

ReNoC: A Network-on-Chip Architecture with Reconfigurable Topology

Mikkel B. Stensgaard^{†‡} and Jens Sparsø[†]

Technical University of Denmark (DTU)[†]
Informatics and Mathematical Modelling (IMM)
2800 Kgs. Lyngby, Denmark
{mikkel.stensgaard,jsp}@imm.dtu.dk

Teklatech[‡]
Borgergade 20, 2
1300 Copenhagen K
mikkel@teklatech.com

Abstract

This paper presents a Network-on-Chip (NoC) architecture that enables the network topology to be reconfigured. The architecture thus enables a generalized System-on-Chip (SoC) platform in which the topology can be customized for the application that is currently running on the chip, including long links and direct links between IP-blocks. The configurability is inserted as a layer between routers and links, and the architecture can therefore be used in combination with existing NoC routers, making it a general architecture. The topology is configured using energy-efficient topology switches based on physical circuit-switching as found in FPGAs.

The paper presents the ReNoC (Reconfigurable NoC) architecture and evaluates its potential. The evaluation design shows a 56% decrease in power consumption compared to a static 2D mesh topology.

1 Introduction

Every new CMOS technology generation enables the design of larger and more complex systems on a single integrated circuit. The increasing complexity also means that design, test and production costs reach levels where large volumes must be produced for a chip to be feasible. The time it takes to get a new product to the market (time-to-market) thereby also increases. As envisioned in [1], this trend seems to make ASICs infeasible for the main bulk of applications - the development time will simply be too long and the cost too high.

For many applications a more general System-on-Chip (SoC) platform chip could be a viable solution. Such a SoC platform would contain many different IP-blocks including RAMs, CPUs, DSPs, IOs, FPGAs and other coarse and fine grained programmable IP-blocks. The communication is provided by means of a flexible communication infrastruc-

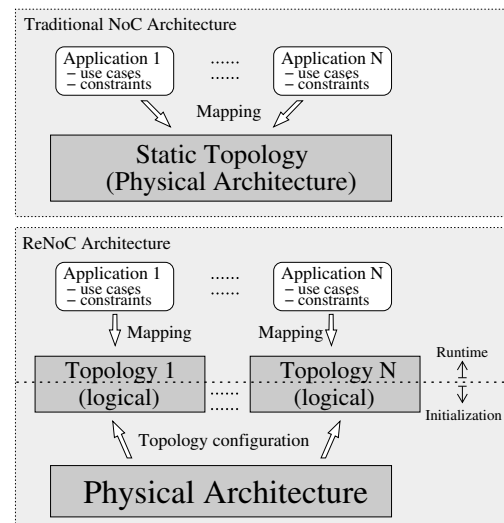


Figure 1. The ReNoC architecture enables a logical network topology to be configured by the application running on the physical SoC platform.

ture in the form of a Network-on-Chip (NoC) [2, 3]. This allows the same SoC platform to be used in a wide range of different applications and thereby increases the production volume.

As the same SoC platform is to be used for many different applications, the NoC must be able to support a wide range of bandwidth and Quality-of-Service (QoS) requirements. The requirements of the applications can be very different, and the NoC must therefore be very flexible. Currently, the only way to provide such flexibility is to employ a large packet-switched NoC with an over-engineered total bandwidth capacity. Such a NoC would take a significant part of the SoCs silicon area and only a fraction of its capacity is utilized by a given application.

In this paper we present the ReNoC (Reconfigurable

NoC) architecture that enables the network topology to be configured by the application running on the SoC. Figure 1 illustrates the difference between ReNoC and a traditional NoC architecture with static topology. In ReNoC, the NoC architecture as viewed by the application is actually a logical topology built on top of the real physical architecture. The logical topology is configured in an initialization phase before the application starts, denoted 'initialization' in the figure. This allows the topology to be configured based on the communication requirements of the application using energy-efficient topology switches. The topology switches are implemented using physical circuit-switching as found in FPGAs, to minimize the power consumption and area overhead. The motivation for inserting a configurable layer below existing NoC architectures is that physical circuit-switching is far more efficient (in terms of area, power and speed) than intelligent, complex packet-switching which therefore must be avoided when possible. The communication requirement for the application is therefore used to configure a logical topology that minimizes the amount of packet-switching.

The novelty of the ReNoC architecture is that it combines packet-switching and physical circuit-switching within the same NoC. It thereby includes the best of both worlds - flexibility from packet-switching and energy-efficiency from physical circuit-switching. This combination makes it possible to create application-specific topologies in a general NoC-based SoC platform. ReNoC can be used in combination with any packet-switched router, making it an extension to any traditional NoC architecture.

