Testi del Syllabus

Resp. Did.	**GIORGI ROBERTO**	**Matricola: 005709**

Docente	**GIORGI ROBERTO, 9 CFU**

Anno offerta:	**2017/2018**
Insegnamento:	**109156D - HIGH PERFORMANCE COMPUTER ARCHITECTURE**
Corso di studio:	**IG005 - COMPUTER AND AUTOMATION ENGINEERING – INGEGNERIA INFORMATICA E DELL'AUTOMAZIONE**
Anno regolamento:	**2017**
CFU:	**9**
Settore:	**ING-INF/05**
Tipo Attività:	**B - Caratterizzante**
Anno corso:	**1**
Periodo:	**Primo Semestre**

Testi in italiano

Lingua insegnamento	ENGLISH
Contenuti	Introduction to parallel computing. Multicore and multithreaded processors. Scaling and Law of Amdhal. Branch Prediction. Speculative execution. Superscalar Architecture. Out-of-order execution. Introduction to VLIW architectures. Ordering of memory accesses. Multimedia/Vector Operations. Parallel programming. Programming for application-specific processors and graphics (GPU). Introduction to reconfigurable logic. Programming in distributed environments (OpenMP and MPI). Datacenters.
Testi di riferimento	* MAIN TEXT: M. Dubois, M. Annavaram, P. Stenstrom, "Parallel Computer Organization and Design", Cambridge University Press, 2012, ISBN: 978-0-521-88675-8 * J.L. Hennessy, D.A. Patterson, "Computer Architecture: A Quantitative Approach" 5th Edition, Morgan Kaufman/Elsevier, 2012, ISBN: 978-0-12-383872-8. * D. Culler, J.P. Singh, A. Gupta, "Parallel Computer Architecture: A Hardware/Software Approach", Morgan Kaufman/Elsevier, 1998, ISBN-10 1558603433. * M.J. Flynn, "Computer Architecture: Pipelined and Parallel Processor Design", Jones and Bartlett Publishers, Inc., 1995, ISBN 0867202041 * David B. Kirk and Wen-mei W. Hwu, "Programming Massively Parallel Processors: A Hands-on Approach", Morgan Kaufmann (February 5, 2010) ISBN 0123814723.
Obiettivi formativi	Knowledge of the techniques for supporting the parallelism in computer systems. Ability to implement parallel applications.

Prerequisiti	Computer Architecture. Programming Fundamentals.
Metodi didattici	The course includes a theoritical part and a practical part (about 50%). A large part of the hours will be carried out in laboroatory to experience the possibility of estimating the performance through simulators.
Altre informazioni	Both in the written and oral exams (but also in the projects) the student is mainly required to show his/her detailed understanding of the topic, at least at the level shown by the teacher during the lesson. It's greatly appreciated the capacity of reasoning on the problem, rather then a mechanical (pedant) description of the topic. In case of written exercise, we mainly look at the correctness of the solution (in terms of numbers) and a very short justification of the chosen way to carry out the exercise (lengthy general wording is completely unnecessary). In case of oral question, the topic is typically one of the concepts illustrated during the lesson. Elements that are required are, for instance: the proof of the concept/theorem, precise schematic of the system, detailed behavior and functioning, reasons why this solution is used in the real-world.
Modalità di verifica dell'apprendimento	Written examination followed by oral examination. Passing (mark non less then 18/30) the written mid_term+final_term permits to avoid the written exam for the first two regular exams of the academic year (the oral part must be performed in any case). Optionally, you can present a group project (max 3 people) which contributes to the final mark up to 8/30 to be added to the final mark obtained after written and oral exam, in case of a technical project. Instead of the project one can present a technical paper discussion which contributes up to 3/30.
Programma esteso	* Introduction, Pipeline review. * Dynamic Instruction Scheduling * Tomasulo: An Efficient Algorithm for Exploiting Multiple Arithmetic Units * Branch Prediction: speculation of branch condition and branch target. * Predictor types, Bimodal, BHSR, BHT, PHT, 2-level adaptive, GAg, PAg, PAs. Other predictors (gshare, gselect, tournament). * Introduction to Superscalar Processor: general scheme and Renaming. * Full-System Simulator for single-core and multi-core systems. * Superscalar execution example: Re-Order Buffer and Instruction Window. Case studies: MIPS, Alpha, AMD, Pentium. * Software methods to extract Instruction Level Parallelism. * Introduction to multiprocessor systems, Flynn's taxonomy, UMA, NUMA, COMA systems, programming models * Coherence Protocols: Write Update, Write Invalidate, Hybrid. Snoopy based protocols: the MESI and DRAGON protocols * Memory Consistency Models: Sequential Consistency and Relaxed Consistency * Introduction to parallel programming. * Parallelsim and Performance * CUDA Architecture and Programming Elements * Introduction to OpenMP, Cilk, OpenMPI, Dataflow programming models. * Advanced topics: reconfigurable computing; datacenters.

