EECS 579 Fall 2001 Term Projects

A. Outline

This document describes the EECS 579 term project requirements and deadlines, and lists some possible topics. Proposals for projects on other topics are encouraged. You may also propose modified versions of the suggested projects.

Project Selection

Select a project (possibly modified) from this list below, or propose a new project. Projects may be proposed by individual students or by a team of two students. (The latter is for experimental projects only). You will find relevant reference material in the text, the course reserve file in the library, and on the World Wide Web. Copies of most of the cited papers can be accessed on-line via the UM Library (Mirlyn). *Some reading and thinking is necessary in order to select and properly assess a project. The time spent doing this now will pay off later.*

Proposal

Submit a short proposal (two typed pages maximum, excluding appendices for the main proposal) no later than **Tuesday October 23.** In the event that your first choice is not accepted, also state a second choice and be prepared to expand it into a full proposal later, if asked.Every effort will be made to give students their first choice, but a reasonable distribution of the topics is required. Projects will be assigned by the instructor based on the quality of the proposals. Your proposal should have the following format:

1. Title, student name(s) and e-mail address(es)
   Project outline: objectives, method of attack, expected results, and other pertinent data.
   Time schedule listing dates of major "milestones"
   Work division (applies to two-person projects only)
   References (list only those actually consulted in preparing the proposal)
2. Brief second proposal (title and one paragraph), in case your first one is not accepted.

Progress Report

A one-page written progress report will be due **Tuesday, November 20**. This should summarize how the project is progressing, describe any unexpected difficulties encountered, and describe any significant changes in the project plans.

Oral Presentation

Short oral presentations of the projects to the class will be scheduled during the period **December 11–13**. Presentation time will be 15 minutes or so per project. (Note that the final exam is on Friday December 21, 10:30am – 12:30pm.)

Final Written Report

The project will be documented in a term report due by **Monday December 17**, at 5:00pm**.** This report will be a major factor in determining project grades. The term reports should be of professional quality and be in the usual format of a technical report or a journal paper. Experimental projects may have shorter reports, with detailed material such as program listings, sample runs, etc. placed in appendices.

All **term reports** should be computer-printed and have the following standard format:

Title page
Abstract (half-page summary of the report)
Sec. 1. *Introduction* (problem definition, background material, prior work, goals, methods)
Secs. 2 through *k*. (body of the report in as many sections as needed—usually 3 to 5)
Sec. *k* + 1. *Conclusions* (evaluation of results, lessons learned, and suggestions for improvement or further work)
References
Appendices (if appropriate)

B. Project Suggestions

A list of possible term projects follows. *The cited references should be viewed as a starting point only* in the project selection process. Copies of most of the papers mentioned here are accessible at on-line via the UM library. Note that there is a huge and up-to-date Bibliography at the end of Bushnell and Agrawal.

Experimental Projects

1. Implementation of the combinational FAN algorithm

Write a program for any available computer system to implement the FAN ATPG algorithm for large combinational circuits. Your program should be able to generate efficiently a complete test set for all SSL faults in a given circuit. Incorporate all (or most of) the speedup heuristics of the FAN algorithm. Also incorporate some simulation features to allow one test to detect many faults. Run your program on test cases, including the ISCAS-85 benchmark circuits, and compare your program's performance to published data on other programs in the literature. **Ref.** Text, Chap. 7.

2. Implementation of the sequential PODEM algorithm

Write a program for any available computer system to implement a version of the PODEM algorithm for small to mid-sized sequential circuits. The primary goals here are accuracy and complete SSL fault coverage, rather than fast computing speed. Run your program on some test circuits, including some of the smaller ISCAS-89 benchmarks, and compare your program's performance to published data on other programs. **Refs.** Text, Chap. 8.

3. Testing a digital IC from the VLSI design class

This project is aimed primarily at students who have taken a VLSI design class (EECS 427 or 627), and (ideally) have designed and fabricated a chip that needs to be tested. Chips designed by others may be used if available and adequately documented. We have a high-performance computer-driven IC tester (Hewlett-Packard Model 82000), which we will use as the testing tool. This project will involve deriving tests for, and actually testing, the said chips for functional correctness, and performing such parametric measurements as schmoo plots. Some recent VLSI designs should be available from other students' projects in previous semesters.

Survey/Research Projects

4. Hardware design verification techniques

EECS 79 is concerned with testing circuits that are subject to physical defects. A related, but harder problem is to devise testing methods for faults (design errors) created in the design process—this is the basic testing problem in software. Survey and evaluate the methods of design verification that are applicable to verifying VLSI chips, focusing on methods that attempt to model design errors explicitly and to generate tests for the modeled errors automatically. The relations between

verification testing and physical fault testing should be discussed and analyzed.
Refs. (1) M.C. McFarland: “Formal verification of sequential hardware: a tutorial,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, pp. 633-654, May 1993. (2) D. Van Campenhout et al. “High-level design verification of microprocessor via error modeling,” *ACM Trans. on Design Automation of Electronic Systems*, vol. 3, pp. 581–599, Oct. 1998.

