Hindawi Publishing Corporation
VLSI Design
Volume 2012, Article ID 730835, 16 pages
doi:10.1155/2012/730835

Review Article

Flexible LDPC Decoder Architectures

Muhammad Awais and Carlo Condo

Department of Electronics and Telecommunications, Politecnico di Torino, 10129 Torino, Italy

Correspondence should be addressed to Carlo Condo, carlo.condo@polito.it

Received 4 November 2011; Revised 14 February 2012; Accepted 22 February 2012

Academic Editor: Amer Baghdadi

Flexible channel decoding is getting significance with the increase in number of wireless standards and modes within a standard. A flexible channel decoder is a solution providing interstandard and intrastandard support without change in hardware. However, the design of efficient implementation of flexible low-density parity-check (LDPC) code decoders satisfying area, speed, and power constraints is a challenging task and still requires considerable research effort. This paper provides an overview of state-of-the-art in the design of flexible LDPC decoders. The published solutions are evaluated at two levels of architectural design: the processing element (PE) and the interconnection structure. A qualitative and quantitative analysis of different design choices is carried out, and comparison is provided in terms of achieved flexibility, throughput, decoding efficiency, and area (power) consumption.

1. Introduction

With the word flexibility regarding channel decoding, we mean the ability of a decoder to support different types of codes, enabling its usage in a wide variety of situations. Much research has been done in this sense after the great increase in number of standards, standard complexity, and code variety witnessed during the last years. Next-generation wireless standards such as DVB-S2 [1], IEEE 802.11n (WiFi) [2], IEEE 802.3an (10GBASE-T) [3], and IEEE 802.16e (WiMAX) [4] feature multiple codes (LDPC, Turbo), where each code comes with various code lengths and rates. The necessity for flexible channel decoder intellectual properties (IPs) is evident and challenging due to the often unforgiving throughput requirements and narrow constraints of decoder latency, power, and area.

This work gives an overview of the most remarkable techniques in context of flexible channel decoding. We will discuss design and implementation of two major functional blocks of flexible decoders: processing element (PE) and interconnection structure. Various design choices are analyzed in terms of achieved flexibility, performance, design novelty, and area (power) consumption.

The paper is organized as follows. Section 2 provides a brief introduction to LDPC codes and decoding. Section 3 gives an overview of flexible LDPC decoders classifying them on the basis of some important attributes for example, parallelism, implementation platforms, and decoding schedules. Sections 4 and 5 are dedicated to PE and interconnection structure, respectively, where we depict various design methodologies and analyze some state of the art flexible LDPC decoders. Finally, Section 6 draws the conclusions.

2. LDPC Decoding

2.1. Introduction. LDPC codes [5] are a special class of linear block codes. A binary LDPC code is represented by a sparse parity check matrix **H** with dimensions $M \times N$ such that each element h_{mn} is either 0 or 1. N is the length of the codeword, and M is the number of parity bits. Each matrix row $\mathbf{H}_{(i,(1\leq j\leq N))}$ introduces one parity check constraint on the input data vector $x = \{x_1, x_2, \ldots, x_N\}$:

$$\mathbf{H_i} \cdot \mathbf{x}^{\mathbf{T}} = \mathbf{0} \bmod \mathbf{2}. \qquad (1)$$

The complete **H** matrix can best be described by a Tanner graph [6], a graphical representation of associations between code bits and parity checks. Each row of **H** corresponds to a check node (CN), while each column corresponds to a variable node (VN) in the graph. An edge e_{ji} on the Tanner Graph connects a VN$_j$ with CN$_i$ only if the corresponding element h_{ij} is a1 in **H**. If the number of edges entering in a node is constant for all nodes of the graph, the LDPC code is called regular, being otherwise irregular in case of variable node degree. Irregular LDPC codes yield better decoding performance compared to regular ones.

Next-generation wireless communication standards adopt structured LDPC codes, which hold good interconnection, memory and scalability properties at the decoder implementation level. In these codes, the parity check matrix $\mathbf{H}$ is associated to a $\mathbf{H}_{\text{BASE}}$ matrix, as defined in [7]:

$$\mathbf{H}_{\text{BASE}} = \begin{bmatrix} \Pi_{0,0} & \Pi_{0,1} & \dots & \Pi_{0,N_b} \\ \Pi_{1,0} & \Pi_{1,1} & \dots & \Pi_{1,N_b} \\ \vdots & \vdots & \ddots & \vdots \\ \Pi_{M_b,0} & \Pi_{M_b,1} & \dots & \Pi_{M_b,N_b} \end{bmatrix}. \quad (2)$$

$\mathbf{H}_{\text{BASE}}$ has M_b block rows and N_b block columns; it is expanded, in order to generate the $\mathbf{H}$ matrix, by replacing each of its entries $\Pi_{i,j}$ with a $z \times z$ permutation matrix, where z is the expansion factor. The permutation matrix can be formed by a series of right shifts of the $z \times z$ identity matrix according to a determined shifting factor, equal to the value $\Pi_{i,j}$. The same base matrix is used as a platform for all the different code lengths related to a selected code rate: implementation of a full-mode decoder is thus a challenging task, due to huge variations in code parameters. For example, current IEEE 802.16e WiMAX standard features four code rates, that is, 1/2, 2/3, 3/4, and 5/6 with $\mathbf{H}_{\text{BASE}}$ matrices of size 12×24, 8×24, 6×24, and 4×24, respectively. Each code rate comes with 19 different codeword sizes ranging from 576 bits ($z = 24$) to 2304 bits ($z = 96$), with granularity of 96 bits ($\Delta z = 4$).

Algorithm 1 (The Standard TPMP Algorithm).

(1) *Initialization: For $j \in \{1, \dots, N\}$*

$$\alpha^0_{i,j} = \ln \frac{P\left(VN_j = 0 \mid y_j\right)}{P\left(VN_j = 1 \mid y_j\right)} = \frac{2y_j}{\sigma^2}. \quad (3)$$

(2) *CN Update Rule: $\forall CN_i$, $i \in \{1, \dots, M\}$ do*

$$\beta^n_{i,j} = \operatorname{sgn} \beta^n_{i,j} \cdot \left|\beta^n_{i,j}\right|, \quad (4)$$

$$\operatorname{sgn} \beta^n_{i,j} = \prod_{j' \in \mathcal{N}(i) \setminus j} \operatorname{sgn}\left(\alpha^{(n-1)}_{ij'}\right),$$
$$\left|\beta^n_{i,j}\right| = \underset{j' \neq j}{\otimes} \left(\alpha^{n-1}_{i,j'}\right). \quad (5)$$

(3) *VN Update Rule: $\forall VN_j$, $j \in \{1, \dots, N\}$ do*

$$\alpha^n_{i,j} = \alpha^0_{i,j} + \sum_{i' \in \mathcal{M}(j) \setminus i} \beta^n_{i',j}. \quad (6)$$

(4) *Decoding: For each bit, compute it's a posteriori LLR*

$$\alpha^n_j = \alpha^0_{i,j} + \sum_{i' \in \mathcal{M} \setminus (j)} \beta^n_{i',j}. \quad (7)$$

Estimated codeword is $\widehat{C} = (\widehat{c}_1, \widehat{c}_2, \dots, \widehat{c}_N)$, where element $\widehat{c}_j$ is calculated as

$$\widehat{c}_j = \begin{cases} 0 & \text{if } \alpha^n_j > 0 \\ 1 & \text{else.} \end{cases} \quad (8)$$

If $\mathbf{H}(\widehat{C})^T = 0$, then stop, with correct codeword $\widehat{C}$.

2.2. LDPC Decoding Algorithms. The nature of LDPC decoding algorithms is mainly iterative. Most of these algorithms are derived from the well-known belief propagation (BP) algorithm [5]. The aim of BP algorithm is to compute the a posteriori probability (APP) that a given bit in the transmitted codeword $c = [c_0, c_1, \dots, c_{N-1}]$ equals 1, given the received word $y = [y_0, y_1, \dots, y_{N-1}]$. For binary phase shift keying (BPSK) modulation over an additive white Gaussian noise (AWGN) channel with mean 1 and variance σ^2, the reliability messages represented as logarithmic likelihood ratio (LLR) are computed in two steps: (1) check node update and (2) variable node update. This is also referred to as two-phase message passing (TPMP). For nth iteration, let $\alpha^n_{i,j}$ represent the message sent from variable node VN_j to check node CN_i, $\beta^n_{i,j}$ represent the message sent from CN_i to VN_j, $\mathcal{M}(j) = \{i : \mathbf{H}_{ij} = 1\}$ is the set of parity checks in which VN_j participates, $\mathcal{N}(i) = \{j : \mathbf{H}_{ij} = 1\}$ the set of variable nodes that participate in parity check i, $\mathcal{M}(j) \setminus i$ the set $\mathcal{M}(j)$ with check CN_i excluded, and $\mathcal{N}(i) \setminus j$ the set $\mathcal{N}(i)$ with VN_j excluded. The standard TPMP algorithm is described in Algorithm 1.

As given in (4), the CN update consists of sign update and magnitude update, where the latter depends on the type of decoding algorithm, of which several are commonly used (Table 1). The sum product (SP) algorithm [8] gives near-optimal results; however, the implementation of the transcendental function $\Phi(x)$ requires dedicated LUTs, leading to significant hardware complexity [9]. Min-Sum (MS) algorithm [10] is a simple approximation of the SP: its easy implementation suffers an 0.2 dB performance loss compared to SP decoding [11]. Normalized Min-Sum (NMS) algorithm [12] gives better performance than MS by multiplying the MS check node update by a positive constant λ_k, smaller than 1. Offset Min-Sum (OMS) is another improvement of standard MS algorithm which reduces the reliability values β^n_{ij} by a positive value β: for a quantitative performance comparison for different CN updates, refer to [13, 14].

2.3. Layered Decoding of LDPC Codes. Modifying the VN update rule (6):

$$\alpha^n_{i,j} = \alpha^n_j - \beta^n_{i,j}, \quad (9)$$

we can merge the CN and VN update rules into a single operation, where the CN messages $\beta^n_{i,j}$ are computed from $\alpha^{(n-1)}_j$ and $\beta^{(n-1)}_{i,j}$. This technique is called layered decoding [15]. Layered decoding considers the $\mathbf{H}$ matrix as a concatenation of l layers (block rows) or constituent subcodes, that is, $\mathbf{H}^{\mathbf{T}} = [\mathbf{H}^{\mathbf{T}}_{\mathbf{1}} \mathbf{H}^{\mathbf{T}}_{\mathbf{2}}, \dots, \mathbf{H}^{\mathbf{T}}_{\mathbf{l}}]$, where the column weight

Table 1: Check node update for LDPC decoding algorithms.

