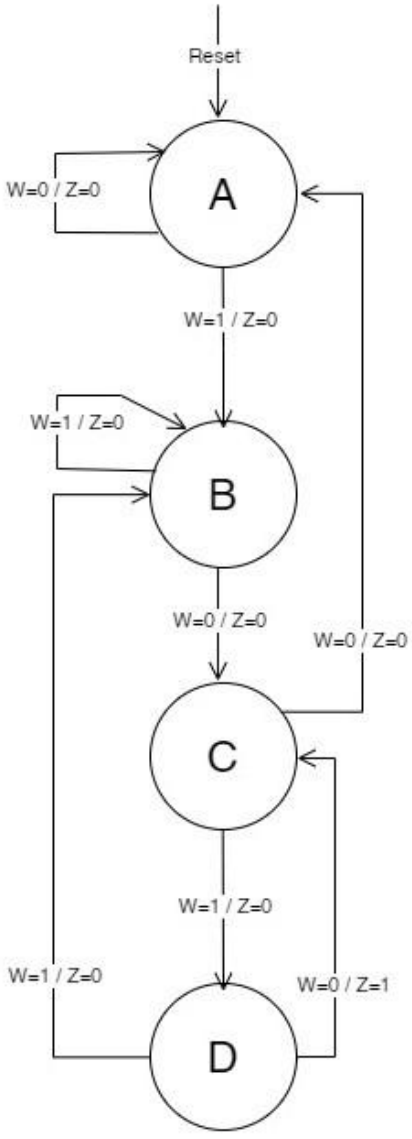
CSE460 Lab Assignment 2

Name : Abdullah Hasan Sajjad Rafi
ID : 19301097
Section : 5

Problem 1 : An FSM has an input w and an output z. The machine has to generate z = 1 when the following patterns in w are detected: 1010; otherwise, z = 0. The machine should reset when (Reset = 0).

Answer :

a. Mealy type FSM will be used
b. State diagram :



c. State-assigned table :

Present state (y2 y1)	Next state Y2 Y1		Output (Z)	
	w=0	w=1	w=0	w=1
A(00)	A(00)	B(01)	0	0
B(01)	C(10)	B(01)	0	0
C(10)	A(00)	D(11)	0	0
D(11)	C(10)	B(01)	1	0

Code :

```
module prob1_1(clock, reset, w, z, present_state,
next_state);

input clock, reset,w;

output reg z;
output reg [1:0] present_state, next_state;

parameter stateA= 2'b00,
stateB= 2'b01,
stateC= 2'b10,
stateD= 2'b11;

always@(posedge clock)
begin
     if(reset==1)
          begin
               present_state = stateA;
               next_state = stateA;
          end
     else
          begin
               present_state = next_state;
               case(present_state)

                    // stateA
                    stateA: if(w == 0)

                    begin
                    next_state = stateA;
                    z =  0;
                    end
```

```
else if(w == 1)
begin
next_state = stateB;
z = 0;
end

// stateB
stateB: if(w == 0)

begin
next_state = stateC;
z =  0;
end

else if(w == 1)

begin
next_state = stateB;
z = 0;
end

// stateC
stateC: if(w == 0)

begin
next_state = stateA;
z =  0;
end

else if(w == 1)

begin
next_state = stateD;
z = 0;
end


// stateD
stateD: if(w == 0)

begin
next_state = stateC;
z =  1;
end
```

```
                        else if(w == 1)

                        begin
                        next_state = stateB;
                        z = 0;
                        end

                    endcase
            end
end
endmodule
```

E.

as we are using mealy type FSM so, when we will get 1010 sequence then on that same clock cycle we will get z = 1 as output.

now, in the timing diagram from 0ns — 50ns the input was 0 so, we stay at state A(00). Then from 50ns — 100ns input was 1 and present state was B. Next three clock cycly's positive edge the input was 0 1 0 sequentially. Which means we should get a high output for z in time frame 200ns - 250ns and the output z=1 then.

