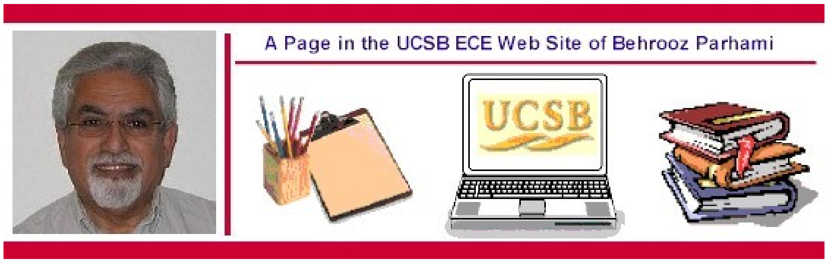

ECE 254B: Information on Previous Offerings

Behrooz Parhami: 2008/08/27 || 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 254B: Parallel Processing

Previous offerings of the course

- ECE 254B: Spring Quarter 2006
- ECE 254B: Spring Quarter 2005
- ECE 254B: Winter Quarter 2004
- ECE 254B: Winter Quarter 2003
- ECE 254B: Winter Quarter 2002
- ECE 254B: Spring Quarter 2001
- ECE 254B: Spring Quarter 2000

Return to: Top of this page

ECE 254B: Spring Quarter 2006

Course: ECE 254B – Advanced Computer Architecture: Parallel Processing, University of California, Santa Barbara, Spring 2006, Enrollment Code 10835

Catalog entry: **254B. Advanced Computer Architecture: Parallel Processing. (4) PARHAMI.** *Prerequisite: 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. (S)

Instructor: Behrooz Parhami, Room 5155 Engineering I, Phone 805-893-3211, parhami at ece dot ucsb dot edu

Meetings: MW 10:00-11:30, Phelps Hall, Room 1431

Consultation: Open office hours, held in Room 5155 Engineering I – M 11:30-1:00, W 3:30-5:00

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.

Prerequisite: Basic computer architecture at the level of ECE 154; an advanced architecture course, such as ECE 254A, would be helpful but can be waived.

References: *Required Text*–Parhami, B., *Intro. Parallel Processing: Algorithms and Architectures*, Plenum, 1999. Click on the title to view detailed information about the textbook, its errata, and downloadable PowerPoint presentations used in course lectures.
Supplement 1, optional – Dally, W.J. and B.P. Towles, *Interconnection Networks: Principles and Practices*, Morgan Kaufmann, 2004. Get more info via www.mkp.com (search for the title).
Supplement 2, optional – Because the focus of this course is on architecture and its interplay with algorithms, *Sourcebook of Parallel Computing* (ed. by J. Dongarra et al, Morgan Kaufmann, 2003), which deals primarily with software and application topics, constitutes helpful supplementary reading. Get more info via www.mkp.com (search for the title).
Journals – *IEEE Trans. Computers*, *IEEE Trans. Parallel and Distributed Systems*, *J. Parallel & Distributed Computing*, *Parallel Computing*, *Parallel Processing Letters*. Also, *IEEE Computer* and *IEEE Concurrency* (the latter ceased publication in late 2000) for broad introductory papers.
Conferences – 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).
Electronic Resources at UCSB
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 these three components with the given weights:

25% -- Homework: see the "deadlines" column in course calendar for schedule.

25% -- Closed-book midterm exam: see the course calendar for date and coverage.

50% -- Research project participation and report: see the description below.

Research: The class will cooperate in investigating a well-defined problem relating to interconnection networks for parallel processing. A brief period in some class sessions will be devoted to defining the research problem, dividing up the work, reviewing the progress, and setting research goals. Please follow the research paper guidelines to format your research report. Students who obtain results of publishable quality will receive an "A" for the course, regardless of homework and midterm exam grades.

Calendar:

Course lectures, homework assignments, and exams have been scheduled as follows. This schedule will be strictly observed. Textbook chapters are specified for lectures and exams. It is a good idea to look over the covered chapter(s) *before* each lecture if possible.

Day & Date	Chap	Lecture Topic	Deadlines, Links, Notes
Mon. 4/3/06	1	Introduction to parallel processing	
Wed. 4/5/06	2	A taste of parallel algorithms	HW#1 posted (chaps. 1-4)
Mon. 4/10/06	3-4	Complexity and models	
Wed. 4/12/06	5	PRAM model and basic algorithms	Research project defined
Mon. 4/17/06	6	More shared-memory algorithms	HW #1 due, HW #2 posted (chaps. 5-8)
Wed. 4/19/06	7	Sorting and selection networks	
Mon. 4/24/06	8	Other circuit-level examples	Research project background
Wed. 4/26/06	9	Sorting on a 2D mesh or torus	HW #2 due, HW #3 posted (chaps. 9-12)
Mon. 5/1/06	10	Routing on a 2D mesh or torus	
Wed. 5/3/06	11	Other mesh/torus algorithms	Project assignments
Mon. 5/8/06	12	Mesh variations and extensions	HW #3 due
Wed. 5/10/06	1-12	Midterm exam: closed book (10:00-11:50 AM)	Note the extended time
Mon. 5/15/06	13	Hypercubes and their algorithms	
Wed. 5/17/06	14	Sorting and routing on hypercubes	HW #4 posted (chapters 13-17)
Mon. 5/22/06	15	Other hypercubic architectures	
Wed. 5/24/06	16	Other interconnection networks	Project plan and references due
Mon. 5/29/06		Holiday (Memorial day): No lecture	
Wed. 5/31/06	17	Emulation and scheduling	HW #4 due
Mon. 6/5/06	18	Data storage, input, and output	
Wed. 6/7/06	19-20	Reliability and system issues	
Wed. 6/14/06			Final project report due by 5:00 PM

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 254B s2006, Homework #1: Fundamental concepts (chapters 1-4, due Mon. 4/17/06)

Do the following problems from the textbook: 1.2 [10 pts.], 1.12b [10 pts], 2.6 [10 pts.], 2.14 [20 pts], 3.1 [20 pts.], 3.4 [15 pts.], 4.5 [15 pts.]

ECE 254B s2006, Homework #2: Shared-memory and circuit models (chapters 5-8, due Wed. 4/26/05)

Do the following problems from the textbook: 5.6 [15 pts.], 5.9 [10 pts.], 6.2 [15 pts.], 6.13 [15 pts.], 7.6 [15 pts.], 7.8 [15 pts], 8.7 [15 pts]

ECE 254B s2006, Homework #3: Mesh and mesh-based architectures (chapters 9-12, due Mon. 5/8/05)

Do the following problems from the textbook: 9.7 [20 pts.], 9.9 [10 pts.], 10.9 [20 pts.], 11.1 [15 pts.], 12.6 [20 pts.], 12.10 [15 pts]

ECE 254B s2006, Homework #4: Interconnection networks (chapters 13-17, due Wed. 5/31/05)

Do the following problems from the textbook or defined below: 13.3 [15 pts.], 13.B [15 pts.], 14.8 [20 pts.], 14.C [15 pts.], 15.9 [15 pts.], 16.7 [20 pts]

13.B Embedding multigrids and pyramids into hypercubes
a. Show that a 2D multigrid whose base is a 2^{q-1}-node square mesh, with q odd and q ³ 5, and hence a pyramid of the same size, cannot be embedded in a q-cube with dilation 1.
b. Show that the 21-node 2D multigrid with a 4?´?4 base can be embedded in a 5-cube with dilation 2 and congestion 1 but that the 21-node pyramid cannot.

14.C Hypercube routing algorithm
Consider a q-cube in which each bidirectional link has been replaced by two unidirectional links in opposite directions. Dividing the unidirectional links into "upward" links (connecting x to y with $x < y$) and "downward" links (connecting x to y with $x > y$), we can view the hypercube as two unidirectional cubes formed by the upward and downward links. A proposed routing algorithm works as follows. The route taken by a message going from node u to node w passes through an intermediate node v such that the path from u to v in the upward cube and the path from v to w is in the downward cube (v can be the same as u or w).
a. Show that the proposed algorithm always chooses a shortest path from u to w.
b. Prove that the algorithm is free from deadlock if used for wormhole routing.
c. How can we benefit from this algorithm in devising a deadlock-free routing algorithm for a bidirectional hypercube?

ECE 254B s2006, Reading list for midterm exam

Remember that the midterm exam ends at 11:50 AM (not 11:30 as our normal classes). The following sections are excluded from the first 12 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, 11.4, 11.6, 12.6.

ECE 254B s2006, Reading list for final exam

Not applicable to this quarter.

ECE 254B s2006, Research Project Notes

Due to exchange of original ideas that are as yet unpublished, most research project notes will be handed out in hard-copy format rather than posted to this Web page.

