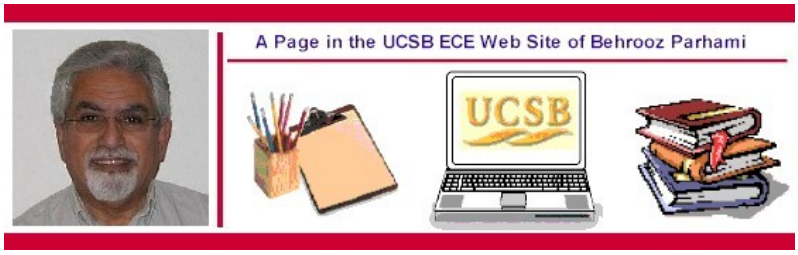

ECE 257A: Fault-Tolerant Computing

Behrooz Parhami: 2007/11/30 || 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

Background and history of ECE 257A

Professor Parhami took over the teaching of ECE 257A in the fall quarter of 1998. Previously, the course had been taught primarily by Dr. John Kelly, who instituted the two-course sequence ECE 257A/B, the first covering general topics and the second (now discontinued) devoted to his research focus on software fault tolerance. Borrowing from his experience in teaching dependable computing at other universities and based on an extensive survey of the field that he published in 1994, Professor Parhami oriented the course toward an original multilevel view of impairments to computer system dependability and techniques for avoiding or tolerating them. The levels of this models, in increasing order of abstraction, are: defects, faults, errors, malfunctions, degradations, and failures. A textbook based on this multilevel model of dependable computing is in preparation.

Link to previous offerings of ECE 257A

ECE 257A: Fall Quarter 2007 offering

This area reserved for important course announcements: There will likely be no further updates to this web page for fall quarter 2007. Updated versions of all course lectures have been posted. The instructor will hold extra office hours on Monday 12/3, from 2:30 to 4:00 PM (the day before HW#4 is due). There will also be extra office hours on Tuesday 12/11, from 10:00 AM to 12:00 PM (the day before the final exam). Our final exam will be on Wednesday 12/12, from 9:00 (not 8:00, as indicated in the schedule of classes) to 11:00 AM.

Course: ECE 257A – Fault-Tolerant Computing, University of California, Santa Barbara, Fall 2007, Enrollment Code 11775

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

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 2:00-3:30, Thursdays 12:00-1:30

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 prefect 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.

Shooman, 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); European Dependable Computing Conf. (EDCC, since 1994, 7th to be held in 2008); 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 9/27	0	Course intro, pretest, take-home survey	pdf, ppt, risks forum	Homework: general requirements
T 10/02	1	Background and motivation	pdf, ppt, handout1	Completed take-home surveys due
R 10/04	2	Dependability measures	pdf, ppt, handout1	HW#1 posted (Lectures 1-4)

T 10/09	3	Combinational modeling	pdf, ppt	
R 10/11	4	State-space modeling	pdf, ppt	Research topic defined
T 10/16	5	Defect avoidance and circumvention	pdf, ppt	HW#1 due; solutions handed out
R 10/18	6	Fault testing	pdf, ppt	HW#2 posted (Lectures 5-8)
T 10/23	7	Fault masking	pdf, ppt	
R 10/25	8	Error detection	pdf, ppt	
T 10/30	9	Error correction	pdf, ppt	HW#2 due; Preliminary references due
R 11/01	10	Malfunction diagnosis and tolerance	pdf, ppt	
T 11/06	1-9	Midterm exam: open-book (10:00-11:50 AM)	Sample problems	Held in class; note extended time
R 11/08	11	Degradation management and reversal	pdf, ppt	HW#3 posted (Lectures 9-12)
T 11/13	12	Failure confinement	pdf, ppt	Paper title and references due
R 11/15	13	Hardware implementation strategies	pdf, ppt	
T 11/20	14	Self-checking modules	pdf, ppt	HW#3 due
R 11/22		Thanksgiving holiday (no lecture)		HW#4 posted (Lectures 13-16)
T 11/27	15	Reconfiguration and voting	pdf, ppt, voting	Paper abstract and outline due
R 11/29	16	Software reliability and redundancy	pdf, ppt, nvp-at	
T 12/04	17	Algorithm design methods	pdf, ppt	HW#4 due
R 12/06	18	Agreement and adjudication	pdf, ppt, adjud	
T 12/11		Extra office hour, 10:00-12:00		Complete paper due by noon
W 12/12	10-18	Final exam: open-book (9:00-11:00 AM)		Held in class

