PAPER

A Low-Complexity Step-by-Step Decoding Algorithm for Binary BCH Codes

Ching-Lung CHR[†a)], **Szu-Lin SU**[†], *Members*, *and* **Shao-Wei WU**[†], *Nonmember*

SUMMARY A low-complexity step-by-step decoding algorithm for t-error-correcting binary Bose-Chaudhuri-Hocquenghem (BCH) codes is proposed. Using logical analysis, we obtained a simple rule which can directly determine whether a bit in the received word is correct. The computational complexity of this decoder is less than the conventional step-by-step decoding algorithm, since it reduces at least half of the matrix computations and the most complex element in the conventional step-by-step decoder is the "matrix-computing" element.
key words: *BCH code, step-by-step decoding, matrix computation, computational complexity*

1. Introduction

The Bose-Chaudhuri-Hocquenghem (BCH) codes are a class of powerful multiple-error-correcting cyclic codes [1]–[3]. One popular error-correcting decoding procedure for binary BCH codes includes three major steps [1, 2, 4, 5]:

1) Calculate the syndrome values S_i, $i = 1, 2, \ldots, 2t$ from the received word.
2) Determine the error location polynomial $\sigma(x)$.
3) Find the roots of $\sigma(x)$, and then correct errors.

Massey first presented another well-known decoding method, the step-by-step decoding algorithm, for general BCH codes [6]. The step-by-step decoding algorithm [6]–[9] involves changing received symbols one at a time, checking whether the weight of the error pattern has been reduced. The common procedure of this method for decoding the binary BCH codes also consists of the following three steps:

a) Calculate the syndrome values S_i, $i = 1, 2, \ldots, 2t$ from the received word.
b) Temporarily change one received bit and then check whether the number of errors has been reduced. If so, the received bit is erroneous and shall be corrected.
c) Following the same procedure as step (b), check the received bits one by one.

The step-by-step decoding method avoids calculating the coefficients and searching for the roots of the error-location polynomial, so it is may be less complex than the standard algebraic method. The conventional step-by-step decoding algorithm has not been widely employed for BCH codes with large error-correcting capability owing to its requirement for calculation of the determinant of the syndrome matrix.

To achieve real-time decoding, [9] proposed a step-by-step decoder for t-error-correcting binary BCH codes in which a shift-syndrome generator is added and the matrix-calculation circuit is realized by systolic array so that the average computation time for the real-time decoder is only two logic-gate delays. This decoder has adopted the logic concept in the comparison of the number of errors. However, it has not tried to reduce the number of matrix-calculations.

This paper presents a modified step-by-step decoding algorithm for t-error-correcting binary BCH codes. By using logical analysis, the determination whether a received bit is erroneous in the step-by-step decoding algorithm as proposed in [9] can be further simplified into general functions. The new decoder requires only approximately half of the matrix calculations as the decoder in [9]. Thus, the computational complexity of this decoder is much less, since the most complex element of the step-by-step decoder is the "matrix-computing" element. Furthermore, the simple and regular decoding procedure also makes the decoder suitable in hardware realization.

The remainder of this paper is organized as follows. Section 2 describes the binary BCH codes and the step-by-step decoding algorithm proposed in [9]. Section 3 introduces a new step-by-step decoding algorithm and compares its calculation complexity with previous algorithm. Section 4 presents an example of the new decoder. Finally, Section 5 provides some concluding remarks.

2. Preliminaries

A t-error-correcting binary BCH code is capable of correcting any combination of t or fewer errors in a block of $n = 2^m - 1$ digits. For any positive integer m ($m \geq 3$) and t ($t < 2^{m-1}$), there exists a binary BCH code with the following parameters:

Block length: $n = 2^m - 1$
Number of information bits: $k \geq n - mt$
Minimum distance: $d_{\min} \geq 2t + 1$.

The generator polynomial of the code is specified in terms of its roots over the Galois field GF(2^m). Let α be a primitive element in GF(2^m). The generator polynomial $g(x)$ of the code is the lowest degree polynomial over GF(2), which has $\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{2t}$ as its roots. Let $\Phi_1(x)$, $\Phi_3(x), \ldots, \Phi_{2t-1}(x)$ be the distinct minimum polynomials of

Manuscript received April 23, 2004.
Manuscript revised August 9, 2004.
Final manuscript received October 7, 2004.
[†]The authors are with the Department of Electrical Engineering, National Cheng Kung University, Tainan 701, Taiwan, R.O.C.
a) E-mail: pipn@ms45.hinet.net

$\alpha, \alpha^3, \ldots, \alpha^{2t-1}$, respectively. Then, $g(x)$ is given by

$$g(x) = LCM\{\Phi_1(x), \Phi_3(x), \ldots, \Phi_{2t-1}(x)\}. \quad (1)$$

Let $v(x) = \sum_{i=0}^{n-1} v_i x^i$ be a systematic codeword and $e(x) = \sum_{i=0}^{n-1} e_i x^i$ be an error polynomial. Then, the received word can be expressed as

$$r(x) = \sum_{i=0}^{n-1} r_i x^i = v(x) + e(x). \quad (2)$$

The weight of the error pattern $e(x)$ is the number of errors in the received word $r(x)$. The corresponding syndromes can be calculated by

$$S_i = r(\alpha^i) = v(\alpha^i) + e(\alpha^i) = e(\alpha^i), i = 1, 2, \ldots, 2t. \quad (3)$$

For $1 \le v \le t$, the $v \times v$ syndrome matrix is defined as

$$\boldsymbol{M}_v = \begin{bmatrix} S_1 & 1 & 0 & \cdots & 0 \\ S_3 & S_2 & S_1 & \cdots & 0 \\ & \vdots & & & \vdots \\ S_{2v-1} & S_{2v-2} & S_{2v-3} & \cdots & S_v \end{bmatrix}.$$

The relations between the syndrome matrices and the number of errors in $r(x)$ can be found by theorem 9.11 in [1], or property 4′ in [6]. The theorem is restated below.

Theorem 1: For any binary BCH code and any v such that $1 \le v \le t$, the syndrome matrix $\boldsymbol{M}_v$ is singular if the number of errors is $v - 1$ or less, and is nonsingular if the number of errors is v or $v + 1$.

