# On the higher efficiency of parallel Reed-Solomon turbo-decoding

Cédric Leroux, Christophe Jego, Patrick Adde, Michel Jezequel

# On the higher efficiency of parallel Reed-Solomon turbo-decoding

Camille LEROUX, Christophe JEGO, Patrick ADDE, Michel JEZEQUEL
Institut TELECOM, TELECOM Bretagne, CNRS Lab-STICC FRE 3167
firstname.lastname@telecom-bretagne.eu

***Abstract*— In this paper, we demonstrate the higher hardware efficiency of Reed-Solomon (RS) parallel turbo decoding compared with BCH parallel turbo decoding. Based on an innovative architecture, this is the first implementation of fully parallel RS turbo decoder. A performance analysis is performed showing that RS Block Turbo Codes (RS-BTC) have decoding performance equivalent to Bose Ray-Chaudhuri Hocquenghem-Block Turbo Codes (BCH-BTC). A ratio between the decoder throughput and the decoder area is used to show the higher efficiency of the RS full parallel turbo decoder. Finally an implementation of a (31,29)$^2$ RS block turbo decoder on a high performance board including 6 Xilinx Virtex5 FPGAs is detailed. The resulting turbo decoder has an information throughput above 6Gb/s while the working frequency is only 45MHz. It shows that RS BTC are an attractive solution for low cost Gb/s fiber optical communications.**

## I. Introduction

Since their invention, turbo codes and block turbo codes [1][2] have been used in most challenging telecommunications systems such as cellular communication systems, digital video broadcasting or wireless metropolitan area networks. It has been demonstrated that block turbo codes (BTC) are well adapted codes for a very high speed context [3]. A high throughput BCH turbo decoder implementation was proposed for Optical Transport Networks (OTN) [4]. However, in contrast to OTN, some very high speed applications such as Passive Optical Networks (PON) require a low-cost optical infrastructure with good performance. A PON is point-to-multipoint optical network in which passive splitters are used to connect a single Optical Line Termination (OLT) to many users. In such a context, Forward Error Correction (FEC) were recently introduced to compensate the signal degradation due to low cost optical components and to increase the number of users connected to the same OLT [5]. In PONs, even if decoding performance constraints are not as strict as in OTN, increasing the coding gain reduces the cost of the infrastructure per user. Current normalized solutions (B-PON,G-PON) include an optional (255,239) RS code giving a 3dB coding gain at 2.5Gb/s. Next generation PONs will use Wavelength Division Multiplexing (WDM) technology in order to increase both data rate (above 10Gb/s) and the number of users. In such a context, FECs with higher coding gain and increased throughput will be needed. So far, no realization or architecture of a very high speed BTC decoder has been proposed for PONs. In this article, we show that BTCs present notable performance improvement compared to the (255,239) RS code. It is also demonstrated that parallel RS turbo decoding gives better hardware efficiency compared to parallel BCH turbo decoding.

This work was supported with funds from the Brittany Region under the competitiveness pole Interaccess

This paper first recalls the basic principle of decoding for product codes: their construction and turbo decoding. In section III, we propose a performance study highlighting the good performance of BTC decoding compared to (255,239) RS. Section IV presents the highly parallel architecture of the BTC decoder. In section V, a complexity and throughput analysis shows the better hardware efficiency of RS turbo decoding. Finally, section VI describes a prototype of an RS(31,29)$^2$ turbo decoder in an FPGA experimental setup.

## II. BTC coding and decoding principle

A product code is a concatenation of systematic linear block codes. The product code inherits the properties of the elementary codes that it is composed of. Let us consider two identical systematic linear block codes $C$ having parameters $(n, k, \delta)$, where $n$ and $k$ stand for code length, the number of information symbols and the minimum Hamming distance respectively. The parameters of the resulting product code are given by: $n_p = n^2$, $k_p = k^2$, $\delta_p = \delta^2$ and $R_p = R^2$ (code rate). Thus, it is possible to construct powerful product codes using linear block codes. The turbo decoding process involves sequentially decoding the rows and columns of the product code matrix by exchanging soft information. This decoding is called Soft Input Soft Output (SISO) decoding [2].

