

ARCHITECTURE OF SYSTEMS ON CHIP

Luca Benini,

Università di Bologna, Italy

Keywords: Systems-on-chip, CMOS Technology, International Technology Roadmap for Semiconductors, Digital Circuits, Multi-core, Low Power, Design Technology

Contents

1. Introduction
 2. Basic Concepts and Definitions
 3. Historical Perspectives
 4. Current Trends
 5. Design Automation
- Glossary
Bibliography
Biographical Sketches

Summary

In this chapter we have first provide basic concepts and definitions to understand the concept of system-on-chip and the related business and technology contexts. We then give a historical excursus of SoC evolution in the last two decades. We analyze the state of the art in SoC architectures, restricting our focus on Consumer Portable SoCs which are probably the largest volume and fastest evolving, due to the extremely dynamic and fiercely competitive market. We analyze performance requirements and architectural trends, highlighting the main challenges in architectural evolution and promising approaches to address them. Finally, we survey design technology, methods and tools needed to support the evolution of SoCs in the next decade.

1. Introduction

Silicon technology has entered the nanometer regime. At the time of writing (2011), CMOS technology with 32nm minimum feature size is in volume production, while 28nm is rapidly ramping up. Billions of elementary devices can be integrated on a single die: 32nm CMOS features an impressive $1.5\text{Mgate}/\text{mm}^2$ density. Integration of an entire digital system onto a single silicon die is indeed possible. On the other hand, the cost for developing a new technology generation and ramping it up to volume production is in the multiple billions range, and the non-recurring engineering cost for developing a new chip is in the tens of millions range. To recover this huge non-recurring cost, integrated device manufacturers (IDMs) need to focus on extremely high-volume products, namely, complex and flexible *systems-on-chips*. In this chapter we will provide an overview of the history, the current trends and the main challenges in system-on-chip architectures and design flows.

2. Basic Concepts and Definitions

A System-on-Chip (SoC) is a single-die device that contains all the necessary hardware and electronic circuitry for a complete system in a given application domain. This includes on-chip memory (volatile and non-volatile), one or more processors, peripheral interfaces, I/Os, data converters, and other components that comprise a complete computer system. The completeness requirement is often relaxed in practice: most SoCs do require other components to work properly, such as, for instance, power supply ICs and discrete devices, but the main computational functions and differentiating value-added features are hosted on the SoC.

The key rationale for building a SoC is not an abstract notion of full-system integration, but the push to achieve cost reduction by leveraging Moore's law, which provides exponentially lower cost-per-function over time. In other words, if it is technologically feasible to integrate multiple sub-systems onto a single silicon die, then the cost of the resulting SoC would be smaller than the sum of the costs of individual chips integrated on a board. It is important to note however that this assertion holds true only if the production volume of the SoC is sufficiently high to amortize the non-recurring costs associated to design and manufacturing. Such costs are exponentially increasing with technology scaling; hence SoC integration is a high-risk endeavor which requires careful risk analysis and cost estimation.

From an economic viewpoint [1], a SoC is a product class and design style that integrates technology and design elements into a wide range of high-complexity, high-value semiconductor products. Because of the no-recurring costs mentioned earlier, manufacturing and design technologies for SoC are typically developed only for high-volume market drivers. SoCs are the evolution of Application-Specific Integrated Circuits (ASICs). The term "ASIC" refers to a business model which decouples technology from fabrication, by defining a "handoff" process from the design team to the silicon foundry. In addition ASIC design is generally performed via a highly-automated design methodology, where the chip designer works predominantly at the functional level, specifying the design in a high-level hardware description language, and invoking automatic logic synthesis and place-and-route with a standard-cell methodology).

Similarly to ASIC, a SoC is an integrated circuit designed specifically for an application. Differently from traditional ASICs, which emphasized specialized-function hardware design, SoCs tend to maximize reuse of existing blocks or "cores"—i.e., minimize the amount of the chip that is newly or directly created. Reused blocks in SoC include analog and high-volume custom cores, as well as blocks of software technology. A key challenge is to develop, create and maintain reusable blocks, also called intellectual-property (IP) cores so that they are available for SOC integration. For economic reasons, custom functions are seldom created, as reducing design cost and design risk is paramount. In addition, reusable cores might require characterization attributes such as "field of use" or "assumed design context" that are not normally specified when designing a stand-alone component. Creation of an IC design artifact for reuse (IP-core) is substantially more difficult and labor-intensive (by factors estimated at between 2× and 5×) than creation for one-time use.

