Architecture for a Smart Reed-Solomon Decoder

Emmanuel Boutillon, Arnaud Dehamel
Département COMELEC
Ecole Nationale Supérieure des Télécommunications (ENST)
46 rue Barrault, 75634 Paris cedex 13, France
emmanuel.boutillon@enst.fr

Abstract- **This paper describes a VLSI architecture for a Smart Reed-Solomon Decoder (SRSD). The SRSD use the RS code both as an forward error correction code and as an error control code. It uses information about the reliability of the received symbols to select "a priori" one (or more) efficient decodings that combine correction of errors and erasures. Once the decoding is processed, the SRSD also performs an "a posteriori" evaluation of the decoding process in order to reject low reliability decoded codewords.**

I. Introduction

The well known Reed-Solomon (RS) codes are usually used for forward error and/or erasure corrections [1-4]. Nevertheless, they can also be efficiently used as a simple check code for some other applications (wireless LAN for example), with an optional capability of error and/or erasure correction if the result of the correction is sufficiently reliable. In this paper, the principle of an RS decoder taking into account information about the reliability of the received symbols is presented. It performs one (or more) efficient decodings that combine correction of errors and erasures while maintaining an error control property. The overall VLSI architecture is described together with the performance results.

In section II, we present the principle of the adaptive decoding strategy. The decoding process is explained in section III and finally, the systolic Euclid architecture, modified to perform a combination of error and erasure corrections, is presented in section IV.

II. Adaptive Decoding Strategy

Let us consider an RS(n=2^m-1, k, d) Reed-Solomon code over GF(2^m), with message length n, number of information symbols k and minimum Hamming distance $d = n + 1 - k$. Let r be the number of redundant symbols, i.e. $r = n - k, = d - 1$.

Each of the r redundant symbols can be considered as a token during the decoding process. The correction of an erasure (a non-detected symbol) needs one token while the correction of a mistake needs two tokens (one for the position, one for the value). Thus, the correction of any set of a erasures and b errors with $a + 2b \le r$ can be made. The $c = r - (a + 2b)$ remaining symbols are used as control symbols to verify that the corrected word using $a + 2b$ tokens really belongs to the code. A formal demonstration can be found in [5].

The choice of (a, b, c) is based on the probability P_m of mis-correction (message accepted with errors after correction) and the distribution of the reliabilities of the received symbols. Once the decoding process is finished, an a posteriori evaluation of the corrected code-word is performed in order to reject codewords with non-consistent correction. A non consistent correction can be the correction of a symbol received with a high reliability, or, more generally, a correction where the "distance" between the received symbol and the corrected one is above a given threshold.

For example, let us consider an application with $P_m = 10^{-6}$ using an RS(7,3,5) code, i.e., $r = 4$. Let us assume that the symbols are received with 2 bits of reliability, as defined in figure 1.

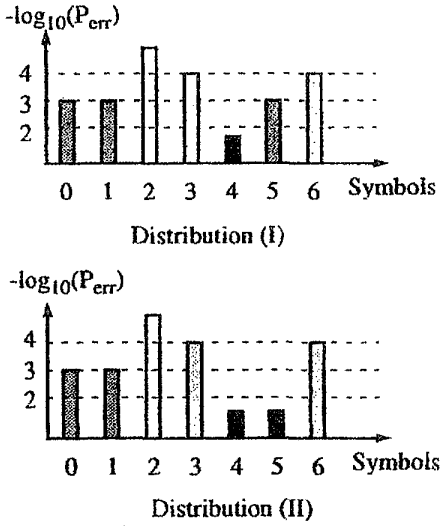

Fig. 1. Example of reliability distributions.

The distribution (I) of Fig. 1 leads to (a, b, c) = (1 ,1, 1) and K = {4} to correct the erasure (symbol 4, which has a probability of error greater than 10^{-2}) and one possible mistake for one of the 3 symbols with a probability of

0-7803-5491-5/99/$10.00 © 1999 IEEE

Authorized licensed use limited to: UNIVERSITE DE BRETAGNE SUD. Downloaded on December 12, 2008 at 13:18 from IEEE Xplore. Restrictions apply.

error of 10^{-3} (i.e. symbols 0, 1 and 5). If, for example, an error is found for symbol number 2, the correction is not coherent and the a posteriori evaluation process will reject the codeword.

Distribution (II) of Fig. 1 leads to $(a, b, c) = (2, 0, 2)$ and K={4,5}. Indeed, for the case of two errors among the symbols of reliability 10^{-3}, the correction of two erasures and one error ($(a, b, c) = (2, 1, 0)$) leads to a mis-correction.

