Systems on Chip: Evolutionary and Revolutionary Trends

Arnaldo Rui Brás

Dep. Informática, Universidade do Minho
4710-057 Braga, Portugal
arb@igifp.min-saude.pt

Abstract. Emerging computational needs imposed by the market lead the major industries to adopt the "systems-on-chip" design. This paper briefly demonstrates the whys and the hows associated with this activity. It shall be outlined the major issues that must be accomplished in the process of designing, verifying and publishing those systems. Finally, it will establish a link between these techniques with the latest achievements of science in the computer science domain.

1 Introduction

Since the construction of the first ENIAC computer, to our days, computing design techniques, materials, tools and strategies suffered so many changes at an incredible rate that the subject, itself, is a matter for a vast range of literature.

In the 90's decade, taking a natural advantage of the Very Large Scale Integration VLSI and Electronic Design Automation (EDA), a phenomenon came out revolutionizing the traditional way of designing and prototyping hardware systems. Such phenomenon is usually called Systems-on-Chip, commonly designated by SoCs.

The term system-on-chip has been granted to specific computing devices that can incorporate in one single chip the functionality of several general purpose or application specific chips. Basically, this technique aims to throw into a common chip the functionality of several of those chips. In order to achieve this goal, design, verification and testing techniques for integrated circuits (IC) suffered an undeniable evolution.

We can see this type of computer systems implemented in so many different systems such as TV Set-Top boxes, ATM peripherals, medical equipment, routers, modems, Ethernet adapters etc…

This communication emphasizes the importance of the Systems on Chip in the evolutionary path of computer design both in industry and academic environments. It will outline and document the most prominent architectural characteristics, problem issues, design techniques, strategies and methodologies used when building these components. Finally, will present some of the main advances in today's science that may, in some way, interfere drastically with the future evolution of the computing design techniques.

2 Scope and Historical Evolution

Since the arrival of the first transistor based computer that integration and high scale integration became one of the main concerns in the hardware design techniques.

In early 70's relatively high levels of integration were achieved, but the concern for miniaturizing and building more and more complex circuitry were kept as one of the goals in leading computer construction and chip design.

Architectural designs at this time, consisted in a CPU, eventually with an FPU and a few electronic components attached "on-board", namely the peripherals. Later on, with the technological evolution, some of these components evolved and became themselves processing units with their own on-board components. Graphical and Audio interfaces, Modems, Codecs, Digital Signal Processors (DSP) and networking boards are a few good examples of such components.

During the 80's decade, the complexity and functionality of these components evolved and soon, constructors began to develop efforts in making them compatible with the widest range of microprocessors. In the late 80's, early 90's period, the appearance of these components, integrated in Embedded Systems or Application Specific Integrated Circuits (ASIC), became a reality.

Conventionally, ASIC design involved development of medium complexity Integrated Circuits (of less than 500,000 gates). These had a cycle time of roughly 6 months, were processed with 0.35u technology, and were essentially made up of core logic and some hard macros.

With rapid advances in semiconductor processing technologies, the density of gates on the die increased in the line that Moore's law predicted. This allowed the realization of complicated designs on the same IC. Over the last few years, with the advent of bleeding edge technology applications like HDTV and 3rd generation mobile devices, an increasingly evident need has been of incorporating the traditional microprocessor, memories and peripherals, i.e., the whole system, on a single chip.

Paradoxically today, the emergence of system-on-chip technology brings a whole spectrum of opportunities and challenges. Opportunities come with the drastic reduction in the overall cycle time of the system with superior performance levels. Challenges are the result of deep sub-micron complexities, testability issues and time-to-market pressures.

3 Design and Architectural Aspects

One of the key aspects in a SoC that determines its functionality, performance and success is, with no doubt, its architecture. The objective is to maximize system performance and minimize time to market. The system architecture is the right place to start this objective.

The factor that most affects the SoC architecture is its target domain. This parameter determines which components are going to be thrown into the chip and those who should be kept outside the chip. In this section shall be discussed the main design techniques and some key issues regarding the System on Chip architectural aspects.

