ENERGY-EFFICIENT NOC FOR BEST-EFFORT COMMUNICATION

Pascal T. Wolkotte, Gerard J.M. Smit

University of Twente, Department of EEMCS
P.O. Box 217, 7500 AE Enschede
The Netherlands
E-mail: P.T.Wolkotte@utwente.nl

Jens E. Becker

University of Karlsruhe, Institute for
Information Processing Technology (ITIV)
D76128 Karlsruhe, Germany
E-mail: Beckerj@itiv.uni-karlsruhe.de

ABSTRACT

A Network-on-Chip (NoC) is an energy-efficient on-chip communication architecture for Multi-Processor System-on-Chip (MPSoC) architectures. In an earlier paper we proposed a energy-efficient reconfigurable circuit-switched NoC to reduce the energy consumption compared to a packet-switched NoC. In this paper we investigate a chordal slotted ring and a bus architecture that can be used to handle the best-effort traffic in the system and configure the circuit-switched network. Both architectures are compared on their latency behavior and power consumption. At the same clock frequency, the chordal ring has the major benefit of a lower latency and higher throughput. But the bus has a lower overall power consumption at the same frequency. However, if we tune the frequency of the network to meet the throughput requirements of control network, we see that the ring consumes less energy per transported bit.

1. INTRODUCTION

In the Smart chipS for Smart Surroundings (4S) project [1] we propose a heterogeneous Multi-Processor System-on-Chip (MPSoC) architecture with run-time software and tools. The MPSoC architecture contains a heterogeneous set of processing tiles interconnected by a Network-on-Chip (NoC). The run-time software determines a near optimal mapping of applications to the heterogeneous architecture. The architecture including the run-time software can replace inflexible ASICs for future ambient systems.

These ambient systems are typically battery powered and have to support a wide range of applications so they have to be flexible as well as energy-efficient. To map ambient applications on a parallel architecture like a MPSoC we assume the application is represented as communicating parallel processes. One possible representation is a Kahn based process graph model [2], which is a directed graph with nodes representing sequential processes and edges representing FIFO communication between processes.

To reduce the energy consumption of the overall application we aim to map the processes on the processing tile that can execute it most efficiently. Due to the mapping of processes to processing tiles on the MPSoC communication is introduced.

Traditionally communication between processing tiles is based on a shared bus. But for larger MPSoC with many processing tiles it is expected that the bus will become a bottleneck from both a performance, scalability and energy point of view [3]. Therefore, we propose a multi-hop NoC, where the network consists of a set of routers interconnected by links. In this paper we assume a regular two dimensional mesh topology of the routers. Every router is connected with its four neighboring routers via bidirectional point-to-point links and with a single processing tile via the tile interface.

1.1. NoC Architecture

The MPSoC architecture is organized as a centralized system: one processing node, called Central Coordination Node (CCN), performs system coordination functions.

The main task of the CCN is to manage the system resources. It performs run-time mapping of the newly arrived applications to suitable processing tiles and inter-process communication to network links [4]. The CCN also tries to satisfy Quality of Service (QoS) requirements, to optimize the resources usage and to minimize the energy consumption. Using the centralized approach we are able to define a NoC with guarantees for the QoS constraints for the communication network.

For the NoC we defined a network that can both handle guaranteed throughput (GT) traffic and best-effort (BE) traffic. The guaranteed throughput traffic is defined as data streams that have a guaranteed throughput and a bounded latency. The best-effort traffic is defined as traffic where neither throughput nor latency is guaranteed. The BE traf-

This research is conducted within the Smart Chips for Smart Surroundings project (IST-001908) supported by the Sixth Framework Programme of the European Community. The Software described in this document is furnished under a license from Synopsys (Northern Europe) Limited. Synopsys and Synopsys product names described herein are trademarks of Synopsys, Inc.

0-7803-9362-7/05/$20.00 ©2005 IEEE

fic handles traffic like configuration data, interrupts, status messages etc.

The defined network is a reconfigurable circuit switched network [5]. The CCN configures this network and connects processing tiles via a part of the physical link, where the bandwidth and latency is guaranteed and independent of other traffic in the network. The best-effort traffic is not directly supported by the circuit switched network, but is handled via a separate control network, which is subject of this paper.

