



## ECE 257A: Information on Previous Offerings

Behrooz Parhami: 2007/09/20 || E-mail: parhami at ece.ucsb.edu || Other contact info at: [Bottom of this page](#)  
Go up to: [B. Parhami's course syllabi](#) or [his home page](#)

[Link to the most recent offering of ECE 257A: Fault-Tolerant Computing](#)

### Previous offerings of the course

- [ECE 257A: Fall Quarter 2006](#)
- [ECE 257A: Fall Quarter 1998](#)

Return to: [Top of this page](#)

### ECE 257A: Fall Quarter 2006 offering

**This area reserved for important course announcements:** If you have any last-minute questions about homework or the course material, I will be in my office all morning (9:00-12:00) on Friday, 12/8. Per our agreement in class on Thursday 11/30, the course's final exam will be on Friday, December 8, 1:00-3:00 PM, in Room 4164 (ECE Conference Room) in Harold Frank Hall.

- Course:** ECE 257A – Fault-Tolerant Computing, University of California, Santa Barbara, Fall 2006, Enrollment Code 49585
- Catalog entry:** **257A. Fault-Tolerant Computing. (4) PARHAMI.** *Prerequisite: ECE 154. Lecture, 4 hours.* Basic concepts of dependable computing. Reliability of nonredundant and redundant systems. Dealing with circuit-level defects. Logic-level fault testing and tolerance. Error detection and correction. Diagnosis and reconfiguration for system-level malfunctions. Degradation management. Failure modeling and risk assessment. (F)
- Instructor:** Behrooz Parhami, Room 5155 Harold Frank Hall (Engr I), Phone 805-893-3211, [parhami@ece.ucsb.edu](mailto:parhami@ece.ucsb.edu)
- Meetings:** Tuesdays and Thursdays, 10:00-11:30 AM, Phelps 1431
- Consultation:** Open office hours, held in Room 5155 Harold Frank Hall (Engr I) – Tuesdays 11:30-1:00, Thursdays 8:30-10:00
- Motivation:** Dependability concerns are integral parts of engineering design. Ideally, we would like our computer systems to be perfect, always yielding timely and correct results. However, just as bridges collapse and airplanes crash occasionally, so too computer hardware and software cannot be made totally immune to unpredictable behavior. Despite great strides in component reliability and programming methodology, the exponentially increasing complexity of integrated circuits and software systems makes the design of perfect computer systems nearly impossible. In this course, we study the causes of computer system failures (impairments to dependability), techniques for ensuring correct and timely computations despite such impairments, and tools for evaluating the quality of proposed or implemented solutions.
- Prerequisites:** Basic computer architecture at the level of ECE 154.
- References:** **Required textbook** – None (class handout or reference will be provided before each lecture)  
**Other useful books, not required** –  
Pradhan, D.K. (ed.), *Fault-Tolerant Computer System Design*, Prentice-Hall, 1996. [out of print, as of 9/2006]  
Siewiorek, D.P. and R.S. Swarz, *Reliable Computer Systems: Design and Evaluation*, Digital Press, 2nd ed., 1992.  
Johnson, B.W., *Design and Analysis of Fault-Tolerant Digital Systems*, Addison-Wesley, 1989.  
Lala, P.K., *Self-checking and Fault-Tolerant Digital Design*, Morgan Kaufmann, 2001.  
Shoorman, M.L., *Reliability of Computer Systems and Networks*, Wiley, 2002  
**Journals** – *IEEE Trans. Dependable and Secure Systems* (new, since 2004), *IEEE Trans. Computers*, *IEEE Trans. Reliability*, *IEEE Trans. Software Engineering*, *ACM Trans. Computer Systems*, and *Information Processing Letters*. Also, *IEEE Computer*, *IEEE Micro*, *IEEE Design & Test of Computers*, and *ACM Computing Surveys* are good sources for broad introductory papers.  
**Conferences** – Int'l Conf. Dependable Systems and Networks (DSN, annual, since 1971; formerly known as FTCS), Pacific Rim Int'l Symp. Dependable Computing (PRDC, since 1989), IFIP Int'l Working Conf. Dependable Computing for Critical Applications (DCCA; discontinued and merged with FTCS to form DSN), Int'l Symp. Software Reliability Engineering (ISSRE, annual, since 1990).  
**Electronic resources at UCSB** – Journals and conference proceedings listed above, as well as many other useful references, can be accessed electronically via:  
<http://www.library.ucsb.edu/eresources/databases/> (electronic journals, collections, etc.)  
<http://www.library.ucsb.edu/subjects/engineering/ece.html> (research guide in ECE)
- Evaluation:** Students will be evaluated based on three components, with the given weights:  
20% -- Homework assignments (see the course calendar for general requirements and schedule)  
40% -- Open-book/notes midterm exam (see the course calendar for date and coverage)  
40% -- Open-book/notes final exam (see the course calendar for date and coverage)
- Research:** An optional research paper may be substituted for the final exam. A student interested in this option, will review a subfield of dependable computing or do original research on a selected topic. A [list of research topics](#) is provided below; however, please feel free to propose your own topic for approval. A publishable report earns an "A" for the course, regardless of homework and midterm grades. See the "deadlines" column in course calendar for schedule and due dates.