According to Theorem 1, the number of errors in $r(x)$ can be determined from the values of the determinants of the syndrome matrices $\det(\boldsymbol{M}_v)$, $v = 1, 2, \ldots, t$. For example, if $\det(\boldsymbol{M}_1) \neq 0$, $\det(\boldsymbol{M}_2) \neq 0$, $\det(\boldsymbol{M}_3) \neq 0$, and $\det(\boldsymbol{M}_v) = 0$, for $v = 4, 5, \ldots, t$, then three errors have occurred. Hence, to acquire the information of the number of errors, the step-by-step decoding algorithm is concerned whether the values of $\det(\boldsymbol{M}_v)$, $v = 1, 2, \ldots, t$, equal to zero, and a decision vector $\boldsymbol{m}$ composed of decision bits m_v, $v = 1, 2, \ldots, t$, is defined as [9]

$$\boldsymbol{m} = (m_1, m_2, \cdots, m_t),$$

where $m_v = 0$ if $\det(\boldsymbol{M}_v) = 0$ and $m_v = 1$ if $\det(\boldsymbol{M}_v) \neq 0$.

The decision vector of a general t-error-correcting binary BCH code can be expressed as follows:

1) $\boldsymbol{m} = (0^t)$ if no error has occurred, where 0^t denotes t consecutive identical 0 bits. For example, vector $(0^3) = (0, 0, 0)$.
2) $\boldsymbol{m} = (1, 0^{t-1})$ if one error has occurred.
3) $\boldsymbol{m} \in \{(\times^{u-2}, 1, 1, 0^{t-u})\}$ if u errors, $2 \le u < t$, have occurred, where the "×" means a value of either 0 or 1.
4) $\boldsymbol{m} \in \{(\times^{t-2}, 1, 1)\}$ if t errors have occurred.

For example, if $t = 2$, the decision vector can be (0, 0) for no error, (1, 0) for a single error, or (1, 1) for two errors.

Consequently, the number of errors can be correctly determined by the decision vector $\boldsymbol{m}$ if and only if the weight of error pattern is t or less. If the received word $r(x) = \sum_{i=0}^{n-1} r_i x^i$ is modified by changing temporarily a selected bit at position x^p, $0 \le p \le n - 1$, then the modified received word becomes

$$r^p(x) = r(x) + x^p = v(x) + e(x) + x^p = v(x) + e_{\bar{p}}(x), \quad (4)$$

where $e_{\bar{p}}(x) = e(x) + x^p$ and the subscript "$\bar{p}$" in $e_{\bar{p}}(x)$ indicates that the magnitude of the x^p position of $e(x)$ is temporarily changed.

Then, the modified syndrome matrix becomes

$$\boldsymbol{M}_{v,\bar{p}} = \begin{bmatrix} S_{1,\bar{p}} & 1 & 0 & \cdots & 0 \\ S_{3,\bar{p}} & S_{2,\bar{p}} & S_{1,\bar{p}} & \cdots & 0 \\ & \vdots & & & \vdots \\ S_{2v-1,\bar{p}} & S_{2v-2,\bar{p}} & S_{2v-3,\bar{p}} & \cdots & S_{v,\bar{p}} \end{bmatrix},$$

where $S_{i,\bar{p}} = e(\alpha^i) + (\alpha^i)^p = S_i + \alpha^{ip}$, $i = 1, 2, \ldots, 2t$.

The corresponding decision vector $\boldsymbol{m}_{\bar{p}}$ can also be defined as

$$\boldsymbol{m}_{\bar{p}} = (m_{1,\bar{p}}, m_{2,\bar{p}}, \cdots, m_{t,\bar{p}}),$$

where $m_{v,\bar{p}} = 0$ if $\det(\boldsymbol{M}_{v,\bar{p}}) = 0$ and $m_{v,\bar{p}} = 1$ if $\det(\boldsymbol{M}_{v,\bar{p}}) \neq 0$, $v = 1, 2, \ldots, t$.

Hence, $\boldsymbol{m}$ is the decision vector of original syndromes and $\boldsymbol{m}_{\bar{p}}$ is the decision vector of temporarily changed syndromes. Whether the bit at position x^p of $r(x)$ is erroneous can be determined from the difference between $\boldsymbol{m}$ and $\boldsymbol{m}_{\bar{p}}$. The step-by-step decoding algorithm proposed in [9] is then described as follows:

1) Determine the original syndromes and the decision vector $\boldsymbol{m} = (m_1, m_2, \ldots m_t)$.
2) Change the magnitude of the x^p position of $r(x)$ temporarily and determine the modified decision vector $\boldsymbol{m}_{\bar{p}} = (m_{1,\bar{p}}, m_{2,\bar{p}}, \cdots, m_{t,\bar{p}})$.
3) Using $\boldsymbol{m}$ and $\boldsymbol{m}_{\bar{p}}$, determine the value of $\hat{e}_p(t)$, which is the estimated value at the x^p position of the error pattern $e(x)$ by

$$\hat{e}_p(1) = m_1 \bar{m}_{1,\bar{p}}, \quad (5a)$$
$$\hat{e}_p(2) = (m_1 \bar{m}_{1,\bar{p}}) \bar{m}_2 \bar{m}_{2,\bar{p}} \vee m_1 m_{1,\bar{p}} m_2 \bar{m}_{2,\bar{p}}, \quad (5b)$$

and

$$\hat{e}_p(l) = [\hat{e}_p(l-1)) \bar{m}_l \bar{m}_{l,\bar{p}}] \vee m_{l-2,\bar{p}} m_{1-1} m_{l-1,\bar{p}} m_l \bar{m}_{l,\bar{p}}, \quad (5c)$$

$3 \le l \le t$, where t is the error-correcting capability of the binary BCH code, "∨" is the logical operator "OR," and $\bar{m}_i$ is the complement of m_i. (**Proof**: See the Appendix.)
4) Send the output bit $\hat{r}_p = r_p + \hat{e}_p(t)$.

