

Course Syllabus
14:332:437 Concepts in Digital Systems Design
Fall 2007
Electrical and Computer Engineering Department
School of Engineering
Rutgers University

Prof. Michael L. Bushnell

August 24, 2007

PHILOSOPHY: This course presents a systematic approach to designing digital logic circuits for file servers, engineering work stations, defense electronics, personal computers, and telecommunications. At the end of this course, you will be able to:

1. Describe and design computer hardware using the Verilog *Hardware Description Language* and a *Field-Programmable Gate Array Chip*.
2. Rapidly design combinational logic that works.
3. Rapidly design complex state machines (which are present in all practical computers) that work.
4. Design logic and state machines using an Automatic Logic Synthesis program.
5. Implement state machines using Field-Programmable Gate Arrays.
6. Design high-speed computer arithmetic circuits.
7. Design a computer to be fault-tolerant (to continue operating in the presence of hardware failures).
8. Design a computer memory using error-correcting codes.
9. Design a computer so that it can test itself with built-in circuitry.

COURSE DESCRIPTION: Hardware description using the Verilog language. Systematic MUX-based combinational and sequential logic design, including variable-entered Karnaugh maps, advanced state machine theory (state assignment, state minimization, meta-stable state detection and correction, and handling of asynchronous inputs), and design for testability. Computer arithmetic circuits. Synthesis of combinational and sequential circuits to account for timing delays. Fault-tolerant computer design, including triple-modular redundancy, and machine organization for testing and fault-tolerance. Clock organization and distribution. Redundant logic elimination. Hamming error correcting codes for memories and busses and built-in self-testing circuits.

LECTURER: Mike Bushnell
Office: Room 624, CORE Bldg.
Telephone: (732) 445-4854
FAX: (732) 445-4775
Email Address: bushnell@caip.rutgers.edu
Office Hours: Tues. and Thurs., 1:30-4:30 p.m.
Lecture Times: Mon., Wed. 3:20-4:40
Classroom: SEC-209
Course Laboratory: CORE 533 & EE 103 UNIX Work Station Room
Teaching Assistants: Srihitha Yerabaka
CORE 605
srihitha@eden.rutgers.edu
(732) 318-9768

Office Hours:

PREREQUISITES:

- 332:231 Digital Logic Design
- 332:233 Digital Logic Design Lab.
- 332:331 Computer Architecture and Assembly Language
- 332:333 Computer Architecture Lab.

TWO-TERM SEQUENCE: This required course is part of a two-term sequence. The second term course, 332:438 *Capstone Design – Digital Systems*, is a pure design course, with no academic content, and the combination of the 332:437 and 332:438 courses satisfies the ECE Dept. capstone design course requirement.

TEXTS:

- *Self-Checking and Fault-Tolerant Digital Design*, by Lala, Academic Press/HBJ, ISBN # 0-12-434370-8.
- *The Verilog Hardware Description Language*, by Don Thomas and Philip Moorby, Kluwer Academic Publishers, 5th Edition, 2003, ISBN # 1-4020-7089-6.
- Optional Text: *Logic Design & Switching Theory*, by Muroga, Krieger Publishing Co., ISBN # 1-57524-036-X.
- Optional Text: *Digital Design Principles and Practices*, by John F. Wakerley, Prentice Hall.
- Optional Text: *Essentials of Electronic Testing for Digital, Memory, & Logic VLSI Circuits*, 1st Edition, by Michael L. Bushnell and Vishwani D. Agrawal, Springer, 2000, ISBN # 0-7923-7991-8.

COMPUTER USAGE: Students will use the Synopsys logic synthesis system to design and simulate hardware described in the Verilog hardware description language.

GRADING:

Homeworks 1-10 (these enable you to do the exams)	1.25 % each	12.5 %
Practicums 1-4	3.125 % each	12.5 %
Examination 1		22 %
Examination 2		23 %
FINAL EXAM		30 %

POLICIES:

1. Constructive suggestions from students about this course will be greatly appreciated.
2. STUDENTS ARE EXPECTED TO WORK INDEPENDENTLY ON BOTH HOMEWORK AND EXAMINATIONS. NO COLLABORATION IS PERMITTED, UNLESS YOU ARE OTHERWISE INSTRUCTED. YOUR WORK MUST BE YOUR OWN. We will report you to the Deans for cheating if we find that you work is the same as another student's work.
3. Statute of limitations policy: You have one week after the day each homework or examination is returned to you to ask for a grade change. Grades will be changed only during this week and only if:
 - (a) We added up your score wrong.
 - (b) We made an error in grading a particular problem.

At the end of the term, we will not argue about points on the homeworks or on the exams.

4. If you are ill on the date of an examination, or if you must be away, please notify the professor BEFORE the examination, if possible, or at least as EARLY as possible. Medical slips for absences from the examination are required if you were ill or unavoidably absent.
5. All course materials are available on the course web site:
<http://www.caip.rutgers.edu/~bushnell>
6. Homework is due at the beginning of lecture on the due date. Late homework gets a 0 grade. The only exceptions are for documented accidents, illnesses, or deaths in the family.
7. Do your lab work early. Most students let lab work wait until the night before it is due, and then are unable to get a computer since they are all taken.
8. In order to control cheating problems in this course, we have introduced 4 Practicums. Half of your homework grade is from the written homework that you hand in, and half is from the 4 Practicums. A Practicum will be held in the CORE 601A and CORE 533 UNIX laboratories by appointment. You will have one hour to complete an assignment under the supervision of the Professor or TA in the lab. You must sign up for appointments to do Practicums.

