Variable-Rate VLSI Architecture for 400-Gb/s Hard-Decision Product Decoder

N.B. When citing this work, cite the original published paper.

(article starts on next page)

Variable-Rate VLSI Architecture for 400-Gb/s Hard-Decision Product Decoder

Vikram Jain, *Student Member, IEEE,* Christoffer Fougstedt, and Per Larsson-Edefors, *Senior Member, IEEE*

Abstract—**Variable-rate transceivers, which adapt to the conditions, will be central to energy-efficient communication. However, fiber-optic communication systems with high bit-rate requirements make design of flexible transceivers challenging, since additional circuits needed to orchestrate the flexibility will increase area and degrade speed. We propose a variable-rate VLSI architecture of a forward error correction (FEC) decoder based on hard-decision product codes. Variable shortening of component codes provides a mechanism by which code rate can be varied, the number of iterations offers a knob to control the coding gain, while a key-equation solver module that can swap between error-locator polynomial coefficients provides a means to change error correction capability. Our evaluations based on 28-nm netlists show that a variable-rate decoder implementation can offer a net coding gain (NCG) range of 9.96–10.38 dB at a post-FEC bit-error rate of 10^{-15}. The decoder achieves throughputs in excess of 400 Gb/s, latencies below 53 ns, and energy efficiencies of 1.14 pJ/bit or less. While the area of the variable-rate decoder is 31% larger than a decoder with a fixed rate, the power dissipation is a mere 5% higher. The variable error correction capability feature increases the NCG range further, to above 10.5 dB, but at a significant area cost.**

I. INTRODUCTION

Fiber-optic communication systems have traditionally been designed to transmit data at one fixed, maximal information bit rate. This entailed the use of fixed modulation schemes, fixed transmission power, and fixed forward error correction (FEC) codes, designed for the worst-case optical path. But since many paths in optical networks have lengths shorter than that of the worst case [1], this conservative design approach underutilizes the network's capacity. With the goal of increasing the spectral efficiency in optical networks, the idea of elastic optical networks (EONs) [2] emerged with the spectrum-sliced elastic optical path (SLICE) approach proposed in 2008 [3]. EONs have become important insofar as they allow for elastic provision of bandwidth, to maintain high spectral efficiency under varying traffic demands.

While an increased spectral efficiency has been the focus of EONs, it is possible to harness the varying channel conditions

and traffic demands to reduce energy dissipation. Several studies to reduce energy dissipation at the network level exist, e.g., early work where sleep cycles in network nodes have been introduced [4] and later work where traffic prediction has been used improve energy efficiency in software-defined networks (SDNs) [5]. However, studies on how we can harness the traffic and channel variations to improve the energy efficiency of the physical layer, including transceiver digital signal processing (DSP) and FEC circuits, are largely missing. Rasmussen et al. suggested a rate-adaptive FEC scheme [6] which they claim can give power reductions of up to 75 % by reducing code rate during periods of low traffic demands. The work of Rasmussen et al. considers varying the number of decoding iterations to regulate the performance and power dissipation of FEC, however, VLSI aspects are not considered.

In parallel to the development of flexible optical networks, the quest for higher capacity has lead to spectrally-efficient coherent technology being used in the physical layer. Gradually this technology is being adopted for shorter distances, such as passive optical networks [7] and datacenter interconnects [8], [9], however, it has the drawback that it requires transceivers with complex DSP and FEC. This means that operating coherent technology at the highest possible bit rate becomes costly from a DSP and FEC power dissipation perspective. In this context, it would be beneficial to adapt transmission parameters to the channel conditions: For example, if the channel conditions are benign, we can lower the FEC code overhead, effectively increasing the code rate and improving the energy efficiency of DSP and FEC circuits.

A rate-adaptive transceiver has the ability to adapt its bit rate to the current channel conditions in order to maximize the spectral efficiency [10]. Approaches used to enable transceiver flexibility include constellation shaping, time-domain hybrid modulation formats, and variable-rate FEC codes [11]. Ghazisaeidi et al. showed that increasing the number of different available FEC code rates can significantly help in maximizing the total capacity for different fiber distances [12]. Their use of 52 different FEC code rates, however, raises the question on how to practically design variable-rate FEC circuits. Having one decoder unit for each supported code [12], [13] would cause transceiver cost to increase significantly. Rather than having an area-wasting replication of several fixed-rate decoders, we can instead consider one variable-rate reconfigurable FEC VLSI architecture.

Rate-adaptive coding schemes have primarily been designed with soft-decision (SD) decoding [14]–[18], which is known for its high coding gain. Optical communication, however, has very stringent coding gain requirements and require codes with

very large block lengths, which makes the corresponding SD decoding VLSI architectures complex. While SD decoders for the 400G standard [19] have been demonstrated [20], hardly any throughput or latency margins are left to introduce circuits to handle reconfiguration of code rate. For other application areas, such as wireless systems, SD decoding throughput can be substantially higher since unrolling is a practical option [21], [22]. These throughputs can be achieved because post-FEC bit-error rates (BERs) as low as 10^{-15} are not targeted.

Concatenated schemes, using a combination of an inner and an outer code, have been introduced to balance coding gain and complexity: Previous work in this area includes the concatenation of variable-rate outer Reed-Solomon HD codes and an inner repetition code using soft combining for further rate variation [23] as well as the concatenation of a fixed-rate outer HD staircase code and an inner polar code, whose short block lengths can make implementation of variable-rate circuit features practical [24].

Here we present the VLSI architecture of a FEC decoder, which supports several different code rates. The FEC decoder is based on HD product decoding, which is amenable to high-throughput implementations [25] and can thus sustain very high bit rates, making it suitable for high-capacity coherent technology, and very low latencies, making it suitable for optical networks. While the presented VLSI architecture in itself supports several different modes, nothing precludes it from being used in a concatenated scheme. In addition to the FEC decoder handling variable code rates, which was initially presented at ICECS'19 [26], we will also introduce and discuss a decoder whose error-correction capability can be varied.

The VLSI decoder implementations that we will demonstrate use shortening of component codes, varying decoding iterations and varying error-correction capability (t) to support code overheads in a range from 21.9 to 49.0 %. While the overheads and iterations were selected taking throughput for 400 Gb/s and above optical systems into consideration (note that overheads beyond 60 % provide diminishing results in terms of coding gain [27]), our design approach can be extended to other decoder configurations as well.

Section II reviews product codes and decoders based on BCH component codes. We present the variable-rate product decoder architecture in Section III and the variable-rate, variable-t product decoder in Section IV. Section V describes features of a multi-rate decoder that can operate in excess of the 400 Gb/s, whereas Section VI explains our evaluation strategy. Finally, results are given in Section VII.

II. BACKGROUND

A. BCH Component Codes

Bose-Chaudhuri-Hocquenghem (BCH) codes [28] are a class of random error-correcting cyclic codes expressed by the set of parameters BCH(n,k,t), where n is the block length, k is the number of information bits, and t is the number of errors that can be corrected. Primitive narrow-sense binary BCH codes are defined using a primitive element α of a Galois field, $GF(2^m)$, where m is a positive integer. For these codes, parameters are related as $n = 2^m - 1$ and $n - k = m \cdot t$. Other

important parameters include code rate, defined as the ratio of number of information bits to the number of bits in the code block, $R = \frac{k}{n}$, and the overhead expressed as $\text{OH} = \frac{n}{k} - 1$.

