Northeastern University

Report for Experiment #1
Eight Bit Adder

Trevor Smith
February 10, 2022

Prelab:

1.

a	b	f	ovf
$8'd0$	$8'd0$	0000 0000	0
$8'd12$	$8'd34$	0000 1110	0
$-8'd12$	$-8'd34$	1101 0010	0
$8'd100$	$-8'd50$	0011 0010	0
$-8'd100$	$8'd50$	1100 1110	0
$8'd100$	$8'd100$	1100 1000	1
$-8'd100$	$-8'd100$	1 0011 1000	1
$8'd3$	$8'd0$	0000 0011	0

For -100 -100 I'm aware the 9th bit would be chopped off, just showing the theoretical value.

2.

$ab\backslash^f$	0	1
00	0	1
01	0	0
11	1	0
10	0	0

Where A=a[0], B=b[0], f=f[0],

$$\overline{AB}f + AB\overline{f} = ovf$$

3. We almost test every bit, which is somehow easier with two's complement since the minus sign has a tendency to make zeros into ones. However, we never test the 1's bit. As such, I added 3 + 0 as a test in the above table.

Results and Analysis:

An eight-bit adder was fully implemented, tested, and verified using the Vivado IDE. After some fiddling, the outputs were all found to match expected values. The program was then ported to the PYNQ board, using provided mappings, and tested in real-time. While the base implementation of the addition was just a wrapper for verilog's '+' operator, the overflow was developed and tested as well, which transformed the base verilog adder into a two's complement eight-bit adder.

Conclusion and Recommendations:

An eight-bit adder was successfully implemented and tested. The most significant challenge in completing this lab was by far the Vivado software and infamiliar verilog syntax. As such, this was a useful learning opportunity. In the future, more advanced functions and programs will be possible based on growing fluency with these tools.

Appendices:

0.1 Appendix A: Design Program Files

```
module eightbit_adder(
    input [7:0]a,
    input [7:0]b,
    output [7:0]f,
    output ovf
    );

    assign f = a + b;
    assign ovf=f[7]? ~(a[7]|b[7]):a[7]&b[7];

endmodule


module adder8_tb();

    //Inputs
    reg [7:0]a;
    reg [7:0]b;
    //Outputs
    wire [7:0]f;
    wire ovf;

    //UUT
    eightbit_adder UUT
       (.a(a),
        .b(b),
        .f(f),
        .ovf(ovf)
    );

    initial
        begin

            #200;
            a=8'b1;
            b=8'b1;

            #100;
            a = 8'd12;
            b = 8'd34;

            #100;
            a = -8'd12;
            b = -8'd34;

            #100;
            a = 8'd100;
            b = -8'd50;
```

```
            #100;
            a = -8'd100;
            b = 8'd50;

            #100;
            a = -8'd12;
            b = -8'd34;

            #100;
            a = 8'd100;
            b = 8'd100;

            #100;
            a = -8'd100;
            b = -8'd100;

        end

endmodule
```

0.2 Appendix B: Output Screen Capture

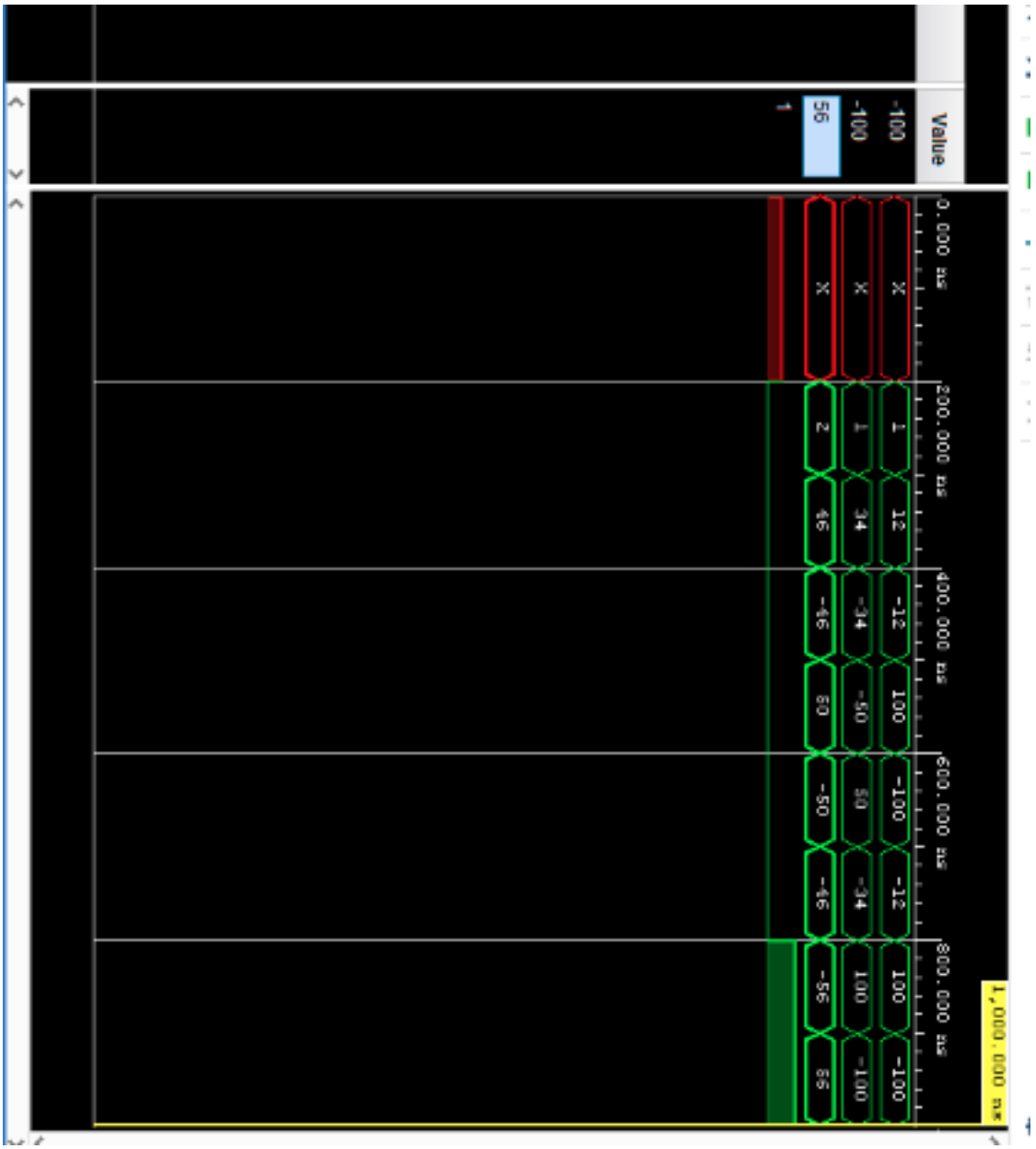

Figure 1: Test bench output