International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:3, No:4, 2009

Pipelined Control-Path Effects on Area and Performance of a Wormhole-Switched Network-on-Chip

Faizal A. Samman, Thomas Hollstein and Manfred Glesner

Abstract—This paper presents design trade-off and performance impacts of the amount of pipeline phase of control path signals in a wormhole-switched network-on-chip (NoC). The numbers of the pipeline phase of the control path vary between two- and one-cycle pipeline phase. The control paths consist of the routing request paths for output selection and the arbitration paths for input selection. Data communications between on-chip routers are implemented synchronously and for quality of service, the inter-router data transports are controlled by using a link-level congestion control to avoid lose of data because of an overflow. The trade-off between the area (logic cell area) and the performance (bandwidth gain) of two proposed NoC router microarchitectures are presented in this paper. The performance evaluation is made by using a traffic scenario with different number of workloads under 2D mesh NoC topology using a static routing algorithm. By using a 130-nm CMOS standard-cell technology, our NoC routers can be clocked at 1 GHz, resulting in a high speed network link and high router bandwidth capacity of about 320 Gbit/s. Based on our experiments, the amount of control path pipeline stages gives more significant impact on the NoC performance than the impact on the logic area of the NoC router.

Keywords—Network-on-Chip, Synchronous Parallel Pipeline, Router Architecture, Wormhole Switching

I. Introduction

Networks-on-Chips (NoC) is a bridge concept from Systems-on-Chip (SoCs) into Multiprocessor System-on-Chip (MPSoC). A SoC design approach uses sometimes more than one processing element (PE) to implement an integrated circuit for a certain system application. The PEs send messages to other PEs for sharing computational processes to complete tasks. A sophisticated communication structure is needed for inter-processor data exchange. Rather than using a bus for single communication among PEs, or using point-to-point communication, a concept of shared segmented communication infrastructures is proposed to support application-scalability.

Research studies with the Arteris NoC [1] have demonstrated the feasibility and advantages of NOCs over traditional bus-based architecture but have not focused on compatible communication standards. Computer bus-based communication architectures do not easily handle the real-time data flows associated with networking, telecommunication, and multimedia data streams [2]. On-chip networks will meet the distinctive challenges of providing functionally correct and reliable operation of interacting system-on-chip components [3].

The effectiveness of NoC platforms for MPSoCs is more and more significant when a huge number of PEs is used in the MPSoC. Therefore, NoCs enable promising concepts for the design of supercomputers with multiprocessor cores. NoCs have also potential applications in integrated control systems, e.g. in automotive electronic control and entertainment systems. The NoC concept has potential to provide sustainable platforms and proposes a new paradigm in SoC architecture and multiprocessor systems [4].

F. A. Samman, T. Hollstein and M. Glesner are with Institute of Microelectronic Systems, Technische Universität Darmstadt, Karlstr. 15, 64283 Darmstadt, Germany. E-mail: [faizal.samman, thomas.hollstein, glesner]@mes.tu-darmstadt.de.
F. A. Samman is also with Dept. of Electrical Engineering, Hasanuddin University at Makassar. Jl. Perintis Kemerdekaan km. 10, Makassar 90245, Indonesia [E-mail: faizalas@unhas.ac.id]. He is currently pursuing Doctoral Degree with DAAD (*Deutscher Akademischer Austausch-Dienst*) Scholarship Award at Technische Universität Darmstadt, Germany.

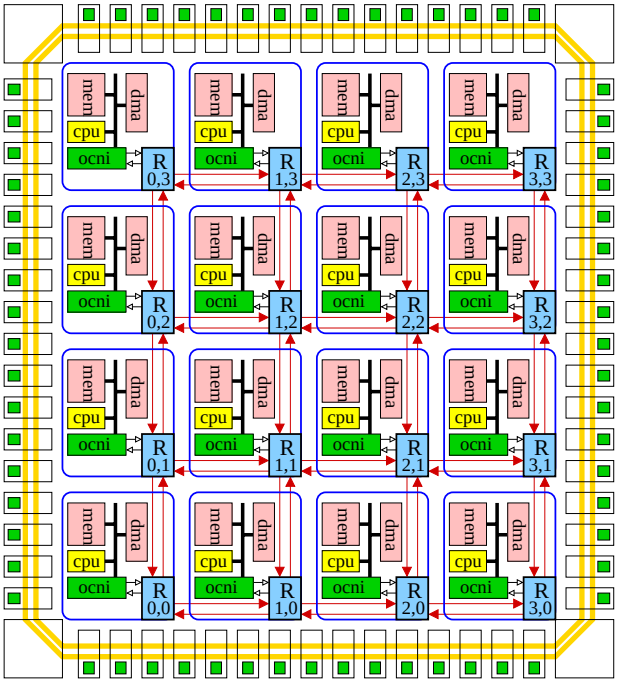


Fig. 1. Chip multiprocessor (CMP) system on mesh-connected NoC.

An example of a NoC-based (networked) CMP system is presented in Fig. 1. The chip consists of 16 tiles in a 2D 4x4 mesh topology. Each tile consists of a bus-based microprocessor system, an on-chip network interface (OCNI) and a router. The bus-based microprocessor system can comprise one or two CPU (central processing unit) blocks, a memory block, a direct memory access (DMA) controller, and/or other componnents. The OCNI is a component that assembly a data to be a packet before the data is sent to another tile, and disassembly the packet to be a data before the data is sent to the bus-based microprocessor system.

The Network topology could influence the scalability and performance of the NoC. The architecture and routing decision must meet bandwidth requirements and should be scalable for wide range of applications. Some NoCs that have been developed with Mesh topology are NOSTRUM [5], SoCBUS [6], RAW [7], PNoC [8], CLICHÉ [9] and HiNoC [10]. OCTAGON NoC [11] uses octagon topology. Fat tree topology is used in SPIN [12], and its extended version DSPIN [13] uses mesh distribution of clusters. Some NoC prototypes use also a NoC router with N number of ports to develop an irregular NoC topology e.g. by Æthereal NoC [14]. The other examples are a flexible regular and irregular topology presented in [15], and Xpipes NoC [16] that supports a customized topology. ASNoC [17] is an application specific NoC, where the design methodology supports the development of NoCs with 2-D mesh and hierarchical irregular topologies.

Inter switch/router data communication can be made also synchronously. NoC prototypes that use asynchronous data communication are e.g. CHAIN [18], ASPIDA [19], MESCAL [20], PROTEO [21], and ANoC [22]. MANGO [23] exhibits even an asynchronous clock-less. In synchronous designs, global clock-trees are distributed, which leads to electromagnetic interference effect and clock power consumption. Asynchronous communication design is a promising concept, but lacks of industrial standard tools support, especially with respect to testability issues.

