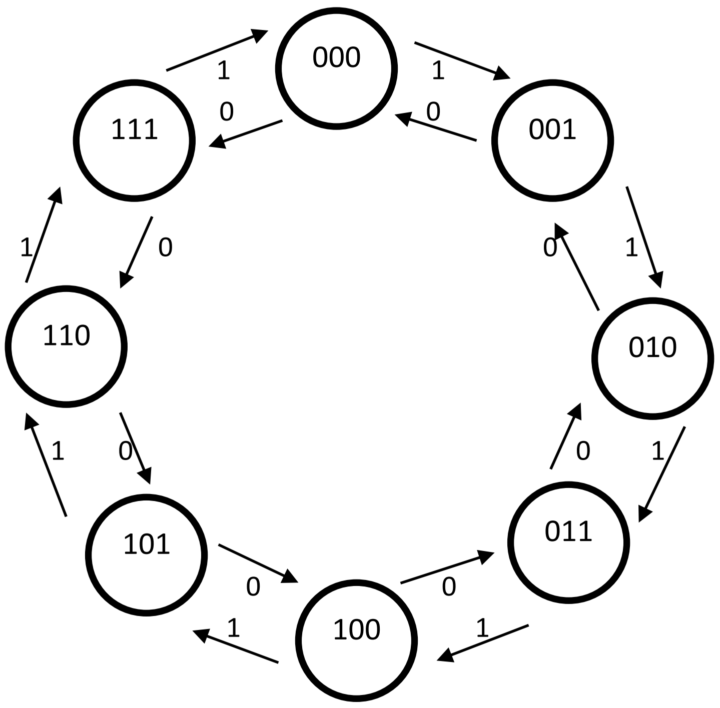


Truth Table

| $C_{1\ prev}$ | $C_{2\ prev}$ | $C_{3\ prev}$ | M | $C_1$ | $C_2$ | $C_3$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 |

K-Map for $C_1$

| $C_3M$ \ $C_1C_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 0 | 1 | 0 |
| 01 | 0 | 0 | 1 | 1 |
| 11 | 0 | 1 | 0 | 1 |
| 10 | 0 | 0 | 1 | 1 |

K-Map for $C_2$

| $C_3M$ \ $C_1C_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 0 | 0 | 1 |
| 01 | 0 | 1 | 1 | 0 |
| 11 | 1 | 0 | 0 | 1 |
| 10 | 0 | 1 | 1 | 0 |

