Hermes-AA: A 65nm Asynchronous NoC Router with Adaptive Routing

Julian J. H. Pontes, Matheus T. Moreira, Fernando G. Moraes, Ney L. V. Calazans

Faculty of Informatics - FACIN, Pontifical Catholic University of Rio Grande do Sul - PUCRS
Porto Alegre, RS, Brazil
julian.pontes@pucrs.br, matheus.moreira@acad.pucrs.br, {fernando.moraes, ney.calazans}@pucrs.br

ABSTRACT

This work presents the architecture and ASIC implementation of Hermes-AA, a flexible fully asynchronous network on chip router employing an adaptive routing algorithm. Hermes-AA enables communication between router and synchronous processing elements. The ASIC implementation of the router employed standard CAD tools and a specifically developed library of standard cell components. Area and timing characteristics for 65nm technology attest the quality of the design, which displays a maximum aggregated throughput of 7.75 Gbits/s.

I.INTRODUCTION

Networks on Chip (NoCs) and asynchronous circuits are leaving the status of purely research topics and becoming mainstream approaches for enterprise design teams and computer aided design (CAD) tools. There are already commercial examples of SoCs using NoCs as interconnection architecture [1] and globally asynchronous locally synchronous (GALS) NoC-based SoC designs [2].

The main advantages of NoC-based SoCs are higher level of communication parallelism and design scalability, when compared to bus-based SoCs. Asynchronous circuits in turn, appear as a solution to the limitations that the synchronous strategy is facing in advanced technology nodes, like clock skew and clock tree power dissipation.

Considering the advantages mentioned above, the adoption of asynchronous NoCs seems a natural trend as an interconnection solution for GALS systems. Accordingly, several works about asynchronous NoCs have been proposed in the literature. However, there are NoC design techniques that have not yet been covered in asynchronous implementations like adaptive algorithms. Actually, most asynchronous NoCs employ source-based routing to simplify the router architecture.

Adaptive algorithms are capable to provide more parallelism in the communication than source based routing and distributed but deterministic algorithms.

Globally Asynchronous Locally Synchronous (GALS) design techniques may help overcoming limitations of synchronous design while maintaining a mostly synchronous design flow [3]. GALS techniques simplify the task of reaching the overall timing closure for SoCs, but typically require the addition of synchronization interfaces between each pair of communicating modules.

The rest of this paper is divided into four Sections. Section II describes related work and positions the new proposition with regard to it. Section III describes the architecture of the Hermes-AA router, while Section IV discusses the ASIC implementation of Hermes-AA. Section V presents conclusions and directions for further work.

II.RELATED WORK

Most real life implementations of SoCs using NoCs rely on GALS SoC design, but these currently suggest the use of fully synchronous techniques to build NoC routers. However, during the last decade there has been a small, yet steady movement towards research and implementation of fully asynchronous routers and corresponding NoCs. An encompassing review of the state of the art for fully asynchronous NoCs appeared in [4]. Table 1 shows part of that review, adding data about the Hermes-AA NoC proposed herein. Suppressed data from the original reference include historical propositions of asynchronous interconnects and FPGA prototyping of asynchronous NoCs hardly useful in real life systems. The first five NoCs/routers in Table 1 (QoS, MANGO, asynchronous As. QNoC, ANoC and ASPIN), Hermes-A and Hermes-AA all propose ASIC implementations of routers and links for 2D mesh topologies.

978-1-4244-6683-2/10/$26.00 ©2010 IEEE

Table 1 – A partial comparison of fully asynchronous interconnection networks and/or routers for GALS SoCs. Legend: **A2S, S2A** – Async. to Sync./Sync. to Async., **As.** -Asynchronous, **BE** – Best Effort service, **DI** - delay insensitive, **GS** – guaranteed service, **Ir-reg/Reg** - Irregular/Regular, **N.A.** - Information Not Available, **OCP** – Open Core Protocol, **VC** – virtual channel.

