ICT-MobileSummit 2008 Conference Proceedings
Paul Cunningham and Miriam Cunningham (Eds)
IIMC International Information Management Corporation, 2008
ISBN: 978-1-905824-08-3

Enhanced Iteration Control for Ultra Low Power LDPC Decoding

Timo LEHNIGK-EMDEN[1], Norbert WHEN[1], Friedbert BERENS[2]
[1]*Microelectronic System Design Research Group, University of Kaiserslautern, Erwin-Schroedinger-Str., 67663 Kaiserslautern, Germany*
Email: {lehnigk, wehn}@eit.uni-kl.de
[2]*STMicroelectronics N.V., 1228 Plan-les-Ouates, Geneva, Switzerland*
Email: friedbert.berens@st.com

Abstract: Today's and tomorrow's wireless systems and applications require high throughput and low latency. Additionally low energy consumption and efficient air bandwidth usage are very important to provide mobile services anytime and anywhere. Channel coding is one of the key techniques to fulfill these requirements. LDPC codes are used in current and upcoming standards due to their outstanding communications performance. Decoding an LDPC code is an iterative process. The energy consumption and latency are strongly related to the number of iterations. Thus an iteration control is an efficient way to save energy. State-of-the-art iteration control implementations need at least one iteration to detect an error free received frame. In this paper we present a new technique which detects an error free received frame before the decoding process starts. We show that especially under good transmission conditions the saving of iterations and energy is remarkable. The WIMEDIA UWB scenario is used as simulation environment. Complexity estimations and implementation results are presented using a 65nm low power technology.

Keywords: LDPC, energy, low power, iteration control, implementation, WIMEDIA, UWB

1. Introduction

Upcoming short range wireless standards based on Ultra Wide Band (UWB) are reaching data rates up to 5 Gbit/s in the near future. The WIMEDIA [16] standard Release 1.5 should handle up to 1 Gbit/s and the Release 2.0 targets data rates of up to 3 Gbit/s. In order to guarantee an acceptable link quality and a high throughput advanced channel coding schemes become mandatory. Thus the traditional convolutional code (CC) has to be replaced by an advanced coding scheme. LDPC codes are a very good candidate for this replacement, since they have an outstanding communication performance and they are already standardized in other wireless standards like IEEE WIMAX(802.16e) [7] or IEEE WiFi(802.11n) [6]. The main drawback of these codes is the complexity, the latency and the related power consumption of the decoders. Recently a new code class of Ultra-Sparse (U-S)-LDPC based on the know LDPC code classes used in WIMAX and WIFI have been proposed in order to minimize the complexity and the energy consumption [10]. These codes are specially designed for the use in the WIMEDIA standard [2]. The complexity w.r.t. silicon area is in the range of the already used traditional CC decoders giving a communications performance boost of up to 4.5 dB [2][9][12][17]. Due to the iterative decoding process of LDPC decoding the power consumption of the decoder is higher in general than the power consumption of the Viterbi decoder used for the decoding of a CC code. This is true only if no iteration control is being used. LDPC decoders with iteration control schemes [10] show a lower power consumption than a Viterbi decoder. The known iteration control schemes mainly target the low and medium SNR region of the wireless transmission system.

In this paper we propose a new technique to complement these existing schemes in the high SNR region in order to take advantage of the increased number of error free codewords received under these operational conditions. The additional power savings in the high SNR region can each up to 60% as compared to the know schemes. Taking into account the implementation of a complete iteration control framework into the U-S-LDPC it can be shown that the average power consumption of the proposed LDPC decoder can be 50% below the consumption of a Viterbi decoder on a FPGA. The remaining of the paper is organized as follows. Section 2 gives a short introduction to LDPC codes and decoding. Section 3 addresses LDPC decoder hardware architecture. In Section 4 state-of-the-art iteration control schemes are presented and the new enhanced iteration control schemes for LDPC codes. Simulations, complexity estimations and implementation results are given in Section 5 . Section 6 concludes the paper.

