Other ways of interpreting the operations of the circuit in Fig. 2 are possible. For instance the circuit may be viewed as one of solving the polynomial equation

$$\sigma_0 + \sigma_1 x + \sigma_2 x^2 + \cdots + \sigma_t x^t = 0$$

with the x's substituted by 1, α, α^2, $\cdots$, α^{n-1}, respectively. In fact, an alternative circuit could be designed according to the polynomial

$$\sigma_0 x^t + \sigma_1 x^{t-1} + \cdots + \sigma_t = 0,$$

but it would require either the data in the buffer to be lower-order first, or the α^k-multipliers be replaced by α^k-dividers.

Acknowledgment

The author is indebted to his colleagues Drs. J. E. Meggitt, D. T. Tang, and C. V. Freiman, of Thomas J. Watson Research Center, Yorktown Heights, N. Y., for helpful suggestions in the development of these ideas. Helpful comments were also contributed by Drs. A. H. Frey and F. Corr of IBM Communications Systems Center, Bethesda, Md.

References

[1] W. W. Peterson, "Error-Correcting Codes," John Wiley and Sons, Inc., New York, N. Y.; 1961.
[2] E. Gorog, "Some new classes of cyclic codes used for burst-error correction," *IBM J. Res. Developm.*, vol. 7, pp. 102–111; April, 1963.
[3] R. C. Bose and C. R. Ray-Chaudhuri, "On a class of error-correcting binary group codes," *Information and Control*, vol. 3, pp. 68–79; 1960.
[4] A. Hocquenghem, "Codes correcteurs d'erreurs," *Chiffres*, vol. 2, pp. 147–156; September, 1959.
[5] W. W. Peterson, "Encoding and error-correction procedures for the Bose-Chaudhuri codes," IRE TRANS. ON INFORMATION THEORY, vol. IT-6, pp. 459–470; September, 1960.
[6] N. Zieler and D. Gorenstein, "A class of error-correcting codes in p^m symbols," *J. Soc. Ind. Appl. Math.*, vol. 9, pp. 207–214; June, 1961.
[7] J. E. Meggitt, "Error-correcting codes and their implementation for data transmission systems," IRE TRANS. ON INFORMATION THEORY, vol. IT-7, pp. 234–244; October, 1961.
[8] R. B. Banerji, "A decoding procedure for double-error correcting Bose-Ray-Chaudhuri codes," PROC. IRE, (Correspondence), vol. 49, p. 1585; October, 1961.
[9] T. C. Bartee and D. I. Schneider, "An electronic decoder for Bose-Chaudhuri-Hocquenghem error-correcting codes," IRE TRANS. ON INFORMATION THEORY, vol. IT-8, pp. S 17–24; September, 1962.
[10] A. A. Albert, "*Fundamental Concepts on Higher Algebra*," University of Chicago Press, Chicago, Ill.; 1956.
[11] T. C. Bartee and D. I. Schneider, "Computation with finite fields," *Information and Control*, vol. 6, pp. 79–98; 1963.

Nonrandom Binary Superimposed Codes

W. H. KAUTZ, MEMBER, IEEE, AND R. C. SINGLETON, SENIOR MEMBER, IEEE

Summary—**A binary superimposed code consists of a set of code words whose digit-by-digit Boolean sums (1 + 1 = 1) enjoy a prescribed level of distinguishability. These codes find their main application in the representation of document attributes within an information retrieval system, but might also be used as a basis for channel assignments to relieve congestion in crowded communications bands. In this paper some basic properties of nonrandom codes of this family are presented, and formulas and bounds relating the principal code parameters are derived. Finally, there are described several such code families based upon (1) q-nary conventional error-correcting codes, (2) combinatorial arrangements, such as block designs and Latin squares, (3) a graphical construction, and (4) the parity-check matrices of standard binary error-correcting codes.**

Manuscript received October 18, 1963. The research reported was performed at Stanford Research Institute, Menlo Park, Calif.

W. H. Kautz is with the Computer Techniques Laboratory, Engineering Division, Stanford Research Institute, Menlo Park, Calif.

R. C. Singleton is with the Mathematical Sciences Department, Engineering Division, Stanford Research Institute, Menlo Park, Calif.

I. Introduction

THE FOLLOWING two coding problems arise in the representation and handling of data in a certain type of information retrieval system, to be described in detail below. Let the *sum* of two n-digit binary code words be their digit-by-digit Boolean sum; for example,

$$\begin{array}{r} 0\ 1\ 1\ 0\ 0\ 1 \\ \vee\ 0\ 1\ 0\ 0\ 1\ 0 \\ \hline 0\ 1\ 1\ 0\ 1\ 1 \end{array}$$

We seek a large number N of code words such that, for a given small positive integer m, every sum of up to m different code words is distinct from every other sum of m or fewer code words (Problem 1), or logically includes no code word other than those used to form the sum (Problem 2). It will be shown shortly that these two problems are intimately related, hence their simultaneous consideration in this paper.

A code whose code words satisfy the condition of Problem 1 will be said to be *uniquely decipherable* of order m, abbreviated UD_m. This name derives directly from the definition, which guarantees that any sum word composed of up to m constituent code words of a UD_m code can be decomposed into constituent code words in only one way. For example, the list of eight 7-digit code words,

$$\begin{array}{cccccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{array}$$

not only contains no duplicates, but when augmented with all $\binom{8}{2} = 28$ pairwise sums of code words still contains no duplicates. (This fact can be verified by listing all of the pairwise sums, or more easily by checking separately the manner in which sums having three and four *ones* are formed.) Thus, this set of eight code words constitutes a UD_2 code.

A code whose code words satisfy the condition of Problem 2 will be said to be *zero-false-drop* of order m, (ZFD_m). This name derives from the retrieval application, to be described in the next section. The three 3-digit code words having a single *one*, namely,

$$\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array},$$

clearly form a ZFD_2 code, since no pairwise sum such 1 1 0 can logically include the other code word, 0 0 1. In fact, somewhat trivially, this code is also ZFD_3. Note that it is also UD_2 and UD_3.

In Section II there is a description of the origin of the need for superimposed codes and their applications—a discussion which may be skipped by the reader interested in codes only for their own sake. Basic properties of these codes and bounds on the code size N in terms of the order m and the code-word length n are derived in Sections III and IV. Several families of codes of arbitrarily large size and order are then developed in Sections III-VII.

II. Applications

A. Retrieval Files

A superimposed code such as a ZFD code may be utilized in an information retrieval file as follows [1]–[3]. Before encoding, the retrieval file consists of a long list of entries, one for each document in the file. Each entry contains an identification number of the document (for later physical retrieval), plus a short list of attributes, called *descriptors*, which are selected from a descriptor dictionary to describe the contents of the document in question. A typical dictionary might contain a number N of descriptors between 10^3 and 10^4, and the maximum number m of descriptors per document would normally fall between 5 and 15 for a given file. The file size is essentially unlimited.

An inquiry to such a file takes the form of a prescribed list of "quiz" descriptors, and a test as to (a) *whether* and (b) *which* documents in the file have included in their associated descriptor lists all of the descriptors on the quiz list. Thus, mechanization of the file and the inquiry process requires that all of the document data be encoded so that this inclusion test can be performed rapidly and with a minimum of equipment. Methods are already available for efficiently encoding the identification numbers, and for determining which documents (Step b of the test) respond to an inquiry [4], if a means is available for determining only whether or not any documents respond (Step a). ZFD_m codes are proposed for this latter purpose, for encoding the descriptor portions of each document entry in the file.

To this end let each of the N descriptors in the dictionary be assigned a unique n-digit binary code word of a ZFD_m code. The descriptor list associated with each document is then represented by a new n-digit word, which is obtained by forming the digit-by-digit Boolean *sum* of the code words of all of its constituent descriptors. The code words of the quiz descriptors are summed into a quiz word in identical fashion. It then follows directly from the ZFD_m property of the code that, as long as no more than m descriptors are associated with any one document, *the quiz word is logically contained in a particular document word if and only if all of the quiz descriptors are included among the descriptors associated with the document.* If this inclusion test is satisfied for any one or more document words in the file, in response to an inquiry, then it may be arranged so that an output is provided from the file. Otherwise, no output is obtained.

Various electrical and mechanical realizations of this type of retrieval file have been constructed or proposed, [5]–[7] and several are commercially available. For example, if edge-notched cards are used, each document is represented by a card which carries the binary sum word as a pattern of notches over n possible notch positions on one or more edges of the card, the bottom edge, say. An inquiry can be made by resting a stack of such cards on a set of small bars that are raised up underneath the stack in those notch positions corresponding to the location of *ones* in the quiz code word. All the cards having notches in at least these positions will remain stationary, while the unwanted cards will be raised, and can be separated from the desired set.

Codes presently in use for such retrieval files are generated by a random selection process [1]–[3]. Each descriptor code word is formed by placing a few *ones* (typically, three or four) randomly in an n-digit binary field. The proper value of n for this random superimposed code can be determined by statistical analysis, to reduce to a prescribable minimum the probability that an un-

wanted document will drop out during an inquiry [8]–[12]. Such a "false drop" could occur if a sum code word logically included one or more code words *other* than those used to compose it.