Characteristics → NoC	Topology	Routing / Flow Control	Network Interface	Asynchronous Style	Links and encoding	Implementation
QoS [5]	2D Mesh 4 3GS/1BE VCs	XY / wormhole, credit-based	N.A.	QDI	1-of-4 DI / 8-bit flits	Simulation only
MANGO [6]	2D Mesh (Ir-reg/Reg) 4GS/1BE VCs	Source / N.A.	A2S, S2A, OCP	4-phase bundled-data	Dual-rail, 2-ph. DI / 33-bit flits	130nm, 650Mflits/s, ASIC
As. QNoC [7]	2D Mesh (Ir-reg/Reg) 8VCs	Source / wormhole, credit-based with preemption	N.A.	4-phase bundled-data	10-bit flits	180 nm, 200Mflits/s, ASIC
ANoC [8]	2D Mesh (Ir-reg/Reg) / 2 VCs	Source, odd-even / wormhole	A2S, S2A FIFOs	QDI	34-bit flits	65nm, 550Mflits/s, ASIC
ASPIN [9]	2D Mesh (Reg)	Distributed XY / worm-hole, EOP	A2S, S2A FIFOs	Bundled-data	Dual-rail, 4-ph., 34-bit flits	90nm, 714Mflits/s
Hermes-A [4]	2D Mesh (Reg)	Distributed XY / worm-hole / BOP-EOP	Dual-Rail SCAFFI [10]	Dual-rail / bun-dled data	Dual-Rail	180nm, 727Mbits/s, ASIC
Hermes-AA	2D Mesh (Reg)	Distributed, West first, Adaptive / wormhole, BOP-EOP	Dual-Rail SCAFFI [10]	Dual-rail / bun-dled data	Dual-Rail	65nm, 7.1Gbits/s, ASIC

In some cases there is mention to adequacy to support other topologies as well. This is not the case for ASPIN, because of the chosen router organization. In this NoC the router ports are distributed around the periphery of the PE, making inter router links small compared intra router links. This facilitates connection of PEs by abutment, but prevents easy use of topologies other than 2D mesh. Even a similar 2D torus is problematic to build in this case.

Four of the NoCs (QoS, MANGO, asynchronous QNoC, ANoC) claim support to quality of service through the use of virtual channels and/or special circuits (GS routers). ANoC is the most developed of the proposals and presents the best overall performance. It has been successfully used to build at least two complete integrated circuits [11]. However, most of the characterization for ANoC (and for other asynchronous NoCs) derives from a detailed knowledge of the application in sight. If the application has unpredictable dynamic behavior, a more flexible approach to topology choice, routing and incorporating the capacity to take decisions based on dynamic information of the network can be fundamental. These are some reasons behind the proposal of Hermes-A and Hermes-AA, the last of these described in the next Sections. For details on Hermes-A the reader may refer to [4].

Concerning adaptive routing schemes, these allow that packets take different paths through the network every time congestion is detected. They must do so while ensuring deadlock freedom. Classical examples of adaptive routing algorithms proposed for NoCs are those based on the turn model, originally proposed for multiprocessors [12] and odd-even routing [13]. These algorithms do not specify the mechanism by which a particular path is selected, and several policies can be chosen for that. In NoCs, designers usually opt for the simplest schemes, like using a fixed order to choose alternative paths. More complex schemes may include the use of randomization [14]. Also, the information used to take the routing decisions in case of congestion detection may use only local information, where the first free output port is selected in a router, or use an extended neighborhood, or even base decisions on global network congestion information [15]. In this work, to keep design complexity restricted to the asynchronous design techniques, a very simple routing scheme is employed: the use of the deadlock-free West-first turn model algorithm, with fixed order for alternative paths and with only local information for used for decision making.

III.THE HERMES-AA ROUTER

Unlike most other asynchronous routers, Hermes-AA employs a distributed routing scheme, where the router itself decides which path incoming packets will follow. The coupling of this scheme to adaptive routing algorithms allows solving network congestion problems in real time. Another charac-

teristic of Hermes-AA is that it uses independent arbitration at each port. Distributed routing and scheduling are characteristics shared by e.g. Hermes-AA and the ASPIN NoC.

A traditional 2D mesh topology NoC with wormhole packet switching is the test environment used to validate the Hermes-AA router. Each router in the experimental setup comprises up to five ports: East, West, North, South and Local. As usual in direct NoCs, the Local port is responsible for the communication between the NoC and the router local PE. Although the design framework developed for this project has a parameterizable flit width, to facilitate discussion all experiments described herein assume the use of 8-bit flits. The packet format is rather simple: the first flit contains the XY address of the destination router and subsequent flits contain the packet payload. Two sideband signals control the transfer of arbitrarily-size: begin of packet (BOP), activated with the first flit of a packet and end of packet (EOP), activated with the last flit. All intermediate flits have BOP=EOP=0.

