

ProMINoC: An efficient Network-on-Chip design for flexible data permutation

Phi-Hung Pham $^{\rm a)},$ Jongsun Park, and Chulwoo Kim $^{\rm b)}$

Dept. of Electronics and Computer Engineering, Korea University Anam-dong, Seongbuk-gu, Seoul 136–713, Korea a) pph@kilby.korea.ac.kr b) ckim@korea.ac.kr

Abstract: This paper presents a novel Network-on-Chip design to efficiently support data-interleaving with arbitrary permutation rule. The proposed NoC offers a run-time conflict resolution for interleaved data under arbitrary permutation rule by using a circuit-switching approach combined with a dynamic path-probing scheme. Experimental results in a $0.18 \,\mu\text{m}$ STD-cell CMOS process show that the proposed NoC can offer an aggregate bandwidth of up to 522.4 Gb/s, while occupying a compact area of $0.473 \,\text{mm}^2$ (52 kGates). A comparison with other interleaving networks shows the efficiency of the proposed design. **Keywords:** Network-on-chip, on-chip router, data permutation **Classification:** Integrated circuits

References

- J. Dielissen, N. Engin, S. Sawitzki, and K. van Berkel, "Multistandard FEC Decoders for Wireless Devices," *IEEE Trans. Circuits Syst. II*, vol. 55, no. 3, pp. 284–288, 2008.
- [2] C. Neeb, M. J. Thul, and N. Wehn, "Network-on-chip-centric approach to interleaving in high throughput channel decoders," *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 2, pp. 1766–1769, May 2005.
- [3] H. Moussa, O. Muller, A. Baghdadi, and M. Jezequel, "Butterfly and Benes-Based on-Chip Communication Networks for Multiprocessor Turbo Decoding," *Proc. Design Automation and Test in Europe (DATE)*, pp. 654–659, April 2007.
- [4] G. Masera, F. Quaglio, and F. Vacca, "Implementation of a Flexible LDPC Decoder," *IEEE Trans. Circuits Syst. II*, vol. 54, no. 6, pp. 542–546, 2007.
- [5] H. Moussa, A. Baghdadi, and M. Jezequel, "Binary de Bruijn interconnection network for a flexible LDPC/turbo decoder," *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 97–100, May 2008.
- [6] S. C. Liew, M.-H. Ng, and C. W. Chan, "Blocking and nonblocking multirate Clos switching networks," *IEEE/ACM Trans. Netw.*, vol. 6, no. 3, pp. 307–318, 1998.
- [7] P.-H. Pham, P. Mau, and C. Kim, "A 64-PE folded-torus intra-chip communication fabric for guaranteed throughput in Network-on-Chip based applications," *Proc. IEEE Custom Integrated Circuits Conference (CICC)*,





pp. 645–648, Sept. 2009.

1 Introduction

An important trend of modern reliable wireless systems is to implement Forward Error Correction (FEC) platforms capable of simultaneous supporting *multi-standard* and *multi-mode* [1]. By the advances in VLSI technology, multi-processor system-on-chips with Network-on-Chip (NoC) infrastructures have been adopted to flexibly meet the requirement of computation and communication for such platforms [2, 3, 4, 5]. One of the challenges in these NoC designs is to handle the intensive interleaving of exchanged data among the processing components [2, 3, 4, 5]. This challenge becomes harder when the interleaving (permutation) rule, varying from one standard to another and within a single standard, even can be considered as random. In this situation, an efficient NoC must be designed to flexibly route *any permutation* from the network inputs to its outputs, with the aim of fully exploiting the parallelism of the FEC (e.g., LDPC/Turbo) decoder architectures. Besides, the *minimization of the NoC overhead* is critical for limited on-chip implementation cost (i.e., area- and energy-efficiency).

