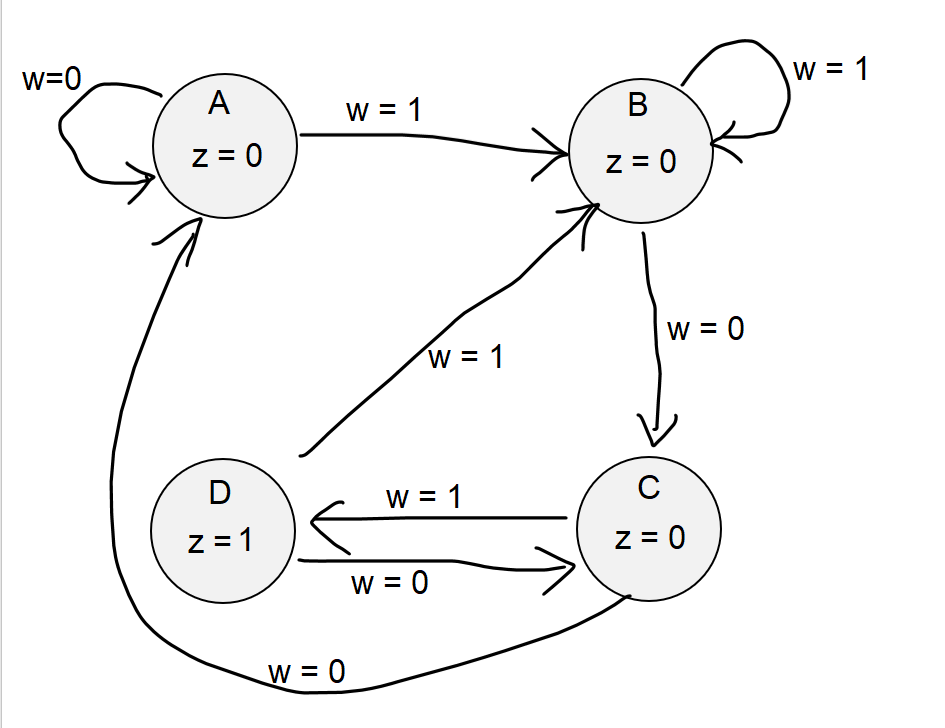
**CSE460 Lab Assignment 2**

Name: Ashfia Khanam

ID:18301231 Section: 11

**Problem 1:**

**State Diagram:**

****

**Code:**

Graphical user interface, text, application, email

Description automatically generated

**Simulation Report:**

Graphical user interface, application

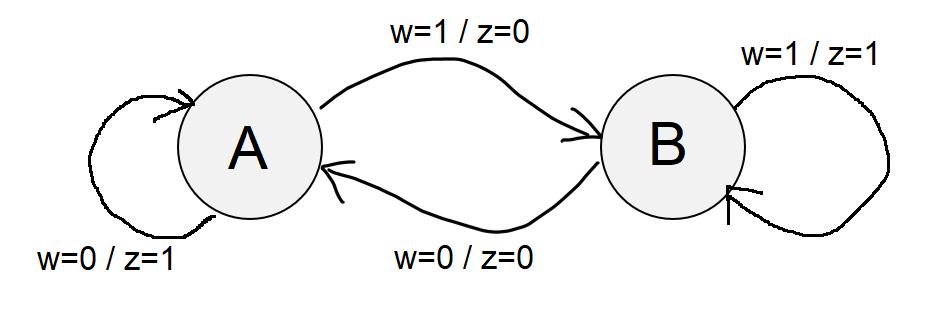
Description automatically generated

**Explanation:**

This a Moore type FSM for pattern “101”. If the bit pattern of “101” is detected, then the output will be set to 1 in the next clock cycle. Here in the simulation report we can see that there is 1 then 0 and then again 1 in “w”. after detecting this pattern, in next clock cycle “z” is 1.

