Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2008, Article ID 658042, 14 pages
doi:10.1155/2008/658042

Research Article

Reed-Solomon Turbo Product Codes for Optical Communications: From Code Optimization to Decoder Design

Raphaël Le Bidan, Camille Leroux, Christophe Jego, Patrick Adde, and Ramesh Pyndiah

Institut TELECOM, TELECOM Bretagne, CNRS Lab-STICC, Technopôle Brest-Iroise, CS 83818, 29238 Brest Cedex 3, France

Correspondence should be addressed to Raphaël Le Bidan, raphael.lebidan@telecom-bretagne.eu

Received 31 October 2007; Accepted 22 April 2008

Recommended by Jinhong Yuan

Turbo product codes (TPCs) are an attractive solution to improve link budgets and reduce systems costs by relaxing the requirements on expensive optical devices in high capacity optical transport systems. In this paper, we investigate the use of Reed-Solomon (RS) turbo product codes for 40 Gbps transmission over optical transport networks and 10 Gbps transmission over passive optical networks. An algorithmic study is first performed in order to design RS TPCs that are compatible with the performance requirements imposed by the two applications. Then, a novel ultrahigh-speed parallel architecture for turbo decoding of product codes is described. A comparison with binary Bose-Chaudhuri-Hocquenghem (BCH) TPCs is performed. The results show that high-rate RS TPCs offer a better complexity/performance tradeoff than BCH TPCs for low-cost Gbps fiber optic communications.

1. INTRODUCTION

The field of channel coding has undergone major advances for the last twenty years. With the invention of turbo codes [1] followed by the rediscovery of low-density parity-check (LDPC) codes [2], it is now possible to approach the fundamental limit of channel capacity within a few tenths of a decibel over several channel models of practical interest [3]. Although this has been a major step forward, there is still a need for improvement in forward-error correction (FEC), notably in terms of code flexibility, throughput, and cost.

In the early 90's, coinciding with the discovery of turbo codes, the deployment of FEC began in optical fiber communication systems. For a long time, there was no real incentive to use channel coding in optical communications since the bit error rate (BER) in lightwave transmission systems can be as low as 10^{-9}–10^{-15}. Then, the progressive introduction of in-line optical amplifiers and the advent of wavelength division multiplexing (WDM) technology accelerated the use of FEC up to the point that it is now considered almost routine in optical communications. Channel coding is seen as an efficient technique to reduce systems costs and to improve margins against various line impairments such as beat noise, channel cross-talk, or nonlinear dispersion. On the other hand, the design of channel codes for optical communications poses remarkable challenges to the system engineer. Good codes are indeed expected to provide at the same time low overhead (high code rate) and guaranteed large coding gains at very low BER [4]. Furthermore, the issue of decoding complexity should not be overlooked since data rates have now reached 10 Gbps and beyond (up to 40 Gbps), calling for FEC devices with low power-consumption.

FEC schemes for optical communications are commonly classified into three generations. The reader is referred to [5, 6] for an in-depth historical perspective of FEC for optical communication. First-generation FEC schemes mainly relied on the (255, 239) Reed-Solomon (RS) code over the Galois field GF(256), with only 6.7% overhead. In particular, this code was recommended by the ITU for long-haul submarine transmissions. Then, the development of WDM technology provided the impetus for moving to second-generation FEC systems, based on concatenated codes with higher coding gains [7]. Third-generation FEC based on soft-decision decoding is now the subject of intense research since stronger FEC are seen as a promising way to reduce costs by relaxing the requirements on expensive optical devices in high-capacity transport systems.

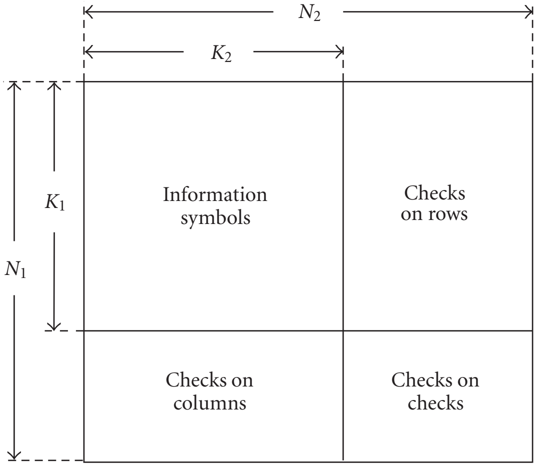

FIGURE 1: Codewords of the product code $\mathcal{P} = \mathcal{C}_1 \otimes \mathcal{C}_2$.

First introduced in [8], turbo product codes (TPCs) based on binary Bose-Chaudhuri-Hocquenghem (BCH) codes are an efficient and mature technology that has found its way in several (either proprietary or public) wireless transmission systems [9]. Recently, BCH TPCs have received considerable attention for third-generation FEC in optical systems since they show good performance at high code rates and have a high-minimum distance by construction. Furthermore, their regular structure is amenable to very-high-data-rate parallel decoding architectures [10, 11]. Research on TPCs for lightwave systems culminated recently with the experimental demonstration of a record coding gain of 10.1 dB at a BER of 10^{-13} using a $(144, 128) \times (256, 239)$ BCH turbo product code with 24.6% overhead [12]. This gain was measured using a turbo decoding very-large-scale-integration (VLSI) circuit operating on 3-bit soft inputs at a data rate of 12.4 Gbps. LDPC codes are also considered as serious candidate for third generation FEC. Impressive coding gains have notably been demonstrated by Monte-Carlo simulation [13]. To date however, to the best of the authors knowledge, no high-rate LDPC decoding architecture has been proposed in order to demonstrate the practicality of LDPC codes for Gbps optical communications.

In this work, we investigate the use of Reed-Solomon TPCs for third-generation FEC in fiber optic communication. Two specific applications are envisioned, namely 40 Gbps line rate transmission over optical transport networks (OTNs), and 10 Gbps data transmission over passive optical networks (PONs). These two applications have different requirements with respect to FEC. An algorithmic study is first carried out in order to design RS product codes for the two applications. In particular, it is shown that high-rate RS TPCs based on carefully designed single-error-correcting RS codes realize an excellent performance/complexity trade-off for both scenarios, compared to binary BCH TPCs of similar code rate. In a second step, a novel parallel decoding architecture is introduced. This architecture allows decoding of turbo product codes at data rates of 10 Gbps and beyond. Complexity estimations show that RS TPCs better trade-off area and throughput than BCH TPCs for full-parallel decoding architectures. An experimental setup based on field-programmable gate array (FPGA) devices has been successfully designed for 10 Gbps data transmission. This prototype demonstrates the practicality of RS TPCs for next-generation optical communications.

The remainder of the paper is organized as follows. Construction and properties of RS product codes are introduced in Section 2. Turbo decoding of RS product codes is described in Section 3. Product code design for optical communication and related algorithmic issues are discussed in Section 4. The challenging issue of designing a high-throughput parallel decoding architecture for product codes is developed in Section 5. A comparison of throughput and complexity between decoding architectures for RS and BCH TPCs is carried out in Section 6. Section 7 describes the successful realization of a turbo decoder prototype for 10 Gbps transmission. Conclusions are finally given in Section 8.

2. REED-SOLOMON PRODUCT CODES

2.1. Code construction and systematic encoding

Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two linear block codes over the Galois field GF(2^m), with parameters (N_1, K_1, D_1) and (N_2, K_2, D_2), respectively. The product code $\mathcal{P} = \mathcal{C}_1 \otimes \mathcal{C}_2$ consists of all $N_1 \times N_2$ matrices such that each column is a codeword in $\mathcal{C}_1$ and each row is a codeword in $\mathcal{C}_2$. It is well known that $\mathcal{P}$ is an (N_1N_2, K_1K_2) linear block code with minimum distance D_1D_2 over GF(2^m) [14]. The direct product construction thus offers a simple way to build long block codes with relatively large minimum distance using simple, short component codes with small minimum distance. When $\mathcal{C}_1$ and $\mathcal{C}_2$ are two RS codes over GF(2^m), we obtain an RS product code over GF(2^m). Similarly, the direct product of two binary BCH codes yields a binary BCH product code.

Starting from a $K_1 \times K_2$ information matrix, systematic encoding of $\mathcal{P}$ is easily accomplished by first encoding the K_1 information rows using a systematic encoder for $\mathcal{C}_2$. Then, the N_2 columns are encoded using a systematic encoder for $\mathcal{C}_1$, thus resulting in the $N_1 \times N_2$ coded matrix shown in Figure 1.

2.2. Binary image of RS product codes

Binary modulation is commonly used in optical communication systems. A binary expansion of the RS product code is then required for transmission. The extension field GF(2^m) forms a vector space of dimension m over GF(2). A binary image $\mathcal{P}_b$ of $\mathcal{P}$ is thus obtained by expanding each code symbol in the product code matrix into m bits using some basis $\mathcal{B}$ for GF(2^m). The polynomial basis $\mathcal{B} = \{1, \alpha, \dots, \alpha^{m-1}\}$ where α is a primitive element of GF(2^m) is the usual choice, although other basis exist [15, Chapter 8]. By construction, $\mathcal{P}_b$ is a binary linear code with length mN_1N_2, dimension mK_1K_2, and minimum distance d at least as large as the symbol-level minimum distance $D = D_1D_2$ [14, Section 10.5].