2. LDPC Codes

LDPC codes are linear block codes defined by a sparse binary matrix H, called the parity check matrix. The set of valid codewords C satisfies

$$Hx^T = 0, \qquad \forall x \in C. \quad (1)$$

A column in H is associated to a codeword bit, and each row corresponds to a parity check. A nonzero element in a row means that the corresponding bit contributes to this parity check. The complete code can best be described by a Tanner graph [13], a graphical representation of the associations between code bits and parity checks. Code bits are shown as so called variable nodes (VN) drawn as circles, parity checks as check nodes (CN) represented by squares, with edges connecting them accordingly to the parity check matrix. Figure 1 shows a Tanner graph for a generic LDPC code with N variable and M check nodes with a resulting code rate of $R = (N - M)/N$.

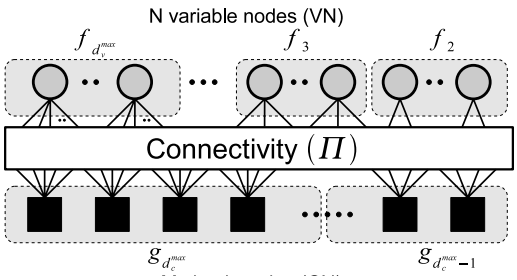

Figure 1: Tanner Graph for a LDPC Code

The number of edges supplying each node is called the node degree. If the node degree is constant for all CNs and VNs, the corresponding LDPC code is called regular, otherwise it is called irregular. Note that the communications performance of an irregular LDPC code is known to be generally superior to which of regular LDPC codes. The degree distribution of the VNs $f_{[d_v^{max},\ldots,3,2]}$ gives the fraction of VNs with a certain degree, with d_v^{max} the maximum variable node degree. The degree distribution of the CNs can be expressed as $g_{[d_c^{max},d_c^{max}-1]}$ with d_c^{max} the maximum CN degree, meaning that only CNs with two different degrees occur [14].

To obtain a good communications performance of an LDPC code the degree distribution should be optimized with respect to the codeword size N. The degree distribution can be optimized by density evolution as shown in [14]. Furthermore, the resulting Tanner graph should have cycles as long as possible to ensure that the iterative decoding algorithm works properly. A cycle in the Tanner graph is defined as the shortest path from a VN back to its origin without traveling an edge twice. Especially cycles of length four have to be avoided [13]. For details on the used U-S LDPC code the reader is referred to [2].

2.1 Decoding Algorithm

LDPC codes can be decoded using the message passing algorithm [3]. It exchanges soft-information iteratively between variable and check nodes. Updating the nodes can be done with a canonical, two-phased scheduling: In the first phase all variable nodes are updated, in the second phase all check nodes respectively. The processing of individual nodes within

one phase is independent and can thus be parallelized. The exchanged messages are assumed to be log-likelihood ratios (LLR). Each variable node of degree d_v calculates an update of message k according to:

$$\lambda_k = \lambda_{ch} + \sum_{l=0, l\neq k}^{d_v-1} \lambda_l, \qquad (2)$$

with λ_{ch} the corresponding channel LLR of the VN and λ_l the LLRs of the incident edges. The check node LLR update can be done in an either optimal or suboptimal way, trading off implementation complexity against communications performance. Our approach uses the suboptimal calculation to reduce implementation complexity.

The simplest suboptimal check node algorithm is the well-known Min-Sum algorithm [4], where the incident message with the smallest magnitude determines the output of all other messages:

$$\lambda_k = \prod_{l=0, l\neq k}^{d_c-1} \mathrm{sign}(\lambda_l) \cdot \min_{l=0, l\neq k} (|\lambda_l|). \qquad (3)$$

The resulting performance comes close to the optimal Sum-Product algorithm only for high rate LDPC codes (R $\geq$ 3/4) with relatively large CN degree d_c. It can be further optimized by multiplying each outgoing message with a message scaling factor (MSF) of 0.75. For lower code rates the communications performance strongly degrades.

