陈彦帆 2018K8009918002

18.

对于[-x]补，booth编码逻辑输出pi = ~xi，并且进位c为1。华莱士树将pi的每一位拆开，与其他数相加成两个数，最后通过一个加法器把两个数加起来。对于[-x]补，booth编码逻辑输出pi = ~x(i-1)，并且进位c为1。

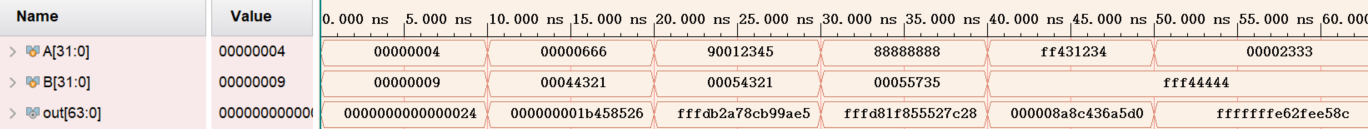
20.

不能。因为浮点数的精度是有限的但π的精度是无限的。

19.

华莱士树设计：第一层5个，第二层4个（多一个用于接收进位），第三层和第四层2个，第五层和第六层1个。

32位booth两位乘法器需要64个华莱士树，16个booth逻辑，1个64位（快速进位）加法器

 选取6个仿真测试点，如下图，经验算全部通过。

源代码：

`timescale 1ns / 1ps

module Multiplier\_32(

input [31:0] A,

input [31:0] B,

output [63:0] product

);

wire [63:0] p0;

wire [63:0] p1;

wire [63:0] p2;

wire [63:0] p3;

wire [63:0] p4;

wire [63:0] p5;

wire [63:0] p6;

wire [63:0] p7;

wire [63:0] p8;

wire [63:0] p9;

wire [63:0] p10;

wire [63:0] p11;

wire [63:0] p12;

wire [63:0] p13;

wire [63:0] p14;

wire [63:0] p15;

wire c0,c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13,c14,c15;

booth b0(

.y2(B[1]),

.y1(B[0]),

.y0(0),

.X({{32{A[31]}},A[31:0]}),

.p(p0),

.c(c0)

);

//省略实例化booth b1-booth b14

booth b15(

.y2(B[31]),

.y1(B[30]),

.y0(B[29]),

.X({{2{A[31]}},A[31:0],30'b0}),

.p(p15),

.c(c15)

);

wire cout0\_0,cout0\_1,cout0\_2,cout0\_3,cout0\_4,cout0\_5,cout0\_6,cout0\_7,cout0\_8,cout0\_9,cout0\_10,cout0\_11,cout0\_12,cout0\_13,cout1\_0,cout1\_1,cout1\_2,cout1\_3,cout1\_4,cout1\_5,cout1\_6,cout1\_7,cout1\_8,...... cout63\_13;

wire C0,C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12,C13,C14,C15,C16,C17,C18,C19,C20,C21,C22,C23,C24,C25,C26,C27,C28,C29,C30,C31,C32,C33,C34,C35,C36,C37,C38,C39,C40,C41,C42,C43,C44,C45,C46,C47,C48,C49,C50,C51,C52,C53,C54,C55,C56,C57,C58,C59,C60,C61,C62,C63;

wire S0,S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,S11,S12,S13,S14,S15,S16,S17,S18,S19,S20,S21,S22,S23,S24,S25,S26,S27,S28,S29,S30,S31,S32,S33,S34,S35,S36,S37,S38,S39,S40,S41,S42,S43,S44,S45,S46,S47,S48,S49,S50,S51,S52,S53,S54,S55,S56,S57,S58,S59,S60,S61,S62,S63;

wire [63:0] in\_A;

wire [63:0] in\_B;