3. TURBO DECODING OF RS PRODUCT CODES

Product codes usually have high dimension which precludes maximum-likelihood (ML) soft-decision decoding. Yet the particular structure of the product code lends itself to an efficient iterative "turbo" decoding algorithm offering close-to-optimum performance at high-enough signal-to-noise ratios (SNRs).

Assume that a binary transmission has taken place over a binary-input channel. Let $\mathbf{Y} = (y_{i,j})$ denote the matrix of samples delivered by the receiver front-end. The turbo decoder soft input is the channel log-likelihood ratio (LLR) matrix, $\mathbf{R} = (r_{i,j})$, with

$$r_{i,j} = A \ln \frac{f_1(y_{i,j})}{f_0(y_{i,j})}. \tag{1}$$

Here A is a suitably chosen constant term, and $f_b(y)$ denotes the probability of observing the sample y at the channel output given that bit b has been transmitted.

Turbo decoding is realized by decoding successively the rows and columns of the channel matrix $\mathbf{R}$ using soft-input soft-output (SISO) decoders, and by exchanging reliability information between the decoders until a reliable decision can be made on the transmitted bits.

3.1. *SISO decoding of the component codes*

In this work, SISO decoding of the RS component codes is performed at the bit-level using the Chase-Pyndiah algorithm. First introduced in [8] for binary BCH codes and latter extended to RS codes in [16], the Chase-Pyndiah decoder consists of a soft-input hard-output Chase-2 decoder [17] augmented by a soft-output computation unit.

Given a soft-input sequence $\mathbf{r} = (r_1, \ldots, r_{mN})$ corresponding to a row ($N = N_2$) or column ($N = N_1$) of $\mathbf{R}$, the Chase-2 decoder first forms a binary hard-decision sequence $\mathbf{y} = (y_1, \ldots, y_{mN})$. The reliability of the hard-decision y_i on the ith bit is measured by the magnitude $|r_i|$ of the corresponding soft input. Then, N_{ep} error patterns are generated by testing different combinations of 0 and 1 in the L_r least reliable bit positions. In general, $N_{\text{ep}} \leq 2^{L_r}$ with equality if all combinations are considered. Those error patterns are added modulo-2 to the hard-decision sequence $\mathbf{y}$ to form candidate sequences. Algebraic decoding of the candidate sequences returns a list with at most N_{ep} distinct candidate codewords. Among them, the codeword $\mathbf{d}$ at minimum Euclidean distance from the input sequence $\mathbf{r}$ is selected as the final decision.

Soft-output computation is then performed as follows. For a given bit i, the list of candidate codewords is searched for a *competing* codeword $\mathbf{c}$ at minimum Euclidean distance from $\mathbf{r}$ and such that $c_i \neq d_i$. If such a codeword exists, then the soft output r'_i on the ith bit is given by

$$r'_i = \left(\frac{\|\mathbf{r} - \mathbf{c}\|^2 - \|\mathbf{r} - \mathbf{d}\|^2}{4} \right) \times d_i, \tag{2}$$

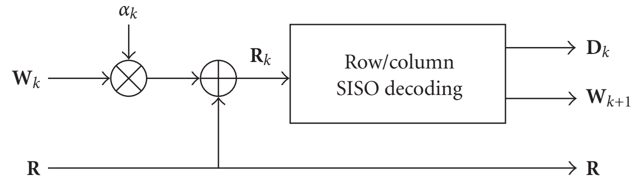

FIGURE 2: Block diagram of the turbo-decoder at the kth half-iteration.

where $\|\cdot\|^2$ denotes the squared norm of a sequence. Otherwise, the soft output is computed as follows:

$$r'_i = r_i + \beta \times d_i, \tag{3}$$

where β is a positive value, computed on a per-codeword basis, as suggested in [18]. Following the so-called "turbo principle," the soft input r_i is finally subtracted from the soft output r'_i to obtain the extrinsic information

$$w_i = r'_i - r_i \tag{4}$$

which will be sent to the next decoder.

3.2. *Iterative decoding of the product code*

The block diagram of the turbo decoder at the kth half-iteration is shown in Figure 2. A half-iteration stands for a row or column decoding step, and one iteration comprises two half-iterations. The input of the SISO decoder at half-iteration k is given by

$$\mathbf{R}_k = \mathbf{R} + \alpha_k \mathbf{W}_k, \tag{5}$$

where α_k is a scaling factor used to attenuate the influence of extrinsic information during the first iterations, and where $\mathbf{W}_k = (w_{i,j})$ is the extrinsic information matrix delivered by the SISO decoder at the previous half-iteration. The decoder outputs an updated extrinsic information matrix $\mathbf{W}_{k+1}$, and possibly a matrix $\mathbf{D}_k$ of hard-decisions. Decoding stops when a given maximum number of iterations have been performed, or when an early-termination condition (stop criterion) is met.

The use of a stop criterion can improve the convergence of the iterative decoding process and also reduce the average power-consumption of the decoder by decreasing the average number of iterations required to decode a block. An efficient stop criterion taking advantage of the structure of the product codes was proposed in [19]. Another simple and effective solution is to stop when the hard decisions do not change between two successive half-iterations (i.e., no further corrections are done).

4. RS PRODUCT CODE DESIGN FOR OPTICAL COMMUNICATIONS

Two optical communication scenarios have been identified as promising applications for third-generation FEC based on RS TPCs: 40 Gbps data transport over OTN, and 10 Gbps data transmission over PON. In this section, we first review the own expectations of each application with respect to FEC. Then, we discuss the algorithmic issues that have been encountered and solved in order to design RS TPCs that are compatible with these requirements.

4.1. FEC design for data transmission over OTN and PON

40 Gbps transport over OTN calls for both high-coding gains and low overhead (<10%). High-coding gains are required in order to insure high data integrity with BER in the range 10^{-13}–10^{-15}. Low-overhead limit optical transmission impairments caused by bandwidth extension. Note that these two requirements usually conflict with each other to some extent. The complexity and power consumption of the decoding circuit is also an important issue. A possible solution, proposed in [6], is to multiplex in parallel four powerful FEC devices at 10 Gbps. However 40 Gbps low-cost line cards are a key to the deployment of 40 Gbps systems. Furthermore, the cost of line cards is primarily dominated by the electronics and optics operating at the serial line rate. Thus, a single low-cost 40 Gbps FEC device could compete favorably with the former solution if the loss in coding gain (if any) remains small enough.

For data transmission over PON, channel codes with low cost and low latency (small block size) are preferred to long codes (>10 Kbits) with high-coding gain. BER requirements are less stringent than for OTN and are typically of the order of 10^{-11}. High-coding gains result in increased link budget [20]. On the other hand, decoding complexity should be kept at a minimum in order to reduce the cost of optical network units (ONUs) deployed at the end-user side. Channel codes for PON are also expected to be robust against burst errors.

4.2. Choice of the component codes

On the basis of the above-mentioned requirements, we have chosen to focus on RS product codes with less than 20% overhead. Higher overheads lead to larger signal bandwidth, thereby increasing in return the complexity of electronic and optical components. Since the rate of the product code is the product of the individual rates of the component codes, RS component codes with code rate $R \geq 0.9$ are necessary. Such code rates can be obtained by considering multiple-error-correcting RS codes over large Galois fields, that is, GF(256) and beyond. Another solution is to use single-error-correcting (SEC) RS codes over Galois fields of smaller order (32 or 64). The latter solution has been retained in this work since it leads to low-complexity SISO decoders.

First, it is shown in [21] that 16 error patterns are sufficient to obtain near-optimum performance with the Chase-Pyndiah algorithm for SEC RS codes. In contrast, more sophisticated SISO decoders are required with multiple-error-correcting RS codes (e.g., see [22] or [23]) since the number of error patterns necessary to obtain near-optimum performance with the Chase-Pyndiah algorithm grows exponentially with mt for a t-error-correction RS code over GF(2^m).

In addition, SEC RS codes admit low-complexity algebraic decoders. This feature further contributes to reducing the complexity of the Chase-Pyndiah algorithm. For multiple-error-correcting RS codes, the Berlekamp-Massey algorithm and the Euclidean algorithm are the preferred algebraic decoding methods [15]. But they introduce unnecessary overhead computations for SEC codes. Instead, a more simpler decoder is obtained from the direct decoding method devised by Peterson, Gorenstein, and Zierler (PGZ decoder) [24, 25]. First, the two syndromes S_1 and S_2 are calculated by evaluating the received polynomial $r(x)$ at the two code roots α^b and α^{b+1}:

$$S_i = r(\alpha^{b+i-1}) = \sum_{\ell=0}^{N-1} r_\ell \alpha^{\ell(b+i-1)}, \quad i = 1, 2. \tag{6}$$

If $S_1 = S_2 = 0$, $r(x)$ is a valid codeword and decoding stops. If only one of the two syndromes is zero, a decoding failure is declared. Otherwise, the error locator X is calculated as

$$X = \frac{S_2}{S_1} \tag{7}$$

from which the error location i is obtained by taking the discrete logarithm of X. The error magnitude E is finally given by

$$E = \frac{S_1}{X^b}. \tag{8}$$

Hence, apart from the syndrome computation, at most two divisions over GF(2^m) are required to obtain the error position and value with the PGZ decoder (only one is needed when $b = 0$). The overall complexity of the PGZ decoder is usually dominated by the initial syndrome computation step. Fortunately, the syndromes need not be fully recomputed at each decoding attempt in the Chase-2 decoder. Rather, they can be updated in a very simple way by taking only into account the bits that are flipped between successive error patterns [26]. This optimization further alleviates SISO decoding complexity.

