**Simulation of 8:1 Mux using Verilog code**

Multiplexer

module aa(sel, a, y);

input [2:0] sel;

input [7:0] a;

output y;

reg y;

always @(sel,a)

begin

case(sel)

3'b000:y=a[0];

3'b001:y=a[1];

3'b010:y=a[2];

3'b011:y=a[3];

3'b100:y=a[4];

3'b101:y=a[5];

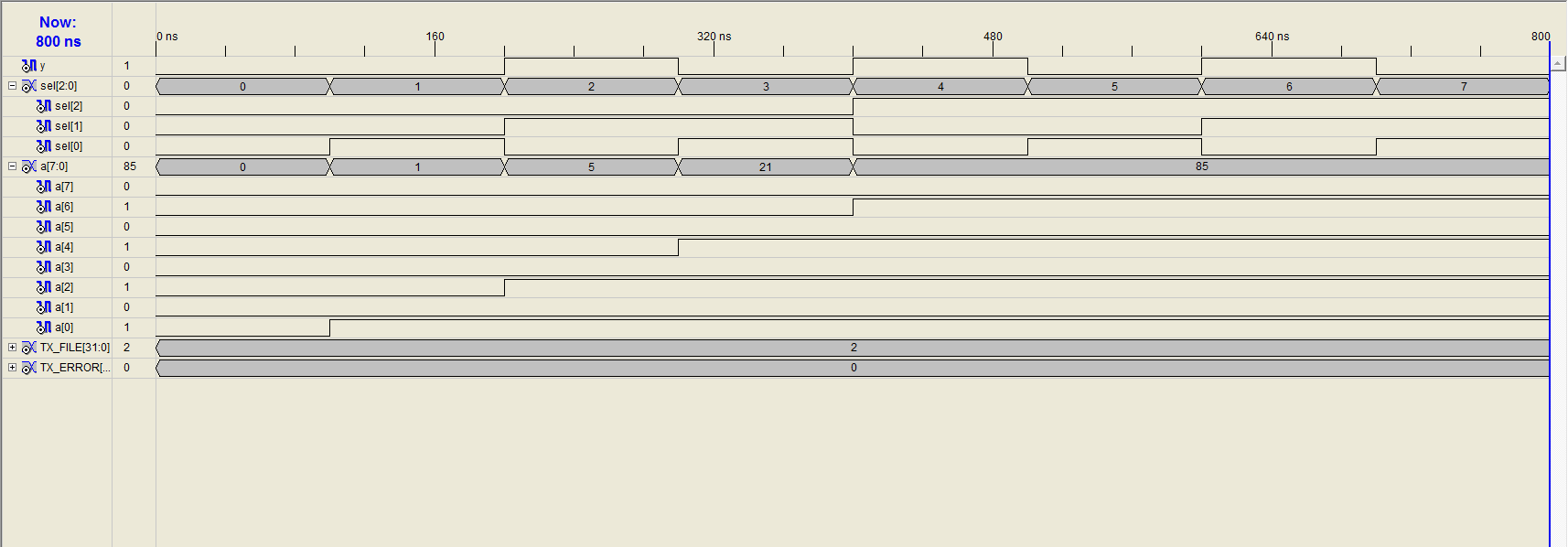
3'b110:y=a[6];

3'b111:y=a[7];

endcase

end

endmodule



**Simulation of D flipflop with +ve edge triggering using Verilog**

D flipflop

module df1(clock, d, q, qb);

input clock, d;

output q, qb;

reg q, qb;

