

Survey on Architecture for Network on Chip

Vinayakumar V Sajjan, Asst.Prof, Raji C
VLSI and Embedded System,
School of ECE,Reva University, Bangalore

Abstract—Moore's law can be applied to Multiprocessor architectures and platforms, and these architectures depend on concurrency and synchronization in both hardware and software that enhances the system performance and design productivity. These platforms are expected to present highly reusable, scalable, predictable, cost and energy-efficient architectures. As we are in Billion transistors era, the main challenges in deep sub-micron technologies which are categorized by gate lengths in the range of 60-90 nm, that leads to errors in signal integrity, non-scalable wire delay and unsynchronized communications. Many of these challenges can be address using the Network on Chip (NoC) Architecture. In this paper, we have summarized over 55 research papers and contributions in NoC domain.

Keywords—Network on chip, routing algorithms, switching techniques, QoS,buffer

I. INTRODUCTION

On a chip having billion transistors, it is challenging to send global signals across the chip in real-time bounds [1]. If we implement a System on Chip (SoC) which is synchronized by a global clock signal, the circuit is more prone to electromagnetic interference (EMI) [2]. The legacy system design or chipsets are based on clock trees and critical paths. These clock trees and critical paths lead to an increased amount of power consumption. As a result, System on Chip (SoC) are not power efficient. Also, it's challenging to manage these clock trees due to clock skew problems [3].

In comparison with synchronous designs, asynchronous designs are modular and do not suffer from issues such as higher power consumption, clock skew and EMI. There is more complexity in designing asynchronous systems as compared to designing synchronous systems [4]. Managing clock arrival time and designing a glitch free circuit are complicated in case of asynchronous system. There is not much support from Electronic Design Automation [EDA] industry for asynchronous system. Thus researchers have merged the ideas of asynchronous and synchronous designs. One such example is globally asynchronous and locally synchronous [GALS] solution. Globally asynchronous and locally synchronous divide a system into smaller units, locally decoupled synchronous regions and then composes a few among them to yield a localized subsystem. These subsystems and synchronous regions would be easier to integrate into a global solution and verify. There are asynchronous ways in which all the local synchronous regions will communicate at the system level. Hence, these synchronous regions need not be synchronized to single global clock. This approach considerably reduces requirement of chip wide clock trees. Designers can solely focus on local synchronous regions only, which is comparatively less complex than the complete system. There is a flexibility to reduce the clock speed of a given synchronous region independent of other such regions. The power consumption in the system can be reduced and can be managed better. One of the globally asynchronous and locally synchronous solution is Network on chip. NoC is designed to improve productivity by supporting modularity and reuse of complex cores. Hence, higher level of abstraction in the architectural modeling of future system is enabled.

II. LITERATURE SURVEY

A. Topology

There are various topologies for NOC architecture in terms of communication perspective. Some of these are butterfly, octagon, mesh, torus, ring and irregular interconnection networks [5], [6]. Many researchers have used these different NOC topologies for their NOC implementations. Kim et al. have used a star-based NOC that communicated using the principle of CDMA (Code Division Multiple Access) [7]; Adriahtenaina et al. proposed a tree-based implementation of NOC [8], which explains that each node of the tree behaves as a router in NOC; Pande et al. compared various network topologies for interconnection networks in terms of energy dissipation, latency and throughput [6]. Many researchers say that a 2-D mesh NOC architecture will be more efficient in terms of ease of implementation, latency and power consumption as compared to other topologies. The Octagon NOC topology is demonstrated in [9], which is an example of a novel regular NOC topology.

B. Router Architecture

Working principle of NOC architectures are based on packet-switched networks. This has led to emerging and efficient principles for design of routers for NOC [10]. Assume that the topology consists of four inputs and four outputs from/to other routers which are connected in mesh topology, and another input and output from/to the Network Interface (NI). Routers can implement various functionalities - from simple switching to intelligent routing. Since embedded systems are constrained in power consumption and area, but still they need high data rates, routers need to be designed with hardware usage in mind. Routers may be designed with no queuing (buffering) in circuit switching. Some amount of buffering is needed, to support bursty data transfers in case of packet switched networks. Such data originate in multimedia applications such as video streaming. Buffers can be present, at the output, at the input or at both input and output [11].

