Advanced Wireless Digital Baseband Signal Processing Beyond 100 Gbit/s

Stefan Weithoffer, Matthias Herrmann, Claus Kestel, Norbert Wehn

Advanced Wireless Digital Baseband Signal Processing Beyond 100 Gbit/s

Stefan Weithoffer, Matthias Herrmann, Claus Kestel, Norbert Wehn
Microelectronic Systems Design Research Group, University of Kaiserslautern
67663 Kaiserslautern, Germany
{weithoffer, herrmann, kestel, wehn}@eit.uni-kl.de

***Abstract*—The continuing trend towards higher data rates in wireless communication systems will, in addition to a higher spectral efficiency and lowest signal processing latencies, lead to throughput requirements for the digital baseband signal processing beyond 100 Gbit/s, which is at least one order of magnitude higher than the tens of Gbit/s targeted in the 5G standardization. At the same time, advances in silicon technology due to shrinking feature sizes and increased performance parameters alone won't provide the necessary gain, especially in energy efficiency for wireless transceivers, which have tightly constrained power and energy budgets.**

In this paper, we highlight the challenges for wireless digital baseband signal processing beyond 100 Gbit/s and the limitations of today's architectures. Our focus lies on the channel decoding and MIMO detection, which are major sources of complexity in digital baseband signal processing. We discuss techniques on algorithmic and architectural level, which aim to close this gap. For the first time we show Turbo-Code decoding techniques towards 100 Gbit/s and a complete MIMO receiver beyond 100 Gbit/s in 28 nm technology.

I. Introduction

The first generations of mobile communication systems have shifted the communication from landline to handheld devices, followed by the third (3G) and fourth (4G) generations, which marked the advent of the mobile internet. The amount of devices, which will mostly be wirelessly connected, is expected to increase to 50 billion in the year 2020.

While the fifth generation (5G) of mobile communications systems is being standardized by the 3GPP, the monthly data traffic per smartphone is estimated to increase to 18 GB per month in 2021 [19]. The rapid increase in both data volume and data rate (or throughput) is a continuation of the evolution from 3G to 4G networks, and the downlink data rate, for example, will be at least 20 times higher compared to 4G LTE-A [30].

Beyond 5G, which will have data rates in the order of tens of Gbit/s, the throughput requirements for mobile communication systems will be higher than 100 Gbit/s. Figure 1 illustrates this evolution of the peak downlink data rate in the context of the 3GPP mobile communication standards. The main contributers to this trend are applications like streaming video, which make up over 50% of the annual mobile traffic increase.

At the same time, the latency is becoming more and more constrained. This is evident from Figure 2, which illustrates the trend towards roundtrip latencies in the order of μs.

978-1-5386-0446-5/17/$31.00 © 2017 IEEE

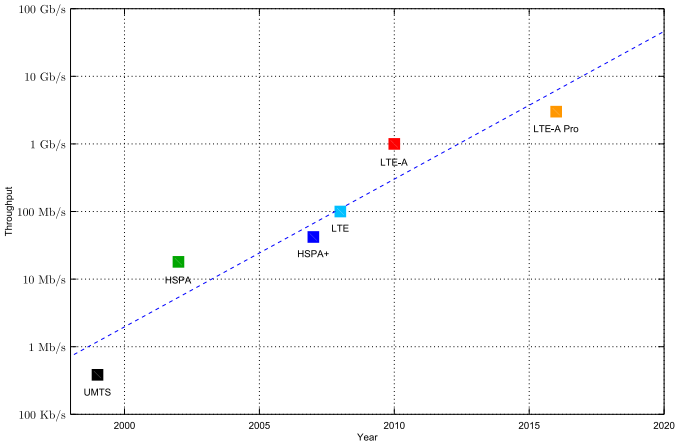

Fig. 1. Throughput Evolution in 3GPP Mobile Communication Standards

Accordingly, the evolution of the length of the *Transmission Time Intervalls* (TTIs), in which the transmission of a data block is organized on the physical layer, shows the same trend. The shrinking latencies are expected to enable a range of applications, which have been described by the term Tactile Internet, i.e. remote driving, free-viewpoint video, machine-to-machine or smart grid synchronization [8].

