This document is a preprint of: D. Swist, S. Friedrich, E. Matus et al., "Implementation Aspects of RLL Encoding for Zero-Crossing Modulation based Wideband Communications," in Proceedings of International Conference on Mobile and Miniaturized Terahertz Systems (ICM2TS 2025), Dubai, United Arab Emirates, Feb 2025. Invited.

Implementation Aspects of RLL Encoding for Zero-Crossing Modulation based Wideband Communications

Daniel Swist, Simon Friedrich, Emil Matúš, Meik Dörpinghaus, and Gerhard Fettweis
Vodafone Chair Mobile Communications Systems, Technische Universität Dresden, 01062 Dresden, Germany
{daniel.swist, simon.friedrich, emil.matus, meik.doerpinghaus, gerhard.fettweis}@tu-dresden.de

***Abstract*—For wideband terahertz communications systems the analog-to-digital converter power consumption is expected to become a major challenge. The use of receivers with 1-bit quantization and oversampling w.r.t. the transmit signal bandwidth in combination with zero-crossing modulation is considered as a promising solution to realize energy-efficient wideband communications systems. Such a zero-crossing modulation can be realized using a combination of runlength-limited (RLL) sequences and faster-than-Nyquist signaling. However, the high-speed implementation of the RLL encoding and decoding poses severe challenges due to their inherently serial nature. In the present paper, we present an implementation and synthesis results of an RLL encoder in 22 nm FDX CMOS technology, demonstrating that data rates in the order of 100 Gbps can be achieved. Moreover, technology scaling shows the potential for even higher throughputs.**

I. Introduction

The rapid evolution of data-intensive applications is driving the demand for wireless communications systems capable of achieving data rates of 100 gigabit-per-second (Gbps) and beyond. Terahertz (THz) communications systems are emerging as a critical enabler to achieve such data rates, leveraging the extensive bandwidth available in the THz spectrum [1]–[3]. However, the large sampling frequencies required by high-bandwidth THz systems lead to a substantial increase in analog-to-digital converter (ADC) power consumption [4], [5]. To address this challenge, reducing the ADC bit resolution to 1 bit combined with temporal oversampling w.r.t. the signal bandwidth and the use of zero-crossing modulation (ZXM) [6], [7], presents a promising solution for balancing power efficiency and high-speed data transmission in future THz communications systems.

The use of temporal oversampling allows to partially compensate the loss in communication performance due to 1-bit quantization. With ZXM realized using a combination of run-length-limited (RLL) sequences and faster-than-Nyquist (FTN) signaling spectral efficiencies of up to $4\,\mathrm{bit/s/Hz}$ have been observed [8]. However, processing speed in digital baseband systems is significantly limited by algorithmic and technological constraints, which become increasingly severe for the envisioned data rates, ultimately limiting the practically achievable maximum throughput.

In a digital circuit, the largest delay of an electrical signal propagation is called critical path. This delay determines the maximum clock frequency and, in turn, the achievable throughput. While traditional methods like pipelining and parallelization using block processing are commonly used to enhance throughput [9], they present unique challenges when applied to RLL encoding. In [8], the RLL encoder used for ZXM is inherently highly serial, with a direct feedback loop, which cannot be made uncritical by splitting and introducing buffer registers. This presents a major bottleneck for high-speed implementation and complicates efforts to implement pipelining or parallelization. Thus, innovative architectural approaches are required to address these limitations in high-speed processing.

In addition to algorithmic constraints, the physical limitations of the underlying semiconductor technology further restrict the achievable speed. Although high-speed technologies such as Emitter-Coupled Logic (ECL) may offer higher clock frequencies [10], complementary metal–oxide–semiconductor (CMOS) technology provides a more power-efficient implementation option and is favorable in many aspects for THz communications [11]. Hence, it is vital to anaylze which data rates can be achieved using RLL encoder implementations in commonly available CMOS technologies.

