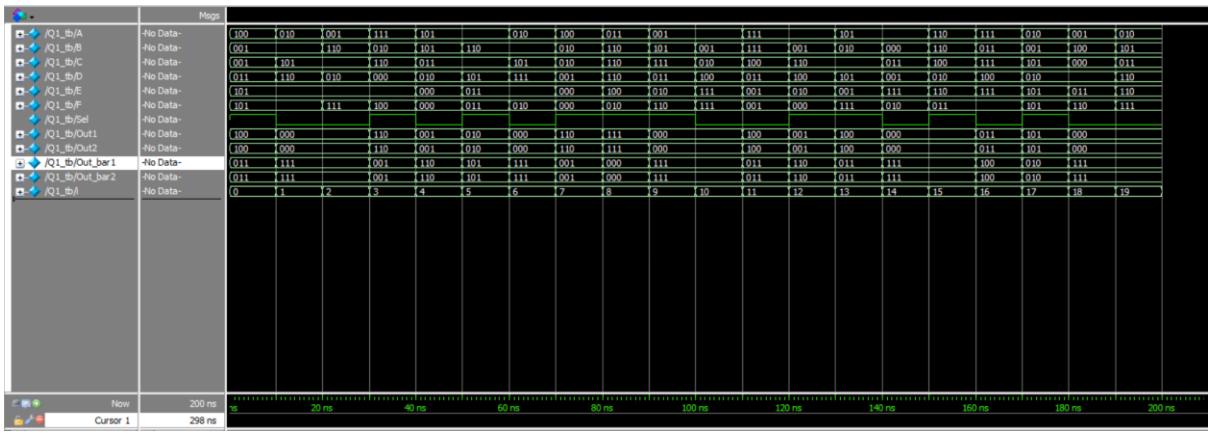
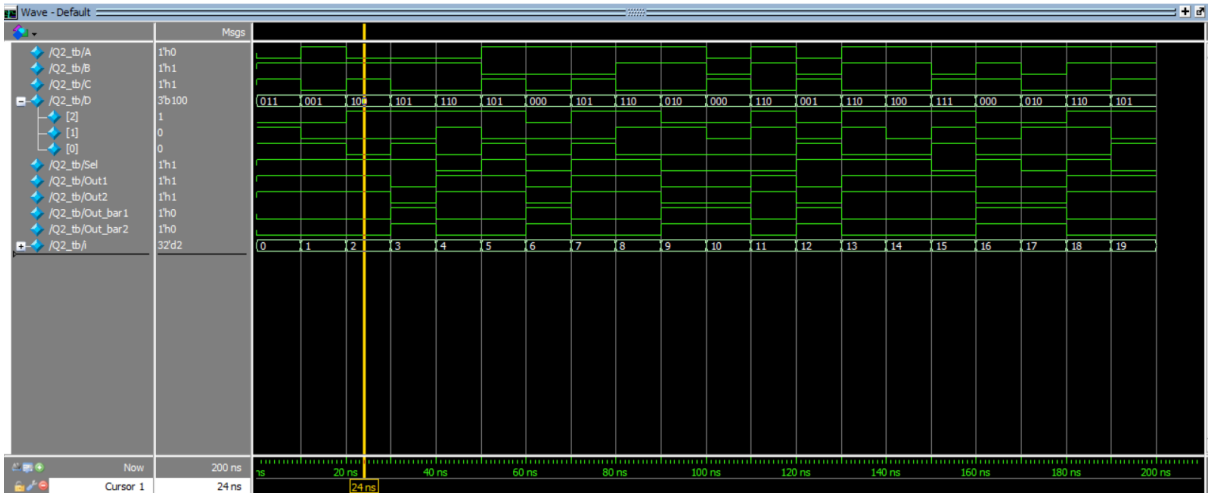
```
/*****************************Q1************************************/
module Q1_GM(A,B,C,D,E,F,Sel,Out,Out_bar);
parameter W=1;
input [W-1:0]A,B,C,D,E,F;
input Sel;
output[W-1:0] Out,Out_bar;
wire [W-1:0] r1,r2;
assign Out_bar=~Out;

assign Out=(Sel)?(D~^E~^F):A&B&C;
endmodule

module Q1_BM(A,B,C,D,E,F,Sel,Out,Out_bar);
parameter W=1;
input [W-1:0]A,B,C,D,E,F;
input  Sel;
output reg[W-1:0] Out,Out_bar;
always @(*) begin
        if (Sel)
        Out=(D~^E~^F);
        else
        Out=A&B&C;
        Out_bar=~Out;
end
endmodule

module Q1_tb();
parameter W=3;
reg [W-1:0]A,B,C,D,E,F;
reg Sel;
wire[W-1:0] Out1,Out_bar1,Out2,Out_bar2;
```

```
Q1_GM #(W)dut1(A,B,C,D,E,F,Sel,Out1,Out_bar1);

Q1_BM #(W)dut2(A,B,C,D,E,F,Sel,Out2,Out_bar2);

integer i=0;

initial begin

for(i=0;i<20;i=i+1)begin

A=$random;

B=$random;

C=$random;

D=$random;

E=$random;

F=$random;

Sel=$random;

#10

if(Out1!=Out2)begin

$display("Error i= %d , A=%b  ,  B=%b   , D=%b   , E=%b   , F=%b   , Sel=%b   , Out1=%b   , Out2=%b
",i,A,B,C,D,E,F,Out1,Out2);

$stop;

end

end

$stop;


end


endmodule
```



```
/**************************************************Q2**************************
******************************/

module Q2_DM(D,A,B,C,Sel,Out,Out_bar);
input [2:0]D;
input A,B,C;
input Sel;
output Out,Out_bar;
wire  r1,r2;
not(Out_bar,Out);
or(r1,D[2],D[0]&D[1]);
xnor(r2,A,B,C);
assign Out=(Sel)?r2:r1;
endmodule

module Q2_BM(D,A,B,C,Sel,Out,Out_bar);
input [2:0]D;
input A,B,C;
input Sel;
output reg Out,Out_bar;
always @(*) begin
        if (Sel)
        Out=~(A^B^C);
        else
        Out=(D[0]&D[1])|D[2];
        Out_bar=~Out;
end
endmodule
module Q2_tb();
reg A,B,C;
reg [2:0]D;
```

```
reg Sel;

wire Out1,Out_bar1,Out2,Out_bar2;

Q2_DM dut1(D,A,B,C,Sel,Out1,Out_bar1);

Q2_BM dut2(D,A,B,C,Sel,Out2,Out_bar2);

integer i=0;

initial begin

for(i=0;i<20;i=i+1)begin

A=$random;

B=$random;

C=$random;

D=$random;

Sel=$random;

#10

if(Out1!=Out2)begin

$display("Error i= %d , A=%b  ,  B=%b   , C=%b   ,D=%b  , Sel=%b   , Out1=%b   , Out2=%b
",i,A,B,C,D,Sel,Out1,Out2);

$stop;

end

end

$stop;

end

endmodule
```



```
/*************************************************Q3*****************************
*****************************/

module Q3_DM(X,Y);
input[3:0]X;
output[1:0]Y;

assign Y=(X[3])?2'b11:(X[2])?2'b10:(X[1])?2'b01:2'b00;
endmodule

module Q3_BM(X,Y);
input[3:0]X;
output reg [1:0]Y;
always @(*)begin
if(X[3])
Y=2'b11;
else if(X[2])
Y=2'b10;
else if(X[1])
Y=2'b01;
else
Y=2'b00;
end
endmodule

module Q3_tb();
reg [3:0]X;
wire [1:0]Y1,Y2;
Q3_DM dut1(X,Y1);
Q3_BM dut2(X,Y2);
integer i=0;
```

```
initial begin
for(i=0;i<20;i=i+1)begin
X=$random;
#10
if(Y1!=Y2)begin
        $display("Error X=%b     Y1=%b  Y2=%b",X,Y1,Y2);
        $stop;
        end
end
$stop;
end
initial begin
        $monitor("X=%b          Y1=%b  Y2=%b",X,Y1,Y2);
end
endmodule
```

