Survey of NoC and Programming Models Proposals for MPSoC

Eduard Fernandez-Alonso[1], David Castells-Rufas[1] , Jaume Joven[2] and Jordi Carrabina[1]

**[1] CAIAC, Universitat Autònoma de Barcelona
Edifici Enginyeria, Campus UAB, 08193, Bellaterra, Spain**

**[2] LSI, École Polytechnique Fedérale de Lausanne
Lausanne, Switzerland**

Abstract

The aim of this paper is to give briefing of the concept of network-on-chip and programming model topics on multiprocessors system-on-chip world, an attractive and relatively new field for academia. Numerous proposals from academia and industry are selected to highlight the evolution of the implementation approaches both on NoC proposals and on programming models proposals.

Keywords: *Survey, Network-on-Chip, Parallel Programming, Multiprocessor*

1. Introduction

Future embedded System-on-Chip (SoC) will probably be made up of tens or hundreds of heterogeneous Intellectual Properties (IP) cores, which will execute one parallel application or even several applications running in parallel. These systems could be possible due to the constant evolution in technology that follows the Moore's law, which will lead us to integrate more transistors on a single die, or the same number of transistors in a smaller die.

For such SoCs, Network-on-Chip (NoC) architectures are the solution for the scalability problem. Traditional on-bus communication-based solutions, Figure 1a, pose serious problems related to the integration of several IP cores. When the number of participating components connected is between four and ten, the bus system will produce a performance bottleneck problem [1].

An alternative for on-bus communication solution is to substitute the bus connection by a fully crossbar system, Figure 1b, but as the number of participating components rises, the complexity of the wires could be dominant over the logical parts.

Finally, NoC-based interconnection system was presented as the solution to these problems. The NoC [1][2][3][4][5] entails a unified solution to On-Chip communication and the possibility to do scalable systems at supportable levels of power consumption. In embedded MultiProcessor System-on-Chip (MPSoC) systems, NoCs can provide a flexible communication infrastructure, in which several components such as microprocessor cores, MCU, DSP, GPU, memories, and other IP components can be interconnected.

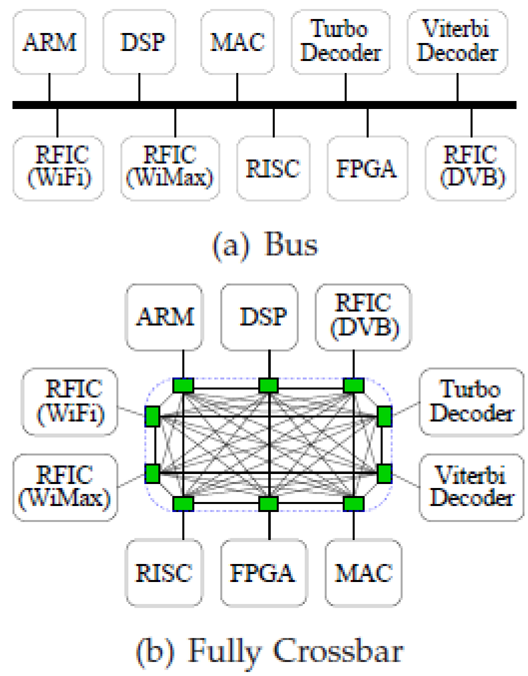

Figure 1 Interconnection systems a) Bus-based system approach b) Crossbar-based system approach.

NoCs have been extensively discussed in several regular publications and special issues, from journals to conferences and workshops, and also, the NoC topic has inspired symposiums like ACM/IEEE International Symposium on Network-on-Chip. Figure 2 shows the evolution of the interest the NoC topic has raised in the last decade in terms of hits when "network-on-chip" is searched in the IEEE Xplore digital library [6]. Figure 2 also shows the evolution of another two topics combined with NoCs: publications about NoC-based MPSoC systems have been increasing their popularity year by year, whereas the other topic, parallel programming model for such systems, seems not to be increasing its popularity among the academia, despite the interest shown by the industry in such topic.

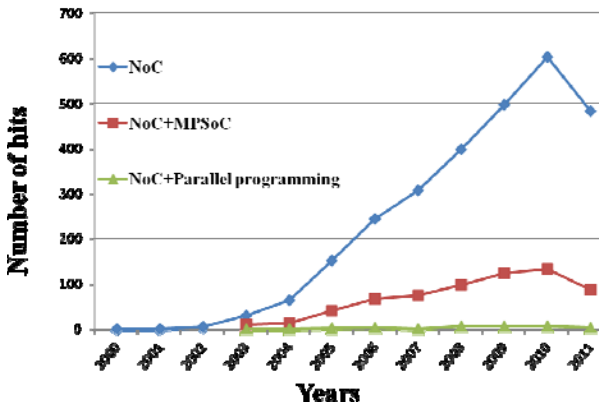

Figure 2 IEEE Xplorer hits for different "network-on-chip" searches.

In this paper, we present an extended survey of NoC proposals and parallel programming models for MPSoC systems and examples of implementations from academia and industry. We analyzed over 100 relevant papers about NoC systems and parallel programming models proposals for MPSoCs.

2. Hardware Proposals

A large summary of NoC proposals is shown in Table 1 in subsection 2.4, where switching scheme and topology are given for each NoC and explained in the following subsections. Although Table 1 is very extensive, it is not possible to be all-inclusive. However, it is quite representative of NoC designs.

2.1 NoC Components

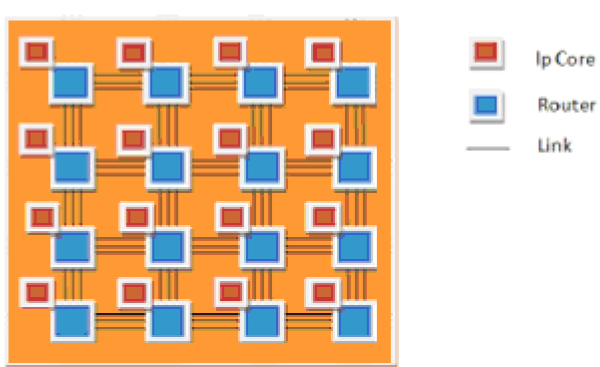

Figure 3 NoC representation.

Basically, there are just three main components on a NoC-based system:

- Network Interface Controller (NIC). The NIC implements the interface between each IP node and the communication infrastructure. The architecture of the NIC component can be divided into two modules. The first one is focused on the interaction with the computation or memory node bus, and the other one is focused on the interaction with the rest of the NoC. This component is called in many different ways in NoC literature: NI for Network Interface, NA for Network Adapter, or RNI for Resource Network Interface are some examples.

