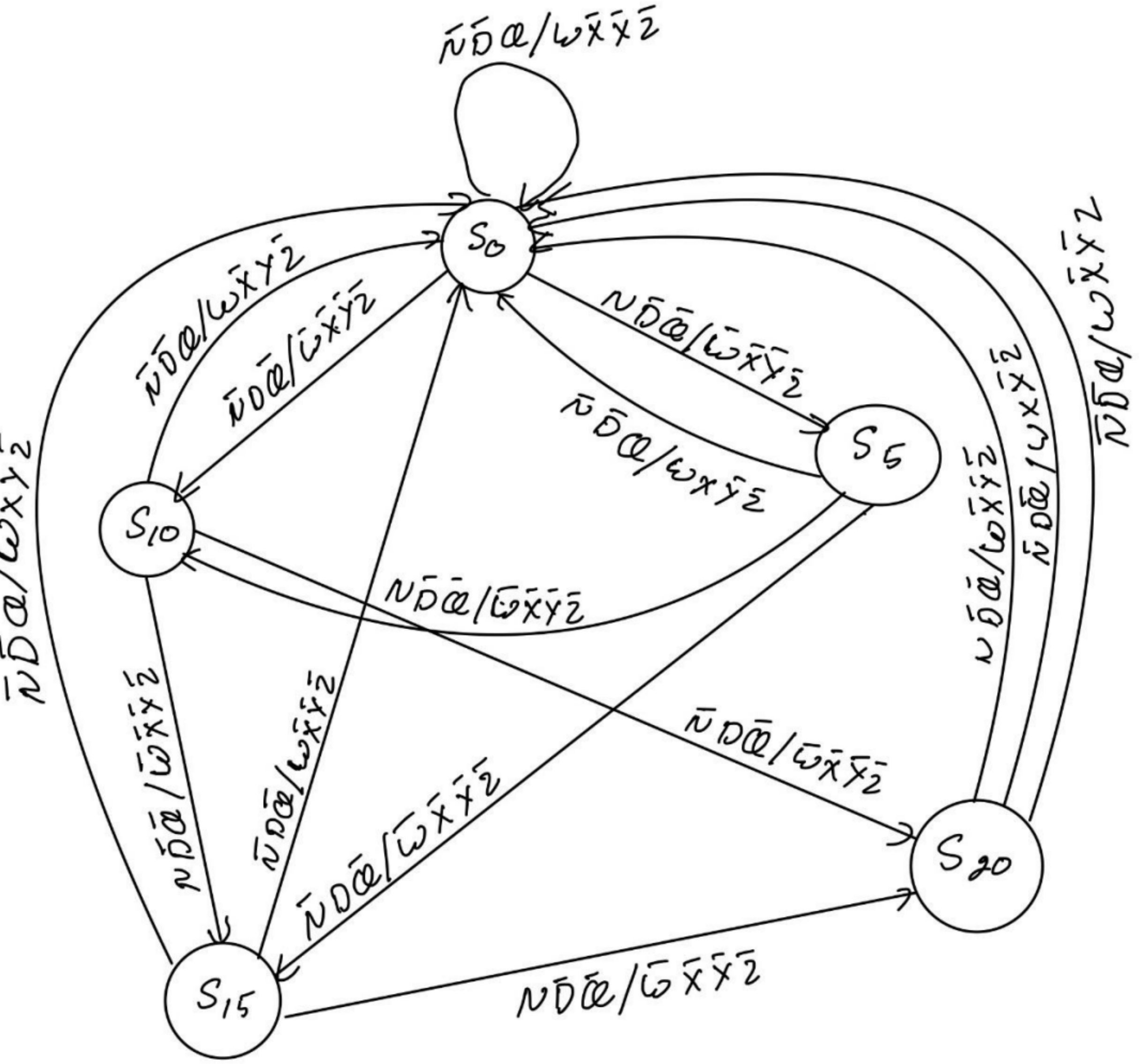
Name: Tanveer Singh Kahlon
Date: 11/07/2022
Lab 6: Vending Machine with testbench

