
Performance Evaluation of MIC@R NoC for Real-Time Applications

Rafik Ben-Tekaya^{* a, b}, Adel Baganne^b, Kholdoun Torki^c and Rached Tourki^a

^a EμE Lab, Faculty of Sciences Monastir, 5000, Tunisia.

^b LESTER Lab -CNRS, UBS University, BP 92116, 56321 Lorient, France.

^c CMP, INPG, 46 Avenue Filex Viallet 38031, Grenoble Cedex, France.

*E-mail: rafik.tekaya@gmail.com

E-mail: Adel.baganne@gmail.com

E-mail: kholdoun.torki@imag.fr

E-mail: rached.tourki@fsm.rnu.tn

*Corresponding author.

Abstract: The smart electronic homes evolution is strongly related to the System-on-Chip (SoC) development. This, in turn, requires an efficient intercommunication between its Intellectuals Proprieties (IPs). Network-on-Chip (NoC) represents the suitable solution.

This paper presents a design and implementation of MIC@R NoC architecture performed with non-uniform traffics. This architecture offers lowest routing latency (1 cycle) and allows supporting several adaptive routing algorithms. The proposed NoC architecture is implemented in ASIC technology and performed in 2D mesh networks. In this paper we present a study of NoC evaluation. This NoC uses the four routing algorithms: Deterministic X-Y, Fully Adaptive (FA), Proximity Congestion Awareness (PCA) and Proximity Hot-Spot Awareness (PHSA). The PHSA scheme is a novel routing technique proposition that is more efficient than the other ones. NoC performance evolution is measured with non-uniform traffics that are *hot-spot* and *transpose* patterns. Obtained results show that MIC@R router combined with proposed routing techniques is efficient in terms of low latency and generic aspect.

Keywords: System-on-Chip (SoC), Network-on-Chip (NoC), Real time application, routing algorithms.

Reference to this paper should be made as follows: R. Ben-Tekaya, A. Baganne, K. Torki and R. Tourki, "Performance Evaluation of MIC@R NoC for Real-Time Application". *Int. J. Computer Aided Engineering and Technology*, Vol. x, No. x, pp. xxx-xxx.

Biographical notes: **Rafik Ben Tekaya** received his M.S. degree in Electronics from the Science Faculty of Tunis, Tunisia, in 2005. Currently, he is a PhD student. His research interests include Network-on-Chip concept and design methodology, multimedia application. He is working in collaboration with LESTER Laboratory, Lorient, France and CMP, INPG, France.

Adel Baganne received his M.S.degree and the Ph.D degree from Rennes University (France) in 1994 and 1997, respectively. He is presently an Associate Professor at the UBS University since 1998 and a member of the LESTER-CNRS Lab. He has published a number of research papers in the area of computer architecture and SoC design. His research interests include NOC design, communication synthesis, High level synthesis and CAD tools.

Kholdoun Torki received his Ph.D. degree in Microelectronics from the Institut National Polytechnique de Grenoble, France, in 1990. During 1990-1994 he was engineer at CMP, and from 1994 he joined CNRS (Centre National de la recherche scientifique). He is currently Technical Director of CMP since 2002. His research interest includes Deep Submicron design-flows and methodologies, Thermal simulation and measurements at IC level, and Magnetic memories (MRAM) modeling and simulation.

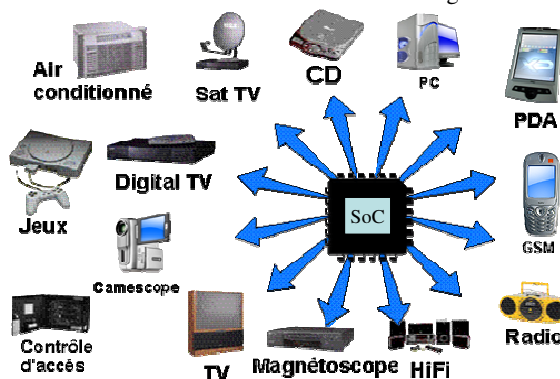
Rached Tourki received the B.S. degree in Physics (Electronics option) from Tunis University, in 1970; the M.S. and the Doctorat de 3eme cycle in Electronics from Institut d'Electronique d'Orsay, Paris south University in 1971 and 1973 respectively. From 1973 to 1974 he served as microelectronics engineer in Thomson CSF. He received the Doctorat d'etat in Physics from Nice University in 1979. Since this date he has been professor in Microelectronics and Microprocessors with the physics department, Faculte des Sciences de Monastir. His current research interests include: Digital signal processing and hardware software codesign for rapid prototyping in telecommunications

1 Introduction

Semiconductor development is a fundamental parameter to smart homes industry. Homes devices performance is strongly related to the integrated VLSI (Very Large Scale Integrated) technology. This, in turn requires a high performance digital and analogue circuits process.

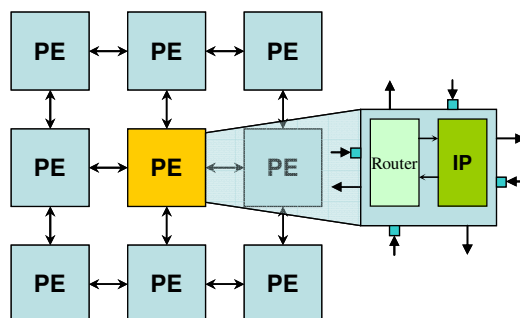
Today System-on-Chip (SoC) represents the highest technology of VLSI. A simple silicon chip can include in same time multiprocessors, RAMs, DSPs... with more than a billion transistors. These components are organised as Processing Elements (PE) integrated in the same SoC. They are called Intellectual Propriety (IPs) blocks taking hardware or software forms. Real time applications need an efficient communication support in the SoC. Hence, the interconnection between IPs plays a major role in SoC performance.

Figure 1 high performance homes electronic devices are based on the integrated SoC



Network-on-Chip (NoC) approach represents a promising solution to SoC intercommunication [14]. It aims to replace traditional inter-chip connections such as shared buses and it is able to address the increasing interconnect complexity challenges. Such architecture is composed of arrays of IP blocks, such as embedded processors that communicate with each other through intelligent routers (Figure 2). These routers should be designed to be reusable and their primary role should be to ease the integration process of the functional IPs. Also, to have a reliable and efficient communication in NoC traffic, it must improve throughput and reduce pipeline latency. So, this takes place by designing an effective router with smart routing algorithms.

Figure2 Smart SoC integrated in the network form, in each node there is a Processing Element (PE) formed by Router and IP.



The router represents the major element of NoC design. However basic router architecture can be partitioned into two main groups based on functionality: the *datapath* and *control plane* [1]. The router datapath handles the storage and movement of packet's payload and consist of a set of input/output buffers and a switch (or crossbar). The remaining blocks implement the control plane of the router and are responsible for coordinating the packets movement through the datapath.

The *latency reduction* is an important factor to improve the reliability and the communication efficiency in the NoC. It allows the buffering resources reduction and it helps to resolve many constraints imposed by real-time applications. Latency reduction can be obtained by reducing routing and arbitration delays in the router. It can be obtained too by the implementation of smart adaptive routing algorithms that reduce the chance for packets to enter hot-spots or faulty components, and hence reduce the blocking probability of packets. Mullins et al [6], [7] proposed a single stage router with Virtual Channels and with a doubly speculative pipeline to minimize deterministic routing latency. However among the disadvantages of such approach, the intra-router delay does not imply adaptability to the network congestion. The results given in [6] do not seem to take contention aspect into consideration.

Packet-switch networks employing Virtual Channel (VC) flow control have been proposed as one approach to implement a chip-wide interconnection network, because that helps to avoid the deadlock states. However VC

implementation in each input port requires complex control logic and extra buffering spaces FIFO, which causes more silicon overhead and especially more power consumption. Starting from these observations many designers such as in [4-8-9-10-12] addressed the design of NoCs without VC.

In this work, we present the implementation and performance evaluation of MIC@R router [2-3]. The router architecture is evaluated with several routing schemes without VC, which offers lowest latency that equals to 1 clock-cycle. Compared to previous work [2-3] some improvements are proposed in this paper, in particular the ASIC implementation and the novel routing algorithm proposition called *Proximity Hot-Spot Awareness (PHSA)*. This novel routing scheme presents more performance in term of latency and throughput compared to others techniques.

