Map of Computer Science (CAO)

Artiom Leliuhin

May 2024

1 Introduction

Computer Architecture and Organization is the foundation of the field of Computer Science. Its origins lay back to the mid of 20th century when resources were tens of thousand times more modest in comparison to nowadays.

One of the pioneers of the Computer Science domain is John von Neumann, who is best known known for his work in the early development of computers. John von Neumann introduced an architecture design in 1945, see *von Neumann's architecture*, that has been applied since then in almost every electronic device. On a large scale, its ability to treat instructions as data is what makes assemblers, compilers, linkers, loaders, and other automated programming tools possible. The architecture allowed "programs that write programs" along with studies of Formal Languages and Automata Theory. This has made a sophisticated computing ecosystem flourish around von Neumann architecture machines.

This design back then was limited in memory management and instruction sets for performing tasks for general purposes and required different machines for various kind of problems in computation. The architecture has experienced a huge growth leaving behind only 5 main concepts described below.

To reduce cost and size of early fragile and expensive switches used in early computation era, such as vacuum tubes, as well as to increase reliability and speed, a radical new electronic switch was needed. In 1947 the group of scientists invented a transistor [1], that was a turning point in computational organization. *Moore's law* demonstrated how the number of tiny devices called *transistors* doubles every two years. Nowadays, current CPUs designs comprise billions and trillions of transistors that work at near light speed, being extremely fast and energy-efficient.

2 Main Problems and Activities

To continue to deliver performance improvements, industry has turned to largely evolutionary designs. With computer architecture design being so central to IT's success, it is tremendously important to envision future needs and trends. Rather, society's computing needs call for us to create new computer architectures designed from the ground up to support the diverse and challenging requirements of current and future computer systems. These changes include:

- First class support for security, programmability and reliability.
- New mechanisms enhancing performance and scalability.
- Reducing cost of energy.
- Predictable and efficient behavioral principles of newly designed hardware.

2.1 Examples

One of the widely-studied parts of computer engineering is designing energy-efficient and productive CPUs and GPUs, working at multicore levels. Some of the current projects like Apple's M-series chips involve heterogeneous computing principles, that allows to combine different types of cores (CPUs and GPUs) within one small chip. This approach allows for better optimization of performance and power efficiency for specific workloads.

Another good example of current design trends are life-support devices or spectrum analyzing machines for medical discoveries and healthcare. IBM Summit Supercomputer for COVID-19 Research is specifically designed for high-performance computing, featuring over 27,000 NVIDIA GPUs and IBM Power9 CPUs interconnected with high-speed links. The computer architecture of Summit enables it to perform complex simulations and data analysis tasks required for medical research.

Tesla, an electric vehicle manufacturer, has developed its own custom-designed Full Self-Driving (FSD) computer, which is at the heart of its autonomous driving technology. Designed with automotive-grade reliability and energy efficiency in mind, the Tesla FSD computer integrates multiple processing cores, hardware accelerators for AI inference, and specialized I/O interfaces for sensor data fusion.

2.2 Dependencies and Connections

- **Programming Languages:** CAO designs influence how programming languages like Python or C++ are optimized to run efficiently on hardware. Conversely, language features and demands, such as parallelism or memory management, may shape hardware designs to accommodate specific language requirements.

- **Bioinformatics:** CAO enables the rapid processing of vast biological datasets, crucial in fields like genomics. Insights from bioinformatics, such as algorithms for DNA sequencing, inform the development of specialized hardware tailored to handle these computational tasks efficiently.

- **Algorithms:** CAO impacts the performance of algorithms by providing hardware support for key operations. For instance, hardware accelerators for matrix multiplication benefit machine learning algorithms. Similarly, algorithmic innovations, like new sorting techniques, may drive the need for hardware optimizations to execute these algorithms more efficiently.

- **Operating Systems and Networks:** CAO provides the foundation for operating systems and network infrastructure. Hardware features like memory management units and network interface controllers are essential for OS and network functionality. Conversely, the demands of operating systems and network protocols may influence hardware designs to prioritize features such as low-latency communication or virtualization support.

