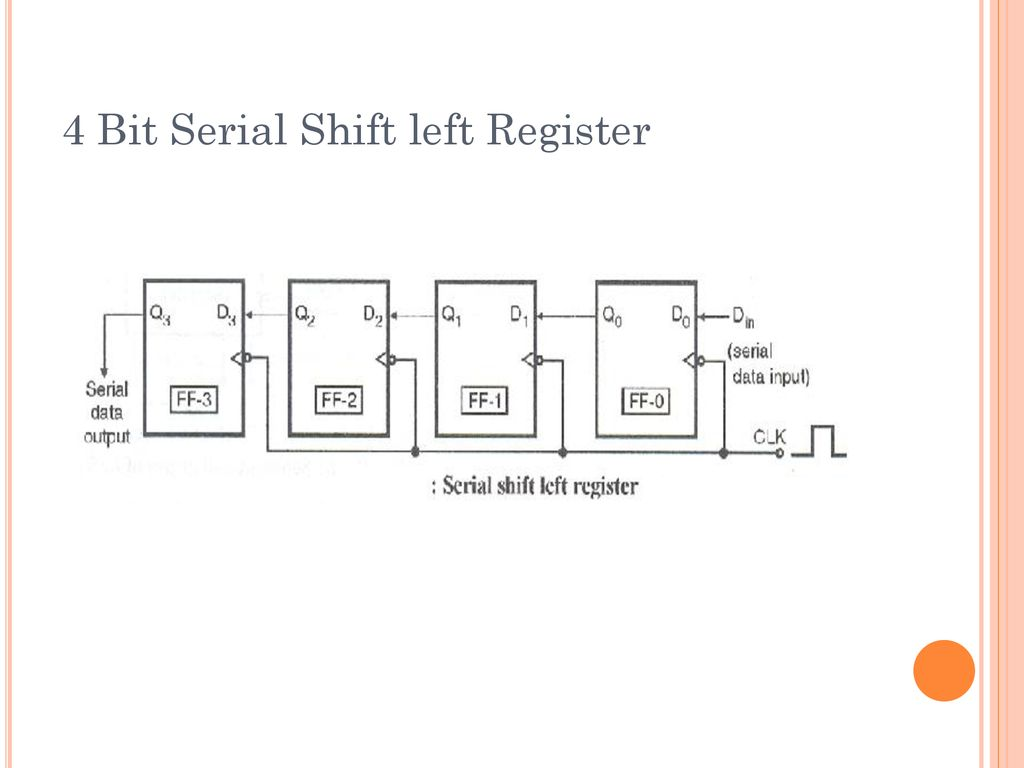
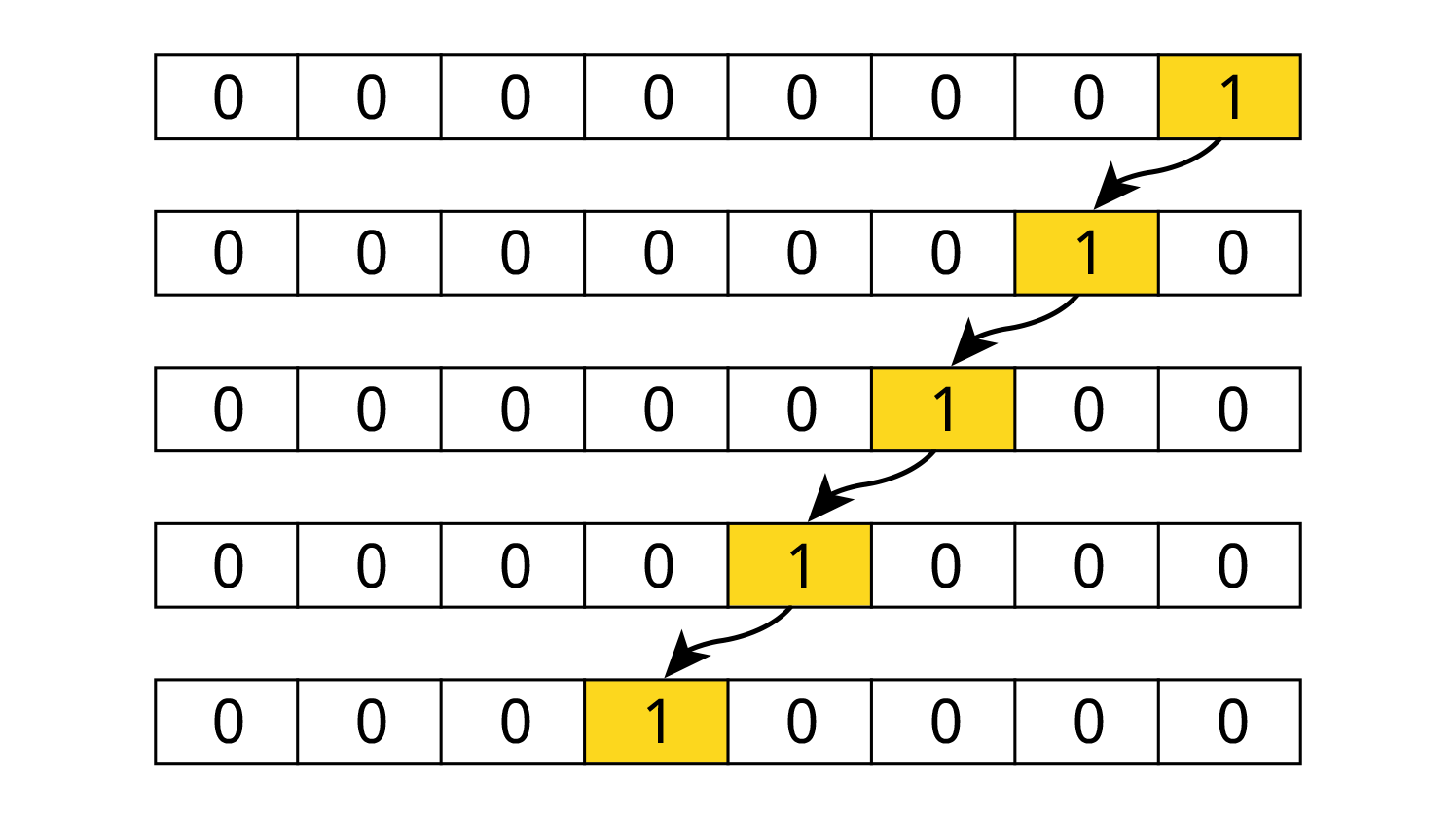
**PROBLEM:**5. Write Verilog code for 4-bit left shift register.

