Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Department of System Programming and Specialized Computer Systems

COMPUTER ARCHITECTURE. MICROARCHITECTURE

Working program of educational discipline (syllabus)

Requisites of educational discipline

Higher education level	**First (bachelor)**
Knowledge domain	12 Information technologies
Specialty	123 Computer Engineering
Educational program	System Programming and Specialized Computer Systems
Status of the discipline	Selective
Form of education	Full-time (day-time)
Year of preparation, semester	III year, spring semester
Teaching hours	4 credits ECTS, 120 hours (lectures – 36 hours, lab classes – 18 hours, self-student studying – 66 hours.)
Semester control / control activities	Modular test, Semester test
Schedule	http://rozklad.kpi.ua
Language of study	English
Information about supervisors of the course / professors	Lectures: assist. of SPSCS department, PhD Oleksii A. Molchanov, kpi.spscs.oml@gmail.com Lab classes: assist. of SPSCS department, PhD Oleksii A. Molchanov, kpi.spscs.oml@gmail.com
Discipline placement	Campus of Igor Sikorsky KPI: https://campus.kpi.ua

Program of educational discipline

1. Description of the discipline, its purpose, subject of study and learning outcomes

The discipline "Computer architecture. Microarchitecture" studies the architectures of modern computers with an emphasis on the peculiarities of the development and programming of microarchitectures and their application to solving typical problems of purely software systems. It also studies the multi-level organization of computers, features of individual levels, and connections between them. Knowledge of computer architecture allows us to:

- understand how the software described in some high-level language is transformed into the language of machine commands executed by the computer;
- learn about the peculiarities of the functioning of microarchitectures, their design, development, and programming;
- learn about the difference between different families of microarchitectures;
- find out how you can speed up the execution of tasks that require a large number of typical calculations performed programmatically with the help of specialized microarchitectures;
- etc.

The subject of the discipline: microarchitectures and their multi-level organization.

The purpose of the discipline: training highly qualified specialists who have the necessary knowledge about the architectural hierarchy of modern computers, which is needed for the analysis, design, and programming of microarchitectures, as well as their application for solving typical computing tasks.

After mastering the academic discipline, students must demonstrate the following learning outcomes.

Knowledge of:

- scientific provisions underlying the functioning of computer tools, systems, and networks;
- different levels of computer architecture, as well as connections between these levels;
- types and features of designing and programming microarchitectures;
- implementation peculiarities of microarchitectures, which were developed to speed up the execution of typical computing tasks.

Skills:

- solve problems of analysis and synthesis of means characteristic of the specialty;
- design and program microarchitectures;
- identify, classify, and describe the operation of various microarchitectures;
- to formalize the received work results in the form of presentations and scientific and technical reports.

2. Prerequisites and post requisites of the discipline (place in the structural and logical scheme of education according to the relevant educational program)

Prerequisites — disciplines, the study of which should precede the study of this discipline:

- Data structures and algorithms;
- Programming;
- System programming;
- Peripheral devices;
- Computer logic;
- Schematic engineering (first semester);
- Parallel programming.

Post requisites — disciplines, the study of which must be preceded by the study of this discipline:

- Computer systems;
- Computer support of telecommunications;
- System software;
- Design technologies of computer systems.

3. The content of the discipline

Chapter I. Computer architecture hierarchy. Connections between architectural levels.

Topic 1.1 General information about computer architecture.

Introduction to computer architecture. Multilevel organization of modern computers. Historical milestones in the development of computers. Types of computers. Basic definitions and principles of computer architecture organization.

Topic 1.2 Instruction set architecture.

Computer instruction set. CISC and RISC. Data formats and operations with them. Addressing modes.

Topic 1.3 The architecture of the level of applied languages.

The process of translation of programs of the level of applied languages into programs of the assembler level.

Topic 1.4 The architecture of the level of assembler.

Assembler. Assembly language. Assembly process.

Topic 1.5 Level of operating system architecture.

Virtual memory. Multi-tasking in processors.

Chapter II. Microarchitectures.

Topic 2.1 Architecture families.

x86 architecture. Stack architecture. RISC-V architecture. MIPS architecture.