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 f2007, Homework #1: Dependability measures and modeling (Lectures 1-4, due Tuesday, Oct. 16)

1. [Mean time to failure; 5 points] A particular computer model has a constant failure rate of l. What percentage reduction in l would lead to MTTF improvement by 25%? By 100%?

2. [Data availability; 15 points] Redo examples 1.1, 1.2, and 1.3 of the first course handout, assuming that the 5 sites are interconnected as a bidirectional ring network, rather than a complete network. Of course, it no longer makes sense to assume that access to the data is allowed only via a direct link. When data is accessed indirectly, all nodes and links on the indirect path must be functional for access to be successful. In your analysis, consider the worst case regarding where data copies are located relative to the user site.

3. [Availability of a 2-state system; 15 points] A system never fails but is shut down for 30 minutes of preventive maintenance after every two hours of operation. The first two hours of operation begin at time 00:00. (a) Plot the availability of this system in the interval [0, *t*], for 00:00 <= *t* <= 24:00, as a function of *t*. (b) Derive an expression for the interval availability $A(t)$ as a function of *t* (*Hint:* the expression will involve "floor" or "ceiling" operations). (c) Show that the availability of this system tends to 0.8 for large *t*.

4. [Combinational modeling of phased missions; 15 points] Consider a system composed of *n* resources, numbered 1 to *n*, having reliabilities $R_1(t)$, $R_2(t)$, . . . , $R_n(t)$, and a f-phase mission in which the set S_j of resources needed in phase *j* is a subset of S_{j-1} for 2 <= *j* <= f. (a) Write down an expression for the reliability of the system if the completion of all f phases is required for mission success. (b) Present the special case of your expression assuming exponential reliability formulas.

5. [State-space modeling; 25 points] Consider an automobile with four regular tires and one spare tire. The failure rate of a regular tire is l. A spare fails at the rate r < l when it is in the trunk and at the rate s > l when it is used. When the spare is in use, the replaced failed tire is repaired at the rate m. Assume no repair for a failed spare in the trunk, because this event is usually not detected. The system fails when any two tires are unusable. (a) Construct a state-space model for this system and derive the associated reliability equation. (b) What do we need to add to the model in order to allow the derivation of steady-state availability? (c) Relate this example to a computer-based system that you describe.

6. [Microresearch assignment; 25 points] According to news stories published in the last week of September 2007, the newest version of Microsoft Excel contains a flaw that leads to incorrect values in rare cases. For example, when multiplying 77.1 by 850, 10.2 by 6,425 or 20.4 by 3,212.5, the number 100,000 is displayed instead of the correct result 65,535. Similar errors are observed for calculations that produce results close to 65,536. Study this problem using Internet sources and discuss in *one typed page* (single spacing is okay) the nature of the flaw, why it went undetected, exactly what causes the errors, and how Microsoft plans to deal with the problem.

ECE 257A f2007, Homework #2: Dealing with low-level impairments (Lectures 5-8, due Tuesday, Oct. 30)

1. [Defects and yield; 10 points] Every three years, from 1980 to 1992, DRAM chips increased in capacity by a factor of 4 and in die area by a factor of about 1.5 (beginning with 64 Kb and 0.15 cm^2 in 1980), while yields remained virtually constant at around 45%. Assuming that these trends have continued up to the present day, what can you say about trends in defect density and memory cost? State all your assumptions.

