Butterfly and Benes-Based on-Chip Communication Networks for Multiprocessor Turbo Decoding

Hazem Moussa, Olivier Muller, Amer Baghdadi, Michel Jézéquel

Electronics Department, ENST Bretagne, Technopôle Brest Iroise, 29238 Brest, France
{hazem.moussa, olivier.muller, amer.baghdadi, michel.jezequel}@enst-bretagne.fr

Abstract

Several research activities have recently emerged aiming to propose multiprocessor implementations in order to achieve flexible and high throughput parallel iterative decoding. Besides application algorithm optimizations and application-specific instruction-set processor design, the on-chip communication network constitutes a major issue in this application domain. In this paper, we propose to use multistage interconnection networks as on-chip communication networks for parallel turbo decoding. Adapted Benes and Butterfly networks are proposed with detailed hardware implementation of network interfaces, routers, and topologies. In addition, appropriate packet format and routing for interleaved/deinterleaved extrinsic information exchanges are proposed. The flexibility of these on-chip communication networks enables their use for all turbo code standards and constitutes a promising feature for their reuse for any similar interleaved/deinterleaved iterative communication profile.

1. Introduction

Versatile and parallel implementations are recently being widely investigated in the field of digital communication systems such as wireless communication, fiber-optic communication, and storage applications. This trend is driven mainly by continuously emerging new standards and applications in this field. In fact, besides various functional modes and diverse compulsory and/or optional techniques inside a single standard, these new applications often require multi-standard interoperability and severe performance requirements in terms of throughput and error rates.

To reduce the error rate in digital communications, with a lower signal-to-noise ratio (closer to Shannon limit), turbo (iterative) processing algorithms have recently emerged [1][2]. These algorithms, which originally concerned channel coding, are being currently reused over the whole digital communication system, like for equalization, demodulation, synchronization, and MIMO.

Algorithm parallelization of turbo decoding has been widely investigated these last years. Several implementations were also proposed. Some of these implementations succeeded in achieving high throughput for specific standards through tight optimizations and a highly dedicated architecture [3]. However, such implementations do not consider *flexibility*. Unlike these implementations, others include software and/or reconfigurable parts to achieve the required flexibility while achieving lower *throughput* [4]. Among those which tackled performance and flexibility constraints simultaneously we can cite the parallel multiprocessor implementation presented in [5][6]. In this work, an advanced heterogeneous communication network that optimizes data transfer and enables parallel turbo-decoding implementation was proposed. Authors present a ring based network on chip which only needs 2*N* connections and which is capable of incorporating up to eight processing units. This architecture was extended later to become a chordal ring (3*N*) allowing more individual nodes to be connected and then generalized to network design using random graphs as basic topologies. However, scalability and available bandwidth offered by all these architectures remain limited and routers complexity increases significantly.

In fact, in turbo decoding [1], extrinsic information is exchanged iteratively between component decoders. For each iteration, these exchanges become more and more massive with decoder level parallelism in order to achieve high throughput [7]. In addition, 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-modes and multi-standard features. In this context, hardware implementations of parallel turbo decoders require the integration of complex topology and routing resources supporting the intensive interleaved memory accesses.

In this paper, two multistage interconnection network architectures are proposed in order to handle the on-chip communications in multiprocessor parallel turbo decoders. The main feature of these network architectures (Butterfly and Benes based topologies) is their scalability enabling seamless tradeoff between hardware complexity and available bandwidth for turbo decoding. The rearrangeable quality of these topologies, allowing all possible permutations between its inputs and outputs to be realized, leads for full flexibility to implement any interleaved communication profile. Obviously, the two networks must be adapted for the turbo decoding application. Indeed, the interleaving (respectively deinterleaving) of the extrinsic information will be realized thanks to the destination address of the packet transporting the information and, according to the interleaving laws. Moreover, the packet payload definition also depends on the data type that must be transmitted, here extrinsic information.

The remainder of the paper is organized as follows. The following section characterizes the communications involved in parallel turbo decoding. Sections 3 and 4 detail the implementation architectures of the proposed multistage interconnection networks, including topology, routers, network interfaces, packet format, and routing algorithms. Section 5 presents the synthesis results and provides performance analysis and comparison of results. Finally,

978-3-9810801-2-4/DATE07 © 2007 EDAA

section 6 concludes the paper.

2. Parallel Turbo Decoding

With an aim of fulfilling the requirements of new standards, turbo decoders must be able to achieve high throughputs through parallelization. Parallelism techniques have been widely explored in this field. These techniques are classified in three levels of parallelism [7]: BCJR [8] metric level, BCJR-SISO decoder level, and Turbo-decoder level.

BCJR metric level parallelism, which achieves optimal area efficiency, has a limited parallelism degree and it can be fully exploited through the design of an SIMD application-specific instruction-set processor. On the other hand, turbo-decoder level parallelism is almost never used due to its expensive area cost and non improvement of decoding latency. So, further increasing in parallelism degree, while preserving area efficiency, should exploit the BCJR-SISO decoder level.

At this level, parallelism can be applied either on sub-blocks and/or on component decoders. In Figure 1, the two component decoders can process the iterated frame concurrently, and for both, multiple processors can be designed for parallel processing of frame sub-blocks. Here, the main issue concerns the iterative extensive data exchanges (*extrinsic information*) between the two components decoders. Besides the severe communication bandwidth requirements in high throughput turbo decoder context, the exchanged information should be interleaved or deinterleaved before transmission. Interleaving is a permutation that scrambles the data sequence of frames. Interleaving rule varies from one standard to another, and within a single standard different interleaving rules are proposed for different frame lengths and data rates. Thus, the requirement of a fully flexible on-chip communication network interconnecting the two component decoders implies its ability to efficiently vehicle any permutation from its input ports to its output ports. Efficiency requirement means scalability within optimal tradeoffs between performance (bandwidth, latency) and hardware complexity (area cost).

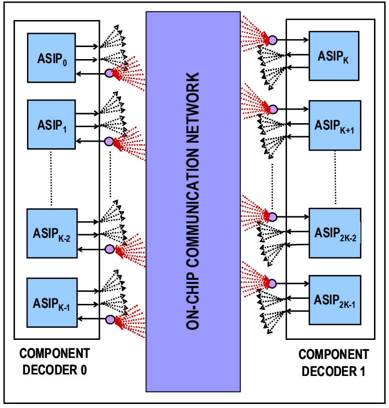

Figure 1. Extrinsic information exchanges in BCJR-SISO decoder level parallelism

Another point which should be noted is that each processor can output two extrinsic information data (Figure 1) at each emission if the processor implements the butterfly decoding scheme [9]. However, as dual-port memories are area expensive (almost twice the size of single-port ones), single-port input local memories are considered, thus only one extrinsic information data can be received by each processor at a time (Figure 1).

3. Butterfly Network

The Butterfly network [10] is a multistage interconnection network with 2-input 2-output routers and unidirectional links. The advantages of this topology are: first, the logarithmic diameter of the network ($\log_2 N$ with N the number of network input ports) which gives a number of routers equal to $\frac{N}{2}\log_2 N$; then the recursive structure of the network (a network of diameter D is obtained with two networks of diameter D-1) which enables huge scalability; and finally a very simple routing that uses directly the bits of the destination address for the selection of the output port at each stage of the network.

However the Butterfly network lacks path diversity since it presents a unique path between each source and each destination. This increases the risk of conflict inside routers and implies the usage of queues to bufferize packets.

Figure 2 represents a Butterfly network that connects 8 ASIPs each producing 2 packets for 8 memories attached to destination ASIPs. Network Interfaces (NI) are used between network and processor/memories.

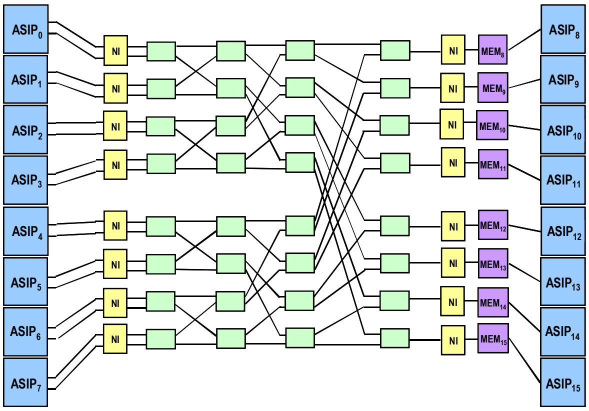

Figure 2. Extrinsic information transmission network based on the Butterfly topology

3.1. Routing Technique

The Destination-Tag routing is used. This deterministic routing uses a digit of the destination address in the header of the packets to select the output port at each router along the path from the source to the destination.

3.2. Packet Format

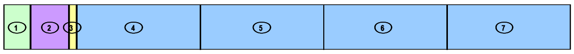

Figure 3.Packet Format

Packets (figure 3) are composed of a header and a payload. The header includes the previously defined routing information ① and the destination memory address ②. The first field has a width of D bits, D being the diameter of the network and the second field a width of $\left\lfloor \log_2 \frac{N}{P} \right\rfloor + 1$ bits, where N is the length of the frame to be decoded and P the number of processors in an interleaving domain. The payload gathers a one-bit flag ③ indicating whether intra symbol permutation, as defined in the DVB-RCS standard [11], is applied to this packet. The fields ④ to ⑦, each one of 16 bits, represent the extrinsic information corresponding to the reception of each possible symbol.

3.3. Routers

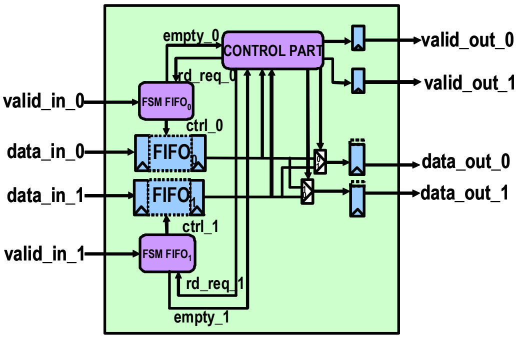

Figure 4.Butterfly Router Architecture

Routers (figure 4) propagate packets from the source to the destination memory, thanks to routing information. Indeed, the most significant bit of routing information in the header determines the output of the current router and is then discarded in output registers.

Router architecture is divided into two parts: a datapath with the FIFO queues, the multiplexers and the registers; and a control part for the management of the FIFO FSMs, the multiplexer selection signals and the generation of the validity output signals.

FIFO queues are mandatory to preserve packets when conflicts occur on the router output. In this version of the network, the selected depth of the FIFOs is equal to 2^i, where i is the stage number of the Router (i is in the range of 0 to D-1, with D the network diameter). Data are written into the FIFOs on the rising edge of the clock and into the output registers on the falling edge. This is the same for the Network Interfaces.

3.4. Last Stage Routers

As the number of packets to be transmitted is twice the number of output ports, the topology requires special routers for the last stage (figure 5) of the network, enabling the incoming flow of packets to be multiplexed. Thus, last stage routers have only one output register instead of two and do not perform routing.

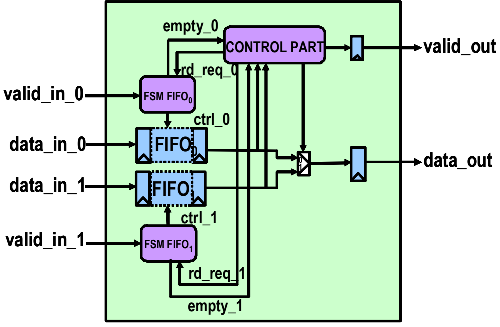

Figure 5.Butterfly Last Stage Router Architecture

3.5. Transmission Network Interfaces

The Transmission Network Interface (figure 6) transforms the header of the packets coming from the ASIPs into the routing information and the write address of destination memory. This transformation, performed by the conversion table, is exactly the interleaving (or deinterleaving).

Each Interface is composed of this conversion table, two registers for the input data and two registers for input validity signals. At the output of the Interface, two registers recopy the input validity signals and two data registers receive the packets formed by the concatenation of the payload of initial packets with information resulting from the conversion table.

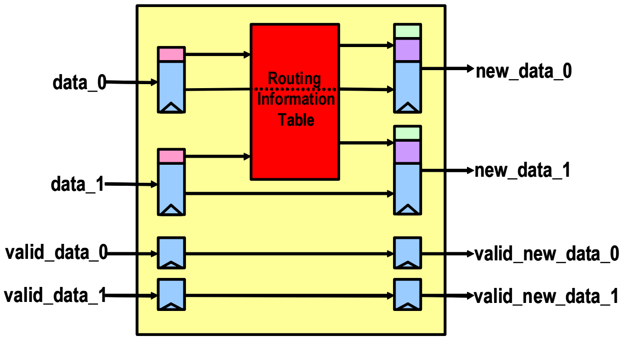

Figure 6.Butterfly TX Network Interface

3.6. Reception Network Interfaces

At the level of the network outputs, the interfaces (figure 7) are rather simple because they only have one registered input and one single output feeding the single-port destination memory. Moreover, two multiplexers are mandatory to perform the intra symbol permutation.

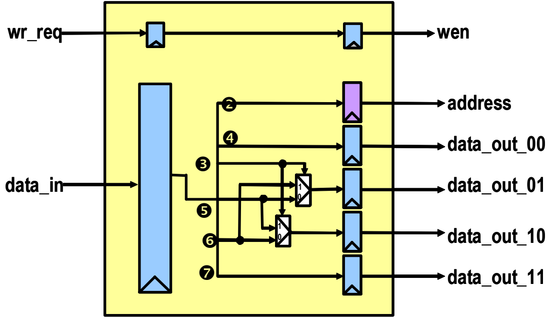

Figure 7.Butterfly RX Network Interface

3.7. Conflict management with priorities

The Symbol Reliability Difference (SRD) criterion, proposed in [12], estimates each symbol's contribution to the convergence of the turbo decoding process.

As proposed in [12], unnecessary extrinsic information exchange according to an appropriate threshold can be inhibited to reduce the traffic by 20% with negligible error rate degradation. However, according to simulations with a Butterfly network, this traffic reduction has no influence on tailoring of FIFO depth and consequently on global area.

Based on the SRD criterion, packets can be affected by a priority depending of their contribution to the convergence of the turbo decoding process. Thus in case of conflicts, a Butterfly router takes care of packet contribution rather than performing a Round-Robin arbitration.

In addition to packet contribution estimation, the turbo decoding process can predict for a packet the maximal propagation time required to cross the network according to interleaving laws and decoding scheme. Consequently a propagation priority can be affected to each packet.

The number of bits for contribution and propagation priorities is scaled as desired in application. Our example uses 2 bits for contribution priority and 2 bits for propagation priority. So a new priority field of 4 bits is inserted in the header to help arbitration of conflicting packets. Propagation priority is set on the Most Significant Bits (MSB) of this field to ensure urgent packets are delivered before important packets.

4. Benes 2N-N Network

Since we wish to comply with any type of interleaving for a turbo-decoding application, we need an interconnection network which supports all the possible permutations of its inputs with its outputs. Moreover, this network must offer path diversity in order to reduce the conflicts between packets as much as possible.

The Benes network [13] is one of the already existing networks which has these characteristics. Built from two Butterflies put back-to-back, its diameter is almost the double of that of Butterfly: $2\log_2 N-1$. In addition, the latency is constant for all the couples (source, destination) and it corresponds to the network diameter. However, this network avoids the conflicts if and only if all the paths have a different destination. But this is not the case for the turbo-decoding application because interleaving (respectively deinterleaving) ends in potentials conflicts.

The suggested solution is to choose the packets to be transmitted so that for each cycle, none is intended for the same network output port.

On the basis of this constraint, the Benes topology was modified and gave the Benes 2N-N network, illustrated in figure 8.

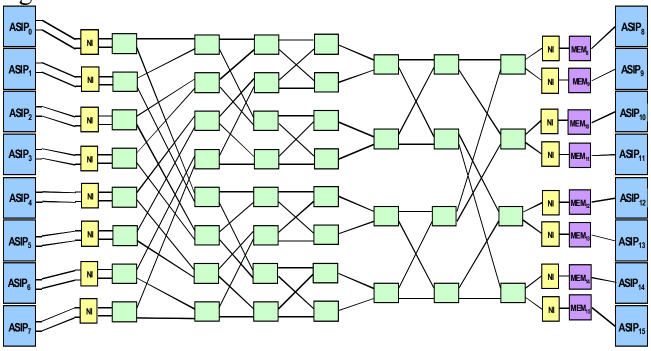

Figure 8. Extrinsic information transmission network based on the Benes 2N-N topology

4.1. Routing Technique

The routing in the Benes 2N-N network differs from that of the Butterfly in the fact that a preprocessing must be done to compute the routing information. The main idea is: no conflict will occur if we apply a preprocessed routing rule. This rule conditions the choice of the output of the Router to be borrowed for the packets which are intended for adjacent network output ports (S/S+1 where S is even). If one packet wishes to reach the output port S and another packet the port S+1, these packets must each take complementary router outputs (0 for the first/1 for the second or vice versa). Moreover, each of the first $\frac{D}{2}$ network stages is divided into 2^i (i being the stage number and in the range of 0 to $\frac{D}{2}-1$) subsets of Routers, to which we apply this rule with S' the new value of the network output port to be reached. S' is defined as such: $S'=\frac{S}{2^i}$.

If the network has a D diameter, the routing information will be on D-1 bits (and not D because of the Central Router). The Benes 2N-N topology being based on traditional Benes which is the concatenation of inverse Butterfly and Butterfly networks, there is no need to determine the whole of D-1 bits and only the first $\frac{D}{2}$ bits are enough because the remaining $\left(\frac{D}{2}-1\right)$ bits coincide with the address of the destination memory.

Table 1 illustrates an example of the routing information computation for a Benes 16-8 network.

Table 1 Example of routing information computation for a Benes 16-8 network

INPUT	Init Dst
0	0
1	5
2	-1/0
3	7
4	2
5	-1/7
6	-1/5
7	6
8	1
9	-1/2
10	3
11	-1/0
12	-1/3
13	4
14	-1/7
15	-1/4

Z	O
0/0	1/5
2/-1	3/7
4/2	5/-1
7/6	6/-1
9/-1	8/1
11/-1	10/3
13/4	12/-1
14/-1	15/-1

D = Init Dst

Z	O
0/0	2/-1
7/3	4/1
9/-1	11/-1
14/-1	13/2

Z	O
1/2	3/3
5/-1	6/-1
8/0	10/1
12/-1	15/-1

D = Init Dst/2

Z	O
0/0	7/1
9/-1	14/-1

Z	O
2/-1	4/0
13/1	11/0

Z	O
1/1	5/-1
12/-1	8/0

Z	O
3/1	6/-1
15/-1	10/0

D = Init Dst/4

INPUT	Routing Information
0	000000
1	100101
2	010000
3	110111
4	011010
5	101111
6	111101
7	001110
8	101001
9	000010
10	111011
11	011000
12	100011
13	010100
14	001111
15	110100

To avoid conflict, we chose to schedule all the packets by allocating to each one a certain time slot. In this way, at each cycle, none of the packets will have the same destination. This sequencing, established for all the packets at the input of the network, must be statically precomputed and the resulting slots stored in the conversion table of the Transmission Network Interface.

Table 2 shows an example of the slot assignments for the packets at the input of a Benes 16-8 network. As we can see, memory 0 is addressed at the same time by the packets of inputs 1, 4, 11, 13 and 15 to which the slots 2, 3, 1, 4 and 0 are allocated.

Table 2. Example of slot allocation for a Benes 16-8 network

Input	Dest	SLOT 0	SLOT 1	SLOT 2	SLOT 3	SLOT 4
0	7	X	X	7	X	X
1	0	X	X	0	X	X
2	6	6	X	X	X	X
3	1	X	1	X	X	X
4	0	X	X	X	0	X
5	5	5	X	X	X	X
6	3	3	X	X	X	X
7	2	X	2	X	X	X
8	7	7	X	X	X	X
9	1	1	X	X	X	X
10	4	4	X	X	X	X
11	0	X	0	X	X	X
12	7	X	7	X	X	X
13	0	X	X	X	X	0
14	2	2	X	X	X	X
15	0	0	X	X	X	X

4.2. Packet Format

The packets for the Benes 2N-N network have the same format as that of the Butterfly network (section 3.2), but different routing information field (previous section).

4.3. Routers

In the Benes 2N-N routers (Figure 9), the FIFOs are replaced by simple registers because no more conflicts have to be managed. The width of the output registers is equal to that of the input registers minus 1 bit. The synchronization of the writes is similar to those of the Butterfly network.