Return to: Top of this page

CE 254B: Spring Quarter 2005

This area reserved for important course announcements: **Research paper and final course grades have been assigned. An e-mail message, with comments on the research paper, will be sent to each student no later than Friday, June 17, 2005. Have a nice summer!**

Course: ECE 254B – Advanced Computer Architecture: Parallel Processing, University of California, Santa Barbara, Spring 2005, Enrollment Code 48413

Catalog entry: **254B. Advanced Computer Architecture: Parallel Processing.** **(4) PARHAMI.** *Prerequisite: 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. (S)

Instructor: Behrooz Parhami, Room 5155 Engineering I, Phone 805-893-3211, parhami at ece dot ucsb dot edu

Meetings: MW 12:00-1:30, Phelps Hall, Room 1431

Consultation: Open office hours, held in Room 5155 Engineering I – M 2:00-3:30, W 3:30-5:00

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.

Prerequisite: Basic computer architecture at the level of ECE 154; an advanced architecture course, such as ECE 254A, would be helpful but can be waived.

References: *Required Text*–Parhami, B., *Intro. Parallel Processing: Algorithms and Architectures*, Plenum, 1999. Use the link to view detailed information about the textbook and its errata.
I found out in early April, 2005, that the UCSB bookstore is selling our textbook at a much higher price than what is available from multiple on-line sources (they have priced it even higher than the book's list price). The price quoted to me is almost double what was posted on the empty shelves before the books arrived at the bookstore. That posted price was the basis of my recommendation to you to wait, rather than purchase the book on-line. This is quite surprising and goes against the spirit of college bookstores that use their bulk purchasing power to get the best prices for students. You may want to shop around for a lower price. If this practice of getting the textbooks two weeks into the quarter and pricing them ridiculously high continues, I will boycott the bookstore and ask my students to buy all their books on-line.
Supplement 1, optional – Dally, W.J. and B.P. Towles, *Interconnection Networks: Principles and Practices*, Morgan Kaufmann, 2004. Get more info via www.mkp.com (search for the title).
Supplement 2, optional – Because the focus of this course is on architecture and its interplay with algorithms, *Sourcebook of Parallel Computing* (ed. by J. Dongarra et al, Morgan Kaufmann, 2003), which deals primarily with software and application topics, constitutes helpful supplementary reading. Get more info via www.mkp.com (search for the title).
Journals – *IEEE Trans. Computers*, *IEEE Trans. Parallel and Distributed Systems*, *J. Parallel & Distributed Computing*, *Parallel Computing*, *Parallel Processing Letters*. Also, *IEEE Computer* and *IEEE Concurrency* (the latter ceased publication in late 2000) for broad introductory papers.
Conferences – 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).
Electronic Resources at UCSB
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 these three components with the given weights:

25% -- Homework: see the “deadlines” column in course calendar for schedule.

25% -- Closed-book midterm exam: see the course calendar for date and coverage.

50% -- Research project participation and report: see the description below.

Research: The class will cooperate in investigating a well-defined problem relating to interconnection networks for parallel processing. A brief period in some class sessions will be devoted to defining the research problem, dividing up the work, reviewing the progress, and setting research goals. Please follow the research paper guidelines to format your research report. Students who obtain results of publishable quality will receive an "A" for the course, regardless of homework and midterm exam grades.

Calendar:

Course lectures, homework assignments, and exams have been scheduled as follows. This schedule will be strictly observed. Textbook chapters are specified for lectures and exams. It is a good idea to look over the covered chapter(s) *before* each lecture if possible.

Day & Date	**Chap**	**Lecture Topic**	**Deadlines, Links, Notes**
Mon. 3/28/05	1	Introduction to parallelism	
Wed. 3/30/05	2	A taste of parallel algorithms	HW#1 posted (chaps. 1-4)
Mon. 4/4/05	3-4	Complexity and models	
Wed. 4/6/05	5	PRAM model and basic algorithms	Research project defined
Mon. 4/11/05	6	More shared-memory algorithms	HW #1 due, HW #2 posted (chaps. 5-8)
Wed. 4/13/05	7	Sorting and selection networks	Extended due date for HW#1
Mon. 4/18/05	8	Other circuit-level examples	Research project background
Wed. 4/20/05	9	Sorting on a 2D mesh or torus	HW #2 due, HW #3 posted (chaps. 9-12)
Mon. 4/25/05	10	Routing on a 2D mesh or torus	
Wed. 4/27/05	11	Other mesh/torus algorithms	Project assignments
Mon. 5/2/05	12	Mesh variations and extensions	HW #3 due
Wed. 5/4/05	1-12	Midterm exam: closed book (12-2 PM)	Note the extended time
Mon. 5/9/05	13	Hypercubes and their algorithms	
Wed. 5/11/05	14	Sorting and routing on hypercubes	HW #4 posted (chapters 13-17)
Mon. 5/16/05	15	Other hypercubic architectures	
Wed. 5/18/05	16	Other interconnection networks	Project plan and references due
Mon. 5/23/05	17	Emulation and scheduling	
Wed. 5/25/05	18	Data storage, input, and output	HW #4 due
Mon. 5/30/05		Holiday (Memorial day): No lecture	
Wed. 6/1/05	19-20	Reliability and system issues	
Wed. 6/13/05		(Extended from the due date of 6/8/05)	Final project report due by 12:00 noon

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.

Course Grade Statistics

Item	**Out of**	**Max**	**Mean**	**Median**	**Min**	**SD**
HW #1	100	66	46	44	35	n/a
HW #2	100	77	55	47	44	n/a
HW #3	100	71	58	66	28	n/a
HW #4	100	98	77	63	53	n/a
HW avg.	100	78	61	62	42	n/a
Research	100	85	75	78	60	n/a
Midterm	100	88	53	49	16	n/a
Course*	100	82	70	70	58	n/a
Course	A (4.0)	A	3.5	3.5	B	n/a

* Course = 25%(HW avg.) + 25%(Midterm) + 50%(Research)

ECE 254B s2005, Homework #1: Fundamental concepts (chapters 1-4, due Mon. 4/11/05)

Do the following problems from the textbook or defined below: 1.6 [20 pts.], 2.1 [20 pts.], 2.J [15 points], 3.9 [10 pts.], 3.13 [15 pts.], 4.12 [20 pts.]

Problem 2.J An integer sequence S of length n is stored in the n leaves of a complete binary tree. Develop an efficient parallel algorithm that determines whether the sequence is sorted in nondescending order from left to right. If the sequence is not nondescending, the algorithm should also flag each out-of-order element (one that is less than the element to its left).

ECE 254B s2005, Homework #2: Shared-memory and circuit models (chapters 5-8, due Wed. 4/20/05)

Do the following problems from the textbook or defined below: 5.1 [10 pts.], 5.8 [15 pts], 6.7 [20 pts.], 7.5 [20 pts.], 7.E [15 pts.], 8.4 [20 pts.]

Problem 7.E Selection networks – An (n, k)-selector is an n-input, single-output network of comparators that selects the kth smallest of the n inputs. For example, an $(n, 1)$-selector outputs the smallest element and an (n, n)-selector yields the largest element. (a) Prove or disprove that the zero-one principle is also valid for selection networks. (b) Design delay-minimal and cost-minimal (5, 2)- and (5, 3)-selectors. (c) Prove or disprove that the delay of an (n, k)-selector is strictly less than that of an n-sorter. (d) Prove or disprove that the cost of an (n, k)-selector is strictly less than that of an n-sorter.

ECE 254B s2005, Homework #3: Mesh and mesh-based architectures (chapters 9-12, due Mon. 5/2/05)

Do the following problems from the textbook or defined below: 9.4 [20 pts.], 9.8 [15 pts.], 10.11 [15 pts], 10.B [20 pts.], 11.E [15 pts.], 12.1 [15 pts.]

Clarification for Problem 12.1: In part b, the assumption is that in any given step, all processors must communicate along the same dimension. This is sometimes referred to as *uniaxis communication*.

Problem 10.B Interval routing in 2D mesh – Consider an interval routing scheme, as defined in problem 10.14, with the intervals attached to the four mesh links for node (i, j) corresponding to the submeshes defined by bounds on row and column numbers for the destination node (k, l) as follows: above, if $k < i$, l ³ j; right, if k ³ i, $l > j$; below, if $k > i$, l £ j; left, if k £ i, $l < j$. (a) Draw a few sample paths with various source destination pairs for the above mesh routing algorithm and state the general properties of the paths selected. (b) Compare the algorithm to row-first routing with respect to routing delay and buffer requirements in the worst case. (c) Compared to row-first routing, is the traffic going through the links more balanced or less balanced in this algorithm? (d) Is the algorithm deadlock-free if used with wormhole routing? Why or why not? (e) Modify the above regions or intervals for the case of a square 2D torus to achieve balance in the traffic for the various links.

