

# DCM: An IP for the Autonomous Control of Optical and Electrical Reconfigurable NoCs

Wolfgang Büter, Christof Osewold, Daniel Gregorek, Alberto García-Ortiz  
Institute of Electrodynamics and Microelectronics (ITEM.ids), University of Bremen  
{bueter, osewold, gregorek, agarcia}@ids.uni-bremen.de

**Abstract**—The increasing requirements for bandwidth and quality-of-service motivate the use of parallel interconnect architectures with several degrees of reconfiguration. This paper presents an IP, called Distributed Channel Management (DCM), to extend existing packet-switched NoCs with a reconfigurable point-to-point network seamlessly, i.e., without the need for any modification on the routers. The configuration of the reconfigurable network takes place dynamically and autonomously, so that the topology can be changed at run time. Furthermore, the architecture is scalable due to the autonomous decentralized administration of the links.

The Paper reports a thorough experimental analysis of the overhead of the approach at the gate level that considers different network parameters such as flit size and timing constraints.

**Keywords**—NoC, reconfigurable NoC, optical NoC, SoC

## I. INTRODUCTION

The integration capabilities of current SoCs allow an increasing number of subsystems to be implemented in one chip. In this new scenario, the communication architecture is a critical factor determining the performance of application specific many-tile SoCs. Although point-to-point connections present an optimal solution for the performance and the energy consumption, they are not scalable. Networks-on-Chip provide a scalable communication architecture for MPSoCs.

The power consumption of a NoC is approximately proportional to the number of hops between two communicating units [1]. Therefore, it is essential to design a reconfigurable architecture to guarantee the energy efficiency and QoS in various applications. The energy consumption can be minimized considerably by reducing the average distance between source and destination [2].

State-of-the-art design methodologies consider the traffic characteristics of an application to reduce that distance either during the design time or through configuration at the start of program. Another approach to reduce the distance and the energy consumption for the transmission is to use parallel networks with better energetic properties. Networks with low-swing transmission techniques and optical transmission media can lower energy consumption substantial [3][4][5][6].

The traffic-characteristics of the individual applications vary significantly and the communication-network needs to be continuously adjusted to the current conditions during run time. More than 1500 NoC-Configurations are analyzed [7], evincing no configuration to give optimal performance-results for a range of traffic-characteristics. Due to the requirements of the dynamic traffic-characteristics in different applications and functions research is going towards a variety of application-978-3-9815370-2-4/DATE14/©2014 EDAA

specific and reconfigurable NoCs [2][8][4].

In ReNoC the routers of a packet-switched NoC are extended by topology switches. This switches serve as a wrapper and can be configured to bypass the routers [8]. An alternative is presented in [9]. A virtual channel serves as a bypass in this architecture. If the highest priority is assigned to this channel, the pipeline of the router is bypassed.

The topology of the NoC is extended by configuration switches in [10]. These can be configured to establish a point-to-point connection between physically not adjacent tiles.

These approaches have in common that the task graph of the executed application is known prior to the configuration and the network is configured before the start. Furthermore, these works do not discuss in detail how the configuration of the topology is performed.

Dynamic configuration of the path has been proposed in [2]. The probability of traffic leaving an output port taking a 0° turn is 70-80%. Exploiting this fact, muxes at the entrance and the exit of the corresponding directions help making the bypass configurable and produce a long-range skip-link. The link-configuration is dynamic without requiring a task-graph in advance, however skip-links cannot accomplish turns.

In the previous approaches, the path-configuration can have negative impact on the remaining traffic in the networks: the number of hops for packets that do not use a pre-configured path is increased if the configured path blocks the shortest way. A parallel optical configurable circuit-switched network is introduced in [11]. The arbitration occurs locally and electrically. The switches consist of optical resonator-structures which lie in the top metal layers. This parallel optical network shows different transmission characteristics than an electrical one, since the energy needed is independent of the transmission path length.

An interconnect-architecture, where the arbitration occurs through an optical token is introduced in [12]. An optical data-path is triggered and the data-line passes through each PE, such as in a ring. Thus leading to the drawbacks that available bandwidth of passing PEs is shared and the arbitration takes very long due to the used light-mode. In order to improve the arbitration, a stream of several tokens is introduced in [13]. The availability of the bandwidth is improved, however the bandwidth is still shared between every participating PE.

