Shift-Register Synthesis and BCH Decoding

JAMES L. MASSEY, MEMBER, IEEE

Abstract—**It is shown in this paper that the iterative algorithm introduced by Berlekamp for decoding BCH codes actually provides a general solution to the problem of synthesizing the shortest linear feedback shift register capable of generating a prescribed finite sequence of digits. The shift-register approach leads to a simple proof of the validity of the algorithm as well as providing additional insight into its properties. The equivalence of the decoding problem for BCH codes to a shift-register synthesis problem is demonstrated, and other applications for the algorithm are suggested.**

I. Introduction

In the following section, the problem of finding the shortest linear feedback shift register that can generate a given finite sequence of digits is studied. In Section III, an algorithm is developed that yields a simple recursive solution for this problem by synthesizing for $n = 1, 2, \cdots$ the shortest register that can generate the first n digits of this sequence. Sections IV and V provide a review of certain properties of shift-register sequences and of Bose–Chaudhuri–Hocquenghem (BCH) codes, and culminate in a demonstration that the major decoding problem for BCH codes is a shift-register synthesis problem of the type above. The shift-register synthesis algorithm of Section III is then seen to coincide with the iterative algorithm introduced recently by Berlekamp [1] for decoding the BCH codes. Finally, some additional applications for the algorithm are suggested.

II. Length Properties of LFSR's

A general *linear feedback shift register* (LFSR) of length L is shown in Fig. 1 and consists of a cascade of L unit delay cells, or stages, with provision to form a linear combination of the cell contents, which then serves as the input to the first stage. The output of the LFSR is assumed to be taken from the last stage. The initial contents $s_0, s_1, \cdots, s_{L-1}$ of the L stages coincide with the first L output digits, and the remaining output digits are uniquely determined by the recurson

$$s_j = -\sum_{i=1}^{L} c_i s_{j-i}, \qquad j = L, L+1, L+2, \cdots. \tag{1}$$

The output digits and the feedback coefficients $c_1, c_2, \cdots, c_L$ are assumed to lie in the same field F, which can be either a finite field $GF(q)$, or an infinite field, such as the real number field. There is no requirement that $c_L \neq 0$ (i.e., the last stage of the LFSR need not be tapped).

An LFSR is said to *generate* a finite sequence $s_0, s_1, \cdots, s_{N-1}$ when this sequence coincides with the first N output digits of the LFSR for some initial loading. If $L \geqq N$, the LFSR always generates the sequence. If $L < N$, it follows from (1) that the LFSR generates the sequence if and only if

$$s_j + \sum_{i=1}^{L} c_i s_{j-i} = 0, \qquad j = L, L+1, \cdots, N-1. \tag{2}$$

The following simple theorem will play a key role in the subsequent development

Theorem 1

If some LFSR of length L generates the sequence $s_0, s_1, \cdots, s_{N-1}$ but not the sequence $s_0, s_1, \cdots, s_{N-1}, s_N$, then any LFSR that generates the latter sequence has length L', satisfying

$$L' \geqq N + 1 - L. \tag{3}$$

Proof: For $L \geqq N$, the theorem is trivially true so we may suppose that $L < N$. Let $c_1, c_2, \cdots, c_L$ and $c_1', c_2', \cdots, c_{L'}'$ denote the connection coefficients of the two LFSR's in question and assume that $L' \leqq N - L$, in violation of (3). By hypothesis

$$-\sum_{i=1}^{L} c_i s_{j-i} \begin{array}{ll} = s_j, & j = L, L+1, \cdots, N-1 \\ \neq s_N, & j = N, \end{array} \tag{4}$$

and

$$-\sum_{k=1}^{L'} c_k' s_{j-k} = s_j, \qquad j = L', L'+1, \cdots, N. \tag{5}$$

Therefore, it follows that

$$-\sum_{i=1}^{L} c_i s_{N-i} = +\sum_{i=1}^{L} c_i \sum_{k=1}^{L'} c_k' s_{N-i-k} \tag{6}$$

where the use of (5) in rewriting the left-hand side of (6) is justified by the fact that $\{s_{N-L}, s_{N-L+1}, \cdots, s_{N-1}\}$ is a subset of $\{s_{L'}, s_{L'+1}, \cdots, s_{N-1}\}$. Upon interchange of the order of summation, (6) becomes

$$\begin{aligned} -\sum_{i=1}^{L} c_i s_{N-i} &= +\sum_{k=1}^{L'} c_k' \sum_{i=1}^{L} c_i s_{N-k-i} \\ &= -\sum_{k=1}^{L'} c_k' s_{N-k} \\ &= s_N \end{aligned} \tag{7}$$

where use has been made of (4) and (5), respectively. The use of (4) is justified by the fact that $\{s_{N-L'}, s_{N-L'+1}, \cdots, s_{N-1}\}$ is a subset of $\{s_L, s_{L+1}, \cdots, s_{N-1}\}$. But (7)

Manuscript received March 7, 1967; revised June 24, 1968. This work was supported in part by the National Aeronautics and Space Administration (Grant NsG-334 at the M. I. T. Research Laboratory of Electronics and Grant NGR 15-004-026 at the University of Notre Dame, Notre Dame, Ind.) and by the Joint Services Electronics Program, Contract DA 208-043-AMC-02536)E at Massachusetts Institute of Technology, Cambridge.