3.1 Design Issues

High-performance designs beyond sub-half-micron technologies, with clock frequencies over 100MHz, have received much attention from de design community and the EDA industry in recent years. Interconnect delay has become a significant factor affecting design performance. The traditional approach of sequential, flat physical design (PD), with multiple iterations of placement, routing, and timing verification, can no longer provide an effective chip design solution.

In order to solve this problem a technique of both Hardware and Software (HW/SW co-design) design was adopted. Intellectual Property (IP), a software block describing the structure and behavior of a digital system component, became the main tool for system designers. Basically IP, comes out in three forms: Soft IP, Firm IP and Hard IP, and can be integrated in the design of more complex systems.

These three approaches have advantages and disadvantages that resume to greater flexibility and loss of performance for the Soft IP approach and the reverse for the Hard IP approach. The target domain for the system, mainly conditions the adoption of one of these.

The problem of HW/SW co-design itself has so many dimensions that covering all of them is proved to be impossible within a single design environment. There are tens of different methodologies, frameworks and co-design tool-sets having their own niche in the global application domain. This explains fuzziness of definitions and the fact that terminology cannot follow rapid changes in hot spots of digital design world.

There are two main approaches for SoC/ASIC design: Top-down system design and platform based design.

In the top-down system design the end-user product is designed to meet specific market needs. Intellectual Property (IP) selection and integration are key factors for success. The producer must combine the best components by buying some and building others.

In the platform-based design a silicon product is designed to perform specific system functions. IP definition, validation and delivery are the main tasks to achieve. The producer must create a virtual SoC reference design.

3.2 Architectural Issues

On the architectural side, the choice of design components and their interaction are crucial in determining the overall system performance. On the performance side the compliance to system specifications or requirements such as external environment, real-time demands, power consumption, total throughput (such as the number of concurrent channels supported) have to be taken in consideration.

When designing SoCs, factors like the processors, the bus architecture and its overall interconnection, the memories, their management and access, custom logic for the optional and/or reconfigurable peripheral devices and the design software tools must be taken in consideration.

The main issues that must be respected when planning a SoC are:

- Host chip - The selection of the host chip, the physical layer of the SoC, is usually a RISC/MIPS CPU in order to benefit from the modularity and well-fitted design provided by these CPUs. The most popular and widely adopted processors are normally the ARM and MIPS processors because there is a wide range of implemented and well tested peripheral IPs for these processors.
- Main Cores – The selection of the mapped CPUs scheduled to the SoC are often a matter for special care in the planning stage. Here the expression *divide and conquer* rules in the floor planning stage. RISC cores, are often selected, once these architectures are efficient, well proven, easy to implement and don't take much space.
- Interconnectivity – This has to do with the choice of the internal BUS connecting the CPUs with other components in the system. Here, designers also assume general purpose, universal and well known interfaces in order to achieve both modularity and possible reconfiguration. The following issues are normally evaluated for bus architecture:
    - Standard (e.g. AMBA) versus custom Bus.
    - Arbitration (scheduling algorithms).
    - Bus width.
    - Bus pipelining (wait state versus split transaction).
    - Direct Memory Access (DMA) with or without arbitration.
    - Hierarchy (e.g. system bus versus peripheral bus).

- Peripherals – Peripherals implemented in the SoC are strictly connected to the objectives assumed for the system. The choice of these peripherals is conditioned by the former choices and IP availability. Parameterization and reconfigurability are some aspects that can determine the final peripheral set

Scalability, parallelism, multiprocessing and pipelining of the core components of SoCs are often conditioned by the options taken in the floor planning stage. In fact, all the main characteristics of the adopted cores are partially inherited by the system.

4 Application domain and features

Classification of designs on base of application is complicated due to target context switching in time. System under design today is tomorrow's module or library element. Still, division into three overlapping areas on base of domination is quite general:

- **Data flow**: systems with heavy signal processing needs as speech, audio, radar images, video, graphics and multimedia.
- **Control flow**: in general, real-time systems belong here. Actually, only some layers of real-time systems are control-intensive, the rest is composed from data processing elements, memories etc.
- **Memory intensive**: systems, where main attention is paid to implementing the suitable distributed, hierarchical, and specialized memory architecture like in telecommunication switching systems. Data processing in these systems is reduced mostly to comparison operations

This section outlines some groups of applications and the way they affect the architecture. The following subsections will illustrate a Platform-based reference architecture, a Data-flow oriented SoC, with reconfigurable characteristics and a revolutionary reconfigurable SoC architecture that can address both Data-flow and Control-flow applications.

