ICIC Express Letters
Volume **10**, Number **9**, September **2016**

ICIC International ©2016 ISSN 1881-803X
pp. **2235–2240**

A FAST ALGORITHM FOR CORRECTING ERASURES OF BCH CODES

Erl-Huei Lu[1], Chih-Wen Shih[1], Chia-Jung Li[1] and Tso-Cho Chen[2]

[1]Department of Electrical Engineering
Chang Gung University
No. 259, Wenhua 1st Rd., Guishan Dist., Taoyuan City 333, Taiwan
lueh@mail.cgu.edu.tw

[2]Department of Avionics
China University of Science and Technology
No. 200, Zhonghua St., Hengshan Township, Hsinchu County 312, Taiwan
noahchen8@gmail.com

Received March 2016; accepted June 2016

Abstract. *This letter presents an erasure correcting algorithm of binary BCH codes. The algorithm is much faster than a conventional erasure correcting method because it can directly correct erased bits without determining the error locator polynomial and finding the roots of the polynomial. Without loss of generality, a hardware decoder for correcting four erasures is also demonstrated. Since the hardware decoder has simple and regular structure, it is well-suited for VLSI implementation.*
Keywords: Hardware decoder, Erasure correcting, Soft-decision decoding, Turbo product codes, VLSI

1. **Introduction.** Soft-decision decoding (SDD) of a code can achieve 2-3dB of coding gain, compared with hard-decision decoding (HDD) [1]. To date, the Chase-II algorithm [2] has been one of the most efficient and widely-used SDD algorithms. However, the algorithm requires a lot of HDD and arithmetic operations for decoding a codeword. To reduce the computational complexity, Reid III et al. [3] developed a simple and fast SDD algorithm for BCH codes based on the error magnitude evaluating algorithm introduced in [4]. Notably, the algorithm in [4] is adopted to correct erasures of BCH codes by [3].

Turbo product codes (TPCs) can be iteratively decoded in soft-input soft-output (SISO) manner to achieve performances near the Shannon capacity limit [5]. The SISO decoding manner needs to process soft input information using SDD and yield extrinsic information for the next iterative decoding. Thus, it may not be suitable for some particular applications where precise soft information is not available or high-speed decoding is critical, such as data storage systems and optical fiber communication. Decoding a codeword in hard-input hard-output (HIHO) manner only requires one HDD such that it is much simpler and faster than that in SISO manner. Consequently, several HIHO decoding algorithms have been developed for decoding TPCs [6,7]. On the other hand, using HIHO algorithms to iteratively decode TPCs may appear closed-chains error patterns (CCEPs), which degrades the decoding performance [8]. To remedy the drawback, Al-Dweik and Sharif [7] adopted an erasure decoder to correct CCEPs.

A conventional method for correcting erasures of BCH codes needs to perform twice HDD [1], and each HDD includes two time-consuming processes: determining the error locator polynomial (ELP) and finding the roots of the ELP. Recently, an efficient algorithm was introduced to evaluate error magnitudes of RS codes [9], which uses a syndrome refining process to refine syndromes such that it can avoid computing the error evaluator polynomial. Motivated by the method, a fast algorithm is proposed in this letter to correct

erasures of BCH codes without determining the ELP and finding the roots of the ELP. Furthermore, since the error-magnitude extracting process in [9] has been simplified to form the estimating process of the new algorithm, the operation of computing inverses can be avoided by the new one. Finally, a hardware decoder for correcting erasures of BCH codes is designed based on the new algorithm. Since the hardware decoder is simple and regular, it is well-suited for VLSI implementation.

2. **Preliminaries and Notations.** An (n, k) t-error-correcting binary BCH code of length $n = 2^m - 1$ can be defined in terms of the roots of its generator polynomial. Let α be a primitive element in the Galois field $GF(2^m)$, where m is an integer with $m \geq 3$. Then the generator polynomial $g(x)$ is the lowest degree polynomial over $GF(2)$, which has α, α^2, ... and α^{2t} as its roots. Assume that there are v (for $v \leq 2t$) erasures in a received polynomial. Then, the error polynomial can be written as $e(x) = e_{l_1}x^{l_1} + e_{l_2}x^{l_2} + \cdots + e_{l_v}x^{l_v}$, where $e_{l_i} \in GF(2)$ for $i = 1, 2, \ldots, v$, and $0 \leq l_1 < l_2 < \cdots < l_v \leq n-1$. Let $\delta_i = e_{l_i}$ for $i = 1, 2, \ldots, v$. Then, the syndromes for correcting these erasures can be defined as

$$S_w = \delta_1 \alpha^{wl_1} + \delta_2 \alpha^{wl_2} + \cdots + \delta_v \alpha^{wl_v}, \text{ for } w = 1, 2, \ldots, 2t. \tag{1}$$

