**Wallace 4 bit**

module wallace(

input [3:0] a,

input [3:0] b,

output [7:0] c

);

wire [4:0]s1,d1,s,d;

wire v;

//level 1

assign c[0]=a[0]&b[0];

ha ha1(a[0]&b[1],(a[1]&b[0]),c[1],d[0]);

fa fa1(a[0]&b[2],a[1]&b[1],a[2]&b[0],s[0],d[1]);

fa fa2(a[0]&b[3],a[1]&b[2],a[2]&b[1],s[1],d[2]);

fa fa3(a[1]&b[3],a[2]&b[2],a[3]&b[1],s[2],d[3]);

ha ha2(a[2]&b[3],a[3]&b[2],s[3],d[4]);

//level 2

ha ha3(s[0],d[0],c[2],d1[0]);

fa fa4(a[3]&b[0],s[1],d[1],s1[0],d1[1]);

ha ha4(s[2],d[2],s1[1],d1[2]);

ha ha5(s[3],d[3],s1[2],d1[3]);

ha ha6(a[3]&b[3],d[4],s1[3],d1[4]);

assign s1[4]=0;

//level3 adder

adder5 aq(s1,d1,c[7:3],v);

endmodule

module ha(x,y,z,q);

input x,y;

output z,q;

assign z=x^y;

assign q=x&y;

endmodule

module fa(x,y,j,z,q);

input x,y,j;

output z,q;

assign z=(x^y)^j;

assign q=(x&y)|(y&j)|(j&x);

endmodule

module adder5(

input [4:0]a,

input [4:0]b,

output [4:0]c,

output carry );

wire [4:0]p,g,c1;

assign p=a^b;

assign g=a&b;

assign c1[0]=0;

assign c1[1]= g[0];

assign c1[2]= g[1] | (p[1]&g[0]) ;

assign c1[3]= g[2] | (p[2]&g[1]) | p[2]&p[1]&g[0] ;

assign c1[4]= g[3] | p[3]&g[2] | (p[3]&p[2]&g[1]) | p[3]&p[2]&p[1]&g[0] ;

assign carry= g[4]|p[4]&g[3] | p[4]&p[3]&g[2] | (p[4]&p[3]&p[2]&g[1]) | p[4]&p[3]&p[2]&p[1]&g[0] ;

assign c=p^c1;

endmodule

**Wallace 8 bit**

module wallace(a,b,c);

input [7:0]a,b;

output [15:0]c;

wire [11:0]s,q,q1,s1,s2,q2,j,k,j1,k1,m,n,a1,b1;

//first stage

assign c[0]=a[0]&b[0];

ha ha1(a[0]&b[1],b[0]&a[1],c[1],s[0]);

fa fa1(a[1]&b[1],a[0]&b[2],a[2]&b[0],q[0],s[1]);

fa fa2(a[2]&b[1],a[1]&b[2],a[3]&b[0],q[1],s[2]);

fa fa3(a[2]&b[2],a[3]&b[1],a[1]&b[3],q[2],s[3]);

ha ha2(a[4]&b[0],a[0]&b[4],q1[2],s1[3]);

fa fa4(a[3]&b[2],a[2]&b[3],a[4]&b[1],q[3],s[4]);

fa fa5(a[1]&b[4],a[5]&b[0],a[0]&b[5],q1[3],s1[4]);

fa fa6(a[3]&b[3],a[4]&b[2],a[2]&b[4],q[4],s[5]);

fa fa7(a[5]&b[1],a[1]&b[5],a[6]&b[0],q1[4],s1[5]);

fa fa8(a[4]&b[3],a[3]&b[4],a[5]&b[2],q[5],s[6]);

fa fa9(a[2]&b[5],a[6]&b[1],a[1]&b[6],q1[5],s1[6]);

ha ha3(a[7]&b[0],a[0]&b[7],q2[5],s2[6]);

fa fa10(a[4]&b[4],a[5]&b[3],a[3]&b[5],q[6],s[7]);

fa fa11(a[6]&b[2],a[2]&b[6],a[7]&b[1],q1[6],s1[7]);

fa fa12(a[5]&b[4],a[4]&b[5],a[6]&b[3],q[7],s[8]);

fa fa13(a[3]&b[6],a[7]&b[2],a[2]&b[7],q1[7],s1[8]);

fa fa14(a[5]&b[5],a[6]&b[4],a[4]&b[6],q[8],s[9]);

ha ha4(a[7]&b[3],a[3]&b[7],q1[8],s1[9]);

fa fa15(a[6]&b[5],a[5]&b[6],a[7]&b[4],q[9],s[10]);

fa fa16(a[6]&b[6],a[7]&b[5],a[5]&b[7],q[10],s[11]);

//2nd stage

ha ha5(q[0],s[0],c[2],k[0]);

fa fa17(a[0]&b[3],q[1],s[1],j[0],k[1]);

fa fa18(q[2],q1[2],s[2],j[1],k[2]);

fa fa19(q[3],q1[3],s[3],j[2],k[3]);

fa fa20(q[4],q1[4],s[4],j[3],k[4]);

ha ha7(s1[4],a[0]&b[6],j1[3],k1[4]);

fa fa21(q[5],q1[5],s[5],j[4],k[5]);

ha ha6(s1[5],q2[5],j1[4],k1[5]);

fa fa22(q[6],s[6],q1[6],j[5],k[6]);

fa fa23(s1[6],a[1]&b[7],s2[6],j1[5],k1[6]);

fa fa24(q[7],q1[7],s[7],j[6],k[7]);

fa fa25(q[8],q1[8],s[8],j[7],k[8]);

fa fa26(q[9],s1[9],s[9],j[8],k[9]);

ha haa7(q[10],s[10],j[9],k[10]);

fa fa27(a[7]&b[6],a[6]&b[7],s[11],j[10],k[11]);

