A Low Cost Network-on-Chip with Guaranteed Service Well Suited to the GALS Approach

Ivan MIRO PANADES
STMicroelectronics
Minatec Bat. BOC
17, avenue des Martyrs
38054 Grenoble, France
ivan.miro-panades@st.com

Alain GREINER
Universite Pierre et Marie Curie
4, place Jussieu
75252 Paris, France
alain.greiner@lip6.fr

Abbas SHEIBANYRAD
Universite Pierre et Marie Curie
4, place Jussieu
75252 Paris, France
abbas.sheibanyrad@lip6.fr

***Abstract*- The paper presents the DSPIN micro-network, that is an evolution of the SPIN architecture. DSPIN is a scalable packet switching micro-network dedicated to GALS (Globally Asynchronous, Locally Synchronous) clustered, multi-processors, systems on chip. The DSPIN architecture has a very small foot-print and provides to the system designer both guaranteed latency, and guaranteed throughput services for real-time applications.**

Keywords- DSPIN, SPIN, Network on Chip, NoC, System on Chip, SoC, Globally Asynchronous Locally Synchronous, GALS, Mesochronous, Bi-synchronous FIFO.

1. Introduction

The SPIN micro-network architecture was the first published [1] attempt to solve the bandwidth bottleneck, when interconnecting a large number of IP cores in multi-processors SoC (System-on-Chip). After this, a large number of NoC (Network-on-Chip) architectures have been published [5,6,17,22,23]. The ÆTHEREAL [5] micro-network designers insisted on the importance of QoS (Quality of Service), such as guaranteed throughput, bounded latency, or controlled jitter, in applications with real time constraints [21].

Simultaneously, a physical implementation of a 32 ports SPIN [19] micro-network by STMicroelectronics helped to identify several limitations of the initial SPIN architecture: The fully synchronous approach of the SPIN network is not compatible with the GALS (Globally Asynchronous, Locally Synchronous) paradigm. The SPIN fat-tree topology has the best theoretical diameter, but lack of modularity and flexibility for an industrial product. The adaptive routing algorithm improved the saturation threshold, but destroyed the “in-order delivery” property, and increased the router complexity. Finally, the hard macro-cell approach of SPIN was not suited to a standard cells based design flow.

As SPIN, the DSPIN micro-network has been designed to support shared memory multiprocessor architectures. However, DSPIN had new requirements: It must be suited to the GALS paradigm, it must be synthesizable with conventional synchronous design flows, it must be really scalable, it must provide a predictable latency & throughput, and have a smaller footprint than SPIN...

In Section 2 we analyze briefly the state of the art in NoC architectures. We present in Section 3 the DSPIN architecture. Finally, the Section 4 contains experimental results, regarding both the performances and the silicon area.

2. BACKGROUND

Various architectures have been proposed for interconnecting dozens of IP cores with the aim to get good performances and small foot-print.

Most of the first published NoC architectures were based on the packet switching paradigm. It is well known that packet-switched networks (similarly to conventional shared busses) provide basically a best effort (BE) service, with no guaranty of bounded latency or throughput.

As the BE is not adequate for real-time applications, in some NoC, the packets are labeled with priority bits. That allows assigning higher priority to the most critical communications [17]. However, this is not enough, because “hard” guaranty of service requires some form of resource reservation (such as the Time Division Multiplexing approach in time-slotted busses).

Some networks introduced the guaranteed service (GS) like ÆTHEREAL [5], NOSTRUM [22] and MANGO [6]. ÆTHEREAL uses the Time Division Multiplexing (TDM) technique to allocate slots for the GS communications. NOSTRUM uses the concept of looped containers in a TDM fashion to route the GS packets. Both of them require a synchronous implementation of the SoC because of the TDM synchronicity. The MANGO approach is a fully clockless network connected to clocked IP cores. This approach allows a GALS implementation and supports BE and GS by means of virtual channels. Asynchronous approaches are promising but the testability issues and the lack of commercial tools slow down the introduction of these techniques. In this paper, we present a low cost, synthesizable (in standard synchronous design flow) NoC providing GS and BE communications and compatible with the GALS approach.

3. DSPIN ARCHITECTURE

DSPIN is a micro-network dedicated to shared memory, clustered, multiprocessors architectures. The complete system on chip is supposed to be a composition of synchronous sub-systems. Each sub-system (or cluster) contains one or several processors, one or several physical memory banks, optional IP cores such as hardware coprocessors, or I/O controllers, and a local interconnect. Even if the architecture is physically clusterized, all processors, in all clusters share the same “flat” address space, and any processor in the system can address any target or peripheral. Each subsystem (i) can be clocked by a different clock signal CK_i, and all CK_i signals can be fully asynchronous (regarding both the frequency and the phase).

