UNIMAP: UNIFIED FRAMEWORK FOR NETWORK-ON-CHIP APPLICATION MAPPING RESEARCH

Ciprian RADU, Lucian VINȚAN

"Lucian Blaga" University of Sibiu, Romania, Engineering Faculty, Emil Cioran st. no. 4, 550025 Sibiu, Phone +40-269-217928, Fax +40-269-212716, e-mail {ciprian.radu, lucian.vintan}@ulbsibiu.ro

Abstract: Network-on-Chip architectures have been recently proposed to address the limitations of bus-based interconnection networks. They focus on modularity, scalability and intrinsic support for heterogeneous Systems-on-Chip. Network-on-Chip application mapping is an NP-hard problem that deals with the topological mapping of Intellectual Property cores onto network tiles. This paper outlines the main contributions of an ongoing PhD research, which addresses this problem. At the Advanced Computer Architecture & Processing Systems (ACAPS) research lab, we developed UniMap, a framework that evaluates and optimizes algorithms for Network-on-Chip application mapping, through a unified approach. Using this tool, multiple application mapping algorithms can be tested on the same network architecture. The framework is also intended to be flexible so that different interconnection designs can be used when comparing the performance of different algorithms. We present here UniMap's design and we show how it contributes to the Network-on-Chip application mapping research.

Key words: Network-on-Chip, NoC, application mapping, evaluation, optimization, simulation, framework

1. Introduction

The Network-on-Chip (NoC) architecture is a communication network that is used on a chip. The idea of a Network-on-Chip appeared in the 90's. However, research started only from year 2000. The most important papers that pioneered this new research field are the ones of: Guerrier and Greiner [1], Hemani et al. [2], Dally and Towles [3], Wingard [4], Rijpkema et al. [5], Kumar et al. [6] and Micheli and Benini [7].

In comparison with traditional, bus-based, interconnection networks, Networks-on-Chip have the following main characteristics: structured wiring, modularity, scalability, reliability, data abstraction (packet-based communication) and network modeling (network layers).

According to Duranton et al. [8], NoCs will have an increasing importance in the following years. The growing interest in this area of research is stressed out in HiPEAC's vision: interconnects is one of the clusters on which HiPEAC's roadmap is built.

The Network-on-Chip proposed by Dally and Towles [3] was introduced as a better alternative to global wiring structures, used to interconnect different Intellectual Property (IP) blocks. Their NoC has a regular tile-based architecture that offers several advantages over traditional interconnection networks. The structured network wiring allows for a better control of the electrical parameters of the network's wires. This provides the opportunity to obtain reduced power consumption. Another advantage of NoCs is given by modularity and standard network interfaces, which provide re-usability and interoperability of the modules. Wiring resources are shared by the communicating IPs: when one module is not communicating, other modules can still use the wiring resources used by the (now) idle module. No global wiring is used by a Network-on-Chip. The IP cores communicate by sending packets to one another.

The major NoC research problems are formulated by Marculescu et al. [9]: (1) traffic modeling and benchmarking, (2) application scheduling, (3) application mapping, (4) routing, (5) switching, (6) Quality of Service (QoS) and congestion control, (7) power and thermal management, (8) reliability and fault tolerance, (9) topology design, (10) router design, (11) network channel design, (12) floorplanning and layout design, (13) clocking and power distribution, (14) analysis and simulation and (15) prototyping, testing and verification. All these NoC research problems lead to a very complex and difficult to explore design space, which is mainly characterized by three dimensions: communication infrastructure, communication paradigm and application.