1. State transition diagram:


S_0
S_5
S_{10}
S_{15}
S_{20}
$\bar{N}\bar{D}\bar{Q}/W\bar{X}\bar{Y}\bar{Z}$
$N\bar{D}\bar{Q}/\bar{W}\bar{X}\bar{Y}\bar{Z}$
$\bar{N}\bar{D}Q/WX\bar{Y}\bar{Z}$
$\bar{N}D\bar{Q}/\bar{W}\bar{X}\bar{Y}\bar{Z}$
$\bar{N}\bar{D}Q/W\bar{X}Y\bar{Z}$
$\bar{N}\bar{D}Q/WXY\bar{Z}$
$N\bar{D}\bar{Q}/\bar{W}\bar{X}\bar{Y}\bar{Z}$
$\bar{N}D\bar{Q}/\bar{W}\bar{X}\bar{Y}\bar{Z}$
$\bar{N}\bar{D}Q/W\bar{X}\bar{Y}\bar{Z}$
$\bar{N}D\bar{Q}/\bar{W}\bar{X}\bar{Y}\bar{Z}$
$\bar{N}D\bar{Q}/\bar{W}\bar{X}\bar{Y}\bar{Z}$
$N\bar{D}\bar{Q}/\bar{W}\bar{X}\bar{Y}\bar{Z}$
$N\bar{D}\bar{Q}/\bar{W}\bar{X}\bar{Y}\bar{Z}$
$N\bar{D}\bar{Q}/WX\bar{Y}\bar{Z}$
$\bar{N}D\bar{Q}/WX\bar{Y}\bar{Z}$
$\bar{N}\bar{D}Q/W\bar{X}XZ$

2. State encoding table:

Input

Coin	N	D	Q
Nickel	1	0	0
Dime	0	1	0
Quarter	0	0	1

Output

Dispense	w
Return Nickel	x
Return Dime	y
Return 2 Dimes	z

State Encoding

State	ps2	ps1	ps0
S_0	0	0	0
S_5	0	0	1
S_{10}	0	1	0
S_{15}	0	1	1
S_{20}	1	0	0

3. State transition table with symbolic state representation and with encoded version, you should include both.

	Present state			Inputs			Next state			Output			
	ps2	ps1	ps0	N	D	Q	NS2	NS1	NS0	W	X	Y	Z
S_0	0	0	0	1	0	0	0	0	1	0	0	0	0
S_0	0	0	0	0	1	0	0	1	0	0	0	0	0
S_0	0	0	0	0	0	1	0	0	0	1	0	0	0
S_5	0	0	1	1	0	0	0	1	0	0	0	0	0
S_5	0	0	1	0	1	0	0	1	1	0	0	0	0
S_5	0	0	1	0	0	1	0	0	0	1	1	0	0
S_{10}	0	1	0	1	0	0	0	1	1	0	0	0	0
S_{10}	0	1	0	0	1	0	1	0	0	0	0	0	0
S_{10}	0	1	0	0	0	1	0	0	0	1	0	1	0
S_{15}	0	1	1	1	0	0	1	0	0	0	0	0	0
S_{15}	0	1	1	0	1	0	0	0	0	1	0	0	0
S_{15}	0	1	1	0	0	1	0	0	0	1	1	1	0
S_{20}	1	0	0	1	0	0	0	0	0	1	0	0	0
S_{20}	1	0	0	0	1	0	0	0	0	1	1	0	0
S_{20}	1	0	0	0	0	1	0	0	0	1	0	0	1

4. Logic formulas for next state and output:

Equations for Next state

$$NS2 = \boxed{\overline{pS2} \cdot pS1 \cdot \overline{pS0} \cdot D + \overline{pS2} \cdot pS1 \cdot pS0 \cdot N}$$

$$NS1 = \overline{pS2} \cdot \overline{pS1} \cdot \overline{pS0} \cdot D + \overline{pS2} \cdot \overline{pS1} \cdot pS0 \cdot N + \overline{pS2} \cdot \overline{pS1} \cdot pS0 \cdot D$$

$$+ \overline{pS2} \cdot pS1 \cdot \overline{pS0} \cdot N$$

$$= \boxed{\overline{pS2} \cdot \overline{pS1} \cdot D + \overline{pS2} \cdot N (pS1 \oplus pS0)}$$

$$NS0 = \overline{pS2} \cdot \overline{pS1} \cdot \overline{pS0} \cdot N + \overline{pS2} \cdot \overline{pS1} \cdot pS0 \cdot D$$

$$+ \overline{pS2} \cdot pS1 \cdot \overline{pS0} \cdot N$$

$$= \boxed{\overline{pS2} \cdot \overline{pS0} \cdot N + \overline{pS2} \cdot pS1 \cdot pS0 \cdot D}$$

Equations for outputs :-

$$W = \overline{ps2}\cdot\overline{ps1}\cdot\overline{ps0}\cdot Q + \overline{ps2}\cdot\overline{ps1}\cdot ps0\cdot Q + \overline{ps2}\cdot ps1\cdot\overline{ps0}\cdot Q$$

$$+ \overline{ps2}\cdot ps1\cdot ps0\cdot D + \overline{ps2}\cdot ps1\cdot ps0\cdot Q + ps2\cdot\overline{ps1}\cdot\overline{ps0}\cdot N$$

$$+ ps2\cdot\overline{ps1}\cdot\overline{ps0}\cdot D + ps2\cdot\overline{ps1}\cdot\overline{ps0}\cdot Q$$