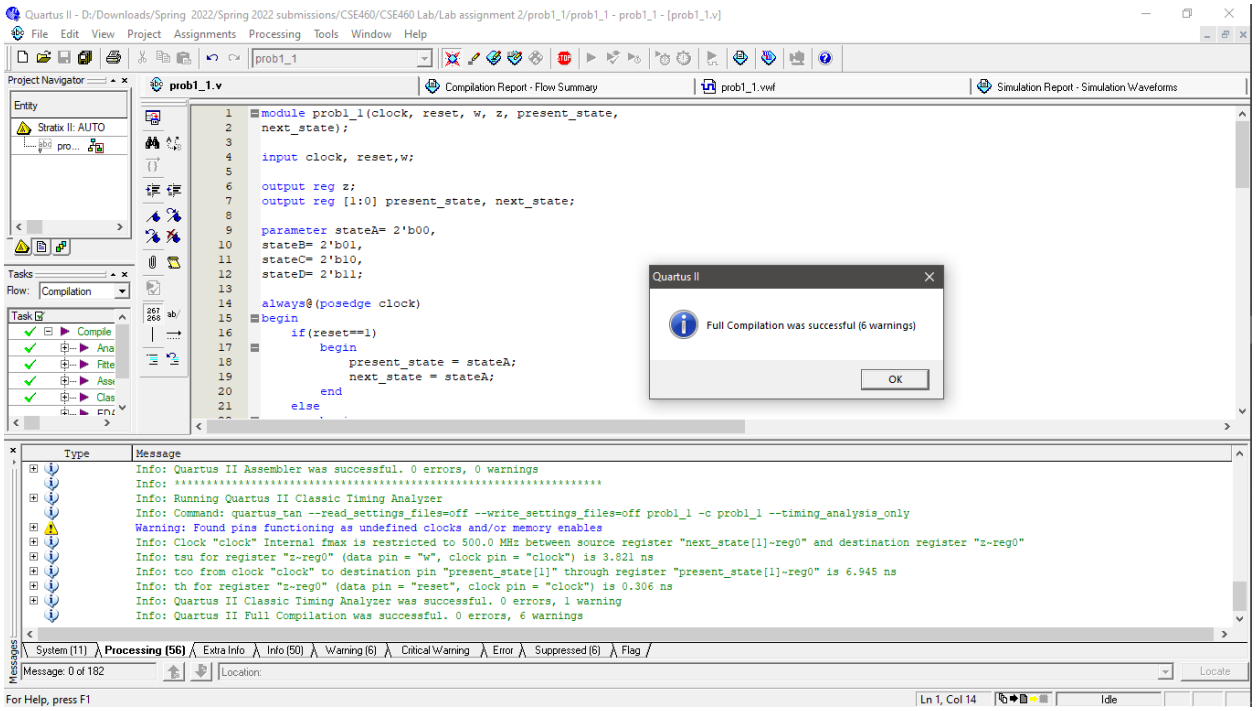
From 250ns - 300ns z=0 cause the input was 1 but from 300ns to 350ns the input was 0 so, the output z=1 because of the overlapping sequence we get 1010 from 150ns — 350ns.

From the simulation I get,

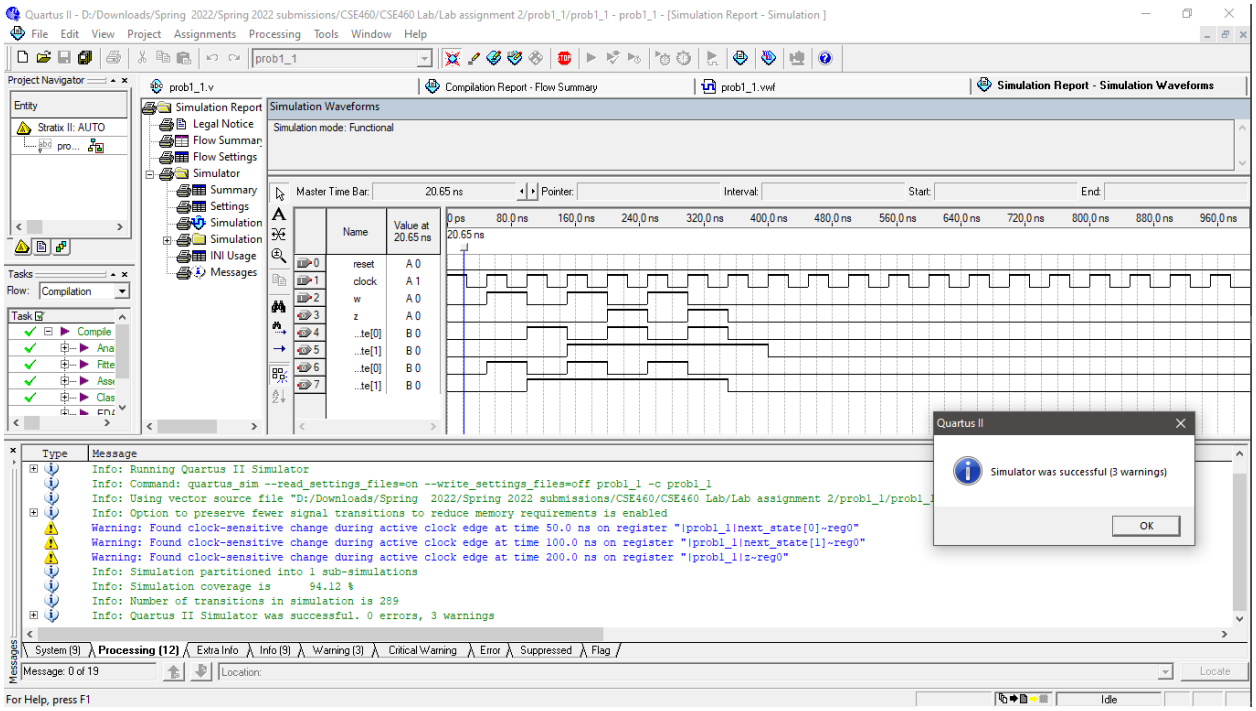
clock	(0-50)ns t_1	(50-100)ns t_2	(100-150)ns t_3	(150-200)ns t_4	(200-250)ns t_5	(250-300)ns t_6	(300-350)ns t_7	(350-400)ns t_8	(400-450)ns t_9
w	0	1	0	1	0	1	0	0	0
z	0	0	0	0	1	0	1	0	0

