270006 - EC - Computer Organization

Coordinating unit: 270 - FIB - Barcelona School of Informatics
Teaching unit: 701 - AC - Department of Computer Architecture
Academic year: 2018
Degree: BACHELOR’S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2010). (Teaching unit Compulsory)
ECTS credits: 7,5 Teaching languages: Catalan

Teaching staff

Coordinator: - Joan Manuel Parcerisa Bundo (jmanel@ac.upc.edu)
Others: - Agustin Fernández Jiménez (agustin@ac.upc.edu)
- Angel Toribio Gonzalez (angelt@ac.upc.edu)
- Antonio Jose Peña Monferrer (antoniop@ac.upc.edu)
- Calvin Bulla (calvin.bulla@upc.edu)
- David López Álvarez (david@ac.upc.edu)
- Jordi Tubella Murgadas (jordit@ac.upc.edu)
- Jose M. Cela Espin (josem.cela@ac.upc.edu)
- Jose Maria Arnau Montañes (jarnau@ac.upc.edu)
- Josep-Llorenç Cruz Diaz (cruz@ac.upc.edu)
- Lluc Álvarez Martí (lluca@ac.upc.edu)
- Miguel Moretó Planas (mmoreto@ac.upc.edu)
- Pedro Benidicte Illescas (pedro.benicde@upc.edu)
- Ruben Tous Liesa (rtous@ac.upc.edu)

Prior skills

Students are expected to understand the operation of a simple computer, its structure and internal functioning in blocks, as well as the basic elements of the high-level languages used to program simple applications.

They should be able to personally organise the study of this and related subjects and have a basic grounding in mathematics sufficient for the analysis, abstraction and synthesis required for the course.

Degree competences to which the subject contributes

Specific:
CT1.1A. To demonstrate knowledge and comprehension about the fundamentals of computer usage and programming, about operating systems, databases and, in general, about computer programs applicable to the engineering.
CT1.1B. To demonstrate knowledge and comprehension about the fundamentals of computer usage and programming. Knowledge about the structure, operation and interconnection of computer systems, and about the fundamentals of its programming.
CT3.6. To demonstrate knowledge about the ethical dimension of the company: in general, the social and corporative responsibility and, concretely, the civil and professional responsibilities of the informatics engineer.
CT5.2. To know, design and use efficiently the most adequate data types and data structures to solve a problem.
CT6.2. To demonstrate knowledge, comprehension and capacity to evaluate the structure and architecture of computers, and the basic components that compound them.
CT7.1. To demonstrate knowledge about metrics of quality and be able to use them.

General:
G2. SUSTAINABILITY AND SOCIAL COMPROMISE: to know and understand the complexity of the economic and social
phenomena typical of the welfare society. To be capable of analyse and evaluate the social and environmental impact.

**Teaching methodology**

Theory classes will combine lectures, in which the lecturer presents, explains and exemplifies the topics of study, with open discussion on issues raised and their advantages and disadvantages.

Problem-solving classes will be based on three different approaches: resolution by the lecturer commented on by students; resolution by students individually; and resolution by students in groups. Once students have completed problems, the lecturer will provide feedback on errors.

Laboratory classes will be similar to problem-solving classes, except that exercises will be worked on in pairs using semi-automatic correction tools. Laboratory exercises will be assessed on an ongoing basis to encourage diligence in students.

**Learning objectives of the subject**

1. Understand the hierarchical levels of a computer.
2. Understand the ISA of a standard RISC processor and the qualitative and quantitative performance implications of its design.
3. To know how to represent and perform operations with integers in various formats: two's complement, sign-magnitude and bias.
4. To know how to represent and perform operations with real numbers in IEEE 754 floating-point format.
5. Understand how data structured in vectors, matrices and tuples are stored and accessed.
6. To learn how to translate high-level programs to assembly language (and vice versa) using a standard ABI. The high-level elements to be translated are expressions, conditionals, loops and subroutines.
7. Design arithmetic units for multiplication and division operations with natural numbers.
8. Understand and manage the exception and interrupt concepts.
9. Understand the internal structure and operation of cache memory, especially aspects that affect system performance.
10. Understand the usefulness of virtual memory, its basic functioning and the support hardware required.
11. Understand compilation, linkage and loading processes in software development.
12. Meet task completion deadlines.
13. Understand the factors that affect microprocessor performance, consumption, sustainability and impact on the environment.

