Low Complexity LDPC Code Decoders for Next Generation Standards

T. Brack, M. Alles, T. Lehnigk-Emden,
F. Kienle, N. Wehn
Microelectronic System Design Research Group
University of Kaiserslautern
67663 Kaiserslautern, Germany

N.E. L'Insalata, F. Rossi, M. Rovini,
L. Fanucci
Department of Information Engineering
University of Pisa
via G. Caruso, 56122, Pisa, Italy

Abstract

This paper presents the design of low complexity LDPC codes decoders for the upcoming WiFi (IEEE 802.11n), WiMax (IEEE802.16e) and DVB-S2 standards. A complete exploration of the design space spanning from the decoding schedules, the node processing approximations up to the top-level decoder architecture is detailed. According to this search state-of-the-art techniques for a low complexity design have been adopted in order to meet feasible high throughput decoder implementations. An analysis of the standardized codes from the decoder-aware point of view is also given, presenting, for each one, the implementation challenges (multi rates-length codes) and bottlenecks related to the complete coverage of the standards. Synthesis results on a present 65nm CMOS technology are provided on a generic decoder architecture.

1 Introduction

The increasing demand of high data rate and reliability in modern communication systems is pushing next-generation standards toward error correction schemes allowing high throughput decoding with near Shannon limit performance.

At present, Low-Density Parity-Check (LDPC) codes [9] are among the best candidates to meet these requirements. However, first works on hardware implementations [1] pointed out the huge complexity associated to a LDPC decoder even for short length codewords. The peculiarities of the decoding algorithm (iterative processing, transcendental operators, pseudo-random message exchange) strongly affect traditional VLSI systems metrics (area, speed, power) making it difficult to meet feasible implementation requirements without spoiling communication performance [4].

Resorting to joint code-decoder design techniques [17] has become a common practice to ease the implementation of high data rate decoder architecture. Indeed all upcoming standards featuring the use of LDPC codes as WiFi (IEEE 802.11n) [12], WiMax (IEEE802.16e) [13] and DVB-S2 [7], adopt architecture-aware LDPC codes. Despite the co-design approach, the need for a further reduction in complexity is still an appealing issue, specially when coping with the variety of code lengths and rates exhibited by the above mentioned standards.

After reviewing the state-of-the-art for low complexity design (alternative schedules, node processing approximation), the paper focuses on how these techniques are used on actual decoder implementations targeting next-generation standards. In particular, an analysis on the LDPC codes for WiFi, WiMax and DVB-S2 is carried on highlighting the issues and the complexity overhead related to the complete coverage of these standards. Moreover, in line with the decoder-aware techniques used for these standards, the introduction of an optional LDPC codes for an OFDM-UWB system featuring a very low complexity decoder implementation is discussed.

2 Fundamentals of LDPC Decoding

LDPC codes are linear block codes characterized by a sparse binary matrix $\mathbf{H}$, called parity-check matrix. The set of valid codewords C is defined as:

$$H \bullet x^T = 0, \qquad \forall x \in C \tag{1}$$

The code can also be described by means of a bipartite graph, known as Tanner graph (see Figure 1). A Tanner graph is made up of two entities, variable nodes (VN) and check nodes (CN), connected each other through a set of edges. An edge links the check node i to the variable node j if the element $H_{i,j}$ of the parity check matrix is non-null.

The optimal LDPC decoding is achieved with a two-phase message passing algorithm (also known as *flooding*), that can be described as an iterative exchange of probabilistic messages along the edges of the Tanner graph [16]. The algorithm proceeds iteratively until a maximum number of iterations or a stopping rule is met. Inputs of the algorithm are the intrinsic *Log-Likelihood Ratios* (LLRs), also referred to as *a priori* information. Intrinsic LLRs measure

