High-speed Reed–Solomon decoder for correcting errors and erasures

C.-H. Wei, PhD
C.-C. Chen, MSc
G.-S. Liu, MSc

Indexing terms: Error-correction coding, Reed-Solomon decoder

Abstract: A Reed–Solomon decoder for errors-and-erasures correction, based on a new algebraic decoding algorithm, is presented. This high-speed decoder requires only n clock cycles for decoding each received n-symbol block. A serial structure that requires very few multipliers and provides a general expression to calculate the coefficients of the erasure-locator polynomial is also presented. A (15, 11) RS decoder and its shortened version (7, 3) RS decoder are used as design examples to illustrate the operating procedure of the new decoding algorithm.

1 Introduction

The encoder/decoder for an RS code differs from a binary encoder/decoder in that it operates on multiple bits instead of individual bits [1–4]. An (n, k) RS code is a block sequence of symbols in a Galois field $GF(2^m)$. This sequence of symbols can be considered as the coefficients of a code polynomial $C(X) = c_0 + c_1 + c_2X^2 + \cdots + c_{n-1}X^{n-1}$, where $c_i \in GF(2^m)$. The parameters of an (n, k) RS code are listed as follows:

m = number of bits per symbol
$n = 2^m - 1$ = block length of a codeword in symbols
k = number of information symbols in a codeword
t = maximum number of error symbols that can be corrected
$d = n - k + 1 = 2t + 1$ = minimum distance of the code.

For the t-error correcting RS code, the generator polynomial is given by

$$g(X) = (X - \alpha)(X - \alpha^2) \cdots (X - \alpha^{2t}) \quad (1)$$

where α is a primitive element in $G(2^m)$. Let $M_i(X)$ be the minimum polynomial of α^i, then $M_i(X) = X + \alpha^i$, where $i = 1, 2, \ldots, 2t$, and $g(X) = M_1(X)M_2(X) \cdots M_{2t}(X)$. Let $K(X) = c_{2t} + c_{2t+1}X + \cdots + c_{n-1}X^{k-1}$ be the information polynomial, then the encoded RS code polynomial in systematic form is represented by

$$C(X) = K(X)X^{n-k} + \text{mod}\,\{K(X)X^{n-k}/g(X)\}$$
$$= c_0 + c_1X + \cdots + c_{n-1}X^{n-1} \quad (2)$$

where mod $\{K(X)X^{n-k}/g(X)\}$ indicates the remainder polynomial of $K(X)X^{n-k}$ divided by $g(X)$.

An erasure is an error for which the error position is known but the magnitude is not. An RS code with minimum distance d is capable of decoding any pattern of w errors and s erasures as long as $2w + s < d$. Some RS decoding algorithms for errors-and-erasures correction have already been proposed [1–5], including a standard algebraic method [1–4], a transform method [6, 7], and Euclid's method [1, 7]. Recently, a modified step-by-step decoding algorithm [8, 9] for t-error-correcting cyclic codes was presented. In this paper, by combining the standard algebraic decoding method and the step-by-step decoding concept, a *new* algebraic decoding method is proposed to reduce the complexity of computation such that it can be easily implemented by VLSI technology.

2 Standard algebraic decoding algorithm

The standard algebraic decoding method [3–5] for errors-and-erasures correction of RS codes is described briefly as follows. Suppose that a code vector $C(X)$ is transmitted, errors and erasures occur such that a received vector $r(X) = C(X) + e(X) = r_0 + r_1X + \cdots + r_{n-1}X^{n-1}$ is obtained. The error pattern $e(X)$ can be described by a list of values and locations of its nonzero components. The location will be given in terms of an error-location number which is simply α^j for the $(n - j)$th symbol. Thus, each nonzero component of $e(X)$ is described by a pair of field elements Y_i (the error value) and X_i (the error-location number), where X_i and Y_i are both elements of $GF(2^m)$. If X_i is known and Y_i is unknown, then (U_i, V_i) is used to replace the pair (X_i, Y_i), where U_i and V_i denote the erasure locator and erasure magnitude, respectively.

Suppose that $w \leqslant t$ errors occur in positions $X_1, X_2, \ldots, X_w$, with nonzero magnitudes $Y_1, Y_2, \ldots, Y_w$, respectively. Also, s erasures occur in positions $U_1, U_2, \ldots, U_s$ with respective magnitudes $V_1, V_2, \ldots, V_s$. Furthermore,

Paper 9458I (E5, E7), first received 20th February 1992 and in revised form 11th January 1993

The authors are with the Institute of Electronics and Center for Telecommunications Research, National Chiao Tung University, Hsin Chu, Taiwan, Republic of China

This work was supported by the National Science Council of Republic of China, under grant NSC82-0404-E009-122.

assume that $2w + s \leqslant d$. The first step in the decoding process is to calculate the syndrome values. Since $C(X)$ is a code vector and has these elements as roots, the syndromes S_q are given by

$$S_q = r(\alpha^q) = C(\alpha^q) + e(\alpha^q) = e(\alpha^q) = \sum_{i=1}^{w} Y_i X_i^q + \sum_{i=1}^{s} V_i U_i^q \quad (3)$$

for $q = 1, 2, \ldots, 2t$.

