ECE 506: Architecture Of Parallel Computers

In Workflow
1. 14ECE GR Director of Curriculum (dgyu@eos.ncsu.edu; paulf@ncsu.edu)
2. 14ECE UG Director of Curriculum (mjescuti@ncsu.edu; jtuck@ncsu.edu)
3. 14ECE Grad Head (14ECE Grad Head@ncsu.edu)
4. 14ECE UnderGrad Head (ddstanci@ncsu.edu; gbyrd@ncsu.edu)
5. 14CSC GR Director of Curriculum (rouskas@ncsu.edu)
6. 14CSC UG Director of Curriculum (bahler@ncsu.edu)
7. 14CSC Grad Head (14CSC Grad Head@ncsu.edu)
8. 14CSC UnderGrad Head (vouk@ncsu.edu)
9. COE CC Coordinator UG (amy_matthews@ncsu.edu)
10. COE CC Chair UG (All)
11. COE Final Review UG (amy_matthews@ncsu.edu)
12. COE Dean UG (jerome_lavelle@ncsu.edu)
13. COE CC Coordinator GR (rfillin@ncsu.edu)
14. COE CC Chair GR (john_classen@ncsu.edu)
15. COE Final Review GR (rfillin@ncsu.edu)
16. COE Dean GR (reeves@csc.ncsu.edu)
17. OUCC Review (courses-curricula@ncsu.edu, aeherget@ncsu.edu, lamarcus@ncsu.edu)
18. jtuck (jtuck@ncsu.edu)
19. UCCC Coordinator (courses-curricula@ncsu.edu, aeherget@ncsu.edu, lamarcus@ncsu.edu)
20. UCCC Meeting (courses-curricula@ncsu.edu, aeherget@ncsu.edu, lamarcus@ncsu.edu)
21. UCCC Chair (andy_nowel@ncsu.edu)
22. OUCC Final Signature (barbara_kirby@ncsu.edu)
23. OUCC Final Review (lamarcus@ncsu.edu)
24. ABGS Coordinator (mnosbis@ncsu.edu)
25. ABGS Meeting (mnosbis@ncsu.edu)
26. ABGS Chair (mnosbis@ncsu.edu)
27. Grad Final Review (mnosbis@ncsu.edu)
28. PeopleSoft (none)