978-3-9810801-2-4/DATE07 © 2007 EDAA

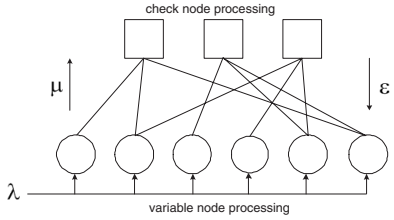

Figure 1. Tanner Graph of an LDPC Code

the reliability of the received data only based on channel observation. At the q-th iteration, the VN n receives the messages $\epsilon_{m,n}^{(q)}$ from the neighboring CNs and propagates back the updated messages $\mu_{m,n}^{(q)}$ computed as:

$$\mu_{m,n}^{(q)} = \lambda_n + \sum_{i \in \{\mathcal{M}_n \backslash m\}} \epsilon_{i,n}^{(q)} \tag{2}$$

where λ_n is the intrinsic LLR on the current bit, and $\mathcal{M}_n$ is the set of CNs connected to VN n. At the same time, a refined estimation on the transmitted bit is produced, also referred to as soft output (SO):

$$y_n^{(q)} = \lambda_n + \sum_{i \in \mathcal{M}_n} \epsilon_{i,n}^{(q)} \tag{3}$$

Then, in the next semi-iteration, the generic CN m combines together messages $\mu_{m,n}^{(q)}$ from the neighboring VNs to compute the updated messages $\epsilon_{m,n}^{(q+1)}$, which are sent back to the respective VN. Update is performed separately on signs and magnitudes:

$$-\operatorname{sgn}\left(\epsilon_{m,n}^{(q+1)}\right) = \prod_{j \in \{\mathcal{N}_m \backslash n\}} -\operatorname{sgn}\left(\mu_{m,j}^{(q)}\right) \tag{4}$$

$$\left|\epsilon_{m,n}^{(q+1)}\right| = \Phi^{-1}\left\{\sum_{j \in \{\mathcal{N}_m \backslash n\}} \Phi\left(\left|\mu_{m,j}^{(q)}\right|\right)\right\} \tag{5}$$

with $\mathcal{N}_m$ the set of VNs connected to CN m, and

$$\Phi(x) = \Phi^{-1}(x) = -\log\left(\tanh\left(\frac{x}{2}\right)\right) \tag{6}$$

Considerations on how to implement the transcendental operator in (6) with low-complexity hardware can be found in Section 3.2.

3 Low-Complexity Decoder Design

Meeting throughput requirements of forthcoming standards ask for a massive parallelization of the decoding process leading to noticeable chip area. On the contrary, architectures with a low degree of parallelism needs high clock frequencies, thus affecting the power consumption that is critical for mobile devices.

The following section reviews different state-of-the-art techniques for reducing the complexity of the decoder in terms of VLSI metrics with the attempt of minimizing the BER implementation loss (IL).

3.1 Modified Decoding Schedules

Modified schedules get rid of the classical two-phase *flooding* decoding (see Section 2) as to improve the convergence speed, so that the same communication performance can be achieved with a reduction in the decoding time. This is advantageous especially for those standard proposals (e.g. WiFi IEEE 802.11n) where the latency budget for channel decoding is poor (6 μs or less). In layered decoding [11, 17] the parity check matrix is considered as being made of a sequence of horizontal or vertical *layers*, hence the names of *horizontal* and *vertical shuffle* also used to indicate this technique. The layered decoding principle for horizontal layers is expressed by:

$$-\operatorname{sgn}\left(\epsilon_{m,n}^{(q+1)}\right) = \prod_{j \in \{\mathcal{N}_m \backslash n\}} -\operatorname{sgn}\left(y_n^{(q)}[k] - \epsilon_{m,j}^{(q)}\right) \tag{7}$$

$$\left|\epsilon_{m,n}^{(q+1)}\right| = \Phi^{-1}\left\{\sum_{j \in \{\mathcal{N}_m \backslash n\}} \Phi\left(\left|y_n^{(q)}[k] - \epsilon_{m,j}^{(q)}\right|\right)\right\} \tag{8}$$

Equations (7) and (8) were derived by merging the VN and the SO update rules (2)–(3) with the CN update rules (4)–(5). This way the VN phase is spread on the CN updates and the estimate of SO $y_n^{(q)}$ is refreshed after any CN update, as expressed by:

$$y_n^{(q)}[k] = y_n^{(q)}[k-1] - \epsilon_{m,n}^{(q-1)} + \epsilon_{m,n}^{(q)} \tag{9}$$

where k is the time step within an iteration. The key point of the layered schedule is the intermediate update, within an iteration, of the SO estimates and their propagation to the next layers, boosting the convergence speed and saving up to 50% of the iteration time [11].

