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

Department of Computer Engineering

FPGA programming technologies
Working program of the academic discipline (Syllabus)

Details of the academic discipline

Level of higher education	***First (bachelor's degree)***
Branch of knowledge	*12 Information technologies*
Specialty	*123 Computer engineering*
Educational program	*Computer Engineering*
Discipline status	*Selective*
Form of education	*intramural(daytime)*
Year of education, semester	*3rd year, spring*
Scope of the discipline	*4 credits, 120 hours,* *Lectures 18 (36 hours), Laboratory 9 (18 hours)*
Semester control/ control measures	*Test*
Lessons schedule	*According to the schedule for the spring semester of the current academic year at http://rozklad.kpi.ua*
Language of teaching	*Ukrainian*
Information about head of the course / teachers	Lecturers: professor of the CE department, Ph.D., docent, Klymenko Iryna Anatoliyivna, ikliryna@gmail.com; senior teacher of the CE department Yuriy Mykolayovych Vynogradov, vinograd514kpi@gmail.com; Laboratory: Vynogradov Yuriy Mykolayovych, vinograd514kpi@gmail.com, Haiday Anatoliy Ruslanovych, tolya.hei@gmail.com
Placement of the course	The training course is hosted on the "Sikorsky" distance learning platform in the Google Workspace for Education environment: https://classroom.google.com/c/NDg4NTYwMDIzMDA0?cjc=fb6xoef

Program of the Academic Discipline

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

Programmable logic (FPGA) has recently become the main technology used to create electronic systems in various fields of application. The acquired skills and knowledge will be relevant in the areas of developing solutions for the Internet of Things (IoT), embedded systems, smart systems. Development of hardware systems for the automotive industry, industrial automation, high-performance computing, artificial intelligence and machine learning.

The purpose of the discipline "FPGA Programming Technologies" is to study digital design and design methods for FPGA, hardware description languages for the development of digital devices, the basics of digital device design using Verilog (digital design), methods and tools of functional modeling and structural synthesis in modern CAD.

The discipline enhances the following general and professional competencies:

- ЗК2 - ability to learn and acquire modern knowledge;
- ФК1 - ability to apply the legislative and regulatory framework, as well as national and international requirements, practices and standards in order to carry out professional activities in the field of computer engineering.
- ФК2 - ability to use modern methods and programming languages for development of algorithms and software.
- ФК5 - ability to use the tools and systems of design automation to development of components of computer systems and networks, Internet applications, cyber-physical systems, etc.
- ФК9 - ability to systematically support, use, adapt and operate existing information technologies and systems.
- ФК13 - ability to solve problems in the field of computer and information technologies, determine the limitations of these technologies.
- ФК14 - ability to design systems and their components taking into account all their aspects. life cycle and task, including creation, configuration, operation, maintenance and disposal.
- ФК16 - ability to design, implement and maintain the high-performance parallel and distributed computer systems and their components using FPGAs, modules and CAD systems.

In accordance with the above, strengthened general and professional competencies will provide the following learning outcomes:

- ability to design digital circuits and devices using modern design methods and equipment design languages;
- Verilog programming skills;
- ability to use Verilog hardware programming language for high-level synthesis of digital circuits;
- ability to use modern professional tools for digital design and modeling;
- skills and practical experience in using modern CAD and Inlel/Altera products;
- the ability to perform functional modeling and debugging of devices in ModelSym CAD;
- ability to perform structural synthesis in CAD Quartus II Intel/Altera.
- the ability to implement devices on modern Intel/Altera boards DE2 Board, DE10 Board, DE1 SoC Board;
- will gain experience in teamwork on common projects;
- will gain experience defending their own decisions in a professional discussion and presenting the results of their own developments;
- will gain experience using computer engineering competencies in practice;
- will receive basic knowledge of programming and digital design methods, which will allow, if necessary, to quickly switch to the use of hardware programming language VHDL, chips of other FPGA manufacturers, in particular Xilinx, as well as other CAD for functional and structural synthesis on FPGA and ASIC, in particular Synopsys, Aldec, Cadence ;
- will receive basic training for continuing professional training in the direction of hardware design for complex systems on FPGA and ASIC, systems-on-a-chip (SoC), embedded systems, smart systems. hardware and software implementations of artificial intelligence systems using FPGAs;
- will receive the potential of basic knowledge in further professional training, including independently.

