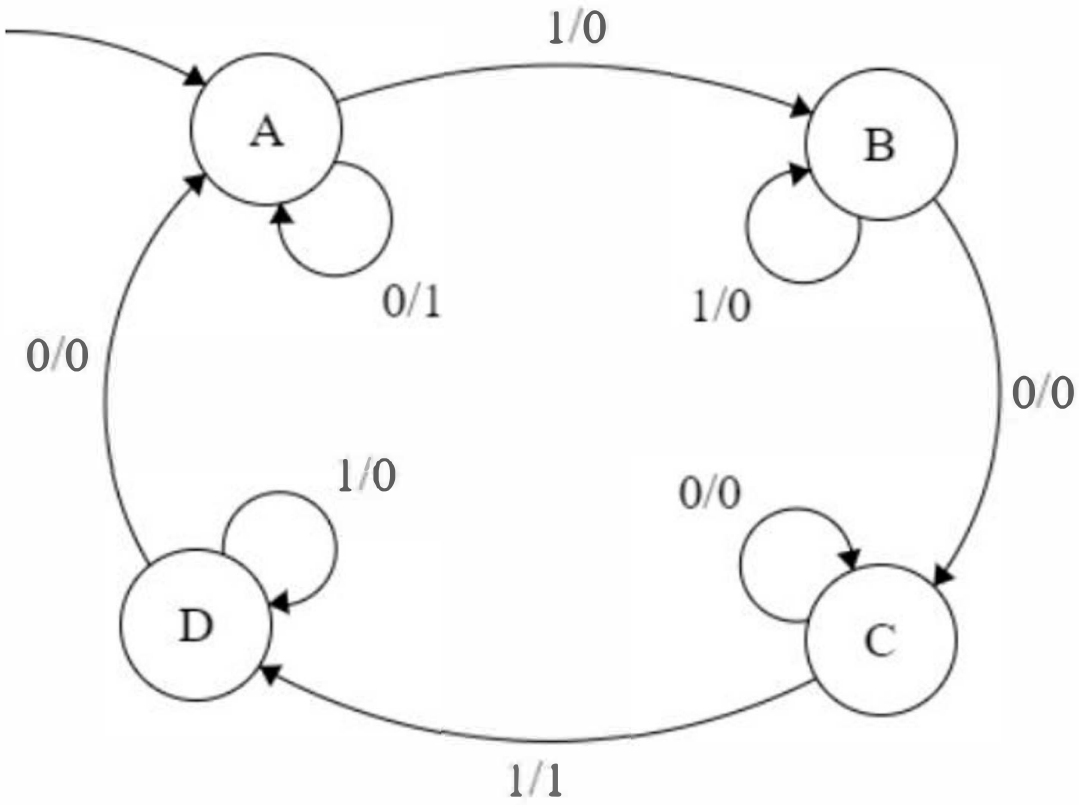
This document is a PDF version of old exam questions by topic, ordered from least difficulty to greatest difficulty.

Questions:

- Summer 2019 Final Q2
- Fall 2018 Final F2B
- Spring 2018 Midterm 2 Q2E
- Summer 2018 Final Q7 Part 2
- Fall 2019 Midterm Q5a
- Fall 2015 Final MT2-1B
- Fall 2019 Final Q5c
- Summer 2018 Midterm 2 Q1

Question 2: FSM - 8 pts

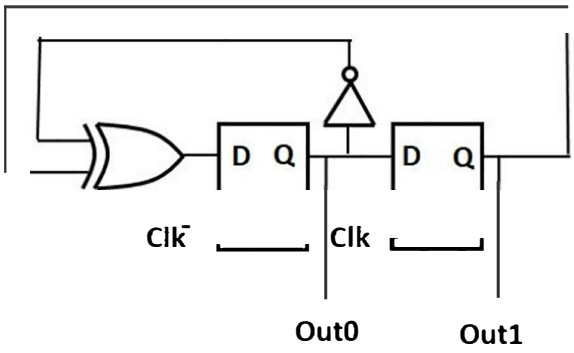
FSM Question For the following Finite State Machine, fill out the remainder of the table.



