D1.2 V2

B5G Wireless Tb/s FEC KPI Requirement and Technology Gap Analysis

Project number:	760150
Project acronym:	EPIC
Project title:	EPIC: Enabling Practical Wireless Tb/s Communications with Next Generation Channel Coding
Start date of the project:	1st September, 2017
Duration:	36 months
Programme:	H2020-ICT-2016-2

Deliverable type:	Report
Deliverable reference number:	ICT-760150 / D1.2 / V2.0
Work package contributing to the deliverable:	WP1
Due date:	June 2019 – M22
Actual submission date:	3rd July, 2019

Responsible organisation:	IMEC
Editor:	Claude Desset
Dissemination level:	PU
Revision:	2.0

Abstract:	To quantify the FEC improvements needed for the seven Beyond-5G Tb/s use-cases considered in EPIC, a three-step approach has been followed. First, a detailed state-of-the-art analysis of Turbo codes, LDPC codes and Polar codes has been conducted and the best implementations have been scaled to 7 nm bulk CMOS to take scaling improvements into account. Then the gap between the EPIC FEC requirements and the scaled reference designs has been assessed. Power density and energy efficiency emerge as the biggest challenges.
Keywords:	Beyond 5G requirements, Tb/s use cases, Forward error correction, Turbo codes, LDPC codes, Polar codes, CMOS, CMOS scaling, 7 nm technology node.

The EPIC project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 760150.

Editor

Meng Li, André Bourdoux, Claude Desset (IMEC)

Contributors (ordered according to beneficiary numbers)

Bianca Michal (TEC)

Onur Sahin (IDCC)

Altuğ Süral, Orhun Arpaci, Göksu Sezer, Yiğit Ertuğrul, Ertuğrul Kolağasıoğlu, Erdal Arıkan (POL)

Norbert Wehn, Claus Kestel, Matthias Herrmann, Stefan Weithoffer (TUKL)

Leefke Grosjean, Hugo Tullberg, Guido Carlo Ferrante (EAB)

Catherine Douillard, Amer Baghdadi, Charbel Abdel Nour (TB)

Jhon Jimenez (CRE)

Disclaimer

The information in this document is provided “as is”, and no guarantee or warranty is given that the information is fit for any particular purpose. The content of this document reflects only the author`s view – the European Commission is not responsible for any use that may be made of the information it contains. The users use the information at their sole risk and liability.

Executive Summary

Future Beyond-5G use cases are expected to require wireless speeds in the Terabit/s range. This sets a number of tough challenges on the physical layer and especially on the Forward Error Correction (FEC), which is the core technology addressed by the EPIC project.

This deliverable quantifies the FEC implementation related performance improvements that the EPIC project must achieve. The global methodology in this deliverable follows three steps: defining the use cases and their requirements, describing the state-of-the-art and quantifying the gap to be bridged between requirements and current state-of-the-art.

Seven challenging Beyond-5G Terabit/s use cases are first described in detail, namely: data kiosk, virtual reality, intra-device communication, wireless fronthaul/backhaul, data centers, hybrid fiber-wireless networks, and high-throughput satellites. The main characteristics covered in this description are the system setup, the system related requirements (bit and frame error rate, throughput, latency, power, cost, flexibility) and FEC related requirements (bit and frame error rate, throughput, latency, energy efficiency, area efficiency, and power density).

Then, a comprehensive state-of-the-art search is conducted to assess the performance of leading FEC technologies with strong potential to achieve the desired requirements. The EPIC project focuses on three code classes, namely Polar Codes, Turbo Codes and Low Density Parity Check Codes. The best designs are selected as reference and their FEC performance data are scaled to the 7 nm digital bulk CMOS technology node to incorporate the improvements expected by CMOS scaling by 2020 and beyond. This allows to make a fair assessment of the different designs possible and to compare between different code classes.

Finally, the scaled state-of-the-art performance data is compared against the FEC requirements of the seven use cases to obtain the FEC performance gaps that need to be closed for the realization of the anticipated wireless Terabit/s use cases. The gap analysis has not considered the communications performance since this is not always given in implementation-oriented state-of-the-art papers.

The gap analysis shows that improvement is needed in many areas but from an implementation point of view, the biggest challenge is power consumption, hence the energy efficiency and power density. The 1 Tb/s net data throughput under a 1 GHz clock constraint is only feasible by unrolling/functional parallelism. However, unrolling under the area constraint of 10 mm^2 becomes only feasible for smaller block lengths and limits the number of decoding iterations. In some use cases, where the transmit power can be increased to compensate for the weakness of the FEC or the suboptimality of the decoding algorithm, this is not a challenge. Sub-microsecond latencies are also challenging. The unrolled architecture seems to be a key architecture and will need to be revisited to improve its flexibility in code rate and block length.

These performance gaps define the FEC targets for the work conducted within the EPIC project.

This document was revised after the first project review based on reviewer requirements. Modifications include an argumentation on code block length for each code family (Section 2.1), specification of code rates for each use case (Section 2.2), revision of some latency specs (Section 2.2) and prioritization of use cases for our future research (Chapter 4).

Content

List of Figures

List of Tables

Chapter 1 Introduction

Societal development changes how we live, work, and interact. The technical development of wireless communication systems allows us to do previously unimaginable things. For example, watching movies was previously confined in space and time, to movie theatres or living rooms and at determined times. Today we can enjoy high-quality media while on the move, anywhere and anytime, thanks to wireless mobile broadband. However, as technical possibilities develop, so do the applications. For the future, we foresee a continued increasing demand on data rates as we continue to move from standard TV to ultra-high-definition TV and virtual reality.

Forward Error Correction (FEC) is a core technology component of any digital communication system and is the enabler of practicable Beyond-5G (B5G) wireless Terabit/s (Tb/s) solutions. The main goal of EPIC is to develop a set of new implementation-ready FEC technologies that meet the cost and performance requirements of a variety of future wireless Tb/s use cases. The EPIC project puts great emphasis on evaluating the commercial viability of the project outcomes targeting B5G systems, based on a systematic idea-to-market approach as an integral part of the project execution.

In the past, steady progress in silicon technology - as predicted by Moore’s law - could be counted on as the enabler of such large leaps in data rates in a few technology nodes (generations), without the need for major algorithmic innovations on the FEC design part. Indeed, already for some decades, simple technology scaling has been in effect and served as a major enabler of wireless communications at ever increasing data rates, with improved energy and area efficiency and at reduced cost. A key finding is the prediction that the upgrade to Tb/s wireless data rates will not be as smooth: the improvements carried by silicon technology progress in the next decade will significantly fall short of meeting the Tb/s FEC challenge. Moreover, FEC implementation at Tb/s will require several Tera-FLOPS of computational power, bringing forth the emergence of power density on the silicon chip as a key factor thereby adding new dimensions into the already challenging FEC design space. The EPIC project is based on the thesis that major algorithmic and architectural innovations, in an EPIC holistic design framework, will be required in the design and implementation of FEC algorithms to make wireless communications at Tb/s rates feasible. This will be applied to the three code classes considered in the EPIC project: Turbo codes, Low Density Parity Check (LDPC) codes and Polar codes, whose bit error rate (BER) performance come close to the Shannon limit.

The global methodology in this deliverable follows three steps: defining the use cases and their requirements, describing the state-of-the-art (SoA) and quantifying the gap to be bridged between requirements and current SoA.

For the first step (definition of the use cases), a review of the B5G use cases studied in current technical literature and standardization work has served as a starting point. The final choice of the use cases described in this deliverable was made based on several criteria: for the EPIC project, only use cases that directly require advances in the FEC technology are relevant. The use cases were chosen sufficiently different from each other to achieve a large diversity. The identified use cases are: data kiosk, virtual reality, intra-device communication, data centre, hybrid fiber-wireless networks, wireless fronthaul/backhaul and high-throughput satellites. Each use case offers a specific set of challenges; collectively, these use cases therefore present a diverse set of FEC design challenges. For each of the use cases, we first detail the following system level Key Performance Indicators (KPI): BER performance, throughput, latency, power consumption, cost and volume. After that, the KPI related to FEC implementation (area, area efficiency, energy efficiency and power density) are derived by scaling to different silicon technology nodes based on the power budget, cost budget and chip volume from market driven perspective for the different use cases. For silicon implementation, a realistic KPI estimation further needs to consider the practical silicon fabrication size, yield, packaging method and thermal dissipation. Hence, since certain scaling may result in unrealistic values, we will not allow the area and power density to grow too large. We will consider the following values “within reach”, although very challenging and thus deserving extensive research: target throughput from 500 Gb/s to 1000 Gb/s, latency from 0.5 ms down to 100 ns, BER from 10^{-6} down to 10^{-12}, area efficiency around 100 Gb/s/ mm², energy efficiency around 1 pJ/bit and power density around 0.1 W/mm². These EPIC targets represent an improvement of approximately 10x–100x in throughput, 10x–50x in energy efficiency and 3x–30x in area efficiency over the SoA. All throughput figures refer to the net data throughput (information bits) and not to the coded throughput.

For the second step (SoA), we conduct a detailed SoA search for the three code families. We review papers and references achieving the highest possible performance in terms of the KPI targeted by the EPIC project. Furthermore, we describe the key techniques and architectures used in these papers. This gives an overview of the different techniques used in the SoA and gives an indication of their suitability to reach the EPIC goals. Both generic techniques (such as parallelization, pipelining) and code specific techniques are covered.

For the third step (gap analysis), we scale the most interesting references from the SoA to the 7 nm CMOS technology node and then assess the gap between the scaled SoA and the EPIC targets. The rationale is as follows. The EPIC project objectives span the timeline of 2020 and beyond. During this timeline, the silicon technology is expected to progress to 7 nm - and possibly further - from the presently available larger technology nodes (such as 65, 40, 28 and 16 nm), hence resulting in faster digital systems and higher densities [1]. Part of the gaps to the goal of Tb/s FEC targets will be closed thanks to advances in silicon technology alone. The gaps left over after considering the advances in silicon technology are the real challenges for the EPIC project. Therefore, when analyzing the gap between the goal and the SoA, all the implementation metrics (throughput, latency, energy efficiency, area efficiency and power density) are scaled to the 7 nm technology node. Based on the International Technology Roadmap for Semiconductors (ITRS) roadmap [2] and NVIDIA study on future exascale computing [3], we estimated that moving one architecture from 28 nm to 7 nm technology will bring, approximately, a factor x12 reduction in area, a factor x4 improvement in energy efficiency and a factor x3 increase in clock speed (maximum operating frequency) but with a limitation of 1 GHz clock frequency. The scaling factors derived from these values will be used to compute the performance improvements when scaling from any technology node to 7 nm.

This deliverable is organized as follows. In Chapter 2, we give an in-depth description of each use case including all system level KPI and FEC KPI. In Chapter 3, the SoA of high speed decoders and the gap analysis between SoA and target KPI of the seven use cases for the three different code families are detailed. Chapter 4 provides a summary as well as concluding remarks.

Chapter 2 System and FEC KPI for Use Cases

In this chapter, we list seven B5G use cases that we have identified as being particularly interesting in the context of new FEC technologies, together with important performance targets. They are: data kiosk, virtual reality, intra-device communication, wireless fronthaul/backhaul, data centers, hybrid fiber-wireless networks, and high-throughput satellites. Each of the use cases is described with respect to the system setup, the system related requirements (BER/FER, throughput, latency, power, cost, flexibility), which are related to FEC design and implementation specification and FEC related KPI (BER/FER, throughput, latency, energy efficiency, area efficiency and power density). The deep scaled CMOS technology node brings significant implementation performance gain, e.g. the clock rate is increasing approximately 1.3 times for every technology node. Taking this important aspect into account, the FEC level KPI are listed under 3 different technology nodes (28 nm, 16 nm and 7 nm). The 10 nm is the intermediate technology node hence is not included in the tables of this chapter. After the analysis of the use cases with an emphasis on the FEC related requirements, it is possible to distinguish different types of use cases, thus, allowing us to categorize and then steer our efforts in the following work.

2.1 Methodology

As of today, there are no wireless systems targeting a throughput of 1 Tb/s. Deriving the FEC level KPI for the seven use cases listed here is therefore a challenging task. Here, we take a top down method: The FEC is a key component of the physical layer (PHY) chip of a communication device. Parameters such as size, energy consumption, etc. of the FEC are therefore tightly connected to the parameters of the PHY chip, including cost. Using this connection, the FEC parameters such as area efficiency, energy efficiency and power density can be derived. The method is described as follows:

a) System level KPI determination

The system level KPI: BER/FER, latency, throughput, power consumption, cost and flexibility in code length and code rate are either obtained based on standards [4] [5] or estimated based on results in the literature.

b) FEC level KPI extrapolation

The FEC level KPI are: BER, latency, throughput, area efficiency, energy efficiency, power density, and flexibility in code length and code rate.

Name of KPI	Unit	Explanation
Throughput	Gb/s	The net information throughput.
Area	mm^2	Area of the decoder circuit.
Power	Watt	Total power dissipation by the decoder circuit.
Area Efficiency	Gb/s/mm^2	Throughput per unit area.
Power Density	W/mm^2	Power dissipation per unit area.
Energy Efficiency	pJ/bit	Energy required for decoding one information bit.
Latency	µs/ms/ns	Duration of decoding one codeword.
Frequency	MHz	Achievable clock frequency of the decoder.

Table 1: KPI explanation

Let us first introduce the following notation:

- The cost of the whole device C_{unit}.
- The percentage of the device cost assigned to the PHY X_{PHY}.
- The percentage of the PHY cost assigned to the FEC X_{FEC}.
- The power consumption of the whole device P_{unit}.
- The percentage of the power consumption of the device given to the PHY Y_{PHY}.
- The percentage of the power consumption of the PHY given to the FEC Y_{FEC}.
- The expected number of devices initially sold (volume).
- The throughput on FEC level (same as on system level).

The FEC level cost and power budget can be obtained from equations (2.1) and (2.2).

$$FEC_{\text{cost}} = C_{\text{unit}}\, X_{\text{PHY}} X_{\text{FEC}} \qquad (2.1)$$

$$FEC_{\text{power}} = P_{unit}\, Y_{\text{PHY}} Y_{\text{FEC}} \qquad (2.2)$$

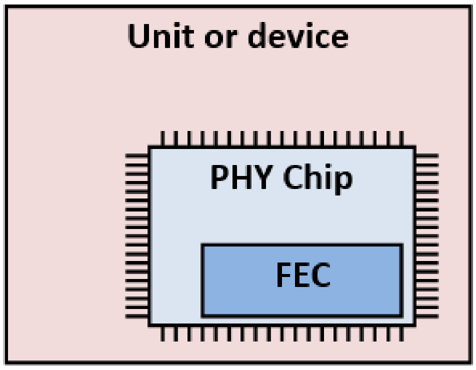


Figure 1: Physical device illustration

$$\text{FEC\_chip\_area} = \left(FEC_{\text{cost}} - \frac{\text{mask\_price}}{\text{volume}}\right) * \text{wafer\_area/wafer\_price} \qquad (2.3)$$

The total chip cost consists of the die cost, the packaging and testing cost. Furthermore, it largely depends on the yield. In the following we assume that the die cost is the dominant cost part. The die cost is composed of wafer and mask cost. The wafer price is more foundry independent while the mask set price differs a lot from different foundries. The wafer and mask cost we used in EPIC is shown in Table 2. Based on the above assumption, the FEC area under different technology node with the cost limitation can be derived from equation (2.3).

Node	28 nm	16 nm	7 nm
Wafer size (diam)	12”	12”	12”
Full mask set price*	€1.5M	€3.2M	€6.5M
Wafer price*	€5k	€11k	€22k

Table 2: Wafer and mask costs for different technologies

The other FEC level related KPI can be derived from system level KPI as follows:

$$\text{Area efficiency} = \frac{\text{Throughpout}}{\text{FEC\_chip\_area}} \ (\text{bit/s/mm}^2) \qquad (2.4)$$

$$\text{Energy efficiency} = \frac{FEC_{\text{power}}}{\text{Throughput}} \ (\text{pJ/bit}) \qquad (2.5)$$

$$\text{Power density} = \frac{FEC_{\text{power}}}{\text{FEC\_chip\_area}} \ (\text{W/mm}^2) \qquad (2.6)$$

c) Code length selection

For standardized applications there are generally specific requirements on the code lengths to support. For instance, with OFDM systems it is generally more efficient to have an integer relationship between the number of sub-carriers and the code length such that code blocks are aligned to the modulation symbols. However, for the 7 use cases considered in this deliverable, the selected applications are still generic in order to derive general coding requirements and not directly related to frozen standards. Hence there are no strict requirements from that side and code length recommendations are possible within EPIC in view of other arguments. In particular, the trade-offs between coding performance (coding gain) and implementation aspects (throughput, complexity) should be considered.

Concerning this trade-off between coding performance and implementation, there are a few general trends. For instance, very short codes will be worse in coding performance while very long codes will generally suffer from a too high complexity. However, these trends are only generic and in order to propose recommended code length values, they need to be investigated within the 3 code families considered in EPIC, as each family can have a very specific behaviour as function of the code length.

For each code family, first insights on implementation aspects will guide the selection of typical lengths for which the EPIC targets in throughput and complexity are expected to be in reach. Additionally, error rate simulations and existing literature can be used in order to check whether a sufficient coding performance is obtained for selected code lengths.

Recommendations for Turbo Codes

According to a recent survey of efficient error-correcting codes in the short frame size regime, Turbo Codes (TC) are known to provide excellent coding gains in the moderate frame size regime (typically 150-2000 bits) and, if carefully designed, for short frame sizes as well (typically 50-150 bits) [6]. For instance, a 16-state tail-biting TC is shown to perform at 0.75 dB from Gallager's random coding bound for a frame error rate equal to 10^{-6}.

In order to achieve the EPIC goals, highly parallel architecture templates have to be investigated. Among the considered candidates, the fully pipelined iteration unrolled XMAP decoder architecture is the most promising one. With this architecture, the serial MAP decoding of complete frames is functionally parallelized. For such a decoder, the size of the pipelines grows quadratically with the size of the input frame, which sets a complexity limit on the manageable frame sizes in the order of a few hundreds of bits.

The first investigated architectures will target fixed frame sizes. However, some flexibility in this regard can be introduced. To allow efficient flexible implementations, an overlap between the different interleavers corresponding to the different sizes is desired, resulting in more regular connections and less multiplexing between different frame size configurations. The Almost Regular Permutation (ARP) interleaver template [7, 8] is able to provide these features while keeping the error correction performance close to the finite-length coding bounds. Some frame size flexibility (from a few tens up to a few hundreds of bits) can then be achieved at the price of a tolerable overhead.

However, for such frame sizes, the code is not able to perform close to the channel capacity, which can only be approached for very long data frames. The theoretical error correction performance loss for the transmission of frames of 100 information bits is around 2 dBs in the AWGN channel and it is necessary to reach a size of 1000 bits so that the loss becomes less than 1dB [9].

In order to increase the frame size, a particular form of spatial coupling can be introduced in the TC scheme by slightly modifying the ARP interleaver model so that it has some properties similar to the well-known convolutional interleaver [10]. With this kind of interleaver, the existing TC decoder architecture could be used to decode longer frames with minor modifications. Frame sizes up to a few thousand bits will then be possible.

In conclusion, use cases requiring short to medium frame sizes, from a few tens of bits up to several thousands of bits, will be targeted for TCs. TCs are also particularly well suited to use cases where flexibility is required. The code rate flexibility is not an issue for TCs, since ARP interleavers allow rate-compatible punctured TCs with very low error floors to be designed.

Recommendations for LDPC codes

In EPIC we classify block lengths of LDPC codes into three categories: moderate block lengths i.e. 800-2k bits, long block lengths, i.e. 2k-8k bits, and ultra-long block lengths, i.e. 8k-100k bits. Smaller code block sizes (e.g. 128 bit) are not considered, since in this domain LDPC codes are clearly outperformed by Turbo codes [11].

For moderate and long block sizes, we focus on quasi-cyclic LDPC block codes, since they support efficient implementation and have been shown to achieve good performance in a variety of settings. Moderate and long block lengths are long enough to allow for good communications performance while still short enough to allow for an efficient block decoder implementation with high throughput. To achieve 1 Tb/s throughput under 1 GHz frequency constraint, at least 1000 bits need to be processed in one clock cycle resulting in a huge parallelism requirement. Depending on the block length different degrees of decoding parallelism, and thus different decoding architectures, are required to achieve Tb/s throughput. Consequently, we employ the iteration unrolled LDPC block decoder architecture (highest degree of parallelism) for moderate block lengths and a (frame-interleaved) partial-parallel LDPC block decoder (medium degree of parallelism) for long code lengths.

For ultra long block lengths, EPIC focuses on terminated LDPC convolutional codes (or spatially coupled LDPC codes), since they allow for hardware-efficient window decoding. The decoding complexity of the window decoder exploits the LDPC-CC's convolutional nature and allows the code length to potentially increase to infinity without affecting the hardware complexity. Multiple code rates will be supported so that the codes can be used in a variety of use-cases. For use-cases such as Data kiosk, Back/Front-haul, Intra-device communication and Data centers, code lengths in the order of N = 100k could be feasible and perhaps even suggested. This is mostly due to the high quality their corresponding wireless channels exhibit, as these consist of somewhat invariant system set-ups. Likewise, under these system conditions, the usage of code rates such as 3/4, 5/6 and 9/10 could thoroughly be achieved and even encouraged, in order to fully leverage the available bandwidth.

Recommendations for Polar codes

EPIC use cases have very demanding requirements in terms of throughput, BER and latency. There is a second set of requirements that EPIC imposes with respect to energy and area efficiency. Although EPIC does not set explicit targets for coding gains, it is clear that EPIC solutions must provide substantial coding gains to gain acceptance. In this preliminary study, we take a (1024,854) polar code as the initial starting point for polar coding solutions in EPIC (each codeword of this polar code carries 854 bits of user data in a frame of length 1024). Length 1024 is large enough to provide good coding gains. Furthermore, in earlier projects, we were able to implement this code on high-end FPGAs and achieve 100 Gb/s data throughput. We think 1 Tb/s will be within reach using 16 nm ASIC technology. We leave the option of using longer polar codes open if the initial ASIC implementation studies suggest that length 1024 polar codes are readily implementable without violating EPIC ASIC KPIs.

We anticipate that using polar codes with block-lengths in tens of thousands will not be feasible due to the sequential nature of decoders for polar codes. The pipeline depth will grow at least linearly with the length of the code and both hardware complexity and latency will quickly exceed the EPIC constraints. As a remedy, we plan to use concatenated schemes where an outer code (such as a single-parity check (SPC) code) is combined with inner polar codes. Such an architecture will allow constructing codes with lengths in tens of thousands and provide superior coding gains. For example, one can use a two-dimensional coding scheme where 7 copies of a (1024,854) polar code are laid out as rows and a SPC code is used for computing parity bits for each column. The overall length of this code is 8192 and the rate is (5/6)*(7/8). This type of two-dimensional coding is attractive for implementing polar codes in EPIC since decoding of the 8 rows can be carried out in parallel. If one of the eight decoders fails and means of detecting which decoder has failed is provided as part of the concatenation scheme, the SPC code can restore the data. Simulation results show that such a two-dimensional coding scheme can achieve 6 dB coding gain.

As for rate and length flexibility, the recursive u|u+v structure of polar codes can be exploited to provide some degree of flexibility. For example, the (1024,854) polar code mentioned above contains a copy of a (512,493) polar code and a copy of a (512,361) polar code. The (512,493) code in turn contains two polar codes at length 256, etc. The hardware architecture of EPIC polar encoders/decoders may mirror this recursive structure of polar codes. So, a given encoder/decoder for a polar code can be used to implement various polar codes at a set of pre-determined lengths and coding rates. In WP2, we will carry out a more detailed study of how to provide flexibility with respect to rate and code length using this approach along with concatenation.

The selected code length will follow those requirements based not so much on which of the 7 use cases is considered but rather on which of the 3 code families is investigated. On the contrary, the code rate is expected to be more specific to the different use cases. Based on different requirements in target throughput, target error rate and fluctuations of the wireless propagation, different use cases will require more or less flexibility and coding gain. Hence the suggested code rate values are discussed within the individual use cases in Section 2.2.

d) Data refinement with saturation based on realistic IC constraints

In some cases, the area number derived from the corresponding use case FEC cost is too large for realistic chip fabrication. This area number can exceed feasible chip area sizes that are realistic for an FEC IP on a System-on-a-Chip (SoC). Thus, we limit the maximum area for the FEC IP to a maximum of 10 mm² for all technology nodes.

FEC decoders with Tb/s throughput expected to be power-hungry. Removal of heat from the silicon surface becomes a major problem when the power density is increased further as technology scales down to deeply-scaled technology nodes such as 7 nm. Considering the thermal dissipation issue, when the power dissipation of device is around 5 Watt or more, a heat slug needs to be integrated on the Ball Grid Array (BGA). BGAs are substrate based packages where interconnection of the die to the substrate can either be made by wire bond Heat slug (HS) BGA or High performance Flip Chip (HFC) BGA:

- HS BGA **~ 5 to 6 W** of thermal dissipation under natural convection
- HFC BGA **~ 6 to 8 W** of thermal dissipation under natural convection

If the power dissipation exceeds the mentioned values, additional methods (e.g. heat sink / cooling fan) on system level are required. Based on the mentioned realistic area and power budget, 0.1 W/mm² is assumed as a realist power density limit. Given the assumption of an area of 10 mm² and a power density of 0.1 W/mm², a FEC decoder with 1 Tb/s throughput has a power budget of 1 Watt, an energy efficiency of 1 pJ/bit and an area efficiency of 100 Gb/s/ mm². Table 3 summarizes these implementation bounds.

Area limit	10 mm^2
Area efficiency limit	100 Gb/s/mm^2
Energy efficiency limit	~1 pJ/bit
Power density limit	0.1 W/mm^2

Table 3: Bounds on implementation KPI with realistic constraints

2.2 System and FEC KPI for Use Cases

A review of the B5G use cases studied in current technical literature and standardization work has served as a starting point for the EPIC project. The final choice of the use cases described in this deliverable was made based on several criteria: for the EPIC project, only use cases that directly require advances in the FEC technology are relevant. Furthermore, a use case is considered viable if its implementation is expected to be feasible with improvement of current chip manufacturing technologies and production cost lies within reasonable limits. The identified use cases are: data kiosk, virtual reality, intra-device communication, data centre, hybrid fiber-wireless networks, wireless fronthaul/backhaul, and high-throughput satellites. Each use case offers a specific set of challenges; collectively, these use cases therefore present a diverse set of FEC design challenges.

2.2.1 Data Kiosk

Wide coverage with high data rates has become a basic need for everyone in modern society. However, there are situations/locations where high data rates cannot be guaranteed, are simply not available, or come at a high cost. If the data to be accessed is not time-critical and can be downloaded/uploaded as a bulk, a possible solution is that the data transfer takes place at a designated station, where the user often passes by (e.g. a train station, shopping mall etc.). Such a data transfer station is called the data kiosk. We can categorize the data kiosk in two classes depending on whether the data exchange is the main reason for interacting (user connects with data kiosk for data exchange only) or whether there is another main reason (user connects with data kiosk e.g. while passing through an airport gate). The two different scenarios are illustrated in Figure 2. In both cases the user is situated in front of the machine, holding its terminal close to a marked area. However, in the first case, the time for the data exchange is not as critical as the user will not leave the data kiosk until the download is finished. In the second case as the user is not consciously connecting to the data kiosk, the data exchange should only take a very short duration of time. In the following we will focus on the second scenario.

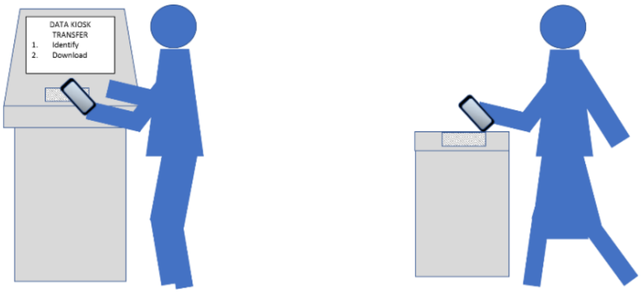


Figure 2: Two possible scenarios using a data kiosk.

The IEEE P802.15 Working Group for Wireless Personal Area Networks studies the data kiosk use case for the amendment IEEE 802.15.3d [12] [13]. The data kiosk itself is an infra-structure product. The device it is used with (mobile phone, etc.) is however an end-user product.

2.2.1.1 System Setup and Requirements

A data kiosk use case scenario has two interacting devices: The one device is the data kiosk, which is a machine installed at a fixed location with good public access, e.g. a subway station. The other device is the user terminal. Any electronic device such as mobile phones, digital cameras, computers, game devices etc. could potentially be used together with the data kiosk. However, the device needs to support wireless access, multi-stream transmission, and have sufficient processing power to allow for Tb/s throughputs.

The data kiosk is connected to the network through wired connections, allowing the transmission of high data rates without the challenges a wireless channel imposes. The connection between the user and the kiosk terminal is however wireless. The distance between the user terminal and the data kiosk is in the order of a few centimetres. There is no interference from other terminals. The wireless channel between transmitter and receiver can therefore be considered as a high-quality line-of-sight (LOS) channel which is mostly static and frequency flat [14]. But reflections can result in a more complicated channel model [13]. To achieve Tb/s throughput, the bandwidth of the transmitted signal needs to be very large. A possible candidate frequency band for the data kiosk is the 275-325 GHz frequency band [12]. The overall time during which the user terminal is connected to the data kiosk should be very short. During this time, the data kiosk needs to establish the connection, identify the user, transmit or receive the desired data, and assure that the data transfer was successful. Encryption might be relevant as well. Retransmissions in the form of HARQ can be used.

In the context of Tb/s communication, we will focus on the scenario where the user decides on initiating either a download or an upload. The traffic on the transmission link is therefore highly asymmetric. Any post-processing of the transferred data (e.g. by a video player) is done after the entire download or upload is finished. It is assumed that the user, while passing by the turnstile, is connected to the data kiosk for about 1 s. During this time depending on the channel conditions a bulk transmission of about 10-1000 Gb is expected. While the terminal is connected to the data kiosk, the terminal has only a minimal connection to other networks. This is necessary to use the full connecting capacity of the user terminal to achieve a high degree of parallelization in terms of antenna diversity.

2.2.1.2 System-level KPI

The data kiosk use case has several interesting challenges: a machine like a data kiosk will only be used if it provides clear advantages. In a society, where the user is accustomed to constantly being able to stream content with almost no delay and at high data rates, heavy downloads using a data kiosk will only be accepted if they happen in the blink of an eye. Considering the scenario where the data kiosk is collocated with e.g. a train station turnstile a reasonable assumption is that user and data kiosk are connected for about 1 second.

In terms of form factor and energy consumption, the data kiosk itself has little constraints as it is installed at a fixed location and is connected to the power supply grid. If the data kiosk is co-located with another device, e.g. a turnstile, then the placement of antennas and other parts to the data kiosk may be subject to constraints. The user terminal however is a battery powered mobile device and therefore has strong constraints regarding form factor and energy consumption [14].

Accessing the memory of the user terminal at a speed of Tb/s requires very advanced processing techniques. As for the data kiosk the transmission of the data is only in bulk, the algorithms involved can work on large amounts of data with no streaming characteristic which usually allows for simpler coding algorithms that require only a low degree of flexibility. However, there is no possibility for retransmission once the user has left the data kiosk, putting sever constraints on the BER.