To cope with these issues, the contribution of our paper is a flexible unit which controls various electrical and optical on-chip communication networks decentrally and autonomously. The DCM can be applied in a packet-switched NoC without any modifications. We define a set of functions to realize that flexibility. Furthermore we demonstrate the generality by the implementation of the DCM in three different state-of-the-art

architectures.

The rest of the paper is organized as follows: Section II presents the architecture of the DCM, and afterwards the integration for several state of the art architectures. In section III, we report the gate level synthesis results and evaluate the IP. Finally, section IV concludes the paper.

## II. ARCHITECTURE

In order to extend a packet-switched NoC by a parallel circuit-switched NoC, we propose a dedicated module which performs the management of the circuit-switched channels autonomously. The IP, called Distributed Channel Management (DCM), allows to increase the functionality of a packet-switched NoC without modifications in the routers. The DCM is used to configure a parallel network for guaranteed traffic. The best-effort traffic as well as the ingoing and outgoing configuration messages are exchanged using the basic network.

Figure 1 illustrates the strategy to extend a packet-switched network using the DCM. Each DCM is inserted between a PE and the router's local port, and is connected to a configuration switch as well. These switches build the parallel circuit-switched network and are configured through the DCMs self-signal. The communication to and from the PE is undisturbed. Furthermore, there is no need to connect a complete DCM to every router: the topologies of the basic and the parallel networks can be selected independently. This feature allows an increasing flexibility, which is essential when working with optical NoCs.

The task for the DCM is to manage the circuit-switched paths and to control the switches of the parallel network using a decentralized and dynamic strategy. When the application running at the PE requires a new (circuit-switching) path, it sends a configuration message to a DCM. The exact path-topology can be decided by the PE at run-time using the latest network-information available. The DCM which receives the request handles the configuration of all the switches involved by sending parallel messages to the needed DCMs. Every addressed DCM confirms or denies the request autonomously. Finally, the responses are combined in a single message (i.e., ACK/NACK) which is sent to the PE. As an additional feature, the switches can be deconfigured automatically if the path cannot be acknowledged. Every PE can create independently and simultaneously a new path; the DCM structure ensures the consistency of the paths.

The architecture of the module is divided into two sub-modules DCMC and DCMD (see fig. 1) which can be implemented independently. The data transfer as well as the (potential)

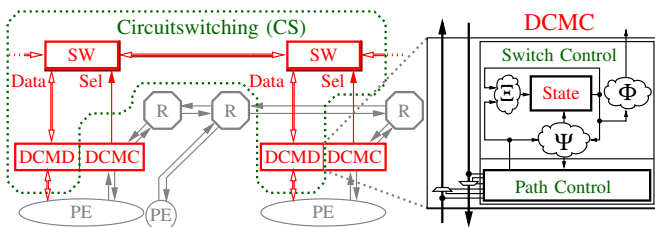


Figure 1: Generic packet-based network extended by DCM

synchronization of different clock domains between circuit-switched and packet-switched NoCs occurs in the DCMD module. It is composed by two submodules, the Tx-data and the Rx-data. The Tx-data converts the PE messages to the bit width of the parallel NoC; while the Rx-Data performs the opposite operation. These blocks can handle a WDM strategy as usually employed in optical on-chip interconnects.

The DCMC is responsible for detection, extraction and processing of the configuration messages. It operates as a kind of bypass which capture the configuration messages, therefore the router does not need to be modified. Parts of the configuration messages are stored in the DCMC to control the switches. Moreover, the results of the configuration request evaluation are generated in the answer-generator. The DCMC is authorized to interrupt a PE's data transfer and replace the current transfer by own data. A combinational function verifies the received configuration is compatible to the current configuration. Provided that it is compatible, a new control signal for the switch is generated. Additionally, the information to configure the switch is stored and a response is generated. To send the reply, a potentially active transfer of the PE is interrupted or stopped for the duration of the message.

