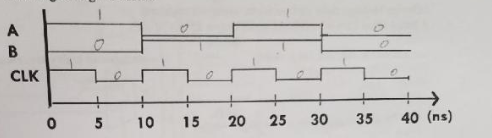
Write Verilog Code For Each Of The Following Diagram:



module circuit\_tb;

reg a,b,clk;

wire [1:0] y;

circuit uut(.a(a), .b(b), .clk(clk));

initial clk = 1;

always #5

clk = ~clk;

initial

begin

a=1; b=0;

#10;

a=0; b=1;

#10;

a=1; b=1;

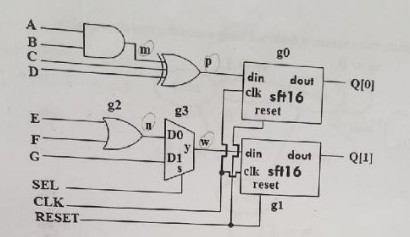
#10;

a=0; b=0;

#20;

$stop;

end

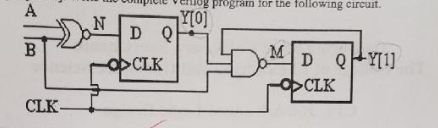


module circuit(a,b,c,d,e,f,g,sel,clk,reset,q);  
   input a,b,c,d,e,f,g,sel,clk,reset,q;  
   output [1:0] q;  
   wire m,n,p,w;  
   assign p = m^c^b;  
   or g2(e,f);  
   mux2to1 g3(.d0(a), d1(b), .s(sel), ,y(w));  
   sft16 g0(.din(p), .clk(clk), .reset(reset), .dout(q[0]));  
   sft16 g1(.din(w), .clk(clk), .reset(rest), .dout(q[1]));  
   endmodule

module circuit(a,b,c,d,e,f,g,sel,clk,reset,q);  
   input a,b,c,d,e,f,g,sel,clk,reset,q;  
   output reg [1:0] q;  
   wire m,n,p,w;

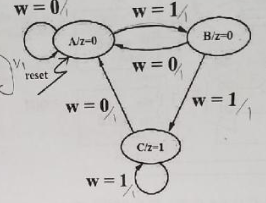
assign m = a&b;  
   assign p = m^(c^d);  
assign n = (e|f);  
   mux2to1 g3(.d0(n), d1(g), .s(sel), .y(w));  
   sft16 g0(.din(p), .clk(clk), .reset(reset), .dout(q[0]));  
   sft16 g1(.din(w), .clk(clk), .reset(reset), .dout(q[1]));  
   endmodule

module circuit(a,b,c,d,e,f,g,sel,clk,reset,q);  
   input a,b,c,d,e,f,g,sel,clk,reset,q;  
   output [1:0] q;  
   wire m,n,p,w;  
   assign p = m^c^b;  
   or g2(n,e,f);  
   mux2to1 g3(n, G, SEL, w);  
   sft16 g0(clk, p, reset, q[0]);  
   sft16 g1(clk, w, RESET, q[1]);  
   endmodule



module circuit(a,b,clk,y);  
   input a, b, clk;  
   output [1:0] y;

reg [1:0] y;  
   wire m,n;  
   assign n = ~(a^b);  
   assign m = ~(y[1] & y[0] & b);  
   always@(clk)  
       begin  
           y[0] <= n  
           y[1] <= m;  
       end



module fsm(reset, clk, a, y);

input reset, clk, a;

output z;

reg z;

parameter

A = 2'b00; B = 2b'01; C = 2'b11;

reg [2:0] cs,ns;

always@(posedge clk or posedge reset)

begin

case(cs)

if reset

cs <= A;

else

cs <= ns;

end

always@(cs or a)

begin

A: if(a)

ns = B;

else

ns = A;

B: if(a)

ns = A;

else

ns = C;

C: if(a)

ns = A;

else

ns = C;

default:

ns = A;

endcase

end

always@(cs or a)

begin

case(ns)

A: y=0;

B: y=0;

C: y=1;

default: y =0;

endcase

end

endmodule

2 to 1 Multiplexer In Verilog hierarchy design method

Module mux2to1(d0,d1,s,y);

Input d0,d1,s;

Output y;

Wire m,n;

Assign m = ~s&d[0];

Assign n = s&d[1];

Assign y = m|n;

Design 16 stage right shift serial in, serial out registers

Serial input data is “din” and serial output data is “dout”

Module sft16(clk,din,reset,dout);

Input clk,din,reset;

Output dout;

always@(posedge clk or posedge reset)

begin

if(reset ==1)

dout <= 0;

else if(clk == 1)

dout <= din;

end

endmodule

Module sft16(clk,din,reset,dout);

Input clk,din,reset;

Output [15:0] dout;

Reg [15:0] dout;

always@(posedge clk or posedge reset)

begin

if(reset)

dout <= 0;

else

dout <= {din, q[15:1]};

end