5. Test generation using parallel processing techniques

Study the suitability of general-purpose or special-purpose parallel computers for speeding up test generation. What computer architectures and algorithm types are suitable for test generation at various design levels? Opportunities exist to do some experimentation (testing ideas or implementing a complete program) using various parallel systems to which we may have access, including various networks of Unix workstations (NOWs) and an IBM SP computer. Familiarity with parallel processing techniques is recommended for this project. **Ref.** P. Banerjee: *Parallel Algorithms for Computer-Aided Design*, Prentice-Hall, 1994. [UM Media Union library TK 7874.75.B361 1994]

6. Synthesis of efficient deterministic test generators

Most BIST techniques for logic circuits use low-cost (pseudo)random test-pattern generators. This is not so easy to do in the case of deterministic test generation, where a given set of tests is to be produced, e.g., tests previously generated by PODEM.Many large, practical circuits can be fully tested (100% SSL fault coverage) with small but carefully selected test sets. For BIST, these tests must be generated on-chip. Storing them in a ROM is one possibility, but it usually takes a lot of hardware (chip area, which can be unacceptable. This project will investigate the design of small sequential circuits (special-purpose FSMs) that generate a specific, deterministic set of test patterns, while not requiring a lot of logic. **Ref:** Text, Chap. 15.

7. Analog and mixed-signal testing techniques.

This research project will address some key testing problems in circuits that combine analog and digital components, such as digital signal processors (DSPs). Although digital testing is well understood, the same cannot be said for analog testing. Simple, easy-to-use fault models like the SSL model exist for digital testing, but the modeling of analog faults is far more difficult and poorly understood. Built-in self-test (BIST) is now making inroads in commercial digital designs; however, mixed-signal BIST remains in a rudimentary state.This project will investigate the modeling of analog faults and the development of BIST methods for mixed-signal applications. **Ref.** Text. Chap. 10–11.

8. Testing circuits composed of IP (intellectual property) cores

The problems associated with testing large predesigned digital modules, such as microprocessors and I/O interface controllers, is currently a hot topics in digital design. Cores are typically functional blocks whose internal implementation details are only partially specified. Testing such cores is very difficult, especially when they are combined with other cores and user-designed logic. The goal of this project is to survey the issues associated with testing cores, analyze the main problems, and to propose some solutions to the problem of high-coverage testing of embedded cores in system- on-chip (SOC) designs. **Refs**. (1) Text Chap. 18. (2) H. Kim and J.P. Hayes: “Realization-independent ATPG for designs with unimplemented blocks,” *IEEE Trans. on Computer-Aided Design*, vol. 20, pp.290–306, Feb. 2001. (3) Hyungwon Kim’s Ph.D. thesis (1999) in the 579 reserve file.

9. On-line testing of complex digital ICs

For critical applications such as car or airplane control, on-line fault detection and monitoring is as important as manufacturing testing This project involves developing and implementing a design plan for on-line testing of a new digital IC of the system-on-chip (SOC) type. This plan should include both periodic testing for permanent hardware faults as well as continuous testing for transient faults. As it is not always possible to test all parts of an IC continuously, a rational plan for prioritizing faults should be provided. The overall plan should contain provisions for internal hardware modifications as well as external testing. For example, if the chip contains a microprocessor, it is expected that the microprocessor will play a key role in testing. **Ref**. H. Al-Asaad et al.“Online BIST for embedded systems,” *IEEE Design & Test of Computers*, vol. 15, Issue 4, pp. 17 -24, Oct.-Dec. 1998.

C. Sources of Information

Books

Textbook (Bushnell and Agrawal).

EECS 579 course reserve file in the Media Union Library.

Other books: see Bushnell and Agrawal: Appendix C and Bibliography.

Journals

IEEE Design and Test magazine

IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

Journal of Electronic Testing (JETTA)

IIEEE Transactions on VLSI Systems

Conference Proceedings

International Test Conference (ITC) Proceedings (Oct.)

VLSI Test Symposium (VTS) Proceedings (May)

Design Automation Conference (DAC) Proceedings (July)

All the above are annual IEEE-sponsored research-oriented conferences, and are accessible on-line. The Media Union Library also keeps hard copies of IEEE conference proceedings together in a special, blue-bound collection.

The World Wide Web

This is a vast, unstructured source of information on all topics. Recent papers (or relevant Ph.D. theses and technical reports) are easily found and copied from the Web. Search by topic, author name, university or company name. Using the University Library’s web site *http://www.lib.umich.edu/ummu/ieee/* you can access electronic versions of all IEEE conference and journal papers published in the last 12 years or so by the IEEE.

When all else fails, check with J. P. Hayes for help in locating information.

D. Schedule Summary

Project Deadlines

Proposals due:	Tuesday, October 23
Progress reports due:	Tuesday, November 20
Oral presentations:	December 11–13 (times to be scheduled later)
Written reports due:	Monday December 17, 5:00pm

J. P. Hayes 10/8/01