In the simulation at (200-250)ns and at (300-350)ns the output is high

Compilation report:



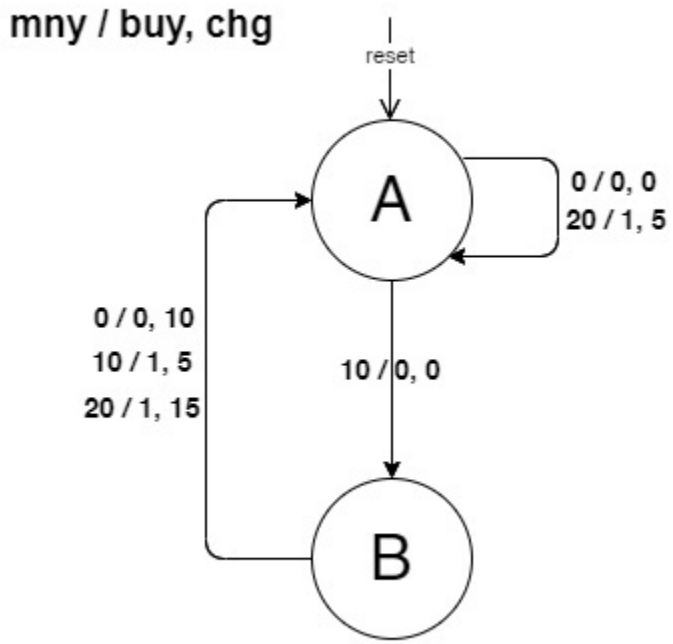
Simulation report:



Problem 2 : You have to design a vending machine for a 15 Tk product. User's money, returned money by the machine, and product bought condition is represented as mny (2-bit input), chg (output), and buy (1-bit output).
The vending machine can only accept inputs: no money (mny = 00), Tk 10 (mny = 01), and Tk 20 (mny = 10). If no money is inserted into the machine, it will immediately return the user's money back. Once an acceptable input is more than or equal to 20 Tk, the machine immediately generates an output (buy=1), goes back to the initial state, and gives back the change (if required).

Answer :

State diagram :



The machine will produce 4 types of changes.

0tk, 5tk, 10tk, 15tk

As we have 4 changes so we need 2bit for chg output to represent all changes.

0tk ⟶ 00
5tk ⟶ 01
10tk ⟶ 10
15tk ⟶ 11

clk	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
my	0	10	0	0	10	10	0	0	20	0
buy	0	0	0	0	0	1	0	0	1	0
chg	0	0	10	0	0	5	0	0	5	0

chg 0tk : at t_4 user didn't gave any money so the change is 0 tk

chg 5tk : at t_9 users gives 20 tk so the by = 1 and change 5tk

chg 10 tk : at t_2 user gives 10 tk and t_3 user gave nothing so machine will return to state A and return is 10tk at t_3

chg 15 tk : this scenario isn't present in table but if user gives 10 tk and then 20 tk consecutively then user will get 15 tk change

A. State-assigned table :

Present state (y)	Next state Y			Output					
				buy			**chg**		
	(mny2 mny1)			(mny2 mny1)			(mny2 mny1)		
	00 (0tk)	**01** (10tk)	**10** (20tk)	**00** (0tk)	**01** (10tk)	**10** (20tk)	**00** (0tk)	**01** (10tk)	**10** (20tk)
A(0)	A(0)	B(1)	A(0)	0	0	1	00 (0tk)	00 (0tk)	01 (5tk)
B(1)	A(0)	A(0)	A(0)	0	1	1	10 (10tk)	01 (5tk)	11 (15tk)

