Transactions Papers

On Complexity, Energy- and Implementation-Efficiency of Channel Decoders

Frank Kienle, *Member, IEEE,* Norbert Wehn, *Senior Member, IEEE*, and Heinrich Meyr, *Life Fellow, IEEE*

***Abstract*—Future wireless communication systems require efficient and flexible baseband receivers. Meaningful efficiency metrics are key for design space exploration to quantify the algorithmic and the implementation complexity of a receiver. Most of the current established efficiency metrics are based on counting operations, thus neglecting important issues like data and storage complexity.**

In this paper we introduce suitable energy and area efficiency metrics which resolve the afore-mentioned disadvantages. These are decoded information bit per energy and throughput per area unit. Efficiency metrics are assessed by various implementations of turbo decoders, LDPC decoders and convolutional decoders. An exploration approach is presented, which permit an appropriate benchmarking of implementation efficiency, communications performance, and flexibility trade-offs. Two case studies demonstrate this approach and show that design space exploration should result in various efficiency evaluations rather than a single snapshot metric as done often in state-of-the-art approaches.

***Index Terms*—Channel coding, algorithmic complexity, energy efficiency, design space exploration.**

I. Motivation

TODAY, high-end smart phones have to support multiple radio standards, advanced graphic- and media applications and many other applications resulting in a workload of about 100 giga operations per second in a power budget of 1 Watt [1]. The baseband processing in the radio part (mainly front-end processing, demodulation and decoding) requires more than 50% of the overall workload in a state-of-the-art 3.5G smart phone. To achieve higher spectral efficiency new transmission techniques like MIMO will be established. However, this will increase the workload even further. Thus, there is a strong need for *efficient* wireless baseband receivers. The overall efficiency of a baseband receiver depends on

- *communications efficiency*: expressed by the spectral efficiency and signal-to-noise ratio (SNR). The requirements on the communications efficiency have the largest impact on the selected baseband processing algorithms.
- *implementation efficiency*: related to silicon area, power and energy. Here, the energy efficiency is the biggest challenge due to the limited available battery power in many devices.
- *flexibility*: in software defined radio, receivers should be flexible/configurable at run-time since they have to support multiple standards. There are various silicon implementation styles ranging from general purpose architectures, over DSPs and ASIPs down to fully physically optimized IP blocks which strongly differ in their implementation efficiency but also in their flexibility. Thus, for each building block of the receiver a detailed analysis of algorithmic flexibility and service parameter flexibility requirements has to be carried out. Service parameter flexibility for a specific channel code is for example varying code rates, block sizes with respect to a given throughput. The goal is to find an optimal implementation style with respect to a flexibility/cost trade-off. Thus, advanced baseband receivers are heterogeneous multi-core architectures implemented in different design styles.

System requirements are very often specified by communication standards such as UMTS, LTE and WiMAX, which define different services in terms of required communications performance and system data throughput, i.e. information bits per second. To obtain an efficient baseband implementation, a careful and elaborate *design space exploration* has to be performed. This is a very challenging task due to the size and the multi-dimensionality of the design space. Therefore it is mandatory to prune the design space in an early stage of the design process. In this process the algorithms have to be selected, optimized, and quantitatively compared to each other with respect to their system performance and implementation efficiency.

Appropriate metrics are key for efficient design space exploration to measure the algorithmic and the implementation complexity, respectively.

A. Algorithmic Complexity

There exists no universal measure for complexity. The huge field of complexity theory and the resulting 'O' notation is mainly focused towards software implementation and cannot be adapted in a straight forward fashion to hardware implementation. The algorithmic complexity, with respect to

Paper approved by G. M. Vitetta, the Editor for Equalization and Fading Channels of the IEEE Communications Society. Manuscript received March 19, 2010; revised October 18, 2010 and July 12, 2011.

This work has been partly supported by the UMIC Research Center, RWTH Aachen University.

F. Kienle and N. Wehn are with the Microelectronic Systems Design Research Group, University of Kaiserslautern, Kaiserslautern, Germany (e-mail: {kienle, wehn} @eit.uni-kl.de).

H. Meyr is with RWTH Aachen University, Aachen, Germany (e-mail: heinrich.meyr@iss.rwth-aachen.de).

Digital Object Identifier 10.1109/TCOMM.2011.092011.100157

hardware implementation, has to be tailored to its application space, in this case the communication system.

Thus, it is important to understand the application under consideration. From a communication system point of view we can separate digital processing in the baseband into two parts: the so called 'inner modem' and 'outer modem' [2], respectively. The task of the inner modem is the extraction of symbols from the received signal waveform, i.e., equalization, channel estimation, interference cancellation and synchronization. The outer modem performs demodulation, de-interleaving and channel decoding on the received symbols. A large diversity exists in the various baseband processing algorithms with respect to operation types, operation complexity, and data types especially between the inner and out modem. Thus, algorithmic complexity in baseband processing should be separately analyzed for the inner and the outer modem, respectively.

Van Berkel determined the complexity of various algorithms in baseband processing. In his remarkable and comprehensive article [1] he focused on the number of 'algorithmic' operations which have to be performed per received bit by the algorithms of a baseband receiver for different communication standards. As a consequence the algorithmic complexity for the inner and outer modem is measured in giga operations per second (GOPs). Using this metric it can be seen that sophisticated decoding schemes like turbo and LDPC codes have a much larger complexity - measured in GOPs - than the algorithms of the inner modem. Eberli from ETH Zürich uses a similar metric [3] for measuring complexity in baseband processing. The importance of appropriate metrics to compare forward error correcting codes is recently shown in [4]. Two improved metrics are suggested, one metric is based on operations the other metric is based on the required storage demands per decoded bit. The major weakness of these metrics is the missing link between both metrics which prevents an application to the resulting implementation complexity.

B. Implementation Complexity

Often implementation complexity is derived from algorithmic complexity in a straight forward manner. E.g. Eberli [3] uses a cost factor for each algorithmic operation which reflects its implementation cost. In this way we can derive area efficiency and energy efficiency which is even more important for baseband receivers.