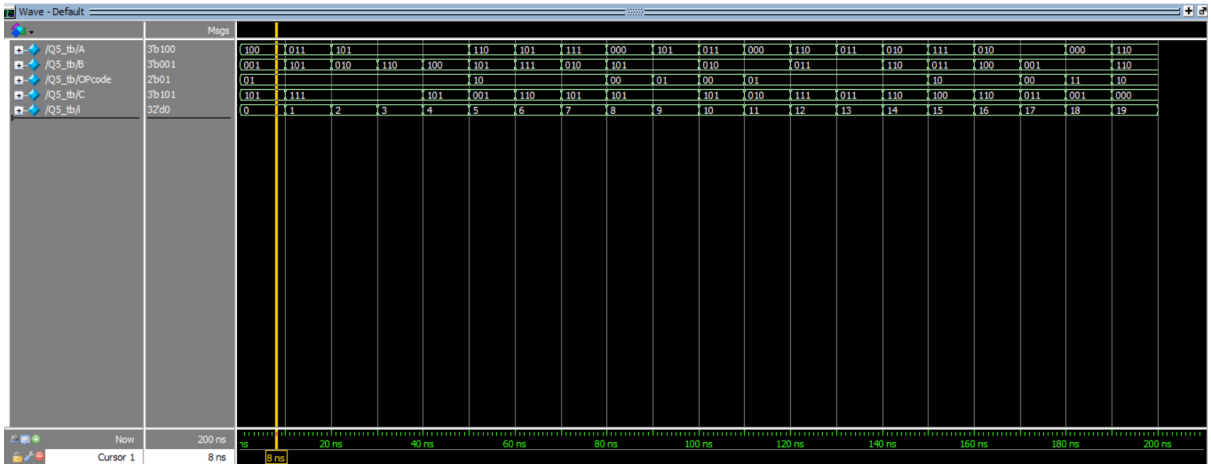
```
/*********************************************Q4*********************************
*****************************/
module Q4_DM(A,B,C);
parameter N=1;
input  [N-1:0]A,B;
output [N-1:0]C;
assign C=A+B;
endmodule

module Q4_BM(A,B,C);
parameter N=1;
input  [N-1:0]A,B;
output reg [N-1:0]C;
always @(*) begin
C=A+B;
end
endmodule

module Q4_tb();
parameter N=3;
reg  [N-1:0]A,B;
wire [N-1:0]C1,C2;
Q4_DM #(N) dut1(A,B,C1);
Q4_BM #(N) dut2(A,B,C2);
integer i=0;
initial begin
for(i=0;i<20;i=i+1)begin
A=$random;
B=$random;
#10
if(C1!=C2)begin
```

```
        $display("Error A=%b    B=%b    C1=%b  C2=%b",A,B,C1,C2);
        $stop;
        end
end
$stop;
end
initial begin
        $monitor("A=%b          B=%b    C1=%b  C2=%b",A,B,C1,C2);
end
endmodule
```

```
/*********************************************Q5*********************************
*******************************/

module Q5_ALU_SM(A,B,OPcode,C);
parameter N=3;
input [N-1:0]A,B;
input [1:0]OPcode;
output [N-1:0]C;
wire [N-1:0]r1;
Q4_BM #(N) dut1(A,B,r1);
assign C=(OPcode[0])?(OPcode[1])?A^B:A|B:(OPcode[1])?A-B:r1;
endmodule

module Q5_tb();
parameter N=3;
reg  [N-1:0]A,B;
reg [1:0]OPcode;
wire [N-1:0]C;
Q5_ALU_SM #(N) dut(A,B,OPcode,C);
integer i=0;
initial begin
for(i=0;i<20;i=i+1) begin
A=$random;
B=$random;
OPcode=$random;
#10;
end
$stop;
end
initial begin
        $monitor("A=%b          B=%b    OPcode=%b       C2=%b",A,B,OPcode,C);
```