This paper presents and evaluates the ReNoC architecture, and is organized as follows. Section 2 states some basic terminology. Section 3 discusses related work and the observations behind the ReNoC architecture. The ReNoC architecture is presented in section 4, while section 5 presents an evaluation of the architecture. Section 6 contains implementation details for the evaluation before the results are presented and discussed in section 7. Section 8 concludes the paper and discusses future work.

2 Terminology

This section introduces key terms used in the paper.

Physical architecture is the actual physical layout of the NoC architecture as shown in the lower part of figure 1.

Logical topology is the topology that is configured on top of the physical architecture as shown in figure 1. This is the topology as it is viewed by the application.

Physical circuit-switching is used to denote a dedicated physical connection. Once the connection is set up, data can be transferred through the connection without any header information and no routing or arbitration is needed. This is not to be confused with *virtual* circuit-switching such as

Time-Division Multiplexing (TDM).

Router is used to denote any packet-switched router. The router might implement Quality-of-Service features such as TDM, and/or prioritization of data.

3 Motivation and Related Work

Most NoC research has focused on packet-switching, which is very flexible as it allows the same physical link to be shared by many different connections. Typically, general purpose topologies, such as the widely used 2D mesh, are employed. In these homogeneous topologies, a packet passes several routers, even when the communicating IP-blocks are localized close to each other. As future SoC platforms are expected to contain hundreds of IP-blocks the NoC needs to support an even larger number of connections and many connections span a large number of routers. This means that routers have to be faster to provide the required bandwidth and that more buffers are needed to support the large number of independent connections. As many applications have communication constraints, routers also get more complex in order to support different levels of Quality-of-Service, such as bandwidth and latency guarantees or prioritization of traffic. Routers therefore contribute a very large part of the total NoC power consumption. In [4], for example, each port in a 5x5 router uses 10 times more power than a 2 mm link in a 130 nm technology.

The key to obtaining lower area, latency and power consumption in the NoC, is to exploit knowledge of the application running on the SoC. In the context of homogeneous topologies, a few long links [5] can be inserted in the topology. This allows connections spanning many routers to bypass these routers using the long links, and it thereby decreases the amount of traffic in the intermediate routers. A more efficient option is to generate a heterogeneous, application-specific topology that matches the communication requirements for the application running on the SoC. This includes long links, and direct links between IP-blocks. Application-specific topologies have been shown to be very energy, area, and latency efficient compared to regular topologies, and have recently been receiving more attention [6, 4, 7, 8, 9]. Common for this research is that only static topologies are considered and the usage of application-specific topologies is therefore limited to application-specific chips designed for a single, or a number of very similar, applications.

In contrast to packet-switching, physical circuit-switching enables efficient, direct, physical connections to be set up between IP-blocks. As connections are dedicated, no buffering, arbitration and routing are needed and physical circuit-switching is therefore very energy, area, and latency efficient. In [10], the authors report that the delay is decreased by 85% and the energy by 70% by bypassing

FIFO buffers and synchronization logic. On the other hand, physical circuit-switching is inflexible as links can not be shared, and few articles have considered it in the context of NoC. An example of a physical circuit-switched NoC is [11], where connections can be set up directly between IP-blocks. The connections are configured using a separate packet-switched network which is also used for Best Effort (BE) traffic. The disadvantage is that the connections can not be shared, and that two separate networks exist.

The goal of the ReNoC architecture is to combine the best from the worlds of packet-switching and physical circuit-switching. Physical circuit-switching is used to set up energy-efficient end-to-end connections between IP-blocks and/or to form (long) links between routers, bypassing intermediate routers (which may or may not be powered down). Thereby, the topology can be changed by reconfiguring the circuit-switches (in the following, denoted topology switches due to their functionality) while packet-switching can be used to share the circuit-switched connections when the flexibility is needed. To our knowledge, no previous work has been done in combining packet-switching and physical circuit-switching such that the two methods co-exist in the same architecture.

A somewhat related idea is presented in [12, 13]. Here the authors argue - from an algorithm and parallel processing viewpoint - that the interconnect topology in a multi-processor platform should be reconfigurable. The authors suggest that the network be implemented in FPGA technology, but beyond this the papers offer limited information on implementation issues. Our work is different in that it originates in a desire to provide efficient interconnect in (heterogeneous) multi-core systems-on-chip, by combining packet-switching, circuit-switching and reconfigurability, and in that we present an implementation which is reconfigurable at a more coarse grained level. The latter results in a higher performance and a more cost-effective solution.

4 ReNoC Architecture

In this section we present the Reconfigurable NoC (ReNoC) architecture. First, the basic concepts of the architecture are explained through a simple example, before the generality of the architecture is discussed.

4.1 Basic Concepts

Figure 1 shows an overview of the ReNoC architecture. As introduced in section 1, it allows a logical topology to be configured on top of the real physical architecture. The topology configuration is transparent for the application, and the application experiences the topology as an ordinary static topology.

