COURSE DATA

DATA SUBJECT

Code: 34658
Name: Computer organization
Cycle: Undergraduate Studies
ECTS Credits: 6
Academic year: 2026-27

STUDY (S)

Degree	Center	Acad. year	Period
1400 - Degree in Computer Engineering	Escola Tècnica Superior d'Enginyeria	2	Second quarter
1936 - Double Degree Program in Mathematics-Telematics Engineering	Facultat de Ciències Matemàtiques	3	Second quarter

SUBJECT-MATTER

Degree	Subject-matter	Character
1400 - Degree in Computer Engineering	Computer engineering	COMPULSORY
1936 - Double Degree Program in Mathematics-Telematics Engineering	Tercer curso	COMPULSORY

COORDINATION

ARNAU LLOMBART VICENTE

REAÑO GONZALEZ CARLOS

SUMMARY

The course "Computer Organization" is a compulsory subject in the second degree course in Computer Engineering. It is part of the subject "Computer Engineering" and is assigned a dedication of 6 ECTS taught in the second semester of the second course.

This course aims to introduce students the basics architecture of current processors and multiprocessor systems from the point of view of process design and system memory, as the programmer. Similarly, the student must be able to evaluate the performance of a monoprocessor or multiprocessor system and propose improvements in both, system architecture and in the code executed. For this simultaneous level description of register transfer level description with the memory and processor-level description of the operating system.

The course starts from the unicycle, multicycle and pipelined processor revised previous courses and introduces the segmentation from a generic point of view as a first approximation to take advantage of instruction level parallelism of the code (ILP). Subsequently, the MIPS processor is particularized.

Constraints arise together with segmentation, in which cases it raises the detention of the pipeline (risks). The types of risk are classified and strategies to combat them are suggested. Finally, the interruptions problem is afforded from the segmentation point of view.

Afterwards, we introduce the superscalar processors as a way to overcome the limitation of one instruction per cycle. We propose a generic architecture that shows the advantages of this approach to maximize the ILP. Similarly are the problems that come with such architectures (new types of risk), proposing strategies to resolve them: Tomasulo algorithm + buffer renaming, reordering buffer and branch predictors. Finally it is introduced the support hardware required to support execution of multiple threads in a processor, evaluating their strengths, weaknesses and strategies for peak performance.

VLIW processors are introduced as a paradigm of how obtain performance with a low-cost approach. This topic introduces software planning techniques basic block (unrolling loops and software segmentation) along with other extended block techniques planning. Finally predicated instructions are introduced as an alternative way to reduce the conditional jump instructions.

Analyze the performance of a parallel system is not as easy as on a monoprocessor system, where execution speed of programs is the main measure. We will have to introduce concepts such as productivity measurement. And we also see that not all systems can be evaluated in the same way, the final goal of a parallel application is not always finish early. The accuracy of the result and the ability to analyze complex problems can be other factors to take into account the performance of these systems.

The main part of a multiprocessor is designing the memory system. We must ensure that access to each of the memory modules are made in a coherent and consistent way. All processors should see the memory so that when a certain direction is read, always it is obtained the latest written value.
We will discuss the protocols used in current multiprocessor memory system design, both based on surveys such as those based directory.

Another key part of getting on a multiprocessor system a parallel application, consisting of multiple processes running in parallel; to work properly are the synchronization events. These allow processes to communicate with each other to exchange values and also can be synchronized to properly execute parallel algorithms that execute on multiprocessors.

The laboratory classes focus on maximizing the performance code execution in both monoprocessor and multiprocessor systems. This should have a thorough knowledge of computer system architecture to maximize its features.

PREVIOUS KNOWLEDGE

RELATIONSHIP TO OTHER SUBJECTS OF THE SAME DEGREE

There are no specified enrollment restrictions with other subjects of the curriculum.

OTHER REQUIREMENTS

For this second year subject it is recommended to have passed the subject of Computer Structure and to know concepts introduced in the field of Informatics. It is also advisable to have certain ability in programming, both in high-level languages and assembler.

COMPETENCES / LEARNING OUTCOMES

1400 - Degree in Computer Engineering

G10 - Knowledge to perform measurements, calculations, assessments, appraisals, surveys, studies, reports, scheduling and other similar work in the field of computer engineering, in accordance with both the knowledge and the specific skills acquired in the degree.

G1 - Ability to design, write, organise, plan, develop and sign projects in the field of computer engineering aimed at the design, development or exploitation of computer systems, services and applications.

G4 - Ability to define, evaluate and select hardware and software platforms for the development and implementation of computer systems, services and applications, in accordance with both the knowledge and the specific skills acquired in the degree.

G6 - Ability to design and develop computer systems and centralised or distributed computer architectures which integrate hardware, software and networks, in accordance with both the knowledge and the specific skills acquired in the degree.

G8 - Knowledge of basic subject areas and technologies that serve as a basis for learning and developing new methods and technologies, and of those which provide versatility to adapt to new situations.

IC2 - Ability to analyse, evaluate and select the most appropriate hardware and software platforms to support embedded and real-time applications.

R14 - Knowledge and application of the fundamental principles and basic techniques of parallel, concurrent, distributed and real-time programming.

R1 - Ability to design, develop, select and evaluate computer applications and systems while ensuring their reliability, safety and quality, according to ethical principles and current legislation and regulations.

R6 - Knowledge and application of basic algorithmic procedures of computer technology to design solutions to problems, by analysing the suitability and complexity of the algorithms proposed.

R7 - Knowledge, design and efficient use of the types and structures of data most suitable for solving a problem.

R9 - Ability to know, understand and evaluate the structure and architecture of computers, and also the basic components that comprise them.

