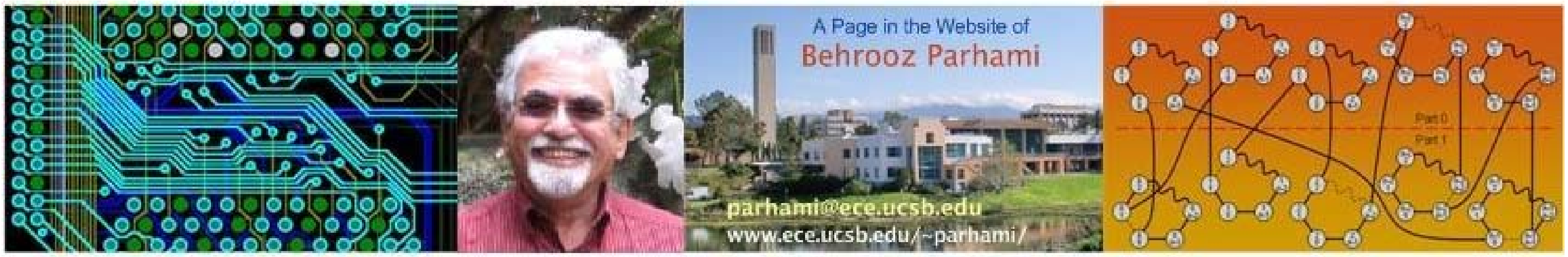


Home & Contact
Curriculum Vitae
Research
Computer arithmetic
Parallel processing
Fault tolerance
Broader research
Research history
List of publications
Teaching
ECE1 Freshman sem
ECE154 Comp arch
ECE252B Comp arith
ECE252C Adv dig des
ECE254B Par proc
ECE257A Fault toler
Student supervision
Math + Fun!
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 Winter 2013

Adv. Computer Architecture: Parallel Processing

Page last updated on 2013 March 25

Enrollment code: 55434
Prerequisite: ECE 254A (can be waived, but ECE 154 is required)
Class meetings: MW 10:00-11:30, Phelps 1431
Instructor: Professor Behrooz Parhami
Open office hours: MW 3:30-5:00, HFH 5155
Course announcements: Listed in reverse chronological order
Course calendar: Schedule of lectures, homework, exams, research
Homework assignments: Five assignments, worth a total of 30%
Exams: None for winter 2013
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

2013/03/25: The winter 2013 offering of ECE 254B is officially over and final course grades have been reported to the Registrar's office. I have sent e-mail feedback on your research papers and am open to discussing them during my spring 2013 office hours.
2013/03/16: All lecture slides for Parts I to V of the textbook have now been updated for winter 2013. Stats for HW5 and the overall homework grade have been posted. For research paper, please submit a PDF file (no hard copy needed) by the 3/21, midnight, deadline.
2013/02/27: The last homework (#5) has been posted to the homework area. It will be due on W 3/13. Updated version of the lecture slides for Part IV of the textbook will be posted by the end of Friday 3/1.
2013/02/18: Homework 4 has been posted to the homework area. It will be due on W 2/27. Updated version of the lecture slides for Part III of the textbook will be posted by the end of Friday 2/22.
2013/02/04: Homework 3 has been posted to the homework area. It will be due on W 2/13. A slightly updated version of the lecture slides for Part II" of the textbook will be posted by the end of today. Students who turned in their research preferences by class time today have been assigned their first-choice topics.
2013/01/29: Fifteen research topics, updated for winter 2013, have been posted to the research area below. Please send me an e-mail with three of these topics, listed in order of your preference (1st to 3rd choices), or bring your choices to class, by the topic selection deadline of M 2/4.
2013/01/25: Updated lecture slides for Parts II' and II" have been posted
2013/01/20: Homework 2 has been posted to the homework area. It will be due on W 1/30. Updated slides for Part II' of the textbook will be posted by F 1/25. I urge you to attend the CS Distinguished Lecture by Leslie Valiant: "The Multi-core Problem as an Algorithmic Problem" (F 2/8, ESB 2001, 11:00-12:00).
2013/01/08: Homework 1 has been posted to the homework area. It will be due on W 1/23. A slightly updated set of slides for Part I of the textbook will be posted by Saturday 1/12.
2013/01/06: The course requirements and schedule are now final. Please pay attention to the points made in the course announcement dated 2012/11/26.
2012/11/26: Welcome to the ECE 254B web page for winter 2013. The following information is provided for planning purposes only. Details will be finalized in late December and updated regularly thereafter. I will be

updating and improving the on-line lecture slides as the course proceeds, so the winter 2013 contents will be different from the current version. Please pay attention to the associated posting date when downloading material for the course.