As the syndrome values are given, the remaining considerations are to find the erro-locators and the corresponding error values, and also to find the erasure values providing that the erasure-locators are known already. Define the erasure-locator polynomial and the error-locator polynomial, respectively, as follows:

$$\sigma_d(x) = \prod_{i=1}^{s}(x - U_i) = \sum_{p=0}^{s} \sigma_{dp} X^{s-p} \quad (4)$$

$$\sigma_p(x) = \prod_{i=1}^{w}(x - x_i) = \sum_{p=0}^{w} \sigma_{ep} x^{w-p} \quad (5)$$

Also, define the modified syndromes as

$$T_{j-s-1} = \sum_{p=0}^{s} \sigma_{dp} S_{j-p} \quad (6)$$

for $j = s + 1, s + 2, \ldots, 2t$. After some mathematical manipulations, the following Newton identities can be obtained [4]:

$$T_{j+w-s-1} + \sigma_{e1} T_{j+w-s-2} + \cdots + \sigma_{e(w-1)} T_{j-s} + \sigma_{ew} T_{j-s-1} = 0 \quad (7)$$

where $j = s + 1, \ldots, 2t$.

As soon as the σ_{ep}'s are obtained, by solving eqn. 7, the error-locators can be found using the Chien search. Let the given erasure-locators and the calculated error-locators be denoted together by $U_1, U_2, \ldots, U_{w-1}, U_w, U_{w+1}, \ldots, U_v$, where $v = w + s$. The generalised erasure-locator polynomial, with U_p being deleted, is defined by

$$\sigma_{dp}(x) = \prod_{i \neq p}^{v}(x - U_i) = \sum_{q=0}^{v-i} \sigma_{dpq} x^{v-1-q} \quad (8)$$

The erasure value corresponding to locator U_p is then given by [4]

$$V_p = \frac{\sum_{q=0}^{v-1} \sigma_{dpq} S_{v-q}}{\sum_{q=0}^{v-1} \sigma_{dpq} U_p^{v-q}} \quad (9)$$

for $p = 1, 2, \ldots, v$. By deleting one erasure at a time, all erasure values can be calculated in turn by eqn. 9.

3 New algebraic decoding algorithm of RS codes

A new algebraic decoding algorithm presented in the following uses the cyclic property of RS codes to simplify the calculation of $\sum_{q=0}^{v-1} \sigma_{dpq} U_p^{v-q}$ in eqn. 9. When one of the erasure locators U_p is equal to unity, $\sum_{q=0}^{v-1} \sigma_{dpq} U_p^{v-q} = \sum_{q=0}^{v-1} \sigma_{dpq}$.

To make $U_p = 1$, the entire received word is cyclically shifted step-by-step. Then we check the lowest-order symbol of the cyclically shifted word whether it is an erasure symbol or not. If there is an erasure symbol at the lowest-order position of the cyclically shifted word, then the corresponding erasure value is calculated using the following formula:

$$V_p = \frac{\sum_{q=0}^{v-1} \sigma_{dpq} S_{v-q}}{\sum_{q=0}^{v-1} \sigma_{dpq}} \quad (10)$$

By the same procedures, all other erasure values can be solved step by step. The new algebraic decoding method is a combination of the standard algebraic decoding method and the step-by-step method.

In the following description a decoder for RS codes of distance 5 is used to illustrate the new algebraic decoding algorithm. This RS decoder has the capability of correcting two errors, or one error and two erasures, or four erasures. In our discussion, only cases for which the received word has, at most, one error and some erasures are considered.

3.1 Decoding algorithm for RS codes of distance 5

If the decoder restricts itself to correcting at most one error, the (n, k) RS code of distance 5 can work in the following ten cases:

(1) no error and no erasure
(2) no error, and one erasure, say (U_1, V_1)
(3) no error, and two erasures, say $(U_1, V_1), (U_2, V_2)$
(4) no error, and three erasures, say $(U_1, V_1), (U_2, V_2), (U_3, V_3)$
(5) no error, and four erasures, say $(U_1, V_1), (U_2, V_2), (U_3, V_3), (U_4, V_4)$
(6) no erasure, and one error, say (X_1, Y_1)
(7) one error, and one erasure, say $(X_1, Y_1), (U_1, V_1)$
(8) one error, and two erasures, say $(X_1, Y_1), (U_1, V_1), (U_2, V_2)$
(9) two errors
(10) more than four erasures

First define the erasure-number indicators F_i as follows:

If one erasure occurs, then the erasure indicator $F_1 = 1$; else $F_1 = 0$
If two erasures occur, then the erasure indicator $F_2 = 1$; else $F_2 = 0$
If three erasures occur, then the erasure indicator $F_3 = 1$; else $F_3 = 0$
If four erasures occur, then the erasure indicator $F_4 = 1$; else $F_4 = 0$
If erasure number is equal or less than two, then ERR = 1; else ERR = 0

Then define the coefficients of erasure-locator polynomial by

$$\left.\begin{aligned} C_{d0} &= F_2; \quad C_{d1} = (U_1 + U_2)F_2 + F_1; \\ C_{d2} &= \alpha^0 \quad \text{if } F_1 = 0 \text{ and } F_2 = 0; \\ &= U_1 \quad \text{if } F_1 = 1 \text{ and } F_2 = 0; \\ &= U_1 U_2 \quad \text{if } F_2 = 1. \end{aligned}\right\} \quad (11)$$

The modified syndromes are defined by

$$\left.\begin{aligned} T_0 &= C_{d0} S_3 + C_{d1} S_2 + C_{d2} S_1 \\ T_1 &= C_{d0} S_4 + C_{d1} S_3 + C_{d2} S_2 \end{aligned}\right\} \quad (12)$$

and the erasure-locator polynomial coefficients with U_p being deleted are

$$\left.\begin{aligned} C_{dp0} &= F_4 \\ C_{dp1} &= (U_1 + U_2 + U_3 + U_4 - 1)F_4 + F_3 \\ C_{dp2} &= (U_1U_2U_3 + U_1U_2U_4 + U_1U_3U_4 \\ &\quad + U_2U_3U_4 - U_1U_2U_3U_4)F_4 \\ &\quad + (U_1 + U_2 + U_3 + U_4 - 1)F_3 + F_2 \\ C_{dp3} &= U_1(W_1)U_2(W_2)U_3(W_3)U_4(W_4) \end{aligned}\right\} \quad (13)$$

where $U_p(W_p) = U_p$ if $U_p \neq 0$, or $U_p(W_p) = 1$ if $U_p = 0$, for $p = 1, 2, 3, 4$. Also $C_{dpq}^{(j)}$, $S_i^{(j)}$, and $U_i^{(j)}$ denote the values of the C_{dpq}, S_i and U_i of the shifted vector $r^{(j)}(x)$, respectively.

