Advances in Radio Science (2004) 2: 181–186

Advances in Radio Science

Application driven evaluation of network on chip architectures for parallel signal processing

C. Neeb, M. J. Thul, and N. Wehn

University of Kaiserslautern, Germany

Abstract. Today's signal processing applications exhibit steadily increasing throughput requirements which can be achieved by parallel architectures. However, efficient communication is mandatory to fully exploit their parallelism. Turbo-Codes as an instance of highly efficient forward-error correction codes are a very good application to demonstrate the communication complexity in parallel architectures. We present a network-on-chip approach to derive an optimal communication architecture for a parallel Turbo-Decoder system. The performance of such a system significantly depends on the efficiency of the underlying interleaver network to distribute data among the parallel units. We focus on the strictly orthogonal n-dimensional mesh, torus and k-ary-n cube networks comparing deterministic dimension-order and partially adaptive negative- first and planar-adaptive routing algorithms. For each network topology and routing algorithm, input- and output-queued packet switching schemes are compared on the architectural level. The evaluation of candidate network architectures is based on performance measures and implementation cost to allow a fair trade-off.

1 Introduction

The network-on-chip approach is a new design paradigm where models and techniques from the computer network community are employed and reevaluated under diversified constraints of the SOC-design approach. Today's technologies allow to integrate vast quantities of functional units on a single chip, where the communication between these components becomes as important as the computations they perform. It is predicted that in the future complex designs with hundreds of functional blocks will be integrated by application-specific networks that offer a high degree of optimization.

Correspondence to: C. Neeb
(neeb@eit.uni-kl.de)

In this work we investigate direct network architectures as the most popular interconnection schemes found in many parallel multiprocessor architectures. According to a classification introduced by Duato et al. (2003) we restrict our attention to the subclass of packet switched, strictly orthogonal point-to-point networks. These networks are characterized by their interconnection topology, that determines how nodes are physically interconnected. Each node represents a processing unit realizing the required computation. A communication specific component, the so called router, is attached to it for data transportation. The router implements the routing algorithm which determines the paths on which data is sent through the network.

The efficiency of a communication architecture is strongly application dependent. For a meaningful evaluation, performance measures like network throughput and latency and also area and energy consumption have to be put into the context of the application. A parallel Turbo-Decoder system presented by Thul et al. (2002a) serves as an application example.

Turbo-Codes belong to the iterative channel coding techniques that exhibit an outstanding forward-error correction capability, which made them part of today's communication standards, e.g. 3GPP (Third Generation Partnership Project), and are often found in the outer modem of wireless transmission systems.

For high-throughput, parallel decoder architectures are employed where data distribution due to interleaving makes up the major bottleneck. Therefore our evaluation of the considered networks is based on the specific demands of efficient interleaving architectures in parallel Turbo-Decoders. It should be noted that this topic is not primary specific to Turbo-Codes and thus can be transfered to e.g. LDPC-Decoders where data distribution due to interleaving is even more challenging.

In Sect. 2 we outline the design space for our network-on-chip approach, introducing the network topologies and the applied routing algorithms. Further two packet switched router architectures based on input- and output- queuing

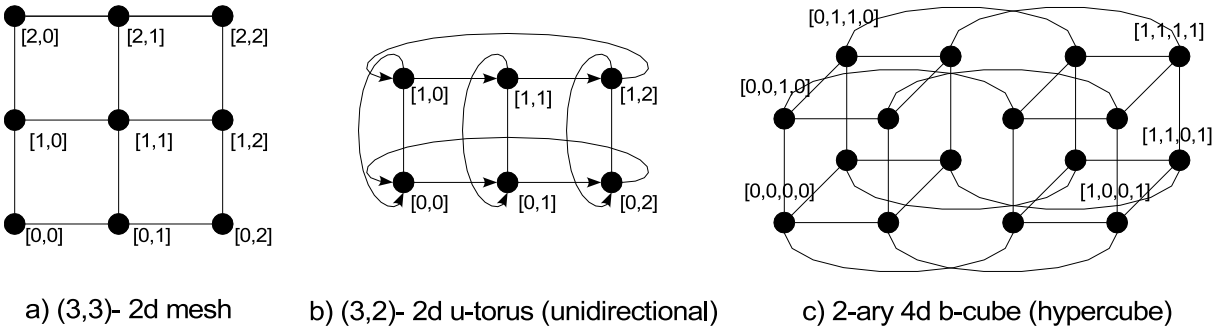

Fig. 1. Examples for mesh, torus and k-ary n-cube network topologies.

schemes are discussed in detail. Section 3 presents the special requirements of interleaver networks in parallel Turbo-Decoders, followed by their evaluation in Sect. 4. In Sect. 5 performance and implementation costs are opposed for an exemplary decoder architecture comprised of 16 processing units. Section 6 concludes this paper.