This paper is organized in the following sections. Section II shows the related works and motivation of this work. Section III presents the two proposed router architecture for the comparative study. The feature, microarchitecture and characteristics of the NoC are presented in Section IV. Performance evaluations of the two proposed router architecture are presented in Section V. The logic cell area consumptions after gate-level synthesis using a logic synthesis tool are exhibited in Section VI. Section VII concludes the work by giving a brief description about the area and performance trade-off of the two proposed architecture, as well as the next future of the work.

II. Related Works and Motivation

NoC architectures with inter-switch synchronous data communication have been presented by some NoC proposals e.g. by NOSTRUM [5], RAW [7], etc. Inter-switch data communication can also be made by using asynchronous technique. Before the data are transmitted from one link to another link, then they must be propagated firstly through internal data paths of the switch (router). The latency of the data propagation in the router is strongly dependent on the number of data path and control path pipelines, as well as the maximum data frequency of the router. Both aspects are related to the microarchitecture of the router, and the feature size of the used technology.

NoCs with TDM-based data transmission and scheduling use a circuit switching method which is naturally representing a synchronous pipeline technique. This kind of data switching is used e.g. by Æthereal NoC [14] and PNoC [8]. In the circuit switching technique, inter- and intra switch data transmission can be controlled using a handshaking mechanism, a link-level control mechanism or the data is scheduled based on a time-division multiple access (TDMA) mechanism.

In the Æthereal, a data can be switched in the next cycle after the data has been buffered in an input Queue. This technique can be used only if we can make sure that there will be no contention between data that occurs during data communication by using the TDMA mechanism. However, in order to guarantee the prerequisite condition, a hard effort must be done in which a global network view is required to develop a global network flow control. The required condition can be formally found when the network flow control is develop at design time. But, at runtime, the effort to realize the contention-free condition is much harder.

The PNoC uses a handshaking mechanism to transfer data from the input queue of a router to the next queue of a next router. The data propagation in the internal data path and the link of the router is delayed for a few cycle, because the master which sends a data must wait for a few cycle to read the condition of the transmission and acception validation signals from a slave which will receive the data. This work uses the circuit switching technique but it is complemented with a mechanism to control link level data transmission.

The work in [24] presents a low latency router that uses a two-stage pipeline technique incorporating a look ahead routing and speculative allocation. In the first stage, the router performs the look ahead routing decision for the next hop, pre-selection of an optimal channel for the incoming packet header. The actual flit transfer is essentially split in two stages i.e., the preliminary semi-switch traversal through the virtual channel selection and the decomposed crossbar traversal. However, the 200 MHz data frequency of the router is lower than our NoC router. Hence, the maximum bandwidth capacity of the link and the NoC router is certainly lower than our NoC proposal.

We cannot present many examples of the data propagation in the internal data paths of the router, because most of the NoC proposals do not present it in detail. This paper presents not only the flexibility of our NoC to make a runtime interconnect configuration, but also presents the impacts of the data and control path signals over the performance and logic cell area of a wormhole-switched network on chip. Our NoC router uses a simple link-level control for inter-switch data transmission and a simple request–acknowledge signals to control data flow in the internal paths of the router. There is no other work that investigates the trade-off between the performance and area of the NoC router with different numbers of pipeline stages of its data and control paths. We realize that our experiments are contrained only for our router architecture. But, at least, this paper will exhibit the results to readers the importance of considering the propagation of data in the internal router to fulfil the bandwidth requirements.

III. Architectural Schemes for Comparative Study

Fig. 2(a) presents the architecture of a router architecture with 2-cycle control path pipeline. As presented in the figure, signal paths in crossbar area of the router are divided into data paths and control paths. The data paths are signals presented in the figure connecting the output ports of the *Queue* (Q) modules and the input ports of the MIM (*Multiplexor with ID-Management Unit*) components. The Routing request (rr) control path signals are signals connecting the output ports of the *Routing Engine* (RE) components and the input ports of the $Arbiter$ (A) components. The other control path signals are the routing acknowledge (ra) signals connecting the output ports of the $Arbiter$ units to the input ports of the *Grant-Routing-Acknowledge* (G) units. For the sake of simplicity, the complete crossbar interconnected signals are not presented in the figure. For area efficiency purpose, the depth of FIFO queue is only 2 registers.

Timing diagram of the inter-router data transfer of the router architecture is presented in Fig. 2(b). Data transmission from a link (in cycle phase 1) at an input side (West input) of the router to another link (in cycle phase 4) at the output side (East output) requires four clock-cycle periods. Data outgoing from the $Queue$ to the output port is delayed for two cycle periods. During these phases, the control signals (routing-request and arbitration signals) are pipelined in two stages. In the other words, the data flit is waiting for routing phase (phase 2) made by the RE module and selection phase (phase 3) made by the $Arbiter$ unit. Hence, we call the router prototype presented in Fig. 2 as the *2-cycle control-path pipeline* router architecture.

As an alternative architecture, a router with 1-cycle control path pipeline is presented in Fig. 3(a). The main difference of the figure with Fig. 2 is exhibited clearly. A new data path pipeline (register) is given between the data output of the queue and the MIM module. We rename the RE module to be REB (*Routing Engine with a Data Buffer*). A data from the $Queue$ module is stored in the cycle in a register of the REB. At the same cycle, the routing mechanism is made in the REB module. The routing direction is then forwarded to the $Arbiter$ module. Therefore, in this new architecture, each data will be delayed for only one cycle period before it is sent to the outgoing link. During this single cycle, the MIM is waiting for an output selection (arbitration, in phase 3) signal made by the $Arbiter$.

The timing diagram of the alternative router architecture is presented in Fig. 3(b). If we compare the timing diagram with the

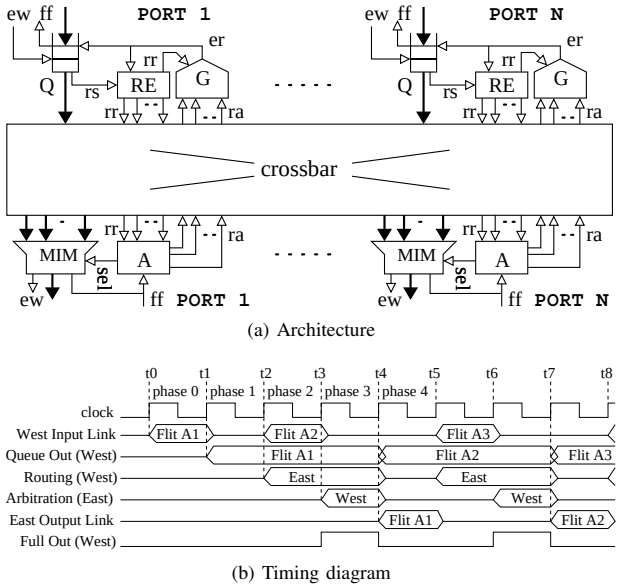


Fig. 2. Architecture of the 2-cycle control path pipeline router and its timing diagram.

