

Received March 5, 2019, accepted March 18, 2019, date of current version April 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2907156

# Data Link Layer Processor for 100 Gbps Terahertz Wireless Communications in 28 nm CMOS Technology

LUKASZ LOPACINSKI<sup>1</sup>, MIROSLAV MARINKOVIC<sup>2</sup>, GORAN PANIC<sup>1</sup>,  
MOHAMED HUSSEIN EISSA<sup>1</sup>, ALIREZA HASANI<sup>3</sup>,  
KARTHIK KRISHNEGOWDA<sup>3</sup>, AND ROLF KRAEMER<sup>1,3</sup>

<sup>1</sup>IHP -Leibniz-Institut für innovative Mikroelektronik, 15236 Frankfurt (Oder), Germany

<sup>2</sup>Arquimea Deutschland GmbH, 15236 Frankfurt (Oder), Germany

<sup>3</sup>Brandenburg University of Technology Cottbus–Senftenberg, 03046 Cottbus, Germany

Corresponding author: Lukasz Lopacinski (lopacinski@ihp-microelectronics)

This work was supported in part by the German Research Foundation (DFG) Project End2End100-Phase-2 (SPP1655) under Grant 318642660, and in part by the German Federal Ministry of Education and Research (BMBF) Project Fast-Spot under Grant 03ZZ0512A.

**ABSTRACT** In this paper, we show our 165 Gbps data link layer processor for wireless communication in the terahertz band. The design utilizes interleaved Reed-Solomon codes with dedicated link adaptation, fragmentation, aggregation, and hybrid-automatic-repeat-request. The main advantage is the low-chip area required to fabricate the processor, which is at least two times lower than the area of low-density parity-check decoders. Surprisingly, our solution loses only  $\sim 1$  dB gain when compared to high-speed low-density parity-check decoders. Moreover, with only 2.38 pJ/bit of energy consumption at 0.8 V, one of the best results in the class of comparable implementations has been achieved. Alongside, we show our vision of a complete 100 Gbps wireless transceiver, including radio frequency frontend and baseband processing. For the baseband realization, we propose a parallel sequence spread spectrum and channel combining at the baseband level. Challenges to high-speed wireless transmission at the terahertz band are addressed as well. To the authors' best knowledge, it is one of the first data link layer implementations that deal with a data rate of  $\geq 100$  Gbps.

**INDEX TERMS** 100 Gbps wireless, terahertz communication, Reed-Solomon coding, data link layer.

## I. INTRODUCTION

Although the sub-terahertz band of 200-300 GHz allows to allocate channel bandwidth of several gigahertz and supports a data rate of 100 Gbps, the wide bandwidth and high data rate require demanding processing. All analog components have to support high gain and linearity over a wide spectrum at ultra-high frequencies. The digital parts, however, have to deal with data rates of 100 Gbps and bit processing time  $< 10$  ps. Thus, we face many difficulties on each design step of such a transceiver. Moreover, we need to keep in mind that wireless communication is used in battery powered devices and has to operate at strictly limited energy limits. This article is mainly focused on the data link layer (DLL) processing, but additionally, we introduce possible fronted

The associate editor coordinating the review of this manuscript and approving it for publication was Khursheed Aurangzeb.

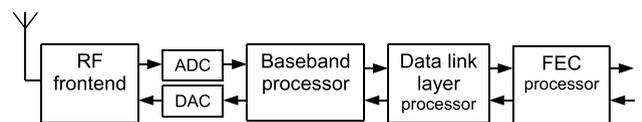


FIGURE 1. Discussed wireless radio-transceiver system.

and baseband implementations. Fig. 1 depicts the architecture of the discussed wireless radio-transceiver.

## A. CHALLENGES TO HIGH-SPEED WIRELESS COMMUNICATIONS

Ultra-high speed wireless systems require either very high bandwidth or very high bandwidth efficiency. In cellular architectures like LTE or 5G high bandwidth efficiency is in the focus. This is due to the limited bandwidth in the available radio bands. Increasing the bandwidth efficiency requires a corresponding increase in signal processing power

that increases the complexity of the baseband processor dramatically. In THz bands, bandwidth limitation is no issue such that these bands are today considered for ultra-high speed communications. If we use 25 GHz bandwidth, at bandwidth efficiency of 4 b/s/Hz, is sufficient enabling less complex baseband processing. THz-channels are known to be highly attenuating and require high-gain-antennas [1]–[4] and highly efficient amplifiers. However, manufacturing the amplifiers and antennas is challenging. At such high frequencies, it is impossible to connect the antenna using wire bonding due to reflections, cross-talks, and attenuation on the bonding wires [5], [6]. Therefore, the antenna has to be integrated into the RF-frontend [6], [7]. This leads, however, to interferences with metal layers of the ASIC and gain reduction. Further problems arise in the design of signal processing within the baseband (BB). For such fast links, typical digital-design is inefficient, because digital technology consumes too much power and silicon area [2], [8]. Thus, the algorithms have to be simplified and do not work as effectively as for slower communication systems. Currently, all problems in the RF-frontend and BB design are shifted to higher layers. In such a case, FEC and data link layer (DLL) must tackle these problems. FEC and DLL are expected to repair channel impairments, and additionally errors caused by lower layers. Thus, the design of FEC and DLL for the targeted system becomes complex, requires a large chip area, high power, and cooling ability [9], [10]. More details on the difficulties of 100 Gbps data link layer design can be found in [11].

## B. POWER CONSUMPTION

A mobile-transceiver for the targeted application has to consume less than 1W or equivalently  $\sim 10$  pJ/bit in the case of 100 Gbps data-rate. This limit includes the whole RF-frontend, BB, and DLL processing. For example, RF-frontend of a 240 GHz system with an output power of  $-4.4$  dBm consumes 1.2 W [4]. This allows to establish a PSK-modulated link of  $\sim 23$  Gbps at a distance of 15 cm [4]. Although the output power, data rate, and distance are smaller than targeted, the RF-frontend alone exceeds the assumed power for the whole transceiver (RF+ADC/DAC+BB+DLL+FEC). Apparently, increasing the data rate and range will increase the consumed power significantly.

The high-speed FEC decoder presented in [12]–[14] is another example of the challenges encountered in the design of 100 Gbps transceivers. Even if the implementation exploits all known techniques to improve LDPC-decoding efficiency, it still needs  $12 \text{ mm}^2$  of silicon and consumes 5 W. Due to the consumed power and low flexibility due to fixed code rate, today's solutions have to be revised on an algorithmic level. Although it is possible to reduce the power of the FEC processor down to  $\sim 600$  mW by applying ultra-high scaled 7 nm technology [15]–[17], it is still far beyond the targeted limit of 1 W for the complete transceiver. This becomes even more challenging when code-rates lower than 13/16 and data rates  $> 100$  Gbps are considered. Lower code rates

support higher gain but require much more computations. Thus, the DC-power will be significantly higher than the estimated 1 W for the complete transceiver.

## II. RELATED WORK

In this section, we first refer to two RF frontend implementations that successfully demonstrate THz and sub-THz communication. As our processor is equipped with compatible interfaces, either of them can be used in our design. Afterward, other analog components required by the data link layer (DLL) processor are also introduced.

### A. 300 GHz RF FRONTEND

The RF frontend proposed by [18], operates at 300 GHz and is able to transfer up to 64 Gbps using QPSK modulation on a distance of 1 m. The chipset is expected to operate at higher data rates or longer distances as well, but the setup is limited by practical constraints of employed instruments [18]. The design uses horn antennas with 24.2 dBi gain and the chip is realized in 35 nm GaAs mHEMT technology [19] with  $f_T$  and  $f_{\max}$  of more than 500 and 1000 GHz, respectively [18], [19].

### B. 240 GHz RF FRONTEND

The other RF-frontend design, of our focus, has been proposed by [20]. The chip is fabricated in IHP 130 nm SiGe BiCMOS technology with  $f_T$  and  $f_{\max}$  of 300 and 500 GHz, respectively [21] and operates at 240 GHz. The recently published revision [20] occupies RF-RX bandwidth of 55 GHz, RF-TX bandwidth of 35 GHz, and supports a data rate of  $\sim 25$  Gbps with BER  $\approx 2e-4$  on a distance of 15-30 cm.

The transceiver uses a double-folded dipole antenna combined with  $40 \text{ mm} \times 40 \text{ mm}$  plastic lens (polyethylene). Such an antenna set provides 14 dBi of combined gain, while the transmitter alone delivers  $-0.8$  dBm of output power [20].

### C. ADC AND DAC UNITS

The next step after RF-fronted up- and down-conversion are ADCs and DACs. The design of digital and analog converters for data rates approaching 100 Gbps is difficult as well. Therefore, at this level, we apply one of two proposed improvements. Instead of processing the whole bandwidth in a single AD or DA converter, we split and merge the signal at the baseband level in the analog domain, before the AD and DA conversions. For this purpose “parallel sequence spread spectrum (PSSS)” and “channel combining” can be employed. We explain both methods in the next two sub-sections.

### D. CHANNEL SPLITTING AND COMBINING

The channel combining circuits are described in [22], with Fig. 2 depicting the basic idea of the operation. The baseband channel splitter divides the analog baseband signal into parallel streams. Each stream can be processed by an individual baseband core, and thus by a separate ADC. Thus the bandwidth and data rate are also divided between streams,

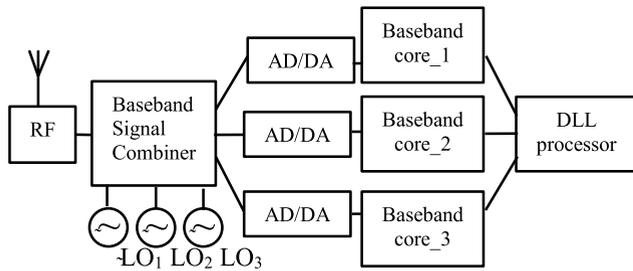


FIGURE 2. Baseband signal combining and splitting.

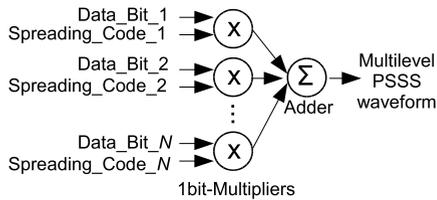


FIGURE 3. Parallel sequence spread spectrum (PSSS) encoder.

and therefore the demands for ADC, DAC, and baseband are significantly reduced. We have incorporated channel splitters and combiners with five and three outputs and inputs into our proposed design.