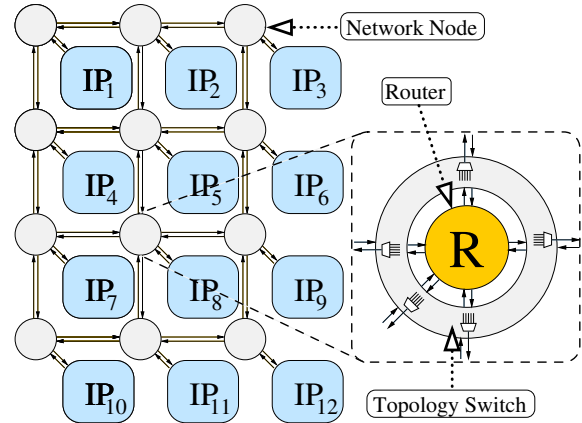


Figure 2. A simple physical architecture where network nodes are connected in a 2D mesh topology. A network node consists of a router that is wrapped by a topology switch.

The fundamental ideas of ReNoC are best explained through an example. For this, figure 2 shows a physical architecture consisting of *network nodes* connected by *links* in an 2D mesh topology. Each network node consists of a conventional NoC *router* which is wrapped by a *topology switch*. The topology switches are used to connect links and routers into a logical topology and they thereby allow different application-specific logical topologies to be configured on top of the same physical architecture. Figure 3 shows two examples of logical topologies that can be created by configuring the topology switches appropriately. As seen, it is possible to form long logical links connecting: (i) Any two IP-blocks, (ii) any two routers, and (iii) any IP-block and router. The physical distance between the IP-block/router does not matter, as long as a logical link can be established. Figure 3 illustrates that it is possible to configure logical topologies that are very different from the basic 2D mesh. If desired, it is also possible to configure a logical topology which is a 2D mesh.

In the logical topologies illustrated in figure 3, many of the routers and several of the links are unused. Clock gating may be used to eliminate the dynamic power consumption of these, and leakage power consumption can be reduced or eliminated completely by the use of power gating techniques. This is a key feature motivating the development of ReNoC. The "physical architecture" and the "logical topology 1" shown in figure 2 are part of the evaluation that is presented in section 5 and are discussed more at that point.

4.2 Topology Switches

As illustrated in figure 2, topology switches are inserted as a layer between the links and the routers, allowing links

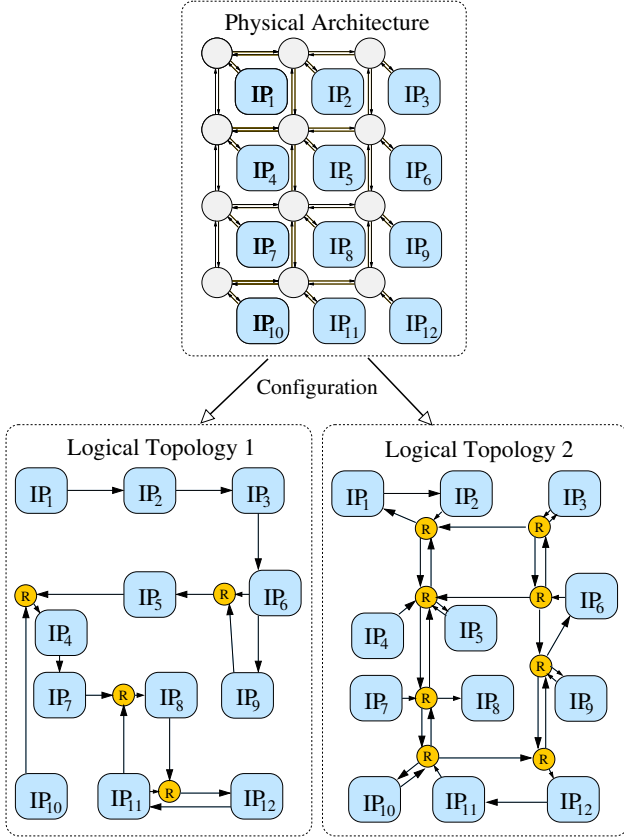


Figure 3. Two possible configurations of the physical architecture in figure 2. By configuring the topology switches appropriately, a wide range of different logical topologies can be created.

to be connected to a port on the router or directly to other links. Topology switches are meant to be configured infrequently such as once every time the chip is powered up, or when a new application is started. Fast reconfiguration is therefore not required, allowing an area and energy efficient implementation. In many respects, a topology switch is analogue to a switch-box in an FPGA and it can be implemented using the same techniques such as pass-gates, tri-state buffers or multiplexers.

