

High-Speed Decoder of Reed-Solomon Codes

Shyue-Win Wei and Che-Ho Wei

Abstract—A high speed decoding algorithm using a modified step-by-step method for t -error-correcting Reed-Solomon codes is introduced. Based on this algorithm, a sequential decoder and a vector decoder are then proposed. These two decoders can be constructed by four basic modules: the syndrome calculation module, the comparison module, the decision module, and the shift-control module. These decoders can be applied for both binary and nonbinary data transmissions working at high data rate. Because of the simplicity in structure and circuit realization, a decoder employing this algorithm can be easily implemented in a monolithic chip by the VLSI technology.

I. INTRODUCTION

THE Reed-Solomon (RS) codes are a special subclass of the nonbinary BCH codes in which the symbol field and the locator field are the same. The RS codes that applied for binary data transmission over $GF(2^m)$ can be implemented by simply treating successive m -bit blocks as 2^m -ary symbols, and will outperform binary codes with the same rate and block length at low output error rates [1]. In general, two primary approaches, algebraic decoding method and transform method, can be employed to decode the RS codes. The transform method is a fast decoding method for large t -error-correcting RS codes. However, since this method needs to perform the transform operation, it is comparable to the standard algebraic method in algorithm and implementation for small t RS codes [2]–[8].

Another algebraic decoding algorithm for BCH codes, known as step-by-step decoding method, was presented by Massey [9] in 1965. The basic principle of the step-by-step decoding algorithm for nonbinary t -error-correcting BCH codes can be summarized as follows: 1) calculate the syndrome values 2) change the syndrome values, using an iteration method, to make the weight of error pattern exactly equal to t . 3) temporarily add in order all the $q - 1$ nonzero elements of $GF(q)$ to the decoding symbol of the received word with testing to determine whether the weight of the error pattern has been reduced. If the weight of error pattern is reduced to $t - 1$, then both the error location and the corresponding error value are found. 4) shift $r(x)$ one symbol and repeat step 3). The difference between the step-by-step method and the standard algebraic method is that the step-by-step method decodes every potential error location and value directly instead of searching the roots of error location polynomial and evaluating the error values.

Paper approved by M. Shahshahani, the Editor for Coding and Applications of the IEEE Communications Society. Manuscript received August 6, 1989; revised March 18, 1992.

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

IEEE Log Number 9211382.

Recently, a new step-by-step method for decoding double-error-correcting binary BCH code has been presented by the authors [10]. In this paper, by modifying the conventional step-by-step algorithm, a high-speed decoding algorithm for RS codes is presented in Section III. A sequential decoder based on the decoding algorithm is then proposed in Section IV. The decoder comprises a syndrome calculation module, a comparison module, a decision module, and a shift-control module. Furthermore, in Section V, a vector structure of line speed decoder for short n small t RS codes is presented. The operation clock rate of the new vector decoder is equal to the line rate, and one complete received word can be decoded within only n clock cycles.

II. REED-SOLOMON CODES

A t -error-correcting Reed-Solomon code with symbols from the Galois field $GF(2^m)$ has the following parameters [2]–[5]:

$$\text{Block length} = n = 2^m - 1.$$

$$\text{Number of parity-check symbols} = n - k = 2t.$$

$$\text{Minimum distance} = d_{\min} = 2t + 1.$$

Let α be a primitive element in $GF(2^m)$. The generator polynomial $g(x)$ of a t -error-correcting RS code of length $2^m - 1$ is the polynomial of degree $n - k$ with coefficients from $GF(2^m)$, and has $\alpha, \alpha^2, \dots, \alpha^{2t}$ as its roots. Let $k(x)$ be the information polynomial, then the encoded codeword $c(x)$ in systematic form can be obtained as $c(x) = k(x)x^{n-k} + \text{Mod}\{k(x)x^{n-k}/g(x)\}$ where $\text{Mod}\{k(x)x^{n-k}/g(x)\}$ indicates the remainder polynomial of $k(x)x^{n-k}$ divided by $g(x)$. Hereinafter, all the codewords are assumed to be in systematic form.