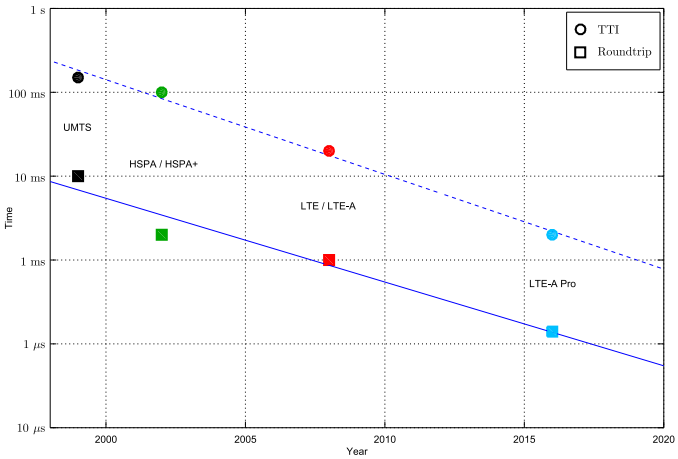

Fig. 2. Roundtrip Latency and TTI Length Evolution in 3GPP Mobile Communication Standards

To satisfy this ever increasing demand for higher data rates

many wireless communication standards, e.g. WiFi, LTE or WiMAX, adopted multiple input multiple output (MIMO) techniques. This technological innovation has vastly improved the spectral efficiency and thus been a key enabler for today's data rates of several Gbit/s. To achieve even higher data rates, higher frequency ranges that offer more bandwidth, must be exploited. A recent example is the WiGig standard (IEEE 802.11ad) that already utilizes carrier frequencies in the 60 GHz band to achieve a maximum throughput of almost 7 Gbit/s. However, utilizing MIMO techniques in millimeter wave systems is still a great challenge and currently a very active research field (e.g. [10]) and a promising approach to achieve data rates of 100 Gbit/s and more.

Transceivers to support such high data rates must be extremely power/energy efficient. The available power budget is typically limited to some Watts for digital baseband processing due to thermal power density contraints. E.g., if we target a throughput of 100 Gbit/s in a 1 W power envelope, only 10 pJ energy is available to process a single bit [14]. For comparison, in 28 nm technology, a 64 bit double precision floating point operation consumes about 20 pJ and transferring 256 bits over a 40 mm wire on a chip costs about 10 nJ. For the future, progress in microelectronics will yield some improvements in the three important metrics, i.e. area, energy, and frequency. Extrapolating down from 28 nm to 7 nm technology will bring approximately a factor 12x reduction in area, a factor 4x in energy efficiency, and a factor 3x in (maximum operating) frequency [26] [12]. Due to the fact that area density increases faster than the improvement in energy efficiency, power density will be one of he biggest challenges and is already a big issue in today's technologies and known as "dark silicon phenomenon" [22]. Thus, for digital baseband signal processing beyond 100 Gbit/s, the improvement in energy efficiency from new silicon technology has to be complemented with improvements on the algorithmic and the architectural level.

Advanced channel coding and modulation are major sources of complexity in digital baseband signal processing and largely contribute to the overall power consumption, area, and most important to the overall latency. The computational workload of channel decoding in today's 3GPP standards is already in the order of 100 Giga operations per second. Thus, in this paper we focus on advanced channel coding and MIMO detection schemes and the challenges for 100 Gbit/s implementation.

II. Channel Coding for 100 Gbit/s

Advanced channel codes like Turbo-Codes and LDPC codes combine randomness with limited locality (i.e. interleaver for Turbo-Codes, Tanner graph for LDPC codes, respectively) and some structures with iterative decoding techniques to achieve near channel capacity. However, energy efficient, low latency, and high throughput implementations require highly parallel architectures with large data locality, large regularity and a minimum of control flow. Thus, there is an inherent discrepancy between the information theoretical and the implementation objectives. To bridge this gap, a joined consideration of code, decoding algorithms and architectures is mandatory and parallelism has to be exploited on all levels [21]. The three prominent channel codes, namely Turbo-Codes, LDPC-Codes and the more recently discovered Polar codes [3], largely differ in this respect:

- Turbo-Code decoding is inherently serial and is mainly performed on data-flow graphs;
- LDPC-Code decoding is inherently parallel and is mainly performed on a data-flow graph;
- Polar code decoding is inherently serial and is performed on a code tree structure.

