“A Rule-Based Expert System for the Diagnosis of Convergence Problems in Circuit Simulation”

Christopher W. Lehman

- Definition of the Problem
- Related Work
- Expert System and User Interface Architecture
- Validation and Verification of the Solution
- Analysis of the Results
- Future Work and Direction of Further Research
- Conclusion

Definition of the Problem

- VLSI designs are too expensive to prototype
- Simulation is the only way to verify the design
- “Classical” simulators are prone to convergence problems
- Convergence problem analysis can be “labor intensive”

Colorado Technical University

• We can no longer do breadboard prototypes because they do not have the same electrical characteristics of the very small geometries of VLSI designs.

Photolithography costs for VLSI manufacturing can exceed $500,000.

The design has to meet specification and be manufactured at a profit the first time through the production cycle.

• The design must be simulated in software prior to manufacturing.

• SPICE – Simulation Program with Integrated Circuit Emphasis – was developed during the 1970’s, and is the “grandfather” of modern circuit simulators.

Most simulators use the Newton-Raphson root solving algorithm to solve for voltage and current values for each node of the circuit. NR can fail to **converge** on a solution if the initial values are of poor quality.

• Analyzing convergence problems entails the inspection of megabytes of text, a long and tedious process for large simulations.

Definition of the Problem

Hypothesis

"A rule-based expert system can diagnose DC Operating Point Analysis convergence problems for analog circuits by presenting to the circuit designer a set of user-configurable options within the HSPICE simulation environment that will allow the circuit simulation to attain convergence with an accuracy rate of at least 90 per cent."

Colorado Technical University

• A rule-based expert system will provide the solution to the convergence problem

• Only DC Operating Point Analysis will be addressed in the research, but this is the first step in the simulation cycle, and it must have a convergence solution in order to continue the process.

• Only analog circuits will be addressed in the research, but they are the type of circuit most prone to convergence problems.

• HSPICE (owned by Synopsys) will be the target simulator used in the research.

• Only a sub-set of HSPICE options will be used in the research. The sub-set will consist of those user-configurable options most often associated with convergence problems.

• The expert system will have at least a 90% accuracy rate over the test cases. i.e. at least 90% of the test cases will converge using the recommendations of the expert system.

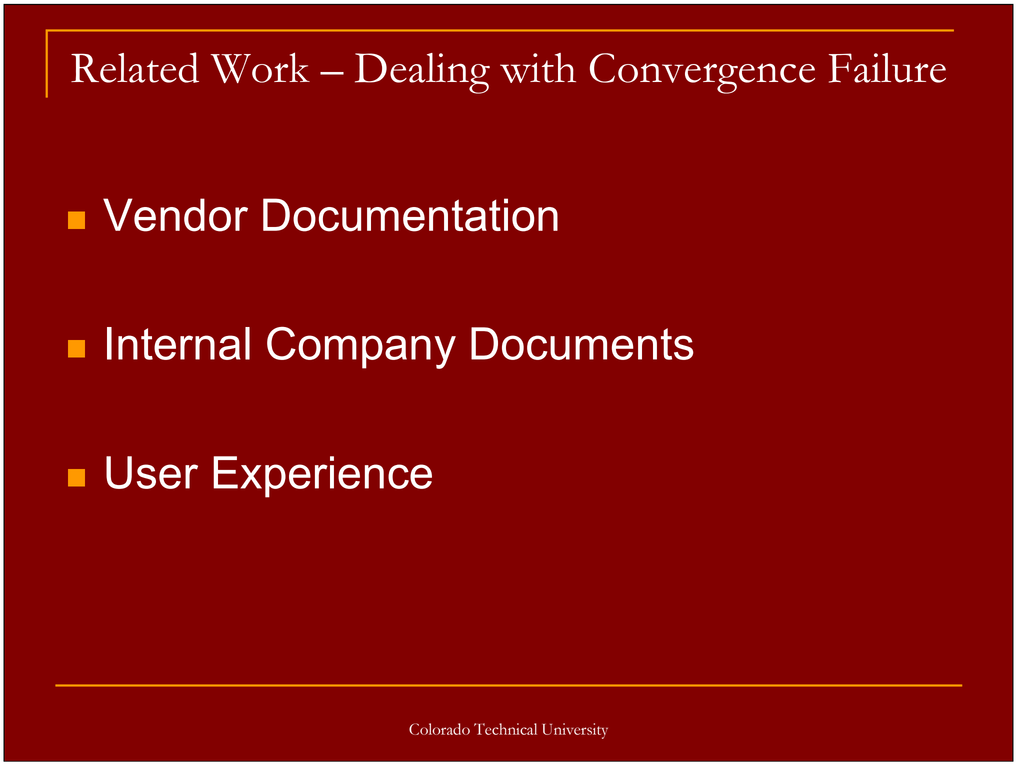
Related Work – Avoiding Convergence Failure

- Algorithmic Strategies
- Relaxation Solvers
- Continuation Methods
- Model Reduction Techniques

• Algorithmic strategies are code added to simulators to detect and deal with convergence problems “on the fly”. This is the prevalent strategy for avoiding convergence failure.