2 Network-on-chip design

2.1 Network topology

Network topologies are traditionally modeled by a strongly connected directed graph $G(N, C)$, where the vertices N represent processing nodes and the edges C the set of connecting physical channels. We focus on n-dimensional meshes, tori and k-ary n-cubes as the most popular representatives of strictly orthogonal networks which are widely employed in commercial multicomputer systems.

They can be defined by a n-dimensional radix vector $\boldsymbol{k}$, where its components $\boldsymbol{k}_i, 0 \leq i < n$ determine the number of nodes along dimension i. A node x is identified by its coordinates $(x_{n-1}, x_{n-2}, ..., x_1, x_0)$, where $0 \leq x_i < \boldsymbol{k}_i$.

In n-dimensional meshes neighboring nodes are connected by bidirectional channels, where each node has from n to $2n$ neighbors and thus making the mesh an irregular network. An example of a 2d-mesh with $\boldsymbol{k} = (3, 3)$ is given in Fig. 1a. For the sake of simplicity undirected edges are used to symbolize two directed edges leading to opposite directions.

In the torus network wrap-around channels are added to the boundary nodes which gives the torus regularity and symmetry (see Fig. 1b). In contrast to the mesh network adjacent nodes can alternatively be linked by unidirectional channels. We refer to this topology as the u-torus whereas in b-tori nodes are interconnected by bidirectional channels respectively.

The k-ary n-cube represents a special case of the torus, where each dimension accommodates the same number of nodes making up to $\boldsymbol{k}^n$ nodes altogether. For $\boldsymbol{k} = 2$ this topology is also referred to as binary n-cube or hypercube (Fig. 1c).

2.2 Routing algorithm

A predominant advantage of the introduced networks is the regular construction pattern and their high degree of symmetry. This essentially simplifies the employed routing algorithms, which specify the set of allowed paths in the network on which data can travel from an emitting source node to a destined target node. The employed routing algorithm is tightly coupled to the network topology and crucial for efficient communication in parallel architectures.

A routing algorithm is denoted as deterministic if the transmission of a packet, containing target address and data, always follows the same path for a pair of source/target nodes. On the opposite, adaptive algorithms provide alternative paths where local traffic estimates can be exploited to circumvent congested channels and hot-spots in the network. However, to achieve reliable communication, routing freedom is essentially restricted to avoid the case of deadlock (Glass and Ni, 1992). In these situations packets cannot advance towards their destination because requested network resources are already allocated by other packets forming a cyclic dependency. They can systematically be avoided by the introduction of virtual channels as proposed by Dally and Seitz (1987). Virtual channels represent logical packet streams inside a router that share the physical channels for transmission. The impact on router implementation is discussed in Sect. 2.3.

In this paper we restrict our analysis on deadlock-free routing algorithms namely the deterministic dimension-order (Dally and Seitz, 1987), the partially adaptive negative-first (Glass and Ni, 1992) and the planar-adaptive routing proposed by Chien and Kim (1992). In dimension-order routing packets are crossing dimensions in a strictly increasing order. Effectively the address offset of the current node and the target node is computed and incrementally reduced to zero starting with the lowest dimension until the target is reached. This process of clearing dimensions in a fixed order always leads to the selection of the same shortest path between a source and a destination node. The negative-first routing algorithm implies less restrictions on the set of allowed paths and thus provides some adaptivity. At first packets are routed adaptively in negative directions and only when no negative

dimension offsets are left then routing proceeds adaptively in positive directions. The planar-adaptive algorithm routes packets in a series of 2d-planes. It provides full adaptivity inside the current plane whereas the transition to the next plane is fixed. Consequently this algorithm can be classified as fully adaptive in 2d-networks whereas only partial adaptivity is provided for higher dimensional networks. To guaranty deadlock-freedom at most three virtual channels for meshes and six virtual channels for tori and cubes are necessary.

