Difference-Set Cyclic Codes

By E. J. WELDON, JR.

(Manuscript received January 14, 1966)

Codes exist which are capable of correcting large numbers of random errors. Such codes are rarely used in practical data transmission systems, however, because the equipment necessary to realize their capabilities — that is, to actually correct the errors — is usually prohibitively complex and expensive. The problem of finding simply implemented decoding algorithms or, equivalently, codes which can be decoded simply with existing methods, is perhaps the outstanding unsolved problem in coding theory today.

In this paper, a new class of random-error-correcting cyclic codes is defined. These codes have two very desirable features: the binary members of the class are nearly as powerful as the best-known codes in the range of interest, and they can be decoded with the simplest known decoding algorithm. Unfortunately there are relatively few codes with useful parameters in this class, despite the fact that the class is infinite.

I. INTRODUCTION

The Bose-Chaudhuri[1]-Hocquenghem[2] (BCH) cyclic codes are, as a class, the best of the known, constructive, random-error-correcting codes. Fortunately a decoding algorithm, which can be implemented with a reasonable amount of equipment, has been found for these codes.[3,4,5]

In this paper, a new class of random-error-correcting cyclic codes is presented. These codes can be implemented much more simply than the BCH codes and are approximately as powerful. Unfortunately, the class is a small one.

II. DIFFERENCE-SET CYCLIC CODES

A simple perfect difference set of order l and modulus $n = l(l - 1) + 1$ is defined as a collection of l integers chosen from the set $\{0, 1, \cdots, l(l - 1)\}$ such that no two of the $l(l - 1)$ ordered differences modulo n are identical. That is, each occurs once. Singer[6] has shown

how to construct such sets when $l = p^s + 1$, p prime, s a positive integer, while Evans and Mann[7] have shown that a perfect difference set cannot be constructed for any other value of $l \leqq 1600$.

Since adding a fixed integer to every element of a perfect difference set clearly results in another such set, no loss of generality is suffered by considering only sets containing the element 0. In what follows, all perfect difference sets will be of this type.

Denote the elements of a perfect difference set of order $p^s + 1$ by $d_1 = 0, d_2, \cdots, d_{p^s+1}$ and let

$$\theta(x) = x^{d_1} + x^{d_2} + \cdots + x^{d_{p^s+1}}$$

be a polynomial in the algebra of polynomials modulo $x^n - 1$. The coefficients of all polynomials are taken from $GF(p^r)$, $r \leqq s$. Consider the n-by-n cyclic matrix Θ over $GF(p^r)$ whose rows are the coefficient vectors of $\theta(x), x\theta(x), \cdots, x^{n-1}\theta(x)$. For reasons which will become apparent shortly, the subspace of n-tuples generated by the rows of this matrix, which is an ideal in the algebra of polynomials modulo $x^n - 1$, will be considered to be the null space of a cyclic code of length n. The rank of this matrix, which will be shown to be the number of check symbols in the code, can be determined as follows. Consider the product

$$\begin{aligned}\theta(x)\theta(x^{-1}) = 1 &+ x^{d_1-d_2} &&+ x^{d_1-d_3} + \cdots + x^{d_1-d_{p^s+1}} \\ &+ x^{d_2-d_1} + 1 &&+ x^{d_2-d_3} + \cdots + x^{d_2-d_{p^s+1}} \\ &\vdots \\ &+ x^{d_{p^s+1}-d_1} + x^{d_{p^s+1}-d_2} + \cdots &&+ 1. \end{aligned} \tag{1}$$

Since the d_i are elements of a perfect difference set, each integer 1, 2, $\cdots$, $n - 1$ appears as an exponent in this polynomial exactly once. Thus, in the algebra of polynomials modulo $x^n - 1$,

$$\theta(x)\theta(x^{-1}) \equiv p^s + 1 + x + x^2 + \cdots x^{n-1}. \tag{2}$$

The reciprocal polynomial of $\theta(x)$, which is denoted by $\theta^*(x)$, is equal to $x^{(\text{degree of } \theta(x))} \cdot \theta(x^{-1})$. Since $p^s \equiv 0 \bmod p^r$, (2) reduces to

$$\begin{aligned}(x - 1)\theta(x)\theta^*(x) &\equiv 0 \\ &= q(x)(x^n - 1) \end{aligned} \tag{3}$$

for some polynomial $q(x)$.† Let $\theta(x) = f(x)h(x)$ where $h(x)$ is the

† If the code symbols are chosen from $GF((p')^r)$, p' a prime not equal to p, then $GCD(\theta(x), x^n - 1)$ equals a nonzero ground-field element, and the code specified by $h(x)$ is the trivial code which has n parity checks and no information symbols.