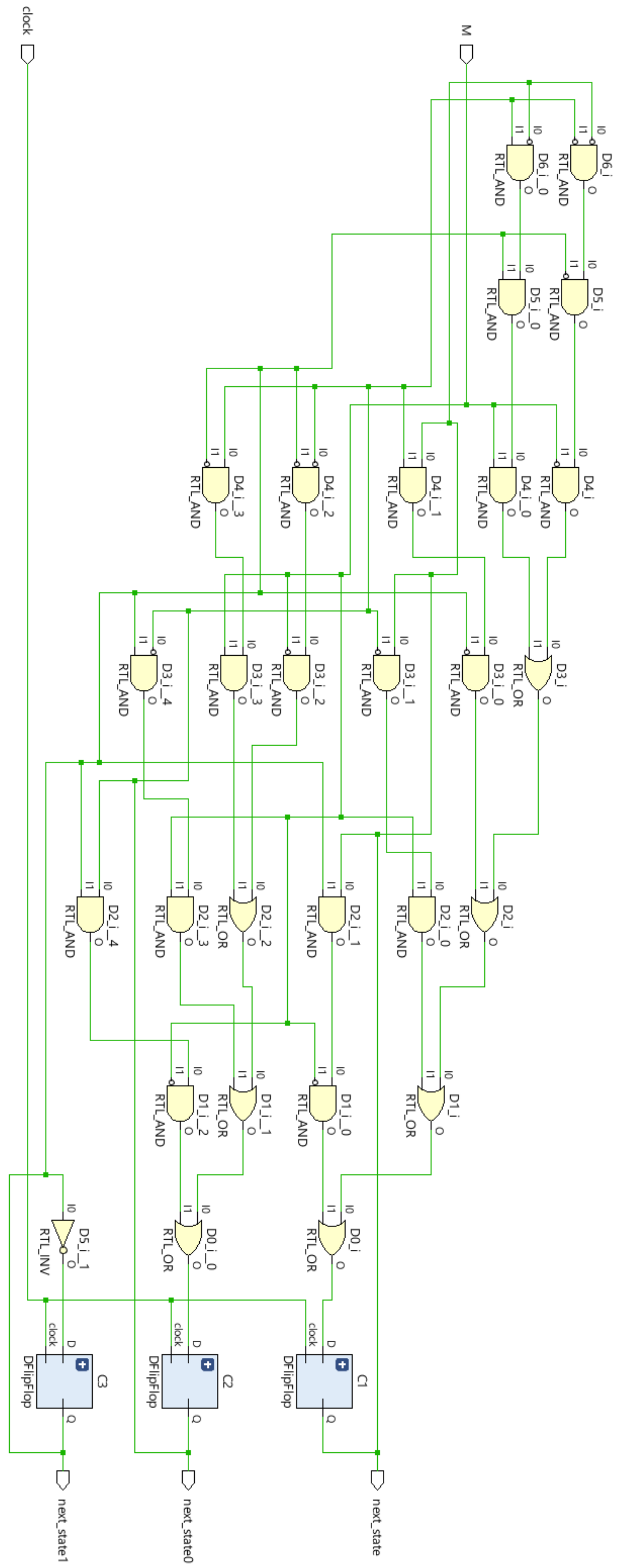
K-Map for $C_3$

| $C_3M$ \ $C_1C_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 1 | 1 | 1 |
| 01 | 1 | 1 | 1 | 1 |
| 11 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 |

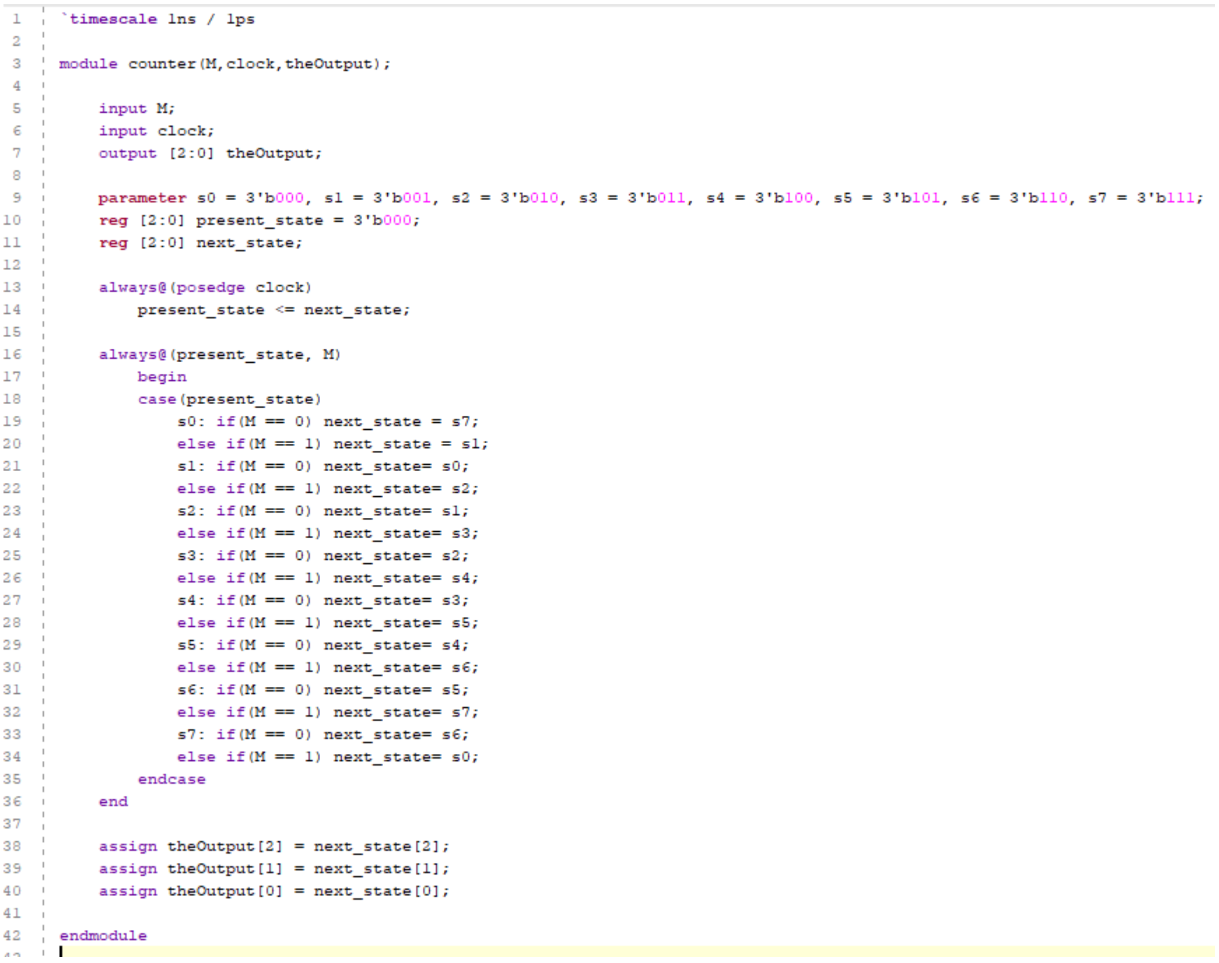
$D_{C1} = C_1'C_2'C_3'M' + C_1'C_2C_3M + C_1C_2C_3' + C_1C_2'M + C_1C_3M'$

