ELC 2137 Lab 05: Intro to Verilog

Yiting Wang

October 15, 2020

Summary

In this Lab we are going to describe digital circuits with code, or a hardware description language (HDL). That's because before we built digital circuits using hardware devices. This becomes cumbersome very quickly. It would be possible to draw circuit schematics in soft-ware then program them onto a device. This is easier/faster than physically wiring by hand and allows for much larger, more complicated designs, but still requires a significant amount of click-and-place.

Q&A

1. Comment on whether the simulations match the expected output values?
   In the half adder and the full adder, the simulations match the expected output values; but in the two bit adder/subtractor, the simulation doesn't match the expected output values.
2. What is one thing that you still dont understand about Verilog?
   I don't know a lot of it casue I am a beginner, but I think I understand all the knowledge in this lab.

Results

Firgure 1 is the block diagrams for half adder module.

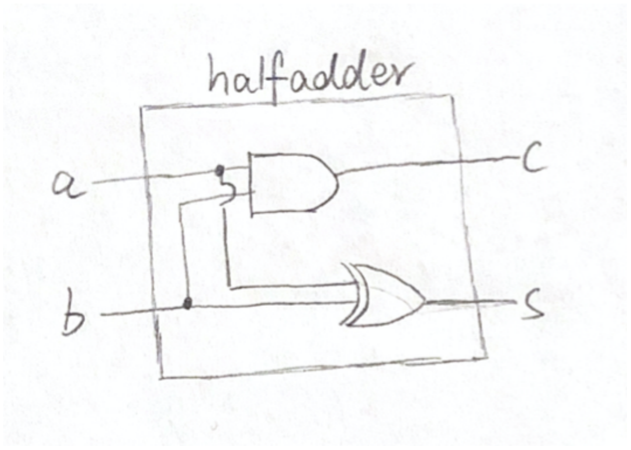

Figure 1: This is the block diagrams for half adder module.

Firgure 2 is the simulation waveform and ERT of half adder, this simulation matches the expected output values, and the code about that is in the Code section.

Time (ns):	0	10	20	30
a	0	0	1	1
b	0	1	0	1
c	0	0	0	1
s	0	1	1	0

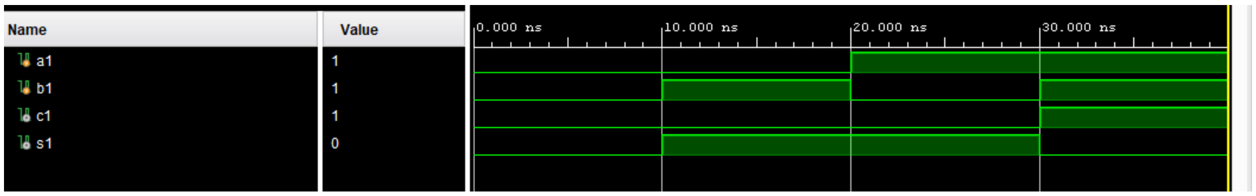

Figure 2: the simulation waveform and ERT of half adder

Firgure 3 is the block diagrams for full adder module.

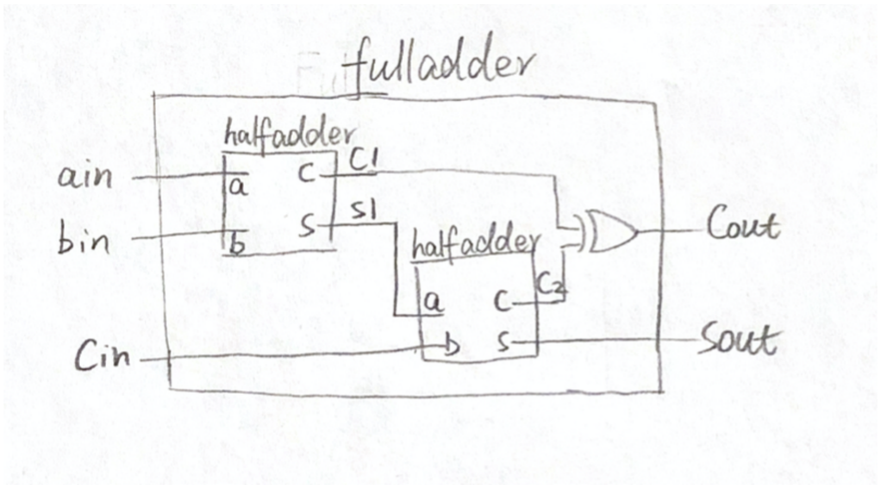

Figure 3: This is the block diagrams for full adder module.

