Vending Machine Controller

Abstract :

The vending machine controller acts as the central intelligence of vending machines, overseeing payment processing, inventory management, product dispensing, and user interaction. Responsible for interfacing with hardware components such as coin mechanisms, bill validators, and display panels, it facilitates the vending process by accepting various forms of payment, monitoring stock levels, and dispensing products upon user selection. Through keypad or touch screen interfaces, it interprets user inputs, provides real-time feedback, and guides customers through the vending process. Additionally, the controller implements safety and security features to safeguard the machine and its contents, including encryption of payment data, tamper-proof enclosures, and monitoring systems. Overall, the vending machine controller ensures a seamless and secure vending experience, making it an indispensable component of modern vending solutions.

Keywords :

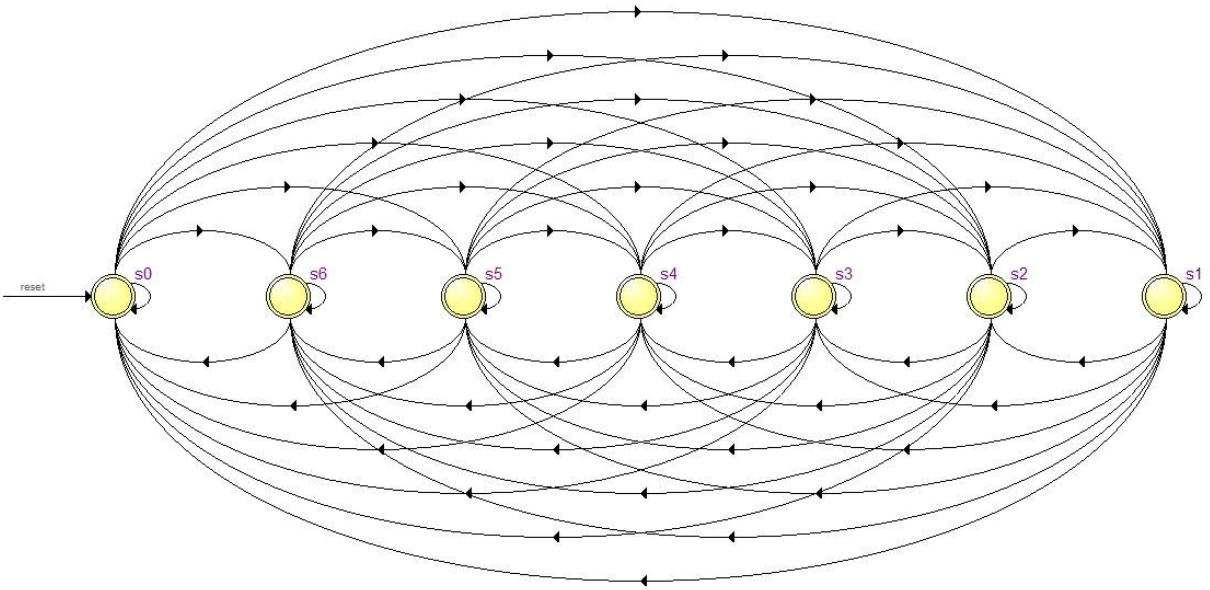
Vending machine, Payment processing, Product dispensing, User interface, Coin mechanism, Touch screen.

Introduction :

The vending machine controller is the core component responsible for managing and coordinating the various functions of a vending machine. It serves as the brain of the vending system, overseeing processes such as payment acceptance, inventory monitoring, product dispensing, and user interaction. This sophisticated electronic device interfaces with the hardware components of the vending machine, including coin and bill acceptors, product dispensers, keypad or touch screen interfaces, and display panels. By orchestrating these components, the controller ensures a seamless vending experience for customers while providing essential features such as real-time feedback, inventory management, and security measures to safeguard the machine and its contents. With its advanced capabilities and robust functionality, the vending machine controller plays a pivotal role in the operation and efficiency of modern vending machines.