The combiner and splitters are realized as a set of analog mixers (Fig. 2) that are fed with different local oscillator frequencies, e.g., 3.75 GHz, 7.5 GHz, and 15 GHz [22]. The chip is fabricated in IHP 130 nm SiGe BiCMOS technology and can be directly integrated with the previously mentioned 240 GHz frontend. The drawback of the latest 3-input combining chip is limited baseband bandwidth of ‘only’ 6.5 GHz [22]. This problem is partially resolved in the next 5-input release, but still, we will need more bandwidth. Therefore we also work on a parallel sequence spread spectrum (PSSS) that is described in the next subsection.

### E. PARALLEL SEQUENCE SPREAD SPECTRUM (PSSS)

The PSSS has been proposed as a spreading technique for different communications systems [24]–[26]. Figure 3 depicts a simplified diagram of a PSSS-encoder. The input data bits are multiplied by direct sequence spread spectrum (DSSS) sequences (e.g., Barker codes or m-sequences), and then added together in the time domain. After that, a multilevel amplitude waveform is produced, which carries  $N$  bits in  $N$  multilevel-chips (multilevel-symbols). Thus, the rate of PSSS modulated data is unchanged. The spreading is performed in the frequency domain, but not in the time domain. In our case, the frequency-spreading ability of the PSSS is rather a drawback than an advantage, because spreading the 100 Gbps signals in the frequency domain leads to ridiculously large bandwidth. The very large PSSS bandwidth is anyway cut out by the RF-frontend and the PSSS modulator itself. The circuits have limited bandwidth, which is usually much lower than the bandwidth of the resulting 100 Gbps PSSS signal. Instead of the spreading, we concentrate on the analog

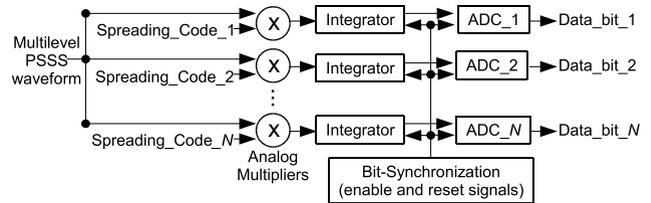


FIGURE 4. Parallel sequence spread spectrum (PSSS) decoder [23].

implementation of the PSSS-based receiver, which has two significant advantages. Firstly, most of the PSSS-receiver can be implemented in the analog domain (Fig. 4), which consumes less power, less chip area, and works much faster than in the digital domain. Secondly, we sample the baseband signal in parallel with  $N$  ADCs (Fig. 4). Thus, the sampling clock is reduced by  $N$ -times, where the  $N$  is the spreading sequence length ( $N = 15$ , according to the work published in [8], [23], [27]). These two advantages of the PSSS allow us to cross the 100 Gbps data rate barrier on the baseband level.

## III. WORK DETAILS AND RESULTS

In the previous parts, we identified the main challenges of high-speed communication and described two RF-frontends and two baseband techniques that can be used for communication in the THz band. To build a complete transceiver, we additionally need a data link layer (DLL) processor with FEC. The design of these elements is explained in this section.

### A. SUPPORTED FUNCTIONALITY

The implemented DLL processor supports three essential functionalities. Firstly, it aggregates the data to 64 kB frames and divides them into 1 kB fragments. Each 1kB-frame-fragment is protected by an individual cyclic redundancy check (CRC) code. Secondly, it uses interleaved Reed-Solomon (RS) FEC codes [28] and supports hybrid automatic repeat request-I (HARQ-I) scheme with selective fragment retransmissions [9]. Thirdly, it reduces the overhead of HARQ-I by a dedicated link adaptation algorithm [29] and an acknowledge compression scheme [28].

### B. SELECTION OF FEC ALGORITHM

The FEC method used for 100 Gbps applications has to be selected very carefully to avoid hardware and power overhead. Tab. 1 compares some selected hard- and soft-decision decoders in terms of hardware complexity. All algorithms are individually parametrized to typical configurations, thus each implementation has a different code rate and shows different error correction performance. At this step, we shortly introduce the hardware complexity, and the correction performance is discussed later in this section. All implementations are tested in a Kintex7 FPGA, keeping in mind that the resources needed for FPGA implementation are correlated with the ASIC area and hardware complexity. As reported in Tab. 1, the largest RS decoder achieves 2.6 times higher normalized throughput than the 1/2-rate

TABLE 1. Resources comparison of selected FEC algorithms.

IMPLEMENTATION	REF.	Normalized THROUGHPUT [kbps/LUT] (ASCENDING ORDER)	THROUGHPUT [Mbps]	FFs	LUTs	MEMORY [kbits]	CODE RATE	$E_b N_0$ AT BER=1E-5 [dB], BPSK, AWGN
TURBO DECODER 3GPP, BLOCK SIZE 5114 5-BIT QUANTIZATION	[34]	12.6	47.8	4081	3785	342	0.333	~1.2
LDPC(16200, 14400 ) 6-BIT QUANTIZATION	[35]	14.2	833	60535	58694	4482	0.889	~6.3
LDPC(10368, 8448) 6-BIT QUANTIZATION	[35]	20.4	1131	57791	55530	4284	0.815	~6
VITERBI DECODER (CONSTR. LEN. 7, TRACEBACK 96) 3-BIT QUANTIZATION	[36]	135.6	286	1719	2113	72	0.5	~4.7
IHP RS(255,223) DECODER 1-BIT QUANTIZATION	[37], [38]	358.8	1993	3039	5554	8	0.874	~6.2
XIL. RS(255,223) DECODER 1-BIT QUANTIZATION	[39]	497.9	730	1327	1466	54	0.874	~6.2
XIL. RS(255,239) G.709 DECODER 1-BIT QUANTIZATION	[39]	2936.9	2376	801	809	36	0.937	~6.8

Viterbi decoder at the cost of 1.5 dB loss, and 17.6 times higher normalized throughput than the LDPC(10368, 8448) code at the cost of 0.2 dB loss. The overall performance of the FPGA-implemented LDPC decoders at the selected code rate is poor. Both implementations require large resources, provide relatively low correction performance and decoding throughput. Later, we compare our RS ASIC implementation to fully-parallel, fully-unrolled ASIC LDPC decoders [12], which achieve higher decoding and correction performance, but due to high resources demand, they are not targeted for FPGA designs.

The selected turbo decoder provides the highest gain at the lowest code rate from the selected decoders, but it requires 28 times more resources than the largest RS. Moreover, it is proven in [13], [14] that turbo decoders have internal decoding dependencies and design of high-speed parallel implementation in hardware is difficult.

The results shown in Tab. 1 can be also interpreted as follows. Considering hardware implementation in a Xilinx Virtex7 VX690T FPGA, which has 433200 LUTs, we require more than 18 development boards to support turbo codes at the decoding throughput of 100 Gbps, assuming 100% chip utilization that cannot be achieved in reality. For LDPC we need more than 11 boards, for Viterbi more than 2 boards, while RS needs only 1 development kit and this has been already proven by us in [30].

We need to keep in mind that Tab. 1 compares hard-decision RS with selected soft-decision algorithms, which are suited to operate on significantly lower code rate (e.g., 1/2, 1/3). Thus, the comparison may lead to false conclusions. The 8-bit RS codes are suited for low overhead, and code rates below 0.874 are used rarely

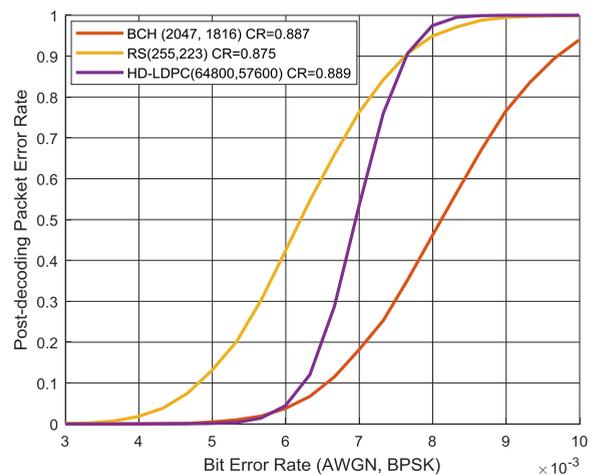


FIGURE 5. Comparison of hard-decision LDPC, RS, BCH decoders at code rate (CR) ≈ 0.883, BPSK modulation, and AWGN channel.

(more in section V.J). In our application, however, we target high-speed communication ( $\geq 100$  Gbps) with low power demand, and therefore the 1-bit quantization and low redundancy overhead are demanded. For other applications, a 1/2-rate LDPC decode will be a better choice probably. Especially, when soft decision decoding, low code rates, and high gain are desired.

To give a better overview of the advantages of RS codes, we additionally compare LDPC, BCH, and RS codes at similar code rates with 1-bit quantized bit input (Fig. 5). In such a case, the decoding conditions for all algorithms are normalized. At packet error rate (PER) equal to 0.5 (AWGN, BPSK), LDPC(64800,57600) code of DVBT-S2 implementation [31], [32] operates at ~ 12% higher

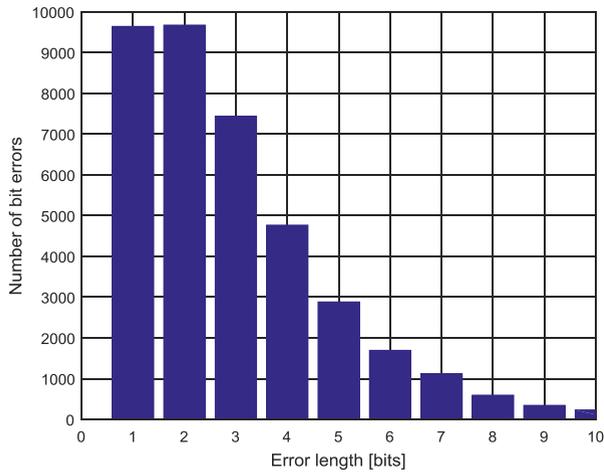


FIGURE 6. Markov chain generated error characteristic used to generate short-burst bit errors.