KPI	Value
BER	< 10^{-12} (< 10^{-14} for offline decoding scenario)
Latency	1 s
Power	5 W
Throughput	1000 Gb/s
Cost	500 €
Flexibility (coding rate)	Moderate (rate 1/2 for offline decoding; rates 3/4 to 9/10 for real-time)

Table 4: System level KPI for the data kiosk use case

2.2.1.3 FEC-level KPI

In order to determine the KPI regarding the FEC unit we use the method described in Section 2.1. As the data kiosk itself has fewer constraints on form factor etc., the analysis is based on the mobile device only. A mobile phone is used as a reference device. Here we make the assumption that, in the future, any high-end mobile phone is going to be equipped with the possibility of downloading in a “data kiosk fashion”. Given the system-level discussion we set the goal to a throughput of 1000 Gb/s. Based on a recent report on the number of iPhones sold in 4th quarter during 2017 [15], we assume the volume value to be 50 million. Considering that to start with only medium to high-end mobile phones will have a data kiosk download feature, a price of 500 € is set per device. To be able to calculate the FEC cost, the physical chip cost percentage is set to 15% where 2% are reserved for the FEC. A mobile phone typically consumes 5 W where about 800 mW (~15%) are used for an active cell radio [16]. However, during the time the user and the data kiosk are connected, the cell radio can be given more power (~60%) as the time of connection is very short. Extremely low error rate transmission is expected, a significant amount (~30%) of power should therefore be given to the FEC unit.

As for the data kiosk use case bulks of data are transmitted without any streaming characteristic, latency cannot be quantified in the same sense as for the other use cases. The overall goal is to transmit as much data as possible during ~1 s. About 0.5 ms could be assigned to the FEC.

Due to the way the data is transmitted, very large block length can be considered and only a low degree of flexibility in terms of block length and code rate is necessary. The best protection (rate 1/2) may be associated to a fast transmission with offline decoding for a power-limited user device, provided the BER is low enough. A few higher coding rates will be sufficient if the propagation is good enough, enabling the system to switch to a more limited protection and hence finish the transmission faster for the end user, targeting high throughput first. Given the above assumptions, the FEC KPI target values are indicated in Table 5. The values exceeding the EPIC project targets are marked orange.

Data kiosk										
Device cost	500	€								
PHY chip cost percentage (A)	15	%								
FEC cost percentage (B)	2	%								
FEC cost	1.5	€								
Device power	5	W								
PHY chip power percentage (C)	60	%								
FEC power percentage (D)	30	%								
FEC power	0.9	W								
Volume	50000000				with area		with area		with area	
Throughput	1000	Gbps			limitation		limitation		limitation	
			Node	**28**	**28**	**16**	**16**	**7**	**7**	nm
			Wafer size (diam)	12	12	12	12	12	12	inch
			Full mask set price	1.5	1.5	3.2	3.2	6.5	6.5	M€
			Wafer price	5	5	11	11	22	22	k€
FEC area	**10**			**21.45**	**10.00**	**9.53**	**9.53**	**4.54**	**4.54**	**mm²**
Area efficiency	**100**			**46.62**	**100.00**	**104.98**	**104.98**	**220.08**	**220.08**	**Gbit/s/mm²**
Energy efficiency	**1**			**0.9**	**0.9**	**0.9**	**0.9**	**0.9**	**0.9**	**pJ/bit**
Power density	**0.1**			**0.042**	**0.090**	**0.094**	**0.094**	**0.198**	**0.198**	**W/mm²**

Table 5: FEC KPI for the data kiosk use case

2.2.2 Mobile Virtual Reality

Merging the real world with the digital world is what the World Economic Forum calls the “Fourth Industrial Revolution” [17]. A cornerstone embodying this idea is the technology of virtual reality and augmented reality. Virtual Reality (VR) is a technology that generates realistic images, sounds and other sensations and thus allows the user to immerse into an entirely computer-generated virtual world. Augmented Reality (AR) is a technology that immerses a user into a partially computer-generated virtual world by overlaying the real world around the user with computer-generated information. The real world is thus ‘augmented’ by virtual content. For both technologies, the immersion of the user can be achieved in different ways, including, for instance mobile phones, tablet PCs, eyeglasses, contact lenses, head-up displays etc. The most promising technologies in terms of experience so far involve a head-mounted display (e.g. the Gear VR, the Hololens, or the Oculus Go). Mobile VR/AR is an extension of AR and VR with the additional requirements that the application should run and be displayed on a mobile or a wearable device and that the application should allow real-time interaction [18].

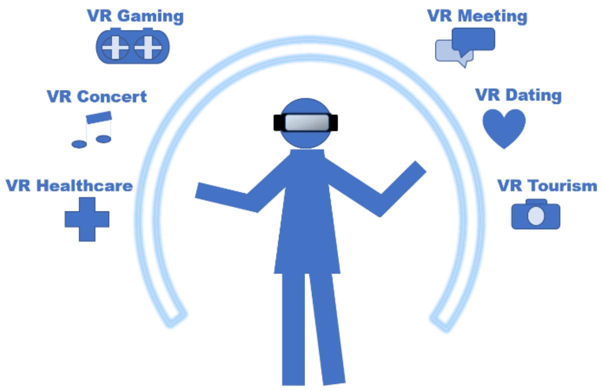


Figure 3: Virtual and augmented reality scenarios

The applications of virtual and augmented reality are wide, ranging from very basic display of information or movies on glasses, to very advanced systems involving a complex virtual layer for high-speed interactive tasks. VR and AR are therefore interesting for applications in e.g. tourism (experience a helicopter flight around New York City), education (medical students performing surgeries in VR), gaming (e.g. Pokémon Go), marketing, sports coaching, and many more. In their current form, VR and AR technologies are however not satisfactory for complex applications as the available headsets cut off the user from their surroundings, hinder mobility, and cause nausea for some users. The performance is expected to be improved significantly with the advent of 5G but for an excellent user experience transmission-, processing- and coding-techniques beyond 5G are necessary. In the following we will focus on VR.

2.2.2.1 System setup and requirements

A mobile VR system consists of both an input and an output system. Although VR could theoretically interact with all five human senses, most applications consider fewer senses. Typically, cameras, accelerometers, a microphone, a GPS, etc. provide the information for the input system allowing the system to perceive orientation, acceleration, location, motion etc. as well as both audio and video. Usually, some form of command input is accepted. The output system involves both the display and audio.

In the context of the EPIC project we are interested in very advanced VR systems that solve the problems of current VR applications concerning, for instance, discomfort and nausea caused by delay, too low resolution etc. For an advanced user experience the requirements on the system are high in many aspects. These include an extreme pixel quantity and quality, as the screen is very close to the eyes. A full 360-degree spherical view that allows the user to steer his/her view by moving both head and body. A stereoscopic display should allow for 3D vision. Moreover, the resolution of the audio should be up to human hearing capabilities and furthermore audio should be reproduced with 3D fidelity. Precise motion tracking is necessary for all the above.

Achieving all these aspects simultaneously is extremely challenging in that it involves massive data transmission and processing. Consequently, one major concern for mobile VR systems is delay. Many different types of delays are present in a VR setup, including sensor delays, network delays and rendering delays. While sensor delays have been reduced to an amount that is imperceptible to humans [19], many non-sensor-specific delays remain: most importantly, the motion-to-photon latency [20], which is defined as the time interval between a user’s physical motion (e.g. rotating the head) and the resulting update of a new frame presented on the display due to the motion. It is commonly assumed that for an acceptable VR experience the motion-to-photon latency should not exceed 60 ms [21]. However, humans start noticing lag at 13 ms [22]. Therefore, the latency should ideally be below 13 ms to be imperceptible. If the motion-to-photon latency is too high, the user experiences simulator sickness [23].

Another source of user discomfort is the Frames Per Second (FPS) rate, referring to the number of unique images shown per second of video. Regular video for film and television is usually played back at 30 FPS. This works for moderate-speed motion. For high-speed immersive experiences such as games, sports etc. video rates of 60 or even 120 FPS are needed to avoid motion blur and disorientation [24] (Current VR devices such as the HTC Vive Desire support 90 FPS).

With the above assumptions, an estimate of the data to be transmitted can be calculated as follows: Within the foveal field of view of a human, our eyes can detect fine-grained dots with a resolution of approximately 200 distinct dots per (angular) degree [25]. A reasonable estimate of how many pixels per degree this amounts to is 200 [24]. By shifting our eyes mechanically, without moving our head, our eyes can see at least 150 degrees horizontally and 120 degrees vertically. This amounts to 30,000 x 24,000 = 720 million pixels. (If we include head movement (180 degrees) and body rotation (360 degrees), this amounts to 2.5 billion pixels for a static image.) At 36 bits/pixel this corresponds to 90 Gbits per image.

For motion video, images are flashed in a sequence. Assuming video rates of 60 FPS, the eye can receive 720 million pixels for each of the 2 eyes, at 36 bits per pixel for full colour, amounting to a total of 3.1 trillion bits/s (= 3.1 terabits/s) [24]. Today’s compression algorithms can reduce the amount of data by a factor 1:300 but the remaining amount of data to be transmitted per second is still very challenging. Furthermore, it is not clear if video encoding/decoding can be carried out at such high data rates in real-time. Tb/s transmission technology reduces the need for sophisticated video compression.

Many VR devices aim at limiting the amount of data transmitted by using advanced algorithms that calculate the current image to be shown based on previous images etc. This puts a heavy burden of processing power on the head-mounted devices which can be difficult to execute given the strong limits on the form factor. An ongoing topic of research is therefore to what extent many of the calculations can be executed remotely in a cloud/server. However, remote calculations introduce additional transmission and network delays, adding to the total delay. As an alternative, we focus here on a setup where the head-mounted device has less processing power but a large memory. The images to be stored are all images that the user could possibly see by moving only the head and/or the eyes. Furthermore, a set of possible images is stored that could potentially be accessed in the very near future. As the user is moving in an entirely virtual world all these images can be pre-calculated remotely and transmitted to the VR device. The processing left for the device is then 'only' to pick the right images for the left and right eye from the database depending on the current direction and focus of the eyes. This approach shifts the burden from processing to transmission at much higher data rates and this is where a Tb/s FEC technology becomes a necessary ingredient.

For some indoor VR applications, the transmission range might only be a few meters but for outdoor applications the range could easily increase to hundred meters. Since the user is moving, the channel is neither static nor frequency flat. Moreover, if several people are using the VR application close by, interference occurs.

In general, for VR applications, the downlink channel dominates the uplink channel in terms of data rates since it involves transmitting all information for the display, meaning that the channel is bi-directional but highly asymmetric.

2.2.2.2 System-level KPI

High quality VR imposes many challenges. With the estimate of 3.1 Tb/s (uncompressed) to be transmitted, the throughput is difficult to achieve. Particularly, since the operating conditions are far from simple. Due to the movement of the user, the channel conditions are in fact very challenging. Even though a line-of-sight channel might be assumed most of the time, interference from other users can significantly impair the performance.

Compression can reduce the throughput requirements but introduces latency. Additionally, compression requires computational resources and hence power. To maximize battery life-time we need to find the best trade-off between compression and transmission rate while maintaining the required quality.

Latency has a major impact on the user experience. Both the absolute latency and the latency jitter must be kept within stringent limits to keep the user experience at acceptable levels. As most of the data has to be available in real-time, retransmissions using HARQ are not admissible. On the other hand, uncoded video for purposes of VR may be more tolerable to transmission errors, which relaxes the BER requirements on the FEC subsystem.

In terms of form factor mobile VR has very strong constraints as the head-mounted devices should only weigh a few hundred grams. The devices are wireless and therefore battery mounted which imposes severe constraints on energy consumption and power density.

Due to the very different types of information transmitted, which have different delay constraints, a high degree in flexibility is necessary for the transmission and error protection. Having flexible coding rates covering the high protection region (1/2) as well as the low-protection region (such as 15/16) and multiple intermediate values will allow the system to adapt to changing propagation conditions in real-time. The lowest coding rates may come at a reduced throughput if then need to be implemented into light devices such as head-mounted displays. Flexible rate adaptation will be needed in order to adapt to changing propagation conditions.

KPI	Value
BER	< 10^{-6}
Latency	13 ms
Power	8 W
Throughput	500 Gb/s
Cost	2000 €
Flexibility (coding rate)	High (from 1/2 to 15/16)

Table 6: System level KPI for the virtual reality use case

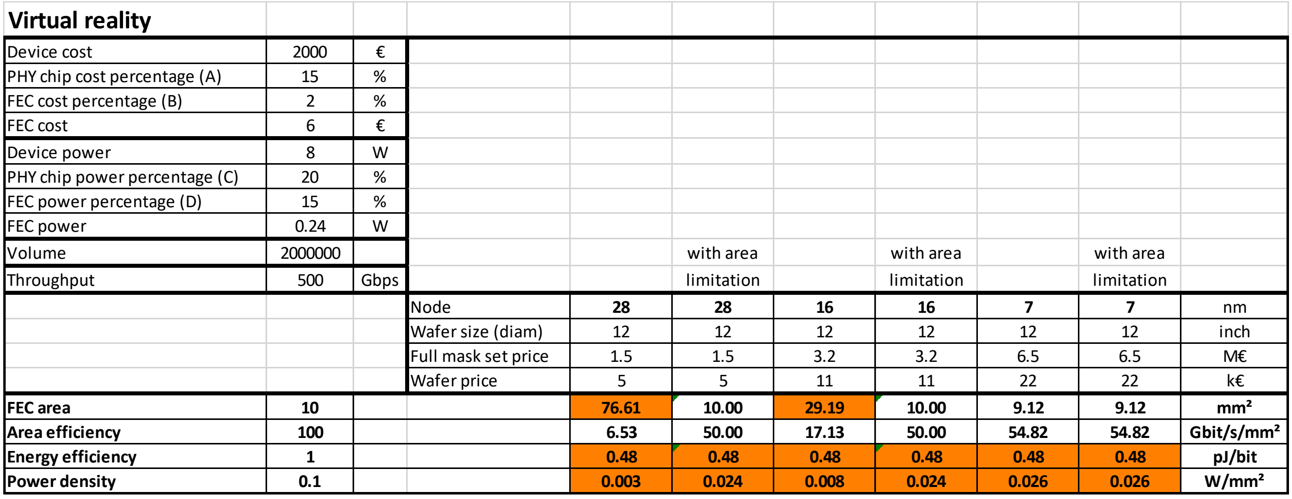
2.2.2.3 FEC-level KPI

In order to determine the KPI regarding the FEC unit we use the method described in section 2.1. On system-level a BER of less than 10^{-4} results in a quality that is rated as excellent by users [26]. For the FEC unit a BER of 10^{-6} is targeted here. Regarding the cost of the device, a HTC VIVE Business Edition is taken as a reference which currently ranges at around 1600€. Setting the goal to developing a very advanced virtual reality device, a cost of 2000€ is assumed. Considering that 2.1 million HTC Vive devices were sold in 2016, the estimated volume is set to 2 million. 15% of the cost is assumed to be needed to develop the PHY chip and FEC is assigned about 2% of the PHY chip.

In terms of power consumption, high-end virtual reality devices like the Microsoft Hololens consume around 8 W [27]. Most of the power is taken by the display but 20% are assumed to be consumed by the PHY unit, with 15% of this being assigned to the FEC unit. The reason for the rather high PHY unit power consumption is that in the use case described here, we focus on a setup where the device has less processing capability/power but instead is equipped with a large memory and can receive and transmit significantly more data.

With a latency constraint of 13 ms at system level, an estimate on FEC level latency in the order of ~0.5 ms seems reasonable. A high degree of flexibility in terms of code rate and block length (similar to mobile phone communication) is necessary.

Given the above assumptions the FEC KPI target values are indicated in Table 7. The values exceeding the EPIC targets are marked orange.



Virtual reality										
Device cost	2000	€								
PHY chip cost percentage (A)	15	%								
FEC cost percentage (B)	2	%								
FEC cost	6	€								
Device power	8	W								
PHY chip power percentage (C)	20	%								
FEC power percentage (D)	15	%								
FEC power	0.24	W								
Volume	2000000				with area		with area		with area	
Throughput	500	Gbps			limitation		limitation		limitation	
			Node	**28**	**28**	**16**	**16**	**7**	**7**	nm
			Wafer size (diam)	12	12	12	12	12	12	inch
			Full mask set price	1.5	1.5	3.2	3.2	6.5	6.5	M€
			Wafer price	5	5	11	11	22	22	k€
FEC area	**10**			**76.61**	**10.00**	**29.19**	**10.00**	**9.12**	**9.12**	**mm²**
Area efficiency	**100**			**6.53**	**50.00**	**17.13**	**50.00**	**54.82**	**54.82**	**Gbit/s/mm²**
Energy efficiency	**1**			**0.48**	**0.48**	**0.48**	**0.48**	**0.48**	**0.48**	**pJ/bit**
Power density	**0.1**			**0.003**	**0.024**	**0.008**	**0.024**	**0.026**	**0.026**	**W/mm²**

Table 7: FEC KPI for the virtual reality use case

2.2.3 *Wireless Intra-Device Communication*

Intra-device communication (IDC) refers to communication between chips on the same Printed Circuit Board (PCB) or between chips on different PCBs in close range (approximately 1 mm up to 10 cm), as illustrated in Figure 4. Currently IDC is realized through wired buses on the PCB. If IDC is realized wirelessly instead, it will allow for reduction of pins on the integrated circuit, simplified wiring on the PCB, and easier portability. Furthermore, IDC can enable high-speed connection links between two or even more boards.

In today's communication standards such as IEEE 802.11ad [5] and IEEE 802.15.3c [28] high data rates close to 10 Gb/s are achievable. In certain applications, however, communication with even higher throughput is needed. High-speed IDC is a new territory. In the literature, it is not yet possible to find an overall system design about this subject. The operational environment of IDC is very important and detailed design work needs to be carried out according to the specific requirements of each environment.

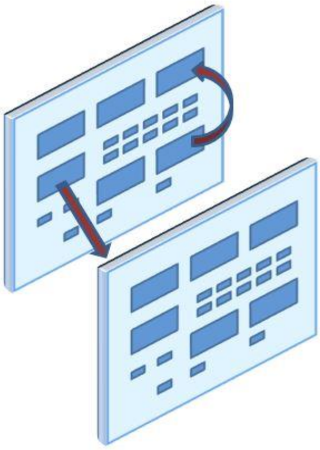


Figure 4: Communication between chips on the same device and on different devices in close range

2.2.3.1 System-level KPI

For the IDC use case, for the FEC unit we assume a BER performance around 10^{-12}. Regarding code flexibility, we assume transmission at a fixed block length but with a code rate depending on the channel conditions.

In terms of latency we consider time frames that are already prevalent in today's PCBs as a reference. When considering the connection between CPU and Random-Access Memory (RAM), for example, information must be transferred from the RAM to the CPU in a matter of nanoseconds, whenever a corresponding request has been made by the CPU. In an *Oracle Sun Fire E25K/E20K* server, for instance, the time for a single data item to be delivered from memory to a CPU, on either the same or another PCB, lies in the range of 200-300 ns [29]. Thus, the latency for wireless IDC should not exceed this value.

When it comes to power usage, we consider the sum of the power demands of the PCB itself, together with the power required for the transceiver and its signal source, as well as the power required in order to perform the FEC. We estimate this value to be 100 W, based on the values that are stated in [30].

Data rates in today's wired IDC are up to 150 Gb/s. Any wireless IDC replacement has to support similar throughput, with possibly higher peak data rates so as to accommodate retransmissions. We quantify the target throughput for this use case with approximately 500 Gb/s.

With regards to cost and volume we estimate the device cost for a PCB with multiple chips to be approximately 200 €, based on a Xilinx Spartan 6 board, which can be found on the market with a price ranging from 50 € to 700 €. The cost of a single chip should be low, due to mass production, and we estimate a value in the range of 2 € - 3 €, or 4 € in the worst case, respectively. As IDC is subject to mass production we consider a volume of 1 million as a reasonable number.

Once the system is installed, a limited flexibility will be required in coding rate, but the selected rate may depend on the actual conditions of the link, meaning that multiple values are required at design time for flexible system set-up. Moreover, the short-distance links and application requirements will put more focus on achieving a high throughput than on high coding gain.

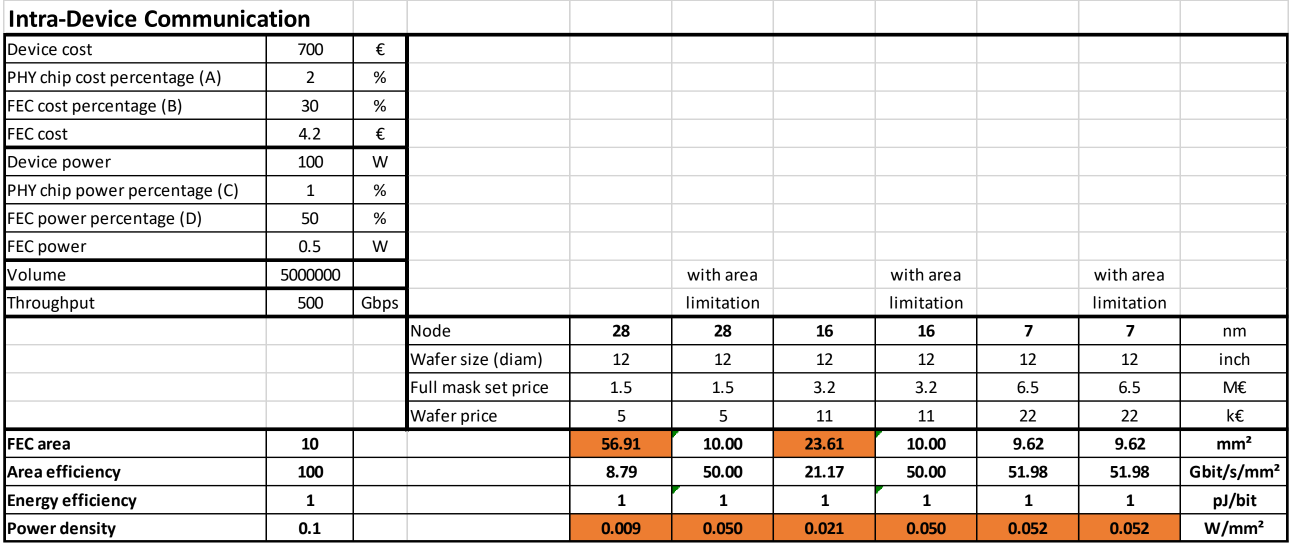
KPI	Value
BER/FER	<10^{-12}
Latency	≈200 ns-300 ns
Power	100 W
Throughput	500 Gb/s
Cost	200 €-700 €
Flexibility (coding rate)	Low (set-up phase only; 3/4 to 9/10)

Table 8: System level KPI for the intra-device communication use case

2.2.3.2 FEC-level KPI

The FEC-level KPI for the IDC are derived from the system level KPI using the methodology developed in section 2.1.

For the device cost, we estimate a worst-case value of 700 € for a PCB board with multiple chips, as the cost might be quite high for certain application scenarios, such as transferring data between the CPU and memory. In this estimation, 2% of the total device cost amount to the physical chip, with 30% of this value, in turn, appertaining to the FEC. In terms of power consumption, we estimate 100 W total power consumption, with 1% amounting to physical chip power and 0.5% to the FEC power. In line with the description given in the previous section, we assume a target latency of 100 ns - 200 ns, depending on whether I/O is incorporated or not. We furthermore assume a generally high flexibility in code length and rate. The FEC level KPI are summarized in Table 9. The values exceeding the EPIC project targets are marked orange.



Intra-Device Communication										
Device cost	700	€								
PHY chip cost percentage (A)	2	%								
FEC cost percentage (B)	30	%								
FEC cost	4.2	€								
Device power	100	W								
PHY chip power percentage (C)	1	%								
FEC power percentage (D)	50	%								
FEC power	0.5	W								
Volume	5000000				with area		with area		with area	
Throughput	500	Gbps			limitation		limitation		limitation	
			Node	**28**	**28**	**16**	**16**	**7**	**7**	nm
			Wafer size (diam)	12	12	12	12	12	12	inch
			Full mask set price	1.5	1.5	3.2	3.2	6.5	6.5	M€
			Wafer price	5	5	11	11	22	22	k€
FEC area	**10**			**56.91**	**10.00**	**23.61**	**10.00**	**9.62**	**9.62**	**mm^2**
Area efficiency	**100**			**8.79**	**50.00**	**21.17**	**50.00**	**51.98**	**51.98**	**Gbit/s/mm^2**
Energy efficiency	**1**			**1**	**1**	**1**	**1**	**1**	**1**	**pJ/bit**
Power density	**0.1**			**0.009**	**0.050**	**0.021**	**0.050**	**0.052**	**0.052**	**W/mm^2**

Table 9: FEC KPI for the intra-device communication use case

2.2.4 Wireless Backhaul/Fronthaul

Fronthaul and backhaul are key technologies that form the link between the base station antennas and the core of a cellular network. The backhaul realizes the connection between the base station Base Band Unit (BBU) and a more centralized element of the network whereas the fronthaul realizes the connection between the base station and the base station Remote Radio Head (RRH).

Two factors exist that will increase the traffic volume in the fronthaul/backhaul network. First, a trend observed in 5G/B5G systems is network densification which means that more and more small cells will be deployed to meet a traffic density target of more than 10 Mb/s per m^2. In terms of fronthaul/backhaul implementation, it is expected that a mix of wired, wireless, digital-over-optical, and radio-over-optical will be used.

Second, the concept of Network Function Virtualization (NFV) and Software Defined Networking (SDN) are two emerging technologies that will allow moving network functions to different physical nodes, thereby leading to great Capital Expenditures (CAPEX) and Operating Expenses (OPEX) savings compared to today’s solutions. E.g. it will be possible to move baseband processing from the base station to the cloud and data centers. However, this will increase the requirements on fronthaul/backhaul transmission systems in terms of traffic volume and latency. Tb/s FEC is likely to become a key technology enabler in future fronthaul/backhaul networks, both in the wireless segment (possibly in the THz frequency band) and the optical segment.

A typical base station architecture with wireless fronthaul and backhaul is illustrated in Figure 5: The User Equipment (UE) is connected to the RRH via the user link, the fronthaul realizes the connection between the RRH and the BBU, and the backhaul realizes the connection between the BBU and the core network.

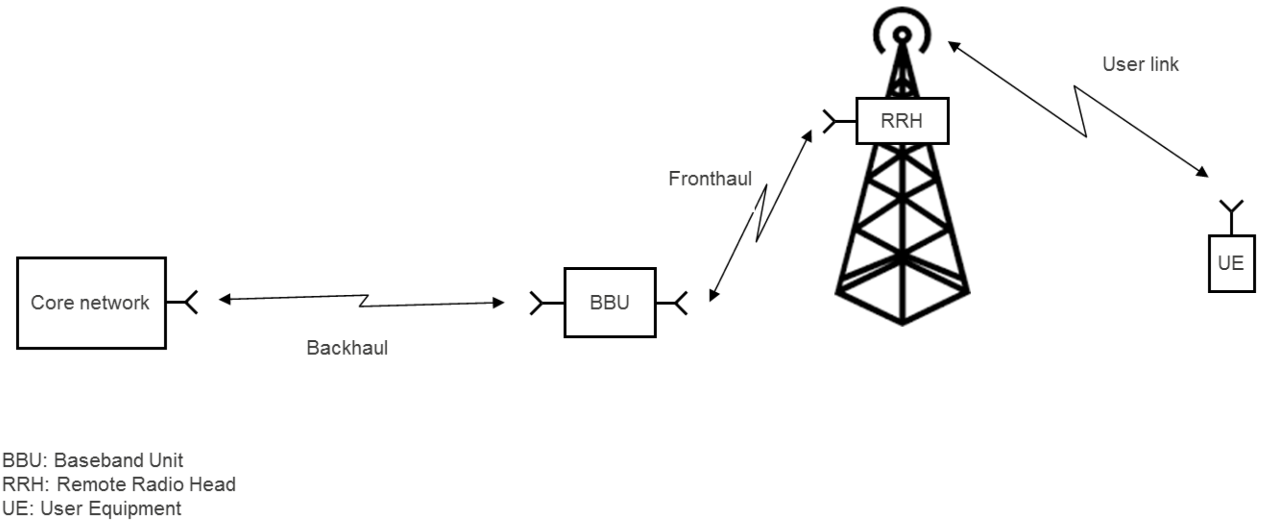


Figure 5: Typical base station architecture with wireless fronthaul and backhaul

2.2.4.1 System-level KPI

When implemented with wireless technologies, fronthaul/backhaul can be very demanding because the throughput requirements can become very high. As an example, a single 20 MHz LTE signal requires approximately 922 Mb/s for a single antenna or 7.37 Gb/s for 8 antennas. Going to higher bandwidths and significantly more antennas can easily bring these throughput requirements in the hundreds of Gb/s regime. A bandwidth of 1 Gb/s and an array with 256 antenna elements are sometimes mentioned as a typical 5G set-up. Using the numbers above, the fronthaul link would have to support throughput of 11.8 Tb/s.

Fronthaul and backhaul links are most often stationary and fixed line-of-sight links. Due to the very high operating frequency (W band (75–110 GHz), D band (110–170 GHz), possibly also 252–325 GHz), antennas with high gain are needed for transmission over a range of several hundred meters. Hence the antenna will have to have high directivity and thus the probability of interference is reduced.

The system level latency could be bounded by extrapolation from existing systems. We will consider the upcoming IEEE 802.11ay [31] standard, defining among other fronthaul and backhaul in the 60GHz band with rates of several tens of Gb/s. The latency requirement (SIFS) of IEEE 802.11ay is 3 µs. The Short Interframe Space (SIFS) in IEEE 802.11ay is the amount of time required for a wireless interface to process a received frame and to respond with a response frame. It is the difference in time between the first symbol of the response frame in the air and the last symbol of the received frame in the air. This includes all PHY and MAC delays.

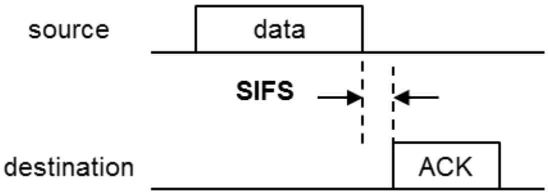


Figure 6: Latency defined by the SIFS in IEEE 802.11ay

The maximum rate of IEEE 802.11ay is approximately 34 Gb/s. By extrapolation, the latency for backhaul (250 Gb/s) would be 3 µs x 34 / 250 = 0.4 µs and for fronthaul (1000 Gb/s) 3µs x 34 / 1000 = 0.1 µs. However, there is no need to down-scale latency requirements linearly with throughput as data packets will be larger for efficient high-throughput fronthaul/backhaul link and hence not require extremely fast ACK messages. Latencies of the order of 0.1 µs would be exceeded simply by the propagation delay of the signals at the speed of light. More realistic targets for fronthaul/backhaul links mention at least 10 µs for the whole system [32] [33]such that 1 µs for FEC is sufficient.

Fronthaul and backhaul are infrastructure links; hence the cost constraint is not as stringent as for consumer devices. Still, due to the trend towards network densification, the proliferation of small cells puts a significant pressure on the cost of base station components. Similarly, as the number of infrastructure nodes increases in a dense network, the power consumption per node must decrease to keep the overall power consumption at a reasonable level.

