University of Southern California
EE658
Diagnosis and Design of Reliable Digital Computers
Summer 2011

Class No.	911/30662D off campus; 906/30661D on-campus
Website:	tbd
Room:	**OHE 100B**
Time:	**TTh 1:00-2:55 PM**
Instructor:	M. A. Breuer
Office Hours:	Th 3:30 -4:30 PM EEB 300C and by appointment
Telephone:	(213) 740–4469
Fax:	(213) 740–9803
Email:	mb@poisson.usc.edu
Teaching Assistant:	tbd
Office hours	TBD
Textbook:	**Digital System Testing and Testable Design** M. Abramovici, M. A. Breuer and A. D. Friedman IEEE Press-revised printing, 1995. ISBN 0-7803-1062-4

Components in grade:	Midterm	20%
	Final	30%
	Project	30%
	Homework	20% (includes some programming projects)
	No Extra credit	
	Total	100%

Grading: Grades (A, B, C, D, F) on exams are based on an absolute system, namely, on what I think a student should be able to demonstrate. Hence students are not in competition with one another; the entire class can get an A or a C. **Therefore, I encourage you to share your knowledge with your fellow students, but not your papers**.

Regrading: If you think we made an error in grading an exam or homework problem, please put **a note on the top of the first page** as to the nature of the problem, e.g. "please regrade problem 4.2," and resubmit to the **instructor only**. If you are later dissatisfied with the regraded score, then make an appointment with the instructor. Graders and TAs do not deal with the regrading process. *In regrading, we may look at all of the answers in your paper; we may retotal your score; you may actually lose points.*

Homework: --Homework and solutions will be made available via the **web**.
--Homework problems, some of which are computationally intensive, are **different** from exam problems, which often deal with a concept. As an example, in EE 101 for homework we might ask a student to find all the prime implicants of a large function, but on an exam we might ask a student: what is a prime implicant; what is the importance of a prime implicant; give a formal definition of a prime implicant, etc.
--Homework is due on the day posted on the assignment. However, homework can be turned in at a latter time--referred to as **delayed--**without any penalty. Homework is considered **delayed** if received by the instructor 5 minutes before the **next** scheduled meeting of this course subsequent to the posted due date. At this time the instructor will time stamp and initialize the paper. You **MUST** also mark your paper **Delayed #1** the 1st time you exercise this option, and **Delayed #2** the second and last time. This allows students to be on travel, to be sick, etc. and not have to miss an assignment. However, you can only take advantage of this concept **twice** during the semester. Homework turned in that does not conform to these guidelines is

considered **late**. Late homework loses 20% of the maximum possible score for being one day late, 30% for being two days late; and is not accepted after this time.

Cheating: University policy will be followed concerning cheating on exams and other course work. No form of cheating is allowed. Look at SCampus or other USC sources to make sure what we consider cheating; it might be different than how it is defined at your previous institution. Please bring a photo ID to all exams. See below for more information.
--You can **discuss** your homework with your fellow students, the instructor or the T.A., but when you sit down to write, **do not use a copy** of anyone else's work. I will follow University policy and turn in suspected problems to the **Office of Student Conduct.**
--**Computer projects** follow the same guidelines unless you are explicitly working on a team project. If in doubt, ASK. Never use code that you get from the web, or other students, etc.

Misc. information:
--All exams will be **closed book** (and notes) unless specified differently.
--You are **encouraged to see the TA and/or the instructor during their office hours**. The grader does not have office hours nor meet with students. Questions submitted via email might not be answered the day they are submitted. In addition, answers requiring figures and/or equations might not be completely answered via email.

Neatness:
--Use **standard engineering documentation** practices on all material that you submit, including your name (print and sign), student ID number, course number, date, page number, engineering quad paper, staple, title of document, e.g. Homework Number 1. You will lose points either directly and/or indirectly on work that we cannot read and/or is messy or does not conform to the standards outlined above. Write on the light side of the paper. In addition, please check grammar and spelling on all assignments you do at home.
--Format to be used:

Staple in top left hand corner
One inch from the top put in the following information.
Your name (printed) **EE 658 Fall 2010** **Date** **page number**
Signature (optional) **Title of Assignment (e.g., Homework #5)**
Student ID