B. Product Codes

A *product* code—a concept of combining smaller *component* codes [29] to form codes that can provide higher error-correction capability—is constructed by encoding information bits row wise, using a row component code, followed by column wise encoding, using column component codes, as shown in Fig. 1a. The minimum distance that results from the twofold encoding over both data and parity is the product of the minimum distances of the component codes.
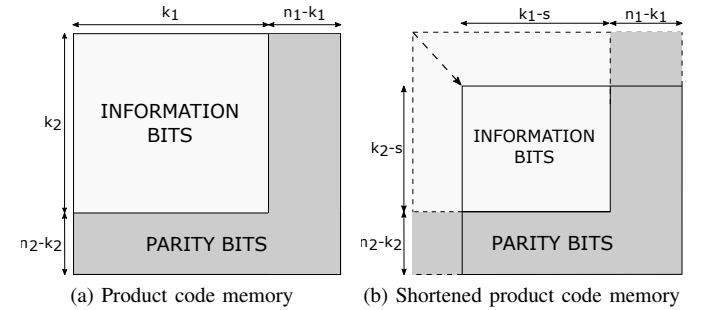

Fig. 1. Product code memory [26].

(a) Product code memory
(b) Shortened product code memory

Product decoding can be implemented using a product code memory which is iteratively decoded by component decoders. Incoming data bits are loaded into the memory after which decoding and error correction of row and column data take place. This process is repeated for a number of decoding iterations. Employing low-complexity component codes can limit the complexity of the product decoder. Thus, component code selection plays a vital role not only for the error correction performance, but also for the speed and complexity of the product decoder.

C. Varying Product Code Overhead

By increasing the code overhead, the coding gain—the improvement in signal-to-noise ratio (SNR) over an uncoded transmission for a certain input (pre-FEC) BER—can be improved. This means that a higher code overhead allows the FEC decoder to maintain a certain target post-FEC BER even when the pre-FEC BER is increased, but this comes at the expense of a reduced information throughput.

The code overhead of component codes is commonly varied by using one of two methods; shortening or puncturing. Puncturing is the process of removing parity bits and substituting erasures, whereas shortening is the process of substituting zeros for some information bit positions at the encoder (these bits are then never transmitted). Both shortening and puncturing result in an increased overhead from the original code—the mother code with its base OH—increasing the coding gain at the cost of throughput. Since puncturing requires more complex error-erasure component decoders, puncturing for increasing code rates [23] is not explored in our work, but we
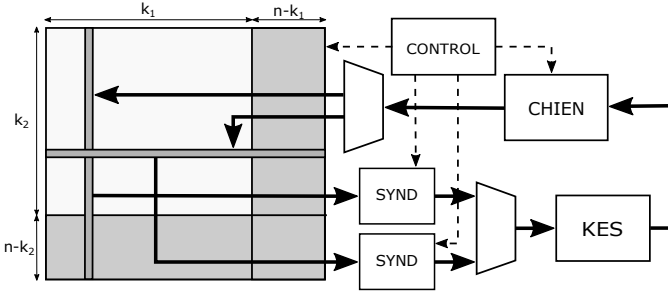
Fig. 2. Block diagram of our variable-rate product decoder (VRPD) [26].

Fig. 3. Product code memory array with reconfigurable logic [26].

use only shortened codes, denoted $n_s = n - s$ and $k_s = k - s$, where s is the number of bits shortened.

A product code can be designed by concatenating two component codes, $\text{BCH}(n_1, k_1, t_1)$ and $\text{BCH}(n_2, k_2, t_2)$. The product code formed is a $n_1 \times n_2$ matrix, with information bits forming a $k_1 \times k_2$ matrix inside it, as shown in Fig. 1a. The code rate of the resulting product code is the product of the code rate of individual component codes, $R = R_1 \cdot R_2 = \frac{k_1}{n_1} \cdot \frac{k_2}{n_2}$, the overhead is $\text{OH} = \text{OH}_1 \cdot \text{OH}_2 = \frac{n_1 \cdot n_2}{k_1 \cdot k_2} - 1$, and the error-correction capability is $t_1 \cdot t_2$. When using shortened BCH codes to construct the product code, the memory becomes a $(n_1 - s) \times (n_2 - s)$ matrix as the enclosed information is reduced to a $(k_1 - s) \times (k_2 - s)$ matrix, as shown in Fig. 1b.

III. VARIABLE-RATE PRODUCT DECODER (VRPD)

We will now review the variable-rate product decoder (VRPD) architecture [26]. A simplified high-level VRPD block diagram is shown in Fig. 2. At the heart of the VRPD design, we can find the baseline fixed-rate product decoder that was previously published [25]. The VRPD architecture consists of four major modules: the SYND (syndrome calculation) module, the KES (key-equation solver) module, the CHIEN (Chien search) module, and the CONTROL (control state-machine) module. The SYND module calculates syndromes, which flag the presence of errors in a received codeword. These syndromes form a system of linear equations which are then solved by the KES and the CHIEN module to find the error location. Finally, the CONTROL module configures the decoder to one of the several modes of operation; modes which are based on varying the code overhead and the number of decoding iterations. (As we will see in our extended design architecture in Section IV, the modes will also incorporate variation of the error-correction capability.)

A. Product Code Memory

At the heart of the product decoder is the product code memory. The product code memory is implemented as a 2D array of $n \times n$. The data received by the decoder is stored into the memory array column-wise until the entire array is filled up. The implementation of the product code memory uses flip-flops instead of SRAM, as these provide more flexibility for row-wise and column-wise reads which are required during the decoding.

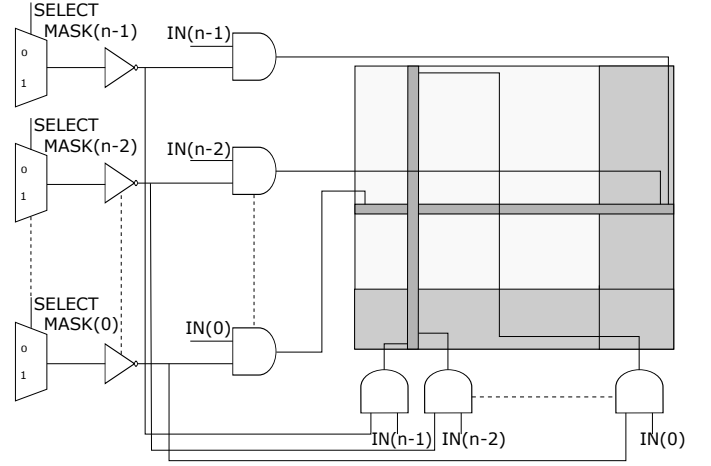To support the operation of the decoder in the variable-rate modes, shortening is applied by substituting zeros in the incoming codeword. Shortening reduces the size of the useful data in the memory array as shown in Fig. 3. In a fixed-rate decoder architecture, the unused section of the memory array is either completely removed or left unaltered. However, in order to support variable rates, the memory array has to be at its original dimensions. Moreover, to prevent any kind of interference with the decoding operation, it becomes obligatory to flush and gate the shortened bits of the memory array. Gating the unused bits enhances the energy efficiency as unwanted switching activity is avoided at these bit locations.

Gating is achieved by masking of the shortened bits by utilizing a mask of size n. The bits of the mask are set to either 1 or 0 based on size of the shortening selected or the mode of decoder configured. When the data arrives at the product decoder, it is first ANDed with this bit mask and then stored in to the memory array. This forces the shortened part of the memory to be zero and thus prevents toggling of the flip-flops. Another advantage of the gating is that it flushes the shortened part of the memory array. This prevents retention of any unwanted data that may cause interference in the decoding particularly in the case of switching between a "low shortening" mode to "high shortening" mode.