//3rd stage

ha ha8(j[0],k[0],c[3],n[0]);

ha ha9(j[1],k[1],m[0],n[1]);

fa fa28(s1[3],j[2],k[2],m[1],n[2]);

fa fa29(j[3],k[3],j1[3],m[2],n[3]);

fa fa30(j[4],k[4],j1[4],m[3],n[4]);

fa fa31(j[5],k[5],j1[5],m[4],n[5]);

fa fa32(j[6],k[6],s1[7],m[5],n[6]);

fa fa33(j[7],k[7],s1[8],m[6],n[7]);

fa fa34(j[8],k[8],a[4]&b[7],m[7],n[8]);

//4th stage

assign a1[2:0]=m[2:0];

assign b1[2:0]=n[2:0];

fa fa35(m[3],n[3],k1[4],a1[3],b1[4]);

assign b1[3]=1'b0;

fa fa36(m[4],n[4],k1[5],a1[4],b1[5]);

fa fa37(m[5],n[5],k1[6],a1[5],b1[6]);

ha ha10(m[6],n[6],a1[6],b1[7]);

ha ha11(m[7],n[7],a1[7],b1[8]);

fa fa38(n[8],j[9],k[9],a1[8],b1[9]);

ha ha12(j[10],k[10],a1[9],b1[10]);

ha ha13(k[11],a[7]&b[7],a1[10],b1[11]);

assign a1[11]=1'b0;

//laststage

//a1[11:0],b1[11:0] adder

adder12 aa1(a1,b1,c[15:4]);

endmodule

module adder12(a,b,sum);

input [11:0]a,b;

output [11:0]sum;

wire v;

assign v=0;

wire gg0,gp0,gg4,gp4,gg8,gp8,c4,c8;

cla4t cc0(a[3:0],b[3:0],v,sum[3:0],gg0,gp0);

cla4t cc1(a[7:4],b[7:4],c4,sum[7:4],gg4,gp4);

cla4t cc2(a[11:8],b[11:8],c8,sum[11:8],gg8,gp8);

assign c4=gg0 ;

assign c8=gg4 |(gg0 & gp4);

assign c12=gg8|gp8&gg4|gp8&gp4&gg0;

endmodule

module cla4t(a,b,cin,sum,gg,gp);

input [3:0]a,b;

input cin;

output [3:0]sum;

output gg,gp;

wire [3:0] p,g,c;

assign p=a^b;//propagate

assign g=a&b; //generate

assign c[0]=cin;

assign c[1]= g[0]|(p[0]&c[0]);

assign c[2]= g[1] | (p[1]&g[0]) | p[1]&p[0]&c[0];

assign c[3]= g[2] | (p[2]&g[1]) | p[2]&p[1]&g[0] | p[2]&p[1]&p[0]&c[0];

assign gp=&p;

assign gg=g[3] | (p[3]&g[2]) | (p[3]&p[2]&g[1]) | (p[3]&p[2]&p[1]&g[0]);

assign sum=p^c;

endmodule

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

input a,b;

output s,c;

assign s=a^b;

assign c=a&b;

endmodule

module fa(a,b,c,s,ca);

input a,b,c;

output s,ca;

assign s=a^b^c;

assign ca=a&b|b&c|c&a;

endmodule

**Wallace 16 bit**

module wallace16(

input [15:0] a,

input [15:0] b,

output [31:0] c

);

wire [27:0]s,d,x,y,m,n,a1,b1,o,p,o1,p1,c1;

wire [24:0]s1,d1,x1,y1;

wire [21:0]s2,d2,a2,b2;

wire [18:0]s3,d3,x2,y2;

wire [15:0]s4,d4,x3,y3,m1,n1,a3,b3;

wire vq;

genvar i,j;

wire [15:0]flag[15:0];

wire [15:0]g[30:0];

generate

for(i=0;i<16;i=i+1)

begin

assign flag[i]=(b[i]==1)?a:16'b0;

end

endgenerate

generate

for(i=0;i<16;i=i+1)

begin

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

begin

assign g[i][j]=flag[i-j][j];

end

end

endgenerate

generate

for(i=16;i<=30;i=i+1)

begin

for(j=0;j<=30-i;j=j+1)

begin

assign g[i][j]=flag[15-j][i+j-15];

end

end

endgenerate

//level 1

assign c[0]=g[0][0];

//block 1

ha ha1(g[1][0],g[1][1],c[1],d[0]);

generate

for(i=2;i<=27;i=i+1)

begin

fa fai(g[i][0],g[i][1],g[i][2],s[i-2],d[i-1]);

end

endgenerate

ha ha2(g[28][0],g[28][1],s[26],d[27]);

//block 2

ha ha3(g[4][3],g[4][4],s1[2],d1[3]);

generate

for(i=5;i<=24;i=i+1)

begin

fa fak(g[i][3],g[i][4],g[i][5],s1[i-2],d1[i-1]);

end

endgenerate

ha ha4(g[25][3],g[25][4],s1[23],d1[24]);

//block 3

ha ha5(g[7][6],g[7][7],s2[5],d2[6]);

generate

for(i=8;i<=21;i=i+1)

begin

fa faki(g[i][6],g[i][7],g[i][8],s2[i-2],d2[i-1]);

end

endgenerate

ha ha6(g[22][6],g[22][7],s2[20],d2[21]);

//block 4

ha ha7(g[10][9],g[10][10],s3[8],d3[9]);

generate

for(i=11;i<=18;i=i+1)

begin

fa faii(g[i][9],g[i][10],g[i][11],s3[i-2],d3[i-1]);

end

endgenerate

ha ha8(g[19][9],g[19][10],s3[17],d3[18]);

//block 5

ha ha9(g[13][12],g[13][13],s4[11],d4[12]);

generate

for(i=14;i<=15;i=i+1)

begin

fa faij(g[i][12],g[i][13],g[i][14],s4[i-2],d4[i-1]);

end

endgenerate

ha ha10(g[16][12],g[16][13],s4[14],d4[15]);