The SISO decoding algorithm can be summarized as follows: do the syndrome based algebraic decoding of the $Tv$ test vectors that have been built by inversions of the $Lr$ least reliable symbols in the received word $[R']_{it}$. The $Lr$ least reliable symbols have been previously searched for. After that, for each corrected test vector, compute the square Euclidean distance between $[R']_{it}$ and the test vector considered. Select the Decided codeWord (DW) having the minimal distance from $[R']_{it}$ and choose $Cw$ competitor words having the closest distance to $[R']_{it}$. Finally, compute the soft output information $[R']_{it+1} = [R]_{it} + \alpha_{it}[W]_{it}$ for each symbol where $[W]_{it}$ is the extrinsic information and $\alpha_{it}$ is a coefficient that allows decoding decisions to be damped during the first iterations. It should be noted that decoding parameters $Lr$, $Tv$, and $Cw$ has a notable effect on performance. Decoding a BCH or RS code only differs in the algebraic decoding phase.

## III. Decoding performance of BTC versus RS code

In this section, the BTC contribution in PONs is highlighted by comparing different BCH-BTC and RS-BTC with a correction power of one to the widely used (255,239) RS code. SISO

decoding complexity is particularly increased for correction powers larger than one. Indeed, higher correction power codes require higher complexity algorithm for algebraic decoding such as the Berlekamp-Massey algorithm [6]. On the contrary, algebraic decoding for a correction power of one enables a direct computation of both locator and error estimator polynomials. Since PONs require low complexity systems, we choose to only consider BTCs with a correction power of one.

Figure 1 gives the decoding performance of different BTCs compared with the (255,239) RS over a Gaussian channel with an On-Off Keying (OOK) mapping. The OOK uncoded curve is also represented. BTC decoding uses the Chase-Pyndiah algorithm with 5 least reliable bits ($Lr$), 16 test vectors ($Tv$) and 3 concurrent words ($Cw$). The soft information is quantized with at least 5 bits ensuring that performance is very close to the floating point model. The comparison between code performance is thus made more consistent.

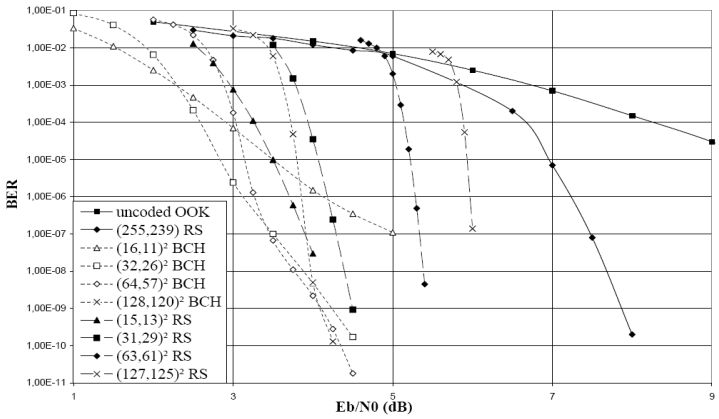


Fig. 1. BTC and (255,239) RS decoding performance

In a general manner, BTCs offer a higher coding gain than (255,239) RS. Decoding performance depends on the type of elementary code (BCH or RS), the code size $n$ and the code rate $R$. Regarding codes with an equivalent rate ($e.g.$ $(31,29)^2$ RS and $(128,120)^2$ BCH), BCH-BTC have a better slope than RS-BTC but are limited in terms of asymptotic performance. Indeed, performance of BCH product codes is penalized by their asymptotic bound. In contrast, RS product codes push down the error floor and the corresponding performance has no change in slope [7]. Thus, RS-BTC tend to be more efficient in low Bit Error Rate (BER) area as required in optical transmission systems.

In the light of coding gain values, five codes remain good candidates in terms of decoding performance: $(32,26)^2$ BCH, $(64,57)^2$ BCH and $(128,120)^2$ BCH converge with the same trend having a fairly similar slope. $(15,13)^2$ RS and $(31,29)^2$ RS have good asymptotic performance with coding gain around 7.5dB at BER=$10^{-10}$. Architectural issues will determine which code is more efficient in terms of complexity for equivalent performance.