Algorithm	Formulation: $\underset{j' \neq j}{\otimes} \left(\alpha_{i,j'}^{(n-1)}\right)$
SP	$\Phi(\sum_{j' \in \mathcal{N}(i) \setminus j} \Phi(\lvert\alpha_{ij'}^{(n-1)}\rvert)$
	$\Phi(x) = -\log(\tanh(x/2))$
MS	$\min_{j' \in \mathcal{N}(i) \setminus j}\{\lvert\alpha_{i,j'}^{(n-1)}\rvert\}$
OMS	$\max\{\min_{j' \in \mathcal{N}(i) \setminus j}\{\lvert\alpha_{i,j'}^{(n-1)}\rvert\} - \beta, 0\}$
	$\beta \geq 0$
NMS	$\lambda \cdot \min_{j' \in \mathcal{N}(i) \setminus j}\{\lvert\alpha_{i,j'}^{(n-1)}\rvert\}$
	$\lambda < 1$

of each layer is at most 1. In this way, a decoding iteration is divided into l subiterations. Formally, the algorithm for layered decoding Min-Sum is described in Algorithm 2.

After CN update is finished for one block row, the results are immediately used to update the VNs, whose results are then used to update the next layer of check nodes. Therefore, an updated information is available to CNs at each subiteration. Based on the same concept, the authors in [7] introduced the concept of turbo decoding message passing (TDMP) [16] using the BCJR algorithm [17] for their architecture-aware LDPC (AA-LDPC) codes. TDMP results in about 50% decrease in number of iterations to meet a certain BER, which is equivalent to $a \times 2$ increase in throughput and significant memory savings as compared to the standard TPMP schedule. Similar to the TDMP schedule is the vertical shuffle scheduling (VSS) [18]: while TDMP relies on horizontal divisions of the parity check matrix, VSS divides the horizontal layers into subblocks. It is a particularly efficient technique with quasicyclic LDPC codes [19], where each subblock is identified by a parity check submatrix.

Algorithm 2 (The Layered Decoding Min-Sum).

(1) *Initialization:* $\forall CN_i,\ i \in \{1, \ldots, M\}$ *do* $\beta_{i,j}^0 = 0$.

(2) *CN Update Rule:* $\forall CN_i,\ i \in \{1, \ldots, M\}$ *do*

$$\alpha_j^n = \alpha_{i,j}^0, \tag{10}$$

$$\begin{aligned}\beta_{i,j}^n &= \prod_{j' \in \mathcal{N}(i) \setminus j} \operatorname{sgn}\left\{\alpha_j^{(n-1)} - \beta_{i'j}^{(n-1)}\right\} \\ &\quad \times \min_{j' \in \mathcal{N}(i) \setminus j} \left|\alpha_j^{(n-1)} - \beta_{i,j'}^{(n-1)}\right|,\end{aligned} \tag{11}$$

$$\alpha_j^n = \alpha_j^n + \beta_{i,j}^n. \tag{12}$$

Estimated codeword is $\widehat{C} = (\widehat{c}_1, \widehat{c}_2, \ldots, \widehat{c}_N)$, where element $\widehat{c}_j$ is calculated as

$$\widehat{c}_j = \begin{cases} 0 & \text{if } \alpha_j^n > 0 \\ 1 & \text{else.} \end{cases} \tag{13}$$

If $\mathbf{H}(\widehat{C})^T = 0$, then stop, with correct codeword $\widehat{C}$.

3. Flexible Decoders

3.1. Parallelism. The standard TPMP algorithm described in the previous section exploits the bipartite nature of the Tanner Graph: since no direct connection is present between nodes of the same kind, all CN (or VN) operations are independent from each other and can be performed in parallel. Thus, a first broad classification of LDPC decoders can be done in terms of the degree of parallelism. The hardware implementation of LDPC decoders can be serial, partially parallel, and fully parallel.

Serial LDPC decoder implementation is the simplest in terms of area and routing. It consists of a single check node, a single variable node, and a memory. The variable nodes are updated one at a time, and then check nodes are updated in serial manner. Maximum flexibility could be achieved by uploading new check matrices in memory. However, each edge of the graph must be handled separately: as a result, throughput is usually very low, insufficient for most of standard applications.

A fully parallel architecture is the direct mapping of Tanner graph to hardware. All node operations (CNs and VNs) are directly realized in hardware PEs and connected through dedicated links. This results in huge connection complexity that in extreme cases dominates the total decoder area and results in severe layout congestion: maximum throughput can be, however, theoretically reached. In [20], a 1024-bit, fully parallel decoder is presented, achieving 1 Gbps throughput with logic density of only 50% to accommodate the complexity of interconnection: it comprises of 9750 wires with 3-bit quantization. None of the parallel implementations in [20–22] grant multimode flexibility due to wired connections. In addition, almost all existing fully parallel LDPC decoders are built on custom silicon, which precludes any prospect of reprogramming. An alternative approach is the partially parallel architecture which divides the node operations of Tanner graph over P PEs, with $P < (N + M)$. This means that each PE will perform the computation associated to multiple nodes, necessitating memories to store intermediate messages between tasks. Time sharing of PEs greatly reduces the area and routing overhead. Partially parallel architectures are studied extensively and provide a good trade off in throughput, complexity, and flexibility, with some solutions obtaining throughputs up to 1 Gbps.

3.2. Implementation Platforms. Hardware implementation of LDPC decoders is mainly dictated by the nature of application. LDPC codes have been adopted by a number of communication (wireless, wired, and broadcast) standards and storage applications: a few of them are briefly summarized in Table 2.

In wireless communication domain, LDPC codes are adopted in IEEE 802.16e WiMAX which is a wireless metropolitan area network (WMAN) standard and IEEE 802.11n WiFi which is a wireless local area network (WLAN) standard. Both standards have adopted LDPC codes as an optional channel coding scheme with various code lengths and code rates. LDPC codes are also used in digital video broadcast via satellite (DVB-S2) standard which requires

Table 2: LDPC codes applications.

Application	Standard	Code length	Code rates	Throughput
WMAN	IEEE 802.16e	576–2304	1/2–5/6	70 Mb/s
WLAN	IEEE 802.11n	648–1944	1/2–5/6	450 Mb/s
Broadcast	DVB-S2	6400,64800	1/4–9/10	90 Mb/s
Wired	10Gbase-T	2048	Arbitrary	6.4 Gbps

very large code lengths of 64800 bits and 16200 bits with 11 different codes rates, and a 90 Mb/s decoding throughput. In wireline communication domain, LDPC codes are adopted in 10 Gbit Ethernet copper (10GBASE-T) standard which specifies a high code rate LDPC code with a fixed code length of 2048 bits, with a very high decoding throughput of 6.4 Gbps.

There is no standard for magnetic recording hard disk; however, they demand high code rate, low-error floor, and high decoding throughput. In [23], a rate-8/9 LDPC decoder with 2.1 Gbps throughput has been reported for magnetic recording. The decoder utilizes four block lengths with maximum consisting of 36864 bits.

The varied nature of applications makes the selection of a suitable hardware platform an important choice. Typical platforms for LDPC decoder implementation include programmable devices (e.g., microprocessors, digital signal processors (DSPs) and application-specific instruction set processors (ASIPs)), customized application-Specific integrated circuits (ASICs), and reconfigurable devices (e.g., FPGAs).

General purpose microprocessors and DSPs utilize strong programmability to achieve highly flexible LDPC decoding, allowing to modify various code parameters at run time. Programmable devices are often used in the design, test, and performance comparison of decoding algorithms. However, they are usually constituted by a limited number of PEs that execute in a serial manner, thus limiting the computational power to a great extent. An LDPC decoder implemented on TMS320C64xx could yield 5.4 Mb/s throughput running at 600 MHz [24]. This performance is not sufficient to support high data rates defined in new wireless standards.

Reconfigurable hardware platforms like FPGAs are widely used due to several reasons. First, they speed up the empirical testing phases of decoding algorithms which are not possible in software. Secondly, they allow rapid prototyping of decoder. Once verified, the algorithm can be employed on the same reconfigurable hardware. It also allows easy handling of different code rates and SNRs, power requirements, block lengths, and other variable parameters. However, FPGAs are suited for datapath intensive designs and have programmable switch matrix (PSM) optimized for local routing. High parallelism and the intrinsic low adjacency of parity check matrix lead to longer and complex routing, not fully supported by most FPGA devices. Some designs [16, 25], used time sharing of hardware and memories that reduces the global interconnect routing, at a cost of reduced throughput.

Customized ASICs are a typical choice which yield a dedicated, high-performance IC. ASICs can be used to fulfill high computational requirements of LDPC decoding, delivering very high throughputs with reasonable parallelism. The resulting IC usually meets area, power, and speed metrics. However, ASIC designs are limited in their flexibility and usually intended for single standard applications only: flexibility, if reached at all, comes at the cost of very long design time and nonnegligible area, power or speed sacrifices. An alternative or parallel approach is the usage of ASIPs, that greatly overcome the limitations of general purpose microprocessors and DSPs. Fully customized instruction set, pipeline and memory achieve efficient, high-performance decoding: ASIP solutions are able to provide inter- and intra-standard flexibility through limited programmability, guaranteeing average to high throughput.

3.3. Decoding Schedule. A partial parallel architecture becomes mandatory to realize flexible LDPC decoding. Generally, functional description of a generic LDPC decoder can be broken down into two parts:

(i) node processors;

(ii) interconnection structure.

A partially parallel decoder with parallelism P consists of P node processors, while an interconnection structure allows various kinds of message passing according to the implemented architecture. Based on the decoding schedule that is, TPMP or Layered decoding, the datapath can be optimized accordingly. Figure 1 shows two possible datapath implementations of partially parallel LDPC decoder which are discussed as follows.

3.3.1. TPMP Datapath. In the TPMP structure depicted in [26] for a generic belief propagation algorithm, each VN consists of 4 dual port RAMs: I, Sa, Sb, and E, as shown in Figure 1(a). RAM I stores the channel intrinsic information, while RAMs Sa and Sb manage the sum of extrinsic information for previous and current iteration, respectively, and RAM E stores the extrinsic information for current iteration. The decoding process consists of D iterations: during iteration $d + 1$, the intrinsic information ($\alpha^0_{i,j}$) fetched from RAM I is added to the contents of RAM Sa ($\sum_{i' \in \mathcal{M}(j)} \beta^d_{i',j}$). Simultaneously, the extrinsic information generated by the current parity check during the previous iteration, $\beta^d_{i,j}$, is retrieved from RAM E and subtracted from the total. The result of the subtraction is fed to the PE, which executes the chosen CN update (see Table 1). The $d + 1$ updated extrinsics are then accumulated with the iteration d ones (RAM Sb) and replace the old extrinsic information in RAM E. At iteration $d + 2$, the roles of RAM Sa and RAM Sb are exchanged.