**Study load**

<table>
<thead>
<tr>
<th></th>
<th>Total learning time: 187h 30m</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory classes:</td>
<td>45h 24.00%</td>
</tr>
<tr>
<td>Practical classes:</td>
<td>15h  8.00%</td>
</tr>
<tr>
<td>Laboratory classes:</td>
<td>15h  8.00%</td>
</tr>
<tr>
<td>Guided activities:</td>
<td>7h 30m 4.00%</td>
</tr>
<tr>
<td>Self study:</td>
<td>105h 56.00%</td>
</tr>
</tbody>
</table>
# Content

## Item 1. Introduction

**Degree competences to which the content contributes:**

**Description:**
Hierarchical description of the computer at different levels of abstraction. Performance metrics. Energy consumption metrics. Amdahl's law.

## Item 2. Assembler and basic data types.

**Degree competences to which the content contributes:**

**Description:**
Introduction to MIPS. Operands. Representation of natural numbers, integers and characters. Pointers, vectors and strings. Instruction formats.

## Item 3. Program translation

**Degree competences to which the content contributes:**

**Description:**
Logic operations and shifts. If and while statements. Subroutines. Compilation, linkage and loading.

## Item 4. Arrays.

**Degree competences to which the content contributes:**

**Description:**
Array storage. Sequential access to vectors and matrices.

## Item 5. Integer and floating point arithmetic.

**Degree competences to which the content contributes:**

**Description:**
Integers: addition, subtraction, multiplication and division. Floating point: representation, addition, multiplication and rounding.

## Item 6. Cache Memory.

**Degree competences to which the content contributes:**

**Description:**
<table>
<thead>
<tr>
<th>Item 7. Virtual Memory.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Degree competences to which the content contributes:</strong></td>
</tr>
<tr>
<td><strong>Description:</strong></td>
</tr>
<tr>
<td>Motivation and terminology. Address translation. Handling misses. TLB.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Item 8. Exceptions / Interrupts.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Degree competences to which the content contributes:</strong></td>
</tr>
<tr>
<td><strong>Description:</strong></td>
</tr>
<tr>
<td>Basic MIPS concepts and hardware support. Detailed functioning of an exception and model generic handling routine. Specific cases: TLB miss. System calls. Interrupts.</td>
</tr>
</tbody>
</table>
## Planning of activities

<table>
<thead>
<tr>
<th>Topic assimilation</th>
<th>Hours</th>
<th>Description</th>
<th>Specific objectives</th>
</tr>
</thead>
</table>
| **Introduction**   |  **13h** | Theory classes: 3h 30m  
Practical classes: 0h 30m  
Laboratory classes: 2h  
Guided activities: 0h  
Self study: 7h | **Topic 1 activities** | **1, 13** |
| **Assemblers and basic data types** | **17h** | Theory classes: 4h  
Practical classes: 2h  
Laboratory classes: 2h  
Guided activities: 0h  
Self study: 9h | **Topic 2 activities** | **2, 3, 5, 6** |
| **Program translation** | **26h** | Theory classes: 7h  
Practical classes: 3h  
Laboratory classes: 3h  
Guided activities: 0h  
Self study: 13h | **Topic 3 activities** | **2, 6, 11** |
| **Arrays** | **12h** | Theory classes: 2h  
Practical classes: 1h  
Laboratory classes: 2h  
Guided activities: 0h  
Self study: 7h | | |
### Description:
Topic 4 activities