While a few false drops can be easily weeded out by the user of a file, they are nevertheless a nuisance, and their occurrence may become intolerable if the number of them becomes too great. Because of the simplifying assumptions made in even the best statistical analysis of random superimposed codes (equal descriptor usage, unrestricted dictionary size, uncorrelated descriptor selection), it is not possible to guarantee a desired minimum false-drop probability without very conservative design choices. Even so, a random code will always have its deviates from the mean performance. Thus, a particular new code can be expected to have a few bad code-word combinations, and there is always a chance that a new code will have poor over-all performance characteristics. Finally, another shortcoming of random codes is that a search with one or more negated descriptors cannot be performed without risking "false misses,"—that is, rejection of desired items. *ZFD* codes do not have this problem.

It is primarily to overcome these shortcomings that the new family of superimposed codes has been studied. Just as with conventional error-correcting codes, they provide completely error-free performance up to a certain level of activity. Analogously, the random superimposed codes correspond to random conventional codes such as have been discussed by Shannon [13] and Elias [14].

It is also true of randomly generated superimposed codes that once a sum code word is formed for a document, it is not generally possible to determine directly from this sum all of the constituent descriptors. That is, the deciphering of sumwords is, in general, not unique. On the other hand, it will be shown in the next section that any ZFD_m code is automatically a UD_m code, so that the sum code words of the new codes are automatically decipherable.

B. Data Communication

Certain crowded communication bands, such as the amateur band, telephone trunk lines, and certain military radio bands, are characterized by a limited number n of channels but a larger number, N, of low-duty users. Thus, it is not possible to assign for all time one channel to each user, and some stratagem must be employed to make the assignments variable and on demand. The usual practice is to employ a master control unit, a switching central, or an "operator" to keep track of which channels are available, and to assign them as needed. In the amateur bands, centralized control is dispensed with, in favor of the less reliable practice of letting each user locate a free channel as best he can.

If one could be assured that no more than m users would be needing the band at the same time, each user could be permanently assigned a *set* of channels on which he was instructed to transmit and/or listen simultaneously. If the assignment were made in accordance with a ZFD_m code, this user could be assured that his set of assigned channels would never *all* be in use at the same time by any other user or group of users. In this manner, he could communicate at any time without consulting a master control unit, subject only to this limitation on the maximum number m of simultaneous users. If this limit is not already imposed by the statistics of use of a particular system, it may not be unreasonable to provide a rudimentary form of master control which notifies all users only when the band is full.

The use of broadbanding techniques for the alleviation of crowding in busy communication bands was argued by Costas [15]. This suggested application of ZFD_m codes might provide a means whereby the practicality of the broadbanding philosophy may be tested.[1]

C. Magnetic Memories

It has been shown [17] that the problem of designing a certain family of multiply-threaded magnetic-core matrix memories can be expressed as the search for a suitable winding pattern which can be expressed in an *N-by-n winding matrix* A. The binary entries of this matrix describe compactly which of the n drive windings are threaded through which of the N cores which compose the memory array. The reader is referred to the literature for a detailed formulation of this problem in matrix terms. We note here only the close relation between the principal design parameter of these arrays, the *selection ratio* s, and the order m associated with the matrix A when it is used as the basis for a superimposed code. In terms of the so-called *excitation matrix*,

$$\Lambda = AA',$$

with elements λ_{ij} $(i, j = 1, 2, \cdots N)$, the selection ratio is

$$s = \left\lceil \frac{\underset{i}{\text{Min}}\ (\lambda_{ii})}{\underset{i \neq j}{\text{Max}}\ (\lambda_{ij})} \right\rceil.$$

It is shown in Section IV that the matrix A is a general representation of a binary superimposed code whose maximum order is bounded by

$$m \geq \left\lfloor \frac{\underset{i}{\text{Min}}\ (\lambda_{ii}) - 1}{\underset{i \neq j}{\text{Max}}\ (\lambda_{ij})} \right\rfloor$$

and (later) that this inequality may frequently be replaced by an equality. As a result of this correspondence between the problem of memory design and the problem of developing desirable superimposed codes, it should be possible to make use of results obtained independently

[1] Another communications application related to binary superimposed codes has been proposed by Cohn and Gorman [16], and has to do with the use of a suggested family of codes having limited superposition properties for the selective calling of stations in a network.

on either problem to generate additional solutions to the other.

In addition, it was shown by Minnick in 1957 that the higher selection ratio obtainable in a multiply-threaded memory may be exchanged for the property of *simultaneous access*—that is, the ability to apply simultaneously more than a single address, and (with proper readout circuitry) to read out simultaneously the contents of the memory at all of these addresses [18]. In fact, it is this particular use of the additional windings that conforms most naturally to the superposition properties of the rows of an A-matrix (code words of a superimposed code).

Many magnetic-core memory arrays may also be used as the basis for the design of *access switches*, which differ from memories mainly in the addition of extra bias or inhibit windings and currents, and in the manner of use [19]. The principal design parameter of access switches is the *load-sharing factor*, which is normally equal to the *difference* between the two quantities which form the *quotient* in the above expression for s. However, we can still expect a mutually beneficial exchange between the catalogs of useful access-switch designs and binary superimposed codes, even though the notions of efficiency do not correspond exactly for the two problems.

III. Theoretical Results

In this section ZFD_m and UD_m codes are given mathematical definitions, and their interrelationship is shown.

The *superposition sum* $z = x \vee y$ (designated as the digit-by-digit Boolean sum up to now) of two n-dimensional binary vectors $x = (x^1, x^2, \cdots x^n)$ and $y = (y^1, y^2, \cdots y^n)$ is defined by:

$$z^i = \begin{cases} 0 & \text{if } x^i = y^i = 0 \\ 1 & \text{otherwise} \end{cases} \qquad i = 1, 2, \cdots n.$$

Also, a vector x is said to be *included in* a vector y if an only if

$$xvy = y.$$

From a given code C_1, which is a collection of N n-dimensional binary vectors called code words, we may readily construct for $k = 2, 3, \cdots N$ the kth *superposition sum set* C_k, which is the collection of all of the superposition sums of these code words of C_1, taken exactly k at a time. Thus, the set C_k contains $\binom{N}{k}$ vectors, which for $k > 1$ are not necessarily all different. In considering the sequence of sets $C_1, C_2, \cdots C_k, \cdots$, we are particularly interested in the value of k at which duplicate vectors first appear, either within the same set C_k, or between C_k and some earlier set. Toward this end, we have the following theorem and corollary.

Theorem 1: If the sets $C_1, C_2, \cdots C_{m+1}$ are disjoint (that is, if no vector occurs in two different sets of this list), then the set C_m contains exactly $\binom{N}{m}$ different vectors.

Proof: Suppose that two of the $\binom{N}{m}$ vectors in C_m were equal:

$$x_1 \vee x_2 \vee \cdots \vee x_m = y_1 \vee y_2 \vee \cdots y_m$$

where $x_1, x_2, \cdots x_m$, and $y_1, y_2, \cdots y_m$ are all code words in C_1. Then

$$y_j \vee x_1 \vee x_2 \vee \cdots \vee x_m = x_1 \vee x_2 \vee \cdots \vee x_m,$$

for every $j = 1, 2, \cdots m$. But C_{m+1} and C_m are disjoint, so that each of the code words $y_1, y_2, \cdots y_m$ must belong to the set of code words $\{x_1, x_2, \cdots x_m\}$. Thus, there are no duplicates in C_m, and C_m must contain $\binom{N}{m}$ different vectors.

Corollary: If the sets $C_1, C_2, \cdots C_{m+1}$ are disjoint, then the set C_k contains exactly $\binom{N}{k}$ different vectors for $k = 1, 2, \cdots m$. This theorem and corollary are used below to relate zero-false-drop and uniquely decipherable codes.

If only $C_1, C_2, \cdots C_m$ are disjoint, then C_m need not contain $\binom{N}{m}$ elements. For example, the code C_1 consisting of the seven cyclic permutations of (1101000) has C_1, C_2, and C_3 disjoint, but C_3 contains only eight elements, rather than $\binom{7}{3} = 35$. Furthermore, if C_1, $C_2, \cdots C_m$ are disjoint and C_m contains $\binom{N}{m}$ elements, $C_1, C_2, \cdots C_{m+1}$ need not be disjoint. For example, the code C_1 with elements $a = (1100)$, $b = (0011)$, and $c = (0110)$ has for C_2, the sum vectors $a \vee b = (1111)$, $a \vee c = (1110)$, and $b \vee c = (0111)$, and for C_3 the single element $a \vee b \vee c = (1111)$: the sets C_1 and C_2 are disjoint, C_2 contains $\binom{3}{2} = 3$ elements, but C_2 and C_3 are not disjoint.

A ZFD_m code may now be defined to be a set C_1 of code words for which no sum $y_1 \vee y_2 \vee \cdots \vee y_j$ of $j \leq m$ code words is included in any other sum $x_1 \vee x_2 \vee \cdots \vee x_k$ of $k \leq m$ code words, unless $y_1, y_2, \cdots y_j$ all belong to the set of code words $x_1, x_2, \cdots x_k$. Clearly, a code that is ZFD_m is also ZFD_k for $1 \leq k < m$ as well. An equivalent and somewhat more intuitive definition follows from the next theorem:

Theorem 2: A code is ZFD_m if and only if no sum $x_1 \vee x_2 \vee \cdots \vee x_k$ of $k \leq m$ code words includes any other code word y_i not used in this sum.

