COMPUTER ARCHITECTURE CORE OF KNOWLEDGE FOR COMPUTER SCIENCE STUDIES

Mile Stojcev, Ivan Milentijevic, Dimitris Kehagias,
Rolf Drechsler, Marjan Gusev

Abstract: This paper describes the Computer Architecture (CA) core of knowledge for Computer Science (CS) students. The core is composed of three levels. Core levels provide balanced knowledge of both hardware and software component of computer system. The core incorporates almost all new aspects in CA. The creation of the core is based on the experience of the lecturers from different academic institutions and represents a joint effort on identifying of main topics in the field of CA.

1. Introduction

The demands of 21st century CS engineers are as follows:

- versatility – ability to adopt the changing demands;
- strong fundamental knowledge in Computer Architecture, programming, operating systems, software engineering, networking and communication;
- hands-on design and laboratory experience.

Computer Architecture is concerned with the structure and behavior of digital computers. It has developed into a discipline for the design and evaluation of computers largely in response to the proliferation for these machines over the last couple decades [1]. Recently the CA field has been extended to include pipeline parallel computer architecture, cache design, multi-computers and multiprocessors, networking and communication, digital signal processing, etc. In order to perform these tasks the CS engineer must possess a wide perspective of the CA world and a system wise insight into computer science needs and capabilities.

This paper describes the joint effort of lecturers in CA field from Faculty of Electronic Engineering - University of Nis, Department of Informatics - Technological Educational Institution of Athens, Institute of Computer Science - University of Bremen and Faculty of Natural Sciences and Mathematics - University of St. Cyril and Methodius Skopje aimed to the identifying of Computer Architecture core of knowledge for Computer Science studies. Working on the Joint European Project TEMPUS CD-JEP-16160/2001 that is entitled "Innovation in Computer Science Curricula in Higher Education" we have noticed that we teach our students in CA to almost the same main topics through the different number of courses at each participating academics institution. We are still not able to adopt the same courses but we have identified the topics that comprise courses. This is the reason why we are suggesting the three-level organization of CA knowledge rather then rigidly defining the courses.

2. Three-level Computer Architecture Core of Knowledge

One of the first problems all computer architecture instructors have is to identify the course contents. The topics have not been standardized in the same fashion as, say, a mathematics or physics courses with plenty of textbooks. Due to architecture innovations and rapid growth of VLSI IC design, during the last two decades creation of comprehensive core of knowledge in Computer Architecture becomes a very difficult task. Several textbooks that covered the core are [1-9]. Our goal as authors of this paper is to summarize the teaching

experience in applying many modern concepts that are involved and explained in selected textbooks. The first step towards developing of CA core of knowledge was to verify that the proposed content of core is indeed essential from the standpoint of industrial sector and various government agencies. We try to accomplish the goal by a balanced combination of CA topics of wider interest, such as: definition of Instruction Set Architecture (ISA), evaluation of computer performance, RISC and CISC concepts, ILP processors, multiprocessor systems and embedded systems. Mentioned main topics we organize in three level CA core (Fig.1). The core is suitable for implementation in to three or four year computer science curricula.

Fig. 1. Three-level CA core organization

a) Introductory Level

The traditional way of introducing CA is concentrate on the following issues: identifying of basic building blocks of computer system, introduction of Instruction Set Architecture (ISA) and description the principle of operation of computer components. Accordingly, for the introductory level we propose the following content:

a1) Introduction to Computer Systems - Basic building blocks: Central Processing Unit (CPU), memory, input-output (I/O).

a2) Data Representation - Fixed-point numbers: binary, hexadecimal and decimal bases. Signed numbers. Floating-point numbers. Character codes: ASCII, EBCDIC, and Unicode.

a3) Digital Logic - Combinational logic circuit. Combinational logic design. Sequential circuits: Latches, flip-flops, Mealy and Moore models, registers and counters. Memory and programmable logic devices. Arithmetic circuits. Tristate devices and buses.