### Specific objectives:
2, 5, 6

| Topic assimilation: Integer and floating point arithmetic | Hours: 28h  
Theory classes: 8h  
Practical classes: 3h  
Laboratory classes: 3h  
Guided activities: 0h  
Self study: 14h |
|---|---|

| Description:  
Topic 5 activities | Specific objectives:  
4, 7 |
|---|---|

### Mid-semester exam (EP)  
Description:  
This exam will reflect the learning objectives for the activities designed to facilitate assimilation of topics 1 to 5.  
Specific objectives:  
1, 2, 3, 4, 5, 6, 7, 11, 13

| Hours: 0h  
Guided activities: 0h  
Self study: 0h |
|---|---|

### Topic assimilation: Cache memory  
Description:  
Topic 6 activities  
Specific objectives:  
9

| Hours: 28h  
Theory classes: 8h  
Practical classes: 3h  
Laboratory classes: 3h  
Guided activities: 0h  
Self study: 14h |
|---|---|
### Topic assimilation: Virtual memory

**Specific objectives:**
10

**Description:**
Topic 7 activities

**Hours:** 14h
- Theory classes: 4h
- Practical classes: 2h
- Laboratory classes: 0h
- Guided activities: 0h
- Self study: 8h

### Topic assimilation: Exceptions and interrupts

**Specific objectives:**
8

**Description:**
Topic 8 activities

**Hours:** 14h
- Theory classes: 4h
- Practical classes: 2h
- Laboratory classes: 0h
- Guided activities: 0h
- Self study: 8h

### Analysis of the environmental implications of computer manufacture, use and recycling

**Specific objectives:**
12, 13

**Description:**
Task designed to evaluate the transferable competency in sustainability

**Hours:** 7h 30m
- Theory classes: 0h
- Practical classes: 0h
- Laboratory classes: 0h
- Guided activities: 0h 30m
- Self study: 7h

### Consolidation of the learning objectives

**Specific objectives:**

**Hours:** 28h
- Theory classes: 0h
- Practical classes: 3h
- Laboratory classes: 0h
- Guided activities: 0h
- Self study: 25h
### Completion of assessment activities

**Description:**
Activities focused on an overall review of the subject

**Specific objectives:**
1, 2, 3, 4, 5, 6, 7, 8, 9, 10

**Hours:**

- Theory classes: 0h
- Practical classes: 0h
- Laboratory classes: 0h
- Guided activities: 7h
- Self study: 0h

### Laboratory exam (EL)

**Description:**
Activities related to assessment

**Specific objectives:**
12

**Hours:**

- Guided activities: 0h
- Self study: 0h

### Final Exam (EF)

**Description:**
This exam will reflect the learning objectives for the activities designed to facilitate assimilation of topics 1 to 8.

**Specific objectives:**
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

**Hours:**

- Guided activities: 0h
- Self study: 0h
Qualification system

Marks will be awarded based on two components: the theory/problem-solving mark (NT) and the laboratory mark (NL), accounting for 80% and 20% of the final mark, respectively.

NT is obtained from a mid-semester exam (EP) mark, accounting for 20% of the total mark, and a final exam (EF) mark, accounting for 60% of the total mark. The EP mark can be recuperated through the EF mark, since the corresponding weight (20%) is applied to the maximum of the two marks.

NL is obtained from a laboratory exam (EP) and a continuous assessment (AC) mark, accounting for 85% and 15% of the NL, respectively. The AC mark is based on laboratory session tasks and the corresponding preparatory work.

The formula for calculating the final mark for the course is:

Mark = \( \max(EP \times 0.20 + EF \times 0.60, EF \times 0.80) + (EL \times 0.85 + AC \times 0.15) \times 0.20 \)

Bibliography

Basic:


Others resources:

Hyperlink

http://docencia.ac.upc.edu/FIB/grau/EC