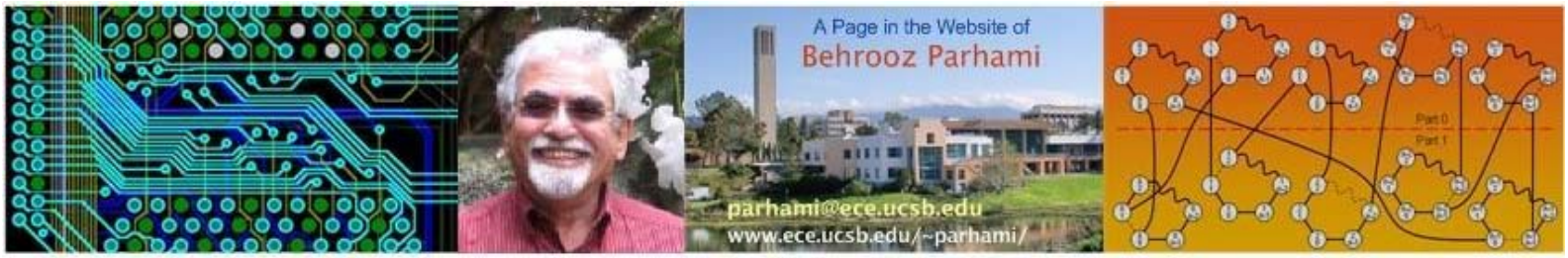


- Home & Contact
- Curriculum Vitae
- Research
  - Computer arithmetic
  - Parallel processing
  - Fault tolerance
  - Broader research
  - Research history
  - List of publications
- Teaching
  - ECE1 Freshman sem
  - ECE154 Comp arch
  - ECE252B Comp arith
  - ECE252C Adv dig des
  - ECE254B Par proc
  - ECE257A Fault toler
  - Student supervision
  - Math + Fun!
- Textbooks
  - Computer arithmetic
  - Parallel processing
  - Dependable comp
  - Comp architecture
  - Other books
- Service
  - Professional activities
  - Academic service
  - Community service
  - Industrial consulting
- Files & Documents
- Useful Links
- Personal

Behrooz Parhami's ECE 254B Course Page for Fall 2010

Adv. Computer Architecture: Parallel Processing

Enrollment code: 52142
Prerequisite: ECE 254A (can be waived, but ECE 154 is required)
Class meetings: TR 4:00-5:30, Phelps 1431
Instructor: Professor Behrooz Parhami
Open office hours: M 11:00-12:30, W 12:30-2:00, HFH 5155
Course announcements: Listed in reverse chronological order
Course calendar: Schedule of lectures, homework, exams, research
Homework assignments: Four assignments, worth a total of 30%
Exams: None for fall 2010
Research paper: Report and short oral presentation, worth 70%
Research paper guidlines: Brief guide to format and contents
Poster presentation tips: Brief guide to format and structure
Grade statistics: Range, mean, etc. for homework and exam grades
References: Textbook and other sources (**Textbook's web page**)
Lecture slides: Available on the textbook's web page
Miscellaneous information: Motivation, catalog entry, history

Course Announcements

2010/11/24: Updated lecture slides for Parts V and VI of the textbook (the latter with very limited changes) have been posted to the book's Web page.
2010/11/21: Comments on the research summary and final references have been e-mailed to each student.
2010/11/16: Homework #4 has been posted (due 11/30) and lecture slides for Part IV of the textbook have been updated for fall 2010. Please submit your final references and one-page summary for your research project by midnight on Saturday 11/20.
2010/10/24: Homework #3 has been posted and lecture slides for Part III of the textbook have been updated for fall 2010.
2010/10/20: Research topics have been assigned. The next research milestone will be submission of a preliminary list of references by T 10/26 [extendable to R 10/28].
2010/10/10: Homework #2 has been posted and lecture slides for Part II" of the textbook have been updated for fall 2010. A number of research topics will be posted later today.
2010/10/05: The due date for Homework #1 has changed to Tuesday 10/12. Updated version of the lecture slides for Part II' is now available.
2010/10/01: Problem statements for Homework #1 have been posted below for students who do not yet have the textbook.
2010/09/26: Homework #1 has been posted and lecture slides for Part I of the textbook have been updated for fall 2010.
2010/09/21: The information appearing on this page is now final. There will be no lecture during the week of November 8-12 (instructor at a conference and Veterans Day holiday).
2010/07/09: Welcome to the ECE 254B web page for fall 2010. The following tentative information is provided for planning purposes only. Details will be finalized in September 2010 and updated weekly thereafter.