From our experience with applications for base-band processing in digital wireless communications, which includes HiperLAN/2, UMTS, Bluetooth and digital radio [6], [7], [8] we observe for the NoC requirements that:

- These applications can be represented as a Kahn based process graph.
- The majority of data traffic between the processes has periodic behavior and requires real-time guarantees.
- The application and the resulting communication will remain active for seconds as an end-user will use an application for seconds or minutes.

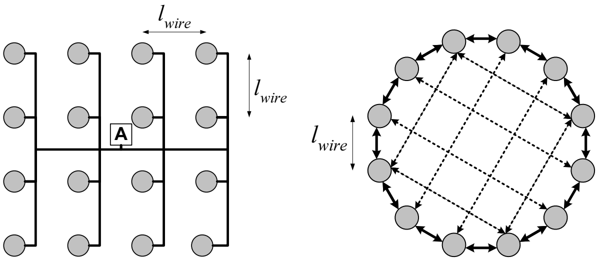
In [5] these application characteristics are used to compare a packet-switched wormhole router [9] with the reconfigurable circuit-switched router on their energy consumption and time performance. It turns out that the circuit switched router matches very well the streaming behavior of the applications. The major advantage of the circuit switched router is 3.5 times lower power consumption compared to the packet-switched solution. The disadvantage of the circuit switched router is that it has to be configured by an external control network. Solutions for this control network are analyzed in this paper.

The rest of this paper is organized as follows. In section 2 the two architecture options of the best-effort control network are presented. To analyze the performance of the networks we describe a power model for wires in section 3. This model is used in section 4 for the performance analysis of the control network. Section 5 concludes the paper.

2. NETWORK-ON-CHIP ARCHITECTURE

As described in the introduction, we have a hardware architecture containing a set of heterogeneous processing tiles interconnected by a Network-on-Chip. For the moment the tiles and NoC are synchronized by the same clock.

Fundamentally, sharing resources and giving guarantees are conflicting, and efficiently combining guaranteed throughput (GT) traffic with best-effort (BE) traffic is hard [10]. By using dedicated techniques for both types of traffic we try to reduce the total area and power consumption. In this paper we concentrate on the architecture for communication of best-effort traffic. This best-effort traffic also includes the configuration packets that are required for (re)configuring the circuit switched network.



(a) Bus layout (b) Chordal ring layout

Fig. 1. Configuration network layout

2.1. Control Network

For the control network we evaluate two different communication architectures (see Figure 1) that connect all the processing tiles, where l_{wire} is the distance between two neighboring processing tiles[1] and N is the number of processing tiles in the system. The first architecture is a bus architecture where every tile has both a master and a slave port. The second architecture is a chordal slotted ring architecture. Both architectures can support only single packet transfers and no burst transfers. Burst transfers are handled by the guaranteed throughput network.

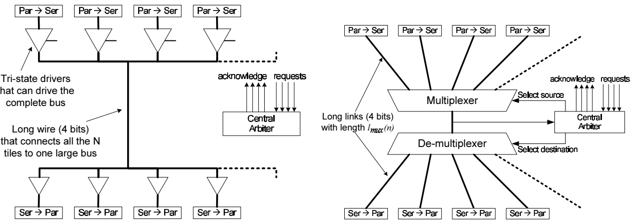
We assume that the number of wires that can be used for the total communication network (GT and BE) is limited. Therefore, we serialize the data for the BE network in the same manner as the GT traffic. A second reason for serializing the data is that the required performance for the BE part of the traffic is low (does not require a very high bandwidth) [5]. For both control networks we serialize a 24 bits data-item (8 bits control/address, 16 bits data) into a packet of l_{packet} flits. In this paper we transport 4 bits in parallel, which makes l_{packet} equal to six flits.

2.1.1. Bus

The first architecture is a bus architecture. An example of a bus architecture is given in Figure 1(a). The bus has a centralized arbiter (A) that controls the access to the bus. Every tile in the bus can act as a master and as a slave. If a tile wants to access the bus it sends a request to the central arbiter. The arbiter decides with a round-robin arbitration schedule which requesting tile may use the bus for a single bus-transfer. The tile, which request is granted, can use the bus for transferring one single packet of 24 bits.