Graph representations for energy and area efficiency are commonly used for design space exploration:

- A two dimensional energy efficiency graph: one axis corresponds to the algorithmic complexity, e.g. measured in $GOPs$, and the other axis to the power, e.g. measured in mW, consumed when providing the corresponding operations/second. Each point in this graph describes the *energy efficiency metric* of a given implementation, i.e. *operations/second/power unit*, usually measured in $GOPs/mW$. Since energy corresponds to power multiplied with execution time, each point gives the *operations/energy* measured in $operations/Joule$.
- In a similar way we can set up an area efficiency graph in which one axis represents the needed area. Each point in this graph yields the *area efficiency metric*, i.e. *operations/second/area unit*, usually measured in $GOPs/mm^2$.

Note that the energy and area efficiency for the same algorithmic complexity can vary by several orders of magnitude, depending on the selected implementation style. By far the highest energy efficiency is achieved by physically optimized circuits, however, at the expense of limited algorithmic and service parameter flexibility. The highest flexibility, at the expense of low energy efficiency, is achieved by programmable digital signal processors. As said the designer has to find a compromise between the two conflicting goals by trading off flexibility vs. energy efficiency. However, in contrast to area and energy efficiency, flexibility is hard to quantify. The optimum design point has thus to be understood qualitatively. It depends on the application and a large number of economic and technical considerations. We can combine energy and area efficiency in a two dimensional design space in which the two axes correspond to area efficiency and energy efficiency, respectively. This is a well known representation of the design space.

C. Assessment of Metrics

Area, throughput and especially energy in many system-on-chip implementations are dominated by data-transfers and storage schemes [5] and not by the computations themselves. However, common metrics as described above focus solely on operations, and do not consider data-transfer and storage issues at all. Thus, these metrics are only valid if the operations dominate the implementation complexity. This is the case in data-flow dominated algorithms like an FFT calculation, correlation or filtering which are dominating the algorithms in the inner modem.

However, the *channel decoding algorithms* in the outer modem largely differ from the algorithms used in the inner modem. Here, the operations to be performed are non-standard operations (e.g. tanh) using non-standard data types (e.g. 7 bit fix-point). But more importantly, the overall implementation complexity, especially energy, is dominated by data-transfers and storage schemes.

The transitions from 3G to LTE advanced require an improvement of two orders of magnitude in energy efficiency. This improvement will come from technology scaling to a small extent only [6]. A general trend towards co-design of algorithm and architecture can be seen in new standards. Since the traditional separation of algorithm and architecture design leads to suboptimal results.

In channel decoding the co-design focuses on data-transfer and storage schemes [7][8][9]. Examples with respect to communication standards are special interleavers for turbo codes (e.g. LTE standard [10]) and special structures of the parity check matrix for LDPC codes (e.g. DVB-S2 standard [11]). These special structures allow an efficient parallel implementation of the decoding algorithm with small overhead in data-transfer and storage. GOPs based metrics do not at all reflect such specific structures.

Another important issue is flexibility. Flexibility on the service parameter side, e.g., code rates and block sizes in the

TABLE I: Reference decoders: service parameters and implementation results in 65nm technology.

Channel Decoder	**Flexibility**	**Max. Blocksize**	**Throughput [Mbit/s]**	**Frequency [MHz]**	**Area [mm2]**	**Dynamic Power [mW]**
ASIP [12]	CC Binary TC Duo-binary TC	N=16k	40 (64-state) 14 (6 iter) 28 (6 iter)	385 (P&R)	0.7 (P&R)	∼100
LTE TC [13]	R=1/3 to R=9/10 (by puncturing)	N=18k	150 (6.5 iter)	300 (P&R)	2.1 (P&R)	∼300
LDPC flexible	R=1/4 to R=9/10	N=16k	30 (R=1/3 40 iter) 100 (R=1/2 20 iter) 300 (R=5/6 10 iter)	385 (P&R)	1.172 (P&R)	∼389
LDPC WiMedia 1.5 [14]	R=1/2 to R=4/5	N=1.3k	640 (R=1/2 5 iter) 960 (R=3/4 5 iter)	265	0.51	∼193
CC Decoder	64-state NSC		500	500	0.1	∼37

case of channel decoding, have a large impact on the implementation complexity. The cost of the overhead introduced by flexibility is normally not considered by looking only at the operations of the algorithm.

In summary, efficiency metrics based on GOPs only are questionable. Particularly, for non-data flow dominated algorithms, as they entirely neglect important issues like data and storage complexity, algorithm/architecture co-design and flexibility.

In this paper we focus on channel decoding as application. The contributions of this paper are:

- We show that metrics based on GOPs can lead to wrong conclusions.
- We introduce meaningful and suitable metrics for energy and area efficiency.
- We present an approach for design space exploration based on these metrics.

II. Reference Designs

Reference designs are key to assess various metrics. Many publications on VLSI implementations exist for turbo decoders, e.g. [15][16] [17][18][19][20] and LDPC decoders, e.g. [21][22][23][24][25]. However, performing an objective assessment of metrics requires detailed and complete information about the implementation cost and the communication performance. The difficulty of comparing different publications is shown in Section V after deriving suitable metrics for comparison.

For the sake of demonstration, we selected 5 different channel decoder implementations which were designed by our research group in the last couple of years. All test benches, fixed point models, and technology related data are completely available. Every decoder is designed with the same design methodology: hand optimized VHDL code and synthesized using Synopsis Design Compiler. The decoders differ in services (throughput, block sizes, code rates), decoding algorithms, and implementation styles. Selected channel codes are convolutional codes, turbo codes and LDPC codes which covers a large spectrum of channel codes used in todays standards. The 5 different decoders are:

- An application specific instruction set processor (ASIP) [12] capable of processing binary turbo codes, duo-binary turbo codes and various convolutional codes (CC) with varying throughputs dependent on code rate and decoding scheme.
- A turbo decoder, which is LTE [10] compliant. The maximum throughput is 150Mbit/s at 6.5 decoding iterations.
- An LDPC decoder optimized for flexibility, supporting two different decoding algorithms, code rates from R=1/4-9/10 and a maximum block length of 16384.
- An LDPC decoder which is WiMedia 1.5 compliant and optimized for throughput, supporting code rates from R=1/2-4/5 with two block lengths N=1200 and N=1320 bits [14].
- A convolutional decoder with 64-states which is WiFi [26] compliant.

