Lab 1 Report

ECE 154A

Vishal Seenivasan

1. Hours Spent on Lab

This lab took about 10 hours of work across the work total to complete. The majority of my time was spent trying to figure out specific details of Verilog's behavior. Specifically, I had trouble figuring out how to utilize the memreadh function to read in testvectors from the tv file into the ModelSim simulation.

2. Table of Test Vectors

Test	F[2:0]	A	B	Result	Zero	Over	Car	Neg
ADD 0+0	0	00000000	00000000	00000000	1	0	0	0
ADD 0+(-1)	0	00000000	FFFFFFFF	FFFFFFFF	0	0	0	1
ADD 1+(-1)	0	00000001	FFFFFFFF	00000000	1	0	1	0
ADD FF+1	0	000000FF	00000001	00000100	0	0	0	0
ADD 7FFFFFFF+1	0	7FFFFFFF	00000001	80000000	0	1	0	1
SUB 0-0	1	00000000	00000000	00000000	1	0	1	0
SUB 0-(-1)	1	00000000	FFFFFFFF	00000001	0	0	0	0
SUB 1-1	1	00000001	00000001	00000000	1	0	1	0
SUB 100-1	1	00000100	00000001	000000FF	0	0	1	0
SUB 80000000-1	1	80000000	00000001	7FFFFFFF	0	1	1	0
SLT 0,0	5	00000000	00000000	00000000	1	0	0	0
SLT 0,1	5	00000000	00000001	00000001	0	0	0	0
SLT 0,-1	5	00000000	FFFFFFFF	00000000	1	0	0	0
SLT 1,0	5	00000001	00000000	00000000	1	0	0	0
SLT -1,0	5	FFFFFFFF	00000000	00000001	0	0	0	0
AND FFFFFFFF, FFFFFFFF	2	FFFFFFFF	FFFFFFFF	FFFFFFFF	0	0	0	1
AND FFFFFFFF, 12345678	2	FFFFFFFF	12345678	12345678	0	0	0	0
AND 12345678, 87654321	2	12345678	87654321	02244220	0	0	0	0
AND 00000000, FFFFFFFF	2	00000000	FFFFFFFF	00000000	1	0	0	0
OR FFFFFFFF, FFFFFFFF	3	FFFFFFFF	FFFFFFFF	FFFFFFFF	0	0	0	1
OR 12345678, 87654321	3	12345678	87654321	97755779	0	0	0	1
OR 00000000, FFFFFFFF	3	00000000	FFFFFFFF	FFFFFFFF	0	0	0	1
OR 00000000, 00000000	3	00000000	00000000	00000000	1	0	0	0

Table 1. ALU Operations

3. alu.v file

```
module alu(input [31:0] a, b,
           input [2:0] f,
           output [31:0] result,
           output zero,
           output overflow,
           output carry,
           output negative);

          wire [31:0] adder_b;
          wire [31:0] and_result;
          wire [31:0] or_result;
          wire [31:0] sum;
          wire [31:0]slt;
          wire cout;

          assign adder_b = f[0] ? ~b : b;
          assign and_result = a & b;
          assign or_result = a | b;
          assign {cout, sum} = a + adder_b + f[0];
          assign slt[31:1] = 0;
          assign slt[0] = overflow ^ sum[31];

          assign result = (f[2] & f[0]) ? slt : (f[1] ? (f[0]?
or_result : and_result) : (f[0]? sum : sum));
          assign zero = &(~result);
          assign overflow = (~(f[0] ^ b[31] ^ a[31]) & (a[31] ^
sum[31]) & (~f[1]));
          assign carry = (~f[1] & ~f[2]) & cout;
          assign negative = result[31];
endmodule
```

