Manoj M Mallya

200905130

Section : C2

Roll no : 23

LAB – 5

Exercises :

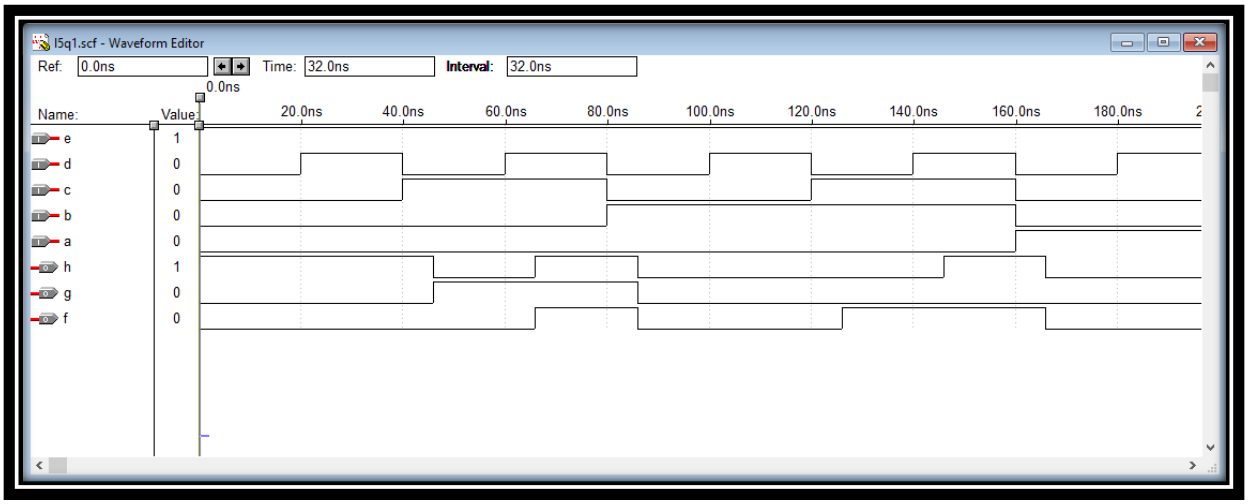
1. Design and simulate a combinational circuit with external gates and a 4 to 16 decoder built using a decoder tree of 2 to 4 decoders to implement the functions below. F= ab'c + a'cd + bcd' , G=acd' + a'b'c and H=a'b'c' + abc + a'cd

Verilog code :

```
module dec2to4(x,y,e,f);
input x,y,e;
output [3:0]f;
reg [3:0]f;
always @ (x or y)
begin
f=0;
if(e==1)
begin
case({x,y})
0:f[0]=1;
1:f[1]=1;
2:f[2]=1;
3:f[3]=1;
endcase
end
```

```
end
endmodule
module l5q1(a,b,c,d,e,f,g,h);
input a,b,c,d,e;
output f,g,h;
wire [3:0]m;
wire [15:0]y;
dec2to4 stage0(a,b,e,m);
dec2to4 stage1(c,d,m[0],y[3:0]);
dec2to4 stage2(c,d,m[1],y[7:4]);
dec2to4 stage3(c,d,m[2],y[11:8]);
dec2to4 stage4(c,d,m[3],y[15:12]);
assign f=y[3]|y[6]|y[7]|y[10]|y[11]|y[14];
assign g=y[2]|y[3]|y[10]|y[14];
assign h=y[0]|y[1]|y[3]|y[7]|y[14]|y[15];
endmodule
```

Output :