This paper makes several contributions to address these challenges and enable data rates of $100\,\mathrm{Gbps}$ and beyond. First, we analyze the RLL encoding scheme for ZXM described in [8] focusing on its implementation challenges and propose an optimization strategy using composite states that improves throughput. Second, we present implementation and synthesis results of the RLL encoder in GlobalFoundries (GF) 22 nm FDX CMOS technology, demonstrating its feasibility and performance. Third, we estimate the impact of technology scaling on throughput, providing insight into how advances in semiconductor technology can further enhance system performance. Finally, we propose a system-level concept that

This work has been funded in part by the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme within the project "TeraGreen", Grant Agreement No 101139117, and in part by the German Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung, BMBF) as part of the GreenICT-E4C project under grant number 16ME0426K. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

979-8-3503-6583-2/25/$31.00 © 2025 IEEE

© 2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

This document is a preprint of: D. Swist, S. Friedrich, E. Matus et al., "Implementation Aspects of RLL Encoding for Zero-Crossing Modulation based Wideband Communications," in Proceedings of International Conference on Mobile and Miniaturized Terahertz Systems (ICM2TS 2025), Dubai, United Arab Emirates, Feb 2025. Invited.

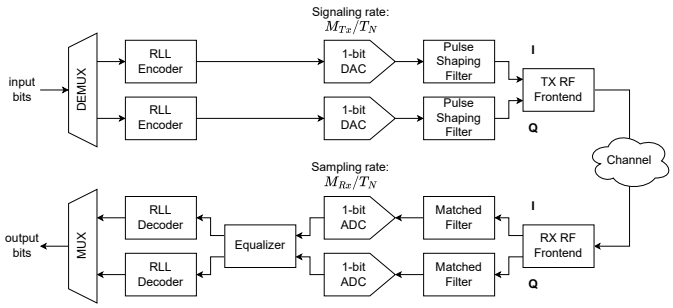


Fig. 1: Overview of ZXM system

outlines architectural strategies for achieving data rates of $100\,\mathrm{Gbps}$ and beyond in RLL encoder implementations.

II. ZXM System Model

ZXM is a concept for receivers using 1-bit quantization with temporal oversampling designed to mitigate spectral efficiency limitations compared to systems using receivers with fine grained amplitude resolution, which allow the use of high-order modulation schemes [6], [7]. ZXM encodes information in the temporal spacing between zero-crossings, effectively shifting modulation from the amplitude domain to the time domain. Thus it is tailored to receivers with 1-bit quantization which are only able to resolve zero-crossings of the receive signal. Two fundamental principles drive the ZXM system:

- **Faster-than-Nyquist (FTN) signaling** increases the signaling rate beyond Nyquist rate at the digital-to-analog converter (DAC) and hence reduces the temporal spacing between transmitted symbols. M_{Tx} symbols are transmitted per Nyquist interval T_N determined by the transmit signal bandwidth. This allows to place zero-crossings on a finer grid to increase spectral efficiency [6], [7] but yields inter-symbol-interference (ISI).
- **Runlength-limited (RLL) sequences** are binary sequences that impose constraints on the number of consecutive identical symbols, ensuring a minimum and maximum runlength of samples of equal sign between zero-crossings to control signal transitions and reduce ISI. RLL sequences are generated from (d, k) binary sequences, which enforce at least d and at most k zeros between each '1', by non-return-to-zero-inverse (NRZI) encoding, where each '1' causes a switch in the sign [12]. The minimum distance between zero-crossings in the transmitted signal is critical for controling ISI introduced by the FTN signaling. According to [8] the RLL encoding is realized using a finite-state machine (FSM), which produces sequences satisfying the (d, k) constraint.

A block diagram of the considered ZXM-based communications system as it has been described in [8] is shown in Fig. 1. At the baseband, the transmitter receives a sequence of bits from the channel coder (not shown), which are demultiplexed to produce two separate bit streams transmitted on the in-phase and quadrature (IQ) components of a complex-valued quadrature modulation. These bit streams are independently processed by the RLL encoder. The RLL encoded sequences are then fed to 1-bit DACs operating at a FTN signaling rate of M_{Tx}/T_N where we choose $M_{Tx} = d+1$ [8] with d being the d-constraint of the RLL encoder. Increasing d allows higher spectral efficiencies [8] at the expense of increased signaling rates of the DAC. The DAC outputs are filtered through pulse shaping filters, which determine T_N and form the baseband IQ signal. The radio frequency (RF) frontend performs upconversion to the carrier frequency and amplifies the signal for transmission. The signal is transmitted over a wireless channel. At the receiver the RF frontend downconverts the signal to baseband, where it is filtered by matched filters and digitized by 1-bit ADCs using temporal oversampling at a rate of M_{Rx}/T_N with $M_{Rx} \geq M_{Tx}$. The digitized sequences are processed by an equalizer. The sequences are then decoded by the RLL decoder, with the aim to recover the original RLL encoder input bits, and the resulting bit streams are multiplexed before being further processed by the channel decoder (not shown).

