Vincent Han

Lab 1 Notebook

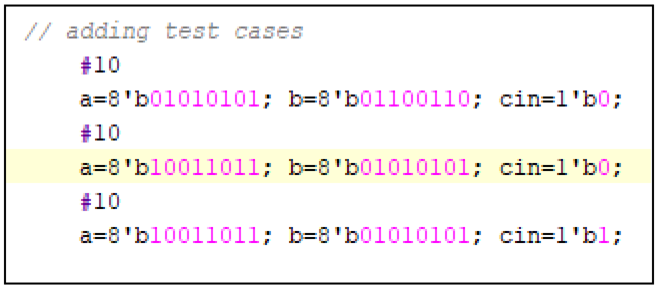
June 10, 2023

Brief Description:

This lab was an introduction to using Vivado. We had created a full adder, and then created an 8 bit adder. We then tested the functionalities of them by using a testbench file. All of the .v files were premade, and the only adjustments that were made to the testbench was to add more test cases.

Discussion of Results:

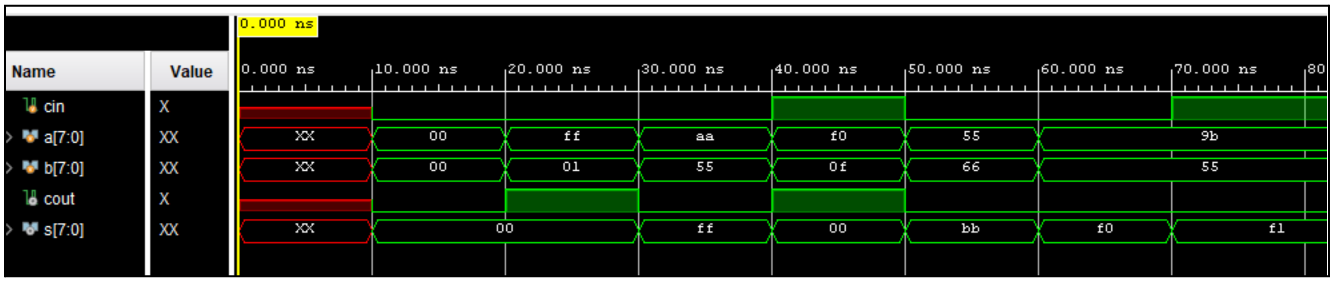
a. I had used a few different test cases. I had just tried 2 more additions, and then one that used subtraction. I just chose the numbers randomly, but then I wanted to make sure that the subtraction worked as well.

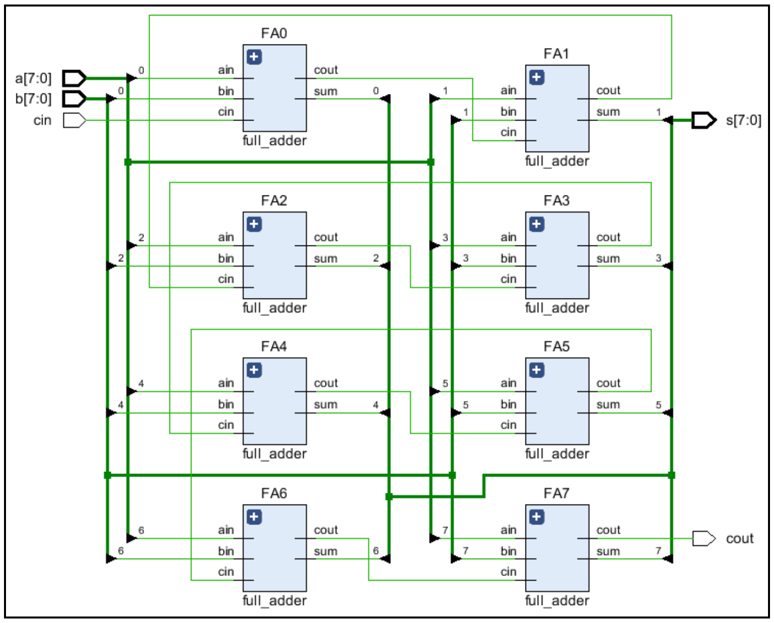


```
// adding test cases
    #10
    a=8'b01010101; b=8'b01100110; cin=1'b0;
    #10
    a=8'b10011011; b=8'b01010101; cin=1'b0;
    #10
    a=8'b10011011; b=8'b01010101; cin=1'b1;
```

b. I did not have to modify the code that was given to me. The adder had worked correctly.