The author is with the University of Notre Dame, Notre Dame, Ind. 46556

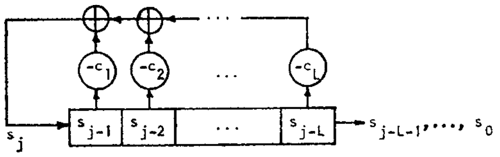


Fig. 1. General L-stage linear feedback shift-register (LFSR).

contradicts (4) proving that the assumption $L' \leqq N - L$ is untenable. We conclude that $L' \geqq N + 1 - L$ as was to be shown.

Now let $\mathbf{s}$ denote an infinite sequence $s_0, s_1, s_2, \cdots$ so that $s_0, s_1, \cdots, s_{N-1}$ forms the first N digits of $\mathbf{s}$. We define $L_N(\mathbf{s})$ as the minimum of the lengths of all the LFSR's that generate $s_0, s_1, \cdots, s_{N-1}$. By our earlier remarks, $L_N(\mathbf{s}) \leqq N$. Moreover, $L_N(\mathbf{s})$ must be monotonically nondecreasing with increasing N. By way of convention, we shall say that the all-zero sequence is generated by the LFSR with length $L = 0$, and therefore that $L_N(\mathbf{s}) = 0$ if and only if $s_0, s_1, \cdots, s_{N-1}$ are all zeros.

Lemma 1

If some LFSR of length $L_N(\mathbf{s})$ generates $s_0, s_1, \cdots, s_{N-1}$, but not $s_0, s_1, \cdots, s_{N-1}, s_N$, then

$$L_{N+1}(\mathbf{s}) \geqq \max\,[L_N(\mathbf{s}), N + 1 - L_N(\mathbf{s})].$$

Proof: From the monotonicity of $L_N(\mathbf{s})$, we have $L_{N+1}(\mathbf{s}) \geqq L_N(\mathbf{s})$. Under the hypothesis of the lemma, Theorem 1 implies that $L_{N+1}(\mathbf{s}) \geqq N + 1 - L_N(\mathbf{s})$. Therefore the lemma follows.

Lemma 1 will be used in the next section to demonstrate the minimality of the length of a shift register found by a synthesis algorithm for LFSR's. A consequence of the resulting development will be a proof that the inequality in Lemma 1 can be replaced by an equality.

III. The LFSR Synthesis Algorithm

In this section, a recursive algorithm is derived for producing one of the LFSR's of length $L_N(\mathbf{s})$, which generates $s_0, s_1, \cdots, s_{N-1}$ for $N = 1, 2, 3, \cdots$. The discussion will be facilitated by defining the *connection polynomial* of the LFSR of Fig. 1 as the polynomial

$$C(D) = 1 + c_1 D + c_2 D^2 + \cdots + c_L D^L \qquad (8)$$

which has degree at most L in the indeterminate D. By way of convention, we take $C(D) = 1$ for the LFSR of length $L = 0$.

When $s_0, s_1, \cdots, s_{N-1}$ are all zeros but $s_N \neq 0$, then $L_{N+1}(\mathbf{s}) = N + 1$ since any shorter LFSR must be initially loaded with all zeros and thus could generate only further zeros. Moreover, any LFSR with $L = N + 1$ suffices to generate $s_0, s_1, \cdots, s_{N-1}, s_N$ in this case. Note further that Lemma 1 holds with equality in this circumstance.

For a given $\mathbf{s}$, let

$$C^{(N)}(D) = 1 + c_1^{(N)} D + \cdots + c_{L_N(\mathbf{s})}^{(N)} D^{L_N(\mathbf{s})} \qquad (9)$$

denote the connection polynomial of a minimal-length $L_N(\mathbf{s})$ LFSR that generates $s_0, s_1, \cdots, s_{N-1}$. As an inductive hypothesis, assume that $L_N(\mathbf{s})$ and some $C^{(N)}(D)$ have been found for $N = 1, 2, \cdots, n$ with equality obtaining in Lemma 1 for $N = 1, 2, \cdots, n - 1$. We seek then to find $L_{n+1}(\mathbf{s})$ and some $C^{(n+1)}(D)$, and to show that equality obtains in Lemma 1 for $N = n$.