assign in\_A = {S63,S62,S61,S60,S59,S58,S57,S56,S55,S54,S53,S52,S51,S50,S49,S48,S47,S46,S45,S44,S43,S42,S41,S40,S39,S38,S37,S36,S35,S34,S33,S32,S31,S30,S29,S28,S27,S26,S25,S24,S23,S22,S21,S20,S19,S18,S17,S16,S15,S14,S13,S12,S11,S10,S9,S8,S7,S6,S5,S4,S3,S2,S1,S0};

assign in\_B = {C62,C61,C60,C59,C58,C57,C56,C55,C54,C53,C52,C51,C50,C49,C48,C47,C46,C45,C44,C43,C42,C41,C40,C39,C38,C37,C36,C35,C34,C33,C32,C31,C30,C29,C28,C27,C26,C25,C24,C23,C22,C21,C20,C19,C18,C17,C16,C15,C14,C13,C12,C11,C10,C9,C8,C7,C6,C5,C4,C3,C2,C1,C0,c15};

adder64 add64(

.cin(c15),

.A(in\_A),

.B(in\_B),

.sum(product),

.cout()

);

//assign product = in\_A + in\_B;

Wallace w0(

.n0(p0[0]),

.n1(p1[0]),

.n2(p2[0]),

.n3(p3[0]),

.n4(p4[0]),

.n5(p5[0]),

.n6(p6[0]),

.n7(p7[0]),

.n8(p8[0]),

.n9(p9[0]),

.n10(p10[0]),

.n11(p11[0]),

.n12(p12[0]),

.n13(p13[0]),

.n14(p14[0]),

.n15(p15[0]),

.cin0(c0),

.cin1(c1),

.cin2(c2),

.cin3(c3),

.cin4(c4),

.cin5(c5),

.cin6(c6),

.cin7(c7),

.cin8(c8),

.cin9(c9),

.cin10(c10),

.cin11(c11),

.cin12(c12),

.cin13(c13),

.c11(cout0\_0),

.c12(cout0\_1),

.c13(cout0\_2),

.c14(cout0\_3),

.c15(cout0\_4),

.c21(cout0\_5),

.c22(cout0\_6),

.c23(cout0\_7),

.c24(cout0\_8),

.c31(cout0\_9),

.c32(cout0\_10),

.c41(cout0\_11),

.c42(cout0\_12),

.c51(cout0\_13),

.c61(C0),

.s61(S0)

);

//此处省略实例化 Wallace w1-Wallace w63

endmodule

module booth(

input y2,

input y1,

input y0,

input [63:0] X,

output [63:0] p,

output c

);

wire addx,add2x,subx,sub2x;

assign addx = ~y2&y1&~y0|~y2&~y1&y0;

assign add2x = ~y2&y1&y0;

assign subx = y2&y1&~y0|y2&~y1&y0;

assign sub2x = y2&~y1&~y0;

assign c = subx | sub2x;

assign p[0] = subx&~X[0] | addx&X[0] | sub2x;

genvar nbit;

generate

for(nbit = 1; nbit<64; nbit = nbit+1)

begin: kkk

assign p[nbit] = subx&~X[nbit]|sub2x&~X[nbit-1]|addx&X[nbit]|add2x&X[nbit-1];

end

endgenerate

endmodule

module Full\_adder(

input A,

input B,

input cin,

output sum,

output cout

);

assign sum = ~A & ~B & cin | ~A & B & ~cin | A & ~B & ~cin | A & B & cin;

assign cout = A & B | A & cin | B & cin;

endmodule

module Wallace(

input n0,

input n1,

input n2,

input n3,

input n4,

input n5,

input n6,

input n7,

input n8,

input n9,

input n10,

input n11,

input n12,

input n13,

input n14,

input n15,

input cin0,

input cin1,

input cin2,

input cin3,

input cin4,

input cin5,

input cin6,

input cin7,

input cin8,

input cin9,

input cin10,

input cin11,

input cin12,

input cin13,

output c11,

output c12,

output c13,

output c14,

output c15,

output c21,

output c22,

output c23,

output c24,

output c31,

output c32,

output c41,

output c42,

output c51,

output c61,

output s61

);

wire s11,s12,s13,s14,s15;

wire s21,s22,s23,s24;

wire s31,s32;

wire s41,s42;

wire s51;

