

Superimposed Codes

SRI PROJECT NO. EU-3101

ACC. NO. 5961

REPORT NO. QR2, Supplement A COPY NO. 4, 6

CLIENT U.S.A.F. Rome Air Development Center

CONTRACT NO. AF 30(602)-2142

TITLE THE ORGANIZATION OF A MEMORY SYSTEM  
FOR INFORMATION RETRIEVAL APPLICATIONS

AUTHOR C. P. Bourne

DATE ISSUED 6/60

CLASSIFICATION U

~~EXCLUDED FROM~~ Refer requests to: Rome Air Development Center, Air Research and  
Development Command, U.S.A.F., Griffiss Air Force Base, New York  
Attn: David W. Sharp, Capt.  
Contracting Officer RCKCI

101 0036629

U

3101

1 FEB 60



To: MIRF File - Project 3101 cc: Goldberg, Austad  
Green, Frei, Bourne

23 March 1960

From: Charles Bourne

Subject: References on Superimposed Codes

Brenner, C. "Experience in Setting Up and Using the Zatocoding System"  
Zator Tech. Bull. 26. (also published as Chapter 11 of Information  
Systems in Documentation, edited by Shera et al).

Documentation, Taube, M., "Superimposed Coding for Data Storage - with an  
Inc. Appendix of Dropping Fraction Tables", Sept. 1956. Tech. Report  
No. 15 under contract NONR-1305(00). (Also available from OTS  
as Report PB-121 345 and from ASTIA as Report AD-111 261, SRI  
Main Library as No. Z 715, SRI Doc. Center No. DE-3995.

IBM Product Luhn, H. P., "Superimposed Coding with the Aid of Randomizing  
Dev. Lab Squares for Use in Mechanical Information Searching Systems"  
(1956).

John Hopkins "Final Report on Machine Methods for Information Searching",  
University Welch Medical Library Indexing Project, Baltimore, Maryland,  
1955.

Mooers, C. "The Theory of Digital Handling of Non-numerical Information  
and its Implications to Machine Economics", paper presented  
at the March 1950 meeting of the ACM at Rutgers University)  
Zator Tech. Bull. No. 48.

"Logic of Selective Systems", (paper presented at the Am. Math.  
Soc. meeting in Washington D. C., April 1950; abstract published  
in the Bull. of the Am. Math. Soc. 56:349, July 1950) Zator  
Tech. Bull. No. 50.

"Coding, Information Retrieval, and the Rapid Selector" (letter  
to the editor, Am. Doc., Oct. 1950) Zator Tech. Bull. No. 57.

"Zatocoding for Punched Cards" (Math. Appendix included) 1950  
Zator Tech. Bull. No. 30.

"Zatocoding Applied to Mechanical Organization of Knowledge",  
Am. Doc. Jan. 1951, Zator Tech. Bull. No. 16.

"Scientific Information Retrieval Systems for Machine Operation-  
Case Studies in Design", 1951, Zator Tech. Bull. No. 66.

"The Exact Distribution of the Number of Positions Marked in  
a Zatocoding Field", 1952, Zator Tech. Bull. No. 73.

"Choice and Coding in Information Retrieval Systems", Transactions  
of the IRE-PGIT, PGIT-4: 112-113, Sept. 1954.

"Recent Developments in Zatocoding", 1955, Zator Tech. Bull.  
No. 101.

"Information Retrieval on Structured Content", 1955 (the complete paper is available as Zator Tech. Bull. No. 24. The abbreviated version was published in the book: Information Theory-Third London Symposium, edited by Cherry).

"Zatocoding and Developments in Information Retrieval", 1956 Zator Tech. Bull. 25. (also published in ASLIB Proc. Feb. 1956)

"The Application of Simple Pattern Inclusion Selection to Large-Scale Information Retrieval Systems," April 1959, Zator Tech. Bull. No. 131, AD-215 434.

"Information Retrieval Selection Study, Part 1. Extensions of Pattern Inclusion Selection," August 1959, Zator Tech. Bull. No. 133, AD-230 278.

"Information Retrieval Selection Study, Part 2, Seven System Models", Aug. 1959, AD 230 290.

British Patent Specification No. 681,902 (filed Sept. 3, 1948, Complete Specification published Oct. 29, 1952).

Canadian Patent Specification No. 534,926 (filed Sept. 3, 1943, Issued Dec. 25, 1956).

U. S. Patent No. 2,665,694 (date of filing Jan. 3, 1949, issued Jan. 12, 1954).

Ohlman, H.

"Subject-Word Letter Frequencies with Applications to Superimposed Coding", preprints of the Int'l. Conf. On Scientific Information, Washington D.C. (Nov. 1958).

Orosz, G.

"Some Probability Problems Concerning the Marking of Codes into the Superimposition Field," J. Doc. (Dec. 1956)

Wise, C.

"Multiple Word Coding vs Random Coding for the Rapid Selector; A Replay to C. Mooers," Am. Doc. (October 1952).

"Mathematical Analysis of Coding Systems", Punched Cards: Their Application to Science & Industry, edited by R. S. Casey and J. W. Perry (New York, Reinhold Press, 1951, pp. 276-302).



Reprinted from IEEE TRANSACTIONS ON INFORMATION THEORY

Vol. IT-10, Number 4, October 1964

Pp. 363-377

Copyright 1964, and reprinted by permission of the copyright owner

PRINTED IN THE U.S.A.

# 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,

$$\begin{array}{r} 011001 \\ \vee 010010 \\ \hline 011011 \end{array}$$

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,

```

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

```

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,

```

1 0 0
0 1 0
0 0 1,

```

clearly form a  $ZFD_2$  code, since no pairwise sum such 110 can logically include the other code word, 001. 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.<sup>1</sup>

### 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*,

$$\Lambda = AA^t,$$

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

$$s = \frac{\left[ \text{Min}_{i \neq j} (\lambda_{ii}) \right]}{\left[ \text{Max}_{i \neq j} (\lambda_{ij}) \right]}.$$

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

$$m \geq \left[ \frac{\left[ \text{Min}_{i \neq j} (\lambda_{ii}) - 1 \right]}{\left[ \text{Max}_{i \neq j} (\lambda_{ij}) \right]} \right]$$

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

<sup>1</sup> 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 Minniek 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 *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  $x = (x^1, x^2, \dots, x^n)$  and  $y = (y^1, y^2, \dots, y^n)$  is defined by:

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

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  $k = 2, 3, \dots, N$  the  $k$ th *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  $C_1, C_2, \dots, C_k, \dots$ , 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  $C_1, C_2, \dots, C_{m+1}$  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:

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

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

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

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

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

If only  $C_1, C_2, \dots, C_m$  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  $C_1, C_2, \dots, C_m$  are disjoint and  $C_m$  contains  $\binom{N}{m}$  elements,  $C_1, C_2, \dots, C_{m+1}$  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  $y_1 \vee y_2 \vee \dots \vee y_j$  of  $j \leq m$  code words is included in any other sum  $x_1 \vee x_2 \vee \dots \vee x_k$  of  $k \leq m$  code words, unless  $y_1, y_2, \dots, y_j$  all belong to the set of code words  $x_1, x_2, \dots, x_k$ . 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  $x_1 \vee x_2 \vee \dots \vee x_k$  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  $x_1 \vee x_2 \vee \dots \vee x_k$  of  $k \leq m$  code words includes no other code word  $y_i$ , then it cannot include a sum such as  $y_1 \vee \dots \vee y_j \vee \dots \vee y_i$  of  $j \leq m$  code words, unless  $y_1, y_2, \dots, y_j$  all belong to the set of code words  $\{x_1, x_2, \dots, x_k\}$ .

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



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

*Proof:*

- 1) If the sets  $C_1, C_2, \dots, C_{m+1}$  are disjoint, then a code word  $y_i$  can be included in the sum  $x_1 \vee x_2 \vee \dots \vee x_k$  for  $k \leq m$  only if  $y_i$  is one of the code words  $x_1, x_2, \dots, x_m$ , 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  $x_1 \vee x_2 \vee \dots \vee x_i = y_1 \vee y_2 \vee \dots \vee y_k$ . But if each  $y_i$  is one of the code words  $x_1, x_2, \dots, x_i$ , then we cannot have  $j < k$ , and thus  $C_1, C_2, \dots, C_{m+1}$  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  $C_1, C_2, \dots, C_m$  are disjoint, and  $C_m$  contains  $\binom{N}{m}$  different vectors.

*Proof:*

- 1) Suppose that the sets  $C_1, C_2, \dots, C_m$  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  $C_1, C_2, \dots, C_m$  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:

$$\begin{aligned} ZFD_m &\Rightarrow UD_m \Rightarrow ZFD_{m-1} \\ &\Rightarrow UD_{m-1} \Rightarrow \dots \Rightarrow ZFD_1 \Rightarrow UD_1. \end{aligned}$$

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

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

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  $C_1, C_2, \dots, C_m$ , and noting that this number cannot exceed the number of nonzero,  $n$ -digit binary numbers:

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

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*  $\lambda_{ij}$  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  $i$ th code word accounts for just  $\binom{w_i}{\lambda_{\max} + 1}$  of them. Summing over all  $N$  code words, then, we have the condition:

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

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

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

Moreover, if  $w_i \geq m \lambda_{\max} + 1$ , then the  $i$ th 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  $\lambda_{\max}$  positions. Thus, a code with minimum weight  $w_{\min}$  and maximum overlap  $\lambda_{\max}$  is  $ZFD_m$  for all  $m$  up to some value which satisfies

$$m \geq \left\lceil \frac{w_{\min} - 1}{\lambda_{\max}} \right\rceil, \quad (4)$$

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

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

$$A = AA^t,$$

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  $A$  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  $\lambda_{\max}$ -tuple appears in two or more code words of a code, this code is  $ZFD_m$  but not  $ZFD_{m+1}$  for

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

*Proof:* The code is at least  $ZFD_m$  by the bound (4). But if every  $\lambda_{\max}$ -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.<sup>2</sup>

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

corresponding to (2), namely,

$$\sum_{i=1}^{N_1} \left( \frac{w_i - 1}{\lambda_{\max} + 1} \right) + \sum_{i=N_1+1}^N \left( \frac{w_i}{\lambda_{\max} + 1} \right) \leq \left( \frac{n - N_1}{\lambda_{\max} + 1} \right).$$

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  $\lambda_{\max}$  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

$$N \leq \frac{\binom{n}{\lambda_{\max} + 1}}{\binom{w}{\lambda_{\max} + 1}} \quad (5)$$

and

$$m \geq \left\lceil \frac{w - 1}{\lambda_{\max}} \right\rceil. \quad (6)$$

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

$$N \leq \left\lceil \frac{n}{w} \left[ \frac{n-1}{w-1} \left[ \frac{n-2}{w-2} \left[ \dots \left[ \frac{n-\lambda_{\max}}{w-\lambda_{\max}} \right] \dots \right] \right] \right] \right\rceil; \quad (7)$$

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

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

<sup>2</sup> 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,

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

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 <sub>$m$</sub>  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 super-

imposed 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 <sub>$m$</sub>  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

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

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 <sub>$m$</sub>  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  $n = 2^r - 1$ , for any  $r = 2, 3, 4, \dots$ , is equal to the coefficient of  $x^w$  in the polynomial [21]

$$\begin{aligned} P(x) &= \frac{1}{n+1} \{ (1+x)^n + n(1-x)(1-x^2)^{n-1/2} \} \\ &= 1 + \frac{n(n-1)}{6} x^3 + \frac{n(n-1)(n-3)}{24} x^4 + \dots \end{aligned}$$

Thus, all  $N = n(n-1)/6$  code words of weight  $w = 3$  can be used for a ZFD <sub>$m$</sub>  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 <sub>$m$</sub>  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  $0, 1, \dots, q-1$  are to be replaced by the  $q$ -digit binary vectors  $100 \dots 0, 010 \dots 0, \dots, 000 \dots 1$ , 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

$$n = qn_q \quad (11)$$

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

$$m \geq \left\lceil \frac{n_q - 1}{n_q - d_q} \right\rceil.$$

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

$$m = \left\lceil \frac{n_q - 1}{k_q - 1} \right\rceil. \quad (12)$$

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

$$N_q = N = q^{k_q} \quad (13)$$

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 ( $\geq 3$ ), and which uses any values of  $k_q$  and  $n_q$  that satisfy

$$q + 1 \geq n_q \geq k_q + 1 \geq 3. \quad (14)$$

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  $k_q \geq 2$ ):

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

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

$$\frac{k_q = 2:}{n = q(1 + m)}$$

$$N = q^2$$

$$q \geq m$$

$$\frac{k_q = 3:}{n = q(1 + 2m)}$$

$$N = q^3$$

$$q \geq 2m.$$



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
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):

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

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

$$m_q = \left\lceil \frac{n_q - 1}{k_q - 1} \right\rceil.$$

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

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

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.<sup>3</sup> 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

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

<sup>3</sup> 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

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

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.<sup>4</sup> Let the set of  $L$  pairwise orthogonal Latin squares of order  $q$  be  $S_1, \dots, S_k, \dots, S_L$ , written as matrices in the symbols  $0, 1, 2, \dots, q - 1$ , with general element  $s_{ij}^{(k)}$ . Then a set of  $L$  pairwise orthogonal Latin squares  $T_1, \dots, T_k, \dots, T_L$  of order  $q^2$  and of general partitioned form

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

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.

<sup>4</sup> 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  $(v, k, b, r, \lambda)$  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  $\lambda$ . 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

$$A = S' \quad (\text{the transpose of } S),$$

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

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

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<sup>5</sup>

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

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-

<sup>5</sup> 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

$$\begin{aligned} k=3: v=1 \text{ or } 3 \pmod{6}, b=r(2r+1)/3, r=(v-1)/2; \\ \text{so that } w=3, n=1 \text{ or } 3 \pmod{6}, \\ N=n(n-1)/6, m=2 \\ k=4: v=3r-1, b=rw/4, r=\text{prime power}; \text{ so that } \\ w=4, (n-1)/3=\text{prime power}, N= \\ n(n-1)/12, m=3. \end{aligned}$$

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

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

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	$\vee$	01	10
0	0	1	01	01	11
1	1	0	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  $2\mu$  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:

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

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



It is possible to show that if every pair of columns of  $H^t$  is recoded in  $A$  according to the transformation

$$\begin{aligned} 00 &\rightarrow 1000 \\ 01 &\rightarrow 0100 \\ 10 &\rightarrow 0010 \\ 11 &\rightarrow 0001, \end{aligned}$$

then the resulting  $A$ -matrix represents a  $UD_3$  code if the  $H^t$ -matrix represents a triple-error-correcting code. The Bose-Chaudhuri codes [30] for  $e = 3$  have a matrix  $H^t$  with  $2^\mu - 1$  rows and no more than  $3\mu$  columns, for every positive integral value of  $\mu \geq 3$ . Since a pair of columns may be selected from  $H^t$  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  $q$ -ary ZFD<sub>3</sub> codes obtained by iterated composition with  $k_q = 2$  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<sub>2</sub> 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  $i$ th digit position of the left portion and the  $j$ th 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  $g_{i,j}$  has the value 1 when and only when the code contains such a code word having ones in the  $i$ th and  $j$ th 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  $(v, k, b, r, \lambda)$  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

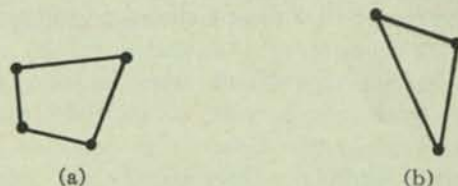


Fig. 1.

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".<sup>6</sup>

<sup>6</sup> 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:

$t = 2$	$n = 5$	$N = 5$
$t = 3$	$n = 10$	$N = 15$
$t = 7$	$n = 50$	$N = 175$
$t = 57$	$n = 3250$	$N = 92,625$

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

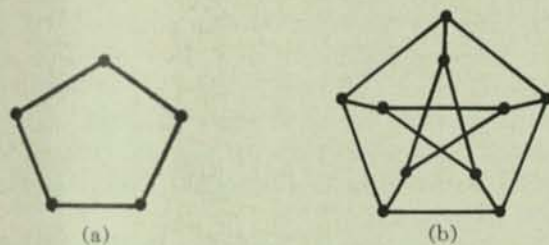


Fig. 2.

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

$$(a_1)(b_1)(c_{11})$$

$$(a_2)(b_2)(c_{22})$$

etc.,

in which the partial words  $a_1, a_2, \dots$  and  $b_1, b_2, \dots$  are selected independently as code words of the smaller code. The third partial words,  $c_{11}, c_{22}, \dots$  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 than  $ZFD_2$  codes of the same length.

If the partial words  $a_1, a_2, \dots$  and  $b_1, b_2, \dots$  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

$$(a_1)(b_1)(c_{11}) \quad \text{and} \quad (a_1)(b_2)(c_{12})$$

$$(a_2)(b_2)(c_{22}) \quad (a_2)(b_1)(c_{21})$$

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

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

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  $2 \times 2$  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  $3 \times 3$  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

$$\begin{array}{c} 001 \\ 010 \\ 100, \end{array}$$

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

$$\begin{array}{ccccccc} 001 & 001 & 1 & & & & \\ 001 & 010 & 0 & & & & \\ 001 & 100 & 0 & & & & \\ 010 & 001 & 0 & & & & \\ 010 & 010 & 1 & & & & \\ 010 & 100 & 0 & & & & \\ 100 & 001 & 0 & & & & \\ 100 & 010 & 0 & & & & \\ 100 & 100 & 1 & & & & \end{array}$$

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  $C_{p+1}$ ,  $p = 1, 2, \dots$ , 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

$$C_{p+1} = \begin{bmatrix} 11X & 10Y & 10Z & 00X & 01Y & 00Z & 00X & 00Y & 01Z \\ 10Z & 11X & 10Y & 00Z & 00X & 01Y & 01Z & 00X & 00Y \\ 10Y & 10Z & 11X & 01Y & 00Z & 00X & 00Y & 01Z & 00X \\ \hline 00X & 00Y & 01Z & 11X & 10Y & 10Z & 00X & 01Y & 00Z \\ 01Z & 00X & 00Y & 10Z & 11X & 10Y & 00Z & 00X & 01Y \\ 00Y & 01Z & 00X & 10Y & 10Z & 11X & 01Y & 00Z & 00X \\ \hline 00X & 01Y & 00Z & 00X & 00Y & 01Z & 11X & 10Y & 10Z \\ 00Z & 00X & 01Y & 01Z & 00X & 00Y & 10Z & 11X & 10Y \\ 01Y & 00Z & 00X & 00Y & 01Z & 00X & 10Y & 10Z & 11X \end{bmatrix},$$

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  $2 \times 2$  minor falling entirely within one of the ninths will certainly satisfy the condition. In fact, any  $2 \times 2$  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

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

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,

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

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 understanding 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.

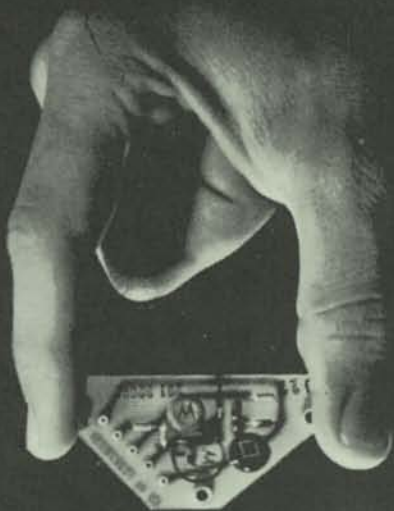
#### REFERENCES

- [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 Zato-coding.
- [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 zato-coding 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 Zato-coding 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. Stiasny, "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. Ashenurst, "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.









## We amplify careers at **MOTOROLA** in Phoenix

*The d-c amplifier shown above is a vital component of the solid state switching circuitry within the Minuteman Command Receiver*

Engineers discover, after joining the Military Electronics Division of Motorola in Phoenix, that they have a new-found enthusiasm for their work and a fresh sense of accomplishment. That's because all professional personnel are individually selected and then assigned to challenging state-of-the-art projects which fully utilize their training, experience, and creativity. We can thus provide greater career opportunities for our engineers and also broaden Motorola's capabilities as a leader in the field of advanced military electronics.

### SPECIFIC OPPORTUNITIES ARE:

Antennas and Propagation  
Command and Control  
Missile and  
Space Instrumentation  
Ground Support Equipment  
Digital Logic Systems  
Integrated Circuitry  
Reliability Analysis  
Reliability Program Coordination

Parts Reliability  
Data Acquisition, Processing  
and Display  
Radar and Radar Transponders  
Guidance and Navigation  
Space Communications  
Telemetry  
Instrumentation and Display  
Test Engineering

Write Phil Nienstedt, Manager of Recruitment, Dept. 6111-B

An Equal  
Opportunity  
Employer



**MOTOROLA**  
**Military Electronics Division**

WESTERN CENTER • P.O. BOX 1417, SCOTTSDALE, ARIZONA

Motorola also offers opportunities at Chicago, Illinois, and at Culver City and Riverside, California

### MORE REVIEW

ing sensitive only within a portion of the visible spectrum while the diode works through the infrared region. The phototube, however, is superior in terms of bandwidth and equivalent resistance. A typical tube can operate over a 3:1 bandwidth with a resistance greater than 100K ohms, but the diode, limited by shunt capacitance, has no more than 1000 ohms impedance, in a 5- to 10-percent frequency band.

The most common use envisioned for modulated light beams is as a communication carrier. Such a system could have a bandwidth of several thousand megacycles and a beam width of only a few miles at distances as great as that to the moon. Other possible uses exist in the fields of surveying and astronomy, where distances could be measured to small fractions of a microwave wavelength, and emitted light studied for possible natural modulation frequencies.

ROBERT JOE PRICKETT

### meeting review

#### BINARY SUPERPOSITION

The first fall meeting of PGIT was held late in September, at the Philco auditorium in Palo Alto. The speaker, Dr. William H. Kautz, addressed an audience of about 30 on the subject, "Data Communication through Binary Superposition Channels." Dr. Kautz is a senior research engineer, specializing in switching theory, coding theory, and logical design at SRI's computer techniques laboratory.

The speaker described a new family of binary codes developed originally for an information retrieval application. The first portion of his talk was concerned with this retrieval application, then a possible communications application of these novel codes was sketched. Finally, the third and major portion of the discussion centered on detailed properties of the codes, and techniques for constructing them.

#### Retrieval Application

The retrieval application was described in terms of a model for information retrieval from a large file of documents—10,000 to 1,000,000 documents. Each document is provided with an accession number and a set of relevant descriptors. These descriptors are chosen from a dictionary of terms, and each such descriptor is assigned a binary code word of fixed length,  $n$ , containing a small number



ploy the heterodyne scheme, for it has advantages in frequency selectivity and angular discrimination. The best optical filters pass a band 100 gc wide, but a heterodyne system, detecting only at the intermediate frequency, could look at much narrower bandwidths. And the local oscillator and received signals must be exactly parallel when they strike the detector surface. Otherwise, the beat frequency will vary in phase across the face of the detector and will be canceled out in the resulting current. This property enables the detector to serve also as a high-gain antenna.

A modulator has been built using a microwave cavity and a bar of KDP ( $\text{KH}_2\text{PO}_4$ ). Crystals of KDP exhibit a nonuniform directional dielectric constant when subjected to an electric field. This property is employed to produce elliptical polarization on a polarized light wave passing through the cavity. The simplest device consists of a cavity, containing the crystal of KDP, resonating in the  $\text{TM}_{010}$  mode. The unmodulated light beam is fed through a polarizer and into one end of the cavity. It then passes through the KDP crystal which is at the center of the microwave electric field. Here the two perpendicular components of the light wave are delayed unequal amounts to produce elliptical polarization. The now modulated light beam then leaves the cavity and passes through a second polarizer to produce an amplitude modulated, linearly polarized beam. Several other crystals can be used as modulators, including ADP ( $\text{NH}_4\text{H}_2\text{PO}_4$ ), ZnS, and CuCl. The last two look very good electrically, but have proven difficult to grow. A modulator employing KDP has produced 15 percent modulation on a laser beam when driven by one watt at 3 gc.

#### PIN Diode

A second demodulation scheme uses a PIN diode (a three-layer diode having positive- and negative-doped regions separated by a layer of intrinsic material). When the diode is properly biased, all carriers are swept out of the intrinsic region and no current flows. But, if the intrinsic region is then bombarded by light, electrons are freed and flow out in the form of a detected current. The diode can operate over a much broader part of the spectrum than can the photo cathode, the latter be-

(Continued on page 12)



Model B-221



Model B-801



Model B-601

## WAYNE KERR transformer ratio arm bridges: RF, Universal, VHF Admittance

— high-accuracy measurements over  
an exceptionally wide frequency range

**Model B-601/RF transformer ratio arm bridge**—measures resistance ( $10\Omega$ — $10\text{M}\Omega$ ), capacitance ( $.01$ — $20,000\text{mmf}$ ), and inductance ( $0.5\mu\text{h}$ — $50\text{mh}$ ) with 1% accuracy over a 15KC-5mc frequency range. Instrument can also be used to measure complex impedances (balanced, unbalanced, or with grounded center point), and impedances between any pair of terminals in a 3-terminal network.

Unit is particularly useful in measurements of transistors, and elements with very low  $Q$ . By use of transfer admittance techniques to extend the admittance range, instrument will also measure  $Y$  parameters of semiconductors.

**Model B-221/Universal Bridge**—for 2-, 3-, or 4-terminal measurements of impedance or transfer admittance with an accuracy of 0.1%. Range of Model B-221Q—capacitance:  $0.0002\text{ mmf}$ — $100,000\text{ mfd}$ . Resistance:  $50\mu\Omega$ — $50,000\text{ M}\Omega$ . Inductance:  $0.005\mu\text{h}$ — $50\text{ Mh}$ .

This bridge features a visual readout which displays resistive and reactive terms independently, avoiding any confusion from large multiplying factors. Instrument is unaffected by impedance of test leads, hence can be used to determine temperature coefficients of components under test conditions, or for any remote in-situ measurement.

**Model B-801/VHF Admittance Bridge**—measures impedances throughout a frequency range of 1-100 mc. Accuracy:  $\pm 2\%$ . This 3-terminal instrument permits in-circuit measurements, and is used for measurements of balanced and unbalanced admittances of antennas, etc. Can also be used to determine transistor characteristics under working conditions. For measuring frequencies from 50 to 250 mc, specify Model B-901. In both instruments, calibration is independent of frequency . . . is read out in terms of conductance, and either positive or negative capacitance.

For further information or demonstration, contact



WAYNE KERR

—Gertsch—

GERTSCH PRODUCTS, Inc.

3211 S. La Cienega Blvd., Los Angeles 16, Calif. • Upton 0-2761 • VErmont 9-2201  
Northern California Office: 794 West Olive, Sunnyvale, California, REgent 6-7031



of binary 1's. The coded descriptors pertaining to any given document are superposed on the same n-bit field. Superposition here means a bit-wise inclusive-OR operation (as in the punching of holes in a card). An individual in search of information concerning, say, superconductive thin films consults the dictionary of terms and finds "superconductivity" and "thin films" among them. The n-bit code vector obtained by superposition of the coded descriptors for "superconductivity" and "thin films" is used as a "quiz word" against the entire document file. The test is on logical inclusion (rather than identity match); thus, any document whose superposed descriptors have 1's in each position where the quiz word has 1's will be caused to "drop out," i.e., be selected for the searcher's attention. Among the documents (accession numbers) thus called to attention will certainly appear all documents bearing both desired descriptors. However, other documents also may appear in the output, since the fortuitous combination of 1's from entirely unrelated descriptors may place 1's in all the specified positions. Such documents are called "false drops"; they are rejected only by user inspection. Clearly, a useful system must have a fairly low false-drop rate.

#### Existing Systems

Systems are in existence that employ this kind of superposed coding for document retrieval. In the past, coding assignment of descriptor codes for these systems has been accomplished by random selection of code words, with some weeding out of obviously interfering codes. (For example, the same or nearly the same code words should not be assigned to different descriptors.) Such randomly selected codes will still suffer, of course, from false drops. However, the false-drop rate is statistically predictable, at least on the basis of suitable randomness assumptions. In practice, these assumptions are not always met. For example, not all descriptors are used with the same frequency, nor are they used independently of each other. Thus, the predicted false-drop rate for randomly selected superposed codes is only a rough guide.

Random superposed codes suffer from another limitation relative to the desired retrieval application. It was

(Continued on page 14)

## SOLID STATE PULSE/RF AMPLIFIER

### ... SPOTLIGHTING STATE-OF-THE-ART PERFORMANCE AT OFF- THE-SHELF PRICES

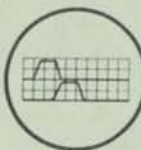
The model GT150 solid-state wideband pulse/rf amplifier provides extremely good pulse fidelity for fast pulses of less than 3 nanoseconds risetime... plus excellent stability. These amplifiers offer engineers state-of-the-art performance at off-the-shelf prices.

#### Specifications:

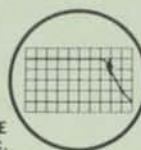
- Risetime: <3 ns
- Frequency Coverage: 1000 cps to 150 mc (min.)
- Gain: 40 db min. (48 db typical)
- Input & Output: Designed for 90 $\Omega$  systems. Will operate in any low impedance system without performance degradation.
- Output Capability: 0.5v p-p min. at 90 $\Omega$  (1.5v typical).

#### ... Plus these added features

- Noise figure: < 10 db average
- Regulated power supply
- Stability: 0.03 db/volt line change (typical)
- Zero warmup time
- Fast recovery time on overloads to 60 db.
- Gain control range of 15 db min.
- Standard BNC connectors
- No microphonics
- Size: 3" h x 4" d x 9" w
- Price: \$355



INPUT & OUTPUT WAVEFORMS  
EG + G Model 751 Puls-  
er Viewed on Tektronix  
Sampling System. Time  
Base: 5 nsec/div.



FREQUENCY RESPONSE  
Marker @ 150 mc.



**RHG ELECTRONICS LABORATORY, INC.**  
94 Milbar Blvd., Farmingdale, N.Y., NY 11737



Area Rep: WALTER ASSOCIATES  
P.O. Box 790  
Menlo Park, Calif., DA 3-4606



# ENGINEERS SCIENTISTS MANAGERS

B.S., M.S., Ph.D.

Top openings for:

CIRCUIT DESIGNERS  
SYSTEMS ENGINEERS  
ENGINEERING  
MANAGERS

in

Communications Systems  
Data and Telemetry Systems  
Control and Servo Systems  
Microwave and Propagation  
Solid-State Devices  
Microwave Tubes  
Microcircuitry

For personal and  
confidential referrals  
to our Client Companies'  
Management  
and Engineering  
Staffs, at no charge  
to you, submit resume  
or phone for appointment

## NORTHERN CALIFORNIA PERSONNEL

(a technical agency)

407 CALIFORNIA AVE.  
PALO ALTO  
DA 6-7390

### MORE REVIEW

required that document codes be uniquely decomposable into the original set of descriptors from which they were composed, assuming that no more than a given number,  $m$ , of descriptors apply to each document. A code with this property is called "uniquely decipherable to order  $m$ ," or " $UD_m$ " for short. Random superposed codes fail to possess this property.

#### Kautz/Singleton

The codes developed by Dr. Kautz and his associates, principally Dr. R. C. Singleton, are specifically tailored to provide this  $UD_m$  capability by systematic assignment of descriptor codes. Moreover, they possess zero false-drop rate when no more than  $m$  descriptors are superposed per quiz word used for searching. Thus, these codes are also " $ZFD_m$  codes" (zero false drops to order  $m$ ). The speaker indicated later that these two properties are related in an interesting way.

A possible application of these ideas to communications was pointed out. Consider the problem of sharing  $n$  channels of the frequency spectrum among  $N$  transmitters, where  $N > n$ . This is usually done on a specific time and frequency basis, with exclusive assignment of a given frequency to a given transmitter (at least for certain hours of the day). However, using the concept of superposed coding, one might assign to each transmitter several channels of the whole "field" of  $n$ , selected either randomly or on the basis of Kautz's codes. If no more than  $m$  transmitters are "on" at a given time, it should still be possible for a receiver to extract the desired information from one transmitter, even though other transmitters may be simultaneously using some of these frequencies. Similar schemes have been suggested by Costas (see Proc. IRE, Dec. 1959, pp. 2058-68). They may be more efficient in the utilization of bandwidth in congested situations than conventional frequency assignment techniques.

#### Code Construction

The formal problem of superposed code construction is as follows: "Find a large number  $N$  of  $n$ -bit code words, such that for a given integer,  $m$ , every (inclusive-OR) sum of up to  $m$  words is (a) distinct from all other such sums ( $UD_m$  property), or (b) does not logically include any other code word ( $ZFD_m$  property)."

In order that the resulting code be efficient, it is, of course, desirable that  $N$  be as large as possible. The speaker pointed out that very little is known about this optimality question.

The relationship referred to earlier is the implication:

$$ZFD_m \Rightarrow UD_m \Rightarrow ZFD_{m-1}, \text{ etc.}$$

In the connection, there is also an interesting analogy to the properties of standard error-correcting codes.

Construction of these superposed codes was first described in terms of a constant,  $w$ , the number of 1's per descriptor code. Codes with  $w = 1$  yield the trivial code with  $N = n$  that is  $ZFD_1$ . For  $w = 2$  one can find  $UD_2$  codes for which  $N$  behaves roughly as  $n^{3/2}/2$ . Actual values of  $N$  for small  $n$  are:

$n$	4	5	6	7	8	10	...	50
$N$	3	5	6	8	10	15	...	175

These  $UD_2$  codes were constructed by an ingenious graph-theoretical approach. In fact, let the  $n$  nodes of a linear graph be interpreted as positions of 1's in a word, and let a branch joining two nodes represent a weight-two descriptor code. Then the problem of constructing  $UD_2$  codes is seen to be equivalent to finding linear graphs of  $n$  nodes and  $N$  branches which contain no closed loops of fewer than five branches (i.e., no triangles and no quadrilaterals). The maximum values of  $N$  are shown above.

Techniques using balanced incomplete block designs (and partially balanced incomplete block designs) may be employed to construct  $ZFD_2$  codes with  $w = 3, 4$ , etc. For these codes  $N \sim n^{2/6}$ .

Dr. Kautz described in some detail how the properties of error-correcting codes can be exploited to construct  $UD_2$  for which  $N = 2^{n/4} - 1$ . They are obtained by first selecting a parity check matrix for a double-error-correcting Bose-Chaudhuri code of length  $N$ . This matrix will have  $N$  columns and  $n/2$  rows. Consider the column vectors of this matrix. Each column vector is doubled in length (to  $n$ ) by replacing zeros by the digits, 01, and by replacing ones by the digit pair, 10. The resulting set of  $N$  vectors form a  $UD_2$  code.

#### Latin Squares

Other techniques for construction of superposed codes involve the use of Latin squares to permit the com-



position of given codes to yield larger codes. For example, a given  $UD_2$  code with parameters  $n$  and  $N$  can be iterated to yield a  $UD_2$  code with parameters,  $n' = 3n$  and  $N' = N^2$ . Extension of this technique by using sets of orthogonal Latin squares yields a  $ZFD_m$  code with parameters  $n'' = (m+1)n$  and  $N'' = N^2$ , provided that it is possible to find  $m - 1$  orthogonal Latin squares of size  $N$  by  $N$ . (This is known to be possible for  $m \leq N$  if  $N$  is a power of a prime number.)

As an example of this technique, Dr. Kautz mentioned that the (trivial) code with  $n = N = 8$  and  $w = 1$  could be iterated to yield first a  $ZFD_6$  code with  $n' = 56$  and  $N' = 64$ , which can in turn be iterated to result in a  $ZFD_6$  code with  $n = 392$  and  $N = 4096$ . A randomly selected code for  $N = 4096$  would need  $n \approx 100$  in order to reduce the false-drop probability below  $10^{-4}$ . However, the deterministic code is better in terms of its guaranteed performance level, the lack of dependence of this performance on descriptor frequency and interdescriptor correlation, and also its unique decipherability up to six constituent descriptors.

Several other construction techniques were mentioned, among them the use of nonbinary error-correcting codes as a starting point, and also other composition methods.

#### Unknown Optimals

During the question-and-discussion period following the lecture, Dr. Kautz was asked if these codes were in any sense optimal. His answer was essentially that almost nothing is known regarding this point; the information theoretic point of view has not yet shed any light on it. Another questioner asked if multidimensional extensions of Latin squares, e.g., Latin cubes, had any application here. Dr. Kautz's answer indicated that there are a number of different generalizations of this sort—most of them not applicable to the coding problem. The possible use of some sort of Latin hypercube, as well as some other composition methods, is the subject of present study. He plans to publish this material early next year.

The small, but alert, audience present at the lecture seemed to be quite intrigued by the novel ideas presented by Dr. Kautz, judging by the

(Continued on page 16)

## COMPLETE NORTHERN CALIFORNIA COVERAGE FROM MOXON.....

A modern manufacturer's representative is no longer a happy-go-lucky fellow with a battered briefcase... It takes a large, technically competent field staff plus complete service, sales, and advertising backup to give you the coverage you need... Moxon Electronics, an organization of over 30 people, is this type of representative... Call your Moxon Man often.

### MEET THE SAN MATEO STAFF



**Dave Peters**  
*Regional Manager*

Dave is one of the oldest (in experience) Moxon Men, having joined the firm in 1957 B.Sp. (Before Sputnik). Before coming north to head up the San Mateo office, he was one of the top Moxon Sales Engineers covering the San Fernando Valley and Southern Coast which included the important Pacific Missile Range and Vandenberg Air Force Base.



**Gene Ward**  
*Sales Engineer*

Gene recently joined the Moxon organization after four years at MELABS where he was branch engineering manager. He has had extensive experience in microwave instruments and systems, and holds an EE degree from the University of California.



**Gary Schmidt**  
*Service and Inside Technical*

A welcome addition to the San Mateo office is Gary, who joins Moxon after four years with Neely Enterprises in customer and field service. In addition to acting as application engineer Gary will also set up a local service department.



**Vivian Stikes**  
*Office*

Vivian has been with Moxon Electronics since 1955 and knows the products backwards and forwards... so for accurate prices, delivery dates, and fast follow-up information, ask for Viv.

### PLUS IMPORTANT SALES, SERVICE, AND ADVERTISING BACKUP

Our first office was located in the basement of Mox's San Mateo home in 1951, and Mox still spends a good portion of his time calling on Bay Area customers... Another frequent visitor is Larry Courtney, who is responsible for the company's advertising and promotional activities... our Service Manager, Darrell Tomlinson, is "on call" at all times to assist our new Northern California service man in the shop, in the field, or in the training of customers... That's why we say, "You get *complete* coverage from Moxon Electronics."



15 41st Avenue, San Mateo, California  
Flreside 5-7961

SERVING NORTHERN CALIFORNIA FOR OVER 10 YEARS

#### REPRESENTING

ALFRED, ATI, ASTRODATA, CLAIREX, CMC, J-OMEGA, MARCONI, RUTHERFORD, SYSTEMS RESEARCH, TALLY, TRYGON, AND VIDAR.



# STANDARD FREQUENCY RECEIVERS



## MEASURE AND STANDARDIZE FREQUENCIES TO ONE PART IN 10 BILLION!

MODEL VLA Receiver Phase Comparator  
and MODEL SRA Servo Phase Shifter

make an automatic system for standardizing local frequencies to VLF standard frequency broadcasts. Graphic records of corrections made to local oscillator signals. Uses inexpensive oscillators. Each unit  $3\frac{1}{2}$ " standard rack panel.

MODEL VLA \$1490 MODEL SRA \$1990

Three frequency receivers now available



## MODEL WWVC COMPARATOR

Highly sensitive ( $1 \mu\text{V}$ ) crystal controlled receiver for utilizing WWV and WWVH transmissions in precision audio and radio frequency work and time interval measurements. 2" oscilloscope, 3" speaker.

$5\frac{1}{4}$ " x 19" x  $9\frac{1}{2}$ " rack mount ..... \$790  
Cabinet model ..... \$850



## MODEL SR7-H WWV RECEIVER

Dual conversion type. S meter, phone jack and speaker. Noise limiter, antenna trimmer, and provision for oscilloscope or headphones.  $3\frac{1}{2}$ " x 19" x  $8\frac{1}{2}$ " rack mount or  $3\frac{1}{2}$ " x 12" x  $8\frac{1}{2}$ " bench model ..... \$345



## MODEL WVTR RECEIVER

All transistorized for utilizing WWV and WWVH broadcasts. Frequencies to 25 mc-crystal controlled. Rack  $3\frac{1}{2}$ " x 19" x  $5\frac{1}{2}$ ".

Battery power \$560 AC power supply \$590  
Portable WWVT ( $9$ " x  $12$ " x  $5$ " available \$590

## SPECIFIC PRODUCTS

P.O. BOX 425 21051 COSTANSO ST.

WOODLAND HILLS, CALIF.

DIAMOND 0-3131 AREA CODE: 213

Prices & specifications subject to change

## MORE REVIEW

number of impromptu groups seen discussing these ideas during the coffee break which followed the lecture. It was felt that the 1962-63 season had got off to a flying start by this stimulating talk.

BERNARD ELSPAS

## meeting review

### PGCS MEETING WITH AIEE

On September 25 Cecil M. Kortman, manager, systems design, electronic systems, research and engineering, Lockheed Missiles and Space Company, presented an informative and interesting paper, titled "Sample Data Telemetry for Satellite Applications," to a joint meeting of the Communications Divisions of AIEE and PGCS. The paper was supplemented by a large number of excellent slides depicting both systems' concepts and actual hardware.

Mr. Kortman laid the groundwork for his paper with the past history of satellite telemetry, going back to the utilization of the early FM/FM systems. The various classes of data normally handled between the space vehicle and the tracking station were analyzed with respect to the basic requirements of speed and accuracy. These requirements were then compared with the capabilities of the various present-day modulation techniques.

Examples of past and present packaging of the space-vehicle hardware emphasized the progressive strides that have been made in this field.

Photos of tracking-station facilities illustrated the high degree of integration required by these systems. Ground-station predetection recording techniques, demodulation techniques, and analogs to digital converters were described. Also considered from the system data analysis flow chart was the reduction of redundancy that could be achieved in data processing.

MAURICE H. KEBBY

## meeting review

### NOISE IN OPTICAL MASERS

The October 17 meeting of PGED/PGMTT at Stanford heard Dr. William Louisell, visiting associate professor, now on leave from Bell Telephone Labs, speak on quantum noise in optical masers, showing that the limit of sensitivity of the optical maser is determined by what is called quantum noise. The problem was treated by

statistical methods, and the results give a noise figure that is in agreement with experimental data.

First, giving a brief review of the changes necessary to convert from a classical formulation of a problem to a quantum formulation, Dr. Louisell discussed the harmonic oscillator and the electromagnetic field in a cavity. These problems were set up in almost identical manner, and in fact the electromagnetic field in the cavity could be considered to be a collection of harmonic oscillators with a total energy equal to the sum of the energy of the individual harmonic oscillators. Then to change to a quantum formulation of the problem, the two variables, momentum, and position or electric and magnetic field, which commute in the classical problem, are now restricted so that the commutator of these two variables (now considered as operators) is  $i\hbar$ . The electric field in the cavity now may only have certain discrete values, that is, only a value corresponding to some integral number of photons.

The formulation of the problem in statistical terms involves setting up a function of the field amplitude which will give the probability of the field having any given amplitude. It is possible to set up a generating function which will give all properties of interest for the system merely by differentiation, instead of integrating separately to find the various moments required.

The example used to depict the amplifier or attenuator was a cube of material in which atoms are being excited by radiation from outside. It was assumed that the atoms had previously reached an equilibrium temperature with a Boltzman distribution of energy. Then for the case of no input signal and no input noise, the statistical generating function was obtained. This generating function showed that there was an output noise signal even with the zero input conditions, which must then be of quantum origin. The energy distribution in this noise signal is gaussian and at optical frequencies corresponds to a temperature of about  $10,000^\circ\text{K}$ /photon and, at microwave frequencies, corresponds to only a fraction of a degree. It was pointed out, however, that the very high equivalent noise temperature at optical frequencies is not a meaningful parameter.

(Continued on page 19)



9 April 1965

→ Bourne

Title: Development of Non-Random Binary Superimposed Codes

Objective: Develop and evaluate a family of non-random binary superimposed codes, for a broad range of parameter values appropriate to information retrieval systems. These codes should be simultaneously (a) efficient--that is, not wasteful of binary digits--and (b) easily implemented in encoding and deciphering circuitry or programs.

Importance of the Problem: Superimposed codes are required in one type of content-addressed information retrieval system, in order that the list of "descriptors" which represent each document may be compactly stored in a manner which facilitates later recall, in response to an inquiry. The way in which this is done is described in detail in a recent technical paper by SRI authors.\* There are also shown the distinct advantages of non-random codes usually employed in such retrieval systems. Several non-random code families are described, but the problem of developing really efficient and easily implementable codes still remains.

There is also a potential application of these codes to certain problems in making channel assignments within crowded communication bands.

SRI Approach: The development of these new codes will follow the approach now classical for the development of any new code: we seek an algebraic or combinatorial structure which generalizes known, simple examples of the sought-for code to the general case, in such a way that high efficiency is maintained. There

---

\*W. H. Kautz, and R. C. Singleton, "Non Random Binary Superimposed Codes,"



J. Dorn

Dec 1956

Vol 12, No. 4

# SOME PROBABILITY PROBLEMS CONCERNING THE MARKING OF CODES INTO THE SUPERIMPOSITION FIELD

by DR. G. OROSZ<sup>1</sup> AND DR. L. TAKÁCS<sup>2</sup>

Budapest

We deal here with some problems of superimposed random coding that arise when marking codes into the codefield. Our investigations are concerned with a general model. Systems used in practice are special cases of this, and may be deduced from it by making appropriate simplifying assumptions. We give exact formulas for the distribution of the marked sites in a codefield and for the distribution of multiple marking of a site. These results are obtained by applying a general theorem in probability due to K. Jordan. Some acquaintance with superimposed random coding and knowledge of basic combinatorial and probability theory are assumed.

1. Suppose there are  $n$  sites in the codefield, grouped into  $p$  subfields containing  $n_1, n_2, \dots, n_p$  sites respectively. Thus

$$n = n_1 + n_2 + \dots + n_p.$$

Random codes or 'words' consisting of  $\nu$  marked sites are to be written into this codefield. Of the marked sites,  $\nu_1$  are to fall in the first subfield,  $\nu_2$  in the second, ... and  $\nu_p$  in the  $p$ th. Thus

$$\nu = \nu_1 + \nu_2 + \dots + \nu_p.$$

The number of distinct patterns, or alphabet of letters, that can be formed by marking  $\nu_i$  of the  $n_i$  sites in the  $i$ th subfield is the number of different ways of selecting  $\nu_i$  distinct objects out of  $n_i$ . To wit

$$(n_i! \nu_i) = n_i! / (n_i - \nu_i)! \nu_i!$$

Each distinct word is made up from a letter formed in each of the subfields. Thus the size of the vocabulary, which is the number of words that can be formed thus, is

$$V(n, \nu) = (n_1! \nu_1)(n_2! \nu_2) \dots (n_p! \nu_p). \quad (1)$$

Suppose a selection of  $N$  distinct words drawn from this vocabulary are superimposed on the codefield. The number of such selections is

$$(V(n, \nu)! N). \quad (2)$$

2. The number of letters that can be formed out of  $\nu_i$  marks in the  $i$ th subfield without marking any of  $k_i$  specified sites is

$$((n_i - k_i)! \nu_i).$$

<sup>1</sup> Library of the Eötvös L. University.

<sup>2</sup> Mathematical Institute of the Hungarian Academy of Sciences.



Applying this to the whole codefield we see that the number of words that have  $k$  specified sites unmarked— $k_1$  being in the first subfield,  $k_2$  in the second, &c.—is

$$V(n-k, \nu) = ((n_1 - k_1)! \nu_1) ((n_2 - k_2)! \nu_2) \dots ((n_p - k_p)! \nu_p)$$

and  $N$  distinct words can be drawn from this restricted vocabulary in

$$(V(n-k, \nu)! N) \quad (3)$$

ways.

(2) and (3) show that, if all words are equally likely, the probability of  $k$  specified sites being unmarked,  $k$  ranging from zero to  $n$ , is

$$(V(n-k, \nu)! N) / (V(n, \nu)! N). \quad (4)$$

3. In general the number of sites marked when  $N$  words are superimposed is less than the sum of the number of marked sites in each word severally, because some of the marked sites may be common to two or more words. To get information about this the probability distribution of unmarked and marked sites must be found. The tool for this is K. Jordan's theorem giving the probability of *exactly*  $k$  events occurring out of a possible  $n$  (see p. 234). Here we will take non-marking as an occurrence of an event at a site, and marking as non-occurrence.

Application of the theorem in this sense gives the probability of *exactly*  $k$  marked, and  $n-k$  unmarked, sites as

$$P_k = \sum_{j=n-k}^n (-1)^{j-n+k} (j! (n-k) B_j) \quad (5)$$

where

$$B_j = \sum (n_1! j_1) (n_2! j_2) \dots (n_p! j_p) (V(n-j, \nu)! N) / (V(n, \nu)! N) \quad (6)$$

is the binomial moment of the distribution  $\{P_k\}$ ,  $k$  ranging from zero to  $n$ .

Also we can determine the probability of *at most*  $k$  marked sites with *at least*  $n-k$  unmarked sites. For this we use the second form of K. Jordan's theorem, which gives the probability of *at least*  $k$  events occurring out of a possible  $n$  (see p. 234). The required probability is found to be

$$\mathfrak{P}_k = \sum_{j=n-k}^n (-1)^{j-n+k} ((j-1)! (n-k-1) B_j) \quad (7)$$

By means of the moments  $B_j$  we can determine also the mean and variance of  $P_k$ . For this it is enough to consider  $B_1$  and  $B_2$ :

$$B_1 = \sum_{i=1}^p n_i \{ (1 - (\nu_i/n_i)) V(n, \nu)! N \} / \{ V(n, \nu)! N \} \quad (8)$$

$$B_2 = \sum_{i \neq k} n_i n_k \{ (1 - (\nu_i/n_i)) (1 - (\nu_k/n_k)) V(n, \nu)! N \} / \{ V(n, \nu)! N \} + \\ + \sum_i (n_i! 2) \{ (1 - (\nu_i/n_i)) (1 - \nu_i/(n_i-1)) V(n, \nu)! N \} / \{ V(n, \nu)! N \}. \quad (9)$$



Whence we have: the mean number of unmarked sites

$$B_1,$$

the mean number of marked sites

$$n - B_1,$$

and the variance of the number of unmarked sites

$$2B_2 + B_1 - B_1^2.$$

4. The probability of *specified* sites being marked can be found similarly by applying the general theorem to the occurrence of none of the events. Thus the probability of  $k$  specified sites being marked (regardless of what happens to the other sites) is

$$Q_k = \sum_{j=0}^k (-1)^j \sum V(k, j) (V(n-j, \nu)! N) / (V(n, \nu)! N), \quad (10)$$

the summations being over all partitions of  $j$  and  $k$ .

5. Because the words are superimposed, a marked site may belong to more than one of the  $N$  words. A site in the  $i$ th subfield can be an element of  $(n_i - 1! \nu_i - 1)$  of the  $(n_i! \nu_i)$  possible  $i$ th letters and cannot be an element of the remaining  $(n_i - 1! \nu_i)$ . Hence the probability that, when  $N$  words are superimposed, a site in the  $i$ th subfield be common to  $j$  words—that is, be marked  $j$  times—is

$$p_j^{(i)} = \frac{\{(v_i/n_i) V(n, \nu)! j\} \{1 - (v_i/n_i)\} V(n, \nu)! (N-j)!}{\{V(n, \nu)! N\}} \quad (j = 0, 1, 2, \dots, N). \quad (11)$$

The average multiplicity of marking at a site is

$$\sum_{j=0}^N j p_j^{(i)} = N v_i / n_i$$

and the variance of this is

$$\sum_{j=0}^N (j - N v_i / n_i)^2 p_j^{(i)} = N (v_i / n_i) (1 - v_i / n_i) \{1 - (N-1) / (V(n, \nu) - 1)\}.$$

The mean number of  $j$ -fold marked sites in the whole codefield is

$$\sum_{i=1}^P n_i p_j^{(i)}.$$



## APPENDIX

The theorem of K. Jordan states that:

(1) the probability of *exactly*  $k$  events occurring out of  $n$  possible events,  $A_1, A_2, \dots, A_n$ , is

$$P_k = \sum_{j=k}^n (-1)^{j-k} (j! / k!) B_j \\ = B_k - (k+1)! / k! B_{k+1} + (k+2)! / k! B_{k+2} - \dots + (-1)^{n-k} (n! / k!) B_n;$$

(2) the probability of *at least*  $k$  events is

$$\mathfrak{P}_k = \sum_{j=k}^n (-1)^{j-k} (j-1! / (k-1)!) B_j \\ = B_k - (k! / (k-1)!) B_{k+1} + (k+1! / (k-1)!) B_{k+2} - \dots + (-1)^{n-k} (n-1! / (k-1)!) B_n.$$

In both expressions  $B_j = \sum P(A_{i_1}, A_{i_2}, \dots, A_{i_j})$ ,

where the right-hand summand denotes the probability of joint occurrence of the  $j$  specified events,  $A_{i_1}, A_{i_2}, \dots, A_{i_j}$ , whether the remainder occur or no. The summation extends over all  $j$ -fold selections from all the  $A$ 's without repetitions, so the number of terms in each sum is  $(n! / j!)$ .

Inversely the  $B_j$ 's may be expressed in terms of the probabilities,  $P_k$  or  $\mathfrak{P}_k$ , thus

$$B_j = \sum_{k=j}^n (k! / j!) P_k \\ = \sum_{k=j}^n (k-1! / (j-1)!) \mathfrak{P}_k.$$

## REFERENCES

- See JORDAN, KÁROLY. A valószínűség-számítás alapfogalmai. (French abstract: Les fondements du calcul des probabilités.) *Matematikai és Fizikai Lapok*, 34 (1927), 109-36.
- FELLER, W. *An introduction to probability theory and its applications*. (J. Wiley and Sons, New York, 1950), pp. 64, 74.



have been gathered together with great labor and expense, from one library to another.

From time to time a professor who has built up a great subject collection in one institution transfers to another and will then, if unable to bring his special collection with him, want to build up a second one even if most of the books concerned will very rarely be used. One of the questions that the future must face is whether collections that fall in this category can be transferred bodily from one institution to another as the demand for the books shifts. And, carrying the problem one step farther, can university authorities get together and say that only one will do the advanced work in certain limited fields and that others will refrain from duplicating that work? Would this be restraint in trade?

Another question which libraries have not yet faced squarely is whether a library which takes the responsibility for a limited field, spends the money to buy the books, catalogue them and serve them, can be expected to make them available without charge to scholars who have no connection with it, when its own students pay high tuition before they are allowed to use the library.

Whatever may come out of the situation described in the above paragraph, it must be realized that coöperation among libraries at a distance will always be confined largely to books that are little used or very expensive, and this expense must be considered to in-

clude not only the cost of purchasing the material, but the cost of cataloguing it, storing it, and making it readily available. Each of these four types of expenditure is important.

No attempt will be made here to consider the place of new methods of reproducing printed materials in coöperative development of research collections, but that place is sure to be of growing importance.

Finally, the point should be made that books that are used frequently should be in even small college libraries. The same is true for many reference works and for standard bibliographies which are of use in learning of the existence of material and, if possible, in determining the library in which it is located. And this leads us to say again that coöperative acquisition should in most cases deal with material that is used almost altogether by advanced scholars in their original research or for graduate students preparing their doctoral dissertations. These are the books and pamphlets that make the difference between the good college and the great university library collection, that make necessary the tremendous staffs for cataloguing and public service and the library buildings that at present prices may cost \$5,000,000 to \$10,000,000 or more to construct. It is the costs involved directly or indirectly in these books and pamphlets that have made coöperative acquisition desirable, will undoubtedly make it necessary, and should make it more satisfactory as the years roll by.

## ZATOCODING APPLIED TO MECHANICAL ORGANIZATION OF KNOWLEDGE

CALVIN N. MOOERS\*

### ABSTRACT

The mechanical organization of knowledge for retrieval of stored information can no longer neglect the developments of point-to-point communication theory, since both deal with information and handle it by machines.

The most versatile retrieval systems are those which delegate a separate tally to each information item, and which impress marks on the tally for the machine to read and to use for selective purposes. Coding is the relationship between these marks and the intellec-

\*President, Zator Company, Boston, Mass.



tual content of the information items. Coding determines the complexity of the selective machine and the utility of the whole process. A set of invariant coding principles is stated which define maximum coding efficiency for any tally selecting machines, and parallels are drawn between these principles and the conclusions of modern point-to-point communication theory. Zatocoding is defined--the system which superimposes random subject code patterns on the tally--and it is found to obey each of the invariant principles of coding efficiency while still allowing the simplest possible selector machine structure

- I Introduction
- II Information retrieval by machine
- III Retrieval systems and communication theory
- IV Principles and practice of Zatocoding

## I - INTRODUCTION

Zatocoding is a system of coding for selecting recorded intelligence by means of a machine. It is of special importance to the documentalist because it gives him powerful new tools of idea specification for retrieving information from storage. Zatocoding has practical economic consequences because it can enormously enhance the information-handling capabilities of the most elaborate large-scale machines, and also because it can be practiced with extremely simple machines.

Machine handling of ideas is an old art in the field of point-to-point wire and radio communication. Yet, over many years, communication technology has had little influence upon the closely related problems of documentary information retrieval. Within the last decade there has been a burst of developments and fundamental advances in communications technology which can no longer be ignored if documentation techniques in information searching

are not to be left far behind. Since the war, the author's research has been directed primarily toward this aspect of documentation.

The nature of the advance represented by Zatocoding can best be followed by drawing parallels to recent progress in communications and related technology. Typical advances are frequency-modulated radio-telephony, military radar, electronic digital computing machines, and pulse-code systems of speech modulation. These advances stem from two sources: improved components and improved systems. Typical new or improved components are high-powered magnetrons, cavity resonators, photoelectric tubes, and semi-conductor diodes. Equally important is the recent development of entirely new systems or schemes for using these improved components. The components needed to construct a frequency-modulated radiotelephony system were available in the earliest days of radio broadcasting. Yet frequency modulation as a possibility was actually discarded because at that time there was no realization of the tremendous advantages that could be secured through its use in a properly designed system.

Recently developed communications systems --including computing machines which have to talk to themselves and their operators--owe a great deal to the new discipline called "communication theory." Communication theory<sup>1</sup> deals with systems of transformation of information from one medium to another, as from voice to electric current; with modes of representation of information within circuits, storage elements, and transmission channels; with interaction between noise, operator and equipment imperfections and the modes of information representation; and with ways to overcome these imperfections. The relevance of communication theory to documentary problems is obvious when we realize that a document of paper, a voice, or a television picture can each carry intelli-

<sup>1</sup>C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, v. 27, pp. 379-423, 623-656 (July, October, 1948).

C. E. Shannon, "Communication in the Presence of Noise," *Proc. I.R.E.*, v. 37, pp. 10-21 (January 1949).

William G. Tuller, "Theoretical Limitations on the Rate of Transmission of Information," *Proc. I.R.E.*, v. 37, pp. 468-478 (May 1949).

Norbert Wiener, "Cybernetics," John Wiley & Sons, New York, 1948.

The bibliographies of the Shannon and Tuller papers are recommended.



gence, and that their transmission and utilization have many common problems.

Zatocoding<sup>2</sup> is a new system, related to these new communication theories. It is especially capable of handling the problems of documentary information retrieval by a machine. As measured by capability of dealing with ideas, and by economy in machine structure, Zatocoding is the most efficient coding system presently known. Its efficiency is related to the several close analogies that exist between it and certain of the system requirements for high-efficiency point-to-point signalling. In particular, I wish to mention that Zatocoding has parallels to the probability and choice concept in coding of messages,<sup>3</sup> and to the use of random codes<sup>4</sup> for transferring message ideas in communication theories.

Zatocoding is a considerable departure from the earlier methods of coding used for information retrieval, and therefore it has some unusual characteristics. Referring to Fig. 1, code patterns, representing the descriptor ideas, are superimposed in the single coding field, and thus they intermingle and mix. The coding patterns are initially generated by a mechanical random process, and they are then assigned to the subject ideas in any order. For selection, the marks and spaces of the codes are not matched, but instead pattern inclusion of the marks only is employed. Rejection of unwanted material is governed by statistical rules, allowing a slight percentage of "extra" unwanted material to come out with the desired material. In most cases, the coding and idea-manipulating capabilities of a given set of equipment can be enormously enhanced by Zatocoding, as a few examples will show.

The Rapid Selector<sup>5</sup> operates on a film memory in which the coding field for each frame has some 216 positions that can be

marked by opacities. Assuming a collection of 5,000,000 documents--comparable to the Library of Congress--a Zatocoding pattern of 8 marks per subject idea can be used. The coding field can hold up to 18 such subject patterns, and selections can be made upon any combination of these patterns. The size of the Zatocoding descriptive vocabulary is unlimited. In comparison, the present Rapid Selector coding system (and associated optical-electronic adjuncts) records only 6 subjects in the field, it searches and selects upon only one subject, and combinations of subjects cannot be used to specify selection.

A Hollerith card has 960 positions that can be marked by punching out holes. If the collection of records is moderate-sized, that is, of less than 10,000 pieces, Zatocoding patterns of only four punches can be used. In this case, 165 different subject ideas or record statements can be put into the card by Zatocoding for use in selection and correlation. An entire medical or sociological case history might be recorded on only one card. Again, the size of the descriptive vocabulary is unrestricted. With conventional procedures, it is impossible to record 165 statements on a Hollerith card because there are only 80 columns which are available for numerical codes.

The very high efficiency of Zatocoding makes it the technique of choice for coding the simpler edge-notched cards. The typical Zatocard shown in Fig. 1 has only 40 positions that can be notched. Yet this card is suitable for a collection of 10,000 items, or larger, and it can be sorted with Zator equipment shown in Fig. 2 at speeds of around 800 cards per minute. (The electronic Rapid Selector is only 12 times faster.) Seven independent, cross-referencing subjects can be notched by Zatocoding into the 40-position Zatocard.

Another version of the Zatocard has 72 field positions and it can hold 13 subject ideas.

<sup>2</sup>C. N. Mooers, "Putting Probability to Work in Coding Punched Cards," paper, 112th Meeting American Chemical Society, New York City, September 1947. U.S. and foreign patents pending on Zatocoding.

C. N. Mooers, "Information Retrieval Viewed as Temporal Signalling," Proceedings of the International Congress of Mathematicians, Cambridge, Massachusetts, September 1950.

<sup>3</sup>Shannon (1948) *ibid.* p. 379.

<sup>4</sup>Shannon (1949) *ibid.* p. 17.

<sup>5</sup>Anon., "Report for the Microfilm Rapid Selector," Office of Technical Services, U.S. Dept. of Commerce, Washington, D.C. (1949).



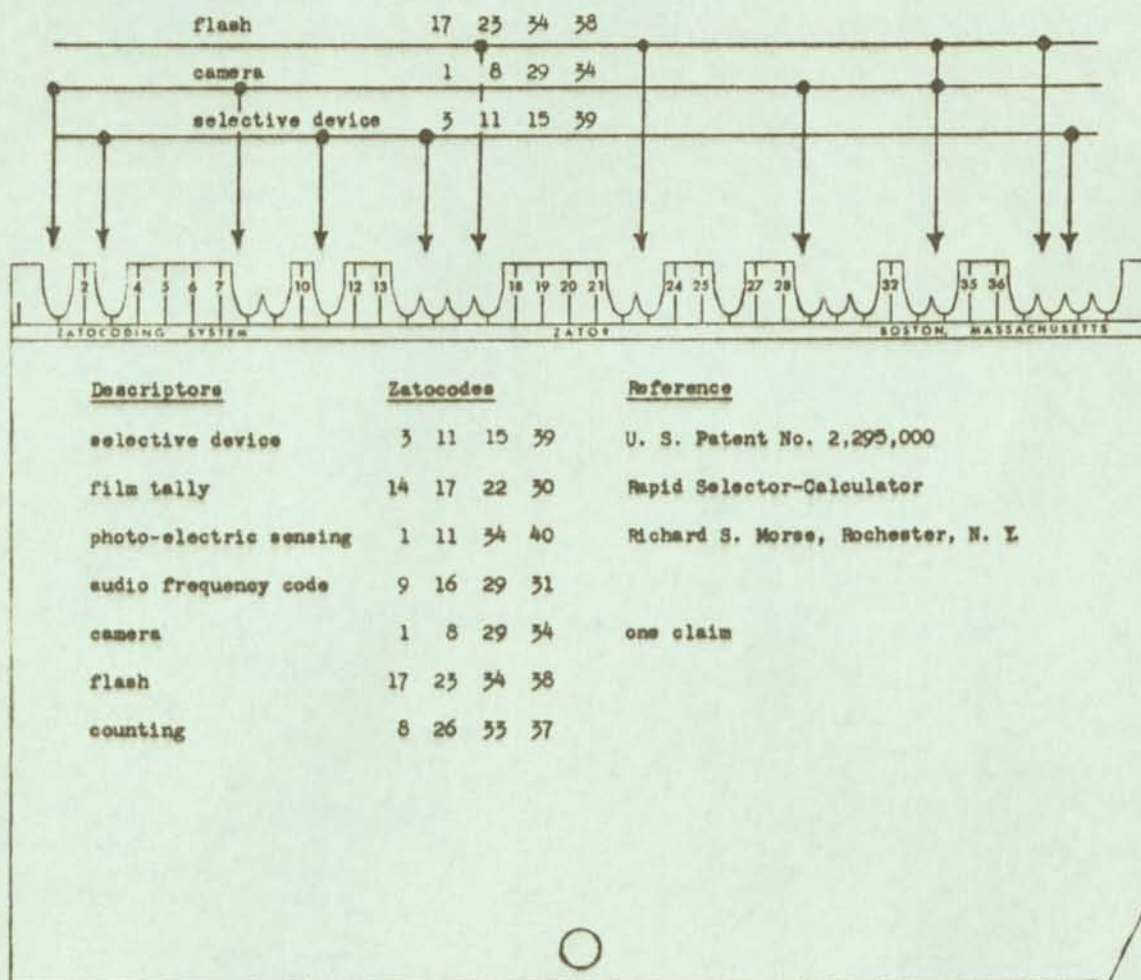
Simultaneously Selective Patterns

Fig. 1. ZATOCODING, illustrated with a 5 by 8 inch edge notched Zatocard for the tally. Note that the random Zatocode patterns in the field overlap and intermingle. Selection on the combination of three descriptors, "flash," "camera," and "selective device," is according to the inclusion of the pattern of arrows into the pattern of notches in the coding field. Zato cards are sorted by the selector shown in Figure 2.

It is suitable for more complicated indexing problems or larger collections.

To explain why Zatocoding is able to give such large advantages over older methods, it is necessary to examine concepts inherent in information retrieval by a machine and to compare these concepts with the more familiar methods of classification, indexing, and card-cataloging. Following this discussion,

the section on retrieval systems and communication theory states some new principles of information retrieval systems and shows how they apply to machine operation irrespective of the intellectual schemes. The final section, on principles and practice of Zatocoding, outlines the characteristic features of Zatocoding and gives the step-by-step procedure for the design of a Zatocoding system.





Fig. 2. THE ZATOR SELECTOR, capable of sorting edge notched Zatocards at the rate of 800 cards per minute. The black box-like portion of the Zator Selector is vibrated by a motor in the base. At the bottom of the box, a row of holes allows pins to be inserted to form the selective grid pattern. For selection, a pack of Zatocards is placed notched edge down on the vibrating grid. The rejected cards are those whose edge notches do not fit the pattern of the grid; hence they are supported on top of the grid. The edge notches of the accepted cards fit the grid and the cards drop down through the grid to a distance equal to the depth of the notches. In this way, the accepted cards are offset slightly below the rejected cards. To separate the rejected cards from the accepted cards the operator spears the pack of rejected cards through the small hole on the edge opposite the notches and lifts the pack out. The accepted cards, being offset below the rejected cards, are not engaged by the spearing tool; they drop out from the pack onto the table.



## II - INFORMATION RETRIEVAL BY MACHINE<sup>6</sup>

Information retrieval is the name for the process or method whereby a prospective user of information is able to convert his need for information into an actual list of citations to documents in storage containing information useful to him. It is the finding or discovery process with respect to stored information. It is another, more general, name for the production of a demand bibliography. Information retrieval embraces the intellectual aspects of the description of information and its specification for search, and also whatever systems, techniques, or machines that are employed to carry out the operation. Information retrieval is crucial to documentation and organization of knowledge.

