

# Survey of Turbo, LDPC and Polar Decoder ASIC Implementations

Shuai Shao, *Student Member, IEEE*, Peter Hailes, *Student Member, IEEE*, Tsang-Yi Wang, *Senior Member, IEEE*, Jwo-Yuh Wu, *Senior Member, IEEE*, Robert G Maunder, *Senior Member, IEEE*, Bashir M Al-Hashimi, *Fellow, IEEE*, Lajos Hanzo, *Fellow, IEEE\**

**Abstract**—Channel coding may be viewed as the best-informed and most potent component of cellular communication systems, which is used for correcting the transmission errors inflicted by noise, interference and fading. The powerful turbo code was selected to provide channel coding for Mobile Broad Band (MBB) data in the 3G UMTS and 4G LTE cellular systems. However, the 3GPP standardization group has recently debated whether it should be replaced by Low Density Parity Check (LDPC) or polar codes in 5G New Radio (NR), ultimately reaching the decision to adopt the LDPC code family for enhanced Mobile Broad Band (eMBB) data and polar codes for eMBB control. This paper summarises the factors that influenced this debate, with a particular focus on the Application Specific Integrated Circuit (ASIC) implementation of the decoders of these three codes. We show that the overall implementation complexity of turbo, LDPC and polar decoders depends on numerous other factors beyond their computational complexity. More specifically, we compare the throughput, error correction capability, flexibility, area efficiency and energy efficiency of ASIC implementations drawn from 110 papers and use the results for characterising the advantages and disadvantages of these three codes as well as for avoiding pitfalls and for providing design guidelines.

## I. INTRODUCTION

In cellular communication systems such as 3G UMTS [1] and 4G LTE [2], wireless transmission is used to convey data between user equipment and basestations, where the latter act as gatekeepers to the Internet and telephone networks. However, the received data typically differs from the transmitted data, owing to transmission errors caused by noise, interference and fading. In order to correct these transmission errors, cellular communication systems use forward error correction channel codes. More

specifically, a channel encoder is used in the transmitter (be it the user equipment or the basestation) to convert each so-called information block comprising  $K$  data bits into a longer encoded block comprising  $N > K$  encoded bits, which are transmitted. In the receiver, the additional  $(N - K)$  encoded bits provide the channel decoder with redundancy that allows it to detect and correct transmission errors within the original  $K$  information bits.

If the noise, interference or fading is particularly severe, then a low coding rate  $R = K/N$  will be required for the channel decoder to successfully detect and correct all transmission errors. However, a low coding rate implies the transmission of a high number  $N$  of encoded bits, which consume precious transmission time, energy and bandwidth resources. Therefore, desirable channel codes allow the successful detection and correction of transmission errors at coding rates  $R$  that approach the theoretical channel capacity [3].

In the past couple of decades, several near-capacity channel codes have emerged, including the classic turbo codes [4] adopted by the 3G UMTS [1] and 4G LTE [2] mobile broadband standards, the Low Density Parity Check (LDPC) codes [5] ratified by the WiFi [6], WiMax [7], WiGig [8], DVB-S2 [9] and 10GBase-T [10] standards as well as the more recent polar codes [11]. Both the turbo and LDPC codes employ an iterative decoding process, in which each successive attempt at decoding the information block informs the next, until the process converges to a legitimate codeword. By contrast, polar codes select the recovered information block from a list of candidates obtained from the associated parallel successive cancellation decoding processes, in which the decoding of each successive information bit informs the decoding of the next. For all three codes, the channel decoder

\*L. Hanzo would like to acknowledge the financial support of the EPSRC projects EP/Noo4558/1, EP/PO34284/1, of the Royal Society's GRFC Grant as well as of the European Research Council's Advanced Fellow Grant QuantCom.

TABLE I: List of abbreviations appeared in this paper

ABBREVIATIONS	
64QAM	$M = 64$ -ary Quadrature Amplitude Modulation
APP	{A Posteriori} Probability
ACS	Add Compare Select
ASIC	Application Specific Integrated Circuit
AWGN	Additive White Gaussian Noise
BEC	Binary Erasure Channel
BPSK	Binary Phase Shift Keying
BER	Bit Error Rate
BLER	Block Error Rate
CN	Check Node
DTS	Double Thresholding Strategy
eMBB	enhanced Mobile Broad Band
FER	Frame Error Ratio
FPGA	Field Programmable Gate Array
FPTD	Fully-Parallel Turbo Decoder
HARQ	Hybrid Automatic Repeat reQuest
IoT	Internet of Things
LDPC	Low Density Parity Check
Log-MAP	Logarithmic Maximum {A Posteriori}
Log-BCJR	Logarithmic Bahl-Cocke-Jelinek-Raviv
LLR	Logarithmic Likelihood Ratio
LOS	Line-of-Sight
LTE	Long Term Evolution
MAP	Maximum A-Posteriori
MBB	Mobile Broad Band
mMTC	massive Machine Type Communication
mmWave	Millimeter Wave
NB-IoT	Narrow Band Internet of Things
NSW	Non-Sliding Window
NR	New Radio
PCM	Parity Check Matrix
QPSK	Quaternary Phase Shift Keying
QPP	Quadratic Polynomial Permutation
SC	Successive Cancellation
SCL	Successive Cancellation List
SNR	Signal to Noise Ratio
VN	Variable Node
URLLC	Ultra-Reliable Low Latency Communication

has a much higher complexity than the corresponding encoder, since it uses iterative or parallel decoding processes, which rely on probabilistic representations of the encoded bits to overcome the uncertainty introduced by noise, interference and fading. Owing to this, it is the error correction performance and implementation characteristics of the channel decoder that are typically the main concerns when designing a channel code. These implementation characteristics include reconfiguration flexibility, processing throughput, processing latency, energy efficiency and hardware efficiency.

At the time of writing, the 3GPP standardization group is deliberating on the 5G specifications under the terminology of New Radio (NR), where the turbo code of the operational 3G UMTS and 4G LTE systems has been replaced by an LDPC code for its enhanced Mobile Broad Band (eMBB) data mode, supported by

a polar code in the eMBB control mode. To elaborate a little further, the turbo code has been replaced, because it is considered to be incapable of efficiently achieving the multi-Gbps processing throughput required for eMBB. In addition to having an increased throughput, the eMBB mode aims for attaining an improved coverage, which will enable faster file-downloads and more reliable video streaming, for example. Furthermore, 5G is also targeting both Ultra-Reliable Low Latency Communication (URLLC) and massive Machine Type Communication (mMTC) [12] applications. Explicitly, URLLC will offer significantly improved error correction capability and latency, in order to support mission-critical applications, such as autonomous vehicles [13]. By contrast, mMTC will offer significantly improved energy efficiency for the Internet of Things (IoT).

There are several survey papers [14]–[22], which discussed the implementation of channel decoders. For example, Roth *et al.* reviewed the trade-offs of area, throughput, and energy-efficiency in the content of the Very-large-scale Integration (VLSI) implementation of LDPC decoders. Brejza *et al.* disseminated a comprehensive treatise on the design of energy-aware turbo coding and its related applications. Hailes *et al.* reviewed the LDPC decoder implementations based on Field Programmable Gate Arrays (FPGAs) at a glance. Hence the main channel decoding implementation oriented survey papers are listed in Table II.

Against the background described above, this treatise provides an overview and comparison of turbo, LDPC and polar codes, with particular focus on their capability to meet the different requirements associated with the eMBB, URLLC and mMTC applications of 5G, as detailed in Section II. The operation and characteristics of turbo, LDPC and polar codes are detailed in Section III. The processing throughput, error correction capability and flexibility of 110 published Application Specific Integrated Circuit (ASIC) based turbo, LDPC and polar implementations will be characterised in Section IV as a function of their area- and energy-efficiency. Finally, Section V presents recommendations for channel decoder designers as well as conclusions and opportunities for future work. The structure of this paper has been illustrated in the Fig. 1 for better understanding.

## II. THE REQUIREMENTS FOR 5G

Like 3G UMTS and 4G LTE, the aim of the 3GPP 5G NR is to continue the trend of offering substantially improved user experience and more diverse applications for cellular communications. However, this will

TABLE II: Comparison of different channel coding implementation survey papers.

Year	Paper	Focus/Main contribution
2010	H. Mercier <i>et al.</i> [14]	Survey of error correcting codes for channels corrupted by synchronization errors with potential applications and challenges
2011	C. Roth <i>et al.</i> [15]	Survey of area, throughput, and energy-efficiency trade-offs in the VLSI implementation of LDPC decoders
2012	Nasruminallah and L. Hanzo [16]	Source and channel decoding and their iterative decoding and transmission schemes offered to multimedia communication systems
2013	H. Chen and R. G. Maunder and L. Hanzo [17]	Low-complexity turbo coding techniques and a holistic hybrid ARQ design
2015	Y. Fang <i>et al.</i> [18]	Protograph LDPC codes and their applications
2015	E. Ankan <i>et al.</i> [19]	Challenges and new directions in spatially coupled LDPC codes, non-binary LDPC codes, and polar coding
2016	M. F. Brejza <i>et al.</i> [20]	Turbo coding and energy-aware design guidelines for energy-constrained wireless applications
2016	P. Hailes <i>et al.</i> [21]	LDPC decoder implementations based on FPGA
2017	H. Ben Thameur <i>et al.</i> [22]	LDPC convolutional codes design based on different scheduling techniques
	This paper	Research of turbo, LDPC and polar decoder ASIC implementation and design comparison

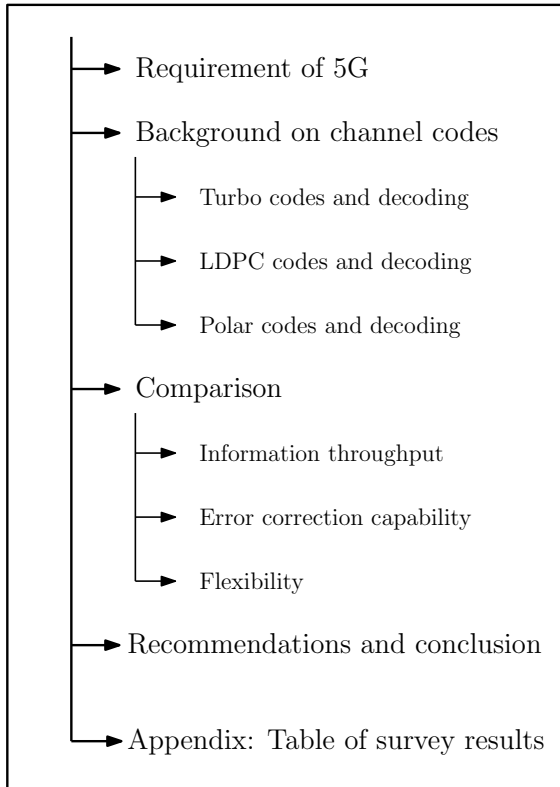


Fig. 1: Structure of this paper

impose stricter requirements upon all system components, including the channel code [23], as summarized in the following discussions. Note that the different eMBB, and mMTC applications for 5G impose different requirements, which may be impossible to meet

simultaneously using a single channel code. Therefore, it may be expected that different channel codes are adopted for different applications.

#### A. Processing throughput

The strictest throughput requirements are imposed by the eMBB applications of 3GPP 5G NR. A peak transmission throughput of 20 Gbps is targeted for these eMBB applications, which is much higher than the 1 Gbps achieved by 4G LTE. During video streaming, this significantly improved throughput will enable opportunistic forward buffering, when the channel conditions are favorable, for example. This will substantially increase the reliability of streamed video, which currently suffers from 'stutter' when the channel conditions become unfavorable.

Since all received information has to pass through the channel decoder, it must offer an information throughput of  $T = 20$  Gbps. Achieving this information throughput will require a high degree of parallel processing. If we (perhaps optimistically) assume that  $I = 10$  decoding steps (namely iterations or successive cancellation steps) are required and that the channel decoder processors can operate at a clock frequency of  $F = 1$  GHz, then at least  $P = I \cdot T / F = 200$  parallel processors will be required. This parallelism may be implemented internally, by using an array of  $P$  processors that collaborate during the processing of each block. Alternatively, a high information throughput can be achieved using external parallelism, where separate channel decoders are used for processing multiple blocks at the same time, or where multiple blocks are 'unrolled' and pipelined through the same decoder at

the same time [24]. However, this external parallelism approach does not achieve the same low latency benefit of internal parallelism, as it will be discussed in the following Section II-B.

### B. Processing latency

The strictest latency requirements are imposed by the URLLC applications of 3GPP 5G NR. An end-to-end latency of 0.5 ms is targeted for these URLLC applications, which is much lower than the 10 ms achieved in 4G LTE [25]. This significantly improved latency will allow user inputs made on a user equipment to be delivered to the cloud, processed on a cloud computer and then returned to update the display of the user equipment, without the user perceiving an objectionable delay, for example. This will enable new applications in user-specific 3D video rendering, augmented reality, remote control and mobile gaming, among others. Furthermore, since machines are more sensitive to latency than humans, these ultra-low latencies will support new mMTC applications, such as swarm robotics, factory automation, as well as vehicular control and safety applications.

However, an end-to-end latency of 0.5 ms implies a physical layer latency of 8 to 67  $\mu\text{s}$  [26], which allows data to be transmitted immediately after receiving control information, within the guard interval duration of 8 to 67  $\mu\text{s}$  in the same time slot. Furthermore, the channel decoder must share this latency budget with many other physical layer components, such as channel estimation and demodulation. Owing to this, the channel decoder should target processing latencies as low as 0.665  $\mu\text{s}$  [26]. This is achieved by channel decoders employing internal parallelism, which comprise different processors that collaborate on the processing of each block. Using this approach, an information throughput of  $T = 20$  Gbps for data blocks comprising  $K = 10,000$  bits implies a processing latency of  $L = K/T = 0.5\mu\text{s}$ , which meets the requirement described above. Note that while a high throughput can be achieved by using separate channel decoders to process multiple blocks at the same time, this approach does not improve the processing latency beyond that of each individual channel decoder. In other words, this relationship only holds under the assumption that no parallel processing of multiple frames is carried out at the same time, no pipelined operation of successive decoding steps is employed

### C. Error correction capability

The strictest error correction capability requirements are imposed in the URLLC applications of 3GPP 5G NR. The target for these URLLC applications is for only 1 block in every 100,000 to suffer from excessive transmission errors that cannot be corrected by the channel decoder, when the transmission throughput approaches the capacity of the communication link between the user equipment and basestation. More specifically, the URLLC channel code must not exhibit an error floor above a Block Error Rate (BLER) of  $10^{-5}$  [25]. This represents an order of magnitude improvement upon that of the 4G LTE turbo code, which has to guarantee a BLER below  $10^{-4}$  [27]. This additional improved error correction requirement further aggravates the improved latency requirement of Section II-B, since it makes it ten times less likely that the receiver will have to activate Hybrid Automatic Repeat reQuest (HARQ) [17], [28] to request a retransmission of erroneously decoded information, which imposes a significant additional latency. Despite this however, HARQ will remain a vital component of 5G, in order to enable ultra-reliable communication.

### D. Flexibility

The summary of requirements for 5G NR channel decoder has been illustrated in Fig. 2. The design of channel decoder has higher priority for variety characteristics based on different scenarios. Unlike other specifications, flexibility is necessary among three main dimensions of 5G NR, owing to the wide range of use cases targeted by each of eMBB, URLLC and mMTC. Owing to this, the channel code must support a wide variety of information block lengths  $K$ , as well as a wide variety of coding rates  $R = K/N$ . For example, short blocks comprising as few as tens or hundreds of information bits may be expected to be typical in URLLC, mMTC and control applications, while long blocks comprising as many as thousands of bits are typical in eMBB applications.

Likewise low coding rates will be required in rural areas, where basestations are deployed sparsely for covering large cells, while high coding rates may be used for ultra-dense urban deployments having strong Line-of-Sight (LOS). If the channel code does not support a wide variety of information block lengths  $K$ , then each information block may have to be padded with a high number of wasteful dummy bits for ensuring that its length becomes one of those legitimate ones supported by the channel code. Likewise, if the channel code does not support a wide variety of coding

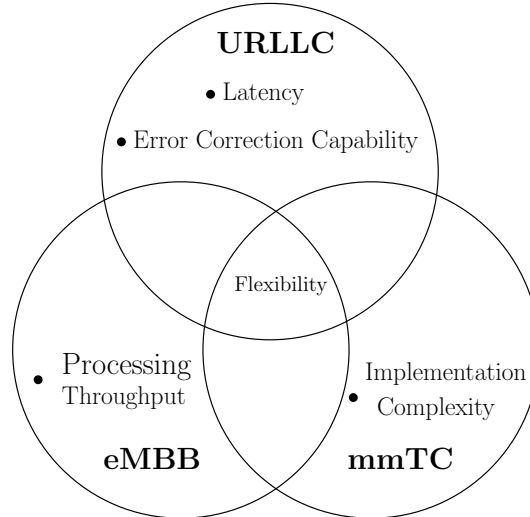


Fig. 2: Summary of requirements for 5G based on different scenario

rates  $R$ , then it may be necessary to select a rate that is lower than it is actually required by the current level of noise, interference and fading. Again, that implies the transmission of a high number of wasteful encoded bits. In both cases, the wasteful bits translate into wasted bandwidth. More specifically, the waste results in each transmission having a higher bandwidth, duration or energy than it is actually required, preventing other users from communicating at the same frequency, time or location, without suffering from increased interference. This will therefore degrade the throughput, latency and error correction capability that can be offered by the 3GPP 5G NR. For this reason, the 5G flexibility requirement is key to unconditionally fulfilling the other challenging 5G requirements of Sections II-A to Section II-C.

#### E. Implementation complexity

The implementation complexity of a channel code determines its hardware resource requirement and energy consumption. The strictest implementation complexity requirements are imposed by the mMTC applications of 3GPP 5G NR. In these mMTC applications, low-cost IoT devices have to operate continuously without recharging, while requiring very low chip area and energy consumption. In the uplink, these requirements are imposed upon the channel encoder, while they are imposed on the decoder in the downlink. In eMBB and URLLC applications, the hardware and energy efficiencies of the channel encoder and decoder have to be at least as good as those of 4G LTE.