Topic 2.2 Application of an FPGA for calculations.

FPGA. Stack architecture SM16. SM16 instruction set. SM16 assembler and simulator. Speeding up packet parsing with SM16.

4. Training materials and resources

Basic literature:

1) Andrew S. Tanenbaum, Todd Austin "Structured Computer Organization. 6th edition"

2) David A. Patterson, John L. Hennessy "Computer Organization and Design. 5th edition"

3) William Stallings "Computer Organization and Architecture, 10th Edition"

Additional literature:

1) Сергієнко А. М. "Архітектура комп’ютерів: Конспект лекцій"

2) Матвієнко М. П., Розен В. П., Закладний О. М. "Архітектура комп’ютерів"

3) Тарарака В. Д. "Архітектура комп’ютерних систем"

4) Intel 64 and IA-32 Architectures Software Developer's Manual

5) The Java® Virtual Machine Specification Java SE 11 Edition

Educational content

5. Methods of mastering the discipline (educational component)

The educational content of the discipline includes lectures and laboratory works (computer practicum).

Lectures:

Chapter I. Computer architecture hierarchy. Connections between architectural levels

Topic 1.1 General information about computer architecture

Lecture 1. Introduction to computer architecture. Multilevel organization of modern computers.

Lecture 2. Historical milestones in the development of computers. Types of computers.

Lecture 3. Basic definitions and principles of computer architecture organization.

Topic 1.2 Instruction set architecture.

Lecture 4. Computer instruction set. CISC and RISC.

Lecture 5. Data formats and operations with them.

Lecture 6. Addressing modes.

Topic 1.3 The architecture of the level of applied languages.

Lecture 7. The process of translation of programs of the level of applied languages into programs of the assembler level.

Topic 1.4 The architecture of the level of assembler.

Lecture 8. Assembler. Assembly language. Assembly process.

Laboratory works (computer practicum) №1.1 and №1.2

Test

Topic 1.5 Level of operating system architecture.

Lecture 9. Virtual memory

Laboratory work (computer practicum) №2

Lecture 10. Multi-tasking in processors.

Chapter II. Microarchitectures.

Topic 2.1 Architecture families.

Lecture 11. x86 architecture.

Lecture 12. Stack architecture.

Lecture 13. RISC-V architecture.

Lecture 14. MIPS architecture.

Topic 2.2 Application of an FPGA for calculations.

Lecture 15. FPGA.

Lecture 16. Stack architecture SM16. SM16 instruction set.

Laboratory work (computer practicum) №3

Lecture 17. SM16 assembler and simulator.

Lecture 18. Speeding up packet parsing with SM16.

Modular test.

Lab classes.

In the laboratory classes, students must complete four laboratory works and acquire the appropriate skills and knowledge of working with individual levels of the computer architectural hierarchy.

Laboratory works (computer practicum):

1.1. Translation of high-level languages into low-level languages. Part 1.

1.2. Translation of high-level languages into low-level languages. Part 2.

2. Instruction set architecture. Transformation of virtual addresses.

3. Programming of stack processor SM16.

Studying educational classes should take place sequentially, according to the schedule of classes.

6. Self-student studying (SSS)

Self-student studying includes:

- preparation for lectures by studying the previous lecture material as well as the literary sources on which it is based (a list of literature sources and numbers of sections is provided together with the lecture material);

- preparation for laboratory works by getting acquainted with the task and methodical instructions for their implementation, including the study of theoretical material necessary for answering control questions for laboratory works;
- performing tasks of laboratory work and creating a report for each of them; the report should highlight the results of the laboratory work (code listing, analysis of the obtained results, etc.).

Control of knowledge and skills is carried out on:

- lab classes by checking laboratory work reports, checking implemented programs and surveys;
- in lecture classes by conducting a survey and modular control work.

The deadline for performing laboratory work is determined individually for each laboratory work and is specified by tutor when laboratory task is provided to students.