The topology switch in figure 2 connects 4 links, an IP-block and a 5 port router. It must be able to connect links directly to each other, or to a single port on the router. If the "outside" ports connecting the topology switch to the links, including the link to the local IP-core, are denoted L_N , L_E , L_S , L_W and L_{IP} (north, east, south, west, and local IP) and if the "inside" ports connecting the topology switch to the router are labeled R_N , R_E , R_S , R_W and R_{IP} then the topology switch must support the following directional connections:

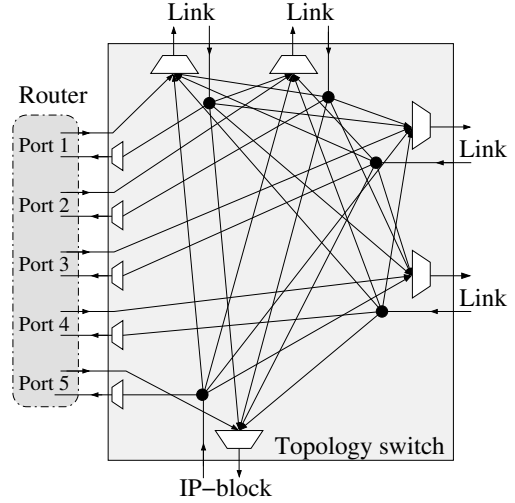


Figure 4. A multiplexer-based implementation of an asymmetric topology switch that can be used to connect 4 links, an IP-block and a 5-port router. Links can be connected directly to other links, the IP-Block or the corresponding port on the router.

- $L_i \rightarrow L_j$ where $i, j \in \{N, E, S, W, IP\}$ and $i \neq j$
i.e., incoming links can be connected directly to outgoing links - thereby bypassing the router altogether.
- $L_i \rightarrow R_i$ where $i \in \{N, E, S, W, IP\}$
i.e., incoming links can be connected to the corresponding ports on the router.
- $R_i \rightarrow L_i$ where $i \in \{N, E, S, W, IP\}$
i.e., ports on the router can be connected to outgoing links.

Figure 4 shows a possible multiplexer-based implementation of such an asymmetric topology switch.

If the links use low-swing signalling, it is also possible to implement the topology switches using low-swing switches as presented by Dally [14]. It should be noted, though, that low-swing links and low-swing topology switches cannot be implemented using standard cell libraries as it requires custom circuitry.

4.3 Routers

As there is a clear separation between topology switches and routers, the architecture is not restricted to a specific router. The only requirement is that the link width, including wires for flow-control, matches the ports on the router. In principle the communication protocol is defined by the routers and the topology switches and links act as passive

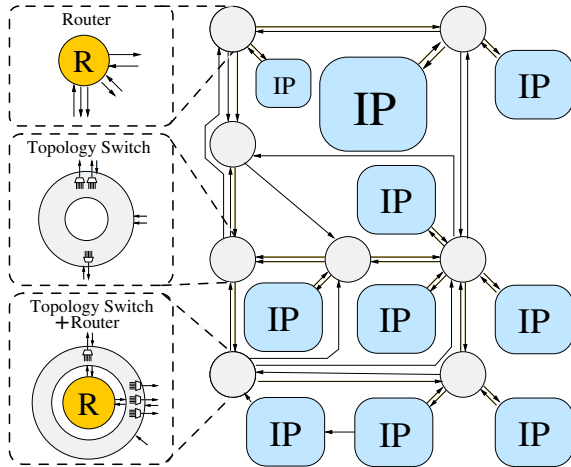


Figure 5. Example of a complex, heterogeneous, physical architecture. Network nodes can contain a router, a topology switch, or both. Several IP-blocks can be connected to the same network node, several links can exist between network nodes, and IP-blocks can be directly connected.

circuit-switched interconnects. This means that the architecture can be used in combination with any existing router. The routers can contain Virtual Channels (VC), Quality-of-Service (QoS) implementations such as TDM, queuing-buffers, and can be implemented using synchronous or asynchronous circuit techniques. The ReNoC concept can thus be used with existing routers including *Æthereal* [15], *Mango* [16], and *Xpipes* [17].

4.4 Generalization and Discussion

The physical architecture is not restricted to a simple 2D mesh as has been considered so far for illustration purposes. The physical architecture can be organized as any topology such as a tree, a mesh, some heterogeneous topology or hierarchical topology. To illustrate the full potential of ReNoC, figure 5 shows a heterogeneous physical architecture. As shown in the figure, network nodes can contain a router, a topology switch, or both. Links between network nodes can be both bi- and uni-directional, and several links can exist between two specific network nodes. Direct physical links can also exist between IP-blocks, and several IP-blocks can be connected to the same network node. Note, that routers do not need to have the same number of ports as the number of links that is connected to the topology switch. The topology switch enables the same router port to be connected to different links depending on the configuration. Hence, the router port becomes a sharable resource.