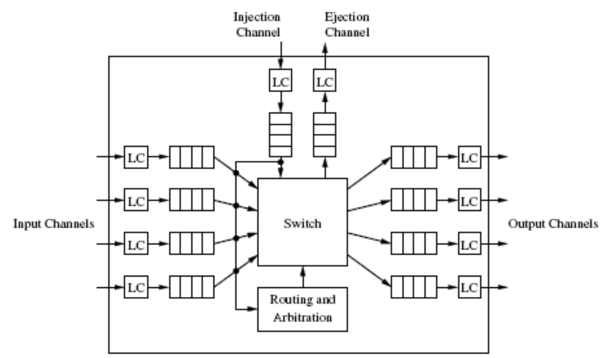

Figure 4 Generic Router block diagram from DUATO[107] (LC = Link Controller)

- Router. Also called switch. These components are in charge of forwarding data to the next tail. On the routers we can find implemented the routing protocol, buffer capabilities and the switching method. In general, the router component is composed of the next elements: Arbitrer, which its main task is to grant channels (selecting an input port and an output port) and route packets; Crossbar, of n input x n output ports that direct the input packet to the corresponding output port; and buffer or queue, if that is the case – as it is in the packet switching protocols – which is used to buffer incoming and outgoing data in the router.
- Links. Links are the specific connections that provide communication between components.

2.2 Switching Method

Traditional multiprocessor networks techniques have been adapted to on-NoC-based multiprocessor systems. There are basically two switching methods: Circuit-Switching and Packet-Switching.

In Circuit-Switching, a path from source to destination is reserved before the information is emitted through the NoC components. All data are sent following the reserved path, which is released after the transfer has been completed.

In Packet-Switching, there is not a reserved path from source to destination; instead, data are forwarded hop by hop using the information contained in the packet. Thus, in each router the packets are buffered before being forwarded to the next router.

In Packet-Switching we can distinguish three choices according to how packets are stored and forwarded to routers: Store-and-Forward, Virtual Cut-Through and Wormhole.

- In Store-and-Forward protocol, the packet is stored completely before forwarded to the next hop. Thus, if the router in the path does not have sufficient buffer space, the packet is stalled. This method requires buffering capacity for at least one full packet.
- In Virtual Cut-Through protocol, the packet is forwarded to the next hop once it is guaranteed that the full packet can be stored. The main difference with Store-and-Forward is that there is no need to wait for the storage of the complete packet before forwarding it to the next hop. However, this method also requires buffering capacity for at least one full packet.
- In Wormhole protocol, each packet is further divided into small units called flits. There are three different types of flits; header, body and tail. The header flit reserves a path between hops and establishes a channel where one or many body flits – which contain the packet information – follow, and finally the tail flit releases the reserved path. The major advantage of Wormhole method is that there is no need of buffering capacity for a single packet.

2.3 Topology

Since NoC interconnection systems have replaced traditional bus interconnection systems, many topologies have been proposed, most of them adapted to the constraints of the embedded world from parallel multiprocessors systems. Topologies can be classified following geometric criteria as:

- Regular topologies. Mesh-like, fat-tree, ring, torus or star are examples of regular topologies. Figure 5
- Irregular topologies. Custom designs or application oriented designs as Figure 5 shows are examples of irregular topologies.

But topologies can also be divided into networks where all nodes are attached to a core and networks where they are not:

- Direct topologies. In direct topologies all nodes are attached to a computational or memory core. Mesh, torus or rings are examples of direct topologies. Figure 5
- Indirect topologies. In indirect topologies not all nodes are attached to a core. Trees or star topologies are examples of indirect networks. Figure 5

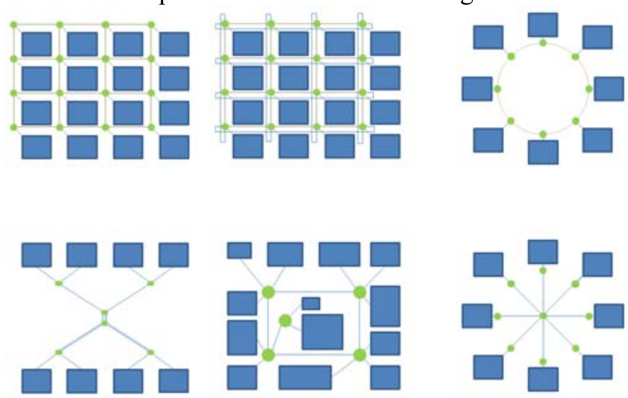

Figure 5 Some basic network topologies. a) Mesh (up-left), b) Torus (up-middle), c) Ring (up-right), d) Fat-tree (down-left), e) Custom (down-middle), and f) Star (down-right). All links are bidirectional.

2.4 List of Proposals

Many researches have been done in MPSoC field and in NoC interconnection system.

In this section, we give a brief clarifying vision of the State-of-the-Art of NoC proposals.

Table 1: NoC proposals summary (C: Circuit; P: Packet; WP: Wormhole Packet; SF: Store and Forward Packet; VT: Virtual Cut-Through Packet)

	NAME	YEAR	SWITCHING	TOPOLOGIES	IMPLEMENTATION
1	SPIN	2000	WP	Fat-tree	Synthesis, Simulator
2	MESCAL	2001	P	Custom	Simulator
3	MicroNet	2001	P	Custom	Simulator
4	Carloni	2002	P	Point-to-point	No (analytical)
5	Chain	2002	P	Chain	ASIC
6	CLICHE	2002	P	Mesh	Simulator
7	Eclipse	2002	P	Mesh	Simulator

8	J. HU et al.	2002	C	Point-to-Point	Simulator
9	Octagon	2002	P	Ring	Simulator
10	RAW	2002	WP	Mesh	ASIC
11	Aethereal	2003	WP	Custom	Synthesis
12	Pande P.P. et al.	2003	WP	Butterfly Fat-Tree	Synthesis, Simulator
13	aSoC	2004	C	Mesh	Synthesis, Simulator
14	Blackbus	2004	SF	Mesh	Simulator
15	DyAD	2004	WP	Mesh	Synthesis, Simulator
16	H.C. Chi et al.	2004	SF	Torus	ASIC
17	Hermes	2004	WP	Mesh	FPGA, Simulator
18	Microspider	2004	WP	Ring	Synthesis
19	Mondinelli	2004	SF	Fat-Tree	ASIC
20	Nexus	2004	C	Crossbar	ASIC
21	Nostrum	2004	SF	Mesh	Simulator
22	Proteo	2004	P	Custom Point-to-Point	Synthesis, Simulator
23	Qnoc	2004	WP	Mesh, Fat-Tree, Torus	Synthesis
24	RaSoc	2004	WP	Mesh	Synthesis
25	Ring road	2004	SF	Ring	Simulator
26	SNA	2004	P	Crossbar, Star-Mesh	Simulator
27	T .Felicijian et al.	2004	WP	Mesh	Simulator
28	ANOC	2005	WP	Mesh	Synthesis
29	Arteris	2005	WP	Custom	Simulator, ASIC
30	ASPIDA	2005	P	Chain	Synthesis, Simulator
31	CDMA NoC	2005	P	Star-Star, Star-Mesh	Synthesis
32	HiNoC	2005	C	Mesh	No (analytical)
33	IMEC NoC	2005	VT	Adaptable topology	Synthesis
34	J. Kim et al.	2005	WP	Mesh, Torus	Simulator
35	J. Xu et al.	2005	P	Crossbar	Synthesis, Simulator
36	Mango	2005	WP	Mesh	Synthesis
37	Pnoc	2005	C	Custom	FPGA
38	P. Wolkote et al.	2005	C	Mesh	Synthesis
39	SDM NoC	2005	C	Mesh	Synthesis, Simulator
40	SoCBus	2005	C	Mesh	ASIC
41	Xpipes	2005	WP	Custom	Synthesis, Simulator
42	A. Bouhraoua et al.	2006	WP	Fat-Tree	Synthesis, Simulator
43	ASNoC	2006	P	Custom	Simulator
44	B. Ahmad et al.	2006	WP	Torus	Simulator
45	Crossroad	2006	P	Irregular	Synthesis, Simulator
46	DSPIN	2006	C	Mesh	Synthesis, Simulator
47	dTDMA	2006	C	Bus	Simulator
48	H.G. Lee et al.	2006	P	Custom Point-to-Point	FPGA
49	HIBI	2006	C/P	Hierarchical Bus	FPGA
50	INoC	2006	P	Irregular	Simulator
51	K. Lee et al.	2006	P	Hierarchical Star	Synthesis
52	Lochside	2006	WP	Mesh	ASIC