2.2 Layered Decoding

Layered decoding applies a different message schedule than the classical two-phase decoding. It was originally proposed by Mansour [11] and denoted as turbo decoding message passing (TDMP), then it was referred to as layered decoding by Hocevar [5]. It is applicable to partly parallel architectures where not all nodes are processed in parallel. The basic idea is to process a subset of CNs and to pass the newly calculated messages immediately to the corresponding VNs. The VNs update their outgoing messages in the same iteration. The next CN subset will thus receive newly updated messages which improves the convergence speed and therefore increases communications performance for a given number of iterations [19].

3. LDPC Decoder Architectures

A partly parallel architecture template is sufficient to ensure the throughput for the UWB LDPC decoder. Thus the layered decoding approach is applicable. In this architecture P edges are processed per clock cycle. To ensure code rate and codeword size flexibility the functional nodes are realized in a serial manner. Thus each functional unit can accept one message per clock cycle.

To reach reasonable communications performance and minimize area, the check nodes are implemented with an optimized version of the Min-Sum algorithm (Section 2.1). The input channel values are represented by 6 bit values. For more details the reader is referred to [2].

4. Iteration Control

Iterative decoding algorithms show an inherent dynamic behavior, i.e., the number of iterations strongly depends on the transmission channel condition which is characterized by the signal-to-noise ratio which changes over time. Instead of using a fixed number of iterations, the number of iterations can be controlled by an intelligent iteration control mechanism. In [18] it was shown that iteration control is the most efficient technique for energy saving in a turbo decoder system without sacrificing communications performance. The arguments are also valid for LDPC decoding.

4.1 State-of-the-Art

An efficient iteration control has to distinguish between decodable and undecodable blocks at an early stage of the decoding process. LDPC codes already provide an implicit stopping criterion for decodable blocks by simply checking for an all-zero syndrome if a codeword has been successfully decoded, see Equation 1 . Normally, only this stopping criterion is used in state-of-the-art implementations. More advanced stopping criteria tackle especially the undecodable codewords [8].

As soon as a codeword is most certainly too corrupted to be successfully decoded, the decoder stops its decoding process. After detecting an undecodable codeword the LDPC decoder can be switched off (i.e., clock gating is activated on gate level) in low SNR regions instead of wasting the maximum number of iterations all the time and thus wasting energy. This stopping criterion is based on a monitoring of the variable nodes reliability (*VNR*).

$$\Lambda^m = \lambda_{ch}^m + \sum_{l=0}^{d_v^m - 1} \lambda_l^m, \; VNR = \sum_{m=0}^{N-1} |\Lambda^m| \qquad (4)$$

The *VNR* is the sum of the absolutes of all intermediate VN results. The calculation of this value has a very low implementation complexity. If the *VNR* does not change or decreases within two successive decoding iterations the decoding process is stopped. The stopping criterion has to be switched off when the *VNR* passes a threshold (VNR_{off}) which is SNR dependent. This threshold has to be determined only once for each code rate. With an appropriate VNR_{off} the loss in communications performance can be decreased almost to zero. For more detailed information on this criterion the reader is referred to [8].

A similar approach for undecodable blocks is presented in [15]. There the number of satisfied parity-checks is measured instead of the *VNR* . If the number of unsatisfied checks is bigger than a threshold or the increasing of the satisfied checks is below a certain threshold for a given number of iterations, the decoder stops and the block is marked as non decodable. This approach is claimed to be independent of the channel conditions. The false alarm rate can be lower than of the *VNR*§ method, but the saving of iterations is smaller.

4.2 Enhanced Iteration Control

Many systems operate most of time in a relatively high SNR region. That means that the input error rate of the channel decoder is very small or in other words, a lot of transmitted blocks are completely error free. In this case no decoding step is needed for a systematic code. All LDPC codes which are used in current standards are systematic codes.

In the layered architecture the iteration control has to calculate the syndrome for the inherent stopping criterion. The syndrome calculation depends on the check node operation. That means the syndrome is not complete until all check node operations for one iteration have finished. In other words, the decoder has to make at least one iteration to proof the parity check condition. If an error free block is received, the energy for this iterations is lost without gaining any new information.

