Josh Monson
620 Homework 2
9/17/2014

## Test Plan

Verify That:

1. C becomes 0 after reset is asserted.
2. When opcode is 2'b00 the sum of A and B appears on C after one clock cycle.
    a. Must be two's complement
3. When opcode is 2'b01 the difference of A and B appears on C after clock cycle.
    a. Must be two's complement
4. When opcode is 2'b10 the bitwise inverse of A appears on C after one clock cycle.
5. When opcode is 2'b11 the reduction OR of B appears on C after one clock cycle.

## Procedure

The fact that there were only 11 input bits made it possible to exhaustively test all 2048 input combinations. For each input combination, the result was calculated and then compared to the output of the ALU (C). For all input combinations the ALU worked performed properly.

However, it is not possible in most cases to exhaustively test all input combinations to a circuit. So it might be interesting to consider which directed tests I might have performed to ensure that the circuit was working correctly.

To prove that the reset is functioning properly, we would want to ensure that the register resets to zero when the reset is asserted regardless of which opcode or inputs were applied. My directed tests would have included asserting the reset while applying several randomly chosen opcode and inputs and demonstrating that the circuit operates properly.

For add and subtract I would have tried a variety of random combinations of positive and negative numbers. I would also increment and decrement by one through the entire space. Then I may also have incremented by powers of 2 (2, 4, 8, etc.). To get a good feel that each of the bits in the adder was individually responsive.

Bitwise operation are the easiest to vet since each bit is operated on individually. The correct functionality can be shown using two tests: all ones and all zeros. Other directed tests could include rotating a single one or single zero through each of the inputs. This help demonstrate that each of the bits is independent of the other bits. It would also be important to show that the result is independent of the B input. This would be done by cycling different patterns through B while performing additional tests on B.

For the reduction OR I would have rotated a single one through each of the inputs as well as have tested all ones and all zeros. Then I would have tested several inputs on A to show that the result is independent of A.

## ALU Test-bench Code

```
`default_nettype none
`timescale 1ns/100ps

module alu_testbench ();

  reg clk;
  wire reset;
  wire [1:0] Opcode;
  wire signed [3:0] A;
  wire signed [3:0] B;
  wire signed [4:0] C;

  reg signed [10:0] test_vector;
  reg [4:0] expected_result;
  integer errors;

  always begin
    #10 clk = !clk;
  end

  ALU_4_bit ALU(clk,
                reset,
                Opcode,
                A, B, C );

  assign A =  test_vector[3:0];
  assign B =  test_vector[7:4];
  assign reset = test_vector[8];
  assign Opcode = test_vector[10:9];

  initial begin
    $display("Starting Testbench:");
    clk = 0;
    test_vector = 11'b11111111111;
    errors = 16'd0;
    forever begin
      @ (negedge clk)
      //Update Test Vector
      test_vector = test_vector + 1'b1;
      //Wait to allow wire value to propegate
      #1 //ns
      //Define Expected Results based on Current Inputs
       if(reset) begin
          expected_result = 5'b0;
       end else begin
         case (Opcode)
           2'b00: expected_result = A + B;
           2'b01: expected_result = A - B;
           2'b10: expected_result = ~A;
```

```
            2'b11: expected_result = |B;
          endcase
        end

      @ (posedge clk)

      #1 //ns

      if(expected_result != C) begin
        errors = errors + 1;
        $display("Error @ time %d: Reset: %d Opcode: %d A: %d B: %d C: %d
Expected Result: %d", $time, reset, Opcode, A, B, C, expected_result);
      end

      if(test_vector == 11'b11111111111) begin
        $display("Test Complete: Found %d Errors", errors);
        $finish;
      end


    end
  end

endmodule
```

**Transcript Window Output (Working)**

do testbench.do
# vsim -novopt alu_testbench
# Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment1/work.alu_testbench
# Loading work.alu_testbench

# Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment1/work.ALU_4_bit
# Loading work.ALU_4_bit
# Starting Testbench:
# Test Complete: **Found 0 Errors**
# ** Note: $finish : /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment1/alu_testbench.v(65)

#  Time: 40971 ns  Iteration: 0  Instance: /alu_testbench

## Transcript Window Output (Not Working) Changed ~A to A

```
o testbench.do

# vsim -novopt alu_testbench
# Refreshing
/net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment1/work.alu_testbench
# Loading work.alu_testbench
# Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment1/work.ALU_4_bit
# Loading work.ALU_4_bit
# Starting Testbench:

# Error @ time            20511: Reset: 0 Opcode: 2 A:   0 B:   0 C:   0 Expected Result: 31
# Error @ time            20531: Reset: 0 Opcode: 2 A:   1 B:   0 C:   1 Expected Result: 30
# Error @ time            20551: Reset: 0 Opcode: 2 A:   2 B:   0 C:   2 Expected Result: 29
# Error @ time            20571: Reset: 0 Opcode: 2 A:   3 B:   0 C:   3 Expected Result: 28
# Error @ time            20591: Reset: 0 Opcode: 2 A:   4 B:   0 C:   4 Expected Result: 27
# Error @ time            20611: Reset: 0 Opcode: 2 A:   5 B:   0 C:   5 Expected Result: 26
# Error @ time            20631: Reset: 0 Opcode: 2 A:   6 B:   0 C:   6 Expected Result: 25
# Error @ time            20651: Reset: 0 Opcode: 2 A:   7 B:   0 C:   7 Expected Result: 24
# Error @ time            20671: Reset: 0 Opcode: 2 A:  -8 B:   0 C:  -8 Expected Result:  7
# Error @ time            20691: Reset: 0 Opcode: 2 A:  -7 B:   0 C:  -7 Expected Result:  6

…

#Error @ time            25411: Reset: 0 Opcode: 2 A:   5 B:  -1 C:   5 Expected Result: 26
# Error @ time            25431: Reset: 0 Opcode: 2 A:   6 B:  -1 C:   6 Expected Result: 25
# Error @ time            25451: Reset: 0 Opcode: 2 A:   7 B:  -1 C:   7 Expected Result: 24
# Error @ time            25471: Reset: 0 Opcode: 2 A:  -8 B:  -1 C:  -8 Expected Result:  7
# Error @ time            25491: Reset: 0 Opcode: 2 A:  -7 B:  -1 C:  -7 Expected Result:  6
# Error @ time            25511: Reset: 0 Opcode: 2 A:  -6 B:  -1 C:  -6 Expected Result:  5
# Error @ time            25531: Reset: 0 Opcode: 2 A:  -5 B:  -1 C:  -5 Expected Result:  4
# Error @ time            25551: Reset: 0 Opcode: 2 A:  -4 B:  -1 C:  -4 Expected Result:  3
# Error @ time            25571: Reset: 0 Opcode: 2 A:  -3 B:  -1 C:  -3 Expected Result:  2
# Error @ time            25591: Reset: 0 Opcode: 2 A:  -2 B:  -1 C:  -2 Expected Result:  1
# Error @ time            25611: Reset: 0 Opcode: 2 A:  -1 B:  -1 C:  -1 Expected Result:  0

# Test Complete: Found         256 Errors

# ** Note: $finish    :
/net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment1/alu_testbench.v(65)

#    Time: 40971 ns  Iteration: 0  Instance: /alu_testbench
```