Problem 11.E FIR filter implemented as a linear array – A finite impulse response (FIR) filter produces a sequence of outputs y_j in response to the sequence of inputs x_i such that y_j = å $a_k x_{j-k}$, where the a_k are the filter's coefficients and the first p – 1 outputs are ignored. Show how a linear array of p processors can perform this computation, with the input x_t accepted, and the output y_t produced, in cycle 2t.

ECE 254B s2005, Homework #4: Interconnection networks (chapters 13-17, due Wed. 5/25/05)

Do the following problems from the textbook or defined below: 13.7 [20 pts.], 13.F [15 pts.], 14.10 [15 pts.], 15.13 [15 pts.], 16.5 [20 pts.], 17.13 [15 pts.]

Problem 13.F 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

ECE 254B s2005, Reading list for midterm exam

Remember that the midterm exam ends at 2:00 PM (not 1:30 as our normal classes). The following sections are excluded from the first 12 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, 8.6, 9.6, 11.4, 12.6 (pp. 250-253).

ECE 254B s2005, Reading list for final exam

Not applicable to this quarter.

ECE 254B s2005, Research Project Notes

Due to exchange of original ideas that are as yet unpublished, most research project notes will be handed out in hard-copy format rather than posted to this Web page.

Return to: Top of this page

ECE 254B: Winter Quarter 2004

This area reserved for important course announcements: Homework 5 (last homework) has been posted below.

Course:	ECE 254B – Advanced Computer Architecture: Parallel Processing, University of California, Santa Barbara, Winter 2004, Enrollment Code 11882
Catalog entry:	**254B. Advanced Computer Architecture: Parallel Processing.** **(4) PARHAMI.** *Prerequisite: 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. (S)
Instructor:	Behrooz Parhami, Room 5155 Engineering I, Phone 805-893-3211, parhami at ece dot ucsb dot edu
Meetings:	MW 4:00-5:30, Phelps Hall, Room 1431
Consultation:	Open office hours, held in Room 5155 Engineering I – M 11:00-12:30, W 12:30-2:00
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.
Prerequisite:	Basic computer architecture at the level of ECE 154; an advanced architecture course, such as ECE 254A, would be helpful but can be waived.
References:	*Required Text*–Parhami, B., *Intro. Parallel Processing: Algorithms and Architectures*, Plenum, 1999. Use the link to view detailed information about the textbook and its errata. *Supplement 1, optional* – Because the focus of this course is on architecture and its interplay with algorithms, *Sourcebook of Parallel Computing* (ed. by J. Dongarra et al, Morgan Kaufmann, 2003), which deals primarily with software and application topics, constitutes helpful supplementary reading. Get more info via www.mkp.com (search for the title). *Supplement 2, optional* – Dally, W.J. and B.P. Towles, *Interconnection Networks: Principles and Practices*, Morgan Kaufmann, 2004. Get more info via www.mkp.com (search for the title). *Journals* – *IEEE Trans. Computers*, *IEEE Trans. Parallel and Distributed Systems*, *J. Parallel & Distributed Computing*, *Parallel Computing*, *Parallel Processing Letters*. Also, *IEEE Computer* and *IEEE Concurrency* (the latter ceased publication in late 2000) for broad introductory papers. *Conferences* – 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).

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

20% -- Homework: see the "deadlines" column in course calendar for schedule.

20% -- Research project participation: see the description below.

20% -- Closed-book midterm exam: see the course calendar for date and coverage.

40% -- Open-book final exam: see the course calendar for date, time, and coverage.
Due to the very small class size, it was decided to increase the research orientation of the course and to replace the final exam grade with the research project grade.

Research: The class will cooperate in investigating a well-defined problem relating to interconnection networks. A brief period in some class sessions will be devoted to defining the research problem, dividing up the work, reviewing the progress, and setting research goals. Please follow the research paper guidelines to format your research report. Students who obtain results of publishable quality will receive an "A" for the course, regardless of homework and exam grades.

Calendar:

Course lectures, homework assignments, and exams have been scheduled as follows. This schedule will be strictly observed. Textbook chapters are specified for lectures and exams. It is a good idea to look over the covered chapter(s) *before* each lecture if possible.

Day & Date	Chap	Lecture Topic	Deadlines, Links, Notes
Mon. 1/5/04	1	Introduction to parallelism	
Wed. 1/7/04	2	A taste of parallel algorithms	
Mon. 1/12/04	3-4	Complexity and models	HW#1 posted (chapters 1-4)
Wed. 1/14/04	5	PRAM and basic algorithms	Research project defined
Mon. 1/19/04		Holiday (MLK birthday): No lecture	
Wed. 1/21/04	6	More shared-memory algorithms	HW #2 posted (5-6), HW #1 due
Mon. 1/26/04	7	Sorting and selection networks	Research project background
Wed. 1/28/04	8	Other circuit-level examples	HW #3 posted (7-8)
Mon. 2/2/04	9	Sorting on a 2D mesh or torus	HW #2 due, Project assignments
Wed. 2/4/04	10	Routing on a 2D mesh or torus	
Mon. 2/9/04	11-12	Other mesh algorithms and variations	HW #4 posted (9-12), HW #3 due
Wed. 2/11/04	1-10	Midterm exam: closed book (4-6 PM)	Note the extended time
Mon. 2/16/04		Holiday (Pres. day): No lecture	
Wed. 2/18/04	13	Hypercubes and their algorithms	
Mon. 2/23/04	14	Sorting and routing on hypercubes	HW #5 posted (13-16), HW #4 due
Wed. 2/25/04	15-16	Other parallel architectures	Project abstract and references due
Mon. 3/1/04	17	Emulation and scheduling	
Wed. 3/3/04	18	Data storage, input, and output	HW #5 due
Mon. 3/8/04	19-20	Reliability and systems issues	
Wed. 3/10/04	21-24	Implementation topics and examples	Final project report due
Fri. 3/12/04	1-20	Final exam: open book (2:00-5:00 PM)	Cancelled; replaced by research

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 254B w2004, Homework #1: Fundamental concepts (chapters 1-4, due Wed. 1/21/03)

Do the following problems from the textbook or defined below: 1.D, 2.2, 2.5, 3.7ab, 4.7

In Problem 3.7b, assume that n is an even power of 2.

In Problem 4.7c, assume that p is divisible by s; i.e., $p = ks$.

Problem 1.D: Parallel and pipelined processing -- In a particular plant, laptop computers are assembled in a 3-stage pipelined process, with stage delays all being the same. There are three separate assembly lines, each with an input-to-output latency of 30 minutes. Plot the speed-up due to having three assembly lines, relative to using a single assembly line, when n laptop computers (1 £ n £ 10) are to be assembled. Then derive a general speed-up formula for a p-pipeline, h-segment system relative to a single-pipeline system.

ECE 254B w2004, Homework #2: Abstract shared memory (chapters 5-6, due Mon. 2/2/03)

Do the following problems from the textbook or defined below: 5.2, 5.5, 5.10, 6.2, 6.B.

Problem 6.B: Plurality voting on the PRAM -- Devise an efficient PRAM algorithm for “plurality voting”. There are p processors and a vector $x[1..p]$ of data values. The algorithm should compute a pair of results y and w such that the value $y = x[j]$ appears w times in x and no other value has more appearances. For example, x = 5 7 4 5 4 should yield $y = 5$ or $y = 4$ with $w = 2$, since both 5 and 4 appear twice in x, whereas 7 appears only once. (a) Describe your algorithm in high-level terms, using known subalgorithms if needed. (b) Analyze the worst-case running time of the algorithm. (c) How much simpler can the algorithm become if standard “majority voting” is desired? Assume that a majority always exists and that only the single result y is to be computed.

ECE 254B w2004, Homework #3: Circuit model of parallelism (chapters 7-8, due Mon. 2/9/03)

Do the following problems from the textbook or defined below: 7.7, 7.D, 7.O, 8.6abc, 8.C.

Problem 7.D: Delay and cost of sorting networks -- Let $C_{min}(n)$ and $D_{min}(n)$ be the minimal cost and minimal delay for an n-sorter constructed only of 2-sorters; the two lower bounds may not be realizable in a single circuit. **(a)** Prove or disprove: $D_{min}(n)$ is a strictly increasing function of n; i.e., $D_{min}(n + 1) > D_{min}(n)$. **(b)** Prove or disprove: $C_{min}(n)$ is a strictly increasing function of n; i.e., $C_{min}(n + 1) > C_{min}(n)$.

Problem 7.O: Batcher sorting networks -- Verify that based on the cost recurrence given in Section 7.4, the cost of a Batcher sorting network is $n(\log_2 n)^2/4$ to within lower-order terms. Then, using the substitution method of Section 3.6, find an exact expression for $C(n)$ that includes all lower-order terms.