The NoC is evaluated in a 2D Mesh topology and functioned with *deterministic X-Y* [12], *fully adaptive (FA)*, *Proximity Congestion Awareness (PCA)* [9] and *Proximity Hot-Spot Awareness (PHSA)* routing schemes. In case of PCA and PHSA, routing decision is helped by monitoring the congestion status in the proximity in order to limit *deadlock* and *livelock*. However, PHSA is characterized by hot-spot states detection from proximity routers. To evaluate NoC latency performance, we chose the *transpose* and *Hot-Spot* patterns traffics such as the work presented in [4]. These traffic modes represent the worst case of NoC communication conflicts [13].

The paper is organized as follows. In section 2 we present our MIC@R router. Then, section 3 details the using adaptive routing algorithms. We evaluate both our architecture and routing algorithms by simulation in section 4. FPGA implementation and ASIC synthesis described in section 5. Finally we conclude our paper in section 6.

2 MICA@R Router Model

In this section we recall the MIC@R router architecture [2-3]. It includes n input/output ports, so in the 2D mesh or torus topologies the number n is five: North, East, Local, South and West. Each port has an input buffer for temporary storage of information. The local port establishes a communication between the router and its local core. The others ports of the router are connected to the neighbouring routers. The output port is composed by the following signals (see figure 3): (1) *req_out*: control signal coded in single bit which indicates the data availability; (2) *Data_out*: data to be sent coded in the 32 bits; (3) *Credit_In*: control signal indicating successful data reception (2 bits coding). In case of using PCA and PHSA routing, the credit signal is coded with 5 bits (3 supplementary bits for stress values coding). Table 1 depicts the credit signal states. Each input port includes the following signals: (1) *req_in*, (2) *Credit_out* and (3) *Data_in*, which proceed in the same way as the output port signals.

Figure 3 MIC@R Router Architecture.

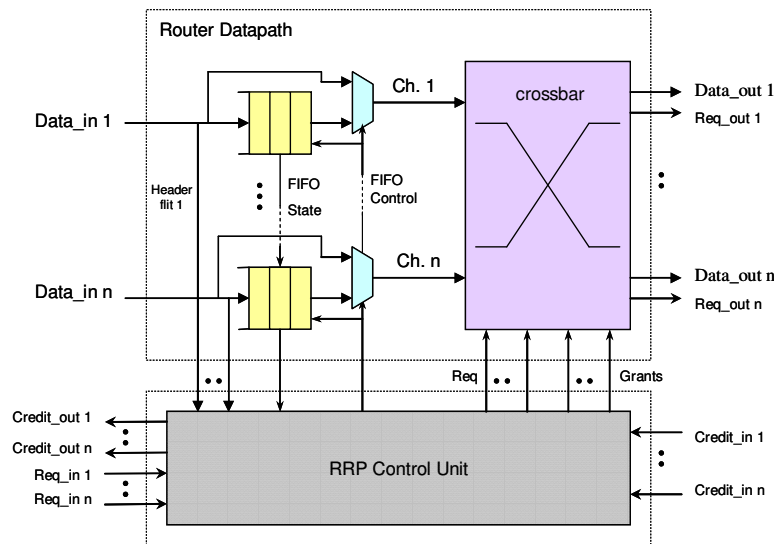


Table 1 Credit signal State

b1	b0	Credit responses
0	0	Ready to Receive
0	1	Okay (data received)
1	0	Congested router
1	1	ERROR

Notice that “congested router” status corresponds to the contention state caused by the neighbouring router outputs conflicts and “congested buffers” status corresponds to cases where the neighbouring FIFO is full. We used a simple handshake protocol to deal with flow control and correctly sending and receiving data.

One point favouring the use of explicit handshake protocols is the possibility to implement asynchronous interconnection between synchronous modules, enabling a *Globally Asynchronous Locally Synchronous (GALS)*. Thus, when the router needs to send data to a neighbour router, it put the *data_out* signal and activates the *req_out* signals. Once the neighbour router stores the data from its *data_in* signal, it asserts its *credit_out* signal, until the transmission is complete. The *header flit* contains the address of target router coded with 8 bits, and the number of flits in the packet payload coded also with 8 bits. The address information is expressed in XY coordinates, where X represents the horizontal position and Y the vertical one. Once the header flit is routed, the remaining packet flits complete the transfer by the same output port.

The internal router architecture is shown in figure 3. Its datapath bloc contains a simple FIFO in each input port and a crossbar. The routing computation is carried out by a Fast Parallel Routing (*FPR*) control unit. It consists of parallel processing executing a routing arbitration in a one step.

This router can be used for 2D mesh networks or for other topologies, such as Torus, Ring, Fat Tree etc... In this work, all results are obtained with a 2D mesh topology and in wormhole routing context. With wormhole switching technique, a packet is divided into *flits*. The first flit is designated as the *header flit*, which contains routing information and leads the packet through the network. When the header flit is blocked from advancing due to lack of output channels, all of the flits wait at their current nodes for available channels.

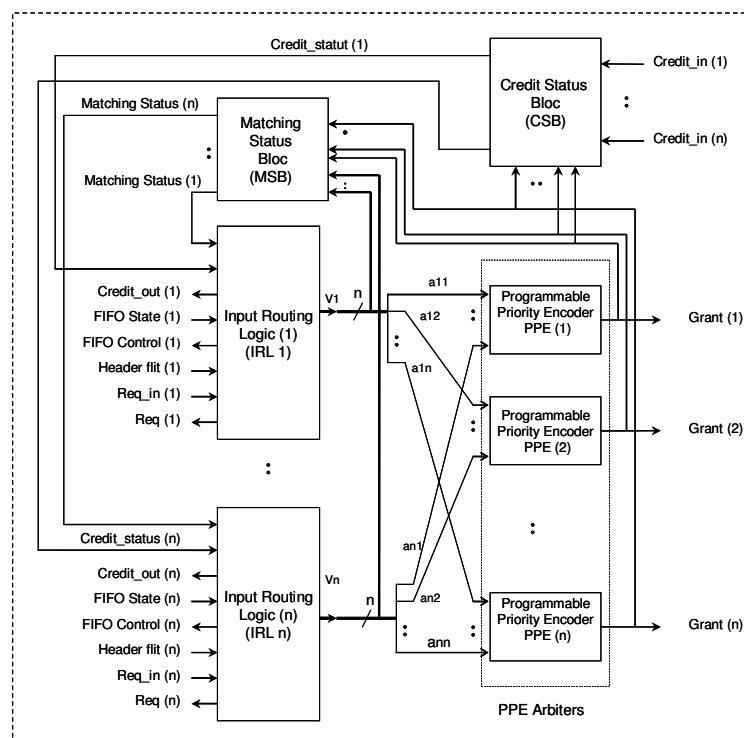
The detailed architecture of the *Fast Parallel Routing (FPR)* control unit is depicted in figure 4. All control signals are integrated in the *FPR* unit. This block is characterized by a fast and parallel execution of routing and arbitration such that it can route in parallel all input packets in only one clock-cycle. In this section we depict this lowest latency and the generic aspect of the architecture.

The *FPR* unit is composed of the following functional blocks:

1. N Input Routing control Logic (*IRL*) blocs (1 *IRL*/input port) that implement the used routing scheme.
2. *PPE* arbiters' bloc that provides the input/output matching decisions.
3. Matching Status Bloc (*MSB*) that compares the *PPE* arbiters with *IRL* outputs.
4. Credits Status Bloc (*CSB*) that offers the credits responses from the neighboring routers.

The *IRL* module controls in parallel, the input FIFO and the routing process. This module is composed of two main blocks: *FIFO Control* and *Routing Computation*. The first block handles the FIFO writing/reading control. It stores the input packet header flit and forwards it directly to the crossbar in order to be sent.

Figure 4 FPR Control Unit Architecture



The *PPE* arbiters block is composed of n Programmable Priority Encoder (*PPE*) sub-blocks. Each *PPE* module controls directly the router switch module, that matches the input and output port. It is composed mainly by a priority encoder module, in which the higher and lower input priority can be pointed. The credits signals obtained from the neighbouring routers are stored in the *MSB* block. They are forwarded to the corresponding *IRL* according to the Grant's signals values.