• Relaxation solvers break the simulation equations into smaller parts, and then perform calculations by guessing (relaxing) the behavior of the surrounding subsystems until convergence is obtained. Some late model mixed signal simulators use this method as part of their convergence strategy.

• Continuation and model reduction methods are effective but require the modification of circuit element models from the standard formats. Larger corporations with in-house staff to support custom models and simulators have been successful in using these methods.

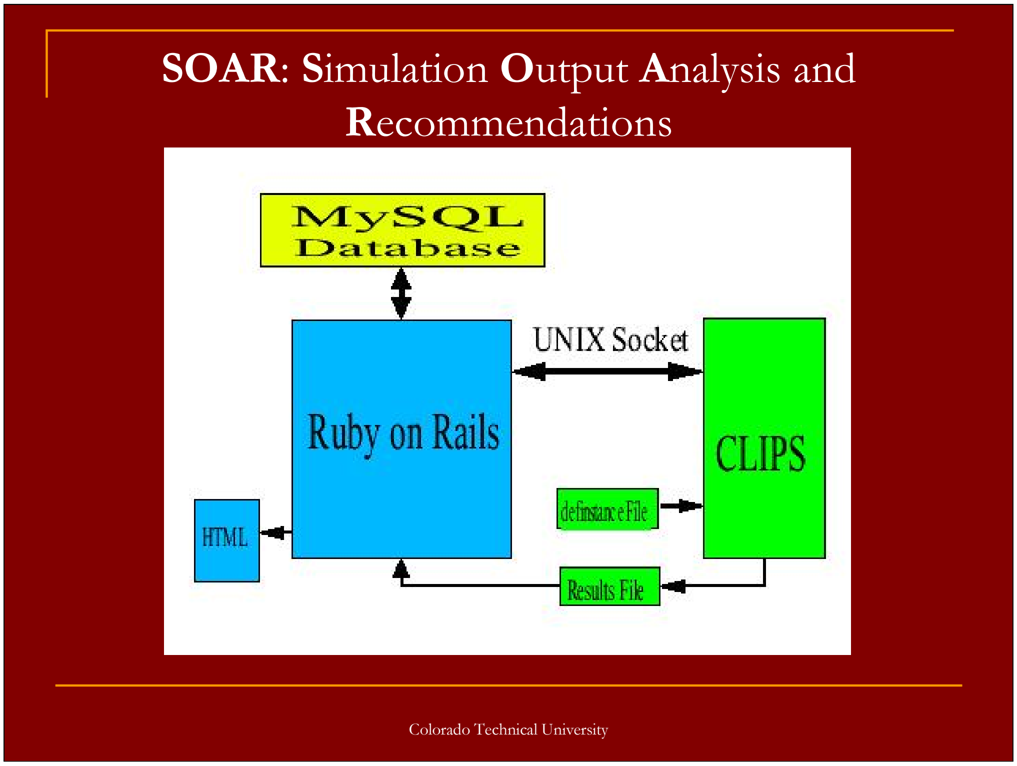


• Vendors of circuit simulators will often have a body of work that deals with procedures to analyze convergence failures

• Companies sometimes have internal documents describing procedures to take when convergence problems are encountered.

• Experienced designers have developed techniques for working through convergence analysis. This experience may be documented, but often it resides only in the designer's head.

• Expert systems are especially suited for converting written and human knowledge into rule sets.



• A database of simulation information is extracted from the original simulation output file. Database queries are generally faster than cycling through large text files.