B. Syndrome Calculation

The SYND module calculates syndromes by implementing the vector-matrix multiplication, $S = u \cdot H^T$, where S is the syndrome vector which consists of $2 \cdot m \cdot t$ elements, m is a positive integer used to represent the Galois field, $GF(2^m)$, u is the incoming codeword, and H^T is the transpose of the parity check matrix. The given operation is performed in the Galois field of $GF(2^m)$ as modulo-2 arithmetic, which transforms addition and multiplication as XOR and AND operations, respectively. The vector-matrix multiplication can be simplified to an XOR tree, in which each syndrome element becomes a set of bitwise XOR operations of the codeword bits where the parity check matrix elements are 1. In a fixed-rate design, the shortened part of XOR tree in the syndrome calculation module can be removed from hardware.

In the variable-rate decoder, the XOR tree is not pruned in hardware in order to support the different decoder modes.
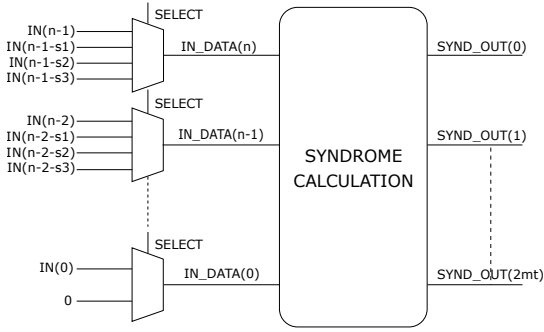
Fig. 4. Syndrome calculation module with reconfigurable logic [26].

Fig. 5. Chien search module with reconfigurable logic [26].

When the codeword is shortened, the SYND module shifts the codeword to the most significant bits (MSBs) using a set of multiplexers as shown in Fig. 4. The lower significant bits (LSBs) equal to the shortened bits are updated to zero. The zeros at the shortened bits prevent any switching activities at these positions. The shifted bits are then passed to the original XOR tree to generate the $2 \cdot m \cdot t$ syndromes. These syndromes are used in the KES module to calculate the coefficients of an error-locator polynomial.

C. Chien Search

The CHIEN module implements the Chien search algorithm used to evaluate the roots of the error-locator polynomial generated by the KES module. The KES module uses direct-solution Peterson method [30] to evaluate the coefficients of the error-locator polynomial. In the CHIEN module, primitive element α^i of the Galois field, where i is an integer from 0 to $n-1$, is substituted in the error-locator polynomial. Multiplications in the CHIEN module are done using finite-field multipliers (FFM) and additions are done using modulo-2 arithmetic in GF or XOR. If the value of the polynomial evaluated is zero, it represents an error at the bit position i.

To support high throughput, the CHIEN module is implemented as a fully unrolled set of FFMs. (While other less area-consuming schemes to identify roots of error-locator polynomials have been proposed [31], [32], they result in longer timing paths which would need to be pipelined to sustain throughput. This would significantly increase clock and register power dissipation. In addition, latency would increase, unless the clock rate is significantly raised.) The module requires $n \cdot t$ FFMs for each of the n component decoders. In a fixed-rate design, the FFMs corresponding to the shortened part are removed. However, to support all the decoder modes, the entire hardware architecture of the CHIEN module is retained. The large number of FFMs increases the power dissipation and requires extra circuitry for gating the unwanted computation.

In order to prevent switching activity at the shortened part of the codeword, the coefficients received from the KES module are ANDed with enable signals for the bits at which the evaluation of the error-locator polynomial are not required. The enable signals are selected based on the mode selected in the CONTROL module. The gated coefficients are then passed to the original CHIEN module and the resulting error signal
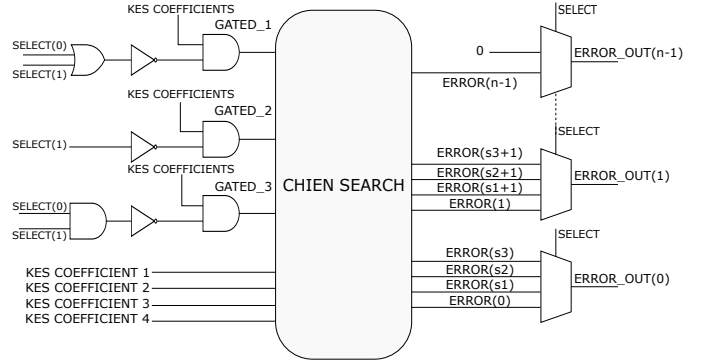
is shifted back to the LSB by using a set of multiplexers as shown in Fig. 5.

IV. VARIABLE-RATE VARIABLE-t PRODUCT DECODER

This section describes the exploration into a variable-rate, variable-t architecture (VRVTPD). The VRVTPD architecture adds the ability to vary the error-correction capability (t) to broaden the achievable range of coding gains of the original VRPD architecture, which had a fixed $t = 3$. As mentioned in Section II, the error-correction capability of a component code represents the number of errors that the decoder can correct. Thus, increasing the t can result in an improvement in coding gain. However, increasing the t also requires additional hardware, both in the base modules of the product decoder as well in the hardware to support the configurability, and leads to larger area and larger power dissipation. The VRVTPD design is capable of varying code rate, decoding iteration and t between three and four.

When $t = 4$, the OHs for the component codes are different as compared to $t = 3$, owing to a change in the base OH of component codes when t changes. In order to support the variation in the OHs between $t = 3$ and $t = 4$, extra hardware has to be included in the modules of the VRPD architecture. In addition, the KES module of the VRPD design, which was identical to the baseline fixed-rate decoder, now also has to be modified to support the variable t.

A. Product Code Memory

The product code memory for the VRVTPD design requires similar mask generation as the VRPD's product code memory as shown in Fig. 6. Since the selected OHs are not the same for $t = 3$ and $t = 4$ modes, an additional set of multiplexers is added to the architecture as the existing hardware of the VRPD design cannot generate the required masks for the OHs of $t = 4$ mode. With the addition of these multiplexers, the design can generate masks for the selected OHs in $t = 3$ and $t = 4$ modes. At the output of the mask generation multiplexers, another set of multiplexers is added to select masks between $t = 3$ and $t = 4$ modes. An example of operation of the multiplexers to select the correct mask is when switching between $t = 3$ and mode 2 of 25.0 % OH, to $t = 4$ and mode 2 of 33.3 % OH. In the first case, 28 bits are shortened and a mask of upper 28
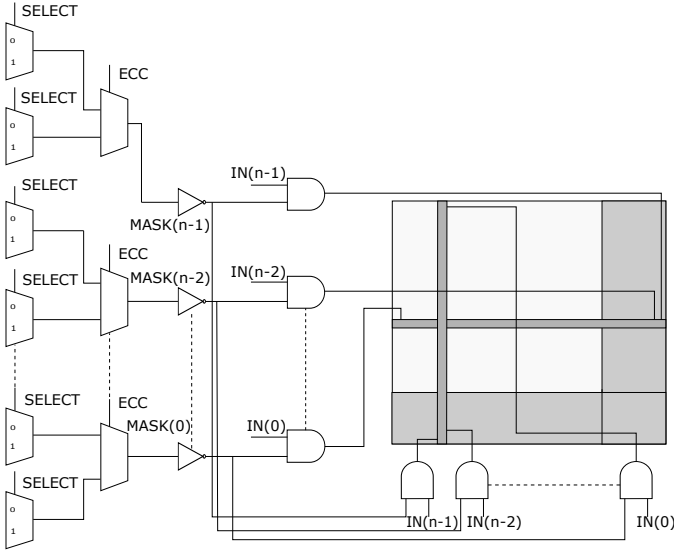
Fig. 6. Product code memory for VRVTPD.