The main idea of the new approach is to detect an error free block before the decoder starts at a lower complexity and a lower energy consumption than one decoder iteration. Therefore we propose to check the parity check condition of Equation 1 before the decoding step. If we detect an error free block no decoding is needed and the decoder can be completely switched off.

An efficient way to check the parity-check condition is to encode the received systematic information. An encoder encodes the received systematic bits and an *Error Detect logic*, compares the calculated parity bits with the received parity bits. If there is no different a valid codeword is received and the decoder would produce the same result. The

received data can direct pass the decoder and transferred to the next processing step, see Figure 2 . In this case no decoding step is needed. The U-S LDPC is encodable with a linear time complexity [2].

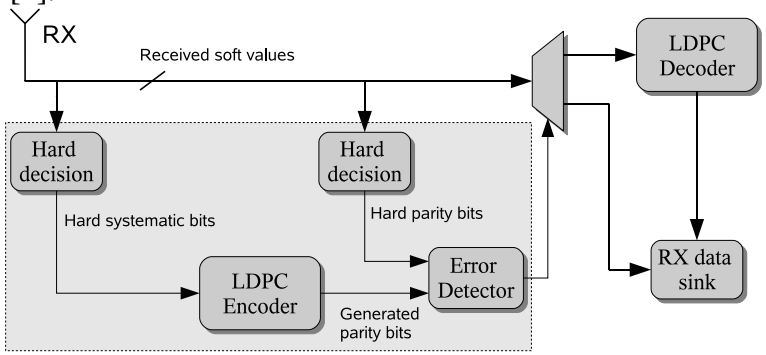

Figure 2: Enhanced Iteration Control

This approach can go one step further. A certain number of errors in the packet can be detected and corrected with this technique. If we assume a high SNR, the number of erroneous bits in one packet is very small. We can use the following heuristic to detect errors.

If only a few generated parity bits differ from the received ones, we can assume that the systematic part is correct with a certain probability. The parity part contains no useful information for the upper layers. The probability of a false alarm depends on the number of wrong parity bits the SNR and the code. If you choose the number of allowed different parity bits to one or two there is a guarantee for the U-S LDPC code and the decoder implementation of Section 3 that the decoder produces the same systematic information like the received one.

5. Results

In this section simulation results of the proposed iteration control in the WIMEDIA UWB simulation chain are given. Furthermore implementation results in a 65nm low power technology are presented.

5.1 WIMEDIA UWB System Model

The WIMEDIA UWB standard is based on a multiband OFDM air interface with and without frequency hopping. The overall US UWB band ranging from 3.1 GHz to 10.6 GHz is split into 14 sub-bands using 528 MHz of bandwidth with 128 OFDM sub-carriers. The sub-bands are grouped to form 5 band groups. Four of them contain 3 sub-bands and one consists of 2 sub-bands. In this paper we focus on the first band group ranging from 3.1 GHz to 4.8 GHz. In order to evaluate the communications performance of the WIMEDIA UWB standard a SystemC based simulation chain has been implemented for the full data path of the system. Key parameters of the WIMEDIA UWB system are depicted in Table 1.

The used LDPC code is specially designed for the UWB environment to get a low decoder complexity at a high throughput and a high coding gain. [2][1].

Table 1: WIMEDIA Physical Layer Parameters

Parameter	Value
Data rate	53Mbit/s to 960Mbit/s
Data carriers	100
FFT size	128 points
Symbol Duration	312.5ns (incl. Guard)
Channel Coding	CC with K = 7, LDPC Code
Carrier Modulation	QPSK, DCM, 16-QAM
Channel Models	CM1, CM2, CM3, CM4

5.2 Simulation Results

We selected the 16-QAM case with an air data rate of 960 Mbit/s and the 128 point FFT for our simulations. The input quantization is 6 bit for the LDPC decoder, which yields a communications performance loss of less than 0.2 dB in comparison to a floating point implementation. This is a good compromise between implementation complexity and communications performance degradation.

