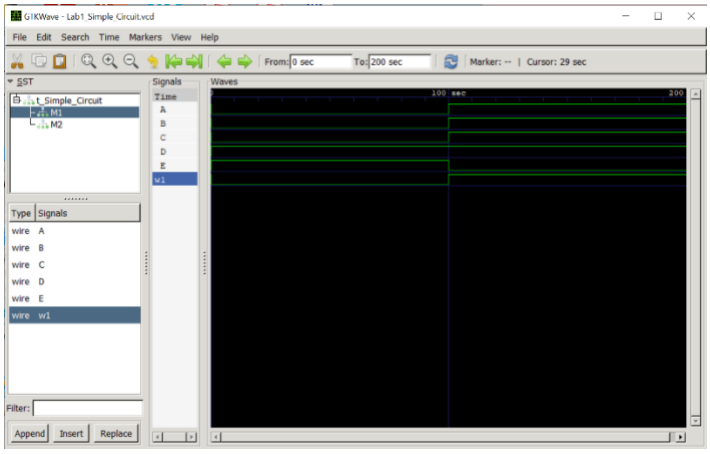
109550182 莊婕妤

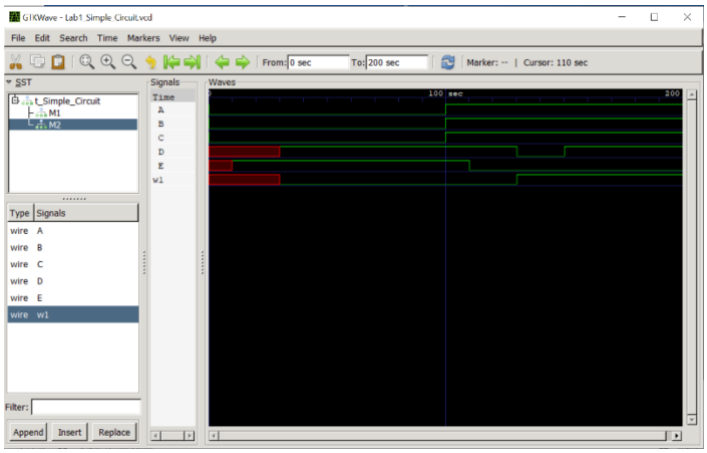
# Digital Circuit Design Lab1 Report

**1.** A(a) The waveform of the stimulation result

Simple_Circuit.v



Simple_Circuit_prop_delay.v

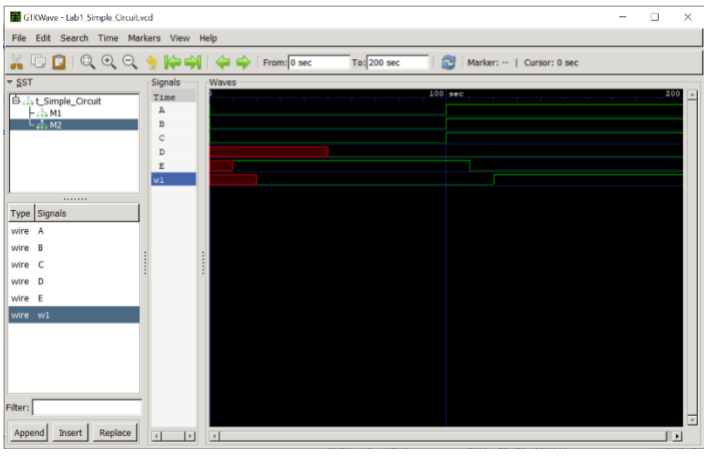


Difference between the waveforms:

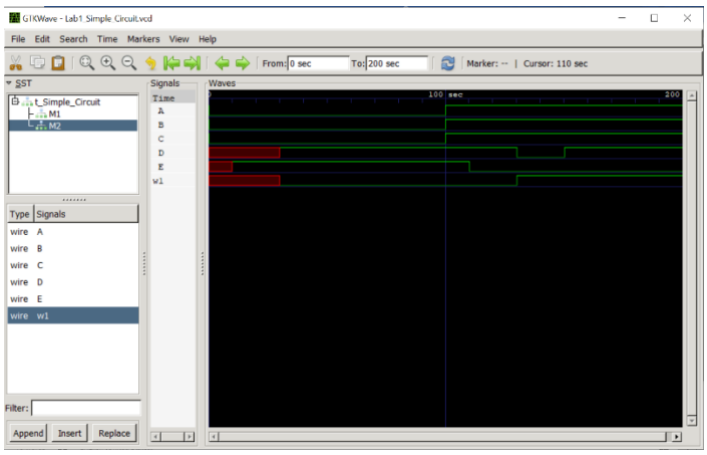
The major difference is that the circuit in Simple_Circuit_prop_delay.v has propagation delays, which means that some gates require more time to change values based on the input changes. As you can see from the two waveforms, the output D, E and the wire w1 are all affected by the propagation delay. To elaborate more by detail, we can take the output E as an example. The propagation delay of a NOT gate is 10 time units, therefore the output E will change its output value 10 time units after input C changes its value.

