

**CSE 460**

**VLSI DESIGN**

**Lab Assignment 4**

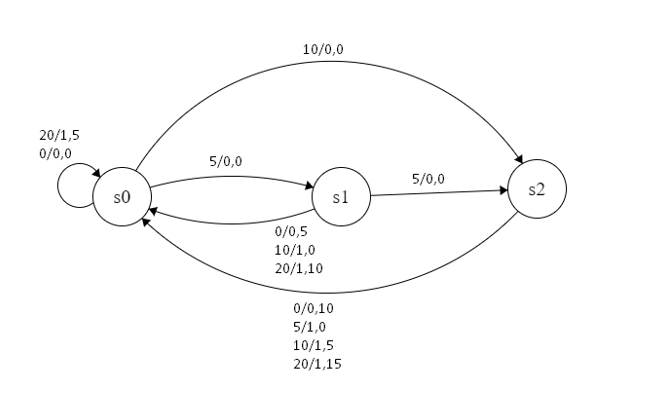
**Name: Dipak Debnath Arka**

**ID: 20301066**

**Sec: 09**

***Problem:*** 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 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 15 Tk, the machine immediately generates an output (buy=1), goes back to the initial state, and gives back the change (if required).

**1)State Diagram:**

****

**2) How many types of changes (return) will the machine produce? How many bit/bits should 'chg' output have to represent the returned money in the code?**

**Ans.** The machine will give back 0, 5, 10, and 15 Taka. For the returned money to be represented in the code, the chg output will require 2 bit. Binary representations of the return money are 0tk=00, 5tk=01, 10tk=10, and 15tk=11

**3) State-assigned table:**

|  |
| --- |
| **Present  state** **Next State Output Change** |

|  |
| --- |
| 0 5 10 20 0 5 10 20 0 5 10 2 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S0 | S0 | S1 | S2 | S0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 5 |
| S1 | S0 | S2 | S0 | S0 | 0 | 0 | 1 | 1 | 5 | 0 | 0 | 10 |
| S2 | S0 | S0 | S0 | S0 | 0 | 1 | 1 | 1 | 10 | 0 | 5 | 15 (11) |

**4) Verilog Code:**

module lab4(cash\_in, rst, clk, purchase, cash\_return);

/\*

cash\_in=0,5,10,20 TK

price = 15 TK

\*/

input [1:0] cash\_in;

input rst, clk;

output reg purchase;

output reg [1:0] cash\_return;

reg [1:0]current\_state,next\_state;

parameter state0 = 2'b00, state1 = 2'b01, state2 = 2'b10;

always @(current\_state, cash\_in)

case (current\_state)

state0: if (cash\_in == 2'b00)

begin

next\_state = state0;

purchase = 0;

cash\_return = 0;

end

else if (cash\_in ==2'b01)

begin

next\_state = state1;

purchase = 0;

cash\_return = 0;

end

else if (cash\_in ==2'b10)

begin

next\_state = state2;

purchase = 0;

cash\_return = 0;

end

else if (cash\_in ==2'b11)

begin

next\_state = state0;

purchase = 1;

cash\_return = 2'b01;

end

state1: if (cash\_in == 2'b00)

begin

next\_state = state0;

purchase = 0;

cash\_return = 2'b01;

end

else if (cash\_in ==2'b01)

begin

next\_state = state2;

purchase = 0;

cash\_return = 0;

end

else if (cash\_in ==2'b10)

begin

next\_state = state0;

purchase = 1;

cash\_return = 2'b00;

end

else if (cash\_in ==2'b11)

begin

next\_state = state0;

purchase = 1;

cash\_return = 2'b10;

end

state2: if (cash\_in == 2'b00)

begin

next\_state = state0;

purchase = 0;

cash\_return = 2'b10;

end

else if (cash\_in ==2'b01)

begin

next\_state = state0;

purchase = 1;

cash\_return = 0;

end

else if (cash\_in ==2'b10)

begin

next\_state = state0;

purchase = 1;

cash\_return = 2'b01;

end

else if (cash\_in ==2'b11)

begin

next\_state = state0;

purchase = 1;

cash\_return = 2'b11;

end

endcase

always @(posedge rst, posedge clk)

if (rst == 1)