Network-on-Chip application mapping problem is defined by Hu and Marculescu [10] as the topological

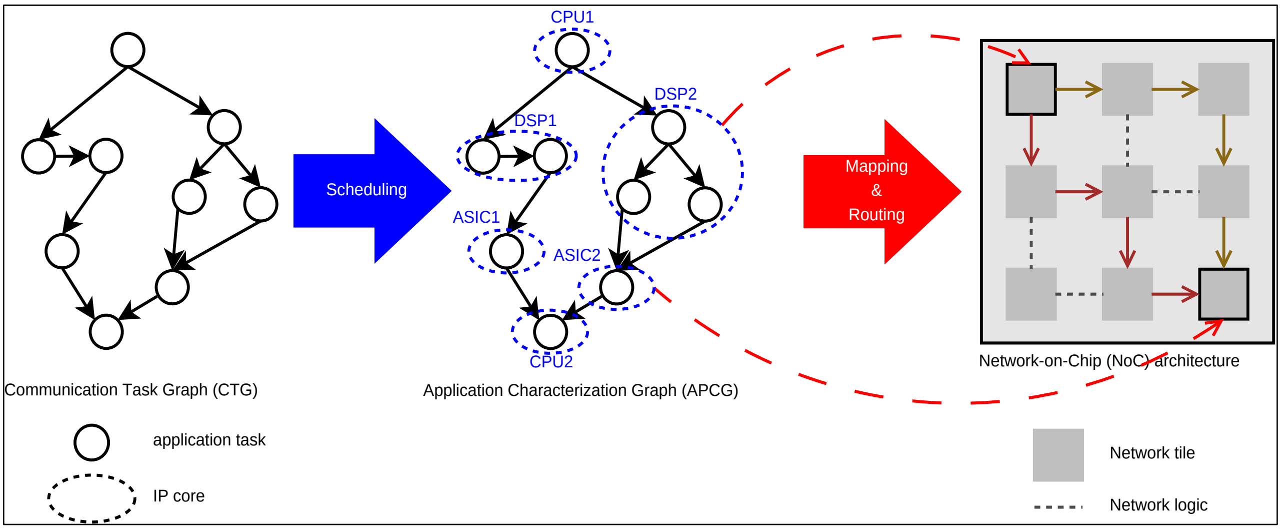

Figure 1. Network-on-Chip application mapping problem, in relation with scheduling and routing problems

placement of IP cores onto NoC tiles. Mapping is an instance of the quadratic assignment problem, which is shown to be NP-hard by Garey and Johnson [11]. Let *N* be the number of IP cores that need to be mapped onto *M* network nodes (N ≤ M). There are $\frac{M!}{(M-N)!}$ possible mappings. Obviously, this number increases factorially with the problem size. For example, a NoC with 4x4 tiles allows 16! mappings, a number high enough that several years would be required by a processor to run an exhaustive search. Therefore, heuristic algorithms are required. The purpose of such an algorithm is to determine the best topological placement of cores onto network nodes. The optimality of a certain mapping is given by the best trade-off between some network performance metrics like bandwidth, latency, energy consumption etc.

Before the IPs can be mapped onto network nodes, the assignment of tasks to heterogeneous IP cores must be determined. This is why the application mapping problem is related to the scheduling problem.

As shown by Hu and Marculescu [12], the application mapping problem is tightly connected to the routing problem, too (see Figure 1). While a good mapping of cores onto network nodes can lead to energy savings, the routes used by the cores to communicate can have a great impact on the NoC's performance (due to network congestion).

The existing application mapping algorithms are evaluated only on some specific Network-on-Chip architectures (e.g.: 2D mesh NoC topology). Also, they cannot be directly compared because a common evaluation methodology is still missing (each researcher used his own simulation methodology).

This paper proposes a unified approach for the application mapping algorithms' evaluation and optimization, called UniMap. As a PhD research, we developed at the Advanced Computer Architecture & Processing Systems (ACAPS) [13] research lab a flexible framework which allows the study of application mapping algorithms on different NoC designs. More precisely, an evaluation of these algorithms on different scalable NoC topologies is performed.

Since the relevance of application mapping algorithms is still mainly researched on a small class of NoCs (e.g.: 2D meshes), this unified framework will contribute at determining the most suitable mapping algorithm, for a certain applications set and NoC design. Due to the huge design space of NoCs, application mapping algorithms can be optimized for a specific NoC architecture.

Therefore, UniMap contributes to an Automatic Design Space Exploration (ADSE) of NoCs with the purpose of finding the best mapping (in terms of energy consumption, network latency and other objectives), for any (parallel) application, on a given NoC architecture.

