CSE 232 SPRING 2020
HOMEWORK 3
Muhammed Eyüp
1801042679

1. Compute the clock period for the following clock frequencies.

a. 50 kHz (early computers)
T = 1/f => T = 1/50000 = 2 * 10^{-5} s = 20 μs

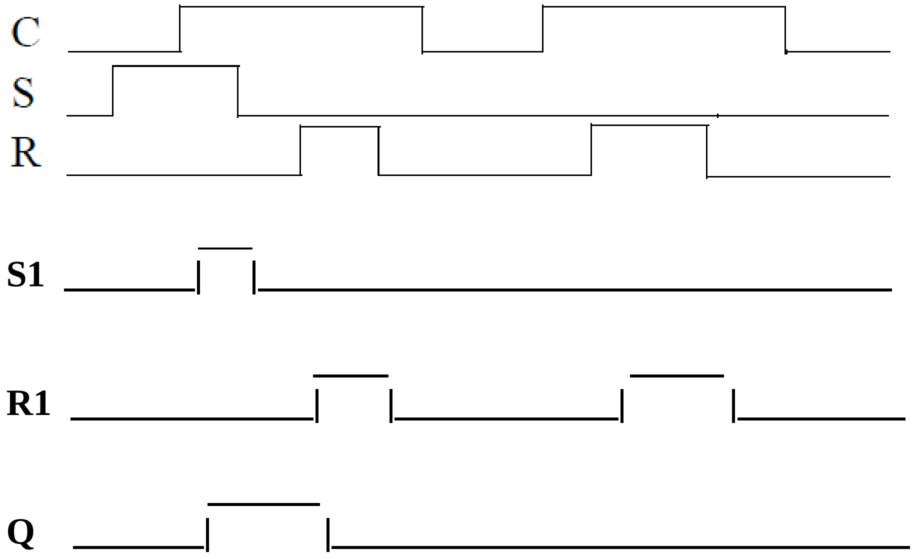
b. 300 MHz (Sony Playstation 2 processor)
T = 1/f => T = 1/(3*10^{8}) = 3.34 * 10^{-8} s = 33.4 ns

c. 3.4 GHz (Intel Pentium 4 processor)
T = 1/f => T = 1/(3.4*10^{9}) = 3 * 10^{-10} s = 0.3 ns

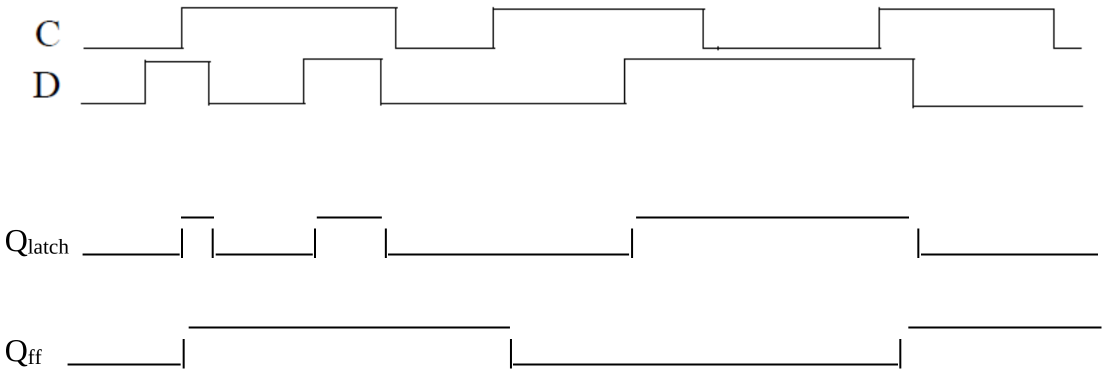
d. 10 GHz (PCs of the early 2010s)
T = 1/f => T = 1/(10*10^{9}) = 1 * 10^{-10} s = 0.1 ns

e. 1 THz (1 terahertz) (PCs of the future?)
T = 1/f => T = 1/(1*10^{12}) = 1 * 10^{-12} s = 1 ps

2. Trace the behavior of a level-sensitive SR latch for the input pattern in below figure. Assume S1, R1, and Q are initially 0. Complete the timing diagram for S1, R1 and Q, assuming logic gates have a tiny but non-zero delay.



3. Compare the behavior of D latch and D flip-flop devices by completing the timing diagram adding Q (latch) and Q (flip-flop) in below figure. Provide a brief explanation of the behavior of each device. Assume each device initially stores a 0.