53	N. Kavaldjev et al.	2006	WP	Mesh	Synthesis, Simulator
54	ProtoNoC	2006	C	Mesh	Simulator
55	R. Marculescu et al.	2006	WP	Mesh	Synthesis, Simulator
56	Slim-spider	2006	VT	Star	Synthesis
57	XGFT	2006	WP	Fat-Tree	Synthesis, Simulator
58	Ambric MPPA NoC	2007	P	Mesh	ASIC
59	B. Feero et al.	2007	WP	3D-Mesh	Synthesis, Simulator
60	IBM Cell EIB	2007	P	Ring-Star	ASIC
61	Intel TeraFLOPS	2007	WP	Mesh	ASIC
62	M. Hosseinabady et al.	2007	WP	Custom	Synthesis, Simulator
63	NoCMaker	2007	C/P	Mesh	FPGA
64	Pavlidis	2007	WP	3D-Mesh	Synthesis
65	Polaris	2007	WP	Mesh, 3D-Mesh, Torus, Ring, Fat-tree	Synthesis
66	SCC NoC	2007	WP	3-Ary 2-Cube	ASIC
67	STNoC	2007	P	Custom	ASIC
68	TRIPS	2007	WP	Mesh	ASIC
69	Faust	2008	WP	Mesh	ASIC
70	EVC-NoC	2008	P	Mesh	Simulator
71	MoCSYS	2008	C/P	Mesh	FPGA
72	Tile64	2008	WP	Mesh	ASIC
73	ALPIN NoC	2009	P	Mesh	FPGA
74	PMCNOC	2009	P	Mesh	Synthesis, Simulator
75	XHiNoC	2009	WP	Mesh	FPGA
76	RampSoC	2010	C/P	Point-to-Point, Mesh, Star-Wheels	FPGA
77	Qsys	2011	P	Custom	FPGA

2.5 Conclusions of Proposals

Table 1 represents several NoC prototypes developed in industries and in academic literature. The table presents some existing NoC prototypes that have been published with different network topologies and switching methods along the last decade. As it can be seen, the most commonly used topology is mesh-like (Figure 5). Intel-TERAFLOPS [70] system is an example of industry NoC-based MPSoC that uses a mesh topology, and it is composed of 80 homogeneous computing elements interconnected by an 8x10 2D-Mesh network topology. Custom topologies (Figure 5) are also frequent in literature. M. Hosseinabady et al. [71] NoC is an example of a custom or application-oriented topology. M. Hosseinabady et al. NoC topology is created from the Generalized de Bruijn Graph of the application. Another representative topology group is tree-like topologies. In this group, the most important ones are Fat-tree topologies (Figure 5). SPIN [9] or Polaris [74] systems are examples of NoC-based systems using Fat-tree topology. From the switching method point of view, circuit switching is less used and preferable than packet switching methods, being Wormhole switching the most addressed protocol in literature and industry.

Simulation and synthesis are the most popular evaluation and implementation methods, despite the fact that simulation can suffer from simplifications and inaccuracies, as it has been pointed in [106]. On the other hand, real prototyping can solve these problems, but making measurements becomes harder with higher costs. It is also relevant the presence of FPGA implementations. Recent FPGA devices already offer more than 1 million Logic Elements and more than 40 Megabits of embedded memory, which is large enough for MPSoC implementations, and it will become a cheaper way to implement MPSoC systems than ASIC option.

IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 2, No 3, March 2012
ISSN (Online): 1694-0814
www.IJCSI.org

3. Programming Models Proposals

As it has been remarked previously, the main reason to adopt the NoC paradigm is the will to answer the question “how to solve the scalability problem?”.

However, there are many other variables to take into account, particularly adopting NoC architecture. Having a scalable communication system is not enough to achieve fully scalability, since it is mandatory for the programmer to have enough tools to design applications that will run efficiently on MPSoC systems.

The programming model is the necessary way that permits programmers to abstract the logic of applications and translate it to the hardware platform system. Programming models are the bridge that must save the gap between hardware and software trying to raise productivity and efficiency. Therefore, if the programming model is developed from a hardware point of view (that is, bottom-up) then programming the system could be a tough task, decreasing productivity. If the programming model is developed the other way around with a top-down approach, then the efficiency could be affected due to the difficulties that could appear when mapping the application in the hardware system. An additional goal that must be included in the many-core embedded systems is scalability. A programming model must provide scalability – that is, the increase of performance of the system while increasing the hardware resources assigned to the system.

3.1 Traditional Parallel Programming Models

Typically, there are two traditional parallel programming models:

- Shared-memory model: when communication occurs implicitly through a global address space accessible for all processors. This model implies to ensure data coherence and synchronization. Systems based on this model usually have a shared memory architecture, which can suffer a performance bottleneck due to the memory hierarchy.
- Message passing: when communication occurs between a sender and a receiver. Message passing model implies a set of processors with no shared address and also implies collaboration between sender and receiver. The most common primitives used for communication in this model are *send* and *receive*, and always a *send* operation must match a *receive* operation. This model could overcome the non-determinism and the scalability limits that cache coherence protocols introduce in shared memory architectures. The main drawbacks of message passing model are that the programmer must explicitly implement the parallelism and data distribution dealing with data dependencies and inter-process communication and synchronization.

Other parallel programming models are:

- Data parallel model: when data partitioning determines parallelism and several processors perform the same operation concurrently.
- Thread-based model: when a process has multiple threads running concurrently.

3.2 Existing Programming Models over NoC-Based Systems

A large number of MPSoC-specific programming models have been defined in the last years based on shared memory or message passing models. Examples of programming models will include OpenMP[92] for shared memory architectures, or MPI[97] for message passing. Below, we will describe the implementation of some of these existing parallel programming models.