The rest of this paper is organized as follows. Section 2 briefly presents some related work. The next section describes UniMap's design and outlines its features. Then, some results are presented and finally, we conclude by summarizing UniMap's contribution to the research community and outlining the directions for future work.

2. Related work

The application-based selection of a NoC topology is addressed by Murali and Micheli [14]. A general mapping algorithm extends NMAP [15] so that it can be applied on other topologies too, not just on a 2D mesh. Thus, topologies like torus, hypercube, 3-stage clos and butterfly are also considered. A tool called SUNMAP is designed with the purpose of automatically selecting the best NoC topology for a given application. The general mapping algorithm is used to produce a mapping of cores onto the researched topologies. The tool uses multiple routing protocols: dimension ordered routing, minimum-path and traffic splitting. The best topology is selected based on floorplanning information. Also, the following objectives are considered: the minimization of the average packet latency, by satisfying bandwidth constraints, and the minimization of power consumption, by satisfying area constraints. SUNMAP is a complex tool for automatically evaluating different topologies for Networks-on-Chip, in an application-aware context. However, only a single application mapping algorithm is considered. Taking into account other mapping algorithms too, will provide a more comprehensive view on the performance of different NoC architectures. SUNMAP is focused on network topology selection and generation, rather than application mapping. The mapping is static, i.e. it does not take into consideration the dynamic effects of the network. Mapping evaluation uses analytical models, instead of a NoC simulator.

Ascia et al. [16] proposed a framework for simulation and exploration of the mapping space. They use a NoC simulator to evaluate the mappings. Also, their approach is multiobjective. The generated mappings try to optimize both power and performance metrics. Three kinds of algorithms are used: genetic, branch-and-bound and a multiobjective version of NMAP. Their NoC architecture uses a 2D mesh topology with Dimension Order Routing and wormhole switching.

The first of the approaches presented in this section is flexible in terms of NoC architecture. The second approach is complementary to the first: it is flexible in terms of NoC application mapping algorithm.

UniMap takes the advantages of both approaches presented above. It aims to be flexible in both application mapping algorithms and NoC architecture design space. Multiple application mapping algorithms can be used to map real applications onto different NoC architectures. The mappings can be evaluated using either analytical models or a Network-on-Chip simulator. Also, multiple objectives can be aimed when searching for the best mapping. UniMap represents a framework for the evaluation and optimization of NoC application mapping algorithms in a unified manner.

3. UniMap design

This section describes UniMap's design, which is illustrated in Figure 2.

Our developed framework is made of the following major components:

- a model for representing real applications;
- a module for scheduling the application tasks onto IP cores;
- a module that contains application mapping algorithms;
- a model for representing different Network-on-Chip architectures;
- a Network-on-Chip simulator.

This design tries to reflect as best as possible the interaction between the Network-on-Chip application mapping problem and the other two NoC problems with which it interacts (see Figure 1). Also, in order to keep UniMap flexible, reusable and modular, we try to maintain its main components, enumerated above, as decoupled as possible.

We describe real applications and Network-on-Chip architectures through abstract eXtensible Markup Language (XML) models. This way, the three UniMap modules (Scheduler, Mapper and NoC simulator) do not interact directly. This approach theoretically allows any NoC simulator to be used with UniMap. Similarly, any scheduling or mapping algorithm can be integrated as easy as possible. Our XML interface models are lightweight representations for real applications and they extract only the important characteristics of the NoC architectures. They incur some overhead but, they offer flexibility, scalability, modularity and reusability. The disadvantage is that applications are not modeled in detail. Actually, Network-on-Chip benchmarking is still an open problem. The Open Core Protocol International Partnership (OCP-IP) currently works with some of the most prestigious NoC research groups from the world to build a suitable benchmarking methodology for Network-on-Chip simulation. We recently found from Salminen et al. [17] that OCP-IP started developing, in parallel with us, a design similar to UniMap's, for addressing Network-on-Chip problems.