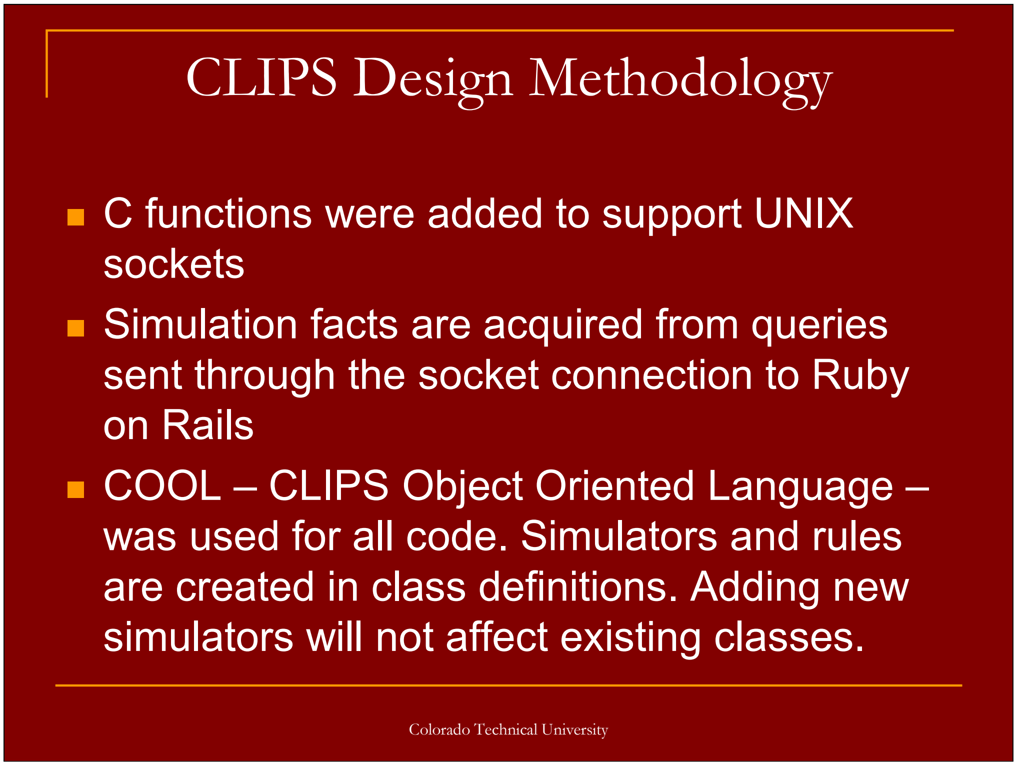
• Ruby is a scripting language developed in 1995. It has the scripting power of Perl with the OOP features of SmallTalk.

• Ruby on Rails (RoR) uses the Ruby language as a code generation facility for rapid deployment of web applications that use databases. RoR uses a Model View Controller (MVC) paradigm that allows the partitioning of program development into logical areas of functionality. SOAR contains roughly 400 lines of RoR code.

• CLIPS (C Language Integrated Production System) was developed by NASA in 1985. CLIPS is a rule-based forward chaining expert system utilizing the Rete pattern matching algorithm. Expert systems allow higher levels of knowledge abstraction than conventional programming, and are flexible enough to emulate human logic. Rules are used to represent heuristics or “rules of thumb” that are often used by human experts when attempting to solve problems. The rules are matched against a fact set that represents the knowledge associated with the problem to be solved. SOAR contains roughly 400 lines of CLIPS code, utilizing over 50 rules.

• A UNIX socket establishes a communications link between the RoR and CLIPS components of SOAR.

• Output to the user includes recommendations, line numbers and a database query facility.



- The new C functions were compiled and linked to a custom CLIPS binary file on the FreeBSD operating system
- RoR formats the database response to the query into a CLIPS *definstance* statement (a CLIPS fact). CLIPS saves the fact to a file. When all queries are complete, CLIPS reads in the *definstance* list containing all the facts known about the simlulation.
- Use of the COOL syntax will allow additional simulators to be added to CLIPS as new classes. The existing classes are not affected, so verification and validation is required only for the newly created classes.

CLIPS Verification and Validation

- Two test cases were derived specifically to test for CLIPS "false positives" and "false negatives"
- CLIPS recommendations were entered in the circuit stimulus, and the simulation was re-run to test for convergence
- HSPICE options not recommended were altered, and the simulation was re-run to test for convergence

• Case 1 exercises all CLIPS rules. CLIPS should issue a "no recommendations could be made" message.

Case 2 exercises all CLIPS rules. CLIPS should issue a recommendation for every HSPICE option covered in the rule set.

• A test case status of "Pass" is true only if the circuit converges when the CLIPS recommended course of action is used during another simulation run.

