RS Paper*

*Note: Sub-titles are not captured in Xplore and should not be used

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address

4th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address

5th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address

6th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address

***Abstract*—Pending.**

***Index Terms*—pending, pending, pending, pending**

I. Introduction

In modern communications and storage systems, it is observed the increasingly demand for solutions of high speed data rates and reliability with economic viable hardware. Error Correction Coding (ECC) techniques plays an important role to fulfill these opposite requirements. They aim to incorporate redundancy to the transmitted data and restrict the characteristics of the output signal during encoding in order to augment the capability to correctly extract the original signal at the decoder portion after receiving it from unreliable and noisy communication channels [1]. Reed-Solomon (RS) codes is one of the most popular ECC methods that uses a block-by-block basis to correct burst errors and erasures in data. The adequate compromise between effectiveness and implementation complexity guarantees the widespread use of RS codes in many applications until today.

The paper entitled ”Polynomial Codes over Certain Finite Fiels” [2] published in June 1960 was the first ~~appearance~~ of RS codes. At this time, a pursue to the most optimal and mechanizable ECC was apparent after Claude Shannon theory defines the upper bound of the channel capacity at which is possible to transmit error-free information using a proper coding method. Richard Hamming developed the first expressive practical work on ECC algorithms using linear algebra in early 1950s, and then Bose, Ray-Chaudhuri and Hocquenghem generalized his experiments in 1959 - being known as BCH codes. Finally, Irving Reed and Gustave Solomon co-invented a special case of BCH codes (RS codes) that employs an efficient decoding algorithm that enabled its wide range usage [3].

Since its invention, many technologies and standards have adopted RS codes with a variety of parametric configurations. The Compact Disc (CD) system was the first consumer mass application which employed RS codes [4], and since then many other used adopted it such as satellite and mobile systems, DVD and barcodes, and digital television. The literature presents many custom hardware implementations of RS Codecs even though they only differs in parametric terms. Current Hardware Description Languages (HDL) - Verilog and VHDL - already provide ways to express an parametric Register Transfer Level (RTL) architecture; however, it has not been explored in previous works. Also, verification of generic RTL blocks might be challenging and requires a proper methodology to assess implementation against block requirements. Most papers that explore RS codecs only provide a limited validation of the RTL implementation without analyzing metrics and results that certifies the block verification.

This work proposes a generic RTL implementation in VHDL and verification of a RS codec. The primary goal of this study is to present an open access IP of a RS codec that could be used as baseline for future works, and not to provide an optimized RTL architecture of it. The block will be verified by using Formal Verification (FV) approach, which is an emerging technology that has been becoming widely adopted in IP designs. Verifying all parametric combinations would not be feasible - or even impossible -, then this paper will use IEEE 802 as background. The IEEE 802 group, which develops standards and recommended practices for local, metropolitan, and other area networks, catalogs many commercial RS codec specifications with a diverse range of parametric setups that will be explored in this work.

The paper is organized as in the following: Section II introduces theoretical aspects of RS codes, section III analyzes related works, section IV presents the proposed RTL architecture of the RS codec, section V examines and exhibits results of the adopted verification approach, section VI explores

Identify applicable funding agency here. If none, delete this.

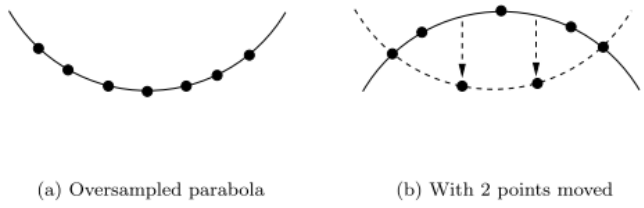
synthesis aspects of the implemented IP and the paper is concluded in section VII.

II. Reed-Solomon Codes: Theoretical Aspects

What are RS codes? This fundamental question can be the start point for discussing the theory behind it. The creators of RS codes already answered it using few lines in [5]:

> “They are simply sets of algebraic curves defined by polynomials with a limited range of degrees. These curves when graphed are a set of discrete points - the abscissas and ordinates are values in a Galois Field (GF). The degree limitation allows recovery of the complete curve even when the graph is assumed to be smudged and erased at many points. The relation of this idea to error control on noise-corrupted digital communication channels is immediate.”