Several related on-chip network designs targeting intensive data permutation in FEC decoding platforms are reported in literature. Previous work in [2] presents the parameterization (e.g., choosing buffer depth, routing algorithm, etc.) of a general-purpose 2D-mesh packet-switched network for interleaving data, rather than an optimized network design for bandwidth-/ area-efficiencies. In [3], several networks are proposed to solve the runtimeconflict of permutated data but requiring either costly FIFO queues with priority (as for the Butterfly) or complex time-slot allocation with more routing stages (as for the Benes). To meet flexibility, the Beneš network [4] requires a huge pre-calculation related to the code to configure the switches. The de Bruijn network [5] avoids buffering of conflicting data by using a complex dynamic routing mechanism with a deflection technique. This complexity results in degradation of operating frequency, and the deflection scheme induces energy-inefficiency of data transfer. This work proposes a novel and efficient Multistage Interconnection Network-on-Chip with a dynamic probing pathsetup scheme, called ProMINoC, to support on-chip data interleaving under any permutation rule.

2 Proposed Network-on-Chip design description

2.1 Network-on-Chip architecture with probing path-setup procedure

The basic idea of ProMINoC is the combination of a pipeline circuit-switching approach with the non-blocking property of a multistage interconnection network (MIN), in which data paths are dynamically established for conflict-free pipelined data transfer.





The circuit-switching approach removes the overhead of data buffers at switching nodes and achieves high bandwidth due to the pipelined design. The Clos network [6] is a kind of MINs that is widely used to build scalable switches in macro networks. A three-stage Clos network is defined as C(n, m, p), where n represents the number of inputs in each of p first-stage switches and m is the number of second-stage switches. We propose C(4, 4, 4)as a topology for ProMINoC to support data interleaving from 16 inputs to 16 outputs (Fig. 1a). This non-blocking topology is chosen so that a pathsetup scheme can always find a connection from an input to any idle output without rearranging existing ones. The path-setup scheme is proposed based on a conflict-free dynamic probing procedure. The concept of "probing" was first introduced in [7], used with the pipelined circuit-switching approach. In ProMINoC, a set of *Request* (*Req*) and *Answer* (*Ans*) (Fig. 1b) is used in inter-switch interconnection to support the probing procedure and the three phases (i.e., setup, transfer, and release) of an end-to-end communication. The probing procedure occurs in the setup phase, in which a probe (setup flit) containing destination address dynamically searches for an available path in a non-repetitive manner. When the probe reaches the destination, an ACK propagates back to the source to trigger a pipelined data transfer (i.e., the transfer phase). In the release phase, the *Req* is set to 0 right after the last data flit is sent.



Fig. 1. (a) Proposed 16x16 ProMiNoC with binary addressing scheme and (b) its inter-switch interconnection.

The probing path-setup operation can be illustrated though an example with Fig. 1 b. It is assumed that a probe from a source (e.g., an input of switch **01**) sets up a path to a target destination (e.g., an available output of switch **22**). The probe will non-repetitively try paths through 2^{nd} -stage switches **10**, **11**, **12**, and **13**. For instance, if the link **01-10** is available, the probe first tries this link (Req = 1) and then arrives at the switch **10**.

EiC

© IEICE 2010 DOI: 10.1587/elex.7.861 Received April 26, 2010 Accepted May 20, 2010 Published June 25, 2010

• If the link 10-22 is available, the probe will reach the switch 22 and



meet the target output. And then an ACK (i.e., Ans = Ack) propagates back to the input to trigger data transfer.

If the link 10-22 is occupied, the probe will move back to the switch 01 (Ans = Back) and the link 01-10 is released (Req = 0). From the switch 01, the probe can try the rest idle links leading to 2nd-stage switches in the same manner. By mean of moving back the probe when facing blocked link and trying others, the run-time conflict-free feature of the path-setup is guaranteed.

By this way, the probing procedure, combined with the non-blocking feature of the proposed topology, ensures to find a path from the input to the available output. The signal Ans = nAck (Fig. 1 b) is used for end-to-end flow-control when overflow occurs at the destination.

2.2 The switch designs

A common architecture of the switches is proposed as in Fig. 2 a. The switch architecture includes INPUT CONTROLs (ICs), OUTPUT CON-



(c)

Fig. 2. (a) The switch architecture with (b) the simplified FSM diagram and (c) the probing algorithms for switches in each stage.





TROLs (OCs), an ARBITER and a CROSSBAR. The incoming probe in the setup phase can be transported through the data path to save the wiring costs.