Approval Path
   Donna Yu (dgyu): Approved for 14ECE GR Director of Curriculum
2. Wed, 21 Oct 2015 19:00:32 GMT
   James Tuck (jtuck): Approved for 14ECE UG Director of Curriculum
   Melissa Nosbisch (mnosbis): Approved for 14ECE Grad Head
   Gregory Byrd (gbyrd): Approved for 14ECE UnderGrad Head
5. Thu, 19 Nov 2015 15:43:30 GMT
   Georgios Rouskas (rouskas): Approved for 14CSC GR Director of Curriculum
6. Thu, 11 Feb 2016 01:16:10 GMT
   Dennis Bahler (bahler): Approved for 14CSC UG Director of Curriculum
7. Fri, 29 Apr 2016 14:50:07 GMT
   Melissa Nosbisch (mnosbis): Approved for 14CSC Grad Head
8. Fri, 29 Apr 2016 15:32:01 GMT
   Mladen Vouk (vouk): Approved for 14CSC UnderGrad Head
9. Mon, 02 May 2016 16:32:38 GMT
   David Parish (dwparish): Approved for COE CC Coordinator UG
10. Mon, 02 May 2016 16:36:11 GMT
   David Parish (dwparish): Approved for COE CC Meeting UG
11. Mon, 02 May 2016 16:37:53 GMT
   David Parish (dwparish): Approved for COE CC Chair UG
12. Thu, 27 Oct 2016 20:34:21 GMT
   David Parish (dwparish): Approved for COE Final Review UG
   Jerome Lavelle (jerome_lavelle): Approved for COE Dean UG
   Robyn Fillinger (rfillin): Approved for COE CC Coordinator GR
   John Classen (john_classen): Approved for COE CC Chair GR
   Robyn Fillinger (rfillin): Approved for COE Final Review GR
17. Thu, 03 Nov 2016 21:12:01 GMT
   Douglas Reeves (reeves): Approved for COE Dean GR
18. Mon, 07 Nov 2016 14:24:40 GMT
   Alexandra Hergeth Huggins (aeherg): Approved for OUCC Review
   James Tuck (jtuck): Approved for jtuck
20. Wed, 16 Nov 2016 17:29:03 GMT
   Li Marcus (lamarcus): Approved for UCCC Coordinator
   Li Marcus (lamarcus): Rollback to jtuck for UCCC Meeting
22. Thu, 01 Dec 2016 15:03:47 GMT
   James Tuck (jtuck): Approved for jtuck
23. Thu, 01 Dec 2016 21:31:40 GMT
   Alexandra Hergeth Huggins (aeherg): Approved for UCCC Coordinator
24. Thu, 01 Dec 2016 21:32:45 GMT
   Alexandra Hergeth Huggins (aeherg): Approved for UCCC Meeting
25. Fri, 02 Dec 2016 14:07:31 GMT
   Andrew Nowell (andy_nowell): Approved for UCCC Chair
26. Tue, 06 Dec 2016 04:28:25 GMT
   Barbara Kirby (barbara_kirby): Approved for OUCC Final Signature
27. Tue, 06 Dec 2016 21:17:37 GMT
   Li Marcus (lamarcus): Approved for OUCC Final Review
28. Fri, 09 Dec 2016 20:40:00 GMT
   Melissa Nosbisch (mynosb): Approved for ABGS Coordinator
29. Thu, 19 Jan 2017 15:47:11 GMT
   Peter Harries (pjharrie): Approved for ABGS Meeting

Date Submitted: Thu, 02 Apr 2015 21:52:43 GMT

**Viewing: ECE 506/CSC 406/CSC 506/ECE 406 : Architecture Of Parallel Computers**

Changes proposed by: jtuck

**Change Type**

Major

**Course Prefix**

ECE (Electrical and Computer Engineering)

**Course Number**

506

**Course ID**
004515

Dual-Level Course
Yes

Dual-Level Course Number:
406

Cross-listed Course
Yes

Cross-listed with Subject Code(s)

Course Prefix:
CSC

Title
Architecture Of Parallel Computers

Abbreviated Title
Arch Parallel Comp

College
College of Engineering

Academic Org Code
Electrical & Computer Engineering (14ECE)

CIP Discipline Specialty Number
14.1001

CIP Discipline Specialty Title
Electrical and Electronics Engineering

Term Offering
Fall and Spring

Year Offering
Offered Every Year

Effective Date
Spring 2017

Previously taught as Special Topics?
No

Course Delivery
Face-to-Face (On Campus)

Grading Method
Graded/Audit

Credit Hours
3
Course Length
16 weeks

Contact Hours (Per Week)

Component Type | Contact Hours
--- | ---
Lecture | 3.0

Course Attribute(s)

Course Is Repeatable for Credit
No

Instructor Name
Yan Solihin

Instructor Title
Professor

Grad Faculty Status
Full

Anticipated On-Campus Enrollment
Open when course_delivery = campus OR course_delivery = blended OR course_delivery = flip

Enrollment Component | Per Semester | Per Section | Multiple Sections? | Comments
--- | --- | --- | --- | ---
Lecture | 1 | 50 | No | 

Course Prerequisites, Corequisites, and Restrictive Statement
Prerequisites For ECE 406: ECE 310.

Is the course required or an elective for a Curriculum?
No

Catalog Description
The need for parallel and massively parallel computers. Taxonomy of parallel computer architecture, and programming models for parallel architectures. Example parallel algorithms. Shared-memory vs. distributed-memory architectures. Correctness and performance issues. Cache coherence and memory consistency. Bus-based and scalable directory-based multiprocessors. Interconnection-network topologies and switch design. Brief overview of advanced topics such as multiprocessor prefetching and speculative parallel execution. Credit is not allowed for more than one course in this set: ECE 406, ECE 506, CSC 406.

