CSE 232 SPRING 2020

HOMEWORK 3

ABDULLAH CELIK
171044002

1.

a) 50 kHz

$1 \text{ kHz} = 10^3 \text{ Hz}$
$50 \text{ kHz} = 5 * 10^4 \text{ Hz}$

$1 \text{ s} = 10^9 \text{ ns}$
$10^9 / (5 * 10^4) = 2 * 10^4 \text{ ns} = 20 \ \mu\text{s}$
Period = 20 μs

b) 300 MHz

$1 \text{ MHz} = 10^6 \text{ Hz}$
$300 \text{ MHz} = 3 * 10^8 \text{ Hz}$

$1 \text{ s} = 10^9 \text{ ns}$
$10^9 / (3 * 10^8) = 3.334 \text{ ns}$
Period = 3.334 ns

c) 3.4 GHz

$1 \text{ GHz} = 10^9 \text{ Hz}$
$3.4 \text{ GHz} = 34 * 10^8 \text{ Hz}$

$1 \text{ s} = 10^9 \text{ ns}$
$10^9 / (34 * 10^8) = 0.294 \text{ ns} = 294 \text{ ps}$
Period = 294 ps

d) 10 GHz

$1 \text{ GHz} = 10^9 \text{ Hz}$
$10 \text{ GHz} = 10^{10} \text{ Hz}$

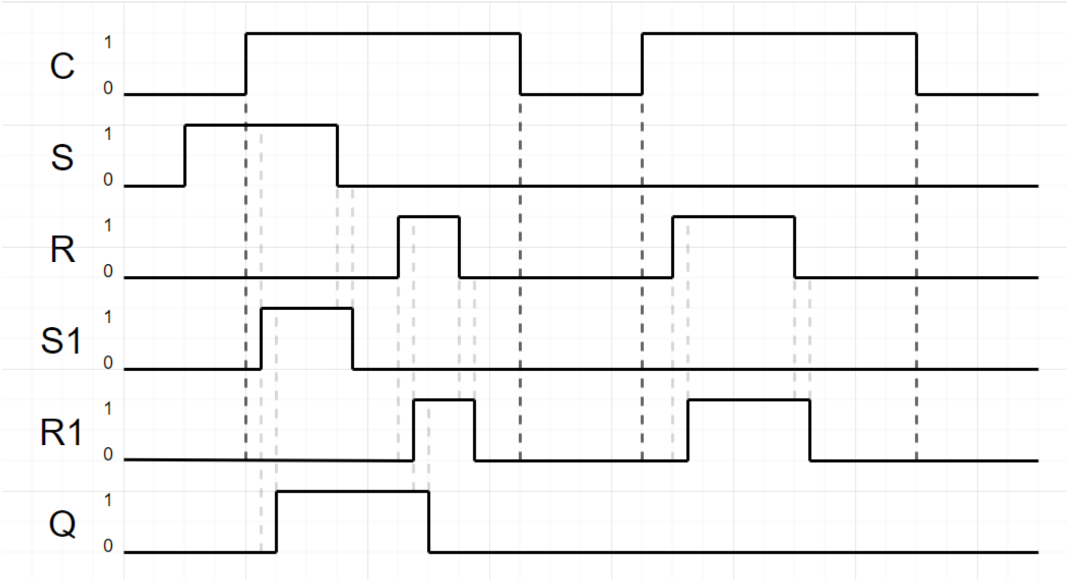
$1 \text{ s} = 10^9 \text{ ns}$
$10^9 / 10^{10} = 0.1 \text{ ns} = 100 \text{ ps}$
Period = 100 ps

e) 1THz

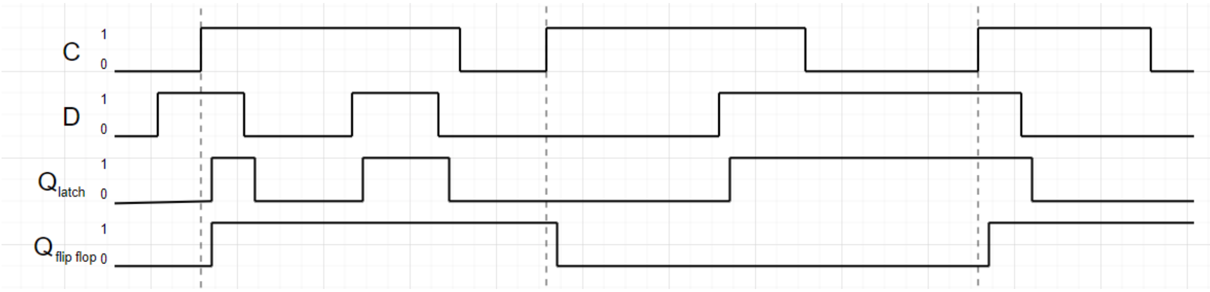
$1 \text{ THz} = 10^{12} \text{ Hz}$

$1 \text{ s} = 10^9 \text{ ns}$
$10^9 / 10^{12} = 0.001 \text{ ns} = 1 \text{ ps}$
Period = 1 ps

2.