**CIRCUIT DIAGRAM:**

****

**TRUTH TABLE:**

**D3 D2 D1 D0**

****

**VERILOG CODE:**

**D FLIP FLOP:**

module D\_FLIPFLOP(input d,

input rst,

input clk,

output reg q);

always @ (posedge clk )

case(rst)

1:q <= 0;

0:q <= d;

endcase

endmodule

**COUNTER:**

module LSREGISTER\_4BIT( input d0,

output q0,q1,q2,q3,

input CLK,

input RST);

wire Q0,Q1,Q2,Q3;

D\_FLIPFLOP D00(.d(Q2),.q(q3),.rst(RST),.clk(CLK));

D\_FLIPFLOP D01(.d(Q1),.q(q2),.rst(RST),.clk(CLK));

D\_FLIPFLOP D10(.d(Q0),.q(q1),.rst(RST),.clk(CLK));

D\_FLIPFLOP D11(.d(d0),.q(q0),.rst(RST),.clk(CLK));

assign Q0=q0;

assign Q1=q1;

assign Q2=q2;

assign Q3=q3;

endmodule

**TEST BENCH:**

module LS\_REGISTER4BITTB;

reg D,r,clock;

wire QA,QB,QC,QD;

LSREGISTER\_4BIT uut(.d0(D),.q0(QD),.q1(QC),.q2(QB),.q3(QA),.RST(r),.CLK(clock));

initial

begin

clock <=0;

end

always #50 clock=~clock;

initial

begin

r=1;D=1;

#200

r=1;D=0;

#100

r=0;D=1;

#70

r=0;D=0;

#200

r=0;D=0;

#150

r=0;D=0;

#100

r=0;D=1;

