Fundamentals of computer design

J. Daniel García Sánchez (coordinator)
David Expósito Singh
Javier García Blas

Computer Architecture
ARCOS Group
Computer Science and Engineering Department
University Carlos III of Madrid

1. Course goals

This is the first module in the course *Computer Architecture*. The module intends to provide a general overview of the whole course. We will start reviewing the main goal of the course.

> **Goal**: guide the student to know basic concepts about the **architecture of a computer** and the impact that those concepts have on **applications performance** and computer systems.

To achieve this goal, during the course, we will focus on the acquisition of three skills:

1. Ability to know, understand, and evaluate **computer architectures**, as well as their **basic components**.
2. Knowledge and application of the fundamental principles and basic techniques of **parallel and concurrent programming**.
3. Ability to analyze and evaluate **computer architectures**, including **parallel platforms**, as well as to develop and **optimize software** for those platforms.

To successfully follow this course, students are expected to have followed previously at least courses in the following subjects:

- Computer structure.
- Operating systems.
- Programming.

With this approach the course is structured in six modules:

1. Fundamentals of computer design.
2. Performance evaluation of computer systems.

3. Instruction level parallelism.
4. Memory hierarchy.
5. Introduction to multiprocessors.
6. Parallel and concurrent programming models.

2. Module structure

This introductory module is structured in a single lesson that presents the course and offers a general overview of computer design.

The lesson has the following general structure:

1. Introduction.
2. History perspective.
3. Classification of computer systems.
4. Parallelism.
5. Computer architecture.

3. Lesson contents

3.1. Introduction

The term computer architecture has multiple definitions. In the course material, two possible definitions for the term are provided. Please, note both definitions are complementary. Besides, the knowledge of the computer architecture is essential to understand the trends in the evolution of computers as well as their limitations.

A fundamental aspect in the past evolution of processors is dominated by the well-known *Moore's Law*. That law defines the evolution of the number of transistors per chip over time. Although there have been times in which this translated into processor performance increases, this has not always been the case. In fact, since 2005, while *Moore's Law* has been still valid we have observed that clock frequency has stabilized. To gain more insight in this aspec,t we recommend that you read carefully the article *"The free lunch is over"* provided as complementary reading.

In the historic evolution of processors, there has been two relevant points. The first happened in the 80s with the emergence of the RISC technology. After that, we observed a sustained yearly increase in processor performance of around 52 %. The second one happened around 2005, as those increases could not be sustained any longer and consequently the industry adopted the multi-core technology.

3.2. Historic perspective

From the historic point of view, three important milestones can be identified and classified as revolutions.

The first one was the introduction of the microprocessor in the early 70s, which allowed to integrate enough transistors into a single chip so that all needed components for a 16 bits processor could be included in a single chip. This lead to the emergence of new market segments.

The second revolution was the incorporation of *instruction level parallelism* (ILP) that allowed to increase performance hiding details to programmers so that he could have essentially the same sequential view on the execution model.

Finally, the third revolution was the introduction of explicit support for data parallelism and the support for multiple threads of execution. This model makes parallelism visible to programmers. This introduction was caused by the diminishing increase in performance provided by instruction level parallelism.

As architectonic trends three models of parallelism can be identified:

- **DLP** (*Data Level Parallelism*): Allows to apply the same operation to multiple data in parallel.
- **TLP** (*Thread Level Parallelism*): Allows to run multiple threads simultaneously.
- **RLP** (*Request Level Parallelism*): Allows to process multiple requests at the same time.

3.3. Computers classification

In the materials, you will find a classification of computers in different categories: personal mobile devices, desktop computers, servers, clusters, and embedded computers. It is important to note that each of these categories exposes different characteristics as well as variations in costs.

3.4. Parallelism

In the materials we perform a theoretical estimation of the characteristic that would be needed to build a machine capable to process 1 Teraflop and store 1 Terabye of data. The conclusion is that it would need 3 Angstroms per byte, which makes this sequential approach unfeasible. However, this performance can be achieved with parallel approaches.

From the applications point of view, we can distinguish two kinds of parallelism: data parallelism and task parallelism. This is a property from applications. From the point of view of hardware, we can distinguish four approaches for parallelism: ILP, vector architectures, TLP, and RLP. This is a property from hardware. However, it is true that there are machines combining several of those models.

A traditional classification of the possible types of parallel architectures is the one given by Flynn (1966). This classification is carried out attending to two main criteria: the number of simultaneous instruction flows and the number of simultaneous data stream flows. Those two criteria lead to four categories SISD, SIMD, MISD, and MIMD. However, the MISD class is theoretical and there is no known industrial application of it. On the other hand, the MIMD class leads to two subclasses depending whether the architecture is highly coupled (TLP) or weakly coupled (RLP).

3.5. Computer architecture

There are different views on the computer architecture. Each one of those views focuses in some important attribute of the architecture. Please, follow the materials to identify the design space in each of those dimensions and the decisions that have been taken for some of the most well known architectures.