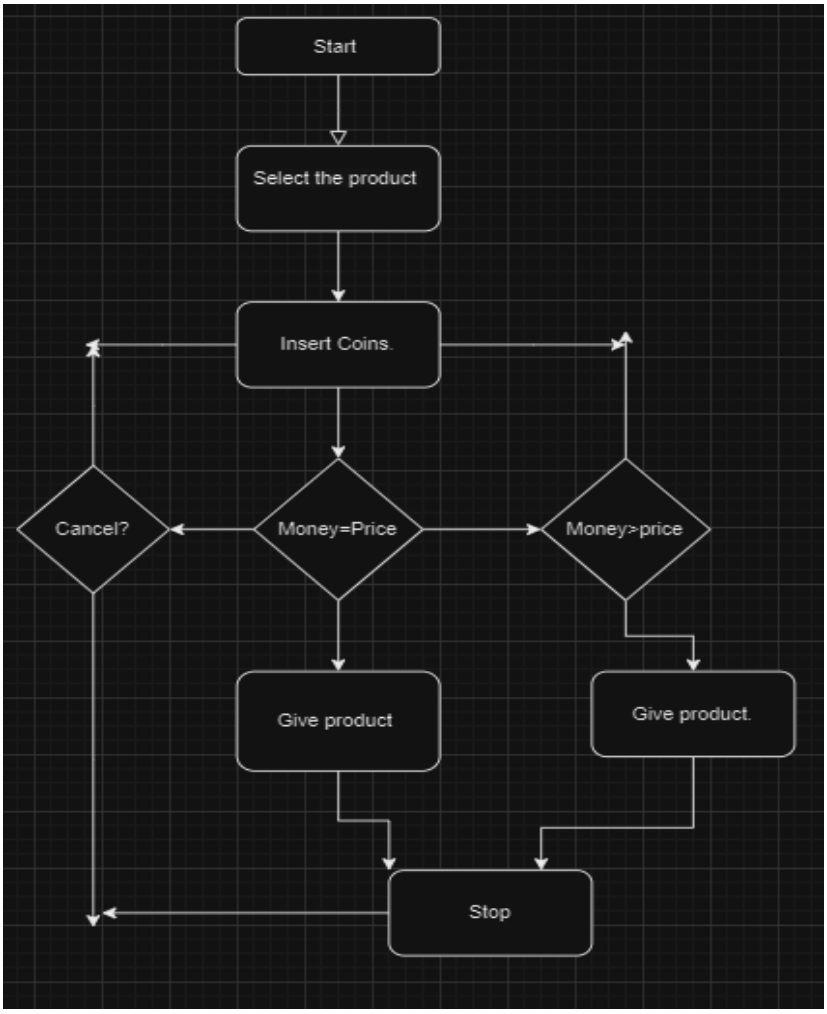
Vending machine :

State diagram:



- Here there are total four products.
- S0,S1,S1,S3,S4,S5,S6, represent the state in the form of different coin inputs.
- We can enter any desired amount between 0 to 35 in multiple of 5 only .
- The output will be 1 when the desired amount is received.