3.1. Application model

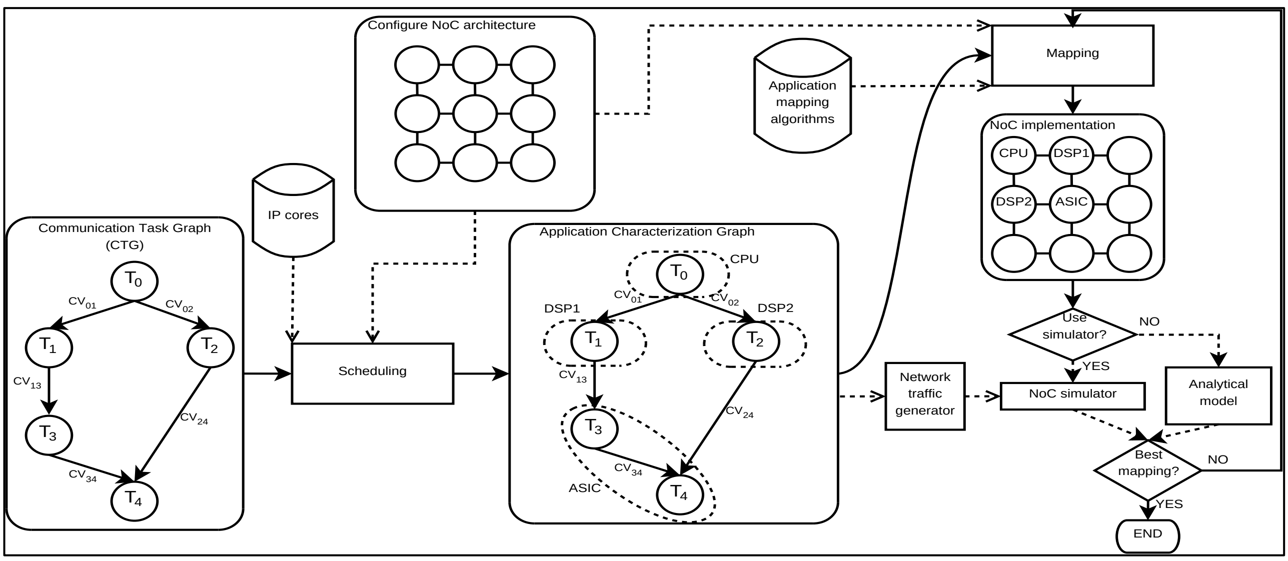

Figure 2. UniMap design flow

With a Communication Task Graph (CTG) [9], an application is partitioned into a set of tasks. A CTG also specifies the data dependencies among these tasks and how much volume of information is communicated from one task to another (e.g.: CV_{01} denotes the communication volume from task T_0 to task T_1). Therefore, a CTG captures only the communication behavior of a real application.

We obtain CTGs in two distinct ways: (1) from realistic embedded applications, using the E3S benchmark suite [18] and (2) from real-world multithreaded applications, using the CETA tool [19].

A Communication Task Graph is described in UniMap using an XML structure. UniMap describes the IP cores from E3S through an XML structure also. For each IP, information like thread execution time and power consumption, for a given application task, are known.

3.2. Scheduler module

The application tasks must be first assigned to the heterogeneous IP cores. This is typically done with a scheduling algorithm. Besides simply assigning tasks to IPs, a scheduling algorithm also determines tasks' execution order. This is useful when dealing with real-time constraints. Since the main goal of this framework is to address the mapping problem, we developed a simple scheduler. From the IP core library, our scheduler selects, for each application task, the core that executes that task the fastest.

The output of the scheduling algorithm is an Application Characterization Graph (APCG) [9]. Compared to the CTG, the APCG specifies the assignment of application tasks to IP cores and it represents the input for the mapping phase. Like the CTG, the APCG is also described through an XML representation.

3.3. Mapper module

The Mapper module holds the application mapping algorithms library of the unified framework. An application mapping algorithm has the role of topologically placing the IP cores onto the available Network-on-Chip nodes. It uses the Application Characterization Graph provided by the Scheduler module and it assigns all the IP cores from this graph to NoC nodes. The mapping produced by the algorithm is also described using an XML schema.

We have successfully integrated two of the first algorithms proposed by the research community for NoC application mapping. Simulated Annealing and Branch-and-Bound, introduced by Hu and Marculescu [10] [12], are now part of UniMap.

