A Router Architecture for Connection-Oriented Service Guarantees in the MANGO Clockless Network-on-Chip

Tobias Bjerregaard and Jens Sparsø
Informatics and Mathematical Modelling
Technical University of Denmark (DTU), 2800 Lyngby, Denmark
{tob, jsp}@imm.dtu.dk

Abstract

On-chip networks for future system-on-chip designs need simple, high performance implementations. In order to promote system-level integrity, guaranteed services (GS) need to be provided. We propose a network-on-chip (NoC) router architecture to support this, and demonstrate with a CMOS standard cell design. Our implementation is based on clockless circuit techniques, and thus inherently supports a modular, GALS-oriented design flow. Our router exploits virtual channels to provide connection-oriented GS, as well as connection-less best-effort (BE) routing. The architecture is highly flexible, in that support for different types of BE routing and GS arbitration can be easily plugged into the router.

1 Introduction

Due to the effects of scaling micro-chip technologies, it has become unavoidably necessary to differentiate between local and global on-chip communication [18]. The concept of communication-centric design derives from the fact that global communication is becomming more and more costly, in relation to processing power. Physical issues of deep submicron technologies as well as design complexity issues make current, poorly scalable solutions for global on-chip communication, such as busses, unsuited for future large scale system-on-chip (SoC) designs. There is a general consensus that shared, segmented interconnection networks, so called networks-on-chip (NoC), constitute a viable solution space to the problems faced today [4][7][10]. NoC facilitates a truely modular and scalable design approach in which IP reuse is an integral part. This is called upon in order to make efficient use of the exploding amount of resources available to chip designers, and so within the constraints of ever shortening design cycles.

Chip-wide synchronous operation is becoming prohibitively difficult to achieve in large, high performance chips [1]. Possible solutions are in the direction of the *globally asynchronous locally synchronous* (GALS) concept [13], by which a chip consists of synchronous islands which communicate asynchronously. Since a NoC spans the entire chip, it seems natural not to attempt a globally synchronous implementation but instead use a GALS approach or even a fully asynchronous, i.e. clockless, realization. This is illustrated in Figure 1. The very nature of a clockless NoC also makes the integration of cores with different timing characteristics an integral part of the design flow, promoting the concept of modularity. Clockless circuits have a number of advantages over clocked ones. Since they make use of self-timed, data-driven control they operate at the maximum speed possible, and they have zero

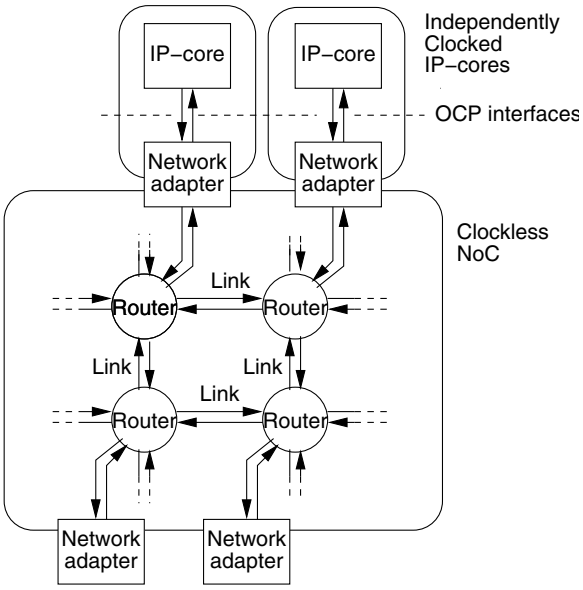

Figure 1. A shared clockless network connecting independently clocked IP cores facilitates modularity in large scale SoC designs.

Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE'05)
1530-1591/05 $ 20.00 IEEE

dynamic power consumption when idle [17]. These advantages make them ideal for implementing NoC [14].

