

returnSuccess // ELEC40006

This is the GitHub repository containing all our files for our Final Project of EEE/EIE Year 1 at Imperial College London (2019/2020).

Code written by :

- Bertil de Germay de Cirfontaine - CID 01767839
- Daniel Romano - CID 01728878
- Tanguy Perron - CID 01722386

Prerequisites

Boost 1.71.0 was used to write the Matrix library that we use. Therefore, it is required in order to compile our simulator (see below in the "Project files" section for the installation command). The rest of the files were written using standard C++ libraries on Ubuntu and compiled using GCC 7.5.0.

Guidelines to use the simulator

- To compile the simulator, run the script `compile_simulator.sh` using the command `$ bash compile_simulator.sh`.
- To run the simulator, use the command `$./Simulator < input_netlist.txt`
- The output is automatically recorded in a file `simdata.txt` (if it previously exists, it will be overwritten).

Note

- When using inductors and capacitors, make sure to use a timestep a lot smaller than the smaller period of the sources in the circuit (around at least 100 times smaller works best).

Project files

Libraries and other external files:

- Matrix library [matrix_library/] - by hbtalha [[Library GitHub Repository](#)]
 - Matrix.h - merged header of our Matrix struct and the one from the library
 - Matrix.cpp - definitions for the methods of the library
- Boost - required by the matrix library to compile the program
 - Install using: `$ sudo apt-get install libboost-all-dev`
- plotsim.m [matlab/] - MatLab script given to plot the results of our simulation

List of program files:

- simulator.cpp
 - AC simulator: the final program.
- network_header.hpp

- Contains all network structures and classes, as well as important function declarations.
- network_functions.cpp
 - Contains all definitions of the functions associated with parsing the input circuit.
- analysis_header.hpp
 - Contains important general function declarations regarding the analysis of a circuit.
- analysis_functions.cpp
 - Contains all definitions of the functions associated with the analysis of a circuit.

List of test files:

Test programs [test_programs/] :

- test_1_network_output.cpp
 - Test program that establishes a Network and then tests the output capability of the main code.
- test_2_network_io.cpp
 - Simple test program that reads a Networks and prints it directly.
- test_3_analysis_listnodesandtimes.cpp
 - Test program that reads and prints a network, and then also prints the sorted list of its nodes.
- simulator_dc.cpp
 - Simple DC simulator (one iteration with $t=0$).

Test circuits [test_circuits/] :

- test_circuit_1.txt
 - [I/O] A simple input file to test the i/o capabilities. (not a tested working circuit)
- test_circuit_2.txt
 - [I/O] A more complex input circuit. (not a tested working circuit)
- test_circuit_3.txt
 - [General] A simple test circuit.
- test_circuit_4.txt
 - [Analysis] Full circuit with voltage sources and resistors to test the conductance matrix.
- test_circuit_5.txt
 - [Analysis] Same as test_circuit_3 but using unusual node names.
- test_circuit_6.txt
 - [Debug DC] Circuit for debugging purposes: similar to circuit 4.
- test_circuit_7.txt
 - [DC] Same as circuit 6, but using an ungrounded current source.
- test_circuit_8.txt
 - [DC] Similar to circuit 4, but with a current source and testing edge cases.
- test_circuit_9.txt

- [AC (only R)] Similar to circuit 4, but with a sine voltage source, a current source and more sensible .tran instructions.
- test_circuit_10.txt
 - [AC] Similar to circuit 9, but with C and L instead of some resistors.
- test_circuit_11.txt
 - [AC] RCL test circuit
- test_circuit_12.txt
 - [AC] RC test circuit
- test_circuit_13.txt
 - [AC] RL test circuit
- test_circuit_14.txt
 - [DC] RC test circuit
- test_circuit_15.txt
 - [DC] Two capacitors in parallel: doesn't work (conductance matrix not invertible).
- test_circuit_16.txt
 - [DC] RCC circuit - A resistor and two capacitors in parallel.
- test_circuit_17.txt
 - [AC] RCC circuit - A resistor and two capacitors in parallel.
- test_circuit_complex.txt
 - [AC] Complex circuit to test the capabilities of our simulator.

Debugging files [debugging/]:

- debug.sh - Debugging script that compiles simulator_ac.cpp with debugging instruction for runtime