Different routing strategies have led to Various designs and implementations of router architectures in the literature. Circuit switched router architecture for NOC is proposed in Wolkotte et al [12], while a packet switched router architecture is proposed in Dally and Towles [13]. A wormhole based packet forwarding design for a NOC switch is proposed in Albenes and Frederico [14].

Packet switched router architecture is proposed in Dally and Towles provides better intelligence and forwards the packet based on the packet header information. It can be fine-tuned to use full bandwidth and can be designed to provide higher power efficiency and performance.

C. Routing protocol

There are various ways to classify routing algorithms as shown in figure 1. The packets are destined to single host in case of unicast routing, the packets are destined to multiple destinations in case of multicast routing. Due to the presence of point-to-point communication links among various components inside a chip, unicast routing strategies seem to be a practical approach for on chip communication. Based on the routing decision, unicast routing can be further classified into four classes: distributed routing, centralized routing, source routing and multiphase routing.

A centralized controller controls the data flow in a system in centralized router. The routing decisions are taken at the point of data generation in source routing, the routing decisions are determined as the packets/flits flow through the network in distributed routing. Multiphase routing is hybrid of source and destination routing.

Routing algorithms can also be implemented using lookup table and Finite State Machine (FSM). Implementation of lookup table routing algorithms are more popular. Lookup table based routing tale are implemented in software, where a lookup table is stored in every node. Routing algorithm can be modified by replacing the entries of the lookup table. Routing algorithms based on finite state machines may be implemented either in software or in hardware.

Based on their adaptability these routing algorithms may be further classified as deterministic routing and adaptive routing. Deterministic routing always follows a deterministic path on the network. Examples of deterministic routing algorithms are north first, south first, east first, XY routing, and west first. To avoid congested paths in the network adaptive routing algorithms need more information about the network. These routing algorithms are obviously more complex to implement, hence they are more expensive in area, cost and power consumption. Hence, we need to consider a right QoS (Quality-of-Service) metric before employing these algorithms.

Routing algorithms can be fault-tolerant algorithms such as backtracking. A channel is reserved before a flit is forwarded, in case of progressive algorithms. Some routing algorithms send packets/flits only in the direction that is nearer to the destination. These routing algorithms are referred as profitable algorithms. A misrouting algorithm may forward a packet/flit away from the destination as well. Based on the availability of routing paths, routing algorithms can be classified as partial routing algorithms and complete routing algorithms.

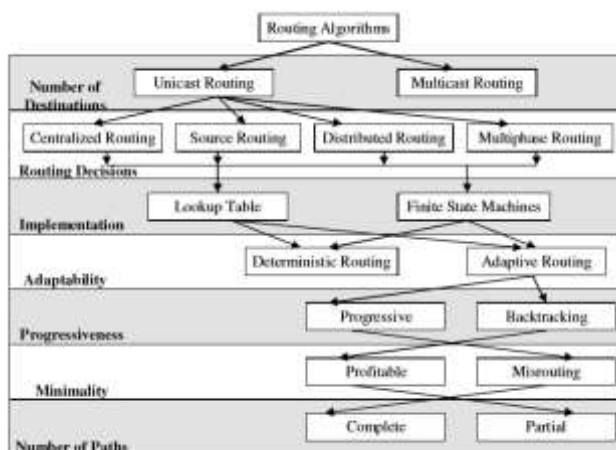


Figure 1. Routing algorithms.

NOC can be implemented using various algorithms. Many researchers prefer static routing algorithms and performed communication analysis based on the static behavior of NOC processes, thus, determining the static routing for NOC. Siebenborn et al. and Hu et al. used a CDG (Communication Dependency Graph) to analyze inter-process communications [15] [16].

XY routing or street sign routing algorithms are the most used NOC implementations. In [17], a comparison of adaptive routing algorithms and deterministic algorithms (dimension-order) for torus, mesh and cube networks was presented. Mello et al. researched the performance of minimal routing protocol in NOC [18]. They concluded that the minimal routing provided better results than adaptive routing for on-chip-communications, as the adaptive routing concentrates on the traffic in the center of the NOC.