$$= \boxed{\begin{aligned}&\overline{ps2}\cdot\overline{ps1}\cdot\overline{ps0}\cdot Q + \overline{ps2}\cdot\overline{ps1}\cdot ps0\cdot Q + \overline{ps2}\cdot ps1\cdot\overline{ps0}\cdot Q\\ &+ \overline{ps2}\cdot ps1\cdot ps0\,(D+Q) + ps2\cdot\overline{ps1}\cdot\overline{ps0}\,(N+D+Q)\end{aligned}}$$

$$x = \overline{ps2}\cdot\overline{ps1}\cdot ps0\cdot Q + \overline{ps2}\cdot ps1\cdot ps0\cdot Q + ps2\cdot\overline{ps1}\cdot\overline{ps0}\cdot D$$

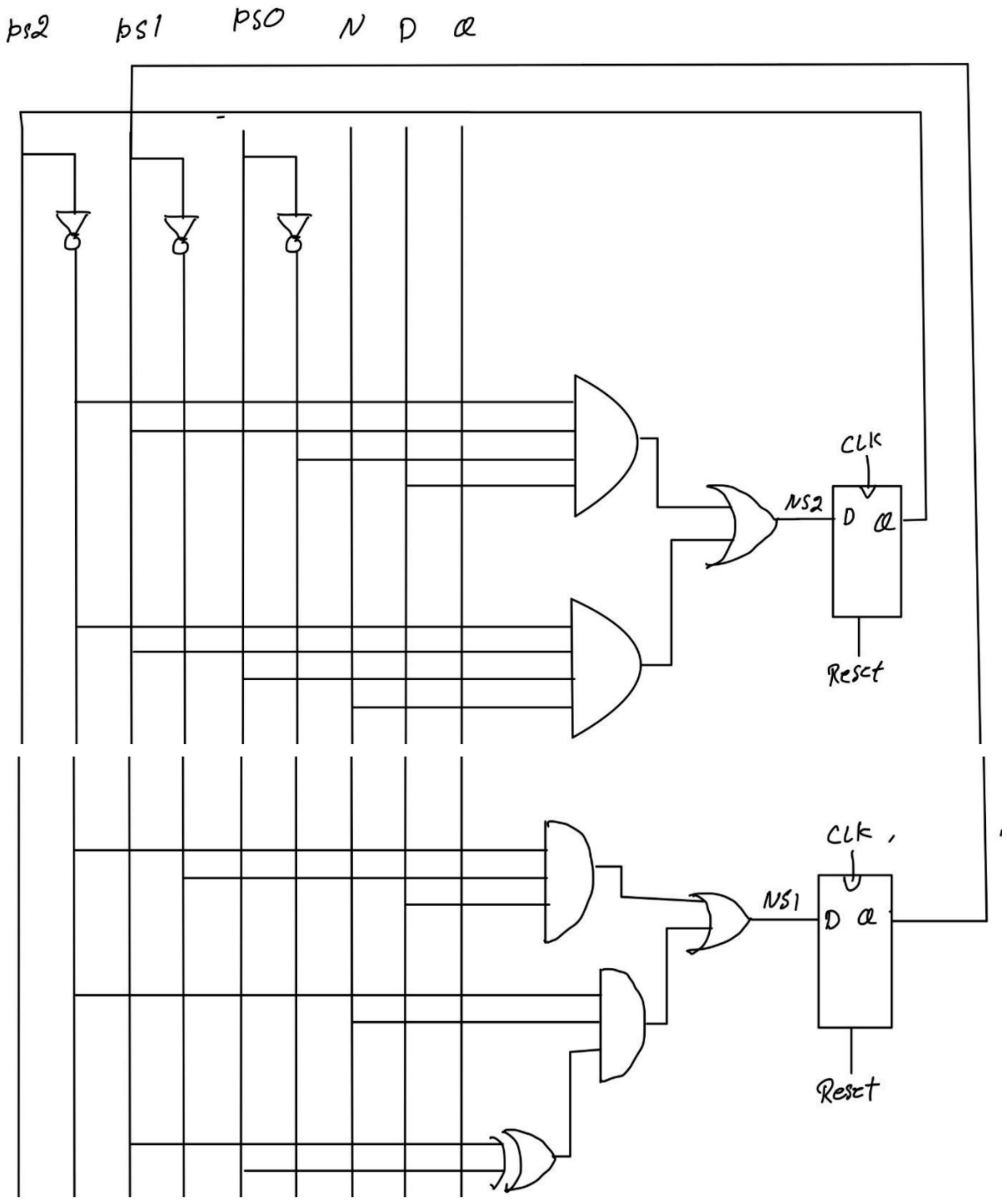
$$= \boxed{\overline{ps2}\cdot ps0\cdot Q + ps2\cdot\overline{ps1}\cdot\overline{ps0}\cdot D}$$