Proof: The sufficiency follows directly from the definition. If the sum $x_1 \vee x_2 \vee \cdots \vee x_k$ of $k \leq m$ code words includes no other code word y_i, then it cannot include a sum such as $y_1 \vee \cdots \vee y_i \vee \cdots \vee y_j$ of $j \leq m$ code words, unless $y_1, y_2, \cdots y_j$ all belong to the set of code words $\{x_1, x_2, \cdots x_k\}$.

In terms of the sequence of sets $C_1, C_2, \cdots C_k, \cdots$, we then have the following theorem.

Theorem 3: A code C_1 is ZFD_m if and only if the sets $C_1, C_2, \cdots C_{m+1}$ are disjoint.

Proof:

1) If the sets $C_1, C_2, \cdots C_{m+1}$ are disjoint, then a code word y_1 can be included in the sum $x_1 \vee x_2 \vee \cdots \vee x_k$ for $k \leq m$ only if y_1 is one of the code words $x_1, x_2, \cdots x_m$, so that C_1 is ZFD_m.
2) If C_1 is ZFD_m, suppose that C_j and C_k, for some $1 \leq j < k \leq m + 1$, have a common element $x_1 \vee x_2 \vee \cdots \vee x_j = y_1 \vee y_2 \vee \cdots \vee y_k$. But if each y_i is one of the code words $x_1, x_2, \cdots x_j$, then we cannot have $j < k$, and thus $C_1, C_2, \cdots C_{m+1}$ are disjoint.

A UD_m *code* may be defined to be a set C_1 of code words such that equality of any two sum vectors, each composed of no more than m code words, implies that the two sets of constituent code words of the sum vectors are identical. Thus Theorem 4 follows.

Theorem 4: A code C_1 is UD_m if and only if the sets $C_1, C_2, \cdots C_m$ are disjoint, and C_m contains $\binom{N}{m}$ different vectors.

Proof:

1) Suppose that the sets $C_1, C_2, \cdots C_m$ are disjoint and that C_m contains $\binom{N}{m}$ different elements. Then each set C_k for $1 \leq k \leq m$ contains $\binom{N}{k}$ different elements, and no two superposition sum vectors, each composed of no more than m code vectors but not composed of identical constituents, can be equal without contradicting either the condition that $C_1, C_2, \cdots C_m$ be disjoint, or that C_k contains $\binom{N}{k}$ different elements for $1 \leq k \leq m$.
2) Suppose that the code C_1 is UD_m. Since equality of two superposition sums of $\leq m$ code words implies identity of the two sets of code words, C_1, C_2, C_m are disjoint, and C_m contains $\binom{N}{m}$ different elements.

The relationship between ZFD and UD codes now follows directly from Theorems 3 and 4, and may be summarized as:

$$ZFD_m \Rightarrow UD_m \Rightarrow ZFD_{m-1} \Rightarrow UD_{m-1} \Rightarrow \cdots \Rightarrow ZFD_1 \Rightarrow UD_1.$$

Moreover, as shown earlier by counter examples in terms of the sets C_k, the reverse implications do not in general hold:

$$ZFD_{m-1} \not\Rightarrow UD_m \not\Rightarrow ZFD_m, \quad \text{etc.}$$

An alternative statement of the ZFD_m condition is as follows. Imagine that the code words in C_1 are arranged as the rows of an N-by-n matrix A. Then theorem 5 follows.

Theorem 5: The code C_1 is ZFD_m if and only if every subset of $m + 1$ rows of A contains an $(m + 1)$-columned identity submatrix.

Proof: The condition that C_1 be ZFD_m is equivalent to the requirement that in each subset of $m + 1$ rows of A, no one row may be included in the sum of the other m. This will be the case if, and only if, each row of this $(m + 1)$-rowed submatrix has a *one* in some column in which all other rows have a *zero*. Conversely, if every subset of $m + 1$ rows contains an identity submatrix of order $m + 1$, then no one of these rows may be included in the sum of the other m; hence, C_1 is ZFD_m.

IV. Bounds

A weak upper bound on the size N of a n-digit UD_m code can be obtained by merely counting the total number of different vectors in the sets $C_1, C_2, \cdots C_m$, and noting that this number cannot exceed the number of nonzero, n-digit binary numbers:

$$\sum_{k=1}^{m} \binom{N}{k} \leq 2^n - 1. \tag{1}$$

Better bounds result through the use of some intermediate parameters. The number of *ones* in code word x_i is called the *weight* w_i of that code word, while the *overlap* λ_{ij} between two code words x_i and x_j is simply their dot product—that is, the number of digit positions in which both words have *ones*. It will be convenient to refer to the minimum weight $w_{\min} = \text{Min}_i\, w_i$ and the maximum overlap $\lambda_{\max} = \text{Max}_{i,j}\, \lambda_{ij}$, $i \neq j$, where the Min and Max operations are taken over all N code words.

Now if a given code has a maximum overlap $\lambda_{\max}$ for all pairs of code words, then no particular $(\lambda_{\max} + 1)$-tuple of *ones* (that is, no set of $\lambda_{\max} + 1$ particular digit positions) can appear in more than one code word. The total possible number of such $(\lambda_{\max} + 1)$-tuples over n positions is just $\binom{n}{\lambda_{\max} + 1}$, and the ith code word accounts for just $\binom{w_i}{\lambda_{\max} + 1}$ of them. Summing over all N code words, then, we have the condition:

$$\sum_{i=1}^{N} \binom{w_i}{\lambda_{\max} + 1} \leq \binom{n}{\lambda_{\max} + 1}. \tag{2}$$

If all code words have the same weight w, this bound reduces to

$$N \leq \frac{\binom{n}{\lambda_{\max} + 1}}{\binom{w}{\lambda_{\max} + 1}}. \tag{3}$$

Moreover, if $w_i \geq m\,\lambda_{\max} + 1$, then the ith code word cannot possibly be contained in the sum of any m other code words, since it overlaps each of these other code

words in no more than λ_{max} positions. Thus, a code with minimum weight w_{min} and maximum overlap λ_{max} is ZFD_m for all m up to some value which satisfies

$$m \geq \left[\frac{w_{min} - 1}{\lambda_{max}}\right], \tag{4}$$

where the brackets denote the integer part of the quantity within.

In terms of the A-matrix, we may observe immediately that the λ_{ij} are the off-diagonal elements of the N-by-N matrix

$$\Lambda = AA^t,$$

while the w_i are the diagonal elements: $w_i = \lambda_{ii}$. Therefore, the search for an n-digit, N-word, ZFD_m code C_1, for which the lower bound (4) on the largest order m is maximized, is equivalent to the search for an N-by-n A-matrix which maximizes in Λ the ratio of the smallest diagonal element (less one) to the largest off-diagonal element.

The following theorem provides a condition under which the order m of a ZFD_m code is equal to the bound (4).

Theorem 6: If every λ_{max}-tuple appears in two or more code words of a code, this code is ZFD_m but not ZFD_{m+1} for

$$m = \left[\frac{w_{min} - 1}{\lambda_{max}}\right].$$

Proof: The code is at least ZFD_m by the bound (4). But if every λ_{max}-tuple appears in two or more code words, then for any code word whose weight is $w_i \leq (m+1)\lambda_{max}$, there can be found $(m+1)$ other code words whose sum contains it. Thus, the code cannot be ZFD_{m+1}.

If a code is ZFD_m for a value of m higher than the minimum set by the bound (4), numerous overlap possibilities are ruled out by the presence of code words of weight less than $m\,\lambda_{max} + 1$. The following theorem shows this for the case of words with weight no greater than m.

Theorem 7: If any code word of a ZFD_m code has weight no greater than m, it must have a *one* in some position in which no other code word has a *one*.

Proof: If not, this code word would be contained in some sum of m other code words, and the code would not be ZFD_m.

It follows directly that if all code words of a ZFD_m code have weight $w_i \leq m$, then $N \leq n$; *i.e.*, the number of code words is then no greater than the number of positions in an individual code word. Equality ($N = n$) is then achieved only if all code words have weight one.[2]

If *some* of the code words of a ZFD_m code have weights no greater than m, say $w_i \leq m$ for $i = 1, 2, \cdots N_1$, then the number N of code words satisfy a revised condition corresponding to (2), namely,

$$\sum_{i=1}^{N_1} \binom{w_i - 1}{\lambda_{max} + 1} + \sum_{i=N_1+1}^{N} \binom{w_i}{\lambda_{max} + 1} \leq \binom{n - N_1}{\lambda_{max} + 1}.$$

This bound takes into account the fact that at least N_1 of the n positions are used only once in the code. In fact, any code word whose weight is no greater than m can have its weight reduced to one without reducing the order m of the code, since each such code word has a *one* in some position in which no other code word has a *one*. Similarly, any code word whose weight exceeds $m\,\lambda_{max} + 1$ can have its weight reduced to this value, by arbitrary deletion of *ones*, without reducing the order m of the code. If the value of λ_{max} is decreased as a result of these deletions, the process can be repeated. Thus, given any ZFD_m code, another possibly different ZFD_m code having the same values of n, N, and m, but with all weights w_i equal to unity or satisfying $m + 1 \leq w_i \leq m\,\lambda_{max} + 1$, can be derived.

