Preface

Nowadays, the technology has become an essential pawn in our life that is not restricted anymore to academic research or critical missions; but it is moving away to provide the simplest and easiest services that we need or desire for our daily life. With the expanse of technology and the rising of new trends every day, the necessity to process information anywhere and anytime is becoming the main goal of developers and manufacturers.

Systems on chip (SoCs) are embedded systems composed of several modules (processors, memories, input/output peripherals, etc.) on a single chip. With SoCs, it is now possible to process information and execute critical tasks at higher speed and lower power on a tiny chip. This is due to the increasing number of transistors that can be embedded on a single chip, which keeps doubling approximately every 2 years as Intel co-founder Gordon Moore predicted in 1965. This made shrinking the chip size while maintaining high performance possible. This technology scaling has allowed SoCs to grow continuously in component count and complexity and evolve to systems with many processors embedded on a single SoC. With such high integration level available, the development of multi and many cores on a single die has become possible.

Historically, the SoCs paradigm has evolved from fairly simple unicore single memory designs to complex homogeneous/heterogeneous multicore SoC (MCSoC) systems consisting of a large number of intellectual property (IP) cores on the same silicon. To meet the challenges arising from high computational demands posed by latest state-of-the-art embedded and 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 in embedded and in general purpose computing. By splitting a set of tasks among multiple cores, the operating frequency necessary for each core can be reduced, thereby facilitating a reduction in the voltage on each core. As 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 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 (HW/SW) co-designs, targeting multicore SoCs for application specific domains. To decide on the lowest cost mix of cores, designers must iteratively map the device's 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 book is organized into nine chapters. The book stands independent and we have made every attempt to make each chapter self-contained as well.

Chapter 1 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 are 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. Readers with a 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 presents fundamental and advanced on-chip interconnection network technologies for multi- and many-core SoCs. 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 towards meeting latency, area, and power constraints. Reader will also understand practical issues about what system architecture (topology, routing, flow control, NI, and 3D integration) 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 towards 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's 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.).

Chapter 9 ties together previous chapters and presents a real embedded multicore SoC system design targeted for elderly health monitoring. For this book, we used our experience to illustrate the complete design flow for a multicore SoC running an electrocardiogram (ECG) application in parallel. Thanks to the recent technological advances in wireless networking, embedded microelectronics, and the Internet, computer and biomedical scientists are now capable to fundamentally modernize and change the way health care services are deployed. Discussions on how to design the algorithms, architecture, register-transfer level implementation, and FPGA prototyping and validation for ECG processing are presented in details.

This book took nearly 2 years to complete. It evolved from our first book and is derived from our teaching experiences in embedded system designs and architecture to both undergraduate and graduate students. Multicore systems 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.

The author is thankful to numerous colleagues and graduate students for their lively discussions and their help in preparing the manuscript of this book. Special thanks are due to the publishers in bringing out this book quickly, yet maintaining very high quality.

Aizu-Wakamatsu, Japan Abderazek Ben Abdallah

http://www.springer.com/978-981-10-6091-5

Advanced Multicore Systems-On-Chip
Architecture, On-Chip Network, Design
Ben Abdallah, A.
2017, XXVI, 273 p. 171 illus., 88 illus. in color.,
Hardcover
ISBN: 978-981-10-6091-5