Justification for each revision:
1. For several semesters, we have offered an undergraduate section of this course as an ECE 492, and this CAF makes that undergraduate section official. This will increase students awareness of the course and allow them to officially count it toward their major specialization electives.

2. We have modified the course learning outcomes to differentiate the expectations of ECE 406 and ECE 506 students. Course assignments and exams also reflect this difference. For example, 506 projects will include additional requirements that support their additional learning outcomes. Similarly, exams will include additional questions that evaluate these extra learning outcomes.

3. We've updated the course textbook, schedule, and syllabus to reflect how Dr. Solihin currently teaches the course.

4. ECE 506 is cross-listed as CSC 506.
5. The pre-requisites now reflect recent changes to our curriculum (old 406 is now 310).

6. A statement has been added to the Catalog Description indicating that students can only get credit for one of ECE 406, ECE 506 or CSC 506.

**Does this course have a fee?**

No

**Is this a GEP Course?**

No

**Consultation**

**Instructional Resources Statement**

Dr. Solihin will teach this as part of his existing teaching load.

**Course Objectives/Goals**

This course addresses topics in parallel computer architectures, i.e. architectures that support parallel processing. The course overviews fundamental issues related to parallel processing: programming paradigms, correctness, and performance; and case studies of modern parallel systems.

This course is intended for:

- First-year PhD students who are looking to build foundational knowledge in parallel computer architecture and parallel programming, as one major area of computer architecture. The topics covered in this course hopefully will enable them to read technical and research papers and understand more advanced concepts as well as understand the implications of various designs in parallel computer architecture.
- First-year Master’s degree students who are looking to build broad knowledge in the area of parallel computer architecture and parallel programming.
- Final-year undergraduate students who have taken basic computer organization course, and are looking to learn more advanced concepts in computer architecture, especially in the area of multiprocessor and multicore technology, as well as in parallel programming.
- Graduate students should take this course prior to taking the more advanced ECE 706 (Parallel Processing).

**Student Learning Outcomes**

By the end of the course, students in ECE 406/506 and CSC 506 will be knowledgeable in the following concepts in parallel computing/architecture:

- Explain why parallel architectures are needed, and where and how they are used.
- Compare and contrast different parallel programming models: shared memory (e.g. OpenMP) vs. message passing.
- Describe and apply parallel programming constructs in software: locks, barriers, point-to-point synchronization
- Describe and evaluate multiple parallelization techniques: loop level, task level, and algorithm level.
- Identify correctness issues in parallel programs related to variable scope, synchronization points, and computation ordering.
- Describe common performance bottlenecks related to loop transformations, thread scheduling, locality, page allocation, and false sharing.
- Describe the memory hierarchy and organization for a parallel computer, in particular, cache organization, write policy (write-through vs. write-back), replacement policy.
- Describe a bus-based multiprocessor architecture.
- Describe cache coherence protocols on bus-based machines and evaluate their latency and bandwidth trade-offs.
- Describe hardware support for synchronization primitives.
- Define memory consistency.
- Describe and evaluate directory-based cache coherency on distributed shared memory machines.
- Describe common interconnection networks in use today.

Students in ECE 506 and CSC 506 have these additional learning outcomes:
- Apply multiple parallelization techniques to solve a parallel computing problem: loop level, task level, and algorithm level.
- Implement multiple cache coherence and evaluate their relative performance and behavioral characteristics.
- Compare and contrast different memory consistency policies, such as sequential consistency, processor consistency, release consistency, etc.
- Evaluate trade-offs in the design and organization of memory hierarchies for parallel computers.

