Sai Kathika saiprem@ucsb.edu
Pete Makrygiannis pmakrygiannis@ucsb.edu
Rahul Varghese rvarghese@ucsb.edu

ECE 154A Lab 1

1. Please indicate how many hours you spent on this lab. This will be helpful for calibrating the workload for next time the course is taught. Failure to provide may result in a loss of points.

My group worked on this project for a total of 7 hours.

2. Your table of test vectors (Table 1).

	Test	F[2:0]	A	B	Y	Zero
2	ADD 0+0	2	OOOOOOOO	OOOOOOOO	OOOOOOOO	1
3	ADD 0+(-1)	2	OOOOOOOO	FFFFFFFF	FFFFFFFF	0
4	ADD 1+(-1)	2	OOOOOOO1	FFFFFFFF	OOOOOOOO	1
5	ADD FF+1	2	OOOOOOFF	OOOOOOO1	OOOOO100	0
6	SUB 0-0	6	OOOOOOOO	OOOOOOOO	OOOOOOOO	1
7	SUB 0-(-1)	6	OOOOOOO0	FFFFFFFF	OOOOOOO1	0
8	SUB 1-1	6	OOOOOOO1	OOOOOOO1	OOOOOOOO	1
9	SUB 100-1	6	OOOOO1OO	OOOOOOO1	OOOOO063	0
10	SLT 0,0	7	OOOOOOOO	OOOOOOOO	OOOOOOOO	1
11	SLT 0,1	7	OOOOOOOO	OOOOOOO1	OOOOOOO1	0
12	SLT 0,-1	7	OOOOOOOO	FFFFFFFF	OOOOOOOO	1
13	SLT 1,0	7	OOOOOOO1	OOOOOOOO	OOOOOOOO	1
14	SLT -1,0	7	FFFFFFFF	OOOOOOOO	OOOOOOO1	0
15	AND FFFFFFFF, FFFFFFFF	3	FFFFFFFF	FFFFFFFF	FFFFFFFF	0
16	AND FFFFFFFF, 12345678	3	FFFFFFFF	12345678	12345678	0
17	AND 12345678, 87654321	3	12345678	87654321	2244220	0
18	AND 00000000, FFFFFFFF	3	OOOOOOOO	FFFFFFFF	OOOOOOOO	1
19	OR FFFFFFFF, FFFFFFFF	4	FFFFFFFF	FFFFFFFF	FFFFFFFF	0
20	OR 12345678, 87654321	4	12345678	87654321	97755779	0
21	OR 00000000, FFFFFFFF	4	OOOOOOOO	FFFFFFFF	FFFFFFFF	0
22	OR 00000000, 00000000	4	OOOOOOOO	OOOOOOOO	OOOOOOOO	1

3. Your alu.v file.

```
module alu (input logic signed [31:0] a,
                    input logic signed [31:0] b,
                    input logic [2:0] f,
                    input logic clk,
                    output logic signed [31:0] y,
                    output reg zero
);
  reg [31:0] sub32bit;

  always @(posedge clk) begin
    if (f == 3'b010) begin
      y = (a + b);
    end
    if (f == 3'b110) begin
      y <= (a - b);
    end
    if (f == 3'b011) begin
      y <= (a & b);
    end
    if (f == 3'b100) begin
      y <= (a | b);
    end
    if (f == 3'b111) begin
          sub32bit <= (a - b);
      y = sub32bit[31] ? 1 : 32'h0;
    end
    if(y == 32'h0) begin
                    assign zero = 1;
    end else begin
                    assign zero = 0;
          end
  end
Endmodule
```

4. Your alu.tv file.

```
2 00000000 00000000 00000000 1
2 00000000 FFFFFFFF FFFFFFFF 0
2 00000001 FFFFFFFF 00000000 1
2 000000FF 00000001 00000100 0
6 00000000 00000000 00000000 1
6 00000000 FFFFFFFF 00000001 0
6 00000001 00000001 00000000 1
6 00000100 00000001 00000063 0
7 00000000 00000000 00000000 1
7 00000000 00000001 00000001 0
7 00000000 FFFFFFFF 00000000 1
7 00000001 00000000 00000000 1
7 FFFFFFFF 00000000 00000001 0
3 FFFFFFFF FFFFFFFF FFFFFFFF 0
3 FFFFFFFF 12345678 12345678 0
3 12345678 87654321 02244220 0
3 00000000 FFFFFFFF 00000000 1
3 FFFFFFFF FFFFFFFF FFFFFFFF 0
3 12345678 87654321 97755779 0
3 00000000 FFFFFFFF FFFFFFFF 0
3 00000000 00000000 00000000 1
```

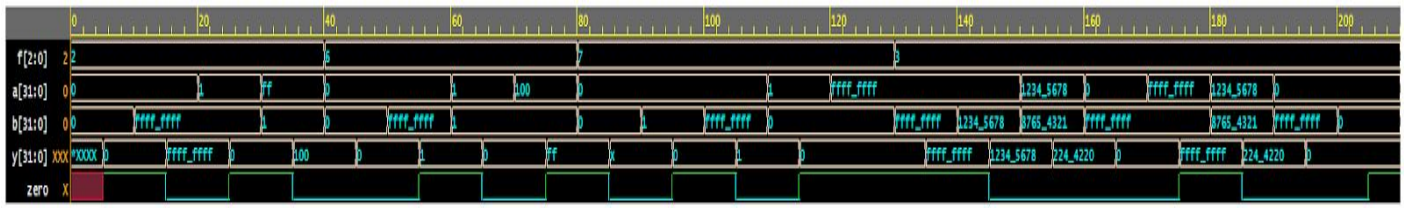
5. Your testbench.v file.

```
        module alu_tb #(
);

  logic clk;
  reg [31:0] data [0:105];
  reg [2:0] f;
  reg [31:0] a;
  reg [31:0] b;
  reg flag;
  integer i;
  reg [31:0] y_i;
  logic [31:0] y;

  reg zero;
  reg zero_i;


initial begin
  clk = $urandom_range(1,0);
  forever begin
    #5ns;
    clk =!clk;
  end
end

//initial begin

  alu #()
  alu_inst (
          .a (a),
          .b (b),
          .f (f),
                  .y (y),
    .zero (zero),
          .clk (clk)
          );

initial begin
  $dumpfile("dump.vcd");
  $dumpvars;
end
```