Policy and control

7. Policy of academic discipline (educational component)

Requirements for the student:

- the student is obliged to attend lectures and laboratory classes and actively work on learning the material taught in them;
- at the lecture, the teacher uses his presentation material; practices the practical part on a virtual machine;
- laboratory works are defended in two stages: in the first stage students perform tasks, prepare an electronic report and send it to the teacher; the second stage is the defense of laboratory work in a laboratory session;
- the test survey is conducted during a lecture session using all available materials;
- modular control work is performed in a lecture session;
- penalty points are awarded for untimely completion of laboratory work. The deadline for the defense of a laboratory work without penalty points is determined for each laboratory work and provided for the student with the assignment. The total number of penalty points is no more than 10;
- incentive (additional) points are awarded for: creative and timely performance and defense of all laboratory work; execution of the creative works; the total number of incentive points cannot exceed 10% of the rating scale;
- the policy and principles of academic integrity are defined in Chapter 3 of the Code of Honor of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute". More details: https://kpi.ua/code;
- norms of ethical behavior of students and employees are defined in Chapter 2 of the Code of Honor of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute". More details: https://kpi.ua/code.

8. Types of control and rating system of assessing learning outcomes (RSA)

Types of control and RSA:

- 4 laboratory works (totally 60 points) with next RSA:
  - Laboratory work 1:
    - quality of the report design — 0-5 points;
    - quality of program implementation — 0-2 points;
    - answer to control questions during the defense of laboratory work — 0-3 points;
    - compliance with the deadlines for laboratory work — 0-2 points;
    - maximum grade for laboratory work — 12 points;

    - when receiving 0 points for at least one of the first three components of the score or when the sum of points is less than 6 points, the work is considered not defended, since such a rating indicates an unsatisfactory level of assimilation of knowledge and acquisition of skills and abilities by the student during laboratory work;
  - Laboratory work 2:
    - quality of the report design — 0-5 points;
    - quality of program implementation — 0-2 points;
    - answer to control questions during the defense of laboratory work — 0-3 points;
    - compliance with the deadlines for laboratory work — 0-2 points;
    - maximum grade for laboratory work — 12 points;
    - when receiving 0 points for at least one of the first three components of the score or when the sum of points is less than 6 points, the work is considered not defended, since such a rating indicates an unsatisfactory level of assimilation of knowledge and acquisition of skills and abilities by the student during laboratory work;
  - Laboratory work 3:
    - quality of the report design — 0-2 points;
    - quality of program implementation — 0-7 points;
    - quality of program testing and tests implementation — 0-4 points;
    - answer to control questions during the defense of laboratory work — 0-4 points;
    - compliance with the deadlines for laboratory work — 0-3 points;
    - maximum grade for laboratory work — 20 points;
    - when receiving 0 points for at least one of the first four components of the score or when the sum of points is less than 10 points, the work is considered not defended, since such a rating indicates an unsatisfactory level of assimilation of knowledge and acquisition of skills and abilities by the student during laboratory work;
  - Laboratory work 4:
    - quality of the report design — 0-2 points;
    - quality of program implementation — 0-5 points;
    - quality of program testing and tests implementation — 0-2 points;
    - answer to control questions during the defense of laboratory work — 0-4 points;
    - compliance with the deadlines for laboratory work — 0-3 points;
    - maximum grade for laboratory work — 16 points;
    - when receiving 0 points for at least one of the first four components of the score or when the sum of points is less than 8 points, the work is considered not defended, since such a rating indicates an unsatisfactory level of assimilation of knowledge and acquisition of skills and abilities by the student during laboratory work;
- 1 penalty point is charged for each week after the deadline for laboratory work; the maximum number of penalty points is determined individually for each laboratory work (see above);
- test survey — 15 points. The survey consists of test questions. 1-2 points are awarded for each correct answer to a test question;
- modular test — 25 points. The modular test consists of test questions. 1-3 points are awarded for each correct answer to a test question;
- up to 5 incentive (additional) points are awarded for a creative approach, timely completion (before the deadline for laboratory work, determined by the teacher when issuing the assignment) and defense (up to two weeks after the deadline for laboratory work) of all laboratory work (i. e. not for each laboratory work separately but in total for all laboratory work);
- students may be awarded up to 5 incentive (additional) points for performing creative works (participating in competitive works, in the R&D, writing scientific articles in the Information Technology field, etc.) or for writing abstract (to increase the overall assessment after performing all types of control of learning outcomes);
- calendar control (attestation) is carried out twice a semester as a monitoring of the current state of meeting the syllabus requirements. *The condition for the first attestation* is to obtain at least