3. Proposed Decoding Algorithm

A new step-by-step decoding algorithm for t-error-correcting binary BCH codes is proposed as follows. The algorithm follows the same idea as that proposed in [9], but further reduces the amount of the determinant-calculation of syndrome matrices by means of logical analysis. The logical analysis is based on the fact that only $2t+1$ cases are possible when we want to determine the value at the x^p position ($0 \le p \le n-1$) of the error pattern $\boldsymbol{e}$, $\boldsymbol{e} = (e_0, e_1, \ldots, e_{n-1})$, for a t-error-correcting binary BCH code. For example, for a binary (15, 7) BCH code ($t = 2$), all five possible cases of e_p are expressed in Table 1 and described as below:

1) If no error occurs, then the value of e_p must be correct (i.e. e_p =0).
2) If one error occurs, then the value of e_p can be erroneous or correct (i.e. $e_p = 1$ or $e_p = 0$).
3) If two errors occur, then the value of e_p can be erroneous or correct (i.e. $e_p = 1$ or $e_p = 0$).

In the following, we assume that the signal-to-noise ratio (SNR) of the communication is large enough such that the number of errors in a received codeword is t or fewer for a t-error-correcting binary BCH code.

Theorem 2: For a t-error-correcting binary BCH code, the estimated error value at position x^p of the error pattern, $\hat{e}_p(t)$, $n-k \le p \le n-1$, can be given by

$$\hat{e}_p(1) = \bar{m}_{1,\bar{p}}, \tag{6a}$$

$$\hat{e}_p(l) = \hat{e}_p(l-1)\bar{m}_l \vee m_l\bar{m}_{l,\bar{p}}, \quad \text{for } 2 \le l \le t. \tag{6b}$$

Proof:

In the case $t = 1$: Let $\boldsymbol{w(e)}$ be the weight of error pattern $\boldsymbol{e}$, and $e_p(1)$ be the value of position x^p in $\boldsymbol{e}$, $n-k \le p \le n-1$. Three possible cases apply in the determination of the value of $e_p(1)$, as shown in row 2 of Table 2. Row 4 is the decision bit m_1. If no error occurs [$\boldsymbol{w(e)} = 0$], $\det(\boldsymbol{M}_1) = 0$, then $m_1 = 0$. If one error occurs [$\boldsymbol{w(e)} = 1$], $\det(\boldsymbol{M}_1) \ne 0$, then $m_1 = 1$. $\boldsymbol{w(e_{\bar{p}})}$ and $m_{1,\bar{p}}$ indicates the weight of the error pattern and the decision bit for changing the received digit r_p, respectively. It is easy to see that $\hat{e}_p(1) = \bar{m}_{1,\bar{p}}$.

In the case $t \ge 2$: There are $2t+1$ possible cases in determining the value at position x^p [$e_p(t)$, $n-k \le p \le n-1$] of the error pattern $\boldsymbol{e}$, as shown in row 2 of Table 3. As the case of $t = 1$, the decision bits $m_1, m_2, \ldots, m_{t-1}, m_t, m_{1,\bar{p}}, m_{2,\bar{p}}, \ldots, m_{t-1,\bar{p}}$ and $m_{t,\bar{p}}$ in Table 3 can be determined by Theorem 1. All $2t+1$ possible estimated values of $e_p(t)$, as shown in the bottom row of Table 3, are equal to $\hat{e}_p(t-1)\bar{m}_t \vee m_t\bar{m}_{t,\bar{p}}$. Hence, in a similar way, we can get

Table 1 The possible various cases of e_p for $t = 2$.

w(e)	0	1		2	
e_p	0	1	0	1	0

Note: **w**(**e**) denotes the weight of error pattern **e**.

$$\begin{aligned}\hat{e}_p(l) &= \hat{e}_p(l-1)\bar{m}_l \vee m_l\bar{m}_{l,\bar{p}} \\ &= \{[(\bar{m}_{1,\bar{p}}\bar{m}_2) \vee m_2\bar{m}_{2,\bar{p}}]\cdots\}\bar{m}_l \\ &\quad \vee m_l\bar{m}_{l,\bar{p}}, \text{for } 2 \le l \le t.\end{aligned}$$

Q. E. D.

Equations (6a) and (6b) can be further simplified as shown in the following theorem.

Theorem 3: For a t-error-correcting binary BCH code, the estimated error value at position x^p of the error pattern, $\hat{e}_p(t)$, $n-k \le p \le n-1$, can be given by

$$\hat{e}_p(1) = \bar{m}_{1,\bar{p}}, \tag{7a}$$

$$\hat{e}_p(2) = m_1\bar{m}_{2,\bar{p}}, \tag{7b}$$

and

Table 2 The logic analysis in Theorem 2 for $t = 1$.

w(e)	0	1	
$e_p(1)$	0	1	0
$\mathbf{w(e_{\bar{p}})}$	1	0	2
m_1	0	1	1
$m_{1,\bar{p}}$	1	0	1
$\hat{e}_p(1) = \bar{m}_{1,\bar{p}}$	0	1	0

Table 3 The logic analysis in Theorem 2 for $t \ge 2$.

W(**e**)	0	1		2		⋯	t-1		t	
$e_p(t)$	0	1	0	1	0	⋯	1	0	1	0
$\mathrm{W}(\mathbf{e}_{\bar{p}})$	1	0	2	1	3	⋯	t-2	t	t-1	t+1
m_1	0	1	1	1	1	⋯	×	×	×	×
$m_{1,\bar{p}}$	1	0	1	1	×	⋯	×	×	×	×
$\hat{e}_p(1) = \bar{m}_{1,\bar{p}}$	0	1	0	0	×	⋯	×	×	×	×
m_2	0	0	0	1	1	⋯	×	×	×	×
$m_{2,\bar{p}}$	0	0	1	0	1	⋯	×	×	×	×
$\hat{e}_p(2) = \hat{e}_p(1)\bar{m}_2 \vee m_2\bar{m}_{2,\bar{p}}$	0	1	0	1	0	⋯	×	×	×	×
⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮	⋮	⋮
m_{t-1}	0	0	0	0	0	⋯	1	1	1	1
$m_{t-1,\bar{p}}$	0	0	0	0	0	⋯	0	1	1	×
$\hat{e}_p(t-1) = \hat{e}_p(t-2)\bar{m}_{t-1} \vee m_{t-1}\bar{m}_{t-1,\bar{p}}$	0	1	0	1	0	⋯	1	0	0	×
m_t	0	0	0	0	0	⋯	0	0	1	1
$m_{t,\bar{p}}$	0	0	0	0	0	⋯	0	1	0	1
$\hat{e}_p(t) = \hat{e}_p(t-1)\bar{m}_t \vee m_t\bar{m}_{t,\bar{p}}$	0	1	0	1	0	⋯	1	0	1	0

Note: The "×" denotes a value of either 0 or 1.