The semantic of the configuration messages depends on the characteristics of the parallel network and its switches. We formalize the control information with three vectors, the configuration request vector  $\vec{v} \in \mathbb{B}^K$ , the state vector  $\vec{st} \in \mathbb{B}^N$  and the control vector  $\vec{c} \in \mathbb{B}^M$ .  $K$  is the number of configuration-bits in the transmitted message,  $N$  is the number of bits of the current state of the configuration and  $M$  is the number of bits for controlling the switch. Intuitively,  $\vec{c}$  contains the information required to set-up a new path in a switch,  $\vec{st}$  contains the state information that needs to be stored in the DCMC, and  $\vec{c}$  is the output control vector matching the control signals of the switch. To formalize the functionality of the DCMC, we define the following three functions:

$$\begin{aligned} \Psi &: \mathbb{B}^N \times \mathbb{B}^K \rightarrow \mathbb{B} \text{ (compatibility function)} \\ \Xi &: \mathbb{B}^N \times \mathbb{B}^K \rightarrow \mathbb{B}^N \text{ (configuration merge function)} \\ \Phi &: \mathbb{B}^N \rightarrow \mathbb{B}^M \text{ (output-control function)} \end{aligned} \quad (1)$$

Basically,  $\Psi$  determines whether a new configuration request is compatible with the current state;  $\Xi$  combines a new configuration request with the current state and  $\Phi$  maps the current state into the control signals of the switch.

In order to illustrate the flexibility of our approach and the definition of  $\Psi$ ,  $\Xi$ , and  $\Phi$  in practical scenarios, the following sub-sections discuss the implementation of three different interconnect topologies using the DCM infrastructure. The examples (see fig. 2) have been selected to cover a wide range of implementation alternatives.

### A. Implementation Optical Network I

The first example is a parallel NoC consisting of an electrical and an optical layer. The architecture, similar to [14], uses an electrical mesh NoC and an optical torus. It is summarized in fig. 2a. The *optical Switches*, physically located in an extra optical layer, are managed by the *DCMs* which are located in the electrical layer. For our implementation, we use the following strategy: Normally there is no need to control every resonatorstructure of an optical switch to control the light from one direction to another. This results in a don't

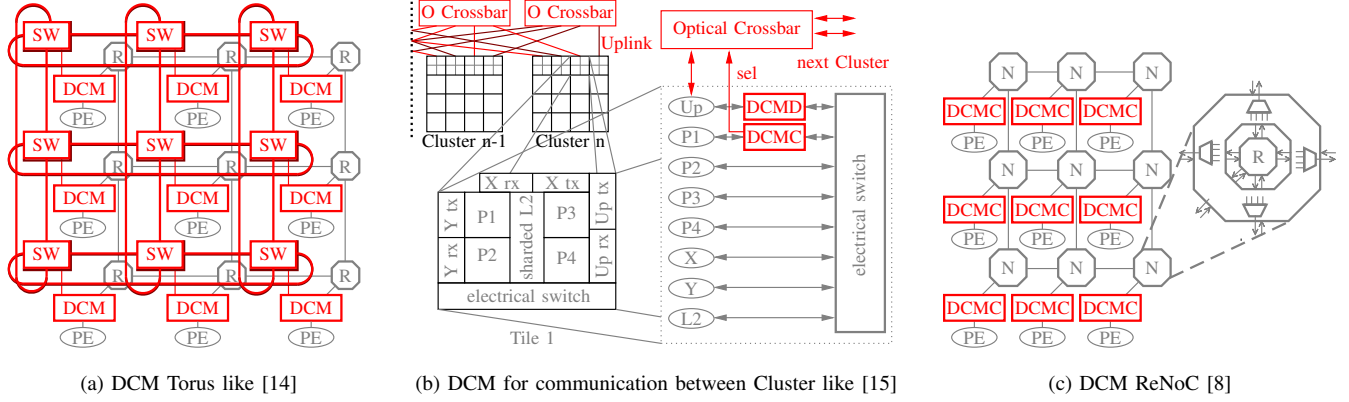


Figure 2: Implementation of different architectures with the Distributed Channel Management Unit (DCM)

care value for unused resonatorstructure. In our implementation we split these information in two vectors, one is containing the information for the requested state and the other contains the don't care information for the state vector.

