

Project 4: Logic Simulation

Adam Sumner
ECE 449

Initial Release Due: 11/20/2015
Final Release Due: 12/04/2015

1 Initial Release

The features implemented in the initial release included implementing support for all of the combinational logic gates. This included being able to and, or, xor, not, buf, evl_one, and evl_zero inputs, resulting in the correct output. This decision was made because it would be easy to implement quickly and would allow for more time to focus on the harder algorithmic approaches to solving sequential logic circuits.

2 Test Cases

The test cases used were based on the previous tests from the netlist construction project. These test cases used only combinational logic, which could accurately test the results of the implemented gates for the initial release. Each test case included different combinations of the types of gates, ensuring the implemented solution performed correctly.

3 Final Release

The final release implemented the recognition of evl_outputs and writing the results to an output file. Furthermore, it includes a way to convert bus outputs into a hexadecimal format. The sequential logic for flip flops was implemented, but the solution did not pass all of the golden test cases. Due

to timing constraints from other projects being developed concurrently, a solution was designed, but could not be implemented in time. This solution involved a queuing system that would replace old values after the results had been written to the file. Likewise, this meant that `evl_lut` and `tris` gates were not able to be implemented in time. Once again, a solution had been developed, but it could not be integrated with the code in time for the final release deadline due to higher priority projects being concurrently developed. The final release is semi-functional, passing simple circuit test cases, but more complex circuits, such as a 32-bit cpu, cannot be simulated. With about a weeks more time, the simulator would be fully functional. Sacrifices were made at the cost of this simulator's functionality, but this is a common dilemma in industry. With a team of designated programmers, this issue would not happen.