Figure 3 shows the average number of iterations for various SNRs with the inherent LDPC stopping criterion only and the criteria for decodable and undecodable blocks presented in the Section 4.1 . The threshold VNR_{off} for the iteration control for undecodable blocks is selected in such a way that the communications performance loss is lower than 0.01 dB. The third curve shows new the enhanced stopping criterion combined with the state-of-the-art criteria. We can make the following observations:

• The detection mechanism for undecodable blocks gives the highest gain in the non-convergence SNR region (below 15 dB). It saves up to 4.5 iterations, i.e. 45%, assuming a maximum number of 8 iterations. There the state-of-the-art approach gives the same result like the new enhanced technique.

• In the waterfall SNR region (15 dB to 23 dB) the state-of-the-art criteria yields a maximum average number of only 5.5 iterations. That means that the iteration control mechanism can save up to 32% of iterations.

• In the error floor SNR region (above 23 dB) the new enhanced criterion can decrease the number of iterations up to 50% compared to the inherent criterion only. For a high SNR region the mean number of iterations is less than one.

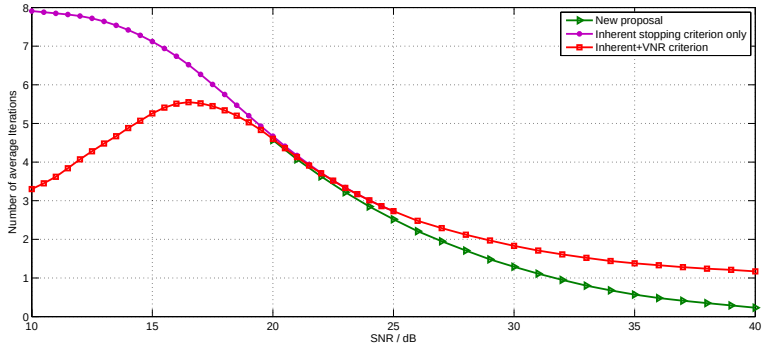

Figure 3: Average Number of Iteration of the LDPC Decoder with Different Stopping Criteria vs. SNR

The new approach saves energy for blocks which are error free at the receiver. The disadvantage is that we have to encoder each received block before decoding. Therefore the energy saving of this approach is related to the number of error free received blocks. Figure 4 shows the input block error rate at the receiver. A linear growing of the number of

error free blocks with a increasing SNR can be observed. At the highest SNR point of 40 dB, 80% of the received blocks are completely error free and no decoding step is needed.

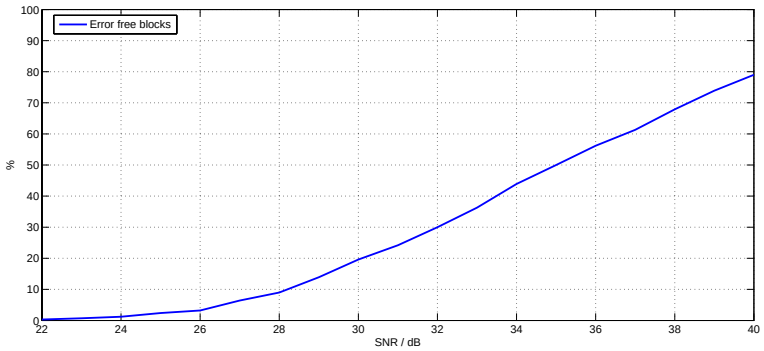

Figure 4: Input PER before the Decoder vs. SNR

Figure 5 shows the energy consumption for decoding one block of the current used CC code and the proposed LDPC code in a FPGA implementation [10]. It is shown that the energy consumption of the LDPC decoder increases linearly with the number of iterations. In this environment decoding one LDPC codeword up to 7 iterations needs less energy than decoding one CC codeword.