2.3 Router architectures

The implementation of the three mentioned routing algorithms is based on two packet switched router architectures. They mainly differ in the way pending packets are internally buffered in the case of contention whereas the appropriate choice does not depend on the algorithm itself.

In the input-queued router (IQ) depicted in Fig. 2a a dedicated input-queue (FIFO buffer) is attached to each inport of the router. Physical channel flow-control (FC) is realized by link-controllers (LC) that stop neighboring nodes sending data packets in case of a full queue. To support virtual channels for deadlock-avoidance extra queues are inserted where each is assigned to a single virtual channel. In this case the link controller must demultiplex incoming packets of the same physical channel on the appropriate virtual channel queue. An additional virtual channel arbiter (VC-Arbiter) selects a packet in one of these queues and forwards it to the address decoder (AD) where the actual routing is done. Dependent on the target address of the packet, one (deterministic routing) or multiple outports (adaptive routing) are requested for further packet delivery. The physical connectivity is realized by a crossbar-switch which is controlled by the scheduler to resolve contending requests for the same outport at the same time. These conflicts are the main reason for the noticeable degradation of the throughput in input-queued routers. We choose the so called SLIP scheduling algorithm (McKeown, 1995) for implementation as it is based on a fair round-robin scheme.

The output-queued router (OQ) circumvents this switching loss by the assignment of a dedicated queue for each pair of in-/out-channels. Hence multiple packets can be forwarded to the same outport at the same time as they go to different out-queues. Again a round-robin scheme is used to multiplex the contents of the queues on the outgoing physical channels. The high throughput and channel utilization of this architecture comes at the expense of the large number of queues needed restricting applicability to low-dimensional routers only. Figure 2b illustrates the output-queued architecture without additional virtual channel queues for the sake of clarity.

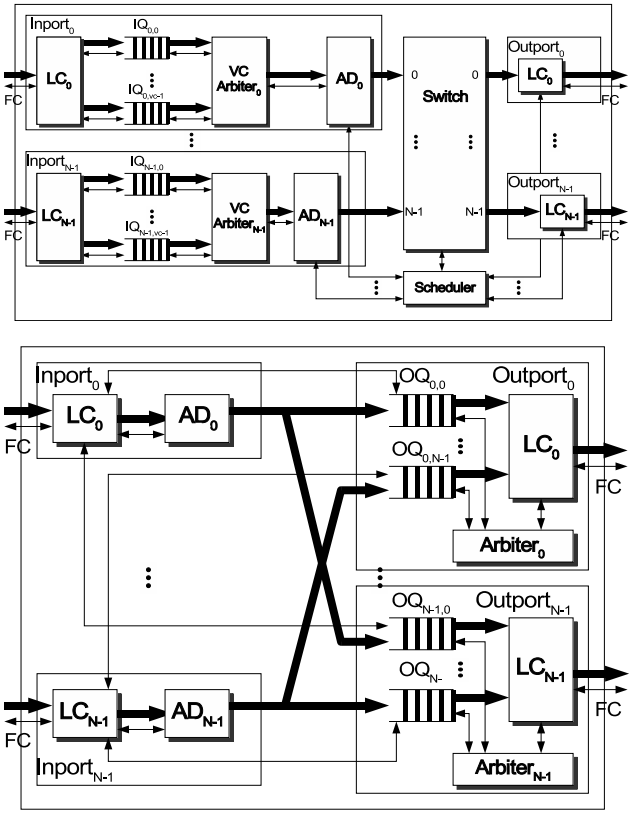

Fig. 2. Router architectures based on **(a)** input-queuing and **(b)** output-queuing.

3 Interleaving in parallel turbo-decoders

For the evaluation of our network-on-chip approach we refer to concurrent interleaving in a partial parallel Turbo-Decoder as the application background. Interleaving is used in many channel coding schemes and essentially impacts the communication performance of Turbo-Codes. It determines a rearrangement of data inside a block to scramble their processing order and thus to break up neighborhood-relations effectively. In parallel Turbo-Decoders each processing unit works on a subblock of data individually where information must be exchanged according to the adopted interleaving scheme.

