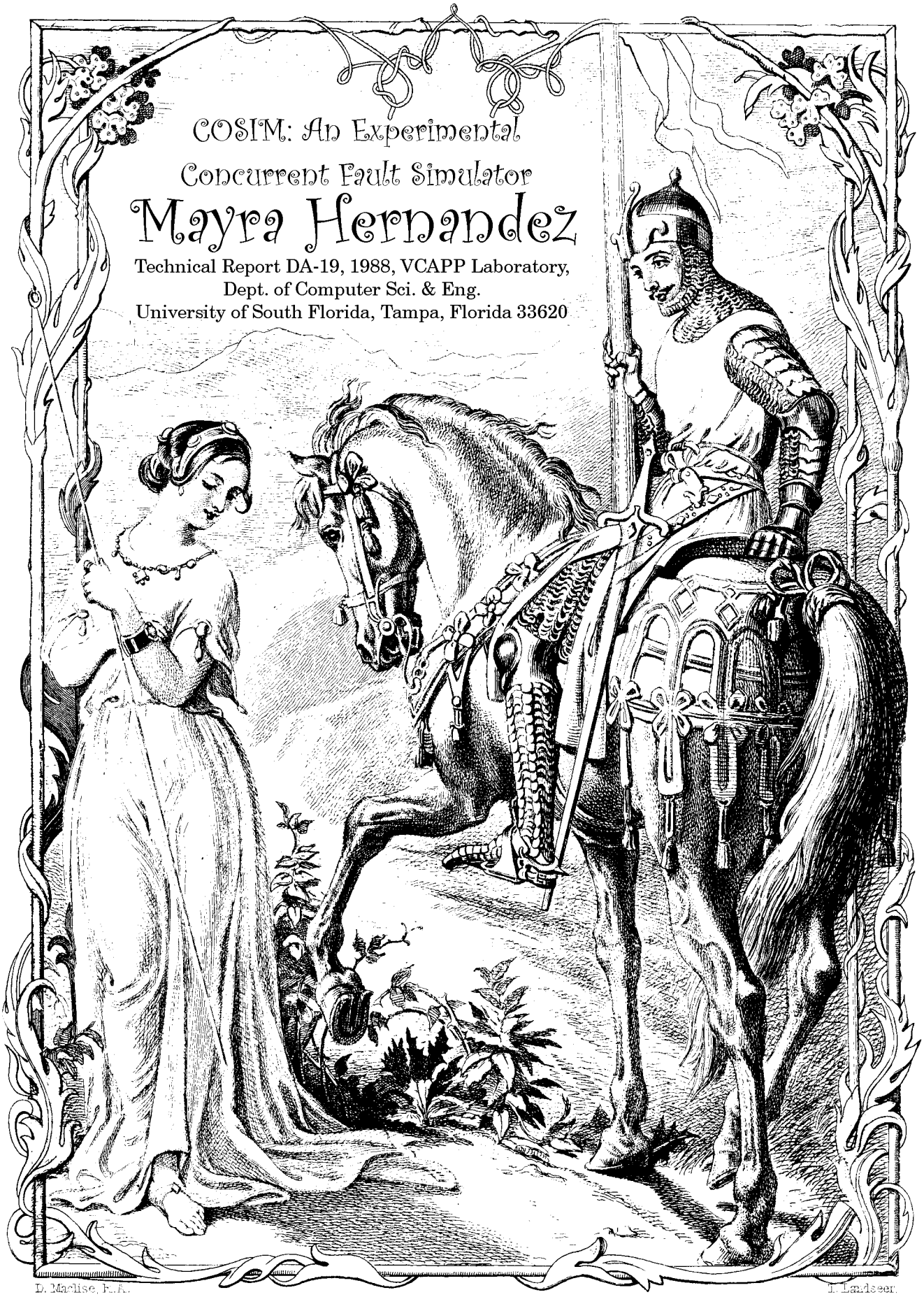
COSIM: An Experimental Concurrent Fault Simulator

Mayra Hernandez

Technical Report DA-19, 1988, VCAPP Laboratory,
Dept. of Computer Sci. & Eng.
University of South Florida, Tampa, Florida 33620



COSIM: AN EXPERIMENTAL CONCURRENT FAULT SIMULATOR

EXTENDED ABSTRACT

Mayra N. Hernandez
Peter M. Maurer
Department of Computer Science and Engineering
University of South Florida
Tampa, FL 33620

1. INTRODUCTION

Simulation is a process through which it is possible to study how different parts of a system interact without actually implementing a system or even a prototype.It is important because building and testing a prototype is time consuming and expensive. If the prototype does not comply with the requirements of a project, making changes to it or rebuilding it can use a great deal of valuable time whereas a simulator can be changed easily and quickly.