3.2.1 Shared-Memory Programming Models over NoC-Based Systems

OpenMP: for Open Multi-Processing. OpenMP is an API for shared-memory multiprocessing in C, C++ and Fortran. In OpenMP all threads can access the shared data, but private data can be accessed only by the thread that owns such data. OpenMP expresses parallelism using a set of compiler directives called #pragma. OpenMP is supported on Cell processor, for example.

CUDA [93]: for Compute Unified Device Architecture. CUDA, provided by Nvidia, is an example of programming model used in industry. CUDA is a software platform for parallel computing in C, C++ and Fortran on Nvidia GPUs (graphic processing unit). CUDA requires the programmer to write special code for performing parallel processing.

OpenCL[94]: for Open Computing Language. OpenCL is an open standard for parallel applications over multi-core platforms with CPUs and GPUs. OpenCL is developed by Khronos group [95], which is formed by several industrial partners as IBM, ARM, AMD, or Intel. OpenCL is based on the model of one host plus one or more computing devices, which are a collection of one or more CPUs or GPUs. In OpenCL execution model, the code for an OpenCL device is written in C and it is called kernel, and a collection of kernels and other functions is a program. OpenCL provides APIs for writing kernel in C, and APIs are used to define and control the platforms, which are the hardware abstraction of diverse computational resources. In OpenCL the memory management is explicit, and it is responsibility of the programmer to move data from host to the computer device global and local memories and back.

3.2.2 MP Programming Models over NoC-Based Systems

MCAPI [96]: for Multicore Communications API. MCAPI is a research work from Multicore Association that defines a set of lightweight multicore communication API for closely distributed embedded systems (multiple cores on a chip). MCAPI provides three modes of communication: messages, connected-channels packets, and connected-channels scalars. MCAPI is independent from language, processor, and operating system.
MPI: for Message Passing Interface. MPI has been recently adopted as a standard. It basically specifies a set of point-to-point and collective communication primitives and creation and management of process primitives. MPI is language-independent, and recently a large number of traditional message passing interface programming models are being proposed for MPSoCs. Table 2 shows different MPI-like approaches for MPSoC systems compared to two of the most known implementations.
The open source OpenMPI library supports over 300 MPI standard commands, just the same as the also open source available MPICH library does. However, both implementations are out of range of the memory size available on embedded MPSoCs, since OpenMPI has a huge code size of 40MB for all layers size, and MPICH code size is over 47MB for all layers.

Table 2: MPI over NoC implementations

	OpenMPI [87][88]	**MPICH [89]**	**TMD-MPI[90]**
Availability	Open Source	Open Source	Proprietary
MPI library size	25MB	7MB	9Kb
All layers size	40Mb	47Mb	--
MPI commands supported	300	300	11
	SoC-MPI0[91]	**RAMPSoC-MPI [85]**	**ocMPI [7]**
Availability	Proprietary	Proprietary	Open Source
MPI library size	11-16 Kb	37 Kb	11Kb
All layers size	--	43Kb	14Kb
MPI commands supported	jun-18	18	11

TMD-MPI implementation was designed for embedded systems, and the library has a footprint of just 9KB. It is a proprietary implementation that supports 11 MPI commands limited to a particular architecture, which makes TMD-MPI very fast but not portable to other architectures. SoC-MPI is another lightweight library that implements from 6 to 18 MPI commands. It was specially designed for embedded systems as a proprietary library and requires from 11KB to 16KB.
RAMPSoC SoC shows a runtime adaptive MPSoC with a proprietary subset of standard MPI library whose footprint is 43KB for all layers.
Finally, ocMPI was also designed for embedded MPSoC systems as an open source project. This implementation is a very lightweight library that supports up to 11 standard MPI commands.

3.3 Other Parallel Programming Models Implementations

- TBB[98][99]: for Intel Threading Building Blocks, it is a commercially supported open-source C++ template library for shared-memory programming model.
- X10[100]: X10 is a class-based object-oriented programming language from IBM.
- Pthreads: for POSIX Threads, it is a POSIX standard for threads.
- StreamIT[101]: from MIT, it is a programming language specifically designed for streaming systems.
- Cilk/Cilk++[102][103] is a C-based runtime system for shared-memory parallel programming developed by MIT.
- Chapel[104]: from Cray, it is a parallel programming language developed as an open source with contributions from academia and industry.
- Axum[105]: from Microsoft, it is a programming model based on .Net.

4. Conclusions

Network-on-chip topic remains as an attractive research field for academia that raises its popularity year by year. Related topics such as multiprocessor systems-on-chip based on network-on-chip architectures have boosted in academic and industry works in the last years.
This paper shows an overview of the evolution of the state-of-the-art on NoC-based MPSoCs and parallel programming models for multiprocessor systems-on-chip.

Acknowledgments

This work was partly supported by the European cooperative CATRENE project CA104 COBRA, the Spanish Ministerio de Industria, Turismo y Comercio project TSI-020400-2010-82 and the Catalan Government Grant Agency Ref. 2009SGR700.

The authors want to acknowledge Joan Carles Chak Ma for sharing his knowledge of shared-memory languages.

References

[1] A. JANTSCH and H. TENHUNEN. Networks on Chip. Kluwer Academic Publishers, 2003.

[2] Benini L. et al. Networks on chips: a new SoC paradigm. IEEE Computer, 35(1). 2002, pp. 70-78.

[3] L. Benini and G. De Micheli, "Powering networks on chips," in Proc.ISSS, 2001.

[4] W. J. Dally and B. Towles, "Route packets, not wires: on-chip interconnection networks," in Proc. Design Automation Conf., 2001.

[5] S. Kumar, A. Jantsch, J.-P. Soininen, M. Forsell, M. Millberg, J. Öberg, J. Tiensyrjä, and A. Hemani, "A network on chip architecture and design methodology," in Proc. ISVLSI, 2002.

[6] http://ieeexplore.ieee.org/Xplore/guesthome.jsp

[7] J. Joven, "A lightweight MPI-based programming model and its HW support for NoC-based MPSoCs", PhD Forum DATE, IEEE/ACM Design, Automation and Test in Europe (DATE'09), April 2009, Nice, France.