end

endmodule



/**********Q6**********/

module Q6_seg_SM(A,B,OPcode,ENable,a,b,c,d,e,f,g);

input [3:0]A,B;

input [1:0]OPcode;

input ENable;

output reg a,b,c,d,e,f,g;

wire [3:0] C;

Q5_ALU_SM #(4) dut(A,B,OPcode,C);

initial begin

	$display("C	E a b c d e f g");

	$monitor("%h %b %b %b %b %b %b %b ",C,ENable,a,b,c,d,e,f,g);

end

always @(*) begin

	if (~ENable) begin

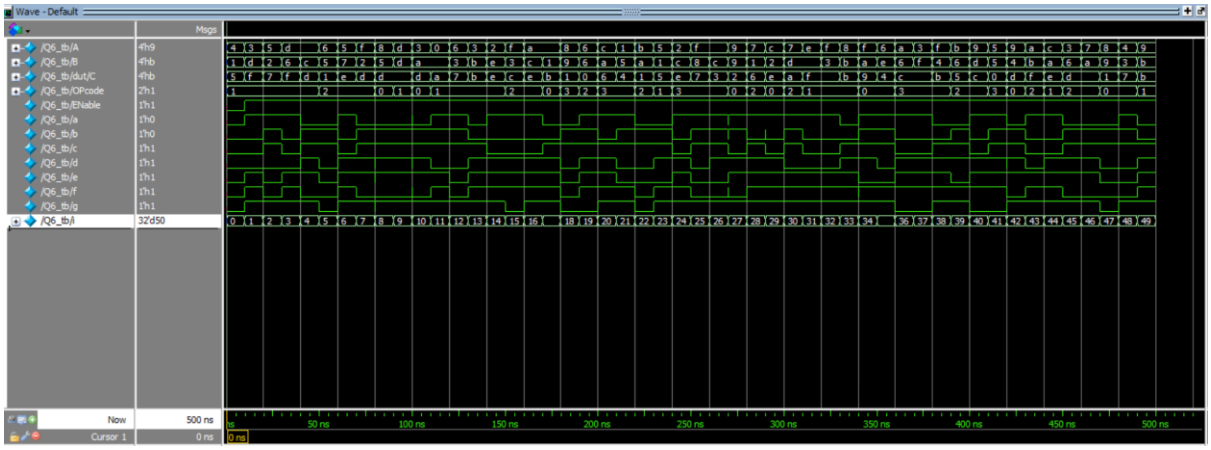
		{a,b,c,d,e,f,g}=7'b0;