4.4. Central Routers

In order to multiplex the outgoing packets, a second type of Routers, Central Routers (figure 10), has been designed. They have only one output port and all the data registers have identical size.

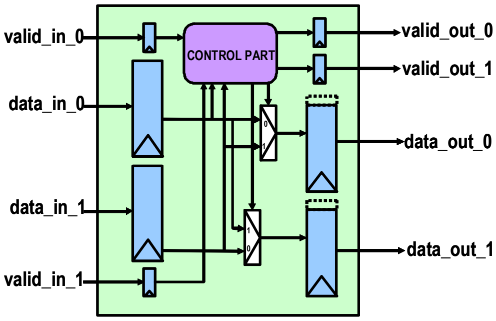

Figure 9. Benes 2N-N Router Architecture

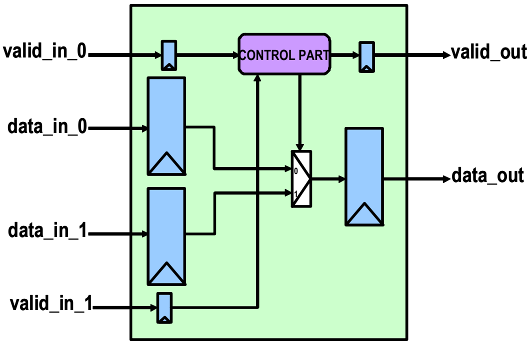

Figure 10. Benes 2N-N Central Router Architecture

4.5. Network Interfaces

Based on those of the Butterfly network, the Transmission Network Interfaces (figure 11) have two down counters which are used to delay packet transmission according to the associated time slot. This latter is stored in the routing table and is taken as the initialization value of the down counter. Benes and Butterfly Reception Network Interfaces are identical.

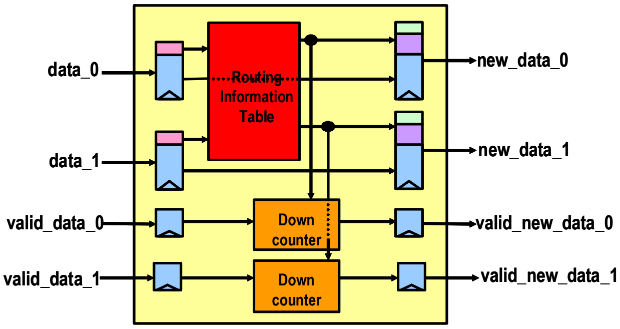

Figure 11. Benes 2N-N TX Network Interface

In the following table, we have summarized the differences between the two proposed network architectures in terms of stage numbers, router numbers, routing algorithm, bufferization and conflicting packets management.

Table 3. Butterfly and Benes 2N-N networks comparison

	Butterfly	Benes 2N-N
# of stages	$\log_2 N$	$2\log_2 N-1$
# of Routers	$\frac{N}{2}\log_2 N$	$\frac{N}{2}\log_2 N+\frac{N}{4}(\log_2 N-1)$
Routing Algorithm	Destination Tag	Preprocessed routing calculation method
Bufferization	Input FIFOs	None
Conflicting packet managment	Bufferization	Packets scheduling (Time slots)

5. Synthesis and Results Analysis

In order to evaluate the efficiency of the proposed architectures, a synthesizable RTL VHDL description of the two networks (Routers and Network Interfaces) was developed for a DVB-RCS turbo-decoder with 8 processors per interleaving domain, a block length of 106 bytes, and 8 ROMs with 53 words of 9 bits depth for Butterfly and 15 bits for Benes 2N-N.

To estimate the area consumption including Network Interfaces and the aggregate bandwidth of the two networks, synthesis was carried out with ST 0.18 µm technology under worst-case conditions. Table 4 shows the results obtained using the 0.18 µm technology for Butterfly, Butterfly with priorities management, and Benes 2N-N networks. Area complexity is compared regarding Transmission Network Interfaces, Reception Network Interfaces, and Routers. Number of routers in the considered architecture is 32 for Butterfly topology and 44 for Benes 2N-N topology. The priority packet field is implemented on 4 bit width for the second Butterfly network.