The procedure of the new decoding algorithm for (n, k) RS codes of distance 5 is listed below.

(i) Calculate the erasure number, syndromes, erasure locations, and detect error numbers:

(a) Get the four syndromes S_1, S_2, S_3, S_4

(b) Get the four erasure locators U_1, U_2, U_3, U_4 and the erasure indicators F_1, F_2, F_3, F_4, and ERR

(c) If one of the following two conditions occurs:

(1) the number of erasures is larger than four

(2) if two errors occur when there is no erasure then the error-flag is reset and do nothing to the received word.

(ii) Calculate the only-one error locator

If the number of erasures is not larger than two (ERR = 1) then calculate $U_3 = (T_1/T_0)$, and modify the erasure indicators as follows:

If $U_3 \neq 0$ and $F_2 = 1$, then $F_2 = 0$ and $F_3 = 1$

If $U_3 \neq 0$ and $F_1 = 1$, then $F_1 = 0$ and $F_2 = 1$

If $U_3 = 0$, then do nothing to F_1, F_2, F_3

(iii) Let $j = 1$

(iv) Calculate the erasure-locator polynomial with U_p being deleted:

(1) Get $U_i^{(j)} (i = 1, 2, 3, 4)$

(2) Get $C_{dp0}^{(j)}, C_{dp1}^{(j)}, C_{dp2}^{(j)}, C_{dp3}^{(j)}$ from $U_i^{(j)}$

(v) Calculate the erasure values:

(1) Get $S_i^{(j)}$ $(i = 1, 2, 3, 4)$

(2) Calculate $V_p^{(j)}$ with

$$V_p^{(j)} = \frac{C_{dp0}^{(j)} S_3^{(j)} + C_{dp1}^{(j)} S_2^{(j)} + C_{dp2}^{(j)} S_1^{(j)}}{C_{dp0}^{(j)} + C_{dp1}^{(j)} + C_{dp2}^{(j)}}$$

(3) If an erasure flag is found in the symbol shifted out and no ERROR-FLAG is found from step (i), then $V_p = V_p^{(j)}$; otherwise $V_p = 0$

(4) Add $V_p^{(j)}$ to the symbol shifted out

(vi) If $j < n$, let $j = j + 1$ and go back to (iv)

If the decoding circuit is used for decoding the $(n - f, k - f)$ RS code, a shortened version of (n, k) RS code, then the proper syndromes for decoding the received symbol r_{n-1-f} is equal to the remainder resulting from dividing $X^f r(X)$ by $M_i(X)$ where $M_i(X) = (X - \alpha^i)$ for $i = 1, 2, \ldots, 2t$. This computation can be accomplished with a one-stage LFSR, while the input is multiplied by a constant field element before entering the LFSR. Computing all $S_i^{(f)}$ in this way, the extra f shifts of the syndrome register can be avoided, and the decoding algorithm discussed in the previous Section is directly applicable to the shortened RS codes.

3.2 Serial structure for calculating the coefficients of erasure-locator polynomial

The computation to find the coefficients σ_{dpq} of the erasure-locator polynomial with U_p being deleted is very complicated. To simplify the hardware complexity, a serial structure for calculating these coefficients is proposed here.

The erasure-locator polynomial defined by eqn. 4 is rewritten as

$$\sigma_d(X) = \prod_{q=1}^{s} (X - U_q) = \sigma_{ds} + \sigma_{d(s-1)}X + \cdots + \sigma_{d1}X_{s-1} + \sigma_{d0}X_s$$

In fact, $\sigma_d(X)$ can be calculated serially. This is because multiplying a polynomial, say $A(X)$, by $(X - U_q)$ is equivalent to multiplying $A(X)$ by X and subtracting the polynomial obtained from multiplying each coefficients of $A(X)$ by U_q. From this point of view, $\sigma_d(X)$ can be obtained by the following procedures:

(i) Set $\sigma_d(X) = 1$ and $A(X) = 1$

(ii) Let $q = 1$

(iii) If $q < s$, then perform

(*a*) multiply $\sigma_d(X)$ by X

(*b*) multiply $A(X)$ by U_q

(*c*) subtract $A(X)$ from $\sigma_d(X)$

(*d*) set $A(X) = \sigma_d(X)$.

(iv) If $q < s$, then increment q by 1 and go to step 3, else stop

The erasure-locator polynomial with U_p being deleted, defined by eqn. 8, can be rewritten as

$$\sigma_{dp}(x) = \prod_{q \neq p} (X - U_q) = \sigma_{dp(s-1)} + \sigma_{dp(s-2)}X + \cdots + \sigma_{dp1}X^{s-2} + \sigma_{dp0}X^{s-1}$$

Then $\sigma_{dp}(X)$ can be obtained by the same algorithm with slight modification as described below.