Let $e(x)$ be a received error pattern, then the received word is given by

$$\begin{aligned} r(x) &= c(x) + e(x) \\ &= r_0 + r_1x + r_2x^2 + \dots + r_{n-k}x^{n-1}. \end{aligned} \quad (1)$$

The coefficients of $c(x)$, $e(x)$, and $r(x)$ are elements from $GF(2^m)$. The syndrome values of a received word can be obtained from

$$S_i^0 = r(\alpha^i) = e(\alpha^i) \in GF(2^m) \quad i = 1, 2, \dots, 2t. \quad (2)$$

The syndrome values can also be written as

$$S_i^0 = \sum_{j=1}^t Y_j X_j^i \quad i = 1, 2, \dots, 2t \quad (3)$$

where X_j is the error location of the j th error symbol and Y_j is the corresponding error value. Therefore, the decoding task

is, given the syndrome values, to find the error locators and the error values.

III. DECODING ALGORITHM

A modified step-by-step decoding algorithm is presented in the following. The algorithm corrects the errors directly in terms of the differences between the original syndrome values and the temporarily changed syndrome values. This idea is based on the fact that various weights of error patterns can be distinguished from one another in terms of the relations among syndrome values. The relations among syndrome values of the RS codes can be found by using property 5 of [9], theorem 9.9 of [11], or theorem 7.2.2 of [3]. These theorems state the fact that if the number of errors is $j - 1$ or less then the syndrome matrix N_j^0 is singular, and if the number of errors is j then the syndrome matrix N_j^0 is nonsingular, where

$$N_j^0 = \begin{bmatrix} S_1^0 & S_2^0 & S_3^0 & \cdots & S_j^0 \\ S_2^0 & S_3^0 & S_4^0 & \cdots & S_{j+1}^0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ S_j^0 & S_{j+1}^0 & S_{j+2}^0 & \cdots & S_{2j-1}^0 \end{bmatrix}$$

Since the step-by-step decoding method involves changing received symbols one at a time with testing to determine whether the weight of the error pattern has been reduced, the weight of error pattern in the decoder may be $t + 1$ or less. Therefore, the variation in the weight of the error pattern from 0 up to $t + 1$ should be discriminated by the decoder if the step-by-step method is employed. That is, various weights of the error patterns should be distinguished from one another. The weight of error pattern can be determined from the values of $\det(N_j^0)$ ($j = 1, 2, \dots, t + 1$), where $\det(N_j^0)$ indicates the determinant of the syndrome matrix N_j^0 . However, the calculation of $\det(N_{t+1}^0)$ requires the syndrome value S_{2t+1}^0 which can not be obtained from the received words for a t -error-correcting RS code. Thus, it is necessary to define a modified syndrome matrix $\det(\bar{N}_{t+1}^0)$ as

$$\det(\bar{N}_{t+1}^0) = \det(\bar{N}_{t+1}^0) + S_{2t+1}^0 \det(N_t^0) \quad (4)$$

where the submatrix N_t^0 is the cofactor of element S_{2t+1}^0 in matrix N_{t+1}^0 . Clearly, $\det(\bar{N}_{t+1}^0)$ is composed of S_j^0 , $j \leq 2t$. Consequently, if $t - 1$ or less errors have occurred, then $\det(\bar{N}_{t+1}^0) = 0$ because $\det(N_{t+1}^0) = \det(N_t^0) = 0$; if $t + 1$ errors have occurred and $\det(N_t^0) = 0$, then $\det(\bar{N}_{t+1}^0) \neq 0$ because $\det(N_{t+1}^0) \neq 0$. Since we are only concerned whether the values of $\det(N_j^0)$, $j = 1, 2, \dots, t$, and $\det(\bar{N}_{t+1}^0)$ are equal to zero, the results can be expressed by $t + 1$ binary decision bits, h_j^0 ($j = 1, 2, \dots, t + 1$), defined by

