Preface

Special Issue on Networks on Chip

Axel Jantsch, Johnny Öberg and Hannu Tenhunen

Royal Institute of Technology, Stockholm, Sweden

Every increase in capacity and complexity of integrated circuits has spurred the quest for ways to restrict the design space without sacrificing optimality. The microprocessor has done this in a very effective way, because it represents a general pattern to organize computation in hardware. The separation into a data-path with a number of optimized data manipulation units, a control machine that orchestrates the data manipulations, and the memory that stores the control and input, output and intermediate results, is effective and useful for every kind of application. Thus it provides an architectural pattern that implies an effective design methodology allowing to quickly implement any given algorithm. In fact, this architectural pattern is so general that the hardware implementation could be standardized and the architecture implementation is performed by storing a sequence of control instructions in memory, i.e. in software. The 1980s and 1990s saw a further steady increase of on-chip capacity and with it the investigation into a large variety of different variants of this basic architectural pattern leading to CISCs, RISCs, DSPs, ASIPs and a large number of customized ASIC architectures.

In the 1990s, the increase of on-chip computation capacity could not effectively be used to simply enhance the traditional processor architecture. In many application cases it became more efficient to use two or more processor cores and run algorithms in parallel rather than make just one very fast processor. This was driven in part by the difficulties to make a single processor ever faster in a cost and power efficient way, and in part by the desire to reuse existing algorithms and applications and just integrate them on one chip. This allowed also to use and integrate special purpose hardware engines such as DSPs, memory management units, protocol implementations, encoders and decoders, etc. for specific parts of the application. In essence, the design space made available on chip by many million transistors has been restricted to the reuse and integration of a few dozens or hundreds of (Intellectual Property) IP blocks.

When the number of IP blocks grows further, the complexity of their interaction increases exponentially, thus calling for a new round of design space

restriction. The central difficulty lies now in the complexity of interaction of many concurrent activities and tasks at both the application and the architecture level. In addition deep sub-micron effects pose formidable physical design challenges for long wires and global on-chip communication. Consequently, communication has to be addressed in a very systematic manner. Communication at several levels from physical wires to inter task information exchange is the central focus of the emerging area of networks on chip (NoC). Essentially a NoC platform provides a communication infrastructure to which all computation and storage resources can connect to. They can use the platform's services to synchronize and communicate with other resources in a transparent manner. While the focus of NoC is the provided communication infrastructure, the central idea of the platform is to restrict the design space in a meaningful way and to effectively separate application development from implementation as illustrated in figure 1.

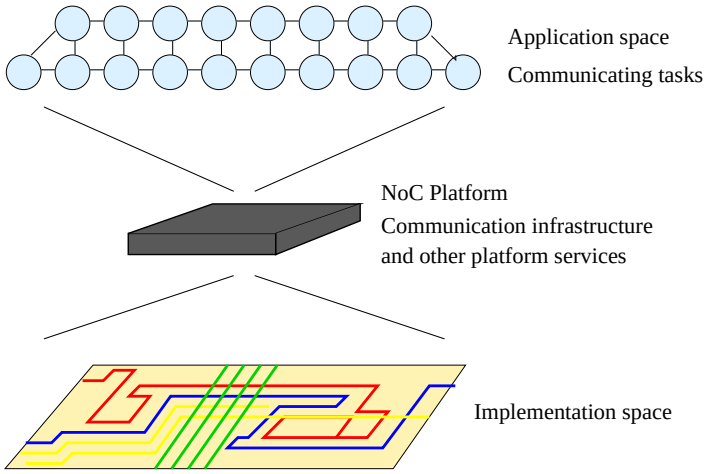

Fig. 1. The platform as interface between application design and platform implementation.

Since the main challenges addressed by platforms come from the complexity of communication and interconnection design, a sophisticated communication architecture will probably be central to most platforms emerging in the next few years. A likely scenario is that in a few years a small number of standardized platforms will emerge since this will maximally benefit both application and platform developers. Application developers can develop the application independently of the physical platform implementation and they can easily migrate to a new, more powerful one. In fact, end users may be able to insert new platform instances in the same way as they can buy and use new processors today. Depending on the definition of the platform, re-compilation and re-mapping may or may not be necessary.

Platform developers will benefit from a small set of standard platform definitions because they can design and optimize a platform implementation independently of a particular product. Their investment cost can be shared by

many products allowing for much more aggressive optimization and higher development cost than a single or small number of products could justify.

The jury is still out on the number of different platforms, how general purpose or how domain specific they will be. This will be determined by complex performance-cost-power trade-off functions as well as by a number of non-technical factors in the market place.

The NoC area has grown dramatically as a research topic over the last few years. Today, probably over 30 groups at Universities, research institutes, and industry are active in this area. For this special issue four out of 23 submitted papers have been selected. They represent several interesting investigations into NoC core issues.

J. Liu, M. Shen, L-R. Zheng and H. Tenhunen present a thorough study of wiring and interconnect issues in future technology generations. Based on this analysis a methodology for designing the inter-switch connection is proposed to optimize bandwidth given delay, power and area constraints. Moreover, a switch design is proposed that implements a circuit switched time division multiplexing switching technique.

The second article by T. Ye, L. Benini and G. DeMicheli also studies architectural trade-off parameters. In particular they compare the different routing policies hot potato routing, dimension order routing and their own proposed adaptive contention look-ahead routing scheme. Furthermore, they analyze the effect of packet size on the performance and the power consumption.

The following article by E. Bolotin, I. Cidon, R. Ginosar and A. Kolodny is also an architecture study but with strong emphasis on Quality of Service (QoS). All traffic is grouped into the four following service classes: Signalling traffic for control, Real-time for delay constrained data streams, RD/RW for memory access, and Block-transfer for bursty traffic. Each class is defined in terms of delay and throughput requirements. For these service classes a NoC is proposed, which features a mesh architecture and a deterministic, wormhole routing scheme.

Finally, a modelling framework for simulation and analysis of NoCs is presented by M. Coppola, S. Curaba, M. Grammatiakakis, G. Maruccia and F. Papariello. OCCN (On-chip Communication Network) is a SystemC based simulation environment that defines a communication API (Application Programmers Interface) for high level message passing communication. This framework allows to model, simulate and analyze applications that express their communication in terms of the communication API.

We hope this issue represents an interesting snap-shop of current NoC research activities and offers valuable results. As such, we think it constitutes another

step towards standardized platforms with an on-chip communication network and versatile communication services as central components.

Finally, we would like to thank all the authors who submitted papers for this special-issue and the many reviewers for their efforts and their excellent work.

Axel Jantsch
Johnny Öberg
Hannu Tenhunen
Guest editors