## IV. Ultra high speed architecture

In this section, a highly parallel architecture of the turbo decoder involving a duplication of elementary decoders connected by interconnection networks is described.

### A. Full-parallel architecture of the turbo-decoder

Figure 2 depicts a full-parallel architecture for the product code turbo decoding proposed in [8]. The major advantage of this full-parallel architecture is that it enables the memory block of $4 \times q \times n^2$ symbols between each half-iteration to be replaced by simple interconnection networks.

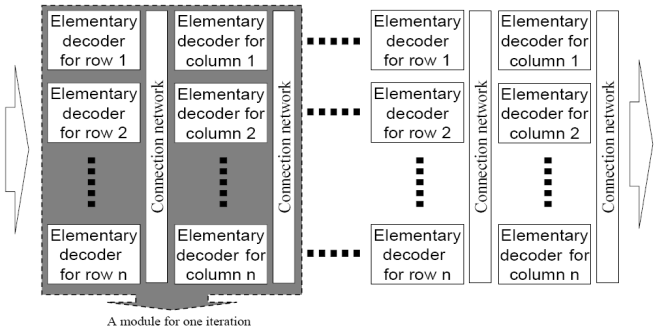


Fig. 2. Full-parallel decoding of product codes

Each duplicated module is dedicated to one iteration. However, it is possible to process several iterations by a same module. In our approach, $2n$ elementary decoders and 2 connection networks are necessary for one module. In fact, the full-parallel turbo decoder complexity mainly depends on the complexity of the elementary decoder and the code size $n$.

### B. Elementary SISO decoder

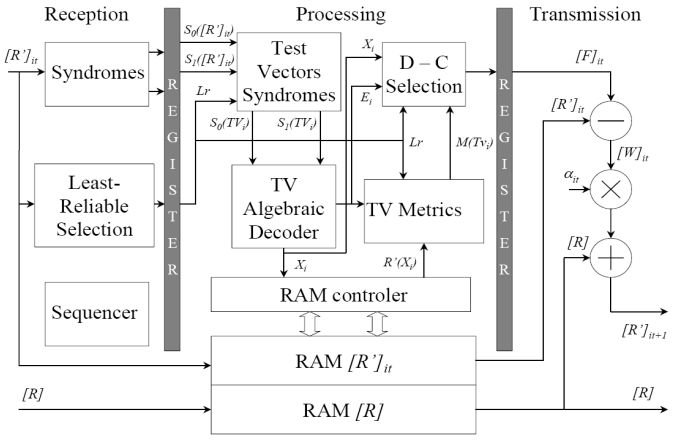


Fig. 3. RS SISO decoder architecture

The SISO decoder architecture shown in figure 3 is structured in three pipelined stages identified as reception, processing and transmission units. The reception and transmission units process one symbol per clock period while the processing unit process $Tv$ test vectors. The reception unit computes syndromes $S_0$, $S_1$ and locate the $Lr$ least reliable bits of the word received $[R']_{it}$. This unit is also in charge of the decoder sequencing by generating control signals. The processing unit computes the error position $X_i$ and value $E_i$ of the currently processed test vector from the previously computed syndromes $S_0(Tv_i)$ and $S_1(Tv_i)$. The Euclidean distance values (amso called metrics) $M(Tv_i)$ of each vector is then calculated enabling the D-C bloc to select the decided word (the one having the minimum metric value) and the competitor ones in order to generate the

new reliability $[F]_{it}$. The transmission unit computes the scaled extrinsic information $\alpha_{it}[W]_{it}$ and the soft output $[R']_{it+1}$. A new $(m \times q) - bits$ symbol is then transmitted during each clock period. $q$ is the number of quantization bits of the soft information. The decoding process needs to access the $[R]_{it}$ and $[R']_{it}$ values during the three decoding phases. For this reason, these words are implemented in six Random Access Memories (RAM) of size $n \times m \times q$ controlled by a finite state machine. Since the error position $X_i$ is computed in $GF(32)$, the RAM is also adressed in $GF(32)$. In our architecture, this SISO decoder is the main elementary block of the BTC decoder, consequently, low-complexity SISO decoders are required.