Input	-	1	0	0	1	1	0	0	0
Next State	A								
Output	-								

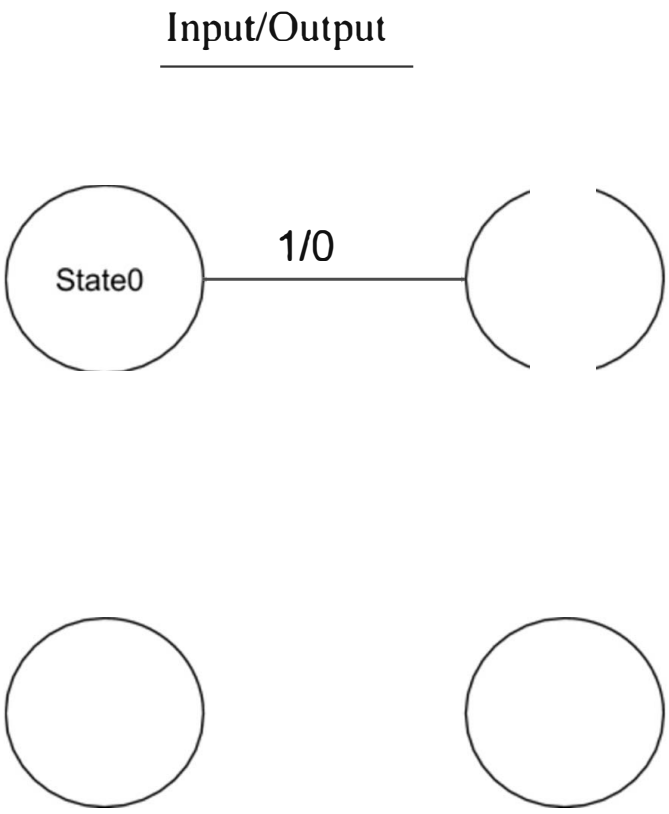
F2) *SDS* (20 points = 8+5+7, 30 minutes) SID_____

b) The logic implementation of a state machine is shown in the figure below. How many states does this state machine have? (Assume that it always starts from Out0=0, Out1=0)



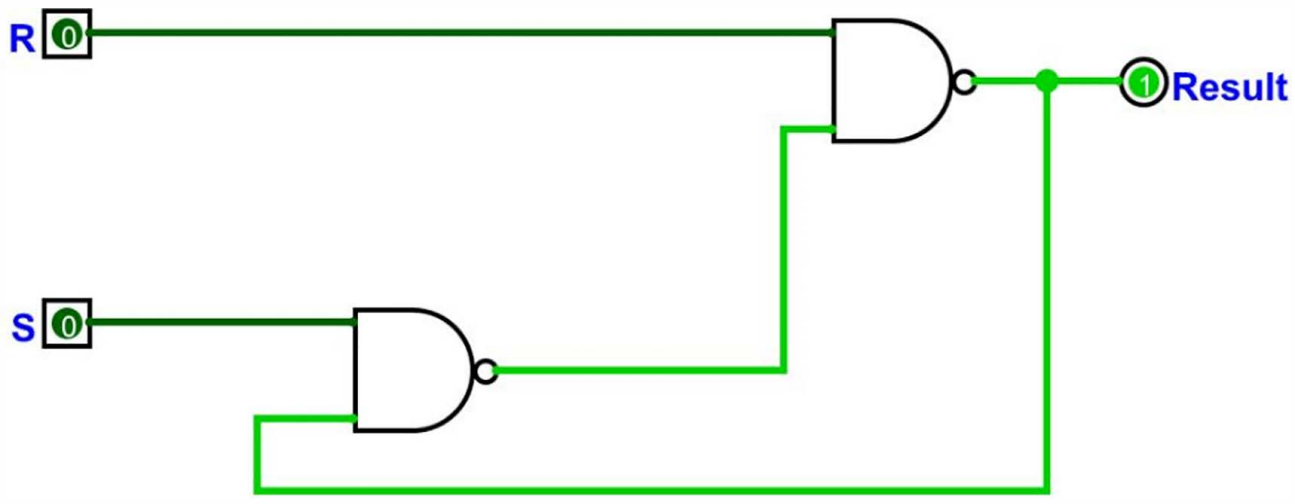
Number of states = ○1 ○2 ○3 ○4 ○5 ○6 ○7 ○8 ○9