Here, the hardware (or area) efficiency quantifies a channel encoder's or decoder's information throughput as a ratio to its ASIC area, which is measured in Mbps/mm<sup>2</sup>. Meanwhile the energy efficiency quantifies the number of information bits that may be decoded per nJ of energy consumed, which is equivalent to Mbps/mW.

### III. BACKGROUND ON THE CANDIDATE CHANNEL CODES

This section provides background discussions of the three mainstream channel codes considered in this paper, namely the turbo, LDPC and polar codes. In particular, we will detail the decoders of those codes, where the decoders have much higher complexity than the corresponding encoders. The following subsections discuss each code in turn, with reference to the structures illustrated in Figure 4.

#### A. Turbo codes

Turbo codes may be considered to be very mature in mobile broadband applications, since they were selected to provide flexible channel coding in the 3G UMTS [1] and 4G LTE [2] standards, which have been widely adopted world wide. Furthermore, turbo codes have been selected to provide channel coding in the 4G Narrow Band Internet of Things (NB-IoT) standard [29] for machine type communications.

A turbo encoder [30] comprises a parallel concatenation of two convolutional encoders [31], which are separated by an interleaver. The interleaver creates a

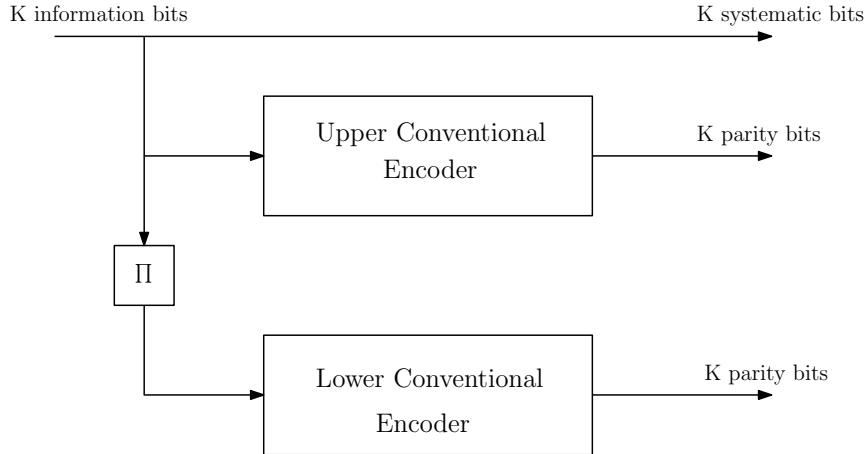


Fig. 3: Turbo encoder schematic.

replica of the  $K$ -bit information block, but rearranges the order of its bits according to a predetermined interleaving pattern. Following this, the two convolutional encoders encode the pair of differently ordered information block and each generates a  $K$ -bit parity block, as shown in Figure 3. These are typically concatenated together with a third replica of the  $K$ -bit information block, which is referred to as the systematic block. Puncturing and repetition [32] may also be used to remove or introduce additional bits for obtaining the resultant  $N$ -bit encoded block. The  $N$ -bit encoded block is then modulated and transmitted over the wireless channel, where it is exposed to noise interference and fading. Owing to the uncertainty introduced by these effects, the demodulator will typically be unable to express absolute confidence in the values of the  $N$  bits in the encoded block. Instead, the demodulator may express its confidence in the value of each bit using the corresponding Logarithmic Likelihood Ratio (LLR) [33]. Each LLR provides the logarithm of the ratio of the probabilities of the corresponding bit having the value 0 and 1. This approach enables soft turbo decoding, which iteratively exploits all received information, despite the uncertainty introduced by the wireless channel.

More specifically, the  $N$  LLRs of the encoded block may be decomposed into a systematic block and two parity blocks, each comprising  $K$  LLRs. These may be entered into a pair of convolutional decoders, which operate on the basis of the Maximum A-Posteriori (MAP) algorithm, also known as the Logarithmic Bahl-Cocke-Jelinek-Raviv (Log-BCJR) algorithm [34]. This employs a trellis diagram [31] for describing the relationships between the encoded bits,

which may be recursed in forward and backward directions [35] to obtain a block of  $K$  extrinsic LLRs pertaining to the information block. These  $K$  extrinsic LLRs are obtained by combining the received LLRs using both addition and Jacobian logarithm [34] operations, which are also known as the  $\max^*$  operation. The  $\max^*$  operation used for combining two LLRs  $x$  and  $y$  is given by

$$\max^*(x, y) = \max(x, y) + \ln(1 + e^{-|y-x|}), \quad (1)$$

which is simplified as  $\max^*(x, y) \approx \max(x, y)$  in the so-called max-log-MAP algorithm [34]. The order of these  $K$  extrinsic LLRs may be rearranged by the interleaver and entered into the other convolutional decoder as a block of  $K$  *a priori* LLRs. This supports the operation of the other convolutional decoder, enabling it to generate its own block of  $K$  extrinsic LLRs pertaining to the information block. In return, these extrinsic LLRs can be reordered by the interleaver and forwarded as *a priori* LLRs to the first convolutional decoder. These additional *a priori* LLRs may be exploited during a second decoding attempt of the first convolutional decoder, in order to improve its operation. This process may continue, with the two convolutional decoders iteratively exchanging successively ever-higher-quality extrinsic LLRs, until the process converges to a legitimate codeword. The LLRs pertaining to the information block may then be converted into the most likely bit values and output [20].

As shown in Figure 4(a), turbo codes may be considered to employ a regular structure, comprising two rows of  $K$  trellis stages corresponding to the two convolutional decoders, connected by the interleaver.

In Figure 4(a), the vertical exchange of information between the trellis stages via the interleaver corresponds to the iterative exchange of extrinsic LLRs, while the horizontal exchange of information along the two rows of trellis stages corresponds to the forward and backward recursions of the Log-BCJR algorithm. Since every trellis stage is identical, the turbo code equally protects the  $K$  bits in each information block. The complexity of the turbo code structure scales with  $K$ , since the interconnections of the interleaver pattern may be described by a  $1 \times K$  vector. Owing to the regularity of this structure, turbo codes can be readily designed to reuse hardware to flexibly support a wide range of information block lengths  $K$ . However, the forward and backward recursions within the Log-BCJR algorithm impose data dependencies, which many can limit the degree of parallel processing in practical implementations. This impediment has motivated substantial research efforts, which have mitigated the data dependencies and increased the degree of parallel processing using the techniques such as Non-Sliding Window (NSW) [36], radix-4 [36] and Fully-Parallel Turbo Decoder (FPTD) [37]. The impact of the turbo decoder architecture on the error correction capability, throughput, area efficiency and energy efficiency will be characterised in the scatter plots of Section IV. Furthermore, it is typically necessary to define a different interleaver pattern for each supported information block length  $K$ . In particular, the 4G LTE turbo code employs a Quadratic Polynomial Permutation (QPP) construction technique for generating 188 different interleaver patterns for information block lengths  $K$  in the range of 40 to 6144 bits [38].

In order to contrast the two extreme architectures of fully-serial and fully-parallel turbo decoding, their corresponding decoding schedules are illustrated in Tables III and IV. As shown in Table III, the fully-serial turbo decoder uses a single processor operating over  $K$  clock cycles to complete each of the forward and backward recursions on each of the upper set of trellis stages and lower set of trellis stages. In total, each decoding iteration of the fully serial turbo decoder requires  $4 \times K$  clock cycles, owing to the data dependencies imposed by the Log-BCJR algorithm. However, these data dependencies are ignored in the fully-parallel turbo decoder of Table IV, where a separate processor is used for each individual algorithmic block, enabling the forward and backward recursions of both upper and lower sets of trellis stages to be completed at the same time. Therefore, one iteration of fully-parallel turbo decoding can be finished in a

single clock cycle, using a total of  $2 \times K$  processors.

The examples of Tables III and IV illustrate the contrast between serial and parallel turbo decoders. The serial schedule efficiently propagates information along the forward and backward recursions, enabling a particular error correction capability to be achieved with fewer iterations and lower complexity than the parallel schedule. Despite this, the parallel schedule requires significantly fewer clock cycles overall, leading to superior throughput and latency. The Area and energy requirement of ASIC implementations of turbo decoders tends to be dominated by the capacity of the memory, which is the same for both the serial and parallel schedules. Owing to this, the high throughput of the parallel schedule typically leads to superior area and energy efficiency. However, the reconfiguration flexibility typically remains limited when implementing a parallel schedule due to the requirement for a fully laid out interconnection network that supports all interleaver designs, as will be detailed in Section IV.

Since the turbo code structure scales with the information block length  $K$ , the turbo decoder can be said to decode the information bits directly, rather than recovering the encoded bits and then extracting the information bits. Owing to this, the decoding complexity depends mainly on the information block length  $K$  and hence it does not vary significantly when the coding rate  $R$  is varied using puncturing and repetition, as shown in Figure 6. More specifically, the computational complexity  $C$  in terms of Add Compare Select (ACS) operations can be seen to be linearly proportional to the information block length  $K$  in Figure 7. In other words, the complexity of a turbo decoder is on the order of  $C = O(K)$ . In this way, the information throughput, and hence the decoding latency, hardware efficiency and energy efficiency do not vary significantly with the coding rate  $R$ , as shown in Figure 5a. This is illustrated by the analogy of Figure 5b, which shows that the information throughput of a turbo decoder remains constant, when the coding rate  $R$  is adjusted by varying the puncturing or repetition.

## B. LDPC codes

LDPC codes belong to an even more mature channel coding family than turbo codes, since they were proposed by Gallager in his PhD thesis half-a-century ago! They have also been selected for a variety of standards conceived for diverse applications. Perhaps most famously, LDPC coding is employed in WiFi [6], which is designed for wireless local area networking offering support for 12 combinations of medium infor-

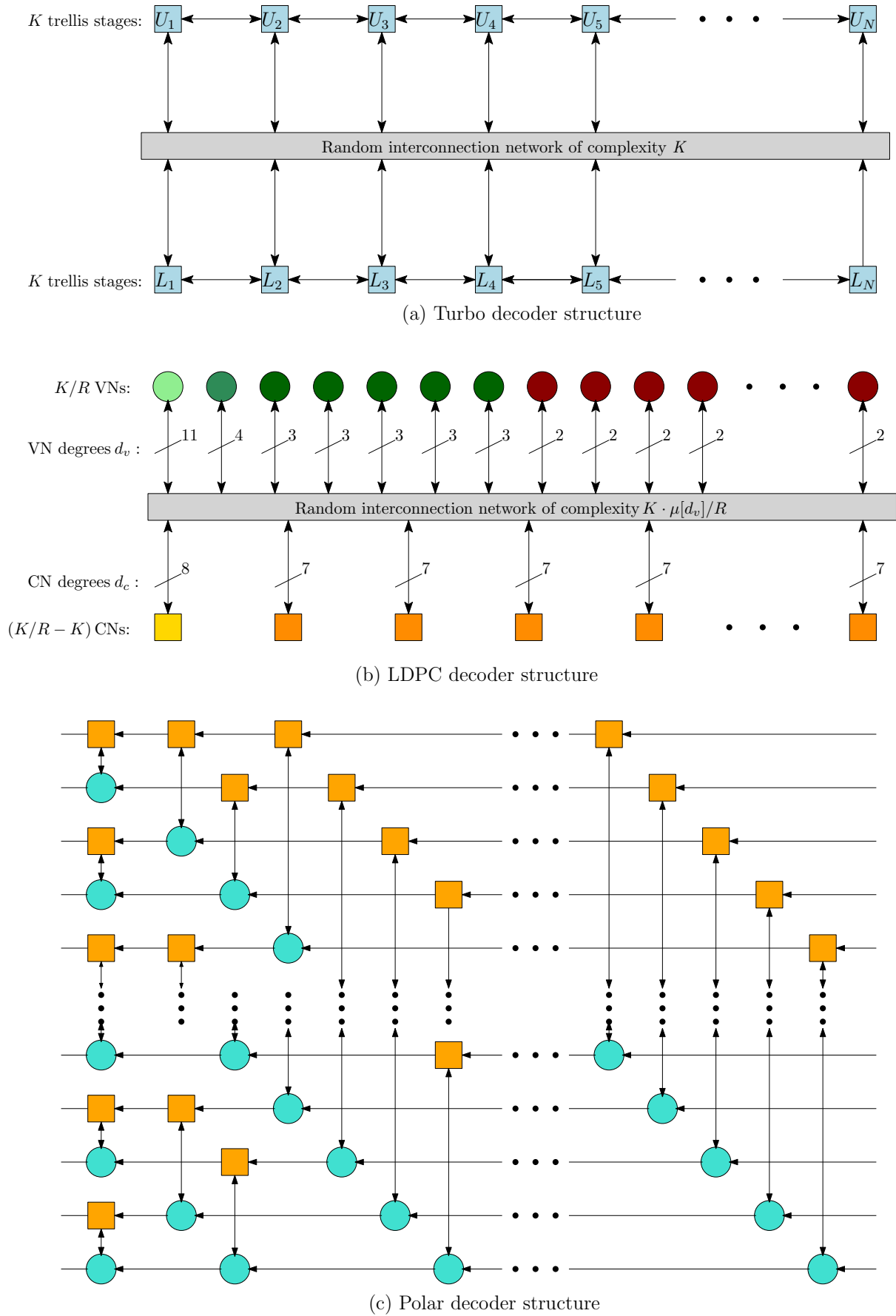
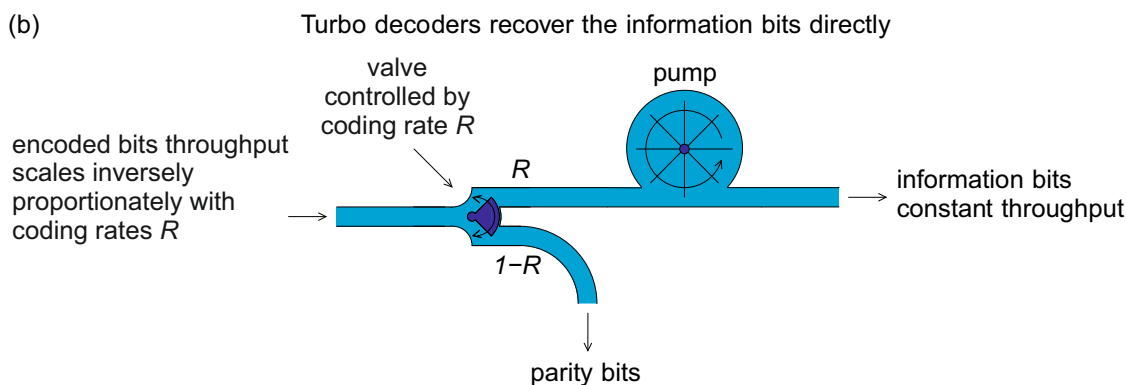
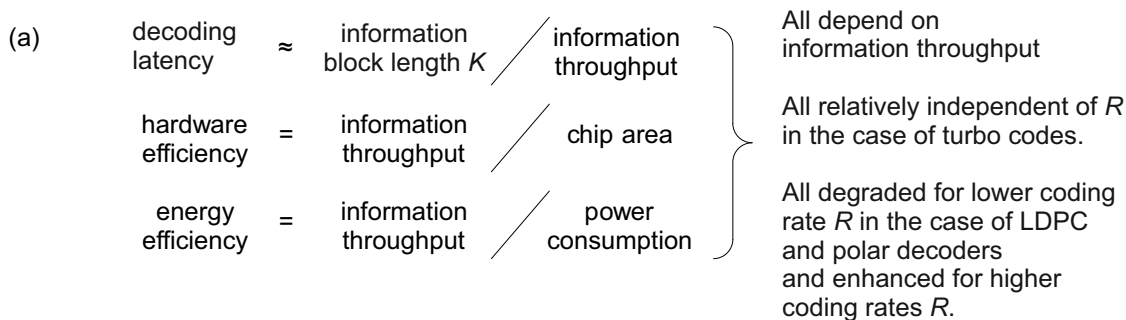


Fig. 4: Structures of turbo, LDPC and polar decoders.





(c) LDPC decoders recover the encoded bits, then extract the information bits.  
 Polar decoders recover the information and frozen bits, then extract the information bits

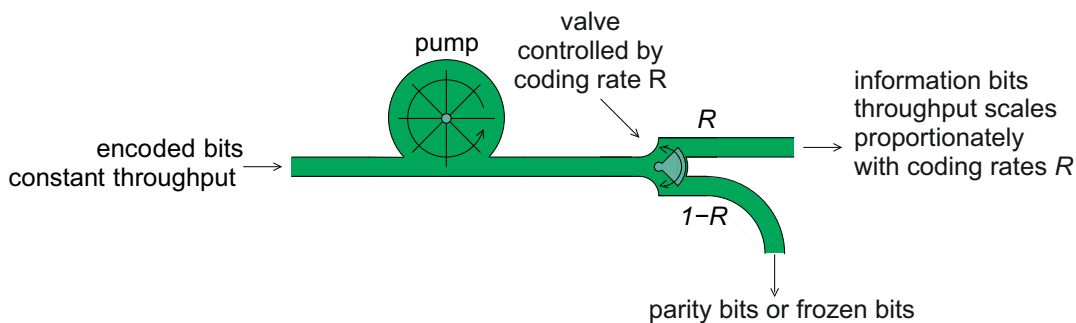


Fig. 5: (a) The relationship between information throughput and decoding latency, hardware efficiency and energy efficiency, as well as an analogy using pumps, valves and pipes, to illustrate how the coding rate  $R$  of (b) turbo and (c) LDPC or polar decoders affects their encoded and information throughputs.

TABLE III: Decoding schedule for one iteration of fully serial turbo decoder.

Clock cycle	$1^{st}$	$2^{nd}$	$3^{rd}$	...	$K^{th}$	$K + 1^{th}$	...	$2K^{th}$
Processor 1	$U_1$	$U_2$	$U_3$	...	$U_K$	$U_{K-1}$	...	$U_1$
Clock cycle	$2K + 1^{th}$	$2K + 2^{th}$	$2K + 3^{th}$	...	$3K^{th}$	$3K + 1^{th}$	...	$4K^{th}$
Processor 1	$L_1$	$L_2$	$L_3$	...	$L_K$	$L_{K-1}$	...	$L_1$

TABLE IV: Decoding schedule for one iteration of fully parallel turbo decoder.