BER than the RS. The LDPC decoder uses up to 50 decoding iterations and is based on a powerful sum-product algorithm (SPA) with floating-point arithmetic [33]. The tested LDPC algorithm works on binary quantized input data, like RS and BCH, but the internal decoding stages are represented by floating-point variables and are performed by SPA (please do not confuse it with bit-flipping). Such algorithms are usually used for software realizations only, and for hardware, the min-sum approximation with fixed-point logic is commonly employed [12]. Additionally, the number of decoding iterations is significantly lowered, thus the presented DVBT-S2 decoder realized in software shows very good correction performance.

The loss to the BCH decoder that operates on block length very similar to the RS is higher, and the BCH corrects up to 25% more bit errors. This situation changes when the AWGN channel is replaced with an error characteristic that generates single and short-burst errors. To demonstrate this, we prepared a Markov chain BER generator, which produces an error characteristic as shown in Fig. 6. In such conditions, the RS, HD-LDPC, and BCH algorithms achieve results as shown in Fig. 7. The RS decoder shows better performance than the BCH as well as the complex SPA-LDPC and this is the main advantage of RS codes. The codes, in general, are very efficient against burst errors. Although the coding gain for the AWGN channel and the operable code rate is very limited, they have low complexity and achieve high decoding throughput in hardware and software [10]. In the remaining parts of this paper, we use RS codes as a base of our data link layer processor and prove that this lightweight FEC can be used for low-power, high-speed data link layer.

### C. INTERLEAVED RS CODES

Although the selected hard decision interleaved RS codes have limited correction performance, we favor RS over LDPC due to two reasons. Firstly, the RS requires very low resources to support high-speed decoding. Secondly, the PSSS baseband processor delivers only binary-quantized bits and cannot

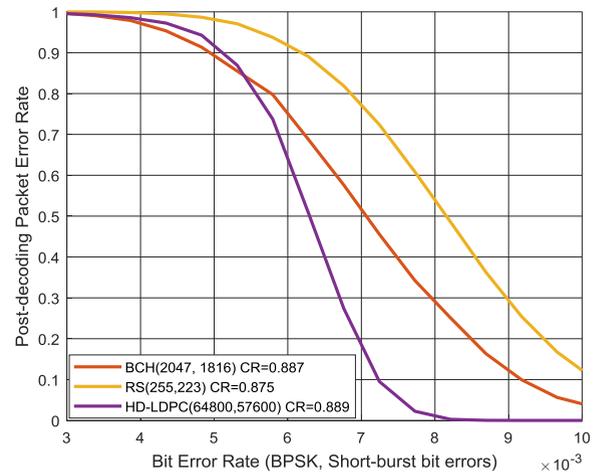


FIGURE 7. Comparison of LDPC, RS, BCH decoders at code rate (CR) almost equal to 0.883. BPSK modulation and short-burst errors are considered.

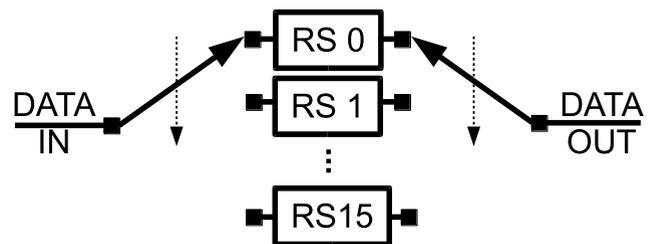


FIGURE 8. Implemented interleaved RS codes for FEC coding.

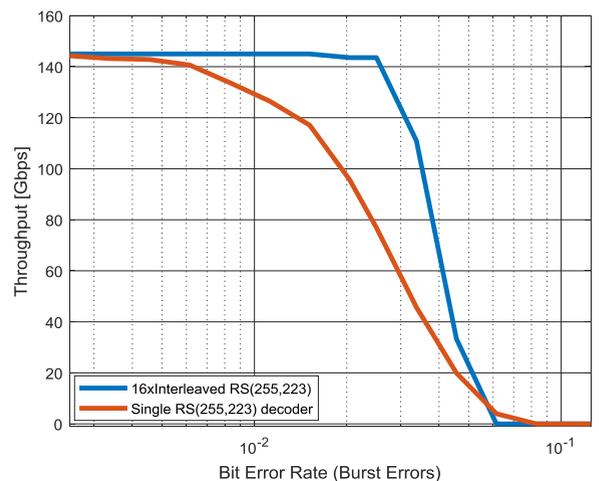


FIGURE 9. Comparison of correction performance for a single RS decoder and 16 x interleaved RS decoders for burst errors. Error characteristic is generated by a two-state Markov model with weighted average burst error length of 50 bits.

support soft-decision LDPC decoding. Thus, for our application, the RS is a more practical solution. Furthermore, we try to mitigate the gain loss of the RS codes by two means. Firstly, we interleave the decoders (Fig. 8), and therefore we can correct a longer burst error. In general, the symbol-interleaving improves correction performance for burst errors (Fig. 9). Long sequences of errors are interleaved among

multiple decoders, and therefore the effective number of erroneous symbols per decoder is reduced. This is important for our application because at 100 Gbps any synchronization error or voltage ripple destroys tens or even hundreds of consecutive bits. Thus, an extremely strong correction performance against burst errors is desired. Secondly, we designed a dedicated fragmentation and link adaptation schemes that improve the interleaved RS coding efficiency. Despite the fact that soft decision LDPC codes provide higher correction performance, we should note that ultra-high-speed LDPC decoders for  $\geq 100$  Gbps use hardware optimized decoding schemes and usually show lower error correction performance than sum-product (SPA) decoding. In the worst case, we lose only  $\sim 1$  dB as compared to soft decision LDPC shown in [12], considering AWGN channel and dedicated data fragmentation for RS codes. We need to keep in mind that similar fragmentation can be proposed for LDPC, as well as it is possible to implement an LDPC decoder with higher gain than in [12]. In [10], [33]–[35], we publish more details on the employed high-speed interleaved RS codes.

The interleaver size depends on the word size and interfaces available for the targeted technology. For Virtex7 FPGA, we usually interleave the data between eight RS decoders. This gives the processing speed of 64 bit/clock and fits the bus size of High-Speed Serial Transceivers [40], which are used as the main communication interfaces. Thus, for Xilinx FPGAs, the interleaver size is fixed to eight, or multiple of eight when the transceivers are combined in parallel. This gives the best power and area efficiency because the data do not have to be restructured and fits perfectly to the communication interfaces. In such a case, only routing resources are required to construct the interleaver.

In the case of ASIC implementation, we have more freedom and we can select any arbitrary defined size. Based on the results shown in section V, we know that to reach 100 Gbps with RS(255,223) coding, we need to combine at least 7 decoders ( $7 \times 14.7$  Gbps = 102.9 Gbps). Although a single RS decoder achieves up to 14.7 Gbps at 2.1 GHz, this mode is not recommended due to dissipated energy. The chip needs to run at the highest voltage (1.1V) and all power optimization options have to be disabled (e.g., clock gating, static power optimizations). This is reflected in the energy efficiency, which will be no better than  $\sim 15$  pJ/bit. Therefore, we increase the number of decoders and reduce the voltage and clock frequency. Moreover, we enable clock gating and optimize static power (more in section V). In such a case, a single RS decoder runs at 3.15 Gbps only, but the energy is optimized to  $\sim 2.4$  pJ/bit at 0.8V. This means that we need to place at least 32 decoders in parallel to reach 100 Gbps. In this paper, however, we compromise the area and energy, thus we decided to use 16 decoders.

Fig. 10 depicts energy efficiency as a function of the interleaver size, while Fig. 11 shows the expected ASIC area. To utilize energy optimization features, e.g., clock gating and static power optimizations, we need to place at least 11 decoders to reach 100 Gbps (max. 9.1 Gbps/decoder).

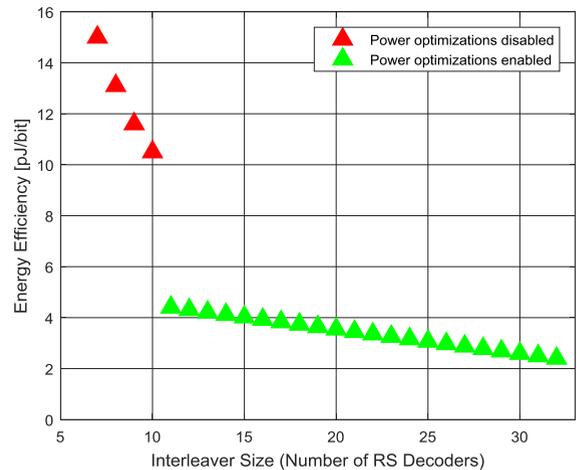


FIGURE 10. Energy efficiency as a function of interleaver size. At least 11 decoders need to be placed to utilize energy optimization features that are discussed in section V. User data throughput fixed at 100 Gbps is considered.

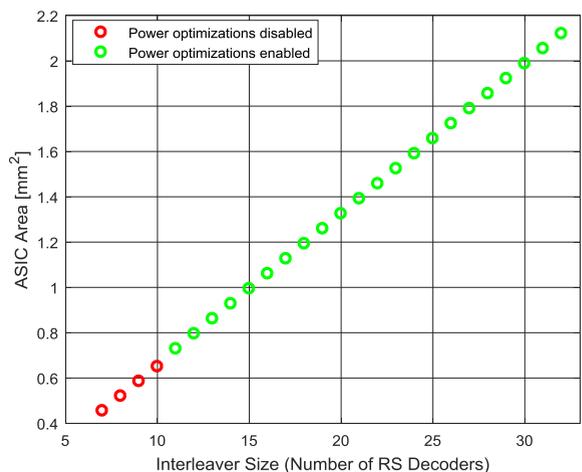


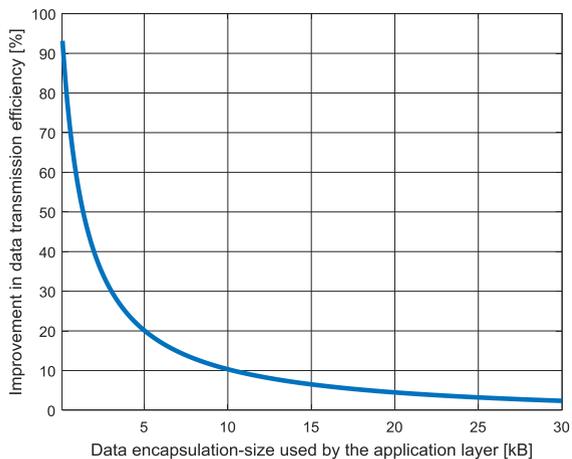
FIGURE 11. ASIC area as a function of interleaver size. User data throughput fixed at 100 Gbps is considered.