The control signals can be formally defined as follows:

$$\vec{st} = \{\vec{st}_{sel}, \vec{st}_{dc}\}, \quad \vec{v} = \{\vec{v}_{sel}, \vec{v}_{dc}\}$$

$$\Psi(\vec{st}, \vec{v}) = \bigwedge_i \neg(\vec{v}_{sel}[i] \oplus \vec{st}_{sel}[i]) \vee (\vec{v}_{dc}[i] \wedge \vec{st}_{dc}[i])$$

where  $\vec{v}_{sel}[i]$  is the  $i$ -th bit of  $\vec{v}_{sel}$ . Each vector contains a selection part (i.e.,  $\vec{st}_{sel}$  and  $\vec{v}_{sel}$ ) and a don't care part (i.e.,  $\vec{st}_{dc}$  and  $\vec{v}_{dc}$ ). Simply, a bit is compatible if the selection bits are equal or they include a don't care.

Due to the several physical implementations for the optical switch (e.g., [16]), we do not detail the boolean function  $\Phi(\vec{st})$ . It can be trivially obtained for a particular switch.

### B. Implementation Optical Network II

The second example is a cluster-based system [15]. Each cluster contains a mesh of tiles consisting of four processing elements, a shared L2 cache, and inside-communication ports for the X- and Y-directions. The tile's components are connected through an electrical switch, while optical waveguides provide the inter-tile communication. Finally, up-link-ports provides the optical communication between those clusters using optical crossbars. Figure 2b summarizes the architecture.

The DCM infrastructure can be used to control the optical crossbars autonomously. Each tile requires a DCM, which can be associated to any *Processing Element*. In our implementation, the DCM is placed between the electrical switch and the *Processing Element 1* (P1). Therefore, the configuration requests are sent to P1. There is no need for changes in P1, since the DCM controller filters and reacts to the configuration message autonomously.

We structure the configuration request vector,  $\vec{v} \in \mathbb{B}^4$ , in two components of 2-bit:  $\vec{v}_{set}$  and  $\vec{v}_{sel}$ . They contain the input port (set value) and output port (selected value) which are requested respectively. The one-hot coded vector  $\vec{st}_{set}$  defines the used output port of the resonator structures of the optical crossbar. If the value is set to  $st_{set}[i] = 0$  the  $i$ -th output port is not configured and a input port can be assigned to that output port. The vector  $\vec{st}_{sel\_j} \in \mathbb{B}^n$  defines the selected input for any

output. It follows:

$$\vec{st} = \{\vec{st}_{sel\_0}, \vec{st}_{sel\_1}, \vec{st}_{sel\_2}, \vec{st}_{sel\_3}, \vec{st}_{set}\}$$

$$\Psi(\vec{st}, \vec{v}) = \neg \vec{st}_{set}[i] \text{ for } i = \vec{v}_{set}$$

$$\Xi(\vec{st}, \vec{v}) = \{\vec{nx}t_0, \vec{nx}t_1, \vec{nx}t_2, \vec{nx}t_3, \vec{nx}t_{set}\}$$

$$\vec{nx}t_j = \begin{cases} \vec{st}_{sel\_j} & \text{for } j \neq \vec{v}_{set} \\ \vec{v}_{sel} & \text{for } j = \vec{v}_{set} \end{cases}$$

$$\vec{nx}t_{set}[i] = \begin{cases} 1 & \text{for } i = \vec{v}_{set} \\ \vec{st}_{set}[i] & \text{for } i \neq \vec{v}_{set} \end{cases}$$

### C. Implementation Electrical Network

Even if our main goal is to control parallel interconnect architectures, it is also possible to address reconfigurable NoCs based on a single network. To demonstrate that potential, this section describes the management of ReNoC [8] with the DCM infrastructure.