Philosophy

I. Why take this course:

One key issue in any product is quality, which is related to correctness, reliability and maintainability. One way to control and/or measure quality is via **testing**. Since a significant percentage (50-80%) of newly fabricated VLSI chips are defective, they all must be thoroughly tested to determine which are the good ones, and which are the bad (faulty) ones. In the development and production of digital systems, such as VLSI circuits, testing is a significant cost, and in some cases is over 50% of the cost of manufacturing. In addition, these costs are growing year after year. Thus a major part of this industry focuses on testing, with a goal of doing both a better job in differentiating between good and bad chips, and reducing costs. This is a very challenging problem.
This course will prepare two types of individuals who want to be involved in the design of VLSI chips. One type is a **VLSI design engineer**. Design-for-test, built-in self-test, and test generation are an integral part of the design process, and all designers need to understand these issues. In addition, CAD tools support the entire design process. Many of these tools deal with test issues, hence **CAD tool developers** need to be experts in the field of test.
Because of higher clock rates and VLSI scaling, test issues are becoming design issues, and vice versa. New fault models deal with both capacitive and inductive coupling (crosstalk), ground bounce (simultaneous switching noise), and substrate related noise. Hence test issues span the spectrum from low-level circuit theory to high-level behavioral operation.

II. Course Description:

This course focuses on (1) the testing of digital circuits and (2) the design of computer aided design (CAD) tools to automate aspects of testing. We primarily consider a gate/flip-flop model of a circuit but in some cases we deal with a lower level model such as transistor, or a higher-level model, such as a functional or register-transfer (RT, VHDL, Verilog) model. The course covers the following topics.

- **Modeling** – here we briefly study how to represent digital circuits in a computer aided engineering (CAE) environment.
- **Defects/failure/faults/errors** – we next study the manufacturing process and determine how and why circuits either are faulty when manufactured, or else fail in the field. We study different aspects of characterizing and modeling faults, such as opens and shorts.
- **Logic simulation** – we study how to design CAD tools that will efficiently and accurately simulate digital circuits at the Boolean level using various delay models.
- **Fault simulation** – we next extend the simulation techniques to include fault simulation. Here our CAD tools carry out the following process: given a logic circuit, a sequence of binary input patterns and a list of potential faults in the circuit, determine the output the circuit would produce in response to the input patterns if any of the faults were present in the circuit. From this we can measure the quality, i.e., fault coverage, of the input "test" patterns with respect to a fault model. We can also extend the results to include fault diagnosis.
- **Test generation** – this CAD tool, also know as automatic test pattern generation (ATPG), and solves the following problem: given a circuit and a potential fault, generate a test that if applied to the circuit would detect the fault if the fault were present. We study many algorithms that go into ATPG systems.
- **Design-for-test** - because ATPG is such a complex problem it is often the case that designers modify their designs so that ATPG systems will be more effective. This area, where designs are enhanced to aid testability, is called design-for-test (DFT).
- **Built-in self-test** – deals with a form of testing where the circuit tests "itself", rather than being tested by a piece of automatic test equipment (ATE). The study of built-in self-test (BIST) will focus on topics such as test pattern generation, signature compaction, linear shift registers (LFSRs), and BIST architectures.
- **Boundary scan** – we will also discuss the IEEE 1149.1 Boundary Scan Standard that is used primarily to test printed circuit boards (PCBs) and multi-chip modules (MCMs).
- **Delay faults and testing** –deep submicron circuits exhibit several unique fault modes, some of which lead to excess signal delay. Special test generation and simulation techniques are required to address these issues
- **Memory tests** – of course a significant part of many chips consists of cache and RAM. Because of their regularity in layout and circuitry, they exhibit special fault modes and are conducive to special test procedures. Many of these issues will be covered in this module.
- **Fault diagnosis** – while a process is still immature and a design is being debugged, it is important to identify the reason why chips that do not work properly. This problem falls partly under the area of fault diagnosis, (and partly under verification). We will investigate several techniques for narrowing down the search to locate faults in a circuit.
- **IDDQ testing** – while most circuits are tested by applying input test patterns and observing the responses, in some cases we can detect certain faults, such as shorts, by applying tests and measuring the quiescent current in the power grid. A higher than normal current usually indicates some type of a problem.
- **Fault tolerant system** – Often it is important that a system be very reliable, such as if it were in control of a power plant or an airplane. In this case a fault might occur while the system in operational, but we might still want to system outputs to be correct. To achieve this, techniques exists that support fault-tolerant computing. These include the use of error correcting codes and redundant modules.