III. Principle of Decoding

Let us present the key equations for decoding a Reed-Solomon code before describing the modified Euclid algorithm.

A. Key decoding equation

Let $K = \{k_i, i=1..a\}$ be the set of known erasure positions and $U = \{u_j, j=1..b\}$ the set of unknown error positions. The locator polynomial $\lambda[X]$ is defined by

$$\lambda[X] = \lambda_k[X] \cdot \lambda_u[X] \tag{1}$$

where

$$\lambda_k[X] = \prod_{1 \le i \le a} (1 + X \cdot \alpha^{k_i}) \tag{2}$$

is the erasure locator polynomial and

$$\lambda_u[X] = \prod_{1 \le i \le b} (1 + X \cdot \alpha^{u_i}) \tag{3}$$

is the error locator polynomial.

Let $S[X]$ be the received message syndrome ($\deg(S[X]) = r - 1$) and let $R[X]$ be the evaluator polynomial defined by:

$$R[X] = \sum_{i \in (K \cup U)} \frac{\lambda[X]}{(1 - X \cdot \alpha^i)} \tag{4}$$

Then, the key decoding equation is:

$$\begin{cases} \lambda[X] \cdot S[X] = R[X] \mod X^r \\ \deg(\lambda[X]) \le a + b \\ \deg(R[X]) < a + b \end{cases} \tag{5}$$

From the locator polynomial and the evaluator polynomial, the $a + b$ non zero values e_i of the error polynomial $E[X]$ are obtained from the locator and the evaluator polynomial:

$$(e_i \neq 0 \Leftrightarrow \lambda(\alpha^{-i}) = 0) \rightarrow \left(e_i = \alpha^i \cdot \frac{R[\alpha^{-i}]}{\lambda'[\alpha^{-i}]} \right) \tag{6}$$

A mathematical derivation of the above equations can be found in [5].

B. Decoding procedure

The purpose of the decoding process is to obtain the solution of the key equation (5). The algorithm is initialized by the two following equations:

$$\begin{cases} (E_a)_0 \quad \lambda a_0[X] \cdot S[X] = Ra_0[X] \mod X^r \\ (E_b)_0 \quad \lambda b_0[X] \cdot S[X] = Rb_0[X] \mod X^r \end{cases} \tag{7}$$

with

$$\begin{cases} \lambda a_0[X] = 1 & Ra_0[X] = S[X] \\ \lambda b_0[X] = 0 & Rb_0[X] = X^r \end{cases} \tag{8}$$

The first a steps of the decoding process are iterative multiplications of equation $(E_a)_i$, for $i = 1..a$:

$$(E_a)_i \leftarrow (1 + X \cdot \alpha^{k_i}) \cdot (E_a)_{i-1} \tag{9}$$

in order to obtain

$$\begin{cases} \lambda a_a[X] = \prod_{1 \le i \le a} (1 + X \cdot \alpha^{k_i}) = \lambda_k[X] \\ Ra_a[X] = \lambda_k[X] \cdot S[X] \mod X^r \end{cases} \tag{10}$$

Then, the next $2b$ steps are a classical Euclid's algorithm. Each iteration aims to decrease the degree of $Ra[X]$ (or $Rb[X]$) by one while increasing the degree of $\lambda a[X]$ (or $\lambda b[X]$) by one, and this by linearly combining the equations (E_a) and (E_b) as explained in [4].

After $2b$ iterations, if $\deg(Ra[X]) < a + b$, the decoding process is considered as successful. Error positions and error and erasure magnitudes are deduced from (6). Otherwise, the decoding fails and the message is not accepted.

Consider, as an example, the simple RS(7,3,5) Reed Solomon code defined over GF(8)=(Z/2Z)[X]/(X^3+X+1). The generator polynomial $G[X]$ is given by:

$$G[X] = \prod_{0 \le i \le 3} (1 + X \cdot \alpha^{-i}) \tag{11}$$

Authorized licensed use limited to: UNIVERSITE DE BRETAGNE SUD. Downloaded on December 12, 2008 at 13:18 from IEEE Xplore. Restrictions apply.

where α is the root of GF(8). Consider the transmission of an RS(7,3,5) codeword in which two errors occur, the first one of value α^2 on the coefficient of X^5 (position 5) and the second one of value α^1 on the coefficient of X^2 (position 2). The error polynomial is then:

$$E[X] = \alpha^1 \cdot X^2 + \alpha^2 \cdot X^5 \tag{12}$$

The syndrome $S[X]$ of the error polynomial $E[X]$ is then:

$$S[X] = \sum_{0 \le i \le 3} E[\alpha^i] \cdot X^i = (\alpha^4, \alpha^1, 0, \alpha^1) \tag{13}$$

where, by convention, the rightmost coefficient is the coefficient of the highest order (here X^3) and the leftmost coefficient the coefficient of X^0.