3.3.2. Layered Datapath. The layered decoding datapath described in [27] is shown in Figure 1(b). The VN structure is simplified and consists of RAM I only, which stores α^d_j at each subiteration. Equation(9) is computed inside the check node, that consists of a PE, a FIFO, and RAM S which stores $\beta^{d-1}_{i,j}$. During iteration d, these are subtracted from

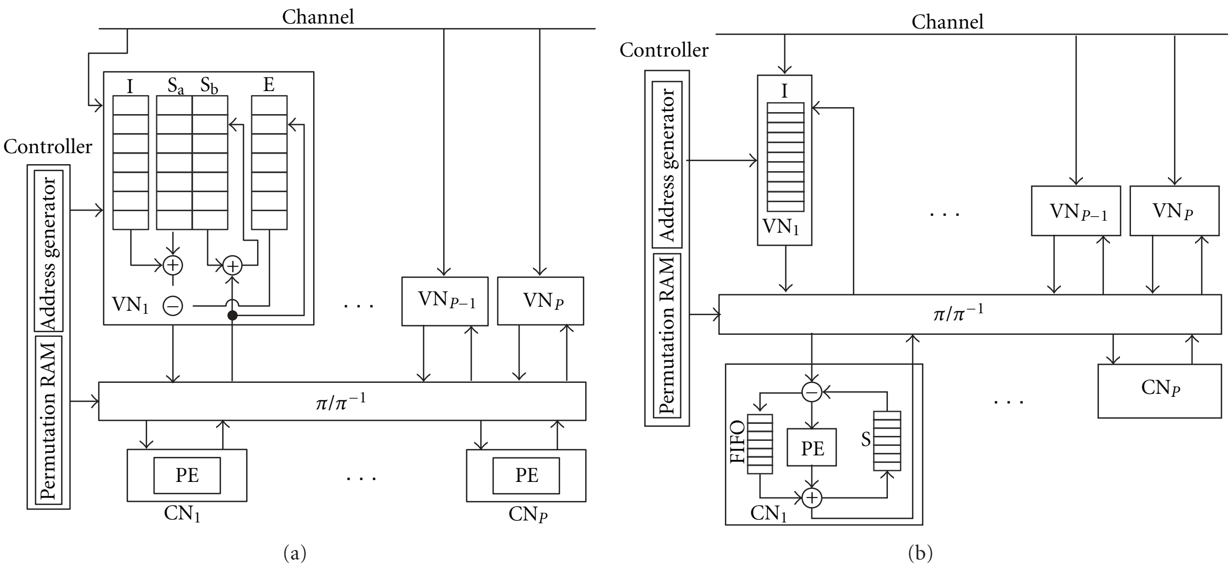

Figure 1: Generalized datapath of LDPC Decoder. (a) TPMP Decoding, (b) Layered Decoding.

the message incoming from RAM I, to generate the VN-CN message. The updated extrinsic generated by PE is added to the corresponding input coming from the FIFO, storing the resulting $\beta^d_{i,j}$ in RAM S.

In both datapath architectures described above, assignment of PEs to nodes (VNs and CNs) is determined by a given code structure and can be done efficiently by designing LDPC codes using permuted identity matrices. Considering parallelism $P = z$, the z VNs are connected to z CNs through a $z \times z$ interconnection network π/π^{-1} which has to realize the permutations of identity matrix. Typically, a highly flexible barrel shifter allows all possible permutations (rotations) of identity matrix. In some implementations, a single node type joining both VN and CN operations is present, thus changing the nature and function of the connections. The controller is typically composed of address generators (AGs) and permutation RAM (PRAM). The address generators generate the address of RAMS (I, Sa, Sb and E), while the permutation RAM generates the control signals for permutation network according to the rotation of identity matrix. Multimode flexibility is achieved by reconfiguring AGs and PRAMs each time a new code needs to be supported.

In order to realize an efficient LDPC decoder, optimization is required both at PE and interconnection level. Overall complexity and performance of decoder are largely determined by the characteristics of these two functional units. In the next two sections we will discuss them in detail and analyze various design choices aimed at realizing high-performance flexible LDPC decoder.

4. Processing Element

The PE is the core of the decoding process, where the algorithm operations are performed. Its design is an important step that heavily affects overall performance, complexity and flexibility of decoder. The PE can be designed to be serial, with internal pipelining to maximize throughput, or parallel, processing all data concurrently. Depending on this initial choice, critical design issues can arise in either latency and memory requirements or complex interconnection structures and extended logic area.

4.1. Serial PE.

4.1. Serial PE. As described in Section 2.1, the LDPC codes specified by majority of standards are based on the so-called structured LDPC codes. Considering a decoder parallelism $P = z$, as in state-of-the-art layered decoders, one sub matrix (equal to P edges) is processed per clock cycle, with one operation completed by each PE working in a serial fashion. Figure 2(a) shows a generalized architecture for serial PE implementing the Min-Sum algorithm. In Min-Sum decoding, out of all LLRs considered by a CN, only two magnitudes are of interest, that is, minimum and the second minimum. The PE works serially maintaining three variables, namely, MIN, MIN2, and INDEX. MIN and MIN2 store the minimum and second minimum of all values, respectively, whereas INDEX stores the position index of minimum value. Each time a new VN-CN message α_{ij} is received, its magnitude is compared with MIN and MIN2, possibly substituting one of the two, with consequent position storage in INDEX. For each outgoing message β_{ij}, either the value is MIN ($i \neq$ INDEX) or MIN2 ($i =$ INDEX). Such method avoids storing all VN-CN messages and results in considerable memory saving in CN kernel.

Table 3 collects some information about WiMAX and WiFi standards different parameters. Mb denotes the number of block rows in $\mathbf{H}_{\text{BASE}}$ matrix whereas W_r and W_c denote the maximum row and column weights (i.e., CN and VN degrees), respectively. A full-mode LDPC decoder for WiMAX must support 6 code rates with weights ranging from 7 to 20. Serial CN implementation is particularly suitable for this scenario, as it allows run time flexibility to process any value of CN degree with the same number of comparators, allowing efficient hardware usage. However,

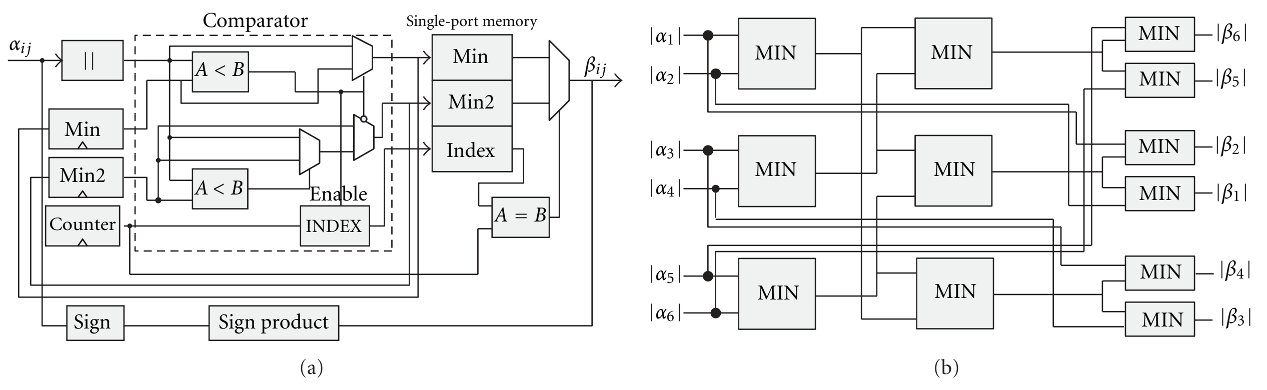

Figure 2: Min-Sum PE block scheme. (a) Serial Approach, (b) Parallel Approach.

Table 3: $\mathbf{H}_{\text{BASE}}$ parameters for WiMaX and WiFi LDPC codes.

Code rate		1/2	2/3	3/4	5/6
$\mathbf{H}_{\text{BASE}}$ matrix (WiMax/WiFi)		12×24	8×24	6×24	4×24
M_b (WiMax/WiFi)		12	8	6	4
$W_r - W_c$	WiMax	$7-6$	$11-6$	$15-6$	$20-4$
	WiFi	$8-12$	$11-8$	$15-6$	$22-4$

very large values of CN degree increase the latency and limit the achievable throughput to a great extent, requiring a high degree of parallelism to achieve medium-to-high throughputs (Table 4).

4.2. Parallel PE. Realizing high throughput decoders (supporting data rates up to few hundred Mb/s) either asks for massive parallelism or high clock frequency, resulting in significant area and power overhead. However, parallelism at CN level can bring significant increase in throughput with affordable complexity. A parallel PE manages all VN-CN messages in parallel and writes back the results simultaneously to all connected VNs. This results in lower update latency and consequently higher throughput. A parallel Min-Sum PE for $d_c = 6$ is shown in Figure 2(b). This unit computes the minimum among different choices of five out of six inputs. PE outputs the result to output ports corresponding to each input which is not included in the set, for example, $\beta_1 = \min(a_2, a_3, \ldots, a_6)$. The PE is capable of supporting all values of d_c less than or equal to 6, whereby unused inputs are initialized to $+\infty$. Supporting higher values of d_c requires additional circuitry which adds to complexity and latency of PE. As shown in the figure, the complexity of PE is dominated by logic components (e.g., comparators) and increases almost linearly with node degree. Such type of PE architectures is mostly employed to structured LDPC codes, where the check node degrees are either fixed or show small variations throughout the decoding process. To achieve code rate flexibility, the check node PE is synthesized for maximum check node degree (d_{cmax}) required by a particular application, supporting all values of d_c less than or equal to d_{cmax}.

4.3. State-of-the-Art. Flexibility as a design parameter is not always addressed as an important figure of merit, but various design techniques have been reported in the literature which can be compared in terms of throughput, complexity, and number of supported decoding modes, thus evaluating the obtainable degree of flexibility.

4.3.1. ASIC Implementations. The partially parallel decoder presented by Kuo and Willson in [32] offers a simple and tailored solution to the mobile WiMAX problem. The designed ASIC is able to work, upon reconfiguration, on all the mobile WiMAX standard LDPC codes. The quasicyclic structure of such codes allows an effective implementation of the layered decoding approach, here exploited with a variable degree of parallelism, and simple interconnection between memories and processing units. The implemented decoding algorithm is the OMS, and it is fixed. Each component of [32] is not flexible per se, but a serial architecture and programmable parallelism extend its range of usable codes to any code with parameters smaller than or equal to the WiMAX ones (block length, column and row weights, total number of exchanged information), although without guaranteeing compliance with standard throughput requirements.

In [33], the intrastandard flexibility comes together with a choice among two decoding approaches, the layered decoding and the TPMP. Although this ASIC performance has been evaluated only in case of the structured QC-LDPC codes of WiMAX, the true benefit from the dual algorithm comes in case of unstructured codes. The usually more performing layered decoding generates data collisions that are transparent to the two-phase decoding process, enabling the presence of superimposed sub-matrices in the code $\mathbf{H}$ matrix. The central processing unit consists of a Reconfigurable Serial Processing Array (RSPA) incorporating a serial Min-Sum PE and reconfigurable logarithmic barrel shifter. The RSPA can be dynamically reconfigured to choose between the two decoding modes according to different block LDPC codes. With intelligent hardware reuse via modular design the overhead due to the double decoding approach is reduced to a minimum, with an overall acceptable power consumption. The decoder operates at 260 MHz achieving

Table 4: Flexible LDPC decoder ASIC implementations. CMOS technology process (Tech), area occupation (A), A_{norm} (normalized area @ 130 nm), scheduling (sched. TDMP/TPMP), code type (C.T), block length (N), number of decoding modes supported (DM), flexibility (flex.), dt (design time), rt (run time reconfigurable), decoding iterations (It.), throughput (T.P), clock frequency (f), PE structure (PE) (serial Se, parallel Pa), number of datapaths (Dp), throughput area ratio (TAR)(Mb/s × It/mm^2 = T.P × It/A_{norm}), decoding efficiency (DE) (bits/cycle = T.P × It/f), and flexibility efficiency (FE) (DM × bits/cycle/mm^2 = DE × DM/A_{norm}).

Design	Tech. (nm)	A mm^2	A_{norm} mm^2	Sched.	C.T	N Bits	DM	Flex.	It.	T.P Mb/s	f MHz.	PE	Dp	TAR	DE	FE
		1.337	5.348		QC-WiMAX	576–2304	114		25, 20	48–333			24–96	1245.3	16.7	356.0
[28]	65	3.861	15.444	TDMP	DVB-S2	64800	20	D.T	50, 15	60–708	400	Se	90	687.6	26.6	34.4
		1.023	4.092		QC-WiFi	648–1944	12		25, 20	54–281			27–81	1373.4	14.1	41.4
[29]	65	0.51	2.04	TDMP	WiMedia	1200–1320	8	R.T	5, 3	1120–1220	264	Pa	90	1794.1	13.9	54.5
[30]	90	9.60	20.03	TDMP	DVB-S2	64800	20	R.T	15	181–998	320	Se	360	747.4	46.8	46.7
[31]	90	0.42	0.87	TDMP	QC-WiFi	1944	12	R.T	30	43	294	Pa	324	1482.7	4.4	60.7
[32]	180	3.39	1.768	TDMP	QC-WiMAX	576–2304	114	R.T	10	68	100	Se	24	384.6	6.8	438.5
[33]	130	6.3	6.3	TDMP/TPMP	Block-LDPC	576–2304	114	R.T	15	205	260	Se	24–96	487.5	11.8	213.5
[34]	130	2.46	2.46	Overlapped TDMP	QC-WiMAX	576–2304	114	R.T	15	248–287	150	Se	96	1750.0	28.7	1330.0
[27]	130	3.843	3.843	TDMP/TPMP	QC-WiMAX	576–2304	114	D.T	10,15	83–610	333	Se	24–96	2380.9	18.3	4.8
[35]	90	0.679	1.416	TDMP	QC-WiMAX	576–2304	114	R.T	8–12	200	400	Pa	16	1694.9	4.0	322.0
[36]	90	6.25	13.04	TDMP	QC-WiMAX	576–2304	114	R.T	20	105	150	Se	24–96	161.0	14.0	122.4
[37]	130	8.29	8.29	TDMP	QC-WiMAX	576–2304	19	R.T	2–8	222	83.3	Pa	4–8	214.2	5.3	12.1
[38]	130	4.94	4.94	TPMP	Arbitrary	1536	Arbitrary	R.T	2–8	86	125	Pa	Arbitrary	139.3	1.37	N/A

a handsome throughput which meets the WiMAX standard specifications.

When designing an efficient multi mode decoder a typical approach is to find similarities between different modes and then implementing common parts as reusable hardware components. Controlling the data flow between reusable components guarantees multi mode flexibility. One of such efforts is the work by Brack et al. [27]. It portraits an IP core of a full mode LDPC decoder that can be synthesized for a selected code rate specified by WiMAX standard. The unified decoder architecture proposes two different datapaths for TPMP and layered decoding, as in [33], and combines them in a single architecture sharing the components common to both. Code rate and codeword size flexibility is achieved by realizing serial check node PEs, and the chosen decoding algorithm is $\lambda - 3$ Min [43].

As discussed in Section 3.3.2, a partially parallel TDMP decoder performs serial scanning of block rows of $\mathbf{H}_{\text{BASE}}$ matrix. The check node reads the bit-LLR connected to a block row serially and stores them in FIFO whose size is proportional to row weight W_r. For multirate irregular QC decoders, the utilization ratio of FIFO is low for smaller W_r. In addition, due to random location of nonzero submatrices and correlations between consecutive block rows of $\mathbf{H}_{\text{BASE}}$, extrinsic information exchange can lead to memory access conflicts. These limitations were addressed by Xiang et al. in [34], whereby the authors presented an overlapped TDMP decoding algorithm. The design proposes a block row and column permutation criterion in order to reduce correlation between consecutive rows and uniform distribution of zero and nonzero matrices in columns, with a smart memory management technique. The resulting decoder is a full-mode, QC LDPC decoder for WiMAX. The decoder achieves a maximum throughput of 287 Mb/s, with support for other similar QC-LDPC codes.

An interesting way to tackle the flexibility issue is proposed in [35]. Here the different code parameters are handled via what has been called processing task arrangement. This work presents a decoder based on the decoding approach described in [44], layered message passing decoding with identical core matrices (LMPD-ICMs). LMPD-ICM is a variation of the original layered decoding: the $\mathbf{H}$ matrix is partitioned in several layers, with each layer yielding a core matrix. This consists of the nonzero columns of that layer. The resulting core matrix is further divided into smaller and identical tasks. Applying LMPD-ICM to a QC-LDPC code reveals that core matrices of layers are column-permuted versions of each other and show similarities not only among different layers in a single code, but also among different codes within a same code class. This technique is applied to QC-LDPC codes for WiMAX in [35], and a novel task arrangement algorithm is proposed to assign the processing operation for a variety of QC-LDPC codes to different PEs. The design features four-stage pipelining for task execution, flexible address generation to support multirate decoding, and early termination strategy which dynamically adjusts the number of iterations according to SNR values to save power. The decoder achieves a moderate throughput of 200 Mb/s at 400 MHz frequency utilizing parallelism P of 4.

A single design flow is exploited in [28] to provide three different implementations, each supporting a different standard. The adaptive single-phase decoding (ASPD) [45] scheduling is enforced, that allows to detach the decoder's memory requirements from the weight of rows and columns of the $\mathbf{H}$ matrix, leaving them dependent on the codeword length only. Though sacrificing up to 0.3 dB in BER performances, this technique accounts for 60–80% reduction in memory bits. The offset Min-Sum decoding algorithm is employed: different sizes of memories are able to comply with DVB-S2, 802.11n, and 802.16e standards. At run time, the standard serial node architecture enables intrastandard flexibility.

A classical layered scheduling is used in the DVB-S2 decoder proposed in [30]: the 360 PEs, whose architecture is detailed along with the iteration timing, are able to process a whole layer concurrently. A 360×360 barrel shifter manages the interlayer communication: since the number of rows that compose the layer never change in DVB-S2, a change of code will mean a different workload on the communication structure, but very easy reconfiguration.

The work in [36] presents a reconfigurable full-mode LDPC decoder for WiMAX. A so-called phase overlapping algorithm similar to TDMP is proposed which resolves the data dependencies of CNs and VNs of consecutive subiterations and overlaps their operation. The proposed decoder features serial check nodes with Min-Sum algorithm implementation. Parallelism of 96 yields a throughput of 105 Mb/s at 20 iterations.

In addition to serial check node architectures, the state-of-the-art for flexible LDPC decoders also reports some solutions utilizing parallel check nodes. The work in [37] proposes a reconfigurable multimode LDPC decoder for Mobile WiMAX. The authors applied the matrix reordering technique [46] to the $\mathbf{H}_{\text{BASE}}$ matrix of rate 1/2 WiMAX. This improved matrix reordering technique allowing overlapped operation of CNs and VNs and results in 68.75% reduction in decoding latency compared to nonoverlapped approach. A reconfigurable address generation unit and improved early stopping criterion help to realize a low-power flexible decoder which supports all the 19 block lengths (576–2304) of WiMAX. Parallel check nodes implementing Min-Sum algorithm help to achieve a throughput of 222 Mb/s with low frequency of 83.3 MHz and parallelism of 4.

The work in [38] features a parallel check node based on divided group comparison technique, adaptive code length assignment to improve decoding performance, and early termination scheme. The proposed solution is run time programmable to support arbitrary QC-LDPC codes of variable codes lengths and code rates. However, no compliance with standardized codes is guaranteed. A reasonable throughput of 86 Mb/s at 125 MHz is achieved. In [47], the authors presented a parallel check node incorporating the value reuse property of Min-Sum algorithm, for nonstandardized, rate 0.5 regular codes.

The WiMedia standard [48] requires very high throughput: the work by Alles et al.[29] manages to deliver more than 1 Gb/s throughputs for most code lengths and rates. The result is achieved via the instantiation of 3 PEs with internal parallelism of 30: the similarity of WiMedia codes with the QC-LDPC of WiMAX allows a multiple submatrix-level parallelism in the decoder.

A more technological point of view is given in [31], where low-power VLSI techniques are used in a 802.11n LDPC decoder design. The decoder exploits the TDMP VSS technique with a 12-datapath architecture: separate variable-to-check and check-to-variable memory banks are instantiated, one per type for each datapath. Each of these "macrobanks" contains 3 "microbanks," each storing 9 values per word. The internal degree of parallelism is effectively sprung up to 12×27. VLSI implementation is efficiently tackled in less-than-worst case thanks to VOS [49] and RPR [50] techniques, saving area and power consumption.