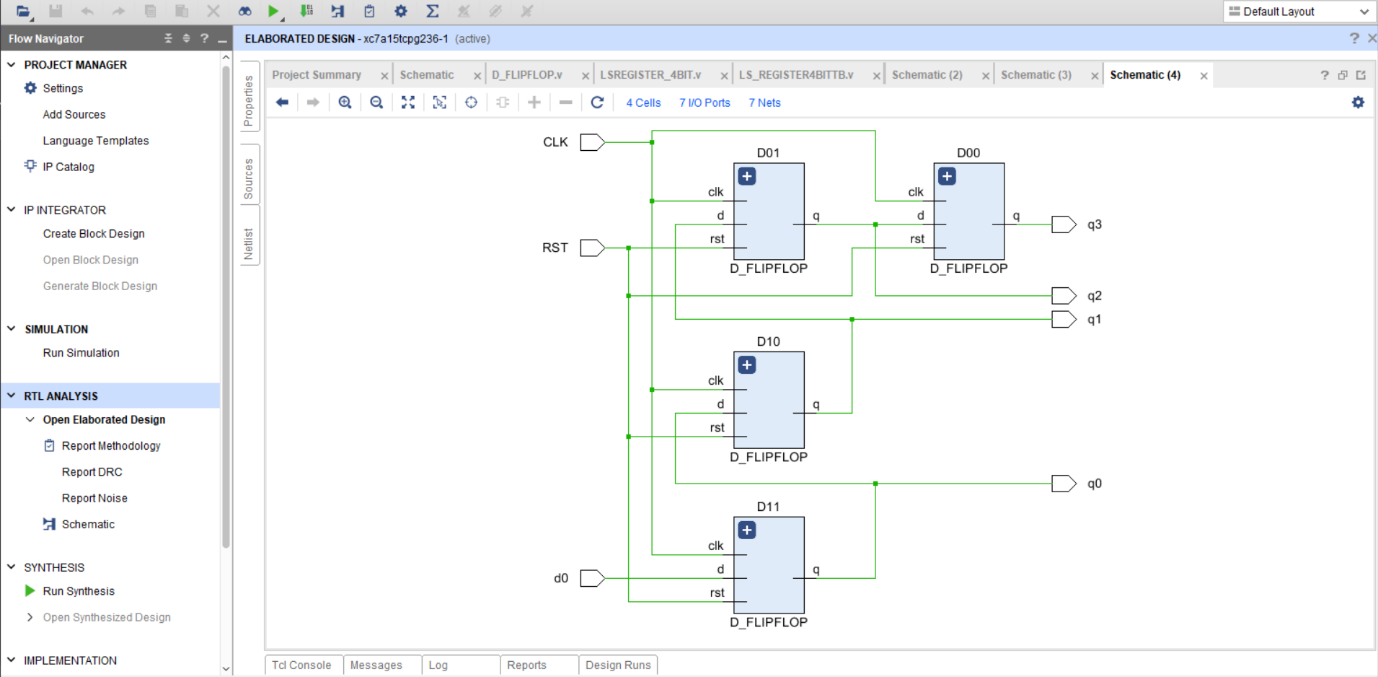
#200

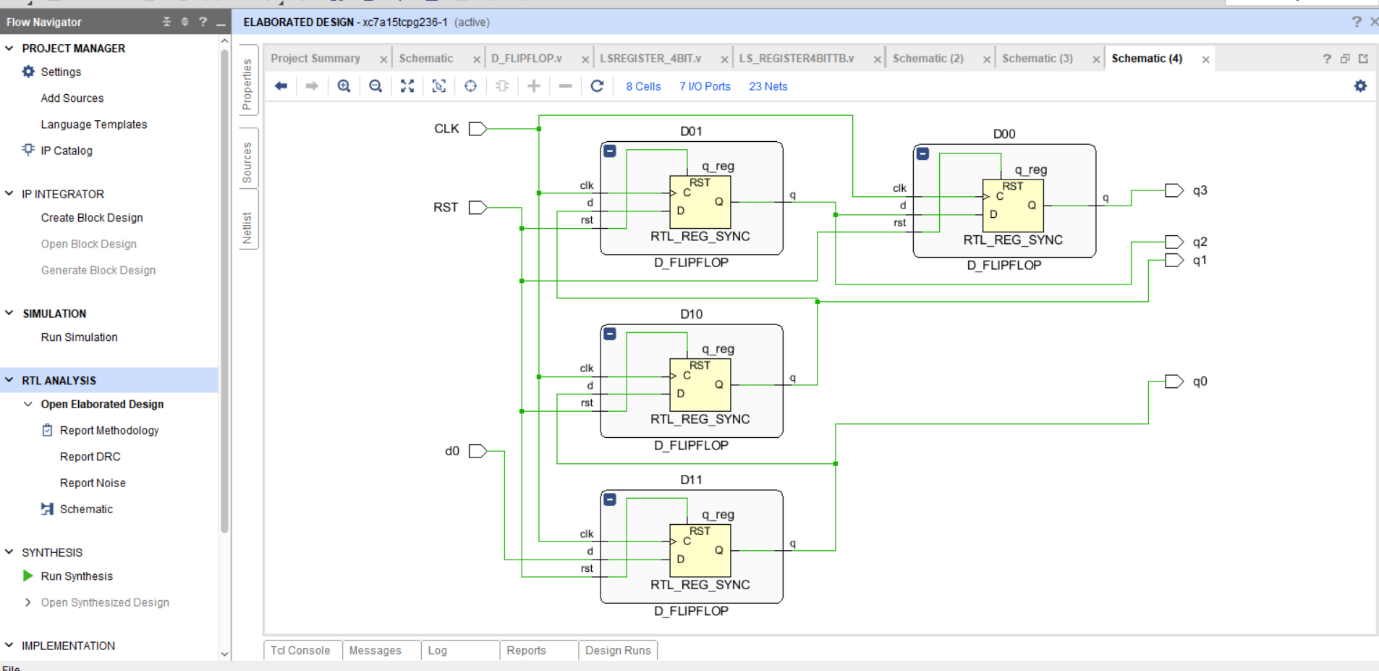
r=0;D=0;

