Ryan Sullivan

U73687595

EC413 Lab4

**Modules:**

**FA\_str:** A 1-bit full adder in structural Verilog without delays.

**FA\_str\_D:** A 1-bit full adder in structural Verilog with gate delays.

**FA\_4bit:** A 4-bit full adder using the FA\_str module as a basis for a 4-part ripple carry adder without delays.

**FA\_4bit\_D:** A 4-bit full adder using the FA\_str\_D module as a basis for a 4-part ripple carry adder with gate delays.

**FA\_32bit:** A 32-bit full adder using the FA\_4bit as a basis for an 8-part ripple carry adder without gate delays.

**FA\_32bit\_D:** A 32-bit full adder using the FA\_4bit\_D as a basis for an 8-part ripple carry adder with gate delays.

**FA\_64bit\_RCA:** A 64-bit full ripple carry adder using the FA\_32bit module as a basis for a 2 module design without gate delay.

**FA\_64bit\_RCA\_D:** A 64-bit full ripple carry adder using the FA\_32bit\_D module as a basis for a 2 module design with gate delay.

**FA\_64bit\_CSA:** A 64-bit full carry select adder using the FA\_32\_bit module as a basis for this 3 module design with a continuously assigned output, the second half of sum and c\_out, determined by the c\_out output of the first module. The other two modules are there to precompute the possible second half of sum and c\_out, where c\_in is equal to 1 or 0 respectively.

**Verify\_64bit:** A 64-bit full adder done in behavioral Verilog, used as a means of testing the outputs of the FA\_64bit\_RCA, FA\_64bit\_RCA\_D, FA\_64bit\_CSA, FA\_64bit\_CSA\_D modules.

**FA\_64bit\_tb:** The testbench that compares the FA\_64bit\_RCA, FA\_64bit\_RCA\_D, FA\_64bit\_CSA, FA\_64bit\_CSA\_D, and Verify\_64bit modules.

**Timing Diagrams:**

While all the adders produced the correct outputs under the assumptions (two 32-bit unsigned binary numbers) the gate delay modules lagged behind with a consistent 4 ns delay, during which their output would either display the previous sum, given the inputs, or an intermediate sum wherein only a few of the inputs had been fully processed.

**Example:**

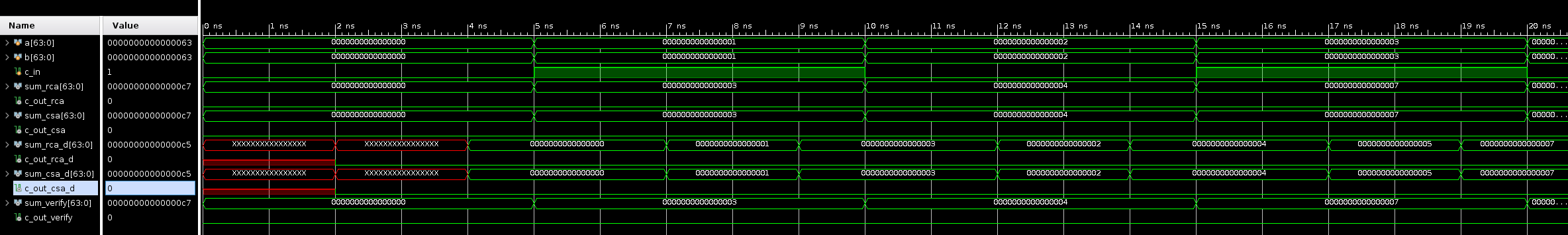


Figure 1: The first 3 rows are inputs a,b,c\_in and each subsequent pair of rows corresponds to the outputs of the modules, RCA, CSA, RCA with delay, CSA with delay, and Verify

The figure above shows the clear timing delay for the sums of both modules with delays included, as their sum outputs lag behind a full 4 ns, their c\_out outputs only 2 ns

**Delays:**

The delays for the two gate delayed adders should be, 128 for the 64-bit ripple carry adder and 64 gate delays for the 64-bit carry select adder. This is due to the fact that the gate delayed full adder has a critical path with 2 gates and there are 4 being used in the 4-bit adder in a ripple carry setup, there are 8 of them being used in the 32-bit adder in ripple carry setup and there are two being used in ripple carry implementation in the 64-bit RCA but three being used in CSA setup in the 64-bit CSA, thus halving the expected gate delay. However, the delay between the two is exactly the same, this may be due to the program not fully implementing the two extra 32-bit adders in the 64-bit CSA until the carry select is generated by the first 32-bit adder.