



HAL
open science

Binary de Bruijn interconnection network for a flexible LDPC/turbo decoder

Hazem Moussa, Amer Baghdadi, Michel Jezequel

► **To cite this version:**

Hazem Moussa, Amer Baghdadi, Michel Jezequel. Binary de Bruijn interconnection network for a flexible LDPC/turbo decoder. IEEE International Symposium on Circuits and Systems (ISCAS 2008), May 2008, Seattle, WA, United States. pp.97-100, 10.1109/ISCAS.2008.4541363 . hal-02194923

HAL Id: hal-02194923

<https://hal.science/hal-02194923>

Submitted on 26 Jul 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Binary de Bruijn Interconnection Network for a Flexible LDPC/Turbo Decoder

Hazem Moussa, Amer Baghdadi, Michel Jézéquel
Electronics Department, Institut TELECOM/TELECOM Bretagne
Technopôle Brest Iroise, 29238 Brest, France
firstname.lastname@telecom-bretagne.eu

Abstract— This paper proposes a novel on-chip interconnection network adapted to a flexible multiprocessor LDPC/turbo decoder and based on the de Bruijn network. The main characteristics of this network –including its logarithmic diameter, scalable aggregate bandwidth, and optimized routing technique– allow it to efficiently support the communication-intensive nature of the two decoding techniques. We present a detailed hardware implementation of the routers and the network interfaces as well as the packet format and the routing algorithm. In order to evaluate the performance of the proposed network, a generic RTL VHDL description has been developed and synthesized with ST CMOS 0.18 μm technology. The flexibility and the scalability of this on-chip communication network enable it to be used in the emerging multi-code applications and standards. In addition, the results obtained for a 16-processor network demonstrate a major aggregate bandwidth of 296 Gbps with a relative small area of 3.56 mm^2 .

I. INTRODUCTION

To reduce the error rate in digital communications, with a lower signal-to-noise ratio (closer to the Shannon limit), iterative decoding processing algorithms have recently emerged [1][2] in order to improve the error correction of the transmitted data. Indeed, thanks to appropriate channel coding (parallel recursive systematic convolutional codes for turbo codes and linear block codes with a sparse parity check matrix for LDPC codes), the noisy received information can be recovered by the decoding process.

Although these two kinds of error correcting codes have similar performance, they differ in certain cases where sometimes turbo codes are better than LDPC codes or vice versa [3]. This is why in many telecommunication applications, it may be desirable to have a flexible system able to support a variety of error correction standards or a variety of codes [4][5], and in particular both LDPC and turbo codes.

A few multi-code implementations of channel decoders have recently been proposed to provide an answer to the need for flexibility.

In [5], the implementation of a dual mode decoder for 3GPP2 was described, supporting the Viterbi decoding of a 256 state convolutional code and the MAP decoding of a turbo code with many shared hardware components.

The authors in [6] propose a generic approach to support all possible interleaving for any type of LDPC or turbo code. This approach presents a mapping algorithm avoiding the conflicting memory accesses generated by the interleaving laws of the codes. Indeed, an appropriate memory addressing scheme, based on spatial and temporal permutations, prevents conflicts occurring.

To the best of our knowledge, none of the existing works has presented a flexible architecture supporting LDPC and turbo codes except the contribution proposed in [7]. In this work, the authors target a reconfigurable decoder for a wireless system made of a number of configurable Add-Compare-Select decoders that exchange messages through multiple memory banks. However, the presented architecture requires a huge memory capacity in order to avoid the conflicting accesses. Besides, a complex interconnecting architecture, not presented, is required to achieve full flexibility with limited throughput penalty.

Finally, there is a wide and growing interest in realizing channel decoders supporting more than one code and many researches are being recently conducting in this direction.

On the other hand, the new emerging standards and applications require high performance in terms of throughput. For this purpose, the parallelization of the decoding algorithms has been widely investigated where several implementations, based on multiprocessor systems, have been proposed. Furthermore, the implementation issues of these multiprocessor architectures mainly come from the communication structure between the processing elements inside the decoder. The huge quantity of data (extrinsic information) exchanged by the processing elements requires an efficient interconnection structure, which represents a major challenge for flexible decoder implementation.

In this paper, we propose and evaluate an adapted de Bruijn network (with routers and network interfaces) as an efficient interconnection network of a flexible parallel LDPC/turbo decoder. The main features of this network architecture are its scalability enabling seamless trade-off between hardware complexity and available bandwidth for LDPC/turbo decoding and its flexibility allowing any interleaving communication profile to be implemented.