Note that the layered schedule is compatible with a parallelization of the architecture (i.e. a layer spanning multiple rows), but collision-free decoding can be achieved only for those codes where the rows of the parity matrix are grouped into subsets in which the column weight is at most one [11]. However, it is demonstrated in [19] that approximated layered decoding is very robust even in the support of highly non-layered codes.

Another alternative to the *flooding* schedule is the *Adaptive Single Phase Decoding* (ASPD) [5]. Basically, it adaptively updates a single metric for each VN in the codeword. This metric is continuously cumulated in a running sum with leakage and fed to the CN processor combined with the *intrinsic* LLRs. The leakage coefficients for the running sum are tuned onto each bit in the codeword and dynamically adapted through the decoding process.

The main rationale behind the ASPD is reducing the total memory of the decoder. Memory requirements are decoupled from the number of edges in the Tanner graph and are related only to the length of the codeword. For a reduction

in memory size ranging from 60 to 80%, the achieved BER performance in *fixed point* arithmetic exhibits an IL from 0.2 to 0.3 dB for mid-length codewords (about 4Kbits) [5].

3.2 Node Processors Approximations

In a low complexity scenario the choice for serial architectures of the elementary processing units (CN & VN processors) is almost compulsory. In fact, serial processors offer an inherent low complexity and high flexibility by easily managing different node degrees with very small resources overheads. This last feature is particularly useful for those codes where the node degree is spread (see Section 4.1).

Note that a serial architecture naturally fits most of the approximations proposed for the CN elaborations, where the gate complexity of the decoder is generally concentrated. Indeed, for low-complexity solutions, the direct implementation of the hyperbolic function of (6) is usually avoided by resorting to successive applications of the binary operator:

$$g(a,b) \doteq \min(a,b) - \log\left(\frac{1+\mathrm{e}^{-|a-b|}}{1+\mathrm{e}^{-|a+b|}}\right) \qquad (10)$$

In [22] the proposed M-min^* (11) slightly modifies the original version in order to reduce the complexity, and, above all, to increase the numerical stability in *fixed point* decoding [18].

$$\text{M-min}^*(a,b) \doteq \min(a,b) + \log\left(\frac{\mathrm{e}^{|a-b|}}{1+\mathrm{e}^{|a-b|}}\right) \qquad (11)$$

Min-Sum decoding [8] performs the roughest approximation considering in (10) only the minimum, but the correction with a constant term has also been proposed.

After selecting one implementation for the binary operator of the CNP, a possible architecture supporting serial data flow and performing exact marginalization is based on a double recursion scheme, as proposed in [18]. This architecture only needs three operators, but data flow control for the forward and backward metrics is quite complex and area consuming. A significant save can be obtained if a dedicated value is computed only for a subset of the incoming messages while a common value is assigned to the remaining ones. In [10] only the set $\mathcal{N}_\lambda(i)$ of the λ incoming messages with the lowest reliability is considered for elaboration. In particular the output magnitude of the messages in $\mathcal{N}_\lambda(i)$ is evaluated as usual, while for the others a common value is elaborated by applying the binary operator to the whole set of $\mathcal{N}_\lambda(i)$.

A similar strategy is followed in [20] where proper marginalization is performed just for the P messages with the lowest reliability but, in this case, the whole set of the input messages is considered for elaboration. In [4] a low-complexity CNP has been designed for the DVB-S2 implementing the algorithm described in [10] with $\lambda = 3$, while in [20] communication performance and implementation results are presented for a processor architecture featuring just $\lambda = 1$. Note that the use of partial marginalization strategies in conjunction with the adoption of a layered schedule can bring a reduction of the message memory since fewer magnitude values need to be stored[20].

4 Decoder-Aware LDPC Codes

All current standardized LDPC codes incorporate a common set of features which made feasible implementation possible in the first place: Accumulator-based matrices to allow for linear encoding complexity, structured matrices to be mapped on partly parallel architectures, and permuted identity sub matrices to ease network implementation. These aspects can be summarized under the term of decoder-aware code design. In this section we present three standards employing decoder-aware LDPC codes and show corresponding synthesis results on a 65 nm technology.