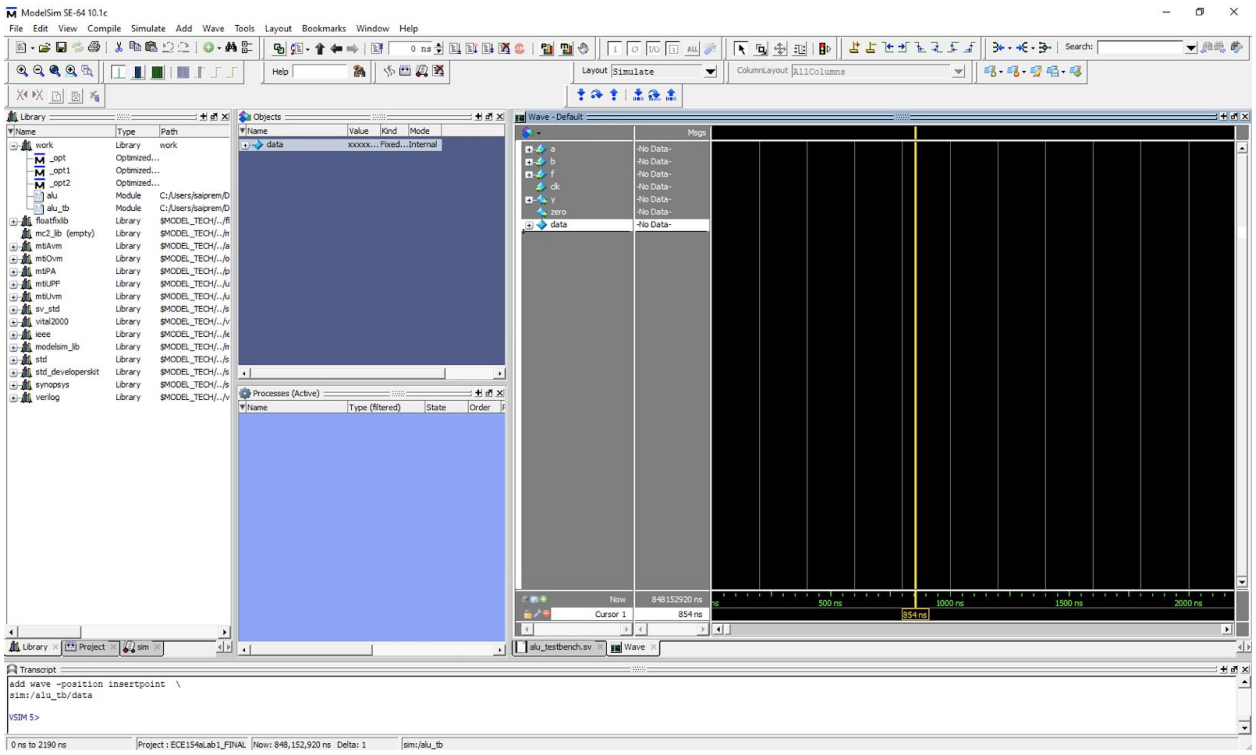
```
initial $readmemh("alu.tv", data);
//initial $readmemh("test.tv", data);
initial i = 0;
  always @ (negedge clk) begin
    //for (i=0; i < 15; i = i+5) begin
    f = data[i];
    a = data[i+1];
    b = data[i+2];
    y_i = data[i+3];
    zero_i = data[i+4];
      i = i+5;
/*    $display("y : %h", y);
    $display("f : %b", f);
    $display("a : %h", a);
    $display("b : %h", b);
    $display("y_i : %h", y_i);
    $display("zero_i : %h", zero_i);
    if (y_i != y) begin
        $display("Failed!");
      end else if (zero_i != zero) begin
        $display("Failed!");
    end
*/
    if (i > 106) begin
      $finish;
    end

      //$display("f: %d, a: %h, b: %h, y: %h, zero: %d", f, a, b, y, zero);
    end
endmodule
```

6. Images of your test waveforms. Make sure these are readable and that they're printed in hexadecimal. Your test waveforms should include only the following signals in the following order, from top to bottom: f, a, b, y, zero. Please change the radix of the f signal to unsigned decimal, and a, b, and y signals to hexadecimal.

This is the waveform produced by eda playground.



Our group spent a lot of time trying to get modelsim to output the waveforms, but we were unable to.



7. If you have any feedback on how we might make the lab even better for next quarter, that's always welcome. Please submit it in writing at the end of your lab

Modelsim was a completely new environment that I felt complicated this lab and made my group put in many more hours than what was required. Additionally, the errors that modelsim produced did not help us debug our code. We spent numerous hours trying to learn how to simulate and produce waveforms(watched numerous tutorials and read articles) but we still had

no luck. For ECE 152A we used eda playground which was much more intuitive and friendly to use.