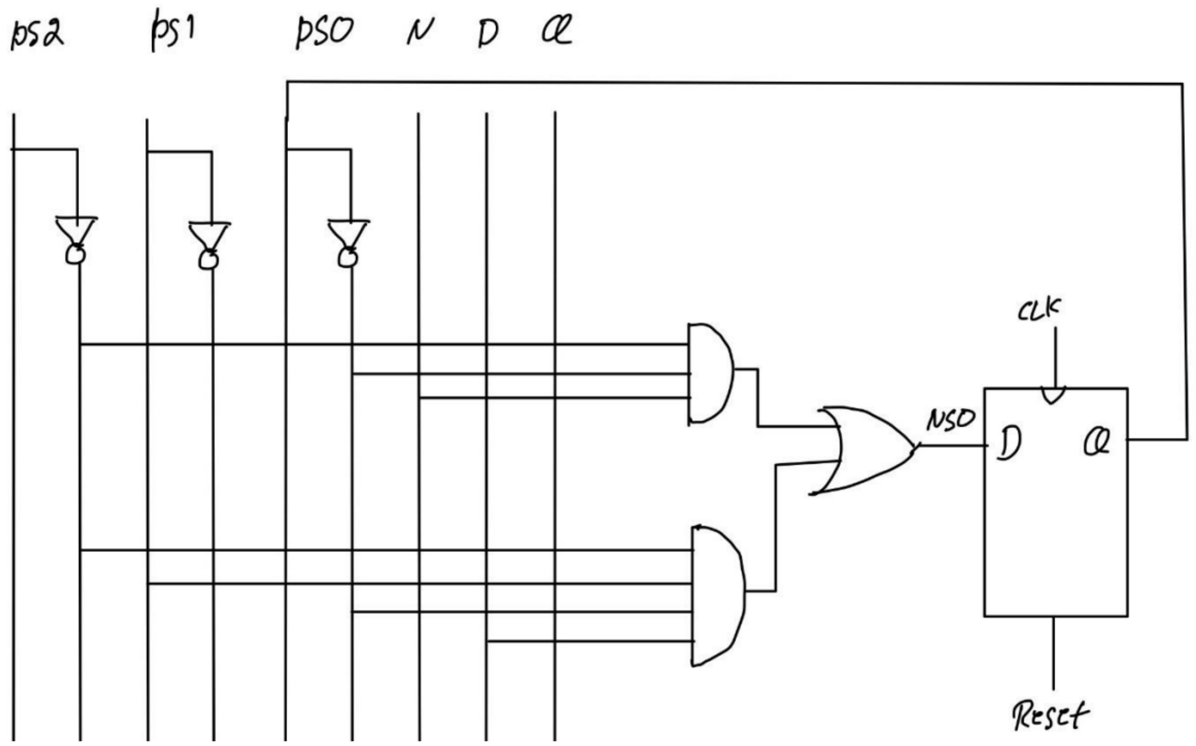
$$Y = \overline{ps2}\cdot ps1\cdot\overline{ps0}\cdot Q + \overline{ps2}\cdot ps1\cdot ps0\cdot Q$$

