Course guide
270006 - EC - Computer Organization

Last modified: 30/01/2024

Unit in charge: Barcelona School of Informatics
Teaching unit: 701 - DAC - Department of Computer Architecture.

Degree: BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2010). (Compulsory subject).

Academic year: 2023 **ECTS Credits:** 7.5 **Languages:** Catalan

LECTURER

Coordinating lecturer: JORDI TUBELLA MURGADAS

Others:

Primer quadrimestre:
OCTAVIO CASTILLO REYES - 11, 12
JOSE M. CELA ESPIN - 41, 42
JOAN MANUEL PARCERISA BUNDO - 41, 42
ANGEL TORIBIO GONZALEZ - 11, 12

Segon quadrimestre:
DAVID ÁLVAREZ ROBERT - 23, 31
ADRIÀ ARMEJACH SANOSA - 21, 22, 23
ROGER BAIG VIÑAS - 51, 52, 53
OCTAVIO CASTILLO REYES - 32, 33
JOSE M. CELA ESPIN - 41, 42, 51, 52, 61, 62
ANTONIO CORTÉS ROSSELLÓ - 21, 22
JOSEP-LLORENÇ CRUZ DIAZ - 61, 62
AGUSTÍN FERNÁNDEZ JIMÉNEZ - 11, 12, 13
ANTONIO JOSE PEÑA MONFERRER - 43, 53
RUBÉN TOUS LIESA - 41, 42, 43
JORDI TUBELLA MURGADAS - 31, 32, 33

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.

Generical:
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

Type	Hours	Percentage
Hours small group	15,0	8.00
Hours large group	60,0	32.00
Self study	112,5	60.00

Total learning time: 187.5 h

CONTENTS

Item 1. Introduction

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.

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

Item 3. Program translation

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

Item 4. Arrays.

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

Item 5. Integer and floating point arithmetic.

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

Item 6. Cache Memory.

Description:
Concepts and terminology. Direct mapping. Block diagrams and performance metrics. Associativity. Replacement. Multilevel caches.

Item 7. Virtual Memory.

Description:
Motivation and terminology. Address translation. Handling misses. TLB.

Item 8. Exceptions / Interrupts.

Description:
Basic MIPS concepts and hardware support. Detailed functioning of an exception and model generic handling routine. Specific cases: TLB miss. System calls. Interrupts.

ACTIVITIES

Topic assimilation: Introduction

Description:
Topic 1 activities

Specific objectives:
1, 13

Related competencies :
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.

Full-or-part-time: 8h 30m
Theory classes: 2h
Practical classes: 0h 30m
Self study: 6h

Topic assimilation: Assemblers and basic data types

Description:
Topic 2 activities

Specific objectives:
2, 3, 5, 6

Full-or-part-time: 18h
Theory classes: 5h
Practical classes: 2h
Laboratory classes: 2h
Self study: 9h

Topic assimilation: Program translation

Description:
Topic 3 activities

Specific objectives:
2, 6, 11

Full-or-part-time: 26h
Theory classes: 7h
Practical classes: 3h
Laboratory classes: 3h
Self study: 13h

Topic assimilation: Arrays

Description:
Topic 4 activities

Specific objectives:
2, 5, 6

Full-or-part-time: 13h
Theory classes: 3h
Practical classes: 1h
Laboratory classes: 2h
Self study: 7h

Topic assimilation: Integer and floating point arithmetic

Description:
Topic 5 activities

Specific objectives:
4, 7

Full-or-part-time: 26h
Theory classes: 6h
Practical classes: 3h
Laboratory classes: 3h
Self study: 14h

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

Related competencies :
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.

Full-or-part-time: 1h 30m
Guided activities: 1h 30m

Topic assimilation: Cache memory

Description:
Topic 6 activities

Specific objectives:
9

Full-or-part-time: 27h
Theory classes: 7h
Practical classes: 3h
Laboratory classes: 3h
Self study: 14h

Topic assimilation: Virtual memory

Description:
Topic 7 activities

Specific objectives:
10

Full-or-part-time: 14h
Theory classes: 4h
Practical classes: 2h
Self study: 8h

Topic assimilation: Exceptions and interrupts

Description:
Topic 8 activities

Specific objectives:
8

Full-or-part-time: 12h
Theory classes: 2h
Practical classes: 2h
Self study: 8h

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

Description:
Task designed to evaluate the transferable competency in sustainability

Specific objectives:
12, 13

Related competencies :
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.

Full-or-part-time: 7h
Self study: 7h

Consolidation of the learning objectives

Description:
Activities focused on an overall review of the subject

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

Full-or-part-time: 29h 30m
Practical classes: 3h
Self study: 26h 30m

Laboratory exam (EL)

Description:
This exam will reflect the learning objectives for the activities designed to facilitate assimilation of topics 1 to 6.

Specific objectives:
2, 3, 4, 5, 6, 7, 9, 11, 12

Full-or-part-time: 2h
Guided activities: 2h

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

Related competencies :
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.

Full-or-part-time: 3h
Guided activities: 3h

GRADING 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*0.20 + EF*0.60, EF*0.80) + (EL*0.85 + AC*0.15)*0.20

BIBLIOGRAPHY

Basic:
- Patterson, D.A.; Hennessy, J.L4a. Estructura y diseño de computadores: la interfaz software/hardware. 4a. ed. Barcelona [etc.]: Reverté, 2011. ISBN 9788429126204.

RESOURCES

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