- **Software Engineering:** CAO influences software design by providing the underlying hardware platform. Software engineers optimize code to take advantage of hardware features like caching or vectorization. Conversely, software engineering practices, such as modularization or abstraction, may influence hardware designs to support efficient software execution.
- **Databases:** CAO supports the efficient processing of database queries by providing hardware features like optimized memory access or parallel processing capabilities. Database requirements, such as fast disk I/O or efficient indexing, may drive hardware designs to prioritize these features.
- **AI and Robotics:** CAO provides the computational horsepower needed for AI tasks like neural network training or robotic control. Advances in AI and robotics drive the demand for specialized hardware accelerators optimized for these tasks, such as GPUs for deep learning or sensor fusion.
- **Graphics:** CAO powers graphics processing units (GPUs) used in rendering realistic graphics and visual effects. Demands from graphics-intensive applications, such as gaming or 3D modeling, may drive hardware designs to prioritize features like shader performance or texture compression.
- **Human-Computer Interaction:** CAO enables responsive user interfaces by providing hardware support for input/output devices and display rendering. Human-computer interaction research drives demands for hardware features that enhance user experience, such as touch sensitivity or gesture recognition.
- **Computational Science:** CAO supports scientific computing tasks, such as simulations or data analysis, by providing high-performance computing resources. Demands from computational science, such as large-scale parallelism or fast floating-point calculations, may influence hardware designs to prioritize these capabilities.
- **Organizational Informatics:** CAO provides the hardware infrastructure for organizational systems, such as servers or data centers. Requirements from organizational informatics, such as reliability or scalability, may influence hardware designs to meet the needs of enterprise-level computing.

3 Important notes

3.1 Jonh von Neumann's Architecture

The von Neumann architecture—also known as the von Neumann model or Princeton architecture—is a computer architecture based on a 1945 description by John von Neumann, and by others, in the First Draft of a Report on the EDVAC. The document describes a design architecture for an electronic digital computer with these components:

- A processing unit with both an arithmetic logic unit and processor registers
- A control unit that includes an instruction register and a program counter
- Memory that stores data and instructions
- Input and output mechanisms

3.2 Moore's law

The observation that the number of transistors on computer chips doubles approximately every two years is known as Moore's Law. Moore's Law is not a law of nature, but an observation of a long-term trend in how technology is changing. The law was first described by Gordon E. Moore, the co-founder of Intel, in 1965.1

3.3 Point-Contact Transistors

Named the "transistor" by electrical engineer John Pierce, Bell Labs publicly announced the revolutionary solid-state device at a press conference in New York on June 30, 1948. A spokesman claimed that "it may have far-reaching significance in electronics and electrical communication." Despite its delicate mechanical construction, many thousands of units were produced in a metal cartridge package as the Bell Labs "Type A" transistor.

4 Important venues

4.1 Conferences

- **International Symposium on Computer Architecture (ISCA):** This is a premier conference for presenting and discussing the latest research on computer architecture. It covers a wide range of topics, including processor design, memory systems, storage systems, and networking. (link)

- **IEEE/ACM International Symposium on Microarchitecture:** This conference focuses on the design and implementation of microprocessors and other computing systems. It covers topics such as instruction set architecture, pipeline design, cache design, and power management. (link)
- **IEEE/ACM International Symposium on Microarchitecture HPCA:** The International Symposium on High-Performance Computer Architecture:This conference covers research on high-performance computer architectures, including vector processors, multiprocessors, and GPUs. (link)

4.2 Journals

- **IEEE Transactions on Computers:** This journal publishes a broad range of research on computer science and engineering, including computer architecture. (link)
- **The Journal of Supercomputing:** This journal focuses on research on high-performance computing systems, including computer architecture. (link)
- **Microprocessor Report:** This is a monthly magazine that covers the latest news and trends in computer architecture. (link)

5 Local Studies in WUT

Currently, West University of Timisoara features several related courses: CA (Computer Architecture), FLAT (Formal Languages and Automata Theory) and many more related to the field disciplines, see *Curriculum WUT*. The subjects examined in the university deeply cover many logical and practical subtleties, allowing to graduate with high-level skills and competencies. As a 1-st year student who finished the CA course, discussed theoretical material along with practical labs (CTF challenges) introduced me and my colleagues to this versatile field, imposing main concepts and nowadays design features. It gave a good grasp for a newcomer student, that is eager to work in embedded systems domain.

6 References

1. https://www.computerhistory.org/siliconengine/invention-of-the-point-contact-transistor/

2. https://www.bbc.co.uk/bitesize/guides/zhppfcw/revision/3

3. https://ourworldindata.org/moores-law