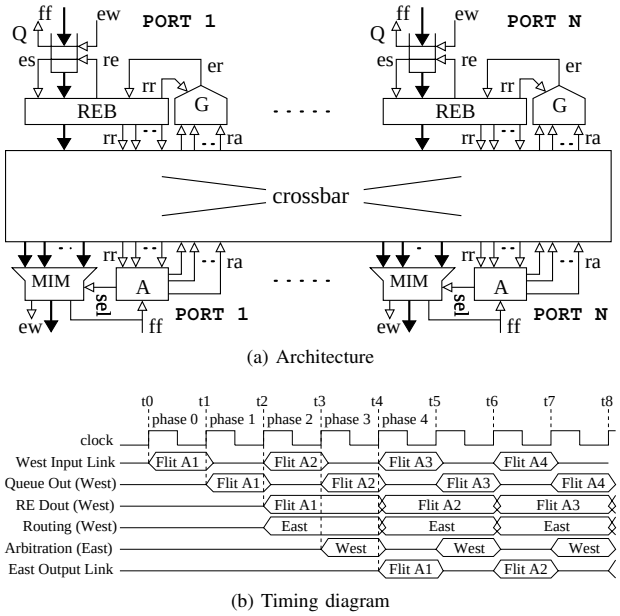


Fig. 3. Architecture of the 1-cycle control path pipeline router and its timing diagram.

aforementioned diagram (Fig. 2(b)), then it seems that the latency to switch a new data to the outgoing link can be reduced from two to one cycle. We can see also in Fig. 2(b) that a flit coming to input link in every two-cycle period will also experience an additional one cycle latency. The Flit $A1$ in phase 1 is stored in $Reg(0)$ of the $Queue$. When Flit $A2$ comes to West input port in phase 2, it is then stored in $Reg(1)$ of the $Queue$ in the next cycle (phase 3). Thus, the full flag of the $Queue$ is set in the phase 3, because Flit $A1$ is still in the $Queue$ waiting for a grant acknowledge signal from the $Arbiter$ at the requested output port.

We call the alternative architecture as the *one-cycle control path pipeline* router architecture. The alternative architecture can increase the bandwidth capacity of the router communication channels, because the router can upload a data flit in every (maximum) 2-cycle period onto the NoC links. We can observe again the East output Link Signal presented in Fig. 2(b) and in Fig. 3(b). However, the logic area of this router will increase because of the additional register in the routing engine modules. The trade-off between performance and logic area of both router architectures will be discussed in Section V and VI.

IV. Router Characteristics and Microarchitecture

A. Packet Format and Routing Mechanism

1) Packet Format: The packet format used in the NoC is presented in Fig. 4. The 39-bit packet consists of a header flit followed by payload flits. The $32^{nd} - 1^{st}$ bit are the data word. The $36^{th} - 33^{rd}$ bits represent the ID-tag (Identity) bits, while the $39^{th} - 37^{th}$ bits represent the type of the flits. The flit type can be a header, a data body, and the end of databody (the tail flit). A header flit contains source and target addresses of the message.

The 3D source and target address of the packet are asserted in the header flit. The $Z-$address is reserved for further development of a 3D NoC topology or hierarchical NoC. Each message is associated as single packet even if the size of the message is very large. It means that each unicast message will have only one header flit to find routing directions on each intermediate nodes. The payload data

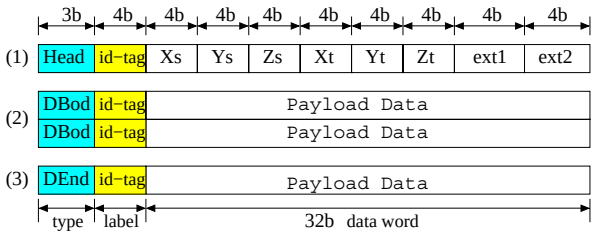


Fig. 4. Packet format.

will then track the path set up by the header flit. Therefore, the terms "message" and "packet" used this paper have similar interpretation.

The tail flit of each message has functionality to close the link reservation. Flits belonging to the same message will always have the same local identity number (ID-tag) to differentiate it from flits of other messages, when it passes through every communication link of the NoC. The ID-tag of each message will vary over different communication links in order to provide a flexible link setup and wire (link) sharing mechanism at runtime mode.

2) ID-based Routing Mechanism: Messages or packets in the NoC is routed based ID-tag. Each flit of the messages has a local ID-tag present on the $36^{th} - 33^{rd}$. Flit belonging to the same message will always have a similar ID-tag on every communication resource. The local ID-tag is updated to manage ID slot allocation for messages which share the communication link and to allow message interleaving on the link. The left side of Fig. 5 presents how three messages can be interleaved in the same link based on the ID-tag identification process of the messages.

The RE or REB module consists of the routing table and the routing state machine. The routing table consists of M numbers of register that is similar to the number of available ID slots on each link of the NoC routers. The right side of the Fig. 5 presents that the routing table of NoC router consists of 15 ID slots.

When a header flit comes from an incoming port with ID-tag h and appears on the output of the $Queue$ module, then the

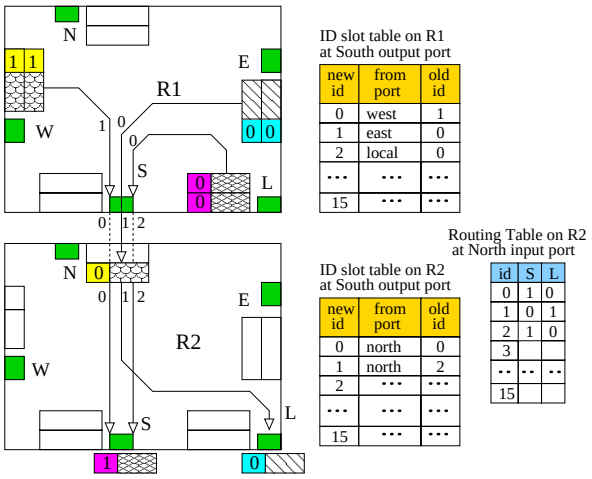


Fig. 5. ID-based routing organization.

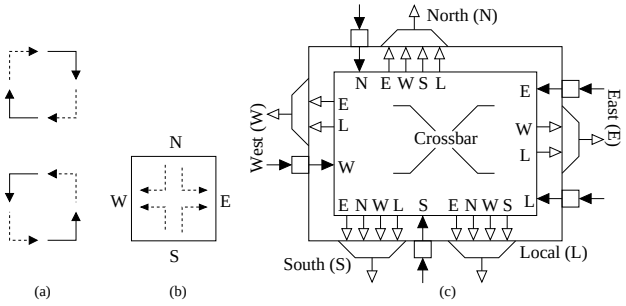


Fig. 6. (a) Turn models for static XY routing, (b) prohibited turns in the switch, and (c) crossbar connections of the mesh router.