3.1 Distributed clustered architecture

DSPIN means Distributed, Scalable, Predictable, Interconnect Network. In order to avoid deadlocks in request/responses traffic, DSPIN contains two fully separated sub-networks for requests and responses. Both request and response sub-networks are distributed: each cluster contains two local routers (one for the requests, one for the responses), as well as one network interface controller (NIC). Figure 1 shows a generic cluster architecture. The routers are the switching modules of the network. The network controller adapts the network protocol to the local interconnect protocol.

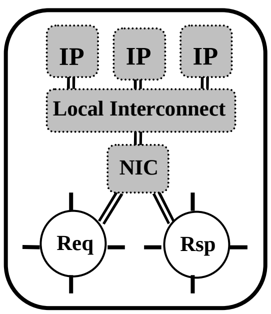

Figure 1. Typical cluster architecture

The IPs are connected to the NIC through the local interconnect, which is the only access to the network. The topology of the network is organized in a two dimension mesh distribution of clusters as shown in Figure 2. Each cluster is connected to the north, south, east and west neighbors by means of point-to-point links. The communication between IPs in different clusters is done by traveling through as many routers as necessary (more precisely N+1 routers, if N is the Manhattan distance between the communicating clusters).

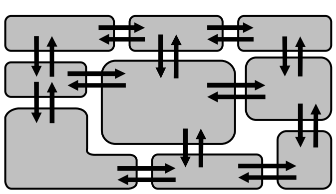

Figure 2. Network topology

The physical links between routers are implemented with FIFOs (black arrows in Figure 2). The mesh topology simplifies the routing algorithm, and strongly minimizes the silicon area of the switching hardware. There is no constraint on the size or shape of clusters. Just the mesh topology has to be respected, to guarantee the routing path between all the clusters.

3.2 Routing algorithm

As SPIN, DSPIN is a packet-switched network. Packets are divided into flits. A flit is the smallest flow control unit handled by the network. The first flit of a packet is the head flit and the last flit is the tail. The size of the flits is 34 bits word (32 bits for DATA and 2 bits for control). The two control bits are Begin of Packet (BOP), and End of Packet (EOP). BOP is set on the head flit, and EOP is set on the tail flit. The head flit includes the destination cluster address in the DATA field. This “cluster address” is defined in absolute coordinates X and Y, encoded on 4 bits each one, allowing a maximal 16 * 16 = 256 clusters topology.

When a router receives the first flit of a packet, the destination field is analyzed and the flit is forwarded to the corresponding output port. As DSPIN uses wormhole routing [15,16], the rest of the packet is also forwarded to the same port until the tail flit.

DSPIN uses the determinist and dead-lock free X-first [7] algorithm to route the packets over the network. With this algorithm, the packets are first routed on the X direction and then on the Y direction. The deterministic property of the X-first algorithm guarantees the "in-order delivery" of the network. The X-first is actually used for the request packets, but we use Y-first on the response packets, in order to guaranty the same path for the request and the response, and also maximize the number of GS communications.

3.3 Mesochronous clock distribution

Clock distribution in synchronous systems becomes a major issue [18] and the dissipated power becomes non negligible [20]. On the other hand, most existing IPs relies on synchronous design. The Globally Asynchronous, Locally Synchronous (GALS) approach [2,3] is a way to solve this problem. In the GALS approach, synchronous islands, or sub-systems communicate asynchronously. In DSPIN, the synchronous islands are the clusters and the asynchronous communications are carried out by bi-synchronous FIFOs, which are described hereinafter.

To maximize the throughput of the network, make the network latency predictable and be compatible with a GALS approach, the entire network is clocked by a mesochronous clock distribution, where all the routers have different clock signals CK_R_i, with the same frequency but different phase. The phase shift (skew) between the clocks signals CK_R_i and CK_R_{i+1}, in neighbor clusters, can be large but is bounded. The bi-synchronous FIFOs between routers handle this bounded clock skew to exchange data safely. The distribution of a mesochronous frequency along the entire circuit is cheaper in terms of design effort and power consumption than a synchronous one due to the unnecessary tree balance.