On the basis of the above arguments, two RS product codes have been selected for the two envisioned applications. The $(31, 29)^2$ RS product code over GF(32) has been retained for PON systems since it combines a moderate overhead of 12.5% with a moderate code length of 4805 coded bits. This is only twice the code length of the classical (255, 239) RS code over GF(256). On the other hand, the $(63, 61)^2$ RS product code over GF(64) has been preferred for OTN, since it has a smaller overhead (6.3%), similar to the one introduced by the standard (255, 239) RS code, and also a larger coding gain, as we will see later.

4.3. Performance analysis and code optimization

RS product codes built from SEC RS component codes are very attractive from the decoding complexity point of view. On the other hand, they have low-minimum distance $D = 3 \times 3 = 9$ at the symbol level. Therefore, it is of capital interest to verify that this low-minimum distance does not introduce error flares in the code performance curve that would penalize the effective coding gain at low BER. Monte-carlo simulations can be used to evaluate the code performance down to BER of 10^{-10}–10^{-11} within a reasonable computation time. For lower BER, analytical bounding techniques are required.

In the following, binary on-off keying (OOK) intensity modulation with direct detection over additive white Gaussian noise (AWGN) is assumed. This model was adopted here as a first approximation which simplifies the analysis and also facilitates the comparison with other channel codes. More sophisticated models of optical systems for the purpose of assessing the performance of channel codes are developed in [27, 28]. Under the previous assumptions, the BER of the RS product code at high SNRs and under ML soft-decision decoding is well approximated by the first term of the union bound:

$$\mathrm{BER} \approx \frac{d}{mN_1N_2}\frac{B_d}{2}\,\mathrm{erfc}\left(Q\sqrt{\frac{d}{2}}\right), \tag{9}$$

where Q is the input Q-factor (see [29, Chapter 5]), d is the minimum distance of the binary image $\mathcal{P}_b$ of the product code, and B_d the corresponding multiplicity (number of codewords with minimum Hamming weight d in $\mathcal{P}_b$). This expression shows that the asymptotic performance of the product code is determined by the bit-level minimum distance d of the product code, not by the symbol minimum distance D_1D_2.

The knowledge of the quantities d and B_d is required in order to predict the asymptotic performance of the code in the high Q-factor (low BER) region using (9). These parameters depend in turn on the basis $\mathcal{B}$ used to represent the 2^m-ary symbols as bits, and are usually unknown. Computing the exact binary weight enumerator of RS product codes is indeed a very difficult problem. Even the symbol weight enumerator is hard to find since it is not completely determined by the symbol weight enumerators of the component codes [30]. An average binary weight enumerator for RS product codes was recently derived in [31]. This enumerator is simple to calculate. However simulations are still required to assess the tightness of the bounds for a particular code realization. A computational method that allows the determination of d and A_d under certain conditions was recently suggested in [32]. This method exploits the fact that product codewords with minimum symbol weight D_1D_2 are readily constructed as the direct product of a minimum-weight row codeword with a minimum-weight column codeword. Specifically, there are exactly

$$A_{D_1D_2} = (2^m - 1)\binom{N_1}{D_1}\binom{N_2}{D_2} \tag{10}$$

distinct codewords with symbol weight D_1D_2 in the product code $\mathcal{C}_1 \otimes \mathcal{C}_2$. They can be enumerated with the help of a computer provided the number $A_{D_1D_2}$ of such codewords is not too large. Estimates $\hat{d}$ and $B_{\hat{d}}$ are then obtained by computing the Hamming weight of the binary expansion of those codewords. Necessarily, $d \leq \hat{d}$. If it can be shown that product codewords of symbol weight $>D_1D_2$ necessarily have binary minimum distance $>\hat{d}$ at the bit level (this is not always the case, depending on the value of $\hat{d}$), then it follows that $d = \hat{d}$ and $B_d = B_{\hat{d}}$.

Table 1: Minimum distance d and multiplicity B_d for the binary image of the $(31, 29)^2$ and $(63, 61)^2$ RS product codes as a function of the first code root α^b.

Product code	mK^2	mN^2	R	b	d	B_d
$(31, 29, 3)^2$	4205	4805	0.875	1	9	217,186
				0	14	6,465,608
$(63, 61, 3)^2$	22326	23814	0.937	1	9	4,207,140
				0	14	88,611,894

This method has been used to obtain the binary minimum distance and multiplicity of the $(31, 29)^2$ and $(63, 61)^2$ RS product codes using *narrow-sense* component codes with generator polynomial $g(x) = (x - \alpha)(x - \alpha^2)$. This is the classical definition of SEC RS codes that can be found in most textbooks. The results are given in Table 1. We observe that in both cases, we are in the most unfavorable case where the bit-level minimum distance d is equal to the symbol-level minimum distance D, and no greater. Simulation results for the two RS TPCs after 8 decoding iterations are shown in Figures 3 and 4, respectively. The corresponding asymptotic performance calculated using (9) are plotted in dashed lines. For comparison purpose, we have also included the performance of algebraic decoding of RS codes of similar code rate over GF(256). We observe that the low-minimum distance introduces error flares at BER of 10^{-8} and 10^{-9} for the $(31, 29)^2$ and $(63, 61)^2$ product codes, respectively. Clearly, the two RS TPCs do not match the BER requirements imposed by the envisioned applications.

One solution to increase the minimum distance of the product code is to resort to code extension or expurgation. However this approach increases the overhead. It also increases decoding complexity since a higher number of error patterns are then required to maintain near-optimum performance with the Chase-Pyndiah algorithm [21]. In this work, another approach has been considered. Specifically, investigations have been conducted in order to identify code constructions that can be mapped into binary images with minimum distance larger than 9. One solution is to investigate different basis $\mathcal{B}$. How to find a basis that maps a nonbinary code into a binary code with bit-level minimum distance strictly larger than the symbol-level designed distance remains a challenging research problem. Thus, the problem was relaxed by fixing the basis to be the polynomial basis, and studying instead the influence of the choice of the code roots on the minimum distance of the binary image. Any SEC RS code over GF(2^m) can be compactly described by its generator polynomial

$$g(x) = (x - \alpha^b)(x - \alpha^{b+1}), \tag{11}$$

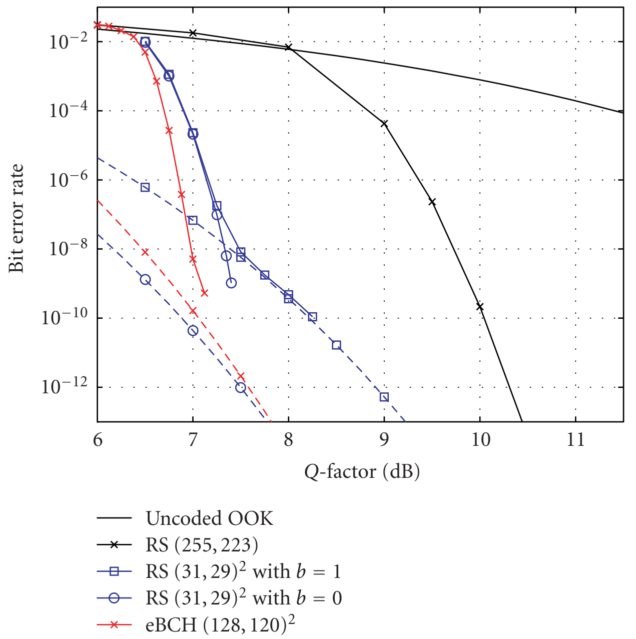

FIGURE 3: BER performance of the $(31, 29)^2$ RS product code as a function of the first code root α^b, after 8 iterations.

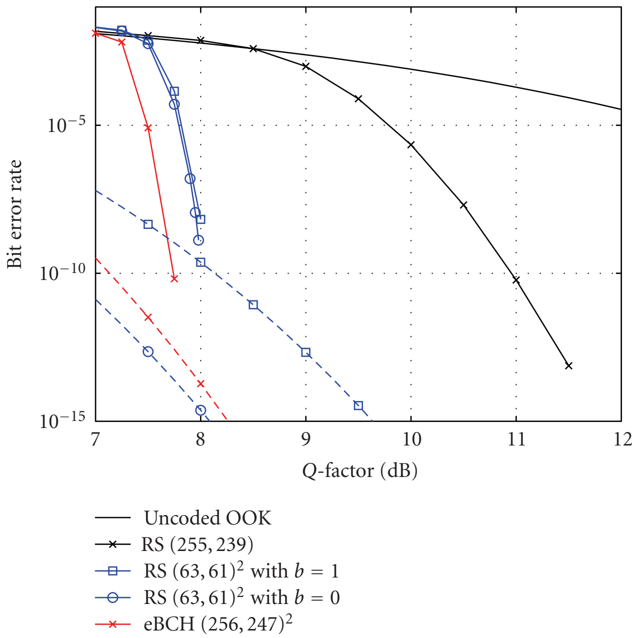

FIGURE 4: BER performance of the $(63, 61)^2$ RS product code as a function of the first code root α^b, after 8 decoding iterations.

where b is an integer in the range $0 \cdots 2^m - 2$. Narrow-sense RS codes are obtained by setting $b = 1$ (which is the usual choice for most applications). Note however that different values for b generate different sets of codewords, and thus different RS codes with possibly different binary weight distributions. In [32], it is shown that *alternate* SEC RS codes obtained by setting $b = 0$ have minimum distance $d = D + 1 = 4$ at the bit level. This is a notable improvement over classical narrow-sense ($b = 1$) RS codes for which $d = D = 3$. This result suggests that RS product codes should be preferably built from two RS component codes with first root α^0. RS product codes constructed in this way will be called alternate RS product codes in the following.