For the physical bus architecture we evaluated two types of bus layouts that are depicted in Figure 2.

[1] In this paper we assume that all tiles have an equal size. Similar results can be obtained with tiles that do not have an equal size


Par → Ser
Tri-state drivers that can drive the complete bus
Long wire (4 bits) that connects all the N tiles to one large bus
acknowledge
requests
Central Arbiter
Ser → Par
Multiplexer
De-multiplexer
Long links (4 bits) with length $l_{mux}(n)$
Select source
Select destination

(a) Tristate bus (b) Multiplexer bus

Fig. 2. Bus options

The tristate bus connects all the tiles via bidirectional wires that directly connect all the processing tiles. The tiles connect their outputs via a tristate driver to the bus. The advantage of this bus-layout is that the number of wires is minimal. But the load noticed by the drivers of the bus is high, due to the many receivers and large active wire length (that spans among all the tiles). The number of active receivers in the bus is equal to the number of tiles (N). The total length of the wires required to reach all the receivers is equal to the total length of wires in the bus. This is equal to $(N-1)\cdot l_{wire}$ in the regular mesh layout of Figure 1(a).

The multiplexer bus connects all the tiles outputs with point-to-point wires to a central multiplexer near the arbiter (see Figure 2(b)). The arbiter selects the granted tile and analyzes the destination address. Using this address, the arbiter will forward the data to the destination tiles by selecting the correct output of the central de-multiplexer. Compared to the tristate bus the data will only be driving wires that are needed between the source and destination (i.e. between source and multiplexer, source and arbiter, de-multiplexer and destination). This will decrease the amount of wire segments that are active, but it will increase the total amount of wires on the MPSoC.

2.1.2. *Chordal slotted ring*

The second architecture is a chordal slotted ring architecture that has a perimeter equal to N hops. An example of a chordal ring is given in Figure 1(b). The ring is organized as two uni-directional rings ($ring_0$ and $ring_1$) that connect all the tiles in a large ring. On the ring there are one or more dynamically created slots that each can transport a data packet of 24 bits. At every tile the slot is stored, the destination address is analyzed and forwarded to the next tile if the destination is not reached. Due to the two uni-directional rings the maximum number of hops between two tiles is $N/2$. Because of the store-and-forward mechanism the ring can handle $2\cdot N$ slots concurrently.

During operation of the total SoC not all tiles may be required. To reduce the number of hops extra short-cuts (indicated by dashed lines in Figure 1(b) between tiles in the ring are possible. At run-time we can re-configure the ring and use the short-cuts instead of the standard ring route. The re-configuration of the ring is handled by the CCN. This will reduce the ring-perimeter and therefore the maximum number of hops. Furthermore, in-active parts of the ring can be switched-off to save energy[2].

3. PERFORMANCE ANALYSIS WIRE

In section 4 we analyze the performance of the control network. The performance of the logic can be determined by modelling the design in VHDL and using Synopsys Power Compiler [11] for power estimation in .13 technology. This tool does not include the long wires between the logic blocks. For the long wires between the logic blocks we use an analytical model of a wire.

For the power figures of a wire we include the drivers and repeaters that are required in a link between two routing structures. In [12] the power of a link between two routers is given by:

$$P_{link} = \left(P_{drivers} + P_{repeaters} + P_{wire}\right)\cdot N_{wires} \qquad (1)$$

where N_{wires} is equal to the number of parallel wires of the link, which is equal to 4 in this paper. Each power factor can be defined as the sum of dynamic and leakage power. In this paper we only focus on the dynamic energy consumption of the links as leakage power in .13 technology is minimal [13]. In this paper we do not include repeaters for performance-increase of the wires, thus $P_{repeaters}$ is set to zero. Via simulation we discovered that for frequencies is less than 100 MHz the repeaters can be safely ignored.