In the routing scheme implementation only the *IRL* has to be configured and customized while the other blocks are kept unchanged. Otherwise this router can support different routing schemes with deterministic or

adaptive protocols without changing the body architecture. This generic aspect constitutes a key point in NoCs design with variant routing schemes. In particular, this can be efficiently exploited in reconfigurable platforms where routing strategies are modified and updated dynamically.

3 NOC Design with Proposed Routing Schemes

In this section we present the different routing schemes that are used to evaluate our NoC performances. In fact, the routing mode can be classified into two main types [12]: Deterministic and adaptive routing.

When deterministic, the path is completely determined by the source and the destination address. Among the advantages, it presents a simple architecture that allows to be easy implemented. However, as the packet injection rate increases, the deterministic routers are likely to suffer from throughput degradation as they can not dynamically respond to network congestions.

Compared to the deterministic routing, the adaptive routing presents a more complex logic structure, but it helps in achieving better performance specifically under non-uniform traffic patterns. However, the use of adaptive routing requests large time of processing congestion. In our case the proposed router provides lowest latency (1 cycle) in the absence of congestion. This constitutes a major advantage when using adaptive algorithms. Moreover, MIC@R architecture includes the advantageous of both deterministic and adaptive routing algorithms such as DyAD scheme [10]. In this work, we implemented a *deterministic X-Y* routing and 3 adaptive routing schemes: *FA*, *PCA* and *PHSA*.

3.1. Fully Adaptive

A typical *Fully Adaptive* (FA) routing algorithm gives priorities to the desired outputs, so packets are routed toward the destination in the absence of congestion, but allows routing on unproductive outputs to increase path diversity [1]. Thus, packets may be directed over channels that increase the distance from the destination to avoid a congested or failed channel. We used the XY routing that first routes packets along the X-axis. Once the packets reach the column wherein lies the destination router, they are then routed along the Y-axis.

Figure 5 FA Routing

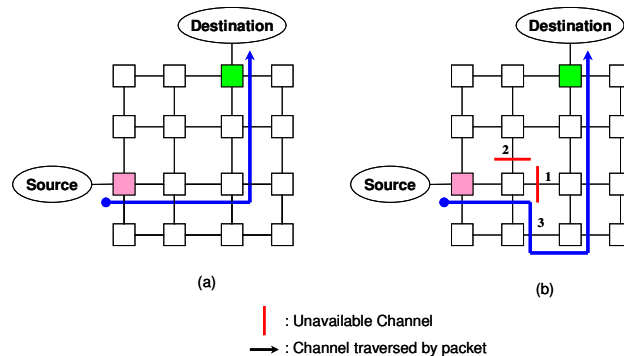


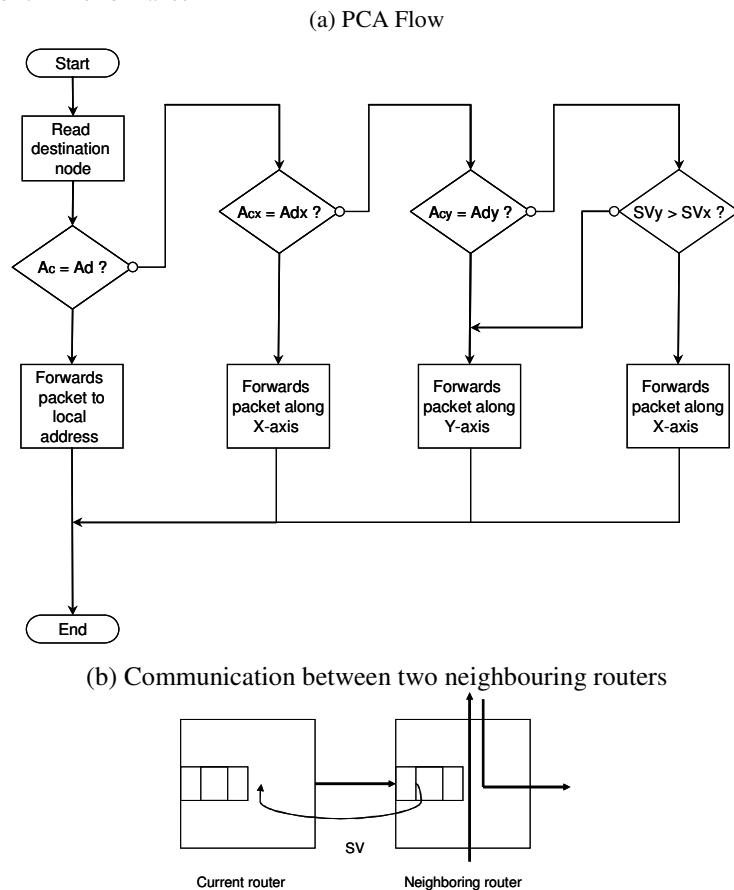
Figure 5 (a) gives an example of a transfer between a source and a destination without congestion in the entire path. It covers five hops, thus the End-to-End latency is equal to 5 clock-cycles. In the case of contention state inside one router (see figure 5 (b)), each routing process of the optimal free output port consumes only one clock-cycle. Thus, in the worst-case there are only 3 clock-cycles consumed by the router in order to find a free output channel. However, this *FA* algorithm suffers from *livelock* problem.

3.2. Proximity Congestion Awareness

In Proximity Congestion Awareness (*PCA*) routing scheme, the current routing decision is helped by monitoring the congestion status in the proximity in order to reduce the congestion states and increase throughput. The present work implements the *PCA* approach such as proposed in [9]. Each packet only travels along the shortest path between the source and the destination. If there are multiple shortest paths available, the routers will help the packet to choose one of them based on the congestion condition of the network. The used *Stress Value* (*SV*) is the number of occupied cells in all the input buffers. Thus data packet is sent to the neighbour with the smallest *SV*. When the *SV* from X and Y channels are equal, the current router forwards the packet through X-axis. At each cycle the *SV* is updated and transferred through credit signals. Hence each router stores the instant *SV* for all neighbours to be used for appropriate path selection.

Lets Ac current node address, Ad destination node address, Acx neighbouring node address according X-axis, Acy neighbouring one according to the Y-axis and respectively Adx , Ady destination address according to the X-axis and Y-axis. Lets also SVx and SVy be the stress value respectively for X and Y-axis, the *PCA* performance is showed as follows (Figure 6).