In order to generalize the network and accept heterogeneous IPs, each IP can have its own clock frequency CK_IP_i, without any relation to the local router clock CK_R_i. The bi-synchronous FIFOs between the NIC and the router do the adaptation between CK_IP_i and CK_R_i.

3.4 The long wire issue

In deep sub-micron processes, the largest parts of the delays are related to the wires. In multi-million gates SoCs, the timing closure can become a nightmare, as place & route tools have difficulties to cope with long wires. The DSPIN architecture is an attempt to solve this problem by partitioning the SoC into isolated clusters.

As shown in Figure 3, the DSPIN router is not a centralized macrocell: it is split in 5 separated modules (North, East, South, West & Local), that are physically distributed on the clusters borders. This feature, combined with the mesh topology allows us to classify the network wires in two classes:

- Inter-cluster wires: connecting modules of adjacent clusters. Example: the East module of cluster (Y,X) is connected to module West of cluster (Y,X+1). As those components can be made very close from each other, inter-cluster wires are short wires.
- Intra-cluster wires: connecting modules of the same cluster. Example: West module connects to North, South, East and Local modules in a tree manner. Those wires are the long wires, but the wire length is bounded by the physical area of a given synchronous domain, the cluster.

These properties allow synthesizing, placing and routing each cluster as an independent module. Moreover it relies on standard synchronous design flow without any time constraints between other clusters.

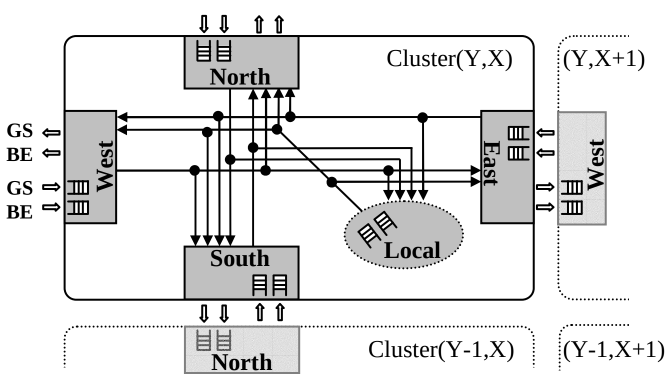

Figure 3. Distributed router architecture in the cluster

3.5 Guaranteed Service

DSPIN supports two types of traffic: Best Effort (BE) and Guaranteed Service (GS). GS is intended to be used on real-time or latency sensitive applications where latency, throughput or jitter is a major issue. In a fully synchronous system, the GS can be obtained by allocating slots on a Time Division Multiplexing (TDM) channel. However, TDM is difficult in GALS systems, where different IPs or clusters can run at different frequencies. We use a virtual channel (VC) approach to introduce the guaranteed services. Virtual channel allows sharing a physical resource between logically independent channels. In DSPIN, the shared resources are not the inter-cluster wires neither the bi-synchronous FIFOs, but are the intra-clusters long wires and multiplexers. As depicted in Figure 3, there are separate bi-synchronous FIFOs for BE and GS traffic. Moreover, no traffic storage is done between FIFOs. Therefore, the storage resources are fully separated for BE and GS traffic guaranteeing traffic independency. A packet is considered GS if it is injected on a GS port and is considered BE if is injected on a BE port.

Finally, in each module (North, East, South, West & Local), three multiplexers (two output multiplexers and one input multiplexer) compose the switching hardware, which are controlled by three state machines. As shown in Figure 4. Due to the X-first routing algorithm, the output multiplexers for the East and West modules are reduced to simple (2 inputs to 1 output) multiplexers. (For example the packets coming from North port cannot be routed to East nor West port).

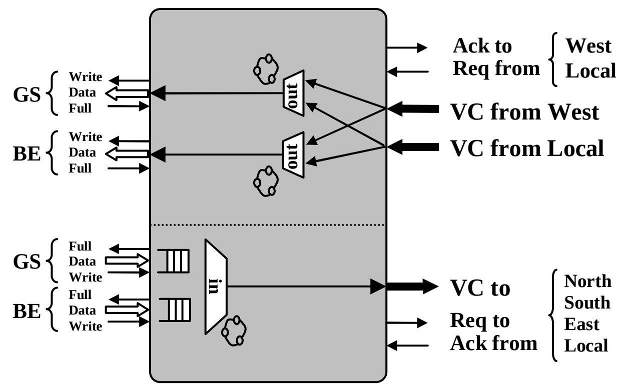

Figure 4. West router module detail

A virtual channel interconnects one input multiplexer to the output multiplexers of other modules. Over the virtual channel the BE and GS packets are sent in a dynamic TDM fashion. TDM approach is possible because the router is fully embedded in a synchronous domain. The allocation of the dynamic TDM slots is round-robin, to ensure that each type of traffic (GS & BE) can obtain 50% of the total bandwidth if required. If the GS FIFO is empty, all the slots (100% of the bandwidth) are allocated to the BE FIFO, and reciprocally if BE FIFO is empty. This feature maximizes the utilization of the virtual channel.

