**#Bugh\_wooley M**

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 05/07/2024 03:36:26 PM

// Design Name:

// Module Name: bm

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

// First defining the full adder component that will be used in the circuit

module full\_adder(a, b, cin, s, cout);

input a, b, cin;

output s, cout;

assign s = a ^ b ^ cin;

assign cout = (a&b) | (b&cin) | (cin&a);

endmodule

// Defining black block as shown in the circuit diagram

module black\_box(sin, cin, a, b, sout, cout);

input a, b, cin, sin;

output cout, sout;

full\_adder f(a&b, sin, cin, sout, cout);

endmodule

// Defining grey block as shown in the circuit diagram

module grey\_box(sin, cin, a, b, sout, cout);

input a, b, cin, sin;

output cout, sout;

full\_adder f(!(a&b), sin, cin, sout, cout);

endmodule

module baugh\_wooley(x, y, s);

input [7:0] x, y;

output [15:0] s;

// defining constant logic values

supply1 one;

supply0 zero;

// internal results which are required by next stage adders

wire c00, c01, c02, c03, c04, c05, c06, c07, c10, c11, c12, c13, c14, c15, c16, c17, c20, c21, c22, c23, c24, c25, c26, c27, c30, c31, c32, c33, c34, c35, c36, c37, c40, c41, c42, c43, c44, c45, c46, c47, c50, c51, c52, c53, c54, c55, c56, c57, c60, c61, c62, c63, c64, c65, c66, c67, c70, c71, c72, c73, c74, c75, c76, c77;

wire s00, s01, s02, s03, s04, s05, s06, s07, s10, s11, s12, s13, s14, s15, s16, s17, s20, s21, s22, s23, s24, s25, s26, s27, s30, s31, s32, s33, s34, s35, s36, s37, s40, s41, s42, s43, s44, s45, s46, s47, s50, s51, s52, s53, s54, s55, s56, s57, s60, s61, s62, s63, s64, s65, s66, s67, s70, s71, s72, s73, s74, s75, s76, s77;

wire s1, s2, s3, s4, s5, s6, s7, s8;

wire c1, c2, c3, c4, c5, c6, c7, c8;

// Naming the block in the diagram corresponsing to xi and yj inputs as bbij

// s0

black\_box bb00(zero, zero, x[0], y[0], s00, c00);

assign s[0] = s00;

// s1

black\_box bb10(zero, zero, x[1], y[0], s10, c10);

black\_box bb01(s10, c00, x[0], y[1], s01, c01);

assign s[1] = s01;

//s2

black\_box bb20(zero, zero, x[2], y[0], s20, c20);

black\_box bb11(s20, c10, x[1], y[1], s11, c11);

black\_box bb02(s11, c01, x[0], y[2], s02, c02);

assign s[2] = s02;

//s3

black\_box bb30(zero, zero, x[3], y[0], s30, c30);

black\_box bb21(s30, c20, x[2], y[1], s21, c21);

black\_box bb12(s21, c11, x[1], y[2], s12, c12);

black\_box bb03(s12, c02, x[0], y[3], s03, c03);

assign s[3] = s03;

//s4

black\_box bb40(zero, zero, x[4], y[0], s40, c40);

black\_box bb31(s40, c30, x[3], y[1], s31, c31);

black\_box bb22(s31, c21, x[2], y[2], s22, c22);

black\_box bb13(s22, c12, x[1], y[3], s13, c13);

black\_box bb04(s13, c03, x[0], y[4], s04, c04);

assign s[4] = s04;

//s5

black\_box bb50(zero, zero, x[5], y[0], s50, c50);

black\_box bb41(s50, c40, x[4], y[1], s41, c41);

black\_box bb32(s41, c31, x[3], y[2], s32, c32);

black\_box bb23(s32, c22, x[2], y[3], s23, c23);

black\_box bb14(s23, c13, x[1], y[4], s14, c14);

black\_box bb05(s14, c04, x[0], y[5], s05, c05);

assign s[5] = s05;

//s6

black\_box bb60(zero, zero, x[6], y[0], s60, c60);

black\_box bb51(s60, c50, x[5], y[1], s51, c51);

black\_box bb42(s51, c41, x[4], y[2], s42, c42);

black\_box bb33(s42, c32, x[3], y[3], s33, c33);

black\_box bb24(s33, c23, x[2], y[4], s24, c24);

black\_box bb15(s24, c14, x[1], y[5], s15, c15);

black\_box bb06(s15, c05, x[0], y[6], s06, c06);

assign s[6] = s06;