**Problem 2:**

**State Diagram:**

****

**Code:**

Graphical user interface, text, application

Description automatically generated

**Simulation Report:**

**Graphical user interface, application, table, Excel

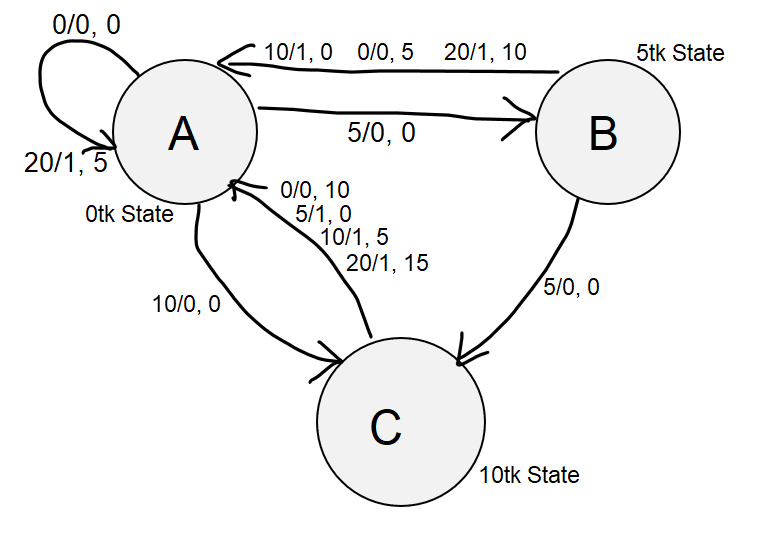
Description automatically generated**

**Explanation:**

This is a Mealy type FSM which detects sequence of bit pattern “11” or “00”. If the bit pattern of “11”, the “z” will be “1” at same clock cycle or if it is “00”, the “z” will be “1” at same clock cycle is detected, then the output will be set to 1 in the current clock cycle. In this simulation report we can see that the place where w is continuously 1 or 0 (two or more times), the z becomes 1 after detecting second similar value (11… or 00…).

**Problem 3:**

**State Diagram:**

****

**Code:**

module task3(Clock,purchase,cash\_in, cash\_return);

input Clock;

input [2:1] cash\_in;

output reg purchase;

output reg [2:1] cash\_return;

reg [2:1] next\_state;

parameter [1:0] A = 0, B = 1, C = 2;

always @(posedge Clock)

case(next\_state)

A: if(cash\_in == 0)

begin

next\_state <= A;

purchase <= 0;

cash\_return <= 0;

end

else if(cash\_in == 1)

begin

next\_state <= B;

purchase <= 0;

cash\_return <= 0;

end

else if(cash\_in == 2)

begin

next\_state <= C;

purchase <= 0;

cash\_return <= 0;

end

else if(cash\_in == 3)

begin

next\_state <= A;

purchase <= 1;

cash\_return <= 1;

end

B: if(cash\_in == 0)

begin

next\_state <= A ;

purchase <= 0;

cash\_return <=1;

end

else if(cash\_in == 1)

begin

next\_state <= C;

purchase <= 0;

cash\_return <= 0;

end

else if(cash\_in == 1)

begin

next\_state <= A;

purchase <= 1;

cash\_return <= 0;

end

else if(cash\_in == 3)

begin

next\_state <= A;

purchase<=1;

cash\_return <= 2;

end

C: if(cash\_in == 0)

begin

next\_state <= A;

purchase <= 0;

cash\_return <= 2;

end

else if(cash\_in == 1)

begin

next\_state <= A;

purchase <= 1;

cash\_return <= 0;

end

else if(cash\_in == 2)

begin

next\_state <= A;

purchase <= 1;

cash\_return <= 1;

end

else if(cash\_in == 3)

begin

next\_state <= A;

purchase <= 1;

cash\_return <= 3;

end

endcase

endmodule

**Simulation Report:**

**Graphical user interface, application, table, Excel

Description automatically generated**

**Explanation:**

Here,

5tk = 1; 10tk = 2; 20tk = 3.

In our vending machine firstly, we put 20tk (showing as 3 in cash\_in) for the 15tk product. So, at the next clock cycle our purchase got successful (showing as 1 in purchase) and we got 5tk back (showing as 1 in cash\_return). After sometimes we put a 10tk (showing as 2 in cash\_in) and then a 5 tk (showing as 1 in cash\_in). At the next clock cycle, we get our 15tk product (showing as 1 in purchase). Then using 3 clock cycles we put 5tk total 3 times (showing as 1 in cash\_in). After that our purchase got successful (showing as 1 in purchase). Finally, we put two 10tk (showing as 2 in cash\_in) and in return we get 5tk back (showing as 1 in cash\_return) and the product (showing as 1 in purchase). Therefore, our vending machine is working correctly.