Even without knowing GF, it is possible to have a glimpse of how RS codes work. It basically relies on ”fitting” points in a plane by the polynomial of smallest degree that passes through these points [6]. Basic polynomial mathematics affirms that 2 points can be fitted by a straight line, 3 points by a parabola, and so on. If some redundancy is added by oversampling a given polynomial function, the curve can be correctly guessed even though some recorded points move unintentionally. For instance, assuming that a parabola, which can be represented by $k = 3$ points, is recorded in $n = 7$ points (Fig. 1a). If $t = 2$ points are moved vertically, the original polynomial can still be recovered by finding a single parabola that fits as many as possible of the 7 points (Fig. 1b).


(a) Oversampled parabola

(b) With 2 points moved

Fig. 1. Example with $n = 7$ and $k = 3$.

It is noticeable that if t is greater than half of the number of extra points, an incorrect polynomial function may fit the greatest number of points. Therefore, $t = (n-k)/2$ represents the maximum capacity of error correction. RS codes can be seen in such way, but applied in the universe of field arithmetic. Therefore, RS codes are represented as a set of length n vectors (codewords) where their elements (symbols) consist of m binary digits (Fig. 2). The original message takes k vector positions, then there are $n - k$ redundant symbols (parity). A RS codec configuration is usually referred by the notation RS(n, k).

RS codes are linear - all codewords are sums of codewords - and cyclic - every cyclic shift of a code word is also a codeword [8]. These properties make the hardware implementation of the codec feasible, and this is granted only because of the application of GF during encoding and decoding processes. Its algebra establishes restricted rules in a finite field, thus every arithmetical operation will result in an element within a finite number of elements. GF is built using a primitive element α which is root of a primitive polynomial used as reference to generate the field elements. For digital world it is convenient to choose $\alpha = 2$ since it represents the binary notation. Hence, the total number of elements in a GF corresponds to 2^m. The Table 1? is a example of GF(2^3).

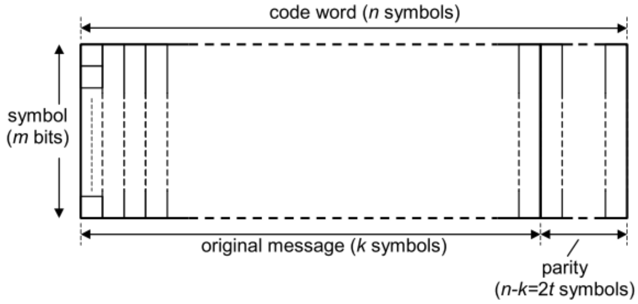
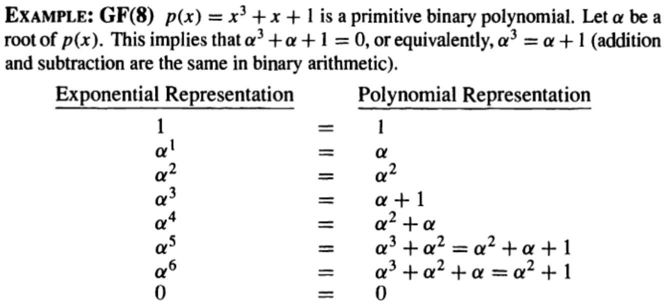


Fig. 2. Generic parametric configuration of RS codes.


Example: GF(8) $p(x) = x^3 + x + 1$ is a primitive binary polynomial. Let α be a root of $p(x)$. This implies that $\alpha^3 + \alpha + 1 = 0$, or equivalently, $\alpha^3 = \alpha + 1$ (addition and subtraction are the same in binary arithmetic).

Exponential Representation		Polynomial Representation
1	=	1
α^1	=	α
α^2	=	α^2
α^3	=	$\alpha + 1$
α^4	=	$\alpha^2 + \alpha$
α^5	=	$\alpha^3 + \alpha^2 = \alpha^2 + \alpha + 1$
α^6	=	$\alpha^3 + \alpha^2 + \alpha = \alpha^2 + 1$
0	=	0

Fig. 3. Generation of GF(2^3) - use this image as reference for creating the image for GF(2^3).

The approach to generate RS codes published in the original paper [2] takes a sequence of symbol information, $\{m_0, m_1, ..., m_{k-2}, m_{k-1}\}$, to build the polynomial $P(x) = m_0 + m_1 x + ... + m_{k-2}x^{k-2} + m_{k-1}x^{k-1}$. Every GF($\alpha^m$) element is evaluated in $P(x)$, and it results in a system of α^m linear equations in k variables (1). This system can be solved using any $\binom{\alpha^m}{k}$ combinations of equations. If the number of corrupted symbols $t \leq (\alpha^m - k)/2$, the most occurring solution of $\binom{\alpha^m}{k}$ linear systems corresponds the original message. However, this method clearly requires much computational resources.