4.1 SoC-RaPTor Architecture

Based on the ARM processor family, the SoC-RaPTor Platform from Wipro, is a virtual SoC reference design used for system design and generation in this company. Being a Platform-based strategy, this model, enables design teams to save time in the floor planning stage and in choosing the right bus architecture.

To Wipro, the key aspects of a successful SoC design strategy are: *Architectural strategy*, *Validation strategy*, *Design for Testing strategy*, *Synthesis & Backend strategy* and *Integration strategy*. Adopting this strategy [9], synthesis analysis resumes to testing and verification, enabling the design team to deliver in a time-to-market their new SoC solutions.

Adding, changing, or removing components to this Platform-based architecture are the main operations made in the design process. Logical Synthesis was replaced by Physical Synthesis and, each and every one of the constituent components of the system, has a hard implemented testing module to satisfy verification and validation requirements. Design-for-Testing (DFT) strategy is adopted, since early design stages, in order to meet time-to-market requirements. Parameterization and Functional partitioning are the two main key issues in design-for-integration techniques applied to address the challenge in integrating IP cores into the SoC.

4.2 Orca Series 4 Reconfigurable Embedded SoC Architecture

Lucent Technologies Microelectronics Group presents a new concept in SoC architectures introducing a reconfigurable embedded Soc approach [7]. This feature, taking advantage of the reconfigurability of Field Programmable Gate Arrays (FPGA) is present in the ORCA Series 4 architectures. The figure bellow demonstrate the three basic design configurations (from left to right: Field Programmable System Chip, Generic FPGA and Embedded Programmable Macrocell) for ORCA Series 4 products come in:

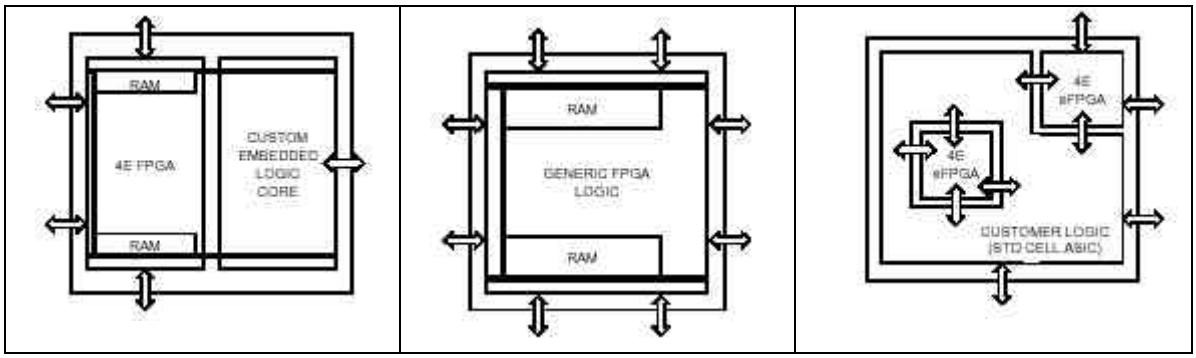

Figure 1 - Basic Components for Reconfigurable Embedded Architecture [7]

The architecture consists of four basic elements: Programmable Logic Cells (PLC), programmable I/O cells (PIO), embedded block RAMs (EBR) and system level features, with an on-chip CPU based on a PowerPC II processor core. These elements are interconnected with a routing fabric of both global and local wires. An array of PLCs is surrounded by common interface blocks, which provide interface to the adjacent PLCs, or system blocks. Each PLC contains a Programmable Functional Unit (PFU) SLIC, local routing resources and configuration RAM. Most of the FPGA logic is performed in the PFU, but, decoders, PAL-like functions and 3-state buffering can be performed in the SLIC.

Each PIO contains four programmable I/O pads and is interfaced through a common interface block to the FPGA block. The PIO is split in two pairs of I/O pads with independent clocks. The new programmable I/O cells allow designers to select I/Os which meet many new communication standards permitting the device to hook up directly without any external interface translation.