From a network perspective an all-to-all communication must be established. A fully interconnected architecture where all units are linked by dedicated channels exhibits a high wiring effort which becomes infeasible for increasing degrees of parallelization. Additionally, write conflicts occur when two packets are sent to the same unit at the same time. As only one packet will be accepted by the receiver, the queuing of data becomes inevitable. A dedicated buffer per channel has to be provided which further increases implementation costs drastically.

This issue has already been addressed by Thul et al. (2002c) for the first time where interleaving is identified as the upcoming bottleneck for high-throughput Turbo-Decoding. An optimized architecture based on a ring topology has been presented by Thul et al. (2002b). Both architectures do not employ any flow-control mechanism. They require large queues which are dimensioned by means of

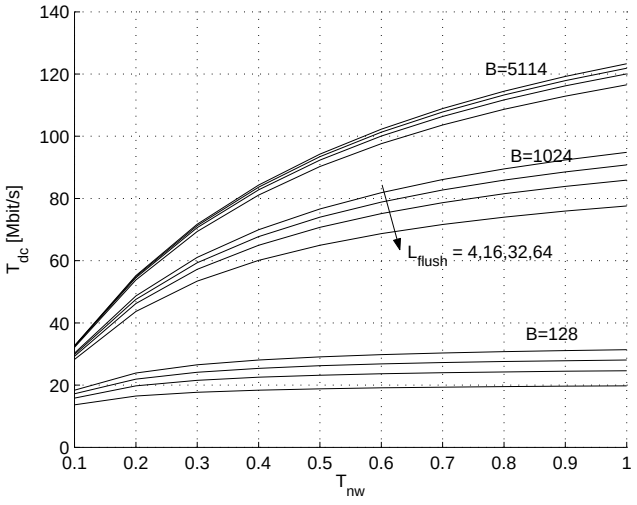

Fig. 3. Impact of the interleaver network throughput and latency on the throughput of a parallel turbo-decoder ($N = 16$, $f = 200\,\text{MHz}$).

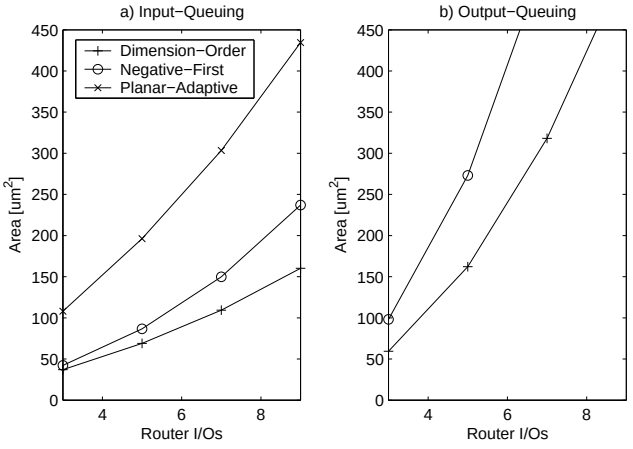

Fig. 4. Quantitative comparison of router complexity for n-dimensional meshes, $Q = 4$.

exhaustive simulations. We will refer to the latter as a benchmark of our approach that requires substantially smaller buffers.

To rate the various network architectures in terms of performance, we derive a model to capture the interleaving specific requirements and to quantify the impact on the overall decoder. The throughput of the considered parallel Turbo-Decoder is given as:

$$T_{dc} = \frac{B \cdot f}{L_{dc}}$$

with B the data block length, f the clock frequency and L_{dc} the decoder latency. Data blocks of length B are partitioned into N subblocks of length B/N where each is decoded separately by one of the N processing units. The decoder latency L_{dc}

$$L_{dc} = B + 2It(\frac{B}{N} + 2W + c + L_{il})$$

amounts to the number of clock cycles needed to decode a complete data block and mainly depends on the interleaver latency L_{il}. According to the given interleaving scheme, N data values have to be communicated per clock cycle. The contribution of the interleaver to the decoder latency can be subdivided into two terms:

$$L_{il} = L_{\text{stall}} + L_{\text{flush}} = \frac{B}{N}(\frac{1}{T_{nw}} - 1) + L_{\text{flush}}(T_{nw}, Q)$$