Flow chart

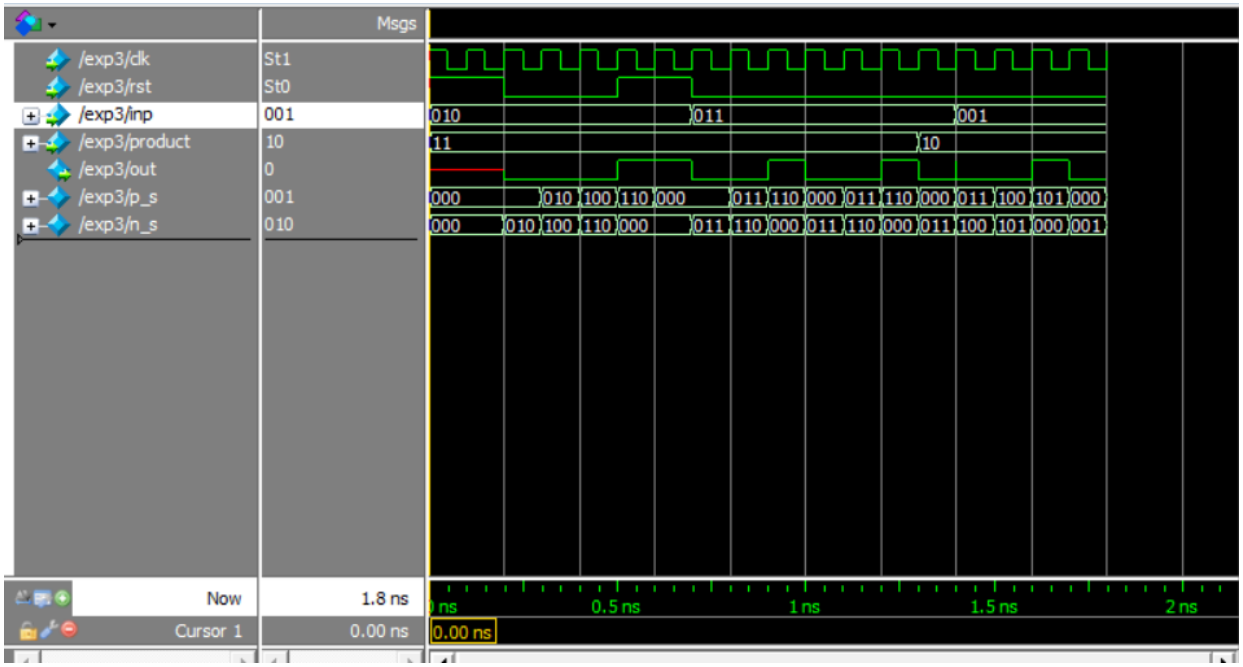


Flow Of Code :

1. **Begin :** Start of the Program
2. **Initialize Parameters** : Initialize the parameters as input and output and the parameters that has to be taken.
3. **State assignment**: Assign the states corresponding to the different inputs of money.

4. **Construction of FSM**:
   Write each and every possible state from the given state until the output is received.
5. **End** : Terminate the Program

Waveform :



Conclusion :

In summary, the vending machine controller acts as the central control unit of vending machines, managing payment transactions, product inventory, and user interactions. It interfaces with hardware components to ensure smooth operation and guides customers through the vending process. With advanced features and security measures, it safeguards the machine and enhances the overall vending experience, making it an essential component of modern vending solution

References :

1. Samir Palnitkar: A guide to Verilog hdl.

Appendix :