Basically, the router architecture employs a delay insensitive (DI), 4-phase, dual-rail encoding. Each input port interface consists of 21 wires: 16 wires carry the 8-bit dual-rail flit value (DR-Data), four wires contain the dual-rail BOP and EOP information and the last is a single rail acknowledge signal. The router detects data availability when every pair of wires that define each bit value in the DR-Data signal is distinct from “00”. I.e. the all zeroes value in DR-Data is the *spacer* for the DI code. More about the employed DI design style is available in classical textbooks like [16].

A. Input Port

Figure 1 depicts the Hermes-AA input port structure as a simplified asynchronous data-flow diagram [16]. There are three alternative paths in this module, one used for the first flit (1), one for intermediate flits (2) and one for the last flit (3). In Figure 1 two wires represent each bit. Thus, a 10-bit path is in fact a 20-wire bus.

When BOP is signaled at the input port, the first demux selects the path that feeds the module responsible for computing the path to use. This module receives ten information bits that are forwarded (8 data bits plus EOP and BOP), plus four destination bits using dual-rail one-hot encoding. Note that just the bit associated to the selected path is enabled in this 4-bit code.

Since the routing decision must be kept for all flits in a packet, a loop was added to register the decision. The loop appears in Figure 1 as a chain of three asynchronous registers (4) in order to enable the data flow inside the 4-phase dual-rail loop. Each two successive asynchronous stages communicate using an handshake operations [16]. Thus, in this kind of circuit it is not possible that three successive stages exchange two data simultaneously. Exactly three stages are then the minimum necessary to propagate information circularly. Less than three stages incur in deadlock. This can be better understood remembering that between every two valid data there is always a spacer, and that before propagating a spacer the first data must be copied to the next stage.

After computing the output port where to send the incoming flit, the rightmost module in Figure 1 (Output demux) chooses the right path to send the flit, based on the 4-bit routing information. Subsequent flits in a packet go through the lower output of the leftmost demux and are input to a second demux after the fork element. This demux looks for the EOP bit before choosing the right direction for each flit. If there is no EOP indication the flit follows path (2) to the first merge component. Otherwise, the S-Control module is used. When the S-Control receives the last flit, it propagates this flit to output A. When transmission finishes, the S-Control starts a data transfer at output B. This is required to change the output port involved in packet transfer, achieved sending a flit with BOP = EOP = 1. The output port will then discard this flit and release the arbiter solicitation.

a) Adaptive Path Calculation

Figure 2 depicts the route computation architecture. In some direct 2D topologies like 2D meshes or tori, each router is usually identified by its X-Y coordinates. The first flit of a packet carries the destination X address in the less significant bits and the destination Y address in the most significant bits. When a flit is accompanied by an active BOP signal, it feeds the Path Calculation module. It arrives at the input of a completion detector (CD). Detection of a valid dual rail data token causes the propagation of the actual router X-Y coordinates to two subtraction circuits and starts the sampling of the output ports state. The results of the subtraction and the output port states will determine the path a packet must follow.

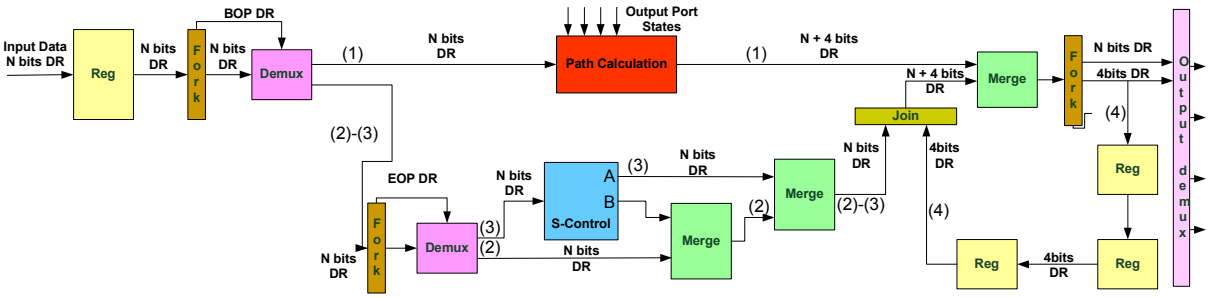

Figure 1: Hermes-AA router input port architecture. All paths employ dual-rail encoding.