D. Switching technique

Based on network characteristics switching techniques can be classified in to circuit switched networks and packet switched networks. A physical path is reserved before transmitting the data packets in circuit switched networks. The packets are transmitted without reserving the entire path in case of packet switched networks. Packet switched networks are classified as Virtual Cut Through Switching (VCT) , Wormhole and Store and Forward (S&F)(see Figure 2). Only the header flit experiences latency in wormhole switching networks. Other flits belonging to the same packet simply follow through the path taken by the header flit. The entire packet is blocked, if the header flit is blocked. Buffering of the packet is not required. Hence, the size of the chip drastically reduces. Major drawback of this switching technique is a higher latency. Hence, it is not a suitable switching technique for real-time data transfers. Al-Tawil et al. provided a well-structured survey of Wormhole Routing techniques and its comparison with other switching techniques [19].

A packet is forwarded only when there is enough space available in the receiving buffer to hold the entire packet in S&F switching forwards. Hence, there is no need for dividing a packet into flits. As it does not require circuits such as a flit builder, a flit sequencer, a flit decoder and a flit stripper overhead is reduced considerably. This kind of switching technique requires a large amount of buffer space at each node.

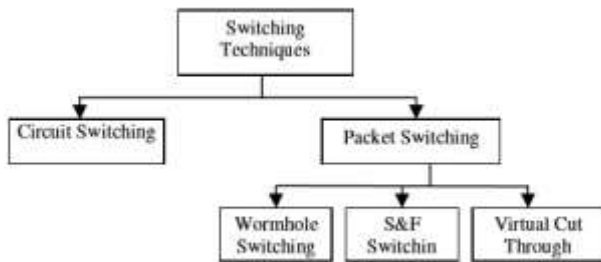


Figure 2. Switching techniques.

It may not be a feasible solution for embedded applications. Store-and-forward switching is explained in the CLICHÉ implementation of a NOC [2]. This switching technique is employed by Millberg et al. in their Nostrum NOC implementation [20]. In VCT switching, a packet is forwarded to the next router as soon as there is enough space to hold the packet. However, unlike S&F, the VCT algorithm divides a packet into flits, which may be further divided into phits. It has the similar buffer requirement as S&F. This kind of NOC implementations has not been adopted.

By combining different switching techniques, Ad-hoc switching techniques can be also developed. For instance, VCs can be used for each class of traffic, while each channel is operated as per the principles of circuit switching. The Ethereal [10], [21] and Mongo NOC implementations use such a combination of techniques [22], [23].

E. Flow Control

Flow control determines how network resources, such as control state, channel bandwidth and buffer capacity are allocated to a packet traversing the network. The flow control may be buffered or buffer less (see Figure 3).

The Bufferless Flow Control has more latency and less throughput than the Buffered Flow Control. The Buffered Flow Control is categorized into T-Error Flow Control, Credit Based Flow Control, ACK/NACK Flow Control, STALL/GO Flow Control, and Handshaking Signal based Flow Control.

In Credit Based Flow Control, an upstream node keeps count of data transfers, and thus the available free slots are termed as credits. Once the transmitted data packet is either consumed or further transmitted, a credit is sent back. Bolotin et al. used Credit Based Flow Control in QNOC [24], [25].

In case of Handshaking Signal Based Flow Control, whenever a sender transmits any flit a VALID signal is sent. The receiver acknowledges by asserting a VALID signal after consuming the data flit. Zeferino et al. used handshaking signals in their SoCIN NOC implementation [26].

In the ACK/NACK protocol a copy of a data flit is kept in a buffer until an ACK signal is received. On assertion of ACK, the flit is deleted from the buffer; instead if a NACK signal is asserted then the flit is scheduled for retransmission. Bertozzi, Benini, and Micheli used this flow control technique in their XPIPES implementation [27], [28], [29].

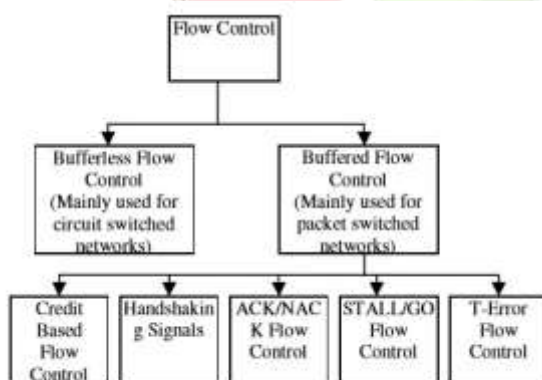


Figure 3. Flow control techniques.