Table 4. ASIC synthesis results of the Butterfly and BeneS networks on 0.18 µm technology

	Butterfly	Butterfly with priority	Benes 2N-N
Frequency	302 *MHz*	250 *MHz*	416 *MHz*
Area of one TX NI	0.0250 *mm²*	0.0262 *mm²*	0.0376 *mm²*
Area of one RX NI	0.0104 *mm²*	0.0104 *mm²*	0.0104 *mm²*
Area of one RT	0.0670 *mm²*	0.0690 *mm²*	0.0219 *mm²*
Total Area	2.4170 *mm²*	2.5010 *mm²*	1.3460 *mm²*
Payload Bandwidth	154 *Gb/s*	128 *Gb/s*	213 *Gb/s*
Latency Min/Max	16/63 *ns*	20/76 *ns*	19/53 *ns*

The Benes 2N-N network enables a frequency gain of 37.7% for Butterfly without management of priorities and of 66.4% for Butterfly with management of priorities. Furthermore the consumed surface is reduced from approximately 44.3% and 46.2%.

Moreover, taking priorities into account within the Butterfly network leads to a frequency reduction of 17.2% and an area increase of 3.47%.

These results are compared in table 5 with other 16 Extrinsic Information producer topologies: Chordal RIBB [5], Random Network with Bubble Flow Control [6] and (4,4)-2D Mesh [14].

Table 5. ASIC synthesis results of the Butterfly, BeneS, Chordal RIBB, Random Network and 2D Mesh on 0.18 µm technology for a number of16 Extrinsic Information producers

	Frequency (MHz)	*Area (mm²)*	*Aggregate Bw (Gb/s)*
Benes 2N-N	416	1.34	213
Butterfly	302	2.5	154
Chordal RIBB	200	4.19	64
Random Network Bubble Flow Control	170	1.35	54.4
(4,4)-2D Mesh (IQ)	200	1.2	64

The frequency obtained with our networks is 1.5 to 2 times higher than the one of the other networks. Furthermore the Benes 2N-N topology implies small area as routers in this network do not integrate buffering resources. Although Butterfly topology implies the use of FIFOs, the proposed Butterfly network area remains 1.6 times smaller than the Chordal RIBB and approximately twice as large as the Random or the Mesh ones.

The obtained values of aggregate bandwidth are due to the support of double-binary codes. Consequently the number of exchanged extrinsic information is 4 instead of one and the payload of a packet for these networks is 4 times larger than the payload of a packet for UMTS standard.

As simulations and synthesis were done for a given number of the input ports of the Benes and Butterfly networks (N = 16), table 6 proposes for an arbitrary number N an complexity estimation of the two proposed networks in terms of total size, aggregate bandwidth and latency.

Table 6. Butterfly and Benes complexity estimations in terms of Total Area, Payload Bandwidth and Latency

	Butterfly	Benes 2N-N
Frequency @0.18µm	302 *MHz*	416 *MHz*
# of RTs	$O(N\log_2 N)$	$O(N\log_2 N)$
Area of one RT	$O(\frac{N}{\log_2 N})$	$O(1)$
Total Area	$O(N^2)$	$O(N\log_2 N)$
Aggregate Bandwidth (Gb/s)	$64\text{x}Freq_{Butterfly}\text{x}\frac{N}{2}$	$64\text{x}Freq_{Benes}\text{x}\frac{N}{2}$
Latency Min/ Max	$\log_2 N+1$ / $N-1+\log_2 N$ cy.	$2\log_2 N$ / $N-2+2\log_2 N$ *cy.*

First of all, for the Benes network and for the Butterfly network the total number of routers varies in $N\log_2 N$. But the size of the Butterfly routers depends on N (due to buffering increase), whereas it is almost constant for Benes. Consequently, the total size of a Benes network will evolve in $N\log_2 N$ and in N^2 for the Butterfly network.

The clock frequency of the networks is almost constant for a given technology and does not depend on the number of input ports. Lastly, there is a disparity in latency for the two networks. For Benes, the packets cross the network in at least $2\log_2 N$ cycles and at most $N-2 + 2\log_2 N$ cycles (waiting in network interface). Butterfly latency varies between a minimum of $\log_2 N + 1$ cycles and a maximum of $N-1+\log_2 N$ cycles.

