CSD description (English)

Prior skills

Students must have acquired a solid theoretical understanding of electronic technology and the accompanying practical skills developed during the subject Electronics for Telecommunications (1A).
Students should understand basic aspects of computer programming, which is the objective of the subjects Introduction to Computers (1A) and Programming Project (1B).
Students would also benefit from practices of generic skills developed in Semesters 1A and 1B, in particular team work, self-directed learning and spoken and written communication.

Regulations for carrying out activities

Project assignments have to be handed in before the given due date.
Projects are assessed using rubrics indicating criteria and their respective weight.
Team members have to fairly and honestly participate in project development and reporting, and have to be responsible of anything handed in in their name. Instructors will encourage continuous discussion with cooperative groups on anything related with the projects and their assessment, because it is understood that the act itself of assessment contributes enormously to achieve the learning of the specific content and the cross-curricular skills associated to this subject.

Learning objectives

On completion of the subject, students will be able to:

- Work in cooperative teams to: specify, organise, develop and verify projects, track the study time and timetable and solve team conflicts.
- Obtain materials for self-directed study from libraries and internet to solve the course assignments and problems of a similar complexity (preference should be given to materials in English).
- Report and document their work using handmade sketches, graphics, pictures, tables, word processing software, project management applications and other software following the quality criteria stipulated in the subject guides.
- Analyse, simulate and design logic functions using diagrams, schematics, Boole Algebra, Espresso algorithm, logic simulators, VHDL, and electronic design automation tools (EDA).
- Explain the basic technological features of digital circuits: voltages, noise levels, tri-state, propagation delays, power consumption, etc.
- Synthesise combinational circuits and sequential systems (finite state machines, FSM) in programmable logic devices like CPLD and FPGA, based on hierarchical structures of components, using VHDL and the EDA tools associated to the design flow.
- Design, simulate and implement an advanced digital system of medium complexity based on a datapath (data registers and ALU) and a control unit (FSM).
- Classify commercial microcontrollers considering architecture and processing capability, explain their most usual applications in the context of small embedded systems, and describe the integrated development environment (IDE) comprising hardware and software associated to the design process proposed by a given vendor.
- Design, simulate and implement projects based on microcontrollers using C language, the FSM coding style, basic I/O techniques (polling and interrupts), and the IDE tools supplied by a given vendor (Microchip, Atmel, Texas Instruments, etc.)
- Give an oral presentation of one of the projects implemented in the course, preparing all the necessary materials and following the instructions.

Teaching methodology

The subject is based on cooperative learning (CL) and problem/project-based learning (PBL). Students work in base groups of three members for the whole semester with the goal of implementing the projects that have to be handed in before the due date following the quality criteria (rubrics) and the established methodology. Emphasis is given to what the student does, applying repetitively the same technique of problem-solving: specifications, planning, development, final verification of the project using simulation or laboratory training materials; and the detailed reviewing of the student's deliverables.

In classroom, instructors discuss basic theory, tutorial examples and the specifications and general plan of the project to be solved.
In the laboratory session students learn how to use the EDA tools for implementing and testing digital circuits.

The session on directed academic activities is used as an extra support for problem solving and to give guidelines for the self-directed work out of the class: project management, task sharing, tutorials on the use of software or laboratory instrumentation, reviewing of the documentation, and group processing, etc.

We must remark that the weekly autonomous work out of class is essential to complete in time the assigned projects and their documentation and with the required quality.

In order to promote the use of English, this is the language used throughout the course, in class, in all the learning materials, in office tutoring time or email (teaching content through English).

Students answer an anonymous questionnaire to evaluate the course development and lecturer performance, with the aim of detecting dysfunctions and good ideas for future improvements.

Content
Chapter 1: Combinational circuits
- Number systems, logic and basic arithmetic operations, information codes.

- Logic functions (Boolean algebra, logic gates, minterms, maxterms, sum of products (SoP), product of sums (PoS), truth table.

- Simplification of logic functions (Espresso algorithm).

- Electrical characteristics of digital circuits' technology: logic levels, noise margins, propagation delay, power consumption, three-state outputs.

- Basic combinational standard logic (multiplexers, decoders, etc.) and arithmetic blocks (adders, comparators, etc.)

- Design flow for combinational circuits in VHDL language using commercial EDA tools (Intel, Xilinx, Lattice Semiconductors, etc.): specifications, planning, synthesis in programmable logic devices sPLD/CPL/FPGA, and functional and gate-level simulation (Active-HDL, ModelSim, Xilinx ISim, etc.) using test benches.

- Characterisation and measurements of combinational circuits: propagation delay, speed of computation, power consumption, comparison of alternative designs, etc.

Related activities
- Problems and tutorials in Chapter 1.

Chapter 2: Sequential systems
- Definition and specification of a sequential system: symbol, state diagram, timing diagram, functional description.

- Asynchronous 1-bit memory cells (RS and D latches). Application to timer and clock circuits, and massive semiconductor memories.

- Synchronous 1-bit memory circuits (flip-flops RS, JK, D, and T).

- General structure of a synchronous finite-state machine (FSM): state register, outputs logic and next state logic.

- Design flow for sequential circuits in VHDL language using commercial EDA tools (Intel, Xilinx, Lattice Semiconductors, etc.): specifications, planning, synthesis in programmable logic devices sPLD/CPL/FPGA, and functional and gate-level simulation (Active-HDL, ModelSim, Xilinx ISim, etc.) using test benches.

- Standard sequential systems: counters, data and shift registers. Frequency divides.

- Concept and architecture of an advanced digital system of medium complexity based on a datapath (data registers and ALU) and a control unit (FSM).