The ZXM-specific portion of the digital baseband processing on the transmitter side consists solely of the RLL encoder, making it the primary factor limiting throughput. Therefore, it is crucial to develop efficient methods for its implementation.

III. The THz Scenario

Analyzing the feasibility and developing practical implementation strategies of the described system are critical. The data rates and bandwidth of such THz systems will depend on the application, characteristics of the RF frontends, and the available bandwidth. In [13], for the example of a board-to-board communication inside a server, a bandwidth of 30 GHz at 190 GHz carrier frequency is described as a practically relevant configuration, aiming at $100\,\mathrm{Gpbs}$. This requires a spectral efficiency of $3.33\,\mathrm{b/s/Hz}$ if no further techniques like polarization multiplexing are applied, which can be achieved with ZXM, see [8, Fig. 10].

Real-time digital processing for such data rates presents significant challenges. In the possible selection of suitable architectural blocks [14], current digital signal processor and field programmable gate array technologies lack the flexibility and performance required for the baseband processing of ZXM at these high data rates. Implementing high-speed processing macros using custom circuits is complex and premature without proof-of-concept designs. While fully custom or even mixed-signal designs may eventually be required for widespread implementation, it is benefical to assess the performance of implementations in easily accessible technology. The integration of CMOS into THz wireless communications architectures is an open question [11]. Hence, this work considers a standard cell macro implementation in GF 22 nm FDX CMOS technology to assess the feasibility and throughput of an RLL encoder implementation for THz systems.

IV. RLL Encoder Design

The RLL encoder maps the input bits onto RLL sequences. In this work, we consider the FSM codes described in [8]. The RLL encoding process shown in Fig. 2 consists of two steps:

- encoding of bits onto (d, k) sequences by the RLL FSM
- non-return-to-zero-inverse (NRZI) encoding

© 2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

This document is a preprint of: D. Swist, S. Friedrich, E. Matus et al., "Implementation Aspects of RLL Encoding for Zero-Crossing Modulation based Wideband Communications," in Proceedings of International Conference on Mobile and Miniaturized Terahertz Systems (ICM2TS 2025), Dubai, United Arab Emirates, Feb 2025. Invited.

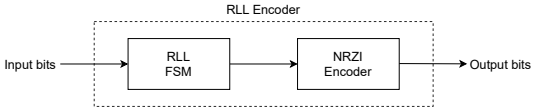


Fig. 2: Block diagram of RLL encoder consisting of RLL FSM and NRZI encoder

TABLE I: Parameters of the RLL FSM codes in [8]

(d,k)	**input** p	**output** q	**states** s	**code rate** r_{RLL}	**state bits** w
$(1,\infty)$	2	3	3	2/3	2
$(2,\infty)$	1	2	4	1/2	2
$(3,\infty)$	3	7	9	3/7	4
$(4,\infty)$	3	8	10	3/8	4

The FSM codes in [8] allow the mapping of the bits onto (d,k) sequences and the NRZI encoding in one step. While this joint representation is useful for the decoding, it doubles the number of states of the FSM in comparison to the FSM describing the encoding into the (d,k) sequence, coming along with increased logic complexity and an extended critical path. To enable pipelining and accelerate processing, it is beneficial to implement the RLL FSM and NRZI encoding separately.

The NRZI encoder in the system model in [8] assumes a bipolar signal output (positive and negative voltage levels), while standard cell digital logic typically operates with unipolar logic (low and high voltage levels). Since the concept of ZXM requires specialized high-speed 1-bit DACs, we motivate implementations of an optimized DAC module with a unipolar interface that integrates NRZI logic. Since, the NRZI logic is simple (switching the output level when a '1' is encountered) this can easily be integrated in an analog circuit, which would facilitate interfacing to common digital technology.

A. RLL FSM Codes

The RLL FSM codes provided in [8] map input bit sequences of size p to q output bits, yielding a code rate of $r_{\text{RLL}} = \frac{p}{q}$. In each iteration of the FSM, this mapping is determined based on the current state by a codebook. The number of states s varies depending on the parameter d, and the number of bits needed for state encoding is $w = \lceil \log_2(s) \rceil$. Table I shows the parameters of the FSMs described in [8].