$$
\begin{gathered}
P(0) = m_0 \\
P(\alpha) = m_0 + m_1\alpha + m_2\alpha^2 + ... + m_{k-1}\alpha^{k-1} \\
P(\alpha^2) = m_0 + m_1\alpha^2 + m_2\alpha^4 + ... + m_{k-1}\alpha^{2(k-1)} \\
... \\
P(\alpha^{m-1}) = m_0 + m_1\alpha^{m-1} + m_2\alpha^{2(m-1)} + ... + m_{k-1}\alpha^{(k-1)(q-1)}
\end{gathered} \tag{1}
$$

The most practical approach takes advantage of Generator Polynomials (GP) (2) for codification. The roots of GP are composed by consecutive elements of the adopted GF at extent of 0 to t. It can be demonstrated that cyclic codes can always be defined by a GP [3], which implies that the resulting codeword contains coefficients of a polynomial that is divisible by the GF. This assumption culminates in a more efficient hardware implementation for RS codec. The next section describes encoding and decoding processes of RS codes using GPs.

$$g(x) = (x + \alpha^0)(x + \alpha^1)...(x + \alpha^{2t-1}) \quad (2)$$

A. *RS Encoder*

- Principle
- Algorithm

B. *RS Decoder*

- General description of the process
- Diagram

1) Syndrome:

- Principle
- Algorithm

2) Euclidean:

- Principle
- Algorithm

3) Chien Search:

- Principle
- Algorithm

4) Forney:

- Principle
- Algorithm

C. *RS in IEEE 802*

- Introduction and details
- Table with the RS configurations
- Plot BER vs SNR for all configs

III. REED SOLOMON - RTL IMPLEMENTATION

A. *RS Encoder*

- Interface and explain how it works. Show waveform example for the most common operator conditions. Use https://wavedrom.com/ for it - no print screens from simulator waveform visualizer.
- Diagram with architecture and explanation
- State machine

B. *RS Decoder*

- Interface and explain how it works. Show waveform example for the most common operator conditions. Use https://wavedrom.com/ for it - no print screens from simulator waveform visualizer.
- Diagram with architecture and explanation

1) Syndrome:

- Architecture
- State Machine

2) Euclidean:

- Architecture
- State Machine

3) Chien Search:

- Architecture
- State Machine

4) Forney:

- Architecture
- State Machine

5) FIFOS:

- Calculation of FIFO size

IV. REED SOLOMON - VERIFICATION

- Explain the challenges of verifying a generic RTL. Introduce formal verification and explain why it fits this problem.
- Methodology. Use this reference: A Coverage-Driven Formal Methodology for Verification Sign-off

A. *RS Encoder*

- Testplan
- Formal test bench arch.
- Results

B. *RS Decoder*

- Testplan
- Formal test bench arch.
- Results

C. *Verification bug tracking*

Show plots about number of bugs found and time.

D. *Formal effort analysis*

The idea here is show how formal verification scales (or not) with more complex RS configurations

V. REED SOLOMON - SYNTHESIS

- Estimation of Area and number of gates. Here we can compare our IP with commercial IPs.
- Synthesis report
- Critical path and maximum clock

VI. REED SOLOMON - IMPLEMENTATION DETAILS

- show the vhdl file structure and tests
- Explain pyhton script for generating constants

VII. REED SOLOMON - CONCLUSION

...

ACKNOWLEDGMENT

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks . . .”. Instead, try “R. B. G. thanks. . .”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

References

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first . . ."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

References

[1] Geisel, William A. ”Tutorial on Reed-Solomon error correction coding.” (1990)

[2] Reed, Irving S., and Gustave Solomon. ”Polynomial codes over certain finite fields.” Journal of the society for industrial and applied mathematics 8.2 (1960): 300-304.

[3] Stephen B. Wicker; Vijay K. Bhargava., (1994) "An Introduction to Reed-Solomon Codes," Prentice-Hall

[4] KAS Immink, ”Reed-Solomon Codes and the Compact Disc" in SB Wicker and VK Bhargava, Eds., Reed-Solomon Codes and Their Applications, IEEE Press, 1994.

[5] Reed, Irving S., and Gustave Solomon. ”Reed-Solomon Codes: A Historical Overview" in SB Wicker and VK Bhargava, Eds., Reed-Solomon Codes and Their Applications, IEEE Press, 1994.

[6] Jack Keil Wolf, "An Introduction to Reed-Solomon Codes", www.ece.tamu.edu/ hpfister/courses/ecen604/rspoly.pdf.

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[8] Clarke, C. K. P. ”Reed-Solomon error correction.” BBC R&D White Paper, WHP 31 (2002).

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove the template text from your paper may result in your paper not being published.