Firgure 4 is the simulation waveform and ERT of full adder, this simulation matches the expected output values, and the code about that is in the Code section.

Firgure 5 is the block diagrams for two bit adder/subtractor module.

Firgure 6 is the simulation waveform and ERT of two bit adder/subtractor, this simulation doesn't match the expected output values, and the code about that is in the Code section.

Time (ns):	0	10	20	30	40	50	60	70
cin	0	0	0	0	1	1	1	1
a	0	0	1	1	0	0	1	1
b	0	1	0	1	0	1	0	1
c	0	0	0	1	0	1	1	1
s	0	1	1	0	1	0	0	1

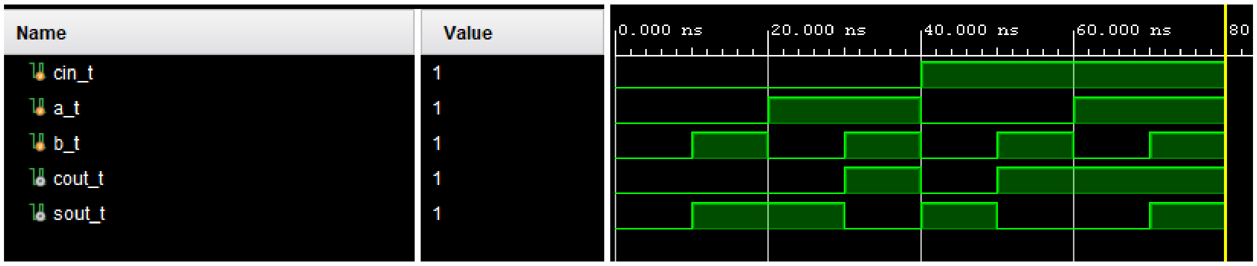

Figure 4: the simulation waveform and ERT of full adder

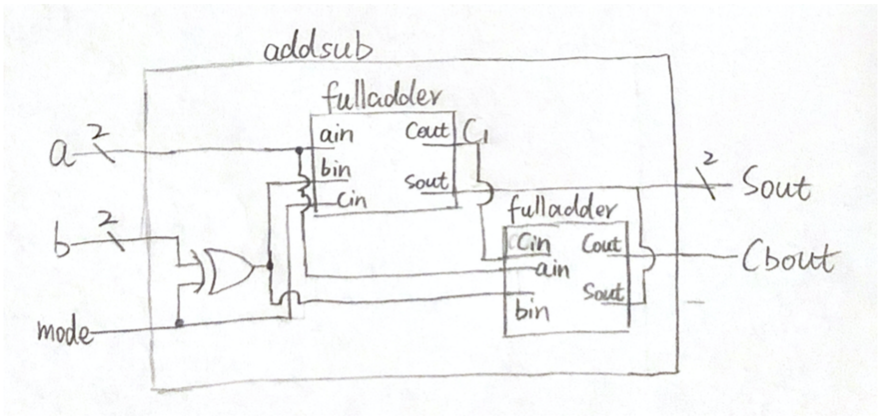

Figure 5: This is the block diagrams for two bit adder/subtractor module.