In the STALL/GO scheme, two wires are used for flow control between each pair of sender (producer) and receiver (consumer). When there is an empty buffer space, a GO signal is activated. Upon the unavailability of buffer space, a STALL signal is activated. None of the present NOC implementations have employed this flow control scheme.

The T-Error Flow Control scheme is very complex as compared to other flow control mechanisms. It aims at enhancing the performance at the cost of reliability. Real time systems operating in a noisy environment must avoid the use of this flow control mechanism. None of the present NOC implementations has employed this flow control scheme.

F. Virtual Channel

Another important aspect of NOC is the design of a virtual channel (VC). A virtual channel splits a single channel into two channels, virtually providing two paths for the packets to be routed. There can be two to eight virtual channels. The use of VCs reduces the network latency at the expense of area, power consumption, and production cost of the NOC implementation. However, there are various other added advantages offered by VCs.

Network deadlock/livelock: More than one output path per channel are present in VC. There is a lesser probability of deadlock in the network. The livelock probability is eliminated (these deadlock and livelock are different from the architectural deadlock and livelock, which are due to violations in inter-process communications).

Performance improvement: A packet/flit waiting to be transmitted from an input/output port of a router/switch will have to wait if that port of the router/switch is busy. However, VCs can provide another virtual path for the packets to be transmitted through that route, thereby improving the performance of the network. Supporting guaranteed traffic: A VC may be reserved for the higher priority traffic, thereby guaranteeing the low latency for high priority data flits [30], [24].

G. Reduced wire cost

In today's technology the wire costs are almost the same as that of the gates. It is likely that in the future the cost of wires will dominate. Thus, it is important to use the wires effectively, to reduce the cost of a system. A virtual channel provides an alternative path for data traffic, thus it uses the wires more effectively for data transmission. Hence, the wire width on a system can be reduced (number of parallel wires for data transmission). For example, we may choose to use 32 bits instead of 64 bits. Therefore, the cost of the wires and the system will be reduced.

Bjerregaard and Sparso have proposed the design and implementation of a virtual channel router using asynchronous circuit techniques [23], [24].

H. Buffer implementation

A higher buffer capacity and a larger number of virtual channels in the buffer will reduce network contention, thereby reducing latency. The use of need to be carefully studied and optimized as they are area hungry. Zimmer et al. and Bolotin et al. proposed a simple implementation of a buffer architecture for NOC [31], [32]. Zimmer et al. implemented buffers using 0.18 μm technology to estimate the cost and area of buffers needed for NOC. The Proteo implementation of a buffer architecture has been described in [33]. Gupta et al. studied the trade-off between buffer size and channel bandwidth to secure constant latency. They concluded that increasing the channel bandwidth is preferable to reducing the latency in NOC.

I. Error correction and decoding

The need for implementation of fault tolerant, error detection, and error correction techniques is not certain for onchip implementations. Frederico, Santo, and Susin proposed a fault tolerant routing protocol for NOC [14]. Bolotin et al. in their implementation of QNOC [24], [25] argued that the communication strategies for on chip network may be considered reliable, while [34] and [35] proposed a fault tolerant flow control technique and fault tolerant routing algorithms for NOC architecture respectively. Zimmer, Jantasch, and Bertozzi, Binini, and Micheli proposed error detection and correction schemes for data on NOC links [36], [37].

J. Transmission lines and links

Another important component of a NOC is the design of interconnects. Barger et al. proposed a transmission line based design of interconnects for NOC [38]. Morgenshtein et al. compared serial and parallel links for interconnect implementations [39]. Parallel links are better compared to serial link. Parallel links can be used to obtain higher bandwidth by multiplexing them.

