Performance and Complexity Analysis of advanced Coding Schemes in the 4MORE Project

Timo Lehnigk-Emden, Torben Brack, Matthias Alles, Norbert Wehn
Microelectronic System Design Research Group, University of Kaiserslautern, 67663 Kaiserslautern, Germany
{lehnigk, brack, alles, wehn}@eit.uni-kl.de

Marie-Hélène Hamon, Pierre Pénard, Rodolphe Legouable
FTR&D RESA/BWA laboratory, 35512 Cesson Sevigne cedex, France
{mhelene.hamon, pierre.penard, rodolphe.legouable}@francetelecom.com

Friedbert Berens
Computer Systems Division - UWB BU, STMicroelectronics, 1228 Plan-les-Ouates/Geneva, Switzerland
friedbert.berens@st.com

***Abstract*— The IST project 4MORE [1] has the objective to implement SoC based demonstrator platform for MC-CDMA based future mobile radio system. The basic system parameters are based on a refinement of the MATRICE [2] air interface definition. As one of the elementary building blocks of a future airinterface two innovative channel coding schemes have been investigated in the scope of the 4MORE project, namely Low-Density Parity-Check codes (LDPCC) and Duo-Binary Turbo Codes (DB-TC), respectively. In this paper, the communications performance and the hardware architecture of the two channel coding schemes will be presented in the context of the 4MORE project. The communications performance is demonstrated for three different application scenarios with two different code rates each. Furthermore, the architectures of two decoder realizations are mapped on Xilinx FPGA platforms. Finally, we compare the architecture efficiency of the two decoders taking into account the throughput and the number of used FPGA slices.**

I. Introduction

Low-Density Parity-Check Codes (LDPCC) and Duo-Binary Turbo Codes (DB-TC) are two of the best performing channel codes known today.

LDPC codes were invented by Gallager in 1963 [3], they were almost forgotten for nearly 30 years. Rediscovered by MacKay in the mid-90s and enhanced to irregular LDPC codes by Richardson et. al. in 2001 [4], they are now to be used for forward error correction in a vast number of upcoming standards like DVB-S2 [5], WiMax (IEEE802.16e), and wireless LAN (IEEE 802.11n). Providing very high decoding throughput and outstanding communications performance, they will probably become the channel coding scheme of choice for years to come.

The invention of Turbo Codes by C. Berrou and al. in 1993 [6] has triggered a significant amount of research in the field of error-correction codes. Decoded iteratively, these codes enable outstanding performance, getting closer to theoretical limits than any other codes. The principle has been applied then to other codes, leading to new families of advanced error-correction codes. Turbo Codes have been introduced into several standards, addressing a broad range of systems: UMTS, cdma2000, DVB-RCS [7], DVB-RCT [8], WiMax (IEEE 802.16) [9]. Duo-Binary Turbo Codes have been introduced more recently, enhancing several aspects of the first Turbo Codes. This paper would like to give a fair comparison taking into account different operational conditions, communications performance and implementation complexity of the two competing channel coding schemes in the scope of a potential future mobile radio system. The paper is structured as follows: Section II shortly introduces Duo-Binary Turbo and LDPC codes, followed by their decoding strategies. Architecture templates of the two decoders are presented in section Section III. Section IV describes the simulations environment. In Section V synthesis result on an FPGA implementation and the communications performance for three application scenarios are presented.

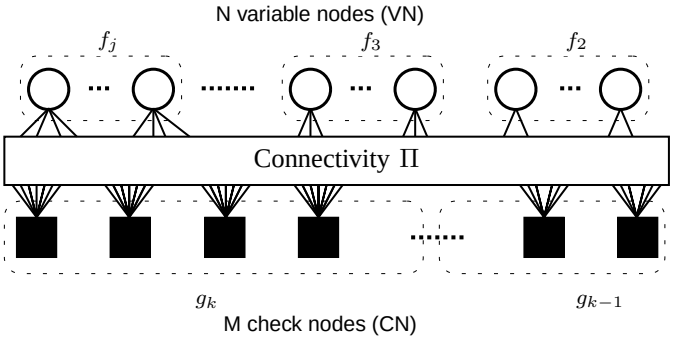

Fig. 1. Tanner Graph for an Irregular LDPC Code

II. Codes and Decoding Strategies

A. Low-Density Parity-Check 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

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

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 [4], 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 irregular LDPC code with N variable and M check nodes with a resulting code rate of $R = (N - M)/M$.