In figure 5, some of the topology switches contains a large number of link ports. If these topology switches were to allow links to be connected in all possible combinations, they would consume a large amount of energy and area. Instead, we envision the large topology switches to be highly asymmetric such that each incoming port can only be connected to a subset of the outgoing ports.

The logical topology must be configured such that the latency of the slowest *logical* link does not exceed the clock period. If needed, it is also possible to pipeline the *logical* links by inserting pipeline registers in the topology switches or on the physical links. Depending on the physical link length and the operating frequency of the router, it might be enough to have pipeline registers in a subset of the topology switches. As in any design, clock-skew is also an issue to consider but this issue is beyond the scope of this paper.

In our current work the logical topology is configured at initialization time, and the different use-cases of an application will run on this logical topology. This is the scenario illustrated in figure 1. More elaborate architectures and scenarios are possible which allow run-time configuration of an individual logical topology for each use-case, but this implies significant added complexity and is beyond the scope of this paper.

5 Evaluation

The purpose of the evaluation is to demonstrate the potential of the ReNoC architecture and estimate the overhead of the topology switches. This is done by mapping an application onto a NoC architecture with a static 2D mesh topology as well as a simple ReNoC architecture in two different topology configurations. The physical architecture is chosen such that it, besides being configured as an application-specific topology, can be configured as an ordinary 2D mesh. This illustrates a ReNoC architecture that is a general platform where all IP-blocks are able to communicate but which can also be configured in an application-specific topology.

In the following we describe the application, the physical architecture, and the router choice in more detail. The implementation details are presented in section 6 and the results in section 7.

5.1 Benchmark Application and Network Topologies

As benchmark we use the Video Object Decoder (VOPD) application that is presented in [6] where an application specific, hard-wired topology is compared to a 2D mesh topology. Figure 6(a) illustrates the Task Graph of the VOPD. Each node in the graph represents a task while the edges denote the average required bandwidth between tasks

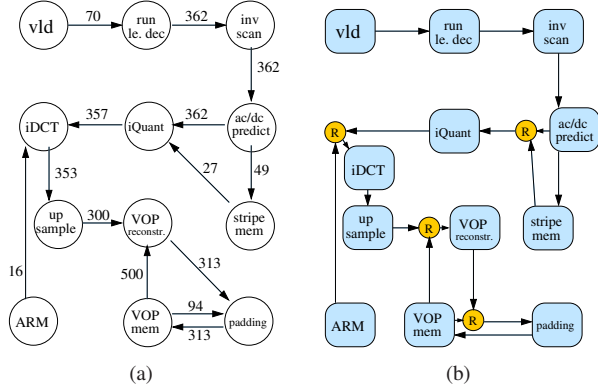


Figure 6. (a) Task graph of the VOPD application. Edges denote bandwidth in Mbit/s. (b) ReNoC architecture where a logical application-specific topology is configured for the VOPD application.

in Mbit/s. The placement of the tasks in the graph represents the mapping onto the architectures used in the evaluation. The following architectures are used for comparison:

Static mesh: A static 2D mesh topology used as reference. It is similar to the topology shown in figure 2 where each network node contains a statically connected router.

ReNoC mesh: The ReNoC architecture that is configured to provide a 2D mesh logical topology similar to *Static mesh*. This configuration is used to characterize the overhead of the topology switches.

ReNoC specific: The ReNoC architecture that is configured with the application specific topology shown in figure 6(b).

5.2 Physical ReNoC Architecture

The physical architecture used in the evaluation is shown in figure 2 and was introduced in section 4.1. Each network node consists of a router wrapped by a topology switch. There are 3 different types of network nodes that differ in the number of connected links and router size. The two network nodes in the middle of the mesh contains 5x5 routers and connect 4 links. The network nodes in the sides, bottom, and top contains 4x4 routers and connect 3 links, while the network nodes in the corners contains 3x3 routers and connect 2 links.

As explained in section 4.2, topology switches are constructed such that they can connect links in all possible combinations as well as links directly to corresponding ports on the router.

We assume that each physical link has a length of 1 mm which allows the IP-block to be approximately 1 mm^2 .

5.3 Router Choice

The benefits of ReNoC depends on the relative energy consumption of topology switches and routers. If the routers consume much more energy than the topology switches, ReNoC will have a clear advantage as an application-specific topology decreases the amount of traffic in the routers. If the topology switches, on the other hand, consume as much energy as the routers, the overhead of the topology switches will dominate the total power dissipation making ReNoC infeasible.

In order to obtain a reasonably fair evaluation, which does not overestimate the benefits of the ReNoC concept, it is important to choose a router whose bandwidth and features does not significantly exceed the requirements of the application. A router with advanced QoS features, or a heavily pipelined router operating at 900 MHz, use much more power than a simple low-frequency router. High operating frequency also means a large power consumption in clocked elements even when no flits are passing through the router. The level of clock-gating is also a very important factor as routers provide bandwidth in Gbit/s and therefore can be idle for large periods of time.