In this paper we present a worm-hole NoC router to be used in MANGO (*Message-passing Asynchronous Network-on-Chip providing Guaranteed services through OCP interfaces*) the NoC being developed at DTU. Our implementation is based on clockless circuit techniques. To our knowledge, MANGO is the first clockless NoC to provide connection-oriented guaranteed services (GS) as well as connection-less best-effort (BE) routing. It has features, area and performance comparable to a similar class of clocked routers. Additional benefits are an inherent support for GALS systems and zero dynamic idle power. The novelty lies in the concepts of establishing the GS paths and the application of methods we have developed in [5] to implement non-blocking behaviour amongst several independent streams in the shared part of the link/router. We demonstrate with a 0.12 μm CMOS standard cell implementation.

The paper focuses more on the router architecture and less on circuit realizations. In Section 2 we provide argumentation for the implementation of GS in NoC, and in Section 3 we provide an overview of MANGO. In Sections 4 and 5 we explain the details of the router architecture, and explain how it is possible to provide GS via connections. Finally, in Section 6 we describe our implementation and give a conclusion in Section 7.

2 Background

When designing NoCs, much research done in the area of multicomputer networks can be readily build upon. The trade-offs differ however, and while pincount and global wiring have been the main limiting factors in multicomputer networks, NoC designers face constraints on area and power. Also latency requirements are more stringent in tightly coupled single-chip systems. As a result, NoC demands simple solutions to ensure high speed and low area.

Traditionally multicomputer networks support best-effort (BE) routing. Much NoC research builds on this by improving on BE routing efficiency under the constraints of single-chip system design [15][12]. In particular the use of virtual channels (VCs) – logically independent channels sharing a physical link – is employed. We propose to use the VC routing resources somewhat differently, i.e. for establishing GS connections. While an improvement of BE routing simply makes better use of existing wires, connection-oriented GS routing adds explicit new features: packets travel along virtual circuits, avoiding the mutual influence that BE packets routed on the same logical network may experience. From an application-level perspective, the predictability that GS incurs promotes system integrity, and simplifies functional verification since there is less call for taking into account the complex dynamics of

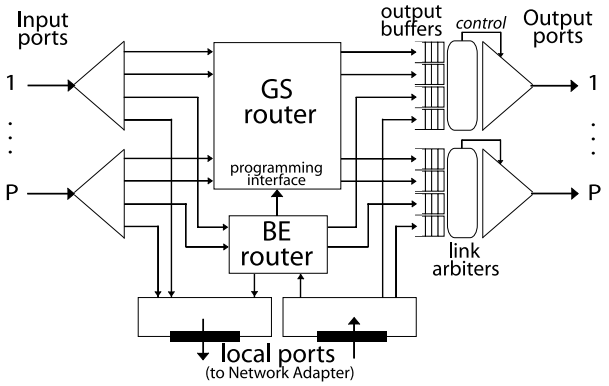

Figure 2. The MANGO router conceptually.

a shared communication structure. At the circuit level, the circuits needed to implement GS also turn out to be simpler than those needed for BE, since less arbitration is called for during routing control.

Other NoCs which implement GS are ÆTHEREAL [8] and NOSTRUM [11]. Both employ variants of time division multiplexing (TDM) for allocating bandwidth. TDM is not possible in a clockless NoC which has no notion of time, thus other approaches must be taken. In [9], a clockless NoC which provides differentiated services by prioritizing VCs is presented. Though this approach delivers improved latency for certain connections, no *hard* guarantees are provided. In [5] and [6] we have explored the use of VCs in clockless NoC for providing hard per connection bandwidth and latency guarantees respectively. In this paper we describe a supporting router architecture in which the proposed access schemes can be applied, providing end-to-end GS in a segmented network structure.

3 The MANGO Router

As shown in Figure 1, a MANGO NoC consists of network adapters (NA), routers and links. Each IP core is connected to the network through an NA, providing high level communication services, i.e. OCP transactions [2], on the basis of primitive services implemented by the network. Each NA, which also performs the synchronization between the clocked IP core and the clockless network, is connected to a router. The routers are connected by links in a grid-type structure, either homogeneous or heterogeneous. To keep speed up, long links can be implemented as pipelines.