SoCs exploit aggressively technology evolution (Moore's Law), since moving to a scaled technology is an inexpensive way of achieving a better (smaller, lower power, and faster) part with low design effort. From the design technology viewpoint, SoC design is characterized by relatively conservative design methods and design goals and have traditionally being characterized by inferior clock frequency and layout density compared to custom-designed processors (MPUs). However, the quality gap between full-custom and ASIC/SoC has steadily been reduced. From the 2001 ITRS roadmap edition, ASIC and MPU logic densities were modeled as being equal; and in addition "custom quality on an ASIC schedule" has been getting closer and closer by virtue of improved physical synthesis and tuning-based standard-cell methodologies. At the same time, MPUs have evolved into SOC's following two directions of evolution. First, MPUs are increasingly designed as IP-cores to be included in SoCs. Second, MPUs are themselves designed as multi-core SoCs to improve reuse and design productivity and achieve an acceptable power and performance. Furthermore, SoCs for a number of high-end market sectors, such as networking and gaming, are characterized by increasingly demanding performance specifications with required performance metrics (e.g., per-die floating point operations per second, or per-die external I/O bandwidth) beyond those of conventional general-purpose processors. In light of these requirements, SOC designs are now the driver for the growth of key parameters, such as number of cores per die, maximum frequency per core, and per-pin I/O bandwidth.

The fundamental design challenges for SoCs are implementation productivity and manufacturing cost. The fundamental technology challenge is the heterogeneous integration of components from multiple implementation fabrics such as logic, Flash and DRAM memory, analog and radio frequency (RF), micro electro-mechanical systems (MEMS). As highlighted above, SoCs are characterized by heavy reuse of IP-cores to improve design productivity, and by system integration of heterogeneous technologies, to provide low cost and high integration. Cost considerations drive the deployment of low-power process and low-cost packaging solutions, along with fast-turnaround time design methodologies. Integration considerations drive the demand for heterogeneous technologies in which particular system components (memory, sensors, etc.) are implemented, as well as the need for chip-package co-optimization. As a consequence, SoCs drive the convergence of multiple technologies not only in the same system package, but also in the same manufacturing process.

The need to build heterogeneous systems on a single chip is driven by such considerations as cost, form-factor, connection speed/overhead, and reliability. Process complexity is a major factor in the cost of SoC applications, since more technologies assembled on a single chip requires more complex processing. Cost considerations limit the number of technologies on a given SOC. Today, a number of technologies (MEMS, GaAs) are more cost-effectively integrated vertically or side-by-side in the same package module depending and not on the same die. In this case we use the term system-in-package (SiP).

In the remainder of this chapter we will give an historical excursus on the recent architectural evolution of SoCs. We will focus mostly on recent history, starting from the late nineties. In this period, SoCs have been integrating multiple heterogeneous processors, as well as complex software infrastructure for supporting parallel execution

and manage distributed processing, communication and storage resources. Our main emphasis will therefore be on Multi-processor Systems-on-chip architectural evolution rather than system integration of heterogeneous technologies (MEMS, RF, Optical, etc.). Following the historical review, we will survey current architectural trends and we will survey key design and design automation challenges. Our discussion will include an outlook towards future and emerging technologies and applications.

3. Historical Perspectives

In this section we provide an overview of (MP) SoC evolution. A complete treatment is beyond the scope of this chapter, and the interested reader is referred to several survey papers and books on this topics [2][3][4]. We will follow the lines highlighted by Wolf et al. [5] to survey representative examples that illustrate the types of MPSoCs that have been designed and the historical trends in their development.

