# IS1500 Computer Organization and Components 9.0 credits

## Datorteknik och komponenter

This is a translation of the Swedish, legally binding, course syllabus.

## Establishment

Course syllabus for IS1500 valid from Autumn 2010

## Grading scale

A, B, C, D, E, FX, F

## Education cycle

First cycle

## Main field of study

Technology

## Specific prerequisites

**Completed upper secondary education including documented proficiency in Swedish corresponding to Swedish B and English corresponding to English A. For students who received/will receive their final school grades after 31 December 2009, there is an additional entry requirement for mathematics as follows: documented proficiency in mathematics corresponding to Mathematics A. And the specific requirements of mathematics, physics and chemistry corresponding to Mathematics D, Physics B and Chemistry A**

## Language of instruction

The language of instruction is specified in the course offering information in the course catalogue.

## Intended learning outcomes

Computer Engineering is the subject of a computer's internal operation and design.

The course shows how C programs correspond to assembly-language instructions. These instructions are fetched and executed by hardware built from digital components.

Students learn low-level programming in C and assembly language, and use these tools to explore the internals of a pipelined microprocessor. Building upon this foundation, students also work with performance improvements such as cache memories, and parallel execution.

After finishing the course, the student will be able to

- Explain the parts of a computer and the workings of each part
- Identify and use symbols for digital logic gates and blocks
- Describe the difference between combinatory logic and sequential logic
- Explain the design of simple buses and memories
- Describe arithmetic systems for computers, and convert numbers to and from different arithmetic systems
- Explain the internals of the microprocessor, expecially for pipelined processors
- Explain and use the connection between low-level programming in C, in assembly language, and in machine language
- Write assembly-language programs to solve small computing problems
- Write programs in assembly language and/or the C language, to perform input and output operations using polling and/or handshaking
- Write programs in assembly language and/or the C language, using interrupts and traps
- Explain the internals of cache memories, and calculate the expected hit rate when executing a given computer program
- Explain how the operation of one or several processors can be shared, among several programs or among parts of one program
- Explain synchronization, semaphores, and/or other methods for controlled co-operation between different programs or parts of one program, when executing on one or several processors
- synkronisering, semaforer och/eller andra metoder för att flera olika program och programdelar ska kunna samverka på ett kontrollerat sätt vid körning på en eller flera processorer

## Course contents

Computer Engineering is the subject of a computer's internal operation and design.

The course shows how C programs correspond to assembly-language instructions. These instructions are fetched and executed by hardware built from digital components.

Students learn low-level programming in C and assembly language, and use these tools to explore the internals of a pipelined microprocessor. Building upon this foundation, students also work with performance improvements such as cache memories, and parallel execution.

Truth table, gates, Boolean equations.
Combinatory logic and logical circuits.

Sequential circuits, memories and buses.

The internal operation of a computer - what is a program, and how does the microprocessor execute the program

The C language for Java programmers

Subroutines/functions/methods - in C, in assembly language, and in the microprocessor

Computer arithmetic: representations of integers and floating point numbers - how the microprocessor performs calculations

Low-level programming: mixing C and assembly-language code

Global and local variables in C, and in assembly language

Parameters and return values: pointers, call by reference, and call by value

Using a bus structure for internal communication - connecting the microprocessor to memory and to input/output devices

Communication, interrupts, direct memory access (DMA), block-data transfer

Internal design of a pipelined microprocessor with caches

The laboratory exercises both sharpen and examine the students' understanding of digital components, assembly language programming, polled input and output, low-level programming in C, interrupts and exceptions, cache memories, and co-operating parallel programs.

## Course literature

David A Patterson and John L Hennessy: Computer Organization and Design – The Hardware/Software Interface, Fourth Edition. Morgan Kaufmann 2009. ISBN 978 0 12 374493 7.
Nios II Processor Reference Manual, Altera.
Lecture notes, exercises, lab instructions and other material.

## Examination

- ANN1 - Component Demostration, 1.5 credits, grading scale: P, F

- TEN1 - Examination, 3.0 credits, grading scale: A, B, C, D, E, FX, F
- LAB1 - Laboratory Works, 4.5 credits, grading scale: P, F

Based on recommendation from KTH's coordinator for disabilities, the examiner will decide how to adapt an examination for students with documented disability.

The examiner may apply another examination format when re-examining individual students.

If the course is discontinued, students may request to be examined during the following two academic years.

## Other requirements for final grade

- ANN1 – Component exercise, 1,5 hp. Grading: P, F
- LAB1 – Laboratory exercises, 4,5 hp. Grading: P, F
- TEN1 – Examination , 3 hp. Grading: A, B, C, D, E, FX, F

## Ethical approach

- All members of a group are responsible for the group's work.
- In any assessment, every student shall honestly disclose any help received and sources used.
- In an oral assessment, every student shall be able to present and answer questions about the entire assignment and solution.