**Full adder:**

module full\_adder(

input a,

input b,

input c,

output s,

output carry

);

assign s = a ^ b ^ c;

assign carry = (a & b) | (b & c) | (a & c);

Endmodule

**TEST bench**

module tb\_full\_adder();

reg a,b,c;

wire s,carry;

full\_adder uut(.a(a),.b(b),.c(c),.s(s),.carry(carry));

initial

begin

$monitor($time,"a=%b,b=%b,c=%b,s=%b,carry=%b",a,b,c,s,carry);

#5 a=0;b=0;c=0;

#5 a=0;b=0;c=1;

#5 a=0;b=1;c=0;

#5 a=0;b=1;c=1;

#5 a=1;b=0;c=0;

#5 a=1;b=0;c=1;

#5 a=1;b=1;c=0;

#5 a=1;b=1;c=1;

#5 $finish;

end

Endmodule

**Half adder:**

module half(a,b,s,c);

input a,b;

output s,c;

xor x1(s,a,b);

and x2(c,a,b);

endmodule

**Test bench:**

timescale 1ns / 1ps

module half\_tb;

// Inputs

reg a;

reg b;

// Outputs

wire s;

wire c;

// Instantiate the Unit Under Test (UUT)

half uut (

.a(a),

.b(b),

.s(s),

.c(c)

);

initial begin

$display("A B | S C");

$display("-----------");

// Apply inputs

a = 0; b = 0; #10;

$display("%b %b | %b %b", a, b, s, c);

a = 0; b = 1; #10;

$display("%b %b | %b %b", a, b, s, c);

a = 1; b = 0; #10;

$display("%b %b | %b %b", a, b, s, c);

a = 1; b = 1; #10;

$display("%b %b | %b %b", a, b, s, c);

$finish;

end