In ReNoC (see fig. 2c), the routers of the network are wrapped by the topology switches. They allow the reservation of circuit-oriented paths bypassing the Routers. In order to control the topology switches, the DCMs are placed between the PE and the network node. Since ReNoC is build by a single network, there is no need for the DCM; only the DCM is required. In parallel networks, the channel is usually deconfigured by the DCM which has requested the path. It should be noticed that this is not safe in topologies with a single network: the deconfiguration messages may not be able to reach all the DCM modules, since the previously configured circuit-oriented path stills resources from packet oriented network. The deconfiguration messages can be sent by the DCM at the destination of the channel safely.

Let us consider now the control of the topology switches. As usual, U-turns are forbidden. Thus, the 5x5 switch can be implemented by five 4x1 muxers which are controlled by five 2-bit signals. Analogous to the previous example, we define the vectors  $\vec{v}$  and  $\vec{st}$ , as well as the functions  $\Phi$  and  $\Xi$  as follows:

$$\vec{v} = \{\vec{v}_{sel}, \vec{v}_{set}\}$$

$$\vec{st} = \{\vec{st}_{sel\_0}, \vec{st}_{sel\_1}, \vec{st}_{sel\_2}, \vec{st}_{sel\_3}, \vec{st}_{sel\_4}, \vec{st}_{set}\}$$

$\vec{v}_{sel}$  indicates the input port.  $\vec{v}_{set}$  describes the output port.  $\vec{st}_{sel}$  is the full sel-signal for the switch, which is generate from the partial inputs.  $\vec{st}_{set}$  describes in one-hot coding which output

## IV. CONCLUSION

has already been configured. This results in the following functions:

$$\begin{aligned} \Psi(\vec{st}, \vec{v}) &= -\vec{st}_{set}[i] \text{ when } i = \vec{v}_{set} \\ \Xi(\vec{st}, \vec{v}) &= \{\vec{nxt}_0, \vec{nxt}_1, \vec{nxt}_2, \vec{nxt}_3, \vec{nxt}_4, \vec{nxt}_{set}\} \end{aligned} \quad (5)$$

To calculate  $\vec{nxt}_j$  and  $\vec{nxt}_{et}[i]$  we are using the same functions like in section II-B.

## III. EXPERIMENTAL RESULTS

In this section, we analyze the area, power and performance results of the DCM at gate level. Our results are based on a 65nm low power technology. To the best of our knowledge, there are no other architectures with similar functionality and flexibility. Thus, a direct comparison with other designs is not possible. To put our result in context, we relate the overhead of our implementation with a conventional router. As reference, we take a 65nm 5x5 router which needs a area of  $0.031mm^2$  in the case of 32-bit flit size.

First of all, we analyze the overhead of the DCM for the first example discussed in section II. We analyze the area, frequency and power consumption depending on different parameters such as the maximum number of nodes which can be configured simultaneously and the bit width of the flits. Table I reports the results of the DCM.

Compared to the aforementioned router, the DCMC requires

Table I: Synthesis results: DCM

	nodes	flit [Bit]	freq. [MHz]	area [ $\mu m^2$ ]	total power [ $\mu W/GHz$ ]
DCMC	3	32 / 64	1785 / 1754	2530 / 2729	633 / 682
	7	32 / 64	1785 / 1724	3349 / 3511	802 / 817
	15	32 / 64	1724 / 1886	4804 / 4886	1093 / 1114
	width	flit [Bit]	freq. [MHz]	area [ $\mu m^2$ ]	total power [ $\mu W/GHz$ ]
DCMD	2	32 / 64	2940 / 2940	1616 / 2371	2572 / 3631
	4	32 / 64	2940 / 2940	1377 / 2499	2083 / 3625
	8	32 / 64	3030 / 3030	1274 / 2351	2291 / 3589

an overhead of around 8% - 16%, while the DCMD needs around 5% of overhead. Thus the complete overhead of the DCM is just 13% - 21% of the router area. Similar results can be drawn in terms of power. The frequency of the DCMC, above 1.3GHz, does not limit the conventional router.