$D_{C2} = C_2' C_3'M' + C_2 C_3'M + C_2' C_3M + C_2C_3M'$

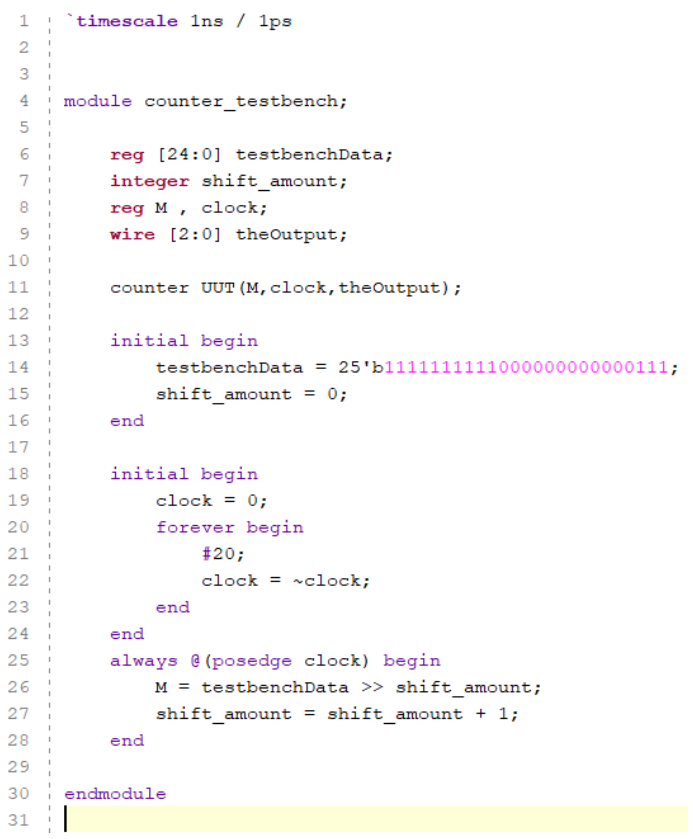
$D_{C3} = C_3'$



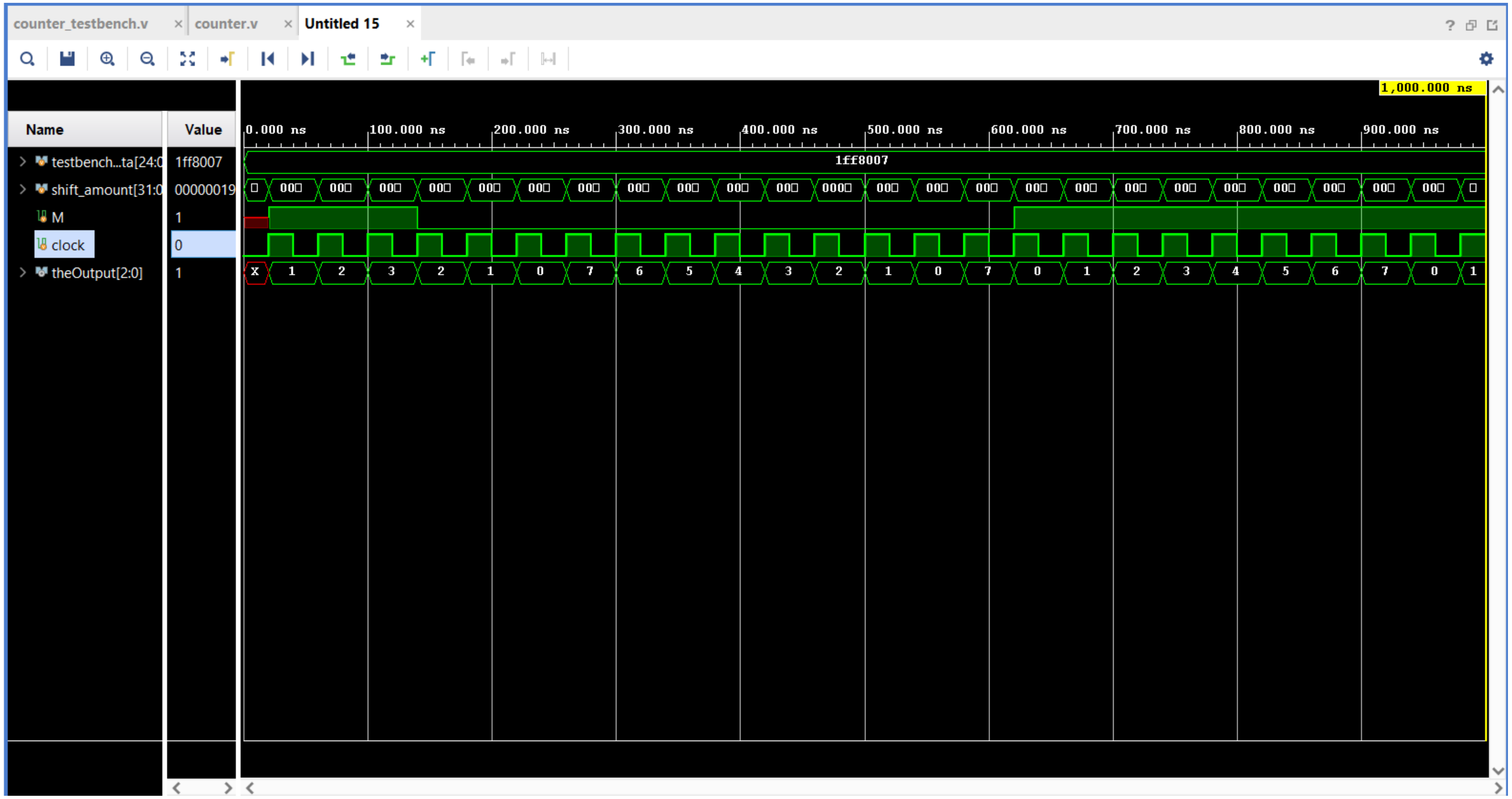
clock
M
D6_i
D6_i_0
D5_i
D5_i_0
D4_i
D4_i_0
D4_i_1
D4_i_2
D4_i_3
D3_i
D3_i_0
D3_i_1
D3_i_2
D3_i_3
D3_i_4
D2_i
D2_i_0
D2_i_1
D2_i_2
D2_i_3
D2_i_4
D1_i
D1_i_0
D1_i_1
D1_i_2
D0_i
D0_i_0
D5_i_1
RTL_AND
RTL_OR
RTL_INV
C1
C2
C3
DFlipFlop
next_state
next_state0
next_state1



```verilog
`timescale 1ns / 1ps

module counter(M,clock,theOutput);

    input M;
    input clock;
    output [2:0] theOutput;

    parameter s0 = 3'b000, s1 = 3'b001, s2 = 3'b010, s3 = 3'b011, s4 = 3'b100, s5 = 3'b101, s6 = 3'b110, s7 = 3'b111;
    reg [2:0] present_state = 3'b000;
    reg [2:0] next_state;

    always@(posedge clock)
        present_state <= next_state;

    always@(present_state, M)
        begin
        case(present_state)
            s0: if(M == 0) next_state = s7;
            else if(M == 1) next_state = s1;
            s1: if(M == 0) next_state= s0;
            else if(M == 1) next_state= s2;
            s2: if(M == 0) next_state= s1;
            else if(M == 1) next_state= s3;
            s3: if(M == 0) next_state= s2;
            else if(M == 1) next_state= s4;
            s4: if(M == 0) next_state= s3;
            else if(M == 1) next_state= s5;
            s5: if(M == 0) next_state= s4;