Thus, the challenges for decoder implementations achieving 100 Gbit/s and beyond are fundamentally different between Turbo-Codes, LDPC codes and Polar codes. This is illustrated in Table I. It contains projections of selected high throughput state-of-the-art decoder implementations down to 7 nm technology utilizing the scaling factors of the previous section. Note, that each design features different kind of codes, code block lengths and code rate flexibility, and communications performance is not considered. Thus, a direct comparison is not possible. However, the elemental conclusions that emerge from this table are:

- For a given communications performance LDPC code decoders show a distinct throughput advantage over Turbo and Polar code decoders, due to the inherently parallel nature of the LDPC code decoding (belief propagation algorithm).
- Existing Turbo-Code decoders cannot achieve 100 Gbit/s even in 7 nm technology.
- Turbo and Polar codes offer built-in flexibility with respect to code block sizes and code rates.
- Power density is the biggest challenge for existing decoders exceeding the 100 Gbit/s barrier.

The two most prominent techniques to achieve high throughput on architectural level are spatial parallelism and functional parallelism (pipelining). Pipelining has some efficiency advantages compared to spatial parallelism, but is limited in its applicability if control-flow and feedback loops play a major role. Channel decoding is mainly data-flow dominated. Thus, to achieve throughput beyond 100 Gbit/s, data-flow and tree structures can be flattened, or "unrolled", and pipelined respectively. This approach produces large locality on the layout level, but comes at the cost of decrease in flexibility. Note, that an early VLSI implementation of a Turbo-Code decoder also used iteration unrolling [13].

A. High Throughput Turbo-Code Decoding

In a basic Turbo-Code decoder, two component decoders process a complete code block alternatingly and exchange extrinsic data through an interleaver/de-interleaver. Commonly, one highly parallelized, monolithic BCJR decoder core functions alternatingly as component decoder 1 and 2 [6]. However,

TABLE I
SELECTED TURBO/LDPC/POLAR CODE IMPLEMENTATIONS PROJECTED TO 7 NM

Code Ref.	Code length	Code rate support	Process [nm]	Area [mm^2]	Freq [MHz]	TP [Gbit/s]	Area eff. [Gbit/s/mm^2]	Energy eff. [pJ/bit]	Power dens. [W/mm^2]
Turbo [23]	18432	LTE	7	0.2	4730	25	123	50	6
Turbo [15]	18432	LTE	7	2.0	2400	92	46	65	3
LDPC [32]	2048	0.84	7	0.1	4095	278	2999	6	19
LDPC [21]	672	13/16	7	0.2	660	480	2057	1.5	3
Polar [7]	1024	Any	7	0.03	20	22	656	0.9	0.6
Polar [1]	1024	Any	7	0.07	2250	60	888	9	8
Polar [9]	1024	0.5	7	0.40	3735	3825	9914	1.7	17

the BCJR algorithm is inherently sequential and thus, achieving a throughput of tens of Gbit/s is a challenging task for Turbo-Code decoders.

Turbo-Code decoder architectures for high throughput decoding employ one of the following approaches to parallelize the code block processing on component decoder level:
Parallel Map (PMAP): The complete code block is split into sub-blocks, which are processed in parallel on multiple sub-decoder cores. The spatial parallelization approach is used in most reported state-of-the-art architectures [11], [23], [5].
Fully Parallel Map (FPMAP): Splitting the code block into sub-blocks of size one in combination with shuffled decoding [31] leads to the FPMAP architecture [17]. It has been demonstrated to achieve a very high throughput, however at the cost of a decreased area efficiency and flexibility [15].
Pipelined Map (XMAP): The data block is split into sub-blocks, which are decoded in a pipeline that implements an unrolling of the recursive state metric calculations [18], [27]. The X-shaped pipeline structure, that gives the architecture its name, has been proven optimal with respect to the amount of state memory storage [16].
For large degrees of parallelism, the splitting of the code block into (small) sub-blocks necessitates additional calculations to compute estimates for the state metrics at the sub-block borders in order to mitigate a degradation of the *Frame Error Rate* (FER) performance [18]. This, in turn, limits the maximum degree of parallelization for the PMAP and XMAP architectures because of the increasing length of the necessary acquisition calculations [18]. For the same reason, the FPMAP, which essentially works on sub-blocks of size 1, needs up to three times more decoding iterations as PMAP/XMAP decoders. This effect is expected to be even more pronounced at high code rates close to 1. Note, however, that one full decoding iteration for the FPMAP has only a two clock cycle latency, which is substantially shorter than for PMAP/XMAP decoders.