//level 2

//block 1

ha ha11(s[0],d[0],c[2],y[0]);

fa fas(s[1],d[1],g[3][3],x[0],y[1]);

generate

for(i=2;i<=23;i=i+1)

begin

fa fag(s[i],d[i],s1[i],x[i-1],y[i]);

end

endgenerate

fa fa2(s[24],d[24],d1[24],x[23],y[24]);

fa fa3(s[25],d[25],g[27][3],x[24],y[25]);

fa fa4(s[26],d[26],g[28][2],x[25],y[26]);

fa fa5(d[27],g[29][0],g[29][1],x[26],y[27]);

//block 2

ha ha12(g[6][6],d1[4],x1[3],y1[4]);

ha ha13(d1[5],s2[5],x1[4],y1[5]);

generate

for(i=6;i<=20;i=i+1)

begin

fa fog(s2[i],d1[i],d2[i],x1[i-1],y1[i]);

end

endgenerate

fa fa6(d2[21],g[23][6],g[23][7],x1[20],y1[21]);

ha ha14(d1[22],g[24][6],x1[21],y1[22]);

ha ha15(d1[23],g[25][5],x1[22],y1[23]);

ha ha16(g[26][3],g[26][4],x1[23],y1[24]);

//block 3

ha ha17(s3[9],d3[9],x2[8],y2[9]);

fa fa7(g[12][12],s3[10],d3[10],x2[9],y2[10]);

fa fa8(s3[11],d3[11],s4[11],x2[10],y2[11]);

fa fa9(s3[12],d3[12],s4[12],x2[11],y2[12]);

fa fa10(s3[13],d3[13],s4[13],x2[12],y2[13]);

fa fa11(s3[14],d3[14],s4[14],x2[13],y2[14]);

fa fa12(s3[15],d3[15],d4[15],x2[14],y2[15]);

fa fa13(s3[16],d3[16],g[18][12],x2[15],y2[16]);

fa fa14(s3[17],d3[17],g[19][11],x2[16],y2[17]);

fa fa15(d3[18],g[20][9],g[20][10],x2[17],y2[18]);

//block 4

ha ha18(d4[13],g[15][15],x3[12],y3[13]);

ha ha19(d4[14],g[16][14],x3[13],y3[14]);

ha ha20(g[17][12],g[17][13],x3[14],y3[15]);

//level 3

//block 1

ha ha21(x[0],y[0],c[3],b1[0]);

ha ha22(x[1],y[1],a1[0],b1[1]);

fa fa16(x[2],y[2],d1[3],a1[1],b1[2]);

generate

for(i=3;i<=23;i=i+1)

begin

fa fig(x[i],y[i],x1[i],a1[i-1],b1[i]);

end

endgenerate

fa fa17(x[24],y[24],y1[24],a1[23],b1[24]);

ha ha23(x[25],y[25],a1[24],b1[25]);

ha ha24(x[26],y[26],a1[25],b1[26]);

ha ha25(g[30][0],y[27],a1[26],b1[27]);

//block 2

ha ha26(y1[6],g[9][9],a2[5],b2[6]);

ha ha27(y1[7],s3[8],a2[6],b2[7]);

ha ha28(y1[8],x2[8],a2[7],b2[8]);

generate

for(i=9;i<=17;i=i+1)

begin

fa fiig(y1[i],x2[i],y2[i],a2[i-1],b2[i]);

end

endgenerate

fa fa18(g[21][9],y1[18],y2[18],a2[17],b2[18]);

ha ha29(g[22][8],y1[19],a2[18],b2[19]);

ha ha30(d1[21],y1[20],a2[19],b2[20]);

ha ha31(x3[13],y3[13],a3[12],b3[13]);

ha ha32(x3[14],y3[14],a3[13],b3[14]);

//level 4

//block 1

ha ha33(a1[0],b1[0],c[4],n[0]);

ha ha34(a1[1],b1[1],m[0],n[1]);

ha ha35(a1[2],b1[2],m[1],n[2]);

fa fa19(a1[3],b1[3],y1[4],m[2],n[3]);

fa fa20(a1[4],b1[4],y1[5],m[3],n[4]);

generate

for(i=5;i<=19;i=i+1)

begin

fa foog(a1[i],b1[i],a2[i],m[i-1],n[i]);

end

endgenerate

fa fa21(a1[20],b1[20],y1[21],m[19],n[20]);

fa fa22(a1[21],b1[21],y1[22],m[20],n[21]);

fa fag36(a1[22],b1[22],y1[23],m[21],n[22]);

ha ha37(a1[23],b1[23],m[22],n[23]);

ha ha38(a1[24],b1[24],m[23],n[24]);

ha ha39(a1[25],b1[25],m[24],n[25]);

ha ha40(a1[26],b1[26],m[25],n[26]);

//block 2

ha ha41(d4[12],b2[10],m1[9],n1[10]);

ha ha42(b2[11],x3[12],m1[10],n1[11]);

ha ha43(a3[12],b2[12],m1[11],n1[12]);

fa fa23(b2[13],a3[13],b3[13],m1[12],n1[13]);

fa fa24(y3[15],b2[14],b3[14],m1[13],n1[14]);

//level 5

ha ha44(m[0],n[0],c[5],p[0]);

generate

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

begin

ha hoog(m[i],n[i],o[i-1],p[i]);

end

endgenerate

generate

for(i=5;i<=8;i=i+1)

begin

fa fhoog(m[i],n[i],b2[i+1],o[i-1],p[i]);

end

endgenerate

generate

for(i=9;i<=13;i=i+1)

begin

fa faoog(m[i],n[i],m1[i],o[i-1],p[i]);

end

endgenerate

generate

for(i=14;i<=19;i=i+1)

begin:uss

fa fiog(m[i],n[i],b2[i+1],o[i-1],p[i]);

end

endgenerate

assign m[26]=b1[27];

generate

for(i=20;i<=26;i=i+1)

begin:we

ha hioog(m[i],n[i],o[i-1],p[i]);

end

endgenerate

//level 6

ha ha45(o[0],p[0],c[6],p1[0]);