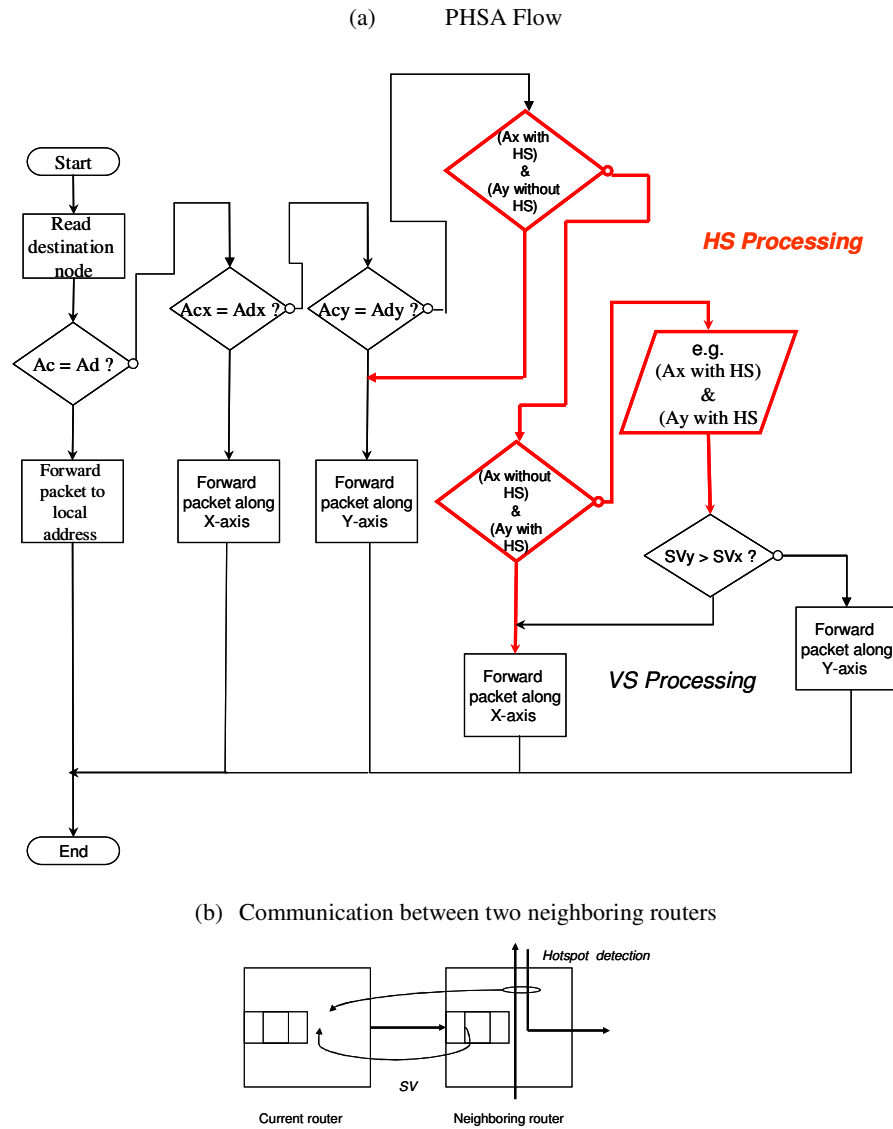
Figure 6 PCA Algorithm Performance



3.2. Proximity Hot-Spot Awareness

The Proximity Hot-Spot Awareness (*PHSA*) technique includes the same *PCA* techniques. However it also includes the *hot-spot* detection property from proximity routers. So the current router receives from neighbouring routers not only the *SV* information but the *hot-spot* information too. This information prevents the packet to enter in contention, thus it gives more chance to send the data on to free paths. The traffic signal indicates whether the corresponding channel is busy or free. Based on this information, the packet can choose the path to the next available router. In our case we have implemented a proximity awareness scheme performed with *minimal-path* adaptive routing. The use of minimal routing helps, not only to reduce the energy consumption of communication, but it limits the *deadlock* and avoids *livelock* problems. *PHSA* is performed as the following (Figure 7).

Figure 7 PHSA scheme



In next section we show the importance of hot-spot detection property addition.

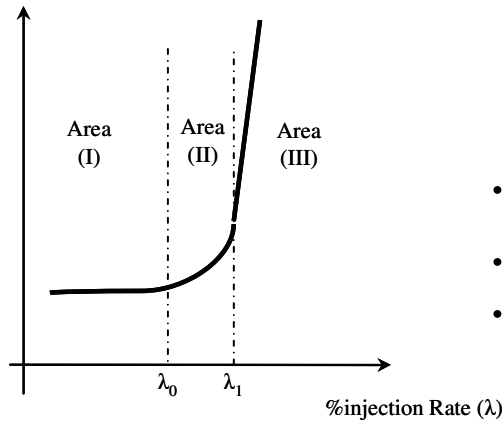
4. Simulation Evaluation

To evaluate latency behaviour of our architecture we have developed traffic pattern generators which send the packets successively to targets, under two traffic patterns: *transpose* and *Hot-Spot*, which are similar to the work presented in [4]. Latency evolution is measured according to packets injection rate (λ). Application-specific (i.e. real time multimedia) traffic performed with non-uniform traffic, so transpose and mostly hot-spot traffic represent the worst case pattern [13]. For this reason we chose these traffic modes. In fact the latency evolution curve according to the packet injection rate passes through three principal areas: *Linear area*, *saturation angle area* and *saturation area*. The following figure shows the general curve form of latency evolution.

Figure 8

FPR Control Unit Architecture

Latency (cycles)



- (I): Linear Area
- (II): Saturation Angle Area
- (III): Saturation Area

To have efficient and reliable communication, the designer must reduce saturation area. So in this work we proposed lowest routing latency architecture that includes some adaptive routing schemes to increase area (I) and (II). Hence we evaluated our NoC with different routing algorithms and with four routing latency delays: one cycle, 2 cycles, 3 cycles and 4 cycles, in order to reduce routing latency reduction, smart routing schemes must be implemented in the router for non-uniform traffics.

4.1. Transpose traffic

For transpose traffic we used mesh 8x8 and 12x12 Network. Moreover, in these simulations we have used various packets' lengths: under 32, 24, 16 and 8 flits. Figures 9, 10, 11, 12, and 13 show how the performance of the *deterministic*, *FA*, *PCA* and *PHSA* routing changes with respect to the network load for an 8x8 and 12x12 mesh and mostly with routing latency delay. In total we injected about 350,000 packets.

Figure 9(a) shows the average latency evaluation with 8x8 NoC and figure 9(b) with 12x12 one. These figures show that *FA* and *Deterministic* schemes perform with low latency close to *PCA* and *PHSA* in case of low network workload. However, when the packet injection rate increases, the *FA* routers are likely to suffer from latency increasing and it can not respond to *livelock* problem. In contrast, *PCA* and mostly *PHSA* present an increase of the packets' chances to avoid congested links better than others schemes in the high rate injection. It also avoids the *livelock*. Compared to other reported literature works, we think that our results are promising. For example latency results presented in [8] (in which packet routing is carried out in 2 stages) and obtained under transpose traffic starts from 24 cycles (low injection rate) and reach the 100 cycles at about 28 % injection rate. In our case and in the same traffic conditions, our latency evolution curve (figure 9-a) begins from 8 cycles and reaches the 100 cycles when the injection rate is beyond 41%.

Figure 9 different routing schemes performance in one clock-cycle latency and with NoC 8x8 and 12x12 sizes.

Fig. 9 (a) NoC 8x8 size

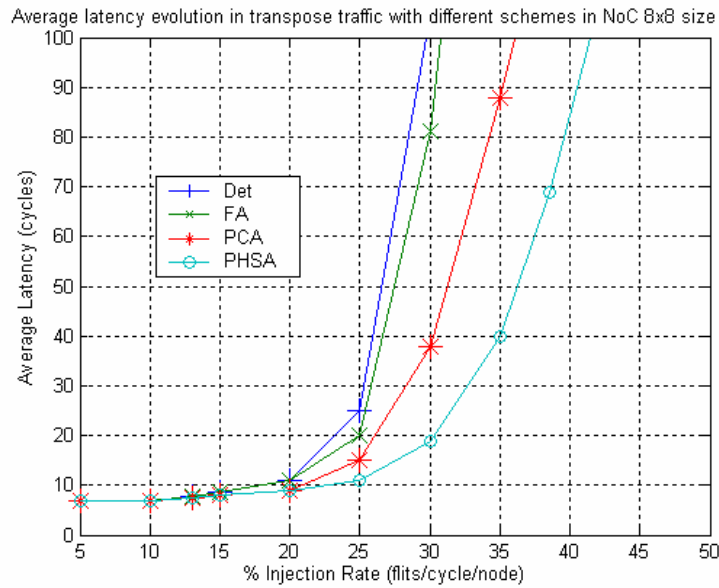
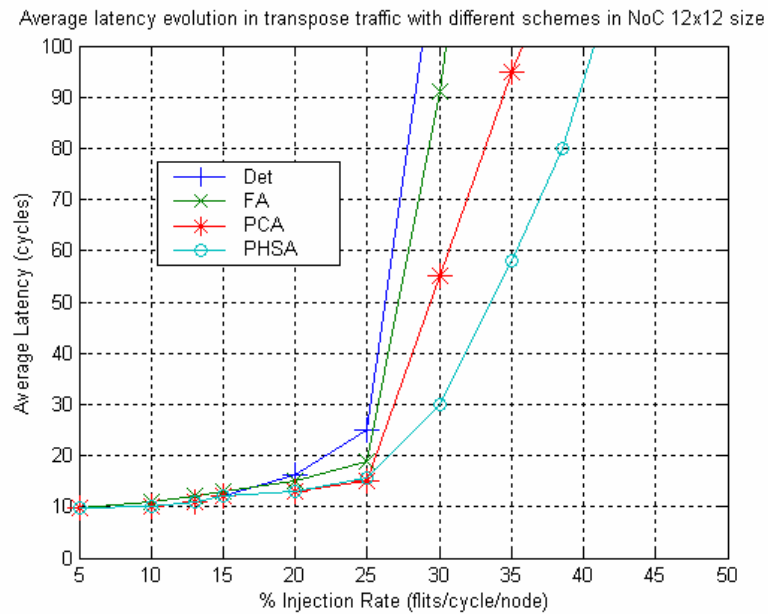


Fig. 9 (b) NoC 12x12 size



Figures (10, 11, 12, 13) show, in the two cases (NoC 8x8 and 12x12 sizes), that a routing latency limitation delay is a major factor to reduce saturation area. This limits the packets congestion and improves communication reliability in the NoC traffic and then makes easier the implementation of the Quality of Service (QoS). However, the one clock-cycle routing latency architecture gives the best performance for all used routing schemes. Moreover, these curves show clearly that smart routing scheme plays an important role to increase area (I) and (II). However simulation results show that PHSA is the best routing algorithm. In fact the hot-spot awareness information from neighbouring routers minimises the probability of packets to enter in congestion.