4.1 Standardized LDPC Codes

The DVB-S2 satellite video broadcasting standard [7] was designed for an exceptional error performance at very low SNR ranges (up to FER $\leq 10^{-7}$ at -2.35dB E_S/N_0). Thus the specified LDPC codes use a large block length of 64800 bit with 11 different code rates ranging from $1/4$ to $9/10$. This results in **large storage requirements for up to 285000 messages and demands high code rate flexibility** at the same time to support all specified node degrees, as shown in the first row of Table 1.

The current WiMax 802.16e [13] standard features LDPC codes as an optional channel coding scheme. It consists of six different code classes with different VN and CN distributions, spanning four different code rates from $1/2$ to $5/6$ (see second row of Table 1). All six code classes have the same general parity check matrix structure and support 19 codeword sizes, ranging from 576 to 2304 with a granularity of only 96 bit. This **codeword size flexibility** is the most challenging aspect of this standardized LDPC code family. The interest of the standardization committee in the improvement in throughput and communications performance achievable through layered decoding (see Section 3.1) is evident since an optimal sequence of layers [19] is specified for two of the code classes [2].

The upcoming WiFi 802.11n [12] standard will also feature LDPC codes as an optional channel coding scheme. It utilizes 12 different codes utilizing four code rates from $1/2$ to $5/6$ for each of the three different codeword sizes of 648, 1248, and 1944 bit. The most complicated issue with this code is the **CN and VN flexibility** needed to fully support this standard (see third row of Table 1).

	Codeword Size			Code Rate			CN Degree			VN Degree			# Edges
	#	min	max	#	min	max	#	min	max	#	min	max	
DVB-S2	1	64800	64800	11	1/4	9/10	11	4	30	7	2	13	285120
WiMax 802.16e	19	576	2304	4	1/2	5/6	7	6	20	4	2	6	8448
WiFi 802.11n	3	648	1944	4	1/2	5/6	9	7	22	8	2	12	7128
Ultra-Sparse	1	9600	9600	1	3/4	3/4	1	11	11	2	2	3	26400

Table 1. Parameters of selected Decoder-Aware LDPC Codes

4.2 Application Specific LDPC Codes

For future applications, LDPC decoders with very high throughput and excellent communications performance are required. At the same time, chip area should be small to provide cost-efficient solutions. To get these results, all well-known techniques used by the standardized codes presented in Section 4.1 have to be exploited. This puts constraints on the code design, but does not provide the actual code itself and still offers a large degree of freedom.

Application specific code design became mandatory in the development of an optional LDPC channel decoder to be used in a sophisticated OFDM-UWB system [21]. Such systems have to provide very high net throughput for upcoming services, therefor LDPC channel coding is suggested to become an addition to the current UWB standard. The code rate of 3/4 and codeword size of 9600 bit were given by the system MAC layer. Silicon area has to be as small as possible to enable low-cost consumer products. For the communications performance, the convolutional decoder used in the current UWB standard had to be outperformed at a packet error rate of 10^{-3} for the IEEE 802.15.3a CM1 and CM2 channel models.

For this purpose, a so-called Ultra-Sparse LDPC code [3] was designed by using the 2V-PEG approach presented in [15]. In the last row of Table 1 the exceptional small maximum VN and CN degree for a rate 3/4 is demonstrated. With only 26400 edges to process, this LDPC code is especially suited for high throughput decoding by using only a small number of parallel functional units. Typically these codeword size corresponds to more than 35000 edges, utilizing 30% more area to achieve the same throughput. It also allows for layered decoding which is very difficult to achieve for rate 3/4 codes.

4.3 LDPC Decoder Architectures

Two different LDPC decoder architectures are considered: A two-phase decoder implementing the flooding schedule, and one for layered decoding. The structure of both architectures is partly parallel, thus only a subset of nodes in the Tanner graph is instantiated as variable node and check node processors. The node processors work in a serial manner what gives the needed flexibility regarding the variable node and check node degrees. However, this serial architecture prevents the standardized codes from being decoded with layered scheduling due to the latency of d_c clock cycles introduced by the CNP. Because of the maximal variable node degree of these codes it can not be guaranteed that the updated message is computed before it is being needed for another check node. For the proposed Ultra-Sparse LDPC code with the low d_v^{max} of three this constraint for layered decoding is easily satisfied. In both architectures the permutation networks are realized by logarithmic barrel shifters to process all LDPC codes specified by permuted identity matrices.

