Abhishek Gaur
U42265597
abhigaur@bu.edu

EC551 Lab 1

Anubhav Gupta
U80332699
anubhavg@bu.edu

1-Bit Full Adder

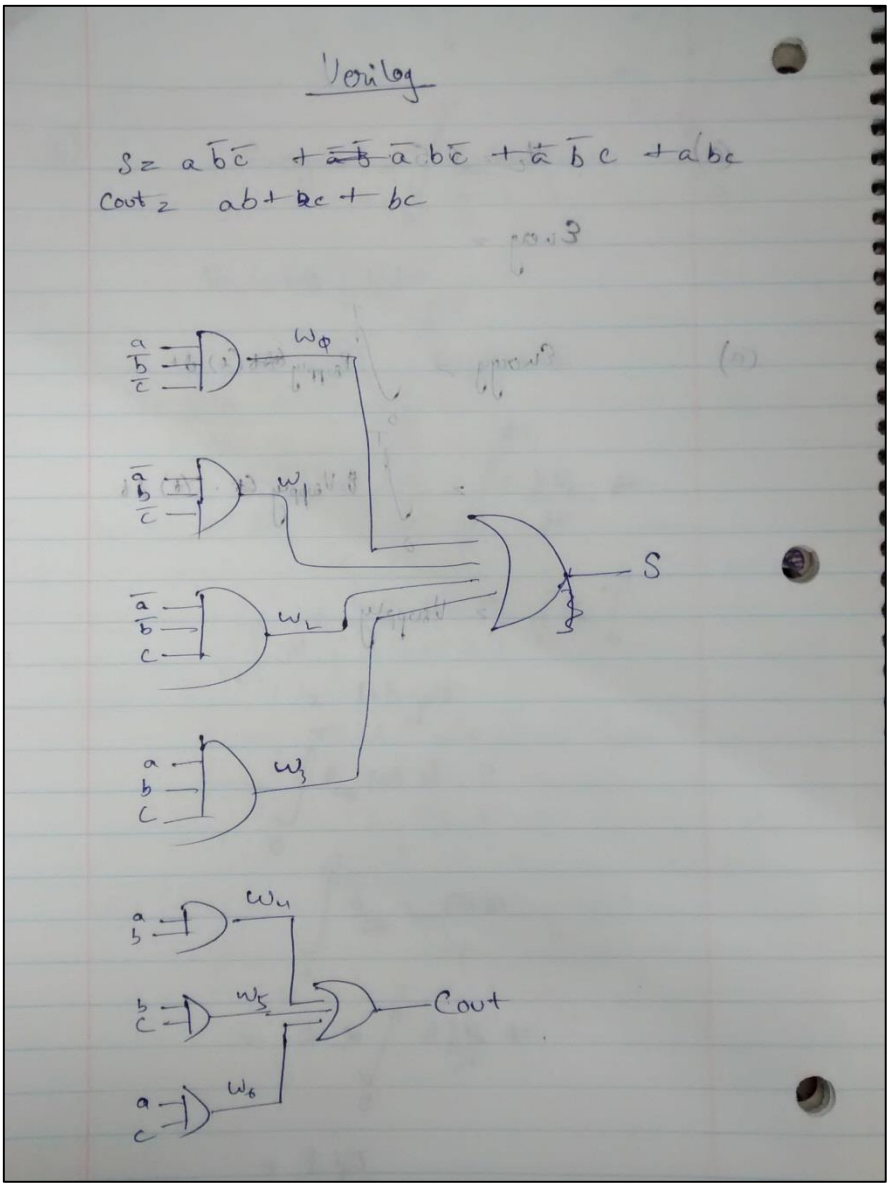
File Name: *Full_Adder.v*

Testbench Name: *Full_Adder_Testbench*

Waveform: *Full_Adder_Waveform*

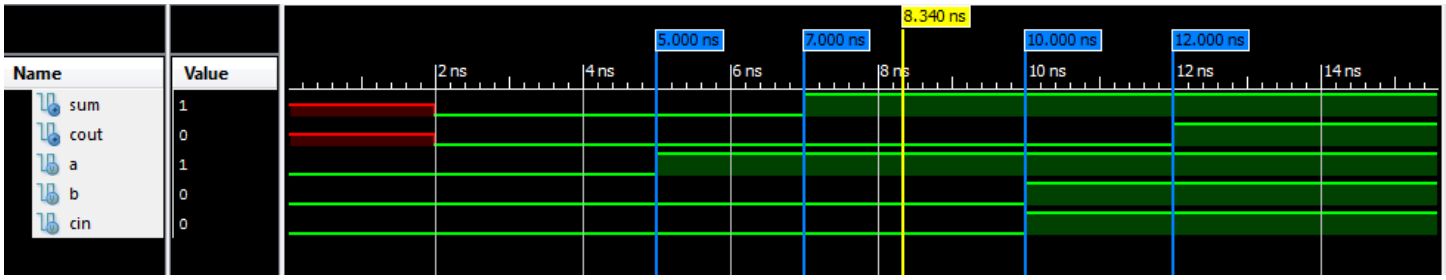
Description:

Firstly, we made a simple 1-bit full adder using just the AND, OR and NOT gates. We get the Sum and the carry as the output in this adder. (Refer the image below)



Propagation delay of a 1-bit full adder is 2ns since the delay of AND and OR gate is 1ns each.

