Formal Methods for Networks on Chips

Kees Goossens, Philips Research, The Netherlands

1 Introduction

Systems on a chip (SOC) are complex embedded systems consisting of many hardware and software blocks. As the complexity of SOCs grows, the focus is less on the computation, and increasingly on communication. This results in a shift from design based on platforms [13] (design templates) to design style that is *communication-centric* [21, 15]. In this new paradigm, on-chip interconnects must address both the deep-submicron challenges (managing the number of long wires, timing closure, etc.) and complexity (scalability, quality of service, etc.). *Networks on chips* (NOC) have emerged as a new type of interconnect that can solve these problems [3, 2, 12].

In this paper we introduce the Æthereal NOC [9, 17, 19, 7] as an example to identify when and where formal methods can play a rôle in this field of research. NOCs use the same basic concepts as computer networks (packets and routers), but the trade-offs that must and can be made are very different. Wires are relatively shorter, NOC resources are relatively expensive compared to the computation resources are interconnected, and the on-chip environment is more stable than off-chip (e.g. for data loss and synchronisation). As a result, many new NOC architectures have been developed.

The Æthereal NOC was one of the first to offer not only *best-effort* communication services (BE), but also (100%) *guaranteed services* (GS), in particular uncorrupted lossless ordered communication with minimum bandwidth and maximum latency. Networks in general consist of a number of routers that send packets to one another. Because the NOC is made up of a number of distributed arbiters (the routers), giving global (end-to-end) performance guarantees is challenging [22, 16]. Without going into details how this is done in Æthereal, we describe its *goals*, the *concepts* (model) underlying Æthereal, and the resulting *view for the user*. We also discuss how the concepts translate into *architectures and implementations*, and how formal methods can help here. We will structure our discussion around Figure 1.

2 Goals and Concepts

The goals of the Æthereal NOC reflect its intended use in real-time embedded systems for consumer electronics: a) predictable (hard real-time) behaviour, b) reduce time to market, c) efficient (low cost).

These goals are reflected in the concepts (arrow 1 in Figure 1): a NOC that offers guaranteed communication services (a) based on a clear analytical model, called *contention-free routing*. Guaranteed services require resource reservations for the worst case. However, by reusing unused resources for BE services (without guarantees), we achieve high utilisation [8] (c). A formal model for performance guarantees allows the construction of a *design flow* for synthesis and mapping, configuration, simulation, and performance verification based on analytical models [6]. Guaranteed service thus reduce design time (b) by automation, but more important, they make SOC design *compositional*. Because computation modules each have guaranteed services they do not interfere with one another, and hence can be designed and tested independently from one another and the NOC.

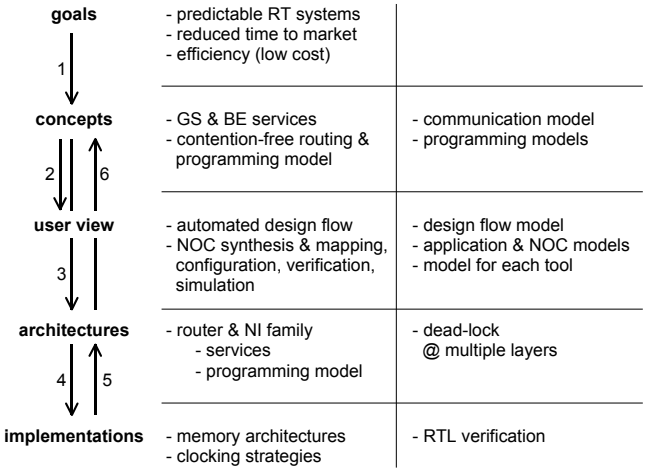

Figure 1. Overview

As mentioned in the introduction, the distributed arbitration in routers conflicts with offering guaranteed global (end-to-end) services in a NOC. Æthereal's contention-free routing offers guaranteed services to the user with a conceptually simple time-division-multiple-access (TDMA) scheme. The innovation lies in the NOC model that implements the *global* TDMA in a *distributed* manner. The NOC model uses synchronous data flow (SDF) [14, 18, 7].