No complex mathematical concepts are used in this class, but you better have a *solid background* in digital logic design and programming.

III. Prerequisites:

The course assumes you are **very familiar** with logic design (EE 101) and computer programming, preferably C or C++. Knowledge of EE 477, 552, 457 and CSCI 455 is helpful but not essential. While you all think you know logic design, in reality you do not know as well as you need to in order to pass this course. In addition, if you have little experience in programming, then you will have trouble doing the project. Your progress on the project will be monitored by the TA who will quickly identify who is a beginning programmer.

IV. Course Goals:

To give a student a broad overview of the key technical concepts related to testing of VLSI circuits. Students will have the opportunity to understand and apply key algorithms to problems, and, to a small extend, develop new algorithms. Students will be briefly introduced to some of the commercial tools that exist in the area of test. By understanding these idea, students can either carry out future work in the area of (1) **design**, where they can effectively use test tools and/or enhance their designs using DFT or BIST concepts, (2) **tool development**, where they work on CAD/CAE projects developing advanced tools for design and test engineers, or (3) **test**, where they are responsible for developing high quality tests for circuits.

V. Learning Objectives:

Students are expected to be able to understand, analyze, implement, evaluate, extend, and apply the concepts presented in this course. They must also be able to demonstrate their abilities in these areas via exams, homework and projects.

VI. Computer Engineering Curriculum:

This course, EE658, is part of the Computer Engineering curricula in Computer Aided Design (CAD). We have four courses in this area, namely EE658, EEXXX CAD-I –for MS and PhD students, EE581 CAD-II – mathematical background for advanced CAD, EE 680 CAD-III – advance topics in physical design, and EE681 CAD-IV – advanced topics in synthesis and verification. This material is an excellent complement to the VLSI courses, such as EE577a, b.

Lecture schedule

Days	Lecture topic (approximate)	You should read	Misc.
5/19	Introduction to the course	Cht. 1 + 101 notes	Detail review of 101
24	TA lecture on C++ and data structures	Posted notes	Detail review of 101
26	End of Introduction	Cht. 2	Review C++
31	Defects, faults and errors	Posted notes	Review C++
6/2, 7	Delay, hazards and logic simulation	Cht. 3, 4	Review C++
9, 14	Fault simulation	Cht. 5	C++ programming exercise
16, 21	Test generation	Cht. 6	
6/23	**Project description**		**Set up groups**
28	**Midterm** & Fault diagnosis	Cht. 12	
30, **7**/5	Design-for-test (DFT)	Cht. 9	
7/5			**Initial project report**
7, 12	Compression and Built-in self-test (BIST)	Cht. 10 & 11	
14	Delay testing	Posted notes	**Detailed project algorithmic report**
19	Memory testing	Posted notes	
21	Project solution ideas		
7/26	**FINAL**		**Coding completed** **Last day of classes**
29			**Debugging completed**
29-30			**Run individual test cases and document results**
31, 8/1			**Integration completed**
8/2			**2-hour demo/group**
4			**Final project report due with results for test cases**

Orange items refer to the project.

As the semester progresses, more details will be provided concerning what parts of chapters need not be read. In addition, supplemental material will be provided via the web. In addition, students should spend about one - two hour **before** class reading any posted notes and chapter assignments relevant to the lecture to be given.

Supplemental Material

Much supplemental information will be made available to students via the course web site. Books that cannot be made available this way can be found in libraries. Below is a somewhat comprehensive list of books relevant to this course.