The remainder of the paper is organized as follows. Section II highlights the communications involved in parallel LDPC/turbo decoding. Then section III details the implementation

architectures of the proposed interconnection network, including topology, routing algorithm, packet format, routers, and network interfaces. Finally, section IV presents the synthesis results and provides performance analysis and comparison. The paper is concluded in section V.

II. COMMUNICATION REQUIREMENTS OF FLEXIBLE PARALLEL ITERATIVE CHANNEL DECODER

In parallel turbo decoding, extrinsic information is iteratively and concurrently exchanged between component decoders. For each iteration, these exchanges become more and more massive with decoder level parallelism [8].

In addition, the communication traffic profile, which depends on the turbo code interleaving rule, can be considered as random if we target flexibility in order to support multi-mode and multi-standard features. In this context, hardware implementations of parallel turbo decoders require the integration of a complex interconnection structure supporting the intensive interleaved memory accesses.

The two component decoders in Fig. 1 concurrently process the iterated frame, and for both, multiple processors can be designed for the parallel processing of frame sub-blocks.

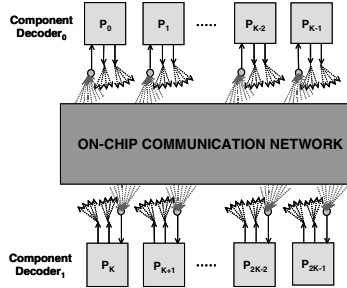


Figure 1. Extrinsic information exchanges in a parallel turbo decoder

Concerning LDPC codes, their decoding process is directly related to the structure of the parity check matrices. Indeed, the codes can also be described by a bipartite graph, called a Tanner graph [2], made up of two types of nodes: variable nodes (or bit nodes) (VNs) corresponding to the bits of a codeword and check nodes (CNs) corresponding to the parity constraints (an edge in the graph represents a one in the parity check matrix).

Therefore, two categories of processors are necessary: VN and CN Processors (VNP and CNP). Thus, the architecture of a decoder (Fig. 2) can be defined as being a system made up of VNPs, CNPs, and an interconnection structure allowing the exchange of messages between processors. Mapping the Tanner graph onto this architecture involves: (1) associating one or more VN (resp. CN) to one VNP (resp. CNP) and (2) implementing Tanner graph edges through a corresponding message addressing in the interconnection structure.

However, similarly to parallel turbo decoders, partially-parallel code-independent LDPC decoders, which offer the best compromise in terms of area, throughput, and flexibility [9], introduce a problem of conflicting accesses to the memories containing the exchanged messages.

In addition, the computations being relatively simple, implementation issues mainly come from the communication structure between the VNPs and CNPs. Indeed, the

communications rapidly become intensive because they depend on the number of nodes, the node degrees and the number of iterations.

In this situation, a flexible on-chip interconnection network must be designed with the aim of fully exploiting the parallelism of the LDPC/turbo decoder architecture by reducing the message latency, alleviating the memory conflicts and efficiently routing any permutation from the network input ports to its output ports.

However, even if several efficient network architectures have already been proposed to solve the communication problem inside flexible LDPC decoders [10] or flexible parallel turbo decoders [11], none of them have been designed to handle the communications in a decoder supporting LDPC and turbo codes at the same time.

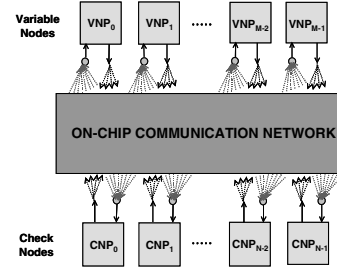


Figure 2. Extrinsic information exchanges in a partially-parallel LDPC decoder

III. BINARY DE BRUIJN NETWORK

The de Bruijn network, based on the de Bruijn graph [12], was chosen due to its appropriateness to support the communications of a multiprocessor LDPC/turbo decoder. In fact, the structure of the network allows any permutation to be routed efficiently thanks to the path diversity offered by the network. The conflicting memory accesses are alleviated because the conflicting packets can be deviated appropriately until they attain the targeted processor rather than being blocked or buffered.

In addition, the logarithmic diameter of the network ($\log_2(N)$) leads to small latencies and the number of routers is reduced because it is a direct network. Moreover, its recursive structure makes it highly scalable. Finally, in [13], it has been shown that the de Bruijn network is suitable for VLSI implementation. NASA developed an 8192-processor de Bruijn network used by a Viterbi decoder for the Galileo space mission to Jupiter [14].

