CE 6303.001

HW1 Report

**Luke Allen** NetID: laa170930

**Yu Feng** NetID: yxf160330

09/01/2021

# Problem Description:

The problem that was given was to design a two-bit full adder in both structural and behavioral models, verify the functionality using a testbench, simulate in ModelSim and present resulting output waveforms.

# Adder Description:

1. Structural

Since 2-bit adder can be viewed as chaining two 1-bit full adders, each processing one bit of data from each operand at a time, a 1-bit adder is described first.

To get the sum of two operand a and b, one needs to first XOR a and b, then XOR the product with carry in. With the XOR product of a and b stored as temporary value t1, ANDing t1 with the carry in value produces temporary value t2. Lastly, by ANDing values a and b (stored in t3), and ORing t2 with t3 produces the expected carry out value. Here is the schematic described above:

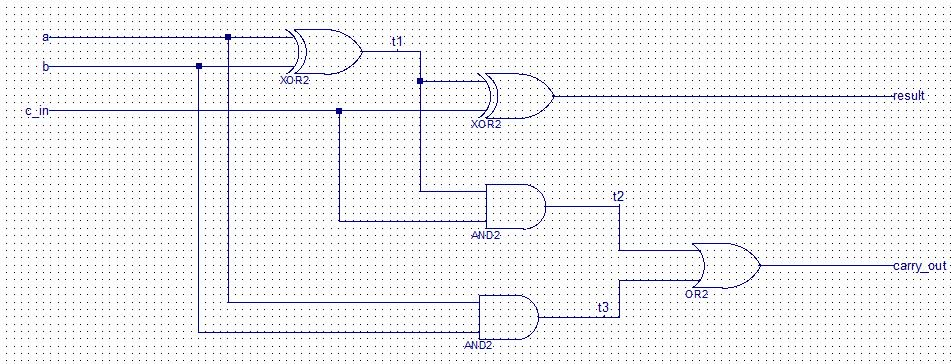


Figure 1: Schematic reference of 2-bit adder in Structural Verilog

Now, a 1-bit full adder is ready. To obtain a 2-bit adder, two 1-bit adders are connected in the following way:

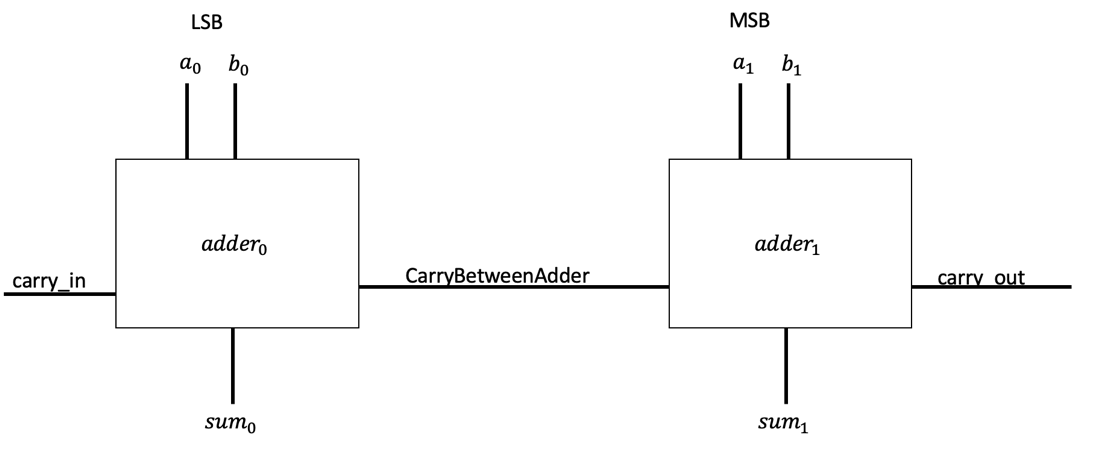


Figure 2: Connection between 2 adders

The least significant bits (LSB) of operands a and b, a0 and b0, are connected to adder0 along with carry in. The carry out for adder 0 is sent to the carry in of adder1, forming the carryBetweenAdder signal. The most significant bits (MSB) of operands a and b, a1 and b1, are connected to adder1. Thus, the result of the 2-bit adder can be found at sum1 from adder1, the most significant bit of the result; sum0, the least significant bit of result from adder0; and carry out from adder1.

1. Behavioral

The behavioral model was based on the same design as the structural model, as it consists of two one-bit full adder modules that are instantiated to create a two-bit full adder.

Rather than defining each individual gate like in the structural model, the behavioral model simply uses Boolean logic statements to achieve the same design. There are other ways to achieve a behavioral model such as using switch statements or if-else statements, however this was the simplest way to design a behavioral model of a two-bit full adder, as the logic was fairly simple. The statements for the one-bit adders are as follows:

Two of these full adders are instantiated in the two-bit adder module and connected as shown in *figure 2*, creating a behavioral model of a two-bit adder.