### Lecture plan:

Day & Date	Lecture	Lecture Topic	References	Deadlines and Notes
R 9/28	1	Introduction, motivation, and course goals	Links appeared here	

T 10/03	2	Dependability measures	Links appeared here	HW#1 posted (Lectures 1-4)
R 10/05	3	Combinational modeling	Links appeared here	
T 10/10	4	State-space modeling	Links appeared here	Research topic defined
R 10/12	5	Defect avoidance and circumvention	Links appeared here	HW#1 due (deadline extended)
T 10/17	6	Fault testing	Links appeared here	HW#1 due, HW#2 posted (Lectures 5-8)
R 10/19	7	Fault masking	Links appeared here	
T 10/24	8	Error detection	Links appeared here	
R 10/26	9	Error correction	Links appeared here	HW#2 due; Preliminary references due
T 10/31		Instructor away at conference (no lecture)		
R 11/02	10	Malfunction diagnosis and tolerance	Links appeared here	
T 11/07	1-9	Midterm exam: open-book (10:00-11:50 AM)		Held in class; note extended time
R 11/09	11	Degradation allowance and management	Links appeared here	HW#3 posted (Lectures 9-12)
T 11/14	12	Failure confinement	Links appeared here	Paper title and references due
R 11/16	13	Hardware implementation strategies	Links appeared here	
T 11/21	14	Self-checking modules	Links appeared here	HW#3 due, HW#4 posted (Lectures 13-15)
R 11/23		Thanksgiving holiday (no lecture)		
T 11/28	15	Reconfiguration and voting	Links appeared here	Paper abstract and outline due
R 11/30	16	Software reliability and redundancy	Links appeared here	
T 12/05	17	Algorithm design methods	Links appeared here	HW#4 due
R 12/07	18	Agreement and adjudication	Links appeared here	
F 12/08	10-18	Final exam: open-book (1:00-3:00 PM)		Held in 4164 HFH (Engineering I), Complete paper due

### Homework: General Requirements

Solutions should be brought to class on the due date and handed in before the lecture begins. Late homework will not be accepted, so plan to start work on your assignments early. Use a cover page that includes your name, course and assignment number for your solutions. Staple the sheets and write your name on top of every sheet in case sheets are separated. Although some cooperation is permitted, direct copying will have severe consequences.