Books:

- M. Abramovici, M. A. Breuer and A. D. Friedman, Digital system testing and testable design, John Wiley and Sons, Publishers, 1990. **TK7874.A23**
  This book is designed for use as a text for graduate students, covering basics of fault modeling, simulation, and testing, design for testability, built-in self-test, logic and system level diagnosis, and PLA testing.
- J. Altet and A. Rubio, Thermal testing of integrated circuits, Kluwer Academic Publishers, 2002. **TK7874.A44**
  It offers a multidisciplinary focus on thermal testing. The techniques can be applied either to the packaging of the components, or directly to the components themselves.
- A.P. Ambler, M. Abadir and S. Sastry, Economics of design and test, Ellis Horwood Limited, 1992.
  This text contains the extended versions of papers presented at the 1st International Workshop on the Economics of Design and Test. The focus of this workshop was the economic analysis of design and test of digital systems.
- T. Anderson and P.A. Lee, Fault Tolerance: principles and practice, Prentice-Hall, 1981. **QA 76.9F38A53**.
  Primarily dealing with fault tolerance.
- P.H. Bardell, W.H. McAnney and J. Savir, Built-in test for VLSI – pseudorandom techniques, John Wiley and Sons, 1987. **TK 7874.B374**.
  Excellent text dealing with theoretical issues of BIST.
- F. P. M. Beenker, R. G. Bennetts and A. P. Thijssen, Testability concepts for digital ICs: The macro test approach, Kluwer Academic Publishers, 1995. **TK7874.65.B44**
  It reflects the activities on testability concepts for complex digital ICs as performed at Philips Research Laboratories in Eindhoven, The Netherlands.
- R. G. Bennetts, Design of testable logic circuits, Addison-Wesley Publishing Co., 1984. **TK 7868.L6B45**.
  Basic text covering simulation, test generation, DFT and BIST. Good practical information.
- R. G. Bennetts, Introduction to digital board testing, Crane Russak, N.Y., 1982.
  Engineering oriented book on testing of devices.
- M. A. Breuer and A.D. Friedman, Diagnosis and reliable design of digital systems, Computer Science Press, 1976. **TK 7868.D5B73**.
  This is a classic text on testing but a little out of date. No information on DFT or BIST.
- M. Burns and G. W. Roberts, An introduction to mixed-signal IC test and measurement, Oxford University Press, 2001. **TK7874.B825**
  It contains the basic course material for mixed-signal test and measurement, accuracy, resolution, mixed-signal ATE tester, ADC and DAC sampling theory, etc.
- M. D. Bushnell and V. D. Agrawal, essentials of Electronic Testing, Springer, 2000, ISBN 0-7923-7991-8, **TK7874.75.B87**
- S. T. Chakradhar, V. D. Agrawal and Michael L. Bushnell, Neural models and algorithms for digital testing, Kluwer Academic Publishers, 1991. **TK7868.L6C44**
  Neural networks, with their evolution-like computing capability, show hope for massive parallelism. This book proposes a new modeling technique for the purpose of test generation for digital logic circuit.

- H. Y. Chang, E. Manning and G. Metze, Fault diagnosis of digital systems, Wiley-Intersciences, 1970.
  Early book on a specific system for test generation and simulation.
- J. Coffron, Using and trouble-shooting the MC68000, Reston Publishing Co, 1983.
- J. Coffron, Using and trouble-shooting the Z-8000, Reston Publishing Co., 1982.
  Testing a specific system.
- L. Crouch, Design-for-test for digital IC's and embedded core systems, Prentice Hall PTR, 1999.
  The book contains five chapters covering the topics of test, ATPG, scan, memory test, and cores. It is written by a person from industry that has been working on test issues for many years.
- R. David, Random testing of digital circuits, theory and applications, Marcel Dekker, Inc., 1998.
  This book is based on the author's work in the area of random pattern testing. Some analytical tools have been developed in this book for random pattern testing.
- R.J. Feugate, Jr. and S.M. McIntyre, Introduction to VLSI testing, Prentice-Hall, 1988, **TK 7874.F48**.
  Very practical discussion on many test issues related to testing VLSI chips.
- A.D. Friedman and P.R. Menon, Fault detection in digital circuits, Prentice-Hall, 1971.
  First general text on testing. Many topics here are not covered elsewhere.
- H. Fujiwara, Logic testing and design for testability, The MIT Press, 1985. **TK7868.L6F85**.
- D. Gizopoulos (ed.), Advances in electronic testing: Challenges and methodologies, Springer, 2006, ISBN 10 0-387-29408-2
  Good text covering test and DFT and BIST, but no problem set.
- N. K. Jha, Testing and reliable design of CMOS circuits, Kluwer Academic Publishers, 1990. **TK7871.99 M44J49**.
  This book has been written as a reference text for courses on digital system testing and fault-tolerant computing offered at the senior or graduate level.
- N. K. Jha and S. K. Gupta, Testing of digital systems, Cambridge University Press, 2003. ISBN 0 521 77356 3
  Excellent and modern text covering a large area of test technology and in great depth. Some complex notation.
- B. W. Johnson, Design and analysis of fault tolerant digital systems, Addison-Wesley Publishing Company, Inc., 1989. **QA76.9.F38J64**
  This book is an introductory text and can be of immediate use to individuals with no exposure to fault-tolerant computing. It includes 13 examples of fault-tolerant systems.
- J. B. Khare and W. Maly, From contamination to defects, faults and yield loss, simulation and applications, Kluwer Academic Publishers, 1996. **TK7874.75.K47**
  The objective of this book is to challenge the traditional approach to VLSI problems by discussing the core of the interface between manufacturing and testing, i.e., the contamination-defect-fault relationship.
- P. Lala, Fault tolerance and fault testable hardware design, Prentice-Hall 1985. **TK 7888.3.L27**.
  Good overview of testing and fault tolerance, but no problem sets.
- H. V. D. Linden, Automatic test pattern generation for three-state circuits, Ph.D. thesis, University of Delft, 1996. **ISBN 90-9009585-3.**
  The main subject is ATPG for structural production testing to detect single stuck-at faults in combinational or scan-based digital logic circuits.
- P. Mazumder and K. Chakraborty, Testing and testable design of high-density random-access memories, Kluwer Academic Publishers, 1996. **TK7895.M4 M38**
  This is a basis text to learn about memory technology and related concepts of testing and design for testability.
- E. J. McCluskey, Logic design principles, Prentice-Hall, 1986, **TK 7868.L6M38**.