There are some fundamental differences between the two-phase decoder (Figure 2) and the layered decoder architecture (Figure 3). The two-phase decoder contains two sum RAMs that are used to accumulate all incoming messages of a variable node. During one iteration one sum RAM is used to compute $\mu_{m,n}^{(q)}$ by subtracting the corresponding message from the message RAM and adding the channel value of the channel RAM. The second sum RAM is needed to build new sums for the next iteration, hence both RAMs are swapped after each iteration. In contrast the layered decoder stores the *a posteriori* information in the channel RAM. This RAM contains the sum of channel value and all incoming messages of a variable node ($y_n^{(q)}$), thus only the corresponding message has to be subtracted in the check node block (CNB) to obtain $\mu_{m,n}^{(q)}$. When the CPN has computed new messages these are stored in the message RAM and added to the $\mu_{m,n}^{(q)}$ bypassed by the FIFO. Using *a posteriori* values also allows for using only one permutation network for the layered architecture. More details regarding both architectures can be found in [4] and [3].

4.4 Implementation Results

The standardized LDPC codes have been mapped on the two-phase decoder architecture template (Figure 2) and synthesized using the current 65nm technology from STMicroelectronics. The target frequency was set to 400 Mhz which allows for using relatively small memories with cycles times up to 2.5ns for reduced area utilization. All decoder implementations utilize 3-Min check nodes to support adequate communications performance even for the low code rates 1/4-1/2. Table 2 shows the final results for each code, structured by the functional decoder elements shown in Figure 2. It also shows the net and air throughput range, starting from the smallest size with lowest code rate combination up to the largest high-rate one. Therefor the maximum number of iterations is set to allow sufficient

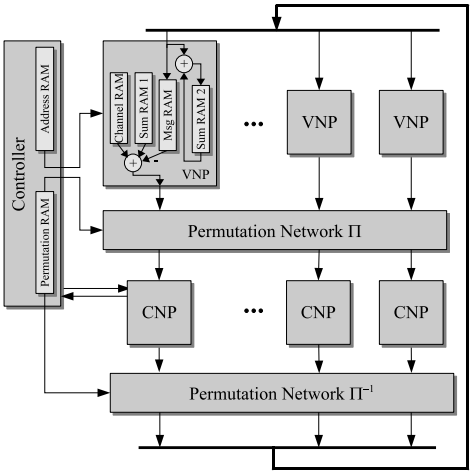

Figure 2. Two-Phase Decoder Architecture

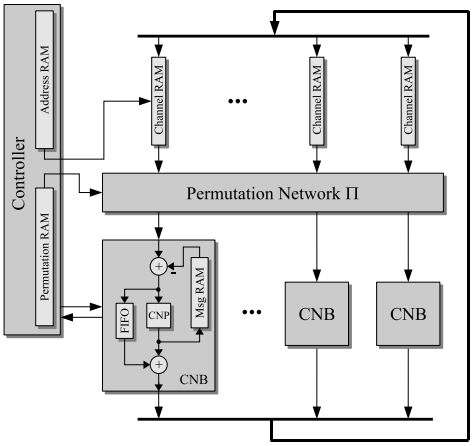

Figure 3. Layered Decoder Architecture

communications performance for each code rate, from 50 for rate $1/4$ to only 15 for rate $9/10$.

The first column shows the DVB-S2 LDPC decoder implementation with an overall area of $3.8mm^2$, where **85% of the area is dedicated to memory**. Instead of utilizing the maximal possible parallelization degree supported by the code structure as already presented in [14] and [4], the method proposed in [6] has been used to scale down the throughput for set-top box applications. Net throughput ranges from 60 for rate $1/4$ to more than 700 Mbps for rate $9/10$, easily reaching 90 Mbps for rate $3/5$ as required by broadcast service specifications. The implementation results for the WiMax LDPC decoder are shown in the second column. The size of the **network corresponds to 25% of the logic area compared to only 10% for the other standard implementations**, reflecting the enormous codeword size flexibility to be supported. More detailed explanation about the network structure is given in [2]. The achieved air throughput is above 70 Mbps as required by the current WiMax specification. The last column presents the results