Figure 10 Deterministic scheme performance on different routing latency with NoC 8x8 and 12x12 sizes.

Fig. 10 (a) NoC 8x8 size

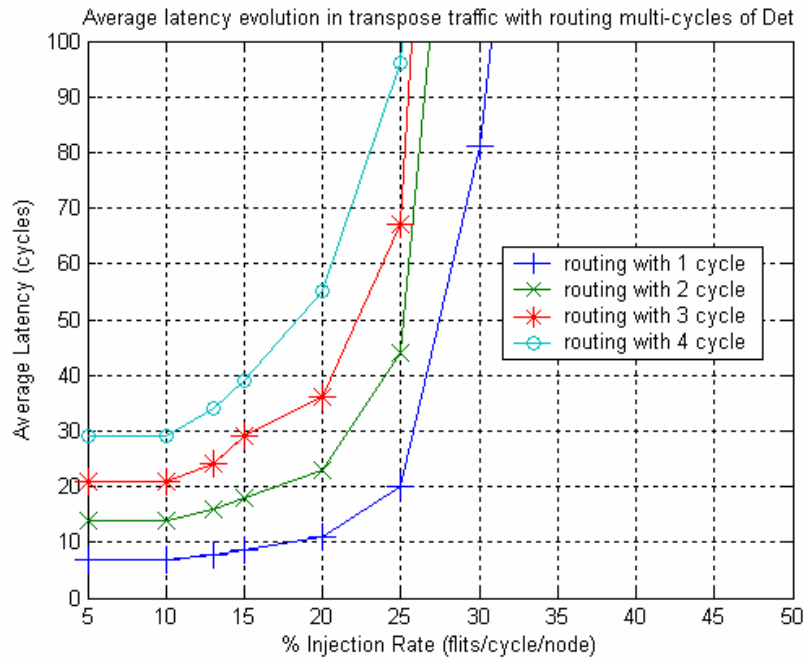


Fig. 10 (b) NoC 12x12 size

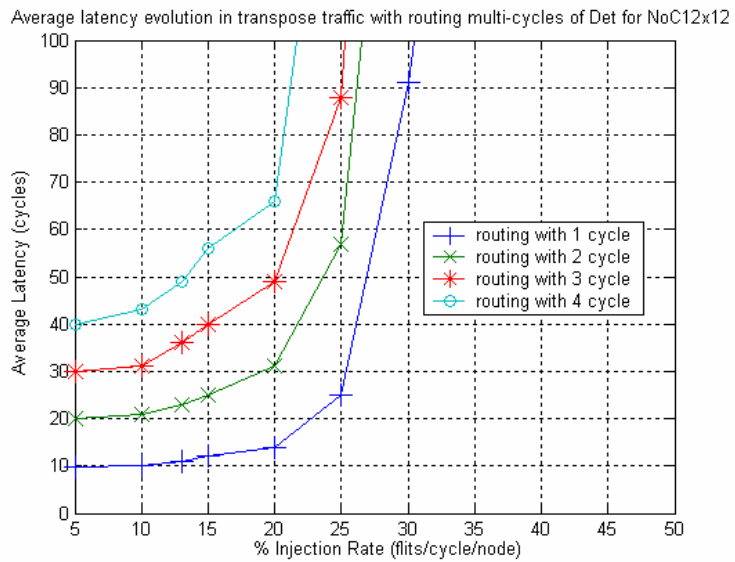


Figure 11 FA scheme performance on different routing latency with NoC 8x8 and 12x12 sizes.

Fig.11 (a) NoC 8x8 size

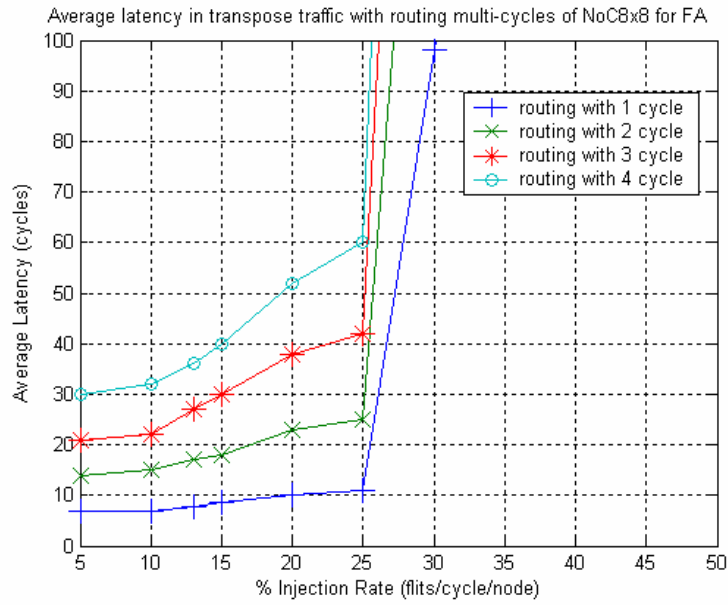


Fig. 11 (b) NoC 12x12 size

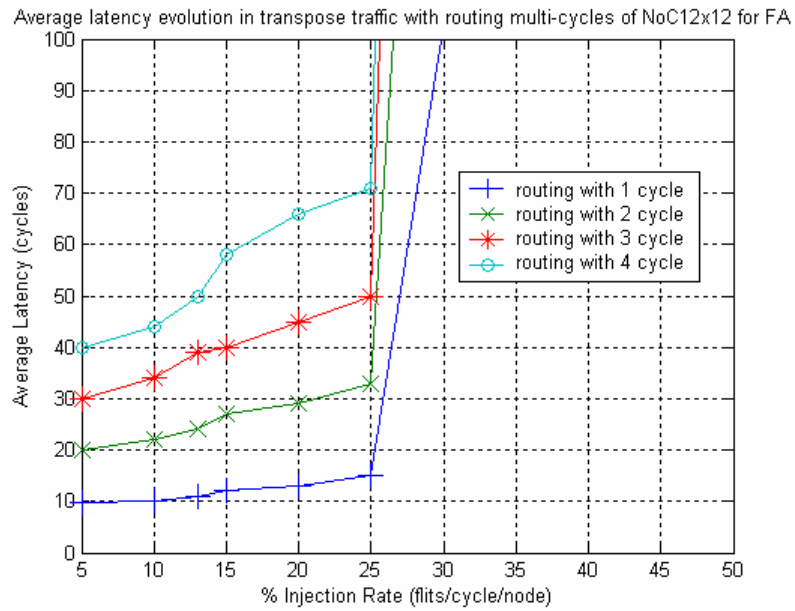


Figure 12 PCA scheme performance on different routing latency and with NoC 8x8 and 12x12 sizes.