Main Code :

```
// To design a vending machine for purchasing product.

// Here we have taken 4 products:
// product 0 -- price-15
// product 1 -- price-25
// product 2 -- price-30
// product 3 -- price-35

// Now we will define the different states

// 000 -- price 0  //state 0
//001 --price 5    //state 1
//010 --price-10   //state 2
//011 --price-15   //state 3
//100 --price-20   //state 4
//101 --price-25   //state 5
//110 --price-30   //state 6

module exp3(clk,rst,out,inp,product);


// Declaration of inputs.
input clk,rst;
input [2:0] inp;
input [1:0] product;



//Declaration of outputs.
output reg out;



//Declaration of state in the form of binary assignments.
parameter s0=3'b000;//0
parameter s1=3'b001;//5
parameter s2=3'b010;//10
parameter s3=3'b011;//15
parameter s4=3'b100;//20
parameter s5=3'b101;//25
parameter s6=3'b110;//30
parameter s7=3'b111;//35


// Declaration of present and next state.
```

```
reg [2:0] p_s,n_s;

// beginning of the main code.
always@(posedge clk)
begin
if(rst)
begin
p_s<=s0;
end
else p_s<=n_s;
end



always@(*)
begin
if(rst) n_s<=s0;
else
begin
case(p_s)
s0:
if(product==2'b00)
begin
if(inp==3'b000)
begin
n_s<=s0;
out<=0;
end
else if(inp==3'b001)
begin
n_s<=s1;
out<=0;
end
else if(inp==3'b010)
begin
n_s<=s2;
out<=0;
end
else if(inp==3'b011 ||inp==3'b100 || inp==3'b101 || inp ==3'b110 || inp==3'b111)
begin
n_s<=s0;
out<=1;
end
end



else if(product==2'b01) // price 25

begin
```

```
if(inp==3'b000)
begin
n_s<=s0;
out<=0;
end
else if(inp==3'b001)
begin
n_s<=s1;
out<=0;
end
else if(inp==3'b010)
begin
n_s<=s2;
out<=0;
end
else if(inp==3'b011)
begin
n_s<=s3;
out<=0;
end
else if(inp==3'b100)
begin
n_s<=s4;
out<=0;
end
else if(inp==3'b101|| inp==3'b110 || inp==3'b111)
begin
n_s<=s0;
out<=1;
end
end

else if(product==2'b10)//price 30
begin
if(inp==3'b000)
begin
n_s<=s0;
out<=0;
end
else if(inp==3'b001)
begin
n_s<=s1;
out<=0;
end
else if(inp==3'b010)
begin
n_s<=s2;
out<=0;
end
else if(inp==3'b011)
begin
n_s<=s3;
out<=0;
```

```
end
else if(inp==3'b100)
begin
n_s<=s4;
out<=0;
end
else if(inp==3'b101)
begin
n_s<=s5;
out<=0;
end
else if(inp==3'b110 || inp==3'b111)
begin
n_s<=s0;
out<=1;
end
end




else if(product==2'b11)
begin
if(inp==3'b000)
begin
n_s<=s0;
out<=0;
end
else if(inp==3'b001)
begin
n_s<=s1;
out<=0;
end
else if(inp==3'b010)
begin
n_s<=s2;
out<=0;
end
else if(inp==3'b011)
begin
n_s<=s3;
out<=0;
end
else if(inp==3'b100)
begin
n_s<=s4;
out<=0;
end
else if(inp==3'b101)
begin
n_s<=s5;
out<=0;
end
else if(inp==3'b110)
```

```
begin
n_s<=s6;
out<=0;
end
else if(inp==3'b111)
begin
n_s<=s0;
out<=1;
end
end

s1:
if(product==2'b00)
begin
if(inp==3'b000)
begin
n_s<=s1;
out<=0;
end
else if(inp==3'b001)
begin
 n_s<=s2;
 out<=0;
end
else if(inp==3'b010||inp==3'b011||inp==3'b100||inp==3'b101||inp==3'b110||inp==3'b111)
begin
 n_s<=s0;
 out<=1;
end
end


else if(product==2'b01)
begin
if(inp==3'b000)