By the induction hypothesis, we have from (2) that

$$s_j + \sum_{i=1}^{L_n(\mathbf{s})} c_i^{(n)} s_{j-i} = \begin{cases} 0, & j = L_n(\mathbf{s}), \cdots, n-1 \\ d_n & j = n, \end{cases} \qquad (10)$$

where d_n, which we call the *next discrepancy*, is the difference between s_n and the $(n + 1)-st$ digit generated by the minimal-length LFSR, which we have found to generate the first n digits of $\mathbf{s}$. If $d_n = 0$, then this LFSR also generates the first $n + 1$ digits of $\mathbf{s}$ so that $L_{n+1}(\mathbf{s}) = L_n(\mathbf{s})$, and we may now take $C^{(n+1)}(D) = C^{(n)}(D)$.

If $d_n \neq 0$, a new LFSR must be found to generate the first $n + 1$ digits of $\mathbf{s}$. In this latter case, let m be the sequence length before the last *length change* in the minimal-length registers, i.e.,

$$\begin{aligned} L_m(\mathbf{s}) &< L_n(\mathbf{s}) \\ L_{m+1}(\mathbf{s}) &= L_n(\mathbf{s}). \end{aligned} \qquad (11)$$

Since a length change was required, the LFSR with connection polynomial $C^{(m)}(D)$ and length $L_m(\mathbf{s})$ could not have generated $s_0, s_1, \cdots, s_{m-1}, s_m$. Therefore, from (2) we have

$$s_j + \sum_{i=1}^{L_m(\mathbf{s})} c_i^{(m)} s_{j-i} = \begin{cases} 0, & j = L_m(\mathbf{s}), \cdots, m-1 \\ d_m \neq 0, & j = m. \end{cases} \qquad (12)$$

By the induction hypothesis, Lemma 1 holds with equality for $N = m$ so that

$$L_{m+1}(\mathbf{s}) = L_n(\mathbf{s}) = \max\,[L_m(\mathbf{s}), m + 1 - L_m(\mathbf{s})]$$

and in particular, because of (11), this gives

$$L_n(\mathbf{s}) = m + 1 - L_m(\mathbf{s}). \qquad (13)$$

We now claim that the connection polynomial

$$C(D) = C^{(n)}(D) - d_n d_m^{-1} D^{n-m} C^{(m)}(D) \qquad (14)$$

is a valid next choice for $C^{(n+1)}(D)$. Note first from (14) that the degree of $C(D)$ is at most

$$\max\,[L_n(\mathbf{s}), n - m + L_m(\mathbf{s})] = \max\,[L_n(\mathbf{s}), n + 1 - L_n(\mathbf{s})]$$

where the equality follows from (13). Hence $C(D)$ is an allowable connection polynomial for a LFSR of length L where

$$L = \max\,[L_n(\mathbf{s}), n + 1 - L_n(\mathbf{s})]. \qquad (15)$$

Moreover, it follows from (14) that

$$\begin{aligned} s_j + \sum_{i=1}^{L} c_i s_{j-i} &= s_j + \sum_{i=1}^{L_n(\mathbf{s})} c_i^{(n)} s_{j-i} - d_n d_m^{-1} \\ &\quad \cdot \left[s_{j-n+m} + \sum_{i=1}^{L_m(\mathbf{s})} c_i^{(m)} s_{j-n+m-i} \right] \\ &= \begin{cases} 0 & j = L, L+1, \cdots, n-1 \\ d_n - d_n d_m^{-1} d_m = 0, & j = n \end{cases} \end{aligned}$$

where the last equalities result from the use of (10) and (12). Therefore, it follows from (2) that the LFSR of length L with connection polynomial $C(D)$ generates the $n + 1$ digits $s_0, s_1, \cdots, s_n$. Since L in (15) satisfies Lemma 1 with equality, we conclude that $L = L_n(\mathrm{s})$, and therefore that equality in Lemma 1 is always obtained. Thus we have proved Theorem 2.

Theorem 2

If some LFSR of length $L_N(\mathrm{s})$, which generates $s_0, s_1, \cdots, s_{N-1}$, also generates $s_0, s_1, \cdots, s_{N-1}, s_N$, then $L_{N+1}(\mathrm{s}) = L_N(\mathrm{s})$. Conversely, if some LFSR of length $L_N(\mathrm{s})$ that generates $s_0, s_1, \cdots, s_{N-1}$ fails to generate $s_0, s_1, \cdots, s_{N-1}, s_N$, then $L_{N+1}(\mathrm{s}) = \max\,[L_N(\mathrm{s}), N + 1 - L_N(\mathrm{s})]$.