All decoders are synthesized on a 65nm CMOS technology under worst case conditions with $V_{dd} = 1.0V, 120^{\circ}C$. Power estimations are based on nominal case $V_{dd} = 1.1V$. Table I gives an overview of the key parameters of the different decoders. P&R indicates that the corresponding data are post-layout data. The payload (information bits) throughput depends on the number of decoding iterations for turbo and LDPC codes which also impacts the communications performance. Thus, the throughput is specified depending on the number of iterations. We chose our own designs as reference designs, as these were the only designs for which we had complete access to all information, which is necessary for a fair assessment of metrics. This does not limit the validity of the statements and conclusions made in this paper.

In Table II we show the number of algorithmic operations required to process the different types of convolutional codes, turbo codes and LDPC codes. Bit-true C-reference models are used for operation counting. In all algorithms of Table II the operations are fixed point additions with varying bit width. We normalized all operations to an 8 bit addition. The number of operations is related to information bits which have to be decoded. The total number of operations, which have to be performed per second, depends on the code rate R and throughput, which in turn depends on the number of iterations for LDPC and turbo code decoding. Two different

TABLE II: Number of normalized algorithmic operations per decoded information bit for different channel decoders dependent on throughput and code rate R.

Channel Code	# iterations	GOPs (w.r.t. throughput) 100 Mbit/s	300 Mbit/s	1 Gbit/s
CC (64-state)		20	60	200
LDPC Min-Sum	10 40	15/R 60/R	45/R 180/R	150/R 600/R
LDPC λ-3-Min	10 40	50/R 200/R	150/R 600/R	500/R 2000/R
3GPP Turbo (Max-Log)	2 6	28 84	84 252	280 840

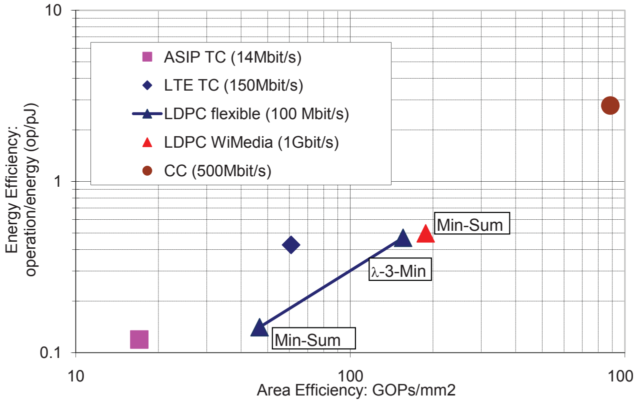


Fig. 1: Energy efficiency versus area efficiency based on operations

algorithms for LDPC codes are utilized in our LDPC decoders. Both algorithms are suboptimal algorithm approximating the belief propagation algorithm: the Min-Sum algorithm with a scaling factor and the λ-3-Min algorithm [27] which is a more accurate approximation. However, the latter needs about 3.3 times more operations. This more accurate approximation is required when lower code rates $R < 0.5$ have to be supported, as for example in DVB-S2 decoders [23]. The flexible LDPC decoder supports both decoding algorithms, the WiMedia LDPC decoder is based on the Min-Sum algorithm only.

III. Suitable Metrics

Figure 1 shows the two dimensional design space for our decoders, covering area and energy efficiency based on operations. One axis represents the area efficiency ($GOPs/mm^2$), and the other axis the energy efficiency (op/pJ). In this graph, efficient architectures w.r.t. area and energy are located in the upper right corner. Less efficient architectures are placed in the lower left corner.

The convolutional decoder seems to be the most efficient decoder while the ASIP shows to be the decoder with the lowest efficiency. An interesting observation is the efficiency of the flexible LDPC decoder. The efficiency largely increases in both directions (area, energy) when replacing the Min-Sum by the λ-3-Min algorithms which is the more complex algorithm in terms of operations. As described in the previous section the GOPs for this algorithm increases by a factor 3.3.

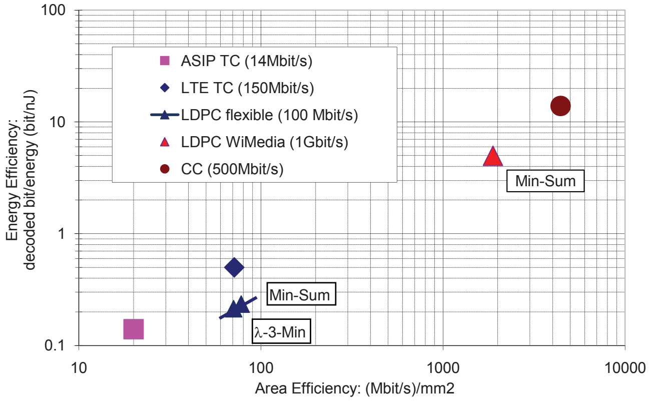


Fig. 2: Design space based on suitable metrics. Decoded information bit per energy over information bit throughput per area unit.

Furthermore, we see that the λ-3-Min based flexible LDPC decoder has nearly the same efficiency as the less flexible WiMedia decoder which is optimized for high throughputs. This observation seems to be contradictory, we would expect that a less flexible decoder, optimized for throughput, has a much higher efficiency compared to a highly flexible decoder architecture. This aspect shows already the problem with GOPs based metrics.

The problem with GOPs based metrics will become more obvious after introducing appropriate metrics which resolve the aforementioned anomaly. Instead of referring to operations per task we refer to the number of decoded information bits. Metrics normalized to the number of information bits allow to compare competing architectures for a given algorithm since the efficiency metrics are independent of the specific operations and data types used to execute the algorithm. All implementation issues like data-transfer and storage are taken into account since the metrics are oblivious of how the task has been executed. Such a metric considers the result only and not how the result is calculated. Furthermore, a metric normalized to the number of information bits allows to compare different coding schemes as a function of the communication parameters (modulation, signal to noise ratio, bandwidth). In particular, iterative decoding algorithms can be compared in a meaningful way to non-iterative algorithms.

We define two metrics for implementation efficiency as follows

- energy efficiency metric: *decoded information bit per energy* measured in bit/nJ
- area efficiency metric: *information bit throughput per area unit* measured in $Mbit/s/mm^2$

We mapped our decoders to the design space which is based on these metrics. The result is shown in Figure 2. Again efficient architectures are in the upper right corners, inefficient architectures are in the lower left corner.