Course Calendar

Course lectures, homework assignments, midterm exam, and research milestones have been scheduled as follows. This schedule will be strictly observed. In particular, no extension is possible for homework due dates or research milestones. Each lecture covers topics in 1-2

Blog & books
Favorite quotations
Poetry
Pet peeve
Virtual retirement
CE Program
ECE Department
UCSB Engineering
UC Santa Barbara

chapters of the textbook. Chapter numbers are provided in parentheses, after day & date. PowerPoint and PDF files of the lecture slides can be found on the **textbook's web page**.

Day & Date (book chapters) Lecture topic [Homework posted/due] {Special notes}
R 09/23 (1) Introduction to parallel processing {Introductory survey}

T 09/28 (2) A taste of parallel algorithms [HW1 posted]
R 09/30 (3-4) Complexity and parallel computation models

T 10/05 (5) The PRAM shared-memory model and basic algorithms
R 10/07 (6A) More shared-memory algorithms [HW1 due]

T 10/12 (6B-6C) Shared memory implementations and abstractions [HW2 posted]
R 10/14 (7) Sorting and selection networks {Research topic defined}

T 10/19 (8A) Search acceleration circuits
R 10/21 (8B-8C) Other circuit-level examples [HW2 due]

T 10/26 (9) Sorting on a 2D mesh or torus architectures [HW3 posted] {Preliminary references due}
R 10/28 (10) Routing on a 2D mesh or torus architectures

T 11/02 (11) Other algorithms for mesh/torus architectures
R 11/04 (12) Mesh/torus variations and extensions

T 11/09 No lecture: Instructor away at a conference
R 11/11 No lecture: Veterans Day holiday [HW4 posted]

T 11/16 (13) Hypercubes and their algorithms [HW3 due] {Final references and 1-page research summary due --> extended to midnight on Sat. 11/20}
R 11/18 (14) Sorting and routing on hypercubes

T 11/23 (15-16) A sampling of other interconnection networks [HW4 due --> extended to 11/30]
R 11/25 No lecture: Thanksgiving holiday

T 11/30 (17-18) Task scheduling and input/output {Instructor/course evaluation surveys}
R 12/02 Research poster presentations

R 12/09 {Final research paper due by midnight; please submit a hard copy and a PDF file}

W 12/15 {Course grades to be submitted by midnight}

Homework Assignments

-Turn in solutions in class before the lecture begins.
-Because solutions will be handed out on the due date, no extension can be granted.
-Use a cover page that includes your name, course name, and assignment number.
-Staple the sheets and write your name on top of each sheet in case they are separated.
-Although some cooperation is permitted, direct copying will have severe consequences

Homework 1: Introduction, models, and complexity (ch. 1-4, due R 2010/10/07, 4:00 PM)
Do the following problems from the textbook: 1.11, 2.5, 2.9, 3.7cd

Literature review: Denning, P. J. and J. B. Dennis, "The Profession of IT: The Resurgence of Parallelism," *Communications of the ACM*, Vol. 53, No. 6, pp. 30-32, June 2010.
Read the article above and discuss the following questions (typed answers, not taking more than 2 pages in total, are required).

a. What was the first parallel computer, what kind of parallel architecture did it have, and why wasn't it ultimately successful?
b. Why didn't commercial vendors pay much attention to parallel processing until very recently?
c. What is the "determinacy theorem"?
d. Why are transaction systems inherently nondeterminate?
e. Name, and very briefly define, the three principles of modular composability for parallel tasks.
f. Why does the use of virtual memory simplify parallel programming?
g. What would be a reasonable way of avoiding cache consistency problems altogether?
h. What is functional programming (FP)? Name two successful examples of FP languages.
i. Give an example of a "wheel" that is being reinvented by today's parallel processing researchers.