Code :

```
module exp2_2(clock, reset, mny, buy, present_state, next_state,
chg);

     input clock, reset;
     input [1:0]  mny;

     output reg buy, present_state, next_state;
     output reg [1:0] chg;

     parameter      stateA= 0, //0tk/final state
                    stateB= 1, //10tk state

                    n = 15,//price of my product

                    R0= 2'b00, //0tk return
                    R5= 2'b01, //5tk return
                    R10= 2'b10, //10tk return
                    R15= 2'b11; //15tk return

     always@(posedge clock)
     begin
               if(reset==1)
               begin
                    present_state = stateA;
                    next_state = stateA;
               end
               else
               begin
                    present_state = next_state;

                    case(present_state)
                    stateA: if(mny == 2'b00) // 0 tk
                                   begin
                                        next_state = stateA;
                                        buy =0;
                                        chg = R0;
                                   end
```

```
                    else if(mny == 2'b01) // 10 tk
                         begin
                              next_state = stateB;
                              buy =0;
                              chg = R0;
                         end
                    else if(mny == 2'b10) // 20 tk
                         begin
                              next_state = stateA;
                              buy =1;
                              chg = R5;
                         end

               stateB: if(mny == 2'b00) // 0 tk
                         begin
                              next_state = stateA;
                              buy =0;
                              chg = R10;
                         end
                    else if(mny == 2'b01) // 10 tk
                         begin
                              next_state = stateA;
                              buy =1;
                              chg = R5;
                         end
                    else if(mny == 2'b10) // 20 tk
                         begin
                              next_state = stateA;
                              buy =1;
                              chg = R15;
                         end
               endcase
          end
end
endmodule
```

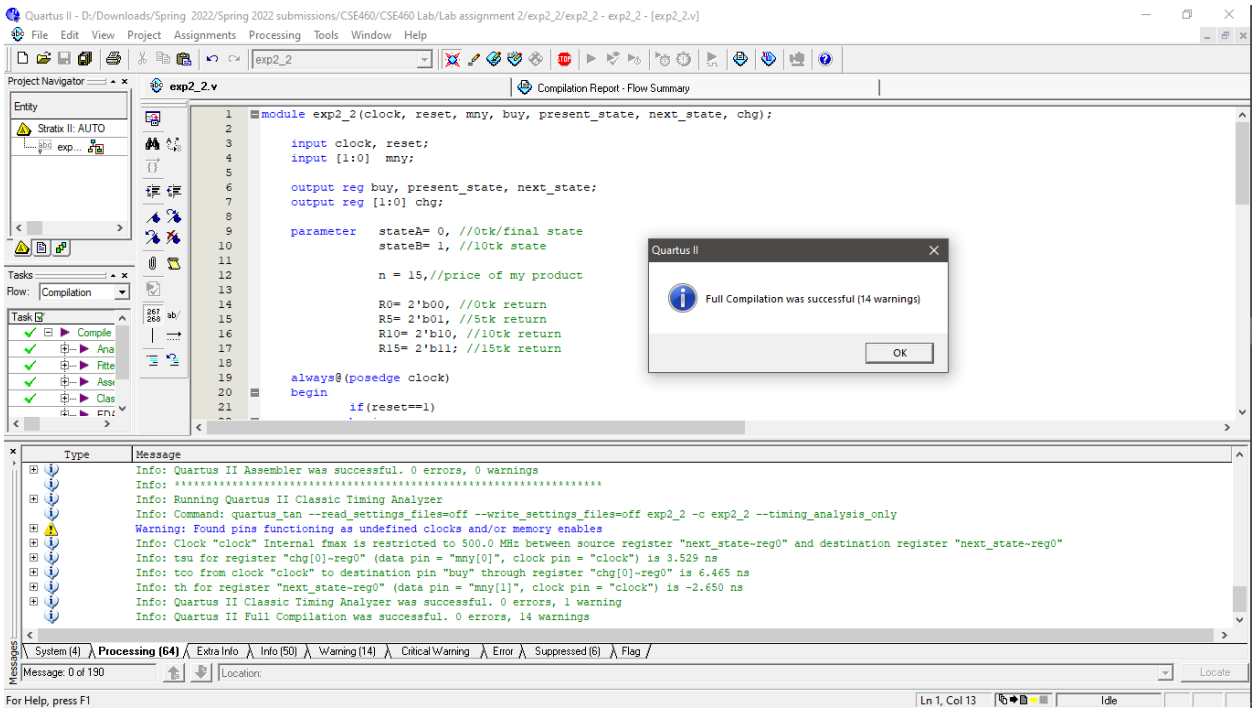
E.

In the simulation at time (300-350) ns the buy = 1 cause in the past two clock cycle the input was 10 + 10 = 20tk. the reason why in this time frame the change is 5tk.

Also in the (450-500) ns the output is buy = 1 cause the past clock cycles input is 20tk. The change is 5tk.

At (150-200) ns the buy = 0 and change is 10 tk cause the past two clock cycle's input was 10 tk.

Compilation report:



Simulation report: