Rediscovering Logarithmic Diameter Topologies for Low Latency Network-on-Chip-based applications

Rediscovering Logarithmic Diameter Topologies for Low Latency Network-on-Chip-based applications

Carlo Condo, Maurizio Martina, Massimo Ruo Roch, Guido Masera
Electronics and Telecommunications Department
Politecnico di Torino
Torino, Italy
Email: carlo.condo(maurizio.martina, massimo.ruoroch, guido.masera)@polito.it

***Abstract*—Low-latency Network-on-Chip (NoC) applications have tight constraints on the clock budget to perform communication among nodes. This is a critical aspect in NoC-based designs where the number of clock cycles spent for communication depends mainly on the topology and on the routing algorithm. This work deals with logarithmic diameter topologies, that were proposed for computer networks, and shows that an optimal shortest-path routing algorithm can be efficiently implemented on this kind of topologies by means of a very simple circuit. The proposed circuit is then exploited to reduce the area and the power consumption of a recently proposed NoC-based design. Experimental results show that the proposed circuit allows for a reduction of about 14% and 10% for area and power consumption respectively, with respect to a shortest-path routing-table-based design.**

I. Introduction

Network-on-Chip (NoC) is a design paradigm that has been proposed mainly to improve the flexibility and the scalability of on-chip interconnection systems [1]–[4]. Several directions have been investigated in the last years to improve the NoC efficiency and reliability and to reduce NoC complexity and power consumption (e.g. [5]–[7]), including wireless NoCs [8] based on the Ultra-Wide-Band technology [9]. These figures depend not only on application characteristics but also on design parameters including NoC topology, Routing Algorithm (RA) and scheduling policy [10]–[12]. As highlighted in many works, 2D mesh and 2D mesh-like topologies are well suited for tile-based ASIC implementation (e.g. [13], [14]). However, recent works pointed out from different perspectives that several cases of practical interest have tight throughput and latency constraints. Significant examples are i) application-specific NoCs [15], where the NoC used to interconnect different IPs is tailored around the application, and ii) intra-IP NoCs [16], where very low complexity NoC structures are employed to interconnect processing elements inside an IP. Minimizing the number of clock cycles spent to send a message from the source node to the destination node is a fundamental aspect. Thus, the maximum distance between two nodes, i.e. the diameter of the topology, plays a significant role in reducing the delivery time. Indeed, not only efficient RAs but also logarithmic diameter topologies, such as de-Bruijn [17] and Kautz [18] ones, have attracted the attention of some researchers [19]–[24]. Moreover, in [25] an optimum VLSI layout for de-Bruijn graphs is proposed to make VLSI implementation feasible. Furthermore, de-Bruijn and Kautz topologies have several interesting properties including self-routing [26]. This property has been exploited in [27] to derive a shortest-path RA to connect any pair of nodes. To the best of our knowledge, the implementation of this RA has not been addressed yet in the open literature. Besides, in order to achieve scalable and reliable interconnection networks, distributed routing must be employed. A flexible solution to support most RAs and topologies relies on routing-tables. Unfortunately, this approach does not scale in terms of latency and area. To overcome this limitation Logic-Based-Distributed-Routing (LBDR) [28] and its improved versions [14], [29] were proposed for mesh-derived topologies[1].

Inspired by the LBDR approach, this work shows that the shortest-path RA described in [27] can be implemented in a distributed fashion leading to lower complexity than the routing-table-based approach. The proposed solution is exploited to reduce the complexity of the intra-IP-NoC-based architecture proposed in [31]. It is worth pointing out, for the sake of clarity, that the RA proposed in [27] is optimal and that this work is focused on the efficient hardware implementation of the shortest-path RA.

The rest of the paper is organized as follows. Section II presents logarithmic topologies and summarizes the main characteristics that can be exploited in NoC design. In Section III the optimal shortest-path RA presented in [27] is revised and in Section IV a simple circuit to implement it is proposed. The proposed circuit is exploited in Section V to reduce the complexity of the intra-IP-NoC based architecture proposed in [31]. Finally, in Section VI conclusions are drawn.

II. Logarithmic diameter topologies

De-Bruijn and Kautz topologies are obtained by building directed graphs according to the following definitions.

Definition 1. *A de-Bruijn sequence is an array of q elements, where each element is taken from an alphabet $\mathcal{A}$ with l symbols.*

Thus, a de-Bruijn graph is made of nodes labeled with de-Bruijn sequences. Let $\mathbf{v} = v_{q-1}, \ldots, v_0$ and $\mathbf{w} = w_{q-1}, \ldots, w_0$ be the labels (expressed as de-Bruijn sequences)

[1]For a survey on LBDR the reader can refer to [30]

of two nodes v and w in a de-Bruijn graph, where v and w are decimal numbers and $v_i, w_i \in \mathcal{A}$ with $0 \leq i \leq q-1$. There is an arc from node v to node w if $w_i = v_{i-1}$ for $1 \leq i \leq q-1$, that is $\mathbf{w}$ is obtained by left-shifting $\mathbf{v}$ and by placing in the rightmost position a symbol from $\mathcal{A}$. As a consequence, each node is connected to l nodes. Thus, the graph is regular with degree $D = l$ and the number of nodes is $P = l^q$ (Fig. 1 (a)). Unfortunately, de-Bruijn graphs have self-loops (one node connected to itself). Self loops can be avoided using Kautz graphs.

Definition 2. *A Kautz sequence is an array of q elements, where each element is taken from an alphabet $\mathcal{A}$ with l symbols avoiding sequences with equal symbols in consecutive positions.*

A Kautz graph is made of nodes labeled with Kautz sequences and there is an arc from node v to node w if $w_i = v_{i-1}$ for $1 \leq i \leq q-1$ (with $w_1 \neq w_0$). In other words there is an arc from node v to node w if $\mathbf{w}$ is obtained by left-shifting $\mathbf{v}$ and by placing in the rightmost position a symbol from $\mathcal{A}$, subject to the constraint that the result is a Kautz sequence. As a consequence, each node is connected to $l-1$ nodes so the graph is regular with degree $D = l-1$ and the number of nodes is $P = l \cdot (l-1)^{q-1}$ (Fig. 1 (b)).

Thus, in general, de-Bruijn and Kautz graphs for given P and D not always exist. To overcome this limitation generalized de-Bruijn and generalized Kautz graph have been proposed [32]–[34].

Definition 3. *A generalized de-Bruijn graph has an arc from node v to node w if (1) holds true:*

$$w = (D \cdot v + r) \bmod P \tag{1}$$

with $0 \leq v \leq P-1$ and $0 \leq r \leq D-1$ (Fig. 1 (c)).

Definition 4. *A generalized Kautz graph has an arc from node v to node w if (2) holds true:*

$$w = -(D \cdot v + r) \bmod P \tag{2}$$

with $0 \leq v \leq P-1$ and $1 \leq r \leq D$, or equivalently

$$w = [(D \cdot (P-1-v) + r] \bmod P \tag{3}$$

with $0 \leq v \leq P-1$ and $0 \leq r \leq D-1$ (Fig. 1 (d).

Unfortunately, both generalized de-Bruijn and generalized Kautz graphs have self loops. However, as shown in [26] the number of self loops s in generalized de-Bruijn and generalized Kautz graphs are $D \leq s \leq 2 \cdot D-2$ and $0 \leq s \leq D$ respectively. Besides, in generalized Kautz graphs $s = 0$ is achieved when

$$P \bmod (D+1) = 0, \tag{4}$$

that is $D+1$ is a divider of P [26]. Moreover, as detailed in [33], [34], generalized de-Bruijn and generalized Kautz graphs have logarithmic diameter. In particular, the diameter of generalized de-Bruijn and generalized Kautz graphs are $\lceil \log_D(P) \rceil$ and $\lceil \log_D(P \cdot (D-1) + D) \rceil - 1$, where the latter one is the lower bound for directed Moore graphs [35] and $\lceil x \rceil$ is the minimum integer not smaller than x. As a consequence, generalized Kautz graphs not only have less self-loops than generalized de-Bruijn ones but they are optimal from the diameter size point of view. According to [26] the number of self-loops in generalized Kautz topologies is $s = b \cdot \lfloor D/b \rfloor$, where $b = \gcd(P, D+1)$ and $\lfloor x \rfloor$ is the maximum integer not larger than x.

Algorithm 1 Optimal shortest-path RA for generalized Kautz topologies

```
if v = w then
    z ← 0
else
    z ← 1
    found ← FALSE
    while NOT found do
        if z is odd then
            g ← [w + (v + 1) · D^z] mod P
        else
            g ← [w − v · D^z] mod P
        end if
        if g < D^z then
            found ← TRUE
            for n = 1 to z do
                if n is odd then
                    t_n ← D − 1 − g_n
                else
                    t_n ← g_n
                end if
            end for
        else
            z = z + 1
        end if
    end while
end if
```

III. Optimal shortest path routing for generalized Kautz topologies

In [26], the authors proved that generalized de-Bruijn and generalized Kautz graphs have self-routing property and that there exist a path of length $m = \lceil \log_D(P) \rceil$ that connects any pair of nodes. This result has been extended in [27] where the self-routing property is exploited to derive a shortest path RA to connect any pair of nodes. For each pair of nodes the RA computes a tag t that is used to build the shortest path. The tag is then converted into an array of D-ary elements and the number of elements in the array ($z \leq m$) is the length of the path. Algorithm 1 shows the steps to compute t, where v and w are the source and the destination node respectively. Once the tag has been computed it is used to derive the routing path as follows:

$$y_{n-1} = [D \cdot (P-1-y_n) + t_{n-1}] \bmod P \tag{5}$$

where $n = 1, \ldots, z$ and $y_z = v$ and $y_0 = w$.

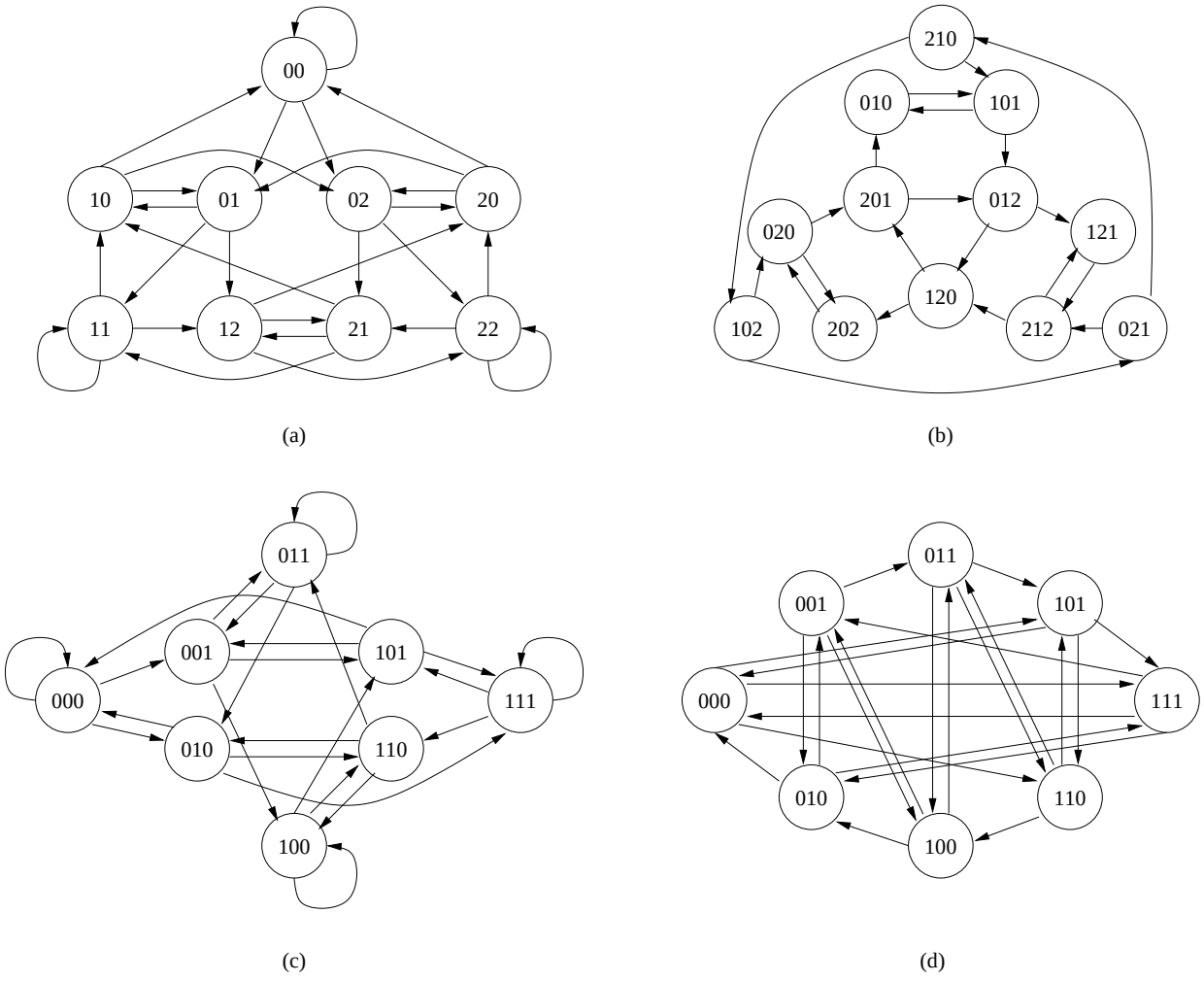

Fig. 1. Example of logarithmic diameter graphs: (a) $q = 2, l = 3$ ($D = 3, P = 9$) de-Bruijn graph, (b) $q = 3, l = 3$ ($D = 2, P = 12$) Kautz graph, (c) $D = 3, P = 8$ generalized de-Bruijn graph, (d) $D = 3, P = 8$ generalized Kautz graph.

IV. Digital circuit for distributed optimal shortest path routing

As it can be inferred from the description in Section III, the shortest path from node v to node w can be computed resorting to the recurrence in (5). Since the recurrence order is one, there is no need to keep trace of the previous steps. In other words, when a packet comes to node y_n, only w and y_n are required to compute y_{n-1}. As a consequence, Algorithm 1 can be reused simply replacing v with y_n and the *for* loop at line 14 in Algorithm 1 reduces to the computation of t_{z-1}. Moreover, (5) can be rewritten as

$$y_{n-1} = (\chi_n + t_{n-1}) \bmod P \tag{6}$$

where $\chi_n = [D \cdot (P-1-y_n)] \bmod P$ is only a function of y_n and so it can be precalculated and stored into each node. On the other hand, the term t_{n-1} depends on w and y_n (Algorithm 1) so it can be implemented with some constants and few logic as shown in Fig. 2 and detailed in the following paragraphs.

The proposed circuit to implement the shortest path routing is made of two parts: the first one computes g, the second one represents g as D-ary array, whose elements are g_n, and implements (6). The first part of the circuit is obtained observing that the *while* loop at line 6 in Algorithm 1 is devoted to find both z, i.e. the length of the shortest path from node v to node w, and g, required to compute the tag t. However, since $z \leq m = \lceil \log_D(P) \rceil$ the search of z can be performed in parallel. Indeed, the proposed circuit computes m candidates for g, where the i-th candidate ($g^{(i)}$), according to lines 8 and 10 of Algorithm 1, is

$$g^{(i)} = \begin{cases} \left[w + (v+1) \cdot D^i\right] \bmod P & \text{if i is odd} \\ \left[w - v \cdot D^i\right] \bmod P & \text{otherwise} \end{cases} \tag{7}$$

Then, each candidate is compared to a threshold (D^i) and a priority encoder selects $g = g^{(z)} = \min_i \{g^{(i)} < D^i\}$. As it can be observed, (7) can be rewritten as $g^{(i)} = (w + k_i) \bmod P$ where $k_i = [(v+1) \cdot D^i] \bmod P$ when i is odd and $k_i = P - [(v \cdot D^i) \bmod P]$ otherwise. In both cases k_i depends only on v so it can be precalculated and stored in each node. The $\bmod P$ operation is implemented by a subtracter and a multiplexer by checking the sign of $w + k_i - P$: if the sign is

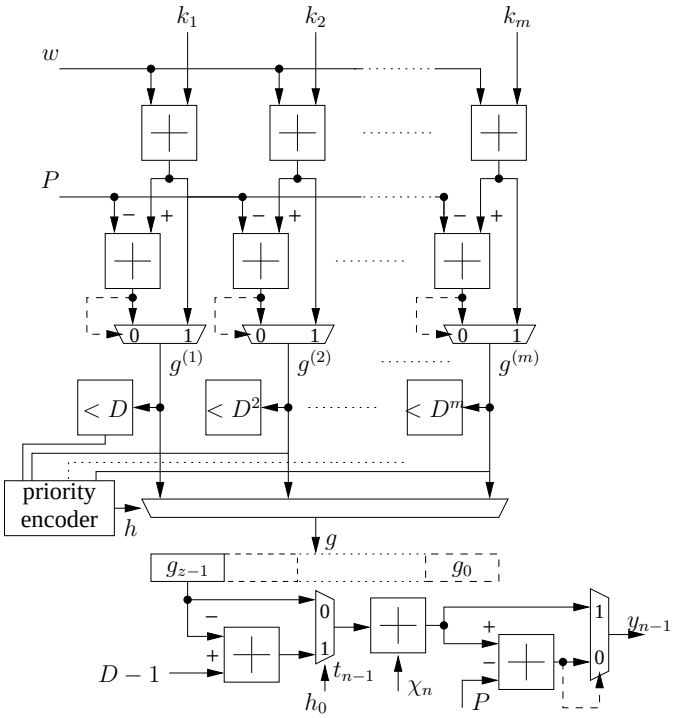

Fig. 2. General architecture to implement the RA described in Algorithm 1

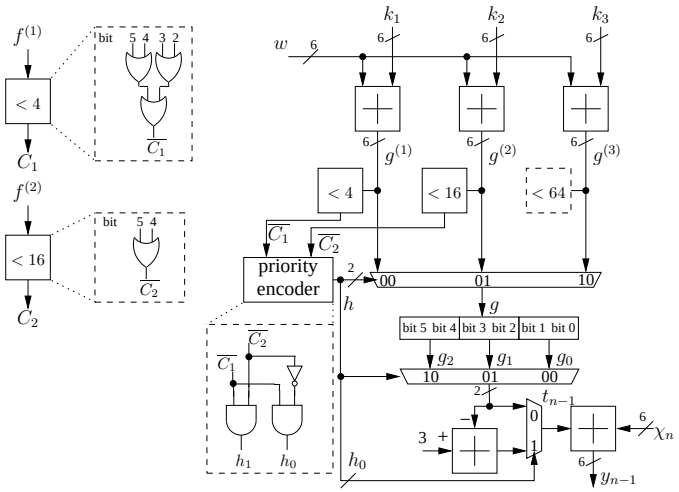

Fig. 3. Architecture of the RA in Algorithm 1 when $D = 4$ and $P = 64$

positive then $g^{(i)} = w + k_i - P$, otherwise $g^{(i)} = w + k_i$.