Full\_adder a11(

.A(n0),

.B(n1),

.cin(n2),

.sum(s11),

.cout(c11)

);

Full\_adder a12(

.A(n3),

.B(n4),

.cin(n5),

.sum(s12),

.cout(c12)

);

Full\_adder a13(

.A(n6),

.B(n7),

.cin(n8),

.sum(s13),

.cout(c13)

);

Full\_adder a14(

.A(n9),

.B(n10),

.cin(n11),

.sum(s14),

.cout(c14)

);

Full\_adder a15(

.A(n12),

.B(n13),

.cin(n14),

.sum(s15),

.cout(c15)

);

Full\_adder a21(

.A(s11),

.B(s12),

.cin(s13),

.sum(s21),

.cout(c21)

);

Full\_adder a22(

.A(s14),

.B(s15),

.cin(n15),

.sum(s22),

.cout(c22)

);

Full\_adder a23(

.A(cin0),

.B(cin1),

.cin(cin2),

.sum(s23),

.cout(c23)

);

Full\_adder a24(

.A(cin3),

.B(cin4),

.cin(0),

.sum(s24),

.cout(c24)

);

Full\_adder a31(

.A(s21),

.B(s22),

.cin(s23),

.sum(s31),

.cout(c31)

);

Full\_adder a32(

.A(s24),

.B(cin5),

.cin(cin6),

.sum(s32),

.cout(c32)

);

Full\_adder a41(

.A(s31),

.B(s32),

.cin(cin7),

.sum(s41),

.cout(c41)

);

Full\_adder a42(

.A(cin8),

.B(cin9),

.cin(cin10),

.sum(s42),

.cout(c42)

);

Full\_adder a51(

.A(s41),

.B(s42),

.cin(cin11),

.sum(s51),

.cout(c51)

);

Full\_adder a61(

.A(s51),

.B(cin12),

.cin(cin13),

.sum(s61),

.cout(c61)

);

endmodule

module adder4(

input c0,

input p0,

input p1,

input p2,

input p3,

input g0,

input g1,

input g2,

input g3,

output c1,

output c2,

output c3,

output P,

output G

);

assign c1 = g0 | p0&c0;

assign c2 = g1 | p1&g0 | p1&p0&c0;

assign c3 = g2 | p2&g1 | p2&p1&g0 | p2&p1&p0&c0;

assign P = p3&p2&p1&p0;

assign G = g3 | p3&g2 | p3&p2&g1 | p3&p2&p1&g0;

endmodule

module adder64(

input cin,

input [63:0] A,

input [63:0] B,

output [63:0] sum,

output cout

);

// 64位并行快速进位加法器。第一层有16个adder4，第二城有4个，第三层有1个。

wire p0\_0,p0\_1,p0\_2,p0\_3,p0\_4,p0\_5,p0\_6,p0\_7,p0\_8,p0\_9,p0\_10,p0\_11,p0\_12,p0\_13,p0\_14,p0\_15,p0\_16,p0\_17,p0\_18,p0\_19,p0\_20,p0\_21,p0\_22,p0\_23,p0\_24,p0\_25,p0\_26,p0\_27,p0\_28,p0\_29,p0\_30,p0\_31,p0\_32,p0\_33,p0\_34,p0\_35,p0\_36,p0\_37,p0\_38,p0\_39,p0\_40,p0\_41,p0\_42,p0\_43,p0\_44,p0\_45,p0\_46,p0\_47,p0\_48,p0\_49,p0\_50,p0\_51,p0\_52,p0\_53,p0\_54,p0\_55,p0\_56,p0\_57,p0\_58,p0\_59,p0\_60,p0\_61,p0\_62,p0\_63;