(e) Draw the State Transition Diagram for a Finite State Machine that, given a sequence of binary digits, outputs 1 if the second most recently seen digit is 1 and outputs 0 otherwise. For example, an input of 01100111 has the output 00110011. Label all states and transition inputs/outputs you draw. You may or may not need all 6 states.



SID: _____

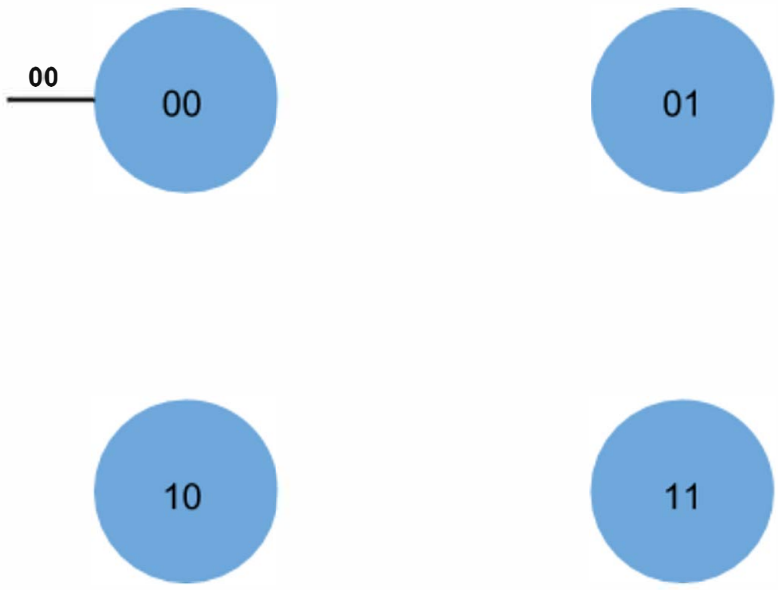
Part 2: We now connect the "X" output to the output of our second NAND gate with the hope of making something useful. The corresponding diagram is shown below.



a) In order to determine the functionality of this circuit, we decide to model it as an FSM. In this FSM, **we must flip one of either R or S (and not both) in each time step.**

Represent this as an FSM using the fewest states below. Each transition should be labeled with the bit that you're flipping, and the output of the FSM where the output is "Result".
It should be of the form: `<R or S> / <Result>.` (e.g. `"R/1"`).

We arbitrarily start at state "00". The first bit represents R and the second bit represents S.