Figure 2 shows a conceptual picture of the MANGO router. The router implements a number of uni-directional ports. Two of these are local ports which connect to the NA. The local ports consist of a number of physical interfaces. The remaining ports are network ports which, via point-to-point links, connect the router to neighboring routers. Each of these ports implement a number of independently

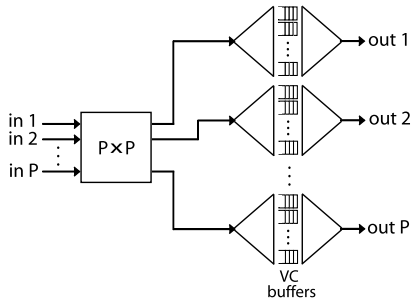

Figure 3. Generic output buffered virtual channel router architecture.

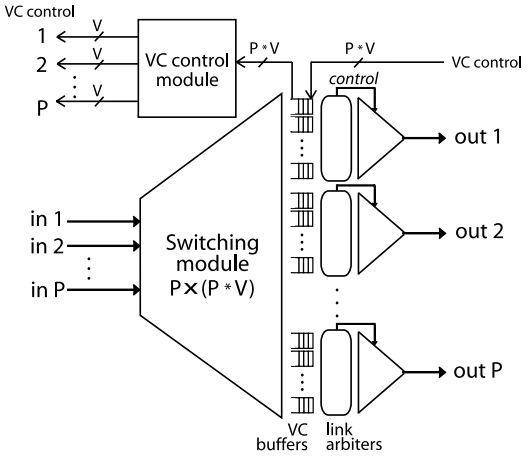

Figure 4. The MANGO GS router implements non-blocking input to output switching.

buffered VCs. Network ports and local ports implement the same type of interface. It is the function of the NA to translate communication to and from network packet format.

Internally, the router consists of a BE router, a GS router, output buffers, and link arbiters. The BE and the GS router are separately implemented. This is done in order to make the router modular. A similar approach is seen in the ÆTHEREAL router [16]. The BE router dynamically source-routes connection-less data packets, according to the routing path defined in the packet header. A subset of the VCs are allocated for BE routing. The GS router uses the remaining VCs to route header-less data streams on statically programmable connections. In order to make hard service guarantees, GS connections must be logically independent of other network traffic. In MANGO, a connection implements a logical point-to-point circuit between two different local ports in the network, by reserving a sequence of independently buffered VCs. The GS router provides non-blocking switching between the input ports and the output buffers, therefore *GS can be realized purely on the basis of link access arbitration*. This will become clear in the following sections. The GS connections are set up by programming these into the GS router via the BE router.

4 The GS Router

4.1 Overview

Figure 3 shows a generic output buffered VC router with P input and output ports; a $P \times P$ switch is followed by a split module, which splits the incoming traffic to individual VC buffers at the output ports. Since several input ports may attempt to access the same output port simultaneously, congestion may occur. This makes the architecture unsuitable for providing service guarantees.

Figure 4 illustrates our connection-oriented GS router architecture. Again P is the number of ports on each routing node, while V is the number of VCs on each port. As touched upon in Section 3, a connection is a reserved sequence of VCs through the network. In the router, incomming flits (flowcontrol units) are guided through the switching module (Section 4.2) – by way of *steering bits* which are appended to the flits in the previous router – to the VC buffer that has been reserved for the given connection.

To prevent flits of different VCs from stalling and blocking each other on the links, VC control must be employed. This ensures that flits will only be transmitted on a link, if there is free buffer space in the target VC buffer. The VC control module (Section 4.3) establishes a VC control channel from the VC buffers back to the VC buffers in the previous router; a step back on a given connection. For each connection, a router thus stores the steering bits needed to guide flits to the VC buffer reserved for the connection in the *next* router, as well as *control channel bits* used to establish a VC control channel back to the VC buffer in the *previous* router. This information is programmed into the router using BE packets (via the programming interface, Figure 2). As seen, the setup information for each hop on a connection is stored in two places: one for the flit forward path, and one for the VC control reverse path. This overhead was accepted because it facilitates some very simple circuits.