It is known, from the input reliabilities of this particular example,, that an erasure occurred in the fifth position , $(a, b, c) = (1, 1, 1)$ and $K = \{5\}$ is set for the decoding. Table 1 describes the different steps of the algorithm. The initial equations are $(E_a)_0$ and $(E_b)_0$. The first $a = 1$ step is the multiplication of equation $(E_a)_0$ with the partial locator polynomial $\lambda_t[X] = (1 + X.\alpha^5)$, according to eq. (9) Then, the Euclid's algorithm is performed; setting

$$\begin{cases} (E_a)_2 \leftarrow (E_a)_1 \\ (E_b)_2 \leftarrow X \cdot (E_a)_1 + \alpha^1 \cdot (E_b)_1 \end{cases} \tag{14}$$

which reduces the degree of $Rb[X]$ and setting

$$\begin{cases} (E_a)_3 \leftarrow \alpha^6 \cdot (E_a)_2 + \alpha^1 \cdot (E_b)_2 \\ (E_b)_3 \leftarrow (E_b)_2 \end{cases} \tag{15}$$

which reduces the degree of $Ra[X]$.

	X^0	X^1	X^2		X^0	X^1	X^2	X^3	X^4
$(E_a)_0$	α^0	0	0	$\times S[X] =$	α^4	α^1	0	α^1	0
$(E_b)_0$	0	0	0	$\times S[X] =$	0	0	0	0	α^0
$(E_a)_1$	α^0	α^5	0	$\times S[X] =$	α^4	α^4	α^6	α^1	0
$(E_b)_1$	0	0	0	$\times S[X] =$	0	0	0	0	α^0
$(E_a)_2$	α^0	α^5	0	$\times S[X] =$	α^4	α^4	α^6	α^1	0
$(E_b)_2$	0	α^0	α^5	$\times S[X] =$	0	α^6	α^4	α^4	0
$(E_a)_3$	α^6	α^2	α^6	$\times S[X] =$	α^3	α^2	0	0	0
$(E_b)_3$	0	α^0	α^5	$\times S[X] =$	0	α^6	α^4	α^4	0

Table 1: Example of mixed decoding

In $a+2b = 3$ steps, condition (5) is achieved since:

$$\begin{cases} \lambda a_3[X] \cdot S[X] = Rb_3[X] \mod X^r \\ \deg(\lambda a_3) = 2 \le a + b \\ \deg(Ra_3) = 1 < a + b \end{cases} \tag{16}$$

One can verify that, with the decoding equation (6), the error polynomial can be reconstructed.

IV. Architecture Implementation

We describe now the main characteristics of a hardware architecture and the modification of the Euclid's algorithm in order to include the erasure correction process.

A. *Global architecture*

The overall architecture is shown in Fig. 2.

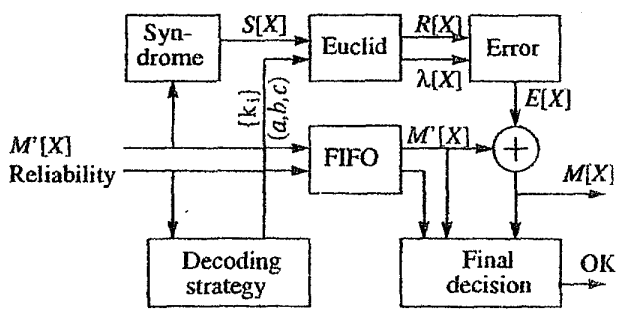

Fig. 2. Overall architecture

The received message $M'[X]$ is stored in a FIFO while the syndrome $S[X]$ is computed. At the same time, the distribution of the reliability of the symbols is evaluated by the decoding strategy block. Once $S[X]$ is computed and the decoding strategy selected, the modified Euclid algorithm is performed. The error polynomial $E[X]$ is thus built from $R[X]$ and $\lambda[X]$. Finally, the delayed received message and the error polynomial are added to find the corrected message. At this stage, the final decision block verifies that the result of the correction is consistent with the reliability of the symbols; otherwise, it rejects the received codeword. In cases where several decoding strategies are explored, "final decision" makes the final decision: rejection or choice of the best codeword.

Authorized licensed use limited to: UNIVERSITE DE BRETAGNE SUD. Downloaded on December 12, 2008 at 13:18 from IEEE Xplore. Restrictions apply.