3.



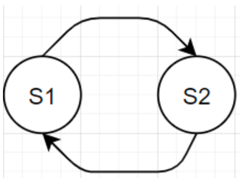
D Latch is level-sensitive. Stores D when C = 1. D Latch follows D while C is 1.
D Flip-Flip is edge triggered. Stroes D when C changes from 0 to 1. D flip-flop only loads D during C rising edge.

4.

a) 4 = 2 bits ($2^2 = 4$)
b) 8 = 3 bits ($2^3 = 8$)
c) 9 = 4 bits ($2^4 = 16$)
d) 23 = 5 bits ($2^5 = 32$)
e) 900 = 10 bits ($2^{10} = 1024$)

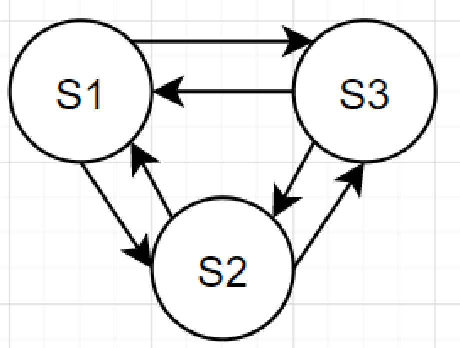
5. If an FSM has N states, what is the maximum number of possible transitions that could exist in the FSM? Assume that no pair of states has more than one transition in the same direction, and that no state has a transition point back to itself. Assuming there are a large number of inputs, meaning the number of transitions is not limited by the number of inputs? Hint: try for small N, and then generalize.

For N = 2



There are 2 transition.
(S1->S2 , S2->S1)

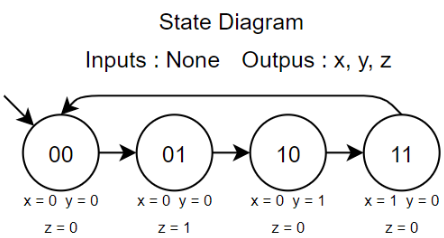
For N=3



There are 6 transition.
(S1->S2, S2->S1, S1->S3, S3->S1, S2->S3, S3->S2)

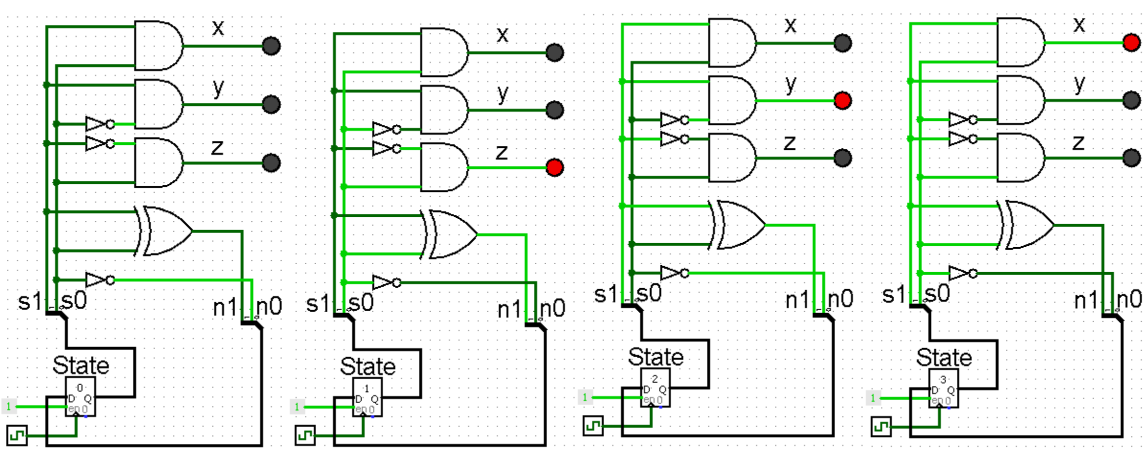
From the following, each state makes N-1 transitions with other states except itself. There are N states. So there are **N * (N-1)** maximum number of possible transitions that could exist in the FSM.

