C. P. Bourne

SUPPLEMENT A TO QUARTERLY REPORT 4 CONTRACT AF 30(602)-2142

RANDOM SELECTION RATES FOR SINGLE-FIELD SUPERIMPOSED CODING

By: Richard C. Singleton

Prepared for:

ROME AIR DEVELOPMENT CENTER AIR RESEARCH AND DEVELOPMENT COMMAND
GRIFFISS AIR FORCE BASE ROME, NEW YORK

STANFORD RESEARCH INSTITUTE

MENLO PARK, CALIFORNIA

SRI

STANFORD RESEARCH INSTITUTE

MENLO PARK, CALIFORNIA

SRI

November 1960

Supplement A to Quarterly Report 4

RANDOM SELECTION RATES FOR SINGLE-FIELD SUPERIMPOSED CODING

By: Richard C. Singleton

SRI Project 3101

Prepared for:

ROME AIR DEVELOPMENT CENTER
GRIFFISS AIR FORCE BASE

AIR RESEARCH AND DEVELOPMENT COMMAND
ROME, NEW YORK

Contract AF 30(602)-2142

Approved:

J. Reid Anderson

J. Reid Anderson
Manager, Computer Techniques Laboratory

Jerre D. Noe
Assistant Director of Engineering Research

Copy No. 19

ABSTRACT

In the design of single-field superimposed coding systems for information retrieval, it is necessary to obtain estimates of the average number of unwanted entries that will be selected from a document file during a search. It has been customary to base these estimates on approximate solutions of a mathematical model of the system. In this report, a computational procedure for obtaining an exact solution of this mathematical model is described; this procedure is based on an application of the theory of Markov processes.

CONTENTS

TABLES

RANDOM SELECTION RATES FOR SINGLE-FIELD SUPERIMPOSED CODING

I INTRODUCTION

Several procedures have been proposed for coding the contents of documents in a file so that those pertaining to a selected combination of categories can be identified by a subsequent search. In one method in use, the "Zatocoding" system,[1*] each subject category is represented by a unique pattern of N *ones* (*i.e.*, marked positions) in a field of fixed length F, called a descriptor; the balance of the field is filled *zeros*. These descriptors are originally chosen at random from the collection of all $V(F,N) = \binom{F}{N} = F!/N!(F-N)!$ possible descriptor patterns, called here the vocabulary. The individual subject descriptors for each document are combined, by taking their logical sum, into a composite descriptor for the document.

To perform a quiz of the file to identify those documents pertaining to a specified combination of subjects, the descriptors for these subjects are combined by forming their logical sum, and the search process locates those file items having descriptors containing a *one* in every position in which the quiz descriptor has a *one*.

For example, a particular document might be coded as pertaining to subjects A, B, C, and D, as follows:

1 0 0 1 0 0 0 0 0 0	Subject A
0 1 0 0 0 0 0 0 1 0	Subject B
0 0 1 0 1 0 0 0 0 0	Subject C
0 0 0 1 0 0 0 0 1 0	Subject D
1 1 1 1 1 0 0 0 1 0	Composite Descriptor.

[*] References are listed at the end of the text.

Then if the file is searched for all documents pertaining to both subjects B and D, the composite descriptor

$$\begin{array}{r} 0\;1\;0\;0\;0\;0\;0\;0\;1\;0 \\ 0\;0\;0\;1\;0\;0\;0\;0\;1\;0 \\ \hline 0\;1\;0\;1\;0\;0\;0\;0\;1\;0 \end{array} \;,$$

formed by taking the logical sum of the descriptors for subjects B and D, will select the above document.

In addition to the desired file entries, the search process will ordinarily result in the selection of some entries which do not correspond to the quiz. The average proportion of the file appearing as unwanted selections can be controlled in the original design of the system. Increasing the field length, F, will reduce the proportion of unwanted selections, at the expense of handling descriptors of increased length. Increasing the number of descriptors combined to form the quiz will also reduce the proportion of unwanted selections, but at the same time it will increase the likelihood that a desired file entry would be overlooked in the search. Similarly, a change in any of the other design parameters will result in a change in the proportion of unwanted selections. However, in order to determine the set of design parameters which is optimum for a given set of cost functions, it is first necessary to be able to estimate the average proportion of the file which will be selected as unwanted entries during.a search.