(i) Set $\sigma_{dp}(X) = 1$ and $A(X) = 1$

(ii) Let $q = 1$

(iii) If $q = p$, then go to step 4, else if $q < s$, then perform

(*a*) multiply $\sigma_{dp}(X)$ by X

(*b*) multiply $A(X)$ by U_q

(*c*) subtract $A(X)$ from $\sigma_{dp}(X)$

(*d*) set $A(X) = \sigma_{dp}(X)$.

(iv) If $q < s$, then let $q = q + 1$ and go to step 3, else stop

4 Realisation of RS decoders

Fig. 1 shows the functional block diagram of the RS decoder for errors- and-erasures correction. The decoder is constructured by three basic modules: buffer module, syndrome and erasure-locator calculation module, and errors-and-erasures correcting module. The function of each module for the (15, 11) RS decoder will be described in the following. The symbols of the (15, 11) RS code are in $GF(2^4)$. For a received word $r(x) = r_0 + r_1X + \cdots + r_{14}X^{14}$, each of r_i can be denoted by four binary digits for code symbol plus one binary digit for erasure flag. If the code symbol is an erasure, then set the corresponding erasure flag to 1, otherwise reset the erasure flag to 0. The syndrome values can then be found from

$$S_i = r(\alpha^i) = \text{mod } \{r(X)/(X + \alpha^i)\} \quad \text{for } i = 1, 2, 3, 4$$

where mod $\{r(X)/(x + \alpha^i)\}$ indicates the remainder of $r(X)$ divided by $X + \alpha^i$. Furthermore, the syndrome values cyclically shifted j places to the right are defined as

$$S_i^j = \text{mod}\ \{X^j r(X)/(X + \alpha^i)\} \quad \text{for } i = 1, 2, 3, 4.$$

In practice, the syndrome values can be calculated by using four one-stage linear feedback shift registers (LFSRs) in hardware, and the shifted syndrome values can be obtained by shifting the contents of LFSRs with inputs blocked.

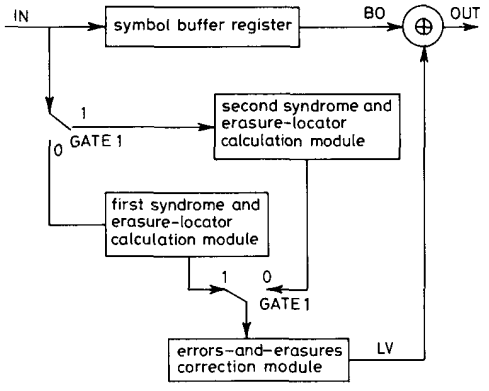

Fig. 1 *Block diagram of RS decoder for error-and-erasure decoding*

4.1 Buffer module

The buffer module is composed of 16 one-symbol shift registers in $GF(2^4)$. Each one-symbol shift register is constructed by four binary shift registers linked in parallel. The first 15 symbol registers are used to store the received word, while the last one is used to latch the symbol which is shifted out and is ready for decoding.

4.2 Syndrome and erasure-locator calculation modules

Fig. 2 shows the diagram of a syndrome calculation module and an erasure–locator calculation module. The erasure-locator calculation module calculates the erasure locators from the erasure-flag of the word.

The new algebraic decoding algorithm requires $2n$ shift operations for decoding one received word. The first n shifts are used to calculate the initial syndrome values and erasure-locators, the remaining n shifts are used to decode the errors and erasures in the received word. Since the number of shifts required for decoding errors and erasures is equal to that for calculating the initial syndrome values and erasure-locators, the calculation of syndrome and erasure-locator of the next received word can be performed at the same time. Thus, the average number of shifts required for decoding one received n-symbol block is just n. This line-speed decoding capability can be achieved by using two syndrome and erasure-locator calculation modules in the decoder structure as shown in Fig. 1. The two modules interchange each other to calculate the syndrome values and erasure-locators of the received words. When a block is being shifted into the buffer module, the first syndrome and erasure-locator calculation module is working to calculate the syndrome values and erasure-locators of the current received block, while the second one holds the calculated syndrome values and the erasure–locators of the previous block which are ready for the errors-and-erasures correction modules and others. As soon as the current block is completely received, the calculated result from the first syndrome and erasure-locator calculation module is passed to the errors-and-erasures correction modules for decoding that block, and the second syndrome and erasure-locator calculation module is enabled to calculate the syndrome values and erasure-locators of next received block.

4.3 Errors-and-erasures correction module

The errors-and-erasures correction module is used to calculate the error-locator and the erasure values from the calculated result of the syndrome and erasure-locator calculation module. The block diagram of the error-and-erasures correction module is illustrated in Fig. 3, which comprises an erasure-locator polynomial calculation

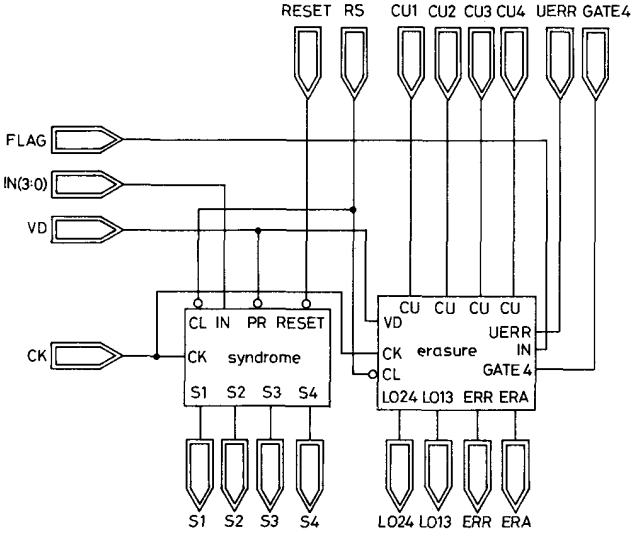

Fig. 2 *Syndrome-and-erasure locator calculation module*

circuit, a modified syndrome calculation circuit and an error-locator and erasure value calculation circuit.