We will be concerned here with information recorded and stored in a collection. The unit of information is a single document, paper, or report. Storage facilities and reproduction processes are not considered.

The subject matter of each document or other unit of information is characterized or described by means of a set of "descriptors" taken from a formal vocabulary of such terms. A "subject heading list" will call to mind a rough approximation of what is meant here.

If we allow the word "machine" to extend to any physical or material structure that is useful for information retrieval, we find that classified documents, index lists, and card catalogues are all machines for the organization of knowledge. This being so, it will be useful to examine them here to give perspective to our discussion.

Classification is a system of mechanical organization of knowledge in which usually the documents themselves are placed in a physical arrangement defined by a "classification schedule." The classification schedule is a predetermined listing of combinations and patterns of descriptors, with the order of listing of the combinations and patterns usually supported by a set of logical, or arbitrary, arguments and conventions. A classification of documents is thus a passive machine. Classification systems

are usually held to be systems for information retrieval, and not merely elaborate forms of storage. If so, then it should be possible for a prospective user of information to find his information by first consulting the classification schedule, and then by going directly to the desired documents. Moreover, he should have a fairly high expectation of success in efficiently finding what he wants to by this operation. As we all know, classification systems do not in fact meet this ideal. I believe there are some fundamental reasons why they can never do so.

An index is (usually) an alphabetical listing of single descriptors and their simpler combinations. The index format may be either a list on a sheet of paper or entries on a set of cards. To each such entry there is a page number or other citation to the location of the original document or unit of information. The list of index entries is the tangible machine of information retrieval. The documents themselves may be stored in any order, which is a great advantage. Information retrieval is accomplished by naming one or two or three descriptors, finding an entry with the desired combination in the list, and then following the citations to the original document. Indexes are powerful tools of information retrieval, though they run into difficulty when handling a large number of interacting descriptors. They must be subdivided finely enough so that a human searcher will not tire before finding his entry--a serious weakness of a practical index. We note, however, that an index has greater flexibility than a classification system in handling useful combinations and variations of descriptor patterns.

Human frailties can be avoided by turning the search job over to an active non-human machine. Each unit of information in the collection is then given a tally which can be manipulated by the machine. The tally may be a punch card, a frame of photographic film, or a section of a magnetic recording tape. The tally bears two kinds of information: a) a citation to the storage location of the unit of information, and b) the set of all descriptors applicable to that unit of information. These

<sup>6</sup>Compare: C. N. Mooers, "The Theory of Digital Handling of Non-Numerical Information and its Implications to Machine Economics," Zator Technical Bulletin No. 48, Zator Company, Boston, 1950.



descriptors are written on the tally in a fashion that can be "read" by the machine, such as by punches, dots, or magnetic spots.

When the tallies are cards and a human being instead of a machine reads the descriptors recorded on them, the system reduces to the ordinary card catalogue. However, humans are expensive to use--especially compared to the simplest machines, such as the Zator machine--and they have many frailties. They are slow, inaccurate, inattentive, and quite definitely limited in their ability to search through multitudes of cards for descriptor combinations of high multiplicity.

To return to the non-human machine: The descriptors used on the tallies must be broad for greatest utility. "Aircraft" and "helium" are excellent descriptors, but the typical subject heading "aircraft, lighter than air, helium filled" is too narrow and specialized. Such a narrower meaning can be generated with broad descriptors, for finding purposes, by specifying both "aircraft" and "helium" among the descriptors of the information.

Each document or unit of information is characterized by a set of descriptors taken from the vocabulary of descriptors. Each descriptor of the set applies to, or is true in some way of, the information content of the unit of information. The descriptors operate independently in this type of characterization. The fact that there are several descriptors in the set may mean that they formed some interacting combination in the original document, or it could just as well mean that they relate to independent ideas scattered through the document. Using descriptors in this fashion drops almost all relationships between the ideas represented by the descriptors.

Is this an undesirable degeneration of the information? Is it therefore abhorrent? Evidence shows that it is not, for several reasons. In the first place, it actually works very well in actual information retrieval systems. Also, relationships are subtle things, depending upon the point of view in most information situations. A person looking for information cannot be expected to have a cut-and-dried point of view, nor can he be sure of the relationship. All he is sure of is a very few of the objective or concrete facts of the situation, that is, the descriptors of the kind mentioned.

Here a distinction is very important. The

distinction is between communication of information by language (which we don't need) and the appropriate language for finding information (the kind we must use in information retrieval). The point is worth a longer discussion than can possibly be given here.

The relationship between the machine language of punches, dots, or spots on the tallies and the intellectual content of the original documents given by descriptors is called the coding. The very nature and design of the machine used for information retrieval is determined primarily by the coding system adopted. The actual competence of the system to retrieve information also depends almost entirely upon the coding adopted.

Serious mistakes in coding schemes have been made by adopting or designing a mechanism first, and then using Procrustean techniques to force the coding scheme to fit the limitations of the machine. Certain experiments with tabulating machines provide examples of this approach. Equally serious mistakes can arise from taking the methodology of other systems--e.g., the decimal symbolism developed for classification by physical array--and forcing such a methodology upon machines which could otherwise be made highly capable and versatile.

Efficient coding systems for the tally method do exist. Therefore, it is pertinent for us to review the reasons why the machine-sorted tally method deserves special consideration as a technique of information retrieval:

1. The load of tally manipulation, descriptor sensing, and the problem of choice or selection is all turned over to a machine.

2. High-speed tally-searching machines already exist and speedier machines have been proposed. Such a machine can very rapidly scan all the tallies in a collection and compile an exhaustive demand bibliography to the specifications of any prospective user of information.

3. Pre-determined schedules of descriptor combinations are not necessary nor are they employed with tallies. The descriptors operate independently. Thus retrospective searches can be made for interrelations or correlations between descriptor ideas that were not foreseen when the descriptors were separately recorded on the tallies. In contrast, according to conventional library methods, the searcher is limited to interre-



lations actually foreseen and recorded by the cataloguer. Separate descriptor ideas are very objective, while the interrelations (dear to the cataloguer) depend upon the point of view. Unfortunately the points of view of the engineer and the cataloguer are often different.

4. The combination of many descriptors is used to specify a search. In coding the structure of an organic chemical compound, 15 or 20 descriptors might be used on one tally. To search for this compound and its relatives, as many as 6 or 8 descriptors might be used to specify its class. With a capable code, a machine can handle this easily. Human methods would be most inefficient and liable to error.

5. The documents themselves are stored in any convenient order or fashion. Storage is uncomplicated by classification.

6. Changes, or the inclusion of new knowledge or descriptors, do not upset the system or the storage arrangement. Because the descriptors are used independently, new ones are simply added to the vocabulary along with the rest. Unlike a classification schedule, the vocabulary is a mere listing of descriptors and there is no attempt at logical structure.

7. Finally, an important conceptual point: By the tally method the knowledge contained in the collection of documents is given no actual organization until a user requests information. Only at that time is "organization" brought into being. Then the request is framed by a set of descriptors, the machine scans the tallies, and prepares a demand bibliography. In this sense, every time a request is met, the collection is "organized" from a new point of view.

### III - RETRIEVAL SYSTEMS AND COMMUNICATION THEORY

Tally methods hold outstanding possibilities for mechanical organization of knowledge--provided that a competent coding system is adopted. In this section we will consider those general principles of coding which hold irrespective of machine details, tally details, or philosophy of coding. These are invariant principles of cod-

ing systems. When these principles are obeyed, the coding system is efficient--and conversely. These coding principles have some remarkable parallels to principles otherwise developed in communication theory. Certain of these parallels will be indicated through references.

The speed of scanning the tallies is roughly proportional to the complexity of the searching machine, and is inversely proportional to the size of the tally field. Certainly, it will take longer, or require a better machine, to search a tally having a field with 1,000 positions than to search one with 100 positions. It is highly desirable to scan all the tallies.

Principle 1: SMALL FIELD. The size of the tally field should be made as small as is compatible with the other requirements of the problem.

With a small tally field, made desirable by selector simplicity and overall speed, it is necessary to plan for maximum utilization of the field. To give the tallies the greatest possible powers of separation or selection, the patterns of marks and spaces on the tally fields should have the greatest variety possible. No blank areas should recur on tallies because of information not recorded. Neither should similar patterns recur frequently on the same portions of the field.

With these assumptions it can be shown algebraically that the greatest number of different patterns in the tally fields--and thus the greatest variety--will occur when approximately one-half of the positions in the fields are marked, and when the patterns are as random as possible.<sup>7</sup>

Principle 2: FIFTY PER CENT. Whatever the coding system, maximum utilization occurs when the density of marks in the field is in the neighborhood of 50 per cent.

Principle 3: RANDOM PATTERNS. Maximum utilization requires random patterns in the tally fields.

<sup>7</sup> Completely random patterns correspond to "white noise" of signal theory. Compare Shannon (1949) *ibid.* p. 17. Randomness of this kind gives an ultimate coding of intelligence, and Zatocoding utilizes this property.



The coding system should be specifically designed only for finding, searching, discovering, or retrieving information. The coding should not be influenced in any way by features of arrangement or storage of documents. Neither should the coding be concerned with communicating the information in the document. The coding should talk about--not communicate--information. If a man wants information on the effect of temperature on his experiment, he should not be forced to anticipate that the critical temperature is 10 degrees before he can cause a machine to produce his information from the files. He wants to ask the retrieval system a question about temperature in his experiment, not tell it. If he knew the answer was 10 degrees, he wouldn't need to go to an information source. The descriptors used on the tallies should only outline the information of the document. They should specify only what kinds of information can be found there.

**Principle 4: RETRIEVAL LANGUAGE.**

The coding of the tallies should carry only "retrieval language." The ordinary "communicative language" that conveys actual information remains in the document itself, or if it is on the tally at all, it is confined to the written abstract and it is not marked in the coding field.

The distinction between communicative and retrieval language is stressed here because it has long been an ideal in library science to "pin-point" the documents by a classification symbolism. Pin-pointing in this fashion is tantamount to communication. Theoretical studies in our organization, and practical experience gained from setting up a wide variety of successfully operating information retrieval systems, point to the definite conclusion that communicative language or pin-pointing is incompatible with successful retrieval.

Another invariant principle is the concept of choice<sup>8</sup> in an information retrieval system. One does not pick information or documents from an infinite universe of documents. Selection is always referred to a finite--though possibly very numerous--collection in storage

somewhere. Therefore, all that information retrieval can accomplish is a choice of one or more documents from such a finite collection.

How much choice is involved? Certainly it is much easier to choose one item from a collection of ten than it is to choose one item from a collection of ten thousand. How much easier? Also, if it is easier to make the first kind of choice, a simpler coding should suffice for the smaller collection. Fortunately, these matters have a very simple and satisfactory quantitative formulation.

The choice of an item from a collection of two involves only a single two-valued decision: choose the first, or choose the second. We can designate the two values by 0 and 1, and we can call these digits. It requires two such elementary decisions in series to choose one item from a collection of four. We first split the collection in half, and decide which half we want, then in that half we decide which of the two objects we will finally choose. Thus for a collection of four items, the choice which represents any one item can be specified by two digits in series, and in fact the four objects can be respectively represented by the symbols 00, 01, 10, and 11. It is noted that this is precisely the binary enumeration system<sup>9</sup> that is much used in the internal operation of contemporary electronic digital computers. To enumerate eight objects will require three digits. Note that  $2^3 = 8$ .

**Principle 5: CHOICE.** If there is a collection of no more than  $2^S$  objects, we can specify a unique choice of one of them by means of a symbol having no more than  $S$  digits, each representing an elementary two-valued decision.

Values of  $S$  for collections of several sizes are as follows:

Size of Collection	Number of Two-valued Decisions for Choice
10	4
1,000	10
100,000	17
1,000,000	20

<sup>8</sup> Compare: Shannon (1948) *ibid.* p. 379.

<sup>9</sup> Also called "dyadic numbers." Compare G. Birkhoff and S. McLane, "A Survey of Modern Algebra," pp. 33-34, Macmillan, New York, 1944.



A machine for the selection of tallies scans or reads the marks and spaces in the tally field. The machine actually compares the patterns found on the tally field with some pattern of marks or spaces held in the machine to define the selection. Each mark or space in the pattern of the machine represents one value of a possible two-valued decision, and if there are  $S'$  such marks, then the pattern has the capability of making a unique choice among as many as  $2^{S'}$  objects. If the size of the searching pattern  $S'$  is larger than it needs to be, as compared to the magnitudes of the choice among the actual number of items in the collection, the coding is inefficient.

Most machines and systems discussed in the literature are extremely inefficient. Therefore they are able to record fewer descriptors in their coding fields, and they require more apparatus for searching than might otherwise be the case. Samain<sup>10</sup> employs a selection pattern of 36 marks and spaces to select upon a single descriptor. Most of his examples of selection are by two or three descriptors, having combined patterns of 72 or 108 places respectively. He mentions collections of the order of one million pieces. Such collections would allow adequate choice by a selection pattern of only 20 places. If he always used two or more descriptors for selection, he could represent any descriptor by a pattern of only 10 places instead of his present 36.

These considerations give rise to the next principle.

**Principle 6: DESCRIPTOR LENGTH.** The number of effective marks or spaces in the pattern representing a descriptor should be set by the requirements of choice among the actual collection, and not by the size of the vocabulary as is the usual practice now.

Let us consider a collection of 4,000 documents which has a notched card information retrieval system with 1,000 descriptors in the vocabulary list. It is known in advance that all

selections from this collection will be stated with three or more descriptors acting together. How many marks or spaces are required for the descriptor patterns?

The conventional answer is that each descriptor must be given a pattern 10 places long, otherwise the descriptor patterns will repeat. This pattern length is ridiculous when analyzed according to the choice required by selection. Since three such patterns will always be used in selection, the total selective pattern would always be 30 places long.

A choice among 4,000 pieces really requires a selective pattern of only 12 places. The correct answer, therefore, is that the individual descriptor patterns need have only four marks or spaces.

A paradox now appears. The patterns of some descriptors may be duplicates of other descriptors, yet we can still define the choice of the tally we want. The paradox is resolved when we notice that in addition to the desired choice, there is a statistical possibility that some unwanted descriptor patterns on unwanted tallies will by chance duplicate the patterns being searched. When this happens, extra tally selections will occur, though their number will usually be inconsiderably small and their occurrence can be approximately predicted from the details of the coding. The phenomenon of such "extra tallies" is inextricably bound up with the adjustment of the length of descriptor patterns in this fashion to increase coding efficiency.

As a final consideration in this section, we note that the manner in which the descriptor code patterns are recorded in the tally field has an enormous influence on the complexity of the selector machine and upon the competence of the whole system for information retrieval. While these factors can be evaluated quantitatively for different manners of recording, all that will be done here is to describe the several simplest possibilities. There are two design choices to be made. First, the tally field can be subdivided into mutually exclusive subfields with each subfield taking only one descriptor pattern, or the

<sup>10</sup> Jacques Samain, "A New Apparatus for Classification and Selection of Documents and References by Perforated Cards," pp. 680-685, also pp. 158, 230, and 265 in "Reports and Papers Submitted, The Royal Society Scientific Information Conference, 21 June to 2 July 1948," The Royal Society, London, 1948. See also Samain, pp. 22-26, Rept. 17th Conf. Int. Fed. Docmn. (1947).



field can be left undivided and the descriptor patterns can be superimposed one on top of the other in the same field.

Design Choice 1: "Mutually exclusive subfields" vs. "superimposed codes."

Second, the system can be operated so that the selector looks for a given descriptor pattern in some invariant position in the field (e.g., the descriptor for "red" is always found in the third subfield), or the selector mechanism can be made more complicated so that it can search out descriptor patterns in many alternative (subfield) and positions in the field.

Design Choice 2: "Invariant position" vs. "alternative position" scanning.

For example, Samain employs mutually exclusive subfields with alternative position scanning. Alternative position scanning is a very powerful technique. However, it requires an elaborate and expensive sensing mechanism because of the need to search many alternative locations in the field.

Conventional, small-scale, notched card systems employ mutually exclusive subfields, but avoid the expense and complications in sensing mechanisms by using fixed locations for descriptor codes, "invariant position scanning." This compromise severely limits the usefulness of such information retrieval systems and makes them incapable of handling many problems.<sup>11</sup>

#### IV - PRINCIPLES AND PRACTICE OF ZATOCODING

Zatocoding can now be defined as a system for using machine-sorted tallies for information retrieval in which the coding system has superimposed codes and the selector employs invariant position scanning. The great advantage of this combination of design choices is that the powerful capabilities of alternative position scanning can effectively be attained while using a very simple invariant position scanning

machine. Moreover, Zatocoding follows each of the coding principles set out in the preceding section. Therefore, the Zatocoding system allows simple machine structure, gives the maximum efficiency of codes, gives minimum size of coding field, and allows maximum scanning speed for a given situation.

By means of an example, let us follow through the steps in setting up a Zatocoding information retrieval system.<sup>12</sup> To make the example quite definite, at each step we will give the appropriate numerical quantities along with the basic design formula when possible.

Example: We wish to design a retrieval system for a collection of 4,000 documents. There are 1,000 descriptors in the vocabulary. We wish to use as a tally a simple notched-edge card having only 40 positions in its coding field (Small field, Principle 1). From an analysis of our problem, we anticipate that, in almost all cases in which we wish to make a search for information in this collection, we can define the selection by at least three descriptors acting in combination.

The design proceeds as follows: The collection of 4,000 pieces is just less than  $2^{12}$ , so that the choice involved in selection will require a pattern of 12 operating positions (Choice, Principle 5). Almost always, three or more descriptors will be used in selection, so a single descriptor pattern need have only  $12/3 = 4$  marks in its code pattern (Descriptor length, Principle 6).

Zatocoding uses no subfields. The whole field receives the codes, and so the marks of the individual code patterns are superimposed one after the other on the field. The spaces of one code pattern may be filled up by the marks of another code pattern. Therefore, only the marks in the pattern have an invariant meaning. A Zatocode is then really a pattern of marks only, not of marks and spaces. In our example each individual pattern must contain four marks. Since the field has 40 positions in it, the possible number of different patterns of four marks is given by the number of combinations of 40 things taken four at a

<sup>11</sup>C. N. Mooers, "Zatocoding for Punched Cards," Zator Technical Bulletin No. 30, pp. 6-9, Zator Company, Boston, 1950. This contains a complete discussion of the effects of the compromise.

<sup>12</sup>Ibid. pp. 9-19.



time, which is precisely 91,390. However, only 1,000 patterns out of this set are needed, and the question is how to choose them.

It is imperative, for the successful operation of Zatocoding, that these patterns (assigned to the descriptors) be chosen by some mechanically random process from the set of all patterns (Random patterns, Principle 3). The reason for the randomness is that the marks from the successive patterns overlap and intermingle to some extent, as they are put into the field, with the marks that are already there. This intermingling is at a minimum only when the patterns are random patterns. There is no possible systematic way of assigning patterns to descriptors which is better than by completely random assignment. In fact, several systematic assignments that have been tried gave impossibly bad performance.

In order to assign Zatocoding patterns to the descriptors, we proceed as follows. The patterns are first generated. Balls numbered from one to 40 are mixed and then drawn four at a time from a container, to be replaced for the next random draw. These four numbers are an appropriate random pattern. A list of such random patterns is made. We already have a vocabulary list of descriptors. It doesn't matter in what order they are. The random patterns are then assigned to descriptors by giving the first pattern to the first descriptor, the second to the second, and so on for all the descriptors. That is all, for code assignment.

I am often asked whether the patterns of marks and spaces cannot be assigned according to some analogy to the successive digits of a decimal classification symbolism. Such a method seems attractive only until it is analyzed with respect to its consequences with superimposed codes. Because similar marks in the patterns would overlap with undue frequency, the method is worthless. Random patterns must be used with Zatocoding.

Since the Zatocoding patterns are placed in the same coordinate position in the single coding field of a tally, how do we know when the coding field is full and can hold no more patterns? With mutually exclusive subfields, the situation is very simple. Only as many codes can be recorded as there are subfields. A mathematical analysis of the Zatocoding situation shows that the optimum amount of informa-

tion is carried in the tally field when around 50 per cent of the field positions are marked by descriptor codes. This result confirms, by another method, our conclusion of the previous section (Fifty per cent, Principle 2) concerning maximum utilization.

To return to our example, the codes have four marks apiece, and are recorded or notched into a field of 40 positions. Therefore, it would seem that we could record  $\frac{(4)(40)}{4} = 5$  different

codes. Actually because the codes overlap, we can place more codes than this in the field before the density of marks reaches 50 per cent. The rule is that with random codes the sum of the marks of all the codes is equal to 69 per cent of the number of field positions when there is an average density of marks of 50 per cent. In our example, the number of Zatocode patterns that can be notched into the card is thus  $\frac{(0.69)(40)}{4} = 6.9$  or effectively 7 patterns.

Machine Zatocoding selection of the patterns in the tally fields occurs when each mark in the selection-defining pattern is matched by a corresponding mark at the same position in the field of the selected tally. This is illustrated in Fig. 1. The tally field may have more marks than the selective pattern, but it cannot have fewer and be selected. This type of selection is called "selection by pattern inclusion," and the selective pattern is "included" in the field pattern. Zatocoding selections are made usually according to several descriptor patterns operating simultaneously. For instance, the selection might be according to "flash," "camera," and "selective device." Only those tallies which bear the marks of all three of these descriptors are specified for selection. In the terminology of symbolic logic, this selection is according to the "logical product" of the three descriptors.

It is expected that there may be other unwanted tallies which bear none of the desired descriptors, but whose patterns by chance combine in such a way as to imitate the selective pattern. These produce "extra" selected tallies. By proper design of the Zatocoding system according to the rules above, these extra tallies are made inconsequential. Desired tallies are never excluded by such chance operations; instead, a few other tallies may be included in the selection. This is



all right. By application of probability theory, the average number of extra tallies to be expected can be predicted. If the design rules given here are exactly followed, only one extra tally per selection will occur on the average whatever the size of the collection.

Control of the number of extra tallies is accomplished through the length of the pattern which defines the selection of the wanted and the rejection of the unwanted tallies. The logic is as follows: The selective pattern always matches the field pattern of the desired tallies, and they are always selected. We can eliminate them from our consideration of how the extras are rejected. All the tally fields are coded with random codes having an average density of marks of 50 per cent. A single mark in the selective pattern will reject by chance only half of the unwanted tallies in the collection. It will accept the others. The second mark in the selective pattern will in turn reject half the unwanted tallies that were accepted by the first mark. This cuts the number of unwanted tallies to one quarter, and so on. Each successive mark added to the selective pattern improves the accuracy of the rejection by a factor of one half. In the example, the selective pattern of 12 marks can be expected to exclude all of the unwanted tallies in the collection of 4,000 except one, on the average. We note also that, if we were forced to make a search defined by only two, instead of three, descriptors, the selective pattern of only 8 marks would allow possibly 16 extra tallies to appear. These can easily be recognized among the desired tallies, and can be discarded.

The same design principles apply to any other size of collection, design of tally, or machine. A collection of a hundred million ( $10^8$ ) items would take a Zatocoding pattern of 27 marks for a descriptor to give almost perfect

performance (with an average of one extra tally on a single-descriptor selection. A Hollerith card could take 25 such descriptors. Selection could be made on any combination of descriptors, or on single descriptors, taken from a vocabulary of unlimited size. By comparison, the more usual methods of coding, with numerical codes and alternative subfield position scanning, results in a very complex machine and a vocabulary limited to 1,000 descriptors at most.

The extreme simplicity of Zatocoding selection, employing pattern inclusion selection and invariant position scanning, reflects in a corresponding simplicity and economy of the selective machine. At ordinary speeds of tally scanning (in the order of 1000 per minute) a simple mechanical grid, sensing notches in the edge of cards in a pack, is completely adequate. By going to more elaborate technologies, of the kind now used for television broadcasting from movie films, more than 1,000 tallies can be sensed per second. By several further modifications in the manner of use of these electron-optical scanning devices, selection speeds in the order of 1,000,000 tallies per second are apparently possible. In another paper<sup>13</sup> the author has described such a super-speed machine, calling it a DOKEN or "documentary engine," and has discussed the associated organs appropriate to such scanning speeds.

In conclusion, where the simplest and highest speed selector is desired for sorting upon the logical product of descriptors, Zatocoding is preferred because it realizes the maximum efficiency in coding. It achieves this efficiency by matching the intellectual choice represented by the selective descriptors to the choice represented by the number of items in the collection. The innovations of Zatocoding are the use of superimposed random codes, and a statistical prediction and control of the associated phenomena of extra selected tallies.

<sup>13</sup>C. N. Mooers, "Making Information Retrieval Pay," paper, 118th Meeting American Chemical Society, Chicago, September 1950. Published as Zator Technical Bulletin No. 55, Zator Company, Boston, 1951.



## MULTIPLE CODING AND THE RAPID SELECTOR

CARL S. WISE AND JAMES W. PERRY

### INTRODUCTION

Documentary information — the record of human experience — is essentially polydimensional in character. An historical event, for example, has such dimensions as time, place, persons and organizations involved, and nature of action occurring. The record of a surgical operation is concerned with such different types of variables as the patient's symptoms, the pathological conditions thereby deduced, treatment, and the result of treatment. A chemist describes his experiments in terms of the interacting materials, the reaction conditions, the substances produced and accompanying effects, such as the evolution of heat.

Our use of recorded information is also, in the great majority of instances, polydimensional in character. An historian will rarely be interested in all facts recorded about a given country, province or city, nor is it likely that he will wish to be informed of all historical events which are recorded as happening during a given time interval. It is much more probable that he will be seeking records dealing with the history of some one country during a certain period of time. Similarly a physician will ordinarily not wish to know about all patients having a given set of symptoms, or all instances in which a given treatment was prescribed. It is much more likely that his interest will be centered on the combination of some certain treatment and a given set of symptoms. A chemist will rarely be interested in all syntheses carried out under a given set of conditions, e.g. high pressure. He is much more likely to be interested in the effect of a certain set of reaction conditions on individual compounds or groups of compounds.

### INDEXING AND CLASSIFYING METHODS

As discussed in a recent paper,<sup>1</sup> the polydimensional nature of recorded information is also evident when one observes how subject indexes are constructed. Thus in Chemical Abstracts — recently characterized<sup>2</sup> as having "the best indexing which has been achieved anywhere to date" — index entries almost without exception relate to some combination of entities, concepts and processes. This becomes clearer on considering a typical group of entries.<sup>3</sup>

Lignocellulose, adhesive extenders from, 8737i  
defibering of, P 1980c, P 2436cd.  
as filler for urea-HCHO condensation products, P 6865d.  
hydrolysis of, P 405b.  
hydrolysis of, to pentosans and hexosans from, P 5593b.

<sup>1</sup> "Conventional and Mechanized Search Methods." S. W. Cochran and J. W. Perry. *Ind. and Eng. Chem.* In press.

<sup>2</sup> "The Rapid Selector." Ralph R. Shaw. *J. of Documentation* 5, 164-171 (1949).

<sup>3</sup> *Chem. Abstrs.* 43, 10667 (1949).



laminated board from, P 3197c.

phenol condensation products from, P 1217a.

supersonic vibrations in sepn. of cellulose in, P 404i.

Classification, another widely used method of organizing information, uses different types of criteria for establishing classes and sub-classes. Synthetic resins, for example, are classified<sup>4</sup> on the basis of chemical structure, physical properties, processing methods, practical applications, etc. A classification system consists, in essence, of some permutation of a given type of criteria.

In constructing an index, it is not practical to provide separate entries for every combination of entities, concepts and operations mentioned in the material being indexed. If this were attempted, the index would be too bulky. Nor is it practical to establish as separate classes and sub-classes every possible permutation of all basic criteria used in classification. If this were attempted, the resulting complexity of the system would defeat its own purpose. Skill in indexing and classifying consists, to a large degree, in foreseeing and selecting from all possible combinations of entities, concepts and operations, those which appear to be most likely to prove useful to the user of the index or classification scheme.

No amount of human skill in devising indexes and classification schemes, however, can anticipate future trends in viewpoint and in research. Yet it is precisely in the direction of the unexpected correlation and the surprising result, that the most spectacular progress is made.

### CONCEPT CODING

A start toward overcoming the practical restrictions imposed on conventional indexing and classifying methods has been made during recent years by the application of punched cards.<sup>5</sup> The methods developed for using punched cards — though differing considerably in detail — have one feature in common; different holes, or combinations of holes, are used to indicate different types of characteristic variables. As a consequence, sorting operations can be directed to any one variable (e.g. geographical location, or time interval) or combination of variables. Punched cards have permitted us to take the first steps toward developing systems of information analysis capable of permitting searches to be directed to new, unforeseen combinations of entities, concepts and operations. It is the purpose of this paper to point out how one of the coding methods developed with punched cards might be applied to the rapid selector which has been under test at the Library of the Department of Agriculture during the past year.<sup>6</sup>

The coding method to be considered, as developed by one of us (CSW) for "Key-sort" cards, permits coding to be based on a vocabulary of 456,976 concepts, of which any sixteen (or less) may be coded on a single card. No restrictions whatever are imposed on the nature of the concepts coded on any one card. This method of coding resembles

<sup>4</sup> U. S. Patent Office, *Manual of Classification*, Amended April 1948, Washington, D. C.

<sup>5</sup> *Bibliography on the Uses of Punched Cards*. L. Ferris, K. Taylor and J. W. Perry, 2d Edition. Office of the Secretary, American Chemical Society, Washington, D. C.



conventional indexing insofar as it may make use of an extensive vocabulary. It differs, however, from indexing in several respects. Perhaps the most important of these is the ability to direct a search to any one concept or to any combination thereof. Another distinguishing feature is the fact that the card coding method does not scatter entries throughout an alphabetic register as an index does, but punches the concepts pertaining to a given item of information in a single card and thus keeps concepts pertaining to a given item grouped together. The fact that only sixteen concepts can be indicated on a single card has not proved troublesome in actual experience with punched cards.

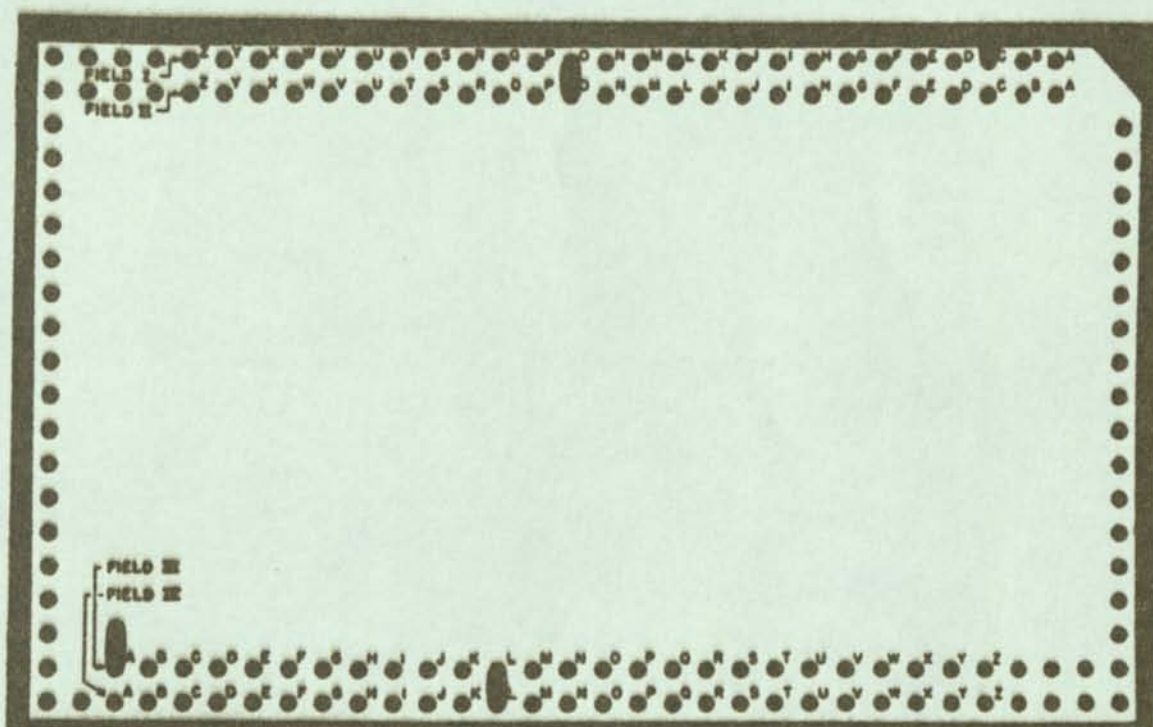


FIGURE I

The mechanics of this coding method are quite simple. Each concept is assigned a code designation of four letters. Thus "starch" might be assigned the code STAR and "Applied Chemical and Dye Corp." the code ACAD. The punching of these code designations is carried out using four rows ("fields") of twenty-six holes each. The four rows of holes are used to code the four successive letters in the code designations with the first letter always punched in the first row, the second letter in the second row, etc. Figure I shows a card punched for the code designation COAL. The punching in each of the rows of holes is accomplished in such a fashion that the sorting operation results in a downward motion of all cards corresponding to the code being searched.

Let us consider how the following abstract might be coded.



## ABSTRACT

Ion exchangers in sugar-juice purification. Mario Garino (Univ. Genoa, Italy). *Ind. saccar. ital.* 41, 103-5 (1948). — In some expts. made on juice of Italian manuf., it was observed that to eliminate 1 g. of nonsugar substance, it is necessary to use 15.5g. of resin. — G. A. B.

CODING OF ABSTRACT (from *Chemical Abstracts*, 42, 921/g (1948)).

Subject Headings	Coding of Subject Headings			
M. Garino (the author).....	M	G	A	R
Ind. saccar. ital.....	I	S	I	T
Italy.....	I	T	A	L
Sugar-juice.....	S	U	J	U
Nonsugars.....	N	S	U	G
Ion-exchange.....	I	O	N	E
Resins.....	R	E	S	I

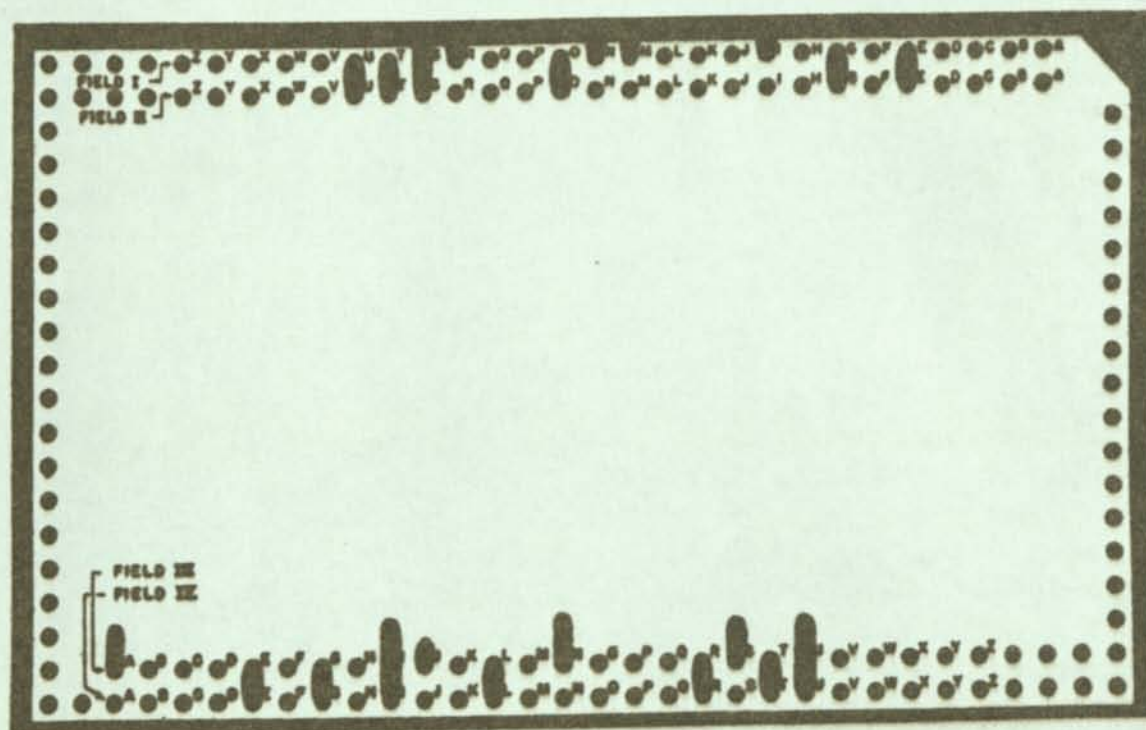


FIGURE II

A card with corresponding punching is shown in Figure II. If it were desired, the coding could be extended to include additional entries, e.g. PURI for purification.

With regard to searching operations, it is obvious that this card will be selected by a search directed to any one of the codes punched in the card. It is perhaps a little less obvious that a search directed to the combination "Ion-exchange," "Resin," "Sugar-juice," "Non-sugar" does not require that each of the four code combinations (viz.



IONE, RESI, SUJU and NSUG) be searched separately. Instead, all cards pertaining to the desired combination of concepts will respond to a searching operation directed to the code combination IEJG (built up from IONE, RESI, SUJU and NSUG). Alternately one might use the combination SSSE (from SUJU, NSUG, RESI and IONE). The possibility of using more than four sorting needles is also worthy of mention. Thus we might use six needles to search for N and R in the first row of holes (NSUG and RESI), S and U in the second row (NSUG and SUJU), N in the third row (IONE) and U in the fourth row (SUJU).

As already noted, it is not practical to punch more than sixteen four-letter codes on any one card. The reason for this is that the chance of undesirable interference between codes becomes excessive. How this arises can be seen by referring to the coding of our example abstract as shown in *Figure II*. This card will respond to a sort directed to the code designation STAR (SUJU, ITAL, ITAL, MGAR). If we use STAR as our code designation for "starch," then our example card will appear as an unwanted, extra card in a search directed to STAR alone. A search directed to "starch" (STAR) and "enzyme" (ENZY) will, however, not result in undesirable selection of our unwanted card. It can be stated as a general rule that the probability of unwanted extra cards appearing drops off rapidly as the number of codes being searched increases. A mathematical analysis of the probability relationships has been made<sup>6</sup> and will be published<sup>7</sup> in the near future. Although the mathematics is not difficult, the complete analysis is rather lengthy. We shall limit ourselves to stating a few of the results.

If nine entries were punched in our cards, then the mathematical analysis predicts that a searching operation directed to any four-letter code designation will result, on an average, in seven unwanted extra cards being selected per thousand cards sorted. If the sorting is directed to two of the four-letter code designations, then the probability of extra cards drops to less than one per ten thousand cards sorted. If three code designations are used in sorting our expectation of obtaining an extra card practically vanishes — it is less than one extra card per million being searched.

Similarly if we punch sixteen entries in our cards, then in a sort directed to a single four-letter code designation the mathematical analysis predicts that we may expect less than three extra cards per hundred being searched. As before the number of extra cards drops rapidly if our searching operations are directed to two and three four-letter code designations. The corresponding expectations of extra cards are less than one per thousand and less than four per hundred thousand, respectively.

Practical experience<sup>7,8</sup> with "Keysort" cards coded by this general method has been well in line with these predictions concerning extra cards.

<sup>6</sup> "Generalized Punched-Card Codes for Chemical Bibliographies." C. S. Wise. Paper presented at the 112th national meeting of the American Chemical Society, New York, N. Y., September, 1947.

<sup>7</sup> "Chemical Literature Studies. Some Mathematical Possibilities in Mechanical Indexing and Sorting Systems." C. S. Wise. Paper presented at the 115th national meeting of the American Chemical Society, San Francisco, Calif., April, 1949.

<sup>8</sup> *Punched Cards, Their Application to Science and Industry*. Edited by R. S. Casey and J. W. Perry, Reinhold Publishing Co., New York. In press.

<sup>9</sup> "A Practical Application of a Punched-Card System Utilizing the Superposition of Codes." A. F. Isbell. Paper presented at the 114th national meeting of the American Chemical Society, Portland, Ore., September, 1948.



## APPLICATION TO THE RAPID SELECTOR

According to information<sup>9, 10</sup> available to us, the rapid selector employs an optical-electronic system for scanning a reel of motion picture film on which are entered both abstracts and corresponding index entries. The abstracts are entered by microphotography one after the other on one longitudinal half of the film. They are indexed in conventional fashion<sup>2</sup> and a seven-digit number assigned to each index entry. The other longitudinal half of the film provides space for entering these seven-digit numbers as patterns of transparent and opaque dots. The code numbers corresponding to the index entries for a given abstract are entered one after another on the film as a group. Abstracts are selected by interrogating the device with any one of the characterizing seven-digit code numbers from the index.

As set up at present, the rapid selector machine is a device for high speed consultation of conventional indexes. In other words, those restrictions, which publication of indexes on printed sheets has in the past imposed on indexing methods, have been carried over into the *modus operandi* of the rapid selector. It is our purpose to point out one possibility of overcoming these restrictions by a radical change in the coding method together with a minor alteration of the rapid selector's scanning system. In order to understand this possibility, it is necessary to consider the rapid selector's scanning operation in a little more detail.

A coding area as set up at present measures 0.280 inches by 0.340 inches. As shown in *Figure III*, this area is divided into twelve rows each made up of eighteen squares. Two adjacent rows of squares are used to code each seven-digit number. The coding of any given seven-digit number is a uniquely determined pattern of opaque and transparent squares as shown in *Figure IV*. Before starting a search, a mask is prepared which has holes punched corresponding to the opaque squares in the pattern being sought. During the searching operation light passes through both the mask and the code areas on the moving film until such time that the punching in the mask corresponds exactly to the number being sought. This momentary blackout, detected with the aid of one or more photocells, trips an electronic circuit and thus, with the aid of high-speed flash photography, takes a picture of the abstract corresponding to the desired index entry. Since only one searching mask is used, it is possible, with the existing arrangement, to search for only one index entry. In other words this arrangement does not permit searches to be directed to combinations of index entries.

It would be possible to search for combinations by adapting to the rapid selector the form of concept coding described above with "Keysort" cards. Such adaptation might be done in a variety of ways. We shall limit ourselves at this time to describing only one particularly simple form of adaptation.

As already noted, the coding area in the rapid selector is being used at present to

<sup>9</sup> Report for the Microfilm Rapid Selector. Engineering Research Associates, Inc., St. Paul, Minn. and Arlington, Va.

<sup>10</sup> "Microfilm Selection Equipment in Information Work." H. T. Engstrom. *Ind. and Eng. Chem.* In press.



code six numbers of seven digits each in six double rows of squares with eighteen squares in each row. Scanning is directed to one double row of holes representing some one seven-digit number. Two changes would be required. In the first place, the searching mask would have to be extended to cover all the squares used for coding. It would

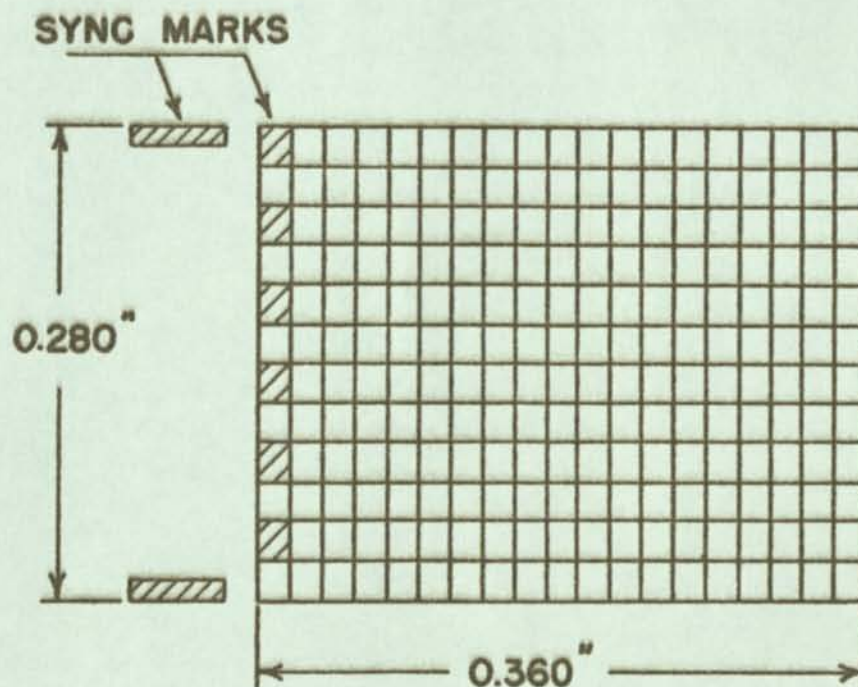


FIGURE III

seem likely that this would involve relatively minor changes in the optical portion of the scanning device. The electronic trip circuit and the photoflash recording of selected items would scarcely require essential change. Secondly, the assignment of significance to the squares in the coding area would have to be changed. A simple setup, which



FIGURE IV

illustrates the underlying principle, would be the following. Let us agree to restrict our coding to banks of thirteen squares each in the twelve rows of holes. Then in each double row of holes we can assign one single square to a single letter of the alphabet. In six double rows of squares, we can code six different letters, just as on the "Keysort" card we coded four different letters. This would permit us to use a six-letter code



designation for each entry used in indexing. The operation equivalent to punching our "Keysort" card would be rendering opaque an appropriate square in the coding area. Our searching mask, as at present, would be punched for those code designations being sought. But this searching mask could be punched for a plurality of code designations, with the certainty that if all the letters in the codes being sought had not been entered in any one coding area, light would pass through both the coding area and the mask and the corresponding abstract would not be selected. By being made to operate in this fashion, the rapid selector could be used to search for combinations of concepts not specifically envisaged at the time of indexing.

It is instructive to consider the coding capability of the simple system outlined above. In the first place, such a code using six letters has available 308,915,776 code designations. As with our "Keysort" cards, a plurality of these coding combinations can be entered in any one coding area. For reasons to be explained elsewhere,<sup>7</sup> there would be advantages in not permitting the number of code combinations entered in one field to exceed sixteen. If this were done, mathematical analysis predicts that a search directed to one six-letter code could be expected to produce less than one unwanted extra abstract per hundred being searched. If the search were directed to two six-letter code designations the expectation of an unwanted abstract drops to less than one in ten thousand searched. If the number of six-letter code designations is increased to three, then the probability of an unwanted abstract drops to less than one in a million. It should be recalled in this connection that the simple coding system outlined above makes use of only thirteen of the eighteen squares in each of the twelve rows. By proper use of the remaining sixty squares in each coding field the above mentioned expectations of obtaining unwanted, extra abstracts can be greatly improved. We shall not attempt at this time to describe in detail how this might be done. One possibility would be to assign digits from one to ten to the previously unassigned squares in two adjacent rows of squares and to use these extra digits in the coding.

### RESUME

The simple coding scheme proposed greatly increases the number of code designations available, increases the number of entries that can be made in any one code field and permits extension of the scope of search from a single coded entry up to combinations of at least sixteen coded entries. It appears likely that these advantages can be gained without the need for extensive change in the present design of the rapid selector.



## **Multiple Word Coding vs. Random Coding for the Rapid Selector.**

### **A Reply to Calvin N. Mooers**

In a previous communication (1) an improved method of coding for the Rapid Selector was suggested. Since then, a rather severe criticism of this has been made by Calvin N. Mooers (2). Thanks are due to him for pointing out that our paper may have given a slightly misleading impression to some people. However, this was not caused by our inadequate statistical analysis, as Mooers seems to feel, but instead resulted from a too brief treatment of the word coding method. Although these matters have already been treated elsewhere (3), it now seems desirable to give a more detailed summary in the pages of *American Documentation*. Furthermore, in answer to the spirited critique of Mooers, several new details will be mentioned.

The main objections of Mooers seem to be only two in number. First, the multiple coding method is accused of being inflexible, and secondly, he believes that the non-random nature of the English alphabet is a fatal defect. These objections will now be discussed in detail.

#### **FLEXIBILITY**

Mooers believes word coding to be inflexible in that it does not permit variation in the number of marks in certain of the code patterns. That is, he thinks it is not possible to use a "two-letter, four-letter, or even fifteen-letter code for certain of the ideas". Actually the writer has already published the details of a simultaneous use of a three-letter and a five-letter code. The same flexibility has been used in a word coding method for the Dyson cipher for chemical compounds, which may use anything from one letter up to twenty or more. Furthermore, in this same paper (3) is explained the important new principle of indicating relationships between words, as well as the simultaneous indication of combinations of words. Or, to put it in mathematical terms, multiple word coding is flexible enough to record combinations of ideas, as does Zatocoding, but it can also record permutations of ideas.

Not only words, but complete sentences can be coded by word coding techniques. Perhaps Mooers' objection to our use of the word "poly-dimensional" was due to his former lack of knowledge concerning the linearly-independent nature of Dyson coding or of sentence coding. In fairness to Mooers it should be pointed out that the Punched Card book was still in press at the time of his October 1950 criticism.

Word coding also is flexible enough to be adaptable directly to the literature without being dependent upon the continuous use of a coding dictionary. Unfortunately Zatocoding does require this extra and unnecessary step. Furthermore the ideas and the codes selected to represent them are totally unrelated, since the latter are obtained by a mechanical random process (4). At this point it should be stressed that there are logical reasons to assume that a code book will have as many headings (main places to look) as an index directly to the words themselves. As Crane and Bernier (5) put it, "Consider a single heading, 2-Butene. Where could information about this compound be put, except under 2-Butene or a synonym? ... In passing, it should be noted that if some means of simplification of an index to a mechanized system should be found and the number of headings reduced, then this same improvement could probably be applied to an index directly to the literature."

Moreover a random system like Zatocoding has a double handicap, because it requires two translations, once when the material is first coded, and again when the information retrieval step occurs. Undoubtedly this causes more inefficiency than does the use of a non-random alphabet in word coding.

#### **A RANDOM DISTRIBUTION OF THE ALPHABET FOR CODING PURPOSES.**

Mooers obtains a  $Q$  of 0.619 when  $n = 16$  on a non-random alphabet and 75% of this or a  $Q$  of 0.466 for an equally distributed alphabet. This can be accepted as substantially correct.



For example it has been found by actual experimentation (3) that with respect to extra cards a 26 letter non-random alphabet is equivalent to a theoretical 19 letter random alphabet (i.e. 73% of 26). This means that we can code only 75% as many words if we confine ourselves to using an unmodified 26-letter alphabet as we could if we had a perfectly random 26 letter system.

Now in spite of the non-random distribution of letters in pure English text, there are at least two good reasons why this is not a fatal defect.

(1) Word Coding Can Use a Modified English Text.

In actual practice, word coding is more complex than the simple outline given in our article on the Rapid Selector. For example, it uses Arabic numerals, Greek letters, Dyson symbols, punctuation marks, superscripts and subscripts (6). This being the case, the calculations by Mooers concerning our method are all based on a false assumption. In reality, our values are on the conservative side. In the interest of simplicity in explaining both the code and the somewhat involved mathematics, we deliberately restricted ourselves to an alphabet of only 26 letters per field, instead of the much larger possibility of 36 per field.

(2) Word Coding Can Use a Magnified and Random-type Alphabet.

There is one other detail that should have been stressed. This is the possibility of converting the alphabet into a theoretical random one. This has been done by others and could easily be adapted to word coding for the Rapid Selector. The clearest description of the general idea is given in a discussion of triangular methods of alphabetical coding by Cox, Casey, and Bailey (7). They describe it as follows: "To facilitate alphabetical coding, we made a study of the alphabetical distribution in the first, second, and third letters of proper names, as they occur in the Author Index of Chemical Abstracts, in American Men of Science, and in two small private bibliographies (Tables 2, 3 and 4).... The six-position double-hole field shown in Figure 6 can be used to code the third letter. Table 4 shows that E, L, N, and R occur most frequently as the third letter in proper names. The six-position double-hole field can code 30 symbols, four

more than the 26 letters of the alphabet, so the extra places are used to split E, L, N, and R according to whether the fourth letter of the name, which follows E, L, N, R, is in the first half of the alphabet or the second half."

In this same paper, Cox et al. also use an alphabet shorter than 26 letters which they make by combining some of the letters. Thus the 26 letter alphabet, which admittedly is not random, may either be expanded or contracted, and in either case the resulting modified "alphabet" approaches the theoretical random-type.

This possibility of having more or less than 26 positions per field led C. S. Wise to make his extensive mathematical calculations (3) into the proper way to divide the total number of holes into fields. Thus if the information problem is too large for a 26 letter alphabet, it can be expanded into a statistically random "alphabet" of 30, 100, or even 1000 positions per field by the general method of Cox, Casey, and Bailey. The only difference is that the word coding method uses combinations of directly coded fields instead of the combinations of selective-triangular-type fields used by the above authors.

## CONCLUSIONS

Word coding has been shown to be extremely flexible, faster than a random method requiring dependence on a coding dictionary, and adaptable to a completely random alphabetical system. It also can show relationships between words or ideas, in addition to showing mere combinations of ideas. For these reasons the writer believes it to be generally more efficient than Zatocoding. However this single field random sorting method used by Mooers does make use of sound statistics, as has already been pointed out (3). Its weak points are in failing to have some of the practical advantages obtainable by multiple word coding.

Carl S. Wise

## Literature Cited:

1. C.S.Wise and J.W.Perry, "Multiple Coding and the Rapid Selector", v. I, pp. 76-83, American Documentation (April 1950)
2. C.N.Mooers, "Coding, Information Retrieval,



- and the Rapid Selector", v. I, pp. 225-229, American Documentation (Oct. 1950)
3. C.S. Wise, "Mathematical Analysis of Coding Systems", Chapter 20 of Punched Cards, Their Applications to Science and Industry. . Edited by R. S. Casey and J. W. Perry, Reinhold Publishing Co., New York. 1951.
  4. C.N. Mooers, "Zatocoding Applied to Mechanical Organization of Knowledge", v. II, pp. 20-32, American Documentation (Jan. 1951)
  5. E.J. Crane and C.L. Bernier, "Indexing and Index-Searching", Chapter 23 of Punched Cards. Ibid.
  6. C.S. Wise, "A Punched-Card File Based on Word Coding", Chapter 6 of Punched Cards. Ibid.
  7. G.J. Cox, R.S. Casey and C.F. Bailey, "Recent Developments in Keysort Cards", J.Chem. Educ. 24, 65-70 (Feb. 1947).

## *A Report on the Operation of Auxiliary Publication Service*

During the period of January 1 to November 14, 1952, 350 additional documents were accessioned, making a total now on file of 2,742.

During the period, 537 copies of documents were ordered and serviced.

The following journals and institutions deposited documents during the period:

ADJUTANT GENERAL'S OFFICE  
AMERICAN JOURNAL OF PHYSICAL  
ANTHROPOLOGY  
AMERICAN JOURNAL OF PHYSIOLOGY  
AMERICAN JOURNAL OF SOCIOLOGY  
AMERICAN PSYCHOLOGIST  
BRITTONIA  
CHEMICAL ENGINEERING PROGRESS  
DR. HENRY FIELD  
GENETIC PSYCHOLOGY MONOGRAPHS  
GENETICS  
HUMAN RELATIONS  
INDUSTRIAL AND ENGINEERING  
CHEMISTRY  
JOURNAL OF ABNORMAL AND SOCIAL  
PSYCHOLOGY  
JOURNAL OF APPLIED MECHANICS  
JOURNAL OF APPLIED PHYSIOLOGY  
JOURNAL OF APPLIED PSYCHOLOGY  
JOURNAL OF CLINICAL INVESTIGATION  
JOURNAL OF COMPARATIVE AND  
PHYSIOLOGICAL PSYCHOLOGY  
JOURNAL OF CONSULTING PSYCHOLOGY  
JOURNAL OF EXPERIMENTAL  
PSYCHOLOGY  
JOURNAL OF GENERAL PSYCHOLOGY

JOURNAL OF HEREDITY  
JOURNAL OF PHYSICAL CHEMISTRY  
JOURNAL OF THE AMERICAN CHEMICAL  
SOCIETY  
JOURNAL OF THE ELECTROCHEMICAL  
SOCIETY  
JOURNAL OF THE OPTICAL SOCIETY  
DR. HEINZ MARKMANN  
MEDICAL CLINICS OF NORTH AMERICA  
NORTH CAROLINA STATE COLLEGE  
TRANSLATION SERVICE  
OKLAHOMA AGRICULTURE AND  
MECHANICAL COLLEGE  
THE PHYSICAL REVIEW  
PROCEEDINGS OF THE GENERAL  
DISCUSSION ON HEAT TRANSFER  
PROCEEDINGS OF THE INSTITUTE OF  
RADIO ENGINEERS  
PSYCHOLOGICAL SERVICE CENTER  
BULLETIN  
PSYCHOMETRIKA  
TAT NEWSLETTER  
UNIVERSITY OF MAINE STUDIES

This report supplements the paper entitled "15 years of Experience with Auxiliary Publication" presented to the Board of Trustees meeting July 25, 1951 and published in AMERICAN DOCUMENTATION, Vol. II, No. 2, and also the report on auxiliary publication dated January 29, 1952 presented to the annual meeting 1952.

Watson Davis



PAPERS AND PUBLICATIONS: January 1957 - March 1960  
Request by ZTB number and title

I. Papers on Information Retrieval Theory

MOOERS' LAW; OR, WHY SOME RETRIEVAL SYSTEMS ARE USED AND OTHERS ARE NOT.

ZTB No. 136 (December 1959), 2 pp.

An excerpt, with modifications, from the last section of "Information Retrieval Selection Study, Part II: Seven System Models," ZTB 133, Part II; RADC-TR-59-173, listed below.

Some retrieval systems, although technically rather poor, nevertheless receive intensive use, while other systems, sometimes technically very much better, receive very little customer use. Why? MOOERS' LAW states: "An information retrieval system will tend not to be used whenever it is more painful and troublesome for a customer to have information than for him not to have it." This assertion is discussed.

INFORMATION RETRIEVAL SELECTION STUDY. PART II: SEVEN SYSTEM MODELS, by Calvin N. Mooers.

ZTB No. 133, Part II; RADC-TR-59-173 (August 1959), 39 pp. (Work supported by the U. S. Air Force, Rome Air Development Center, through Contract AF 30(602)-1900.)

The problem of fitting information retrieval systems to the needs of the users is examined and a variety of information handling situations involving retrieval-like functions is considered. The system models are designed for: (1) the laboratory scientific information retrieval; (2) high output performance for many users; (3) high performance on the input side; (4) cooperative input; (5) routing, in which documents provide the "questions" and a person is the retrieval "answer"; (6) storage of facts alone; and (7) the machine in the role of a teacher. The last section discusses the factors that determine why some retrieval systems are used and others are not.

THE APPLICATION OF SIMPLE PATTERN INCLUSION SELECTION TO LARGE-SCALE INFORMATION RETRIEVAL SYSTEMS, by Calvin N. Mooers.

ZTB No. 131; RADC-TN-59-157; ASTIA AD No. 215 434 (April 1959), 20 pp. (Work supported by the U. S. Air Force, Rome Air Development Center, through Contract AF 30(602)-1900.)

Simple pattern inclusion selection, ordinarily described as "superimposed coding," is able to provide selection only according to the logical product of the prescribing descriptors, i.e., according to "AND". The advantages and limitations of this coding method are discussed. Conditions under which it can be advantageously used are stated, and code system design rules are given.

INFORMATION RETRIEVAL SELECTION STUDY. PART I: EXTENSIONS OF PATTERN INCLUSION SELECTION, by Calvin N. Mooers.

ZTB No. 133, Part I; RADC-TR-59-169 (August 1959), 39 pp. (Work supported by the U. S. Air Force, Rome Air Development Center, through Contract AF 30(602)-1900.)

Techniques for extending the capabilities of simple pattern inclusion selection are described. The extensions provide selection according to "OR", "NOT", "GREATER THAN", and "FOLLOWED BY". The required selective logic and its implications to machine complexity are discussed. It is shown that selection according to "NOT" is unable to produce the results ordinarily claimed for it; this result holds irrespective of the digital code used.

THE INTENSIVE SAMPLE TEST FOR THE OBJECTIVE EVALUATION OF THE PERFORMANCE OF INFORMATION RETRIEVAL SYSTEMS, by Calvin N. Mooers.

ZTB No. 132; RADC-TN-59-160 (August 1959), 20 pp. (Work supported by the U. S. Air Force, Rome Air Development Center, through Contract AF 30(602)-1900.)

The intensive sample test uses as its standard an approximation to a perfect information retrieval system. The hypothetical perfect system is defined as one in which every retrieval system customer has read in detail every document in the collection with respect to every question of possible interest to him. The test uses representative samples of customers and of documents. Each customer examines a subsample of documents, formulates questions to be answered by the documents he has examined, and rates question-document pairs as "crucial", "revelant" or "not revelant". The questions are given to the retrieval system operators, and their results are scored only with respect to the question-document pairs in the sample. Measures of performance and statistical considerations are discussed.

SOME MATHEMATICAL FUNDAMENTALS OF THE USE OF SYMBOLS IN INFORMATION RETRIEVAL, by Calvin N. Mooers.

ZTB No. 123; RADC-TN-59-133; ASTIA AD No. 213 782 (April 1959), 18 pp. (Paper presented at the International Conference on Information Processing, UNESCO, Paris, France, June 13-23, 1959. Work supported by the U. S. Air Force, Rome Air Development Center, through Contract AF 30(602)-1900.)

A lattice theory model and formalism is presented for the process of gaining information through observation, reporting the information, and retrieving it. A number of theorems result.



A MATHEMATICAL THEORY OF THE USE OF LANGUAGE  
SYMBOLS IN RETRIEVAL, by Calvin N. Mooers.

ZTB No. 122 (1959), 38 pp. (Preprint of a paper presented at Area 6 of the International Conference on Scientific Information, Washington, D. C., November 16-21, 1958; to be published in the Proceedings of the Conference.)

A topological model is presented which relates the language symbols of retrieval to the documents retrieved. The model is applied to families of retrieval systems: using (1) descriptors, (2) characters with hierarchy, and (3) characters with logic. The retrieval operation is represented by a transformation from a point in a retrieval space P to a point in a document space L. Two different retrieval transformations are defined.

RETRIEVAL BY THE METHOD OF PROXIMITY  
TRANSFORMATIONS, by Calvin N. Mooers.

ZTB No. 113 (February 1958), 41 pp. (Unpublished; a limited number of preliminary copies are available.)

A theory of retrieval for discovering documents that are similar in content to a given document, yet with the precise manner of the similarity being unspecified. Documents are represented by points in an abstract space. A measure of closeness or proximity is discussed which has the property that documents similar in subject content are close to the given document. The measure of similarity can be biased with respect to different subjects (such as physics or biology) through the use of pre-descriptors that perform proximity transformations on the points in the abstract space. A method of coding is described and areas of application are discussed.

SOME MATHEMATICAL IDEAS NEEDED FOR A RETRIEVAL  
THEORY, by Calvin N. Mooers.

ZTB No. 111 (September 1957), 36 pp. (Unpublished; a limited number of preliminary copies are available.)

A tutorial paper sketching some of the mathematical ideas to be used in the development of a mathematical theory of information retrieval. Topics included are: sets, elements, and objects; classes; defining relations and equivalence classes; partial ordering, disjointness and conjointness; duality, chain condition and level; cup and cap; trees, lattices, and Boolean lattices; transformation between sets; transformations between algebras; and spaces and open sets.

II. Papers on Inductive Inference

A PROGRESS REPORT ON MACHINES TO LEARN TO  
TRANSLATE LANGUAGES AND RETRIEVE INFORMATION,  
by R. J. Solomonoff.

ZTB No. 134; AFOSR-TN-59-646 (October 1959), 17 pp.  
(Work supported by the U. S. Air Force Office of  
Scientific Research, through Contract AF 49(638)-376.)

This paper discusses theoretical work which makes it possible for a machine to be programmed to learn to translate between two simple, synthetic languages, after it has been given a large set of correct examples in a suitable training situation. A system is described in which a machine would learn to assign descriptors or other search indices to documents after having been given a large set of examples. Limitations and problems are discussed. A unified method is proposed for resolving the difficulties of both routines.

PROGRESS REPORT: RESEARCH IN INDUCTIVE INFERENCE  
FOR THE YEAR ENDING 31 MARCH 1959, by R. J.  
Solomonoff.

ZTB No. 130; AFOSR-TN-59-218; ASTIA AD No. 216  
240 (May 1959), 12 pp. (Work supported by the U. S. Air  
Force Office of Scientific Research through Contract  
AF 49(638)-376.)

Research is reported on methods of discovering the grammars of phrase structure languages, on self-improving machines, and on approximation languages. The concept of "language" has been generalized to include patterns of many extremely diverse types.

THE MECHANIZATION OF LINGUISTIC LEARNING, by  
R. J. Solomonoff.

ZTB No. 125; AFOSR-TN-59-246; ASTIA AD No. 212 226  
(April 1959), 16 pp. (Paper presented at the Second International Congress on Cybernetics, Association for Cybernetics, Namur, Belgium, September 3-10, 1958. Work supported by the U. S. Air Force, Office of Scientific Research, through Contract AF 49(638)-376.)

Techniques for the mechanization of inductive inference have been extended to the discovery of the grammatical rules of certain elementary language types, including phrase structure languages, in a suitable training situation. It is also shown that in some idealized cases it is useful to consider the problem of learning to translate languages as a problem in discovering the grammatical rules of a new type of "generalized language."



**A NEW METHOD FOR DISCOVERING THE GRAMMARS OF PHRASE STRUCTURE LANGUAGES**, by R. J. Solomonoff.

ZTB No. 124; AFOSR-TN-59-110; ASTIA AD No. 210 390 (April 1959), 13 pp. (Paper presented at the International Conference on Information Processing, UNESCO, Paris, France, June 13-23, 1959. Work supported by the U. S. Air Force, Office of Scientific Research, through Contract AF 49(638)-376.)

A technique similar to the one described by Chomsky and Miller for finite state languages is used to give a complete grammatical description of a phrase structure language. This is discovered by a systematic process of deletion and reinsertion of phrases and pairs of phrases, and the use of a "teacher" or equivalent to determine if the resulting sentences are acceptable sentences in the language.

**AN INDUCTIVE INFERENCE MACHINE**, by R. J. Solomonoff.

ZTB No. 128 (1957), 7 pp. (Preprint of a paper presented at the 1957 Convention of the Institute of Radio Engineers, New York, and published in the Convention Record, Section on Information Theory. Mr. Solomonoff was at that time associated with Technical Research Group, New York.)

The machine is designed to operate as human beings seem to. Inductive inferences are made by classifying events and the outcome of events within suitable categories. Accuracy of inference is largely dependent upon how good the categories are. Categories are tested empirically, and new categories are formed by the operation of a small set of transformations and tested again. These are combined with other categories and tested, and this process is repeated over and over again. The behavior of a simplified machine in learning to perform some arithmetic operations on the basis of a set of correctly worked examples is analyzed in detail.

**III. MISCELLANEOUS**

**THE NEXT TWENTY YEARS IN INFORMATION RETRIEVAL: SOME GOALS AND PREDICTIONS**, by Calvin N. Mooers.

ZTB No. 121; AFOSR-TN-59-245; ASTIA AD No. 212 225 (March 1959), 18 pp. (Paper presented at the Western Joint Computer Conference "New Horizons in Computer Technology," San Francisco, March 3-5, 1959. Work supported in part by the U. S. Air Force Office of Scientific Research through Contract No. AF 49(638)-376.)

Future information retrieval machines will assign descriptors to text (a crude kind of mechanical translation), and assist the customer in using a retrieval system. Several forms of

education of the customer by machine are predicted: helping him formulate search requests, helping him read the documents uncovered in the search, and producing essays on any given subject upon request. Machines can become archival devices to store facts, not texts. Human-machine communication will become very important.

**THE DUFFER UNIT: THE REQUIREMENT FOR DESK-TOP KEYBOARD TRANSCRIPTION UNITS IN INTEGRATED MECHANIZED RETRIEVAL SYSTEMS**, by Calvin N. Mooers.

ZTB No. 126 (April 1959), 8 pp.

The proposed duffer unit is an individual desk-top input-output transcriber for permitting direct customer interrogation and access to the contents of a large-scale information retrieval system. In addition to present machines for input transcription by skilled typists and high-speed machine print-out, we need foolproof, relatively slow-speed keyboard input and tapeprinting output devices matched to the human who is a duffer at the keyboard. Each customer of the retrieval system should have his own duffer unit giving him direct access to the central information system for quick interrogation and rapid response.

← temporarily out of print.



# Z A T O R C O M P A N Y

140 1/2 MOUNT AUBURN STREET · CAMBRIDGE 38 · MASSACHUSETTS · TROWBRIDGE 6-6776

May 24, 1960

Mr. R. William Austad  
Research Engineer  
Computer Techniques Laboratory  
Stanford Research Institute  
Menlo Park, California

Dear Mr. Austad:

Your letter arrived just as Mr. Mooers was about to leave for a trip out of the country, so he was not able to answer in person.