Output Waveform from Verilog Code:



As we can see in the image above, the inputs ‘a’ and ‘b’ change at **5ns**, whereas the ouputs ‘sum’ and ‘cout’ attain their values at **7ns**, hence a **2ns** delay as calculated.

64-Bit Ripple Carry Adder

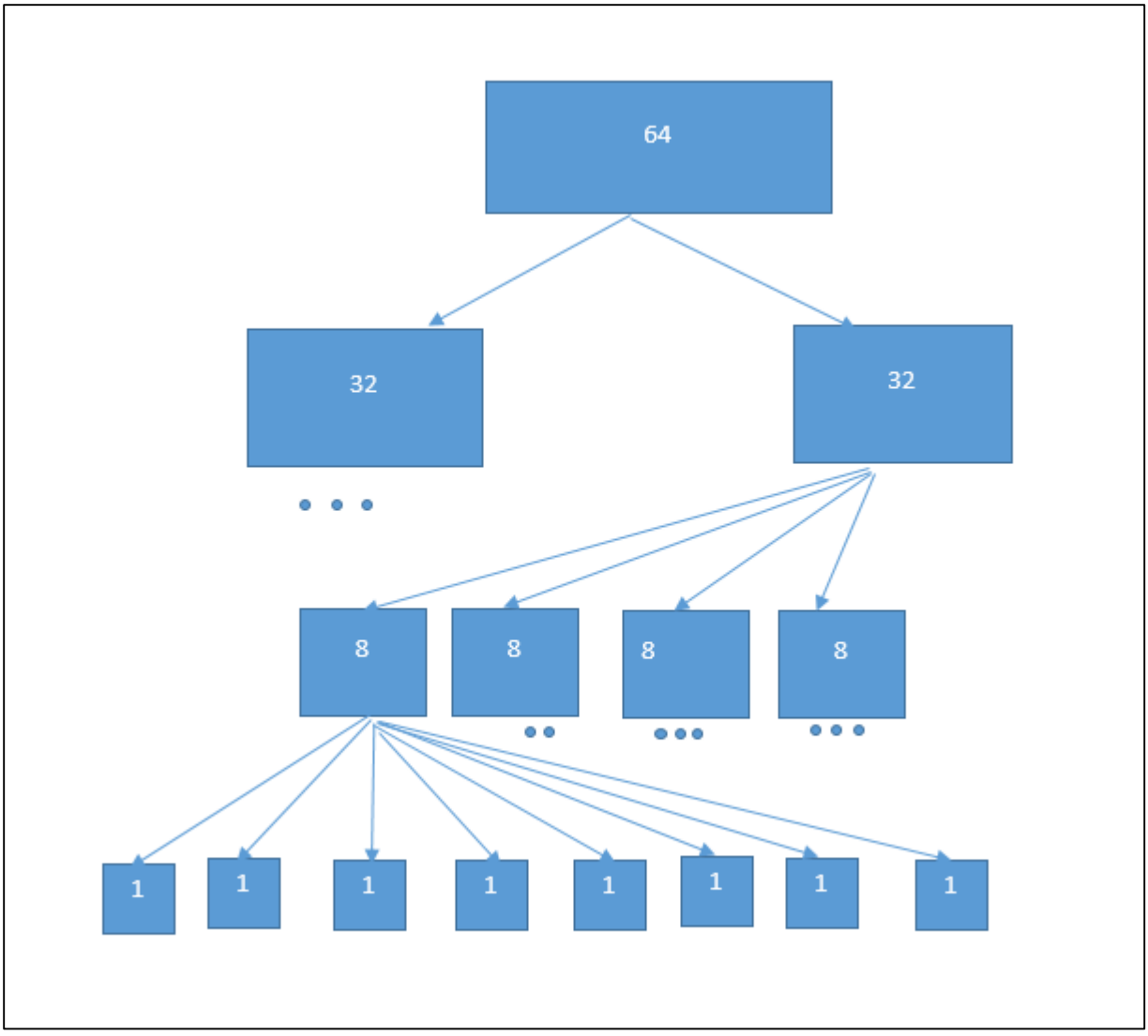
File Name: *FA_64Bit.v*

Testbench Name: *FA_64Bit_Testbench*

Waveform: *FA_64Bit_Waveform*

Description:

We used the hierarchical model to design the 64-bit Ripple Carry adder. We made a use of 32-bit, 8-bit and 1-bit adders in designing the hierarchy of the 64-bit adder. (Refer image below)



The propagation delay of the 64-bit ripple carry adder is calculated as follows:

Propagation Delay of 1-bit Full Adder = 2ns (1ns each for AND and OR gate)

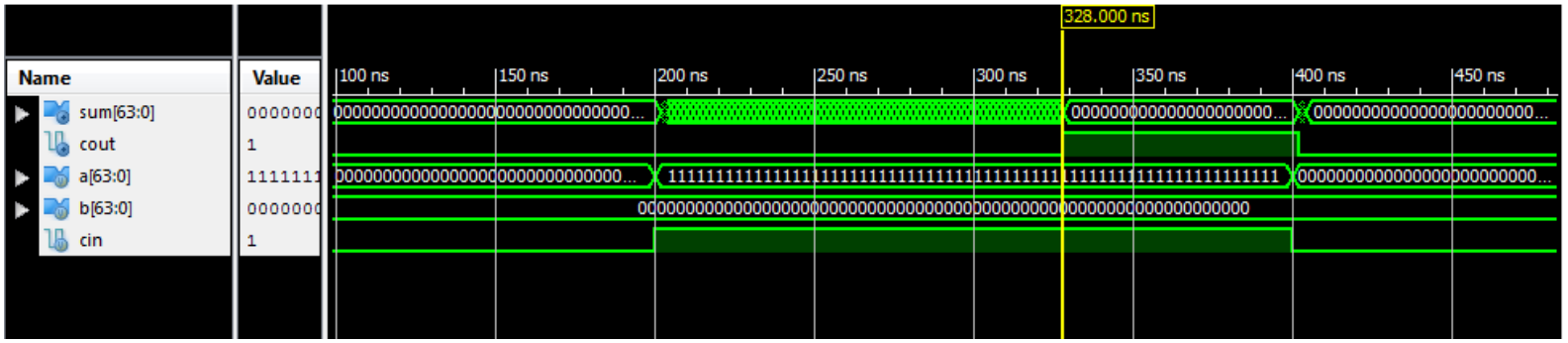
Propagation Delay of 8-bit Full Adder = 16ns

Propagation Delay of 32-bit Full Adder = 64ns

Propagation Delay of 64-bit Full Adder = 128ns

Hence, the propagation delay of the 64-bit Ripple Carry Adder: ***128ns***. The reason being that each full adder must wait for the carry bit to be calculated from the previous full adder. So, the propagation delay is **2n**, where n is the number of bits.

Output Waveform from Verilog Code:



As we can see in the image above, the inputs ‘a’ and ‘b’ change at **200ns**, whereas the ouputs ‘sum’ and ‘cout’ attain their values at **328ns**, hence a **128ns** delay as calculated.