In [13] it is shown that the dynamic power consumption of a link (wire including the driver) is equal to:

$$P_{link_{dyn}} = \left\{\alpha\left(s\left(c_p + c_0\right) + c\cdot l_{wire}\right)V_{DD}^2 f_{clk}\right\}\cdot N_{wires} \qquad (2)$$

Where α is equal to the switching factor (or activity factor), l_{wire} the length of the wire in mm and c_0, c_p, c and s are determined by the process, wire pitch and wire dimensions. We use $c_0 = 1.7[fF], c_p = 3.5[fF], c = 240[fF/mm]$ and $s = 151$, which are the values given for $0.13\mu m$ technology by [13]. For the voltage we use a $V_{DD} = 1V$, which is also used for the power estimation of the logic blocks.

The activity factor is data (the amount of bit-toggles) and load dependent. In a typical data-stream we have a possibility of 50% possibility for a data change from 0 to 1 or visa versa. Therefore, for typical data-streams the activity factor is then only related to the load on the link $\alpha = 0.5\cdot L_{link}$, where L_{link} is the average load of the link, with $0 \le L_{link} \le 1$. With $\alpha = 0.5\cdot L_{link}$, $f_{clk} = 1MHz$ and $V_{DD} = 1V$ we get the power consumption in $\mu W/MHz$:

$$\begin{aligned} P_{link_{dyn}} &= 0.5\cdot\left(s(c_0 + c_s) + c\cdot l_{wire}\right)\cdot N_{wires}\cdot L_{link} \\ &= \left(0.39 + 0.12\cdot l_{wire}\right)\cdot N_{wires}\cdot L_{link} \qquad (3) \end{aligned}$$

[2] Also incorrect behaving tiles can be switched off, so the ring can avoid one or more defect tiles.

4. PERFORMANCE ANALYSIS CONTROL NETWORK

The performance of the control network is analyzed on two points. 1) Latency and 2) Power.

4.1. Latency

4.1.1. Bus

The latency of the bus is determined by two components: 1) The time for a bus transfer and 2) The time of the arbiter to grant a request of the tile. The time of a bus transfer ($T_{bus_{transfer}}$) after a request is granted is equal to l_{packet} due to the serialization of the data.

The time to grant a request is at least 1 cycle after the request has been sent to the arbiter. But this may be increased, due to: 1) The number of other tiles that have an active request, 2) The priorities of these requests, 2) the average load (L_{bus}) of the requests and 4) The time to handle a request, which is equal to the time required for a bus transfer.

In our bus configuration we have built a round-robin arbitration with N requesting tiles. This results that all requests have the same priority level and that at most $(N-1)$ requests have to be served before the request of a specific tile is granted. The average waiting time is therefore equal to:

$$T_{bus_{request}} = 0.5 \cdot (N-1) \cdot L_{bus} \cdot T_{bus_{transfer}} \tag{4}$$

The total latency T_{bus} of one single bus transfer is then equal to:

$$T_{bus} = 1 + T_{bus_{request}} + T_{bus_{transfer}} \tag{5}$$

4.1.2. Ring

The latency of the ring is determined by three components: 1) The time to access the ring and put the complete packet on the ring $T_{ring_{access}}$, 2) The time to pass through intermediate hops of the ring and 3) The time required to get off the ring $T_{ring_{leave}}$. The total latency T_{ring} of one single packet transferred between two tiles is then equal to:

$$T_{ring} = T_{ring_{access}} + T_{hop} \cdot (N_{hops} - 1) + T_{ring_{leave}} \tag{6}$$

Where N_{hops} is the distance between two tiles on the ring (i.e. two neighboring tiles have N_{hops} equal to 1) and T_{hop} is the number of cycles to pass one ring block. Because the priority in the ring-arbiter is always in favor of the ring $T_{hop} = 5$ [cycles] and $T_{ring_{leave}} = 1$ [cycle]. The time to put data on the ring at least $l_{packet} + 2$ [cycles]. But this can increase if the sending tile is blocked, due to the locally observed load of the ring (L_{ring}). For $l_{packet} = 6$ the average access latency of the ring is measured with a VHDL-model of the ring. The relation between the average access latency of $T_{ring_{access}}$ and the observed load of the ring is presented in Figure 3.