greatest common divisor of $\theta(x)$ and $x^n - 1$. That is,

$$h(x) = GCD(\theta(x), x^n - 1). \tag{4}$$

As a result the ideals (cyclic codes) in the algebra of polynomials modulo $x^n - 1$ generated by $\theta(x)$ and $h(x)$ are identical. Thus, the rank of the matrix θ is simply n minus the degree of $h(x)$, i.e., the number of check digits in the code generated by $g(x) = (x^n - 1)/h(x)$.

In an accompanying paper,[8] Graham and MacWilliams prove that there are exactly

$$n - k = \left[\binom{p+1}{2}\right]^s + 1 \tag{5}$$

check symbols in a difference-set cyclic (n,k) code over $GF(p)$. But since $\theta(x)$ and $x^n - 1$ are both polynomials over $GF(p)$, so are $h(x)$ and $g(x)$, regardless of the field from which the code symbols are chosen. Thus, (5) holds over $GF(p^r)$, $r \leqq s$, as well.

Although the derivation of (5) is fairly involved, it is easy to see that $n - k \leqq (n+1)/2$. For each zero of $x^n - 1$ except unity must be a zero of either $\theta(x)$, $\theta^*(x)$ or both, so it must also be a zero of either $h(x)$, $h^*(x)$ or both. That is,

$$(x - 1)h(x)h^*(x) = r(x)(x^n - 1)$$

where $r(x)f(x)f^*(x) = q(x)$. Therefore, the degree of $h(x)$ that is, k, cannot be less than $(n-1)/2$ and $n - k \leqq (n+1)/2$.

As defined, $g(x)$ is the generator polynomial of the code whose null space is generated by $h(x)$. Two polynomials multiply to zero in the algebra of polynomials modulo $x^n - 1$ only if the dot product of their coefficient vectors, with the order of the components reversed in one of them, is zero. Thus, the coefficient vectors of $\theta^*(x)$, $x\theta^*(x)$, $\cdots$, $x^{n-1}\theta^*(x)$ are in the null space of the code generated by $g(x)$. Each of these vectors represents a generalized parity check equation on certain symbols in each code word. Since each of these equations has $p^s + 1$ nonzero terms, each of the n symbols in every code vector is involved in exactly $p^s + 1$ equations. This follows from the fact that all the rows of the matrix θ are cyclically shifted versions of the first row and thus all columns and rows have the same weight. Furthermore, because a perfect difference set generates each difference exactly once, no two of the $p^s + 1$ equations which check a particular symbol can both check any other symbol. For if they could, this would imply that some differ-

ence was generated twice by the perfect difference set, which is impossible.

Consequently, these equations form an "orthogonal check set" of order $p^s + 1$ on the symbol in question. Massey[9] has defined such a set to consist of a collection of equations, all of which check a particular symbol, with the property that no two symbols appear together in more than one equation. He has shown that if it is possible to form an orthogonal check set of order $d - 1$ on any symbol in a cyclic code, then the code has minimum distance at least d and can be decoded with majority-logic decoding. Thus, difference-set codes have minimum distance at least $p^s + 2$ and, as described in Section III, can be decoded in a very straightforward manner.

Although codes exist over all finite fields, binary codes are, from a practical viewpoint, the most interesting. Table I contains a list of the first few binary difference-set codes and their generator polynomials. These codes have several interesting properties. For example, let d' denote the minimum distance of a code which can be realized by threshold decoding and let $\bar{d}$ denote the minimum distance of its dual code.