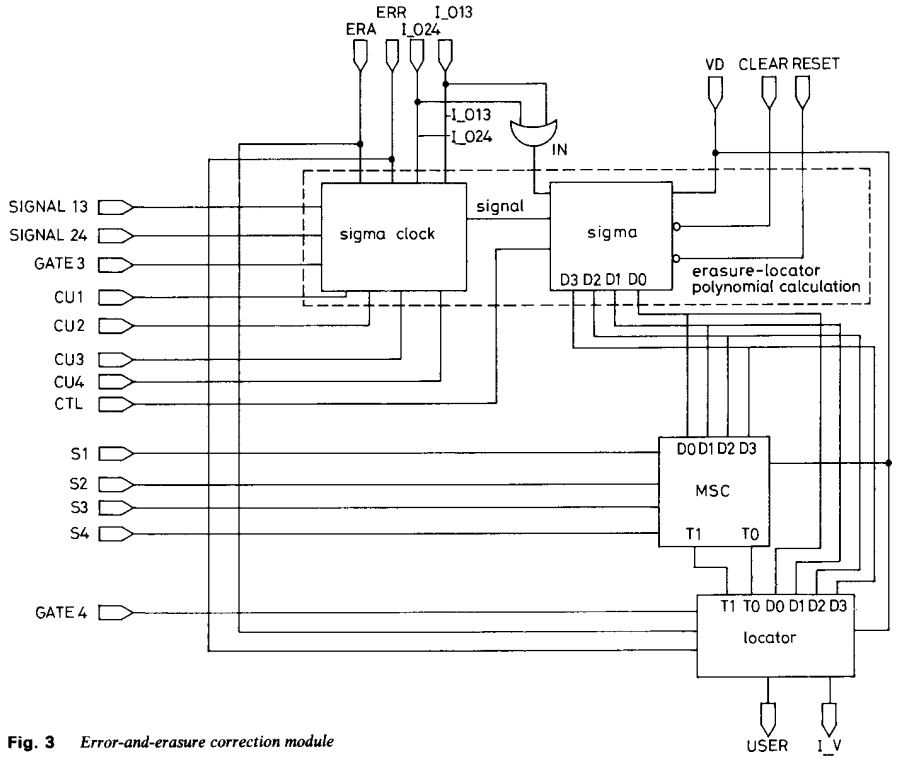

Fig. 3 *Error-and-erasure correction module*

The erasure-locator polynomial calculation circuit comprising a sigma circuit and a sigma clock is used to calculate the coefficients of the erasure-locator polynomial using the algorithm derived. The sigma circuit first set $\sigma_{dq}(x) = 1$, then multiply $\sigma_{dq}(x)$ by $(X - U_q)$, $q = 1, 2, 3, 4$. This circuit can be realised by three multipliers, four adders, and four pieces of one-stage LFSRs. The modified syndromes T_1, T_0, and T_p can be obtained from the syndrome values and the coefficients of the erasure-locator polynomial by the following equations:

$$T_1 = \sigma_{d1} S_4 + \sigma_{d2} S_3 + \sigma_{d3} S_2$$

$$T_0 = \sigma_{d0} S_4 + \sigma_{d1} S_3 + \sigma_{d2} S_2 + \sigma_{d3} S_1 = T_p$$

Therefore, the modified syndrome calculation circuit can be implemented by seven multipliers and five adders. As soon as the coefficients of the erasure-locator polynomial and modified syndrome values are known, error-locator ($UERR$) or the erasure values (LV) can be obtained by

$$UERR = T_1/T_0$$

and

$$\mathrm{LV} = T_0/(\sigma_{d0} + \sigma_{d1} + \sigma_{d2} + \sigma_{d3})$$

4.4 Finite field multiplier, inverse and adder

The decoding speed of the decoder is dominantly determined by the delay (computation time) of the multiplier. The multiplication of two elements in $GF(2^4)$ can be achieved by using a $m = 4$ cellular-array multiplier [10]. Another type of Massey-Omura multipliers [11] taking only m units of gate delay to complete the multiplication of two elements in $GF(2^m)$ can be employed to increase the decoding speed, although some extra basis invert operations should be added for transforming the elements represented by a conventional basis $\{1, \alpha, \alpha, \alpha^2, \ldots, \alpha^{m-1}\}$ into the ones represented by a normal basis $\{\alpha, \alpha^2, \alpha^4, \ldots, \alpha^{2m-1}\}$. For any α in the finite field $GF(2^m)$, $\alpha^{2^m} = \alpha$. Hence, the inverse of α is $\alpha^{-1} = \alpha^{2^m-2}$. The inverse in $GF(2^4)$ can be accomplished by using combinational logic circuits, and the addition in $GF(2^4)$ can be accomplished by using 2-input Exclusive-OR (XOR) gates.

4.5 Control module

Fig. 4 shows the required control signals for the (15, 11) RS decoder for error and erasure correcting. This module is implemented in Fig. 5. The function of each control signal is described in the following list.

(1) CK is the data rate of the received word.

(2) CKX8 is an internal clock of the decoder, and its frequency must be at least 8 times faster than that of CK.

(3) VD is the DC voltage source.

(4) RESET is the initial reset of the decoder.