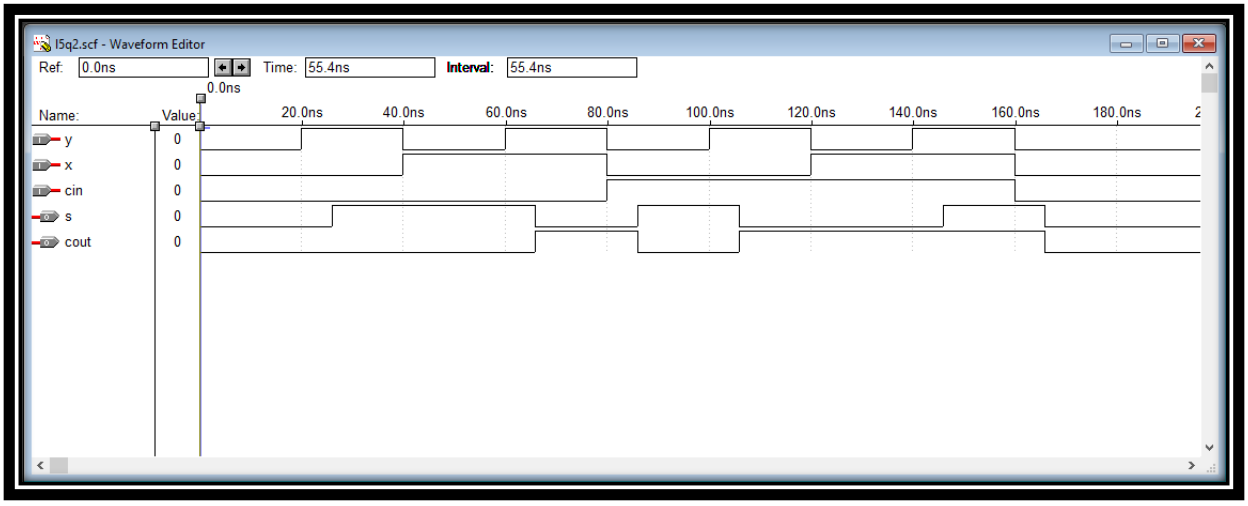
2. Design and implement a full adder using 2 to 4 decoder(s) and other gates.

Verilog code :

```
module dec2to4(x,y,e,f);
input x,y,e;
output [3:0]f;
reg [3:0]f;
always @ (x or y)
begin
f=0;
if(e==1)
begin
case({x,y})
0:f[0]=1;
1:f[1]=1;
2:f[2]=1;
3:f[3]=1;
```

```
endcase
end
end
endmodule
module l5q2(cin,x,y,s,cout);
input cin,x,y;
output s,cout;
wire [7:0]m;
dec2to4 stage1(x,y,~cin,m[3:0]);
dec2to4 stage2(x,y,cin,m[7:4]);
assign s=m[1]|m[2]|m[4]|m[7];
assign cout=m[3]|m[5]|m[6]|m[7];
endmodule
```

Output :



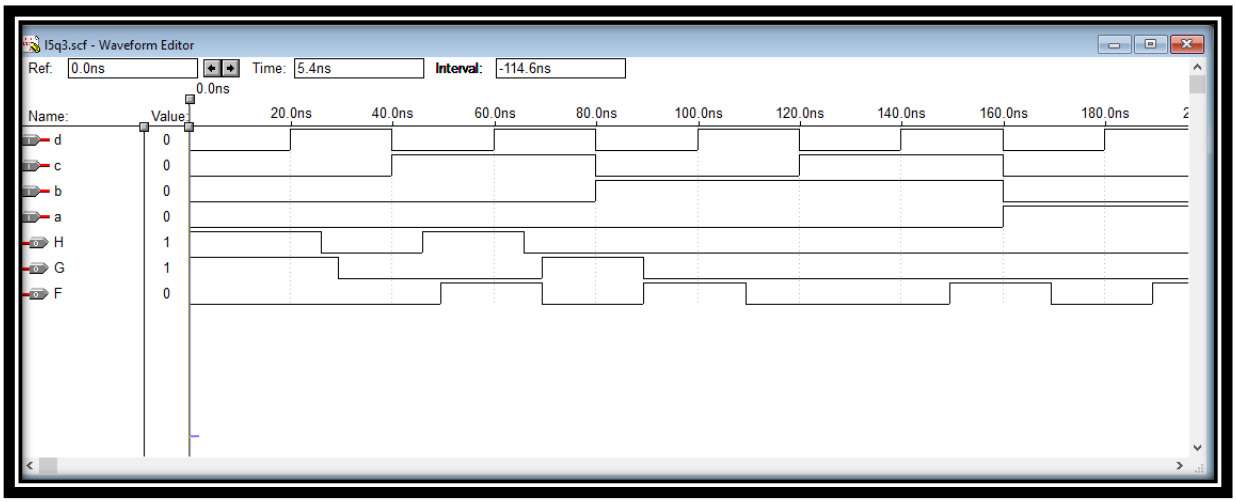
3. Design and simulate the circuit with 3 to 8 decoder(s) and external gates to implement the functions below. F(a, b, c, d)= Σm(2,4,7,9) G (a, b, c, d)= Σm (0,3,15) H(a, b, c, d)= Σm(0,2,10,12)

