

Design a vending machine in Quartus for a 15 Tk product

by

Ishraq Ahmed Esha

19301261

A Lab Report 4 submitted to the CSE460 VLSI Design

Course of Sec: 5

Brac University

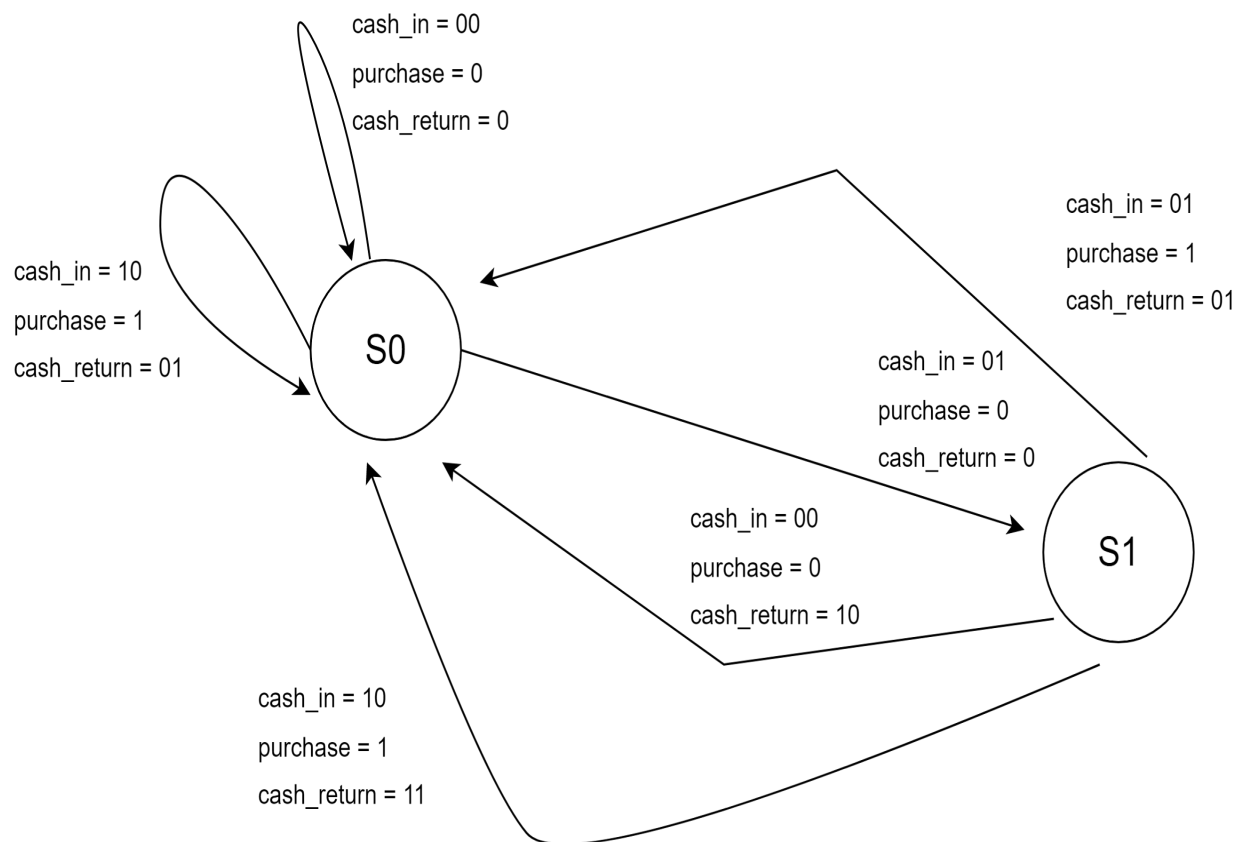
November 16, 2022

Problem Statement

You have to design a vending machine in Quartus for a 15 Tk product. User's money, returned money by the machine, and product bought condition is represented as cash_in (2-bit input), chg (output), and buy (1-bit output) respectively.

The vending machine can only accept three inputs: no money (cash_in = 00), Tk 10 (cash_in = 01), and Tk 20 (cash_in = 10). 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).

A) State Diagram



B)

There will be four types of changes (return) the machine will produce. These are 0 Tk (00), 5 Tk (01), 10 Tk (10) and 15 Tk (11). As the machine can accept only three inputs and from the state diagram we are observing that there can be four possible change types.