The switching module is non-blocking. This means that the latency from the time that a flit is granted access to a link, until it arrives at the designated VC buffer in the following router, is constant. In order to provide hard GS for a connection there is thus only call for link access arbitration. The link arbiter (Section 4.4) is the key element in providing GS. It arbitrates amongst the VCs contending for access to the link, implementing the type of GS that is provided.

4.2 The Switching Module

The switching module makes it possible for all input ports to route flits to any combination of VC buffers. It

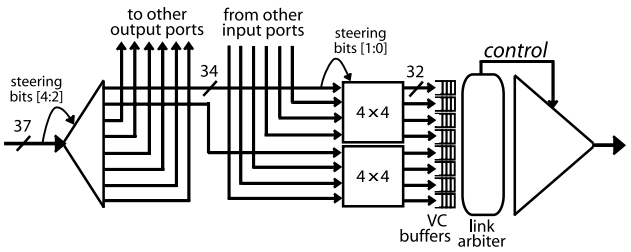

Figure 5. Input/output port pair of the switching fabric of a 5x5 GS router with 8 VCs/port.

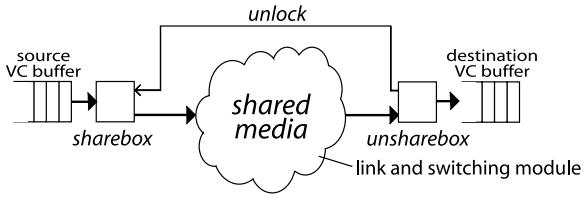

Figure 6. Share-based VC control.

is made entirely without any form of arbitration necessary, making its performance highly predictable and simplifying its design, saving both area and routing latency. Since a VC buffer is part of only one connection, no congestion will occur as only one input will attempt to route to one VC buffer at any given time. Figure 5 shows an input/output port pair in a 5×5 port router with eight VCs on each port. This configuration will form the basis for the implementation being used as an example of the architecture. The diagram indicates how the remaining ports are connected. An input port needs only connect to four output ports as it is not useful to route flits back in the direction they came from. The diagram also indicates how a total of five steering bits are used. The first three bits are used by a split module, which directs the flit to one of two 4×4 switches at each output port. At each switch two more steering bits are used to direct the flit to one of four VC buffers. At each stage, the steering bits used are stripped, as they are no longer needed.

The switching module, which consitutes a considerable part of the total router area, scales linearly with the number of VCs, and thus with the number of connections supported.

4.3 The VC Control Module

In this work we employ *share-based* VC control, which we have described in [5]. The scheme, illustrated in Figure 6, uses a single wire per VC to implement non-blocking access to a shared media, e.g. a link. When admitting a flit to the media the *sharebox* locks, not allowing further flits to pass. The flit passes across the media, to the *unsharebox* at the far side. The unsharebox implements a latch, into which the flit is accepted. When the flit in turn leaves the unsharebox, the *unlock* control wire toggles. This unlocks the sharebox, admitting another flit to the media. As long as the media is deadlock free, no flit will stall within it.

In the MANGO router, the link and the switching module constitute the shared media. Even if the link cycle for each flit transmitted on a VC is long, the full link bandwidth is exploited by the unlock handshake of different VCs overlapping. A single VC cannot utilize the full link bandwidth. However with an appropriate link access scheme, our goal to provide well defined worst case guarantees, can be met.