L_{stall} comprises the fraction of clock cycles a processing unit has to be stalled to avoid overloading the network and hence loosing any packets. The probability of a processing stall particularly depends on the normalized average throughput T_{nw} of the interleaver network. Before the next half-iteration can be started, all data has to be distributed. Thus we define the flush latency L_{flush} as the time span between the arrival of the last data value at the entrance and the time it leaves the interleaver network. During this time internal data queues are emptied which is mainly influenced by the network throughput and the depth Q of the router queues. The impact on the overall throughput of the Turbo-Decoder is illustrated in Fig. 3 for $N = 16$.

4 Network evaluation

To allow a fair trade-off of the suitability of a network architecture for interleaving, we oppose implementation costs and performance benefits. For further discussions we make the following reasonable assumptions:

1. Data is distributed with equal probability of $1/N$ to any of the target units. This is implied by good interleavers from a communication point of view where data is nearly evenly spread over the address space. We therefore adopt a uniform traffic model where one data value can be delivered by each unit per clock cycle.
2. A physical channel of the network is capable to transfer a single data packet in one clock cycle. We assume a typical channel width of 20 bits for UMTS compliant interleaving.

4.1 Router implementation cost

Figure 4 shows a comparison of the logic area requirements for a single router with respect to the routing algorithm and the number of I/O-ports. For this we assume routing in mesh networks which imply a minimal set of virtual channels and an equal queue-depth to store four packets. The results were obtained through synthesis based on a standard $0.18\,\mu$m technology using Synopsis Design Compiler.

All input-queued routers opposed in Fig. 4a are dominated by the size of the queues which make up 50–70% of the overall area. With an increasing degree of adaptivity the routers exhibit a growing complexity of the scheduler (8–22%) and

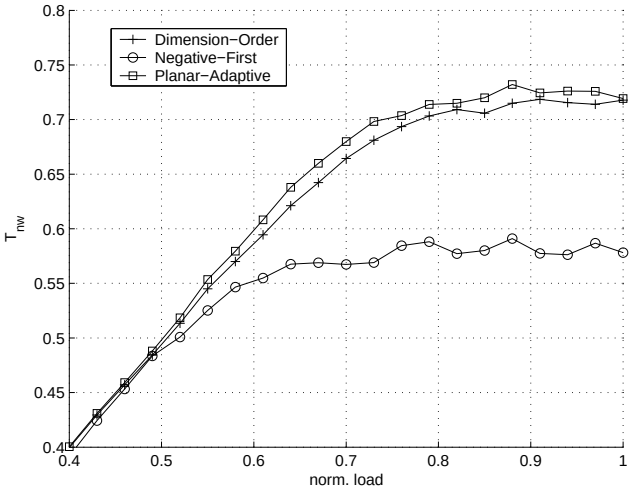

Fig. 5. Normalized throughput of the routing algorithms using input-queuing in a (4,4)-2d-mesh, $Q = 6$.

the crossbar switch (9–16%). The address decoders which implement the routing algorithm have only a slight influence of about 4–12% on the total area. Particularly for the planar-adaptive router the additional queues and extra arbiters for virtual channel support are expensive to implement.

The output-queued routers opposed in Fig. 4b need substantially more queues compared to the input-queued routers with a contribution of about 85% to the total area. Due to the restrictions of the dimension-order algorithm in contrast to the negative-first, less queues are needed to implement the former. We do not consider the planar-adaptive router for output-queuing as the need for virtual channels requires again a multiple of the already high number of queues.

4.2 Router performance

To quantify the impact of the routing algorithms on network performance for interleaving, simulations are used to measure throughput under varying load. For this, a (4,4)-2d-mesh network using input-queued routers is loaded with uniform traffic. As shown in Fig. 5 negative-first routing achieves the lowest throughput of all for high loads whereas planar-adaptive and dimension-order nearly perform equally. With increasing network dimension, throughput of planar-adaptive routing proved to be even worse compared to deterministic routing. This can be explained by the fact that partially adaptive routing uses local traffic estimates to make a routing decision which distorts the evenness of the global uniform traffic. Opposed to this, deterministic routing maintains this characteristic, spreading traffic more evenly across the network. Similar results are obtained for torus and cube networks as well. Consequently we regard the deterministic dimension-order routing as the best suited algorithm for interleaving. It achieves the highest throughput and comes at lowest implementation cost.