Fig. 7. Syndrome calculation module for VRVTPD.

bits is generated by the multiplexers. In the second case, 16 bits are shortened and a mask of upper 16 bits is generated. When switching between the two modes of t, the ECC signal sets the output multiplexers to select between the two masks for correct operation of the decoder.

B. Syndrome Calculation

As stated before, due to the difference in the OHs selected between $t = 3$ and $t = 4$, two sets of multiplexers, one for $t = 3$ and another for $t = 4$, are used to shift the codeword to the MSB depending on the OH selected. Another multiplexer is placed at the output of the previous multiplexers to select the shifted codeword corresponding to the t mode as shown in Fig. 7. In the syndrome calculation module, the number of syndromes calculated depends on the t selected as the number of syndromes calculated is equal to $2 \cdot m \cdot t$. Thus, when $t = 4$, more syndromes are calculated than when $t = 3$. When $t = 3$, the additional syndromes calculated should be avoided to prevent unwanted compute and power dissipation. Therefore, another multiplexer is placed before the syndrome calculation, which sets the input to the SYND module corresponding to the $t = 4$ syndromes to zero. Finally, the shifted codeword is then forwarded to the original syndrome calculation module.

C. Key Equation Solver

When $t = 4$, the coefficients generated by the KES module are completely different from the coefficients for $t = 3$ as shown in Eq. 1 and Eq. 2.

$$
\begin{aligned}
\Lambda_0 &= S_1^3 + S_3 \\
\Lambda_1 &= \Lambda_0 S_1 \\
\Lambda_2 &= S_1^2 S_3 + S_5 \\
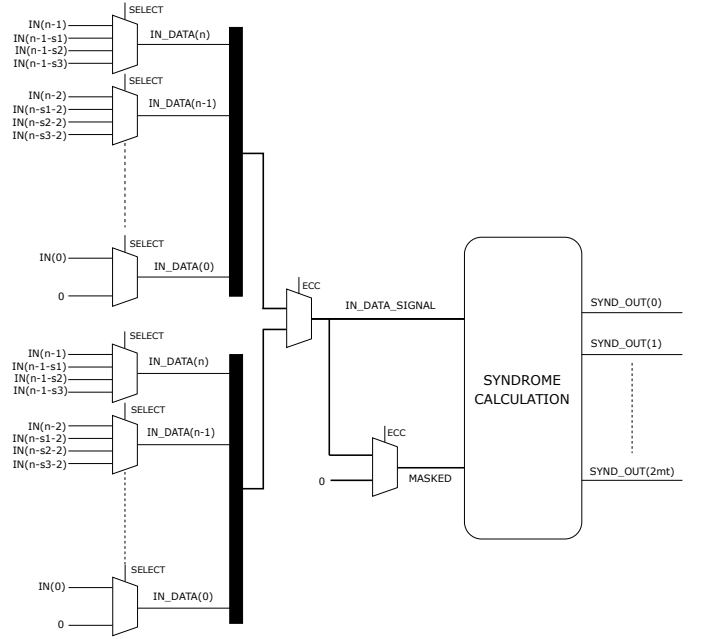\Lambda_3 &= \Lambda_0^2 + S_1 \Lambda_2.
\end{aligned}
\tag{1}
$$

$$
\begin{aligned}
\Lambda_0 &= S_1(S_1^5 + S_5) + S_3(S_1^3 + S_3) \\
\Lambda_1 &= S_1 \Lambda_0 \\
\Lambda_2 &= S_1(S_1^7 + S_7) + S_3(S_1^5 + S_5) \\
\Lambda_3 &= S_1^4 S_5 + S_1^2 S_7 + S_3(S_1^6 + S_3^2) \\
\Lambda_4 &= S_1^3(S_1^7 + S_7) + S_3(S_1^7 + S_1 S_3^2 + S_7) \\
&\quad + S_5(S_1^5 + S_1^2 S_3 + S_5).
\end{aligned}
\tag{2}
$$

These equations are derived from the direct-solution Peterson method for $t = 3$ and $t = 4$, as shown in [25].

Since there is limited functional similarity between the two required KES modules, FFMs for both modules necessary for generating coefficients for $t = 3$ and $t = 4$ have to be included in the VRVTPD design. Here, only the FFMs generating identical terms of the coefficient generation equations were reused, e.g., S_1^3 can be used in both modes. The variable-t KES module was implemented such that any unwanted multiplication was avoided when switching between ts. As shown in Fig. 8, a multiplexer driven by the ECC signal sets the input to the FFMs, which are computing coefficients for $t = 4$ to zero when operating in $t = 3$ mode. This gates the $t = 4$ coefficient generation hardware not being used and reduces power dissipation. The terms generated by the FFMs for $t = 3$ are then used in its coefficient generation module, where an XOR operation is used when a summation is required. When operating with $t = 4$, however, the FFMs for both $t = 3$ and $t = 4$ receive their respective syndromes and the terms which are common between them are generated by the FFMs for $t = 3$. The terms from both the ECC FFMs are then used in the coefficient generation module for $t = 4$ to generate its coefficients. Another set of multiplexers at the output selects the relevant coefficient value based on the ECC signal.
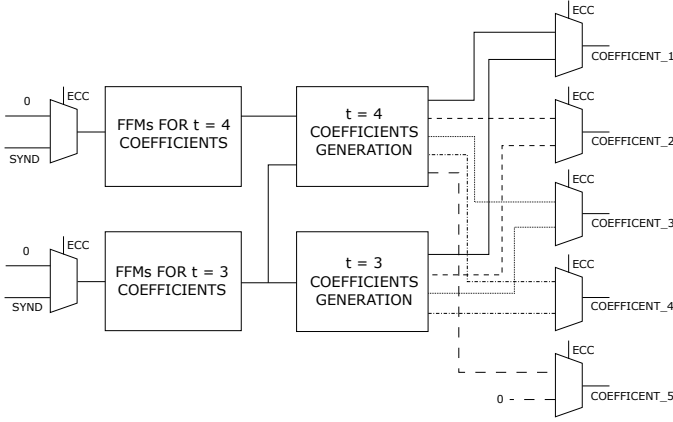
Fig. 8. Variable-t KES module. Different stroke styles of the wires represent pair of wires connected to the same multiplexer.