The second part of the circuit relies on representing g as an array of at most m D-ary elements. The element at position $h = z-1$, that is g_{z-1}, is used to compute t_{z-1} as in lines 16 and 18 of Algorithm 1. The least significant bit of h, referred to as h_0, selects the correct value for t_{z-1}, that is $D-1-g_{z-1}$ if h is odd, g_{z-1} otherwise. Finally, t_{z-1} is employed to obtain the next node as in (6), namely it is added with χ_n and the $\bmod P$ operation is implemented by a subtracter and a multiplexer by checking the sign of $(\chi_n + t_{n-1}) - P$: if the sign is positive then $y_{n-1} = (\chi_n + t_{n-1}) - P$, otherwise $y_{n-1} = \chi_n + t_{n-1}$.

As it can be inferred from Algorithm 1 and Fig. 2, significant complexity reduction can be achieved if both D and P are powers of two. Indeed, all modulo P operations and the generation of D-ary elements are implemented with no hardware cost exploiting binary representation. On the other hand, being both D and P powers of two (4) cannot be satisfied and so some self-loops have to be tolerated. As an example in Fig. 3 the architecture of the RA for $D = 4$ and $P = 64$ is shown, where modulo P operations are obtained by letting the data wrap (when an overflow occurs) and both comparators and priority encoder are implemented with few logic gates.

V. Case of study: intra-IP NoC for a turbo/LDPC decoder architecture

As summarized in the introduction, application specific and intra-IP NoCs [15], [16] have tight throughput and latency constraints with low complexity requirements. Thus, logarithmic diameter topologies and shortest-path RA are ideal candidates for such applications. Significant examples are in the field of multimedia signal processing [36], telecommunications [10] and channel code decoding, such as turbo codes [37], Low-Density-Parity-Check (LDPC) codes [19] or both [31]. It is worth noting that application specific and intra-IP NoCs also have particular traffic patterns and packet structures. Tailoring the NoC topology around the specific characteristics of the traffic patterns is a viable solution to both reduce the NoC complexity and increase the NoC throughput. As an example, irregular NoCs have been successfully used for video processing applications [38], [39]. On the contrary, in turbo and LDPC code decoders the traffic is almost uniform both in time and space [40]. Thus, regular topologies able to minimize the maximum distance between two nodes (as logarithmic diameter topologies) are a natural choice. Moreover, in turbo and LDPC code decoders the traffic is deterministic as it is imposed by the interleaver and parity check matrix respectively. As a consequence, all the routing and scheduling information can be precomputed by the means of a simulator, e.g. [41], and so the sequence of commands for each routing element can be precalculated as well and stored in memories. However, this approach requires a large amount of memory becoming almost impractical in some real cases [42]. Thus, routing and scheduling information have to be computed on-the-fly with limited complexity and delay. For this class of applications a low complexity circuit for shortest-path routing, as the one presented in Section IV, is a scalable alternative to routing-tables. Also the structure of the packet is usually simple and tailored around the application. As an example in [31] packets are made of a header, containing the destination node, and a payload containing one datum the memory location where the datum will be stored at destination node. Thus, the packet is moved in the network as a whole without resorting to flits.

The node architecture in these applications is made of a Processing Element (PE), devoted to computation, and a Routing Element (RE) to send and receive data. The RE relies on a simple structure made of a $(D + 1) \times (D + 1)$

crossbar, $D+1$ input FIFOs and output registers, a RA and a scheduler (see Fig. 4). As discussed in [41] shortest-path

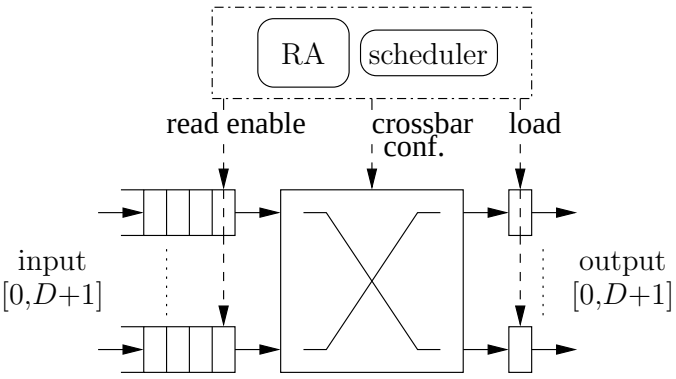

Fig. 4. Architecture of a node for intra-IP NoCs

RAs coupled with Round-Robin (RR) and Longest-Queue-First (LQF) scheduling policies are the most suited solutions for channel code decoding applications. Let us take the intra-IP NoC-based multi-mode turbo/LDPC decoder architecture presented in [31] as a case of study. This architecture relies on a $D=3$, $P=22$ nodes generalized Kautz topology where the shortest-path RA is distributed and performed via routing tables. In the following, the architecture presented in [31] is extended to the case $D=4$, $P=32$ and it will be referred to as Routing-Table-based NoC (RT-NoC) architecture. The RT-NoC architecture is then modified replacing the routing tables with the circuit-based RA described in Section IV. This new architecture will be referred to as RA-NoC. Both architectures have been described in VHDL and implemented on a CMOS 90 nm standard cell technology for a 200 MHz target clock frequency [43]. As an example the post place and route layout of the proposed intra-IP RA-NoC for $D=4$, $P=32$ is shown in Fig. 5. Stemming from the design in [31], the FIFOs in each node have been conservatively sized to eight locations to prevent deadlock.