A large change in the relative and absolute positions can be observed for some decoders, when comparing the graph to Figure 1 which was based on GOPs metrics.

- The difference in the efficiency between the two instances of the flexible LDPC decoder (Min-Sum and the λ-3-Min decoder, respectively) is now much smaller. Moreover the Min-Sum decoder is more efficient than the other ones which was not the case in the conventional design space. This matches our expectations since the data-transfer and storage scheme in both decoders is nearly identical. The increase in computation results in only a small energy and area increase by about 10%. Both flexible LDPC decoders are targeting the same throughput.

- The efficiency of the WiMedia decoder which is optimized rather for throughput than for flexibility, is now much larger than the efficiency of the flexible LDPC decoder which again matches our expectations.

To summarize, in applications which are dominated by data-transfers and storage schemes, like here in channel coding, the change in the number of operations in the algorithm has only a small impact on area and energy. Comparing Figure 1 and Figure 2 demonstrates clearly the problem with GOPs based metrics.

So far we have focused on the implementation complexity but have not discussed the important aspect of flexibility and communication performance. In the following we will investigate the relationship between communication performance, algorithmic/service parameter flexibility and implementation efficiency.

IV. Approach for Design Space Exploration

In the previous section we investigated the absolute and the relative positions of the different decoders to each other. However, equally important in this space is the shift of an efficiency point when specific communication parameters are changed.

The following relevant parameters will be considered: the frame error rates (FER), i.e. communications performance, coding techniques, code rates, number of iterations and throughput.

We present two case studies to demonstrate an appropriate approach for design space exploration. The resulting different values for area and energy efficiencies illustrate the strong dependency between communications performance and implementation efficiency.

The two case studies will show that a single quantitative measure to select the best channel code does not exist. The first case study is concerned with implementation efficiency and compares non-iterative decoding techniques (convolutional codes) to iterative decoding techniques (LDPC codes). The second case study compares two different iterative decoding techniques (LDPC and Turbo codes) with code rate flexibility.

A. Implementation driven design space exploration

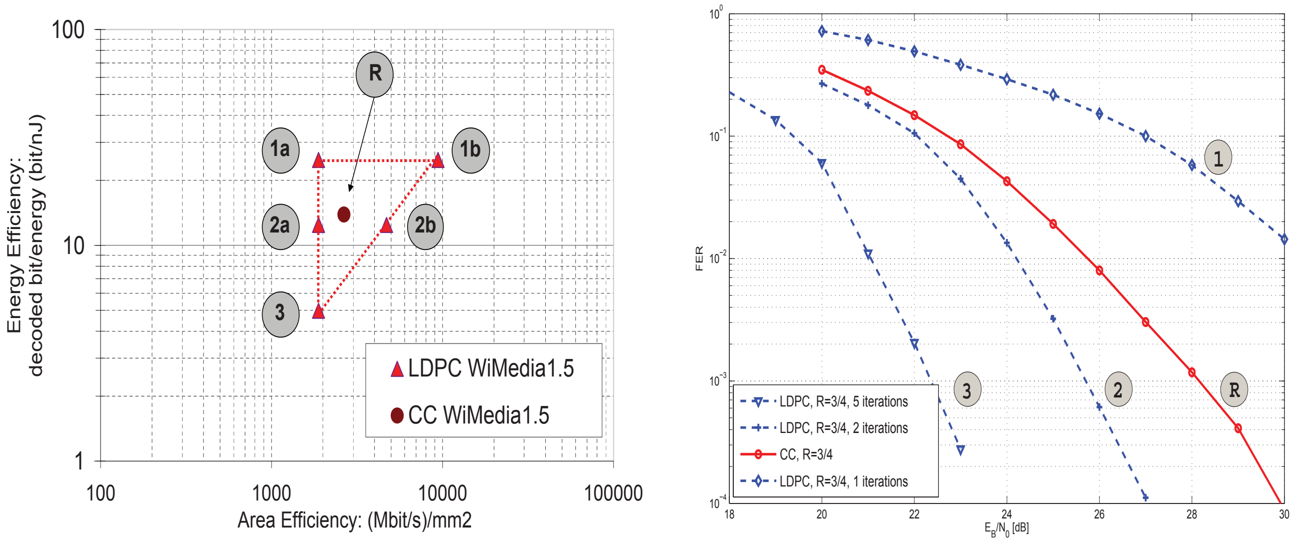
The first design study reflects the discussion performed during the WiMedia 1.5 standardization process. WiMedia [28] features low complexity devices for UWB communication. The older WiMedia 1.2 standard uses convolutional codes as channel coding technique which served as a reference design. LDPC codes were considered as a promising candidate which had to be compliant to the given service parameters like the throughput of 960 Mbit/s and the code rate $R = 0.75$. Thus, new LDPC codes were developed according to the code/architecture co-design approach [14] resulting in an LDPC decoder which has a much higher efficiency than e.g. a highly flexible LDPC decoder.

As already shown in Figure 2 the area and energy efficiency of the resulting WiMedia 1.5 LDPC decoder is lower compared to a convolutional decoder. However, this comparison does not at all consider the communications performance. At most five decoding iterations can be performed by the LDPC decoder to comply with the throughput requirements of the standard. As already pointed out, the number of iterations strongly impacts the communications performance of the LDPC decoder. The frame error rates as a function of the number of iterations are contrasted with implementation efficiency in Figure 3. Point 3 in the design space figure corresponds to the WiMedia 1.5 decoder when performing 5 iterations (this was the decoder assumption in the previous figures when we referred to the WiMedia LDPC decoder). The communication figure shows that this decoder has a 4dB better communication performance than the convolutional decoder. The communication performance is comparable to that of the convolutional decoder if the LDPC performs only two iterations instead of five (case 2 in Figure 3). Finally, executing only one iteration in the LDPC decoder results in a communication performance which is about 4dB worse than the convolutional decoder (case 1 in Figure 3).