Considering the state-of-the-art Turbo-Code decoders in Table I, it becomes clear, that even assuming 7 nm technology, a throughput beyond 100 Gbit/s is not attainable with state-of-the-art monolithic decoder cores and improvements on different design levels are mandatory.

A solution to increase the throughput for state-of-the-art LTE Turbo-Code decoders, which requires minimal changes to the decoder architecture and is applicable to all decoder types, is *Iteration Balancing* [28]. For iteration balancing, the iteration control, which is used to terminate the decoding process early in the event of successful decoding, is extended to the *Transport Block* (TB) level. It makes use of the fact, that the transmission data in LTE is organized in TBs, which consist of several code blocks. Instead of specifying a maximum budget n_{HI}^{max} of iterations that the decoder can spend for the decoding of individual code blocks, the iteration budget now considers the complete TB. Thereby, the decoder can make use of iterations that are saved by terminating the decoding process early for some code blocks. By distributing the saved iterations, the overall number of iterations needed to decode the complete TB can be reduced. For a TB size of 97896 bit, i.e. 16 code blocks of 6144 bit (coded: 18432 bit), a reduction of up to 30% without sacrificing FER performance has been shown for the architecture from [11]. This translates to an increase in throughput of about 42%. Also note, that, because of the higher dynamic range, the potential increase in minimum throughput - and with it the architecture efficiency - increases for larger TBs.

Throughput can be further increased by spatial parallelization, i.e. several decoder cores are instantiated, each of them processes a single code block individually. However, this decreases the architectural efficiency whereas the latency of a single code block is not reduced. Another possibility is functional parallelization that exhibits a much better architectural efficiency than spatial parallelization. Here, the various decoding iterations are unrolled and the individual iteration stages are pipelined. Instead of calculating a complete iteration in parallel as with the FPMAP architecture, a fully pipelined decoder would still calculate the recursions sequentially while the data travels through the pipeline. Since the XMAP architecture already follows a pipelining approach for the MAP decoder, it is best suited as basis for this unrolled architecture.

Figure 3 illustrates this new approach and contrasts it with the straightfoward decoder parallelization. It is easy to see that this streaming-like approach increases the architecture efficiency by using less extrinsic memory. Since the extrinsic memory makes up a significant percentage of XMAP decoders,

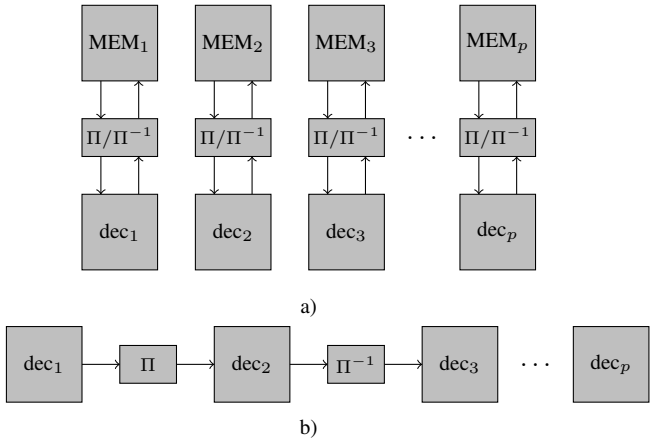

Fig. 3. Multiple Decoder case and Unrolled Iterations.

the architectural efficiency is much better. Furthermore, the fully unrolled architecture is highly data-flow oriented. The unidirectional routing and high data locality allows efficient placement & routing with modern synthesis tools and greatly reduces wiring congestions.

TABLE II
UNROLLED TURBO-CODE DECODER SYNTHESIS RESULTS.