Moreover, our proof of Theorem 2 was a constructive proof, which establishes the validity of the following algorithm for synthesizing a shortest LFSR to generate the sequence $s_0, s_1, \cdots, s_{n-1}$.

LFSR Synthesis Algorithm (Berlekamp Iterative Algorithm):

```
1) $1 \to C(D)$    $1 \to B(D)$    $1 \to x$
   $0 \to L$       $1 \to b$       $0 \to N$
2) If $N = n$, stop. Otherwise compute
        $d = s_N + \sum_{i=1}^{L} c_i s_{N-i}$.
3) If $d = 0$, then $x + 1 \to x$, and go to 6).
4) If $d \neq 0$ and $2L > N$, then
   $C(D) - d\, b^{-1} D^x B(D) \to C(D)$
   $x + 1 \to x$
   and go to 6).
5) If $d \neq 0$ and 2L $\leqq N$, then
   $C(D) \to T(D)$ [temporary storage of $C(D)$]
   $C(D) - d\, b^{-1} D^x B(D) \to C(D)$
   $N + 1 - L \to L$
   $T(D) \to B(D)$
   $d \to b$
   $1 \to x$.
6) N + 1 $\to N$ and return to 2).
```

For every n, when $N = n$ and step 2) has just been reached, then the quantities produced by the algorithm bear the following relations to the quantities appearing in the development preceding Theorem 2:

$$\begin{aligned} C(D) &= C^{(n)}(D) \\ L &= L_n(\mathrm{s}) \\ x &= n - m \\ d &= d_n \text{ (assuming the} \\ &\quad \text{computation in step 2) is performed)} \\ B(D) &= C^{(m)}(D) \\ b &= d_m. \end{aligned}$$

That the algorithm implements the procedure derived preceding Theorem 2 should be evident except for the following two points. First, step 5) is carried out only when, according to Theorem 2, a length change is needed. In this case, the present $C(D)$ for subsequent iterations will be the last connection polynomial before the latest length change and therefore becomes the new $B(D) = C^{(m)}(D)$. Second, suppose that the first nonzero d occurs in step 2) with $N = k$. This implies $s_0 = s_1 = \cdots = s_{k-1} = 0$ and $s_k \neq 0$. At this time, $L = L_k(\mathrm{s}) = 0$ and, therefore, the sequence length before the last length change is undefined, since no LFSR can have length less than zero. Thus the rule of (14) for computing the next connection polynomial is not applicable. However, in this case, the initialization in step 1) has the effect of causing step 5) to be applied, which then results in $C(D) = C^{(k+1)}(D) = 1 - dD^{k+1}$ and $L = L_{k+1}(\mathrm{s}) = k + 1$. We have already pointed out that any length $k + 1$ LFSR is a valid solution for this case.

In Fig. 2 the results are shown for the application of the algorithm to the binary $[F = GF(2)]$ sequence $s_0, s_1, \cdots, s_4 = 1, 0, 1, 0, 0$. Note that the resulting LFSR is singular (i.e., $c_3 = 0$) and the last stage is not tapped.

A logical circuit for implementing the algorithm is shown in Fig. 3 and is seen to require $3L_o + 1$ memory cells, where each cell can store a digit in the field F, and where L_o is the maximum length of an LFSR that can be produced with this circuitry.

Up to this point we have considered only the problem of finding one of the minimal-length registers that generate a specified sequence, but the set of *all* minimal-length $L_n(\mathrm{s})$ LSFR's that generate $s_0, s_1, \cdots, s_{n-1}$ can also readily be found from the LFSR synthesis algorithm. From Theorem 2, we observe that when some LFSR of length $L_N(\mathrm{s})$ that generates $s_0, s_1, \cdots, s_{N-1}$ fails to generate $s_0, s_1, \cdots, s_{N-1}, s_N$, there will then be a length change $[L_{N+1}(\mathrm{s}) > L_N(\mathrm{s})]$ if and only if $2L_N(\mathrm{s}) \leqq N$. It follows that the minimal-length LFSR is *unique* if and only if $2L_N(\mathrm{s}) \leqq N$. Therefore, when the algorithm terminates with $2L > n$, the resulting minimal-length LFSR is not unique. In this case, however, the resulting LFSR would be the unique solution if the additional digits $s_n, s_{n+1}, \cdots, s_{2L-1}$ were to be specified in agreement with the output sequence of this LFSR. Moreover, for any assignment of these $2L - n$ additional digits, only steps 3) or 4) of the algorithm would be used to produce new connection polynomials, i.e., the pattern of the $2L - n$ next discrepancies d serve only to determine a polynomial multiple of the unchanging $B(D)$, which will be added to produce the final $C(D)$, and some choice of this pattern must result in every possible LFSR of length $L_n(\mathrm{s})$ that generates $s_0, s_1, \cdots, s_{n-1}$. These remarks are summarized in the following theorem.