Being an infrastructure component, fronthaul and backhaul should not degrade the overall performance of the link reliability. Hence, very low error-probability in the range of 10^{-13} is needed, which has a strong impact on the performance requirements of candidate coding schemes. Limited run-time flexibility is expected on the coding parameters but having a few rates will enable an optimization of each individual link based on its configuration. It will be adapted when conditions change but fast run-time adaptation is not expected.

The system level KPI detailed in Table 10 for the backhaul use case and in Table 11 for the fronthaul use case.

KPI	Value
BER	$<10^{-13}$
Latency	1 µs
Power	100 W
Throughput	250 Gb/s
Cost	10000 €
Flexibility (coding rate)	Moderate (set-up phase mostly, 3/4 to 9/10)

Table 10: System level KPI for the backhaul use case

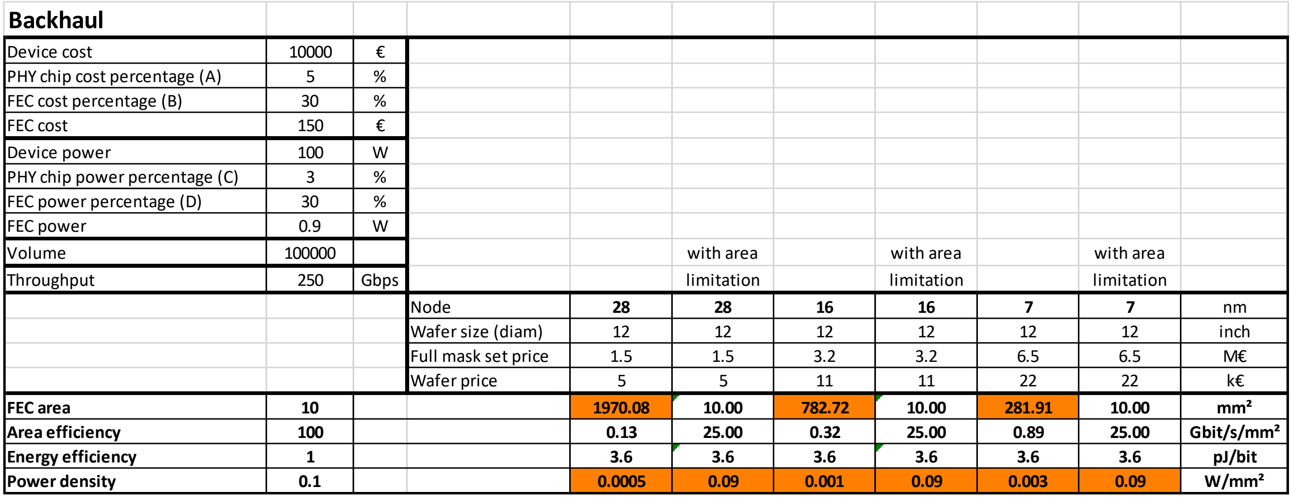
KPI	Value
BER	$< 10^{-12}$
Latency	1 µs
Power	20 W
Throughput	1000 Gb/s
Cost	1000 €
Flexibility (coding rate)	Moderate (set-up phase mostly, 3/4 to 9/10)

Table 11: System level KPI for the fronthaul use case

2.2.4.2 FEC-level KPI

The FEC level KPI for the fronthaul and backhaul were derived from the system level KPI using the methodology developed in section 2.1. The results are shown in Table 12 for the backhaul use case and in Table 13 for the fronthaul use case. The values exceeding the EPIC targets are marked orange.

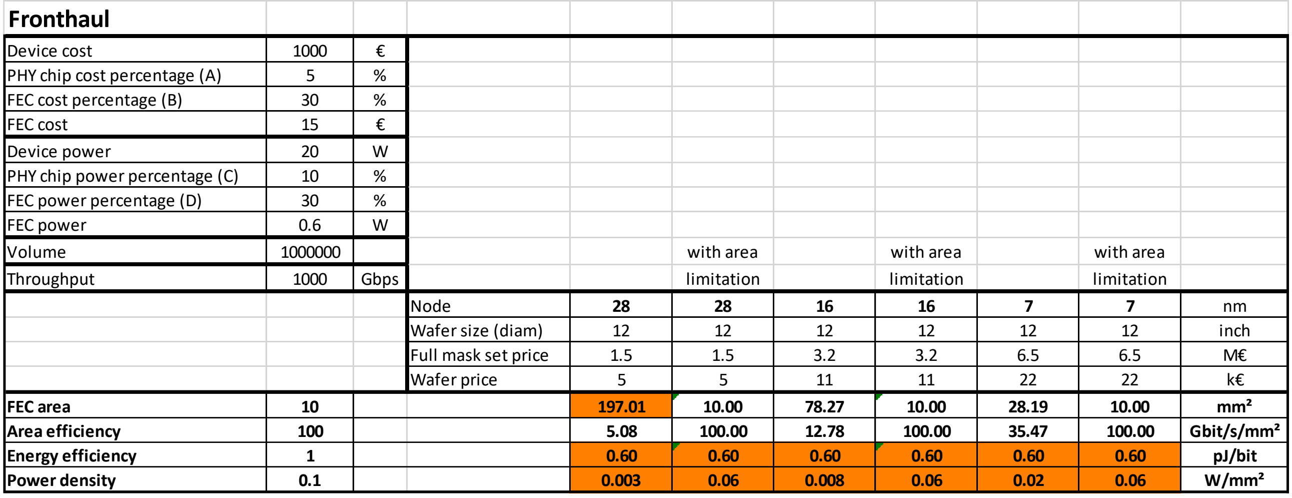
The resulting area and energy efficiencies are within the EPIC targets but the power density is a bit beyond EPIC target (quite logically since the area was reduced by a huge factor).



Backhaul										
Device cost	10000	€								
PHY chip cost percentage (A)	5	%								
FEC cost percentage (B)	30	%								
FEC cost	150	€								
Device power	100	W								
PHY chip power percentage (C)	3	%								
FEC power percentage (D)	30	%								
FEC power	0.9	W								
Volume	100000				with area		with area		with area	
Throughput	250	Gbps			limitation		limitation		limitation	
			Node	**28**	**28**	**16**	**16**	**7**	**7**	nm
			Wafer size (diam)	12	12	12	12	12	12	inch
			Full mask set price	1.5	1.5	3.2	3.2	6.5	6.5	M€
			Wafer price	5	5	11	11	22	22	k€
FEC area	**10**			**1970.08**	**10.00**	**782.72**	**10.00**	**281.91**	**10.00**	**mm²**
Area efficiency	**100**			**0.13**	**25.00**	**0.32**	**25.00**	**0.89**	**25.00**	**Gbit/s/mm²**
Energy efficiency	**1**			**3.6**	**3.6**	**3.6**	**3.6**	**3.6**	**3.6**	**pJ/bit**
Power density	**0.1**			**0.0005**	**0.09**	**0.001**	**0.09**	**0.003**	**0.09**	**W/mm²**

Table 12: FEC KPI for the backhaul use case

For the fronthaul, the allowed FEC area is also too large (but not as much as for the backhaul) and has been forced down to 10 mm². The resulting area efficiency is within the EPIC targets but the energy efficiency and power density are slightly above the EPIC target. This is due to the very high throughput requirement (1 Tb/s).



Fronthaul										
Device cost	1000	€								
PHY chip cost percentage (A)	5	%								
FEC cost percentage (B)	30	%								
FEC cost	15	€								
Device power	20	W								
PHY chip power percentage (C)	10	%								
FEC power percentage (D)	30	%								
FEC power	0.6	W								
Volume	1000000				with area		with area		with area	
Throughput	1000	Gbps			limitation		limitation		limitation	
			Node	**28**	**28**	**16**	**16**	**7**	**7**	nm
			Wafer size (diam)	12	12	12	12	12	12	inch
			Full mask set price	1.5	1.5	3.2	3.2	6.5	6.5	M€
			Wafer price	5	5	11	11	22	22	k€
FEC area	**10**			**197.01**	**10.00**	**78.27**	**10.00**	**28.19**	**10.00**	**mm²**
Area efficiency	**100**			**5.08**	**100.00**	**12.78**	**100.00**	**35.47**	**100.00**	**Gbit/s/mm²**
Energy efficiency	**1**			**0.60**	**0.60**	**0.60**	**0.60**	**0.60**	**0.60**	**pJ/bit**
Power density	**0.1**			**0.003**	**0.06**	**0.008**	**0.06**	**0.02**	**0.06**	**W/mm²**

Table 13: FEC KPI for the fronthaul use case

2.2.5 Data Centers

Data Centers (DC) are main components of the IT systems that power various critical operations of institutions from enterprises of all sizes to internet-service providers. The DCs host storage, processing/computation functions and various applications under one umbrella. They also serve end-user sub-systems or users. The storage-, processing- and computation functions of a DC usually scale orders of magnitude higher than conventional server-client set-ups. This rather central hosting and computation operation necessitates substantial processing power capabilities in the DCs, as well as robust, high throughput, and low-latency connectivity in the DC networks.

A high-level architecture of a typical DC is shown in Figure 7 below. State-of-the-art DC designs follow a layered approach which has proven to satisfy high-performance requirements of DCs, including scalability, flexibility, resilience, and maintenance [32]. The connectivity in each layer—core, aggregation, and access—plays a crucial role in the DC operations with each layer having varying connectivity requirements. For instance, the core layer, which is responsible of high-speed packet routing for all in-and-out traffic flows, mostly requires an order of magnitude higher throughput (e.g. 10 Gigabit Ethernet (GbE) [32]) compared with the access layer where local servers are connected to each other with relatively low throughput requirements (GbE [32]). On the other hand, as discussed below, the recent evolution of high performance DCs necessitates further enhancements in scalability. This places wireless technologies as potential connectivity solutions, replacing wired/cable connectivity technologies in at least part of DCs (e.g. 10 GbE and GbE in Figure 7.

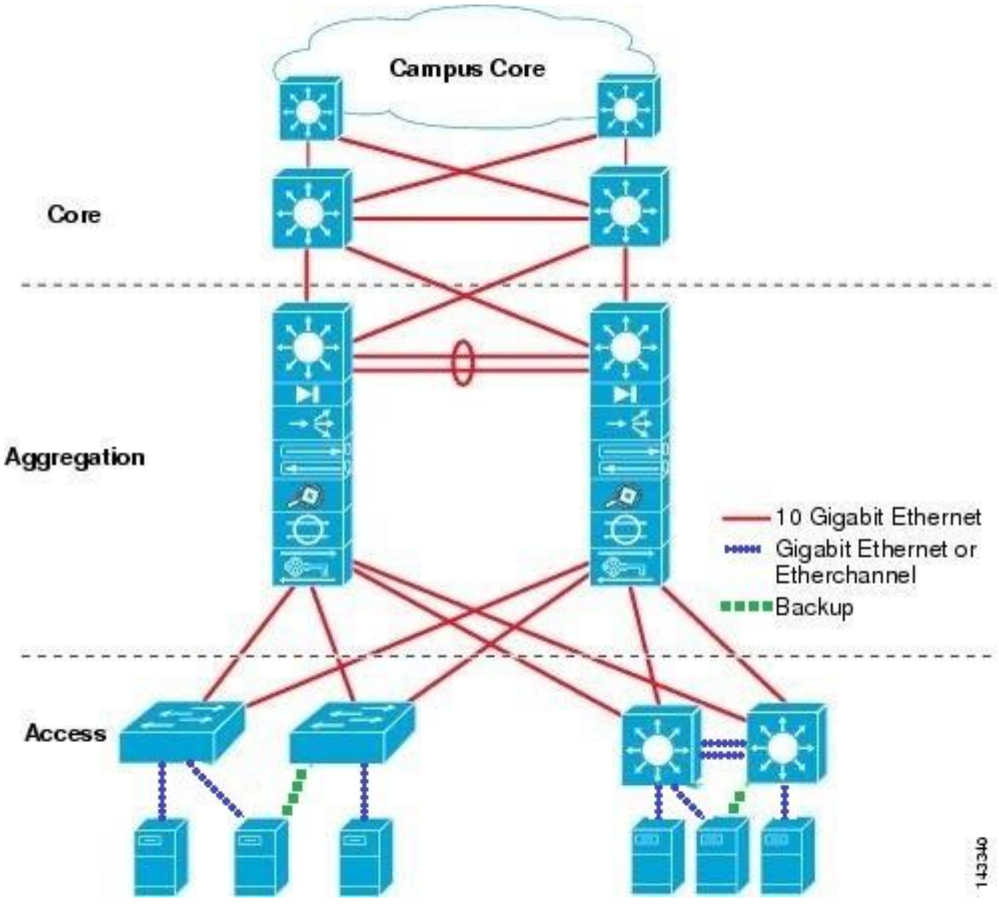


Figure 7: A typical data centre high level architecture by Cisco

2.2.5.1 System set-up and requirements

Typical data centers are mainly constructed by static placement of server racks under one or multiple close-by physical locations. The capability of the DCs in terms of computation power and storage are determined by the number and individual capacity of the servers and the racks that are allocated for specific applications and services. In a typical DC, the server racks are inter-connected via wire to ensure the bandwidth and robustness that are critical in the execution of large amounts of operations. The standard intra-DC connections are currently deployed by direct Peripheral Component Interconnect (PCI) links or 10G Ethernet, connecting the servers placed in the racks ranging from tens of centimetres as well as the racks themselves, with distances in the orders of meters [32].

On the other hand, the ever-increasing application of virtualization techniques in the DCs, along with the cloud computing services, has started to alter the standard DC network architectures. The DC networks are destined to enable virtual and physical distributed and dynamic server and rack clustering features, thereby providing on-demand and high-performance computation and storage facilities. Applications of wireless connectivity solutions within and/or between server racks are therefore seen as a possible solution to the incumbent wired connections [33], [34]. Several critical requirements in a DC, e.g. physical protection, heat removal, energy consumption, etc., impose relatively strict constraints on the indoor physical deployment of server racks. As shown in Figure 8, the racks are statically positioned to allow LOS connectivity, multi-hop operation or indirect LOS transmission with 3D beamforming [34].

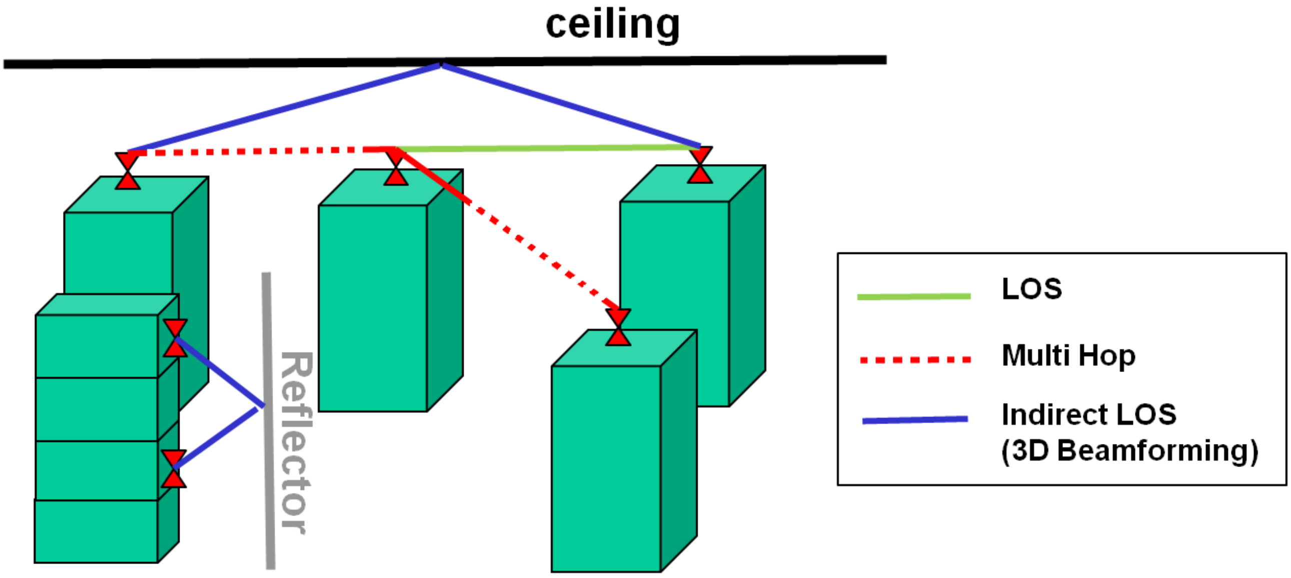


Figure 8: Server racks deployment and possible connectivity models

A wired link replacement in a DC set-up would clearly necessitate directional and ultra-high throughput wireless links. Currently available bands such as 275–320 GHz can be the primary candidates for the wireless link frequency bands.

2.2.5.2 System-level KPI

The evolution of DCs from centralized and static deployment scenarios towards distributed and virtualized instances impose additional constraints particularly in the transmission range of the links. The key challenges of a potentially robust wireless link technology for DCs are the ultra-high throughput, very low error-probability, and low latency guarantees. It is anticipated that future DC farms will host connectivity ranges as far as 100 m, which requires a careful design and deployment of THz point-to-point systems, including a robust and low-latency FEC solution. In the following, we provide information on the most important system KPI requirements.

DCs are the key information sinks and carry out high-accuracy operations, such as authentication procedures, BERs in the order of 10^{-13} are targeted [34]. On the other hand, wireless fixed-beams are most feasible solutions for DCs considering the static nature of the deployment, hence resulting in stable (interference-free) channels with almost constant SNR. The requirements on the code flexibility are therefore rather low.

The link-level latency for DCs greatly depends on the operating connectivity solution. For the Ethernet solutions with inter-process communication (IPC), with enabling technologies such as remote direct memory access (RDMA), and Internet Wide-area RDMA Protocol (iWARP), the latencies in the order of 3-5 µs can be achieved. Moreover, proprietary solutions that achieve sub-3 µs latency figures can also be found [35]. As for the direct PCI link technology, latency figures as low as 500 ns have been commercially available [36].

Various server types can be used in the overall architecture of a DC. For the sake of analysis, we assume a standard dual-processor 2U server, which is commonly used in DCs, and consumes a total of 450 W as a unit. As demonstrated in [37], PCI card of this standard 2U server consumes 41 W.

State-of-the art link-level communication in DCs is mostly based on direct PCI links or 100G Ethernet technologies [38]. With the constantly increasing computation power requirements of the DCs, the link capacities shall ideally support Tb/s throughput and be a complimentary solution to the wired links [34]. Due to the wide range of specification and vendor selection, the 2U server prices can vary significantly. However, an average price in the order of €2500 can be observed in the commercial markets.

Once the system is installed, a limited flexibility will be required in coding rate, but the selected rate may depend on the actual conditions of the link and be optimized during the set-up phase.

KPI	Value
BER	$< 10^{-13}$
Latency	0.5 µs
Power	50 W
Throughput	1000 Gb/s
Cost	2500 €
Flexibility (coding rate)	low (set-up phase only; 3/4 to 9/10)

Table 14: System level KPI for the data center use case

2.2.5.3 FEC-level KPI

Using the methodology described in section 2.1, in this section, we identify FEC-level KPI of Data centre use case. Table 15 depicts the critical FEC KPI based on the system-level KPI analysed in section 2.2.5.2. The values exceeding the EPIC project targets are marked orange. As it can be observed, with 7 nm technology, the area and power density requirements shall be met. On the other hand, we observe that area efficiency and energy efficiency figures are slightly higher than the target EPIC KPI.

Data Center										
Device cost	2500	€								
PHY chip cost percentage (A)	1	%								
FEC cost percentage (B)	10	%								
FEC cost	2.5	€								
Device power	50	W								
PHY chip power percentage (C)	5	%								
FEC power percentage (D)	30	%								
FEC power	0.75	W								
Volume	10000000				with area		with area		with area	
Throughput	1000	Gbps			limitation		limitation		limitation	
			Node	**28**	**28**	**16**	**16**	**7**	**7**	nm
			Wafer size (diam)	12	12	12	12	12	12	inch
			Full mask set price	1.5	1.5	3.2	3.2	6.5	6.5	M€
			Wafer price	5	5	11	11	22	22	k€
FEC area	**10**			**34.29**	**10.00**	**14.46**	**10.00**	**6.14**	**6.14**	**mm^2**
Area efficiency	**100**			**29.16**	**100.00**	**69.15**	**100.00**	**162.98**	**162.98**	**Gbit/s/mm^2**
Energy efficiency	**1**			**0.75**	**0.75**	**0.75**	**0.75**	**0.75**	**0.75**	**pJ/bit**
Power density	**0.1**			**0.022**	**0.075**	**0.052**	**0.075**	**0.122**	**0.122**	**W/mm^2**

Table 15: FEC KPI for the data center use case

2.2.6 Hybrid Fiber-Wireless Networks

Fibber-optic technology is currently the primary connectivity solution for the majority of high-speed telecommunication systems and sub-systems including fixed-line infrastructure (e.g. core to end-user internet distribution), mobile infrastructure, and data centers. Commercial fiber-optic solutions, particularly Time Division Multiplexing-based Gigabyte Passive Optical Networks (TDM GPON) already offer hundreds of Gb/s throughputs with close-to-market proof-of-concept solutions delivering multiples of Tb/s data rates [39][24]. However, issues such as cost and physical limitations on deployment (e.g. due to terrain and/or permission requirements) put significant constraints on the utility of fiber solutions. Therefore, wireless replacements and/or extensions of high-throughput communication links are seen as substantial complimentary solutions to the underlying fiber-optic links. This trend is expected to become more relevant in 5G and B5G systems with the evolution of communication networks into smaller (e.g. micro, nano-networks), distributed, and re-configurable deployments which result in additional burdens on the physical installation of fiber-optic cables in various parts of the networks. To address such issues, hybrid fiber-wireless solutions will be used ubiquitously in 5G and B5G systems.

Figure 9 depicts a utilization of wireless connectivity solution along with the incumbent wired links in a conventional telecommunication transport network. In this example, the distribution of data from Point-of-Presence (PoP) to the access network is provided through a hybrid fiber-wireless solution.

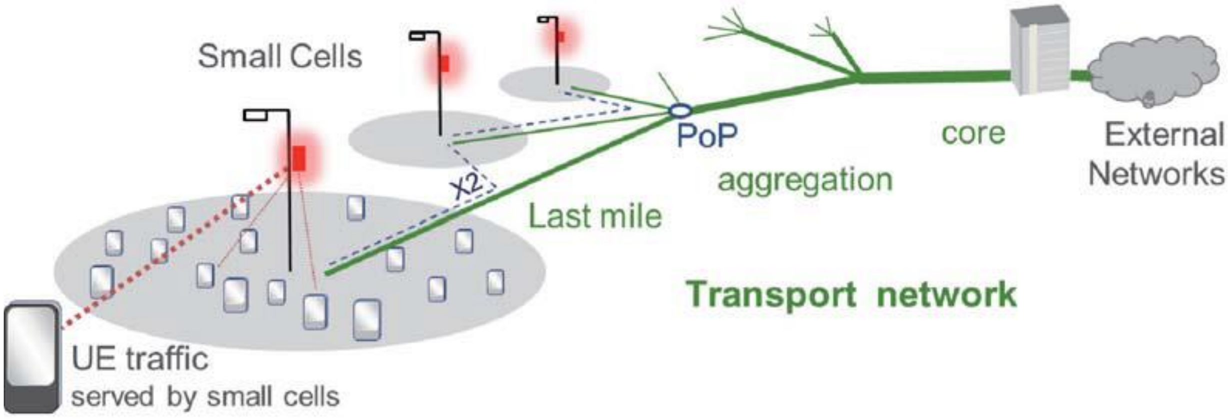


Figure 9: Hybrid fiber-wireless links in a telecommunication transport network

2.2.6.1 System set-up and requirements

A hybrid fiber-wireless network consists of fiber-optic and wireless transport components as well as an interface between these elements. The interfaces are particularly important for seamless fiber-wireless end-to-end systems as efficient translation of fiber-optic signals into wireless signals is crucial. In that regard, the ITU-T G.709 specification [39] already provides a clear description and details of high throughput fiber-optic interfaces, yet effective translation interfaces are part of active research [25]. On the other hand, the key requirements of potential hybrid fiber-wireless links to the incumbent and future fiber-optic link directly relate to the latter's performance requirements. For instance, in terms of throughput and error performance, an admissible wireless solution should have similar performance as the fiber optic links.

An important baseline model to identify the requirements is obtained from the SoA fiber-optic solutions and its expected evolutions for 5G-and-beyond systems. In that regard, a close look at the ITU-T G.709, which defines the technical requirements for fiber communication standards, provides benchmark requirements for the hybrid fiber-wireless links as well [39].

2.2.6.2 System-level KPI

In the following, we provide details on the system level KPI for the hybrid fiber-wireless use case, mainly based on the ITU-T G.709 specifications.

In terms of error performance, a maximum BER in the order of 10^{-12} is considered to be necessary considering the requirements on fiber-optic component set in [39]. Moreover, due to the wireless channel characteristics such as relatively less stable channel coefficients, interference degradations, higher path-loss, etc., the wireless link extension should also support a flexible Modulation and Coding Scheme (MCS) assignment.

Latency requirements significantly depend on the application scenarios at hand, however very stringent latency figures, e.g. in the order of smallest frame periods in G.709 standard, e.g. ODU_period = 4µs can be considered as an upper bound latency benchmark for the overall L1 communications.

For the power estimate, we consider a commercial fiber transceiver that is able to support 100 Gb/s throughput with specification given in [40] and power consumption around 5W.

High throughput operations, in the Tb/s range, manifest itself as the first critical challenge of hybrid fiber-wireless solutions, therefore placing THz bands, 275 GHz and above, as the leading spectrum for wireless solutions in order to harvest the available frequency bands.

We assume a benchmark cost figure based on the average cost of a 100 Gb/s fiber-optic transceiver in the current commercial market, which is around 1000€.

Limited run-time flexibility is expected on the coding parameters but having a few rates will enable an optimization of each individual link based on its configuration. It will be adapted when conditions change but fast run-time adaptation is not expected.

KPI	Value
BER	$< 10^{-12}$
Latency	1 µs
Power	5 W
Throughput	1000 Gb/s
Cost	1000 €

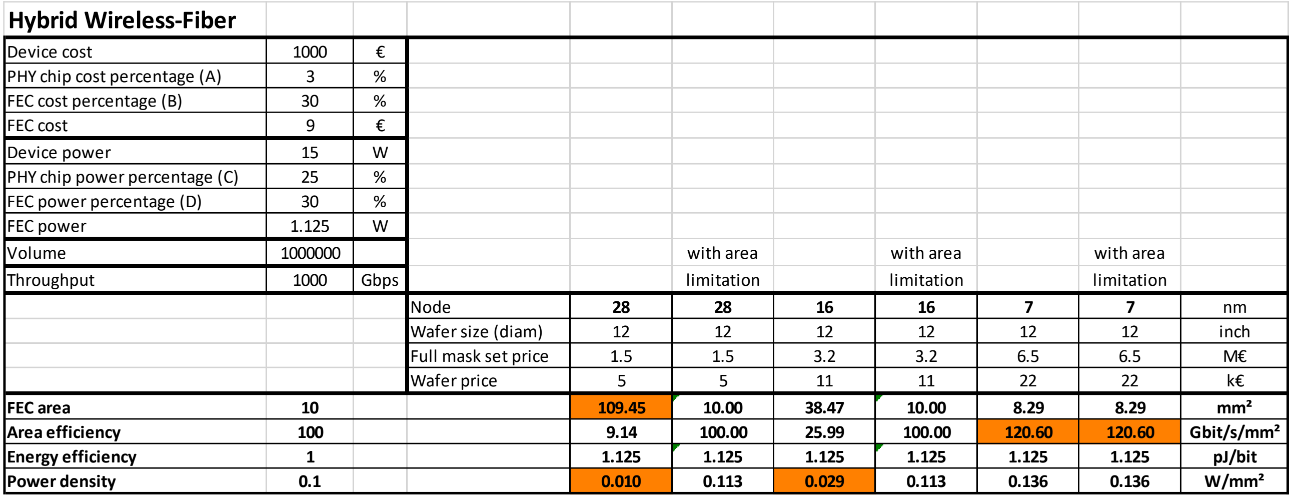
KPI	Value
Flexibility (coding rate)	Moderate (set-up phase mostly, 3/4 to 9/10)

Table 16: System level KPI for the hybrid fiber-wireless networks use case

2.2.6.3 FEC-level KPI

Using the methodology described in section 2.1, in this section, we identify FEC-level KPI of hybrid fiber-wireless use case [40]. The values exceeding the EPIC project targets are marked orange.

The FEC-level KPI are summarized in Table 17. Here, the values exceeding the EPIC targets are marked orange. It can be observed that the estimated KPI requirements for the hybrid fiber-wireless use case mostly satisfy the EPIC requirement set. The energy efficiency figure, e.g. 1.125 pJ/bit, on the other hand manifests itself as a rather stringent design constraint.



Hybrid Wireless-Fiber										
Device cost	1000	€								
PHY chip cost percentage (A)	3	%								
FEC cost percentage (B)	30	%								
FEC cost	9	€								
Device power	15	W								
PHY chip power percentage (C)	25	%								
FEC power percentage (D)	30	%								
FEC power	1.125	W								
Volume	1000000				with area		with area		with area	
Throughput	1000	Gbps			limitation		limitation		limitation	
			Node	**28**	**28**	**16**	**16**	**7**	**7**	nm
			Wafer size (diam)	12	12	12	12	12	12	inch
			Full mask set price	1.5	1.5	3.2	3.2	6.5	6.5	M€
			Wafer price	5	5	11	11	22	22	k€
FEC area	**10**			**109.45**	**10.00**	**38.47**	**10.00**	**8.29**	**8.29**	**mm²**
Area efficiency	**100**			**9.14**	**100.00**	**25.99**	**100.00**	**120.60**	**120.60**	**Gbit/s/mm²**
Energy efficiency	**1**			**1.125**	**1.125**	**1.125**	**1.125**	**1.125**	**1.125**	**pJ/bit**
Power density	**0.1**			**0.010**	**0.113**	**0.029**	**0.113**	**0.136**	**0.136**	**W/mm²**

Table 17: FEC KPI for the hybrid fiber-wireless Networks use case

2.2.7 High-Throughput Satellites

High-throughput Satellites (HTS) are a new generation of satellite communication systems, which deliver a throughput several times higher than conventional Fixed Satellite Services (FSS) at a lower cost. In 2011, the HTS ViaSat-1 could deliver 140 Gb/s of aggregated capacity, which by the time, was more than the combined capacity of all FSS satellites of North America [41]. For 2019, the first of 3 HTS satellites comprising the ViaSat-3 project is expected to be operating, each of which is anticipated to have as much bandwidth as all the rest of the satellites in the world combined, including all of the HTS that are now under construction, according to ViaSat's CEO Mark Dankberg [42]. Each of the ViaSat-3 satellites is expected to deliver over 1 Tb/s of network capacity.

One of their main characteristics is the use of multiple smaller spot beams instead of one large beam over a specific coverage area, as illustrated in Figure 10. Non-adjacent spot beams (shown with the same colour) re-use the same frequency segment of the available spectrum, increasing the overall bandwidth. Additionally, narrower beams have a higher power density per area and thus higher spectral efficiency for each beam. Together, these two aspects contribute to the higher throughput that HTS achieve.

