ECE120: Introduction to Computing Optional C Exercise for Week 7

Finding the Stable States of a Sequential Feedback Circuit

Overview:

In this exercise, you are asked to write a program that identifies all stable states of a sequential feedback circuit given a set of inputs to the circuit. The circuit consists of three cross-connected, two-input gates, and each student is given a randomly selected combination of gates (chosen from AND, OR, XOR, XNOR, NAND, and NOR). The topology and gates of your unique circuit can be found in "steady_states_spec.txt", a file in your ECE210 working directory. To check out the files, please use the "svn update" command in your working directory, and all the files for this exercise will appear in the folder named "OPTIONAL7".

The learning objectives of the exercise are as follows:

- Gain skill reading and analyzing simple C code
- Be able to formulate logic expressions on single bits in C

To help you with this exercise, you are given an example program that identifies all the stable states of an S-R latch. This code walks over all possible inputs, assumes a value for Q, calculates the resulting values around the feedback loop, and checks the logic circuit for stability by comparing the assumed value of Q with the value produced by the feedback loop calculation. This code prints each stable state found. For example, it prints "0 0 0 1" for R = S = 0, and Q = 0, $\bar{Q}$ = 1. Figure 1 is the circuit of the example code.

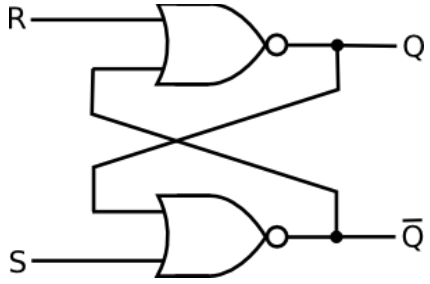
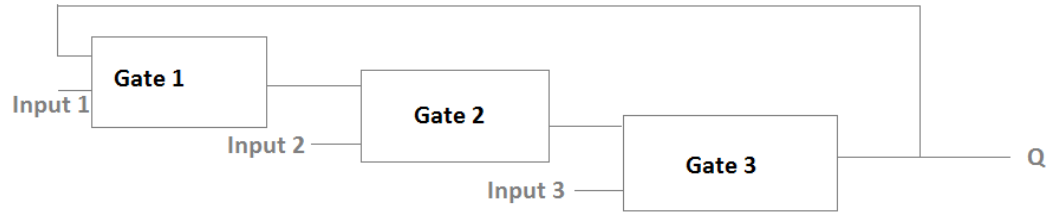


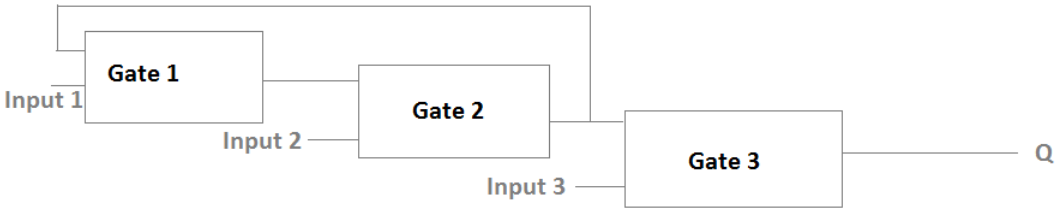
Figure1

You need to understand how the example code works, and write your own code to determine the stability of your circuit on a given state. Your circuit will have one of the following three topologies:

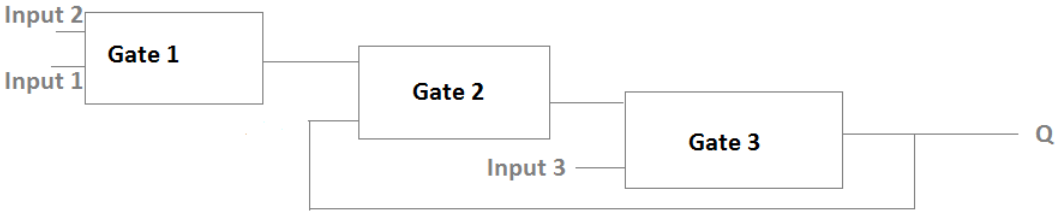
Topology 1:



Topology 2:



Topology 3:



Your circuit's topology and gates are specified in the file "steady_states_spec.txt".

Specification:

You should implement the C function main in the file "steady.c" to do the following:

1. Use printf to print a prompt: "Please enter values for input1, input2, input3.\n".
2. Use a single scanf to read input values for the parameters input1, input2, and input3 (in order). Please use format "%d %d %d" or "%d%d%d".
3. Calculate the output of the circuit and compare it with the assumed output.
4. Please print the inputs and internal states (the outputs of each gate) of each stable state found with a single printf: ("Stable state found at %d %d %d %d %d %d\n", input1, input2, input3, Gate1out, Gate2out, Gate3out). For example, if only one stable state exists for inputs "1 1 0" and all of the gates output 0, print "Stable state found at 1 1 0 0 0 0". If the inputs cannot end up in a stable state, please print nothing.

Compilation:

Use the command “gcc -Wall -g steady.c -o opt7” to compile your code.
Use “./opt7” to run your code.
Please note that if the code that you submit cannot be compiled, you will receive no feedback.

Checkout and feedback:

Checkout:

Use the "svn update" in your working directory to get the folder “OPTIONAL7” which has all the files. You can check out a new ECE120 working directory using “svn checkout https://subversion.ews.illinois.edu/svn/fa16-ece120/netid ece120”, where netid is your netid.

Commit:

After you commit your code to your svn repository (using the “svn commit” command), our tool will begin to grade your code. The grading should take a couple of minutes.

Feedback:

Please use the "svn update" command to get the feedback after the grading is completed. The feedback will be in separate files named “*.test”, for example, “0.test”. The feedback files are usually one or two sentences indicating your error. Test cases are also available for some errors. For this exercise, we will first test your printf formats, so please make sure that your printfs follow the requirements. For any questions regarding the test cases and this exercise, please email me at beipang2@illinois.edu or to Prof. Lumetta at lumetta@illinois.edu.
If the feedback has one and only one file “0.test” which says you have passed all the tests, then congratulations, you have completed this exercise.