UNIVERSIDAD AUTÓNOMA DE MADRID

17820 - COMPUTER STRUCTURE

Information of the subject

Code - Course title: 17820 - COMPUTER STRUCTURE

Degree: 473 - Graduado/a en Ingeniería Informática
474 - Graduado/a en Ingeniería Informática y Matemáticas
722 - Graduado/a en Ingeniería Informática
734 - Graduado/a en Ingeniería Informática y Matemáticas (2019)

Faculty: 350 - Escuela Politécnica Superior

Academic year: 2020/21

1. Course details

1.1. Content area

Ingeniería de computadores, Informática

1.2. Course nature

Basic Training

1.3. Course level

Grado (EQF/MECU 6)

1.4. Year of study

1

1.5. Semester

Second semester

1.6. ECTS Credit allotment

6.0

1.7. Language of instruction

English

1.9. Recommendations

It is highly recommended to have completed the subject Computer Basics of the first semester.

1.10. Minimum attendance requirement

Two evaluation modalities are proposed: CONTINUOUS assessment and NON-CONTINUOUS assessment. These modalities can be applied **independently for the theoretical contents and for the practical contents**. By default, it is assumed that all students opt for a CONTINUOUS assessment modality.

The application of CONTINUOUS assessment for the theoretical contents is linked to the completion and obtaining of a minimum grade of the proposed activities during the development of the course.

The application of the CONTINUOUS evaluation for the practical contents is linked to the attendance and the realization and obtaining of a minimum grade of the activities proposed in the practical sessions in the laboratory.

CONTINUOUS and NON-CONTINUOUS ASSESSMENTS FOR THEORETICAL CONTENTS.

In both modalities, the theory class attendance is not mandatory but strongly recommended.

VERY IMPORTANT

Without the need to notify previously, tests can be done in any class. These tests have their weight only in the Continuous Evaluation. The absence during these sessions implies a grade of zero points in the activity.

The details about the evaluation regulations for each of the two modalities are included in the "Regular Assesment" section of this guide.

CONTINUOUS ASSESSMENT FOR PRACTICAL CONTENTS (LABORATORY).

In the CONTINUOUS evaluation modality, the student must attend all the practical classes and develop the activities that are proposed.

Always for duly justified reasons, the student may miss a maximum of 2 practice sessions (4 hours). In the case of reaching a greater number of faults, it will be excluded from this evaluation modality.

NON-CONTINUOUS ASSESSMENT FOR PRACTICAL CONTENTS (LABORATORY).

In this modality, the attendance to the practical lessons is not mandatory but strongly recommended.

The details about the evaluation regulations for each of the two modalities are included in the "Regular Assesment" section of this guide.

1.11. Subject coordinator/s

Alberto Sanchez Gonzalez

https://autoservicio.uam.es/paginas-blancas/

1.12. Competences and learning outcomes

1.12.1. Competences

C9 Capacidad de conocer, comprender y evaluar la estructura y arquitectura de los computadores, así como los componentes básicos que los conforman.
IC3 Capacidad de analizar y evaluar arquitecturas de computadores, incluyendo plataformas paralelas y distribuidas, así como desarrollar y optimizar software para las mismas.

1.12.2. Learning outcomes

In this course, it is learned to use a specific hardware description language. Using this language some basic arithmetic-logic circuits are also learned. The basic concepts related to the architecture of current processors are studied and analyzed based on the definition of a particular instruction set, the data path for the designed architecture is designed, and the control is learned. A simple low-level language (machine and assembler) is also studied for the designed system. Finally, the architecture of an elementary system of memory is studied.

1.12.3. Course objectives

The objectives of this subject are:

GENERAL OBJECTIVES	
G1	Apply the different design techniques for the realization of a digital system.
G2	Build a system based on its description in different levels of abstraction.

G3	Demonstrate the influence of the memory hierarchy and other evolutions in the performance of a computer system.
G4	Design complex digital systems, using specific hardware description languages (VHDL)
G5	Use design and programming tools (EDA) to design digital circuits.

SPECIFIC OBJECTIVES	
LESSON 1.- Digital Design and VHDL	
1.1.	Describe digital circuits using the specifications of the VHDL language.
1.2.	Given a digital circuit in VHDL, use the simulation and debugging tools to determine its correct functioning, and in case of failure, identify and correct the faults.
1.3.	Given a digital circuit in VHDL, determine its operation.
1.4.	Given a digital circuit in VHDL, identify syntax errors.
1.5.	Given a digital circuit in VHDL, identify functionality errors.
LESSON 2.- Arithmetic Logic Unit (ALU)	
2.1.	Describe the basic components that make up a computer system.
2.2.	Describe the different algorithms and digital circuits for the logical operations and, or, xor, etc.
2.3.	Describe the different algorithms and digital circuits for the arithmetic operations of adding, subtracting and multiplying.
2.4.	Describe in your own words what is an ALU and the different registers and flags that compose it.
2.5.	Given a certain set of operation codes for logical and arithmetic operations, design the corresponding ALU.
LESSON 3.- Processor I: Instruction set and machine code	
3.1.	Describe in your own words the concepts of machine language, operation code, addressing mode, size and instruction format, operating source and destination and immediate data.
3.2.	Assemble and disassemble machine code, aided by a table that contains the coding of the instructions.
3.3.	Indicate how the state of the computer (record content, data memory and input and output ports) is modified after the execution of an instruction or at the end of the execution of small programs written in assembly language (maximum 10 instructions), from an initial state of the computer.
3.4.	Write small programs in assembly language of the proposed processor (maximum 10 instructions) whose functionality is specified by a text or by a simple sentence of a high-level language. These programs can be defined as functions or macros to be incorporated into other programs of greater length.
3.5.	Write small programs (maximum 10 instructions) where the stack is used as a system for passing parameters in function calls or

	subroutines. Indicate the contents of the stack and the records associated with it after the execution of programs written in assembly.
LESSON 4.- Processor II: Design and control of the datapath. Single-cycle architecture.	
4.1.	From the schematic of a digital system, describe a certain operation through the language of record transfer (RTL).
4.2.	From the RTL description of a certain operation, design the data path that implements it.
4.3.	Once the data path for the proposed processor architecture is known, indicate the value of the active signals or buses of the Process Unit and of the Control Unit for each of the original instructions of the processor, in the case that each instruction runs in a single cycle.
4.4.	Design the control machine for the unicycle data path of the proposed processor or for a digital system of similar complexity.
4.5.	Starting from the data path unicycle of the proposed processor, complete the design and control of it, so that the processor correctly executes the original instructions plus a new machine language instruction (of complexity equivalent to the originals). It starts with the definition of the new instruction (via RTL language) and its codification.
LESSON 5.- Processor III: Design and control of the datapath. Multicycle architecture	
5.1.	Identify the processes that occur chronologically in the execution of an instruction
5.2.	Once the data path for the proposed processor architecture is known, indicate the value of the active signals or buses of the Process Unit and of the Control Unit for each of the original instructions of said processor, in the case that each instruction can run in a different number of cycles.
5.3.	Design the control machine for the multi-cycle data path of the proposed processor or for a digital system of similar complexity.
5.4.	Starting from the multi-cycle data path of the proposed processor, complete the design and the control, so that the processor executes correctly the original instructions plus a new machine language instruction (of complexity equivalent to the original ones). It starts with the definition of the new instruction (via RTL language) and its codification.
LESSON 6.- Memory maps. Real numbers operations.	
6.1.	Establish an address map based on the system requirements and identify the access addresses to each device from an address map.
6.2.	Describe the different ways of representing real numbers on a computer.
6.3.	Operations with real numbers in floating point.

1.13. Course contents

Summarized syllabus

1. Digital design and VHDL.
2. Arithmetic Logic Unit (ALU).

3. Processor I: Instruction set and machine code.
4. Processor II: Design and control of the datapath. Single-cycle architecture.
5. Processor III: Design and control of the datapath. Multicycle architecture.
6. Memory maps. Real numbers operations.

Detailed syllabus

1. Digital design and VHDL
- 1.1. Introduction
  - 1.1.1. Modules: entity and architecture
  - 1.1.2. Simulation and Synthesis
- 1.2. Combinational Logic
  - 1.2.1. Logic gates
  - 1.2.2. Conditional assignment
  - 1.2.3. Internal variables, numbers and buses
  - 1.2.4. Case and if Statements
- 1.3. Structural modeling
- 1.4. Sequential Logic
  - 1.4.1. Synchronous registers
  - 1.4.2. Latches, Flip-flops and implicit memory
- 1.5. Testbenches

2. Arithmetic Logic Unit (ALU)
- 2.1. Basic processor architecture
- 2.2. Logic and arithmetic operations
  - 2.2.1. Logic operators
  - 2.2.2. Adders and substractors
  - 2.2.3. Shifters and multipliers
- 2.3. ALU design
  - 2.3.1. State register. Carry (C), Overflow (V), Sign (N) and Zero (Z)