generate

for(i=1;i<=8;i=i+1)

begin:come

ha goog(o[i],p[i],o1[i-1],p1[i]);

end

endgenerate

generate

for(i=9;i<=13;i=i+1)

begin:good

fa gfg(o[i],p[i],n1[i+1],o1[i-1],p1[i]);

end

endgenerate

generate

for(i=14;i<=25;i=i+1)

begin

ha dog(o[i],p[i],o1[i-1],p1[i]);

end

endgenerate

//assign o1[24]=p[25];

assign o1[27:25]=3'b0;

assign p1[27:26]=2'b0;

ccs28bit q(o1,p1,c1,vq);

assign c[31:7]=c1[24:0];

endmodule

module ccs28bit(

input [27:0] a,

input [27:0] b,

output [27:0] sum,

output cout

);

wire v;

assign v=0;

wire gg0,gp0,gg4,gp4,gg8,gp8,c4,c8,g;

wire gg12,gp12,gg16,gp16,gg20,gp20,gg24,gp24,c12,c16,c20,c24;

cla4t cc0(a[3:0],b[3:0],v,sum[3:0],gg0,gp0);

cla4t cc1(a[7:4],b[7:4],c4,sum[7:4],gg4,gp4);

cla4t cc2(a[11:8],b[11:8],c8,sum[11:8],gg8,gp8);

cla4t cc3(a[15:12],b[15:12],c12,sum[15:12],gg12,gp12);

cla4t cc4(a[19:16],b[19:16],c16,sum[19:16],gg16,gp16);

cla4t cc5(a[23:20],b[23:20],c20,sum[23:20],gg20,gp20);

cla4t cc6(a[27:24],b[27:24],c24,sum[27:24],gg24,gp24);

assign c4=gg0 ;

assign c8=gg4 |(gg0 & gp4);

assign c12=gg8|gp8&gg4|gp8&gp4&gg0;

assign c16=gg12| gp12&gg8 |(gp12&gg4 & gp8) | (gp12&gg0 & gp4 & gp8);

assign c20=gg16 |gp16&gg12| gp16&gp12&gg8 |(gp16&gp12&gg4 & gp8) | (gp16&gp12&gg0 & gp4 & gp8);

assign c24=gg20|gp20&gg16 |gp20&gp16&gg12| gp20&gp16&gp12&gg8 |(gp20&gp16&gp12&gg4 & gp8) | (gp20&gp16&gp12&gg0 & gp4 & gp8);

assign g=gg24|gp24&gg20|gp24&gp20&gg16|gp24&gp20&gp16&gg12|(gp24&gp20&gp16&gp12&gg8) | (gp24&gp20&gp16&gp12&gg4 & gp8 ) | (gp24&gp20&gp16&gp12&gg0 & gp4 & gp8 );

assign cout=g ;

endmodule

module cla4t(a,b,cin,sum,gg,gp);

input [3:0]a,b;

input cin;

output [3:0]sum;

output gg,gp;

wire [3:0] p,g,c;

assign p=a^b;//propagate

assign g=a&b; //generate

assign c[0]=cin;

assign c[1]= g[0]|(p[0]&c[0]);

assign c[2]= g[1] | (p[1]&g[0]) | p[1]&p[0]&c[0];

assign c[3]= g[2] | (p[2]&g[1]) | p[2]&p[1]&g[0] | p[2]&p[1]&p[0]&c[0];

assign gp=&p;

assign gg=g[3] | (p[3]&g[2]) | (p[3]&p[2]&g[1]) | (p[3]&p[2]&p[1]&g[0]);

assign sum=p^c;

endmodule

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

input a,b;

output s,c;

assign s=a^b;

assign c=a&b;

endmodule

module fa(a,b,c,s,ca);

input a,b,c;

output s,ca;

assign s=a^b^c;

assign ca=a&b|b&c|c&a;

endmodule

**Wallace 32 bit**

module wallace32(

input [31:0] a,

input [31:0] b,

output [63:0] c );

wire [60:0]a2,b2,a1,b1,ga,h,g1,h1,p,q,p1,q1,r,t,r1,t1,r2,t2,w4,e4,e3,w3,e2,w2,w1,e1,e,w;

wire [60:0]x6,y6,x5,y5,x4,y4,x3,y3,x2,y2,x1,y1,x,y,s,d,s1,d1,s2,d2,s3,d3,s4,d4,s5,d5,s6,d6,s7,d7,s8,s9,d8,d9,s10,d10;

genvar i,j;

wire [31:0]flag[31:0];

wire [31:0]g[62:0];

generate

for(i=0;i<32;i=i+1)

begin:fi

assign flag[i]=(b[i]==1)?a:32'b0;

end

endgenerate

generate

for(i=0;i<32;i=i+1)

begin:gi

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

begin:gii

assign g[i][j]=flag[i-j][j];

end

end

endgenerate

generate

for(i=32;i<=62;i=i+1)

begin:aq

for(j=0;j<=62-i;j=j+1)

begin:aqq

assign g[i][j]=flag[31-j][i+j-31];

end

end

endgenerate

//level 1

assign c[0]=g[0][0];

//block 1

ha ha1(g[1][0],g[1][1],c[1],d[0]);

generate

for(i=2;i<=60;i=i+1)

begin

fa fai(g[i][0],g[i][1],g[i][2],s[i-2],d[i-1]);

end

endgenerate

ha ha2(g[60][0],g[60][1],s[59],d[60]);

//block 2

ha ha3(g[4][3],g[4][4],s1[2],d1[3]);

generate

for(i=5;i<=57;i=i+1)

begin

fa fak(g[i][3],g[i][4],g[i][5],s1[i-2],d1[i-1]);

end

endgenerate

ha ha4(g[58][3],g[58][4],s1[56],d1[57]);

//block 3

ha ha5(g[7][6],g[7][7],s2[5],d2[6]);

generate

for(i=8;i<=54;i=i+1)

begin

fa faki(g[i][6],g[i][7],g[i][8],s2[i-2],d2[i-1]);

end

endgenerate

ha ha6(g[55][6],g[55][7],s2[53],d2[54]);