Clearly, then, the elimination of any weight-one code word and its corresponding digit position from a ZFD_m code will reduce by one both the n and N-values of the code, without changing its order m. In a similar manner, any ZFD_m code may be augmented with any number of weight-one code words, to increase both n and N by the same amount, without changing the order m of the code. While this process of "linear" decrease or increase may be useful in obtaining codes of particular desired sizes from other known codes, its inefficiency indicates that a search for more perfect codes should exclude weight-one code words, allowing only weights in the range

$$m + 1 \leq w_i \leq m\lambda_{max} + 1.$$

If all code words have the same weight w, then the bounds (2) and (4) above reduce to

$$N \leq \frac{\binom{n}{\lambda_{max} + 1}}{\binom{w}{\lambda_{max} + 1}} \tag{5}$$

and

$$m \geq \left[\frac{w - 1}{\lambda_{max}}\right]. \tag{6}$$

Johnson has provided some refinements of (5). In our notation, these read:

$$N \leq \left[\frac{n}{w}\left[\frac{n-1}{w-1}\left[\frac{n-2}{w-2}\left[\cdots\left[\frac{n-\lambda_{max}}{w-\lambda_{max}}\right]\cdots\right]\right]\right]\right]; \tag{7}$$

$$N \leq \left[\frac{n(w - \lambda_{max})}{w^2 - n\lambda_{max}}\right] \quad \text{when} \quad w^2 > n\lambda_{max}. \tag{8}$$

Also, interchanging *zeros* and *ones*,

$$N(n, w, \lambda_{max}) = N(n, n - w, n - 2w + \lambda_{max}).$$

In the special case $\lambda_{max} = 1$, the weight reduction process described above yields weights of unity and $\lambda_{max} + 1 = m + 1$, and no others. "Linear" deletion of

[2] However, UD_m codes with $w = m$ and $N > n$ do exist, as will be shown in Section VI.

the weight-one words then yields a constant-weight code which achieves the lower bound (6): $m = w - 1$. These codes are discussed in detail in Section V.

Finally, for a constant-weight UD_m code, the bound (1) may be refined to

$$\sum_{k=1}^{m} \binom{N}{k} \leq \sum_{i=w}^{mw} \binom{n}{i},$$

in which the right-hand sum expresses the number of possible n-digit binary vectors whose weights lie between w and mw. Even if the weight is not constant, then for any UD_m code we have the inequality

$$\sum_{k=1}^{m} \binom{N}{k} \leq \sum_{i=m}^{n} \binom{n}{i},$$

which may be verified by showing (in a comparison of the right-hand side with (1)) that the presence of any code words of weight $w_i < m$ makes it easier, not harder, to satisfy the inequality.

V. Construction of *ZFD* Codes

A. Codes Based Upon Conventional Binary Error-Correcting Codes

Our approach to the problem of constructing *ZFD* codes is to search among the known families of conventional error-correcting codes for those which have desirable superposition properties, or which can be modified to have these properties. This search has yielded a number of potentially useful code families of arbitrary order and of arbitrarily large size and length. However, further work would undoubtedly lead to better codes, as most of those given here can be augmented with additional code words (N increased) without reducing the values of n or m.

For given n and m, the "linear" augmentation process described in Section IV shows that the maximum size $N_{\max}(n, m)$ of a *ZFD* code is strictly increasing with n, since

$$N_{\max}(n, m) \geq N_{\max}(n - 1, m) + 1.$$

Thus codes of any particular size or length can be formed from the next smaller member of one of the code families offered in this section. Similarly, such particular codes may be obtained by deletion of digits and/or code words from larger codes. Furthermore,

$$N_{\max}(n, m) \geq N_{\max}(n, m') \quad \text{if} \quad m' \geq m.$$

The list of the n weight-one n-digit binary vectors (*i.e.*, the code defined by $A = I$, the n-by-n identity matrix) provides a trivial example of a ZFD_m code, having $N = n = m$, which cannot be augmented to form a larger code of the same length and order. These codes achieve the bound (1), and will be used later in this section as building blocks for the construction of larger codes by composition methods.

One large class of known binary codes, the binary group codes [20], can be ruled out for direct use as superimposed codes. Since these codes contain the zero vector, they are only UD_1, and not even ZFD_1. Even with the zero vector deleted, most of them are not ZFD_1, since the code usually contains a vector of large weight (such as 111 $\cdots$ 11) which includes at least one of the vectors of small weight.

If all of the code words of a ZFD_m code are constrained to have the same weight, however, its overlap $\lambda_{\max}$ may be related to the minimum number d of differing digits between any pair of code words; namely

$$d = 2(w - \lambda_{\max}),$$

which allows the bound (6) to be written

$$m \geq \left\lceil \frac{w - 1}{w - \frac{d}{2}} \right\rceil. \tag{10}$$

The quantity d may now be identified as the (minimum) *distance*, which characterizes the error-correcting property of a group code (or of any binary error-correcting code, for that matter). Thus, the search for a ZFD_m code of fixed weight w can be viewed as the search for a constant-weight conventional error-correcting code of distance

$$d = \frac{2w(m - 1) + 2}{m}.$$

One simple way to generate constant-weight error-correcting codes is to extract all words of the desired weight w from an arbitrary error-correcting code. This selection will certainly not reduce the distance. In fact, if the distance of the original code is odd, the selection will increase it to the next even value, since two code words of the same weight can differ only in an even number of digits. For example, it is known that the number of weight-w words in the Hamming single-error-correcting ($d = 3$) code of length $n = 2^\nu - 1$, for any $\nu = 2, 3, 4, \cdots$, is equal to the coefficient of x^w in the polynomial [21]

$$P(x) = \frac{1}{n + 1} \{(1 + x)^n + n(1 - x)(1 - x^2)^{n-1/2}\}$$

$$= 1 + \frac{n(n - 1)}{6} x^3 + \frac{n(n - 1)(n - 3)}{24} x^4 + \cdots.$$

Thus, all $N = n(n - 1)/6$ code words of weight $w = 3$ can be used for a ZFD_m code of length n. Since the distance of the constant-weight portion of this code is now $d = 4$, the order of the code, from (10), is at least $m = 2$.

Unfortunately, most group codes do not lead to interesting ZFD_m codes, because of the property of group codes that the distance equals the weight of the minimum-weight nonzero code word; thus, $d \leq w$. If this weight is even, then (10) gives (for $w > 1$)

$$m \geq \left\lceil \frac{w - 1}{w - \frac{w}{2}} \right\rceil = \left\lceil 2 - \frac{2}{w} \right\rceil = 1,$$

and if the weight is odd,

$$m \geq \left\lceil \frac{w-1}{w-\frac{w+1}{2}} \right\rceil = 2.$$

While these are only lower bounds on m, they can be expected to be close to the actual order, unless the number N of weight-w code words is very much less than the bound (3) would indicate may be possible. Therefore, constant-weight *ZFD* codes of large order must be generated either from one of the few known nonsystematic codes, or from a method other than selection from classical binary error-correcting codes. The *ZFD* codes constructed below are derived from q-nary error-correcting codes and from the block designs of statistics.

B. Codes Based on q-nary Codes

A q-nary error-correcting code is a code whose code-word digits are members of a set of q basic symbols [20]. If $q = 2$, we have a binary code, and the symbols 0 and 1 are generally used. However, our main interest in this section is with values of q greater than two. Many q-nary codes are known which have various lengths n_q and various q-nary distances d_q (minimum number of differing q-nary digits between any pair of code words) [20], [22].

We intend to form a binary superimposed code from a q-nary code by replacing each q-nary symbol by a unique binary pattern. To simplify the discussion, assume initially that each of the q binary patterns has unit weight and length q. Thus, the q-nary symbols 0, 1, $\cdots$ $q-1$ are to be replaced by the q-digit binary vectors $100\cdots0$, $010\cdots0$, $\cdots$, $000\cdots1$, respectively. (The generalization to other binary patterns will be described in the next subsection.) A q-nary code of length n_q is therefore transformed into a binary code of length

$$n = qn_q \tag{11}$$

and the binary distance is twice the q-nary distance: $d = 2d_q$. The number $N = N_q$ of code words remains the same. Since the binary code has constant weight $w = n_q$ (one *one* per q-nary digit), its *ZFD* order is given by (10), and is

$$m \geq \left\lfloor \frac{n_q - 1}{n_q - d_q} \right\rfloor.$$

In the interests of maximizing m for fixed length n_q and size N_q, we seek q-nary codes whose distance is as large as possible. A study of maximal-distance q-nary codes has revealed several code families, and some interesting special properties, when the code is *separable*—that is, when the number n_q of digits can be separated into k_q (independent) information digits and $r_q = n_q - k_q$ (dependent) check digits. These results have been reported in a separate paper [22]. In particular, it has been shown that the distance is bounded according to

$$d_q \leq r_q + 1,$$

so that for *maximal-distance separable* (MDS) q-nary codes, for which $d_q = r_q + 1$, the maximum order is

$$m = \left\lfloor \frac{n_q - 1}{k_q - 1} \right\rfloor. \tag{12}$$

Equality in this expression follows directly from Theorem 6, and the observation that each $\lambda_{\max} = (k_q - 1)$-tuple is repeated just q times. Also, the k_q independent digits imply a total of

$$N_q = N = q^{k_q} \tag{13}$$

code words in the code. These three relations, (11), (12), and (13), therefore relate the parameters q, k_q, and n_q of MDS q-nary codes to the parameters n, N, and m of the binary superimposed codes derivable from them.