We have computed the binary minimum distance d and multiplicity A_d of the $(31, 29)^2$ and $(63, 61)^2$ alternate RS product codes. The values are reported in Table 1. Interestingly, the alternate product codes have a minimum distance d as high as 14 at the bit-level, at the expense of an increase of the error coefficient B_d. Thus, we get most of the gain offered by extended or expurgated codes (for which $d = 16$, as verified by computer search) but without reducing the code rate. It is also worth noting that this extra coding gain is obtained without increasing decoding complexity. The same SISO decoder is used for both narrow-sense and alternate SEC RS codes. In fact, the only modifications occur in (6)–(8) of the PGZ decoder, which actually simplify when $b = 0$. Simulated performance and asymptotic bounds for the alternate RS product codes are shown in Figures 3 and 4. A notable improvement is observed in comparison with the performance of the narrow-sense product codes since the error flare is pushed down by several decades in both cases. By extrapolating the simulation results, the net coding gain (as defined in [5]) at a BER of 10^{-13} is estimated to be around 8.7 dB and 8.9 dB for the $\text{RS}(31, 29)^2$ and $\text{RS}(63, 61)^2$, respectively. As a result, the two selected RS product codes are now fully compatible with the performance requirements imposed by the respective envisioned applications. More importantly, this achievement has been obtained at no cost.

4.4. Comparison with BCH product codes

A comparison with BCH product codes is in order since BCH product codes have already found application in optical communications. A major limitation of BCH product codes is that very large block lengths (>60000 coded bits) are required to achieve high code rates ($R > 0.9$). On the other hand, RS product codes can achieve the same code rate than BCH product codes, but with a block size about 3 times smaller [21]. This is an interesting advantage since, as shown latter in the paper, large block lengths increase the decoding latency and also the memory complexity in the decoder architecture. RS product codes are also expected to be more robust to error bursts than BCH product codes. Both coding schemes inherit burst-correction properties from the row-column interleaving in the direct product construction. But RS product codes also benefit from the fact that, in the most favorable case, m consecutive erroneous bits may cause a single symbol error in the received word.

A performance comparison has been carried out between the two selected RS product codes and extended BCH(eBCH) product codes of similar code rate: the $\text{eBCH}(128, 120)^2$ and the $\text{eBCH}(256, 247)^2$. Code extension has been used for BCH codes since it increases minimum distance without increasing decoding complexity nor decreasing significantly the code rate, in contrast to RS codes. Both eBCH TPCs have minimum distance 16 with

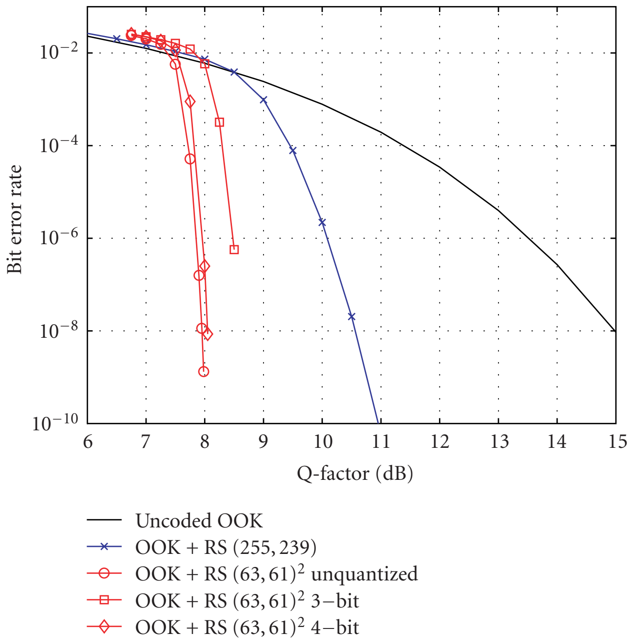

Figure 5: BER performance for the $(63, 61)^2$ RS product code as a function of the number of quantization bits for the soft-input (sign bit included).

multiplicities 85344^2 and 690880^2, respectively. Simulation results after 8 iterations are shown in Figures 3 and 4. The corresponding asymptotic bounds are plotted in dashed lines. We observe that eBCH TPCs converge at lower Q-factors. As a result, a 0.3-dB gain is obtained at BER in the range 10^{-8}–10^{-10}. However, the large multiplicities of eBCH TPCs introduce a change of slope in the performance curves at lower BER. In fact, examination of the asymptotic bounds shows that alternate RS TPCs are expected to perform at least as well as eBCH TPCs in the BER range of interest for optical communication, for example, 10^{-10}–10^{-15}. Therefore, we conclude that RS TPCs compare favorably with eBCH TPCs in terms of performance. We will see in the next sections that RS TPCs have additional advantages in terms of decoding complexity and throughput for the target applications.

4.5. *Soft-input quantization*

The previous performance study assumed unquantized soft values. In a practical receiver, a finite number q of bits (sign bit included) is used to represent soft information. Soft-input quantization is performed by an analog-to-digital converter (ADC) in the receiver front-end. The very high bit rate in fiber optical systems makes ADC a challenging issue. It is therefore necessary to study the impact of soft-input quantization on the performance. Figure 5 presents simulation results for the $(63, 61)^2$ alternate RS product code using $q = 3$ and $q = 4$ quantization bits, respectively. For comparison purpose, the performance without quantization is also shown. Using $q = 4$ bits yields virtually no degradation with respect to ideal (infinite) quantization, whereas $q = 3$ bits of quantization introduce a 0.5 dB penalty. Similar conclusions have been obtained with the $(31, 29)^2$ RS product code and also with various eBCH TPCs, as reported in [27, 33] for example.

5. FULL-PARALLEL TURBO DECODING ARCHITECTURE DEDICATED TO PRODUCT CODES

Designing turbo decoding architectures compatible with the very high-line rate requirements imposed by fiber optics systems at reasonable cost is a challenging issue. Parallel decoding architectures are the only solution to achieve data rates above 10 Gbps. A simple architectural solution is to duplicate the elementary decoders in order to achieve the given throughput. However, this solution results in a turbo decoder with unacceptable cumulative area. Thus, smarter parallel decoding architectures have to be designed in order to better trade-off performance and complexity under the constraint of a high-throughput. In the following, we focus on an (N^2, K^2) product code obtained from with two identical (N, K) component codes over GF(2^m). For 2^m-ary RS codes, $m > 1$ whereas $m = 1$ for binary BCH codes.

5.1. *Previous work*

Many turbo decoder architectures for product codes have been proposed in the literature. The classical approach involves decoding all the rows or all the columns of a matrix before the next half-iteration. When an application requires high-speed decoders, an architectural solution is to cascade SISO elementary decoders for each half-iteration. In this case, memory blocks are necessary between each half-iteration to store channel data and extrinsic information. Each memory block is composed of four memories of mN^2 soft values. Thus, duplicating a SISO elementary decoder results in duplicating the memory block which is very costly in terms of silicon area. In 2002, a new architecture for turbo decoding product codes was proposed [10]. The idea is to store several data at the same address and to perform semiparallel decoding to increase the data rate. However, it is necessary to process these data by row and by column. Let us consider l adjacent rows and l adjacent columns of the initial matrix. The l^2 data constitute a word of the new matrix that has l^2 times fewer addresses. This data organization does not require any particular memory architecture. The results obtained show that the turbo decoding throughput is increased by l^2 when l elementary decoders processing l data simultaneously are used. Turbo decoding latency is divided by l. The area of the l elementary decoders is increased by $l/2$ while the memory is kept constant.