64-Bit Carry Select Adder

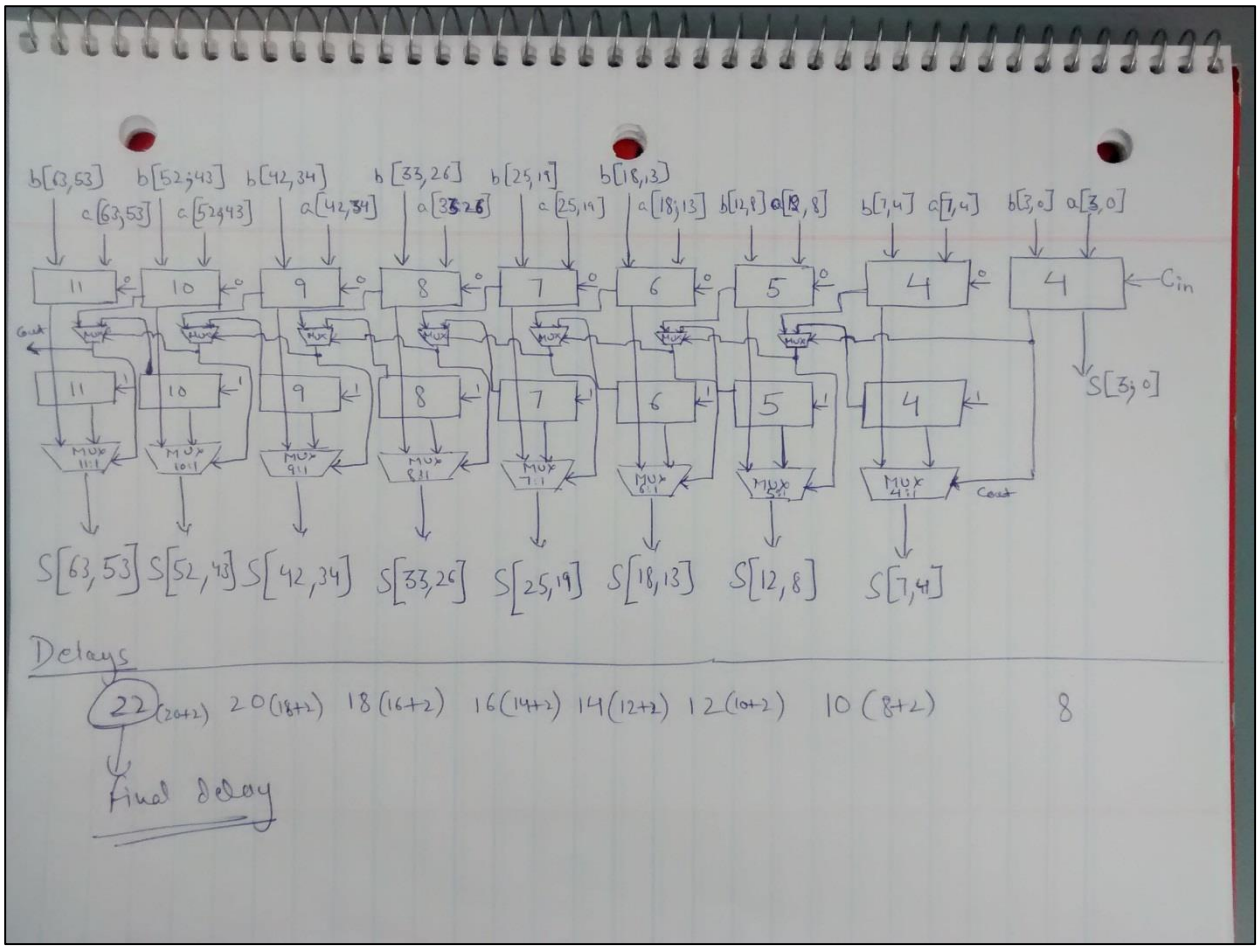
File Name: *Carry_Select_Adder.v*

Testbench Name: *Carry_Select_Adder _Testbench*

Waveform: *Carry_Select_Adder _Waveform*

Description:

We used the hierarchical model to design the 64-bit Carry-Select adder. The design of the circuit is as shown in the image below.

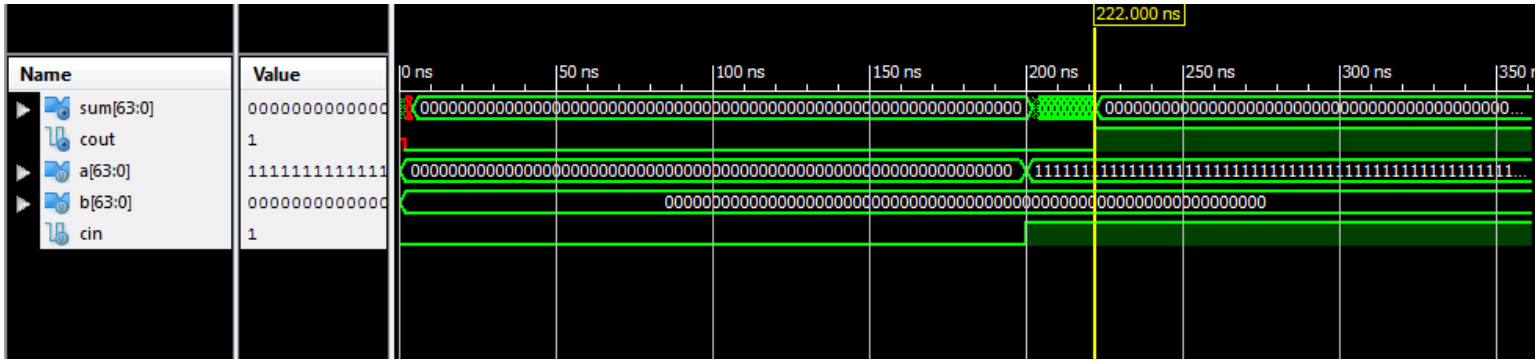


As we can see, use of ripple carry adders and MUXes of variable sizes minimized the delay to **22ns** only.

The only propagation delay in this adder is the 2ns delay that is caused by the AND and OR gate (equivalent to a MUX) between two adder stages. These MUXes are used to select the carry that needs to propagate further.

The other MUXes used are to select the correct sum of the succeeding adders and their delay is not considered while calculating the overall delay of the carry-select adder.

Output Waveform from Verilog Code:



As we can see in the image above, the inputs ‘a’ and ‘b’ change at **200ns**, whereas the ouputs ‘sum’ and ‘cout’ attain their values at **222ns**, hence a **22ns** delay as calculated.