Routing Engine computes a routing direction of the header flit. The routing direction is then stored in the routing table, exactly in the register number h in accordance with its ID-tag. In other words, the routing direction made the *Routing Engine* module is compressed, and the key used by databody or tail flits to open the compressed routing direction is the ID-tag. Therefore, when payload flits come from the incoming port also with ID-tag h (belonging to the same message with the header flit), then the routing direction will be open from the routing table using the ID-tag key of the payload flit. Based on this routing technique, the payload flits can track the correct paths even when the flits are interleaved in the same queue with other flits of different messages. The following subsection will present how the local ID tag is updated and organized to enable the wire sharing methodology.

3) Local ID-tag Organization: The ID-tag management unit is implemented in the MIM component of the NoC router. The unit is in charge of organizing, updating, and mapping the old local ID-tag of a message to the new ID-tag before the message enters an outgoing link. Fig. 5 shows how the ID-tags of three messages is updated by the ID management unit at South output port of the $R1$ router. The messages come from the East, West and Local ports with ID-tag number 0, 1 and 0, respectively. The ID slot table at the South output port of the $R1$ router is presented in the figure.

When a header flit with ID-tag h is switched from input port P_i to an outgoing port, then at the same cycle, its ID-tag will be updated. The ID management will look for a free local ID tag. When a free ID tag j has been found, then this local ID-tag j is assigned to the header flit. The slot number j of the slot table is indexed based on two local information i.e., the old local ID-tag h and the input port P_i. The ID-tag of payload flit is updated by the ID management unit by identifying the old local ID-tag and from input port the flits come. Therefore, flits belonging to the same message will always have the same local ID-tag on each outgoing port of the NoC routers. Alone with the ID-based routing mechanism, this ID-updating mechanism supports the flexible wire through-share methodology.

B. Routing Algorithm and Crossbar Switch Structure

The router prototypes implemented in this paper use a static XY or X-first routing algorithm. A message is routed firstly to X-coordinate direction (East or West) if $x_{offset} = |x_{target} - x_{source}| > 0$, and then to Y-coordinate direction (North or South) if $y_{offset} = |y_{target} - y_{source}| > 0$. Fig. 6(a) presents the turn model of the static XY routing algorithm for clockwise (upper) and counter-clockwise (bottom) turn models. The dashed lines and the solid lines represent the prohibited and the allowed turns. If the prohibited turns are implemented on the router as presented in Fig. 6(b), then it looks that the turns North-to-East, North-to-West, South-to-East and South-to-West are prohibited. Fig. 6(c) shows the crossbar interconnet structure of the switch/router based on the static XY routing algorithm.

C. Simultaneous Parallel Crossbar Connection

Our NoC can switch maximum N simultaneous crossbar interconnects in parallel. The N number depends on the number of I/O pairs in the router. This features is certainly not a new contribution in the NoC research area. However, we will present in this section, how the NoC performs such advantageous feature of a modern NoC router design with high bandwidth capacity. In line with the timing diagrams presented in Fig. 2(b) and in Fig. 3(b), three phases are needed to switch a data flit to an outgoing link after the flit has been buffered in an input queue. Fig. 7 presents graphical views (left side) and structural views (right side) of a five-simultaneous crossbar interconnection The five simultaneous connections are

1) Routing-Request Phase: After the flit is buffered and appears at the output port of the $Queue$, then in the next cycle phase, the RE module computes routing request bit signals and sends the signals to an $Arbiter$ unit at the requested output port. Fig. 7(a) shows the graphical view and structural view of this phase.

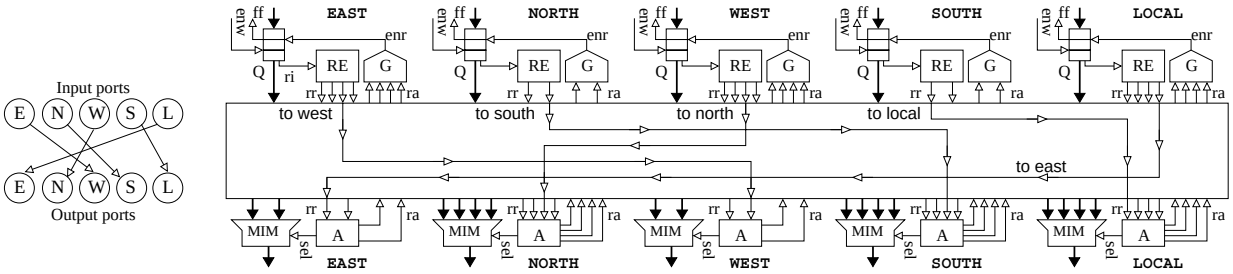
2) Request-Acknowledge Phase: When the $Arbiter$ units at the requested output ports detect the routing request signals, then each $Arbiter$ sends back a grant or routing acknowledge signal to the RE component. At the same cycle the $Arbiter$ unit sends also a selection signal to the MIM component. This phase is presented by Fig. 7(b).

3) Output-Switching Phase: The $Arbiter$ responses the routing request signals by sending two signals as mentioned in phase IV-C2. The output selection signal determines a data from an input port that will be switched out to the outgoing port, and the routing acknowledge signal enables concurrently to read the data from the input port. Hence, in the next cycle, the considered data will be switched out from input ports to output ports as presented in Fig. 7(c).