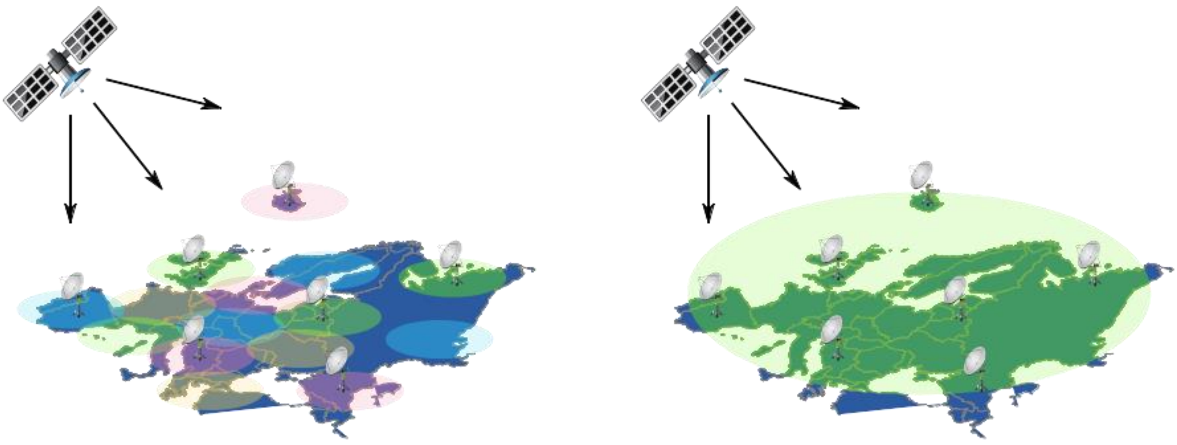


Figure 10: High-throughput satellite scenario

As communication techniques have changed from analogue to digital, satellite systems start introducing on-board digital signal processing. This not only serves for regeneration and routing of the transmitted information, but introduces flexibility as well with respect to the kind of services that a satellite network can provide.

In the context of 5G and B5G, these two aspects make HTS a good candidate to provide connectivity to larger areas, where terrestrial networks have limitations or simply have no reach, such as e.g. commercial aircrafts. This endeavour of integrating satellite into the 5G network architecture already has on-going initiatives, such as the “Satellite and Terrestrial Network for 5G (SaT5G) project” [43]. The goal is to make satellite as easy to integrate into the broader telecom network as any other 5G compliant technology, becoming a true “plug-and-play” solution [30]. This integration requires that satellites can handle a lot more capacity, which is how HTS come into play.

"The H2020 program for satellite communication technologies is currently encouraging research of HTS for these purposes, with topics such as higher frequency bands, as well as very high throughput optical feeder links up to 1 Tb/s " [44]. In this context, the coding techniques developed in the EPIC project could therefore be of high relevance in these endeavours.

2.2.7.1 System-level KPI

BER and Flexibility: Current standards for satellite communications, such as DVB-S and DVB-S2, provide Quasi-Error-Free (QEF) quality of transmission. This means less than one uncorrected error event per hour, which corresponds to a BER of 10^{-10} to 10^{-11} [45]. At frame level, a FER of 10^{-5} and even FER down to 10^{-7} are achievable at an SNR in the range of -3 to 20 dB for the DVB-S2X standard [46]. A typical system must therefore be highly flexible in terms of code-rate, whereas the code-length remains mostly fixed.

Most communication satellites are located at a geo-stationary orbit (GEO), around 35786 km above the earth's equator [45]. At this altitude they orbit at the same angular speed as the earth, appearing almost fixed in the sky to an observer on the ground. However, this comprises a delay in the communications, due to the propagation speed of radio waves. Although these travel at the speed of light (~300000 km/s), it takes them 120 ms to travel that distance. Therefore, communicating via a GEO satellite directly overhead at the equator has an approximate 240 ms delay, which can increase up to 280 ms depending on how far the ground stations are with respect to each other. Bi-directional communication would have delays of at least half a second. Satellites in lower orbits have much lower propagation delays, ranging from 73 ms to 10 ms. However, unlike GEO satellites, these ones move across the sky with respect to the ground, thus constellations of several of these are necessary to ensure coverage of a specific area.

HTS are characterized by their multi-beam technology and frequency re-use, which consists of using the same frequency band several times, in a way that the total capacity is increased without increasing the allocated bandwidth. One satellite can thereby cover a specific area using tens or even hundreds of narrow beams, with a greater capacity than if a single global beam had been used instead. Currently, one beam with 3,5GHz of bandwidth in Ka-band [47] can have a throughput of 10 Gb/s, assuming a 16-PSK modulation (supported by both DVB-S2 and DVB-S2X). Therefore, one satellite can have 100 to 1000 Gb/s of aggregated capacity [48] [49]. However, depending on the type of services and network configuration, this capacity could be unevenly distributed, where some beams alone could require more capacity than others. A good example are HTS networks configured in star topologies for direct to home (DTH) services as well as video-on-demand and interactive TV (iTV). In this case, a feeder earth station or hub could be serving an area corresponding to several spot-beams at once, which would therefore require an individual capacity of 10x or even 100x Gb/s. On the other hand, a constellation of several HTS, deployed for global coverage, could have inter-satellite communication links of 100x Gb/s between them, as a form of "inter-satellite backhaul".

Although a single link in Ka-band does not have enough bandwidth to deliver the necessary throughput, using the higher Q-/V-bands and even W-band in near-future satellite systems could potentially solve this issue [50] [51] [29]. Large bandwidth availability is the main reason why these bands are becoming attractive, as that part of the spectrum hasn't been densely allocated yet [52]. On the other hand, various technical challenges such as channel characterization and hardware requirements must be addressed before Q-/V-/W-band solutions are widely deployed [50].

Today’s GEO communications satellites have power consumption in the range of 7 to 15 kW [53], where a significant portion is used by the payload for amplification and transmission. For this use case, a total power consumption of 10 kW is assumed, of which 0,01% is used for the PHY chip, and 50% thereof for the FEC unit.

Unlike most commoditized products, HTS and satellites in general are not mass-produced, as they are extremely specialized and costly. Some estimates indicate that 300 satellites with a mass of over 50kg will be launched on average each year by 2026 [54]. From these numbers, a conservative and yet visionary estimation on the number of HTS produced can be made of 100 to 1000 HTS over the next decade.

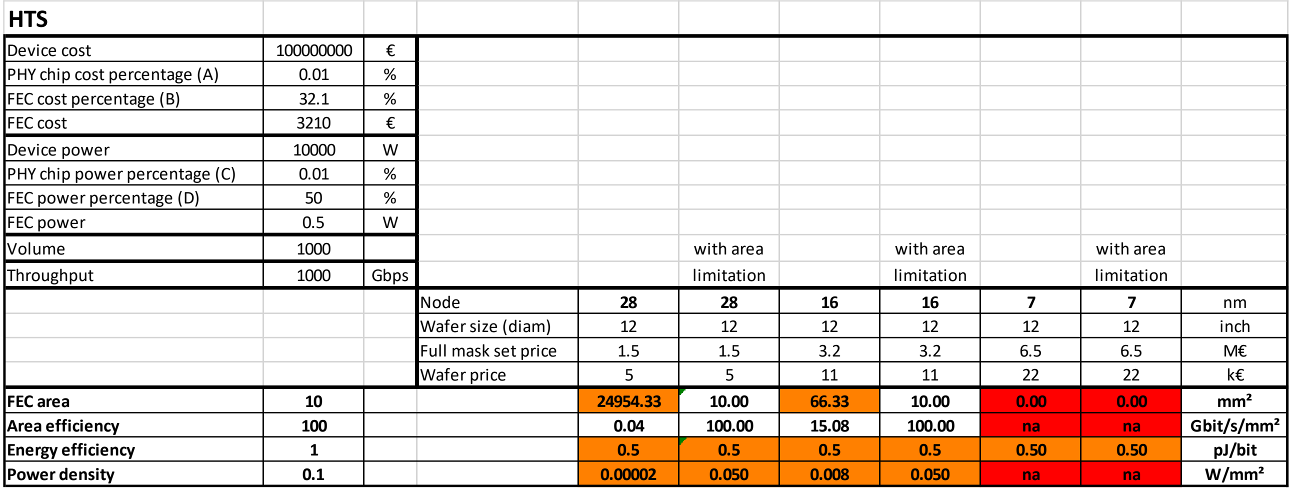
In terms of cost, most communications satellite projects range from $300 - $600 million, including the spacecraft, launch and launch insurance. These are typically high, up-front and fixed costs, with unique risk factors, which are typically recouped over the expected 15-year lifetime of the satellite [55]. A very conservative estimate of $100 million of manufacture costs for an average GEO satellite can be reasonable. However, depending on their complexity, size, features and other factors, this value could increase significantly. From this cost, a 0,01% is assumed to be needed for the PHY chip, of which about 32% is assumed to be FEC costs.

2.2.7.2 FEC-level KPI

In order to determine the FEC-level KPI, we first used the method described in section 2.1. Given the assumptions for power, cost and volume made above, Table 18 with the target FEC-level KPI was obtained. The values exceeding the EPIC targets are marked orange.

From these results, it can be seen that an ASIC implementation with 7 nm technology is not feasible, indicated by FEC area of 0 mm^2, which means that the costs are not sufficient. Those non-feasible entries related to area are marked red in Table 18. On the other hand, a FEC area greater than the targeted value (10 mm^2) could in theory be afforded for the costs assumed above in 16 nm and 28 nm technology. However, these are very marginal results, corresponding to a best-case scenario of 1000 units, whereas in practice the volume could be considerably lower. For volumes lower than 800 and 300, manufacture is already unfeasible in 16 nm and 28 nm technology respectively. Nevertheless, ASIC implementations for these volumes could be afforded by means of Multi-Project Wafer (MPW) services [56]. On these services, several designs from various customers are integrated onto one wafer, reducing the necessary cost per design.

In general, these kinds of designs are usually implemented in FPGA, due to the low volumes of manufacture.



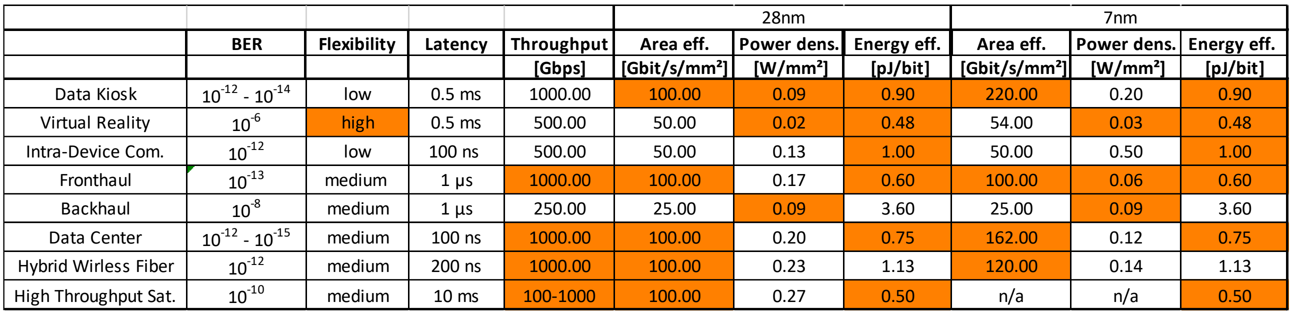
HTS										
Device cost	100000000	€								
PHY chip cost percentage (A)	0.01	%								
FEC cost percentage (B)	32.1	%								
FEC cost	3210	€								
Device power	10000	W								
PHY chip power percentage (C)	0.01	%								
FEC power percentage (D)	50	%								
FEC power	0.5	W								
Volume	1000				with area		with area		with area	
Throughput	1000	Gbps			limitation		limitation		limitation	
			Node	**28**	**28**	**16**	**16**	**7**	**7**	nm
			Wafer size (diam)	12	12	12	12	12	12	inch
			Full mask set price	1.5	1.5	3.2	3.2	6.5	6.5	M€
			Wafer price	5	5	11	11	22	22	k€
FEC area	**10**			**24954.33**	**10.00**	**66.33**	**10.00**	**0.00**	**0.00**	**mm²**
Area efficiency	**100**			**0.04**	**100.00**	**15.08**	**100.00**	**na**	**na**	**Gbit/s/mm²**
Energy efficiency	**1**			**0.5**	**0.5**	**0.5**	**0.5**	**0.50**	**0.50**	**pJ/bit**
Power density	**0.1**			**0.00002**	**0.050**	**0.008**	**0.050**	**na**	**na**	**W/mm²**

Table 18: FEC KPI for the high-throughput satellites use case

In terms of latency, additional delays might come from buffering demodulated frames prior to decoding. For a symbol rate of 27 MBaud, it represents only 1 ms of added delay. This value significantly reduces to 3 µs, for HTS currently operating in Ka-Band with a bandwidth of 3.5 GHz. The latency attributed to FEC processes strongly depends on the architecture and type of the decoder itself. Usually it is not a matter of concern, since it is mostly insignificant compared to the delay introduced by the radio waves' propagation. Therefore, latency values of at most 10 ms are still acceptable, since 20 ms added to the overall delay of the system is still not perceivable in most services.

2.3 Conclusion

The seven use cases detailed above have different FEC performance target (BER, flexibility) and implementation KPI (latency, throughput, area efficiency, energy efficiency, power density) target due to different application scenarios. Table 19 presents a summary of the KPI for the seven use cases. In Table 19, all the KPI more stringent than the EPIC objectives (FEC throughput > 1 Tb/s with, at the same time, an energy efficiency of < 1 pJ/bit, power density of < 0.1 W/mm^2, area efficiency of > 100 Gb/s/mm^2) are marked orange.



					28nm			7nm		
	BER	**Flexibility**	**Latency**	**Throughput**	**Area eff.**	**Power dens.**	**Energy eff.**	**Area eff.**	**Power dens.**	**Energy eff.**
				[Gbps]	**[Gbit/s/mm²]**	**[W/mm²]**	**[pJ/bit]**	**[Gbit/s/mm²]**	**[W/mm²]**	**[pJ/bit]**
Data Kiosk	10^{-12} - 10^{-14}	low	0.5 ms	1000.00	100.00	0.09	0.90	220.00	0.20	0.90
Virtual Reality	10^{-6}	high	0.5 ms	500.00	50.00	0.02	0.48	54.00	0.03	0.48
Intra-Device Com.	10^{-12}	low	100 ns	500.00	50.00	0.13	1.00	50.00	0.50	1.00
Fronthaul	10^{-13}	medium	1 µs	1000.00	100.00	0.17	0.60	100.00	0.06	0.60
Backhaul	10^{-8}	medium	1 µs	250.00	25.00	0.09	3.60	25.00	0.09	3.60
Data Center	10^{-12} - 10^{-15}	medium	100 ns	1000.00	100.00	0.20	0.75	162.00	0.12	0.75
Hybrid Wirless Fiber	10^{-12}	medium	200 ns	1000.00	100.00	0.23	1.13	120.00	0.14	1.13
High Throughput Sat.	10^{-10}	medium	10 ms	100-1000	100.00	0.27	0.50	n/a	n/a	0.50

Table 19: Summary of FEC level KPI for seven different use cases

The BER performance requirement of the virtual reality use case is less demanding than the BER performance of the other use cases.

Two of the use cases (intra-device communication and data centers) need less flexibility in code length and coding rate; virtual reality needs high flexibility in code length and coding rate; the remaining use cases have medium requirement in flexibility.

The most stringent latency requirements are expected to be of the order of 100 to 200 ns for intra-device communications. Most other use cases (fronthaul/backhaul, data centre, hybrid wireless/fibre) have latency requirements of at least 0.5 to 1 µs. Virtual reality and data kiosk have less strict requirement in latency, targeting the ms range. High speed satellites do not really pose latency constraints.

Four of the use cases (data kiosk, fronthaul, data centre and hybrid wireless fiber) have a throughput target of 1 Tb/s, and the remaining use cases have throughput target at 100s Gb/s.

Three of the use cases (virtual reality, backhaul and intra-device communication) have an area efficiency target less than 100 Gb/s/mm^2, while the remaining use cases have an area efficiency target at 100 Gb/s/mm^2 under 28 nm technology node. As the cost of chip fabrication increases significantly, the affordable chip area with the same cost decreases a lot from 28 nm to 7 nm technology, which leads to higher area efficiency target when the technology node moves from 28 nm to 7 nm.

Two of the use cases (virtual reality and backhaul) have power density target smaller than 0.1 W/mm^2 under both 28 nm and 7 nm technology. The remaining use cases all have power density target larger than 0.1 W/mm^2 under both 28 nm and 7 nm technology. Due to the limited volume and radiation hardening issue, the high-speed satellite is not suitable for 7 nm IC fabrication and hence, is not considered here.

Two of the use cases (backhaul and hybrid wireless fiber) have an energy efficiency target larger than 1 pJ/bit whereas the remaining use cases have an energy efficiency target smaller than 1 pJ/bit.

If we look at this table horizontally, some use cases are less stringent. For example, the virtual reality is not so demanding in BER, latency and power density. However, fronthaul, data centre, and hybrid wireless fiber are three use cases having challenges in almost every KPI.

The chosen use cases are sufficiently different from each other to achieve a large diversity, however once our focus change from system level to FEC level, some use cases have similar KPI. The FEC level KPI derived based on system level KPI and the methodology detailed in Section 2.1 are, for many use cases, more demanding then the EPIC project objectives. However, the derived FEC KPI and the EPIC project objectives are in the same order of magnitude.

Chapter 3 State of the Art and Gap Analysis

To gauge the feasibility of the EPIC targets and expose the fundamental challenges in enabling wireless Tbps link technology, we begin by looking at the SoA in FEC implementations. In the meeting of 3GPP 5G standardization committee, August 2016, a large number of independent studies have been submitted that compare Turbo, LDPC, and Polar codes in terms of their communication and implementation performance. These three code families are the leading contenders for the 5G standard and also the code classes of primary interest in EPIC. In order to make a fair and usable comparison between different architectures, all the implementation related KPI of the references architectures are summarized and scaled to the emerging 7 nm CMOS FinFET technology.

This chapter is organized as follows: In section 3.1, we describe the methodology of scaling. From section 3.2 to 3.4, a concise overview of the SoA FEC decoding in current wireless communication is provided for Turbo, LDPC and Polar codes. A gap analysis between the SoA and the use case targets is performed for each of the use cases introduced in the previous chapter. The most important challenges are identified per code family. Finally, section 3.5 contains a summary and concluding remarks.

3.1 Methodology

In order to make a coherent and consistent assessment of the SoA and to evaluate how much improvement is needed on existing implementations in the gap analysis, we adopt the following steps:

- Scale existing SoA designs to 7 nm CMOS technology node (detailed in section 3.1.1);
- Scale to the desired throughput (detailed in section 3.1.2);
- Analyze how far the result is from the desired target.

3.1.1 Scaling to 7 nm CMOS Technology Node

We start from an existing SoA design having

- Throughput (T)
- Clock frequency (F)
- Area (A)
- Power consumption (P)
- Area efficiency (AE)
- Energy efficiency (EE)
- Power density (PD)

Thanks to the CMOS technology scaling [2], we have three scaling factors:

- Clock frequency scaling factor ($S_f > 1$)
- Area scaling factor (S_A < 1)
- Energy efficiency scaling factor (S_{EE} < 1)

The scaling values have been calculated as follows. Based on the ITRS roadmap [2] and NVIDIA study on future “exascale” computing [3], we estimate that moving a given architecture from 28 nm to 7 nm technology will bring, approximately, a factor x12 reduction in area, a factor x4 improvement in energy efficiency and a factor x3 increase in clock speed (maximum operating frequency). Based on these scaling factors, we can compute the scaling factors when going from any technology node (length L in nm) to the 7 nm technology node as follows (the base 4 in the logarithm comes from the ratio 28/7):

- $S_F(L \rightarrow 7) = 3^{\log_4(L/7)}$ (> 1 when scaling down)
- $S_{EE}(L \rightarrow 7) = 4^{-\log_4(L/7)}$ (< 1 when scaling down)
- $S_A(L \rightarrow 7) = 12^{-\log_4(L/7)}$ (< 1 when scaling down)

Once the scaling factors S_F, S_{EE}, S_A from the SoA node to the 7 nm node have been computed, the scaling is applied following the formulas indicated in Table 20. It should be noted that, for simplicity, we neglect the leakage power in this scaling. This will lead to some under-estimation of the power but this is acceptable for this high level analysis where we are interested in the orders of magnitude.

Parameter name	Parameter in SoA technology node	Scaling factor	Scaled parameter
Throughput	T	S_F	$T \cdot S_F$
Clock Frequency	F	S_F	$F \cdot S_F$
Area	A	S_A	$A \cdot S_A^n$
Power	$P = EE \cdot T$	$S_{EE} \cdot S_F$	$P \cdot S_{EE} \cdot S_F$
Area efficiency	$AE = T/A$	S_F/S_A	$AE \cdot S_F/S_A$
Energy efficiency	EE	S_{EE}	$EE \cdot S_{EE}$
Power density	$PD = P/A$	$S_{EE} \cdot S_F/S_A$	$PD \cdot S_{EE} \cdot S_F/S_A$

Table 20: Formulas for the technology node scaling methodology

3.1.2 *Scaling to the Desired Throughput*

The scaling to the desired throughput may lead to unrealistic clock frequency values. Hence, we clip the maximum frequency at 1GHz for the down-scaled technology node. 1GHz is a reasonable frequency for a SoC IP. As already mentioned in the previous paragraphs, the maximum FEC area is set to 10 mm^2 if the area that results from the FEC cost of the various use cases exceeds the 10 mm^2. Based on these considerations, our methodology consists of the following steps:

- We only scale the SoA to 7nm, all values are rounded off to one digit after the comma for clarity.
- FEC area and maximum frequency are clipped to 10 mm^2 and 1GHz respectively for all technology nodes if the corresponding values are exceeding these limits.
- If the area is below 10 mm^2 and the throughput target is not reached, we apply spatial parallelism, i.e. make multiple instances of the FEC cores until the target throughput is achieved or the area limit is reached.
- Finally, area efficiency, energy efficiency, power density, latency are calculated for each technology node.

After this scaling procedure is applied, the gap analysis for the different use cases and code families can be performed. Table 21 shows an example of scaling on a *hypothetical* SoA design, going from 40 nm to 7 nm. In the fifth column of the table, we also show a colouring rule that will be used in the gap analysis:

- Green: the obtained value is better than the targeted value.
- Orange: the obtained value is worse than the targeted value by a factor 5 at most.
- Red: the obtained value is worse than the targeted value by a factor larger than 5.

This colouring serves as a visual indicator of how much effort is needed to reach the targeted value; it will be used in the gap analysis section (section 3.2.2, 3.3.2 and 3.4.2).

	SoA	SoA scaled to 7nm	SoA scaled to 7nm, Clock freq is limited and area is increased to achieve desired throughput	SoA scaled to 7nm, Area is limited to 10mm²	Limit value
Technology (*nm*)	40	7	7	7	-
Frequency (*MHz*)	800.0	3184.0	1000.0	1000.0	clipped
Power (m*W*)	200.0	139.3	35000.0	2487.5	1000
Throughput (*Gb/s*)	1.0	4.0	1000.0	71.1	1000
Area (*mm*^{2})	4.0	0.1759	140.7	10.0	clipped
Area efficiency (G*b/s/mm*^{2})	0.3	22.6	7.1	7.1	100
Energy efficiency (*pJ/bit*)	5.0	0.9	0.9	0.9	1
Power density (*W/mm*^{2})	0.1	0.8	0.2	0.2	0.1

Table 21: Example of technology node scaling (hypothetical example)

3.2 Turbo Codes

This section assesses the gap between the EPIC use case requirements and the SoA in Turbo coding technology. Subsection 3.2.1 presents the SoA of decoding architectures for Turbo codes that are likely to meet or to approach EPIC targets. Then, subsection 3.2.2 provides performance figures for these architectures when scaled to 7 nm CMOS technology and analyses the gap between the scaled SoA architectures and the EPIC requirements for the Tb/s use cases identified in the project, in terms of FEC KPI. Subsection 3.2.3 provides a summary of the gap analysis results for Turbo decoding architectures.

3.2.1 SoA of Turbo Codes

Turbo codes, developed in the early nineties, were the first practical codes to closely approach the channel capacity [57]. They were adopted in the third and fourth generations of wireless mobile communication standards. In the following, we briefly describe the Turbo decoding principle and its components. Thereafter, we present an overview of SoA high-throughput Turbo decoding hardware architectures for Turbo codes.

3.2.1.1 Turbo Decoding Principle

The structure of a conventional Turbo encoder calls for a parallel concatenation of two recursive systematic convolutional encoders [57]. At the receiving side, the corresponding decoder consists of two component decoders, commonly implementing a derived version of the BCJR algorithm [58]. Practical component decoders implement a logarithmic variant of the algorithm called Log-MAP algorithm or its simplified version named Max-Log-MAP algorithm. The latter is particularly popular due to its low complexity [59]. In the following, the BCJR, Log-MAP or Max-Log-MAP algorithms are referred to as MAP decoding algorithms.

The component decoders are connected through an interleaver π and a de-interleaver π^{-1}, as described in Figure 11. In digital implementations of Turbo decoders, the feedback loop is implemented using an iterative process which repeatedly activates decoders 1 and decoder 2. Each processing of one constituent decoder is counted as one half-iteration (HI) and a complete run of the closed loop is counted as a full iteration.

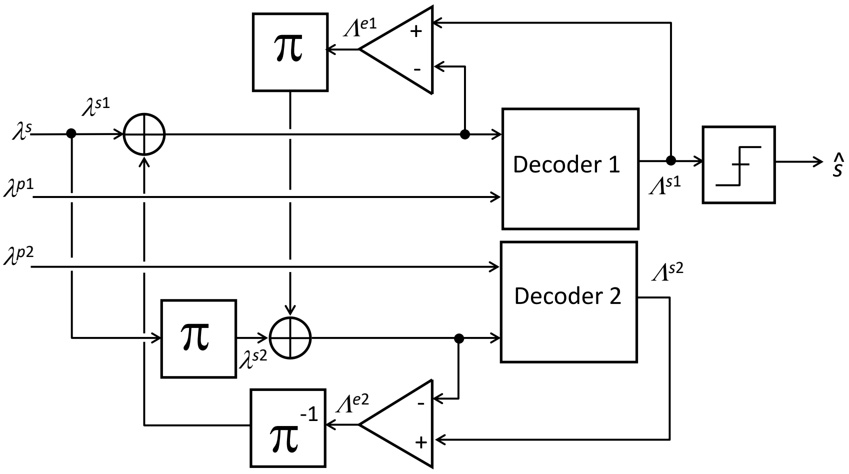


Figure 11: Turbo decoding principle

At the component decoder level, the estimation process is based on forward and backward recursions in the code trellis to compute state metrics. Figure 12(a) illustrates the metrics calculation in the trellis. In most implementations, the computations are scheduled as shown in Figure 12(b). First, the forward state metrics α_k are calculated recursively and stored for each trellis stage k, $k = 1 \cdots K$, during the α recursion. Then, during the β recursion, the backward state metrics β_k, the MAP estimates Λ_k, as well as the extrinsic values Λ_k^e, $k = K \cdots 1$, are calculated using the stored forward state metrics.

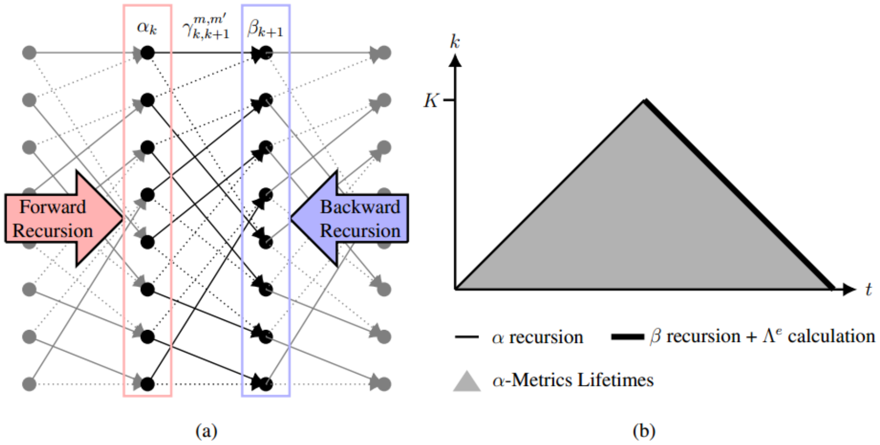


Figure 12: State metric calculation and storage in the BCJR, Log-MAP or Max-Log-MAP algorithm

3.2.1.2 The Turbo Code Interleaver

The type of interleaver has a big impact both on the code error rate performance and on the decoder architecture. Interleaver tables, for example have to be stored in memory, while on-the-fly address calculation requires a dedicated address generator. Furthermore, the type of interleaver may limit the degree of parallelization at the decoder level because of memory access conflicts. These conflicts occur when, due to interleaving, the same memory needs to be accessed at more than one address in the same clock cycle. An interleaver is said to be conflict free if no such collisions occur. The fourth generation of mobile communication standards LTE [60], LTE-A [61] and LTE-A Pro [62] all use Quadratic Permutation Polynomial (QPP) interleavers [63], which have been proven to be conflict free, if the degree of parallelization divides the information block size K.

3.2.1.3 Practical Considerations for the Implementation of the MAP Component Decoders

Figure 12(b) shows that the amount of storage required at the component decoder level for the state metrics as well as the decoding latency increase linearly with the block size K. To decrease both storage requirements and decoding latency, a *sliding window* scheme (also called *windowing*) is used. It loosens the data dependency of the recursive state metric calculation by starting it at arbitrary positions in the block with approximated initialization values. Thereby the trellis of size K is split into sub-trellises or windows of size L_{WS}. This significantly reduces the decoding latency from $2K$ recursion steps down to $K + L_{WS}$ recursion steps, as shown in Figure 13. Note that the state metrics at the window borders need to be carefully initialized to avoid any error rate performance degradation of the decoding process. State metric initialization techniques are not discussed in this document.