TABLE I
PRODUCT AND COMPONENT CODE PARAMETERS FOR $t = 3$

Product code overhead	n	k
Base OH = 21.9 %	255	231
OH = 25.0 %	227	203
OH = 33.1 %	180	156
OH = 40.0 %	155	131

TABLE II
PRODUCT AND COMPONENT CODE PARAMETERS FOR $t = 4$

Product code overhead	n	k
Base OH = 30.8 %	255	223
OH = 33.3 %	239	207
OH = 41.2 %	202	170
OH = 49.0 %	177	145

D. Chien Search

Similar to the other modules, the CHIEN module of the VRVTPD design requires additional hardware to operate with different OHs in the $t = 3$ and $t = 4$ modes. A high-level block diagram of the implemented CHIEN module in the VRVTPD design is shown in Fig. 9. The coefficients of the error-locator polynomial coming from KES module are required to be gated at the shortened bit positions. However, since the OHs for $t = 3$ and $t = 4$ are different from one another, the shortened bit positions vary between the two and the same gating mechanism from VRPD design cannot be reused. An additional set of multiplexers (MUXES) are added at the output of the generated gating signals and depending on the OH selected in the different t mode the gating is applied to the shortened bits. An example of the operation of these muxes can be given when switching between $t = 3$ and mode 2 of 25.0 % OH to $t = 4$ and mode 2 of 33.3 %. In the first case, 28 bits are shortened and need to be gated, but in the second case, 16 bits are shortened and need to be gated. The muxes are used such that at the overlapping bits (17 to 28) GATED_2 is applied when in $t = 4$ mode and GATED_1 is applied when in $t = 3$ mode. The gated coefficients are then used in the CHIEN module to generate the error signals for both the ts. The error signals generated are then shifted to the LSB using multiplexers at the output of the CHIEN module and then, based on the ECC, the final error signal is selected.

V. DESIGN PARAMETER EXPLORATION

The design target for this work was to achieve a product decoder which can provide high coding gain at a throughput of 400 Gb/s and above [19]. The design choice that met these requirements was a product code with BCH(255, 231) as component codes. For the t value, we confined our design to $t = 3$ and $t = 4$, as these t values have been shown to achieve coding gains above 10 dB [33]. The modes of the variable rate decoder are based on selection of the overheads (OH), the decoding iterations (#IT) and the value of t. Table I and Table II provides a summary of the code and overhead parameters for $t = 3$ and $t = 4$, respectively.

The decoding iterations (#IT) are used to extend the modes of operation of the decoder. In this work, decoding iterations from three to five are selected as these provide good coding gain. The decoding iterations can be increased to achieve higher coding gain, but as the number of decoding iterations increases, the throughput drops. Also, decoding iterations above five yield diminishing returns on the coding gain [33]. A combination of one of the different modes based on OHs and decoding iterations can provide the desired coding gain. For example, combining a high overhead with a larger number of decoding iterations can result in high coding gain.

VI. EVALUATION AND ASIC IMPLEMENTATION

The implementation and evaluation of both our product decoder architectures are carried out in the framework of an application-specific integrated circuit (ASIC). The decoders are implemented in VHDL and Cadence Incisive is used for the functional verification of the designs using behavioral and netlist simulations. Two VHDL testbenches are used; one for the functional verification of the design and another for BER analysis. The first testbench consists of two random number generators (RNG) based on the uniform procedure in VHDL. The first RNG is used to generate uniformly-distributed data which is passed through our product encoder design in order to generate encoded data. The second RNG is used to generate errors or bit flips with a probability corresponding to the provided input SNR. The repetition period of these RNGs is set to approximately 2.3×10^{18} for each set of seed values. The errors generated from the second RNG are added to the encoded data and the resulting data is passed through the decoders to verify their error-correction capability.

The second testbench is used to generate all-zero data streams to which errors are induced using an RNG with a probability corresponding to the input SNR/pre-FEC BER. This data is then passed through the decoders, which try to detect and correct the errors. For a given pre-FEC BER, the testbench continues to run until the output of the decoder reaches 50 uncorrected blocks. The post-FEC BER is calculated by taking the ratio of the number of errors in the decoded blocks to the total number of bits in the decoded blocks. The post-FEC BER
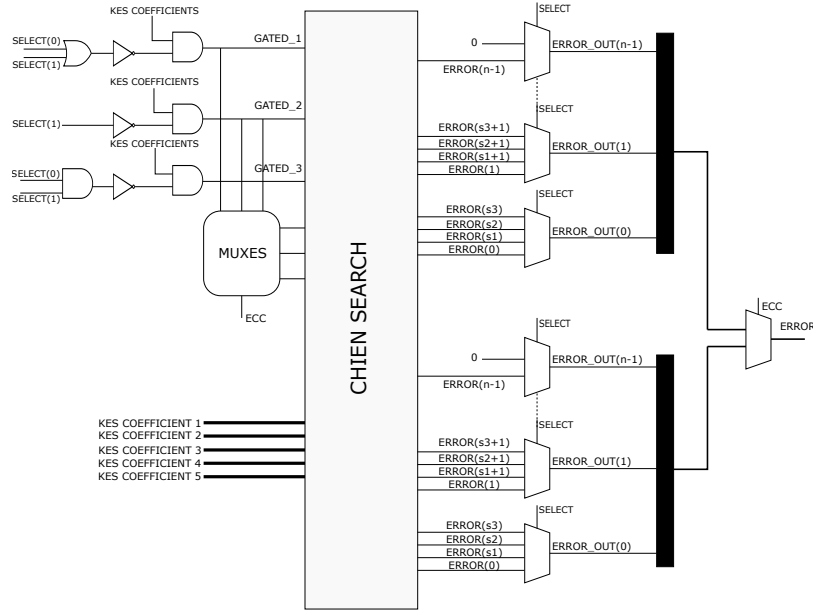
Fig. 9. Chien search for VRVTPD.

is extrapolated using the function *berfit* in MATLAB and a plot for the BER is generated. For optical communication 10^{-15} is considered a common target for post-FEC BER [34] and is used in this work to define net coding gain (NCG). Another important aspect of the BER plot is the error floor. The error floor is the region of operation where the performance of the decoder starts degrading and the BER curve does not follow the waterfall model. In this work, the error floor is estimated using the method proposed by Justesen [35].

The architectures are synthesized in Cadence Genus to a low-leakage library of a 28-nm 0.9-V fully-depleted silicon-on-insulator (FD-SOI) process technology, assuming slow conditions (slow transistor corners and a temperature of 125 °C). Based on an architectural analysis, the target clock rate is set to 610 MHz. Power analysis is performed using a testbench to generate switching activity information, which is then back-annotated to the generated netlist. This analysis is performed using the typical corner at a temperature of 25 °C. (Since low-leakage cells are used, leakage is negligible [25].) In addition, clock-tree power is estimated using Cadence Genus. The power and energy metrics are obtained at the same post-FEC BER used in the NCG analysis.

The complete process of functional verification, BER and power analysis is performed individually for all the modes of operation of the decoders. For the VRVTPD design, the evaluation was limited to functional verification of the design in $t = 3$ and $t = 4$ and the BER analysis was also limited to the base OH in the two t modes.

VII. RESULTS

In this section we describe the results obtained from netlist evaluations performed on the implementations of the proposed decoder VLSI architectures.

A. Reference Designs

The algorithm used in the KES module (Fig. 2) plays an important role in terms of throughput and power dissipation of a decoder. In general, Berlekamp-Massey (BM) and its different optimizations like simplified inverse-free BM (SiBM) [36], [37] are iterative in nature and require at least t clock cycles to compute the error-locator polynomial. This iterative operation has a detrimental effect on the product decoder as it increases latency and lowers throughput, which leads to higher energy per bit. Thus, approaches like Peterson [38] and direct-solution Peterson [30], which is used in the proposed architectures to compute the error-locator polynomial in a single cycle, are important alternatives to the iterative approach.

