

Brief Contributions

Analysis of Errors and Erasures in Parity Sharing RS Codes

G.C. Cardarilli, *Member, IEEE*, S. Pontarelli,
M. Re, *Member, IEEE*, and
A. Salsano, *Member, IEEE*

Abstract—Reed Solomon (RS) codes are widely used to protect information from errors in transmission and storage systems. Most of the RS codes are based on $GF(2^8)$ Galois Fields and use a byte to encode a symbol providing codewords up to 255 symbols. Codewords with more than 255 symbols can be obtained by using $GF(2^m)$ Galois Fields with $m > 8$, but this choice increases the complexity of the encoding and decoding algorithms. This limitation can be superseded by introducing Parity Sharing (PS) RS codes that are characterized by a greater flexibility in terms of design parameters. Consequently, a designer can choose between different PS code implementations in order to meet requirements such as Bit Error Rate (BER), hardware complexity, speed, and throughput. This paper analyzes the performance of PS codes in terms of BER with respect to the code parameters, taking into account either random error or erasure rates as two independent probabilities. This approach provides an evaluation that is independent of the communication channel characteristics and extends the results to memory systems in which permanent faults and transient faults can be modeled, respectively, as erasures and random errors. The paper also provides hardware implementations of the PS encoder and decoder and discusses their performances in terms of hardware complexity, speed, and throughput.

Index Terms—Reliability, testing, and fault tolerance, error control codes, coding and information theory.

1 INTRODUCTION

ERROR Correction Codes (ECC) are widely used to protect data transmitted over noisy channels since they are able to detect and correct large classes of errors in the coded data stream. The choice of the suitable ECC depends on the maximum number of errors the code is able to detect and correct, the error type (burst, random, etc.), and the speed performance of the coding and decoding units. The Reed-Solomon code is an ECC used in a large class of applications. The complexity of the RS encoder and decoder architectures increases both with the symbol size and with the codeword length. To avoid complexity growth, PS codes have been proposed for telecommunication applications [1] and for magnetic recording systems [2]. PS codes are obtained by concatenating two different RS codes. The performance of the overall code with respect to the Bit Error Rate (BER) depends on the parameters of each RS code used to build the PS code. Moreover, in PS codes, an additional parameter M is defined [1] and, therefore, more parameters are available to the designer to match the design specifications. In the above-cited papers, the suitability of PS codes for high performance hardware implementations is suggested, but neither the hardware implementation nor a comparison of the performance is presented. This paper is organized as follows: Section 2 introduces RS and PS codes, while Section 3 presents an

evaluation of the relationship between the BER and erasures and random errors rates. In Section 4, the BER with respect to the various design parameters has been evaluated. Sections 5 and 6 illustrate, respectively, the encoder and decoder design, while, in Section 7, conclusions are drawn.

2 REED SOLOMON AND PARITY-SHARING CODES

An RS codeword is composed of n symbols of m bits, conventionally of $n \leq 2^m - 1$, and the dataword length is composed of k symbols, i.e., the redundancy added by the code is $n - k$ symbols [6]. In RS codes, $r = k/n$ is called the code rate. An $RS(n, k)$ code is capable of correcting up to $2 \cdot e + s \leq n - k$ errors, where s is the number of erasures and e is the number of random errors. An erasure indicates a position in the codeword for which the codeword symbol is unknown. This position is made available by processing information related to the transmission channel that is called “channel side information” [7]. The construction of a two-level PS code is similar to a product code. A group of K_2 systematic $RS(N_1, K_1)$ codewords are stacked together and the last $N_1 - M$ columns are reencoded by using an $RS(N_2, K_2)$ code. Only the first $M > K_1$ symbols of each (N_1, K_1) codeword and the $(N_1 - M) \times (N_2 - K_2)$ shared-parities are transmitted (see Fig. 1).

The code rate r of a parity-sharing code is

$$r = \frac{K_1 K_2}{K_2 M + (N_1 - M)(N_2 - K_2)} \quad (1)$$

and a rate gain is achieved if $N_2 < 2K_2$, while using a code $RS(2K_2, K_2)$, the code rate of the PS code is $\frac{K_1}{N_1}$. Therefore, in PS codes, five parameters can be changed in order to match the performance of the target application. The decoding of a two-level PS code requires three steps:

1. Decoding of the row codewords. In this step, the $N_1 - M$ untransmitted symbols are handled by the decoder-like erasures.
2. Decoding of the $(N_1 - M)$ column codewords. In this step, the algorithm tries to correct any error and fill in the missing symbols that could not be decoded in the first row decoding step.
3. Redecoding of the row codewords. The rows are decoded again to fill in any remaining parities and recover the original codewords.