#### D. DATA AGGREGATION

Data aggregation is a widely used technique that significantly increases transmission performance in wireless systems. In our implementation, we set the minimal transmission frame length to 64 kB. Thus, we avoid frames shorter than 64 kB by merging the data when the system is fully loaded. This, in turn, reduces the total number of frames and frame-preambles, which are attached to each frame. In short, we reduce the transmission overhead. The overhead's influence on the effective throughput can be estimated as follows:

$$Throughput = \frac{Data}{Overhead + Data} \times PHY\_datarate. \quad (1)$$

The improvement in the performance of our method depends on the data size that is transmitted over the link (Fig. 12). For example, the throughput is increased by 47% when a typical 1.5 KB Ethernet data size is considered. In such a case, the aggregation module merges 43 Ethernet-frames



**FIGURE 12.** Improvement in data transmission efficiency gained by the implemented data aggregation scheme.

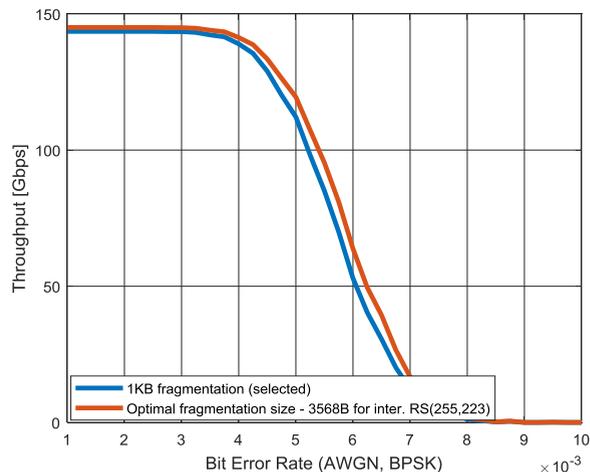
into a single wireless-frame that is transmitted over the air ( $43 \times 1.5\text{KB} = 64.5\text{KB}$ ).

**E. DATA FRAGMENTATION AND SELECTIVE FRAGMENT REPETITIONS**

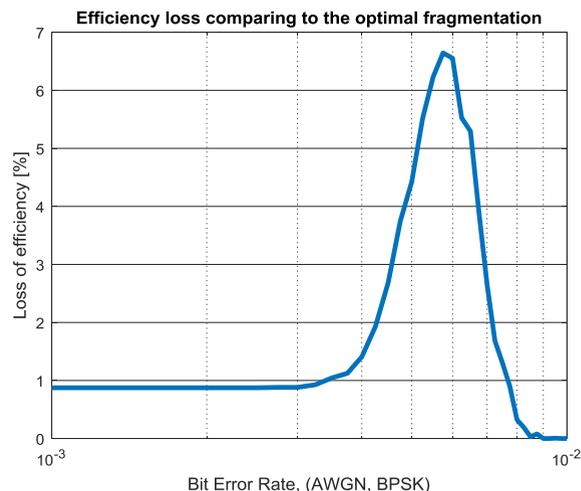
Although the 64kB-aggregation scheme significantly improves transmission efficiency for short frames, the aggregated frames are more sensitive to bit errors. Thus, we need efficient FEC and ARQ mechanisms that recover and retransmit corrupted data. Due to the targeted processing speed of 100 Gbps, we use the simplest HARQ-I method that is enhanced by selective fragment repetitions and link adaptation. In our case, the 64 KB frames are logically divided into 1 KB fragments, which have unique addresses and CRC sums. Thus, in case of bit errors, our ARQ retransmits 1KB-data fragments, instead of retransmitting the whole 64 KB-frames.

The selected 1 KB retransmission size is a tradeoff between the optimality and simplicity needed for practical realizations. From the data delivery efficiency point of view, the size should be equal to the message length of the employed FEC method. Then, the processor retransmits only the defected code words and the number of transmitted headers and CRCs remains at low. In our case, we use a set of 16 interleaved RS codes with variable message length in the range of 3568B – 4048B ( $16 \times 223\text{B} - 16 \times 253\text{B}$ ). This size depends on BER, which influences the overhead generated by the RS encoders (link adaptation, more in section III.F). Thus, we should adapt the retransmission size to BER continuously. Such an approach, however, leads to very complex implementation. The ARQ module needs to re-fragment the user data each time when the FEC code rate is changed and needs to keep the track of irregular fragment addressing. By fixing the size to 1 KB, we significantly reduce the complexity of the ARQ at the cost of reduced transmission efficiency (Fig. 13 and Fig. 14).

For  $\text{BER} < 3\text{e-}3$  and RS(255,223), the efficiency degradation of  $\sim 1\%$  is caused by redundant fragment-headers



**FIGURE 13.** Comparison of the transmission efficiency for the 1KB fragmentation and the optimal size. RS(255,223) encoding and AWGN channel are considered.



**FIGURE 14.** Comparison of the transmission efficiency for the 1KB fragmentation and the optimal size. RS(255,223) encoding and AWGN channel are considered.

and CRCs. For such a low BER, retransmissions are infrequent. For  $\text{BER} \in (3\text{e-}3, 6\text{e-}3)$ , we lose up to 7% of efficiency due to the fragment retransmission mismatch. For example, the ARQ has to retransmit a single interleaved-RS(255,223) block of the length of 3568 B ( $16 \times 223\text{B}$ ), but in our scheme, we need to retransmit  $5 \times 1\text{KB}$  (5120B) in the worst case. For  $\text{BER} > 6\text{e-}3$ , the wireless link is down regardless of the selected fragmentation scheme.

From the statistical point of view, the probability of error-free transmission of small fragments is higher than the probability of transmission of long frames. For our system, these probabilities can be estimated by the following formula:

$$P = \sum_{i=1}^r \left( \frac{(1-b)^{li} r!(1-(1-b)^l)^{r-i}}{i!(r-i)!} \right), \quad (2)$$

where  $P$  is the probability of error-free data delivery after  $r$  retransmissions,  $b$  is the bit error rate, and  $l$  is the data size in bits. More details on the data fragmentation can be found in [9], [10], [28], [41].

**F. LINK ADAPTATION**

Link adaptation algorithm tracks the link quality and finds the tradeoff between FEC redundancy and ARQ data repetitions. In short, the algorithm selects one of RS(255,  $k$ ) codes, where  $k$  is in the range of 223 to 253, so that the fragment error rate and FEC overhead are compromised. To keep the retransmission rate on a low level with low FEC overhead, we solve two inequalities in real time. The first inequality (3) compares whether the fragment error rate in the receiving stream (the left side of (3)) is higher than the RS redundancy (the right side of (3)):

$$\frac{\text{Erroneous fragments with RS (255, } k \text{) coding}}{\text{All fragments}} < \frac{255 - k}{255}. \tag{3}$$

If (3) is not satisfied, then the code rate has to be decreased and a more robust RS(255,  $k - 2$ ) code has to be used to reduce the retransmission. This means that more redundancy is added to the data frames.

The second inequality (4), increases the FEC-code rate and reduces the redundancy when the fragment error rate at an increased code rate will be low enough to satisfy (3). Thus, we need to solve the following relation for the RS(255,  $k + 2$ ) code:

$$\frac{\text{Erroneous fragments with RS(255, } k + 2 \text{)}}{\text{All fragments}} < \frac{255 - k + 2}{255}. \tag{4}$$

In (4), we need to predict the number of erroneous frame fragments at an increased code rate represented by RS(255,  $k + 2$ ) coding, which is relatively difficult to calculate. The processor decodes the data using RS(255,  $k$ ) and predicting the fragment error rate at RS(255,  $k + 2$ ) decoding is challenging. Thus, we estimate the number of erroneous frame fragments at RS(255,  $k + 2$ ) code by the RS-block error rate as follows:

$$\frac{\text{Err. fragm. with RS(255, } k + 2 \text{)}}{\text{All fragments}} \approx \frac{\text{Err. RS (255, } k + 2 \text{) blocks}}{\text{All RS (255, } k \text{) blocks}}. \tag{5}$$

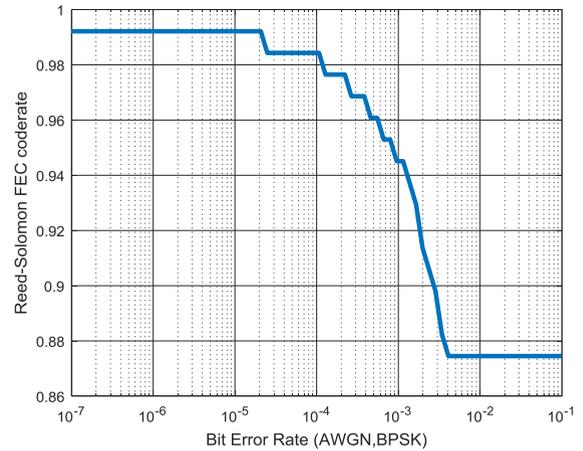
After that, (4) can be modified to (6):

$$\frac{\text{Erroneous RS (255, } k + 2 \text{) blocks}}{\text{All RS (255, } k \text{) blocks}} < \frac{255 - k + 2}{255} \tag{6}$$

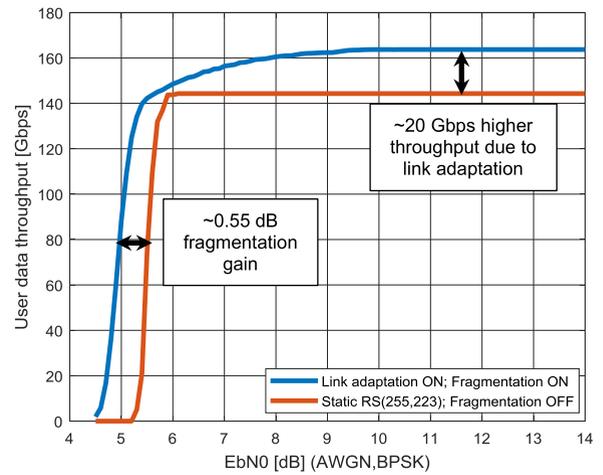
and can be easily solved, because RS(255,  $k + 2$ ) code corrects up to  $s$  symbols in an RS-block, where  $s$  is defined as:

$$s = \left\lfloor \frac{255 - k - 2}{2} \right\rfloor. \tag{7}$$

Thus, we simply count the number of symbol-errors in each RS-block and compare it with  $s$ . After that,



**FIGURE 15. Link adaptation scheme. The algorithm changes the FEC code rate according to the internally estimated fragment error rate.**



**FIGURE 16. Improvement in data transmission gained by the implemented fragmentation and link adaptation schemes. 28 nm SLP GlobalFoundries process at 1.1V is considered.**

a minimum-filtering is applied to improve the stability of the communication.