Mr. Mooers asked me to write you that the formulas in your letter of May 19 are correct assuming that all descriptors are used with equal frequency and without correlation between descriptors. Empirical results with Zatocoding systems shows that the formulas tend to be conservative.

However, for non-uniform use of descriptors and for use of descriptors that end to be correlated with other descriptors, corrections may be necessary. Mr. Mooers said that he would have to discuss with you the details of your application and the point at which corrections would be required. He certainly could not do this in a letter, and his reports have never developed this point in sufficient detail to be helpful to you.

Mr. Mooers therefore suggests that you get in touch with him personally some time after June 28.

Very truly yours,

ZATOR COMPANY

*Charlotte D. Mooers*

Charlotte D. Mooers (Mrs. Calvin N.)



## SUPERIMPOSED PUNCHING OF NUMERICAL CODES ON HAND-SORTED PUNCH CARDS

JAMES W. PERRY\*

### Summary.

Both the users and processors of information--particularly in the realm of science and technology--have long been aware of the fact that the information in a patent, paper or similar document will be characterized by a variety of features, such as various entities (substances, devices, apparatus, etc.), their attributes, one or more processes or interactions, attendant circumstances and the results of interaction. The multidimensional nature of information has been the basis for successful application of sorting devices permitting information to be selected on the basis of any one feature or combination of features. (1) Experiments with newly developed IBM equipment, (2) have demonstrated that grouping concepts into arrays sometimes called abstraction ladders is a convenient method for making generic terminology available as a basis for conducting searches. The present paper outlines a punching system which would permit abstraction ladders to be coded for hand-sorted punched cards. The punching system is undergoing test at the present time. The results of these tests will be reported subsequently.

### Introduction.

Hand-sorted punched cards are supplied by the manufacturers with holes perforated in rows along the periphery of the card. Although the rows of holes along one or more edges of the card are often arranged two deep (occasionally three, or even four deep), for the moment we shall consider only those holes that form a row nearest an edge of the card. Any one of the holes so positioned may be rendered operative with regard to sorting, by cutting away the cardboard between the hole in question and the periphery of the card. This operation--usually referred to as punching--converts the hole into a notch. The sorting operation

consists of taking a group of cards, aligning them, inserting a sorting needle into some one hole, and then ruffling them so that the cards that have been punched (i.e. notched) at the hole will drop from the needle. (3)

Of themselves, the punching and sorting operations are mechanical manipulations devoid of inherent meaning. Intelligent use of the cards requires that meaning be attributed to the punching of a given hole. This may be done most simply by assigning some one meaning to the punching of a single hole. Thus Breger (4) attributed the following meaning to the punching of certain individual holes when building up a file of information on the geochemistry of coal formation.

#### Examples of Headings Used by Breger

Genesis (of coal, humus, petroleum, etc.)  
Peat  
Lignite  
Bituminous coal  
Lignin  
Wood  
Humus, humic acids, etc.

The possibilities of this simple approach--sometimes called direct coding--must not be underestimated. Since sorting operations can be directed to combinations of holes it is possible to select cards dealing with combinations of subjects. Thus, it is easy for Breger to select those cards dealing with "humus," "origin" and "bituminous" and in this way he can search out all information in his file dealing with the role of humus in the origin (or formation) of bituminous coal. On the other hand, there is no denying the fact that using a single hole for each subject severely limits the number of subjects that may be punched and are thus made available for analyzing information when building up the file. This limitation may well prove intolerable when dealing with complex subject matter.

\*Bjorksten Research Laboratories, Madison, Wisconsin; address at present, Massachusetts Institute of Technology, Cambridge, Mass.



A number of less simple punching schemes --all based on assigning meanings to different combinations of holes--have been devised. (3) One such scheme permits any one of the digits, 1 to 9, to be punched in a group of four holes. With this system, punching a single hole suffices to indicate the digits 1, 2, 4, or 7, while 3 is indicated by punching the holes used separately for 1 and 2 and 5 by punching the 1 and 4 holes, 6 by the 2 and 4 holes, 8 by the 1 and 7 holes and 9 by the 2 and 7 holes. Although having useful features, this punching scheme and its variations suffer from the limitation that, for any one card, only one numeral can be punched in the field of four holes. Thus 1 and 2 punched separately would be indistinguishable from the punching for 3 while a card punched for both 6 and 8 would respond to a search directed at any one of the four holes or any combination of them. It is true that this system can be used to punch very large numbers. Thus successive fields of four holes each can be used to indicate successive digits. For example, such fields--twenty-four holes in all--permit any one number up to a million to be punched in any one card. However, the limitation of one number per card severely restricts the usefulness of this type of punching scheme. In practice, this punching system has proved highly effective for coding such data as a serial number, e.g. of a machine part or patent, the date of a transaction, number of items sold, price per item, date of birth, publication date, volume and page number for a given paper. In general, this coding scheme works well for those types of data which require a single value to be entered on any one card.

If, however, it is desired to have available a large list of subject headings from which several may be selected to characterize the subject matter of some one item of information, then the coding systems described above have not proved entirely satisfactory. This paper is written to outline a simple coding scheme designed to permit any six of 99,999 code numbers to be entered on a single hand-sorted punched card. Each code number corresponds both to some specific subject heading and to at least two more generic headings. The punching scheme to be presented permits the organization of subject headings into arrays sometimes referred to as abstraction ladders.

A simple example suffices to suggest the possibilities. For purposes of analyzing information dealing with minerals, we might set up a number of broad groupings of terminology, one of which might be related to minerals themselves and to which the number 3 might be arbitrarily assigned. Under this main heading we might decide to set up "carbonaceous minerals" as a principal subgroup, with "coal" as a sub-subgroup and, subordinate thereto, "bituminous coal". The abstraction ladder might then be encoded as

Minerals	3---
Carbonaceous minerals	309--
Coal	3095-
Bituminous	30952

In assigning code numbers, it is well to take into account the number of subgroups that it may be advantageous to set up at any one level. Thus in our example, assignment of two digits to specify the subgroup "carbonaceous minerals" makes it possible for the code to accommodate in all 99 categories of equal rank with that subgroup, while assignment of one digit each to designate "coal" and "bituminous coal" provides the possibility of setting up 9 categories on each of these two levels. (In tests now in progress the zeros are being reserved for general categories, not further specified).

The organization of terminology into arrays for analysis of information will not be discussed in detail in this paper. Suffice it to say that experimental investigation both at the U.S. Patent Office and at MIT have demonstrated the usefulness of properly designed arrays of terminology for the analysis of information preparatory to searching by mechanical devices. (5)

With these arrays, using a five digit number, e.g. 30952, to code and punch a specific entry, such as "bituminous coal", automatically causes the card to be punched for more generic headings, in this case, 3095- for "coal", 309-- for "carbonaceous mineral", and 3---- for "mineral". In describing the punching system the initial digit (i.e. 3 in the code number 30952) will be called the first digit and successive digits are counted from left to right (thus, in our example 0 is the second digit, 9 the third, 5 the fourth and 2 the fifth).

The balance of this paper will be devoted to presenting the punching system which has been designed so that both more generic terminology,



e.g. "mineral", "carbonaceous mineral", and also more specific terminology, e.g. "coal", "bituminous coal", will be equally available, for defining and directing a search. After describing the system, its mathematical analysis will be summarized in order to provide indication of how the system may be expected to work.

The first digit in each code number is punched directly in a field of ten holes. (See Figure I). For the second digit a field of twenty holes is set up and this is divided into two subfields. The decision as to which subfield is to be used hinges on the first digit in the code number. The left-hand subfield is used if the first digit is 0-4 and the right-hand subfield if the first digit is 5-9. Once the proper subfield has been selected, the hole labeled to correspond with the second digit is sought out and punched. The third digit is punched directly in a field of ten holes. These three fields are located at the top edge of the card. The fourth and fifth digits are each punched in a field of twenty holes at the bottom of the card. Both these fields are divided into two subfields of ten holes each. The decision as to which subfield to use in punching the fourth and fifth digits is based on the preceding digit in exactly the same way as when punching the second digit in Field 2.

In punching the cards preparatory for sorting, the same fields are used for punching successive code numbers. The punching system will readily accommodate up to six code numbers. Figure II shows Fields 1-3 of three example cards punched, respectively, for the first three digits of one, two and six code numbers.

In considering the operating characteristics of this punching system, attention will be centered at first on Fields 1-3 in which the first three digits of six code numbers are punched.

As is evident from Figure II, considerable overlapping between code numbers may occur and this may result in "phantom" combinations causing a card to respond in an undesired fashion. Thus the card in Figure II, punched in Fields 1-3 for the numbers 718, 346, 893, 414, 508, 112, will be sorted out when a search in Fields 1-3 is directed to the number 318 even though the card was not punched for that number because the information entered on the card was not concerned with the subject

heading associated with 318. Such unwanted cards, if they should appear in sufficient number, might become troublesome. It is worthwhile to investigate mathematically the probability of such cards appearing.

In applying probability theory to this type of situation, Wise (6) has first assumed that there is equal probability that any one number will be punched. He then derived the following equation:

$$G = H - H \left( \frac{H-1}{H} \right)^X \quad (I)$$

where H is the total number of holes in a field; X is the number of digits entered in the field, i.e. the number of instructions to punch the card (In our case, we are investigating the situation when six code numbers are punched. Hence  $X = 6$ ); and G is the number of holes actually punched. In general, G will be smaller than X as, under the basic assumption of equal probability of punching any one number, there is some chance that if one of X punching instructions causes a hole to be punched, another one of the X instructions might be directed to the same hole. The formula given takes this possibility into account.

Applying this formula to our three fields we find that  $G/H$ , i.e. the fraction of the field actually punched, is as follows:

Field Number	G/H
1	0.47
2	0.265
3	0.47

If we insert one needle each at random into the three fields, the fraction of the file that would be predicted to drop out would be  $(0.47)(0.265)(0.47) = 0.059$  or 5.9%. This percentage is an approximate measure of the number of random cards that would be dropped out. For small files, say up to 1000 cards, this dropping fraction might correspond to a tolerable number of unwanted cards. For larger files, further discrimination to eliminate unwanted cards might have to be provided and a possible means for accomplishing this will be presented subsequently. Before doing that, let us consider the dropping fraction for a search directed to the first two fields. This would be  $(0.47)(0.265) = 0.125$  or 12.5%. Interpretation of the significance of this percentage must take into account that it is an average value and that it includes both wanted cards--i.e. those



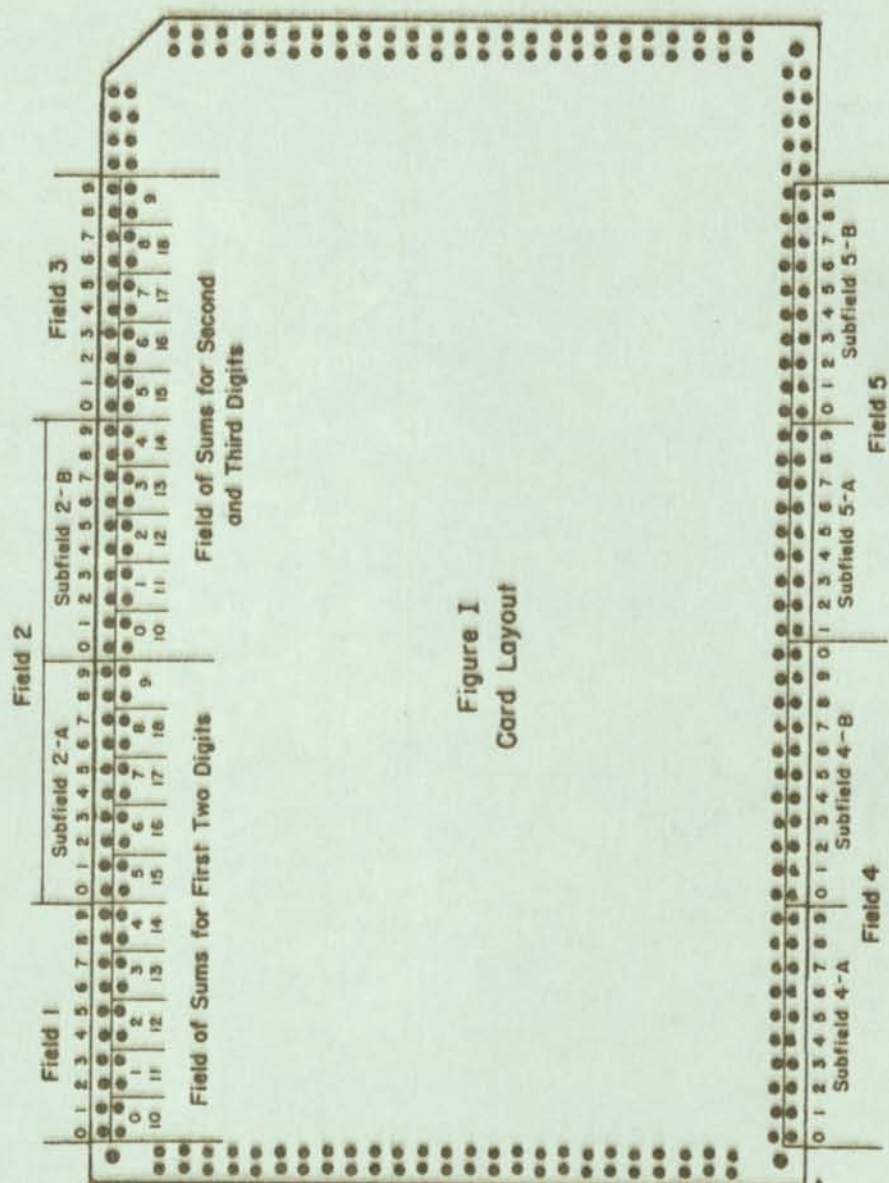
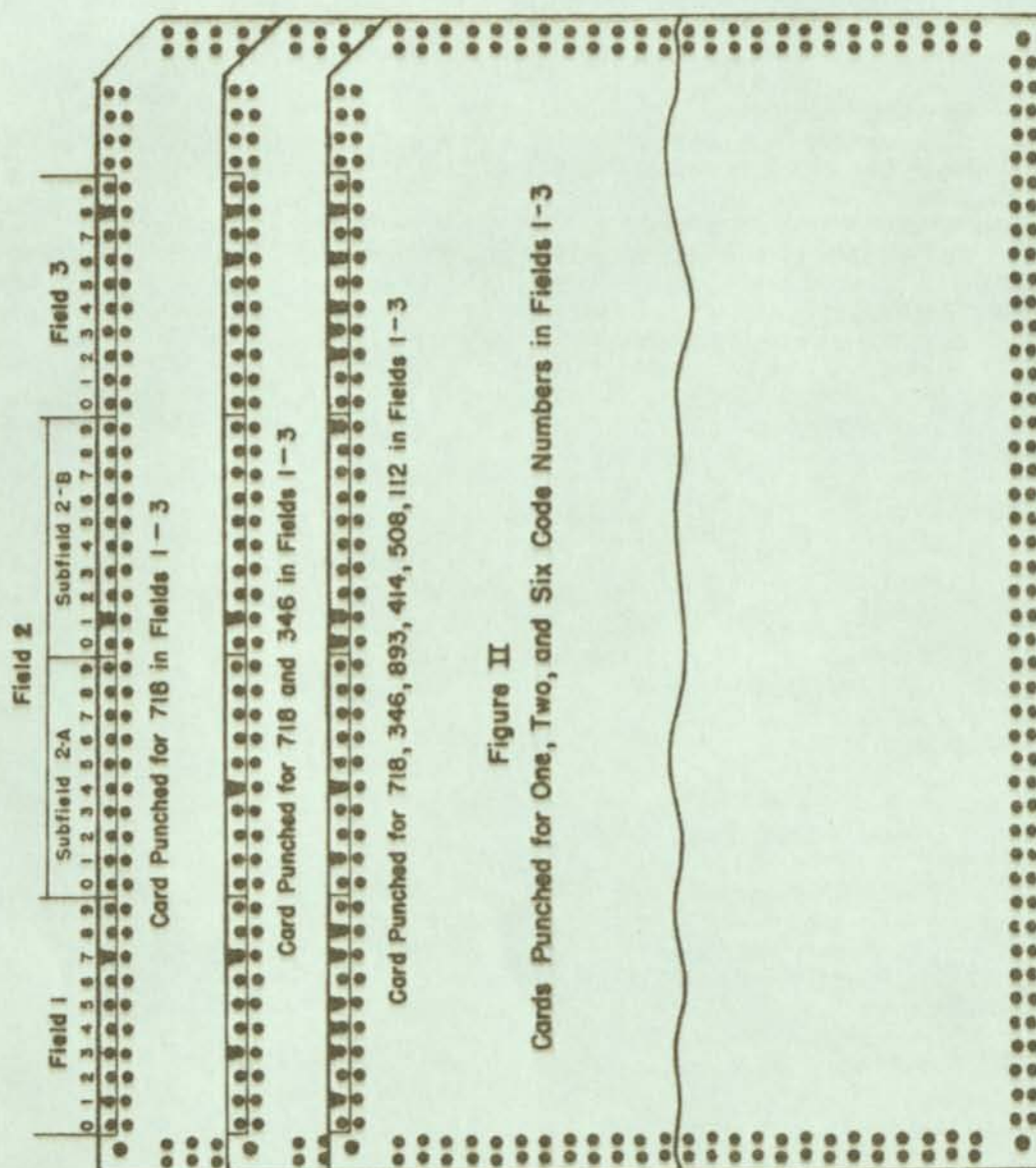


Figure I  
Card Layout







punched for the number in question--and also unwanted cards, i.e. those selected because of "phantom" combinations. For a range of 99 subjects with 6 subjects punched per card, the fraction of cards punched for any one subject would be 6/99 or very nearly 6%. In other words, our calculations would suggest that for every wanted card we might expect one additional unwanted one. Such a ratio might prove intolerably large and here again further discrimination might prove helpful.

In considering how to provide additional discrimination, we first observe that the tendency for individual digits of different code numbers to act independently is the reason why "phantom" numbers are generated. This undesirable tendency can be counteracted by providing a closer link between successive digits of each code number. To this end the sum of the first two digits of each code number is punched in the card and, in another field, the sum of the second and third digits. The mode of punching these sums requires further explanation.

In the first place, these sums are punched in the inner row of holes along the top of the card. These inner holes are punched by clipping out the cardboard between a given inner hole and its companion outer hole. The inner holes are not punched to the edge of the card so that punching them will not interfere with the punching of the outer holes. When a searching operation is directed to some one inner hole, cards in which that hole has been punched do not as a rule, fall free of the needle but drop about one-fourth of an inch. The next step in completing separation of the cards is to align them, then withdraw the sorting needle and reinsert it in the upper left-hand hole--a hole that is never punched. On ruffling the cards, those that previously dropped about one-fourth of an inch will now fall free of the needle.

In order that sums of successive digits may provide maximum discrimination it is necessary to take into account the frequency with which certain sums will be generated when two digits are taken at random. The table of sums shows, for example, that 18 results only from adding 9 with 9, while 8 may result from adding up eight different combinations (0 + 8; 2 + 6; 3 + 5; 6 + 2; 4 + 4, etc.).

	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	10
2	2	3	4	5	6	7	8	9	10	11
3	3	4	5	6	7	8	9	10	11	12
4	4	5	6	7	8	9	10	11	12	13
5	5	6	7	8	9	10	11	12	13	14
6	6	7	8	9	10	11	12	13	14	15
7	7	8	9	10	11	12	13	14	15	16
8	8	9	10	11	12	13	14	15	16	17
9	9	10	11	12	13	14	15	16	17	18

If the sums are written out in ascending numerical order, the corresponding frequency of occurrence increases from 1 to 10 and then decreases to 1 again.

<u>Sums</u>	0	1	2	3	4	5	6	7	8	9
	10	11	12	13	14	15	16	17	18	
<u>Occurrence</u>	1	2	3	4	5	6	7	8	9	10
	9	8	7	6	5	4	3	2	1	

By pairing the sums off--and treating the sum 9 independently--the joint probability of occurrence associated with each of the pairs can be made equal.

<u>Paired Sum</u>	0	1	2	3	4	5	6	7	8	9
	10	11	12	13	14	15	16	17	18	
<u>Occurrence</u>	10	10	10	10	10	10	10	10	10	10

The paired sums and the sum 9 generated by adding the first two digits of a code number are coded in a field of twenty inner holes. Two holes are used for each pair of sums (and 9). Which of the two holes is punched is made dependent on the numerical value of the second digit of the code number. If this digit is 0-4, then the left hole of the pair is punched, while if the second digit is 5-9, the right hole of the pair is punched. This method of punching insures that each hole will be punched with equal frequency. Consequently, with six code numbers punched in the field, application of formula I indicates that the fraction of the field punched will be 0.265.

The paired sums generated by adding the second and third digits are also coded in a second field of twenty inner holes. As before, two holes are used for each pair of sums and the left hole of the pair is punched if the third digit of the code number is 0-4 while the right hole



of the pair is punched if the third digit is 5-9. As before, the fraction of the field punched will be 0.265.

If a search is directed at random to any one hole in each of the first two fields and to the field in which the sums of the first two digits are punched, then the percentage of cards that would be expected to be selected would be, on an average,  $(0.47)(0.265)(0.265) = 0.034$  or 3.4%. While for a search directed to the first three fields and the two fields for sums, the percentage of cards that would be expected to drop would be, on an average,  $(0.47)(0.265)(0.265)(0.47)(0.265) = 0.004$  or 0.4%.

As searching operations are extended to the fourth and fifth digits, the percentage of cards that would respond, on an average, would be, respectively, 0.1% and 0.03%.

It may have occurred to the thoughtful reader that a random search directed to the first two fields and the field of sums for the first two digits is predicted to result in selecting on an average 3.4% of the cards in the file, while the percentage of cards coded for the first two digits of any one code number will be 6/99 or very nearly 6%. This apparent discrepancy is caused, as a little reflection makes clear, by the fact that the coding system is set up so that certain random searches will result in more cards being selected than in others. Thus certain random searches will not correspond to the punching for any one code number. As an example, consider a search directed in the first field to 3, in the second field to 9 preceded by 5-9, and in the appropriate field of sums to the sum pair 6-16 associated with 0-4 as a second digit. Punching no one code number in a card will cause it to be selected by such a search. Some cards may nevertheless respond; for example, a card punched for 32 and 79 (and any other four numbers of two digits each, for example 25, 56, 84, 64.) If instead our random search were redirected in the first field to 8, in the second field to 2 preceded by 5-9, and in the appropriate field of sums to the sum pair 0-10 associated with 0-4 as the second digit, then all cards punched for 82 will respond and so also will the previously mentioned example card punched for 32, 69, 25, 56, 84, and 64. This second type of search, which corresponds to some actual code number, would be the only one ever conducted

when using a card file. Hence, a further examination of the possibility of obtaining unwanted cards is in order.

Let us imagine, first of all, a search directed to some digit in Field 1. This search will cause 47% of the cards to drop. These will consist of cards punched for the number being searched (6%) and unwanted cards (41%). To effect further separation, the cards previously selected are submitted to a sorting operation based on the second digit punched in Field 2. In estimating the effectiveness of this searching operation for eliminating unwanted cards, we must recall to mind, first of all, that we are assuming that each card is punched for six code numbers--the one for which we are searching and five others. Each of these five will require the punching of five numbers randomly distributed over the twenty holes of Field 2. On an average, therefore, 5/20 or one-fourth of our unwanted cards will be punched at any one hole in Field 2. Hence, of the 41% of unwanted cards, three-fourths or 30.8% of the original file will be eliminated. After this sorting operation the isolated cards will consist of all the unwanted cards (6% of the original file) and nearly twice as many unwanted cards (10.2% of the original file). Three-fourths of these remaining unwanted cards will be eliminated by a sorting operation directed to that hole in the appropriate field of sums which corresponds to the two digit code number for which we are searching. Thus after these four sorting operations, the selected cards will consist, on an average, of all the wanted cards (6% of the original file) and less than half as many unwanted cards (2.6% of the original file).

In estimating the probability of obtaining unwanted cards, the order of succession of the searching operations was represented as being Field 1, Field 2 and the appropriate field of sums. In order to reduce the number of cards to be handled as rapidly as possible, it is advisable, in the first searching operation, to use two needles, one each in Field 1 and in Field 2. Mastering the knack of a two-needle search may require a little practice, but such a search in one operation will isolate, on an average, from the file 16.2% of the cards, which would then require resorting in the field of sums.

Before concluding discussion of searches directed to two-digit code numbers, it should be noted that a search directed to one such



number would actually occur in practice only when there is interest in selecting out all cards dealing with one--and only one--broad subject. It is much more likely--the nature of information being what it is (1)--that a search will be directed simultaneously to several broad subjects. If only two broad subjects are searched simultaneously, the proportion of unwanted cards becomes smaller. If Fields 1 and 2 and the appropriate field of sums are searched, we find on applying the same reasoning as before, that more than four wanted cards will be selected for each unwanted card. The percentage of the file appearing as unwanted cards calculates out as being, on an average, 0.00085%. This means that, for all practical purposes, an unwanted card would appear rarely, if ever, in a search directed to two broad subjects, each punched as a two-digit code number. If our simultaneous search to two broad subjects makes use only of Fields 1 and 2, then the percentage of unwanted cards would be 1.4% of the total file. It might not be necessary in many cases to conduct a second supplemental search directed to the fields of sums.

Similar calculations with respect to searches directed to Fields 1, 2, and 3, and the two fields of sums shows that when searching is directed to the first three digits of a code number the percentage of the file appearing as unwanted cards would, on an average, be nearly 0.4%. This small percentage corresponds to a number of unwanted cards, almost certainly too small to cause serious difficulty when working with a file of moderate size. For a similar search directed to two subjects, each represented by three-digit code numbers, the percentage of the file appearing as unwanted cards would be less than 0.01%. For such a search conducted without benefit of the two fields of sums, less than 0.4% of the file would be expected to appear as extra cards.

#### Concluding Remarks.

The punching system described requires less than half the holes on a standard "E-Z Sort" card. (7) This means, of course, that there are plenty of holes remaining for further coding if that should prove necessary.

No punching system can be better than thinking devoted to the question of how information is to be analyzed and encoded. Much careful thought must be devoted to developing an effective system of analysis if the coding system described is to function with maximum effectiveness.

#### REFERENCES

- (1) Perry, J.W. "Information Analysis for Machine Searching". *American Documentation* **1**, 133-139 (1950).  
Wise, Carl S. and Perry, J.W. "Multiple Coding for the Rapid Selector". *American Documentation* **1**, 76-83 (1950).
- (2) *Chem. & Eng. News*, **29**, 4214 (1951)  
Perry, J.W. and Casey, R.S. "Literature, Mechanized Searching", in "Encyclopedia of Chemical Technology", Vol. 8 (in press). Interscience Encyclopedia, Inc., New York, 1952.  
Perry, J.W. "The ACS Committee on Scientific Aids to Literature Searching", *Chem. & Eng. News*. (in press).
- (3) Casey, Robert S. and Perry, J.W. "Elementary Manipulations of Hand-Sorted Punched Cards". Chapt. 2 in "Punched Cards. Their Application to Science and Industry", edited by Robert S. Casey and J.W. Perry. Reinhold Publishing Co., New York, 1951.
- (4) Breger, Irving A. "Application of Simple Coding Procedures to a Specific Problem", Chapt. 3 in "Punched Cards. Their Applications to Science and Industry" cited above.
- (5) Bailey, M.F., Lanham, B.H. and Leibowitz, J. "Mechanized Searching in the U.S. Patent Office". Abstracts of Papers, 120th Meeting, American Chemical Society, Sept., 1951, p. 2F.  
Williams, T.M., Reid, A.M. and Perry, J.W. Unpublished experiments.
- (6) Wise, Carl S. "Mathematical Analysis of Coding Systems". Chapt. 20 in "Punched Cards. Their Applications to Science and Industry" cited above.
- (7) E-Z Sort Systems, Ltd., San Francisco, Calif.



## AN OPTIMAL PUNCH CARD CODE FOR GENERAL FILES\*

PAUL T. GILBERT, JR.\*\*

### INTRODUCTION

In a single-field superimposed coding system, each subject to be recorded on a card is associated with a pattern of signals or marks (punches) selected from a field of possible locations for such marks. The card has only one such field and there are no restrictions on the location, within the field, of the marks constituting the pattern. When the cards are sorted, during a given search, they are tested for the presence of the pattern of marks corresponding to the subject sought, and all cards displaying that pattern will be selected.

But, because there is no restriction upon the distribution or arrangement of the marks, the several patterns corresponding to the several subjects on each card generally overlap and may partially or, by chance, even completely coincide. Moreover, the pattern for any one subject may completely coincide with a group of marks present among those corresponding to several other subjects on the card. Because of this, when the cards are sorted for a particular subject, cards may be selected which do not relate to that subject. Such cards are called unwanted, extra, or "false drops," and the ratio of their number to that of the total file of cards (or the total file of unwanted cards) is called the dropping fraction,  $F$ , of extra cards.

The object of code design is to keep  $F$  minimal while providing the greatest freedom in the coding of information on the cards and the greatest speed, convenience, compactness, and simplicity — i.e., the greatest *economy* — compatible with an acceptably small  $F$ . As these desiderata conflict, the problem is one of selecting the best compromise. Fortunately, this compromise may be fairly well defined.

Zatocoding: The advantages of single-field superimposed coding have been discussed by Mooers<sup>1,2,3</sup>, who employs it in his system of "Zatocoding." Wise<sup>4</sup> has examined the mathematical basis of such coding, and records experimental observations of dropping fractions obtained with a single-field superimposed system. The characteristic feature of Zatocoding is the use of randomly selected numbers for translating any subject into a set of marks. As an example of one type of Zatocoding, a card may have 40 punch positions, constituting the field. For any subject, four numbers are selected at random (as by drawing from a hat) from among the numbers 1 to 40, and these four numbers are the code for the subject. For example, the subject "rare earth" may be assigned the code 22-24-34-39, and whenever "rare earth" is encoded, the positions 22, 24, 34, and 39 are punched. In seeking information on rare earths from the card file, one sorts it for the punch positions 22, 24, 34, 39, simultaneously or in any order, by any suitable mechanical means, such as the shaking box devised by Mooers. The cards thereby selected will include all those mentioning rare earths, in addition, usually, to several cards which happen to be punched in these four positions but are not concerned with rare earths.

A record of all such codes is kept in a coding dictionary, to which reference is made whenever one wishes to prepare a card for the file, or to sort the file for a desired subject. Subjects or terms thus coded are called descriptors and may consist of a single word, a phrase, a number, or any other symbol having a specific meaning which is to be indexed. Mooers, in describing his system, emphasizes the importance of randomness in the selection of descriptor codes. Departures from

\*This paper represents work done for Beckman Instruments in the spring of 1952, and was originally presented before the American Chemical Society in Los Angeles in 1953.

\*\*Beckman Instruments Inc., Fullerton, California.



randomness result in loss of efficiency, i.e., other things being equal, an increase of  $F$ .

**Orthographic Coding:** In considering multiple-field versus single-field superimposed coding systems, Wise shows that the latter yields the smallest  $F$  for a given total number of punch positions, or holes, and a given number of subjects per card. However, its advantage over multiple-field systems in this respect is not very great and Wise favors multiple-field coding because it introduces the possibility of ordering the cards to some extent. But the attainment of maximal efficiency requires the use of a random code, which precludes ordering with respect to any aspect of the subject unless restrictions are to be placed upon the manner of coding the subjects. It is this complete freedom from restriction—freedom from the need for recognizing categories—that makes Zatocoding particularly attractive. For a miscellaneous file I therefore agree with Mooers in favoring a random single-field system for providing the greatest flexibility as well as the greatest efficiency. If the file is special and devoted to a relatively few subjects, the use of more than one field is indicated. In such cases an arrangement comprising a few small direct-coded fields, coupled with a larger superimposed-coded field, may be most useful.

The subject of this paper represents a modification of the Zatocoding principle, wherein, instead of random numbers, certain pairs of letters (or other elements) taken from the descriptor itself, automatically provide the code. In using this method, one merely chooses from the spelling of the descriptor, according to an easily memorized pattern, those letter pairs designated for the code, and punches them into the edges of the card. The field of punch spaces, or holes, is correspondingly marked with groups of letter pairs, so that each pair belongs to a particular space. This system might therefore properly be called an orthographic single-field superimposed code.

The rationale and the mathematical arguments underlying the design of a successful orthographic code are rather involved and will, therefore, be given following the presentation of the practical steps for the design of such a code.

#### A RECOMMENDED ORTHOGRAPHIC CODE

**Descriptor Code:** The steps in designing an

orthographic code are given below. First, the descriptor coding system must be specified, which designates the letter pairs to be chosen for punching. This is shown in Table I. As an example, according to the table, the descriptor  $pH$ , having two letters, is punched in the spaces on the card corresponding to  $Ph$ ,  $Hp$  and  $Pp$  (i.e., the three letter pairs formed, respectively, of the first and second, the second and first, and the first letter doubled). In this discussion, the first letter only of a letter pair will be capitalized. "Coulometry" with ten letters is punched in the spaces for  $Ou$ ,  $Om$ ,  $Tr$ ; "Ion exchange resin" is punched  $On$ ,  $Xc$ ,  $An$ ,  $Er$ ,  $Si$ . In general, sufficiently long descriptors are punched into the spaces corresponding to the pairs obtained from the second and third letters of each consecutive group of three letters. The rule is further imposed that if, among the first three punches for a descriptor, any punch by chance coincides with another, the next higher space is punched (or the next beyond that if the first is already occupied by a punch for this same descriptor). Thus all descriptors will have at least three distinct punches. Of course, if several descriptors are to be punched on a card, punches from different descriptors may, and frequently do, occupy the same space.

TABLE I: RECOMMENDED DESCRIPTOR CODE

No. of letters	Letter pairs to be punched
1	1,1 and the next two higher spaces
2	1,2 2,1 1,1
3	1,2 2,3 3,1
4	1,2 2,3 3,4
5	1,2 3,4 4,5
6	1,2 3,4 5,6
7	2,3 4,5 6,7
8	2,3 5,6 7,8
9, 10, 11	2,3 5,6 8,9
12, 13, 14	2,3 5,6 8,9 11,12
15, 16, 17	2,3 5,6 8,9 11,12 14,15

**Field Specification:** Second, the spaces of the field must be assigned to letter pairs, or groups of them. For a medium-sized field the best possible such assignment demands a field of 55 spaces, as is shown in Table II. To illustrate the interpretation of the 55 symbols of this table,  $A_{ug}$  designates the space assigned to the letter pairs  $Au$ ,  $Av$ ,  $Aw$ ,  $Ax$ ,  $Ay$ ,  $Az$ ,  $Aa$ ,  $Ab$ ,  $Ac$ ,  $Ad$ ,  $Ae$ ,  $Af$  and  $Ag$ .  $B$  is the space for all letter pairs beginning with  $B$ ;  $HafJK$  provides for  $Ha$ ,  $Hb$ ,  $Hc$ ,  $Hd$ ,  $He$ ,  $Hf$ , and all pairs beginning with  $J$  or  $K$ . The  $(26)^2$  or 676 possible English letter pairs have thus been distributed among the 55 spaces of the field.

To encode the subject  $pH$ , one punches the



spaces designated *Pah*, *Hgz*, *PizQ*; to encode *Spectrophotometry* one punches *Pah*, *Tmr*, *PizQ*, *Est*. In this latter example note the application of the rule regarding overlapping: because *Ph* would fall in the space *Pah* already occupied by *Pe* (the first punch for the descriptor), *Ph* is punched in the next higher space, viz., *PizQ*; further, *To* falls in the same space as *Tr*, already punched, but as there are already three distinct punches the overlap is allowed to occur and *To* yields no new punch. Lastly, *Et* falls in the space *Est*. If both *pH* and *Spectrophotometry* are to be punched on the same card, then after *pH* has been punched, *Spectrophotometry* yields only the additional punches *Tmr* and *Est*, because *Pah* and *PizQ* have already been punched for *pH*.

TABLE II: RECOMMENDED FIELD SPECIFICATION

Aug	Doe	F	Isv	Nhr	Rbe	Til
Ahm	Dft	G	Lae	Nsz	Rfn	Tmr
Anr	Ecf	HafJK	Lfm	Oug	Rol	Tsd
Ast	Egm	Hgz	Lnz	Ohm	Rua	Ulp
B	Enq	Iwc	Maf	On	Shh	Uqk
Cah	Er	Idm	Mgz	Oot	Sio	VWX
Cio	Est	In	Nad	Pah	Spt	Y
Cpz	EubZ	Ior	Neg	PizQ	Teh	

Now, if the file is to be sorted for the subject *pH*, all the cards are tested for punches in the positions *Pah*, *PizQ* and *Hgz*. All cards having the descriptor *pH* will then be found. Cards having *Spectrophotometry* but not *pH* will not drop out unless they, by chance, have other descriptors (e.g., *Photometry*) which yield a punch in the position *Hgz*. Also, of course, still other cards having none of the descriptors mentioned, but having punches in the three spaces tested, will drop out and must be discarded by hand selection, or, by further restricting the subject *pH*, to retest those cards already sorted. For example, if the searcher is interested only in *pH* meters, he may, at the outset, test the whole file in the positions *Pah*, *Hgz*, *PizQ*, *Maf*, *Teh*, *Er*, which are the codes for *pH* and *Meter* separately. Much better selectivity will then be obtained, that is, there will be far fewer extra (unwanted) cards than if the file were sorted only for *pH*. If he suspects that the makers of the file might have coded *pH* meters as a single descriptor *pH-meter*, he will test the spaces *Hgz*, *Est*, *Er*. He may not test simultaneously for *Pah*, *Hgz*, *PizQ*, *Maf*, *Teh*, *Er*, *Est*, because this will yield only cards containing *pH*, *Meter*, and *pH-meter* (plus possibly a very few extra cards). If he wants all

cards containing *pH* and *Meter* or *pH-meter* the two sortings must be conducted separately. This emphasizes the principle that, within limits, a compound subject is better coded by parts, because if coded as a single descriptor its parts cannot be sorted separately. Also the code of the single descriptor gives poorer selectivity, as it is usually shorter, than do the combined codes of the several partial descriptors.

### RATIONALE OF ORTHOGRAPHIC CODING

*Frequency Distributions of Letters:* A system of multiple-field *word coding* has been used by Wise<sup>4,5</sup> which anticipates some of the features of orthographic coding. But, as he shows, there is appreciable loss of efficiency due to the non-random or nonuniform distribution of punches in his word coding. Utilization of his fields is only about 73% efficient, and this causes the dropping fractions of extra cards to be several times as great as those theoretically attainable. Recognizing the nonuniform density of frequency distribution characterizing a system in which all the letters of the alphabet are given equal weight, Cox, Casey and Bailey<sup>6</sup> studied, with a view to improving coding systems, the relative frequencies of the first, second, and third letters of proper names and separated the alphabet of author indexes into 50 equal parts.

The great bulk of all recorded information is expressed, or is generally sought, in terms of words the elements of which are letters. If, in the present system, the letters of the descriptors (digits of numerical descriptors and other symbols, neither letters nor digits, are treated similarly but more easily) are to be used for coding, one is confronted with 26 of them (in English), with widely varying relative frequencies. To assign at least one space of the field to the least frequent letters, and several spaces, in proportion to their frequencies, to the others, requires an unduly large number of spaces, and leaves unanswered the question of how to choose one of the several spaces of the more frequent letters. This problem is resolved, if the field is divided among the single letters in such a way that each is represented proportionally to its frequency. Then, in order to preserve uniform distribution of punches, the same principle must be applied to the designation of these several subspaces for the commoner letters. To keep the most desirable



feature of the system, viz., simplicity of automatic descriptor coding, some other easily recognized feature of the word must be used in deciding which of these subspaces to select when one of the commoner letters is to be punched. The simplest scheme appears to be the use of another letter of the descriptor. This suggests the using of pairs of letters as the primary entities for coding. To get the same frequency distribution of the second letter of the pair, as obtains for randomly-chosen first letters, it is necessary to choose a second letter so far removed from the first, within the descriptor, as to be essentially uninfluenced by the first letter. This is not possible for short descriptors. The first letter following a given random letter does not exhibit the same frequency distribution, and the second letter following a given letter has a frequency distribution still decidedly influenced by the reference letter. Even the third letter would not be essentially uninfluenced. Moreover, in choosing letter-pairs thus widely separated in the word, the coding becomes more difficult, and requires greater care and increases the possibility of error. On the contrary, the pairs of letters most easily recognized, remembered, and recorded are pairs of consecutive letters.

Pairs of consecutive letters, however, exhibit a special frequency distribution, and if the 676 pairs are to be equitably assigned to a number of spaces constituting a coding field, their frequency distribution must be measured. In obtaining such a measurement, descriptors must be chosen at random, and it is important to select a proper source of such descriptors. As the coding system is intended for completely miscellaneous and general files and indexes of a more or less technical character, one may assume that *Chemical Abstracts*, which covers many fields allied to chemistry, is rich in the special terminology of biology, mathematics, etc., and is also completely cosmopolitan, might be a suitable source of descriptors for letter-pair counting. Selecting words, phrases, and names likely to be used in file sorting, in a random manner from various parts of *Chemical Abstracts*, I counted 10,000 letter pairs, taking each letter of each descriptor as the initial letter of a pair, and including also the pair consisting of the last letter followed by the first. While revealing the frequencies of the 676 pairs, this count provided, incidentally, the frequencies of the 26 single letters. These are given in Table III.

Table III

## Frequencies of Single Letters in Technical Descriptors

A	7.79%	J	0.15%	S	5.09%
B	1.51	K	0.75	T	7.82
C	4.98	L	5.85	U	3.26
D	3.06	M	3.84	V	0.77
E	10.67	N	7.11	W	0.52
F	1.38	O	7.93	X	0.49
G	2.14	P	3.26	Y	1.68
H	2.69	Q	0.18	Z	0.44
I	9.35	R	7.42		

The frequencies depend upon the fraction of proper names admitted to the count. For example, *L* and *O* become more frequent when proper names were omitted, and *S* less so, while *I* and *C* were little affected. I had no satisfactory criterion for determining the fraction of proper names but I used what I thought to be a judicious proportion. Some files will be devoted, largely, to personal names and others mainly to subjects. These distributions are valid, moreover, only for files in English. The distribution of single letters and pairs will vary with the language and should be separately determined for any language in which orthographic coding is to be used. No doubt the differences in this respect between English and the several Teutonic and Romance languages would not be great; those languages, having a less heterogeneous etymology than English, would probably exhibit an even less uniform frequency distribution of letter pairs. A moderate admixture of descriptors in languages other than English should not cause noticeable loss of efficiency.

**Important Parameters:** To assure sufficient selectivity, i.e., sufficiently small dropping fractions for files which may become quite large and which may, not infrequently, have to be sorted for single descriptors, a minimum number of field spaces, *H*, must be imposed. For the simpler systems for general filing recommended by the Zator Co., *H* = 40. This is adequate for files of approximately 10,000 cards provided not over about  $28/N$  descriptors (*N* being the number of punches per descriptor) are to be punched on each card. A careful consideration of the requirements of adequate cross-indexing for a highly miscellaneous technical file convinced me that  $28/N$  is unduly restrictive (as will be shown later, *N* should not average less than 3) but if as many as 12 or 13 descriptors are permitted for the more crowded cards one will seldom need to make more than one card for the indexing of any



single item. The word *item* should be understood as referring to a technical article or abstract, a page or section of a research notebook, a piece of correspondence, a section of a book, a manufacturer's bulletin, a memorandum, a set of data, etc.; clearly, a whole book or a long article would require several or perhaps many cards for complete indexing. However, a single item not capable of subdivision without loss of indexability, may require as many as a dozen descriptors. On the other hand many items will require few descriptors, possibly only one. Dilution of the file by cards with few descriptors allows the use on some cards of a number of descriptors exceeding the optimum specified by the theory of single-field superimposed coding.

Therefore, I considered the optimum for  $H$  to be between 50 and 60 for files of a few thousand cards which may need to be sorted occasionally for single descriptors. If at least two descriptors are to be used for each sorting, such a range of  $H$  would be useful for files of 100,000 cards; but many situations will arise in which only one descriptor can be used in the search, and in such cases the average dropping fraction in an optimal coding system with  $H$  between 50 and 60 would amount to thousands of cards of the 100,000 total in the file. To accommodate single-descriptor searches with larger files, progressively larger values of  $H$  are required: for 100,000 cards  $H$  would have to be about 200 in order to keep the extra cards down to about 100 for a single descriptor. A field of 100 spaces would be useful, on the same criterion, for a file of about 20,000 cards. It is necessary to weigh the extra cost and labor of coding and sorting involved with larger values of  $H$  against the greater time needed for hand-sorting extra cards when information represented by a single descriptor is to be retrieved. It is probably worth the acceptance of much hand-sorting for the occasional single-descriptor search when  $H$  is 55, provided the file does not greatly exceed 10,000 cards.

**Field Partition:** The choice of 55 for  $H$ , was dictated by the peculiarities of the distribution shown in Table III. For ease in locating the spaces to be punched, the letter pairs should be arranged in almost alphabetical order along the edge of the card. To avoid confusion, the borders of adjacent first letters should not overlap. For example, because the frequency of  $F$  is somewhat less than  $1/H$  while that of  $G$

appreciably exceeds  $1/H$ , it might be assumed that, in the interest of attaining a uniform distribution, the one space should include all pairs beginning with  $F$  plus a few beginning with  $G$ , while the next space should contain the balance for  $G$ ; but this practice greatly complicates the field designation. By use of cardboard strips cut to lengths proportional to the frequencies of the 26 letters (Table III), and placed on a converging grid, I found that for  $H$  in the range 50 to 60, more or less, by far the best fit between letter-frequencies and integral numbers of field spaces could be obtained with  $H = 55$ . The problem resembles that of selecting 16 for the atomic weight of oxygen in order to make the greatest number of elements have the most nearly integral atomic weights. This choice of 55 for  $H$ , nevertheless, demands certain groupings of letters:  $V$ ,  $W$  and  $X$  were put in one space;  $Q$  was placed with the second half of  $P$ ;  $J$  and  $K$  with the first half of  $H$ ; and  $Z$  with the last sixth of  $E$ . Thus  $Z$  is the only badly misplaced letter; but it occurs infrequently and this one exception is easily remembered. Similarly,  $J$  and  $K$ , in the wrong position with respect to  $I$ , seldom occur. A great deal of juggling and successive eliminations of possible systems convinced me that this was the best partition of the 55 spaces for first letters (cf. Table II), combining departures from uniform distribution of punches (most nearly equal probabilities of being punched for all the spaces in the field) with maximal convenience of finding spaces.

The secondary division demanded equal care. For a given first letter, the 26 letter pairs usually show an extremely uneven distribution. In particular,  $Er$ ,  $In$ , and  $On$  are so abundant (Table IV) that with  $H = 55$  they slightly overfill a single space. If  $H$  were much greater than 55 these three letter pairs would need to be subdivided. Such a division would be very awkward for  $H$  up to about 80, as  $Er$ ,  $In$ , and  $On$  could hardly occupy an integral number (1 or 2) of spaces without badly distorting the distribution. For  $H$  between 80 and about 120, these pairs could be equitably assigned to two spaces each, in which case the frequency distribution of the letters next following  $Er$ ,  $In$ , and  $On$ , respectively, would have to be measured, and the two spaces for each split according to the third letter of the triplet. Returning to  $H = 55$ , the other, less frequent letter pairs are such that several can be grouped in each space. Use would be more convenient if, for first letters having more than



Table IV

## Frequencies of Pairs of Consecutive Letters in Technical Descriptors

	a	b	c	d	e	f	g	h	i	j	k	l	m
A	6	14	53	31	4	4	17	0	19	0	5	108	38
B	19	1	0	5	26	0	0	1	17	0	0	14	0
C	75	2	10	3	48	1	1	49	30	0	21	8	5
D	18	6	6	1	84	2	4	9	71	1	0	9	2
E	50	12	93	37	27	34	13	8	14	1	4	86	46
F	6	0	1	0	11	11	0	0	27	0	1	31	1
G	27	10	4	5	44	4	5	10	4	1	2	22	8
H	33	3	1	0	51	1	1	2	30	1	0	11	5
I	21	15	114	63	25	16	14	1	1	1	0	56	23
J	8	0	0	0	1	0	0	0	1	0	0	0	0
K	6	2	2	0	14	1	1	0	7	3	0	0	2
L	78	4	19	13	88	17	5	6	103	0	5	45	16
M	57	7	9	4	115	4	2	2	63	0	0	2	14
N	70	4	34	63	91	11	91	10	57	5	6	6	4
O	10	10	25	31	9	7	24	6	11	1	3	84	71
P	35	0	1	2	66	0	0	72	10	0	1	32	9
Q	0	0	1	0	0	0	0	0	0	0	0	0	0
R	118	24	39	14	95	11	8	2	104	0	3	2	41
S	20	2	26	1	38	8	8	22	79	1	9	7	8
T	62	3	5	6	148	0	5	53	208	1	0	2	14
U	4	16	23	4	6	1	3	0	10	0	1	37	78
V	5	0	0	0	36	0	0	0	33	0	0	0	0
W	18	1	2	1	7	0	0	1	4	0	0	1	2
X	4	2	7	1	0	0	0	0	14	0	0	0	6
Y	9	9	17	10	13	2	6	4	6	0	0	27	5
Z	4	0	0	0	12	0	0	1	8	0	0	4	0

Table IV - continued

	n	o	p	q	r	s	t	u	v	w	x	y	z
A	94	2	29	0	81	57	158	6	11	4	11	13	2
B	0	33	0	0	19	8	0	10	0	0	0	0	0
C	3	97	11	0	15	9	92	12	0	0	0	4	6
D	3	8	3	0	20	12	4	28	0	2	0	8	0
E	122	11	40	4	202	80	127	9	10	5	15	3	1
F	0	13	0	0	17	0	2	8	0	0	0	0	0
G	6	1	5	0	19	4	9	1	1	2	0	0	0
H	5	55	9	0	7	7	5	5	1	4	0	24	0
I	204	137	16	6	11	44	81	35	29	0	3	0	14
J	0	3	0	0	0	0	0	2	0	0	0	0	0
K	0	1	1	0	4	5	0	1	0	2	0	1	0
L	5	70	11	0	1	9	21	35	7	3	0	29	0
M	1	39	34	0	0	23	3	12	0	0	0	1	1
N	5	27	13	1	19	45	91	19	2	0	1	7	6
O	218	13	34	0	89	27	55	31	5	20	18	5	0
P	0	44	21	0	29	7	9	5	0	0	0	8	0
Q	0	0	0	0	0	0	0	10	0	0	0	0	0
R	10	108	29	0	19	26	16	31	4	4	0	37	1
S	1	49	42	1	0	37	80	32	0	5	0	4	2
T	1	70	8	0	126	11	18	22	1	3	0	30	4
U	27	10	9	0	48	22	18	0	2	0	1	0	0
V	0	3	0	0	1	1	0	3	0	0	0	0	0
W	0	8	0	0	2	0	3	0	1	0	0	4	0
X	0	0	5	0	2	2	4	0	0	1	1	0	2
Y	1	2	19	0	17	26	4	1	3	1	1	0	0
Z	0	2	0	0	3	1	0	0	0	0	0	2	0



one space, the spaces could be so divided that the first of the second letters in the first space were *a*. E.g., for first letter *A*, the subdivision *Aal, Amq, Ars, Atz* would be more convenient to use than the adopted division *Aug, Ahm, Anr, Ast*, which requires that one think in terms of a circular alphabet with *a* following *z*. But the *Aal* division, although the best of its kind, involves decidedly larger departures from uniform distribution than the *Aug* division. In some cases it was, apparently, only by sheer good fortune that even a consecutive grouping of the second letters could be made, regardless of whether one of these groups began with *a*.

**Descriptor Code:** To avoid pitfalls, the specification of the descriptor code (Table I) likewise required care in formulation. For words of a given number of letters, the frequency distribution of the first pair of letters will differ from that of the pair consisting of the second and third letters, or of subsequent pairs. Also, certain letters of words in many categories, such as the names of the metals or the names of measuring instruments, are more characteristic of the word, or perhaps more useful in distinguishing closely similar but distinctly different words, e.g., *phosphoric, phosphonic; phosphate, phosphite; photometer, photometry*, than other letters. But an unduly difficult code that requires precounting the letters of the descriptor and laying-out of the code as a separate step, may waste much time in coding and produce frequent errors. The special advantage of the orthographic over the numerical Zator system is that no coding dictionary is needed for either coding or sorting. Thus if the orthographic code is so simple that the translation from spelling the descriptor to punching can be carried out mentally without delaying the punching, the coding operation requires, in effect, no time at all. Consideration of categories of words from the viewpoint of differentiating them with the fewest punches, together with the mathematical theory of the relation between descriptor code and dropping fractions, and experience in using several alternate coding systems, led to the adoption of the code of Table I.

Many alternative systems of partitioning the field and choosing a descriptor code were considered and discarded for reasons of inefficiency or inconvenience. For example, deliberate omission of letters, such as rare letters or, perhaps, all vowels, resulted in inevitable loss

of selectivity among closely related words, loss of identity of certain shorter words, and needless cases of high dropping fractions that could be easily avoided in a properly designed system recognizing all the letters. The system finally adopted has proved so easy to use and has yielded such excellent efficiency that it appears entirely satisfactory from every viewpoint.

For those who may wish to redesign the field partition for larger values of *H*, Table IV gives the counts of the 676 letter pairs from a series effectively totalling 9951 pairs. It must be emphasized that in undertaking such a design, to violate the principle of uniform distribution is to waste spaces. Because convenience and economy depend so strongly upon efficiency in terms of dropping fraction, the wasting of spaces by under-filling should be avoided by devising the most equitable distribution of letter pairs—or triplets when necessary—compatible with speed of card punching.

### MATHEMATICAL THEORY

To obtain experimental proof of the efficiency of the coding system, it was necessary to predict the dropping fractions. The equations of Wise<sup>4</sup> and the close approximations of Mooers<sup>3</sup> are not satisfactory for such a prediction when the number of descriptors per card varies widely. According to Wise, if *N* is the number of punches per descriptor, *D* the number of descriptors per card, *H* the number of spaces in the field, and *G* the average number of spaces actually punched in the card (which will be less than *ND* because of coincidences), then

$$G/H = 1 - \left(\frac{H-N}{H}\right)^D \quad (1)$$

According to Mooers,

$$G/H = 1 - e^{-ND/H} \quad (2)$$

These expressions, for practical systems, generally differ by less than 1%. The ratio *G/H* is the average fraction of spaces punched, and is equal to the total dropping fraction for sorting with one needle, *F*<sub>1</sub>. If *S* is the number of needles used in sorting, i.e., the number of field spaces tested for punches, then according to Wise, the total dropping fraction for sorting with *S* needles is exactly



$$F_S = S^C G / S^C H \quad (3)$$

in which  $A^C B$  is the binomial coefficient or the number of combinations of  $B$  things taken  $A$  at a time, given by  $A^C B = B! / A!(B - A)!$  According to Mooers, approximately

$$F_S = (G/H)^S \quad (4)$$

For  $S = 1$ , equations (3) and (4) are the same, but because the values of  $G$  calculated in the two different ways differ slightly, the same difference appears in  $F_1$ . Equation (3) gives progressively smaller values for  $F_S$  as  $S$  increases, than does equation (4), because (4) does not allow for the fact that after the file is sorted for one position, all the cards selected are already punched in that position and the average number of remaining punches is less than the number that would be found in  $H - 1$  spaces of a pack of unsorted cards.

But in an actual file with  $D$  variable, the trend is in the opposite direction. The  $F_S$  values tend to rise above the values indicated by (4) as  $S$  increases, and so (4) comes closer to the actual situation than (3). This is because for higher values of  $S$  the extra cards are, to an increasing extent, those with larger  $D$ , and these cards are dominant in the smaller dropping fractions. In a code with  $N$  punches per descriptor, the exact average dropping fraction will be given by

$$F_S = \sum_{G=S} \sum_{D=1} ND^P G \cdot P_D \cdot S^C G / S^C H \quad (5)$$

in which  $ND^P G$  is the probability that a card with  $D$  descriptors will have exactly  $G$  spaces punched and  $P_D$  is the probability that a card will have exactly  $D$  descriptors. Since  $ND$  is often written  $X$  (the total number of intended punches or punching "instructions"),  $ND^P G$  can be written  $X^P G$ , the probability that for  $X$  intended punches there will be  $X - G$  coincidences. The value of  $X^P G$  depends implicitly on  $N$ .

**Number of Descriptors:** In this evaluation,  $P_D$  is the inexact element, varying from file to file. It was measured for the miscellaneous file used in the test, and the resulting slightly erratic distribution was smoothed to yield the  $P_D$  values given in Table V. It must be noted that  $P_D$  depends also upon the manner in which descriptors are to be subdivided: *pH* and *Meter* are two descriptors but *pH-Meter* is one.

Table V

Frequency Distribution of Numbers of Descriptors per Card

D	P <sub>D</sub>	D	P <sub>D</sub>
1	.002	8	.119
2	.017	9	.112
3	.043	10	.103
4	.083	11	.090
5	.118	12	.086
6	.125	13	.011
7	.122		

Because one might be interested in *pH* and *Meter* separately, these should be recorded as distinct descriptors. On the other hand, it seems pointless to split *General Electric* into two descriptors—unless, in the interest of obtaining improved selectivity with long words in a system with fixed  $N$ , this name be split merely to give it more punches. This, however, is unnecessary with the code of Table I. The identification of long names of organic compounds is one of the severest problems facing the student of coding. Systematic schemes have been devised by several people, and one of the Zator systems deals very neatly with organic compounds. Though I have not been particularly concerned with organic compounds, it is clear that Mooer's system could be used in the same file with an orthographic system without conflict. For casual indexing of organic compounds in a miscellaneous file, the orthographic code of Table I works sufficiently well, and, circumvents most of the immediate difficulties. *Trichloroethylene*, despite the length of the word, can be coded as a single descriptor without loss of selectivity; that is, the probability of its having the same code as any closely related compound is very slight. If it is so coded it cannot, however, be found by searching for *Ethylene* or for *Chloro* compounds. For a general file, one will have to exercise judgment in the naming, choice, arrangement, and splitting of long or complicated subjects; thus the searcher may have to try several different descriptors if there is doubt. This problem is common to all systems for the retrieval of information. For example, if a card refers to a table of the density of methanol at various temperatures, the descriptors might be listed as *Methanol*, *Methyl*, *Alcohol*, *Density*, *Temperature-coefficient*, *Expansivity*. And, if one wishes to fill the card with extra (admittedly possibly useful) words, *Specific-gravity*, *Cubical*, *Expansion*, *Coefficient*, etc. may be added. As this can be carried to extremes, one should strive to emulate the judgment so



admirably displayed in the indexes of *Chemical Abstracts*. Yet, the indexes of even that journal will not answer every reasonable question. How, for example, can one find all applications of flame photometry except by perusing the entire journal, abstract by abstract? Obviously the single-field superimposed coding system greatly simplifies such a search, provided the maker of the file has had the forethought to punch *Flame*, *Photometry*, and *Application* on every pertinent card. The good judgment used in *Chemical Abstracts* was a guide in preparing the cards which gave the values in Table V.

**General Theory of Dropping Fractions:** The evaluation of equation (5) requires the tabulation of  $X^{PG}$ . If  $N$  exceeds 1, the formulas for  $X^{PG}$  are of little help and a straightforward calculation by sliderule is as quick as any method except the use of a complex computer.  $X^{PG}$  was thus calculated throughout the significant range (up to  $D = 13$  and  $G = 13N$ ) for  $H = 55$  and  $N = 2$  and 3. To reduce arithmetic, a certain amount of extrapolation and interpolation can be made by means of graphs of  $\log X^{PG}$  against  $(X - G)$  and by means of approximate equations of the type

$$X^{PG} = 10^{aD - 10^b - cG} \quad (6)$$

in which  $a$ ,  $b$  and  $c$  are empirical constants. This plots as a straight line in certain ranges on semi-log-log paper.

If  $N = 1$ , it is easy to show that

$$X^{PG} = \sum_{n=1}^G (-1)^{G-n} {}_n C_G \cdot G^C H^{(n/H)X} \quad (7)$$

The general expression for dropping fractions in a system having no rule against coincidences among punches is

$$F_S = \sum_{G=S} \sum_{X=G} \sum_{n=1}^G (-1)^{G-n} {}_n C_G \cdot G^C H^{(n/H)X} \cdot P_X \quad (8)$$

in which  $P_X$  is the probability that a card has exactly  $X = ND$  punching instructions. The index  $X$  does not really depend upon  $G$  because the addends become zero when  $X$  is less than  $G$ , and (8) would still be valid if the second summation started at  $X = 0$ .

Whereas (8) gives an approximation to the true  $F_S$  in a practical system, it is desirable to evaluate  $F_S$  for systems with  $N = 2$  and  $N = 3$ , no coincidences among the  $N$  punches for a single descriptor being allowed. This is done by means of (5) and Table V. The purpose is to find out how much improvement there might be in going from  $N = 2$  with its relatively poor word-selectivity but favorable  $G/H$  values, to  $N = 3$  with its better word selectivity but larger  $G/H$ . (Note that for  $N = 2$ ,  $H = 55$ , there are only  ${}_2 C_{55} = 1485$  possible two-punch combinations, so that if there are several thousand different descriptors in the file many will inevitably have identical codes; but  ${}_3 C_{55} = 26,200$  allows a much greater effective vocabulary of descriptors without overlapping). The results for  $F_S$  become more instructive if we compare, not  $F_S$  for  $S$  equal to integral multiples of  $N$ , but  $F_T$ , defined as the average dropping fraction resulting from sorting the file for  $T$  descriptors.  $S$  will average less than  $TN$  when  $N$  exceeds 1, because of punching coincidences.

$$F_T = \frac{TN}{\sum_{S=N}} F_S \cdot TNPS = \frac{TN}{\sum_{S=N}} \sum_{G=S} \sum_{D=1} \quad (9)$$

$$TNPS \cdot DN^{PG} \cdot P_D \cdot SC_G / SC_H,$$

in which  $TNPS$  is the same function as  $DN^{PG}$  with  $D = T$  and  $G = S$ . Table VI shows the calculated values of  $F_T$  for  $N = 2$  and 3 for selection of one, two and three descriptors.

Table VI

Average Dropping Fractions  $F_T$ 

T:	1	2	3
N = 2:	.0584	.00454	.000473
N = 3:	.0438	.00319	.000296
ratio:	1.36	1.52	1.60

The advantage obtained with  $N = 3$  is not as great as might have been expected, considering the much larger vocabulary of distinct descriptor codes available with  $N = 3$ . By allowing for the greater time required to punch a code with  $N = 3$  and the greater time required to sort unwanted cards with  $N = 2$ , it can be shown that  $N = 3$  becomes the more economical if the file, regardless of its size, is sorted at least



several hundred times for a single descriptor or several thousand times for two simultaneous descriptors. As it is likely that many sortings will be done before the file wears out and must be duplicated, economy favors  $N = 3$ , but not by a great margin.

**Choice of Descriptor Code:** On the basis of these results it becomes possible to estimate, quite closely, the performance characteristics of certain descriptor codes which are more practical than those which simply assign  $N$  punches to each descriptor, regardless of length. Four codes which were considered are listed in Table VII. The symbols in this table are interpreted thusly: for Code A, exactly two punches (no coincidences allowed) are assigned to words of up to 7 letters; 3 punches, of which one may coincide with another (the average will be 2.96 punches when  $H = 55$ ), to words of 8 to 11 letters; 4 punches, among which up to 2 coincidences are allowed (i.e., at least 2 distinct punches required), to words of 12 to 14 letters, etc. For Code D, 3 punches are assigned to words of up to 4 letters; 4 punches (no coincidences) to words of 5 to 14 letters; 5 punches (one coincidence allowed) to words of 15 to 17 letters, etc. These descriptor codes are increasingly "dense," i.e., involve increasing numbers of punches per card.

The average number of punching instructions per card for any code is given by

$$X = \sum_L X_L f_L D_a \quad (10)$$

in which  $X_L$  is the average number of distinct punches (not punching instructions) for a descriptor of  $L$  letters (as listed in Table VII),  $f_L$  is the fraction of descriptors having  $L$  letters, and  $D_a$  is the average number of descriptors per card.

Table VII

## Practical Descriptor Codes

A:	2,	2.96 <sub>11</sub>	3.91 <sub>14</sub>	4.84 <sub>17</sub>	5.75 <sub>20</sub> etc.
B:	2,	3 <sub>11</sub>	3.95 <sub>14</sub>	4.87 <sub>17</sub>	5.78 <sub>20</sub> etc.
C:	3 <sub>11</sub>	3.95 <sub>14</sub>	4.87 <sub>17</sub>	5.78 <sub>20</sub> etc.	
D:	3,	4 <sub>14</sub>	4.93 <sub>17</sub>	5.84 <sub>20</sub> etc.	

The  $F_S$  for these codes must now be obtained from the  $X$  values found from equation (10), and by interpolation and comparison with the  $F_S$  for the pure codes  $N = 2$  and  $N = 3$  and

estimated  $F_S$  for  $N = 4$ . The labor of calculating  $F_S$  directly for  $N = 4$  would be very great, but the extrapolation is, I believe, sufficiently reliable. With the assumption that for mixed codes, like those of Table VII, the curves of  $(F_S)/S$  against  $S$  can be found by interpolation between the corresponding curves of pure codes (an assumption which must err on the safe side); and, with the  $f_L$  values, obtained by a count of 130 cards, shown in Table VIII, the pertinent data for the several pure and mixed codes were calculated. The results are shown in Table IX. The datum of chief interest in Table IX is  $F_d$ , the effective average dropping fraction for a one-descriptor search, given by

$$F_d = \sum_F F_{X_L} f_L \quad (11)$$

in which  $F_{X_L}$  is the function  $F_S$  for  $S = X_L$ ; if  $X_L$  is not integral. It is easy to interpolate  $(F_{X_L})^{1/X_L}$  between adjacent integral values of  $S$ .

It should be noted that the distribution shown in Table VIII concerned a sample for which  $D_a = 7.74$ , the value which was used in calculating the data for codes A, B, C and D and the values of  $X$  and of  $F_1$  by the Mooers and Wise formulas. On the other hand, the distribution of Table V implies  $D_a = 7.47$ ; this value is involved in the  $F_S$  for the pure codes. The corresponding errors in the  $F_S$  for the mixed codes are slight and, again, on the safe side.

Table VIII

## Frequency Distribution of Numbers of Letters per Descriptor

L	$f_L$	L	$f_L$
2	.010	11	.059
3	.027	12	.031
4	.067	13	.052
5	.146	14	.007
6	.094	15	.005
7	.120	16	.000
8	.139	17	.006
9	.092	18	.002
10	.143		

Whereas Codes B, C and D give nearly minimal values of  $F_d$ , the average dropping fraction for an average single descriptor, it is important to consider the  $F_S$  values for single descriptors of certain lengths. For Code A,  $F_2 = .096$  is the average dropping fraction for descriptors of up to 7 letters;  $F_{2.96} = .0327$  is the fraction for descriptors of 8 to 11 letters. For Code B,  $F_2 = .1235$  obtains for descriptors



of up to 3 letters, and  $F_3 = .0470$  for 4 to 11 letters. For Code C,  $F_3 = .0478$  applies to descriptors of up to 11 letters. For Code D,  $F_3 = .0835$  is the fraction for descriptors of up to 4 letters, and  $F_4 = .0442$  applies for 5 to 14 letters. Clearly any mixed code in which the shorter descriptors have  $X_L$  much below the average suffers from excessively large dropping fractions when the file is sorted for these shorter descriptors alone. Code C, which lies in the optimal range of economy, and which, by assigning extra punches to longer words, is not prejudicial to them, does not have this defect with respect to the short words. The shortest descriptors have 3 punches, and the average number of punches per descriptor ( $X/D_a$ ) is 3.11. For these reasons Code C appears to be almost uniquely the best descriptor code when  $H = 55$ , and the frequency distributions of numbers and lengths of descriptors is as shown in Tables V and VIII. It is this code which takes the particular form embodied in Table I.

#### EXPERIMENTAL TEST

*The Files:* The theoretical expectation of dropping fractions having been thus completely worked out, a test was made of the actual efficiency of files built upon these principles. A file of 442 cards was punched according to Code A, and another file of 126 cards was punched according to Code D. The two files were random portions of a well-shuffled file of 568 3 by 5 inch cards with the descriptors and references or abstracts typed on them. In preparing the abstracts the descriptors were typed or underlined in red to make them conspicuous for coding and sorting. Each card has a 1/4 inch hole in the upper right-hand corner and a 5/32 inch hole in the lower left, to permit checking the orientation of all the cards in the pack, and to facilitate sorting. The cards were punched with a hand-punch producing a notch 3/16 inch wide across the base, 1/8 inch wide at the inner edge, and 1/4 inch deep. Punches were located with a template card having the first 28 spaces, Aug through Lnz (Table II), marked on the lower edge, each space occupying 1/6 inch, and the remaining 27 spaces (Maf through Y) marked on the upper edge. In addition, for coding numbers or number-letter combinations, the first 25 even numbers (00, 02, 04, ..., 48) were marked above the first 25 spaces of the lower edge, and the next 25 (50 to 98) were marked in

the first 25 spaces of the upper edge. Odd numbers were punched each in the space belonging to the preceding even number. Numerical descriptors could then be treated like words, and were subject to the same descriptor code. In punching, the template was clipped to each card in turn, and the descriptors read and punched into the edges. One to two minutes were required per card — about 10 seconds per descriptor.

*Card Sorting:* A box operating like the Zator sorting box was made, with 28 holes, 6 per inch, accommodating 3 mm steel knitting needles. For a given descriptor, or set of descriptors, needles were set in the box corresponding to all the punches on one edge, and those cards keying to the needles on that edge could be made to drop by the depth of a notch (1/4 inch) below the rest, and were separated by skewering them through the holes in the corners. Because of the roughness of the edges and the slight and variable curvature of the cards, shaking a box did not satisfactorily drop the wanted cards if the entire pack numbered more than a few dozen. However, a jet of compressed air, played over the upper surface of the pack, proved to be very effective in forcing the dropped cards downward, permitting separation. The air jet consecutively and momentarily separated each card from its neighbors while simultaneously pushing it downward. The larger file of 442 cards could be sorted in this manner in a single pass. Each edge, of course, had to be sorted separately. The method was inefficient in several respects, but sufficed for the purpose of the test.

*Choice of Test Descriptors:* The file punched with Code A was tested for various values of from 2 to 7 with 101 descriptors, and the file with Code D was tested for  $S = 4$  with 67 descriptors. It became fairly evident that the manner of choosing descriptors influences the dropping fractions. If test descriptors are confined to those known to occur frequently in the file,  $F_d$  (for unwanted cards) tends to be larger than is the case when the descriptors are chosen from a sufficiently random source. In this respect, even *Chemical Abstracts* did not appear to be a sufficiently random source. It may be assumed that there must be at least  $10^6$  useful technical descriptors in the world, more probably  $10^7$ ; while the number of different descriptors in the test file was of the order



of  $10^3$ . The probability that a random descriptor should occur in the test file can therefore hardly exceed .001. Yet descriptors chosen from *Chemical Abstracts* were present in this file at least once in over 10% of the cases. This reflects the fact that both the file and *Chemical Abstracts* were biased in favor of chemistry. Because of these considerations, it was deemed permissible to separate the  $F_d$  values obtained with descriptors *not* occurring in the file (as representing more nearly a random sample from the universe of descriptors), from those obtained with descriptors *occurring* in the file (representing the vastly smaller group of subjects biased by fore-knowledge of the contents of the file).

of extra cards selected,  $A$  is the total number of cards in the file, and  $W$  is the number of wanted cards—those which should be and are selected. The total sort is  $E$  plus  $W$ . Because of the extremely small probability of the occurrence on any card of any one descriptor randomly chosen from the universe, the theory developed on the preceding pages applies to the unwanted cards in the file, and should be tested with  $E/(A - W)$ , not with  $(E + W)/A$ . Biasing of a file by partial specialization in certain subjects might be expected to increase the  $F_d$  even for random descriptors above the theoretical, but, as shown below, this does not happen, and the increase of  $F_d$  for occurring descriptors is not pronounced. That there apparently

Table IX

Characteristics of Descriptor Codes for  $H = 55$ .

Code	$X$ (Eqn. 10)	$\frac{X}{D_a}$	$F_1$ (Moore's)	$F_1$ (Wise)	$F_1$ (Eqn. 5)	$\sqrt{F_2}$	$\sqrt[3]{F_2}$	$\sqrt[4]{F_2}$	$F_d$
$N = 2$	15.5	2	.246	.248	.238	.243	.248	.251	.0594
$N = 3$	23.2	3	.345	.347	.334	.344	.352	.358	.0438
$N = 4$	31.0	4	.431	.434	.418	.432	.445	.456	.0435
A	20.3	2.63	.310		.300	.309	.315	.320	.0598
B	23.8	3.08	.352		.341	.351	.360	.366	.0466
C	24.1	3.11			.343	.354	.363	.369	.0448
D	30.3	3.92	.424		.411	.425	.437	.448	.0480

A file will be tested more often for descriptors expected to occur in it, than for those less likely to occur. If such biasing causes  $F_d$  to increase above the theoretical the extent of this departure should be known. But, on the other hand, to find whether the descriptor code and field partition yield results agreeing with the theoretical, a completely random sample of unbiased descriptors should be used—i.e., descriptors essentially not occurring in the file. This would seem to be a contradiction; but with a probability of occurrence of only .001 at most, the chance that 1 descriptor in 100 should occur is only 0.1 or less, and a sample of 100 nonoccurring descriptors may be considered at least 99.9% unbiased.

The files subjected to test were by no means completely miscellaneous. A little less than half of each was devoted to various aspects of the subject of flame photometry. In accordance with the tendency discussed above, descriptors chosen from within the subject of flame photometry appeared to give somewhat larger dropping fractions than other descriptors. It must be emphasized that these are dropping fractions of unwanted or *extra cards*, given by  $E/(A - W)$ , in which  $E$  is the number

is an increase with occurring descriptors supports the opinion of Wise that a multiple-field system is better for specialized files. But the magnitude of the increase is such as to indicate that a file must be quite highly specialized before the single-field system is to be preferred to a multiple-field.

**Test Results:** The results with the files of Codes A and D are shown in Table X. It is to be noted that the descriptor *Plasma* (in Code A) gave a particularly unfavorable  $F_2$  because its code was identical with that of *Flame*; the results are evaluated with and without *Plasma*. Malfunctions of this kind provide a good argument in favor of larger values of  $N$  despite apparently unimportant gains of average efficiency. One might say that the noise of the system is worse with smaller  $N$ . Additional tests with  $S$  above 2 with Code A were not sufficiently numerous to give results of significance; however, they agreed, roughly, with theory. The  $\sigma_m$  values are standard errors of the mean, obtained by dividing the standard errors of the individual  $F_S$  by the square root of one less than the number of trials.

Regarding the tendency for descriptors



occurring in the file to give higher dropping fractions, it can be calculated from Table X that for Code D there is a probability of .95 that the mean  $F_4$  for nonoccurring descriptors is actually less than that for descriptors occurring at least twice in the file; and for Code A the probability is .964 that the mean  $F_3$  for descriptors occurring not more than 3 times is less than that for descriptors occurring more than 3 times. But, if the case of *Plasma* is omitted, the probability becomes .946. Nevertheless, the totality of data makes it 99% certain that occurring descriptors give higher average dropping fractions than completely random descriptors. The differences of  $F_3$  for Code A or of  $F_4$  for Code D for the two classes of descriptors appear to amount to a factor having a probable value of only about 1.5, for these particular semi-specialized files.

$$Q = \frac{\log (1 - (F_S)^{1/S}_{theor})}{\log (1 - (F_S)^{1/S}_{obs})} \quad (13)$$

For example, if  $Q$  is .9 in a particular experiment, the average dropping fractions may be expected to equal the theoretical dropping fractions for the same code operating in a .9  $H$  spaces, and the system is as efficient as a 100% efficient system of .9  $H$  spaces. As mentioned earlier, Wise observed a  $Q$  of about .73 in using his word coding system. The efficiencies of the tested orthographic systems are shown in Table XI.

The general weighted mean efficiency factor for all descriptors not occurring "frequently" — i.e., not over once with Code D or three times with Code A (for which the file was three times as large) — is .99 with an experi-

Table X

## Observed Mean Dropping Fractions of Extra Cards

Code	W	No. of trials	S	$F_S$	$\sigma_m$ of $F_S$
A	0		2	.1047	.0150
	1		2	.0731	.0088
	0,1		2	.0927	.0101
	2,3		2	.121	.037
	0-3		2	.1016	.0107
	4 up		2	.149	.0240
	4 up (a)		2	.128	.0125
D	0	38	4	.0389	.0043
	1	14	4	.0583	.0128
	0,1	52	4	.0441	
	2 up	15	4	.0609	.0125
	all	67	4	.0476	

(a): Omitting the case of "plasma."

**Efficiency Factor:** To calculate the efficiency of utilization of the 55 spaces from the data of Table X, the  $S$ -th root of the observed  $F_S$  must be converted into the corresponding  $X$ , according to Mooers, by means of

$$1 - (F_S)^{1/S} = e^{-X/F} \quad (12)$$

This does not give the true value of  $X$ ; but if it is compared with the corresponding value of  $X$  found by equation (12) from the theoretical value of  $F_S$  from Table IX, the ratio of these  $X$  values (theoretical over observed) gives an efficiency factor  $Q$ , which is an accurate measure of the extent to which the field has been utilized. Hence

mental standard error of the mean of about .03. The weighted mean for strictly nonoccurring descriptors is also .99. These statements presuppose that the theoretical calculations are exactly correct. Unevenness of the distribution functions  $P_D$  and  $f_L$ , random errors of sample selection, uncertainties of interpolation, error in the assumption of applicability of the theory of pure codes to mixed codes, and inconsistencies in  $D_a$ , all contribute to the uncertainty of the theoretical  $F_S$  values. As pointed out earlier, these errors may raise, slightly, the calculated values above their true values. The efficiency factors found above may therefore be slightly too high; but this error is probably smaller than the .03 standard error of measurement of the final result.



Inasmuch as Codes A and D have shown practically perfect efficiency — an effective loss of perhaps no more than one space out of 55, despite a mean departure of 15% from uniform filling of the spaces due to inequities of the field partition — it seems certain that Code C, which was not tested but which, for the reasons earlier enumerated, is recommended as the best code, would be equally efficient. The dropping fractions shown for Code C in Table IX should therefore be accepted as valid.

### CONCLUSIONS

**Extension to Larger Fields:** The theory having been corroborated experimentally, it should be permissible to calculate  $F_S$  for codes similar to C for fields of any size. If  $H$  is increased, then, either  $D_a$  can be increased in proportion, the same descriptor code being kept and the dropping fractions remaining unaltered; or  $D_a$  can be left unchanged (with great improvement in dropping fractions); or  $X$  can be increased by increasing the average  $N$ , with some improvement of selectivity. Since  $NCH$  increases very rapidly with both  $N$  and  $H$  ( $N$  small and  $H$  large), there seems to be little advantage to increasing the average  $N$  above 4 when  $H$  is much more than 55. For instance, with  $H = 72$  and  $N = 4$ , there are  $4C_{72} = 1.0$  million different codes, and the probability of accidental coincidence of codes for different descriptors is negligible except for the very largest miscellaneous files, such as the card catalog of a great library. This should not be construed as an argument against more than 4 punches for long descriptors, which often differ orthographically in subtle ways and will defy distinction

unless the descriptor code is sufficiently generous of punches to minimize the likelihood of failure to incorporate those subtle differences into the code.

**Advantages:** To summarize, in particular: Code C for fields of moderate size will yield an average dropping fraction within 2% or 3% of the absolute minimum theoretically available; compared with Code D it entails 20% less punching and gives 7% better  $F_d$ ; and for no descriptor does  $F_S$  exceed .048 for  $H = 55$  whereas Codes A, B and D (and any other differing in like manner from C) will give about twice as great an  $F_S$  for certain descriptors.

In general: The labor of translating descriptors into orthographic code is so slight that it does not add to punching time and may be called negligible. By contrast, all other coding systems which I know, require a translation step which consumes time. The Zator system (which provided the basis and suggested the idea of orthographic coding), in its general form, requires the use of a dictionary or separate file to control the random number sets. This increases the labor of coding several-fold. There is absolutely no limit, in superimposed single-field coding, to the nature or amount of information (within the foreseeable bounds of human endeavor) that can be recorded in a file, and the decisions that have to be made regarding the manner of expressing this information in orthographic code are no more difficult than in the simplest possible indexing systems. Every concept must have a name, and almost all names are arbitrary and subject to uncertainty and variability. Further, there is no limit to the length or complexity of a descriptor, unless one employs absurdities. Because

Table XI

Observed Efficiency Factors

Code	W	Observed $(F_S)^{1/5}$	$\sigma_m$ of $(F_S)^{1/5}$	Q	$\sigma$ of Q
A	0	.323	.023	.95	.09
	1	.270	.016	1.17	.08
	0,1	.304	.016	1.02	.07
	2,3	.347	.053	.87	.16
	0-3	.318	.017	.97	.06
	4 up	.386	.031	.76	.10
	4 up (a)	.357	.018	.84	.06
D	0	.443	.012	1.01	.03
	1	.490	.027	.88	.07
	0,1	.458		.97	
	2 up	.496	.025	.87	.06
	all	.466		.95	

(a): Omitting the case of "plasma."



the translation step has been suppressed into a process of much simplicity, the probability of error is minimized. Economy and efficiency approach their theoretical limits, but with the retention of complete flexibility. Dropping fractions can be controlled by choice of field size, although for any new size recalculation of a good letter-pair partition is demanded; and in case the file is overspecialized its convenience can be improved by adding extra fields for direct, selective, or sequence coding. By suitable mechanization the coding and punching could be made simultaneous with the typing of descriptors on the cards, and, for sorting, a device might be envisaged for removing the desired cards in a single, simple operation.

**Mechanics:** For files of approximately 10,000 cards, hand-sorting is sufficiently rapid. It is probably most easily performed with cards of the Keysort type (McBee Co., Athens, Ohio), although Mooers claims excellent results with his shaking-box selector. The ideal card, for the code recommended in Tables I and II, should have 55 holes along the upper edge. By punching out the holes indicated by the code, a card is made which drops completely out of the pack when the pack is needed for any descriptor on that card. The use of compressed air in sorting appears to have advantages of speed, neatness, and prevention of excessive wear. Durability is important for cards which may be handled many thousands of times, and the use of air for sorting will result in less stress and friction than will the more conventional flexing and shaking techniques.

Keysort cards with 55 holes on one edge would be unduly large for most indexing purposes. A good size is the 3.3" x 7.5" card of the IBM type; but the McBee Co. considers it impracticable to punch more than 33 holes on an edge, allowing 1/2 inch at each corner. If  $H$  were to be reduced to 33, so that the code could be accommodated on one edge of an IBM-size card (5 holes per inch),  $F_d$  for the file studied in this report would increase to .15, a fraction intolerably high except for small files of perhaps not over a thousand cards. The probability of searching for one descriptor instead of more than one, multiplied by the time required to

hand-sort the extra cards for a one-edge system, must be balanced against the extra time required to sort both edges of a pack of cards on those occasions when sorting one edge yields too many extra cards. It would seem preferable, on this basis, if an IBM card with 33 holes on each edge were adopted, that  $H$  be increased to 66.  $F_d$  for Code C would then become .0286, an improvement of 56% over  $F_d$  for  $H = 55$ . In many searches the needling of one edge only would yield a sufficiently small sort to make the second needling either very easy or unnecessary. An alternative to the use of 66 holes would be to assign 55 for the field partition of Table II, and reserve the remaining 11 for special purposes. A further improvement would be the invention of a better system applicable to two rows of holes on one edge, with a deep and shallow punch permitting free dropping of the selected cards. It seems unlikely, however, that better over-all economy could be attained with any such system without elaboration of the mechanization. Although this investigation has been concerned primarily with the development of orthographic coding for the improvement of hand-sorted punched-card indexing systems, it is certain that the orthographic method can be applied profitably to any mechanical or electronic system for the coding and retrieval of information.\*

#### REFERENCES

1. C. N. Mooers, "Logic of Selective Systems," paper presented before the American Mathematical Society, Washington, D. C., April 1950. *Bull. Am. Math. Soc.*, V. 56: 349 (1950).
2. C. N. Mooers, *American Documentation*, V. 2; 20 (1951).
3. Zator Co., 79 Milk St., Boston, Mass., *Technical Bulletins* 30, 48, 55, 59 and 64.
4. C. S. Wise, "Mathematical Analysis of Coding Systems," Chap. 20 of *Punched Cards*, edited by R. S. Casey and J. W. Perry: N. Y., Reinhold, 1951.
5. C. S. Wise, "A Punched-card File Based on Word Coding." *Ibid.*, Chap. 6.
6. G. J. Cox, R. S. Casey and C. F. Bailey, *J. Chem. Ed.*, V. 24; 65 (1947).

\*In this connection it is interesting to note that this technique is quite analogous to that used by Ascher Opler, of the Dow Chemical Co., in encoding chemical formulas for searching with the larger IBM computers. Our readers may also be interested in knowing that one of the reviewers of this manuscript reported that this technique "is of extreme significance for hand-sorted punched card files because it will permit the recording of complex relationships among index entries." — Ed.



job, gives her special opportunities for observing at first-hand the problem of the degree to which documents can, or should, be made available to the reader. The Chairman for the meeting was Mr. G. A. of the Dunlop Rubber Company, Ltd., Chairman of the Branch. The paper, with a short summary of the discussion, is printed on p. 51. At the December meeting of the Midlands Branch Mr. J. Bird, Librarian of the branch, spoke on 'The role of professional periodicals in library work'. Mr. K. W. Humphreys, Librarian of Birmingham University, acted as the Chair. Mr. Bird took up the theme argued by Dr. Urquhart at the October London meeting, and discussed its implications for librarians from the point of view of training for future developments and responsibilities. The paper is printed in full in this issue.

The Northern Branch of Aslib held a half-day meeting on 4th November. Thirty-four members were the guests of the United Steel Works, Ltd., at the Swinden Laboratories, Rotherham. No papers were read.

Our last item we include a summary, by Miss C. V. Cutler, of the annual report of the National Central Library covering the period from 1st March, 1954, to 28th February, 1955.

E. M. R. DITMAS,  
*Editor.*

## ZATOCODING AND DEVELOPMENTS IN INFORMATION RETRIEVAL

By CALVIN N. MOOERS

*Proprietor, Zator Company, Boston, Massachusetts*

*London, 7th September, 1955*

### INTRODUCTION

IT is a pleasure, as a visitor to your country, to find myself invited to talk to this audience about information retrieval. The invitation, as it came to me, contained the implication that there was a considerable curiosity here about my activities in this field in the United States. Some preliminary reconnoitring before the meeting convinced me that this was indeed the case. Your curiosity, then, furnishes me with an excellent excuse to talk about a number of things that I have been doing over the past few years.

Information retrieval is concerned with the finding of information. Its problems can usually be considered quite apart from matters of how the documents containing the information are stored. My own particular interest lies in the devising and application of machines, and particularly digital machines like card sorters, to information retrieval. A little later I will describe in detail one of my machines and the Zatocoding System of which it is an essential component.

### RETRIEVAL MACHINES AND THE LIBRARY

The use of machines for the retrieval of information is a subject sure to stir up a good deal of interest among many audiences. Along with this interest, I find there are usually a number of misapprehensions about the application of the machines. For this reason I want to open by telling something about where these machine retrieval systems have been installed, what they are used for, and who decides to install and to operate them. The big surprise is the 'who'. Professional librarians have not taken a very active part in using these machines. This fact should be a matter of serious concern to the profession.

To my knowledge, no university library collection, nor large municipal or governmental library, nor any other large archival collection is organized by a machine retrieval system. At the present, the reason for this is primarily economic. The cost per item for organizing such a collection is too high to be justified. Machine retrieval systems are found almost exclusively in research or scientific applications. They are used to organize relatively small collections of documents of high utility such as research reports and selected parts of the journal literature. Among my clients the typical collection has about 5,000 items organized. They will have spent between \$1 and \$5 per document to do the organization.

A company that has a machine retrieval system will usually have in addition a special library collection with a librarian in charge. The



machine system is usually separate from the library, except possibly for the storage of the documents. In particular, this means that the intellectual aspects of dealing with the retrieval of these high-utility documents is not placed in the hands of a librarian. This situation is not accidental, and there are good reasons for it.

In a nutshell, the reason is: these documents are so valuable that the scientific staff cannot afford *not* to take over the intellectual organization of them. Modern retrieval machines have played a part in forming this decision. These machines make it possible for the scientific staff to do this organization without taking up an inordinate amount of their time. The machines have also eliminated certain deficiencies and constraints that have long driven scientific and engineering minds away from the techniques of classification and indexing. The machines permit new modes of information description and selection. These new modes have a very real fascination for the scientists. The scientists say that the new modes of selection make sense. What is more, they consider that the new techniques make so very much sense that they are actually willing to do the work involved in applying the machines to their information collections. In my experience, it is the director of research and his immediate staff who usually decide to put in a machine retrieval system. They are also the ones who follow through the initial stages of setting up the intellectual side of the retrieval system. Later, when the set-up period is over, the analysis of the incoming documents is handled by engineers or technicians. Typing and punching the cards is turned over to clerks. Librarians are seldom to be seen.

Librarians are left out of the picture because they all too often have neither the inclination nor the background that is needed. The majority of special librarians in the United States do not have any scientific experience or training. Most of them have majors in history, literature or languages. They cannot be expected to react to the technical content of a document in the same way a bench scientist will. Many librarians are uneasy about machines. Mathematics for most of them is a cause for panic. This is unfortunate, because the modern retrieval devices are machines, and a true understanding of the implication of their coding does involve a little bit of simple mathematics.

Perhaps the most devastating reason for not employing a librarian to help on a machine retrieval installation is that the very intellectual techniques of classification that have been drilled into him during his training are a major hindrance to his operating a modern machine retrieval system. These are fighting words, and I wish I could make a major digression to document them from my experience. In the course of my work in directing clients in the installation of machine systems I have had continuously to assist people in unlearning notions about classification as it is taught in library schools. Unless these rigid notions are abolished, one cannot take the full advantage of the power and flexibility in intellectual organization permitted by the machines. We shall discuss some of these points in the later section on descriptors. One example of the mischief of rigid notions is given by the unsuccessful experiment in

machine retrieval during 1946-49 at the Atomic Energy Establishment at Harwell.<sup>1</sup> Aside from a poor use and choice of commercially available machines, this experiment was much hampered by the strict use of the Universal Decimal Classification.

Where, then, does the trained librarian fit into the picture? It is my prediction, based upon observation of actual installations, that the working scientists will continue to supervise the intellectual aspects of retrieval of certain special classes of documents and information having a very high value to them. The librarian and his staff will co-operate at various levels to make this easier for the scientist, possibly by taking over part of the typing, clerical and warehousing aspects of the job. However, the classes of special documents needing this kind of detailed attention by the scientific staff will constitute only a fraction of a company's total documentary collection. The rest of the collection, being the majority of the total bulk, will continue to be the direct responsibility of the library staff. This part of the collection may or may not be organized by retrieval machines, depending upon a variety of circumstances. One of these circumstances will be whether the librarians in the near future will be ready and capable of exploiting the full possibilities presented by the machines.

#### THE ZATOCODING SYSTEM

Now, to change the subject, I wish to give a description of the Zatocoding System for information retrieval. This is the mechanical retrieval system with which I have had most of my experience. It is also a system about which many of you have expressed curiosity as to how it works, and why. With the help of some pictures and diagrams I hope to explain it to you.

At the outset, it must be realized that the Zatocoding System is a highly integrated retrieval system. The apparent simplicity of its various parts is deceptive. The parts have been devised or developed with a single goal in mind: to give the greatest user satisfaction in information retrieval. If a change is made in any one of the interlocked parts of the system, the change is usually found to react seriously and detrimentally upon the performance of the system as a whole. For example, Zatocoding (as a coding scheme alone) could be practised with 'needlesort' cards instead of with the Zator '800' Selector. The immediate penalty would be that card sorting speeds would drop down to one-fifth of the original speeds. Other apparently superficial changes in other parts of the system would be found to be quite as detrimental.

Zatocoding Systems, as they are currently employed, all have a well-defined target. They are not used for any library collection or any file of papers. Neither do we ordinarily organize collections of books. Books are often sufficiently well catalogued so that they already give an adequate degree of user satisfaction. Instead, the target is usually a collection of high-utility reports, journal articles or other papers. The collections number between 1,000 and 20,000 items. High utility of the items to be organized is the controlling factor, because a machine retrieval system

*depends on the sorting job!*



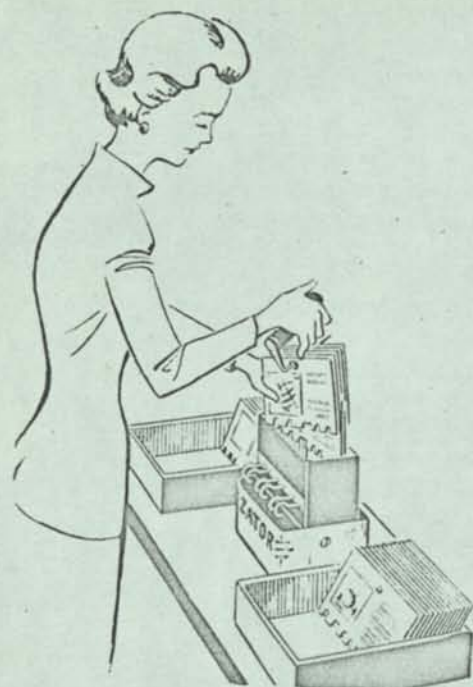


Fig. 1.—Sorting cards with the Zator '800' Selector. Cards are taken from one of the side trays, are sorted, and are then placed in the other side tray. The accepted cards are dropped to the table in front of the machine.

Zator '800' Selector and the edge-notched Zato cards. This is the most tangible part of the system, and yet in some ways it is the least important part. The second part is the new technique of random pattern subject codes with notch patterns representing subjects superimposed into the edge of the card. This is the Zato coding technique. The virtue of Zato coding is that it allows a very simple machine to perform rapidly a kind of search that otherwise (with non-superimposed code patterns) would require either use of a complex and expensive machine or would require a time-consuming sorting and resorting of the cards with a simpler machine. The third part of the Zato coding

involves considerable time and expense to set up and maintain. To balance this cost the pay-off from the accessible information must be relatively large and predictable. Thus a company will have a variety of retrieval systems. There may be a Zato coding System for a collection of research reports. A classified shelving scheme may be used for hard-cover books and bound volumes. A vertical file may be used for technical correspondence.

A Zato coding retrieval system has three parts. There is the mechanical part represented by the

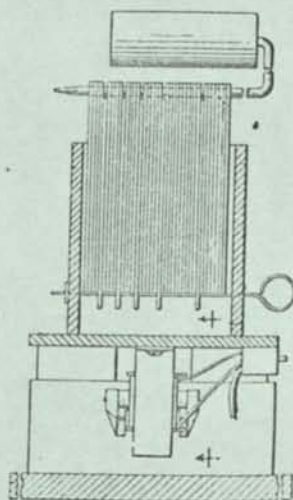


Fig. 2.—Cross-section of the Zator '800' Selector showing the vibrating motor and the manner in which most of the cards stay on top of the selecting rods.

System is the most important. It is the system of 'descriptors' by means of which retrieval questions are turned into prescriptions for search. Our use of descriptors represents one of the more subtle changes over prior practice, and it is a little hard to explain them without actually working with them. However, I shall try to tell how they and the rest of the system go together.

The Zato coding System is the kind of retrieval system in which one card is made up for each of the reports in the collection. Notches in the edges of the cards permit a mechanical sorter to scan the cards and to select some of them. The subject content of each report is related to the pattern of notches in the card by the coding scheme. Therefore the sorter is able, by a strictly mechanical process, to select cards from a pack according to subject matter of the reports which the cards represent. All the cards are scanned for each retrieval question. Scanning all the cards every time is not a disadvantage when the selector machine is as fast as the Zator machine. Such complete scanning has an enormous advantage in that the cards need not be kept in any order. All card filing is thus eliminated.

#### THE ZATOR SELECTOR

The selector machine is the easiest part of the Zato coding System to understand. Figure 1 shows the Zator '800' Selector in operation. A pack of about 200 cards is placed into the black boxlike top of the selector. The box is vibrated by a little motor, as shown in Figure 2. Near the bottom of the box are rods or needles running from front to back. Each of the rings you see in the picture is attached to a rod. By means of the rings it is easy to pull out the rods and to insert them again in a different selective pattern. The Zato cards, like the one shown in Figure 4, bear notches in their edges representing the different subjects. To select upon a pack of such cards, the pack is placed in the selector machine with the notched edges down. That is, the notched edges of the cards rest upon the sorting rods. Most of the cards in the pack will rest upon the top of the grid formed by these rods. However, some, as shown by Figure 3, will have notches in the position of each of the selector rods. These cards will not be supported on top of the rods. They will shake down from the rest of the pack. They are the desired cards. Looking again at Figure 3, we see that the rejected cards can now be engaged by a rod which is stuck through the line of holes near the top edge of the cards. The desired cards, having slipped down a little bit, are not so engaged. When the rod

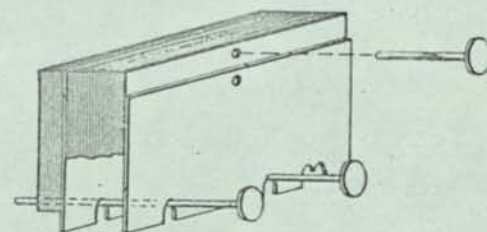


Fig. 3.—Diagram of how the cards whose notches fit the pattern of the selector rods drop down from the rest of the pack.



is lifted, the rejected cards are carried on the rod and are lifted out of the selector. The desired cards drop free from the pack to the table top.

The whole operation of selection with the Zator '800' Selector is very easy. The selector holds an easy handful of cards. About one second of the vibrating action of the selector is sufficient to separate the cards. Therefore the sorting speed really depends upon how nimble you are with your hands. Speeds of better than 800 cards per minute are easily attained. Thus the designation, the '800' Selector.

Before long I expect to make available a Zator '200' Selector. It will have the advantages of being cheaper, of not requiring an electrical connection, and of having brief-case portability. Its speed of 200 cards per minute, while lower than the other selector, is nevertheless on par with Hollerith machine card-sorting speeds.

Zatocards come in two styles. One style has notches only in a single edge; it has only forty notching sites. The style of card that is preferred by the commercial clients has two edges given over to notches; it has a total of seventy-two notching sites. Nearly twice as much descriptive indication can be notched into such a card. In sorting these double-edged cards the selector is set up to scan the top edge of the cards. All the cards are run through. This gives a partial selection, a pack amounting to only a few hundred cards. The selector is then set up for the patterns on the bottom edge of the cards. The small pack of partially selected cards is run through. It goes very rapidly because there are a few hundred cards at most. The cards that come out of the second selection are the desired cards. Most of the selection time is taken by running the first edge of the double-edged cards. Thus the speed of selection is almost independent of which style of card is used.

#### THE RANDOM CODE SCHEME

The second part of the Zatocoding System is the random superimposed coding scheme called Zatocoding.<sup>2</sup> To explain its significance, I might point out that there are two different ways of coding information into notches in the edge of a card.<sup>3</sup> In the first way of coding, the pattern of notches for any subject 'A' is carefully kept separate and distinct on the card from the pattern of notches for any other subject 'B'. This is the common-sense way of doing things, and is the way that has ordinarily been used. It has the disadvantage that one never can be sure whether it is the pattern for 'A' or the pattern for 'B' that is notched on the right-hand side, in the middle, or on the left-hand side of a card. In scanning you either have to make a separate search for each possible location, or you have to use a complicated and expensive machine to try all the many combinations for you.

In the other way of coding, which is the Zatocoding method, the patterns of notches for both subjects 'A' and 'B', and the patterns for any other subjects besides, are all overlapped or superimposed in an undivided area of the card. One would think that this would lead to an awful mix-up. The strange thing is that it doesn't, provided you go about the matter properly, using the Zatocoding method. In order to keep the patterns

from resembling each other too much each pattern is made as different from the others as possible. One way to do this is to use random patterns. These are patterns generated by flipping a coin, or by some other similar means. In fact, any patterns that are 'random-like' in the sense that they have very little resemblance to each other, and that have notches falling with approximately equal incidence on all the sites, can be used for coding. Numbers that you make up out of your head or that you might take from the telephone book are not random, because certain digits and combinations will recur too often.

The Zatocoding method of using superimposed random-like code patterns is illustrated by Figure 4. On the card shown, the various subject descriptors have been written out. Opposite each of the descriptors are

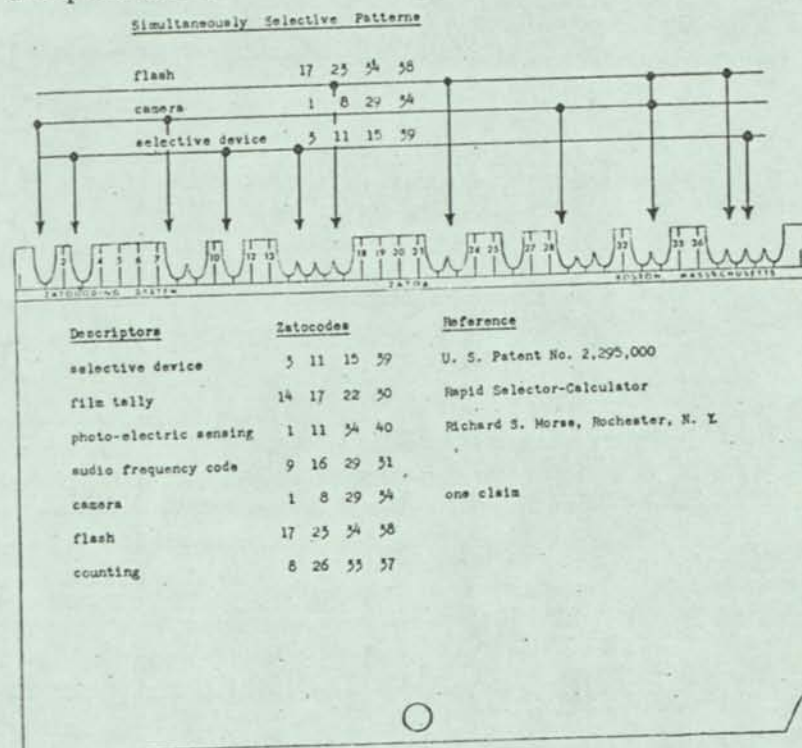


Fig. 4.—A Zatocard illustrating the manner that the random-like codes are superimposed in the edge of the card. The lines and arrows at the top illustrate how the codes for the three selective descriptors are combined to give the selective pattern. The rule of selection is that the pattern of the arrows must be included within the pattern of notches on a selected card.

the four Zatocode numbers corresponding to the four notches of each descriptor pattern. In the cards of an actual installation the codes would not be written on the cards in this way, though the descriptors would be.

Figure 4 also illustrates the manner in which selection is performed by



Zatocoding. In the case shown, three subjects simultaneously define the desired selection. They are: 'flash', 'camera' and 'selective device'. The individual codes, and the method of their combination to form the total selective pattern, are shown by the diagram. The arrows correspond to the total selective pattern of rods as set up in the card selector. Evidently the card shown will be selected, because there are notches in every position where there is a selector rod. Note that a selected card may have more subjects (and thus more notches) than the selecting prescription. The selector extracts all cards that have at least all of the prescribed subjects. Fortunately this is the kind of selection that is most useful for an information retrieval system.

This kind of selection is also easy to mechanize with a simple machine. All the patterns are coded into the one area of the card, so the machine need be able to look in only one place. It is as easy to look for the joint occurrence of several patterns as it is to look for one pattern. This is why a simple machine like the '800' Selector can perform complicated jobs of retrieval.

Zatocoding has one harmless peculiarity that often disturbs people far more than it should. When one sets up the codes in the selector and then carries out the selection operation on a collection of cards one gets a pack of selected cards. Besides the handful of cards that contain the subjects that were prescribed, there will often be two or three cards whose subjects do not correspond in any way with the prescribed subjects. These are the 'extra cards' of Zatocoding. They are harmless, because they are so few in number and are so easy to discard. We do not worry about them. Absolutely none of the desired cards with the prescribed subjects is missed by the coding. This minor foible is a logical consequence of the use of superimposed coding. Superimposed coding has the great advantage of permitting a very simple selector machine. For this advantage the few extra cards are the price we pay. By varying the number of notches in each Zatocode pattern we can make the number of extra cards as small as we desire. Thus the foible is completely under control.

While Zatocoding has quite an interesting mathematical and statistical foundation, it is not necessary to be a mathematician to use the method. I like to look upon the Zatocoding method as a mere mechanical part of the system. By now its design is thoroughly tested by practice. Like the machine selector, it can be used according to some very simple instructions. We do not worry about the theory, since we know it will operate quite as predicted. For this reason we have no difficulty in letting clerical personnel assign Zatocodes to subjects, code or punch the Zatocards, or work the machine selector.

#### THE DESCRIPTOR DICTIONARY SYSTEM

In sharp contrast to the straightforward mechanics of the code scheme and machine is the third part of the Zatocoding System. This part is called the descriptor dictionary system. For successful performance of an information retrieval system it is the most important part. I call it a

dictionary 'system' because it is not merely a list of subject words. Instead it comprises several different kinds of lists, each having a definite set of rules for its use. The descriptor dictionary system is primarily an intellectual tool. It is the means for coupling the mind of the information searcher to the hardware of the Zatocoding System so that the hardware can do the work of selecting wanted subject matter from the file.

An information retrieval installation, and particularly the dictionary system, should be orientated towards the requirements of the user. This might seem to be an elementary and generally practised principle. It is not, however, as we shall see when we examine some of the consequences of the principle and compare them with some accepted library practices. For one thing, user groups differ. One organization may be concerned with the use of polymers in the manufacture of pressure-sensitive tapes. Another organization may be concerned with the physical chemistry of polymers used in adhesives. To a first approximation their library collections may be identical. If the orientation of the retrieval installation were to be made primarily towards the content of the collections, the descriptor dictionary systems would be nearly identical. This kind of orientation seems to be the tendency in accepted library practice. That is why librarians find it so odd when I say that two collections of this kind may have descriptor systems that are quite different. In my installations the descriptor systems are different to the extent that the two groups of users are faced with different types of problems; or that they ask different questions; or that they ask them in different ways. To give the greatest convenience to each user group I develop for each a special vocabulary of descriptors. [This technique is described in the answer to the query by Mr. Thorne in the discussion.] Only in this way can the high intrinsic value of a special information collection within a company or agency be matched by the performance of a retrieval system.

Now to answer the question, 'What are these descriptors, and how do they differ from what librarians have been using all along?' A descriptor is closely related to the 'subject heading' of library practice, though a descriptor is usually broader in meaning. For instance, a subject heading might be: 'oils—effect of temperature on viscosity'. In descriptor analysis we would use the separate descriptors 'oils', 'thermal' and 'viscosity'. These descriptors, when taken together, delineate the idea of the subject heading. Each descriptor stands for an idea or concept, generally of rather broad scope. The descriptor word is merely a symbol for an assigned descriptor meaning. Descriptor meanings are chosen or assigned in a way that will facilitate retrieval by the user group. Retrieval meanings need not conform strictly to standard technological usages of the descriptor word. Because the meanings are often slightly different from the ordinary usage, it is essential that the descriptor dictionary system includes a list of 'scope notes', one note for each descriptor. We use an alphabetically arranged list of scope notes to make the full range of chosen meanings easily accessible to anyone desiring to use the Zatocoding System. Let me reassure you that these special descriptor meanings are private meanings, for use in retrieval only, and that there is no intent



(nor likelihood) of imposing them upon ordinary speech or technical writing within or outside a company.

The fact that descriptors are conceptually broad has a number of desirable consequences. Because they are broad, the user's intellectual universe of search can be covered by a relatively small list of descriptors. It may seem strange to a librarian, used to thousands of subject headings, to learn that our retrieval installations employ only about 250 or 350 descriptors. Because there are so few descriptors, they are relatively easy to remember. This is a real advantage in using them. The very breadth of meaning of each descriptor makes it easy to decide its applicability to a given document. We try to avoid finely drawn distinctions between closely related ideas and descriptors. Precision is not lost by using the broad descriptors. Narrow ideas can almost always be synthesized by the use of several descriptors. Since there are so few descriptors, we ordinarily try to set them all out on a single sheet of paper in such a way that they will be easy to find and to use. We call this sheet the 'descriptor schedule'. Copies of the schedule are put in the hands of the scientific personnel who use the installation. They are also given a booklet containing the scope notes and a set of simple directions for using the retrieval system. To further assist in finding and using the correct descriptors the alphabetical tabulation of the descriptors and their scope notes has interpolated in it cross-references, from various words and expressions that are not descriptors, to presumed descriptors.

The orientation towards the user is particularly evident in the manner of analysis of the incoming documents. Here again we find significant departures from accepted library practice. In analysis we make no attempt to take the message of the document and to write a little abstract using descriptor words in such a way that the message of the document is preserved. I believe this philosophy is mistaken. It is the same erroneous philosophy that tries to use the infinite range of possible U.D.C. symbols as highly precise words in an artificial language, and that tries to express and to preserve a document's message by one or two of such symbols chosen for their 'pin-point' precision. At the symbolic and coding level retrieval, and not message preservation, must be our goal. I have turned the philosophy inside out. We use a small schedule of individually broad descriptor concepts, and in analysis we choose the half-dozen or dozen descriptors whose broad meaning touches upon the meaning in the document, yet we make no attempt to preserve the particular message of the document.

This philosophy of analysis was determined by our decision to put the person analysing the incoming document into the same kind of a situation that the information user faces. The user has a difficult problem. He is confronted by nothing but a schedule of descriptors, elaborated by some scope notes. With these tools he must discover how to retrieve information whose nature may be in large part unknown to him, but which, when he sees it, will appear useful to him.

In the analysis process we try to put ourselves in this user's shoes. First we scan or read the document. We then put the document to one

side and we concentrate upon the descriptor schedule. We work down the page, descriptor by descriptor, exactly as if it were a check list. For each descriptor we ask, 'Would anyone who would be interested in the message of this document try to use this descriptor as part of his retrieval prescription?' Or, 'Does the meaning of this descriptor touch in any way (relevant to the user group) upon the message of the document?' If the answer is 'yes' for any descriptor, this descriptor is chosen as one of the descriptors to characterize the document. This is really a filtering technique, and we call it that. We filter the schedule of descriptors through the message of the document. Those that stay in the filter are the chosen descriptors. Cases of doubt about the applicability of a descriptor are resolved by choosing it. Such a doubtful descriptor may be just the one that will later be tried in a retrieval prescription by some eventual user.

This very definite and explicit use of the filtering technique seems to be new in retrieval systems. For us it has proven to be invaluable in giving the systems a consistent intellectual structure. Consistency is a real problem. There may be as many as six or more contributing analysts. This group will change over the years. Yet their efforts, in the form of notched cards, accumulate. We are quite satisfied with the high level of consistency that has been achieved when the filtering technique has been rigorously applied.

Another desirable consequence of the filtering technique is that it allows us to make a substantial downgrading in the required level of technical background or competence of the analysing personnel. If there were no recourse to a filtering approach there would be heavy demands upon the ability of the analyst, because he would have to foresee all the possible future uses of a document. Of course, this is very difficult. The schedule of descriptors eliminates most of this problem, because in a fashion it is a check list of future contingencies as worked out by the top people in the laboratory. Yet, to use it, the analyst need make only very simple judgments. We have found with many kinds of documents that an intelligent technician can do the analysis on almost all the documents, with only the most difficult items being saved for analysis by an engineer.

The burden of using a schedule of 250 or 350 descriptors can be eased by a simple device. In Figure 5 is shown a portion of about one-quarter of the descriptor schedule of one of my clients.<sup>4</sup> It is seen that the descriptors are grouped, with each group being composed of rather similar descriptors. At the top of each of the groups there is a question, such as, 'Is there a type of fluid flow?' In using this kind of a schedule the analyst first looks at these questions. If the answer to any of the questions is 'yes', then the analyst picks out the one or more appropriate descriptors below the question. If the answer is 'no', he goes on to the next question. Use of the filtering technique in this dictionary system then amounts to working through a list of about twenty questions rather than through 250 individual descriptors. Carefully chosen 'leading' questions as in this example can make the analysis particularly easy.

I wish to stress that this grouping of descriptors is not a classification. Grouping is merely a device for convenience. Another device for con-



venience is that the same descriptor may appear in two or more groups, which is certainly against all canons of classification. We find that an alphabetically arranged schedule of descriptors is useless for analysis by the filtering technique, and it is not used.

An actual analysis proceeds as follows. The first decision of the analyst is whether to include the document (or class of documents) or not. Obviously worthless matter should not be allowed to raise costs or to dilute the system. In most cases, by the time the analyst sees the documents they have passed a threshold test of utility. He then scans or reads the document. Depending upon the obscurity of the writing, or the richness

What material was studied?	Is the process dynamic (rather than static)?	Are there specific aerodynamic loads?	Is structural strength and elasticity involved?
Metals	Vibrations	Lift	Stress and strain
Gases	Transient response	Drag	
Plastics	Impact	Moment	Plasticity
Aluminum	Stability	Gust	Failure
Magnesium	Velocity	Pressure	Ultimate properties
Titanium		Center of application e.g., aerod. center, center of pressure, etc.	Material properties
Air			Aeroelasticity
			Flutter

What is the type of fluid flow?	Is it a stability and control problem?	Or is there another aerodynamics problem?	Is a thermal process involved?
Fluid flow	Stability	Boundary layer	Thermodynamics
Internal flow	Control	Aeroelasticity	Thermodynamic-constants
Subsonic	Static	Flutter	Combustion
Transonic	Dynamic = Trans. resp.	Downwash	Heat transfer
Supersonic	Longitudinal	Stall and buffet	Cooling
Hypersonic	Lateral	Interference	Convection
Laminar	Derivatives	Hydraulics	Conduction
Turbulence	Damping	Trajectory	Thermal
Slip flow	Weight and balance e.g., center of gravity, moments of inertia, etc.	Droplets	Radiation
Compressibility		Modifying Technique	Aerodynamic-heating
Viscosity		Performance	
Vortices			
Shock waves			
Finite span			

Fig. 5.—Part of a typical schedule of descriptors showing the grouping of the descriptors and the manner in which leading questions are used.

of the content (there is often an inverse correlation), this scanning takes from five to twenty-five minutes. Fifteen minutes is not a pessimistic average for technical reports. He then takes up the descriptor schedule and uses the check list of questions, writing down the chosen descriptor words on the Zatocard. This takes him about two minutes. The card then goes to the clerical staff for typing the title, and sometimes the abstract, of the report, and for coding and punching. The significant point about the analysis is that the greatest part of the analyst's time is taken by his merely gaining a familiarity with the message of the document. If the content of the document is to be probed to this depth, there can be no short cut to this time for analytical assimilation. Let me stress that this

step of assimilation accounts for 50 to 75 per cent. of the total cost of running a retrieval system! No mere change from one library method or retrieval machine to another can alter this time of assimilation.

Of the various parts of the descriptor dictionary system, I have so far described the schedule of descriptors and the alphabetically arranged list of scope notes. This brings us to the actual coding portion of the dictionary system. Since Zatocoding uses random code patterns for the descriptors, and since random patterns are rather tricky to remember and to transfer from book to card, we have a problem in accuracy. Its solution lies in the elimination of the mental transfer step. Our technique is illustrated in Figure 6, which shows part of one page of a code dictionary. The descriptor words are listed alphabetically down the page. Across

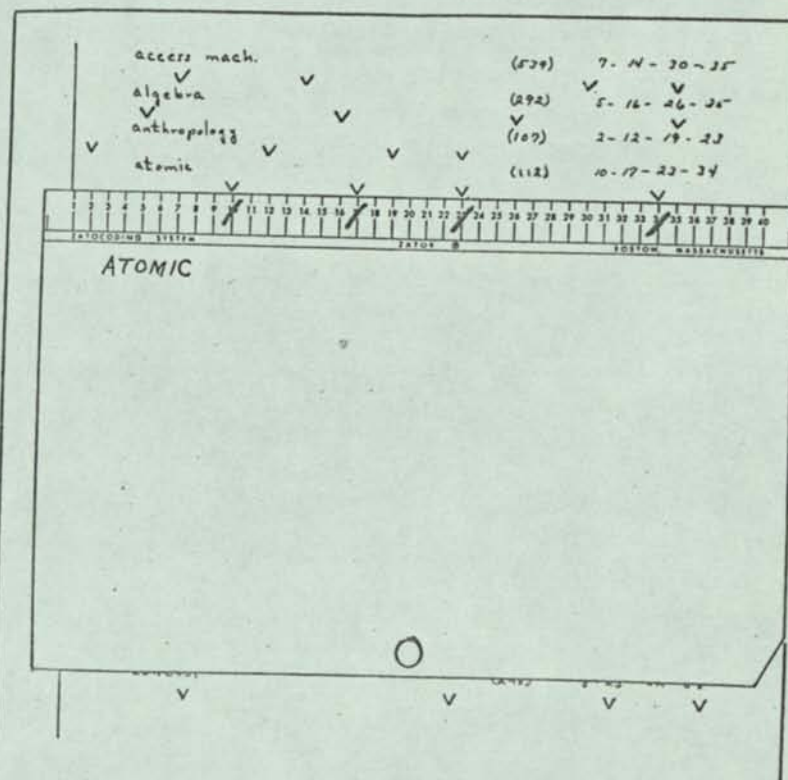


Fig. 6.—The code dictionary in use for transferring code patterns to the Zatocards. Positions of the code notches are marked on the card by pencil, and are punched later.

each line are the descriptor word, a control number for the code pattern, which I will ignore, and the group of four numbers representing the four randomly placed notches for the descriptor pattern. To use this code dictionary the coding clerk reads a descriptor from the card, finds the page and line of the descriptor, and lays the card down on the page under the descriptor entry. He aligns the notching position 'number one'



of the card with the vertical fiducial line on the page. When the card is in this position the 'V'-shaped marks on the page coincide precisely with the positions on the card that are to receive code notches. The clerk transfers these locations to the card with pencil marks. There is no mental step nor remembering of code patterns. Accuracy is very high. After all the descriptors on the card have been coded, the marked sites are notched out with a hand 'ticket punch'. The completed card is then tossed anywhere in the file.

The last component of the descriptor dictionary system that I shall discuss is the device we use for dealing with authors' names, company names, trade-marks, and the like. Instead of setting up these as individual descriptors, and explicitly assigning them code patterns, a technique of ciphering is used to produce the random-like code patterns. This process,

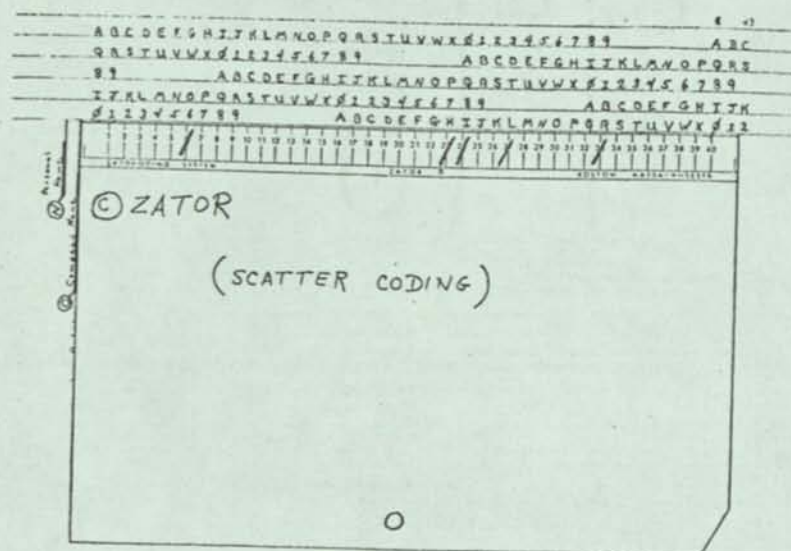


Fig. 7.—Diagram showing the method of scatter coding which is used for authors' names, corporate names, trade names, etc.

which we call 'scatter coding', is illustrated in Figure 7 for the company name 'Zator'. A card to be coded is laid down on the scatter code sheet with the left-hand edge of the card opposite the index line 'C' for company names. Then the first five letters of the name are spelled out, using the five rows of alphabets on the sheet, beginning at the top. Note that these alphabets are displaced so that letters of high frequency like 'e' will not coincide at the same site on the card, and so that there will be an approximate uniformity of incidence of notches in the manner required by the Zatocoding method. These scatter codes are sufficiently random-like, provided that no more than three or four are used on one card. We have some standard rules to eliminate useless parts of names like 'corporation', and to do so in a consistent and repeatable fashion.

There are a large number of possible ciphering methods for scatter coding, all of which produce random-like codes. The one shown has simplicity, but it would not be suitable for coding all the items notched into the card. In a case where a cipher method must be used exclusively, as in a collection of names, we have found that a more elaborate cipher based on letter pairs will give the degree of randomness called for by Zatocoding principles. Other workers have since rediscovered ciphers of this class and, in a burst of enthusiasm, have gone on to suggest that all word lists and code dictionaries can thereby be abolished. This suggestion is not as good as it sounds. For one thing, ciphering is a mental operation and is very prone to mistakes. What is worse, if the descriptor list is eliminated it is impossible to use the filtering technique or to secure its considerable virtues.

The proof of a retrieval system is in its using. This brings me to a matter that is seldom considered in discussions of retrieval techniques or machines. The information user is generally unable, for a variety of very good reasons, to prescribe at the outset exactly what he wants from the document collection. What happens is that he formulates the best trial prescription for a search that he can, and he sorts the cards. He looks over the selected cards. On the basis of the titles and abstracts on the selected cards, he gets a better idea of how to prescribe a second search. Then he makes a second search, or even a third search, each time getting more useful information and better citations. Such a succession of searches is the only way to find things when the original question is diffuse or when the needed facts or theories are unknown to the searcher. A succession of searches cannot be avoided by improving the retrieval system. It is due to a shortcoming in the user. In this situation, the retrieval system should be planned and used so that the machine helps the user to overcome his limitations. The machine can do so by providing a feedback of information from the collection to the user, educating the user as he searches, and enabling him to converge on the information he needs.

In order that such a cyclic search process may operate, the machine search time must be relatively short, in the order of only a few minutes. One of the reasons that a good index such as the Industrial Arts Index is so useful and generally satisfactory for some kinds of retrieval is that the cyclic search time is very short, and that intermediate answers in the form of titles are available quickly. This time factor, more than any other, sets the upper limit to the size of collection that I would recommend for retrieval by the Zator '800' Selector. A thirty-minute search time is about as long as a scientist or engineer would like to wait for an answer, and this corresponds to about 25,000 cards. By this same consideration, I seriously doubt the practicality of some suggestions advanced for setting up large-scale retrieval machines in a few central geographical locations. Most workers would have to query such a system by telephone or mail, and the cyclic search time would be in the order of days or perhaps a week.

#### FUTURE DEVELOPMENTS

It seems to me that we can expect interesting future developments in



retrieval devices along at least three lines. The first is the appearance of some new high-performance devices which are inexpensive, portable and suitable for use by the individual scholar or scientist. I feel this is a safe prediction, because I have two of such devices now in development. The second line is the application of retrieval techniques to large-scale general-purpose computing machines and to large, complex special devices in the class of the Eastman Minicard machines. This will be of particular importance to governmental bureaux having large bodies of information that must be serviced for purposes of their own internal operations. Servicing of Social Security records is a case in point.

The third line is a little more speculative, but rather more exciting. It is the possibility that retrieval devices for certain very large collections will become both simple and relatively cheap. I am presently studying one class of such a device, which I call 'Zatopar'. This device will allow the user to 'plug in' a set of descriptors and, within one minute, to get out the volume and page numbers, or serial numbers, in a list of citations. My speculations anticipate table-top devices containing no electronics or elaborate mechanisms. The common denominator of such inexpensive retrieval devices is that they be manufactured in an 'edition' like a book, and that they be used to index a widely used collection of information. An example is the indexing of *Chemical Abstracts*, with about 1.5 million items, by a device costing perhaps less than \$1,000. Smaller collections could be serviced by cheaper devices. Considerably larger collections seem to be within its scope.

For any device to be able to handle up to a million or more items, with cyclic search times of but a few minutes, and yet be cheap, presents no little problem. Certainly the aspect of cost alone precludes expensive electronic components. Nevertheless, it has seemed that a heavy use of electronic circuitry has been an inevitable part of proposals for achieving high-level retrieval performance. I believe that this dependence upon electronics is a transitory situation. In another paper<sup>5</sup> I have explored the thesis that memory capacity (such as marks on a record), which is cheap, can be traded for manipulative organs (like electron tubes), which are expensive. This problem was studied with respect to the retrieval of structured information (e.g. electronic schematic diagrams and chemical formulae) for which all previous approaches had required the use of expensive manipulative organs. My conclusion was that the trade of memory capacity for manipulative organs could be made, and that the trade was very desirable for at least a large class of structured information.

The essence of the argument leading to my conclusion is easy to state. In the past we have performed the analysis of documents and have coded the descriptors or other descriptive characters into the retrieval system memory in a straightforward fashion, quite as a telegraph codes the words of a telegram. When the time came for a search to be made we required the search machine to figure out whether our search prescription had any relevance to each and every one of the coded representations for each document. Expensive manipulative organs were therefore called into action for each and every search, requiring a costly retrieval machine. My

paper showed how the descriptors furnished by the analyst can be predigested into a new kind of codes for the retrieval system memory. When economy of memory capacity is not the object, the descriptor codes can be preset into all the possible combinations that could ever be needed for search. Fortunately, when we use an appropriate theoretical basis the number of combinations is not too large. Since this job needs to be done only once, we can hire an electronic computing machine to do the work of preparing the 'copy' for the retrieval system memory. Thereafter, the selective machine need only be capable of the simplest kind of a matching operation upon the prepared codes. Such a search machine can be exceedingly simple, with little or no loss in capability. There are probably several possible ways that this technique of predigestion can be carried out. The one in my paper depends upon the mathematical lattice properties of interlocking sets built upon the descriptors of the documents. This particular technique is applicable to a variety of machines, including those of the rapid selector category, the Minicard machines, the simplest sorted card systems, in addition to the Zatopar device for which it was studied. When the technique is combined with the simpler forthcoming high-performance retrieval devices, some very attractive joint cataloguing and union cataloguing projects for ordinary library materials become a future possibility.

#### EPILOGUE

Machines used for information retrieval are right now a very disturbing element. Developments have been so rapid that before a librarian or research administrator has been able to understand one process several new methods have been announced. While there is admittedly much confusion at all levels, there is, I believe, much reason for optimism. We are probably at the threshold of practical solutions to some very old (and some very new) problems in the retrieval of documentary information. The challenge is large, but the opportunity for us all is quite as great.

In closing, I want to acknowledge my deep indebtedness to my clients who have suggested or initiated so many of the practices and ideas appearing in this paper and who have had the courage to try out these ideas in their retrieval installations.

#### REFERENCES

1. ASHTHORPE, H. D. 'The punched card indexing experiment at the library of the Atomic Energy Research Establishment, Harwell.' *Aslib Proceedings*, vol. 4, pp. 101-4 (May, 1952).
2. MOOERS, C. N. *Improvements in or relating to encoding information on punched record cards and other equivalent media*. British Patent No. 681,902, filed 3rd September, 1948. U.S. patent pending.
3. MOOERS, C. N. 'Zatocoding applied to mechanical organization of knowledge.' *American Documentation*, vol. 2, pp. 20-32 (January, 1951).
4. This schedule is reproduced through the courtesy of Allied Research Associates, Boston, Massachusetts, U.S.A., who use it in their Zatocoding System.
5. MOOERS, C. N. 'Information retrieval on structured content.' Chapter in the book *Proceedings of the Third London Symposium on Information Theory* (1955), edited by Colin Cherry. Butterworth, London (in publication); also in an expanded version in *American Documentation* (to be published).



### Discussion

MR. AGARD EVANS (Ministry of Works) asked whether the high percentage of retrieval claimed by Mr. Mooers was dependent upon the size of the collection.

SPEAKER replied that efficiency in retrieving desired references is independent of the size of collection, but does depend upon the adequacy of analysis of the documents. The extra cards are another matter, and for a given selective prescription they occur as a fixed fraction of the total collection. The same prescription on a larger collection will give more extra cards.

DR. D. J. CAMPBELL (Institute of Cancer Research) commented on the New York Electron Microscope Society's bibliography on punched cards, the coding of which was obviously based on Mr. Mooers' ideas. The system did not allow for subject approach at all levels. To find all material on tumours, twenty-six five-letter combinations had to be used.

SPEAKER said that these cards did not use the superimposed coding described in the paper, but used another form of numerical coding. The difficulty mentioned by Dr. Campbell is not due to the coding method. It is the fault of a poorly worked-out schedule of descriptors. This fault is probably a consequence of the group's lack of experience in setting up retrieval systems. The speaker said he had found it essential to work closely with each of his clients while they were setting up their descriptor systems. Left to their own devices, the clients were sure to make mistakes resulting in problems of this kind—and a few other mistakes besides.

MR. B. C. VICKERY (I.C.I., Ltd., Akers Division) asked if the code were made up of both definite and broad terms.

SPEAKER replied that he uses both broad and relatively specific descriptors. Referring back to the preceding question, he said that a well-chosen broad descriptor could have taken care of the 'tumours' problem. Ordinarily there is complete freedom to use broad or narrow descriptors in any combination. Sometimes a limited number of descriptors are 'linked' into a fixed hierarchical order by the way the code patterns are set up. For example, in one installation, when the descriptor 'mittens' is coded, the code pattern for 'clothing' goes on the card automatically at the same time. The same is true for 'boots'. When a search is prescribed by 'clothing', all the linked items emerge, whether or not anyone thought to put in the code for 'clothing'. The speaker emphasized that although these linkages are easy to set up, and often appeared attractive, closer study at almost all of the installations had resulted in the decision not to use the technique. The technique has a clumsiness, it is a source of confusion, and these factors outweighed the advantages.

MR. J. BIRD (Aslib) asked if the technique was the same as Perry's 'abstraction ladders'.

SPEAKER said no, and added he did not think abstraction ladders would be useful in retrieval. He said that Perry's abstraction ladders are almost identical in concept with the system of the Universal Decimal Classification, with the one using letters and the other numbers in its symbolism. So far, there has been no announcement of an installation successfully using Perry's system. A coding system similar in principle, based upon the U.D.C., was tried at Harwell, and the record shows it contributed to the failure of that experiment. An abstraction ladder may have as many as a dozen levels, counting upwards. The linkages tried by the speaker had only two levels, and three levels would be his limit.

Abstraction ladders present some perplexing problems that have yet to be elucidated. Before one can begin coding at all, every kind of object in the universe must be fitted into some ladder, and the listing of all these ladders seems to be an endless task. There have been no suggestions as to how this can be made into a finite operation. Another problem is that 'dog' is both a 'mammal' and a 'friend of man'. This puts him into two different abstraction ladders and gives him two different letter codes. Any other object can similarly be put into a large (and unlimited) number of different abstraction ladders, and would thus have as many different codes.

DR. D. J. CAMPBELL said that he considered two or three levels to be inadequate.

SPEAKER said he thought he agreed with Dr. Campbell. One must distinguish between the need to have different levels of generality and precision as Dr. Campbell probably had in mind and the number of levels in an internally linked coding system. With

descriptors there may be no levels of internal linkage in the coding. Yet the selective prescriptions 'A', 'AB', 'ABC' and 'ABCD' give four different levels of precision, obtained merely by conjoining descriptors. Since a typical card may have a dozen descriptors, in this sense a dozen levels of precision are available.

MR. A. H. HOLLOWAY (Ministry of Supply, TPA.3/TIB) asked (a) whether the speaker had any proposals for extending his system for use with larger collections, and (b) was he right that, in a collection of about 2,000 documents which were assigned the speaker's average of five and a half descriptors apiece, about half the cards selected in a search would be 'false drops'.

SPEAKER in answer to (a) said that the card sorter shown would not extend conveniently to collections larger than about 25,000 or 50,000 items, but that machines suitable for larger collections seemed probable (part of the discussion of such machines was included in the paper as revised for publication). In answer to question (b) the number of extra cards has no relation to the number of cards coming out with the desired content. In searching with two descriptors upon a pack of cards in which half the sites bear notches, about 1 card in 256 of those sorted will appear as an extra. For a pack of 2,000 such cards there will be approximately  $2,000/256 = 8$  extra cards. There will be approximately eight extras whether we search upon an impossible prescription like 'flying' 'horses' having no desired cards, or we search upon a prescription like 'brown' 'horses' which might produce twenty desired cards. Also, if the cards have an average of only five and a half descriptors, rather than their full capacity of thirteen, then less than half of the cards' sites will be notched, and the number of extras will drop sharply.

MR. C. W. CLEVERDON (College of Aeronautics) said that it had been interesting to learn that organizations were willing to spend up to \$5 per document on indexing. It bore out what the Aslib Aeronautical Group had been saying, namely that some types of material should be indexed more intensively than others, and it was probable that from an economic aspect it would be better to use two systems. He asked if any research had been done on comparing the costs of indexing to the efficiency in different systems.

SPEAKER said that such cost figures came from estimates of yearly cost divided by the number of cards prepared per year. The costs would be somewhat lower if special abstracts were not put on the cards. In the opinion of the speaker, so long as the same level of analysis was used, the cost of maintaining the system would be about the same whatever system was used. To cut costs, you must cut down on the time spent by analysts in reading; they cannot go into the documents so deeply. A less detailed analysis might make appropriate a simpler retrieval system, such as an ordinary vertical filing system. On the other hand, the various systems differ greatly in the efficiency and reliability with which they perform information retrieval. One measure that the speaker has used to measure information retrieval performance is the size of collection that can be thoroughly searched in thirty minutes by the particular system. For schemes now in use or proposed, the 'thirty-minute size' varies from a few hundred documents to millions of documents.

MR. R. G. THORNE (Royal Aircraft Establishment, Farnborough) asked how Mr. Mooers chose his descriptors.

SPEAKER said that getting the descriptors is an empirical process. When a Zatocoding System is to be set up at a client's company, a working panel, usually consisting of the director of research and some of the top scientific personnel, is gathered. The speaker works with them. They start with a pile of reports on top of a desk, take the first one, read the abstract and look it over, and then ask, 'Why would anyone in this company be interested in using this report?' The answer may come out that it is about *propellers*, that it is about *aerodynamics*, and that it is a *wind-tunnel study*. Each of these is written down as a presumptive descriptor. The same empirical process is followed with the next report, and so on. Sad experience, on more than one occasion, has given convincing proof that descriptors 'dreamed up' in an armchair, without reference to actual reports, are worthless. By the time that some fifty reports (selected to give a good cross-section of the company's interests) have been worked over, better than 80 per cent. of all the final descriptors have been discovered. At this stage the descriptors are written out on a large sheet. This is the rough draft of the schedule. Related descriptors are grouped in the draft. Some more reports are then studied, using the draft descriptor schedule to



analyse the reports. This adds a few more descriptors, and rough spots in the draft schedule are ironed out. During this stage scope notes are being written on how to use the new descriptors. It has been the speaker's experience that the client's people on the working panel will have put in a total of less than 150 hours from the beginning of the process until the time that the schedule is ready to hand over to the typist for final typing. The speaker follows the client's work very closely during this whole set-up period. His goal is to teach the client thoroughly, so the client will be capable of operating the system with no need of further help. During the set-up period he usually holds three sessions, each one day long, at the client's plant. The setting-up period usually lasts about two or three months. A follow-up visit is made a year later. The descriptor schedules are remarkably stable, and at the end of the first year only about ten or fifteen descriptors out of about 300 in the schedule require substantial modification. Less adjustment is required in succeeding years.

Mr. B. C. VICKERY thanked the speaker for his lucid explanation and asked how the groups of descriptors were chosen.

SPEAKER replied that grouping was merely for convenience. Appropriate descriptors were easier to find when related ones were placed near each other. When the possibility of grouping is explained to clients they enthusiastically take up the idea. They have definite ideas about the relationships between the different parts of their technology, and they group the descriptors in a fashion that will be most useful to them.

Mr. S. WHELAN (Ministry of Supply Radar Research Establishment) asked if Mr. Mooers' retrieval system could be used with I.B.M. or Hollerith cards.

SPEAKER replied that a licence to use the patented system in the United Kingdom would be needed, and that one of the more expensive of the I.B.M. machines could do the kind of selection required.

Mr. AGARD EVANS thanked the speaker.

## THE FUTURE PUBLIC TECHNOLOGICAL LIBRARY SERVICES

### SHOULD A NEW PUBLIC TECHNICAL LIBRARY SERVICE BE BASED ON THE TECHNICAL COLLEGES?\*

By D. J. URQUHART, PH.D.

