# A Fast Algorithm for Encoding the (255,223) Reed-Solomon Code Over GF($2^8$)

R. L. Miller and T. K. Truong
Communications Systems Research Section

I. S. Reed
University of Southern California

*A new scheme for reducing the numerical complexity of the standard Reed-Solomon (R-S) encoding algorithm is developed. As an example, the encoding of a (255,223) R-S code (NASA standard for concatenation with convolutional codes) is shown to require 75 percent fewer multiplications and 61 percent fewer additions than the conventional method of computation.*

## I. Introduction

In this article, the fast syndrome method developed in Ref. 1 and Lagrange interpolation are used to develop a fast algorithm for encoding R-S codes. It is shown that both the number of multiplications and additions of this new scheme is substantially fewer than is required by the conventional encoding techniques.

An advantage of this new algorithm aside from speed is that its first step consists of performing a syndrome-like calculation, which can be implemented by using the existing syndrome algorithm used in the decoder. This fact can be used to lower the total hardware cost of the encoder-decoder system.

## II. Encoding Procedure

Let $n = 2^m - 1$ be the block length of an R-S code of designed distance $d$ in $GF(2^m)$. The number of $m$-bit message symbols is $k = n - d + 1$.

To encode the $k$ information symbols into an $n = 2^m - 1$ symbol R-S code word, one first defines the generator polynomial

$$G(x) = \prod_{i=1}^{d-1} (x - \alpha^i)$$

when $\alpha$ is a primitive $n$th root of unity. The code consists of all multiples of $G(x)$, subject to the constraint that $x^n = 1$.

Let $a_i$ for $d - 1 \leqslant i \leqslant n - 1$ be the message symbols, and define

$$I(x) = \sum_{i=d-1}^{n-1} a_i x^i$$

In order to generate the code word with information symbols corresponding to $I(x)$, proceed as follows: let

$$I(x) = Q(x)G(x) + R(x) \qquad (1)$$

where $Q(x)$ is a quotient polynomial, $G(x)$ is the generator polynomials, and $R(x)$ is the remainder upon dividing $I(x)$ by $G(x)$. Finally, $I(x)$ is encoded into

$$C(x) = I(x) - R(x) \qquad (2)$$

The new encoding procedure of an R-S code is composed of the following two steps:

(1) Compute $I(\alpha^i)$ for $1 \leqslant i \leqslant d - 1$ by the technique which is used to compute syndromes in the decoder. Note that by Eq. (2), $I(\alpha^i) = R(\alpha^i)$ for $1 \leqslant i \leqslant d - 1$.

(2) Compute $R(x)$ from $R(\alpha^i)$ using Lagrange interpolation

$$R(x) = \sum_{i=1}^{d-1} R(\alpha^i) E_i(x)$$

where $E_i(x)$ is defined by

$$E_i(x) = \frac{\prod\limits_{j \neq i} (x - \alpha^j)}{\prod\limits_{j \neq i} (\alpha^i - \alpha^j)} \text{ for } 1 \leqslant i \leqslant d - 1 \qquad (3)$$

The degree of $E_i(x)$ is $d - 2$; hence, the degree of $R(x)$ is at most $d - 2$. Thus, the parity symbols consist of the $d - 1$ coefficients of $R(x)$, as desired. Note that a direct computation of $R(x)$ in Eq. (1) involves $(d - 1)\cdot(n - d + 1)$ multiplications and $(d - 1)\cdot(n - d + 1)$ additions. If one uses a fast syndrome calculation, say, for the case $n = 255$, $k = 233$, it is shown in the following example that the encoder requires only 1812 multiplications and 2764 additions instead of the 7136 multiplications and 7136 additions required by a more conventional computation. This results in a significantly faster encoding scheme.

### Example

Let $n = 255$ be the block length of an R-S code of designed distance $d = 33$ over $GF(2^8)$. This code will correct any combination of 16 or fewer symbol errors. The first step of the encoding process is to compute $I(\alpha^i)$ for $1 \leqslant i \leqslant 32$. That is,

$$I(\alpha^j) = \sum_{i=0}^{255-1} a_i \alpha^{ij} \text{ for } 1 \leqslant j \leqslant 32 \qquad (4)$$

where $\alpha$ is an element of order 255 in $GF(2^8)$, and $a_i = 0$ for $0 \leqslant i \leqslant 31$. Note that Eq. (4) is the same formula as Eq. (1) in Ref. 1. Thus, using the same computing procedure, one obtains $I(\alpha^j)$ for $1 \leqslant j \leqslant 32$. It follows from Ref. 1 that the total number of multiplications and additions needed to compute the $I(\alpha^i)$ for $1 \leqslant j \leqslant 32$ is 852 and 1804, respectively. The second step of the encoding process is to compute $R(x)$ defined in Eq. 3, i.e.,

$$R(x) = \sum_{i=1}^{32} R(\alpha^i) E_i(x) \qquad (5)$$

Since $E_i(x)$ can be pre-computed, $32 \times 30 = 960$ multiplications and 960 additions are needed to compute $R(x)$. Hence, the total number of multiplications and additions required for encoding is $852 + 960 = 1812$ and $1804 + 960 = 2764$, respectively. In contrast, the total number of multiplications and additions for encoding by conventional methods is $223 \times 32 = 7136$ each.

## Acknowledgment

The authors wish to thank Prof. J. Massey of the Electrical Engineering Department, University of California, Los Angeles, for suggesting the possibility of a fast encoding algorithm of the type presented in this paper.

## Reference

1. Truong, T. K., Miller, R. L., and Reed, I. S., "A Fast Technique for Computing Syndromes of BCH and R-S Codes," *Electronics Letters*, Vol. 15, No. 22, pp. 720-721, October 25, 1979.