Charles Habermehl
5/3/2018
CS 365

Final Project: Seesaw Logic Gate

The results of the simulation point towards a logic gate combination with three AND gates that feed their outputs into an OR gate for the final output. I used a logic gate generation program to make the diagram of the seesaw logic gate architecture in Figure 2 based on the truth table in Figure 3. To come to this conclusion, I derived a system of ODEs from the given seesaw gate logic reaction in Figure 1 of the project assignment.

To build the system of ODEs we had to pull out the individual reactions of each step of the seesaw reaction. We are then able to create a single vector array ODE that represents the seesaw reaction. The index values correspond to the initialized values in the y0 array. This allows us to pass the initialized values as an array into the function for integration, thus allowing us to control our initial concentration values as well as set our other concentrations to 0 for the reactions. The reaction rate constants were set to k = 5e-5 and k_{fast} = 5e-2. When this ODE is integrated, and we capture the output from the for plotting we see the graph in Figure 1.

Figure 1 in the assignment starts with three inputs that are reacted with a gate species and outputs an intermediate gate species as well as an intermediate output species. The intermediate gate species are used when determining the value of the outputs in the vectorized ODE. After the input reactions, the outputs of the input reactions can be used to create a threshold value which allows us to determine if the final output signal is true or false. We capture the signal using the threshold equation of the ODE. After capturing our threshold value, we can move to capturing a value for our signal which is done by combining the input from our initial inputs and a signal gate species which will give us an output signal value as well as an output to combine with our reporter reaction. The fuel reaction comes in next, using the fuel species and signal out species to produce a fuel output species and another intermediate output value. The fuel reaction helps to push the reaction to completion in combination with the inputs to create an output. In the last reaction we see the intermediate output generated by the signal gate in combination with a reported species that gives us our final output value and a little bit of waste.

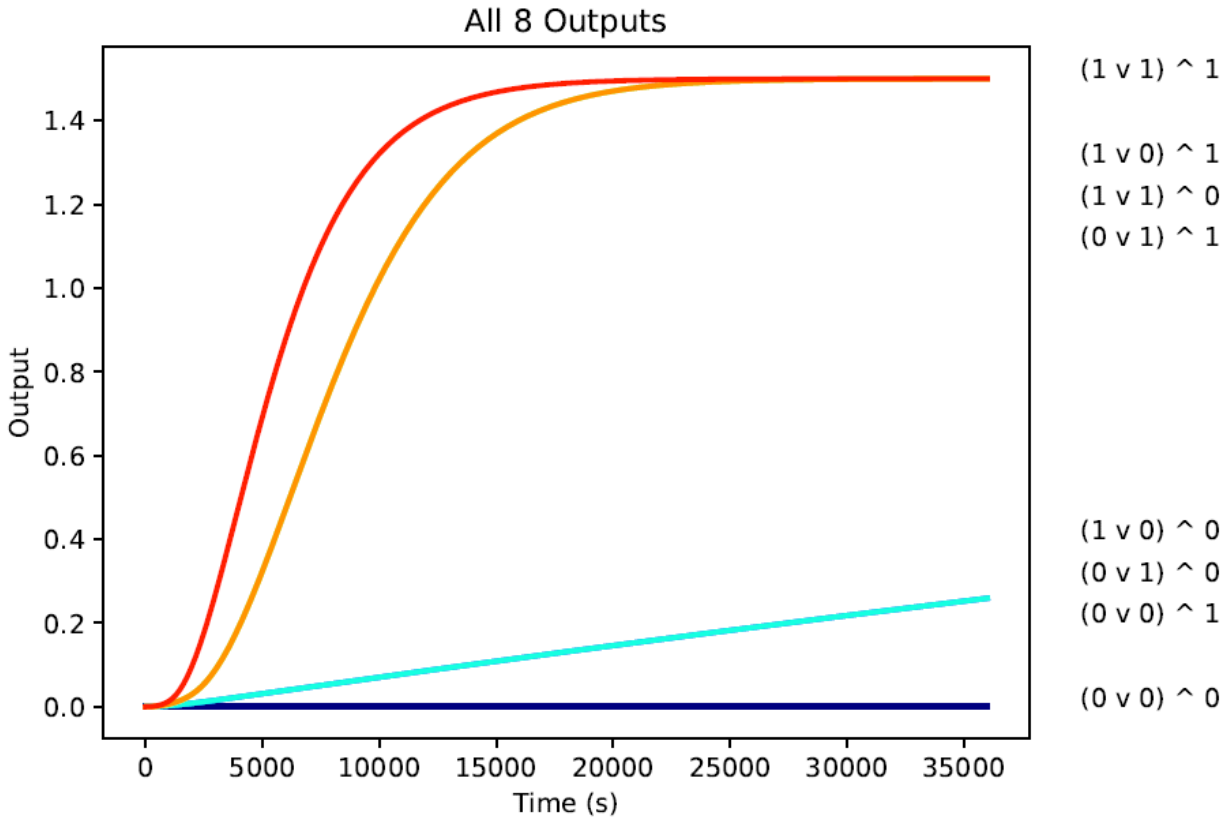
I was able to deduce the logic circuit through the graphed outputs of each individual output combination. Based upon the thresholding value to generate a true or false output I was able to determine which combinations of inputs created a true value and build a truth table (Figure 3) based upon the inputs of the reaction. Then with the truth table I was able to build a logic circuit that represented it. The design of the circuit is such that if not at least two of the inputs are true you will always get a false out. The result of this is three AND gates feeding into one OR gate. This means that it is impossible to get a true value out of the circuit unless you are feeding in at least two true inputs because the AND gates will only report falsely if otherwise.