*Department of Scientific and Industrial Research*

*London, 21st October, 1955*

THE topic I wish to present is not a simple one and the position is confused by discordant voices each advocating different solutions to quite different problems. I will present to you certain facts which seem to me to be important, and some possible interpretations of these facts. The whole is designed to provoke discussion of the problem of improving our technical library service. The essential thesis is that the future lines of development of public technological library services will be mainly influenced by factors outside the control of the library profession but which must be appreciated if librarians are not to act like King Canute and attempt to stop the incoming tide. Lest my reference to this anecdote should conjure up a picture of a spot on the English coast and a twice-daily tide, let me urge you to treat the world as our stage and the next few decades as our unit of time.

Against that background I want to present to you a series of propositions. The first proposition is this: *Library services are determined by social needs and not the aspirations of librarians or budgetary considerations.*

I know that whenever I attend library conferences or read about them I find that one theme dominates. It is that libraries need more money. The repetition of this theme by itself does not help its realization, for, despite the current administrative arrangements, libraries are but instruments. They exist only to meet some of the requirements of society. It is on the basis of these requirements, and these requirements only, that arguments for additional resources should be based. Let us not forget that the total expenditure on libraries in this island is a mere bagatelle compared with the total national income. In the past, libraries have been treated as cultural amenities. If it could be shown by a consideration of economic values that it was desirable to increase the present library expenditure tenfold, that could be achieved. So far the case for spending considerably more on libraries rather than other things has not been prepared, or, if it has, it has not been voiced sufficiently clearly in the right places.

It is the job of Aslib or the Library Association to present a general case for libraries. I wish only to draw your attention to some consequences of a simple fact which is well known to all of us. The fact is this. We must eat. We in this island cannot feed ourselves without importing. We live

\* The views expressed in this paper are intended to provoke discussion. They are not necessarily the views of the author's department.



## EDITOR'S CORNER

Publication in the April 1950 number of *AMERICAN DOCUMENTATION* of an article "Multiple Coding and the Rapid Selector" by Carl S. Wise and James W. Perry has prompted a spirited critique by Mr. Calvin S. Mooers, President of the Zator Company of Boston. In furtherance of the aims of *AMERICAN DOCUMENTATION* it has been decided to devote the entire space available in this issue for editorial comment to the communication. The opinions expressed are entirely those of Mr. Mooers.

### CODING, INFORMATION RETRIEVAL, AND THE RAPID SELECTOR

The stimulating article by Wise and Perry<sup>1</sup> opens the discussion of a large and important field of documentation which I personally hope will receive considerable attention in future issues of *AMERICAN DOCUMENTATION*. Pertinent recorded information cannot be used unless its very existence — and then its location — is discovered in a large collection. Thus "information retrieval," the discovery process, must be distinguished from the warehousing and reproduction aspects of documentation.

Indexing systems and classification systems are characterized by the authors as being incapable of practicable information retrieval when faced by the changing demands of future trends in viewpoint and research. With this conclusion and with the brief supporting arguments of the authors, I definitely agree. Indexing systems and classification systems are enormously defective when objectively compared to the functional requirements of efficient information retrieval. The defects are logically inherent in the two systems and no amount of ingenuity, revision, or disputation can remove them. However, these two documentary procedures have by now become so well established, and have accumulated such a weight of precedent and authority, that any brief critical assertions about their utility cannot be expected to make much of a dent in the situation. What is needed is an extended critical analysis backed by a tightly-knit line of reasoning and many solid examples. We can hope

<sup>1</sup> C. S. Wise and J. W. Perry, "Multiple Coding and the Rapid Selector," v. 1, pp. 76-83, *American Documentation* (April 1950).

that from some source — and the sooner the better — there will appear such a full and rigorous treatment of the logic of this situation.

To use the terms "polydimensional" and "dimensional" in describing the way in which a piece of information can be factored into a set of cooperative descriptive concepts is to invite future nomenclature trouble when the underlying theory of information retrieval is finally developed more completely. About the factoring process there is no question. But, unless the terms "polydimensional" and "dimensional" are used only in a figurative sense to suggest the interplay and cooperation of the descriptive factors, employment of these terms will soon lead to divergences in meaning and to the misdirection of thought. Dimensionality in the simpler cases deals with (1) the number of Cartesian coordinate axes of a space, or (2) the number of linearly-independent components of a vector, and in a very much more sophisticated sense, (3) the extension<sup>2</sup> of these concepts to the large — though limited — class of objects which make up a "separable metric space," which then includes concepts (1) and (2). In my investigations of this subject, I have found that the concept-factors cannot be given a strict dimensional interpretation compatible with the basic ideas underlying any of these three notions of dimensionality.

A severe criticism must be directed towards the proposal that the particular code described by the authors be installed in a large-scale information retrieval machine of the type of the Rapid Selector. This criticism is based upon the inefficiency of this code in the utilization of expensive machinery, upon the code's inflexibility, and upon the code's incapability of utilizing with greatest economy the coded areas on the film strip. Large-scale information retrieval installations will be expensive by any measure, and every effort must be made to secure efficiency in these directions. Economy in machine complexity leads to the additional benefit of fewer breakdowns and periods of mechanical inoperativeness. Greater efficiency in storage of selective coding on the film gives the additional benefits of less film to store and a more rapid scanning to the

<sup>2</sup> W. Hurewicz and H. Wallman, "Dimension Theory," Princeton University Press, Princeton (1948).



end of the shorter film or, alternatively, of much more selective information stored on the same length of film.

Only one aspect of inflexibility will be mentioned here. In a coding system depending in its selection upon the operation of statistics, as that described by the authors does, it should be possible to vary the number of marks in certain of the code patterns. That is, in the authors' language, it should be possible to use two-letter, four-letter, or even fifteen-letter codes for certain of the ideas. Only in this way will the system be versatile enough to meet the varied intellectual-statistical demands of the operation of a large information collection. These matters have been previously treated elsewhere.<sup>3</sup>

Non-economical use of the coded areas of the film strip leads to other — and equally serious — sources of inefficiency. The authors' coding method forms the code patterns by spelling out the initials or fragments of words from a statement of a subject. Thus, "resins" receives the code "RESI," etc. The letters employed in such a code do not appear with equal frequencies, since such letters as K and Y are seldom used in the language, while E or S appears very often. What happens then in the coded area of the film (or in any other selective card or device) is that after only a few codes have been put into the coding field, the positions for E and S will most surely be marked. On the other hand, even after a full set of descriptive factors have been coded into the area, the positions such as K and Y will be marked in only a very few cases. Statistically, it is almost always possible to predict that the positions corresponding to those letters of the first kind will be marked, and that those of the second kind will be unmarked — irrespective of the subjects coded. Therefore, the coding positions corresponding to the letters of both high frequency and low frequency will really have a very low utility in actual selection. This is because we know a-priori — independent of the subject — how they will be marked. There would be little loss if these letters were left out altogether.

Efficiency in the utilization of the coding area demands that code-mark frequencies be equalized.

<sup>3</sup> C. N. Mooers, "Zatocoding for Punched Cards," Zator Technical Bulletin No. 30, Zator Company, Boston (1950).

I first stated this requirement several years ago in a discussion of coding principles.<sup>4</sup> Violation of the principle can lead to serious inefficiencies with large collections and when a large number of codes must be used to describe each item of recorded information, as in the Rapid Selector. The effect is particularly critical when a large number of codes are recorded in a single field. While the effects of a violation may be tolerable in a small collection (1,000 items) of hand-sorted cards, and when no heavy demands are placed upon the coding capabilities, this "statistical" tolerance does not extend to the large, heavily-coded collection. There the non-uniformity of frequencies definitely introduces a serious waste of coding capability.

Because of these non-uniformities in frequency, which apparently are not taken into account by the authors' computations, their stated values for the occurrence of extra or unwanted selections are in considerable error. In what follows, these non-uniform frequencies will be included in the computation for the rate of occurrence of the extra selections, and comparisons showing the impairment of efficiencies will be made.

By the Wise and Perry method for Rapid Selector coding, the coding field attached to one frame of film is divided into six subfields each containing 26 positions, one position for each letter of the alphabet. We shall now focus our attention upon one subfield and consider the statistical probabilities for the different positions being marked when there are one, two, or in general  $n$  different codes added to the field. Let  $p(x,n)$  symbolize the probability of the letter position designated by  $x$  being marked when there are  $n$  codes in the field. When  $n = 1$ , there is only one code, and thus one mark in the subfield in a single letter position. For different single codes, this mark will appear at different positions, and thus  $p(x,1)$  represents the letter frequencies for all the letter-codes.

The statistics of the selection process depends upon  $p(x,n)$  which is easily computed from  $p(x,1)$ . Since  $p(x,1)$  is the probability of the position  $x$  having a mark when  $n = 1$ ; then  $1 - p(x,1)$  is the probability of this position being unmarked. For

<sup>4</sup> C. N. Mooers, "Putting Probability to Work in Coding Punched Cards," American Chemical Society, abstracts of papers, 112th meeting in New York City, September 1947, p. 14E.



two codes  $(1 - p(x,1))^2$  is the probability of position  $x$  being unmarked (since the occurrences are independent), and so on. Thus

$$1 - p(x,n) = (1 - p(x,1))^n, \text{ or} \\ p(x,n) = 1 - (1 - p(x,1))^n$$

Consider now the statistical occurrence of extra selections (in the one subfield) with respect to a large number of typical codes, with the letter frequencies given again by  $p(x,1)$ . We want to average the probabilities over all the possible letters in order to find a weighted probability depending jointly upon the marks in the subfield described by  $p(x,n)$  and upon the selection code probabilities described by  $p(x,1)$ . The weighted probability for the one subfield comes out as:

$$\frac{\sum_{x=1}^{26} p(x,1) p(x,n)}{\sum_{x=1}^{26} p(x,1)} = Q$$

With selection based upon six subfields, the weighted probability for the occurrence of extra selections is simply  $Q^6$ , assuming selection by a single six-letter code. If it is a two- or three-code selection, with no repetition of code letters, then the ratio of extra selections will be respectively  $Q^{12}$  and  $Q^{18}$ .

Thus, everything is seen to depend upon  $p(x,1)$ . The values of these letter frequencies vary from language to language, are different when counted for the first, second, third, etc. letters of a word, and so on. The mode of making divisions between words, as illustrated by the codes shown in the article, will also make some difference in the statistics. Some practical compromise on a letter frequency distribution must be assumed. As a source of the frequencies, we will use the list compiled by Fletcher Pratt for English text,<sup>5</sup> with an adjustment made for the frequency shifts due to the elimination of the article "the." The Wise and Perry method has a high preponderance of initials, but the same source indicates that initials have an even greater lack of uniformity in frequency than text. Therefore, by taking the text frequencies we should have a conservative estimate of the losses in coding efficiency.

<sup>5</sup> F. Pratt, "Secret and Urgent, the Story of Codes and Ciphers," p. 252, Blue Ribbon Books, Garden City, N. Y. (1939).

For simplicity in listing and computation, the letters are grouped by mean frequency, as follows:

Letter	Frequency ( $p(x,1)$ )
E	0.142
A, O, N, R, I, S, T	.076
D, L, F, C, M, U, H	.030
G, Y, P, W, B	.019
V, K, X, J, Q, Z	.003

We note that if the frequencies of the letters were uniform, each would have the frequency 0.038.

The computation is now straightforward and merely involves the substitution of these numbers for  $p(x,1)$  into the formula for  $p(x,n)$  and  $Q$  for a given choice of the value  $n$ . It is then found that  $Q = 0.619$  when  $n = 16$  as suggested by the authors. In comparison, if the frequencies had been equally distributed, we would have had  $Q = 0.466$ , or less than  $1/2$ . Because the ratio of extra selections depends upon a very high power of  $Q$ , these differences in the magnitude of  $Q$  — which may seem small at this point — become very significant when we take the sixth, twelfth, or eighteenth power.

By computing the powers  $Q^6$ ,  $Q^{12}$ , and  $Q^{18}$  we arrive at the corrected values for the ratios of extra selections, and these can be compared with those stated by the authors and which seemingly are based upon the erroneous presumption that non-uniformities in frequency can be neglected:

	Ratios given by Wise and Perry	Actual ratios found by considering frequencies
One-code selection less than 1/100		1/18
Two-code selection less than 1/10,000		1/350
Three-code selection less than 1/1,000,000		1/6,000

The promise is far short of the performance that can be statistically expected. By the corrected ratios, a three-code selection upon the Library of Congress collection (assumed here to have five million items) would result in over 800 unwanted selections, that is, information items having nothing to do with the subjects desired, instead of the more reasonable five extra selections that might be expected from the authors' original figures.



The actual loss in coding efficiency due to non-uniform frequencies, and incurred through the use of the authors' method, can be objectively measured. We will make the comparison in efficiency by considering the situation in which the selective concepts are recoded in such a way as to achieve a complete uniformity of frequencies, that is, by the use of random patterns of marks for the codes. This is the method of Zatocoding, and is described elsewhere.<sup>6</sup> In the article the effective coding field was taken as 6x26 positions, or 156 positions. Here, we will let the size of the coding field be the unknown variable, and we will let  $n = 16$ , and will let the ratio of extras for a single code selection be  $1/18$  as found above. It is then found that the authors' performance could be duplicated in a coding field having only 97 positions, instead of 156 positions.

Therefore, neglect of the frequency distributions of the letters used in the codes results in an inefficiency measured by a 61 per cent increase in the required coding field space, and simultaneously an equal increase either in machine complexity or in running time during the scanning operation.

In considering a type of coding based upon words and fragments of words, as in the Wise and Perry coding, there is one more deleterious effect that is of first magnitude. In the study above, we have seen that single-letter frequencies cannot be neglected in arriving at an estimate of the occurrence of extra selections. Neither can the letter-pattern frequencies be neglected. Otherwise flagrant situations like the following will occur (where we form the codes with the first three letters from each stem):<sup>7</sup>

Word	Six-letter code
radar, automatic tracking	R A D A U T
radioactivity	R A D A C T
radio antenna	R A D A N T
radio arsenic	R A D A R S
radioautography	R A D A U T

Here are distinctive subjects taken from three different fields, yet in one case the codes are identical and in the other cases there are only one or two (rather than six) letters of difference in the codes

<sup>6</sup> C. N. Mooers, ref. 3.

<sup>7</sup> Note that by forming the codes from the first six letters in series, or from the first four and two letters of the words, we find no relief from this very high similarity of codes.

to allow them to be distinguished. The statistics of the situation shows<sup>8</sup> that if "radar, automatic tracking" is the subject of selection, then approximately one-half of all the references on "radioactivity" will also appear as unwanted extra selections due to the high correlation (all but one letter) of these codes.

This situation prevails in a more obnoxious form within the same subject field. Consider an attempt to make a selection upon "radioautography" by the use of "radio phosphorus," where we intentionally want to exclude such items as "radioautography" by "radio arsenic." In spite of our use of a dual-statement prescription, such an exclusion depends only upon two letter positions, or a probability of one-quarter. Total elimination of correlations, giving six code letters of difference, gives the powerful discrimination of  $1/128$  between these two subjects.<sup>9</sup>

Besides confusing similarly spelled ideas, such correlations result in a very low number of effective selective positions and a correspondingly high ratio of extra selections when measured against the whole collection. Thus, a selection upon "radio autography" with "radio arsenic" has (because of duplications) only eight different selective positions operating (instead of twelve), and therefore gives a ratio of extra selections of  $Q^8 = 1/45$ . By eliminating correlations, as is done by Zatocoding,<sup>10</sup> there are eleven to twelve selective positions operating, and the ratio of extra selections becomes  $(1/2)^{11} = 1/2,000$ , when measured against the whole collections.

Let us see what can be done in a specific instance by using the full field of 12x18 positions that is available in the Rapid Selector and by applying it to a collection of five million items (e.g. the Library of Congress). The Rapid Selector field has a total of 216 positions available for coding. Using Zatocoding, with its high coding efficiency, we first set the absolute number of extras to be expected in scanning the five million items with a three-code prescription to a value less than unity (actually less than  $1/3$  in this case). Then, from the formulas given elsewhere<sup>11</sup> we find that a single code should

<sup>8</sup> C. N. Mooers, ref. 3.

<sup>9</sup> C. N. Mooers, ref. 3.

<sup>10</sup> C. N. Mooers, ref. 3.

<sup>11</sup> C. N. Mooers, ref. 3.



have a random pattern consisting of eight marks in the field. The field can accommodate eighteen such patterns, and therefore it can code eighteen descriptive aspects for each information item. The number of available code patterns or designations comes to about  $10^{14}$  or a hundred million million.

The debilities and criticisms cited before all disappear by this method. Using standard procedures, the number of marks in the code patterns can be varied with ease when that is statistically desirable. The spotty use of the coding field is eliminated by the intentional use of uniform frequencies in the codes. Correlations between codes, and their undesirable consequences, also disappear. The ratio of extra selections is at a true minimum value consistent with the amount of information being coded into the field. One-, two-, and three-code selections now have an actual ratio for extras of less than 1/500, 1/50,000, and 1/5,000,000, respectively. In short, figures such as these are the true measure of the performance that can be expected with the Rapid Selector using an efficient coding.

We can conclude that a revision of the coding for the Rapid Selector will surely produce advantages, since the present system allows selection only by a single statement, and the coded field records only six different statements. Nevertheless, proposals for a coding system for a large-scale selective

device should be advanced with caution, particularly when the coding system depends upon statistics for certain features of its operation. Extrapolation from a small hand-sorted punch card collection is misleading if it is not accompanied by a rigorous mathematical or statistical study of the factors involved and their consequences. The fundamental principles that govern the validity of superimposed coding — what the authors call "multiple coding" — have been available for several years. Their extension to very large collections has been studied in detail,<sup>12</sup> and these theoretical predictions have now been justified by experience in collections running to as large as 30,000 cards. Superimposed coding, particularly in the refinement known as Zatocoding (as in the last example), is the most efficient coding method known at the present time for the selection of information.<sup>13</sup> It can be expected to be a most important factor in the future design and performance of large-scale information selection instruments.

CALVIN N. MOOERS

<sup>12</sup> C. N. Mooers, "The Application of Random Codes to the Gathering of Statistical Information," Zator Technical Bulletin No. 31, Zator Company, Boston, 1949. Based upon an M. I. T. thesis (math.), January 1948.

<sup>13</sup> C. N. Mooers, "The Theory of Digital Handling of Non-Numerical Information and its Implications to Machine Economics." Paper presented before the Association for Computing Machinery, March 1950. Zator Technical Bulletin No. 48.



**Superimposed Coding**  
**With The Aid Of Randomizing Squares**  
**For Use In Mechanical Information**  
**Searching Systems**

H. P. LUHN

*Chapt. 23 in Randomized Coding, Copyright*

**IBM**

PRODUCT DEVELOPMENT LABORATORY

INTERNATIONAL BUSINESS MACHINES CORPORATION  
POUGHKEEPSIE, NEW YORK



## TABLE OF CONTENTS

Introduction .....	1
A Review of the Reasons For Coding .....	3
Letter Codes .....	3
Word Codes .....	4
Cryptographs .....	5
A New Scheme of Superimposed Coding .....	5
The Construction of the Code .....	6
Special Consonant Code.....	9
Significant Letter Spelling .....	10
ELCO (Eliminate and Count) Word Code.....	11
Self-Demarcating Word Code.....	11
The Design of an 8 x 8 Randomizing Square for Letters Only....	12
The Design of a 10 x 6 Randomizing Square for Mixed Symbols..	19
Checking Schemes .....	23
Single Row Recording on Punched Cards .....	23
Recording on Tapes .....	25
Conclusion .....	25



SUPERIMPOSED CODING WITH THE AID OF RANDOMIZING  
SQUARES FOR USE IN MECHANICAL INFORMATION  
SEARCHING SYSTEMS

by

H. P. Luhn

ABSTRACT

Superimposed coding is the technique of recording a plurality of units of information in a common coding field for making them available for simultaneous analysis. This technique produces secondary code combinations which, while unintended, might nevertheless represent valid codes. Such occurrences are minimized if the code combinations used have, as a set, the property of randomness. The encoding method described in this paper brings about this property by a standard operation. This eliminates the necessity of arbitrarily assigning random numbers to given terms and of maintaining a dictionary of such assignments. With the new methods, the randomized designation becomes a function of the original term and may therefore uniquely be derived when or wherever required. The paper also contains a description of the ELCO code, another form of encoding derived by standard procedures rather than by assignment.

The original manuscript for this report was dated June 15, 1956.



# SUPERIMPOSED CODING WITH THE AID OF RANDOMIZING SQUARES FOR USE IN MECHANICAL INFORMATION SEARCHING SYSTEMS

by H. P. Luhn

## INTRODUCTION

The problem of how to most effectively place information on records is important in any mechanical process used to scan records in order to take desired information from them. In business records, a limited number of classes of information terms are used. Because of the limited number of these classes, it has been the custom to insert the information in fixed areas or fields on the records. With the use of punched cards, this system of fixed fields became particularly significant. This was because once a card reading machine was adjusted for a given record format, there was no question as to the meaning of the information recorded in it.

The type of information discussed here is of a more general character. Great difficulty is encountered in assigning fields to the many classes of information terms that might possibly occur. The number of such classes may vary from record to record and there may be more than one term that could be assigned to a given class. Obviously, if allowance was made for a maximum of such variations, the number of fields would be so numerous that the record form would assume impractical dimensions.

One way to overcome this obstacle is to abandon the concept of fixed fields and to record information in serial form, separating and identifying classes of terms by special division marks. This method is being used in column-by-column card scanning systems\* and in systems where continuous tapes serve as the recording means.

---

\*H. P. Luhn, "The IBM Electronic Information Searching System"  
International Business Machines Corporation, Engineering  
Laboratories, Poughkeepsie, N. Y., 1952



In cases where recordings of the above type are not feasible, two systems have been developed to bring about the effect of serial scanning. The first system uses punched cards in conventional card processing devices. It consists of duplicating a given record as many times as there are terms to be scanned and in such a way that each of the resulting cards has a different one of the terms in a fixed field. This process is sometimes referred to as 'field rotation' and after reading the fixed field of all of the cards in the set, all the information contained in the record has been scanned\*.

The second system is designed to give similar results with the use of a single card. It is referred to as 'Superimposed Coding' and consists of recording a plurality of information terms, one over the other, into one common field. An example of this type of recording is given in Fig. 1 where three code numbers have been recorded in a single four column field of an IBM card. The original numbers were 1576, 2419 and 8079. However, the ten resulting punched holes may be interpreted as standing for many other numbers such as 1579, 2516, 8476, etc. The merging of such codes produces secondary combinations which might represent unintended, yet valid, terms contained in a given dictionary of coded items. It is therefore necessary to provide a means which will minimize the interference caused by such spurious information.

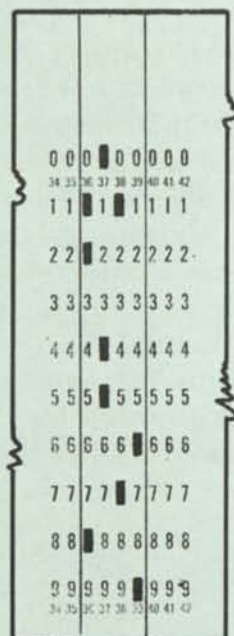


FIG. 1

\*An example is: The Punched Card System of the Chemical-Biological Coordination Center, National Research Council, Washington, D. C.



In solving this problem, reliance is made on the random way in which letters or numbers happen to be combined to constitute a term and on the random way in which such terms are being used. If this random nature is substantially absent among a set of given terms, a recoding method can be reverted to. One such method consists of substituting random numbers for the given terms\*.

It is the objective of the system described below to achieve the effect of randomizing non-random terms without the device of recoding.

### A REVIEW OF THE REASONS FOR CODING

Before going into the details of the proposed codes, it is necessary to review some of the reasons why codes are used for recording information.

The word 'code' is rather loosely used as a collective name for three distinct types of writing. They are:

Letter Codes. Communication by means other than the spoken language has been made possible by the invention of writing. This can be done by means of idiographs or by phonetic spelling with the aid of letters. These methods of writing rely on two-dimensional representations or symbols of varying forms used to designate the ideas or sounds for which they stand. When it became desirable to communicate by means other than speaking or writing, a severe limitation was imposed by the medium through which such communication was to be established. Whether this medium was the drum, or smoke, or a pair of electric wires, only a limited number of different signals could be produced and a substantially one-dimensional system of writing had to be developed. Examples are the Morse Code and the Teletype Code. These codes serve to represent letters and numbers as a series of unmodulated signals varying in length, in spacing, or in both.

It is this type of letter code which has made possible the manipulation of numeric and alphabetic information by machine. Such codes may take the form of holes punched in a card or tape, or of magnetic marks

---

\*Claire K. Schultz and Robert T. Ford, "Random Coding for Recording and Searching Literature by Means of Punched Cards", Research Lab., Sharp & Dohme Division, Merck & Co., Inc., West Point, Pa.



on a magnetic tape, or of the mechanical or electrical signals actuating the various devices which make up a machine. The only purpose of the letter code in connection with machines is to facilitate internal communication. In most cases the use of such a code is not apparent since manual input and readable output are accomplished by the use of conventional letters and numbers.

Word Codes. The second type of code involves substituting another word, a sequence of numerals, a letter, or some other form of notation for a word or several words. This also includes the assignment of serial numbers systematically or arbitrarily to things or persons. One advantage of these substitutions is the reduction of writing space or time. This is exemplified by the use of abbreviations such as M for male and F for female. In communication, the high cost of sending cables overseas has resulted in the compilation of code books which permit the reduction of lengthy messages into a few five-letter words. The Bentley's Code and the ABC Code are examples of this.

The space problem is particularly acute in punched cards where it is desirable to pack a maximum of information into a limited space. The most compact arrangement is the assignment of a single element in a time sequence or a single position in a two-dimensional recording array to a given information term. Such systems are in wide use for the recording of statistical information. The records of the Census Bureau and the 'peek-a-boo' card system of W. E. Batten are typical examples.

Numbers are substituted for alphabetic information for several reasons. Assigning telephone numbers facilitated the layout and location of the plug holes on the telephone switchboard and the eventual organization of automatic switching devices. Early punched-card-operated office machines were limited to numerical information only. Because of this limitation, customer's names and other alphabetical information had to be translated into numerical designations. Numbers could be assigned in a systematic fashion to include information such as: 'type of customer,' and, 'district where located'. The use of numbers to classify a given thing is well exemplified by the Dewey Decimal System. The assignment of part numbers is another good example.

Word codes may be defined as external codes with respect to machine operations. They replace one word for another to reduce space requirements, facilitate mechanical manipulation, and to enable systematic organization of subject matter.



Cryptographs. The third type of code is used to conceal the subject matter. While it is true that word codes usually cannot be deciphered without reference to a dictionary or directory, this inconvenience is the price that must be paid for the advantage derived. However, in Cryptography the sole purpose is to prevent deciphering by unauthorized persons. This type of code has no usefulness within the field of information searching in general.

### A NEW SCHEME OF SUPERIMPOSED CODING

The superimposed coding scheme described here is based on word coding to the extent that a pair of letters is recorded as a single mark within a two-dimensional recording area. In adapting it to punched card operations, the peculiarities of punched card equipment have been taken into consideration. The most important fact is that most standard machines are designed to read cards in a parallel fashion. In the case of an IBM card, all of its 80 columns are read simultaneously, that is, in parallel. The individual marks within the 12 possible positions in each column are read serially on a differential time basis. Thus, a given hole is identified by its column and by the instant in time at which it passes the reading elements of the machine.

Superimposed coding schemes rely on this two-dimensional arrangement of recording and are identified as the intersections of columns and rows within a field of a fixed size. When information recorded in this fashion has to be read for the purpose of scanning, the process of comparison or matching has to be done in this parallel and serial fashion. This process is wasteful in time and equipment as well as in recording space utilization. The new scheme proposes to rearrange the contents of a two-dimensional field and to record it in one-dimensional form within a single row across a card. The result of this is that the equivalent of 12 such fields may be read consecutively with the passage of a single card. The advantages of this approach will be described.

In recording information for searching purposes, it is desirable to signify the relationship of the various information elements. Records enumerating many things and their characteristics should reflect which characteristics refer to which thing. Superimposed coding does not permit the expression of such relations and differentiations in a single field. The remedy might consist of using a plurality of fields, each field containing information elements that are directly related. However, the provision of several fields side by side on a single card would defeat the



basic simplicity strived for and would require special equipment within the machine for multiplexing the process of scanning. The use of as many cards as there are fields required would increase the size of the files, and the searching time, and would have many other drawbacks.

In the art of information searching the terms 'words' and 'sentences' are often used to express the relationship between the various information elements. Words within a sentence express a closer degree of relationship to each other than words in different sentences. The information elements punched into a given field may therefore be referred to as the 'words' and the total of these words within a field may be referred to as a 'sentence'. In using the linear form of superimposed coding as proposed in the new scheme, as many as 12 sentences may be punched on a single record card. Each of these sentences is read in a parallel fashion for a single-cycle comparison or matching operation. If desired, several sentences in sequence may be tied together by special marks to form the equivalent of 'paragraphs'.

#### THE CONSTRUCTION OF THE CODE

Superimposed coding produces a certain amount of unwanted, though valid combinations or words. When constructing such codes, special attention is directed toward minimizing such spurious words or at least minimizing the effects caused by their presence. The quality of resolution of such a scheme depends on a number of variables such as:

1. Size of collection of records.
2. Size of the recording field.
3. Number of recording fields.
4. Number of marks used per word.
5. Number of words constituting the dictionary.
6. Number of words entered into a field.
7. Number of words to be matched.
8. Randomness of the letters or numbers employed in the words.

The statistical aspects of superimposed coding schemes have been investigated and described by C. S. Wise and others\*. In dimensioning

---

\*"Punched Cards", A Collection of Articles by Various Authors, Edited by R. S. Casey and J. W. Perry. Reinhold Publishing Co., New York, 1951.



and constructing the new scheme, proper recognition has been given to the findings of the above authors. It should be realized, however, that at this time there is not available any statistical information derived from actual applications which might confirm the theoretically derived values for spurious matches.

The system's degree of tolerance for unwanted selections may differ with the field of application. Whatever the degree of these values, the system is failsafe in that it produces at least all of the matches asked for. There is another feature which must be tolerated in return for the compactness derived from superimposed coding. Once information has been encoded and superimposed within a field, there is no obvious way of decoding the scrambled marks back into the words originally encoded. Therefore it is necessary to list these words in a more conventional manner on the respective coded records or to maintain a master file which can be referred to by way of a reference number when it is desired to identify the words actually encoded.

Word coding is being employed in most cases to overcome restrictions and difficulties imposed by information-handling facilities. The process of encoding and decoding words with the aid of code books or dictionaries is time consuming. Any scheme which will simplify this task will therefore be a desirable improvement. If, for instance, words could be spelled out directly and without the aid of a code book, a great deal of time and effort could be saved. The new scheme pays particular attention to this phase, and a method has been derived which makes such procedures reasonably feasible.

Because of this desirable aspect of a system, the new encoding scheme will be developed by applying it first to words in their original spelling. Let us assume a square or matrix having 26 rows and 26 columns. By writing the 26 letters of the English alphabet along both coordinates, all two-letter permutations of the alphabet are designated by the 676 intersections. Now considering the horizontal axis to represent the first letter of a pair and the vertical axis the second letter of a pair we may then spell out any word in the following manner: Let us take the word CHESTER. The first pair of letters is CH and this is represented by a mark at the intersection of the C row and the H column. Subsequent letter pairs of the word can be marked similarly at the appropriate intersection of the matrix. If an odd letter remains at the end of the word, it can be paired with a blind symbol represented by a 27th column. The result would be the entry of four marks into the matrix.



The new method does not operate in this manner. It spells progressive pairs in single letter steps. According to this procedure, the word CHESTER would be spelled in seven pairs, as CH-HE-ES-ST-TE-ER. In this manner an interlinked chain is formed so the second letter of each pair is also the first letter of the next pair. This interlinking may be carried one step further by bringing the end around to close the chain on itself. This is done by considering the last letter and the first letter of a word as the final pair. Therefore, the complete spelling of the above word is: CH-HE-ES-ST-TE-ER-RC, and the number of resulting marks equals the number of letters of the word.

The result of this method of chain and ring spelling is that the sequence of letters has been established as a closed system. Therefore, any mark outside this system or ring must belong to another word. The end-around spelling of the last and first letter, the pair RC, prevents an accidental match with a portion of a word like ROCHESTER, which would differ in the end-around spelling of RR. By the same token, a word like CHEST would not match with portions of either of the two previous words because of the spelling of its end around pair as TC. The occurrence of the word CHEST in both CHESTER and ROCHESTER may be ascertained by searching for this word minus the end-around link TC. It is apparent that if the words had been spelled in the form of unrelated pairs of letters, such differentiations would not have been possible.

Because the words were spelled as a ring, there might be a question as to where a word begins. If it is important to indicate this, the addition of an extra letter such as Q at the end or the beginning of the word would serve to mark the break in the ring. Another way of marking the end of a word would be to omit the end-around spelling and pair the last letter with an 'End' symbol represented by a 27th column.

Additional words may be spelled similarly and added in this square. Intersections of the resultant chains would remove the possibility of unique interpretations of the marks. These points of confusion are less likely to arise than in a system where the various marks are entirely unrelated. While the 26 by 26 square might be desirable from a safety point of view, its size is impractical and actually wasteful. Usually the size of a square may be reduced substantially without seriously impairing its usefulness because of the following considerations.

The statistical rate of usage of the letters of the alphabet in spelling words varies considerably. Also, certain letter combinations may never occur. Therefore the 26 by 26 square may contain intersections which will never be used. Some will be used rarely and others will be



used most of the time. Therefore, it is an objective of design to arrive at a scheme where the probability of a mark appearing in any one of the fields is reasonably even. Replacing words by random numbers would accomplish this but this type of re-coding is the very thing that the new system was devised to avoid.

Randomizing the marks is accomplished instead by reducing the size of the square and by assigning several letters to each of the rows and columns. The letters are then grouped in such a fashion that the combined averages of usage for each row or column are as closely even as conditions permit.

While squares of varying sizes may be constructed, consideration was given to the ultimate intended use of the system: recording the contents of a square in a single row of an 80-column IBM card. The largest square that could be accommodated is an 8 by 8 square which requires 64 positions across the card and leaves 16 columns for recording serial numbers and other information.

Although the spelling of conventional words is possible with the new method, it is essential that it be equally adaptable to more compact and less redundant schemes of spelling. It should also be possible to encode combinations of numerals such as serial numbers and numeric codes.

Attention was given to the following alphabetic spelling schemes:

1. Conventional Spelling
2. Special Consonant Code
3. Significant Letter Spelling
4. ELCO Code
5. Self-Demarcating Word Code

Conventional Spelling requires no explanation but the other schemes will be described before going into the procedure of arriving at randomizing squares.

Special Consonant Code. This code is used to provide a simple, systematic method for deriving code words from the original words. It is a variation of the conventional consonant code which normalizes words by deleting all vowels, the letters W, H, and Y, and the duplicate in double letters. However, in order to keep words from being deleted by this process, as in the case of the word 'WAY', the following procedure is proposed:



'Starting from the right, strike out all duplicates of double letters and the U of QU. Then starting from the right again, strike out vowels and the letters W, H, and Y, but stop short if this process reaches a remainder of 3 letters. (OLIVE is reduced to 'OLV', WAY remains 'WAY').

The consonant code is particularly useful for encoding proper names because it overcomes many of the variations which such names are subjected to. Additional conventions may be introduced to handle the usage of K, CK, C, of TS, TZ, Z and other common variations.

Significant Letter Spelling. This encoding scheme involves a process of abbreviating common words or names to a fixed minimum.\* The theory behind this scheme is that less frequently used letters provide greater differentiation among abbreviations. Words are systematically reduced by eliminating letters in accordance with a letter use frequency table. Of the letters in a word, the one having the highest frequency ranking is dropped first, the one next in frequency is dropped next, and so on, until the fixed minimum of remaining letters has been reached. Double letters are treated as single letters and the U of QU is disregarded. Among similar letters the last one is dropped first.

A sequence of 4 letters appears to be sufficient for the average application. The first letter of a word is retained as being significant because of its position and is excluded from the reduction process. This may aid in identifying and indexing the abbreviations.

To a degree, this process results in a more even usage of the letters of the alphabet and therefore promotes random distribution of superimposed marks.

A frequency scale which might be used for the reduction process is that compiled by R. T. Griffith and published in the "Journal of Franklin Institute." The scale is as follows:

E	T	A	O	N	I	S	R	H	L	D	C	U	M	F	Y	W	G	P	K	B	V	X	J	Q	Z
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

Examples: APPARATUS = APRU

SEQUENCE = SQNC

---

\*Carl A. Cline, "An Edge-Notched Index Card System for Mechanical Sorting", Paper presented to the Div. of Chem. Literature, American Chemical Society, New York, September, 1954.



ELCO (Eliminate and Count) Word Code. This code is here proposed to furnish an added degree of differentiating over the Significant Letter Spelling just described. The procedure for deriving an ELCO code word is as follows and is applied only to words of more than 4 letters:

'Starting from the right strike out all duplicates of original double letters and the U of QU. In doing so, write over each of the letters, thus eliminated, the number assigned to it in the letter frequency scale. If more than 3 letters remain, strike out additional letters, with the exception of the word-starting letter, in descending order of frequency ranking as given by the letter scale. Again write the scale number over each of the letters stricken out. With similar letters, strike out the one farthest to the right. When the word has been reduced to 3 letters, add up the values of the letters eliminated. If the total is over 26, deduct 25 as often as necessary. On the frequency scale find the letter which corresponds to the value of the derived total. This letter becomes the 4th letter of the ELCO word, the first letter being the starting letter of the original word and the 2nd and 3rd letters being the most significant of the remaining letters of the word. In case the original word has 3 or less letters, add the letter Z, standing for 'zero', on the right to bring the word to 4 letters.'

The following examples have been derived with the aid of the Franklin Institute scale previously given and numbered for the computation of ELCO code words:

		<u>ELCO</u>	<u>Significant Letter Spelling For Purposes of Comparison</u>
3 5 1		9=H	
CANCER	=	CCRH	CNCR
CONCERN	=	CCRF	CNCR
CONCERT	=	CCRC	CNCR
CONCRETE	=	CCRU	CNCR
PATENT	=	PANN	PATN
PATENTEE	=	PANS	PATN
FAT	=	FATZ	

Self-Demarcating Word Code. These code words are made up of sequences of consonants and vowels in such a manner that several of them may be written side by side without separating them by special marks. All of these code words begin and end with certain consonants. Three-letter words have a vowel in the middle while four-letter words have either two



vowels or one vowel and the letter L or R. There are some exceptions to these rules\*. This systematic spelling of code words is particularly suitable for the new system and contributes to the reduction of spurious matches.

#### THE DESIGN OF AN 8 x 8 RANDOMIZING SQUARE FOR LETTERS ONLY

The five methods of coding described here have been made the basis for the manner in which the letters of the alphabet have been grouped in the indexes of rows and columns of the encoding matrix. The problem was to come up with an arrangement that would achieve a comparable degree of random distribution for all of the five modes of spelling.

Because in an eight by eight square at least three letters must be assigned to each row or column, precautions were taken to insure differentiation of spelling among the three letters of each group. If the same groupings were applied to the vertical as well as the horizontal index, the letters of a group would be treated in identical fashion. Therefore, a given chain would represent all of the words that could be derived by permuting the various letters of all the groups in the chain. This would also mean that the sequence of the letters of a pair would not be expressed since, for example, the pair AB and the pair BA would be represented by the same intersection. This situation was substantially overcome by grouping the letters of the vertical set differently from those from the horizontal set. This was done in such a manner that no two or more letters of a group of a horizontal set would re-occur in a group of the vertical set. This requirement meant the creation of two sets of groups and the optimization of randomness in each set.

For statistical information on the frequency of usage of letters in the English language, the table in the "Journal of Franklin Institute" was used. Since this table did not differentiate between the various positions of letters within a word, it was deemed advisable to modify its values to reflect the frequency of starting letters on the basis of listings in Webster's Collegiate Dictionary. An average of four letters per word was assumed for the usage of the matrix and a new table of values was computed on the basis of one starting letter and three average letters. In Table No. 1, Fig. 2, the three sets of values are shown side by side. Table No. 2 lists the newly derived values in descending order.

---

\*H. P. Luhn, "Self Demarcating Code Words", IBM Engineering Laboratory, Poughkeepsie, N. Y., 1953



Letter Frequency Computation

Table 1				Table 2	
Letter	Frequency		Combined Average $\frac{3F + 1W}{4}$	Listing in the New Order of Frequency	
	Franklin	Webster			
A	8.1	6.4	7.5	E	9.5
B	1.5	5.4	2.5	T	8.5
C	2.9	9.7	4.5	A	7.5
D	3.7	5.1	4.0	S	7.5
E	12.1	3.7	9.5	I	6.5
F	2.3	4.3	3.0	O	6.0
G	2.0	3.3	2.5	N	6.0
H	5.4	3.9	5.0	R	5.5
I	7.3	3.9	6.5	H	5.0
J	0.1	0.9	0.5	C	4.5
K	1.7	0.8	1.5	D	4.0
L	4.0	3.4	4.0	L	4.0
M	2.5	5.2	3.0	P	3.5
N	7.3	2.0	6.0	M	3.0
O	7.5	2.3	6.0	F	3.0
P	1.9	8.6	3.5	B	2.5
Q	0.1	0.6	0.2	U	2.5
R	6.0	4.7	5.5	W	2.5
S	6.1	12.0	7.5	G	2.5
T	9.3	6.0	8.5	Y	2.0
U	2.8	1.6	2.5	K	1.5
V	1.0	2.1	1.5	V	1.5
W	2.1	3.2	2.5	J	0.5
X	0.1	0.1	0.1	Q	0.2
Y	2.1	0.3	2.0	Z	0.2
Z	0.1	0.3	0.2	X	0.1

FIG. 2



The grouping of the letters into the final arrangement was made to produce, as nearly as possible, a reasonably even distribution of combined averages of letters:

1. For fully spelled-out words.
2. For the Consonant Code.
3. For the Significant Letter and the ELCO Code.
4. For the Self-Demarcating Code Words.

The following rules for distributing the letters were observed:

1. To try for an optimum distribution of consonants for the Consonant Code.
2. To assign the vowels and the letters L and R singly, e.g., one to a row or column, in order to optimize the distribution of the inner letters of the Self-Demarcating Code words.
3. To pair each of the letters L and R with one of the consonants W, H, Y.

A table of groupings arrived at on the basis of the above considerations is given in Fig. 3. The left half of the table is the distribution for one axis and the right half of the table is the distribution for the other axis. Columns b, c, and d contain the consonants except W, H, and Y. Column b contains the first eight consonants of Table 2 listed downward in descending order, except for S and R. Column c contains the next eight consonants of Table 2 in descending order listed upwards. In column c of the right-hand portion of the table the sequence of the letters has been changed by transposing the entries in column a. Thus, 1 and 2 have been interchanged, as well as 3 and 4, 5 and 6, and 7 and 8. The vowels and W, H, and Y were then entered into column a in accordance with the above rules. The letters W and Y in the left-hand side of the table have been paired with R and L respectively. Column a of the right-hand side was then derived from column a of the left-hand side by transposing the first four and the last four entries from column a on the left. In columns d and d' the two consonants X and Z were entered by adding them to combinations which contain a vowel. They were not combined with U because of the degree of significance this vowel has in the significant letter code. In the right-hand side of the table the association of these two letters was varied so that they would not appear together with the letters they were combined with on the left-hand side.



Letter Grouping for an  
8 x 8 Randomizing Square

	a	b	c	d	Cons. Code b/c/d	Total	a'	b'	c'	d'	Cons. Code b'/c'/d'	Total
1	U 2.5	T 8.5	Q .2		8.7	11.2	O 6	T 8.5	J .5		9	15
2	I 6.5	S 7.5	J .5	X .1	8.1	14.6	A 7.5	S 7.5	Q .2	Z .2	7.9	15.4
3	W 2.5	R 5.5	V 1.5		7	9.5	Y 2	R 5.5	K 1.5		7	9
4	E 9.5	N 6	K 1.5		7.5	17	H 5	N 6	V 1.5		7.5	12.5
5	O 6	C 4.5	G 2.5	Z .2	7.2	13.2	U 2.5	C 4.5	B 2.5		7	9.5
6	A 7.5	D 4	B 2.5		6.5	14	I 6.5	D 4	G 2.5		6.5	13
7	Y 2	L 4	F 3		7	9	W 2.5	L 4	M 3		7	9.5
8	H 5	P 3.5	M 3		6.5	11.5	E 9.5	P 3.5	F 3	X .1	6.6	16.1

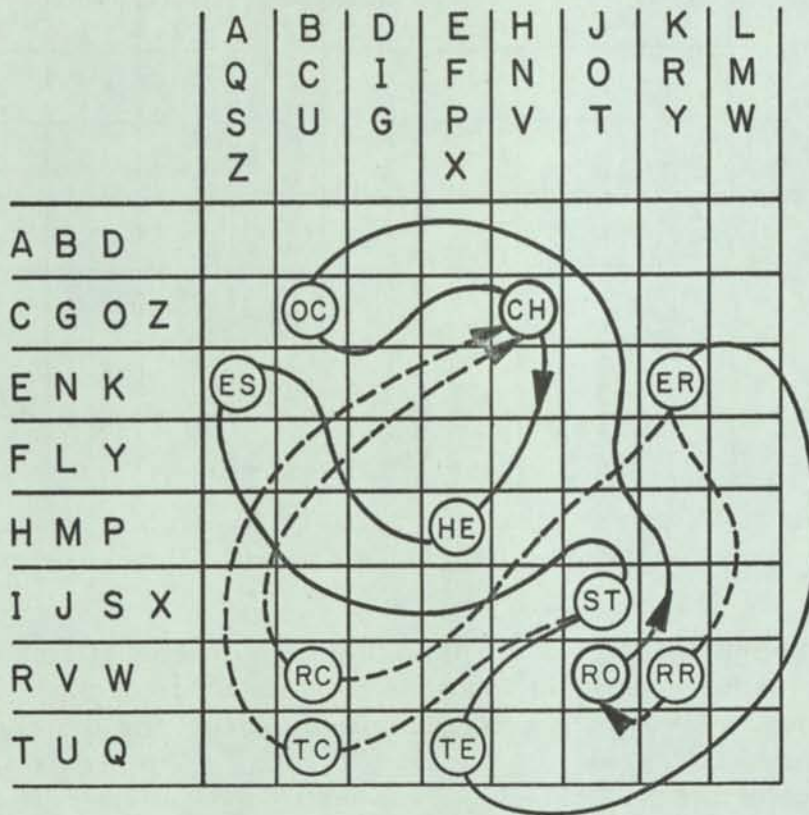
FIG. 3

Beside the columns of Fig. 3 are the combined averages, first for the consonants of the consonant code and then for the whole group. It is apparent that the grouping favors the consonant combination for the consonant code, the combined averages ranging from 6.5 to 9. As far as self-demarcating code words are concerned, a reasonably even distribution is assured by having each group contain outside letters and inside letters in the same proportion. The total averages range from 9 to 17.

While it is felt that the above arrangement is a reasonable solution for the application it was designed for, other groupings may be in order when dealing with other codes or languages. In this connection reference is made to "Interlingua" because of its application to scientific information. In any case the procedures covered here will facilitate the construction of an optimum matrix.

In order to arrive at the final recording square, the two lists represented by the two halves of the table in Fig. 3 have been arranged at right angles to each other to designate intersecting rows and columns. In this case, the left-hand list has been used as an index for the rows and the right-hand list as an index for the columns. The resulting arrangement is shown in Fig. 4. For convenience of reference the letters within a group





COMPLETE  
SPELLING

WORDS ENCODED:

CHEST  
CHESTER  
ROCHESTER

FIG. 4



and the groups have been re-arranged in alphabetical sequence, since the actual position of a group with respect to other groups is immaterial.

As an example, in the square of Fig. 4 an entry has been made of the words 'CHEST', 'CHESTER', and 'ROCHESTER' with all the letters spelled out. The various marks have been interconnected to indicate the chain formed by the sequence of the pairs of letters. The end-around portion of the spelling in each of the three cases has been indicated by dotted lines and the affected marks have been labelled accordingly. The arrows in each case point at the beginning of each of the words. The function of end-around spelling to differentiate the three words becomes apparent in this diagram.

In Fig. 5 there is an example of each of the patterns created by consonant spelling, significant letter spelling, and self-demarcating code spelling. They are shown as applied to an abstract of the same subject matter given as a single sentence in Fig. 6. The sentence has been written in eight lines, each line containing a 'notion' which might have been chosen by the editor as a differentiating element. The example is given to illustrate the various word codes and is not intended to teach a particular method of abstracting.

In the column to the right of the complete sentence, the notional terms assigned by the editor are given. The word codes for these are listed in a separate column for each of the four spelling methods.

The pattern derived by consonant code spelling is shown in Fig. 5. The various words have been numbered and the marks in the square have been identified by these numbers to facilitate tracing the procedure of chain spelling. The spelling of the 28 letters resulted in 25 marks, three of which are double entries.

The pattern created by significant letter spelling is shown in the lower left of Fig. 5. In this case the 32 letters resulted in 25 marks of which five are double entries and one is a triple entry.

The pattern of self-demarcating code spelling is shown in the lower right of Fig. 5 where 29 letters resulted in 24 marks including five double entries.







Abstract	Notion	Cons. Code	Sig. Letter Code	ELCO Code	Self- Dem. Code
The recording	write	WRT	WRIT	WRIA	WRIT
of information	inform	NFRM	IFRM	IFMW	NIF
by superimposed	merge	MRG	MERG	MRGT	MERJ
code combinations	code	COD	CODE	CODE	KOD
and the selection	select	SLCT	SLCT	SLCO	SLEX
of desired documents	book	BOK	BOOK	BOOK	BOOK
by a statistical method	approximate	PRXMT	APXM	APXS	PROX
of scanning and matching	scan	SCN	SCAN	SCAN	XAN

FIG. 6

# THE DESIGN OF A 10 x 6 RANDOMIZING 'SQUARE' FOR MIXED SYMBOLS

These design procedures were formulated to create squares with alphabetic indexes. It is apparent that any other set of symbols could be adapted for randomizing and for the distribution of code marks by the method of chain spelling. The use of numerals for serial numbers and for numeric codes makes it desirable to enter such information in superimposed fashion and preferably in conjunction with alphabetic information.

A scheme which permits entries in mixed symbols will be described next. For the purpose of this example it has been assumed that the frequency of usage is the same for all numerals. It has also been assumed that the size of the square should be similar to the square developed previously so the code patterns could be recorded on a punched card.



Because there are ten numerals, the format for this square was chosen because it provided a balanced distribution of the ten entries in the indexes of the rows as well as the columns. A ten by five square accomplishes this by assigning one numeral each to the ten rows and by assigning the five pairs; 05, 16, 27, 38, 49 to the five columns. In order to identify the end of a chain, an "End Mark" column is provided for entering the last character of a term instead of pairing it with the first character of a term as in the previous examples. This sixth column brings the size of the square to ten fields high and six fields wide.

The letters of the alphabet have been grouped into two index sets, one for the ten rows and one for the five columns. The same principles of distribution have been used as were used in the previous examples. The letters 'Q' and 'I' have been arranged to coincide with the numerals '0' and '1' to overcome confusion between these symbols.

Fig. 7 shows the pattern created by an example of five number code entries comprising 30 characters.

For the purpose of comparison the word code entries of the eight by eight square have also been entered into the 10 x 6 version as shown in Fig. 8.

The groupings of characters and the combined averages of letter usage are given below.

<u>Row Index</u>				<u>Column Index</u>			
Average				Average			
All				All			
Letters				Letters			
				Cons.			
				Code			
COZ	0	10.7	4.7	FHOQS	05	21.7	10.7
DIJ	1	11	4.5	GIKLP	16	18	11.5
AMV	2	12	4.5	ACTWZ	27	23.1	13.1
BL	3	6.5	6.5	BDRUV	38	16	13.5
EFK	4	14	4.5	EJMNZY	49	21.2	9.7
GR	5	8	8				
HT	6	13.5	8.5				
NW	7	8.5	6				
PQU	8	6.2	3.7				
SXY	9	9.6	7.6				



		F H O Q S	G I K L P	A C T W X	B D R U V	E J M N Y Z	END MARK
		0 5	1 6	2 7	3 8	4 9	
C O Z	0	(2)	(2)				
D I J	1		(3)	(1)			
A M V	2			(1)	(4)	(3)	(1)
B L	3			(3)		(1)	(2) <sub>4</sub>
E F K	4	(1)	(1) <sub>1,3</sub>				
G R	5	(2) <sub>5</sub>				(5)	
H T	6			(1)	(2) <sub>3</sub>		
N W	7					(1) <sub>3</sub>	(5)
P Q U	8		(4)	(2) <sub>4</sub>			
S X Y	9			(1)			(3)

# NUMBER CODES

(1) 34F1746X2

(2) PC5063

(3) 2,416,379

(4) 828L

(5) 55N

(30 CHARACTERS)

18 SINGLE  
4 DOUBLE  
1 TRIPLE  
23 MARKS

FIG. 7



		F H O Q S	G I K L P	A C T W X	B D R U V	E J M N Y Z	END MARK
		0 5	1 6	2 7	3 8	4 9	
C O Z	0	(4)	(6)	(5)	(4)	(8)	
D I J	1						(4)
A M V	2			(7)	(3)		(2)
B L	3	(6)		(5)			
E F K	4				(2)		(6)
G R	5		(3)	(1) <sub>7</sub>		(2)	(3)
H T	6						(1) <sub>5,7</sub>
N W	7	(2)			(1)		(8)
P Q U	8				(7)		
S X Y	9		(5)	(8)		(7)	

CONSONANT CODE

(1) WRT

(2) NFRM

(3) MRG

(4) COD

(5) SLCT

(6) BOK

(7) PRXMT

(8) SCN

(28 LETTERS)

23 SINGLE  
1 DOUBLE  
1 TRIPLE  
25 MARKS

SIG. LETTER CODE

○	○	○	○		
○		○		○	
	○			○	○
○		○			
			○		○
	○			○	○
					○
			○		○
		○			
	○	○		○	

16 SINGLE  
8 DOUBLE

24 MARKS (32 LETTERS)

S.D.C.

○	○	○	○		
○		○			○
				○	
○				○	
○		○	○		○
○	○			○	
					○
	○		○		○
			○		
	○	○			○

21 SINGLE  
4 DOUBLE

25 MARKS (29 LETTERS)

FIG. 8



## CHECKING SCHEMES

Numeric codes which utilize self-checking features for eliminating transcription errors\* are particularly effective in superimposed coding schemes. This is so because the check digit, commonly used in such schemes, is systematically computed and consequently acts as a unique differentiating element. It will therefore be well to take advantage of this device because it tends to reduce the rate of spurious matches. These checking systems are equally adaptable to mixed and to purely alphabetic codes.

### SINGLE ROW RECORDING ON PUNCHED CARDS

It was considered more convenient to first demonstrate the new system and the various examples in their original two-dimensional form. As was mentioned earlier, one of the objectives was the creation of single row recordings of the patterns contained in the squares, each row representing a 'sentence.' This requirement is achieved by laying the rows of the squares end-to-end in a given order.

This is illustrated in Fig. 9 by an 80 Column IBM card for the combination alphabetic and numeric scheme. The 60 columns from 21 through 80 have been assigned to this scheme. The remaining 20 columns from 1 through 20 are available for the recording by conventional punch code of such information as serial number, which identifies the particular card as well as some broad class designations. Of this space, column 19 may be used to tie rows together to indicate 'paragraphs'. The absence of a punched hole in this column would indicate that the associated row terminates a 'paragraph'. Column 20 may be used to indicate the presence of numerical data in the associated row. If a recording exceeds the capacity of a single card, it may be necessary to tie two or more cards together by way of punches in another special column.

The principle of serial scanning of rows and the machine techniques that may be used for this kind of recording are similar to the ones developed in connection with the U. S. Patent Office searching experiment of 1950. The processing in this experiment was done by an IBM Type 101 machine, equipped for row-by-row searching. Since machine methods which

---

\*"Self-Checking Number System", International Business Machines Corporation, New York City, Publication : Form No. 22-6022-0. See also U. S. Patent No. 2,731,196 to H. P. Luhn dated Jan. 17, 1956.



# IBM CARD PUNCHED WITH 4 OF 12 POSSIBLE SENTENCES

LINE	6 X 10 RANDOMIZING SQUARE									
	0	1	2	3	4	5	6	7	8	9
1	000	000000000000000000	000000000000000000	000000000000000000	000000000000000000	000000000000000000	000000000000000000	000000000000000000	000000000000000000	000000000000000000
2	111	111111111111111111	111111111111111111	111111111111111111	111111111111111111	111111111111111111	111111111111111111	111111111111111111	111111111111111111	111111111111111111
3	222	222222222222222222	222222222222222222	222222222222222222	222222222222222222	222222222222222222	222222222222222222	222222222222222222	222222222222222222	222222222222222222
4	333	333333333333333333	333333333333333333	333333333333333333	333333333333333333	333333333333333333	333333333333333333	333333333333333333	333333333333333333	333333333333333333
5	444	444444444444444444	444444444444444444	444444444444444444	444444444444444444	444444444444444444	444444444444444444	444444444444444444	444444444444444444	444444444444444444
6	555	555555555555555555	555555555555555555	555555555555555555	555555555555555555	555555555555555555	555555555555555555	555555555555555555	555555555555555555	555555555555555555
7	666	666666666666666666	666666666666666666	666666666666666666	666666666666666666	666666666666666666	666666666666666666	666666666666666666	666666666666666666	666666666666666666
8	777	777777777777777777	777777777777777777	777777777777777777	777777777777777777	777777777777777777	777777777777777777	777777777777777777	777777777777777777	777777777777777777
9	888	888888888888888888	888888888888888888	888888888888888888	888888888888888888	888888888888888888	888888888888888888	888888888888888888	888888888888888888	888888888888888888
10	999	999999999999999999	999999999999999999	999999999999999999	999999999999999999	999999999999999999	999999999999999999	999999999999999999	999999999999999999	999999999999999999

The 4 entries on this card are those of the 4 squares shown in figures 7 and 8

FIG. 9



might be applied to perform searches are not a topic of this paper, reference is made to the report on this experiment for further information\*.

### RECORDING ON TAPES

The creation of single row recordings facilitates the processing by means of certain devices. However, the method of encoding just described has advantages which make its use attractive also for use in those devices which are capable of searching two-dimensional arrays of information. Under certain conditions, a substantial saving of recording space can be achieved and the process of searching can be simplified.

If the code words used in the preceding example were to be spelled out, at least 40 characters and division marks would be required. When using self demarcating code words, this number may be reduced to 30. If a six-bit code were used to record this, the space required would be the equivalent of  $30 \times 6$  or 180 bits. The recording of a ten by six randomizing square would require a space for only 60 bits. This is the space needed for ten characters so that in this particular instance a reduction of 4 or 3 to 1 could be realized. Because of this reduction, less storage space and functional capacity would be required and processing time would be materially shortened. Whether these reductions would pay off depends on the relative merits of the statistical and the discrete searching methods as applied to a given situation.

If it is desired to record randomizing squares on seven-channel punched or magnetic tape, this could readily be done by sectioning the squares into six-bit strips and by recording these across the tape the same way as normal characters are recorded. If desired, a bit count may be made for each row and a redundancy bit may be added in the seventh channel where required. In the case of the  $10 \times 6$  square, no rearrangement of the pattern is required.

### CONCLUSION

The principle of distributing code entries by the use of randomizing squares and the principle of chain spelling have been demonstrated with the aid of letter frequency tables derived from common English literature. In order to obtain greatest efficiency, it would be advisable to compile special frequency tables for scientific literature.

---

\*Mechanized Searching in the U. S. Patent Office, M. F. Bailey, B. E. Lanham, and J. Leibowitz, Journal of the Patent Office Society, Vo. 35, pp. 566-587.



The principles described above may be applied equally well to any foreign language and to artificial languages such as 'Interlingua', provided letter use frequency tables for these languages are available.

Superimposed coding offers many advantages where recording space is at a premium. For this reason it is being used extensively in marginally punched card systems. For the users of such systems the randomizing square method of recording may offer additional advantages.

H. P. Luhn  
June 15, 1956





DIVISION OF ENGINEERING RESEARCH

7 October 1960

Quarterly Status Report 3  
Covering the Period 1 July to 30 September 1960  
Stanford Research Institute Project 3101

MULTIPLE INSTANTANEOUS RESPONSE FILE

by  
Jack Goldberg

Contract AF 30(602)-2142

Prepared for  
Rome Air Development Center  
Griffiss Air Force Base  
Rome, New York

This report is intended for internal management uses of the Contractor and the Air Force.

Copy No. 19



## I OBJECTIVE

The object of the project is to investigate the feasibility of construction of a special file to be used in the retrieval of information. This file should contain descriptions of up to two million documents, and it should behave in such a way that shortly after an inquiry is made, indications should appear simultaneously for all documents, if any, whose descriptions logically include the descriptions and relationships specified in the inquiry. This behavior is summed up in the title of the project: "Multiple Instantaneous Response File."

The behavior described may be contrasted with conventional large-scale retrieval machines, in which data stored in some kind of a memory are scanned sequentially by a specialized logical testing organ. It is expected that a multiple instantaneous response file (MIRF) would require an intimate merging of the logical testing function and the storage function in its working elements.

## II TECHNICAL STATUS

The research of the Third quarter will be described under the following headings: Use Aspects, Coding, Magnetics, Cryogenics, and Optics. Most of the work of this quarter has concentrated on coding and on magnetic realization.

### A. Use Aspects

Although the objectives of the project do not include study of the possible use of a MIRF, the question has naturally arisen in the course of the feasibility study. For example, in the second quarterly progress report a simple analysis was presented of a hypothetical serial scheme using available devices, for comparison with a MIRF, with the assumption that both systems had the same average output rate. This comparison showed that it is not necessary to use a MIRF to obtain high average rates.

The analysis suggested that the primary advantages of a MIRF, deriving from its low delay time, would be (1) use in a man-machine cycle, in which a user can quickly improve his question on the basis of the machine's response, and (2) the simplification of handling the traffic of incoming and outgoing data in a busy center.

In a recent paper,<sup>1</sup> Professor Y. Bar-Hillel suggests that the best use for a machine in literature retrieval is in a man-machine cycle, in which the machine quickly presents the answers to the man's formulations in a given index system, but in which the man is responsible for making the semantic links to alternative formulations. He bases this conclusion on his belief that it is futile to try to construct a closed set of rules by which a computer could autonomously arrive at good judgments of the closeness of topics relative to a user's interest, except with enormously complicated document descriptions. He states:

---

<sup>1</sup>"Some Theoretical Aspects of the Mechanization of Literature Searching," Tech. Report 3, Contract N62558-2214, U.S. Office of Naval Research, Hebrew University, Jerusalem (April 1960).



"The only steps in a complete literature search which I can see as being usefully turned over to computers are:

- (1) The storage of the references and their index sets
- (2) The matching of some Boolean function over a given set of topic terms with the various stored index sets
- (3) The printing out of reference lists corresponding to these matchings."

This point of view is an encouraging one for the MIRF program, since a MIRF would be an ideal machine companion, permitting the user to make many variations in his interrogation without annoying delays between question and response.

It is important to distinguish between literature searching and fact retrieval. The former requires a search for things whose relatedness must be deduced from imperfect and limited terms of a simple language, whereas the latter is a search which, almost by definition, will be satisfied by positive responses to a Boolean test on the symbols which comprise a description--e.g., the ages, dimensions, I.Q.'s, etc., in a personnel file. The advantages of a man-machine team would be great in literature searching, but not necessarily for fact retrieval, since, in fact retrieval, the inquirer will usually be able to state the terms of interest and their logical connection without need for many modifications.

It is appropriate, in the light of the above remarks, to consider the significance of the logical OR (or "Union," or "disjunction") operation in the interrogation test. An important function of this operation is to act as a "net" to catch items which are described by synonymous (for the purpose of the user) terms or which belong to a set which can be described only by enumerating its members. The single order could always be broken down into elementary components, each of which could comprise a separate test, the result of which would be summed in a list.

Since all previous large retrieval machines were serial, the availability of the OR test was vital, since the restriction to elementary conjunctive expressions would have required either the repetition of long serial searches for each elementary quiz or the provision of many parallel (and expensive) detectors. In a MIRF, decomposition of a disjunction into a set of elementary interrogations would not appear to be burdensome in time or equipment. It may be, therefore, that the OR function is of considerably less importance in a MIRF than in a serial machine.

Another important question in the area of usage is that of providing the supplementary data beyond the mere list of serial numbers, to aid the searcher in deciding whether the response to his quiz is satisfactory and adequate. This is an especially difficult subject to consider in the abstract, without consideration of a particular subject matter or system of users. In any case, consideration of the various schemes for realization of the MIRF thus far indicates that economic considerations will severely limit the amount of information (perhaps only the index set) which can be recorded for each item.



Thus it will be appropriate to consider various "back-up" media, such as books, cards, etc., which will present the necessary supplementary data. The medium should be low-cost and random-access in construction to take advantage of the address information provided by the MIRF.

In the comparison of a MIRF with a hypothetical serial system mentioned above, it was assumed that the serial system was capable of grouping interrogations and distributing responses from and to many interrogators. This involves a queue with fixed service time, whereas a MIRF, serving many sources by time multiplex, would involve a queue with, to a first approximation, negligible service time. The simpler system integration problem of the MIRF may be considered as a positive entry in a list of advantages and disadvantages.

## B. Coding

### 1. Transformations on Superimposed Code Designs

Since it is expected that a MIRF will be realized using two-state elements, there has been continuing study of binary codes suitable for representing the file data and for performing the required logical tests. The amount of information which can be represented by a given binary code field depends upon the length of the field and on the number of marks which are allowed. In some physical realizations, the cost of a mark may be much greater than the cost of an unmarked field position; thus it is desirable to be able to transform one code design to another in order to obtain the most economical structure.

In Quarterly Status Report 2, a number of alternate codes were discussed, suitable for data which are subject to equality and size comparisons. In the third quarter, this study was extended to codes designed for inclusion tests--in particular, the codes in which descriptions and interrogations are composed by superposition of elementary sets of marks.

Superimposed codes are of two kinds, \* "single field" and "multiple field." In the first, there is no restriction on the location of marks, while in the second, each component descriptor is composed of one mark in each of several fields, each mark taking on locations within its field independently of the other marks, if the rule of pattern generation is a random one. The number of places in the field and the number of marks permitted must be designed to allow a sufficient vocabulary and to minimize false responses due to the inherent ambiguity of the scheme, but to avoid wasted capacity.

According to the analyses published in the literature,<sup>2,3</sup> the marks in the fields are best utilized when the fraction of places marked in the

---

\* In all discussions of codes, unless otherwise noted, it will be assumed that all data fields have the same length.

<sup>2</sup> C. Wise, Punched Cards, 2 ed., Chapter II, (Reinhold Publishing Corp., New York, New York, 1958).

<sup>3</sup> C. Mooers, "Zato Coding for Punched Cards," Zator Tech. Bull. 30 (1950).



average document is, in the single-field case,  $\frac{1}{2}$ , and, in the multiple-field case  $\frac{1}{e}$ . One subject of consideration in this quarter has been the effect on coding efficiency of departures from optimum design. For single-field coding it may be asked, for a given (small) error rate, how many positions must be added to the field length if the number of marks is reduced below that specified for optimum design. The analysis and a set of curves for a representative case are presented in Appendix A. Briefly, the results are favorable--i.e., the optimum is not sharp, and useful departures from it do not require unreasonable expansion of the field.

For example, assuming a maximum of ten descriptors per file document and a minimum of four descriptors in a quiz, the optimum number of marks for a random (generally false) response rate of  $10^{-6}$  responses per file item is five marks in a field of 81 positions. Due to occasional double marking, the average number of marked positions will be 38.3 (with a maximum of 50). If, now, the number of marks per descriptor is reduced to three, the field must be lengthened to 87 in order to keep the random response rate (called "dropping fraction" by some authors) constant. The average number of marked positions will be 25.8.

Thus, reducing the average number of marks from 38 to 26 required increasing the field length from 81 to 87 (for the assumed file parameters). For those files in which the cost of a mark is more than half the cost of a field position, such a departure from "optimum" would be justified.

There is another method for reducing the number of marks in a superimposed single field code, which may be explained by the following three steps:

- (1) Assume a binary pattern resulting from superposition of conventional descriptors--e.g.,

0 1 0 0 1 1

- (2) Group the binary digits so as to represent a number in a higher base code--e.g., for a base  $2^3 = 8$ , grouping three bits at a time, as follows:

0 1 0, 0 1 1

- (3) Represent the new digits using a code in which the number of marks is restricted--e.g., for this example, a "1 out of 8" code. This will now give a representation of greater length, but with fewer marks, as follows:

0 0 0 0 0 0 1 0, 0 0 0 0 0 1 0 0

A cost analysis for different radices is given in Appendix B. The best results given by this transformation are much poorer than those given by variation of the basic code.