Note the resulting set of points in the design space for the different cases. The different efficiency points in Figure 3 are obtained by distinguishing two cases:

- The system throughput is not changed w.r.t. WiMedia 1.5. constraint (scenario a in Figure 3). In this scenario only the energy efficiency is improved (points $3 \rightarrow 2a \rightarrow 1a$). Obviously the decoding time decreases when the decoder executes a smaller number of iterations resulting in a negative time lag. This time lag can be exploited for energy efficiency improvement. For example clock and power supply could be completely switched off when decoding is finished. This reduces energy and leakage current. Another possibility is to slow down the frequency (frequency scaling). This reduces the energy by the same amount as in the previous case but the peak power consumption during decoding instead of leakage is minimized. The most efficient technique is voltage scaling in which the voltage is reduced which results in the highest energy efficiency.
- The system throughput is changed (scenario b in Figure 3). In this scenario the area efficiency increases by the same amount as the throughput increases due to smaller number of iterations (points $3 \rightarrow 2b \rightarrow 1b$).

We see that in terms of efficiency the LDPC decoder is increasing with decreasing communication requirements, i.e. number of iterations. Thus, the *decoder efficiency is represented by various efficiency evaluations instead of a single point* in the design space. This varying set of points results from varying communication performance requirements. We also see that the efficiency of the LDPC decoder outperforms the convolutional decoder at the same communication performance.



(a) Implementation Efficiency

(b) Communications performance

R :	Reference convolutional decoder with code rate	($R = 0.75$) and fixed throughput of 1 Gbit/s
1 :	LDPC code performing 1 iteration	
a)	identical throughput (1 Gbit/s)	~ 4dB worse communications performance
b)	5 times higher throughput (5 Gbit/s)	~ 4dB worse communications performance
2 :	LDPC code performing 2 iteration	
a)	identical throughput (1 Gbit/s)	~ 1dB better communications performance
b)	2.5 times higher throughput (2.5 Gbit/s)	~ 1dB better communications performance
3 :	LDPC code performing 5 iteration	
	identical throughput (1 Gbit/s)	~ 4dB better communications performance

Fig. 3: Implementation efficiency and communications performance for WiMedia 1.5 standard (16-QAM, CM1 channel according to IEEE 802.15.3a [14])

B. Communications performance driven exploration

In the second design study we compare two iterative decoding techniques and put emphasis on the resulting communications performance. An LTE turbo decoder is compared with a flexible LDPC decoder which supports code rate flexibility. The right graph in Figure 4 shows the communication performance for the two decoding schemes for different code rates ($R = 0.5$ and $R = 0.83$) and iteration numbers. The number of information bits is $K = 6140$ in all cases. Frame error rates are based on fixed point simulations matching the hardware implementation. We use the communications performance of the turbo decoder with 6.5 iterations as reference point for both code rates. The 6.5 iterations result from the throughput constraint of 150Mbit/s which is specified in the LTE standard. The 6.5 iterations fulfill the LTE communications performance requirements for all code rates.

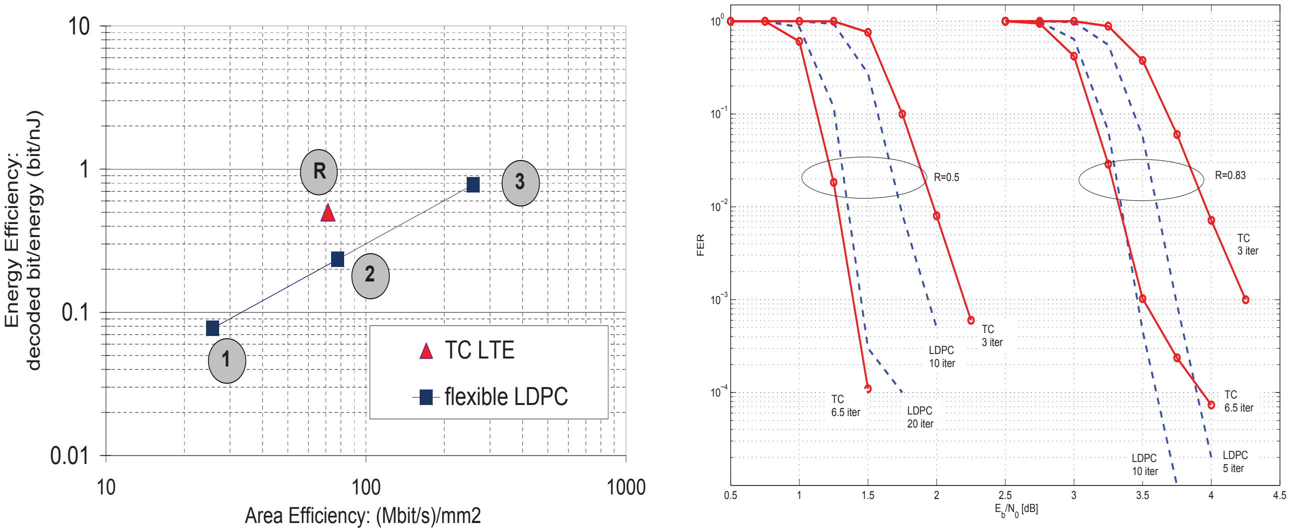
It is well known that the communication performance in LDPC decoding depends on the number of iterations *and* the code rate. The LDPC decoder under investigations provide large code rate flexibility, i.e., the hardware can support various code rates. The LDPC decoder requires 10 iterations for $R = 0.83$ and 20 iterations for $R = 0.5$ to match the performance of the turbo decoder. For a code rate of $R = 1/3$ even 40 iterations are mandatory (this is not shown in Figure 4b).

The corresponding results in the implementation space are noteworthy. The turbo decoder efficiency is identical for all code rates (see left graph in Figure 4). Thus, we have only one fixed efficiency point. This is due to the fact that the code rate flexibility is implemented by puncturing which has negligible impact on throughput, area and energy.

However, the situations is completely different for the flexible LDPC decoder. For a given communications performance the code rate has strong impact on the number of required iterations. This iteration number influences the implementations efficiency as we have seen in the previous case study. But beside this impact of the iteration number, there is also a direct impact of the code rate on the implementation efficiency since lower code rates require also a more accurate decoding algorithm (λ-3-Min algorithm instead of the less complex Min-Sum algorithm). The resulting efficiency points are shown in the left graph of Figure 4. We see that the efficiency increases in both directions with increasing code rate (points $1 \rightarrow 2 \rightarrow 3$).

The important observation in this exploration is the varying implementation efficiency of the flexible LDPC decoder. The different efficiency evaluations result from the required code rate flexibility in the LDPC decoder which is necessary to match the communications performance with respect to a competitive turbo code decoder. We see that analyzing only one code rate, and thus one snap shot, could result in a wrong efficiency conclusions.