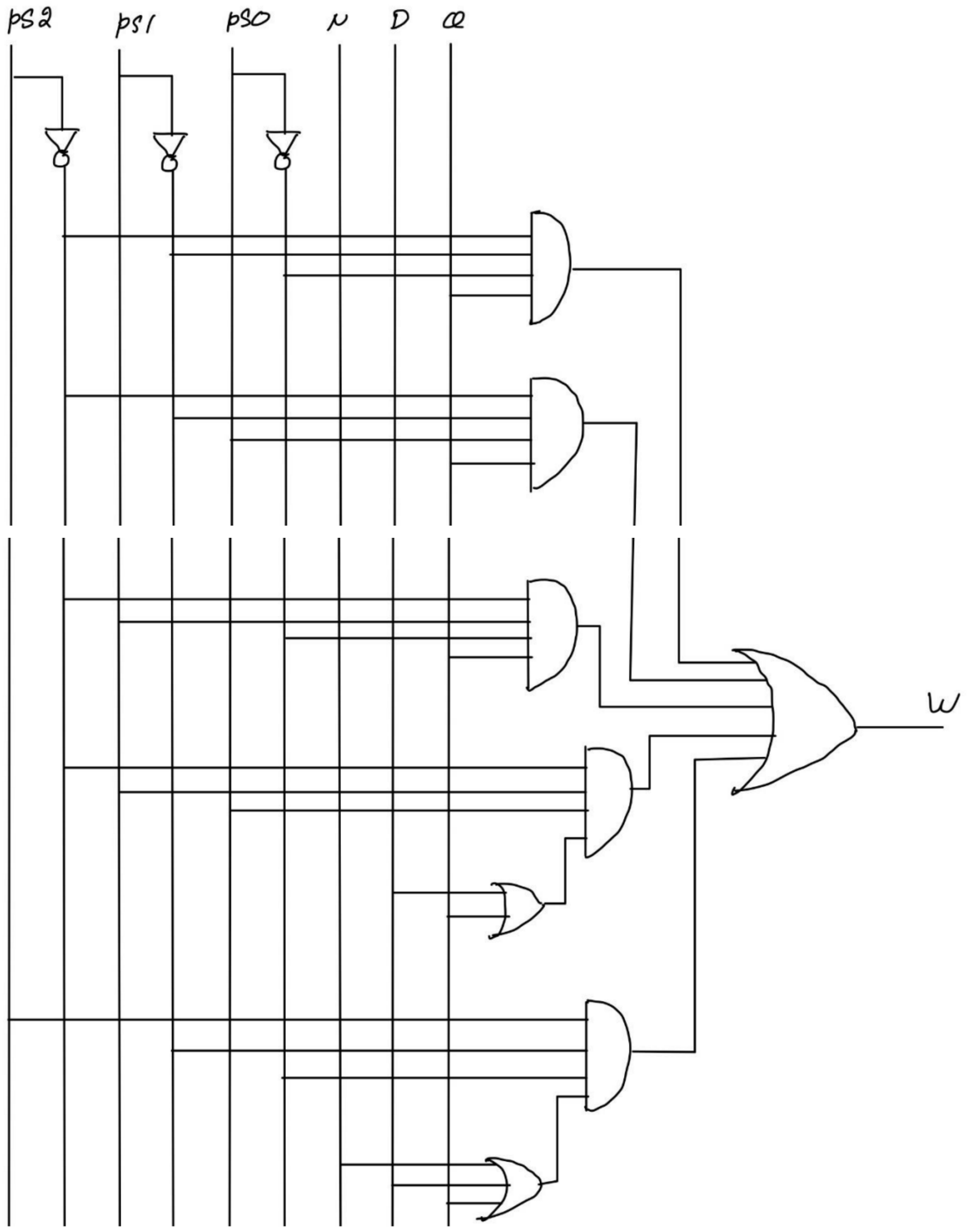
$$= \boxed{\overline{ps2}\cdot ps1\cdot Q}$$

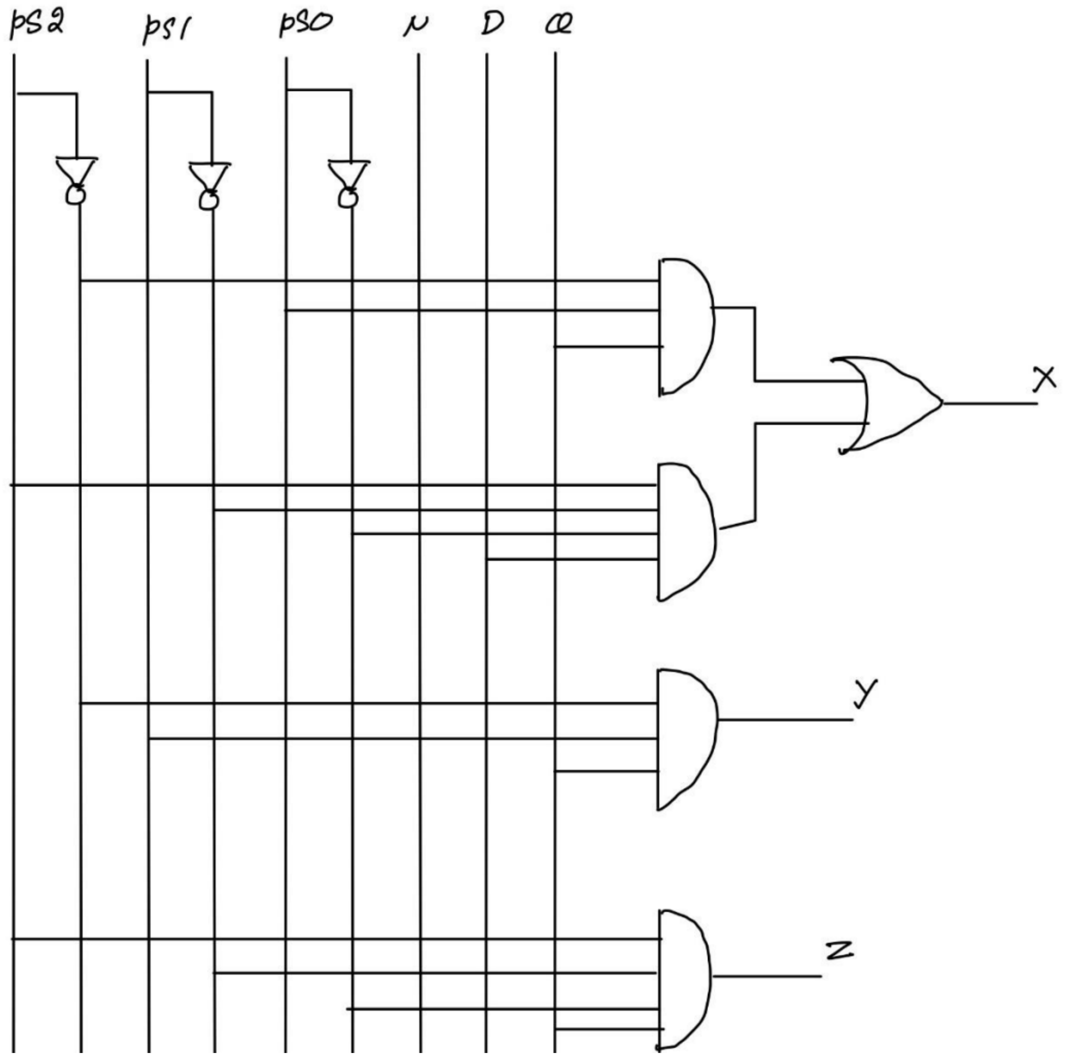
$$z = \boxed{ps2\cdot\overline{ps1}\cdot\overline{ps0}\cdot Q}$$

