CO1016 Computer Systems

Credits: *20* **Convenor:** *Dr. R. Crole* **Semester:** *1st*

Prerequisites:	*none*		
Assessment:	*Coursework: 15%*	*Three hour exam in January: 85%*	
Lectures:	*30 hours*	**Problem Classes:**	*10 hours*
Surgeries:	*10 hours*	**Class Tests:**	*2 hours*
		Private Study:	*148 hours*

Subject Knowledge

Aims This module teaches the basic principles and technical details of the structure and operation of a modern computer.

Learning Outcomes Students should be able to: describe the memory-I/O model and top-level hardware; solve problems in computer arithmetic; give an account of, and solve problems, in propositional logic and digital circuit theory and practice; give a summary of, and solve simple problems in, the MIPS assembly and machine language, including addressing methods; summarise the technical details of an elementary (single-core) processor.

Methods Class sessions together with course notes, recommended textbook, worksheets, printed solutions, and some additional hand-outs and web support.

Assessment Marked problem-based worksheet and class tests; traditional written problem-based examination.

Skills

Aims To teach students scientific writing and problem solving skills.

Learning Outcomes Students will be able to: write short, clear, note based, summaries of technical knowledge; solve abstract and concrete problems (both routine seen, and simple unseen), including numerical data.

Methods Class sessions together with worksheets.

Assessment Formative: Marked problem-based worksheets, class tests. Summative: traditional written problem-based examination.

Explanation of Prerequisites No specific knowledge is required, but a very rudimentary understanding of logic and discrete mathematics will be helpful.

Course Description While modern computers are complex devices, there are a small number of key components from which the majority of computers are composed. This module will provide a broad picture of a modern computer, covering key hardware and software components: Very roughly speaking, hardware refers to physical artefacts such as a keyboard or memory chip, and software to programs which are stored using magnetic or electrical systems (although we shall see that we need to be a little more precise). In particular, there is an emphasis on hardware, and we only look at very "simple/low-level" software. The module will teach details of computer arithmetic (arithmetic calculations take place when almost any program runs), processors (we look at a single core in this introductory module) and memory (circuits for storing data). The design of fundamental circuits is explained, together with the details of many of the basic hardware circuits which are built from the fundamental circuits. The course also explains in detail how computers perform simple arithmetic, covering the theory and also the actual circuits. The circuit details of a very small processor–single core–are explained, based upon all of the previous material, and the program instructions that the processor executes are explained in some detail. This complete description of a single core processor is the culmination of the module, and students will then be equipped to read further material about full-scale modern multi-core processors.

Detailed Syllabus

Examples throughout the course will be based on the MIPS Instruction Set Architecture. See the module web page for industry details.

The top level view of a modern computer: memory, processors, I/O, the fetch, decode, execute cycle. Memory layouts and the Endian systems. The memory hierachy and simple details of cache memory.

The binary number system. Binary arithmetic: basic definitions, algorithms for computing arithmetic operations. 2s-complement integers. Overflow and correctness conditions.

Basic digital electronics: Elementary logic and truthtables. Gates for implementing (Boolean models of) simple logical propositions, and the composition of gates to make more complex circuits. Multiplexors, decoders, and related circuits. Clocks. Implementation of atomic Arithmetic Logic Units (ALUs) via digital circuits. Construction of a 3-bit ALU (and 32-bit ALU, non-examinable). Simple memory circuits, including CD latch. Basics of register files.

The MIPS instruction set and simple MIPS programs. A subset of the MIPS language treated in detail at the assembly and machine levels. [Note: slt instruction is non-examinable.] Semantics, machine fields, branch calculations, and assembly/machine translations.

Construction of a simple (core) datapath via composition of atomic ALUs, register file(s), memory(ies) and other basic components. Description of MIPS control program. The interaction of the datapath and control to make a processor. [Computing performance.]

Reading List

[B] Hennessy and Patterson, *Computer Organization and Design;*
ISBN: 9780123747501 (4th) 9780124077263 (5th), Morgan Kaufmann, 2013, fifth edition.
Note: fourth edition is okay..

[C] Tanenbaum, *Structured Computer Organization;*
ISBN-10: 0132916525. ISBN-13: 9780132916523, Prentice Hall (Pearson), 2013 (sixth edition).

[C] Stallings, *Computer Organization and Architecture;*
ISBN-10: 0132916525. ISBN-13: 9780132916523, Prentice Hall (Pearson), 2013 (eighth edition), 0130351199.

[C] Burrell, *Fundamentals of Computer Architecture,* Palgrave Macmillan.

[C] Carpinelli, *Computer Systems: Organization and Architecture,* Pearson (Addison Wesley).

[C] Clements, *The Principles of Computer Hardware,* Oxford University Press.

[C] Englander, *The Architecture of Computer Hardware and Systems Software,* Wiley.

[C] Hayes, *Computer Architecture and Organization,* McGraw Hill.

[C] Hamacher, Vranesic and Zaky, *Computer Organization, 5th edition,* McGraw Hill.

Resources

Course notes, web page, study guide, worksheets, handouts, lecture rooms with two OHPs, past examination papers, past tests.

Module Evaluation

Course questionnaires, course review.