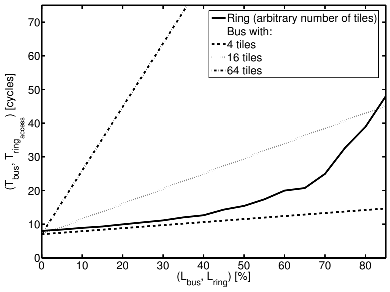


Fig. 3. Access latency noticed by the sending tile

4.1.3. Comparison

Figure 4 depicts the average latency that one single packet notices to access the communication network and reach its destination. For the bus this is equal to T_{bus} and is independent on the distance between sending and receiving tile. For the ring this is equal to T_{ring} and is dependent on the distance between the sending and receiving tile. In Figure 4 we assumed that the traffic is uniform distributed. This results in a average number of required hops equal to $N/4$. This figure shows that the latency of a packet on the ring increases with the number of tiles.

4.2. Power

The power consumption of the proposals for the control network depends on the number of tiles (N), the length of the actively used wires between the tiles and the traffic load generated by the tiles L_{tiles}, where $0 \leq L_{tiles} \leq 1$. We varied the amount of traffic to derive the load dependency. For all used data-sets, we observed an average switching activity of 50%.

4.2.1. Bus

For the power consumption of the total bus we have three components: 1) The interface block at every tile interface. This block is used to serialize/de-serialize the data and control the request/acknoledge protocol with the arbiter, 2)Wires

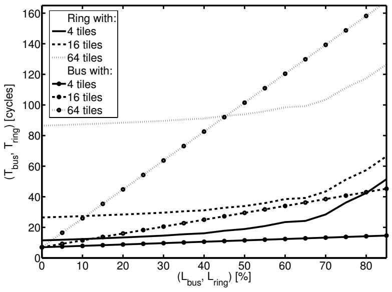


Fig. 4. Tile-to-Tile latency noticed by the packet

for data, select and request/acknowledgement lines and 3) Centralized arbiter and its multiplexer.

The power consumption of a interface-block can be estimated with a linear equation depending on an constant offset and a variable part depending on the input and output load:

$$P_{interface_{dyn}} = ib_s + ib_{d_{in}} \cdot L_{bus2tile} + ib_{d_{out}} \cdot L_{tile2bus} \quad (7)$$

where $L_{bus2tile} = L_{tile2bus}$ is the load of the bus interface in each direction. We varied both input load ($L_{bus2tile}$) and output load ($L_{tile2bus}$) of the interface to derive the constants ib_s, $ib_{d_{in}}$ and $ib_{d_{out}}$. Using Synopsys power compiler we estimated the constants, which are equal to $ib_s = 4.30$, $ib_{d_{in}} = 2.01$ and $ib_{d_{out}} = 1.30$ for $.13 \mu m$ and $V_{DD} = 1V$.

The power consumption in the wires in the bus depends on the layout of the bus. If we use the central multiplexer option we have point-to-point wires to the central multiplexer/de-multiplexer. The maximum length of the point to point wires determines the maximum frequency of the bus. This maximum length is equal to $l_{wire} \cdot (\sqrt{N} - 1)$, which is the distance from a corner of the chip to central placed multiplexer and de-multiplexer. The average length of the point-to-point wires determines the power consumption at an uniform distributed traffic offer. The average length of the active wires is equal to: $\bar{l}_{mux} \approx 0.5 \cdot l_{wire} \cdot \sqrt{N}$ if the SoC is organized in $\sqrt{N}$ x $\sqrt{N}$ processing tiles like Figure 1(a) and the arbiter is centralized. For non-square and/or non-centralized layouts the average length of the wires increases.

Combining the average wire length with Equation 3 and 7 the power dissipation for a 4 bit wide bus can be estimated. It equals the total offset in dynamic power consumption of all the interfaces, twice the power consumption of an point-to-point link (one from source to the multiplexer and one from de-multiplexer to the destination) with the minimum average length $\bar{l}_{mux}$ and the load dependent power consumption of both receiving and sending interfaces. The total power consumption of the bus is than equal to:

$$\begin{aligned} P_{bus} &= P_{interface} + P_{wires} + P_{arbiter} \\ &\approx ib_s \cdot N + (ib_{d_{in}} + ib_{d_{out}}) \cdot L_{bus} + \\ &\quad 2 \cdot \big(0.5 \cdot L_{bus}(s(c_0 + c_s) + c \cdot \bar{l}_{mux}) \cdot N_{wires}\big) \\ &= 4.30 \cdot N + \big(6.45 + 0.96 \cdot \bar{l}_{mux}\big) \cdot L_{bus} \qquad (8) \end{aligned}$$

Where the energy consumption of the arbiter, multiplexers and request/acknowledgement wires are neglected.