The difference between a D latch and a D flip-flop is that a D latch is level sensitive i.e. it changes the output (Q) according to the input (D) as long as the clock is 1 whereas the D flip-flop is edge triggered i.e. it changes the output (Q) according to the input (D) only when a new clock cycle begins despite the change in D

4. FSMs with the following numbers of states, indicate the smallest possible number of bits for a state register representing those states:

a. 4: 4 states need a 2bit register to store as 2bit register can store 2^2 = 4 different states

b. 8: 8 states need a 3bit register to store as 3bit register can store 2^3 = 8 different states

c. 9: 9 states need a 4bit register to store as 4bit register can store 2^4 = 16 different states

d. 23: 23 states need a 5bit register to store as 5bit register can store 2^5 = 32 different states

e. 900: 900 states need a 10bit register to store as 10bit register can store 2^{10} = 1024 different states

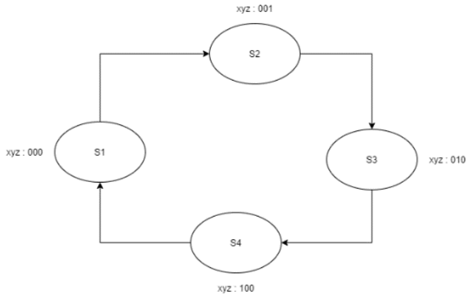
5. If an FSM has N states, what is the maximum number of possible transitions that could exist in the FSM? Assume that no pair of states has more than one transition in the same direction, and that no state has a transition point back to itself. Assuming there are a large number of inputs, meaning the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

Each state can transition to N-1 remaining states assuming the is N state in total and each state cannot transition to itself then there is N-1 possible transitions for each state and there is N states in total so the total number of possible transitions is N(N-1) transitions.

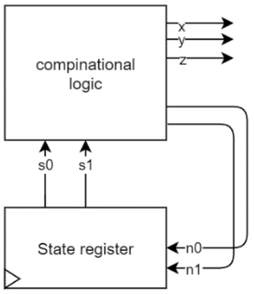
6. Draw a state diagram for an FSM with no inputs and three outputs x, y, and z. xyz should always exhibit the following sequence: 000, 001, 010, 100, repeat. The output should change only on a rising clock edge. Make 000 the initial state. Using the process for designing a

controller, convert the FSM to a controller, stopping once you have created the truth table and derive the Boolean expressions.

Inputs			outputs					
state	**s0**	**s1**	**x**	**y**	**z**	**n0**	**n1**	**Next State**
S1	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**S2**
S2	**0**	**1**	**0**	**0**	**1**	**1**	**0**	**S3**
S3	**1**	**0**	**0**	**1**	**0**	**1**	**1**	**S4**
S4	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**S1**

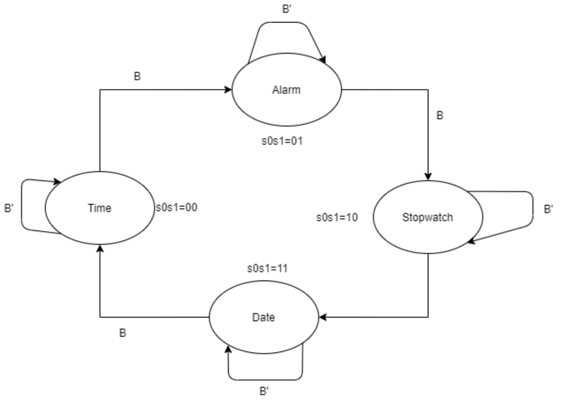


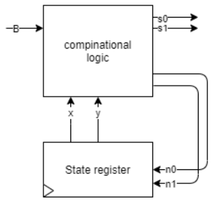
x = s1s0
y = s1s0’
z = s1’s0
n1 = s1’s0 + s1s0’ = s1 XOR s0
n0 = s1’s0’ + s1s0’ = s0’



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.

Inputs				outputs				
state	x	y	B	n0	n1	s0	s1	Next State
Time	0	0	0	0	0	0	0	Time
	0	0	1	0	1	0	1	Alarm
Alarm	0	1	0	0	1	0	1	Alarm
	0	1	1	1	0	1	0	SW
SW	1	0	0	1	0	1	0	SW
	1	0	1	1	1	1	1	Date
Date	1	1	0	1	1	1	1	Date
	1	1	1	0	0	0	0	Time





$s0 = x'yB + xy'B' + xy'B + xyB'$

$\quad = x'yB + xy' + xB'$

$s1 = x'y'B + x'yB' + xy'B + xyB'$

$\quad = y'B + yB' = y \text{ XOR } B$

$n0 = s0$

$n1 = s1$