In our comparison, we use as reference two different fixed-rate product decoder designs: a) FRPD3, which is based on BCH(255, 231, 3), and b) FRPD4, which is based on BCH(255, 223, 4). In addition, we use a decoder (SIBM3) based on BCH(255, 231, 3) which uses the iterative SiBM approach for its KES module. Table III shows the implementation results obtained, using four iterations at the base OHs which are 21.9 % for $t = 3$ and 30.8 % for $t = 4$ (see Table I and Table II). These three implementations are used as reference baselines for the VRPD and VRVTPD implementations.

TABLE III
EVALUATION RESULTS FOR REFERENCE DESIGNS WITH #IT = 4.

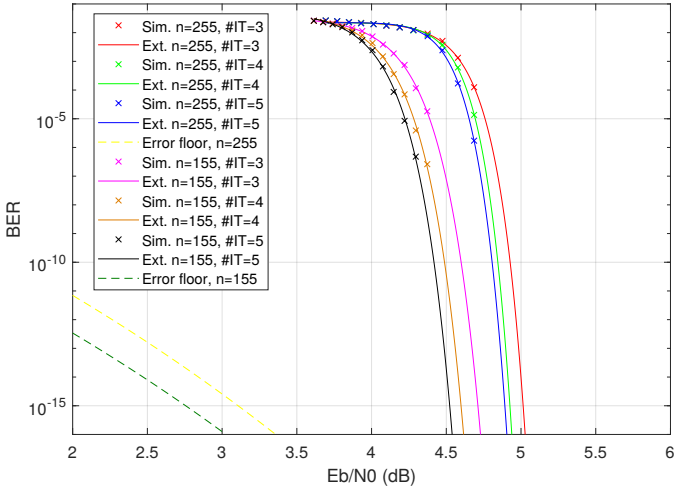|	FRPD3	FRPD4	SiBM3
Cell area (mm^2)	6.69	9.11	7.54
Code rate, R	0.82	0.76	0.82
Throughput (Gb/s)	1252	1167	775
Block decoding latency (ns)	42.61	42.62	68.83
NCG @ BER 10^{-15} (dB)	10.06	10.5	10.08
Power @ BER 10^{-15} (mW)	788.49	1305.56	866.62
Energy @ BER 10^{-15} (pJ/info. bit)	0.63	1.11	1.12

Fig. 10. Output BER as a function of E_b/N_0. Simulated data (Sim.) are used for extrapolation (Ext.) in MATLAB with the *berfit* function. E_b/N_0 for an encoded system stands at 14.99 dB for an output BER of 10^{-15} [26].

The data in Table III confirms two trends: a) The SiBM-based product decoder cannot sustain as high throughput and short latency as those of decoders based on the direct-solution Peterson approach. b) As we increase t, the area and power dissipation of a decoder based on the direct-solution Peterson approach grow very fast, rendering this approach too complex for $t > 4$.

B. Variable-Rate Product Decoder (VRPD)

Fig. 10 shows the output BER as a function of E_b/N_0 for the VRPD decoder modes that represent the net coding gain (NCG) extremes, i.e., 21.9 % and 40.0 % OH. The estimated coding gain ranges are 0.31, 0.33, and 0.38 dB for three, four and five iterations, respectively. A wider range of 0.5 dB can be achieved if the decoder is operated with three iterations for the base OH and with five iterations for 40.0 % OH. However, five iterations with 40.0 % OH cannot attain the targeted throughput of 400 Gb/s. Therefore, our design considers a coding gain range of 0.42 dB obtained from three iterations with base OH to four iterations with 40.0 % OH. Miscorrections are observed when the decoder is exposed to high input SNR, which is an expected outcome for FECs. In addition, the NCGs at a post-FEC BER of 10^{-15}, for different decoder modes, are shown in Table IV.

The coding gain range depends on the block length of the component codes. In this design, the minimum coding gain is limited by the base overhead of the fixed-length mother component code, BCH(255, 231). Conversely, the upper limit of coding gain, i.e., the highest OH, is limited by the constraint of achieving throughputs in excess of 400 Gb/s. The coding range could be extended further by utilizing a higher OH, but this has the consequence of reducing throughput. Another alternative to increase the coding gain range is to utilize longer component codes, e.g., BCH(511, 484), whose product code has a base OH of 11.5 %, but this leads to a more complex decoder with higher energy dissipation. Increasing the error-correction capability t, as suggested in Section IV, can increase

the coding gains of individual modes with a skew in the range. However, the area and power cost of higher t must also be considered. This will be the topic of Section VII-C.

Table IV presents the result of the VRPD implementation. When implementing a decoder having a mother code on top of which a varying overhead is introduced, the support for flexible code overheads increases circuit area and power dissipation [27]. However, the power dissipation of the VRPD decoder operating in the 21.9 % OH mode is only 5 % higher than that of the fixed-rate 21.9 % decoder (FRPD3 in Table III). The area cost for introducing this flexibility is 31 % over the FRPD3 design. Note that this increase in area is for having all the four modes of operation in a single design. If fixed-rate decoders were used, then the number of chips required will be equal to the number of modes of operation and each design has its own area cost. Variation in iteration provides a very resource-efficient alternative to obtain variable coding gains at fixed code rates. However, it cannot provide such a large range in coding gain obtainable by multi-rate design. As shown here, a combination of OH and iteration variation can provide several modes with a wider range of operation.

TABLE IV
EVALUATION RESULTS FOR VRPD DECODERS [26].

	#IT	Overhead (OH)			
		21.9 %	25 %	33.1 %	40 %
Cell area (mm^2)		8.78			
Code rate, R		0.82	0.80	0.75	0.71
Throughput (Gb/s)	3	1628	1257	742	523
	4	1252	967	571	402
	5	1017	785	464	327
Block decoding latency (ns)	3	32.78			
	4	42.61			
	5	52.45			
NCG @ BER 10^{-15} (dB)	3	9.96	10.05	10.16	10.27
	4	10.06	10.14	10.24	10.38
	5	10.08	10.23	10.35	10.46
Power @ BER 10^{-15} (mW)	3	941	822	599	524
	4	830	768	525	461
	5	770	710	479	422
Energy @ BER 10^{-15} (pJ/info-bit)	3	0.58	0.65	0.81	1.00
	4	0.66	0.79	0.92	1.14
	5	0.76	0.90	1.03	1.29

The component decoders are implemented in a fully block-parallel manner, resulting in high-speed block decoding with latency well below 100 ns and estimated throughputs as high as 1.6 Tb/s when operating with three iterations and 21.9 % OH mode. Moreover, owing to the usage of clock gating, the architecture is highly energy efficient with a maximum energy per information bit being as low as 1.29 pJ/bit.

Since errors activate component decoders, the error correcting process over time gradually suppresses power dissipation: The average power dissipation decreases with an increasing number of iterations, since for each iteration fewer and fewer errors remain in memory. On the other hand, the energy per information bit increases, since the loss of throughput caused by an increasing number of iterations dominates the reduced power dissipation.

Table IV also shows that as we reduce the OH, power dissipation increases but energy per bit reduces. This trend is caused by us leveraging all the inherent throughput of the decoders. In principle, we could just as well keep the throughput constant as we vary the OH. This would enable reductions in both power and energy per bit. In this context, if channel conditions are benign, we can lower the FEC code overhead to significantly improve the FEC energy efficiency. In addition, the higher code rate would be highly beneficial to the DSP circuits, which would avoid operating on redundant data [39].