	end

	else if (ENable) begin

```
        case(C)
        4'h0:{a,b,c,d,e,f,g}=7'b111_1110;
        4'h1:{a,b,c,d,e,f,g}=7'b011_0000;
        4'h2:{a,b,c,d,e,f,g}=7'b110_1101;
        4'h3:{a,b,c,d,e,f,g}=7'b111_1001;
        4'h4:{a,b,c,d,e,f,g}=7'b111_0011;
        4'h5:{a,b,c,d,e,f,g}=7'b011_1011;
        4'h6:{a,b,c,d,e,f,g}=7'b101_1111;
        4'h7:{a,b,c,d,e,f,g}=7'b111_0001;
        4'h8:{a,b,c,d,e,f,g}=7'b111_1111;
        4'h9:{a,b,c,d,e,f,g}=7'b111_1011;
        4'hA:{a,b,c,d,e,f,g}=7'b111_0111;
        4'hB:{a,b,c,d,e,f,g}=7'b001_1111;
        4'hC:{a,b,c,d,e,f,g}=7'b100_1110;
        4'hD:{a,b,c,d,e,f,g}=7'b011_1101;
        4'hE:{a,b,c,d,e,f,g}=7'b100_1111;
        4'hF:{a,b,c,d,e,f,g}=7'b100_0111;
        default : {a,b,c,d,e,f,g}=7'b0;
        endcase
        end
end
endmodule

module Q6_tb();
reg [3:0]A,B;
reg [1:0]OPcode;
reg ENable;
wire a,b,c,d,e,f,g;
Q6_seg_SM dut(A,B,OPcode,ENable,a,b,c,d,e,f,g);
integer i=0;
initial begin
```

```
ENable=0;
for(i=0;i<50;i=i+1) begin
A=$random;
B=$random;
OPcode=$random;
#10;
ENable=1;
end
$stop;
end
endmodule
```



C Enable a ,b,c,d,e,f,g

Transcript

```
VSIM 33> run -all
# C     E  a  b  c  d  e  f  g
# 5     0  0  0  0  0  0  0  0
# f     1  1  0  0  0  1  1  1
# 7     1  1  1  1  0  0  0  1
# f     1  1  0  0  0  1  1  1
# d     1  0  1  1  1  1  0  1
# 1     1  0  1  1  0  0  0  0
# e     1  1  0  0  1  1  1  1
# d     1  0  1  1  1  1  0  1
# d     1  0  1  1  1  1  0  1
# d     1  0  1  1  1  1  0  1
# a     1  1  1  1  0  1  1  1
# 7     1  1  1  1  0  0  0  1
# b     1  0  0  1  1  1  1  1
# e     1  1  0  0  1  1  1  1
# c     1  1  0  0  1  1  1  0
# e     1  1  0  0  1  1  1  1
# b     1  0  0  1  1  1  1  1
# 1     1  0  1  1  0  0  0  0
# 0     1  1  1  1  1  1  1  0
# 6     1  1  0  1  1  1  1  1
# 4     1  1  1  1  0  0  1  1
# 1     1  0  1  1  0  0  0  0
# 5     1  0  1  1  1  0  1  1
# e     1  1  0  0  1  1  1  1
# 7     1  1  1  1  0  0  0  1
# 3     1  1  1  1  1  0  0  1
# 2     1  1  1  0  1  1  0  1
# 6     1  1  0  1  1  1  1  1
# e     1  1  0  0  1  1  1  1
# a     1  1  1  1  0  1  1  1
# f     1  1  0  0  0  1  1  1
# b     1  0  0  1  1  1  1  1
# 9     1  1  1  1  1  0  1  1
# 4     1  1  1  1  0  0  1  1
# c     1  1  0  0  1  1  1  0
# b     1  0  0  1  1  1  1  1
# 5     1  0  1  1  1  0  1  1
# c     1  1  0  0  1  1  1  0
# 0     1  1  1  1  1  1  1  0
# d     1  0  1  1  1  1  0  1
# f     1  1  0  0  0  1  1  1
# e     1  1  0  0  1  1  1  1
# d     1  0  1  1  1  1  0  1
# 1     1  0  1  1  0  0  0  0
```