Course Calendar

Course lectures, homework assignments, exams, 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 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}
M 01/07 (1) Introduction to parallel processing
W 01/09 (2) A taste of parallel algorithms [HW1 posted, chs. 1-4]

M 01/14 (3-4) Complexity and parallel computation models
W 01/16 (5) The PRAM shared-memory model and basic algorithms

M 01/21 No lecture: Martin Luther King Holiday [HW2 posted, chs. 5-6]
W 01/23 (6A) More shared-memory algorithms [HW1 due]

M 01/28 (6B-6C) Shared memory implementations and abstractions
W 01/30 (7) Sorting and selection networks [HW2 due]

M 02/04 (8A) Search acceleration circuits [HW3 posted, chs. 7-8] {Research topic defined}
W 02/06 (8B-8C) Other circuit-level examples

M 02/11 (9) Sorting on a 2D mesh or torus architectures {Preliminary references due}
W 02/13 (10) Routing on a 2D mesh or torus architectures [HW3 due; postponed to 02/20]

M 02/18 No lecture: President's Day Holiday [HW4 posted, chs. 9-12]
W 02/20 (11) Other algorithms for mesh/torus architectures

M 02/25 (12) Mesh/torus variations and extensions {Final references and 1-page research summary due}
W 02/27 (13) Hypercubes and their algorithms [HW4 due]

M 03/04 (14) Sorting and routing on hypercubes [HW5 posted, chs. 13-16]
W 03/06 (15) Other hyprcubic architectures

M 03/11 (16) A sampling of other interconnection networks {Class cancelled}
W 03/13 (17-18) Task scheduling and input/output [HW5 due] {Instructor/course evaluation surveys}

R 03/21 {Final research paper due by midnight; please submit a PDF file}

T 03/26 {Course grades are due 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 (chs. 1-4, due W 2013/01/23, 10:00 AM)
Do the following problems from the textbook or stated below: 1.10, 1.19, 2.5, 3.5, 3.26, 4.12
1.19 Parallel processing effectiveness
Show that the asymptotic time complexity for the parallel addition of n numbers does not change if we use $n/\log_2 n$ processors instead of $n/2$ processors required to execute the computation graph of Fig. 1.14 in

minimum time.

3.26 Comparing asymptotic complexities

Order the complexities listed at the bottom of slide 71 for Part I of the textbook in connection with the maximum network flow problem in asymptotic order, from most complex to least complex, assuming:
a. $e = \Theta(n^2)$ b. $e = \Theta(n)$

Homework 2: Shared memory model of parallel processing (chs. 5-6, due W 2013/01/30, 10:00 AM)
Do Problems 5.5 (see correction below), 5.9, 5.16, 6.3, and 16.13 (ch. 16) from the textbook.
Correction to Problem 5.5: Change both instances of $i + s$ to $j + s$.

Homework 3: Circuit model of parallel processing (chs. 7-8, due W 2013/02/13, 10:00 AM)
Do Problems from the textbook or stated below: 7.2, 7.4, 7.6, 7.16ab, 8.9, 8.23

8.23 Two-pick arbiter

A two-pick arbiter is a circuit that receives *n* request signals and issues grant signals to two of the requesting units, according to some built-in priority scheme.
a. Formulate the design of a 2-pick fixed-priority arbiter as a parallel prefix computation.
b. Discuss how a rotating-priority two-pick arbiter might be designed.

Homework 4: Mesh- and torus-connected computers (chs. 9-12, due W 2013/02/27, 10:00 AM)
Do the following problems from the textbook: 9.6, 9.9, 10.7, 11.1, 12.2

Homework 5: Hypercubic and other networks (chs. 13-16, due W 2013/03/13, 10:00 AM)
Do the following problems from the textbook or stated below: 13.3, 13.21, 14.4, 15.2abc, 16.2

13.21 Embedding parameters

The three examples in Fig. 13.2 suggest that dilation, congestion, and load factor are orthogonal parameters in the sense that knowing two of them does not provide much, if any, information about the third. Reinforce this conclusion by constructing, when possible, additional examples for which dilation, congestion, and load factor are (respectively):
a. 1, 1, 2 b. 1, 2, 1 c. 2, 1, 1 d. 2, 1, 2 e. 2, 2, 2

Sample Exam and Study Guide [Does not apply to winter 2013]

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: **Manish V. Lavekar**)
[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: TBD)
[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: **Sowmyan Kosthubadharan**)
[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: **Rui Wu**)
[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: TBD)
[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.