It should be mentioned that there exist simple interrogation rules in the transformed number system for accomplishing the necessary inclusion operations.



## 2. Improved Analysis of Superimposed Coding Principles

The analyses of the behavior of superimposed codes published by Mooers and Wise use approximations which appear intuitively reasonable but which cannot be justified on strict mathematical grounds. For example, the distribution of the number of marks resulting from superposition of a fixed number of randomly selected descriptors is characterized by the mean of the distribution, both for the file items and for the interrogation. The expected number of random responses ("drops") is then obtained by combinatorial operations on the mean of the distribution of marks in the file items and either a specific interrogation with a given number of marks or the mean of the distribution of marks in a set of interrogations.

Properly, the number of random responses should be obtained by operations on the distribution of marks rather than on the mean of the distribution. Such an analysis is not easy, and the existing analyses have presented their results as upper limits.

Dr. R. Singleton, of the Mathematical Statistics group at SRI has made some progress towards rigorous analysis of the problem. He is currently preparing a report on his studies, and is considering the preparation of a program of computation based on his analysis.

All the above analyses, of course, assume an ideal model--i.e., with random, uniform selection of descriptors in both file items and interrogations.

### C. Magnetics

The most active investigations into physical realizability has been in the area of magnetics. This study will probably continue through the duration of the project for various reasons--e.g., the high state of development of many materials, both in high quality and low cost, the flexibility of structural design, and the extensive design experience available.

In Appendix C there are presented several possible types of organization of a magnetic MIRF. There is little doubt that, using available devices--e.g., ferrite memory and switch cores, a MIRF could be realized in one or more of the structures; however, the cost (or inconvenience) of assembly of the MIRF would be unreasonable. Most of the study of magnetic realization has been a search for the proper combinations of physical structure and materials and assembly techniques which give a simple, economical means of recording the file data.

Numerous schemes have been considered and dropped and there are presently several under serious study. At the present time it would be premature to perform detailed analyses of the various schemes. Rather it is planned to make rough studies of the most promising of the very many possibilities, then choose the best of these for detailed study.

Appendix C presents a brief qualitative description of basic file organizations, coding techniques, and examples of physical elements for a magnetic realization.



In a large file, power dissipation may become significant, and so there has been some study of magnetic materials which would require low driving currents. In the second quarter, a number of different metallic alloy magnetic materials were tested. Unfortunately, the more magnetically sensitive of these are physically delicate, and the cost of adequately protecting a metal element appears to make them unfeasible for MIRF.

In the third quarter, hysteresis characteristic tests were conducted on experimental ferrite cores furnished by Telemeter Magnetics Inc. The results of 60-cycle tests are summarized in Table I, below. For comparison, it may be noted that common memory cores have coercive forces in the range 0.7 to 1.5 oersteds.

TABLE I  
MAGNETIC CHARACTERISTICS OF EXPERIMENTAL LOW-DRIVE CORES

Sample No.	D179	D180	D181
Coercive Force ( $H_c$ ) at 20°C (oersteds)	0.32	0.27	0.13
Temp. Coeff. of $H_c$ (oersteds/°C)	0.0019	0.0017	0.0017
Relative Total Flux	1.46	1.4	1
Temp. Coeff. of Flux $\left( \begin{array}{c} \% \text{ of } 20^\circ\text{C} \\ \text{Value} \\ \text{per } ^\circ\text{C} \end{array} \right)$	0.52	0.54	1.4

All of the materials are useable, and have good squareness, but the most sensitive (i.e., requiring the least amount of drive to achieve saturation), D181, requires the most careful temperature control.

#### D. The Cryogenic MIRF

As previously reported in Supplement C to Quarterly Report 2, we have been considering methods whereby the unique characteristics of superconductivity might be employed to realize a "two-terminal MIRF" of large storage capacity. Two incentives for pursuing this investigation are (1) the fact that the very ideal nature of superconductive switching phenomena offers the possibility of constructing files of almost arbitrarily large size which may be interrogated as a single unit, and (2) the fact that the methods visualized for constructing such a file appear to meet the difficult cost criteria which must be associated with files of very large capacity.

In the early planning stages of the MIRF project it was realized that superconductivity would certainly deserve investigation in device configurations which had never previously been constructed or tested. Accordingly, a small cryogenic laboratory was set up (at a total equipment cost of less than \$1000) so that we would have facilities for showing the feasibility of our proposed cryogenic devices.



Along experimental lines, however, our total cryogenic effort to date has been quite small, consisting mainly of assembling and checking out the apparatus and making two or three simple experiments with bulk materials.

Plans for the next quarter include the construction and testing of one or two feasibility "samples" of MIRF file structure. These devices will be full-fledged miniature MIRFs patterned after the arrangement shown in Fig. 1 (this figure was reproduced as Fig. 4 in Supplement C of Quarterly Status Report 2). The first prototype will consist of eight words of 9 bits each. Three bits of each word will be dedicated to explicit coding of a binary accession number. The remaining 6 bits will be used as a field for superimposed coding. The data mesh and quiz-conductor array (shown in Fig. 2 of this report) will be etched from thick films of tin and lead respectively on glass substrates. The shape of the quiz conductors has been chosen so that no switching of horizontal conductors in the data mesh will occur. This allows each set of conductors to be fabricated from one material only, a great convenience. Data entry will be accomplished by mechanically removing some of the data mesh conductors after the initial mesh pattern has been produced by a photo-etching process.

## E. Optics

### 1. Background

The term "optics" is used very loosely here, to refer to possible realizations for MIRF using light as a medium for propagating information. Two "classic" types of structure were described in the last quarterly report, which might be characterized as "volumetric," or "planar," respectively. The "peek-a-boo" may be considered as "volumetric," since in the classic form, light must propagate through a stack of baffles. The many schemes of the "Filmorex" type may be considered as "planar," since there is only one storage plane between light source and light detector.

Early in the project it was evident that a peek-a-boo realization of MIRF would not require the file structure used in the conventional mechanical (card) form--i.e., one card (or baffle) per descriptor. Rather, it is possible to use combinatorial coding as in other realizations--e.g., a binary field of one hundred or so bits, in explicit or superimposed coding, requiring one or two baffles per bit.

In the first two quarters, a number of "optical" schemes were conceived and recorded for further study. The planar types enjoyed the feature of low-cost masks, but suffered from the high cost of presenting coded light signals to all the elements in the file, whereas the volumetric types did not essentially require coded light sources, but required expensive and unwieldy elements.

### 2. Use of Electro-luminescent and Photoconductive Elements

In the past quarter, several structures have been conceived employing combinations of electro-luminescent elements and photoconductive elements, which



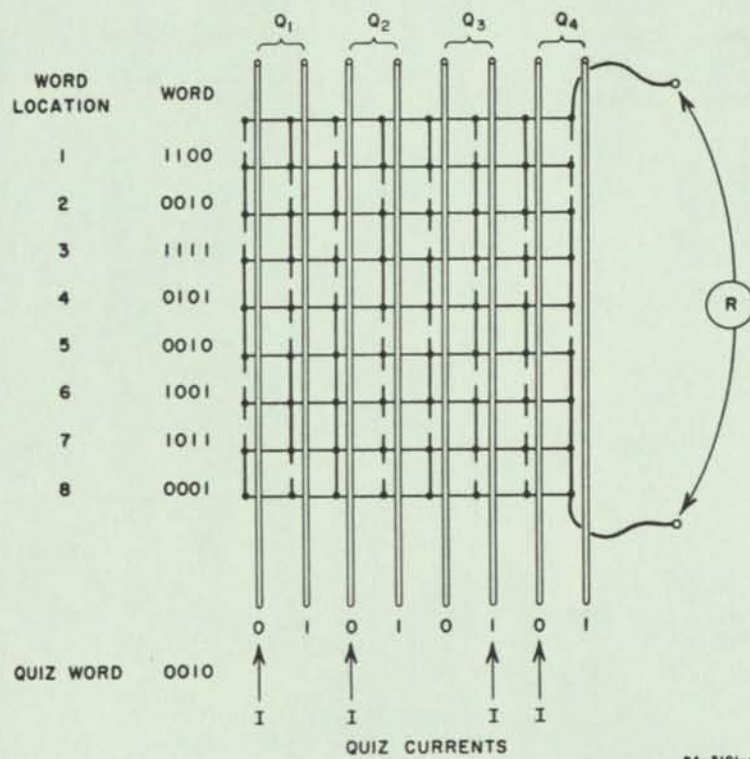


FIG. 1  
A PROPOSED SUPERCONDUCTIVE TWO-TERMINAL MRF



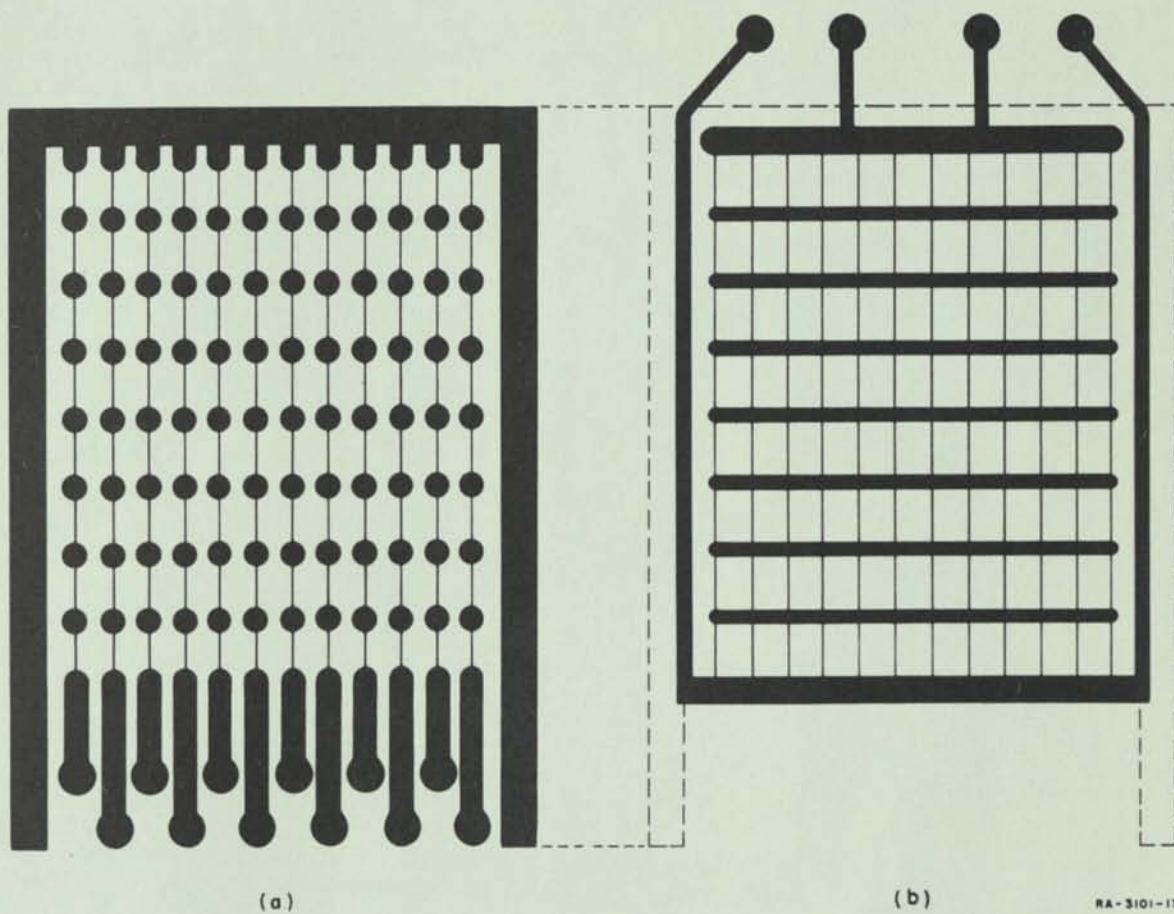


FIG. 2  
CONDUCTOR PATTERNS FOR A CRYOGENIC MRF

RA-3101-17



offer very attractive possibilities in both planar and volumetric structures. Simple sketches of possible structures are given in Figs. 3, 4, and 5.

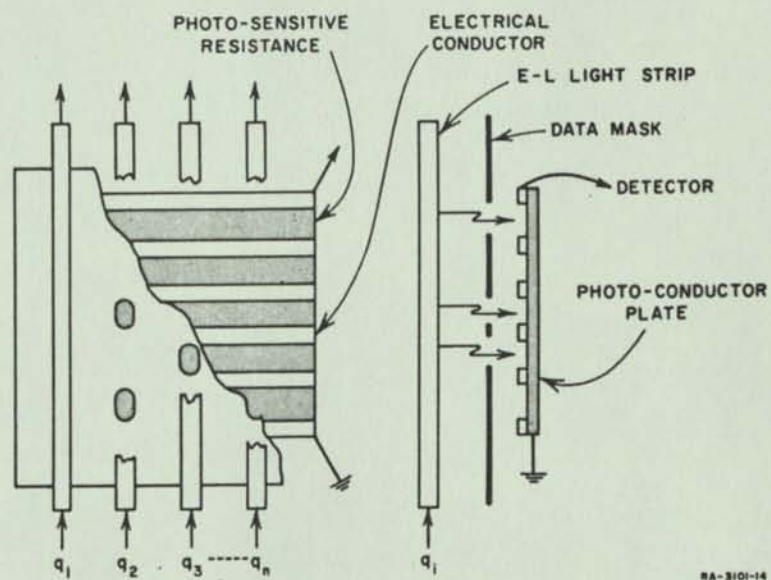
Figure 3 shows a planar structure, consisting of a set of electro-luminescent strips, energized according to the bits of an interrogation, whose light may pass through the holes punched in a data mask to be detected by a special photoconductive detector surface. As shown, this surface is made of a base of uniformly deposited photoconductive material upon which are deposited horizontal strip conductors. Successive pairs of conductor strips serve to define a data word, since if any light is permitted to fall on a region, the lowered conductivity of the region between strips causes the two strips to come to about the same potential. The logic is entirely the same as that of the superconductive structure described in Appendix C of Quarterly Status Report 2. Since, however, the ratio of "off" to "on" resistance is not infinite, the strips would be grouped and detected by separate threshold elements. It appears that the size of the groups may be large, since the dark-to-light resistance ratio is quite large.

Figure 4 shows two successive elements of a volumetric (peek-a-boo) structure. The long, vertical, unshaded rectangle represents the edge view of an electro-luminescent cell, with conductive surfaces, the right-hand surface transparent and the left-hand surface opaque. Bonded to the left-hand face are two photoconductor cells whose right-hand faces are electrically connected to the common conductive face of the luminescent cell, but whose left-hand faces are connected to two separate signal voltage busses. A file would be made up of a stack of sheets composed of such elements, all elements on a given sheet serving a particular bit in the code field. The sheets would be separated by punched paper or card masks, containing the stored data. The punches in the data mask allow light to fall upon either one or the other of the two photoconductors. If, at a given bit position, light shines on a conductor, and the bus to which the conductor is connected is energized by an interrogation voltage, this voltage will be applied to the entire luminescent cell. The light produced will shine on both regions of the next mask. If the quiz voltages of all the bits are applied to conductors whose mask is punched, a chain of light emission and detection will pass through the entire set of baffles composed of such elements, to be detected at the last plate. It may be noted that, since light is regenerated, precise mechanical registration is not needed, and that since most chains will be blocked at early stages, only a small fraction of the total possible light production capability in the file may be expected at any given quiz.

Figure 5 is a kind of dual realization of Fig. 4. The same types of regenerative elements are used, but the light in a successful chain will pass back and forth successively through the two faces of a single mask.

So far, only rough qualitative consideration has been given to these and similar optical schemes. Several good features make them worthy of further study. First, data would be easily changeable; second, the technology of the materials involved appears to be reaching a satisfactory development with respect to reliable life and low cost; and third, the materials and the logic appear amenable to uniform mass production.





RA-S101-14

FIG. 3  
A PLANAR OPTICAL MRF



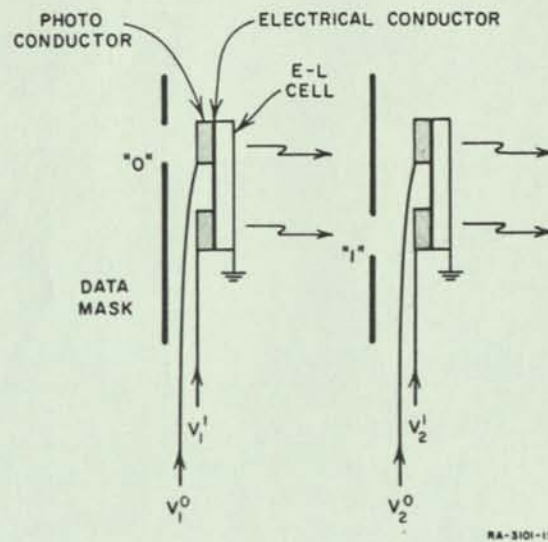


FIG. 4  
ELEMENTS FOR AN ELECTRO-LUMINESCENT  
PEEK-A-BOO MIRF, FIRST TYPE

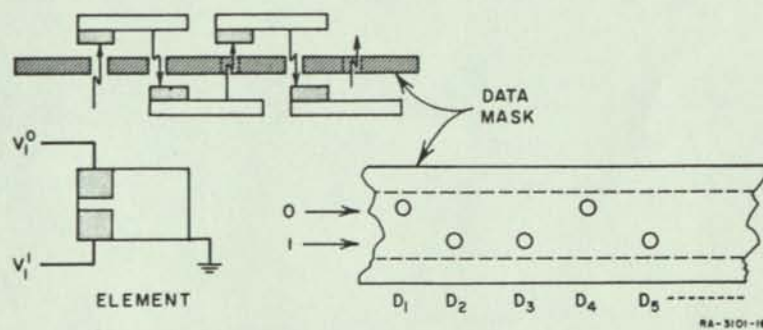


FIG. 5  
ELEMENTS FOR AN ELECTRO-LUMINESCENT  
PEEK-A-BOO MIRF, SECOND TYPE



The structures are easily adaptable to both explicit and superimposed coding, and the low cost of the threshold elements may make even more sophisticated logical structures possible.

### 3. Light Inversion

In the mechanical peek-a-boo, a match is indicated by the presence of a spot of light at a given point in a dark field. In the search for an optical realization of MIRF, several non-moving light-gating elements have been considered, such as a multi-input Kerr cell, or a pressure-sensitive plastic-rod polarizer, etc., which would give a spot of light for a non-match condition, and darkness for a match. It is to be expected that a single detector, either human or electronic, would have much greater difficulty recognizing a spot of darkness in a field of light than a spot of light against darkness.

It would be convenient, therefore, to be able to "invert" the light signal--i.e., to convert a spot of light to a spot of darkness, and vice versa. The "light inverter," screen described in the next paragraph is possible, and indeed, commercial versions, used in certain photographic processing, exist.

Some phosphorescent substances go into long-lived meta-stable states when irradiated by short-wave-length light (UV or blue light). When, subsequently, infra-red light falls on such a substance, the meta-stable state is destroyed, either by complete extinction of the phosphorescence (called "quenching") or by immediate release of all stored light energy. In both types of these substances, the final result is the same: dark spots on all places of incidence of infra-red light, and phosphorescent spots at places where there was no infra-red light.<sup>4,5</sup>

Some samples of such phosphors have been supplied by the United States Radium Corporation, but have not yet been tested.

### III SUMMARY

The major areas of study in the third quarter were Use Aspects, Coding, and Magnetics, with some preparatory work and study of cryogenic and optical realizations.

In considering use aspects it was observed that a MIRF would make feasible a close, rapid, man-machine system, and that this would tend to overcome many

---

<sup>4</sup>H. W. Leverenz, Luminescence of Solids (John Wiley & Sons, Inc., New York, New York, 1950).

<sup>5</sup>R. Ward, "Preparation and Properties of Infra-Red Sensitive Strontium Selenide and Sulfide-Selenide Phosphors," J. Opt. Soc. Am. 36, pp. 351-352 (1946).



shortcomings of rigid classification schemes. Further, due to the very rapid response time, simple combinatorial interrogation functions might be satisfactory, rather than the complicated ones needed to overcome speed limitations in serial machines. Another possible advantage over serial systems would be the simplification of the input-output queueing problem in a busy document center.

Since it appears that the most feasible MIRFs will be limited to little more than serial number output, it is important to consider different "Back-up" media, such as books and cards.

Superimposed Coding has been considered in two aspects. First a study was made to determine the relative efficiency of non-optimum marking schemes, which might be important if the cost of a mark is significantly different from that of a place in the code field. The results indicated that only a modest extension of the code field is necessary to allow for useful restrictions on the number of marks. Secondly, a more rigorous analysis of the behavior of a superimposed code file was started than has previously been published, with very promising results. These results will be presented in the next quarterly report.

A description was presented of the basic principles of realization by static magnetic elements. The detailed work of evaluation of different specific realizations is in progress. The characteristics of some experimental, commercial, low-drive ferrite material were measured and presented.

In the third quarter the various equipments (such as transfer tubes, pumps, etc.) needed to make the cryogenic test facility operational were built or purchased. A pattern has been prepared which will be used to etch out a prototype, eight-word (9 bits each) MIRF. This device will be built, and testing will start, in the coming quarter.

Several attractive structures using electro-luminescent and photoconductive materials have been conceived. Further consideration will be given to this technique, but full study may not be started until the fifth quarter. The possibility has been recognized of accomplishing inversion of a light signal (light-for-dark and dark-for-light), using the "quenching" phenomena in certain phosphors. Materials and devices are commercially available, and may be useful for certain kinds of optical realizations.

#### IV FUNDS

As of 30 September 1960, expenditures total approximately \$79,943, leaving a balance of \$76,220. This balance is considered satisfactory for the completion of the project.

#### V PERSONNEL

The following persons participated in the research of the third quarter: Dr. E. Frei, J. Goldberg (Project Leader), M. Green, H. Heckler, Dr. R. Singleton, and E. Van De Riet.

The biographies of Mr. Heckler, Dr. Singleton, and Mr. Van De Riet follow.



Heckler, Clarence H., Jr. - Research Engineer, Devices and Circuits Group,  
Computer Techniques Laboratory

Mr. Heckler served as a radio operator in the U.S. Army from 1943 to 1946. From 1946 to 1948 he was self employed, operating a television repair shop. In 1950 Mr. Heckler completed a three-year course in Electronics at Temple University. From 1950 to 1954 he was a Research Engineer for the Burroughs Corporation at Paoli, Pennsylvania, carrying on applied research on magnetic phenomena as applied to digital applications. In 1954-1955 he was employed by the Magnetic Metals Company, Camden, New Jersey, as a Research Engineer working on the processing of untra-thin magnetic tapes for digital applications. From 1955 until he joined the staff of Stanford Research Institute in 1958, Mr. Heckler was a Development Engineer at the Radio Corporation of America, Camden, New Jersey, working on the application of magnetic phenomena to digital data-handling devices and was project engineer of a program investigating a millimicrosecond memory system.

At Stanford Research Institute his work has involved the application of magnetics to new digital data-handling devices and the application of multi-aperture magnetic elements in an all-magnetic logic system.

Mr. Heckler is an associate member of the Institute of Radio Engineers.

Van De Riet, Edwin K. - Research Engineer, Devices and Circuits Group,  
Computer Techniques Laboratory

Mr. Van De Riet received a Bachelor of Electrical Engineering degree from the University of Minnesota in 1948, specializing in communications, and an M.S. degree in Electrical Engineering, specializing in computers, from the University of California in 1953. In 1953, while at the University of California, he supervised circuit and logical design work on CALDIC. During 1943-1945 he served in the U.S. Navy.

From 1948 to 1951 he was a field engineer with the Schlumberger Well Surveying Corporation of Houston. During 1953-1955 he was a Junior Engineer, later a Senior Engineer, at Marchant Research, Inc., Oakland, doing small-scale digital computer work. In 1946 he was employed by Marchant Calculators, Inc., in Oakland, as a Senior Engineer supervising work on small transistorized digital computers.

In April 1958 Mr. Van De Riet joined the staff of Stanford Research Institute, where he has been working on multi-aperture magnetic devices for computers.

Mr. Van De Riet is a member of the Institute of Radio Engineers, the IRE Professional Group on Electronic Computers, Eta Kappa Nu, and Sigma Xi.

Singleton, Richard C. - Mathematician

Dr. Singleton received both B.S. and M.S. degrees in Electrical Engineering in 1950 from the Massachusetts Institute of Technology. In 1952, he received



the M.B.A. degree from Stanford University Graduate School of Business. He holds also the degree of Ph.D. in mathematical statistics from Stanford University, conferred in 1960. His Ph.D. research was in the field of stochastic models of inventory processes, applying the general theory of Markov processes.

Dr. Singleton has been employed by Stanford Research Institute since January 1952. During this period, he has engaged in operations research studies, in the application of electronic computers to business data processing, and in general consulting in the area of mathematical statistics.

He is a member of a number of professional societies, including the Institute of Radio Engineers, the Operations Research Society of America, the Research Society of America, and Eta Kappa Nu.



## APPENDIX A

### DESIGN FOR NON-OPTIMUM SINGLE-FIELD SUPERIMPOSED CODES

It is of interest to determine the effect on the field size required for a given random response rate, due to restrictions on the number of marks in the field. The following is based on the analysis of C. Mooers given in Zator Bulletin No. 30. Three simplifying approximations are involved: (1) use of an exponential form for the expected value of the punching ratio rather than the algebraic expression, (2) use of a fractional form for random response rate rather than the combinatorial form, and (3) operations with averages of probability distributions rather than with the distributions themselves.

Mooers, recognizing points (1) and (2), presents his results as upper limits. Since the purpose of the following study is to determine only roughly the dependency of field size on marking restrictions, the same approximations will be made. It is hoped that the more rigorous analysis presently being conducted will give a measure to the amount of error due to these approximations.

Assume the following definitions:

M = the number of descriptors composed in every file item

L = the number of descriptors composed in a given interrogation

N = the expected number of marks resulting from the superposition of M descriptors in a file item

S = the expected number of marks resulting from the superposition of L descriptors in an interrogation

$f_d$  = the probability that a given file item will respond to a randomly composed interrogation. " $f_d$ " is known as "the dropping fraction" or "random response rate."

From Mooers (op. cit.),

$$f_d = \left( \frac{G}{F} \right)^S, \quad \text{exact value: } \frac{\binom{G}{S}}{\binom{F}{S}}$$

$$S = F \left( 1 - e^{-\frac{NL}{F}} \right), \quad \text{exact value: } F \left[ 1 - \left( 1 - \frac{N}{F} \right)^L \right]$$

$$\frac{G}{F} = \left( 1 - e^{-\frac{NM}{F}} \right), \quad \text{exact value: } \left[ 1 - \left( 1 - \frac{N}{F} \right)^M \right]$$



It follows that

$$\log f_d = F \left( 1 - e^{-\frac{NL}{F}} \right) \log \left( 1 - e^{-\frac{NM}{F}} \right) .$$

This function is plotted in Figs. A-1 and A-2, for  $L = 4$  and  $M = 10$ , for  $N = 2, 3, 4, 5, 6$ , and for  $F$  from 70 to 80. Figure A-1 shows  $f_d$  vs.  $F$  for various  $N$ , and Fig. A-2 shows  $f_d$  vs.  $N$  for various  $F$ . On Fig. A-1 there is also indication of the regions of different optimum  $N$ .



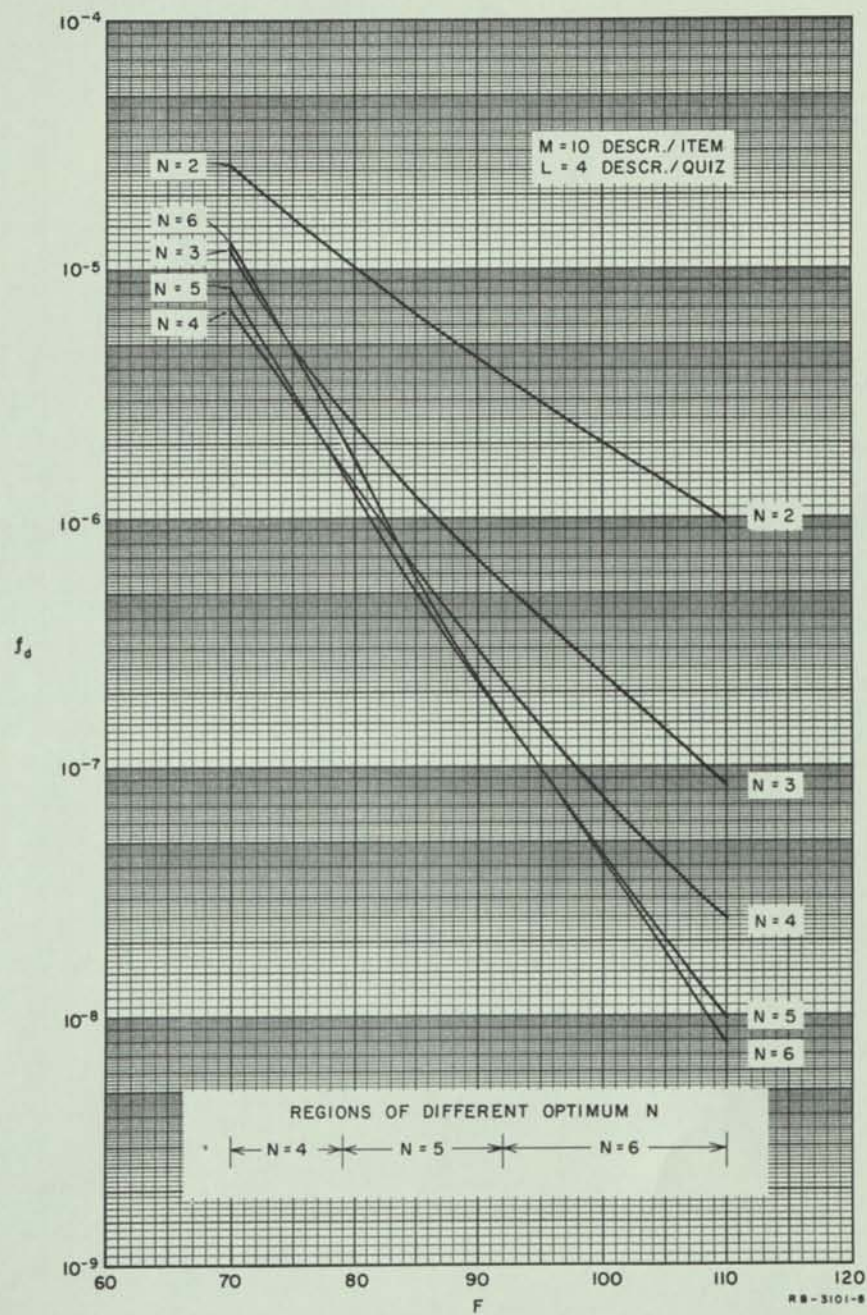


FIG. A-1  
DROPPING FRACTION ( $f_d$ ) VS FIELD LENGTH ( $F$ ) FOR VARIOUS NUMBER  
OF MARKS IN BASIC DESCRIPTOR ( $N$ )



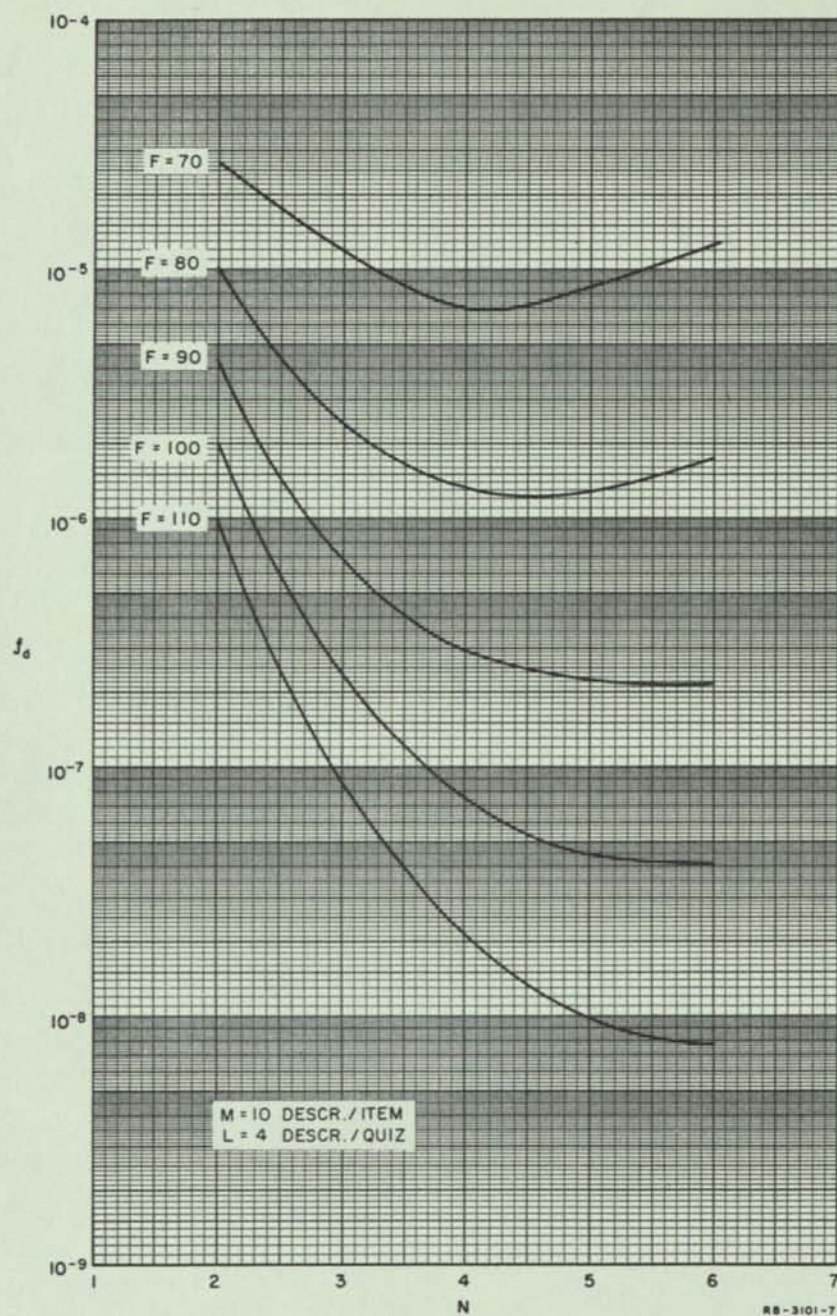


FIG. A-2

DROPPING FRACTION ( $f_d$ ) VS THE NUMBER OF MARKS IN BASIC DESCRIPTOR ( $N$ ) FOR VARIOUS FIELD LENGTHS



## APPENDIX B

### TRANSFORMATION OF A BINARY SUPERIMPOSED FIELD INTO A FIELD BASED ON A HIGHER RADIX

Assume a field designed according to the rules given by C. Mooers, and a file in which all items have the same number of descriptors. If the number of binary places is  $S$ , the maximum number of marks in a field will be  $0.69S$  and the average number of marks will be  $0.5S$ ,

Let the field be divided into  $\frac{S}{n}$  groups of  $n$  bits each, and let the state of each group be newly represented by one mark in a field of  $2^n$  places. The maximum number of marks will now be  $\frac{S}{n}$  and the number of places  $\frac{S}{n} 2^n$ .

$$\text{The number of places added} = \frac{S}{n} 2^n - S$$

$$\text{The maximum number of marks saved} = 0.69S - \frac{S}{n}$$

$$\text{"Cost ratio"} = \frac{\text{places added}}{\text{marks saved}} = \frac{2^n - n}{0.69n - 1} \cdot$$

This ratio is tabulated in Table A-1. The smallest "Cost Ratio," 4.7, occurs for  $n = 3$ .

<u>n</u>	<u>Cost Ratio</u>
1	(no savings)
2	5.3
3	4.7
4	6.8
5	11

TABLE A-1

COST RATIO VS.  $N$  ( $N = \log_2$  OF NEW NUMBER BASE)



## APPENDIX C

### INTRODUCTION TO STATIC MAGNETIC REALIZATION OF MIRF

The purpose of this appendix is to describe some basic principles applying to the realization of a MIRF using static ferromagnetic elements, and to give some simple examples of possible physical structures. All the techniques described assume that the output of the file will be a set of signals representative of the serial numbers of the pertinent items, with, perhaps, the addition of a few tens of bits of supplementary information for each item.

#### 1. Basic Structures

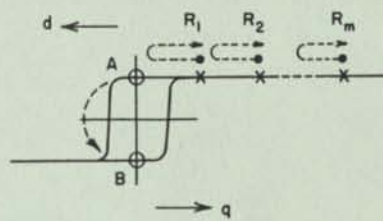
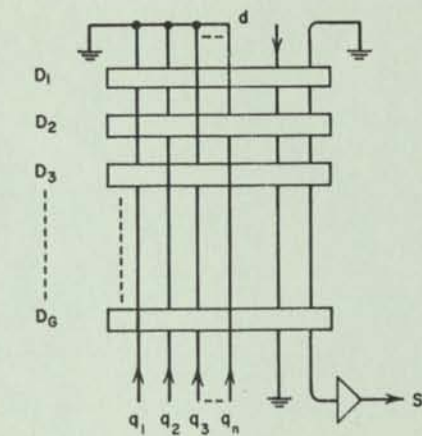
Figure C-1 shows three basic structures. In Fig. C-1(a), one core or other closed magnetic path of square-hysteresis-loop material is used for each file item. The set of cores,  $D_1, D_2$ , etc., is linked by quiz wires marked "q"; the representation of a given data bit consists of linking or non-linking of a wire and a given core. To interrogate the file, currents are driven into selected quiz lines (analogously to the inserting of needles into edge-punched cards); the drive line--marked "d"--is pulsed, and, if a match is present in the set of cores, a signal will appear at the sense amplifier, S. The accompanying hysteresis diagram shows the mode of operation. If a given core is linked by a wire or wires which carry current for a given interrogation, the resulting mmf will drive the core to one of the points on the upper saturation line,  $R_1, R_2$ , etc., depending on the number of linkages. Any core which is not linked by current will remain at A. When the drive pulse, d, is applied to all cores, the cores at points  $R_1$ , etc., shuttle elastically--i.e., without net flux change, but any core at A will be switched to point B, experiencing a net change of flux, and yielding the voltage signal characteristic of such a change.

Figure C-1(b) shows a structure which is, in a sense, dual to that of C-1(a). Here, instead of one mmf signal per quiz-bit, applied to one magnetic path per item, there is one emf signal per quiz bit, applied to one conductive path per item. In particular, the quiz signals are applied by switching quiz cores, q, which induce voltages on item wires, D, depending on the pattern of linkages of wires and cores. If one or more voltages are induced on a given item wire it will cause a unique sense core to switch. The set of sense cores may be driven and sensed, by lines "d" and "S," and if any core is found not to have been switched previously, its switching is evidence of a match.

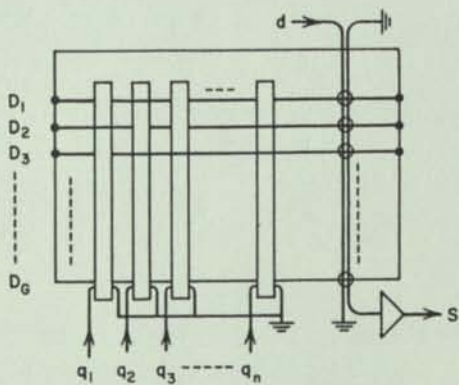
Figure C-1(c) is similar to C-1(b). Here, the representation of each bit is accomplished by a separate switch core whose presence or absence serves to link or isolate an item wire and a quiz wire. Sensing is done as in Fig. C-1(b).

These three schemes may be realized by an infinite variety of physical elements and structures. There are several other classic structures, of perhaps lesser promise for MIRF, which will be described in a later report.

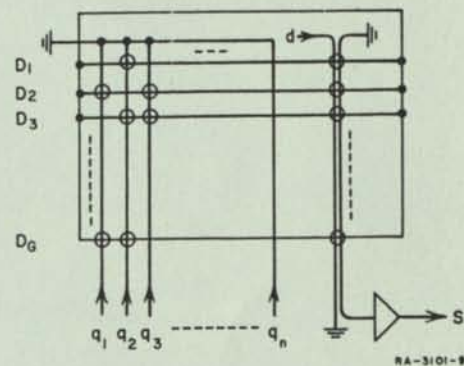




(a)



(b)



(c)

FIG. C-1  
BASIC MAGNETIC MRF STRUCTURES



## 2. Coding

Figure C-2 gives examples of wiring techniques for realizing different codes. The symbols are especially suitable for structures of the kind shown in Fig. C-1(a) above, but the interpretation for other structures is easily seen. A heavy horizontal line represents the edge view of a magnetic core, a vertical line represents a quiz wire, and a short diagonal mark represents the point of linkage of a wire and a core; the letters  $Q_a$ , etc., represent quiz-current sources. The figures show the representation of sample data fields, and the accompanying table shows the rules appropriate to the particular code, for representing a "1," a "0," and an "X" ("don't care") symbol in both the data and quiz fields.

Figure C-2(a) applies to simple Inclusion Coding. Figure C-2(b) is also for Inclusion Coding, but the addition of the special feedback bias wire, linking all cores, causes an inversion of the rules for representation of Data bits.

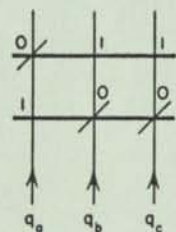
Figure C-2(c) applies to explicit coding in which there are two wires per bit. Application of a feedback wire, shown in Fig. C-2(d), permits reduction to one wire per bit, but at the price of bidirectional current drivers and the need to have two wiring senses (shown on the figure by diagonal marks of opposite sense).

## 3. Some Possible Elements

Figure C-3 shows four examples chosen from a great variety of possible structures. Figure C-3(a) is intended to represent schematically a machine for automatic wiring of toroids according to the data patterns to be stored. The rectangles and heavy segments represent respectively, electromagnets and iron needles arrayed in a line, each representative of a bit in the data field. A core is shown at the right, ready to pass down the line of magnets. As it passes a magnet it may or may not be threaded by the needle, depending upon synchronized signals obtained, say, from a punched paper tape. If a needle is released to link a core, it is mechanically restored to its magnet after the core passes. After the core passes all needles, it is pushed down the bundle of wires forming a chain of cores. Such a machine as described above does not exist, and would have to be developed.

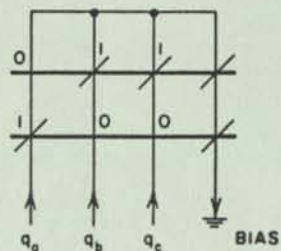
Figures C-3(b), (c), and (d) represent attempts to avoid threading long wires through closed cores. Figure C-3(b) represents two flat metal "U" pieces, with large overlap to minimize the reluctance of the air gap. The wires designed to link a particular element could be gathered separately, then linked by the open pieces, which would be closed upon each other. Figure C-3(c) shows three identical ferrite cores, each with a slot to permit slipping of the wires into the core. After linking, the slots would be displaced, and the three pieces clamped together, forming a single magnetic path. Tests have been made on these elements in the laboratory. The faces of such elements have been ground and polished in order to minimize the air gap, which in this structure, is primarily from face to face, rather than across the large slots. The results have been very promising. Figure C-3(d) represents a very





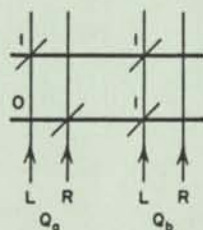
	DATA	QUIZ
I	NO LINK	CURRENT
O, X	LINK	NO CURRENT

(a) SIMPLE INCLUSION CODING



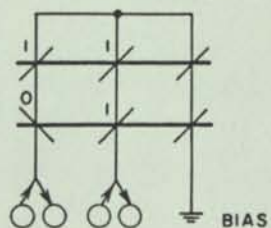
	DATA	QUIZ
I	LINK	CURRENT
O, X	NO LINK	NO CURRENT

(b) INCLUSION CODING WITH BIAS



	DATA LINK	QUIZ CURRENT
I	LEFT	RIGHT
O	RIGHT	LEFT
X	NONE	NONE

(c) EXPLICIT CODING, TWO-WIRE



	DATA	QUIZ
I	FWD	IN
O	REV	OUT
X	—	NONE

(d) EXPLICIT CODING, ONE-WIRE

RA-3101-10

FIG. C-2  
WIRING RULES FOR VARIOUS CODES



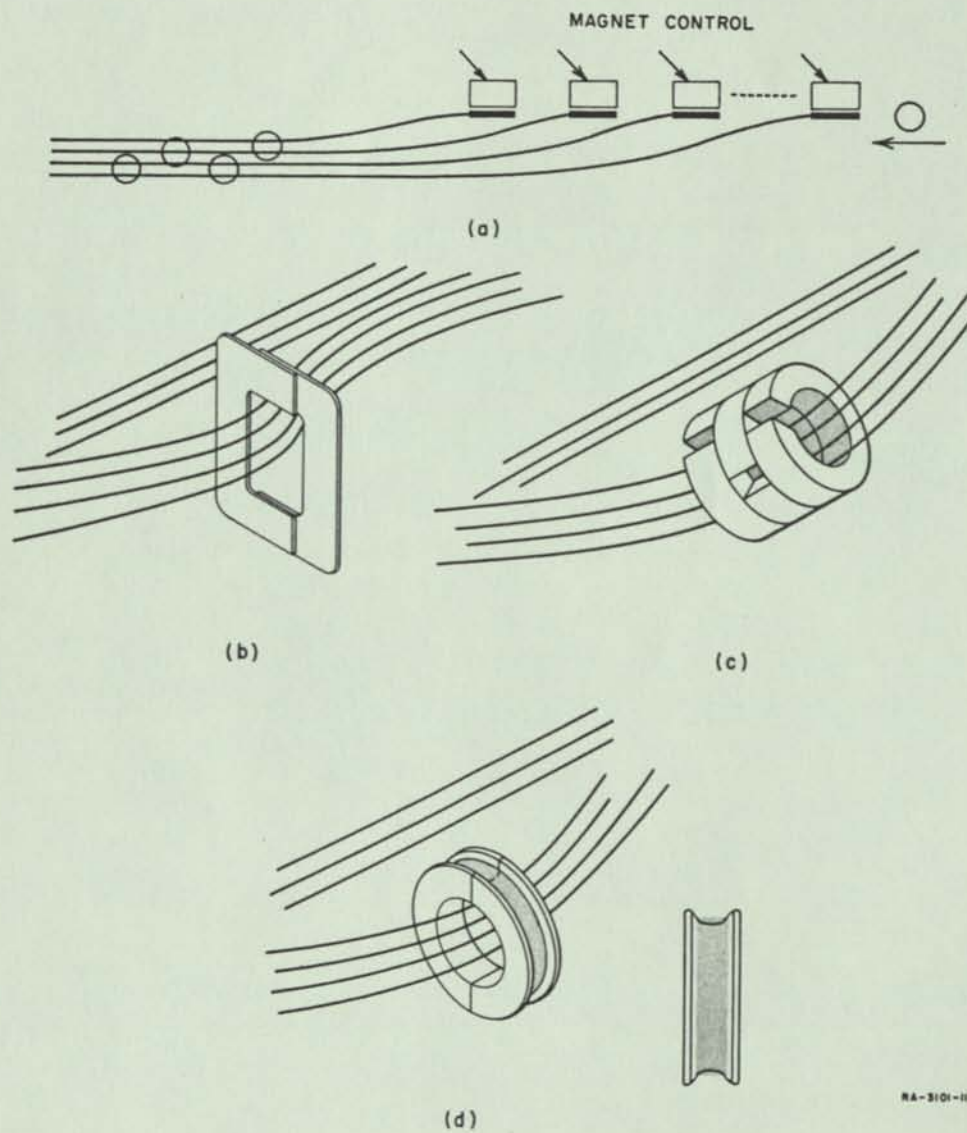


FIG. C-3  
EXAMPLES OF MAGNETIC ELEMENTS



hypothetical structure which is of interest because of its construction principle. The split toroid is of plastic or ceramic material. After assembly, a closed magnetic path would be formed by electro-chemical deposition of iron in the groove provided.

All of the schemes illustrated in Fig. C-3 are attempts to avoid breaking the drive line into segments (later to be joined by soldering, etc.). If this were permitted, many other schemes would result.

Figure C-4 shows a use of Twistors.<sup>6</sup> Vertical lines represent quiz wires placed next to an array of horizontal twistor wires, each representing one file item. At each intersection, according to the data bit stored, there may be placed a permanent magnet which would block the ferromagnetic coupling between twistor and quiz wire at the intersection. Any linkage between a quiz current and an unblocked twistor junction would produce an emf on the twistor line, which would switch the special sense core on each line. All sense cores would then be sensed to determine if any one had not been previously switched. This scheme is clearly an implementation of the basic structure shown in Fig. C-1. Since an engineering analysis of this circuit has not yet been made, the feasibility of the scheme cannot be evaluated at this time; however, it is presented here because of one unique feature not shared by the other schemes described--i.e., the easy changeability of data. At the Bell Laboratories, several economical techniques have been developed for producing coded arrays of permanent magnets on removable cards, films, etc.

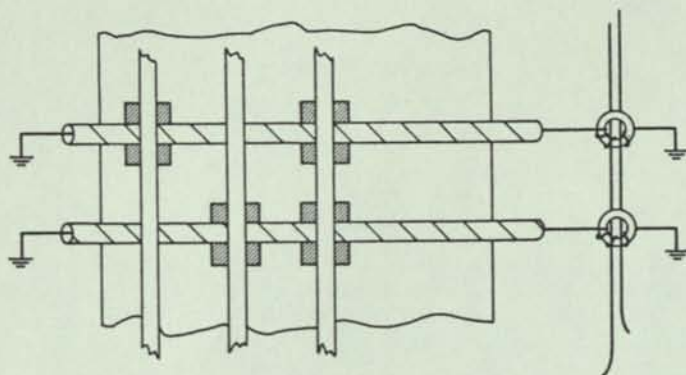
#### 4. File Hierarchies

A file of one million items would no doubt require several sensing lines and several driving lines, for electrical and physical reasons. It would therefore be sensible to make some of the accession number digits coincide with these natural divisions. Figure C-5 illustrates a grouping of the items into a two-dimensional array of sub-groups. The data quiz wires, and a certain number of address interrogation wires are common to all groups, but each group is characterized by a unique pair of (single) drive and sense wires. At the first quiz, all drive wires would be energized. If any sense amplifier responded, the drive wires could be interrogated in smaller subdivisions until a given drive wire found a positive response. In this way, by operations on the drive and sense wires, each group containing a matching item can be quickly isolated. When a group is isolated, the hitherto passive address wires would be interrogated in the manner described in Appendix A of Quarterly Status Report 2, to obtain the individual responses.

---

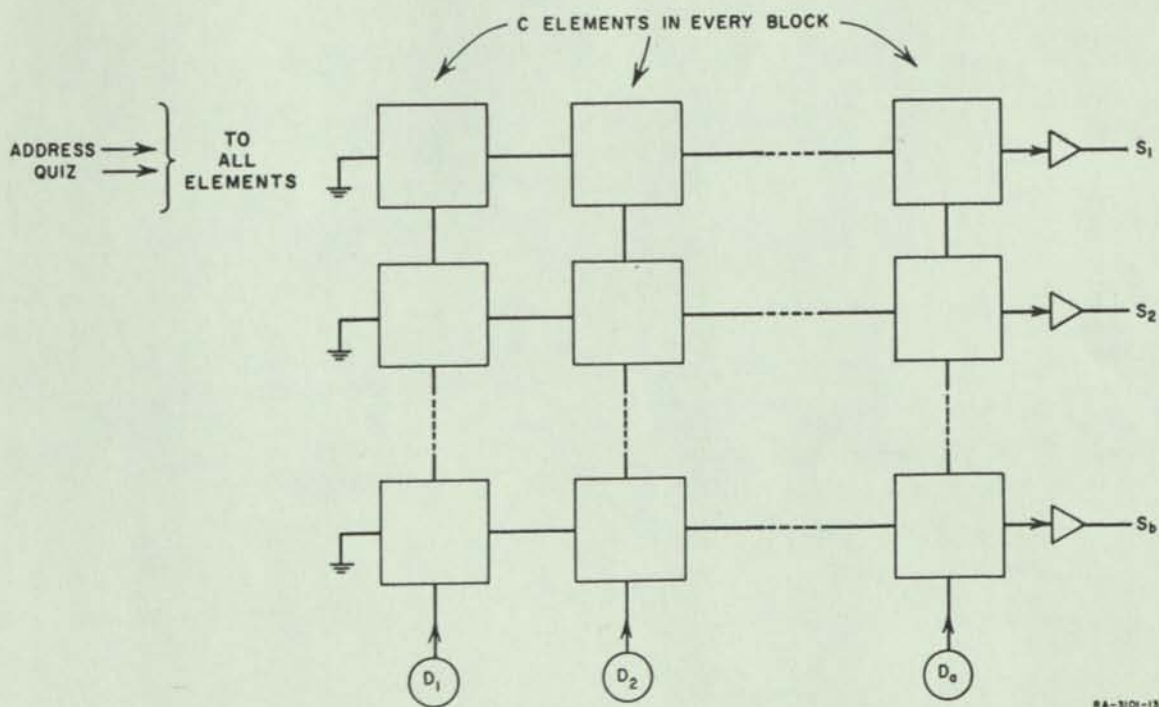
<sup>6</sup>J. J. De Buske, J. Janik, Jr., and B. H. Simons, "A Card Changeable Nondestructive Readout Twistor Store," Proceedings of the Western Joint Computer Conference, San Francisco, California (March 1959).





3101-12

FIG. C-4  
A "TWISTOR" STRUCTURE



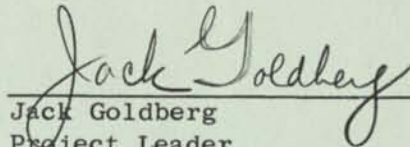
RA-3101-13

FIG. C-5  
SUBDIVISION OF A MAGNETIC MIRF

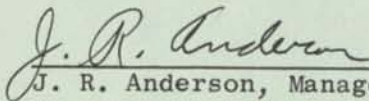


If there are  $a$  drive wires,  $b$  sense wires, and  $c$  items per block,  $abc > 10^6$ , and  $\log_2 C$  address wires will be needed. For example, if  $a = b = 128$ , there will be 64 items per block, and the number of address wires needed is six. This may be compared to the twenty wires needed if there were no subdivision of the file.

Prepared by:

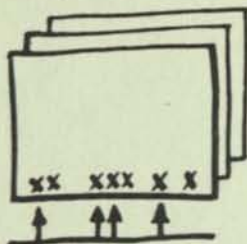
  
\_\_\_\_\_  
Jack Goldberg  
Project Leader

Approved by:

  
\_\_\_\_\_  
J. R. Anderson, Manager  
Computer Techniques Laboratory



*tried for 1 hour 20 min*  
~~DE 13231~~



ZTB-131

# THE APPLICATION OF SIMPLE PATTERN INCLUSION SELECTION TO LARGE-SCALE INFORMATION RETRIEVAL SYSTEMS

Calvin N. Mooers

APRIL 1959

CONTRACT AF 30(602)-1900

PREPARED FOR

ROME AIR DEVELOPMENT CENTER  
AIR RESEARCH AND DEVELOPMENT COMMAND  
UNITED STATES AIR FORCE

GRIFFISS AIR FORCE BASE  
NEW YORK

# ZATOR COMPANY

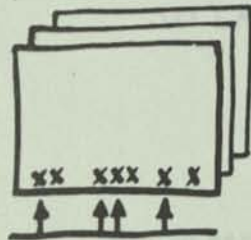
140 1/2 MOUNT AUBURN STREET, CAMBRIDGE 38, MASS.











ZTB-131

## THE APPLICATION OF SIMPLE PATTERN INCLUSION SELECTION TO LARGE-SCALE INFORMATION RETRIEVAL SYSTEMS

Calvin N. Mooers

APRIL 1959

CONTRACT AF 30(602)-1900

PREPARED FOR

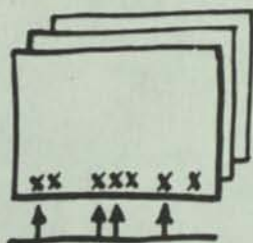
ROME AIR DEVELOPMENT CENTER  
AIR RESEARCH AND DEVELOPMENT COMMAND  
UNITED STATES AIR FORCE

GRIFFISS AIR FORCE BASE  
NEW YORK

# ZATOR COMPANY

140 1/2 MOUNT AUBURN STREET, CAMBRIDGE 38, MASS.





ZTB-131

# THE APPLICATION OF SIMPLE PATTERN INCLUSION SELECTION TO LARGE-SCALE INFORMATION RETRIEVAL SYSTEMS

Calvin N. Mooers

APRIL 1959

CONTRACT AF 30(602)-1900

PREPARED FOR

ROME AIR DEVELOPMENT CENTER  
AIR RESEARCH AND DEVELOPMENT COMMAND  
UNITED STATES AIR FORCE

GRIFFISS AIR FORCE BASE  
NEW YORK

# ZATOR COMPANY

140<sup>1</sup>/<sub>2</sub> MOUNT AUBURN STREET, CAMBRIDGE 38, MASS.



## ABSTRACT

### THE APPLICATION OF SIMPLE PATTERN INCLUSION SELECTION TO LARGE-SCALE INFORMATION RETRIEVAL SYSTEMS

Calvin N. Mooers

Zator Company, Cambridge, Mass., U. S. A.

Simple pattern inclusion selection coding for information retrieval can speed up the serial scanning rate of some present retrieval machines 4 to 25 times. It may also make possible machines with simpler hardware or selective circuits. One present device has a potential retrieval scanning rate of 36 million items per hour with pattern inclusion selection. Set against the advantages of this method of coding are certain peculiarities or limitations: (1) Retrieval prescriptions can be formed by conjoint descriptors only (combined only by AND). (2) The coding method produces a small fraction of extra noise selections. For many retrieval applications, particularly with some very large collections, these peculiarities are of less importance than the gain in scanning speed. The conditions under which simple pattern inclusion selection can be advantageously used are stated, and the code system design rules are given.



## TABLE OF CONTENTS

INTRODUCTION . . . . .	1
DEFINITION OF SIMPLE PATTERN INCLUSION SELECTION . . . . .	3
PROPERTIES OF SIMPLE PATTERN INCLUSION SELECTION CODING . . . . .	6
DESIGNING A PATTERN INCLUSION CODE SYSTEM . . . . .	8
APPLICATION TO SEVERAL EXISTING EQUIPMENTS . . . . .	12
DISCUSSION AND CONCLUSIONS . . . . .	17
APPENDIX . . . . .	19
REFERENCES . . . . .	20



# THE APPLICATION OF SIMPLE PATTERN INCLUSION SELECTION TO LARGE-SCALE INFORMATION RETRIEVAL SYSTEMS

Calvin N. Mooers

## INTRODUCTION

The choice of a coding scheme to be used in an information retrieval system has a profound effect upon the required complexity of the selective apparatus and upon the speeds at which selection can operate. One coding scheme that leads (1) to simple selective apparatus and (2) to high machine speeds is simple pattern inclusion selection. So far as is presently known, simple pattern inclusion selection coding is unique in the enormous advantages it can provide in these two directions.

With simple hand-sorted notched cards, pattern inclusion selection has often been employed because of the pressing necessity to compress as much selective information into as few notched sites on the card as possible. When used to perform such compression, this method of coding may make the difference between a usable and a non-usable card system. The method of coding is also able to greatly simplify and to speed up the clerical or manual operations required for performing selections on various combinations of descriptors. These advantages have been sufficiently great so that many hand-sorted card retrieval systems have used pattern inclusion selection coding during the past twelve years.

With large-scale retrieval machines employing magnetic, optical, and electronic modes of scanning and selection, pattern inclusion selection can sometimes also provide outstanding advantages over other methods of coding. This paper will describe how the scanning rates of some of the present



machines can be speeded up by factors of 4 to 25 times through the use of pattern inclusion selection coding. In view of the advantages that can accrue, it seems strange that the larger machines have so seldom been operated with this type of coding, but there are several possible explanations for this situation.

In my estimation, the first reason why pattern inclusion selection coding has not been used with larger machines is that all too often the machines have been applied to relatively trivial retrieval problems, or to retrieval problems on small collections that do not exert sufficient pressure on the machine capabilities. When the collections are relatively small, or when the requirements for rapid response can be more or less side-stepped, there is little pressure to secure the ultimate in code and machine performance. Consequently, available machines have often appeared to be capable of handling the jobs presented to them. With larger collections, or with demands for much higher response speeds, this situation can be expected to change. We can then expect machine limitations to become more and more bothersome, and all techniques which can aid the machine in its task, such as improved schemes of selection and coding, will be deserving of exploitation.

The second reason why pattern inclusion selection has not been used with the larger machines is that the method has definite peculiarities, which have appeared to some people to be detrimental. However, we should bear in mind that the peculiarities of pattern inclusion selection are a direct consequence of the very features which make this method of coding so advantageous in some respects over other methods of coding. For this reason we should carefully balance the peculiarities of the method against its considerable advantages, remembering that we have to balance advantages and disadvantages with any method of coding, and not only with pattern inclusion selection coding.



The peculiarities of simple pattern inclusion selection are of two kinds. The first peculiarity results from the fact that the code scheme gains its efficiency by being stripped down to certain digital essentials, and in this stripped down form, some of the so-called logical operations are not easily handled. Selection by the concurrence of descriptors (logical AND) is easily handled. On the other hand, selection on the basis of alternation of descriptors (so-called OR), or selection according to the specified omission of descriptors (so-called NOT), is not handled well. Numerical relationships cannot be handled at all. However, it should be emphasized that many retrieval situations (though not all) can get along very well in selection using only the concurrence of descriptors in forming selective prescriptions. It is particularly with such retrieval systems that this paper is concerned.

The second peculiarity of pattern inclusion selection is the small additive "noise" which appears in the selective process. Selections are exact in the sense that documents in the prescribed set are never lost. However, selection is inexact in the sense that a few additional documents not prescribed may also come out of the selective process. This additive selective noise is statistical phenomena, completely under control at the time the code system is designed. The additive noise will always be present (in principle), but the noise can be made as small in amount as desired (i.e. less than one chance in a billion per selection) by suitable design of the codes.

These two peculiarities should not preclude — as they currently seem to — the use of the pattern inclusion selection in large machine systems. In many instances, the advantages of this kind of selection can far outweigh the disadvantages or the unusual features involved in applying the method.

#### DEFINITION OF SIMPLE PATTERN INCLUSION SELECTION

Simple pattern inclusion selection is a mode of serial selection. Code mark carriers called tallies\* (which may be frames of film, cards, or

\* The words underlined are being introduced and defined for purposes of this paper.



portions of a tape), each representing or standing for a document, are moved past a scanning head or reading device. The tallies each bear a coordinate system, scale, or matrix according to which individual code marks are placed upon sites in the tally matrix. Unmarked sites of the matrix are called blanks. The set of marks in the matrix is called a pattern of marks.

The scanning head or device (and associated parts) also has a matrix which contains a selection defining pattern of code marks. This pattern is compared with, or placed in register with, each tally matrix pattern of marks as the tallies go by serially during the scanning or search process.

Unlike most code selection schemes, the criterion for selection is very simple with pattern inclusion selection. The code pattern of marks on a tally matrix is selected or accepted if each and every mark in the matrix with the selection defining pattern corresponds to a marked site in a tally matrix at the instant the two matrices are in register during the scan process. Non-marked sites in the selection defining matrix indicate sites in the tally matrix which are ignored in the selective decision. In Fig. 1, Case A indicates an instance of acceptance while Case B is an instance of rejection.

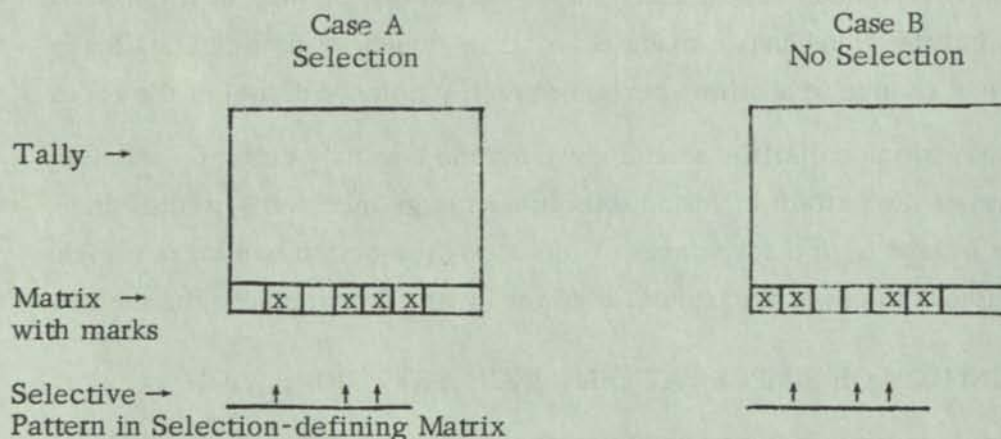


FIGURE 1



Because of the way in which selection is defined, the selection defining matrix pattern of marks must be included within the tally matrix pattern of marks in order for the tally matrix to be accepted. This is the origin of the name of this kind of coding.

Pattern inclusion selection is quite different from exact pattern matching selection, which is the more usual kind of code selection. According to exact pattern matching selection, both the marks and blanks of the selection defining matrix pattern must correspond exactly to the marks and blanks in the matrix (or some subdivision of the matrix) of the accepted tally. Because both marks and blanks must be matched, exact matching generally requires more physical structure or hardware than simple pattern inclusion selection. no -

Information retrieval generally requires more than one descriptor to be associated with any one document. Moreover, prescriptions for retrieval selection generally contain two, three, or more descriptors. To handle these situations, pattern inclusion selection has techniques for indicating a multiplicity of descriptors in any coding matrix. (References 1, 2). In the most elementary application of the scheme, each descriptor is assigned to, or is associated with, a set of N marks scattered across the coding matrix. Each set of N such marks is then a code pattern for a descriptor. The set of all code patterns is originally chosen so that the marks have a uniform incidence on the various sites of the matrix, and so that the patterns have a lack of correlation or similarity with one another. One easy way to attain these requirements is to use a mechanical random process, such as drawing numbered balls from an urn. The patterns for each descriptor are then listed in a coding dictionary, and the listing is not usually changed during the operation of the system.

To represent the several descriptors needed to delineate the subject content of a document in a tally matrix, the patterns of marks for the descriptors



are superimposed in the matrix. By this is meant that the marks for the first descriptor pattern are entered into the matrix, then the marks for the second descriptor pattern are entered, and so on, with the final matrix having any given site marked if any of the individual descriptor patterns have a mark in that site. The rest of the sites in the matrix are left blank. This method of coding the tally matrices is called code pattern superimposition.

The retrieval search pattern is generated in the same way. In other words, the descriptor patterns for the descriptors prescribing the retrieval are combined by superimposition to give the search pattern.

In the search operation, as described previously, a document tally is accepted whenever the selection defining pattern is included in the tally matrix pattern. A moment's reflection will make it clear, from the manner of generating combined patterns by superimposition of marks, that whenever the set of prescribing descriptors is included within the set of document delineating descriptors, then the selective pattern must necessarily be included within the tally matrix pattern.

#### PROPERTIES OF SIMPLE PATTERN INCLUSION SELECTION CODING

Pattern inclusion selection, as defined, has several important manipulative properties. The first is that coding and selection is commutative on the descriptors. Given a selective prescription involving descriptors A and B, the result of a selection is the same whether retrieval is prescribed according to AB or to BA. The reason for this is that the sequence of entering descriptor patterns into a matrix makes no difference to the final resulting superimposed pattern.

The second property is that the code patterns of pattern inclusion selection are strongly nonlocal. (Reference 3) To explain this it is necessary to point out that in some code systems, the code matrix is divided



into submatrices and only one descriptor code pattern is entered in each submatrix. In some of these code systems, the interpretation or meaning of a pattern depends jointly upon the arrangement of marks in the pattern and upon the number of submatrix containing the pattern (i.e., the location of the pattern). Such a code system is called a local code. As was pointed out by Mooers (Reference 4), local codes have severe disadvantages in selection, particularly because they limit the combinations of descriptors that can be employed to delineate a document. Specifically, if two descriptors have code patterns that must be recorded in the same submatrix, then these two descriptors cannot be used together to delineate the same document.

To avoid this restriction, nonlocal code systems were introduced. Nonlocal codes have become very popular in retrieval systems employing such machines as the IBM 101, the Eastman Minicard, the French Filmorex, and others. In a nonlocal code, the coding matrix is divided into compartments or submatrices, each submatrix having the same number of sites. Each submatrix takes a code pattern for a single descriptor. As used with these machines, the code system is nonlocal, because the meaning of a particular descriptor pattern does not depend upon which submatrix it is recorded in. However, for purposes of search, it is impossible to predict in which submatrix the patterns for the retrieval-prescribing descriptors will be found. Consequently for each tally, the search machine must be capable of looking in each of the submatrices of the tally for each descriptor of the prescription. Where there are  $n$  submatrices and  $m$  prescribing descriptors, the machine must make something like  $n \times m$  matching operations, keeping track of the results, in order to decide whether to accept any tally representing one document. The machine must perform this operation for as many tallies as there are documents in the collection. To accomplish  $n \times m$  matching operations, and to combine the results into an accept or reject decision, requires much more machine complexity than



if a single matching operation (or an inclusion selection) can be used, with tally acceptance depending completely upon the single match. So many matching operations per document tally also slows down the overall search speed.

As was previously pointed out, a single pattern inclusion will suffice to give a complete selective decision for any number of conjointly prescribing descriptors when using pattern inclusion selection. In view of the terminology introduced above, this method of coding and selection can be called strongly nonlocal: (1) because the descriptor patterns do not have a meaning dependent upon their location in the matrix — there is only one location, and there are no submatrices; and (2) because the selective machine does not have to hunt within the matrix to find where the code patterns of the several prescribing descriptors may have been recorded.

Besides the great advantages of selective machine simplicity, pattern inclusion code selection has the additional advantage in that it is often able to compress more descriptor code patterns into a tally matrix of a given size than when ordinary coding methods with nonlocal code patterns are used. The reasons behind this compression have been treated in several papers by Mooers. (References 4, 5) It is sufficient to point out here that code compression by a factor of two to five is not unusual. Since the scanning speed depends in a large measure upon the number of recorded bits per document that must be handled, it is clear that if the subject content of a document can be represented by  $\frac{1}{2}$  or  $\frac{1}{5}$  the number of bits otherwise required, the search speeds can go up by corresponding factors of 2 or 5.

#### DESIGNING A PATTERN INCLUSION CODE SYSTEM

In planning for the application of pattern inclusion codes, it is first necessary to determine the values of the parameters which influence the design



of the code system. The first parameter is the size  $C$  of the collection, in terms of the number of documents or items that must be discriminated. For most serious applications,  $C$  will be at least  $10^4$  items.

The next most relevant parameter is the "reasonable least" number of prescribing descriptors. The term "reasonable least" needs explanation. Retrieval prescriptions employ a variable number of descriptors. Prescriptions seldom, if ever, employ only one descriptor to prescribe a search, since a single descriptor would retrieve upon such a broad classification that an appreciable part of the file (e.g. 1 to 10 per cent of the file) would be retrieved. More generally, three or four descriptors might be used. If the anticipated use of the collection is such that a reasonable search would seldom use fewer than  $L$  descriptors in a prescription, then  $L$  is the reasonable least number of prescribing descriptors.

Typical information items require six to ten or more descriptors in their delineation. Experience in the use of a collection will reveal the number of delineating descriptors ordinarily required. If we exclude the very unusual items which have an unusually high number of delineating descriptors, then there will be some upper limit, a "reasonable most" number  $M$  of delineating descriptors that are needed for delineation. As to the unusual items, it is generally possible, such as by dividing the items into more than one part, to fit all cases into the limits imposed by the number  $M$ .

As a final parameter, the tolerable noise level in selection must be specified. The number  $E$  of extra tallies that will appear during selection upon any prescription is in proportion to the number of tallies scanned i.e. is proportional to the number  $C$  of items in the collection. The convenient manner of expressing the noise is by the (maximum) average noise level ratio  $R$  when the prescription contains only  $L$  descriptors. Where the (average) maximum number of noise tallies selected by  $L$  de-



scriptors is  $E_{\max}$ , then  $R = E_{\max}/C$ . If a collection has 100,000 items, a tolerable noise level ratio  $R$  would be  $10^{-4}$ , and under the noisiest of allowed prescriptions (that is, with prescriptions using only  $L$  descriptors) a selection would produce an average of only 10 extra noise tallies scattered among the many prescribed tallies. If the ratio  $R$  were to be set at  $10^{-5}$ , the average worst selection for the same case would produce only one extra tally. In most retrieval applications, the appearance of a few extra noise tallies is no inconvenience since they are only one of the several kinds of selected tallies that have to be discarded upon closer examination of the selected items. In any case, the maximum average number of noise items  $E_{\max}$  is under control during the code design period, depending upon the value chosen for  $R$ . However,  $R$  cannot be decreased indefinitely, because the only way we have to force the noise level down is to use codes with more marks. Such codes require more coding space, and this slows down the selection speed. Therefore some balance must be set between the speed of selection and the convenience of very low noise. We cannot ask that both advantages be maximized.

The parameters which have been defined are:

- $C$  the number of items distinguished in the collection
- $L$  the reasonable least number of descriptors in a prescription
- $M$  the reasonable most number of descriptors delineating an item
- $E_{\max}$  the (average) maximum number of noise selections with  $L$  prescribing descriptors
- $R$  the noise ratio  $= E_{\max}/C$ .

In terms of these parameters, each descriptor code pattern should contain  $N$  marks where (See Appendix and Ref. 1)

$$N = (\text{integer}) (1/L) (-\log_2 R) \\ = (\text{integer}) (1/L) (3.31) (-\log_{10} R)$$

and where (integer) indicates that the nearest integral value is to be



taken for following product. Then the least number  $S$  of sites that the coding matrix must have in order to contain  $M$  descriptors is

$$S = (\text{integer}) (1.445) NM.$$

It should be specifically noted that neither  $N$  nor  $S$  depends upon the size of the vocabulary of the descriptors, i.e. the number of different descriptors used in the retrieval system.

These formulas will be illustrated by working through two typical examples.

EXAMPLE 1. Let:

$$C = 24,000 \text{ items}$$

$$L = 3$$

$$M = 35$$

$$R = 1/50,000 = 2 \times 10^{-5},$$

then

$$N = (\text{integer}) (1/L) (3.31) (-\log_{10} 1/50,000)$$

$$= (\text{integer}) (0.333) (3.31) (+4,699)$$

$$= (\text{integer}) (5.18) = 5 \text{ marks per pattern}$$

and

$$S = (\text{integer}) (1.445)(5) (35)$$

$$= (\text{integer}) (253)$$

$$= 253 \text{ sites per tally matrix.}$$

The maximum average number of noise selections, for prescriptions of only  $L = 3$  descriptors, is  $E_{\text{max.}} = RC = \frac{24,000}{50,000} = 0.48$  tallies per search. In other words, many selections will produce no extra selections, and less than half will produce only one.

EXAMPLE 2. Let:

$$C = 1,000,000$$

$$L = 3$$

$$M = 20$$

$$E_{\text{max.}} = 50 = RC,$$



then  $R = E_{\max.}/C = 5 \times 10^{-5}$ ,  $N = 5$ , and  $S = 145$ .

We see in the second of these two examples, by relaxing the requirements on the noise ratio  $R$  (allowing larger  $E_{\max.}$ ), and by allowing fewer descriptors in a maximal delineation (as indicated by  $M$ ), that the size  $S$  of the required matrix for a collection of a million items may be less than for a collection of only 24,000 items. This shows the importance of carefully designing a pattern inclusion code system.

#### APPLICATION TO SEVERAL EXISTING EQUIPMENTS

##### A. Document Data Index Set, AN/GSQ-26, Computer Controls Company.

This machine is a high-speed magnetic tape scanning device. As presently organized, it uses nonlocal codes, usually with a tally matrix of 840 sites divided into 20 submatrices of 42 sites each. One submatrix holds one descriptor. Selection is according to "Minicard logic". That is, selection is defined by conjunction, alternation, and non-occurrence (so-called AND, OR, and NOT) of the prescribing descriptors, with provision for "phrase grouping" of descriptors within items. One tally matrix represents one document or item. The machine is able to scan 260 matrices of 840 sites per second, giving a scanning rate of  $2.18 \times 10^5$  sites or bits per second.

According to the builders of this machine, it would not be difficult to make the machine capable of pattern inclusion selection at this scanning rate. In this mode of selection, the tally matrix size of 840 sites is considerably larger than is needed for typical retrieval problems, as the preceding examples indicate.

The following discussion will be based upon the use of a tally matrix of  $S = 200$  sites, which as Example 2 shows, is adequate for some (but not all) retrieval problems on collections of 1,000,000 or more items. For the exceptional problems, other values of  $S$  would be used.



With a tally matrix of 200 sites, instead of 840, the scanning rate of the AN/GSQ - 26 can be speeded up by the factor of  $840/200 = 4.2$  times. In other words, the item scanning rate could be increased from 260 to  $4.2 \times 260 = 1092$  items per second, or 3.9 million items per hour. At the same time, the electronic circuitry required to perform the selection might substantially be simplified.

With simple pattern inclusion selection, all selective prescriptions would have to be in terms of the conjunction of the prescribing descriptors. The output of the selection, as at present with the AN/GSQ - 26, would be either a frame number of a tally matrix, or a document number. Depending upon the number of pattern inclusion circuits incorporated into the selection mechanism, more than one search prescription could be handled simultaneously.

B. Computer Set, General Information Data, AN/GSQ - 18(xw-1) "Magnacard", The Magnavox Corp.

The Magnacard machines as under development represent an integrated system for dealing with information on magnetic cards. An individual card can hold up to 5,000 bits of information, and these cards can be released from a storage magazine, run past a reading or scanning head, and stacked again in a second magazine at the rate of 100 cards per second. If the recorded information on any card trips the selective circuits, this card can be picked out on the fly and sent to a third magazine for further treatment. Alternatively, the information content of the selected card can be read out on the fly.

The usual manner for using such a card handling information system is to delegate one card to each document or information item. Part of the card (e.g. 1000 bits) might be used to record the coded descriptors delineating the items, while the rest of the card would contain alphanumeric text of various sorts, including document numbers, titles and the like. In this



manner of using the magnetic cards, the maximum scanning rate is only 100 cards or items per second.

The scanning function of retrieval can often be separated from the independent function of text storage. The cards need contain very little more than item delineating codes. With the scanning operation loaded down full time in processing only the delineative codes, the item scanning rate will go up. As we have seen, a coding matrix of 200 sites per item can handle relatively large collections. Twenty five matrices of 200 sites each can be placed on a Magnacard using pattern inclusion coding and selections. Used in this way, the Magnacard machine (used as a scanning device) would operate almost as a tape scanner, but its speed would be very high. Codes for 2500 documents can be scanned per second, or 9 million items can be scanned per hour. As selections occurred, either the document numbers would be read from the cards on the fly, or the cards would be diverted into another magazine to be read later. In any event, the speed-up in item scanning rate made possible with pattern inclusion selection is of the order of 25 times as compared to the conventional manner of using unit cards of this type.

Of course the speed-up would be somewhat smaller if conventional codes could be used, and the codes for several documents could be placed on the same card. Assuming that 1,000 bits would be adequate to delineate a document, the codes for 5 documents could be placed on a card, giving a scanning rate of 500 items per second. On such an assumption, simple pattern inclusion selection would give a speed advantage of only five times over conventional codes, but it would still give the advantage of simplicity of selective circuits.

The computation of the speed-up with multiple items conventionally coded on a card is no simple matter, and the assumptions made above depend upon such things as descriptor vocabulary size and number of delineating descriptors. These matters are discussed in detail in Reference 5.



This high scanning rate presumes that the Magnacard magazines can be brought to the scanning machine at the rate of two per minute, either manually or by use of a large-scale automatic magazine storage device such as the Magnacard File Block. This rate also presumes that the cards do not have to be run back into the same magazine after they have been scanned; to do so would cut the speed by a factor of two. By using double sets of magazines, added run back time is easily avoided.

C. Document Data Processing Central, AN/GSQ - 11 "Minicard", Eastman Kodak Co.

The Minicard machines compose an integrated system for dealing with graphic and digital information recorded photographically on miniature cards, the cards being uniformly cut short snips of unperforated 16 mm film.

At its maximum digital capacity, a Minicard can hold 2772 bits. At present, the Minicard sorting and selecting machine is passing these cards at the rate of 16 cards per second. As presently employed, one document or item is handled on one Minicard, though it should be made clear that by the process of subdividing the nonlocal codes used into phrase groups, one card dealing with one document can effectively deal with a number of different information items which would otherwise require separate cards. Presuming that phrase group subdivision is fully exploited, it is slightly confusing to try to specify the maximum number of items that could be recorded on a Minicard. However, on the basis of earlier work by Mooers (References 4, 5) it is clear that the nonlocal codes as used in Minicard do not approach the efficiency in use of matrix sites of superimposed codes with pattern inclusion selection.

On the basis of 200 sites per item, using pattern inclusion selection codes, a Minicard can receive the codes for 13 items. This provides a



potential scanning rate of  $13 \times 16 = 208$  items per second, or  $\frac{3}{4}$  million per hour. The potential increase in scanning rate is between 1 and 13, depending upon the way subdivision into phrase groups is used to place more than one item on a card.

#### D. The Ampex Videotape Recorder, Ampex Corporation

Each of the preceding devices has been discussed with respect to its ability rapidly to scan a digital record. It was pointed out that pattern inclusion selection usually minimizes the circuits or hardware needed to make a selective decision according to this record. In each of the machines considered, the factor limiting the overall scanning rate was the speed at which the recorded digits could be passed under the reading head.

Videotape Recorders were developed for recording television video material on magnetic tape. For this purpose they have a nominal 5-megacycle band width. In recent discussion with an Ampex engineer, I learned that engineers at Ampex have been reasonably successful in recording and reading out digital information at a rate of approximately 2 million bits per second. Taking this rate as a basis for discussion, it is evident that by using a matrix of 200 sites per item, some  $10^4$  items could be scanned per second, or 36 million items per hour. A collection of one million items could be scanned in 100 seconds.

The preceding results, for the various devices mentioned with  $S = 200$  bits per item, are summarized in the following tabulation:

<u>Device</u>	<u>Estimated Speed-up</u>	<u>Items Scanned Per Hour</u>	<u>Minutes Needed to Scan One Million Items</u>
CCC Tape Scanner	4.2 times	$3.9 \times 10^6$ items/hr.	15.4 minutes/ $10^6$ items
Magnacard	5 to 25	$9.0 \times 10^6$	6.7
Minicard	1 to 13	$0.75 \times 10^6$	80.0
Video Tape	xx	$36 \times 10^6$	1.7



DISCUSSION AND CONCLUSIONS

It has been shown that pattern inclusion selection can substantially speed up retrieval scanning rates and can lead to simpler hardware, provided certain determining conditions for its use are true of the overall retrieval situation. Because these conditions are important, and because pattern inclusion selection is no panacea for all the problems of retrieval, these conditions of application will be summarized.

1. The first condition to be satisfied is that a high scanning speed, with only a document or file number output, be more desirable than a slower scanning speed, but with a more complete alphanumeric or graphic output. This first condition will be especially true with very large collections, whose nature requires total scan, with searches that will typically produce a relatively small number of desired output items. An example would be a collection of 10 or 100 million items, with a usual search producing 100 or fewer useful items. In such a case, a very high scanning rate is of paramount importance, while detailed and immediate alphanumeric output is of low importance.

2. The second condition to be satisfied is that the retrieval questions can be sufficiently well expressed in terms solely of the conjunction of a set of descriptors (the AND combination). For a great many retrieval situations, retrieval questions can be so framed, despite much popular opinion to the contrary. Since it is not the purpose of this paper to argue retrieval logic, the reader is referred to two other papers by Mooers (References 6, 7) which discuss aspects of this point. When the second condition is too stringent, slightly more complicated methods of pattern inclusion selection can be employed. Some more elaborate methods of this kind will be described in a succeeding paper in this series.



3. The third condition to be satisfied is that whatever machine is to be employed, this machine must have an adequately high rate of digital scan. The reason behind this condition is that the advantages secured by using simple pattern inclusion selection are largely proportional to the scanning rate of the machine employed. For example, in retrieval situations satisfying conditions 1 and 2, the Magnacard machine would gain by use of simple pattern inclusion selection. On the other hand, simple pattern inclusion selection coding would not be a particularly suitable or advantageous mode of coding in most applications of the Minicard machines. This machine is more advantageously used in other ways with other kinds of coding.

In conclusion, simple pattern inclusion selection is applicable and is advantageous in certain retrieval situations. Where it is applicable, it may permit simpler hardware, or it may lead to higher scanning rates, or both. It has several subtle differences or peculiarities as compared to the more often used codes and these peculiarities must be taken into account. Decision on whether or not simple pattern inclusion selection should be used in a specific instance generally involves a number of interrelated factors. It is probable, for very large collections, that pattern inclusion selection will be a most useful technique. Where the logical or combinatorial capabilities of simple pattern inclusion selection are too restricting, several extensions of the method (to be described in a following paper) can provide greater versatility. In any event, pattern inclusion selection coding methods will not displace other coding methods; each coding method should be used in the situation where it is the most advantageous and appropriate.

\* \* \*

13 April 1959

ZTB 131



## APPENDIX

Let  $C$ ,  $L$ ,  $E_{\max}$ ,  $R$ ,  $S$ , and  $N$  be defined as on page 10. Then as discussed in Ref. 1, the noise ratio is

$$R = E_{\max}/C = (1/2)^X$$

where  $X$  is the total number of marks in the selective pattern. If each selective descriptor has  $N$  marks, the worst selectivity, or the highest noise, occurs when  $X = LN$ . The noise due to this worst case must be made acceptable. In other words, for a given  $R$  and  $L$ , an appropriate descriptor pattern length  $N$  must be found which will give an acceptably low noise. This is accomplished as follows. Since

$$R = (1/2)^X = (1/2)^{LN},$$

we solve for  $N$ ,

$$\log_2 R = -LN$$

$$N = (1/L)(-\log_2 R) = (1/L)(\log_2 10)(-\log_{10} R).$$

giving:  $N = (1/L)(3.31)(-\log_{10} R).$

This value is not generally integral, while  $N$  must be integral. Therefore, the "(integer)" is prefixed to these expressions to indicate that the nearest integer is to be taken.

In Ref. 1 the optimum use of a coding matrix was shown to occur when the numerical sum of all the pattern marks of the descriptor code patterns recorded in the matrix came to 69 percent of the number of sites in the matrix. Since  $M$  is the "reasonable most" number of descriptors of  $N$  marks each used in a matrix, the number of sites in the matrix must be set by

$$MN = 0.69S$$

or  $S = (\text{integer})(1.445)MN.$



## REFERENCES

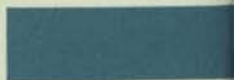
1. Mooers, C. N., "Zatocoding Applied to Mechanical Organization of Knowledge," American Documentation, v. 2, pp. 20-32 (Jan. 1951).
2. Mooers, C. N., "Zatocoding and Developments in Information Retrieval," ASLIB Proceedings v. 8, pp. 3-22 (Feb. 1956).
3. The use of the terminology "local" and "non-local" is due to V. P. Cherenin and B. M. Rakov, in their pamphlet "Experimental Information Machine of the Institute of Scientific Information of the U.S.S.R. Academy of Sciences," (in English), Moscow, 1955. The concepts involved are much older.
4. Mooers, C. N., "Zatocoding for Punched Cards," Zator Technical Bulletin No. 30, Boston, Zator Company (1950).
5. Mooers, C. N., "Choice and Coding in Information Retrieval Systems," Transactions, Institute of Radio Engineers Professional Group on Information Theory PGIT-4, pp. 112-118 (Sept. 1954).
6. Mooers, C. N., "A Mathematical Theory of Language Symbols in Retrieval," Proceedings of the International Conference on Scientific Information, Washington, National Academy of Sciences, (1958) (in publication).
7. Mooers, C. N., "Some Mathematical Fundamentals of the Use of Symbols in Information Retrieval," Proceedings of the First International Conference on Information Handling, Paris, UNESCO, (1959) (in publication). RADC-TN-59-133, ASTIA Document No. AD-213 782.

\*

\*

\*







mean given by Wire:  $G = H - H \left( \frac{H-1}{H} \right)^x$   $x = \text{no. descriptors}$   
 (or ones)  
 Distribution of Total No. of Marks in the Single Impermired Field  
 Dropping Fraction  $\leftarrow F \rightarrow$

$P\{\text{that a file entry with exactly } i \text{ ones in its composite description will be selected by a search with exactly } j \text{ ones in the composite search description}\}$   
 assume: each of the  $\binom{F}{i}$  possible patterns of  $i$  ones equally likely.

$$= \frac{\binom{i}{j}}{\binom{F}{j}} \quad \text{for } 0 \leq j \leq i \quad \text{and } N \leq i \leq F$$

$$= 0 \quad \text{otherwise}$$

(2) Any single descriptor has exactly  $N$  works. If any other descriptor is added to the first descriptor to form a composite descriptor, the number of works in the composite figure is not known with certainty. However, the <sup>probability</sup> distribution for the no. of works can be computed. (By two different methods, under 2 diff. assumptions.)



$$\overleftarrow{\quad RC \quad} \overrightarrow{\quad}$$

Strain

● Expected number (i.e. average) of holes:  $G_e = RC - RC \left(1 - \frac{1}{RC}\right)^X$

Wise

$$\overleftarrow{\quad H \quad} \overrightarrow{\quad}$$

$X = \text{no. of subjects} = \text{no. descriptors}$

~~$(G) \text{ no. positions filled} = H - H \left(\frac{H-1}{H}\right)^X$~~

Wise: 
$$F_d = \frac{\frac{G!}{Y! (G-Y)!}}{\frac{H!}{Y! (H-Y)!}} = \frac{\binom{G}{Y}}{\binom{H}{Y}}$$

● Movers: 
$$F_d = \left(\frac{G}{H}\right)^Y$$



$$D \left( \frac{H}{N-H} \right) H - H = 0$$

$$D \left( \frac{H}{N-H} \right) - 1 = \frac{H}{0}$$

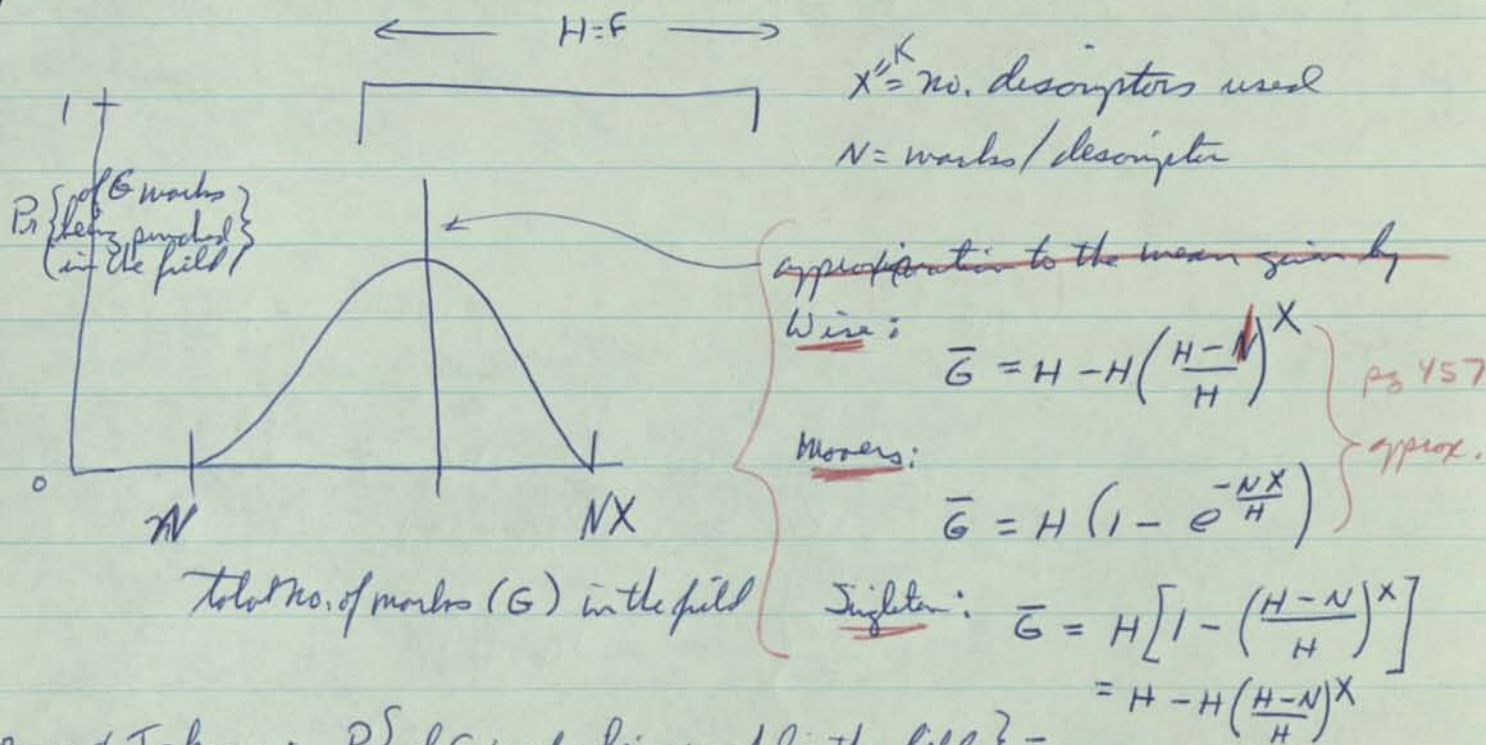
$$\left( \frac{H}{N-H} - 1 \right) H =$$

$$\left( \frac{H}{N-H} - 1 \right) H = 0$$

$$\frac{H}{N-H} - 1 = \frac{H}{0}$$



# Distribution of the total number of marks in the Superimposed Field



Ross & Takacs:  $P\{ \text{of } G \text{ marks being punched in the field} \} =$   
 (sample w/ replacement)

Singleton:  
 (sample w/ replacement)



## Dropping Fraction

Moores:  $F_d = \left(\frac{G}{H}\right)^Y$  upper bound

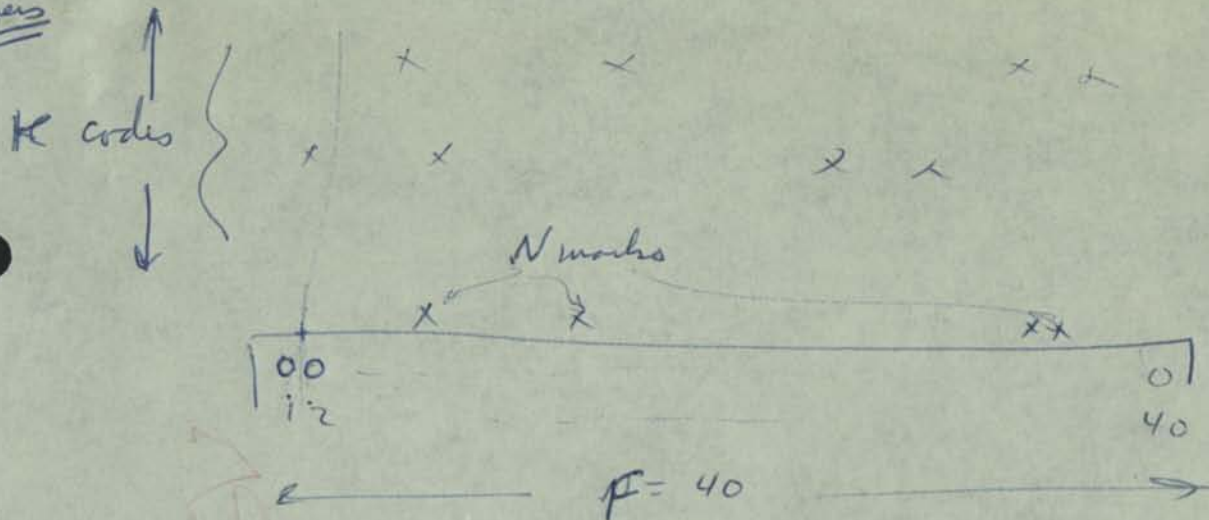
$G = \text{avg. no. holes per word}$   
 $Y = \text{no. ~~holes~~ descriptors}$

Wise:  $F_d = \frac{\binom{G}{Y}}{\binom{H}{Y}}$  approximation

Def  $F_d \equiv$  chance that a sort will cause a card taken at random to drop.



Moore's



$$P\{\text{any one position has a mark}\} = \frac{N}{F} = p$$

K=1 code punched

$$P\{\text{hole 1 has a mark}\} = \frac{N}{F}$$

$$P\{\text{hole 1 has no mark}\} = 1 - \frac{N}{F}$$

K=2 codes punched

$$P\{\text{hole 1 has a mark}\} = 1 - P\{\text{hole 1 has no marks}\} = 1 - \left(1 - \frac{N}{F}\right)^2 = \frac{N}{F} + \frac{N^2}{F^2}$$

binomial expansion of  $(q+p)^K$

$M$  = multiplicity of marks  
ie) overlap

$$M=0$$

$$(1-p)^K$$

=  $P\{\text{NO marks}\}$   
ie) no overlap since only 1 code is punched.

$$M=1$$

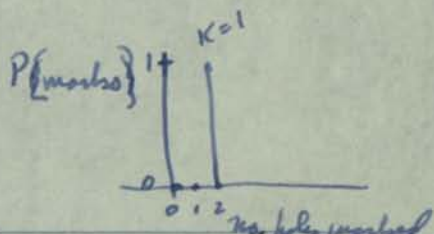
$$K p (1-p)^{K-1}$$

=  $P\{\text{1 punch}\}$

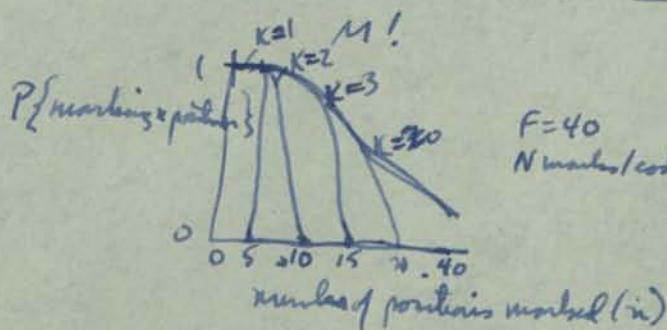
$$M=2$$

$$\frac{K(K-1)}{2} p^2 (1-p)^{K-2}$$

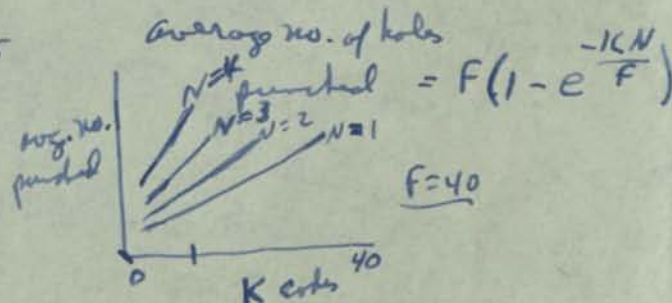
=  $P\{\text{2 punches}\}$



$$P(M) = \frac{e^{-Kp} (Kp)^M}{M!} = \frac{e^{-\frac{KN}{F}} \left(\frac{KN}{F}\right)^M}{M!}$$

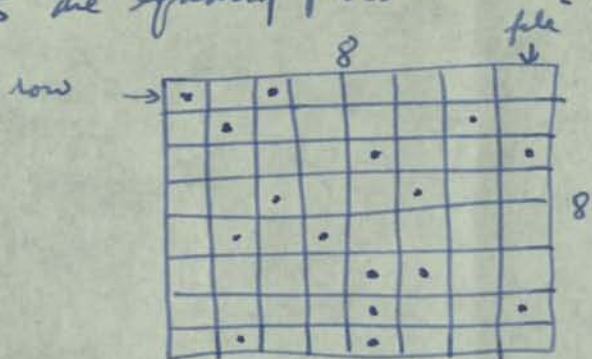


$F=40$   
 $N \text{ marks/code} = 5$





4. Two pawns are placed on each row of a chess board in such a way that all arrangements are equally probable.  $P\{\text{exactly } K \text{ files are empty}\} = ?$



$\binom{8}{2}$  possible arrangements for each row

As a check, it should be noted (by inspection) that the probability = 0 for  $K = 7, 8$ .

The probability that one specified square<sup>(file)</sup> will be blank in a given row, is given by  

$$P\{\text{one specified square in a row} = \text{blank}\} = \frac{\text{number of ways to distribute the pawns in the remaining squares}}{\text{total number of ways to distribute the pawns}}$$

$$= \frac{\binom{7}{2}}{\binom{8}{2}} = \frac{6}{8}$$

The probability that all the rows have a blank in the same specified square =  $\left(\frac{6}{8}\right)^8$   
 Since we don't care where the specified file is located, we have answered the question

$K=1$   $P\{\text{one file is empty}\} = \left(\frac{6}{8}\right)^8$

there is a slight possibility that there may be another blank file in the remaining 7.  
 thus we have actually computed  $P\{\text{at least one file is empty}\}$ .

$K=2$   $P\{2 \text{ specified squares in a row are blank}\} = \frac{\binom{6}{2}}{\binom{8}{2}} = \frac{15}{28}$

$P\{2 \text{ files are empty}\} = \left(\frac{15}{28}\right)^8$

$K=3$   $P\{3 \text{ specified squares in a row are blank}\} = \frac{\binom{5}{2}}{\binom{8}{2}} = \frac{5}{14}$

$P\{3 \text{ files are empty}\} = \left(\frac{5}{14}\right)^8$

$K=4$   $P\{4 \text{ blank squares in a row}\} = \frac{\binom{4}{2}}{\binom{8}{2}} = \frac{3}{14}$

$P\{4 \text{ files empty}\} = \left(\frac{3}{14}\right)^8$

$K=5$   $P\{5 \text{ blank squares in a row}\} = \frac{\binom{3}{2}}{\binom{8}{2}} = \frac{3}{28}$

$P\{5 \text{ files empty}\} = \left(\frac{3}{28}\right)^8$

$K=6$   $P\{6 \text{ blank squares in a row}\} = \frac{\binom{2}{2}}{\binom{8}{2}} = \frac{1}{28}$

$P\{6 \text{ files empty}\} = \left(\frac{1}{28}\right)^8$

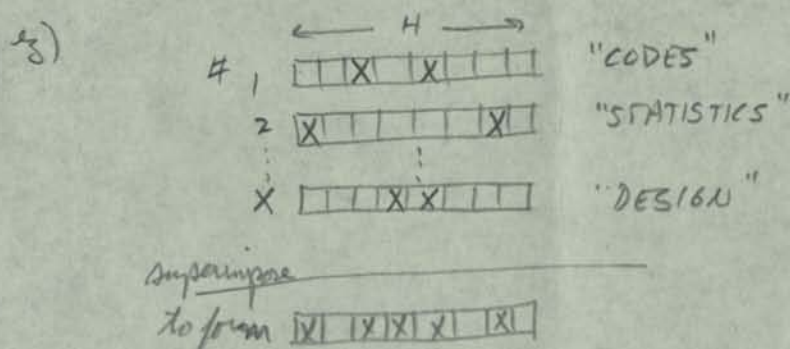
$K=7, 8$   $P=0$

$$P_K = \sum_{r=K}^6 (-1)^{r-K} \binom{r}{K} \binom{8}{r} \left[ \frac{\binom{8-r}{2}}{\binom{8}{2}} \right]^8$$

$$P_K = \sum_{r=K}^6 (-1)^{r-K} \binom{r}{K} \binom{8}{r} \left[ \frac{\binom{8-r}{2}}{\binom{8}{2}} \right]^8$$



This exercise is similar to the problem of designing superimposed codes for the library applications. Each <sup>unique</sup> subject word is represented by a random assignment of marks in a coding field (e.g. 2 marks in 8 spaces). The first representation of indexing information is the superimposition of the patterns of the subject words into a single field.



For this case, the average number of marks in the superimposed code is given by

$$G = H - H \left( \frac{H-Y}{H} \right)^X \quad , Y = \text{no. marks/word}$$

For the checkerboard case, the average number of marked files is

$$G = 8 - 8 \left( \frac{8-2}{8} \right)^8 = 7.2 \quad \frac{5}{10}$$

This is also similar to the classical occupancy problem. For example, with a uniform distribution of  $R$  things in  $n$  cells, the

$$P_m = P\{\text{exactly } m \text{ cells are empty}\} = \binom{n}{m} \sum_{v=0}^{n-m} (-1)^v \binom{n-m}{v} \left(1 - \frac{m+v}{n}\right)^R$$

(ref. Feller, 1<sup>st</sup> ed., pg 69)

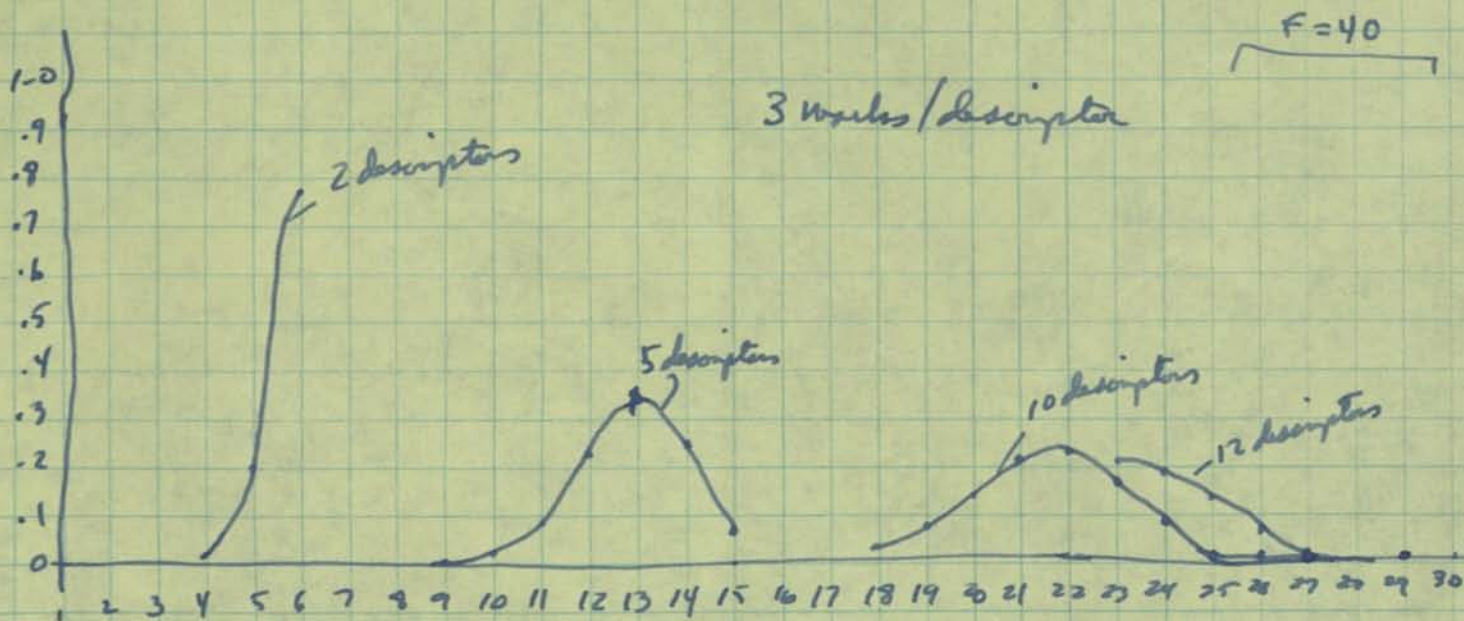
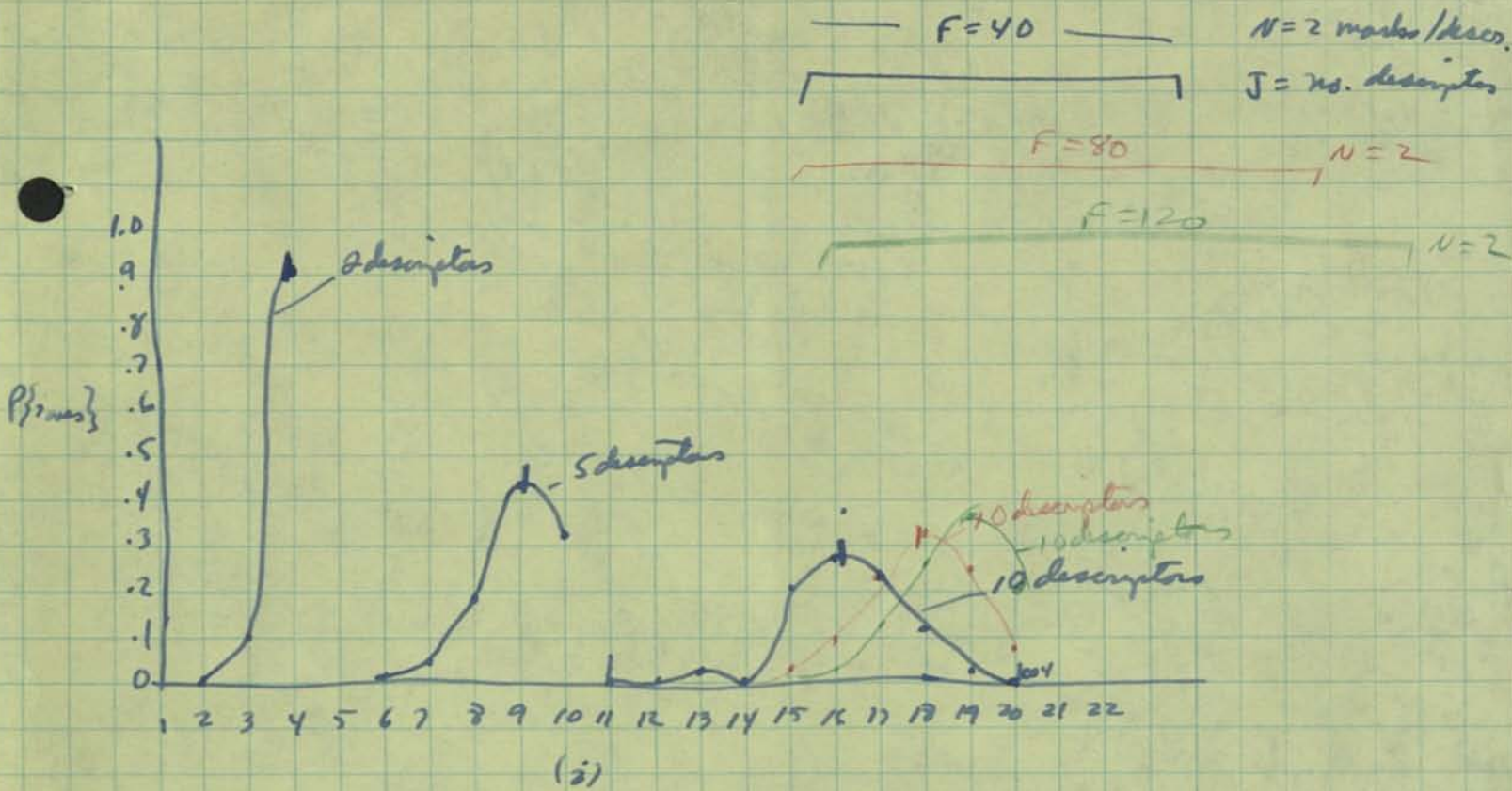
For this problem, the probability of  $m$  empty files will have an upper bound of  $(P_m)^8$ .

It would appear to be an equivalent problem if all the pawns (after distribution on the rows) were all moved directly down to the bottom row, so that all the pawns were distributed on the bottom row. We are then interested in the probability of an empty cell in  $m$  <sup>8-cell</sup> row which has 16 pawns placed on it. The above expression is pertinent.

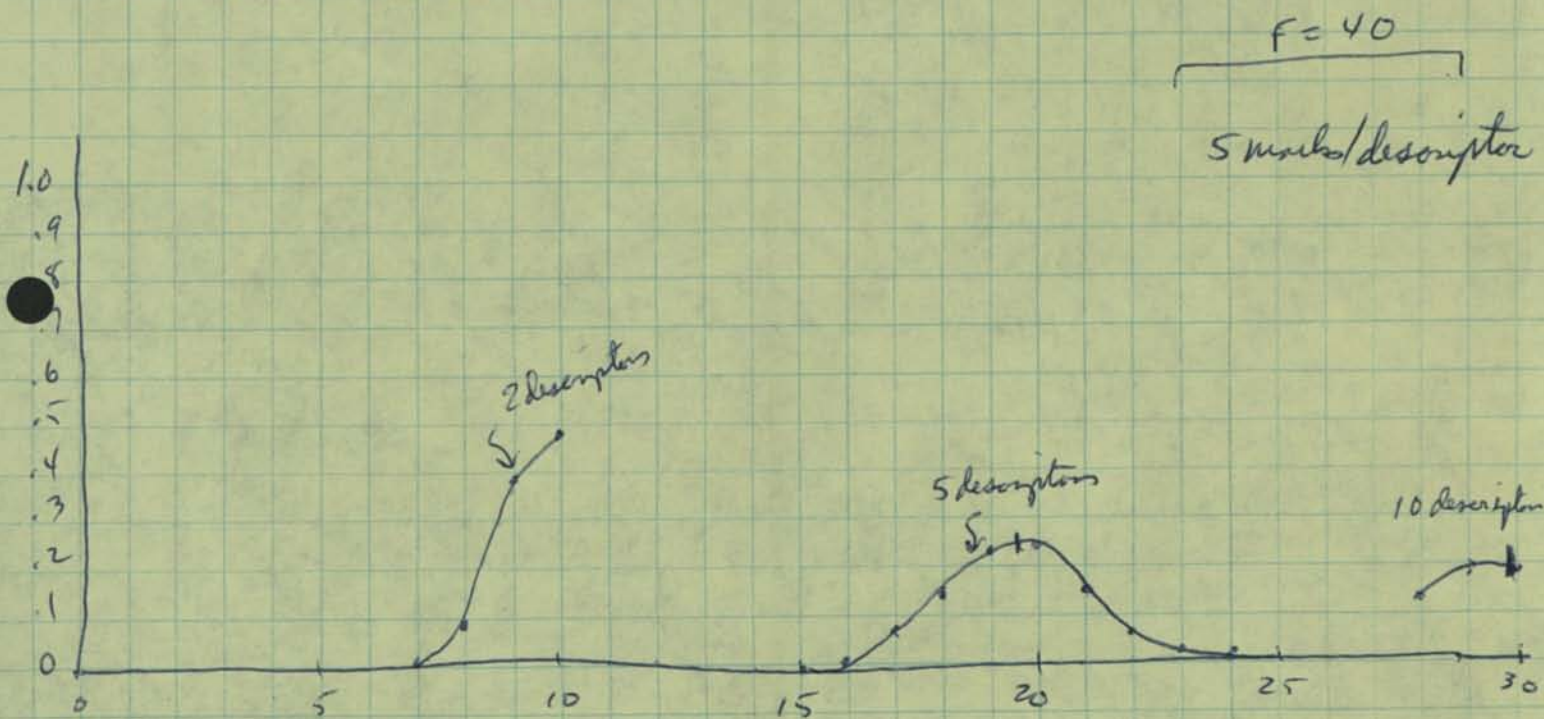
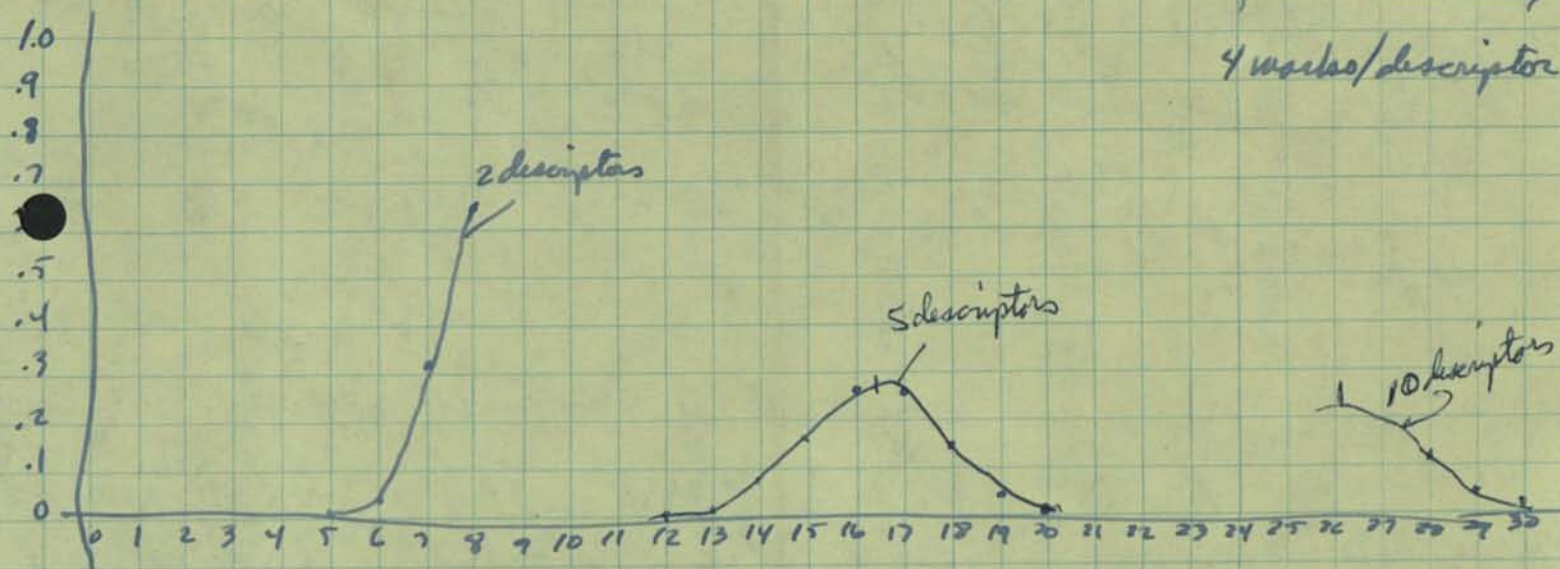
$$P_m = \binom{8}{m} \sum_{v=0}^{8-m} (-1)^v \binom{8-m}{v} \left(1 - \frac{m+v}{8}\right)^{16}$$

$n=8$   
 $R=16$





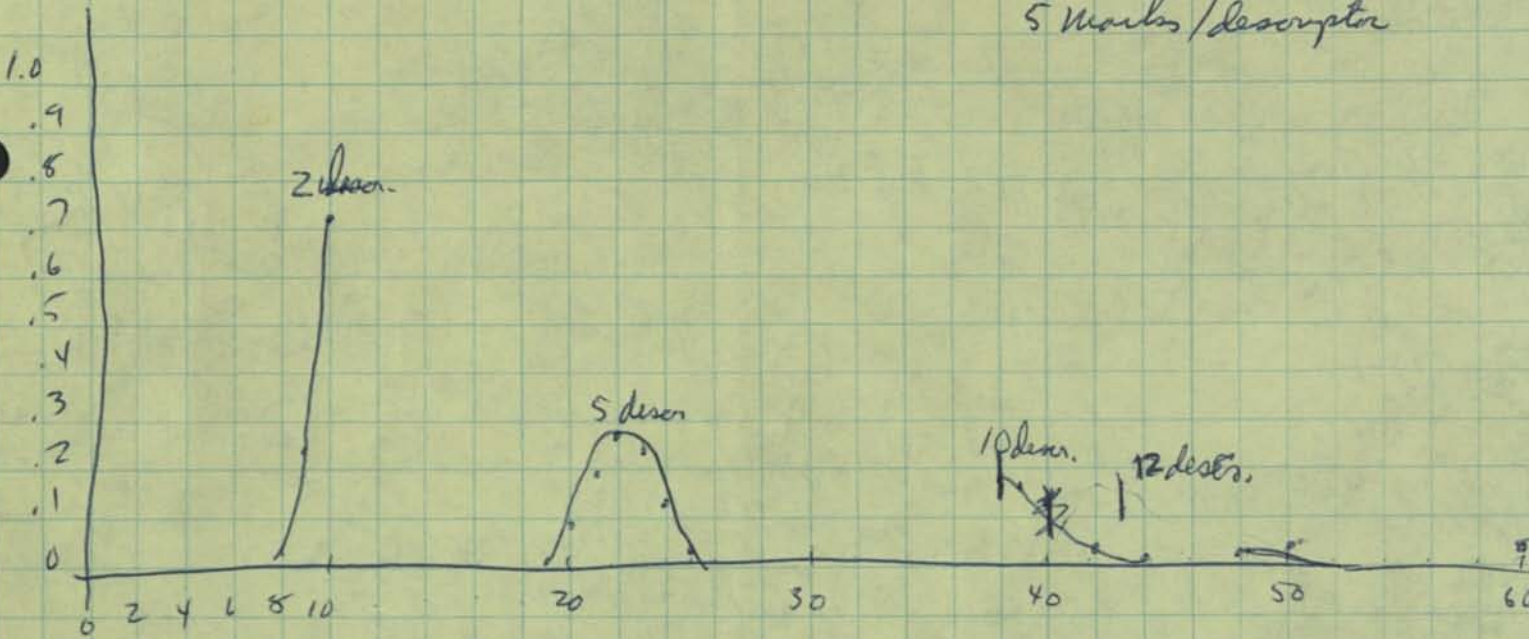






F=80

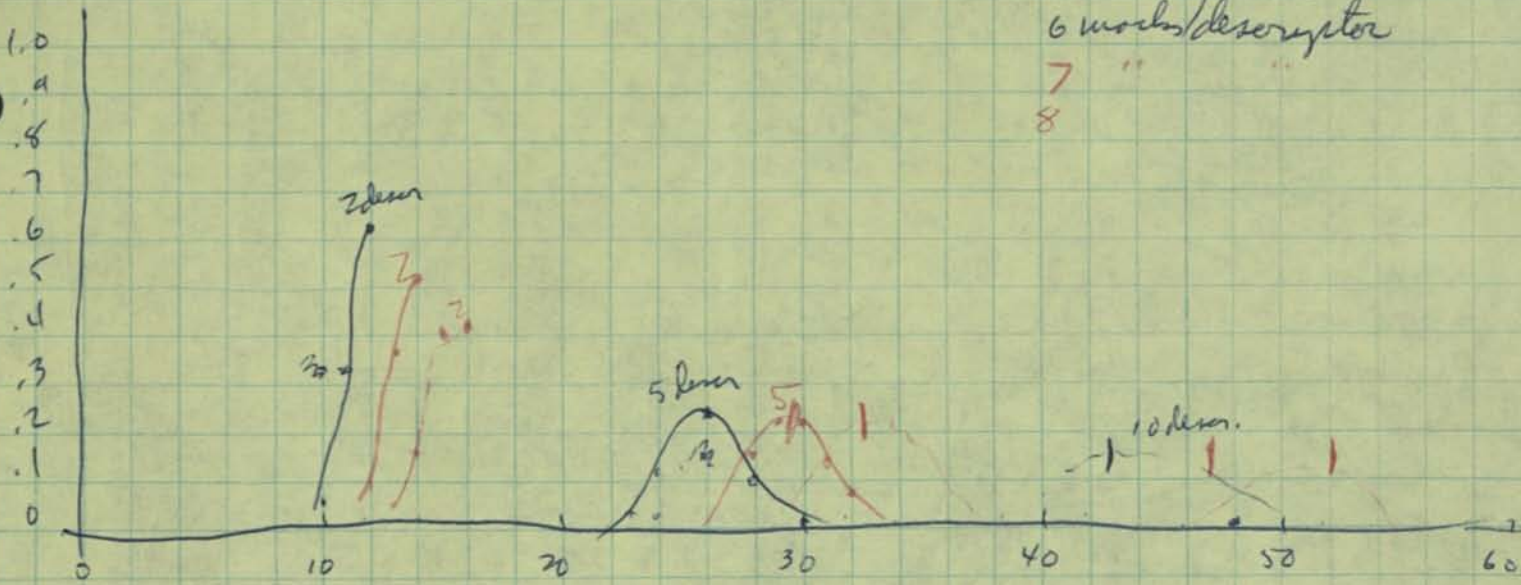
5 marks/descriptor



F=80

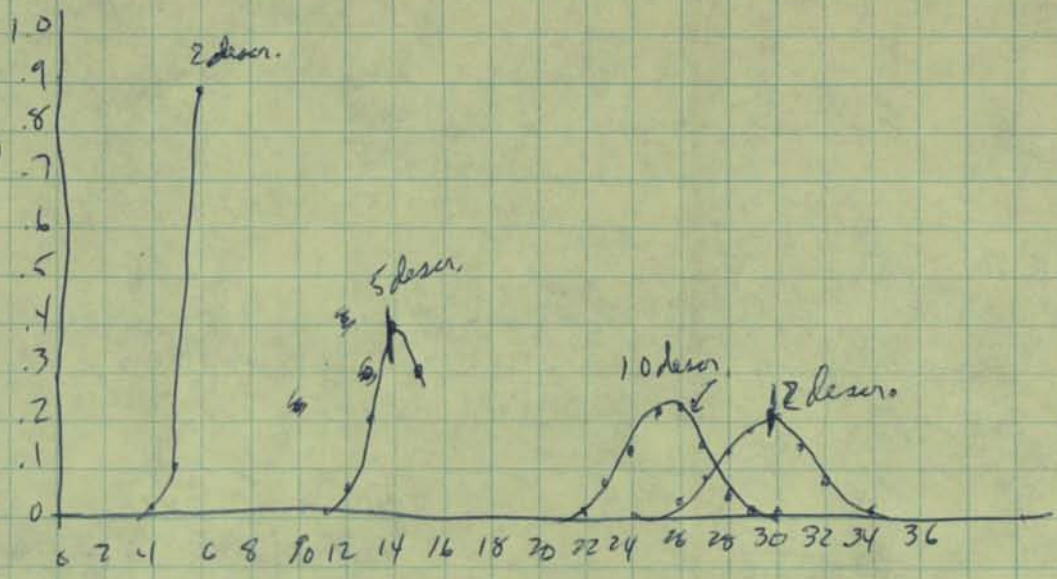
6 marks/descriptor

7  
8

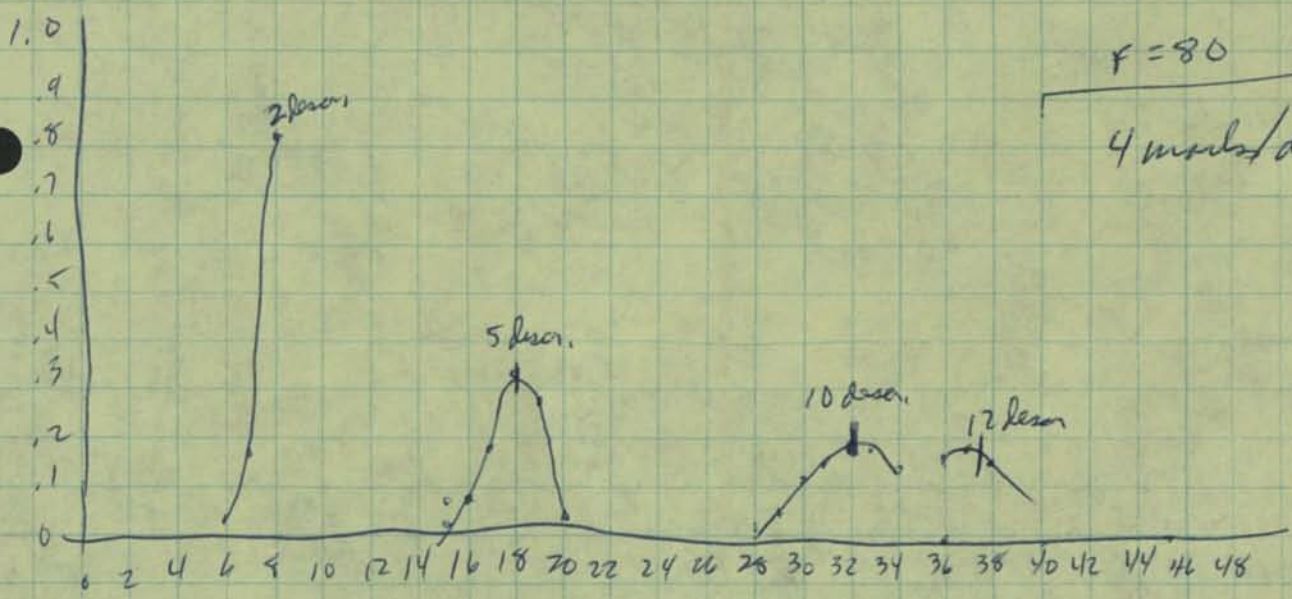




$f=80$   
3 marks/descriptor



$f=80$   
4 marks/descrip.





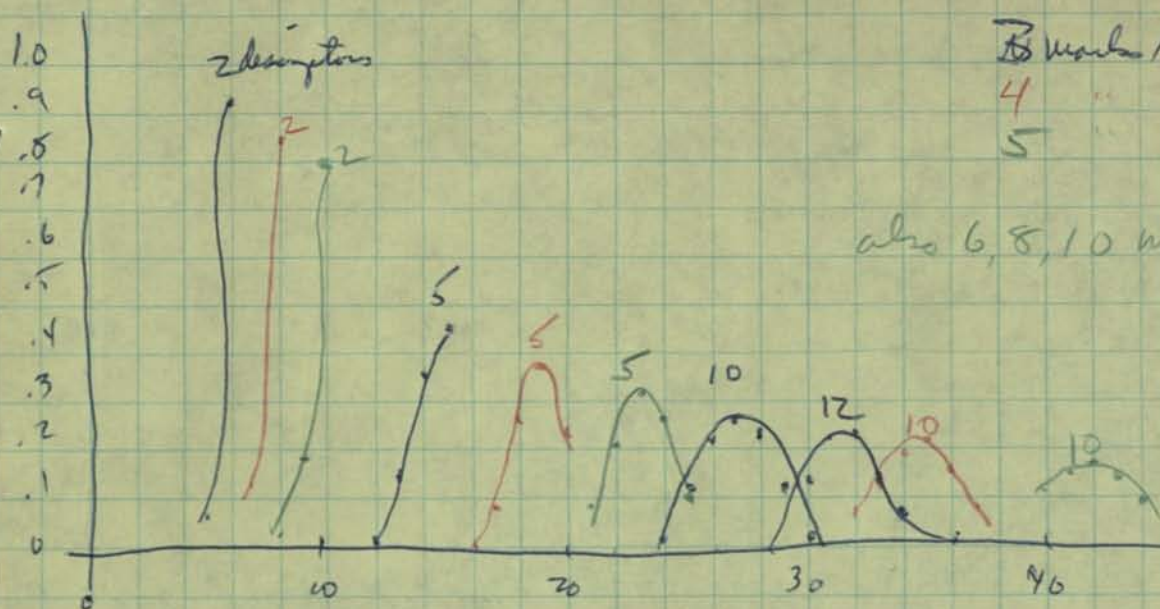
$$F=120$$

Words/descriptor

4

5

also 6, 8, 10 words/descriptor  
available



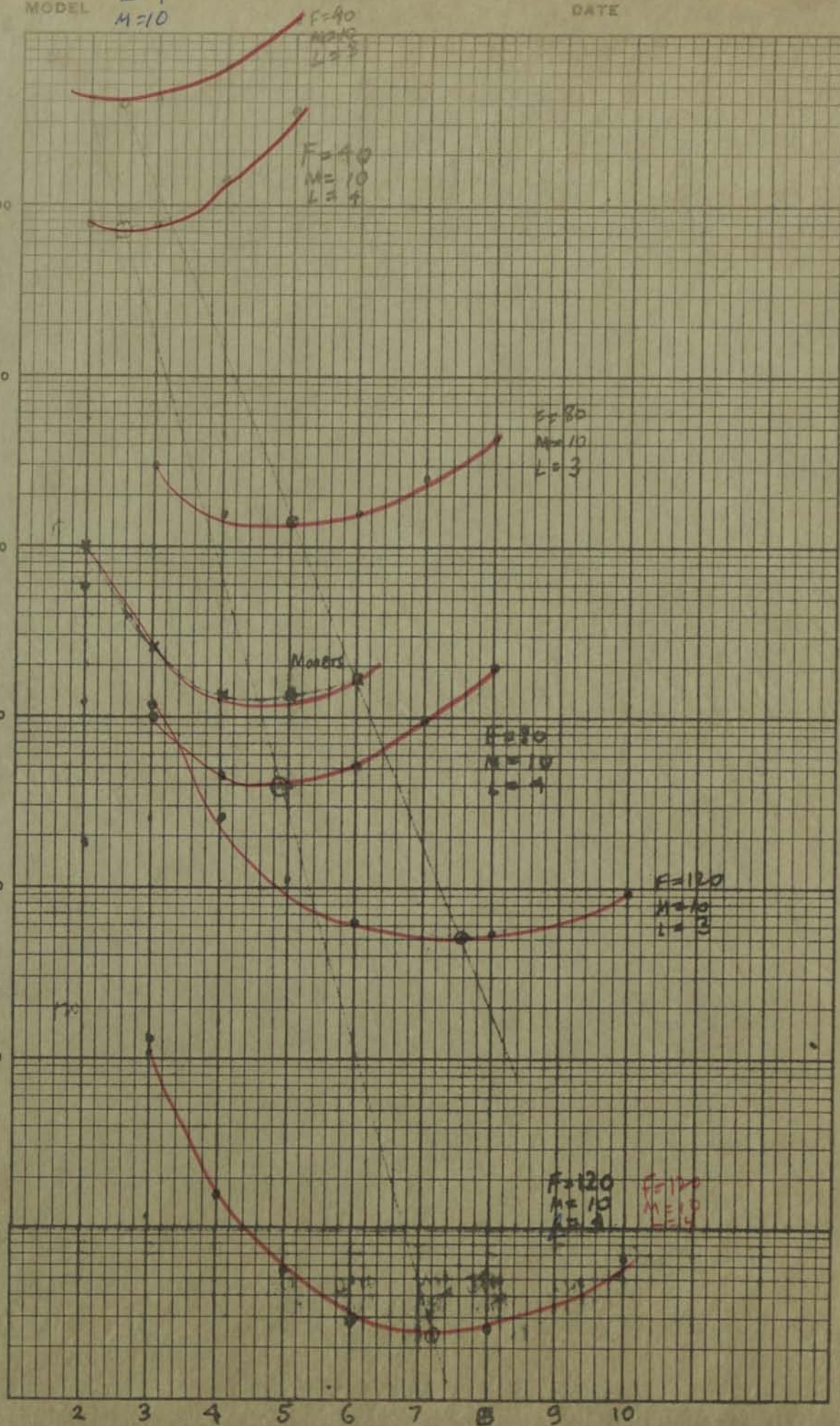


MODEL

M=10

DATE

KE  
SEMI-LOGARITHMIC  
Krupp & Esser Co.  
7 CYCLES X 80 DIVISIONS



Smiley  
C. 11



$f_d$

$N = 10$

$T$

$f_d$

F	N <sub>a</sub>	T	E <sub>20</sub>	W <sub>10</sub>	M <sub>10</sub>	T <sub>mean</sub>
40	2	5	350 -2	626 -2	104 -1	16
		10	194 -4	940 5	108 -3	
		15	807 -8	1448 -9	113 -5	
		20	285 -13	204 -16	117 -7	
80	5	10	357 -3	281 -3	591 -3	38
		20	316 7	980 8	350 6	
		30	341 12	558 14	207 9	
		40	945 19	295 27	122 12	
		50	761 30	351 35	723 16	
120	8	10	709 3	627 3	946 3	60
		20	242 6	132 6	894 6	
		30	331 10	610 11	895 9	
		40	139 14	297 16	799 12	
		50	116 19	252 23	756 15	
		60	919 26	404 35	715 18	
		70	123 33	175 47	626 21	
		80	359 46	—	639 24	

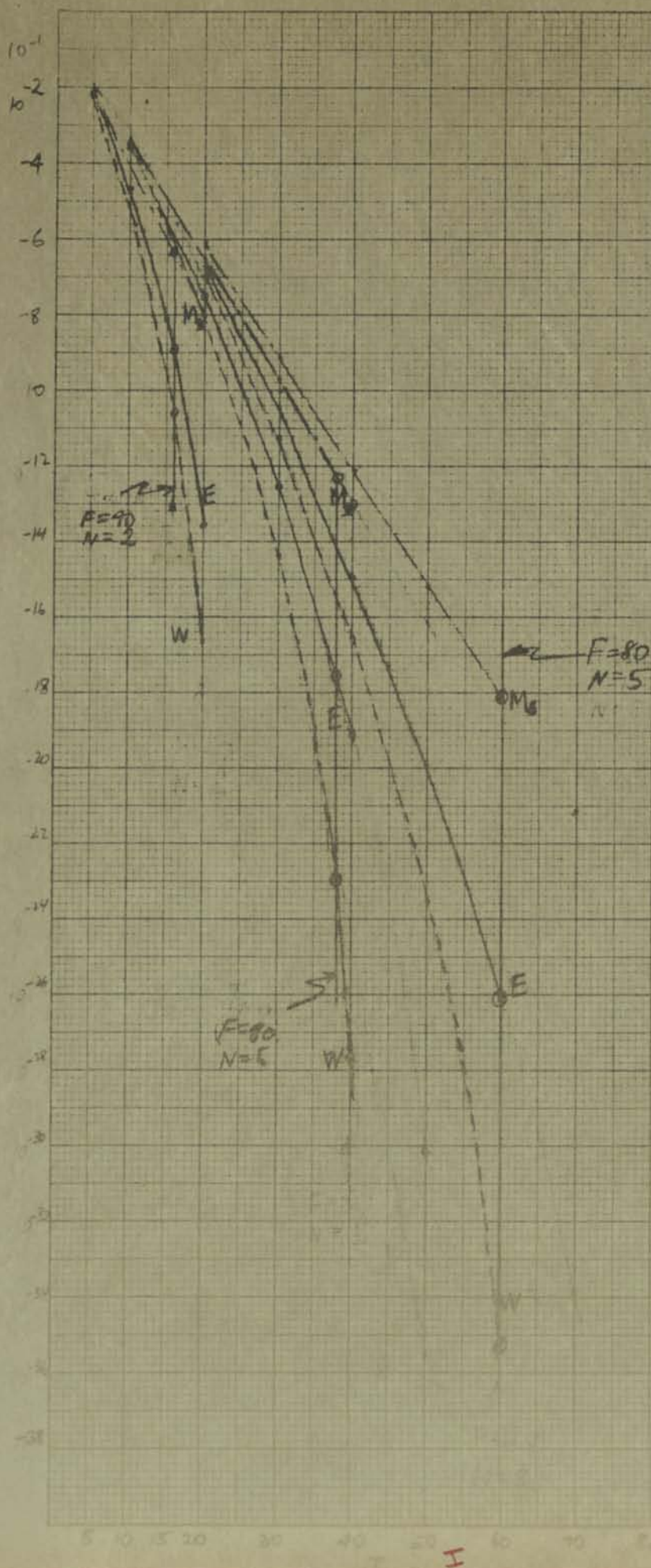


Comparison of Exact, Wise's  
and Moores' Calculations of  
Random Selection Rate  
for a file of descriptors, each  
composed of ten descriptors  
cases.