(5) RS1 and RS2 are used to reset the syndrome and erasure-locator calculation modules after the results are computed and passed out for every $2n$ clock cycles which is the decoding time of a received block.

(6) IN is the serially received data input.

(7) FLAG is the serial received erasure-flag.

(8) ERR is an indicator to indicate whether the decoder has error correction capability or not. If the received word contains no more than two erasures, then ERR = 1, which indicates that the decoder has one error

correction capability; else ERR = 0, which indicates that the decoder can not correct any error.

(9) ERA is an indicator to indicate that there is an erasure whose locator is 1.

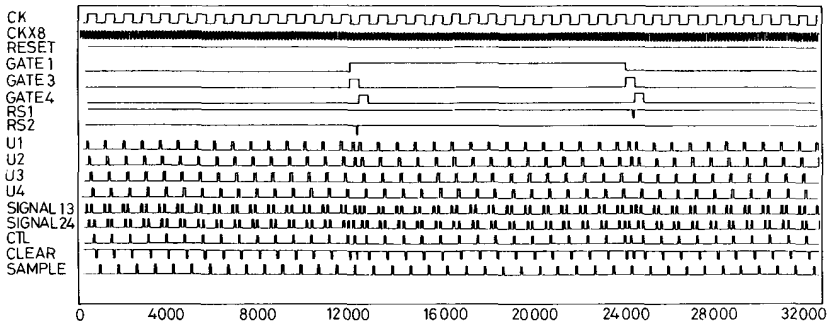

Fig. 4 *Timing diagram of control signals*

(10) GATE1 is the signal used to control some switch circuits of the decoder. GATE1 will be low and high alternately for every n cycles. When GATE1 is low, the current input symbols from IN will be switched to the first syndrome- and erasure-locator calculation module to calculate its initial syndrome values and erasure-locators. Then the calculated syndrome values and erasure-locators of the previous received block in the second syndrome and erasure-locator calculation module will be passed to the errors-and-erasures correction modules for further processing. After n clock cycles, GATE1 will be high and the operations of the first and second syndrome and erasure-locator calculation modules alternate with each other.

(11) GATE3 will be high for one-half clock cycle after GATE1 has changed. This signal is used to select the erasure-locator polynomial calculation circuit to calculate the coefficients of the erasure-locator polynomial.

(12) GATE4 will be high for one-half clock cycle after GATE1 has changed one-half clock cycle. This signal is used to select the error-locator and erasure value calculation module to calculate the error-locator *UERR* and save *UERR* as an erasure-locator. When GATE4 is low, the error-locator and erasure value calculation module is selected to calculate the erasure value *LV*.

(13) CU_1, CU_2, CU_3, CU_4 are used to control the data input of the erasure-locator polynomial calculation module. When CU_1 (or CU_3) is high, the current erasure locator LO1 (or LO3) from the erasure-locator calculation module is passed to the erasure-locator polynomial calculation module via the data bus LO13. Similarly, when CU_2 (or CU_4) is high, the current erasure-locator LO2 (or LO4) from the erasure-locator calculation module is passed to the erasure-locator polynomial calculation module via the data bus LO24.

(14) CLEAR is used to reset the sigma circuit after the proper coefficients of the erasure-locator polynomial are latched.

(15) SIGNAL13 and SIGNAL24 are the signals which are put into the sigma-clock circuit to synthesise SIGNAL, the LFSR triggering clock of the sigma circuit.

(16) CTL is the latch clock of the sigma circuit to latch the proper coefficients of the erasure-locator polynomial calculated by the sigma circuit.

(17) SAMPLE is the sampling clock to sample the final correct result after the error or the erasure symbol has been removed from the decoding symbol.

4.6 Operation sequence and simulation results

Assume that $r(X) = r_3 X^4 + r_2 X^7 + r_1 X^{11}$ is the received block, where $r_1 = 5$, $r_2 = 7$, $r_3 = 6$, and r_1 is an error, while r_2, r_3 are erasures. The simulation results of the (15, 11) RS decoder is illustrated in Fig. 6. The period of CKX8 equals to 50 time units and that of CK equals to 800 time units. The operation sequence of the decoder is described as follows:

(1) In the first 15 cycles (time 0 to 12000), GATE1 is slow, so $r(X)$ is shifted into the buffer module and in the same time, syndrome values and erasure-locators are calculated in the first syndrome and erasure-locator calculation module.

(2) At time 12000, GATE1 switches to high and the next word is ready for decoding, while the results calculated in the first syndrome and erasure-locator calculation module are switched to the errors-and-erasures correction module.

(3) At time 12400, GATE4 switches from low to high to select the error-locator and-erasure value calculation module to compute the error-locator of the first received word and save the computed error-locator as erasure-locator. Since r_3 is an error and its locator is α^{11}, the calculated *UERR* is equal to *EH*, where *H* denotes the hexdecimal representation. GATE4 switches from high to low after the UERR is obtained.

(4) At time 12400, the highest order symbol of $r(X)$ is shifted out and the syndrome values and erasure-locators in the first syndrome and erasure-locator calculation module are cyclically shifted once, while the first symbol of the next block is received and its syndrome values and erasure-locators are calculated in the second syndrome and erasure-locator calculation module.

(5) From time 12400 to time 24400, the error and erasure correction module checks each LSFRs of the erasure-locator. If one of the erasure-locators is found to be 1 (i.e. α^0), then it indicates that the currently shifted out symbol is an erasure, and the erasure value *LV* will be calculated from the errors-and-erasures correction module.

(6) After the erasure value *LV* is known, *LV* is added to the output of the buffer module BO, which generates the output AD. The output AD is sampled at the rising edge of SAMPLE and the final decoded result of the first symbol of $r(X)$ is obtained as the output OUT.

