**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