Student Evaluation Methods

<table>
<thead>
<tr>
<th>Evaluation Method</th>
<th>Weighting/Points for Each</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quizzes</td>
<td>10%</td>
<td>Per module quizzes.</td>
</tr>
<tr>
<td>Homework</td>
<td>10%</td>
<td>406 students will have fewer problems or problems of reduced scope.</td>
</tr>
<tr>
<td>Project</td>
<td>20%</td>
<td>Machine Problems; 406 students will have problems of reduced scope.</td>
</tr>
<tr>
<td>Midterm</td>
<td>30%</td>
<td>Midterm Exam</td>
</tr>
<tr>
<td>Final Exam</td>
<td>30%</td>
<td>Final Exam</td>
</tr>
</tbody>
</table>

Topical Outline/Course Schedule

<table>
<thead>
<tr>
<th>Topic</th>
<th>Time Devoted to Each Topic</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1.5 weeks</td>
<td>Read Ch. 1.</td>
</tr>
<tr>
<td>Parallel Programming Models</td>
<td>0.5 weeks</td>
<td>Read Ch. 2.</td>
</tr>
<tr>
<td>Parallel Programming Techniques</td>
<td>2 weeks</td>
<td>Read Ch. 3.</td>
</tr>
<tr>
<td>Correctness and Performance</td>
<td>1.5 weeks</td>
<td>Read Ch. 4.</td>
</tr>
<tr>
<td>Parallel Programming with Linked Data Structures</td>
<td>0.5 weeks</td>
<td>Read Ch. 5.</td>
</tr>
<tr>
<td>Introduction to Memory Hierarchy</td>
<td>1 week</td>
<td>Read Ch. 6.</td>
</tr>
<tr>
<td>Coherence, Consistency, Synchronization</td>
<td>0.5 weeks</td>
<td>Read Ch. 7.</td>
</tr>
<tr>
<td>Bus-based Cache Coherent Multiprocessors</td>
<td>2 weeks</td>
<td>Read Ch. 8.</td>
</tr>
<tr>
<td>Synchronization</td>
<td>1 week</td>
<td>Read Ch. 9.</td>
</tr>
<tr>
<td>Memory Consistency Model</td>
<td>1 week</td>
<td>Read Ch. 10</td>
</tr>
<tr>
<td>Interconnection Network</td>
<td>0.5 weeks</td>
<td>Read Ch. 12</td>
</tr>
<tr>
<td>Distributed Systems</td>
<td>1 week</td>
<td>Read Ch. 11</td>
</tr>
</tbody>
</table>

Syllabus


Additional Documentation

minosbis 12/9/2016: Ready for ABGS Reviewers.

ABGS Reviewer Comments:
- Does the prerequisite apply to CSC 406, as well? Syllabus says that for ECE 406 “To enroll, you must have taken ECE 310 (Design of Complex Digital Systems). In Computer Science, an equivalent course may be CSC 312 (Computer Organization and Logic) or CSC 456 (Computer Architecture and Multiprocessors). Should the alternative courses be listed in CIM as well so that students know this before the course starts? RESPONSE: CSC prerequisites should be listed in the prerequisite field.

- The syllabus should be more definitive in listing the additional requirements for graduate students rather than the statement “ECE/CSC 506 students may have more requirements/or they may have questions than ECE 406 students that pertain to the additional learning outcomes of ECE/CSC 506.” It should be clearly stated as to how the 500-level differs from the 400-level.

Course Reviewer Comments

bahler (Thu, 11 Feb 2016 01:15:14 GMT): Syllabus makes mention of "492 students". This document mentions "406 students".

lamarcus (Fri, 18 Nov 2016 21:31:28 GMT): Rollback: At instructor request and also for Approved Pending items: - Clarifying prerequisites: The items in this field all need to be codable in Peoplesoft, so ECE 310 is fine, but are there a series of courses that would translate to "sufficient knowledge of hardware design principles?" If not, that item can definitely go in the catalog description!- Adding a statement to the catalog description to the effect of "Credit is not allowed for more than one course in this set: ECE 406, ECE 506, CSC 406" etc to indicate that a student will not be able to take this course more than once for credit. This is already built into the system functionally, but the committee would like a written statement added in order to alert students.

Key: 1364