One approach to estimating the proportion of unwanted entries selected is to describe the system by an idealized mathematical model, and to calculate the proportion on the basis of this model. This approach is taken here. Another possible approach would be to collect experience data from systems in actual use.

In the mathematical model treated here, it is assumed that the file being searched is composed of a small number of desired entries, corresponding to the quiz performed, and with the balance of the file made up by combining descriptors selected at random, independent of the search descriptors. Ordinarily the probability of constructing an additional desired entry by this random process is very small compared with the probability of constructing an entry that will be selected as unwanted by the search process;* thus, the former probability is neglected, and the probability of selecting an unwanted entry is estimated by the probability of selecting an entry from a randomlv constructed file.

In one model, referred to here as Model I, it is assumed that the sampling process used to construct the random file is carried out with replacement. Under this assumption, Wise[2] has derived an approximation, and Mooers[1] an upper bound, to the probability of selecting an unwanted file entry; these calculations are both based on the average number of *ones* in a file entry descriptor. However, in order to calculate the exact probability of selecting an unwanted file entry, one must determine first the actual probability distribution of the number of *ones* in a file entry descriptor. A method is given here for obtaining this probability distribution, and the use of this distribution in the calculation of random selection probabilities is demonstrated. The approach used is basically that indicated in an earlier paper by Mooers.[3] The mathematical model implied in Mooers' paper is here formulated explicitly, and the theory of Markov processes is used to formulate practical computation procedures.

The alternative model in which the sampling process used to construct the random file is carried out without replacement, referred to here as Model II, has been studied by Orosz and Takács.[4] They derive the probability distribution of the number of *ones* in a composite descriptor for that model.

For the range of parameter values of usual interest, Models I and II lead to essentially identical probability distributions. Model I is adopted here, since it appears easier to use in computing actual numerical results.

The method of calculating random selection rates, using the probability distribution determined according to either Model I or II, is shown in Sec. II. In Sec. III, the method of computing the probability distribution of the number of *ones* in a composite descriptor under Model I is derived. In Sec. IV, the results for Model II are stated without proof. A simple example is carried out in Sec. V to illustrate the calculation of probability distributions and random selection rates under Model I. In Sec. VI, the possible use of the results of this analysis in the design of an optimum system is discussed briefly. Finally in Sec. VII, possible modifications to improve the mathematical model are suggested.

* For Model II, the probability of constructing an additional desired entry, is $\binom{V-L}{M-L}\Big/\binom{V}{M}$. For Model I, this probability is even slightly smaller.

II CALCULATION OF SELECTION RATE

First, a method will be shown for calculating the probability of selecting a random file entry with a given number of *ones* in its composite descriptor as a result of a search composed of a given number of *ones*. Then this calculation is extended to the case in which the number of *ones* in the file entry and in the search are given as random variables with known probability distributions, rather than as fixed numbers.

Suppose that a search of the file is being made, with exactly j *ones* in the composite search descriptor. Then if a file entry with exactly i *ones* in its composite descriptor is chosen at random, with each of the $\binom{F}{i}$ possible patterns of the i *ones* equally likely, the probability that the file entry will be selected by the search is

$$R_{i,j}(F) = \begin{cases} \dfrac{\binom{i}{j}}{\binom{F}{j}} & \text{for } 0 \leq j \leq i \text{ and } N \leq i \leq F \\ 0 & \text{otherwise .} \end{cases} \qquad (1)$$

These values can be arrayed in an $F + 1$ by $F + 1$ selection probability matrix $R(F)$; as indicated, this matrix is a function only of the field size F.

Now if one descriptor is selected at random from the total vocabulary, it will have N *ones*, with probability one. If a second descriptor is selected at random from the total vocabulary and combined with the first to form a composite descriptor, the number of *ones* in the composite descriptor is not known with certainty. However, the probability distribution for the number of *ones* can be computed. Methods for computing this distribution under two different assumptions are shown in Secs. III and IV. This distribution can be arranged as a $F + 1$ by one column matrix $Q(2, F, N)$, with elements

$$Q_{i,2}(F, N) = Pr(i \textit{ ones} \text{ in composite of 2 descriptors, each with } N \textit{ ones} \text{ in a field of length } F). \qquad (2)$$

Similarly, the probability distribution of the number of *ones* in the composite descriptor after K descriptors have been combined can be represented as an $F + 1$ by one column matrix $Q(K, F, N)$; as indicated, this distribution is a function only of K, F, and N.

