Simple Logic Simulation

EE 677 Project

Madhu kiran
100070046

Problem Statement

- To simulate a given circuit given the different gates and nodes involved in the circuit.
- The output should evaluate the final output values as well as the intermediate node values.

Input format

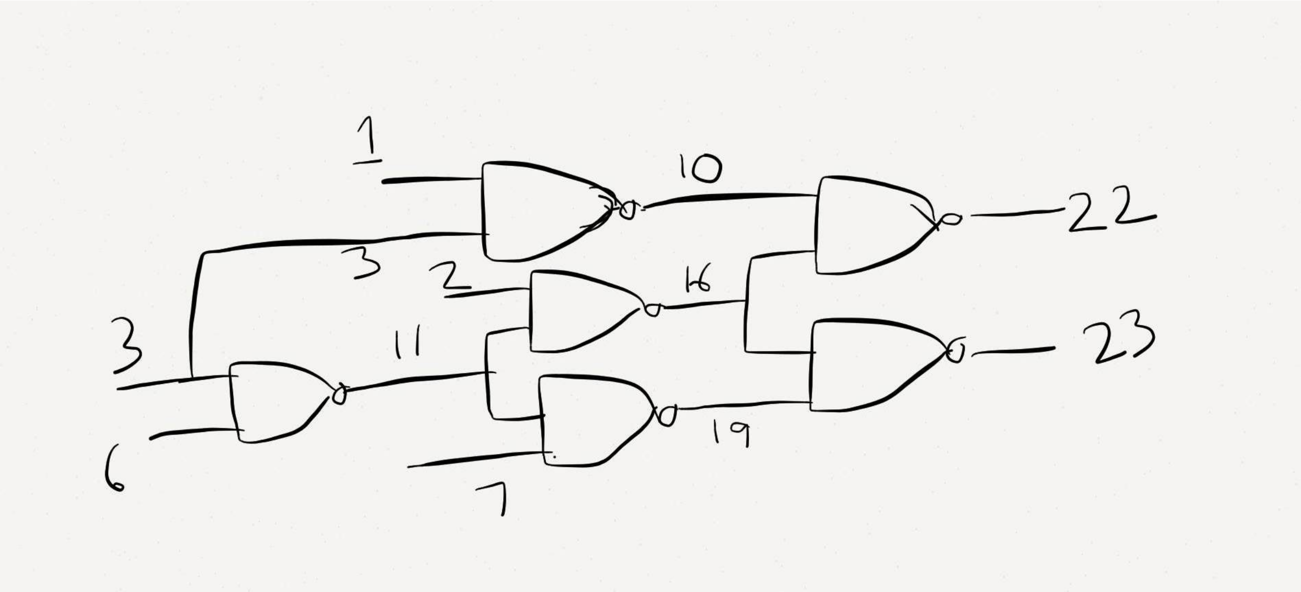
```
# 5 inputs
# 2 outputs
# 0 inverter
# 6 gates ( 6 NANDs )

INPUT(1)
INPUT(2)
INPUT(3)
INPUT(6)
INPUT(7)

OUTPUT(22)
OUTPUT(23)

10 = NAND(1, 3)
11 = NAND(3, 6)
16 = NAND(2, 11)
19 = NAND(11, 7)
22 = NAND(10, 16)
23 = NAND(16, 19)
```