The ARBITER has two roles: first, as a cross-connection between the Ans_Outs and the ICs through the *Grant bus*, and second, as a referee for requests from the ICs. When an incoming probe arrives at an input, the corresponding IC observes the output status through the *Status bus*, and requests the ARBITER to grant it access to the corresponding OC through the *Request bus*. When accepting this request, the ARBITER also cross-connects the Ans_Out with the IC through the *Grant bus*. With the second role, the ARBITER, based on a static priority rule, resolves contention when several ICs are requesting the same free output. After this resolution, only one IC is accepted while the rest are considered as facing a blocked link (i.e., Ans = Back).

A simplified FSM diagram of IC is illustrated in Fig. 2b, with common states according to switching operation. It is noted that some states can be omitted according to the probing algorithm in each stage. For example, the state BACK is not implemented for switches in the 1st and 3rd stages. Assuming that a probe contains the 4-bit address of the destination, i.e., $D_3 D_2 D_1 D_0$ (see Fig. 1a for the addressing scheme). To support the proposed pathprobing scheme, ICs are implemented with different probing algorithms according to their switch stage, as shown in Fig. 2 c. In the 1st stage, the switch tries the free outputs in a non-repetitive manner (e.g., with order from outputs $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$) to avoid the searching of same path that may result in live-lock. The 2nd- and 3rd-stage switches rely on two MSBs (D_3D_2) and two LSBs (D_1D_0) of the destination address, respectively, to route the probe. The OCs work as re-timing stages for the ARBITER commands from Control bus and control the CROSBAR. The CROSSBAR is a 4x4 full-connecting matrix designed with output multiplexers. The OCs and the ARBITER are triggered by the rising and the falling edges of clock, respectively. By this implementation, the path-probing is dynamically processed by the switch in one clock cycle.

3 Experimental results

In ProMINoC, after a path is set up, the pipelined packet latency is just the packet serialization time plus the propagation time over the network diameter (i.e., 3 hops). By analyzing the design of ProMINoC with dynamic probing scheme, the setup latency for a one-to-one mapping is in the range of [8, 10, 12, 14] system clock cycles. The ProMINoC configured with 64-bit data-width is synthesized using the Synopsys tool in a 0.18 μ m CMOS STD-cell technology, under 1.8 V typical case. The worst-case latency (including setup-time for changing of permutation rule) of a 64-bit permuted packet is around 35 ns (18 cycles at F = 510 MHz). This worst-case value even outperforms those in [3], which are in a range of 53 ns~76 ns. This clearly shows the advantage of proposed dynamic probing scheme of ProMINoC for minimization





of setup latency. The implementation result of ProMINoC is compared with other on-chip interleaving networks, as shown in Table I. Due to the variety in topology, switching complexity, degree of flexibility (i.e., permutation rule, number of interleaving in/outputs), payload and flit size, technology, etc., it is difficult to compare quantitatively the values in Table I. Nevertheless, the overall efficiency of a network can be estimated based on the ratio of Aggregate Bandwidth to Network Overhead. Regarding this estimation, the ProMINoC shows the best efficiency over the others.

	[2]	[3]	[4]	[5]	Proposed
CMOS Tech. (µm)	0.18	0.18	0.13	0.18	0.18
Topology	(4,4) 2D	Butterfly	Benes	de Bruijn	3-stage Clos
	Mesh (IQ)	-		_	_
Inputs x Outputs	16 x 16	16 x 8	32x32	16x16	16x16
Frequency (MHz)	200	302	245	266	510
Aggregate BW	64	154	62.72	296	522.4
(Gb/s)					
Network Area	1.2	2.144	0.98	3.565	0.473
(mm^2)					

 Table I. Comparison of synthesis results with other on-chip interleaving networks.

4 Conclusion

In this paper, we have presented a novel NoC design with dynamic pathprobing scheme for arbitrary permutation of data. By experimental results, the proposed NoC offers a high aggregate bandwidth, while occupying a modest area. The comparison shows the efficiency of proposed NoC over other related designs.

Acknowledgments

This work was supported by Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MEST) (R0A-2007-000-20059-0) and by a Korea University Grant.