A. Network Architecture and Processor Mapping

The de Bruijn graph consists of $N=k^m$ nodes, where each node is identified by a vector of size m in the k -ary number system. For our network, we chose the binary system. Thus, a node $v_m v_{m-1} \dots v_1$ has its two output edges connected with nodes $v_{m-1} v_{m-2} \dots v_1 \mathbf{0}$ and $v_{m-1} v_{m-2} \dots v_1 \mathbf{1}$.

As an example, Fig. 3 presents the schematic view of a complete de Bruijn network for $N=8$ processors ($m=3$). Each processor is connected to a network interface and each network interface connected to a router.

Concerning the mapping of the decoding elements on the network graph, we chose to simply allocate the first $N/2$ processors to the first component decoder (in the case of turbo codes) or to the VNPs (in the case of LDPC codes).

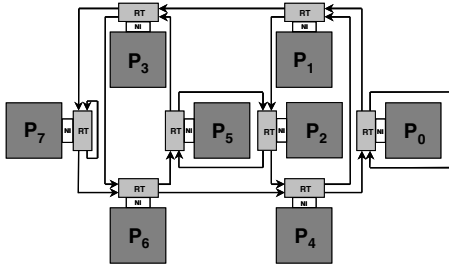


Figure 3. Schematic view of an 8-processor network with processors, routers, and network interfaces

B. Routing Algorithm and Packet Arbitration

The shortest path routing algorithm [15] was implemented thanks to its ability to reduce latencies. However, the algorithm being sequential and in order to bring the number of cycles back to one, we modified the algorithm of [15] by unrolling the “for” loop, and developed another method to compute the shortest path output. In order to reduce the size of routers by avoiding packet buffering when packets are in conflict, the routing algorithm was supplemented by a mechanism deflecting the “youngest” packet. This requires storing “age” information (Time Stamp TS) within the packet updated at each hop.

Moreover, in [11], a defined priority criterion is proposed to improve the packet arbitration inside a parallel turbo decoder. This criterion represents the contribution of the packets to the convergence of the iterative process. Consequently, we propose to apply this criterion also for iterative LDPC decoding with the effect of adding a priority field in the packet header.

It is worth to note that the de Bruijn network introduces some variable latencies which can be intrinsically tolerated in this case of iterative probabilistic decoding. Indeed, if an extrinsic information is used before being updated, the new value will be taken into account at the next iteration. Furthermore, the convergence of the decoding process is improved through the contribution priority field.

C. Packet Format

Packets are divided into two parts: the header and the payload. The first one is composed of the destination router identifier, the TS value, the priority field and a one-bit flag indicating whether intra symbol permutation, as defined in certain standards like WiMAX, is applied to this packet. The payload is made up of the destination memory write address and of the extrinsic information on 64 bits (the 8 LSB for LDPC codes and all the 64 bits for turbo codes). The latter corresponds to the reception of each possible symbol of a double binary turbo code (4 x 16 bits).

D. Router

The router architecture (Fig. 4) is divided into two units: operative and control units. The first one is composed of flip-flops, registers, and multiplexers implementing the packet switching and selecting the packets coming from routers and those coming from the processor. The control unit generates the multiplexer selection signals as well as the validity signals according to the routing algorithm and conflicting packet arbitration. Each port of a de Bruijn router has an associated control signal used to validate the data passing through the port. Moreover, there are two output signals corresponding to the read

request of the emission FIFOs contained in the network interface.