4.3.2. ASIP Implementations. Future mobile and wireless communication standards will require support for seamless service and heterogeneous interoperability: convolutional, turbo, and LDPC codes are established channel coding schemes for almost all upcoming wireless standards. To provide the aforementioned flexibility, ASIPs are potential candidates. The state-of-the-art reports a number of design efforts in this domain, thanks to good performance and acceptable degree of flexibility.

The work portrayed in [39] outlines a multicore architecture based on an ASIP concept. Each core is characterized by two optimized instruction sets, one for LDPC codes and one for turbo codes. The complete decoder only requires 8 cores, since each of them can handle three processing tasks at once. The simple communication network maintains this parallelism, allowing for efficient memory sharing and collision avoidance. The intrinsic flexibility of the ASIP approach allows multiple standards (WiFi, WiMAX, LTE, and DVB-RCS) to be easily supported: the exploitation of the diverse datapath allows a very high best case throughput, while the reduced network parallelism keeps the complexity low.

A possible dual turbo/LDPC decoder architecture is described in [40]. Here, a novel approach to processing element design is proposed, allowing a high percentage of shared logic between turbo and LDPC decoding. The TDMP approach allows the usage of BCJR decoding algorithm for both codes, while the communication network can be split into smaller supercode-bound interleavers, effectively merging LDPC and turbo tasks.

In [41], the authors proposed a flexible channel coding processor FlexiCHAP. The proposed ASIP extends the capabilities of previous work FlexiTrep [51] and is capable to decode convolutional codes, binary/duobinary turbo codes and structured LDPC codes. The proposed ASIP is based on the single-instruction multiple datapath (SIMD) paradigm [52], that is, a single IP with an internal data parallelism greater than one, and processes whole submatrix of parity check matrix with single instruction. Multistandard multi mode functionality is achieved by utilizing 12-stage pipeline and elaborated memory partitioning technique. The proposed ASIP is able to decode binary turbo codes up to 6144 information bits, duobinary turbo codes and convolutional codes upto 8192 information bits. LDPC decoding capability of block length up to 3456 bits and check node degree up to 28 is sufficient to cover all code rates of WiMAX and WiFi standards. The ASIP achieves payloads of 237 Mb/s and 257 Mb/s for WiMAX and WiFi, respectively.

The solution proposed in [42] makes use of a single SIMD ASIP with maximum internal parallelism of 96. A combined architecture is strictly designed for LDPC codes, but the supported ones range from binary (WiFi and WiMAX) to nonbinary (Galois Field of order 9): the decoding approach is turbo decoder-based. While the decoding mode can be changed at runtime, flexibility is guaranteed at design time, by instancing wide enough rotation engines for the different LDPC submatrix sizes, and a sufficient number of memories. These memories dominate the area occupation, mainly due to the non-binary decoding process.

Tables 4 and 5 summarize the specifications of various state-of-the-art ASIC and ASIP solutions for flexible LDPC decoders discussed above. To simplify the comparison, the area of each decoder has been scaled up to 130 nm process represented as normalized area (A_{norm}). A parameter called throughput to area ratio (TAR) defined as $\mathrm{TAR} = \mathrm{Throughput} \times \mathrm{It}/\mathrm{Area}$ has also been included in the table to evaluate the area efficiency of proposed decoders. Another metric named decoding efficiency (DE) given as $\mathrm{DE} = (\mathrm{Throughput} \times \mathrm{It.})/f$ [53] has been defined to give a good comparison regardless of different clock frequencies. DE gives the number of decoded bits per clock cycle per iteration, while Dp is the total degree of parallelism, taking into account both the number of PEs and their possible multiple datapaths.

To evaluate the effective flexibility of each decoder, and its cost, a metric called *flexibility efficiency* is introduced, and computed as

$$\mathrm{FE} = \frac{\mathrm{DE} \times \mathrm{DM}}{A_{\mathrm{norm}}}. \tag{14}$$

It gives a measure of each decoder's flexibility through its different decoding modes (DMs), taking in account the normalized throughput performances (DE) in relation to the normalized area occupation A_{norm}. The metric is applied to ASIC decoders only, since in these cases the cost of flexibility is reflected on the area much more directly than in the ASIP case.

As shown in Table 4, the work in [27] dominates in terms of throughput and TAR but offers only design time flexibility (effective $\mathrm{D.M} = 1$): for this reason, its FE value is very low. On the contrary, the design time flexibility of [28] results in a runtime flexibility once the standard to be supported has been chosen: since the implementation of the WiMAX decoder requires a higher number of codes to be supported at the same time than DVB-S2 and WiFi, its FE will be higher. Best DE value is held by the DVB-S2 decoder presented in [30], together with an average TAR. Explicitly enabling the decoding of just 20 codes, however, lowers its FE measure. Among serial PE-based run time flexible solutions discussed above, the work in [34] achieves very high throughput, TAR and DE with a small area occupation of 2.46 mm^2, yielding the best FE of all decoders. A full-mode reconfigurable solution based on parallel check node in [35]

Table 5: Flexible LDPC decoders ASIP Implementations. CMOS technology process (Tech), area occupation (A), normalized area (A_{norm})@ 130 nm, code type (C.T), flexibility (Flex.) design time (D.T), run time (R.T), maximum throughput (T.P), maximum iterations (It.), number of datapaths (Dp), operating frequency (f), processing element (PE) (serial Se, parallel Pa), throughput area ratio (TAR)(Mb/s × It/mm^2 = t.p × It/A_{norm}), and decoding efficiency (DE) (bits/cycle = t.p × It/f).

Design	Tech. (nm)	A mm^2	A_{norm} mm^2	C.T	Flex.	It.	T.P Mb/s	f MHz.	Dp	PE	TAR	DE
Multicore ASIP [39]	90	2.6	5.42	LDPC-{WiMAX,WiFi} Turbo-{BTC-LTE,DBTC-WiMAX}	R.T	10 6	{312, 263} {173, 173}	500	24	Se	{575.6, 485.2} {191.4, 191.4}	{6.24, 5.26} {2.07, 2.07}
2D NOC ASIP [40]	130	N/A	N/A	Turbo LDPC	R.T	8	86.5 11.2	200	16	Se	N/A	3.46 0.448
FlexiCHAP [41]	65	0.62	2.48	LDPC-{WiMAX, WiFi} Turbo {BTC,DBTC}	R.T	10–20 5	{237, 257} {18.6, 37.2}	400 (max.)	27	Se	{955.6, 1036.3} {37.5,75.0}	{5.9, 6.42} {0.23, 0.46}
Bin/non-Bin [42]	65	3.4	13.6	Bin LDPC-{WiMAX, WiFi} Non-Bin LDPC {GF(9)}	R.T	10 1	90 12.5	400	96	Se	66.2 0.92	2.25 0.03

achieves a handsome throughput of 200 Mb/s and the best FE among the parallel node solutions. Its A_{norm} of 1.416 mm^2 is the minimum among all WiMAX solutions discussed above, but with very low DE stains overall performance.

Among the ASIP solutions (Table 5), the work in [40] cannot effectively be compared to the others in terms of area, not providing complete estimations. The work in [41] yields the higher TAR and DE in LDPC mode: the solution proposed in [39], however, yields the best decoding efficiency and the top TAR in turbo mode, while at the same time reaching a very good DE in LDPC mode too.

5. Interconnection Structures

As shown through the previous sections, in the great majority of current LDPC decoders, some kind of intradecoder communication is necessary. Except for very few single-core implementations based on the single-instruction single datapath (SISD) paradigm, the need for message routing or permutation is a constant throughout the wireless communication state of the art. As a first classification, two scenarios can be roughly devised:

(i) *single PE architectures*: some state-of-the-art decoders propose single core solutions with internal parallelism greater than one, that rely on smart memory sharing and on programmable permutation networks. These decoders make often use of TDMP and VSS, that require either reduced communication or very regular patterns: the involved interconnection structures are simple;

(ii) *partially parallel architectures*: referring to the graph representation of the LDPC **H** matrix within a selected decoding approach, it is possible to map the graph nodes onto a certain number of processing cores. In the partially parallel approach, the number of graph nodes is much higher than the PEs. Each node is connected to a set of other nodes distributed on the available PEs: different nodes will have different links, resulting in a widely varied PE-to-PE communication pattern. This situation calls for flexible and complex interconnection structures.

5.1. Shift and Shuffle Networks.

Structured LDPC codes decoding, regardless of their implementation, often require shift or shuffle operation to route information between PEs or to/from memories. This is particularly true for some kinds of LDPC codes, as QC-LDPC and *shift*-LDPC [54].

The barrel shifter (BS) is a well-known circuit designed to perform all the permutations of its inputs obtainable with a shift operation, thus being well suited for the circularly shifted structure of QC-LDPC **H** matrix.

Rovini et al. in [55] exploit the simple structure of the barrel shifter to design a circular shifting network for WiMAX codes. This network must be able to handle all the different submatrix sizes of the standard, thus effectively becoming a multisized circular shifting (MS-CS) network. This MS-CS network is composed of a number of $B \times B$ BSs, where B is the greatest common divisor among all the supported block sizes. Each BS rotates of the same shift amount all the blocks of B data, that are subsequently rearranged by an adaptation network into the desired order according to the current submatrix size. Implementation results show that the proposed MC-CS network outperforms in terms of complexity previous similar solutions as [56–59], with a saving ranging from 30.4% to 67.2%.

In [36] is designed a shift network for WiMAX standard, based on a self-routing technique. The network is sized to handle the largest submatrix size of the standard, 96: when decoding a smaller code, dummy messages are routed as well, with a dedicated flag. Two stages of barrel shifters provide the shift function to real and dummy messages alike, together with a single permutation network: a lookup engine finally selects the useful ones basing its decision on the flag bits, shift size and submatrix size.

Barrel shifters, though providing the most immediate implementation of the shift operation, often lack the necessary flexibility to directly tackle multiple block sizes. For this reason, they are usually joint to more complex structures.

One of the most common implementations among the simplest interconnection structures is the Benes network (BeN). This kind of network is a rearrangeable nonblocking network frequently used as a permutation network. Defining S_M the number of inputs and outputs, an $S_M \times S_M$ BeN can perform any permutation of the inputs creating a one-to-one relation with the outputs with $(S_M/2 \times (2\log_2 S_M - 1))2 \times 2$ switches. Its stand-alone use is thus confined to situations in which sets of data tend not to intertwine: its range of usage, though, can be extended through smart scheduling.

