MINISTRY OF EDUCATION AND SCIENCE

TECHNICAL UNIVERSITY - VARNA

FACULTY OF COMPUTER SCIENCES AND AUTOMATION

APPROVED

DEAN:....................................

(Assoc. Prof. Eng. Nedyalko Nicholloff, PhD)

SYLLABUS

Discipline „*MICROPROCESSORS*“, code: 21

Included in the Curriculum of the specialty: *SOFTWARE AND INTERNET TECHNOLOGIES*

Professional field of study: *COMMUNICATION AND COMPUTER EQUIPMENT – 5.3.*

Higher Education Qualification: *BACHELOR'S DEGREE.*

Faculty, providing the organizational and methodological training: *FACULTY OF COMPUTER SCIENCES AND AUTOMATION.*

Department, providing instruction on the discipline: *COMPUTER SCIENCE AND ENGINEEING*

Varna, 2018

Excerpt from the curriculum

No by order	Name of the discipline	Forms of assessment				Auditorium workload						Extracurricular activities	Student total workload	Credits
		Examination	Continuous Assessment	Course project	Pass / Failed	Lectures	Seminars			Laboratory classes	Total hours			
							Seminar classes	Course project	Course work					
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
21	Microprocessors	●				15				30	45	80	125	5

Semester: *FOURTH*

ANNOTATION

The discipline acquaints the students with the architecture of the 32-bit microprocessors. The main microprocessor structural units, the system bus, its signals and various cycles, the programming model, instruction set, memory addressing modes, exceptions and interrupts are examined. The C++, C and Assembly programming languages are used.

A. Lectures (Topics)

Topic 0.	<u>Introduction:</u> Development of the microelectronic technologies for VLSI IC production. Brief history of the 32-bit microprocessors (MP) x86 and „ARM“.	1 hour
Topic 1.	<u>Programming model of the MP:</u> The concept of programming model. Modes of operation. General purpose registers. Specialised registers. Flags of the condition code register (CCR). Features. Overview of the programming model of other MPs.	1 hour
Topic 2.	<u>Instruction set:</u> Instruction types. Instruction format. „Operand 2“. Addressing modes. Instruction set orthogonality.	2 hours
Topic 3.	<u>Structure of the MP:</u> Main functional units in the MP. Internal busses. Pipeline operation.	2 hours
Topic 4.	<u>System bus:</u> Address and data bus signals. Control signals. Data transfer organisation. Cycle types. Timing diagrams.	2 hours
Topic 5.	<u>Floating point unit:</u> Multiply and accumulate, Divide and square root, and Load/store pipelines. Operating modes. Short vector processing. Register file. Programming model. Instructions. Exceptions.	2 hours
Topic 6.	<u>Exceptions and interrupts:</u> Exceptions. Interrupts – types and connection to the MP modes. Exception and interrupt vector table. Reset of the MP.	1 hour
Topic 7.	<u>Memory management unit:</u> Functions. Registers. Address translation. Descriptors. Caching and buffering. Faults. Write buffer.	1 hour
Topic 8.	<u>MP architecture development:</u> Development of the MP to 64-bit architecture. Graphics processors. Manycore MPs.	2 hours
Topic 9.	<u>Brief information about other MPs:</u> Conditional branches and carry in MPs without a CCR („Alpha“, MIPS). MPs with a „register window“ (SPARC). „Hello, world!“ programmes for various MPs and operation systems (OS).	1 hour

Total: 15 hours

B. Seminars

NONE

Total: 0 hours

C. Laboratory classes

Topic 0.	Acquaintance with the nanocomputer, OS „Linux“, the „nano“ editor and the „make“ utility. Hexadecimal system. Bitwise operations in C and C++. The „bitset“ template.	2 hour
Topic 1.	Programming model. Working with instruction and addressing mode reference tables. Debugging a programme with „gdb“.	2 hours
Topic 2.	Data transfer instructions. Arithmetic instructions. Linear Assembly programmes.	2 hours
Topic 3.	Control transfer instructions. Conditional and unconditional branches. Branched programmes.	2 hours
Topic 4.	Cyclic programmes.	2 hours
Topic 5.	Bit manipulation instructions.	2 hours
Topic 6.	First midterm test.	2 hours
Topic 7.	Memory data transfer instructions. Addressing modes. Working with [sets of] bytes, half-words and words. Unknown programme operation analysis.	2 hours
Topic 8.	Working with double words. Working with arrays – part I.	2 hours
Topic 9.	Working with arrays – part II.	2 hours
Topic 10.	Working with arrays – part III.	2 hours
Topic 11.	Working with strings of characters.	2 hours
Topic 12.	Second midterm test.	2 hours
Topic 13.	Transforming numbers into strings of characters and vice versa – part I.	2 hours
Topic 14.	Transforming numbers into strings of characters and vice versa – part II.	2 hours

Total: 30 hours

D. Practical training

NONE

Total: 0 hours

E. Course project

NONE

Total: 0 hours

F. Forms and organisation of the assessment throughout the semester

Forms of assessment throughout the semester	Score – K1
Active participation in the laboratory classes (with comprehension)	10
Tests	20
Total	30