$$h_j^0 = 1 \quad \text{if } \det(N_j^0) = 0 \quad j = 1, 2, \dots, t \quad (5a)$$

$$h_{t+1}^0 = 1 \quad \text{if } \det(\bar{N}_{t+1}^0) = 0 \quad (5b)$$

and, a $(t + 1)$ -tuple decision vector H^0 is then defined as

$$H^0 = (h_1^0, h_2^0, \dots, h_{t+1}^0). \quad (6)$$

Suppose the $\det(N_j^0)$ for $j = 1, 2, \dots, t$ and $\det(\bar{N}_{t+1}^0)$ can be found, the decision vector of a general t -error-correcting RS code can be determined as follows:

If there is no error, then $H^0 \in \phi_0 = \{(1^{t+1})\}$ where 1^{t+1} indicates $t + 1$ consecutively identical tuple of "1". For example, vector $(1^3) = (1, 1, 1)$.

If there is one error, then $H^0 \in \phi_1 = \{(0, 1^t)\}$.

If there are p errors, $2 \leq p < t$, then $H^0 \in \phi_p = \{(X^{p-1}, 0, 1^{t-p+1})\}$, where the symbol "X" can be "0" or "1."

If there are t errors, then $H^0 \in \phi_t = \{(X^{t-1}, 0, X)\}$.

If there are $t + 1$ errors, then $H^0 \in \phi_{t+1} = \{(X^{t-1}, 0, X), (X^{t-1}, 1, 0)\}$.

Where, ϕ_j ($0 \leq j \leq t + 1$) is a set of all possibly happened decision vectors that j errors have occurred.

From the above rules, we find that a) for the error patterns whose weights are t or less, the various weights of error patterns can be distinguished from one another b) the error pattern with weight $t + 1$ can be distinguished from the error pattern whose weight is $t - 1$ or less. Thus, the number of errors can be correctly determined in terms of the pattern of decision vector if and only if the weight of error pattern is t or less. It will be shown later that the above two statements ensure that any combination of t errors or less can be correctly decoded without ambiguity. Since the RS codes are an important subclass of cyclic codes, if the first symbol of $r(x)$ can be decoded correctly for all correctable error patterns, then the entire word can be decoded correctly with the same circuitry [11]. Let us first denote that

$$S_i^1 = S_i^0 + \beta, \quad \beta = \alpha^j, \\ j \in \{0, 1, \dots, n - 1\} \quad (7)$$

where S_i^1 ($i = 1, 2, \dots, 2t$) are the syndrome values of $r(x) + \beta$. According to the above definitions, some other decision bits can be defined in the following:

$$h_j^1 = 1 \quad \text{if } \det(N_j^1) = 0 \quad j = 1, 2, \dots, t \quad (8a)$$

$$h_{t+1}^1 = 1 \quad \text{if } \det(\bar{N}_{t+1}^1) = 0 \quad (8b)$$

where

$$N_j^1 = \begin{bmatrix} S_1^1 & S_2^1 & S_3^1 & \cdots & S_j^1 \\ S_2^1 & S_3^1 & S_4^1 & \cdots & S_{j+1}^1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ S_j^1 & S_{j+1}^1 & S_{j+2}^1 & \cdots & S_{2j-1}^1 \end{bmatrix} \\ j = 1, 2, \dots, t + 1$$

and

$$\det(\bar{N}_{t+1}^1) = \det(N_{t+1}^1) + S_{2t+1}^1 \det(N_t^1).$$

Finally, these decision bits can be used to form a decision vector H^1 :

$$H^1 = (h_1^1, h_2^1, \dots, h_{t+1}^1). \quad (9)$$

Theorem 1: For a t -error-correcting RS code, if ϕ_j ($j = 1, 2, \dots, t$) can be distinguished from one another and ϕ_{t+1} can be distinguished from ϕ_j ($j \leq t - 1$), then any error pattern whose weight is t or less can be corrected by a step-by-step decoding algorithm.

Proof: 1) If the weight of the received error pattern is 1, then $H^0 \in \phi_1$. Consider temporarily changing the received symbols r_{n-1}, \dots, r_0 one at a time through all n possible error values ($\beta = \alpha^p, p \in \{0, 1, \dots, n-1\}$). Suppose that r_j is an erroneous symbol and β is the error value, then changing r_j will reduce the weight of error pattern and hence $H^1 \in \phi_0$. Suppose r_j is an erroneous symbol but β is not the error value, then only error value of r_j is changed while the weight of error pattern is not changed, thus, $H^1 \in \phi_1$. Suppose r_j is a correct symbol, then changing r_j will increase the weight of error pattern to two and hence $H^1 \in \phi_2$. Since ϕ_0, ϕ_1 , and ϕ_2 can be distinguished from one another, the error pattern can be correctly decoded.

2) If the weight of the received error pattern is 2, then $H^0 \in \phi_2$. Consider temporarily changing the received symbols r_{n-1}, \dots, r_1 one at a time through all n possible error values. Suppose r_j is an erroneous symbol and β is the error value, then changing r_j will make $H^1 \in \phi_1$. Suppose r_j is an erroneous symbol but β is not the error value, then $H^1 \in \phi_2$. Suppose r_j is a correct symbol, then changing r_j will make $H^1 \in \phi_3$. Since ϕ_1, ϕ_2 , and ϕ_3 can be distinguished from one another, the error symbol can be corrected. After the first error symbol has been corrected, this case is reduced to case 1).

•
•
•

t) If the weight of the received error pattern is t , then $H^0 \in \phi_t$. Consider temporarily changing the received symbols r_{n-1}, \dots, r_{t-1} one at a time through all n possible error values. Suppose r_j is an erroneous symbol and β is the error value, then changing r_j will make $H^1 \in \phi_{t-1}$. Suppose r_j is an erroneous symbol but β is not the error value, then $H^1 \in \phi_t$. Suppose r_j is a correct symbol, then changing r_j will make $H^1 \in \phi_{t+1}$. Since ϕ_t and ϕ_{t+1} can be distinguished from ϕ_{t-1} , the error symbol can be corrected. After the first error symbol has been corrected, this case is reduced to case $t-1$).

In summary, any combination of t or less can be decoded correctly with a step-by-step method.

From Theorem 1, we have the following corollary:

Corollary 1: a) If the error pattern whose weight is $t-1$ or less, then we need only to consider changing the decoder symbol r_j ($j \leq n-1$) once with testing a nonzero element (say, α^0). If $H^0 \in \phi_j$ and $H^1 \in \phi_{j+1}$ ($j < t$), then the symbol r_j can be determined immediately as a correct symbol without testing any other nonzero elements. (b) If the error pattern whose weight is exactly equal to t , then consider changing the decoding symbol r_j ($j \leq n-1$) with testing value $\beta = \alpha^p$, $p = 0, 1, \dots, n-1$. If $H^0 \in \phi_t$ and $H^1 \in \{(x^{t-1}, 1, 0)\}$ for a testing β with $p < n-1$, then the symbol r_j can be determined as a correct symbol without testing the other nonzero elements.

Using Theorem 1 and Corollary 1, the modified step-by-step decoding algorithm of a t -error-correcting RS code can be described as follows:

1) Calculate the initial syndrome values S_i^0 ($i = 1, 2, \dots, 2t$) and then obtain H^0 .

2) Shift $r(x)$.

3) Let $j = 0$.

4) Let $\beta = \alpha^j$, then obtain $S_i^1 = S_i^0 + \beta$ and H^1 .

5) If $H^0 \in \phi_p$ and $H^1 \in \phi_{p+1}$ (where, $0 \leq p < t$), then go to step 10).