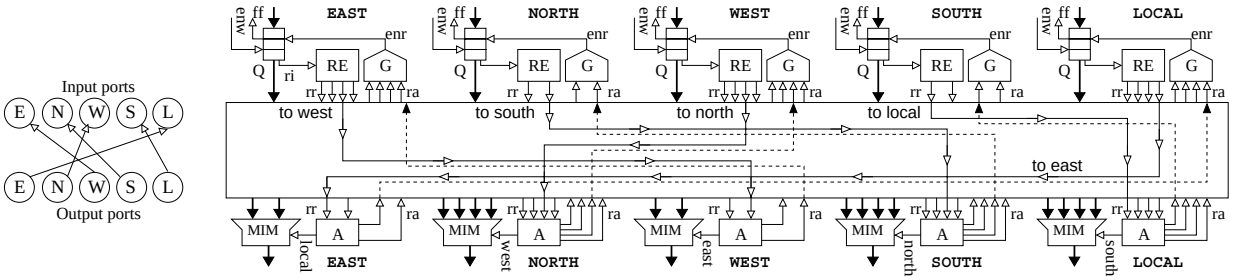
V. Performance Evaluation

A. Traffic Scenario

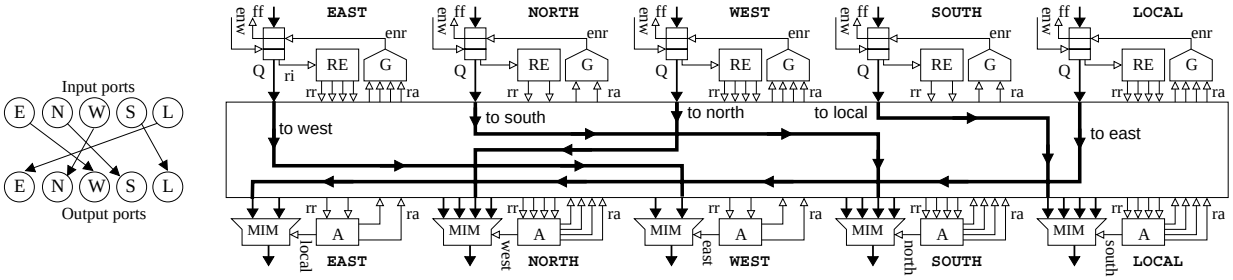
We use a transpose benchmark to evaluate the NoC performance over two alternative prototypes presented in Section In this scenario, a $4x4$ mesh topology is selected. A processing element (PE) at node (i, j) will send N number of flits (a message) to another PE at node (j, i) (like matrix transpose operation). The message is then routed using the static XY routing algorithm presented in Section IV-B.



(a) Routing-Request signals



(b) Request granting or acknowledgement



(c) Data outgoing

Fig. 7. Request-Grant-Accept mechanism to switch data in a router.

Six internode data communication pairs are established in this scenario. An internode data communication is a pair of PE sending a message and a PE receiving the sent message. Communication 1 (*Comm 1*) is a communication pair between node (1,0) as data sender and (0,1) as the data acceptor. The communication pairs are represented as $Comm\ 1|(1,0) \Leftrightarrow (0,1)$. The rest of the communication pairs are represented as $Comm\ 2|(2,0) \Leftrightarrow (0,2)$, $Comm\ 3|(3,0) \Leftrightarrow (0,3)$, $Comm\ 4|(2,1) \Leftrightarrow (1,2)$, $Comm\ 5|(3,1) \Leftrightarrow (1,3)$ and $Comm\ 6|(3,2) \Leftrightarrow (2,3)$. Fig. 1 is helpful to overview the aforementioned scenario showing the (x, y) address of each processing element.

The (Register Transfer Level) RTL simulation is run in which six test pattern generators at the node (i, j) send various N numbers of flits to the destinations at the node (j, i). We use transfer latency measured in clock cycle metric to present the required number of clock cycles to transfer the last flit of each message from the data sender to the data acceptor. Each message injected from the sender nodes is encoded in such a way that it can be correctly identified at acceptor node, and can be differentiated from other message in the NoC. The data flits of each message are numbered in-order, so that it is easy for us to check whether any or some flits loose or are dropped in the NoC. The integrity of the message will be evaluated at the acception node by our test pattern evaluator units.

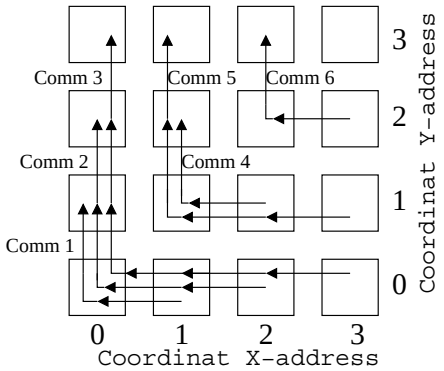


Fig. 8. The data flows under transpose traffic scenario.

The main objective of the simulation is to check the quality of service provided by the router in such a way that all accepted data in the target nodes are equal to all data sent from the data sender node. We select the matrix transpose traffic scenario because the traffic pattern shows a potential data contention between certain considered traffics. The transpose traffics scenario under 2D 4x4 mesh NoC topology with static routing algorithm will present traffic patterns

TABLE I
COMMUNICATION RATES BETWEEN DATA SENDERS AND ACCEPTORS MEASURED IN FLITS PER CYCLE (fpc).

Comm.	2-cyc. ctrl. pipe.	1-cyc. ctrl. pipe.	Outperform.
Comm 1	0.166 fpc	0.259 fpc	1.499 times
Comm 2	0.166 fpc	0.249 fpc	1.497 times
Comm 3	0.111 fpc	0.166 fpc	1.498 times
Comm 4	0.166 fpc	0.247 fpc	1.486 times
Comm 5	0.166 fpc	0.249 fpc	1.498 times
Comm 6	0.332 fpc	0.497 fpc	1.497 times

in such a way that one, two and three messages are contenting to share the same communication links. Therefore, we can analyze the effect of the contentions over the NoC performance. We do not show other traffic scenarios, e.g. bit complement, bit reversal, butterfly, bit rotation or uniform traffic scenario, because they do not perform more traffic contentions than the transpose traffic.

B. Traffic Evaluation

When data flits are injected to the NoC under transpose traffic scenario, then the path of each sender–acceptor nodes is presented in Fig. 8. Data communications of $Comm$ 1, $Comm$ 2 and $Comm$ 3 have to share the communication links of node (1,0) to node (0,0) and node (0,0) to node (0,1). Data communications of $Comm$ 4 and $Comm$ 5 must share the communication links of node (2,1) to node (1,1) and node (1,1) to node (1,2). While the $Comm$ 6 has no contention with the other messages.

The performance evaluations of the two router prototypes are presented in Fig. 9. Variable numbers of flits are injected from the data sender nodes i.e., 250, 500, 1500, 2000, 2500, 3000, 3500 and 4000 flits per data producer node. In general, the figure shows that the transfer latencies of the tail flits of each message are increased linearly with the increased number of the flits from data sender nodes. The transfer latencies of the router prototype with 2-cycle control path pipeline are larger than the other router with 1-cycle control path pipeline.

This situation is actually can be prior estimated qualitatively when analyzing the timing diagram of both router prototypes presented in Section III. However, the impacts of the stage numbers of the pipelined control-path is quantitatively presented in Table I. It looks that the average communication rates can be increased about 1.5 times when the control path pipeline is reduced 1 cycle period by adding data path in parallel with the control path of the routing engine. In other words, the routing control path is made at the same overlapping cycle time with the new data buffer in the REB module (new data path pipeline).