To make the comparison fair, we have implemented a simple, low-power, packet-switched router and topology switch. The router architecture is presented in detail in section 6 and is a standard architecture as presented by Dally [18]. The router is operating at 100 MHz, which gives a maximum bandwidth of 2.4 Gbit/s per link if flits contain 32 bit data and a packet is made of 4 flits with the first flit being a dedicated header flit. This bandwidth is more than enough for the VOPD application with the largest needed average bandwidth between two tasks being 500 Mbit/s.

6 Implementation

The evaluation is conducted using area and energy models for routers, topology switches and links. Routers and topology switches have been synthesized and power characterized using commercial synthesis and power characterization tools using estimated wire-load models while link characterization is based on figures from existing literature. All figures are based on low-leakage cells from a commercial 90 nm standard cell library, using a 1 V supply voltage at nominal parameters. The designs are implemented for low-power applications operating at 100 MHz.

Table 1 summarizes the area and energy consumption of the models. Four figures are stated for each model: (i) Area is simply the area reported by the synthesis tool, (ii) energy/packet is the average energy consumed when sending

Table 1. Characterization of the routers, topology switches and link.

Module	Area (mm^2)	Energy per packet (pJ)	Leakage (μW)	Idle power (μW)
Link, 1mm	-	21	-	-
5x5 Router	0.061	32	8.6	136
Topology Switch	0.007	0.6/0.8	0.7	-
4x4 Router	0.047	31	6.7	109
Topology Switch	0.005	0.6/1.1	0.6	-
3x3 Router	0.032	30	4.7	82
Topology Switch	0.003	0.6/1.3	0.3	-

Table 2. Characterization of the modules in the 5 port router with 2 virtual channel buffers in each input port.

Module	Area (mm^2)	Energy per packet (pJ)	Leakage (μW)	Idle power (μW)
Input Port	8900	21.1	1.2	18.8
Virtual Channel	4300	16.4	0.6	8.7
Output Port	1350	5.7	0.15	6.3
5x5 Switch	3800	2.6	0.4	-
VC Allocator	5100	1.6	0.8	11.3
Switch Allocator	900	0.8	0.13	-

a packet based on random data, (iii) leakage is the leakage power consumption, and (iv) idle power is the dynamic power that is always consumed - independent of the use. Idle power accounts for clocking of clock-gates and registers that are not clock-gated. A packet contains a 96 bits of data.

In the following routers, topology switches and links are discussed in more detail.

6.1 Routers

Figure 7 shows an overview of our router architecture. The router is a conventional source-routed, input-buffered, packet-switched router with Virtual Channels (VCs) as presented by Dally [18, chapter 16]. Input ports contain 2 Virtual Channel (VC) buffers, each capable of holding 4 flits which is implemented using a small register file. Besides the registers in the VCs, there is a single register in the output ports. A packet contains a dedicated header flit, followed by 3 payload flits. The flit size is 34 bits - 32 bits for

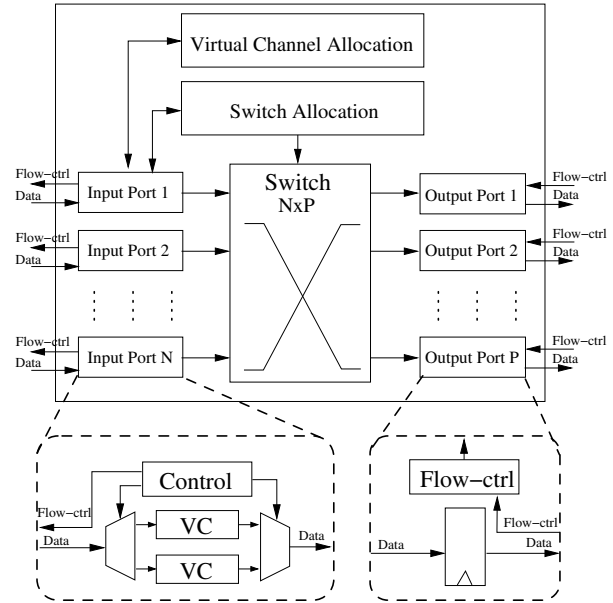


Figure 7. Overview of the router architecture used in the evaluation.

data/header, 1 bit to indicate the virtual channel, and 1 bit to indicate the last flit in a packet. Besides these 34 bits, the link contains a single bit to indicate the presence of a flit as well as 2 bits for flow-control which is credit-based. Hence, the total link width is 37 bits, including flow-control. The router is synthesized to operate at 100 MHz and is single-cycled - meaning that it can perform virtual channel allocation, switch allocation and switch traversal in a single cycle. As there is a register in the output ports, this means that it ideally takes one cycle to traverse the router and one cycle to traverse the link. The router can actually be synthesized to operate at 400 MHz (with 5 input and 5 output ports), but this speed is not needed and leads to a larger router that consumes more energy.