For a data rate R at the input of the encoder, the code rate determines the required signaling rate of the 1-bit DAC:

$$f_{DAC} = \frac{M_{Tx}}{T_N} = \frac{q}{p} \cdot R \tag{1}$$

The subsequent state is determined based on the current state and on the input bits, and is used as input in the next iteration. A direct approach to implementing this FSM is to process one FSM iteration per clock cycle. Fig. 3 illustrates the block diagram of the RLL FSM. While RLL encoding does not involve any resource-heavy numerical operations, the feedback loop in the FSM is essential and creates a critical path that cannot be split and limits parallelization. This FSM follows the Mealy machine model, where the feedback loop is the primary bottleneck for high-speed implementation.

B. Implementation Approach

Since the number of input bits p and the maximum number of state encoding bits w is low (cf. Table I), the simplest method to implement the FSM in a digital circuit is by using flip-flops to store state information and combinational logic for state transitions and output generation. Other methods of implementing the FSM as a look-up table using memory elements (e.g. SRAM) introduce additional complexity and may introduce delays, particularly in the feedback loop, increasing the length of the critical path.

The RLL FSM has a throughput of $R = p \cdot f_{CLK}$ with the clock rate f_{CLK} for which the FSM is synthesized. The maximum clock frequency is determined by the maximum time delay after synthesis and depends on the circuit complexity and the underlying technology.

C. Composite States

To increase the number of bits processed per clock cycle, composite states can be used. Multiple state transitions are processed in a single clock cycle by pre-computing state transitions for $L \cdot p$ input bits and concatenating the corresponding output bit sequences to a sequence of $L \cdot q$ output bits. The number of merged FSM transitions, denoted by L, determines how many original states are processed per clock cycle in the composite state FSM. The numbers of input and output bits of the FSM logic are then $N_i = L \cdot p + w$ and $N_o = L \cdot q + w$, respectively. Hence, the throughput per clock cycle at the input

$$\frac{R}{f_{CLK}} = L \cdot p \tag{2}$$

and likewise at the output is linearly dependent on L. However, the achievable clock frequency might also be affected due to increasing complexity with increased L. Therefore it is crucial to evaluate the net achievable throughput after RTL synthesis and optimization.

V. RLL Encoder Implementation and Throughput Analysis

To evaluate the achievable throughput, hardware designs were implemented as outlined in Section IV and synthesized for GF 22 nm FDX technology (0.8 V, 25 °C) using the Synopsys Design Compiler. The implementation includes all state machines for $d \in \{1,2,3,4\}$ as described in [8]. Additionally, the state merging approach discussed in Section IV-C was applied for composite state levels $L \in \{2,3,4\}$, with $L = 1$ (unmerged) serving as the baseline.

Fig. 4a shows the maximum synthesized f_{CLK} for the various FSM configurations. As expected, the frequency decreases with increasing logic complexity, primarily due to the extended critical path in the feedback loop. However, the decrease is smaller than the factor L.

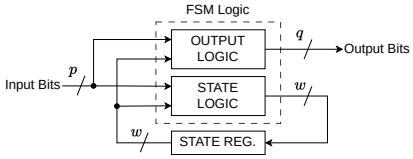
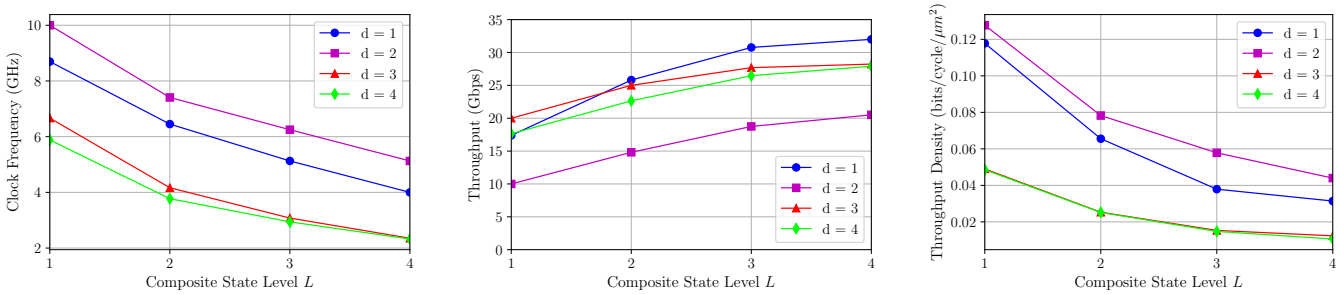


Fig. 3: Block diagram of the RLL FSM

© 2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

This document is a preprint of: D. Swist, S. Friedrich, E. Matus et al., "Implementation Aspects of RLL Encoding for Zero-Crossing Modulation based Wideband Communications," in Proceedings of International Conference on Mobile and Miniaturized Terahertz Systems (ICM2TS 2025), Dubai, United Arab Emirates, Feb 2025. Invited.