A PS code can handle the following error conditions:

1. $2e_i + s_i \leq N_1 - K_1$, where s_i and e_i are the number of erasures and random errors in the i row codeword, respectively.
2. $2e_j + s_j \leq N_2 - K_2$, where s_j and e_j are the number of erasures and random errors in the j column codeword, respectively.
3. $2e' + s' \leq N_1 - K_1$, where s' is the number of rows with $2e_i + s_i > M - K_1$ and e' is the number of rows with $e_i > M - K_1$.

The motivation for using PS codes is given in [1]. In particular, the author underlines that, for the majority of the cases, only a part of the check symbols are needed to recover the correct codeword, i.e., only in a few cases are all of the check symbols needed to recover the correct codeword. Moreover, the author explains the performance of long Reed-Solomon codes, showing that, with long codewords, they operate in a region in which the law of large numbers can be applied. This implies that, if the provided redundancy increases up to the overall error rate $2 \cdot e + s$, a very

• The authors are with the Department of Electronic Engineering, University of Rome “Tor Vergata,” Rome, Italy.
E-mail: {cardarilli, pontarelli, re, salsano}@ing.uniroma2.it.

Manuscript received 1 Dec. 2005; revised 27 Oct. 2006; accepted 16 Jan. 2007; published online 18 July 2007.

Recommended for acceptance by B. Bose.

For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TC-0427-1205.

Digital Object Identifier no. 10.1109/TC.2007.70773.

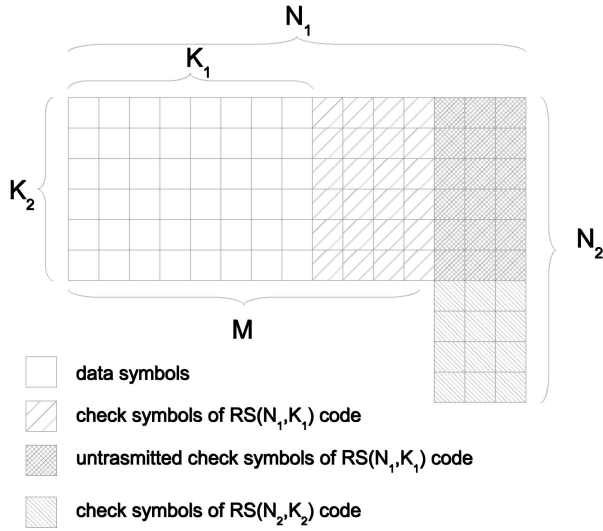


Fig. 1. Parity sharing RS codeword.

rapid drop in the probability of the decoder failure is obtained. Given the overall error rate, if shorter codes are used, a more gradual drop is obtained if the number of check symbols increases up to $2 \cdot e + s$.

3 PARITY SHARING CODES WITH ERRORS AND ERASURES

In this section, the PS code performances are evaluated by computing the BER with respect to the different PS code parameters starting from the error conditions described in the previous section. An expression for the PS codes BER is given in [1] and [2]. In this paper, these results have been extended by taking into account either the random error or the erasure events as two independent probabilities. This is necessary in order to provide a BER evaluation that is

- independent of the communication channel characteristics and
- useful for the analysis of memory systems in which permanent faults can be treated like erasures and transient faults [3] can be seen as random errors.

The calculations presented in this paper are partially based on the discussions presented in [4] and [5].

In the following, p and λ are defined as the error and erasures probabilities, while $P_n(p, \lambda)$ is the probability that an n length RS codeword fails to decode, i.e., the probability that $i \in [0, n]$ random errors and $j \in [0, n]$ erasures, with $2i + j \geq n - k + 1$, occurs.