The number of edges on each node is called the node degree. If the node degree is identical for all variable nodes, 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 $f_{[j,\ldots,3,2]}$ gives the fraction of VNs with a certain degree, with j the maximum variable node degree. The degree distribution of the CNs can always be expressed as $g_{[k,k-1]}$ with k the maximum CN degree, meaning that only CNs with degree k and $k-1$ will occur.

1) Optimal Decoding: 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 straightforward parallelized.

The exchanged messages are assumed to be log-likelihood ratios (LLR). Each variable node of degree i calculates an update of message k according to:

$$\lambda_k = \lambda_{ch} + \sum_{l=0, l\neq k}^{i-1} \lambda_l, \tag{2}$$

with λ_{ch} the corresponding channel LLR of the VN and λ_i the LLRs of the incident edges. The check node LLR update can be done in an either optimal or suboptimal way, trading of implementation complexity against communications performance. Optimal check node decoding can be done calculating:

$$tanh(\lambda_k/2) = \prod_{l=0, l\neq k}^{i-1} tanh(\lambda_l/2), \tag{3}$$

Equation 2 and Equation 3 combined yield the Sum-Product or Belief-Propagation algorithm. Hardware realizations of this function can become very complex, especially if different CN degrees have to be supported.

2) Suboptimal Decoding: The simplest suboptimal check node algorithm is the well-known Min-Sum algorithm [10], where the incident message with the smallest magnitude mainly determines the output of all other messages:

$$\lambda_k = \prod_{\forall i, i\neq k} \operatorname{sign}(\lambda_i) \cdot \min_{\forall i, i\neq k} (|\lambda_i|) \tag{4}$$

The resulting communications performance can be improved by scaling the updated messages with a so called message scaling factor (MSF). In this paper, we will always refer to the Min-Sum algorithm while assuming an MSF of 0.75. 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. For lower code rates the communications performance strongly degrades.

3) Optimized Update for Variable Nodes: For most linear encodable LDPC codes, the update of variable nodes corresponding to the parity information is inherently simple due to their fixed node degree of two. As Equation 2 implies, the incoming messages at one edge are sent out on the second edge plus the corresponding channel LLR value and vice versa. Optimized update scheme means that these nodes are updated as soon as possible, even during one iteration. This results in an remarkable faster convergence in the decoding process, thus less iterations are necessary to achieve the same communications performance [11]. This method is most effective for low rate codes because of their large number of VN nodes, for example 50% for code rate of $1/2$.

B. Duo-Binary Turbo Codes

DB-TC belong to a new class of convolutional Turbo Codes, introduced in [12][13] and generalized in [14]. These Turbo Codes mainly differ from the classical binary convolutional Turbo Codes by their constituent codes: instead of binary circular recursive systematic constituent (CRSC) codes, they are based on m-ary RSC codes. In the case of m = 2, quaternary CRSC codes are employed (i.e. information bits are encoded by couples and not individually as in the classical approach) and the resulting Turbo Code is labelled Duo-Binary Turbo Code.

1) Interleaving: Permutations from the classes of Almost Regular Permutations, presented in [15], are particularly appropriate for the internal interleaving of Turbo Codes. These permutations have been proposed and adopted into different standards: IEEE 802.16 [9], DVB-RCS [7] and DVB-RCT [8]. Interleaving is specified in these standards by one equation and four parameters, which are adjusted according to the information size. The internal interleaver of the Duo-Binary Turbo Code takes advantage of the particularity of these Turbo Codes, i.e. the double-binary input, to improve the design of the permutation. Interleaving is performed on two levels: Intra-couple permutation, performed inside couples: for a predetermined set of couples, the pairs of bits are switched. Inter-couple permutation, performed between couples: similarly to the bit interleaving in a classical turbo code, the couples are interleaved. This two-fold permutation improves asymptotic performance as it instils some irregularity and disorder in the interleaving process, thanks to the intra-couple switching applied to a part of the couples.