The Series 4 also provides system-level functionality by its microprocessor interface, embedded system bus, quad-port embedded block RAMs, universal programmable phase-locked loops and the addiction of networking specific phase-lock loops.

The microprocessor interface MPI provides a gateway between the FPGA blocks and the PowerPC microprocessors. All MPI transactions use the embedded system bus.

This is an example of both dataflow and control-flow design systems. The FPGAs functionality is determined by an internal configuration RAM. The internal configuration/initialization can reside externally in an EEPROM or any other storage media. The RAM can be loaded by master/slave serial or parallel, asynchronous peripheral modes and the MPI. Daisy chaining of multiple devices and partial reconfiguration are allowed.

4.3 Application Specific Reconfigurable Core for Dataflow Oriented Computing

The Laboratoire d'Informatique et de Robotique et de Microelectronique de Montpellier (LIRMM) [4], created and disposed a new reconfigurable SoC architecture. Implementing

reconfigurabilty by the use of FPGA components, this architecture expands the target domain of SoCs/ASIC components.

The revolutionary issue in this architecture is featuring a highly optimized, DSP-like coarse-grained reconfigurable block: the DataPath node. The operative layer is no longer Cell Level Based (CLB), but use a coarse-grained component: the Dnode. This component is configured by a microinstruction code.

The architecture also includes a configuration component: the configuration layer, which is, basically constituted by a RAM component and a custom RISC core, with a dedicated instruction set, containing the microinstruction code for each Dnode active in the system. The RISC core main task is to manage (dynamically or not) the configuration of the Dnodes network and to control data communications between the reconfigurable core and the host CPU. Using special feedback pipelines, forming a reverse Dataflow, complex routing structures are avoided.

This architecture proposes a coarse grained arithmetic block (less efficient, but dynamically reconfigurable) which proofs its efficiency in data oriented processing.

5 Recent Achievements Unveil Today's Chip Manufacturing

A single drop of water contains trillions of computers. Recently, came out the first computer based on DNA/RNA Cells. The model beyond this achievement is been a direct application of the Turing machines to the biological knowledge. This molecular machine implements a finite automaton whose basic operation is described by Prof. Shapiro in [2].

Transistor action in a single organic molecule. In October 2001, scientists from the Bell Laboratories finally managed to build a transistor at the molecule scale size [5]. With this achievement the granularity of the micron scale suffers a million times increase. Hence, the architectural problems posed until recently in a VLSI system cease to be so critical.

An Architecture Description Language. Verification is one of the most complex and expensive tasks in the current SoC design process. Many existing approaches employ a bottom-up strategy to the pipeline validation, where the validation synthesis is reverse-engineered from its RT-level implementation. The Center for Embedded Computer Systems at the University of California of Irvine presents a new approach to address this problem [8].

Assuming a digital system as a graph [8] with balanced weights, this method allows the description of a system by a set of Architecture Description Language (ADL) constructs, levering the architect's knowledge to a top-down approach to meet both structure and behavior of the SoC.

This methodology presents a formal approach for automatic validation of architectures described in ADL and contributes for a better Design Space Exploration (DSE). This technique already proved its usefulness verifying various known architectures such as: MIPS R10K, TI C6x, PowerPC, DLX and ARM that represent RISC, DSP, VLIW, and Superscalar architectures [8].

Amorphous Computing. A new Programming Paradigm is under development at MIT's Artificial Intelligence Laboratory [10]. Basically, amorphous computing combines organizational principles and programming languages for obtaining a coherent behavior from a series of unreliable parts that are interconnected in unspecified ways. The main idea comes from combining latest developments in micro-fabrication and biology that make possible

to assemble a huge number of almost identical information processing agents at almost no cost. Further developments of this theory will use basic cellular computing as an approach to constructing digital logic circuits within living cells representing logic levels by concentrations of DNA-binding proteins.

Beating the Clock FEETzero Is an Asynchronous Machine.Sun Labs Asynchronous Design has been working on FLEETzero [11][12], a prototype chip with raw speed roughly twice that today's chips. While today's chips use "synchronous" circuits with a global clock to manage activity, this new concept, which FLEETzero implements, based on asynchronous logic elements that produce timing signals only where and when needed.