a4) ISA - Hardware components of the ISA. Structure of instructions. Addressing modes. Conditional operations. Stack and subroutine. Macros. Assembly language programming. Languages and machine: compilation process, Assembly process, linking and loading.

a5) Component of Computer systems - Memory: hierarchy, RAM, ROM, chip organization, memory modules, cache memory, and virtual memory. Input/Output: Simple bus architecture, I/O principles, handshaking, buffering, programmed I/O, interrupt driven I/O, direct-memory access. Computer peripherals: mass storage, input devices, output devices.

b) Intermediate level

Provides a core curriculum and performs three important functions. It takes an in-depth look at a modern processor and fills in the detail omitted in the more general first course. It covers topics that emphasize the hardware-software interface, such as architectural support for

operating systems. Finally, it provides a foundation for an advanced course in architecture. The intermediate level contains the following topics:

b1) Performance of computer systems - Measuring performance. Metrics. Benchmark programs. Comparing performance.
b2) Instruction set architecture - a case study - Architecture concepts, Operation cycle. Register set. Operand addressing. Addressing modes. Instruction set. Data transfer instructions. Stack instructions. Data manipulation instructions, arithmetic instructions, logical and bit manipulations, shift instructions, floating point computation, conditional branch instructions, program control instructions, procedure call and return instructions, types of interrupts and processing external interrupts.
b3) Processor: Datapath and Control - Complex Instruction Set Computer (CISC): Instruction set architecture. Data path organization, microprogrammed and hardwired control organization. Reduced instruction set computer (RISC): Instruction set architecture, addressing modes, datapath organization, and control organization. CISC vs. RISC.
b4) Pipelining - Overview of pipelining. Pipelined datapath. Pipelined control. Data hazards and forwarding. Data hazards and stalls. Branch hazards. Exceptions. Introduction to super scalar design. Instruction level parallelism. Comparative architectures. Low-level support for high-level languages.
b5) Cache memory - Cache organization and placement policies. Read policies and performance models. Instructions buffering schemes. Data caches. Unified caches. Split and multilevel caches.
b6) Architectural support for operating systems - Memory management. Multitasking and interrupts. Protection mechanisms.
b7) I/O Subsystem - Bus structures: Interconnection structures, Bus interconnection. Simple bus architecture. Bridge based bus architectures. Multiple-bus hierarchies. Storage technology: Mass storage - magnetic disks, magnetic tapes, and optical disks.
b8) Multiprocessing and network - Introduction to networks. Parallel processing concepts, Single Instruction Multiple Data (SIMD) / Multiple Instruction Multiple Data (MIMD).

c) Advanced level

Provides an in-depth treatment of issues in modern computer architecture. This level emphasizes method for accelerating processor and system performance. It explores parallel processing and network computing in greater depth and includes advances in audiovisual techniques in multimedia systems. The advanced level should cover the following topics:

c1) Computational models and concept of computer architecture - Parallel processing.
c2) Enhancing performance – Instruction Level Parallelism (ILP), ILP processors - Superscalar processors, superpipelined processors, superpipeline-superscalar processors. Very Long Instruction Word (VLIW) architectures. Branch prediction. Prefetching. Speculative execution. Multithreading. Audi-video support for multimedia. Redundant Array of Independent Discs (RAID) architectures. Scalability.
c3) Instruction level data-parallel architectures - Data-parallel architecture. SIMD architectures. Associative and neural architectures. Data-parallel pipelined and systolic architectures.

c4) Thread and process level parallel architectures - MIMD architectures. Multi-threaded architectures. Distributed memory MIMD architectures. Shared memory MIMD architectures. Hypercube, butterfly, shuffle-exchange, crossbar topologies. Cache coherence protocols. Memory models and memory consistency.
c5) Network and distributed systems - Introduction to Local Area Networks (LANs) and Wide Area Networks (WANs). Layered protocol design, ISO/OSI, TCP/IP. Impact of architectural issues on distributed algorithms. Network computing. Distributed multimedia systems.
c6) Configurable computing - System based on reconfigurable hardware. Systems based on evolvable hardware.
c7) Embedded Computing Systems - Embedded computing. Processor for embedding system. Embedded computing platforms. Program design and analysis. Processes and operating systems. Hardware accelerators. Networks. System design techniques. Design examples. Configurable computing.

