Preface

Systems On-Chip designs have evolved over time from fairly simple unicore single memory designs to complex homogeneous/heterogeneous multicore SoC architectures consisting of a large number of IP (Intellectual Property) blocks on the same silicon. To meet the challenges arising from high computational demands posed by latest consumer electronic devices, most current systems are based on such paradigm, which represents a real revolution in many aspects of computing.

The attraction of multicore processing for power reduction is compelling. By splitting a set of tasks among multiple processor cores, the operating frequency necessary for each core can be reduced, thereby facilitating a reduction in the voltage on each core. Because dynamic power is proportional to the frequency and to the square of the voltage, we are able to obtain a sizable gain, even though we may have more (multiple?) cores running.

As more and more cores are integrated into these designs to share the ever increasing processing load, the primary challenges are geared toward efficient memory hierarchy, scalable system interconnect, new programming models, and efficient integration methodology for connecting such heterogeneous cores into a single system capable of leveraging their individual flexibility.

Current design methods are inclined toward mixed hardware/software (SW/HW) codesigns, targeting multicore SoCs for application specific domains. To decide on the lowest cost mix of cores, designers must iteratively map the devices functionality to a particular HW/SW partition and target architectures. In addition, to connect the heterogeneous cores, the architecture requires high performance-based complex communication architectures and efficient communication protocols, such as hierarchical bus, point-to-point connection, or the recent new interconnection paradigm—Network-on-Chip. Software development also becomes far more complex due to the difficulties in breaking a single processing task into multiple parts that could be processed separately and then reassembled later. This reflects the fact that certain processor jobs could not possibly be easily parallelized to run concurrently on multiple processing cores and that load balancing between processing cores especially heterogeneous cores is extremely difficult.

This second edition of this book stands independent and we have made every attempt to make each chapter self-contained as well. It is organized in 11 chapters. The first chapter introduces Multicore Systems On-Chip (MCSoCs) architectures

and explores SoCs technology and the challenges it presents to organizations and developers building next generation multicore SoCs based systems.

Understanding the technological landscape and design methods in some level of details is very important. This is because so many design decisions in multicore architecture today are guided by the impact of the technology. Chapter 2 presents design challenges and conventional design methods of MCSoCs. It also describes a so called scalable core-based method for systematic design environment of application specific heterogeneous multicore SoC architectures. The architecture design used in conventional methods of multicore SoCs and custom multiprocessor architectures are not flexible enough to meet the requirements of different application domains and not scalable enough to meet different computation needs and different complexities of various applications. Therefore, designers should be aware of existing design methods and also be ready to innovate or adapt appropriate design methods for individual target platform.

Understanding the software and hardware building blocks and the computation power of individual components in these complex MCSoCs is necessary for designing power, performance, and cost-efficient systems. Chapter 3 describes in details the architectures and functions of the main building blocks that are used to build such complex multicore SoCs. Students with relevant background in multicore SoC building blocks could effectively skip some of the materials mentioned in this chapter. The knowledge of these aspects is not an absolute requirement for understanding the rest of the book, but it does help novice students or beginners to get a glimpse of the big picture of a heterogeneous or homogeneous MCSoC organization.

Whether homogeneous, heterogeneous, or hybrid multicore SoCs, IP cores must be connected in a high-performance, scalable, and flexible manner. The emerging technology that targets such connections is called an on-chip interconnection network, also known as a network on chip (NoC), and the philosophy behind the emergence of such innovation has been summarized by William Dally at Stanford University as *route packets, not wires*. Chapters 4–6 investigate 2D-NoC, 3D-NoC, and 2D/3D NoC Network Interface (NI) designs. These chapters focus on the architecture and design of Network-on-Chip (NoC) and the NI. Efficient, lightweight NI interfaces are critical for overall latency reduction. For an effective concurrent multicore SoCs, a programmer needs a fast on-chip network transport, fast and easy-to-use network interfaces, and predictable network performance. These three chapters are all very important part of the book since they allow the reader to understand what needed microarchitecture for on-chip routers and network interfaces are essential toward meeting latency, area, and power constraints. Reader will also understand practical issues about what system architecture (topology, routing, flow control, NI) is most suited for these on-chip networks.