K. Network interface

The network interface (NI) is responsible for packetization and depacketization of data traffic, in addition to conventional interfacing. This functionality may be implemented either with hardware or with software. Bhojwani and Mahapatra [40] compared software and hardware implementations of NI. They showed that the software implementation of NI takes about 47 cycles to complete packetization/depacketization, while the hardware version takes only 2 cycles. Substantial research has been conducted to propose the right data formats needed for various layers in the protocol stack. Ethernet and XPIPER NOCs use the OCP protocol, while SPIN and Proteo NOC have integrated the Virtual Component Interface (VCI) protocol in their implementations.

L. QoS

New algorithms have been proposed in this domain to reduce power consumption and area requirements while securing cost optimization [41]. One of the main concerns in NOC is to be able to reduce the latency of operation. Therefore, there are various levels of latency metrics that may be offered. Router architectures for supporting GT (Guaranteed Bandwidth) and BE (Best Effort) services have been proposed [10].

M. Arbitration Techniques

A NOC, which is capable of supporting different classes of service levels such as best effort and guaranteed traffic, needs to support an arbitration mechanism. This arbitration mechanism schedules a flit for transmission on the output path. There are various arbitration mechanisms such as RR (Round Robin), FCFS (First Come First Serve), PB (Priority Based), and PRBB (Priority Based Round Robin). Usually, FCFS and RR are used for best effort data flits and PBRR or PR is used for guaranteed traffic. SPIN [8], [41] and RASoC implemented RR arbitration, while QNOC [22], [23], XPIPER [27], [28], [29], and the Philips NOC [10], [21] have employed PBRR arbitration.

N. Architectural issues

A NOC system may be categorized based on the customization and parameterization capabilities embedded in its architecture. A NOC architecture may further be defined as a homogenous architecture or a heterogeneous architecture. A heterogeneous architecture will have a fixed topology and cannot be customized as per an application requirement. Therefore, the design time with such an architecture will be less. However, a homogenous architecture may be customized each time as per the application requirement and may be more efficient in terms of area, power and latency of operation. Many NOC implementations support a homogenous architecture. XPIPES supports a heterogeneous architecture [27], [28].

III. CONCLUSION

The NOC concept elegantly separates the concerns of computing and communication, and is expected to be ideally suited to address this increased system complexity and declining system productivity. Researchers have well addressed NOC architectures and hardware-related issues. Still, an integrated approach for modeling, co-designing and co-developing HW-SW with a NOC architecture is missing. Application mapping strategies and feasible applications for NOC are other important aspects that need to be addressed in more detail. We need to research low cost, area and power efficient solutions of NOC for it to be applicable in the embedded systems industry.