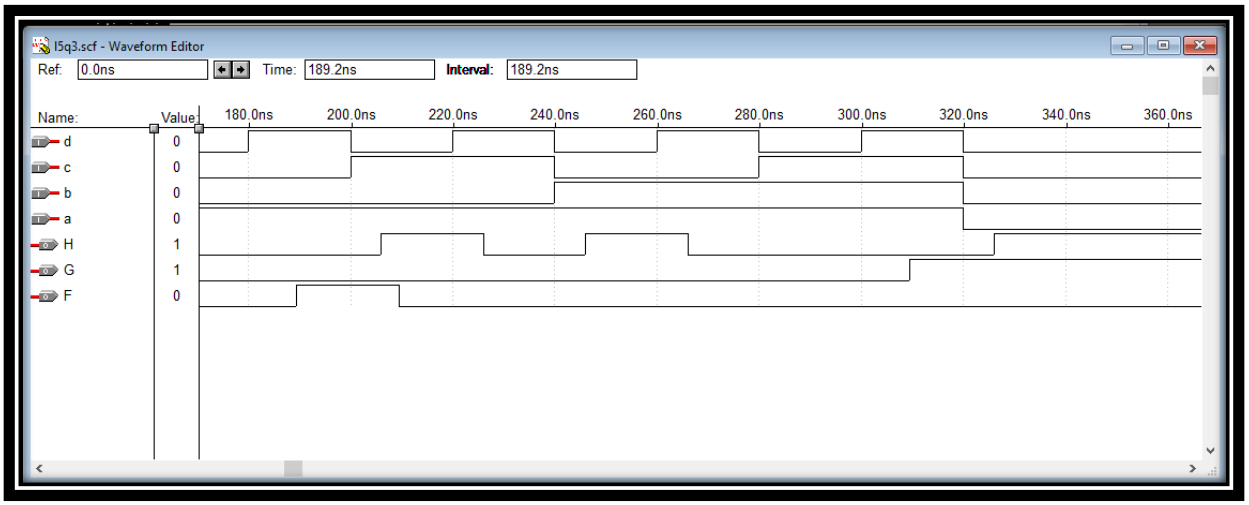
Verilog code :

```
module dec3to8(q,e,f);
input [2:0]q;
input e;
output [7:0]f;
integer k;
reg [7:0]f;
always @ (q or e)
begin
f=0;
if(e==1)
for(k=0;k<8;k=k+1)
if(q==k)
f[k]=1;
end
endmodule
module l5q3(a,b,c,d,F,G,H);
input a,b,c,d;
output F,G,H;
wire [2:0]q;
wire [15:0]y;
assign q={b,c,d};
dec3to8 stage1(q,~a,y[7:0]);
dec3to8 stage2(q,a,y[15:8]);
```

```
assign F=y[2]|y[4]|y[7]|y[9];
assign G=y[0]|y[3]|y[15];
assign H=y[0]|y[2]|y[10]|y[12];
endmodule
```

Output :





****

LAB – 6

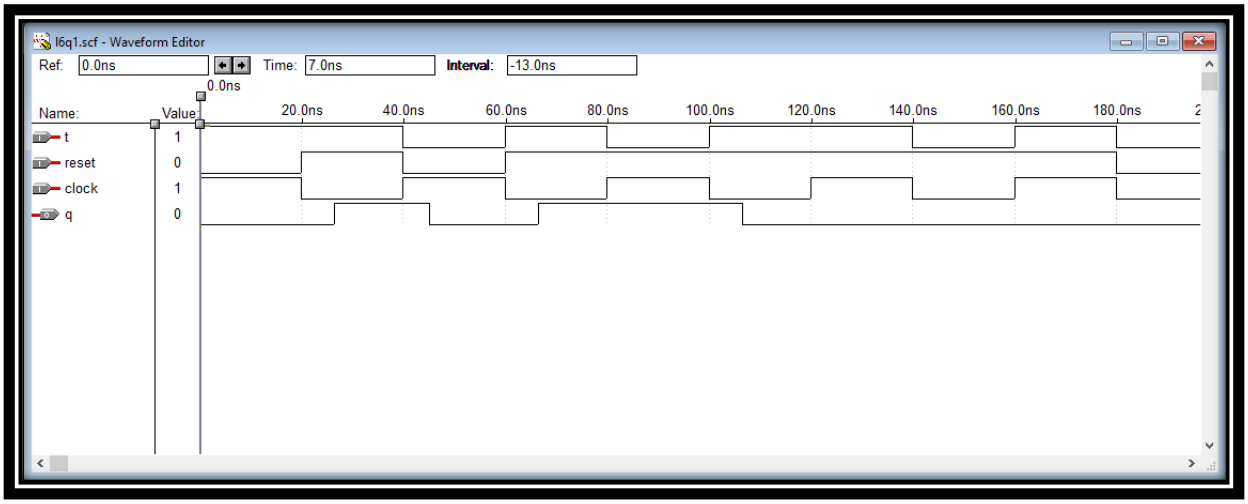
Exercise :