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

When studying the discipline "FPGA programming technology" it is advisable to use the knowledge gained during the study of previous disciplines: "Introduction to the Linux operating system", "Computer

logic", "Theory of electric circuits and signals", "Computer electronics ", "Computer circuitry", "Computer architecture. Part 1. Control and arithmetic devices", "Computer architecture. Part 2. Processors", "Programming", "Algorithms and data structures", "Programming technologies in C for embedded systems" (selective), "Foreign language".

The discipline is basic for the courses: "Intelligent systems design technologies" (elective), "Technologies of testing (QA) of embedded systems" (elective), "Infrastructure IT project management" (elective), "Computer modeling", "Organization of computing processes ", "Computer architecture. Part 3. Microprocessor devices", "Computer architecture. Course work", "System programming", "Computer networks", "Computer systems", "Technologies of designing computer systems" (selective), "Research and design of parallel systems" (selective), "Technologies of parallel programming computer systems" (selective).

3. The structure of the credit module

Introduction

Chapter 1. Introduction to the discipline "FPGA programming technologies"

Topic 1.1. Historical aspects and trends of elemental base development.

Topic 1.2. Trends in the development of modern digital systems under the conditions of the 4th Industrial Revolution

Topic 1.3. Modern technologies of design and development of digital systems - system on a crystal (SoC) technology. The main aspects of using software logic integrated circuits (PLCs) for the development of digital systems.

Chapter 2. Overview of technologies for programming modern digital devices on FPGAs

Topic 2.1. The life cycle of the process of designing and developing hardware. Unified design flow of digital circuits (Design Flow).

Topic 2.2. The concept of functional modeling and structural synthesis of a digital circuit.

Topic 2.3. Overview of modern CAD.

Topic 2.4. Overview of hardware description languages.

Topic 3.5. Architecturally oriented design.

Chapter 3. Practical aspects of the structural synthesis of digital circuits in CAD

Topic 4.1 Overview of the DE1-SoC development board

Topic 4.2 Overview of the Intel Quartus Prime (Quartus II) design automation system.

Topic 4.4. Methods of entering the project into CAD. Graphic design. Compilation and analysis of compilation results.

Topic 4.5. Development of digital node modules. Structural synthesis and verification of the project.

Topic 4.5 Development of storage device modules

Topic 4.6 Development of control system modules

Topic 4.7. DE1-SoC board firmware.

Chapter 4. Theoretical and practical foundations of functional modeling of digital circuits

Topic 4.1. Practical aspects of the cycle of functional modeling of digital circuits in modern CAD.

Topic 4.2. Overview of the Mentor Graphics (Simens) ModelSim design automation system for functional modeling of hardware description languages.

Topic 4.3. Fundamentals of programming in the hardware description language Verilog.

Topic 4.4. Basic issues of creating TestBench levels in the Verilog language.

Topic 4.5. Examples of programming and debugging of computer hardware units in the Verilog language.

Topic 4.6. Using the built-in C debugger in the ModelSim environment.

Topic 4.6. Integration of CAD Quartus II and ModelSim.

Chapter 5. Development directions of technologies of embedded systems and devices of the Internet of Things (IoT) in the conditions of the 4th industrial revolution (Industry 4.0/IoT)

Topic 5.1. An overview of technologies related to the 4th industrial revolution

4. Educational resources and materials

4.1. Basic literature

1. Computer circuitry: Laboratory workshop [Electronic resource] : study guide for the student's educational program "Computer Systems and Networks" in specialty 123 "Computer Engineering" / Verba OA, Zhabin VI, Klymenko IA, Tkachenko VV; Igor Sikorsky Kyiv Polytechnic Institute - Electronic text data (1 file: 8.64 MB) - Kyiv : Igor Sikorsky Kyiv Polytechnic Institute, 2019. - 110 p.
2. FPGA Programming Technologies : Laboratory workshop [Electronic resource]: training . help _ for studies _ of the educational program "Computer systems and networks" by specialty 123 "Computer engineering" / O. A. Verba, V. I. Zhabin , I. A. Klymenko, V. V. Tkachenko; KPI named after Igor Sikorsky. – Electronic text data (1 file: 8.64 MB). – Kyiv: KPI named after Igor Sikorskyi, 2019. – 110 p.