Code Length	Core Area [mm^2]	Freq. [MHz]	TP [Gb/s]	Area Efficiency [Gb/s/mm^2]
192	9	750	48	5.33

Table II lists synthesis results for a prototype of an unrolled XMAP Turbo-Code decoder in 28 nm Fully Depleted Silicon On Insulator (FD-SOI) technology under worst case *power/voltage/temperature* (PVT) conditions, and gives estimates for area and throughput. The synthesis results for the decoder, that consists of 12 half-iteration stages and supports a information block length of 64 bit show, that unrolling of the decoding iterations is a promising approach towards very high throughput Turbo-Code decoders. Extrapolating these number to 7 nm technology results in a throughput of about 140 Gbit/s.

B. High Throughput LDPC Decoding

Unlike Turbo-Codes, LDPC decoding is inherently parallel. The decoding is based on the iterative message exchange between the check and variable node sets of the Tanner Graph. The calculations for the check and variable nodes can easily be done in parallel and the decoder throughput is only limited by the data exchange between the node sets. Similar to the fully unrolled approach for Turbo-Code decoders discussed in the previous section, the iterative decoding of LDPC codes can be unrolled to a fully pipelined decoder. In [4], [20] and [21], the Tanner graph is mapped directly into hardware for each iteration and the individual iteration stages are connected via pipeline registers. However, the number of iterations and the code block size have to be fixed for this approach. With this approach, throughput in the order of hundreds of Gbit/s (160 Gbit/s for 28 nm FD-SOI), and energy efficiency in the order of single digit pJ/bit (6 pJ/bit in 28 nm FD-SOI) are possible in future technologies. The rows for references [20], [4] in Table I show the projections to 7 nm technology.

However, achieving these throughput numbers with an increased flexibility with respect to code rates and block size, however, remains still an open topic for research.

C. High Throughput Polar Code Decoding

Polar Codes, invented 2008 by Erdal Arikan, are the first codes proven to achieve channel capacity for Binary Symmetric Memoryless Channels (BSMC). They belong to the class of multilevel concatenated codes and use the phenomenon of channel polarization to maximize coding efficiency which distinguishes them from the similar Reed-Muller-Codes.

In contrast to Turbo-Codes and LDPC codes, polar codes exhibit a quite regular structure. But at the same time the recursive nature of polar codes hinders an efficient parallelization. The standard decoding algorithm, namely successive cancellation and its derivatives, work on a code tree structure. Although successive cancellation has a low implementation complexity, the decoding itself is inherently serial and limits the throughput. A possibility to increase the throughput is the unrolling of the traversal on the tree. This results in a data-flow architecture that can be pipelined in a similar way as described above [9]. This approach can achieve a throughput far beyond 100 Gbit/s, however results in a big power density challenge, is limited to the fast simplified successive cancellation algorithm, i.e. without list/CRC, and has no flexibility at all. Omitting the pipeline register results in a pure combinatorial decoder without any registers [7] which largely reduces the power, but also largely reduces the throughput.

Belief propagation is a further decoding option [1]. This algorithm works on the factor graph. The belief propagation is alike to LDPC decoding parallel. However, the decoding requires a very large number of iterations to be competitive in communications performance, which prohibits an unrolling of the iterations. Implementation results for the discussed decoding methods projected to 7 nm technology are listed in Table I.

III. MIMO RECEIVERS

MIMO techniques in combination with bit-interleaved coded modulation enable the transmission of independent and separately encoded data streams from each antenna in the same frequency band. In return, the receiver needs to perform computationally complex MIMO detection to separate the data streams and to generate bitwise log-likelihood ratios (LLRs) for the subsequent channel decoder. Since both MIMO detector and channel decoder must jointly meet the requirements with regard to throughput and energy efficiency the design of a MIMO receiver is particularly challenging. As argued in Section II very high throughput channel decoders employ fully pipelined architectures, thus requiring LLR values for a complete code block per clock cycle. In the following

we discuss challenges for high throughput MIMO detection matching the full pipeline utilization of the channel decoders.

Advanced MIMO detectors traverse a tree structure to find a shortest path. Although a tree has a high regularity and large data locality, the actual detection of the best path in this decision tree is difficult to parallelize. Exhaustive search, i.e., computation of all paths and selection of the minima allows for a maximum parallelism but is infeasible due to the large tree size. Therefore, advanced detection algorithms, like the sphere decoding, use sophisticated pruning techniques to limit the number of visited tree nodes to a very small subset. However this adaptive pruning at runtime requires a control driven sequential processing of an unpredictable number of nodes which limits parallelism. As a consequence sphere decoders are limited to throughputs below 1 Gbit/s on current silicon technology [2].