MDS q-nary codes are known to exist for several ranges of parameter values [22], but the most useful family for present purposes is the set for which q is any prime power (≥ 3), and which uses any values of k_q and n_q that satisfy

$$q + 1 \geq n_q \geq k_q + 1 \geq 3. \tag{14}$$

In the conversion of these codes to *ZFD* codes, we may note from (12) that for prescribed m, and for any particular values of q and k_q, the use of a length n_q larger than $1 + m(k_q - 1)$ serves only to increase n while N and m remain constant. With this minimum value of n_q, therefore, the parameters of the *ZFD* code family are (for $k_q \geq 2$):

$$\left.\begin{aligned} n &= q\{1 + m(k_q - 1)\} \\ N &= q^{k_q} \end{aligned}\right\} \tag{15}$$

where q is any prime power, and $q \geq m(k_q - 1) \geq 3$. (The inequality (14) is now satisfied automatically.) We have therefore demonstrated the existence of *ZFD* codes of arbitrarily large size and order, and whose size N grows exponentially with length n, for fixed order m.

This lower bound on q governs the minimum size of these *ZFD* codes; for example, for $k_q = 5$, then $q \geq 4m$, and

$$n \geq 4m(1 + 4m)$$

$$N \geq (4m)^5$$

so that q-nary-based codes with $k_q \geq 5$ are extremely large—certainly too large to be of much interest for the types of applications discussed in Section II. Even for $k_q = 4$, reasonably sized codes exist only when the maximum order m is small (2 or 3). For the other cases, we have

$k_q = 2$:	$k_q = 3$:
$n = q(1 + m)$	$n = q(1 + 2m)$
$N = q^2$	$N = q^3$
$q \geq m$	$q \geq 2m$.

When $k_q = 2$, codes are known for which q is not restricted to be a prime power, but the range of nq is now reduced from that given by (14) to

$$L(q) + 2 \geq n_q \geq 3$$

where $L(q)$ is the number of pairwise orthogonal Latin squares of order q. Again, using the minimum value of n_q, the same expressions for n and N result, but now the integer q must be chosen large enough so that

$$L(q) \geq m - 1.$$

It is known that $L(q)$ is no greater than $q - 1$, and is at least as great as one less than the smallest prime-power factor contained in q [23]; *e.g.*, $L(12) = L(2^2 \cdot 3) \geq 2$, and $L(12) \leq 11$. When q is itself a prime power, these limits are equal, and the bound stated earlier ($q \geq m$) results.

When $k_q = 2$ and $m = 2$, then $n_q = 3$ and only one Latin square is needed; any value of $q \geq 3$ is satisfactory as a basis for the resulting weight-three ZFD_2 code having $n = 3q$ and $N = q^2$.

The construction of MDS q-nary codes is described in Singleton's paper [22]. Suffice it to note at present that the family presented above includes as special cases the Reed-Solomon [24] q-nary codes ($n_q = q - 1$), q-nary "parity-check" codes ($r_q = 1$), simple repetition codes ($k_q = 1$), part of the family of Golay [25] single-error-correcting q-nary codes ($r_q = 2$, $n_q = q^2 - 1$), and several codes based on orthogonal Latin squares ($k_q = 2$) [26].

These q-nary-based ZFD codes are certainly inefficient in one respect, in that they are *split-field* codes; that is, each code word's binary digit sequence, or *field*, can be separated into distinct sections (the sections have the same lengths for all code words) which are encoded separately. Each of the w sections has length n_q and contains a single *one*. In general, such a code may then be augmented with additional words, without decreasing its distance (hence its order), by letting the number of *ones* in each section increase above unity. For example, the ZFD_2 code based upon the ternary code with $k_q = 2$, $n_q = 3$, has the $N = q^2 = 9$ code words

0 0 1 0 0 1 0 0 1
0 0 1 0 1 0 1 0 0
0 0 1 1 0 0 0 1 0
0 1 0 0 0 1 1 0 0
0 1 0 0 1 0 0 1 0
0 1 0 1 0 0 0 0 1
1 0 0 0 0 1 0 1 0
1 0 0 0 1 0 0 0 1
1 0 0 1 0 0 1 0 0.

Without increasing the length $n = 3q = 9$, or decreasing m, three more code words may be added:

1 1 1 0 0 0 0 0 0
0 0 0 1 1 1 0 0 0
0 0 0 0 0 0 1 1 1

yielding a ZFD_2 code of size $N = 12$.

C. Codes Based on Composition with q-nary Codes

It was assumed in the last Section V-B that each digit of the q-nary code was represented as a weight-one binary q-tuple. However, there is no reason why a more general representation of these q symbols cannot be used, provided only that any set of up to m different such symbol representations has a superposition sum which itself satisfies the ZFD_m property. Thus, q words from any ZFD_m code containing at least q words may be used. Since such a code may have a word length less than q (the length of the weight-one ZFD_m code used previously), the total number n of binary digits necessary for the q-nary-derived superimposed code may be much less than qn_q, the earlier value.

This type of q-nary symbol representation may be advantageously regarded as a method of composition, in which a small ZFD code, having parameters n_0, N_0, and m_0, say, may be converted into a larger ZFD code, having parameters n_1, N_1, and m_1, on the basis of an n_q-digit q-nary code having k_q independent digits. The relations between these parameters are direct extensions of (11), (12), and (13):

$$n_1 = n_0 n_q$$
$$N_1 = q^{k_q}$$
$$m_1 = \min(m_0, m_q),$$

where q is a prime power now bound by the inequality $n_q - 1 \leq q \leq N_0$, and

$$m_q = \left[\frac{n_q - 1}{k_q - 1}\right].$$

Using a value of n_q no larger than necessary to render $m_0 \geq m_q = m_1 = m$, we get

$$\left.\begin{aligned} n_1 &= n_0\{1 + m(k_q - 1)\} \\ N_1 &= q^{k_q} \end{aligned}\right\} \tag{16}$$

where

$$m(k_q - 1) \leq q \leq N_0.$$

The choice of a weight-one code for the smaller ZFD code means that $n_0 = N_0 = m_0 = q$, and yields the code family (15) derived in Section V-B.

Starting with a simple weight-one code, repeated compositions can be carried out, keeping the order m fixed, to build up arbitrarily large ZFD codes.[3] Different q-nary codes may be used at each stage of the composition. If the same type of q-nary code is used (except for the value of q itself, which is replaced by $q' = N_1$), then a second composition on the code (16) yields directly

$$\left.\begin{aligned} n_2 &= n_0\{1 + m(k_q - 1)\}^2 \\ N_2 &= q^{k^2 q} \end{aligned}\right\}$$

[3] Of course the original code for repeated composition need not be a weight-one code, or even a q-nary code, but can be any ZFD code with the proper parameters. The block design codes of Section V-D can serve as particularly good original codes.

where

$$m(k_q - 1) \leq q' = N_1 = q^{k_q}.$$

(Clearly, if q is a prime power, then $q' = q$ is also a prime power.) After c such compositions on a weight-one original code, there results

$$\left.\begin{aligned} n_c &= q\{1 + m(k_q - 1)\}^c \\ N_c &= q^{k^c q} \end{aligned}\right\} \tag{17}$$

where, as before, the prime power q must satisfy $q > m(k_q - 1)$.

The number c of compositions may be optimized with respect to q and k_q by noting that replacement of c by $c - 1$ can be compensated for by replacing q by q^{k_q}, to keep N_c constant. This substitution changes the length to $q^{k_q}\{1 + m(k_q - 1)\}^{c-1}$, which represents an increase (c too small) or decrease (c too large), depending on whether

$$\frac{q^{k_q - 1}}{1 + m(k_q - 1)}$$

is greater or less than unity, respectively. Thus, for given m and N, q should be selected in accordance with not only a lower bound, but now an upper limit as well:

$$1 + m(k_q - 1) \leq q^{k_q - 1} \leq 1 + m(k_q - 1)^{k_q}.$$

For $k_q = 2$, this range becomes

$$1 + m \leq q \leq (1 + m)^2,$$

and for $k_q = 3$,

$$(1 + 2m)^{1/2} \leq q \leq (1 + 2m)^{3/2}.$$

(In this last case, the lower limit is satisfied automatically, since $q \geq 2m$.)

When $k_q = 2$, this composition method is valid even when q is not a prime power, provided only that $L(q) \geq m - 1$, as before. The validity follows directly from the fact that

$$L(q^2) \geq L(q),$$

an inequality which may be established without difficulty on the basis of the following construction.[4] Let the set of L pairwise orthogonal Latin squares of order q be $S_1, \cdots S_k \cdots S_L$, written as matrices in the symbols $0, 1, 2, \cdots q - 1$, with general element $s_{ij}^{(k)} \cdots$. Then a set of L pairwise orthogonal Latin squares $T_1, \cdots, T_k \cdots T_L$ of order q^2 and of general partitioned form

$$T_k = \begin{bmatrix} T_{11}^{(k)} & T_{12}^{(k)} & \cdots \\ T_{21}^{(k)} & T_{22}^{(k)} & \\ \vdots & & \end{bmatrix},$$

where $T_{ij}^{(k)}$ is a q-by-q array, can be constructed by letting

$$T_{ij}^{(k)} = S_k + qs_{ij}^{(k)} J,$$

where J is a q-by-q array of all *ones*.

[4] This proof is due to B. Elspas.