With the rise of multicore and many-core systems, concurrency becomes a major issue in the daily life of a programmer. Thus, compiler and software development tools will be critical toward helping programmers create high performance software. Programmers should make sure that their parallelized program codes would not cause race condition, memory access deadlocks, or other faults

that may crash their entire systems. Chapter 7 describes a novel parallelizing compiler design for high performance computing.

Power dissipation continues to be a primary design constraint and concern in single and multicore systems. Increasing power consumption not only results in increasing energy costs, but also results in high die temperatures that affect chip reliability, performance, and packaging cost. Chapter 8 provides a detailed investigation of power reduction techniques for multicore SoC at components and network levels. Energy conservation has been largely considered in the hardware design, in general and also in embedded multicore system components, such as CPUs, disks, displays, memories, and so on. Significant additional power savings could be also achieved by incorporating low power methods into the design of network protocols used for data communication (audio, video, etc.).

Soft-core processors are becoming increasingly common in modern multicore SoCs. A soft-core processor is a programmable processor that can be synthesized to a circuit, typically integrated into a larger multicore SoC. Chapter 9 describes architecture and design results of a low power Soft-core 32-bit QueueCore architecture. This core is an efficient architecture which can be easily programmed and integrated in a multicore SoC platform.

Chapter 10 introduces practical hardware design issues of a multi-mode processor architecture targeted for embedded applications. In an embodiment of this processor, a single instruction stream consists of two different programming models. This is effectively achieved dynamically with an execution-mode-switching and sources-results computing mechanisms.

Current and future generations of embedded biomedical applications require more flexible and cost-effective computing platforms to meet its rapidly growing market. The programmable embedded multicore SoC systems appear to be an attractive solution in terms of ease of programming, design cost, power, portability, and time-to-market. The first step toward such complex systems is to characterize biomedical applications on the target architecture. Such studies can help us understand the design issues and the trade-offs in specializing hardware and software systems. Chapter 11 ties together previous chapters and presents a real embedded multicore SoC system design targeted for biomedical applications (i.e., ECG processing). For this book, we used our experience to illustrate the complete design flow for a multicore SoC running an electrocardiogram (ECG) application in parallel. More specifically, discussions on how to design the algorithms, architecture, and register transfer level implementation for ECG processing; discussions of the FPGA prototype, and validation are described.

Acknowledgments

The second edition of this book took nearly 3 years to write. It evolved and is derived from our teaching experiences in embedded system designs and architecture to both undergraduate and graduate students. Multicore paradigm created

stupendous opportunities to increase overall system performance, but also created many design challenges that designers must now overcome. Thus, we must continue innovating new algorithms and techniques to solve these challenges. We must also continue with our efforts to better educate computer science and computer engineering students in both embedded multicore architectures and programming.

Abderazek Ben Abdallah

Author Biography

Abderazek Ben Abdallah is currently a Senior Associate Professor in the School of Computer Science and Engineering, the University of Aizu, Japan where he is engaged in advanced computer system research and education. Previously, he was a Research Associate, then an Assistant Professor at the University of Electro-Communications at Tokyo (2002–2007). He has been a regular visiting Professor at Huazhong University of Science and Technology (HUST), and Hong Kong University of Science and Technology (KUST) since 2010. His research interests lie primarily in systems, including embedded real-time systems, energy-efficient system design, on-chip interconnection networks, parallel systems, and innovative architectures. He participated in new architecture development and led several middle-scale VLSI development efforts on several projects. He received the 2010 national prize for outstanding research in the field of computer systems and information technology. He has published more than 100 peer-reviewed journal and conference papers, edited one book, wrote one book, and several book chapters in these areas. He is a member of IEEE, ACM, and IEICE.

http://www.springer.com/978-94-91216-91-6

Multicore Systems On-Chip: Practical Software/Hardware Design

Ben Abdallah, A.

2013, XXVI, 273 p. 196 illus., 79 illus. in color.,
Hardcover
ISBN: 978-94-91216-91-6
A product of Atlantis Press