If it is assumed that each file entry descriptor is formed by combining exactly M vocabulary descriptors, the number of *ones* in the composite descriptor for a file entry will have a probability distribution which can be represented as above by the column matrix $Q(M, F, N)$. If this matrix is pre-multiplied by the transpose $R^t(F)$ of $R(F)$, then the resulting $F + 1$ by one column matrix

$$S(M, F, N) = R^t(F)Q(M, F, N) \tag{3}$$

will have as its elements

$$\begin{aligned} S_i(M, F, N) &= Pr(\text{selection of a randomly chosen file entry, given that the search descriptor contains exactly } i \text{ } \textit{ones}) \\ &= \sum_{j=0}^{F} R_{i,j}(F)Q_{j,M}(F, N) \quad . \end{aligned} \tag{4}$$

In the design of information retrieval systems, these values are useful in estimating the expected rate of selecting unwanted file entries during a search on a quiz descriptor containing i *ones*. Methods of calculating an approximate value of $S_i(M, F, N)$ are given by Mooers[1] and Wise;[2] these approximations are based on the mean number of *ones* in a file entry rather than on the probability distribution of the number of *ones*. (In a later paper,[3] however, Mooers suggests calculating the exact selection rate by essentially the method followed here.)

A problem closely related to the above is that of estimating the expected rate of selecting unwanted file entries during a search on an arbitrarily chosen descriptor formed by combining L vocabulary descriptors. The theoretical analysis given here leads to a useful answer to this problem. If a quiz descriptor is formed by combining L descriptors, chosen at random from the vocabulary, the probability distribution of the number of ones in the quiz descriptor can be represented by the column matrix $Q(L, F, N)$. Then the probability of selecting an arbitrarily chosen file entry using this arbitrarily selected quiz is given by the single number

$$
\begin{aligned}
D(L, M, F, N) &= Q^{t}(L, F, N)S(M, F, N) \\
&= Q^{t}(L, F, N)R^{t}(F)Q(M, F, N) \\
&= \sum_{i=0}^{F} \sum_{j=0}^{F} Q_{i,L}(F, N)R_{i,j}(F)Q_{j,M}(F, N) \quad . \qquad (5)
\end{aligned}
$$

To understand the meaning of this number, it may help the reader to consider the following conceptual experiment. Suppose that a sample of size M is selected at random from the vocabulary, where all possible samples of size M are equally likely to be drawn. Then suppose that a second sample of size L is selected at random from the vocabulary, where all possible samples of size L are equally likely. If the composite descriptor for each sample is constructed by forming the logical sum of the individual descriptors for that sample, $D(L, M, F, N)$ is the probability that the composite descriptor for the sample of size M has a *one* in every position for which there is a *one* in the composite descriptor for the sample of size L. It is not specified at this point whether or not a sample may contain duplications of descriptors, *i.e.*, in the usual terminology, whether the sampling is done with replacement or without; this difference in concept distinguishes the approaches used in Secs. III and IV, respectively, to calculate the probability distributions $Q(K, F, N)$.

The above analysis can be extended to the case in which the file entries are not all coded with the same number of descriptors. If the maximum number of descriptors used is H, and if the probability distribution of the proportion of entries with each number of descriptors is given by the H by one column matrix $\mathfrak{m}$, where

$$
\mathfrak{m}_k = Pr(M = k) \quad \text{for} \quad k = 1, 2, \ldots H \quad , \qquad (6)
$$

then the probability of selecting an arbitrary file entry with a quiz composed of L descriptors is given by

$$
\begin{aligned}
D(L, \mathfrak{m}, F, N) &= Q^{t}(L, F, N)R^{t}(F)Q(F, N)\mathfrak{m} \\
&= \sum_{i=0}^{F} \sum_{j=0}^{F} \sum_{k=1}^{H} Q_{i,L}(F, N)R_{i,j}(F)Q_{j,k}(F, N)\mathfrak{m}_k \quad . \qquad (7)
\end{aligned}
$$

