Netlist Solver EE24B141-A4

Aim: Evaluating a system based on a given netlist and stimulus
Files: digitalsim.py

Instructions to run: *python3 digitalsim.py <filepath>.net*

The input file must be a .net conforming to the given input constraints and the output file is generated in the same folder with the name <filename>.json. This output can be visualised using wavedrom

Implementation:

The problem was approached using principles of object oriented programming and compiler design. Gates are represented using the Gate class which stores context information and the behaviour of the gate. The circuit is implemented using a syntax tree and a symbol lookup table which are stored in the Circuit class.

Functions:

1) parse_netlist()
   Takes in the given text and uses regex capturing to extract the required data.
   Generates a symbol lookup table for information regarding a gate
   Checks whether number of given arguments matches type of gate
2) AST_generate()
   Generates a tree which contains the topological order of gates
   Checks for logic and any unevaluable gates (infinite loops, undefined symbols)
3) Simulate()
   Takes in stimulus values for inputs and walks through the tree to create a time series representation for the gates and outputs in the circuit
4) To_wavedrom_json()
   Dumps the output to a wavedrom readable json format

Conclusion:

The program allowed for a deep-dive in compiler design and the various ways programs can be parsed and evaluated. The final approach was picked for an easier debugging experience and it generates the outputs that are expected of it. An example has been provided in the next page. Verbose details regarding implementation are given in the docstrings and comments present in the code, hence this report is concluded here.

Example

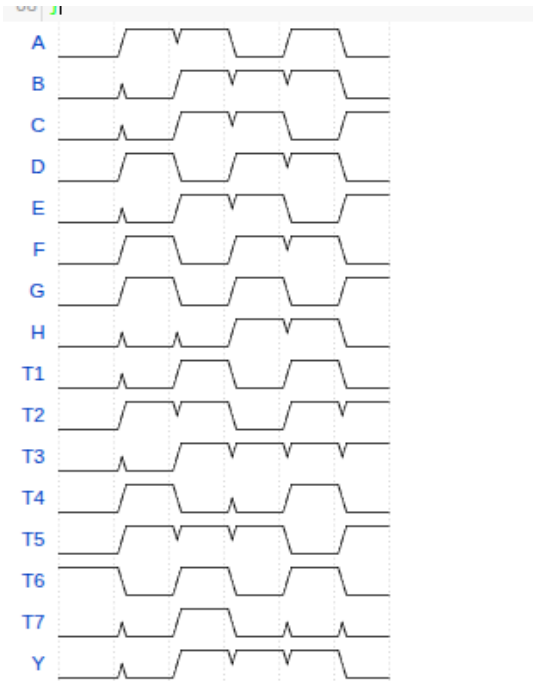
Terminal output

```
ronnie@ronnie-OMEN:~/Desktop/apl_material/A4_Digsim$ python3 digitalsim.py complex4.net
complex4.json
```

Netlist

```
INPUTS: A B C D E F G H
OUTPUTS: Y
GATES:
  T1 = AND(A, B)
  T2 = XOR(C, D)
  T3 = OR(T1, E)
  T4 = AND(T2, F)
  T5 = XOR(T3, T4)
  T6 = NOT(G)
  T7 = AND(T5, T6)
  Y  = OR(T7, H)
STIMULUS:
  0  0 0 0 0 0 0 0 0
  1  1 0 0 1 0 1 1 0
  2  1 1 1 0 1 0 0 0
  2  0 1 1 0 1 0 0 0
  3  0 1 1 1 1 1 1 1
  4  1 1 0 1 0 1 0 1
  5  0 0 1 0 1 0 1 0
```

Wavedrom



Topology