(a) Maximum synthesized clock frequencies (b) Throughput at the encoder input (c) Throughput density

Fig. 4: Analysis of different FSM configurations after synthesis

A. Throughput and Area Analysis

Despite the reduced clock frequency, the overall throughput increases with an increasing composite state level L, as shown in Fig. 4b. This is due to the scaling of the number of bits processed per cycle, which compensates for the smaller f_{CLK}.

The increase in the composite state level L also results in greater area usage. Fig. 4c shows the throughput density, defined as the number of bits processed per cycle per unit area. It is evident that throughput density declines as the number of merged states increases. Further, a worst case power estimation after synthesis was done. Depending on the composite state level L, the power consumption is within a range of $[0.4; 1.4]\,\mathrm{mW}$. For the highest level $L = 4$, the designs consume $\{0.8; 0.6; 0.9; 1.0\}\,\mathrm{mW}$ for $d = \{1; 2; 3; 4\}$ (Table II). A detailed workload dependent power analysis is outside the scope of this work and will require further investigation.

B. Technology Scaling Impact

The expected throughput for different technology nodes was also estimated based on the scaling model in [15]. By converting the maximum delay of the synthesized circuits into clock frequency, the scaled clock frequency can be determined as

$$f_y = \frac{\mathrm{DelayFactor}_x}{\mathrm{DelayFactor}_y} f_x \tag{3}$$

where f_y is the clock rate at the target technology node and f_x is the clock rate at the reference node. The technology dependent delay factors are provided in [15]. We took the 20 nm reference node in [15] as it comes closest to the 22 nm used. Fig. 5 compares the scaled throughput of the synthesized results, showing that significant performance gains can be achieved, with some FSMs potentially reaching 100 Gbps.

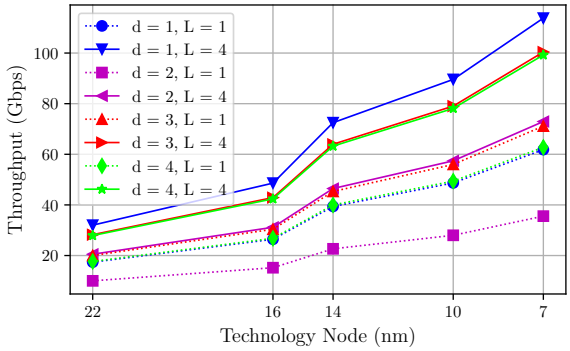


Fig. 5: Throughput at the encoder input estimated by scaling for different technology nodes (all at 0.8 V) compared to synthesis results

VI. System-Level Architecture

As Fig. 5 shows, using a single RLL FSM poses a significant challenge for achieving data rates of 100 Gbps or beyond. While the concept of RLL encoding described in [8] assumes continuous processing of the input bit sequences, in the following we limit the sequence length, i.e., consider a block processing. While this approach leads to a small reduction in the achievable spectral efficiency, it allows the parallelization of RLL encoding in separate blocks. The approach of parallel streams is commonly used in hardware architectures for optical transceivers to achieve high throughput. The proposed system-level architecture is shown in Fig. 6.

A. Parallel Processing for Higher Throughput

To achieve higher throughput, P instances of the composite state RLL FSM operate in parallel. For this purpose, the input bits are demultiplexed into P parallel streams, each with a bit width of $L \cdot p$, corresponding to the bit width of the individual RLL FSMs. Each of these RLL FSMs generates RLL sequences of length $D \cdot L \cdot q$ bits which are stored in individual shift registers. These RLL sequences are subsequently concatenated in a latch register. When concatenating independently encoded RLL sequences, the d-constraint may be violated at the block boundaries, potentially causing zero-crossings to be too close together to be resolved after transmission over a bandlimited channel. To ensure the minimum runlength constraint, d zeros are appended at the end of each of the individual RLL sequences. Therefore, the shift register outputs are latched with d zeros between them. The demultiplexer then serializes the encoded sequences into a single bitstream. As discussed in Section IV, we propose integrating NRZI encoding into the 1-bit DAC, implemented using specialized high-speed technology as an optimized 1-bit DAC module.