4. alu.tv file

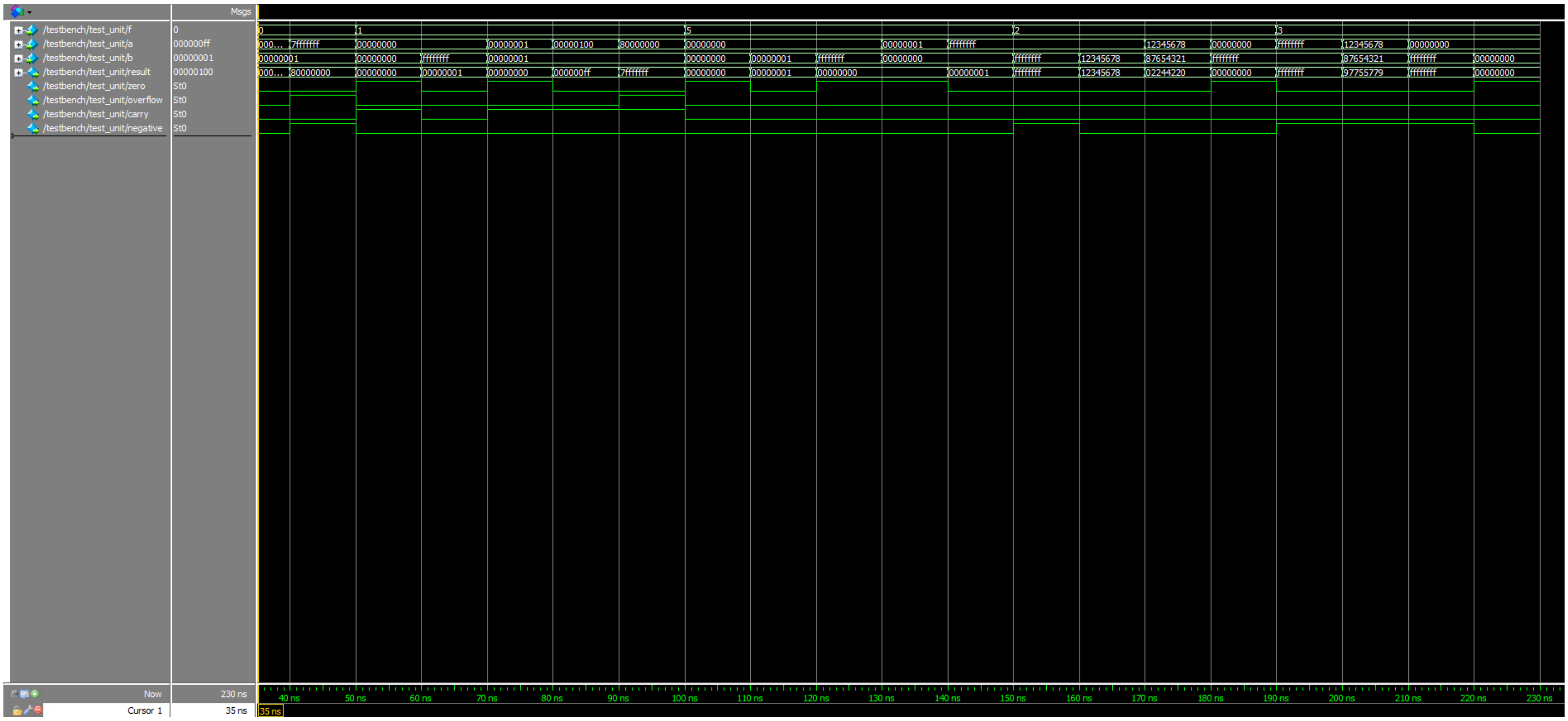
```
0 00000000 00000000 00000000 1 0 0 0
0 00000000 FFFFFFFF FFFFFFFF 0 0 0 1
0 00000001 FFFFFFFF 00000000 1 0 1 0
0 000000FF 00000001 00000100 0 0 0 0
0 7FFFFFFF 00000001 80000000 0 1 0 1
1 00000000 00000000 00000000 1 0 1 0
1 00000000 FFFFFFFF 00000001 0 0 0 0
1 00000001 00000001 00000000 1 0 1 0
1 00000100 00000001 000000FF 0 0 1 0
1 80000000 00000001 7FFFFFFF 0 1 1 0
5 00000000 00000000 00000000 1 0 0 0
5 00000000 00000001 00000001 0 0 0 0
5 00000000 FFFFFFFF 00000000 1 0 0 0
```

```
5 00000001 00000000 00000000 1 0 0 0
5 FFFFFFFF 00000000 00000001 0 0 0 0
2 FFFFFFFF FFFFFFFF FFFFFFFF 0 0 0 1
2 FFFFFFFF 12345678 12345678 0 0 0 0
2 12345678 87654321 02244220 0 0 0 0
2 00000000 FFFFFFFF 00000000 1 0 0 0
3 FFFFFFFF FFFFFFFF FFFFFFFF 0 0 0 1
3 12345678 87654321 97755779 0 0 0 1
3 00000000 FFFFFFFF FFFFFFFF 0 0 0 1
3 00000000 00000000 00000000 1 0 0 0
```

