COURSE DESCRIPTION CARD - SYLLABUS

Course name

Programmable logic design

Course

Field of study	Year/Semester
Computing Science	1/2
Area of study (specialization)	Profile of study
Internet of Things	general academic
Level of study	Course offered in
Second-cycle studies	Polish
Form of study	Requirements
full-time	elective

Number of hours

Lecture	Laboratory classes	Other (e.g. online)
30	30	
Tutorials	Projects/seminars	

Number of credit points

5

Lecturers

Responsible for the course/lecturer:

dr inż. Marek Kropidłowski

email: marek.kropidlowski@put.poznan.pl

tel: 616652297

Instytut Informatyki

adres: ul. Piotrowo 3 60-965 Poznań

Responsible for the course/lecturer:

dr hab. inż. Paweł Śniatała

email: pawel.sniatala@put.poznan.pl

tel: 616652297

Instytut Informatyki

adres: ul. Piotrowo 3 60-965 Poznań

Prerequisites

Knowledge: Student starting this module should have a basic knowledge in the field of analogue and digital electronics, structured programming and scripting languages.

Skills: The student should be able to obtain information from the indicated sources, as well as understand the need to expand his competences and be ready to cooperate in a team.

Social Competences: The student should show such features as: honesty, responsibility, perseverance, cognitive curiosity, creativity, personal culture, respect for other people.

Course objective

1. To provide students with knowledge on the construction and operation of programmable devices typically used in modern digital systems and IoT.

2. Present students a set of development technologies for modeling digital devices, designing reusable components, FPGA-based prototyping.

3. Developing students' skills in solving technical problems in the field of complex digital system design.

4. Shaping teamwork skills in students - the ability to cooperate in the design teams and in the preparation of final research reports.

Course-related learning outcomes

Knowledge

1. has advanced and detailed knowledge related to selected areas of computer science, developing digital systems, prototyping system-on-chip for hardware verification,
2. has knowledge about new technologies in the area of hardware-software development and embedded systems
3. has advanced and detailed knowledge regarding hardware life cycle which involves developing a reprogrammable hardware system and testing it.

Skills

1. is able to acquire, combine, interpret and evaluate information from literature, databases and other information sources (in mother tongue and English); draw conclusions, and formulate opinions based on it,
2. is able to combine knowledge from different areas of computer science (and if necessary from other scientific disciplines) to formulate and solve engineering tasks related to hardware-software development,
3. is able to design and develop a hardware layer of complex digital system,
5. is able to design (according to a provided specification which includes also non-technical aspects) a digital system using technologies learned during the course,
6. is able to work in a group, performing a role of programmable hardware designer.

Social competences

1. understands that knowledge and skills related to computer science quickly become obsolete,
2. knows how new development technologies and tools could be helpful to solve practical problems like developing a digital system.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Formative assessment:

a) lectures: based on the answers to the questions which test understanding of material presented on the lectures

b) laboratory classes: based on the assessment of the tasks done during classes and as a homework

Summative assessment:

a) verification of assumed learning objectives related to lectures within an online written test. The final grade is determined using the following scale: (90%, 100%] -> 5.0, (80%, 90%] -> 4.5, (70%, 80%] -> 4.0,

(60%, 70%] -> 3.5, (50%, 60%] -> 3.0, (0%, 50%] -> 2.0.
b) verification of assumed learning objectives related to laboratories is based on:
- design contest and verification of the laboratory tasks. The final grade is determined using the following scale: (90%, 100%] -> 5.0, (80%, 90%] -> 4.5, (70%, 80%] -> 4.0, (60%, 70%] -> 3.5, (50%, 60%] -> 3.0, (0%, 50%]-> 2.0.

Getting extra points for activity during classes, especially for:
- proposing to discuss additional aspects of the issue,
- effectiveness of applying the acquired knowledge while solving a given problem,
- ability to work within a team that practically performs a specific task in a laboratory,
- comments related to the improvement of teaching materials.

Programme content

The lecture program includes the following topics:
- VHDL IEEE 1076-2008 modeling concepts
- RTL description, HDL good practice, coding tips and techniques
- combinational and sequential logic description, FSM modeling and synthesis
- test bench and verification features
- reusable and parameterized models
- FPGA-base implementation of digital system
- modern programmable structures: FPGA/CPLD/FPAA/FPOA/3D-PLD/PSoC
- IP-centric paradigm in digital design
- partial and dynamic reconfiguration (PDR/IRL)
- “softcore” processors and heterogeneous FPGA architectures (LEON, Microblaze, Zynq)
- FPGA-based prototyping and verification technique (on-chip instrumentation, in-system debug, hardware-in-the-loop)

Laboratory classes are conducted in the form of 2-hour lab exercises, preceded by a 2-hour instructional session at the beginning of the semester. Exercises are carried out by 2-person teams.
The program of laboratory classes includes the following topics:
- synthesis results vs HDL coding style (Xilinx XST, Mentor Graphics Physical Synthesi)
- advanced testing using VHDL (Mentor Graphics Modelsim)
- hardware-software codesign (Xilinx ISE, XSDK)
- IP-cenric design (Xilinx Vivado)
- FPGA-based prototyping (Xilinx ILA, OCI)

Teaching methods

1. Lecture with multimedia presentation (diagrams, formulas, definitions, etc.) supplemented by the content of the board.

2. Laboratory exercises: multimedia presentation, presentation illustrated with examples given on the board and performance of tasks given by the teacher - practical exercises.

Bibliography

Basic

1. Mark Zwoliński, Projektowanie układów cyfrowych z wykorzystaniem języka VHDL, WKŁ2007, ISBN: 9788320616354.
2. Andrew Rushton, VHDL for Logic Synthesis, Third Edition, John Wiley & Sons, 2011, ISBN: 978-0-470-68847-2
3. Scott Hauck, Andre DeHon, Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation / Edition 1, Elsevier Science, November 2007, ISBN: 0123705223

Additional

1. Michael Keating, Pierre Bricaud, Reuse Methodology Manual for System-on-a-Chip Designs / Edition 3, Springer-Verlag New York, August 2007, ISBN: 0387740988
2. Peter J. Ashenden, The Designer's Guide to VHDL / Edition 3, Elsevier Science, June 2008, ISBN:0120887851.
3. Piotr Zbysiński, Jerzy Pasierbiński, Układy programowalne pierwsze kroki, Wydanie II, Wydawnictwo BTC, Warszawa 2004, ISBN: 83-910067-0-0
4. Steve Kilts, Advanced FPGA Design: Architecture, Implementation, and Optimization, John Wiley & Sons, June 2007, ISBN: 0470054379.

Breakdown of average student's workload

	Hours	ECTS
Total workload	122	5
Classes requiring direct contact with the teacher	64	2,6
Student's own work (literature studies, preparation for laboratory classes, preparation for tests, techical reports preparation) [1]	58	2,4

[1] delete or add other activities as appropriate