2. [Fault testing; 20 points] A half-adder consists of an XOR gate and an AND gate producing the sum and carry-out bits, respectively, when adding two input bits *x* and *y*. A full-adder, with an additional carry-in input, can be built from two HAs and an OR gate. (a) Discuss functional and structural testing of a single-bit FA built as above. (b) Repeat part a for a 4-bit ripple-carry adder built of four FAs. (c) How would you enhance the testability of the 4-bit ripple-carry adder if you were allowed to use only one extra pin?

3. [Testability; 15 points] Quantify the testability of each line of the single-bit full-adder circuit defined in Problem 2 (i.e., built from two HAs and an OR gate). Based on the results obtained, where would you place a single testpoint?

4. [Modeling for fault masking; 20 points] Consider the following fault-masking redundancy scheme with voting. There are five modules that feed a 3-out-ot-5 voter. When a disagreement is detected among the modules, the disagreeing module is purged (switched out), along with one of the good ones, and the voter is reconfigured to act as a 2-out-of-3 unit. The next detected disagreement causes the disagreeing module and one of the good ones to be purged, leaving a simplex system. As usual, we ignore the possibility of simultaneous multiple faults. (a) Construct and solve a state-space reliability model for this system, assuming perfect coverage, switching, and voting. State all your assumptions clearly. (b) Outline a method for improving the reliability of this hardware redundancy scheme that does not involve adding extra modules (only the switching and voting parts can change).

5. [Switch-voter design for fault masking; 20 points] For the redundancy scheme discussed in Problem 4a, design the required 5-input, 1-output voter-switch, with 5 internal FFs indicating which units are contributing to the formation of the output. Assume that the five modules produce 1-bit results. Feel free to use muxes, comparators, and other standard circuits in your design (the design need not be at the gate level).

6. [Error detection; 15 points] A k-bit data word is transmitted bit-serially over a noisy channel. Each received bit is in error with probability of 0.01, independent of other bits. (a) What is the maximum allowable value for k if the received data word should not have more than 10% probability of being in error? (b) What is the probability of an undetected error in a $(k + 1)$-bit word, containing a parity bit and k data bits, where k is the maximum determined in part a? (c) Assuming that the error rate of the channel is beyond our control, what do you suggest we do to cut the error probability of part b in half?

ECE 257A f2007, Homework #3: Dealing with high-level impairments (Lectures 9-12, due Tuesday, Nov. 20)

1. [Error correction; 20 points] Consider the information dispersal scheme of Example 1.3 in the first course handout. (a) Characterize this scheme as an error code, deriving its pertinent characteristics: data width, redundancy, and detection/correction capabilities. (b) Generalize the derivation of part a to the case of k data bits with a p-out-of-q dispersal scheme, deriving the number r of redundant bits, the total number n of bits, and detection/correction capabilities as functions of k, p, and q.

2. [Malfunction diagnosis; 25 points] This problem relates to 1-step t-diagnosability of a collection of subsystems interconnected as an $m \times m$ 2D torus network in which node (i, j) in row i, column j, is connected to the four neighboring nodes $(i \pm 1, j)$ and $(i, j \pm 1)$, where all arithmetic is modulo m and $m > 4$. Links are bidirectional and allow testing in either direction. (a) By defining a suitable testing graph, show that 2D torus is at least 1-step 2-diagnosable; in other words, $t \geq 2$. (b) Can you prove a stronger diagnosability result for the 2D torus? (*Reference:* Araki, T., and Y. Shibata, "Diagnosability of Networks Represented by the Cartesian Product," *IEICE Trans. Fundamentals*, Vol. E83-A, No. 3, pp. 465-470, March 2000.)