//block 4

ha ha7(g[10][9],g[10][10],s3[8],d3[9]);

generate

for(i=11;i<=51;i=i+1)

begin

fa faii(g[i][9],g[i][10],g[i][11],s3[i-2],d3[i-1]);

end

endgenerate

ha ha8(g[52][9],g[52][10],s3[50],d3[51]);

//block 5

ha ha9(g[13][12],g[13][13],s4[11],d4[12]);

generate

for(i=14;i<=48;i=i+1)

begin

fa faij(g[i][12],g[i][13],g[i][14],s4[i-2],d4[i-1]);

end

endgenerate

ha ha10(g[49][12],g[49][13],s4[47],d4[48]);

//block 6

ha ha11(g[16][15],g[16][16],s5[14],d5[15]);

generate

for(i=17;i<=45;i=i+1)

begin

fa fajk(g[i][15],g[i][16],g[i][17],s5[i-2],d5[i-1]);

end

endgenerate

ha ha12(g[46][15],g[47][15],s5[44],d5[45]);

//block 7

ha ha13(g[19][18],g[19][19],s6[17],d6[18]);

generate

for(i=20;i<=42;i=i+1)

begin

fa fakl(g[i][18],g[i][19],g[i][20],s6[i-2],d6[i-1]);

end

endgenerate

ha ha14(g[43][18],g[43][19],s6[41],d6[42]);

//block 8

ha ha15(g[22][21],g[22][22],s7[20],d7[21]);

generate

for(i=23;i<=39;i=i+1)

begin

fa faklo(g[i][21],g[i][22],g[i][23],s7[i-2],d7[i-1]);

end

endgenerate

ha ha16(g[40][21],g[40][22],s7[38],d7[39]);

//block 9

ha ha17(g[25][24],g[25][25],s8[23],d8[24]);

generate

for(i=26;i<=36;i=i+1)

begin

fa fakljjo(g[i][24],g[i][25],g[i][26],s8[i-2],d8[i-1]);

end

endgenerate

ha ha18(g[37][24],g[37][25],s8[35],d8[36]);

//block 10

ha ha19(g[28][27],g[28][28],s9[26],d9[27]);

generate

for(i=29;i<=33;i=i+1)

begin

fa faklhj(g[i][27],g[i][28],g[i][29],s9[i-2],d9[i-1]);

end

endgenerate

ha ha20(g[34][27],g[34][28],s9[32],d9[33]);

//block 11

ha ha21(g[31][30],g[31][31],s10[29],d10[30]);

//level 2

//block 1

ha haq1(s[0],d[0],c[2],y[0]);

fa faq1(s[1],d[1],g[3][3],x[0],y[1]);

generate

for(i=2;i<=56;i=i+1)

begin

fa faq2(s[i],d[i],s1[i],x[i-1],y[i]);

end

endgenerate

fa faq3(s[57],d[57],g[59][3],x[56],y[57]);

ha haq2(s[58],d[58],x[57],y[58]);

ha haq3(s[59],d[59],x[58],y[59]);

ha haq4(g[62][0],d[60],x[59],y[60]);

//block 2

ha haq5(g[6][6],d1[4],x1[3],y1[4]);

ha haq6(d1[5],s2[5],x1[4],y1[5]);

generate

for(i=6;i<=53;i=i+1)

begin

fa faq4(d1[i],s2[i],d2[i],x1[i-1],y1[i]);

end

endgenerate

fa faq5(d1[54],d2[54],g[56][6],x1[53],y1[54]);

//block 3

ha haq7(s3[9],d3[9],x2[8],y2[9]);

fa faq6(g[12][12],s3[10],d3[10],x2[9],y2[10]);

generate

for(i=11;i<=47;i=i+1)

begin

fa faq7(s3[i],d3[i],s4[i],x2[i-1],y2[i]);

end

endgenerate

fa faq8(g[50][12],d3[48],s3[48],x2[47],y2[48]);

ha haq8(s3[49],d3[49],x2[48],y2[49]);

ha haq9(s3[50],d3[50],x2[49],y2[50]);

ha haq10(d3[51],g[53][9],x2[50],y2[51]);

//block 4

ha haq11(g[15][15],d4[13],x3[12],y3[13]);

ha haq12(d4[14],s5[14],x3[13],y3[14]);

generate

for(i=15;i<=44;i=i+1)

begin

fa faq9(d4[i],d5[i],s5[i],x3[i-1],y3[i]);

end

endgenerate

fa faq10(g[47][15],d4[45],d5[45],x3[44],y3[45]);

//block 5

ha haq13(s6[18],d6[18],x4[17],y4[18]);

fa faq11(g[21][21],s6[19],d6[19],x4[18],y4[19]);

generate

for(i=20;i<=38;i=i+1)

begin

fa faq12(d6[i],s6[i],s7[i],x4[i-1],y4[i]);

end

endgenerate

fa faq13(g[41][21],d6[39],s6[39],x4[38],y4[39]);

ha haq14(d6[40],s6[40],x4[39],y4[40]);

ha haq15(d6[41],s6[41],x4[40],y4[41]);

ha haq16(d6[42],g[44][18],x4[41],y4[42]);

//block 6

ha haq17(g[24][24],d7[22],x5[21],y5[22]);

ha haq18(d7[23],s8[23],x5[22],y5[23]);

generate

for(i=24;i<=35;i=i+1)

begin

fa faq14(d7[i],s8[i],d8[i],x5[i-1],y5[i]);

end

endgenerate

fa faq15(g[38][24],d7[36],d8[36],x5[35],y5[36]);

//block 7

ha haq19(s9[27],d9[27],x6[26],y6[27]);

fa faq16(g[30][30],s9[28],d9[28],x6[27],y6[28]);

fa faq17(s10[29],s9[29],d9[29],x6[28],y6[29]);

fa faq18(g[32][30],s9[30],d9[30],x6[29],y6[30]);

ha haq20(s9[31],d9[31],x6[30],y6[31]);

ha haq21(s9[32],d9[32],x6[31],y6[32]);

ha haq22(g[35][27],d9[33],x6[32],y6[33]);