Field length	No. of 1's in deser
F	N
40	2
80	5
120	8

N chosen for optimum selection  
for each F.

M<sub>s</sub> = Moores' values  
E = Exact values  
W = Wise's values

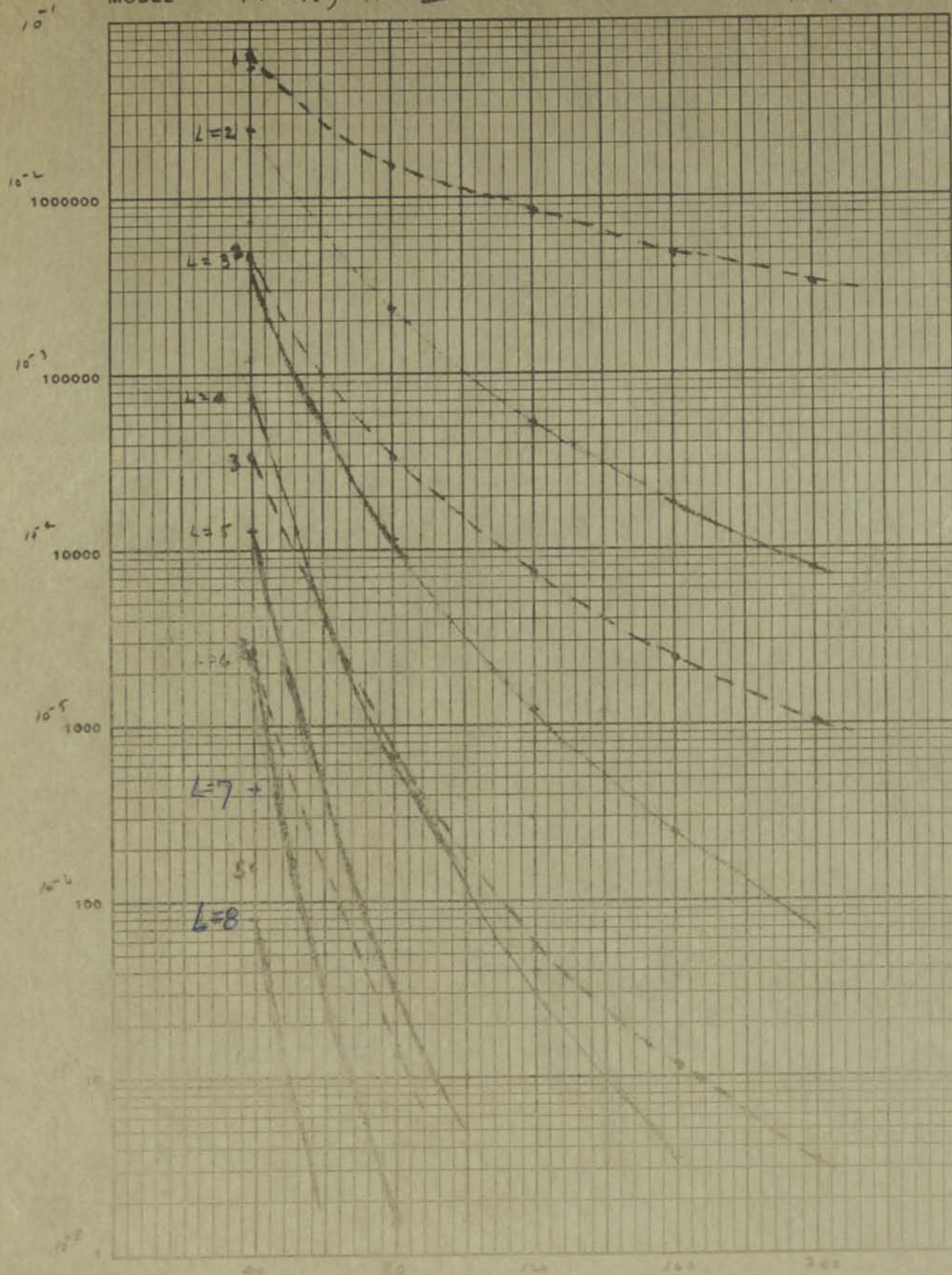




Decreasing  $M$  improves the  
false drop rate. This is  
worse case for present tables.

MODEL  $M=6, N=7$   
 $M=10, N=2$

DATE 10/28/60





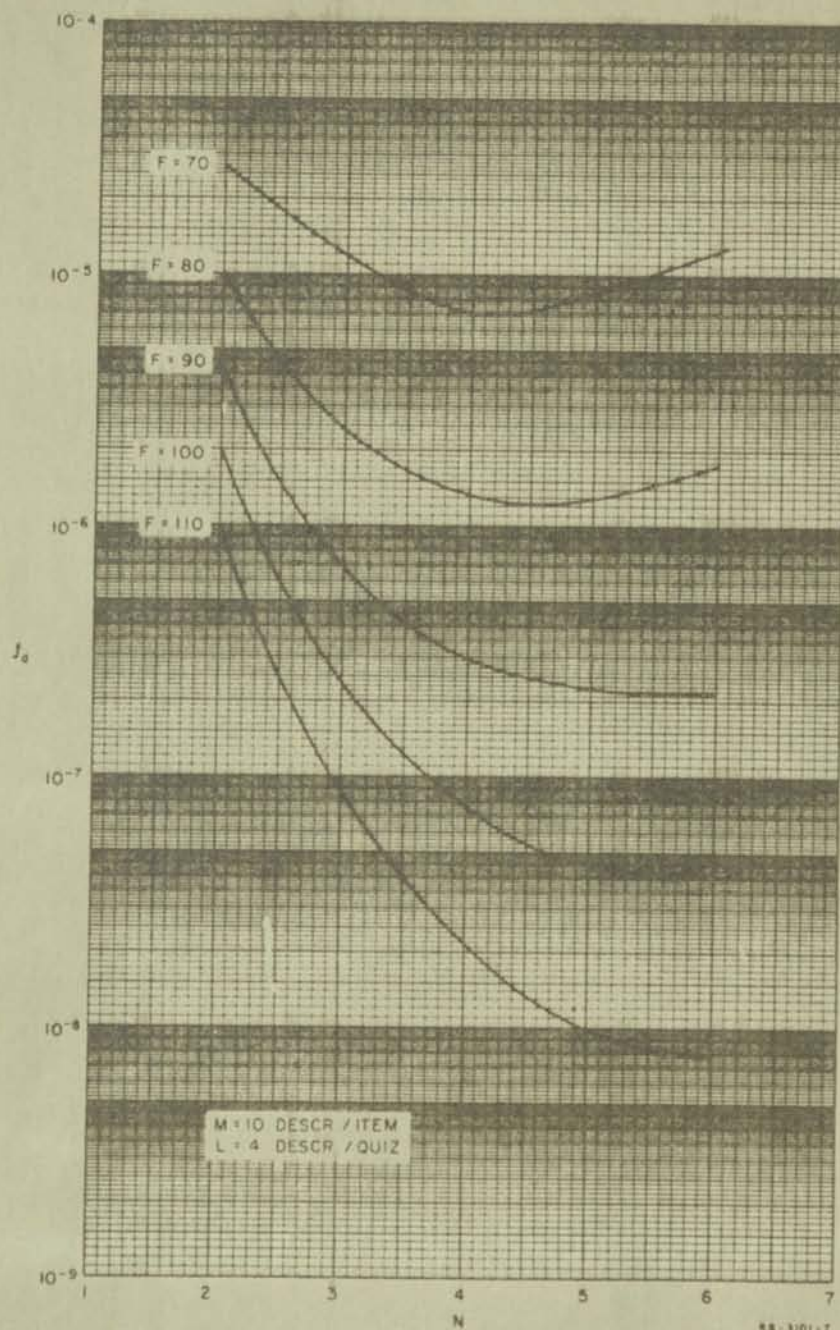


FIG. A-2  
DROPPING FRACTION ( $f_d$ ) VS THE NUMBER OF MARKS IN BASIC  
DESCRIPTOR (N) FOR VARIOUS FIELD LENGTHS

MIRF  
QPR #3



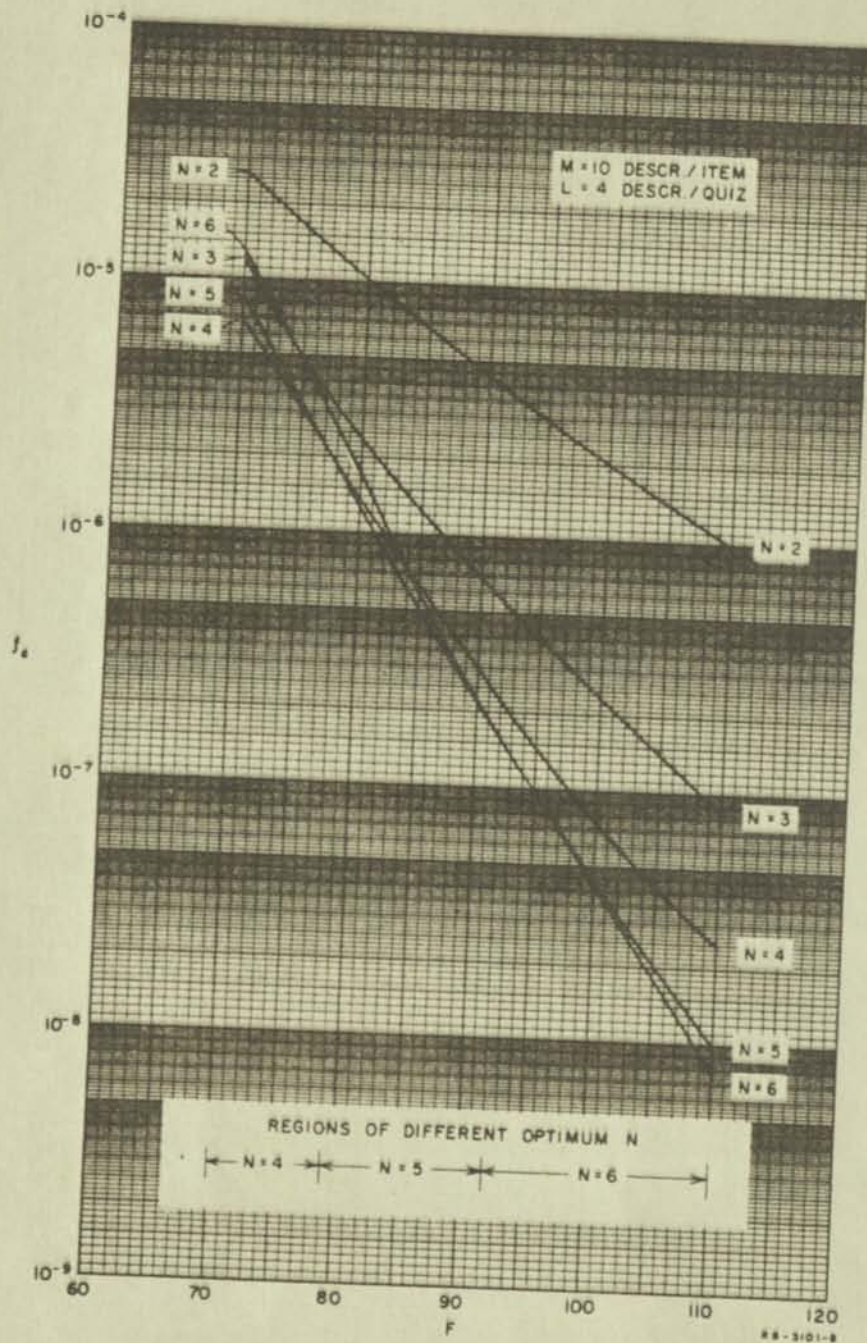
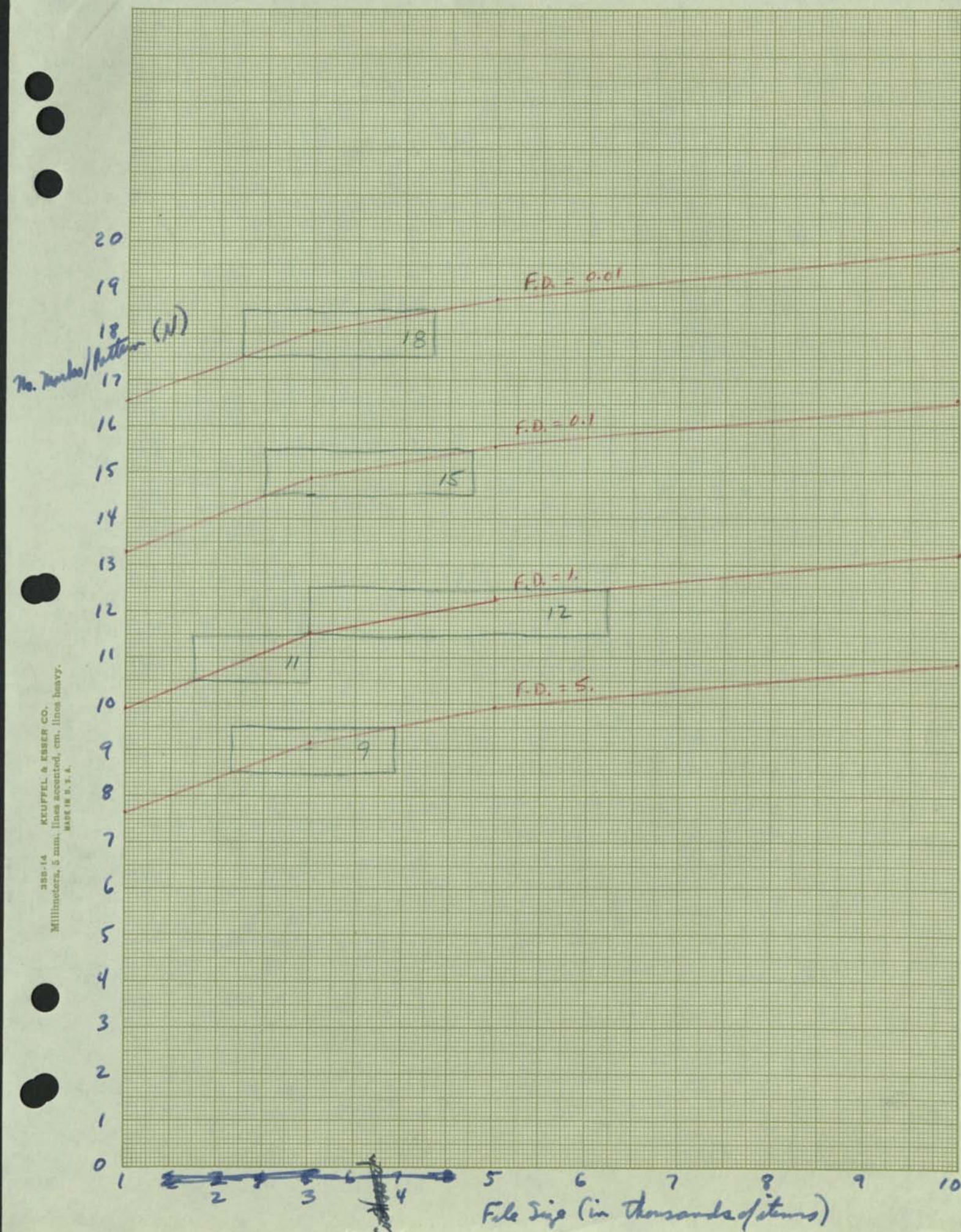


FIG. A-1  
 DROPPING FRACTION ( $f_d$ ) VS FIELD LENGTH (F) FOR VARIOUS NUMBER  
 OF MARKS IN BASIC DESCRIPTOR (N)



Word Marks/Pattern Required for Specifying Single Code Fields with Single Descriptors





# # Sigs. Code Design for Freq., P.W, PRF, Scan <sup>(only when taken alone)</sup>

need  $N$  modes/pattern, where

$$N = -\frac{3.31}{L} \log_{10} R$$

$$= -3.31 \log_{10} \frac{1}{3000} = 3.31 \log_{10} (3000)$$

$$= 3.31 (3.477)$$

$$= 11.5$$

$$= 12$$

$L$  = lower bound of no. descriptions used for searching = 1

$R$  = tolerable noise ratio

=  $\frac{\text{max no. false drops}}{\text{file size}}$

$$= \frac{1}{3000}$$

file size	$N$
1000	10
3000	15
10,000	13

$L=1$   
no. false drops = 1

File Size	$\log_{10} R$	$N$
1000	3.	9.93
3000	3.477	11.51
5000	3.699	12.3
10,000	4.	13.24

no. false drops = 5  
 $L=1$

File Size	$\log_{10} R$	$N$
1000	2.301	7.64
3000	2.778	9.2
5000	3.	9.95
10,000	3.301	10.9

$L=1$   
no. false drops = 0.1

File Size	$\log_{10} R$	$N$
1000	4.	13.3
3000	4.477	14.9
5000	4.699	15.6
10,000	5.	16.6

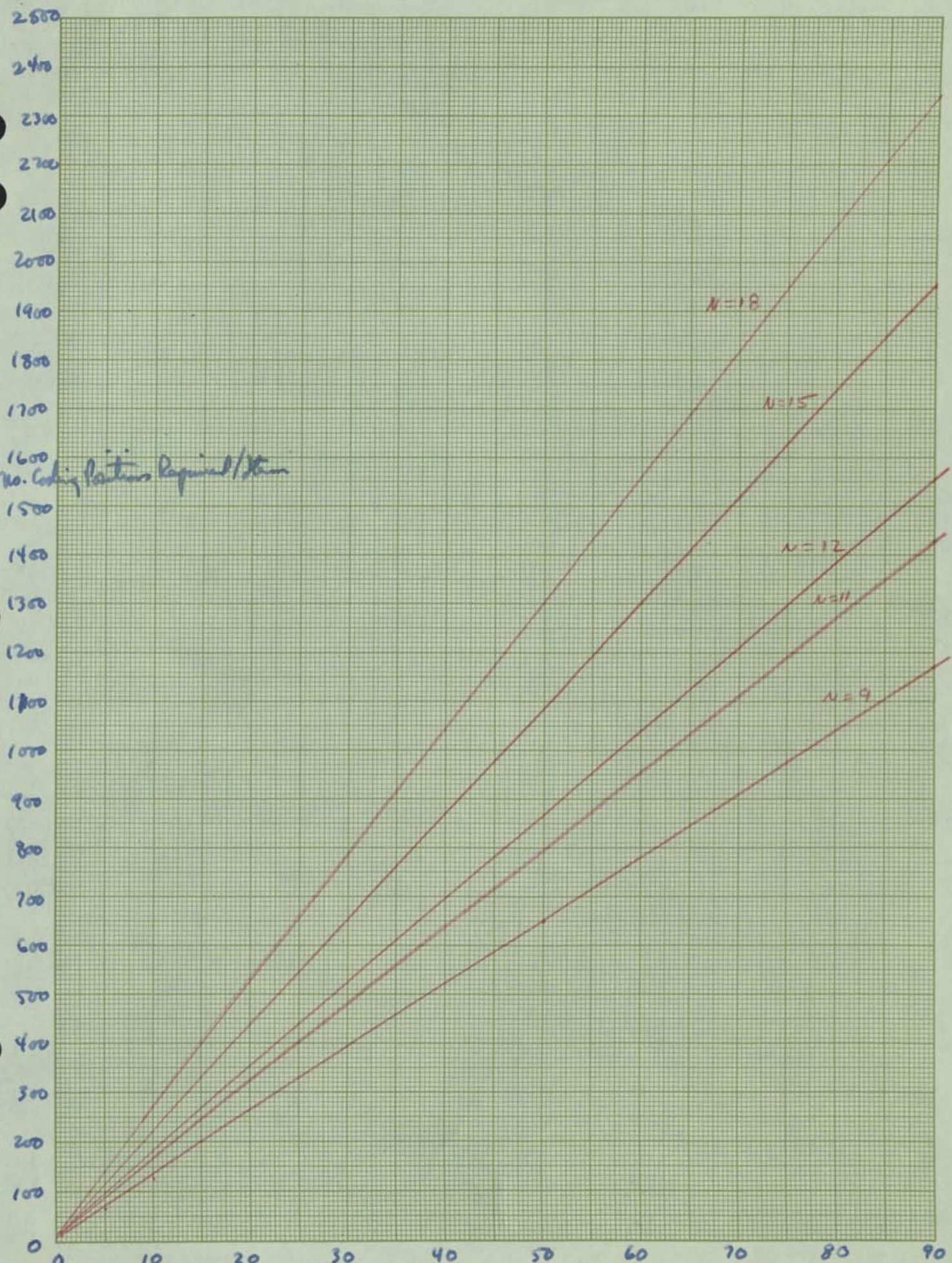
$L=1$   
no. false drops = 0.01

File Size	$\log_{10} R$	$N$
1000	5.	16.6
3000	5.477	18.1
5000	5.699	18.8
10,000	6.	19.9



Total no. Coding Positions Required/Item  
(S)

K&E 10 X 10 TO THE CM. 358-14  
KEUFFEL & ESSER CO. MADE IN U.S.A.

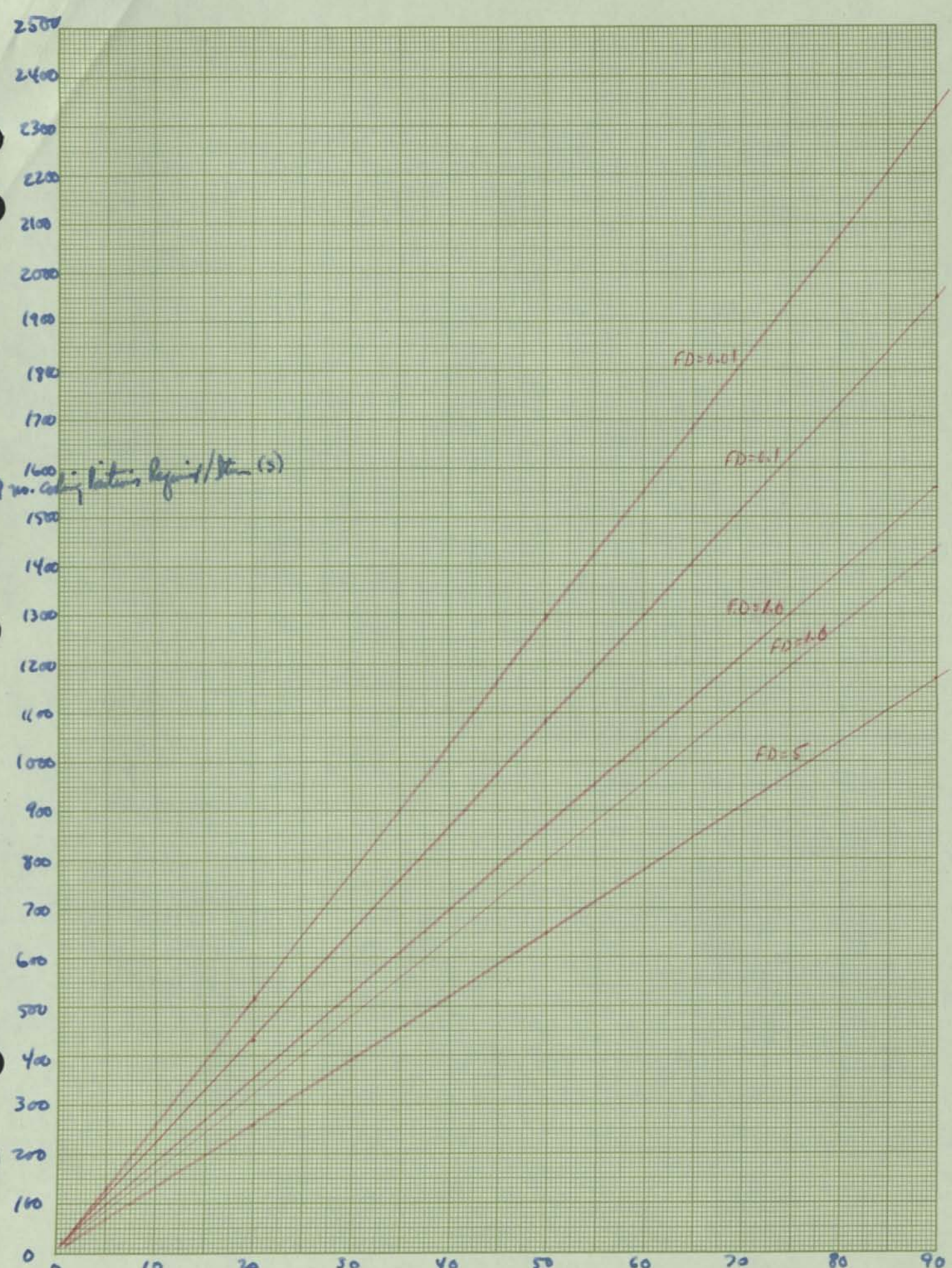


Max no. Descriptors used for Indexing (M)



Total no. coding letters required/letter (S)

K&E 10 X 10 TO THE CM. KEUFFEL & ESSER CO. MADE IN U.S.A. 358-14



No. Descriptors Used to Describe Each Parameter



Max. of 20 descriptors / freq.

False Drop	N	Sf
0.01	18	520
.1	15	432
1	11-12	318-347
5	9	260

Max. of 5 descriptors / freq

False drop	N	Sf
.01	18	130
.1	15	108
1	11-12	80-87
5	9	65

Max. of 50 descr. / doc

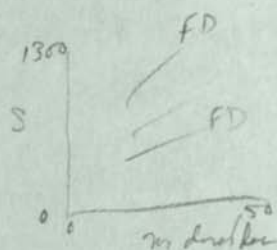
F.D.	N	Sf
.01	18	1300
.1	15	1085
1	11-12	795-870
5	9	650

Max. of 1 descr. / doc

FD	N	S
.01	18	26
.1	15	17-22
1	11-12	16
5	9	13

Max. 90 descr. / doc

.01	2340
.1	1950
1	1430-1560
5	1170





# Single Sp. Gate Field

$N = \text{no. marks/pattern} =$

<u>False Pos.</u>	<u>N</u>
5	9
1	12
0.1	15
0.01	18

$L = \text{lower bound of disc. used for resolving} = 1$

$$S = 1.445 \text{ NM}$$

$$M = \text{max. no. disc. used for indexing} = 20$$

$$= 1.445(20)9 = 261 \text{ for F.D.} = 5$$

$$()12 = 347 \quad 1$$

$$()15 = 434 \quad 0.1$$

$$()18 = 520 \quad 0.01$$



# Superimposed Code Design

Need  $N$  words/pattern, where

$$N = \left\lceil \left(\frac{1}{L}\right)(3.31)(-\log_{10} R) \right\rceil ; R = \frac{E_{\text{max}}}{\text{file size}} = \frac{\text{max. depth of L desc.}}{\text{file size}}$$

$$= \left\lceil 3.31(-\log_{10} \frac{E_{\text{max}}}{3500}) \right\rceil$$

$L = 1$  in all cases  
file size = 3500

~~For~~  $E_{\text{max}} = 5$

$$N = \left\lceil 3.31(-\log_{10} \frac{5}{3500}) \right\rceil$$

$E_{\text{max}} \text{ (F.D.)}$	$N$
5	9
1	12
0.1	15
0.01	18

(see Fig.)

$$S = \left\lceil 1.445 NM \right\rceil$$

;  $M$  = upper bound of no. of descriptors used for indexing.

$$= \begin{cases} 20 & \text{freq} \\ 4 & \text{IRF} \\ 3 & \text{PW} \\ 3 & \text{Sec/Ref.} \end{cases}$$



For  $E_{max} = 5$ ,  $N = 9$

$$S = 1.445(9)20 = 261$$

$$1.445(9)4 = 52.0$$

$$1.445(9)3 = 39.0$$

freq.

PRF

PW, Scan/Ref.

For  $E_{max} = 1$ ,  $N = 12$

$$S = 1.445(12)20 = 346.$$

$$" \quad 4 = 69$$

$$" \quad 3 = 52$$

freq

PRF

PW, Scan

For  $E_{max} = 0.1$ ,  $N = 15$

$$S = 1.445(15)20 = 434.$$

$$4 = 87.$$

$$3 = 65.$$

For  $E_{max} = 0.01$ ,  $N = 18$

$$S = 1.445(18)20 = 520.$$

$$4 = 104$$

$$3 = 78$$



need  $N$  marks / pattern, where

$$N = \left\langle \left( \frac{1}{L} \right) (3.31) (-\log_{10} R) \right\rangle.$$

$F_{\text{file}} = 1000$  items

$R = \text{tolerable noise ratio} = \frac{E_{\text{noise}}}{C} = \frac{\text{max no. of spec. drops w/ L desc.}}{\text{file size}}$

$$\frac{1 \text{ file drop}}{1000 \text{ items in file}} = 10^{-3}$$

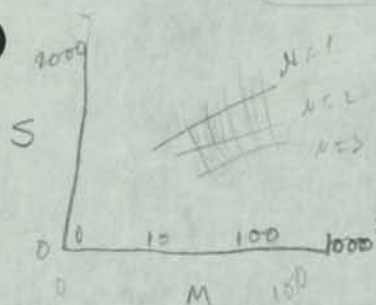
$L = \text{lower bound of no. of desc. used for marking}$   
 $= 1$

9.93

$$N = \left\langle 3.31 (-\log_{10} 10^{-3}) \right\rangle = \left\langle 3.31 (3) \right\rangle$$

$$= 10$$

$$S = \left\langle 1.445 N M \right\rangle$$



$M = \text{upper bound of no. desc. used for indexing}$

$$= \boxed{2500}$$

need to reduce this  
 suggest 100 file items w/ 25 desc.

$$= \left\langle 1.445 (10) (2500) \right\rangle = \left\langle \right.$$

$$\approx 35,000 !$$

$$= \left\langle 1.445 (10) 25 \right\rangle = 360.$$

$$N=13$$

$$M=1000$$

$$S=18,860$$

$$M=7000$$

$$S=5860$$

$$M=1000$$

$$S=1880$$

$$M=150$$

$$S=2800$$

$$M=10$$

$$S=188$$

$$M=5$$

$$S=18.8$$

$$M=1$$

$$S=1.88$$

$$M=10$$

$$S=752$$

$$N=4$$

$$M=1000$$

$$S=5860$$

$$M=500$$

$$S=2900$$

$$M=100$$

$$S=580$$

$$M=10$$

$$S=58$$

$$M=40$$

$$S=232$$

$$M=1$$

$$S=5.8$$



# Sup. Code Design for Frey, PW, PRF, Scan (when taken alone)

5 positions in the code field. where

$$S = 1.445 NM$$

$M$  = upper bound of no. descriptions  
used for indexing

$$N=9$$

$M$	$S = 9(1.445)M$
1	13
5	65
10	130
15	195
20	260
30	390
40	520
50	650
90	1170
100	1300

$$N=18$$

$M$	$S = 18(1.445)M$
1	26
5	130
10	260
15	390
20	520
50	1300
90	2600
100	2600

$$N=11$$

$M$	$S = 11(1.445)M$
1	16
5	79.5
10	159
15	238
20	318
50	795
90	1300
100	1590

$$N=12$$

$M$	$S = 12(1.445)M$
1	17
5	87
10	173
15	260
20	347
50	870
90	1560
100	1730

$$N=15$$

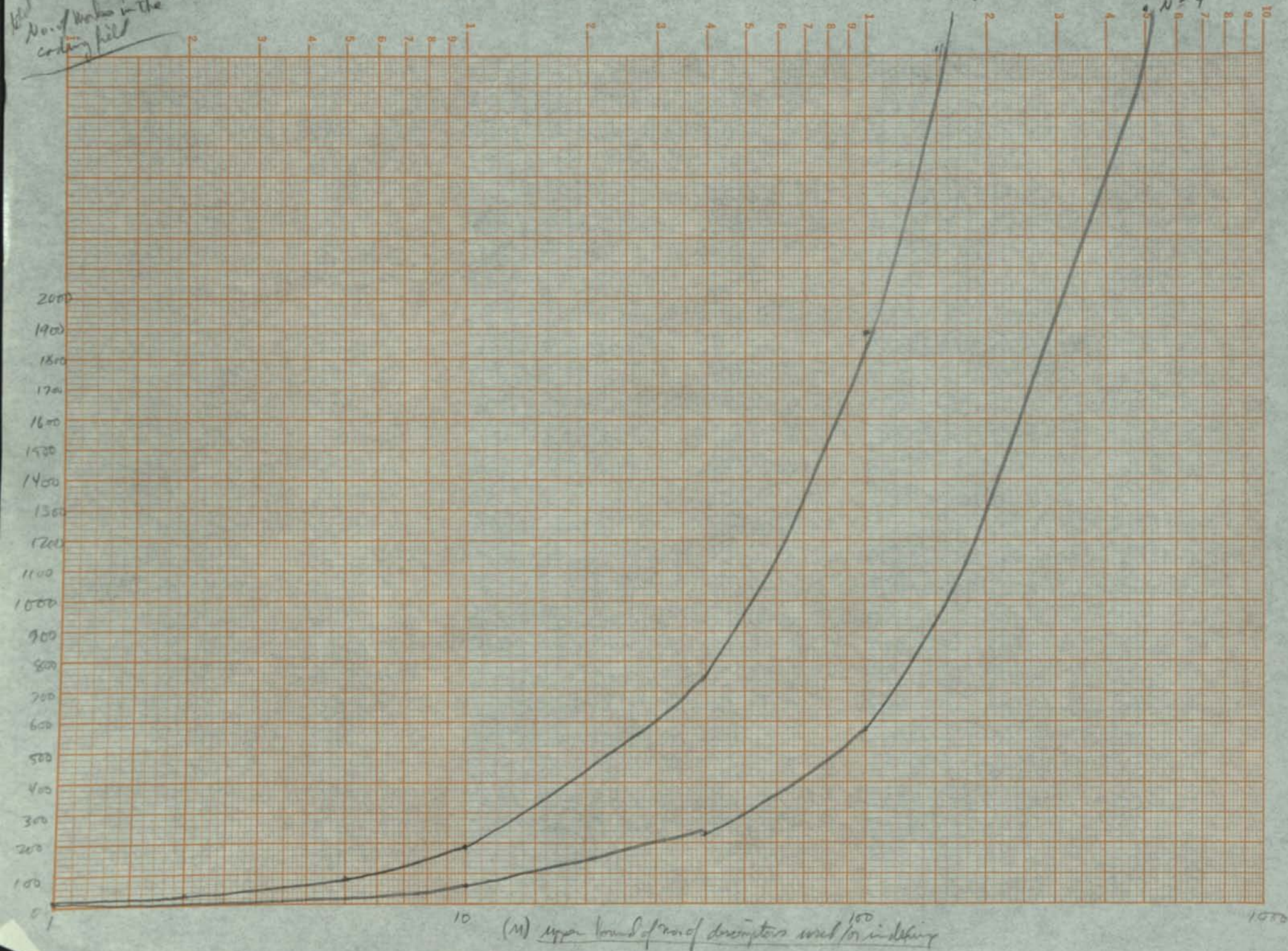
$M$	$S = 15(1.445)M$
1	21.6
5	108
10	216
15	325
20	432
50	1080
90	1970
100	2160



*total  
No. of holes in the  
coding field*

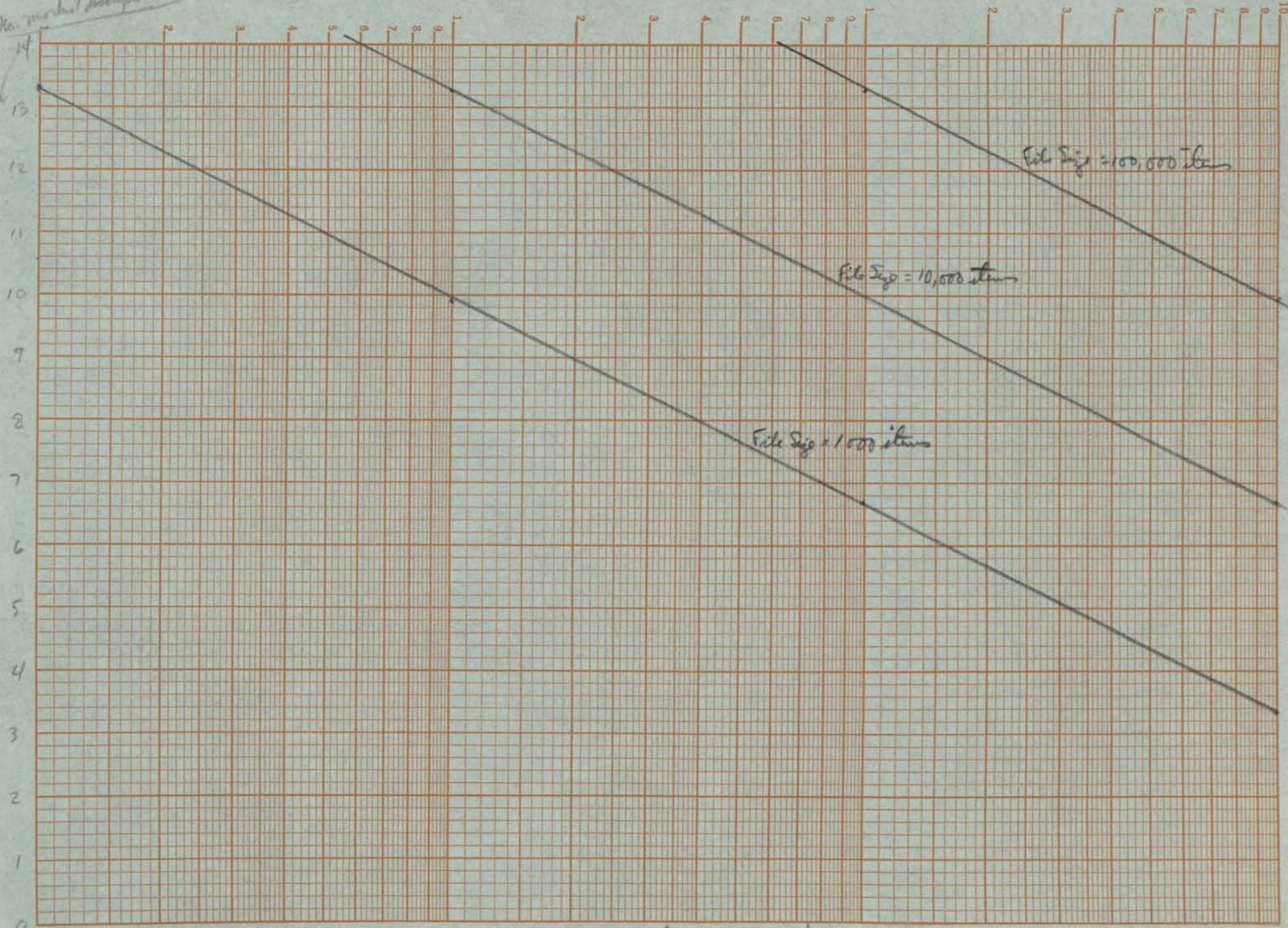
$N=13$

$N=4$





No. words/descriptor (N)



max. no. files descriptors with one 10 word descriptor (Kmax)



$$R = \frac{E_{\text{tot}}}{C}$$

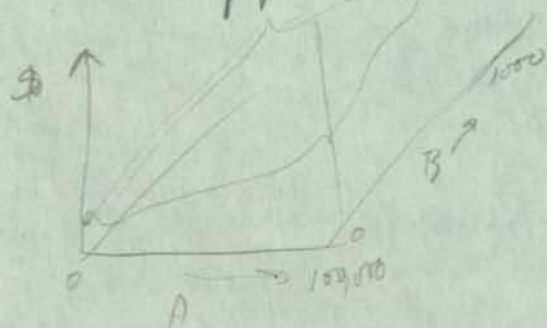
$E_{\text{tot}}$	$C$	$R$	$-\log_{10} R$	$N = 3.31 R$
1	1000	$10^{-3}$	3	9.93
10	"	$10^{-2}$	2	6.62
100	"	$10^{-1}$	1	3.31
.1	"	$10^{-4}$	4	13.24

$$N = \left\lceil \frac{1}{2} (3.31) (-\log_{10} R) \right\rceil$$

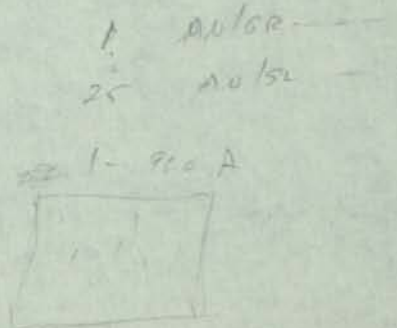
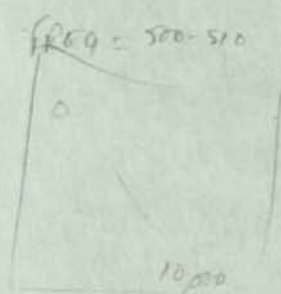
$E_{\text{tot}}$	$C$	$R$	$-\log_{10} R$	$N = 3.31 R$
.1	10,000	$10^{-5}$	5	16.55
.1	"	$10^{-4}$	4	13.24
10	"	$10^{-3}$	3	9.93
100	"	$10^{-2}$	2	6.62

$E_{\text{tot}}$	$C$	$R$	$-\log_{10} R$	$N = 3.31 R$
10	100,000	$10^{-4}$	4	13.24
100	"	$10^{-3}$	3	9.93

$$\text{Cost} = A(\text{no. of file items}) + B(\text{size of file items})$$



if item size is restricted to — units,  
how many file items will be required?





$$\text{total memory per line item} = (\text{no. } \overset{\text{parts/line item}}{\text{file items}}) (\text{no. bits/} \overset{\text{part}}{\text{item}})$$

$$\text{for each file item, the no. of bits} = S \rightarrow 1.445 \text{ (max no. descr. used for)}$$

$$S = 1.445 \frac{(\text{max. no. descriptors used for indexing}) (3.31)}{(\text{least no. of descriptors used for indexing})} \left( -\log_{10} \frac{\text{max no. file items}}{\text{file size}} \right)$$

$$S = 4.78 (\text{max no. descr. used for indexing}) \left( -\log_{10} \frac{\text{max no. file items}}{\text{file size}} \right)$$

$$S = 4.78 (\text{max no. descr. used for indexing}) (\log \text{ total file size})$$

let  $D$  = total no. descriptors to be used for each line item

let the line item be divided into  $P$  parts or sub elements

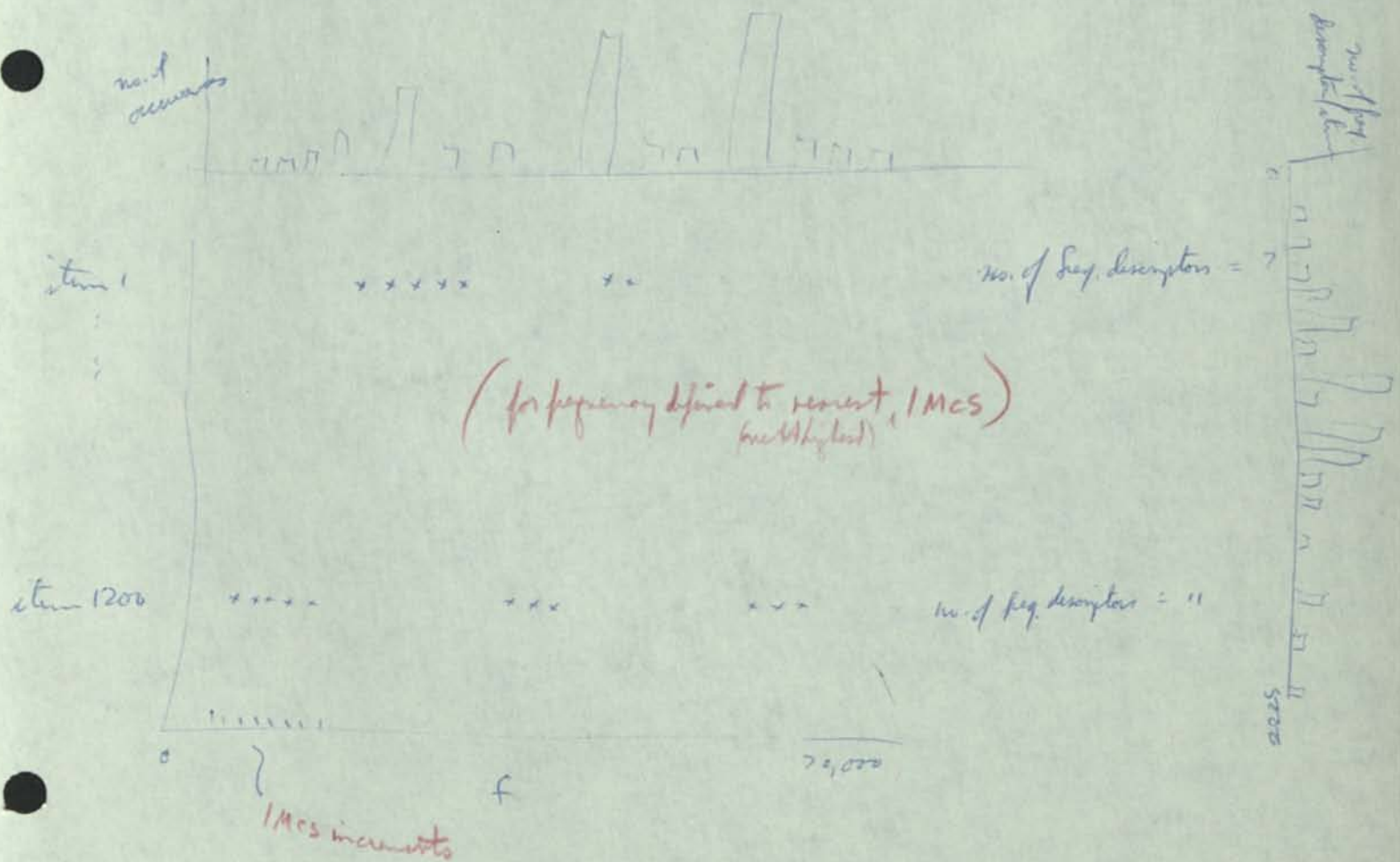
then  $\frac{D}{P}$  = max no. descr. used for indexing each part

$$\begin{aligned} \text{let total file size} &= C_{\text{items}}^{\text{line}} = 4000? \\ &= PC \text{ parts} \end{aligned}$$

$$S = 4.78 \left( \frac{D}{P} \right) \log PC$$



item	freq. (Mcs)	PRF	P.W.	Sol rotation
1-926	9000-9160	1350-1850	0.5	5.45
199	2400	2000	12.8	7.5-10
	55	200,000 ← highest	6-6.6	<del>3.8-60</del> ← highest
		247,934 ←	0.75	WD
Day 1-1200	23750-24750	200 ← lowest	100.0 ← highest	Man
	22-28 ← lowest	12.5	.065	0.33
	1-24 ← lowest	FM	.05 ← lowest	0.17 ← lowest
	1.7-2.0	CW	65-85	10-30
	34360-35160 ← highest	0-3500 ← lowest	1.02 ← lowest	0-10 ← lowest
	70,000 ← highest	FM/AM	6000. ← highest	240 ← highest
	0.18 ← lowest (LORAN)	12	CW	
	0.09 ← CYTAC	0.	Pφ	
			12-28	





8

		λ		λ	λ		x
x			x				
		λ		x			
					λ	x	
x			x		λ		x
x	x				x		

8

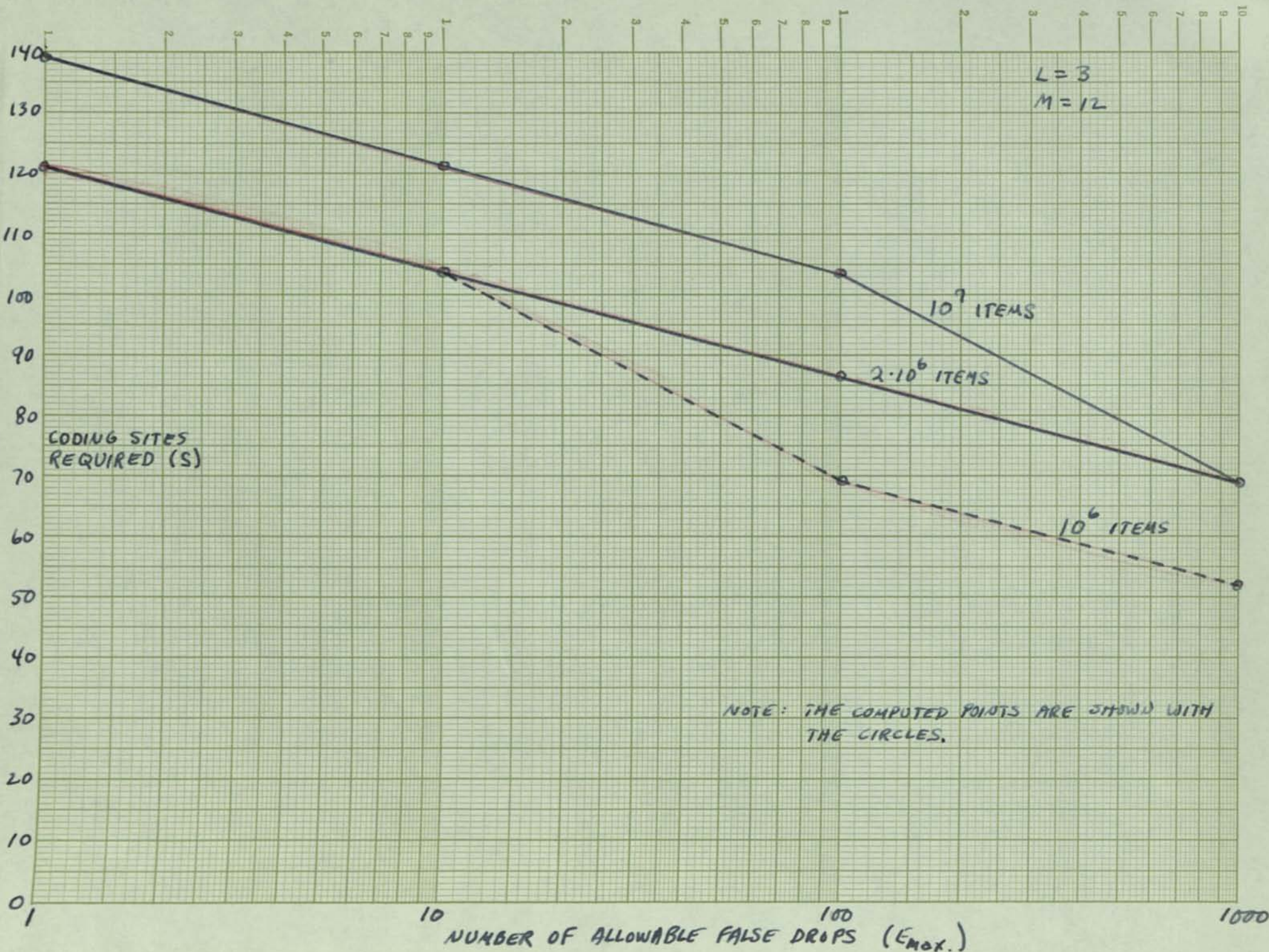
2 marks uniformly distributed in each row.

$P\{\text{exactly } k \text{ columns empty}\} = ?$

$$C_2^8 \left( \frac{C_2^6}{C_2^8} \right) \text{ mentioned as a factor by Madhav.}$$



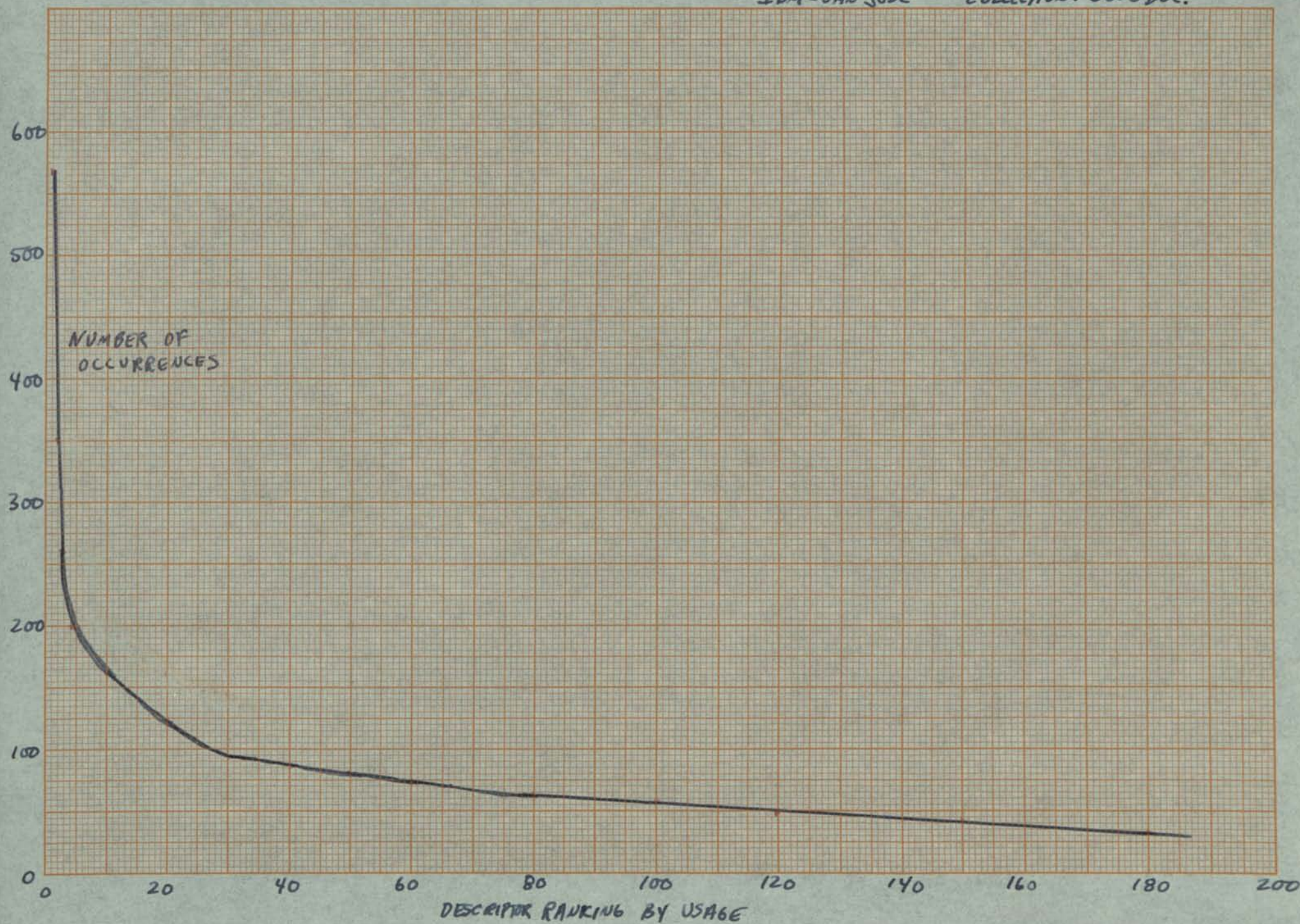
$L = 3$   
 $M = 12$





IBM - SAN JOSE

DICTIONARY: 6300 TERMS  
COLLECTION: 5000 DOC.





want to estimate

$$\frac{\text{Total no. false drops}}{\text{Total no. of drops}}$$

if originally had  $10^6$  items &  $10^3$  questions, & we chose a sample of  
 $10^4$  " &  $10^2$  "

Technique I  
will get

$$\frac{X_1 + X_2 + \dots + X_{100}}{T_1 + T_2 + \dots + T_{100}}$$

Technique II

will also be concerned with the distribution of how often you will get  $\frac{x}{T} = \text{constant}$ .

3) we want  $\frac{x}{T} = .2$ , now this is a Bernoulli problem, & we can choose the sample size.  $(\frac{x}{T} = .1 \text{ or } .9)$

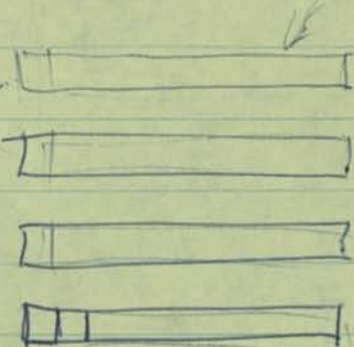
$$\frac{q}{np}$$



What happens if a particular <sup>word</sup> pattern appears more often than it would randomly? (e.g. popular descriptors)

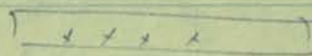
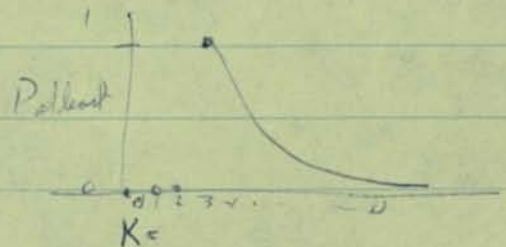
What happens if these aren't independent of each other?

e.g. descriptor A is often used with descriptor B



uniform distr.

What happens if there are a variable number of descriptors/descriptors?



$P\{K \text{ marks in the field}\} = ?$

possibility of enough unique combinations so that the uniqueness is improved?



(8) diff. ways to P

$$\binom{50}{2} = \frac{50!}{2!(48!)} = \frac{49 \cdot 50}{2} = 25 \cdot 49$$

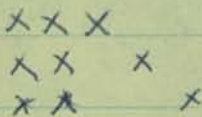


for a fixed no. of descriptors



old way

no. possible descriptors this could represent = ?



new way - no overlaps.

no. words = multiple of marks/descriptors.

no. possible descriptors this could represent = ?



Fixed word size + no. of marks/word

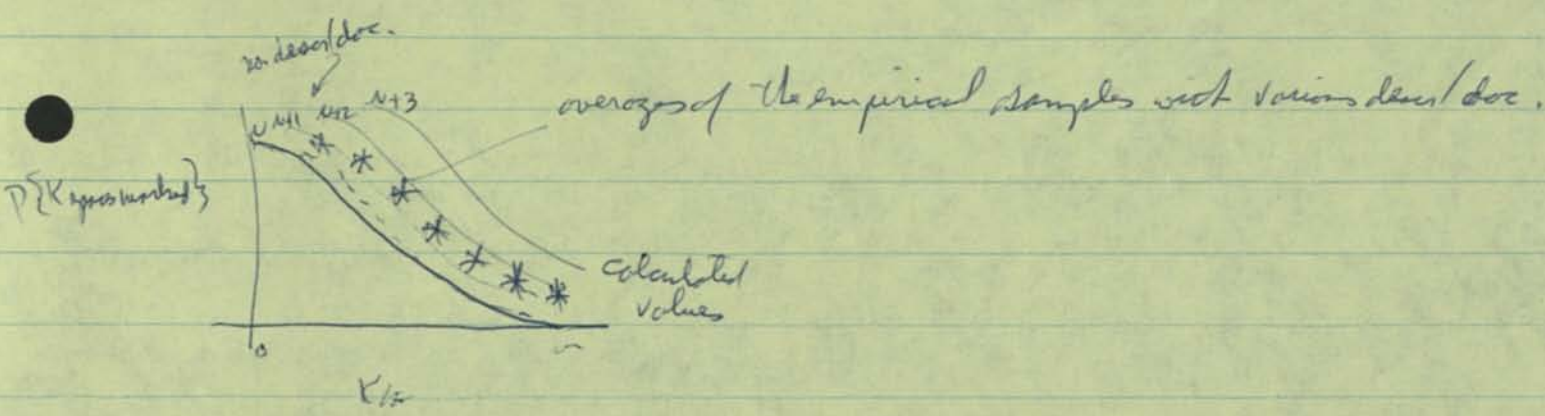
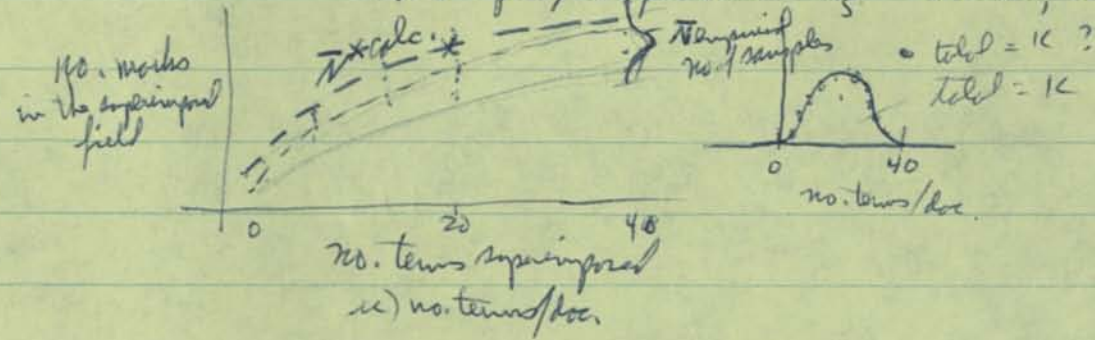


dictionary

# Experiment

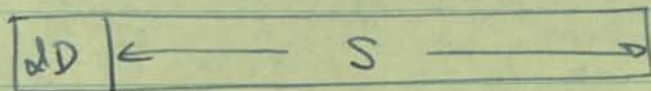
from a dictionary of descriptors, pull a random number of terms (0 - 40) & superimpose. Then count the no. of marks.

Repeat this test many times. The random no. of terms can be a specified function (e.g. normal, skewed or truncated normal, or uniform distribution)





## Superimposed Coding with Unique Binary Numbers for each Descriptor

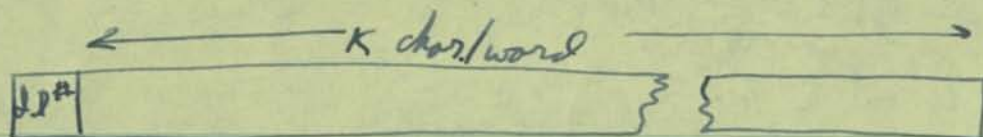


Assume a collection of  $10^6$  items, an allowable false drop ( $E_{max}$ ) = 100,  
and a least number of terms ( $L$ ) = 3.

<u>M (max. no. descriptors)</u>	<u>N (binary marks/descriptor)</u>	<u>S (total binary digits required)</u> <span style="float: right;">excluding the 10<sup>6</sup></span>
3	4	14
6	4	29
12	4	69
20	4	96
40	4	191



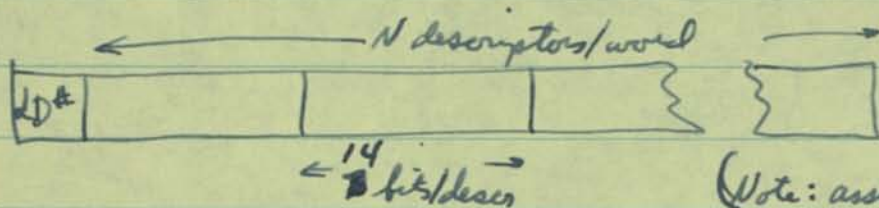
## Alphabetic Descriptor Spelling with Variable Descriptor Length but Fixed Word Length



K (alpha char/word)      total K bits (6 bits/char.)

20	120
100	600
200	1200

## Binary Descriptor Spelling with Fixed Descriptor & Word Lengths



(Note: assume a dictionary of  $2^{14} = 16,384$  descriptors.)

N (descrip./word)      total bits for indexing (not including the ID#)

1	14
10	140
20	280
40	560



# Prime Product Indexing

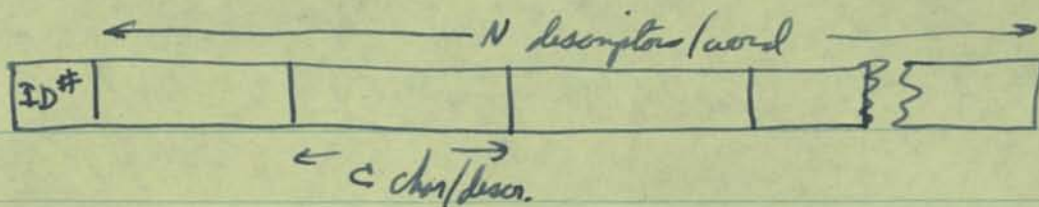
Including the 10<sup>th</sup>

LD#	prime product
-----	---------------

Vocabulary size (descriptors)	N (descriptors/word)	largest magnitude product	total descriptors required	total descriptors required to the 10 <sup>th</sup> power	total descriptors required to the 10 <sup>th</sup> power
5000	1	48,593	5	20	16
<del>5000</del>	2	$2.36 \times 10^9$	10	40	32
	4	$5.57 \times 10^{18}$	19	76	63
	6	$1.32 \times 10^{28}$	29	116	95
	10	$7.38 \times 10^{46}$	47	188	156
	12	$1.74 \times 10^{56}$	57	228	187
	20	$5.44 \times 10^{93}$	94	396	312
	40	$2.96 \times 10^{187}$	188	752	623
10000	1	$1.05 \times 10^5$	6	24	17
	2	$1.1 \times 10^{10}$	11	44	34
	4	$1.21 \times 10^{20}$	21	84	67
	6	$1.33 \times 10^{30}$	31	124	101
	10	$1.61 \times 10^{50}$	51	204	167
	12	$1.8 \times 10^{60}$	61	244	200
	20	$2.1 \times 10^{75}$	76	304	250
	40	$2.59 \times 10^{100}$	101	404	330
	40	$6.71 \times 10^{200}$	201	804	668



# alphanumeric<sup>Descriptor</sup> Spelling with Fixed Descriptor & Word Lengths



$N$ (desc./word)	$C$ (alpha char/descr.)	total C bits (6 bits/char)	total bits for word (including ID#)
1	4	24	24
	5	30	30
	6	36	36
	10	60	60
	20	120	120
	30	180	180
2	40	240	240
	4	24	48
	5	30	60
	6	36	72
	10	60	120
	20	120	240
4	30	180	360
	40	240	480
	4	24	96
	5	30	120
	6	36	144
	10	60	240
6	20	120	480
	30	180	720
	40	240	960
	4	24	144
	5	30	180
	6	36	216
10	10	60	360
	20	120	720
	30	180	1080
	40	240	1320
	4	24	240
	5	30	300
20	6	36	360
	10	60	600
	20	120	1200
	30	180	1800
	40	240	2400
	4	24	480
40	5	30	600
	6	36	720
	10	60	1200
	20	120	2400
	30	180	3600
	40	240	4800
	4	24	960
	5	30	1200
	6	36	1440
	10	60	2400
	20	120	4800
	30	180	7200
	40	240	9600