The two explorations have shown that implementation efficiency for iterative decoders often results in many points instead of a single point in the design space. The different evaluations result from the strong interrelation between



(a) Implementation Efficiency

(b) Communications performance

R :	Reference of LTE turbo decoder	
	150 Mbit/s throughput for all code rates	reference TC performance at 6.5 iterations
1 :	LDPC code at code rate $R = 1/3$	
	$\sim$ max throughput 30Mbit/s	$\sim$ 40 iterations to match TC performance
2 :	LDPC code at code rate $R = 1/2$	
	$\sim$ max throughput 90Mbit/s	$\sim$ 20 iterations to match TC performance
3 :	LDPC code at code rate $R = 0.83$	
	$\sim$ max throughput 300Mbit/s	$\sim$ 10 iterations to match TC performance

Fig. 4: Implementation efficiency and communications performance of LTE turbo code/decoder and a flexible LDPC decoder (BPSK, AWGN channel).

communication performance, flexibility and implementation efficiency.

V. Comparison of Published Decoder Architectures

Comparing different decoder architectures objectively, even given proper metrics, is a very challenging task. This is due to the fact that published decoders

- are based on different technology nodes (e.g. 180nm, 130nm, 65nm) and characterization assumptions (e.g. worst case, nominal case or best case operating conditions) and
- differ in their communications performance with respect to the chosen architectural parameters.

To highlight the problem of a fair comparison, we list selected state-of-the-art turbo decoder architectures in Table III. This table shows the implementation parameters of the various decoders, i.e. frequency, area, throughput, technology node, but also parameters which have a strong impact on the communications performance. First, we will discuss technology parameters that influence the achievable throughput followed by architectural parameters affecting the communications performance.

A. *Technology Parameters*

Different technologies have different area, timing (frequency) and energy characteristics. In principle, it is possible to normalize the area from one technology node (e.g. 130nm) analytically to a reference technology node (e.g, 65nm) [29], but scaling timing and energy is very difficult for advanced technology nodes. Due to space limitations, we focus here on only some aspects of technology scaling.

In larger technologies, e.g. a 130nm process, the critical path is normally found in the logic. In turbo decoders, for example, it is typically determined by the add-compare-select unit in the maximum a posteriori (MAP) processing kernel. In advanced technologies of 65nm and less, memories are becoming more and more the limiting factor in terms of timing and energy [29]. Today, we can see large variations in timing and energy efficiency of memories even in the same technology node, depending on the technology provider and the final memory instantiation [30]. Consequently, the assumption that memories scale as good as the logic, when scaling published timing and energy to reference node, is not true [31][29].

The assumptions under which published frequency and energy data were obtained are another important issue. Timing and energy characterization can be performed under nominal, worst and best case operating conditions. These conditions are related to voltage supply, temperature and technology variations. The achievable frequency can vary up to a factor of two between worst case and best case conditions. E.g., under worst case conditions the decoder presented in [13] has a maximum frequency of $f_{cyc} = 300$MHz, but achieves $f_{cyc} = 450$MHz under nominal case conditions. In Table III we list the operating conditions, when specified in the publication. Measurements of fabricated chips are often performed under nominal or even best case conditions, whereas some of the listed decoders assume worst case conditions. Depending on the assumed operating conditions, we see large differences in achievable frequencies, which in turn results in large variations in the final throughput, see Section VII.

B. Communications Performance

The communication performance is mainly affected by three parameters, which also have a strong impact on throughput, area, and energy: input quantization (Q), window length (WL), and acquisition length (AL). Efficient decoder architectures split the input block into several so-called windows, to reduce the memory requirements and to permit parallel processing. To counteract the consequent degradation of the communications performance, an acquisition step is introduced at the window borders. Of course, the input quantization has a large and direct effect on the communications performance. The quantization of data internal to the turbo decoder can be derived from the input quantization. Typically, a 5 bit input quantization shows a 0.2dB performance degradation compared to a 6 bit input quantization, assuming the same number of iterations and a code rate of $R = 1/3$. Most of the published turbo decoder architectures are instantiated with 5 or 6 bits for input quantization.

To highlight the impact of window and acquisition length on the communications performance, we compare the two turbo decoders [32] and [13]. They share the same quantization, technology, and basic decoder architecture, but strongly differ in the window length and acquisition length (*WL*=8, *AL*=20 compared to *WL*=32, *AL*=96). For a code rate $R = 1/3$ and a given number of iterations, the two decoders show a nearly identical communications performance. However, for $R = 0.95$, the highest code rate specified in the LTE standard, the difference in the communications performance after 6 iterations is about 2dB. The decoder from [32] achieves the desired communications performance only after 9 iterations, at a much lower throughput, which is first order inverse proportional to the number of iterations, see Section VII.

It is possible to derive some first order analytical equations showing the impact of the parameters (*WL*, *AL*) on throughput and area (Section VII), but there is no analytical equation for the dependency of the communications performance on these parameters. Instead Monte Carlo simulations have to be performed, making it nearly impossible to compare the decoders in Table III from a communications performance point of view, unless the publications give detailed communications performance graphs for the relevant parameters of the given standards. Although sometimes such graphs are published, they typically cover only a subset of a standard's parameters and do not cover their corner cases. E.g., the turbo decoder from [19] is LTE compliant with respect to the supported interleavers and code rates, but it is not clear how many iterations are required to fulfill the LTE communications performance requirements at the code rate of $R = 0.95$.

These examples show that comparing only architectural results of different published decoders may lead to entirely wrong conclusions. The communications performance results have to be published as well, not for one point but for various service parameters, to allow a comparison of implementation efficiencies.

VI. Conclusion

Understanding the trade-offs between implementation efficiency, communications performance and flexibility will be key for designing efficient baseband receivers. Meaningful efficiency metrics are mandatory to explore and evaluate the resulting huge design space. We introduced and discussed suitable energy and area efficiency metrics which are based on decoded information bit per energy and throughput per area unit. Various channel decoder implementations were utilized to examine these efficiency metrics with respect to the achieved communications performance and with respect to the decoder flexibility. The presented approaches allow to systematically compare different realizations by jointly considering: implementation efficiency, communications performance and flexibility.