Theorem 3

Suppose the LFSR synthesis algorithm is applied to the sequence $s_0, s_1, \cdots, s_{n-1}$ and let L, $C(D)$, x, and $B(D)$ denote the values when the algorithm terminates. If $2L \leqq n$, then $C(D)$ is the connection polynomial of the unique minimal-length L LFSR that generates the sequence. If $2L > n$, then the set of polynomials

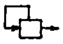
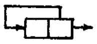
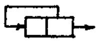
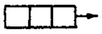
N	L	C(D)	LFSR	x	B(D)	b	s_N	d
0	0	1		1	1	1	1	1
1	1	1 + D		1	1	1	0	1
2	1	1		2	1	1	1	1
3	2	$1 + D^2$		1	1	1	0	0
4	2	$1 + D^2$		2	1	1	0	1
5	3	1		1	$1 + D^2$	1		

Fig. 2. Example of application of the LFSR synthesis algorithm to the binary sequence $s_0, s_1, s_2, s_3, s_4 = 1, 0, 1, 0, 0$.

$$\{C(D) + Q(D)\, D^x B(D):$$

$$\text{degree of } Q(D) \text{ less than } 2L - n\},$$

is the set of connection polynomials for all of the minimal-length-L LFSR's that generate the sequence.

For instance, in the example shown in Fig. 2, Theorem 3 gives the allowable $Q(D)$ to be either 0 or 1. Hence the set of connection polynomials $\{1, 1 + D + D^3\}$ specifies both $L = 3$ LFSR's that generate the given $n = 5$ sequence. The following is an immediate consequence of Theorem 3

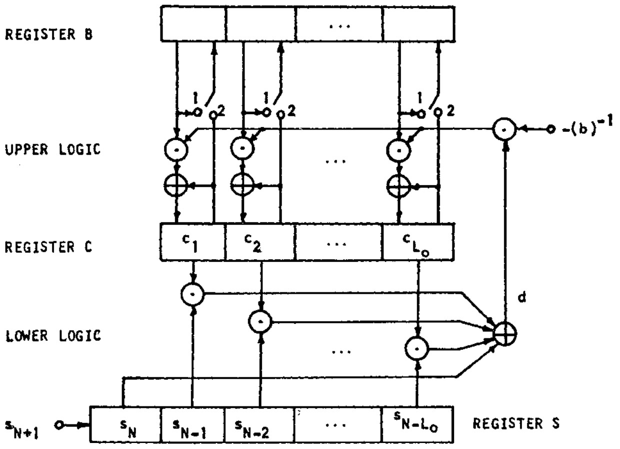
Corollary

If $2L_n(\mathrm{s}) < n$, then the LFSR synthesis algorithm will already have produced the unique minimal-length solution, i.e., $L = L_n(\mathrm{s})$ and $C(D) = C^{(n)}(D)$, when $N = 2L_n(\mathrm{s})$ in (2), i.e., after only the first $2L_n(\mathrm{s})$ digits have been processed by the algorithm.

For instance, if the sequence $s_0, s_1, \cdots, s_{n-1}$ is a nonzero cycle of length $n = 2^{100}$-1 from a 100 stage maximal-length LFSR, then the algorithm has necessarily found the unique generating LFSR after the first $2L = 200$ digits have been processed.

The LFSR synthesis algorithm given in this section is (practically) identical to the iterative algorithm developed by Berlekamp [1] for decoding the BCH codes, as will be seen in Section V. It should be noted that when $2L = N + 1$ and $d \neq 0$, it is then permissible to modify step 4) of the algorithm so that $B(D)$ is replaced by the old $C(D)$. The reason for this is that it can be shown that rather than taking $C^{(m)}(D)$ as the last connection polynomial before a length change, it suffices more generally to choose $C^{(m)}(D)$ as any of the previous connection polynomials for which $d_m \neq 0$ and $m - L_m(\mathrm{s})$ is maximized. When $d_n \neq 0$ and $2L_n(\mathrm{s}) = n + 1$, then $n - L_n(\mathrm{s}) = m - L_m(\mathrm{s})$ so that $C^{(n)}(D)$ is an allowable replacement for $C^{(m)}(D)$. Berlekamp's algorithm contains an additional test for deciding whether to replace $C^{(m)}(D)$ in this case, but there seems to be no advantage deriving from it so that we have excluded such a test from the LFSR synthesis algorithm.