B. Impact on Code Rate

The zero-padding between the individual RLL blocks leads to a reduction in the encoding rate depending on the parameter D, resulting in the effective ZXM code rate

$$r_{\mathrm{impl}} = \frac{L \cdot D \cdot p}{L \cdot D \cdot q + d} = r_{\mathrm{RLL}} \frac{L \cdot D}{L \cdot D + \frac{d}{q}} \tag{4}$$

© 2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

This document is a preprint of: D. Swist, S. Friedrich, E. Matus et al., "Implementation Aspects of RLL Encoding for Zero-Crossing Modulation based Wideband Communications," in Proceedings of International Conference on Mobile and Miniaturized Terahertz Systems (ICM2TS 2025), Dubai, United Arab Emirates, Feb 2025. Invited.

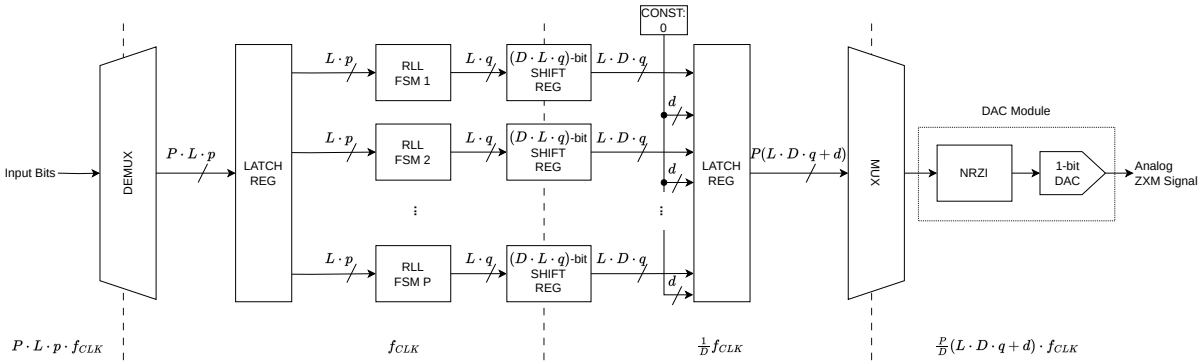


Fig. 6: System-level RLL encoder architecture for high throughput

i.e., due to the parallelization the code rate of the ZXM code is reduced by the factor $r_{\text{impl}}/r_{\text{RLL}}$. To maintain the target bit rates, the signaling rate of the DAC and the sampling rate of the ADC must be increased by the inverse of this factor.

C. Performance Evaluation

Table II shows exemplary results on the estimated performance of a system using $P = 4$ parallel encoder chains for the RLL codes with different d, with optimized state machines. Here we choose $D = 20$, yielding a minimal reduction in the effective code rate while using an appropriate amount of registers. For the most area-demanding case of $d = 4$, the entire shift register stage and the final latch register each require approximately $2.6\,\text{kbit}$. The results demonstrate that throughputs of up to $455\,\text{Gbps}$ can be realized.

Moreover, regarding the example of board-to-board communication inside a server stated in Section III with a required data rate of $100\,\text{Gbps}$, the results in Table II show that the RLL encoder can even be realized in $22\,\text{nm}$ technology using the RLL code with $d = 3$, which provides sufficient spectral efficiency, see [8, Fig. 10], including some overhead for channel coding.

VII. Conclusion

This work presents an analysis of RLL encoding for THz communications systems with receivers using 1-bit quantization, focusing on implementation challenges and throughput optimization. By synthesizing RLL encoders in GF 22 nm FDX CMOS technology and employing architectural strategies such as parallel processing and state merging, we demonstrated that data rates in the range of $100\,\text{Gbps}$ can be achieved. Additionally, we analyzed the effects of technology scaling, highlighting the potential for even higher throughput using advanced nodes such as 7 nm. Future work will focus on analyzing and optimizing power consumption, and exploring implementation strategies for the decoding of ZXM sequences being computationally even more demanding.