Table 4 The logic analysis in Theorem 3 for t is odd.

$W(\mathbf{e})$	0	1		2		3		⋯	t-3		t-2		t-1		t	
$e_p(t)$	0	1	0	1	0	1	0	⋯	1	0	1	0	1	0	1	0
$W(\mathbf{e}_{\bar{p}})$	1	0	2	1	3	2	4	⋯	t-4	t-2	t-3	t-1	t-2	t	t-1	t+1
$m_{1,\bar{p}}$	1	0	1	1	×	×	×	⋯	×	×	×	×	×	×	×	×
$\hat{e}_p(1)=\bar{m}_{1,\bar{p}}$	0	1	0	0	×	×	×	⋯	×	×	×	×	×	×	×	×
m_2	0	0	0	1	1	1	1	⋯	×	×	×	×	×	×	×	×
$m_{3,\bar{p}}$	0	0	0	0	1	0	1	⋯	×	×	×	×	×	×	×	×
$\hat{e}_p(3)=\hat{e}_1(1)\bar{m}_2 \vee m_2\bar{m}_{3,\bar{p}}$	0	1	0	1	0	1	0	⋯	×	×	×	×	×	×	×	×
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
m_{t-3}	0	0	0	0	0	0	0	⋯	1	1	1	1	×	×	×	×
$m_{t-2,\bar{p}}$	0	0	0	0	0	0	0	⋯	0	1	0	1	1	×	1	×
$\hat{e}_p(t-2)=\hat{e}_p(t-4)\bar{m}_{t-3} \vee m_{t-3}\bar{m}_{t-2,\bar{p}}$	0	1	0	1	0	1	0	⋯	1	0	1	0	×	×	×	×
m_{t-1}	0	0	0	0	0	0	0	⋯	0	0	0	0	1	1	1	1
$m_{t,\bar{p}}$	0	0	0	0	0	0	0	⋯	0	0	0	0	0	1	0	1
$\hat{e}_p(t)=\hat{e}_p(t-2)\bar{m}_{t-1} \vee m_{t-1}\bar{m}_{t,\bar{p}}$	0	1	0	1	0	1	0	⋯	1	0	1	0	1	0	1	0

Table 5 The logic analysis in Theorem 3 for t is even.

$W(\mathbf{e})$	0	1		2		⋯	t-3		t-2		t-1		t	
$e_p(t)$	0	1	0	1	0	⋯	1	0	1	0	1	0	1	0
$W(\mathbf{e}_{\bar{p}})$	1	0	2	1	3	⋯	t-4	t-2	t-3	t-1	t	t-2	t-1	t+1
m_1	0	1	1	1	1	⋯	×	×	×	×	×	×	×	×
$m_{2,\bar{p}}$	0	0	1	0	1	⋯	×	×	×	×	×	×	×	×
$\hat{e}_p(2)=m_1\bar{m}_{2,\bar{p}}$	0	1	0	1	0	⋯	×	×	×	×	×	×	×	×
⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
m_{t-3}	0	0	0	0	0	⋯	1	1	1	1	×	×	×	×
$m_{t-2,\bar{p}}$	0	0	0	0	0	⋯	0	1	0	1	1	0	1	×
$\hat{e}_p(t-2)=\hat{e}_p(t-4)\bar{m}_{t-3} \vee m_{t-3}\bar{m}_{t-2,\bar{p}}$	0	1	0	1	0	⋯	1	0	1	0	×	×	×	×
m_{t-1}	0	0	0	0	0	⋯	0	0	0	0	1	1	1	1
$m_{t,\bar{p}}$	0	0	0	0	0	⋯	0	0	0	0	0	1	0	1
$\hat{e}_p(t)=\hat{e}_p(t-2)\bar{m}_{t-1} \vee m_{t-1}\bar{m}_{t,\bar{p}}$	0	1	0	1	0	⋯	1	0	1	0	1	0	1	0

$$\hat{e}_p(l) = \hat{e}_p(l-2)\bar{m}_{l-1} \vee m_{l-1}\bar{m}_{l,\bar{p}}, \quad \text{for } 3 \le l \le t. \quad (7c)$$

The proof of Theorem 3 is similar to that of Theorem 2 and its analysis refers to Table 4 (for t is odd) and Table 5 (for t is even). For example, if $t = 3$, then $\hat{e}_p(3) = \hat{e}_p(1)\bar{m}_2 \vee m_2\bar{m}_{3,\bar{p}}$, and if t= 4, then $\hat{e}_p(4) = \hat{e}_p(2)\bar{m}_3 \vee m_3\bar{m}_{4,\bar{p}}$.

According to Theorem 3, the procedure of the proposed step-by-step decoding algorithm for a t-error-correcting binary BCH code can be summarized as follows:

Table 6 The equations and matrix-calculations of the error correctors for t-error-correcting binary BCH decoders, t = 1, 2, 3, 4, 5; $n-k \le p \le n-1$.