TABLE I — LIST OF BINARY DIFFERENCE-SET CYCLIC CODES

s	$n = 2^{2s} + 2^s + 1$	k	$d = 2^s + 2$	$t = 2^{s-1}$	Generator polynominal, $g(x)$	Difference—set polynomial $\theta(x)$
1	7	3	4	1	4,3,2,0	3,2,0
2	21	11	6	2	10,7,6,4,2,0	11,8,7,2,0
3	73	45	10	4	28,25,22,16,12,8,6,4,2,0	45,42,36,29,25,24,10,2, 0
4	273	191	18	8	82,77,76,71,67,66,56,52,48, 40,36,34,24,22,18,10,4,0	201,196,186,167,166, 159,128,126,115,112, 103,67,50,46,24,18,0
5	1057	813	34	16	244,242,236,234,232,228, 226,224,222,216,214,212, 211,210,209,208,203,202, 201,200,199,198,195,194, 193,191,189,188,186,184, 183,182,181,180,179,178, 177,176,175,174,169,167, 166,165,164,161,160,158, 155,154,153,151,150,149, 147,146,142,141,138,137, 135,132,131,129,126,124, 123,122,121,120,116,115, 114,111,108,106,105,103, 101,98,96,95,88,83,81, 79,76,75,74,72,71,70,68, 62,59,55,52,51,48,47,45, 43,41,39,37,35,33,32,28, 27,26,23,22,18,17,14,11, 5,4,3,1,0	1023,990,924,905,879, 792,754,702,697,677, 597,555,528,511,452, 439,348,338,298,277, 255,219,138,127,109, 63,54,31,15,7,3,1,0

Since any symbol appears in all equations orthogonal on that symbol and no other symbol appears in more than one of these equations, it is clear that the following bound holds for all block codes decoded with threshold decoding:

$$(d' - 1)(\bar{d} - 1) \leqq n - 1. \tag{6}$$

For difference-set codes $d' = 2^s + 2$, and in the binary case at least, $\bar{d} = 2^s + 1$, the weight of $\theta(x)$.[8] Thus, the equality holds in (6) and the codes are, in this peculiar sense, optimal.

In summation, it has been shown that there exists a class of cyclic codes of length $n = p^{2s} + p^s + 1$, i.e., those generated by $g(x) = (x^n - 1)/h(x)$ where $h(x) = GCD(\theta(x), x^n - 1)$, which have

$$\left[\binom{p+1}{2}\right]^s + 1$$

parity symbols[8] and which have the polynomials $\theta(x)$ and its multiples in their null spaces. Because of this latter property, the codes have minimum distance of at least $p^s + 2$. Also, in what follows, it is shown that this property can be used to implement these codes as random-error correctors in a remarkably simple manner.

III. IMPLEMENTATION

Because the codes are cyclic they can be encoded simply. See pages 148 and 149 in Peterson.[10]

Massey's majority-logic implementation of Meggitt's[11] general decoder for cyclic codes can be used to decode difference-set cyclic codes. A decoder of this type is shown in Fig. 1; it operates as follows. With the switch in the D position, the k-symbol data sequence is shifted into the syndrome and data registers simultaneously. When the entire data block has been entered, the switch is thrown to position P and the n-k received parity checks are shifted into the syndrome register, forming the syndrome. At this time, the output of the majority gate equals the additive inverse of the noise digit which was added to the first data symbol by the channel, provided that fewer than p^{s-1} errors occurred in the word.† Consequently, all that must be done to correct the first symbol is to add the output of the majority gate to the received symbol. This is done by the adder ($\oplus$) during the first shift of the registers.

† The majority gate has the following characteristics: Its output equals the additive inverse of the ground-field element which occurs most frequently among its $p^s + 1$ inputs, provided that that element occurs at least $p^{s-1} + 1$ times. Otherwise it equals zero.

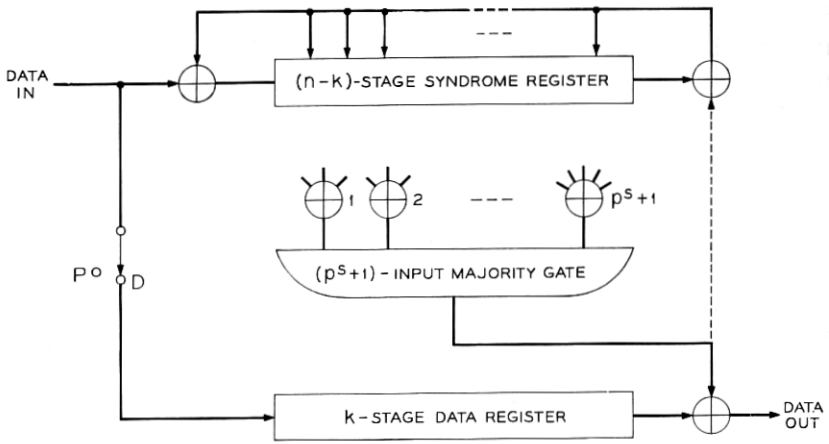

Fig. 1 — Fixed threshold majority-logic decoder for difference-set cyclic codes.

