SYLLABUS[1]

1. Information about the Program

1.1 Higher education institution	Politehnica University of Timişoara
1.2 Facultaty[2] / Departament[3]	Automation and Computers/ Computers
1.3 Chair	-
1.4 Domain of study	Computers and Information Technology
1.5 Study level	Bachelor
1.6 Study programme / Qualification	Computers / engineer

2. Information about the Course

2.1 Course			**Computer Organization**				
2.2 Lecturer			S.l. dr. ing. Mihai Udrescu				
2.3 Academic staff for seminars/labs			drd. ing. Alexandru Topîrceanu				
2.4 Study year	2	2.5 Semester	4	2.6 Assessment type	E	2.7 Course type	Mandatory

3. Total time estimated (hours/ semester of didactical activities)

3.1 Hours / week	5	of which: 3.2 lecture hours	2	3.3 seminar/lab hours	2
3.4 Total curriculum hours	135	of which: 3.5 lecture hours	28	3.6 seminar/lab hours	42
Time distribution					hours
Study using manuals, support materials, bibliography and notes					30
Supplementary documentation in library, speciality electronic platforms and on site					7
Supplementary preparation for seminars/labs, homeworks, reviews, portofolios and essays					28
Tutoring activities					2
Exams					4
Others					

3.7 Total - hours of individual study	65
3.8 Total - hours per semester	141
3.9 Credits	5

4. Prerequisites (if appropriate)

4.1 curriculum	• Computer Architecture, Digital Logic
4.2 competencies	• Basic physics, mathematics, hardware and software knowledge

5. Conditions (if appropriate)

5.1 for lectures	• Lecture room, support devices: laptop, presenter, projector, blackboard.
5.2 for seminars/labs	• Laboratory room with 15-20 computers – programming environment for Android applications, for hardware design using Modelsim, Altera DE2 development board, blackboard

6. Specific competencies acquired

Professional competencies [4]	• Operating with fundamentals of sciences, engineering, and computer science • Designing hardware, software and communication components • Problem solving using the instruments of computer science and engineering

[1] Formularul corespunde Fişei Disciplinei promovată prin OMECTS 5703/18.12.2011 (Anexa3);

[2] Se înscrie numele facultăţii care gestionează programul de studiu căruia îi aparţine disciplina;

[3] Se înscrie numele departamentului căruia i-a fost încredinţată susţinerea disciplinei şi de care aparţine titularul cursului;

[4] Aspectul competenţelor profesionale va fi tratat cf. Metodologiei OMECTS 5703/18.12.2011. Se vor prelua competenţele care sunt precizate în Registrul Naţional al Calificărilor din Învăţământul Superior RNCIS (http://www.rncis.ro/portal/page?_pageid=117,70218&_dad=portal&_schema=PORTAL) pentru domeniul de studiu de la pct. 1.4, programul de studii de la pct. 1.6 din această fişă şi materia în cauză

	• Improving the performance of hardware, software and communication systems • Designing, managing the lifecycle, integration and integrity of hardware, software and communication systems
Transversal competencies	

7. Objectives of the course (issued from the list of the comptenecies acquired)

7.1 Aim	• Acquiring the basic knowledge required by specifying, designing and assessing modern computer systems with an emphasis on memory hierarchies and pipelined structures
7.2 Specific objectives	• Acquiring a general view on system-level design • Conveying the principles and techniques required by designing efficient memory hierarchies • Learning the main design techniques for pipelined systems

8. Content

8.1 Lecture	Hours	Instruction methods
1. Introduction 1.1 Information: bits and context 1.2 Program translation 1.3 Compilation 1.4 Processors and instructions 1.5 Caches and memory hierarchy 1.6 Operating system and hardware	2	Blackboard presentation, PPT/slides and oral talk, supporting conversation, explanations and instantiations.
2. Machine-level representation of programs 2.1 Program encoding 2.2 Data formats 2.3 ALU operations 2.4 Control 2.5 Procedures	6	
3. Processor architecture 3.1 Logic design and hardware control 3.2 Sequential implementations 3.3 Pipelining	8	
4. Memory hierarchy 4.1 Storage 4.2 Caches 4.3 Cache-frindly codes 4.4 Virtual memory 4.5 Address translation 4.6 Memory mapping 4.7 Dynamic memory allocation	8	
5. System-level I/O 5.1 Opening and closing files 5.2 Reading and writing files 5.3 Sharing files 5.4 I/O redirection 5.5 Standard I/O	4	

References

1. R. Bryant, D. O'Hallaron. Computer Systems: A Programmer's Perspective 2nd Ed. Prentice Hall, 2010
2. D.Patterson, J.Hennessy. Computer Organization and Design, Fourth Edition: The Hardware/Software Interface (The Morgan Kaufmann Series in Computer Architecture and Design), 2011
3. J.Hennessy, D.Patterson. Computer Architecture. A Quantitative Approach (5th Edition), Morgan Kaufmann Publishers, 2011

8.2 Seminar/lab	Hours	Instruction methods
1. Design and simulation with the ModelSim suite, as well as the FPGA prototyping of a pipelined 4-bit CPU	4	Issue presentation, discussion, questions and answers, computer-based design and FPGA design.
2. Design and simulation with the ModelSim suite, as well as testing of a 2-level cache memory	8	
3. Design and simulation with the ModelSim suite, as well as testing of a virtual memory, including memory management unit and translation lookaside buffer	8	
4. Design and simulation with the ModelSim suite, as well as the FPGA Altera DE2 prototyping of a sequential 8-bit processor	8	

References
1. ***, Altera DE2 Development and Education Board. User Guide, Altera Corporation, 2006
2. Ankita Goel, Hamid Mahmoodi, Systems Design Flow using Altera's FPGA Development Board (DE2-115 T-Pad), SFSU - Embedded Systems Tutorial, 2012
3. Sabih H. Gerez. Algorithms for VLSI Design Automation. Wiley, 2005

9. Correlation between the course content and the requirements of the specialists in the field and the expectations of the main employers

- The knowledge related to system-level design is essential for efficiently building any computing system (be it hardware, software, or embedded)
- The major employers from the field of computer and information technology are frequently adopting solutions based on modern hardware-software platforms, which rely on the techniques described in this course.

10. Assessment

Activity type	10.1 Assessment criteria	10.2 Assessment methods	10.3 Weight in final mark
10.4 Lecture	Solving a performance evaluation problem	Written exam	20 %
	Solving a multi-core system typical design problem	Written exam	30 %
10.5 Seminar /labs	Solving the lab exercises	Computer demonstation, oral presentation	35 %
	Homework	Computer demonstation, oral presentation	10 %
	Lab class attendance	Keeping track of lab attendance	5 %
10.6 Minimal performance standards (minimal specific knowledge required for passing the exam, the means to assess mastering the specific knowledge)			
• The correct performance evaluation of: a given pipelined system and a cache memory. • Design and test of two average-level problems: one from the field of pipelining and the other from the field of memory hierarchy.			

11. International compatibility

1	Carnegie Mellon University	http://www.ece.cmu.edu/courses/items/18741.html
2	Duke University	http://people.ee.duke.edu/~adh39/courses/fall_2012/cs250
3	Stanford University	http://scpd.stanford.edu/search/publicCourseSearchDetails.do?method=load&courseId=11710

Date

Signature of the course instructor
Ș.l. dr. Ing. Mihai UDRESCU

Signatures of the academic staff for seminars/labs
dr. ing. Alexandru Topîrceanu

……..... ……

Date of approval in the Department

Signature of the Department Director
Prof. dr. Ing. Vladimir Ioan CREŢU

……..……..……