Problem 8.C: Distributed dictionary machines -- The p processors of a linear array hold n records in sorted order such that the leftmost processor holds the records with the smallest keys. As the processors compute, they produce new records or cause records to be deleted. Discuss in high-level terms a protocol to be followed by the processors for insertion and deletion of records such that the sorted order of the records is maintained and the processor loads (number of records held) are roughly balanced over the long run.

ECE 254B w2004, Homework #4: Mesh and mesh-based architectures (chapters 9-12, due Mon. 2/23/03)

Do the following problems from the textbook or defined below: 9.3, 10.5, 10.F, 11.2, 12.2.

Problem 10.F: k-to-1 routing on a 2D mesh -- Show that if each processor of p-processor square mesh begins with one packet and if up to k packets can have the same destination, the greedy routing algorithm can take $Q(min(kp^{1/2}, p))$ steps in the worst case.

ECE 254B w2004, Homework #5: Low-diameter architectures (chapters 13-16, due Wed. 3/3/03)

Do the following problems from the textbook or defined below: 13.8, 13.B, 14.12, 15.1, 16.1.

Problem 13.B: Embedding multigrids and pyramids into hypercubes -- **(a)** Show that a 2D multigrid whose base is a 2^{q-1}-node square mesh, with q odd and q ³ 5, and hence a pyramid of the same size, cannot be embedded in a q-cube with dilation 1. **(b)** Show that the 21-node 2D multigrid with a 4 ´ 4 base can be embedded in a 5-cube with dilation 2 and congestion 1 but that the 21-node pyramid cannot.

Note on Problem 15.1: In part f, ignoring the hint can lead to a simpler solution.

ECE 254B w2004, Reading list for midterm exam

Remember that the midterm exam ends at 6:00 PM (not 5:30 as our normal classes). 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, 8.6, 9.6, 10.6.

ECE 254B w2004, Reading list for final exam

Not applicable to this quarter.

ECE 254B w2004, Research Project Notes

Due to exchange of original ideas that are as yet unpublished, most research project notes will be handed out in hard-copy format rather than posted to this Web page.

Return to: Top of this page

ECE 254B: Winter Quarter 2003

This area reserved for important course announcements: **The final exam and course grade stats have been posted below. If you would like to be notified of your final and course grades, please send e-mail with an explicit request for e-mail transmission of your grades. To look at your final exam paper and my solutions, stop by during my office hours (same as this quarter for spring 2003). I hope that you enjoyed the course and took with you knowledge that is of lasting value in our fast-moving field.**

Course:	ECE 254B – Advanced Computer Architecture: Parallel Processing, University of California, Santa Barbara, Winter 2003, Enrollment Code 11718
Catalog entry:	**254B. Advanced Computer Architecture: Parallel Processing. (4) PARHAMI.** *Prerequisite: 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. (S)
Instructor:	Behrooz Parhami, Room 5155 Engineering I, Phone 805-893-3211, parhami at ece dot ucsb dot edu
Meetings:	MW 4:00-5:30, Building 387 (on UCen Rd., west of Psychology), Room 103
Consultation:	Open office hours, held in Room 5155 Engineering I – M 11:00-12:30, W 12:30-2:00
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.

Prerequisite: Basic computer architecture at the level of ECE 154; an advanced architecture course, such as ECE 254A, would be helpful but can be waived.

References: *Required Text*–Parhami, B., *Intro. Parallel Processing: Algorithms and Architectures*, Plenum, 1999. Use the link to view detailed information about the textbook and its errata.
Supplement – Because the focus of this course is on architecture and its interplay with algorithms, *Sourcebook of Parallel Computing* (ed. by J. Dongarra et al, Morgan Kaufmann, 2003), which deals primarily with software and application topics, constitutes helpful supplementary reading. Get more info via www.mkp.com (search for the title).
Journals – IEEE Trans. Computers, *IEEE Trans. Parallel and Distributed Systems*, *J. Parallel & Distributed Computing*, *Parallel Computing*, *Parallel Processing Letters*. Also, *IEEE Computer* and *IEEE Concurrency* (the latter ceased publication in late 2000) for broad introductory papers.
Conferences – 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).

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

25% -- Homework: see the “deadlines” column in course calendar for schedule.

25% -- Closed-book midterm exam: see the course calendar for date and coverage.

50% -- Open-book final exam: see the course calendar for date, time, 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 parallel processing or do original research on a selected topic. A publishable report earns an “A” for the course, regardless of homework and midterm grades. See the “Research deadlines” column in the course calendar for schedule and due dates. Please follow the research paper guidelines to format your paper.

Calendar:

Course lectures, homework assignments, and exams have been scheduled as follows. This schedule will be strictly observed. Textbook chapters are specified for lectures and exams. It is a good idea to look over the covered chapter(s) *before* each lecture if possible.

Day & Date	Chap	Lecture Topic	Deadlines, Links, Notes
Mon. 1/6/03	1	Introduction to parallelism	
Wed. 1/8/03	2	A taste of parallel algorithms	
Mon. 1/13/03	3-4	Complexity and models	HW#1 posted (chapters 1-4)
Wed. 1/15/03	5	PRAM and basic algorithms	
Mon. 1/20/03		Holiday (MLK birthday): No lecture	HW #2 posted (5-6)
Wed. 1/22/03	6	More shared-memory algorithms	Research topic defined, HW #1 due
Mon. 1/27/03	7	Sorting and selection networks	HW #3 posted (7-8), HW #2 due
Wed. 1/29/03	8	Other circuit-level examples	
Mon. 2/3/03	9	Sorting on a 2D mesh or torus	HW #4 posted (9-10), HW #3 due
Wed. 2/5/03	10	Routing on a 2D mesh or torus	Preliminary references due
Mon. 2/10/03	11-12	Other mesh algorithms and variations	HW #4 due
Wed. 2/12/03	1-10	Midterm exam: closed book (4-6 PM)	Note the extended time
Mon. 2/17/03		Holiday (Pres. day): No lecture	HW #5 posted (11-12)
Wed. 2/19/03	13	Hypercubes and their algorithms	
Mon. 2/24/03	14	Sorting and routing on hypercubes	HW #6 posted (13-14), HW #5 due
Wed. 2/26/03	15-16	Other parallel architectures	Paper title and references due
Mon. 3/3/03	17	Emulation and scheduling	HW #7 posted (15-18), HW #6 due
Wed. 3/5/03	18	Data storage, input, and output	Preliminary paper abstract due
Mon. 3/10/03	19-20	Reliability and systems issues	HW #7 due
Wed. 3/12/03	21-24	Implementation issues and examples	Full paper due 5:00 PM, Fri. 3/14/03
Mon. 3/17/03	1-20	Final exam: open book (2-5 PM)	Place: Room 2162, Engr. I

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.

Course Grade Statistics

Item	Out of	Max	Mean	Median	Min	SD
HW #1	100	82	72	72	65	6
HW #2	100	74	62	61	51	7
HW #3	100	94	76	84	42	18
HW #4	100	80	66	63	58	7
HW #5	100	95	59	57	36	18
HW #6	100	96	57	54	29	21
HW #7	100	95	59	58	43	17
Midterm	100	89	66	63	51	14
Final/Paper	100	72	65	68	50	8
Course*	100	78	65	63	52	8
Course	A (4.0)	A+	3.3	B	C	0.8

* Course = 25%(HW total / 7) + 25%(Midterm) + 50%(Final or Paper)

ECE 254B w2003, Homework #1: Fundamental concepts (chapters 1-4, due Wed. 1/22/03)

Do the following problems from the textbook or defined below: 1.1, 1.8, 2.7, 3.8, 3.A

Problem 3.A Comparing algorithms Suppose two different parallel algorithms for solving a given problem lead to the following recurrences for the computation time. Which algorithm is asymptotically faster and why?

a. $T(n) = 2T(n/2) + n/(\log_2 n)^2$

b. $T(n) = T(n/2) + n(\log_2 n)^2$

ECE 254B w2003, Homework #2: Abstract shared memory (chapters 5-6, due Mon. 1/27/03)

Do the following problems from the textbook or defined below: 5.3, 5.12, 6.3, 6.4, 6.D

Problem 6.D Geometric problems on circles The x and y coordinates for the centers of n circles and their radii are given as the n-vectors X, Y, and R in the shared memory of an n-processor PRAM.

a. Design an efficient parallel algorithm to determine if any pair of these circles intersect.
b. Evaluate the worst-case complexity and the efficiency of your algorithm.
c. Specify the PRAM submodel used for your algorithm and discuss simplifications or complications if other submodels are assumed.
d. Discuss if the solution can be speeded up by using more than n processors.
e. Discuss how the solution is slowed down if p processor are used, where $p < n$.

