



Comparing Circuit Topologies using an Abstract Data Model

Elizabeth Lee; Randy Lober, Antonio Gonzales,
Jonathon Kwok, Jonathan Woodbridge

Problem Statement:

Netlists, circuit modeling files, are unique to each simulator. In some software, if a circuit component is added, changed, or removed, custom names may revert to default values. For this reason, it is desired to be able to use an abstract data model to read in two netlists and compare the topology of the circuits to determine if these circuits are the same.

Objective:

The main objective of this project is the ability to determine if two circuits have identical topologies.

Impact and Benefits:

The ability to compare circuit topologies will reduce engineering rework involving translating circuit models from commercial tools such as Cadence OrCAD into the Sandia modeling tool, Xyce. By recognizing topologically identical circuits and mapping differences in device name, Sandia engineers will not need to reconvert circuits that have already been processed into their Xyce counterparts.

Approach:

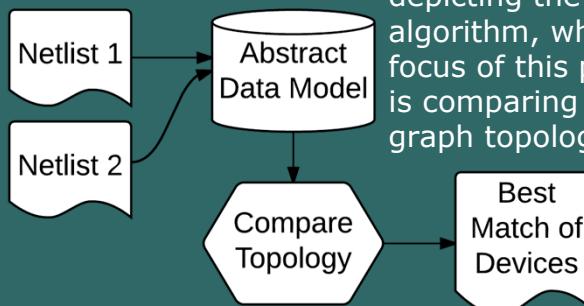


Figure 1: A flowchart depicting the algorithm, where the focus of this project is comparing the graph topology.

Results:

The topology comparison involves generating strings to fuzzy hash then reducing these strings using the Hungarian algorithm. This method is able to successfully map differences in name from topologically similar circuits. Future work will involve improving the algorithm to accommodate more complex circuit components and identifying similarities between circuits.

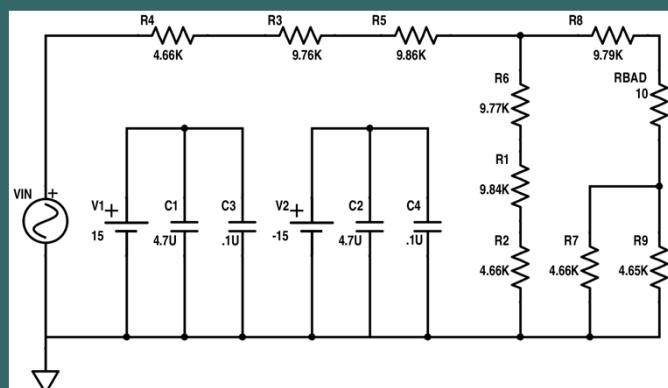


Figure 2: This is the circuit schematic of the netlist used as a test case for topology comparison.

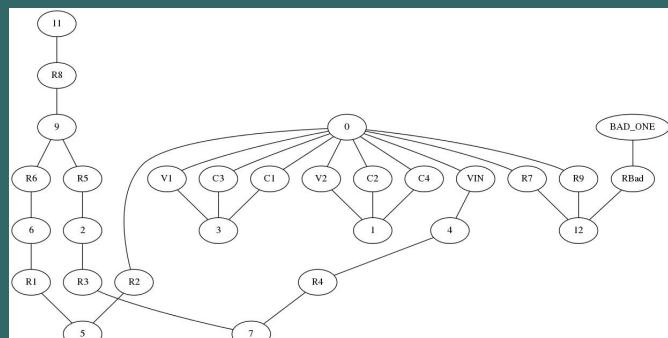


Figure 3: The Python Graphviz module provides a graph of the abstract data model representation of the circuit schematic shown in Figure 2.