1. Write behavioral Verilog code for a negative edge triggered T FF with asynchronous active low reset.

Verilog code:

```
module l6q1(t,clock,reset,q);
input t,clock,reset;
output q;
reg q;
always@ (negedge clock or negedge reset)
begin
if(!reset)
q<=0;
else
case(t)
0:q<=q;
1:q<=~q;
endcase
end
endmodule
```

Output:



2. Write behavioral Verilog code for a positive edge-triggered JK FF with synchronous active high reset.

Verilog code:

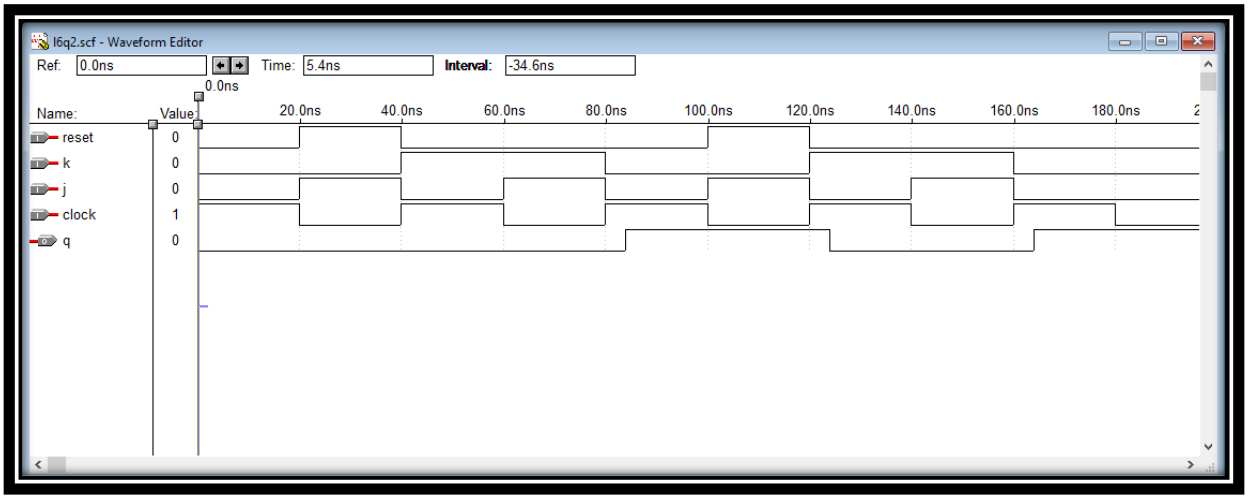
```
module l6q2(j,k,clock,reset,q);
input j,k,clock,reset;
output q;
reg q;
always@ (posedge clock)
begin
if(reset)
q<=0;
else
case({j,k})
0:q<=q;
1:q<=0;
2:q<=1;
3:q<=~q;
```

endcase

end

endmodule

Output:



3. Design and simulate the following counters

a) 4-bit ring counter.

Verilog code:

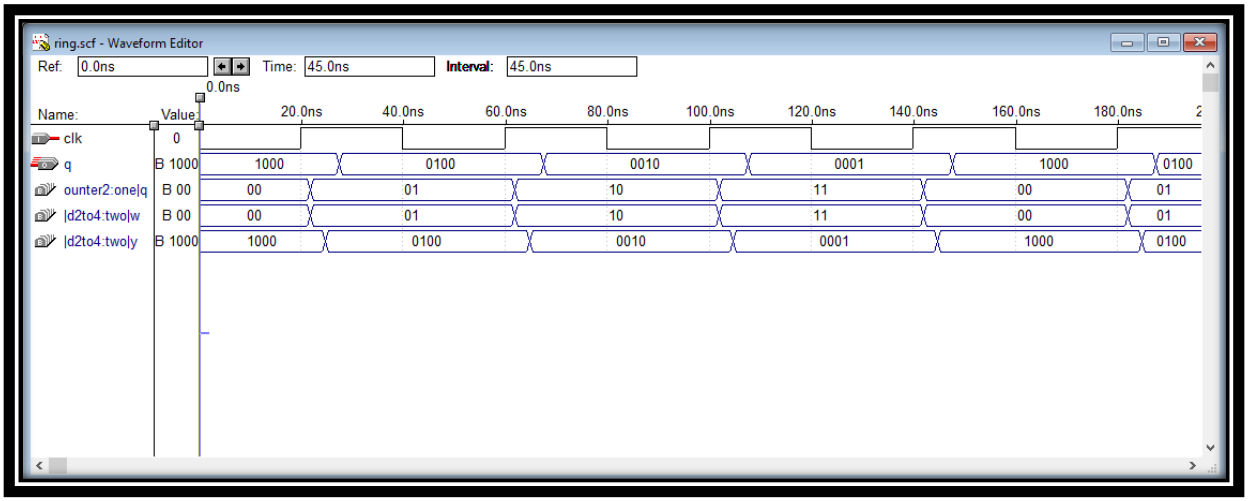
```
module ring(clk,q);
input clk;
output[3:0]q;
wire [1:0]count;
wire en=1;
bitcounter2 one(clk,count[1:0]);
d2to4 two(en,count[1:0],q[3:0]);
endmodule
module dff11(clk,d,q);
input clk,d;
output q;
```

```
reg q;
always@(posedge clk)
begin
q=d;
end
endmodule
module d2to4(en,w,y);
input en;
input [1:0]w;
output [3:0]y;
reg [3:0]y;
always@(en or w)
begin
y=0;
if(en==1)
case(w)
0:y[3]=1;
1:y[2]=1;
2:y[1]=1;
3:y[0]=1;
endcase
end
endmodule
module bitcounter2(clk,q);
input clk;
output[1:0]q;
dff11 ffo(clk,q[1]^q[0],q[1]);
```

dff11 ff1(clk,~q[0],q[0]);

endmodule

Output:



b) 5 bit Johnson counter.

Verilog code:

module john(clk,q);

input clk;

output [4:0]q;

dff1 ffo(clk,~q[4],q[0]);

dff1 ff1(clk,q[0],q[1]);

dff1 ff2(clk,q[1],q[2]);

dff1 ff3(clk,q[2],q[3]);

dff1 ff4(clk,q[3],q[4]);

endmodule

module dff1(clk,d,q);

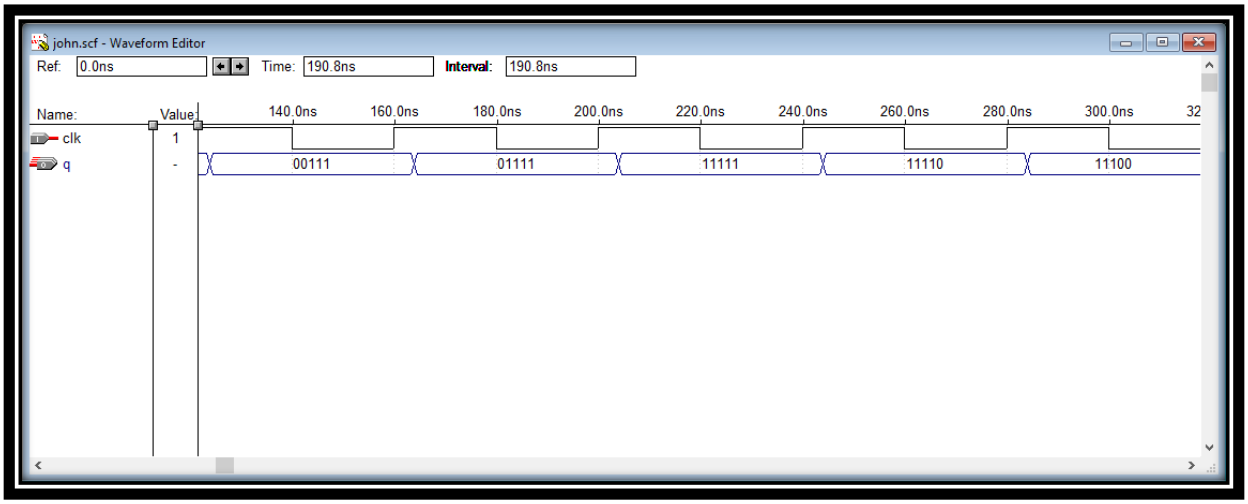
input clk,d;

output q;

reg q;

```
always@(posedge clk)
begin
q=d;
end
endmodule
```

Output:



****