The cycle time of the VC link is sensitive to the forward latency of the flits. In this regard, clockless circuits are advantageous, since they make possible a very short forward latency per pipeline stage. An advantage of the share-based VC control scheme is that it is much cheaper, both area and power wise, than the commonly used credit-based VC control scheme. As shown in [5], the two schemes can be used together, controlling access to the same link. This is useful when the same link is used for both GS and BE routing. During BE routing it is clearly an advantage using credit-based VC control to improve the average case performance.

As indicated in Figure 4, each input port of the router implements an output unlock wire for each VC, likewise each output port implements an input unlock wire per VC. Under the share-based VC control scheme, the VC control module establishes control channels between arbitrary VC buffers and the inputs, simply by circuit switching the unlock wires, originating in the VC buffers, unto the VC control outputs at the input ports. Thus the VC control module implements a non-blocking $(P * V) \times (P * V)$ switch. The control channel bits stored in the router map the VC buffers to the appropriate input VC wires, according to the connection path. In our 5×5, 8 VC router we simply use $5 * 8$ instantiations of a $(5 - 1) * 8$-input multiplexer. For larger number of VCs, it might prove worthwhile to implement a more complex switch structure, e.g. a Clos network, which scales better. The mapping between input and output VCs can be considered static during connection usage.

4.4 Link Access

The MANGO router implements output buffers. Since a connection is a reserved sequence of VCs, the target VC buffer of a flit entering the router is deterministic. There is no need for arbitrating amongst VCs on different output ports. Placing the VC buffers at the output simplifies the link access arbitration scheme. Thus simple, fast and low area circuits can be used to implement access schemes. Since the path across the link, through the switching module, to the target VC buffer is fully deterministic and congestion free, by way of the non-blocking switching module described in Section 4.2, the link access is the only critical point on a connection, with regard to congestion. If guaran-

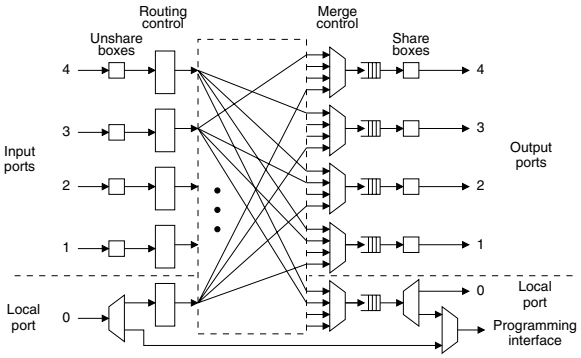

Figure 7. The BE router.

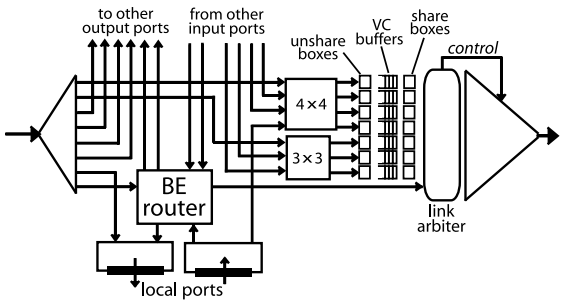

Figure 8. The BE router is integrated into the GS router, using a subset of the VCs.

tees can be made for link access, guarantees can be made for the entire connection. Thus link access arbitration is what facilitates GS in MANGO.

The *link arbiter* shown in Figure 2, 4, 5 and 8 implement the arbitration scheme. According to the arbitration amongst contending VCs, the flits are merged unto the shared link, which connects to the neighboring router. The steering bits which indicate the destination of the flit are appended before the flit is transmitted. In the router implementation presented herein, as a demonstration of the general architecture, we have implemented a fair-share access scheme [5], by which each of 8 VCs are guaranteed a minimum of one 8th of the total link bandwidth. To keep the area down, our output buffers are a single flit deep plus one flit in the unsharebox. This is enough to ensure the fair-share scheme to function over a sequence of links, providing a hard lower bound on the throughput of a connection. If a VC does not use its allocated bandwidth, the link is automatically used by another contending VC.