c. Reports from lab1

i. 

ii. 

iii. Utilization report is below

1. Slice Logic

Site Type	Used	Fixed	Available	Util%
Slice LUTs*	8	0	20800	0.04
LUT as Logic	8	0	20800	0.04
LUT as Memory	0	0	9600	0.00
Slice Registers	0	0	41600	0.00
Register as Flip Flop	0	0	41600	0.00
Register as Latch	0	0	41600	0.00
F7 Muxes	0	0	16300	0.00
F8 Muxes	0	0	8150	0.00

* Warning! The Final LUT count, after physical optimizations and full implementation, is typically lower. Run opt_design after synthesis, if not already completed, for a more realistic count.

```
1.1 Summary of Registers by Type
--------------------------------

+-------+--------------+-------------+--------------+
| Total | Clock Enable | Synchronous | Asynchronous |
+-------+--------------+-------------+--------------+
| 0     |            _ |           - |            - |
| 0     |            _ |           - |          Set |
| 0     |            _ |           - |        Reset |
| 0     |            _ |         Set |            - |
| 0     |            _ |       Reset |            - |
| 0     |          Yes |           - |            - |
| 0     |          Yes |           - |          Set |
| 0     |          Yes |           - |        Reset |
| 0     |          Yes |         Set |            - |
| 0     |          Yes |       Reset |            - |
+-------+--------------+-------------+--------------+


2. Memory
---------

+----------------+------+-------+-----------+-------+
|    Site Type   | Used | Fixed | Available | Util% |
+----------------+------+-------+-----------+-------+
| Block RAM Tile |    0 |     0 |        50 |  0.00 |
|   RAMB36/FIFO* |    0 |     0 |        50 |  0.00 |
|   RAMB18       |    0 |     0 |       100 |  0.00 |
+----------------+------+-------+-----------+-------+
* Note: Each Block RAM Tile only has one FIFO logic available and therefore can
accommodate only one FIFO36E1 or one FIFO18E1. However, if a FIFO18E1 occupies a Block
RAM Tile, that tile can still accommodate a RAMB18E1


3. DSP
------

+-----------+------+-------+-----------+-------+
| Site Type | Used | Fixed | Available | Util% |
+-----------+------+-------+-----------+-------+
| DSPs      |    0 |     0 |        90 |  0.00 |
+-----------+------+-------+-----------+-------+


4. IO and GT Specific
---------------------

+-----------------------------+------+-------+-----------+-------+
|          Site Type          | Used | Fixed | Available | Util% |
+-----------------------------+------+-------+-----------+-------+
| Bonded IOB                  |   26 |     0 |       106 | 24.53 |
| Bonded IPADs                |    0 |     0 |        10 |  0.00 |
| Bonded OPADs                |    0 |     0 |         4 |  0.00 |
| PHY_CONTROL                 |    0 |     0 |         5 |  0.00 |
| PHASER_REF                  |    0 |     0 |         5 |  0.00 |
| OUT_FIFO                    |    0 |     0 |        20 |  0.00 |
| IN_FIFO                     |    0 |     0 |        20 |  0.00 |
| IDELAYCTRL                  |    0 |     0 |         5 |  0.00 |
| IBUFDS                      |    0 |     0 |       104 |  0.00 |
| GTPE2_CHANNEL               |    0 |     0 |         2 |  0.00 |
| PHASER_OUT/PHASER_OUT_PHY   |    0 |     0 |        20 |  0.00 |
| PHASER_IN/PHASER_IN_PHY     |    0 |     0 |        20 |  0.00 |
| IDELAYE2/IDELAYE2_FINEDELAY |    0 |     0 |       250 |  0.00 |
| IBUFDS_GTE2                 |    0 |     0 |         2 |  0.00 |
| ILOGIC                      |    0 |     0 |       106 |  0.00 |
| OLOGIC                      |    0 |     0 |       106 |  0.00 |
+-----------------------------+------+-------+-----------+-------+


5. Clocking
-----------

+------------+------+-------+-----------+-------+
|  Site Type | Used | Fixed | Available | Util% |
+------------+------+-------+-----------+-------+
| BUFGCTRL   |    0 |     0 |        32 |  0.00 |
| BUFIO      |    0 |     0 |        20 |  0.00 |
| MMCME2_ADV |    0 |     0 |         5 |  0.00 |
| PLLE2_ADV  |    0 |     0 |         5 |  0.00 |
| BUFMRCE    |    0 |     0 |        10 |  0.00 |
| BUFHCE     |    0 |     0 |        72 |  0.00 |
| BUFR       |    0 |     0 |        20 |  0.00 |
+------------+------+-------+-----------+-------+


6. Specific Feature
-------------------

+-------------+------+-------+-----------+-------+
|  Site Type  | Used | Fixed | Available | Util% |
+-------------+------+-------+-----------+-------+
| BSCANE2     |    0 |     0 |         4 |  0.00 |
| CAPTUREE2   |    0 |     0 |         1 |  0.00 |
| DNA_PORT    |    0 |     0 |         1 |  0.00 |
| EFUSE_USR   |    0 |     0 |         1 |  0.00 |
| FRAME_ECCE2 |    0 |     0 |         1 |  0.00 |
| ICAPE2      |    0 |     0 |         2 |  0.00 |
| PCIE_2_1    |    0 |     0 |         1 |  0.00 |
| STARTUPE2   |    0 |     0 |         1 |  0.00 |
| XADC        |    0 |     0 |         1 |  0.00 |
+-------------+------+-------+-----------+-------+


7. Primitives
-------------


+----------+------+---------------------+
| Ref Name | Used | Functional Category |
+----------+------+---------------------+
| IBUF     |   17 |                  IO |
| OBUF     |    9 |                  IO |
| LUT5     |    8 |                 LUT |
| LUT3     |    4 |                 LUT |
+----------+------+---------------------+
```