In [60] a flexible ASIC architecture for high-throughput *shift*-LDPC decoders is depicted. *Shift*-LDPC codes are subclass of structured LDPC: the **H** of an $(N, M)(M_b, N_b)$ *Shift*-LDPC is structured as

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{1,1} & \mathbf{P_m}\mathbf{H}_{1,1} & \cdots & \mathbf{P_m}^{N_b-1}\mathbf{H}_{1,1} \\ \mathbf{H}_{2,1} & \mathbf{P_m}\mathbf{H}_{2,1} & \cdots & \mathbf{P_m}^{N_b-1}\mathbf{H}_{2,1} \\ \cdots & \cdots & \ddots & \cdots \\ \mathbf{H}_{M_b,1} & \mathbf{P_m}\mathbf{H}_{M_b,1} & \cdots & \mathbf{P_m}^{N_b-1}\mathbf{H}_{M_b,1} \end{bmatrix}, \tag{15}$$

where $N \times M$ are the dimensions of the H matrix, and the leftmost M_b submatrices are randomly row-permuted versions of the $z \times z$ identity matrix $\mathbf{I}$. Matrix $\mathbf{P_m}$ identifies a $z \times z$ permutation matrix, obtained by cyclically shifting right the columns of $\mathbf{I}$ by a single position. The operations involved in the definition of each the $M_b \times N_b$ submatrices guarantee that $\mathbf{P_m}^k\mathbf{H}_{i,1}$ is $\mathbf{P_m}^{k-1}\mathbf{H}_{i,1}$ with the rows shifted up one position, so all matrices of row i can be found cyclically shifting $\mathbf{H}_{i,1}$.

To exploit at best the proprieties of *Shift*-LDPC, a VSS scheme has been selected, along with a highly parallel implementation: z variable node units (VNUs) and M check node units (CNUs) perform a whole iteration in N_b steps. Since every **H** submatrix is a shifted version of the previous one, also the connections between z VNUs and z CNUs shift cyclically every clock cycle: this observation leads to the joint design of the Benes global permutation network and the CN shuffler.

The BeN is used to define the links between VNUs and CNUs: these links are static, once the parameters z, N_b, and the structure of the M_b leftmost $\mathbf{H}_{x,1}$ have been fixed. Its high degree of flexibility is exploited to guarantee support over a variety of different codes. The inter-CN communication required by the VSS approach is handled by the CN shuffle network. Its function is to cyclically shift the submatrix rows assigned to each CNU: this means that while each CNU will be physically connected to the same VNU for the whole decoding, the row of the $\mathbf{H}$ matrix they represent will change. The BeN has consequently no need to be rearranged.

The flexible decoder is able to achieve 3.6 Gb/s with an area of 13.9 mm^2 in 180 nm CMOS technology: the area occupation is relatively small w.r.t. the very high degree of parallelism thanks to the nonuniform 4-bit quantization scheme adopted.

In [61], a SIMD-based ASIC is proposed for LDPC decoding over a wide array of standards. The ASIC is composed of 12 parallel datapaths able to decode both turbo and LDPC codes through the BCJR algorithm. As most of the SIMD cores, the decoder handles communication by means of shared memories. Memory management can be challenging, especially in case the parallel datapaths are assigned to fractions of the same codeword. According to [62], it is possible to avoid collisions in such cases with adhoc mapping of the interleaving laws together with one (for LDPC) or two (for turbo) permutation networks to interface with memories. These two networks are implemented in [61] with 8×8 BeN, the one at the input of the extrinsic values memory being transparent in LDPC mode.

Not every supported standard require all the 12 datapaths to be active: the chosen parallelism is the minimum necessary for throughput compliance, and the same can be said for the working frequency. The implementation results show full compliance with WiMAX, WiFi, 3GPP-HSDPA, and DVB-SH, at the cost of 0.9 mm ^{2} in 45 nm CMOS, technology, and total power consumption of 86.1 mW.

One of the limitations of the traditional BeN is the number of its inputs and outputs, that are bound to be a power of 2. However, LDPC decoders often need a permutation of different size: for example, WiMAX codes require shift permutations of sizes corresponding to the possible expansion factors, that is, from 24 to 96 with steps of 4. In [63], an alternative switch network is designed that makes use of 3×3 switches as well, leading to a more hardware-efficient design. The introduction of 3×3 switches allows in fact $S_M = 3 \times 2^i$. A fully compliant WiMAX LDPC decoder shift operation can thus be implemented with a novel 96×96 switch network: it requires $3 \times 2^i + 3 \times 2^i \log_2 2^i = 5762 \times 2$ switches, against the 832 necessary for a traditional 128×128 BeN. Together with efficient control signal generation, this solution outperforms in terms of both complexity and flexibility other modified Benes-based decoders as [57, 64], that exploit a secondary BeN to rearrange the first one.

In [65], Lin et al. propose an optimized Benes-based shuffle network for WiMAX. Unlike [63], the starting point of the design is the nonoptimized 128×128 BeN: from here all the switches are removed where no signal is passed, whereas switches with fixed output are replaced by wires. This adhoc trimming technique, together with an efficient algorithm for control signal creation, allow a 26.6%–71.1% area reduction with respect to previously published shift network solutions like [27, 55, 66].

Similar to the BeN is the Banyan network (ByN) [67], that can be seen as a trade-off between the flexibility of the BeN and the complexity of the BS. Although not non-blocking in general, the ByN is non-blocking in case of shift operations only. Moreover, it is composed of averagely half of the 2×2 switches of a BeN and requires fewer control signals.

The work described in [68] portrays a highly parallel shuffle network based on the ByN paradigm. Like the BeN, also ByN is bound to a power-of-two number of inputs: as Oh and Parhi have done in [63], also here the introduction of 3×3 switches allows to handle WiMAX standard various submatrix sizes. The implemented decoder guarantees a very high degree of flexibility with complexity lower or comparable to [36, 63–65].

5.2. Networks-on-Chip. Networks-on-Chip (NoCs) [69] are versatile interconnection structures that allow communication among all the connected devices through the presence of routing elements. Recently, LDPC decoders for both turbo and LDPC codes based on the NoC paradigm have been proposed, thanks to the intrinsic flexibility of NoCs. NoC-based decoders are multiprocessor systems composed of various instantiations of the same IP associated to a routing element, which are linked in defined pattern. This pattern can be represented with a graph, in which every node corresponds to a processor and a router: the arcs are the physical links among routers, thus identifying a topology.

In [70], a De Bruijn topology [71] NoC is proposed for flexible LDPC decoders. Since the TPMP has been selected, the NoC must handle the communication between VNs and CNs. The NoC design, however, is completely detached from code and decoding parameters, effectively allowing usage for any LDPC code. The router embeds a modified shortest path routing algorithm that can be executed in 1 clock cycle, together with deadlock-free and buffer-reducing arbitration policies and is connected to its PE via the network interface. The network is synthesized and compared to other explored network topologies, as the 2-dimensional mesh [72], Benes [73] and MDN [74]: the degree of flexibility and scalability that the proposed topology guarantees is unmatched.

The performance of another topology, the 2D toroidal mesh, is evaluated in [40]. The routing element implements the near-optimal X-Y routing for the torus/mesh [75]. A whole set of communication-centric parameters is varied in order to evaluate the impact of the network latency on the whole decoder performance. It has been shown that small PE sending periods R, that is, cycles between two available data from the processor, increase latency to unsustainable levels, with the smallest values at $R \approx 7$. Also, the variations in throughput due to different NoC parallelisms are shaded by the impact of latency.

The work in [76] describes a flexible LDPC decoder design tackling the communication problem with two different NoC solutions. The first network is a De Bruijn NoC

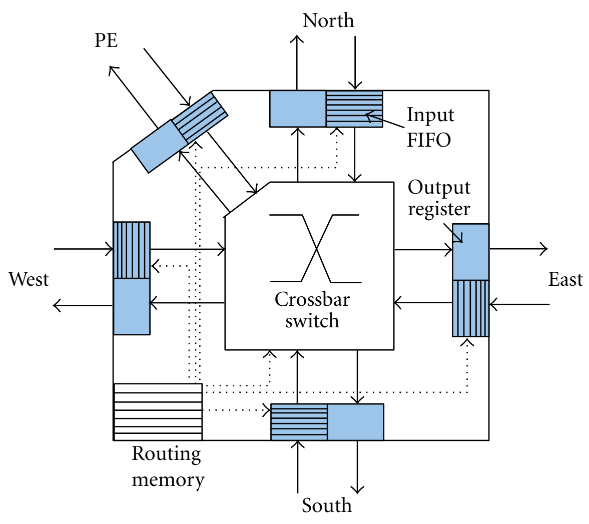

FIGURE 3: ZONoC routing element.

adopting online dynamic routing, implementing the same modified shortest path algorithm described in [70], while the second, a 2D torus, is based on a completely novel concept named zero overhead NoC (ZONoC). Given a mapping of VNs and CNs over a topology, the message exchange pattern is deterministic, along with the status of the network at each instant. The ZONoC exploits this propriety by running offline simulations and storing routing information into dedicated memories, effectively wiping out the time spent for routing and traffic control. Overall network complexity is scaled down, since no routing algorithm is necessary; FIFO length can be trimmed to the minimum necessary, while router architecture is as simple as possible (Figure 3). A cross-bar switch controlled by the routing memory receives messages from the FIFOs connected to its PE and other routers, while outgoing messages are sent to as many output registers. Implementation results show a significant reduction in complexity with respect to [28, 36, 56, 72] and comparable or superior throughput.

5.3. Reducing the NoC Penalty. The NoC approach guarantees a very high degree of flexibility and, in theory, a NoC-based decoder can reach very high throughput. The achievable throughput is proportional to the number of PEs: but increasing the size of the network means rising the latency, and thus degrading performance back. Very few state of the art solutions have managed to solve this problem, and those who do suffer from large complexity and power consumption. We have tried to overcome these shortcomings in some recent works.

5.3.1. NoC-Based WiMAX LDPC Decoder. The solution described in [76] supports the WiMAX standard LDPC codes, but does not guarantee a high enough throughput. Stemming from it, we have developed an LDPC ZONoC-based decoder fully compliant with WiMAX standard: although having a more convoluted graph structure, relies on a smaller number of exchanged messages and guarantees $a \times 2$ factor in convergence speed. We designed a sequential PE implementing the normalized Min-Sum decoding algorithm, as described by Hocevar in [77]: unlike in [76], we adopt the layered decoding approach. The PE architecture is independent of code parameters, and the memory capacity sets the only limit to the size of supported codes. Together with the PE, we devised a decoder reconfiguration technique to upload the data necessary for routing and memory management when switching between codes.

In order to comply with WiMAX standard throughput requirements, the size of the 2D torus mesh has been risen from 16 nodes to 25. As detailed in [78], the decoder guarantees more than 70 Mb/s for all rates and block sizes of the standard, with an area of 4.72 mm^2 in 130 nm CMOS technology.

