DEGREE CURRICULUM

COMPUTER ARCHITECTURE

Coordination: Concepció Roig Mateu

Academic year 2015-16

COMPUTER ARCHITECTURE 2015-16

Subject's general information

Subject name	Computer Architecture
Code	102014
Semester	4
Typology	compulsory
ECTS credits	6
Groups	1 big group (GG) theoretical part, 3 medium groups (GM) problems/practices.
Theoretical credits	3
Practical credits	3
Coordination	Concepció Roig Mateu
Office and hour of attention	Monday and Thursday from 10 to 11 h. Room 3.13 EPS
Department	Informàtica i Enginyeria Industrial
Teaching load distribution between lectures and independent student work	Globally the subject has 60 hours of presential classes and 120 hours of individual working of students.
Modality	Presencial
Important information on data processing	Consult this link for more information.
Language	Catalan
Degree	Degree in Computer Enginneering
Distribution of credits	3 credits of big group (GG) for the theoretical part, 3 credits of medium group (GM) for problems/practices.
Office and hour of attention	Monday and Thursday from 10 to 11 h. Room 3.13 EPS
E-mail addresses	roig@diei.udl.cat

COMPUTER ARCHITECTURE 2015-16

CONCEPCIÓ ROIG MATEU

Subject's extra information

This subject is held during the second semester of the second course of the degree.

This is a compulsary subject.

To follow up the subject they are required to have the knowledge of functional units composing the computer system, that are studied in the previous subjects of Computer Organization I and II.

Learning objectives

- Studying the global operation and the levels of the memory hierarchie in the computer.
- Learning the organization of the information in the memory system in order to have efficiency in the acces.
- Learning the pipeline mechanism to execute instructions inside the processor and evaluate its performance.
- Understanding the steps involved in complex operations solved in the arithmetic unit.
- Analyzing different solutions in terms of efficiency and cost. Being able to find which design solutions provide the best tradeoff between cost and performance.

Competences

Degree-specific competences

- GII-FB5: Knowledge of the structure, organization, workings and inter-connexion of computer systems, the basis of their programming, and their applications in the resolution of engineering problems.

- GII-CRI9: Ability to know, understand and evaluate computer structures and architecture, as well as the basic components which constitute them.

Degree-transversal competences

- EPS5: Ability for abstraction and critical, logical and logical reasoning.

Subject contents

1. Memory hierarchie

1.1. Introduction

General concepts

Principle of locality

1.2. *Cache memory*

Cache memory configurations

Mapping and identification of blocs

Bloc replacing algorithms

Writting policies

Data consistency

Cache performance

1.3. Main memory

Organization for improving performace

Alternative technologies

1.4. Virtual memory

Elements of virtual memory

Page table

TLB (Transaction Look-aside Buffer).

2. Pipeline processing

2.1. Basic concepts

2.2. Hazard management

Estructural hazards

Data hazards

Control hazards

2.3. Influency of instruciton set

2.4. Superescalar execution

3. Arithmetic processing

3.1. Adder circuits.

Half-adder, full-adder, parallel adder.

Carry-look-ahead.

adder/substractor circuit.

3.2. Binary multiplication algorithms.

3.3. Binary division algorithms

3.4. Floating point arithmetic

Floating point format

Approximate representation: rank and precision

Add and substract operations

Multiplication and division operations

Methodology

Classes are divided in different groups, big group (GG), where they attend all the students of the subject and medium group (GM) where there only assist part of the students. The contents of the different kind of groups are divided in the following way:

GG: They are expositive classes where they are shown the main contents on the subject.

GM: they are classes to solve exercices related to the contents exposed in the GG classes, in a participative and interactive way. They also carry out lab practices of memory hierachy with the simulator SMPcaché and of pipeline execuiton with the simulator WinMIPS64.

Development plan

Week 1: Memory hierarchie. 2 h. theory + 2 h. problems.

Week 2: Memory hierarchie. 2 h. theory + 2 h. problems.

Week 3: Memory hierarchie. 2 h. theory + 2 h. problems.

Week 4: Memory hierarchie. 2 h. theory + 2 h. problems.

Week 5: Memory hierarchie. 2 h. theory + 2 h. practices of laboratory.

Week 6: Memory hierarchie. 2 h. theory + 2 h. practices of laboratory.

Week 7: Pipeline processing. 2 h. theory + 2 h. problems

Week 8: Pipeline processing 2 h. theory + 2 h. problems

Week 9. Partial evaluation activities.

Week 10: Pipeline processing. 2 h. theory + 2 h. practices of laboratory.

Week 11: Pipeline processing. 2 h. theory + 2 h. practices of laboratory.

Week 12: Arithmetic processing. 2 h. theory + 2 h. problems.

Week 13: Arithmetic processing. 2 h. theory + 2 h. problems.

Week 14: Arithmetic processing. 2 h. theory + 2 h. problems.

Week 15: Arithmetic processing. 2 h. theory + 2 h. problems.

Weeks 16 and 17. Partial evaluation activities.

Week 18. Seminars.

Week 19. Recuperation evaluation activities.

Evaluation

N_P1: mark first partial exam

N_P2: mark second partial exam

N_Pr: practices mark

The qualification of the subject is calculated by applying the following expression:

FINAL_MARK = 30% N_P1 + 50% N_p2 + 20% N_Pr

To pass the subject, it is necessary that FINAL_MARK is greater than or equal to 5.

In the case of not passing the subject there is the option of recuperate it through a final exam. In this case, the mark is calculated as following:

N_rec: retest mark

FINAL_MARK = 80% N_rec + 20% N_Pr

Bibliography

Stallings W., *Organización y arquitectura de computadores*. (7 edición) Prentice-Hall.

Hamacher C., Vranesic Z., Zaky S. *Organización de computadores* (5ªedición). McGraw-Hill.

Ortega J., Anguita M., Prieto A. *Arquitectura de computadores.* Thomson.

Hennessy J. L., Patterson D. A. *Computer Architecture. A Quantitative Approach*. Morgan Kaufmann.