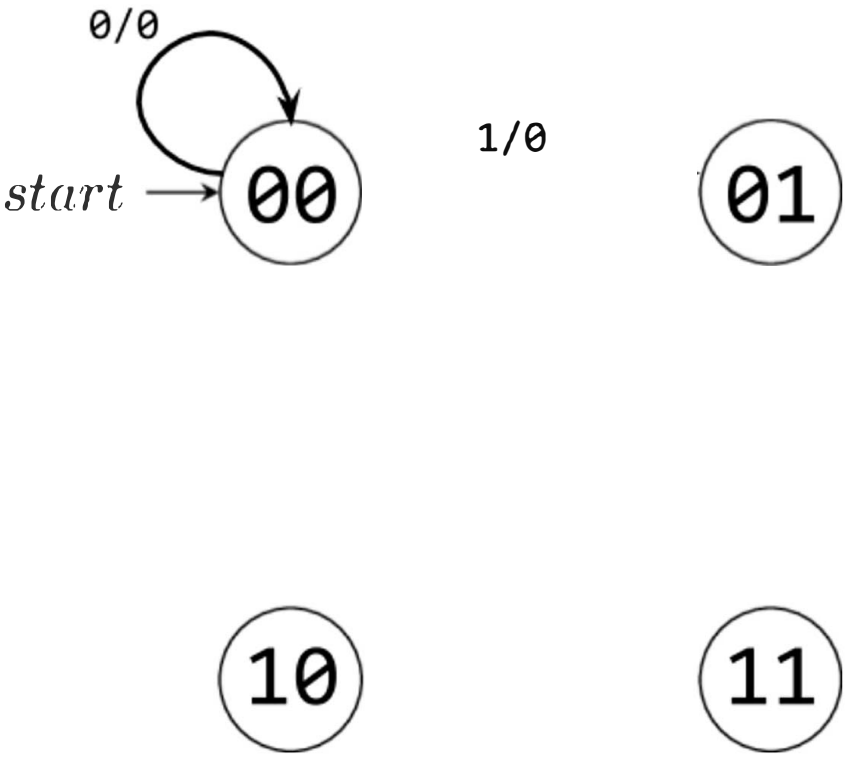


Q5) *What kind of Algebra do ghosts like? Booooooolean Algebra!* **(20 pts = 7 + 7 + 6)**

Write an FSM that takes in an n-bit binary number (starting with the MSB, ending with the LSB) and performs a **logical right shift by 2** on the input. E.g., if our input is `0b01100`, then our FSM should output `0b00011`.

Input (MSB → LSB)	`0`	`1`	`1`	`0`	`0`
Output	`0`	`0`	`0`	`1`	`1`

a) Fill in the following FSM with the correct transitions and outputs. Format state changes as (input / output); we've done two for you. This is the **minimum** number of states; you may not add any more.