### ECE 257A f2006, Homework #1: Dependability measures and modeling (Lectures 1-4, due Thursday, Oct. 12)

- [Mean time to failure; 10 points] A particular computer model has a constant failure rate of  $\lambda$ . If a large number  $m$  of identical machines of this type run continuously for a period of time equal to their MTTF, how many of them are expected to be still working at the end? (Hint: The correct answer isn't  $m/2$ .) By what time do we expect 90% of the machines to have failed?
- [Failure independence; 15 points] A department has three networked servers. The maintenance logs for the month of September indicate that all three servers were up and running 83% of the time, two were running (while the third one was down) 16% of the time, and only one was running at all other times. Show that the logged data are consistent with, but do not prove, the hypothesis that the three servers are equally reliable and fail independently.
- [Data availability with triplication; 25 points] Redo example 1.2 of the first course handout (Oct. 5), quantifying the probability of being able to access a data file if, whenever link malfunctions make direct access impossible, the requesting site can obtain a copy indirectly via the remaining site that does not itself hold a copy of the data, provided the latter is operational and can access the required file. State your assumptions clearly. Because the availability expression for this problem can grow quite complicated, use the parameters  $L$  and  $S$  in lieu of  $a_L$  and  $a_S$  for brevity.
- [Combinational modeling; 10 points] A safety critical subsystem is built of three computation modules, each having the reliability  $r_M$ , and a voter with reliability  $r_V$ . Represent this system as a series-parallel system. Verify that your model is indeed correct by computing its reliability and comparing to that of the original system.
- [State-space modeling; 20 points] A computer system experiences one failure per 1000 hours of operation. Upon failure detection, a diagnostic process is run. In 95% of the cases, the cause of failure can be removed based on the diagnostic (e.g., through system reset or software reload) and service is restored in about 15 minutes. In the remaining 5% of the cases, a repair person must be summoned and the expected time to service restoration is around 12 hours. Construct a state-space model for this system and derive its availability.
- [Reading assignment; 20 points] Read the paper "Evaluation of Software-Implemented Fault-Tolerance (SIFT) Approach in Gracefully Degradable Multi-Computer Systems" by Avresky, Geoghegan, and Varoglu (*IEEE Trans. Reliability*, Vol. 55, No. 3, pp. 451-457, September 2006). Explain the model shown in Fig. 1 of the paper in terms of its states (e.g., why are there pairs of primed and unprimed states?) and transition rates.

### ECE 257A f2006, Homework #2: Dealing with low-level impairments (Lectures 5-8, due Thursday, Oct. 26)

- [Bathtub curve; 15 points] Argue that, with proper interpretation, the hardware failure rate trend represented in a bathtub curve is applicable to software as well. Take time  $t = 0$  to represent the completion of the first fully functional prototype of the software, which is then followed by various pre-release versions and, eventually, the first released version, with the latter ideally occurring at the end of infant mortality. Reference for the aging phase: Grotke, Michael, Lei Li, Kalyanaram Vaidyanathan, and Kishor S. Trivedi, "Analysis of Software Aging in a Web Server," *IEEE Trans. Reliability*, Vol. 55, No. 3, pp. 411-420, September 2006.
- [Fault testing; 15 points] Find a minimal set of tests for all s-a-0 and s-a-1 faults in the full-adder circuit shown in slide 12 of lecture 6.
- [Testability; 15 points] Quantify the testability of each line of the full-adder circuit shown in slide 12 of lecture 6. Based on the results obtained, where would you place a single testpoint?
- [Modeling for fault masking; 15 points] Consider the following fault-masking redundancy scheme with voting. There are four modules. Initially, three modules are operating concurrently and one is a spare. When a disagreement is detected between the three operating modules, the disagreeing module is purged (switched out) and the spare is used to replace it. After the spare has been switched in, a detected disagreement causes the disagreeing module and one of the good ones to be purged, leaving a simplex system. State all your assumptions clearly. Construct and solve a state-space reliability model for this system, assuming perfect coverage, switching, and voting.
- [Switch-voter design for fault masking; 20 points] For the redundancy scheme discussed in Problem 4, design the required 4-input, 1-output voter-switch, with 4 internal FFs indicating which units are contributing to the formation of the output. Feel free to use of muxes, comparators, and other standard circuits in your design (the design need not be at the gate level).
- [Error detection; 20 points] A multibit unidirectional error affects the bits in the same direction; that is, all changes are either 0-to-1 or 1-to-0, but not both. Show that a product code with check modulus  $2^a - 1$  detects all burst errors affecting  $a - 1$  or fewer consecutive bits of a codeword and all unidirectional errors affecting any  $a - 1$  or fewer bits.