(NOTE: REGISTER B WILL CONTAIN COEFFICIENTS OF B(D) SHIFTED x - 1 POSITIONS TO RIGHT)



RULES OF OPERATION: ACTIVATE LOWER LOGIC.
IF d = 0, SHIFT B AND S REGISTERS ONE POSITION.
IF d ≠ 0 AND 2L > N, MOVE SWITCHES TO POLE 1 AND ACTIVATE UPPER LOGIC, THEN SHIFT B AND S REGISTERS ONE POSITION.
IF d ≠ 0 AND 2L ≤ N, MOVE SWITCHES TO POLE 2 AND ACTIVATE UPPER LOGIC, REPLACE b BY d AND REPLACE L BY N + 1 - L, THEN SHIFT B AND S REGISTERS ONE POSITION AND LOAD A 1 INTO THE FIRST STAGE OF REGISTER B.

Fig. 3. A logical circuit for implementing the LFSR synthesis algorithm.

IV. Classical Description of LFSR Sequences

In this section, we review some properties of LFSR-generated sequences with a view toward applying this material to BCH codes in the sequel.

It will prove convenient to describe the sequence $\mathrm{s} = s_0, s_1, \cdots$ by its Huffman D-transform

$$S(D) = s_0 + s_1 D + s_2 D^2 + \cdots. \tag{16}$$

From (8) and (16), we see that (2) simply specifies that the degree j term in the product $C(D)\, S(D)$ vanishes for $j = L, L + 1, L + 2, \cdots$. Hence, (2) may be rewritten as

$$C(D)\, S(D) = P(D)$$

or

$$S(D) = \frac{P(D)}{C(D)} \tag{17}$$

where

$$P(D) = p_0 + p_1 D + \cdots + p_{L-1} D^{L-1} \tag{18}$$

is a polynomial of degree less than L. Moreover, from (17) and (18), we find the matrix equation

$$\begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_{L-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ c_1 & 1 & \cdots & 0 & 0 \\ & & \vdots & & \\ c_{L-1} & c_{L-2} & \cdots & c_1 & 1 \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_{L-1} \end{bmatrix} \tag{19}$$

which relates the coefficients of $P(D)$ to the connection coefficients and the initial contents of the LFSR. Since the matrix in (19) is nonsingular, it follows that for every

$P(D)$ as in (18) there will be a unique corresponding assignment of initial conditions. We may summarize in Theorem 4.

Theorem 4

The output sequences generated by an L-stage LFSR with connection polynomial $C(D)$ is the set $\{\mathbf{s}\}$ of sequences corresponding to the set of transforms

$$\left\{S(D) = \frac{P(D)}{C(D)}, \text{ degree of } P(D) \text{ less than } L\right\}.$$

Theorem 4 shows that $\mathbf{s}$ is an output sequence of some LFSR if and only if its transform $S(D)$ is a rational function, i.e., a ratio of polynomials $A(D)/B(D)$, with $B(0) \neq 0$. Moreover, if $A(D)$ and $B(D)$ are relatively prime polynomials (i.e., have no common factor of degree one or greater), then it follows directly from Theorem 4 that $B(D)$, within a constant factor required to make $B(0) = 1$, is the unique connection polynomial of the shortest LFSR that generates $\mathbf{s}$, and the length of this LFSR is the maximum of the degree of $B(D)$ and the degree of $A(D)$ plus one. Restating these remarks, we have the following.

Corollary

If $S(D) = P(D)/C(D)$ where $P(D)$ and $C(D)$ are relatively prime polynomials and $C(0) = 1$, then $C(D)$ is the connection polynomial of the shortest LFSR that generates the sequence $\mathbf{s}$ whose transform is $S(D)$, and

$$L_n(\mathbf{s}) = \max\,[\text{degree of } C(D), 1 + \text{degree of } P(D)]. \quad (20)$$

V. Application to Decoding of the BCH Codes

Let $g(X) = g_0 + g_1X + \cdots + g_{r-1}X^{r-1} + X^r$, $g_0 \neq 0$, be a monic polynomial of degree r, $r \geqq 1$, with coefficients in some finite field $GF(q)$. Let n be the least integer such that $g(X)$ divides $X^n - 1$. With every n-tuple $\mathbf{f} = [f_0, f_1, \cdots, f_{n-1}]$ of elements of $GF(q)$, associate the polynomial $f(X) = f_0 + f_1X + \cdots + f_{n-1}X^{n-1}$ of degree less than n. Then the *cyclic code* generated by $g(X)$ is the set of n-tuples $\mathbf{f}$ such that $g(X)$ divides $f(X)$. The length is n digits and the code redundancy is r digits.

A Bose-Chaudhuri-Hocquenghem (BCH) code [2] is a cyclic code where $g(X)$ is chosen to be the minimum-degree monic polynomial with coefficients in $GF(q)$ having $\alpha^{m_o}, \alpha^{m_o+1}, \cdots, \alpha^{m_o+d-2}$ as roots where α is a specified nonzero element of $GF(q^m)$, m_o is some positive integer, and d, $d \geqq 2$, is any integer such that the $d - 1$ specified roots of $g(X)$ are all distinct. We shall call such a code a BCH (α, q, m_o, d) code when we wish to specify the main parameters. It is well known that such a BCH code has minimum distance at least d, and d is sometimes called the *design distance* of the code.