Because of the cyclic nature of the codes, repeating this process k times corrects all errors in the data section of the word.

Performance can be improved slightly at virtually no cost by the addition of the dotted connection and its associated adder to the decoder. This circuit removes the effects of corrected errors from the syndrome as the errors are corrected. This can be seen as follows. The first (highest-order) data symbol is checked by the parity check symbols stored in the shift register stages just to the left of the feedback connections. (This can be seen by examining the generator matrix of the code in systematic form.) If a particular ground-field element is added to the first information symbol to correct it, in order to remove the error from the parity check equations which checked that symbol, it is necessary to add the same ground-field element to each of these parity check symbols. This can be accomplished by adding the symbol to the feedback signal and shifting the register once. Successive corrections can be made in exactly the same manner.

Although a code will correct all error patterns of weight not greater than p^{s-1} and some of greater weight if this connection is omitted, many more patterns of weight greater than p^{s-1} will be corrected if it is included.

The presence of this connection also enables the decoder to detect undecodable error patterns. This is done by shifting the syndrome register $n - k$ times after the data symbols are corrected. A decodable pattern will have an all-zero syndrome after correction, while an undecodable pattern will not.

The connections from the syndrome register to the $p^s + 1$ summing circuits are made as follows. Each of the $p^s + 1$ equations orthogonal on the first information symbol involves one or more parity check symbols. The sum modulo p^r of all check symbols involved in the ith equation, $i = 1, 2, \cdots, p^s + 1$, equals the ith check sum orthogonal on the first information symbol. Thus, the inputs to the ith summing circuit are simply those parity check symbols involved in the ith equation.

This point is illustrated in Fig. 2 in which the decoder for the (73,45) binary code listed in Table I is depicted. Table II lists the 28-bit parity check sections of the nine composite parity check equations orthogonal on the first information symbol. Each of these equations manifests itself as one of the nine inputs to the majority gate of Fig. 2.†

A modification of threshold decoding, Variable Threshold Decoding, has been used to decode quasi-cyclic codes.[12] It can also be used here to improve performance somewhat, at the cost of a slight increase in complexity. Instead of keeping the threshold set at $p^{s-1} + 1$, it is initially set at its maximum value, $p^s + 1$, and an attempt is made to decode each of the n symbols of the received word. When an n-symbol cyclic revolution of the syndrome has been completed without any changes being made, the threshold is reduced by one and another attempt is made. If another complete revolution is made with no changes, the threshold is lowered again. If a change is made, however, the threshold is immediately raised by one and decoding continues. Upon the completion of the revolution, the threshold is again lowered by one.

Eventually one of two things must occur. Either the threshold will drop to its minimum value and remain there, or it will enter into some sort of limit cycle wherein it changes repetitively between two or more levels. In a practical system this latter difficulty can be obviated by terminating decoding after a fixed number of attempts, and if it is appropriate, signaling a detected error if the syndrome is not all zeros. It seems likely that roughly $2d$ cyclic revolutions of the word will suffice to decode nearly all decodable error patterns.

IV. A COMPARISON

In an unpublished report,[13] the author estimated that approximately 3600 transistors would be required to instrument a decoder for the (273,200), $d = 18$, code formed by shortening the (511,438) primitive BCH code.‡ Also it was estimated that the decoder's internal circuitry

† Using a combination of sequential and combinational circuits, rather than strictly combinational circuits, would undoubtedly result in a slightly cheaper, but conceptually more complex decoder.