ECE 254B w2003, Homework #3: Circuit model of parallelism (chapters 7-8, due Mon. 2/3/03)

Do the following problems from the textbook or defined below: 7.2ab, 7.3, 7.9, 8.9, 8.A

Problem 8.A PRAM parallel search revisited In Section 8.1, it is shown that the speedup achieved by a p-processor PRAM in searching a sorted list of size n is about $\log_2 p$ which is not impressive. However, any such application is unlikely to be dealing only with a static list. More likely, items have to be added to, and deleted from, the sorted list dynamically. With the same assumptions as in the aforementioned discussion, do the following:

a. Propose and analyze an efficient parallel algorithm for adding an item to the sorted list.
b. Repeat part a for the deletion operation.
c. What are the speedups in parts a and b compared to the sequential solutions?
d. An application involves a stream of add, delete, and search operations. These operations are randomly intermixed, with a fraction a being adds, a fraction d being deletes, and the rest (i.e., $1 - a - d$) consisting of search operations. What is the expected overall speedup for this application if run on the PRAM using the algorithms above?
e. Under what conditions is linear $O(p)$ speedup possible in part d?
f. An alternative to keeping the list in sorted order is to keep it in arbitrary order. Hence we get four implementation options corresponding to sorted/unsorted list and sequential/parallel solution. Redo part d, computing the speed-up of each of the two parallel alternatives with respect to the best sequential solution.

Special notes on the solution to Problem 8.A: A common error in the solutions offered for this homework was averaging of speedups for various parts of an algorithm to derive the overall speedup. The following example shows why this leads to incorrect results. Consider an algorithm with two parts. Alternative A does the first part in 5 s and the second part in 1 s. Alternative B does the first part in 1 s and the second part in 5 s. Both solutions take 6 s, assuming the two parts are equally weighted (are always done together). Speedup of A over B is thus 1.0. Speedups for the two parts are 0.2 and 5.0, when viewed in isolation. The average of the two speedups is 2.6 which is quite different from the correct answer of 1.0.

ECE 254B w2003, Homework #4: Data movement on 2D arrays (chapters 9-10, due Mon. 2/10/03)

Do the following problems from the textbook or defined below: 9.10a, 9.Ba, 10.4, 10.12, 10.14abc

Problem 9.B Selection on a 2D mesh

a. Show why the following proposed algorithm for selection on a 2D mesh is not viable. The mesh is divided into four quadrants. The median of the elements in each quadrant is determined using the same selection algorithm recursively. Then the second largest value s among the 4 medians is found using two semigroup computations. Next, each processor determines if its value is <, >, or equal to s. The number of elements in each class is found using 3 semigroup computations. Finally, it is decided how the computation should continue, the elements among which selection must be performed are compressed into a square submesh in the upper left corner, and the next phase is begun.
b. Would adding a global bus, or row/column buses, to the mesh change the conclusion of part a?

ECE 254B w2003, Homework #5: Mesh algorithms and variants (chapters 11-12, due Mon. 2/24/03)

Do the following problems from the textbook or defined above/below: 9.Bb, 11.5, 11.9, 12.3, 12.C

Problem 12.C Meshes with diagonal links Consider an m ´ m mesh augmented with 2 diagonal links, one connecting processors (0, 0) and $(m - 1, m - 1)$ and the other connecting processors (0, m – 1) and (m – 1, 0).

a. Show that the diameter of such an augmented mesh is no more than 1.5m.

b. Can you derive the exact diameter of the m ´ m mesh augmented with diagonal links?

c. Derive the bisection width and suggest a lower bound on the time for sorting.

d. Formulate an efficient algorithm for finding the maximum of m^2 numbers, stored one per processor. The algorithm should take advantage of the diagonal links.

ECE 254B w2003, Homework #6: The hypercube architecture (chapters 13-14, due Mon. 3/3/03)

Do the following problems from the textbook or defined below: 13.2, 13.A, 13.C, 14.5, 14.15

Problem 13.A Embedding X-trees into hypercubes Extend the result of Problem 13.6 to X-trees. [Correction to the statement of Problem 13.6: Node labels begin from 1 and end with 2^q – 1.]

Problem 13.C Embedding into hypercubes

a. Show that even after removing a quarter of the leaf nodes from a (2^q – 1)-node complete binary tree, where q ³ 5, the resulting graph is not a subgraph of the q-cube.

b. Show that a forest (collection of independent trees) composed of 2^h complete binary trees, each having $2^{q–h}$ – 1 nodes, is a subgraph of the q-cube when h ³ 1.

ECE 254B w2003, Homework #7: Other architectures and topics (chapters 15-18, due Mon. 3/10/03)

Do the following problems from the textbook or defined below: 15.2, 15.A, 16.12, 17.12, 18.12

Problem 15.A Shuffle-exchange graphs

a. Derive the diameter of an n-node shuffle-exchange graph.

b. Consider the (n/2)-node graph formed by merging each node u in the n-node shuffle-exchange graph with its binary complement node u'. Show that the resulting graph has degree 3 and diameter at most 1.5 $\log_2 n$.

c. Show that r perfect shuffles return an n-card deck (n even) to its original order iff $2r$ = 1 mod (n – 1).

d. Prove that 8 perfect shuffles are required and sufficient to restore a deck of 52 cards to its original order.

ECE 254B w2003, Reading list for midterm exam

Remember that the midterm exam ends at 6:00 PM (not 5:30 as our normal classes). 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, 8.6, 9.6.

ECE 254B w2003, Reading list for final exam

The final exam will emphasize the material covered after the midterm (Chapters 11-20). All midterm exclusions apply for the final as well. Additionally, the following sections are excluded from the final exam: 11.4, 11.6, 12.6, 13.5, last page of 14.3, 15.6, 16.3, 16.5, 16.6, last page of 17.2, 17.6, 18.6, 19.6, all of Chapter 20.

ECE 254B w2003, Possible Research Topics

For list of possible topics, please see the previous offerings of the course. An updated list of research topics will not be produced unless some students indicate an interest in pursuing the research option during this quarter.

Return to: Top of this page

ECE 254B: Winter Quarter 2002

This area reserved for important course announcements: I hope that you enjoyed the course. Course grades have been sent to the Registrar. Send e-mail to the instructor to request your final exam and course grades.

Course:	ECE 254B – Advanced Computer Architecture: Parallel Processing, University of California, Santa Barbara, Winter 2002, Enrollment Code 49965
Instructor:	Behrooz Parhami, Room 5155 Engineering I, Phone 805-893-3211, parhami at ece dot ucsb dot edu
Meetings:	MF 4:00-5:30, Room 1930 Buchanan
Consultation:	Open office hours, held in Room 5155 Engineering I – M 11-12, W 1-2, F 1-2
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.
Prerequisite:	Basic computer architecture at the level of ECE 154; an advanced architecture course, such as ECE 254A, would be helpful but is not required.
References:	*Required Text*–Parhami, B., *Intro. Parallel Processing: Algorithms and Architectures*, Plenum, 1999. Use the link to view detailed information about the textbook and its errata. *Journals* – *IEEE Trans. Computers*, *IEEE Trans. Parallel and Distributed Systems*, *J. Parallel & Distributed Computing*, *Parallel Computing*, *Parallel Processing Letters*. Also, *IEEE Computer* and *IEEE Concurrency* (the latter ceased publication in late 2000) for broad introductory papers. *Conferences* – 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).
Evaluation:	Students will be evaluated based on these three components with the given weights: 25% -- Homework: see the "deadlines" column in course calendar for schedule. 25% -- Closed-book midterm exam: see the course calendar for date and coverage. 50% -- Open-book final exam: see the course calendar for date, time, 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 parallel processing or do original research on a selected topic. A publishable report earns an "A" for the course, regardless of homework and midterm grades. See the "Research deadlines" column in the course calendar for schedule and due dates. Please follow the research paper guidelines to format your paper.

Calendar:

Course lectures, homework assignments, and exams have been scheduled as follows. This schedule will be strictly observed. Textbook chapters are specified for lectures and exams. It is a good idea to look over the covered chapter(s) *before* each lecture if possible.