Two bits will be needed for the chg output to present the return money in the code. In the above discussion we can see that the four types of change value are represented with two bits.

C) State Assigned Table

Present State	Next State			Output		
	00	01	10	00	01	10
0	0	1	0	0	0	1
1	0	0	0	0	1	1

D) Code

```
module assignment_4(clk,reset,cash_in,cash_return,purchase,current_state,next_state);
    input clk,reset;
    input [0:1] cash_in;

    output reg purchase;
    output reg [1:0] cash_return;
    output reg current_state, next_state;

    parameter S0 = 1'b0, S1 = 1'b1;
    parameter [1:0] in_0tk = 2'b00,
    in_10tk = 2'b01,
    in_20tk = 2'b10;

    parameter [1:0] ret_0tk = 2'b00,
    ret_5tk = 2'b01,
    ret_10tk = 2'b10,
    ret_15tk = 2'b11;

    always @(posedge clk, posedge reset)

    begin
        if(reset == 1)
            begin
                next_state = S0;
                current_state = S0;
                purchase = 0;
                cash_return = ret_0tk;
            end
        else
            begin
                current_state = next_state;
                case(current_state)
                    S0:
                        begin
                            if(cash_in == in_0tk)
                                begin
                                    next_state = S0;
                                    purchase = 0;
                                end
                            else if(cash_in == in_10tk)
                                begin
                                    next_state = S1;
                                    purchase = 1;
                                end
                            else if(cash_in == in_20tk)
                                begin
                                    next_state = S1;
                                    purchase = 1;
                                end
                            else
                                next_state = S0;
                                purchase = 0;
                            end
                        end
                    S1:
                        begin
                            if(cash_in == in_0tk)
                                begin
                                    next_state = S0;
                                    purchase = 0;
                                end
                            else if(cash_in == in_10tk)
                                begin
                                    next_state = S0;
                                    purchase = 0;
                                end
                            else if(cash_in == in_20tk)
                                begin
                                    next_state = S0;
                                    purchase = 0;
                                end
                            else
                                next_state = S1;
                                purchase = 1;
                            end
                        end
                endcase
            end
        end
    end
```