begin
n_s<=s1;
out<=0;
end
else if(inp==3'b001)
begin
n_s<=s2;
out<=0;
end
else if(inp==3'b010)
begin
n_s<=s3;
out<=0;
end
else if(inp==3'b011)
begin
n_s<=s4;
out<=0;
end
else if(inp==3'b100 || inp==3'b101 || inp==3'b110 ||inp==3'b111 )
```

```
begin
n_s<=s0;
out<=1;
end
end


else if(product==2'b10)
begin
if(inp==3'b000)
begin
n_s<=s1;
out<=0;
end
else if(inp==3'b001)
begin
n_s<=s2;
out<=0;
end
else if(inp==3'b010)
begin
n_s<=s3;
out<=0;
end
else if(inp==3'b011)
begin
n_s<=s4;
out<=0;
end
else if(inp==3'b100)
begin
n_s<=s5;
out<=0;
end
else if(inp==3'b101 || inp==3'b110 || inp==3'b111)
begin
n_s<=s0;
out<=1;
end
end

else if(product==2'b11)
begin
if(inp==3'b000)
begin
n_s<=s1;
out<=0;
end
else if(inp==3'b001)
begin
n_s<=s2;
out<=0;
end
else if(inp==3'b010)
begin
```

```
n_s<=s3;
out<=0;
end
else if(inp==3'b011)
begin
n_s<=s4;
out<=0;
end
else if(inp==3'b100)
begin
n_s<=s5;
out<=0;
end
else if(inp==3'b101)
begin
n_s<=s6;
out<=0;
end
else if(inp==3'b110 || inp==3'b111)
begin
n_s<=s0;
out<=1;
end
end



s2:

if(product==2'b00)
begin
if(inp==3'b000)
begin
n_s<=s2;
out<=0;
end
else if(inp==3'b001 || inp==3'b010 || inp==3'b011 || inp==3'b100 || inp==3'b101 || inp==3'b110 ||
inp==3'b111)
begin
n_s<=s0;
out<=1;
end
end


else if(product==2'b01)
begin
if(inp==3'b000)
begin
n_s<=s2;
out<=0;
end
else if(inp==3'b001)
begin
n_s<=s3;
```

```
out<=0;
end
else if(inp==3'b010)
begin
n_s<=s4;
out<=0;
end
else if(inp==3'b011 || inp==3'b100 || inp==3'b101 || inp==3'b110 || inp==3'b111)
begin
n_s<=s0;
out<=1;
end
end

else if(product==2'b10)
begin
if(inp==3'b000)
begin
n_s<=s2;
out<=0;
end
else if(inp==3'b001)
begin
n_s<=s3;
out<=0;
end
else if(inp==3'b010)
begin
n_s<=s4;
out<=0;
end
else if(inp==3'b011)
begin
n_s<=s5;
out<=0;
end
else if(inp==3'b100 || inp==3'b101 || inp==3'b110 || inp==3'b111)
begin
n_s<=s0;
out<=1;
end
end


else if(product==2'b11)
begin
if(inp==3'b000)
begin
n_s<=s2;
out<=0;
end
else if(inp==3'b001)
begin
n_s<=s3;
out<=0;
```

```
end
else if(inp==3'b010)
begin
n_s<=s4;
out<=0;
end
else if(inp==3'b011)
begin
n_s<=s5;
out<=0;
end
else if(inp==3'b100)
begin
n_s<=s6;
out<=0;
end
else if(inp==3'b101 || inp==3'b110 || inp==3'b111)
begin
n_s<=s0;
out<=1;
end
end
```

```
s3:
if(product==2'b00)
begin
if(inp==3'b000 || inp==3'b001 || inp==3'b010 || inp==3'b011 ||inp==3'b100
||inp==3'b101||inp==3'b110||inp==3'b111)
begin
n_s<=s0;
out<=1;
end
end
```

```
else if(product==2'b01)
begin
if(inp==3'b000)
begin
n_s<=s3;
out<=0;
end
else if(inp==3'b001)
begin
n_s<=s4;
out<=0;
end
else if(inp==3'b010|| inp==3'b011 ||inp==3'b100 ||inp==3'b101||inp==3'b110||inp==3'b111)
begin
n_s<=s0;
out<=1;
end
```

```
end

else if(product==2'b10)
begin
if(inp==3'b000)
begin
n_s<=s3;
out<=0;
end
else if(inp==3'b001)
begin
n_s<=s4;
out<=0;
end
else if(inp==3'b010)
begin
n_s<=s5;
out<=0;
end
else if(inp==3'b011||inp==3'b100 ||inp==3'b101||inp==3'b110||inp==3'b111)