Fig. 12 (a) NoC 8x8 size

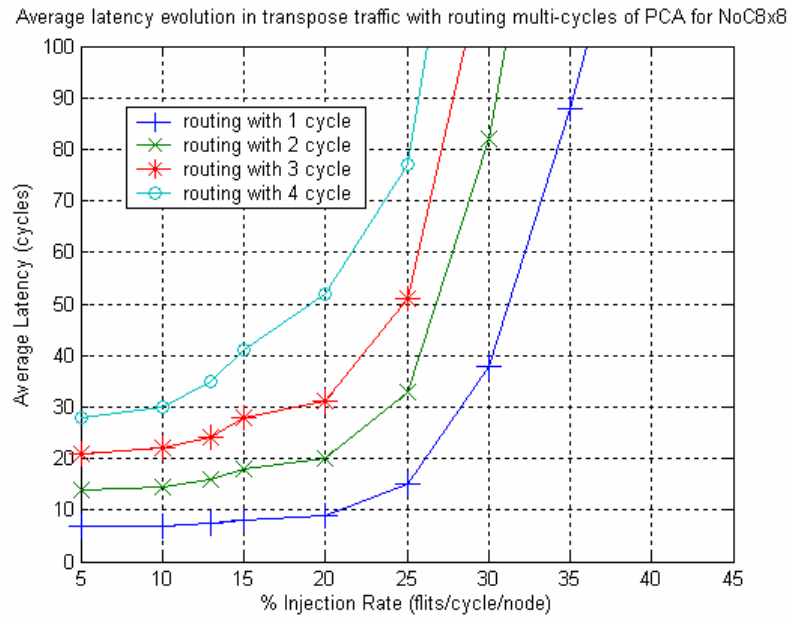


Fig. 12 (b) NoC 12x12 size

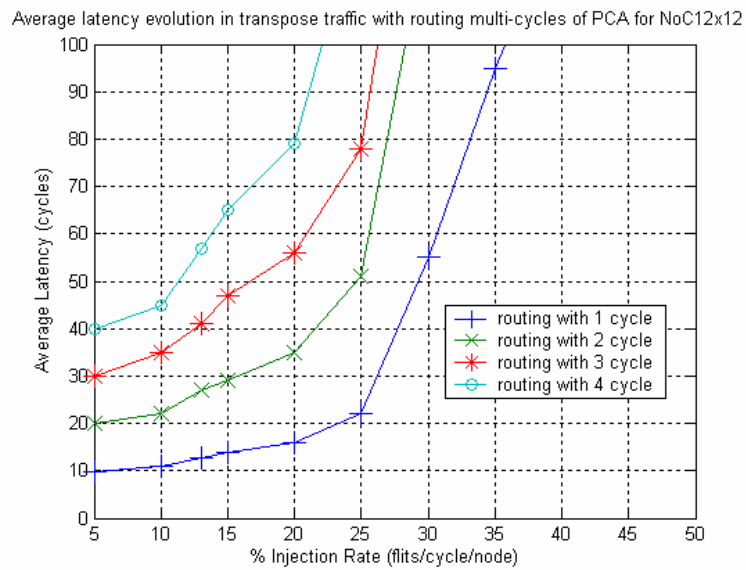


Figure 13 PHSA scheme performance on different routing latency and with NoC 8x8 and 12x12 sizes.

Fig. 13 (a) NoC 8x8 size

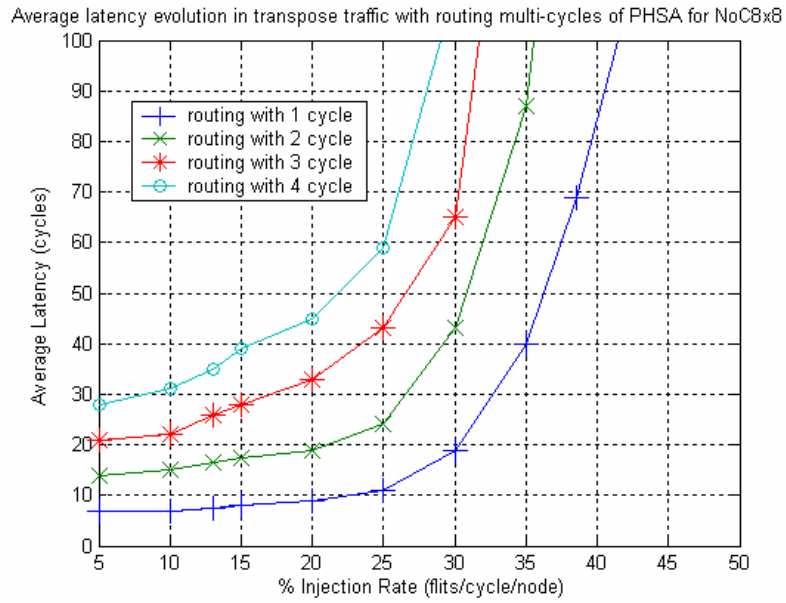
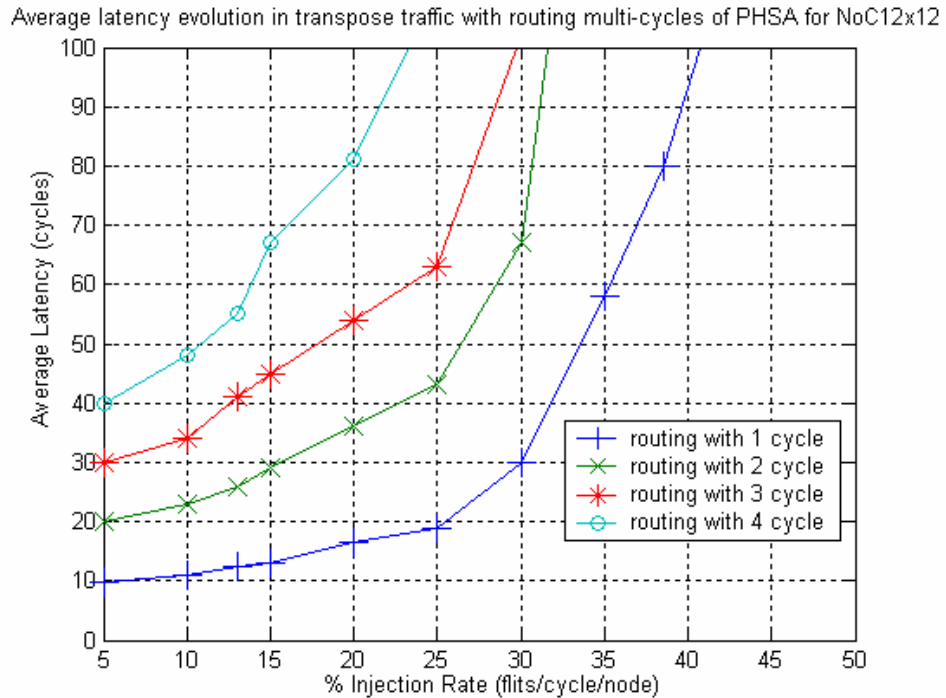


Fig. 13 (b) NoC 12x12 size



4.2. Hot-Spot traffic

In hot-spot traffic latency evaluation we have used 4x4 NoC. We chose a 40% as *hot-spot* percentage. Otherwise, we assume one node to be destination for six sources. The remaining nine nodes perform under a random uniform traffic mode. Then we measure the average latency of total NoC traffic. In total, we injected about 30,000 packets.

In all figures we notice that PHSA routing scheme is the best approach. However, these figures clearly show that under hot spot-traffic (which represents the worst case of NoC traffic), the one clock routing latency delay is significant factor to reduce saturation area. The reliability of the NoC communication is then improved.