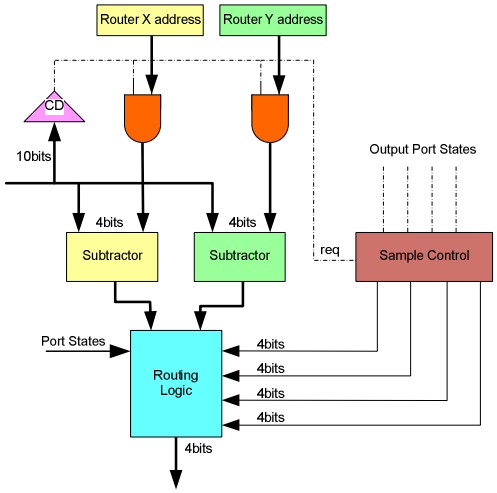

Figure 2: Hermes-AA route computation circuit.

Table 2 shows an example decision table for the West Input Port. To determine the destination output port, the Routing Logic verifies if the subtraction results are zero (XZ - YZ) and the subtraction signals (XS - YS).

Table 2 – West First decision table.

XZ	YZ	XS	YS	Target
0	0	0	0	North/East
0	0	0	1	South/East
0	1	0	-	East
1	0	-	0	North
1	0	-	1	South
1	1	-	-	Local

If both subtractions are 0, the packet reached the target router and proceeds to the Local port. In the case there are two possible destinations (first two rows in the Table), route computation takes into account the output router states to decide. The output port state is obtained using an OR of all arbiter requests for each port (displayed in Figure 4). In this way, the West Input Port receives the state of all other Output ports, indicating if they are able to accept a new connection. The overall performance is practically the same as for the XY algorithm for packets with several flits. The main difference is the time spent to calculate the destination.

Figure 3 shows a circuit that was developed to sample the current state of an output port. This control circuit is needed because the state of the output port can change while the path is under calculation. A metastability filter can be used to avoid the “11” value at the output of the sampler.

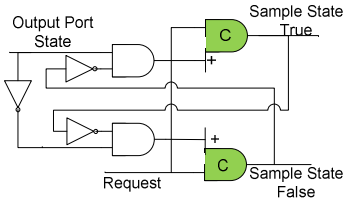

Figure 3: State sampler circuit.

B. *Output Port*

In the Hermes-AA router each output port receives four data flows. For instance, Figure 4 shows the Local output port structure, that receives data from input ports North, South, East and West.

An arbiter circuit controls the behavior of each output port. This arbiter achieves fairness with a structure of six 2-input, 2-output arbiters connected in a shuffle-exchange topology. Each atomic arbiter decides which request to serve from between two input requests, using a first-come-first-served strategy. This allows the processing of up to four simultaneous input port requests. The bit used to produce the request to the output port is that produced by the logic that computes routing on the input port. Since this bit is a dual-rail representation, conversion to single-rail is necessary, since arbiters are the only single-rail module in the output port. A 2-input C-element with one negated input executes the conversion.

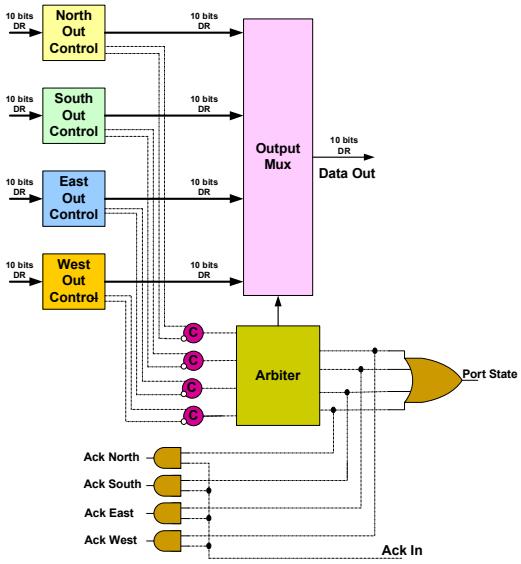

Figure 4: Local output port structure. Dashed lines represent actual wires. Solid lines represent dual-rail encoded lines.

IV.RESULTS

The target technology was the 65nm STMicroelectronics general purpose standard *Vt* with an asynchronous standard cell library designed for asynchronous circuits in the same technology.

The area and power results are extracted from physical synthesis after RC extraction. The timing results are extracted from timing simulation. The power analysis was done based on the switching activity extracted from the simulation. The operating conditions used were 25°C, 1 Volt.

Figure 5 describes the relationship between throughput of a link of router and the IP frequency.