3. Conclusion

Three-level CA core of knowledge for CS students is presented in this paper. The main concern in creating such a core was to concentrate on increased industry demands for designing and applying of computer systems, as well as to accept the impact of rapid development of VLSI integrated circuits on computer architectures. Our goal was to create a balance between basics of CA (introductory level), uptodate concepts (intermediate level) and challenges of new trends (advanced level).

4. References

1. Hayes, J. P.: Computer Architecture and Organization, third edition, Mc Graw-Hill Book Company, New York, 2000.
2. Stallings, W: Computer Organization and Architecture: Principle of Structure and Function, fifth edition, MacMillan Publishing Company, 2002.
3. Patterson D., Hennessy J.: Computer Organization and Design: The Hardware/Software Interface, second edition, Morgan Kaufman Publisher's, San Mateo, 1997.
4. Carpinelli J.D.: Computer Systems Organization & Architecture, first edition, Addison Wesley Longman, Boston, 2001.
5. Hennessy J., Patterson D.: Computer Architecture: a quantitative approach, second edition, Morgan Kaufmann Publishers, Inc, San Francisco, 1997.
6. Flynn, M. J.: Computer Architecture: Pipelined and parallel processor design, Jones & Bartlett, 1995.
7. Sima, Fountain, Kacsuk: Advanced Computer Architecture, a design space approach, Addison Wesley, 1997.
8. Gusev, M.: Contemporary Computer Systems, Medis Informatika, 1998.
9. Stojcev, M.: RISC, CISC and DSP processors, Faculty of Electronic Engineering, Nis, 1997.

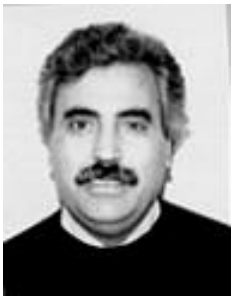
Mile Stojcev

Stojcev M. received the B.S., M.S., and Ph.D. in electrical engineering in 1970, 1977 and 1982, respectively, from the Faculty of Electronic Engineering, Nis, Serbia. Hi is currently a full professor in the Department of Electronics At the Faculty of Electronic Engineering, Nis. His research interests include computer architecture, VLSI IC design and embedded systems. Hi is a coordinator of Tempus project CD-JEP 16160/2001.

Ivan Milentijevic

Milentijevic I. received the B.S., M.S., and Ph.D. in computer science in 1989, 1994 and 1998, respectively, from the Faculty of Electronic Engineering, Nis, Serbia. Hi is currently an assistant professor at Computer Science Department at the Faculty of Electronic Engineering, Nis. His research interests include computer architecture, fast digital arithmetic and parallel processing.

Dimitris Kehagias



Kehagias D. received the B.Eng. and M.Eng. degrees in Electrical and Computer Engineering from Concordia University, Montreal Canada, in 1981 and 1983 respectively. He is an Associate Professor at the Department of Informatics, TEI of Athens, Greece. His research interests focus on computer architecture, multicomputer systems and applications of microprocessors. He is a member of the Technical Chamber of Greece (TEE).

Rolf Drechsler

Drechsler R. received his diploma and Ph.D. in computer science from the J.W. Goethe-University in Frankfurt am Main, Germany, in 1992 and 1995, respectively. He is a professor for computer architecture at the University of Bremen, Germany. His research interests include verification, logic synthesis, and evolutionary algorithms.

Marjan Gusev

Gusev **M.** is head of the Institute of Informatics at the Faculty of Natural Sciences and Mathematics, Sts. Cyril and Methodius University in Skopje, Macedonia, since 1999 and manager of Wireless Application Laboratory. He finished his PhD studies in Loughborough, UK and Ljubljana, Slovenia in the field of parallel processing. He has published many papers in the field of computer architecture, parallel processing, e-business and mobile applications.