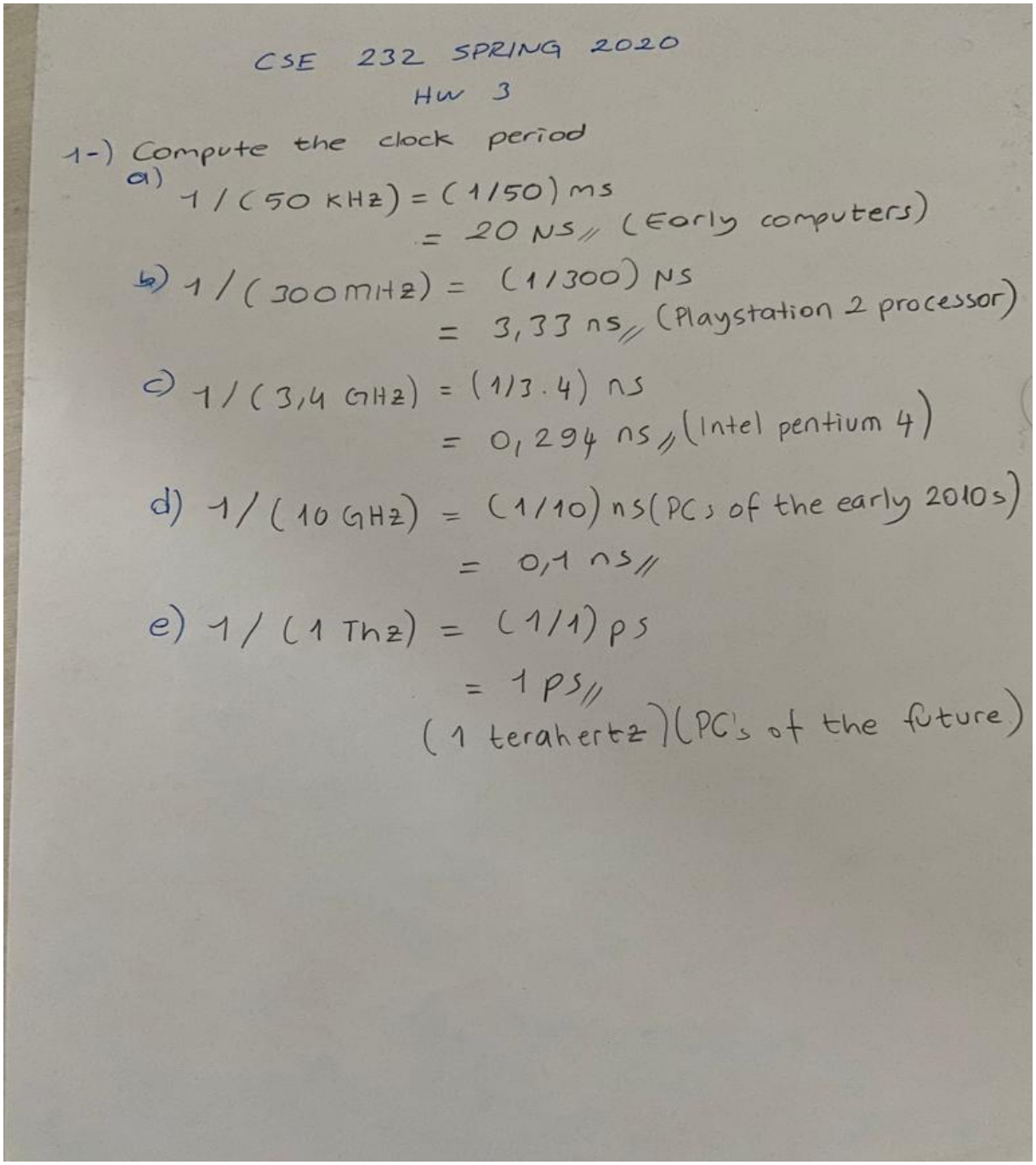
1801042693

Muhammet Fikret ATAR

Cse 232 Spring 2020 Hw3



CSE 232 SPRING 2020

HW 3

1-) Compute the clock period

a) $1/(50\ KHz) = (1/50)\ ms$

$= 20\ NS$ // (Early computers)

b) $1/(300\ MHz) = (1/300)\ NS$

$= 3,33\ ns$ // (Playstation 2 processor)

c) $1/(3,4\ GHz) = (1/3.4)\ ns$

$= 0,294\ ns$ // (Intel pentium 4)

d) $1/(10\ GHz) = (1/10)\ ns$ (PCs of the early 2010s)