In this expression, $Q(F, N)$ is the $F + 1$ by H matrix with columns $Q(1, F, N)$, $Q(2, F, N)$, ... $Q(H, F, N)$. Similarly, if a variable number of descriptors are combined in forming searches of the file, and if the probability distribution of the proportion of quizzes with each number of descriptors is given by the H by one column matrix $\mathfrak{L}$, then the probability of selecting an arbitrary file entry with an arbitrary quiz is

$$\begin{aligned} D(\mathfrak{L}, \mathfrak{m}, F, N) &= \mathfrak{L}^{t} Q^{t}(F, N) R^{t}(F) Q(F, N) \mathfrak{m} \\ &= \sum_{i=0}^{F} \sum_{j=0}^{F} \sum_{k=1}^{H} \sum_{l=1}^{H} \mathfrak{L}_{l} Q_{i,l}(F, N) R_{i,j}(F) Q_{j,k} \mathfrak{m}_{k} \quad . \end{aligned} \tag{8}$$

III PROBABILITY DISTRIBUTION OF NUMBER OF ONES IN COMPOSITE DESCRIPTOR—MODEL I

In this section, the probability distribution $Q(K, F, N)$ of the number of *ones* in a composite descriptor formed by combining K descriptors, each with N *ones*, selected at random with replacement from the total vocabulary, is obtained. Each of the $V^K/K!$ possible samples are considered to be equally likely.

If the sequence $\{Y(K)\}$, $(K = 1, 2, \ldots)$, of random variables is considered, where $Y(K)$ represents the number of *ones* in the composite descriptor after K descriptors have been combined, then it is observed that the sequence $\{Y(K)\}$ forms a Markov chain with stationary transition probabilities (see Chapt. XV of Feller[5]). In other words, the random variable $Y(K + 1)$ given $Y(K)$ depends only on the value of $Y(K)$, and not on K or on the values of $Y(1), Y(2), \ldots, Y(K - 1)$. The one-step Markov transition probabilities for this process are given by the $F + 1$ by $F + 1$ matrix $P(F, N)$ with elements

$$P_{i,j}(F, N) = \frac{\binom{i}{N - j + i}\binom{F - i}{j - i}}{\binom{F}{N}} \quad \begin{array}{lcl} \text{for} & j = & 0, 1, \ldots F \\ \text{and} & i = & 0, 1, \ldots F \end{array} \qquad (9)$$

$$= Pr(\text{addition of one descriptor increases the number of } \textit{ones} \text{ in composite from } i \text{ to } j),$$

where the usual extended factorial function is used to evaluate the binomial coefficients. It should be noted that the matrix $P(F, N)$ depends on F and N, but not on K. If the $F + 1$ by one matrix $Q(K, F, N)$ is identified with the probability distribution of $Y(K)$, then these probability distributions are obtained by successively forming the matrix products

$$\begin{aligned} P^t(F, N)\, Q(1, F, N) &= Q(2, F, N) \\ P^t(F, N)\, Q(2, F, N) &= Q(3, F, N) \qquad (10) \\ &\vdots \\ P^t(F, N)\, Q(K, F, N) &= Q(K + 1, F, N) \quad , \\ &\vdots \end{aligned}$$

where the initial distribution $Q(1, F, N)$ is

$$Q_{i,1}(F, N) = \begin{cases} 1 & \text{for} \quad i = N \\ 0 & \text{otherwise.} \end{cases} \qquad (11)$$

The mean and variance of $Y(K)$ can be calculated from the distribution $Q(K, F, N)$, once it is obtained. However, it is also possible to obtain them by a different line of reasoning, without computing the explicit distribution of $Y(K)$.(see Chapt. IX of Feller[5]). If

$$X_i(K) = \begin{cases} 1 & \text{if the } i\text{th position of composite descriptor has a } \textit{one} \\ 0 & \text{otherwise} \end{cases} \qquad (12)$$

for

$$i = 1, 2, \ldots F,$$

then

$$Y(K) = \sum_{i=1}^{F} X_i(K) \quad .$$

Also,

$$\begin{aligned} Pr[X_i(K) = 1] &= 1 - Pr[X_i(K) = 0] \\ &= 1 - Pr(\textit{zero} \text{ in } i\text{th position for all } K \text{ descriptors}) \\ &= 1 - \left(\frac{F - N}{F}\right)^K \quad \text{for} \quad i = 1, 2, \ldots F \quad . \qquad (13) \end{aligned}$$