Processor	$1^{st}$	$2^{nd}$	$3^{rd}$	...	$K^{th}$	$K + 1^{th}$	...	$2K^{th}$
Clock cycle 1	$U_1$	$U_2$	$U_3$	...	$U_K$	$L_1$	...	$L_K$

mation block lengths  $K$  and high coding rates  $R$ . The WiMAX standard [39] designed for mobile broadband communication employs an LDPC code that supports a similar range of information block lengths  $K$  and coding rates  $R$ , but with 144 options within that range. In the WiGig standard [8] ratified for Millimeter Wave (mmWave) communication, an LDPC code supporting 4 combinations of medium information block lengths  $K$  and high coding rates  $R$  is employed. The DVB-S2 standard [9] for satellite communication employs an LDPC code, which supports 29 combinations for high information block lengths  $K$  and medium to high coding rates  $R$ . In the Ethernet 10GBASE-T standard [40], an LDPC code supporting only a single fixed medium information block length  $K$  and a high coding rate  $R$  is employed for facilitating a high processing throughput. The impact of these different standards [8], [9], [39], [40] upon the error correction capability, throughput, area efficiency and energy efficiency, will be characterized in the scatter plots of Section IV.

An LDPC encoder obtains an  $N$ -bit encoded block  $\mathbf{U}$  by multiplying a block of  $K$  information bits  $\mathbf{x}$  by a  $(K \times N)$  generator matrix  $\mathbf{G}$  in  $\text{GF}(2)$ , according to  $\mathbf{U} = \mathbf{G} \cdot \mathbf{x}$ . In contrast to a turbo decoder, an LDPC decoder employs an irregular factor graph structure [21], which applies unequal protection to the  $K$  bits in each information block. More specifically,  $N$  LLRs pertaining to the encoded block are forwarded to a corresponding set of  $N$  Variable Nodes (VNs), which are connected to  $(N - K)$  Check Nodes (CNs). In detail, the  $V^{th}$  VN is connected to  $d_v$  number of CNs, where  $d_v$  is referred to as the degree of the VN. Similarly, the  $C^{th}$  CN is connected to  $d_c$  number of VNs, where  $d_c$  is referred to as the degree of the CN. In a similar manner to a turbo decoder, an iterative sum product decoding process [21] is used for exchanging extrinsic LLRs between the connected VNs and CNs.

More specifically, the extrinsic LLR provided by a VN for each of its connected CNs is obtained as the sum of the LLRs provided by all other connected CNs plus the LLR provided by the channel. Meanwhile, the extrinsic LLR provided by a CN for each of its connected VNs is obtained by the so-called box-plus summation of the LLRs provided by all other connected VNs. To elaborate a little further, the box-plus sum of two LLRs  $x$  and  $y$  is given by

$$\begin{aligned} \min^*(x, y) = & \text{sign}(x) \cdot \text{sign}(y) \cdot \min(|x|, |y|) \\ & + \ln(1 + e^{-|x+y|}) - \ln(1 + e^{-|x-y|}) \end{aligned} \quad (2)$$

which is simplified to

$$\min^*(x, y) \approx \text{sign}(x) \cdot \text{sign}(y) \cdot \min(|x|, |y|) \quad (3)$$

in the so-called min-sum algorithm [41].

The complexity of this factor graph structure scales with the encoded block length  $N$ , since the interconnections between the VNs and CNs in the LDPC decoder may be described by a  $(N - K) \times N$  matrix, which is referred to as the Parity Check Matrix (PCM) [21]. Note that the dimensions of the PCM grow as the coding rate  $R$  is reduced.

The computations performed within each LDPC VN and CN typically impose a significantly lower complexity than the computations performed for each trellis stage of a turbo code, hence typically leading to a lower LDPC decoder complexity, particularly at high coding rates. However, while a turbo decoder has  $K$  random interconnections in its interleaver, an LDPC factor graph has  $K \cdot \mu[d_v]/R$  random interconnections, where  $\mu[d_v]$  is the average number of CNs that each VN is connected to. Depending on the design of the LDPC code, an LDPC decoder may have several times more random interconnections than an equivalent turbo decoder, particularly at lower coding rates. For example, in the case of the WiFi LDPC codes, where we

have  $\text{mean}[d_v] = 3.5$  [42], the  $R = 1/2$  interconnection complexity is 7 times higher than that of a turbo decoder having the same information block length  $K$ . Since different components of an LDPC decoder have different irregular numbers of connections, it can be a significant challenge to implement flexible LDPC decoders that support various block lengths  $K$  and coding rates  $R$ , at a high throughput, as it will be demonstrated in Section IV-C. This flexibility is typically achieved by defining a different PCM for each supported combination of information block length  $K$  and coding rate  $R$ , when requiring the decoder to support a wide variety of PCMs.

However, in contrast to turbo decoders, LDPC decoders only have connectivity through the interleaver. More specifically, LDPC decoders do not suffer from the data dependencies that are imposed by the serial nature of the forward and backward recursions used by the Log-BCJR algorithm during turbo decoding. Owing to this, the designers of LDPC decoders are free to implement parallel processing to a wide variety of degrees and using a wide variety of techniques, including block parallel [43], row parallel [44] and fully parallel [45] arrangements. In analogy to the fully-serial and fully-parallel turbo decoding of Tables III and IV, LDPC decoding can be completed using the more-serial layered belief propagation schedule [46] or the extreme of the fully-parallel flooding schedule [45]. As in turbo decoding, a higher degree of parallelism leads to higher throughput, area and energy efficiency, but degraded error correction performance and flexibility, as it will be discussed in Section IV.

Furthermore, as shown in Figure 6, the complexity, throughput, latency, hardware efficiency and energy efficiency of LDPC decoders are degraded at low coding rates  $R$ , owing to two fundamental reasons. Firstly, the number of rows in an LDPC PCM grows as the coding rate  $R$  is reduced, which implies having a higher number of CNs requiring computation. Secondly, the number of columns in the PCM and the number of VNs in the factor graph is given by the encoded block length  $N$ , which dictates the input and output interface to the LDPC decoder, as shown in Figure 4(b). Since LDPC decoders must recover  $1/R$  encoded bits in order to decode each information bit, their information throughputs scale down proportionately with the coding rate  $R$ , as illustrated by the analogy of Figure 5a. This analogy shows that the information throughput of an LDPC decoder varies when the coding rate is adjusted, since this controls the specific fraction of the  $N$  recovered bits that correspond to information

bits. As the coding rate of an LDPC is increased, the dimensions of its PCM are reduced and its complexity  $C$  is decreased, as shown in Figure 6. However, when the coding rate  $R$  is fixed, increasing the information block length  $K$  will increase the dimensions of the PCM, hence increasing the computational complexity of the LDPC decoder, as shown in Figure 7. In general terms, it may be said that the complexity of an LDPC decoder is on the order of  $C = O(K/R)$ .

### C. Polar codes

While turbo and LDPC codes have been researched and developed over the last 20 years, polar codes [11] were not proposed until 2009. Owing to this, polar codes were not adopted in any standards or consumer devices before the control channel of 3GPP NR eMBB. This limited their scope to proof of concept demonstrators and academic publications.

During polar encoding, the  $K$  information bits are interleaved with  $(M - K)$  frozen bits, which have a fixed value of zero. The frozen bits are positioned according to a prescribed bit pattern, which should be optimised for each supported combination of  $K$  and  $R$ . Here, the number of frozen bits  $(M - K)$  should be chosen for ensuring that the number of bits in the resultant bit sequence is a power of two, yielding  $M = 2^{\lceil \log_2(N) \rceil}$ , where  $\lceil \cdot \rceil$  is the ceiling function. These bits are combined using XOR operations according to a Kronecker matrix [11], in analogy to Figure 4(c), which illustrates the regular recursive structure of a polar code. This process may be represented by the multiplication of the information bit vector  $\mathbf{x}$  by a generator matrix  $\mathbf{G}$  in GF(2), obtaining the encoded bit vector as  $\mathbf{U} = \mathbf{G} \cdot \mathbf{x}$ . For example, in the case of  $M = 8$ , the corresponding Kronecker matrix is given by <sup>1</sup>

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Following this,  $(M - N)$  of the resultant bits are punctured or shortened, in order to reduce the length of the resultant encoded block to  $N$  bits. Alternatively, repetition may be used for increasing  $N$  above  $M$  in

<sup>1</sup>Given the inclusion of frozen bits, the matrix becomes an  $[M \times M]$ -element matrix, rather than  $[N \times K]$  matrix.

cases, where  $N$  is slightly higher than a power of two. A polar decoder comprises a structured graph of VNs and CNs, all having a degree of no more than 3, as shown in Figure 4(c). At the start of the decoding process, the  $N$  LLRs provided by the demodulator are depunctured and forwarded to the inputs on the right of Figure 4(c). During the decoding process, the LLRs are propagated through the check nodes from right to the left of Figure 4(c). Here, each check node combines a pair of LLRs using the box-plus operation of Equation (2) or its min-sum approximation as shown in Equation (3). After the propagation of LLRs, hard decisions are made for each of the information bits on the left of Figure 4(c) in the order commencing from top to bottom. More specifically, these hard decisions are propagated from left to right, to the VNs, allowing them to pass LLRs to the CNs to propagate from right to left. Here, each VN either adds or subtracts an LLR from another, depending on the value of the bit propagated to it. In this way, each successive hard decision enables further hard decisions for subsequent bits. In order to guard against erroneous hard decisions affecting all subsequent decisions, a list of the  $L$  best hard decision based sequences can be maintained during the so-called Successive Cancellation List (SCL) based decoding process [47] where the best candidate is then output at the end of the process. Following this, the  $(M - K)$  frozen bits can be removed and the  $K$  recovered information bits can be output. Note that in the special case of  $L = 1$ , the SCL algorithm becomes the Successive Cancellation (SC) algorithm. In analogy to the fully-serial and fully-parallel turbo decoding schedules of Tables III and IV, polar decoding can be completed using the more-serial SC algorithm and its variants [48], or the fully-parallel belief propagation algorithm [49]. As in turbo and LDPC decoding, a higher degree of parallelism leads to a higher throughput, area and energy efficiency, at the cost of a degraded error correction performance and flexibility, as it will be discussed in Section IV.

Owing to the low degrees of its VNs and CNs, the computational complexity of a polar decoder is relatively low compared to that of an LDPC decoder, as determined by the list size  $L$ , which scales the complexity linearly, as shown in Figure 6. Like LDPC codes, the structure of polar codes scales with the encoded block length  $N$ , rather than with the information block length  $K$  as in turbo codes. Owing to this, the complexity and encoded throughput of a polar decoder typically remains constant, when the encoded block length  $N$  is kept constant, but the information block

length  $K$  is changed. However, since  $1/R$  encoded bits must be decoded in order to recover each information bit, the information throughput typically scales proportionately with the coding rate  $R$ , as illustrated by the stylized analogy of Figure 5c. Owing to this, as shown in Figure 6, the complexity, hardware efficiency and latency of polar decoders are degraded for lower coding rates and improved for higher coding rates, following a similar trend to that of LDPC decoders. On the same note, similarly to LDPC codes, the computational complexity of a polar code grows with the encoded block length  $N$ , as shown in Figure 6 and 7. More specifically, the dimensions of the polar code graph and hence the complexity grows according to the order of  $C = O\left(\frac{K}{R} \log \frac{K}{R}\right)$ . Note however that a simplified successive-cancellation list based algorithm [50] may be used for reducing the computational complexity of polar decoders having low coding rates, for example. More specifically, this technique reduces the complexity of processing the frozen bits, which become more prevalent at low coding rates  $R$ . Note however that fast simplified successive-cancellation list based algorithm [50] may be used for reducing the computational complexity of polar decoders having low coding rates, for example. More specifically, this technique reduces the complexity of processing the frozen bits, which become more prevalent at low coding rates  $R$ .

#### IV. COMPARISON OF TURBO, LDPC AND POLAR DECODERS

This section provides a comprehensive comparison of 100+ state-of-the-art ASIC implementations of turbo, LDPC and polar decoders. The key performance characteristics of a channel decoder are its processing latency, information throughput, error correction capability, flexibility, area efficiency and energy efficiency, as illustrated in Figure 8. When employed in a particular application, certain minimum performance requirements are imposed upon the information throughput (and on the resultant processing latency), error correction capability and flexibility of the channel decoder ASIC implementations. Within the constraints imposed by meeting these requirements, the design of a channel decoder ASIC should focus on optimising the area- and energy- efficiency, since they directly determine the implementation and running cost or the battery lifetime of basestations and user equipment. Motivated by this, the following subsections consider the information throughput, error correction capability and flexibility of the ASIC implementations in turn,

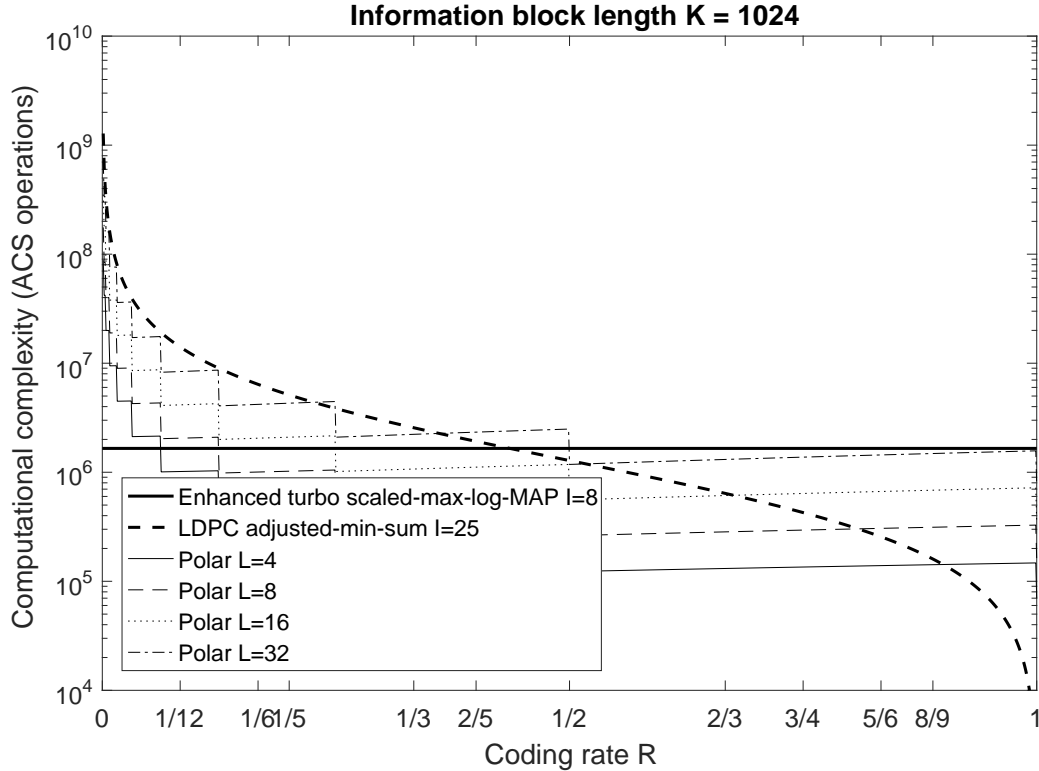


Fig. 6: Computational complexity versus coding rate  $R$  for various enhanced turbo, LDPC and polar decoders having information block lengths of  $K = 1024$ . Here the enhanced turbo code of [51] employs  $I = 8$  decoding iterations, the adjusted min-sum LDPC decoder of [52] employs  $I = 25$  iterations and the polar decoder of [53] employs list sizes of  $L \in \{4, 8, 16, 32\}$ .

each as functions of the area- and energy- efficiency. Each subsection includes plots, which are derived from the data presented in Table VI of the Appendix.

#### A. The information throughput vs. area- and energy-efficiency

Figure 9 characterises the relationship between the information throughput and area-efficiency of 100+ ASIC implementations of channel decoders, where the shape of the data points distinguishes the turbo, LDPC and polar decoders. Where a channel decoder supports multiple coding rates  $R$  and/or block sizes  $K$ , the corresponding information throughputs have been averaged in Figure 9. Furthermore, in order to present a fair comparison amongst ASICs implemented at different technology scales, the results of Figure 9 have all been scaled to 65 nm. Explicitly, 65 nm was chosen, since it is the standard technology scale that is closest to the average of the ASICs considered in this paper. Here, scaling is applied to the performance

characteristics of an ASIC implementation using a scaling factor, which is calculated as the ratio between the ASIC's technology scale and the target technology scale of 65 nm. The scaled information throughput, area and power consumption are obtained by multiplying the ASICs information throughput by the scaling factor [54], the inverse square of the scaling factor and the inverse of the scaling factor, respectively. Owing to this, the area-efficiency and energy-efficiency are proportional to the cube and the square of the scaling factor, respectively.

Figure 9 illustrates the difference between the information throughputs achieved by the turbo, LDPC and polar decoders. The considered turbo decoder implementations appear to have the largest portion of decoders associated with low information throughputs, which may be attributed to their low degrees of parallelism imposed by the data dependencies of the Log-BCJR algorithm, as discussed in Section III-A. Likewise, the parallelism of polar decoder implemen-

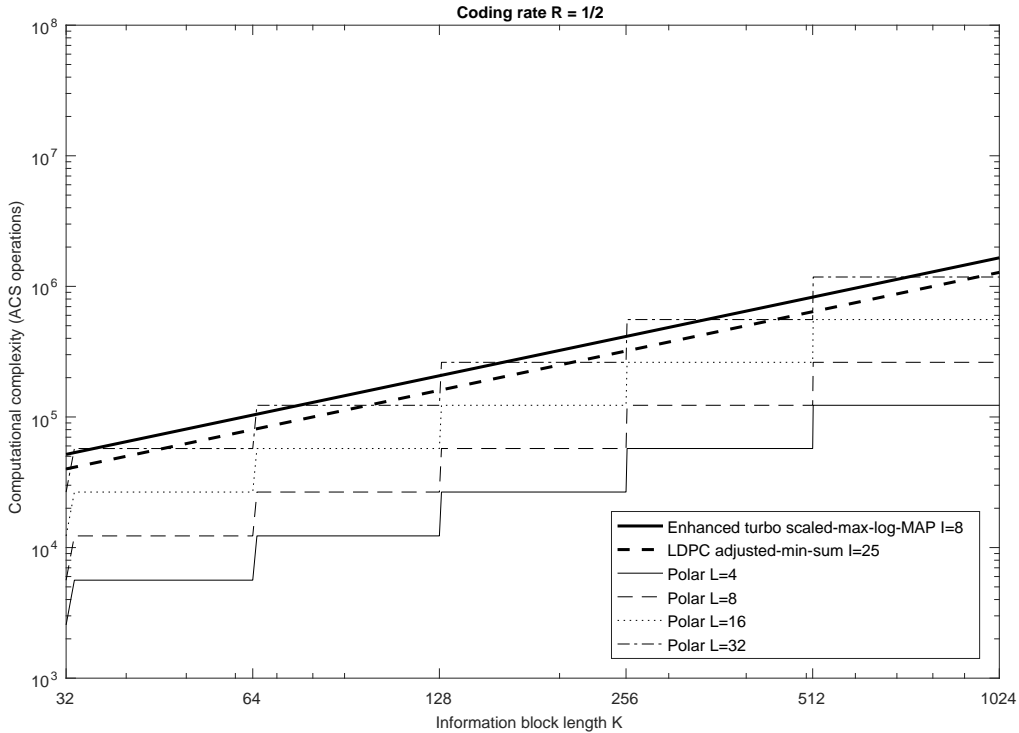


Fig. 7: Computational complexity versus block length  $K$  for various enhanced turbo, LDPC and polar decoders having coding rate  $R = 1/2$ . Here the enhanced turbo code of [51] employs  $I = 8$  decoding iterations, the adjusted min-sum LDPC decoder of [52] employs  $I = 25$  iterations and the polar decoder of [53] employs list sizes of  $L \in \{4, 8, 16, 32\}$ .

tations is typically limited, owing to the strict data dependencies of the SC and SCL decoding algorithms, as we discussed in Section III-C. Finally, the LDPC decoder implementations have the highest portion of decoders associated with high information throughputs, owing to the high degree of parallelism achieved by the min-sum algorithm, as discussed in Section III-B.

As mentioned in Section II-A, attaining a peak information throughput of 20 Gbps is a key requirement for 5G communication. As shown in Figure 9, the only turbo decoder that achieves the 20 Gbps information throughput is that of [37], which uses a fully-parallel architecture for the turbo decoding. Likewise, the only LDPC decoders that have reached information throughputs in excess of 20 Gbps have adopted fully parallel architectures. In particular, a decoded throughput of  $T = 52$  Gbps and a latency of  $0.045 \mu\text{s}$  have been demonstrated in [45] for the 10GBASE-T LDPC code, which has a block length of  $K = 1723$  bits and a

coding rate of  $R = 0.84$ . Likewise, the state-of-the-art LDPC ASIC implementation of [55] achieves not only a high information throughput, but also an outstanding energy efficiency of 987 bit/nJ. However, the LDPC decoders of [45] and [55] only support this single combination of block length  $K$  and coding rate  $R$ , which reflects the relatively inflexible nature of LDPC decoders in general and the rigidity of fully-parallel LDPC decoders in particular. The only polar decoders that achieve a decoded throughput in excess of 20 Gbps have managed this by unrolling hundreds of data blocks and pipelining their successive cancellation decoding through the same hardware [56], as indicated in Table VI. Using this approach, an extremely high throughput of 208 Gbps has been demonstrated for a coding rate of  $R = 1/2$ , although the latency experienced by each  $K = 1024$ -bit data block is  $3.21 \mu\text{s}$ . However, this pipelining approach severely limits the flexibility of the polar decoder. Furthermore, the SC technique for polar decoding results in a degraded error

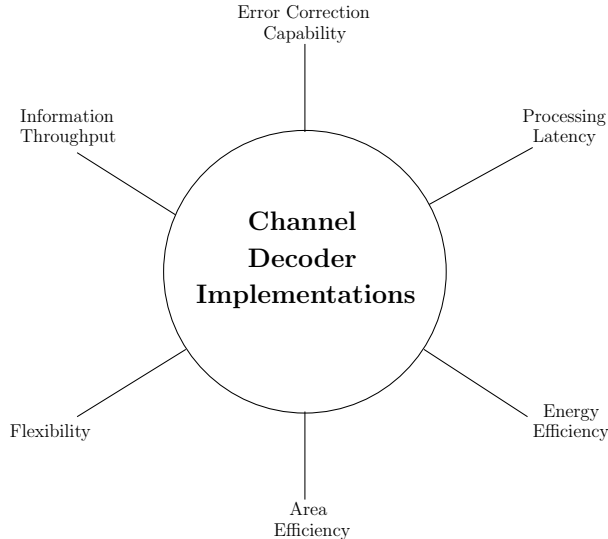


Fig. 8: Key characteristics of ASIC channel decoder implementations

correction capability, compared to the more complex SCL decoding technique, as it will be discussed in Section IV-B.

However, a high degree of parallel processing implies having a large chip area, and a high cost. For this reason, the key consideration in the selection of the 5G channel code is the area-efficiency, which quantifies the ratio of information throughput to chip area, as seen in Figure 9. At first sight, it may be expected that the area efficiency should remain constant upon varying the degree of parallel processing, since this scales both the information throughput and the area.

However, Figure 9 shows that decoders having higher information throughputs and degrees of parallel processing tend to have higher area efficiencies. This may be explained by the observation that while the datapath area of a decoder scales linearly with the parallelism, the memory and controller parts tend to increase slower than linearly. Owing to this, decoders having a high information throughput tend to have high area-efficiencies, that are dominated by the particular parts that scale with the grade of parallelism, such as the datapath. By contrast, decoders having a low information throughput tend to have low area-efficiencies that are dominated by the specific parts that do not scale with the grade of parallelism, such as the memory and controller. For similar reasons, the same trend is shown in Figure 9 in terms of power-efficiency, with higher information throughputs leading to higher power-efficiency.

However, there are some special cases that do not

follow the tendency observed above, such as the fully parallel turbo decoder implementation of [37]. Here, the number of clock cycles required per decoding iteration indeed scales down linearly with the grade of parallelism. However, the number of iterations required by a fully parallel turbo decoder is higher than that of a conventional turbo decoder, when aiming for achieving the same error correction capability. Owing to this, the achievable information throughput does not scale up linearly with the grade of parallelism, leading to reduced area and power efficiency.

#### B. Error correction capability vs. area- and energy-efficiency

As described in Section II-C, there is a 5G requirement to facilitate reliable error correction at channel Signal to Noise Ratios (SNRs) per bit ( $E_b/N_0$ ) that are as close as possible to the channel capacity bound, where a BLER of  $10^{-2}$  is targeted as a complement to HARQ in eMBB applications and  $10^{-5}$  is targeted for URLLC applications. Figure 10 characterises the discrepancy between the channel capacity  $E_b/N_0$  bound and the specific  $E_b/N_0$  value, where a Bit Error Rate (BER) of  $10^{-4}$  is obtained for various channel decoder implementations, when using Binary Phase Shift Keying (BPSK) modulation for communication over an Additive White Gaussian Noise (AWGN) channel. This approach is adopted, since nearly all of the papers considered present BER plots, rather than BLER plots, and consider BPSK and AWGN. Figure 10 also presents the scaled area-efficiency,

while the shape and color of the data points indicate the type of decoder and the scaled energy-efficiency. Note that the legend of Figure 10 is inherited from Figure 9, which is provided below this figure again for convenience. Note that fewer data points are presented in Figure 10 than were presented in Figure 9, since many papers do not characterise the error correction capability of their proposed decoder.

Rather than comparing the minimum  $E_b/N_0$  value required for each channel decoder implementation to achieve the target BER, Figure 10 compares the discrepancy between this  $E_b/N_0$  value and the channel capacity  $E_b/N_0$  bound, which gives a fairer comparison of the error correction capability of various channel decoder implementations having different coding rates  $R$ . In order to illustrate this, Figure 12 plots the Discrete-input Continuous-output Memoryless Channel (DCMC) capacity associated with using BPSK for communication over an AWGN channel as a function of  $E_b/N_0$ . This is compared with points corresponding to each channel decoder implementation, which plot the spectral efficiency of each decoder versus the minimum  $E_b/N_0$  value that it requires to achieve a BER of  $10^{-4}$ , when using BPSK to communicate over an AWGN channel. If ideal Nyquist pulse shaping having zero excess-bandwidth is employed, the spectral efficiency of a decoder is numerically equal to its effective throughput of,  $\eta = R \times \log_2(M)$  [57], where  $R$  is the coding rate of the decoder and  $M = 2$  is the number of constellation points employed by BPSK, resulting in  $\eta = R$ . Note that while many of the papers considered present flexible channel decoders that support different coding rates and hence different spectral efficiencies, they typically only provide hardware performance results for a single coding rate and spectral efficiency, as plotted in Figure 12.

Figure 10 shows that turbo decoders tend to have the best error correction capability, with most of them exhibiting less than 1.5 dB discrepancies from the channel capacity. This may be explained by the long block lengths  $K$ , comprising thousands of information bits that are supported in the Long Term Evolution (LTE) turbo code. This enables stronger error correction than is possible when using blocks comprising only hundreds of information bits, as is typical for the LDPC and polar decoders considered. Owing to this, most LDPC and polar decoders exhibit more than 1.5 dB discrepancy from channel capacity, which may even exceed 2.5 dB. However, there are some specific LDPC and polar decoder implementations that offer similar error correction capability to turbo decoders.

The polar decoder of [48] has a 1.5 dB discrepancy from the channel capacity, as a benefit of using the SCL decoding algorithm with a list size of  $L = 4$ , rather than the SC algorithm. Outside of the ASIC implementation literature, it has been shown [87] that polar decoders employing large list sizes can achieve superior error correction capability to LDPC and turbo decoders employing large numbers of iterations, particularly at short block lengths  $K$ . Furthermore, the LDPC decoder of [63] and several other similar LDPC decoder implementations also exhibit reliable error correction capability as well as extremely high area- and energy-efficiency. These LDPC decoders adopt fully parallel architectures, in order to achieve the high information throughputs supported by the 10GBASE-T Ethernet standard. This allows these decoders to perform a high number of decoding iterations, hence enabling reliable error correction. Note that state-of-the-art LDPC and polar code constructions have been adopted by the 3GPP 5G New Radio standard. It is expected that the error correction performance loss compared to turbo codes will be soon eliminated when implementations of the 5G LDPC and polar codes [42] become available for comparison. Alternatively, the non-binary LDPC ASIC implementation of [66] has largely reduced the error correction performance gap from turbo decoders.

Figure 10 suggests that the error correction capability is not directly correlated with the area- and energy-efficiency, as inferred from the various decoders considered here. However, for a particular decoder, it may be expected that performing more iterations or using a larger SCL list size would improve the error correction capability, albeit at the cost of degrading the information throughput and hence both the area- and energy-efficiency.

### C. Reconfiguration flexibility vs. area- and energy-efficiency

As described in Section II-D, 5G NR requires the channel decoder to support a wide variety of information block lengths  $K$ , as well as coding rates  $R = K/N$ . In this section, we quantify the flexibility of each channel decoder by the number of supported combinations of block length  $K$  and coding rate  $R$ . Figure 11 characterises the flexibility, scaled average area-efficiency and energy-efficiency that is achieved by each turbo, LDPC and polar decoder ASIC considered. In the case of the turbo decoders, the number of information block lengths  $K$  supported is determined by the number of interleavers supported, which is 188



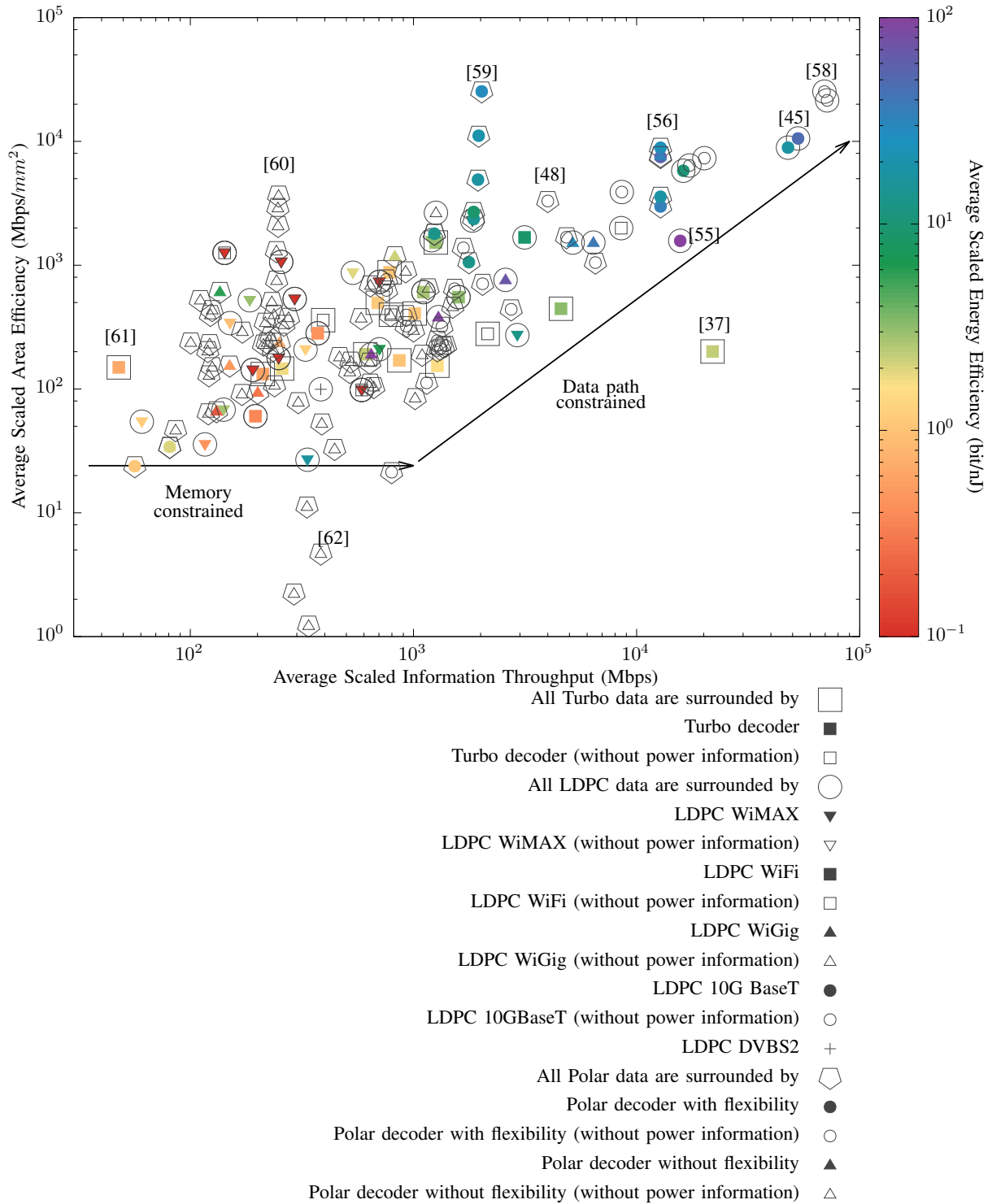


Fig. 9: Area and energy efficiency versus information throughput for various channel decoder implementations.

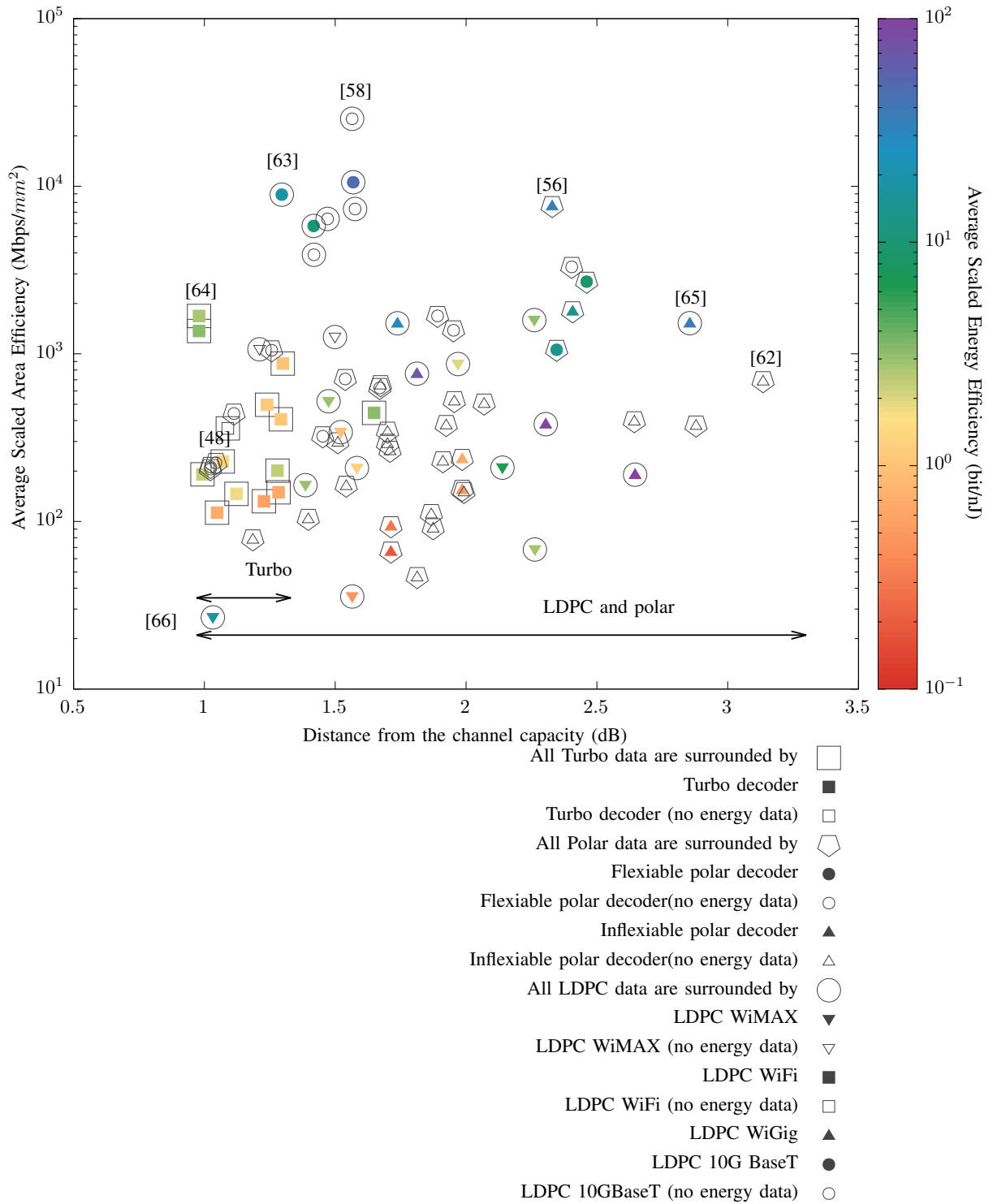


Fig. 10: Area-efficiency ( $Mbps/mm^2$ ) versus the discrepancy between the channel capacity  $E_b/N_0$  bound and the  $E_b/N_0$  value where a BER of  $10^{-4}$  is obtained for various channel decoder implementations.

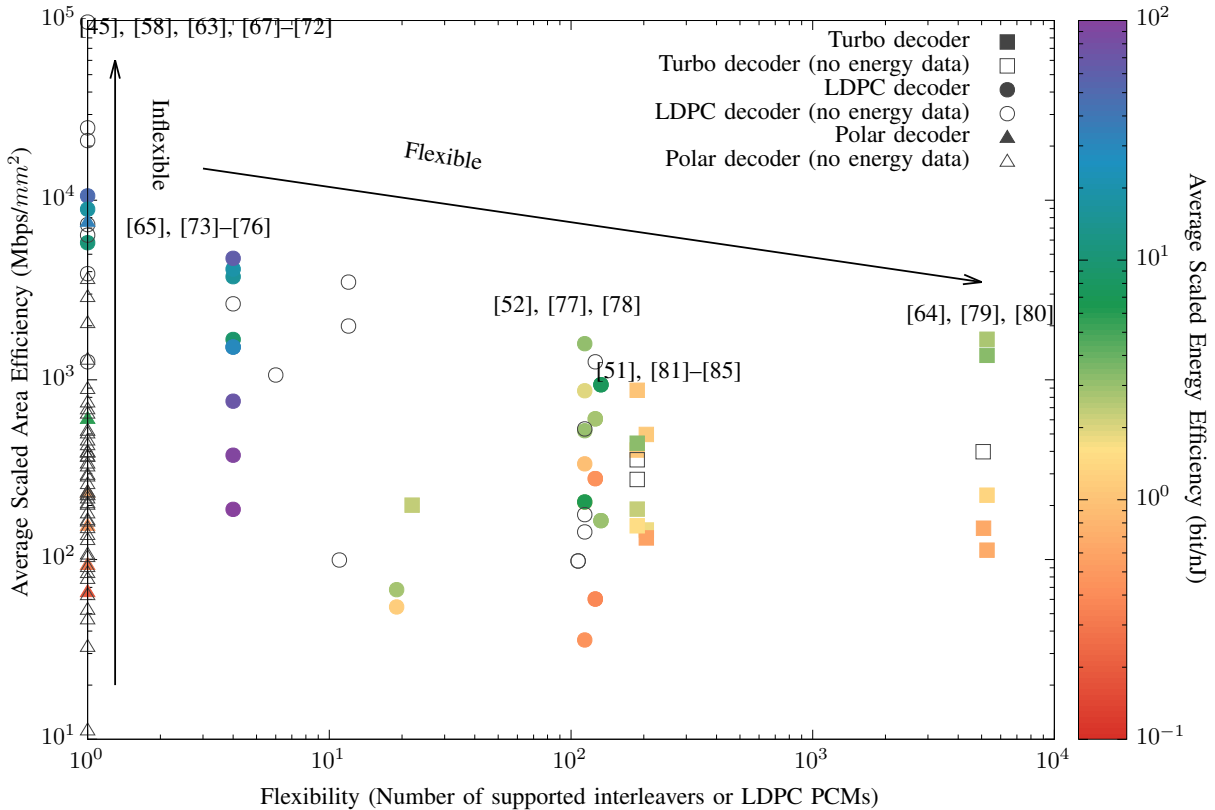


Fig. 11: Area-efficiency ( $Mbps/mm^2$ ) versus the reconfiguration flexibility between state-of-the-art turbo, LDPC and inflexible polar decoder ASICs when scaled to 65 nm.

for the LTE turbo code. As we mentioned in Section III-A, turbo codes achieve single bit granularity for the encoded block length  $N$  using puncturing or repetition, and hence they support a large number of coding rates  $R$ . Rather than quantifying the very large number of combinations of  $K$  and  $R$  supported by the turbo decoders, Figure 11 quantifies only the number of supported information block lengths  $K$  instead. Hence, the flexibility of the turbo decoders considered is even better than that quantified in Figure 11. Note that the 5G LDPC and polar codes [42] flexibly support single-bit granularity for both the information- and encoded-block length, although published implementations are not currently available in the open literature for consideration in this paper.

By contrast, LDPC codes used in the existing standards require a different PCM for each supported combination of block length  $K$  and coding rate  $R$ , hence resulting in the relatively low flexibilities observed in Figure 11. In principle, polar codes can have single bit granularity of the block length  $K$  and coding rate  $R$ , using only a single frozen bit selection sequence.

However, this high grade of flexibility has not yet been demonstrated in any existing ASIC implementations. Indeed, the majority of polar decoder ASICs in the literature support only a single combination of block length  $K$  and coding rate  $R$ , as shown in Figure 11. This may be because many of the existing polar decoder ASIC implementations in the literature have focused their attention on the implementation of the SC or SCL algorithm, without any consideration of the circuits required for frozen bit insertion or rate matching. This is because these circuits can operate relatively independently of the core decoder, which is reminiscent of the rate matching circuit of a turbo decoder.

As shown in Figure 11, the turbo decoder implementations considered have the highest flexibility among the three types of channel decoders, as discussed above. The LDPC decoder ASICs that achieve the highest area-efficiencies [45], [58], [63], [67]–[72] typically support only a single combination of block length  $K$  and coding rate  $R$ . This is because these LDPC decoders adopt fully-parallel architec-

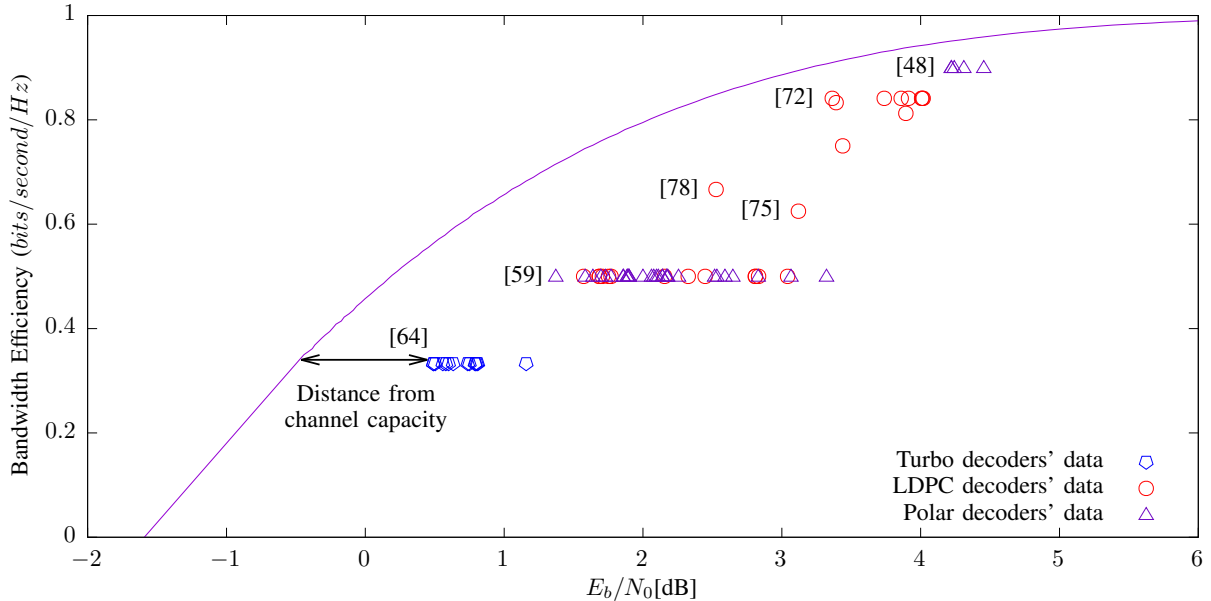


Fig. 12: Bandwidth efficiency versus minimum  $E_b/N_0$  value required to achieve a BER of  $10^{-4}$  for various channel decoder implementations.

tures, which only support a single PCM and a high coding rate  $R$ . More specifically, these ASICs are physically laid out in a manner that resembles the factor graph structure [88] described by the PCM, using hard-wired connections between registers and dedicated computational hardware for each VN and CN. While this approach prevents flexibility, it allows the LDPC decoding process to be completed using a minimal number of clock cycles and without the requirement for additional memory, switchable interconnections or a complex controller. Owing to this lack of flexibility, it is the LDPC decoders of [45], [58], [63], [67]–[72] that offer the best throughputs, latencies, hardware-efficiencies and energy-efficiencies among all of the channel decoder ASICs, as described above. By contrast, the flexible LDPC decoder ASICs of Figure 11 support more than one PCM by employing partially-parallel architectures, such as the row parallel [44] or block parallel [43] architecture. More specifically, these ASICs employ a bank of computational hardware, which can be flexibly reused at different times to perform the processing associated with different VNs and CNs of different PCMs. However, this approach requires the employment of additional memory, switchable interconnections and a complex controller. As exemplified in Figure 13, these components typically occupy around 75% of the chip area in the case of LDPC decoders that support

around 100 PCMs, as may be required for meeting the flexibility requirement of 5G. Owing to this, it is these additional hardware components that dominate the throughput, latency, hardware-efficiency and energy-efficiency of the resultant ASICs, particularly as the number of supported PCMs is increased, as shown in Figure 11. This may be attributed to the irregular structures and high interconnection complexity, as described in Section III. By contrast, turbo decoders have regular structures associated with significantly reduced interconnection complexities. As a result, the corresponding hardware typically occupies only 30% of the chip area, as exemplified in Figure 13. Owing to this, partially-parallel turbo decoders typically offer superior flexibilities, information throughputs, latencies, hardware efficiencies and energy efficiencies than partially-parallel LDPC decoders, as shown in Figure 11.

#### D. Summary

Having compared numerous turbo, LDPC and polar decoder implementations in terms of their information throughput, error correction performance, flexibility, area efficiency and energy efficiency, some relative advantages and disadvantages can be observed. The ASIC implementations of turbo decoders offer the best error correction performance owing to their natural support for low coding rates and long block lengths,

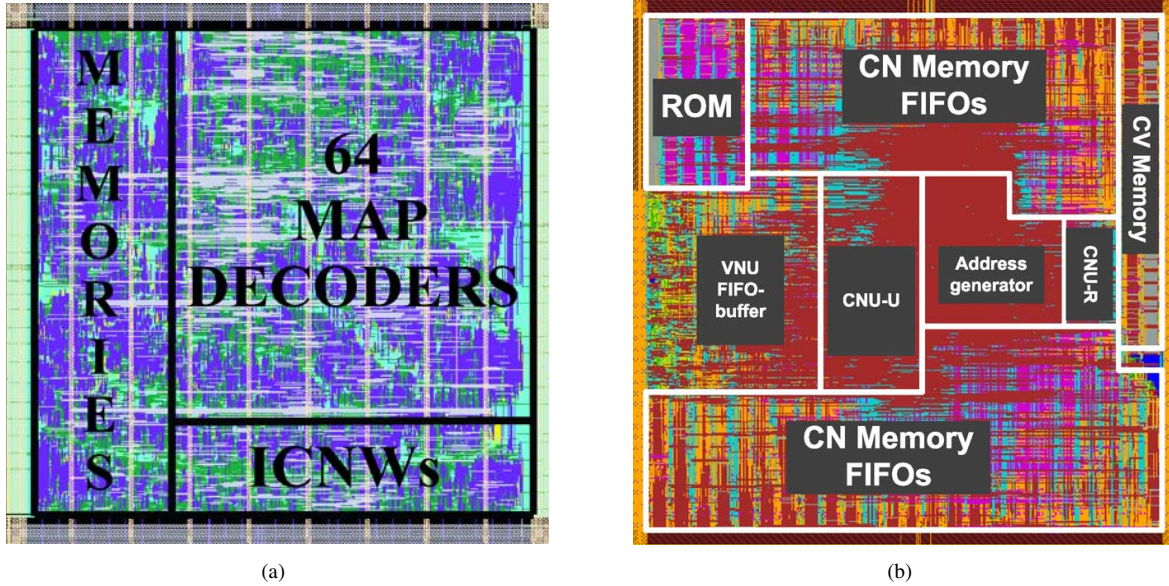


Fig. 13: (a) ASIC layout of the partially-parallel turbo decoder of [86], in which the non-computational components (labeled ‘MEMORIES’ and ‘ICNWs’) occupy 30% of the area. (b) ASIC layout of the partially-parallel LDPC decoder of [52], in which the non-computational components (labeled ‘ROM’, ‘CN Memory FIFOs’, ‘CV Memory’, ‘VNU FIFO-buffer’, ‘Address generator’ and ‘CN Memory FIFOs’) occupy 75% of the area.

as exemplified in [64]. However, limited by the data dependencies of the Log-BCJR algorithm, most turbo decoder ASIC implementations have low area- and power-efficiencies compared to LDPC and polar decoders having the same level of information throughput. ASIC implementations of LDPC decoders have been designed for conformance with various standards. For example, the LDPC 10G-BaseT decoders of [45] and [58], both high information throughput, as well as area- and energy-efficiencies, but no flexibility. Polar decoder ASIC implementations offer outstanding error correction performance for short block lengths, offering about 1dB or more coding gain relative to LDPC decoders having short block lengths. However, polar decoders have relatively limited capability to achieve high information throughput, especially at high area- and energy-efficiencies, unless low list sizes resulting in a degraded error correction capability are adopted, as exemplified in [59]. Additionally, polar codes potentially explicit a high flexibility, although this has not been demonstrated in the open literature as yet. In summary, these types of channel decoder implementations have relative advantages and disadvantages, which must be considered when selecting a channel code for a practical application.

## V. DESIGN RECOMMENDATIONS AND CONCLUSIONS

In closing we present an overview of our design recommendations for future ASIC channel decoder implementations. We commenced by discussing the channel coding requirements of 5G, which include high throughput, low latency, strong error correction capability and low implementation complexity. We have also highlighted the challenging 5G requirements imposed on the channel code to flexibly support a wide variety of block lengths  $K$  and coding rates  $R$ , in order to ensure the efficient exploitation of the bandwidth available and to address the challenging applications of 5G. We have provided comprehensive discussions on the extent to which the 5G requirements can be met by the turbo, LDPC and polar codes. In order to support our discussions, we also presented several plots to characterise a diverse set of as many as 110 ASIC implementations of turbo, LDPC and polar decoders. We have characterised the fundamental trade-offs between the various performance characteristics of channel decoder ASICs. Furthermore, we have demonstrated that the overall implementation complexity of a channel code depends not only on its computational complexity, but also on its interconnection complexity

TABLE V: Comparison between turbo, LDPC and polar decoders included in this paper.

Parameter	Turbo decoder	LDPC Decoder	Polar Decoder
<b>Computational complexity</b>	Higher for most coding rates. Low at low coding rates	Lower for most coding rates. High at low coding rates	Lower for most coding rates
<b>Information throughput</b>	Low. Enhanced by fully-parallel structure	High for high coding rates and parallel structure	High for pipeline structure
<b>Area efficiency</b>	Low for most coding rate. High at low coding rates.	High for most coding rates. Low at low coding rates.	High for most coding rates.
<b>Energy efficiency</b>	Low for most coding rates. High at low coding rates.	High for most coding rate. Low at low coding rates.	High for most coding rates.
<b>Error correction performance</b>	Similar at long block lengths.	Similar at long block lengths.	Superior for short block lengths.
<b>Flexibility</b>	High flexibility	Partial flexibility	Very high potential

and its inherent flexibility.

Our findings are summarised by the comparison of the three types of channel decoder implementations presented in Table V. We observed that most of existing turbo decoders offer limited information throughput owing to the serial nature of the Log-BCJR algorithm. This leads to lower area- and energy-efficiencies at most of the practical coding rates  $R$ , although they maintain these efficiencies at low coding rates, in contrast to the family of LDPC and polar decoders. However, among the decoder ASICs considered in this paper, the turbo decoders have the best error correction performance and offer the highest flexibility in terms of supporting various combinations of block lengths and coding rates. By contrast, LDPC and polar decoder implementations have both a high information throughput as well as high area- and energy-efficiencies for most practical coding rates. However, the state-of-the-art in LDPC and polar coding has recently been substantially advanced during the design of 3GPP 5G New Radio. More specifically, the New Radio implementations of both codes enable flexibility both in terms of the block length and coding rate with single bit granularity. Furthermore, the error correction performance of both codes has been improved relative to previous implementations. While polar codes offer the best error correction performance at short block lengths, turbo, LDPC and polar codes offer similar performance at block lengths above 1000 bits. By contrast, most of the LDPC and polar decoder implementations considered in this paper support only shorter block lengths of up to 1024 bits, where their error correction ability is weaker. While it may be expected that flexible LDPC and polar decoder implementations of 3GPP 5G New Radio codes will emerge in the future, many of the ASIC implementations that were available at the time of writing this paper do not meet the error correction performance

and flexibility requirements of 5G NR. However, the ASIC implementations considered in this paper do not meet the error correction performance and flexibility requirements of 5G NR. In the case of the LDPC decoders, this may be explained by the limited error correction performance and flexibility of the LDPC codes used in the existing standards. Furthermore, despite their great potential, the implementation of polar decoders having full flexibility has not been demonstrated in the previous ASIC implementations. Likewise, at the time of writing only low list sizes have been considered for the ASIC implementation of SCL polar decoders, while an improved error correction performance can only be expected at higher list sizes.

Owing to the lack of some important details in some publications, the comparison of the many ASIC channel decoder implementations considered in this paper has been challenging. Therefore, we recommend valued Colleagues that future ASIC channel decoder designers include the following important details in their implementation-oriented papers:

- Provide values for every parameter of the algorithm and implementation, especially the most important parameters considered in Table VI.
- State both the average and the worst-case processing latency of the decoder for a variety of block lengths and coding rates, or provide an approximate formulaic relationship between the information throughput and processing latency.
- Provide BLER plots and tabulate the  $E_b/N_0$  value that is required to achieve a BLER of  $10^{-2}$  by the decoder, for a variety of block lengths and coding rates. Identify the type of simulated channel and modulation scheme, which should preferably be BPSK modulation for transmission over an AWGN channel, since this allows direct comparison with the majority of previous publications.

- Provide the power- or energy-consumption for a variety of block lengths and coding rates.
- Quantify the flexibility of the decoder in terms of the number of supported block length and coding rate combinations.

There is still a significant amount of further work that can be completed on the implementation of channel decoders. In particular, there is a need for further research on flexible, high-performance, high-efficiency implementations of turbo, LDPC and polar decoders. For example, the hardware implementation of irregular turbo decoding [89] has not yet been demonstrated, which would improve the attainable error correction capability. Likewise, the performance of LDPC decoders can be significantly enhanced by invoking informed dynamic scheduling [90]. However, this benefit comes at the cost of increased implementation complexity owing to its adaptive decoding schedules. Polar decoding can be enhanced by implementing Soft-Input Soft-Output (SISO) decoding [91], which would facilitate ASIC-based turbo detection for the first time for polar codes. Likewise, the ASIC implementation of HARQ [92] for polar decoding is another challenge that remains unsolved at the time of writing. It is not possible to say that any one type of channel code is superior to any other, since all codes have different advantages and disadvantages.

## REFERENCES

- [1] Y. Hawwar, E. Farag, S. Vanakayala, R. Pauls, X. Yang, S. Subramanian, P. Sadhanala, L. Yang, B. Wang, Z. Li, H. Chen, Z. Lu, D. Clark, T. Fosket, P. Mallela, M. Shelton, D. Laurens, T. Salaun, L. Gougeon, N. Aubourg, H. Morvan, N. L. Henaff, G. Prat, F. Charles, C. Creach, Y. Calvez, and P. Butel, "3G UMTS Wireless System Physical Layer: Baseband Processing Hardware Implementation Perspective," *IEEE Communications Magazine*, vol. 44, no. 9, pp. 52–58, Sept 2006.
- [2] Z. Shen, A. Papasakellariou, J. Montojo, D. Gerstenberger, and F. Xu, "Overview of 3GPP LTE-Advanced Carrier Aggregation for 4G Wireless Communications," *IEEE Communications Magazine*, vol. 50, no. 2, pp. 122–130, February 2012.
- [3] C. E. Shannon, "The mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, Jul 1948.
- [4] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes. 1," in *Communications, 1993. ICC '93 Geneva. Technical Program, Conference Record, IEEE International Conference on*, vol. 2, May 1993, pp. 1064–1070 vol.2.
- [5] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 32, no. 18, pp. 457–458, Aug 1996.
- [6] *IEEE 802.11-2012 Standard for Information Technology - Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY)*, 2012.
- [7] K. H. Teo, Z. Tao, and J. Zhang, "The Mobile Broadband WiMAX Standard [Standards in a Nutshell]," *IEEE Signal Processing Magazine*, vol. 24, no. 5, pp. 144–148, Sept 2007.
- [8] C. J. Hansen, "WiGiG: Multi-Gigabit Wireless Communications in the 60 GHz Band," *IEEE Wireless Communications*, vol. 18, no. 6, pp. 6–7, December 2011.
- [9] *ETSI EN 302 307-1 Digital Video Broadcasting (DVB); Second Generation Framing Structure, Channel Coding and Modulation Systems for Broadcasting, Interactive Services, News Gathering and other Broadband Satellite Applications; Part 1: DVB-S2, V1.4.1 ed.*, 2014.
- [10] J. R. Westra, J. Mulder, Y. Ke, D. Vecchi, X. Liu, E. Arslan, J. Wan, Q. Zhang, S. Wang, F. van der Goes, and K. Bult, "Design Considerations for Low-power Analog Front Ends in Full-Duplex 10GBASE-T Transceivers," in *Proceedings of the IEEE 2014 Custom Integrated Circuits Conference*, Sept 2014, pp. 1–8.
- [11] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inform. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul 2009.
- [12] K. Mallinson, "The path to 5G: as much evolution as revolution," 2016. [Online]. Available: <http://3gpp.org/news-events/3gpp-news/1774-5g-wisearbour>
- [13] F.-L. Luo and C. Zhang, *5G Standard Development: Technology and Roadmap*. Wiley-IEEE Press, 2016, pp. 616–. [Online]. Available: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7572796>
- [14] H. Mercier, V. K. Bhargava, and V. Tarokh, "A survey of error-correcting codes for channels with symbol synchronization errors," *IEEE Communications Surveys Tutorials*, vol. 12, no. 1, pp. 87–96, First 2010.
- [15] C. Roth, A. Cevrero, C. Studer, Y. Leblebici, and A. Burg, "Area, throughput, and energy-efficiency trade-offs in the VLSI implementation of LDPC decoders," in *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, May 2011, pp. 1772–1775.
- [16] Nasruminallah and L. Hanzo, "Near-Capacity H.264 Multimedia Communications Using Iterative Joint Source-Channel Decoding," *IEEE Communications Surveys Tutorials*, vol. 14, no. 2, pp. 538–564, Second 2012.
- [17] H. Chen, R. G. Maunder, and L. Hanzo, "A Survey and Tutorial on Low-Complexity Turbo Coding Techniques and a Holistic Hybrid ARQ Design Example," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 1546–1566, November 2013. [Online]. Available: <https://eprints.soton.ac.uk/347846/>
- [18] Y. Fang, G. Bi, Y. L. Guan, and F. C. M. Lau, "A Survey on Protograph LDPC Codes and Their Applications," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 1989–2016, Fourthquarter 2015.
- [19] E. Ankan, N. ul Hassan, M. Lentmaier, G. Montorsi, and J. Sayir, "Challenges and some new directions in channel coding," *Journal of Communications and Networks*, vol. 17, no. 4, pp. 328–338, Aug 2015.
- [20] M. F. Brejza, L. Li, R. G. Maunder, B. Al-Hashimi, C. Berrou, and L. Hanzo, "20 years of turbo coding and energy-aware design guidelines for energy-constrained wireless applications," *IEEE Communications Surveys & Tutorials*, pp. 1–21, June 2015. [Online]. Available: <https://eprints.soton.ac.uk/378161/>
- [21] P. Hailes, L. Xu, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "A Survey of FPGA-based LDPC Decoders," *IEEE Commun. Surveys Tutorials*, vol. 18, no. 2, pp. 1098–1122, Second quarter 2016.
- [22] H. B. Thameur, B. L. Gal, N. Khouja, F. Tlili, and C. Jego, "A survey on decoding schedules of LDPC convolutional codes and associated hardware architectures," in *2017 IEEE Symposium on Computers and Communications (ISCC)*, July 2017, pp. 898–905.
- [23] Samsung, "R1-162181 Channel Coding for 5G New Radio Interface," in *3GPP TSG RAN WG1 #84bis*, Apr. 2016.
- [24] S. P. L.-E. T. Wehn Norbert, Scholl Stefan and M. Alles, *Challenges and Limitations for Very High Throughput Decoder Architectures for Soft-Decoding*. Cham, Switzerland: Springer International Publishing, 2015, pp. 7–31.
- [25] R. S. Groups, "Minimum Requirements Related to Technical Performance for IMT-2020 Radio Interface(s)," February 2017.
- [26] Samsung, "R1-1609073 Discussion on Latency of Channel Codes for NR," in *3GPP TSG RAN WG1 #86*, Oct. 2016.
- [27] A. Nimbalkar, Y. Blankenship, B. Classon, and T. K. Blankenship, "ARP and QPP Interleavers for LTE Turbo Coding," in *Wireless communications and networking conference, 2008. WCNC 2008. IEEE*. IEEE, 2008, pp. 1032–1037.
- [28] J. Hamorsky, U. Wachsmann, J. B. Huber, and A. Cizmar, "Hybrid Automatic Repeat Request Scheme with Turbo Codes," *Brest, France, International Symposium on Turbo Codes*, pp. 247–250, Sept 1997.
- [29] Y. D. Beyene, R. Jantti, K. Ruttik, and S. Iraj, "On the performance of narrow-band internet of things (NB-IoT)," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, March 2017, pp. 1–6.
- [30] J. P. Woodard and L. Hanzo, *Comparative Study of Turbo Decoding Techniques: an Overview*, Nov 2000, vol. 49, no. 6, pp. 2208–2233.
- [31] R. Johannesson and K. S. Zigangirov, *Decoding of Convolutional Codes*. Wiley-IEEE Press, 2015, pp. 550–. [Online]. Available: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7199872>
- [32] R. Bhowmuth and H. Xu, "Repeat-punctured turbo trellis-coded modulation," in *AFRICON, 2011*, Sept 2011, pp. 1–6.
- [33] T. Kailath, "A general likelihood-ratio formula for random signals in Gaussian noise," *IEEE Transactions on Information Theory*, vol. 15, no. 3, pp. 350–361, May 1969.



- [34] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate (Corresp.)," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, Mar 1974.
- [35] S. Yoon and Y. Bar-Ness, "A parallel MAP algorithm for low latency turbo decoding," *IEEE Communications Letters*, vol. 6, no. 7, pp. 288–290, July 2002.
- [36] T. Inseher, F. Kienle, C. Weis, and N. Wehn, "A 2.15Gbit/s Turbo Code Decoder for LTE Advanced Base Station Applications," in *2012 7th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, Aug 2012, pp. 21–25.
- [37] 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*, vol. 4, pp. 323–346, 2016.
- [38] *ETSI TS 136 212 LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and Channel Coding*, V13.1.0 ed., 2016.
- [39] *IEEE 802.16-2012 Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Broadband Wireless Access Systems*, 2012.
- [40] *IEEE 802.3an-2006 Specific requirements Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications Amendment 1: Physical Layer and Management Parameters for 10 Gb/s Operation*, 2006.
- [41] R. G. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, Jan 1962.
- [42] R. G. Maunder, "The 5G Channel Code Contenders," *AccelerComm White Paper*, Aug. 2016. [Online]. Available: <http://www.accelercomm.com>
- [43] H. T. Pham, S. Ajaz, and H. Lee, "Parallel Block-Layered Nonbinary QC-LDPC Decoding on GPU," in *2015 IEEE Workshop on Signal Processing Systems (SiPS)*, Oct 2015, pp. 1–6.
- [44] K. Shimizu, T. Ishikawa, N. Togawa, T. Ikenaga, and S. Goto, "Partially-Parallel LDPC Decoder Based on High-Efficiency Message-Passing Algorithm," in *2005 International Conference on Computer Design*, Oct 2005, pp. 503–510.
- [45] C. C. Cheng, J. D. Yang, H. C. Lee, C. H. Yang, and Y. L. Ueng, "A Fully Parallel LDPC Decoder Architecture Using Probabilistic Min-Sum Algorithm for High-Throughput Applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 9, pp. 2738–2746, Sept 2014.
- [46] D. E. Hocevar, "A reduced complexity decoder architecture via layered decoding of ldpc codes," in *IEEE Workshop on Signal Processing Systems, 2004. SIPS 2004.*, Oct 2004, pp. 107–112.
- [47] B. Li, H. Shen, and D. Tse, "An Adaptive Successive Cancellation List Decoder for Polar Codes with Cyclic Redundancy Check," *IEEE Communications Letters*, vol. 16, no. 12, pp. 2044–2047, December 2012.
- [48] C. Xiong, J. Lin, and Z. Yan, "A multimode area-efficient SCL polar decoder," *IEEE Trans. VLSI Syst.*, 2016.
- [49] Y. S. Park, Y. Tao, S. Sun, and Z. Zhang, "A 4.68Gb/s belief propagation polar decoder with bit-splitting register file," in *Proc. Symp. VLSI Circuits*, June 2014.
- [50] S. A. Hashemi, C. Condo, and W. J. Gross, "Fast and flexible successive-cancellation list decoders for polar codes," *IEEE Transactions on Signal Processing*, vol. 65, no. 21, pp. 5756–5769, Nov 2017.
- [51] C. Studer, C. Benkeser, S. Belfanti, and Q. Huang, "Design and Implementation of a Parallel Turbo-Decoder ASIC for 3GPP-LTE," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 8–17, Jan 2011.
- [52] 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 Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 3, pp. 743–756, 2013.
- [53] Huawei, "R1-1608865 Design aspects of Polar Code and LDPC for NR," in *3GPP TSG RAN WG1 #86*, Oct. 2016.
- [54] K. C. Saraswat and F. Mohammadi, "Effect of Scaling of Interconnections on the Time Delay of VLSI Circuits," *IEEE Journal of Solid-State Circuits*, vol. 17, no. 2, pp. 275–280, Apr 1982.
- [55] K. Cushon, P. Larsson-Edefors, and P. Andrekson, "A High-Throughput Low-Power Soft Bit-Flipping LDPC Decoder in 28 nm FD-SOI," in *ESSCIRC 2018 - IEEE 44th European Solid State Circuits Conference (ESSCIRC)*, Sept 2018, pp. 102–105.
- [56] P. Giard, G. Sarkis, C. Thibeault, and W. J. Gross, "Multi-mode unrolled architectures for polar decoders," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 9, pp. 1443–1453, 2016.
- [57] S. L. Goff, A. Glavieux, and C. Berrou, "Turbo-codes and high spectral efficiency modulation," in *Communications, 1994. ICC '94, SUPERCOMM/ICC '94, Conference Record, 'Serving Humanity Through Communications.'* *IEEE International Conference on*, May 1994, pp. 645–649 vol.2.
- [58] S. S. Tehrani, A. Naderi, G.-A. Kamendje, S. Hemati, S. Manmor, and W. J. Gross, "Majority-based tracking forecast memories for stochastic LDPC decoding," *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4883–4896, 2010.
- [59] O. Dizdar and E. Arıkan, "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*, vol. 63, no. 3, pp. 436–447, 2016.
- [60] Y. Fan and C.-y. Tsui, "An efficient partial-sum network architecture for semi-parallel polar codes decoder implementation," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3165–3179, 2014.
- [61] C.-H. Lin, C.-Y. Chen, and A.-Y. Wu, "Area-efficient scalable MAP processor design for high-throughput multistandard convolutional turbo decoding," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 2, pp. 305–318, 2011.
- [62] J. Sha, X. Liu, Z. Wang, and X. Zeng, "A memory efficient belief propagation decoder for polar codes," *China Communications*, vol. 12, no. 5, pp. 34–41, 2015.
- [63] Q. Zhang, Y. Chen, D. Wu, X. Zeng, and Y.-l. Ueng, "Convergence-optimized variable node structure for stochastic LDPC decoder," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 6535–6539.
- [64] Q. Yang, X. Zhou, G. E. Sobelman, and X. Li, "Network-on-chip for turbo decoders," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 1, pp. 338–342, 2016.
- [65] M. Li, J.-W. Weijers, V. Derudder, I. Vos, M. Rykunov, S. Dupont, P. Debacker, A. Dewilde, Y. Huang, L. Van der Perre, et al., "An energy efficient 18Gbps LDPC decoding processor for 802.11 ad in 28nm CMOS," in *Solid-State Circuits Conference (A-SSCC), 2015 IEEE Asian*. IEEE, 2015, pp. 1–5.
- [66] Y. Toriyama and D. Markovi, "A 2.267-Gb/s, 93.7-pJ/bit Non-Binary LDPC Decoder With Logarithmic Quantization and Dual-Decoding Algorithm Scheme for Storage Applications," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 8, pp. 2378–2388, Aug 2018.
- [67] Q. Zhang, Y. Chen, D. Wu, X. Zeng, and Y.-l. Ueng, "An area-efficient architecture for stochastic LDPC decoder," in *Digital Signal Processing (DSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 244–247.

- [68] A. Naderi, S. Mannor, M. Sawan, and W. J. Gross, "Delayed stochastic decoding of LDPC codes," *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5617–5626, 2011.
- [69] D. Wu, Y. Chen, Q. Zhang, Y.-l. Ueng, and X. Zeng, "Strategies for reducing decoding cycles in stochastic LDPC decoders," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 9, pp. 873–877, 2016.
- [70] D. Wu, Y. Chen, Q. Zhang, L. Zheng, X. Zeng, and Y.-l. Ueng, "Latency-optimized stochastic LDPC decoder for high-throughput applications," in *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*. IEEE, 2015, pp. 3044–3047.
- [71] Z. Zhang, V. Anantharam, M. J. Wainwright, and B. Nikolic, "An efficient 10GBASE-T ethernet LDPC decoder design with low error floors," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 4, pp. 843–855, 2010.
- [72] A. Cevrero, Y. Leblebici, P. Ienne, and A. Burg, "A 5.35 mm<sup>2</sup> 10GBASE-T Ethernet LDPC decoder chip in 90 nm CMOS," in *Solid State Circuits Conference (A-SSCC), 2010 IEEE Asian*. IEEE, 2010, pp. 1–4.
- [73] M. Li, F. Naessens, M. Li, P. Debacker, C. Desset, P. Raghavan, A. Dejonghe, and L. Van der Perre, "A processor based multi-standard low-power LDPC engine for multi-Gbps wireless communication," in *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*. IEEE, 2013, pp. 1254–1257.
- [74] A. Balatsoukas-Stimming, N. Preys, A. Cevrero, A. Burg, and C. Roth, "A parallelized layered QC-LDPC decoder for IEEE 802.11 ad," in *New Circuits and Systems Conference (NEWCAS), 2013 IEEE 11th International*. IEEE, 2013, pp. 1–4.
- [75] M. Weiner, M. Blagojevic, S. Skotnikov, A. Burg, P. Flatresse, and B. Nikolic, "27.7 A scalable 1.5-to-6Gb/s 6.2-to-38.1 mW LDPC decoder for 60GHz wireless networks in 28nm UTBB FDSOI," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*. IEEE, 2014, pp. 464–465.
- [76] M. Li, Y. Lee, Y. Huang, and L. Van der Perre, "Area and energy efficient 802.11 ad LDPC decoding processor," *Electronics Letters*, vol. 51, no. 4, pp. 339–341, 2015.
- [77] M. Alles, T. Vogt, and N. Wehn, "Flexichap: A reconfigurable ASIC for convolutional, turbo, and LDPC code decoding," in *Turbo Codes and Related Topics, 2008 5th International Symposium on*. IEEE, 2008, pp. 84–89.
- [78] B. Xiang, D. Bao, S. Huang, and X. Zeng, "A fully-overlapped multi-mode QC-LDPC decoder architecture for mobile WiMAX applications," in *Application-specific Systems Architectures and Processors (ASAP), 2010 21st IEEE International Conference on*. IEEE, 2010, pp. 225–232.
- [79] J. Cheng, Q. Yang, and X. Zhou, "Design of a high parallelism high throughput HSPA+ Turbo decoder," in *ASIC (ASICON), 2015 IEEE 11th International Conference on*. IEEE, 2015, pp. 1–4.
- [80] G. Wang, H. Shen, Y. Sun, J. R. Cavallaro, A. Vosoughi, and Y. Guo, "Parallel interleaver design for a high throughput HSPA+/LTE multi-standard turbo decoder," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 5, pp. 1376–1389, 2014.
- [81] X. Chen, Y. Chen, Y. Li, Y. Huang, and X. Zeng, "A 691 Mbps 1.392 mm<sup>2</sup> configurable radix-16 turbo decoder ASIC for 3GPP-LTE and WiMAX systems in 65nm CMOS," in *Solid-State Circuits Conference (A-SSCC), 2013 IEEE Asian*. IEEE, 2013, pp. 157–160.
- [82] S. Belfanti, C. Roth, M. Gautschi, C. Benkeser, and Q. Huang, "A 1Gbps LTE-advanced turbo-decoder ASIC in 65nm CMOS," in *VLSI Circuits (VLSIC), 2013 Symposium on*. IEEE, 2013, pp. C284–C285.
- [83] A. Ahmed, M. Awais, A. ur Rehman, M. Maurizio, and G. Masera, "A high throughput turbo decoder VLSI architecture for 3GPP LTE standard," in *Multitopic Conference (INMIC), 2011 IEEE 14th International*. IEEE, 2011, pp. 340–346.
- [84] C.-H. Lin, C.-Y. Chen, E.-J. Chang, and A.-Y. Wu, "A 0.16 nJ/bit/iteration 3.38 mm<sup>2</sup> turbo decoder chip for WiMAX/LTE standards," in *Integrated Circuits (ISIC), 2011 13th International Symposium on*. IEEE, 2011, pp. 168–171.
- [85] L. Li, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "A low-complexity turbo decoder architecture for energy-efficient wireless sensor networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 1, pp. 14–22, 2013.
- [86] R. Shrestha and R. P. Paily, "High-throughput turbo decoder with parallel architecture for LTE wireless communication standards," *IEEE Trans. Circuits Syst. I*, vol. 61, no. 9, pp. 2699–2710, Sept 2014.
- [87] I. Tal and A. Vardy, "List Decoding of Polar Codes," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.
- [88] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Sig. Process. Mag.*, vol. 21, no. 1, pp. 28–41, jan 2004.
- [89] J. Wang, S. X. Ng, A. Wolfgang, L.-L. Yang, S. Chen, and L. Hanzo, "Near-capacity three-stage MMSE turbo equalization using irregular convolutional codes," in *Proc. Int. Symp. on Turbo Codes*, Munich, Germany, apr 2006.
- [90] H. Lee, Y. Su, and Y. Ueng, "A Rate-Compatible Low-Density Parity-Check Convolutional Coding Scheme Using Informed Dynamic Scheduling," in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, June 2017, pp. 1–5.
- [91] D. D. Le, D. P. Nguyen, T. H. Tran, and Y. Nakashima, "Joint polar and run-length limited decoding scheme for visible light communication systems," *IEICE Communications Express*, vol. 7, no. 1, pp. 19–24, 2018.
- [92] K. Tian, R. Liu, A. Vardy, and R. Wang, "Low-Complexity Hybrid ARQ Scheme for Polar Codes with Higher-Order Modulation," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec 2017, pp. 1–6.
- [93] Y. Sun and J. R. Cavallaro, "Efficient hardware implementation of a highly-parallel 3GPP LTE/LTE-advance turbo decoder," *INTEGRATION, the VLSI journal*, vol. 44, no. 4, pp. 305–315, 2011.
- [94] W. Zhang, S. Chen, X. Bai, and D. Zhou, "A full layer parallel QC-LDPC decoder for WiMAX and Wi-Fi," in *2015 IEEE 11th International Conference on ASIC (ASICON)*, Nov 2015, pp. 1–4.
- [95] D. Bao, B. Xiang, R. Shen, A. Pan, Y. Chen, and X. Y. Zeng, "Programmable architecture for flexi-mode QC-LDPC decoder supporting wireless lan/man applications and beyond," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 1, pp. 125–138, 2010.
- [96] C.-H. Liu, C.-C. Lin, S.-W. Yen, C.-L. Chen, H.-C. Chang, C.-Y. Lee, Y.-S. Hsu, and S.-J. Jou, "Design of a multimode QC-LDPC decoder based on shift-routing network," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 56, no. 9, pp. 734–738, 2009.
- [97] Y. Sun and J. R. Cavallaro, "A low-power 1-gbps reconfigurable LDPC decoder design for multiple 4g wireless standards," in *SOC Conference, 2008 IEEE International*. IEEE, 2008, pp. 367–370.
- [98] C. Howland and A. Blanksby, "Parallel decoding architectures for low density parity check codes," in *Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on*, vol. 4. IEEE, 2001, pp. 742–745.
- [99] T. H. Tran, Y. Nagao, H. Ochi, and M. Kurosaki, "Asic design of 7.7 gbps multi-mode LDPC decoder for IEEE 802.11 ac," in *Communications and Information Technologies (ISCIT)*,

- 2014 14th International Symposium on. IEEE, 2014, pp. 259–263.
- [100] K. Gunnam, G. Choi, W. Wang, and M. Yearly, “Multi-rate layered decoder architecture for block LDPC codes of the IEEE 802.11 n wireless standard,” in *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*. IEEE, 2007, pp. 1645–1648.
- [101] B. Xiang, D. Bao, S. Huang, and X. Zeng, “An 847–955 Mb/s 342–397 mW Dual-Path Fully-Overlapped QC-LDPC Decoder for WiMAX System in 0.13 $\mu$  m CMOS,” *IEEE Journal of Solid-State Circuits*, vol. 46, no. 6, pp. 1416–1432, 2011.
- [102] X.-Y. Shih, C.-Z. Zhan, C.-H. Lin, and A.-Y. Wu, “An 8.29 mm<sup>2</sup> 52 mW Multi-Mode LDPC Decoder Design for Mobile WiMAX System in 0.13 $\mu$  m CMOS Process,” *IEEE Journal of Solid-State Circuits*, vol. 43, no. 3, pp. 672–683, 2008.
- [103] C.-H. Liu, S.-W. Yen, C.-L. Chen, H.-C. Chang, C.-Y. Lee, Y.-S. Hsu, and S.-J. Jou, “An LDPC decoder chip based on self-routing network for IEEE 802.16 e applications,” *IEEE Journal of Solid-State Circuits*, vol. 43, no. 3, pp. 684–694, 2008.
- [104] G. Gentile, M. Rovini, and L. Fanucci, “Low-complexity architectures of a decoder for IEEE 802.16 e LDPC codes,” in *Digital System Design Architectures, Methods and Tools, 2007. DSD 2007. 10th Euromicro Conference on*. IEEE, 2007, pp. 369–375.
- [105] T. Brack, M. Alles, T. Lehnigk-Emden, F. Kienle, N. Wehn, N. E. L’Insalata, F. Rossi, M. Rovini, and L. Fanucci, “Low complexity LDPC code decoders for next generation standards,” in *Proceedings of the conference on Design, automation and test in Europe*. EDA Consortium, 2007, pp. 331–336.
- [106] Y.-L. Wang, Y.-L. Ueng, C.-L. Peng, and C.-J. Yang, “Processing-task arrangement for a low-complexity full-mode WiMAX LDPC codec,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 2, pp. 415–428, 2011.
- [107] X.-Y. Shih, C.-Z. Zhan, C.-H. Lin, and A.-Y. Wu, “A 19-mode 8.29 mm<sup>2</sup> 52-mW LDPC Decoder Chipp for IEEE 802.16 e System,” in *VLSI Circuits, 2007 IEEE Symposium on*. IEEE, 2007, pp. 16–17.
- [108] D. Yang, G. Yu, X. Zou, Y. Deng, and J. Zhong, “The design and verification of a novel LDPC decoder with high-efficiency,” in *Integrated Circuits (ISIC), 2014 14th International Symposium on*. IEEE, 2014, pp. 256–259.
- [109] S. Huang, D. Bao, B. Xiang, Y. Chen, and X. Zeng, “A flexible LDPC decoder architecture supporting two decoding algorithms,” in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*. IEEE, 2010, pp. 3929–3932.
- [110] K. V. Kumar, R. Shrestha, and R. Paily, “Design and implementation of multi-rate LDPC decoder for IEEE 802.16 e wireless standard,” in *Green Computing Communication and Electrical Engineering (ICGCCEE), 2014 International Conference on*. IEEE, 2014, pp. 1–5.
- [111] T.-C. Kuo and A. N. Willson, “A flexible decoder IC for WiMAX QC-LDPC codes,” in *Custom Integrated Circuits Conference, 2008. CICC 2008. IEEE*. IEEE, 2008, pp. 527–530.
- [112] X. Peng, Z. Chen, X. Zhao, D. Zhou, and S. Goto, “A 115mw 1Gbps QC-LDPC decoder ASIC for WiMAX in 65nm CMOS,” in *Solid State Circuits Conference (A-SSCC), 2011 IEEE Asian*. IEEE, 2011, pp. 317–320.
- [113] T. T. Nguyen-Ly, V. Savin, K. Le, D. Declercq, F. Ghaffari, and O. Boncalo, “Analysis and Design of Cost-Effective, High-Throughput LDPC Decoders,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 3, pp. 508–521, March 2018.
- [114] M. Milicevic and P. G. Gulak, “A Multi-Gb/s Frame-Interleaved LDPC Decoder With Path-Unrolled Message Passing in 28-nm CMOS,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 10, pp. 1908–1921, Oct 2018.
- [115] A. Mishra, A. Raymond, L. Amaru, G. Sarkis, C. Leroux, P. Meinerzhagen, A. Burg, and W. Gross, “A successive cancellation decoder ASIC for a 1024-bit polar code in 180nm CMOS,” in *Solid State Circuits Conference (A-SSCC), 2012 IEEE Asian*. IEEE, 2012, pp. 205–208.
- [116] C. Leroux, A. J. Raymond, G. Sarkis, and W. J. Gross, “A semi-parallel successive-cancellation decoder for polar codes,” *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 289–299, 2013.
- [117] B. Yuan and K. K. Parhi, “Low-latency successive-cancellation polar decoder architectures using 2-bit decoding,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 4, pp. 1241–1254, 2014.
- [118] A. Balatsoukas-Stimming, A. J. Raymond, W. J. Gross, and A. Burg, “Hardware architecture for list successive cancellation decoding of polar codes,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 8, pp. 609–613, 2014.
- [119] B. Yuan and K. K. Parhi, “Low-latency successive-cancellation list decoders for polar codes with multibit decision,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 10, pp. 2268–2280, 2015.
- [120] A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg, “LLR-based successive cancellation list decoding of polar codes,” *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5165–5179, 2015.
- [121] C. Xiong, J. Lin, and Z. Yan, “Symbol-decision successive cancellation list decoder for polar codes,” *IEEE Transactions on Signal Processing*, vol. 64, no. 3, pp. 675–687, 2016.
- [122] Y. Fan, J. Chen, C. Xia, C.-y. Tsui, J. Jin, H. Shen, and B. Li, “Low-latency list decoding of polar codes with double thresholding,” in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1042–1046.
- [123] J. Lin, C. Xiong, and Z. Yan, “A reduced latency list decoding algorithm for polar codes,” in *Signal Processing Systems (SiPS), 2014 IEEE Workshop on*. IEEE, 2014, pp. 1–6.
- [124] Y. Fan, C. Xia, J. Chen, C.-Y. Tsui, J. Jin, H. Shen, and B. Li, “A low-latency list successive-cancellation decoding implementation for polar codes,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 2, pp. 303–317, 2016.
- [125] J. Lin and Z. Yan, “An efficient list decoder architecture for polar codes,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 11, pp. 2508–2518, 2015.
- [126] J. Lin, C. Xiong, and Z. Yan, “A high throughput list decoder architecture for polar codes,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 6, pp. 2378–2391, 2016.
- [127] X. Liu, Q. Zhang, P. Qiu, J. Tong, H. Zhang, C. Zhao, and J. Wang, “A 5.16Gbps decoder ASIC for Polar Code in 16nm FinFET,” in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, Aug 2018, pp. 1–5.
- [128] P. Giard, A. Balatsoukas-Stimming, T. C. Müller, A. Bonetti, C. Thibeault, W. J. Gross, P. Flatresse, and A. Burg, “PolarBear: A 28-nm FD-SOI ASIC for Decoding of Polar Codes,” *CoRR*, vol. abs/1708.09603, 2017. [Online]. Available: <http://arxiv.org/abs/1708.09603>
- [129] P. Giard, A. Balatsoukas-Stimming, G. Sarkis, C. Thibeault, and W. J. Gross, “Fast low-complexity decoders for low-rate polar codes,” *Journal of Signal Processing Systems*, vol. 90, no. 5, pp. 675–685, May 2018. [Online]. Available: <https://doi.org/10.1007/s11265-016-1173-y>

- [130] R. Shrestha and A. Sahoo, "High-Speed and Hardware-Efficient Successive Cancellation Polar-Decoder," *IEEE Transactions on Circuits and Systems II: Express Briefs*, pp. 1–1, 2018.
- [131] P. Giard, A. Balatsoukas-Stimming, T. C. Mller, A. Bonetti, C. Thibeault, W. J. Gross, P. Flatresse, and A. Burg, "Polar-Bear: A 28-nm FD-SOI ASIC for Decoding of Polar Codes," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 7, no. 4, pp. 616–629, Dec 2017.
- [132] C. Xia, J. Chen, Y. Fan, C. Tsui, J. Jin, H. Shen, and B. Li, "A High-Throughput Architecture of List Successive Cancellation Polar Codes Decoder With Large List Size," *IEEE Transactions on Signal Processing*, vol. 66, no. 14, pp. 3859–3874, July 2018.



**Shuai Shao** received the BEng degree in Electronic Engineering from the University of Southampton, in 2016. Currently, he is working towards the Ph.D. degree with the University of Southampton. His research is focused on channel coding for next-generation communication. His research interests are turbo coding, polar coding, channel decoder implementations and LDPC coding for the next-generation

wireless communication. He participated in cooperative research with the National Sun Yat-Sen University in Taiwan for two months in 2018.



**Peter Hailes** received the degree in electronic engineering with Mobile and Secure Systems from the Department of Electronics and Computer Science, University of Southampton, and the masters degree (Hons.) in 2013. He then stayed on to undertake research toward the Ph.D. degree in advanced hardware implementations of LDPC decoders. His other research interests include field-programmable gate arrays, error correction coding, embedded hardware/software design, and high-level software development.



**Tsang-Yi Wang (S01-M04)** received the B.S. and M.S. degrees from the National Sun Yat-sen University, Kaohsiung, Taiwan, in 1994 and 1996, respectively, and the Ph. D. degree in electrical engineering from the Syracuse University, NY, in 2003. From 2004 to 2006, he was an Assistant Professor in the Graduate Institute of Communication Engineering at National Chi Nan University, Nantou, Taiwan. In

February 2006, he joined the faculty of the Institute of Communications Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan, as an Assistant Professor, and in August 2008 he became an Associate Professor. He was promoted to Professor in August 2012. He has served as a Reviewer Editor of Journal of Wireless Communications and Mobile Computing. His research mainly focuses on distributed detection and estimation with applications in wireless communications and wireless sensor networks. Dr. Wang received the 2008 Best Paper Award for Young Scholars awarded from IEEE Information Society Taipei Chapter and IEEE Communications Society Taipei/Tainan Chapter.



**Jwo-Yuh Wu (M' 04)** received the B.S., M.S., and Ph.D. degrees from National Chiao Tung University, Taiwan, in 1996, 1998, and 2002, respectively, all in Electrical and Control Engineering. From 2003 to 2007, he was a Post-Doctoral Research Fellow with the Department of Communications Engineering, National Chiao Tung University. Since 2008, he has been with the Department of Electrical and Computer Engineering, and Institute of Communications Engineering, National Chiao Tung University, where he is currently a Full Professor. His general research interests are in signal processing, wireless communications, control systems, linear algebra, and applied functional analysis. He received several research awards, including the 2015 Academia Sinica Research Award for Junior Research Investigators, the Ta-You Wu Memorial Award from the Ministry of Science and Technology of Taiwan in 2014, the 12th Y. Z. Hsu Scientific Paper Award (Communication Technology Category) in 2014, and the 2013 Best Paper Award for Young Scholars by the IEEE Information Theory Society Taipei Chapter and the IEEE Communications Society Taipei/Tainan Chapter. He served as the chair of IEEE Information Theory Society Taipei Chapter during Aug. 2017 July 2019.



**ROBERT G. MAUNDER** has studied with the School of Electronics and Computer Science, University of Southampton, UK, since October 2000. He was awarded a first class honours BEng in Electronic Engineering in July 2003, as well as a PhD in Telecommunications in December 2007. He began a lectureship in November 2007 and was promoted to Associate Professor in March 2013 and to Professor in August 2017. He was awarded Senior Member status of the IEEE in December 2012, Chartered Engineer status of the IET in November 2013 and Fellow status of the IET in January 2017. Robs research interests include joint source/channel coding and the holistic design of algorithms and hardware implementations for wireless communications. He has published a number of IEEE papers in these areas. He is the founder and CTO of AccelerComm Ltd, which is commercialising his research as soft-IP.