Homework 2: Shared memory and circuit model (ch. 5-8, due R 2010/10/21, 4:00 PM)
Do the following problems from the textbook: 5.4, 5.12, 6.13, 7.7, 8.8

Literature review: Patterson, D., "The Trouble with Multi-Core," *IEEE Spectrum*, Vol. 57, No. 7, pp. 28-32 & 53, July 2010.
Read the article above and discuss the following questions (typed answers, not taking more than 2 pages in total, are required).
a. Why was the migration to multicore processor chips inevitable?
b. Why have processor clock frequencies remained stagnant over the past few years?
c. Why is automatic parallelization unlikely to be very successful beyond a handful of cores?
d. What are the main reasons for computer graphics applications being among the most successful applications of parallel processing?
e. In what way does cloud computing facilitate the widespread adoption of parallel programming methods, when compared with the shrink-warp model of software distribution?
f. How do modern FPGAs facilitate the development of parallel software for manycore processor chips of the future?
g. What role will graphic processing units (GPUs) play in the success or failure of future multicore chips?

Homework 3: Mesh- and torus-connected computers (ch. 9-12, due T 2010/11/16, 4:00 PM)
Do the following problems from the textbook: 9.2, 9.8, 10.14abc, 11.2

Literature review, on extending Amdahl's law:
Hill, M. D. and M. R. Marty, "Amdahl's Law in the Multicore Era," *IEEE Computer*, Vol. 41, No. 7, pp. 33-38, July 2008.
Woo, D. H. and H.-H. S. Lee, "Extending Amdahl's Law for Energy-Efficient Computing in the Many-Core Era," *IEEE Computer*, Vol. 41, No. 12, pp. 24-31, December 2008.
Read the two articles above, present a half-page summary of each, and comment on their commonalities and differences (typed answers, not taking more than 2 pages in total, are required).

Homework 4: Hypercubic and other networks (ch. 13-16, due T 2010/11/30, 4:00 PM)
Do the following problems from the textbook: 13.1, 13.10, 14.4, 14.10, 16.3

Literature review, on energy-efficient supercomputing:
Cameron, K. W., "A Tale of Two Green Lists," *IEEE Computer*, Vol. 43, No. 9, pp. 86-88, September 2010.
Read the article above and answer the following questions (typed answers, not taking more than 2 pages in total, are required).
a. Why is energy-efficiency of interest for supercomputers?
b. What is the SPECpower benchmark?
c. What is the Green500 list and how does it relate to the Top500 list?
d. What aspects of energy-efficiency are assessed by SPECpower and Green500?
e. For the top 10 supercomputers on the Top500 list, how does energy cost over the lifetime of the system compare with the initial acquisition cost?
f. Answer the previous question for the top 10 machines on the Green500 list.
g. How does the average number of processors in the top 10 machines on the Green500 list compare with that of the top 10 machines on the Top500 list?
h. Expanding from the previous question, discuss whether power efficiency is correlated with the number of processors used.

Sample Exam and Study Guide [Does not apply to fall 2010]

The following sample exam using problems from the textbook is meant to indicate the types and levels of problems, rather than the coverage (which is outlined in the course calendar). Students are responsible for all sections and topics, in the textbook and class handouts, that are not explicitly excluded in the study guide that follows the sample exam, even if the material was not covered in class lectures.

Sample Midterm Exam (105 minutes)

Textbook problems 2.3, 3.5, 5.5 (with $i + s$ corrected to $j + s$), 7.6a, and 10.7a; note that problem statements might change a bit for a closed-book exam.