The bounds for i and j are $0 \leq i + j \leq n$ and $2i + j \geq n - k + 1$ and the number of combinations for given i and j are

$$\binom{n}{i, j} = \frac{n!}{i!j!(n-i-j)!}.$$

Therefore, the expression for $P_n(p, \lambda)$ is

$$P_n(p, \lambda) = \sum_{\substack{0 \leq i+j \leq n \\ 2i+j \geq n-k+1}} \binom{n}{i, j} p^i \lambda^j (1-p-\lambda)^{n-i-j} = \sum_{i=0}^n \sum_{j=\max(0, n-k+1-2i)}^{n-i} \binom{n}{i, j} p^i \lambda^j (1-p-\lambda)^{n-i-j}. \quad (2)$$

Using this expression, the probability of a decoder failure during the first decoding step can be computed.

The probability that a horizontal codeword fails when all vertical codewords fail is given as

$$G_\infty(p, \lambda) = P_M(p, \lambda), \quad (3)$$

where M out of N_1 symbols are transmitted in each row. Moreover, the probability of i horizontal failures among all of the horizontal codewords except the first is given by

$$F_i = \binom{K_2 - 1}{i} G_\infty^i (1 - G_\infty)^{K_2 - 1 - i}. \quad (4)$$

The second step of the decoding algorithm is evaluated by defining the probability $p_{vfail}^{(i)}$ that a vertical decoding fails when

- there are i horizontal erasures from the failure of the first decoding step and
- there are j random errors and k erasures in the last $N_2 - K_2$ symbols.

For the $N_2 - K_2$ check symbols, the probability of the occurrence of a random error is p and the probability of the occurrence of an erasure is λ . For the first K_2 symbols, the only errors to be considered are the failures of the horizontal decoding. For each horizontal failure, the error correction capability of the vertical codeword decreases by one unit. Starting from these considerations, (2) is modified. The number of random errors that can be corrected is $t(i) = \lfloor \frac{N_2 - K_2 - i}{2} \rfloor$, the number of correctable erasures without random errors in the check symbols is $N_2 - K_2 - i$, and the number of elements of the set of the trinomial distribution is $N_2 - K_2$. Therefore, the expression for $p_{vfail}^{(i)}$ is

$$p_{vfail}^{(i)} = \sum_{\substack{0 \leq j+k \leq N_2 - K_2 \\ 2j+k \geq N_2 - K_2 - i + 1}} \binom{N_2 - K_2}{j, k} \cdot p^j \lambda^k (1-p-\lambda)^{(N_2 - K_2 - j - k)}. \quad (5)$$

In the last decoding step, it can be noticed that, for the $N_1 - M$ symbols of the horizontal codeword, the erasure probability is $p_{vfail}^{(i)}$, while, for the remaining M symbols, we have a probability p of a random error occurrence and a probability λ of an erasure occurrence.

The number of random errors that can be corrected is $t(u) = \lfloor \frac{N_1 - K_1 - u}{2} \rfloor$, where u is the number of failures in the vertical decoding, $(N_1 - K_1 - u)$ is the number of correctable erasures without random errors in the M symbols, the number of elements of the set of the trinomial distribution is M . The first horizontal codeword fails the decoding step given i failures in the first decoding step with probability G_i , given by

$$G_i = \sum_{u=0}^{N_1 - M} \binom{N_1 - M}{u} (p_{vfail}^{(i)})^u (1 - p_{vfail}^{(i)})^{N_1 - M - u} \cdot \sum_{\substack{0 \leq v+w \leq M \\ 2v+w \geq N_1 - K_1 - u + 1}} \binom{M}{v, w} p^v \lambda^w (1-p-\lambda)^{(M-v-w)}. \quad (6)$$

Finally, the Decoder Failure Rate (DFR) is (see [1])

$$DFR = \sum_{i=0}^{K_2 - 1} F_i G_i \quad (7)$$

and, if $N_f = 1 + (K_2 - 1)G_\infty$ is the average number of horizontal word failures, the average Word Failure Rate (WFR) is

$$WFR = N_f \cdot DFR. \quad (8)$$

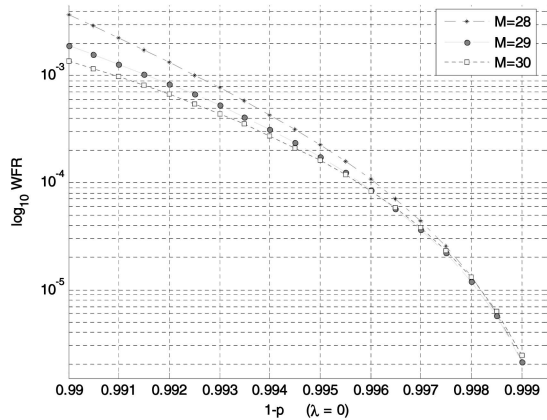


Fig. 2. Word failure rates with different M and $\lambda = 0$.