Early SoCs in the nineties featured typically one processor serving as central processing unit (CPU), one special-function processor and a number of fixed-function sub-systems targeted to specific applications, internal memory and plenty of IO-interfaces. An example of such a late 1990s SoC is shown in Figure 1 [6]. The figure summarizes the main features of Infineon's E-GOLD+ SoC for second-generation digital telephony, implemented in 0.25 μ m technology. The architecture is a heterogeneous dual-core, featuring a general purpose CPU (the C166CBC processor) for running the phone OS, the GUI and for overall system coordination, and a DSP for digital baseband decoding. A significant amount of on-chip data memory (both volatile and non-volatile) and program memory is integrated on the chip. Even though the corresponding silicon area is not highlighted in the chip micro-photograph, several dedicated hardware blocks are also integrated, for modulation, digital filtering and equalization. In addition a number of standard IOs are supported (IrDA, keypad, etc.). The chip also integrates the analog front-end. From the architectural viewpoint, this SoC is only moderately parallel in terms of IP-cores, and highly heterogeneous. The trends toward multiple programmable cores, architectural heterogeneity and complex on-chip memory hierarchy are already evident in this early SoC. Note that the E-GOLD+ design pre-dates by almost one decade multicore processors for general-purpose computing.

As demonstrated by early GSM transceivers SoCs, such as E-GOLD+, a multi-processor SoC platform, i.e. an architectural template where multiple heterogeneous processors can be instantiated was already considered a viable concept in the nineties [7]. The Lucent Daytona chip [8], introduced in year 2000 and shown in Figure 2, pushed this trend aggressively, and it is among the first examples of market-ready scalable MP-SoC architectures, which allow the instantiation of multiple arrays of homogeneous processors. Daytona was designed for wireless base stations, in which identical signal processing is performed on multiple data channels. This is a perfect use case for a parallel architecture, as the target application is embarrassing parallel. Hence, Daytona was developed as a symmetric architecture with four CPUs attached to a high-speed bus. The CPU architecture is based on an enhanced SPARC V8: 16 \times 32 multiplication, division step, touch instruction, and vector coprocessor. Each CPU has an 8-KB 16-bank cache. Each bank can be configured as instruction cache, data cache, or scratch pad. The processors are connected with the memory system and peripherals

through an advanced on-chip bus capable of multiple outstanding transactions and out-of-order completion. The processors share a common address space in memory; the caches snoop to ensure cache coherency. The chip was 200 mm² and ran at 100 MHz at 3.3 V in a 0.25-µm CMOS process.