List Fixed Complexity Sphere Decoding (LFSD) has been introduced as an approach to overcome the complexity problem by performing an exhaustive search on a constrained tree only. This subtree is defined at design time by fixing the number of nodes per layer and thus the number of paths to a small subset. For the first time we propose to apply the concept of "unrolling" (as described in the context of iterative decoders) to the different layers of the tree. The resulting deeply pipelined architecture with a directed data-flow from the root to the leave layer is depicted in Figure 4. However, this comes again at the cost of reduced flexibility with respect to the number of antennas and modulation and some degradation of the communications performance.

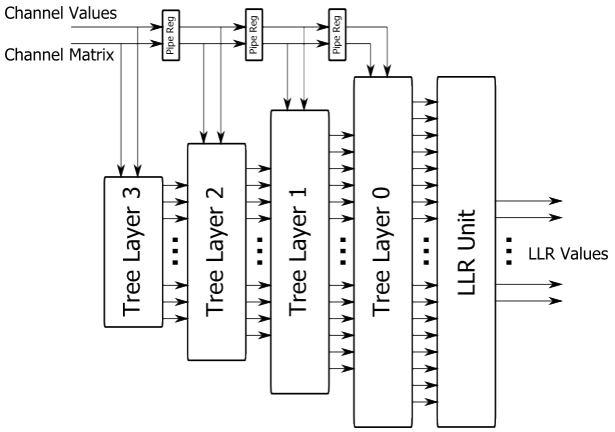

Fig. 4. Deeply pipelined, "unrolled" architecture for LFSD.

.

Other approaches that have been introduced for maximum throughput are k-best detection [29] and its soft-output extension, the path preserving trellis-search (PPTS) detector [25]. In [24], the authors present a fully parallel PPTS detector in 65 nm technology that achieves a throughput of 6.4 Gbit/s. However, this detector suffers in area efficiency, especially for higher order modulations (> 16 QAM).

Faster MIMO detection is limited due to the fact that MIMO detection operates on transmission vectors that typically consist of a small number of bits. As a consequence even the deeply pipelined MIMO detector achieves much lower throughput compared to a deeply pipelined channel decoder, that operates on much larger code blocks. However, since the different transmission vectors associated with one code block can be considered independent, the detection can be performed in parallel on multiple detectors.

E.g., assuming a 160 Gbit/s LDPC decoder with a code block size of 672 bit, as presented in the previous section, and a 4×4 64 QAM transmission system, a complete code block can be mapped on 28 independent transmission vectors, each 24 bits respectively. Thus, we can use 28 MIMO detectors in an array structure to match the throughput requirements of the channel decoder. Such a detector array is shown on the left side of Figure 5. Each detector is highlighted with a different color. The detectors are implemented as deeply pipelined LFSD with list size 16, performing the detection of all transmission vectors associated with a complete code block in each clock cycle. The total area requirement of the detectors in 28 nm FD-SOI technology is 4.3 mm^2 and the maximum operating frequency under worst case conditions is 240 MHz, resulting in a throughput of approximately 160 Gbit/s. The right side of Figure 5 depicts the aforementioned deeply pipelined LDPC decoder in the same technology. Here, each color highlights the groups of variable and check nodes associated with one iteration. The decoder features 10 iterations and supports a code rate of $5/8$, which is slightly different to the LDPC decoder in [21] (9 iterations, code rate $13/16$). The area of the LDPC decoder core is 3.3 mm^2.

The detector array can be interconnected via a fixed interleaving network to the LDPC decoder, resulting in a fully pipelined MIMO-BICM receiver. The total area requirement of the MIMO-BICM receiver is approximately 8 mm^2 and the resulting energy efficiency about 25 pJ/bit in 28 nm technology (6 pJ/bit extrapolated to 7 nm).