4.2. Additional literature

1. Жабін В.І., Жуков І.А., Клименко І.А., Стіренко С.Г. – Арифметичні та управляючі пристрої цифрових ЕОМ: Навчальний посібник. – К.:ВЕК +, 2008. – 176 с.
2. Сергієнко А.М. VHDL для проектування обчислювальних пристроїв. – К.: ПП "Корнійчук", ТОВ "ТИДДС", 2003. – 208 с.
3. Документація до навчально-налагоджуваної плати Altera DE2 Board. Посібник користувача. DE2 Development and Education Board User Manual (terasic.com.tw). – 2008 Terasic Technologies.
4. Документація до навчальної плати DE1-SoC Board. Terasic - SoC Platform - Cyclone - DE1-SoC Board. 2022 Copyright Terasic Inc.
5. Quartus II Handbook. Volume 1. Design and Synthesis. Quartus II Handbook Version 15.0.0 (intel.com)
6. Quartus II Handbook Volume 3: Verification (intel.com) Mentor Graphics ModelSim and QuestaSim Support, Quartus II Handbook Volume 3: Verification (intel.com)

4.3. Information resources

1. The course of video lectures is on the "Sikorsky" distance learning platform in the Google Workspace for Education environment: https://classroom.google.com/c/NDg4NTYwMDIzMDA0?cjc=fb6xoef

5. Laboratory work

The purpose of laboratory work is to acquire practical skills in working with equipment based on the Inlel/Altera platform, independent creation of a digital system on the FPGA system, gaining experience in functional modeling and structural synthesis and product verification in modern CAD.

Practical tasks are first built on simple examples of programming computer hardware components. Next, an introduction to the main stages of the full cycle of development of a digital technology device is offered, including functional synthesis, structural synthesis, verification of designs and programming of a FPGA microcircuit. The basic issues of creating TestBench levels in the Verilog language are considered. CAD models ModelSym MentorGraphic, Quartus II are used to check the correctness of the created projects and debugging. Issues of integration of CAD Quartus II and ModelSim are considered. Altera debug boards DE2 Board (Cyclon II), DE10 Standard Board (Cyclon IV), DE1 SoC Board (Cyclon V) are used for experiments.

Topics of laboratory works:

Laboratory work 1. Introductory lesson. Test
Laboratory work 2. Fundamentals of digital design in the QUARTUS II design automation system. Graphic design. Structural synthesis of combinational schemes. Compilation. Verification. Board firmware.

Laboratory work 3. Fundamentals of functional modeling in the ModelSim design automation system**.** Creating a new project. Development of a digital device module. Transmission of input effects in the tcl language. Compilation, modeling, analysis of functional modeling results.
Laboratory work 4. Development of schemes of combinational devices and typical nodes of digital devices in the Verilog language.
Laboratory work 5. Creating a project with a hierarchical structure in CAD ModelSim using the example of implementing nodes for performing arithmetic operations.
Laboratory work 6. Creating a TestBench level for testing and debugging the project in the Verilog language.
Laboratory work 7. Development of own project "Arithmetic-logic block with common microoperations".
Laboratory work 8. Transfer of the project "Arithmetic-logic block with general microoperations" to CAD Quartus II. Structural synthesis. Verification. Board firmware.

6. Students' self-study work

Types of independent work:
- preparation for classroom classes (0.5 hours x 18 lectures = 9 hours);
- preparation for express tests (recommended 1-2 hours x 2 tests = 2-4 hours)
- preparation and processing of calculations based on primary data obtained in laboratory classes, performing laboratory work, solving problems, posting results on GitLab (recommended 2-4 hours x 8 laboratory works = 16 - 32 hours);
- execution of modular control work (2 MKR x 4-8 hours = 8-16 hours).