The NOC must be (re)configured at run time with the

user's connections, and several models for this are supported [7]. Issues such as absence of dead-lock and termination of configuration must be addressed at several levels as we shall see in Section 4.

3 User Views

A formal model for performance guarantees allows the construction of a design flow [6] (Figure 1(2)). Embedded systems require application-specific NOCs, with hard real-time performance guarantees. To *synthesise* NOCs and to *map* computation modules to the NOC's ports, it is essential to have an analytical models of the application's required performance, and of the NOC's performance and cost (area and power dissipation) [11]. The same holds for the *configuration* of the NOC. Configured NOCs can be simulated in VHDL. SystemC simulation uses a (flit-level) abstraction of the NOC based on the SDF model. However, performance verification using simulation is per definition slow, and never exhaustive. NOCs without guaranteed services, i.e. a tractable analytical performance model, can hope for no more. But Æthereal's *performance verification* tool analytically computes worst-case throughput, latency, and buffer sizes for guaranteed services [4]. In theory NOC synthesis and configuration can be correct by construction, but we use performance verification as a redundant check. It also verifies NOCs and configurations that have been modified or created by users.

4 Architectures and Implementations

The NOC model and the architecture that implements it are obviously closely related (Figure 1(3)). For example, the clocking strategy (synchronous, GALS, or otherwise) must implement the NOC SDF model. Dead-lock is a known problem in networks, and in the absence of special buffer classes we use an extension of turn-model routing [11]. Moreover, Æthereal never drops packets, and hence each connection uses end-to-end flow control to ensure that data does not wait in the NOC and to potentially cause dead-lock. However, the NOC is used to configure itself, using connections. A configuration connection therefore can not have end-to-end flow control. A formal approach to this problem is presented in [5]. Similar work is presented in [20]. Ideally, dead-lock freedom would be proven for *any instance* of the NOC, preferably in combination with formal NOC synthesis and configuration, like for formal hardware verification [10].

Several different parametrised router architectures are available [7], and these should be shown to implement the conceptual model of Section 2 (Figure 1(4)). Model checking can be used to verify gate-level implementation versus RTL. Implementations give feasibility and cost feed-back about architectures (Figure 1(5)). Architectures do the same for NOC concepts (Figure 1(6)).

5 Conclusions

We have give some indications where formal methods can be used in the NOC research field. We are already taking steps beyond NOCs toward building predictable SOC by using data flow models for communication (NOC) and computation [1]

References

[1] M. Bekooij, O. Moreira, P. Poplavko, B. Mesman, M. Pastrnak, and J. van Meerbergen. Predictable embedded multiprocessor system design. In *Proc. Int'l Workshop on Software and Compilers for Embedded Systems (SCOPES)*, LNCS 3199. Springer, Sept. 2004.

[2] L. Benini and G. De Micheli. Networks on chips: A new SoC paradigm. *IEEE Computer*, 35(1):70–80, 2002.

[3] W. J. Dally and B. Towles. Route packets, not wires: on-chip interconnection networks. In *Proc. Design Automation Conference (DAC)*, pages 684–689, 2001.

[4] O. P. Gangwal, A. Rădulescu, K. Goossens, S. González Pestana, and E. Rijpkema. Building predictable systems on chip: An analysis of guaranteed communication in the æthereal network on chip. In P. van der Stok, editor, *Dynamic and Robust Streaming In And Between Connected Consumer-Electronics Devices*. Kluwer, 2005.

[5] B. Gebremichael, F. Vaandrager, M. Zhang, K. Goossens, E. Rijpkema, and A. Rădulescu. Deadlock prevention in Æthereal protocol. In *Submitted*, 2005.

[6] K. Goossens, J. Dielissen, O. P. Gangwal, S. González Pestana, A. Rădulescu, and E. Rijpkema. A design flow for application-specific networks on chip with guaranteed performance to accelerate SOC design and verification. In *Proc. Design, Automation and Test in Europe Conference and Exhibition (DATE)*, Mar. 2005.

