# Course Change Proposal

**Form A**

<table>
<thead>
<tr>
<th>Academic Group (College):</th>
<th>Academic Organization (Department):</th>
<th>Date:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engineering &amp; Computer Science</td>
<td>Computer Science</td>
<td>January 18, 2011</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type of Course Proposal:</th>
<th>Department Chair:</th>
<th>Submitted by:</th>
</tr>
</thead>
<tbody>
<tr>
<td>New _ Change _ Deletion _</td>
<td>Cui Zhang</td>
<td>Behnam Arad</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Does this course fulfill a requirement for single-subject or multiple subject credential students?</th>
<th>For Catalog Copy:</th>
<th>Semester Effective:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes _</td>
<td>No _</td>
<td>Yes _</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CCE (Extension):</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes _</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>This course replaces experimental course Subject Area (prefix) and Catalog Nbr (course number):</th>
</tr>
</thead>
<tbody>
<tr>
<td>If changing an existing course, should new version be considered a repeat of the original version? If so, the same Course ID will be maintained. If not, a new Course ID will be assigned. Note: In PeopleSoft terminology, the Course ID is the unique system identifier, not the Catalog Nbr.</td>
</tr>
<tr>
<td>Yes _</td>
</tr>
</tbody>
</table>

## Change from:

<table>
<thead>
<tr>
<th>Subject Area (prefix) &amp; Catalog Nbr (course no.):</th>
<th>Title:</th>
<th>Units:</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSC 205</td>
<td>Computer Systems Structure</td>
<td>3</td>
</tr>
</tbody>
</table>

## Change to:

<table>
<thead>
<tr>
<th>Subject Area (prefix) &amp; Catalog Nbr (course no.):</th>
<th>Title:</th>
<th>Units:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same</td>
<td>same</td>
<td></td>
</tr>
</tbody>
</table>

## JUSTIFICATION:

Catalog description is being updated to reflect current content.

## NEW COURSE DESCRIPTION: (Not to exceed 80 words, and language should conform to catalog copy. See http://www.csus.edu/umanual/acad.htm - Guidelines for Catalog Course Description)

Overview of computer systems organization and design. Concepts of instruction set architecture, interactions of hardware-software interface, principles of performance analysis, processor design, instruction level parallelism, survey of contemporary architectures, hierarchical memory design and analysis, interfacing I/O devices, parallel processing and multiprocessing, and introduction to EDA tools and methodologies for computer systems design and verification.

**Note:**

Prerequisite: Fully classified graduate status in Computer Science, Software Engineering or Computer Engineering. Enforced at Registration: Yes _ | No _

Corequisite: Enforced at Registration: Yes _ No _

Graded: Letter _ | Credit/No Credit _

Instructor Approval Required? Yes _ | No _

Course Classification (e.g., lecture, lab, seminar, discussion): Title for CMS (not more than 30 characters)

Seminar (05) | Computer Systems Structure

Cross Listed? If yes, do they meet together and fulfill the same requirement, and what is the other course.

Yes _ | No _

How Many Times Can This Course be Taken for Credit? _1__

Can the course be taken for Credit more than once during the same term? Yes _ | No _
FOR NEW COURSE PROPOSALS OR SUBSTANTIVE CHANGES ONLY:

**Description of the Expected Learning Outcomes:** Describe outcomes using the following format: “Students will be able to: 1), 2), etc.” See the example at http://www.csus.edu/acaf/example.htm

**Attach a list of the required/recommended course readings and activities** [Note: it is understood that these are updated and modified as needed by the instructor(s).] This attachment should be forwarded only to your Dean's office, not Academic Affairs.

**Assessment Strategies:** A description of the assessment strategies (e.g., portfolios, examinations, performances, pre-and post-tests, conferences with students, student papers) which will be used by the instructor to determine the extent to which students have achieved the learning outcomes noted above.

**For whom is this course being developed?**
Majors in the Dept X__ Majors of other Depts ___ Minors in the Dept ___ General Education ___ Other ___
Is this course required in a degree program (major, minor, graduate degree, certificate)? Yes ___ No ___
If yes, identify program(s):

Does the proposed change or addition cause a significant increase in the use of College or University resources (lab room, computer facilities, faculty, etc.)? Yes ___ No X__
If yes, attach a description of resources needed and verify that resources are available.

Indicate which department or programs will be affected by the proposed course (if any). MS in Comp Science and MS in Software Engr

The Department Chair's signature below indicates that affected programs have been sent a copy of this proposal form.

**Accessibility:** Following course approval, and prior to the start of the semester in which the new or revised course will be taught for the first time, an accessibility checklist [available at http://www.csus.edu/accessibility/checklist.html] shall be completed and submitted to the appropriate Dean’s office. An accessible syllabus shall also be made available online, preferably prior to the start of that semester’s open registration period.

**Approvals:** If proposed change, new course or deletion is approved, sign and date below. If not approved, forward without signing to the next reviewing authority, and attach an explanatory memorandum to the original copy.

**Signatures:**

<table>
<thead>
<tr>
<th>Department Chair:</th>
<th>Date:</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Signature]</td>
<td>1/8/2011</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>College Dean or Associate Dean:</th>
<th>Date:</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Signature]</td>
<td>2/15/2011</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CPSP (for school personnel courses only)</th>
<th>Date:</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Signature]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Associate Vice President and Dean for Academic Programs</th>
<th>Date:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Distribution: Academic Affairs (original), Department Chair and College Dean. Dean's office to send original after approval to Academic Affairs, at mail zip 6016. An electronic copy must also be sent.

5/20/2010
COURSE DESCRIPTION

Dept., Number  CSC 205  Course Title  Computer Systems Structure  
Semester hours  3  Course Coordinator  Behnam S. Arad  
URL (if any):  http://gaia.ecs.csus.edu/~arad/csc205

Proposed Catalog Description

Overview of computer systems organization and design. Concepts of instruction set architecture, interactions of hardware-software interface, principles of performance analysis, processor design, instruction level parallelism, survey of contemporary architectures, hierarchical memory design and analysis, interfacing I/O devices, parallel processing and multiprocessing, and introduction to EDA tools and methodologies for computer systems design and verification.

Prerequisite: Fully classified graduate status in Computer Science, Software Engineering or Computer Engineering.

Current Catalog Description

Overview of computer systems structure, covering hierarchical structure from software and hardware points of view. Concepts of relocation, linking, and loading; hardware-software interfaces from both application program and operating system points of view. Various CPU structures, including RISC and CISC machines, survey of tightly and loosely-coupled architecture, introduction to pipelined, distributed, and parallel systems, computer system communication principles including local and wide-area networks concepts, and various CAD tools and methodologies are introduced.

Prerequisite: Fully classified graduate status in Computer Science, Software Engineering or Computer Engineering.

Textbook


References


Course Goals

1. Provide students with a thorough understanding of design and performance issues of computer organization
2. Familiarize students with impact of parallelism on different aspects of computer organization
3. Expose students to modeling and verification of computer systems using a hardware description language
Prerequisites by Topic

*Thorough understanding of:*
- Combinational circuit design
- Sequential circuit design
- Machine language programming
- Instruction execution cycle
- Memory internal organization

*Basic Understanding of:*
- CPU organization
- Control unit design
- Memory hierarchy
- Bus cycle
- Computer arithmetic
- I/O interface
- Direct memory access
- Operating systems principles

*Exposure to:*
- Instruction pipelining
- Multiprocessor architecture
- Computer networking principles

Major Topics Covered in the Course

1. History and overview (1 hour)
2. Introduction to a hardware description language (4 hours)
3. Instruction set architecture (1 hour)
4. Computer arithmetic (2 hours)
5. Hardware/software interface with emphasis on instruction level parallelism (4 hours)
6. Processor design (10 hours)
7. A survey of contemporary architectures including graphics and computing GPUs (2 hours)
8. Interfacing and device communication (5 hours)
9. Hierarchical memory design including cache and virtual memory (9 hours)
10. Multicores, multiprocessors, and clusters (4 hours)
11. Performance evaluation and enhancements (3 hours)
Outcomes

Thorough understanding of:
- Processor design
- Instruction pipelining
- Memory hierarchy and optimization
- Fundamentals of I/O interfacing
- Impact of parallelism on different aspects of computer organization

Basic Understanding of:
- Instruction set architecture
- Branch prediction and speculative execution
- Super scalar and super pipelined instruction datapath
- Improving memory bandwidth (memory architecture)
- Improving data transfer rate (bus architecture)
- Performance evaluation
- Instruction-level parallelism

Exposure to:
- Data coherency in multiprocessor systems
- Introduction to parallel architectures
- Introduction to different bus protocols
- EDA tools for hardware design and verification

Laboratory projects

1. Introduction to EDA tools for modeling and validation (1 week)
2. Modeling of hardware components in an HDL (3-4 weeks)
3. Design, modeling, and validation of a computer system in an HDL. (5-6 weeks)

Oral and Written Communications

No significant component.

Social And Ethical Issues

No significant component.

Theoretical Content

- Performance analysis principles including Amdahl’s law
- CPU design principles
- Principle of locality
- Instruction level parallelism.
Problem Analysis

- CPU performance
- Tradeoffs of various CPU design alternatives
- Access time of hierarchical memory
- Efficiency of various I/O techniques
- Performance enhancement through parallel or distributed processing

Solution Design

- CPU design
- Cache memory design

Revised by Behnam Arad, 10/12/2010