If $H^0 \in \phi_t$ and $H^1 \in \{(x^{t-1}, 1, 0)\}$, then go to step 10).

7) If $H^0 \in \phi_p$ and $H^1 \in \phi_{p-1}$ (where, $1 \leq p < t$), then change the magnitude of the first symbol of the shifted $r(x)$ by adding the value β . Replace S_i^0 and H^0 by S_i^1 and H^1 , respectively, and go to step 10).

8) If $H^0 \notin \{\phi_0, \phi_1, \dots, \phi_{t-1}\}$ and $H^1 \in \phi_{t-1}$, then change the magnitude of the first symbol of shifted $r(x)$. Replace S_i^0 and H^0 by S_i^1 and H^1 , respectively, and go to step 10).

9) If $j \leq n-2$, the set $j = j+1$ and go to step 4).

10) If all the k information symbols have been checked and corrected, then this algorithm is completed; otherwise, go to step 2).

This modified step-by-step algorithm needs only $n+k$ shift operations to decode one received word where n shift operations are used for calculating the initial syndrome values S_i^0 in step 1) and the other k shift operations are used for correcting the errors in information part. Furthermore, steps 4)–9) constitute an iteration loop used to test the n or less possible nonzero error values for decoding a symbol of $r(x)$. Theoretically, the number of total iterations for decoding a received word is in the range $(k, n \times k)$. However, the worst case (i.e., the case with $n \times k$ iterations) occurs only when the following two conditions are met: a) t errors have occurred, and the location of highest order error falls in the parity check part (i.e., $X_t \leq \alpha^{n-k-1}$, or $X_t = \alpha^{n-k}$ and $Y_t = \alpha^{n-1}$). b) H^1 of X_t never belongs to $\{(x, \dots, x, 1, 0)\}$ in the testing for all k information symbols. In fact, for most of the RS codes, the probability of worst case is very small. Clearly, when the order of location of the t th error is higher, the required number of iterations is smaller. If the weight of the error pattern is less than t , then the number of iterations would be less than $k + Y_1 + Y_2 + \dots + Y_{t-1}$. In practice, the average number of iterations for a specified RS code over the transmission medium can be estimated by using computer simulation. The estimated average number of iterations is related to the decoding speed of the decoder and the required size of the buffer that preceded the decoder. For example, consider the (63, 59, 7) RS code applied in a mobile fading channel with a fading frequency $f_d = 20$ Hz and signal-to-noise ratio (SNR) = 20 dB, the average number of iterations is 317 per word, which is only eleventh of the worst case (i.e., $63 \times 59 = 3591$ iterations). Furthermore, as the SNR increases to 27.5 dB, the average number of iterations is reduced to 114 per word because the occurrence probability of three errors becomes smaller. Comparing this modified step-by-step decoding algorithm to the conventional step-by-step algorithm, we find that the conventional decoding algorithm has two drawbacks: 1) it needs an iteration loop to change the received syndrome values to obtain an error pattern with exact weight t 2) it adds in order all the nonzero elements in $\text{GF}(2^m)$ to every correct symbols. That is, it requires $n(k-t) + Y_1 + Y_2 + \dots + Y_t$ iterations (nearly $n \times k$ for small t) for every received word. Therefore, the modified

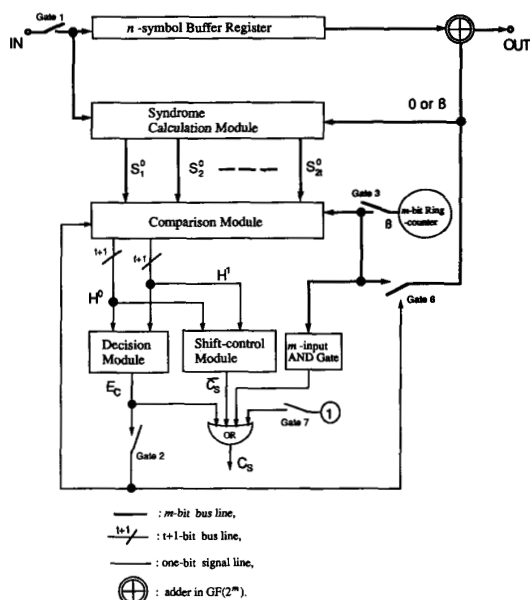


Fig. 1. The functional block diagram of the sequential decoder.

step-by-step decoding algorithm is much faster than the conventional step-by-step decoding algorithm.