//level 3

//block 1

ha haa1(x[0],y[0],c[3],w[0]);

ha haa2(x[1],y[1],e[0],w[1]);

fa faa1(d1[3],x[2],y[2],e[1],w[2]);

assign x1[56:54]=d1[57:55];

generate

for(i=3;i<=56;i=i+1)

begin

fa faa2(x[i],y[i],x1[i],e[i-1],w[i]);

end

endgenerate

ha haa3(x[57],y[57],e[56],w[57]);

ha haa4(x[58],y[58],e[57],w[58]);

ha haa5(x[59],y[59],e[58],w[59]);

//block 2

ha haa6(g[9][9],y1[6],e1[5],w1[6]);

ha haa7(s3[8],y1[7],e1[6],w1[7]);

ha haa8(y1[8],x2[8],e1[7],w1[8]);

generate

for(i=9;i<=50;i=i+1)

begin

fa faa3(y1[i],x2[i],y2[i],e1[i-1],w1[i]);

end

endgenerate

ha haa9(y1[51],y2[51],e1[50],w1[51]);

//block 3

ha haa10(x3[13],y3[13],e2[12],w2[13]);

ha haa11(x3[14],y3[14],e2[13],w2[14]);

fa faa4(g[18][18],x3[15],y3[15],e2[14],w2[15]);

fa faa5(s6[17],x3[16],y3[16],e2[15],w2[16]);

generate

for(i=17;i<=41;i=i+1)

begin

fa faa6(x3[i],y3[i],x4[i],e2[i-1],w2[i]);

end

endgenerate

assign x3[45]=d4[46];

generate

for(i=42;i<=45;i=i+1)

begin

ha haa6(x3[i],y3[i],e2[i-1],w2[i]);

end

endgenerate

//block 4

ha haaz7(d7[21],y4[20],e3[19],w3[20]);

ha haaz8(y4[21],x5[21],e3[20],w3[21]);

assign x5[36]=d7[37];

generate

for(i=22;i<=36;i=i+1)

begin

fa faa7(y4[i],x5[i],y5[i],e3[i-1],w3[i]);

end

endgenerate

ha haaz9(y4[37],d7[38],e3[36],w3[37]);

ha haaz10(y4[38],d7[39],e3[37],w3[38]);

//block 5

ha haaz11(x6[27],y6[27],e4[26],w4[27]);

ha haa12(x6[28],y6[28],e4[27],w4[28]);

fa faa8(x6[29],y6[29],d10[30],e4[28],w4[29]);

ha haa13(x6[30],y6[30],e4[29],w4[30]);

ha haa14(x6[31],y6[31],e4[30],w4[31]);

ha haa15(x6[32],y6[32],e4[31],w4[32]);

//level 4

//block 1

ha hab1(e[0],w[0],c[4],t[0]);

ha hab2(e[1],w[1],r[0],t[1]);

ha hab3(e[2],w[2],r[1],t[2]);

assign e1[4:3]=y1[5:4];

assign e1[53:51]=y1[54:52];

generate

for(i=3;i<=53;i=i+1)

begin

fa fab1(e[i],w[i],e1[i],r[i-1],t[i]);

end

endgenerate

assign e[59]=y[60];

generate

for(i=54;i<=59;i=i+1)

begin

ha hab4(e[i],w[i],r[i-1],t[i]);

end

endgenerate

//block 2

ha hab5(d4[12],w1[10],r1[9],t1[10]);

ha hab6(x3[12],w1[11],r1[10],t1[11]);

ha hab7(w1[12],e2[12],r1[11],t1[12]);

assign e2[45]=d4[47];

generate

for(i=13;i<=45;i=i+1)

begin

fa fabb1(w1[i],e2[i],w2[i],r1[i-1],t1[i]);

end

endgenerate

ha hab8(d4[48],w1[46],r1[45],t1[46]);

//block 3

ha hac1(e3[20],w3[20],r2[19],t2[20]);

ha hac2(e3[21],w3[21],r2[20],t2[21]);

ha hac3(e3[22],w3[22],r2[21],t2[22]);

assign e4[23]=g[27][27];

assign e4[24]=s9[26];

assign e4[25]=x6[26];

assign e4[32]=y6[33];

generate

for(i=23;i<=32;i=i+1)

begin

fa fac1(e3[i],w3[i],e4[i],r2[i-1],t2[i]);

end

endgenerate

assign e3[38]=y4[39];

generate

for(i=33;i<=38;i=i+1)

begin

ha haac1(e3[i],w3[i],r2[i-1],t2[i]);

end

endgenerate

//level 5

//block 1

ha havv1(r[0],t[0],c[5],q[0]);

generate

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

begin

ha hav1(r[i],t[i],p[i-1],q[i]);

end

endgenerate

assign r1[8:5]=w1[9:6];

assign r1[50:46]=w1[51:47];

generate

for(i=5;i<=50;i=i+1)

begin

fa fav1(r[i],t[i],r1[i],p[i-1],q[i]);

end

endgenerate

generate

for(i=51;i<=58;i=i+1)

begin

ha hav2(r[i],t[i],p[i-1],q[i]);

end

endgenerate

//block 2

ha hav3(y4[18],t1[16],p1[15],q1[16]);

ha hav4(y4[19],t1[17],p1[16],q1[17]);

ha hav5(e3[19],t1[18],p1[17],q1[18]);

ha hav6(t1[19],r2[19],p1[18],q1[19]);

generate

for(i=20;i<=37;i=i+1)

begin

fa faaav2(t1[i],r2[i],t2[i],p1[i-1],q1[i]);

end

endgenerate

fa faqq1(t1[38],t2[38],y4[40],p1[37],q1[38]);

ha haqq1(t1[39],y4[41],p1[38],q1[39]);

ha haqq2(t1[40],y4[42],p1[39],q1[40]);