Basic undergraduate text on logic design. Chapter 10 deals with design for testability.

- A. Miczo, Digital logic testing and simulation, Harper and Row, 1986**. TK 7868.D5M49**.
  Nice text covering many areas with problems.
- S. Mourad and Y. Zorian, Principles of testing electronic systems, John Wiley and Sons Inc., 2000. **TK7867.M697**
  Physical defects, test pattern generation for detecting faults, the relationship of testing to the design cycle and the design for test practice are detailed in the book.
- B. Nadeau-Dostie, Design for at-speed test, diagnosis and measurement, Kluwer Academic Publishers, 2000. **TK7874.D47 497**
  The basic theme of this book is that embedded test offers significant reductions in design and test engineering efforts, time to market, and cost of design and test of chips, boards, and systems.
- K. P. Parker, Integrating design and test: Using CAE tools for ATE programming, IEEE Computer Society Press IEEE, 1987.
  Practical information on test program development and ATE.
- J. Rajski and J. Tyszer, Arithmetic built-in self-test for embedded systems, Prentice Hall PTR, 1998. **TK7895.E42 R35**
  This book is based on the results of research in arithmetic built-in self-test (ABIST). It has a detailed survey of DFT and BIST schemes, as well as the ABIST techniques and building blocks, and ABIST applications.
- R. Rajsuman, Digital hardware testing – Transistor-level fault modeling and testing, Artech House, Inc., 1992.
  This book is devoted to fault models, complexity of testing problem, combinational and sequential circuit testing, and IDDQ testing.
- P. Roth, Computer logic, testing, and verification, Computer Science Press, 1980, **TK7888.4.R67.**
- J. M. Schoen, Performance and fault modeling with VHDL, Prentice Hall, Inc. 1992. **TK7888.4.P47**
  The specific areas covered in this book include system performance modeling in the presence of multiple processors and/or component failures, and the effect of various mappings of algorithms to multiple processors on overall system effectiveness.
- D. P. Siewiorek and R.S. Swarz, The theory and practice of reliable system design, Digital Press, 1982.
  Thorough treatment of fault tolerant systems.
- C. C. Timoc, ed., Selected reprints on Logic Design for Testability, IEEE Society Press, IEEE Catalog No. EH0215-4.
- F. F. Tsui, *LSI/VLSI testability design*, McGraw-Hill, 1987, **TH 7874, T78**.
  Lots of information on scan design concepts developed at IBM.
- L-T Wang, C-W Wu and X. Wen, VLSI test principles and architectures: Design for testability, Morgan Kaufman, 2006. ISBN 978-0-12-307597-6, **TK7874.75.V587.**
  A very comprehensive and up-to-date book.
- L-T Wang,, C. Stroud, Nur Touba, System on chip test architectures, Morgan Kaufmann-Nov, 2007.
- T. W. Williams, Editor, VLSI testing, Vol. 5 of Advances in CAD for VLSI, T. Ohtsuki editor, North-Holland, 1986.
- V. N. Yarmolik and S. N. Demidenko, Generation and application of pseudorandom sequences for random testing, John Wiley and Sons, 1988.
  This text deals with the theoretical and practical problems of generating pseudorandom signals and their use in automatic testing and verification systems for complex engineering systems.