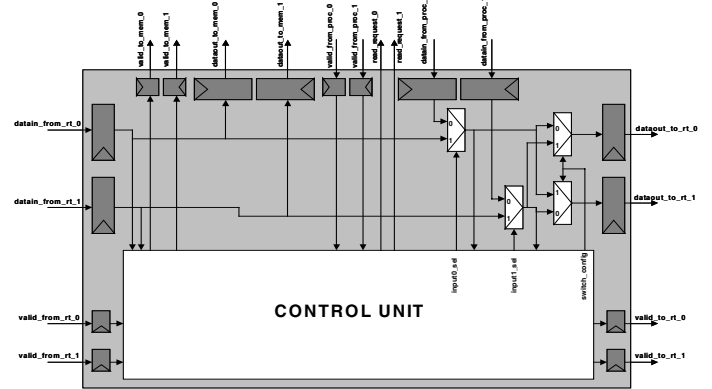


Figure 4. Router architecture

E. Network Interface

The network interface (Fig. 5) manages the packet flow in both directions, from the processor to the network and vice versa. The interface contains a reconfigurable table (RAM) for the construction of the packet header according to the parity check matrix of an LDPC code or according to the interleaving rule of a turbo code. Indeed, the permutation of the extrinsic information is done via the addressing of the packets. Thus, the de Bruijn network can support the LDPC/turbo codes of wireless standards like WiMAX by simply programming the content of the table accordingly. Moreover, packets are stored in two FIFOs at the input and output of the network interface. Finally, two multiplexers implement the intra symbol permutation.

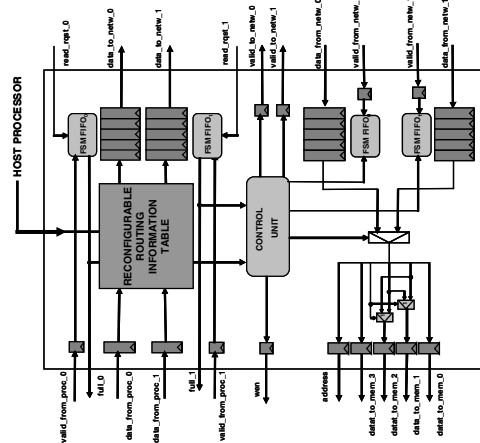


Figure 5. Network Interface architecture

IV. SYNTHESIS AND RESULTS ANALYSIS

For performance evaluation, we developed and synthesized a description in generic RTL VHDL of the complete system. The case study was the WiMAX standard for turbo decoding with a network of 16 processors. About LDPC decoding, we considered a 1024 bits code also mapped onto the same network. Thus, to estimate the area consumption and the aggregate bandwidth, we used ST CMOS 0.18 μm^2 technology. We indicate for each network its topology, the target technology, the maximum clock frequency in MHz, the area in mm^2 , and the estimated aggregate bandwidth in Gbps.

In Table I, we compare the results obtained with two networks recently proposed and based on the Butterfly and Beneš 2N-N topologies [11].

TABLE I. ASIC SYNTHESIS RESULTS OF BUTTERFLY, BENEŠ 2N-N AND DE BRUIJN NETWORKS FOR A 16-PROCESSOR TURBO DECODER

Topology	[11]		Proposed
	Butterfly	Beneš 2N-N	de Bruijn
Technology (μm)	0.18	0.18	0.18
Frequency (MHz)	302	416	266
Network Area (mm^2)	4.834	2.692	3.565
Aggregate Bw. (Gbps)	308	426	296

As the proposed routing of de Bruijn network is more complex (shortest path computation + deflection) than those of Butterfly and Beneš 2N-N networks, the resulting critical path is longer, and thus the maximum clock frequency and aggregate bandwidth are lower. On the other hand, the de Bruijn area cost is in between because at the same time, the network interfaces are bigger (due to the routing table) and the routers are smaller and fewer. However, the de Bruijn network resolves the limitations of the Butterfly and Beneš 2N-N solutions. Indeed, the conflicting packets deflection exempts the usage of buffers. Furthermore, there is no need to pre-compute the routing paths because the choice of the routers to cross is performed dynamically. To finish, the sufficient path diversity provided by the de Bruijn network, coupled to the deflection technique allow implementing optimizations based on application-specific priorities.

Next, we examine the results obtained with other previously proposed flexible LDPC decoder architectures (Table II) for a given codeword size of 1024 bits and based on an interconnection network.

TABLE II. COMPARISON ASIC SYNTHESIS RESULTS OF LDPC DECODERS

Topology	[10]	[16]	[9]	Proposed
	2D-mesh	MDN	Beneš	de Bruijn
Technology (μm)	0.16	0.18	0.13	0.18
Frequency (MHz)	500	200	245	266
Network Area (mm^2)	25	7.92	0.98	3.565
Aggregate Bw. (Gbps)	400	19.2	62.72	296

Frequency, area, and aggregate bandwidth being related to the target technology, it is not possible to compare directly and accurately the values of Table II. However, the de Bruijn network presents the best efficiency ratio between aggregate bandwidth and occupied area.

Regarding scalability, the recursive structure of the Beneš, the 2D-mesh, and the de Bruijn networks makes them highly scalable. On the other hand, the MDN network lacks scalability because the network structure is randomly built by a complex algorithm based on a constructive heuristic.