//level 6

ha hfv1(p[0],q[0],c[6],h[0]);

generate

for(i=1;i<=8;i=i+1)

begin

ha hfv(p[i],q[i],ga[i-1],h[i]);

end

endgenerate

assign p1[14:9]=t1[15:10];

assign p1[45:40]=t1[46:41];

generate

for(i=9;i<=45;i=i+1)

begin

fa hfv2(p[i],q[i],p1[i],ga[i-1],h[i]);

end

endgenerate

generate

for(i=46;i<=57;i=i+1)

begin

ha hfv3(p[i],q[i],ga[i-1],h[i]);

end

endgenerate

generate

for(i=25;i<=30;i=i+1)

begin

ha hfv4(q1[i],w4[i+2],g1[i-1],h1[i]);

end

endgenerate

//level 7

ha hac(ga[0],h[0],c[7],b1[0]);

generate

for(i=1;i<=14;i=i+1)

begin

ha hac1(ga[i],h[i],a1[i-1],b1[i]);

end

endgenerate

assign g1[23:15]=q1[24:16];

assign g1[39:30]=q1[40:31];

generate

for(i=15;i<=39;i=i+1)

begin

fa fac2(ga[i],h[i],g1[i],a1[i-1],b1[i]);

end

endgenerate

generate

for(i=40;i<=56;i=i+1)

begin

ha hac3(ga[i],h[i],a1[i-1],b1[i]);

end

endgenerate

//level 8

ha haca(a1[0],b1[0],c[8],b2[0]);

generate

for(i=1;i<=23;i=i+1)

begin

ha haca1(a1[i],b1[i],a2[i-1],b2[i]);

end

endgenerate

generate

for(i=24;i<=29;i=i+1)

begin

fa faca1(a1[i],b1[i],h1[i+1],a2[i-1],b2[i]);

end

endgenerate

assign a1[56]=h[57];

generate

for(i=30;i<=56;i=i+1)

begin

ha haca2(a1[i],b1[i],a2[i-1],b2[i]);

end

endgenerate

//level 9

ccs za(a2[55:0],b2[55:0],{qa,c[63:9]},cout);

endmodule

module ccs(a,b,sum,cout);

input [55:0]a,b;

output [55:0]sum;

output cout;

assign cout=0;

wire gg0,gp0,gg4,gp4,gg8,gp8,gg12,gp12,gg16,gp16,gg20,gp20,gg24,gp24,gg28,gp28,gg32,gp32,gg36,gp36,gg40,gp40,gg44,gp44,gg48,gp48,gg52,gp52,gg56,gp56,gg60,gp60,gg64,gp64,gg68,gp68,gg72,gp72,gg76,gp76,gg80,gp80,gg84,gp84,gg88,gp88,gg92,gp92,gg96,gp96,gg100,gp100,gg104,gp104,gg108,gp108,gg112,gp112,gg116,gp116,gg120,gp120,gg124,gp124,c4,c8,c12,c16,c20,c24,c28,c32,c36,c40,c44,c48,c52,c56,c60,c64,c68,c72,c76,c80,c84,c88,c92,c96,c100,c104,c108,c112,c116,c120,c124;

wire [3:0]a1,b1,sum1;wire cin;

assign cin=0;

cla4bit cc0(a[3:0],b[3:0],cin,sum[3:0],gg0,gp0);

cla4bit cc1(a[7:4],b[7:4],c4,sum[7:4],gg4,gp4);

cla4bit cc2(a[11:8],b[11:8],c8,sum[11:8],gg8,gp8);

cla4bit cc3(a[15:12],b[15:12],c12,sum[15:12],gg12,gp12);

cla4bit cc4(a[19:16],b[19:16],c16,sum[19:16],gg16,gp16);

cla4bit cc5(a[23:20],b[23:20],c20,sum[23:20],gg20,gp20);

cla4bit cc6(a[27:24],b[27:24],c24,sum[27:24],gg24,gp24);

cla4bit cc7(a[31:28],b[31:28],c28,sum[31:28],gg28,gp28);

cla4bit cc8(a[35:32],b[35:32],c32,sum[35:32],gg32,gp32);

cla4bit cc9(a[39:36],b[39:36],c36,sum[39:36],gg36,gp36);

cla4bit cc10(a[43:40],b[43:40],c40,sum[43:40],gg40,gp40);

cla4bit cc11(a[47:44],b[47:44],c44,sum[47:44],gg44,gp44);

cla4bit cc12(a[51:48],b[51:48],c48,sum[51:48],gg48,gp48);

cla4bit cc13(a[55:52],b[55:52],c52,sum[55:52],gg52,gp52);

//cla4bit cc14(a1,b1,c56,sum1,gg56,gp56);

assign c4=gg0 ;

assign c8=gg4 |(gg0 & gp4) ;

assign c12=gg8 |(gg4 & gp8) | (gg0 & gp4 & gp8);

assign c16=gg12| gp12&gg8 |(gp12&gg4 & gp8) | (gp12&gg0 & gp4 & gp8);

assign c20=gg16 |gp16&gg12| gp16&gp12&gg8 |(gp16&gp12&gg4 & gp8) | (gp16&gp12&gg0 & gp4 & gp8);

assign c24=gg20|gp20&gg16 |gp20&gp16&gg12| gp20&gp16&gp12&gg8 |(gp20&gp16&gp12&gg4 & gp8) | (gp20&gp16&gp12&gg0 & gp4 & gp8);

assign c28=gg24|gp24&gg20|gp24&gp20&gg16 |gp24&gp20&gp16&gg12| gp24&gp20&gp16&gp12&gg8 |(gp24&gp20&gp16&gp12&gg4 & gp8) | (gp24&gp20&gp16&gp12&gg0 & gp4 & gp8);