We have also developed an Optimized Simulated Annealing (OSA) algorithm for NoC application mapping. Compared to a general Simulated Annealing, OSA is much faster, without finding worse solutions. Further details about OSA are available in Radu and Vințan [20].

Currently, we are extending UniMap's Mapper module with Genetic and Evolutionary Algorithms. Other algorithms, like NMAP, may also be integrated.

3.4. Network model

XML schemas are also used in UniMap to describe a Network-on-Chip architecture. We have created models for NoC nodes, links and topologies.

The purpose of the network model is to capture those characteristics about the NoC architecture that are used by the application mapping algorithms.

3.5. Network-on-Chip simulator

As it is shown in Figure 2, a mapping can be evaluated with an analytical model, which provides a simple representation of a NoC architecture. Such an approach is used by Hu and Marculescu in [10][12], where a bit energy analytical model is presented. UniMap implements and uses this model. However, even if it is simple and fast to evaluate, such an approach is not very accurate. Network dynamic effects (like congestion) may only be captured with a simulator.

In order to gain to more accuracy when evaluating mappings, we have developed in UniMap a Network-on-Chip simulator, called ns-3 NoC. It is based on ns-3 [21], a scalable simulator for Internet systems, one of the fastest and most memory efficient simulators currently available. Our NoC simulator currently allows the user to specify: the packet size, packet injection rate, buffer size, network size, switching mechanism (Store-and-Forward, Virtual Cut-Through and Wormhole), routing protocol (XY, YX and two adaptive protocols that consider the network's load), network topology (2D mesh, Irvine) and traffic patterns. It can evaluate the simulated NoC in terms of network latency and throughput. Further details about this simulator are available in Radu and Vințan [22].

Since the first version of ns-3 NoC, we have extended our simulator with: other topologies (2D torus, 3D mesh, 3D torus), a network traffic generator based on real applications (described with CTGs and APCGs) and ORION 2.0, a NoC power and area model, designed by Kahng et al. [23].

3.5.1. Network traffic generator

UniMap emulates the communication between application tasks by modeling each Processing Element (PE) from the Network-on-Chip as the Finite State Machine (FSM) described in Figure 3.

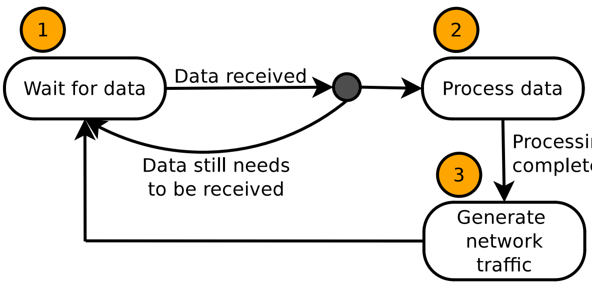

Figure 3. The FSM associated to a PE

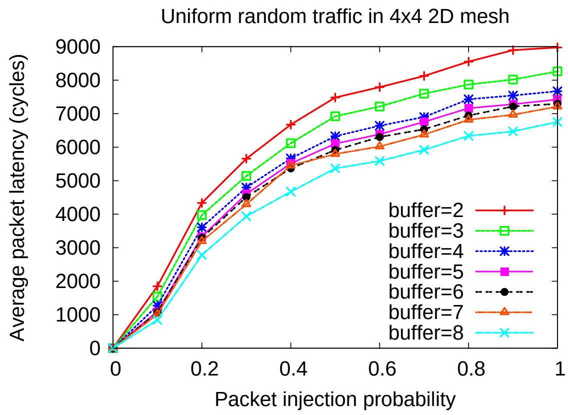

Figure 4. The average packet latency on a 4x4 Irvine NoC architecture, while the size of the input buffers varies uniformly

Initially, all the Processing Elements (IP cores) are in state 1, waiting to receive data to process. The first PE which will enter in the processing phase (state 2) is the one that contains the root task of the CTG. Any PE enters in the processing state after it has received all the data from the other PEs it depends on (data dependencies are modeled by the CTG). Any PE will stay in this state for a period of time equal to the time needed by the IP to process the task. After this time is elapsed, the task's processing is finished, and the PE enters in state 3. At this point, the processed data is injected into the network (the communication volume is specified by the CTG). All these data will be divided into packets and sent through the NoC.