		Conventional step-by-step Algorithm in [9].	Proposed Algorithm in Theorem 3.
t=1	$\hat{e}_p(1)$	$m_1\bar{m}_{1,\bar{p}}$	$\bar{m}_{1,\bar{p}}$
	matrix calculations	$\det(M_1)$	
		$\det(M_{1,\bar{p}})$	$\det(M_{1,\bar{p}})$
t=2	$\hat{e}_p(2)$	$(m_1\bar{m}_{1,\bar{p}})\bar{m}_2\bar{m}_{2,\bar{p}} \vee m_1m_{1,\bar{p}}m_2\bar{m}_{2,\bar{p}}$	$m_1\bar{m}_{2,\bar{p}}$
	matrix calculations	$\det(M_1)$, $\det(M_2)$.	$\det(M_1)$
		$\det(M_{1,\bar{p}})$, $\det(M_{2,\bar{p}})$.	$\det(M_{2,\bar{p}})$
t=3	$\hat{e}_p(3)$	$[(m_1\bar{m}_{1,\bar{p}})\bar{m}_2\bar{m}_{2,\bar{p}} \vee m_1m_{1,\bar{p}}m_2\bar{m}_{2,\bar{p}}]\bar{m}_3\bar{m}_{3,\bar{p}} \vee m_{1,\bar{p}}m_2m_{2,\bar{p}}m_3\bar{m}_{3,\bar{p}}$	$\bar{m}_{1,\bar{p}}\bar{m}_2 \vee m_2\bar{m}_{3,\bar{p}}$
	matrix calculations	$\det(M_1)$, $\det(M_2)$, $\det(M_3)$.	$\det(M_2)$
		$\det(M_{1,\bar{p}})$, $\det(M_{2,\bar{p}})$, $\det(M_{3,\bar{p}})$.	$\det(M_{1,\bar{p}})$ $\det(M_{3,\bar{p}})$
t=4	$\hat{e}_p(4)$	$\{[(m_1\bar{m}_{1,\bar{p}})\bar{m}_2\bar{m}_{2,\bar{p}} \vee m_1m_{1,\bar{p}}m_2\bar{m}_{2,\bar{p}}]\bar{m}_3\bar{m}_{3,\bar{p}} \vee m_{1,\bar{p}}m_2m_{2,\bar{p}}m_3\bar{m}_{3,\bar{p}}\}\bar{m}_4\bar{m}_{4,\bar{p}} \vee m_{2,\bar{p}}m_3m_{3,\bar{p}}m_4\bar{m}_{4,\bar{p}}$	$(m_1\bar{m}_{2,\bar{p}})\bar{m}_3 \vee m_3\bar{m}_{4,\bar{p}}$
	matrix calculations	$\det(M_1)$, $\det(M_2)$, $\det(M_3)$, $\det(M_4)$.	$\det(M_1)$ $\det(M_3)$
		$\det(M_{1,\bar{p}})$, $\det(M_{2,\bar{p}})$, $\det(M_{3,\bar{p}})$, $\det(M_{4,\bar{p}})$.	$\det(M_{2,\bar{p}})$ $\det(M_{4,\bar{p}})$
t=5	$\hat{e}_p(5)$	$\{\{[(m_1\bar{m}_{1,\bar{p}})\bar{m}_2\bar{m}_{2,\bar{p}} \vee m_1m_{1,\bar{p}}m_2\bar{m}_{2,\bar{p}}]\bar{m}_3\bar{m}_{3,\bar{p}} \vee m_{1,\bar{p}}m_2m_{2,\bar{p}}m_3\bar{m}_{3,\bar{p}}\}\bar{m}_4\bar{m}_{4,\bar{p}} \vee m_{2,\bar{p}}m_3m_{3,\bar{p}}m_4\bar{m}_{4,\bar{p}}\}\bar{m}_5\bar{m}_{5,\bar{p}} \vee m_{3,\bar{p}}m_4m_{4,\bar{p}}m_5\bar{m}_{5,\bar{p}}$	$(\bar{m}_{1,\bar{p}}\bar{m}_2 \vee m_2\bar{m}_{3,\bar{p}})\bar{m}_4 \vee m_4\bar{m}_{5,\bar{p}}$
	matrix calculations	$\det(M_1)$, $\det(M_2)$, $\det(M_3)$, $\det(M_4)$, $\det(M_{5,\bar{p}})$.	$\det(M_2)$ $\det(M_4)$
		$\det(M_{1,\bar{p}})$, $\det(M_{2,\bar{p}})$, $\det(M_{3,\bar{p}})$, $\det(M_{4,\bar{p}})$, $\det(M_{5,\bar{p}})$.	$\det(M_{1,\bar{p}})$ $\det(M_{3,\bar{p}})$ $\det(M_{5,\bar{p}})$

1) Calculate the original syndromes S_i (i = 1, 2, 3,. . . , $2t$).
2) Determine initial decision vector $\boldsymbol{m} = (m_2, m_4, \cdots, m_{t-1})$ if t is odd, or $\boldsymbol{m} = (m_1, m_3, \cdots, m_{t-1})$ for t is even.
3) Let $p = n - 1$.
4) Change the magnitude of the x^p position of $r(x)$ temporarily and determine the modified decision vector $\boldsymbol{m}_{\bar{p}} = (m_{1,\bar{p}}, m_{3,\bar{p}}, \cdots, m_{t,\bar{p}})$ if t is odd, or $\boldsymbol{m}_{\bar{p}} = (m_{2,\bar{p}}, m_{4,\bar{p}}, \cdots, m_{t,\bar{p}})$ for t is even.

5) Using $\boldsymbol{m}$ and $\boldsymbol{m}_{\bar{p}}$, determine the value of $\hat{e}_p(t)$, which is the estimated value at the x^p position of the error pattern $e(x)$ by Eqs. (7a), (7b), and (7c).
6) Send the output bit $\hat{r}_p = r_p + \hat{e}_p(t)$.
7) Let $p = p - 1$. If $p = n - k - 1$, then this decoding algorithm is completed. Otherwise, go to step 4.

Table 6 compares the equations and the matrix calculations required in the proposed decoders with those required in [9]. Obviously, the proposed algorithm reduces the number of matrix computations by half at least.

4. Illustrative Example

Consider the (31, 11) binary BCH code with error correcting capability $t = 5$ as an example. Let α denote a primitive element of $GF\ (2^5)$ and $\alpha^5 + \alpha^2 + 1 = 0$. The generator polynomial of the (31,11) binary BCH code is defined as the least common multiple of the minimal polynomials of $1, \alpha, \alpha^2, \ldots,$ and α^{10}. Assume there are three errors in the received polynomial $r(x)$ and $e(x) = x^7 + x^{20} + x^{25}$.

The initial syndrome values are

$$S_1 = 1, S_2 = 1, S_3 = \alpha^8, S_4 = 1, S_5 = \alpha^{19}, S_6 = \alpha^{16},$$
$$S_7 = \alpha^7.$$

Since

$$\det(\boldsymbol{M}_4) = \begin{vmatrix} S_1 & 1 & 0 & 0 \\ S_3 & S_2 & S_1 & 1 \\ S_5 & S_4 & S_3 & S_2 \\ S_7 & S_6 & S_5 & S_4 \end{vmatrix} = 0 \quad \text{and}$$

$$\det(\boldsymbol{M}_2) = \begin{vmatrix} S_1 & 1 \\ S_3 & S_2 \end{vmatrix} = \alpha^{20} \neq 0,$$

so $m_4 = 0$ and $m_2 = 1$.

From Eq. (7), the estimated error value of $e_p(5)$ is

$$\begin{aligned} \hat{e}_p(5) &= (\bar{m}_{1,\bar{p}}\bar{m}_2 \vee m_2\bar{m}_{3,\bar{p}})\bar{m}_4 \vee m_4\bar{m}_{5,\bar{p}} \\ &= (\bar{m}_{1,\bar{p}} \cdot 0 \vee 1 \cdot \bar{m}_{3,\bar{p}}) \cdot 1 \vee 0 \cdot \bar{m}_{5,\bar{p}} \\ &= \bar{m}_{3,\bar{p}} \end{aligned} \tag{8}$$

For $20 \leq p \leq 30$ the temporarily changed syndrome values are

$$S_{i,\bar{p}} = S_i + \alpha^{ip}, \quad i = 1, 2, 3, 4, 5.$$

Then,

$$\begin{aligned}
&S_{1,\overline{30}} = \alpha^{17}, S_{1,\overline{29}} = \alpha^3, S_{1,\overline{28}} = \alpha^{26}, S_{1,\overline{27}} = 1,\\
&S_{1,\overline{26}} = \alpha^{28}, S_{1,\overline{25}} = \alpha^{21}, S_{1,\overline{24}} = \alpha^{15}, S_{1,\overline{23}} = \alpha^{12},\\
&S_{1,\overline{22}} = \alpha^7, S_{1,\overline{21}} = \alpha^{25}, S_{1,\overline{20}} = \alpha^8;\\
&S_{2,\overline{30}} = \alpha^3, S_{2,\overline{29}} = \alpha^6, S_{2,\overline{28}} = \alpha^{21}, S_{2,\overline{27}} = 1,\\
&S_{2,\overline{26}} = \alpha^{25}, S_{1,\overline{25}} = \alpha^{21}, S_{1,\overline{24}} = \alpha^{15}, S_{2,\overline{23}} = \alpha^{24},\\
&S_{2,\overline{22}} = \alpha^{14}, S_{2,\overline{21}} = \alpha^{19}, S_{2,\overline{20}} = \alpha^{16};\\
&S_{3,\overline{30}} = \alpha^{16}, S_{3,\overline{29}} = \alpha^7, S_{3,\overline{28}} = \alpha^{21}, S_{3,\overline{27}} = \alpha^{27},\\
&S_{3,\overline{26}} = \alpha^{28}, S_{3,\overline{25}} = \alpha^4, S_{3,\overline{24}} = \alpha^6, S_{3,\overline{23}} = \alpha^{25},\\
&S_{3,\overline{22}} = \alpha^{14}, S_{3,\overline{21}} = \alpha^{23}, S_{3,\overline{20}} = \alpha^2;\\
&S_{4,\overline{30}} = \alpha^6, S_{4,\overline{29}} = \alpha^{12}, S_{4,\overline{28}} = \alpha^{11}, S_{4,\overline{27}} = 1,\\
&S_{4,\overline{26}} = \alpha^{19}, S_{4,\overline{25}} = \alpha^{22}, S_{4,\overline{24}} = \alpha^{29}, S_{4,\overline{23}} = \alpha^{17},\\
&S_{4,\overline{22}} = \alpha^{28}, S_{4,\overline{21}} = \alpha^{27}, S_{4,\overline{20}} = \alpha^1;\\
&S_{5,\overline{30}} = \alpha^{10}, S_{5,\overline{29}} = \alpha^{24}, S_{5,\overline{28}} = \alpha^{24}, S_{5,\overline{27}} = 1,\\
&S_{5,\overline{26}} = \alpha^{20}, S_{5,\overline{25}} = \alpha^2, S_{5,\overline{24}} = \alpha^{19}, S_{5,\overline{23}} = \alpha^{17},\\
&S_{5,\overline{22}} = \alpha^{22}, S_{5,\overline{21}} = \alpha^3, S_{5,\overline{20}} = \alpha^{30}.
\end{aligned}$$

By calculating the value of

$$\det(\boldsymbol{M}_{3,\bar{p}}) = \begin{vmatrix} S_{1,\bar{p}} & 1 & 0 \\ S_{3,\bar{p}} & S_{2,\bar{p}} & S_{1,\bar{p}} \\ S_{5,\bar{p}} & S_{4,\bar{p}} & S_{3,\bar{p}} \end{vmatrix},$$

we can determine

$$\begin{aligned}
&m_{3,\overline{30}} = 1, m_{3,\overline{29}} = 1, m_{3,\overline{28}} = 1, m_{3,\overline{27}} = 1, m_{3,\overline{26}} = 1,\\
&m_{3,\overline{25}} = 0, m_{3,\overline{24}} = 1, m_{3,\overline{23}} = 1, m_{3,\overline{22}} = 1, m_{3,\overline{21}} = 1,\\
&m_{3,\overline{20}} = 0.
\end{aligned}$$

Hence,

$$\begin{aligned}
&\hat{e}_{30}(5) = \bar{m}_{3,\overline{30}} = 0, \quad \hat{e}_{29}(5) = \bar{m}_{3,\overline{29}} = 0,\\
&\hat{e}_{28}(5) = \bar{m}_{3,\overline{28}} = 0, \quad \hat{e}_{27}(5) = \bar{m}_{3,\overline{27}} = 0,\\
&\hat{e}_{26}(5) = \bar{m}_{3,\overline{26}} = 0, \quad \hat{e}_{25}(5) = \bar{m}_{3,\overline{25}} = 1,\\
&\hat{e}_{24}(5) = \bar{m}_{3,\overline{24}} = 0, \quad \hat{e}_{23}(5) = \bar{m}_{3,\overline{23}} = 0,\\
&\hat{e}_{22}(5) = \bar{m}_{3,\overline{22}} = 0, \quad \hat{e}_{21}(5) = \bar{m}_{3,\overline{21}} = 0,\\
&\text{and } \hat{e}_{20}(5) = \bar{m}_{3,\overline{20}} = 1.
\end{aligned}$$

Therefore, the estimated error pattern of the message part of the received vector is

$$\begin{aligned}
&(\hat{e}_{20}, \hat{e}_{21}, \hat{e}_{22}, \hat{e}_{23}, \hat{e}_{24}, \hat{e}_{25}, \hat{e}_{26}, \hat{e}_{27}, \hat{e}_{28}, \hat{e}_{29}, \hat{e}_{30})\\
&\quad = (1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0).
\end{aligned}$$

5. Conclusions

A novel step-by-step decoding algorithm for t-error-correcting primitive binary BCH codes is proposed in this paper. A simple equation to estimate the error value of the error pattern is obtained by logical analysis. The computational complexity of this decoder is much lower than of the step-by-step decoding algorithm proposed in [9], since the most complex elements in the step-by-step decoder are the "matrix-computing" elements and the proposed algorithm at least reduces half of the matrix computations. Furthermore, as [9] the simple structure also makes it suitable for hardware realization.

References

[1] W.W. Peterson and E.J. Weldon, Error-Correcting Codes, MIT Press, Cambridge, MA, 1972.
[2] E.R. Berlekamp, Algebraic Coding Theory, McGraw-Hill, New York, NY, 1968.

[3] S. Lin and D.J. Costello, Error Control Coding: Fundamentals and Applications. Prentice-Hall, Englewood Cliffs, NJ, 1983.
[4] F.J. MacWilliams and N.J.A. Sloane, The Theory of Error-Correcting Codes, North-Holland, New York, NY, 1977.
[5] W.W. Peterson, "Encoding and error-correcting procedures for the Bose-Chaudhuri codes," IRE Trans. Inf. Theory, vol.IT-6, pp.459–470, Sept. 1960.
[6] J.L. Massey, "Step-by-step decoding of the Bose-Chaudhuri- Hocquenghem codes," IEEE Trans. Inf. Theory, vol.IT-11, no.4, pp.580–585, 1965.
[7] Z. Szwaja, "On step-by-step decoding of the BCH binary code," IEEE Trans. Inf. Theory, vol.IT-13, pp.350–351, 1967.
[8] S.W. Wei and C.H. Wei, "High-speed hardware decoder for double-error-correcting binary BCH codes," IEE Proc., vol.136, no.3, pp.227–231, 1989.
[9] S.W. Wei and C.H. Wei, "A high-speed real-time binary BCH decoder," IEEE Trans. Circuits Syst. Video Technol., vol.3, no.2, pp.138–147, 1993.

Appendix: Proof of Eqs. (5a), (5b), and (5c)

Let $\boldsymbol{w}(\boldsymbol{e})$ be the weight of error pattern $\boldsymbol{e}$, $\hat{e}_p(t)$ be the estimated value of position x^p in $\boldsymbol{e}$, $n-k \leq p \leq n-1$. $\boldsymbol{m} = (m_1, m_2, \cdots, m_t)$ is the decision vector of original syndromes and $\boldsymbol{m}_{\bar{p}} = (m_{1,\bar{p}}, m_{2,\bar{p}}, \cdots, m_{t,\bar{p}})$ is the decision vector of temporarily changed syndromes.

In the case $t = 1$: Three possible cases apply in the determination of the value of $e_p(1)$, as shown in column 2 of Table A· 1. The analysis of Table A· 1 as follows:

1) If no error occurs [$\boldsymbol{w}(\boldsymbol{e}) = 0$, $\boldsymbol{m} = (m_1) = (0)$], then the value of e_p must be correct (i.e. $e_p = 0$). Temporarily change a received bit at position x^p, $n-k \leq p \leq n-1$, then one error occurs [$\boldsymbol{w}(\boldsymbol{e}_{\bar{p}}) = 1$, $\boldsymbol{m}_{\bar{p}} = (m_{1,\bar{p}}) = (1)$].
2) If one error occurs [$\boldsymbol{w}(\boldsymbol{e}) = 1$, $\boldsymbol{m} = (m_1) = (1)$], then the value of e_p can be erroneous or correct (i.e. $e_p = 1$ or $e_p = 0$). Temporarily change a received bit at position x^p, then no or two errors occur [$\boldsymbol{w}(\boldsymbol{e}_{\bar{p}}) = 0$ or 2, $\boldsymbol{m}_{\bar{p}} = (m_{1,\bar{p}}) = (0)$ or (1)]. It is easy to see that the estimated error value of $e_p(1)$ is

$$\hat{e}_p(1) = m_1 \bar{m}_{1,\bar{p}}.$$

In the case $t = 2$: Five possible cases apply in the determination of the value of $e_p(2)$, as shown in column 2 of Table A· 2. As the case of $t = 1$, the decision bits m_1, m_2, $m_{1,\bar{p}}$ and $m_{2,\bar{p}}$ in Table A· 2 can be determined. Then, the estimated error value of $e_p(2)$ is

$$\begin{aligned}\hat{e}_p(2) &= m_1\bar{m}_2\bar{m}_{1,\bar{p}}\bar{m}_{2,\bar{p}} \vee m_1 m_2 m_{1,\bar{p}}\bar{m}_{2,\bar{p}} \\ &= (m_1\bar{m}_{1,\bar{p}})\bar{m}_2\bar{m}_{2,\bar{p}} \vee m_1 m_2 m_{1,\bar{p}}\bar{m}_{2,\bar{p}}\end{aligned}$$

$\{= \overline{(m_2 \oplus m_{1,\bar{p}})} m_1 \bar{m}_{2,\bar{p}}$, Fig. 13 of [9].
In [9], define that
$m_v = 1$ if $\det(M_v) = 0$, $v = 1, 2, \cdots, t.\}$

In the case $t \geq 3$:
For the case $t = 3$: Seven possible cases apply in the determination of the value of $e_p(2)$, as shown in column 2 of

Table A· 1 The logic analysis of Eq. (5a) for $t = 1$.

w(e)	$e_p(1)$	$\mathbf{w}(\mathbf{e}_{\bar{p}})$	m_1	$m_{1,\bar{p}}$
0	0	1	0	1
1	1	0	1	0
	0	2	1	1

Table A· 2 The logic analysis of Eq. (5b) for $t = 2$.

w(e)	$e_p(2)$	$\mathbf{w}(\mathbf{e}_{\bar{p}})$	m_1	m_2	$m_{1,\bar{p}}$	$m_{2,\bar{p}}$
0	0	1	0	0	1	0
1	1	0	1	0	0	0
	0	2	1	0	1	1
2	1	1	1	1	1	0
	0	3	1	1	×	1

Note: The "×" denotes a value of either 0 or 1.