The router is clock-gated at buffer-level, such that it only consumes a small amount of power when it is sparsely used.

Table 2 lists the characterization of the different submodules of a router with 5 input and 5 output ports. The energy/packet is based on simulations using random data both for address and payload data at 20% maximum bandwidth utilization.

6.2 Topology Switches

The topology switches are implemented using multiplexers as illustrated in figure 4. All outgoing ports can be disabled to avoid toggles to be propagated to the corresponding port. Besides multiplexers, each topology switch contains registers to control the multiplexers. In this paper we do not

consider how these configuration registers are written.

The topology switches are synthesized for low energy and area, and as for the router, the energy/packet is based on simulations using random data both for address and payload data at 20% maximum bandwidth utilization. As the topology switches are asymmetric, table 1 states two energy figures for each topology switch. The first figure is the energy consumed when sending a packet to a router port while the second figure is the energy consumed when sending a packet to a link or IP-block.

The worst-case latency of the largest topology switch is 550 ps.

6.3 Links

Energy consumption in the links is based on the SPICE simulated figure presented in [19]: 0.36 pJ/transition/mm at a supply voltage of 1.2 V. Scaling to 1 V this becomes 0.25 pJ/transition/mm, which is the figure used in our evaluation.

The energy/packet is estimated by assuming 50% switching activity on the 34 bits in the flit, 2 transitions on the request wire and 2 transitions on the flow-control. As a packet consists of 4 flits and links are assumed to 1 mm, this sums up to 21 pJ/packet containing 96 bits.

A pessimistic estimate of the latency of a 1 mm link is 120 ps. This is based on an lumped delay model added 50% driver overhead using an estimated wire capacitance and resistance of 0.2 pF/mm and 0.4 K Ω /mm.

7 Results and Discussion

Table 3 shows the area and power consumption of the 3 architectures that were presented in section 5.1. As seen in figure 6(b), only 25% of the routers are used in *ReNoC specific*, and the remaining routers are assumed to be power-gated to decrease the leakage- and idle power consumption.

The area overhead of the ReNoC architecture is found by comparing the area of *Static mesh* with the area of *ReNoC mesh*. The area increases with 10% which shows that the area overhead of the topology switches is small.

The overhead in terms of power consumption is evaluated by comparing *Static mesh* with *ReNoC mesh*, as they both have a 2D mesh logical topology. The topology switches increase the power consumption by 3%, indicating that the overhead in terms of power consumption is minimal.

When an application-specific topology is configured in *ReNoC specific*, the power consumption is decreased by 56% compared to *Static mesh*, including the power consumption in the topology switches. The topology switches only use 5% of the power in *ReNoC specific*.

An important observation that illustrates the potential of the ReNoC architecture, is that only a few ports are used on

the routers in *ReNoC specific*. The area and power consumption can be decreased further by using routers with fewer ports - for example 3 ports. Even though it will no longer be possible to configure a 2D mesh topology, it will still be possible to configure a wide range of different topologies. Thereby the area and power consumption will approach that of a static application-specific topology.

ReNoC is evaluated using a low-power router using an operating frequency of 100 MHz. As the latency of a link and topology switch is 120 and 550ps, respectively, it is possible to traverse approximately 14 links and topology switches within a single cycle - assuming near zero clock skew. ReNoC can also be used in high performance designs. As discussed in section 5.3, the area and power overhead will be relatively smaller if larger, more complex, routers are used and/or the router is clocked at higher frequencies. If the routers are clocked at a higher frequency, it might be necessary to synthesize the topology switches with focus on latency instead of energy. Pipeline registers can also be inserted in the topology switches or physical links as discussed in section 4.4.

8 Conclusion and Future Work

In this paper we have presented the ReNoC architecture that enables the network topology to be reconfigured using energy-efficient topology switches. The architecture was evaluated by mapping an application to a static 2D mesh topology as well as a ReNoC architecture in two different topology configurations. The power consumption was decreased by 56% when configuring an application-specific topology, compared to the static 2D mesh topology. The topology switches increased the area of the NoC architecture with 10%, and only contributed with 5% of the power consumption in the application-specific topology. The evaluation shows that the ReNoC architecture enables application-specific topologies to be configured with little overhead and indicates that the architecture has great potential for future SoC platforms.

Future research include exploration of physical architectures for multiple applications as well as automatic generation of homogeneous and heterogeneous physical architectures. More research also has to be done on efficient implementation and configuration of the topology switches.