D. Codes Based on Block Designs

The block designs of statistics constitute a multi-parameter family of arrangements of objects, which for present purposes may be conveniently represented as matrices of *zeros* and *ones*. The *incidence matrix* S of a so-called *balanced incomplete block design* (BIBD) with parameters (v, k, b, r, λ) has b rows, v columns, k *ones* per row, and r *ones* per column, and is such that the dot product of every pair of columns is just λ. The well-known identities

$$vr = bk$$

$$k(r - 1) = \lambda(v - 1)$$

must be satisfied.

Either the rows or columns of S might be identified with the code words of a constant-weight code. If each *column* of S is a code word, then we have

$$A = S^t \ (= \text{the transpose of } S),$$

so that

$$n = b$$

$$N = v$$

$$w = r$$

$$\lambda_{ij} = \lambda = \lambda_{\max}.$$

But $v \leq b$ for a BIBD, and thus $N \leq n$, yielding an uninteresting family of superimposed codes.

If each *row* of S is regarded as a code word, then

$$A = S,$$

so that

$$n = v$$

$$N = b$$

$$w = k$$

$$\lambda_{ij} \leq \mu_{\max}$$

where $\mu_{\max}$ is the maximum dot product of any pair of *rows* of S. Hence

$$m \geq \left[\frac{w - 1}{\mu_{\max}}\right] = \left[\frac{k - 1}{\mu_{\max}}\right].$$

Unfortunately, there is no simple relationship between $\lambda_{\max}$ and $\mu_{\max}$ for BIBD's in general. If $\lambda = 1$, then it may easily be shown that $\mu_{\max} = 1$, so that[5]

$$m = w - 1 = k - 1,$$

but if $\lambda > 1$, then all that can be said in general is that $2 \leq \mu_{\max} \leq k$.

The theory of block designs is incomplete, although constructions are known for a number of families and for some isolated designs [27], [28]. Unfortunately, the parameter values of practical interest in forming super-

[5] Equality and maximality of this value of m follow directly from Theorem 6.

imposed codes are beyond the range of most of the designs tabled for statistical use. The principal exceptions to this situation occur for $\lambda = 1$, for which useful designs are known with the parameters,

$$(v, k, r, b, \lambda) = \left(n, w, \frac{n-1}{w-1}, N = \frac{n(n-1)}{w(w-1)}, 1\right);$$

specifically, for

$k = 3$: $v = 1$ or $3 \pmod 6$, $b = r(2r+1)/3$, $r = (v-1)/2$; so that $w = 3$, $n = 1$ or $3 \pmod 6$, $N = n(n-1)/6$, $m = 2$

$k = 4$: $v = 3r - 1$, $b = rv/4$, $r =$ prime power; so that $w = 4$, $(n-1)/3 =$ prime power, $N = n(n-1)/12$, $m = 3$.

Note that these codes achieve the bound (5), and therefore cannot be made larger for the same length and the same weight. (In fact, it can be shown that any ZFD_m code achieving this bound is equivalent to a BIBD.) The designs for the $k = 3$ family are called *Steiner Triple Systems* [27], and have had a previous application to coding problems [26].

VI. Construction of UD Codes

A. UD Codes Based on Parity-Check Matrices

While *UD* codes can certainly be obtained by using *ZFD* codes of the same order (see Theorems 3 and 4), it may be possible to take advantage of the less stringent defining condition expressed in Theorem 4 to obtain *UD* codes which are larger than *ZFD* codes of the same order and length. Presented below are three different approaches to the construction of *UD* codes of small order. Some of these turn out to be quite efficient.

The transpose H^t of the parity check matrix H of a conventional binary e-error-correcting code is known to have the property that the set composed of its row vectors and all sums of up to e of them contains no duplicates [29]. This property is exactly what is desired for the A matrix of a UD_e code, except for the type of summation involved: the H^t matrix is based on modulo-2 (exclusive-OR) addition while the A-matrix is based on Boolean (inclusive-OR) addition. Consequently, H^t cannot be used directly as an A-matrix with $m = e$, but we might profitably seek some way to modify H^t so that uniqueness of these row sums is preserved, even under Boolean addition.

We demonstrate below such modifications for $e = 2$ and $e = 3$, yielding UD_2 and UD_3 code families, respectively.

For $e = 2$, let each binary digit in H^t be accompanied in the same row by its complement; e.g.,

$$0 \rightarrow 01$$

$$1 \rightarrow 10.$$

This substitution can be effected by using, for example, the matrix

$$A = [H^t \vdots \bar{H}^t],$$

in which $\bar{H}^t$ is the binary complement of H^t. The addition tables for elements of the H^t and A matrices may now be compared,

$\oplus$	0	1
0	0	1
1	1	0

$\vee$	01	10
01	01	11
10	11	10

Clearly, any pairwise row sum of A can be unambiguously transformed back to the corresponding row sum of H^t:

$$00 \rightarrow 0$$

$$10 \rightarrow 0$$

$$11 \rightarrow 1.$$

Similarly, any row of A can also be uniquely transformed to a row of H^t:

$$01 \rightarrow 0$$

$$10 \rightarrow 1.$$

The dual interpretation of 10 will give rise to no ambiguities, as long as a *row* of A can be distinguished from a *row sum*. This is indeed the case, since no row of A contains 11, but every row sum contains 11 as evidence of differing digits in at least one digit position.

Since uniqueness of rows and row sums is preserved, the matrix A represents a UD_2 code if the matrix H^t represents a 2-error-correcting code. The family of Bose-Chaudhuri codes [30] for $e = 2$ have at most 2μ check digits (number of columns of H^t) and a total of $2^\mu - 1$ digits (number of rows of H^t), for all positive integer values of $\mu \geq 2$. Since A has twice as many columns as H^t, then

$$n \leq 4\mu$$

$$N = 2^\mu - 1.$$

Therefore, for every doubly even value of n, a UD_2 code exists of size

$$N = 2^{n/4} - 1.$$

The exponential growth of these codes guarantees that they will be larger than all previously derived ZFD_2 (hence UD_2) codes, for sufficiently large values of n.

For $e = 3$, intercolumn relationships of H^t must be somehow represented in A, since any form of simple substitution such as $0 \rightarrow \alpha$, $1 \rightarrow \beta$ is not adequate to maintain a distinction between all of the double and triple sums:

$$\left.\begin{aligned} 0 \oplus 0 \oplus 1 &= 1 \\ 0 \oplus 1 \oplus 1 &= 0 \end{aligned}\right\} \text{ but } \alpha \vee \alpha \vee \beta = \alpha \vee \beta \vee \beta;$$

$$\left.\begin{aligned} 1 \oplus 1 &= 0 \\ 1 \oplus 1 \oplus 1 &= 1 \end{aligned}\right\} \text{ but } \beta \vee \beta = \beta \vee \beta \vee \beta.$$

It is possible to show that if *every pair* of columns of H' is recoded in A according to the transformation

$$00 \rightarrow 1000$$
$$01 \rightarrow 0100$$
$$10 \rightarrow 0010$$
$$11 \rightarrow 0001,$$

then the resulting A-matrix represents a UD_3 code if the H'-matrix represents a triple-error-correcting code. The Bose-Chaudhuri codes [30] for $e = 3$ have a matrix H' with $2^\mu - 1$ rows and no more than 3μ columns, for every positive integral value of $\mu \geq 3$. Since a pair of columns may be selected from H' in $\binom{3\mu}{2}$ ways, the number of columns of A is

$$n \leq 4\binom{3\mu}{2} = 6\mu(3\mu - 1)$$

and the number of rows is

$$N = 2^\mu - 1.$$

This code is inefficient for relatively small values of n and N, since $N > n$ only for $\mu \geq 13$ ($n \geq 2964$), but it is asymptotically attractive:

$$N > 2^{\sqrt{n/18}}.$$

This growth rate is about the same as occurred for the q-nary ZFD_3 codes obtained by iterated composition with $k_a = 2$ and $q = 1 + m = 4$:

$$N = 2^{\sqrt{n}}.$$

B. Codes of Weight Two Based on a Graphical Construction

The best codes of constant weight $w = 2$ can not be ZFD_2 codes, according to Theorem 6, but may be UD_2 codes. We derive below two such code families, and show that their size N grows asymptotically as $n^{3/2}$.

Consider first the split-field case, when each of the *ones* is confined to a separate portion of the n-digit code word. Let a given code word have its two *ones* in the ith digit position of the left portion and the jth digit position of the right portion. The entire code may then be expressed compactly in the form of a binary matrix G, whose general entry g_{ij} has the value 1 when and only when the code contains such a code word having *ones* in the ith and jth digit positions of the left and right portions, respectively. Clearly, the size N of the code equals the total number of *ones* in G.

To satisfy the UD_2 condition, no two *ones* in G must occupy the same pair of rows and columns as two other *ones*; that is, no row of G can contain a pair of *ones* in the same two positions as another row. Thus, we seek for G a binary matrix with a fixed semiperimeter n, and containing a maximum number N of *ones*, such that the dot product of any two rows does not exceed unity.

This requirement will be met by the matrix of a BIBD [27] whose parameters (v, k, b, r, λ) are given by:

$$n = v + b$$
$$N = kr$$
$$\lambda = 1.$$

To the extent that these designs exist, N will be maximized for fixed n when the ratio $v/b = k/r$ is as near unity as possible. In the case of complete regularity, therefore, G must be the matrix of a *symmetrical* BIBD: $v = b$ and $k = r$, thus $n = 2v$ and $N = k^2$. These symmetrical designs are known to exist for all values of k for which $k - 1$ is a prime power [27], and from the block design identities they yield the relations:

$$n = 2(k^2 - k + 1)$$
$$N = k(k^2 - k + 1).$$

For these values of k, then, there exist split-field UD_2 codes of weight two and of size

$$N = \frac{n}{4}(1 + \sqrt{2n - 3}).$$

Asymptotically,

$$N \sim \frac{n^{3/2}}{2\sqrt{2}}.$$

Consider next the case when the two *ones* are not restricted to separate portions of the code word. Let each of the n digit positions of the code words now be represented as a node of an n-node graph. Each code word may then correspond to an undirected branch between the two nodes which represent the positions of its two *ones*. In these terms, we wish to place on an n-node graph a maximum number N of branches, subject only to a certain condition which corresponds to the desired UD_2 property: no branch-pair may be incident on the same set of nodes as another branch-pair. Thus, neither of the partial graphs in Fig. 1 is allowed. Cycles of lengths

Fig. 1.

4 and 3 can therefore be excluded, and 2-cycles (duplicate code words) and 1-cycles (weight-one code words) can be ruled out as needlessly wasteful. Therefore, we seek maximal n-node graphs which contain no *closed cycles of length shorter than 5*. Sufficiency of this condition is obvious: every n-node graph whose shortest cycle length is at least 5 generates a UD_2 code of weight 2.

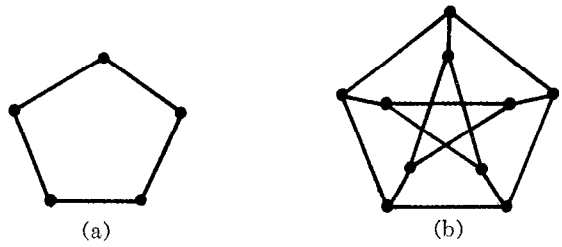
Completely regular graphs of this type have been studied previously by A. J. Hoffman and R. R. Singleton [31], and are called "Moore graphs of diameter 2".[6]

[6] The pertinence of the Hoffman-Singleton paper to the present problem was suggested by E. F. Moore.

In terms of the *degree* t of the graph—that is, the number of branches incident on each node—certain equalities must be satisfied which rule out all but four possibilities:

$$t = 2 \quad n = 5 \quad N = 5$$
$$t = 3 \quad n = 10 \quad N = 15$$
$$t = 7 \quad n = 50 \quad N = 175$$
$$t = 57 \quad n = 3250 \quad N = 92{,}625.$$

The first two graphs are shown in Fig. 2, the third is



(a) (b)

Fig. 2.

listed in Hoffman's paper, and the fourth case is undecided. The code parameters are related to the degree by:

$$n = 1 + t^2$$
$$N = \frac{nt}{2} = \frac{t(1 + t^2)}{2}$$

so that

$$N = \frac{n\sqrt{n-1}}{2}.$$

For values of n intermediate between those listed above, the next larger complete graph may be pruned, one node with its incident branches at a time, to remove the least number of branches at each step. The sizes of some of these intermediate codes are listed in Table I. In any case, we have the approximate asymptotic growth,

$$N \sim \frac{n^{3/2}}{2}$$

which is slightly better than for the corresponding split-field UD_2 codes, but is still poorer than the growth for the UD_2 codes which are based on parity-check matrices of conventional double-error-correcting codes, and which are presented in the previous subsection. The codes of Table I are also much poorer than the simplest q-nary and block-design-based ZFD_2 codes of Section V.

TABLE I

n	N	n	N
5	5	30	70
10	15	35	95
15	25	40	120
20	35	45	145
25	50	50	175

C. Pairwise Composition of UD_2 Codes

It was shown in Section V how a three-section, split-field, ZFD_2 code can be formed from a known ZFD_2 code of one-third the length. The code words of the three-section code have the form

$$(a_1)(b_1)(c_{11})$$
$$(a_2)(b_2)(c_{22})$$
$$\text{etc.},$$

in which the partial words $a_1, a_2, \cdots$ and $b_1, b_2, \cdots$ are selected independently as code words of the smaller code. The third partial words, $c_{11}, c_{22}, \cdots$ are selected from the same code in accordance with a certain Latin square, whose row and column indices are related to the first and second partial words. Thus, from a given ZFD_2 n-digit code having N words we may compose a new ZFD_2 code having $3n$ digits and N^2 words.

We will now show that large UD_2 codes may be similarly composed from smaller UD_2 codes, the only difference being that the length of the third field is considerably less than that required in the ZFD_2 case. Equivalently, UD_2 codes can be formed whose size N is much greater then ZFD_2 codes of the same length.

If the partial words $a_1, a_2, \cdots$ and $b_1, b_2, \cdots$ are selected from a UD_2 code, then the first and second fields of the superposition sum,

$$(a_1 \vee a_2)(b_1 \vee b_2)(c_{11} \vee c_{22})$$

can certainly be individually deciphered into their constituents. Without a suitable third field, however, the two interpretations

$$(a_1)(b_1)(c_{11}) \quad \text{and} \quad (a_1)(b_2)(c_{12})$$
$$(a_2)(b_2)(c_{22}) \qquad\qquad (a_2)(b_1)(c_{21})$$

cannot be distinguished. We therefore require that

$$c_{11} \vee c_{22} \neq c_{12} \vee c_{21}.$$

This condition can be expressed more naturally by arranging the entire set of third-section partial words c_{ij} into a N-by-N matrix C, just as was done for the Latin square. The row and column indices correspond to the selection of the partial words a_i and b_j, respectively. Thus, each element in the matrix C is the third section of one of the N^2 code words being derived. The above condition now reads

$$c_{ij} \vee c_{kl} \neq c_{il} \vee c_{kj}, \qquad i \neq k, \quad j \neq l$$

for every set of four elements which form a rectangle in the matrix. That is to say, opposite diagonal sums must be different for each 2×2 minor of C.

If c_{ij} is limited to a single binary digit, the largest C-matrix meeting this condition is readily seen to be a 3×3 identity matrix:

$$C_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Starting with a 3-digit, weight-one code (which is certainly UD_2), having the three code words

$$\begin{matrix} 0\ 0\ 1 \\ 0\ 1\ 0 \\ 1\ 0\ 0, \end{matrix}$$

the matrix C_1 yields a 7-digit UD_2 code having $3^2 = 9$ words

$$\begin{matrix} 0\ 0\ 1 & 0\ 0\ 1 & 1 \\ 0\ 0\ 1 & 0\ 1\ 0 & 0 \\ 0\ 0\ 1 & 1\ 0\ 0 & 0 \\ 0\ 1\ 0 & 0\ 0\ 1 & 0 \\ 0\ 1\ 0 & 0\ 1\ 0 & 1 \\ 0\ 1\ 0 & 1\ 0\ 0 & 0 \\ 1\ 0\ 0 & 0\ 0\ 1 & 0 \\ 1\ 0\ 0 & 0\ 1\ 0 & 0 \\ 1\ 0\ 0 & 1\ 0\ 0 & 1. \end{matrix}$$

We seek next a 9 by 9 matrix C_2 whose entries c_{ij} satisfy the above minor diagonal condition. In general, we need to convert a 3^p-by-3^p matrix C_p into a 3^{p+1}-by-3^{p+1} matrix C_{p+1}, $p = 1, 2, \cdots$, in such a way that C_{p+1} satisfies the minor condition if C_p does. To this end, suppose that such a C_p satisfies this condition and has a partition into ninths of the form

$$C_p = \begin{bmatrix} X & Y & Z \\ Z & X & Y \\ Y & Z & X \end{bmatrix},$$

where X, Y, and Z are 3^{p-1}-by-3^{p-1} submatrices with vector elements. C_1 certainly has this partition structure. Let the notation $1X$ designate a matrix X, all of whose vector entries are augmented (on the left end, say) with a binary 1; similarly for $0X$ $1Y$, $0Y$, $1Z$, and $0Z$. We will now show that the matrix

$$C_{p+1} = \left[\begin{array}{ccc|ccc|ccc} 11X & 10Y & 10Z & 00X & 01Y & 00Z & 00X & 00Y & 01Z \\ 10Z & 11X & 10Y & 00Z & 00X & 01Y & 01Z & 00X & 00Y \\ 10Y & 10Z & 11X & 01Y & 00Z & 00X & 00Y & 01Z & 00X \\ \hline 00X & 00Y & 01Z & 11X & 10Y & 10Z & 00X & 01Y & 00Z \\ 01Z & 00X & 00Y & 10Z & 11X & 10Y & 00Z & 00X & 01Y \\ 00Y & 01Z & 00X & 10Y & 10Z & 11X & 01Y & 00Z & 00X \\ \hline 00X & 01Y & 00Z & 00X & 00Y & 01Z & 11X & 10Y & 10Z \\ 00Z & 00X & 01Y & 01Z & 00X & 00Y & 10Z & 11X & 10Y \\ 01Y & 00Z & 00X & 00Y & 01Z & 00X & 10Y & 10Z & 11X \end{array}\right],$$

which has the same partition structure as does C_p, also satisfies the minor diagonal condition.