The data communication rate of the $Comm$ 6 presents the best data rate because this sender–acceptor communication pair has no contention with other messages to use the requested communication resources as presented in Fig. 8. Hence it enjoys the maximum bandwidth capacity of the links (about 0.5 flit per cycle). The rest communication pairs share the communication resources in the NoC. Hence, their data rates are lower than the $Comm$ 6. The more messages share the same communication resources with other messages, the lower their data communication rates because the considered messages must share the maximum bandwidth capacity of the communication link. The data rates of $Comm$ 4 for instance use only 50 % (about 0.25 flit per cycle) of the maximum bandwidth capacity of the links because the rest 50 % is shared for $Comm$ 5.

$Comm$ 3 presents the lowest data rates compared with the other data communication pairs. This sender–acceptor communication pair shares the West outgoing link connecting node (2,0) to node (1,0) with $Comm$ 2. The $Comm$ 3 alone with $Comm$ 2 coming from East input port of the router node (1,0) shares the West outgoing link connecting node (1,0) to node(0,0) with $Comm$ 1 injected from Local input port of the router node (1,0). The $Comm$ 3 uses the smaller portion of the maximum bandwidth capacity of the West outgoing link connecting connecting node (2,0) to node(1,0) because of two possible factors. Firstly, because a congestion phenomenon appears on the NoC, and secondly, because the flit-by-flit arbitration mechanism changes the selection instantly in every cycle although the requesting flit has not been selected to acquire the requested output port. Therefore, it has the lowest data communication rate compared with the other communication pairs. The longest path of the $Comm$ 3 from the sender to the acceptor node affects also its communication rate.

TABLE II
TOTAL CELL AREA AND WORKING FREQUENCY FOR BOTH MESH PROTOTYPES USING 130-NM TECHNOLOGY.

	2-cyc. ctrl. pipe.	1-cyc. ctrl. pipe.	Overhead
Total cell area	0.077 mm^2	0.082 mm^2	6.5 %
Working frequency	1 GHz	1 GHz	0 %

TABLE III
COMMUNICATION BANDWIDTH BETWEEN DATA SENDERS AND ACCEPTORS MEASURED IN MEGA-BYTE PER SECOND (MB/s).

Comm.	2-cyc. ctrl. pipe.	1-cyc. ctrl. pipe.	Outperform.
Comm 1	665 MB/s	997 MB/s	1.499 times
Comm 2	664 MB/s	995 MB/s	1.497 times
Comm 3	443 MB/s	664 MB/s	1.498 times
Comm 4	665 MB/s	988 MB/s	1.486 times
Comm 5	664 MB/s	995 MB/s	1.498 times
Comm 6	1327 MB/s	1987 MB/s	1.497 times

VI. LOGIC CELL AREA EVALUATION

In order to analyze the trade-off between the logic area and the performances of our two alternative routers, this section will present the comparison of the total estimated logic cell areas between both routers. Table II exhibits the total cell areas using 130-nm standard-cell technology from *Faraday Technology Corp.* by utilizing *Design Vision* tool from *Synopsys*. The maximum data frequencies of both routers are similar. It seems that the router with 1-cycle control path pipeline has about 6.5 % area overhead over the other router prototype. Both router prototypes can be clocked at cycle-frequency of 1 GHz or in cycle-period of 1 ns ($10^{-9}s$)

If we reformulate the simulation result presented in Table I and the working frequency of the NoCs shown in Table II, then the average communication bandwidth of the sender–acceptor data communication can be calculated. If C_{rate} is the communication rate measured in fpc (flit per cycle), 1 flit consists of 1 Word or 4 Bytes, and F is the working frequency, then we can obtain the communication bandwidth C_{bw} of each sender–acceptor node i.e., $C_{bw} = 4 \times F \times C_{rate}$. Table III shows the calculation results of each sender–acceptor data communication bandwidth measured in Mega-Byte per second.

Table IV shows also the direct comparison of each component in the routers. The MIM modules give the largest area contribution of the total area of the routers. The cell area of the MIM module is large because the *ID-Management Unit* that consists of ID slot tables (registers) is implemented in this module. The implementation