[7] K. Goossens, J. Dielissen, and A. Rădulescu. The æthereal network on chip: Concepts, architectures, and implementations. *IEEE Design and Test of Computers*, Sept-Oct 2005.

[8] K. Goossens, J. Dielissen, J. van Meerbergen, P. Poplavko, A. Rădulescu, E. Rijpkema, E. Waterlander, and P. Wielage. Guaranteeing the quality of services in networks on chip. In A. Jantsch and H. Tenhunen, editors, *Networks on Chip*, chapter 4, pages 61–82. Kluwer, 2003.

[9] K. Goossens, J. van Meerbergen, A. Peeters, and P. Wielage. Networks on silicon: Combining best-effort and guaranteed services. In *Proc. Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pages 423–425, Mar. 2002.

[10] K. G. W. Goossens. Embedding a CHDDL in a proof system. In P. Prinetto and P. Camurati, editors, *Advanced Research Workshop on Correct Hardware Design Methodologies*, pages 359–374. ESPRIT CHARME, North Holland, June 1991. Also as University of Edinburgh LFCS Report ECS-LFCS-91-155.

[11] A. Hansson, K. Goossens, and A. Rădulescu. A unified approach to constrained mapping and routing on network-on-chip architectures. In *Submitted.*

[12] A. Jantsch and H. Tenhunen, editors. *Networks on Chip.* Kluwer, 2003.

[13] K. Keutzer, S. Malik, A. R. Newton, J. M. Rabaey, and A. Sangiovanni-Vincentelli. System-level design: Orthogonalization of concerns and platform-based design. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 19(12):1523–1543, 2000.

[14] E. A. Lee and D. G. Messerschmitt. Synchronous data flow. *Proceedings of the IEEE*, Sept. 1987.

[15] J. Nurmi, H. Tenhunen, J. Isoaho, and A. Jantsch, editors. *Interconnect-Centric Design for Advanced SoC and NoC.* Kluwer, 2004.

[16] J. Rexford, J. Hall, and K. G. Shin. A router architecture for real-time communication in multicomputer networks. *IEEE Transactions on Computers*, 47(10):1088–1101, Oct. 1998.

[17] E. Rijpkema, K. Goossens, A. Rădulescu, J. Dielissen, J. van Meerbergen, P. Wielage, and E. Waterlander. Trade offs in the design of a router with both guaranteed and best-effort services for networks on chip. *IEE Proceedings: Computers and Digital Technique*, 150(5):294–302, Sept. 2003.

[18] E. Rijpkema, K. Goossens, and P. Wielage. A router architecture for networks on silicon. In *Proceedings of Progress 2001, 2nd Workshop on Embedded Systems*, Veldhoven, the Netherlands, Oct. 2001.

[19] A. Rădulescu, J. Dielissen, S. González Pestana, O. P. Gangwal, E. Rijpkema, P. Wielage, and K. Goossens. An efficient on-chip network interface offering guaranteed services, shared-memory abstraction, and flexible network programming. *IEEE Transactions on CAD of Integrated Circuits and Systems*, 24(1):4–17, Jan. 2005.

[20] J. Schmaltz and D. Borrione. A functional approach to the formal specification of networks on chip. In *Proc. Int'l Conference on Formal Methods in Computer-Aided Design (FMCAD)*, volume 3312 of *Lecture Notes in Computer Science*, Nov. 2004.

[21] M. Sgroi, M. Sheets, A. Mihal, K. Keutzer, S. Malik, J. Rabaey, and A. Sangiovanni-Vincentelli. Addressing the system-on-a-chip interconnect woes through communication-based design. In *Proc. Design Automation Conference (DAC)*, pages 667–672, June 2001.

[22] H. Zhang. Service disciplines for guaranteed performance service in packet-switching networks. *Proceedings of the IEEE*, 83(10):1374–96, Oct. 1995.