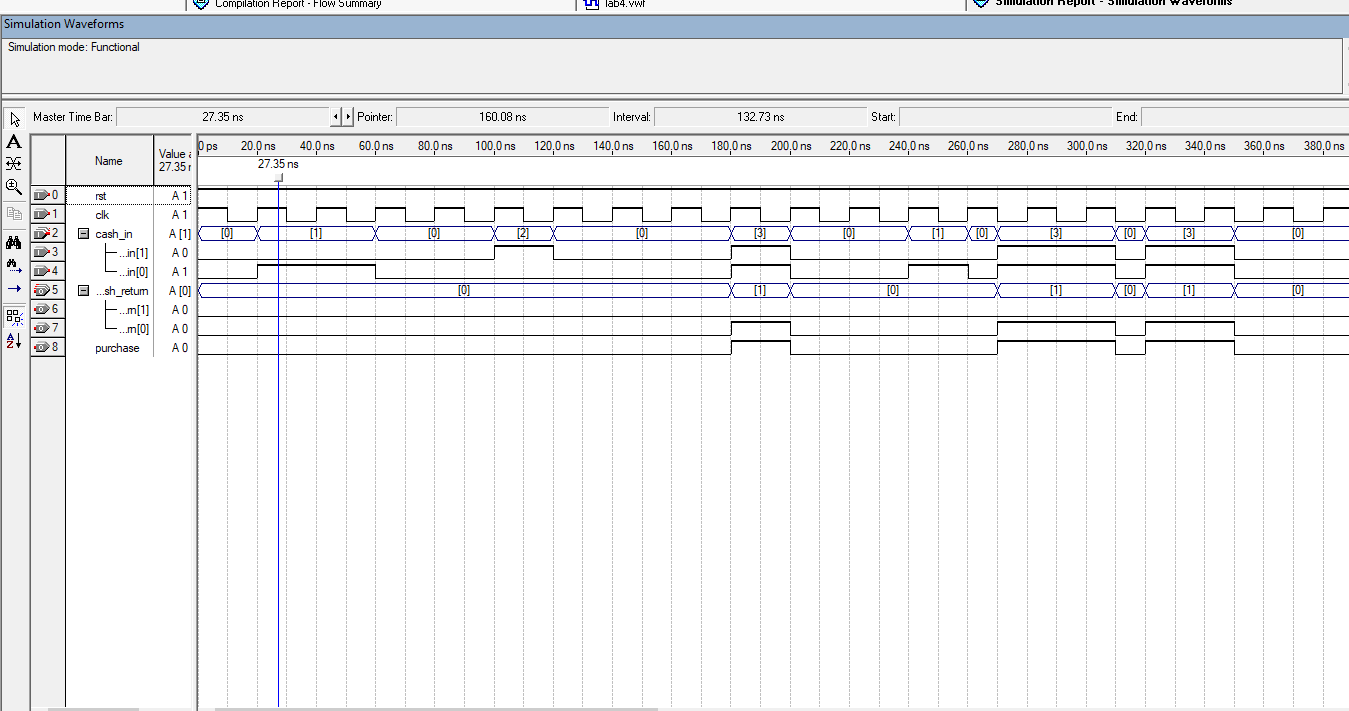
current\_state <= state0;

else

current\_state <= next\_state;

endmodule

**5) Simulation:**

****

**Explanation:** In this experiment, we created a vending machine that only accepts 0 tk, 5tk,10 tk, and 20 tk as input and will offer us a product worth 15 tk while giving us the additional money back. Here, if we insert 10TK, the machine will wait till we insert another 10TK,20TK or 5TK; otherwise, it will refund the 15/10TK/5TK. Nevertheless, if we add an additional  10 or 20 taka, it will save the first 15 taka and return us remaining taka 5TK or 10 TK or 20tk.