2) Advantages over Binary Turbo Codes: Duo-Binary Turbo Codes benefit from several advantages over the binary Turbo Codes:

- Their main characteristic is their double-binary input. As the corresponding trellis has shorter path competitions than the binary trellis, the error packets after decoding are shorter, leading to faster convergence in the iterative decoding process [13]. Furthermore, as the decoder outputs two decisions per clock period, the complexity and latency per decoded bit is significantly reduced.
- Due to the use of circular encoding, the encoder is initialized to a circulation state and retrieves this initial circulation state at the end of encoding operation, and consequently no tail bits are required to close the trellis.
- Asymptotic performances are improved thanks to the two-level permutation.
- Duo-Binary Turbo Codes are highly flexible, as they can be adjusted easily to different blocksizes. The permutation is defined through only one equation and 4 parameters, and only these 4 values will change from one blocksize to another. Duo-Binary Turbo Codes can therefore be defined for a broad range of blocksizes with minimal storage requirements.

3) Decoding: The general structure of a Turbo Decoder is represented on figure Figure 2. Soft-Input Soft-Output (SISO) decoding algorithms are employed. Optimal decoding is performed according to the Maximum A Posteriori (MAP) criterion, but suboptimal decoding algorithms such as the Max-Log-MAP are usually employed. The information exchanged between the decoders are Log-Likelihood Ratios (LLR). Extrinsic Information is exchanged between the two decoders to gradually improve the decoding performance and help the two decoders to converge to a decision.

III. Decoder Architecture Templates

A. LDPC Decoder

A generic architecture template becomes mandatory to allow for an efficient design reuse for different target applications. For an efficient architecture, some restrictions have to be set up:

- A partly parallel approach which can process a certain number of edges per clock cycle concurrently is necessary to support reasonable throughput and flexibility at the same time.

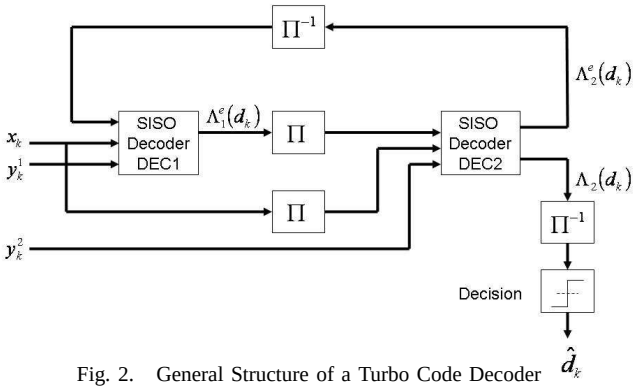

Fig. 2. General Structure of a Turbo Code Decoder

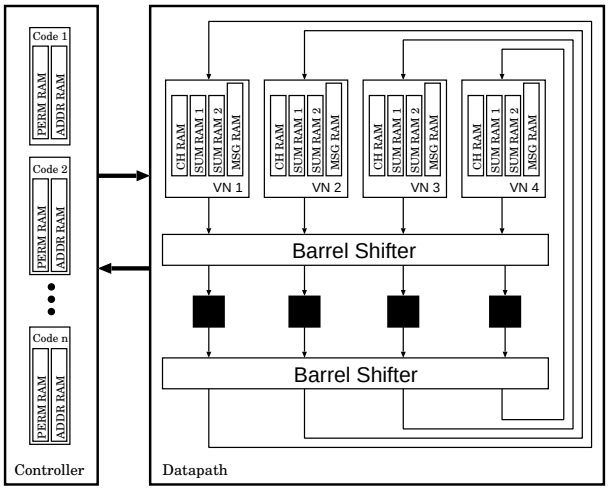

Fig. 3. LDPCC Decoder Architecture Template

- The message exchange network which represents the tanner graph has to be flexible and of manageable complexity without limiting the usable code space to much. Especially routing congestion has to be avoided.
- Check node processing has to be done in a serial manner. While limiting processing speed to one message per clock cycle for each node, this also reduces potential memory conflicts resulting in an enhanced flexibility and support of a much larger code range without the excessive area overhead of parallel check node implementations.

The actual architecture template is shown in Figure 3. The *CN Units*, represented as black boxes, implement the Min-Sum algorithm introduced in Section II-A.2. They read the message LLR belonging to the incident edges of the covered check nodes from the *Message RAM* which have to be subtracted from the sum of the related channel LLR in the *Channel RAM* and the accumulated extrinsic message LLR stored in one of the *Sum RAMs* during the previous iteration. This subtraction is carried out by the *VN Units* reflecting the variable node calculation given by Equation 2. After the check node processing, the resulting extrinsic information is stored back in the appropriate *Message RAM* and accumulated up into the other *Sum RAM*. At the beginning of the next iteration, the two *Sum RAMs* are interleaved. This technique was already presented in [16]. The architecture is able to process the optimized update scheme, see Section II-A.3.

All messages are exchanged via two *Barrel Shifters* which are controlled accordingly to the *Code Vectors* stored inside the *Controller*. It also controls the CN and VN Units and accessing the RAMs. The

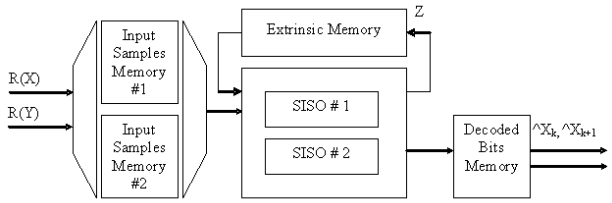

Fig. 4. IP Module Diagramm

Code Vectors represent the LDPC codes to be applied and can be either hard-coded or reloaded during runtime. By using barrel shifters for the message exchange network, the decoder can process all LDPC codes based on permutation matrices. The construction of LDPC codes which can be adapted to barrel shifter based architectures was already demonstrated in [17].

B. DB-TC Decoder

In the frame of the project, the decoder has to be implemented in an FPGA device. Starting from an IP block developed by TurboConcept for this purpose [18], the required throughput is simply achieved by duplicating the base module as many times as required. Through an input control unit, the data packets of the incoming flow are successively multiplexed on the first available decoding module. Each of them performs all the iteration of the decoding process, and at the output, the decoded data flow is re-multiplexed in the right order.
A simplified diagram of the architecture of the IP is shown in figure Figure 4. Its main parts are:

- 2 input memories storing the input samples. Each one contains 1 block.
- A SISO unit
- A memory which contains the extrinsic data for one block
- An output memory storing the decisions bits

The SISO processor implements a simplification of the optimal Maximum A Posteriori algorithm called Sub-MAP. The major simplification consists in working in the logarithmic domain. Therefore, the expensive probabilities multiplications are replaced by additions and comparisons. The module contains a backward recursion unit, a forward recursion unit, and an extrinsic information computation unit. Each SISO processor decodes 2 bits per clock cycle, and the resulting throughput of one decoding module is given by:

$$D = \frac{BSIZE}{BSIZE+8} * \frac{Fclock}{Iterations+1} \quad (5)$$

In the overall architecture of the project, the full bit-rate is obtained by multiplying this modular throughput by the number of modules, since there is no overhead introduced by the multiplexing hardware. One of the mains features of the turbo-code solution is its flexibility. The same pieces of hardware may be used for a very large range of block sizes, and coding rates. The only thing to change from one case to the other is to define the right permutation parameters and puncturing pattern. Therefore, in a given system, the basic TC decoding hardware complexity is driven by the biggest block size.

IV. Simulation environment

The 4MORE (4G MC-CDMA Multi Antenna System on Chip for Radio Enhancements) project [1] investigates broadband mobile radio systems beyond 3G.
It features multiple antennas (MIMO), multi-carrier modulation (OFDM), CDMA, and further advanced coding techniques.
Our simulations are performed using the 4MORE simulation chain,

Carrier Frequency	5.2 Ghz
Channel Bandwidth	60.44 Mhz
MIMO Scheme	2xTX, 2xRX Alamouti
# of OFDM Carriers	624
CDMA Spreading Factor	32 or 16
Channel Model	ETSI BRAN E

TABLE I
SIMULATION CHAIN PARAMETER

Parameter	Scenario 1	Scenario 2	Scenario 3
Modulation Scheme	QPSK	16-QAM	64-QAM
Frame Size	1008 bit	2016 bit	6048 bit
Code Rates R	$1/2, 4/5$	$1/2, 4/5$	$1/2, 4/5$
Codeword LDPCC	1000 bit	2000 bit	6000 bit
Codeword DB-TC	992/1000 bit	1984/2000 bit	5952/6000 bit
# of Users	32	32	16
System Throughput	41.6 Mbps	83.2 Mbps	124.8 Mbps
Throughput/User	1.2 Mbps	2.5 Mbps	7.5 Mbps

TABLE II
SIMULATION SCENARIOS

which is the reference simulation platform for the hardware demonstrator. In this paper only the downlink direction from the base station to the mobile station is studied. The used 4MORE standard parameter are shown in the Table I. The simulations are done, using the ETSI BRAN E channel which is based on the 3GPP SCM MIMO channel, with a mobile speed of 60 km/h, for details see [19].