Other major sources of material on testing can be found in the publications shown below.

Annual Conference Proceedings

- Proc. International Test Conference (ITC)
- Proc. Fault Tolerant Computing Conference (FTC)
- Proc. Design Automation Conference (usually has 2 or 3 sessions on testing) (DAC)
- Proc. VLSI Test Symposium (VTS)
- Proc. Asian Test Symposium (ATS)
- Proc. European Design Automation and Test Conference (DATE)
- Proc. Int'l. Conference on Computer Aided Design (ICCAD)

Miscellaneous Journals

- IEEE Trans. on Computers
- IEEE Trans. on Computer-Aided Design-Circuits and Systems
- IEEE Design and Test of Computers
- Journals of Electronic Testing: Theory and Applications (JETTA)

Trade Magazines

- TEST, (including Journal of ATE)
- Electronics Test

Academic Dishonesty Sanctions

(If there is any discrepancy between what is stated below and official University policy, then University policy take precedence.)

Violation	**Probable Sanctions**
Copying answers from other students on exam.	F for course
One person allowing another to cheat from his/her exam or assignment.	F for course for both persons
Possessing or suing material during exam (crib sheets, notes, books, etc.) which is not expressly permitted by the instructor.	F for course
Continuing to write after exam has ended.	F or zero on exam
Taking exam from room and later claiming that the instructor lost it.	F for course and recommendation for further disciplinary action (possible suspension).
Changing answers after exam has been returned.	F for course and recommendation for further disciplinary action (possible suspension).
Fraudulent possession of exam prior to administration.	F for course and recommendation for suspension.
Obtaining a copy of an exam or answer key prior to administration.	Suspension or expulsion from the university; F for course.
Having someone else take an exam for oneself.	Suspension or expulsion from the university for both students; F for course.
Plagiarism	F for course
Submission of purchased term papers or papers done by others.	F for course and recommendation for further disciplinary action (possible suspension).
Submission of the same term papers to more than one instructor, where no previous approval has been given.	F for both courses.
Unauthorized collaboration on an assignment.	F for the course for both students.
Falsification of information in admission applications (including supporting documentation).	Revocation of university admission without opportunity to reapply.
Documentary falsification (e.g., petitions and supporting materials; medical documentations).	Suspension or expulsion from the university; F for the course when related to a specific course.

Dear Faculty Member,

As we enter a new academic year, I am writing in anticipation of your interest in maintaining high academic integrity standards for your courses and the university. The entire academic community benefits from the establishment and adherence to such standards.

Resources on our Student Judicial Affairs Web site (http://www.usc.edu/student-affairs/SJACS) include two student-oriented publications in both viewable and printable forms:

1. "Guide to Avoiding Plagiarism" addresses issues of paraphrasing, quotations and citations in written assignments, drawing heavily upon materials used in the university's Writing Program;

2. "Understanding and Avoiding Academic Dishonesty" addresses more general issues of academic integrity, including guidelines for adhering to standards concerning examinations and unauthorized collaboration.

Returning students as well as those entering the university for the first time can benefit from clear statements about the academic standards for your class. If either of these publications is useful as a supplement to your instructions for assignments and your course syllabi, it can be printed from or viewed on our Web site.

The "2005-2006 SCampus" (the student handbook) will be available on line by the first week of classes (http://www.usc.edu/scampus). It contains the university's Student Conduct Code and other student-related policies that will be of importance to you. For your convenience, our Web site also includes a link to an electronic version of Appendix B to the Student Conduct Code (academic integrity violation report form).

Please contact our office if you have any questions or wish consultation regarding academic honesty or student behavior in your classes.

Sincerely,

Raquel Torres-Retana
Director
Student Judicial Affairs and Community Standards