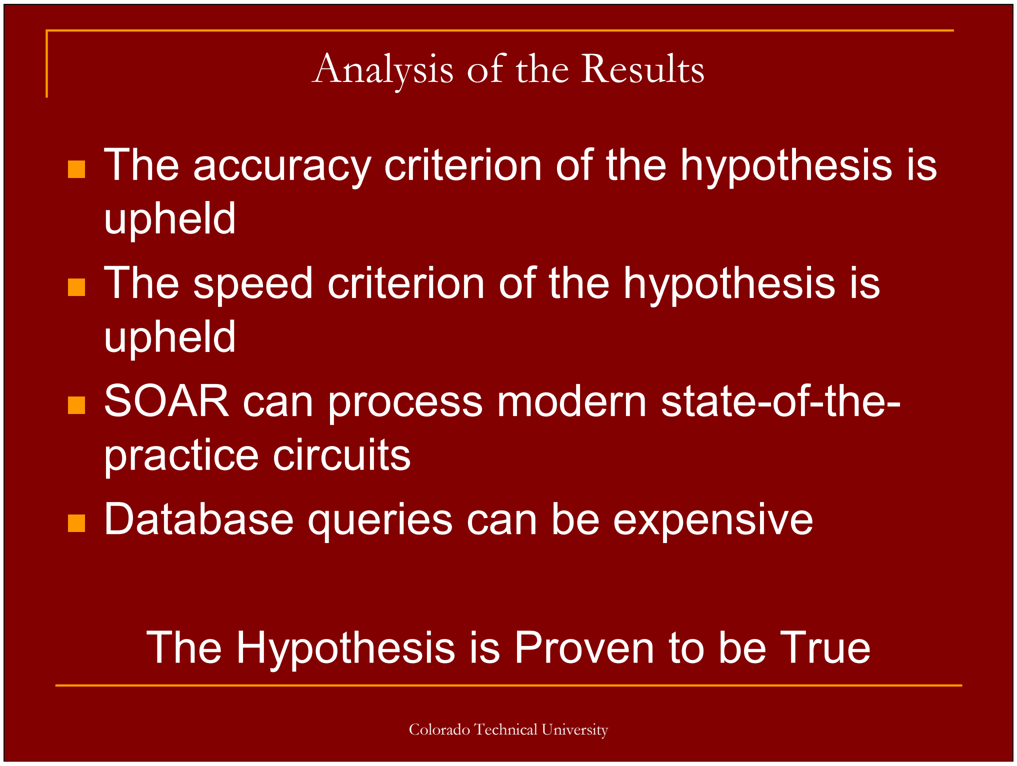
• HSPICE options that are not recommended but that might have an effect on the simulation's ability to converge have their values altered to test for convergence in subsequent runs. A test case status of "Pass" is true only if the circuit converges using the recommended changes but does not converge if non-recommended options are altered.

Validation and Verification of the Solution

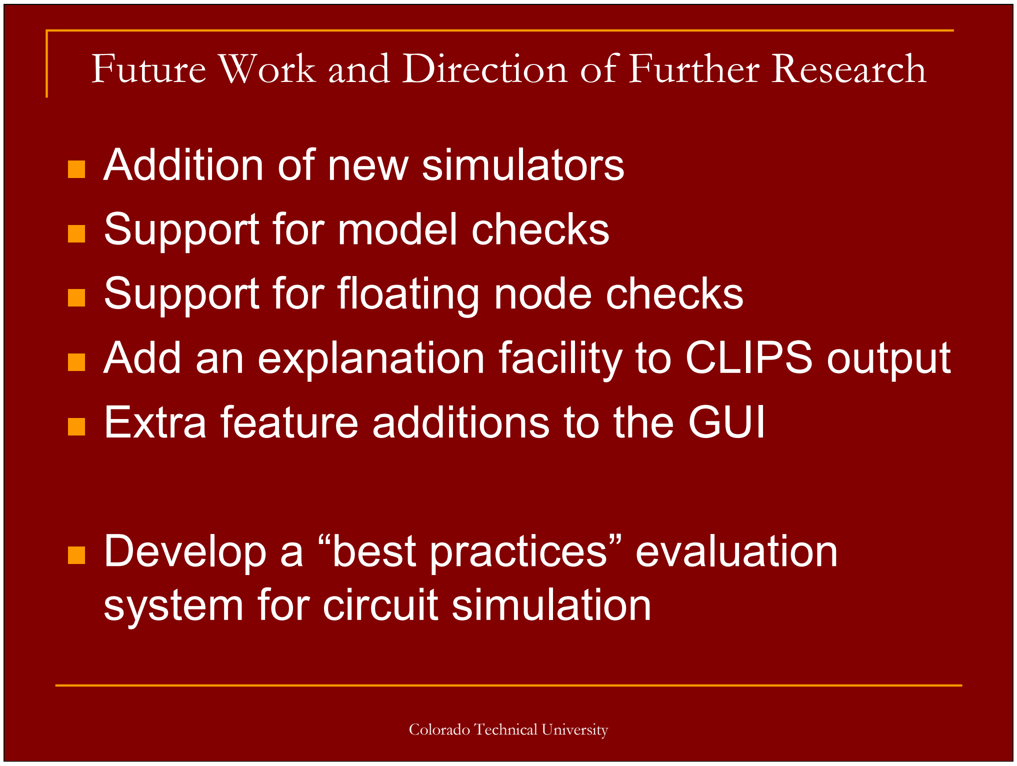
Name	*Test*	*Number of Circuit Nodes*	*Circuit Description*	*CLIPS Run Time (seconds)*	*Pass/Fail Status*
Case1	No Errors	1161	tAA	25	Pass
Case2	All Errors	1161	tAA	25	Pass
Case3	ACCT	1161	tAA	27	Pass
Case4	ITL1	1161	tAA	25	Pass
Case5	CPTIME	1161	tAA	26	Pass
Case6	RELV	16	Trip Point	25	Pass
Case7	ABSV	16	Trip Point	27	Pass
Case8	ABSI	16	Trip Point	25	Pass
Case9	Error Tol.	16	Trip Point	25	Pass
Case10	KCLTEST	510	LVDetect	25	Pass
Case11	DV	25	Senseamp	30	Pass
Case12	DCON	11131	Read 100C	32	Pass
Case13	GMINDC	11131	Read 25C	36	Pass
Case14	CONVERGE	11258	tHA	32	Pass