[8] D. Castells-Rufas, J. Joven, S. Risueño, E. Fernandez, J. Carrabina, T. William, H. Mix. "MPSoC Performance Analysis with Virtual Prototyping Platforms". PSTI, San Diego, USA, September, 2010

[9] Guerrier, P.; Greiner, A.; , "A generic architecture for on-chip packet-switched interconnections ," Design, Automation and Test in Europe Conference and Exhibition 2000. Proceedings , vol., no., pp.250-256, 2000

[10] Adrijean Andriahantenaina, Alain Greiner, "Micro-Network for SoC: Implementation of a 32-Port SPIN network," date, vol. 1, pp.11128, Design, Automation and Test in Europe Conference and Exhibition (DATE'03), 2003

[11] Sgroi, M.; Sheets, M.; Mihal, A.; Keutzer, K.; Malik, S.; Rabaey, J.; Sangiovanni-Vincentelli, A.; , "Addressing the system-on-a-chip interconnect woes through communication-based design," Design Automation Conference, 2001. Proceedings , vol., no., pp. 667- 672, 2001

[12] Wingard, D.; , "MicroNetwork-based integration for SOCs," Design Automation Conference, 2001. Proceedings , vol., no., pp. 673- 677, 2001

[13] Carloni, L.P.; Sangiovanni-Vincentelli, A.L.; , "Coping with latency in SOC design," Micro, IEEE , vol.22, no.5, pp. 24- 35, Sep/Oct 2002

[14] Bainbridge, J.; Furber, S.; , "Chain: a delay-insensitive chip area interconnect," Micro, IEEE , vol.22, no.5, pp. 16- 23, Sep/Oct 2002

[15] Kumar, S.; Jantsch, A.; Soininen, J.-P.; Forsell, M.; Millberg, M.; Oberg, J.; Tiensyrja, K.; Hemani, A.; , "A network on chip architecture and design methodology," VLSI, 2002. Proceedings. IEEE Computer Society Annual Symposium on , vol., no., pp.105-112, 2002

[16] Forsell, M.; , "A scalable high-performance computing solution for networks on chips," Micro, IEEE , vol.22, no.5, pp. 46- 55, Sep/Oct 2002

[17] Jingcao Hu; Yangdong Deng; Marculescu, R.; , "System-level point-to-point communication synthesis using floorplanning information [SoC]," Design Automation Conference, 2002. Proceedings of ASP-DAC 2002. 7th Asia and South Pacific and the 15th International Conference on VLSI Design. Proceedings. , vol., no., pp.573-579, 2002

[18] Karim, F.; Nguyen, A.; Dey, S.; , "An interconnect architecture for networking systems on chips," Micro, IEEE , vol.22, no.5, pp. 36- 45, Sep/Oct 2002

[19] Taylor, M.B.; Kim, J.; Miller, J.; Wentzlaff, D.; Ghodrat, F.; Greenwald, B.; Hoffman, H.; Johnson, P.; Jae-Wook Lee; Lee, W.; Ma, A.; Saraf, A.; Seneski, M.; Shnidman, N.; Strumpen, V.; Frank, M.; Amarasinghe, S.; Agarwal, A.; , "The Raw microprocessor: a computational fabric for software circuits and general-purpose programs," Micro, IEEE , vol.22, no.2, pp. 25- 35, Mar/Apr 2002

[20] Rijpkema, E.; Goossens, K.G.W.; Radulescu, A.; Dielissen, J.; van Meerbergen, J.; Wielage, P.; Waterlander, E.; , "Trade offs in the design of a router with both guaranteed and best-effort services for networks on chip," Design, Automation and Test in Europe Conference and Exhibition, 2003 , vol., no., pp. 350- 355, 2003

[21] Pande, P.P.; Grecu, C.; Ivanov, A.; Saleh, R.; , "High-throughput switch-based interconnect for future SoCs," System-on-Chip for Real-Time Applications, 2003. Proceedings. The 3rd IEEE International Workshop on , vol., no., pp. 304- 310, 30 June-2 July 2003

[22] Jian Liang; Laffely, A.; Srinivasan, S.; Tessier, R.; , "An architecture and compiler for scalable on-chip communication," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on , vol.12, no.7, pp. 711- 726, July 2004

[23] Kenichiro Anjo, Yutaka Yamada, Michihiro Koibuchi, Akiya Jouraku, Hideharu Amano, "BLACK-BUS: A New Data-Transfer Technique Using Local Address on Networks-on-Chips," ipdps, vol. 1, pp.10a, 18th International Parallel and Distributed Processing Symposium (IPDPS'04) - Papers, 2004

[24] Jingcao Hu; Marculescu, R.; , "DyAD - smart routing for networks-on-chip," Design Automation Conference, 2004. Proceedings. 41st , vol., no., pp. 260- 263, 2004

[25] Hsin-Chou Chi; Jia-Hung Chen; , "Design and implementation of a routing switch for on-chip interconnection networks," Advanced System Integrated Circuits 2004. Proceedings of 2004 IEEE Asia-Pacific Conference on , vol., no., pp. 392- 395, 4-5 Aug. 2004

[26] F. MORAES, N. CALAZANS, A. MELLO, L. M¨O LLER, and L. OST. "Hermes: An Infrastructure for Low Area Overhead Packet Switching Networks on Chip". Elsevier, Integration, The VLSI Journal, 38(1):69–93, Oct. 2004.

[27] S. Evain, J. P. Diguet, and D. Houzet, "µspider: a CAD tool for efficient NoC design," in Norchip, Nov. 2004, pp. 218–221.

[28] F. Mondinelli, M. Borgatti, and Z. Vajna, "A 0.13 um 1Gb/s/channel store-and-forward network on-chip," in SOCC, Sep. 2004, pp. 141–142.

[29] Lines, A.; , "Asynchronous interconnect for synchronous SoC design," Micro, IEEE , vol.24, no.1, pp. 32- 41, Jan.-Feb. 2004

[30] Millberg, M.; Nilsson, E.; Thid, R.; Jantsch, A.; , "Guaranteed bandwidth using looped containers in temporally disjoint networks within the nostrum network on chip," Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings , vol.2, no., pp. 890- 895 Vol.2, 16-20 Feb. 2004

[31] Pavlidis, V.F.; Friedman, E.G.; , "3-D Topologies for Networks-on-Chip," Very Large Scale Integration (VLSI)

Systems, IEEE Transactions on , vol.15, no.10, pp.1081-1090, Oct. 2007

[32] Evgeny Bolotin, Israel Cidon, Ran Ginosar, Avinoam Kolodny, QNoC: QoS architecture and design process for network on chip, Journal of Systems Architecture, Volume 50, Issues 2-3, Special issue on networks on chip, February 2004, Pages 105-128

[33] Zeferino, C.A.; Kreutz, M.E.; Susin, A.A.; , "RASoC: a router soft-core for networks-on-chip," Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings , vol.3, no., pp. 198- 203 Vol.3, 16-20 Feb. 2004

[34] Samuelsson, H.; Kumar, S.; , "Ring road NoC architecture," Norchip Conference, 2004. Proceedings , vol., no., pp. 16- 19, 8-9 Nov. 2004

[35] Sanghun Lee; Chanho Lee; Hyuk-Jae Lee; , "A new multi-channel on-chip-bus architecture for system-on-chips," SOC Conference, 2004. Proceedings. IEEE International , vol., no., pp. 305- 308, 12-15 Sept. 2004

[36] Feliciian, F.; Furber, S.B.; , "An asynchronous on-chip network router with quality-of-service (QoS) support," SOC Conference, 2004. Proceedings. IEEE International , vol., no., pp. 274- 277, 12-15 Sept. 2004