Code

File Inclusion

Listing 1: Half Adder Verilog code

```
‘timescale 1ns / 1ps
//
   ///////////////////////////////////////////////////////////////////////////////////

// Company: ELC 2137
// Engineer: Yiting Wang
// Create Date: 09/24/2020
//
   ///////////////////////////////////////////////////////////////////////////////////
```

Time (ns):	0	10	20	30	40	50	60	70	80	90	100	110	120	130
A1A0	00	00	00	00	01	10	10	00	00	00	00	01	10	10
B1B0	00	01	10	11	01	01	00	00	01	10	11	01	01	00
mode	0	0	0	0	0	0	0	1	1	1	1	1	1	1
c	0	0	0	0	0	0	0	0	1	1	1	0	0	0
s1	0	0	1	1	1	1	1	0	1	1	0	0	0	1
s0	0	1	0	1	0	1	0	0	1	0	1	0	1	0

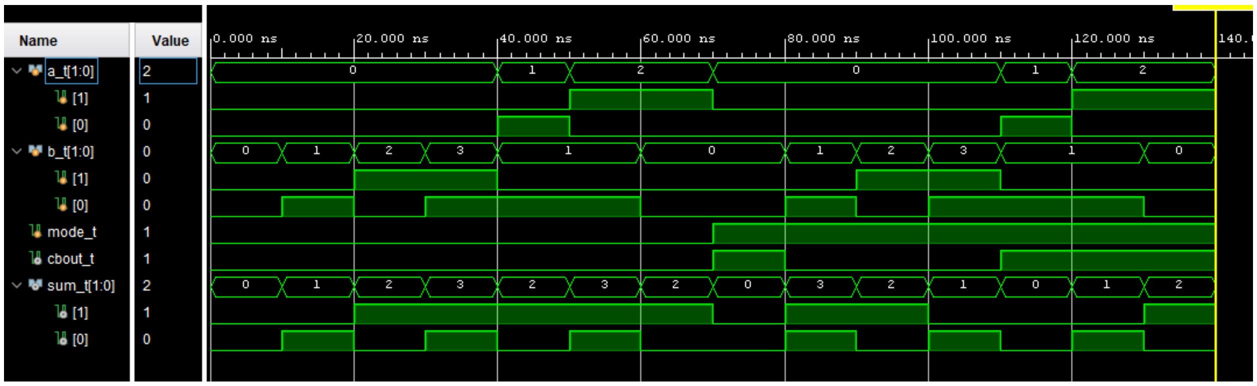

Figure 6: the simulation waveform and ERT of two bit adder/subtractor

```
module halfadder(
    input a,
    input b,
    output c,
    output s
    );

    assign c = a & b;
    assign s = a ^ b;

endmodule // halfadder
```

File Inclusion

Listing 2: Half Adder Test Benches Verilog code

```
`timescale 1ns / 1ps
//
   ////////////////////////////////////////////////////////////////////////////////

// Company: ELC 2137
// Engineer: Yiting Wang
// Create Date: 09/24/2020
//
   ////////////////////////////////////////////////////////////////////////////////



module halfadder_test();
```

```
    reg a1, b1;
    wire c1, s1;


    halfadder dut(
        .a(a1),.b(b1),
        .c(c1),.s(s1)
    );

    initial begin
        a1=0; b1=0; #10;
        a1=0; b1=1; #10;
        a1=1; b1=0; #10;
        a1=1; b1=1; #10;
        $finish;
    end

endmodule //halfadder_test
```

File Inclusion

Listing 3: Full Adder Verilog code

```
`timescale 1ns / 1ps
//
   ////////////////////////////////////////////////////////////////////////////////

// Company: ELC 2137
// Engineer: Yiting Wang
// Create Date: 09/24/2020
//
   ////////////////////////////////////////////////////////////////////////////////



module fulladder(
    input ain,
    input bin,
    input cin,
    output cout,
    output sout
    );

    wire c1, c2, s1;

    halfadder dut1(
        .a(ain), .b(bin),
        .c(c1), .s(s1)
    );

    halfadder dut2(
        .a(cin), .b(s1),
        .c(c2), .s(sout)
    );

    assign cout = c1 ^ c2;

endmodule// fulladder
```