5.3.2. Bandwidth and Power Reduction Methods. While the former decoder is compliant with WiMAX in worst case, that is, when the maximum allowed number of iteration is performed, a codeword is averagely corrected with fewer iterations: the unnecessary iterations significantly contribute to the NoC high-power consumption. In [79], two methods aimed at reducing power and increasing throughput are studied and implemented. The first one is the iteration early-stopping (ES) criterion proposed in [80], that allows to stop the decoding when all the information bits of a codeword are correct, regardless of the redundancy bits. The other is a threshold-based message stopping (MS) criterion, that reduces the traffic load on the network by avoiding injection of values which carry information about high-probability correct bits.

Various combinations of the two methods have been tried, together with different parallelisms of the 2D torus mesh. Implementation of the ES criterion requires a dedicated processing block with minimal PE modifications, while MS requires a threshold comparison block for each PE and switching to online dynamic routing. This is necessary since stopping a message invalidates the statically computed communication pattern. While the ES method guarantees an average 10% energy per frame decoding reduction regardless of the implementation, the MS method's results change with the size of the NoC. Since stopped messages can lead to additional errors, a performance sacrifice must be accepted: among the solutions presented in [79], with 0.3 dB BER loss, a 9-PE NoC is sufficient to support the whole WiMAX standard.

5.3.3. NoC Analysis for Turbo/LDPC Decoders. In [81], an extensive analysis of performance of various NoC topologies is performed in the context of multiprocessor turbo decoders. Flexibility can be explored also in terms of types of code supported: in [82], we have consequently extended the topology analysis to LDPC codes, in order to find a suitable architecture for a dual turbo/LDPC codes. As a case of study, we focused our research on the WiMAX codes.

The performance of a wide set of topologies (ring, spidergon, toroidal meshes, honeycomb, De Bruijn, and Kautz) has been evaluated in terms of achievable throughput and complexity, considering different parallelisms. Exploiting a modified version of the cycle-accurate simulation tool described in [81], a range of design parameters has been taken in consideration, including data injection rate, message collision management policies, routing algorithms, node addressing modes, and structure of the routing element, allowing to span from a completely adaptive architecture to a ZONoC-like precalculated routing.

The simulations revealed the Kautz topology [83] to be the best trade-off in terms of throughput and complexity between LDPC and turbo codes, with a partially adaptive router architecture and FIFO-length based routing. Two separate PEs for turbo and LDPC codes have been designed: different NoC and PE working frequencies allow to trim the message injection rate in the network. The full decoder, complete with turbo and LDPC separated PEs, has been synthesized with 90 nm CMOS technology: the decoder is compliant with both turbo and LDPC throughput requirements for all the WiMAX standard codes. Worst-case throughput results overperform the latest similar solutions as [39, 41, 61, 84], with a small area occupation and particularly low-power consumption (59 mW) in turbo mode.

6. Conclusions

A complete overview of LDPC decoders, with particular emphasis on flexibility, is drawn. Various classifications are depicted, according to degree of parallelism and implementation choices, focusing on common design choices and elements for flexible LDPC decoders. An indepth view is given over the PE and interconnection part of the decoders, with comparison with the current state-of-the-art, the latest work by the authors on NoC-based decoders is briefly described.

References

[1] A. Morello and V. Mignone, “DVB-S2: the second generation standard for satellite broad-band services,” *Proceedings of the IEEE*, vol. 94, no. 1, pp. 210–227, 2006.

[2] J. Lörincz and D. Begušić, “Physical layer analysis of emerging IEEE 802.11n WLAN standard,” in *Proceedings of the 8th International Conference Advanced Communication Technology (ICACT ’06)*, pp. 189–194, February 2006.

[3] The IEEE p802.3an, 10GBASE-T task force, http://www.ieee802.org/3/an/.

[4] “IEEE standard for local and metropolitan area networks part 16: Air interface for fixed and mobile broadband wireless access systems amendment 2: physical and medium access control layers for combined fixed and mobile operation in licensed bands and corrigendum 1,” *IEEE Std 802.16e-2005 and IEEE Std 802.16-2004/Cor 1-2005 (Amendment and Corrigendum to IEEE Std 802.16-2004)*, 2006.

[5] R. Gallager, “Low-density parity-check codes,” *IEEE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[6] R. M. Tanner, “A recursive approach to low complexity codes,” *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, 1981.

[7] M. M. Mansour and N. R. Shanbhag, “A 640-Mb/s 2048-bit programmable LDPC decoder chip,” *IEEE Journal of Solid-State Circuits*, vol. 41, no. 3, pp. 684–698, 2006.

[8] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[9] G. Masera, F. Quaglio, and F. Vacca, “Finite precision implementation of LDPC decoders,” *IEE Proceedings on Communications*, vol. 152, no. 6, pp. 1098–1102, 2005.

[10] N. Wiberg, *Codes and decoding on general graphs*, Ph.D. dissertation, Linkoping University, Linkoping, Sweden, 1996.

[11] M. Daud, A. Suksmono, Hendrawan, and Sugihartono, “Comparison of decoding algorithms for LDPC codes of IEEE 802.16e standard,,” in *Proceedings of the 6th International Conference on Telecommunication Systems, Services, and Applications (TSSA ’11)*, pp. 280–283, October 2011.

[12] J. Chen and M. P. C. Fossorier, “Near optimum universal belief propagation based decoding of LDPC codes and extension to turbo decoding,” in *Proceedings of the IEEE International Symposium on Information Theory (ISIT ’01)*, p. 189, June 2001.

[13] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X. Y. Hu, “Reduced-complexity decoding of LDPC codes,” *IEEE Transactions on Communications*, vol. 53, no. 8, pp. 1288–1299, 2005.

[14] M. Martina, G. Masera, S. Papaliralabos, P. Mathiopoulos, and F. Gioulekas, “On practical implementation and generalization of max* operation for Turbo and LDPC decoders,” *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 4, pp. 888–895, 2012.

[15] E. Yeo, P. Pakzad, B. Nikolić, and V. Anantharam, “High throughput low-density parity-check decoder architectures,” in *Proceedings of the IEEE Global Telecommunicatins Conference (GLOBECOM ’01)*, pp. 3019–3024, November 2001.

[16] M. Mansour and N. Shanbhag, “Memory-efficient turbo decoder architectures for LDPC codes,” in *Proceedings of the IEEE Workshop on Signal Processing Systems (SIPS ’02)*, pp. 159–164, October 2002.

[17] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, 1974.

[18] J. Zhang and M. Fossorier, “Shuffled belief propagation decoding,” in *Proceedings of the 36th Asilomar Conference on Signals Systems and Computers*, vol. 1, pp. 8–15, November 2002.

[19] R. M. Tanner, D. Sridhara, A. Sridharan, T. E. Fuja, and D. Costello, “LDPC block and convolutional codes based on circulant matrices,” *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 2966–2984, 2004.

[20] A. J. Blanksby and C. J. Howland, “A 690-mW 1-Gb/s 1024-b, rate-1/2 low-density parity-check code decoder,” *IEEE Journal of Solid-State Circuits*, vol. 37, no. 3, pp. 404–412, 2002.

[21] L. Fanucci, P. Ciao, and G. Colavolpe, “VLSI design of a fully-parallel high-throughput decoder for turbo gallager codes,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 89, no. 7, pp. 1976–1986, 2006.

[22] V. Nagarajan, N. Jayakumar, S. Khatri, and O. Milenkoviç, “High-throughput VLSI implementations of iterative decoders and related code construction problems,” in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM ’04)*, vol. 1, pp. 361–365, December 2004.

[23] H. Zhong, W. Xu, N. Xie, and T. Zhang, “Area-efficient min-sum decoder design for high-rate quasi-cyclic low-density parity-check codes in magnetic recording,” *IEEE Transactions on Magnetics*, vol. 43, no. 12, pp. 4117–4122, 2007.

[24] G. Lechner, J. Sayir, and M. Rupp, "Efficient DSP implementation of an LDPC decoder," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. V-665–V-668, May 2004.

[25] T. Zhang and K. Parhi, "A 54 Mbps (3,6)-regular FPGA LDPC decoder," in *Proceedings of the IEEE Workshop on Signal Processing Systems (SIPS '02)*, pp. 127–132, October 2002.

[26] E. Boutillon, J. Castura, and F. Kschichang, "Decoder first code desig," in *Proceedings of the 2nd International Symposium on Turbo Codes and Related Topics*, pp. 459–462, September 2000.

[27] T. Brack, M. Alles, F. Kienle, and N. Wehn, "A synthesizable IP core for WiMax 802.16E LDPC code decoding," in *Proceedings of the 17th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '06)*, pp. 1–5, September 2006.

[28] T. Brack, M. Alles, T. Lehnigk-Emden et al., "Low complexity LDPC code decoders for next generation standards," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE '07)*, pp. 331–336, April 2007.

[29] M. Alles, N. Wehn, and F. Berens, "A synthesizable IP core for WiMedia 1.5 UWB LDPC code decoding," in *Proceedings of the IEEE International Conference on Ultra-Wideband (ICUWB '09)*, pp. 597–601, September 2009.

[30] B. Zhang, H. Liu, X. Chen, D. Liu, and X. Yi, "Low complexity DVB-S2 LDPC decoder," in *Proceedings of the 69th IEEE Vehicular Technology Conference (VTC '09)*, pp. 1–5, April 2009.

[31] J. Cho, N. R. Shanbhag, and W. Sung, "Low-power implementation of a high-throughput LDPC decoder for IEEE 802.11N standard," in *Proceedings of the IEEE Workshop on Signal Processing Systems (SiPS '09)*, pp. 40–45, October 2009.

[32] T. C. Kuo and A. N. Willson, "A flexible decoder IC for WiMAX QC-LDPC codes," in *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC '08)*, pp. 527–530, September 2008.

[33] S. Huang, D. Bao, B. Xiang, Y. Chen, and X. Zeng, "A flexible LDPC decoder architecture supporting two decoding algorithms," in *Proceedings of the IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems (ISCAS '10)*, pp. 3929–3932, June 2010.

[34] B. Xiang, D. Bao, S. Huang, and X. Zeng, "A fully-overlapped multi-mode QC-LDPC decoder architecture for mobile WiMAX applications," in *Proceedings of the 21st IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP '10)*, pp. 225–232, July 2010.

[35] Y. L. Wang, Y. L. Ueng, C. L. Peng, and C. J. Yang, "Processing-task arrangement for a low-complexity full-mode WiMAX LDPC codec," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 2, pp. 415–428, 2011.

[36] C. H. Liu, S. W. Yen, C. L. Chen et al., "An LDPC decoder chip based on self-routing network for IEEE 802.16e applications," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 3, pp. 684–694, 2008.