Midterm Exam Study Guide

The following sections are excluded from the first 10 chapters of the textbook to be covered in the midterm exam: 2.6, 3.5, 4.5, 4.6, 6.5, 7.6, 8.5, 9.6.

Research Paper and Presentation

Each student will review a subfield of parallel processing or do original research on a selected and approved topic. A tentative list of research topics is provided below; however, students should feel free to propose their own topics for approval. A publishable report earns an "A" for the course, regardless of homework grades. See the course calendar for schedule and due dates and **Research Paper Guidlines** for formatting tips.

1. Shared Memory Consistency: Models and Implementations (Assigned to: TBD)
[Stei04] Steinke, R. C. and G. J. Nutt, "A Unified Theory of Shared Memory Consistency," *J. ACM*, Vol. 51, No. 5, pp. 800-849, September 2004.
[Adve10] Adve, S. V. and H.-J. Boehm, "Memory Models: A Case for Rethinking Parallel Languages and Hardware," *Communications of the ACM*, Vol. 53, No. 8, pp. 90-101, August 2010.

2. Area/Time/Power Trade-offs in Designing Universal Circuits (Assigned to: TBD)
[Bhat08] Bhatt, S. N., G. Bilardi, and G. Pucci, "Area-Time Tradeoffs for Universal VLSI Circuits," *Theoretical Computer Science*, Vol. 408, Nos. 2-3, pp. 143-150, November 2008.
[Leis85] Leiserson, C. E., "Fat-Trees: Universal Networks for Hardware-Efficient Supercomputing," *IEEE Trans. Computers*, Vol. 3, No. 10, pp. 892-901, October 1985.

3. Optimized Interconnection Networks for Parallel Processing (Assigned to: TBD)
[Gupt06] Gupta, A. K., and W. J. Dally, "Topology Optimization of Interconnection Networks," *IEEE Computer Architecture Letters*, Vol. 5, No. 1, pp. 10-13, January-June 2006.
[Ahon04] Ahonen, T., D. A. Siguenza-Tortosa, H. Bin, and J. Nurmi, "Topology Optimization for Application-Specific Networks-on-Chip," *Proc. Int'l Workshop System-Level Interconnect Prediction*, pp. 53-60, 2004.

4. Trade-offs in Low- vs High-Dimensional Meshes and Tori (Assigned to: TBD)
[Dall90] Dally, W. J., "Performance Analysis of k-ary n-cube Interconnection Networks," *IEEE Trans. Computers*, Vol. 39, No. 6, pp. 775-785, June 1990.
[Agar91] Agarwal, A., "Limits on Interconnection Network Performance," *IEEE Trans. Parallel and Distributed Systems*, Vol. 2, No. 4, pp. 398-412, October 1991.

5. Implementing Deadlock-Free Routing via Turn Prohibition (Assigned to: TBD)
[Glas94] Glass, C. J. and L. M. Ni, "The Turn Model for Adaptive Routing," *J. ACM*, Vol. 41, No. 5, pp. 874-902, September 1994.
[Levi10] Levitin, L., M. Karpovsky, and M. Mustafa, "Minimal Sets of Turns for Breaking Cycles in Graphs Modeling Networks," *IEEE Trans. Parallel and Distributed Systems*, Vol. 21, No. 9, pp. 1342-1353, September 2010.

6. Swapped and Biswapped Networks: A Comparative Study (Assigned to: **Miguel Lastras**)
[Parh05] Parhami, B., "Swapped Interconnection Networks: Topological, Performance, and Robustness Attributes," *J. Parallel and Distributed Computing*, Vol. 65, No. 11, pp. 1443-1452, November 2005.
[Xiao10] Xiao, W. J., B. Parhami, W. D. Chen, M. X. He, and W. H. Wei "Fully Symmetric Swapped Networks

Based on Bipartite Cluster Connectivity," *Information Processing Letters*, Vol. 110, No. 6, pp. 211-215, 15 February 2010.

