1:

module fulladder(cin,x,y,sum,cout);

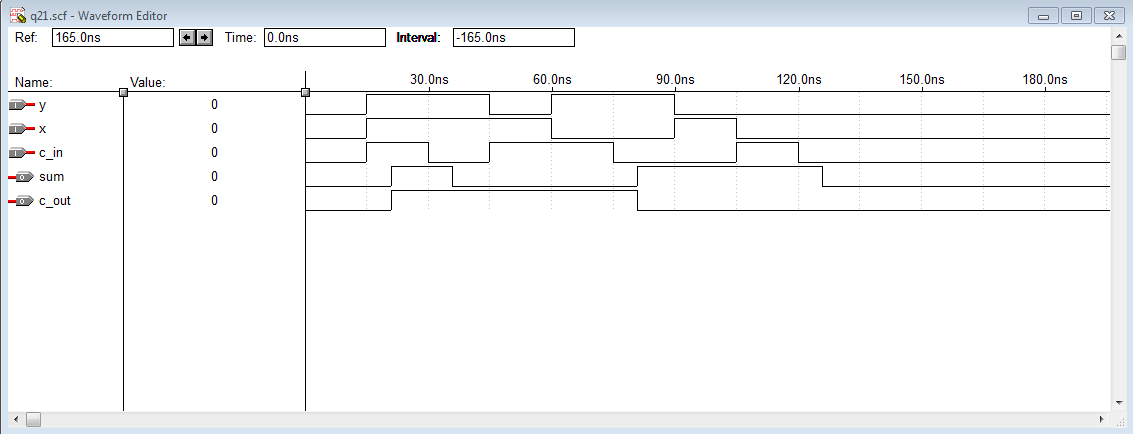
input cin,x,y;

output sum,cout;

assign sum = cin^x^y;

assign cout =(x&y) | (x&cin) |(y&cin);

endmodule



2:

module addsub(cin,x,y,s,cout);

input cin;

input[3:0]x,y;

output[3:0]s;

output cout;

wire[3:1]c;

fulladder stage0 (cin,x[0],y[0]^cin,s[0],c[1]);

fulladder stage1 (c[1],x[1],y[1]^cin,s[1],c[2]);

fulladder stage2 (c[2],x[2],y[2]^cin,s[2],c[3]);

fulladder stage3 (c[3],x[3],y[3]^cin,s[3],cout);

endmodule

module fulladder(cin,x,y,sum,cout);

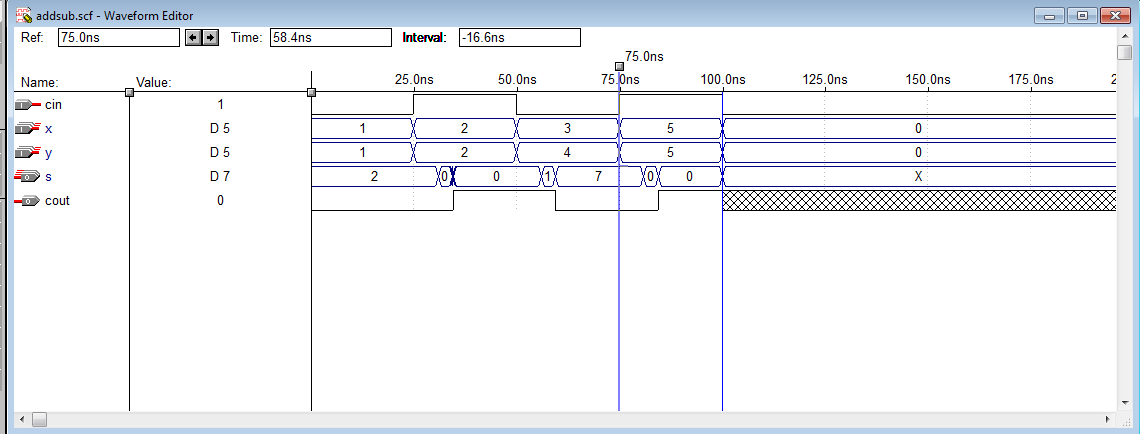
input cin,x,y;

output sum,cout;

assign sum = cin^x^y;

assign cout =(x&y) | (x&cin) |(y&cin);

endmodule



3:

module bcdadd(cin,a,b,sum,cout);

input [3:0]a,b;

input cin;

output [3:0]sum;

output cout;

wire[3:0]z;

wire m;

wire[3:1]k;

wire[3:0]h;

adderm1 stage0 (cin,a,b,z,m);

assign k[1]=z[3]&z[2];

assign k[2]=z[3]&z[1];

assign k[3]=(m|k[1]|k[2]);

assign h[0]=0,h[1]=k[3],h[2]=k[3],h[3]=0;

adderm1 stage1 (cin,z,h,sum,cout);

endmodule

module adderm1(cin,x,y,s,cout);

input cin;

input [3:0]x,y;

output cout;

output [3:0]s;

wire[3:1]c;

fulladder stage0 (cin,x[0],y[0],s[0],c[1]);

fulladder stage1 (c[1],x[1],y[1],s[1],c[2]);

fulladder stage2 (c[2],x[2],y[2],s[2],c[3]);

fulladder stage3 (c[3],x[3],y[3],s[3],cout);

endmodule

module fulladder(cin,x,y,sum,cout);

input cin,x,y;

output sum,cout;

assign sum = cin^x^y;

assign cout =(x&y) | (x&cin) |(y&cin);

endmodule

