Computer Assignment #1 - Gate Level Simulator (Preliminary)

Amirmahdi Joudi 2021-03-16

In this assignment, a gate level simulator is designed which is not line_oriented, and also glitches are shown in the output.
This design is explained below:

Wiring:

Wires are structures. The structure has string name, char value, int event_time and struct next_event. next_event has char value and int event_time. event_time shows the time of last event applied.

Compiler:

The following rules are checked:

- each line except endmodule has to end with ';'.
- module has to have name and it can not be same as keywords
- The necessary '(', ')', '#' and ',' are checked and extra spaces are removed
- all variables have to be defined and then be assigned as inputs and outputs
- wires, inputs and outputs can not have same names
- endmodule is necessary
- delay format can be #(to1,to0) or #(delay) but it can not be ignored as delay 0
- gates first argument is output and it can have any number of inputs

Logic:

A vector of gate pointers are stored here. Input values are received and events are made on wires. Then while there are events on any wire, the nearest events are found and current time is moved to event time. Then wire values change and gates calculate new values. If an output value changes, a final output line is created. This process is done till the end of inputs file.

Gates:

All gates are inherited from class gates. They store wires and int to1 and to0. To do calculations; they make events on outputs.

Timing Structures:

It is a new method for finding delays which is optimized and can handle and show glitches, too. While events found, the nearest one is found and all of the stage events are applied. Then gates calculations are done and new events are created. If the value of outputs change, a line for printing would be made. At the end, all of the lines are passed to main functions and they are printed there.