References

- [1] P. Magarshack and P. G. Paulin, "System-on-chip beyond the nanometer wall," in *DAC '03: Proceedings of the 40th conference on Design automation*. New York, NY, USA: ACM Press, 2003, pp. 419–424.

Table 3. Area and power consumption of the VOPD application in the three architectures used for evaluation.

Architecture	Area (mm^2)			Power consumption (mW)					
	Routers	Topology switches	Total	Routers	Topology switches	Links	Leakage Power Power	Idle Power	Total
<i>Static mesh</i>	0.53	-	0.53	2.39	-	0.84	0.08	1.25	4.56
<i>ReNoC mesh</i>	0.53	0.05	0.58	2.39	0.12	0.84	0.08	1.25	4.69
<i>ReNoC specific</i>	0.53	0.05	0.58	0.65	0.09	0.84	0.03	0.41	2.02

- [2] W. J. Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks," in *Design Automation Conference*, Jun. 2001, pp. 684–689.
- [3] G. de Micheli and L. Benini, "Networks on chip: A new paradigm for systems on chip design," p. 418, 2002.
- [4] S. Murali, P. Meloni, F. Angiolini, D. Atienza, S. Carta, L. Benini, G. D. Micheli, and L. Raffo, "Design of application-specific networks on chips with floorplan information," 2006.
- [5] U. Y. Ogras and R. Marculescu, "'it's a small world after all': NoC performance optimization via long link insertion," *IEEE Trans. on Very Large Scale Integration Systems, Special Section on Hardware/Software Codesign and System Synthesis*, vol. 14, no. 7, Jul. 2006.
- [6] A. Jalabert, S. Murali, L. Benini, and G. D. Micheli, "xpipesCompiler: A tool for instantiating application specific Networks on Chip," in *Design, Automation and Test in Europe (DATE)*, Paris, France, Feb. 2004.
- [7] K. Srinivasan, K. S. Chatha, and G. Konjevod, "An automated technique for topology and route generation of application specific on-chip interconnection networks," in *ICCAD '05: Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 231–237.
- [8] L. Benini, "Application specific noc design," in *DATE '06: Proceedings of the conference on Design, automation and test in Europe*. 3001 Leuven, Belgium, Belgium: European Design and Automation Association, 2006, pp. 491–495.
- [9] J. Xu, W. Wolf, J. Henkel, and S. Chakradhar, "A design methodology for application-specific networks-on-chip," *Trans. on Embedded Computing Sys.*, vol. 5, no. 2, pp. 263–280, 2006.
- [10] S.-J. Lee, K. Lee, and H.-J. Yoo, "Analysis and implementation of practical, cost-effective networks on chips," *IEEE Des. Test*, vol. 22, no. 5, pp. 422–433, 2005.
- [11] P. T. Wolkotte, G. J. M. Smit, G. K. Rauwerda, and L. T. Smit, "An energy-efficient reconfigurable circuit-switched network-on-chip," in *IPDPS '05: Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05) - Workshop 3*. Washington, DC, USA: IEEE Computer Society, 2005, p. 155.1.
- [12] S. Vassiliadis and I. Sourdis, "Flux networks: Interconnects on demand," in *Int. Conf. on Embedded Computer Systems: Architectures, Modeling and Simulation (IC-SAMOS)*, July 2006, pp. 160–167.
- [13] V. S and S. I, "Reconfigurable fabric interconnects," in *Int. Symposium on System-on-Chip (SoC)*, November 2006, pp. 41–44.
- [14] W. J. Dally, "'enabling technology for on-chip interconnection networks' keynote presentation at 'the 1st acm/ieee international symposium on networks-on-chip', 2007." [Online]. Available: http://www.nocsymposium.org/keynote1/dally_nocs07.ppt
- [15] K. Goossens, J. Dielissen, and A. Rădulescu, "The Æthereal network on chip: Concepts, architectures, and implementations," *IEEE Design and Test of Computers*, vol. 22, no. 5, pp. 414–421, Sept-Oct 2005.
- [16] T. Bjerregaard and J. Sparsø, "A router architecture for connection-oriented service guarantees in the mango clockless network-on-chip," in *Proc. Design Automation and Test in Europe (DATE'05), ACM sigda, 2005*, 2005, pp. 1226–1231.
- [17] D. Bertozzi and L. Benini, "Xpipes: a network-on-chip architecture for gigascale systems-on-chip," *Circuits and Systems Magazine, IEEE*, vol. 4, no. 2, pp. 1101–1107, 2004.

- [18] D. W. James and T. B. Patrick, *Principles and practices of interconnection networks*. Morgan Kaufmann Publishers, Inc., 2004.
- [19] A. Banerjee, R. Mullins, and S. Moore, "A power and energy exploration of network-on-chip architectures," May 2007.