[37] X. Y. Shih, C. Z. Zhan, C. H. Lin, and A. Y. Wu, "An 8.29 mm2 52 mW multi-mode LDPC decoder design for mobile WiMAX system in 0.13 μm CMOS process," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 3, pp. 672–683, 2008.

[38] X. Y. Shih, C. Z. Zhan, and A. Y. Wu, "A real-time programmable LDPC decoder chip for arbitrary QC-LDPC parity check matrices," in *Proceedings of the IEEE Asian Solid-State Circuits Conference (A-SSCC '09)*, pp. 369–372, November 2009.

[39] P. Murugappa, R. Al-Khayat, A. Baghdadi, and M. Jezequel, "A flexible high throughput multi-ASIP architecture for LDPC and turbo decoding," in *Proceedings of the 14th Design, Automation and Test in Europe Conference and Exhibition (DATE '11)*, pp. 228–233, March 2011.

[40] M. Scarpellino, A. Singh, E. Boutillon, and G. Masera, "Reconfigurable architecture for LDPC and turbo decoding: a NoC case study," in *Proceedings of the 10th International Symposium on Spread Spectrum Techniques and Applications (ISSSTA '08)*, pp. 671–676, August 2008.

[41] M. Alles, T. Vogt, and N. Wehn, "FlexiChaP: a reconfigurable ASIP for convolutional, turbo, and LDPC code decoding," in *Proceedings of the 5th International Symposium on Turbo Codes and Related Topics (TURBOCODING '08)*, pp. 84–89, September 2008.

[42] F. Naessens, A. Bourdoux, and A. Dejonghe, "A flexible ASIP decoder for combined binary and non-binary LDPC codes," in *Proceedings of the 17th IEEE Symposium on Communications and Vehicular Technology in the Benelux (SCVT '2010)*, pp. 1–5, November 2010.

[43] F. Guilloud, E. Boutillon, and J. Danger, "λ-min decoding algorithm of regular and irregular LDPC codes," in *Proceedings of the 3rd International Symposium on Turbo Codes and Related Topics*, pp. 451–454, September 2003.

[44] Y. Dai, N. Chen, and Z. Yan, "Memory efficient decoder architectures for quasi-cyclic LDPC codes," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 9, pp. 2898–2911, 2008.

[45] M. Castano, M. Rovini, N.E. L'Insalata et al., "Adaptive single phase decoding of LDPC codes," in *Proceedings of the 6th International ITG-Conference on Source and Channel Coding (TURBOCODING), 4th International Symposium on Turbo Codes&Related Topics*, pp. 1–6, April 2006.

[46] I. C. Park and S. H. Kang, "Scheduling algorithm for partially parallel architecture of ldpc decoder by matrix permutation," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '05)*, vol. 6, pp. 5778–5781, May 2005.

[47] K. Gunnam, G. Choi, and M. Yeary, "A parallel VLSI architecture for layered decoding for array LDPC codes," in *Proceedings of the 6th International Conference on Embedded Systems, 20th International Conference on VLSI Design*, pp. 738–743, January 2007.

[48] High Rate UWB PHY and MAC Standard, Standard ECMA-368 Std, http://www.ecma-international.org.

[49] R. Hegde and N. R. Shanbhag, "A voltage overscaled low-power digital filter IC," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 2, pp. 388–391, 2004.

[50] B. Shim, S. R. Sridhara, and N. R. Shanbhag, "Reliable low-power digital signal processing via reduced precision redundancy," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 5, pp. 497–510, 2004.

[51] T. Vogt and N. Wehn, "A Reconfigurable application specific instruction set processor for convolutional and turbo decoding in a SDR environment," in *Proceedings of the Design, Automation and Test in Europe (DATE '08)*, pp. 38–43, March 2008.

[52] M. Flynn, "Very high-speed computing systems," *Proceedings of the IEEE*, vol. 54, no. 12, pp. 1901–1909, 1966.

[53] M. Awais, A. Singh, E. Boutillon, and G. Masera, "A novel architecture for scalable, high throughput, multi-standard LDPC decoder," in *Proceedings of the 14th Euromicro Conference on Digital System Design (DSD '11)*, vol. 31, pp. 340–347, September 2011.

[54] J. Sha, Z. Wang, M. Gao, and L. Li, "Multi-Gb/s LDPC code design and implementation," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 2, pp. 262–268, 2009.

[55] M. Rovini, G. Gentile, and L. Fanucci, "Multi-size circular shifting networks for decoders of structured LDPC codes," *Electronics Letters*, vol. 43, no. 17, pp. 938–940, 2007.

[56] F. Quaglio, F. Vacca, C. Castellano, A. Tarable, and G. Masera, "Interconnection framework for high-throughput, flexible LDPC decoders," in *Proceedings of the Design, Automation and Test in Europe (DATE '06)*, p. 6, March 2006.

[57] K. K. Gunnam, G. S. Choi, M. B. Yeary, and M. Atiquzzaman, "VLSI architectures for layered decoding for irregular LDPC codes of WiMax," in *Proceedings of the IEEE International Conference on Communications (ICC '07)*, pp. 4542–4547, June 2007.

[58] J. Dielissen, A. Hekstra, and V. Berg, "Low cost LDPC decoder for DVB-S2," in *Proceedings of the Design, Automation and Test in Europe (DATE '06)*, vol. 2, pp. 1–6, March 2006.

[59] M. Karkooti, P. Radosavljevic, and J. R. Cavallaro, "Configurable, high throughput, irregular LDPC decoder architecture: tradeoff analysis and implementation," in *Proceedings of the 17th IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP '06)*, pp. 360–367, September 2006.

[60] C. Zhang, Z. Wang, J. Sha, L. Li, and J. Lin, "Flexible LDPC decoder design for multigigabit-per-second applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 1, pp. 116–124, 2010.

[61] G. Gentile, M. Rovini, and L. Fanucci, "A multi-standard flexible Turbo/LDPC decoder via ASIC design," in *Proceedings of the 6th International Symposium on Turbo Codes and Iterative Information Processing (ISTC '10)*, pp. 294–298, September 2010.

[62] A. Tarable, S. Benedetto, and G. Montorsi, "Mapping interleaving laws to parallel turbo and LDPC decoder architectures," *IEEE Transactions on Information Theory*, vol. 50, no. 9, pp. 2002–2009, 2004.

[63] D. Oh and K. K. Parhi, "Low-complexity switch network for reconfigurable LDPC decoders," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 1, pp. 85–94, 2010.

[64] J. Tang, T. Bhatt, V. Sundaramurthy, and K. K. Parhi, "Reconfigurable shuffle network design in LDPC decoders," in *Proceedings of the 17th IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP '06)*, pp. 81–86, September 2006.

[65] J. Lin, Z. Wang, L. Li, J. Sha, and M. Gao, "Efficient shuffle network architecture and application for WiMAX LDPC decoders," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 56, no. 3, pp. 215–219, 2009.

[66] C. H. Liu, C. C. Lin, S. W. Yen et al., "Design of a multimode QC-LDPC decoder based on shift-routing network," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 56, no. 9, pp. 734–738, 2009.

[67] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan Kaufmann Publishers, 1992.

[68] X. Peng, Z. Chen, X. Zhao, F. Maehara, and S. Goto, "High parallel variation banyan network based permutation network for reconfigurable LDPC decoder," in *Proceedings of the 21st IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP '10)*, pp. 233–238, July 2010.

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

[70] H. Moussa, A. Baghdadi, and M. Jézéquel, "Binary de Bruijn on-chip network for a flexible multiprocessor LDPC decoder," in *Proceedings of the 45th Design Automation Conference (DAC '08)*, pp. 429–434, June 2008.

[71] N. De Bruijn, "A combinatorial problem," *Koninklijke Nederlandse Akademie*, vol. 49, pp. 758–764, 1946.

[72] T. Theocharides, G. Link, N. Vijaykrishnan, and M. J. Irwin, "Implementing LDPC decoding on network-on-chip," in *Proceedings of the 18th International Conference on VLSI Design: Power Aware Design of VLSI Systems*, pp. 134–137, January 2005.

[73] G. Masera, F. Quaglio, and F. Vacca, "Implementation of a flexible LDPC decoder," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54, no. 6, pp. 542–546, 2007.

[74] F. Kienle, M. Thul, and N. When, "Implementation issues of scalable LDPC-decoders," in *Proceedings of the 3rd International Symposium on Turbo Codes and Related Topics*, 2003.

[75] D. Seo, A. Ali, W. T. Lim, N. Rafique, and M. Thottethodi, "Near-optimal worst-case throughput routing for two-dimensional mesh networks," in *Proceedings of the 32nd Interntional Symposium on Computer Architecture (ISCA '05)*, pp. 432–443, June 2005.

[76] F. Vacca, G. Masera, H. Moussa, A. Baghdadi, and M. Jezequel, "Flexible architectures for LDPC decoders based on network on chip paradigm," in *Proceedings of the 12th Euromicro Conference on Digital System Design: Architectures, Methods and Tools (DSD '09)*, pp. 582–589, August 2009.

[77] D. E. Hocevar, "A reduced complexity decoder architecture via layered decoding of LDPC codes," in *Proceedings of the IEEE Workshop on Signal Processing Systems Design and Implementation*, pp. 107–112, October 2004.

[78] C. Condo, *A parallel LDPC decoder with Network on Chip as underlying architecture*, M.S. thesis, Politecnico di Torino, 2010.

[79] C. Condo and G. Masera, "A flexible NoC-based LDPC code decoder implementation and bandwidth reduction methods," in *Proceedings of the Conference on Design and Architectures for Signal and Image Processing (DASIP '11)*, pp. 1–8, November 2011.

[80] Z. Chen, X. Zhao, X. Peng, D. Zhou, and S. Goto, "An early stopping criterion for decoding LDPC codes in WiMAX and WiFi standards," in *Proceedings of the IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems (ISCAS '10)*, pp. 473–476, June 2010.

[81] 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: Regular Papers*, vol. 57, no. 10, pp. 2776–2789, 2010.

[82] C. Condo, M. Martina, and G. Masera, "A Network-on-Chip-based high throughput turbo/LDPC decoder architecture," in *Proceedings of the Conference on Design, Automation and Test in Europe (DATE '12)*, 2012.

[83] M. Imase and M. Itoh, "A design for directed graphs with minimum diameter," *IEEE Transactions on Computers*, vol. 32, no. 8, pp. 782–784, 1983.

[84] F. Naessens, B. Bougard, S. Bressinck et al., "A unified instruction set programmable architecture for multi-standard advanced forward error correction," in *Proceedings of the IEEE Workshop on Signal Processing Systems (SiPS '08)*, pp. 31–36, October 2008.