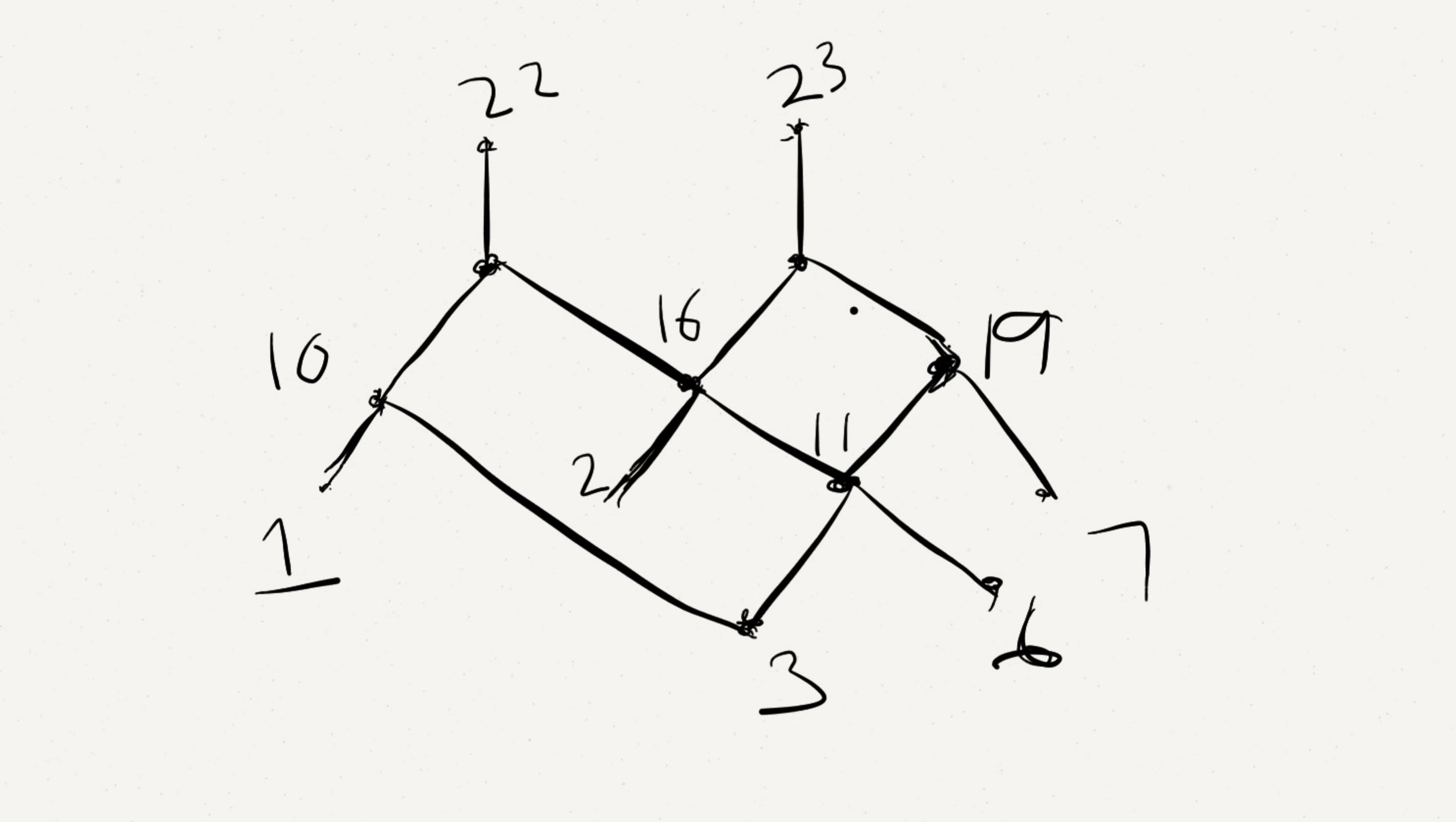
- The circuit is given as input. The format looks as shown for the following circuit:



Implementation (Nodes as objects)

- Each node of the circuit is an object with the following properties:
  - Gate: The gate whose output is the given node. For example in the previous circuit node 22 will have gate value as “NAND”
  - Value: The boolean value of the node
  - Inputs: The nodes which are the inputs for the corresponding gate of the node

Implementation (Breadth first)



Implementation (Breadth first)

- Initially a vector bfs is chosen as the outputs.
- Then we start visiting each node in the vector and keep appending the inputs of each node to the bfs vector. Before appending the new input node to bfs we check if it has already been included in the vector. If yes we delete the previously include node and add it again.
- We start computing our circuit node values in the reverse order of this bfs vector.

Improvements (Parallelizing)

- Here in this approach the node values were being calculated sequentially, but we can calculate the all the node values of one level simultaneously.
- For example the given circuit can be evaluated in 3 levels:
  - First we compute node values of 10, 2, 11, 19 then we compute 16 and then 22, 23.
  - If there are large number of circuit elements the parallel calculations can be done by concatenating the bools to for integers and then doing integer operations to find the corresponding outputs.

Improvements (Event based)

- In this approach whenever the inputs are changed the entire circuit is being re evaluated. But instead we could just evaluate the nodes that changed.
- This can be done by remembering the parents of each of the inputs and computing only those when the inputs are changed.
- This decreases the time order but increases the space required for storage.