The output multiplexer is the switching unit of the router. It selects the data from one input VC and it writes in the corresponding output FIFO. The allocation policy of this multiplexer is controlled by request (Req) and acknowledge (Ack) signals and it is round-robin to guaranty equity between VC. The request and acknowledge signals have separated bits for BE and GS traffic.

We have described until now how to handle two separated traffics on the same switching hardware. In order to guaranty an upper bound for the latency, and a lower bound for the throughput in the GS sub-network, we must guaranty that there will never exist collisions in the GS sub-network (i.e. different GS traffic will not be allocated the same path). This requires some sort of end-to-end resource reservation (circuit switching). Following the Amdahl law, we do not want to pay hardware for un-frequent cases, and the end-to-end GS channel allocator is not implemented in hardware. For most embedded applications, the communication scheme is well known, and the system designer can statically allocate the required (non conflicting) GS channels. If static allocation is not possible, a GS channel allocator is implemented as a software task that will manage a global table of all existing GS paths, and perform dynamic allocation as required by the embedded software application.

3.6 Network Interface Controller

The DSPIN Network Interface Controller interconnects the request and response routers to the local sub-system. The main tasks are protocol conversion and packet building. The NIC provides services at the transport layer on the ISO-OSI reference model, offering to the local sub-system independency versus the network implementation. The DSPIN network features (no packet lost, in-order delivery, dead-lock free as mentioned in section 3.2) simplify the NIC hardware implementation. We have implemented a NIC model compatible with the OCP/VCI [4] protocol, but it can be easily adapted to any shared memory and transaction-based protocol.

Transaction-based protocols are composed of initiator IPs that issue requests and target IPs that returns responses. The DSPIN NIC being a bi-directional bridge, behaves as an initiator and as a target. It controls up to 4 communication channels, two for the initiator (BE & GS) and two for the target (BE & GS).

One task of the NIC is to translate the MSB bits of the 32 bits VCI/OCP address into a cluster address (Y,X) to route the packets over the network. This is done by a look up table (LUT) that must be replicated in each NIC as shown in Figure 5.

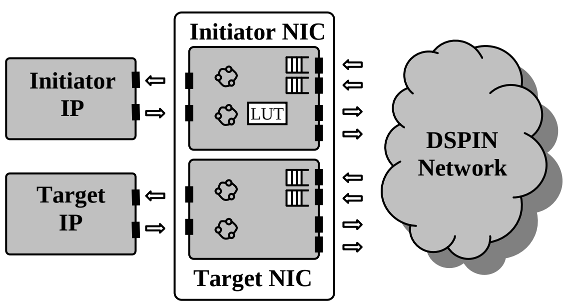

Figure 5. Network Interface Controller

3.7 Bi-synchronous FIFOs

The physical links between two routers and the physical links between a router and the NIC are interconnected with bi-synchronous FIFOs. These FIFOs have two functionalities: buffering the data and interfacing different clock domains. Interfacing two synchronous domains is not a trivial exercise [10], the metastability situations can induce a system failure. The metastability cannot be suppressed, but failure probability (typically expressed in terms of Mean Time Between Failures [9]) can be bounded to an acceptable value by a carefully designed synchronizer.

- The inter-router FIFOs have to interface clock domains having the same frequency, but different phases. Several solutions have been proposed [8,11,12,13,14]. It is always possible to increase the metastability robustness by increasing the latency, and any solution will be a tradeoff between latency and MTBF.
- The FIFOs interfacing the router and the NIC connect two clock domains with unknown frequencies. The cost to pay to guaranty a high MTBF is an increased latency (compared with the inter-router FIFOs). This is acceptable, as the number of those high latency FIFOs is limited.

Our bi-synchronous FIFO [24] has the ability of constructing the whole system in a modular manner. Each cluster of the system can be designed and tested independently and finally assembled without verifying the time constraints between clusters. Its latency is 1-2 clock cycles on a mesochronous environment and 2-3 clock cycles on a fully asynchronous environment. The depth of the BE and GS bi-synchronous FIFOs are 8 and 4 flits respectively. The choice of the depth is a compromise between packet size, network throughput and FIFO area.