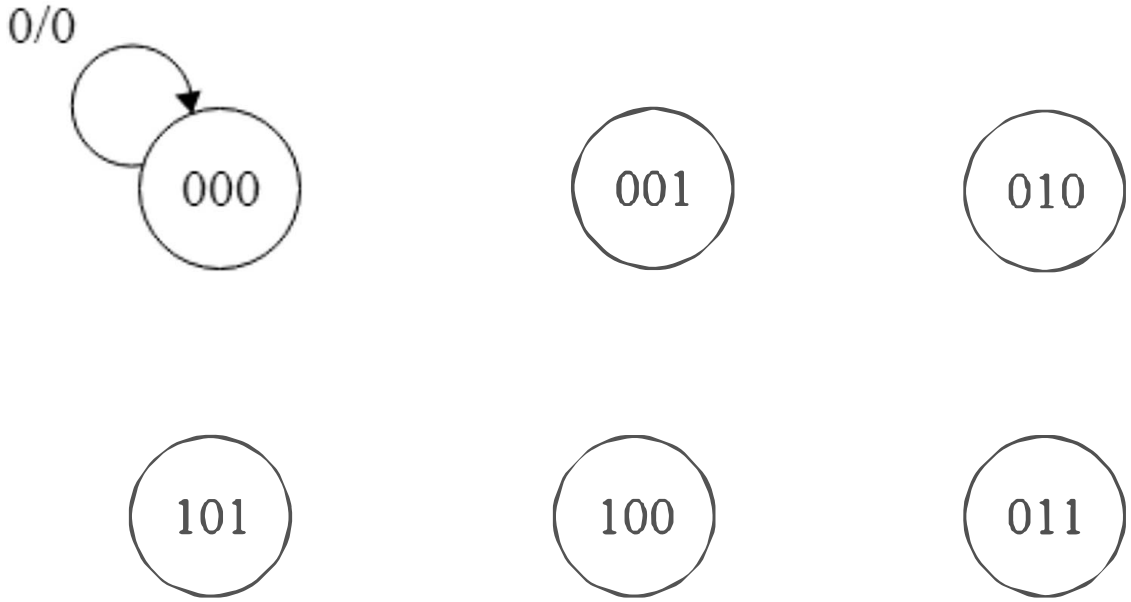


SID: _____

MT2-1: Synchronous Finite State Digital Machine Systems (9 points)

b. Consider the finite state machine below which has 6 states, and a single input that can take on the value of 0 or 1. The finite state machine should output 1 *if and only if* 6 + the sum of all the input values is not divisible by 2 or 3. One transition has been provided; complete the remainder of the diagram.

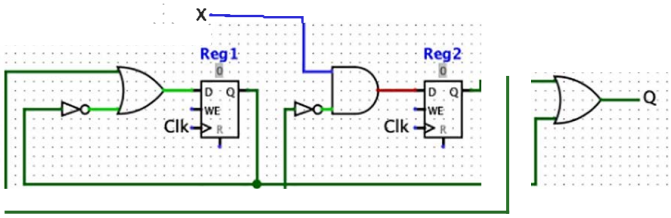
(Hint: If the sum of the inputs is a multiple of 6, then we have 6 + the sum of the inputs = 6n for some n. As 6n is divisible by 2, 6n cannot be prime.)



Fall 2019 Final

Q5) Watch the clock and don't delay! **(30 pts = 2*5 + 10 + 10)**

Consider the following circuit:

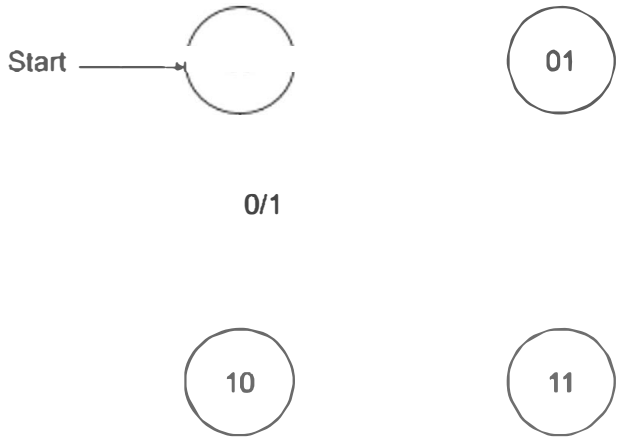


You are given the following information:

- Clk has a frequency of 50 MHz
- AND gates have a propagation delay of 2 ns
- NOT gates have a propagation delay of 4 ns
- OR gates have a propagation delay of 10 ns
- X changes 10ns after the rising edge of Clk
- Reg1 and Reg2 have a clock-to-Q delay of 2 ns