Table 1. Maximum number of nodes per dimension and in total for 2d- and 3d-networks.

Topology	U_c	BW	$\boldsymbol{k}_{\max}$	$N_{\max}$ (2D)	$N_{\max}$ (3D)
u-torus	$\frac{N-1}{2 \cdot BW}$	$2\boldsymbol{k}_1\boldsymbol{k}_2...\boldsymbol{k}_{n-1}$	≤ 3	9	27
mesh	$\frac{N}{2 \cdot BW}$	$2\boldsymbol{k}_1\boldsymbol{k}_2...\boldsymbol{k}_{n-1}$	≤ 4	16	64
b-torus	$\frac{N}{2 \cdot BW}$	$4\boldsymbol{k}_1\boldsymbol{k}_2...\boldsymbol{k}_{n-1}$	≤ 8	64	512

4.3 Topology constraints

Both cost and performance of an interleaver network are heavily affected by the choice of the appropriate topology. In the following we derive a necessary condition for the maximum number of nodes per dimension such that throughput is not degraded by the network topology. For that, we refer to the "bisection" of a topology as the minimum cut to divide the network into two equal sets of nodes. Accordingly the "bisection width" BW comprises the number of channels that have to be cut (see Duato et al., 2003, for further details).

Due to the regular construction and symmetry of the topologies, $(N-1)/2$ packets in u-torus networks and $N/2$ packets in meshes and b-tori respectively have to cross the bisection during each cycle in average. We can therefore easily derive the normalized channel utilization U_c as the fraction of clock cycles a packet allocates a single channel by the following contemplations: Lets assume that dimension zero contains the largest number $\boldsymbol{k}_0 = \boldsymbol{k}_{\max} = \max\{\boldsymbol{k}_i\}$ of nodes in the network with $N = \boldsymbol{k}_0 k_1 ... \boldsymbol{k}_{n-1}$ nodes altogether, then the bisection is orthogonal to this dimension. Since at most one packet can be transferred over a single channel per cycles it follows that $U_c \leq 1$. This condition leads to the maximum number of nodes for any dimension $\boldsymbol{k}_{\max}$ depending on the bisection width BW of the topology summarized in Table 1. Consequently the largest number of nodes can be accommodated by cube topologies where the same number of nodes resides in all dimensions. Thus the number of dimensions of the topology sets an upper bound of nodes that can be integrated by the network.

5 Results

The suitability of the presented networks for interleaving is discussed for an exemplary Turbo-Decoder with $N = 16$ parallel processing units. We investigate a (4,4)-2d-mesh and a 4-ary-2d-b-cube using input- and output-queued dimension-order routing. Both of the considered topologies comply to the condition derived in Sect. 4.3 as they do not exceed the maximum number of four nodes per dimension in 2d-meshes and eight nodes for 2d-b-tori (-cubes) respectively. We exclusively employ deterministic dimension-order routing because it achieves the highest throughput (Sect. 4.2) at the lowest cost in all considered topologies (Sect. 4.1).

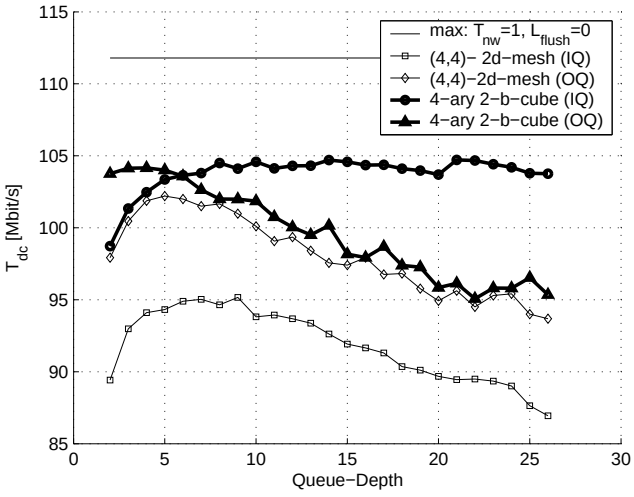

Fig. 6. Impact of router queue-depth on the turbo-decoder throughput using dimension-order routing ($N = 16$, $B = 2048$).