Fig. 15 demonstrates the operation of the algorithm as a function of bit error rate (BER) for the additive white Gaussian noise (AWGN) channel. With BER increase, the algorithm reduces the code rate of RS coders. That is to say, more redundancy is added to frames.

The uncomplicated HARQ-I method combined with link adaptation and selective fragment repetitions achieves pretty good efficiency, so we avoid HARQ-II and HARQ-III schemes [42]. We have already proven that implementing HARQ-II and HARQ-III at the targeted data rate is challenging [9], [11], [42].

Fig. 16 depicts the benefits of the implemented fragmentation and link adaptation. In our case, we achieve up to 20 Gbps higher throughput due to the link adaptation and ~ 0.55 dB higher gain due to the fragmentation. In [10], [29], [43], we publish more details on the employed link adaptation scheme.

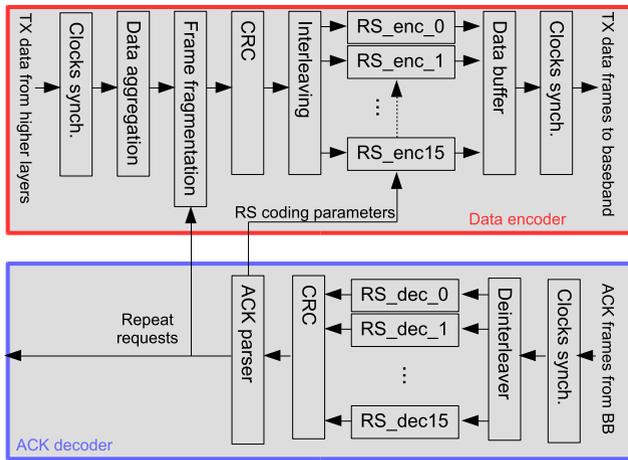


FIGURE 17. Architecture of the 28 nm data link layer processor – transmitter.

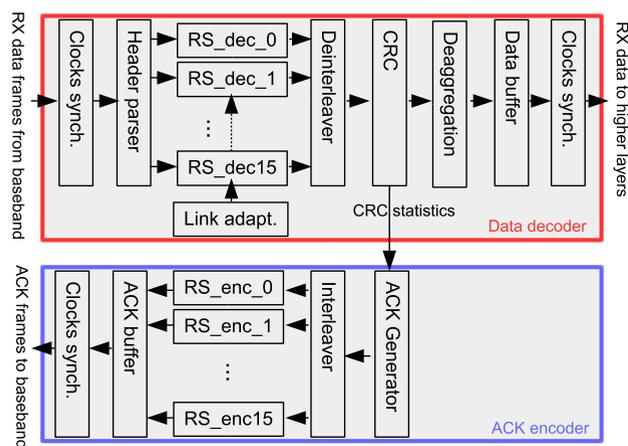


FIGURE 18. Architecture of the 28 nm data link layer processor – receiver.

#### IV. TRANSMITTER AND RECEIVER IMPLEMENTATION

Fig. 17 and Fig. 18 depict the transmitter and receiver implementation, respectively. The design has a 128-bit architecture, which means that 128 bits of data are processed in each clock cycle. Prototyped in Virtex7 FPGA, it is able to achieve  $\sim 9.9$  Gbps at a clock frequency of 156 MHz [10], [30]. When synthesized into 28 nm technology, we set the clock rate to 1.3 GHz and this gives the user data rate of 165 Gbps with RS(255,253) coding, as shown in Fig. 16, and 145.5 Gbps with RS(255,223) coding. Due to similar TX- and RX-architectures, the transmitter and receiver consume the same chip area of  $\sim 1.04$  mm<sup>2</sup> in 28 nm, and achieve the same clock speed and throughput. In fact, the transmitter and receiver have complete transmitting and receiving hardware due to ARQ and acknowledge-processing. The ARQ requires bidirectional communication, even if the user data flow is unidirectional, and therefore the data link layer transmitter has similar complexity as the receiver. Both units own a parallel array of eight RS encoders and decoders [38] with aggregated processing speed equivalent

to  $16 \times 10.313 = 165$  Gbps. All other processing is fast enough to handle the 165 Gbps data rate in a single thread.

#### V. ASIC SYNTHESIS AND LAYOUT

The design is fully implemented in VHDL, synthesized with Genus software, and the layout is made in Innovus. All presented power and energy results are estimated with real signal activity files (VCD) and performed on the chip layout considering typical process conditions. Each data word shifted to the chip is randomly generated and has changed 50% of the data bits as compared to the previous word. Thus, the measurements gave a realistic overview of the power and energy consumption.

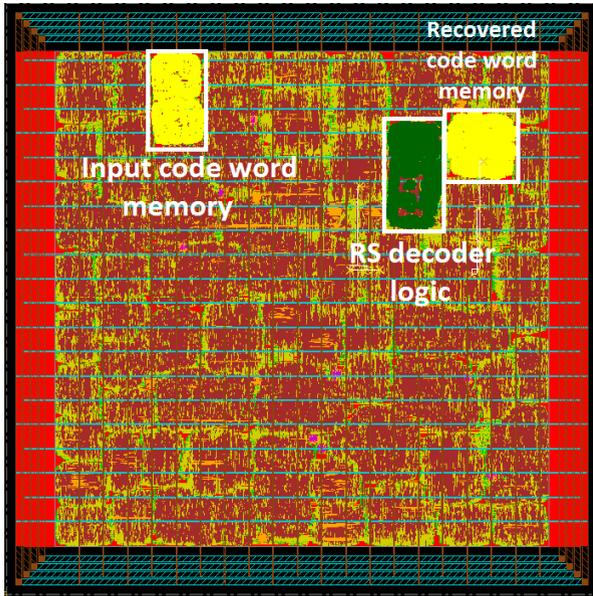
To achieve the reported throughput of 165 Gbps and 4.47 pJ/bit, we performed the following netlist and layout optimizations:

1. The dual port static RAM memories (FIFOs) needed for RS implementation [38] are replaced by Flip-Flop (FF) arrays. This solution sounds insane from the power and area point of view, but the memories are the main bottleneck of throughput in our design. Moreover, planning a chip with memories is more difficult than placing pure logic alone. In our case, we need to place 64 memories each of the size of  $256 \times 8$  bits. Replacing the memories with FF arrays increases the clock speed from 600 MHz up to 2100 MHz, which corresponds to the throughput improvement from 67 Gbps up to 235 Gbps. The performance is increased, but we also increase the chip area from 0.57 mm<sup>2</sup> to 1.02 mm<sup>2</sup> and the power from 0.286 W up to 3.5 W. In the next steps, power optimizations are performed. Mainly, we need to reduce the energy dissipated in the FF-arrays emulating the memory blocks.

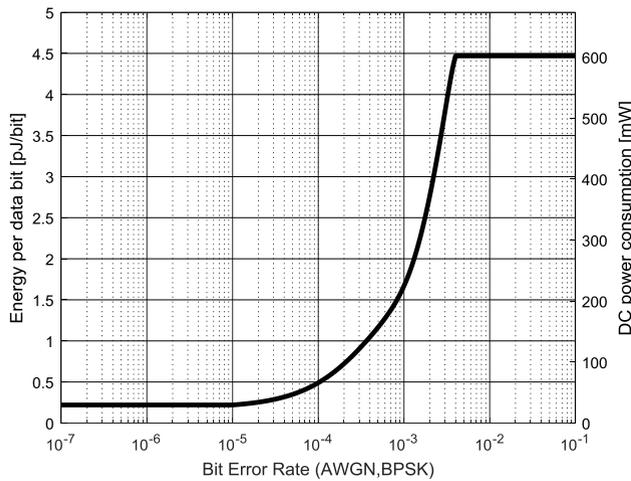
2. In the next step, clock gating is added to reduce the very high dynamic power. In each clock cycle, we read and write just a single byte to each FF-memory. This means that we access only  $\sim 0.19\%$  of the total memory registers in each clock cycle. Thus, we can significantly reduce the power by inserting clock gates and deactivating  $\sim 99.81\%$  of the memory registers. Although the clock gates increase the area by  $\sim 0.02$  mm<sup>2</sup> and reduce the clock by 600 MHz (2100 MHz  $\rightarrow$  1500 MHz), the power is desirably reduced to 0.928W from the initial 3.5W.

3. In the next step, we reduce the static power dissipated by the chip. This is achieved by performing multi-threshold voltage optimizations. In short, for all critical paths, the transistors with the lowest voltage switching threshold are inserted, while for non-critical paths transistors with a high threshold and reduced leakage are used. This reduces the power from 928 mW to 602 mW. After this step, the chip area remains almost unchanged, but the clock frequency is reduced by  $\sim 200$  MHz (1500 MHz  $\rightarrow$  1300 MHz).

The layout of the chip is shown in Fig. 19. We use a doubled VDD-VSS power ring around the placed logic. The IO pads are excluded from the area and power analysis. We highlighted a single RS decoder entity and its belonging code word memories. The input memory is placed close to the



**FIGURE 19.** ASIS layout of the implemented processor (receiver). A single RS decoder and its belonging memories are highlighted in green and yellow, respectively.

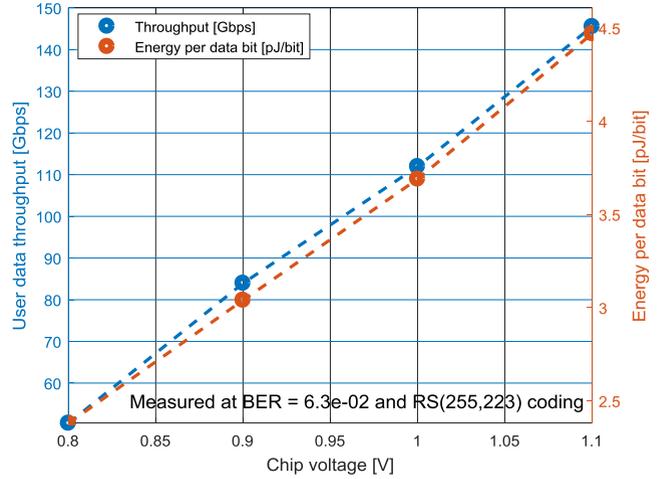


**FIGURE 20.** Average energy consumption per bit for the data link layer receiver as a function of bit error rate. 28 nm SLP GlobalFoundries typical case process at 1.1V is considered.

chip edge due to the input signals routing. The corrected code word, after fixing the evaluated error, is stored in the memory placed next to the decoder. It is possible to reduce the memory size by 25% by removing the bypass FIFOs, which are used to shift-out the originally received code word, in the case when the decoder cannot correct all bit errors.

**A. ENERGY CONSUMPTION**

As mentioned in the introduction, energy and power consumption are one of the most critical parameters of the high-speed transceivers. In our case, the energy and power depend on channel BER and selected FEC code. The energy is mostly consumed by the RS decoders and it is indeed related to the code rate curve shown in Fig. 15.



**FIGURE 21.** Throughput and energy per data bit as a function of chip voltage in the range of 0.8 to 1.1V. BER = 6.3e-2 and coding with the highest energy consumption, RS(255,223), is considered.

Figure 20 shows the variation in energy consumption versus BER. For BER < 1e-5 which is correspondent to the highest code-rate of RS(255,253), the processor consumes extremely low DC-power of 29.7 mW, equivalent to 0.22 pJ/bit. With the increase of BER, the DC-power also increases and saturates at 602 mW, or equivalently 4.47 pJ/bit. This high-power mode corresponds to the lowest code-rate of RS(255,223).

It should be noted that energy is dependent upon the number of Galois field multiplications (8) and additions (9), which in the case of RS(255, k) decoding, are asymptotically  $\Theta(n^2)$  and can be determined for our implementation [38] as follows:

$$\text{Multiplications} = 10 \left[ \frac{255 - k}{2} \right]^2 + 771 \left[ \frac{255 - k}{2} \right] - 255, \tag{8}$$

$$\text{Additions} = 6 \left[ \frac{255 - k}{2} \right]^2 + 764 \left[ \frac{255 - k}{2} \right]. \tag{9}$$

**B. VOLTAGE SCALING**

Although the throughput of 165 and 145 Gbps for RS(255,253) and RS(255,223) satisfies our needs, the maximum energy consumption of 4.47 pJ/bit exceeds our targeted limit. As mentioned in the introduction, our goal is to design a complete 100 Gbps transceiver (RF+BB+DLL) within 1 W power envelope. Assuming that the power is equally distributed between RF, BB, and DLL, we set the power limit of ~ 333 mW for our DLL implementation. This corresponds to the max. ~ 3.8 pJ/bit (information bit) at RS(255,223) coding. One workaround is to adjust the throughput, clock speed, and voltage in order to get some savings in consumed energy. The voltage range for the targeted process is 0.8-1.1V. Surprisingly, the clock speed of our design scales almost linearly with the voltage (Fig. 21), which is not observed for LDPC decoders realized in comparable technologies [14], [44]. The energy per bit looks to be a

**TABLE 2.** Performance comparison of FEC and data link layer processors.

	[12]	[48]	[49]	[44]	[47]	[45]	[14]	[46]	Current work
Technology	ST Micro. 65nm SVT CMOS	40nm G CMOS	40nm LP-CMOS	28nm UTBB FDSOI	28nm CMOS	ST Micro. 28nm FDSOI	28nm FDSOI	28nm	GlobalF. 28nm SLP CMOS
Estimation stage	Post Layout	Post Physical Synthesis	---	---	---	Post Synthesis	Post Layout	Post Layout	Post Layout
Design	ASIC	ASIC	ASIC	ASIC	ASIP	ASIC	ASIC	ASIC	ASIC, 128-bit data bus
Voltage [V]	1.2	0.9	1.1	1.07	0.9	0.9	0.6 - 0.9	0.9	0.8 - 1.1
Frequency [MHz]	257	500	220	260	470	---	---	451	1300 <sup>c</sup> 1000 <sup>d</sup> 750 <sup>e</sup> 450 <sup>f</sup>
FEC	LDPC 802.11ad	LDPC 802.11ad	LDPC 802.11ad	LDPC 802.11ad	LDPC 802.11ad	LDPC (30000, 26786)	LDPC 802.11ad	Polar code (1024,869)	Interleaved RS
Soft-decision decoding	YES, 4-bits	YES, 5-bits	YES, 5-bits	Yes, 5-bit	Yes, ---	Yes, 5-bit	Yes	Yes	NO
Coderate	13/16 ≈0.813	13/16 ≈0.813	1/2 ≈0.5	1/2 ≈0.5	1/2 ≈0.5	≈0.893	---	≈0.848	223/255 ≈0.875
Block size	672	672	672	672	672	30000	672	1024	2040
Eb/N0 [dB] at BER=1e-5 AWGN	~13 QAM64*	---	~3.5 BPSK*	~3.5 BPSK	~3.5 BPSK	~4.5 BPSK	---	---	~5.5** BPSK ~14** QAM64
Throughput [Gbps]	160.8	5.6	6.2	12	18.4	200	160 <sup>a</sup> 57.1 <sup>b</sup>	7.8	145.5 <sup>c</sup> 111.9 <sup>d</sup> 83.9 <sup>e</sup> 50.4 <sup>f</sup>
Chip area [mm <sup>2</sup> ]	12.09	0.16	0.8	0.63	0.78	3.73	2.8	0.35	1.04
Energy per data bit [pJ/bit]	32.49	17.7	32.9	30	18	1.5	6 <sup>a</sup> 2.9 <sup>b</sup>	1.41	4.47 <sup>c</sup> 3.69 <sup>d</sup> 3.04 <sup>e</sup> 2.38 <sup>f</sup>
Normalized throughput [Gbps/mm <sup>2</sup> ]	13.9	35	6.16	19.04	23.6	53.6	57.14 <sup>a</sup> 20.39 <sup>b</sup>	22.28	139.9 <sup>c</sup> 107.6 <sup>d</sup> 80.7 <sup>e</sup> 48.5 <sup>f</sup>
Functionality	FEC decoder	FEC decoder	FEC decoder	FEC decoder	FEC decoder	FEC decoder	FEC decoder	FEC decoder	FEC enc., FEC dec., HARQ-I, link-adapt., aggreg., fragm.

a) at 0.9V; b) at 0.6V; c) at 1.1V and RS(255,223); d) at 1.0V and RS(255,223); e) at 0.9V and RS(255,223); f) at 0.9V and RS(255,223); ---) not specified or unclear; \*) channel not specified, AWGN assumed; \*\*) including 0.55 dB fragmentation gain

linear function as well, but in the targeted range of 0.8-1.1V, a quadratic function fits the points more precisely. Both fitting curves are as follows:

$$\text{Throughput [Gbps]} \approx 313.42x - 199.81, \quad (10)$$

$$\text{Energy per bit} \left[ \frac{\text{pJ}}{\text{bit}} \right] \approx 3x^2 + 1.22x - 0.509, \quad (11)$$

where  $x$  represents the chip voltage and  $x \in [0.8, 1.1]$ .

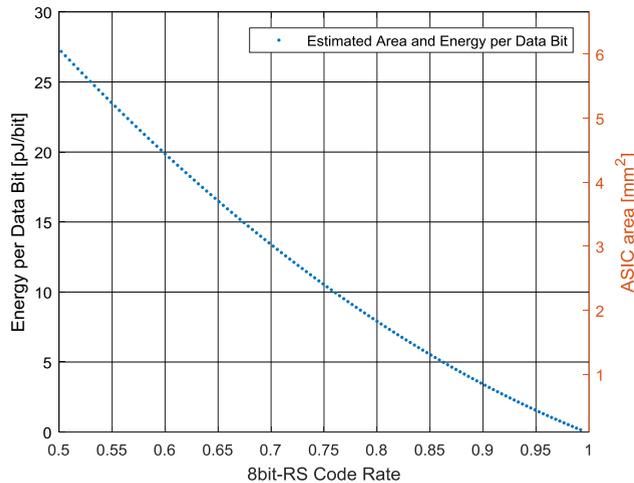
In our case, we need to reduce the throughput to  $\sim 115$  Gbps at  $\sim 1.01$  V to achieve the limit of  $\sim 3.8$  pJ/bit at BER  $\approx 6.3e-2$  with RS(255,223). The BER value of  $6.3e-2$  is the lowest achievable BER for AWGN channel as well as the worst case from the energy consumption point of

view. Assuming the lowest possible voltage of 0.8V, the processor achieves 50.4 Gbps and consumes max. 2.38 pJ/bit.

### C. COMPARISON WITH OTHER PUBLISHED WORK

To the best of our knowledge, there is not any comparable work providing similar comprehensive functionality to our implementation. We, hence, compare our work to existing high-speed LDPC and POLAR decoders (Tab. 2). As stated before, most of the chip resources are utilized by FEC, and therefore, such a comparison is fair.

Compared to high-speed LDPC at similar code rates [12], [45], our hard decision RS loses  $\sim 1$  dB gain. It rather leads to a smaller chip area and significantly higher



**FIGURE 22.** Estimated energy efficiency and area for algebraic, hard-decision 8 bit RS decoders - RS(255,k)  $k \in [127, 253]$  - for code rates lower than 0.8745, assuming 100 Gbps data throughput.

throughput normalized to the area. In our case, we achieve up to 140 Gbps/mm<sup>2</sup>, and this value is at least 2 times higher than for LDPC decoders. We require only 1.04 mm<sup>2</sup>, but LDPC requires 2 – 6 mm<sup>2</sup> at the same data rate. Moreover, we integrate a complete data link layer processor, not only FEC decoders. Thus, we believe that the 1 dB gain loss of the proposed hard decision method is mitigated by the superior area efficiency.

The other very important design parameter is energy efficiency. Our implementation can work with energy as low as 2.38 pJ/bit at 0.8V, which is a moderately good result. The LDPC [45] and POLAR [46] solutions require only 1.5 pJ/bit and 1.42 pJ/bit, respectively. The other 28nm-LDPC decoders consume 2.9 – 30 pJ/bit [44], [47], [14].

The data rate of our solution can be additionally improved. Currently, we process 128 bits/clock and use 1.04 mm<sup>2</sup> chip area only. Thus, there are no technical barriers to use more computation entities in parallel and process more than 165 and 145 Gbps at RS(255,253) and RS(255,223), respectively. This, however, is not the target of our research. We focus on a fully integrated 100 Gbps transceiver (RF + BB + DLL) that operates at  $\leq 1$ W.

#### D. RS CODES WITH CODE RATE LOWER THAN 223/255

In previous sections, we have emphasized the fact that the proposed interleaved RS solution can be used only for low overhead (high code rate), hard-decision, and high-speed FEC decoders. In Fig. 22, we evaluate the energy efficiency and area of our algebraic 8-bit-RS decoder for the code rates lower than  $223/255 \approx 0.8745$ . Assuming that the consumed energy and area is correlated with the number of Galois field multiplications (8), we estimate that 1/2-rate RS decoder will consume  $\sim 27$  pJ/bit and  $\sim 6$  mm<sup>2</sup> when realized in 28 nm technology. This clearly shows that LDPC decoders at this code rate are more energy efficient, e.g. solution presented in [47] needs only 18 pJ/bit at the same code rate. Moreover, the LDPC decoder will provide higher coding gain

by  $\sim 2$  dB, theoretically. The latency of such an RS algebraic decoder would be very high, probably more than 1000 cycles. In contrast to the RS, it is possible to implement a fully-parallel, full-unrolled LDPC decoder that needs only  $\sim 12$  cycles to decode a code word [12]. The very short decoding latency is one more advantage of LDPC codes over the presented interleaved RS solution. It does not mean that it is impossible to construct an efficient decoder for 1/2-rate RS codes, it means only that the presented algebraic solution should not be used for such codes due to practical difficulties, and another algorithm should be selected. One of such potential schemes is published in [50]. Probably, there exist other algorithms that are not yet (practically-)discovered.

## VI. CONCLUSION

In this paper, we presented 28 nm data link layer processor for 100 Gbps wireless communication in the THz-band. This processor uses lightweight interleaved RS codes and requires at least two times less chip area than LDPC decoders at the cost of  $\sim 1$  dB gain. Additionally, we show a dedicated link adaptation, aggregation, fragmentation, and ARQ with selective fragment repetitions. In our case, these methods improve user data throughput by max. 20 Gbps and the gain by  $\sim 0.55$  dB. ASIC post-layout results show that the processor easily achieves 145 Gbps and 165 Gbps at 1.1V with RS(255,253) and RS(255,223), respectively. Energy consumption is as low as 2.38 pJ/bit at 0.8V with RS(255,223). The methods achieve a good trade-off between throughput, energy consumption, and error correction performance for applications that do not require maximal coding gain and soft decision decoding. Additionally, we mention two novel baseband architectures, as well as two RF-frontends capable to work in the THz band. Challenges to high-speed wireless transmission are addressed as well.

## REFERENCES

- [1] F. Boes et al., "Ultra-broadband MMIC-based wireless link at 240 GHz enabled by 64GS/s DAC," in *Proc. 39th Int. Conf. Infr., Millim., THz. Waves (IRMMW-THz)*, Sep. 2014, pp. 1–2.
- [2] K. Krishnegowda, A. Wolf, R. Kraemer, J. C. Scheytt, and I. Kalfass, "Wireless 100 Gb/s: PHY layer overview and challenges in the THz frequency band," in *Proc. 15th Annu. IEEE Wireless Microw. Technol. Conf. (WAMICON)*, Jun. 2014, pp. 1–4.
- [3] T. Messinger et al., "Multi-level 20 Gbit/s PSSS transmission using a linearity-limited 240 GHz wireless frontend," in *Proc. Int. IEEE Conf. Microw., Commun., Antennas Electron. Syst. (COMCAS)*, Nov. 2015, pp. 1–3.
- [4] N. Sarmah et al., "A fully integrated 240-GHz direct-conversion quadrature transmitter and receiver chipset in SiGe technology," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 2, pp. 562–574, Feb. 2016.
- [5] Y. P. Zhang, M. Sun, K. M. Chua, L. L. Wai, and D. Liu, "Antenna-in-package design for wirebond interconnection to highly integrated 60-GHz radios," *IEEE Trans. Antennas Propag.*, vol. 57, no. 10, pp. 2842–2852, Oct. 2009.
- [6] S. K. Reynolds et al., "A silicon 60-GHz receiver and transmitter chipset for broadband communications," *IEEE J. Solid-State Circuits*, vol. 41, no. 12, pp. 2820–2831, Dec. 2006.
- [7] P. V. Testa et al., "On-chip integrated distributed amplifier and antenna systems in SiGe BiCMOS for transceivers with ultra-large bandwidth," *Frequenz*, vol. 71, nos. 9–10, pp. 827–830, Jan. 2017.