First of all, note that the X, Y, and Z portions of the binary vector entries c_{ij} in each of the ninths of C_{p+1} are the same within each ninth. Hence, any 2×2 minor falling entirely within one of the ninths will certainly satisfy the condition. In fact, any 2×2 minor whose corners fall in different ninths will also satisfy the condition for the same reason, except perhaps if its horizontal or vertical corner pairs fall in *corresponding* rows or columns of different ninths. In these cases, however, the added digits serve to keep the condition satisfied, by providing a digit pattern over these corresponding positions exactly as was used in C_1. The first added digit handles the case when the four corners of the minor fall at corresponding locations in four different ninths. The second added digit handles the case when the minor lies entirely within a line of three adjacent ninths, but its left and right corner-pairs (or top and bottom corner-pairs) fall in corresponding columns (rows, respectively) of these three ninths.

As a result, all minors satisfy the diagonal condition, and C_{p+1} is a satisfactory matrix for a UD_2 code.

With each increase of p by one, two binary digits are added to c_{ij}; thus, the entries in C_p are $2p - 1$ binary digits in length. The UD_2 code obtained by iterated composition therefore has, for each positive integral value of p, a size N and a length $n(p)$ given by

$$N = 3^{2^p}, \qquad n(p) = 2n(p-1) + (2p-1),$$

or

$$n(p) = 6 \cdot 2^p - (2p+3).$$

Asymptotically, then,

$$N \sim 3^{n/6}.$$

The ZFD_2 codes obtained by iterated composition also had $N = 3^{2^p}$, but for them, $n(p) = 3n(p-1)$, so $n(p) = 3^{p+1}$. For these values of p, then,

$$N = 3^{1/2(n)\log_3 2},$$

which is much less than the corresponding value of N for the UD_2 codes.

VII. Discussion

We have shown in Sections I–VI how a new class of codes, nonrandom binary superimposed codes, may be used in storage and communication systems, and we have derived for these codes several properties and construction methods over a wide range of parameter values. Not considered in this first investigation of ZFD and UD codes are problems associated with their implementation in encoding and decoding logical circuitry, and the formation of truly optimal codes. Also, it would sometimes be useful to be able to use part of the distance of the codes for noise protection, even if the order of the

code must be reduced to do so, and to determine the trade-off between the degree of error-detection and the order.

In the theoretical area, better upper bounds on N as a function of n and m would be desirable, as would a better undertanding of the inner relationship between *ZFD* and *UD* codes.

Comparison of known *ZFD* codes with one another reveals that the largest short codes are based on block designs, and the largest longer codes are based on q-nary error-correcting codes. Since the block-design codes all have fixed weight, these results suggest that block-design codes of large weight, if they exist and could be found, would turn out to be superior. Indeed, the fact that q-nary-based superimposed codes are split-field codes, and can be augmented in almost every case, indicates an avoidable inefficiency that could be overcome with a more uniform distribution of *ones* throughout the code word, such as occurs in block-design codes.

A comparison of *ZFD* and *UD* codes with random superimposed codes suffers from the same difficulties that are encountered in comparing deterministic and random conventional error-correcting codes. Some sort of channel statistics (here, descriptor usage statistics) must be assumed, in order that a set of quantitatively related error (false-drop) probabilities may be assigned to the occurrence of the various numbers of different types of errors (here, the numbers of quiz and document descriptors). From the point of view of actually carrying out the comparison analytically or computationally, the situation is further complicated in the case of superimposed codes by the unavoidable dependence of the result on additional parameters: the size of the file, and the ratio between the numbers of quiz and document descriptors. Also, in the retrieval application, the meaningfulness of the result is liable to depend rather critically on some assumptions which are not at all met in practice (equal descriptor usage, and lack of interdescriptor correlation).

Acknowledgment

The authors are deeply grateful to Dr. Bernard Elspas, whose early interest and efforts in the study of the class of codes presented in this paper have contributed materially to the over-all investigation. He also aided directly in some of the proofs and results on constant-weight and q-nary codes.

References

[1] C. K. Schultz, "An application of random codes for literature searching," in "Punched Cards, Their Applications to Science and Industry," R. S. Casey, *et al.*, Eds., Reinhold Publishing Corp., New York, N. Y., ch. 10, see also chs. 18 and 23; 1958.
[2] C. N. Mooers, "The Application of Simple Pattern Inclusion Selection to Large-Scale Information Retrieval Systems," Rome Air Development Center, Rome, N. Y., Rept. No. RADC-TN-59-157, Zator Technical Bulletin No. 131; April, 1959. See Bulletin No. 120 for additional references to Zatocoding.
[3] M. Taube, "Superimposed Coding for Data Storage," Documentation, Inc., Washington, D. C., Tech. Rept. No. 15; September, 1956.
[4] E. H. Frei and J. Goldberg, "A method for resolving multiple responses in a parallel search file," IRE Trans. on Electronic Computers, vol. EC-10, pp. 718–722; December, 1961.
[5] C. W. Brenner and C. N. Mooers, "A case history of a zatocoding information retrieval system," in "Punched Cards, Their Application to Science and Industry," R. S. Casey, *et al.*, Eds., Reinhold Publishing Corp., New York, N. Y., ch. 15; 1958.
[6] J. Goldberg, *et al.*, "Multiple Instantaneous Response File," Stanford Research Institute, Menlo Park, Calif., Final Rept., SRI Project 3101, Rept. No. RADC-TR-61-233; August, 1961.
[7] C. A. Rosen, "An approach to a distributed memory," *Proc. 1961 Symp. on the Principle of Self-Organizing Machines*, Pergamon Press, Inc., New York, N. Y., pp. 425–444; 1962.
[8] R. C. Singleton, "Random Selection Rates for Single-Field Superimposed Coding," Stanford Research Institute, Menlo Park, Calif., Suppl. A to Quarterly Rept. 4, Contract AF 30(602)–2142; November, 1960.
[9] C. S. Wise, "Mathematical analysis of coding systems," in "Punched Cards, Their Application to Science and Industry," R. S. Casey, *et al.*, Eds., Reinhold Publishing Corp., New York, N. Y., ch. 21; 1958.
[10] C. N. Mooers, "The Exact Distribution of the Number of Positions Marked in a Zatocoding Field," Zator Co., Boston, Mass., Zator Technical Bulletin No. 73; 1952.
[11] C. Orosz and L. Takacs, "Some probability problems concerning the marking of codes into the superposition field," *J. Documentation*, vol. 12, pp. 231–234; December, 1956.
[12] S. Stiassny, "Mathematical Analysis of Various Superimposed Coding Methods," IBM Research Center, Yorktown Heights, N. Y., IBM Res. Rept. No. RC-103; April, 1959.
[13] C. E. Shannon and W. Weaver, "Mathematical Theory of Communications," University of Illinois Press, Urbana; 1949.
[14] P. Elias, "Error-free coding," IRE Trans. on Information Theory, vol. IT-4, pp. 29–37; September, 1954.
[15] J. Costas, "Poisson, Shannon, and the radio amateur," Proc. IRE, vol. 47, pp. 2058–2068; December, 1959.
[16] D. L. Cohn and J. M. Gorman, "A code separation property," IRE Trans. on Information Theory (Correspondence), vol. IT-8, pp. 382–383; October, 1962.
[17] R. C. Minnick and R. L. Ashenhurst, "Multiple-coincidence magnetic storage systems," *J. Appl. Phys.*, vol. 26, pp. 575–579; May, 1955.
[18] ——, "Simultaneous matrix storage systems," *Proc. International Symp. on the Theory of Switching*, Harvard University Press, Cambridge, Mass., pt. II, pp. 144–148; April, 1957.
[19] R. C. Singleton, "Load-sharing core switches based on block designs," IRE Trans. on Electronic Computers, vol. EC-11, pp. 346–352; June, 1962. R. C. Minnick, "Magnetic core access switches," IRE Trans. on Electronic Computers, vol. EC-11, pp. 352–368; June, 1962. P. G. Neumann, "On the logical design of noiseless load-sharing matrix switches," IRE Trans. on Electronic Computers, vol. EC-11, pp. 369–374; June, 1962. See also the references and bibliographies of Singleton, Minnick, and Neumann.
[20] W. W. Peterson, "Error-Correcting Codes," Mass. Inst. Tech. Press, Cambridge, Mass., and John Wiley and Sons, Inc., New York, N. Y., p. 30 ff.; 1961.
[21] *Ibid.*, pp. 67–70.
[22] R. C. Singleton, "Maximum distance q-nary codes," IEEE Trans. on Information Theory, vol. IT-10, pp. 116–118; April, 1964.
[23] H. B. Mann, "Analysis and Design of Experiments," Dover Publications, Inc., New York, N. Y., chs. 7, 8; 1949.
[24] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *J. Soc. Indus. Appl. Math.*, vol. 8, pp. 300–304; June, 1960.
[25] M. J. E. Golay, "Notes on digital coding," Proc. IRE (Correspondence), vol. 37, p. 657; June, 1949.
[26] W. H. Kautz and B. Elspas, "Single-error-correcting codes for constant-weight data words," submitted to IEEE Trans. on Information Theory.
[27] M. Hall, Jr., "A survey of combinatorial analysis," in "Some Aspects of Analysis of Probability," I. Kaplansky, *et al.*, John Wiley and Sons, Inc., New York, N. Y.; 1958.
[28] W. G. Cochran and G. M. Cox, "Experimental Design," John Wiley and Sons, New York, N. Y., 1957.
[29] Peterson, *op. cit.*, p. 33.
[30] Peterson, *op. cit.*, p. 162 ff.
[31] A. J. Hoffman and R. R. Singleton, "On Moore graphs with diameters 2 and 3," *IBM J. Res. and Develop.*, vol. 4, pp. 497–504; November, 1960.