Operating clock rate plays a major role in the selection of iterations and OH for generating modes. In our design, with the operating frequency of 610 MHz and a target throughput of 400 Gb/s, we are limited to using five iterations which in itself is limited to a maximum OH of 33.1 %. The operating clock rate can be increased to 750 MHz to obtain a higher OH at five iterations. But the cost of raising the clock rate from 610 to 750 MHz, however, would be that area usage increases by 20 %.

To the best of our knowledge, with the exception of our previous paper [26], no variable-rate decoder architectures for optical communication are available in the open literature. Compared to a recently published hard-decision fixed-rate product decoder [40], our slowest variable-rate decoder offers four times higher throughput and much lower latency. It is impractical to fully unroll the corresponding soft-decision decoders for applications which require codes with large block lengths. Thus, such turbo product decoders cannot reach as high throughput as hard-decision decoders, but in exchange provide a higher net coding gain [41].

C. Variable-Rate Variable-t Product Decoder (VRVTPD)

The VRVTPD design was evaluated with $t = 3$ and $t = 4$, with #IT = 4, and at the base OH of 21.9 % and 30.8 %, respectively, and the evaluation results are presented in Table V. The VRVTPD design has an overall area of 14 mm^2 and a power dissipation, for four iterations and the base OH, of 1.108 W and 1.687 W for $t = 3$ and $t = 4$, respectively. In terms of energy per information bit, 0.88 and 1.44 pJ/bit was obtained for $t = 3$ and $t = 4$, respectively. At $t = 3$ and 21.9 % OH, the VRVTPD design dissipates 33 % more power than VRPD at the same configuration. This increase in power dissipation can be owed to the increased area and switched capacitance.

The implementation of the decoder modules is largely determined by the critical paths of the entire decoder. As we add features, from the baseline fixed-rate decoder to the VRPD, and from the VRPD decoder to the VRVTPD, the relative circuit complexity and the criticality of the longest paths of the modules change. In the VRPD implementation, the flexible CHIEN module challenges the timing. Specifically, the generation of the error signal, which is used to clear the errors in the product code module, is on the critical path. In the VRVTPD implementation, circuits that support $t = 4$ and the switching between error-correction capabilities are added to the KES module, thus making the KES module more timing critical than the CHIEN module.

The net coding gain (NCG) of the VRVTPD design for $t = 4$, as shown in Table V, is estimated at 10.5 dB. This NCG value is obtained at the base OH and can be improved further by configuring the VRVTPD design to one of its shortened modes (higher OH). Due to the constraint of high simulation time for such high NCG, our results are limited to the base OH. The VRVTPD design can deliver a much higher coding gain range with its variable-rate and variable-t modes. However, the area and power dissipation of the design could be a limiting factor in deployment to real systems. If the requirements of some systems entail the use of a wider range of coding gain and throughput, the VRVTPD design may be considered. But since the area of the VRVTPD design is significantly larger than that of VRPD, concatenated schemes [23], [24] are probably a more viable option.

TABLE V
EVALUATION RESULTS FOR VRVTPD WITH #IT = 4

	t = 3	t = 4
Cell area (mm^2)	14.01	
Overhead	21.9 %	30.8 %
Throughput (Gb/s)	1252	1167
Block decoding latency (ns)	42.61	42.61
NCG @ BER 10^{-15} (dB)	10.06	10.5
Power @ BER 10^{-15} (mW)	1108	1687
Energy @ BER 10^{-15} (pJ/info. bit)	0.88	1.44

VIII. CONCLUSION

In this work, we have introduced VLSI architectures for high-throughput product decoders featuring variable rates and variable error-correction capabilities. The designs are synthesized in a 28-nm technology and are demonstrated to provide an estimated net coding gain range from 9.96 to 10.5 dB. The decoder designs provide a minimum throughput of 400 Gb/s with a maximum decoding latency of 53 ns. By demonstrating these designs, we explore the viability of flexible FEC decoders for energy-efficient high-throughput systems.

In order to introduce flexibility into the decoder designs, extra logic circuits are required to handle the different code rates and the variable error-correction capability. This increases the area of the variable-rate decoder by 31 % compared to the fixed-rate product decoder. Even though the introduction of flexibility results in increased area, replication of several fixed-rate product decoders would be much more area inefficient.

REFERENCES

[1] M. Jinno, "Elastic optical networking: Roles and benefits in beyond 100-Gb/s era," *IEEE J. Lightw. Technol.*, vol. 35, no. 5, pp. 1116–1124, Mar. 2017.

[2] O. Gerstel, M. Jinno, A. Lord, and S. J. B. Yoo, "Elastic optical networking: a new dawn for the optical layer?" *IEEE Commun. Mag.*, vol. 50, no. 2, pp. s12–s20, Feb. 2012.

[3] M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, T. Yoshimatsu, T. Kobayashi, Y. Miyamoto, K. Yonenaga, A. Takada, O. Ishida, and S. Matsuoka, "Demonstration of novel spectrum-efficient elastic optical path network with per-channel variable capacity of 40 Gb/s to over 400 Gb/s," in *Eur. Conf. Opt. Commun. (ECOC)*, Sept. 2008, p. Th.3.F.6.

[4] B. G. Bathula and J. M. H. Elmirghani, "Green networks: Energy efficient design for optical networks," in *IFIP Int. Conf. on Wireless and Optical Communications Networks (WOCN)*, Apr. 2009, pp. 1–5.

[5] Y. Xiong, J. Shi, Y. Yang, Y. Lv, and G. N. Rouskas, "Lightpath management in SDN-based elastic optical networks with power consumption considerations," *IEEE J. Lightw. Technol.*, vol. 36, no. 9, pp. 1650–1660, May 2018.

[6] A. Rasmussen, M. P. Yankov, M. S. Berger, K. J. Larsen, and S. Ruepp, "Improved energy efficiency for optical transport networks by elastic forward error correction," *IEEE J. Opt. Commun. Netw.*, vol. 6, no. 4, pp. 397–407, Apr. 2014.

[7] E. Harstead, D. van Veen, V. Houtsma, and P. Dom, "Technology roadmap for time-division multiplexed passive optical networks (TDM PONs)," *IEEE J. Lightw. Technol.*, vol. 37, no. 2, pp. 657–664, Jan. 2019.

[8] J. Cheng, C. Xie, Y. Chen, X. Chen, M. Tang, and S. Fu, "Comparison of coherent and IMDD transceivers for intra datacenter optical interconnects," in *Opt. Fiber Commun. Conf. (OFC)*, Mar. 2019, p. W1F.2.

[9] C. Fougstedt, O. Gustafsson, C. Bae, E. Börjeson, and P. Larsson-Edefors, "ASIC design exploration for DSP and FEC of 400-Gbit/s coherent data-center interconnect receivers," in *Opt. Fiber Commun. Conf. (OFC)*, 2020, p. Th2A.38.

[10] P. Layec, A. Morea, Y. Pointurier, and J.-C. Antona, "Rate-adaptable optical transmission and elastic optical networks," in *Enabling Technologies for High Spectral-Efficiency Coherent Optical Communication Networks*, X. Zhou and C. Xie, Eds. John Wiley & Sons, Ltd, 2016, ch. 15, pp. 507–546.

[11] G. Bosco, "Advanced modulation techniques for flexible optical transceivers: The rate/reach tradeoff," *IEEE J. Lightw. Technol.*, vol. 37, no. 1, pp. 36–49, Jan. 2019.