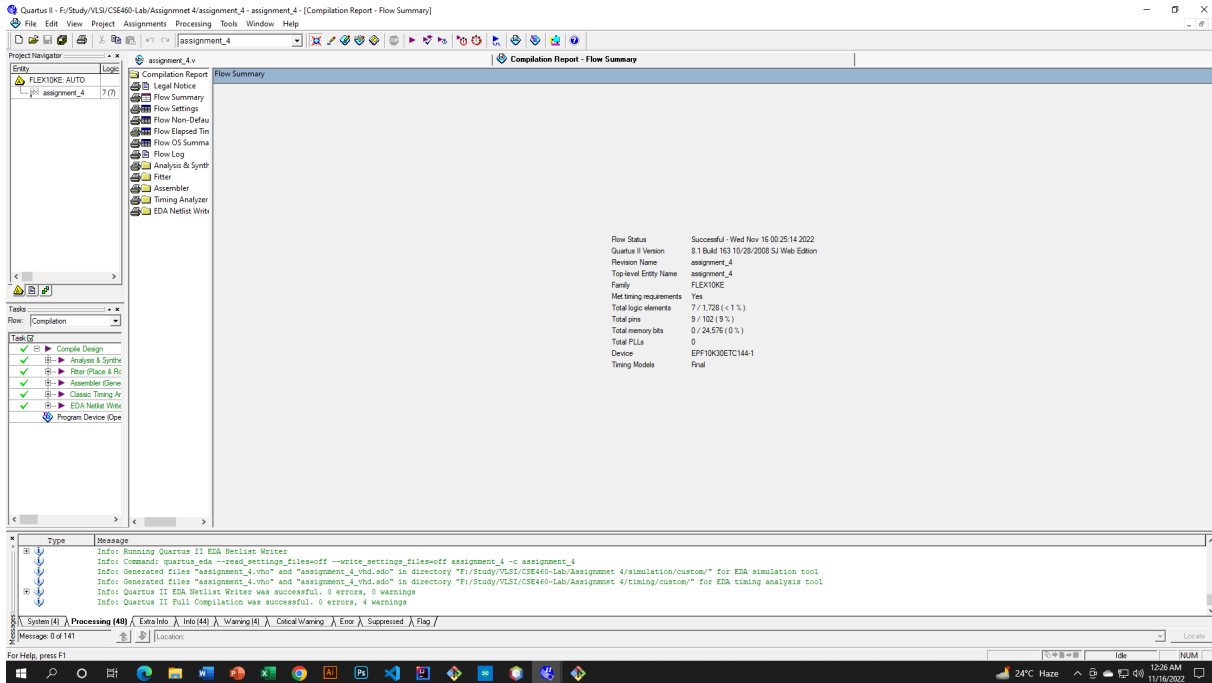
```

        cash_return = ret_0tk;
    end
    else if(cash_in == in_10tk)
    begin
        next_state = S1;
        purchase = 0;
        cash_return = ret_0tk;
    end
    else if(cash_in == in_20tk)
    begin
        next_state = S0;
        purchase = 1;
        cash_return = ret_5tk;
    end
end
S1:
begin
    if(cash_in == in_0tk)
    begin
        next_state = S0;
        purchase = 0;
        cash_return = ret_10tk;
    end
    else if(cash_in == in_10tk)
    begin
        next_state = S0;
        purchase = 1;
        cash_return = ret_5tk;
    end
    else if(cash_in == in_20tk)
    begin
        next_state = S0;
        purchase = 1;
        cash_return = ret_15tk;
    end
end
end
endcase
end
end
endmodule

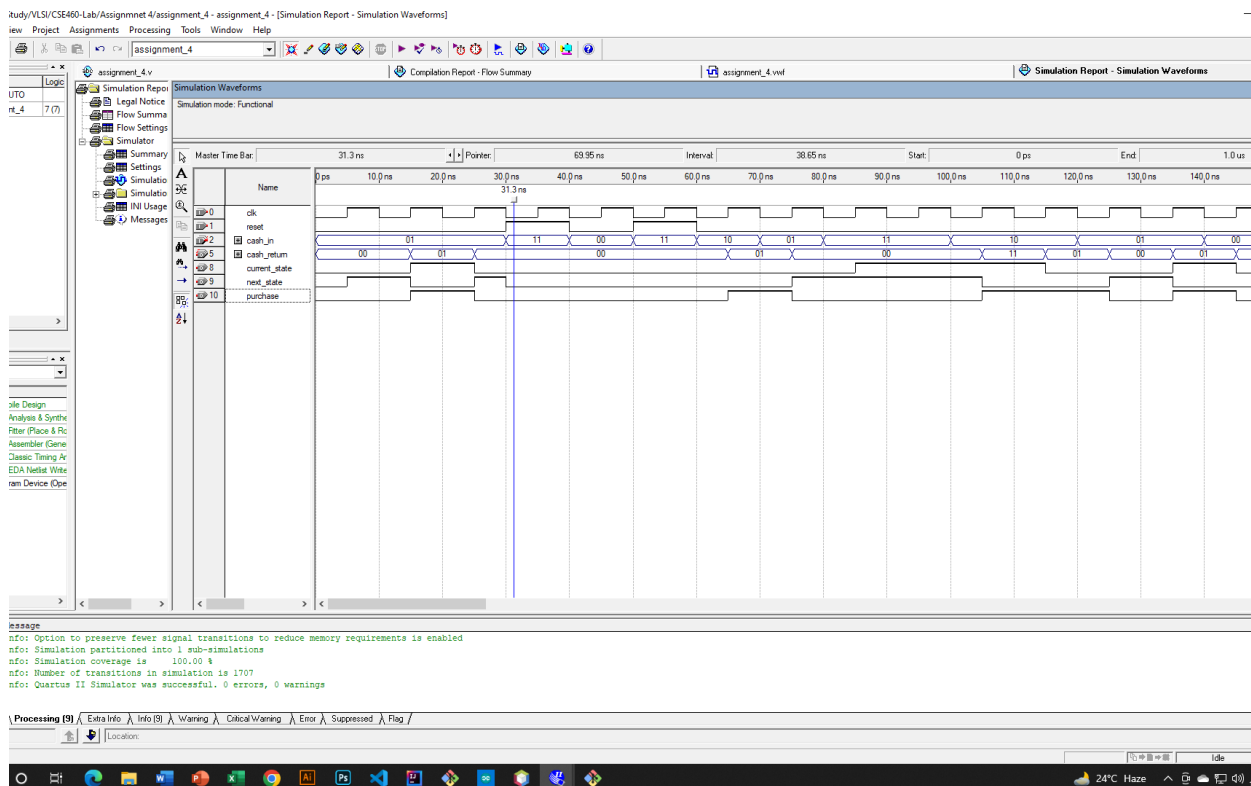
```