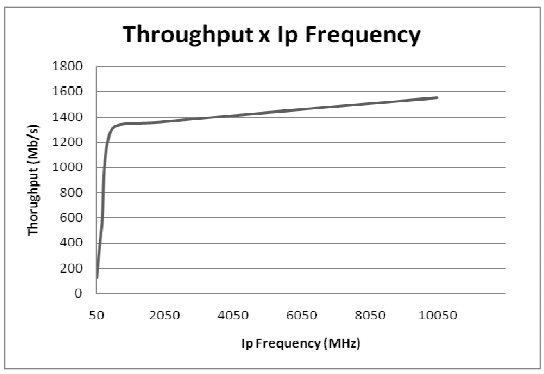

Figure 5: Eight-bit router throughput versus IP Frequency.

For the 8-bit router it was possible to extract the maximum throughput that a router link can deliver, 1.55 Gbits/s. Since a single router can support up to five transmissions in parallel, the maximum router throughput is 7.75Gb/s. This value can be improved with using asynchronous FIFOs at the input ports. Simulation results show that when the router operates with a FIFO, the throughput can reach up to 6.3Gb/s per port. It is important to note that the router throughput goes back to the saturation value (1.55 Gbits/s) when FIFOs are full.

For the employed operating conditions the Input Port latency is 3.709ns and the Output Port latency is 1.351ns. Latency results were obtained when the router is empty. Thus it is a lower bound.

Table 3 compares area results of the Hermes-AA router (using West First router) with an implementation using the XY routing algorithm. It shows that the use of the adaptive algorithm does not impact router area significantly.

Table 3 – West First and XY Hermes-AA routers comparison.

Algorithm	Standard Cell Area	Total Area
WF	76455 μm^2	116034μm^2
XY	75133 μm^2	114456 μm^2

Figure 6 shows a first traffic scenario built to display the benefits of using an adaptive algorithm. In this scenario, the Local and West ports send 9-flit packets in burst to two IPs located to the right and above the current router.

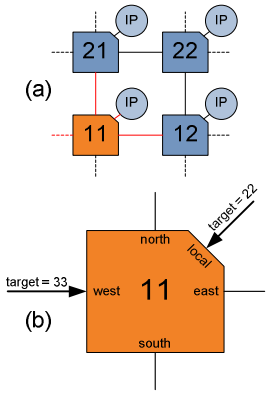

Figure 6: Scenario for evaluating a Hermes-AA router throughput variation with the IP Frequency.

Table 4 shows that the West First algorithm presents a gain of about 71% over the XY in throughput.

Table 4 – West First and XY Throughput comparison.

	XY	WF
Throughput	10Mpackets/s	17Mpackets/s

Power characterization was performed using a traffic scenario where all input and output ports are working in the saturation region of the router. Figure 7 shows the power dissipation of each module for the 8-bit phit Hermes-AA router.

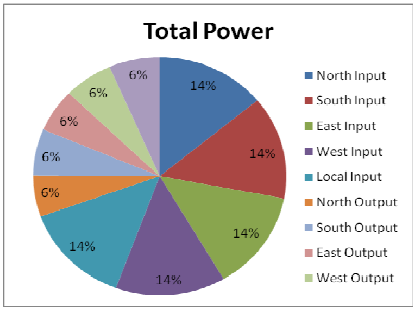

Figure 7: Eight bit router modules proportional power.

Finally, Figure 8 evaluates the distinct components of power for the same scenario. For this particular scenario, the worst component is the internal power dissipation. However, the main concern refers to leakage power dissipation since in normal operation, with less congestion, the switching and internal power of the router should reduce but the leakage power dissipation will remain the same.

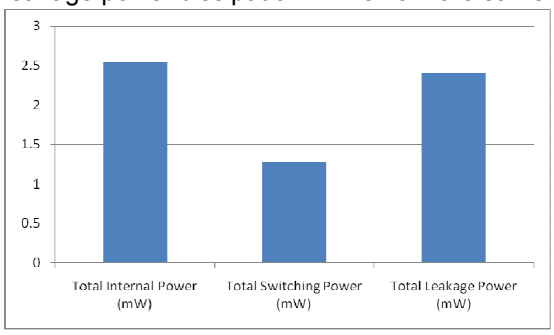

Figure 8: Router power characterization.