Key Steps:

Summary:

a. Yes, since the adder has worked as intended, the goals were met.
b. No difficulties had been encountered, besides some menu hassle.
c. No changes would be made.

Source Code:

Full_adder.v

```
module full_adder(
    output cout,
    output sum,
    input ain,
    input bin,
    input cin
    );
    assign sum = ain^bin^cin;
    assign cout= (ain&bin)|(ain&cin)|(bin&cin);

endmodule
```

Adder8.v

```
module adder8(
    output cout,
    output [7:0] s,
    input [7:0] a,
    input [7:0] b,
    input cin
    );
    wire [7:1] carry;
    full_adder FA0(carry[1],s[0],a[0],b[0],cin);
    full_adder FA1(carry[2],s[1],a[1],b[1],carry[1]);
    full_adder FA2(carry[3],s[2],a[2],b[2],carry[2]);
    full_adder FA3(carry[4],s[3],a[3],b[3],carry[3]);
    full_adder FA4(carry[5],s[4],a[4],b[4],carry[4]);
    full_adder FA5(carry[6],s[5],a[5],b[5],carry[5]);
    full_adder FA6(carry[7],s[6],a[6],b[6],carry[6]);
    full_adder FA7(cout,s[7],a[7],b[7],carry[7]);

Endmodule
```

Adder8_tb.v

```
`timescale 1ns / 1ps
module adder8_tb( );
    reg cin;
    reg [7:0] a,b;
    wire cout;
    wire [7:0] s;

    // instantiate the unit under test uut
    adder8 uut(cout,s,a,b,cin);

    // stimulus (inputs)
    initial begin
        #10 // wait for global reset
        a=8'b00000000; b=8'b00000000; cin=1'b0;
        #10
        a=8'b11111111; b=8'b00000001; cin=1'b0;
        #10
        a=8'b10101010; b=8'b01010101; cin=1'b0;
        #10
        a=8'b11110000; b=8'b00001111; cin=1'b1;

    // adding test cases
        #10
        a=8'b01010101; b=8'b01100110; cin=1'b0;
        #10
        a=8'b10011011; b=8'b01010101; cin=1'b0;
        #10
        a=8'b10011011; b=8'b01010101; cin=1'b1;

    end

    // results (outputs)
    initial begin
```

```
        #5 $display("a = %b, b = %b, cin = %b",a,b,cin); $display("s = %b, cout = %b",s,cout);
        #10 $display("a = %b, b = %b, cin = %b",a,b,cin); $display("s = %b, cout = %b",s,cout);
        #10 $display("a = %b, b = %b, cin = %b",a,b,cin); $display("s = %b, cout = %b",s,cout);
        $display("End Simulation");
    end

endmodule
```

Relevant Output Files: