3GPP TSG RAN WG1 Meeting #87 **R1-1612306**
Reno, Nevada, USA, 14th – 18th November 2016

Agenda item: 7.1.5.1
Source: AccelerComm
Title: On the hardware implementation of channel decoders for short block lengths
Document for: Discussion

I. Introduction

This paper compares the memory, computational and implementational complexity of enhanced turbo codes, Low Density Parity Check (LDPC) codes and polar codes. The following sections discuss these three codes in turn.

II. Enhanced turbo codes

An enhanced turbo code has been detailed in [1]. This design significantly improves upon the distance properties and hence the BLock Error Ratio (BLER) performance of the LTE turbo code by employing tailbiting, enhanced Almost Regular Permutation (ARP) interleavers [2], enhanced puncturing and mother coding rates as low as $R = 1/13$, as required for the NR eMBB control, uRLLC and mMTC channels. Here, the interleaving pattern depends only on the information block length K, while the puncturing depends only on the coding rate R, enabling simple implementation. In all cases, the sets of bits that are punctured at lower coding rates R are subsets of those that are punctured at higher rates. Since the puncturing patterns for successively lower coding rates R build upon each other in this way, the enhanced turbo code offers native rate-compatible support for Incremental Redundancy Hybrid Automatic Repeat reQuest (IR-HARQ).

Besides the requirement to store the additional parity Logarithmic Likelihood Ratios (LLRs) associated with low coding rates, the enhanced turbo decoder of [1] has the same memory requirement as the LTE turbo code. More specifically, Q_cK/R bits of memory are required to store systematic and parity LLRs, where $Q_c = 6$ bits may be used per LLR. Each element of this memory is read twice per iteration, namely once during each of the forward and backward recursions of the scaled-max-log-MAP algorithm [3]. Furthermore, MKQ_m bits of memory are required to store state metrics, where $M = 8$ states are employed in both the LTE and enhanced turbo codes and $Q_m = 9$ bits may be used per state metric. These state metrics are written during the forward recursions of the scaled-max-log-MAP algorithm, so that they can be read during the subsequent backward recursions, in order to generate extrinsic LLRs. Furthermore, KQ_e bits of memory are required to store *a priori* and extrinsic LLRs, where $Q_e = 8$ bits may be used per LLR. Here, *a priori* LLRs may be read from this memory during both the forward and backward recursions, then overwritten with extrinsic LLRs during the backward recursion. Note that the state metric and LLR memories can be reused by both the upper and lower decoders, since they are operated alternately. The total number of memory accesses is given by $IK(12/R + 336)$, as characterized in Figure 1, for the case of performing $I = 8$ iterations for an information block length of $K = 1024$ bits.

As shown in Figure 2, the scaled-max-log-MAP algorithm for the enhanced turbo code of [1] has only a modest computational complexity cost, compared to that of the LTE turbo code. At the cost of increasing the complexity of turbo encoding, tailbiting actually reduces the complexity of turbo decoding. This is because tailbiting eliminates the requirement for termination bits and their processing, as shown in Figure 2. The enhanced interleaver designs of [1] are simply reparametrizations of the same ARP interleavers used in

the LTE turbo code, therefore imposing no additional complexity. While the enhanced puncturer operates in a different manner to the LTE puncturer, its storage requirement is no greater than 16 bits per supported coding rate R [1]. As shown in the attached Matlab code and in Figure 2, the employment of a mother coding rate of $R = 1/13$ increases the computational complexity of the enhanced turbo decoder by around 25%, relative to that of the LTE turbo decoder, which employs a mother coding rate of $R = 1/3$. More specifically, the computational complexity associated with performing I iterations of the upper and lower decoders is increased from $76KI$ and $74KI$ Addition or Compare/Select (ACS) operations, to $102KI$ and $100KI$, respectively. However, this complexity can be reduced by precomputing various summations of the parity LLRs before beginning the iterative decoding process, albeit at the cost of requiring additional memory to store these summations. A further complexity reduction can be achieved by eliminating the processing relating to punctured bits, when employing coding rates greater than $R = 1/13$. Indeed, a significant computational complexity reduction can be achieved by limiting the mother coding rate to $R = 1/5$, meeting the requirement for the NR eMBB data channel. In this case, the upper and the lower decoders perform $84KI$ and $82KI$ ACS operations, representing only a 10% increase in computational complexity, relative to the LTE turbo decoder. As shown in Figures 3 and 4, the computational complexity of $I = 8$ iterations of the scaled-max-log-MAP algorithm for the enhanced turbo code does not depend on the coding rate R and scales linearly with information block length K.

An enhanced turbo decoder designed for short block lengths can be efficiently implemented in hardware using $P = 8$ parallel processing elements, each of which can process a different window of K/P trellis stages in parallel, as in the example turbo decoder Application Specific Integrated Circuit (ASIC) [4] characterized in Table I. These hardware resources can be fully-exploited across the full-range of supported information block lengths K, provided that they are multiples of $P = 8$, which is necessary to avoid contention during ARP interleaving. In this way, the area- and energy-efficiency of the hardware implementation can be flexibly maintained across all supported coding rates and information block lengths. Since the enhanced turbo decoder of [1] has a computational complexity that is no more than 25% higher than that of the LTE turbo decoder, it may be expected that the corresponding area- and energy-efficiencies would be only slightly degraded relative to those presented in Table I.

Observation 1: The computational complexity of scaled-max-log-MAP decoding of the enhanced turbo code of [1] is only 10% higher than that of the LTE turbo code for the eMBB data channel and only 25% higher for the eMBB control, uRLLC and mMTC channels.

Observation 2: Turbo decoder ASICs having $P = 8$ parallel processing elements can maintain area- and energy-efficiency across all supported coding rates and information block lengths.

III. LDPC CODES

Qualcomm have proposed a flexible LDPC code [6], which is based on adjusted-min-sum decoding. As described in [6], the total check node memory requirement is $K(1/R-1)(2(Q_i-1)+10)$ bits, where $Q_i = 5$ is the number of bits employed per LLR. Each element of this memory is read d_c times per iteration and is written d_c times per iteration, where $d_c = 5$ is the average degree of the check nodes [6]. Furthermore, the number of bits of memory used for channel LLRs and check-to-variable node messages is Q_cK/R bits, where $Q_c = 7$ is the number of bits employed per LLR. Each element of this memory is read $d_v = d_c(1 - R)$ times per iteration and is written $d_v = d_c(1 - R)$ times per iteration, where d_v is the average degree of the variable nodes. So the total number of memory accesses is given by $250KI(1/R - 1)$, where $I = 25$ iterations are typically required to match the BLER performance of the enhanced turbo code of [1]. As shown in Figure 1, the LDPC decoder of [6] performs a higher number of memory accesses than the enhanced turbo decoder of [1] for coding rates of $R = 2/3$ and below. Note that these low coding rates are particularly important in the eMBB control, uRLLC and mMTC channels. Indeed, at a coding rates of $R = 1/6$ and below, the number of memory accesses performed by the LDPC decoder is more than an order of magnitude higher than that of the enhanced turbo decoder.

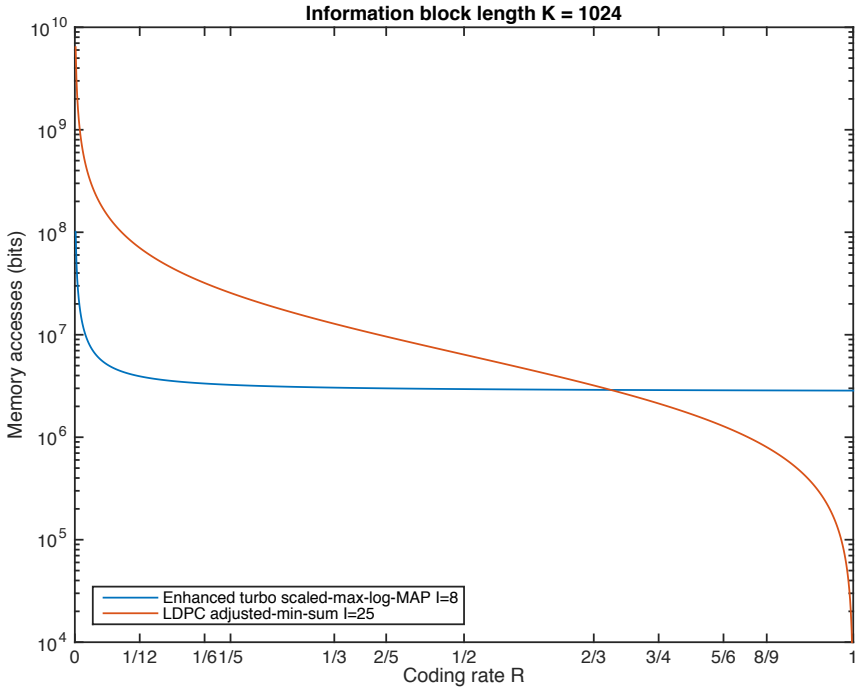

Fig. 1. Memory accesses versus coding rate R for enhanced turbo and LDPC decoders having information block lengths of $K = 1024$.

This highlights the observation that the area- and energy-efficiency of LDPC decoders is dominated by memory, rather than by computation [7]. Indeed, a very high memory bandwidth is required in order to meet the latency budget in the worst case, which is encountered for low coding rates R. This fundamental problem leads to poor area- and energy-efficiency in flexible LDPC decoder ASICs [8].

The above-described memory bandwidth problem is exacerbated by the high computational complexity of LDPC decoders at low coding rates. As described in [6], the computations performed by the Qualcomm adjusted-min-sum LDPC decoder for each edge of the Parity Check Matrix (PCM) in each iteration are as follows.

> "The check node processor requires *two adders* to reconstruct the variable node messages and *one absolute value calculator* and *two comparators* to find the first and second min. *Another adder* is needed to apply the offset. The variable node processor requires *two adders* to reconstruct the input messages and *another two* to calculate the new message."

Each adder, comparator and absolute value calculation corresponds to a single ACS operation, giving a total of 10 ACS operations per edge per iteration. The number of edges is given by $K(1/R-1)d_c$ where the average check node degree is $d_c = 5$. So the overall complexity is given by $50KI(1/R-1)$, where $I = 25$ iterations are typically required to match the BLER performance of the enhanced turbo code. Note that in contrast to turbo decoders, the complexity of LDPC decoders scales with the coding rate R. As shown in Figure 3, the LDPC decoder of [6] has a higher computational complexity than the enhanced turbo decoder of [1] for coding rates of $R = 2/5$ and below. This increased computational complexity and the increased number of memory accesses performed at low coding rates R translates to degraded area- and energy-efficiency in hardware implementations of LDPC decoders. This is shown in Table I for the flexible LDPC decoder [5] that was identified as having the best area- and energy-efficiencies among a survey of over 100 ASICs [8].

At a coding rate of $R = 1/2$, the computational complexity of the LDPC decoder of [6] is comparable to that of the enhanced turbo decoder of [1] across all information block lengths K, as shown in Figure 4. However, the LDPC code of [6] uses a wide range of different LDPC lifting factors Z for different information block lengths K. Owing to this, the degree of parallelism that can be exploited

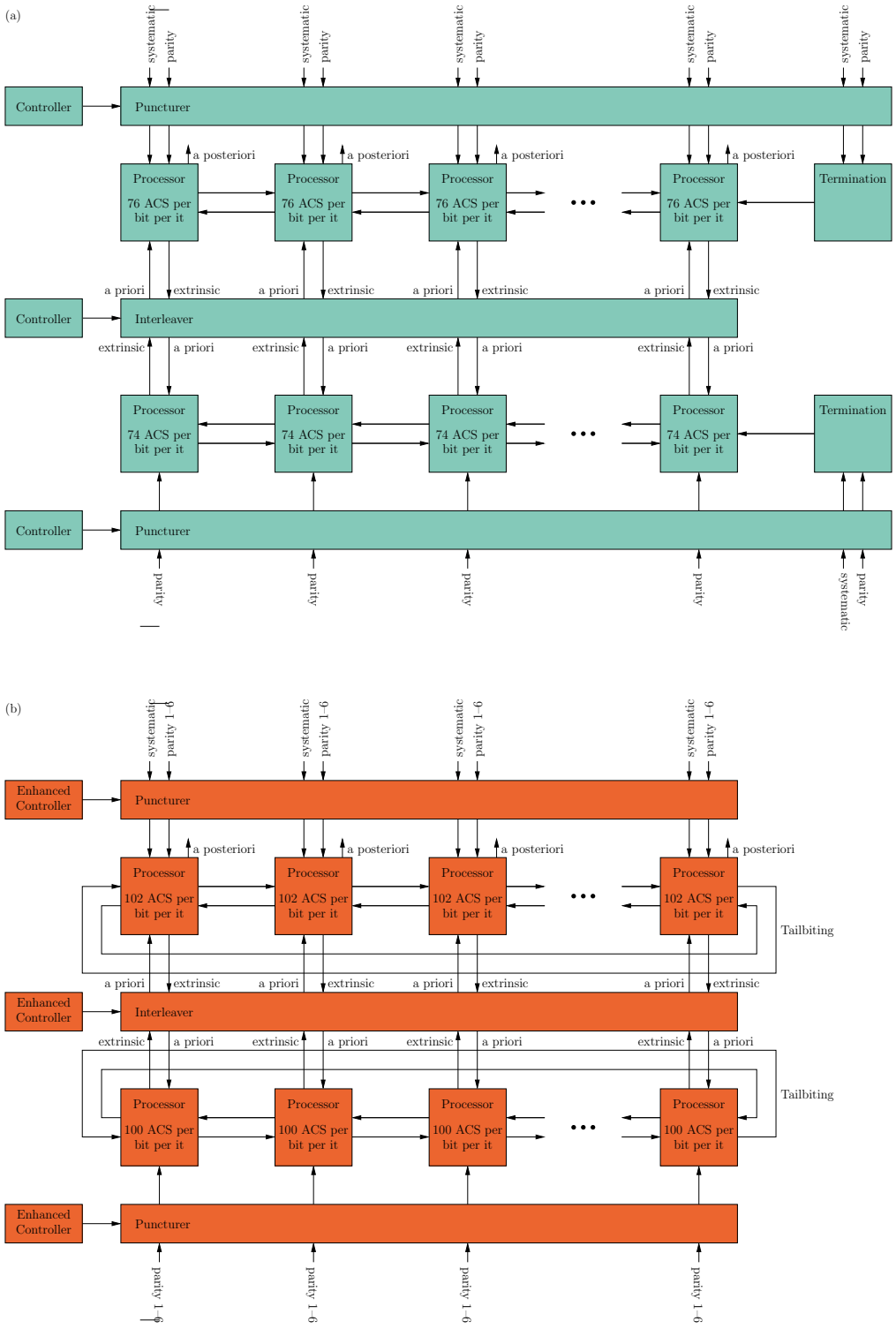

Fig. 2. (a) Schematic of the LTE turbo code, which employs termination, ARP interleavers, puncturing and a mother coding rate of $R = 1/3$. (b) Schematic of the enhanced LTE turbo code of [1], which employs tailbiting, enhanced ARP interleavers, enhanced puncturing and mother coding rates as low as $R = 1/13$.

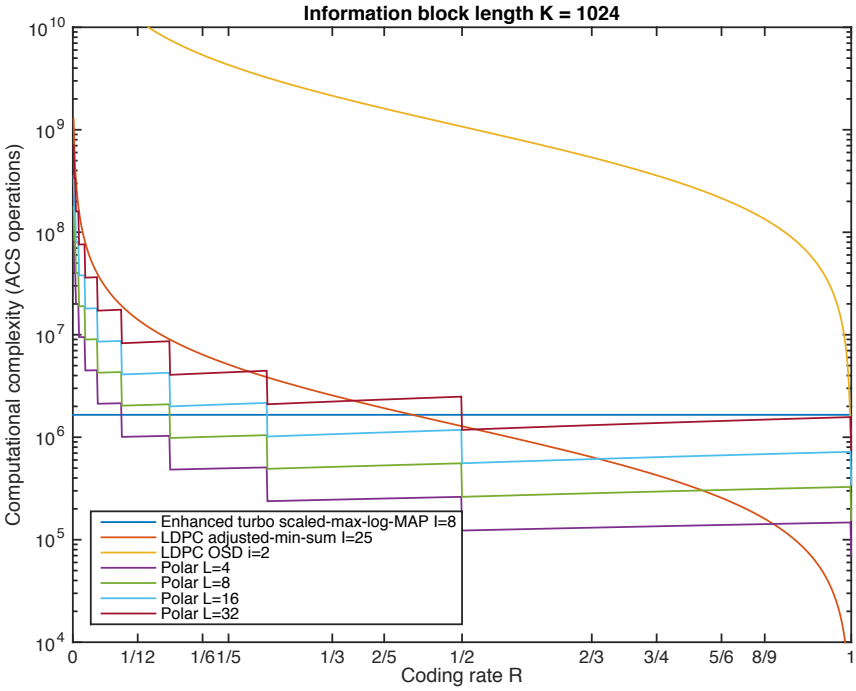

Fig. 3. Computational complexity versus coding rate R for various enhanced turbo, LDPC and polar decoders having information block lengths of $K = 1024$.

in a corresponding ASIC implementation of the LDPC decoder at short block lengths is orders of magnitude lower than can be exploited at high block lengths. As a result, a flexible LDPC decoder ASIC that can achieve very high information throughputs at long information block lengths would achieve orders of magnitude lower information throughputs at short information block lengths. This would cause corresponding degradation in the area- and energy-efficiency of this ASIC. Furthermore, the latency at lower block lengths would be orders of magnitude worse than at higher block lengths, further extending the challenge of meeting the latency budget in the worst case of short block lengths at low coding rates. Indeed, Table I shows that at short block lengths K, the flexible LDPC decoder of [5] has inferior (or comparable) information throughput, latency, area efficiency and energy efficiency than the LTE turbo decoder of [4], across all coding rates R, despite having a lower computational complexity at high coding rates. This may be explained by the significant challenge associated with the implementation of flexible LDPC decoders, which can support different combinations of information block length K and coding rate R, as described above.

Ordered Statistics Decoding (OSD) [9] allows quasi-Maximum Likelihood (ML) BLER performance to be achieved for LDPC codes. However, the complexity of OSD decoding is given by $K^{i+1}(1/R-1)$ ACS operations [9]. Here, i is the order of the OSD decoding, where $i = 3$ or $i = 4$ is typically required to approach the ML BLER performance for LDPC codes [9]. As shown in Figures 3 and 4, OSD decoding has excessively-high computational complexity at all coding rates R and at all but the shortest of information block lengths K, even if $i = 2$ is employed. It is worth noting that OSD LDPC decoding has very different processing requirements to adjusted-min-sum LDPC decoding. Owing to this, an OSD decoder used for short information block lengths would be required to use separate hardware to an adjusted-min-sum decoder used for longer block lengths. Indeed, an LDPC-only approach to the eMBB data channel may require three or more separate LDPC decoders. More specifically, an OSD decoder could be used for short information block lengths K. A flexible layered belief propagation decoder could be used to support all coding rates and block lengths, as well as IR-HARQ. Finally, a flooding decoder could be used to achieve very high information throughputs for selected coding rates and block lengths. However, rather

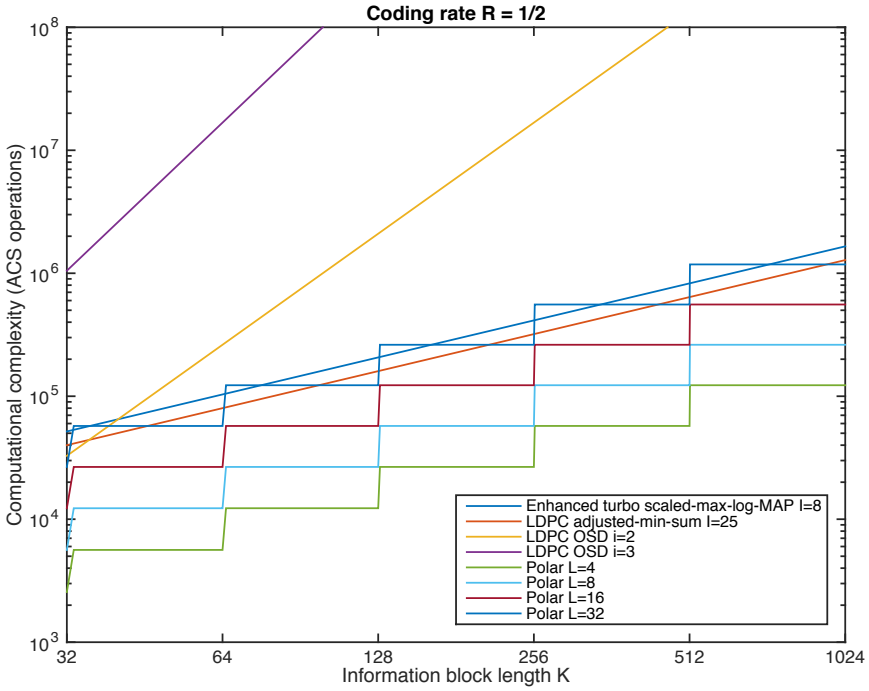

Fig. 4. Computational complexity versus information block length K for various enhanced turbo, LDPC and polar decoders having coding rates of $R = 1/2$.

than employing several complementary LDPC decoders, it has been shown that an improved overall area and power consumption can be achieved by using complementary turbo and LDPC decoders [8].

Observation 3: LDPC decoding performs a higher number of memory accesses than the enhanced turbo decoder of [1] for coding rates of $R = 2/3$ and below. At coding rates of $R = 1/6$ and below, the number of memory accesses performed by the LDPC decoder is more than an order of magnitude higher than that of the enhanced turbo decoder.

Observation 4: LDPC decoding has higher computational complexity than the enhanced turbo decoder of [1] for coding rates of $R = 2/5$ and below.

Observation 5: Flexible LDPC decoder ASICs suffer from degraded area- and energy-efficiency at low coding rates and short block lengths. Turbo decoder ASICs have superior area- and energy-efficiency.

Observation 6: Three or more separate LDPC decoders would be necessary to maintain BLER performance, area efficiency and energy efficiency across all information block lengths. Complementary turbo and LDPC decoders would have superior area and power consumption.

IV. Polar codes

The computational complexity of a polar decoder is given by [10] $L \cdot N \cdot \log_2(N) + L(N-1) + 2K \cdot L \cdot \log_2(2L)$, where $L \geq 2$ is the list size and the mother encoded block length N is the smallest power of two that is greater than K/R. More specifically, a particular combination of information block length K and coding rate R is achieved by puncturing the mother encoded block length N, to produce a number of encoded bits equal to K/R. This leads to a characteristic step function in the relationships between computational complexity and information block length K or coding rate R, as shown in Figures 3 and 4. Note that, in analogy with turbo decoders, some complexity reduction can be achieved by eliminating computations associated with punctured or frozen bits [10].

TABLE I
COMPARISON OF THE STATE-OF-THE-ART TURBO AND LDPC DECODER ASICS OF [4], AND [5].

Paper	[4]			[5]		
Year	2010			2013		
Published in	IEEE J. Solid-State Circuits			IEEE Trans. Circuits Syst. I		
Technology (nm)	90*			90		
Analysis	Measurement			Post-layout		
Code	Turbo			LDPC		
Supported standards	LTE			WiMAX, WiFi and G.hn		
Flexibility	188 information block lengths and full coding rate flexibility			133 combinations of encoded block length and coding rate		
Coding rate R	High 0.95	Medium 0.50	Low 0.33	High 0.83	Medium 0.50	Low –
Information throughput (Mbps)	388**	388**	388**	274**	165**	–
Latency**** for $K = 1024$ (ns)	2640	2640	2640	3733	6222	–
Hardware efficiency (Mbps/mm^2)	227	227	227	50	30	–
Energy efficiency (bit/nJ)	0.71	0.71	0.71	0.74***	0.44***	–

* The turbo decoder hardware characteristics presented in this table have been scaled from 130 nm to 90 nm technology.
** These throughputs have been linearly scaled to $I = 8$ iterations in the case of the turbo decoder and $I = 25$ iterations in the case of the LDPC decoder. In the case of the LDPC decoder, the results that are provided in [5] for short information block lengths are used.
*** The power consumption is stated as 228.36–517.70 mW in [5], but no discussion is provided about how this varies with coding rate or information block length. So, the average value of 373.03 mW has been used to calculate these energy efficiencies.
**** Latency is estimated by dividing the information block length $K = 1024$ by the information throughput, since latency is not quantified in [4] or [5]. Note that while neither of these decoders support information block lengths of exactly $K = 1024$, these estimates are provided for the sake of illustration.

However, in contrast to turbo decoders, the complexity of polar decoders scales with the coding rate R. As shown in Figure 3, polar decoders having list sizes of $L = 32$, 16 and 8 have higher computational complexities than the enhanced turbo decoder of [1] for coding rates that do not exceed $R = 1/2$, $1/5$ and $1/12$, respectively. Note that these low coding rates are particularly important in the eMBB control, uRLLC and mMTC channels. In hardware implementations of polar decoders, this increased computational complexity at low coding rates R translates to degraded area- and energy-efficiency. Furthermore, this creates a significant challenge of meeting the latency budget in the worst case, which is encountered for low coding rates.

Huawei have recently demonstrated a fully-flexible polar decoder ASIC [11], which supports many different combinations of information block length K, coding rate R, mother encoded block length N and list size L. Using 14 nm technology and a clock frequency of 1 GHz, this ASIC achieves area efficiencies of up to 13.43 Gbps/mm^2, although only at its highest supported coding rate of $R = 8/9$, its longest supported encoded block length of $N = 16000$ and its shortest supported list size of $L = 2$. However, for information block lengths of $K <= 1024$, $L = 4$ is the maximum list size that is supported across all coding rates in the range $R = 1/8$ to $R = 8/9$. The polar decoder ASIC of [11] achieves 4.03 Gbps/mm^2 at $K = 1000$, $R = 1/8$ and $L = 4$, while 6.25 Gbps/mm^2 is achieved at $K = 889$, $R = 8/9$ and $L = 4$.

However, there remains a significant challenge associated with the implementation of flexible polar decoders, which can support different combinations of information block length K, coding rate R and

mother encoded block length N. Owing to this and despite having higher computational complexities at many coding rates R, turbo decoder ASICs can readily achieve superior area- and energy-efficiencies. More specifically, several LTE turbo decoder ASICs [4], [12]–[16] including that of Table I have been demonstrated that could complete $I = 8$ decoding iterations with superior area efficiencies in excess of 13.43 Gbps/mm^2, if their areas were scaled to 14 nm technology and if their clock frequencies were correspondingly scaled to no higher than 1 GHz. Furthermore, as described in Section II, by employing $P = 8$ parallel processors, these LTE turbo decoder ASICs could maintain these superior area efficiencies for all information block lengths K and coding rates R, not just at the longest information block lengths K and the highest coding rates R.

Observation 7: Polar decoding with list sizes of $L = 32$, 16 and 8 have higher computational complexity than the enhanced turbo decoder of [1] for coding rates that do not exceed $R = 1/2$, $1/5$ and $1/12$, respectively.

Observation 8: Despite having lower computational complexity at many coding rates R, flexible $L = 4$ polar decoder ASICs have inferior area-efficiency compared to state-of-the-art turbo decoder ASICs.

V. Conclusion

This paper has compared the memory, computational and implementational complexity of enhanced turbo codes, LDPC codes and polar codes.

Observation 1: The computational complexity of scaled-max-log-MAP decoding of the enhanced turbo code of [1] is only 10% higher than that of the LTE turbo code for the eMBB data channel and only 25% higher for the eMBB control, uRLLC and mMTC channels.

Observation 2: Turbo decoder ASICs having $P = 8$ parallel processing elements can maintain area- and energy-efficiency across all supported coding rates and information block lengths.

Observation 3: LDPC decoding performs a higher number of memory accesses than the enhanced turbo decoder of [1] for coding rates of $R = 2/3$ and below. At coding rates of $R = 1/6$ and below, the number of memory accesses performed by the LDPC decoder is more than an order of magnitude higher than that of the enhanced turbo decoder.

Observation 4: LDPC decoding has higher computational complexity than the enhanced turbo decoder of [1] for coding rates of $R = 2/5$ and below.

Observation 5: Flexible LDPC decoder ASICs suffer from degraded area- and energy-efficiency at low coding rates and short block lengths. Turbo decoder ASICs have superior area- and energy-efficiency.

Observation 6: Three or more separate LDPC decoders would be necessary to maintain BLER performance, area efficiency and energy efficiency across all information block lengths. Complementary turbo and LDPC decoders would have superior area and power consumption.

Observation 7: Polar decoding with list sizes of $L = 32$, 16 and 8 have higher computational complexity than the enhanced turbo decoder of [1] for coding rates that do not exceed $R = 1/2$, $1/5$ and $1/12$, respectively.

Observation 8: Despite having lower computational complexity at many coding rates R, flexible $L = 4$ polar decoder ASICs have inferior area-efficiency compared to state-of-the-art turbo decoder ASICs.

REFERENCES

[1] Orange and IMT, "R1-1612938 Enhanced Turbo Codes for NR: Performance Evaluation for eMBB and URLLC," in *3GPP TSG RAN WG1 #87*, Nov. 2016.

[2] A. Nimbalker, Y. Blankenship, B. Classon, and T. K. Blankenship, "ARP and QPP interleavers for LTE turbo coding," in *Proc. IEEE Wireless Commun. Networking Conf.*, Las Vegas, NV, USA, mar 2008, pp. 1032–1037.

[3] J. Vogt and A. Finger, "Improving the max-log-MAP turbo decoder," *IET Electronics Letters*, vol. 36, no. 23, pp. 1937–1939, Nov 2000.

[4] C. Studer, C. Benkeser, S. Belfanti, and Q. Huang, "Design and implementation of a parallel turbo-decoder ASIC for 3GPP-LTE," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 8–17, Jan 2011.

[5] Y. L. Ueng, B. J. Yang, C. J. Yang, H. C. Lee, and J. D. Yang, "An efficient multi-standard LDPC decoder design using hardware-friendly shuffled decoding," *IEEE Trans. Circuits Syst. I*, vol. 60, no. 3, pp. 743–756, March 2013.

[6] Qualcomm, "R1-1610139 Efficient Channel Coding Implementations for EMBB," in *3GPP TSG RAN WG1 #86bis*, Oct. 2016.

[7] E. Amador, R. Pacalet, and V. Rezard, "Optimum LDPC decoder: A memory architecture problem," in *Proc. ACM/IEEE Design Automation Conf.*, July 2009, pp. 891–896.

[8] AccelerComm, "R1-1608584 Complementary turbo and LDPC codes for NR, motivated by a survey of over 100 ASICs," in *3GPP TSG RAN WG1 #86bis*, Sept. 2016.

[9] M. P. C. Fossorier, "Iterative reliability-based decoding of low-density parity check codes," *IEEE J. Selected Areas Commun.*, vol. 19, no. 5, pp. 908–917, May 2001.

[10] Huawei, "R1-164040 On latency and complexity," in *3GPP TSG RAN WG1 #85*, May 2016.

[11] ——, "R1-1608865 Design aspects of Polar Code and LDPC for NR," in *3GPP TSG RAN WG1 #86bis*, Oct. 2016.

[12] X. Chen, Y. Chen, Y. Li, Y. Huang, and X. Zeng, "A 691 Mbps 1.392mm2 configurable radix-16 turbo decoder ASIC for 3GPP-LTE and WiMAX systems in 65nm CMOS," in *Proc. IEEE Solid-State Circuits Conf.*, Nov 2013, pp. 157–160.

[13] S. Belfanti, C. Roth, M. Gautschi, C. Benkeser, and Q. Huang, "A 1Gbps LTE-advanced turbo-decoder ASIC in 65nm CMOS," in *Proc. Symp. VLSI Circuits*, June 2013, pp. C284–C285.

[14] C. Roth, S. Belfanti, C. Benkeser, and Q. Huang, "Efficient parallel turbo-decoding for high-throughput wireless systems," *IEEE Tran. Circuits Syst. I*, vol. 61, no. 6, pp. 1824–1835, June 2014.

[15] A. Ahmed, M. Awais, A. u. Rehman, M. Maurizio, and G. Masera, "A high throughput turbo decoder VLSI architecture for 3GPP LTE standard," in *Proc. IEEE Int. Multitopic Conf.*, Dec 2011, pp. 340–346.

[16] Q. Yang, X. Zhou, G. E. Sobelman, and X. Li, "Network-on-chip for turbo decoders," *IEEE Trans. VLSI Syst.*, vol. 24, no. 1, pp. 338–342, Jan 2016.