IV. Conclusion

In this paper, we discussed challenges and limitations for state-of-the-art channel decoding and MIMO detection in wireless digital baseband signal processing with throughput requirements of 100 Gbit/s and more. We showed, that improvements in silicon technology alone will not provide the necessary gains in terms of throughput and energy efficiency. The remaining performance gaps must be closed on the algorithmic and the architectural level. We demonstrated that flattening/unrolling of data-flow and tree structures in the channel decoder and the MIMO detector are a promising approach, however at the cost of flexibility. For the first time we have shown Turbo-Code decoding techniques towards 100 Gbit/s and a complete MIMO receiver beyond 100 Gbit/s in 28 nm technology.

Acknowledgment

We gratefully acknowledge financial support by the DFG (project-ID: 2442/8-1) and the EU (project-ID: 760150-EPIC).

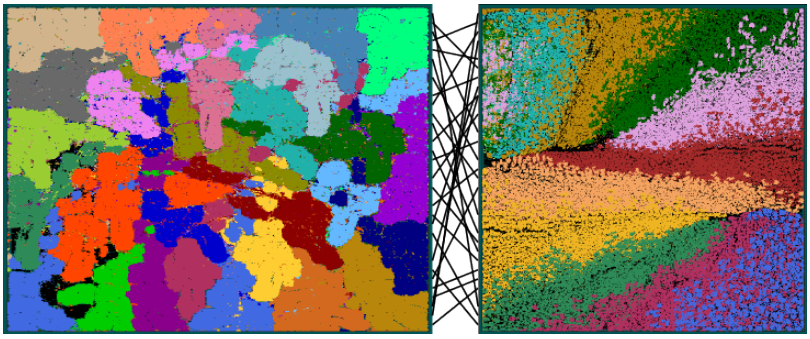

Fig. 5. MIMO-BICM receiver for throughput of 160 Gbit/s: MIMO detector array (left), LDPC decoder (right)

.

REFERENCES

[1] S. M. Abbas, Y. Fan, J. Chen, and C. Y. Tsui. High-Throughput and Energy-Efficient Belief Propagation Polar Code Decoder. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 25(3):1098–1111, March 2017.

[2] Esther Adeva, Thomas Augustin, and Gerhard Fettweis. Optimizing a pipelined MIMO sphere detector for energy efficiency. In *Wireless Communications Signal Processing (WCSP), 2015 International Conference on*, pages 1–6, Oct 2015.

[3] E. Arikan. Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels. *IEEE Transactions on Information Theory*, 55(7):3051–3073, July 2009.

[4] Alexios Balatsoukas-Stimming, Michael Meidlinger, Reza Ghanaatian, Gerald Matz, and Andreas Burg. A Fully-Unrolled LDPC Decoder Based on Quantized Message Passing. *arXiv preprint arXiv:1510.04589*, 2015.

[5] S. Belfanti, C. Roth, M. Gautschi, C. Benkeser, and Qiuting Huang. A 1Gbps LTE-advanced turbo-decoder ASIC in 65nm CMOS. In *VLSI Circuits (VLSIC), 2013 Symposium on*, pages C284–C285, June 2013.

[6] 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)*, pages 1064–1070, Geneva, Switzerland, May 1993.

[7] O. Dizdar and E. Arkan. A High-Throughput Energy-Efficient Implementation of Successive Cancellation Decoder for Polar Codes Using Combinational Logic. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 63(3):436–447, March 2016.

[8] G.P. Fettweis. The Tactile Internet: Applications and Challenges. *Vehicular Technology Magazine, IEEE*, 9(1):64–70, March 2014.

[9] Pascal Giard. *High-Speed Decoders for Polar Codes*. PhD thesis, McGill University Montreal, Canada, 2016.

[10] R. W. Heath, N. Gonzlez-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed. An Overview of Signal Processing Techniques for Millimeter Wave MIMO Systems. *IEEE Journal of Selected Topics in Signal Processing*, 10(3):436–453, April 2016.

[11] Thomas Ilnseher, Frank Kienle, Christian Weis, and Norbert Wehn. A 2.12Gbit/s Turbo Code Decoder for LTE Advanced Base Station Applications. In *2012 7th International Symposium on Turbo Codes and Iterative Information Processing (ISTC) (ISTC 2012)*, Gothenburg, Sweden, August 2012.