[37] Beigne, E.; Clermidy, F.; Vivet, P.; Clouard, A.; Renaudin, M.; , "An asynchronous NOC architecture providing low latency service and its multi-level design framework," Asynchronous Circuits and Systems, 2005. ASYNC 2005. Proceedings. 11th IEEE International Symposium on , vol., no., pp. 54- 63, 14-16 March 2005

[38] Martin, P.; , "Design of a virtual component neutral network-on-chip transaction layer," Design, Automation and Test in Europe, 2005. Proceedings , vol., no., pp. 336- 337 Vol. 1, 7-11 March 2005

[39] Amde, M.; Felicijan, T.; Efthymiou, A.; Edwards, D.; Lavagno, L.; , "Asynchronous on-chip networks," Computers and Digital Techniques, IEE Proceedings - , vol.152, no.2, pp. 273- 283, Mar 2005

[40] D. Kim, M. Kim, and G. E. Sobelman, "Design of a high-performance scalable CDMA router for on-chip switched networks," in ISSOC, 2005, pp.32–35.

[41] Schafer, M.K.F.; Hollstein, T.; Zimmer, H.; Glesner, M.; , "Deadlock-free routing and component placement for irregular mesh-based networks-on-chip," Computer-Aided Design, 2005. ICCAD-2005. IEEE/ACM International Conference on , vol., no., pp. 238- 245, 6-10 Nov. 2005

[42] Bartic, T.A.; Mignolet, J.-Y.; Nollet, V.; Marescaux, T.; Verkest, D.; Vernalde, S.; Lauwereins, R.; , "Topology adaptive network-on-chip design and implementation," Computers and Digital Techniques, IEE Proceedings - , vol.152, no.4, pp. 467-472, 8 July 2005

[43] Jongman Kim; Dongkook Park; Theocharides, T.; Vijaykrishnan, N.; Das, C.R.; , "A low latency router supporting adaptivity for on-chip interconnects," Design Automation Conference, 2005. Proceedings. 42nd , vol., no., pp. 559- 564, 13-17 June 2005

[44] Jiang Xu; Wolf, W.; Henkel, J.; Chakradhar, S.; , "A methodology for design, modeling, and analysis of networks-on-chip," Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on , vol., no., pp. 1778- 1781 Vol. 2, 23-26 May 2005

[45] Bjerregaard, T.; Sparso, J.; , "A router architecture for connection-oriented service guarantees in the MANGO clockless network-on-chip," Design, Automation and Test in Europe, 2005. Proceedings , vol., no., pp. 1226- 1231 Vol. 2, 7-11 March 2005

[46] Hilton, C.; Nelson, B.; , "PNoC: a flexible circuit-switched NoC for FPGA-based systems," Computers and Digital Techniques, IEE Proceedings - , vol.153, no.3, pp. 181- 188, 2 May 2006

[47] Wolkotte, P.T.; Smit, G.J.M.; Rauwerda, G.K.; Smit, L.T.; , "An Energy-Efficient Reconfigurable Circuit-Switched Network-on-Chip," Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International , vol., no., pp. 155a, 04-08 April 2005

[48] Marchal, P.; Verkest, D.; Shickova, A.; Catthoor, F.; Robert, F.; Leroy, A.; , "Spatial division multiplexing: a novel approach for guaranteed throughput on NoCs," Hardware/Software Codesign and System Synthesis, 2005. CODES+ISSS '05. Third IEEE/ACM/IFIP International Conference on , vol., no., pp.81-86, Sept. 2005

[49] T. Henriksson, D. Wiklund, and D. Liu, "VLSI implementation of a switch for on-chip networks," in DDECS, 2003

[50] Bertozzi, D.; Jalabert, A.; Srinivasan Murali; Tamhankar, R.; Stergiou, S.; Benini, L.; De Micheli, G.; , "NoC synthesis flow for customized domain specific multiprocessor systems-on-chip," Parallel and Distributed Systems, IEEE Transactions on , vol.16, no.2, pp. 113- 129, Feb. 2005

[51] Bouhraoua, A.; Elrabaa, E.L.; , "A High-Throughput Network-on-Chip Architecture for Systems-on-Chip Interconnect," System-on-Chip, 2006. International Symposium on , vol., no., pp.1-4, 13-16 Nov. 2006

[52] J. XU, W. WOLF, J. HENKEL, and S. CHAKRADHAR. "A Design Methodology for application-Specific Networks-on-Chip". ACM Trans. on Embedded Computing Systems, 5(2):263–280, May 2006.

[53] Ahmad, B.; Erdogan, A.T.; Khawam, S.; , "Architecture of a Dynamically Reconfigurable NoC for Adaptive Reconfigurable MPSoC," Adaptive Hardware and Systems, 2006. AHS 2006. First NASA/ESA Conference on , vol., no., pp.405-411, 15-18 June 2006

[54] Kuei-Chung Chang; Jih-Sheng Shen; Tien-Fu Chen; , "Evaluation and design trade-offs between circuit-switched and packet-switched NOCs for application-specific SOCs," Design Automation Conference, 2006 43rd ACM/IEEE , vol., no., pp.143-148, 0-0 0

[55] Miro Panades, I.; Greiner, A.; Sheibanyrad, A.; , "A Low Cost Network-on-Chip with Guaranteed Service Well Suited to the GALS Approach," Nano-Networks and Workshops, 2006. NanoNet '06. 1st International Conference on , vol., no., pp.1-5, Sept. 2006

[56] Richardson, T.D.; Nicopoulos, C.; Park, D.; Narayanan, V.; Yuan Xie; Das, C.; Degalahal, V.; , "A hybrid SoC interconnect with dynamic TDMA-based transaction-less buses and on-chip networks," VLSI Design, 2006. Held jointly with 5th International Conference on Embedded Systems and Design., 19th International Conference on , vol., no., pp. 8 pp., 3-7 Jan. 2006

[57] Hyung Gyu Lee; Ogras, U.Y.; Marculescu, R.; Chang, N.; , "Design space exploration and prototyping for on-chip

multimedia applications," Design Automation Conference, 2006 43rd ACM/IEEE , vol., no., pp.137-142, 0-0 0

[58] E. Salminen et al., “HIBI communication network for system-on-chip,” Journal of VLSI Signal Processing-Systems for Signal, Image, and VideoTechnology, vol. 43, no. 2, pp. 185–205, May 2006

[59] Christian Neeb, Norbert Wehn, "Designing Efficient Irregular Networks for Heterogeneous Systems-on-Chip," dsd, pp.665-672, 9th EUROMICRO Conference on Digital System Design (DSD'06), 2006

[60] Kangmin Lee; Se-Joong Lee; Hoi-Jun Yoo; , "Low-power network-on-chip for high-performance SoC design," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on , vol.14, no.2, pp. 148- 160, Feb. 2006