5. testbench.v file

```
module alu_testbench();
  reg[31:0] A, B;
  reg[2:0] control;

  wire[31:0] result;
  wire zero, overflow, carry, negative;

  reg[31:0] expected_result;
  reg ex_zero, ex_overflow, ex_carry, ex_negative;
  reg[31:0] testvector[0:175];
  integer i;

  alu test_unit(
    .a(A),
    .b(B),
    .f(control),
    .result(result),
    .zero(zero),
    .overflow(overflow),
    .carry(carry),
    .negative(negative));

  initial begin

    $readmemh("alu.tv", testvector);
    for(i = 0; i < 175; i = i+8) begin
      control = testvector[i][2:0];
      A = testvector[i+1][31:0];
      B = testvector[i+2][31:0];
      expected_result = testvector[i+3][31:0];
      ex_zero = testvector[i+4][0];
      ex_overflow = testvector[i+5][0];
      ex_carry = testvector[i+6][0];
      ex_negative = testvector[i+7][0];
      #10;

      if((result == expected_result) && (zero == ex_zero) &&
(overflow == ex_overflow) && (carry == ex_carry) && (negative ==
ex_negative)) begin
        $display("Test passed");
      end else begin
        $display("Test failed");
        $display("Inputs - control: %b, A: %h, B: %h", control, A,
B);
        $display("Outputs - result: %h, zero: %b, overflow: %b,
carry: %b, negative: %b", result, zero, overflow, carry, negative);
        $display("Expected - result: %h, zero: %b, overflow: %b,
carry: %b, negative: %b", expected_result, ex_zero, ex_overflow,
ex_carry, ex_negative);
      end
    end
  end
endmodule
```

6. Test Waveforms


Msgs
/testbench/test_unit/f 0
/testbench/test_unit/a 000000ff
/testbench/test_unit/b 00000001
/testbench/test_unit/result 00000100
/testbench/test_unit/zero St0
/testbench/test_unit/overflow St0
/testbench/test_unit/carry St0
/testbench/test_unit/negative St0
0 1 5 2 3
000... 7fffffff 00000000 00000001 00000100 80000000 00000000 00000001 ffffffff 12345678 00000000 ffffffff 12345678 00000000
00000001 00000000 ffffffff 00000001 00000000 00000001 ffffffff 00000000 ffffffff 12345678 87654321 ffffffff 87654321 ffffffff 00000000
000... 80000000 00000000 00000001 00000000 000000ff 7fffffff 00000000 00000001 00000000 00000001 ffffffff 12345678 02244220 00000000 ffffffff 97755779 ffffffff 00000000
Now 230 ns
Cursor 1 35 ns
40 ns 50 ns 60 ns 70 ns 80 ns 90 ns 100 ns 110 ns 120 ns 130 ns 140 ns 150 ns 160 ns 170 ns 180 ns 190 ns 200 ns 210 ns 220 ns 230 ns
35 ns