A fault simulator can help designers improve the quality and testability of circuits by helping to construct input vectors that will be used to test the final product. Concurrent fault simulation [Ulr 69] [Ulr 73] [Ulr 74] [Ulr 80] is the most efficient of the known methods for fault simulation. It was chosen for implementing the experimental fault simulator: *cosim*. *Cosim* is one of a series of CAD tools currently under development at the Department of Computer Science and Engineering of the University of South Florida.

2. COSIM

The *cosim* experimental fault simulator simulates faults in networks at the logic level. Seven types of gates can be analyzed: AND, OR, NOT, NAND, NOR, XOR, and XNOR. All of these functions are implemented using three-valued logic using the values 1, 0, and "don't know". The input files for *cosim* are designed following the syntax specifications of Maurer [Mau 88], which allows gates and nets to be declared, and allows faults to be attached to the inputs and outputs of a gate. Gates for which faults have been specified are called *fault origins*.

Cosim applies each test vector to the inputs of the gate network. The inputs of each fault-free gate are analyzed according to the gate type and an output is evaluated, and faults are (possibly) propagated to other gates using a mechanism called the *fault effect*. When a fault is propagated to a primary output, the fault has been detected.

Cosim uses the zero delay model to simulate circuits, which implies that all simulated gates take the same amount of time to produce their outputs, and that these outputs are produced in zero time. If any signals have the unknown value after the first pass, additional passes are performed to stablize these signals. Subsequent passes simulate only those gates required to stablize the unknown signals. Multiple passes are used to simulate circuits with feedback loops such as RS flip-flops.

After all of the input vectors have been exhausted, a report of all detected and undetected faults is generated.

There are several potential enhancements that could be made to *cosim*. Among these are the use of a unit delay or a variable-delay model instead of the zero-delay model, inclusion of other types of faults in addition to the classical stuck-at-one/stuck-at-zero faults, and performance enhancements such as removing detected faults for all vectors following the one that detects the fault. The logic model could be enhanced to include the high-impedence state.

3. CONCLUSION

Cosim is a logic simulator that can be used to verify logic models of circuits and to obtain fault-coverage information for physical test vectors. The *cosim* concurrent fault simulator is a useful design tool that is proving to be valuable addition to the collection of CAD tools available at the University of South Florida. In addition it will serve as a basis for further research in logic and fault simulation.

4. BIBLIOGRAPHY AND REFERENCES

[Arm 72] Armstrong, D.B., "A Deductive Method for Simulating Faults in Logic Circuits", IEEE Transactions on Computers., Vol C-21, No. 5, May 1972, pages 464-471.

[Bre 72] Breuer, M.A., "A Note on Three-Valued Logic Simulation", IEEE Transactions on Computers, April 1972, pages 339-402.

[Cha 74] Chang, H. et al., "Comparison of Parallel and Deductive Fault Simulation Methods", IEEE Transactions on Computers, Vol C-23, No. 11, November 1974, pages 1132-1138.

[Mau 87] Maurer, P.M., "Injecting Known Errors Into a Combinational Circuit: A Preliminary Report", Department of Computer Science and Engineering, University of South Florida, Tampa, FL, Technical Report no. CSE-87-00008, 1987.

[Mau 88] Maurer, P.M., "FHDL Tutorial", Department of Computer Science and Engineering, University of South Florida, Tampa, FL, Technical Report no. CSE-87-00011,1988.

[Mic 86] Miczo, A., "Digital Testing And Simulation", Harper and Row Publishers, NY, 1986, pages 118-183.

[Rot 67] Roth, J.P., Bouricius, W.G., and Scheider, P.R., "Programmed Algorithms to Compute Tests to Detect and Distinguish between Failures in Logic Circuits", IEEE Transactions on Electronic Computers, Vol. EC-16, October 1967, pages 567-580.

[Tho 75] Thompson, E.W. and Szygenda, S.A., "Digital Logic Simulation in a Time-Based, Table-Driven Environment Part 2. Parallel Fault Simulation", Computer, March 1975, pages 38-44.

[Ulr 69] Ulrich, E., "Exclusive Simulation of Activity in Digital Networks", Communications of the ACM, Vol. 12, No. 2, February 1969, pages 101-110.

[Ulr 73] Ulrich, E. and Baker, T., "The Concurrent Simulation of Nearly Identical Digital Networks", Proceedings 10th Design Automation Conference, 1973, pages 145-150.

[Ulr 74] Ulrich, E. and Baker, T., "The Concurrent Simulation of Nearly Identical Digital Networks", Computer, Vol. 7, April 1974, pages 39-44.

[Ulr 80] Ulrich, E. et al., "High Speed Concurrent Fault Simulation with Vectors and Scalars", Proceedings 17th Design Automation Conference, 1980, pages 374-380.