3. [Modeling of fail-soft systems; 25 points] A fail-soft system consists of 4 processors and 6 disk storage subsystems. The system can perform its essential functions as long as at least 2 processors and 4 disk units are operational. Failure and repair rates are l and m for a processor, d and r for a disk unit. (a) Construct a complete state-space model for this system, assuming the same repair rate for each unit type, regardless of how many of the units have malfunctioned. (b) The model of part a is rather difficult to solve symbolically, so we consider the following approximation for the case l << d (processors are much more reliable than disks). Construct a simplified 4-state model with the assumption that the system is extremely unlikely to fail due to the number of processors dropping below 2. (c) A balance between simplicity and accuracy can be struck as follows. Take the model of part b and add new transitions from each non-failure state to the failure state, in a way that models failure due to the number of processors dropping below 2. What transition rate s should be associated with these new transitions in order to make the model as accurate as possible? *Hint:* When x is small, e^x can be approximated by $1 + x + x^2/2 + x^3/6$.

4. [Reconfiguration; 30 points] The October 2007 issue of *IEEE TC* contains a paper that proposes a new scheme for dealing with the reconfiguration of VLSI arrays. (a) Enumerate the similarities and differences between the proposed scheme and the compensation path method discussed in class. (b) Does the proposed method offer any benefits for arrays of moderate size, such as 8×8, or are the benefits limited to very large arrays? (c) Write a better abstract for the paper that more clearly defines the novelty and benefits of the approach. (*Reference:* Jigang, W., T. Srikanthan, and X. Wang, "Integrated Row and Column Rerouting for Reconfiguration of VLSI Arrays with Four-Port Switches," *IEEE Trans. Computers*, Vol. 56, No. 10, pp. 1387-1400, October 2007.)

ECE 257A f2007, Homework #4: Hardware/software implementation topics (Lectures 13-16, due Tuesday, Dec. 4)

1. [Self-checking design; 25 points] (a) Design a totally-self-checking checker for a Berger code with 31 data bits and 5 check bits. (b) Describe how your design for part a will change when the Berger code's check part holds 31 – *count*(1s), instead of *count*(0s).

2. [Voting; 10 points] The United Nations Security Council consists of five permanent members and 10 nonpermanent members that serve 2-year terms. For a resolution to be approved by the Council, all five permanent members and at least four nonpermanent members must agree. Can this decision scheme be formulated as weighted voting?

3. [Acceptance test (due to Koren and Krishna); 20 points] The correct output, z, of some program has as its probability density function the truncated exponential function given below, where L is a known positive constant: $f(z) = \text{if } 0 \leq z \leq L \text{ then } \mu e^{-\mu z}/(1 - e^{-\mu L}) \text{ else } 0$. On any particular input, the program fails with probability q, in which case it produces an arbitrary value with uniform distribution in $[0, L]$. The penalty of producing an incorrect value is E, while that of producing no value at all is S, where E and S are known constants. An acceptance test is to be set up in the form of a range check which rejects any output that does not fall in $[0, R]$. Find the optimal value of R for which the expected total penalty is minimized.

4. [Program correctness proof; 20 points] (a) What does program fragment 1 compute, assuming than m and n are positive integers? Define a suitable loop invariant for the program and use it to prove its correctness. (b) Repeat part a for program fragment 2.

Program fragment 1

```
input m, n
x := m
y := n
z := 0
while x > 0 do
    if x is even
    then x := x/2; y := 2y
    else x := x – 1; z := z + y
    endif
endwhile
print z
```

Program fragment 2

```
input m, n
x := m
y := n
while y ≠ 0 do
    r := x mod y
    x := y
    y := r
endwhile
print x
```

5. [Microresearch assignment; 25 points] The International Space Station (ISS) experienced a computer-related crisis in June 2007. According to NASA documents, "On 13 June, a complete shutdown of secondary power to all [three] central computer and terminal computer channels occurred, resulting in the loss of capability to control ISS Russian segment systems." Study this ISS incident using Internet sources and discuss in *one typed page* (single spacing is okay) the nature of the crisis, its underlying causes, how the problems were dealt with, and what we can learn from this experience in terms of how to design dependable systems with diverse subsystems and suppliers.

ECE 257A f2007, Notes on homework solutions and other topics

HW2, Problem 3: Please change the label on the output of the right-hand AND gate to "3/16; 1/2; 3/32" and that of the OR gate to "7/64; 1; 7/64".