If a codeword $\mathbf{f}$ in a BCH (α, q, m_o, d) code is transmitted, and an n-tuple $\mathbf{r} = [r_0, r_1, \cdots, r_{n-1}]$ of elements from $GF(q)$ is received, then $\mathbf{e} = [e_0, e_1, \cdots, e_{n-1}] = \mathbf{r} - \mathbf{f}$ is called the *error pattern*. Associating polynomials with $\mathbf{e}$ and $\mathbf{r}$ as was done with $\mathbf{f}$, we have

$$r(X) = f(X) + e(X). \quad (21)$$

With the error polynomial $e(X)$, one associates the weighted power sum symmetric functions $S_1, S_2, \cdots$ defined by

$$S_i = e(\alpha^i), \qquad i = 1, 2, 3, \cdots. \quad (22)$$

Since $g(X)$ divides $f(X)$, all roots of $g(X)$ are also roots of $f(X)$ so that from (21) and (22) it follows that

$$S_i = r(\alpha^i),$$
$$i = m_o, m_o + 1, \cdots, m_o + d - 2, \quad (23)$$

and hence that this set of $d - 1$ consecutive S can be formed *at the receiver*. This can be accomplished with simple logical circuitry [2]. The BCH decoding problem simply stated is the following. Given the $d - 1$ consecutive S_i defined in (23), find the error pattern $e(X)$.

Let t be the Hamming weight of the error pattern $\mathbf{e}$, i.e., the number of nonzero components. If the jth nonzero component in $\mathbf{e}$ is the digit e_k, then $X_j = \alpha^k$ is called the *locator* of this error and $Y_j = e_k$ is the *error magnitude.* X_j is an element of $GF(q^m)$ and Y_j is an element of $GF(q)$. From (22), it follows that

$$S_i = \sum_{j=1}^{t} Y_jX_j^i, \qquad i = 1, 2, 3, \cdots. \quad (24)$$

For binary codes ($q = 2$), the error locators completely describe the error pattern since $Y_j = 1$, $j = 1, 2, \cdots, t$. For general q, Forney [3] has given a simple procedure for determining the error magnitudes given the error locators. Therefore the essential BCH decoding problem reduces to the following. Given S_{m_o}, $S_{m_o} + 1, \cdots, S_{m_o} + d - 2$, find the error locators $X_1, X_2, \cdots, X_t$.

Following Berlekamp [1], we first observe that

$$\frac{1}{1 - X_jD} = 1 + X_jD + X_j^2D^2 + \cdots. \quad (25)$$

Multiplying by $Y_jX_j^{m_o}$ in (25) and summing, we obtain with the aid of (24)

$$\sum_{j=1}^{t} \frac{Y_jX_j^{m_o}}{1 - DX_j} = S_{m_o} + S_{m_o+1}D + S_{m_o+2}D^2 + \cdots. \quad (26)$$

The left-hand side of (26) is recognized to be the partial fraction expansion of $P(D)/C(D)$, where

$$C(D) = \prod_{j=1}^{t} (1 - X_jD) \quad (27)$$

and

$$P(D) = \sum_{j=1}^{t} Y_jX_j^{m_o} \prod_{\substack{k=1 \\ k\neq j}}^{t} (1 - X_kD). \quad (28)$$

Therefore, we may write

$$\frac{P(D)}{C(D)} = S_{m_o} + S_{m_o+1}D + S_{m_o+2}D^2 + \cdots \quad (29)$$

where $C(0) = 1$ and where $P(D)$ and $C(D)$ are relatively prime polynomials. This latter property follows from the fact that if $P(D)$ and $C(D)$ had any common factors of degree at least one, then the partial fraction expansion of their ratio must have fewer nonzero terms than the degree of $C(D)$ contrary to (26). From (27) and (28), we see that the degree of $C(D)$ is exactly t, while the degree of $P(D)$ is less than t. From (29) and the corollary of Theorem 4, Theorem 5 then follows.

Theorem 5

The polynomial $C(D)$ defined by (27) is the connection polynomial of the unique shortest LFSR over $F = GF(q^m)$ that generates the sequence $S_{m_o}, S_{m_o} + 1, S_{m_o} + 2, \cdots$.

