`timescale 1ns / 1ps

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

// Company:

// Engineer:

//

// Create Date: 14:25:01 08/30/2011

// Design Name: alu

// Module Name: C:/Documents and Settings/Administrator/ALU/alutest.v

// Project Name: ALU

// Target Device:

// Tool versions:

// Description:

//

// Verilog Test Fixture created by ISE for module: alu

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

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

module alutest\_edgecases;

// Inputs

reg [15:0] A;

reg [15:0] B;

reg [7:0] Opcode;

reg [15:0] aluControl;

// Outputs

wire [15:0] C;

wire [4:0] Flags;

integer i;

integer failed;

// Instantiate the Unit Under Test (UUT)

ALU uut (

.In1(A),

.In2(B),

.aluControl(aluControl),

.Out(C),

.Flags(Flags)

);

// $monitor("A: %0d, B: %0d, C: %0d, Flags[1:0]:

//%b, time:%0d", A, B, C, Flags[1:0], $time );

//Instead of the $display stmt in the loop, you could use just this

//monitor statement which is executed everytime there is an event on any

//signal in the argument list.

initial begin

// Initialize Inputs

A = 0;

B = 0;

Opcode = 2'b0000\_0000;

failed = 0;

// Wait 100 ns for global reset to finish

/\*\*\*\*\*

// One vector-by-vector case simulation

#10;

Opcode = 2'b11;

A = 4'b0010; B = 4'b0011;

#10

A = 4'b1111; B = 4'b 1110;

//$display("A: %b, B: %b, C:%b, Flags[1:0]: %b, time:%d", A, B, C, Flags[1:0], $time);

\*\*\*\*/

// simulation

////////////1:

//cmp something where A<B

#10

B = 16'b 0000\_1010\_0000\_0000;

A = 16'b 1000\_0000\_1001\_1111;

aluControl = 16'b 0000\_0000\_1011\_0000;

#10

$display("edge case 1: cmp %h, %h evaluated to %h", A, B, C);

//check that N & L flag is set

if(Flags != 5'b 00011)

begin

failed = 1;

$display("Failed in edge case 1: cmp %h, %h has wrong flags %b (should be 00011)", A,B,Flags);

end

//and same number

#10

A = 16'b 1000\_1000\_0000\_0000;

B = 16'b 1000\_1000\_0000\_0000;

aluControl = 16'b 0000\_0000\_0001\_0000;

#10

$display("edge case 1: and %h, %h evaluated to %h", A, B, C);

// check that cmp flags reset & z flag is set

if(Flags != 5'b 00000)

begin

failed = 1;

$display("Failed in edge case 1: and %h, %h has wrong flags %b (should be 00000)", A,B,Flags);

end

//check if passed or not

if(failed == 1)

begin

$display("Failed in edge case 1, see previous outputs");

end

failed = 0;

/////////////2:

// add biggest numbers

#10

A = 16'b 0111\_1111\_1111\_1111;

B = 16'b 0111\_1111\_1111\_1111;

aluControl = 16'b 0000\_0000\_0101\_0000;

#10

$display("edge case 2: %h + %h evaluated to %h", A, B, C);

//check that F flag set

if(Flags != 5'b 00100)

begin

failed = 1;

$display("Failed in edge case 2: add %h, %h has wrong flags %b (should be 00100)", A,B,Flags);

end

// nop to check that no flags set

#10

A = 16'b 1111\_1111\_1111\_1111;

B = 16'b 1111\_1111\_1111\_1111;

aluControl = 16'b 0000\_0000\_0000\_0000;

#10

$display("edge case 2: nop");

// check that no flags set

if(Flags != 5'b 00000)

begin

failed = 1;

$display("Failed in edge case 2: nop has wrong flags %b (should be 00000)", Flags);

end

//check if passed or not

if(failed == 1)

begin

$display("Failed in edge case 2, see previous outputs");

end

failed = 0;

///////////3:

// addu biggest num + 1

#10

A = 16'b 1111\_1111\_1111\_1111;

B = 16'b 0000\_0000\_0000\_0001;

aluControl = 16'b 0000\_0000\_0110\_0000;

#10

$display("edge case 3: addu %h, %h evaluated to %h", A, B, C);

// check CFZ flags set

if(Flags != 5'b 11100)

begin

failed = 1;

$display("Failed in edge case 3: addu %h, %h has wrong flags %b (should be 11100)", A,B,Flags);

end

// sub 1 , 5

#10

A = 16'b 0000\_0000\_0000\_0001;

B = 16'b 0000\_0000\_0000\_0101;

aluControl = 16'b 0000\_0000\_1001\_0000;

#10

$display("edge case 3: sub %h, %h evaluated to %h", A, B, C);

// check no flags set

if(Flags != 5'b 01000)

begin

failed = 1;

$display("Failed in edge case 3: sub %h, %h has wrong flags %b (should be 01000)", A,B,Flags);

end

//check if passed or not

if(failed == 1)

begin

$display("Failed in edge case 3, see previous outputs");

end

failed = 0;

/////////////4:

// subi 1-1

#10

A = 16'b 0000\_0000\_0000\_0001;

B = 16'b 0000\_0000\_0000\_0001;

aluControl = 16'b 1001\_0000\_0000\_0001;

#10

$display("edge case 4: subi %h, %h evaluated to %h", A, B, C);

// check z flag

if(Flags != 5'b 10000)

begin

failed = 1;

$display("Failed in edge case 4: subi %h, %h has wrong flags %b (should be 10000)", A,B,Flags);

end

// not something

#10

A = 16'b 0000\_0000\_0000\_0001;

B = 16'b 0000\_0000\_0000\_0001;

aluControl = 16'b 0000\_0000\_1111\_0001;

#10

$display("edge case 4: not %h evaluated to %h", A, C);

// check no flags

if(Flags != 5'b 00000)

begin

failed = 1;

$display("Failed in edge case 4: not %h has wrong flags %b (should be 00000)", A,Flags);

end

//check if passed or not

if(failed == 1)

begin

$display("Failed in edge case 4, see previous outputs");

end

failed = 0;

///////////5:

//addc 32768 + 32768

#10

A = 16'b 1000\_0000\_0000\_0000;

B = 16'b 1000\_0000\_0000\_0000;

aluControl = 16'b 0000\_0000\_0111\_0000;

#10

$display("edge case 5: addc %h, %h evaluated to %h", A, B, C);

// output sets carry (zero too?)

if(Flags != 5'b 01100)

begin

failed = 1;

$display("Failed in edge case 5: addc %h, %h has wrong flags %b (should be 10000)", A,B,Flags);

end

// shift?

#10

A = 16'b 0000\_0000\_1000\_0000;

B = 16'b 0000\_0000\_0000\_0001;

aluControl = 16'b 0000\_0000\_1000\_0000;

#10

$display("edge case 5: lsh %h evaluated to %h", A, C);

// check flags

if(Flags != 5'b 00000)

begin

failed = 1;

$display("Failed in edge case 5: lsh %h has wrong flags %b (should be 00000)", A,Flags);

end

//check if passed or not

if(failed == 1)

begin

$display("Failed in edge case 5, see previous outputs");

end

end

endmodule