If the bus has the tristate structure we have point-to-multi-point wires. The total length of the active wires is then always equal to $l_{tristate} = (N-1) \cdot l_{wire}$. These wires connect one active driver with N receivers. But due to the point-to-multi-point organization of the wire we constructed a lumped wiring model of the tristate bus. From this model we derived the power consumption in the tristate organized bus:

$$\begin{aligned} P_{bus} &= P_{interface} + P_{wires} + P_{arbiter} \\ &= ib_s \cdot N + (ib_{d_{in}} + ib_{d_{out}}) \cdot L_{bus} + \\ &\quad (0.5 \cdot L_{bus}(s(c_0 \cdot N + c_s) + c \cdot l_{tristate}) \cdot N_{wires}) \\ &= 4.30 \cdot N + (4.37 + 0.51 \cdot N + 0.48 l_{tristate}) \cdot L_{bus} (9) \end{aligned}$$

4.2.2. *Ring*

For the power consumption of the total ring we have two components: 1) Ring block at every tile interface and 2)Wires between the ring blocks. For the ring block the power consumption can be estimated by an linear equation depending on the load of the ring:

$$P_{ring-block_{dyn}} = ir_s + ir_d \cdot (L_{ring_0} + L_{ring_1}) \qquad (10)$$

where L_{ring_x} is the locally observed load of $ring_x$ and $0 \leq L_{ring_x} \leq 1$. With the VHDL-model of the ring block a power estimation with Synopsys power compiler has been performed to find values for the constants ir_s and ir_d. We found the offset constant $ir_s = 8.48$ and the load dependent constant $ir_d = 1.91$ for $.13 \mu m$ and $V_{DD} = 1V$.

Combine Equation 3 and 10 results in the total power consumption of the ring:

$$\begin{aligned} P_{ring} &= (ir_d + 0.5 \cdot (s(c_0 + c_s) + c \cdot l_{wire}) \cdot N_{wires} + \\ &\quad ir_s) \cdot N \cdot (L_{ring_0} + L_{ring_1}) \\ &= 8.48 + (3.48 + 0.48 \cdot l_{wire}) \cdot N \cdot \left(\sum_{i=0}^{1} L_{ring_i} \right) (11) \end{aligned}$$

4.2.3. *Comparison*

Figure 5 depicts the total dynamic power consumption of the control network depending on the number of tiles. The load of the control network is set to 100% to get the worst-case overall power consumption of the control network. The length between the tiles (l_{wire}) is set to 2 mm. From Figure 5 it can be concluded that the most power-efficient control network is the bus with the central multiplexer. But in this figure we do not take the total throughput of the control network into account.

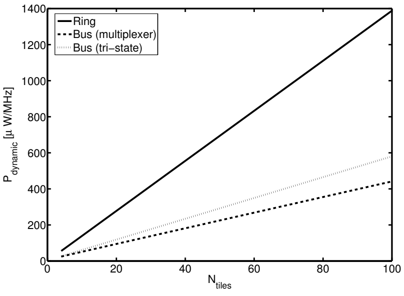


Fig. 5. Dynamic power consumption of total control network

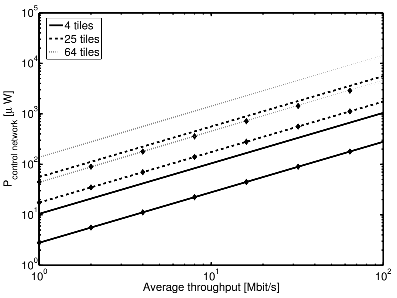


Fig. 6. Dynamic power consumption at a specific average throughput with a network load of 50% (line with diamonds is the ring))

Therefore, we compared the multiplex bus with the chordal slotted ring and normalized the figures to a given throughput. From a latency point of view (see Figure 3) we want to have at most 50% average load on the control network. With a load of $> 50\%$ the access latency of the ring gets too high. Using this constraint on the load we determine the required frequency of the network to be able to offer the required throughput of the control network. With this frequency we determine the dynamic power consumption of the network, which is depicted in Figure 6. Figure 6 shows that: 1) The chordal ring can offer approximately 3.4 times more throughput at the same power consumption and 2) The chordal ring consumes 3.2 times less power at the same throughput requirements.