B. Modified Euclid's algorithm

The hardware implementation of the modified Euclid's algorithm is based on the work of [6], with a pipeline structure. Polynomials are sent serially to a Processing Element (PE). The degree of the coefficient is implicitly given by its time of arrival (from highest to lowest coefficient). For example, multiplication of equation $(E_a)_0$ with $(1 + X \,.\, \alpha^5)$ is performed with the operator of Fig. 3,

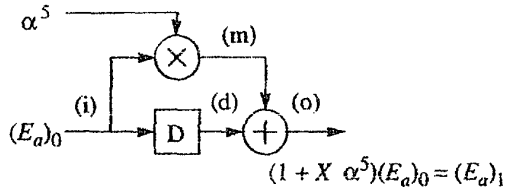

Fig. 3. Pipe-line multiplication

Table 2 shows the data going through paths (i), (m), (d) and (o) of figure 3. The input (i) is the concatenation of polynomial ($\lambda a_0[X]$, $Ra_0[X]$) (see Table 1), from the highest coefficient of $Ra_0[X]$ to the lowest coefficient of $\lambda a_0[X]$. Those two polynomials are multiplied by α^5 in (m), the output of the multiplier, and are delayed by 1 cycle in (d), the output of the register D. That means that data in (m) are multiplied by X relative to data in (d). Thus, data in the output (o) are the sum of the data going through (d) and (m), namely $\lambda a_1[X]$ and $Ra_1[X]$.

(i)	-	α^0	0	0	-	-	α^4	α^1	0	α^1	0
(m)	-	α^5	0	0	-	-	α^2	α^6	0	α^6	0
(d)	α^0	0	0	0	-	α^4	α^1	0	α^1	0	-
(o)	α^0	α^5	0	0	-	α^4	α^4	α^0	α^1	α^6	0

Table 2: Sequence of computation .

In this table, the light grey is used for the coefficient of X^0, while dark grey are for the coefficients of X^4 and above, i.e. dummy coefficients since operations are modulo X^4.

The PE of the Euclid algorithm described in [6] has been modified slightly to perform also iterations of type *a*. A design of this RS decoder using VHDL synthesis gives an additional hardware cost of 20%, including control processes and management of the polynomial degree.

V. Conclusion

In this paper, we have presented the basics of an adaptive Reed-Solomon decoder architecture. We have modified the classical decoder architecture to allow the correction of any set of *a* errors, *b* erasures, while keeping $c = r - (a + 2b)$ control symbols. The decoding strategy combined with the a posteriori evaluation of the decoding result gives significant improvement on the erasure&error correction and control check capabilities of the code. It allows to emulate a decoding process with a total amount of "virtual" redundant symbols r' greater than r.

The additional hardware cost for the decoding process (i.e., Euclid's algorithm) is 20%. The hardware cost of the "decoding strategy" and "final decision" depend on the type of reliability of the received symbols (from a simple scalar to a complete matrix of pairwise probabilities) and the requirement of the application.

This type of decoding can be very useful to improve the effective transmission rate of an ARQ protocol transmission (a wireless local area network for example), since part of the transmission errors are directly and reliability corrected.

References

[1] K. Sunghoon, S. Hyunchul, "An area-efficient VLSI architecture of a Reed-Solomon decoder/encoder for digital VCRs", IEEE Trans. on Consumer Electronics, vol. 43, n°4, Nov. 1997, pp. 1019-27.

[2] A.J. McAuley, "Reliable broadband communication using a burst erasure correcting code", Computer Communication Review, vol. 20, n°4, Sep. 1990, pp. 297-306.

[3] Moon-Ho-Lee, Seung-Bae-Choi, Jin-Su-Chang, "A high speed Reed-Solomon decoder", IEEE Trans. on Consumer Electronics, vol. 41, Nov. 1995, pp. 1142-9.

[4] O. Kyutaeg, S. Wonyong, "An efficient Reed-Solomon decoder VLSI with erasure correction", 1997 IEEE Workshop on Signal Processing Systems, SiPS'97, pp.193-201.

[5] R.J. McEliece, "The theory of information and coding: a mathematical framework for communication", Reading, Mass.: Addison-Wesley Pub. Co., Serie "Encyclopedia of mathematics and its applications", 1977

[6] M. Shao, T.K. Truong, "A VLSI design of a pipeline Reed-Solomon decoder", IEEE Trans. on Computers, vol c-34, n°5, May 1985, pp. 393-403.

Authorized licensed use limited to: UNIVERSITE DE BRETAGNE SUD. Downloaded on December 12, 2008 at 13:18 from IEEE Xplore. Restrictions apply.