---

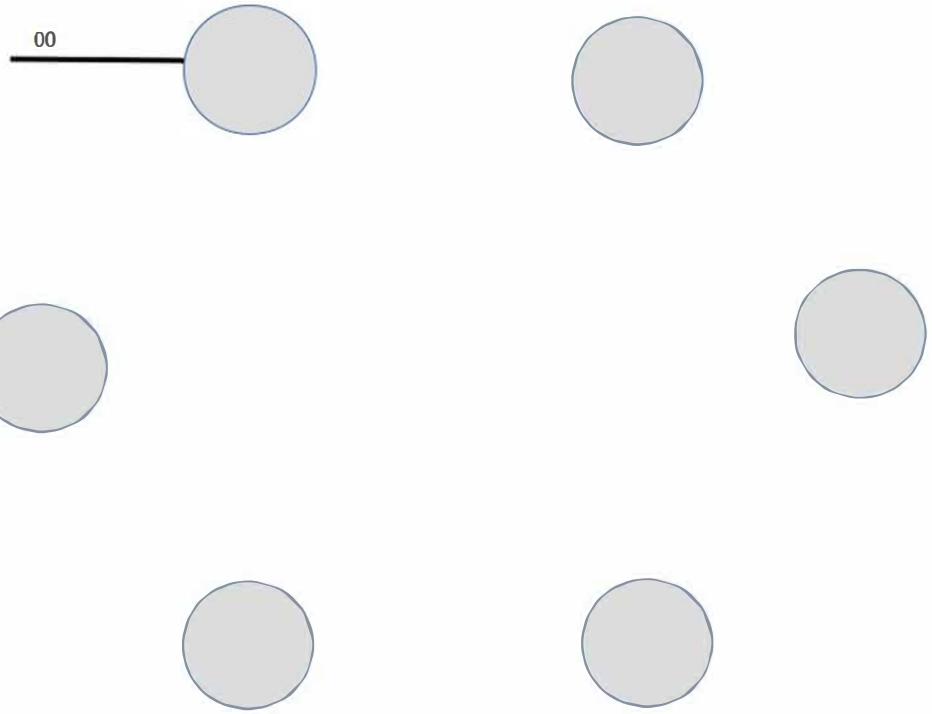
c) Represent the circuit above using an equivalent FSM, where X is the input and Q is the output, with the state labels encoding Reg1Reg2 (e.g., "01" means Reg1=0 and Reg2=1). We did one transition already.



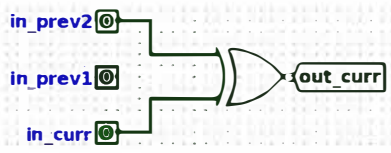
SID: _____

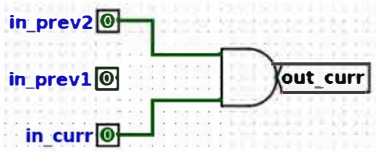
Question 1: FSMSF (11 pts)

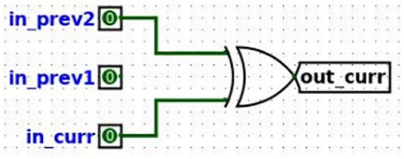
1) Construct an FSM which outputs a 1 when the most recent 3 bits in the input stream are a palindrome. A palindrome is a sequence which is the same when displayed in reverse order. An example of a palindrome is 010, whereas 001 is not. This FSM should continually be updating the output when the input stream changes. For example, when the bitstream is 0010010101, the output is **01001111. The first two outputs are undefined and you should ignore them. Assume the bitstream is always initialized to 00 for you. Your FSM should use the fewest possible number of states. As a reminder you should represent transitions with an arrow, where the tail of the arrow is the origin state and the tip is the destination state. You should also label each transition arrow as `<input> / <output>.` (e.g: 0 / 1)

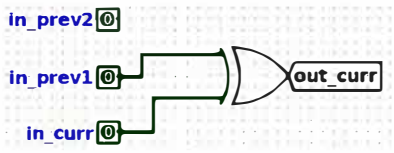


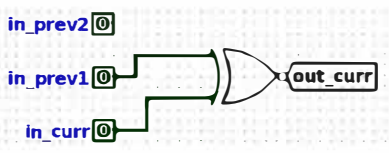
Select the circuit diagram which produces the correct output for the FSM. in_curr is the current input to the FSM (in[i]) , in_prev1 is the previous input (in[i - 1]), and in_prev2 is the input before the previous input (in[i - 2]).

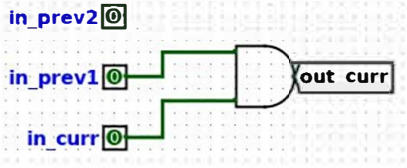
A.


B.


C.


D.


E.


F.


2) Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