[61] Mullins, R.; West, A.; Moore, S.; , "The design and implementation of a low-latency on-chip network," Design Automation, 2006. Asia and South Pacific Conference on , vol., no., pp. 6 pp., 24-27 Jan. 2006

[62] Kavaldjiev, N.; Smit, G.J.M.; Jansen, P.G.; Wolkotte, P.T.; , "A virtual channel network-on-chip for GT and BE traffic," Emerging VLSI Technologies and Architectures, 2006. IEEE Computer Society Annual Symposium on , vol.00, no., pp.6 pp., 2-3 March 2006

[63] Castells-Rufas, D.; Joven, J.; Carrabina, J.; , "A Validation And Performance Evaluation Tool for ProtoNoC," System-on-Chip, 2006. International Symposium on , vol., no., pp.1-4, 13-16 Nov. 2006

[64] Ogras, U.Y.; Marculescu, R.; Hyung Gyu Lee; Naehyuck Chang; , "Communication architecture optimization: making the shortest path shorter in regular networks-on-chip," Design, Automation and Test in Europe, 2006. DATE '06. Proceedings , vol.1, no., pp.6 pp., 6-10 March 2006

[65] Se-Joong Lee; Kangmin Lee; Hoi-Jun Yoo; , "Analysis and implementation of practical, cost-effective networks on chips," Design & Test of Computers, IEEE , vol.22, no.5, pp. 422- 433, Sept.-Oct. 2005

[66] Kariniemi, H.; Nurmi, J.; , "On-Line Reconfigurable XGFT Network-on-Chip Designed for Improving the Fault-Tolerance and Manufacturability of the MPSoC Chips," Field Programmable Logic and Applications, 2006. FPL '06. International Conference on , vol., no., pp.1-6, 28-30 Aug. 2006

[67] Butts, M.; , "Synchronization through Communication in a Massively Parallel Processor Array," Micro, IEEE , vol.27, no.5, pp.32-40, Sept.-Oct. 2007

[68] Feero, B.; Pande, P.P.; , "Performance Evaluation for Three-Dimensional Networks-On-Chip," VLSI, 2007. ISVLSI '07. IEEE Computer Society Annual Symposium on , vol., no., pp.305-310, 9-11 March 2007

[69] Ainsworth, T.W.; Pinkston, T.M.; , "Characterizing the Cell EIB On-Chip Network," Micro, IEEE , vol.27, no.5, pp.6-14, Sept.-Oct. 2007

[70] Vangal, S.R.; Howard, J.; Ruhl, G.; Dighe, S.; Wilson, H.; Tschanz, J.; Finan, D.; Singh, A.; Jacob, T.; Jain, S.; Erraguntla, V.; Roberts, C.; Hoskote, Y.; Borkar, N.; Borkar, S.; , "An 80-Tile Sub-100-W TeraFLOPS Processor in 65-nm CMOS," Solid-State Circuits, IEEE Journal of , vol.43, no.1, pp.29-41, Jan. 2008

[71] Hosseinabady, M.; Kakoee, M.R.; Mathew, J.; Pradhan, D.K.; , "Reliable network-on-chip based on generalized de Bruijn graph," High Level Design Validation and Test Workshop, 2007. HLVDT 2007. IEEE International , vol., no., pp.3-10, 7-9 Nov. 2007

[72] Castells-Rufas, D.; Joven, J.; Risueño, S.; Fernandez, E. & Carrabina, J. NocMaker: A Cross-Platform Open-Source Design Space Exploration Tool for Networks on Chip INA-OCMC Workshop, Paphos, Cyprus, 2009

[73] Pavlidis, V.F.; Friedman, E.G.; , "3-D Topologies for Networks-on-Chip," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on , vol.15, no.10, pp.1081-1090, Oct. 2007

[74] Soteriou, V..; Eisley, N..; Hangsheng Wang; Bin Li; Li-Shiuan Peh; , "Polaris: A System-Level Roadmapping Toolchain for On-Chip Interconnection Networks," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on , vol.15, no.8, pp.855-868, Aug. 2007

[75] llitzky, D.A.; Hoffman, J.D.; Chun, A.; Esparza, B.P.; , "Architecture of the Scalable Communications Core's Network on Chip," Micro, IEEE , vol.27, no.5, pp.62-74, Sept.-Oct. 2007

[76] Palermo, G.; Silvano, C.; Mariani, G.; Locatelli, R.; Coppola, M.; , "Application-Specific Topology Design Customization for STNoC," Digital System Design Architectures, Methods and Tools, 2007. DSD 2007. 10th Euromicro Conference on , vol., no., pp.547-550, 29-31 Aug. 2007

[77] Gratz, P.; Changkyu Kim; Sankaralingam, K.; Hanson, H.; Shivakumar, P.; Keckler, S.W.; Burger, D.; , "On-Chip Interconnection Networks of the TRIPS Chip," Micro, IEEE , vol.27, no.5, pp.41-50, Sept.-Oct. 2007

[78] Lattard, D.; Beigne, E.; Clermidy, F.; Durand, Y.; Lemaire, R.; Vivet, P.; Berens, F.; , "A Reconfigurable Baseband Platform Based on an Asynchronous Network-on-Chip," Solid-State Circuits, IEEE Journal of , vol.43, no.1, pp.223-235, Jan. 2008

[79] Kumar, A.; Li-Shiuan Peh; Kundu, P.; Jha, N.K.; , "Toward Ideal On-Chip Communication Using Express Virtual Channels," Micro, IEEE , vol.28, no.1, pp.80-90, Jan.-Feb. 2008

[80] Janarthanan, A.; Tomko, K.A.; , "MoCSYS: A Multi-Clock Hybrid Two-Layer Router Architecture and Integrated Topology Synthesis Framework for System-Level Design of FPGA Based On-Chip Networks," VLSI Design, 2008. VLSID 2008. 21st International Conference on , vol., no., pp.397-402, 4-8 Jan. 2008

[81] Bell, S.; Edwards, B.; Amann, J.; Conlin, R.; Joyce, K.; Leung, V.; MacKay, J.; Reif, M.; Liewei Bao; Brown, J.; Mattina, M.; Chyi-Chang Miao; Ramey, C.; Wentzlaff, D.; Anderson, W.; Berger, E.; Fairbanks, N.; Khan, D.; Montenegro, F.; Stickney, J.; Zook, J.; , "TILE64 - Processor: A 64-Core SoC with Mesh Interconnect," Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International , vol., no., pp.88-598, 3-7 Feb. 2008

[82] Beigne, E.; Clermidy, F.; Lhermet, H.; Miermont, S.; Thonnart, Y.; Xuan-Tu Tran; Valentian, A.; Varreau, D.; Vivet, P.; Popon, X.; Lebreton, H.; , "An Asynchronous Power Aware and Adaptive NoC Based Circuit," Solid-State Circuits, IEEE Journal of , vol.44, no.4, pp.1167-1177, April 2009

[83] Wang, Nan; Sanusi, Azeez; Zhao, Peiyi; Mohamed, Shaheen; Bayoumi, Magdy A.; , "PMCNOC: A Pipelining Multi-Channel Central Caching Network-on-Chip Communication Architecture Design," Signal Processing Systems, 2007 IEEE Workshop on , vol., no., pp.487-492, 17-19 Oct. 2007

[84] F. A. SAMMAN, T. HOLLSTEIN, and M. GLESNER. “Networks-on-Chip based on Dynamic Wormhole Packet Identity Management”. VLSI Design, Journal of Hindawi Pub. Corp., vol. 2009(Article ID 941701):1–15, Jan 2009.