Thus the mean of $Y(K)$ is

$$\begin{aligned} E[Y(K)] &= \sum_{i=1}^{F} E[X_i(K)] \\ &= F\left[1 - \left(\frac{F - N}{F}\right)^K\right] \quad . \qquad (14) \end{aligned}$$ ←

Wise[2] substitutes this mean into Eq. (1) (as i) to obtain an approximation for Eq. (3). He appears to first round the value of the mean to the nearest integer. However, it should be noted that there is no need to round, since the function on the right side of Eq. (1) can be extended in the usual way to non-integer values of i, using the extended factorial or gamma function. It is likely that use of the unrounded mean would reduce the approximation error in most cases.

Continuing as above, the variance of $Y(K)$ is

$$\begin{aligned} Var[Y(K)] &= E\{Y(K) - E[Y(K)]\}^2 \\ &= E\left(\left\{\sum_{i=1}^{F} X_i(K) - E[Y(K)]\right\}\left\{\sum_{j=1}^{F} X_j(K) - E[Y(K)]\right\}\right) \\ &= \sum_{i=1}^{F} E[X_i^2(K)] + \sum_{i \neq j} E[X_i(K)X_j(K)] - \{E[Y(K)]\}^2 \ . \qquad (15) \end{aligned}$$

The first term on the right side of this equation is evaluated as

$$\sum_{i=1}^{F} E[X_i^2(K)] = \sum_{i=1}^{F} P_r[X_i(K) = 1] = E[Y(K)] \ . \qquad (16)$$

The second term is

$$\begin{aligned} \sum_{i=j} E[X_i(K)X_j(K)] &= \sum_{i \neq j} Pr(X_i = 1, X_j = 1) \\ &= \sum_{i \neq j} Pr(X_i = 1) Pr(X_j = 1 | X_i = 1) \\ &= \sum_{i \neq j} \sum_{k=1}^{K} \binom{K}{k} \left(\frac{N}{F}\right)^k \left(\frac{F-N}{F}\right)^{K-k} \left[1 - \left(\frac{F-N}{F-1}\right)^k \left(\frac{F-1-N}{F-1}\right)^{K-k}\right] \\ &= F(F-1)\left\{1 - \left(\frac{F-N}{F}\right)^K - \left(\frac{F-N}{F}\right)^K \left[1 - \left(\frac{F-1-N}{F-1}\right)^K\right]\right\} \ . \end{aligned}$$

(17)

Thus, substituting Eqs. (14), (16), and (17) into Eq. (15), and simplifying, the result

$$
\begin{aligned}
Var[Y(K)] &= F\left(\frac{F-N}{F}\right)^{K}\left[1 - F\left(\frac{F-N}{F}\right)^{K} + (F-1)\left(\frac{F-1-N}{F-1}\right)^{K}\right] \\
&= F\left(\frac{F-N}{F}\right)^{K}\sum_{k=2}^{K}\binom{K}{k}N^{k}\left[\frac{1}{(F-1)^{k-1}} - \frac{1}{F^{k-1}}\right] \qquad (18)
\end{aligned}
$$

is obtained. The second line form of Eq. (18) is sometimes easier to evaluate than the first.

IV PROBABILITY DISTRIBUTION OF THE NUMBER OF *ONES* IN COMPOSITE DESCRIPTOR—MODEL II

In the previous section, it was assumed that composite descriptors were formed by selecting descriptors at random from the total vocabulary, sampling with replacement. In other words, the possibility that not all descriptors selected to form a composite were different was admitted. Here, the assumption is made that all of the descriptors selected are different.