Colorado Technical University

- The circuit sizes used in the test cases span two orders of magnitude of the number of nodes in the circuit.
- Despite the size diversity in the test cases, the CLIPS run times remained consistently under one in minute in length.
- Cases 1 and 2 were used to validate and verify the CLIPS rule set. In Case 1, all rules were exercised, but the correct output should be (and is) a “could not find a solution” message. In Case 2, all rules are exercised, and the simulation was deliberately altered such that CLIPS should (and does) return a recommendation for each of the HSPICE options covered in the rule set.
- Case 9 combines the RELV, ABSV and ABSI options into one test case. These options have a high interaction with each other, so a special test case was developed to ensure that CLIPS could correctly detect various combinations of the error tolerance option values.
- Cases 12 – 14 represent a full chip simulation of a read cycle in a SRAM.
- The HSPICE options in the test cases represent the user-configurable options most used by designers and most likely to influence the ability of the circuit to converge.
- A “Pass” status means that the circuit converged after making the changes recommended by CLIPS.



- The body of test cases exceeds the required accuracy of 90 per cent.
- SOAR returns a recommendation well under the original four hour time limit. Database queries can significantly speed up program run times, versus sequential file access methods.
- SOAR has been demonstrated to scale up to full chip simulations with little increase in run time.
- Database queries accounted for roughly 80 per cent of the observed test case run times. This is due to the Ruby on Rails assumption that the FastCGI module will be supported in the web server application. Not all web hosting providers are willing to support FastCGI due to security issues and increased maintenance costs. Using only standard CGI, the web server must reload the complete RoR environment and re-establish database connections for each database query.



- With to the use of OOP methods, new simulators (and associated rules) can be added to SOAR with a minimal risk to existing simulator rule sets.
- Model and floating node checks would add significant ROI to SOAR. However, these features were beyond the scope of the dissertation.
- CLIPS can be enhanced to provide a “how I arrived at this conclusion” explanation facility that can inform the user of the logic behind the displayed recommendations.
- As run times increase due to the addition of new functionality, the use of AJAX methods to provide user status messages will become desirable.

- SOAR can be modified to act as a mentoring system or as a best practices system to help new designers through various parts of the design cycle. The capture of design expertise by SOAR may have a higher ROI than that of convergence diagnosis.

Conclusion

The Hypothesis Is Proven

Research accomplishments:

- Transformed simulation data into a database
- Transformed database queries into expert system facts
- Attained the targeted speed and accuracy
- Demonstrates that an expert system can process full chip simulation data

Colorado Technical University

Conclusion (Continued)

SOAR Provides:

- User Friendliness
- Ordered Lists of Simulation Data
- Timely Recommendations
- Easily Interpreted Output

SOAR: A tool to solve circuit simulation convergence problems.