**source code for five 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 five\_bit\_adder(a,b,cin,sum,carry);

input [4:0]a,b;

input cin;

output carry;

output [4:0] sum;

wire c1,c2,c3,c4;

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

full\_adder x2(a[1],b[1],c1,sum[1],c2);

full\_adder x3(a[2],b[2],c2,sum[2],c3);

full\_adder x4 (a[3],b[3],c3,sum[3],c4);

full\_adder\_withoutcarry x5(a[4],b[4],c4,sum[4]);

five\_bit\_carry x6(a,b,cin,carry);

endmodule

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

input [4:0]a,b;

input cin;

output carry;

wire x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x14,x15,x16,x17,x18,x19,x20,x21,x22,x23,x24,x25,x26,x27,x28,x29,x30,x31,x32;

and\_gate a1(x1,a[4],b[4]);

and\_gate a2 (x2,a[3],b[3]);

and\_gate a3 (x3,a[2],b[2]);

and\_gate a4 (x4,a[1],b[1]);

and\_gate a5 (x5,a[0],b[0]);

or\_gate a6 (x6,a[4],b[4]);

or\_gate a7 (x7,a[3],b[3]);

or\_gate a8 (x8,a[2],b[2]);

or\_gate a9 (x9,a[1],b[1]);

or\_gate a10 (x10,a[0],b[0]);

and\_gate a11 (x11,x6,x7);

and\_gate a12(x12,x8,x9);

and\_gate a13(x13,x12,x10);

and\_gate a14(x14,x11,x13);

and\_gate a15(x15,x5,x14);

and\_gate a16(x16,x4,x14);

and\_gate a17(x17,x14,x3);

and\_gate a18(x18,x2,x14);

and\_gate a19(x19,x8,x4);

and\_gate a20(x20,x19,x7);

and\_gate a21(x21,x3,x20);

and\_gate a22(x22,x6,x3);

and\_gate a23(x23,x22,x7);

and\_gate a24(x24,x6,x2);

or\_gate a25(x25,x21,x23);

or\_gate a26 (x26,x25,x24);

or\_gate a27(x27,x17,x16);

or\_gate a28(x28,x27,x15);

or\_gate a29(x29,x18,x28);

and\_gate a30(x30,x14,cin);

or\_gate a32(x32,x1,x26);

or\_gate a31(x31,x29,x32);

or\_gate a33(carry,x30,x31);

endmodule

module five\_bit\_adder\_tb;

reg [4:0]a,b;

reg cin;

wire sum;

wire carry;

reg clk;

integer i,j,k;

five\_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<32;i=i+1)

for(j=0;j<32;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