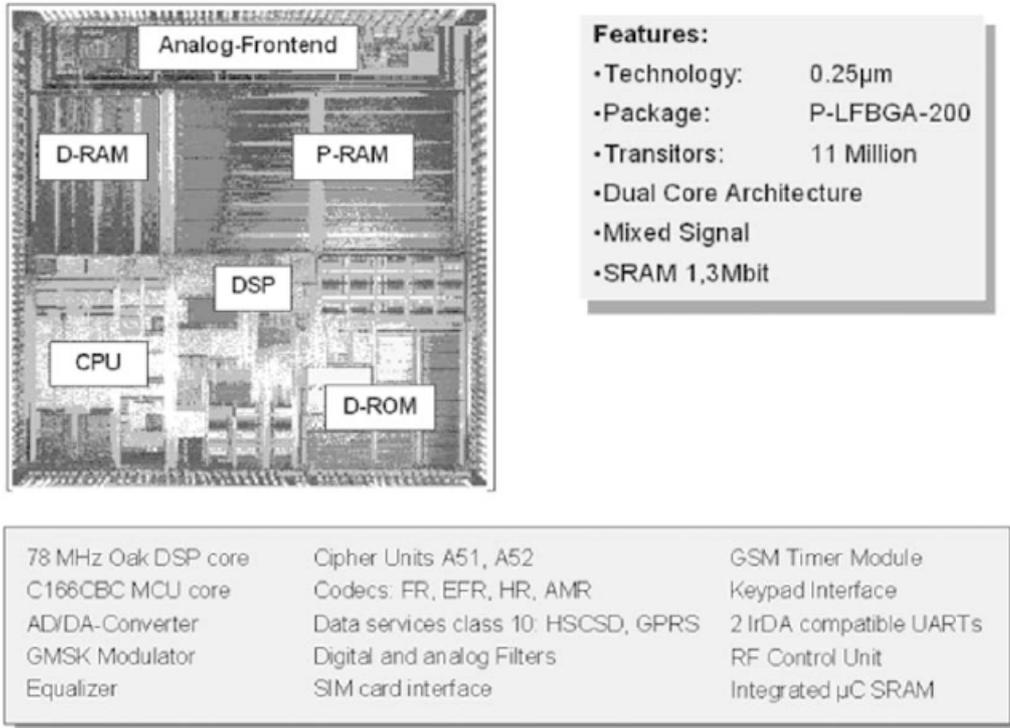


Figure 1. Infineon GSM transceiver E-GOLD+ (1999)

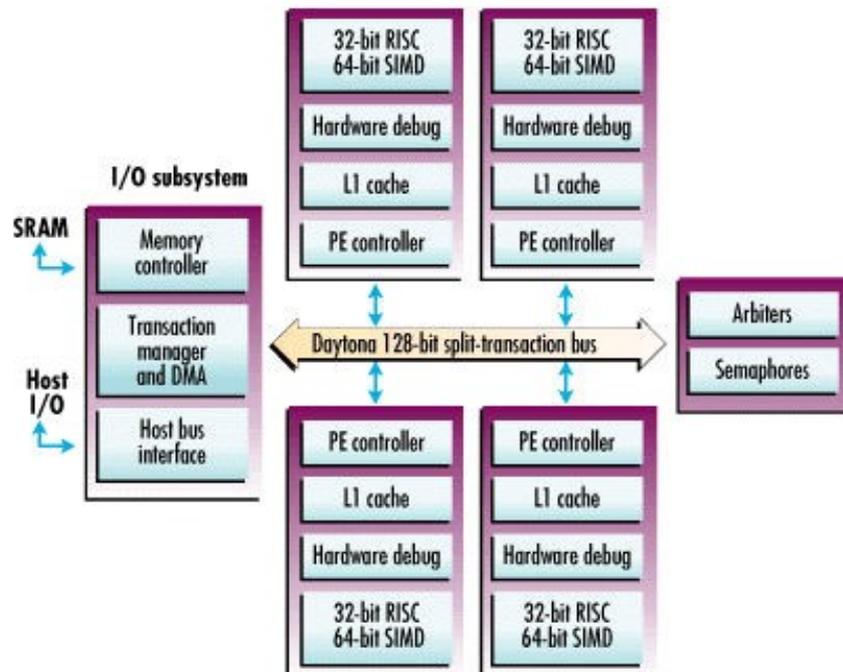


Figure 2. The Lucent Daytona Chip (2000)

In the same time frame as the Daytona, several other scalable MPSoCs were announced, in a wide range of application markets support a range of applications, exhibiting a wide range of architectural variations.

Video and multimedia processing for digital television is another large market where MS-SoCs have been gaining widespread acceptance. An early example of a multimedia processor is the Philips Viper Nexperia [9], shown in Figure 3. The Viper includes two CPUs: a MIPS and a Trimedia very-long instruction word (VLIW) processor. The MIPS acted as a master running the operating system, whereas the Trimedia acted as a slave that carried out commands from the MIPS. The system includes three buses, one for each CPU and one for the external memory interface. Bridges connect the buses.