[12] ITRS 2.0. International Technology Roadmap for Semiconductors, 2015 Edition, Section 5: More Moore.

[13] M Jezequel and P Penard. Turbo4: a high bit-rate chip for turbo code encoding and decoding. 1999.

[14] F. Kienle, N. Wehn, and H. Meyr. On Complexity, Energy- and Implementation-Efficiency of Channel Decoders. *Communications, IEEE Transactions on*, 59(12):3301–3310, Dec 2011.

[15] A. Li, L. Xiang, T. Chen, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo. VLSI Implementation of Fully Parallel LTE Turbo Decoders. *IEEE Access*, 4:323–346, 2016.

[16] M. M. Mansour and N. R. Shanbhag. VLSI architectures for SISO-APP decoders. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 11(4):627–650, August 2003.

[17] R. G. Maunder. A Fully-Parallel Turbo Decoding Algorithm. *IEEE Transactions on Communications*, 63(8):2762–2775, Aug 2015.

[18] M. May, T. Ilnseher, N. Wehn, and W. Raab. A 150Mbit/s 3GPP LTE Turbo Code Decoder. In *Proc. Design, Automation and Test in Europe, 2010 (DATE '10)*, pages 1420–1425, March 2010.

[19] Anette Lundvall Patrik Cerwall, Stephen Carson. Ericsson Mobility Report November 2016, 2016.

[20] Philipp Schläfer, Norbert Wehn, Timo Lehnigk-Emden, and Matthias Alles. A New Dimension of Parallelism in Ultra High Throughput LDPC Decoding. In *IEEE Workshop on Signal Processing Systems (SIPS)*, Taipei, Taiwan, 2013.

[21] Stefan Scholl, Stefan Weithoffer, and Norbert Wehn. Advanced iterative channel coding schemes: When Shannon meets Moore. In *2016 9th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, pages 406–411, Sept 2016.

[22] M. Shafique, S. Garg, J. Henkel, and D. Marculescu. The EDA challenges in the dark silicon era. In *Design Automation Conference (DAC), 2014 51st ACM/EDAC/IEEE*, pages 1–6, 2014.

[23] R. Shrestha and R. P. Paily. High-Throughput Turbo Decoder With Parallel Architecture for LTE Wireless Communication Standards. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 61(9):2699–2710, Sept 2014.

[24] Y. Sun and J. R. Cavallaro. High-Throughput Soft-Output MIMO Detector Based on Path-Preserving Trellis-Search Algorithm. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 20(7):1235–1247, July 2012.

[25] Y. Sun and J. R. Cavallaro. Trellis-Search Based Soft-Input Soft-Output MIMO Detector: Algorithm and VLSI Architecture. *IEEE Transactions on Signal Processing*, 60(5):2617–2627, May 2012.

[26] O. Villa, D. R. Johnson, M. Oconnor, E. Bolotin, D. Nellans, J. Luitjens, N. Sakharnykh, P. Wang, P. Micikevicius, A. Scudiero, S. W. Keckler, and W. J. Dally. Scaling the Power Wall: A Path to Exascale. In *SC14: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 830–841, Nov 2014.

[27] S. Weithoffer, F. Pohl, and N. Wehn. On the applicability of trellis compression to Turbo-Code decoder hardware architectures. In *2016 9th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, pages 61–65, Sept 2016.

[28] S. Weithoffer and N. Wehn. Latency Reduced LTE-A Turbo-Code Decoding with Iteration Balancing on Transport Block Level. In *SCC 2017; 11th International ITG Conference on Systems, Communications and Coding*, 2017.

[29] K.W. Wong, C.Y. Tsui, R.S.K. Cheng, and W.H. Mow. A VLSI architecture of a K-best lattice decoding algorithm for MIMO channels. In *Proc. IEEE Int. Symp. Circuits and Systems ISCAS 2002*, volume 3, 2002.

[30] L. Young. Telecom Experts Plot a Path to 5G. *IEEE Spectrum Magazine*, October 2015.

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

[32] Zhengya Zhang, V. Anantharam, M.J. Wainwright, and B. Nikolic. An Efficient 10GBASE-T Ethernet LDPC Decoder Design With Low Error Floors. *Solid-State Circuits, IEEE Journal of*, 45(4):843–855, 2010.