11. The MapReduce Approach to Parallel Processing (Assigned to: **Ryan M. Pakbaz**)
[Dean10] Dean, J. and S. Ghemawat, "MapReduce: A Flexible Data Processing Tool," *Communications of the*

ACM, Vol. 53, No. 1, pp. 72-77, January 2010.
[Ston10] Stonebraker, M., et al., "MapReduce and Parallel DBMSs: Friends or Foes?" *Communications of the ACM*, Vol. 53, No. 1, pp. 64-71, January 2010.

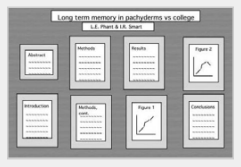
12. Transactional Memory: Concept and Implementations (Assigned to: **Joseph W. Malcolm**)
[Laru08] Larus, J. and C. Kozyrakis, "Transactional Memory," *Communications of the ACM*, Vol. 51, No. 7, pp. 80-88, July 2008.
[Dice09] Dice, D., Y. Lev, M. Moir, and D. Nussbaum, "Early Experience with a Commercial Hardware Transactional Memory Implementation," *Proc. 14th Int'l Conf. Architectural Support for Programming Languages and Operating Systems*, 2009, pp. 157-168.

13. The Notion of Reliability Wall in Parallel Computing (Assigned to: TBD)
[Yang12] Yang, X., Z. Wang, J. Xue, and Y. Zhou, "The Reliability Wall for Exascale Supercomputing," *IEEE Trans. Computers*, Vol. 61, No. 6, pp. 767-779, June 2012.
[Zhen09] Zheng, Z. and Z. Lan, "Reliability-Aware Scalability Models for High-Performance Computing," *Proc. Int'l Conf. Cluster Computing*, 2009, pp. 1-9.

14. FPGA-Based Implementation of Application-Specific Parallel Systems (Assigned to: **Damein L. Morgan**)
[Wang03] Wang, X. and S. G. Ziavras, "Parallel Direct Solution of Linear Equations on FPGA-Based Machines," *Proc. Int'l Parallel and Distributed Processing Symp.*, 2003.
[Wood08] Woods, R., J. McAllister, G. Lightbody, and Y. Yi, *FPGA-Based Implementation of Signal Processing Systems*, Wiley, 2008.

15. Biologically-Inspired Parallel Algorithms and Architectures (Assigned to: TBD)
[Furb09] Furber, S., "Biologically-Inspired Massively-Parallel Architectures—Computing Beyond a Million Processors," *Proc. 9th Int'l Conf. Application of Concurrency to System Design*, 2009, pp. 3-12.
[Lewi09] Lewis, A., S. Mostaghim, and M. Randall (eds.), *Biologically-Inspired Optimization Methods*, Springer, 2009.

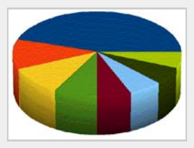
Poster Presentation Tips [Does not apply to winter 2013]



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 = [59, 93], Mean = 73, Median = 71
HW2 grades: Range = [15, 81], Mean = 50, Median = 47
HW3 grades: Range = [42, 80], Mean = 70, Median = 75
HW4 grades: Range = [16, 86], Mean = 60, Median = 68
HW5 grades: Range = [28, 90], Mean = 68, Median = 72
Overall HW grades: Range = [32, 84], Mean = 64, Median = 66
[Does not apply to winter 2013] Midterm exam grades: Range = [xx, xx], Mean = xx, SD = xx, Median = xx

Research paper and presentation grades: Range = [55, 80], Mean = 70, Median = 70

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, revised 1st ed., 2012. 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**).
Offering of ECE 254B in fall 2010 (PDF file)
Offering of ECE 254B in fall 2008 (PDF file)
Offerings of ECE 254B from 2000 to 2006 (PDF file)