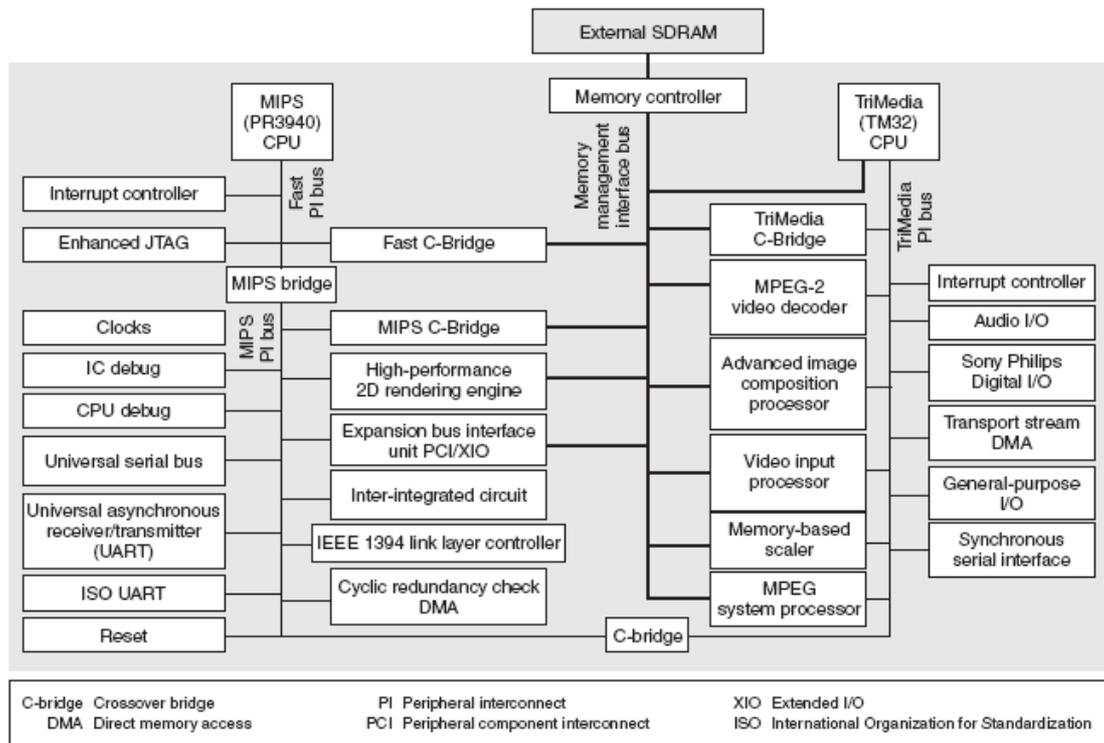


Figure 3. Philips Viper NEXPERIA Chip (2001)

However, the multiprocessing picture is more complicated because some hardware accelerators are attached to the buses. These accelerators perform computations such as color space conversion, scaling, etc. The Viper could implement a number of different mappings of physical memory to address spaces.

-
-

TO ACCESS ALL THE **20** PAGES OF THIS CHAPTER,
Visit: <http://www.eolss.net/Eolss-sampleAllChapter.aspx>