**2.** A(b) The waveform of the stimulation result

Simple_Circuit_prop_delay.v (After Swapping)



Simple_Circuit_prop_delay.v (Before Swapping)

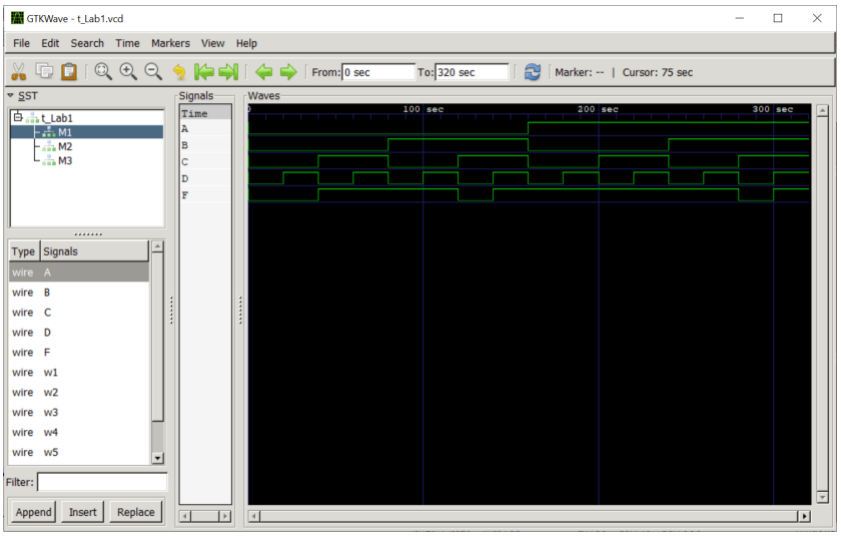


Difference between (a) and (b):

There are two differences between (a) and (b), the output D and wire w1.
For w1, the propagation delay changed to 20 time units after switching the arguments. On the other hand, the output D's propagation delay has increased to 50 time units since the OR gate changed to an AND gate. When there is an OR gate, it can generate an output simply based on one of the paths if the first input item returns 1. Since E is 1 at the beginning, D has a delay of 20 + 10 = 30 time units. Different from the OR gate, the AND gate has to know the two input values in order to generate an output. Therefore, the output D's propagation delay is 30 + 20 (Since 20 > 10) = 50 time units.

**3.** B(d) The waveform of the stimulation result

Lab1_gatelevel.v




I changed the function to Sum-of-Minterms form to detect its accuracy and found out the results are identical to the waveform. Therefore, the results are correct.

F (A, B, C, D)

= (A+B) (C'+D) + B' (A+C)