LDPC Code	**DVB-S2**	**802.16e**	**802.11n**
Codeword Size	**64800 bit**	**576-2304 bit**	**648-1944 bit**
Code Rate	**1/4 - 9/10**	**1/2 - 5/6**	**1/2 - 5/6**
Parallelism	90	24-96	27-81
Quantization	6 bit		
Algorithm	3-Min		
Max. Iterations	50 - 15	25 - 20	25 - 20
Area[mm^2] $65nm$@400 MHz			
VNP	0.130	0.110	0.096
CNP	0.328	0.470	0.395
Network	0.046	0.206	0.065
Memory	3.357	0.551	0.467
Overall Area	3.861	1.337	1.023
Net Throughput	60 - 708 Mbps	48 - 333 Mbps	54 - 281 Mbps
Air Throughput	240 - 786 Mbps	96 - 399 Mbps	108 - 337 Mbps
Latency	$270 - 82\mu s$	$6.0 - 5.7\mu s$	$6.0 - 5.8\mu s$

Table 2. Synthesis Results for standardized LDPC Codes

for the WiFi LDPC decoder. Here the required node flexibility is mainly visible in the check node size, where the **check node utilizes 33% more area compared to one CN used by the DVB-S2 decoder**. A worst case latency of $6\mu s$ is guaranteed for all codes specified.

Table 3 shows synthesis results for the application specific Ultra-Sparse LDPC code, utilizing the previously introduced layered decoder architecture (Figure 3). By using the much simpler Min-Sum check nodes and fast memories, a clock frequency of 500 MHz was achieved for further throughput enhancement. Although the overall area after synthesis is only around **0.5 mm**^{2}, the net throughput easily exceeds **1 Gbps**.

5 Conclusions

Proposals for next-generation standards selected LDPC codes for forward error correction, either mandatory (DVB-S2) or optional for high-throughput modes (IEEE 802.11n and IEEE 802.16e). Despite the use of joint code/decoder design techniques, implementing a fully compliant LDPC decoder is still a challenging task. The lack of homogeneity in the standardized matrices usually leads to an over dimensioned and/or partially compliant decoder.

This paper compared the implementation of decoders for different standardized codes using a generic reference architecture. Even if state-of-the-art low-complexity techniques have been used along with a present CMOS 65nm technology, area figures may still be too high for integrating the decoder in a full modem, especially for low-cost consumer products. It has been also demonstrated that further reduction on chip area can be achieved if the code is designed enforcing the awareness of the decoder implementation bottlenecks. Indeed future standard like UWB propos-

LDPC Code	$f_{[3,2]} = \{3/4, 1/4\}$ $g_{[11]} = \{1\}$
Codeword Size	**9600 bit**
Code Rate	**3/4**
Parallelism	80
Quantization	6 bit
Algorithm	MinSum+MSF, Layered Decoding
Iterations	≤ 10
Area[mm^2] $65nm$@500 MHz	
CNB/CNP	0.212
Network	0.027
Memory	0.265
Overall Area	0.504
Net Throughput	**1.08 Gbps**
Air Throughput	**1.45 Gbps**
Latency	$6.6\mu s$

Table 3. Synthesis Results for a proposed UWB LDPC Code

als have to take into account other issues such as the impact of node distribution or the decoding schedule, in addition to the well-known actual block-partitioned matrices.

6 Acknowledgments

Our special thanks goes to Friedbert Berens from the Computer Peripheral Group of STMicroelectronics, Geneva, Switzerland, for many valuable discussions.

References

[1] A. Blanksby and C. Howland. A 690-mW 1-Gb/s 1024-b, rate-1/2 low-density parity-check code decoder. *IEEE J. Solid-State Circuits*, 37(3):404–412, Mar 2002.

[2] T. Brack, M. Alles, F. Kienle, and N. Wehn. A synthesizable IP Core for WiMax 802.16e LDPC Code Decoding. In *Proc. 2006 Personal Indoor and Radio Communications Conference (PIMRC '06)*, Helsinki, Finland, Sept. 2006.

[3] T. Brack, F. Kienle, T. Lehnigk-Emden, M. Alles, N. Wehn, and F. Berens. Enhanced Channel Coding for OFDM-based UWB Systems. In *Proc. International Conference on Ultra-Wideband (ICUWB 2006)*, Waltham, Massachusetts, Sept. 2006.