6. Draw a state diagram for an FSM with no inputs and three outputs x, y, and z. xyz should always exhibit the following sequence: 000, 001, 010, 100, repeat. The output should change only on a rising clock edge. Make 000 the initial state. Using the process for designing a controller, convert the FSM to a controller, stopping once you have created the truth table and derive the Boolean expressions.



Inputs		Outputs				
s1	**s0**	**x**	**y**	**z**	**n1**	**n0**
0	0	0	0	0	0	1
0	1	0	0	1	1	0
1	0	0	1	0	1	1
1	1	1	0	0	0	0

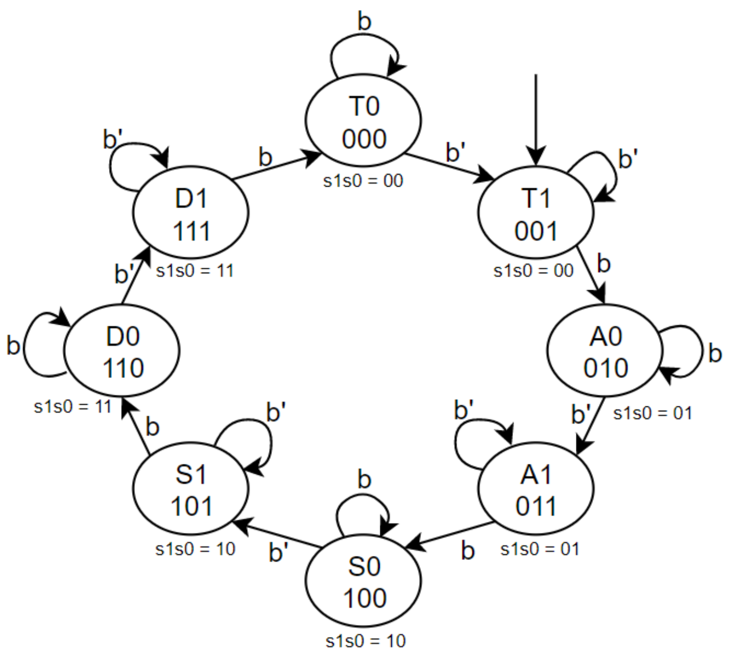
x = s1s0
y = s1s0′
z = s1′ s0
n1 = s1 XOR s0
n0 = s0′



7. A wristwatch display can show one of four items: the time, the alarm, the stopwatch, or the date, controlled by two signals s1 and s0 (00 displays the time, 01 the alarm, 10 the stopwatch, and 11 the date—assume s1s0 control an N-bit mux that passes through the appropriate register). Pressing a button B (which sets B = 1) sequences the display to the next item. For example, if the presently displayed item is the date, the next item is the current time. Create a state diagram for an FSM describing this sequencing behavior, having an input bit B, and two output bits s1 and s0. Be sure to only sequence forward by one item each time the button is pressed, regardless of how long the button is pressed—in other words, be sure to wait for the button to be released after sequencing forward one item. Use short but descriptive names for each state. Make displaying the time be the initial state. Using the process for designing a controller, convert the FSM to a controller, stopping once you have created the truth table and derive the Boolean expressions.

State Diagram

Inputs : b   Outpus : s1, s0



Inputs				Outputs				
r2	r1	r0	b	n2	n1	n0	s1	s0
0	0	0	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	1	0	0
0	0	1	1	0	1	0	0	0
0	1	0	0	0	1	1	0	1
0	1	0	1	0	1	0	0	1
0	1	1	0	0	1	1	0	1
0	1	1	1	1	0	0	0	1
1	0	0	0	1	0	1	1	0
1	0	0	1	1	0	0	1	0
1	0	1	0	1	0	1	1	0
1	0	1	1	1	1	0	1	0
1	1	0	0	1	1	1	1	1
1	1	0	1	1	1	0	1	1
1	1	1	0	1	1	1	1	1
1	1	1	1	0	0	0	1	1

n2 = r1r0(r2 XOR b) + r2(r1′ + r1r0′)
n1 = r1r0′ + r0(r1 XOR b)
n0 = b′
s1 = r2
s0 = r1