Next, we discuss the overhead of the DCM depending on the different switch configuration functions. To facilitate the evaluation, table II reports just the area and power required for implementation of the functions  $\Xi$ ,  $\Phi$ , and  $\Psi$ . For the three examples the frequencies are not reported, since they do not affect the critical paths. As observed in the table, the function

Table II: Synthesis results: Configuration functions

function	Area [ $\mu m^2$ ]	total power [ $\mu W/GHz$ ]
Optical network I	1 052	686
Optical network II	366	663
Electrical network	363	725

used in the "Optical Network I" is the largest one; it requires  $1052\mu m^2$ , which is 3 times the area of the other functions. Thus, the overhead reported in table I represents the worst case of the three examples.

In this paper, we have demonstrated the suitability of a dedicated hardware module for the autonomous control of optical and electrical parallel reconfigurable interconnect architectures. The solution can be easily employed in modern many-tile SoCs, since it can be seamlessly integrated in an existing NoC, it is distributed, autonomous, and can be used at run-time. We have shown the applicability of the approach in real scenarios covering optical NoCs, optical clusters and reconfigurable electrical NoCs.

The analysis of the overhead incurred by the approach highlights the efficiency of the approach even in large architectures, with typical overhead values below 20%.

## REFERENCES

- [1] W. Dally *et al.*, "Route packets, not wires: on-chip interconnection networks," in *Design Automation Conference, 2001. Proceedings*, 2001, pp. 684–689.
- [2] C. Jackson *et al.*, "Skip-links: A dynamically reconfiguring topology for energy-efficient NoCs," in *System on Chip (SoC), 2010 International Symposium on*, 2010, pp. 49–54.
- [3] T. Bjerregaard *et al.*, "A survey of research and practices of network-on-chip," *ACM Comput. Surv.*, vol. 38, no. 1, Jun. 2006.
- [4] C.-H. O. Chen *et al.*, "Smart: A single-cycle reconfigurable NoC for SoC applications," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2013*, 2013, pp. 338–343.
- [5] C.-H. Chen *et al.*, "A low-swing crossbar and link generator for low-power networks-on-chip," in *Computer-Aided Design (ICCAD), 2011 IEEE/ACM International Conference on*, 2011, pp. 779–786.
- [6] X. Zheng *et al.*, "Silicon photonic wdm point-to-point network for multi-chip processor interconnects," in *Group IV Photonics, 2008 5th IEEE International Conference on*, 2008, pp. 380–382.
- [7] M. M. Kim *et al.*, "Polymorphic on-chip networks," in *Proceedings of the 35th Annual International Symposium on Computer Architecture*, ser. ISCA '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 101–112.
- [8] M. B. Stensgaard *et al.*, "Renoc: A network-on-chip architecture with reconfigurable topology," in *Networks-on-Chip, 2008. NoCS 2008. Second ACM/IEEE International Symposium on*, 2008, pp. 55–64.
- [9] M. Modarressi *et al.*, "Virtual point-to-point connections for noCs," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 29, no. 6, pp. 855–868, 2010.
- [10] —, "Application-aware topology reconfiguration for on-chip networks," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 19, no. 11, pp. 2010–2022, 2011.
- [11] A. Shacham *et al.*, "Maximizing gflops-per-watt:high-bandwidth, low power photonic on-chip networks," *Proc. Third Watson Conf. Interaction between Architecture, Circuits, and Compilers*, pp. 12–21, 2006.
- [12] D. Vantrease, "Optical tokens in many-core processors," Ph.D. dissertation, University of Wisconsin, 2010.
- [13] Y. Pan *et al.*, "Flexishare: Channel sharing for an energy efficient nanophotonic crossbar," in *High Performance Computer Architecture (HPCA), 2010 IEEE 16th International Symposium on*, 2010, pp. 1–12.
- [14] A. Shacham *et al.*, "On the design of a photonic network-on-chip," *Minimax Robust MIMO Radar Waveform Design*, 2007.
- [15] R. Morris *et al.*, "Exploring the design of 64- and 256-core power efficient nanophotonic interconnect," *Selected Topics in Quantum Electronics, IEEE Journal of*, vol. 16, no. 5, pp. 1386–1393, 2010.
- [16] H. Gu *et al.*, "A low-power low-cost optical router for optical networks-on-chip in multiprocessor systems-on-chip," in *VLSI, 2009. ISVLSI '09. IEEE Computer Society Annual Symposium on*, 2009, pp. 19–24.