//s7

grey\_box bb70(zero, zero, x[7], y[0], s70, c70);

black\_box bb61(s70, c60, x[6], y[1], s61, c61);

black\_box bb52(s61, c51, x[5], y[2], s52, c52);

black\_box bb43(s52, c42, x[4], y[3], s43, c43);

black\_box bb34(s43, c33, x[3], y[4], s34, c34);

black\_box bb25(s34, c24, x[2], y[5], s25, c25);

black\_box bb16(s25, c15, x[1], y[6], s16, c16);

grey\_box bb07(s16, c06, x[0], y[7], s07, c07);

assign s[7] = s07;

//s8

grey\_box bb71(zero, c70, x[7], y[1], s71, c71);

black\_box bb62(s71, c61, x[6], y[2], s62, c62);

black\_box bb53(s62, c52, x[5], y[3], s53, c53);

black\_box bb44(s53, c43, x[4], y[4], s44, c44);

black\_box bb35(s44, c34, x[3], y[5], s35, c35);

black\_box bb26(s35, c25, x[2], y[6], s26, c26);

grey\_box bb17(s26, c16, x[1], y[7], s17, c17);

full\_adder fa1(s17, c07, one, s1, c1);

assign s[8] = s1;

//s9

grey\_box bb72(zero, c71, x[7], y[2], s72, c72);

black\_box bb63(s72, c62, x[6], y[3], s63, c63);

black\_box bb54(s63, c53, x[5], y[4], s54, c54);

black\_box bb45(s54, c44, x[4], y[5], s45, c45);

black\_box bb36(s45, c35, x[3], y[6], s36, c36);

grey\_box bb27(s36, c26, x[2], y[7], s27, c27);

full\_adder fa2(s27, c17, c1, s2, c2);

assign s[9] = s2;

//s10

grey\_box bb73(zero, c72, x[7], y[3], s73, c73);

black\_box bb64(s73, c63, x[6], y[4], s64, c64);

black\_box bb55(s64, c54, x[5], y[5], s55, c55);

black\_box bb46(s55, c45, x[4], y[6], s46, c46);

grey\_box bb37(s46, c36, x[3], y[7], s37, c37);

full\_adder fa3(s37, c27, c2, s3, c3);

assign s[10] = s3;

//s11

grey\_box bb74(zero, c73, x[7], y[4], s74, c74);

black\_box bb65(s74, c64, x[6], y[5], s65, c65);

black\_box bb56(s65, c55, x[5], y[6], s56, c56);

grey\_box bb47(s56, c46, x[4], y[7], s47, c47);

full\_adder fa4(s47, c37, c3, s4, c4);

assign s[11] = s4;

//s12

grey\_box bb75(zero, c74, x[7], y[5], s75, c75);

black\_box bb66(s75, c65, x[6], y[6], s66, c66);

grey\_box bb57(s66, c56, x[5], y[7], s57, c57);

full\_adder fa5(s57, c47, c4, s5, c5);

assign s[12] = s5;

//s13

grey\_box bb76(zero, c75, x[7], y[6], s76, c76);

grey\_box bb67(s76, c66, x[6], y[7], s67, c67);

full\_adder fa6(s67, c57, c5, s6, c6);

assign s[13] = s6;

//s14

black\_box bb77(zero, c76, x[7], y[7], s77, c77);

full\_adder fa7(s77, c67, c6, s7, c7);

assign s[14] = s7;

//s15

full\_adder fa8(one, c77, c7, s8, c8);

assign s[15] = s8;

endmodule

#**FIR**:

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 05/07/2024 03:47:20 PM

// Design Name:

// Module Name: fir8bit

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

// FIR Filter Module

module FIR (

input clk,

input rst,

input [7:0] x,

input [7:0] h0, h1, h2, h3, h4, h5, h6,

output [15:0] y

);

// Registers and Wires

reg [7:0] x\_values [6:0];

wire [15:0] baugh\_product [6:0];

//wire [7:0] t [6:1];

//wire a1, a2, a3, a4, a5, a6;

// Instantiate Multipliers

//always @\*

// $display("%b,%b,%b,%b,%b,%b,%b",x\_values[0],x\_values[1],x\_values[2],x\_values[3],x\_values[4],x\_values[5],x\_values[6]);

baugh\_wooley mul0 (h0, x\_values[0], baugh\_product[0]);

baugh\_wooley mul1 (h1, x\_values[1], baugh\_product[1]);

baugh\_wooley mul2 (h2, x\_values[2], baugh\_product[2]);

baugh\_wooley mul3 (h3, x\_values[3], baugh\_product[3]);

baugh\_wooley mul4 (h4, x\_values[4], baugh\_product[4]);

baugh\_wooley mul5 (h5, x\_values[5], baugh\_product[5]);

baugh\_wooley mul6 (h6, x\_values[6], baugh\_product[6]);

// RCA (Recursive Carry Adder)

//rca r1 (baugh\_product[0], baugh\_product[1], 0, t[1], a1);