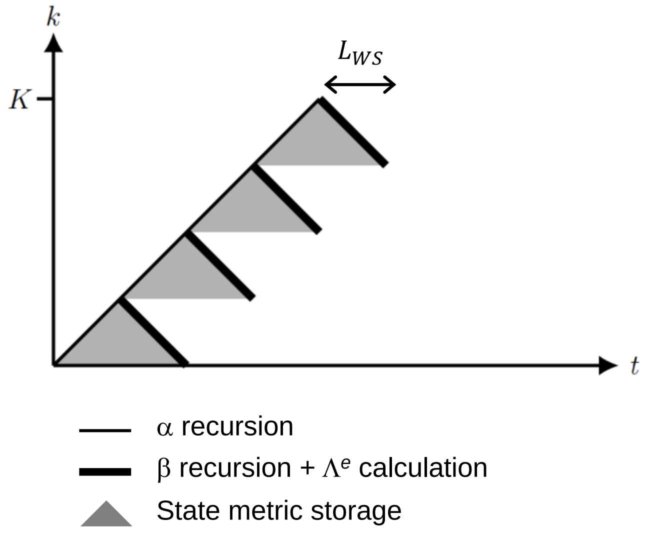


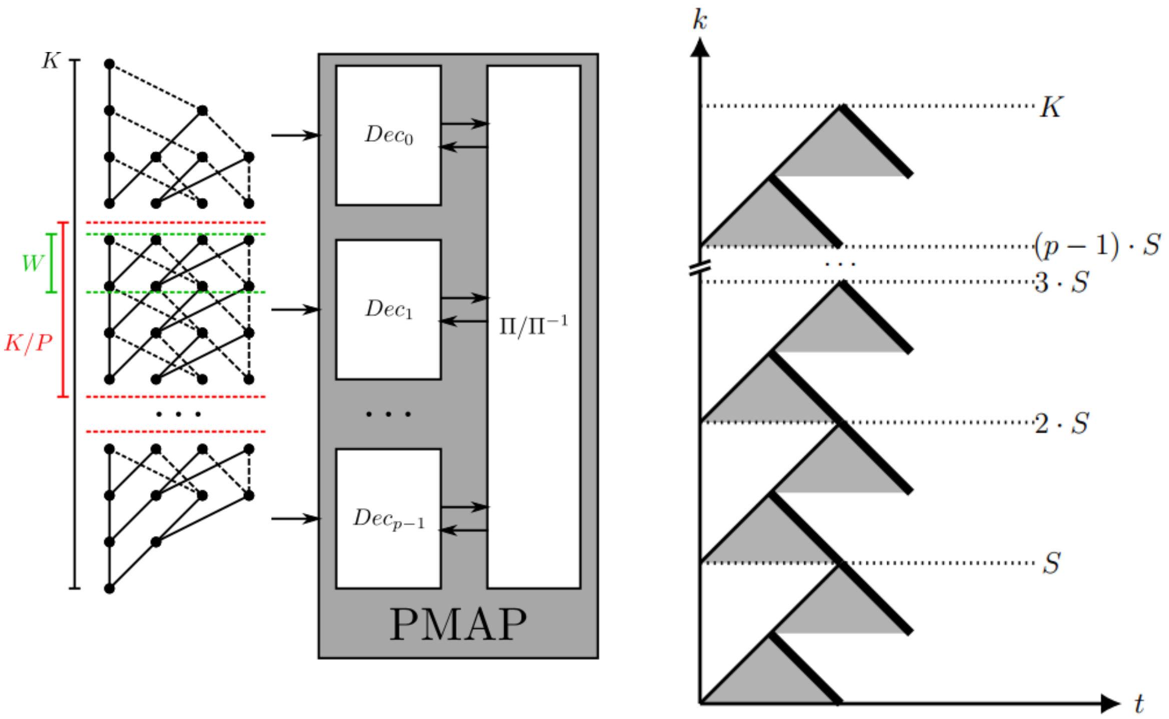
Figure 13: Sliding window principle

3.2.1.4 High-throughput Turbo Decoder Architectures

In the following, we review the different families of hardware architectures suited for high-throughput applications. For a given silicon technology, the throughput of a Turbo decoder is bounded by the critical path, which lies in the recursion units of the MAP decoding cores. Therefore, in order to increase the decoding throughput, the computations of the MAP algorithm need to be performed in parallel. Parallelizing the decoder operations at different levels allows this upper bound to be raised up to the order of Gb/s for current silicon technologies. At the system level, several identical decoders can be employed, which leads to a linear increase in throughput and area. At the MAP decoder level, parallelization is performed on complete code blocks. The code block trellis can, on the one hand, be split into sub-trellises called sub-blocks and processed on parallel on multiple sub-decoder cores. This spatially parallel processing leads to the parallel MAP (PMAP) architecture, explained in section 3.2.1.4.1. At the same level, functional parallelization can be applied by unrolling the calculations of the forward and backward recursions of the MAP algorithm within the sub-blocks, and calculating the recursions in a pipelined fashion, which leads to the so-called cross-MAP (XMAP) architecture described in section 3.2.1.4.2. Recently, a reformulation of the equations of the MAP algorithm was proposed to achieve a fully parallel MAP (FPMAP) architecture, detailed in section 3.2.1.4.3. Additional architectural enhancements can be implemented in specific applications. In the context of the fourth generation of mobile communication systems, such enhancements are also reported in section 3.2.1.4.4.

3.2.1.4.1 The PMAP Architecture

Splitting the code block trellis of size K into sub-trellises allows the decoding process to be spatially parallelized by distributing the sub-trellises to multiple sub-decoder cores [64]. This is illustrated by Figure 14(a), where the p sub-decoder cores are used, which results in sub-trellises or sub-blocks of size $S = K/p$. Additionally, the individual sub-decoder cores may break down the sub-blocks further into windows and apply a sliding window scheme, as illustrated in Figure 14(b).



(a) (b)

Figure 14: PMAP architecture

The throughput TP_{PMAP} for a parallel MAP or PMAP based Turbo decoder with p radix-l sub-decoder cores can be calculated from the following expression:

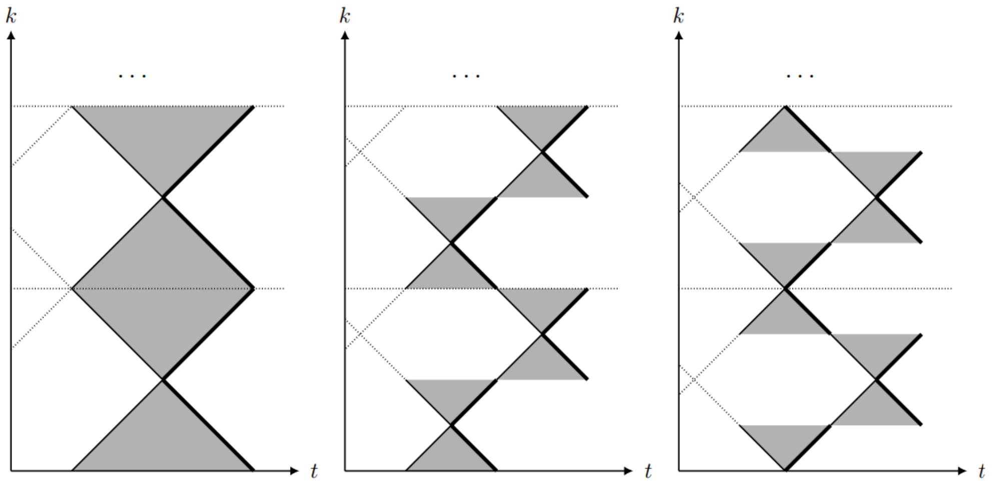
$$TP_{PMAP} = \frac{K}{n_{HI}\left(\frac{K}{p\,\log_2(l)} + L_{WS} + L_{ACQ} + L_P\right) + L_{I/O}} . f \qquad (3.1)$$

The denominator represents the number of clock cycles needed to decode a code block of size K over n_{HI} half-iterations when considering an I/O latency of $L_{I/O}$ clock cycles with a clock frequency$f$. The number of clock cycles to decode one half-iteration is determined by the number of clock cycles needed to process the sub-blocks of size K/p. The individual sub-decoder cores process $\log_2(l)$ trellis steps per clock cycle with a latency of L_{PMAP} which can be written as the sum of the latency for one window L_{WS}, the acquisition latency L_{ACQ} – which depends on the technique adopted to initialize the state metrics at the window and sub-block borders – and the decoder pipeline latency L_P.

Turbo decoders implementing PMAP architectures are the most commonly reported in the literature. Six implementations references report a decoding throughput higher than 1 Gb/s [65] [66] [67] [68] [69] [70].

3.2.1.4.2 The XMAP Architecture

The cross-MAP or XMAP decoder architecture [71] is based on a parallel window decoding scheme. In contrast to sliding window schemes (see Figure 13), where there is a continuous forward or backward recursion, a parallel window scheme calculates multiple windows in parallel. The parameters of this architecture are: the number of recursion patterns working in parallel ρ, the window length L_{WS}, the recursion pattern length $L_{PL} = L_{WS}/\rho$ and the acquisition length L_{ACQ}.



(a) $\rho = 1$ (b) $\rho = 2$ (c) $\rho = 2$, $\Delta_k = L_{WP}/4$

Figure 15: Different XMAP schemes

Figure 15 shows different parallel windowing schemes. In Figure 15(a) there is one recursion pattern ($\rho = 1$), while in Figure 15(b) and (c) there are two recursion patterns ($\rho = 2$). Moreover, in Figure 15(c), the recursion patterns are shifted with a k-axis shift $\Delta_k = L_{WS}/4$ [72]. This configuration is the windowing scheme used in the original XMAP implementation of Worm *et. al* [73], which was proven to be optimal with respect to minimizing the memory needed for channel values, metrics and output values for $L_{WS} \leq 32$ [74].

While PMAP decoders process the sub-blocks serially on parallel sub-decoder cores, the X-windowing scheme in Figure 15(b) and (c) can be mapped to a pipeline. This pipeline consists of a chain of recursion units which are connected through pipeline registers [75], [72]. In this way, the recursion calculation of the windows is functionally unrolled, and each clock cycle one complete window is fed into the pipeline, which processes different trellis steps within the windows in parallel.

Considering a clock frequency f, a code block of size K, a number n_{HI} of half-iterations and an I/O latency of $L_{I/O}$ clock cycles, the throughput TP_{XMAP} for a radix-l XMAP decoder with a window length L_{WS} and an acquision length L_{ACQ} is given by:

$$TP_{XMAP} = \frac{K}{n_{HI}\left(\frac{K}{p\,\log_2(l)} + L_{WS} + L_{ACQ} + \left\lceil\frac{L_{ACQ}}{L_{WS}}\right\rceil\right) + L_{I/O}} \cdot f \qquad (3.2)$$

Two XMAP-based Turbo decoder implementations with decoding throughputs higher that 1 Gb/s can be found in the literature [76] [77].

Two specific techniques have recently been investigated by University of Kaiserslautern to increase the throughput of XMAP-based Turbo decoders.

The first studied technique calls for the use of carry-save arithmetic to perform bit-level pipelined Add-Compare-Select (ACS) operations in the forward and backward recursions of the decoding algorithm. The carry-save representation avoids the carry propagation in the adders. This technique was successfully implemented in a fully LTE-A Pro compatible Turbo decoder architecture synthesized on 65 nm technology, allowing a 14% increase of throughput at the cost of a 40% area increase [78].

For high coding rates, a trellis compression technique can be used to shorten long trellis sections without parity [79]. This technique was implemented in a LTE-A Pro Turbo decoder using the XMAP architecture to decrease the acquisition length for the state metrics initialization for the highest code rate value (0.94). Throughput gains of 5-10% and area savings of more than 14% were demonstrated [80].

3.2.1.4.3 The FPMAP Architecture

Combining shuffled decoding [81] with a splitting of the trellis into sub-trellises of size 1 leads to the fully parallel MAP or FPMAP architecture, which can be seen as a borderline case of the PMAP architecture with $p = K$. The approach was first presented in [82] and an implementation based on the LTE Turbo-Code was recently published in [83].

A schematic of the FPMAP architecture is shown in Figure 16. The processing of the FPMAP algorithm is done using $2K$ processing elements (PE). Each PE computes the branch metrics, the forward and backward state metrics as well as the extrinsic information for one trellis step. The computed state metrics are exchanged with the neighbouring PEs, which is analogue to a state metric handover between the sub-decoder cores in a PMAP decoder. Using the received state metrics, the incoming channel values and the incoming extrinsic information, an outgoing extrinsic information is computed.

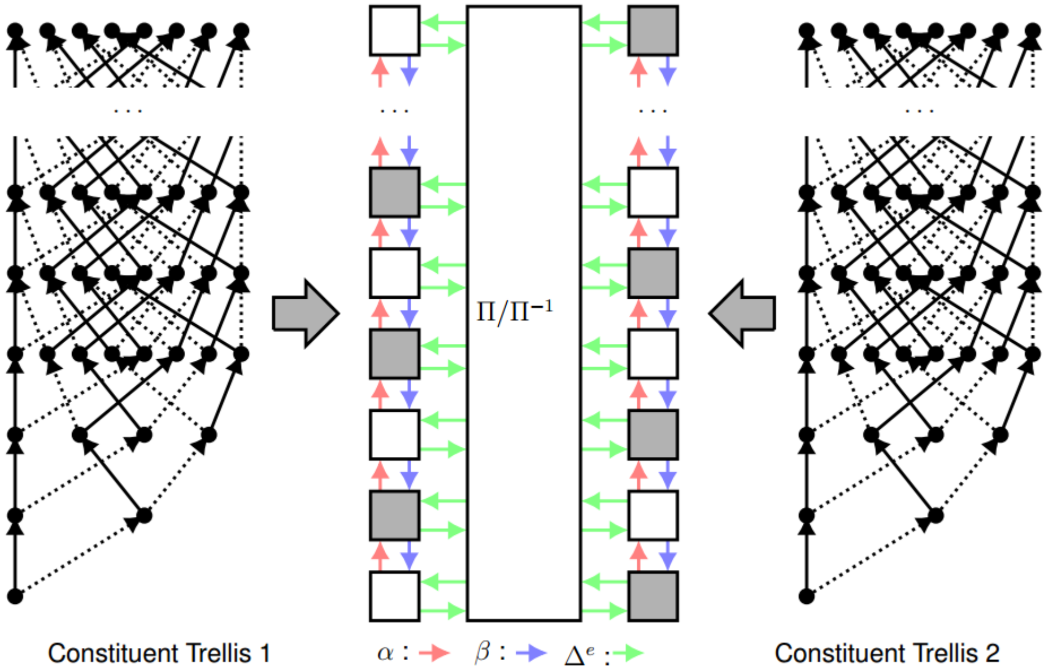


Figure 16: FPMAP architecture

For the LTE Turbo-Code, which uses an odd-even QPP interleaver [63], the PEs can be split into two groups, without any connection between PEs across the groups. The first group (shown in white in Figure 16) contains all PEs that process even trellis steps of constituent code 1 and all PEs that process odd trellis steps of constituent code 2, while the second group (shown in grey in Figure 16) contains the odd PEs and even PEs for constituent code 1 and 2 respectively [82]. Thus, for LTE, one complete decoding iteration can be mapped onto K PEs, which process the first and the second set in turn. In comparison to using $2K$ PEs, this halves the implementation complexity but also halves the decoder throughput.

Compared to SoA PMAP and XMAP implementations, the FPMAP architecture offers a greater throughput due to its fully parallel structure:

$$TP_{FPMAP} = \frac{Kf}{n_{it}} \tag{3.3}$$

where K, f and n_{it} denote the information block size, the clock frequency and the number of iterations.

The major weakness of this architecture is its lack of flexibility with respect to code block sizes. Although in [83], bypass units are proposed to enable different interleaving patterns for different code block sizes, this approach is not suitable when the number of different code block sizes is very large. For example, for LTE, there are 188 different code block sizes and no pattern exists to configure the FPMAP for code block sizes $K \in [784, 6080]$. Furthermore, the combination of one-stage trellises and shuffled decoding degrades the error rate performance of the decoder. While for low code rates more decoding iterations (i.e. $n_{it} \approx 40$) are sufficient to mitigate the resulting performance loss, for higher code rates the performance is degraded.

3.2.1.4.4 Advanced Iteration Control Techniques

High-throughput Turbo decoders aiming at low decoding latencies employ iteration control techniques in order to terminate the decoding iterations as early as possible. In the LTE context, early iteration stopping can be performed using the cyclic redundancy check (CRC) codes implemented at the code block (CB) and transport block (TB) levels.

On the one hand, on-the-fly calculation of the CRC in parallel with the decoding of each half Turbo iteration yields some energy gain [77]. On the other hand, considering iteration control at the TB level instead of the CB level can additionally bring a throughput gain of up to 50% [77].

3.2.1.5 Comparison of SoA Turbo Decoder Implementations

Table 22 provides an overview of state-of-the-art Turbo decoder implementations reported in the literature with decoded information throughput higher than or equal to 1 Gb/s.

Reference	[65] [66]	[67]	[68]	[68]	[69]	[70]	[76]	[77]	[83]
Code - Flexibility	LTE-A	LTE-A	LTE-A	LTE-A	LTE-A	*K*= 4096	LTE	LTE-A	*K*= 6144
Max block size	6144	6144	6144	6144	6144	4096	6144	6144	6144
Architecture	PMAP	PMAP	PMAP	PMAP	PMAP	PMAP	XMAP	XMAP	FPMAP
Technology (*nm*)	65	65	90	90	65	90	45	28	65
Supply voltage (*V*)	1.2	1.1	1.0	1.0	0.9	0.9	0.8	1.0	1.1
Radix/p	4/16	4/32	2/64	2/64	2/64	16/32	4/16	4/16	2/6144
Window size	14-30	192	96	96	64	32		192	--
Nb max it.	5.5	6.0	8.0	5.5	6.0	8.0	5.5	6.0	39.0
Throughput (Gb/s)[1]	**1.0**	**2.2**	**2.3**	**3.3**	**1.3**	**1.4**	**1.7**	**1.4**	**15.8**
Frequency (MHz)	410	450	625	625	400	175	600	825	100
Area (*mm*^{2})	2.5	7.7	19.8	19.8	8.3	9.6	2.0	0.6	109.0
Power (*W*)	1.0	n/a	1.5	1.0	0.8	1.4	0.9	0.7	9.6
Area efficiency (*Gb/s/mm*^{2})	0.4	0.3	0.1	0.2	0.2	0.1	0.8	2.5	0.1
Power density (*W/mm*^{2})	0.4	n/a	0.1	0.1	0.1	0.1	0.4	1.2	0.1
Energy efficiency (*pJ/bit*)	953.6	n/a	637.9	301.5	660.2	968.6	521.0	488.7	608.7
Latency (*µs*)	6.1	2.9	2.7	1.9	4.8	2.9	3.7	4.5	0.4

Table 22: Comparison of SoA Turbo decoder implementations.

[1] Throughput measured at the maximum number of iterations

In order to make the comparison of these reference Turbo decoder implementations over different technology nodes easier, the figures are scaled to 7 nm in Table 23.

Reference	[65] [66]	[67]	[68]	[68]	[69]	[70]	[76]	[77]	[83]
Code - Flexibility	LTE-A	LTE-A	LTE-A	LTE-A	LTE-A	*K*= 4096	LTE	LTE-A	*K*= 6144
Max block size	6144	6144	6144	6144	6144	4096	6144	6144	6144
Architecture	PMAP	PMAP	PMAP	PMAP	PMAP	PMAP	XMAP	XMAP	FPMAP
Technology (*nm*)	7	7	7	7	7	7	7	7	7
Radix/p	4/16	4/32	2/64	2/64	2/64	16/32	4/16	4/16	2/6144
Window size	14-30	192	96	96	64	32		192	--
Nb max it.	5.5	6.0	8.0	5.5	6.0	8.0	5.5	6.0	39.0
Throughput (Gb/s)	**4.2**	**8.9**	**10.6**	**15.5**	**5.3**	**6.5**	**6.0**	**4.1**	**65.4**
Frequency (MHz)	1696.1	1861.6	2921.5	2921.5	1654.7	818.0	2148.2	2475.0	413.7
Area (*mm*^{2})	0.10	0.31	0.60	0.60	0.33	0.29	0.11	0.05	4.39
Power (*W*)	0.7	n/a	1.0	0.7	0.6	0.9	0.6	0.5	6.6
Area efficiency (*Gb/s/mm*^{2})	41.8	28.7	17.6	25.6	15.8	22.3	53.4	90.0	14.9
Power density (*W/mm*^{2})	6.6	n/a	1.6	1.1	1.7	3.1	5.6	11.0	1.5
Energy efficiency (*pJ/bit*)	158.9	n/a	91.1	43.1	110.0	138.4	104.2	122.2	101.5
Latency (*µs*)	1.5	0.7	0.6	0.4	1.2	0.6	1.0	1.5	0.1

Table 23: Comparison of SoA Turbo decoder implementations scaled to 7 nm.

Table 23 shows that directly scaling the figures related to the Turbo decoder described in all the referenced implementations down to 7 nm, except for the full parallel MAP architecture of [83] and the parallel MAP architecture of [70], leads to a clock frequency above 1 GHz which is considered practically unfeasible. Therefore, frequency clipping is necessary as mentioned in section 3.1.2.

3.2.2 *Gap Analysis*

This section aims at analyzing the gap existing between the KPI of the SoA Turbo decoder implementations presented in the previous section and the FEC-level KPI requirements related to the use cases presented in section 2, when technology is scaled down to 7 nm.

The error rate performance of the corresponding decoders has not been taken into account in the analysis. However, since all the decoders fully or partially implement the LTE Turbo code, for the same code block and coding rate, the error performance of the compared decoders are quite close to each other, with the number of iterations mentioned in the tables.

3.2.2.1 Scaling to 7 nm with Frequency Clipping to 1 GHz

In order to take the 1 GHz maximum frequency limitation into account, the figures of Table 23 have been updated in Table 24 .

Note that in Table 24 and in the following tables of this gap analysis section, all the throughput values have been measured at the maximum number of decoding iterations, meaning that no technique for an early stopping of the iterations was implemented.

Reference	[65] [66]	[67]	[68]	[68]	[69]	[70]	[76]	[77]	[83]
Code - Flexibility	LTE-A	LTE-A	LTE-A	LTE-A	LTE-A	*K*= 4096	LTE	LTE-A	*K*= 6144
Max block size	6144	6144	6144	6144	6144	4096	6144	6144	6144
Architecture	PMAP	PMAP	PMAP	PMAP	PMAP	PMAP	XMAP	XMAP	FPMAP
Technology (*nm*)	7	7	7	7	7	7	7	7	7
Radix/p	4/16	4/32	2/64	2/64	2/64	16/32	4/16	4/16	2/6144
Window size	14-30	192	96	96	64	32		192	--
Nb max it.	5.5	6.0	8.0	5.5	6.0	8.0	5.5	6.0	39.0
Throughput (Gb/s)	**2.5**	**4.8**	**3.6**	**5.3**	**3.2**	**6.5**	**2.8**	**1.7**	**65.4**
Frequency (MHz)	1000.0	1000.0	1000.0	1000.0	1000.0	818.0	1000.0	1000.0	413.7
Area (*mm*^{2})	0.10	0.31	0.60	0.60	0.33	0.29	0.11	0.05	4.39
Power (*W*)	0.4	n/a	0.3	0.2	0.4	0.9	0.3	0.2	6.6
Area efficiency (*Gb/s/mm*^{2})	24.6	15.4	6.0	8.8	9.6	22.3	24.9	36.4	14.9
Power density (*W/mm*^{2})	3.9	n/a	0.5	0.4	1.1	3.1	2.6	4.4	1.5
Energy efficiency (*pJ/bit*)	158.9	n/a	91.1	43.1	110.0	138.4	104.2	122.2	101.5
Latency (*µs*)	2.5	1.3	1.7	1.2	1.9	0.6	2.2	3.7	0.1

Table 24: Comparison of SoA Turbo decoder implementations scaled to 7 nm, with clock frequency clipped to 1 GHz.

3.2.2.2 Data Kiosk

	Use case	SoA Turbo Decoder								
	Data kiosk	[65] [66]	[67]	[68]	[68]	[69]	[70]	[76]	[77]	[83]
Code Flexibility		LTE-A	LTE-A	LTE-A	LTE-A	LTE-A	*K*= 4096	LTE	LTE-A	*K*= 6144
Architecture		PMAP	PMAP	PMAP	PMAP	PMAP	PMAP	XMAP	XMAP	FPMAP
Nb decoders		45	14	7	7	13	15	40	99	1
Nb max it.		5.5	6.0	8.0	5.5	6.0	8.0	5.5	6.0	39.0
Throughput (*Gb/s*)	1000	111.2	66.9	25.5	37.0	41.6	98.2	111.3	165.0	65.4
Area (*mm*^{2})	4.5	4.5	4.3	4.2	4.2	4.3	4.4	4.5	4.5	4.4
Power (*W*)	0.9	17.7	n/a	2.3	1.6	4.6	13.6	11.6	20.2	6.6
Area eff. (*Gb/s/mm*^{2})	220.1	24.6	15.4	6.0	8.8	9.6	22.3	24.9	36.4	14.9
Power dens. (*W/mm*^{2})	0.20	3.91	n/a	0.55	0.38	1.05	3.08	2.59	4.44	1.51
Energy eff. (*pJ/bit*)	0.9	158.9	n/a	91.1	43.1	110.0	138.4	104.2	122.2	101.5
Latency (*µs*)	500	2.5	1.3	1.7	1.2	1.9	0.6	2.2	3.7	0.1
Frequency (*MHz*)	1000	1000.0	1000.0	1000.0	1000.0	1000.0	818.0	1000.0	1000.0	413.7

Table 25: Gap analysis of Turbo code implementations for data kiosk use case at 7 nm

- At 7 nm, the scaled throughput, area efficiency and energy efficiency values are far from the requirements.

- For the more area efficient architectures, the throughput has to be multiplied by a factor between 6 and 10 to achieve 1 Tb/s.
- Only the PMAP architecture presented in [68] has a power density close to the requirement, but the throughput achieved by this architecture is 30 times lower than the required throughput.
- For the data kiosk use case, the latency requirement is not very stringent and all the architectures are compliant with the target value.

3.2.2.3 Mobile Virtual Reality

	Use case	**SoA Turbo Decoder**								
	Virtual reality	[65] [66]	[67]	[68]	[68]	[69]	[70]	[76]	[77]	[83]
Code Flexibility		LTE-A	LTE-A	LTE-A	LTE-A	LTE-A	*K*= 4096	LTE	LTE-A	*K*= 6144
Architecture		PMAP	PMAP	PMAP	PMAP	PMAP	PMAP	XMAP	XMAP	FPMAP
Nb decoders		90	29	15	15	27	31	81	198	2
Nb max it.		5.5	6.0	8.0	5.5	6.0	8.0	5.5	6.0	39.0
Throughput (*Gb/s*)	500	222.4	138.6	54.6	79.4	86.4	202.9	225.5	330.0	130.7
Area (mm^2)	9.1	9.0	9.0	9.1	9.1	9.0	9.1	9.1	9.1	8.8
Power (*W*)	0.2	35.3	n/a	5.0	3.4	9.5	28.1	23.5	40.3	13.3
Area eff. ($Gb/s/mm^2$)	54.8	24.6	15.4	6.0	8.8	9.6	22.3	24.9	36.4	14.9
Power dens. (W/mm^2)	0.03	3.91	n/a	0.55	0.38	1.05	3.08	2.59	4.44	1.51
Energy eff. (*pJ/bit*)	0.5	158.9	n/a	91.1	43.1	110.0	138.4	104.2	122.2	101.5
Latency (*µs*)	500	2.5	1.3	1.7	1.2	1.9	0.6	2.2	3.7	0.1
Frequency (*MHz*)	1000	1000.0	1000.0	1000.0	1000.0	1000.0	818.0	1000.0	1000.0	413.7

Table 26: Gap analysis of Turbo code implementations for virtual reality use case at 7 nm

- At 7 nm, although none of the architectures are able to achieve the required throughput, some implementations of the three architecture types presented in section 3.2.1.4 are able to approach the throughput requirement of the virtual reality use case. However, for these architectures, there are still important gaps to be filled in terms of power density and energy efficiency.
- For the virtual reality use case, the latency requirement is not very stringent and all the architectures are compliant with the target value.

3.2.2.4 Wireless Intra-Device Communication

	Use case	**SoA Turbo Decoder**								
	Intra device	[65] [66]	[67]	[68]	[68]	[69]	[70]	[76]	[77]	[83]
Code Flexibility		LTE-A	LTE-A	LTE-A	LTE-A	LTE-A	*K*= 4096	LTE	LTE-A	*K*= 6144
Architecture		PMAP	PMAP	PMAP	PMAP	PMAP	PMAP	XMAP	XMAP	FPMAP
Nb decoders		95	31	15	15	28	32	85	209	2

	Use case	SoA Turbo Decoder								
	Intra device	[65] [66]	[67]	[68]	[68]	[69]	[70]	[76]	[77]	[83]
Nb max it.		5.5	6.0	8.0	5.5	6.0	8.0	5.5	6.0	39.0
Throughput (*Gb/s*)	500	234.7	148.1	54.6	79.4	89.6	209.4	236.6	348.3	130.7
Area (*mm*^{2})	9.6	9.5	9.6	9.1	9.1	9.4	9.4	9.5	9.6	8.8
Power (*W*)	0.5	37.3	n/a	5.0	3.4	9.9	29.0	24.7	42.6	13.3
Area eff. (*Gb/s/mm*^{2})	52.0	24.6	15.4	6.0	8.8	9.6	22.3	24.9	36.4	14.9
Power dens. (*W/mm*^{2})	0.05	3.91	n/a	0.55	0.38	1.05	3.08	2.59	4.44	1.51
Energy eff. (*pJ/bit*)	1	158.9	n/a	91.1	43.1	110.0	138.4	104.2	122.2	101.5
Latency (*µs*)	0.2	2.5	1.3	1.7	1.2	1.9	0.6	2.2	3.7	0.1
Frequency (*MHz*)	1000	1000.0	1000.0	1000.0	1000.0	1000.0	818.0	1000.0	1000.0	413.7

Table 27: Gap analysis of Turbo code implementations for intra-device communication use case at 7 nm

In terms of FEC-level KPI, the intra-device communication use case is very close to the virtual reality use case, except for the latency requirements.

- At 7 nm, although none of the architectures are able to achieve the required throughput, some implementations of the three architecture types presented in section 3.2.1.4 are able to approach the throughput requirement of the intra-device communication use case. The decoder offering the highest throughput is the XMAP architecture described in [77].
- However, for these architectures, there are still important gaps to be filled in terms of power density and energy efficiency.
- Contrary to the previous use case, the latency requirement is much too stringent to be met for most of the existing Turbo decoder architectures, except for the PMAP architecture presented in [70] and the fully parallel MAP architecture presented in [83].

3.2.2.5 Wireless Fronthaul/Backhaul

	Use case	SoA Turbo Decoder								
	Backhaul	[65] [66]	[67]	[68]	[68]	[69]	[70]	[76]	[77]	[83]
Code Flexibility		LTE-A	LTE-A	LTE-A	LTE-A	LTE-A	*K*= 4096	LTE	LTE-A	*K*= 6144
Architecture		PMAP	PMAP	PMAP	PMAP	PMAP	PMAP	XMAP	XMAP	FPMAP
Nb decoders		99	32	16	16	29	34	89	151	2
Nb max it.		5.5	6.0	8.0	5.5	6.0	8.0	5.5	6.0	39.0
Throughput (*Gb/s*)	250	244.6	152.9	58.2	84.7	92.8	222.5	247.7	251.7	130.7
Area (*mm*^{2})	10.0	9.9	9.9	9.7	9.7	9.7	10.0	10.0	6.9	8.8
Power (*W*)	0.9	38.9	n/a	5.3	3.6	10.2	30.8	25.8	30.7	13.3
Area eff. (*Gb/s/mm*^{2})	25	24.6	15.4	6.0	8.8	9.6	22.3	24.9	36.4	14.9
Power dens. (*W/mm*^{2})	0.09	3.91	n/a	0.55	0.38	1.05	3.08	2.59	4.44	1.51
Energy eff. (*pJ/bit*)	3.6	158.9	n/a	91.1	43.1	110.0	138.4	104.2	122.2	101.5
Latency (*µs*)	1	2.5	1.3	1.7	1.2	1.9	0.6	2.2	3.7	0.1

	Use case	SoA Turbo Decoder								
	Backhaul	[65] [66]	[67]	[68]	[68]	[69]	[70]	[76]	[77]	[83]
Frequency (*MHz*)	1000	1000.0	1000.0	1000.0	1000.0	1000.0	818.0	1000.0	1000.0	413.7

Table 28: Gap analysis of Turbo code implementations for backhaul use case at 7 nm

- At 7 nm, four implementations are able to achieve or approach the throughput requirement of the backhaul use case ([65] [66] [70] [76] [77]). For these architectures, the power density and energy efficiency have still to be improved: an improvement factor in the order of 30-50 has to be achieved for both indicators.
- The latency specs of state-of-the-art solutions are mostly in the range of requirements for backhaul links. The fully parallel MAP meets it with a large margin.

	Use case	SoA Turbo Decoder								
	Fronthaul	[65] [66]	[67]	[68]	[68]	[69]	[70]	[76]	[77]	[83]
Code Flexibility		LTE-A	LTE-A	LTE-A	LTE-A	LTE-A	*K*= 4096	LTE	LTE-A	*K*= 6144
Architecture		PMAP	PMAP	PMAP	PMAP	PMAP	PMAP	XMAP	XMAP	FPMAP
Nb decoders		99	32	16	16	29	34	89	218	2
Nb max it.		5.5	6.0	8.0	5.5	6.0	8.0	5.5	6.0	39.0
Throughput (*Gb/s*)	1000	244.6	152.9	58.2	84.7	92.8	222.5	247.7	363.3	130.7
Area (*mm*^{2})	10.0	9.9	9.9	9.7	9.7	9.7	10.0	10.0	10.0	8.8
Power (*W*)	0.6	38.9	n/a	5.3	3.6	10.2	30.8	25.8	44.4	13.3
Area eff. (*Gb/s/mm*^{2})	100	24.6	15.4	6.0	8.8	9.6	22.3	24.9	36.4	14.9
Power dens. (*W/mm*^{2})	0.06	3.91	n/a	0.55	0.38	1.05	3.08	2.59	4.44	1.51
Energy eff. (*pJ/bit*)	0.6	158.9	n/a	91.1	43.1	110.0	138.4	104.2	122.2	101.5
Latency (μs)	1	2.5	1.3	1.7	1.2	1.9	0.6	2.2	3.7	0.1
Frequency (*MHz*)	1000	1000.0	1000.0	1000.0	1000.0	1000.0	818.0	1000.0	1000.0	413.7

Table 29: Gap analysis of Turbo code implementations for fronthaul use case at 7 nm

In terms of FEC-level KPI, the fronthaul use case is much more demanding than the backhaul use case.

- At 7 nm, none of the architectures are able to achieve the required throughput.
- However, four implementations approach or exceed the quarter of the target throughput value, the decoder offering the highest throughput being the XMAP architecture presented in [77]. However, the energy efficiency and power density of this decoder do not meet the corresponding requirements: respective factors of 200 and 75 have still to be gained.
- The latency specs of state-of-the-art solutions are mostly in the range of requirements for fronthaul links. The fully parallel MAP meets it with a large margin.

3.2.2.6 Data Centre

Use case	SoA Turbo Decoder								
Data centre	[65] [66]	[67]	[68]	[68]	[69]	[70]	[76]	[77]	[83]

	Use case	SoA Turbo Decoder								
	Data centre	[65] [66]	[67]	[68]	[68]	[69]	[70]	[76]	[77]	[83]
Code Flexibility		LTE-A	LTE-A	LTE-A	LTE-A	LTE-A	*K*= 4096	LTE	LTE-A	*K*= 6144
Architecture		PMAP	PMAP	PMAP	PMAP	PMAP	PMAP	XMAP	XMAP	FPMAP
Nb decoders		61	19	10	10	18	20	54	133	1
Nb max it.		5.5	6.0	8.0	5.5	6.0	8.0	5.5	6.0	39.0
Throughput (*Gb/s*)	1000	150.7	90.8	36.4	52.9	57.6	130.9	150.3	221.7	65.4
Area (*mm*^{2})	6.1	6.1	5.9	6.0	6.0	6.0	5.9	6.0	6.1	4.4
Power (*W*)	0.8	24.0	n/a	3.3	2.3	6.3	18.1	15.7	27.1	6.6
Area eff. (*Gb/s/mm*^{2})	163.0	24.6	15.4	6.0	8.8	9.6	22.3	24.9	36.4	14.9
Power dens. (*W/mm*^{2})	0.1	3.91	n/a	0.55	0.38	1.05	3.08	2.59	4.44	1.51
Energy eff. (*pJ/bit*)	0.75	158.9	n/a	91.1	43.1	110.0	138.4	104.2	122.2	101.5
Latency (*µs*)	0.1	2.5	1.3	1.7	1.2	1.9	0.6	2.2	3.7	0.1
Frequency (*MHz*)	1000	1000.0	1000.0	1000.0	1000.0	1000.0	818.0	1000.0	1000.0	413.7

Table 30: Gap analysis of Turbo code implementations for data centre use case at 7 nm

In terms of FEC-level KPI, the data centre use case is quite close to the fronthaul use case and mainly differs in terms of area limitation and latency requirements.

- At 7 nm, none of the architectures are able to achieve the required throughput.
- Only the XMAP architecture presented in [77] exceeds a throughput value of 200 Gb/s (the target is 1 Tb/s). However, the energy efficiency and power density of this decoder does not meet the requirements: respective factors of 160 and 45 have still to be gained.
- As for the latency, this implementation cannot meet the requirement. Only the fully parallel MAP decoder is able to meet it in 7 nm.

3.2.2.7 Hybrid Fiber-Wireless Networks

	Use case	SoA Turbo Decoder								
	Hybrid fiber-wireless	[65] [66]	[67]	[68]	[68]	[69]	[70]	[76]	[77]	[83]
Code Flexibility		LTE-A	LTE-A	LTE-A	LTE-A	LTE-A	*K*= 4096	LTE	LTE-A	*K*= 6144
Architecture		PMAP	PMAP	PMAP	PMAP	PMAP	PMAP	XMAP	XMAP	FPMAP
Nb decoders		82	26	13	13	24	28	74	180	1
Nb max it.		5.5	6.0	8.0	5.5	6.0	8.0	5.5	6.0	39.0
Throughput (*Gb/s*)	1000	202.6	124.2	47.3	68.8	76.8	183.2	206.0	300.0	65.4
Area (*mm*^{2})	8.3	8.2	8.1	7.8	7.8	8.0	8.2	8.3	8.3	4.4
Power (*W*)	1.1	32.2	n/a	4.3	3.0	8.5	25.4	21.5	36.7	6.6
Area eff. (*Gb/s/mm*^{2})	120.6	24.6	15.4	6.0	8.8	9.6	22.3	24.9	36.4	14.9
Power dens. (*W/mm*^{2})	0.1	3.91	n/a	0.55	0.38	1.05	3.08	2.59	4.44	1.51
Energy eff. (*pJ/bit*)	1.1	158.9	n/a	91.1	43.1	110.0	138.4	104.2	122.2	101.5

	Use case	SoA Turbo Decoder								
	Hybrid fiber-wireless	[65] [66]	[67]	[68]	[68]	[69]	[70]	[76]	[77]	[83]
Latency (*µs*)	0.2	2.5	1.3	1.7	1.2	1.9	0.6	2.2	3.7	0.1
Frequency (*MHz*)	1000	1000.0	1000.0	1000.0	1000.0	1000.0	818.0	1000.0	1000.0	413.7

Table 31: Gap analysis of Turbo code implementations for hybrid fiber-wireless use case at 7 nm

- At 7 nm, none of the architectures are able to achieve the required throughput. The area limitation only allows three implementations to provide throughputs higher than 200 Gb/s.
- Again, only the XMAP architecture presented in [77] is able to achieve a throughput value of 300 Gb/s. However, the energy efficiency and power density of this decoder does not meet the requirements: respective factors of 110 and 45 have still to be gained.
- As for the latency, this implementation cannot meet the requirement as a deviation factor of 18 is observed with respect to the target. Only the fully parallel MAP decoder is able to meet it in 7 nm.

3.2.2.8 High-Throughput Satellites

	Use case	SoA Turbo Decoder								
	High throughput satellites	[65] [66]	[67]	[68]	[68]	[69]	[70]	[76]	[77]	[83]
Code Flexibility		LTE-A	LTE-A	LTE-A	LTE-A	LTE-A	*K*= 4096	LTE	LTE-A	*K*= 6144
Architecture		PMAP	PMAP	PMAP	PMAP	PMAP	PMAP	XMAP	XMAP	FPMAP
Nb decoders		99	32	16	16	29	34	89	218	2
Nb max it.		5.5	6.0	8.0	5.5	6.0	8.0	5.5	6.0	39.0
Throughput (*Gb/s*)	1000	244.6	152.9	58.2	84.7	92.8	222.5	247.7	363.3	130.7
Area (mm^2)	10	9.9	9.9	9.7	9.7	9.7	10.0	10.0	10.0	8.8
Power (*W*)	0.5	38.9	n/a	5.3	3.6	10.2	30.8	25.8	44.4	13.3
Area eff. ($Gb/s/mm^2$)	–	24.6	15.4	6.0	8.8	9.6	22.3	24.9	36.4	14.9
Power dens. (W/mm^2)	–	3.91	n/a	0.55	0.38	1.05	3.08	2.59	4.44	1.51
Energy eff. (*pJ/bit*)	0.5	158.9	n/a	91.1	43.1	110.0	138.4	104.2	122.2	101.5
Latency (*µs*)	10000	2.5	1.3	1.7	1.2	1.9	0.6	2.2	3.7	0.1
Frequency (*MHz*)	1000	1000.0	1000.0	1000.0	1000.0	1000.0	818.0	1000.0	1000.0	413.7

Table 32: Gap analysis of Turbo code implementations for high throughput satellites use case at 7 nm

- At 7 nm, none of the architectures are able to achieve the required throughput, 1 Tb/s. However four implementations provide throughputs higher than 200 Gb/s, the most efficient architecture being again the XMAP architecture presented in [77]. However, the energy efficiency of these decoders does not meet the corresponding requirement: a factor of more than 200 has still to be gained.
- For the high throughput satellites use case, the latency requirement is not very stringent and all the architectures are compliant with the target value.

3.2.3 Summary of the Gap Analysis for Turbo Codes

At 7 nm, the throughput of some use cases, such as virtual reality, wireless intra-device communications or wireless backhaul, can be approached or even achieved. However, for these use cases, power density and energy efficiency are the main issues to be addressed in the EPIC project.

Different use-cases provide different conclusions with one common aspect related to the fact that the existing SoA architectures are unable to suit EPIC use-case constraints. Despite being far from the EPIC targets in most cases, the PMAP architecture of [65] [66] and the XMAP architecture of [76] [77] seem to be the most promising ones.

Turbo codes offer inherent flexibility due to the use of puncturing for code rate control. This flexibility comes at the price of increased area and reduced efficiency. However, for some of the use cases like virtual reality, fronthaul and backhaul, data centre, hybrid fiber-wireless networks and high speed satellite, different levels (medium to high) of flexibility are required. This inherent feature can therefore partly be compensated for.

An important aspect to be noted is that all existing SoA Turbo code architectures do not apply stopping criteria for the MAP iterative decoding. When applied, such criteria should be able to improve achieved throughput and efficiency values without sacrificing achieved error correction performance. Moreover, the SoA architectures considered in this report were not designed to suit EPIC constraints in the first place. For example, they were all thought to provide performance levels with minimum degradation related to a hardware implementation. When targeting extremely high throughput as in EPIC use cases, trade-offs between performance and complexity can be investigated in order to improve efficiency and reduce power consumption. For example, such trade-offs can include reducing the number of iterations and/or reducing the number of quantization bits. Moreover, introducing additional structuring in code design could facilitate decoding parallelization and improve hardware efficiency. These potential ways of improvement are all to be investigated in the context of EPIC.

3.3 LDPC Codes

3.3.1 SoA of LDPC Codes

3.3.1.1 LDPC Block Codes

LDPC block codes (LDPC-BC) are linear block codes defined by a sparse parity-check matrix H of dimension $M \times N$, i.e., every message x that satisfies $Hx = 0$ in modulo-2 arithmetic is a valid code word. Practically all modern LDPC-BCs are structured LDPC-BCs based on protographs. These codes are defined by an $M_p \times N_p$ non-binary proto-matrix H_p, in which each entry represents the shift-value of a circulant identity matrix of size $Q \times Q$. The parity-check matrix H is then obtained by replacing each entry in H_p, with an identity matrix shifted with the indicated shift-value. We then have $M = QM_p$ and $N = QN_p$.

LDPC-BCs can be represented by a Tanner graph. This bipartite graph contains two sets of nodes denoted as variable nodes (VN) and check nodes (CN). Each variable node represents one column of H and thus corresponds to one of the N code bits, each check node represents one row of H and thus corresponds to one of the M parity checks. Edges between variable and check nodes reflect the “1” entries in H. In belief propagation based decoding, like the min-sum or the sum-product algorithm, variable and check nodes iteratively exchange probabilistic messages representing a node’s respective confidence on a bit decision.

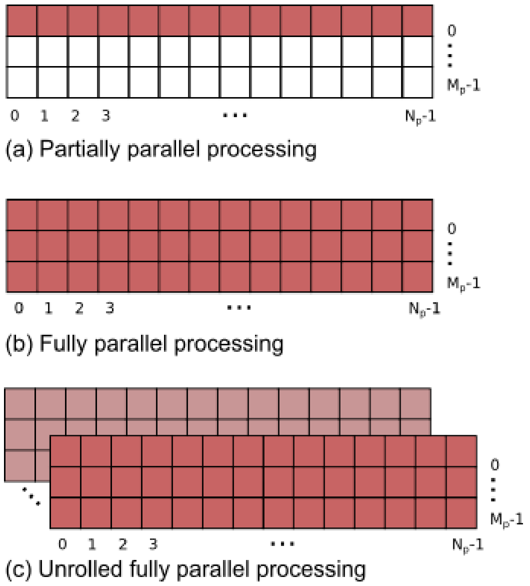
A simple, yet effective model to estimate the throughput of a decoder architecture A is based on the average number of edges the architecture processes in one clock cycle, denoted as $\#proc\_edges(A)$. Let $\#edges(H)$ denote the number of “1”s in H and $R = (N - M)/M$ be the code rate and f the operating frequency, then the information throughput of an architecture A for one iteration can be approximated by

$$T_{BC}(H,A) = \frac{\#proc\_edges(A)}{\#edges(H)} \cdot N_p \cdot Q \cdot R \cdot f \qquad [\mathrm{bits/s/iteration}] \qquad (3.4)$$

Based on equation$T_{BC}(H,A) = \frac{\#proc_edges(A)}{\#edges(H)} \cdot N_p \cdot Q \cdot R \cdot f \qquad [\mathrm{bits/s/iteration}]$ (3.4), we can define three categories of architectures:

1) $\#proc\_edges(A) < \#edges(H)$: partially parallel
2) $\#proc\_edges(A) = \#edges(H)$: fully parallel
3) $\#proc\_edges(A) > \#edges(H)$: unrolled fully parallel

The respective processing of H is illustrated in Figure 17, where submatrices that are processed in parallel are marked in red.

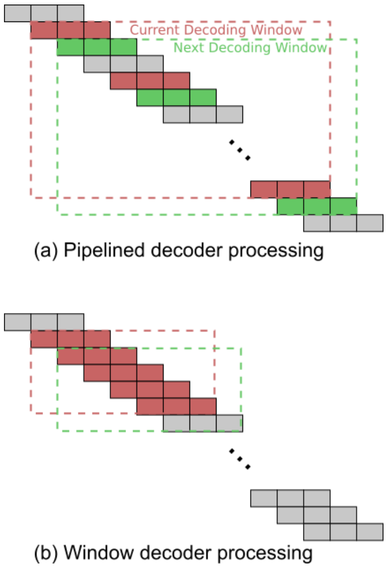


(a) Partially parallel processing

(b) Fully parallel processing

(c) Unrolled fully parallel processing

Figure 17: Hardware mappings for LDPC-BC



(a) Pipelined decoder processing

(b) Window decoder processing

Figure 18: Hardware mappings for LDPC-CC

Partially parallel architectures are applicable for medium throughputs (single to tens of Gb/s). Here, only a subset of edges, more precisely P Q-matrices are processed in parallel. This can be done either in a row-based or in a column-based manner. The sequential processing of rows-/columns allows layered decoding, i.e. taking advantage of intermediate node updates, which accelerates convergence and thus reduces the amount of iterations.

Fully parallel architectures are applicable for high throughputs (tens to hundreds of Gb/s). Here, all variable and check nodes are instantiated and the edges are hardwired between them, i.e. the Tanner graph is mapped one-to-one into hardware. Such an architecture lacks in flexibility, but has no limitations on the structure of H. Routing congestions, especially for large block sizes is a major challenge in this approach. Since all VNs and CNs are updated in parallel, also denoted as two-phase scheduling, it is not possible to have sub-iterations as in layered decoding schedules. As a consequence such an architecture requires more iterations to achieve the same communications performance.

*Unrolled fully parallel architecture*s are applicable for very high throughputs (hundreds to thousands of Gb/s). Here, the decoding iterations are unrolled and pipelined. If the check nodes are also pipelined such an architecture finishes the decoding of a complete code word every clock cycle. Alike in the fully parallel architecture, flexibility is limited and only two-phase scheduling is possible. However, the unrolled architecture implies mainly local wires, which reduces the routing congestions.

Assuming a given code, we can use equation $T_{BC}(H,A) = \frac{\#proc\_edges(A)}{\#edges(H)} \cdot N_p \cdot Q \cdot R \cdot f \quad [\mathrm{bits/s/iteration}]$ (3.4) to determine the number of edges an architecture A must process in parallel to achieve a certain throughput. As an example, we select the well-known WiMAX code with the parameters$N_p = 24$, $Q = 96$, $R = 5/6$ and 5 decoding iterations. Furthermore we assume $f = 800$ MHz and one clock cycle for the processing of one iteration. The respective results are shown in Table 33.

#edges(H) = 7,680	#proc_edges(A)	#proc_edges(A) required for 250 Gb/s	#proc_edges(A) required for 500 Gb/s	#proc_edges(A) required for 1000 Gb/s
Part. parallel (row)	1,920	6,250	12,500	25,000
Fully parallel	7,680	6,250	12,500	25,000
Unrolled	38,400	6,250	12,500	25,000

Table 33: Comparison of the actual number of processed edges of an architecture and the number of required edges for different throughputs.

Table 34 gives an overview on SoA LDPC-BC decoder implementations. To ease the comparison over different technology nodes, Table 35 shows the respective projections to 7 nm.

Ref.	Process techn. [nm]	Arch.	Max. Iter.	Area [mm²]	Freq. [MHz]	Info TP [Gb/s]	Energy eff. [pJ/bit/it]	Area eff. [Gb/s /mm^2]	Power Density [mW/mm^2]
[84]	65	Partially Parallel	10	1,2	500	**3,1**[1]	20,3	2,6	525
[85]	65	Partially Parallel	10	1,3	400	**6,7**[1]	8,0	5,2	414
[86]	28	Partially Parallel	4	0,8	470	**18,4**[2]	4,5	23,6	213
[87]	40	Fully Parallel	24	2,3	530	**5,4**[1]	17,7	2,3	1000
[88]	65	Fully Parallel	11	4,8	195	**30,4**[1]	4,1	6,3	283
[89]	28	Unrolled	9	2,8	238	**130**[1]	0,8	46,4	343
[90]	28	Unrolled	5	16,2	862	**495**[1]	5,4	30,6	824

Table 34: Comparison of SoA LDPC-BC decoder implementations.

1) For the maximum numbers of iterations
2) For an average of 2 iterations

Ref.	Process techn. [nm]	Arch.	Max. Iter.	Area [mm²]	Freq. [MHz]	Info TP [Gb/s]	Energy eff. [pJ/bit/it]	Area eff. [Gb/s /mm^2]	Power Density [mW/mm^2]
[84]	7	Partially Parallel	10	0,02	2924	**18,1**[1]	2,19	820	3070
[85]	7	Partially Parallel	10	0,02	2339	**39,2**[1]	0,86	1636	2420
[86]	7	Partially Parallel	4	0,07	1410	**55,2**[2]	1,13	849	639
[87]	7	Fully Parallel	24	0,10	2109	**21,5**[1]	3,11	213	3980
[88]	7	Fully Parallel	11	0,09	1140	**177,8**[1]	0,44	2011	1656
[89]	7	Unrolled	9	0,23	714	**390,0**[1]	0,21	1671	1029
[90]	7	Unrolled	5	1,35	2586	**1485,0**[1]	1,35	1100	2472

Table 35: Comparison of SoA LDPC-BC decoder implementations scaled to a 7 nm node

1) For the maximum numbers of iterations
2) For an average of 2 iterations

3.3.1.2 LDPC Convolutional Codes

LDPC-CCs are composed of a diagonal band of submatrices $H^{(m)}$. Each row of H is composed of $(m_s + 1)$ submatrices and each submatrix $H^{(m)}$ has $(c - b)$ rows and c columns. The code rate is defined as $R = b/c$. Like LDPC-BCs, the decoding of LDPC-CCs is based on belief propagation. However, the diagonal structure of H facilitates sliding window schedules, which exhibit lower computational complexity compared to block decoding schedules [91]. In literature, there are basically two types of window decoding architectures: the pipelined window decoder [92] and the window decoder architecture as proposed in [93]. The pipelined decoder consists of k processors, where each processor p_i, with $i = 1 \dots k$, processes $(m_s + 1)$ submatrices (=one row of H) in parallel. The rows are assigned to processors such that there is no conflict in the window when accessing the corresponding VNs of the submatrices (highlighted red in Figure 18(a)). Thus, the architecture with k processors has $k \cdot (c - n)$ CN units and $k \cdot (m_s + 1) \cdot c$ VN units. After processing the rows in parallel, the window is shifted diagonally by one submatrix to start the processing of the next columns in parallel (highlighted green in Figure 18(a)). Assuming that #pclkh denotes the number of clock cycles of a processor to process a submatrix, then the information bit throughput of the pipelined window decoder architecture can be approximated by

$$T_{CC}(H, A) = \frac{c}{\#pclkh} \cdot R \cdot f = \frac{b}{\#pclkh} \cdot f \quad \text{[bits/s]} \qquad (3.4)$$

The throughput does not depend on the number of decoding iterations (= number of processors), but strongly on the number of information bits b corresponding to one submatrix. Therefore, large submatrices are required to achieve high throughput. Table 36: Comparison of SoA LDPC-CC decoder implementations. lists SoA pipelined decoder implementations. Again, to ease comparison, the values are also scaled to a 7 nm node in Table 37. Compared to the SoA LDPC-BC decoders in Table 35, the maximum throughput of the LDPC-CC implementations is more than one order of magnitude lower. One reason for this is the low number of information bits that can be decoded in each window step ($= b$). Furthermore, the decoders in [94] and [95] are designed in compliance with the IEEE 1901 standard, which specifies a throughput of only 300Mb/s.

Ref.	Proces s techn. [nm]	Code Rate	#Proc.	Area [mm²]	Freq. [MHz]	Max. info TP [Gb/s]	Max. Energy eff. [pJ/bit/proc]	Area eff. [Gb/s /mm^2]	Power Density [mW/mm^2]
[96]	90	1/2 - 5/6	5	2,2	198	**2,37**	24.0	1,1	126.8
[94]	130	1/2 - 4/5	10	3,6	180	**0,24**	83.3	0,1	56.3
[97]	65	1/2 - 4/5	6	1,2	322	**7,72**	8.9	6,5	344.5
[95]	65	1/2 - 4/5	10	2,3	376	**0,5**	54.0	0,2	120.0

Table 36: Comparison of SoA LDPC-CC decoder implementations.

Ref.	Process techn. [nm]	Code Rate	#Proc.	Area [mm²]	Freq. [MHz]	Max. info TP [Gb/s]	Max. Energy eff. [pJ/bit/proc]	Area eff. [Gb/s /mm^2]	Power Density [mW/mm^2]
[96]	7	1/2 - 5/6	5	0,02	1498	**17,9**	1,86	779	960
[94]	7	1/2 - 4/5	10	0,02	1823	**2,4**	4,49	129	571
[97]	7	1/2 - 4/5	6	0,02	1883	**45,1**	0,95	2060	2015
[95]	7	1/2 - 4/5	10	0,04	2199	**2,9**	5,82	71	702

Table 37: Comparison of SoA LDPC-CC decoder implementations scaled to a 7 nm node.

The window decoder as proposed in [93] operates on a smaller window than the pipelined decoder (see Figure 18(b)). Unlike the pipelined decoder, the window decoder performs multiple iterations I, i.e. VN-/CN-updates, before the window is shifted by one submatrix. The processing within each window corresponds to the processing of a LDPC-BC and can be partially parallel or fully parallel. The smaller processing window compared to the pipelined decoder results in less decoding latency.

So far, no hardware implementation of the window decoder has been published.

3.3.2 Gap Analysis

The following section provides a detailed gap analysis between the FEC KPI of the use cases and the SoA LDPC code decoder implementations presented in the previous section. For the sake of clarity we limit the number of SoA decoders in the gap analysis to one partially parallel [85], one fully parallel [87] and two unrolled fully parallel LDPC-BC decoders [89] [90] as well as two pipelined LDPC-CC decoders [96] [97]. In this way, at least one of the previously discussed architectures is represented in the comparison.

Communications performance is not considered explicitly in the gap analysis. In order to enable a fair comparison, the decoders are scaled to a fixed number of 8 iterations/processors. Depending on the architecture a linear scaling factor was applied. For partially and fully parallel LDPC-BC decoder throughput, latency, area efficiency and energy efficiency were scaled; for the unrolled LDPC-BC decoder and the pipelined LDPC-CC decoder area, power, latency, area efficiency and energy efficiency were scaled.

3.3.2.1 Data Kiosk

	Use case	SoA LDPC Decoder					
	Data Kiosk	[85]	[87]	[89]	[90]	[96]	[97]
		LDPC-BC	LDPC-BC	LDPC-BC	LDPC-BC	LDPC-CC	LDPC-CC
		partially	fully	unrolled	unrolled	pip. WD	pip. WD
Num. of Decoders	-	48	33	3	2	84	42
Throughput (Gb/s)	1000	1005.0	1008.7	1170.0	1148.5	1005.5	1007.0
Area (mm²)	4.54	1.1	3.3	0.6	4.3	3.1	1.2
Power (W)	0.9	1.2	6.3	0.6	4.1	2.0	1.3
Area Eff. (Gb/s/mm²)	220.1	874.5	302.2	1880.4	265.9	325.0	820.5
Pow. Den. (W/mm²)	0.1	1.0	1.9	1.0	1.0	0.6	1.1
Energy Eff. (pJ/bit)	0.9	1.2	6.2	0.5	3.6	2.0	1.3
Latency (µs)	500	0.03	0.06	0.03	0.10	0.34	NA
Freq. (MHz)	1000	1000.0	1000.0	714.0	1000.0	1000.0	1000.0

Table 38: Gap analysis of LDPC code decoder implementations for the data kiosk use case at 7 nm.

- The data kiosk use case is characterized by a high throughput requirement of at least 1 Tb/s. This requirement can be achieved with all SoA decoders when increasing the number of decoder cores.
- Only the unrolled LDPC decoder from [89] shows the potential to meet the requirements in 7 nm, except for the power density limit which is exceeded by a factor of 10.

3.3.2.2 Mobile Virtual Reality

	Use case	SoA LDPC Decoder					
	Virtual Reality	[85]	[87]	[89]	[90]	[96]	[97]
		LDPC-BC	LDPC-BC	LDPC-BC	LDPC-BC	LDPC-CC	LDPC-CC
		partially	fully	unrolled	unrolled	pip. WD	pip. WD
Num. of Decoders	-	24	17	2	1	42	21
Throughput (Gb/s)	500	502.5	519.6	780.0	574.2	502.7	503.5
Area (mm²)	9.1	0.6	1.7	0.4	2.2	1.5	0.6
Power (W)	0.2	0.6	3.2	0.4	2.1	1.0	0.7
Area Eff. (Gb/s/mm²)	54.8	874.5	302.2	1880.4	265.9	325.0	820.5
Pow. Den. (W/mm²)	0.03	1.0	1.9	1.0	1.0	0.6	1.1
Energy Eff. (pJ/bit)	0.48	1.2	6.2	0.5	3.6	2.0	1.3
Latency (µs)	500	0.03	0.06	0.03	0.10	0.34	NA
Freq. (MHz)	1000	1000.0	1000.0	714.0	1000.0	1000.0	1000.0

Table 39: Gap analysis of LDPC code decoder implementations for the mobile virtual reality use case at 7 nm.

- The tight requirements regarding power consumption and related metrics pose a major challenge in the mobile virtual reality use case, whereas throughput and area are of minor concern.

- Power density is even violated by at least a factor of 20. Considering the fact, that there are large margins in area, new architectural concepts on exploiting this “dark silicon” need to be investigated.

3.3.2.3 Wireless Intra-Device Communication

	Use case	SoA LDPC Decoder					
	Intra Device	[85]	[87]	[89]	[90]	[96]	[97]
		LDPC-BC	LDPC-BC	LDPC-BC	LDPC-BC	LDPC-CC	LDPC-CC
		partially	fully	unrolled	unrolled	pip. WD	pip. WD
Num. of Decoders	-	24	17	2	1	42	21
Throughput (Gb/s)	500	502.5	519.6	780.0	574.2	502.7	503.5
Area (mm^2)	9.6	0.6	1.7	0.4	2.2	1.5	0.6
Power (W)	0.5	0.6	3.2	0.4	2.1	1.0	0.7
Area Eff. (Gb/s/mm^2)	52.0	874.5	302.2	1880.4	265.9	325.0	820.5
Pow.Den. (W/mm^2)	0.052	1.0	1.9	1.0	1.0	0.6	1.1
Energy Eff. (pJ/bit)	1	1.2	6.2	0.5	3.6	2.0	1.3
Latency (µs)	0.2	0.03	0.06	0.03	0.10	0.34	NA
Freq. (MHz)	1000	1000.0	1000.0	714.0	1000.0	1000.0	1000.0

Table 40: Gap analysis of LDPC code decoder implementations for the wireless intra-device communication use case at 7 nm.

- With regard to its requirements, the wireless intra-device use case has a lot in common with the mobile virtual reality use case. The main difference lies in the substantially lower latency. However, only the LDPC-CC decoder of [96] cannot meet this value.

3.3.2.4 Wireless Fronthaul / Backhaul

	Use case	SoA LDPC Decoder					
	Backhaul	[85]	[87]	[89]	[90]	[96]	[97]
		LDPC-BC	LDPC-BC	LDPC-BC	LDPC-BC	LDPC-CC	LDPC-CC
		partially	fully	unrolled	unrolled	pip. WD	pip. WD
Num. of Decoders	-	12	9	1	1	21	11
Throughput (Gb/s)	250	251.3	275.1	390.0	574.2	251.4	263.7
Area (mm^2)	10	0.3	0.9	0.2	2.2	0.8	0.3
Power (W)	0.9	0.3	1.7	0.2	2.1	0.5	0.3
Area Eff. (Gb/s/mm^2)	25	874.5	302.2	1880.4	265.9	325.0	820.5
Pow. Den. (W/mm^2)	0.09	1.0	1.9	1.0	1.0	0.6	1.1
Energy Eff. (pJ/bit)	3.6	1.2	6.2	0.5	3.6	2.0	1.3
Latency (µs)	1	0.03	0.06	0.03	0.10	0.34	NA
Freq. (MHz)	1000	1000.0	1000.0	714.0	1000.0	1000.0	1000.0

Table 41: Gap analysis of LDPC code decoder implementations for the wireless backhaul use case at 7 nm.

- The wireless fronthaul use case has the lowest throughput requirements. A single unrolled decoder is sufficient to exceed the 250 Gb/s target throughput.
- The pipelined window decoder from [96] is the only candidate that misses the latency.

- All decoders violate the power density requirement of the wireless backhaul use case. However, in 7 nm the core occupies only a small amount (5% for [89]) of the available area, leaving room for an improvement of the power density. Furthermore the unrolled decoders [89] [90] exceed the target throughput by 56% and 130% respectively. This allows for voltage scaling which can significantly reduce power consumption.

	Use case	SoA LDPC Decoder					
	Fronthaul	[85]	[87]	[89]	[90]	[96]	[97]
		LDPC-BC	LDPC-BC	LDPC-BC	LDPC-BC	LDPC-CC	LDPC-CC
		partially	fully	unrolled	unrolled	pip. WD	pip. WD
Num. of Decoders	-	48	33	3	2	84	42
Throughput (Gb/s)	1000	1005.0	1008.7	1170.0	1148.5	1005.5	1007.0
Area (mm²)	10	1.1	3.3	0.6	4.3	3.1	1.2
Power (W)	0.6	1.2	6.3	0.6	4.1	2.0	1.3
Area Eff. (Gb/s/mm²)	100	874.5	302.2	1880.4	265.9	325.0	820.5
Pow. Den. (W/mm²)	0.06	1.0	1.9	1.0	1.0	0.6	1.1
Energy Eff. (pJ/bit)	0.6	1.2	6.2	0.5	3.6	2.0	1.3
Latency (µs)	1	0.03	0.06	0.03	0.10	0.34	NA
Freq. (MHz)	1000	1000.0	1000.0	714.0	1000.0	1000.0	1000.0

Table 42: Gap analysis of LDPC code decoder implementations for the wireless fronthaul use case at 7 nm.

- Again, the decoder that is closest to fulfilling all requirements is the unrolled decoder architecture from [89].

3.3.2.5 Data Centre

	Use case	SoA LDPC Decoder					
	Data Centre	[85]	[87]	[89]	[90]	[96]	[97]
		LDPC-BC	LDPC-BC	LDPC-BC	LDPC-BC	LDPC-CC	LDPC-CC
		partially	fully	unrolled	unrolled	pip. WD	pip. WD
Num. of Decoders	-	48	33	3	2	84	42
Throughput (Gb/s)	1000	1005.0	1008.7	1170.0	1148.5	1005.5	1007.0
Area (mm²)	6.1	1.1	3.3	0.6	4.3	3.1	1.2
Power (W)	0.75	1.2	6.3	0.6	4.1	2.0	1.3
Area Eff. (Gb/s/mm²)	163.0	874.5	302.2	1880.4	265.9	325.0	820.5
Pow. Den. (W/mm²)	0.1	1.0	1.9	1.0	1.0	0.6	1.1
Energy Eff. (pJ/bit)	0.75	1.2	6.2	0.5	3.6	2.0	1.3
Latency (µs)	0.1	0.03	0.06	0.03	0.10	0.34	NA
Freq. (MHz)	1000	1000.0	1000.0	714.0	1000.0	1000.0	1000.0

Table 43: Gap analysis of LDPC code decoder implementations for the data centre use case at 7 nm.

- All SoA architectures meet the throughput and area efficiency requirements, but miss the target power density.
-

3.3.2.6 Hybrid Fiber-Wireless Networks

	Use case	SoA LDPC Decoder					
	Hybrid Fiber Wireless	[85]	[87]	[89]	[90]	[96]	[97]
		LDPC-BC	LDPC-BC	LDPC-BC	LDPC-BC	LDPC-CC	LDPC-CC
		partially	fully	unrolled	unrolled	pip. WD	pip. WD
Num. of Decoders	-	48	33	3	2	84	42
Throughput (Gb/s)	1000	1005.0	1008.7	1170.0	1148.5	1005.5	1007.0
Area (mm^2)	8.3	1.1	3.3	0.6	4.3	3.1	1.2
Power (W)	0.375	1.2	6.3	0.6	4.1	2.0	1.3
Area Eff. (Gb/s/mm^2)	121	874.5	302.2	1880.4	265.9	325.0	820.5
Pow. Den. (W/mm^2)	0.045	1.0	1.9	1.0	1.0	0.6	1.1
Energy Eff. (pJ/bit)	0.375	1.2	6.2	0.5	3.6	2.0	1.3
Latency (µs)	1	0.03	0.06	0.03	0.10	0.34	NA
Freq. (MHz)	1000	1000.0	1000.0	714.0	1000.0	1000.0	1000.0

Table 44: Gap analysis of LDPC code decoder implementations for the hybrid fiber-wireless networks use case at 7 nm.

- For the hybrid fiber-wireless networks use case no SoA architecture is able to fulfil the tight power, power density and energy efficiency requirements.

3.3.2.7 High-Throughput Satellites

	Use case	SoA LDPC Decoder					
	High Throughput Satellites	[85]	[87]	[89]	[90]	[96]	[97]
		LDPC-BC	LDPC-BC	LDPC-BC	LDPC-BC	LDPC-CC	LDPC-CC
		partially	fully	unrolled	unrolled	pip. WD	pip. WD
Num. of Decoders	-	48	33	3	2	84	42
Throughput (Gb/s)	1000	1005.0	1008.7	1170.0	1148.5	1005.5	1007.0
Area (mm^2)	10	1.1	3.3	0.6	4.3	3.1	1.2
Power (W)	1	1.2	6.3	0.6	4.1	2.0	1.3
Area Eff. (Gb/s/mm^2)	100	874.5	302.2	1880.4	265.9	325.0	820.5
Pow. Den. (W/mm^2)	0.05	1.0	1.9	1.0	1.0	0.6	1.1
Energy Eff. (pJ/bit)	0.5	1.2	6.2	0.5	3.6	2.0	1.3
Latency (µs)	10000	0.03	0.06	0.03	0.10	0.34	NA
Freq. (MHz)	1000	1000.0	1000.0	714.0	1000.0	1000.0	1000.0

Table 45: Gap analysis of LDPC code decoder implementations for the high-throughput satellites use case at 7 nm.

- The latency requirements in the high-throughput satellites use case are relaxed and can be easily fulfilled by all SoA decoders.

3.3.3 Summary of the Gap Analysis for LDPC Codes

In summary, the gap analysis for LDPC codes has shown that even when scaled to 7 nm none of the presented SoA decoders is able to meet the requirements of the EPIC use cases. Considering the different FEC KPI, throughput turns out to be not the dominating challenge for LDPC codes. To achieve the target throughput multiple decoders can work in parallel without sacrificing the respective area constraints. However, the constraints on power consumption and related metrics such as energy efficiency and power density are often not achieved.

Power density is the biggest challenge in all use cases. In fact, none of the SoA decoders was able to meet the power density requirements in any of the use cases. However, it must be noted that the decoder cores occupy only a small amount of the available area (ranging between 5% and 60%). Therefore, new architectures that exploit this area margin to reduce the power density need to be investigated.

Among the different decoder architectures, the unrolled fully parallel decoders [89] [90] exhibit the smallest gap to the requirements. For the data centre, backhaul, intra-device communication and data kiosk use cases, the decoder from [89] is even able to fulfil all requirements except for power density. The low power consumption of the decoder results from the dataflow dominated architecture and the high locality. However, it must be noted that it offers no flexibility w.r.t. code rate and code length. Therefore, bringing flexibility to unrolled LDPC decoder architectures remains one of the main challenges for future research.

3.4 Polar Codes

The goal of this section is to present an assessment of the gap between the EPIC use case requirements and the state-of-the-art in Polar coding technology. Section 3.4.1 examines a selection of decoder implementations as representative of the SoA from the viewpoint of EPIC targets. Section 3.4.2 compares EPIC targets with the performance available by the SoA Polar codes.

3.4.1 SoA of Polar Codes

This section presents a survey of decoders for Polar codes under the EPIC project perspective. In order to identify the SoA of Polar decoders, we briefly explain various decoding algorithms and then indicate some viable implementation options.

A simple decoding algorithm for Polar codes is Successive Cancellation (SC) [98]. Under SC decoding, Polar codes have been proved to achieve the channel capacity of a binary-input discrete memoryless channel (B-DMC). However, SC decoder suffers from two problems. First, it is sequential in nature; thus, it is not suitable for the extreme throughputs demanded by EPIC use cases. Second, its finite-length performance is suboptimal. The first problem is remedied by using various “fast” versions of SC decoder that permit making many decisions in parallel. The second deficiency is rectified by using a SC List (SCL) decoder ([99], [100]) that maintains a list of candidate codewords and selects one in the end with the aid of a CRC. Although the SCL decoder achieves near ML performance with a large list size (list-32), the use of SCL with a list size of 8 or larger appears infeasible in EPIC due to severe constraints on area efficiency and energy efficiency. In order to achieve Tb/s throughput target of EPIC, inherently parallel algorithms such as Majority Logic [101] and Belief Propagation (BP) decoding [102] are preferable. Moreover, some EPIC uses cases (such as wireless intra device communications) require extremely low BERs, which may be difficult to achieve on a stand-alone basis by a Polar code; hence, one should not rule out the use of a concatenation scheme, where, e.g., an outer BCH code is employed to boost the communications performance of a Polar code. In terms of architectures, it appears almost certain from the beginning that parallel structures are the only viable option for meeting the Tb/s throughput requirement of the EPIC project.

The decoders included in the survey are shown in Table 46 with some implementation details. These decoder designs have been selected based on their promise of meeting the requirements of EPIC use cases with some further development effort. In other words, we present these designs as initial design space exploration points for creating a set of EPIC Polar coding solutions.

Reference		[103]	[103]	[104]	[105]	[106]	[107]	[108]	[109]
Short Name for		SC-U	SC-	SC-C	SCL-U	SCL-M	BP-D	BP-E	SCAN
Dec. Algorithm		Succ. Can.	Succ. Can.	Succ. Can.	Succ. Can. List	Succ. Can.	Belief Prop.	Belief Prop.	Soft Can.
Arch. Type		Unrolled Pipelined	Multi Mode	Comb.	Unrolled Pipelined	Multibit Decision	Double Column	Early Stop.	Reduced Latency
Block Length		1024	2048	1024	512	1024	1024	1024	1024
List Size		-	-	-	2	4	-	-	-
Payload		512	1365	$512^{flex.}$	427	512	$512^{flex.}$	512	512
Code Rate		1/2	2/3	$1/2^{flex.}$	5/6	1/2	$1/2^{flex.}$	1/2	1/2
CRC Length		0	0	0	8	N/A	0	0	0
Technology		65nm	65nm	90nm	28 nm	65nm	65nm	45nm	90nm
Iterations		-	-	-	-	-	6.57	23.0	1.025
Freq. (MHz)		500	350	2.5	468	400	300	500	520
Busy Interval		1	28	1	20	1022*	66*	57*	264*
Latency	**(CCs)**	364	504*	1	253*	1022	66*	56	264*
	(µs)	0.7*	1.4	0.7	0.5	2.6*	0.2*	0.1*	0.5
Net Tp. (Gb/s)		256^r	17.1^r	1.3^r	10.0^r	0.2^r	2.3^r	4.5	1.0^r
Area (mm^2)		12.4	3.6	3.2	0.9	2.1	1.5	N/A	1.1
Supply (V)		0.72	1.0	1.3	1.1	N/A	1.0	1.1	N/A
Power (mW)		3830	740	190.7	87	718	477.5	990*	N/A
Area Eff.		20.7*	4.8^r	0.4^r	11.5^r	0.1^r	1.6^r	N/A	0.9^r
Energy Eff.		15.0^r	43.4^r	149.0^r	8.7^r	3590.0^r	204.2^r	220.0	N/A
Power Den.		0.31*	0.21*	0.06*	0.1	0.34*	0.32*	N/A	N/A
Eb/No @10^{-3}		3.2^+	3.6^+	3^+	4.8^+	2.3^+	3.3^+	3.3^+	3.5^+
Eb/No @10^{-5}		3.4^+	N/A	3.4^+	4.9^+	N/A	N/A	N/A	N/A

Table 46: The implementation details of the SoA Polar decoders

* Not provided in the paper, calculated from the presented results

^{r} Provided in the paper, recalculated with respect to net throughput

$^+$ Observed from the communication performance graph presented in the paper

$^{flex.}$ The payload and the code rate are flexible and can be tuned during the run-time.

Salient points of the SoA Polar decoder implementations are given with respect to the type of the decoding algorithm.

SoA Polar decoders with SC algorithm

- The *successive cancellation unrolled (SC-U) decoder* [103] follows a deeply pipelined and unrolled architecture. Due to deep pipelining, the decoder can accept a new codeword at each clock cycle. Although deep pipelining enhances the throughput significantly, one drawback is the excessive memory area (proportional to latency) due to storing all information about every codeword and corresponding internal calculations during decoding.

- *The successive cancellation multimode (SC-M) decoder* [103] uses a partially pipelined and unrolled architecture. Due to partially pipelining, the decoder can accept a new codeword at every 28 clock cycles. Partially pipelining provides a favourable trade-off between throughput and resource complexity. Moreover, the decoder employs a multimode feature in the architecture that brings flexibility in a limited set of constituent code block lengths and code rates.

- The *successive cancellation combinational (SC-C)* decoder [104] is an implementation of an SC decoder using combinational (asynchronous) logic. This decoder takes one (very long) clock cycle to decode an entire codeword. The SCC decoder is one of the most power efficient decoder due to the lack of excessive switching activity in asynchronous logic, which reduces the dynamic power dissipation.

SoA Polar decoders with SCL algorithm

- The *successive cancellation list unrolled (SCL-U) decoder* [105] uses a semi-pipelined unrolled architecture on hardware. The semi-pipelining allows the decoder to accept a new codeword every 20 clock cycles. One drawback is that due to the nature of unrolling, it is infeasible to provide code flexibility with regard to block length or payload size. Due to the SCL decoding algorithm, the communication performance of the SCL-U decoder is preferable, when the spectral efficiency is high.

- The *successive cancellation list multibit (SCL-M) decoder* [106] uses a multibit decision strategy. At each decision step, the decoder can decide four bits simultaneously to enhance the throughput. Furthermore, using SCL algorithm with list-4 provides decoder to track four hard decision candidates together and increases the communication performance significantly. The drawback of the SCL algorithm is the low throughput, which is in the order of Gb/s. If the current technology is considered, at least thousandfold improvement is required to satisfy Tb/s EPIC target.

SoA Polar decoders with BP algorithm

- The *belief propagation double-column (BP-D) decoder* [107] employs naturally parallel and iterative BP algorithm. The decoder can operate on double stages (double-columns) in a single clock period for a better utilization of the clock period. In order to maximize the throughput, the BP-D decoder uses an SNR-aware decoding strategy with early termination based on convergence detection.

- The *belief propagation early stopping criteria (BP-E) decoder* [108] employs an advanced termination algorithm to reduce the average number of iterations. The termination algorithm is more effective in high SNR region.

SoA Polar decoders with SCAN algorithm

- The *soft-cancellation (SCAN) decoder* [109] has an iterative soft-output algorithm with the schedule of the SC decoding algorithm. SCAN decoder requires noticeably less average number of iterations compared to BP decoders to achieve similar communication performance. The decoder has the reduced-latency (RLSC) architecture, which introduces suboptimal calculations to increase the throughput. One drawback of RLSC is the communication performance degradation caused by the suboptimal computational components.

The FEC level KPI were listed in Table 1 as BER, latency, throughput, area efficiency, energy efficiency, power density, flexibility in code length and code rate. Due to the fact that some important KPI are missing for the BP-E decoder (power density, area efficiency) and SCAN decoder (power density, energy efficiency), we cannot carry out further detailed analysis for those decoders. Using the scaling methodology defined in Sec. 3.1, we scaled the other listed SoA Polar decoders to 7 nm CMOS technology in order to make a fair comparison. The scaled results are shown in Table 47 combined with the EPIC targets listed in Table 3.

	EPIC Targets	SoA Polar Decoders					
		[103]	[103]	[104]	[105]	[106]	[107]
		SC-U	**SC-M**	**SC-C**	**SCL-U**	**SCL-M**	**BP-D**
Throughput (Gb/s)	1000	1497.0	100.3	9.7	30.0	1.2	27.3
Area (mm^2)	10	0.23	0.07	0.03	0.07	0.04	0.03
Power (W)	1	2.4	0.5	0.1	0.1	0.5	0.3
Area Eff. (Gb/s/mm^2)	100	6571.6	1521.2	293.4	413.8	29.7	1005.9
Pow. Den. (W/mm^2)	0.1	10.6	7.1	3.4	0.9	11.5	11.1
Energy Eff. (pJ/bit)	1	1.6	4.6	11.6	2.2	385.7	11.0
Latency (µs)	0.02 - 10^3 *	0.12	0.25	0.10	0.18	0.44	0.04
Freq. (MHz)	1000	2923.8	2046.6	18.9	1404.0	2339.0	1754.3

Table 47: Comparison of SoA Polar decoders scaled to 7 nm

* The latency requirement of EPIC varies greatly with respect to the target use case

When the designs are compared with each other and with the EPIC targets, the following points emerge in terms of KPI.

- **Throughput:** The decoders with the unrolled architecture (SCL-U, SC-U SC-M) have a promising throughput results due to having a dedicated resource for each logic operation. However, the Tb/s EPIC throughput target has not been satisfied with the current SoA technology. Although the SC-U decoder appears to exceed 1 Tb/s at 7 nm, the frequency becomes infeasible to maintain. Even at 7 nm, the SCL-M decoder has noticeably low throughput caused by the long latency SCL algorithm with list-4. The SC-C, BP-D decoders have several Gb/s throughputs. For those decoders, a significant effort is required to reach the desired throughput.

- **Area:** The area utilization is proportional to the complexity of the algorithm and implementation. The SC-U decoder utilizes the largest area due to the dedicated hardware components. After the implementation results are scaled to 7 nm, all decoders satisfy the area requirement of EPIC easily.

- **Power:** As we identified before, the SC-C decoder is a power efficient decoder due to the combinational architecture. Although the SCL algorithm is complex, SCL-U decoder has a low power consumption due to the small list size as list-2. Except the SC-U decoder, all decoders satisfy the power requirement of EPIC.

- **Area efficiency:** All decoders except the SCL-M satisfy the area efficiency requirement of EPIC. Using low-complexity and high throughput algorithms increase the area efficiency significantly.

- **Power density:** The power density appears to be one of the most critical KPI as none of the decoders can satisfy the requirement. If we compare the decoders with each other SCL-U has a better power density than the others.

- **Energy efficiency:** Unrolled architecture based decoders have better energy efficiency due to their extremely high throughput, even if they consume more power. There is a significant gap between the energy efficiency of SCL-M and the requirement of EPIC.

- **Latency:** Although the original results indicate that the SCAN decoder requires approximately six times less iterations than the BP-D decoder at 4 dB Eb/No, the latency of the SCAN decoder is longer than the BP decoder due to the parallel processing nature of the BP algorithm. For the same reason, BP decoder has the shortest latency among the listed SoA decoders such that it may be a good candidate for latency-critical use cases such as intra-device communications. In general, the latency of the SC and the SCL algorithm based decoders is longer than the BP based decoders.

- **Frequency:** After the SoA Polar decoder results have been scaled to 7 nm, the clock frequency becomes infeasible to maintain except for the SC-C decoder. The gap analysis has been carried out in Sec. 3.4.2 with the feasible frequency values in order to make some realistic conclusions.

- **Flexibility:** Among the listed SoA decoders, only the SC-C and the BP-D decoders provide code flexibility without severely sacrificing the throughput and energy efficiency. The code flexibility may impact on the communication performance, especially in low SNR regions. Several EPIC use cases such as HTS and VR a require high degree of flexibility due to time varying and frequency selective channel conditions. The unrolled architecture based SCL-U and SC-U decoders do not have any degree of code flexibility and the SC-M decoder has limited flexibility.

- **Communication performance:** In theory, the SCL algorithm has the best communication performance among other candidates; however, it is also the most complex one. It is unclear at this point which decoder provides the best communication performance under the area and power constraints that are certain to set limits on the block lengths and list sizes that can be used with a list decoder.

3.4.2 Gap Analysis

This section presents an assessment of the gap between the EPIC use case targets and the SoA Polar decoders presented in section 3.4.1. The gap analysis has been carried out by scaling the original SoA results with respect to the scaling methods explained in section 3.1. Firstly, we apply a technology scaling to all SoA decoders to make a fair comparison under the same technology. Secondly, we perform a frequency scaling in order to reduce the clock frequency to a feasible value, 1 GHz for the 7 nm CMOS technology. Lastly, we apply a spatial parallelization by increasing the number of parallel decoders, until an area limit defined for each use case is reached. We emphasize the noteworthy points regarding the gap analysis for each EPIC user case.

3.4.2.1 Data Kiosk

The gap analysis for the data kiosk use case is shown in Table 48. Noteworthy points regarding the gap analysis are the following:

- The latency requirement of the data kiosk use case is 0.5 ms, which is rather low requirement. All of the decoders satisfy the latency requirement for the target technologies.

- All of the decoders satisfy both the area and the throughput requirements except the SCL-M. However, none of the decoders satisfies the power density requirement under the area limitation.
- Power density and energy efficiency are problematic for all of the decoders.
- The SCL-U, the SC-U and the SC-M decoders are the most promising candidates for this use case in terms of achieving 1 Tb/s throughput with the least number of parallel decoders.

	Use Case	SoA Polar Decoder					
	Data Kiosk	[103]	[103]	[104]	[105]	[106]	[107]
	-	SC-U	SC-M	SC-C	SCL-U	SCL-M	BP-D
Num. of Decoders		2	21	104	48	115	65
Throughput (Gb/s)	1000	1024	1029.1	1007.4	1025.6	57.6	1013.1
Area (mm^2)	4.5	4.5	4.5	4.5	4.5	4.5	4.5
Power (W)	0.9	1.7	4.8	11.7	2.2	22.2	11.1
Area Eff. (Gb/s/mm^2)	220.1	225.6	226.6	221.9	225.9	12.7	223.3
Pow. Den. (W/mm^2)	0.2	0.36	1.05	2.57	0.49	4.89	2.46
Energy Eff. (pJ/bit)	0.9	1.6	4.6	11.6	2.2	385.7	11
Latency (µs)	500	0.4	0.5	0.1	0.3	1	0.1
Freq. (MHz)	1000	1000	1000	18.9	1000	1000	1000

Table 48: Gap analysis of Polar code decoder implementations for data kiosk use case at 7 nm

3.4.2.2 Mobile Virtual Reality

The gap analysis for the mobile virtual reality use case is shown in Table 49. Noteworthy points regarding the gap analysis are as following:

- The latency requirement of the virtual reality use case is not demanding. Therefore, all the considered decoders satisfy the latency requirement easily.
- The SC-U decoder is the only decoder that satisfies the throughput requirement without spatial paralleling. With spatial paralleling, only the SCL-M decoder cannot satisfy the throughput requirement due to low area efficiency, even if 231 decoders are utilized in parallel.
- For the SC-U decoder, if we reduce the power density usage to 0.03 W/mm^2 in order not to exceed the limit, the area increases to 31.6 mm^2, which does not satisfy the area requirement.
- None of the decoders satisfy the power density limit, the SC-U decoder is closest to the limit.
- The most area efficient decoder is the SC-U due to unrolled and deeply pipelined architecture.
- The SC-U decoder is observed as the most promising decoder. However, there is still a gap between the requirements and the SC-U decoder.

Use Case	SoA Polar Decoder					
Virtual Reality	[103]	[103]	[104]	[105]	[106]	[107]
	SC-U	SC-M	SC-C	SCL-U	SCL-M	BP-D

	Use Case	SoA Polar Decoder					
	Virtual Reality	[103]	[103]	[104]	[105]	[106]	[107]
Num. of Decoders		1	11	52	24	231	33
Throughput (Gb/s)	500	512	539.1	503.7	512.8	115.8	514.4
Area (mm^2)	9.1	9.1	9.1	9.1	9.1	9.1	9.1
Power (W)	0.2	0.8	2.5	5.8	1.1	44.7	5.7
Area Eff. (Gb/s/mm^2)	54.8	56.1	59.1	55.3	56.2	12.7	56.4
Pow. Den. (W/mm^2)	0.03	0.09	0.27	0.64	0.12	4.89	0.62
Energy Eff. (pJ/bit)	0.5	1.6	4.6	11.6	2.2	385.7	11
Latency (µs)	500	0.4	0.5	0.1	0.3	1	0.1
Freq. (MHz)	1000	1000	1000	18.9	1000	1000	1000

Table 49: Gap analysis of Polar code decoder implementations for virtual reality use case at 7 nm

3.4.2.3 Wireless Intra-Device Communication

The gap analysis for the wireless intra-device communication use case is shown in Table 50. Noteworthy points regarding the gap analysis are the following:

- The latency requirement of the intra-device communication use case is very demanding. Only the fastest decoders (SC-C, BP-D) could satisfy these requirements, though other decoders are not far away from the target.
- For the SC-U decoder, if we reduce the power density to 0.05 W/mm2, in order not to exceed the power density limit, the area usage increases to 16.6 mm2.
- The SCL-U decoder and the SC-U decoder are observed as the most promising decoders to achieve the requirements in terms of power density and energy efficiency when the area is 9.6 mm2.

	Use Case	SoA Polar Decoder					
	Intra Dev. Com.	[103]	[103]	[104]	[105]	[106]	[107]
	-	SC-U	SC-M	SC-C	SCL-U	SCL-M	BP-D
Num. of Decoders		1	11	52	24	244	33
Throughput (Gb/s)	500	512	539.1	503.7	512.8	122.3	514.4
Area (mm^2)	9.6	9.6	9.6	9.6	9.6	9.6	9.6
Power (W)	0.5	0.8	2.5	5.8	1.1	47.2	5.7
Area Eff. (Gb/s/mm^2)	52	53.2	56.1	52.4	53.3	12.7	53.5
Pow. Den. (W/mm^2)	0.05	0.09	0.26	0.61	0.12	4.9	0.59
Energy Eff. (pJ/bit)	1	1.6	4.6	11.6	2.2	385.7	11
Latency (µs)	0.2	0.4	0.5	0.1	0.3	1	0.1
Freq. (MHz)	1000	1000	1000	18.9	1000	1000	1000

Table 50: Gap analysis of Polar code decoder implementations for intra-device communication use case at 7 nm

3.4.2.4 Data Centers

The gap analysis for the data centers use case is shown in Table 51. Noteworthy points regarding the gap analysis are the following:

- Only the fastest decoders, the SC-C and the BP-D decoders, can satisfy the latency requirement.
- Under the area limitation of 6.1 mm^2, all of the decoders except SCL-M manage to achieve 1 Tb/s throughput.
- Power, power density and energy efficiency requirements cannot be satisfied with any of the decoders.
- The SCL-U, the SC-U and the SC-M decoders are the most promising candidates for this use case in terms of achieving 1 Tb/s throughput with the least number of parallel decoders.

	Use Case	SoA Polar Decoder					
	Data Centre	[103]	[103]	[104]	[105]	[106]	[107]
	-	SC-U	SC-M	SC-C	SCL-U	SCL-M	BP-D
Num. of Decoders		2	21	104	48	155	65
Throughput (Gb/s)	1000	1024	1029.1	1007.4	1025.6	77.7	1013.1
Area (mm^2)	6.1	6.1	6.1	6.1	6.1	6.1	6.1
Power (W)	0.7	1.7	4.8	11.7	2.2	30	11.1
Area Eff. (Gb/s/mm^2)	163	166.7	167.5	164.1	167	12.7	165
Pow. Den. (W/mm^2)	0.12	0.27	0.78	1.9	0.36	4.88	1.81
Energy Eff. (pJ/bit)	0.8	1.6	4.6	11.6	2.2	385.7	11
Latency (μs)	0.1	0.4	0.5	0.1	0.3	1	0.1
Freq. (MHz)	1000	1000	1000	18.9	1000	1000	1000

Table 51: Gap analysis of Polar code decoder implementations for data centre use case at 7 nm

3.4.2.5 Hybrid Fiber-Wireless Networks

The gap analysis for the hybrid fiber-wireless networks use case is shown in Table 52. Noteworthy points regarding the gap analysis are the following:

- The SCL-U and the SC-U decoders are close to satisfying the latency requirement, but only the SC-C and the BP-D decoders can satisfy it.
- With the area scaling, power density values are significantly reduced; but decoders are still incapable of reaching the target power density value.
- The SC-U decoder is very close to satisfying both power density and energy efficiency requirements but cannot satisfy both.

	Use Case	SoA Polar Decoder					
	Hybrid Fiber-Wireless	[103]	[103]	[104]	[105]	[106]	[107]
	-	SC-U	SC-M	SC-C	SCL-U	SCL-M	BP-D
Num. of Decoders		2	21	104	48	210	65
Throughput (Gb/s)	1000	1024	1029.1	1007.4	1025.6	105.3	1013.1
Area (mm^2)	8.3	8.3	8.3	8.3	8.3	8.3	8.3
Power (W)	1.1	1.7	4.8	11.7	2.2	40.6	11.1
Area Eff. (Gb/s/mm^2)	120.6	123.5	124.1	121.5	123.7	12.7	122.2
Pow. Den. (W/mm^2)	0.14	0.2	0.58	1.41	0.27	4.9	1.34
Energy Eff. (pJ/bit)	1.125	1.6	4.6	11.6	2.2	385.7	11
Latency (µs)	0.2	0.4	0.5	0.1	0.3	1	0.1
Freq. (MHz)	1000	1000	1000	18.9	1000	1000	1000

Table 52: Gap analysis of Polar code decoder implementations for hybrid fiber-wireless use case at 7 nm

3.4.2.6 Wireless Fronthaul/Backhaul

The gap analysis for the wireless fronthaul networks use case is shown in Table 53. Noteworthy points regarding the gap analysis are the following:

- None of the SoA decoders could satisfy the power density or energy efficiency requirements of the fronthaul use case. Further improvements in algorithm and architecture are required to close the gap.
- Although none of the decoders is even close to reaching the target values, the SCL-U and the SC-U decoders are observed to be the most promising SoA decoders.

.

	Use Case	SoA Polar Decoder					
	Fronthaul	[103]	[103]	[104]	[105]	[106]	[107]
	-	SC-U	SC-M	SC-C	SCL-U	SCL-M	BP-D
Num. of Decoders		2	21	104	48	232	65
Throughput (Gb/s)	1000	1024	1029.1	1007.4	1025.6	116.3	1013.1
Area (mm^2)	10	10	10	10	10	10	10
Power (W)	0.6	1.7	4.8	11.7	2.2	44.8	11.1
Area Eff. (Gb/s/mm^2)	100	102.4	102.9	100.7	102.6	11.6	101.3
Pow. Den. (W/mm^2)	0.06	0.17	0.48	1.17	0.22	4.48	1.11
Energy Eff. (pJ/bit)	0.6	1.6	4.7	11.6	2.2	385.7	11
Latency (µs)	1	0.4	0.5	0.1	0.3	1	0.1
Freq. (MHz)	1000	1000	1000	18.9	1000	1000	1000

Table 53: Gap analysis of Polar code decoder implementations for fronthaul use case at 7 nm

The gap analysis for the wireless backhaul use case is shown in Table 54. Noteworthy points regarding the gap analysis are the following:

- With requirements for the backhaul use case not as demanding as those of the other use cases, some SoA decoders can satisfy all the KPI at the same time, based on SC-U and SCL-U architectures.