5.2. *Full-parallel decoding principle*

All rows (or all columns) of a matrix can be decoded in parallel. If the architecture is composed of $2N$ elementary decoders, an appropriate treatment of the matrix allows the elimination of the reconstruction of the matrix between each half-iteration decoding step. Specifically, let i and j be the indices of a row and a column of the $N \times N$ matrix. In full-parallel processing, the row decoder i begins the

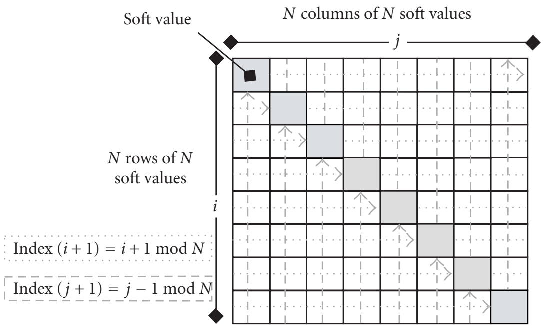
Soft value
N columns of N soft values
j
N rows of N soft values
i
Index (i + 1) = i + 1 mod N
Index (j + 1) = j − 1 mod N

Figure 6: Full-parallel decoding of a product code matrix.

decoding by the soft value in the ith position. Moreover, each row decoder processes the soft values by increasing the index by one modulo N. Similarly, the column decoder j begins the decoding by the soft value in the jth position. In addition, each column decoder processes the soft values by decreasing the index by one modulo N. In fact, full-parallel decoding of turbo product code is possible thanks to the cyclic property of BCH and RS codes. Indeed, every cyclic shift $\mathbf{c}' = (c_{N-1}, c_0, \dots, c_{N-3}, c_{N-2})$ of a codeword $\mathbf{c} = (c_0, c_1, \dots, c_{N-2}, c_{N-1})$ is also a valid codeword in a cyclic code. Therefore, only one-clock period is necessary between two successive matrix decoding operations. The full-parallel decoding of an $N \times N$ product code matrix is described in Figure 6. A similar strategy was previously presented in [34] where memory access conflicts are resolved by means of an appropriate treatment of the matrix.

The elementary decoder latency depends on the structure of the decoder (i.e., number of pipeline stages) and the code length N. Here, as the reconstruction matrix is removed, the latency between row and column decoding is null.

5.3. Full-parallel architecture for product codes

The major advantage of our full-parallel architecture is that it enables the memory block of $4mN^2$ soft values between each half-iteration to be removed. However, the codeword soft values exchanged between the row and column decoders have to be routed. One solution is to use a connection network for this task. In our case, we have chosen an Omega network. The Omega network is one of several connection networks used in parallel machines [35]. It is composed of $\log_2 N$ stages, each having $N/2$ exchange elements. In fact, the Omega network complexity in terms of number of connections and of 2×2 switch transfer blocks is $N \times \log_2 N$ and $(N/2)\log_2 N$, respectively. For example, the equivalent gate complexity of a 31×31 network can be estimated to be 200 logic gates per exchanged bit. Figure 7 depicts a full-parallel architecture for the turbo decoding of product codes. It is composed of cascaded modules for the turbo decoder. Each module is dedicated to one iteration. However, it is possible to process several iterations by the same module. In our approach, $2N$ elementary decoders and 2 connection blocks are necessary for one module. A connection block is composed of 2 Omega networks exchanging the $\mathbf{R}$ and $\mathbf{R}_k$ soft values. Since the Omega network has low complexity, the full-parallel turbo decoder complexity essentially depends on the complexity of the elementary decoder.

5.4. Elementary SISO decoder architecture

The block diagram of an elementary SISO decoder is shown in Figure 2, where k stands for the current half-iteration number. $\mathbf{R}_k$ is the soft-input matrix computed from the previous half-iteration whereas $\mathbf{R}$ denotes the initial matrix delivered by the receiver front-end ($\mathbf{R}_k = \mathbf{R}$ for the 1st half-iteration). $\mathbf{W}_k$ is the extrinsic information matrix. α_k is a scaling factor that depends on the current half-iteration and which is used to mitigate the influence of the extrinsic information during the first iterations. The decoder architecture is structured in three pipelined stages identified as reception, processing, and transmission units [36]. During each stage, the N soft values of the received word $\mathbf{R}_k$ are processed sequentially in N clock periods. The reception stage computes the initial syndromes S_i and finds the L_r least reliable bits in the received word. The main function of the processing stage is to build and then to correct the N_{ep} error patterns obtained from the initial syndrome and to combine the least reliable bits. Moreover, the processing stage also has to produce a metric (Euclidean distance between error pattern and received word) for each error pattern.Finally, a selection function identifies the maximum likelihood codeword $\mathbf{d}$ and the competing codewords $\mathbf{c}$ (if any). The transmission stage performs different functions: computing the reliability for each binary soft value, computing the extrinsic information, and correcting the received soft values. The N soft values of the codeword are thus corrected sequentially. The decoding process needs to access the $\mathbf{R}$ and $\mathbf{R}_k$ soft values during the three decoding phases. For this reason, these words are implemented into six random access memories (RAMs) of size $q \times m \times N$ controlled by a finite-state machine. In summary, a full-parallel TPC decoder architecture requires low-complexity decoders.

6. COMPLEXITY AND THROUGHPUT ANALYSIS OF THE FULL-PARALLEL REED-SOLOMON TURBO DECODERS

Increasing the throughput regardless of the turbo decoder complexity is not relevant. In order to compare the throughput and complexity of RS and BCH turbo decoders, we propose to measure the *efficiency* η of a parallel architecture by the ratio

$$\eta = \frac{T}{C}, \tag{12}$$

where T is the throughput and C is the complexity of the design. An efficient architecture is expected to have a high η ratio, that is, a high throughput with low hardware complexity. In this section, we determine and compare the efficiency of TPC decoders based on SEC BCH and RS component codes, respectively.

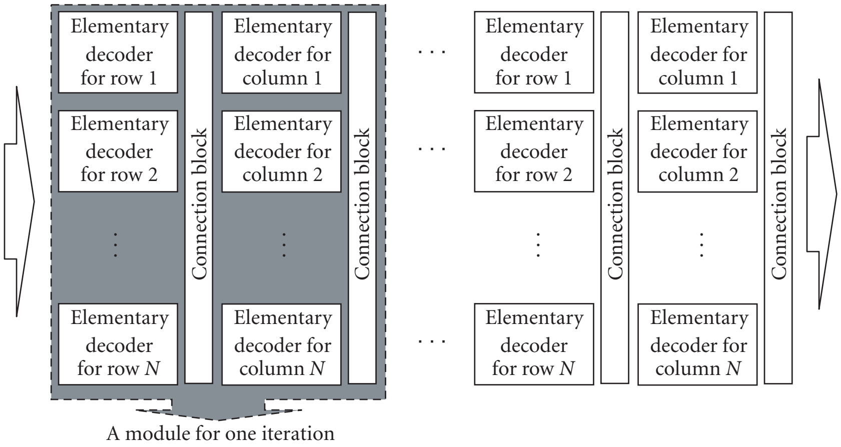
Elementary decoder for row 1
Elementary decoder for row 2
Elementary decoder for row N
Connection block
Elementary decoder for column 1
Elementary decoder for column 2
Elementary decoder for column N
Connection block
. . .
Elementary decoder for row 1
Elementary decoder for row 2
Elementary decoder for row N
Connection block
Elementary decoder for column 1
Elementary decoder for column 2
Elementary decoder for column N
Connection block
A module for one iteration

Figure 7: Full-parallel architecture for decoding of product codes.

6.1. Turbo decoder complexity analysis

A turbo decoder of product code corresponds to the cumulative area of computation resources, memory resources, and communication resources. In a full-parallel turbo decoder, the main part of the complexity is composed of memory and computation resources. Indeed, the major advantage of our full-parallel architecture is that it enables the memory blocks between each half-iteration to be replaced by Omega connection networks. Communication resources thus represent less than 1% of the total area of the turbo decoder. Consequently, the following study will only focus on memory and computation resources.

6.1.1. Complexity analysis of computation resources

The computation resources of an elementary decoder are split into three pipelined stages. The reception and transmission stages have $O(\log(N))$ complexity. For these two stages, replacing a BCH code by an RS code of same code length N (at the symbol level) over GF(2^m) results in an increase of both complexity and throughput by a factor m. As a result, efficiency is constant in these parts of the decoder. However, the hardware complexity of the processing stage increases linearly with the number N_{ep} of error patterns. Consequently, the increase in the local parallelism rate has no influence on the area of this stage and thus increases the efficiency of an RS SISO decoder. In order to verify those general considerations, turbo decoders for the $(15,13)^2$, $(31,29)^2$, and $(63,61)^2$ RS product codes were described in HDL language and synthesized. Logic syntheses were performed using the Synopsys tool Design Compiler with an ST-microelectronics 90 nm CMOS process. All designs were clocked with 100 MHz. Complexity of BCH turbo decoders was estimated thanks to a generic complexity model which can deliver an estimation of the gate count for any code size and any set of decoding parameters. Therefore, taking into account the implementation and performance constraints, this model can be used to select a code size N and a set of decoding parameters [37]. In particular, the numbers of error patterns N_{ep} and also the number of competing codewords kept for soft-output computation directly affect both the hardware complexity and the decoding performance. Increasing these parameter values improves performance but also increases complexity.

Table 2: Computation resource complexity of selected TPC decoders in terms of gate count.