//rca r2 (baugh\_product[2], t[1], 0, t[2], a2);

//rca r3 (baugh\_product[3], t[2], 0, t[3], a3);

//rca r4 (baugh\_product[4], t[3], 0, t[4], a4);

//rca r5 (baugh\_product[5], t[4], 0, t[4], a5);

//rca r6 (baugh\_product[6], t[5], 0, y, a6);

// Shift Register

always @(posedge clk) begin

if (rst == 1) begin

x\_values[0] <= 0;

x\_values[1] <= 0;

x\_values[2] <= 0;

x\_values[3] <= 0;

x\_values[4] <= 0;

x\_values[5] <= 0;

x\_values[6] <= 0;

// y <= 0;

$display("%b,%b,%b,%b,%b,%b,%b,%b",x,x\_values[0],x\_values[1],x\_values[2],x\_values[3],x\_values[4],x\_values[5],x\_values[6]);

end else begin

x\_values[0] <= x;

x\_values[1] <= x\_values[0];

x\_values[2] <= x\_values[1];

x\_values[3] <= x\_values[2];

x\_values[4] <= x\_values[3];

x\_values[5] <= x\_values[4];

x\_values[6] <= x\_values[5];

$display("%b,%b,%b,%b,%b,%b,%b,%b",x,x\_values[0],x\_values[1],x\_values[2],x\_values[3],x\_values[4],x\_values[5],x\_values[6]);

end

end

assign y = baugh\_product[0]+baugh\_product[1]+baugh\_product[2]+baugh\_product[3]+ baugh\_product[4]+baugh\_product[5]+baugh\_product[6];

//$display("%b,%b,%b,%b,%b,%b,%b,%b",y,baugh\_product[0],baugh\_product[1],baugh\_product[2],baugh\_product[3],baugh\_product[4],baugh\_product[5],baugh\_product[6]);

// /\* Accumulate stage of FIR \*/

// always @ (posedge clk)

// begin

// if (rst==1)

// begin

// y <= 0;

// end

// else begin

// y = baugh\_product[0]+baugh\_product[1]+baugh\_product[2]+baugh\_product[3]+ baugh\_product[4]+baugh\_product[5]+baugh\_product[6];

// $display("%b,%b,%b,%b,%b,%b,%b,%b",y,baugh\_product[0],baugh\_product[1],baugh\_product[2],baugh\_product[3],baugh\_product[4],baugh\_product[5],baugh\_product[6]);

// end

// Output Assignment

//end

endmodule

#**Testbench:**

**`timescale 1ns / 1ps**

**//////////////////////////////////////////////////////////////////////////////////**

**// Company:**

**// Engineer:**

**//**

**// Create Date: 05/07/2024 03:50:49 PM**

**// Design Name:**

**// Module Name: firTB**

**// Project Name:**

**// Target Devices:**

**// Tool Versions:**

**// Description:**

**//**

**// Dependencies:**

**//**

**// Revision:**

**// Revision 0.01 - File Created**

**// Additional Comments:**

**//**

**//////////////////////////////////////////////////////////////////////////////////**

**module firTB;**

**reg clk,rst;**

**reg [7:0] a0,a1,a2,a3,a4,a5,a6,x;**

**wire [15:0] y;**

**FIR FIR11(clk,rst,x,a0,a1,a2,a3,a4,a5,a6,y);**

**initial begin**

**clk =0;**

**end**

**always**

**#5 clk = !clk;**

**initial begin**

**rst=1;x=8'b00000000;**

**a0=8'b00000000; //00 => 0.0037**

**a1=8'b11110111; //F7 =>-0.067**

**a2=8'b00100011; //23 =>0.283**

**a3=8'b01111111; //7F=>1**

**a4=8'b00100011; //23=>0.283**

**a5=8'b11110111; //7F=>-0.067**

**a6=8'b00000000; //00=>0.0037**

**#7**

**rst=0;**

**#12 x=8'b00100000; //x7=0.625**

**#12 x=8'b00100000; //x6=0.25**

**#12 x=8'b01000000; //x5=0.75**

**#12 x=8'b01010000; //x4=0.625**

**#12 x=8'b01000000; //x3=0.5**

**#12 x=8'b00100000; //x2=0.25**

**#12 x=8'b01000000; //x1=0.5**

**#12 x=8'b00010000; //x0= 0.125**

**//#10 rst = 0; a=4'd2; x=4'd1;**

**//#10 a=4'd2; x=4'd2;**

**//#10 a=4'd2; x=4'd1;**

**//#10 rst=1; a=4'd2; x=4'd1;**

**//#10 rst=0;a=4'd2; x=4'b1101;**

**//#10 a=4'd2; x=4'd2;**

**end**

**endmodule**