4 BER EVALUATIONS

In this section, the BER characterization of the PS codes is carried out by

- configuring the two concatenated RS codes by choosing $N_1 = 32, K_1 = 26, N_2 = 32,$ and $K_2 = 26,$
- assigning to M the following values: $M = 28, M = 29,$ and $M = 30,$ and
- varying the random error rate p and the erasures rate λ in the range $[10^{-5}, 10^{-2}]$.

The code rates of the obtained PS codes are $r = 0.89, r = 0.87,$ and $r = 0.85,$ respectively. Our attention is focused on M because it can be changed during the operational life of the codec without modifying its internal structure, while a modification of N and K implies structural modifications of the horizontal and vertical RS encoder and/or decoder. Different values of M are used, for example, to face a modification in the error and erasures rates p and λ due to a modification of the channel noise characteristics. Semilogarithmic plots of WFR versus $1 - p$ and $1 - \lambda$ for the three different values of M have been computed. Fig. 2 shows the three plots for $\lambda = 0$.

For high error rates, i.e., for $p > 4 \cdot 10^{-3}$, an increase in M , and, therefore, an increase in the number of transmitted check symbols, corresponds to a reduction in the word failure rate. In this case, the PS code shows a behavior similar to an RS code for which higher code rates correspond to higher WFR. For $p < 4 \cdot 10^{-3}$, an increase in M becomes counterproductive. This effect is due to the relationship between M and the error conditions outlined in

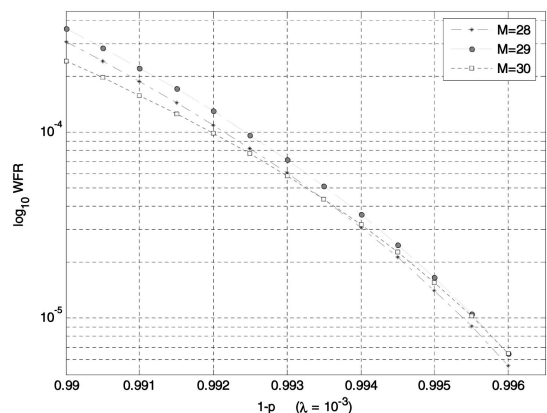


Fig. 3. Word failure rate with different M and $\lambda = 10^{-3}$.

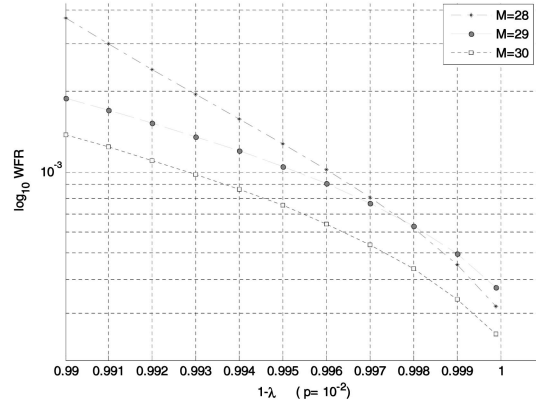


Fig. 4. Word failure rate versus $\lambda @ p = 10^{-2}$.

Section 2. For this range of error rates, a PS codeword with a higher code rate can be used without any penalty in the WFR.

In Fig. 3, the results of a similar experiment with $\lambda = 10^{-3}$ are shown.

In this case, $M = 29$ corresponds to the worst case, while the other M values give a better WFR. Moreover, $M = 28$ gives the best trade-off between the code rate r and the obtainable WFR. The behavior of the WFR with $p = 10^{-2}$ and $p = 10^{-4}$ and for $\lambda \in [10^{-5}, 10^{-3}]$ has been plotted in Fig. 4 and Fig. 5.

With higher error rates ($p = 10^{-2}$), the value of $M = 30$ becomes the best choice for the PS codeword, but, with low values of the erasures rate, $M = 28$ is better than $M = 29$. For $p = 10^{-4}$, an increase in M corresponds to a higher WFR independently by λ . Again, the behavior of the PS code is the opposite of the behavior of an RS code. The results plotted in these diagrams show that, in some conditions, with a smaller number of check symbols, we obtain a better WFR. Moreover, M can be easily adapted to face the modifications in erasures and random error rates.