Post place and route area and switching-activity-based power consumption are summarized for both architectures in Table I. As it can be observed the proposed RA-NoC features an area and a power consumption reduction of about 14% and 10% with respect to the RT-NoC.

Finally, in Fig. 6 the area of one routing-table-based RE is compared with the area of one circuit-based RE as a function of P in the same test conditions employed for the full decoders. As it can be observed, as long as P increases the advantage of the circuit-based RE becomes larger, e.g. when $P=64$ the area reduction is about 20% with respect to one table-based RE.

VI. Conclusion

In this work a circuit to implement the optimal shortest-path RA proposed in [27] for generalized Kautz topologies is presented. The proposed circuit features lower complexity and power consumption than routing-table-based shortest-path RAs. Experimental results on the NoC-based design proposed in [31] show that the complexity of the NoC is reduced by about 14% and the power consumption by about 10%.

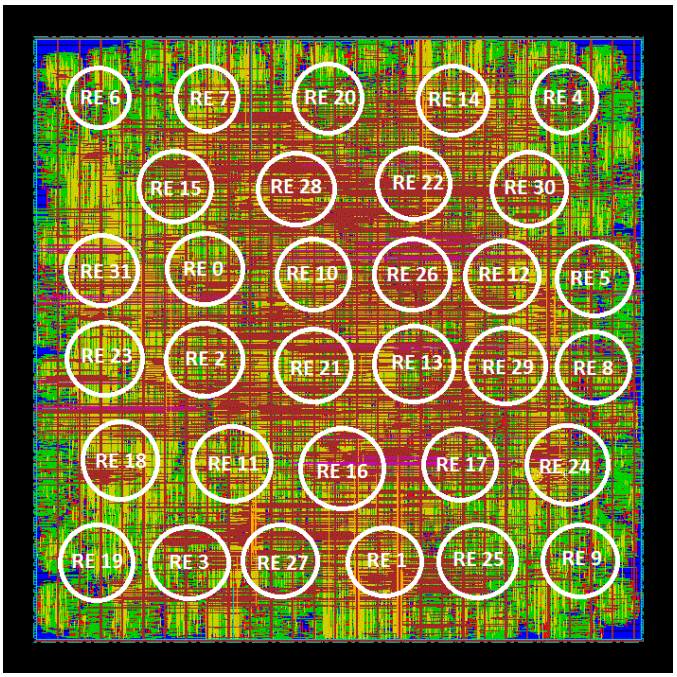

Fig. 5. Post place and route layout of the proposed intra-IP RA-NoC for $D=4$, $P=32$

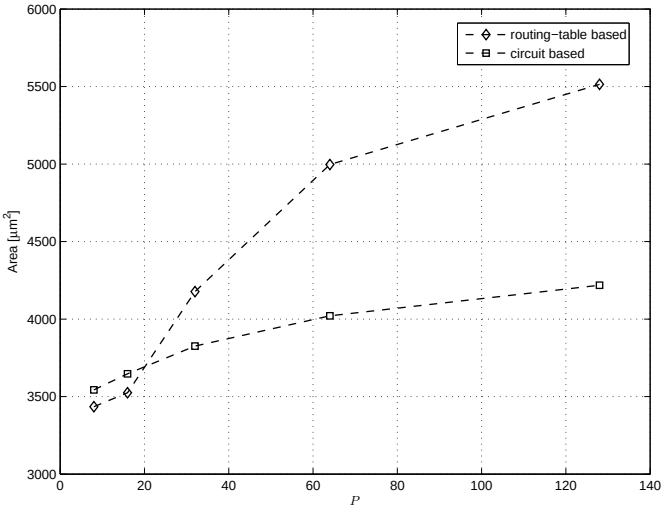

Fig. 6. Area of one routing-table-based RE and one circuit-based RE as a function of P.

TABLE I
AREA (A) AND POWER (POW) CONSUMPTION COMPARISON BETWEEN RT-NOC AND RA-NOC. CMOS 90 NM STANDARD CELL TECHNOLOGY, CLOCK FREQUENCY 200 MHZ, 8-ELEMENT FIFOS, $P=32$, $D=4$.

	A		Pow	
	[mm^2]	% reduction	[mW]	% reduction
RT-NoC	0.807	-	75.1	-
RA-NoC	0.691	-14.4%	67.2	-10.5%

ACKNOWLEDGMENT

This work has been partially funded by the Newcom# project.

REFERENCES

[1] P. Guerrier and A. Greiner, "A generic architecture for on-chip packet-switched interconnections," in *Design, Automation and Test in Europe Conference and Exhibition*, 2000, pp. 250–256.