Figure 14 Latency evaluation under Hot-Spot traffic with 40% under one clock cycle routing latency

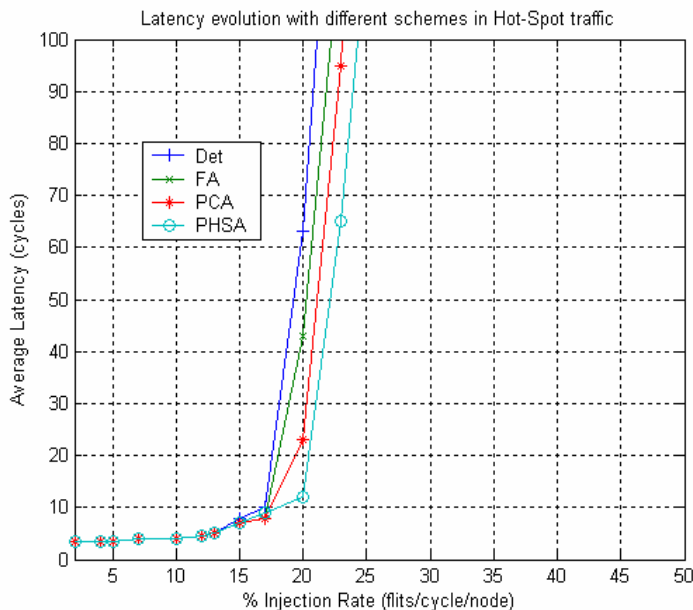


Figure 15 deterministic scheme evaluation

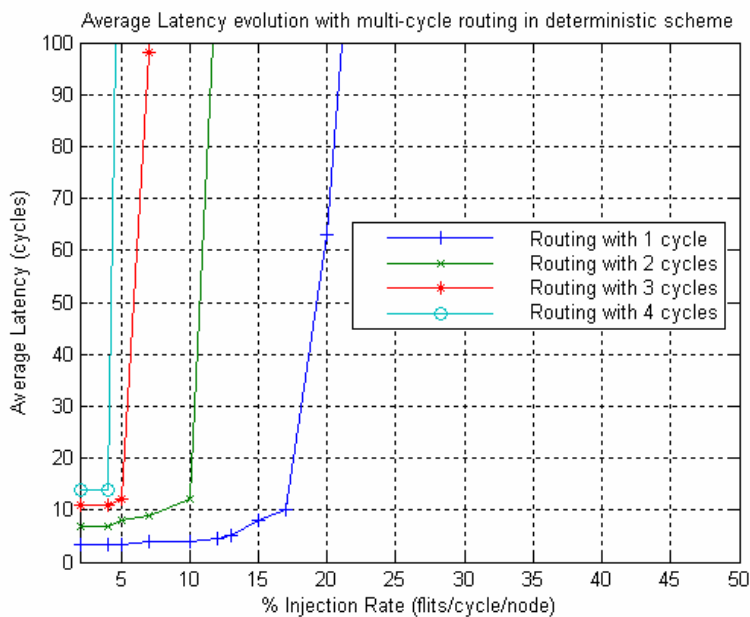


Figure 16 FA scheme evaluation

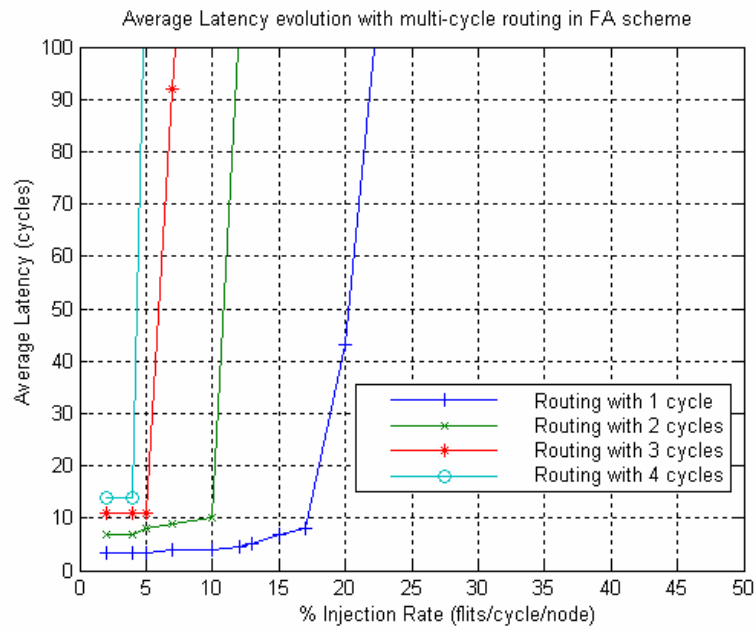


Figure 17 PCA scheme evaluation

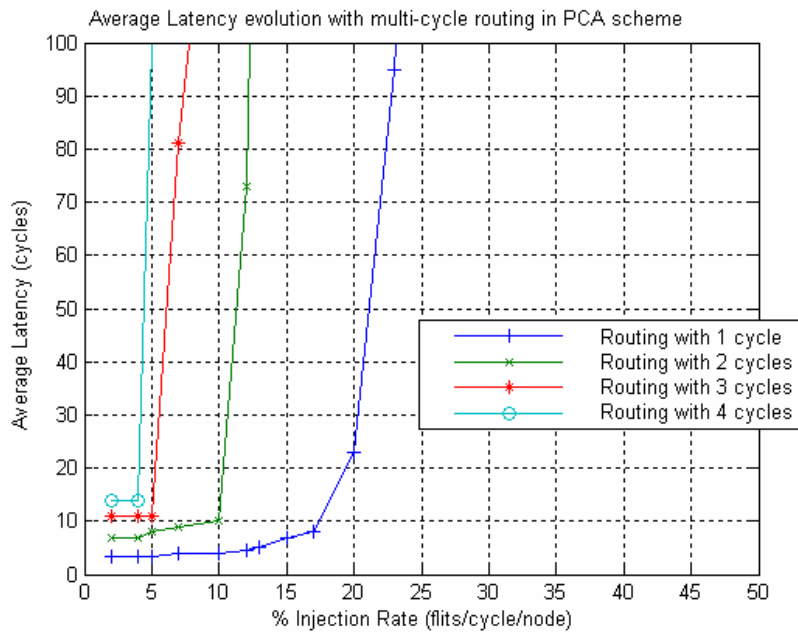
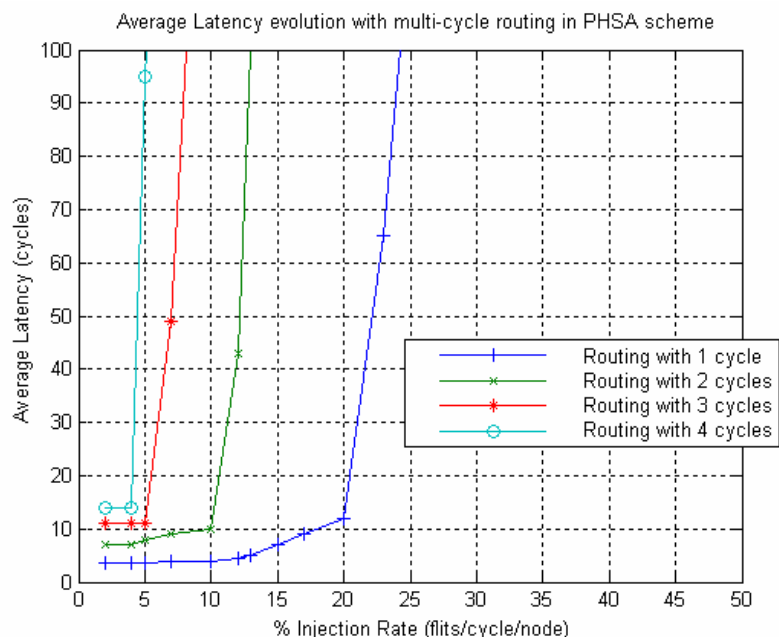


Figure 18

PHSA scheme evaluation



5. Implementation Evolution

5.2. FPGA Implementation

We implemented the 2D mesh 4x4 NoC with different routing schemes on FPGA Xilinx Virtex II P XC2VP100. The implementation on Xilinx Virtex-II-Pro is for hardware prototyping purpose with the XC2VP50 ff1152 board. We limit the NoC implementation at the 4x4 dimension because this FPGA contains in total: 23616 slices Flip-Flops and 47232 LUTs, however the *PHSA* occupies more than 80% of total cells. The chosen network data bus's width is 32 bits. Input FIFOs are implemented using registers in order to achieve better performances and power efficiency, e.g. we haven't used the RAM blocs from the component. The synthesis of five ports router with *PHSA* scheme consumes 1592 slices, 1207 slices Flip-Flops and 2785 LUTs.

The operating frequency is similar for the 2 algorithms: 85 MHz. Each port transmits 32 bit flits. Since each flit takes one clock cycle to be sent, then a five ports router presents a performance of 13,6 Gbits/s ($85 \text{ MHz} * 5 * 32$). In the two implementations, design results with *FA* routing algorithm are close to *PCA* and *PHSA* ones. The extra resources used with *PCA* or *PHSA* are negligible while the performance is consistently better at high NoC workload. Notice that all these interesting results are obtained with FIFO buffers size equal to 6 at each input port. Table 2 shows the synthesis results of NoC 4x4 with speed optimization.