Using the network supporting priorities enables to guarantee that the most useful packets for turbo decoding process convergence arrive first and thus have latency close to $\log_2 N$. On the contrary, low priority packets are delayed in FIFOs and can approach, in worst conditions, a linear latency.

As point of comparison, the Mesh, Chordal ring and Random network have exactly N routers. The router area is constant for Mesh, but not for Chordal Ring and Random networks whose buffer size increases with the increasing number of inputs/outputs of the router.

6. Conclusion

In this paper, we proposed two multistage interconnection networks for parallel turbo decoding based on Butterfly and Benes topologies. The main feature of the presented on-chip networks is their scalability which allows seamless tradeoffs between hardware complexity and aggregate bandwidth. For both networks, detailed hardware architecture of the network interfaces, routers, and topologies, together with packet format and routing technique are proposed. The obtained results demonstrate a scalable aggregate bandwidth of 32xFxN Gbps for both networks, where N is the number of network input ports and F is the clock frequency. The clock frequency of the networks is almost constant for a given technology and does not depend on the number of input ports. Results analysis shows that the Benes 2N-N solution achieves better performance in terms of occupied area and maximum frequency, however it requires a pre-processing step to compute routing tables for each different turbo code interleaver rule. However, the capability for both networks to achieve efficiently any permutation between input and output ports enables their use for all turbo code standards and constitutes a promising feature for its reuse for any similar interleaved/deinterleaved iterative communication profile.

References

[1] C. Berrou, A. Glavieux, P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes", in Proc. 1993 International Conference on Communications (ICC'93), Geneva, Switzerland, 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, Mar. 1999.

[3] D. Gnaëdig, E. Boutillon, M. Jezequel, V. Gaudet, G. Gulak, "On Multiple Slice Turbo Code", 3nd International Symposium on Turbo Codes and Related Topics, Brest, France, pp. 343-346, Sept. 2003.

[4] A. La Rosa, C. Passerone, F. Gregoretti, and L. Lavagno, "Implementation of a UMTS turbo-decoder on a dynamically reconfigurable platform", Proceedings of DATE 2004, Paris.

[5] M. J. Thul, F. Gilbert, N. When, "Concurrent Interleaving Architectures for High-Throughput Channel Decoding", in Proc. 2003 Conference on Acoustics, Speech, and Signal Processing (ICASSP'03), pages 613-616, Hong Kong, P.R. China, Apr. 2003

[6] M. J. Thul, "Parallel interleaving Architectures for High Throughput Turbo-decoders", Ph.D. Thesis, Electronic System Design Research Group, University of Kaiserslautern, Germany, 2005

[7] O. Muller, A. Baghdadi, M. Jézéquel, "Exploring Parallel Processing Levels for Convolutional Turbo Decoding", IEEE International Conference on Information & Communication Technologies: from Theory to Applications (ICTTA'06), April 2006.

[8] L. Bahl, J. Cocke, F. Jelinek, J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate", IEEE Trans. Inf. Theory, vol. IT-20, pages. 284-287, Mar. 1974.

[9] O. Muller, A. Baghdadi, 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, pages 1330-1335, Munich, Germany, Mar. 2006.

[10] J. M. Cooley, J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series", Math. Comp., pages 297–301, 1965.

[11] C. Douillard, M. Jézéquel, C. Berrou, N. Brengarth, J. Tousch, N. Pham, "The Turbo Code Standard for DVB-RCS", in Proc. Of 2nd International Symposium on Turbo Codes & Related Topics, Brest, France, pages 535-538, Sep. 2000.

[12] O. Muller, A. Baghdadi, M. Jézéquel, "Bandwidth reduction of extrinsic information exchange in turbo decoding", IEE Electronics Letters, pages 1104-1106, Sept. 2006.

[13] V. E. Benes, "Mathematical Theory of Connecting Network and Telephone Trafic", New York, NY: Academic, 1965.

[14] C. Neeb, M.J. Thul, N. When, "Network-on-Chip-Centric Approach to Interleaving in High Throughput Channel Decoders", 2005 IEEE International Symposium on Circuits and Systems (ISCAS), pages 1766-1769, Kobe, Japan, May 2005