Table 2. Router queue-depths for optimal trade-off of decoder throughput and network area.

Network	Q	T_{dc} [Mbit/s]	$Area_{nw}$ [mm^2]
4-ary-2d-b-cube (IQ)	5	104	1.3
4-ary-2d-b-cube (OQ)	2	104	1.5
(4,4)-2d-mesh (OQ)	4	102	1.7
(4,4)-2d-mesh (IQ)	6	95	1.2

Network throughput and the flush latency finally depend on the depth of the router queues. Thus further performance simulations are carried out with varying queue-depths under full load (norm. load = 1) to quantify their impact on the overall decoder throughput. A fixed number of packets is sent across the network then the source is switched off to measure the flush latency. For small queue sizes, network throughput tends to deteriorate and thus the processing units often have to be stalled (see Fig. 6). With deeper queues throughput of the interleaver network reaches saturation and the flush latency becomes dominant. We choose the minimal queue-depth where at least 90% of the maximum throughput is achieved for comparison. The total network costs are estimated by extrapolation of the single router area with according queue-depths.

Compared to the architecture proposed by Thul et al. (2002b) an area reduction of factor ten for the interleaver network is achieved. Only very small queues are necessary in our approach to implement a high-throughput interleaver network that is capable to handle all possible interleaving schemes.

6 Conclusion and future work

Strictly orthogonal networks facilitate very efficient implementations of interleaver networks. In this context deterministic routing is superior to partial adaptive algorithms in terms of performance and implementation costs. The employment of network flow-control drastically reduces the sizes of the queue buffers and hence logic area of the whole network.

Particularly high-dimensional topologies offer only limited scalability. Future work will have to focus on the exploration of more sophisticated topologies and deadlock-free routing algorithms that can be tailored to an arbitrary number of nodes.

Acknowledgements. This work has been supported by the Deutsche Forschungsgesellschaft (DFG) under grant We 2442/1-3 within the Schwerpunktprogramm “Grundlagen und Verfahren verlustarmer Informationsverarbeitung (VIVA)”.

References

Chien, A. and Kim, J.: Planar-adaptive Routing: Low-cost Adaptive Networks for Multiprocessors, in Proc. 19th International Symposium on Computer Architecture, 268–277, 1992.

Dally, W. and Seitz, C.: Deadlock-free Message Routing in Multiprocessor Interconnection Networks, in IEEE Transactions on Computers, 547–553, 1987.

Duato, J., Yalamanchili, S., and Ni, L.: Interconnection Networks – An Engineering Approach, Morgan Kaufman Publishers, San Francisco, USA, 2003.

Glass, C. and Ni, L.: The Turn Model for Adaptive Routing, in Proc. 19th International Symposium on Computer Architecture, 278–287, 1992.

McKeown, N.: Scheduling Algorithms for Input-queued Cell Switches, Ph.D. thesis, University of California, Berkeley, 1995.

Third Generation Partnership Project: 3GPP home page, www.3gpp.org.

Thul, M. J., Gilbert, F., Vogt, T., Kreiselmaier, G., and Wehn, N.: A Scalable System Architecture for High-Throughput Turbo-Decoders, in Proc. 2002 Workshop on Signal Processing Systems (SiPS’02), San Diego, California, USA, 152–158, 2002a.

Thul, M. J., Gilbert, F., and Wehn, N.: Optimized Concurrent Interleaving for High-Throughput Turbo-Decoding, in Proc. 9th IEEE International Conference on Electronics, Circuits and Systems (ICECS’02), Dubrovnik, Croatia, 1099–1102, 2002b.

Thul, M. J., Wehn, N., and Rao, L. P.: Enabling High-Speed Turbo-Decoding Through Concurrent Interleaving, in Proc. 2002 IEEE International Symposium on Circuits and Systems (ISCAS’02), Phoenix, Arizona, USA, 897–900, 2002c.