**source code for two bit adder**

module not\_gate(abar,a);

input a;

output abar;

not x1(abar,a);

endmodule

module and\_gate(c,a,b);

input a,b;

output c;

and r1(c,a,b);

endmodule

module or\_gate(c,a,b);

input a,b;

output c;

or r1(c,a,b);

endmodule

module xor\_gate(out,a,b);

input a,b;

output out;

wire abar,bbar,y1,y2;

not\_gate xx1(abar,a);

not\_gate xx2(bbar,b);

and\_gate xx3(y1,a,bbar);

and\_gate xx4(y2,abar,b);

or\_gate xx5(out,y1,y2);

endmodule

module half\_adder (a,b,sum,carry);

input a,b;

output sum;

output carry;

xor\_gate xx1(sum,a,b);

and\_gate xx2 (carry,a,b);

endmodule

module full\_adder(a,b,cin,fsum,fcarry);

input a,b,cin;

output fsum,fcarry;

wire s\_1,c\_1;

half\_adder xx1 (a,b,s\_1,c\_1);

half\_adder xx2(s\_1,cin,fsum,c\_2);

or\_gate xx3 (fcarry,c\_1,c\_2);

endmodule

module full\_adder\_withoutcarry(a,b,cin,s);

input a,b,cin;

output s;

wire s\_1;

xor\_gate x1 (s\_1,a,b);

xor\_gate x2 (s,s\_1,cin);

endmodule

module two\_bit\_adder(a,b,cin,sum,carry);

input [1:0]a,b;

input cin;

output carry;

output [1:0] sum;

wire c1;

full\_adder x1(a[0],b[0],cin,sum[0],c1);

full\_adder\_withoutcarry x2(a[1],b[1],c1,sum[1]);

two\_bit\_carry x3(a,b,cin,carry);

endmodule

module two\_bit\_carry(a,b,cin,carry);

input [1:0]a,b;

input cin;

output carry;

wire x1,x2,x3,x4,x5,x6,x7,x8;

and\_gate a1(x1,a[1],b[1]);

and\_gate a2(x2,a[0],b[0]);

or\_gate a3(x3,a[1],b[1]);

or\_gate a4(x4,a[0],b[0]);

and\_gate a5(x5,x4,x3);

and\_gate a6(x6,x2,x5);

or\_gate a7(x7,x1,x6);

and\_gate a8(x8,x5,cin);

or\_gate a9(carry,x7,x8);

endmodule

module two\_bit\_adder\_tb;

reg [1:0]a,b;

reg cin;

wire carry;

wire [1:0]sum;

reg clk;

integer i,j,k;

two\_bit\_adder s1(a,b,cin,sum,carry);

always

#10 clk=~clk;

initial

begin

clk=0;

a=0;

b=0;

cin=0;

for(i=0;i<4;i=i+1)

for(j=0;j<4;j=j+1)

for(k=0;k<2;k=k+1)

begin

a=i;

b=j;

cin=k;

#20;

$monitor("a=%h, b=%h, cin=%b, carry=%b, sum=%b",a,b,cin,carry,sum);

end

end

endmodule