end

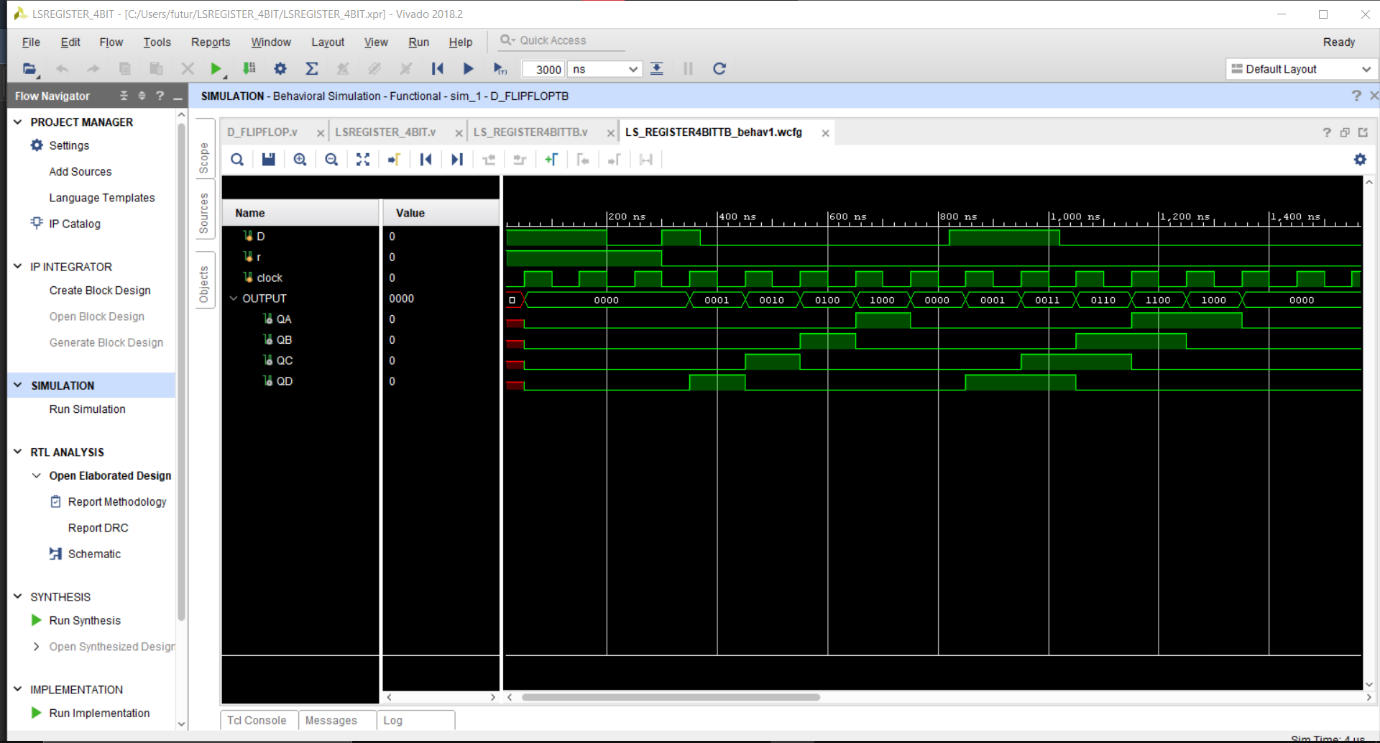
endmodule

**SCHEMATIC DIAGRAM:**

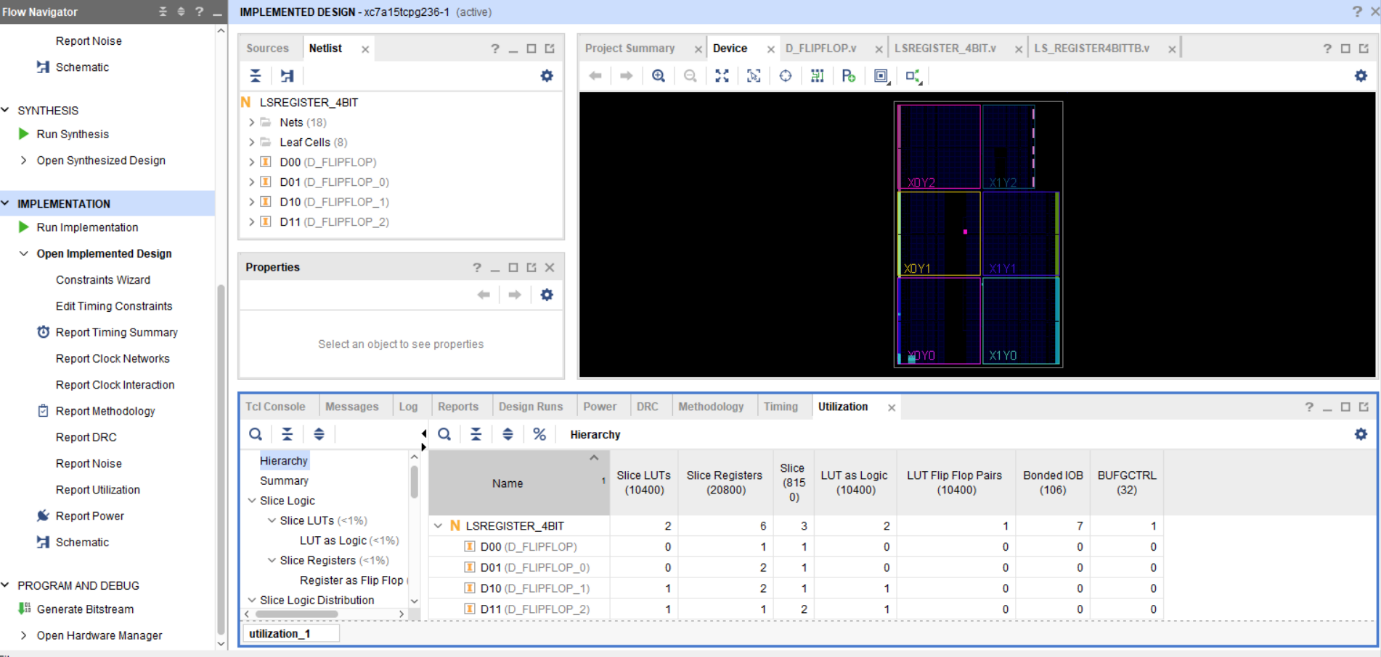




**OUTPUT:**



**REPORT UTILIZATION:**



**CONCLUSION:** WE HAVE LEARN LEFT SHIFT REGISTER DESIGN USING VERILOG CODE