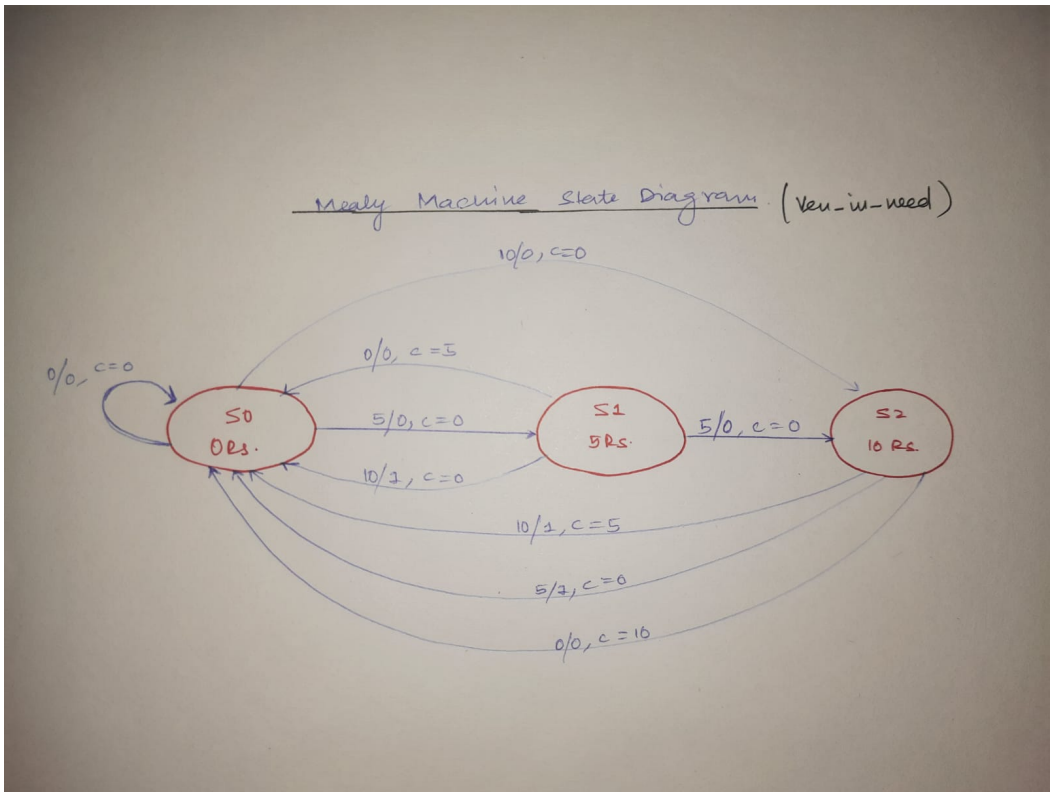
VEN IN NEED

A vending machine dispenses a product to the users based on the amount of money inserted and selection of the product. I have assumed that in my world there exists only Rs.5 and Rs.10 coins the vending machine produces an item of cost Rs.15.

STATE DIAGRAM



DESIGN

```
// We will assume that there are only 5rs and 10rs currency in the system
// We define 3 states 0 for Rs.0 , 01 for Rs.5, 10 for Rs.10
// The price of the product in the vending machine is Rs.15
module ven_in_need(in , rst , clk, out , change);
  input [1:0]in;
  input rst;
  input clk;
  output reg out;
  output reg [1:0]change;

  parameter S0 = 2'b00;
  parameter S1 = 2'b01;
  parameter S2 = 2'b10;
```

```
reg [1:0] current , next;// for current and next state

always@(posedge clk)
  begin
    if(rst)
      begin
        current = S0;
        next = S0;
        change = 2'b00;
      end
    else
      current = next ;

    case(current)
      S0:
        if (in==0)
        begin
          next = S0;
          change = 2'b00;
          out =1'b0;
          end
        else if (in== 2'b01)
        begin
          next = S1;
          change = 2'b00;
          out = 1'b0;
          end
        else
        begin
        next = S2;
        change = 2'b00;
        out = 1'b0;
        end

      S1:
        if (in==0)
        begin
          next = S0;
          change = 2'b01;
          out = 1'b0;
          end
        else if (in== 2'b01)
        begin
```

```
          next = S2;
          change = 2'b00;
          out = 1'b0;
          end
        else
        begin
          next = S0;
          change = 2'b00;
          out = 1'b1;
          end

      S2:
        if (in==2'b00)
        begin
          next = S0;
          change = 2'b10;
          out = 1'b0;
          end
        else if (in== 2'b01)
        begin
          next = S2;
          change = 2'b00;
          out = 1'b1;
          end
        else
        begin
          next = S0;
          change = 2'b01;
          out = 1'b1;
          end
      default:
        begin
          next=S0;
          change=2'b00;
          out=1'b0;
        end
    endcase
  end
endmodule
```

TESTBENCH

```
module ven_in_need_tb;

    reg clk;
    reg [1:0] in;
    reg rst;
    wire out;
    wire [1:0] change;

    ven_in_need ven_in_need_tb(.clk(clk), .rst(rst) , .in(in) , .out(out), .change(change));
    initial
        begin
            $dumpfile ("ven_in_need.vcd");
            $dumpvars(0,ven_in_need_tb);
            rst =1;
            clk = 0;

            #5
            rst = 0;
            in = 1;
            #10
            in = 1;
            #15
            in = 1;
            #20
            $finish;
        end
    always @(posedge clk)
        initial
            begin
                forever clk = ~clk;
            end
endmodule
```