wire g0\_0,g0\_1,g0\_2,g0\_3,g0\_4,g0\_5,g0\_6,g0\_7,g0\_8,g0\_9,g0\_10,g0\_11,g0\_12,g0\_13,g0\_14,g0\_15,g0\_16,g0\_17,g0\_18,g0\_19,g0\_20,g0\_21,g0\_22,g0\_23,g0\_24,g0\_25,g0\_26,g0\_27,g0\_28,g0\_29,g0\_30,g0\_31,g0\_32,g0\_33,g0\_34,g0\_35,g0\_36,g0\_37,g0\_38,g0\_39,g0\_40,g0\_41,g0\_42,g0\_43,g0\_44,g0\_45,g0\_46,g0\_47,g0\_48,g0\_49,g0\_50,g0\_51,g0\_52,g0\_53,g0\_54,g0\_55,g0\_56,g0\_57,g0\_58,g0\_59,g0\_60,g0\_61,g0\_62,g0\_63;

wire c1\_1,c1\_2,c1\_3,c1\_5,c1\_6,c1\_7,c1\_9,c1\_10,c1\_11,c1\_13,c1\_14,c1\_15,c1\_17,c1\_18,c1\_19,c1\_21,c1\_22,c1\_23,c1\_25,c1\_26,c1\_27,c1\_29,c1\_30,c1\_31,c1\_33,c1\_34,c1\_35,c1\_37,c1\_38,c1\_39,c1\_41,c1\_42,c1\_43,c1\_45,c1\_46,c1\_47,c1\_49,c1\_50,c1\_51,c1\_53,c1\_54,c1\_55,c1\_57,c1\_58,c1\_59,c1\_61,c1\_62,c1\_63;

wire p1\_0,p1\_1,p1\_2,p1\_3,p1\_5,p1\_6,p1\_7,p1\_9,p1\_10,p1\_11,p1\_13,p1\_14,p1\_15;

wire g1\_0,g1\_1,g1\_2,g1\_3,g1\_5,g1\_6,g1\_7,g1\_9,g1\_10,g1\_11,g1\_13,g1\_14,g1\_15;

wire c2\_1,c2\_2,c2\_3,c2\_5,c2\_6,c2\_7,c2\_9,c2\_10,c2\_11,c2\_13,c2\_14,c2\_15;

wire p2\_0,p2\_1,p2\_2,p2\_3;

wire g2\_0,g2\_1,g2\_2,g2\_3;

wire c3\_1,c3\_2,c3\_3;

wire p3\_0,g3\_0;

assign cout = p3\_0 & cin | g3\_0;

assign p0\_0 = A[0] | B[0];

//...

assign g0\_63 = A[63] & B[63];

adder4 add1\_0(

.c0(cin),

.p0(p0\_0),

.p1(p0\_1),

.p2(p0\_2),

.p3(p0\_3),

.g0(g0\_0),

.g1(g0\_1),

.g2(g0\_2),

.g3(g0\_3),

.c1(c1\_1),

.c2(c1\_2),

.c3(c1\_3),

.P(p1\_0),

.G(g1\_0)

);

adder4 add1\_1(

.c0(c2\_1),

.p0(p0\_4),

.p1(p0\_5),

.p2(p0\_6),

.p3(p0\_7),

.g0(g0\_4),

.g1(g0\_5),

.g2(g0\_6),

.g3(g0\_7),

.c1(c1\_5),

.c2(c1\_6),

.c3(c1\_7),

.P(p1\_1),

.G(g1\_1)

);

adder4 add1\_2(

.c0(c2\_2),

.p0(p0\_8),

.p1(p0\_9),

.p2(p0\_10),

.p3(p0\_11),

.g0(g0\_8),

.g1(g0\_9),

.g2(g0\_10),

.g3(g0\_11),

.c1(c1\_9),

.c2(c1\_10),

.c3(c1\_11),

.P(p1\_2),

.G(g1\_2)