Day & Date	Chap	Lecture Topic	Deadlines, Links, Notes
Mon. 1/7/02	1	Introduction to parallelism	
Fri. 1/11/02	2	A taste of parallel algorithms	
Mon. 1/14/02	3-4	Complexity and models	HW #1 posted (chapters 1-4)
Fri. 1/18/02	5	PRAM and basic algorithms	
Mon. 1/21/02		Holiday (MLK birthday): No lecture	HW #2 posted (5-6)
Fri. 1/25/02	6	More shared-memory algorithms	Research topic defined, HW #1 due
Mon. 1/28/02	7	Sorting and selection networks	HW #3 posted (7-8), HW #2 due
Fri. 2/1/02	8	Other circuit-level examples	
Mon. 2/4/02	9	Sorting on a 2D mesh or torus	HW #4 posted (9-10), HW #3 due
Fri. 2/8/02	10	Routing on a 2D mesh or torus	Preliminary references due
Mon. 2/11/02	11	Numerical 2D mesh algorithms	HW #4 due
Fri. 2/15/02	1-10	Midterm exam: closed book (4-6 PM)	Note the extended time
Mon. 2/18/02		Holiday (Pres. day): No lecture	HW #5 posted (11-12)
Fri. 2/22/02	12	Other mesh-related architectures	
Mon. 2/25/02	13	Hypercubes and their algorithms	HW #6 posted (13-14), HW #5 due
Fri. 3/1/02	14	Sorting and routing on hypercubes	Paper title and references due
Mon. 3/4/02	15-16	Other parallel architectures	HW #7 posted (15-18), HW #6 due
Fri. 3/8/02	17	Emulation and scheduling	Preliminary paper abstract due
Mon. 3/11/02	18	Data storage, input, and output	
Fri. 3/15/02	19-20	Reliability and systems issues	HW #7 due
Mon. 3/18/02	1-18	Final exam: open book (2-5 PM)	Place: Room 2162 Engineering I

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.

Course Grades

Item	Out of	Max	Mean	Median	Min	SD
HW #1	100	94	73	72	48	16
HW #2	100	77	64	66	40	14
HW #3	100	97	84	87	57	15
HW #4	100	88	72	67	58	13
HW #5	100	91	68	67	34	22
HW #6	100	95	65	57	34	27
HW #7	100	90	68	84	31	27
Midterm	100	78	48	42	30	19
Final/Paper	100	83	65	69	40	18
Course*	100	82	62	61	39	18
Course	A (4.0)	A	3.4	B+	B–	0.6

* Course = 25%(HW total / 7) + 25%(Midterm) + 50%(Final or Paper)

ECE 254B w2002, Homework #1: Fundamental concepts (chapters 1-4, due Fri. Jan. 25)

Do the following problems; all but problem 1.C, given below, are from the textbook.

1.9, 1.C, 2.11, 3.2, 3.6, 4.2

Problem 1.C Amdahl's law

A multiprocessor is built out of 5 GFLOPS processors. What is the maximum allowable sequential fraction of a job in Amdahl's formulation if the multiprocessor's performance is to exceed 1

TFLOPS?

ECE 254B w2002, Homework #2: Abstract shared memory (chapters 5-6, due Mon. Jan. 28)

Do the following problems; all but problem 5.C, given below, are from the textbook.

5.4, 5.11, 5.C, 6.14 (note the correction below)

Problem 6.14 -- replace the problem introduction with the following: Consider the butterfly memory access network depicted in Fig. 6.9, but with only 8 processors and 8 memory banks, one connected to each circular node in columns 0 and 3 of the butterfly.

Problem 5.C   Parallel prefix computation on a linked list

a. Outline the design of an efficient parallel prefix algorithm for the PRAM (no actual code is needed). The *n* data items are stored in the form of a doubly linked list. The array *D*[1..*n*] contains the data items while the integer arrays *F*[1..*n*] and *B*[1..*n*] hold the forward and backward links; i.e., the next and previous elements in sequence. The terminal list element in either direction points to itself. The computed prefix results should be stored in *P*[1..*n*] in the same linked format (i.e., with sequencing links given by *F* and *B*).

b. Which PRAM submodel is implied by your algorithm?

c. If the answer to part b is not EREW, then how can the algorithm be modified to run on this “weaker” submodel with little or no performance penalty?

ECE 254B w2002, Homework #3: Circuit model of parallelism (chapters 7-8, due Mon. Feb. 4)

Do the following problems; all but problem 7.L, given below, are from the textbook.

7.4, 7.13, 7.L, 8.5, 8.8

Problem 7.L   Synthesizing sorting networks

Show that given valid *n*-sorter, the following transformations lead to a valid 2*n*-sorter. (1) Replace each horizontal line by a pair of lines. (2) Replace each 2-sorter connected to a pair of the original lines by a 4-sorter connected to both pairs of lines replacing them. (3) Replace each 4-soter by the 5-comparator network of Fig. 7.4.

ECE 254B w2002, Homework #4: Data movement on 2D arrays (chapters 9-10, due Mon. Feb. 11)

Do the following problems; all but problems 9.A and 10.E, given below, are from the textbook.

9.1, 9.6, 9.A, 10.7, 10.E

Problem 9.A   The shearsort algorithm

Using the result of the analysis for optimized shearsort in Section 9.3:

a. Determine conditions under which shearsort would be faster for 2*m*^2 elements, stored one per processor, when run on a 2*m*-by-*m* mesh (2*m* rows, *m* columns) compared to an *m*-by-2*m* mesh (*m* rows, 2*m* columns).

b. Given *p* elements to be sorted, find the optimal mesh dimensions (*r* and *p*/*r*) that would minimize the sorting time.

Problem 10.E   Greedy routing on a 2D torus

Show that on an *r*-row *p*-processor 2D torus, the greedy row-first routing algorithm solves a 1-1 routing problem in no more than *r*/2 + *p*/(2*r*) steps.

ECE 254B w2002, Homework #5: Mesh algorithms and variants (chapters 11-12, due Mon. Feb. 25)

Do the following problems; all but problem 11.B, given below, are from the textbook.

11.8, 11.B, 12.4, 12.8, 12.11

Problem 11.B   Matrix algorithms on a linear array

a.   A *b*-banded or *b*-diagonal matrix is a matrix for which all the entries not contained in a single band of *b* diagonals are zero. Show how to multiply an *m*-vector by an *m* ´ *m* *b*-diagonal matrix in O(*m*) steps on a *b*-cell linear array.

b.   Draw a linear array and its associated data flow for solving an upper-triangular system of five linear equations.

c.   Draw a linear array and its associated data flow for solving a strictly-lower-triangular system of 5 linear equations (i.e., the main diagonal elements are also 0s).

ECE 254B w2002, Homework #6: The hypercube architecture (chapters 13-14, due Mon. Mar. 4)

Do the following problems from the textbook.

13.1, 13.6, 13.14, 14.4, 14.13

ECE 254B w2002, Homework #7: Other architectures and topics (chapters 15-18, due Fri. Mar. 15)

Do the following problems; all but problem 15.D, given below, are from the textbook.

15.3, 15.D, 16.7, 17.2, 18.11

Problem 15.D   Routing on Benes networks

Show edge-disjoint paths for each of the following permutations of the inputs (0, 1, 2, 3, 4, 5, 6, 7) on an 8-input Benes network.

a. (6, 2, 3, 4, 5, 1, 7, 0)

b. (5, 4, 2, 7, 1, 6, 3, 0)

c. (1, 7, 3, 4, 2, 6, 0, 5)

ECE 254B w2002, Reading list for midterm exam

Remember that the midterm exam ends at 6:00 PM (not 5:30 as our normal classes). 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, 8.6, 9.6.

ECE 254B w2002, Reading list for final exam

The final exam will emphasize the material covered after the midterm (Chapters 11-18). All midterm exclusions apply for the final as well. Additionally, the following sections are excluded from the final exam: 12.6, 13.5, last page of 14.3, 15.6, 16.3, 16.5, 16.6, last page of 17.2, 17.6, 18.6

ECE 254B w2002, Possible Research Topics

For list of possible topics, please see the previous offerings of the course. An updated list of research topics has not been produced because no student indicated an interest in pursuing the research option during this quarter.

Return to: Top of this page

ECE 254B: Spring Quarter 2001

Course:	ECE 254B – Advanced Computer Architecture: Parallel Processing, University of California, Santa Barbara, Spring 2001, Enrollment Code 11015
Instructor:	Behrooz Parhami, Room 5155 Engineering I, Phone 805-893-3211, parhami at ece dot ucsb dot edu
Meetings:	MW 8:30-9:50, Room 1173 HSSB (plus makeup lecture on F 4/20, location TBD)
Consultation:	Open office hours, held in Room 5155 Engineering I – M 11-12, W 1-2, R 3-4
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.
Prerequisite:	Basic computer architecture at the level of ECE 154 and, optionally, ECE 254A.
References:	*Required Text*–Parhami, B., *Intro. Parallel Processing: Algorithms and Architectures*, Plenum, 1999. *Journals* – *IEEE Trans. Computers*, *IEEE Trans. Parallel and Distributed Systems*, *J. Parallel & Distributed Computing*, *Parallel Computing*, *Parallel Processing Letters*. Also, *IEEE Computer* and *IEEE Concurrency* for broad introductory papers. *Conferences* – 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 since 1987/1989), and ACM Symp. Parallel Algorithms and Architectures (SPAA, since 1988).
Evaluation:	Students will be evaluated based on these three components with the given weights: 25% -- Homework: see the "deadlines" column in course calendar for schedule. 25% -- Closed-book midterm exam: see the course calendar for date and coverage. 50% -- Open-book final exam: see the course calendar for date, time, 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 parallel processing or do original research on a selected topic. A publishable report earns an "A" for the course, regardless of homework and midterm grades. See the "Research deadlines" column in the course calendar for schedule and due dates. Please follow the research paper guidelines to format your paper.