REFERENCES

- [1] A. Jantsch and H. Tenhunen, *Network on Chips*, Kluwer Academic Publishers, Boston, 2003.
- [2] S. Kumar, A. Jantsch, J-P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja, and A. Hemani, "A network on chip architecture and design methodology", *IEEE Computer*, pp. 117-124, 2002.
- [3] ARTERIS. 2005. A comparison of network-on-chip and buses. White paper. <http://www.arteris.com/noc/whitepaper.pdf>.
- [4] K. Emerson, "Asynchronous design - An interesting alternative", *Proc. 10th International IEEE Conference on VLSI Design*, 1997, pp. 318-320.
- [5] J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*, Morgan Kaufmann, 2004.
- [6] P. Pratim Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh, "Performance evaluation and design trade-offs for network-on-chip interconnect architectures", *IEEE Transactions on Computers*, vol. 54, no. 8, pp. 10251040, 2005.
- [7] D. Kim, Manho Kim, and G.E. Sobelman, "CDMA-based NoC architecture", *Proc. IEEE Conference on Circuits and Systems*, vol. 1, pp. 137-140, 2004.
- [8] A. Adriahtenaina, H. Charlery, A. Greiner, L. Mortiez, and C.A. Zeferino, "SPIN: a scalable, packet switched, on-chip micro-network", *Proc. IEEE Conference on Design, Automation and Test*, pp. 70-73, 2003.
- [9] F. Karim A. Nguyen, and S. Dey, "An interconnect architecture for networking systems on chips", *IEEE Journal on Micro High Performance Interconnect*, vol. 22, issue 5, pp. 36-45, Sept 2002.
- [10] E. Rijpkema, K. Goossens, A. Radulescu, 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 Proc. on Computers and Digital Techniques*, vol. 150, Issue 5, pp. 294-302, September 2003.
- [11] A. Kumar, D. Manjunath, and J. Kuri, *Communication Networking: An Analytical Approach*, Morgan Kaufmann, 2004.
- [12] P. T. Wolkotte, G. J. M. Smit, G. K. Rauwerda, and L. T. Smit, "An energy-efficient reconfigurable circuitswitched network-on-chip", *Proc. 19th IEEE International Conference on Parallel and Distributed Processing Symposium*, pp. 155-163, 2005.
- [13] J. W. Dally and B. Towles, "Route packets, not wires: On-Chip interconnection networks", *Proc. IEEE International Conference on Design and Automation*, pp. 684-689, June 2001.
- [14] C. Albenes, Zeferino Frederico G. M. E. Santo, Altamiro Amadeu Susin, "ParlS: A parameterizable interconnect switch for Networks-on-Chips", *Proc. ACM Conference*, pp. 204-209, 2004.
- [15] A. Siebenborn, O. Bringmann, and W. Rosenstiel, "Communication analysis for network-on-chip design", *Proc. IEEE International conference on Parallel Computing in Electrical Engineering*, pp. 315-320, 2004.
- [16] J. Hu and R. Marculescu, "Energy-aware communication and task scheduling for network-on-chip architectures under real-time constraints", *Proc. IEEE Conference Design Automation and Test in Europe*, vol. 1, pp. 234239, 2004.
- [17] C. Neeb, M. Thul, and N. Andwehn "Network on-chip-centric approach to interleaving in high throughput channel decoders", *Proc. IEEE International Symposium on Circuits and Systems*, pp. 1766-1769, 2005.
- [18] F. Moraes and N. Calazan, "An infrastructure for low area overhead packet-switching network on chip", *Integration - The VLSI Journal*, vol. 38, Issue 1, pp. 69-93, October 2004.
- [19] K. M. Al-Tawil, M. Abd-El-Barr, and F. Ashraf, "A survey and comparison of wormhole routing techniques in mesh networks", *IEEE Network*, vol. 11, pp. 38-45, 1997.
- [20] M. Millberg, E. Nilsson, R. Thid, S. Kumar, and A. Jantsch. "The Nostrum backbone - A communication protocol stack for networks on chip", *Proc. IEEE International Conference on VLSI Design*, pp. 693, 2004.
- [21] K. Goossens, J. Dielissen, and A. Rădulescu, "A Ethereal network on chip: Concepts, architectures, and implementations", *IEEE Design & Test of Computers*, vol. 22, Issue 5, pp. 414-421, September 2005.
- [22] T. Bjerregaard and J. Sparso, "Virtual channel designs for guaranteeing bandwidth in asynchronous Networkon-Chip", *Proc. of IEEE Norchip Conference*, pp. 269 - 272, November 2004.
- [23] T. Bjerregaard and J. Sparso, "A router architecture for connection-oriented service guarantees in the MANGO clockless Network-on-Chip", *Proc. Of IEEE on Design Automation and Test*, vol. 2, pp. 1226-1231, 2005.
- [24] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "QNoC: QoS architecture and design process for network on chip", *Journal of Systems Architecture*, Volume 50, Issue 2-3 (Special Issue on Network on Chip), pp. 105-128, February 2004.
- [25] E. Bolotin, I. Cidon, R. Ginosar and A. Kolodny, "Cost considerations in Network on Chip", *Integration: The VLSI Journal*, no. 38, 2004, pp. 19-42.