Policy and control

5. Policy of academic discipline (educational component)

Deadlines are set for the performance of laboratory work and modular control work.

Performance of laboratory work outside of the established deadlines is accompanied by penalty points, which are deducted from the grade for the protocol. Modular control work is not accepted beyond the set time.

Penalty points are issued for: untimely submission of laboratory work. The number of penalty points is no more than 10.

Bonus points are awarded for: active participation in lectures; completing current homework, keeping a summary, preparing a message with a presentation on one of the topics of the SRS discipline, etc. The number of bonus points is no more than 10.

Some lecture topics are accompanied by short express tests (for 15 minutes), which include the material of the studied topic and questions that are asked for independent study. The points obtained for the test are included in the semester rating. Current tests are not retaken.

The performance of each laboratory work is preceded by the completion of an individual task and its preparation in the form of a protocol. A student who came to class without a completed protocol is not allowed to do laboratory work. In the first stage, the student defends the results obtained during the performance of an individual task for laboratory work, in the second stage - defends the theory through an oral survey or test. Most of the laboratory works are accompanied by tests to evaluate the studied theoretical and practical material for the laboratory work. The points obtained for the performance of laboratory work, for the test and for the protocol are included in the assessment for the laboratory work.

Testing is carried out in a laboratory session after checking the results of laboratory work. A student who has not completed the individual task before the laboratory work and the test is not admitted.

Performance of laboratory work is mandatory for admission to semester control. The condition of admission to the semester control is the enrollment of all laboratory works and a starting rating of at least 30 points.

A modular test is written during a lecture session without the use of aids (mobile phones, tablets, etc.); the result is forwarded to the corresponding Google Drive directory via a Google form.

The modular control paper is not rewritten in case of a negative grade, a negative grade for the MCR (less than 9 points (<60%)) is equal to 0 points, in this case the modular control work is not counted.

The grade that a student can receive for each laboratory work and for each modular control work is given in table 1 of semester work evaluations, chapter 8 of the syllabus.

Thus, the minimum grade that a student must receive for admission to the semester exam is 60 points, the maximum is 100 points for the completion of all current works for the semester.

Applicants who have fulfilled all admission requirements (completed all laboratory work) and have a rating of less than 60 points, as well as applicants who wish to improve their rating, have the opportunity to pass a semester test in the form of a credit test at the last class on the schedule .

In the case of performance of credit control work, the rating is defined as the sum of points for credit control work and points for individual semester tasks.

The individual work of the student related to the performance of laboratory work is included in the individual semester tasks. The maximum number of points for individual work per semester is 60 points. The maximum mark for the test is 40 points. In this way, the applicant has the opportunity to increase his rating by writing a final test and adding additional points to the number of points received during the semester for individual semester work.

After completion of the credit control work, if the grade for the credit control work is higher than the rating, the applicant receives a grade based on the results of the credit control work. If the grade for the final test is lower than the rating, the applicant's previous rating (with the exception of points for the semester individual task) is canceled and he receives a grade based on the results of the final test. This option forms a responsible attitude of the applicant towards making a decision on the completion of the credit control work, forces him to critically assess the level of his training and carefully prepare for the credit.

6. Types of control and rating system for evaluating learning outcomes (RSE)

The student's semester rating from the credit module is calculated based on a 100-point scale. The rating consists of the points that the student receives for completing 8 laboratory works R_L, two modular control works R_{MCW} and expert tests R_{ET}.

The maximum number of points for laboratory work is 60 points, i.e R_L =60.

The generalized criteria for evaluating laboratory work are as follows:

- the timeliness of the preparation of the protocol for the laboratory session, completeness of the theoretical or practical task in the protocol, the protocol is posted on GitLab on time;
- correct functioning of the developed models on software or hardware, demonstration of own repository on GitLab with laboratory work materials and availability of commits;
- a survey on the subject of laboratory work for crediting the practical part of the work, protection of the results obtained in the work, answers to additional theoretical questions of the teacher, completeness of the report/protocol on the work on GitLab.