Calendar:

Course lectures, homework assignments, and exams have been scheduled as follows. This schedule will be strictly observed. Textbook chapters are specified for lectures and exams. It is a good idea to look over the covered chapter(s) *before* each lecture if possible.

Day & Date	Chap	Lecture Topic	Deadlines, Links, Notes
Mon. 4/2/01	1	Introduction to parallelism	
Wed. 4/4/01	2	A taste of parallel algorithms	
Mon. 4/9/01	3-4	Complexity and models	HW#1 posted (chapters 1-4)
Wed. 4/11/01	5	PRAM and basic algorithms	
Mon. 4/16/01	6	More shared-memory algorithms	HW#2 posted (5-6), HW #1 due
Wed. 4/18/01	7	Sorting and selection networks	Research topics defined
Fri. 4/20/01	8	Other circuit-level examples	Room 216, Bldg 406 (near the library)
Mon. 4/23/01	9	Sorting on a 2D mesh or torus	HW#3 posted (7-8), HW #2 due
Wed. 4/25/01		Instructor away at IPDPS: No lecture	
Mon. 4/30/01	10	Routing on a 2D mesh or torus	HW#4 posted (9-10)
Wed. 5/2/01	11	Numerical 2D mesh algorithms	HW #3 due, Prelim. references due
Mon. 5/7/01	12	Other mesh-related architectures	HW #4 due
Wed. 5/9/01	1-11	Midterm exam: closed book (8-10 AM)	
Mon. 5/14/01	13	Hypercubes and their algorithms	HW#5 posted (11-12)

Wed. 5/16/01	14	Sorting and routing on hypercubes	
Mon. 5/21/01	15	Other hypercubic architectures	HW#6 posted (13-14), HW #5 due
Wed. 5/23/01	16	A sampler of other networks	Paper title and references due
Mon. 5/28/01		Memorial Day: No lecture	HW#7 posted (15-17),
Wed. 5/30/01	17	Emulation and scheduling	HW #6 due, Prelim. paper abstract due
Mon. 6/4/01	18	Data storage, input, and output	
Wed. 6/6/01	19-20	Reliability and systems issues	HW #7 due
Fri. 6/8/01	1-18	Final exam: open book (8:30-11 AM)	Room 216, Bldg 406 (near the library)

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.

Course Grades

Item	**Out of**	**Max**	**Mean**	**Median**	**Min**	**SD**
HW #1	100	100	76	83	45	17
HW #2	100	90	60	63	10	24
HW #3	100	78	69	75	48	11
HW #4	100	83	53	59	15	22
HW #5	100	91	69	80	29	23
HW #6	100	88	72	69	59	9
HW #7	100	115	63	58	41	22
Midterm	100	77	64	64	31	14
Final/Paper	100	86	61	62	42	13
Course*	100	85	63	58	49	11
Course	A (4.0)	A+	3.4	3.0	B–	0.5

* Course = 25%(HW total / 7) + 25%(Midterm) + 50%(Final or Paper)

ECE 254B s2001, Homework #1: Fundamental concepts (chapters 1-4, due Mon. Apr. 16)

Do the following problems from the textbook, plus Problem 1.A given below.

1.4 (p. 21), 2.6 (p. 42), 3.3 (p. 61), 4.1 (p. 82)

Problem 1.A on parallel processing effectiveness: A task graph is built from back-to-back complete binary trees (two complete binary trees merged at their leaves). One root is the starting node and the other root is the final node, with dependency arrows all going in the direction from the former to the latter. There are 2^h nodes at the widest point of the task graph (leaves of the original trees) and n = 3 ´ 2^h – 2 nodes overall. Nodes correspond to unit-time tasks to be scheduled for processing on a parallel machine with p processors and arrows indicate data dependencies or required execution order. Ignore the overhead for scheduling and communication throughout.

a. What is the maximum attainable speedup, given an unlimited supply of processors? What is the minimum number of processors required to achieve this speedup?

b. With p = 2^q processors ($q < h$), find the speed-up and average processor utilization as functions of h and q.

c. What is the acceptable range for p if we were to achieve asymptotically linear speedup?

d. Define an equivalent Amdahl task graph as having one "input" and one "output" task with computation times fn/2 and p parallel tasks executable only after the input task and before the output task, each having computation time (1 – f)n/p, where f (0 £ f £ 1) is the inherently sequential fraction of the corresponding Amdahl job. Compute the fraction f to achieve the same speedup as in part a.

ECE 254B s2001, Homework #2: Abstract shared memory (chapters 5-6, due Mon. Apr. 23)

Do the following problems from the textbook, plus Problem 5.C given below.

5.10 (p. 107), 5.16 (p. 108), 6.1 (p. 125), 6.9b (p. 126)

Note: In Problem 5.16, please do not assume the CRCW PRAM submodel that writes the maximum or minimum of the offered values in memory!

Problem 5.C on parallel prefix computation on a linked list:

a. Outline the design of an efficient parallel prefix algorithm for the PRAM (no actual code needed), assuming that the n data items are stored in the form of a doubly linked list. The array D[1..n] contains the data items while the integer arrays F[1..n] and B[1..n] hold the forward and backward links; i.e., F[i] and B[i] point to the next and previous elements in sequence. The terminal list element in either direction points to itself. The computed prefix results should be stored in P[1..n] in the same linked format (i.e., with sequencing links given by F and B).

b. Which PRAM submodel is implied by your algorithm?

c. If the answer to part b is not EREW, then how can the algorithm be modified to run on this "weaker" submodel with little or no performance penalty?

ECE 254B s2001, Homework #3: Circuit model of parallelism (chapters 7-8, due Wed. May 2)

Do the following problems from the textbook, plus Problem 7.H given below.

7.12 (p. 146), 7.14 (p. 146), 8.1 (p. 165), 8.6 (p. 166)

Problem 7.H on sorting networks with adjacent comparators:

A comparator in a sorting network is "adjacent" if it compares values on lines i and i + 1 for some i in the range [0, n - 2].

a. Prove that any n-input sorting network must contain at least one instance of each of the n - 1 possible adjacent comparators.

b. Let $a(n)$ be the least possible number of comparators in an n-sorter composed solely of adjacent comparators. Establish upper and lower bounds for $a(n)$; that is, $a(n)$ = O(?) and $a(n)$ = W(?).

ECE 254B s2001, Homework #4: Data movement on 2D arrays (chapters 9-10, due Mon. May 7)

Do the following problems from the textbook, plus Problems 9.C and 10.D given below.

9.5 (p. 188), 9.11 (p. 189), 10.2 (p. 208)

Optional problem for extra credit: 9.16 (p. 189)

Problem 9.C on the analysis of shearsort: On a p-processor 2D mesh with one dimension equal to x, where x < p^(1/2), is shearsort faster when x is the number of rows or the number of columns?

Problem 10.D on reversing a sequence on a mesh: Consider a sequence of p values, x_0, x_1, . . . , x_(p–1), stored on a mesh, one value per processor.

a. Outline an algorithm for reversing the sequence if the storage order is row-major.

b. Analyze the complexity of the algorithm given in your answer to part a.

c. Repeat part a for snake-like row-major order.

ECE 254B s2001, Homework #5: Mesh algorithms and variants (chapters 11-12, due Mon. May 21)

Do the following problems from the textbook.

11.3 (pp. 231-232), 11.4* (p. 232), 11.7 (p. 232), 12.7 (p. 254), 12.14 (p. 255)

* Correct "back substitution" to "forward substitution".

ECE 254B s2001, Homework #6: The hypercube architecture (chapters 13-14, due Wed. May 30)

Do the following problems from the textbook.

13.3 (p. 275), 13.4 (pp. 275-276), 13.7 (p. 276), 14.9 (p. 296), 14.11 (p. 297)

ECE 254B s2001, Homework #7: Other networks and topics (chapters 15-17, due Wed. June 6)

Do the following problems from the textbook.

