

NATIONAL INSTITUTE OF TECHNOLOGY

MHRD, Govt. of India

WARANGAL– 506004, TELANGANA, INDIA



STRUCTURED DIGITAL SYSTEM DESIGN

LAB PROJRCT

VENDING MACHINE (COLD-DRINK DISPENSER)

Submitted to:

Dr. R. Rakesh Kumar

Submitted By – Kaushal Kumar Kumawat

M.Sc. Tech Engineering Physics

2nd year Semester – 4

Roll No. – 21PHC1R15

Mealy Machine

format = Input/output, change

```
graph LR; S0((S0)) -- "0/0,0" --> S0; S0 -- "1/0,0" --> S1((S1)); S1 -- "1/1,0" --> S0; S1 -- "2/1,1" --> S0; S1 -- "5/1,4" --> S0;
```

[illegible]

```

module Colddrink_Dispenser(
input clock, reset,
input [2:0] input_coin,
input [1:0] sel_colddrink,
output reg out,coke, pepsi, maaza,
output reg [2:0] change,
output reg [2:0] balance);

// Defining the parameters of the cold drink dispenser
// -> Parameters for the money in INR
parameter [2:0] INR_0 = 3'b000;
parameter [2:0] INR_1 = 3'b001;
parameter [2:0] INR_2 = 3'b010;
parameter [2:0] INR_3 = 3'b011;
parameter [2:0] INR_4 = 3'b100;
parameter [2:0] INR_5 = 3'b101;
parameter [2:0] INR_6 = 3'b110;

// -> Parameters for select cold drink
parameter [1:0] NOTHING = 2'b00;
parameter [1:0] sel_coke = 2'b01;
parameter [1:0] sel_pepsi = 2'b10;
parameter [1:0] sel_maaza = 2'b11;

// -> Parameters for state machine
reg pre_state, nxt_state;

parameter state_0 = 1'b0;
parameter state_1 = 1'b1;

```

```
// Defining the initial conditions
```

```
initial
```

```
begin
```

```
    pre_state <= state_0;
```

```
    nxt_state <= state_0;
```

```
    change <= INR_0;
```

```
    balance <=0;
```

```
    out <= 0;
```

```
    coke <= 0;
```

```
    pepsi <= 0;
```

```
    maaza <= 0;
```

```
end
```

```
// Combinational block for defining the next state of the vending machine
```

```
always @(input_coin, pre_state)
```

```
begin
```

```
    case(pre_state)
```

```
        state_0:
```

```
            if(input_coin == INR_0)
```

```
                nxt_state <= state_1;
```

```
            else if(input_coin == INR_2 && input_coin == INR_5)
```

```
                nxt_state <= state_0;
```

```
        state_1:
```

```
            if(input_coin == INR_1 && input_coin == INR_2 && input_coin == INR_5)
```

```
                nxt_state <= state_0;
```

```
    endcase
```

```
end
```

```
// Defining the nxt_state update
```

```
always @(negedge clock or posedge reset)
```

```
begin
    if(reset)
        pre_state <= state_0;
    else
        pre_state <= nxt_state;
    end
```

```
// Defining the output combination logic
always @(input_coin, sel_colddrink, pre_state)
```

```
begin
    case(pre_state)
        state_0:
            if(input_coin == INR_0)
                begin
                    balance = balance + INR_0;
                end
            else if(input_coin == INR_1)
                begin
                    balance = balance + INR_1;
                end
            else if(input_coin == INR_2)
                begin
                    balance = balance + INR_2;
                end
            else if(input_coin == INR_5)
                begin
                    balance = balance + INR_5;
                end
    end
```

```
state_1:
```

```

    if(input_coin == INR_0)
    begin
        balance = balance + INR_0;
    end
    else if(input_coin == INR_1)
    begin
        balance = balance + INR_1;
    end
    else if(input_coin == INR_2)
    begin
        balance = balance + INR_2;
    end
    else if(input_coin == INR_5)
    begin
        balance = balance + INR_5;
    end
endcase
end

// Defining the type of coldrink based on selected cold drink and balance.
always @(*)
begin
    case(balance)
        INR_0:
        begin
            out = 0;
            change = 0;
        end
        INR_1:
        begin
            out = 0;
            change = 0;
        end
    endcase
end

```

```
end
INR_2:
begin
    out = 1;
    change = balance - INR_2;
end
INR_3:
begin
    out = 1;
    change = balance - INR_2;
end
INR_4:
begin
    out = 1;
    change = balance - INR_2;
end
INR_5:
begin
    out = 1;
    change = balance - INR_2;
end
INR_6:
begin
    out = 1;
    change = balance - INR_2;
end
endcase
if(sel_colddrink == sel_coke && out == 1)
    coke <= 1;
else if(sel_colddrink == sel_pepsi && out == 1)
    pepsi <= 1;
```

```
else if(sel_colddrink == sel_maaza && out == 1)
    maaza <= 1;

if(out == 1)
fork
    #10 pre_state <= state_0;
    #10 nxt_state <= state_0;
    #10 change <= INR_0;
    #10 balance <=0;
    #10 out <= 0;
    #10 coke <= 0;
    #10 pepsi <= 0;
    #10 maaza <= 0;
join
end
endmodule
```