Orosz and Takács[4] consider this model for the more general case of an arbitrary number of subfields. For the present case of a single field, they obtain the probabilities

$$Q^*_{i,k}(F,N) = \binom{F}{i} \sum_{j=F-i}^{F} (-1)^{j-F+i} \binom{i}{F-j} \frac{\binom{V(F-j,N)}{K}}{\binom{V(F,N)}{k}} \qquad (19)$$

corresponding to the $Q_{i,k}(F,N)$ of Model I above, where

$$V(F-j,N) = \binom{F-j}{N} \qquad \text{for } i = 0,\ 1,\ \ldots\ F-N \qquad (20)$$

is the vocabulary size if i specified positions are *zero* in each descriptor. They show that the mean of this distribution is

$$E[Y^*(K)] = F\left\{1 - \frac{\binom{V(F-1,N)}{K}}{\binom{V(F,N)}{K}}\right\}, \qquad (21)$$

and the variance is

$$\mathrm{Var}\ [Y^*(k)] = F(F-1)\frac{\binom{V(F-2,N)}{K}}{\binom{V(F,N)}{K}} + F^2\frac{\binom{V(F-1,N)}{K}}{\binom{V(F-1,N)}{K}}\left\{1 - \frac{\binom{V(F-1,N)}{K}}{\binom{V(F,N)}{K}}\right\}. \qquad (22)$$

From the point of view of describing the underlying information coding process, Model II is probably preferable to Model I. Duplications in descriptor assignment in Model I result in a small downward shift in the probability distribution of the number of *ones* in the composite descriptor, as compared with Model II, thus increasing the values in the selection matrix $S(M, F, N)$ given by Eq. (3). However, this change will be very slight in the parameter range of usual interest. The probability that a file entry composed of M descriptors, selected at random with replacement, contains one or more duplications of descriptors is

$$P_r \text{ (duplication)} = 1 - \frac{\binom{V(F,N)}{M}}{\frac{V^M(F,N)}{M!}} \quad . \qquad (23)$$

For example, for $F = 40$, $N = 4$, and $M = 6$,

$$V(40, 4) = \binom{40}{4} = 91,390$$

and

$$P_r \text{ (duplication)} \approx \frac{M(M-1)}{2V(F,N)} = 0.000164 \quad .$$

This probability is negligible.

From the point of view of ease of computation, Model I appears to be at an advantage with respect to Model II. Thus Model I is used here to obtain the probability distributions for use in computations of random selection probabilities.

V EXAMPLE OF COMPUTATION OF RANDOM SELECTION PROBABILITIES

Here, the calculation of random selection probabilities is shown in detail, under the assumptions of Model I. The case $F = 10$, $N = 2$, and $M = 4$ will be considered; small numbers are chosen so that the various arrays can be shown in full detail.

For this case, the one-step Markov transition probability matrix $P(10, 2)$ is given by the entries $P_{i,j}(10, 2)$ in Table I. Then the probability distributions $Q(K, 10, 2)$ for the number of *ones* in the composite of K descriptors, calculated according to Eq. (10), are given by the entries $Q_{i,k}(10, 2)$ in Table II. The matrix $R(10)$ of selection probabilities, calculated from Eq. (1) is given by the entries $R_{i,j}(F)$ in Table III, where i represents the number of *ones* in the composite file descriptor and j represents the number of *ones* in the composite quiz descriptor. The symmetry of this matrix about the 45 degree angle should be noted.

Forming the random selection probabilities $S(4, 10, 2)$ according to Eq. (3), one obtains the values $S_i(4, 10, 2)$ shown in Table IV. The entries $S_i(4, 10, 2)$ give the probability of selecting a file entry by chance with a quiz containing i *ones*. Values computed by two approximation methods are also listed in Table IV for comparison. Then calculating $D(L, 4, 10, 2)$ according to Eq. (7), one obtains the probability of selecting a randomly chosen file entry with a quiz composed of L descriptors; these probabilities are listed in Table V.

These calculations have been programmed in ALGOL, and tables run on the Burroughs 220 computer for a number of cases of interest. The results indicate in general that the Wise approximation underestimates and the Mooers upper bound overestimates the random selection probabilities. As an example, for a field of length 40, 2 *ones* per descriptor, file entries each composed of 10 descriptors, and a quiz with 12 *ones*, the actual selection probability is 1.15×10^{-4}, the Wise approximation is 3.48×10^{-5}, and the Mooers upper bound is 1.74×10^{-3}. If one wishes to use the present model as a basis for system design, it would appear desirable to calculate exact probabilities.