**Lajos Hanzo** (<http://www-mobile.ecs.soton.ac.uk>) FEng, F'04, FIET, Fellow of EURASIP, received his 5-year degree in electronics in 1976 and his doctorate in 1983 from the Technical University of Budapest. In 2009 he was awarded an honorary doctorate by the Technical University of Budapest and in 2015 by the University of Edinburgh. In 2016 he was admitted

to the Hungarian Academy of Science. During his 40-year career in telecommunications he has held various research and academic posts in Hungary, Germany and the UK. Since 1986 he has been with the School of Electronics and Computer Science, University of Southampton, UK, where he holds the chair in telecommunications. He has successfully supervised 119 PhD students, co-authored 18 John Wiley/IEEE Press books on mobile radio communications totalling in excess of 10 000 pages, published 1800+ research contributions at IEEE Xplore, acted both as TPC and General Chair of IEEE conferences, presented keynote lectures and has been awarded a number of distinctions. Currently he is directing a 60-strong academic research team, working on a range of research projects in the field of wireless multimedia communications sponsored by industry, the Engineering and Physical Sciences Research Council (EPSRC) UK, the European Research Council's Advanced Fellow Grant and the Royal Society's Wolfson Research Merit Award. He is an enthusiastic supporter of industrial and academic liaison and he offers a range of industrial courses. He is also a Governor of the IEEE ComSoc and VTS. He is a former Editor-in-Chief of the IEEE Press and a former Chaired Professor also at Tsinghua University, Beijing. For further information on research in progress and associated publications please refer to <http://www-mobile.ecs.soton.ac.uk>



**Bashir M. Al-Hashimi (M99-SM01-F09)**

is an ARM professor of Computer Engineering, dean of the Faculty of Physical Sciences and Engineering, and the co-director of the ARM-ECS Research Centre, University of Southampton, Southampton, United Kingdom. He has published more than 380 technical papers. His current research interests include methods, algorithms, and design automation tools

for low-power design and test of embedded computing systems. He has authored or co-authored five books and has graduated 35 PhD students. He is fellow of the IEEE.

APPENDIX

TABLE VI: Comparison of ASIC implementation of channel decoders.

Ref.	Code and Standard	Coding Rate	Information Block Length	Average Scaled Throughput (decoded Mbps)	Scaled Average Area Efficiency (Mbps/mm <sup>2</sup> )	Scaled Average Power Efficiency (bit/nJ)	Energy Required ( $E_b/N_0$ in dB)
[64]	Turbo LTE	-	6144	1388	1369.697	3.225806	0.49
[64]	Turbo HSDPA	-	5144	1130	1682.424	2.626197	0.49
[79]	Turbo LTE	-	6144	792	397.9899497	0	-
[80]	Turbo LTE	1/3	6144	1156.154	112.7902	0.65729443	0.5588
[80]	Turbo HSDPA+	1/3	5144	571.8462	228.0382	1.328912467	0.5804
[81]	Turbo LTE	1/3	6144	691.7	496.9109195	1.089291339	0.75
[82]	Turbo LTE	1/3	6144	1013	406.8273092	1.048654244	0.803
[51]	Turbo LTE	1/3	3200	781.2	875.2941176	0.990239574	0.81
[83]	Turbo LTE	1/3	6016	393.7846154	359.4990572	0	0.6
[84]	Turbo LTE	1/3	6144	257.6769231	146.1561624	1.739884693	0.633
[85]	Turbo LTE	1/3	6144	1.426153846	7.811899343	0.34200332	-
[61]	Turbo LTE	1/3	5144	47.8	149.375	0.637758506	0.794
[61]	Turbo LTE	1/3	6144	211.2	132	0.563575717	0.738
[51]	Turbo LTE	1/3	6144	607.6107692	190.9650686	2.233534661	0.503
[51]	Turbo LTE	1/3	6144	4578.923077	444.4823839	3.156789436	1.159
[37]	Turbo LTE	1/3	6144	21900	200.9174312	2.276980661	0.79
[36]	Turbo LTE	0.94	3016	2150	279.2207792	0	-
[93]	Turbo LTE	1/3	6144	1280	154.2168675	1.514792899	-
[94]	LDPC WiFi	2/3	576-2304	586.2307692	98.23210251	-	-
[52]	LDPC WiFi	1/2	336-8640	474.9231	164.6480033	2.884148375	1.573
[52]	LDPC WiFi	5/6	336-8640	2709.692	939.4055106	7.24301611	3.568
[95]	LDPC WiFi	1/2 to 23/24	576-2304	372.2404654	282.0729022	0.434860357	-
[96]	LDPC WiFi	1/2 to 5/6	576-2304	195.6433846	60.30219051	0.370536713	-
[97]	LDPC WiFi	-	576-2304	1107.858528	606.8405124	2.702093971	-
[98]	LDPC WiFi	1/2	1024	142.35	1259.734513	-	2.141
[73]	LDPC WiFi	1/2 to 13/16	1944	3145.88	1677.942495	8.851660101	-
[99]	LDPC WiFi	-	648-1944	8530.510666	1994.433147	-	-
[100]	LDPC WiFi	1/2,2/3,3/4,5/6	648-1944	1618.294	3499.014054	-	-
[94]	LDPC WiMAX	2/3	576-2304	586.2307692	98.23210251	-	-
[101]	LDPC WiMAX	1/2	576-2304	1205	1590.759076	3.035264484	2.448
[102]	LDPC WiMAX	1/2	576-2304	141	68.03377563	2.711538462	2.81
[52]	LDPC WiMAX	1/2	336-8640	474.9231	164.6480033	2.884148375	1.573
[52]	LDPC WiMAX	5/6	336-8640	2709.692	939.4055106	7.24301611	3.568
[103]	LDPC WiMAX	1/2	576-2304	116.3251454	35.68222213	0.440625551	1.752
[104]	LDPC WiMAX	1/2,2/3,3/4,5/6	576-2304	293.5	533.6363636	-	-
[105]	LDPC WiMAX	1/2,2/3,3/4,5/6	576-2304	190.5	142.4831713	-	-
[106]	LDPC WiMAX	2/3	576-2304	184.6176923	521.2689415	2.921166018	2.527
[95]	LDPC WiMAX	1/2 to 23/24	576-2304	372.2404654	282.0729022	0.434860357	-
[96]	LDPC WiMAX	1/2 to 5/6	576-2304	195.6433846	60.30219051	0.370536713	-
[97]	LDPC WiMAX	1/2,2/3,3/4,5/6	576-2304	1107.858528	606.8405124	2.702093971	-
[77]	LDPC WiMAX	1/2	576-2304	142.35	1259.734513	-	-
[107]	LDPC WiMAX	802.16e	576-2304	60.6	54.47191011	1.158699809	2.809
[108]	LDPC WiMAX	5/6	576-2304	248.7993231	177.9806162	-	-
[109]	LDPC WiMAX	1/2	576-2304	328.0492197	208.2852188	1.21499711	1.77
[78]	LDPC WiMAX	1/2	576-2304	535	869.9186992	1.897163121	2.156
[110]	LDPC WiMAX	5/6	576	255.36	1064	-	2.598
[111]	LDPC WiMAX	1/2	576-2304	150.6687598	340.8331373	0.913143999	1.708
[112]	LDPC WiMAX	1/2,2/3,3/4,5/6	576-2304	703.824	209.4714286	6.120208696	2.326
[113]	LDPC WiMAX	1/2,2/3,3/4,5/6	2304	1835	2293	-	2.25

Ref.	Code and Standard	Coding Rate	Information Block Length	Average Scaled Throughput (decoded Mbps)	Scaled Average Area Efficiency (Mbps/mm <sup>2</sup> )	Scaled Average Power Efficiency (bit/nJ)	Energy Required ( $E_b/N_0$ in dB)
[114]	LDPC	1/2,5/8,3/4,13/16	672	2920.6	272.3395	12.0972	-
[105]	LDPC DVB-S2	1/4-10	64800	384	99.45609946	-	-
[65]	LDPC WiGig	1/2	672	6401.893491	1523.01153	38.56562344	3.042
[74]	LDPC WiGig	802.11ad	672	1260	2650.887574	-	-
[73]	LDPC WiGig	802.11ad	672	2140.384615	3752.591918	15.79619642	-
[73]	LDPC WiGig	802.11ad	672	2140.384615	4135.509461	18.66072027	-
[75]	LDPC WiGig	1/2	672	646.1538462	190.3201335	104.2183623	2.833
[75]	LDPC WiGig	5/8	672	1292.307692	380.640267	92.30769231	3.12
[75]	LDPC WiGig	3/4	672	2584.615385	761.2805341	68.01619433	3.439
[75]	LDPC WiGig	4/5	672	5169.230769	1522.561068	28.71794872	3.894
[76]	LDPC WiGig	802.11ad	672	3221.822485	4744.827329	60.10862846	-
[63]	LDPC 10GBASE-T	0.841	2048	17246.82617	6387.713397	-	3.913
[58]	LDPC 10GBASE-T	0.833	2048	71407.6848	21457.67086	-	-
[67]	LDPC 10GBASE-T	0.833	2048	8581.347656	3900.612571	-	3.864
[68]	LDPC 10GBASE-T	0.833	2048	200826.8329	97968.73371	-	-
[69]	LDPC 10GBASE-T	0.833	2048	20124.10156	7317.855114	-	4.018
[45]	LDPC 10GBASE-T	0.841	2048	52909.2503	10566.19644	47.66599126	4.011
[70]	LDPC 10GBASE-T	0.833	2048	69475.26367	25263.73224	-	4.007
[71]	LDPC 10GBASE-T	0.841	2048	47700	8912.096461	17.03571429	3.738
[72]	LDPC 10GBASE-T	0.841	2048	16186.15385	5799.402496	10.38239503	3.86
[55]	LDPC 10GBASE-T	0.83	3600	15680	1572.764	987.4994	-
[66]	LDPC 10GBASE-T	0.89	8352	334.769	26.80265	16.68112	4.2
[115]	Polar Inflexible	0.5	512	271.3846154	608.5235848	5.608407666	-
[116]	Polar Inflexible	0.5	-	246.1	396.9354839	-	2.83
[116]	Polar Inflexible	0.5	-	244.2	230.3773585	-	-
[116]	Polar Inflexible	0.5	-	242.4	128.9361702	-	-
[116]	Polar Inflexible	0.5	-	240.6	63.65079365	-	-
[117]	Polar Inflexible	0.5	512	692.3076923	-	-	-
[60]	Polar Inflexible	0.5	-	497.28	3656.470588	-	-
[60]	Polar Inflexible	0.5	-	493.48	2902.823529	-	-
[60]	Polar Inflexible	0.5	-	494.74	2078.739496	-	-
[60]	Polar Inflexible	0.5	-	486.06	1299.625668	-	-
[60]	Polar Inflexible	0.5	-	487.36	752.0987654	-	-
[60]	Polar Inflexible	0.5	-	474.07	398.3781513	-	-
[60]	Polar Inflexible	0.5	-	470.59	206.7618629	-	-
[60]	Polar Inflexible	0.5	-	476.69	107.1213483	-	-
[118]	Polar Inflexible	0.5	-	250.6153846	150.1467911	-	2.179
[118]	Polar Inflexible	0.5	-	171.6923077	46.62345382	-	2
[56]	Polar Inflexible	0.5	512	25600	7619.047619	33.16062176	2.516
[62]	Polar Inflexible	0.5	-	1165.153846	373.7928501	-	3.066
[62]	Polar Inflexible	0.5	-	931.8461538	181.4071844	-	-
[62]	Polar Inflexible	0.5	-	776.7692308	52.73338225	-	-
[62]	Polar Inflexible	0.5	-	666	11.24443781	-	-
[62]	Polar Inflexible	0.5	-	582.9230769	2.232327392	-	-
[62]	Polar Inflexible	0.5	-	1280.769231	688.1836422	-	3.322
[62]	Polar Inflexible	0.5	-	1048.153846	166.4577849	-	-
[62]	Polar Inflexible	0.5	-	886.1538462	32.787141	-	-
[62]	Polar Inflexible	0.5	-	768.4615385	4.686787536	-	-
[62]	Polar Inflexible	0.5	-	677.7692308	1.227982273	-	-
[119]	Polar Inflexible	0.5	512	300	154.6391753	0.46728972	2.172
[119]	Polar Inflexible	0.5	512	501	236.3207547	0.634177215	2.172
[119]	Polar Inflexible	0.5	512	262	66.16161616	0.178474114	1.9
[119]	Polar Inflexible	0.5	512	401	93.69158879	0.279247911	1.9
[120]	Polar Inflexible	0.5	512	463.8461538	505.2654446	-	2.256

Ref.	Code and Standard	Coding Rate	Information Block Length	Average Scaled Throughput (decoded Mbps)	Scaled Average Area Efficiency (Mbps/mm <sup>2</sup> )	Scaled Average Power Efficiency (bit/nJ)	Energy Required ( $E_b/N_0$ in dB)
[120]	Polar Inflexible	0.5	512	425.0769231	228.9158352	-	2.098
[120]	Polar Inflexible	0.5	512	340.6153846	91.20308801	-	2.061
[121]	Polar Inflexible	0.5	480	342	291.1494845	-	1.887
[121]	Polar Inflexible	0.5	480	433.3846154	343.6176785	-	1.887
[121]	Polar Inflexible	0.5	480	459.6923077	264.0214487	-	1.896
[121]	Polar Inflexible	0.5	480	441.6923077	237.5967228	-	-
[121]	Polar Inflexible	0.5	480	551.0769231	376.5155037	-	2.111
[121]	Polar Inflexible	0.5	480	524.7692308	331.1607815	-	-
[122]	Polar Inflexible	0.5	1024	609.2307692	78.28369419	-	1.372
[123]	Polar Inflexible	0.5	4096	218.8034188	-	-	-
[124]	Polar Inflexible	0.5	512	1273.846154	163.4649659	-	1.729
[125]	Polar Inflexible	0.5	512	221.5384615	523.0598824	-	2.142
[125]	Polar Inflexible	0.5	512	250.6153846	434.4210953	-	-
[125]	Polar Inflexible	0.5	512	200.7692308	237.5967228	-	-
[125]	Polar Inflexible	0.5	512	239.5384615	202.841652	-	-
[126]	Polar Inflexible	0.5	512	1844.307692	892.8869947	-	-
[126]	Polar Inflexible	0.5	512	1578.461538	395.0604373	-	-
[126]	Polar Inflexible	0.5	512	1035.692308	137.5060678	-	-
[126]	Polar Inflexible	0.5	4096	2323.384615	651.2133771	-	1.86
[126]	Polar Inflexible	0.5	4096	2002.153846	297.0935599	-	1.697
[126]	Polar Inflexible	0.5	4096	1326.461538	103.712834	-	1.584
[126]	Polar Inflexible	0.5	16384	2997.129481	461.8951806	-	-
[126]	Polar Inflexible	0.5	16384	2724.94276	219.6867375	-	-
[126]	Polar Inflexible	0.5	16384	2034.480402	84.42476116	-	-
[127]	Polar Inflexible	0.85	32768	1093.4	189.29	-	-
[127]	Polar Inflexible	0.5	32768	639.8	110.75	-	-
[128]	Polar Inflexible	0.5	1024	80.81	34.081	1.957	2.06
[129]	Polar Inflexible	0.5	1024	1240	1797.1	12.9	-
[130]	Polar Inflexible	0.5	1024	825.23	1182.8	2.13	-
[59]	Polar Flexible	0.5	-	4043.076923	25330.79868	28.0466258	-
[59]	Polar Flexible	0.5	-	3918.461538	11112.89467	19.54453217	-
[59]	Polar Flexible	0.5	-	3890.769231	4913.851263	16.97296451	-
[59]	Polar Flexible	0.5	-	3724.615385	2358.217553	14.21488358	-
[59]	Polar Flexible	0.5	-	3544.615385	1057.515405	12.86819348	2.533
[56]	Polar Flexible	0.5	-	25600	7485.380117	56.63716814	-
[56]	Polar Flexible	0.5	-	25600	8888.888889	23.44322344	-
[56]	Polar Flexible	0.5	-	25600	2983.682984	33.77308707	-
[56]	Polar Flexible	0.5	-	25600	3575.418994	17.2972973	-
[129]	Polar Flexible	0.5	-	3720	2695.652174	8.651162791	2.648
[129]	Polar Flexible	0.5	-	2480	1797.101449	12.91666667	-
[48]	Polar Flexible	0.5	-	1514.769231	768.268418	-	-
[48]	Polar Flexible	0.5	-	1514.769231	625.8738405	-	1.857
[48]	Polar Flexible	0.5	-	3345.230769	1382.185739	-	2.139
[48]	Polar Flexible	0.5	-	7994.769231	3303.286613	-	2.591
[48]	Polar Flexible	0.5	-	1855.384615	396.1101036	-	-
[48]	Polar Flexible	0.5	-	1855.384615	322.1982545	-	1.639
[48]	Polar Flexible	0.5	-	4079.076923	708.3552671	-	1.77
[48]	Polar Flexible	0.5	-	9700.615385	1684.567889	-	2.078
[48]	Polar Flexible	0.5	-	2556.923077	245.8390191	-	-
[48]	Polar Flexible	0.5	-	2556.923077	205.2776399	-	4.221
[48]	Polar Flexible	0.5	-	2636.923077	211.7002857	-	4.221
[48]	Polar Flexible	0.5	-	2786.153846	223.6809903	-	4.242
[48]	Polar Flexible	0.5	-	5483.076923	440.1982603	-	4.311
[48]	Polar Flexible	0.5	-	13075.38462	1049.732047	-	4.455



<b>Ref.</b>	<b>Code and Standard</b>	<b>Coding Rate</b>	<b>Information Block Length</b>	<b>Average Scaled Throughput (decoded Mbps)</b>	<b>Scaled Average Area Efficiency (Mbps/mm<sup>2</sup>)</b>	<b>Scaled Average Power Efficiency (bit/nJ)</b>	<b>Energy Required (<math>E_b/N_0</math> in dB)</b>
[131]	Polar Flexible	0.5	1024	80.554	2279.5	220.5	1.8
[127]	Polar Flexible	0.5	1024	797.8	21.295	-	-
[127]	Polar Flexible	0.5	2048	13.29	1.873	-	1.3
[128]	Polar Flexible	0.5	1024	80.812	34.08	1.94	-
[128]	Polar Flexible	0.5	1024	56.38	24.78	1.04	-
[132]	Polar Flexible	0.5	1024	1145	112.12	-	-
[132]	Polar Flexible	0.5	1024	1340	318.4	-	-
[132]	Polar Flexible	0.5	1024	1527	644.9	-	-