It is seen how the GS routing functionality is decoupled from the switching functionality in the router. This makes it an easy and modular task to instantiate new GS schemes.

5 The BE Router

The BE router, shown in Figure 7, implements a simple source routing scheme. The first flit of a packet is the header flit. The two MSBs of the header indicate one of four output ports. Choosing a direction back to where it came from, the packet is routed to the local port. The header is then rotated two bits, positioning the header bits for the next hop. With 32-bit flits, a packet can make a total of 15 hops. The packets are variable length; a control bit is used to indicate the last flit. The outputs arbitrate fairly between input ports contending for access. Once an input port has gained access, it will retain this until the last flit has been routed, thus keeping packet coherency. The interface used to program GS connections into the GS router, is implemented as an extention on port 0, the local port. Figure 8 shows a complete picture of the router, indicating how the BE router is integrated into the GS router architecture. To avoid deadlocks XY-routing is employed.

When a flit enters the BE router, three steering bits have been stripped, and a total of 34 bits remain (see Figure 5), 32 of which are flit data. Of the other two, one is the control bit used to indicate the last flit. The remaining bit can be used to indicate one of two BE VCs. This is not used in the present implementation, but can be used to extend the BE router to provide more complex deadlock free routing, adaptive VC allocation, etc. The VC control of the BE channels is handled separately from the VC control module and can be implemented using a credit-based scheme [5].

The focus of this paper is means of providing GS by way of the non-blocking GS router. The BE router has been included for the sake of completeness and holds lots of potential for improvement. First of all it should implement credit-based VC control, in order to facilitate improved average case performance. The modular architecture of the MANGO router makes it easy to plug in a new BE router.

6 Implementation and Results

We have implemented a 32-bit MANGO router using 0.12 μm CMOS standard cells. The router is designed with 4-phase bundled data clockless control circuits. It is a relatively small entity, thus internal timing is quite predictable. The links between neighboring routers are much longer, and thus more sensitive to timing variations. In order to make assembling a NoC-based SoC a modular and timing safe exercise, and in order to save power, we advocate delay insensitive signaling between routers, e.g. 1-of-4 signaling [3]. This will be realized in future MANGO versions.

The router has local input and output ports providing 5 interfaces each (4 for GS and 1 for BE) and 4 input and 4 output network ports (each implementing 8 VCs). The router simultaneously supports connection-less BE routing plus a total of 32 independently buffered GS connections. Each VC provides one 8th of the bandwidth on a link.

Table 1. Area usage in the MANGO router.

Module	Area
Connection table	0.005 mm^2
Switching module	0.065 mm^2
VC buffers	0.047 mm^2
Link access	0.022 mm^2
VC control	0.016 mm^2
BE router	0.033 mm^2
Total	**0.188 mm^2**

The performance in netlist simulations using worst-case timing parameters (1.08 V/125 C) was 515 MHz per port (795 MHz under typical timing conditions). The pre-layout area was 0.188 mm^2. The area usage is detailed in Table 1. The switching module and the VC buffers together account for more than half of the total area. Much area could be saved by using custom-designed buffers. As a quick comparison, a 0.13 μm instantiation of the globally synchronous ÆTHEREAL NoC, which also provides per connection bandwidth guarantees, had a port speed of 500 MHz and a laidout area of 0.175 mm^2 [8], using custom hardware FIFOs. The ÆTHEREAL router supports up to 256 connections. These are however not independently buffered – a key feature of the MANGO router connections. Thus end-to-end flowcontrol is needed, e.g. using credits. In the MANGO architecture end-to-end flowcontrol is inherent. Also, in order to save area, routing infomation is not stored locally in ÆTHEREAL. Connections thus need to support the routing overhead of a packet header. This is unwarranted in the architecture presented herein.