TABLE II: Estimated performance of a system with $P = 4$ parallel encoder chains ($L = 4$ and $D = 20$) for different FSM configurations. Power and area given per individual synthesized RLL FSM.

d	Throughput 22nm / 7nm (Gbps)	$r_{\text{impl}}/r_{\text{RLL}}$ (%)	Clock Rate Increase (%)	Power @22nm (µW)	Area @22nm (μm^2)
1	128.00 / 455.20	99.59	100.42	771	64
2	82.05 / 291.79	98.77	101.25	632	23
3	112.94 / 401.64	99.47	100.54	921	242
4	111.63 / 396.97	99.38	100.63	996	282

References

[1] S. Dang, O. Amin, B. Shihada, and M.-S. Alouini, "What should 6g be?" *Nature Electronics*, vol. 3, no. 1, p. 20–29, Jan. 2020.

[2] Z. Chen, C. Han, Y. Wu, L. Li, C. Huang, Z. Zhang, G. Wang, and W. Tong, "Terahertz wireless communications for 2030 and beyond: A cutting-edge frontier," *IEEE Commun. Mag.*, vol. 59, no. 11, pp. 66–72, 2021.

[3] C. Castro, R. Elschner, T. Merkle, C. Schubert, and R. Freund, "Experimental demonstrations of high-capacity THz-wireless transmission systems for beyond 5g," *IEEE Commun. Mag.*, vol. 58, no. 11, pp. 41–47, 2020.

[4] B. Murmann, "Energy limits in A/D converters," in *Proc. 2013 IEEE Faible Tension Faible Consommation*, Paris, France, Jun. 2013, pp. 1–4.

[5] ——, "ADC performance survey 1997-2024," https://github.com/bmurmann/ADC-survey.

[6] G. P. Fettweis, M. Dörpinghaus, S. Bender, L. T. N. Landau, P. Neuhaus, and M. Schlüter, "Zero crossing modulation for communication with temporally oversampled 1-bit quantization," in *Proc. 53rd Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, USA, Nov. 2019, pp. 207–214.

[7] L. T. N. Landau, M. Dörpinghaus, and G. P. Fettweis, "1-bit quantization and oversampling at the receiver: Sequence-based communication." *EURASIP J. Wireless Comm. and Networking*, vol. 2018, p. 83, 2018.

[8] P. Neuhaus, M. Dörpinghaus, and G. Fettweis, "Zero-crossing modulation for wideband systems employing 1-bit quantization and temporal oversampling: Transceiver design and performance evaluation," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 1915–1934, 2021.

[9] G. Fettweis and H. Meyr, "High-speed parallel Viterbi decoding: algorithm and VLSI-architecture," *IEEE Commun. Mag.*, vol. 29, no. 5, pp. 46–55, 1991.

[10] A. S. Sedra and K. C. Smith, *Microelectronic Circuits*, 7th ed. New York, NY: Oxford University Press, 2015.

[11] A. Tang, N. Chahat, and E. Decrossas, "CMOS THz communication links for wireless applications: Where do they fit into mobile access and fixed access?" in *Proc. 2014 39th Int. Conf. Infrared, Millimeter, and Terahertz waves (IRMMW-THz)*, Tuscon, AZ, USA, Sep. 2014.

[12] K. Schouhamer Immink, *Codes for Mass Data Storage Systems*. Shannon Foundation Publishers, 2004.

[13] G. P. Fettweis, M. Dörpinghaus, J. Castrillon, A. Kumar, C. Baier, K. Bock, F. Ellinger, A. Fery, F. H. P. Fitzek, H. Härtig, K. Jamshidi, T. Kissinger, W. Lehner, M. Mertig, W. E. Nagel, G. T. Nguyen, D. Plettemeier, M. Schröter, and T. Strufe, "Architecture and advanced electronics pathways toward highly adaptive energy-efficient computing," *Proc. IEEE*, vol. 107, no. 1, pp. 204–231, 2019.

[14] H. Blume, H. Hubert, H. Feldkamper, and T. Noll, "Model-based exploration of the design space for heterogeneous systems on chip," in *Proc. IEEE Int. Conf. Application- Specific Systems, Architectures, and Processors*, San Jose, CA, USA, Jul. 2002, pp. 29–40.

[15] A. Stillmaker and B. Baas, "Scaling equations for the accurate prediction of CMOS device performance from 180nm to 7nm," *Integration*, vol. 58, pp. 74–81, 2017.

© 2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.