V. RESULTS

A. Simulation Results

In our simulations we investigate three different scenarios, described in Table II. They differ in the modulation scheme, the codeword size and the spreading factor to fulfill the constraints of the system. The first and the second scenario support 32 active users with a gross user data rate of 1.2 Mb/s deploying QPSK and 2.5 Mb/s deploying 16-QAM, respectively. The third one is a high throughput scenario adapted for a high SNR region with 16 users, with a gross user data rate of 7.5 Mb/s using 64-QAM. The main goal of the three scenarios is to investigate the communications performance of the DB-TC and the LDPCC with a short (approx. 1000 bits), a medium (approx. 2000 bits) and a large (approx. 6000 bits) codeword size, each with a code rate of $1/2$ and $4/5$. In order to adapt the codeword size of the channel coders to the needed frame size zero padding is used. For the LDPCC, the Min-Sum algorithm with MSF and optimized upate is used as decoding scheme in a floating point model, the DB-TC decoder uses a quantized Max-Log-Map algorithm, with a quantization level which has no impact on the communications performance. The DB-TC decoder uses 8 iterations. The LDPC deocder uses 10 iterations for codes with the code rate of $4/5$ and 20 iterations for the codes with the lower code rate of $1/2$. Figure 5, 6 and 7 show the communications performance of the codes in the three mentioned scenarios.
For the higher rate $4/5$, the communications performance of the LDPCC is about 0.2 db better in Scenario 1. In Scenario 2 and 3 the DB-TC is about 0.4 db better than the LDPCC. For the code rate $1/2$ the DB-TC outperforms the LDPCC up to 1 db in communications performance in term of FER, with the smallest gains in the QPSK scenario and the largest gain in the 16-QAM scenario. By using the optimal decoding algorithms (Belief-Propagation respectively Log-MAP) and more iterations in case of the LDPCC the communications performance of the two codes could be increased at the expense of higher complexity and lower throughput.

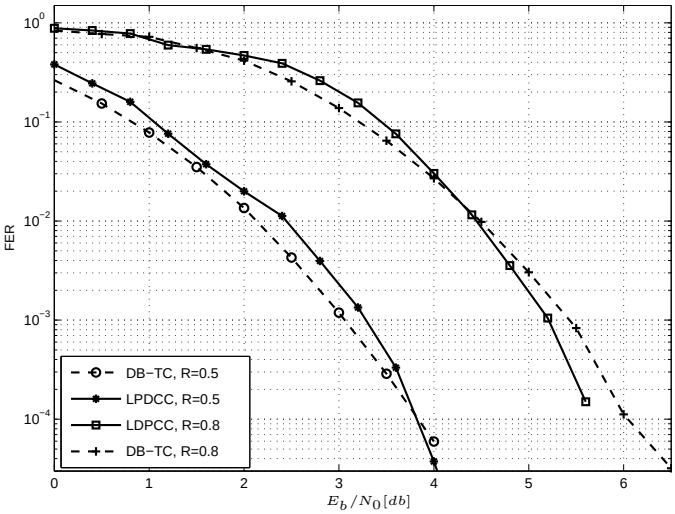

Fig. 5. Communications Performance in Scenario 1 (QPSK)