### ECE 257A f2006, Homework #3: Dealing with high-level impairments (Lectures 9-12, due Tuesday, Nov. 21)

1. [Triangular code for error correction; 15 points] Instead of arranging  $k$  data bits in a square array and attaching  $2k^{1/2} + 1$  parity bits to its rows and columns, we can use a triangular (half-square) shape, as shown below, and place the parity bits on its diagonal. Each parity bit covers all data bits in the same row and the same column, so every parity bit covers the same number of data bits. Compare this code to the 2D square code with regard to redundancy and error correction/detection capabilities.

```
d d d d p
d d d p
d d p
d d p
d p
p
```

2. [Biproduct codes; 15 points] We discussed biresidue codes and their error correction capabilities in class, noting, for example, that a biresidue code with the low-cost check moduli 7 and 15 can correct any single-bit error within 12 data bits. Consider now a biproduct code with the low-cost check moduli 7 and 15 (that is, a coding scheme whereby numbers are multiplied by  $7 \cdot 15 = 105$ ). Derive the error correction capabilities of this code, compare with the biresidue code mentioned above, and generalize to a biproduct code with relatively prime check moduli  $2^a - 1$  and  $2^b - 1$ .

3. [Sequential diagnosis; 15 points] Consider an  $n$ -node directed ring, with nodes numbered 0 to  $n - 1$ . Augment this ring with additional test links that allow nodes 2, 3, . . . ,  $2t - 1$  (besides node  $n - 1$ ) to test node 0. Show that the resulting system is sequentially  $t$ -diagnosable. *Hint:* Let  $x$  be the number of testers, from among the  $2t - 1$  testers, that indicate node 0 to be malfunctioning, and consider various subcases according to the magnitude of  $x$ .

4. [Modeling of fail-soft systems; 20 points] A fail-soft system is composed of 5 modules, the health of any 3 of which is adequate for system availability. Modules malfunction at the rate 1 and repair is performed at the rate  $m = 1001$ , regardless of the number of existing malfunctions. Upon each malfunction, the system recovers (by changing to a configuration having one fewer module) with probability  $c$ , representing the coverage factor. Find the availability of the system as a function of  $c$ , plot its value as  $c$  varies from 0.8 to 1.0, and discuss the observed trend.

5. [Number of checkpoints; 15 points] Consider a computation that takes 100 hours to run. The failure rate (events that create a need to backtrack to the previous checkpoint) is 1 per hour. Each checkpoint adds 100 s to the computation in way of overhead. Two checkpointing schemes are being considered: (1) inserting 3 checkpoints, one after every 25 hours of computation.; (2) inserting a total of 9 checkpoints, one after every 10 hours of computation. Determine the values of  $l$  for which each checkpointing scheme would be preferable to the other.

6. [Failure avoidance/confinement; 20 points] The November 2006 issue of *IEEE Spectrum* (p. 22, "German Maglev Tragedy") reports in a very brief news story that on 22 September 2006, a train built to demonstrate the magnetic levitation technology in northwest Germany collided with a maintenance vehicle left on the track, killing 25 of the 29 people aboard. Using Internet sources, write a 2-page report on this incident, focusing in particular on the role that computer systems played (or should have played) and the inadequacy of failure avoidance, failure confinement, and fault tolerance strategies.