In the process of determining the circuit I played around with the concentrations of the species and determined that increasing the concentrations for the initial values simply speeds up the reaction allowing the logic circuit to reach its threshold value sooner which means the output is reported sooner.

The features of the circuit design are enabled in the function through the three initial input gates which are then forced through the final signal gate and then reported through the final reaction. This corresponds directly with the logic circuit as it has three gates that give an output which is then input into a fourth gate to determine the output. The seesaw reaction needs a fueling reaction and a threshold reaction to make sure the reaction completes fully and that there is a value to determine a true or false output based on the inputs.

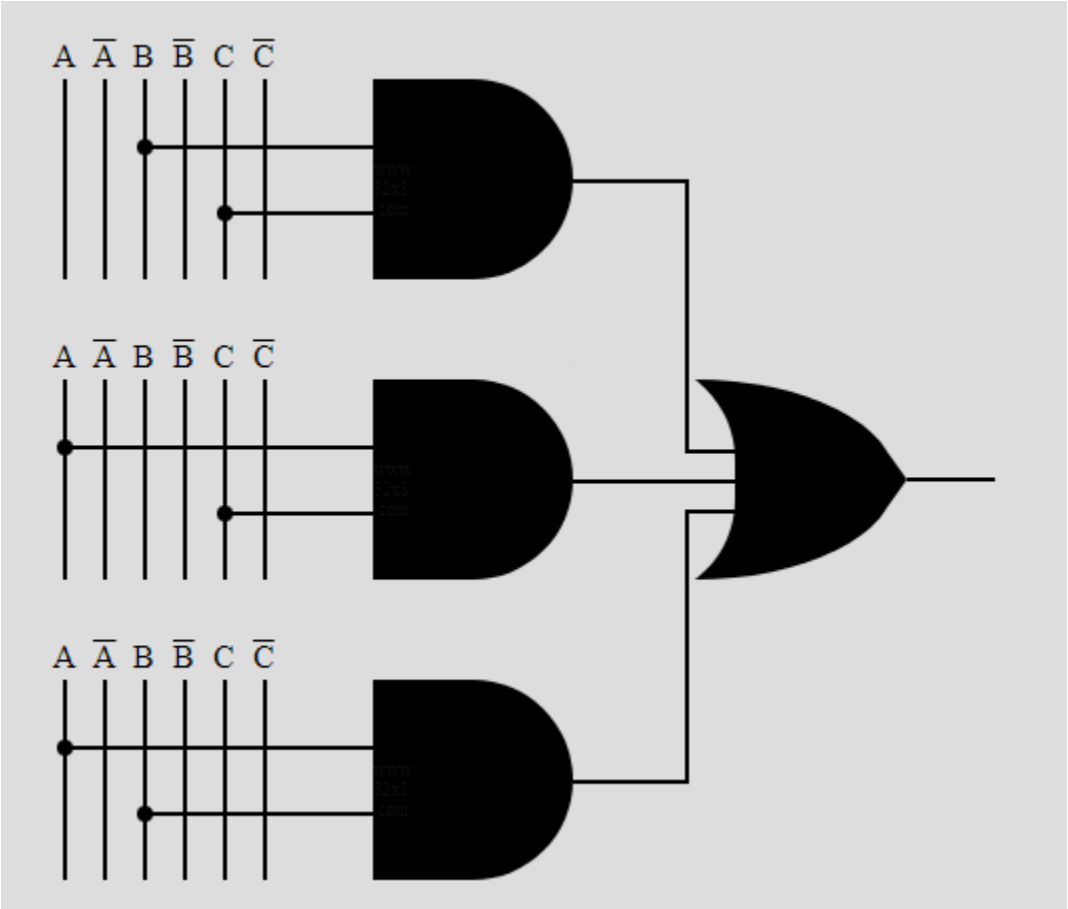
Charles Habermehl
5/3/2018
CS 365

Figure 1



Charles Habermehl
5/3/2018
CS 365

Figure 2



Charles Habermehl
5/3/2018
CS 365

Figure 3

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1