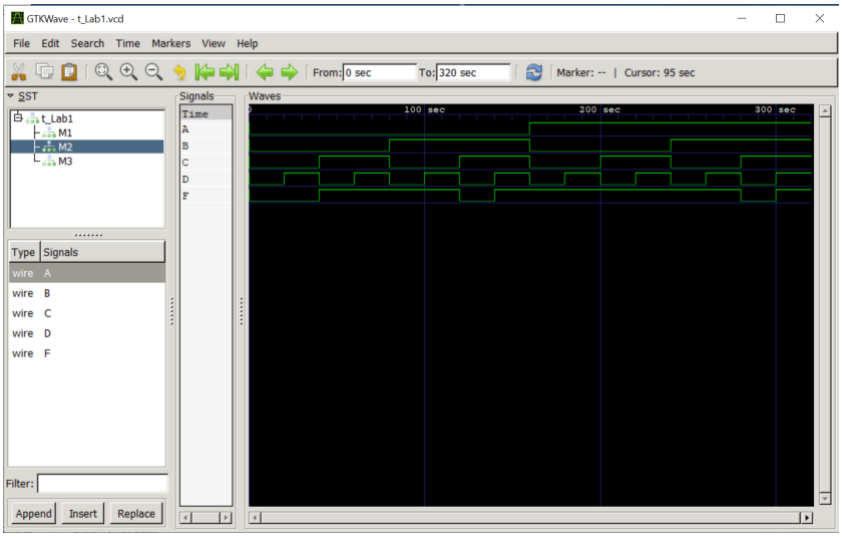
= AC' + AD + BC' + BD + AB' + B'C

= ABC' + AB'C' + ABD + AB'D
+ A'BC' + A'BD + AB'C + A'B'C

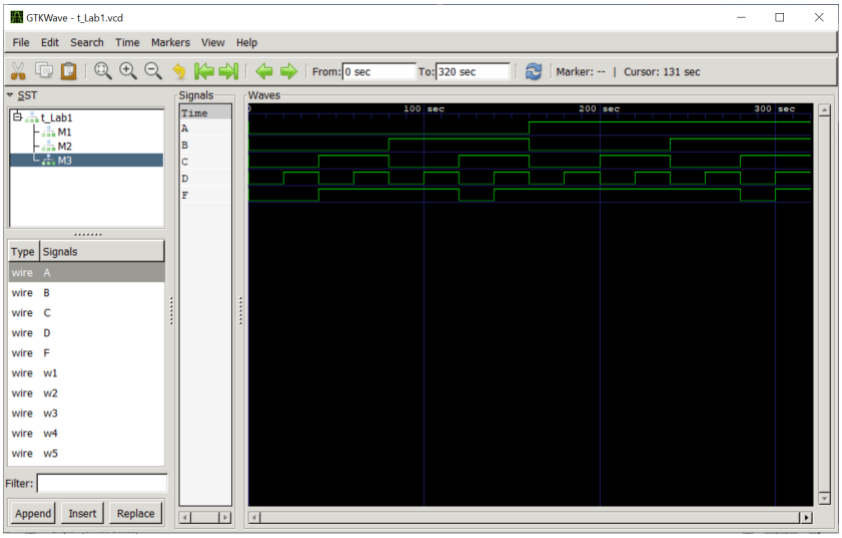
= ABC'D + ABC'D' + AB'C'D
+ AB'C'D' + ABCD + AB'CD
+ A'BC'D + A'BC'D' + A'BCD
+ AB'CD' + A'B'CD +A'B'CD'

= $\sum$ m (2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 15)

Lab1_dataflow.v




Lab1_gatelevel_UDP.v



**4.** The gate input count (GIC) of the current circuit is 14.

Further Simplification of the Boolean equation by Boolean Algebra:

F (A, B, C, D)

= (A + B) (C’ + D) + B’ (A + C)

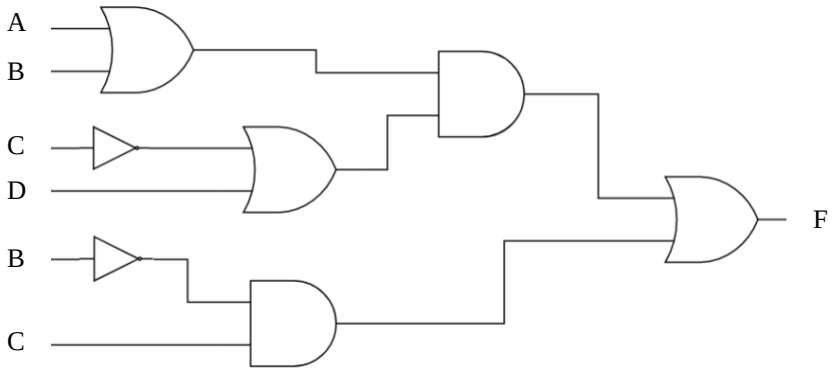
= AC’ + AD + BC’ + BD + AB’ + B’C  (By Theorem*, Consensus)

= AC’ + AD + BC’ + BD + B’C

= (A + B) (C’ + D) + B’C  (GIC = 12)

The gate input count is diminished to 12.
Hence, the circuit does not have the least gate input count.

Logic Diagram:




**5. Reflection**

I spent approximately one and a half days to finish this assignment.

At the beginning, I took a lot of time to figure out how the command window in my computer works and how it connects to Verilog. Besides, the most challenging thing for me is to get familiar with the Hardware Description Language (HDL) since it is a brand-new field to me. I tend to type the wrong syntaxes on my Notepad and got errors in the command window.

Nevertheless, I find this lab assignment quite interesting since I get to really run a digital circuit on my computer and observe how it changes instead of dealing with lots of Boolean equations in class.