assign c32=gg28|gp28&gg24|gp28&gp24&gg20|gp28&gp24&gp20&gg16 |gp28&gp24&gp20&gp16&gg12| gp28&gp24&gp20&gp16&gp12&gg8 |(gp28&gp24&gp20&gp16&gp12&gg4 & gp8) | (gp28&gp24&gp20&gp16&gp12&gg0 & gp4 & gp8);

assign c36=gg32|gg28&gp32|gp28&gp32&gg24|gp28&gp32&gp24&gg20|gp28&gp32&gp24&gp20&gg16 |gp28&gp32&gp24&gp20&gp16&gg12| gp28&gp32&gp24&gp20&gp16&gp12&gg8 |(gp28&gp32&gp24&gp20&gp16&gp12&gg4 & gp8) | (gp28&gp32&gp24&gp20&gp16&gp12&gg0 & gp4 & gp8);

assign c40=gg36|gp36&gg32|gg28&gp32&gp36|gp36&gp32&gp28&gg24|gp28&gp36&gp32&gp24&gg20|gp28&gp36&gp32&gp24&gp20&gg16 |gp28&gp36&gp32&gp24&gp20&gp16&gg12| gp28&gp36&gp32&gp24&gp20&gp16&gp12&gg8 |(gp28&gp36&gp32&gp24&gp20&gp16&gp12&gg4 & gp8) | (gp28&gp36&gp32&gp24&gp20&gp16&gp12&gg0 & gp4 & gp8);

assign c44=gg40|gp40&gg36|gp40&gp36&gg32|gg28&gp32&gp36&gp40|gp28&gp40&gp36&gp32&gg24|gp28&gp40&gp36&gp32&gp24&gg20|gp28&gp40&gp36&gp32&gp24&gp20&gg16 |gp28&gp40&gp36&gp32&gp24&gp20&gp16&gg12| gp28&gp40&gp36&gp32&gp24&gp20&gp16&gp12&gg8 |(gp28&gp40&gp36&gp32&gp24&gp20&gp16&gp12&gg4 & gp8) | (gp28&gp40&gp36&gp32&gp24&gp20&gp16&gp12&gg0 & gp4 & gp8);

assign c48=gg44|gp44&gg40|gp44&gp40&gg36|gp44&gp40&gp36&gg32|gg28&gp32&gp36&gp40&gp44|gp28&gp44&gp40&gp36&gp32&gg24|gp28&gp44&gp40&gp36&gp32&gp24&gg20|gp28&gp44&gp40&gp36&gp32&gp24&gp20&gg16 |gp28&gp44&gp40&gp36&gp32&gp24&gp20&gp16&gg12| gp28&gp44&gp40&gp36&gp32&gp24&gp20&gp16&gp12&gg8 |(gp28&gp44&gp40&gp36&gp32&gp24&gp20&gp16&gp12&gg4 & gp8) | (gp28&gp44&gp40&gp36&gp32&gp24&gp20&gp16&gp12&gg0 & gp4 & gp8);

assign c52=gg48|gp48&gg44|gp48&gp44&gg40|gp48&gp44&gp40&gg36|gp48&gp44&gp40&gp36&gg32|gg28&gp32&gp36&gp40&gp44&gp48|gp28&gp48&gp44&gp40&gp36&gp32&gg24|gp28&gp48&gp44&gp40&gp36&gp32&gp24&gg20|gp28&gp48&gp44&gp40&gp36&gp32&gp24&gp20&gg16 |gp28&gp48&gp44&gp40&gp36&gp32&gp24&gp20&gp16&gg12| gp28&gp48&gp44&gp40&gp36&gp32&gp24&gp20&gp16&gp12&gg8 |(gp28&gp48&gp44&gp40&gp36&gp32&gp24&gp20&gp16&gp12&gg4 & gp8) | (gp28&gp48&gp44&gp40&gp36&gp32&gp24&gp20&gp16&gp12&gg0 & gp4 & gp8);

assign c56=gg52|gp52&gg48|gp52&gp48&gg44|gp52&gp48&gp44&gg40|gp52&gp48&gp44&gp40&gg36|gp52&gp48&gp44&gp40&gp36&gg32|gg28&gp32&gp36&gp40&gp44&gp48&gp52|gp28&gp52&gp48&gp44&gp40&gp36&gp32&gg24|gp28&gp52&gp48&gp44&gp40&gp36&gp32&gp24&gg20|gp28&gp52&gp48&gp44&gp40&gp36&gp32&gp24&gp20&gg16 |gp28&gp52&gp48&gp44&gp40&gp36&gp32&gp24&gp20&gp16&gg12| gp28&gp52&gp48&gp44&gp40&gp36&gp32&gp24&gp20&gp16&gp12&gg8 |(gp28&gp52&gp48&gp44&gp40&gp36&gp32&gp24&gp20&gp16&gp12&gg4 & gp8) | (gp28&gp52&gp48&gp44&gp40&gp36&gp32&gp24&gp20&gp16&gp12&gg0 & gp4 & gp8);

endmodule

module cla4bit(a,b,cin,sum,gg,gp);

input [3:0]a,b;

input cin;

output [3:0]sum;

output gg,gp;

wire [3:0] p,g,c;

assign p=a^b;//propagate

assign g=a&b; //generate

assign c[0]=cin;

assign c[1]= g[0]|(p[0]&c[0]);

assign c[2]= g[1] | (p[1]&g[0]) | p[1]&p[0]&c[0];

assign c[3]= g[2] | (p[2]&g[1]) | p[2]&p[1]&g[0] | p[2]&p[1]&p[0]&c[0];

assign gp=&p;

assign gg=g[3] | (p[3]&g[2]) | (p[3]&p[2]&g[1]) | (p[3]&p[2]&p[1]&g[0]);

assign sum=p^c;

endmodule

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

input a,b;

output s,c;

assign s=a^b;

assign c=a&b;

endmodule

module fa(a,b,c,s,ca);

input a,b,c;

output s,ca;

assign s=a^b^c;

assign ca=a&b|b&c|c&a;

endmodule

**Wallace 64 bit**