Code	Rate	Elementary decoder	Full-parallel module
$(32,26)^2$ BCH	0.66	2 791	178 624
$(64,57)^2$ BCH	0.79	3 139	401 792
$(128,120)^2$ BCH	0.88	3 487	892 672
$(15,13)^2$ RS	0.75	3 305	99 150
$(31,29)^2$ RS	0.88	4 310	267 220
$(63,61)^2$ RS	0.94	6 000	756 000

Table 2 summarizes some computation resource complexities in terms of gate count for different BCH and RS product codes. Firstly, the complexity of an elementary decoder for each product code is given. The results clearly show that RS elementary decoders are more complex than BCH elementary decoders over the same Galois field. Complexity results for a full-parallel module of the turbo decoding process are also given in Table 2. As described in Figure 7, a full-parallel module is composed of $2N$ elementary decoders and 2 connection blocks for one iteration. In this case, full-parallel modules composed of RS elementary decoders are seen to be less complex than full-parallel modules composed of BCH elementary decoders when comparing eBCH and RS product codes of similar code rate R. For instance, for a code rate $R = 0.88$, the computation resource complexity in terms of gate count are about 892, 672 and 267, 220 for the BCH$(128,120)^2$ and RS$(31,29)^2$, respectively. This is due to the fact that RS codes need smaller code length N (at the symbol level) to achieve a given code rate, in contrast to binary BCH codes. Considering again the previous example, only 31×2 decoders are necessary in the RS case for full-parallel decoding compared to 128×2 decoders in the BCH case. Similarly,

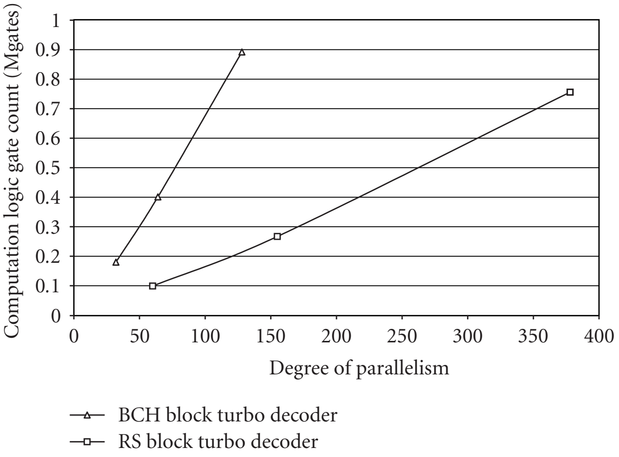

Figure 8: Comparison of computation resource complexity.

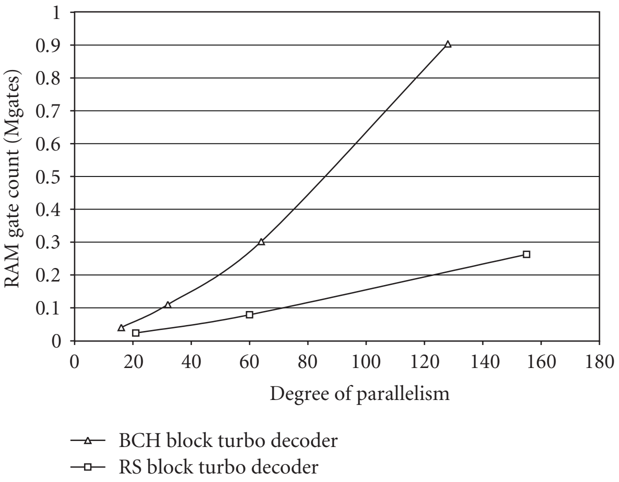

Figure 9: Comparison of internal RAM complexity.

Figure 8 gives computation resource area of BCH and RS turbo decoders for 1 iteration and different parallelism degrees. We verify that higher P (i.e., higher throughput) can be obtained with less computation resources using RS turbo decoders. This means that RS product codes are more efficient in terms of computation resources for full-parallel architectures dedicated to turbo decoding.

6.1.2. Complexity analysis of memory resources

A half-iteration of a parallel turbo decoder contains N banks of $q \times m \times N$ bits. The internal memory complexity of a parallel decoder for one half-iteration can be approximated by

$$S_{\text{RAM}} \simeq \gamma \times q \times m \times N^2, \tag{13}$$

where γ is a technological parameter specifying the number of equivalent gate counts per memory bit, q is the number of quantization bits for the soft values, and m is the number of bits per Galois field element. Using (17), it can also be expressed as

$$S_{\text{RAM}} = \gamma \times \frac{P^2}{m} \times q, \tag{14}$$

where P is the parallelism degree, defined as the number of generated bits per clock period (t_0).

Let us consider a BCH code and an RS code of similar code length $N = 2^m - 1$. For BCH codes, a symbol corresponds to 1 bit, whereas it is made of m bits for RS codes. Calculating the SISO memory area for both BCH and RS gives the following ratio:

$$\frac{S_{\text{RAM}}(\text{BCH})}{S_{\text{RAM}}(\text{RS})} = m = \log_2(N+1). \tag{15}$$

This result shows that RS turbo decoders have lower memory complexity for a given parallelism rate. This was confirmed by memory area estimations results showed in Figure 9. Random access memory (RAM) area of BCH and RS turbo decoders for a half-iteration and different parallelism degrees are plotted using a memory area estimation model provided by ST-Microelectronics. We can observe that higher P (i.e., higher throughput) can be obtained with less memory when using an RS turbo decoder. Thus, full-parallel decoding of RS codes is more memory-efficient than BCH code turbo decoding.

6.2. Turbo decoder throughput analysis

In order to maximize the data rate, decoding resources are assigned for each decoding iteration. The throughput of a turbo decoder can be defined as

$$T = P \times R \times f_0, \tag{16}$$

where R is the code rate and $f_0 = 1/t_0$ is the maximum frequency of an elementary SISO decoder. Ultrahigh throughput can be reached by increasing these three parameters.

(i) R is a parameter that exclusively depends on the code considered. Thus, using codes with a higher code rate (e.g., RS codes) would provide larger throughput.

(ii) In a full-parallel architecture, a maximum throughput is obtained by duplicating N elementary decoders generating m soft values per clock period. The parallelism degree can be expressed as

$$P = N \times m. \tag{17}$$

Therefore, enhanced parallelism degree can be obtained by using nonbinary codes (e.g., RS codes) with larger code length N.

(iii) Finally, in a high-speed architecture, each elementary decoder has to be optimized in terms of working frequency f_0. This is accomplished by including pipeline stages within each elementary SISO decoder. RS and BCH turbo decoders of equivalent code size have equivalent working frequency f_0 since RS decoding is performed by introducing some local parallelism at the soft value level. This result was verified during logic syntheses. The main drawback of pipelining elementary decoders is the extra complexity generated by internal memory requirement.

Table 3: Hardware efficiency of selected TPC decoders.

Code	R	P	T	C	η
$(32,26)^2$ BCH	0.66	32	2.11	201	10.5
$(64,57)^2$ BCH	0.79	64	5.06	508	9.97
$(128,120)^2$ BCH	0.88	128	11.26	1361	8.27
$(15,13)^2$ RS	0.75	60	4.5	128	35.0
$(31,29)^2$ RS	0.88	155	13.64	396	34.4
$(63,61)^2$ RS	0.94	378	35.5	1312	27

Since RS codes have higher P and R for equivalent f_0, RS turbo decoder can reach a higher data rate than equivalent BCH turbo decoder. However, the increase in throughput cannot be considered regardless of the turbo decoder complexity.

6.3. Turbo product code comparison: throughput versus complexity

The efficiency η between the decoder throughput and the decoder complexity can be used to compare eBCH and RS turbo product codes. We have reported in Table 3 the code rate R, the parallelism degree P, the throughput T (Gbps), the complexity C (kgate) and the efficiency η (kbps/gate) for each code. All designs have been clocked at $f_0 = 100$ MHz for the computation of the throughput T. An average ratio of 3.5 between RS and BCH decoder efficiency is observed.

The good compromise between performance, throughput and complexity clearly makes RS product codes good candidates for next-generation PON and OTN. In particular, the $(31,29)^2$ RS product code is compatible with the 10 Gbps line rate envisioned for PON evolutions. Similarly, the $(63,61)^2$ RS product code can be used for data transport over OTN at 40 Gbps provided the turbo decoder is clocked at a frequency slightly higher than 100 MHz.

7. IMPLEMENTATION OF AN RS TURBO DECODER FOR ULTRA HIGH THROUGHPUT COMMUNICATION

An experimental setup based on FPGA devices has been designed in order to show that RS TPCs can effectively be used in the physical layer of 10 Gbps optical access networks. Based on the previous analysis, the $(31,29)^2$ RS TPC was selected since it offers the best compromise between performance and complexity for this kind of application.

7.1. 10 Gbps experimental setup

The experimental setup is composed of a board that includes 6 Xilinx Virtex-5 LX330 FPGAs [38]. A Xilinx Virtex-5 LX330 FPGA contains 51,840 slices that can emulate up to 12 million gates of logic. It should be noted that Virtex-5 slices are organized differently from previous generations. Each Virtex-5 slice contains four look up tables (LUTs) and four flip-flops instead of two LUTs and two flip-flops in previous generation devices. The board is hosted on a 64-bit, 66 MHz PCI bus that enables communication at full PCI bandwidth with a computer. An FPGA embedded memory block containing 10 encoded and noisy product code matrices is used to generate input data towards the turbo decoder. This memory block exchanges data with a computer thanks to the PCI bus.