ASSUMED BACKGROUND

- Boolean Algebra
- Combinational and Sequential Circuit Logic Design
- All kinds of flip-flops and latches

- Electrical Network Theory
- C and/or C++ Language Programming
- RISC Computer Architecture

If you are unfamiliar with any of these concepts, remedial study may be necessary.

**PRELIMINARY HOMEWORK SCHEDULE –
SUBJECT TO CHANGE**

No.	Description
1	VEM Logic Design, MUX-Based Design, Triple-Modular Redundancy
2	Verilog Combinational Logic and Memory Design
3	State Machine Design with Verilog
4	Verilog Counter and State Machine Design
5	State Reduction and Synchronizer Design
6	Large State Machine Design with Verilog
7	Verilog Arithmetic Circuit and Memory System Design
8	Verilog Error Correcting Code Circuit Design
9	Random Logic Built-in Self-Testing (BIST)
10	Field Programmable Gate Array Design

**PRELIMINARY EXAMINATION SCHEDULE –
SUBJECT TO CHANGE**

Exam 1 – Oct. 24, 2007

Exam 2 – Nov. 28, 2007

COURSE OUTLINE

#	Description	Module	Date	Reading
	FAULT TOLERANCE			
1	History of Fault Tolerance – Motivation for it	1 (40)	9/5	Lala, Ch. 1
2	Fault-Tolerance Examples	2 (33)	9/10	Lala, Sect. 6.1 & 6.4
	ADVANCED COMBINATIONAL LOGIC DESIGN			
3	Hardware Design Methodology and Advanced Logic Design	3 (23)	9/12	Lecture Notes
4	Variable-Entered Karnaugh Maps and Mixed-Logic Notation	4 (33)	9/17	Muroga
	VERILOG			
5	Verilog Hardware Description Language Basics Verilog Tutorial	5 (19) & 6 (24)	9/19	Thomas, Ch. 1
6	Verilog Behavioral Modeling and Concurrency	7 (32)	9/24	Thomas, Ch. 2 & 3
7	Verilog and Finite State Machines	8 (23)	9/26	Thomas, Ch. 4 & 6
8	Verilog Example	9 (31)	10/1	Thomas, App. A-F
9	Verilog Event-Driven Simulation	10 (41)	10/3	Thomas, Ch. 8
10	Verilog Language Details	11 (29)	10/8	Thomas, App. A-F
	STATE MACHINES			
11	Finite State Machine Design	12 (47)	10/10	Muroga
12	Finite State Machine Asynchronous Inputs, Clocks, and Hazards	13 (45)	10/15	Muroga
13	Turing Machines and State Machine Sequences	14 (18)	10/17	Lecture notes
14	System Controller Design	15 (18)	10/22	Lecture notes
15	EXAM 1		10/24	
16	Finite State Machine Synchronizers	16 (18)	10/29	Lecture notes
17	FSM Hardware Modification for Reliability Advanced Synchronizers	17 (20) & 18 (18)	10/31	Muroga
	FAULT TOLERANCE (cont'd.)			
18	Time Redundancy	19 (31)	11/5	Lala, Sect. 6.3
19	Information Redundancy – Parity and Checksums	20 (40)	11/7	Lala, Ch. 2
20	Cyclic and Hamming Codes	21 (50)	11/12	Lala, Ch. 2
21			11/14	Lala, Sect. 6.2
	COMPUTER ARITHMETIC			
22	Computer Arithmetic, Booth & Wallace Tree Multipliers/Dividers	22 (29)	11/19	Lecture Notes
	CIRCUIT TESTING			
23	Introduction to Testing	23 (32)	11/26	Lecture Notes
24	EXAM 2		11/28	
25	Testing Methods	24 (39)	12/3	Lecture Notes
26	Built-in Self Testing	25 (26)	12/5	Lecture Notes
27	IEEE Boundary Scan Standard and System Test	26 (32)	12/10	Lecture Notes
	PACKAGING & MICROPROCESSOR DESIGN			
28	Packaging and Microprocessor CORE Organization	27 (5)	12/12	Lecture notes
29	FINAL EXAM			

REFERENCE BOOKS:

- Verilog Books:
 - *Verilog HDL*, by Samir Palnitkar, Prentice Hall, 2003, ISBN # 0-13-044911-3.
 - *Advanced Digital Design with the Verilog HDL*, by Michael D. Ciletti, Prentice Hall, 2003, ISBN # 0-13-089161-4.
- Fault Tolerance and Testing:
 - *Design and Analysis of Fault Tolerant Digital Systems*, by Barry W. Johnson, Addison Wesley (out of print). A great tutorial book on fault tolerance and testing.
- Architecture:
 - *Structured Computer Organization*, by Andrew S. Tanenbaum, Prentice Hall, 1990.
- Finite State Machines:
 - *Switching and Finite Automata Theory*, by Kohavi, McGraw-Hill, 1978. THE classic book on finite state machines.
- Programmable Logic:
 - *Digital Systems Design with Programmable Logic*, by Martin Bolton, Addison Wesley, 1990. A very well-written book that is, unfortunately, quite out of date because it does not mention field-programmable gate arrays.
- Digital Logic Design:
 - *Digital Design – Principles and Practices*, 2nd Edition, by John F. Wakerly, Prentice-Hall, 1994. A good introduction to logic design.
 - *Modern Digital Systems Design*, by John Cheung and Jon Brederson, West Publishing Co.