$= 0,1\ ns$ //

e) $1/(1\ Thz) = (1/1)\ ps$

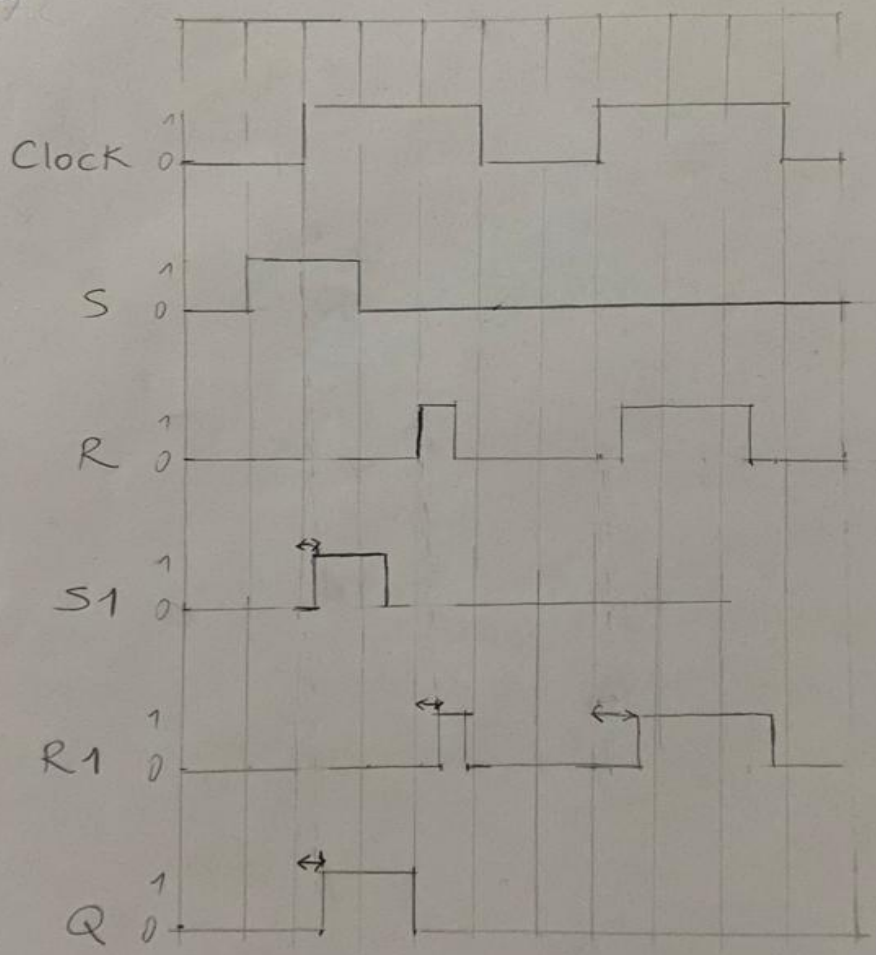
$= 1\ ps$ //

(1 terahertz) (PC's of the future)

CSE 232 SPRING 2020

HW 3

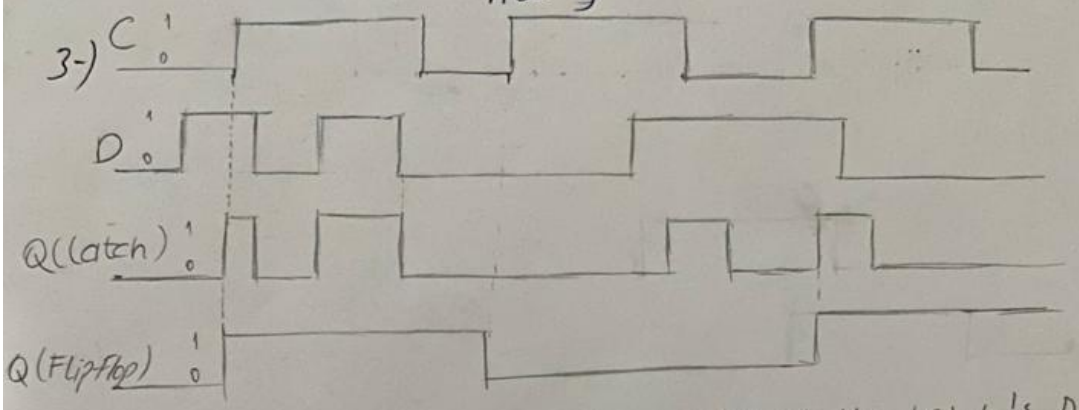
2-)