One decoding iteration was implemented on each FPGA resulting in a 6 full-iteration turbo decoder as shown in Figure 10. Each decoding module corresponds to a full-parallel architecture dedicated to the decoding of a matrix of 31 × 31 coded soft values. We recall here that a coded soft value over GF(32) is mapped onto 5 LLR values, each LLR being quantized on 5 bits. Besides, the decoding process needs to access the 31 coded soft values from each of the matrices $\mathbf{R}$ and $\mathbf{R}_k$ during the three decoding phases of a half-iteration as explained in Section 4. For theses reasons, $31 \times 5 \times 5 \times 2 = 1,550$ bits have to be exchanged between the decoding modules during each clock period $f_0 = 65$ MHz. The board offers 200 chip to chip LVDS for each FPGA to FPGA interconnect. Unfortunately, this number of LVDS is insufficient to enable the transmission of all the bits between the decoding modules. To solve this implementation constraint, we have chosen to add SERializer/DESerializer (SERDES) modules for the parallel-to-serial conversions and for the serial-to-parallel conversions in each FPGA. Indeed, SERDES is a pair of functional blocks commonly used in high-speed communications to convert data between parallel data and serial interfaces in each direction. SERDES modules are clocked with $f_1 = 2 \times f_0 = 130$ MHz and operate at 8 : 1 serialization or 1 : 8 deserialization. In this way, all data can be exchanged between the different decoding modules. Finally, the total occupation rate of the FPGA that contains the more complex design (decoding module + two SERDES modules + memory block + PCI protocol module) is slightly higher than 66%. This corresponds to 34,215 Virtex-5 slices. Note that the decoding module represents only 37% of the total design complexity. More details about this are given in the next section.

Currently, a new design phase of the experimental setup is in progress. The objective is to include channel emulator and BER measurement facilities in order to verify decoding performance of the turbo decoder by plotting some BER curves as in our previous experimental setup [37].

7.2. Characteristics and performance of the implemented decoding module

A decoding module for one iteration is composed of 31 × 2 = 62 elementary decoders and 2 connection blocks. Each elementary decoder uses information quantized on 5 bits with $N_{\text{ep}} = 8$ error patterns and only 1 competing codeword. These reduced parameter values allow a decrease in the required area for a performance degradation which remains inferior to 0.5 dB. Thus a (31, 29) RS elementary decoder occupies 729 slice LUTs, 472 slice Flip-Flops and 3 BlockRAM of 18 Kbs. A connection block occupies only 2,325 slice LUTs. Computation resources of a decoding module take up 29,295 slice Flip-Flops and 49,848 slice LUTs. It means that the occupation rates are about 14% and 24% of a Xilinx Virtex-5 LX330 FPGA for slice registers and slice LUTs, respectively. Besides, memory resources for

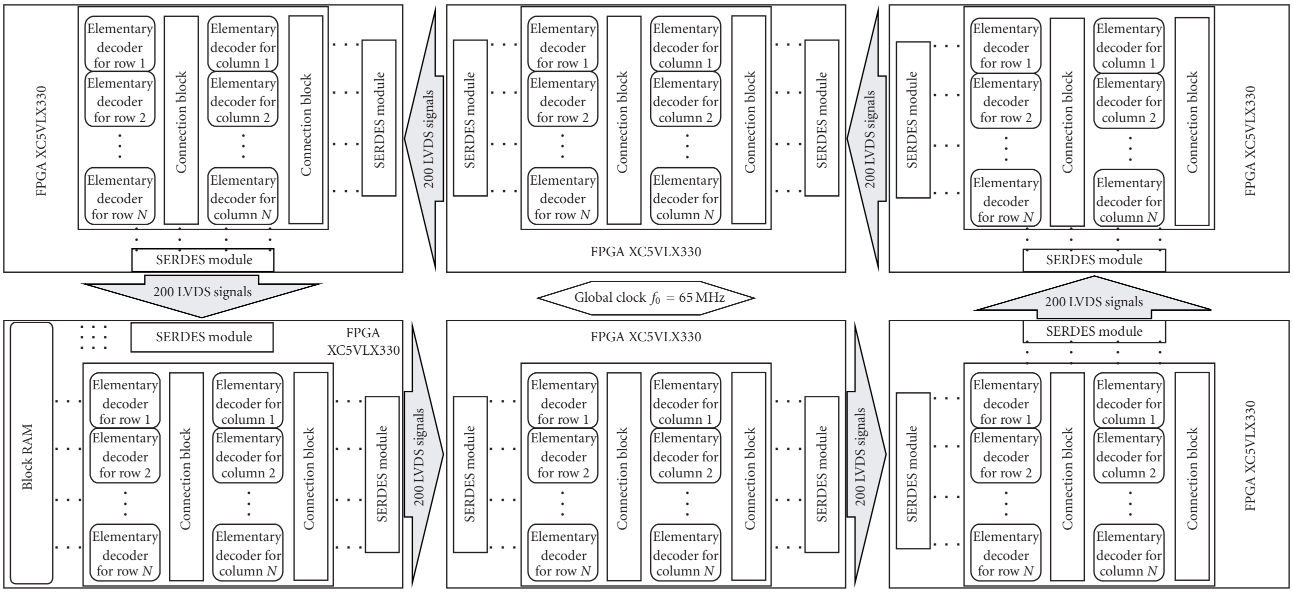
FPGA XC5VLX330
Elementary decoder for row 1
Elementary decoder for row 2
Elementary decoder for row N
Connection block
Elementary decoder for column 1
Elementary decoder for column 2
Elementary decoder for column N
Connection block
SERDES module
SERDES module
200 LVDS signals
SERDES module
Elementary decoder for row 1
Elementary decoder for row 2
Elementary decoder for row N
Connection block
Elementary decoder for column 1
Elementary decoder for column 2
Elementary decoder for column N
Connection block
SERDES module
FPGA XC5VLX330
200 LVDS signals
SERDES module
Elementary decoder for row 1
Elementary decoder for row 2
Elementary decoder for row N
Connection block
Elementary decoder for column 1
Elementary decoder for column 2
Elementary decoder for column N
Connection block
FPGA XC5VLX330
SERDES module
200 LVDS signals
Global clock f_0 = 65 MHz
200 LVDS signals
Block RAM
SERDES module
FPGA XC5VLX330
Elementary decoder for row 1
Elementary decoder for row 2
Elementary decoder for row N
Connection block
Elementary decoder for column 1
Elementary decoder for column 2
Elementary decoder for column N
Connection block
SERDES module
200 LVDS signals
FPGA XC5VLX330
SERDES module
Elementary decoder for row 1
Elementary decoder for row 2
Elementary decoder for row N
Connection block
Elementary decoder for column 1
Elementary decoder for column 2
Elementary decoder for column N
Connection block
SERDES module
200 LVDS signals
SERDES module
SERDES module
Elementary decoder for row 1
Elementary decoder for row 2
Elementary decoder for row N
Connection block
Elementary decoder for column 1
Elementary decoder for column 2
Elementary decoder for column N
Connection block
FPGA XC5VLX330

Figure 10: 10 Gbps experimental setup for turbo decoding of $(31,29)^2$ RS product code.

the decoding module take up 186 BlockRAM of 18 Kbits. It represents 32% of the total BlockRAM available in the Xilinx Virtex-5 LX330 FPGA. Note that one BlockRAM of 18 Kbits is allocated by the Xilinx tool ISE to memorize only $31 \times 5 \times 5 = 775$ bits in our design. The occupation rate of each BlockRAM of 18 Kbits is then only about 4%. Input data are clocked at $f_0 = 65$ MHz resulting in a data rate of $T_{\text{in}} = 10$ Gbps at the turbo-decoder input. By taking into account the code rate $R = 0.87$, the information rate becomes $T_{\text{out}} = 8.7$ Gbps. In conclusion, the implementation results showed that a turbo decoder dedicated to the $(31,29)^2$ RS product code can effectively be integrated to the physical layer of a 10 Gbps optical access network.

7.3. $(63,61)^2$ *RS TPC complexity estimation for a 40 Gbps transmission over OTN*

A similar prototype based on the $(63,61)^2$ RS TPC can be designed for 40 Gbps transmission over OTN. Indeed, the architecture of one decoding iteration is the same for the two RS TPCs considered in this work. For the $(63,61)^2$ RS product code, a decoding module for one iteration is now composed of $63 \times 2 = 126$ elementary decoders and 2 connection blocks. Logic syntheses were performed using the Xilinx tool ISE to estimate the complexity of a $(63,61)$ RS elementary decoder. This decoder occupies 1070 slice LUTs, 660 slice Flip-Flops, and 3 BlockRAM of 18 Kbs. These estimations immediately give the complexity of a decoding module dedicated to one iteration. Computation resources of a $(63,61)^2$ RS decoding module take up 83,160 slice Flip-Flops and 134,820 slice LUTs. The occupation rates are then about 40% and 65% of a Xilinx Virtex-5 LX330 FPGA for slice registers and slice LUTs, respectively. Memory resources of a $(63,61)^2$ RS decoding module take up 378 BlockRAM of 18 Kbits that represents 65% of the total BlockRAM available in the considered FPGA device. One BlockRAM of 18 Kbits is allocated by the Xilinx tool ISE to memorize only $63 \times 6 \times 5 = 1890$ bits. For a $(63,61)$ RS elementary decoder, the occupation rate of each BlockRAM of 18 Kbits is only about 10.5%.