[12] A. Ghazisaeidi, L. Schmalen, I. F. de Jauregui, P. Tran, C. Simonneau, P. Brindel, and G. Charlet, "52.9 Tb/s transmission over transoceanic distances using adaptive multi-rate FEC," in *Eur. Conf. Opt. Commun. (ECOC)*, Sept. 2014, p. PD.3.4.

[13] D. A. A. Mello, A. N. Barreto, T. C. de Lima, T. F. Portela, L. Beygi, and J. M. Kahn, "Optical networking with variable-code-rate transceivers," *IEEE J. Lightw. Technol.*, vol. 32, no. 2, pp. 257–266, Jan. 2014.

[14] G. H. Gho and J. M. Kahn, "Rate-adaptive modulation and coding for optical fiber transmission systems," *IEEE J. Lightw. Technol.*, vol. 30, no. 12, pp. 1818–1828, June 2012.

[15] D. Zou and I. B. Djordjevic, "FPGA-based rate-adaptive LDPC-coded modulation for the next generation of optical communication systems," *Opt. Express*, vol. 24, no. 18, pp. 21 159–21 166, Sept. 2016.

[16] K. Sugihara, S. Kametani, K. Kubo, T. Sugihara, and W. Matsumoto, "A practicable rate-adaptive FEC scheme flexible about capacity and distance in optical transport networks," in *Opt. Fiber Commun. Conf. (OFC)*, Mar. 2016, p. M3A.5.

[17] T. Koike-Akino, D. S. Millar, K. Parsons, and K. Kojima, "Rate-adaptive LDPC convolutional coding with joint layered scheduling and shortening design," in *Opt. Fiber Commun. Conf. (OFC)*, Mar. 2018, p. Tu3C.1.

[18] X. Sun, M. Yang, and I. B. Djordjevic, "Real-time FPGA-based rate adaptive LDPC coding for data center networks and PONs," in *Eur. Conf. Opt. Commun. (ECOC)*, Sept. 2018, p. Th2.51.

[19] "IEEE Standard for Ethernet - Amendment 10: Media Access Control Parameters, Physical Layers, and Management Parameters for 200 Gb/s and 400 Gb/s Operation," *IEEE Std 802.3bs-2017 (Amendment to IEEE 802.3-2015 as amended by IEEE's 802.3bw-2015, 802.3by-2016, 802.3bq-2016, 802.3bp-2016, 802.3br-2016, 802.3bn-2016, 802.3bz-2016, 802.3bu-2016, 802.3bv-2017, and IEEE 802.3-2015/Cor1-2017)*, 2017.

[20] K. Cushon, P. Larsson-Edefors, and P. Andrekson, "Low-power 400-Gbps soft-decision LDPC FEC for optical transport networks," *IEEE J. Lightw. Technol.*, vol. 34, no. 18, pp. 4304–4311, Sept. 2016.

[21] A. Balatsoukas-Stimming, M. Meidlinger, R. Ghanaatian, G. Matz, and A. Burg, "A fully-unrolled LDPC decoder based on quantized message passing," in *IEEE Workshop on Signal Processing Systems (SiPS)*, 2015.

[22] S. Weithoffer, R. Klaimi, C. A. Nour, N. Wehn, and C. Douillard, "Fully pipelined iteration unrolled decoders the road to TB/S turbo decoding," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 5115–5119.

[23] G. Gho, L. Klak, and J. M. Kahn, "Rate-adaptive coding for optical fiber transmission systems," *IEEE J. Lightw. Technol.*, vol. 29, no. 2, pp. 222–233, Jan. 2011.

[24] T. Mehmood, M. P. Yankov, A. Fisker, K. Gormsen, and S. Forchhammer, "Rate-adaptive concatenated polar-staircase codes for data center interconnects," in *Opt. Fiber Commun. Conf. (OFC)*, Mar. 2020, p. Th1I.6.

[25] C. Fougstedt and P. Larsson-Edefors, "Energy-efficient high-throughput VLSI architectures for product-like codes," *IEEE J. Lightw. Technol.*, vol. 37, no. 2, pp. 477–485, Jan. 2019.

[26] V. Jain, C. Fougstedt, and P. Larsson-Edefors, "Variable-rate FEC decoder VLSI architecture for 400G rate-adaptive optical communication," in *26th IEEE Int. Conf. on Electronics, Circuits and Systems (ICECS)*, Nov. 2019, pp. 45–48.

[27] D. A. Morero, M. A. Castrillón, A. Aguirre, M. R. Hueda, and O. E. Agazzi, "Design tradeoffs and challenges in practical coherent optical transceiver implementations," *IEEE J. Lightw. Technol.*, vol. 34, no. 1, pp. 121–136, Jan. 2016.

[28] R. Bose and D. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Inf. Control*, vol. 3, no. 1, pp. 68–79, 1960.

[29] P. Elias, "Error-free coding," *IRE Trans. Inf. Theory*, vol. 4, no. 4, pp. 29–37, Sept. 1954.

[30] S. An, H. Tang, and J. Park, "A inversion-less Peterson algorithm based shared KES architecture for concatenated BCH decoder," in *Int. SoC Design Conf. (ISOCC)*, Nov. 2015, pp. 281–282.

[31] X. Zhang and M. O'Sullivan, "Ultra-compressed three-error-correcting BCH decoder," in *IEEE Int. Conf. Circuits Syst. (ISCAS)*, May 2018.

[32] D. Kim, I. Yoo, and I. Park, "Fast low-complexity triple-error-correcting BCH decoding architecture," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 6, pp. 764–768, June 2018.

[33] B. Li, K. J. Larsen, D. Zibar, and I. T. Monroy, "Over 10 dB net coding gain based on 20% overhead hard decision forward error correction in 100G optical communication systems," in *Eur. Conf. Opt. Commun. (ECOC)*, Sept. 2011, p. Tu.6.A.3.

[34] P. Larsson-Edefors and E. Börjeson, "Power-efficient ASIC implementation of DSP algorithms for coherent optical communication," in *IEEE Photonics Society Summer Topicals Meeting Series (SUM)*, July 2020, p. MA1.1.

[35] J. Justesen, "Performance of product codes and related structures with iterated decoding," *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 407–415, Feb. 2011.

[36] W. Liu, J. Rho, and W. Sung, "Low-power high-throughput BCH error correction VLSI design for multi-level cell NAND flash memories," in *IEEE Workshop on Signal Processing Systems*, Oct. 2006, pp. 303–308.

[37] M. Yin, M. Xie, and B. Yi, "Optimized algorithms for binary BCH codes," in *IEEE Int. Symp. on Circuits and Systems*, May 2013, pp. 1552–1555.

[38] W. Peterson, "Encoding and error-correction procedures for the Bose-Chaudhuri codes," *IRE Trans. Inf. Theory*, vol. 6, no. 4, pp. 459–470, Sept. 1960.

[39] P. Larsson-Edefors, C. Fougstedt, and K. Cushon, "Implementation challenges for energy-efficient error correction in optical communication systems," in *Advanced Photonics 2018*, July 2018, p. SpTh4F.2.

[40] C. Condo, P. Giard, F. Leduc-Primeau, G. Sarkis, and W. J. Gross, "A 9.52 dB NCG FEC scheme and 162 b/cycle low-complexity product decoder architecture," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 4, pp. 1420–1431, Apr. 2018.

[41] Y. Wang, J. Lin, and Z. Wang, "A 100 Gbps turbo product code decoder for optical communications," in *Int. Conf. on Computer and Communications (ICCC)*, 2019, pp. 721–725.