

Georgia Institute of Technology

Digital System Tests – Project 1

Logic Simulator

Lingjun Zhu

GTID # 903433936

Oct. 5, 2018

1. Data Structure

The objective of this project is to write a logic simulator program that can simulate the behavior of digital logic circuits and obtain the output vectors of a circuit based on given input vectors. In order to do this, the program needs to read the circuit netlist and build a data structure to store the logic gate types, wire indices, and the topology of the circuit, etc. The following class diagram shows the data structure used in this project, based on the basic data structures provided by Python.

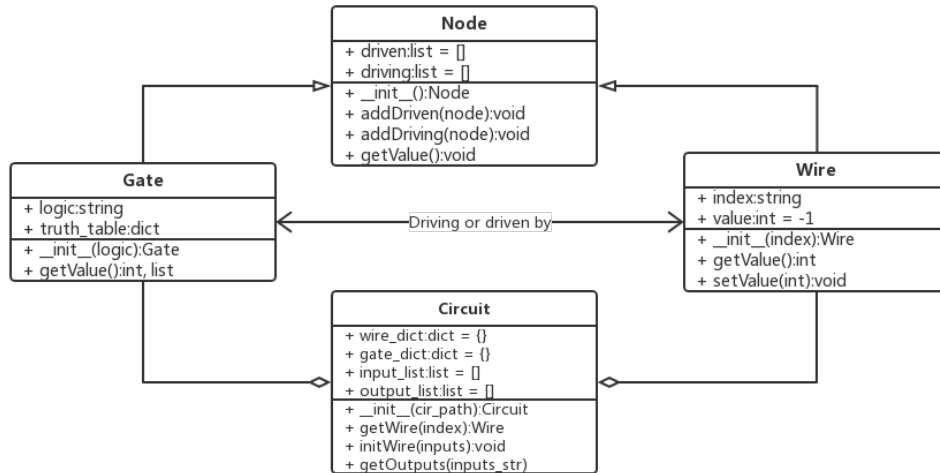


Figure 1.1 The class diagram of the logic simulator

Node is a base class used to describe the topology of a circuit. A **Node** object uses two lists to store the references of its driving nodes and driven nodes separately. It also provide functions to add driven or driving nodes. **Node** is inherited by **Gate** and **Wire**. A **Gate** object uses a hash table to represent the truth table and provides a function to get the output value of the gate. A **Wire** object has a unique index and an initial value of -1, while the value can be checked or set by functions. A wire driven by a gate is represented by putting the reference of the **Gate** object in the driven node list of the **Wire** object, and vice versa.

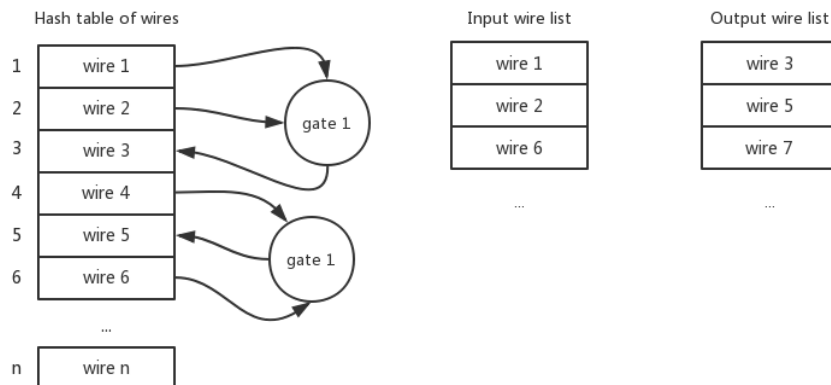


Figure 1.2 The structure of Circuit

A circuit consists of wires and gates. A **Circuit** object uses a hash table to store all the wires and two lists to store input and output wires separately. The initialization of **Wire** and **Gate** objects and the establishment of the connections are completed when the program reads the netlist.

2. Algorithm

First, the program can read in a netlist file and initialize a **Circuit** object based on the netlist. The initialization subroutine is demonstrated by the following flowchart:

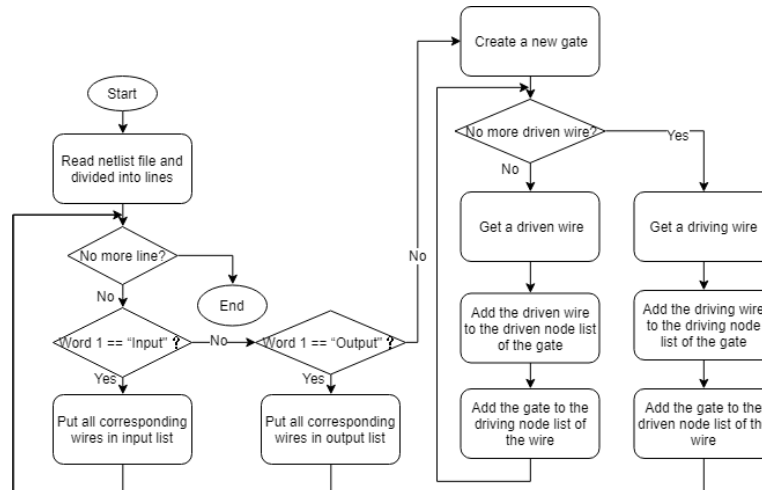


Figure 2.1 The flowchart of the circuit initialization subroutine

After that, the program can simulate the output of the circuit. The calculation subroutine will first initialize the input wires with the input vector and other wires with an initial value of -1. Then the subroutine will push an output wire into a stack then enter a loop. Every time the program pops a wire, it checks if the value of the wire is known. If so, it will continue the loop; otherwise, it will try to calculate the output value of the driving gate. If the gate has driving wires with unknown values, the program will push all the unknown wire into the stack; otherwise the program will set the value of the driven wire as the output value of the gate. By doing this repeatedly, the stack will be empty and the value of an output wire will be obtained. Then the program will push other output wires into the stack and print the output values in order.

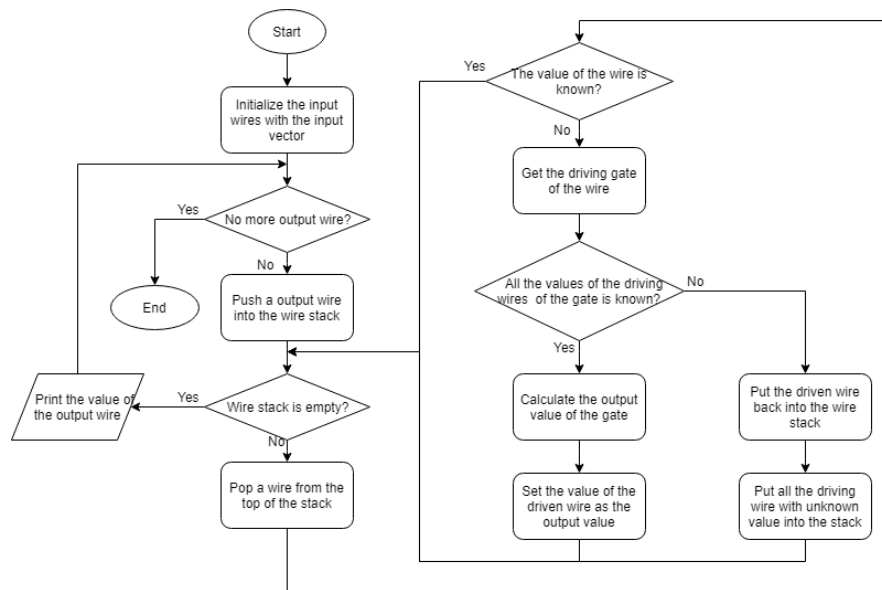


Figure 2.2 The flowchart of the circuit output calculation subroutine

3. Results

The inputs and outputs of the simulated circuits is shown as below:

Table 3.1 The inputs and outputs of the simulated circuits

Circuit	Input Vector	Output Vector
	1110101	1001
	0001010	0100
s27	1010101	1001
	0110111	0001
	1010001	1001
	10101010101010101	00000010101000111000
	01011110000000111	00000000011000001000
s298f_2	11111000001111000	00000000001111010010
	11100001110001100	00000000100100100101
	01111011110000000	11111011110000101101
	101010101010101011111111	101010101010101010101101
	010111100000001110000000	00011110000000100001111100
s344f_2	111110000011110001111111	00011100000111011000111010
	111000011100011000000000	00001101111001111111000010
	011110111100000001111111	10011101111000001001000100
	101010101010101011111111	10101010101010101101010101
	010111100000001110000000	00011110000000101011110000
s349f_2	111110000011110001111111	00011100000111010001111100
	111000011100011000000000	00001101111001110010001111
	011110111100000001111111	10011101111000001010000100

4. Conclusion

In this project, a logic simulator is implemented using Python. Four types of customized data structures are used to simulate the behavior of nodes, gates, wires and circuits. An algorithm based on wire stack is applied to calculate the output vector of a circuit with a given input. And 4 circuits are simulated and the inputs and outputs are shown in the above table.