We have briefly presented in this section the current state of UniMap, our unified framework for Network-on-Chip application mapping evaluation and optimization, which is publicly available at [24].

4. Results

This section presents some of the results that we obtained until now with UniMap.

A preliminary evaluation of UniMap's Network-on-Chip simulator has been done in Radu and Vințan [22]. We showed how differently the performance of the network varies based on the traffic pattern used and

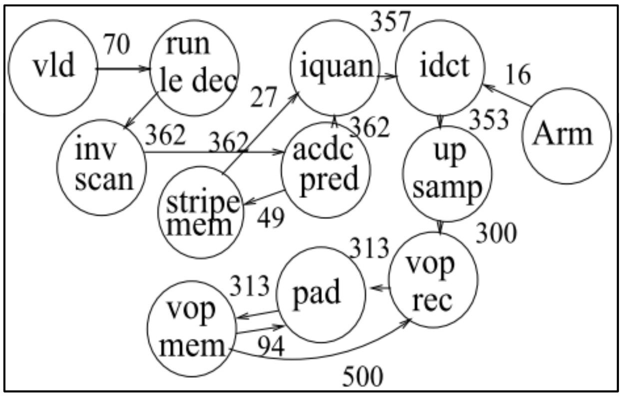

Figure 6. Application Characterization Graph for VOPD

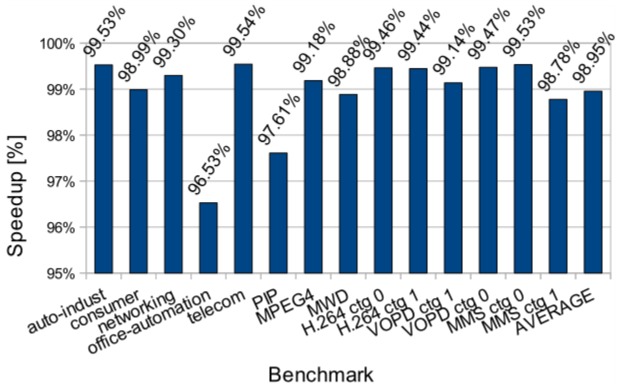

Figure 5. OSA speedup over SA

some NoC parameters, too (like the amount of buffering resources or network clock frequency).

As shown in Figure 4 [22], as more buffering resources are available, the performance of the NoC architecture improves. As expected, the size of the input channel buffers becomes more important as the number of packets injected into the network increases. Using a uniform random traffic pattern, we ran a NoC simulation for 10000 clock cycles, with 1000 warm-up cycles. At each cycle, a flit can be injected in any node of the network, with a certain probability of injection. Dimension Order Routing with wormhole switching has been used on a 4x4 Irvine NoC architecture [25]. Each packet had 9 flits, and the size of the input channels was varied uniformly, from 2 up to 8 flits.

The performance of a NoC architecture is directly influenced by the application mapping algorithm. By comparing the results of different mapping algorithms, it can be determined which algorithm is most suited for mapping a specific application on a certain NoC design.

E3S benchmarks and other real benchmarks were mapped with UniMap, using two NoC application mapping algorithms: Simulated Annealing and Branch-and-Bound. Using an energy bit model, we evaluated the two algorithms in terms of mapping speed, memory consumption and solution quality. We have thus confirmed the findings of Hu and Marculescu: Simulated Annealing (SA) can find better results than Branch-and-Bound (BB), but BB is tens of times faster and more feasible. SA tends to run for days when the 2D mesh NoC size is increased to 10x10.

We also proposed an Optimized Simulated Annealing (OSA) algorithm for Network on Chip application mapping Radu and Vințan [20]. OSA uses application knowledge and performs an implicit and dynamic IP core clustering. Our Simulated Annealing is much faster then a generic SA. As it is shown in Figure 5, OSA is much faster than Hu and Marculescu's Simulated Annealing. We obtained an average speedup of 98.95%. This result is correlated with our theoretical expectations. The "lowest" speedups are on office-automation and PIP, the benchmarks with the smallest number of IP cores. We justify this significant speed gain mainly by the way OSA computes the number of iterations per temperature level. Much more details about OSA are available in Radu and Vințan [20].