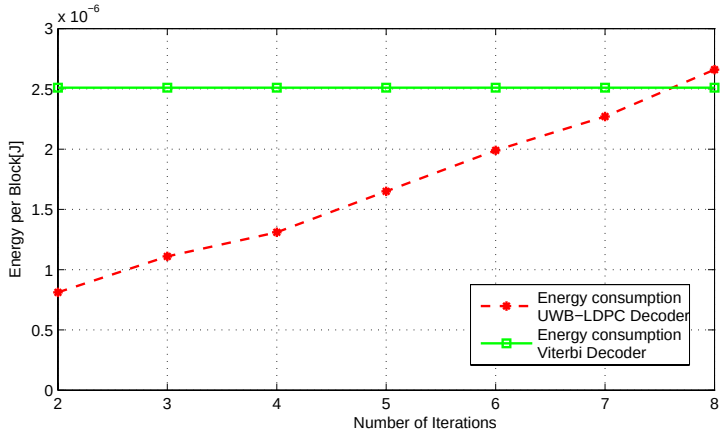

Figure 5: Eergy for UWB-LDPC and Viterbi Decoder, Energy per Block

5.3 Implementation Results

The implementation complexity should be remarkable lower than for one decoder iteration to get a energy saving. Table 2 shows the arithmetic complexity for one encoding step in comparison to one decoder iteration. Only the basic algorithmic operations are presented.

Encoding can be performed with 3600 operations for one block, 3300 *XOR* operations for encoding and 300 *XORs* for comparing the parity bits. One decoding iteration needs 13200 operations. The *Min* operation is the check node calculation, the *Add* operation is for the variable node operation and the *XOR* operations is needed for proof of the parity checks. The decoder uses soft information for each bit, thus the operations have a width of minimum 6 bits. In contrast, the encoder operates only with single hard bits. In summary the encoding complexity is less than 5% in comparison to one soft decoding iteration.

Table 2: Complexity Encoding vs. Decoding Process

LDPC Code	$f_{[3,2]} = \{3/4, 1/4\}$, $g_{[11]} = \{1\}$	
Code word size	1200 bits	
Code rate	3/4	
Operations	Encoder	Decoder
XOR	3600	3300
Min	-	6600
Add	-	6600
Overall	3600	16500

The low complexity of the encoding process of the U-S LDPC code is reflected in the implementation results. Table 3 shows implementation results for an UWB LDPC encoder and decoder. The area is only 0.014 mm^2 compared to the decoder implementation of 0.207 mm^2. The leakage current in an ASIC implementation is proportional to the active area. By switching of the decoder in case of an error free block the active silicon area is less than 10% in comparisons to the active mode. The additional logic in the decoder for the state-of-the-art iteration control is less than 0.01 mm^2 and is contained in the results. The overall complexity for the iteration control including the state-of-the-art and the new one is 0.024 mm^2, which corresponds to 11% of the LDPC decoder [1] complexity. Encoding the received information before decoding increases the latency by 33%.

Table 3: Synthesis Results for the UWB-LDPC Encoder

LDPC Code	$f_{[3,2]} = \{3/4, 1/4\}$, $g_{[11]} = \{1\}$	
Codeword Size	1200 bit	
Code Rate	3/4	
Area[mm^2] 65nm @ 528Mhz @ 6 iterations		
Type	Encoder	Decoder
Parallelism	30	
Logic	0.010	0.112
Memory	0.004	0.095
Overall Area	0.014	0.207
Throughput	3.39 Gbit/s	530 Mbit/s
Latency	0.26 µs	0.77µs

6 . Conclusions

State-of-the-art implementations need at least one decoder iteration to detect an error free received block. We presented an advanced iteration control for LDPC coding which can detect error free blocks before the decoding step with a low complexity by using an LDPC encoder. The new iteration control scheme in cooperation with the state-of-the-art ones reduces the number of iterations and thus energy for all channel conditions for an LDPC decoding scenario. In general the new approach can be used for all systematic codes, which have a linear time encoding complexity.