Bibliography

- [1] International Technology Roadmap for Semiconductors, System Drivers, 2010 Update (www.itrs.net). [The main reference document, continuously updated, specifying the evolution of silicon technology]
- [2] G. Martin, H. Chang, (2003) *Winning the SoC Revolution*, Springer. [Reference textbook for design methodologies for SoCs, circa 2001]
- [3] M. Flynn, W. Luk, (2011) *Computer Systems Design: System-on-Chip*, Wiley. [Reference text on SoC architectures and design]
- [4] M. Keating, (2011) *The Simple Art of SoC Design: Closing the Gap between RTL and ESL*, Springer. [Reference text on design technology with industrial emphasis]
- [5] M. Wolf, A. Jerraya, G. Martin, (2008) Multi-Processor System-on-Chip (MPSOC) technology, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 27, no. 10, pp. 1701-1713. [Survey paper on multi-processor system on chip architectures and design challenges]
- [6] G. Weinberger, (2000) The New Millennium: Wireless Technologies for a Truly Mobile Society, IEEE International Conference on Solid-State Circuits.
- [7] T. Claasen, High speed: not the only way to exploit the intrinsic computational power of silicon, IEEE International Conference on Solid-State Circuits, 1999.
- [8] B. Ackland, et al., (2000) A single-chip, 1.6-billion, 16-b MAC/s multiprocessor DSP, *IEEE Journal of Solid-State Circuits*, Vol. 35, no. 3, pp. 412–424, Mar. 2000. [One of the earliest examples of multi-processors SoCs]
- [9] S. Dutta, R. Jensen, and A. Rieckmann, (2001) Viper: A multiprocessor SOC for advanced set-top box and digital TV systems, *IEEE Design & Test of Computers*, Vol. 18, no. 5, pp. 21–31, Sep./Oct. 2001. [Another early example of multi-processor SoC]
- [10] Pham, et al, (2006) Overview of the architecture, circuit design, and physical implementation of a first-generation CELL processor, *IEEE Journal of Solid State Circuits*. Vol 41, no 1. pp 179-196. [Paper describing in depth architecture and circuit implementation of the first CELL BE chip]
- [11] L. Benini, G. De Micheli, (2006) *Networks on Chip: Technology and Tools*, Morgan-Kaufmann. [Reference textbook on Networks-on-chip, state-of-the-art in 2005]
- [12] G. De Micheli, et al, (2010) Networks on chips: from research to products, IEEE Design Automation Conference. [Survey on industrial adoption of NoCs in 2010]
- [13] B. Calhoun, D. Brooks, (2010) Can Subthreshold and Near-threshold Circuits go Mainstream?, *IEEE Micro*, vol 30, no. 4, pp. 80-85. [Interesting discussion on the trend toward ultra-low voltage CMOS digital design]
- [14] N. Hardavellas, F. Michael, B. Falsafi, A. Ailamaki, (2011) Toward Dark Silicon in Servers, *IEEE Micro*, Vol. 31, no. 4, pp. 6-15. [Discusses the trend toward dark-silicon, i.e. chips that can be powered only partially at any given time]
- [15] A. Bartolini, M. Cacciari, A. Tilli, L. Benini, (2011) A distributed and self-calibrating model-predictive controller for energy and thermal management of high-performance multicores, IEEE Design Automation and Test in Europe Conference. [A paper introducing distributed thermal control strategy for many-core architectures]
- [16] G. Van Der Plas, et al. (2011) Design Issues and Considerations for Low-Cost 3D TSV IC Technology, *IEEE Journal of Solid State Circuits*, Vol. 46, no. 1, pp. 293-307. [Paper describing technology details and chip prototypes of through-silicon via technology]
- [17] A. Shacham, K. Bergman, L. Carloni, (2010) Photonic Networks-on-chip for Future Generations of Chip Multiprocessors, *IEEE Transactions on Computer*, Vol. 57, no. 9, pp. 1246-1260. [Survey paper on on-chip photonic communication]
- [18] M. Huebner, J. Becker, (2010) *Multiprocessor System-on-Chip: Hardware Design and Tool Integration*, Springer. [Textbook on system-on-chip design methodologies]

- [19] M. Toksvig, J. Mathieson, B. Cabral, B. Smith, (2008) NVIDIA Tegra: Enabling Stunning Handheld Graphics & HD Video , *IEEE Hot Chips*. [Survey of Tegra chips and roadmap]
- [20] NVIDIA White Paper, The Benefits of Multiple CPU Cores in Mobile Devices, www.nvidia.com, 2010. [Commercial white paper describing the Tegra mobile application processors roadmap]
- [21] International Technology Roadmap for Semiconductors, Design, 2010 Update (www.itrs.net) [Another section of the ITRS]

Biographical Sketch

Luca Benini is Full Professor at the Department of Electrical Engineering and Computer Science (DEIS) of the University of Bologna. He also holds a visiting faculty position at the Ecole Polytechnique Federale de Lausanne (EPFL) and he is currently serving as Chief Architect for the Platform 2012 project in STmicroelectronics, Grenoble. He received a Ph.D. degree in electrical engineering from Stanford University in 1997.

Dr. Benini's research interests are in energy-efficient system design and Multi-Core SoC design. He is also active in the area of energy-efficient smart sensors and sensor networks for biomedical and ambient intelligence applications.

He has published more than 600 papers in peer-reviewed international journals and conferences, four books and several book chapters. He has been general chair and program chair of the Design Automation and Test in Europe Conference. He has been a member of the technical program committee and organizing committee of several conferences, including the Design Automation Conference, International Symposium on Low Power Design, the Symposium on Hardware-Software Codesign. He has been Associate Editor of several international journals, including the the IEEE Transactions on Computer Aided Design of Circuits and Systems and the ACM Transactions on Embedded Computing Systems. He is a Fellow of the IEEE, a member of the Academia Europaea, and a member of the steering board of the ARTEMISIA European Association on Advanced Research & Technology for Embedded Intelligence and Systems.