E) Simulation & Discussion

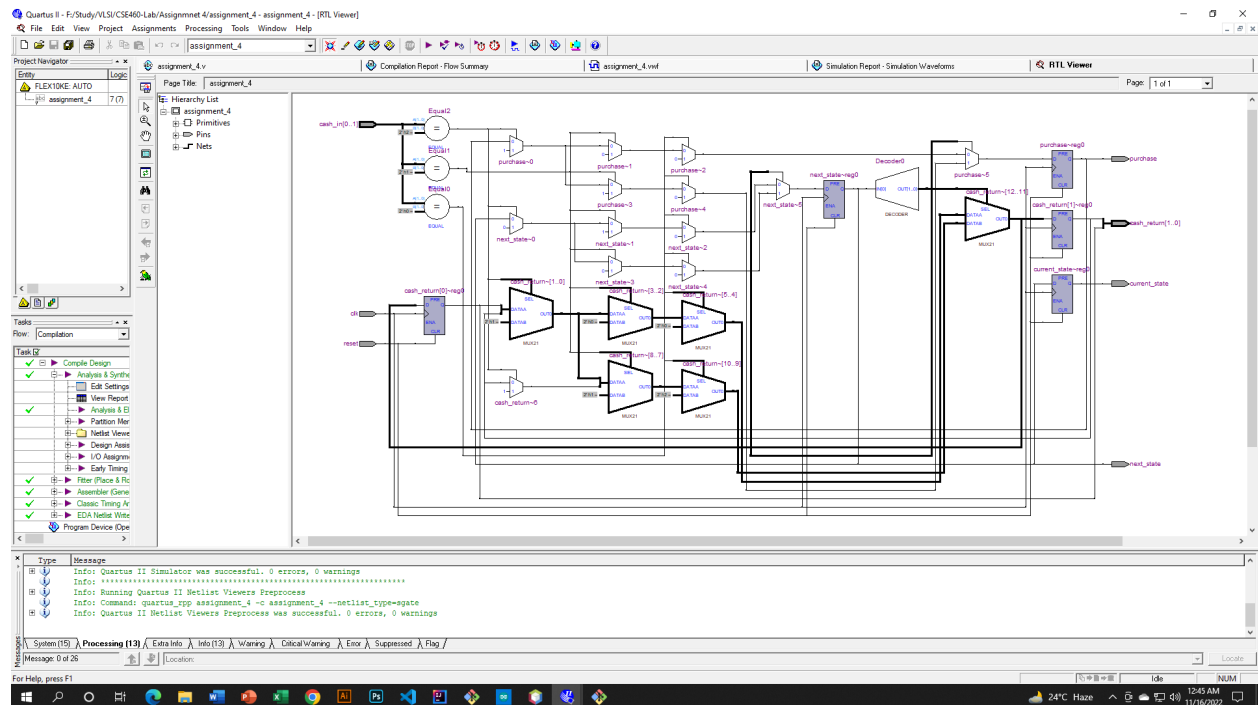
Compilation Report



Simulation Report



RTL View



Discussion about the working mechanism of 15 Tk product vending machine

If we observe the timing diagram then we can see that at the clock period from 65 ns to 75 ns that means the positive edge of the rising clock cycle the reset is off and we give the cash_in input as 10 which is 20 Tk that is higher than 15 Tk so, it matches our condition and purchase goes high from one rising clock edge to another rising clock edge. So, here purchase is complete as it's satisfied our condition and the timing diagram clearly shows this. Now if we look at the cash_return output then it shows us the binary value of that period is 01 which means 5 Tk. In addition we know that our machine is a 15 Tk product machine so as the user gives 20 Tk (10) and the purchase goes high signal then the return value of that high period is 01 which is 5 Tk. This full process is clearly visible with the timing diagram.