5 ENCODER DESIGN

In this section, the design of a $PS(N1, K1, N2, K2, M)$ encoder is shown and an evaluation of its area with respect to the area of an RS encoder with the same code rate and codeword length is given. In particular, an $RS(255, 235)$ and a $PS(32, 28, 32, 26, 30)$ are compared. The two codes are characterized by $r = 0.91$. The PS encoder architecture is shown in Fig. 6 and it is implemented by using the following basic blocks:

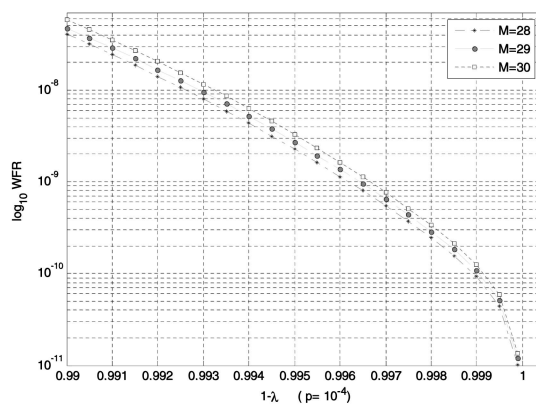


Fig. 5. Word failure rate versus $\lambda @ p = 10^{-4}$.

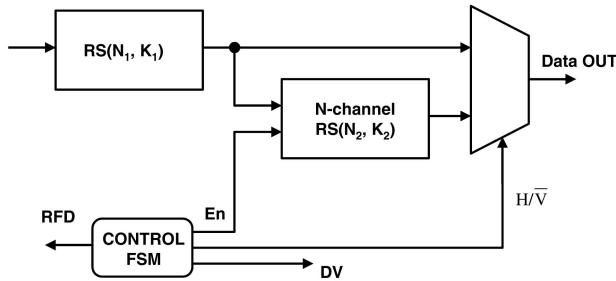


Fig. 6. Parity sharing RS encoder.

- an $RS(N_1, K_1)$ encoder,
- a multiple channel $RS(N_2, K_2)$ encoder,
- a Finite State Machine (FSM) giving the control signal used to synchronize the RS encoders, and
- a multiplexer used to select between the symbols provided by the first encoder and the symbols provided by the second one.

The $RS(N_1, K_1)$ encoder is implemented by using a Linear Feedback Shifter Register (LFSR) [6]. The first M symbols of the output codeword are directly given as output of the PS encoder, while the last $N_1 - M$ symbols are given as input to the second encoder. Because the last $N_1 - M$ symbols of the horizontal codeword form the data symbols for the vertical codeword, they can be processed in interleaved mode. Therefore, a multiple channel encoder with $N_1 - M$ channels is used to obtain the check symbols of the $RS(N_2, K_2)$ code. Fig. 7 shows an example of the multiple channel encoder.

This encoder provides an RS codeword with four check symbols. M is changed by selecting the length of the register chain using a multiplexer. The FSM in Fig. 6 provides the following control signals to synchronize the encoders and the multiplexer:

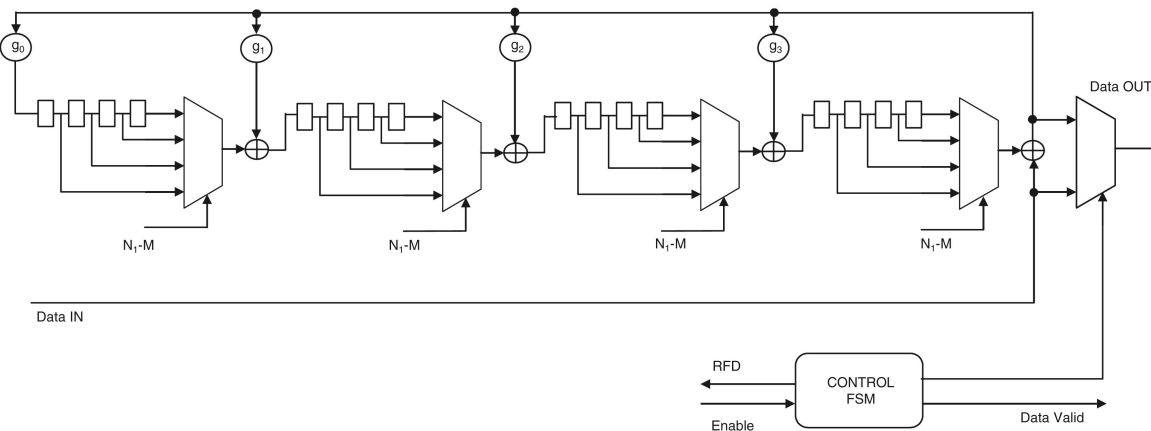


Fig. 7. RS multiple channel encoder.