TEST BENCH

```
`timescale 1ns / 1ps
```

```
////////////////////////////////////////////////////////////////
```

```
// Company:
```

```
// Engineer:
```

```
//
```

```
// Create Date: 16.04.2023 18:15:07
```

```
// Design Name:
```

```
// Module Name: Colddrink_Dispenser_tb
```

```
// Name - Kaushal Kumar Kumawat
```

```
// Roll No. - 21PHC1R15
```

```
////////////////////////////////////////////////////////////////
```

```
module Colddrink_Dispenser_tb();
```

```
reg clock, reset;
```

```
reg [2:0] input_coin;
```

```
reg [1:0] sel_colddrink;
```

```
wire out, coke, pepsi, maaza;
```

```
wire [2:0] change,balance;
```

```
Colddrink_Dispenser CD_1(clock, reset,input_coin,sel_colddrink,out,coke, pepsi,  
maaza,change,balance);
```

```
initial
```

```
begin
```

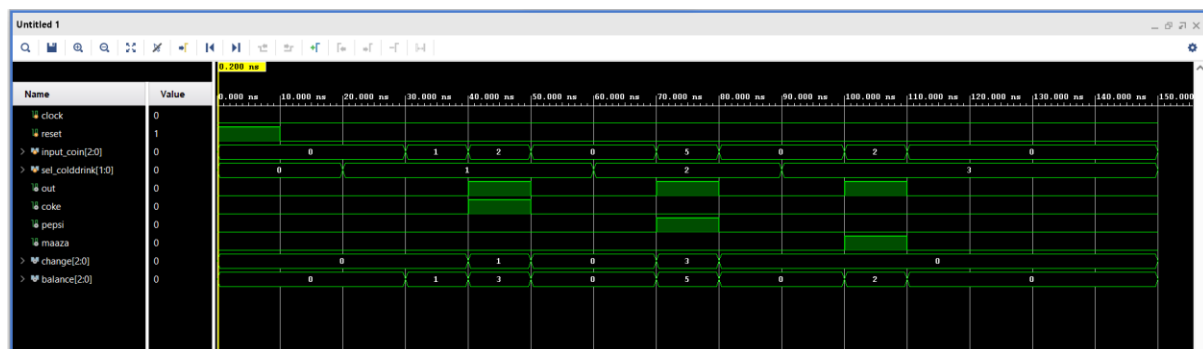
```
clock = 0;
```

```
reset = 1;
```

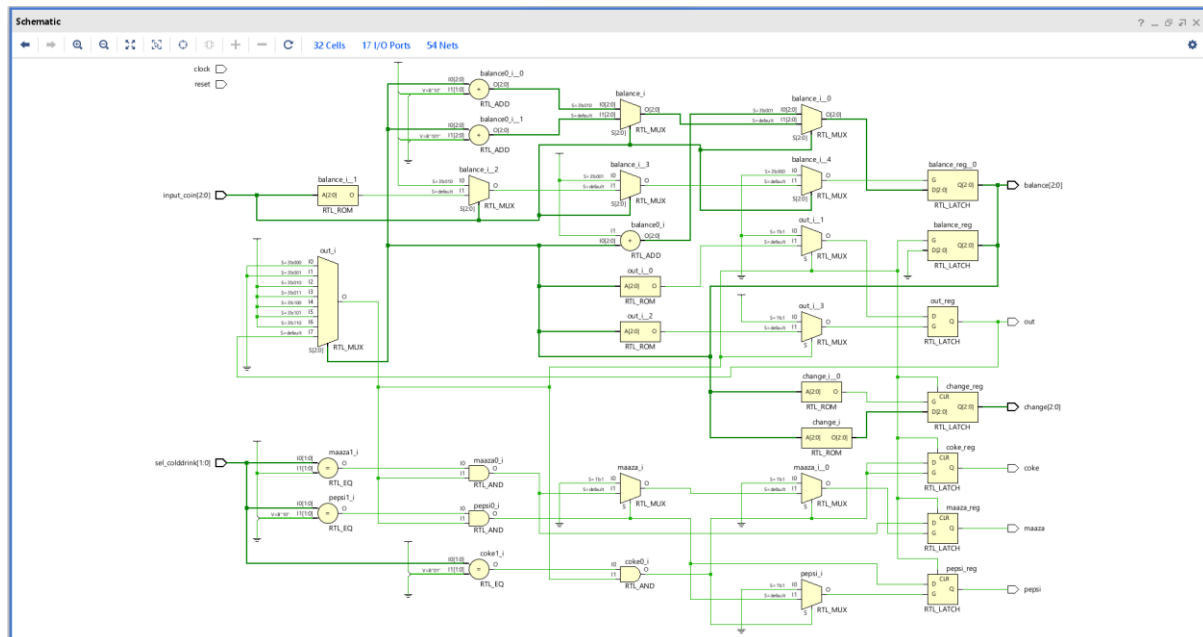
```
input_coin =0;
```

```
sel_colddrink = 0;
```

```
#10 reset = 0;
```



SCHEMATIC



RESULTS

This project is behav of FSM and based on the Mealy machine. There are only 2 states as per mealy state diagram. The code regarding the states is written and results are being verified in the testbench. The circuit is based on the input sensitivity thus the clock does not have any effect on the output and state update process. Thus, the results are being verified successfully.