[2] W. J. Dally and B. Towels, "Route packets, not wires: On-chip interconnection networks," in *Design Automation Conference*, 2001, pp. 684–689.

[3] S. Kumar, A. Jantsch, J. P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja, and A. Hemani, "A network on chip architecture and design methodology," in *IEEE Computer Society Annual Symposium on VLSI*, 2002, pp. 105–112.

[4] L. Benini and G. D. Micheli, "Networks on chips: a new SoC paradigm," *IEEE Computer*, vol. 35, no. 1, pp. 70–78, Jan 2002.

[5] S. V. Tota, M. R. Casu, M. R. Roch, L. Rostagno, and M. Zamboni, "MEDEA: A hybrid shared-memory/message-passing multiprocessor NoC-based architecture," in *Design, Automation and Test in Europe Conference and Exhibition*, 2010, pp. 45–50.

[6] W. Xiohang, M. Palesi, Y. Mei, J. Yingtao, M. C. Huang, and L. Peng, "Low latency and energy efficient multicasting schemes for 3D NoC-based SoCs," in *IEEE International Conference on VLSI and System-on-Chip*, 2011, pp. 337–342.

[7] M. Daneshtalab, M. Ebrahimi, J. Plosila, and H. Tenhunen, "CARS: Congestion-aware request scheduler for network interfaces in NoC-based manycore systems," in *Design, Automation & Test in Europe Conference & Exhibition*, 2013, pp. 1048–1051.

[8] D. Zhao and Y. Wang, "SD-MAC: design and synthesis of a hardware-efficient collision-free QoS-aware MAC protocol for wireless Network-on-Chip," *IEEE Transactions on Computers*, vol. 57, no. 9, pp. 1230–1245, Sep 2008.

[9] M. Crepaldi and D. Demarchi, "A 130-nm CMOS 0.007 mm2 ring-oscillator-based self-calibrating IR-UWB transmitter using an asynchronous logic duty-cycled PLL," *IEEE Transactions on Circuits and Systems II*, vol. 60, no. 5, pp. 237–241, May 2013.

[10] F. Clermidy, N. Cassiau, N. Coste, D. Dutoit, M. Fantini, D. Ktenas, R. Lemaire, and L. Stefanizzi, "Reconfiguration of a 3GPP-LTE telecommunication application on a 22-core NoC-based system-on-chip," in *IEEE/ACM International Symposium on Networks on Chip*, 2011, pp. 261–262.

[11] K. Chen, S. Lin, H. Hung, and A. Wu, "Topology-aware adaptive routing for nonstationary irregular mesh in throttled 3D NoC systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 10, pp. 2109–2120, Oct 2013.

[12] K. S. M. Li, "CusNoC: Fast full-chip custom NoC generation," *IEEE Transactions on VLSI systems*, vol. 21, no. 4, pp. 692–705, Apr 2013.

[13] S. Murali, D. Atienza, P. Meloni, S. Carta, L. Benini, G. D. Micheli, and L. Raffo, "Synthesis of predictable networks-on-chip-based interconnect architectures for chip multiprocessors," *IEEE Transactions on VLSI systems*, vol. 15, no. 8, pp. 869–880, Aug 2007.

[14] S. Rodrigo, J. Flich, A. Roca, S. Medardoni, D. Bertozzi, J. Camacho, F. Silla, and J. Duato, "Cost-efficient on-chip routing implementations for CMP and MPSoC systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 4, pp. 534–547, Apr 2011.

[15] L. Benini, "Application specific NoC design," in *Design, Automation and Test in Europe Conference and Exhibition*, 2006, pp. 1330–1335.

[16] F. Vacca, H. Moussa, A. Baghdadi, and G. Masera, "Flexible architectures for LDPC decoders based on network on chip paradigm," in *Euromicro Conference on Digital System Design*, 2009, pp. 582–589.

[17] N. G. de Bruijn, "A combinatorial problem," *Koninklijke Nederlandse Akademie v. Wetenschappen*, vol. 49, pp. 758–764, 1946.

[18] W. H. Kautz, "Design of optimal interconnection networks for multiproces- sors," *Architecture and Design of Digital Computers*, pp. 249–272, 1969.

[19] H. Moussa, A. Baghdadi, and M. Jezequel, "Binary de Bruijn on-chip network for a flexible multiprocessor LDPC decoder," in *Design Automation Conference*, 2008, pp. 429–434.

[20] M. Hosseinabady, M. R. Kakoee, J. Mathew, and D. K. Pradhan, "Low latency and energy efficient scalable architecture for massive NoCs using generalized de Bruijn graph," *IEEE Transactions on VLSI systems*, vol. 19, no. 8, pp. 1469–1480, Aug 2011.

[21] R. Sabbaghi-Nadooshan, "Kautz mesh topology for on-chip networks," *Journal of Computing*, vol. 3, no. 3, pp. 33–40, Feb 2011.