## V. Complexity and throughput analysis of the full-parallel BTC decoder

Increasing throughput regardless of turbo decoder complexity is not relevant. In order to compare the throughput and complexity of RS and BCH turbo decoders, the efficiency of the parallel BTC decoder is defined as

$$\eta = \frac{T}{C} \tag{1}$$

$T$ is the throughput $(b/s)$ and $C$ the complexity $(gates)$ of the design. An efficient architecture will have a high $\eta$ ratio meaning a high throughput with low hardware complexity. This study aims to determine efficiency for both BCH-BTC and RS-BTC in order to compare them.

### A. BTC decoder throughput analysis

In a fully pipelined architecture, the throughput of the block turbo decoder can be defined as

$$T = P \times R \times f_0 \tag{2}$$

where P is the parallelism degree and corresponds to the number of generated bits per clock period $t_0$, $R$ is the code rate and $f_0 = \frac{1}{t_0}$ the maximum frequency of an elementary SISO decoder. Ultra high throughput can be reached by increasing these three parameters.

$R$ is a parameter that exclusively depends on the code considered. Thus, using codes with a higher code rate ($e.g.$ RS codes) would provide improved throughput. 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 is then expressed as

$$P = n \times m \tag{3}$$

with $m$ the number of bits per symbol. Considering an RS code with a code size $n_{RS}$, $m_{RS} = log(n_{RS}+1)$ whereas BCH codes only have $m_{BCH} = 1$. Therefore, enhanced parallelism degree can be obtained by using non-binary codes ($e.g.$ RS codes) with larger $n$ value. 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 stage within each elementary SISO decoder as shown in figure 3. In the considered architecture, RS and BCH SISO decoders with equivalent code size have equivalent working frequency $f_0$ since RS decoding is performed by introducing some local parallelism at symbol level. This result was verified during logic syntheses. The main drawback of pipelining elementary decoders is the extra-complexity generated by internal memory requirement (registers and RAM). Since RS codes have higher $P$ and $R$ for equivalent $f_0$, RS turbo decoder can potentially reach a higher data rate than BCH. However, the increase in throughput can not be considered regardless of the turbo decoder complexity.

### B. BTC decoder complexity analysis

*1) Computation resource complexity:* The computational resources of a SISO decoder are split into three pipelined stages. The reception and transmission stages have a complexity in $O(\log(n))$. For these two stages, moving from a BCH code to a RS code, both complexity and throughput are increased with a factor $\log(n_{RS}+1)$. As a result, efficiency is constant in these parts of the decoder. However, the processing stage has a hardware complexity in $O(Tv)$. Consequently, the increase in the local parallelism rate has no influence on the area of this stage and thus increases the efficiency of a RS SISO decoder.

In order to verify those general considerations, BCH SISO decoder area was estimated thanks to a complexity model which can derive an estimation of the gate count for any code size and any decoding parameter set [9]. RS turbo decoders for code $(63,61)^2$ RS, $(31,29)^2$ RS and $(15,13)^2$ RS were described in HDL language and synthesized. Logic syntheses were performed using Synopsys Design Compiler with a ST-microelectronics 90nm CMOS process. All designs were clocked at 100MHz. Table I shows the higher efficiency of the computational resources among the parallel BTC decoder. Indeed, when comparing $(31,29)^2$ RS and $(63,61)^2$ RS with the $(128,120)^2$ BCH code, it can be noted that RS codes enable higher throughput with a lower computational resource complexity. The same observation can be done considering $(15,13)^2$ RS and $(32,26)^2$ BCH codes; the throughput is doubled while the complexity is divided by 1.8. RS turbo decoding appears to use computational resources in a more efficient manner.