A detailed approach to the assessment of each laboratory work is given in Table 1.

Table 1. Details of the evaluation of each laboratory work

The name of the class	Form of control	Scores	Admission to the exam by automatic evaluation	Total points
Laboratory work 1.	Entrance test	4	2	4
Laboratory work 2	Completing the task	3	5	8
	Polling	3		
	Protocol on GitLab	2		
Laboratory work 3	Completing the task	3	5	8
	Polling on QA	3		
	Protocol on GitLab	2		
Laboratory work 4	Completing the task	3	5	8
	Polling on QA	3		
	Protocol on GitLab	2		
Laboratory work 5	Completing the task	3	5	8
	Polling	3		
	Protocol on GitLab	2		
Laboratory work 6	Completing the task	3	5	8
	Polling	3		
	Protocol on GitLab / demonstration	2		
Laboratory work 7	Completing the task	3	5	8
	Polling	3		
	Protocol on GitLab / demonstration	2		
Laboratory work 8	Completing the task	3	5	8
	Polling	3		
	Protocol on GitLab / demonstration	2		
Number of points for individual work				60
Express tests at lectures	2 x 5	10	5	10
Modular control work	MCW1 (Тест)	15	9	15
	MCW2	15	9	15
Total points		100	60	100

The maximum number of points per MCW R_{MCW} = 2x15 = 30 points.

MCW1 is conducted in the form of automated testing on the Google Workspace for Education platform. The test consists of 60 questions R_{MCW_2} = 0,25x60 = 15 points

Modular control work MCW2 is performed independently according to an individual task. MCW2 assessment criteria at four levels:

- correct and meaningful answer with explanations in the terms of the subject area: 13 - 15 points;
- correct answer, incomplete explanations: 11 - 12 points;
- the answer contains errors: 9 - 10 points;
- the answer contains significant errors, there are no explanations: 4-8 points;
- no answer: 0 points.

The score for MCW2 is reduced by:

- incorrect registration;
- lack of comments in meaningful terms;
- lack of explanations during calculations.

The maximum number of points for express tests is 10 points, tests are conducted during lectures in the form of automated testing on the Google Workspace for Education platform.

The maximum number of points for the credit control work is R_T = 40 points.

The credit control work is conducted in the form of automated testing on the Google Workspace for Education / moodle platform, consisting of selected questions that were during the semester in MCW, express tests, and defenses of laboratory works. The maximum score for the credit control work R_T = 40 points.

Calendar certification of students (for 8 and 14 weeks of semesters) in the discipline is carried out according to the value of the student's current rating at the time of certification. If the value of this rating is at least 50% of the maximum possible at the time of certification, the student is considered certified. Otherwise, the attestation information is marked as "uncertified".

The number of points a student receives per semester is determined by the formula

$$\mathbf{R = R_L + R_{MCW} + R_{ET}.}$$

The maximum number of points per semester does not exceed RS = 100.

Taking into account the received sum of points, the final grade is determined according to table 3.

If a student writes a test paper, the number of points the student receives per semester is determined by the formula

$$\mathbf{R = R_{IP} + R_T}$$

where, $\mathbf{R_{IP} = R_L.}$

The maximum number of points per semester does not exceed R = 100.

Taking into account the received sum of points, the final grade is determined by table 3.

Table 2. Determination of the semester grade

Scores	*Rating*
100-95	Perfectly
94-85	Very good
84-75	Good
74-65	Satisfactorily
64-60	Enough
Less than 60	Unsatisfactorily
Admission conditions not met	Not allowed

Working program of the academic discipline (syllabus)

Working program of the academic discipline (syllabus):
Compiled by, senior teacher of the CE department Vynogradov Yuriy Mykolayovych
editor, Ph.D., docent, professor of the department of CE Klymenko Iryna Anatoliivna
Approved by the Department of Computer Engineering (protocol No. 10 dated 25.05.2022)
Agreed by the methodical commission of FICT (protocol No. 10 dated 09.06.2022)