15.8 (p. 318), 15.12 (p. 319), 16.1 (p. 340), 16.12 (p. 342), 17.9 (p. 365)

Optional problem for extra credit: Problem 16.E on Clos networks

Consider a Clos network with rs inputs, rs outputs, and three columns (0-2) of switches. Columns 0 and 2 contain r switches, each of which is an s ´ s crossbar. Column 1 contains s switches that are r ´ r crossbars. Zero-origin top-to-bottom indexing is used to identify the switches in each column and their input/output lines. Switch terminals are identified by in(c, b, a) and out(c, b, a), with c being the column index, b the switch (block) index, and a the line index. Intercolumn connections are as follows: for all x and y (0 £ x £ r – 1, 0 £ y £ s – 1), out(0, x, y) is connected to in(1, y, x) and

a. Prove that the Clos network can realize any rs ´ rs permutation. [Hint: You may find the result on perfect matching in Section 17.1 useful.]

b. If the cost of an m ´ m crossbar switch is m^2 units, what are the optimal values of r and s for a given total number n = rs of inputs?

c. Try to prove a more general optimality result that holds for a class of cost functions rather than only for the one given in part b.

ECE 254B s2001, Reading list for midterm exam

Remember that the midterm exam starts at 8:00 AM. The following sections are excluded from the first 11 chapters of the textbook to be covered in the midterm exam: 2.6, 3.5, 4.5, 5.6, 6.5, 7.6, 9.6, 11.4

ECE 254B s2001, Reading list for final exam

The final exam will emphasize the material covered after the midterm (Chapters 12-18). All midterm exclusions apply for the final as well. Additionally, the following sections are excluded from the final exam: 12.6, 13.5, last page of 14.3, 16.5, 16.6, 17.6, 18.6

ECE 254B s2001, Possible Research Topics

The following are some suggested topics for research; they will be assigned on a first-come, first-served basis. In each case, one or two references are given to familiarize you with the subject area and the nature of the work expected. The paper may deal with a subset of the topic area, to be defined through further discussion after work has begun. If, after reviewing the cited reference(s), you are interested in a particular topic, please discuss your interest with the instructor for additional references and/or guidance. You are expected to find most of the needed references on your own. Searching the Internet and consulting the reference sources listed in the course outline are possible avenues.

Topic 1. Mesh-connected computers with hierarchically organized row/column buses [network]

[Pan01] Pan, Y., K. Li, and H. Shen, "An Improved Generalization of Mesh-Connected Computers with Multiple Buses," *IEEE Trans. Parallel and Distributed Systems*, Vol. 12, No. 3, pp. 293-305, Mar. 2001.

[Serr93] Serrano, M.J. and B. Parhami, "Optimal Architectures and Algorithms for Mesh-Connected Parallel Computers with Separable Row/Column Buses," *IEEE Trans. Parallel and Distributed Systems*, Vol. 4, No. 10, pp. 1073-1080, Oct. 1993.

Topic 2. Extended fault diameter of torus networks [mathematical]

[Parh00] Parhami, B., "Extended Fault Diameter of Mesh Networks," *Proc. Int'l Conf. Parallel and Distributed Processing Techniques and Applications*, pp. 1035-1039, June 2000.

Topic 3. Building-block computational algorithms for periodically regular chordal rings [algorithmic]

[Parh99] Parhami, B. and D.-M. Kwai, "Periodically Regular Chordal Rings," *IEEE Trans. Parallel and Distributed Systems*, Vol. 10, No. 6, pp. 658-672, June 1999.

[Nass93] Nassimi, D., "Parallel Algorithms for the Class of +/-(2^b) DESCEND and ASCEND Computations on a SIMD Hypercube," *IEEE Trans. Parallel and Distributed Systems*, Vol. 10, No. 12, pp. 1372-1381, Dec. 1993.

Topic 4. Reconfigurable parallel processor systems [architectural]

[Li91] Li, H. and Q.F. Stout (Eds.), *Reconfigurable Massively Parallel Computers*, Prentice-Hall, 1991.

Topic 5. Building special-purpose parallel systems on FPGAs or FPGA-like arrays [hardware]

[Thib99] Thibeault, C. and G. Begin, "A Scan-Based Configurable, Programmable, and Scalable Architecture for Sliding Window-Based Operations," *IEEE Trans. Computers*, Vol. 48, No. 6, pp. 615-627, June 1999.

[Vill98] Villasenor, J. and B. Hutchings, "The Flexibility of Configurable Computing", *IEEE Signal Processing Magazine*, Vol. 15, No. 5, pp. 67-84, Sep. 1998.

Topic 6. Coordination and scheduling in networks/clusters of workstations [software]

[Amir00] Amir, Y. et al, "An Opportunity Cost Approach for Job Assignment in a Scalable Computing Cluster," *IEEE Trans. Computers*, Vol. 11, No. 7, pp. 760-768, July 2000.

Return to: Top of this page

ECE 254B: Spring Quarter 2000 offering

Course: ECE 254B – Advanced Computer Architecture: Parallel Processing, University of California, Santa Barbara, Spring 2000, Enrollment Code 10512

Instructor: Behrooz Parhami, Room 5155 Engineering I, Phone 805-893-3211, parhami at ece dot ucsb dot edu

Meetings: MW 8:30-9:50 AM, Room 1431 Phelps Hall (no lecture on Mon. 5/29: Memorial Day)

Consultation: Open office hours, held in Room 5155 Engineering I – M 10:00-11:00, T 11:30-12:30, W 1:00-2:00

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.

Prerequisite: Basic computer architecture at the level of ECE 154 and, optionally, ECE 254A.

References: *Text*–Parhami, B., *Intro. Parallel Processing: Algorithms and Architectures*, Plenum, 1999.
Journals – *IEEE Trans. Computers*, *IEEE Trans. Parallel and Distributed Systems*, *J. Parallel & Distributed Computing*, *Parallel Computing*, *Parallel Processing Letters*. Also, *IEEE Computer* and *IEEE Concurrency* for broad introductory papers.
Conferences – Int'l Symp. Computer Architecture (ISCA, since 1973), Int'l Conf. Parallel Processing (ICPP, since 1972), Merged Int'l Parallel Processing Symp. and Symp. Parallel & Distributed Processing (IPPS/SPDP, since 1987/1989), and ACM Symp. Parallel Algorithms and Architectures (SPAA, since 1988).

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

25% -- Homework: see the "deadlines" column in course calendar for schedule.

25% -- Closed-book midterm exam: see the course calendar for date and coverage.

50% -- Open-book final exam: see the course calendar for date, time, 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 parallel processing or do original research on a selected topic. A publishable report earns an "A" for the course, regardless of homework and midterm grades. See the "Research deadlines" column in the course calendar for schedule and due dates.

Calendar:

Following, is a lecture-by-lecture course calendar, including homework assignment and due dates as well as research deadlines. This schedule will be strictly observed. It is a good idea to look over the specified course material *before* each lecture if possible.

Day/Date	Chap.	Lecture Topic	Events and Deadlines
Mon. 4/3	1	Introduction to Parallelism	
Wed. 4/5	2	A Taste of Parallel Algorithms	HW #1 given
Mon. 4/10	3-4	Complexity and Models	
Wed. 4/12	5	PRAM and Basic Algorithms	
Mon. 4/17	6	More Shared-Memory Algorithms	HW #2 given, HW #1 due

Wed. 4/19	7	Sorting and Selection Networks	
Mon. 4/24	8	Other Circuit-Level Examples	Select research topic
Wed. 4/26	9	Sorting on a 2D Mesh or Torus	HW #3 given, HW #2 due
Mon. 5/1	10	Routing on a 2D Mesh or Torus	
Wed. 5/3	11	Numerical 2D Mesh Algorithms	
Mon. 5/8	12	Other Mesh-Related Architectures	HW #3 due
Wed. 5/10	1-11	*Midterm Exam* (closed book, 8:20-9:50 AM)	
Mon. 5/15	13	Hypercubes and Their Algorithms	HW #4 given, Paper title & ref's due
Wed. 5/17	14	Sorting and Routing on Hypercubes	
Mon. 5/22	15-16	Other Parallel Architectures	
Wed. 5/24	17-18	Emulation, Scheduling, and I/O	HW #5 given, HW #4 due
Mon. 5/29		*No Lecture:* Memorial Day holiday	
Wed. 5/31	19-20	Reliability & Systems Issues	Abstract & ref's due
Mon. 6/5	21-22	MIMD Case Studies	HW #5 due
Wed. 6/7	23-24	SIMD Machines and Conclusion	
Mon. 6/12		Research report due by 4:00 PM	Final paper due
Thu. 6/15	1-24	*Final Exam* (open book, 8:30-11:00 AM)	

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 Engineering
University of California, Santa Barbara
Santa Barbara, CA 93106-9560 USA

http://www.ece.ucsb.edu/~parhami/

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

E-mail: parhami at ece.ucsb.edu