- H/\bar{V} switches the multiplexer between the horizontal codeword provided by the first encoder and the vertical codeword provided by the second one,
- En (Enable) is used to assert to the second encoder which symbols must be sampled,
- DV (Data Valid) is used to identify the untransmitted symbol, and
- RFD (Ready For Data) signal advises when the overall PS encoder is able to process a new symbol.

A rough comparison of the area of the PS encoder with respect to the area of the RS encoder (with the same code rate) is obtained by counting the number of check symbols used in the two encoders. In fact, the area of the encoder is proportional to $n - k$ (see [9]). The $RS(255, 235)$ is characterized by $n - k = 20$, while, for the PS code, $(n - k) = (N_1 - K_1) + (N_2 - K_2) = 10$. Therefore, it is expected that the PS encoder area is nearly half of the area of the RS encoder.

Table 1 shows the area and timing performances of the $RS(225, 235)$ encoder, of the $RS(28, 32)$ encoder, of the $RS(26, 32)$ encoder, and of the $PS(32, 28, 32, 26, 30)$ encoder when the encoders are implemented on a XC2VP2-6 Xilinx FPGA.

6 DECODER DESIGN

The encoder architecture is not critical in terms of speed performances, but the design of a high speed RS decoder requires an in-depth analysis of its timing parameters, such as:

1. Processing delay, i.e., the number of clock cycles from the sampling of the first codeword symbol to the sampling of the first symbol of the subsequent codeword. If the processing delay is greater than n , the subsequent codeword must wait until the end of the decoding process. The processing delay depends on the number of check symbols and it is evaluated by using the following expression [8]:

TABLE 1
Comparison between the RS and PS Encoders

	RS(225, 235)	RS(28, 32)	RS(26, 32)	PS
Maximum Clock Frequency (in MHz)	200	210	206	206
Area (in Slices)	220	52	70	130

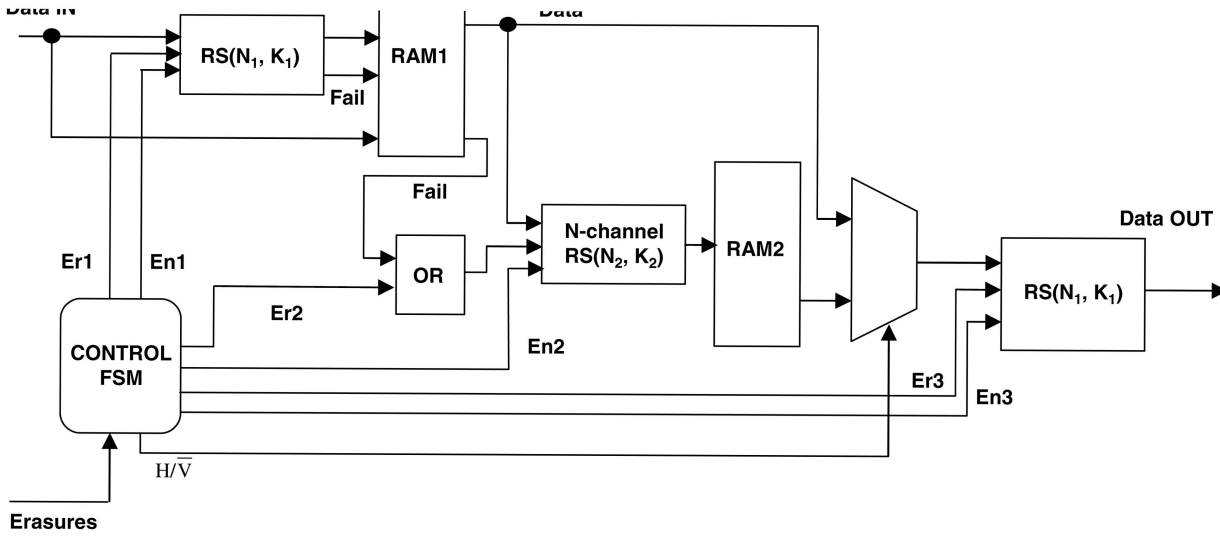


Fig. 8. Parity sharing RS decoder.