3. Processor I: Instruction set and machine code
- 3.1. Assembly Language
  - 3.1.1. Instructions
  - 3.1.2. Operands: Registers, Memory and constants
  - 3.1.3. Machine Language
- 3.2. Instructions set architecture. ISA MIPS
  - 3.2.1. R-type, I-type and J-type Instructions
  - 3.2.2. Memory, Arithmetic/Logical and Branching instructions
  - 3.2.3. Addressing Modes
  - 3.2.4. Writing, compiling, linking and executing a program
- 3.3. Programming
  - 3.3.1. Procedure calls. Stack memory system
  - 3.3.2. Program structures Conditional statements and Loops
  - 3.3.3. Data array

4. Processor II: Design and control of the datapath. Single-cycle architecture.
- 4.1. Single-cycle datapath
  - 4.1.1. Memory instructions: lw, sw
  - 4.1.2. R-type instructions
  - 4.1.3. Conditional brach instructions: beq
- 4.2. Combinational single-cycle control
- 4.3. Adding new instructions: Adding immediate. Inconditional branch (jump)
- 4.4. Timing parameters

5. Processor III: Design and control of the datapath. Multicycle architecture.
- 5.1. Multicycle datapath
  - 5.1.1. Memory instructions: lw, sw
  - 5.1.2. R-type instructions
  - 5.1.3. Conditional brach instructions: beq
- 5.2. Sequential multicycle control
- 5.3. Adding new instructions: Adding immediate. Inconditional branch (jump)

6. Memory maps. Real numbers operations.
- 6.1. Interface between processor and peripherals: address map
  - 6.1.1. Aligned and not aligned blocks.
  - 6.1.2. Memory maps.
- 6.2. Real number operations
  - 6.2.1. Fixed and floating point representation
  - 6.2.2. Addition, subtraction and multiplication with real numbers

1.14. Course bibliography

1. Digital Design and Computer Architecture. D.M. Harris y S.L. Harris. Morgan Kaufman. Second Edition 2013. ISBN: 9780123944245. Ref_UAM: INF/C5200/HAR.
2. Estructura y diseño de computadores: La interfaz software/hardware. D.A. Patterson y J.L. Hennessy. Ed. Reverte 2011. ISBN: 9788429126204. Ref_UAM: INF/C5220/PAT.
3. Computer Organization And Design: The Hardware/Software Interface. D.A. Patterson y J.L. Hennessy. Morgan Kaufmann. 4ª Ed. 2009. ISBN: 9780123744937. Ref_UAM: INF/C5220/PAT.
4. Problemas resueltos de estructura de computadoras. F. García, J. Carretero, J.D. García y D. Expósito. Ed. Paraninfo. ISBN: 978-84-283-3701-4. 2015.
5. Fundamentos de diseño lógico y de computadores. M.M.Mano y C.R.Kime. Prentice Hall. 2005. ISBN: 8420543993. Ref_UAM: INF/C5200/MAN.
6. The Student's Guide to VHDL. P. Ashenden. Morgan Kaufman Pub. 2008. ISBN: 9781558608658. Ref_UAM: INF/C7410D/ASH.
7. Diseño de Sistemas Digitales con VHDL. S.A. Pérez, E. Soto y S. Fernández. Thomson. 2002. ISBN: 8497320816. Ref_UAM: INF/C7410D/PER.
8. Diseño digital avanzado con VHDL: vol 1. F. Machado, S. Borromeo y N. Malpica. Serv. Publicaciones URJC. 2009. ISBN: 9788498494198. Ref_UAM: INF/C7410D/MAC.

Main bibliography and alternatives:

LESSON 1. Digital design and VHDL.
Main: Ref[1] C4.
Alternatives: Ref[6] complete, Ref[7] complete, Ref[8] complete.

LESSON 2. Arithmetic Logic Unit (ALU).
Principal: Ref[1] C5.
Secundarias: Ref[2] C3, Ref[3] C3, Ref[4] C2, Ref[5] C5.

LESSON 3. Processor I: Instruction set and machine code.
Main: Ref[1] C6.
Alternatives: Ref[2] C2, Ref[3] C2, Ref[4] C3, C4 y C9.

LESSON 4. Processor II: Design and control of the datapath. Single-cycle architecture
Main: Ref[1] C7.1 y C7.3.
Alternatives: Ref[2] C4, Ref[3] C4, Ref[4] C5.