);

adder4 add1\_3(

.c0(c2\_3),

.p0(p0\_12),

.p1(p0\_13),

.p2(p0\_14),

.p3(p0\_15),

.g0(g0\_12),

.g1(g0\_13),

.g2(g0\_14),

.g3(g0\_15),

.c1(c1\_13),

.c2(c1\_14),

.c3(c1\_15),

.P(p1\_3),

.G(g1\_3)

);

adder4 add1\_4(

.c0(c3\_1),

.p0(p0\_16),

.p1(p0\_17),

.p2(p0\_18),

.p3(p0\_19),

.g0(g0\_16),

.g1(g0\_17),

.g2(g0\_18),

.g3(g0\_19),

.c1(c1\_17),

.c2(c1\_18),

.c3(c1\_19),

.P(p1\_4),

.G(g1\_4)

);

adder4 add1\_5(

.c0(c2\_5),

.p0(p0\_20),

.p1(p0\_21),

.p2(p0\_22),

.p3(p0\_23),

.g0(g0\_20),

.g1(g0\_21),

.g2(g0\_22),

.g3(g0\_23),

.c1(c1\_21),

.c2(c1\_22),

.c3(c1\_23),

.P(p1\_5),

.G(g1\_5)

);

adder4 add1\_6(

.c0(c2\_6),

.p0(p0\_24),

.p1(p0\_25),

.p2(p0\_26),

.p3(p0\_27),

.g0(g0\_24),

.g1(g0\_25),

.g2(g0\_26),

.g3(g0\_27),

.c1(c1\_25),

.c2(c1\_26),

.c3(c1\_27),

.P(p1\_6),

.G(g1\_6)

);

adder4 add1\_7(

.c0(c2\_7),

.p0(p0\_28),

.p1(p0\_29),

.p2(p0\_30),

.p3(p0\_31),

.g0(g0\_28),

.g1(g0\_29),

.g2(g0\_30),

.g3(g0\_31),

.c1(c1\_29),

.c2(c1\_30),

.c3(c1\_31),

.P(p1\_7),

.G(g1\_7)

);

adder4 add1\_8(

.c0(c3\_2),

.p0(p0\_32),

.p1(p0\_33),

.p2(p0\_34),

.p3(p0\_35),

.g0(g0\_32),

.g1(g0\_33),

.g2(g0\_34),

.g3(g0\_35),

.c1(c1\_33),

.c2(c1\_34),

.c3(c1\_35),

.P(p1\_8),

.G(g1\_8)

);

adder4 add1\_9(

.c0(c2\_9),

.p0(p0\_36),

.p1(p0\_37),

.p2(p0\_38),

.p3(p0\_39),

.g0(g0\_36),

.g1(g0\_37),

.g2(g0\_38),

.g3(g0\_39),

.c1(c1\_37),

.c2(c1\_38),

.c3(c1\_39),

.P(p1\_9),

.G(g1\_9)

);

adder4 add1\_10(

.c0(c2\_10),

.p0(p0\_40),

.p1(p0\_41),

.p2(p0\_42),

.p3(p0\_43),

.g0(g0\_40),

.g1(g0\_41),

.g2(g0\_42),

.g3(g0\_43),

.c1(c1\_41),

.c2(c1\_42),

.c3(c1\_43),

.P(p1\_10),

.G(g1\_10)

);

adder4 add1\_11(

.c0(c2\_11),

.p0(p0\_44),

.p1(p0\_45),

.p2(p0\_46),

.p3(p0\_47),

.g0(g0\_44),

.g1(g0\_45),

.g2(g0\_46),

.g3(g0\_47),

.c1(c1\_45),

.c2(c1\_46),

.c3(c1\_47),

.P(p1\_11),

.G(g1\_11)