Testi in inglese

	ENGLISH
	Introduction to parallel computing. Multicore and multithreaded processors. Scaling and Law of Amdhal. Branch Prediction. Speculative execution. Superscalar Architecture. Out-of-order execution. Introduction to VLIW architectures. Ordering of memory accesses. Multimedia/Vector Operations. Parallel programming. Programming for application-specific

	processors and graphics (GPU). Introduction to reconfigurable logic. Programming in distributed environments (OpenMP and MPI). Datacenters.
	* MAIN TEXT: M. Dubois, M. Annavaram, P. Stenstrom, "Parallel Computer Organization and Design", Cambridge University Press, 2012, ISBN: 978-0-521-88675-8 * J.L. Hennessy, D.A. Patterson, "Computer Architecture: A Quantitative Approach" 5th Edition, Morgan Kaufman/Elsevier, 2012, ISBN: 978-0-12-383872-8. * D. Culler, J.P. Singh, A. Gupta, "Parallel Computer Architecture: A Hardware/Software Approach", Morgan Kaufman/Elsevier, 1998, ISBN-10 1558603433. * M.J. Flynn, "Computer Architecture: Pipelined and Parallel Processor Design", Jones and Bartlett Publishers, Inc., 1995, ISBN 0867202041 * David B. Kirk and Wen-mei W. Hwu, "Programming Massively Parallel Processors: A Hands-on Approach", Morgan Kaufmann (February 5, 2010) ISBN 0123814723.
	Knowledge of the techniques for supporting the parallelism in computer systems. Ability to implement parallel applications.
	Computer Architecture. Programming Fundamentals.
	The course includes a theoritical part and a practical part (about 50%). A large part of the hours will be carried out in laboroatory to experience the possibility of estimating the performance through simulators.
	Both in the written and oral exams (but also in the projects) the student is mainly required to show his/her detailed understanding of the topic, at least at the level shown by the teacher during the lesson. It's greatly appreciated the capacity of reasoning on the problem, rather then a mechanical (pedant) description of the topic. In case of written exercise, we mainly look at the correctness of the solution (in terms of numbers) and a very short justification of the chosen way to carry out the exercise (lengthy general wording is completely unnecessary). In case of oral question, the topic is typically one of the concepts illustrated during the lesson. Elements that are required are, for instance: the proof of the concept/theorem, precise schematic of the system, detailed behavior and functioning, reasons why this solution is used in the real-world.
	Written examination followed by oral examination. Passing (mark non less then 18/30) the written mid_term+final_term permits to avoid the written exam for the first two regular exams of the academic year (the oral part must be performed in any case). Optionally, you can present a group project (max 3 people) which contributes to the final mark up to 8/30 to be added to the final mark obtained after written and oral exam, in case of a technical project. Instead of the project one can present a technical paper discussion which contributes up to 3/30.
	* Introduction, Pipeline review. * Dynamic Instruction Scheduling * Tomasulo: An Efficient Algorithm for Exploiting Multiple Arithmetic Units * Branch Prediction: speculation of branch condition and branch target. * Predictor types, Bimodal, BHSR, BHT, PHT, 2-level adaptive, GAg, PAg, PAs. Other predictors (gshare, gselect, tournament). * Introduction to Superscalar Processor: general scheme and Renaming. * Full-System Simulator for single-core and multi-core systems. * Superscalar execution example: Re-Order Buffer and Instruction Window. Case studies: MIPS, Alpha, AMD, Pentium. * Software methods to extract Instruction Level Parallelism.

* Introduction to multiprocessor systems, Flynn's taxonomy, UMA, NUMA, COMA systems, programming models
* Coherence Protocols: Write Update, Write Invalidate, Hybrid. Snoopy based protocols: the MESI and DRAGON protocols
* Memory Consistency Models: Sequential Consistency and Relaxed Consistency
* Introduction to parallel programming.
* Parallelsim and Performance
* CUDA Architecture and Programming Elements
* Introduction to OpenMP, Cilk, OpenMPI, Dataflow programming models.
* Advanced topics: reconfigurable computing; datacenters.