File Inclusion

Listing 4: Full Adder Test Benches Verilog code

```
`timescale 1ns / 1ps
//
   ////////////////////////////////////////////////////////////////////////////////

// Company: ELC 2137
// Engineer: Yiting Wang
// Create Date: 09/24/2020
//
   ////////////////////////////////////////////////////////////////////////////////




module fulladder_test( );

    reg a_t, b_t, cin_t;
    wire cout_t, sout_t;

    fulladder dut(
        .ain(a_t), .bin(b_t), .cin(cin_t),
        .sout(sout_t), .cout(cout_t)
    );

    initial begin
        a_t=0; b_t=0; cin_t=0; #10;
        a_t=0; b_t=1; cin_t=0; #10;
        a_t=1; b_t=0; cin_t=0; #10;
        a_t=1; b_t=1; cin_t=0; #10;

        a_t=0; b_t=0; cin_t=1; #10;
        a_t=0; b_t=1; cin_t=1; #10;
        a_t=1; b_t=0; cin_t=1; #10;
        a_t=1; b_t=1; cin_t=1; #10;
        $finish;
    end

endmodule
```

File Inclusion

Listing 5: Two Bit Adder/Aubtractor Verilog code

```
`timescale 1ns / 1ps
```

```
//
   ////////////////////////////////////////////////////////////////////////////////

// Company: ELC 2137
// Engineer: Yiting Wang
// Create Date: 09/30/2020
//
   ////////////////////////////////////////////////////////////////////////////////



module addsub(
    input [1:0] a, b,
    input mode,
    output [1:0] sum,
    output cbout
    );

    wire c1, c2;
    wire [1:0] b_n;

    assign b_n[0] = mode ^ b[0];
    assign b_n[1] = mode ^ b[1];

    fulladder dut1(
        .ain(a[0]), .bin(b_n[0]), .cin(mode),
        .cout(c1), .sout(sum[0])
    );

    fulladder dut2(
        .ain(a[1]), .bin(b_n[1]), .cin(c1),
        .cout(c2), .sout(sum[1])
    );

    assign cbout = c2;

endmodule //addsub
```

File Inclusion

Listing 6: Two Bit Adder/Aubtractor Test Benches Verilog code

```
‘timescale 1ns / 1ps
//
   ////////////////////////////////////////////////////////////////////////////////

// Company: ELC 2137
// Engineer: Yiting Wang
// Create Date: 09/30/2020
//
   ////////////////////////////////////////////////////////////////////////////////
```

```
module addsub_test();
    reg [1:0] a_t, b_t;
    reg mode_t;
    wire cbout_t;
    wire [1:0] sum_t;

    addsub dut(
        .a(a_t), .b(b_t), .mode(mode_t),
        .cbout(cbout_t), .sum(sum_t)
    );

    initial begin
        a_t[1] = 0; a_t[0] = 0; b_t[1] = 0; b_t[0] = 0; mode_t = 0; #10;
        a_t[1] = 0; a_t[0] = 0; b_t[1] = 0; b_t[0] = 1; mode_t = 0; #10;
        a_t[1] = 0; a_t[0] = 0; b_t[1] = 1; b_t[0] = 0; mode_t = 0; #10;
        a_t[1] = 0; a_t[0] = 0; b_t[1] = 1; b_t[0] = 1; mode_t = 0; #10;
        a_t[1] = 0; a_t[0] = 1; b_t[1] = 0; b_t[0] = 1; mode_t = 0; #10;
        a_t[1] = 1; a_t[0] = 0; b_t[1] = 0; b_t[0] = 1; mode_t = 0; #10;
        a_t[1] = 1; a_t[0] = 0; b_t[1] = 0; b_t[0] = 0; mode_t = 0; #10;

        a_t[1] = 0; a_t[0] = 0; b_t[1] = 0; b_t[0] = 0; mode_t = 1; #10;
        a_t[1] = 0; a_t[0] = 0; b_t[1] = 0; b_t[0] = 1; mode_t = 1; #10;
        a_t[1] = 0; a_t[0] = 0; b_t[1] = 1; b_t[0] = 0; mode_t = 1; #10;
        a_t[1] = 0; a_t[0] = 0; b_t[1] = 1; b_t[0] = 1; mode_t = 1; #10;
        a_t[1] = 0; a_t[0] = 1; b_t[1] = 0; b_t[0] = 1; mode_t = 1; #10;
        a_t[1] = 1; a_t[0] = 0; b_t[1] = 0; b_t[0] = 1; mode_t = 1; #10;
        a_t[1] = 1; a_t[0] = 0; b_t[1] = 0; b_t[0] = 0; mode_t = 1; #10;

        $finish;
    end

endmodule
```