5. Schematic (combinational logic + resettable FF and outputs) of the FSM:


ps2
ps1
PS0
N
D
Q
CLK
NS2
D
Q
Reset
CLK
NS1
D
Q
Reset


ps2
ps1
ps0
N
D
Q
CLK
NS0
D
Q
Reset


PS2
PS1
PS0
N
D
Q
W



6. Verilog code for the FSM and test bench using test vectors. The test vectors should test all the possible inputs to the vending machine. The test vector should include at least two cases where excess change has been inserted (1Q + 1 D, 6 N, 2 D + 2 N, ...)

a. Verilog Code:

```
module cmpe125_lab6_tanveerkahlon (PS, N, D, Q, clk, reset, NS, W, X, Y, Z);
input N, D, Q, reset, clk;
output reg [2:0] PS, NS;
output reg W,X,Y,Z;
always @ (*)
begin
NS[2] = (~PS[2] & PS[1] & ~PS[0] & D) | (~PS[2] & PS[1] & PS[0] & N);
NS[1] = (~PS[2] & ~PS[1] & D) | (~PS[2] & N & (PS[1] ^ PS[0]));
NS[0] = (~PS[2] & ~PS[0] & N) | (~PS[2] & PS[1] & PS[0] & D);
end
```

```
always @ (posedge clk, posedge reset)
begin
if (reset)
PS <= 3'b000;
else
PS <= NS;
end

//outputs
always @(*)
begin
W = ((~PS[2])&(~PS[1])&(~PS[0])&Q) | ((~PS[2]) &(~PS[1])&(PS[0])&Q) | ((~PS[2])
& (PS[1])& (~PS[0])&Q) | ((~PS[2]) &(PS[1])&(PS[0])&(Q|D)) | ((PS[2])
&(~PS[1])&(~PS[0])&(N|D|Q));
X = (~PS[2] & PS[0] & Q) | (PS[2] & ~PS[1] & ~PS[0] & Q);
Y = ~PS[2] & PS[1] & Q;
Z = PS[2] & ~PS[1] & ~PS[0] & Q;
end
Endmodule
```

b. Test Bench Code:

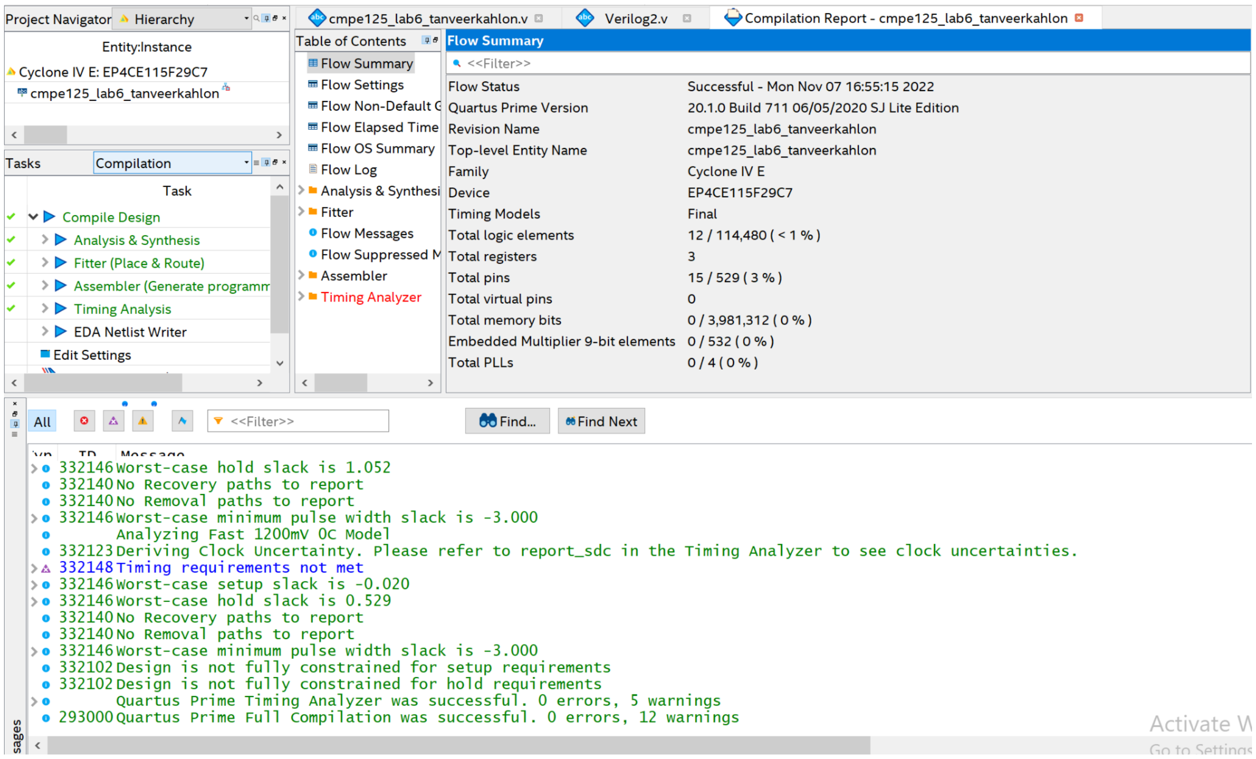
```
module vm_tb();
reg D, Q,N, clk, reset;
reg Dispense_exp, Ret_Nickel_exp, Ret_Dime_exp, Ret_2Dime_exp;
wire [2:0] PS, NS;
wire Dispense,Ret_Nickel,Ret_Dime,Ret_2Dime;
reg [31:0] vectornum, errors;
reg [6:0] testvectors[10000:0];

cmpe125_lab6_tanveerkahlon test (.PS(PS) , .N(N), .D(D), .Q(Q) , .clk(clk), .reset(reset) ,
.NS(NS),
.W(Dispense), .X(Ret_Nickel), .Y(Ret_Dime), .Z(Ret_2Dime));
always
begin
clk = 1; #5; clk = 0; #5;
end

initial
begin
$readmemb ("test_vector.tv", testvectors);
vectornum = 0; errors = 0;
reset = 1; #10; reset = 0;
end
```

```
always @(posedge clk)

begin
{N,D, Q, Dispense_exp, Ret_Nickel_exp, Ret_Dime_exp, Ret_2Dime_exp} =
testvectors[vectornum];
End

always @ (negedge clk)
if (~reset)
begin //skip cycles during reset
if(Dispense!== Dispense_exp) begin //check result
$display("Error:inputs Nickel,Dime, and Quarter = %b with current state %b", {N, D,
Q}, PS);
$display("output Dispense = %b (%b expected)", Dispense, Dispense_exp);
errors = errors + 1;
end

if (Ret_Nickel !== Ret_Nickel_exp) begin // checx result
$display("Error: inputs Nickel, Dime, and Ouarter = %b with current state %b", {N, D,
Q}, PS) ;
$display(" output Return Nickel = %b (%b expected)", Ret_Nickel, Ret_Nickel_exp);
errors = errors + 1;
end
if(Ret_Dime !== Ret_Dime_exp) begin //checx result
$display("Error: inputs Nickel, Dime and Quarter = %b with current state %b", {N, D,
Q}, PS) ;
$display(" output Return Dime = %b (%b expected)", Ret_Dime, Ret_Dime_exp);

errors = errors + 1;
end
if
(Ret_2Dime !== Ret_2Dime_exp) begin // check result
$display("Error: inputs Nickel, Dime, and Quarter = %b with current state %b", {N, D,
Q}, PS) ;
$display(" output Return two Dimes = %b (%b expected) ", Ret_2Dime,
Ret_2Dime_exp);
errors = errors + 1;
end
vectornum = vectornum + 1;
if (testvectors [vectornum] === 7'bx) begin
$display("%d tests completed with %d error", vectornum, errors);
$finish;
end
end
```

endmodule

c. Test Vector:
```
000_0000
001_1000
100_0000
100_0000
100_0000
100_0000
100_1000
100_0000
100_0000
010_0000
010_1100
010_0000
001_1010
100_0000
010_0000
100_0000
001_1001
010_0000
100_0000
010_1000
010_0000
100_0000
001_1110
100_0000
001_1100
```