begin
n_s<=s0;
out<=1;
end
end

else if(product==2'b11)
begin
if(inp==3'b000)
begin
n_s<=s3;
out<=0;
end
else if(inp==3'b001)
begin
n_s<=s4;
out<=0;
end
else if(inp==3'b010)
begin
n_s<=s5;
out<=0;
end
else if(inp==3'b011)
begin
n_s<=s6;
out<=0;
end
else if(inp==3'b100||inp==3'b101||inp==3'b110||inp==3'b111)
begin
n_s<=s0;
out<=1;
end
end
// BAKI
```

```
s4:
if(product==2'b00)
begin
if(inp==3'b000 || inp==3'b001 || inp==3'b010 || inp==3'b011 ||inp==3'b100
||inp==3'b101||inp==3'b110||inp==3'b111)
begin
n_s<=s0;
out<=1;
end
end
else if(product==2'b01)
begin
if(inp==3'b000)
begin
n_s<=s4;
out<=0;
end
else if(inp==3'b001 || inp==3'b010 || inp==3'b011 ||inp==3'b100
||inp==3'b101||inp==3'b110||inp==3'b111)
begin
n_s<=s0;
out<=1;
end
end
else if(product==2'b10)
begin
if(inp==3'b000)
begin
n_s<=s4;
out<=0;
end
else if(inp==3'b001)
begin
n_s<=s5;
out<=0;
end
else if(inp==3'b010 || inp==3'b011 ||inp==3'b100 ||inp==3'b101||inp==3'b110||inp==3'b111)
begin
n_s<=s0;
out<=1;
end
end


else if(product==2'b11)
begin
if(inp==3'b000)
begin
n_s<=s4;
out<=0;
end
else if(inp==3'b001)
begin
n_s<=s5;
```

```
out<=0;
end
else if(inp==3'b010)
begin
n_s<=s6;
out<=0;
end
else if(inp==3'b011 ||inp==3'b100 ||inp==3'b101||inp==3'b110||inp==3'b111)
begin
n_s<=s0;
out<=1;
end
end


s5:
if(product==2'b00)
begin
if(inp==3'b000 || inp==3'b001 || inp==3'b010 || inp==3'b011 ||inp==3'b100
||inp==3'b101||inp==3'b110||inp==3'b111)
begin
n_s<=s0;
out<=1;
end
end
else if(product==2'b01)
begin
if(inp==3'b000 || inp==3'b001 || inp==3'b010 || inp==3'b011 ||inp==3'b100
||inp==3'b101||inp==3'b110||inp==3'b111)
begin
n_s<=s0;
out<=1;
end
end

else if(product==2'b10)
begin
if(inp==3'b000)
begin
n_s<=s5;
out<=0;
end
else if(inp==3'b001 || inp==3'b010 || inp==3'b011 ||inp==3'b100
||inp==3'b101||inp==3'b110||inp==3'b111)
begin
n_s<=s0;
out<=1;
end
end

else if(product==2'b11)
begin
 if(inp==3'b000)
begin
n_s<=s5;
```

```
out<=0;
end
else if(inp==3'b001)
begin
n_s<=s6;
out<=0;
end
else if(inp==3'b010 || inp==3'b011 ||inp==3'b100 ||inp==3'b101||inp==3'b110||inp==3'b111)
begin
n_s<=s0;
out<=1;
end
end


s6:
if(product==2'b00)
begin
if(inp==3'b000 || inp==3'b001 || inp==3'b010 || inp==3'b011 ||inp==3'b100
||inp==3'b101||inp==3'b110||inp==3'b111)
begin
n_s<=s0;
out<=1;
end
end
else if(product==2'b01)
begin
if(inp==3'b000 || inp==3'b001 || inp==3'b010 || inp==3'b011 ||inp==3'b100
||inp==3'b101||inp==3'b110||inp==3'b111)
begin
n_s<=s0;
out<=1;
end
end

else if(product==2'b10)
begin
if(inp==3'b000 || inp==3'b001 || inp==3'b010 || inp==3'b011 ||inp==3'b100
||inp==3'b101||inp==3'b110||inp==3'b111)
begin
n_s<=s0;
out<=1;
end
end
else if(product==2'b11)
begin
if(inp==3'b000)
begin
n_s<=s6;
out<=0;
end
else if(inp==3'b001 || inp==3'b010 || inp==3'b011 ||inp==3'b100
||inp==3'b101||inp==3'b110||inp==3'b111)
begin
n_s<=s0;
```

```
out<=1;
end
end

endcase
end
end
endmodule
```