$$Processing\ Delay = 2 \left(\sum_{i=1}^{n-k-1} i \right) + 3(n-k) + 3 = (n-k)(n-k+2) + 3. \quad (9)$$

2. Latency, i.e., the number of clock cycles from the sampling of a symbol at the decoder input to the time at which the corrected version of that symbol is available in the decoder output. The latency depends on n and on the number of check symbols $n-k$. It can be evaluated by using the following equation [8]:

$$Latency = n + Processing\ Delay + 7. \quad (10)$$

3. Maximum Clock Frequency is the maximum frequency at which the decoder can be clocked
4. Symbol Throughput represents the amount of data that can be processed by the decoder in a time unit.

If the Processing Delay is less than the codeword length, the decoder is able to run in continuous mode and, therefore, it can provide a symbol each clock cycle at the output. Otherwise, a pause between two codewords must be inserted, deasserting the ready signal. In this case, the Symbol Throughput is

$$Symbol\ Throughput = \frac{Max\ Freq. \cdot n}{Processing\ Delay}. \quad (11)$$

The proposed PS decoder architecture is shown in Fig. 8 and it is based on the following blocks:

- two $RS(N_1, K_1)$ decoders,
- a multiple channel $RS(N_2, K_2)$ decoder,
- an FSM giving the control signals used to synchronize the RS decoders, and
- two RAMs used to store the partial results of the decoding algorithm.

The first $RS(N_1, K_1)$ decoder is used for the row codeword decoding. In this step, the erasure signal (Er1) is provided by the control FSM, which asserts that an erasure symbol has been detected by the block processing the “channel side information.” The corrected codewords are saved into RAM1. A fail flag indicates the codewords that have not been corrected during the

first decoding step. During this step, Er1 is asserted if an erasure is detected during the transmission of the first M symbols of the row codeword. Er1 is asserted for the last $N_1 - M$ clock cycles because these symbols are not transmitted in PS codes. The $RS(N_2, K_2)$ decoder is used for the column decoding. In this step, the input data are the check symbols from the output of the first decoder and the last $N_2 - K_2$ symbols of the column codeword. The erasure control signal is the OR of the Fail and Er2 signals. In particular, Fail asserts an erasure in one of the first K_2 symbols of the column codeword, while Er2 asserts an erasure in one of the last $N_2 - K_2$ symbols. In this way, the correctly decoded check symbols are processed as good symbols, while the uncorrectable check symbols are processed as erasures. Instead, for the last $N_2 - K_2$ symbols, Er2 is provided by the block processing the “channel side information.” After this step, the corrected check symbols are written in RAM2 and the third decoding phase starts. This phase uses the second $RS(N_1, K_1)$ decoder and the erasure signal Er3. The control FSM takes as input the erasure signal provided by the “channel side information” block and provides the enable signals (En1, En2, En3) to the RS decoders and the erasures signals (Er1, Er2, Er3) used by the RS decoders as described before.

The continuous streaming condition for the $RS(N_1, K_1)$ decoder is

$$(N_1 - K_1)(N_1 - K_1 + 2) + 3 \leq N_1. \quad (12)$$

This assumption is satisfied by limiting the number of check symbols of the row codeword.

For the $RS(N_2, K_2)$ decoder, the continuous streaming condition is satisfied if the $N_1 - M$ columns are decoded before a new parity sharing codeword is processed. The processing delay of the $RS(N_2, K_2)$ codeword is $(N_2 - K_2)(N_2 - K_2 + 2) + 3$; therefore, the continuous streaming condition can be expressed as

$$(N_1 - M)((N_2 - K_2)(N_2 - K_2 + 2) + 3) \leq K_2 M + (N_1 - M)(N_2 - K_2). \quad (13)$$

The latency of the RS code chosen as the example is very high. In fact, it grows quadratically with the number of check symbols and, for the given example, it becomes about 700 clock cycles (see (10)).