CSE 232 SPRING 2020

HW 3

3-)



D-latch stores whatever value is present at the latch's D input when clock (C=1), and remembers that value when clock (C=0). D-Latch continuously checks the input and changes the output when C=1. Flip-flop changes the output when clock signal goes from high to low or low to high transition. In D-latch, input C acts like an enable signal whereas in flip-flop, C input considered at rising edge or falling edge. D-latch is level triggered. A D-flip flop is edge triggered.

4-) The smallest possible number of bits for a state register is; Number of bits = $\log_2 n$

a) $\log_2 4 = \log_2 2^2 = 2$ //

b) $\log_2 8 = \log_2 2^3 = 3$ //

c) $\log_2 9 = 3,16 \Rightarrow 4$ // (It's must be integer)

d) $\log_2 23 = 4,52 \Rightarrow 5$ // (It's must be integer)

e) $\log_2 900 = 9,8 \Rightarrow 10$ // (It's must be integer)

CSE 232 SPRING 2020

HW 3

5-) For example;

There are 2 state or there are 3 state

A → B
B → A

A → B B → A C → A
A → C B → C C → B

Number of transitions = 2x1=2 Number of transition 3x2=6

N = State

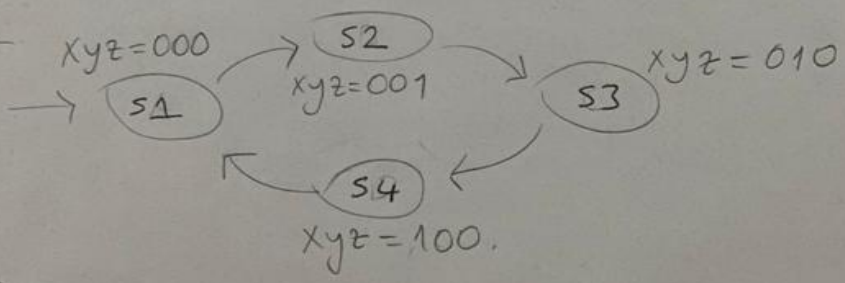
Thus the maximum possible number of transitions

⇒ $\boxed{N(N-1)}$

in FSM with N states.

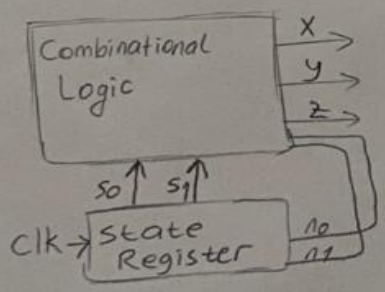
6-) Input : None , Outputs : x, y, z

Step 1



Step 2

a)



CSE 232 SPRING 2020

Hw3

Step 2

b)