7. Screenshots of the compiled Verilog without error:

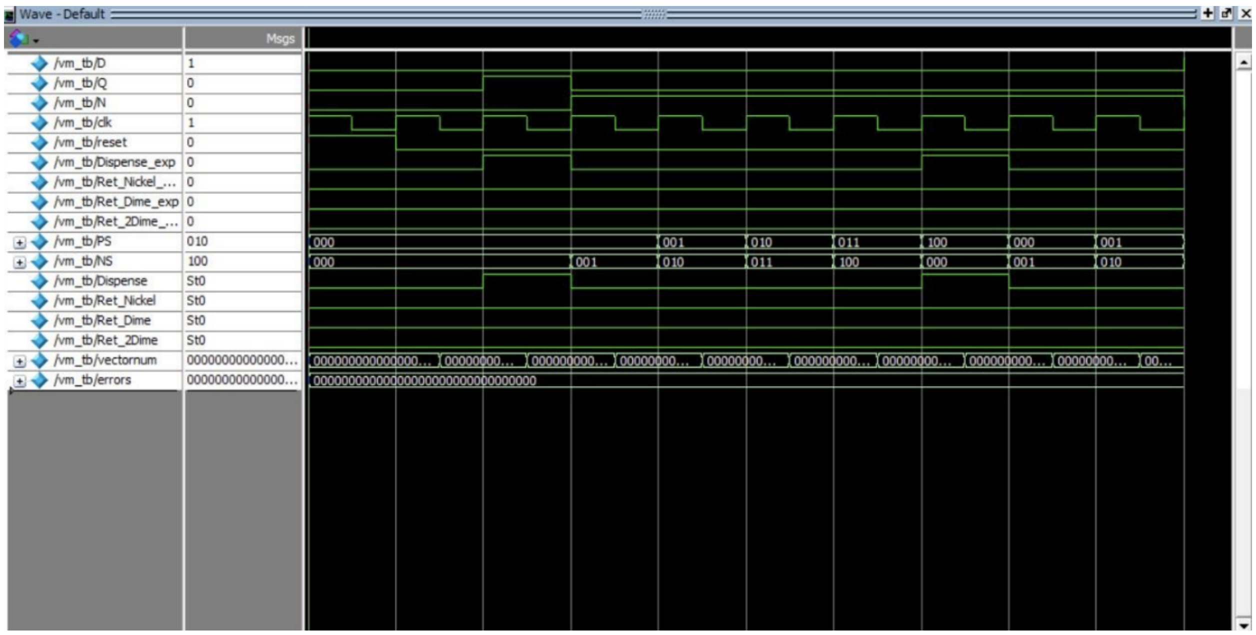


Flow Summary

Flow Status	Successful - Mon Nov 07 16:55:15 2022
Quartus Prime Version	20.1.0 Build 711 06/05/2020 SJ Lite Edition
Revision Name	cmpe125_lab6_tanveerkahlon
Top-level Entity Name	cmpe125_lab6_tanveerkahlon
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	12 / 114,480 (< 1 %)
Total registers	3
Total pins	15 / 529 (3 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)

```
332146 Worst-case hold slack is 1.052
332140 No Recovery paths to report
332140 No Removal paths to report
332146 Worst-case minimum pulse width slack is -3.000
       Analyzing Fast 1200mV 0C Model
332123 Deriving Clock Uncertainty. Please refer to report_sdc in the Timing Analyzer to see clock uncertainties.
332148 Timing requirements not met
332146 Worst-case setup slack is -0.020
332146 Worst-case hold slack is 0.529
332140 No Recovery paths to report
332140 No Removal paths to report
332146 Worst-case minimum pulse width slack is -3.000
332102 Design is not fully constrained for setup requirements
332102 Design is not fully constrained for hold requirements
       Quartus Prime Timing Analyzer was successful. 0 errors, 5 warnings
293000 Quartus Prime Full Compilation was successful. 0 errors, 12 warnings
```

8. Screenshots of the simulation results:



9. A summary describing verification of the implemented logic with the simulation results:

In this lab FSM is designed to function a vending machine which takes three inputs which are nickel, dime and quarter and output as Despenser, nickel, dime, and quarter. First, designed the state diagram which shows how the vending machine would work though the inputs. After that, checked all the inputs and made a encoding tables to make sure how the inputs will be using throughout the simulation. Then implemented the state transition table which shows the current and next state of the input, also it shows the output as well. Then minimized all the equations for next state and outputs which are used to make the handmade schematic. After designed the schematic, this implementation designed in Verilog coding, and included the test bench.