V.CONCLUSIONS AND FUTURE WORK

Hermes-AA demonstrates that asynchronous circuits can be employed as communication architecture for a high performance GALS complex System on a Chip. Ongoing work proceeds in several directions including: (1) support to other turn model adaptive routing algorithms, like the North last and Negative first; (2) enabling Hermes-AA to work with multiple supply voltages and power shutoff features, to reduce the power utilization in idle ports; (3) implementing other NoC topologies such as 2D torus. It is important to note that in the case of a 2D torus, the routing module has to be modified, since pure XY or pure West first routing algorithms are not deadlock-free for this network topology.

ACKNOWLEDGEMENTS

The Authors acknowledge the support of CNPq through research grants 551473/2010-0, 309255/2008-2, and 301599/2009-2. Also, they acknowledge the National Science and Technology Institute on Embedded Critical Systems (INCT-SEC) for the support to this research.

REFERENCES

[1] S. Bell et al., "TILE64™ Processor: A 64-Core SoC with Mesh Interconnect," In: *Int. Solid State Circuits Conf. (ISSCC'08)*, pp. 87-89, Feb 2008.

[2] S. Vangal et al., "An 80-Tile Sub-100-W TeraFLOPS Processor in 65-nm CMOS," *IEEE Journal of Solid-State Circuits*, 43(1), pp. 29-41, Jan 2008.

[3] D. Chapiro, "Globally-Asynchronous Locally Synchronous Systems," PhD Th., Stanford University, Oct 1984, 134 p.

[4] J. Pontes, M. Moreira, F. Moraes, and N. Calazans, "HERMES-A - An Asynchronous NoC Router with Distributed Routing." In: International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS'10), Grenoble, 2010. Accepted for publication.

[5] T. Felicijan and S. Furber, "An Asynchronous On-Chip Router with Quality-of-Service (QoS) Support," In: *17th IEEE Int. SoC Conf. (SOCC'04)*, pp. 274-277, 2004.

[6] T. Bjerregaard, M. Stensgaard, and J. Sparsø, "A Scalable, Timing-Safe, Network-on-Chip Architecture with an Integrated Clock Distribution Method," In: *Design, Automation, and Test Europe (DATE'07)*, pp. 1-6, Apr 2007.

[7] R. Dobkin, R. Ginosar, and A. Kolodny, "QNoC asynchronous router," *Integration the VLSI Journal*, 42(2), pp. 103-115, Feb 2009.

[8] E. Beigné, F. Clermidy, P. Vivet, A. Clouard, and M. Renaudin, "An Asynchronous NoC Architecture Providing Low Latency Service and its Multi-level Design Framework," In: *IEEE Int. Symp. on Asynchronous Circuits and Systems (ASYNC'05)*, pp. 54-63, 2005.

[9] A. Sheibanyrad, A. Greiner, and I. Miro-Panades, "Multi-synchronous and Fully Asynchronous NoCs for GALS Architectures," *IEEE Design and Test of Computers*, 25(6), pp. 572-580, Nov-Dec 2008.

[10] J. Pontes, R. Soares, E. Carvalho, F. Moraes, and N. Calazans. "SCAFFI: An intrachip FPGA asynchronous interface based on hard macros," In: *Int. Conf. on Computer Design (ICCD'07)*, pp. 541-546, 2007.

[11] Y. Thonnart, P. Vivet, and F. Clermidy, "A Fully *Asynchronous* Low-Power Framework for GALS NoC Integration," In: *Design, Automation*, and *Test Europe (DATE'10)*, pp. 33-38, 2010.

[12] C. Glass and L. Ni, "The turn model for adaptive routing," *Journal of the ACM*, 41(5), pp. 874-902, Sep 1994.

[13] G.-M. Chiu. "The Odd-Even Turn Model for Adaptive Routing," IEEE Transactions on Parallel and Distributed Systems, 11(7), pp. 729-738, Jul 2000.

[14] M. Cho, M. Lis, K. Shim, M. Kinsy, and S. Devadas, "Path-Based, Randomized, Oblivious, Minimal Routing," In: *2nd International Workshop on Network on Chip Architectures (NoCArc'09)*, pp. 23-28, Dec 2009.

[15] L. Tedesco, F. Clermidy, and F. Moraes, "A monitoring and adaptive routing mechanism for QoS traffic on mesh NoC architectures," In: *7th IEEE/ACM international conference on Hardware/software codesign and system synthesis (CODES+ISSS'09)*, pp. 109-118, Oct 2009.

[16] J. Sparsø and S. Furber, "Principles of Asynchronous Circuit Design – A Systems Perspective". Kluwer Academic Publishers, Boston, 2001. 354p.