COURSE DESCRIPTION CARD - SYLLABUS

Course name
Architecture of computer systems [S1Teleinf1>ASK]

Course

Field of study
Teleinformatics

Year/Semester
1/2

Area of study (specialization)
–

Profile of study
general academic

Level of study
first-cycle

Course offered in
polish

Form of study
full-time

Requirements
compulsory

Number of hours

Lecture
30

Laboratory classes
30

Other (e.g. online)
0

Tutorials
0

Projects/seminars
0

Number of credit points

5,00

Coordinators

prof. dr hab. inż. Ryszard Stasiński
ryszard.stasinski@put.poznan.pl

Lecturers

dr inż. Krzysztof Arnold
krzysztof.arnold@put.poznan.pl

dr inż. Sławomir Michalak
slawomir.michalak@put.poznan.pl

prof. dr hab. inż. Ryszard Stasiński
ryszard.stasinski@put.poznan.pl

Prerequisites

The ability to program in the C language at an elementary level, preliminary knowledge of digital technology.

Course objective

Knowledge about hardware-software interface of a typical computer (ISA - Instruction Set Architecture) necessary for understanding phenomena leading to reduction of computer computational efficiency when coding an algorithm.

Course-related learning outcomes

Knowledge:
Assembler programming of a 32-bit computer (ARM, or similar), including knowledge about avoiding pipeline hazards, and code optimization for instruction-level parallel processors.

Skills:
Knowledge about contemporary procesor hardware optimization techniques (pipelining, instruction-level parallelism, speculative computing), and memory hierarchy, including problems appearing in multiprocessor systems.

Social competences:
Knowledge about rules governing program construction for software compatibility.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Learning outcomes presented above are verified as follows:
Lecture: final exam, 10 questions for 1 point each, passing level - 5.1 point. Exercises: correctly done lab exercises, correctly written reports, and knowledge verification, 2 colloquia.

Programme content

Lecture:
Computer revolution, testing of computer performance, Amdahl's law, current tendencies in computing equipment
Instruction classes, and their description, instructions and computer components, instruction formats, relations between assembler and high-level language instructions, data structures implementation.
Fixed-point and floating-point implementations of basic arithmetical operations (addition/subtraction, multiplication, division), Error control in IEEE 754 standard, commutativity in floating-point operations.
Instruction phases and their link with computer subsystems, pipelining: idea, simple sequential and pipelined processor, pipeline optimization, pipeline hazards and how they are avoided, Exceptions and interrupts, and their influence on pipeline implementation, instruction-level parallelism: basic solutions, impact on pipeline.
Memory hierachy, why it is necessary, types of caches, cache misses, virtual memory: organization, cooperation with higher memory levels, memory protection - priviledged operating modes, coherence and homogeneity in multiprocessor systems.
Computer peripherals: behaviour and reliability, mass memory technologies, RAID.
The art. Of multiprocessor programming: sequential and parallel parts of an algorithm, communication, multithreading, vector processors, GPU.
Exercises - laboratory:
Fixed-point arithmetic
Other operations of fixed-point unit
Floating-point operations, part 1
Floating-point operations, part 2
Assembler control instructions
Tables in assembler
SIMD mode
Parameter transfer to a function, part 1
Parameter transfer to a function, part 2
Exceptions and interrupts handling
Avoiding pipeline hazards
Multithreading programming

Teaching methods

Lecture - lecture, exercises on laboratory sets

Bibliography

Basic:
D. Patterson, J. Hennessy: Computer Organization and design, ed. 4 (or newer), Elsevier 2009.

Additional:
J. Hennessy, D. Patterson: Computer Architecture: A Quantitative Approach, ed. 4 (or newer), 2011

Breakdown of average student's workload
Hours ECTS
Total workload 116 5.0
Classes requiring direct contact with the teacher 60 3.0
Student’s own work (preparation for tests, preparation for laboratory 56 2.0
classes, literature studies)

Breakdown of average student's workload

	Hours	ECTS
Total workload	116	5,00
Classes requiring direct contact with the teacher	60	3,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	56	2,00