15 points (at the time of attestation). *The condition of the second attestation* is obtaining at least 35 points (at the time of attestation);

- semester control is test. The condition for admission to the semestral test is the enrollment (defense) of all laboratory works and a semester rating greater than or equal to 40 points. With a semester rating of 40-59 points, the student must write a final test. If the student's rating is 60 points or more, the student receives grade "automatically" in accordance with table 1;
- the semestral (assessment) test is valued at 100 points and is written by those students who, at the time of the assessment, have fulfilled the conditions for admission to the assessment and have a rating of 40-59 points. Also, those students who, at the time of the assessment, met the conditions for admission to the assessment and have a rating of 60 or more points, can write the assessment test, if such students want to improve their rating. Before writing the semestral test, the student's semester rating is set to zero. The assessment test consists of test tasks, the total score of which is 60 points, and two theoretical questions of 20 points each. 1-4 points are awarded for each correct answer to a test question. The list of topics of theory questions for the assessment test provided in Appendix 1. Each theoretical question is valued at 20 points according to the following criteria:
    - "excellent" — complete answer, at least 90% of the required information — 18-20 points;
    - "good" — sufficiently complete answer, at least 75% of the required information — 13-17 points;
    - "satisfactory" — incomplete answer, at least 60% of the required information and some errors — 8-12 points;
    - "unsatisfactory" — the answer does not meet the conditions of the "satisfactory" level — 0 points.

Table of correspondence of rating points to final marks according to the university scale (Table 1):

Points	*Final mark*
100-95	Excellent
94-85	Very good
84-75	Good
74-65	Satisfactory
64-60	Enough
Less than 60	Unsatisfactory
Prerequisite for admission to the exam is not completed	Not admitted

9. Additional information about the discipline (educational component)

- a list of some topics submitted for semester control is given in Appendix 1.

Working program of educational discipline (syllabus):

Developed by assistant of the Department of System Programming and Specialized Computer Systems, PhD Oleksii A. Molchanov

Approved by the Department of System Programming and Specialized Computer Systems (protocol № 6 of 28.01.25)

Agreed by Methodical Council of the Faculty of Applied Mathematics (protocol № 8 of 03.02.25)

Appendix 1. Some topics submitted for semester control.

1. Multilevel organization of computers. Definition of architecture, model.

2. The concept of virtual machines. Languages, translation and interpretation.

3. Architectural levels. Definitions and what they refer to.

4. Software and hardware.

5. The history of the development of computers. Generation of computers. General features, examples of systems (without implementation details). Factors influencing the development of computers

6. Moor's law.

7. Dennard's law. Dark silicon.

8. Computer types (with examples).

9. Basic computer architecture definitions.

10. Basic principles of computer construction.

11. Von Neumann's architecture. Harvard Architecture.

12. Computer instruction sets and their selection. Computer instruction structure.

13. CISC.

14. RISC. Overview of principles of building RISC architectures.

15. Comparison of CISC and RISC.

16. Data formats and operations with them. Storing numbers in memory.

17. Representation of integers in computer memory.

18. Representation of floating-point numbers in computer memory.

19. Representation of signed numbers. Direct, inverted, complementary codes, binary shift.

20. Addressing methods (direct, basic, index, indirect methods, etc.)

21. Translation of application level programs into assembly language level programs.

22. Assembler. Assembly language. Features, purpose, command formats, directives.

23. Macros. Macro definition, macro call, macro extension. Comparison of macros and procedures.

24. The process of assembly, layout and loading.

25. Virtual memory. Overlays.

26. Types of virtual memory organization.

27. Page memory organization. Definition of page, page frame. Size of pages.

28. Segment memory organization. Definition of segment, segmentation.

29. Segment-page memory organization.

30. Fragmentation, defragmentation. The principle of locality and the working set. Page replacement policies.

31. Multitasking. The context of the task. Switching task contexts.

32. FPGA.

33. Stack architecture.

34. x86 architecture.

35. SM16 stack microprocessor.