TABLE I
Computational complexity for different parallelism rate

| $P$ | Code | $T(Gb/s)$ | $C_{comput.}(kgates/\frac{1}{2}iter)$ |
|---|---|---|---|
| 32 | $(32,26)^2$ BCH | 2.11 | 89 |
| 60 | $(15,13)^2$ RS | 4.5 | 50 |
| 64 | $(64,57)^2$ BCH | 5.06 | 200 |
| 128 | $(128,120)^2$ BCH | 11.26 | 446 |
| 155 | $(31,29)^2$ RS | 13.64 | 133 |
| 378 | $(63,61)^2$ RS | 35.5 | 378 |

However, even if our architecture enables interleaver memory to be removed, more than a half of a parallel BTC decoder is composed of internal memory.

*2) Memory resource complexity:* A half-iteration of a parallel block turbo decoder contains $n$ RAM banks of $n \times m \times q$ bits. The internal memory complexity of a parallel decoder for a half-iteration can be approximated by

$$S_{RAM} \simeq \gamma \times n^2 \times m \times q \tag{4}$$

where $\gamma$ is a technological parameter specifying the number of equivalent gate counts per memory bit. Using equation 3, it can also be expressed as

$$S_{RAM} = \gamma \times \frac{P^2}{m} \times q \tag{5}$$

Calculating the SISO memory area for both BCH and RS gives the following ratio:

$$\frac{S_{RAM}(BCH)}{S_{RAM}(RS)} = m_{RS} = \log(n_{RS}+1) \quad (6)$$

It shows that RS block turbo decoders have lower memory complexity for a given parallelism rate $P$. This was confirmed by memory area estimations results showed on Figure 4. RAM area of BCH and RS turbo decoders for a half-iteration and different parallelism degrees is plotted using a memory area estimation model provided by ST-Microelectronics.

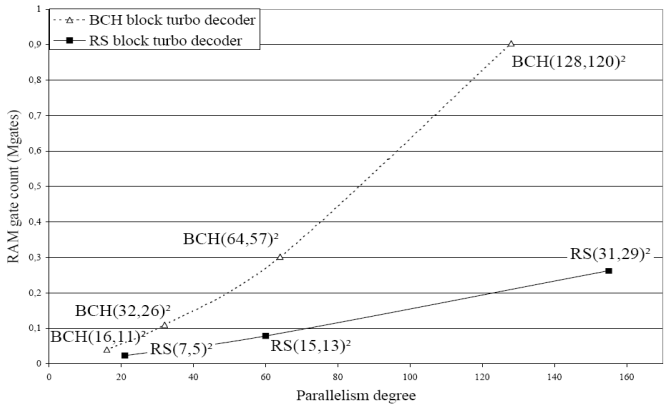


Fig. 4. Comparison of internal RAM complexity

We can observe that higher $P$ ($i.e.$ higher $T$) can be obtained with less memory when using an RS turbo decoder. Full-parallel decoding of RS codes appears to be more memory-efficient than BCH code turbo decoding.

*3) Efficiency comparison:* Table II sumarizes the code rate $R$, the parallelism rate $P$, the throughput $T$ $(Gb.s^{-1})$, the complexity $C$ $(kgates)$ and the efficiency $\eta$ $(kb.s^{-1}.gate^{-1})$ of a half-iteration of parallel BTC decoder for each studied code. An average ratio of 3.5 between RS and BCH decoders efficiency is observed. In some cases, RS codes enable higher throughput with lower complexity.

TABLE II

SELECTED BTC DECODERS HARDWARE EFFICIENCY

| Code | $R$ | $P$ | $T$ | $C$ | $\eta$ |
|---|---|---|---|---|---|
| $(32,26)^2$ BCH | 0.66 | 32 | 2.1 | 201 | 10.5 |
| $(64,57)^2$ BCH | 0.79 | 64 | 5.1 | 508 | 10.0 |
| $(128,120)^2$ BCH | 0.88 | 128 | 11.3 | 1361 | 8.2 |
| $(15,13)^2$ RS | 0.75 | 60 | 4.5 | 128 | 35.0 |
| $(31,29)^2$ RS | 0.88 | 155 | 13.6 | 396 | 34.4 |
| $(63,61)^2$ RS | 0.94 | 378 | 35.5 | 1312 | 27.0 |

The good compromise between performance, throughput and complexity clearly makes RS codes good candidates for high throughput applications such as next generation PON technologies.