HW2, Problem 4: State probabilities and system reliability should be corrected as follows. $p_5 = e^{-5\lambda t}$, $p_3 = 2.5e^{-3\lambda t} - 2.5e^{-5\lambda t}$, $p_1 = 1.875e^{-\lambda t} - 3.75e^{-3\lambda t} + 1.875e^{-5\lambda t}$, $p_0 = 1 - 1.875e^{-\lambda t} + 1.25e^{-3\lambda t} - 0.375e^{-5\lambda t}$, $R = 1 - p_0 = 1.875e^{-\lambda t} - 1.25e^{-3\lambda t} + 0.375e^{-5\lambda t}$. The errors went undetected because the given erroneous solutions actually satisfy the differential equations (they fail the initial conditions, though).

ECE 257A f2007, Sample exam problems

1. [State-space modeling; 20 points] Consider a state-space model for a system that can be in one of three states: G (good), U (bad, failure undetected), F (bad, failure detected). Assume failure rate of l, repair rate of m, and failure detection "rate," modeling the latency of failure detection, of d. Calculate the steady-state availability of this system and discuss the implications of delayed failure detection by comparing your result with that of a two-state system that has immediate failure detection.

2. [Fault testing; 20 points] Show that a tree of 2-input XOR gates, implementing the logic function x_1 Å x_2 Å . . . Å x_n, can be tested for all single s-a-0 and s-a-1 faults with only three test patterns, regardless of the number *n* of inputs. *Hint:* A single test detects all single s-a-1 faults.

3. [Error detection; 20 points] Explain why all single-digit errors are caught by the UPC-A coding scheme which is based on modulo-10 checksum on 11 data digits and 1 check digit, using the weight vector 3 1 3 1 3 1 3 1 3 1 3 1. Explain why all transposition errors (adjacent digits switching positions) are not caught.

4. [Voters for TMR and 5MR; 20 points] Show at least one implementation for a voter (using logic gates and/or standard building-block combinational circuits, such as comparators, multiplexers, and the like) for a 2-out-of-3 word-voter. The three inputs and the voter output are *k*-bit words. The voter need not detect the lack of majority, but must produce the majority value, if one exists, at its output. Can your design be extended to a 3-out-of-5 voter?

5. [Modeling of a phased mission; 20 points] A phased mission is one which requires the availability of different resources in each of several phases of operation. Consider for example a two-phase mission for a computer system with three resources (subsystems) A, B, and C. During phase 1, which lasts L_1 hours, only subsystem A needs to be operational. During phase 2, of duration L_2, proper functioning of subsystem B, plus one of the other two subsystems, would suffice. Such a mission is deemed a success if both phases are completed with the required resources being operational. Assuming exponential reliabilities, with constant failure rates l_A, l_B, and l_C for the three resources (regardless of their being in operation or idle), write down the reliability equation for the two-phase mission just defined.

6. [Malfunction diagnosis; 20 points] Prove directly (i.e., by forming the malfunction syndromes and comparing them to each other, rather than by using general theorems about diagnosability) that an *n*-node directed ring network is 1-step 1-diagnosable but not 1-step 2-dignosable. *Hint:* Take advantage of symmetry to reduce the amount of work.

7. [Checkpointing; 25 points] We discussed optimal checkpointing under the assumption that time overhead per checkpoint is a constant. Suppose that checkpointing overhead is a linear function of checkpointing period, that is, the longer the time interval between checkpoints, the more information there is to store and the longer the time overhead for each checkpoint. Present an analysis of optimal checkpointing in this case, stating all your assumptions.

8. [Self-checking modules; 15 points] Suppose that the correct functioning of a 2-to-4 decoder is to be monitored. Design a totally-self-checking checker to be used at the output of the decoder. Show that your design is indeed totally self-checking.

ECE 257A f2007, 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 || 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 at ece.ucsb.edu