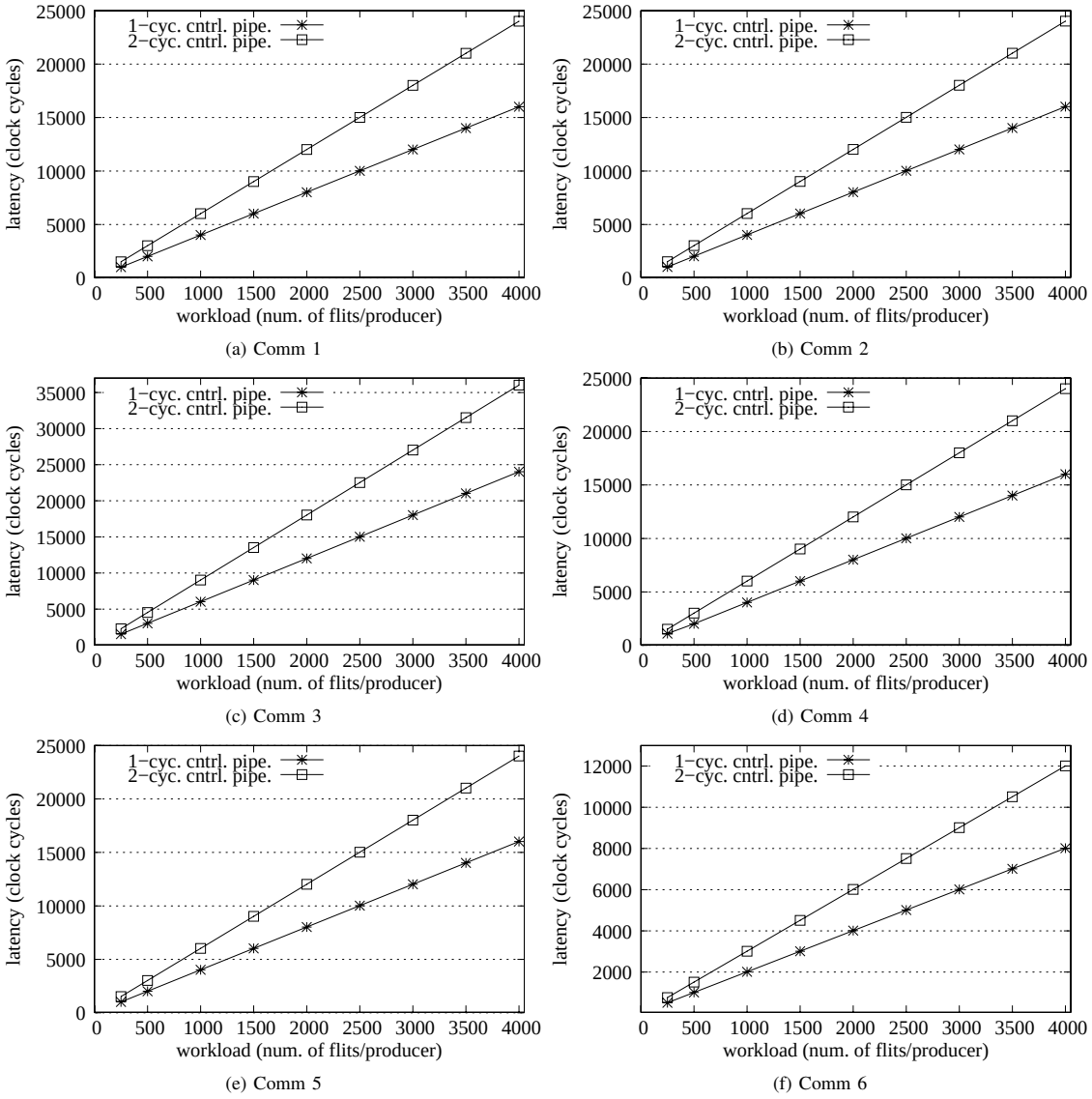


Fig. 9. Comparisons of the transfer latencies of both prototypes.

TABLE IV
AREA COMPARISONS OF EACH COMPONENT FOR BOTH MESH PROTOTYPES USING 130-NM TECHNOLOGY.

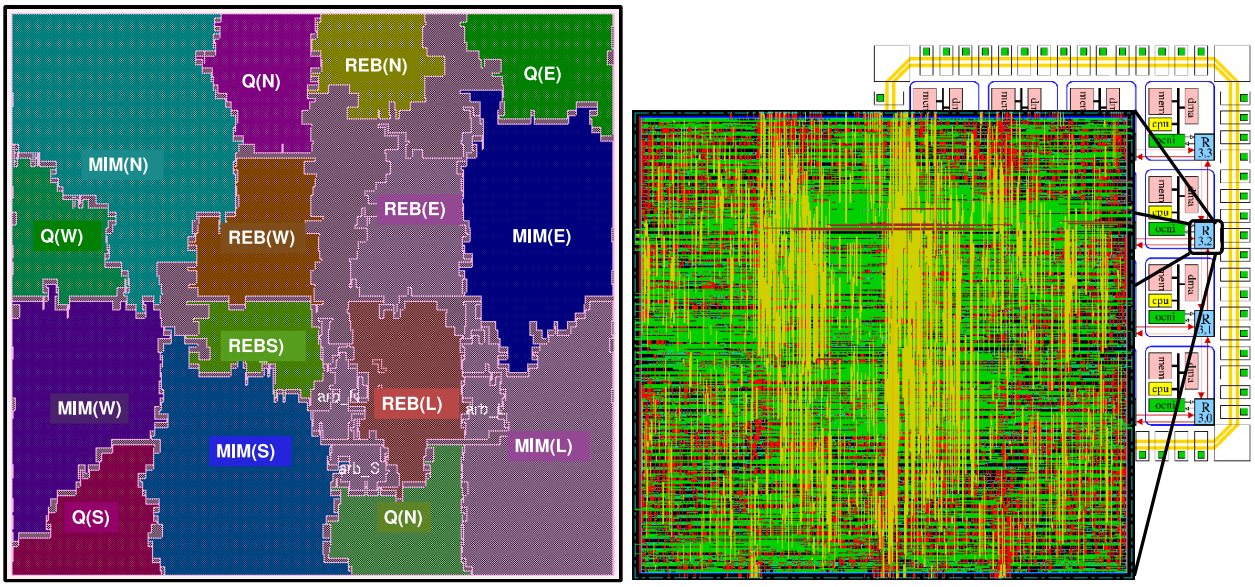
	2-cyc. ctrl. pipe.	1-cyc. ctrl. pipe.
Arbiter (A)	2127 μm^2	2042 μm^2
MIM	45404 μm^2	44987 μm^2
FIFO Queue	17003 μm^2	16608 μm^2
Grant Unit (G)	201 μm^2	195 μm^2
RE/REQ	13094 μm^2	19041 μm^2

of the slot tables consumes also a significant amount of logic cells. As shown in Table IV, the significant difference of the logic cell area is presented by the implementation of the *Routing Engine* modules of the routers. The REB module has larger area than the RE module because an additional register to store data is implemented alone with a routing state machine mechanism in the same module.

The router prototype with 1-cycle control path pipeline has also been implemented on 180-nm CMOS standard-cell technology from UMC (*United Microelectronics Corporation*). The circuit layout of the router is presented in Fig. 10. The left side of the figure presents the floorplanning result. The floorplan view presents the area allocation of each individual component at East (E), North (N), West (W), South (S) and Local (L) incoming and outgoing ports. While the right side figure shows the cell-placement and routing results. The circuit layout is designed using *Silicon Encounter* tool from *Cadence*. Table V exhibits also the cell area of each component type in the router. The working frequency of the NoC prototype using the 180-nm technology is about 505 MHz.

VII. CONCLUDING REMARKS

This paper has presented the area and performance trade-off between two router prototypes having different numbers of pipeline stages in the routers. The first router prototype (2-cycle control path pipeline) has smaller logic cell area compared with the second



(a) Floorplan result

(b) Place and Route Result

Fig. 10. Floorplan, cell-placement and wire routing using 180-nm technology.

TABLE V
CELL AREAS OF THE ROUTER COMPONENTS WITH 1-CYCLE CONTROL PATH PIPELINE USING 180-NM TECHNOLOGY.

Component	Area (μm^2)	Percentage
Arbiter (A)	5397	2.3 %
MIM	130731	58.9 %
FIFO Queue	43057	19.1 %
Grant Unit (G)	493	0.2 %
RE/REM	46113	20.4 %
TOTAL CELL AREA	225791	100.0 %

prototype (one-cycle control path pipeline). The area overhead of the second prototype is about 6.5 % over the first prototype. However, the maximum bandwidth capacity of each link of the second prototype is about 1.5 times larger than the first prototype or about 50 % bandwidth overhead over the first prototype. Therefore, for high performance computing purpose, the second prototype is preferably because the bandwidth overhead (bandwidth gain) is much larger than area overhead (area cost).

The maximum bandwidth capacity of each link of the second prototype is about 2000 MB/s (2 GB/s). Because of the router capability to perform the simultaneous crossbar data interconnect, then for the mesh router with 5 I/O ports, the total maximum bandwidth capacity of the router is about $5 \times 2000\ MB/s = 10\ GB/s$ or about $32 \times 10\ GB/s = 320\ Gbit/s$.

In the future, we will investigate about additional quality of service (QoS) that will be implemented in the network layer of the router. The quality of service can be in term of a guaranteed-bandwidth service that is very important for data streaming-based applications. The design method for automatic generation of NoC RTL-level model is also an interesting topic for a modern CAD-based digital circuit design.