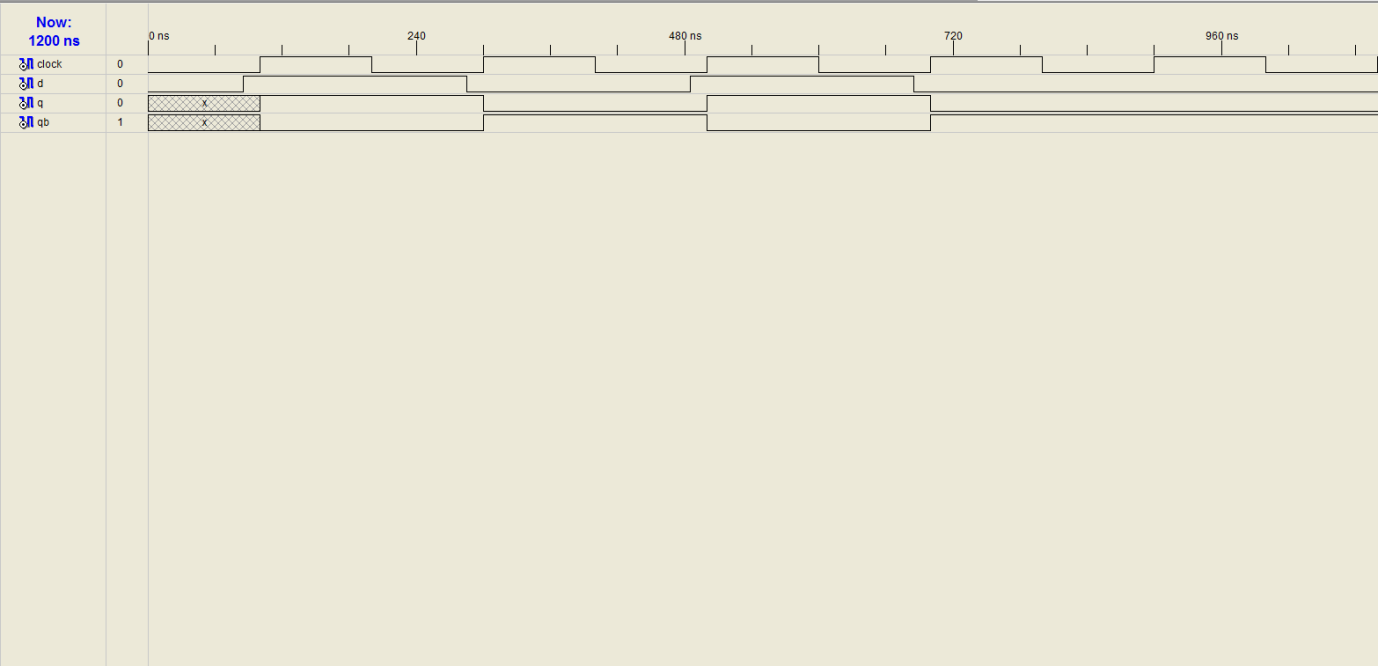
always @(posedge clock)

begin

q=d;

qb=~q;

end

endmodule

**Simulation of mod-8 up counter using Verilog**

Up Counter

module new(clock, reset, q);

input clock;

input reset;

output [7:0] q;

reg [3:0]q;

always @ (posedge clock)

begin

if (reset)

q= 4'b0000;

else

if (q==4'b0111)

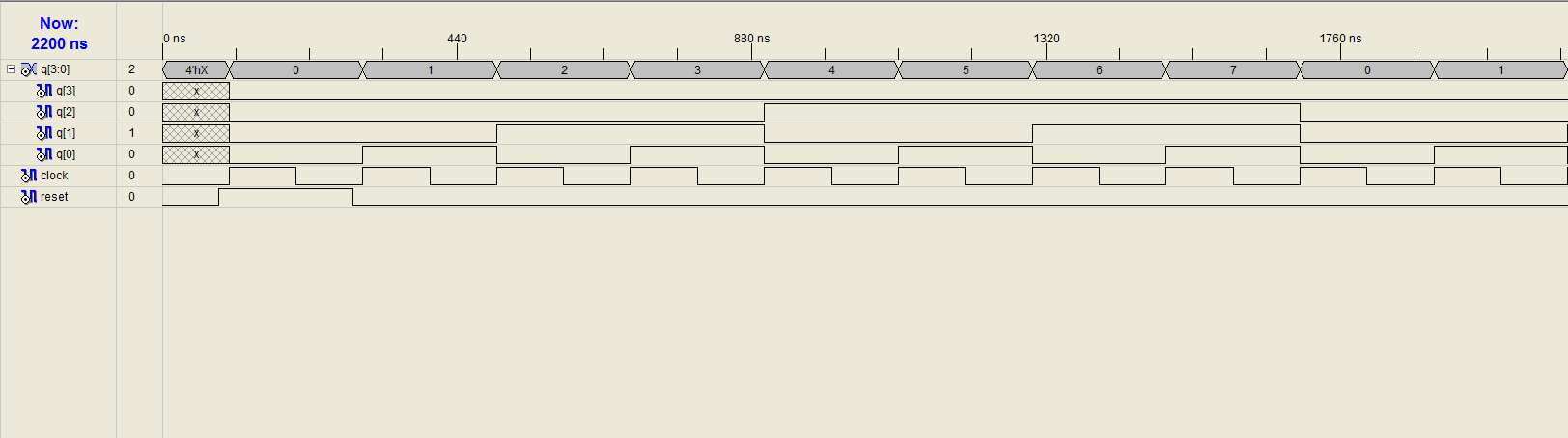
q=4'b0000;

else

q=q+1;

end

endmodule

****

**Simulation of switched tail counter or Johnson Counter**

**using Verilog**

module j1(clk, reset, q);

input clk, reset;

output [3:0] q;

reg [3:0] q;

always @ (posedge clk or posedge reset)

if (reset ==1)

q=4'b0000;

else

q={q[3:0], ~q[3]};

endmodule