We compared some of the results produced by

run le dec	inv scan	pad	vop mem
vld	acdc pred	Arm	vop rec
stripe mem	iquan	idct	up samp

Figure 7. UniMap's best VOPD mapping, onto a 4x3 2D mesh NoC

UniMap with some of SUNMAP's results. For example, we have found out that all UniMap's mapping algorithms managed to find a better mapping onto a 4x3 2D mesh NoC, for the Video Object Plane Decoder (VOPD) application (see Figure 6 [14]). The best mapping found with UniMap is shown in Figure 7, while the best mapping given by SUNMAP is in Figure 8. UniMap's mapping is essentially better because IP core "stripe mem" is placed much closer to IP cores "acdc pred" and "iquan" (the two cores with which core "stripe mem" communicates).

5. Conclusions and further work

We have presented in this paper UniMap: a unified framework for Network-on-Chip application mapping research, developed as a PhD work, at the Advanced Computer Architecture & Processing Systems (ACAPS) research lab.

We showed UniMap's design, which is flexible, modular, reusable and scalable.

UniMap aims to evaluate and optimize different NoC application mapping algorithms on multiple Network-on-Chip architectures. We have evaluated some NoC application mapping algorithms, using an analytical model and also a NoC simulator. We have also optimized an algorithm and showed it produces the same best results, in a much faster time (as compared to the unoptimized version).

As further work, we plan to evaluate some genetic and evolutionary algorithms. We also intend to determine best mappings by considering multiple objectives (like application runtime, network energy consumption and network area). Finally, we intend to use FADSE tool [25] [26] for an automatic design space exploration of NoC architectures. This approach will allow for a more complex approach to application specific NoC mapping and Network-on-Chip architecture synthesis.

UniMap's main original aspect is its unified approach to NoC application mapping. Currently, to the best of our knowledge, there is no such framework publicly available.

idct	Arm	vop mem	pad
iquan	up samp	vop rec	stripe mem
acdc pred	inv scan	run le dec	vld

Figure 8. SUNMAP's best VOPD mapping, onto a 4x3 2D mesh NoC

Acknowledgments

This research was supported POSDRU financing contract POSDRU 7706.

References

[1] P. Guerrier and A. Greiner, "A generic architecture for on-chip packet-switched interconnections," *Proceedings of the conference on Design, automation and test in Europe*, p. 250–256, 2000.

[2] A. Hemani et al., "Network on chip: An architecture for billion transistor era," in *Proceeding of the IEEE NorChip Conference*, 2000, p. 166–173.

[3] W. J. Dally and B. Towles, "Route packets, not wires: on-chip inteconnection networks," in *Proceedings of the 38th annual Design Automation Conference*, Las Vegas, Nevada, United States, 2001, pp. 684-689.

[4] D. Wingard, "Micronetwork-based integration for SOCs: 673," *Proceedings of the 38th annual Design Automation Conference*, p. 677–, 2001.

[5] E. Rijpkema, K. Goossens, and P. Wielage, "A Router Architecture for Networks on Silicon," *IN PROCEEDINGS OF PROGRESS 2001, 2ND WORKSHOP ON EMBEDDED SYSTEMS*, p. 181--188, 2001.

[6] S. Kumar et al., "A network on chip architecture and design methodology," in *isvlsi*, 2002, p. 0117.

[7] G. de Micheli and L. Benini, "Networks on Chip: A New Paradigm for Systems on Chip Design," *Proceedings of the conference on Design, automation and test in Europe*, p. 418–, 2002.

[8] M. Duranton et al., "The HiPEAC Vision," *HiPEAC Roadmap*, 2010. [Online]. Available: http://www.hipeac.net/system/files/LR_3910_hipeac_roadmap-2010-v3.pdf.

[9] R. Marculescu, U. Y. Ogras, L.-S. Peh, N. E. Jerger, and Y. Hoskote, "Outstanding research problems in NoC design: system, microarchitecture,

and circuit perspectives," *Trans. Comp.-Aided Des. Integ. Cir. Sys.*, vol. 28, no. 1, pp. 3-21, 2009.