VII. Appendix

In this section we show the dependencies of various architectural parameters on throughput and area of state-of-the art turbo decoders. We use the following nomenclature and parameters:

t_{cond}	technology node and operating conditions w.r.t. feature size variation, V_{dd}, temperature
f_{cyc}	frequency of the design
Q	quantization of the input data, the quantization of all other variables are derived from this
$iter$	iteration number
K	number of information bits
WL	window length, in hardware the codeword is processed window by window
AL	acquisition length, the number of training steps for the forward or backward recursion.
rdx	radix-2 or radix-4 realization of the recursion units
P	architectural parallelism
C_N	latency due to network to realize the interleaving, strongly depends on P and interleaver structure

The throughput (T) for state-of-the-art turbo decoder architectures can be calculated by the frequency times the number of cycle needed to process an information word of length K. An increased throughput requires a higher parallelism of the architecture, which increases the number of overhead cycles for interleaving $C_N(P)$.The throughput is given as follows:

$$T = f_{cyc} \cdot \frac{K}{2 \cdot iter \cdot \left(\frac{WL+AL}{log_2(rdx)} + \frac{K}{P \cdot log_2(rdx)} + C_N(P) \right)} \quad (1)$$

The frequency itself mainly depends on technology t_{cond}, the quantization of the input data, and the critical path in the combinatorial logic.

The area A_{all} of a turbo decoder is composed of three parts:

$$\begin{aligned} A_{all} = {} & P \cdot A_{MAP}(t_{cond}, Q, WL, AL, rdx) \\ & + A_{ctrl}(t_{cond}) \\ & + A_M(t_{cond}, Q, WL, AL, rdx, P) \end{aligned} \quad (2)$$

TABLE III: Published 3GPP TC decoder architectures with corresponding architectural and technology parameters. The communications performance depends mainly on: input quantization (Q), window length (WL), and acquisition length (AL).

Work	Standard(s)	Results from	Throughput [Mbit/s]	@ iter.	@ f_{cyc} [MHz]	Technology	Area [mm^2]	Parameters strongly affecting communications performance
[15]	HSPA	fabricated chip	24	6	145	180 nm	14.5	Q=5, WL=40, AL=40
[16]	HSPA	fabricated chip	20.2	5.5	246	130 nm	1.2	Q=5, WL=40, AL=40
[17]	HSPA	worst case synthesis	758	6	256	130 nm	13.1	Q=5, WL=32, AL=32
[18]	LTE, WiMAX	fabricated chip	186	8	250	130 nm	10.7	Q=?, WL=128, AL=128
[19]	LTE	fabricated chip	390	5.5	302	130 nm	3.57	Q=5, WL=30, AL=30
[20]	LTE	place & route	1280	6	400	65 nm	8.3	Q=6 , WL=64, AL=64
[32]	HSPA	worst case synthesis	184	6	300	130 nm	3.8	Q=6, WL=8, AL=20
[13]	LTE	fabricated chip	150	6.5	300	65 nm	2.1	Q=6, WL=32, AL=96

A_{MAP} is the logic area for a single MAP processing kernel which has to be instantiated P times depending on the parallelism. A_{ctrl} is the area of the controller which is typically small compared to the other two parts. A_M is the required area for instantiated memories. The area to store the input data ($\mathrm{A}_M^{I/O}$), the area of extrinsic data (A_M^{extr}) which are exchanged between component decoders, and the area to store state metric values used within the processing units (A_M^{MAP}). Thus the area of the memory which is a large portion of the overall area is given by

$$\begin{aligned} A_M &= P \cdot \mathrm{A}_M^{MAP}(t_{cond}, WL, Q, rdx) \\ &\quad + \mathrm{A}_M^{I/O}(t_{cond}, P, K, Q) \\ &\quad + \mathrm{A}_M^{extr}(t_{cond}, P, K, Q) \end{aligned} \tag{3}$$

In a similar way we can derive equations for the energy consumption. However, these results are omitted due to space limitations.

References

[1] C. H. van Berkel, “Multi-core for mobile phones,” in *Proc. Design, Automation. Test Europe Conf. Exhibition*, Apr. 2009, pp. 1260–1265.

[2] H. Meyr, M. Moeneclaey, and S. A. Fechtel, *Digital Communication Receivers*. John Wiley & Sons Inc., 1998.

[3] S. C. Eberli, Ph.D. dissertation, ETH Zurich, Integrated Systems Laboratory, 2009.

[4] S. Galli, “On the fair comparison of FEC schemes,” in *Proc. IEEE Int. Conf. Commun.*, 2010, pp. 1–6.

[5] M. Miranda, C. Ghez, E. Brockmeyer, P. Op De Beeck, and F. Catthoor, “Data transfer and storage exploration for real-time implementation of a digital audio broadcast receiver on a Trimedia processor,” in *Proc. 15th Symp. Integrated Circuits Syst. Design*, Sep. 2002, pp. 373–378.

[6] C. Rowen, “Energy-efficient LTE baseband with extensible dataplane processor units,” in *9th International Forum Embedded MPSoC Multicore*, Aug. 2009.

[7] E. Boutillon, J. Castura, and F. Kschischang, “Decoder-first code design,” in *Proc. 2nd International Symp. Turbo Codes Related Topics*, Sep. 2000, pp. 459–462.

[8] M. Mansour and N. Shanbhag, “Architecture-aware low-density parity-check codes,” in *Proc. IEEE International Symp. Circuits Syst.*, May 2003.

[9] J. Kwak and K. Lee, “Design of dividable interleaver for parallel decoding in turbo codes,” *Electron. Lett.*, vol. 38, no. 22, pp. 1362–1364, Oct. 2002.

[10] “3GPP LTE (Long Term Evolution) Homepage.” Available: http://www.3gpp.org/Highlights/LTE/LTE.htm

[11] European Telecommunications Standards Institude (ETSI), “Digital Video Broadcasting (DVB) Second generation framing structure,channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications; TM 2860r1 DVBS2-74r8.” Available: www.dvb.org.

[12] T. Vogt and N. Wehn, “A reconfigurable ASIP for convolutional and turbo decoding in a SDR environment,” *IEEE Trans. Very Large Scale Integration (VLSI) Syst.*, vol. 16, pp. 1309–1320, Oct. 2008.