From (27), it follows that the t roots of $C(D)$ are the reciprocals of the t error locators. Chien [4] has given a simple means for implementing the task of finding the roots from $C(D)$ so that the essential decoding problem for the BCH codes reduces finally to the following. Given $S_{m_o}, S_{m_o} + 1, \cdots, S_{m_o} + d - 2$, find the polynomial $C(D)$ in (27). From Theorem 5 and the corollary of Theorem 3, it follows that the LFSR synthesis algorithm may be used to solve this decoding problem when the error pattern has weight guaranteed correctable by the design distance of the code. We state this fact as the following corollary.

Corollary

When the weight t of the error pattern $\mathbf{e}$ satisfies $2t \leqq d - 1$, then $C(D)$ defined by (27) is the connection polynomial of the unique shortest LFSR over $GF(q^m)$ that generates the sequence $S_{m_o}, S_{m_o} + 1, \cdots, S_{m_o} + d - 2$ and therefore will be produced when the LFSR synthesis algorithm is applied to this $n = d - 1$ digit sequence.

The determination of $C(D)$ from the sequence given in this corollary is precisely the function of the interative algorithm developed by Berlekamp [1]. In fact, the LFSR synthesis algorithm of Section III is (except for the minor variation noted earlier) precisely the Berlekamp algorithm abstracted from its particular application to the decoding of the BCH codes.

The reader is referred to Berlekamp [1] for 1) a discussion of the simplification that occurs when the algorithm is used with binary BCH codes, namely $d = 0$ automatically in step 2) when N is odd, 2) applicability of the algorithm to errors-and-erasures decoding, and 3) modifications by which the algorithm can be extended to correct some errors of weight t with $2t > d - 1$, essentially by postulating additional S_i, $i > m_o + d - 1$, at the receiver.

VI. Additional Applications

There appears to be a number of interesting applications for the LFSR synthesis algorithm of Section III. The most obvious is that of finding a simple digital device to generate a prescribed binary sequence with useful properties in some application. Less obviously, the algorithm might be used as part of a source coder, or data compressor, for a binary data source whose output contains considerable redundancy. For instance, the source digits might be processed by the algorithm in blocks of 127 digits. Each block could then be represented for transmission as a 7-bit block giving the length L of the shortest LFSR that generates the original sequence, followed by L bits to indicate the values of the tap connections and a further L bits giving the initial contents of the LFSR. Therefore, a total of $2L + 7$ bits would be transmitted in place of the original 127 bits. Such a data compression scheme could be expected to perform efficiently only when the underlying constraints producing the source redundancy were with high probability linear relations among the binary source digits.

VII. Remarks

It should be pointed out that although the $\{c_i\}$ and $\{s_i\}$ considered in Sections II and III were assumed to lie in a field F, the proofs of Theorem 1 and Lemma 1 made no use of the existence of a multiplicative inverse in F. Hence Theorem 1 and Lemma 1 remain valid under the weaker hypothesis that the $\{c_i\}$ and $\{s_i\}$ are elements of a commutative ring.

Two developments that have come to our attention since the initial manuscript of this paper was prepared are deserving of mention. H. H. Harris of the Honeywell Corp., St. Petersburg, Fla. (private communication) has simulated a data compression scheme similar to that described in Section VI and reports an approximate 50-percent data reduction for digitized voice data. Zierler [5] has recently described the BCH decoding problem as a problem in ideals over polynomial rings in terms that are formally equivalent to Theorem 5 above.

Acknowledgment

The debt that the author owes to the work of Dr. E. R. Berlekamp of the Bell Telephone Laboratories is both obvious and gratefully acknowledged. Hopefully, the approach in this paper will yield additional insight into, and be of tutorial value for, Berlekamp's work [1]. It is also a pleasure to acknowledge the stimulating interest and helpful suggestions received from numerous colleagues; specific mention must be made of Profs. R. Gallager and M. Goutman of Massachusetts Institute of Technology, Cambridge, and of Dr. G. D. Forney, Jr., of Codex Corporation, Watertown, Mass.

References

[1] E. R. Berlekamp, "Nonbinary BCH decoding," presented at the 1967 Internat'l Symp. on Information Theory, San Remo, Italy. ——*Algebraic Coding Theory.* New York: McGraw-Hill, 1968, chs. 7 and 10.

[2] W. W. Peterson, *Error-Correcting Codes.* Cambridge, Mass: M. I. T. Press, and New York: Wiley, ch. 9, 1961.

[3] G. D. Forney, Jr., "On decoding BCH codes," *IEEE Trans. Information Theory*, vol. IT-11, pp. 549–557, October 1965.

[4] R. T. Chien, "Cyclic decoding procedures for the Bose–Chaudhuri–Hocquenghem codes," *IEEE Trans. Information Theory*, vol. IT-10, pp. 357–363, October 1964.

[5] N. Zierler, "A complete theory for generalized BCH codes," *Proc. 1968 Symp. on Error Correcting Codes*, H. B. Mann, Ed. New York: Wiley, 1968.