[10] J. Hu and R. Marculescu, "Energy-aware mapping for tile-based NoC architectures under performance constraints," in *Proceedings of the 2003 Asia and South Pacific Design Automation Conference*, Kitakyushu, Japan, 2003, pp. 233-239.

[11] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*. WH Freeman & Co. New York, NY, USA, 1979.

[12] J. Hu and R. Marculescu, "Energy- and performance-aware mapping for regular NoC architectures," *IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS*, vol. 24, no. 4, p. 551--562, 2005.

[13] "Advanced Computer Architecture & Processing Systems," *Advanced Computer Architecture & Processing Systems (ACAPS)*. [Online]. Available: http://acaps.ulbsibiu.ro. [Accessed: 21-Apr-2011].

[14] S. Murali and G. D. Micheli, "SUNMAP: a tool for automatic topology selection and generation for NoCs," in *Proceedings of the 41st annual Design Automation Conference*, San Diego, CA, USA, 2004, pp. 914-919.

[15] S. Murali and G. D. Micheli, "Bandwidth-Constrained Mapping of Cores onto NoC Architectures," in *Proceedings of the conference on Design, automation and test in Europe - Volume 2*, 2004, p. 20896.

[16] G. Ascia, V. Catania, and M. Palesi, "A Multi-Objective Genetic Approach to Mapping Problem on Network-on-Chip," *JUCS*, vol. 22, p. 2006.

[17] E. Salminen, K. Srinivasan, and Z. Lu, "OCP-IP Network-on-chip benchmarking workgroup," *OCP-IP*, Dec-2010. [Online]. Available: http://www.ocpip.org/uploads/dynamic_areas/Cv8XdaKTKDztFpWKPqsl/6189/NoC%20Working%20Group%20Overview%20WP.pdf.

[18] "The Embedded System Synthesis Benchmarks Suite (E3S) website." [Online]. Available: http://ziyang.eecs.umich.edu/~dickrp/e3s/.

[19] A.-H. Liu and R. P. Dick, "Automatic run-time extraction of communication graphs from multithreaded applications," in *Proceedings of the 4th international conference on Hardware/software codesign and system synthesis*, Seoul, Korea, 2006, pp. 46-51.

[20] C. Radu and L. Vinţan, "Optimized Simulated Annealing for Network-on-Chip Application Mapping," presented at the International Conference on Control Systems and Computer Science, Bucharest, Romania, in press.

[21] "The ns-3 network simulator website." [Online]. Available: http://www.nsnam.org.

[22] C. Radu and L. Vinţan, "Optimizing Application Mapping Algorithms for NoCs through a Unified Framework," in *Roedunet International Conference (RoEduNet), 2010 9th*, Sibiu, Romania, 2010, pp. 259 - 264.

[23] A. B. Kahng, B. Li, L.-S. Peh, and K. Samadi, "ORION 2.0: a fast and accurate NoC power and area model for early-stage design space exploration," in *Proceedings of the Conference on Design, Automation and Test in Europe*, 3001 Leuven, Belgium, Belgium, 2009, p. 423–428.

[24] C. Radu, "Unified Framework for Network-on-Chip Application Mapping," *unimap - Project Hosting on Google Code*. [Online]. Available: https://code.google.com/p/unimap/. [Accessed: 04-Feb-2011].

[25] S. E. Lee and N. Bagherzadeh, "Increasing the throughput of an adaptive router in network-on-chip (NoC)," in *Proceedings of the 4th international conference on Hardware/software codesign and system synthesis*, Seoul, Korea, 2006, pp. 82-87.

[26] H. Calborean and L. Vintan, "An Automatic Design Space Exploration Framework for Multicore Architecture Optimizations," in *Roedunet International Conference (RoEduNet), 2010 9th*, Sibiu, Romania, 2010, p. 202–207.

[27] H. Calborean, R. Jahr, T. Ungerer, and L. Vintan, "Optimizing a Superscalar System using Multi-objective Design Space Exploration," presented at the The 18th International Conference On Control Systems And Computer Science, Bucharest Romania, 2011.