## VI. FPGA IMPLEMENTATION

The RS$(31,29)^2$ code was selected to be implemented on a high performance platform. The board includes 6 XilinxVirtex5 LX330 FPGAs [10]. An embedded memory containing 40 encoded and noisy product code matrices is used to generate input data towards the turbo decoder. One decoding iteration was implemented on each FPGA resulting in a 6 full-iteration block turbo decoder. Each decoder uses information quantized on 5 bits with 8 test vectors and 1 concurrent word. These reduced parameter values enable a decrease in the area for a performance degradation less than 0.5dB. The occupation rate of each FPGA is slightly higher than 50%. Input data are clocked at $f$=45MHz resulting in an input throughput of $T_{in}$=7Gb/s. Taking into account the code rate $R$=0.875, the output throughput becomes $T_{out}$=6.13Gb/s. The current working frequency is limited by the available bandwidth between two FPGAs. A full decoding iteration can operate above 50MHz.

Moreover, no back-end optimization has yet been performed. Using partitioning or intelligent place and route strategy will again increase working frequency and throughput above 10Gb/s.

## VII. CONCLUSION

This article shows the advantages of RS product codes decoding in an ultra high speed context. We first have highlighted the good performance of RS-BTCs compared to both BCH-BTC and (255,239) RS. Then the better hardware efficiency of RS parallel turbo decoding has been demonstrated especially in terms of internal memory requirements. Compared to BCH, the RS turbo decoder can reach a higher data rate for equivalent complexity. Finally, an FPGA implementation of a $(31,29)^2$ RS parallel turbo decoder shows that RS turbo decoding can reach several Gb/s. To the best of our knowledge, this is the first multi-gigabit implementation of a RS turbo-decoder. The complexity study showed that most of the area of the block turbo decoder is composed of SISO decoder internal memory resources. Consequently a study for proposing architectures with low-memory requirements in BTC decoding is in progress.

## REFERENCES


[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Communications, 1993. ICC 93. Geneva. Technical Program, Conference Record, IEEE International Conference on*, vol. 2, 23-26 May 1993, pp. 1064–1070vol.2.

[2] R. Pyndiah, A. Glavieux, A. Picart, and S. Jacq, "Near optimum decoding of product codes," in *Global Telecommunications Conference, 1994. GLOBECOM '94. 'Communications: The Global Bridge'., IEEE*, 28 Nov.-2 Dec. 1994, pp. 339–343vol.1.

[3] Z. Chi and K. Parhi, "High speed vlsi architecture design for block turbo decoder," in *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on*, vol. 1, 26-29 May 2002, pp. I–901–I–904vol.1.

[4] H. Tagami, T. Kobayashi, Y. Miyata, K. Ouchi, K. Sawada, K. Kubo, K. Kuno, H. Yoshida, K. Shimizu, T. Mizuochi, and K. Motoshima, "A 3-bit soft-decision ic for powerful forward error correction in 10-gb/s optical communication systems," *Solid-State Circuits, IEEE Journal of*, vol. 40, no. 8, pp. 1695–1705, Aug. 2005.

[5] P. C. J. Briand, F. Payoux and M. Joindot, "Forward error correction in wdm pon using spectrum slicing," *Optical Switching and Networking*, vol. Volume 4, pp. Issue 2,Pages 131–136, June 2007.

[6] E. R. Berlekamp, *Algebraic Coding Theorie.* revised edition, Aegean, 1984.

[7] R. Zhou, R. Le Bidan, R. Pyndiah, and A. Goalic, "Low-complexity high-rate reed-solomon block turbo codes," *Communications, IEEE Trans. on*, vol. 55, pp. 1656–1660, 2007.

[8] C. Jego, P. Adde, and C. Leroux, "Full-parallel architecture for turbo decoding of product codes," in *Electronics Letters*, vol. 42, no. 18, 31 August 2006, pp. 55–56.

[9] C. Leroux, C. Jego, P. Adde, and M. Jezequel, "Towards gb/s turbo decoding of product code onto an fpga device," in *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, 27-30 May 2007, pp. 909–912.

[10] [Online]. Available: http://www.dinigroup.com/DN9000k10PCI.php