- [26] C. A. Zeferino and A. A. Susin, "SoCIN: A parametric and scalable network-on-chip", Proc. 16th Symposium on Integrated Circuits and Systems Design, pp. 169-175, 2003.
- [27] D. Bertozzi and L. Benini, "Xpipes: A network-on-chip architecture for gigascale systems-on-chip", IEEE Circuits and Systems Magazine, vol. 4, Issue 2, pp. 18-31, 2004.
- [28] D. Bertozzi, A. Jalabert, S. Murali, R. Tamhankar, S. Stergiou, L. Benini, and G. De Micheli, "NoC synthesis flow for customized domain specific multiprocessor systems-on-chip", IEEE Transactions on Parallel and Distributed Systems, vol. 16, no. 2, pp. 113-129, 2005.
- [29] M. Dall'Ossa, G. Biccari, L. Giovannini, L. D. Bertozzi, and L. Benini, "XPIPES: A latency insensitive parameterized network-on-chip architecture for multiprocessor SoCs", Proc. 21st International IEEE Conference on Computer Design, pp. 536-539, 2003.
- [30] E. Beigne, F. Clermidy, P. Vivet, A. Clouard, and M. Renaudin, "An asynchronous NOC architecture providing low latency service and its multi-level design framework", Proc. 11th International Symposium on Asynchronous Circuits and Systems (ASYNC), pp. 54-63, 2005.
- [31] E. Bolotin, A. Morgenshtein, I. Cidon, R. Ginosar, and A. Kolodny, "Automatic hardware-efficient SoC integration by QoS Network-on-Chip", Proc. 11th International IEEE Conference on Electronics, Circuits and Systems, pp. 479-482, 2004.
- [32] H. Zimmer, S. Zink, T. Hollstein, and M. Glesner, "Buffer-architecture exploration for routers in a hierarchical network-on-chip", Proc. 19th IEEE International Symposium on Parallel and Distributed Processing, pp., 1-4, April 2005.
- [33] I. Saastamoinen, M. Alho, and J. Nurmi, "Buffer implementation for Proteo network-on-chip", International IEEE Proceeding on Circuits and Systems, vol. 2, pp. 113-116, May 2003.
- [34] M. Pirretti, G. M. Link, R. R. Brooks, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin, "Fault tolerant algorithms for network-on-chip interconnect", Proc. IEEE Proceeding on Computer Society, pp. 46-51, February 2004.
- [35] A. Pullini, Federico Angiolini, D. Bertozzi, and L. Benini, "Fault tolerance overhead in network-on-chip flow control schemes", Proc. ACM Conference, pp. 224-229, 2005.
- [36] H. Zimmer and A. Jantsch, "A fault model notation and error-control scheme for switch-to-switch buses in a network-on-chip", Proc. First International IEEE/ACM/IFIP Conference on Hardware/Software Codesign and System Synthesis, pp. 188-193, 2003.
- [37] D. Bertozzi, L. Benini, and G. De Micheli, "Error control schemes for on-chip communication links: the energyreliability tradeoff", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 24, no. 6, pp. 818-831, 2005.
- [38] A. Barger, D. Goren, and A. Kolodny, "Design and modeling of network on chip interconnects using transmission lines", Proc. 11th IEEE International Conference on Electronics, Circuits and Systems, pp. 403406, 2004.
- [39] A. Morgenshtein, I. Cidon, A. Kolodny, and R. Ginosar, "Comparative analysis of serial vs parallel links in NoC", Proc. IEEE International Conference on System-on-Chip, pp. 185-188, November 2004.
- [40] P. Bhojwani and R. Mahapatra, "Interfacing cores with on-chip packet-switched networks", Proc. 16th International IEEE Conference on VLSI Design, pp. 382-387, 2003.
- [41] P. Bhojwani, R. Mahapatra, J. K. Eun, and T. Chen, "A heuristic for peak power constrained design of networkon-chip (NoC) based multimode systems", Proc. IEEE International Conference on VLSI Design, pp. 124-129, 2005.
- [42] A. Agarwal and R. Shankar, "A layered architecture for NOC design methodology", IASTED International Conference on Parallel and Distributed Computing and Systems, pp. 659-666, 2005.
- [43] U. Y. Ogras and R. Marculescu, "Energy- and performance-driven NoC communication architecture synthesis using a decomposition approach", Proc. IEEE Conference and Exhibition on Design, Automation and Test in Europe, vol. 1, pp. 352-357, 2005.
- [44] J. Madsen, S. Mahadevan, K. Virk, and M. Gonzalez, "Network-on-chip modeling for system-level multiprocessor simulation", Proc. IEEE 14th Conference on Real-Time Systems, pp. 265-274, 2003.
- [45] L. Tang and S. Kumar, "Algorithms and tools for network on chip based system design", 16th IEEE Proc. on Integrated Circuits and Systems Design, pp. 163-168, Sept. 2003.
- [46] K. Srinivasan, K. S. Chatha, and G. Konjevod, "Linear programming based techniques for synthesis of networkon-chip architectures", Proc. IEEE International Conference on Computer Design: VLSI in Computers and Processors, pp. 422-429, 2004.
- [47] R. Chae-Eun, J. Han-You, and Ha Soonhoi, "Many-to-many core-switch mapping in 2-D mesh NoC architectures", Proc. IEEE International Conference on Computer Design: VLSI in Computers and Processors, pp. 438-443, Oct. 2004.
- [48] T. Theocharides, G. Link, N. Vijaykrishnan, and M. J. Irwin, "Implementing LDPC decoding on network-on-chip", Proc. 18th IEEE International Conference on VLSI Design, pp. 134-137, 2005.
- [49] J. Xu, W. Wolf, J. Henkel, S. Chakradhar, and T. Lv. "A case study in networks-on-chip design for embedded video", IEEE Proc. on Design Automation and Test in Europe Conference, vol. 2, pp. 770-775, February 2004.
- [50] N. Genko, D. Atienza, G. De Micheli, J. M. Mendias, R. Hermida, and F. Catthoor, "A complete network-onchip emulation framework", Proc. European IEEE Conference and Exhibition on Design, Automation and Test, vol. 1, pp.246-251, 2005.
- [51] J. Liu, L.-R. Zheng, and H. Tenhunen, "A guaranteed-throughput switch for network-on-chip", Proc. the International Symposium on System-on-Chip, pp. 31-34, 2003.
- [52] S. J. Lee, K. Lee, S. J. Song, and H.-J. Yoo, "Packet-switched on-chip interconnection network for system-onchip applications", IEEE Trans. on Circuits and Systems-II, vol. 52, no. 6, pp. 308-312, 2005.
- [53] M. D. Harmanci, N. P. Escudero, Y. Leblebici, and P. Ienne, "Providing QoS to connection-less packet-switched NoC by implementing DiffServ functionalities", Proc. 2004 International Symposium on System-on-Chip, pp. 3740, 2004.
- [54] M. D. Harmanci, N. P. Escudero, Y. Leblebici, and P. Ienne, "Quantitative modeling and comparison of communication schemes to guarantee quality-of-service in networks-on-chip", Proc. 2005 International Symposium on Circuits and Systems, vol. 2, pp. 1782-1785, 2005.
- [55] D. Wiklund, L. Dake, "SoCBUS: Switched Network on Chip for Hard real Time Embedded Systems", IEEE International Proceedings of Parallel and Distributed Processing symposium, pp. 1-8, April 2003.
- [56] C. A. Zeferino, M. E. Kreutz, and A. A. Susin, "RASoC: A router soft-core for networks-on-chip", Proc. Design Automation and Test in Europe Conference, vol. 3, pp. 198-203, 2004.