‡ This was before Berlekamp's work.[5, 10]

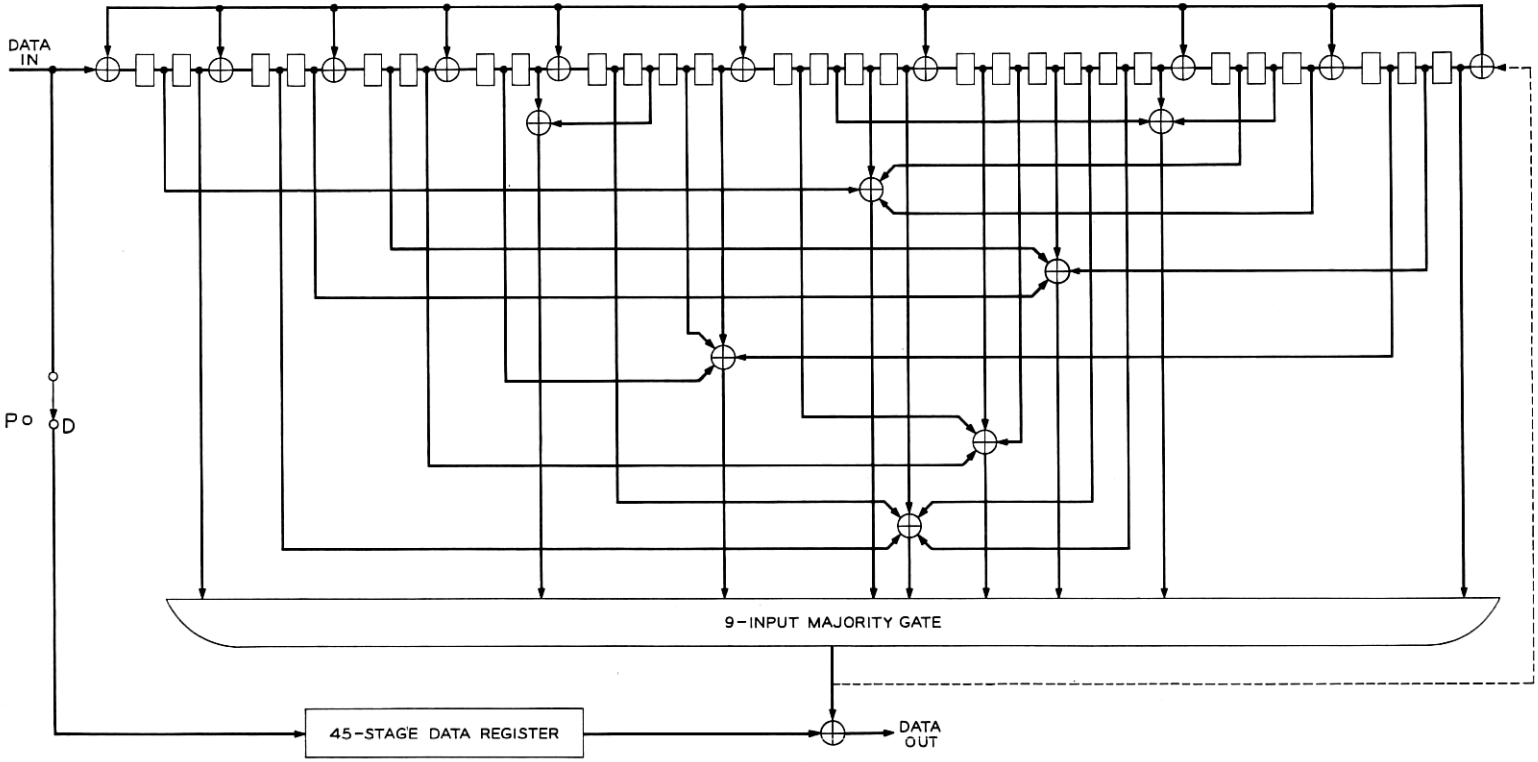

Fig. 2 — Decoder for (73, 45), $d = 10$, binary difference-set cyclic code.

TABLE II—PARITY SECTIONS OF THE NINE COMPOSITE PARITY CHECK EQUATIONS ORTHOGONAL ON FIRST INFORMATION SYMBOL IN (73, 45) CODE.

Eq Number	Bit position																											
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
1																												1
2		1																										
3								1		1																		
4													1									1		1				
5	1													1									1		1			
6				1	1														1								1	
7							1				1	1															1	
8						1							1				1	1										
9			1						1						1				1		1							

would have to operate roughly 85 times faster than line speed to enable it to keep up with the data. These estimates were based on the assumptions that only transistor storage was used and that speed would be sacrificed to reduce the number of transistors needed whenever practical. For example, the decoder is a serial, rather than parallel, device.

Assuming that the syndrome register, exclusive-OR circuits and majority gate are duplicated, the internal circuitry of the decoder for the (273,191), $d = 18$, binary difference set code can operate at line speed. The numbers of transistors required for the various decoder components are tabulated below.