DESCRIPTION OF CONTENTS

1. Pipelining

- Instruction level parallelism (ILP) and pipelining. Concept.

- Ideal and real performance.
- DLX pipeline.
- Data, structural and control risks.
- Pipelined processors interruptions.
- Programming pipelined processors.

Theory: 6. Problems: 2. Laboratory 5. Non-contact hours: 8 +5 (Theory + Lab).

2. Superscalar Processors

- Superscalar architecture.
- Buffer renaming.
- The Tomasulo algorithm.
- Maintaining consistency: ROB. Interruptions.
- Branch prediction.
- Multithreading.

Theory: 6. Problems: 3. Laboratory 5. Non-contact hours: 7 +5.

3. VLIW processors

- VLIW architectures.
- Software segmentation, loop unrolling.
- Traces planning.
- Predicated instructions.

Theory: 3. Problems: 2. Lab: 6. Non-contact hours: 6 +6.

4. Parallel systems performance

- Measuring and reporting performance.
- Performance models.

Theory: 5. Problems: 1. Laboratory 1.5. Non-contact hours: 4 +1.5.

5. Coherence and consistency in multiprocessors

- Coherence caches.
- Protocols based on survey and directory.

- Models of consistency and synchronization.

Theory: 6. Problems: 2. Lab 7.5. Non-contact hours: 9 +7.5.

WORKLOAD

PRESENCIAL ACTIVITIES

Activity	Hours
Theory	30,00
Laboratory	20,00
Classroom practices	10,00
Total hours	**60,00**

NON PRESENCIAL ACTIVITIES

Activity	Hours
Attendance at other activities	0,00
Individual or group project	12,00
Independent study and work	20,00
Preparation of lessons	34,00
Preparation for assessment activities	24,00
Resolution of case studies	0,00
Total hours	**90,00**

TEACHING METHODOLOGY

Theoretical activities.

Description: The lectures will develop the items by providing a global and inclusive vision, analyzing in detail the key issues and more complex, encouraging at all times, participation of students.

Workload for students on the total load of matter: 19%

Practical activities.

Description: Complementing theoretical activities in order to apply the basics and expand the knowledge and experience to be acquired in the course of the proposed work. They include the following types of classroom activities:

- classes of problems and issues in the classroom
- discussion sessions and problem-solving exercises and the students have previously worked
- labs
- oral presentations
- tutorials scheduled (individualized or group) Conducting individual evaluation questionnaires in the classroom with the presence of teachers.

Workload for students on the total charge of the matter: 21%

Personal work.

Description: Realization (outside the classroom) of monographs, literature search directed issues and problems as well as the preparation of classes and exams (study). This is done individually and tries to promote self-employment.

Workload for students on the total charge of the matter: 45%

Working in small groups.

Description: Realization, by small groups of students (2-4) of work, issues, problems outside the classroom. This work complements the work and encourages individual ability to integrate into working groups.

Workload for students on the total charge of the matter: 15%

It will be used the platform of e-learning (Aula virtual) of the University of Valencia in support of communication with students. Through it you will have access to course materials used in class as well as solved problems and exercises.

EVALUATION

Continuous assessment based on participation and the degree of involvement in the teaching-learning process, taking into account regular attendance to onsite activities and the resolution of activities and problems proposed in class. Part of the problem assessment and works is done in the tutorial hours individually, thus making it compulsory to attend tutorials to be evaluated. There will be 2 exams of 2 hours for to complete this assessment. This part will count 50% of the final grade of the first call.

Objective individual exam, consisting of 1 test of 2 hours, conducted in class time, consisting of both theoretical and practical issues as problems. This part will count 30% of the final grade. If the continuous assessment is passed with a mark greater than or equal to 5, the final exam will not be held, and the weight of the continuous assessment will be 80%.

Assessment of practical activities based on the achievement of objectives in the laboratory sessions. In each laboratory session it will be given a questions bulletin that have a dual purpose, firstly to guide the work to be done and secondly to assess how the objectives proposed have been achieved in each laboratory session. This part will count 20% of the final grade.

In the second round examination, students must take the exam of the second call. The exam will consist of a series of theoretical questions in which students must demonstrate knowledge of concepts and relationships, as another set of questions that will assess the practical part of the subject in which the student must prove his fitness and ability to relate the knowledge acquired in the course to the analysis of problems and case studies in the matter field. This exam will count for 80% of the mark. The practicals 10% and the continuous assessment 10%.

Copying or plagiarism of any activity that is part of the evaluation will result in the impossibility of passing the course, and the student will then be subject to the appropriate disciplinary procedures indicated in the ACTION PROTOCOL FOR FRAUDULENT PRACTICES AT THE UNIVERSITY OF VALENCIA (ACGUV 123/2020).

In any case, the evaluation of this subject will be done in compliance with the University Regulations in this regard, approved by the Governing Council on 30th May 2017 (ACGUV 108/2017).

REFERENCES

- Ortega J.; Anguita M.; Prieto A, Arquitectura de Computadores. Ed. Thomson, 2005.
- Hennessy J. L.; Patterson D. A., Computer Architecture a Quantitative Approach. 4ª Edition. Morgan Kaufmann Publishers, 2012. http://links.uv.es/IFDrw2x
- David E. Culler; Jaswinder Pal Singh., Parallel Computer Architecture: a Hardware/Software Approach. Ed. Morgan Kaufmann Publishers. 1999.
- Sima, D.; Fountain, T.; Kacsur, P, Advanced Computer Architecture. Addison-Wesley, 1998
- K. Hwang. Advanced Computer Architecture. Parallelism, Scalability, Programmability, McGraw Hill, 1993.