#### ECE 257A f2006, Homework #4: Hardware implementation topics (Lectures 13-15, due Tuesday, Dec. 5)

1. [Redundant hardware structures; 15 points] Compare the reliabilities and MTTFs for the following three redundancy schemes, each with 4 computational modules. Assume that the voting and switching components are perfectly reliable and that there is no repair. (1) 3-out-of-4 voting. (2) 2-out-of-3 voting and one spare module that is brought on-line when one of the active ones fails; assume identical failure rates for active and spare modules. (3) The pair-and-spare scheme.

2. [Time redundancy with self-dual functions; 20 points] (a) Prove that a  $k$ -bit unsigned, binary adder, with  $2k + 1$  inputs and  $k + 1$  outputs, is self-dual with respect to each of its outputs. (b) Repeat part a for a 1's-complement binary adder that has its carry-out connected to its carry-in. Such an adder has  $2k$  inputs and  $k$  outputs. (c) Repeat part a for a 2's-complement binary adder with  $c_{in} = 0$  ( $2k$  inputs,  $k + 1$  outputs). Is self-duality maintained if the 2's-complement adder is used as an adder-subtractor via a control signal that causes complementation of the second operand and the insertion of  $c_{in} = 1$ ?

3. [Totally self-checking two-rail checker; 15 points] We discussed the design of a circuit, with 4 inputs and 2 outputs, that combines error indicators from two modules (two-rail checker). Show that the design is self-testing with respect to all single stuck-at and multiple unidirectional faults.

4. [Totally self-checking two-rail comparator; 15 points] Discuss the design of a TSC two-rail comparator circuit. The inputs are two-rail encodings of two  $k$ -bit numbers for an arbitrary word width  $k$ . The output is 01 if the input numbers are equal, 10 if they are unequal, and 00 or 11 if there is an input error or if the comparator has an internal fault. Present a complete design with 4-bit two-rail inputs.

5. [Generalized and weighted voting; 20 points] A generalized voting scheme can be specified by listing its agreement sets. For example, standard 2-out-of-3 majority voting with inputs  $A$ ,  $B$ , and  $C$  has the agreement sets  $\{A, B\}$ ,  $\{B, C\}$ ,  $\{C, A\}$ . Show that each of the agreement sets in parts a-d below corresponds to a weighted threshold voting scheme and present a simple hardware voter implementation for at least one of the four cases. (a)  $\{A, B\}$ ,  $\{A, C\}$ ,  $\{A, D\}$ ,  $\{B, C, D\}$ . (b)  $\{A, B\}$ ,  $\{A, C, D\}$ ,  $\{B, C, D\}$ . (c)  $\{A, B, C\}$ ,  $\{A, C, D\}$ ,  $\{B, C, D\}$ . (d)  $\{A, B\}$ ,  $\{A, C, D\}$ ,  $\{B, D\}$ . (e) Is there any generalized voting scheme on 4 inputs that cannot be realized as a weighted voting scheme?

6. [Reconfiguration; 15 points] We have discussed the use of reconfiguration for fault tolerance. Reconfiguration can also be used for flexibility; i.e., use of a common computational resource, such as an FPGA, to solve different problems at the hardware level. Discuss the similarities and differences of these two uses for reconfigurable computing. The following reference might be helpful: Compton, K., and S. Hauck, "Reconfigurable Computing: A Survey of Systems and Software," *ACM Computing Surveys*, Vol. 34, No. 2, pp. 171–210, June 2002.

#### ECE 257A f2006, Notes on homework solutions and other topics

None at this time.

#### ECE 257A f2006, Possible Research Topics

List of topics will be posted here if some students indicate interest in the research option in lieu of final exam. Before starting work on your report, consult [useful guidelines on organization and formatting of a research paper](#).