	Use Case	SoA Polar Decoder					
	Backhaul	[103]	[103]	[104]	[105]	[106]	[107]
	-	SC-U	SC-M	SC-C	SCL-U	SCL-M	BP-D
Num. of Decoders		1	5	26	12	232	16
Throughput (Gb/s)	250	512	245	251.9	256.4	116.3	249.4
Area (mm^2)	10	10	10	10	10	10	10
Power (W)	0.9	0.8	1.1	2.9	0.6	44.8	2.7
Area Eff. (Gb/s/mm^2)	25	51.2	24.5	25.2	25.6	11.6	24.9
Pow. Den. (W/mm^2)	0.09	0.08	0.11	0.29	0.06	4.48	0.27
Energy Eff. (pJ/bit)	3.6	1.6	4.6	11.6	2.2	385.7	11
Latency (μs)	1	0.4	0.5	0.1	0.3	1	0.1
Freq. (MHz)	1000	1000	1000	18.9	1000	1000	1000

Table 54: Gap analysis of Polar code decoder implementations for backhaul at 7 nm

3.4.2.7 High-Throughput Satellites

The gap analysis for the high-throughput satellites use case is shown in Table 55. Noteworthy points regarding the gap analysis are the following:

- The power density and energy efficiency requirements of the HTS are very strict. They are the limiting factors of this use case. Only the SC-U and the SCL-U decoders can come close to satisfying these requirements.
- The latency requirement of HTS is 10 ms, which is not very challenging.

	Use Case	SoA Polar Decoder					
	HTS	[103]	[103]	[104]	[105]	[106]	[107]
	-	SC-U	SC-M	SC-C	SCL-U	SCL-M	BP-D
Num. of Decoders		2	21	104	48	232	65
Throughput (Gb/s)	1000	1024	1029.1	1007.4	1025.6	116.3	1013.1
Area (mm^2)	10	10	10	10	10	10	10
Power (W)	0.5	1.7	4.8	11.7	2.2	44.8	11.1
Area Eff. (Gb/s/mm^2)	100	102.4	102.9	100.7	102.6	11.6	101.3
Pow. Den. (W/mm^2)	0.05	0.17	0.48	1.17	0.22	4.48	1.11
Energy Eff. (pJ/bit)	0.5	1.6	4.6	11.6	2.2	385.7	11
Latency (μs)	1000	0.4	0.5	0.1	0.3	1	0.1
Freq. (MHz)	1000	1000	1000	18.9	1000	1000	1000

Table 55: Gap analysis of Polar code decoder implementations for HTS use case at 7 nm

3.4.3 Summary of the Gap Analysis for Polar Codes

After the SoA Polar decoders are scaled down to 7 nm technology and analysed in section 3.4.2 regarding each use case KPI, the gap between the SoA Polar decoders and EPIC/use case requirements becomes clearer. It is shown that SoA decoders, scaled to 7 nm technology, can reach 1 Tb/s throughput when multiple decoders are used in parallel. The latency requirements of the use cases vary a lot from one use case to another. Therefore, reaching an overall conclusion, regarding latency requirements, is not viable. From Table 48 to Table 55, it is indicated that, power density and energy efficiency requirements are the most significant limiting factors of most of the use cases.

The SC-U decoder [103] appears to be the closest decoder implementation to satisfy all the KPI at the same time; however, there is still a moderate gap between the SC-U decoder and use case requirements. The most problematic areas for the SC-U decoder are observed to be power density, energy efficiency, and its lack of flexibility in both code length and code rate.

The most power density demanding use cases are hardest to achieve as power density is the most limiting KPI. The mobile virtual reality and the wireless fronthaul are so demanding that the power dissipation of many SoA Polar decoders is not even in the same order of magnitude.

The gap analysis shows that there is a significant gap, which will not be covered by advancing silicon technology, between the SoA decoders and EPIC requirements. Therefore, it can be concluded that in order to close the gap, further developments in both algorithms and implementations are necessary.

3.5 Summary of the Gap Analysis for All Codes

When the best SoA decoders of the three code families are scaled to 7 nm, none of them is able to meet all the requirements of the EPIC use cases.

Considering the different FEC KPI, the 1 Tb/s throughput turns out not to be the dominating challenge for LDPC codes and Polar codes. To achieve this 1 Tb/s target throughput, multiple decoders can work in parallel or unrolled architectures can be used without sacrificing the area constraints. However, the unrolled architectures introduce limitation on the block length and number of decoding iterations, and hence the communication performance required by some use cases may not be met.

Area efficiency is also not too crucial for LDPC codes and Polar codes. Turbo codes are able to meet the area efficiency requirement of the backhaul use case, while an improvement factor around x10 to reach area efficiency for the other use cases.

All the three codes classes can meet the latency requirements of virtual reality, data kiosk and high-throughput satellites. For the other use cases, LDPC and Polar decoders are able to reach the latency target. Turbo decoders can’t reach the latency target. But by applying the early stop, the latency could potentially be improved significantly.

Power consumption and related KPI such as energy efficiency and power density are the biggest challenge for all the three code classes. None of the SoA decoders of the three code families are able to meet the power density requirements in any of the use cases. Regarding energy efficiency, LDPC decoders show the best energy efficiency. The LDPC-BC architecture achieves 0.5 pJ/bit at 7 nm, which meets the requirement of many use cases at the cost of limited flexibility. Polar decoders show similar results, from which the SC-U architecture achieves 1.6 pJ/bit at 7 nm, which is close to the requirement of many use cases. Turbo decoders need improvement by a factor of 10 to meet the requirement of many use cases.

The most promising architecture types of all the three codes are the unrolled and fully paralleled architecture. However, the drawback of this architecture is that it offers no flexibility with respect to code rate and code length. Hence, bringing flexibility to unrolled architectures is a challenge for EPIC project.

Chapter 4 Summary and Conclusion

In this deliverable, a wide range of Tb/s use cases are presented: data kiosk, virtual reality, intra-device communication, wireless fronthaul/backhaul, data centre, hybrid fiber-wireless networks, and high-throughput satellites. These use cases are the result of a thorough search of leading industry standards, business platforms, and emerging applications for the most viable and relevant use cases that target B5G FEC design. Each of the use cases is described in detail, providing a complete set of system level KPI (BER/FER, throughput, latency, power, cost, flexibility) and FEC level KPI (BER/FER, throughput, latency, energy efficiency, area efficiency and power density).

In order to focus our research on the most crucial challenges for Tb/s coding solutions, the following arguments will be considered in order to prioritize our use cases. First, out of the seven use cases, two are significantly simpler from coding performance perspective: given the very short distance of their links, both data kiosk and intra-device communications will easily achieve a high signal-to-noise ratio even at limited transmit power. Hence those two use cases are unlikely to require the most advanced solutions in terms of coding gain. Secondly, the high-throughput satellite use case is very specific and has challenges not only related to channel coding: The huge link distance would not enable to close the link budget while achieving a throughput close to 1 Tb/s, even with strong coding solutions. Moreover, even the raw bandwidth available in satellite bands would not allow for such high throughputs without a lot of spatial multiplexing. Hence while high-performance coding is very relevant in order to save power on satellite links, this use case is not expected to reach a practical throughput close to 1 Tb/s. As conclusion, the four other use cases should be investigated with the highest priority for high-throughput coding solutions: virtual reality, front-haul/back-haul, data centres and hybrid fibre/wireless.

We describe the SoA for turbo codes, LDPC codes and polar codes and provide a comprehensive overview of the SoA FEC decoding in current wireless communication systems from an architecture and implementation perspective. By 2020 and beyond, the digital bulk CMOS technology is expected to scale to the 7 nm node. Therefore, the SoA reference designs have been scaled to 7 nm to take the scaling improvement into account.

After the scaling is applied, an assessment of the gaps with respect to the EPIC use cases is performed. KPI performance gaps are detailed for turbo, LDPC and polar codes. It is important to mention that the gap analysis has not considered the communications performance. This is due to the fact that the communications performance (Shannon bound, maximum likelihood decoding) is not always given in many SoA papers that are implementation-oriented.

However, there are trade-offs between communications performance and high throughput that depends on the particular use case. High communications performance typically requires complex decoding algorithms and large number of iterations to achieve near ML performance and large block lengths to approach Shannon bound. On the other side, 1 Tb/s throughput under the clock constraint of 1 GHz is only feasible by unrolling/functional parallelism. However, unrolling under the area constraint of 10 mm2 becomes only feasible for smaller block lengths and limits the number of decoding iterations. However, this is not a challenge for some use cases where transmit power can be increased to compensate for the weakness of the FEC or the suboptimality of the decoding algorithm, e.g., the data kiosk.

The latency requirements are very diverse and will be challenging for the most demanding use cases. Increasing the number of decoding iterations to improve the communications performance will increase the latency.

The biggest implementational challenge for the three code families is achieving the requirements on energy efficiency and power density while maintaining the necessary flexibility required by the use cases.

The overall conclusion from this analysis is that advances in both code design and decoder implementation, e.g. bringing flexibility in code rate and block length to unrolled decoder architectures, are required to meet the EPIC targets. The learnings from the SoA search and the gap analysis will steer the efforts in the project to bridge the gaps between the SoA and the EPIC goals.

Chapter 5 List of Abbreviations

Abbreviation	Translation
ACS	Add-Compare-Select
AR	Augmented Reality
B5G	Beyond 5G
BBU	Base Band Unit
BCH	Bose-Chaudhuri-Hocquenghem
BCJR	Bah, Cocke, Jelinek and Raviv
BER	Bit Error Rate
BGA	Ball Grid Array
BP	Belief Propagation
BP-D	Belief Propagation Double-column
BP-E	Belief Propagation Early Stopping Criteria
CAPEX	Capital Expenditures
CB	Code Block
CMOS	Complementary Metal Oxide Semiconductor
CN	Check Node
CPRI	Common Public Radio Interface
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
DC	Data Centre
FEC	Forward Error Correction
FFS	Fixed Satellite Service
FPMAP	Fully Parallel MAP
FPS	Frames Per Second
Gb/s	Gigabit/sec
GbE	Gigabit Ethernet
GEO	Geostationary Orbit
GW	Gate Way
HARQ	Hybrid Automatic Repeat Request
HFC BGA	High performance Flip Chip BGA
HS BGA	Heat slug BGA
HTS	High Throughput Satellite
IDC	Intra-Device Communication
KPI	Key Performance Indicator
LDPC	Low Density Parity Check
LDPC-BC	Low Density Parity Check Block Code
LDPC-CC	Low Density Parity Check Convolutional Code

Abbreviation	Translation
Log MAP	Logarithmic MAP
LOS	Line of Sight
LTE	Long Term Evolution
LTE-A	Long Term Evolution Advanced
LTE-A Pro	Long Term Evolution Advanced Pro
MAP	Maximum A Posteriori
Max-Log MAP	Logarithmic MAP with the Maximum approximation
MCS	Modulation and Coding Scheme
MIMO	Multiple Input Multiple Output
NFV	Network Function Virtualization
OPEX	Operating Expenses
PCB	Printed Circuit Board
PE	Processing Element
PMAP	Parallel MAP
PHY	Physical Layer
QPP	Quadratic Permutation Polynomial
RAM	Random Access Memory
RDMA	Remote Direct Memory Access
RLSC	Reduced Latency Soft Cancellation
SC	Successive Cancellation
SCAN	Soft Cancellation
SC-C	Successive Cancellation Combinational
SC-M	Successive Cancellation Multibit
SC-U	Successive Cancellation Unrolled
SCL	Successive Cancellation List
SCL-M	Successive Cancellation List Multibit
SCL-U	Successive Cancellation List Unrolled
SIFS	Short Interframe Space
SoA	State-of-the-Art
SoC	System-on-a-Chip
TB	Transport Block
Tb/s	Terabit/sec
VN	Variable Node
XMAP	Cross-MAP

Chapter 6 Bibliography

Bibliography

[1] M. L. e. al, “An energy efficient 18Gbps LDPC decoding processor for 802.11ad in 28nm CMOS,” in *2015 IEEE Asian Solid-State Circuits Conference (A-SSCC)*, Xiamen, 2015.

[2] “ITRS 2.0, International Technology Roadmap for Semiconductors, 2015 Edition, Section 5: More Moore.,” http://www.semiconductors.org/main/2015_international_technology_roadmap_for_semiconductors_itrs/, 2015.

[3] O. V. e. al, “Scaling the Power Wall: A path to Exascale,” in *International Conference for High Performance Computing, Networking, Storage and Analysis*, Nov. 2014.

[4] “IEEE Std 802.15.3d-2017, IEEE Standard for High Data Rate Wireless Multi-Media Networks, Amendment 2: 100 Gb/s Wireless Switched Point-to-Point Physical Layer,” IEEE, 2017.

[5] “IEEE Std 802.11ad, Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band,” 2012.

[6] M. C. Coşkun, G. Durisi, T. Jerkovits, G. Liva, W. Ryan, B. Stein and F. Steiner, "Efficient error-correcting codes in the short blocklength regime," *Physical Communication,* vol. 34, pp. 66-79, June 2019.

[7] C. Berrou, Y. Saouter, C. Douillard, S. Kerouedan and M. Jezequel, “Designing good permutations for turbo codes: towards a single model,” in *EEE International Conference on Communications, (ICC'04)*, Paris, France, June 2004.

[8] R. Garzón-Bohórquez, C. Abdel Nour and C. Douillard, “Protograph-based interleavers for punctured turbo codes,” *IEEE Transactions on Communications,* vol. 66, no. 5, p. 1833–1844, 2018.

[9] S. Dolinar, D. Divsalar and F. Pollara, “Code Performance as a Function of Block Size,” *TMO Progress Report 42-133,* pp. 1-23, 15 May 1998.

[10] G. D. Forney, "Burst-Correcting Codes for the Classic Bursty Channel," *IEEE Transactions on Communication Technology,* vol. 19, no. 5, pp. 772 - 781, Oct. 1971.

[11] G. Liva, L. Gaudio, T. Ninacs and T. Jerkovits, “Code design for short blocks: A survey,” *arXiv preprint arXiv:1610.00873,* October 2016.

[12] ITU-R, “Technical and operational characteristics of the land mobile service applications operating in the frequency range 275-450 GHz,” 2017.

[13] A. Fricke, “TG3d Channel Modelling Document,” IEEE P802.15 working group for Wireless

Personal Area Networks, 2016.

[14] G. Fettweis, F. Guderian and S. Krone, "Entering the Path Towards Terabit/s Wireless Links," in *Design, Automation & Test in Europe (DATE) Conference & Exhibition*, Grenoble, France, Mar. 2011.

[15] CNBC, "https://www.cnbc.com/2017/11/02/how-many-iphones-did-apple-sell-in-q4-2017.html," [Online].

[16] P. Warden, "https://petewarden.com/2015/10/08/smartphone-energy-consumption/," [Online].

[17] K. Schwab, "https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/," World Economic Forum, 2016.

[18] T. Braud, F. Hassani Bijarbooneh, D. Chatzopoulos and P. Hui, "Future Networking Challenges: The Case of Mobile Augmented Reality," in *IEEE International Conference on Distributed Computing Systems*, Atlanta, GA, USA, June 2017.

[19] R. Furlan, "The future of augmented reality: Hololens-Microsoft's AR heaset shines despite rough edges," *IEEE Spectrum,* vol. 53, no. 6, p. 21, 2016.

[20] B. Iribe, "Virtual Reality - a new frontier in computing," Oculus VR, 2013.

[21] S. M. LaValle, A. Yershova, M. Katsev and M. Antonov, "Head tracking for the Oculus Rift," *ICRA,* pp. 187-194, 2014.

[22] M. C. Potter, B. Wyble, C. E. Hagmann and E. S. McCourt, "Detecting meaning in RSVP at 13 ms per picture," *Attention, Perception, and Psychophysics,* vol. 76, no. 2, pp. 270-279, 2014.

[23] R. S. Allison, L. R. Harris, M. Jenkin, U. Jasiobedzka and J. E. Zacher, "Tolerance of termporal delay in virtual environments," in *IEEE Virtual Reality*, Yokohama, Japan, 2001.

[24] E. Bastug, M. Bennis, M. Medard and M. Debbah, "Towards interconnected virtual reality: opportunities, challenges and enablers," *IEEE Commun. Mag.,* vol. 52, no. 6, pp. 110 - 117, June 2017.

[25] M. Pirenne, "Vision and the Eye," *Chapman & Hall,* vol. 47, 1967.

[26] A. A. A. S. O. I. Y. Aderemi, "Modeling, Simulation and Analysis of Video Streaming Errors in Wireless Wideband," Springer, 2013, pp. 15-28.

[27] "https://www.windowscentral.com/microsoft-hololens-processor-storage-and-ram," [Online].

[28] "IEEE Std 802.15.3c-2009, Amendment 2: Millimeter-wave-based Alternative Physical layer Extension," IEEE.

[29] S. Microsystems, "Sun Fire E25K/E20K Systems Overview," 2006. [Online]. Available: https://docs.oracle.com/cd/E19065-01/servers.e25k/817-4136-13/4_Interconnect.html#56694.

[30] O. A. T. Yilmaz, “On the 5G Wireless Communications at the Low Terahertz Band,” 30 November 2016.

[31] “P802.11ay/D1.1, Draft Amendment 7: Enhanced throughput for operation in license-exempt bands above 45 GHz,” IEEE, 2018.

[32] Bjørnstad, “Handling Delay in 5G Ethernet Mobile Fronthaul Networks", *EuCNC 2018*

[33] ETSI GR mWT 012, "*5G Wireless Backhaul/X-Haul*", , V1.1.1 (2018-11).

[34] Cisco, “Data Center Architecture Overview,” https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Data_Center/DC_Infra2_5/DCInfra_1.pdf, 2016.

[35] Y. Cui, H. Wang, X. Cheng and B. Chen, “Wireless Data Center Networking,” *IEEE Wireless Communications,* vol. 8, no. 6, Dec. 2011.

[36] T. Kurner, “TG3d Applications Requirements Document (ARD),” 2015.

[37] Cavium, “Introduction to Ethernet Latency,” http://www.qlogic.com/Resources/Documents/TechnologyBriefs/Adapters/Tech_Brief_Introduction_to_Ethernet_Latency.pdf , 2017.

[38] Dolphin Communications, “300ns Latency Across PCI Express,” http://www.prnewswire.com/news-releases/dolphin-demonstrates-300ns-latency-across-pci-express-network-at-idf-300130148.html , 2015.

[39] “Average Power Use Per Server,” http://www.vertatique.com/average-power-use-server, 2015.

[40] Q. J. Gu, “THz interconnect: the last centimeter communication,” 2015.

[41] ITU-T, “G.709: Interfaces for the optical transport network,” http://www.itu.int/rec/T-REC-G.709-201606-I/en, 2016.

[42] Gigalight, “QSFP28 PSM4 Optical Transceiver,” http://www.gigalight.com/products_detail/productId=241.html, 2018.

[43] J. Amos, *Viasat broadband 'super-satellite' launches,* BBC News, 2011.

[44] H. Caleb, *Dankberg: ViaSat 3 Satellite Will Have More Capacity than the Rest of the World Combined,* 2016.

[45] “SaT5G Project,” [Online]. Available: sat5g-project.eu.

[46] “Satellite communication technologies, Research & Innovation: European Commission,” 27 October 2017. [Online]. Available: ec.europa.eu/research/participants/portal/desktop/en/opportunities/h2020/topics/space-15-tec-2018.html.

[47] G. Maral and M. Bousquet, Satellite Communication Systems, 5 ed., John Wiley & Sons Ltd, 2009.

[48] E. E. 3. 3.-2. v1.1.1, *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 2: DVB-S2 Extensions (DVB-S2X).*

[49] P. A. Jorn Christensen, *ITU Regulations for Ka-band Satellite Networks.*

[50] ViaSat, *ViaSat Unveils First Global Broadband Communications Platform to Deliver Affordable, High-Speed Internet Connectivity and Video Streaming to All,* 2016.

[51] M. Freeman, *ViaSat Is Dramatically Expanding Satellite Broadband,* The SanDiego Union-Tribune, 2017.

[52] European Space Agency (ESA), *W-Band: The Next Frontier for Satcoms,* 2015.

[53] S. D. Fina, M. Ruggieri and A. V. Bosisio, "Exploitation of the W-band for high capacity satellite communications," *IEEE Transactions on Aerospace and Electronic Systems,* January 2003.

[54] A. D. Luise, A. Paraboni and M. Ruggieri, "Satellite communications in W-band: experimental set-up for channel characterization," in *2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No.04TH8720)*, 2004, p. 476.

[55] M. R. Patel, Spacecraft Power Systems, CRC Press, 2005.

[56] Euroconsult, *Satellite Manufacturing & Launch.*

[57] Satellite Industry Association (SIA), *Satellites and export credit financing fact sheet,* 2014.

[58] "Multi-projectwafer service, Wikipedia," 4 February 2018. [Online]. Available: en.wikipedia.org/wiki/Multi-project_wafer_service.

[59] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Transactions on communications,* vol. 44, no. 10, pp. 1261-1271, 1996.

[60] L. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transaction on Information Theory,* Vols. IT-20, no. 2, p. 284–287, 1974.

[61] P. Robertson, E. Villebrun and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *IEEE International Conference on Communications, ICC '95*, Seattle, USA, 1995.

[62] 3GPP, *TS 36 300 Rel-9; LTE; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description,* July 2010.

[63] 3GPP, *TS 36 300 Rel-12; LTE; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description,* October 2015.

[64] 3GPP, *TS 36 300 Rel-13; LTE; Evolved Universal Terrestrial Radio Access (E-UTRA) and*

Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description, January 2016.

[65] A. Nimbalker, Y. Blankenship, B. Classon and T. K. Blankenship, “ARP and QPP interleavers for LTE turbo coding,” in *IEEE Wireless Communications and Networking Conference WCNC 2008*, Las Vegas, USA, 2008.

[66] J.-M. Hsu and C.-L. Wang, “A parallel decoding scheme for turbo codes,” in *1998 IEEE International Symposium on on Circuits and Systems, ISCAS'98*, Monterey, CA, USA, 1998.

[67] S. Belfanti, C. Roth, M. Gautschi, C. Benkeser and Q. Huang, “A 1Gbps LTE-advanced turbo-decoder ASIC in 65nm CMOS,” in *Symposium on VLSI Circuits (VLSIC)*, Kyoto, Japan, 2013.

[68] C. Roth, S. Belfanti, C. Benkeser and Q. Huang, “Efficient parallel turbo-decoding for high-throughput wireless systems,” *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 61, no. 6, p. 1824–1835, 2014.

[69] T. Ilnseher, Kienle K., Weis C. and N. Wehn, “A 2.15 Gbit/s turbo code decoder for LTE advanced base station applications,” in *7th International Symposium on Turbo Codes and Iterative Information Processing (ISTC 2012)*, Gothenburg, Sweden, 2012.

[70] R. Shrestha and R. P. Paily, “[High-throughput turbo decoder with parallel architecture for LTE wireless communication standards,” *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 61, no. 9, p. 2699–2710, 2014.

[71] Y. Sun and J. R. Cavallaro, , “Efficient hardware implementation of a highly-parallel 3GPP LTE/LTE-advance turbo decoder,” *Integration VLSI Journal,* vol. 44, no. 4, pp. 305-315, 2010.

[72] C. C. Wong and H. C. Chang, “High-efficiency processing schedule for parallel turbo decoders using QPP interleaver,” *IEEE Transactions Circuits Systems I: Regular Papers,* vol. 58, no. 6, p. 1412–1420, 2011.

[73] A. Worm, H. Lamm and N. Wehn, “VLSI architectures for high-speed MAP decoders,” in *14th International Conference on VLSI Design*, Bangalore, India, 2001.

[74] M. May, “Architectures for High-throughput and Reliable Iterative Channel Decoders, PhD thesis,” Department of Electrical Engineering and Information Technology, University of Kaiserslautern, May 2013.

[75] A. Worm, H. Michel, F. Gilbert, G. Kreiselmaier, M. J. Thul and N. Wehn, “Advanced implementation issues of turbo-decoders,” in *2nd International Symposium on Turbo Codes & Related Topics*, Brest, France, 2000.

[76] M. M. Mansour and N. R. Shanbhag, “VLSI architectures for SISO-APP decoders,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 11, no. 4, p. 627–650, 2003.

[77] A. Worm, “Implementation issues of turbo-decoders, PhD thesis, ISBN 3-925178-72-4,” University of Kaiserlautern, 2001.

[78] 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, p. 1376–1389, 2014.

[79] S. Weithoffer, "Implementation issues of flexible high-throughput turbo-code decoders for high code rates, PhD Thesis," University of Kaiserslautern, 2018.

[80] S. Weithoffer, K. Kraft and N. Wehn, "Bit-level pipelining for highly parallel turbo-code decoders: a critical assessment," in *IEEE Africon 2017*, Cape Town, South Africa, 2017.

[81] E. Boutillon, J. Sanchez-Rojas and C. Marchand, "Compression of redundancy free trellis stages in turbo-decoder," *Electronics Letters,* vol. 49, no. 7, p. 460–462, 2013.

[82] S. Weithoffer, F. Pohl and N. Wehn, "On the applicability of trellis compression to turbo-code decoder hardware architectures," in *9th International Symposium on Turbo Codes and Iterative Information Processing (ISTC 2016)*, Brest, france, 2016.

[83] J. Zhang and M. P. Fossorier, "Shuffled iterative decoding," *IEEE Transactions on Communications,* vol. 53, no. 2, p. 209–213, 2005.

[84] R. G. Maunder, "A fully-parallel turbo decoding algorithm," *IEEE Transactions on Communications,,* vol. 63, no. 8, p. 2762–2775, 2015.

[85] 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, p. 323–346, 2016.

[86] P. Schläfer, C. Weis, N. Wehn and M. Alles, "Design Space of Flexible Multigigabit LDPC Decoders," VLSI Design, 2012.

[87] Z. Chen, X. Peng, X. Zhao, Q. Xie, L. Okamura, D. Zhou and S. Goto, "A macro-layer level fully parallel layered LDPC decoder SOC for IEEE 802.15.3c application," Proceedings of 2011 International Symposium on VLSI Design, Automation and Test, 2011.

[88] M. Li, J. W. Weijers, V. Derudder, I. Vos, M. Rykunov, S. Dupont, P. Debacker, A. Dewilde, Y. Huang, L. V. d. Perre and W. V. Thillo, "An energy efficient 18Gbps LDPC decoding processor for 802.11ad in 28nm CMOS," 2015 IEEE Asian Solid-State Circuits Conference (A-SSCC), 2015.

[89] M. Korb and T. G. Noll, "Area- and energy-efficient high-throughput LDPC decoders with low block latency," Proceedings of the ESSCIRC (ESSCIRC), 2011.

[90] T. Mohsenin, D. N. Truong and B. M. Baas, "A Low-Complexity Message-Passing Algorithm for Reduced Routing Congestion in LDPC Decoders," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 57, no. 5, pp. 1048–1061, 2010.

[91] S. Scholl, S. Weithoffer and N. Wehn, "Advanced iterative channel coding schemes: When Shannon meets Moore," 9th International Symposium on Turbo Codes and Iterative Information Processing (ISTC), 2016.

[92] R. Ghanaatian, A. Balatsoukas-Stimming, T. C. Müller, M. Meidlinger, G. Matz, A. Teman and A. Burg, "A 588-Gb/s LDPC Decoder Based on Finite-Alphabet Message Passing," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 26, no. 2, pp. 329–340, 2018.

[93] M. Lentmaier, M. M. Prenda and G. P. Fettweis, “Efficient message passing scheduling for terminated LDPC convolutional codes,” IEEE International Symposium on Information Theory Proceedings, 2011.

[94] A. J. Felstrom and K. S. Zigangirov, “Time-varying periodic convolutional codes with low-density parity-check matrix,” IEEE Transactions on Information Theory, vol. 45, no. 6, pp. 2181–2191, 1999.

[95] M. Papaleo, A. R. Iyengar, P. H. Siegel, J. K. Wolf and G. E. Corazza, “Windowed erasure decoding of LDPC Convolutional Codes,” IEEE Information Theory Workshop on Information Theory, 2010.

[96] Y. Chen, Q. Zhang, D. Wu, C. Zhou and X. Zeng, “An Efficient Multirate LDPC-CC Decoder With a Layered Decoding Algorithm for the IEEE 1901 Standard,” IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 61, no. 12, pp. 992–996, 2014.

[97] I. Yoo and I. C. Park, “Low-Power LDPC-CC Decoding Architecture Based on the Integration of Memory Banks,” IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 64, no. 9, pp. 1057-1061, 2017.

[98] C. L. Chen, Y. H. Lin, H. C. Chang and C. Y. Lee, “A 2.37-Gb/s 284.8 mW Rate-Compatible (491,3,6) LDPC-CC Decoder,” IEEE Journal of Solid-State Circuits, vol. 47, no. 4, pp. 817–831, 2012.

[99] C. L. Lin, R. J. Liu, C. L. Chen, H. C. Chang and C. Y. Lee, “A 7.72 Gb/s LDPC-CC decoder with overlapped architecture for pre-5G wireless communications,” IEEE Asian Solid-State Circuits Conference (A-SSCC), 2016.

[100] E. Arıkan, “Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels,” *IEEE Transactions on Information Theory,* vol. 55, no. 7, pp. 3051-3073, 2009.

[101] I. Tal and A. Vardy, “List Decoding of Polar Codes,” *IEEE Transactions on Information Theory,* vol. 61, no. 5, pp. 2213-2226, 2015.

[102] D. E. Muller, “Application of Boolean Algebra to Switching Circuit Design and to Error Correction,” *IRE Trans. Electron. Computers,* vol. EC, no. 3, pp. 6-12, 1954.

[103] I. S. Reed, “A Class of Multiple-Error-Correcting Codes and the Decoding Scheme,” *Transactions of the IRE Professional Group on Information Theory,* vol. 4, no. 4, pp. 38-49, 1954.

[104] G. D. Forney, “Codes on Graphs,” *IEEE Transactions on Information Theory,* vol. 47, no. 2, pp. 520-548, 2001.

[105] 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.

[106] 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.

[107] P. Giard, A. Balatsoukas-Stimming, T. C. Müller, A. Burg, C. Thibeault and W. J. Gross, "A Multi-Gbps Unrolled Hardware List Decoder for a Systematic Polar Code," in *50th Asilomar Conference on Signlas Systems and Computers*, Pacific Grove, CA, 2016.

[108] 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.

[109] Y. S. Park, Y. Tao, S. Sun and Z. Zhang, "A 4.68Gb/s Belief Propagation Polar Decoder With Bit-Splitting Register File," in *Symposium on VLSI Circuits Digest of Technical Papers*, Honolulu, HI, 2014.

[110] B. Yuan and K. K. Parhi, "Early Stopping Criteria for Energy-Efficient Low-Latency Belief-Propagation Polar Code Decoders," *IEEE Transactions on Signal Processing,* vol. 62, no. 24, pp. 6496-6506, 2014.

[111] J. Lin, Z. Yan and Z. Wang, "Efficient Soft Cancelation Decoder Architectures for Polar Codes," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 25, no. 1, pp. 87-99, 2017.