- [57] C. A. Zeferino, M. E. Santo, and A. A. Susin, "ParIS: A parameterizable interconnect switch for networks-onchip", Proc. 17th Symposium on Integrated Circuits and Systems Design (SBCCI'04), pp. 204-209, 2004.
- [58] A. Morgenshtein, I. Cidon, A. Kolodny, and R. Ginosar, "Low-leakage repeaters for NoC interconnects", Proc. IEEE International Symposium on Circuits and Systems, ISCAS 2005, vol. 1, pp. 600-603.
- [59] N. Chabini and W. Wolf, "Reducing dynamic power consumption in synchronous sequential digital designs using retiming and supply voltage scaling", IEEE Transactions on VLSI Systems, vol. 12, no. 6, pp. 573-589, 2004.
- [60] L. Yan, L. Jiong, and N. K. Jha, "Joint dynamic voltage scaling and adaptive body biasing for heterogeneous distributed real-time embedded systems", IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, vol. 24, no. 7, pp.1030-1041, 2005.
- [61] L. Yan, J. Luo, and N. K. Jha, "Combined dynamic voltage scaling and adaptive body biasing for heterogeneous distributed real-time embedded systems", Proc. IEEE International Conference on Computer Aided Design, pp. 30-37, 2003.
- [62] C. G. Lyuh and K. Taewhan, "Low power bus encoding with crosstalk delay elimination [SoC]", 15th Annual Proc. IEEE International Conference on ASIC/SOC, pp. 389-393, September 2002.