Return to: [Top of this page](#)

#### ECE 257A: Fall Quarter 1998 offering

**Course:** ECE 257A – Fault-Tolerant Computing, University of California, Santa Barbara, Fall 1998, Enrollment Code 49858

**Instructor:** Behrooz Parhami, Room 5155 Engineering I, Phone 805-893-3211, [parhami@ece.ucsb.edu](mailto:parhami@ece.ucsb.edu)

**Meetings:** MW 4:00-5:30, Room 1431 Phelps Hall

**Consultation:** Open office hours, held in Room 5155 Engineering I – M 3:00-3:50, W 1:00-1:50, R 9:00-9:50

**Motivation:** Dependability concerns are integral parts of engineering design. Ideally, we would like our computer systems to be perfect, always yielding timely and correct results. However, just as bridges collapse and airplanes crash occasionally, so too computer hardware and software cannot be made totally immune to unpredictable behavior. Despite great strides in component reliability and programming methodology, the exponentially increasing complexity of integrated circuits and software products makes the design of perfect computer

systems nearly impossible. In this course, we study the causes of computer system failures (impairments to dependability), techniques for ensuring correct and timely computations despite such impairments, and tools for evaluating the quality of proposed or implemented solutions.

**Prerequisites:** Basic digital system design at the level of ECE 152A/B and, preferably, ECE 154.

**References:** *Reader* – Portions of two out-of-print textbooks will be reproduced as the main reference for the course. We are now awaiting permission from the publishers.  
*Journals* – *IEEE Trans. Computers*, *IEEE Trans. Reliability*, *IEEE Trans. Software Engineering*, *ACM Trans. Computer Systems*, and *Information Processing Letters*. Also, *IEEE Computer*, *IEEE Micro*, *IEEE Design & Test of Computers*, and *ACM Computing Surveys* are good sources for broad introductory papers.  
*Conferences* – Int'l Symp. on Fault-Tolerant Computing (FTCS, annual, since 1971), Pacific Rim Int'l Symp. Fault-Tolerant Systems (PRFTS, odd years, since 1991), Conf. on Computer Assurance (COMPASS), IFIP Int'l Working Conf. Dependable Computing for Critical Applications (DCCA).

**Evaluation:** Students will be evaluated based on these four components with the given weights:

20% -- Homework assigned on Wed. 9/30, 10/14, 11/4, 11/18, each due in 12 days.

25% -- Closed-book midterm, Wed. 10/28, 4:00-6:00 (covers material up to errors).

10% -- Poster presentation of research project, Wed. 12/2, 4:00-6:00, in class.

45% -- Written research report, due by 4:00 pm on Wed. 12/9.

**Research:** A research paper or term project is required. Subject to be finalized by Wed. 10/21. Preliminary title, abstract, and reference list are due on Wed. 11/11. Final title and references are due on Wed. 11/25. Complete paper is due on Wed. 12/9 by 4:00 pm. All deadlines are firm.

### Lecture plan:

Lectures have been scheduled as follows:

M 09/28	Introduction & motivation	W 09/30	Dependability measures
M 10/05	Combinational modeling	W 10/07	State-space modeling
M 10/12	Defect avoidance/circumvention	W 10/14	Fault testing
M 10/19	Fault masking	W 10/21	Error detection
M 10/26	Error correction	W 10/28	MIDTERM EXAM
M 11/02	Malfunction diagnosis	W 11/04	Malfunction tolerance
M 11/09	Degradation allowance/maintenance	W 11/11	Failure confinement/recovery
M 11/16	Self-checking modules	W 11/18	Reconfiguration & voting
M 11/23	Algorithm design methods	W 11/25	Agreement & adjudication
M 11/30	Software redundancy	W 12/02	Research poster session

Return to: [Top of this page](#) || Go up to: [B. Parhami's course syllabi](#) or [his home page](#)

Dr. Behrooz Parhami, Professor  
 Dept. Electrical & Computer Eng.  
 University of California, Santa Barbara  
 Santa Barbara, CA 93106-9560 USA

Web: <http://www.ece.ucsb.edu/~parhami>



Office phone: +1 805 893 3211  
 Department fax: +1 805 893 3262  
 Office: Rm 5155 Harold Frank Hall  
 Deliveries: Rm 4155 Harold Frank Hall

E-mail: [parhami@ece.ucsb.edu](mailto:parhami@ece.ucsb.edu)