Table A· 3 The logic analysis of Eq. (5c) for $t = 3$.

w(e)	$e_p(3)$	$\mathbf{w}(\mathbf{e}_{\bar{p}})$	m_1	m_2	m_3	$m_{1,\bar{p}}$	$m_{2,\bar{p}}$	$m_{3,\bar{p}}$
0	0	1	0	0	0	1	0	0
1	1	0	1	0	0	0	0	0
	0	2	1	0	0	1	1	0
2	1	1	1	1	0	1	0	0
	0	3	1	1	0	×	1	1
3	1	2	×	1	1	1	1	0
	0	4	×	1	1	×	×	1

Table A· 4 The logic analysis of Eq. (5c) for $t > 3$.

w(e)	$e_p(t)$	$\mathbf{w}(\mathbf{e}_{\bar{p}})$	m_1	m_2	m_3	$\cdots$	m_{t-2}	m_{t-1}	m_t	$m_{1,\bar{p}}$	$m_{2,\bar{p}}$	$m_{3,\bar{p}}$	$\cdots$	$m_{t-2,\bar{p}}$	$m_{t-1,\bar{p}}$	$m_{t,\bar{p}}$
0	0	1	0	0	0	$\cdots$	0	0	0	1	0	0	$\cdots$	0	0	0
1	1	0	1	0	0	$\cdots$	0	0	0	0	0	0	$\cdots$	0	0	0
	0	2	1	0	0	$\cdots$	0	0	0	1	1	0	$\cdots$	0	0	0
2	1	1	1	1	0	$\cdots$	0	0	0	1	0	0	$\cdots$	0	0	0
	0	3	1	1	0	$\cdots$	0	0	0	1	1	0	$\cdots$	0	0	0
3	1	2	×	1	1	$\cdots$	0	0	0	1	1	0	$\cdots$	0	0	0
	0	4	×	1	1	$\cdots$	0	0	0	×	×	1	$\cdots$	0	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
t-1	1	t-2	×	×	×	$\cdots$	1	1	0	×	×	×	$\cdots$	1	0	0
	0	t	×	×	×	$\cdots$	1	1	0	×	×	×	$\cdots$	×	1	1
t	1	t-1	×	×	×	$\cdots$	×	1	1	×	×	×	$\cdots$	1	1	0
	0	t+1	×	×	×	$\cdots$	×	1	1	×	×	×	$\cdots$	×	×	1

Table A·3. As the case of $t = 1$, the decision bits m_1, m_2, m_3, $m_{1,\bar{p}}$, $m_{2,\bar{p}}$ and $m_{3,\bar{p}}$ in Table A·3 can be determined. Then, the estimated error value of $e_p(3)$ is

$$\begin{aligned}\hat{e}_p(3) &= m_1\bar{m}_2\bar{m}_3\bar{m}_{1,\bar{p}}\bar{m}_{2,\bar{p}}\bar{m}_{3,\bar{p}} \vee m_1 m_2 \bar{m}_3 m_{1,\bar{p}}\bar{m}_{2,\bar{p}}\bar{m}_{3,\bar{p}} \\ &\quad \vee m_2 m_3 m_{1,\bar{p}} m_{2,\bar{p}} \bar{m}_{3,\bar{p}} \\ &= [(m_1\bar{m}_{1,\bar{p}})\bar{m}_2\bar{m}_{2,\bar{p}} \vee m_1 m_2 m_{1,\bar{p}}\bar{m}_{2,\bar{p}}]\bar{m}_3\bar{m}_{3,\bar{p}} \vee \\ &\quad m_2 m_3 m_{1,\bar{p}} m_{2,\bar{p}}\bar{m}_{3,\bar{p}} \\ &= \hat{e}_p(2)\bar{m}_3\bar{m}_{3,\bar{p}} \vee m_2 m_3 m_{1,\bar{p}} m_{2,\bar{p}}\bar{m}_{3,\bar{p}}\end{aligned}$$

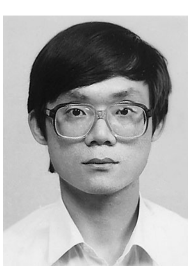
$\{= \overline{(m_2 \oplus m_{1,\bar{p}})} m_1 \bar{m}_3 \bar{m}_{2,\bar{p}} \bar{m}_{3,\bar{p}} \vee m_2 m_3 m_{1,\bar{p}} m_{2,\bar{p}} \bar{m}_{3,\bar{p}}$,
Fig. 14 of [9].$\}$

For the case $t > 3$: All $2t + 1$ possible values of e_p(t), as shown in the column 2 of Table A·4. As the case of $t = 1$, the decision bits $m_1, m_2, \ldots, m_{t-1}, m_t, m_{1,\bar{p}}, m_{2,\bar{p}}, \ldots, m_{t-1,\bar{p}}$ and $m_{t,\bar{p}}$ in Table A·4 can be determined. Hence, in a similar way, we can get

$$\hat{e}_p(t) = \hat{e}_p(t-1)\bar{m}_t\bar{m}_{t,\bar{p}} \vee m_{t-1} m_t m_{t-2,\bar{p}} m_{t-1,\bar{p}} \bar{m}_{t,\bar{p}}$$

Q.E.D.

Ching-Lung Chr was born in Chayi, Taiwan, R.O.C., in 1965. He received the B.S. and M.S. degrees from Chung-Cheng Institute of Technology, Taiwan, in 1988 and 1996, respectively. In 2000, he is the Ph.D. candidate of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan. His research interest is channel coding techniques.



Szu-Lin Su received the B.S. and M.S. degrees from National Taiwan University, Taiwan, R.O.C., in 1977 and 1979 and the Ph.D. degree from the University of Southern California, Los Angeles, in 1985, all in electrical engineering. From 1979 to 1989, he was a Research Member of Chung Shan Institute of Science and Technology, Taiwan, working on the design of digital communication and network systems. Since 1989, he has been with National Cheng Kung University, Tainan, Taiwan, where he is currently a Professor of electrical engineering. His research interests are in the areas of wireless communications, mobile communication networks, satellite communications, and channel coding techniques.

Shao-Wei Wu was born in Taipei, Taiwan, R.O.C., in 1966. He received the B.S., M.S. and Ph.D. degrees from Chung-Cheng Institute of Technology, Taiwan, in 1988, 1995 and 2002, respectively. His research interests are channel coding techniques and cryptography.