# Digital Circuit Simulation

A combinational circuit is made up of gates, which are devices that take Boolean (True / 1 and False / 0) input signals, and output a signal that is a function of the input signals. Gates take some time to compute their functions, so a gate’s output at time τ reflects the gate’s inputs at time τ − δ, where δ is the gate’s delay. For the purposes of this simulator, a gate’s output transitions between

0 and 1 instantly. Gates’ output terminals are connected to other gates’ inputs terminals by wires

that propagate the signal instantly without altering it.

For example, a 2-input XOR gate with inputs A and B (Figure 3) with a 2 nanosecond (ns) delay works as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Time (ns) | Input A | Input B | Output O | Explanation |
| 0 | 0 | 0 |  | Reflects inputs at time -2  Reflects inputs at time -1  0 XOR 0, given at time 0  0 XOR 1, given at time 1  1 XOR 0, given at time 2  1 XOR 1, given at time 3 |
| 1 | 0 | 1 |  |
| 2 | 1 | 0 | 0 |
| 3 | 1 | 1 | 1 |
| 4 |  |  | 1 |
| 5 |  |  | 0 |

A

O

B

Figure 3: 2-input XOR gate; A and B supply the inputs, and O receives the output.

The circuit simulator takes an input file that describes a circuit layout, including gates’ delays, probes (indicating the gates that we want to monitor the output), and external inputs. It then simulates the transitions at the output terminals of all the gates as time progresses. It also outputs transitions at the probed gates in the order of the timing of those transitions.

There are following issues with the program:

* JSON output **( simulator.exe <simfile> json** ) works, but simulator does not output gate transitions configured in simulator file (probe keyword) to the console – running **simulator.exe <simfile>** produces no output when it should.
* The simulation itself can be made much faster, e.g. 5devadas13.in takes a very long time to run in debug build. Please optimize the runtime of the application.
* The JSON output is quite slow. We are not interested in pretty formatting or tree structure, just the output must be well-formed. Could you please optimize it?
* There are other issues in the code (such as wrong types, missing initializers, wrong method declarations and names etc). Please submit code improvement changes/suggestions.

Please make a separate commit for each of the issues.