G. Type of assessment (procedure)

Type of assessment	Score – K2
Exam – written, with oral discussion	70

Final assessment points: K = K1 + K2

H. Reference

A. Basic:

1. Alex van Someren, Carol Atack, “The ARM RISC Chip: A Programmer’s Guide”, ISBN 978-0201624106, Addison Wesley, 1994.
2. The ARM Cookbook (ARM DUYI-0005B), ARM Ltd, 1994.
3. ARM 810 Preliminary Data Sheet (ARM DDI 0081E), ARM Ltd, 1996.
4. ARM Architecture Reference Manual (ARM DDI 0100B), ARM Ltd, 1996.
5. ARM Instruction Set Quick Reference Card’99 http://zap.org.au/elec2041-cdrom/reference/arm-instructions-quickref.pdf
6. VFP11 Vector Floating-Point Coprocessor Technical Reference Manual (ARM DDI 0274B), ARM Ltd, 2003.
7. William Hohl, Christopher Hinds, “ARM Assembly Language: Fundamentals and Techniques”, ISBN 978-1482229851, CRC Press, 2014.
8. Peter J. Knaggs, “ARM Assembly Language Programming”, Trowbridge, 2016.
9. Larry D. Pyeatt, “Modern Assembly Language Programming with the ARM Processor”, ISBN 978-0128036983, Newnes, 2016.
10. Dezső Sima, http://users.nik.uni-obuda.hu/sima/letoltes/magyar/SZA2016_osz/nappali/ARM_processors_lecture_2016_12_07.pptx

B. Additional:

1. ARM Datasheet, ISBN 1852500263, Acorn Computers Ltd, 1987.
2. Peter Norton, “Advanced Assembly Language”, ISBN 0136587747, Brady, 1991.
3. ARM Software Development Toolkit Version 2.0: Programming Techniques (ARM DUI 0021A), ARM Ltd, 1995.
4. ARM Instruction Set Quick Reference Card (ARM QRC 0001H), ARM Ltd, 2003.
5. Andrew N. Sloss, Dominic Symes, Chris Wright, “ARM System Developer’s Guide: Designing and Optimizing System Software”, ISBN 978-1558608740, Morgan Kaufmann, 2004.
6. The ARM Instruction Set – ARM University Program – V1.0 (no ID), ARM Ltd, 2009.
7. ARM1176JZF-S Technical Reference Manual (ARM DDI 0301H), ARM Ltd, 2009.
8. ARM Cortex-A Series Programmer’s Guide (ARM DEN0013D), ARM Ltd, 2014.
9. ARM Architecture Reference Manual – ARMv7-A and ARMv7-R edition (ARM DDI 0406C.c), ARM Ltd, 2014.
10. Procedure Call Standard for the ARM Architecture (ARM IHI 0042F), ARM Ltd, 2015.
11. Stephen Byram Furber, “ARM System-on-Chip Architecture”, ISBN 978-9332555570, Pearson India, 2015.

Author: ....
(Chief Assist.-Prof. Eng. L. Gueorguieff, PhD)

The programme was discussed at a Department Council meeting of the Department of *COMPUTER SYSTEMS AND TECHNOLOGIES*, protocol No. ..../.................

Head of Department: ....
(Assoc. Prof. Eng. Chr. Vultchanoff, PhD)

The programme was discussed at a Department Council meeting of the Department of *SOFTWARE AND INTERNET TECHNOLOGIES*, protocol No. ..../.................

Head of Department: ....
(Assoc. Prof. Eng. Violetta Bozhikova)

The programme was approved at a Faculty Council meeting, protocol No. ..../.............

Dean: ....
(Assoc. Prof. Eng. Nedyalko Nicholloff, PhD)

Code: 21 „Microprocessors“

ECTS credits: 5 Forms of assessments: Exam	Number of hours per week: 1+0+2 Types of assessment: Exam – written, with oral discussion
Department, providing instruction on the discipline: Department of *COMPUTER SYSTEMS AND TECHNOLOGIES.* *FACULTY OF COMPUTER SCIENCES AND AUTOMATION*	

Lecturer: Chief Assist.-Prof. Eng. L. Gueorguieff, PhD

Department: *COMPUTER SYSTEMS AND TECHNOLOGIES*

Tel.: 383628

e-mail: lig@tu-varna.bg

Annotation:

The architecture of the 32-bit microprocessors is studied in the discipline: operational unit structure, internal organisation of the microprocessor, main method of information interchange and instruction set. The programming model of the microprocessor is defined and its registers, addressing modes, exceptions and interrupts are examined. At the laboratory classes, the Assembly language and the basic programming structures that can be implemented with it are studied. The goal here is to master the instructions and the organisation of the computing process, going deep into the operation of the microprocessor at the lowest level up to a single bit. Skills for algorithmisation of linear, branched, cyclic and combined programming structures and their optimal implementation in Assembly, including with translation from the C language, are developed. Calling of subroutines written in Assembler from I/O programmes written in the C language is studied in practice, paying attention to the parameter transfer and returning the result.

The knowledge gained from studying the disciplines „Computer organisation and architectures“, „Basic programming“ и „Object-oriented programming – part I“ is relied upon. The knowledge on microprocessors is used in the discipline „Language processors“.

Main issues of the syllabus content:

1. Structure of the microprocessor, basic units. Programming model, registers.
2. Instruction set. Instruction types. Memory addressing methods.
3. System bus and signals of the microprocessor.
4. Vector floating point unit.
5. Exceptions and interrupts.
6. Memory management unit.
7. Microprocessor architecture development.

Content presentation:

The lectures explain the microprocessor architecture features and make clear the logical structures resulting from them in the sense of the traditional computing criteria. The laboratory classes illustrate in detail the capabilities of the microprocessor through studying the instruction set at the Assembly language level. A special attention is paid on working with the carry, bitwise operations, addressing methods, data arrays, conditional branches, techniques of programming in Assembler and its connection with the C language.