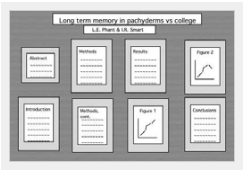
7. Robust Task Scheduling Algorithms for Parallel Processors (Assigned to: **Rune Bilit**)
[Ghos97] Ghosh, S., R. Melhem, and D. Mosse, "Fault Tolerance through Scheduling of Aperiodic Tasks in Hard Real-Time Multiprocessor Systems," *IEEE Trans. Parallel and Distributed Systems*, Vol. 8, No. 3, pp. 272-284, March 1997.
[Beno08] Benoit, A., M. Hakem, and Y. Robert, "Fault Tolerant Scheduling of Precedence Task Graphs on Heterogeneous Platforms," *Proc. Int'l. Symp. Parallel and Distributed Processing*, pp. 1-8, 2008.

8. Artificial Neural Networks as Parallel Systems and Algorithms (Assigned to: TBD)
[Take92] Takefuji, Y., *Neural Network Parallel Computing*, Kluwer, 1992.
[Yao99] Yao, X., "Evolving Artificial Neural Networks," *Proc. IEEE*, Vol. 87, No. 9, pp. 1423-1447, September 1999.

9. Adaptable Parallelism for Real-Time Performance and Reliability (Assigned to: **Vegard Aaker**)
[Moro96] Moron, C. E., "Designing Adaptable Real-Time Fault-Tolerant Parallel Systems," *Proc. Int'l Parallel Processing Symp.*, pp. 754-758, 1996.
[Hsiu09] Hsiung, P.-A., C.-H. Huang, and Y.-H. Chen, "Hardware Task Scheduling and Placement in Operating Systems for Dynamically Reconfigurable SoC," *J. Embedded Computing*, Vol. 3, No. 1, pp. 53-62, 2009.

10. Distributed System-Level Malfunction Diagnosis in Multicomputers (Assigned to: **Amirali Ghofrani**)
[Soma87] Somani, A. K., V. K. Agarwal, and D. Avis, "A Generalized Theory for System Level Diagnosis," *IEEE Trans. Computers*, Vol. 36, pp. 538-546, 1987
[Pelc91] Pelc, A., "Undirected Graph Models for System-Level Fault Diagnosis," *IEEE Trans. Computers*, Vol. 40, No. 11, pp. 1271-1276, November 1991.

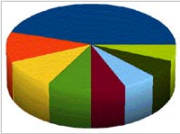
Poster Presentation Tips



Here are some guidelines for preparing your research poster. The idea of the poster is to present your research results and conclusions thus far, get oral feedback during the session from the instructor and your peers, and to provide the instructor with something to comment on before your final report is due. Please send a PDF copy of the poster via e-mail by midnight on the poster presentation day.

Posters prepared for conferences must be colorful and eye-catching, as they are typically competing with dozens of other posters for the attendees' attention. Here is an **example of a conference poster**. Such posters are often mounted on a colored cardboard base, even if the pages themselves are standard PowerPoint slides. In our case, you should aim for a "plain" poster (loose sheets, to be taped to the wall in our classroom) that conveys your message in a simple and direct way. Eight to 10 pages, each resembling a PowerPoint slide, would be an appropriate goal. You can organize the pages into 2 x 4 (2 columns, 4 rows), 2 x 5, or 3 x 3 array on the wall. The top two of these might contain the project title, your name, course name and number, and a very short (50-word) abstract. The final two can perhaps contain your conclusions and directions for further work (including work that does not appear in the poster, but will be included in your research report). The rest will contain brief description of ideas, with emphasis on diagrams, graphs, tables, and the like, rather than text which is very difficult to absorb for a visitor in a very limited time span.