7 Conclusion

In this paper, we have presented MANGO (*Message-passing Asynchronous Network-on-Chip providing Guaranteed services through OCP interfaces*), the NoC being developed at DTU. The MANGO router is implemented using clockless circuit techniques and thus inherently supports a modular, GALS oriented design flow. It exploits virtual channels to provide connection-oriented service guarantees (GS), as well as connection-less best-effort routing. We view the predictability of GS as a way to promote system-level integrity. A 32-bit 5×5 MANGO router using 0.12 μm CMOS standard cells was implemented, which supports 32 independently buffered connections, has a worst-case port speed of 515 MHz and a pre-layout area of 0.188 mm^2.

References

[1] International technology roadmap for semiconductors (ITRS) 2001. Technical report, International Technology Roadmap for Semiconductors, 2001.

[2] Open Core Protocol Specifi cation, Release 2.0, 2003.

[3] W. Bainbridge and S. Furber. Delay insensitive system-on-chip interconnect using 1-of-4 data encoding. In *Proceedings of the 7th International Symposium on Asynchronous Circuits and Systems*, pages 118 – 126, March 2001.

[4] L. Benini and G. D. Micheli. Networks on chips: A new SoC paradigm. *IEEE Computer*, 35(1):70 – 78, January 2002.

[5] T. Bjerregaard and J. Sparsø. Virtual channel designs for guaranteeing bandwidth in asynchronous network-on-chip. In *Proceedings of the IEEE Norchip Conference*, 2004.

[6] T. Bjerregaard and J. Sparsø. A scheduling discipline for latency and bandwidth guarantees in asynchronous network-on-chip. In *Proceedings of the 11th International Symposium on Advanced Research in Asynchronous Circuits and Systems*. IEEE, 2005.

[7] W. J. Dally and B. Towles. Route packets, not wires: On-chip interconnection networks. In *Proceedings of the 38th Design Automation Conference*, pages 684–689, June 2001.

[8] J. Dielissen, A. Radulescu, K. Goossens, and E. Rijpkema. Concepts and implementation of the phillips network-on-chip. In *Proceedings of the IPSOC'03*, 2003.

[9] T. Felicijan and S. B. Furber. An asynchronous on-chip network router with quality-of-service (QoS) support. In *Proceedings IEEE International SOC Conference*, pages 274–277. IEEE, 2004.

[10] A. Jantsch and H. Tenhunen. *Networks on Chip*. Kluwer Academic Publishers, 2003.

[11] M. MillBerg, E. Nilsson, R. Thid, and A. Jantsch. Guaranteed bandwidth using looped containers in temporally disjoint networks within the Nostrum network on chip. In *Proceedings of the Design, Automation and Testing in Europe Conference (DATE'04)*. IEEE, 2004.

[12] R. Mullins, A. West, and S. Moore. Low-latency virtual-channel routers for on-chip networks. In *Proceedings of the International Symposium on Computer Architecture (ISCA) 2004*. IEEE, 2004.

[13] J. Muttersbach, T. Villiger, K. Kaeslin, N. Felber, and W. Fichtner. Globally-Asynchronous Locally-Synchronous Architectures to Simplify the Design of On-CHIP Systems. In *Proc. 12th International ASIC/SOC Conference*, pages 317–321, Sept. 1999.

[14] S. F. Nielsen and J. Sparsø. Analysis of low-power SoC interconnection networks. In *Proceedings of Nordchip 2001*, pages 77–86, 2001.

[15] L.-S. Peh and W. J. Dally. A delay model for router microarchitectures. *IEEE Micro*, 21:26 –34, 2001.

[16] E. Rijpkema, K. G. W. Goossens, A. Radulescu, J. Dielissen, J. V. 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. In *Proceedings of the Design, Automation and Test in Europe Conference (DATE'03)*, pages 350–355. IEEE, 2003.

[17] J. Sparsø and S. Furber. *Principles of Asynchronous Circuit Design - a Systems Perspective*. Kluwer Academic Publishers, Boston, 2001.

[18] D. Sylvester and K. Keutzer. A global wiring paradigm for deep submicron design. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, 19:242–252, 2000.