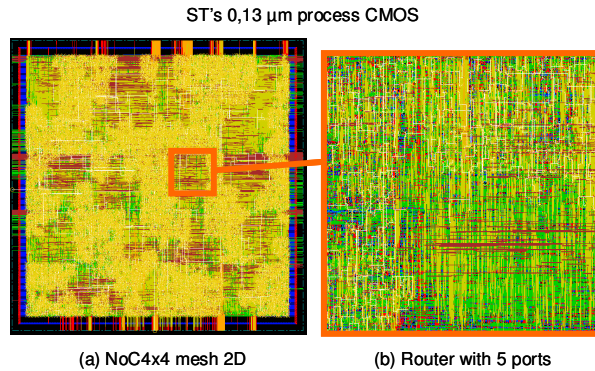
Table 2 FPGA Synthesis of NoC 4x4 for the 3 routing schemes

Routing Scheme	Resources	Used	Used/Total (%)
PHSA	Slices	18901	80%
	Slices Flip Flops	18103	38%
	LUTs	34377	73%
PCA	Slices	16997	72%
	Slices Flip Flops	16915	35%
	LUTs	32865	69%
FA	Slices	15989	67%
	Slices Flip Flops	14989	31%
	LUTs	32151	68%
Deterministic	Slices	14883	63%
	Slices Flip Flops	13938	29%
	LUTs	30122	63%

5.2. ASIC Implementation

In the ASIC design we used 130 nm CMOS process. The design has been synthesized using Synopsys tools. The automatic place and route has been performed using SOC Encounter tools from Cadence. The simulations (netlist and layout) have been executed using ModelSim from Mentor, using SDF timing annotations and back-annotations. The synthesis has been made for a simple router with five ports and for a NoC4x4 including 16 such routers. Figure19 shows the layout of both the router and the NoC. When running all traffic generators the performance limit of the adaptive routing algorithm (Det, FA, PCA and PHSA) is around 300 MHz.

Figure19 ASIC synthesis of MIC@R architecture with PHSA scheme



If only two random packet sources are enabled the maximum clock frequency reaches 370 MHz. So the throughput can be more than 48 Gbits/s per router. Table 3 shows the results of the ASIC synthesis area for all routing schemes. It shows the ASIC synthesis results of NoC 4x4 with speed optimization. We notice a small overhead of the PHSA implementation compared to others routing schemes while the performance is consistently better.

Table 3 ASIC Synthesis of NoC 4x4 for the 4 routing schemes

<i>Routing Scheme</i>	<i>Resources</i>	<i>Router</i>	<i>NoC4x4</i>
PHSA	Number of Cells	15632	261817
	Total Area (mm ²)	0.358	3.611
PCA	Number of Cells	15433	25253
	Total Area (mm ²)	0.351	3.312
FA	Number of Cells	15393	247213
	Total Area (mm ²)	0.344	3.132
Deterministic	Number of Cells	14381	238613
	Total Area (mm ²)	0.310	2.973

6. Conclusion and perspectives

In this work we evaluated a proposed architecture "MIC@R" by simulation and FPGA/ASIC implementation. MIC@R is a low latency router architecture suitable for networks-on-Chips. We also proposed a novel routing scheme called PHSA that presents the best performance in term of latency and throughput. Due to the low latency of our architecture, congestion information about each router's neighbouring routers is swiftly updated. It can be exploited to avoid deadlock states. Results also show that at high injection rate and under non-uniform traffic our architecture combined with PHSA routing reduces significantly overall network latency. Moreover we are focusing on implementing some processing smart homes applications, over this NoC architecture. In the future we are planning to explore some specific network topologies and efficient guaranteed Quality of Services (QoS) that can take advantage of our single-stage router architecture. It will implement a real time application using H.264 codec.

REFERENCES

- 1 W. J. Dally and B. Towles. "Principles and Practices of Interconnect Networks". In book ISBN: 0-12-200751-4 Elsevier 2004.
- 2 R. Ben Tekaya, A. Baganne, R. Tourk. "MIC@R: A Generic Low Latency Router for On-Chip Networks". In *proc IEEE ICECS'07*, pp: 999-1002, December 2007.
- 3 R. Ben Tekaya, A. Baganne, R. Tourk. and K. Torki "Design and Implementation of MIC@R Router of On-Chip Networks". In *proc IEEE IDT'08*, pp: 17-21, December 2008.
- 4 G.-M. Chiu. "The odd-even turn model for adaptive routing". In *IEEE Tran. On Parallel and Distributed Systems*, vol. 11, Issue 7, pp: 729-738, July 2000.
- 5 E. Rijpkema, K.G.W. Goossens, A. Rădulescu, J. Dielissen, J. van Meerbergen, P. Wielage, and E. Waterlander. "Trade Offs in the Design of a Router with Both Guaranteed and Best-Effort Service for Networks on Chip". In *proc DATE'03*, pp: 350-355, 2003.
- 6 R. Mullins, A. West and S. Moore. "Low-Latency Virtual-Channel Routers for On-Chip Networks". In *Proc ISCA.*, pp: 188-197, 2004.
- 7 R. Mullins, A. West and S. Moore. "The Design and Implementation of a Low-Latency On-Chip Network". In *Proc ASP-DAC*, pp: 164-169, January 2006.
- 8 J. Kim, D.Park, T. Theocharides, N. Vijaykrishnan, Chita R. Das. "A Low Latency Router Supporting Adaptivity for On-Chip Interconnects". In *proc DAC'05*, pp: 559-564, 13-17 June 2005.
- 9 M. Li, Q.A. Zeng and W.B. Jone. "DyXY-A Proximity Congestion-Aware Deadlock-Free Dynamic Routing Method for Network on Chip". In *proc DAC'06*, pp: 849-859, 2006.
- 10 J. Hu and R. Marculescu. DyAD – Smart Routing for Networks-on-Chip. In *DAC'04*, pp: 260-263, 2004.
- 11 T. Tao Ye, L. Benini, G. De Micheli, "Packetization and routing analysis of on-chip processor networks". In "Journal of Systems and Architecture", Elsevier, vol. 50, pp: 81-104. Issue 2-3 Feb. 2004.
- 12 L. M. Ni and P. K. McKinley. "A survey of wormhole routing techniques in direct networks". In *IEEE Tran. on Computers*, vol. 26, pp: 62-76, Issue 2 Feb. 1993.
- 13 M. L. Fulgham and Snyder. "Performance of Chaos and Oblivious Routers Under Non-uniform Traffic". *Technical report. Univ. of Washington. July 1993.*
- 14 L. Benini,, G. De Micheli. "Networks on chips: a new SoC paradigm". In *IEEE Computer*, vol.35, pp: 70-78, Issue 1, Jan. 2002.
- 15 W. J. Dally and B. Towles. "Route Packets, Not Wires: On-Chip Interconnection Networks". In *proc Design Automation Conference DAC'04*, pp: 684-689, 2001.
- 16 C.J. Glass and L.M. Ni. "Adaptive Routing in Mesh-Connected Networks". *Proc. Int'l Conf. Distributed Computing Systems*, pp:12-19, June 1992.
- 17 D.H. Linder and J.C. Harden. "An Adaptive and Fault-Tolerant Wormhole Routing Strategy for k-Ary n-Cubes". In *IEEE Trans Computers*, vol. 40, Issue 1, pp: 2-12, Jan. 1991.
- 18 W. J. Dally. "Virtual-Channel flow Control." In *IEEE Tran on Parallel and Distributed System*, vol.3, Issue 2, pp: 194-205, March 1992.
- 19 L. Schwiebert and D.N. Jayasimha. "Optimal Fully Adaptive Minimal Wormhole Routing for Meshes". In *ACM Journal of Parallel and Distributed Computing*, vol .27, Issue 1, pp: 56-70, May 2005.
- 20 J. Duato, S. Yalamanchili, L. Ni. "Interconnection Networks: An Engineering Approach". In book ISBN: 1-55860-852-4 , Elsevier 2003.
- 21 F.Moraes, N.Calazans, A. Mello, L. M. oller, L. Ost. "HERMES: an infrastructure for low area overhead packet-switching network on chip". In Elsevier, *INTEGRATION, the VLSI journal*, vol. 38, Issue 1, pp: 69-93, Oct. 2004.