Let $r(x) = r_0 + r_1 x + \cdots + r_{n-1}x^{n-1}$ denote the received polynomial. Then these syndromes can be computed as

$$S_w = r(\alpha^w) = r_0 + r_1\alpha^w + \cdots + r_{n-1}\alpha^{w(n-1)}, \text{ for } w = 1, 2, \ldots, 2t. \tag{2}$$

For convenience, the notations defined in [9] are also used throughout this letter, which are as follows. Let $P_{i,0} = \alpha^{l_i}$, for $i = 1, 2, \ldots, v$, and

$$P_{i,j} = \alpha^{l_i} + \alpha^{l_j} = P_{i,0} + P_{j,0}, \text{ for } 1 \leq j < i \leq v. \tag{3}$$

We also define

$$Q_{i,0} = P_{i,0} \left(= \alpha^{l_i}\right), \text{ for } i = 1, 2, \ldots, v, \tag{4}$$

and

$$Q_{i,j} = \prod_{m=0}^{j} P_{i,m} = Q_{i,j-1}P_{i,j} = Q_{i,j-1}\left(P_{i,0} + P_{j,0}\right), \text{ for } 1 \leq j < i \leq v. \tag{5}$$

3. **Proposed Algorithm.**

3.1. **Decoding procedure.** Suppose that the $2t$ syndromes have been computed using (2). Then, the erasure-only decoding procedure of the proposed algorithm consists of a syndrome refining process and an erasure estimating process.

Syndrome refining process:

The computing procedure of the process is the same as the syndrome refining process in [9] as follows. Let $S_w^{(1)} = S_w$. Then, from (1) and (4) we have

$$S_w^{(1)} = S_w = \sum_{i=1}^{v} \delta_i Q_{i,0} P_{i,0}^{w-1}, \text{ for } w = 1, 2, \ldots, v. \tag{6}$$

From (5) and (6), we have

$$\begin{aligned} S_w^{(k)} &= S_{w+1}^{(k-1)} + S_w^{(k-1)} P_{k-1,0} \\ &= \sum_{i=k}^{v} \delta_i Q_{i,k-2}\left(P_{i,0} + P_{k-1,0}\right) P_{i,0}^{w-1}, \\ &= \sum_{i=k}^{v} \delta_i Q_{i,k-1} P_{i,0}^{w-1}, \text{ for } k = 2, 3, \ldots, v, \text{ and } w = 1, 2, \ldots, v-k+1. \end{aligned} \tag{7}$$

From (7), we know that the syndrome refining process needs to compute $S_w^{(k)} = S_{w+1}^{(k-1)} + S_w^{(k-1)} P_{k-1,0}$ for $k = 2, 3, 4, \ldots, v$, and $w = 1, 2, \ldots, v-k+1$, iteratively. Thus, the process involves $v(v-1)/2$ additions and $v(v-1)/2$ multiplications over $GF(2^m)$.

Erasure estimating process:

The process estimates erasures recursively, from δ_v to δ_1, as follows. Substituting $k = v$ into (7), we have $S_1^{(v)} = \delta_v Q_{v,v-1}$. Then, $\tilde{\delta}_v = 1$ if $S_1^{(v)} \neq 0$, and $\tilde{\delta}_v = 0$ if $S_1^{(v)} = 0$. Finally, we estimate $\tilde{\delta}_k$ from $\tilde{\delta}_{v-1}$ to $\tilde{\delta}_1$ recursively, as

$$\tilde{\delta}_k = \begin{cases} 1, & \text{if } S_1^{(k)} + \sum_{i=k+1}^{v} \tilde{\delta}_i Q_{i,k-1} \neq 0, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

From the estimating process, we know that $Q_{i,j-1}$ for $2 \leq j < i \leq v$ are needed for estimating erasures. By (5), these $Q_{i,j-1}$ can be computed recursively, as $Q_{i,1} = Q_{i,0}\,(P_{i,0} + P_{1,0})$, $Q_{i,2} = Q_{i,1}(P_{i,0} + P_{2,0})$, $\ldots$, $Q_{i,j-2} = Q_{i,j-3}(P_{i,0} + P_{j-2,0})$ for $2 \leq j < i \leq v$ such that $(v-1)(v-2)/2$ additions and $(v-1)(v-2)/2$ multiplications over $GF(2^m)$ are required. When all $Q_{i,j-1}$ have been obtained, we can compute $S_1^{(k)} + \sum_{i=k+1}^{v} \tilde{\delta}_i Q_{i,k-1}$ for estimating $\tilde{\delta}_k$ from $\tilde{\delta}_v$ to $\tilde{\delta}_1$ with $v(v-1)/2$ additions over $GF(2^m)$.

3.2. **Comparisons.** Suppose that there are v erasures in a received vector of BCH codes. As soon as the syndromes S_w, for $w = 1, 2, \ldots, v$, have been computed using (2), the v erased bits in $r(x)$ can be estimated using the proposed decoding algorithm. According to the above decoding procedure, we know that our algorithm totally requires $(v-1)(3v-2)/2$ additions and $(v-2)^2$ multiplications over $GF(2^m)$ for correcting v erasures, as listed in Table 1. As mentioned in Section 1, the conventional method in [1] needs to perform twice HDD for correcting v erasures of (n, k) BCH codes, where $n = 2^m - 1$. Notably, each of the HDDs is employed to correct t errors (where t is the error correction capability of the BCH codes), which consists of computing syndromes, determining the ELP and finding the roots of the ELP. Based on the Chien's procedure [1], finding all roots for two ELPs of degree t we totally need $2n(t-1)$ additions and $2nt$ multiplications over $GF(2^m)$. Thus, at least $2n(t-1)$ additions and $2nt$ multiplications over $GF(2^m)$ are involved in a conventional methord, as listed in Table 1. Note that $v \leq 2t$. In addition, from Table II in [9], we know that for correcting v erasures the Komo-Joiner algorithm [4] consists of $3v(v-1)/2$ additions, v^2 multiplications and $v(v-1)/2$ inverses over $GF(2^m)$. Thus, our algorithm has lower computational complexity than the conventional method and the Komo-Joiner algorithm.

4. **Hardware Implementation.** Based on the proposed decoding algorithm, the hardware decoder for correcting v erasures consists of a syndrome computing, a syndrome refining, a $Q_{i,j}$ computing and an erasure estimating units, as shown in Figure 1. Without loss of generality, the decoder which is capable of correcting four erasures (i.e., $d_{\min} = 5$)

TABLE 1. Computational complexities of the related algorithms for correcting v erasures

	Conventional algorithm [1]	Komo-Joiner algorithm [4]	Proposed algorithm
Additions over $GF(2^m)$	$> 2n(t-1)$	$3v(v-1)/2$	$(v-1)(3v-2)/2$
Multipilications over $GF(2^m)$	$> 2nt$	v^2	$(v-1)^2$
Inverses over $GF(2^m)$	0	$v(v-1)/2$	0
Determining ELP	necessary	unnecessary	unnecessary

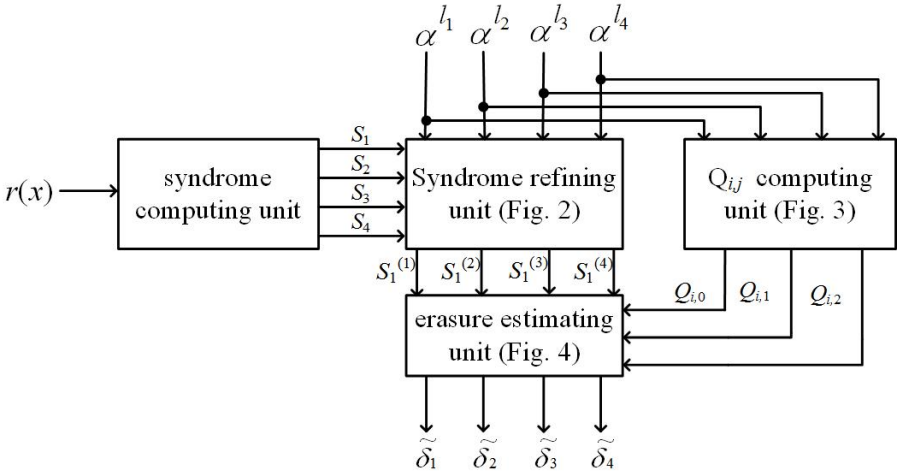

FIGURE 1. Hardware block diagram of the hardware decoder

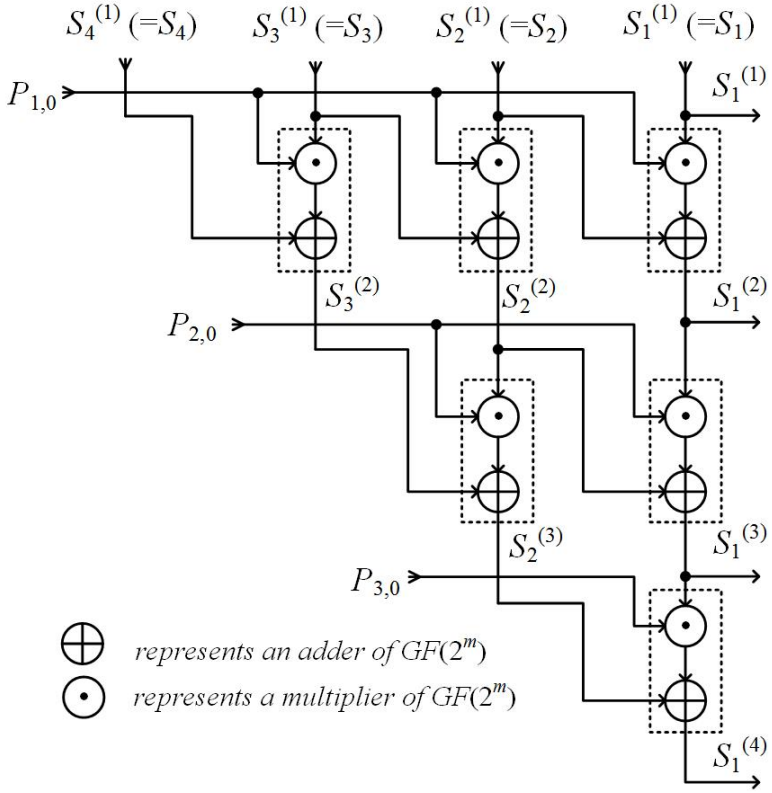

FIGURE 2. The circuit of syndrome refining unit

is used as an example to illustrate the circuits of these units, except for the syndrome computing unit.

Syndrome refining unit: For correcting four erasures, we need to refine S_1, S_2, S_3 and S_4 into $S_1^{(2)}$, $S_1^{(3)}$ and $S_1^{(4)}$. Thus, the unit consists of six cells, as shown in Figure 2, which computes $S_w^{(k)} = S_{w+1}^{(k-1)} + S_w^{(k-1)} P_{k-1,0} \left(= S_{w+1}^{(k-1)} + S_w^{(k-1)} \alpha^{l_{k-1}} \right)$ for $k = 2, 3, 4$ and $1 \leq w \leq 5 - k$.

$Q_{i,j}$ computing unit: According to (5), we can compute $Q_{4,1} = Q_{4,0}(P_{4,0} + P_{1,0})$, $Q_{4,2} = Q_{4,1}(P_{4,0} + P_{2,0})$ and $Q_{3,1} = Q_{3,0}(P_{3,0} + P_{1,0})$. Therefore, the $Q_{i,j}$ computing unit can be constructed by three cells, as shown in Figure 3.

Erasure estimating unit: Based on (8), the hardware circuit of the unit is designed and shown in Figure 4, which estimates erasures in the order of $\tilde{\delta}_4$, $\tilde{\delta}_3$, $\tilde{\delta}_2$ and $\tilde{\delta}_1$, recursively.

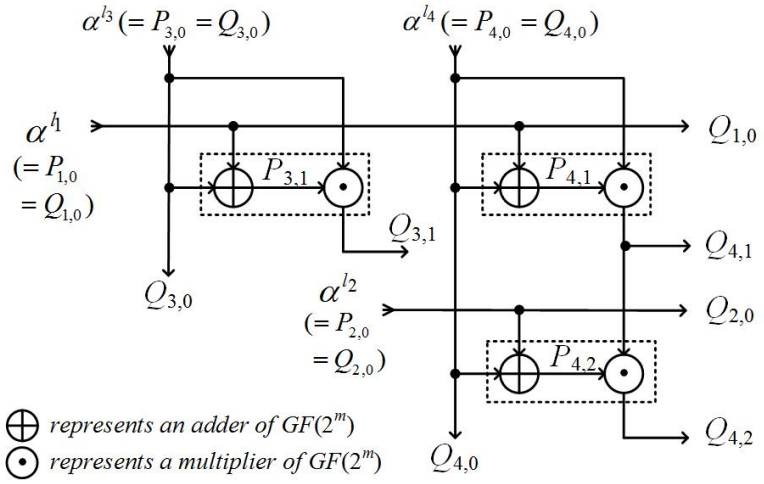

FIGURE 3. The circuit of $Q_{i,j}$ computing unit

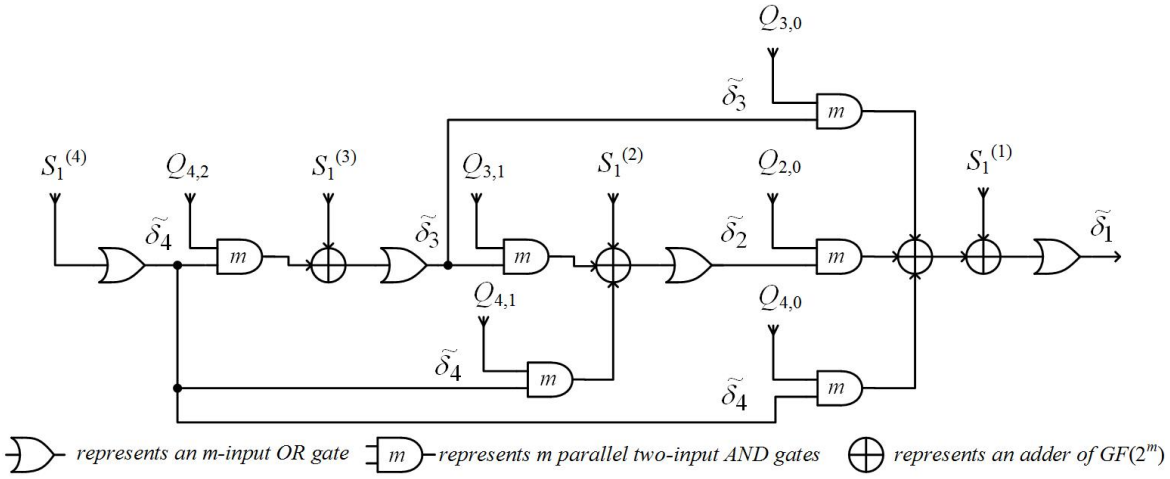

FIGURE 4. The circuit of erasure estimating unit

5. **Conclusion and Remarks.** The proposed algorithm can correct $d_{\min} - 1$ erasures without determining the ELP and finding the roots of ELP, such that its computational complexity is much lower than that of conventional decoding methods. Based on the decoding algorithm, a simple and regular hardware decoder was designed which is very well-suited for VLSI implementation. Moreover, the new decoding algorithm has an elegant property which can check whether the estimated result is correct or not, as follows. Compute $\tilde{S}_w = \tilde{\delta}_1 \alpha^{wl_1} + \tilde{\delta}_2 \alpha^{wl_2} + \cdots + \tilde{\delta}_v \alpha^{wl_v}$ for $w = 1, 3, \ldots, 2t-1$. If $\tilde{S}_w = S_w$, for $w = 1, 3, \ldots, 2t-1$, the estimated result is assumed to be correct; otherwise, the result must be wrong. A wrong result implies that there are some error bits which were not assigned as erasures. Thus, a decoding error alarm can be announced.

REFERENCES

[1] S. Lin and D. J. Costello, *Error Control Coding*, 2nd Edition, Prentice Hall, 2004.
[2] D. Chase, Class of algorithms for decoding block codes with channel measurement information, *IEEE Trans. Inf. Theory*, vol.18, pp.170-182, 1972.
[3] W. J. Reid III, L. L. Joiner and J. J. Komo, Soft decision decoding of BCH codes using error magnitudes, *IEEE Int. Symp. Inf. Theory*, p.303, 1997.
[4] J. J. Komo and L. L. Joiner, Fast error magnitude evaluations for Reed-Solomon codes, *IEEE Int. Symp. Inf. Theory*, p.416, 1995.

[5] R. Pyndiah, Near-optimun decoding of product codes: Block turbo codes, *IEEE Trans. Commun.*, vol.46, pp.1003-1010, 1998.
[6] A. Al-Dweik and B. S. Sharif, Non-sequential decoding algorithm for hard iterative turbo product codes, *IEEE Trans. Commun.*, vol.57, pp.1545-1549, 2009.
[7] A. Al-Dweik and B. S. Sharif, Closed-chains error correction technique for turbo product codes, *IEEE Trans. Commun.*, vol.59, pp.632-638, 2011.
[8] J. Andersen, Product codes for optical communications, *Proc. of ECOC2002*, vol.3, 2002.
[9] E. H. Lu, T. C. Chen and P. Y. Lu, A new method for evaluating error magnitudes of Reed-Solomon codes, *IEEE Commun. Lett.*, vol.18, pp.340-343, 2014.