[4] T. Brack, F. Kienle, and N. Wehn. Disclosing the LDPC Code Decoder Design Space. In *Proc. 2006 Design, Automation and Test in Europe (DATE '06)*, Munich, Germany, Mar. 2006.

[5] M. Castano, M. Rovini, N. L'Insalata, F. Rossi, R. Merlino, C. Ciofi, and L. Fanucci. Adaptive Single Phase Decoding of LDPC Codes. In *Proc. 4th International Symposium on Turbo Codes and Related Topics*, Apr 2006.

[6] J. Dielissen, A. Hekstra, and V. Berg. Low cost LDPC decoder for DVB-S2. In *Proc. 2006 Design, Automation and Test in Europe (DATE '06)*, Munich, Germany, Mar. 2006.

[7] European Telecommunications Standards Institude (ETSI). Digital Video Broadcasting (DVB) Second generation framing structure for broadband satellite applications; EN 302 307 V1.1.1. `www.dvb.org`.

[8] M. Fossorier, M. Mihaljevic, and H. Imai. Reduced complexity iterative decoding of low-density parity check codes based on belief propagation. *IEEE Trans. Commun.*, 47(5):673–680, May 1999.

[9] R. Gallager. *Low-Density Parity-Check Codes*. PhD thesis, Massachusetts Institutes of Technology, 1960.

[10] F. Guilloud, E. Boutillon, and J. Danger. λ-Min Decoding Algorithm of Regular and Irregular LDPC Codes. In *Proc. 3nd International Symposium on Turbo Codes & Related Topics*, pages 451–454, Brest, France, Sept. 2003.

[11] D. Hocevar. A Reduced Complexity Decoder Architecture via Layered Decoding of LDPC Codes. In *IEEE Workshop on Signal Processing Systems, SISP 2004*, pages 107–112, 2004.

[12] IEEE 802.11n. Wireless LAN Medium Access Control and Physical Layer specifications: Enhancements for Higher Throughput. IEEE P802.16n/D1.0, Mar 2006.

[13] IEEE 802.16e. Air Interface for Fixed and Mobile Broadband Wireless Access Systems. IEEE P802.16e/D12 Draft, oct 2005.

[14] F. Kienle, T. Brack, and N. Wehn. A Synthesizable IP Core for DVB-S2 LDPC Code Decoding. In *Proc. 2005 Design, Automation and Test in Europe (DATE '05)*, Munich, Germany, Mar. 2005.

[15] F. Kienle, T. Brack, and N. Wehn. Design of Irregular LDPC Codes for Flexible Encoder and Decoder Hardware Realizations. In *Proc. International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2006)*, Dubrovnik, Croatia, Oct. 2006.

[16] D. MacKay. Good error-correcting codes based on very sparse matrices. *IEEE Trans. Inform. Theory*, 45(2):399–431, Mar 1999.

[17] M. Mansour and N. R. Shanbhag. High-throughput LDPC decoders. *IEEE Trans. VLSI Syst.*, 11(6):976–996, Dec 2003.

[18] M. Rovini, N. L'Insalata, F. Rossi, and L. Fanucci. VLSI Design of a High-Throughput Multi-Rate Decoder for Structured LDPC Codes. In *Proc. 8th Euromicro Conference on Digital System Design (DSD)*, pages 202–209, Aug-Sept. 2005.

[19] M. Rovini, F. Rossi, P. Ciao, N. L'Insalata, and L. Fanucci. Layered Decoding of Non-Layered LDPC Codes. In *Proc. 9th Euromicro Conference on Digital System Design (DSD)*, Aug-Sept. 2006.

[20] M. Rovini, F. Rossi, N. L'Insalata, and L. Fanucci. High-Precision LDPC Codes Decoding at the Lowest Complexity. In *Proc. 14th European Signal Processing Conference (EUSIPCO)*, Sept. 2006.

[21] Standard ECMA-368. High Rate Ultra Wideband PHY and MAC Standard.

[22] F. Zarkeshvari and A. Banihashemi. On implementation of min-sum algorithm for decoding low-density parity-check (LDPC) codes. In *Proc. IEEE GLOBECOM*, volume 2, pages 1349–1353, Nov 2002.