# Results:

Truth Table for adder:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | | | Output, Carry\_in=0 | | | Output, Carry\_in=1 | | |
| A0 | A1 | B0 | B1 | Sum0 | Sum1 | Carry\_out | Sum0 | Sum1 | Carry\_out |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
|  |  |  |  |  |  |  |  |  |  |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
|  |  |  |  |  |  |  |  |  |  |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |

1. Structural

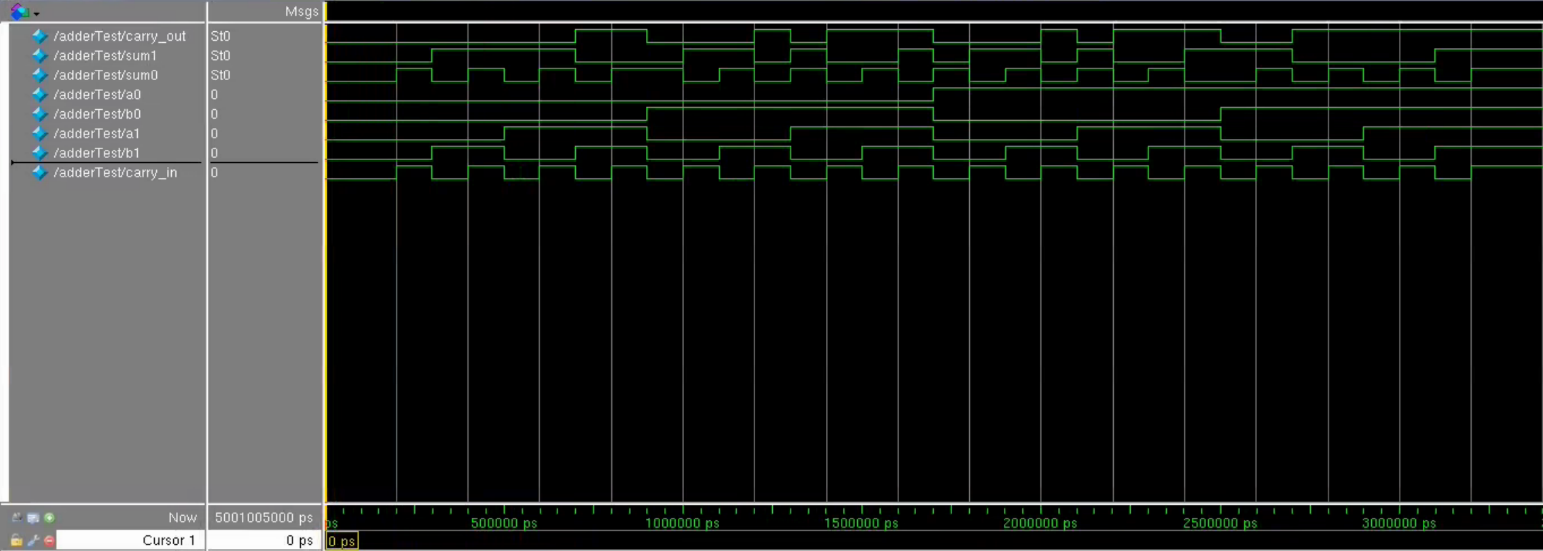


Figure 3:Waveform of Structural Verilog simulation in ModelSim

1. Behavioral

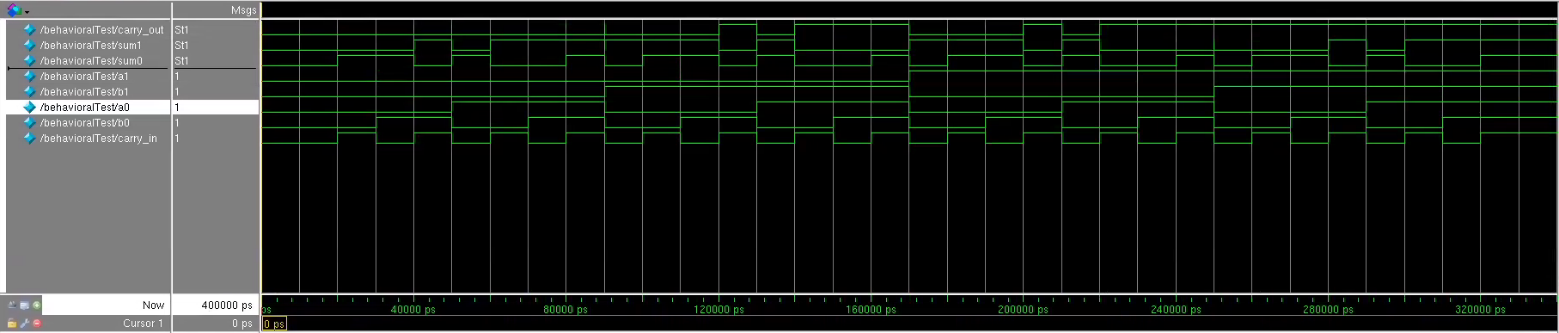


Figure 4:Waveform of Behavioral Verilog simulation in ModelSim