[22] Y. Chen, J. Hu, X. Ling, and T. Huang, "A novel 3D NoC architecture based on De Bruijn graph," *Elsevier Computers and Electrical Engineering*, vol. 38, no. 3, pp. 801–810, May 2012.

[23] C. Y. Tsai, Y. J. Lee, C. T. Chen, and L. G. Chen, "A 1.0TOPS/W 36-core neocortical computing processor with 2.3tb/s Kautz NoC for universal visual recognition," in *IEEE International Solid-State Circuits Conference*, 2012, pp. 480–482.

[24] F. Stas, A. K. Lusala, J. D. Legat, and D. Bol, "Investigation of the routing algorithm in a De Bruijn-based NoC for low-power applications," in *IEEE Faible Tension Faible Consommation*, 2013, pp. 1–4.

[25] M. R. Samatham and D. K. Pradhan, "The De Bruijn multiprocessor network: A versatile parallel processing and sorting network for VLSI," *IEEE Transactions on Computers*, vol. 38, no. 4, pp. 567–581, Apr 1989.

[26] D. Z. Du and F. K. Hwang, "Generalized de Bruijn digraphs," *Networks*, vol. 18, pp. 27–38, 1988.

[27] G. Liu and K. Y. Lee, "Optimal routing algorithms for generalized de Bruijn digraphs," in *International Conference on Parallel Processing*, 1993, pp. 167–174.

[28] J. Flich and J. Duato, "Logic-based distributed routing for NoCs," *IEEE Computer Architecture Letters*, vol. 7, no. 1, pp. 13–16, Jan 2008.

[29] S. Rodrigo, S. Medardoni, J. Flich, D. Bertozzi, and J. Duato, "Efficient implementation of distributed routing algorithms for NoCs," *IET Computers and Digital Techniques*, vol. 3, no. 5, pp. 460–475, Sep 2009.

[30] N. Choudhary and C. M. Samota, "A survey of logic based distributed routing for on-chip interconnection networks," *International Journal of Soft Computing and Engineering*, vol. 3, no. 2, pp. 233–237, May 2013.

[31] C. Condo, M. Martina, and G. Masera, "VLSI implementation of a multi-mode turbo/LDPC decoder architecture," *IEEE Transactions on Circuits and Systems - I*, vol. 60, no. 6, pp. 1441–1454, Jun 2013.

[32] S. M. Reddy, D. K. Pradhan, and J. G. Kuhl, "Direct graphs with minimal and maximal connectivity," School of Engineering, Oakland University, Tech. Rep., 1980.

[33] M. Imase and M. Itoh, "Design to minimize diameter on building-block network," *IEEE Transactions on Computers*, vol. 30, no. 6, pp. 439–442, Jun 1981.

[34] ——, "A design for directed graphs with minimum diameter," *IEEE Transactions on Computers*, vol. 32, no. 8, pp. 782–784, Aug 1983.

[35] W. G. Bridges, "On the impossibility of directed moore graphs," *Journal of Combinatorial Theory*, vol. 29, no. 3, pp. 339–341, 1980.

[36] S. Saponara and L. Fanucci, "Homogeneous and heterogeneous MPSoC architectures with Network-On-Chip connectivity for low-power and real-time multimedia signal processing," *VLSI Design*, vol. 2012, 2012.

[37] G. Wang, Y. Sun, J. R. Cavallaro, and Y. Guo, "High-throughput contention-free concurrent interleaver architecture for multi-standard turbo decoder," in *IEEE International Conference on Application-Specific Systems, Architectures and Processors*, 2011, pp. 113–121.

[38] 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 Conference and Exhibition*, 2004, pp. 884–889.

[39] S. Saponara, M. Martina, M. Casula, L. Fanucci, and G. Masera, "Motion estimation and CABAC VLSI co-processors for real-time high-quality H.264/AVC video coding," *Elsevier Microprocessors and Microsystems*, vol. 34, no. 7-8, pp. 316–328, Nov 2010.

[40] C. Neeb, M. J. Thul, and N. Wehn, "Network-on-chip-centric approach to interleaving in high throughput channel decoders," in *IEEE International Symposium on Circuits and Systems*, 2005, pp. 1766–1769.

[41] M. Martina and G. Masera, "Turbo NOC: A framework for the design of network-on-chip-based turbo decoder architectures," *IEEE Transactions on Circuits and Systems - I*, vol. 57, no. 10, pp. 2776–2789, Oct 2010.

[42] M. Martina, G. Masera, H. Moussa, and A. Baghdadi, "On chip interconnects for multiprocessor turbo decoding architectures," *Elsevier Microprocessors and Microsystems*, vol. 35, no. 2, pp. 167–181, Mar 2011.

[43] A. Pulimeno, M. Graziano, and G. Piccinini, "UDSM trends comparison: From technology roadmap to UltraSparc Niagara2," *IEEE Trans. on VLSI*, vol. 20, no. 7, pp. 1341–1346, Jul 10.1109/TVLSI.2011.2148183.