

Digital Design Project Report: Logic Circuits Simulator

Habiba Seif, Laila Sayed, and Zeina Elsayy

900211308, 900223389, 900223285

CSCE230102 - Digital Design I (2024 Spring)

Section 02

Department of Computer Science and Engineering, AUC

Topic or Problem Definition

A simulator is a tool that allows users to create a virtual representation of a digital circuit and observe its behavior without physically building it. An event-driven logic simulator models the behavior of digital circuits based on events, which occur when signals change their logical states. This approach is beneficial for simulating a dynamic behavior of digital systems where changes happen discretely in response to specific triggers. The goal of this project is to create an event-driven logic circuit simulator that fits certain requirements. The simulator is programmed to take three key inputs: a library file (.lib), a circuit file (.cir), and a stimuli file (.stim). Initially, all circuit inputs and connections are assumed to be in logic "0". The library file, which is structured as a text file, lists the components that may be used in the circuit, each on its own line. The circuit file describes the circuit's setup, including the inputs and components used (gate type, number of inputs, and outputs). Meanwhile, the stimuli file stores the external events that are applied to the inputs. The output expression contains the simulation results, including the time stamps and updated values for inputs, outputs, and wires. This file (.sim) defines how to compute the component's output using the specified inputs. Overall, the simulator produces a simulation file as output when the testing procedure is complete.

Data Structures and Algorithms

The data structures used in this project mainly consisted of maps and vectors. Firstly the maps used were unordered maps. Unordered maps in C++ are related containers that hold elements created by combining a key value and a mapped value. They were utilized in several aspects of the code we made, including but not limited to *wires*, *inputsOfCircuit*, *propDelay*, *gates*, *addGate*, and *parsedData*. Unsorted map assignments were used for storing wires, the main inputs of the circuit, propagation delays in each component in the circuit, and the gate name and gate information. It also is used in the 'addGate' void function which while reading every new line inserts a gate to the unsorted map. While for the element 'parsedData' consisted of a pair, the first element being an unsorted map and the second element just a regular map. Secondly, the vector implementation within our code was frequent, but the type only varied between string and stimulus. Vectors in C++ are often used to dynamically store elements with similar data types. The vector strings were used in *compGate* as *inputs*, *parseCircuitFile* as *gateInputs*, as parameters for the function *getValues*, for the function *readStimuliFromFile* to read the stimuli file and store the information, as parameters for the function *replaceVariables*, and many more. In summary, the data structures predominantly utilized are unordered maps and vectors, each of which is often used in various capacities throughout the code.

A brief explanation of a few main functions mentioned previously, which have a data structure implemented:

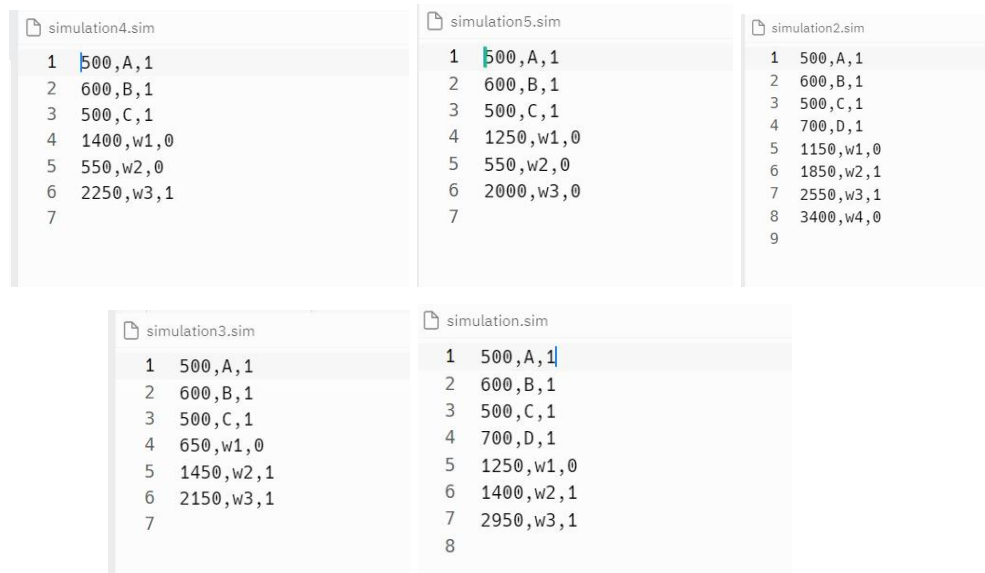
addGate- This function is employed to read the library.lib file.

parsedData- This function takes the circuit map and the input map as parameters as it is passed to the reading circuit function and splits up the components of the circuit (inputs and outputs) between the circuit map and the input map.

compGate- This is a struct that takes the component's *gateName*, *inputs*, and *output* in order to describe the components in the circuit file

Testing the Code

The code after all the implementations were corrected and updated ran smoothly. Here is a screenshot of the simulation files after running:



Comparing it to the objective output we need (according to the project file):

500, A, 1

700, w2, 1

800, B, 1

900, Y, 1

It shows that they are nearly identical, which is the exact result we are aiming for. However, this was achieved after several tries and errors, some of which took hours or even days to rectify. This brings us to the next section of the report, where we address the most significant obstacles we encountered and overcame.

Challenges

The first mistake we ran into was separating circuits. After Dr. Mohamed Shalan's comments on the signup sheet, we quickly updated this error. Each circuit now has its own library, circuit, and stimuli file. This also results in each circuit having the output displayed on its own simulation file. The following challenges however were not so easily solvable. Another small issue was figuring out how to divide the tasks equally, we combated this by mainly working together as one unit. On the rare occasion that we were not collaboratively working, each person focused on one aspect of the project to try to optimize the time we had to work in. The two major challenges we encountered were selecting the appropriate data structure for the library, circuit, and stimuli files, as well as determining how to trim and parse the components in the circuit and stimuli files in particular. To combat the first challenge we did research as well as asking for second opinions. Our fellow classmates and TAs helped guide us to ultimately choosing maps and vectors as our main type of data structure for this project implementation. Furthermore, we each did our own background research to ensure this was the best course of action. The second challenge was subsequently solved with the help of Chatgpt. We sought the code's fundamental concept but received the incorrect result. However, we took the core layout and idea and improved on it, resulting in the desired outcome. Overall, we were able to figure out a solution for every problem we encountered.

Contributions

Each member of the team collaboratively shared and helped each other in each component of this project. Several Zoom meetings, in-person meetings, and text messages were conducted in order to ensure the best possible results with fair distribution. Here is a brief breakdown of each student's work:

Habiba Seif-

Created 2 circuits

Made circ, stim, and sim files for the circuit

Main contributor to the main cpp file

Laila Sayed-

Created 2 circuits

Made circ, stim, and sim files for the circuit

Main contributor to the bonus task

Zeina Elsayy-

Created 1 circuit

Made circ, stim, and sim files for the circuit

Main contributor to the report

References

Chatgpt was used to help guide us through certain parts of the project:

- Creating the *inputsOfCircuit* element for the main inputs
- Inserting the inputs of each gate
- The function *replaceVariables*
- The function *evaluateBooleanExpression*
- Determine how to trim and parse the components in the circuit and stimulus files.