4. EXPERIMENTAL RESULTS

Both synthesizable VHDL models and cycle accurate SystemC models of all DSPIN components have been designed.

We simulated a multi-clusters mesh containing 10x10 clusters. Each cluster contains one BE initiator, one BE target, one GS target, and one optional GS initiator. The average packet latency is measured as the average number of cycles for a round trip from an initiator to a target, and back to the same initiator. For each initiator, the offered load is the ratio between the number of injected flits and the total number of cycles. The BE traffic has a uniform random distribution (each BE initiator randomly send packets to all BE targets). The packet length is a random value between 1 and 16 flits. If we plot the average latency versus the BE offered load (Figure 6), we see a saturation threshold of 25% for the BE traffic (part of the offered load is not accepted by the network), but the GS traffic is not impacted by the BE traffic.

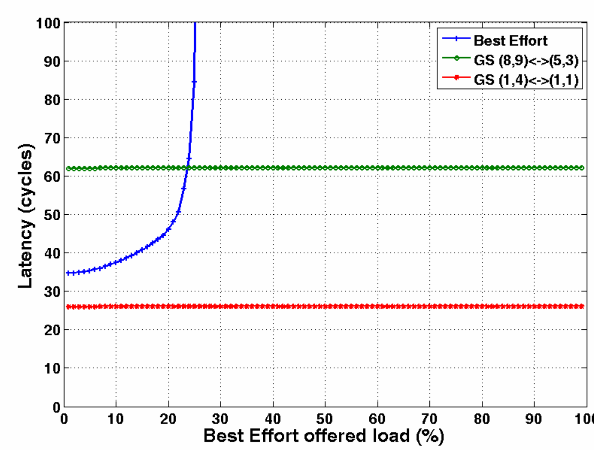

Figure 6. BE and GS latency in function of BE offered load

The latency and throughput of the GS traffic have been analyzed. For example, the latency of the network for the roundtrip between cluster (8,9) and cluster (5,3) is deterministic and equal to 62 cycles. Moreover, the throughput of each GS channel is guaranteed up to 50%, due to the round-robin allocation of the TDM slots. On a 500MHz implementation, each GS channel has a bandwidth of 8 Gbps.

The performances of the BE packets are determined principally by three factors: the packet length, the number of routers between the senders and the receivers and the network load. The longer the packet, the lower the performance, due to the limited depth of the FIFOs. The aggregated BE bandwidth for the 100 clusters is 400 Gbps.

The area evaluation of the DSPIN network has been done for a 90 nm CMOS process. As all DSPIN components are synthesizable, we have computed the silicon area for the FIFOs and the routers using the ST CMOS 90nm standard cell library. Table I shows the Synopsys area after synthesis of one router and the associated FIFOS: 5 BE FIFOs and 5 GS FIFOs. The depth of the BE and GS FIFOs are 8 and 4 flits respectively. The flit size is 34 bits.

TABLE I. AREA COST PER ROUTER

Block	Area
5 BE FIFOs	0.036 mm^2
5 GS FIFOs	0.018 mm^2
Router (without FIFOs)	0.028 mm^2
Total	**0.082 mm^2**

5. CONCLUSIONS

The experience gained in the physical implementation of the 32 ports SPIN network [19] was precious to define a new architecture, well suited to the Globally Asynchronous, Locally Asynchronous (GALS) paradigm. The mesh topology and the deterministic X-first algorithm give a very low cost, synthesizable and distributed router implementation. The mesh topology, associated with the distributed implementation of the router itself solves the problem of long wires. A simple bi-synchronous FIFO and a mesochronous clock distribution solve the problem of asynchronous communication between sub-systems. We demonstrated that the virtual channel approach (generally used to multiplex several logical channels on the physical link between router), can be applied to the router itself, making TDM multiplexing possible in a GALS, clustered multiprocessor architecture. With this low cost method, the DSPIN architecture provides the system designer hard bounds for both the latency (upper bound) and the throughput (lower bound) of a limited number of point-to-point communications.

The silicon area after synthesis of the router and their FIFOs is about 0.082 mm^2 per router, in a 90 nm CMOS process, which is significantly smaller than other published micro-networks providing the same type of services.

REFERENCES

[1] P. Guerrier and A. Greiner. A generic architecture for on-chip packet-switched interconnections, Proc. Design Automation and Test in Europe (DATE'00), pp. 250-256, Mars 2000.