Circuit Function	Number of Transistors
Data register	382
Syndrome registers (2)	328
Exclusive-OR's	400
Majority gates (2)	40
Clock and switches	30
Miscellaneous	20
Total	1200

The complexity of the two decoders is compared in the following table.

Decoder for	Code Efficiency	Number of Transistors	Decoder Clock Rate (multiples of line bit rate)	Ease of Design, Construction, & Testing
BCH Code	0.73	3600	85	difficult
Difference-Set Code	0.70	1200	1	very simple

This type of comparison is quite crude and is admittedly chosen to illustrate the strong points of difference-set codes. Also, in light of Berlekamp's work,[5, 16] it may be possible to reduce the number of transistors required for the BCH decoder by as much as a factor of two and its speed by twice that. However, despite these facts, the difference-set decoder remains a much simpler piece of equipment than the BCH decoder. Also, a comparison of longer, more powerful codes would demonstrate the relative simplicity of the difference-set decoder even more dramatically.

This comparison is not intended to demean the BCH codes or their very elegant and general decoding algorithm. There are certainly many cases, in fact nearly all cases involving long, relatively efficient random-error-correcting codes, in which they are by far the most easily implemented codes. Rather, the comparison is simply intended to point out that, in certain cases, difference-set codes are much easier to implement that the BCH codes, and to suggest that there may be other classes of cyclic codes for which the same is true.

V. CONCLUSIONS

A new, relatively small, class of random-error-correcting cyclic codes has been presented. These codes, which are approximately as powerful as the best cyclic codes for given values of efficiency and length, are very easily implemented. Consequently, they are concluded to be attractive for use in error-control systems where forward-acting random-error-correction is required.

VI. ADDENDA

Subsequent to the discovery of these codes, the author became aware of the unpublished, but earlier, work of L. D. Rudolph.[14] In it a class of threshold-decodable codes, which contains the class of difference-set cyclic codes, is described. Also, (6) has been derived by Mitchell in Ref. 15.

REFERENCES

1. Bose, R. C. and Ray-Chaudhuri, D. K., On a Class of Error Correcting Binary Group Codes, Inform. Control, *3*, 1960, pp. 68–79.
2. Hocquenghem, A., Codes Correcteurs d'erreurs, Chiffres, *2*, 1959, pp. 147–156.
3. Peterson, W. W., Encoding and Error-Correction Procedures for the Bose-Chaudhuri Codes, IRS Trans., *IT-6*, 1960.
4. Chien, R. T., Cyclic Decoding Procedures for Bose-Chaudhuri-Hocquenghem Codes, IEEE Trans., *IT-10*, No. 4, 1964.

5. Berlekamp, E. R., On Decoding Binary Bose-Chaudhuri-Hocquenghem Codes, IEEE Trans., *IT-11*, No. 4, 1965.
6. Singer, J., A Theorem in Finite Projective Geometry and Some Applications to Number Theory, AMS Trans., *43*, 1938, pp. 377–385.
7. Evans, T. A. and Mann, H. B., On Simple Difference Sets, Sankhya, *11*, 1955, pp. 464–481.
8. Graham, R. L. and Mac Williams, Jessie, On the Number of Parity Checks in Difference-Set Cyclic Codes, B.S.T.J., this issue, pp. 1057–1070.
9. Massey, J. L., *Threshold Decoding*, MIT Press, 1963.
10. Peterson, W. W., *Error Correcting Codes*, MIT Press, 1961.
11. Meggitt, J. E., Error-Correcting Codes and Their Instrumentation for Data Transmission Systems, IRE Trans., *IT-7*, No. 4 1961.
12. Townsend, R. L. and Weldon, E. J., Jr., Self-Orthogonal Quasi-Cyclic Codes, To be published, IEEE Trans., *IT-12*, 1966.
13. Weldon, E. J., Jr., Complexity of Peterson-Chien Decoders, unpublished memorandum, 1965.
14. Rudolph, L. D., A Class of Majority Logic Decodable Codes, To be published, IEEE Trans., *IT-12*, 1966.
15. Mitchell, M. E., et al., Coding and Decoding Operations Research, Final Report on Contract AF 19 (604)-6183, AFCRL8, 1961.
16. Berlekamp, E. R., Practical BCH Decoders, IEEE Trans., *IT-13*, to be published, 1967.