Załącznik Nr 5 do Zarz. Nr 33/11/12

Z1-PU7	WYDANIE N1	Strona 1 z 3

(*faculty stamp*)

COURSE DESCRIPTION

1. Course title: MICROPROCESSORS	**2. Course code MS**
3. Validity of course description: 2016/2017	
4. Level of studies: BA, BSc programme (1st cycle of higher education)	
5. Mode of studies: intramural studies / ~~extramural studies~~	
6. Field of study: Control, Electronic and Information Engineering (CEIE)	(FACULTY SYMBOL) RAu
7. Profile of studies: general	
8. Programme: all	
9. Semester: 5	
10. Faculty teaching the course: Institute of Electronics, (RAu-3)	
11. Course instructor: Adam Milik PhD	
12. Course classification: common subjects	
13. Course status: compulsory /elective	
14. Language of instruction: English	

15. Pre-requisite qualifications:

Course attendants are supposed to have general knowledge of digital circuits operation, digital systems design, logic circuits implementation algorithms, basics of programming languages, algorithm development and implementation, principles of computer operation.
Students are also supposed to possess practical skills concerning programming and algorithm implementation with use of high level programming language.

16. Course objectives:

Microprocessor architecture and its operation, instruction execution, bus architecture, interrupt system,
Microprocessor system architecture, interaction with memory and peripheral components, interrupt system concept and implementation, programming with use of assembly language and high level languages.
Implementation of numerical calculations and its performance, selected numerical algorithms (Bresenham, Newton-Raphson, CORDIC)
Basics of serial interfaces, serial data transfer concepts and implementation,
Basics of compilers and automatic recognition of sentences, high level languages from compiler (automatic tools) point of view
Practical aspects of design and implementation of embedded systems, programming and debugging, on chip debug systems, design of custom peripherals, system integration, simulation and modelling of systems

17. Description of learning outcomes:

Nr	Learning outcomes description	Method of assessment	Teaching methods	Learning outcomes reference code
1.	Has knowledge of arithmetic principles of logic circuits, designing and operation of logic sequential and microprogrammable circuits and embedded systems	Test, laboratory task, project	lecture, laboratory, project	K_W10
2.	Knows fundamental data structures and is able to select appropriate one for given problem	test, laboratory task, project	lecture, laboratory, project	K_W12
3.	Has knowledge of computer systems architecture, embedded systems and implementation of simple computer systems	Test, laboratory task, project	lecture, laboratory, project	K_W14
4.	Is able to obtain information from literature and other data sources. Is able to interpret them.	test, laboratory tasks, project	lecture, laboratory, project	K_U1
5.	Is able to analyze algorithms and estimate their complexity	test, project	lecture, laboratory, project	K_U11

6.	Is able to design and program simple microprocessor systems and embedded systems. Is able to use high and low level programming languages and respective development and diagnostic tools	laboratory tasks, project	lecture, laboratory, project	K_U17
7.	Is able to use CAD/CAE software	laboratory tasks, project	lecture, laboratory, project	K_U24
8.	Understands necessity of continues learning	test, laboratory tasks	lecture, class, laboratory	K_K1
9.	Is able to work in group, takes up different goals	laboratory tasks	laboratory	K_K3

18. Teaching modes and hours

Lecture: 15h, Laboratory: 30h, Project: 15h

19. Syllabus description:

Lecture:

Introduction to microprocessors, Evolution from finite state machine through the microprogrammable device to microprocessor. Basic functional blocks of microprocessor and its features. Architecture influence to behavior and performance of computing system.
Instruction execution. Harvard and Princeton (von Neumann) architectures. Sequential and pipelined execution concepts
Instruction set. Argument addressing, memory model and addressing modes, memory protection and virtualization.
Bus architecture of computer system. Bus interface cycles. Controlling timing of cycles. Connecting memories and peripheral devices to microprocessor. Taking bus control over.
Interrupt system. Implementation concept. Classification of interrupt systems. Concept of vector interrupt system.
Serial interfaces RS232, 1-Wire, I2C, USB. Principles of data transmission and reception. Synchronization of receiver, asynchronous and synchronous modes. Data integrity check methods.
Arithmetic. Numeral systems. Integers signed and unsigned. Fixed point numbers. Floating point numbers. Arithmetic operations. Numerical algorithms: line drawing (Bresenham's), trigonometric calculations and rotations (Cordic), Solving non-linear equation (Newton-Raphson)
Basics of automatic statement processing. BNF notation, Syntax diagram. Data flow graphs – intermediate representation of statements.
Hardware Description Language (HDL) introduction to Verilog HDL. HDL vs Programming Language. Basics of automatic synthesis and implementation. Description of combinatorial blocks. Description of sequential blocks.

Class:

Introduction to programming of microprocessor with use of assembler and high level languages. Introduction to assembler. Simple programs in assembler. Determining and implementing a conditionally executed blocks. Design of subprograms. Argument passing to subprograms. Implementing interrupt service routines. Interfacing with peripheral devices.

Laboratory:

Using C/C++ in embedded systems without operating system (OS). Controlling resources of microcontroller. Servicing interrupts. Interfacing real hardware. Handling: displays LED, LCD and graphic LCD, keyboards, sensors. Concurrent operation of the program. Optimization of program for embedded systems. Running On Chip Debug with CAD EDA programming tools
Introduction to HDL logic synthesis and FPGA technology. Implementation of peripheral devices. Creating and using bus functional models in verification. Binding the custom peripheral device with high level programming language.

Projects:

1. Implementation of mundane devices like advanced alarm clocks, timers, cycle computers etc. Students are supposed to implement software layer of the project that binds together simple hardware devices like displays, keyboards and other sensors into fully functional system. The attention is paid to simplicity of use and correct user interface simplicity.
2. Implementation of custom hardware device that supports operation of main design problem. Usually it is a specialized interface unit or arithmetic operation support device. Important part of the design concerns creating, modeling, synthesizing, implementing the device and linking it with microprocessor. The hardware component is linked with software operating by preparing appropriate declaration and drivers. Finally the component is used inside the design to proof its functionality.

20. Examination: (no exam assigned)

21. Primary sources:

M. Morris Mano Computer System Architecture
M. Morris Mano, Charles R. Kime Logic and Computer Design Fundamentals.
Wayne Wolf Computers as Components: Principles of Embedded Computing Systems Design
Steven Furber ARM System-on-chip Architecture
Niklaus Wirth Algorithms + Data Structures = Programs
Brian W Kernighan,.Dennis M. Ritchie The C Programming Language
Clive “MAX” Maxfield, Alvin Brown: The Definitive Guide to How Computers Do Math

22. Secondary sources:

Samir Palnitkar, Verilog HDL, SunSoft Press 1998
Jean-Pierre Deschamps, Gery J.A. Bioul, Gustavo D. Sutter Synthesis of Arithmetic Circuits: FPGA, ASIC and Embedded Systems

23. Total workload required to achieve learning outcomes

Lp.	Teaching mode :	Contact hours / Student workload hours
1	Lecture	15 / 15
2	Classes	0 / 0
3	Laboratory	30 / 15
4	Project	15 / 30
5	BA/ MA Seminar	/
6	Other	/
	Total number of hours	60 / 60

24. Total hours:120

25. Number of ECTS credits: 4

26. Number of ECTS credits allocated for contact hours: 1

27. Number of ECTS credits allocated for in-practice hours (laboratory classes, projects): 3

26. Comments:

Approved:

…………………………… ………………………………………………

(date, Instructor's signature) (*date , the Director of the Faculty Unit signature)*