The simulation result in Fig. 6 illustrates that the error symbol r_1 of $r(X)$ is found at time 15200 and the erasure symbols r_2, r_3 are found at time 18400 and 20800, respectively. The final decoded word of $r(X)$ will be zero which is a codeword of the (15, 11) RS code.

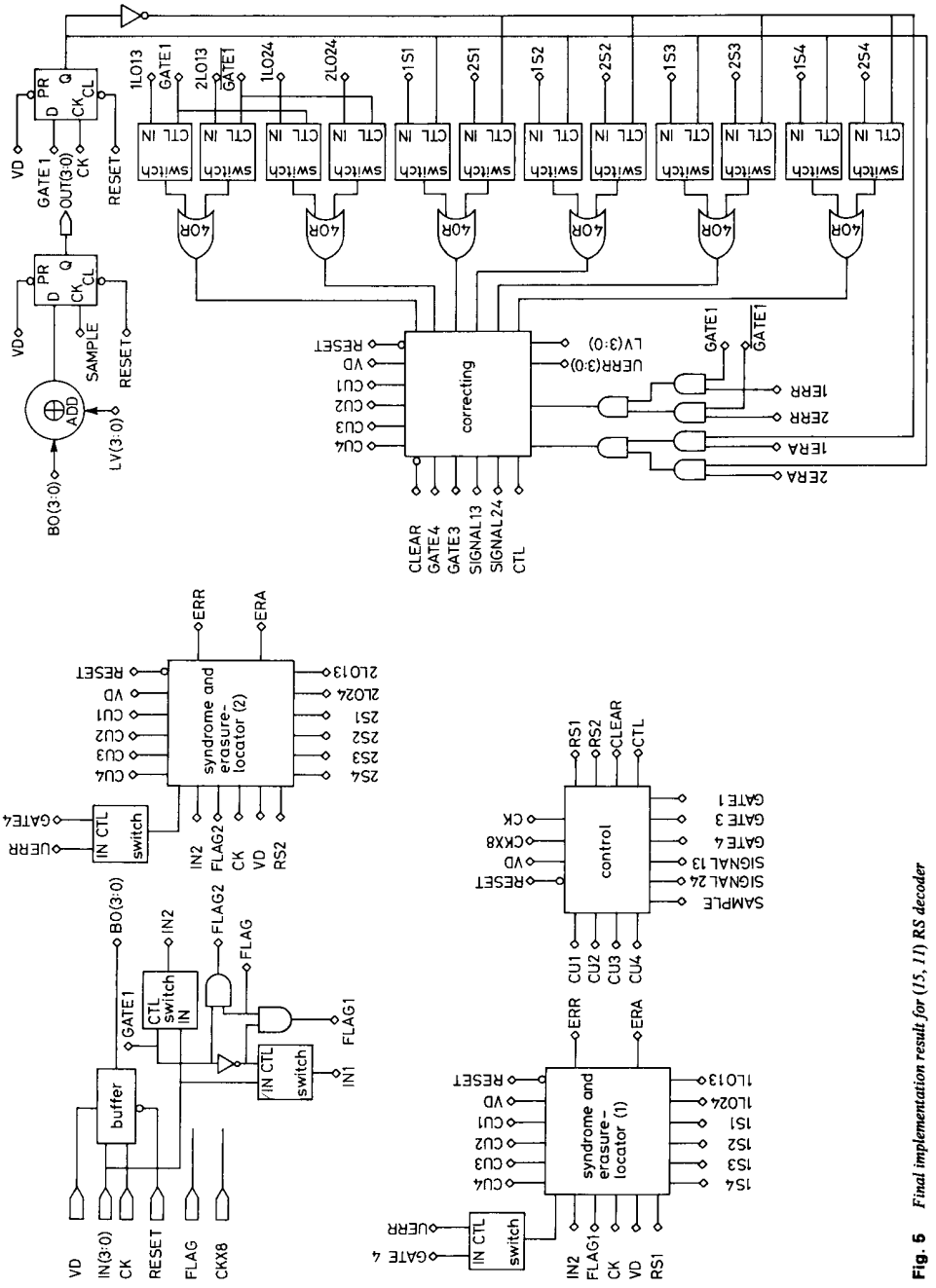

Fig. 6 *Final implementation result for (15, 11) RS decoder*

As described above, the calculation of the erasure-locator polynomial and the error-locator (or erasure values) must be completed within one CK cycle, where CK is the data rate of the received block. Therefore, the values and erasure-locators corresponding to the first decoded symbol of the shortened RS code can be obtained by multiplying $r(X)$ by X^8 before calculating syndrome and erasure-locators.

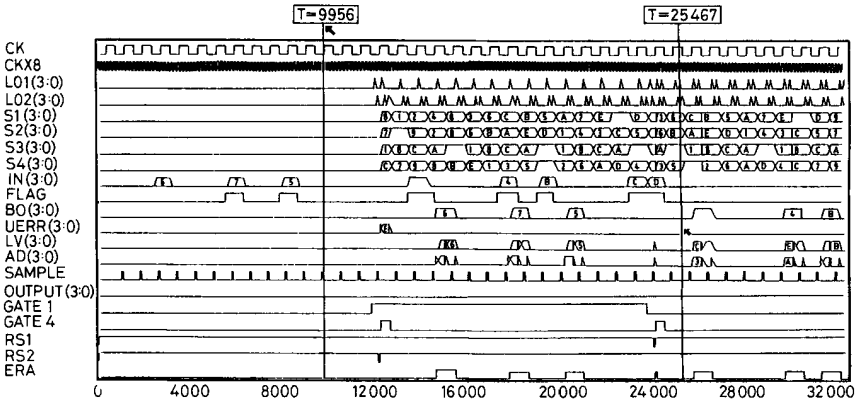

Fig. 6 *Simulation result for (15, 11) RS decoders*

working data rate of the decoder is dominantly determined by the computation time of the errors-and-erasures correction module, especially that of multiplications. The decoder works successfully when the period of CK is 800 time-units, assuming that one gate delay is one time-unit. Furthermore, if an IC technology with 0.5 ns gate delay is employed, for example 1.0 μm CMOS process, then the working data rate of this decoder will be up to 2M symbols/s.