[2] D. M. Chapiro. Globally-Asynchronous Locally-Synchronous systems. PhD thesis, Stanford University, 1984.

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

[4] VSI Alliance. Virtual Component Interface Standard (OCB2 2.0), August 2000. http://www.vsia.com/

[5] E. Rijpkema, K. Goossens, A. Radulescu, J. Dielssen, J. van Meerbergen, P. Wielage and E. Waterlanden. Trade-offs in the design of a router with both guaranteed and best-effort services for networks on chip, IEE. Proc.-Comput. Digit. Tech., Vol. 150, No 5, September 2003.

[6] T. Bjerregaard and J. Sparsø. A router architecture for connection-oriented service guarantees in the MANGO clockless Network-on-Chip, IEEE Proc. Design Automation and Test in Europe (DATE'05), March 2005.

[7] W. J. Dally and C. L. Seitz. Deadlock free message routing in multiprocessor interconnection networks, IEEE Transactions on Computers. C-36, 5, pp. 547-553, May 1987

[8] J. Jex, C. Dike and K. Self. Fully asynchronous interface with programmable metastability settling time synchronizer, Patent 5,598,113, January 1997.

[9] C. Dike and E. Burton. Miller and noise effects in a synchronizing flip-flop, IEEE Journal of Solid-State Circuits, vol 34, pp. 849-855, 1999.

[10] Ran Ginosar. Fourteen ways to fool your synchronizer, Proc. IEEE 9th Int. Symp. on Asynchronous Circuits and Systems (ASYNC'03), 2003.

[11] T. Chelcea and S. M. Nowick. Robust interfaces for mixed-timing systems, IEEE Trans. on Very Large Scale Integration (VLSI) Systems, vol. 12, no. 8, pp 857-873, August 2004.

[12] A. Chakraborty and M.R. Greenstreet. Efficient self-timed interfaces for crossing clock domains, Proc. 9th IEEE Int. Symp. Asynchronous Circuits and Systems (ASYNC'03), May 2003.

[13] Y. Elboim, A. Kolodny and R. Ginosar. A clock tuning circuit for system-on-chip, IEEE Trans. on Very Large Scale Integration (VLSI) Systems, v.11 n.4, p.616-626, August 2003.

[14] Joep Kessels. Register-communication between mutually asynchronous domains, Proc. 11th IEEE Int. Symp. Asynchronous Circuits and Systems (ASYNC'05), 2005.

[15] W. J. Dally and B. Towles. Route packets, not wires: on-chip interconnection networks, Design Automation Conference (DAC 2001), June 2001.

[16] L.M. Ni and P.K. McKinley. A survey of wormhole routing techniques in direct networks, IEEE Computer 2 (1993) 62-75.

[17] E. Bolotin, I. Cidon, R. Ginosar and A. Kolodny. QNoC: QoS architecture and design process for network on chip, Journal of Systems Architecture, 50(2-3), pp. 105-128, February 2004.

[18] D. Peiliang, Y. Rilong, X. Hongbo and Y. Chengfang. Multi-clock driven system: a novel VLSI architecture, Proc. 4th Int. Conf. ASIC, pp. 555-558, 2001.

[19] A. Andriahantenaina and A. Greiner. Micro-network for SoC: Implementation of a 32-port SPIN network, Design Automation and Test in Europe (DATE 2003) pp. 1128-1129, March 2003

[20] W. Qing, M. Pedram and X. Wu. Clock-gating and its application to low power design of sequential circuits, IEEE Trans. Circuits Syst. I, Fundam. Theory Applicat., vol. 47, no3, pp.414-420, Mars 2000.

[21] K. Goossens, J. van Meerbergen, A. Peeters and P. Wielage. Networks on Silicon: Combining Best-Effort and Guaranteed Services, Design Automation and Test in Europe (DATE'02), 2002.

[22] M. Millberg, E. Nilsson, R. Thid and A. Jantsch. Guaranteed bandwidth using looped containers in temporally disjoint networks within the Nostrum network on chip, IEEE Proc. Design Automation and Test in Europe (DATE'04), vol. 2, pp. 890 - 895, Febrary 2004.

[23] D. Bertozzi and L. Benini. Xpipes: A Network-on-Chip architecture for gigascale Systems-on-Chip, IEEE Circuits and Systems Magazine, Q2 2004.

[24] I. Miro Panades. Buffer memory control device (Dispositif de commnade d'une memoire tampon). Patent pending.