Logic Simulator

Group Project Phase #1

The first phase of the group project is the implementation of a simple logic simulator. This is the continuation of your individual assignment on levelization. Therefore, you are already familiar with circuit parser and levelizer. Please read the following instructions carefully:

- Create a private GitLab (or GitHub) with your teammates as collaborators. This way you can share code among team members efficiently.
- The logic simulator simply reads a Boolean (0 or 1) as the input vector and generates the corresponding Boolean output vector.
- The input file format is simply a text file (`.txt`) and every line is representing one primary input. The format of each file should have a line representing the node ID of the PI and its value separated with a comma: `"PI-ID,VALUE"`. Total number of nodes in the input file must be equivalent to the number of primary inputs. Please refer to the sample input file given for `c17.ckt` circuit.
- Starting from this phase, we will have XOR gate.
- The output format is the same as the input format. The order of the rows is not important.
- You can use `C` or `C++` for your implementations. **We strongly suggest transferring your parser and levelizer from `C` to `C++`** as it will be much easier to code your next phases in `C++`.
- It is NOT necessary to keep the structure of `readckt.c` implementation. You are encouraged to start designing your own data structures, classes, functions, etc. As we go forward with the project, you need to add new features and methods to this code, therefore, following the *object oriented programming* (OOP) sounds like the perfect plan.
- You can have different source code (`.c` or `.cpp`) or header (`.h`) files.
- Create one shell script `"compile.sh"` that compiles your project and creates an executable file called `"simulator"`. We will compile your code by `"sh compile.sh"` command and then run it with `"./simulator"`.
- The main menu of the simulator is the same as the one used in parser and levelizer. Add a new option as `"LOGICSIM"`, which takes two string arguments as input: the name of the input file and the name of the output file.

- For submission, all the files should be in one single folder (no sub-folders) named: "EE658_Phase1_⟨GroupNumber⟩", for example: "EE658_Phase1_6 for group 6.
- Please use standard zip for compressing your submission folder using the zip command in Viterbi server (and not your personal computer, we will notice this!).
- To avoid future issues related to compiling with different versions of GCC, please make sure your code can be compiled on Viterbi server.
- Make a text file named "members.txt", and put the USC netID (username) of your group members in separate lines (do NOT include @usc.edu suffix or any names).
- Only one person per group should submit the project. If you have a revision, the same person should update the submission.
- In all communications about your project, please CC all members, and always respond by "Reply to All".
- Use the discussion forum for possible doubts or questions, or visit office hours. Please avoid sending personal emails to the course staff.

Our script will use "unzip" command to unzip your submission, run the "compile.sh" script, run the code with different circuits and input patterns using the same command format as provided above, and finally determine a grade for the member of the group (based on members.txt file). Please make sure your submission follows these instructions.