Grade Statistics



All grades listed are in percent.
HW1 grades: Range = [73, 100], Mean = 83, Median = 80
HW2 grades: Range = [58, 100], Mean = 85, Median = 91
HW3 grades: Range = [62, 100], Mean = 82, Median = 82
HW4 grades: Range = [xx, xx], Mean = xx, Median = xx
[Does not apply to fall 2010] Midterm exam grades: Range = [xx, xx], Mean = xx, SD = xx, Median = xx
Research paper and presentation grades: Range = [xx, xx], Mean = xx, Median = xx

References

Required text: B. Parhami, *Introduction to Parallel Processing: Algorithms and Architectures*,

Plenum Press, 1999. Make sure that you visit the **textbook's web page** which contains an errata. Lecture slides are also available there.

Optional recommended book: Herlihy, M. and N. Shavit, *The Art of Multiprocessor Programming*, Morgan Kaufmann, 2008. Because the focus of our course is on architecture and its interplay with algorithms, this book, which deals primarily with software and programming topics, constitutes helpful supplementary reading.

Research resources:

The follolwing journals contain a wealth of information on new developments in parallel processing: *IEEE Trans. Parallel and Distributed Systems*, *IEEE Trans. Computers*, *J. Parallel & Distributed Computing*, *Parallel Computing*, *Parallel Processing Letters*. Also, see *IEEE Computer* and *IEEE Concurrency* (the latter ceased publication in late 2000) for broad introductory articles.

The following are the main conferences of the field: Int'l Symp. Computer Architecture (ISCA, since 1973), Int'l Conf. Parallel Processing (ICPP, since 1972), Int'l Parallel & Distributed Processing Symp. (IPDPS, formed in 1998 by merging IPPS/SPDP, which were held since 1987/1989), and ACM Symp. Parallel Algorithms and Architectures (SPAA, since 1988).

UCSB library's electronic journals, collections, and other resources

UCSB library's research guide in ECE

Miscellaneous Information

Motivation: The ultimate efficiency in parallel systems is to achieve a computation speedup factor of p with p processors. Although often this ideal cannot be achieved, some speedup is generally possible by using multiple processors in a concurrent (parallel or distributed) system. The actual speed gain depends on the system's architecture and the algorithm run on it. This course focuses on the interplay of architectural and algorithmic speedup techniques. More specifically, the problem of algorithm design for "general-purpose" parallel systems and its "converse," the incorporation of architectural features to help improve algorithm efficiency and, in the extreme, the design of algorithm-based special-purpose parallel architectures, are dealt with. The foregoing notions will be covered in sufficient detail to allow extensions and applications in a variety of contexts, from network processors, through desktop computers, game boxes, Web server farms, multiterabyte storage systems, and mainframes, to high-end supercomputers.

Catalog entry: **254B. Advanced Computer Architecture: Parallel Processing(4) PARHAMI.** *Prerequisites: ECE 254A. Lecture, 4 hours*. The nature of concurrent computations. Idealized models of parallel systems. Practical realization of concurrency. Interconnection networks. Building-block parallel algorithms. Algorithm design, optimality, and efficiency. Mapping and scheduling of computations. Example multiprocessors and multicomputers.

History: The graduate course ECE 254B was created by Dr. Parhami, shortly after he joined UCSB in 1988. It was first taught in spring 1989 as ECE 594L, Special Topics in Computer Architecture: Parallel and Distributed Computations. A year later, it was converted to ECE 251, a regular graduate course. In 1991, Dr. Parhami led an effort to restructure and update UCSB's graduate course offerings in the area of computer architecture. The result was the creation of the three-course sequence ECE 254A/B/C to replace ECE 250 (Adv. Computer Architecture) and ECE 251. The three new courses were designed to cover high-performance uniprocessing, parallel computing, and distributed computer systems, respectively. In 1999, based on a decade of experience in teaching ECE 254B, Dr. Parhami published the textbook *Introduction to Parallel Processing: Algorithms and Architectures* (**Website**).

Offerings of ECE 254B from 2000 to 2006 (PDF file)

Offering of ECE 254B in fall 2008 (PDF file)