[13] M. May, T. Ilnseher, N. Wehn, and W. Raab, “A 150Mbit/s 3GPP LTE turbo code decoder,” in *Proc. Design, Automation Test Europe*, Mar. 2010, pp. 1420–1425.

[14] M. Alles, F. Berens, and N. Wehn, “A synthesizable IP core for WiMedia 1.5 UWB LDPC code decoding,” in *Proc. IEEE International Conf. Ultra-Wideband*, Sep. 2009, pp. 597–601.

[15] M. Bickerstaff, L. Davis, C. Thomas, D. Garrett, and C. Nicol, “A 24Mb/s Radix-4 LogMAP turbo decoder for 3GPP-HSDPA mobile wireless,” in *Proc. IEEE International Solid-State Circuits Conf.*, Feb. 2003, pp. 150–151.

[16] C. Benkeser, A. Burg, T. Cupaiuolo, and Q. Huang, “Design and optimization of an HSDPA turbo decoder ASIC,” *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 98–106, Jan. 2009.

[17] G. Prescher, T. Gemmeke, and T. Noll, “A parameterizable low-power high-throughput turbo-decode,” in *Proc. IEEE International Conf. Acoustics, Speech, Signal Process.*, Mar. 2005, pp. V-25-28.

[18] J.-H. Kim and I.-C. Park, “A unified parallel radix-4 turbo decoder for mobile WiMAX and 3GPP-LTE,” in *Proc. IEEE Custom Integrated Circuits Conf.*, Sep. 2009, pp. 487–490.

[19] C. Studer, C. Benkeser, S. Belfanti, and Q. Huang, “A 390Mb/s 3.57mm^2 3GPP-LTE turbo decoder ASIC in 0.13μm CMOS,” in *Proc. IEEE International Solid-State Circuits Conf. - Digest Tech. Papers*, vol. 53, Feb. 2010, pp. 274–275.

[20] Y. Sun and J. Cavallaro, “Efficient hardware implementation of a highly-parallel 3GPP LTE/LTE-advance turbo decoder,” *Integration VLSI J.*, 2010.

[21] P. Urard, L. Paumier, V. Heinrich, N. Raina, and N. Chawla, “A 360mW 105Mb/s DVB-S2 compliant codec based on 64800b LDPC and BCH codes enabling satellite-transmission portable devices,” in *Proc. IEEE International Solid-State Circuits Conf.*, Feb. 2008, pp. 310–311.

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

[23] S. Muller, M. Schreger, M. Kabutz, M. Alles, F. Kienle, and N. Wehn, “A novel LDPC decoder for DVB-S2 IP,” in *Proc. Design, Automation. Test Europe Conf. Exhibition*, Apr. 2009, pp. 1308–1313.

[24] M. Mansour and N. Shanbhag, “Low-power VLSI decoder architectures for LDPC codes,” in *Proc. International Symp. Low Power Electron. Design*, Aug. 2002.

[25] J. Dielissen, A. Hekstra, and V. Berg, “Low cost LDPC decoder for DVB-S2,” in *Proc. Design, Automation Test Europe*, Mar. 2006.

[26] IEEE 802.11, "Wireless fidelity (Wireless LAN)." Available: http://grouper.ieee.org/groups/802/11/.

[27] F. Guilloud, E. Boutillon, and J. Danger, "λ-Min decoding algorithm of regular and irregular LDPC codes," in *Proc. 3rd International Symp. Turbo Codes Related Topics*, Sep. 2003, pp. 451–454.

[28] "WiMedia alliance." Available: www.wimedia.org.

[29] International Technology Roadmap for Semiconductors, "ITRS home page." Available: http://public.itrs.net.

[30] CMP, "Circuits Multi-Projets." Available: http://cmp.imag.fr/products/ic/

[31] J. M. Rabaey, *Low Power Design Essentials*, 1st edition. Springer, 2009.

[32] M. May, C. Neeb, and N. Wehn, "Evaluation of high throughput turbo-decoder architectures," in *Proc. IEEE International Symp. Circuits Syst.*, May 2007, pp. 2770–2773.

Dr. Frank Kienle received the M.Sc in Communications from the Technical University of Kaiserslautern in 2001. He joint to the Microelectronic Systems Design Research Group and received his Ph.D. in 2006.

From 2006-2008 he had a Post-Doc position at the Microelectronic Systems Design Research Group responsible for industry cooperation's. Since 2009, he holds the position 'Akademischer Rat' and is heading the communications research group in the institute. He has more than 40 publications in various fields.

Dr. Norbert Wehn holds the chair for Microelectronic System Design in the department of Electrical Engineering and Information Technology at the University of Kaiserslautern. He has more than 200 publications in various fields of microelectronic system design and holds several patents. He is chairman of the European Design Automation Association, Chairman of the Research Center "Ambient Systems" at TU Kaiserslautern, associate editor of various journals and member of several scientific advisory boards. In 2003 he served as program chair for DATE 2003 and as general chair for DATE 2005 respectively. His special research interests are VLSI-architectures for mobile communication, forward error correction techniques, low-power, advanced SoC architectures and reliability issues in SoC. Norbert Wehn is IEEE Senior and IEEE Golden Core member.

Dr. Heinrich Meyr received his M.Sc. and Ph.D. from ETH Zurich, Switzerland. He spent over 12 years in various research and management positions in industry before accepting a professorship in Electrical Engineering at Aachen University of Technology (RWTH Aachen). He has retired as Professor Emeritus in 2007 but remains active as a researcher. Presently he is a visiting Professor at EPFL, Switzerland.

During the last thirty years, Dr. Meyr has worked extensively in the areas of communication theory, digital signal processing and CAD tools for system-level design. His academic work is highly cited, he received three IEEE best paper awards, and in 2000 he received the prestigious Vodafon innovation prize. Dr. Meyr is a Life Fellow of the IEEE

At RWTH Aachen he implemented an innovative cooperation model between university and industry with a highly successful fundraising program, without compromising academic excellence. Major industry funding included grants from leading US and European companies.

Dr .Meyr has a dual career as entrepreneur. He has founded several successful companies. The last company he cofounded in 2001 was LisaTek. LisaTek has developed revolutionary technology to design MPSoC. Since 2010 Lisatek is part of Synopsys, the market leader in EDA.