            else if(M == 1) next_state= s6;
            s6: if(M == 0) next_state= s5;
            else if(M == 1) next_state= s7;
            s7: if(M == 0) next_state= s6;
            else if(M == 1) next_state= s0;
        endcase
    end

    assign theOutput[2] = next_state[2];
    assign theOutput[1] = next_state[1];
    assign theOutput[0] = next_state[0];

endmodule
```



```verilog
`timescale 1ns / 1ps


module counter_testbench;

    reg [24:0] testbenchData;
    integer shift_amount;
    reg M , clock;
    wire [2:0] theOutput;

    counter UUT(M,clock,theOutput);

    initial begin
        testbenchData = 25'b1111111111000000000000111;
        shift_amount = 0;
    end

    initial begin
        clock = 0;
        forever begin
            #20;
            clock = ~clock;
        end
    end
    always @(posedge clock) begin
        M = testbenchData >> shift_amount;
        shift_amount = shift_amount + 1;
    end

endmodule
```


counter_testbench.v
counter.v
Untitled 15
1,000.000 ns
Name
Value
testbench...ta[24:0
1ff8007
shift_amount[31:0
00000019
M
1
clock
0
theOutput[2:0]
1
0.000 ns
100.000 ns
200.000 ns
300.000 ns
400.000 ns
500.000 ns
600.000 ns
700.000 ns
800.000 ns
900.000 ns
1ff8007
X 1 2 3 2 1 0 7 6 5 4 3 2 1 0 7 0 1 2 3 4 5 6 7 0 1

İbrahim Burak Tanrıkulu 21827852

Şerife Ebru Yardım 21785784

BBM 233 Lab Experiment 5