- [8] J. C. Scheytt et al., "100 Gbps wireless system and circuit design using parallel spread-spectrum sequencing," *Frequenz*, vol. 71, nos. 9–10, pp. 3591–3598, Jan. 2017.
- [9] L. Lopacinski, M. Brzozowski, and R. Kraemer, "A 100 Gbps data link layer with an adaptive algorithm for forward error correction," in *Proc. Inf. Commun. Technol. Forum, IEICE ICTF*, 2015, pp. 96–101.
- [10] L. Lopacinski, "Improving goodput and reliability of ultra-high-speed wireless communication at data link layer level," Ph.D. dissertation, Brandenburg Univ. Technol. Cottbus—Senftenberg, Senftenberg, Germany, 2017.
- [11] L. Lopacinski, M. Brzozowski, J. Nolte, S. Buechner, and R. Kraemer, "100 Gbps wireless—Challenges to the data link layer," in *Proc. Inf. Commun. Technol. Forum, IEICE ICTF*, 2014, pp. 86–91.
- [12] P. Schläfer, N. Wehn, M. Alles, and T. Lehnigk-Emden, "A new dimension of parallelism in ultra high throughput LDPC decoding," in *Proc. IEEE Workshop Signal Process. Syst.*, Oct. 2013, pp. 153–158.
- [13] N. Wehn, S. Scholl, P. Schläfer, T. Lehnigk-Emden, and M. Alles, "Challenges and limitations for very high throughput decoder architectures for soft-decoding," in *Advanced Hardware Design for Error Correcting Codes*. Cham, Switzerland: Springer, 2015, pp. 7–31.
- [14] S. Scholl, S. Weithoffer, and N. Wehn, "Advanced iterative channel coding schemes: When Shannon meets Moore," in *Proc. Int. Symp. Turbo Codes Iterative Inf. Process. (ISTC)*, Sep. 2016, pp. 406–411.
- [15] S. Weithoffer, M. Herrmann, C. Kestel, and N. Wehn, "Advanced wireless digital baseband signal processing beyond 100 Gbit/s," in *Proc. IEEE Int. Workshop Signal Process. Syst. (SiPS)*, Oct. 2017, pp. 1–6.
- [16] O. Villa et al., "Scaling the power wall: A path to exascale," in *Proc. Int. Conf. for High Perform. Comput., Netw., Storage Anal. (SC)*, 2014, pp. 830–841.
- [17] *International Technology Roadmap for Semiconductors 2.0: Executive Report*, ITRS, Semicond. Ind. Assoc., Washington, DC, USA, 2015. [Online]. Available: <https://www.semiconductors.org/> and <https://www.semiconductors.org/resources/2015-international-technology-roadmap-for-semiconductors-itrs/>
- [18] I. Kallfass et al., "Towards MMIC-based 300 GHz indoor wireless communication systems," *IEICE Trans. Electron.*, vol. E98-C, no. 12, pp. 1081–1090, 2015.
- [19] A. Leuther, A. Tessmann, M. Dammann, H. Massler, M. Schlechtweg, and O. Ambacher, "35 nm mHEMT Technology for THz and ultra low noise applications," in *Proc. Int. Conf. Indium Phosph. Rel. Mater. (IPRM)*, 2013, pp. 1–2.
- [20] M. H. Eissa et al., "Wideband 240-GHz transmitter and receiver in BiCMOS technology with 25-Gbit/s data rate," *IEEE J. Solid-State Circuits*, vol. 53, no. 9, pp. 2532–2542, Sep. 2018.
- [21] IHP GmbH. (2019). *IHP ASIC Technologies*. Accessed: Jan. 23, 2019. [Online]. Available: <https://www.ihp-microelectronics.com/en/services/mpw-prototyping/sigec-bicmos-technologies.html>
- [22] M. H. Eissa, A. Malignaggi, G. Panic, L. Lopacinski, R. Kraemer, and D. Kissinger, "Modular wideband 1–15 GHz transmitter channelizer for high data rate communication," in *Proc. 11th Global Symp. Millim. Waves (GSMM)*, 2018, pp. 1–3.
- [23] A. R. Javed, J. C. Scheytt, K. Krishnegowda, and R. Kraemer, "System design considerations for a PSSS transceiver for 100 Gbps wireless communication with emphasis on mixed signal implementation," in *Proc. 16th IEEE Annu. Wireless Microw. Technol. Conf. (WAMICON)*, Apr. 2015, pp. 1–4.
- [24] A. C. Wolf and M. Mahlig, "Benchmarking of WSN Solutions and IEEE 802.15.4-2006 PSSS based solutions," in *Proc. 9th GI/ITG KuVS Fachgespräch Sensornetze*, 2010, p. 13.
- [25] H. Schwetlick, "PSSS-parallel sequence spread spectrum a potential physical layer for OBAN?" in *Proc. OBAN Workshop Mobility, QoS Secur. Open Wireless Access Netw., 14th IST Mobile Wireless Commun. Summit*. Berlin, Germany: Hochschule für Technik und Wirtschaft, 2005. [Online]. Available: <http://oban.tubit.tu-berlin.de/5-PSSS-Schwetlick.pdf>
- [26] H. Schwetlick and A. Wolf, "PSSS—Parallel sequence spread spectrum a physical layer for RF communication," in *Proc. IEEE Int. Symp. Consum. Electron.*, Sep. 2004, pp. 262–265.
- [27] A. R. Javed and J. C. Scheytt, "System design and simulation of a PSSS based mixed signal transceiver for a 20 Gbps bandwidth limited communication link," in *Proc. 1st URSI Atlantic Radio Sci. Conf. (URSI AT-RASC)*, 2015, p. 1.
- [28] L. Lopacinski, M. Brzozowski, and R. Kraemer, "Data link layer considerations for future 100 Gbps terahertz band transceivers," *Wireless Commun. Mobile Comput.*, vol. 2017, Jan. 2017, Art. no. 3560521.
- [29] L. Lopacinski, M. Brzozowski, R. Kraemer, S. Buechner, and J. Nolte, "Improving energy efficiency using a link adaptation algorithm dedicated for 100 Gbps wireless communication," *AEU-Int. J. Electron. Commun.*, vol. 81, pp. 67–73, Nov. 2017.
- [30] L. Lopacinski, M. Brzozowski, R. Kraemer, S. Buechner, and J. Nolte, "100 Gbps wireless—Data link layer VHDL implementation," in *Proc. 18th Conf. Reconfigurable Ubiquitous Comput. (RUC)*, 2015, pp. 333–336.
- [31] *Frame Structure Channel Coding and Modulation for a Second Generation Digital Terrestrial Television Broadcasting System (DVB-T2)*, document ETSI EN 302 755 v1.2.1-DVB, ETSI, 2010, pp. 1–177.
- [32] *Low-Density Parity-Check Codes From DVB-S.2 Standard—MATLAB DVBS2LDPC*. Accessed: Feb. 25, 2019. [Online]. Available: <https://www.mathworks.com/help/comm/ref/dvbs2ldpc.html>
- [33] *Decode Binary Low-Density Parity-Check Code Specified by Parity-Check Matrix—Simulink—MathWorks Switzerland*. Accessed: Feb. 25, 2019. [Online]. Available: [https://ch.mathworks.com/help/comm/ref/ldpcdecoder.html?searchHighlight=ldpc&tid=doc\\_srchtile](https://ch.mathworks.com/help/comm/ref/ldpcdecoder.html?searchHighlight=ldpc&tid=doc_srchtile)
- [34] *3GPP Turbo Decoder v4.0*, Xilinx Corp., San Jose, CA, USA, 2009.
- [35] *Performance and Resource Utilization for LDPC Encoder/Decoder*. Accessed: Feb. 20, 2019 [Online]. Available: <https://www.xilinx.com/member/ldpc-enc-dec/ldpc.html>
- [36] *LogiCORE IP Viterbi Decoder v9.0—Product Guide, Xilinx Datasheet*, Xilinx Corp., San Jose, CA, USA, 2014.
- [37] M. Marinkovic, M. Piz, C.-S. Choi, G. Panic, M. Ehrig, and E. Grass, "Performance evaluation of channel coding for Gbps 60-GHz OFDM-based wireless communications," in *Proc. IEEE Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2010, pp. 994–998.
- [38] M. Marinkovic, M. Krstic, E. Grass, and M. Piz, "Performance and complexity analysis of channel coding schemes for multi-Gbps wireless communications," in *Proc. IEEE 23rd Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2012, pp. 1937–1943.
- [39] *LogiCORETM IP Reed-Solomon Decoder Product Guide*, Xilinx Corp., San Jose, CA, USA, 2015.
- [40] *7 Series FPGAs GTX/GTH Transceivers, Xilinx Datasheet*, Xilinx Corp., San Jose, CA, USA, 2012.
- [41] L. Lopacinski, M. Brzozowski, R. Kraemer, J. Nolte, S. Buechner, and R. Kraemer, "Design and performance measurements of an FPGA accelerator for a 100 Gbps wireless data link layer," in *Proc. URSI Atlantic Radio Sci. Conf. (URSI AT-RASC)*, 2015, p. 1.
- [42] L. Lopacinski, M. Eissa, G. Panic, M. Brzozowski, A. Hasani, and R. Kraemer, "Implementation of a multi-core data link layer processor for THz communication," in *Proc. IEEE Veh. Technol. Conf.*, Jun. 2018, pp. 1–5.
- [43] L. Lopacinski, J. Nolte, S. Buechner, M. Brzozowski, and R. Kraemer, "Design and implementation of an adaptive algorithm for hybrid automatic repeat request," in *Proc. 18th IEEE Int. Symp. Design Diagnostics Electron. Circuits Syst. (DDECS)*, Apr. 2015, pp. 263–266.
- [44] M. Weiner, M. Blagojevic, S. Skotnikov, A. Burg, P. Flatresse, and B. Nikolic, "A scalable 1.5-to-6Gbps 6.2-to-38.1 mW LDPC decoder for 60 GHz wireless networks in 28 nm UTBB FDSOI," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2014, pp. 464–465.
- [45] K. Cushon, P. Larsson-Edefors, and P. Andrekson, "Low-power 400-Gbps soft-decision LDPC FEC for optical transport networks," *J. Lightw. Technol.*, vol. 34, no. 18, pp. 4304–4311, Sep. 15, 2016.
- [46] P. Giard et al., "PolarBear: A 28-nm FD-SOI ASIC for decoding of polar codes," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 7, no. 4, pp. 616–629, Dec. 2017.
- [47] M. Li et al., "An energy efficient 18 Gbps LDPC decoding processor for 802.11ad in 28 nm CMOS," in *Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, Nov. 2015, pp. 1–5.
- [48] M. Li et al., "An area and energy efficient half-row-paralleled layer LDPC decoder for the 802.11AD standard," in *Proc. SIPS*, 2013, pp. 112–117.
- [49] H. Motozuka, N. Yosoku, T. Sakamoto, T. Tsukizawa, N. Shirakata, and K. Takinami, "A 6.16 Gb/s 4.7 pJ/bit/iteration LDPC decoder for IEEE 802.11ad standard in 40 nm LP-CMOS," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Dec. 2015, pp. 1289–1292.
- [50] G. Schmidt, V. Sidorenko, and M. Bossert, "Decoding reed-solomon codes beyond half the minimum distance using shift-register synthesis," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2006, pp. 459–463.



**LUKASZ LOPACINSKI** received the M.Sc. degree in computer science from the West Pomeranian University of Technology, Szczecin, Poland, in 2009, and the Ph.D. degree from the Brandenburg University of Technology Cottbus–Senftenberg, Germany, in 2017. From 2007 to 2013, he has worked in industrial companies in the field of embedded systems and wireless communication. From 2013 to 2016, he was a Research Assistant with the Brandenburg University of Technology Cottbus–Senftenberg. Since 2016, he has been with IHP Microelectronics, Frankfurt (Oder), Germany.



**MIROSLAV MARINKOVIC** received the Dipl.Ing. and M.S.E.E. degrees from the University of Nis, Serbia, in 2004 and 2008, respectively. From 2008 to 2014, he was with IHP Microelectronics, Frankfurt (Oder), Germany, as a Research Associate. Since 2014, he has been with Arquimea Deutschland GmbH, Frankfurt, as a Digital Design/Verification Engineer. His research activities are focused on digital signal processing for high-speed wireless communications and error control coding.



**GORAN PANIC** was born in Sarajevo, Bosnia and Herzegovina, in 1976. He received the Diploma degree from the Faculty of Electronic Engineering, University of Nis, Nis, Serbia, in 2001, and the Ph.D. degree in engineering from the Technical University Cottbus–Senftenberg, Germany, in 2014. Following his graduation, he worked as a Research Assistant in the area of microprocessor architectures with the University of Nis. He is currently employed as a Scientist with the System

Department of IHP, Frankfurt (Oder), Germany. His research areas include SoC design, and low-power and communication systems.



**MOHAMED HUSSEIN EISSA** received the B.Sc. degree in electrical engineering from Ain Shams University, Cairo, Egypt, in 2009, and the M.Sc. degree in electronics and communications from American University in Cairo, New Cairo, Egypt, in 2014. He was with Silicon Vision LLC, Cairo, where he was involved in the design of ASICs for low-power Bluetooth communication standards. From 2011 to 2014, he was with Hitite Microwave, Cairo, working on the design of transceivers for point-to-point communication. Since 2014, he has been a Research Scientist with the Circuit Design Department, IHP, Frankfurt (Oder), Germany. His current research interest includes RF and millimeter-wave circuit designs for communication applications.



**ALIREZA HASANI** received the B.Sc. and M.Sc. degrees in electrical engineering from the Iran University of Science and Technology (IUST), Tehran, Iran, in 2012 and 2015, respectively. Subsequently, he has been continuing his studying as a Ph.D. student with Brandenburg University of Technology Cottbus–Senftenberg, and as a Guest Scientist with IHP-Microelectronics, Frankfurt (Oder), Germany. Within his Ph.D., he is working mainly on the design of forward-error correction codes for high-throughput communications systems. Among various fields of telecommunications, his research interests are mainly on the different issues related to wireless communications and mobile broadband networks.



**KARTHIK KRISHNEGOWDA** received the bachelor's degree in electronics and communication from Visvesvaraya Technological University, in 2009, and the M.Sc. degree in telecommunication from the Warsaw University of Technology, in 2013, with an Erasmus Mundus Scholarship. He is currently pursuing the Ph.D. degree with the Brandenburg University of Technology Cottbus–Senftenberg, Germany. He is also a Research Assistant with the System Design Department, IHP-Microelectronics, Frankfurt (Oder). His research interest includes the development of physical baseband layer algorithms for high-speed wireless communication.



**ROLF KRAEMER** received the Diploma and Ph.D. degrees in electrical engineering and computer science from RWTH Aachen, in 1979 and 1985, respectively. In 1985, he joined the Philips Research Laboratories, where he worked in different positions and responsibilities. In 1998, he became a Professor with the Technical University of Cottbus, with the joined appointment of the Department Head of wireless systems at the IHP-Microelectronics, Frankfurt (Oder), where he leads a research department with approximately 70 researchers in topics of high-speed wireless communication, context-aware middleware, sensor networks, as well as embedded processors for encryption, and protocol acceleration.

...