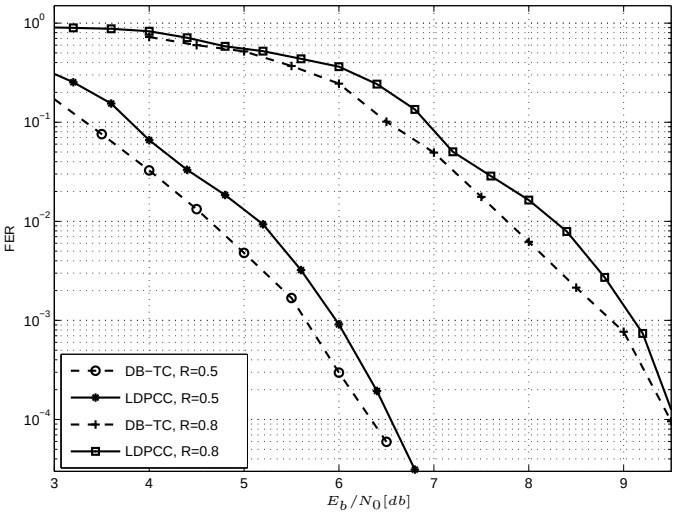

Fig. 6. Communications Performance in Scenario 2 (16-QAM)

B. Synthesis Results

The tables III and IV show the synthesis results for the two decoders. The proposed flexible LDPC decoder can support blocks from 1000 up to 6000 bit with a code rate range from R=$1/2$ to R=$9/10$. The maximum throughput for the largest block size with a code rate of R=$9/10$ is 500 Mbps. The DB-TC can support block sizes from less than 100 bits with a coding rate from R=$1/3$ to R=$7/8$
To get a better view of the two decoders, a VLSI efficiency metric is introduced, describing the throughput related to the number of used slices. Table V compares the throughput and the efficiency of the two decoders for all three scenarios. Note that the proposed DB-TC decoder doesn't support the largest blocksize of 6000 bits.
The DB-TC decoder is more efficient for the Scenarios 1 and 2 in case of the code rate R=$1/2$, in all other cases the LDPC decoder has a smiliar better VLSI efficiency.

VI. CONCLUSION

In this paper we have presented a fair comparison of the two most enhanced channel coding schemes for the use in future mobile radio systems, namely Duo-Binary Turbo Codes and Low-Density Parity-Check Codes. The coding scheme of choice depends on the codeword size, the code rate, and the system requirements regarding throughput and area. If the best possible communications performance has to be

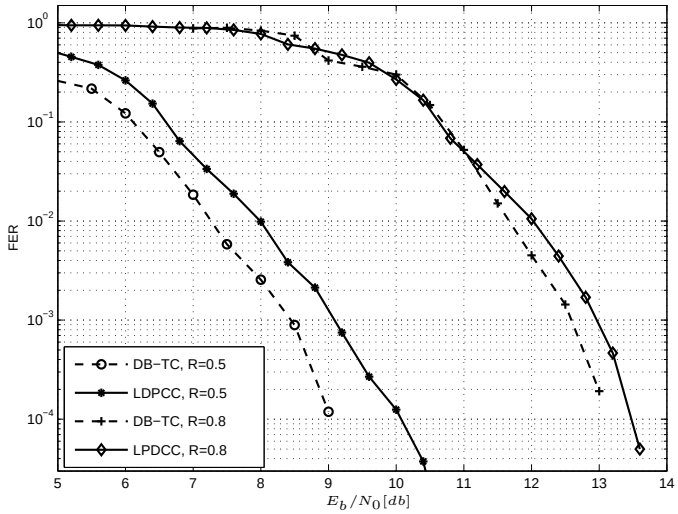

Fig. 7. Communications Performance in Scenario 3 (64-QAM)

Block Size	1000-6000 bit
Code Rate	$1/2$-$9/10$
Parallelism	100
Quantization	6 bit
Algorithm	Min-Sum+MSF Optimized VN-Update
Iterations	5-20
Comm. Perform.	see Section V-A
XC4VLX100 FPGA@100Mhz	
Bit-Nodes	4849 Slices
Check-Nodes	12500 Slices
Controller	148 Slices
Network	4830 Slices
RAM	96 BRAM
Code Vectors	5 BRAM
Overall Logic	23145 Slices
Overall Memory	101 BRAM
Throughput	*60-500 Mbps*

TABLE III
SYNTHESIS RESULTS FOR THE FLEXIBLE LDPC CODE DECODER

achieved, the DB-TC should be used for short blocks with a low-to-medium code rates, while the LDPCC is the suitable solution for long blocks combined with high code rates. Due to the inherent parallelism in the LDPC code structure very high throughput applications (up to 500 Mpbs in this case) can be implemented more efficiently. The DB-TC decoder is scalable downto less than 100 infobit which is not possible for the LDPCC decoder, because of the parallization level and the weak code performance of short LDPCC. In a next step a complexity analyses using a state-of-the-art silicon process will be carried out in order to get more realistic implementation figures for an ASIC implementation.