IV. SEQUENTIAL DECODER

Based on the modified step-by-step decoding algorithm, a structure of sequential decoder is presented. Fig. 1 shows the functional block diagram of the decoder, which comprises a syndrome calculation module, a comparison module, a decision module and a shift-control module. The syndrome calculation module is used to calculate the syndrome values S_1, S_2, \dots, S_{2t} . The comparison module is used to obtain the temporarily changed syndrome values S_i^1 and to determine the decision bits $h_j^l, l = 0, 1; j = 1, 2, \dots, t + 1$. From these decision bits, the decision module can determine whether the decoding symbol of $r(x)$ with a testing value β is an erroneous symbol. If the corresponding erroneous symbol and its error value have been found, the decoder sends a correcting-bit E_c to change the magnitude of the decoding symbol; that is, adding β to the decoding symbol. The syndrome calculation module has been described well in [3] in detail while the design of the other three modules are described in the following.

A. Comparison Module

The comparison module is used to obtain the temporarily changed syndrome values and then to determine the corresponding decision vector. The block diagram of the comparison module for a t -error-correcting RS code is illustrated in Fig. 2. In Fig. 2, the module first obtains the syndrome values S_i^0 or S_i^1 , then, a matrix calculation circuit is used to calculate the determinant of syndrome matrices $\det(N_j^l) \{j = 1, 2, \dots, t\}$ and $\det(\bar{N}_{t+1}^l)$ where $l = 0$ or 1 . When the input are S_i^0 (i.e., gate 3 is open), the matrix calculation obtains $\det(N_j^0)$ and $\det(\bar{N}_{t+1}^0)$; when the input are S_i^1 (i.e., gate 3

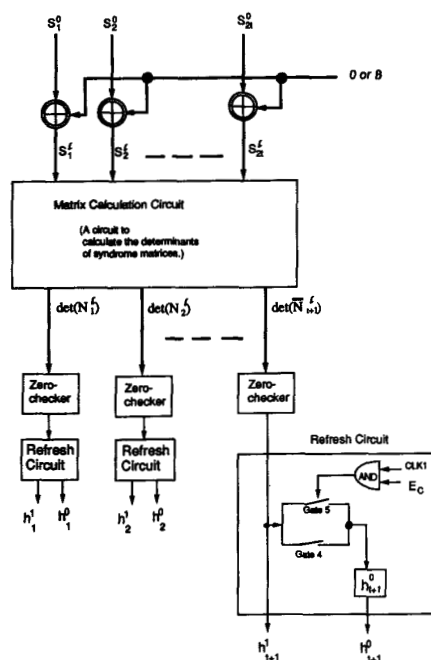


Fig. 2. The comparison module of t -error-correcting RS codes.

is closed), then $\det(N_j^1)$ and $\det(\bar{N}_{t+1}^1)$ are obtained. After finding the values of $\det(N_j^l) (j = 1, \dots, t)$ and $\det(\bar{N}_{t+1}^l)$, the decision bits h_j^l can be determined by using $t + 1$ simple zero-checkers, which are implemented by $t + 1$ pieces of m -input NOR gates. Besides, in Fig. 2, the decision bits h_j^l must be refreshed if an error value has been found, therefore $t + 1$ refresh-circuits are cascaded with the zero-checkers, respectively. In each refresh circuit, gate 4 and the cascaded storage stage are used to save the initial decision bit h_j^0 while gate 5 is used to perform the refreshing operation. That is, replace h_j^0 by h_j^1 .

B. Decision Module

The decision module is used to perform the operations in steps 7) and 8) of the decoding algorithm. When the decision vectors H^0 and H^1 are determined in the comparison module, the decision module can then decide whether or not the testing value β of the corresponding symbol is an error value in terms of the difference between H^0 and H^1 . The decision module can be easily implemented by employing a logical circuit or a ROM of size $2^{2t+2} \times 1$ bits. After the decision, the decision module sends a correcting bit $E_c = 1$ or $E_c = 0$ to control gate 6 for correcting the corresponding symbol, and then refresh the decision bits in the comparison module.

C. Shift-Control Module

The shift-control module is used to perform the operations in steps 5) and 6) of the decoding algorithm. If the module's output is true, $C_s = 1$, the current decoding symbol must be a correct symbol and then the decoder starts to decode the next symbol. The inputs of the module are the same as that of

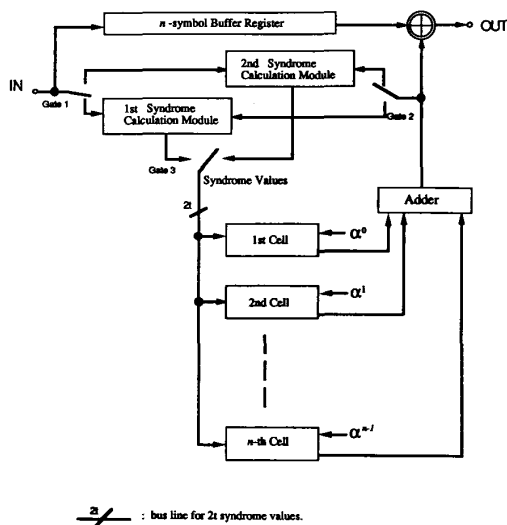


Fig. 3. The functional block diagram of the vector decoder.

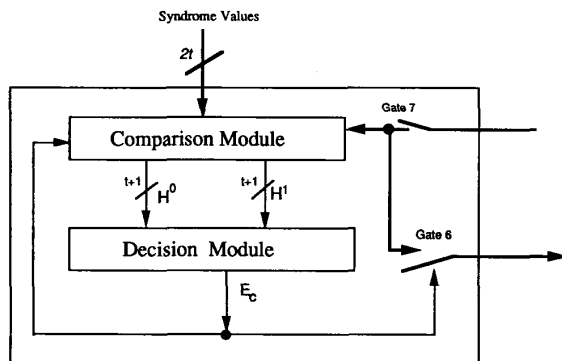


Fig. 4. The block diagram of one cell of the vector decoder.

the decision module, and can also be implemented by a logic circuit or ROM circuit.

V. VECTOR DECODER

In the modified step-by-step decoding algorithm, the operations in steps 4)–9) constitute an iteration loop for testing the n (or less) nonzero values β . If steps 3), 5), 6), and 9) are removed, and the testing of n values of β are simultaneously performed in parallel, then the algorithm can be modified as a vector decoding algorithm since the symbols of received word are decoded one at a time with testing a n -tuple vector which is composed of the n possible error values. Based on the vector algorithm, a new vector structure of line speed decoder is presented. In this line speed decoder, the testing of n values of β can be performed in space instead of in time iteratively. The functional block diagram of the vector decoder, as shown in Fig. 3, is composed of two syndrome calculation modules and n pieces of identical cells where every cell is integrated by a comparison module and a decision module with a fixed testing value α^j shown in Fig. 4. The comparison module and decision module are the same as that described in Section VI.

By duplicating the syndrome calculation module, the decoder may work at an equal speed of line rate, with a group delay of n clock cycles. The possible working data rate of this decoder is determined by the computation time of the cell. For example, in the longest computation path of double-error-correcting RS codes, it needs only to perform one addition, one multiplication and several logic gate delays. By the current technology, the calculation time of the comparison and decision modules can be accomplished within only a few hundred nanoseconds and thus, the decoder can work at a line rate up to several millions symbols per second, that is, about several tens of megabits per second. In practice, this decoder is suitable for the short block length RS codes (i.e., small m and n). Usually, a small n is enough to obtain a high code rate for small t RS codes (e.g., $t = 2$, and $m = 5$, $n = 31$, code rate = 0.87). For these codes of small m , the circuit size of the comparison module and the decision module are very small. The vector decoder avoids the input buffer that required for the sequential decoder and improves the decoding speed, therefore, can work at higher data rate.

VI. CONCLUSIONS

A modification of the conventional step-by-step decoding algorithm for t -error-correcting RS codes has been presented. The algorithm corrects the errors by only comparing the number of errors in current cycle with that in the previous cycle. This decoding algorithm requires only n shift operations for reading the received word and k shift operations for correcting the erroneous symbols. Based on the decoding algorithm, a sequential decoder has been proposed. The sequential decoder needs n clock cycles to calculate the syndrome values and $k \times n$ (or less) clock cycles to decode the received error pattern. The possible highest working data rate of the decoder depends mainly on the computation time (i.e., propagation delay) of the matrix calculation circuit in the comparison module and the SNR of system over the specified transmission medium. In addition, the computation time of the matrix calculation circuit varies with t but is almost independent of m . Thus, this hardware decoder is suitable for high code rate of small t RS codes. Furthermore, parallel testing of n possible error values in space, a vector structure of line speed decoder has been also proposed for short block length RS codes, which requires only n clock cycles to decode one received word. Therefore, both the sequential decoder and the vector decoder can be applied for the nonbinary data transmission and the binary-form data transmission at high-speed data rate. Because of the simplicity in structure and circuit realization, these two types of decoders may be easily implemented in one chip by using VLSI technology.

REFERENCES

- [1] W. C. Gore, "Transmitting binary symbols with Reed-Solomon codes," in *Proc. Princeton Conf. Inform. Sci. Syst.*, 1973, pp. 495–497.
- [2] S. Lin and D. J. Costello, Jr., *Error Control Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1983.
- [3] R. E. Blahut, *Theory and Practice of Error Control Codes*. New York: Addison-Wesley, 1983.
- [4] G. C. Clark and J. B. Cain, *Error-Correcting Coding for Digital Communications*. New York: Plenum, 1981.

- [5] A. M. Michelson, and A. H. Levesque, *Error-Control Techniques for Digital Communication*. New York: Wiley, 1985.
- [6] I. S. Reed, R. A. Scholtz, T. K. Truong, and L. R. Welch, "The fast decoding of Reed-Solomon codes using Fermat theoretic transforms and continued fractions," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 100-106, Jan. 1978.
- [7] H. Okano and H. Imai, "A construction method of high-speed decoder using ROM's for Bose-Chaudhuri-Hocquenghem and Reed-Solomon codes," *IEEE Trans. Comput.*, vol. C-36, pp. 1165-1171, Oct. 1987.
- [8] H. M. Shao, T. K. Truong, L. J. Deutsch, J. H. Yuen, and I. S. Reed, "A VLSI design of a pipeline Reed-Solomon decoder," *IEEE Trans. Comput.*, vol. C-34, pp. 393-402, May 1985.
- [9] J. L. Massey, "Step-by-step decoding of the Bose-Chaudhuri-Hocquenghem codes," *IEEE Trans. Inform. Theory*, vol. IT-11, pp. 580-585, Oct. 1965.
- [10] S. W. Wei and C. H. Wei, "High speed hardware decoder for double-error-correcting binary BCH codes," *IEE Proc.*, vol. 136, Pt. 1, pp. 227-231, June 1989.
- [11] W. W. Peterson, and E. J. Weldon, Jr., *Error-Correcting Codes*. Cambridge, MA: M.I.T. Press, 1972, 2nd ed.