- Specification, planning, development, simulation and prototyping of a simple dedicated processor (serial adder or multiplier, matrix keypad decoder, asynchronous serial communication transmitter or receiver, real-time clock, programmable timer, etc.)

- Characterisation and measurements of sequential circuits: propagation delay, maximum operational frequency, CLK frequency, power consumption, comparison of alternative designs, etc.

Related activities
- Problems and tutorials in Chapter 2.

Chapter 3: Microcontrollers
- Central process unit (CPU). Microprogrammed digital system (logic functions with memories) capable of performing multiple operations with the same hardware.

- Evolution of a dedicated processor towards the architecture of a microprocessor system (processing unit, data and program memory, and peripherals). The microcontroller.

- Architecture of a commercial 8-bit microcontroller: PIC16F/18F, ATmega from Microchip, MPS430 from Texas Instruments, etc. Comparison with other platforms (Arduino, Raspberry Pi, etc.)

- Microcontroller design flow, integrated development environment (IDE), source code, C compiler, virtual laboratory Proteus-VSM simulation, training boards, prototyping.

- Programming in FSM style for a deeper understanding of the concepts in previous chapters, standardisation of the source code for easy error detection -debugging-, reliability and self-assessment.

- Polling (reading) of level-sensitive signals. Signal edge detection using interrupts. Driving (writing) digital outputs.

- Peripherals: counters and timers. The concept of real-time and the use of a crystal quartz oscillator as system clock and time base.

- Other peripherals: A/D converter, PWM and D/A module, LCD display, EEPROM, USART, I2C and ASPI, etc.

Related activities
- Problems and tutorials in Chapter 3.
- Project oral presentation. Cooperative groups will deliver a 10 minutes oral presentation of one of the

projects designed in the course. They have to prepare materials like slides, graphics, whiteboards, posters, etc. The instructor will use rubrics and peer-assessment strategies. The use of English will be especially rewarded with extra grades.

Tutorials and projects

Reviewing and solving the tutorials and designing the projects become the most significant activities in CSD, carrying the learning of the specific content while practising cross-curricular skills. They deal with the design of digital systems and have a duration of one or two weeks. Problems are solved in cooperative base groups through all the class sessions with the guidance of the instructor and also out-of-class study time.

The CL and PBL methodologies implies that students have to attend classes and laboratories with regularity for achieving a good comprehension of the subject content.

The written documentation of the project includes the annotation of the study time required to complete it, the description and distribution of tasks among the group members, a proposal of self-assessment, and the signature statement to reflect a fair participation of each group member.

Let's remark that the sequence of tutorials with typology PBL, is organised to be solved in a chain, like when climbing a ladder, constructing the knowledge step by step. They cannot be solved randomly by students with no initial experience in digital systems. Furthermore, students are encouraged to help each other in order to attain maximum performance, implying that they have to study together taking into account the guidelines discussed in class. Thus, it is not recommended dropping a chapter or the subject temporarily to come back later.

Finally, let's say that the proposed problems “work”, meaning that all of them can be mounted in prototyping boards like kits, and students can experiment with them using simulators or in the laboratory. They are nearer the “real world” than academy, with the aim of motivating and showing the applicability of this subject in everyday life. For instance, students learn on 7-segment or LED matrix displays, data multiplexers, matrix keyboards blocked with a secret key, timers, real-time clocks, programmable systems to control motors or traffic lights, data transmission subsystems, simple calculators, and other circuits of similar complexity. The designs solve the projects using programmable logic devices and VHDL, or microcontrollers and C language. In a way that alternative technological solutions can be easily compared.

A list of example concepts included in the proposed designs is:

- P1. Analysis of basic circuits using logic gates.
- P2. Standard combinational circuits.
- P3. Arithmetic circuits.
- P4. Combinational circuits with multiple components in a hierarchical architecture.

- P5. Simple 1-bit memory cells: latches and flip-flops.
- P6. Finite state machines (FSM).
- P7. Standard sequential components: registers and counters.
- P8. Dedicated processor with datapath and control unit.

- P9. Basic I/O in a microcontroller system.
- P10. Interrupts and FSM style in C language.
- P11. Peripherals: LCD. Project with multiple source files. Software drivers.
- P12. Peripherals: Timers, A/D, etc.

Bibliography

- Roth, C.; Kurian, L. Digital system design using VHDL. 2a ed. Thomson International, 2008. ISBN 0495244708.
- Brown, Stephen D.; Vranesic, Zvonko G. Fundamentals of digital logic with VHDL design. 2nd ed. Boston: McGraw-Hill, 2005. ISBN 0071244824.
- Hwang, E.O. Digital logic and microprocessor design with VHDL. CL-Engineering, 2005. ISBN 0534465935.
- Barnett, R. H.; Cox, S.; O'Cull, L. Embedded C programming and the Atmel AVR. New York: Thomson Delmar Learning, 2006. ISBN 1418039594.
- Reese, Robert B. Microprocessors: from assembly language to C using the PIC18Fxx2. Massachusetts: Da Vinci Engineering Press, 2005. ISBN 1584503785.

Additional resources

Study materials such as lecture notes, tutorials and exam solutions from previous years, electronic circuit simulator, standard digital electronics laboratory equipment, free and licensed software for EDA, and demonstration boards for design with PLD/FPGA and microcontrollers.

This course has a web page at http://digsys.upc.es, from which to freely access to all the materials and the archive of previous CSD courses and other similar subjects on digital systems. This web is a complete replacement of the Atenea digital campus.

This course has a blog at http://taupres.upc.es/blogs/digsys, which is like an agenda for following the course, with voluntary participation of the students through the thread of comments of every post.

Assessment

Link

Francesc J. Robert, CSD coordinator, September 2017