[85] Gohringer, D.; Hubner, M.; Hugot-Derville, L.; Becker, J.; , "Message Passing Interface support for the runtime adaptive multi-processor system-on-chip RAMPSoC," Embedded Computer Systems (SAMOS), 2010 International Conference on , vol., no., pp.357-364, 19-22 July 2010

[86] Altera, Inc. Applying the Benefits of Network on a Chip Architecture to FPGA System Design. Version 1.0, January 2011.

[87] E. Gabriel, G.E. Fagg, G. Bosilca, T. Angskun, J.J. Dongarra, J.M.Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R.H. Castain, D.J. Daniel, R.L. Graham, T.S. Woodall: “Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation”; In Proc. of 11th European PVM/MPI Users’ Group Meeting, Budapest, Hungary, pp. 97-104, Sept. 2004.

[88] R. L. Graham, G. M. Shipman, B. W. Barrett, R. H. Castain, G. Bosilca, A. Lumsdaine: “Open MPI: A High Performance, Heterogenous MPI”; In Proc. of Fifth International Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Networks, Barcelona, Spain, September 2006.

[89] W. Gropp, E. Lusk, A. Skjellum: “Using MPI: Portable Parallel Programming with the Message-Passing Interface”; MIT Press, 1999.

[90] M. Saldana, P. Chow: “TMD-MPI: An MPI Implementation for Multiple Processors Across Multiple FPGAs”; In Proc. of the 16th International Conference on Field-Programmable Logic and Applications (FPL 2006), Madrid, Spain, 2006.

[91] P. Mahr, C. Lörchner, H. Ishebabi, C. Bobda: “SoC-MPI: A flexible Message Passing Library for Multiprocessor Systems-on- Chips”; In Proc. of IEEE International Conference on Mexico, ReConFigurable Computing and FPGAs (ReConFig'08), Cancun, December 2008.

[92] Dagum, L.; Menon, R.; , "OpenMP: an industry standard API for shared-memory programming," Computational Science & Engineering, IEEE , vol.5, no.1, pp.46-55, Jan-Mar 1998 doi: 10.1109/99.660313

[93] NVIDIA CUDA Compute Unified Device Architecture Programming Guide; NVIDIA: Santa Clara, CA, 2007.

[94] A. Munshi, OpenCL Specification Version 1.0, 2008.

[95] http://www.khronos.org/registry/cl/.

[96] Multicore Association. http://www.multicore-association.org.

[97] MPI Forum. http://www.mpi-forum.org

[98] Reinders, J., 2007. Intel Threading Building Blocks.O’Reilly

[99] N. Popovici and T. Willhalm. Putting Intel Threading Building Blocks to work

[100] Phillipe Charles, Christopher Donowa, Kemal Ebcioglu, Christian Grothoff, Alan Kielstra, Christoph Von Praun, Vijay Saraswat, Vivek Sarkar 2005. An object-oriented Approach to Non-Uniform Cluster Computing. ACM OOPSL.

[101] William Thies , Michal Karczmarek , Saman P. Amarasinghe, StreamIt: A Language for Streaming Applications, Proceedings of the 11th International Conference on Compiler Construction, p.179-196, April 08-12, 2002

[102] R.D. Blumofe et. al. Cilk: An eficient multithreaded runtime system. In Proceedings of the Fifth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP), pages 207{216, Santa Barbara, California, July 1995.

[103] Charles E. Leiserson. The Cilk++ concurrency platform. In 46th Design Automation Conference, San Francisco, CA, July 2009. ACM/EDAC/IEEE.

[104] Parallel Programmability and the Chapel Language Bradford L. Chamberlain, David Callahan, Hans P. Zima. International Journal of High Performance Computing Applications, August 2007, 21(3): 291-312.

[105] Axum programmer’s guide. Microsoft corporation.

[106] E. Salminen, A. Kulmala, and T. H¨am¨al¨ainen. On network-on-chip comparison. In Euromicro DSD, Aug. 2007, pp. 503–510.

[107] J. Duato, S. Yalamanchili, N. Lionel. Interconnection Networks: An Engineering Approach. Morgan Kaufmann Publishers Inc., San Francisco, CA, 2002.

Eduard Fernandez-Alonso received his BsC degree in Computer Science and MsC degree in Micro and Nano Electronics from Universitat Autònoma de Barcelona (UAB), Bellaterra, Spain. He is currently at CaiaC (Centre for research in Ambient Intelligence and Accessibility in Catalonia), research center at Universitat Autònoma de Barcelona, where he is doing his PhD studies. His main research interests include parallel computing, Network-on-Chip-Based Multiprocessor Systems, and parallel programming models.

David Castells-Rufas received his BsC degree in Computer Science from Universitat Autònoma de Barcelona. He holds a MsC in Research in Microelectronics from Universitat Autònoma de Barcelona. He is currently the head of the Embedded Systems unit at CAIAC Research Centre at Universitat Autònoma de Barcelona (UAB), where he is doing his PhD studies. His primary research interests include parallel computing, Network-on-Chip Based Multiprocessor Systems, and parallel programming models. He is also associate lecturer in the Microelectronics department of the same university.

Jaume Joven received his MsC degree and PhD degree in Computer Science from Universitat Autònoma de Barcelona (UAB). During his research, he received the best paper award for “xENoC - An eXperimental Network-On-Chip Environment for Parallel Distributed Computing on NoC-based MPSoC Architectures". Nowadays, he continues his research career as a post-doc researcher at EPFL-LSI/iNoCs. His main research interests are focused on the embedded NoC-based MPSoCs, ranging from circuit and system-level design of custom NoCs architectures, up to system-level software for QoS resource allocation and runtime reconfiguration, as well as, middleware and parallel programming models.

Jordi Carrabina leads CAIAC Research Centre at Universitat Autònoma de Barcelona (Spain), member of Catalan IT network TECNIO. He received his PhD degree from Universitat Autònoma de Barcelona. His main interests are Microelectronic Systems oriented to Embedded Platform-based Design using System Level Design Methodologies using SoC/NoC Architectures and Printed Microelectronics Technologies in the Ambient Intelligence Domain. He is a Prof. T. at Universitat Autònoma de Barcelona where is Teaching EE and CS at the Engineering School and in the MA of Micro & Nanoelectronics Engineering and Multimedia technologies, at UAB and Embedded Systems at UPV-EHU.