In the PS decoder, the obtained latency is lower; in fact, each step of the decoding algorithm starts only after L clock cycles, where L is the latency of the previous decoding step. In particular, for the chosen example, the latency of the first and the third

TABLE 2
Comparison between the RS and PS Decoders

	RS Decoder	PS Decoder
Latency	705	222
Maximum Clock Frequency (in MHz)	120	120
Symbol Throughput (in MHz)	69	120
Area (in Slices)	758	569

decoder is 66 clock cycles, while the latency of the second decoder is 90 clock cycles.

The overall latency is the sum of the latencies of each block and is about 222 clock cycles. We suppose that the maximum clock frequency of the RS is unchanged when the RS decoder is used as a building block of the PS decoder and, therefore, the Symbol Throughput is the same.

The area of the parity sharing decoder is evaluated starting from the same assumption used for the RS encoder, i.e., the area is linearly dependent from the number of used check symbols.

In this case, the PS decoder area is about 75 percent of an RS decoder with the same code rate. This result is relative to the implementation of the third step of the decoding algorithm by using a dedicated RS decoder. A reduction in area can be accomplished by sharing the $RS(N_1, K_1)$ decoder in the first and third steps of the decoding algorithm. The drawback of this choice is an increment of the processing delay.

Table 2 shows the area and timing performances of the $RS(255, 235)$ decoder and of the $PS(32, 28, 32, 26, 30)$ decoder when the decoders are implemented on a XC2VP2-6 Xilinx FPGA.

7 CONCLUSIONS

The error correction performance of block codes improves with the codeword length. For RS codes, the length is constrained by the size of the Galois Field over which the code is defined. On the other hand, the computational complexity and the memory required to carry out Galois Fields arithmetic increases with the field size and, therefore, decoding a long RS code can be impractical. Parity-sharing codes can increase the performances of hardware codecs while maintaining the same data rate and a comparable error correction capability as traditional RS codes. This paper analyzes the performance of PS codes in terms of obtainable BER with respect to the code parameters taking into account either random error and erasure rates as two independent probabilities. This approach provides an evaluation that is independent of the communication channel characteristics and extends the results to memory systems in which permanent faults and transient faults can be modeled, respectively, as erasures and random errors. In the paper, the hardware implementations of the PS encoder and decoder have been shown and their performances in terms of hardware complexity, speed, and throughput have been discussed.

REFERENCES

- [1] O. Collins, "Exploiting the Cannibalistic Traits of Reed-Solomon Codes," *IEEE Trans. Comm.*, vol. 43, no. 11, pp. 2696-2703, Nov. 1995.
- [2] M.K. Cheng and P.H. Siegel, "List-Decoding of Parity-Sharing Reed-Solomon Codes in Magnetic Recording Systems," *Proc. 2004 IEEE Int'l Conf. Comm.*, vol. 2, pp. 640-644, June 2004.
- [3] G. Cardarilli, F. Lombardi, M. Ottavi, S. Pontarelli, M. Re, and A. Salsano, "Comparative Evaluation of Designs for Reliable Memory Systems," *J. Electronic Testing: Theory and Applications*, Aug. 2005.
- [4] W.E. Ryan and P. Conoval, "A Method of Analysis for Interleaved Reed-Solomon Coding with Erasure Decoding on Burst Error Channels," *IEEE Trans. Comm.*, vol. 41, no. 3, pp. 430-434, Mar. 1993.
- [5] L.L. Joiner and J.J. Komo, "Errors and Erasures Decoding of BCH and Reed-Solomon Codes for Reduced M-Ary Orthogonal Signaling," *IEEE Trans. Comm.*, vol. 51, no. 1, pp. 57-62, Jan. 2003.
- [6] R.E. Blahut, *Theory and Practice of Error Control Codes*. Addison-Wesley, 1983.
- [7] M.B. Pursley, "Reed-Solomon Codes in Frequency-Hop Communications," *Reed-Solomon Codes and Their Applications*, S.B. Wicker and V.K. Bhargava, eds., pp. 150-174, IEEE Press, 1994.
- [8] Xilinx Logic Core Reed-Solomon Encoder v5.0, data sheet, 2004.
- [9] K. Sato, M. Hattori, N. Ohya, M. Sasano, and N. Shirota, "ARSDS: An Automated Reed-Solomon Decoder and Encoder Synthesis System," *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 611-614, May 1995.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.