This new approach embodies the asynchronous principle that allows the logic itself to manage the time and resources required to move data. This simplifies design by eliminating the circuitry required to manipulate a system clock.

6 Conclusions

Although IP and Hardware-Software co-design were encouraged in the early SoC years, designers tend to prefer hard, well-proven and tested implementations for their SoCs. This fact comes from the natural appearance in the market of a wide range of circuitry previously tested in other systems.

Multiprocessing, Parallelism and Pipelining have been some of the architectural issues that most revolutionized the computing sciences. SoCs, naturally, took advantage of incorporating parallel, multiprocessors with pipelining features in their main chip cores. Still, the newly formed system, don't show a behavior similar to its constituent parts. Unfortunately, there are many interesting ideas about transforming SoC into multi-scalar parallel systems there isn't a formal golden method to do it.

Although recent, as a technology, SoC and ASIC design have suffered various evolutions during its short story. The next step is the standardization and formalization of both Hardware Description Languages (HDL) and Architecture Description Languages (ADL). With this goal acquired, the evolution and evolvement of these systems will occur at a faster rhythm, once prone the verification and validation tasks, which take about 60% of design time.

In the previous section were presented some ideas that surely will modify computer architecture in the future. Imagine what could be achieved when applying these techniques to SoCs…

References

[1] Keating, M., Bricaud P.: Reuse Methodology Manual for System-on-Chip Design. Kluwer Academic Publishers. (October 1998).

[2] Benenson, Y., Paz-Elizur, T., Adar, R., Keinan, E., Zvi L., Shapiro E.: Programmable and autonomous computing machine made of biomolecules. See http://www.weizmann.ac.il/math/users/lbn/public_html/new_pages/Abstract.html.

[3] Smith, M. J. Sebastian: Application-Specific Integrated Circuits… the Course. VLSI Design Series: Addison-Wesley Longman. http://spectra.eng.Hawaii.ed/~msmith/ASICs/.

[4] Sassatelli, G., Robert, M., Torres, L., Cambon, G., Diou, C., Galy, J.: Winner of the University Best IP 2001 Prize: «Application Specific Reconfigurable Core for Data-flow Oriented Computing ». Laboratoire d'Informatique et Robotique et Microelectronique de Montpellier, France. See http://www.us.design-reuse.com/WORKSHOP/lirm.html

[5] Bell-Labs Scientists build the World's smallest transistor, paving the way for the "Nanoelectronics". See http://www.bell-labs.com/news/2001/november/8/1.html. (November 2001).

[6] Sayah, J.Y., Gupta R, Sherlekar, D.D., Honsinger, P.S., Apte, J.M., Bolliger, S.W. , Chen, H.H., Das Gupta, S., Hsieh, E.P., Huber, A.D., Hughes, E.J., Kurzum, Z.M., Rao, T., Tabtieng, V., Valijan, D.Y. Yang: Design planning for high-performance ASICs. IBM Journal of Research and Development, Vol 40, Number 4, (1996).

[7] Lucent Technologies: ORCA Series 4 Reconfigurable Embedded SoC Architecture. Advance Product Brief; May 2000.

[8] Mishra, P., Dutt, N., Nicolau, A.: Architecture Description Language driven Validation of Processor, Memory, and Co-Processor Pipelines. Dept. of Information and Computer Science; University of California Irvine. Technical Report #01-55 (July 2001).

[9] Kamth, U., Kaudin, R.: System-on-Chip Designs: Stategy for Success. Wipro Technologies White paper. (June 2001).

[10] Abelson, H., Allen, D., Coore, D., Hanson, C.: Amophous Computing. MIT Artificial Intelligence Laboratory. See http://www.swiss.ai.mit.edu/projects/amorphous.

[11] Sutherland, I., Fairbanks S.: GasP: A Minimal FIFO Control. Sun Microsystems Laboratories, Mountain View, CA. ASYNC-2001, Salt Lake City. (March, 12 2001). See http://research.sun.com/features/async.

[12] Coates, B., Lexaux, J., Jones, I., Fairbanks, S., Sutherland, I.: FLEETzero: An Asynchronous Switch Fabric Chip Experiment. Sun Microsystems Laboratories, Mountain View, CA. ASYNC-2001, Salt Lake City. (March, 12 2001). See http://research.sun.com/features/async.