NoC Reconfiguration for embedded system application based on the NoC specification at runtime and/or at design-time is also a challenging topic. The reconfiguration parameters can be a network-site reconfiguration, e.g. the routing algorithm implemented in the NoC router, quality of service reconfiguration provided by the NoC router or a processor-site reconfiguration, e.g. functional reconfiguration of each computing element in the processor nodes.

ACKNOWLEDGEMENT

The authors gratefully acknowledge DAAD (*Deutscher Akademischer Austausch-Dienst,* German Academic Exchange Service) for awarding the first author with DAAD-Scholarship to pursue doctoral-degree at Institute of Microelectronic Systems, Darmstadt University of Technology, in Germany.

REFERENCES

[1] P. Martin, "Design of a Virtual Component Neutral Network-on-Chip Transaction Layer," *Proc. Design, Automation and Test in Europe Conf. and Exhibition (DATE'05),* pp. 336-337, 2005.
[2] D. Wingard, "MicroNetwork-Based Integration for SOCs," *Proc. Design Automation Conf. (DAC'01),* pp. 673-677, 2001.
[3] L. Benini and G. De Micheli, "Networks on Chips: A New SoC Paradigm," *IEEE Computer,* vol. 35, pp. 70-78, Jan. 2002.
[4] A. Jantsch and H. Tenhunen, *Networks on Chip*, Kluwer Academic Publisher, Hingham, MA, USA, 2003.
[5] M. Millberg, E. Nilsson, R. Thid and A. Jantsch, "Guaranteed Bandwidth using Looped Containers in Temporally Disjoint Networks within the Nostrum Network on Chip," *Proc. Design, Automation and Test in Europe Conf. and Exhibition (DATE'04),* pp. 890-895, 2004.
[6] D. Wiklund and D. Liu, "SoCBUS: Switched Network on Chip for Hard Real Time Embedded Systems," *Proc. IEEE Int'l Parallel and Distributed Processing Symposium (IPDPS'03),* 8 pp., 2003.
[7] M. B. Taylor, J. Kim, J. Miller, D. Wentzlaff, F. Ghodrat, B. Greenwald, H. Hoffman, et. al., "The Raw Microprocessor: A Computational Fabric for Software Circuits and General-Purpose Programs," *IEEE Micro,* vol. 22, issue 2, pp. 25-35, Mar-Apr. 2002.
[8] C. Hilton and B. Nelson, "PNOC: a flexible circuit-switched NoC for FPGA-based systems," *IEE Proc. Computers and Digital Techniques,* vol. 153, no.3, pp. 181-188, May 2006.

[9] S. Kumar, A. Jantsch, J. -K. Soininen, M. Forsell, M. Millberg, J. Öberg, K. Tiensyrja and A. Hemani, "A Network on Chip Architecture and Design Methodology," *Proc. IEEE Computer Society Annual Symposium on VLSI,* pp. 105-112, 2002.
[10] M. K. -F. Schäfer, T. Hollstein, H. Zimmer, M. Glesner, "Deadlock-Free Routing and Component Placement for Irregular Mesh-based Network-on-Chip," *IEEE/ACM Int'l Conf. on CAD (ICCAD'05),* pp. 238-245, 2005.
[11] F. Karim, A. Nguyen and S. Dey, "An Interconnect Architecture for Networking Systems on Chips," *IEEE Micro,* vol. 22, issue 5, pp. 36-45, Sept-Oct. 2002.
[12] P. Guerrier and A. Greiner, "A Generic Architecture for On-Chip Packet-Switched Interconnection," *Proc. Design, Automation and Test in Europe Conf. and Exhibition (DATE'00),* pp. 250-256, 2000.
[13] I. M. Panades, A. Greiner and A. Sheibanyrad, "A Low Cost Network-on-Chip with Guaranteed Service Well Suited to the GALS Approach," *Proc. the 1st Int'l Conf. and Workshop on Nano-Networks),* pp. 1-5, 2006.
[14] 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. Computers and Digital Techniques,* vol. 150, no. 5, pp. 294-302, Sep. 2003.
[15] T. A. Bartic, J. -Y. Mignolet, V. Nollet, T. Marescaux, D. Verkest, S. Vernalde and R. Lauwereins, "Topology adaptive network-on-chip design and implementation," *IEE Proc. Computers and Digital Techniques,* vol. 152, no.4, pp. 467-472, July 2005.
[16] L. Benini and D. Bertozzi, "Network-on-chip architectures and design methods," *IEE Proc. Computers and Digital Techniques,* vol. 152, no.2, pp. 261-272, Mar. 2005.
[17] J. Xu, W. Wolf, J. Henkel and S. Chakradhar, "A Design Methodology for application-Specific Networks-on-Chip," *ACM Trans. on Embedded Computing Systems,* vol. 5, no. 2, pp. 263-280, May 2006.
[18] J. Bainbridge and S. Furber, "Chain: A Delay-Insensitive Chip Area Interconnect," *IEEE Micro,* vol. 22, issue 5, pp. 16-23, Sept-Oct. 2002.
[19] M. Amde, T. Felicijan, A. Efthymiou, D. Edwards and L. Lavagno, "Asynchronous on-chip networks," *IEE Proc. Computers and Digital Techniques,* vol. 152, no. 2, pp. 273-283, Mar. 2005.
[20] M. Sgroi, M. Sheets, K. Keutzer, S. Malik, J. Rabaey and A. S. Vincentelli, "Addressing the System-on-a-Chip Interconnect Woes Through Communication-Based Design," *The 38th ACM Design Automation Conf. (DAC'01),* pp. 667-672, 2001.
[21] I. Saastamoinen, D. S. -Tortosa and J. Nurmi, "Interconnect IP Node for Future System-on-Chip Designs," *The 1st IEEE Int'l Workshop on Electronic Design, Test and Applications (DELTA'02),* pp. 116-120, 2002.
[22] E. Beigné, F. Clermidy, P. Vivet, A. Clouard and M. Renaudin, "An Asynchronous NOC Architecture Providing Low Latency Service and its Muti-level Design Framework," *Proc. the 11th IEEE Int'l Symp. on Asynchronous Circuits and Systems,* pp. 54-63, 2005.
[23] T. Bjerregaard and J. Sparsø, "Implementation of guaranteed services in the MANGO clockless network-on-chip," *IEE Proc. Computers and Digital Techniques,* vol. 153, no.4, pp. 217-229, July 2006.
[24] J. Kim, D. Park, N. Vijaykrishnan and C. R. Das, "A Low Latency Router Supporting Adaptivity for On-Chip Interconnects, " *ACM Design Automation Conf. (DAC'05),* pp. 559-564, 2005.