LESSON 5. Processor III: Design and control of the datapath. Multicycle architecture
Main: Ref[1] C7.4.
Alternatives: Ref[4] C5.

LESSON 6. Memory maps. Real numbers operations.
Main: Ref[1] C8. Ref[1] C5.
Alternatives: Ref[2] C6, Ref[3] C6, Ref[5] C13, Ref[4] C7.

2. Teaching-and-learning methodologies and student workload

2.1. Contact hours

	#horas
Contact hours (minimum 33%)	78
Independent study time	72

2.2. List of training activities

Activity	# hours
Lectures	42
Seminars	
Practical sessions	
Clinical sessions	
Computer lab	26

Laboratory	
Work placement	
Supervised study	
Tutorials	
Assessment activities	10
Other	

3. Evaluation procedures and weight of components in the final grade

3.1. Regular assessment

CONTINUOUS ASSESSMENT

Students can choose the modality of CONTINUOUS EVALUATION (CE) for the theoretical part, the practical part, or both of them.

Each part, theory and practice, is independent and involves different regulations.

Continuous Assessment: Theory

For the CE in theory, although it is highly recommended, class attendance is not mandatory. The subject is evaluated with a set of in-person activities to be developed during the course. All the activities will be developed, whenever possible, in the common schedule given in the calendar or otherwise in the same class schedule. These activities include two partial tests. If the partial tests are passed, their concepts are exempted on the final exam.

The exemption of the first two partial tests, P1 and P2, implies that, in the case of passing any of them (ExaP1, ExaP2 ≥ 5.0), it is not necessary to test the contents associated with the passed partials in the final exam of the subject in the ordinary assessment.

In the case of not passing any of them (ExaP1 or ExaP2 < 5.0), it is necessary to test the concepts of the pertinent partial exam, always with the third partial ExaP3, in the final exam of the subject in the ordinary assessment.

In the case of not passing any of the two partial tests (ExaP1 and ExaP2 < 5.0), the student must sit the final exam of the subject, as if it were a student who had opted for the non-continuous assessment modality, as explained later.

After the final exam in the ordinary assessment, there will be a set of marks, one for each partial, either that obtained during the course or in the final exam and a fourth mark obtained during the other activities developed during the course.

In the case that a student with a passed partial exam fills the questions of the final exam related to the passed exam, the final exam mark grade will prevail.

The mark corresponding to the Theory (Mark_Theo) is calculated according to the expression:

Mark_Theo= 0.25*ExaP1 + 0.35*ExaP2 + 0.30*ExaP3 + 0.10*OtherActivities

In the case that (Mark_Theo < 5.0), the student must sit the final exam of the subject in its extraordinary assessment as if it were a student who had opted for the non-continuous assessment method, as explained below.

Continuous Assesment: Practice

To pass the practical part, the student must attend all laboratory sessions. Always for duly justified reasons, a student may miss a maximum of 2 sessions of practice (4 hours) and must present the corresponding written proof.

The practices consist in the development, by parts, of the processor studied in the theory, until reaching its

complete design. As it is a cumulative design process, the results of each practice are necessary to complete the final processor architecture. Therefore, each of the design proposals is accompanied by its corresponding test bench, which allows the student to know the correctness of the design made. For this self-correction, the teacher will help the students to improve their designs.

The evaluation of the laboratories is based on in-person exams where exercises similar to the ones done in the laboratory sessions should be done. During all the tasks, all the parts of the microprocessor will be designed and also some assembler programs will be coded. Besides, in the last exam, the students should connect all the microprocessor's parts done during the course. This exam may have an oral test to defend the student's proposal.

The laboratory tasks can be done by groups or individually as stated in each task.

The weights of every laboratory exam for calculating the continuous laboratory mark will be published in Moodle before the beginning of this course.

NON-CONTINUOUS ASSESSMENT

For students who opt for the NON-CONTINUOUS modality in the theory part, in the practices part or in both, their marks will be obtained in the following way:

a. The mark in the Theoretical part is calculated as follows:

- Mark of the final exam (100%).

The final exam will consist of a written test, whose content will cover all the objectives that students must reach in the full course. This test may include both theoretical issues and problems.

b. The mark in the Practical part is calculated as follows:

- Mark of the practice final exam (100%).

The final exam will consist of a practical test, which allows the teacher to evaluate all the concepts developed in the laboratory practices proposed in the subject.