Acknowledgements

This work has been partly carried out in the framework of the IST project PULSERS Phase II (FP6-027142), Pervasive Ultra-wideband Low Spectral Energy Radio Systems Phase II), which is partly funded by the European Union.

References

[1] M. Alles and T. Brack and T. Lehnigk-Emden and F. Kienle and N. Wehn and F. Berens and A. Ruegg. A Survey on LDPC Codes and Decoders for OFDM-based UWB Systems. Proc. 56th Vehicular Technology Conference (VTC Spring '07), Dublin, Ireland, 2007.

[2] T. Brack and F. Kienle and T. Lehnigk-Emden and M. Alles and N. Wehn and F. Berens. Enhanced Channel Coding for OFDM-based UWB Systems. Proc. International Conference on Ultra-Wideband (ICUWB 2006), pages 255--260, Waltham, Massachusetts, 2006.

[3] R. G. Gallager. Low-Density Parity-Check Codes. M.I.T. Press, Cambridge, Massachusetts, 1963.

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

[5] D.E. Hocevar. A reduced complexity decoder architecture via layered decoding of LDPC codes. Proc. IEEE Workshop on Signal Processing Systems (SiPS '04), pages 107--112, Austin,USA, 2004.

[6] IEEE 802.11n. Wireless LAN Medium Access Control and Physical Layer specifications: Enhancements for Higher Throughput. IEEE P802.11n/D3.0, 2007.

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

[8] F. Kienle and N. Wehn. Low Complexity Stopping Criterion for LDPC Code Decoders. Proc. 2005-Spring Vehicular Technology Conference (VTC '05 Spring), pages 606--609, Stockholm, Schweden, 2005.

[9] Sang-Min Kim and Jun Tang and Keshab K. Parhi. Quasi-Cyclic Low-Density Parity-Check Coded Multiband-OFDM UWB Systems. IEEE International Symposium on Circuits and Systems, pages 65-68, 2005.

[10] Timo Lehnigk-Emden and Christian Brehm and Torben Brack and Norbert Wehn and Friedbert Berens and Cem Derdiyok. Energy Consumption of Channel Decoders for OFDM-based UWB Systems. Proc. IEEE International Conference on Ultra-Wideband (ICUWB 2007), Singapore, 2007.

[11] M.M. Mansour and N.R. Shanbhag. High-Throughput LDPC Decoders. IEEE Transactions on Very Large Scale Integration Systems, 11(6):976--996, 2003.

[12] Khiam-Boon Png and Xiaoming Peng and Francois Chin. Performance Studies of a Multi-band OFDM System Using a Simplified LDPC Code. IEEE Ultra-Wideband Systematic and Technologies 2004, 2004.

[13] T. Richardson and R. Urbanke. The Renaissance of Gallager's Low-Density Pariy-Check Codes. IEEE Communications Magazine, 41:126--131, 2003.

[14] T.J. Richardson and M.A. Shokrollahi and R.L. Urbanke. Design of Capacity-Approaching irregular Low-Density Parity-Check Codes. IEEE Transaction on Information Theory, 47(2):619--637, 2001.

[15] Donghyuk Shin and Kyoungwoo Heo and Sangbong Oh and Jeongseok Ha. A Stopping Criterion for Low-Density Parity-Check Codes. Proc. 56th Vehicular Technology Conference (VTC Spring '07), Dublin, Ireland, 2007.

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

[17] J. Tang and T. Bhatt and V. Stolpman. Modified Min-Sum Algorithm for LDPC Decoders in UWB Communications. Proc. IEEE International Conference on Ultra-Wideband 2006, Boston, Massachusetts, USA, 2006.

[18] M. J. Thul and T. Vogt and F. Gilbert and N. Wehn. Evaluation of Algorithm Optimizations for Low-Power Turbo-Decoder Implementations. Proc. 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '02), pages 3101--3104, Orlando, Florida, USA, 2002.

[19] Juntan Zhang and Marc P. C. Fossorier. Shuffled Iterative Decoding. IEEE Transactions on Communications, 53(2):209--213, 2005.