);

adder4 add1\_12(

.c0(c3\_3),

.p0(p0\_48),

.p1(p0\_49),

.p2(p0\_50),

.p3(p0\_51),

.g0(g0\_48),

.g1(g0\_49),

.g2(g0\_50),

.g3(g0\_51),

.c1(c1\_49),

.c2(c1\_50),

.c3(c1\_51),

.P(p1\_12),

.G(g1\_12)

);

adder4 add1\_13(

.c0(c2\_13),

.p0(p0\_52),

.p1(p0\_53),

.p2(p0\_54),

.p3(p0\_55),

.g0(g0\_52),

.g1(g0\_53),

.g2(g0\_54),

.g3(g0\_55),

.c1(c1\_53),

.c2(c1\_54),

.c3(c1\_55),

.P(p1\_13),

.G(g1\_13)

);

adder4 add1\_14(

.c0(c2\_14),

.p0(p0\_56),

.p1(p0\_57),

.p2(p0\_58),

.p3(p0\_59),

.g0(g0\_56),

.g1(g0\_57),

.g2(g0\_58),

.g3(g0\_59),

.c1(c1\_57),

.c2(c1\_58),

.c3(c1\_59),

.P(p1\_14),

.G(g1\_14)

);

adder4 add1\_15(

.c0(c2\_15),

.p0(p0\_60),

.p1(p0\_61),

.p2(p0\_62),

.p3(p0\_63),

.g0(g0\_60),

.g1(g0\_61),

.g2(g0\_62),

.g3(g0\_63),

.c1(c1\_61),

.c2(c1\_62),

.c3(c1\_63),

.P(p1\_15),

.G(g1\_15)

);

adder4 add2\_0(

.c0(cin),

.p0(p1\_0),

.p1(p1\_1),

.p2(p1\_2),

.p3(p1\_3),

.g0(g1\_0),

.g1(g1\_1),

.g2(g1\_2),

.g3(g1\_3),

.c1(c2\_1),

.c2(c2\_2),

.c3(c2\_3),

.P(p2\_0),

.G(g2\_0)

);

adder4 add2\_1(

.c0(c3\_1),

.p0(p1\_4),

.p1(p1\_5),

.p2(p1\_6),

.p3(p1\_7),

.g0(g1\_4),

.g1(g1\_5),

.g2(g1\_6),

.g3(g1\_7),

.c1(c2\_5),

.c2(c2\_6),

.c3(c2\_7),

.P(p2\_1),

.G(g2\_1)

);

adder4 add2\_2(

.c0(c3\_2),

.p0(p1\_8),

.p1(p1\_9),

.p2(p1\_10),

.p3(p1\_11),

.g0(g1\_8),

.g1(g1\_9),

.g2(g1\_10),

.g3(g1\_11),

.c1(c2\_9),

.c2(c2\_10),

.c3(c2\_11),

.P(p2\_2),

.G(g2\_2)

);

adder4 add2\_3(

.c0(c3\_3),

.p0(p1\_12),

.p1(p1\_13),

.p2(p1\_14),

.p3(p1\_15),

.g0(g1\_12),

.g1(g1\_13),

.g2(g1\_14),

.g3(g1\_15),

.c1(c2\_13),

.c2(c2\_14),

.c3(c2\_15),

.P(p2\_3),

.G(g2\_3)

);

adder4 add3\_0(

.c0(cin),

.p0(p2\_0),

.p1(p2\_1),

.p2(p2\_2),

.p3(p2\_3),

.g0(g2\_0),

.g1(g2\_1),

.g2(g2\_2),

.g3(g2\_3),

.c1(c3\_1),

.c2(c3\_2),

.c3(c3\_3),

.P(p3\_0),

.G(g3\_0)

);

Full\_adder fa0(

.A(A[0]),

.B(B[0]),

.cin(cin),

.sum(sum[0]),

.cout()

);

//...

Full\_adder fa63(

.A(A[63]),

.B(B[63]),

.cin(c1\_63),

.sum(sum[63]),

.cout()

);

endmodule