Although flexible, the Beneš network [9] requires a pre-calculation related to the code to configure the switches. The MDN [16] uses a general approach to network design with random graphs as basic topologies. This network lacks flexibility because all permutation patterns must be simulated prior to their use in order to determine buffer size at design-time. Finally, the 2D-mesh network [10] is designed for a flexible decoder. However, the described network can only manage codes with a maximum codeword size of 1024 bits. Consequently, the de Bruijn network, allowing to support any LDPC code, constitutes the most flexible solution.

V. CONCLUSION

In this paper, we have presented the architecture of a new on-chip interconnection network based on the de Bruijn graph and dedicated to a flexible multiprocessor LDPC/turbo decoder. We have detailed the topology and the network architecture, the packet format as well as the routers and the network interfaces. We have also proposed to use priorities specific to LDPC and turbo codes for the arbitration of conflicting packets. The unique trade-off between flexibility and performance (scalability, frequency, area) offered by the de Bruijn network demonstrates its efficiency in the context of a flexible parallel channel decoder supporting LDPC and turbo codes.

REFERENCES

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes," in Proc. International Conference on Communications (ICC'93), Geneva, Switzerland, pp. 1064-1070, May 1993.
- [2] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," IEEE Trans. Inf. Theory, vol. 45, pp.399-431, March 1999.
- [3] T. Lestable, E. Zimmerman, M-H Hamon, and S. Stiglmayr, "Block-LDPC codes vs duo-binary turbo-codes for european next generation wireless systems," IEEE Vehicular Technology Conference, September 2006.
- [4] K. K. Gunnam, G. S. Choi, W. Wang, E. Kim, and M. B. Yeary, "Decoding of Quasi-cyclic LDPC Codes Using an On-the-Fly Computation," in Proc. Asilomar Conference on Signals, Systems and Computers, pp. 1192-1199, October-November 2006.
- [5] C-C. Lin, Y-H. Shih, H-C. Chang, and C-Y. Lee, "A dual mode channel decoder for 3GPP2 mobile wireless communications," in Proc. Solid-State Circuits Conference, pp. 483-486, 21-23 September 2004.
- [6] A. Tarable, S. Benedetto, and G. Montorsi, "Mapping interleaving laws to parallel turbo and LDPC decoders architectures," IEEE Trans. Inform. Theory, vol. 50, no. 9, pp. 2002-2009, September 2004.
- [7] A. L. Chiu, and B. Xia, "Techniques for reconfigurable decoder for a wireless system," INTEL Corporation, US PATENT #WO/2007/002668, January 2007.
- [8] O. Muller, A. Baghdadi, and M. Jézéquel, "ASIP-Based Multiprocessor SoC Design for Simple and Double Binary Turbo Decoding," in Proc. of the conference on Design, Automation and Test in Europe, pp. 1330-1335, March 2006.
- [9] G. Masera, F. Quaglio, and F. Vacca, "Implementation of a flexible LDPC decoder," IEEE Transactions on Circuits and Systems, pp. 542-546, June 2007.
- [10] T. Theocharides, G. Link, N. Vijaykrishnan, and M. J. Irwin, "Implementing LDPC decoding on a network-on-chip," in Proc. of the International Conference on VLSI Design, pp. 134-137, January 2005.
- [11] H. Moussa, O. Muller, A. Baghdadi, and M. Jézéquel, "Butterfly and Benes-based on-chip communication networks for multiprocessor turbo decoding," in Proc. of the conference on Design, Automation and Test in Europe, pp. 654-659, April 2007.
- [12] N.G. de Bruijn, "A combinatorial problem," Koninklijke Nederlandse Akademie v. Wetenschappen, vol. 49, pp. 758-764, 1946.
- [13] 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, pp. 567-581, 1989.
- [14] R. A. Rowley, and B. Bose, "Fault-tolerant ring embedding in de Bruijn networks," IEEE Transactions on Computers, pp. 1480-1486, December 1993.
- [15] C. Francalanci, and P. Giacomazzi, "High-performance self-routing algorithm for multiprocessor systems with shuffle interconnections," IEEE Transactions on Parallel and Distributed Systems, pp. 38-50, January 2006.
- [16] F. Kienle, M. J. Thul, and N. Wehn, "Implementation issues of scalable LDPC-decoders," in Proc. 3rd International Symposium on Turbo Codes and Related Topics, Brest, France, September 2003.