1. Course title: MICROPROCESSORS
2. Course code: UP


4. Level of studies: BA, BSc programme

5. Mode of studies: intramural studies / extramural studies

6. Field of study: ELECTRONICS AND TELECOMMUNICATIONS (FACULTY SYMBOL) AC, E & CS

7. Profile of studies: general

8. Programme: all

9. Semester: 5, 6

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 and design, algorithm implementation, basics of programming languages and principles of computer operations.
Students are also supposed to possess practical skills concerning programming and algorithm implementation with use of high level programming language. It is assumed that students passed the following courses: Fundamentals of Computer Programming (WDI, PPK), Theory of Logic Circuits (PTC), Digital Circuits Design (PUC), Introduction to Computer Systems (SK).

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 assembler and high level languages.
Implementation of numerical calculations and its performance, selected numerical algorithms (Bresenham, Newton-Raphson, CORDIC)
Basics of serial interfaces, methods of data transfer,
Basics of compilers and automatic recognition of sentences, high level languages seen from compiler view
Practical aspects of design and implementation of embedded systems, programming and debugging, design of custom peripherals, system integration

17. Description of learning outcomes:

<table>
<thead>
<tr>
<th>Nr</th>
<th>Learning outcomes description</th>
<th>Method of assessment</th>
<th>Teaching methods</th>
<th>Learning outcomes reference code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>has knowledge of computer architecture with emphasis of hardware architecture</td>
<td>exam, project</td>
<td>lecture, class, project, laboratory</td>
<td>K1_W6</td>
</tr>
<tr>
<td>2</td>
<td>has knowledge of low level and high level programming methods</td>
<td>exam, project</td>
<td>lecture, class, project, laboratory</td>
<td>K1_W7</td>
</tr>
<tr>
<td>3</td>
<td>has detailed knowledge of computer architecture and programming of embedded microprocessor systems</td>
<td>exam, project</td>
<td>lecture, class, project, laboratory</td>
<td>K1_W8</td>
</tr>
<tr>
<td>4</td>
<td>is able to formulate specification of microprocessor systems based on delivered requirements</td>
<td>project, laboratory</td>
<td>class, project, laboratory</td>
<td>K1_U14</td>
</tr>
<tr>
<td>5</td>
<td>is able to design electronic circuits and systems for different purposes and given requirements</td>
<td>project, laboratory</td>
<td>class, project, laboratory</td>
<td>T1A_U16</td>
</tr>
<tr>
<td>6</td>
<td>is able to design, develop and verify electronic circuit or simple electronic system.</td>
<td>project, laboratory</td>
<td>class, project, laboratory</td>
<td>T1A_U16</td>
</tr>
</tbody>
</table>
7. Is able to formulate algorithm, use high level programming languages, use assembly language. Is familiar with use of programming, simulation and debugging EDA tools.
   project, laboratory  class, project, laboratory  T1A_U22

8. Is able to work in group. Is able to take responsibility for assigned tasks. Is aware of responsibility for its own work and ready to work in group following its rules
   project, laboratory  project, laboratory  T1A_K03  T1A_K04

18. Teaching modes and hours
   Lecture: 30h, Class: 15h, Laboratory: 30h Project: 30h

19. Syllabus description:
   **Lecture:**
   Introduction to microprocessor, Evolution from finite state machine through the microprogrammable device to microprocessor. Basic functional blocks of microprocessor and its feature are discussed. The influence to behavior and performance of computing system is shown.
   Instruction execution. Harvard and Princeton (von Neumann) architectures. Sequential and pipelined execution concepts
   Instruction set. Argument addressing model and addressing modes.
   Interrupt system. Implementation concept. Classification of interrupt systems. Concept of vectorized interrupt system.
   Arithmetic. Numerical systems. Integers signed and unsigned. Floating point numbers. Arithmetic operations. Numerical algorithms: line drawing (Bresenham's), trigonometric calculations and rotations (Cordic), Solving non-linear equation (Newton-Raphson)
   Hardware Description Language (HDL) introduction to Verilog HDL. HDL vs Programming Language. Basic of automatic synthesis. Description of combinational 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: semester VI

21. Primary sources:
   M. Morris Mano Computer System Architecture
   Steven Furber ARM System-on-chip Architecture
   Niklaus Wirth Algorithms + Data Structures = Programs
22. Secondary sources:
Samir Palnitkar, Verilog HDL, SunSoft Press 1998
Brian W Kernighan, Dennis M. Ritchie The C Programming Language
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

<table>
<thead>
<tr>
<th>Lp.</th>
<th>Teaching mode</th>
<th>Contact hours / Student workload hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Lecture</td>
<td>30 / 10</td>
</tr>
<tr>
<td>2</td>
<td>Classes</td>
<td>15 / 30</td>
</tr>
<tr>
<td>3</td>
<td>Laboratory</td>
<td>30 / 15</td>
</tr>
<tr>
<td>4</td>
<td>Project</td>
<td>30 / 45</td>
</tr>
<tr>
<td>5</td>
<td>BA/MA Seminar</td>
<td>- / -</td>
</tr>
<tr>
<td>6</td>
<td>Other</td>
<td>15 / 20</td>
</tr>
<tr>
<td></td>
<td>Total number of hours</td>
<td>120 / 120</td>
</tr>
</tbody>
</table>

24. Total hours: 240

25. Number of ECTS credits: 8

26. Number of ECTS credits allocated for contact hours: 4

27. Number of ECTS credits allocated for in-practice hours (laboratory classes, projects): 4

26. Comments:

Approved:

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