	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

$X = s1\,s0$

$y = s1\,s0'$

$z = s1'\,s0$

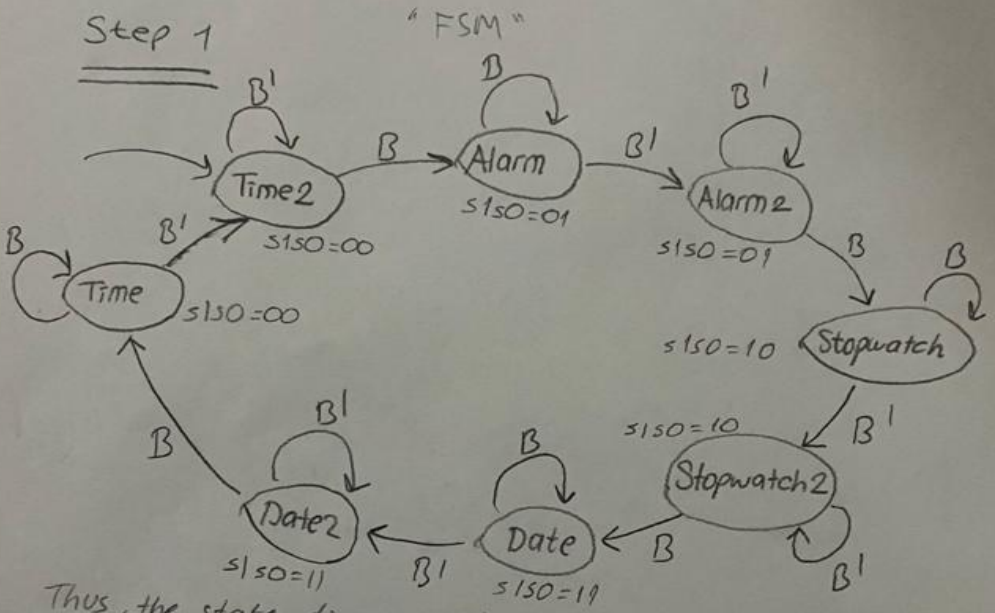
$n1 = s1's0 + s1s0' = s1 \oplus s0$

$n0 = s1's0' + s1s0' = s0'$

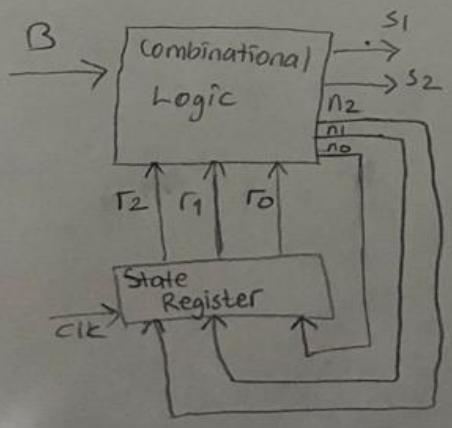
CSE 232 SPRING 2020
HW3

7-)

Step 1



Thus, the state diagram of the FSM is obtained with an input B and two outputs S1 and S0.



Encode States

Time2 000
Alarm 001
Alarm2 010
Stopwatch 011
Stopwatch2 100
Date 101
Date2 110
Time 111.

other page

CSE 232 SPRING 2020

HW3

	inputs				outputs				
	r_2	r_1	r_0	B	S1	S0	n_2	n_1	n_0
Time 2	0	0	0	0	0	0	0	0	0
	0	0	0	1	0	0	0	0	1
Alarm	0	0	1	0	0	1	0	1	0
	0	0	1	1	0	1	0	0	1
Alarm 2	0	1	0	0	0	1	0	1	0
	0	1	0	1	0	1	0	1	1
Stop watch	0	1	1	0	1	0	1	0	0
	0	1	1	1	1	0	0	1	1
stop watch 2	1	0	0	0	1	0	1	0	0
	1	0	0	1	1	0	1	0	1
Date	1	0	1	0	1	1	1	1	0
	1	0	1	1	1	1	1	0	1
Date 2	1	1	0	0	1	1	1	1	0
	1	1	0	1	1	1	1	1	1
Time	1	1	1	0	0	0	0	0	0
	1	1	1	1	0	0	1	1	1

$S_1 = r_2' r_1 r_0 + r_2 r_1' r_0' + r_2 r_1' r_0 + r_2 r_1 r_0'$

$S_1 = r_0 (r_2 \text{ xor } r_1) + r_2 r_0'$

$S_0 = r_2' r_1' r_0 + r_2' r_1 r_0' + r_2 r_1' r_0 + r_2 r_1 r_0'$

$S_0 = r_1 \text{ xor } r_0$

$n_2 = B' + r_2 r_1' r_0' + r_2 r_1' r_0 + r_2 r_1 r_0' + B$

$n_2 = r_2 r_1' + r_2 r_1 r_0' = r_2 (r_1' + r_1 r_0')$

$n_2 = r_2 \cdot (r_1' + r_0')$

$n_1 = B' + r_2' r_1 r_0' + B + B' + r_2 r_1 r_0' + B$

$n_1 = r_0' \cdot (r_2 \oplus r_1)$

$n_0 = B$