ACKNOWLEDGMENT

The work presented in this paper was supported by the European IST project 4MORE (4G MC-CDMA multiple antenna system On chip for Radio Enhancements [1]).

REFERENCES

[1] "MATRICE - MC-CDMA Transmission Techniques for Integrated Broadband Cellular Systems project." [Online]. Available: http://www.ist-matrice.org

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

Decoded Block Size	96-2048 bit
Code Rate	$1/3$-$6/7$
Input Quantization	4 bit
Algorithm	Max-Log-MAP+ESF
Iterations	8
Comm. Perform.	see Section V-A
XC2VP7-7 FPGA@86Mhz	
Overall Logic	2923 Slices
Overall Memory	13 BRAM
Throughput	8.8-9.5 Mbps

TABLE IV
SYNTHESIS RESULTS FOR THE DB-TC DECODER

Decoder	**DB-TC**		**LDPCC**	
Code Rate R	$1/2$	$4/5$	$1/2$	$4/5$
# of Iterations	8		20	10
Throughput [Mbps]				
Scenario 1	9.41	9.46	61.0	135.6
Scenario 2	9.48	9.51	70.4	168.4
Scenario 3	9.53	9.54	78.5	200.8
Efficiency [Mbps/Slices]				
Scenario 1	32.2	32.4	26.3	*58.6*
Scenario 2	32.4	32.5	30.4	*72.8*
Scenario 3	32.6	32.6	33.9	*86.8*

TABLE V
THROUGHPUT AND VLSI EFFICIENCY

[3] T. Richardson and R. Urbanke, "The Renaissance of Gallager's Low-Density Pariy-Check Codes," *IEEE Communications Magazine*, vol. 41, pp. 126–131, Aug. 2003.

[4] "Digital Video Broadcasting (DVB) Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications," ETSI. [Online]. Available: www.dvb.org

[5] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes," in *Proc. 1993 International Conference on Communications (ICC '93)*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[6] "Interaction Channel for Satellite Distribution System," ETSI, December 2000.

[7] "Interaction Channel for Digital Terrestrial Television," ETSI, August 2001.

[8] "Local and Metropolitan Area Network, Part 16: Air Interface for Fixed Broadband Wireless Access Networks," IEEE.

[9] 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*, Brest, France, Sept. 2003, pp. 451–454.

[10] 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.

[11] C. Berrou and M. Jezequel, "Nonbinary convolutional codes for turbo coding," *Electronic Letters*, vol. 35, no. 1, pp. 39–40, January 1999.

[12] C. Berrou, M. Jezequel, C. Doullard, and S. Kerouedan, "The advantages of nonbinary turbo codes," in *Proceedings of Information Theory Workshop*, Cairns, Australia, September 2001, pp. 61–63.

[13] C. Douillard and C. Berrou, "Turbo Codes with rate-m/(m+1) constituent convolutional codes," *IEEE Trans. On Communications*, vol. 53, no. 10, pp. 1630–1638, October 2005.

[14] C. Berrou, Y. Saouter, C. Douillard, S. Kerouedan, and M. Jezequel, "Designing good permutations for Turbo Codes: toward a single model," in *Proceedings of ICC 2004*, vol. 1, June 2005, pp. 341–345.

[15] E. Boutillon, J. Castura, and F. Kschischang, "Decoder-first code design," in *Proc. 2nd International Symposium on Turbo Codes & Related Topics*, Brest, France, Sept. 2000, pp. 459–462.

[16] F. Kienle and N. Wehn, "Design Methodology for IRA Codes," in *Proc. 2004 Asia South Pacific Design Automation Conference (ASP-DAC '04)*, Yokohama, Japan, Jan. 2004, pp. 459–462.

[17] TurboConcept, Product Page TC1000. [Online]. Available: www.turboconcept.com/pages/prod_tc1000pag.html

[18] "Project Broadband Radio Access Networks (BRAN); Hyplerlan Type 2," ETSI, Oct 1999.