5. CONCLUSION

In this paper two communication architectures are analyzed that can be used for best-effort communication in a Network-on-Chip. This best-effort communication is required for control of the tiles and for configuration of the circuit-switched network that handles the guaranteed throughput traffic. We have two implementations: one based on a bus and one based on a chordal slotted ring. Both architectures have been compared on their latency behavior and power consumption.

Based on the results with the latency and throughput analysis we can conclude that the ring is the best suited network if a low clock frequency, high throughput and/or low access latency is required for the control network. Especially, the latency of the bus increases rapidly with the number of tiles.

Based on the results of the power analysis we can conclude that the overall power consumption of the bus is relatively low compared to the ring architecture. This is mainly caused by the smaller interface blocks between the network and the processing tile. However, if we tune the frequency of the network to meet the throughput requirements of the control network, we see that the ring consumes less energy per transported bit. This is due to the concurrent communication streams that can be handled by the chordal slotted ring.

6. REFERENCES

[1] http://www.smart-chips.com.

[2] G. Kahn, “The semantics of a simple language for parallel programming,” in *Information processing*, J. L. Rosenfeld, Ed. Stockholm, Sweden: North Holland, Amsterdam, Aug 1974, pp. 471–475.

[3] L. Benini and G. de Micheli, “Networks on chips: A new soc paradigm,” *IEEE Computer*, vol. 35, no. 1, pp. 70–78, January 2002.

[4] L. T. Smit, *et al.*, “Run-time mapping of applications to a heterogeneous reconfigurable tiled system on chip architecture,” in *Proceedings of the International Conference on Field-Programmable Technology*, December 2004.

[5] P. T. Wolkotte, *et al.*, “An energy-efficient reconfigurable circuit-switched network-on-chip,” in *Proceedings of the 12th Reconfigurable Architectures Workshop (RAW 2005)*, Denver, Colorado, USA, April 4-5 2005.

[6] P. M. Heysters, G. K. Rauwerda, and G. J. M. Smit, “Implementation of a HiperLAN/2 receiver on the reconfigurable montium architecture,” in *Proceedings of the 11th Reconfigurable Architectures Workshop (RAW 2004)*, Santa Fé, New Mexico, USA, April 26-27 2004, iSBN 0-7695-2132-0.

[7] G. K. Rauwerda and G. J. M. Smit, “Implementation of a flexible rake receiver in heterogeneous reconfigurable hardware,” in *Proceedings of the 2004 International Conference on Field Programmable Technology (FPT)*, Brisbane, Australia, December 6-8 2004.

[8] P. T. Wolkotte, G. J. M. Smit, and L. T. Smit, “Partitioning of a drm receiver,” in *Proceedings of the 9th International OFDM-Workshop*, Dresden, Germany, September 2004, pp. 299–304.

[9] N. Kavaldjiev, G. J. M. Smit, and P. G. Jansen, “A virtual channel router for on-chip networks,” in *Proceedings of IEEE International SOC Conference*. IEEE Computer Society Press, September 2004, pp. 289–293.

[10] J. Rexford and K. G. Shin, “Support for multiple classes of traffic in multicomputer routers,” in *Proceedings of the First International Workshop on Parallel Computer Routing and Communication*. Springer-Verlag, 1994, pp. 116–130.

[11] http://www.synopsys.com.

[12] A. Morgenshtein, *et al.*, “Comparative analysis of serial vs. parallel links in networks on chip,” in *Proceedings SOC 2004 International Symposium on System-on-Chip*. Tampere, Finland: IEEE Computer Society Press, Los Alamitos, California, November 2004, iSBN 0-7803-8558-6.

[13] K. Banerjee and A. Mehrotra, “A power-optimal repeater insertion methodology for global interconnects in nanometer designs,” *IEEE Transactions on Electron Devices*, vol. 49, no. 11, pp. 2001–2007, November 2002.