8. CONCLUSION

We have investigated the use of RS product codes for forward-error correction in high-capacity fiber optic transport systems. A complete study considering all the aspects of the problem from code optimization to turbo product code implementation has been performed. Two specific applications were envisioned: 40 Gbps line rate transmission over OTN and 10 Gbps data transmission over PON. Algorithmic issues have been ordered and solved in order to design RS turbo product codes that are compatible with the respective requirements of the two transmission scenarios. A novel full-parallel turbo decoding architecture has been introduced. This architecture allows decoding of TPCs at data rates of 10 Gbps and beyond. In addition, a comparative study has been carried out between eBCH and RS TPCs in the context of optical communications. The results have shown that high-rate RS TPCs offer similar performance at reduced hardware complexity. Finally, we have described the successful realization of an RS turbo decoder prototype for 10 Gbps data transmission. This experimental setup demonstrates the practicality and also the benefits offered by RS TPCs in lightwave systems. Although only fiber optic communications have been considered in this work, RS TPCs may also be attractive FEC solutions for next-generation free-space optical communication systems.

ACKNOWLEDGMENTS

The authors wish to acknowledge the financial support of France Telecom R&D. They also thank Gérald Le Mestre

for his significant help during the experimental setup design phase. This paper was presented in part at IEEE International Conference on Communication, Glasgow, Scotland, in June 2007.

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near shannon limit error-correcting coding and decoding: turbo-codes 1,” in *Proceedings of the IEEE International Conference on Communications (ICC ’93)*, vol. 2, pp. 1064–1070, Geneva, Switzerland, May 1993.

[2] R. G. Gallager, “Low-density parity-check codes,” *IEEE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[3] D. J. Costello Jr. and G. D. Forney Jr., “Channel coding: the road to channel capacity,” *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1150–1177, 2007.

[4] S. Benedetto and G. Bosco, “Channel coding for optical communications,” in *Optical Communication: Theory and Techniques*, E. Forestieri, Ed., chapter 8, pp. 63–78, Springer, New York, NY, USA, 2005.

[5] T. Mizuochi, “Recent progress in forward error correction for optical communication systems,” *IEICE Transactions on Communications*, vol. E88-B, no. 5, pp. 1934–1946, 2005.

[6] T. Mizuochi, “Recent progress in forward error correction and its interplay with transmission impairments,” *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 12, no. 4, pp. 544–554, 2006.

[7] “Forward error correction for high bit rate DWDM submarine systems,” International Telecommunication Union ITU-T Recommandation G.975.1, February 2004.

[8] R. Pyndiah, A. Glavieux, A. Picart, and S. Jacq, “Near optimum decoding of product codes,” in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM ’94)*, vol. 1, pp. 339–343, San Francisco, Calif, USA, November-December 1994.

[9] K. Gracie and M.-H. Hamon, “Turbo and turbo-like codes: principles and applications in telecommunications,” *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1228–1254, 2007.

[10] J. Cuevas, P. Adde, S. Kerouedan, and R. Pyndiah, “New architecture for high data rate turbo decoding of product codes,” in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM ’02)*, vol. 2, pp. 1363–1367, Taipei, Taiwan, November 2002.

[11] C. Jégo, P. Adde, and C. Leroux, “Full-parallel architecture for turbo decoding of product codes,” *Electronics Letters*, vol. 42, no. 18, pp. 1052–1054, 2006.

[12] T. Mizuochi, Y. Miyata, T. Kobayashi, et al., “Forward error correction based on block turbo code with 3-bit soft decision for 10-Gb/s optical communication systems,” *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 10, no. 2, pp. 376–386, 2004.

[13] I. B. Djordjevic, S. Sankaranarayanan, S. K. Chilappagari, and B. Vasic, “Low-density parity-check codes for 40-Gb/s optical transmission systems,” *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 12, no. 4, pp. 555–562, 2006.

[14] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, The Netherlands, 1977.

[15] R. E. Blahut, *Algebraic Codes for Data Transmission*, Cambridge University Press, Cambridge, UK, 2003.

[16] O. Aitsab and R. Pyndiah, “Performance of Reed-Solomon block turbo code,” in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM ’96)*, vol. 1, pp. 121–125, London, UK, November 1996.

[17] D. Chase, “A class of algorithms for decoding block codes with channel measurement information,” *IEEE Transactions on Information Theory*, vol. 18, no. 1, pp. 170–182, 1972.

[18] P. Adde and R. Pyndiah, “Recent simplifications and improvements in block turbo codes,” in *Proceedings of the 2nd International Symposium on Turbo Codes and Related Topics*, pp. 133–136, Brest, France, September 2000.

[19] R. Pyndiah, “Iterative decoding of product codes: block turbo codes,” in *Proceedings of the 1st International Symposium on Turbo Codes and Related Topics*, pp. 71–79, Brest, France, September 1997.

[20] J. Briand, F. Payoux, P. Chanclou, and M. Joindot, “Forward error correction in WDM PON using spectrum slicing,” *Optical Switching and Networking*, vol. 4, no. 2, pp. 131–136, 2007.

[21] R. Zhou, R. Le Bidan, R. Pyndiah, and A. Goalic, “Low-complexity high-rate Reed-Solomon block turbo codes,” *IEEE Transactions on Communications*, vol. 55, no. 9, pp. 1656–1660, 2007.

[22] P. Sweeney and S. Wesemeyer, “Iterative soft-decision decoding of linear block codes,” *IEE Proceedings: Communications*, vol. 147, no. 3, pp. 133–136, 2000.

[23] M. Lalam, K. Amis, D. Leroux, D. Feng, and J. Yuan, “An improved iterative decoding algorithm for block turbo codes,” in *Proceedings of the IEEE International Symposium on Information Theory (ISIT ’06)*, pp. 2403–2407, Seattle, Wash, USA, July 2006.

[24] W. W. Peterson, “Encoding and error-correction procedures for the Bose-Chaudhuri codes,” *IEEE Transactions on Information Theory*, vol. 6, no. 4, pp. 459–470, 1960.

[25] D. Gorenstein and N. Zierler, “A class of error correcting codes in p^m symbols,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 9, no. 2, pp. 207–214, 1961.

[26] S. A. Hirst, B. Honary, and G. Markarian, “Fast Chase algorithm with an application in turbo decoding,” *IEEE Transactions on Communications*, vol. 49, no. 10, pp. 1693–1699, 2001.

[27] G. Bosco, G. Montorsi, and S. Benedetto, “Soft decoding in optical systems,” *IEEE Transactions on Communications*, vol. 51, no. 8, pp. 1258–1265, 2003.

[28] Y. Cai, A. Pilipetskii, A. Lucero, M. Nissov, J. Chen, and J. Li, “On channel models for predicting soft-decision error correction performance in optically amplified systems,” in *Proceedings of the Optical Fiber Communications Conference (OFC ’03)*, vol. 2, pp. 532–533, Atlanta, Ga, USA, March 2003.

[29] G. P. Agrawal, *Lightwave Technology: Telecommunication Systems*, John Wiley & Sons, Hoboken, NJ, USA, 2005.

[30] L. M. G. M. Tolhuizen, “More results on the weight enumerator of product codes,” *IEEE Transactions on Information Theory*, vol. 48, no. 9, pp. 2573–2577, 2002.

[31] M. El-Khamy and R. Garello, “On the weight enumerator and the maximum likelihood performance of linear product codes,” *IEEE Transaction on Information Theory*, arXiv:cs.IT/0601095 (preprint) Jan 2006.

[32] R. Le Bidan, R. Pyndiah, and P. Adde, “Some results on the binary minimum distance of Reed-Solomon codes and block turbo codes,” in *Proceedings of the IEEE International Conference on Communications (ICC ’07)*, pp. 990–994, Glasgow, Scotland, June 2007.

[33] P. Adde, R. Pyndiah, and S. Kerouedan, “Block turbo code with binary input for improving quality of service,” in *Multiaccess, Mobility and Teletraffic for Wireless Communications*,

X. Lagrange and B. Jabbari, Eds., vol. 6, Kluwer Academic Publishers, Boston, Mass, USA, 2002.

[34] Z. Chi and K. K. Parhi, "High speed VLSI architecture design for block turbo decoder," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '02)*, vol. 1, pp. 901–904, Phoenix, Ariz, USA, May 2002.

[35] D. H. Lawrie, "Access and alignment of data in an array processor," *IEEE Transactions on Computers*, vol. C-24, no. 12, pp. 1145–1155, 1975.

[36] S. Kerouedan and P. Adde, "Implementation of a block turbo decoder on a single chip," in *Proceedings of the 2nd International Symposium on Turbo Codes and Related Topics*, pp. 243–246, Brest, France, September 2000.

[37] C. Leroux, C. Jégo, P. Adde, and M. Jezequel, "Towards Gb/s turbo decoding of product code onto an FPGA device," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '07)*, pp. 909–912, New Orleans, La, USA, May 2007.

[38] http://www.dinigroup.com/DN9000k10PCI.php.