5 Conclusion

A high-speed RS decoder for errors-and-erasures correction based on a new algebraic decoding method has been presented. A serial structure which provides a general expression for calculating the erasure-locator polynomial is also proposed. Consequently, a (15, 9) RS decoder with $d_{min} = 7$ can be also applied to decode the (15, 11) or (15, 13) RS code with $d_{min} = 5$, 3, respectively, by only modifying the control module. The (7, 3) RS code, a shortened code derived from (15, 11) RS code, can be decoded by the same decoder structure with slight modification of the buffer module, syndrome and erasure-locator correction module, and control module. Since the block length of this code is 7, only 8 one-symbol registers are needed in the buffer module to store the received block and latch up the shifted-out symbol. The syndrome

Table 1: Gate counts of (15, 11) RS decoder and (7, 3) RS decoder

Module	Decoders	
	(15, 11)RS	(7, 3)RS
Buffer	256	128
Syndrome	324	368
Erasure-locator	344	344
Modified-syndrome	469	469
Sigma-clock	25	25
Sigma	313	313
Error-locator and Erasure-value	145	145
Control	108	104
Other	277	277
Total	2153	2069

The numbers of gates required in the (15, 11) RS decoder and the (7, 3) RS decoder are shown in Table 1. A brief comparison in hardware complexity and computation time between the standard algebraic decoding method and the new algebraic decoding method for RS decoder is also given in Table 2, where the chip area taken by a $GF(2^4)$ multiplier is defined as one unit a and a multiplier delay is defined as one unit mt.

Table 2: Comparison of standard algebraic decoding algorithm and new standard algebraic decoding algorithm for RS decoder

d_{min}	Standard algebraic decoding method			New standard algebraic decoding method		
	Complexity	Delay	Control clocks	Complexity	Delay	Control clocks
5	11a	2mt	Complex	4a	1mt	Simple
7	17a	3mt	Complex	6a	1mt	Simple

6 References

1 BLAHUT, R.E.: 'Theory and practice of error control code' (Addison–Wesley, Reading, MA, 1983)
2 LIN, S., and COSTELLO, D.J. Jr.: 'Error control coding: fundamentals and applications' (Prentice–Hall, Englewood Cliffs, NJ, 1983)
3 PETERSON, W.W., and WELDON, E.J. Jr.: 'Error-correcting codes' (MIT Press, Cambridge, MA, 1972)
4 MICHELSON, A.M., and LEVESQUE, A.H.: 'Error-control techniques for digital communication' (Wiley, New York, 1985)
5 FORNEY, G.D. Jr.: 'On decoding BCH codes', *IEEE Trans. Inform. Theory*, Oct. 1965, **IT-11**, pp. 549–557
6 SHAO, H.M., TRUONG, T.K., DEUTSCH, L.J., YUEN, J.H., and REED, I.S.: 'A VLSI design of a pipeline Reed–Solomon decoder', *IEEE Trans. Comput.*, May 1985, **C-34**, pp. 393–402
7 SHAO, H.M., and REED, I.S.: 'On the VLSI design of a pipeline Reed–Solomon decoder using systolic arrays', *IEEE Trans. Comput.*, Oct. 1988, **C-37**, pp. 1273–1280
8 WEI, S.W., and WEI, C.H.: 'High speed hardware decoder for double-error-correcting binary BCH codes', *Proc. IEE*, June 1989, **136**, Pt. I, (3), pp. 227–231
9 WEI, S.W., and WEI, C.H.: 'On high-speed decoding of the (23, 12, 7) Golay code', *IEEE Trans. Inform. Theory*, May 1990, **IT-36**, (3), pp. 692–695

10 LAWS, B.A., and RUSHFORTH, C.K.: 'A cellular-array multiplier for $GF(2^m)$', *IEEE Trans. Comput.*, Dec. 1971, **C-20**, pp. 1573–1578
11 WANG, C.C., TRUONG, T.K., SHAO, H.M., DEUTSCH, L.J., OMURA, J.K., and REED, I.S.: 'VLSI architecture for computing multiplications and inverses in $GF(2^m)$', *IEEE Trans. Comput.*, Aug. 1985, **C-34**, pp. 709–716
12 KIM, S.W.: 'Error control codes for digital recording systems', *IEEE Trans. Consumer Electronics*, Nov. 1989, **35**, (4), pp. 907–916
13 GOTO, H., ASADA, A., CHIBA, H., SAMPEI, T., NOGUCHI, T., and ARAKAWA, M.: 'A new concept of DATA/DAT system', *IEEE Trans. Consumer Electronics*, Aug. 1989, **35**, (3), pp. 660–670
14 HAYASHI, K.: 'Error correction method for R-DAT and its evaluation', *IEEE ICASSP*, Tokyo, 1986, pp. 9–12
15 CLARK, G.C. Jr., and CAIN, J.B.: 'Error-correction coding for digital communications' (Plenum Press, New York, 1982)