For both modalities: CONTINUOUS and NON-CONTINUOUS:

- Both parts, theoretical and practical, are scored between 0 and 10.
- The final mark of the subject is obtained from the theory and practice marks following the equation:

Mark = 0.4*Mark_Lab + 0.6*Mark_Theo

- To pass the subject it is mandatory to obtain a mark greater than or equal to 5 points, both in theory and in practice. Otherwise, the final note will be:

Mark = (0.4*Min(5, Mark_Lab) + 0.6*Min(5, Mark_Theo))

VERY IMPORTANT: When any type of copy is detected in any of the evaluation activities, whether theoretical or practical, the information reflected in Chapter IV of the document "Normativa de Evaluación Académica de la EPS", approved by the "Junta de Centro" on November 4, 2013, will be applied.

3.1.1. List of evaluation activities

Evaluatory activity	%
Final exam	18%-60%
Continuous assessment	82%-40%

3.2. Resit

a. The grade in the Theoretical part is calculated as follows:

- Mark of the final exam (100%).

The final exam will consist of a written test, whose content will cover all the objectives that students must reach in the full course. This test may include both theoretical issues and problems.

b. The grade in the Practical part is calculated as follows:

- Mark of the practice final exam (100%).

The final exam will consist of a practical test, which allows the teacher to evaluate all the concepts developed in the laboratory practices proposed in the subject.

Final mark:

- Both parts, theoretical and practical, are scored between 0 and 10.
- The final mark of the subject is obtained from the theory and practice marks following the equation:

  Mark = 0.4*Mark_Lab + 0.6*Mark_Theo

- To pass the subject it is mandatory to obtain a mark greater than or equal to 5 points, both in theory and in practice. Otherwise, the final note will be:

  Mark = (0.4*Min(5, Mark_Lab) + 0.6*Min(5, Mark_Theo))

VERY IMPORTANT: When any type of copy is detected in any of the evaluation activities, whether theoretical or practical, the information reflected in Chapter IV of the document "Normativa de Evaluación Académica de la EPS", approved by the "Junta de Centro" on November 4, 2013, will be applied.

3.2.1. List of evaluation activities

Evaluatory activity	%
Final exam	40%-100%
Continuous assessment	60%-0%

4. Proposed workplan

This workplan is tentative and depends on the calendar and the evolution of the subject.

Week	In-person activities
1	◦ Subject introduction. ◦ **L1**. Digital design and VHDL. Lesson: 1.1 ◦ **Pr1.** VHDL Tutorial (I).
2	◦ **L1**. Digital design and VHDL. Lessons: 1.2, 1.3, 1.4 ◦ **Pr1.** VHDL Tutorial (II).
3	◦ **L1**. Digital design and VHDL. Lessons: 1.5 ◦ **L2.** Arithmetic Logic Unit Lessons: 2.1, 2.2 ◦ **Pr2.** Simplified Processor (I)
4	◦ **L2.** Arithmetic Logic Unit. Lessons: 2.3 ◦ **Pr2.** Simplified Processor (II)
5	◦ **L3.** Processor I: Instruction set and machine code. Lessons: 3.1, 3.2 ◦ **Pr2.** Simplified Processor (III)
6	◦ **L3.** Processor I: Instruction set and machine code. Lesson: 3.2 ◦ **Pr3.** MIPS Assembly (I)
7	◦ **L3.** Processor I: Instruction set and machine code. Lesson: 3.2 ◦ **Pr3.** MIPS Assembly (II)
8	◦ **L3.** Processor I: Instruction set and machine code. Lesson: 3.3 ◦ **L4.** Processor II: Design and control of the datapath. Lessons: 4.1, 4.2 ◦ **Pr4.** Final integration of the Processor (I)
9	◦ **L4.** Processor II: Design and control of the datapath. Single-cycle

Week	In-person activities
	architecture. Lesson: 4.2 ◦ **Pr4.** Final integration of the Processor (II)
10	◦ **L4.** Processor II: Design and control of the datapath. Single-cycle architecture. Lessons: 4.2, 4.3 ◦ **Pr4.** Final integration of the Processor (III)
11	◦ **L5.** Processor III: Design and control of the datapath. Multicycle architecture. Lessons: 5.1, 5.2 ◦ **Pr4.** Final integration of the Processor (IV)
12ª	◦ **L5.** Processor III: Design and control of the datapath. Multicycle architecture. Lesson: 5.3 ◦ **Pr4.** Final integration of the Processor (V)
13	◦ **L6.** Memory maps. Real numbers operations. Lesson: 6.1
14	◦ **L6.** Memory maps. Real numbers operations. Lesson: 6.2
May	• Regular final exam
June	• Resit final exam