TABLE I

ONE-STEP MARKOV TRANSITION PROBABILITIES $P_{i,j}(10, 2)$

j =	0	1	2	3	4	5	6	7	8	9	10
i = 0			1.000								
1			0.200	0.800							
2			0.022	0.356	0.622						
3				0.067	0.467	0.467					
4					0.133	0.533	0.333				
5						0.222	0.556	0.222			
6							0.333	0.533	0.133		
7								0.467	0.467	0.067	
8									0.622	0.356	0.022
9										0.800	0.200
10											1.000

TABLE II

PROBABILITY DISTRIBUTIONS $Q(K, 10, 2)$ OF NUMBER OF *ONES*

K =	1	2	3	4
i = 0				
1				
2	1.000	0.022	0.000	0.000
3		0.356	0.032	0.002
4		0.622	0.263	0.050
5			0.498	0.265
6			0.207	0.433
7				0.221
8				0.028
9				
10				
Expected Value	2.000	3.600	4.880	5.904
Variance	0.00	0.28	0.59	0.81

TABLE III

SELECTION PROBABILITIES $R_{i,j}(10)$

j = 0	0	1	2	3	4	5	6	7	8	9	10
i = 0	1.000										
1	1.000	0.100									
2	1.000	0.200	0.022								
3	1.000	0.300	0.067	0.008							
4	1.000	0.400	0.133	0.033	0.005						
5	1.000	0.500	0.222	0.083	0.024	0.004					
6	1.000	0.600	0.333	0.167	0.071	0.024	0.005				
7	1.000	0.700	0.467	0.292	0.167	0.083	0.033	0.008			
8	1.000	0.800	0.622	0.467	0.333	0.222	0.133	0.067	0.022		
9	1.000	0.900	0.800	0.700	0.600	0.500	0.400	0.300	0.200	0.100	
10	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

TABLE IV

RANDOM SELECTION PROBABILITIES *vs* NUMBER OF *ONES* IN QUIZ

	$S_i(4, 10, 2)$	WISE APPROX.*	MOOERS APPROX.†
$i = 0$			
1			
2	0.331	0.322	0.349
3	0.173	0.157	0.206
4	0.084	0.065	0.122
5	0.036	0.021	0.072
6	0.013	0.004	0.042
7	0.004	- 0.001	0.025
8	0.001	0.000	0.015
9			
10			

* Using expected value of 5.904 *ones* per file descriptor,

$$\text{Approx. } S_i(4,\ 10,\ 2) = \frac{\binom{5.904}{i}}{\binom{10}{i}}.$$

† Using expected value of 5.904 *ones* per file descriptor,

$$\text{Approx. } S_i(4,\ 10,\ 2) = \left(\frac{5.904}{10}\right)^i ;$$

Mooers gives this formula as an upper bound for $S_i(M,\ F,\ N)$.

TABLE V

RANDOM SELECTION PROBABILITIES *vs* NUMBER OF DESCRIPTORS IN QUIZ

	$D(L, 4, 10, 2)$
$L = 1$	0.331
2	0.121
3	0.048
4	0.021

VI OPTIMUM SYSTEM DESIGN

If Model I is adopted as a description of the physical system, the random selection probabilities calculated as above may be used as a basis for optimum design of a document coding system. The approach taken will depend on which parameters are assumed fixed and which variable, and on the costs associated with varying parameter values.

To take a single example, suppose that the field length F is fixed, and the probability distribution $\mathfrak{M}$ of the numbers of descriptors used to code file entries is known. Then if the cost of varying the number N of *ones* in a descriptor is neglected, the optimum value of N for a quiz of a given number of descriptors is found by determining that N which minimizes the random selection probability $D\ (L,\ \mathfrak{M},\ F,\ N)$ (subject, of course, to the practical restriction that the resulting available vocabulary size $V(F,N)$ be large enough to meet the requirements of the system). If the minimum random selection probability found for a given L is too large, then one must conclude, if no other parameters are to be changed, that a larger number of descriptors must be combined to perform a quiz. If, on the other hand, a probability distribution $\mathfrak{L}$ of the number of descriptors combined to perform quizzes of the file is given, then the optimum value of N is that which minimizes $D\ (\mathfrak{L},\ \mathfrak{M},\ F,\ N)$.

To consider another example, suppose that the distribution $\mathfrak{M}$ of the number descriptors combined to form file entries is given, and that the system is required to perform searches on a minimum number L of descriptors in a quiz, with a random selection probability not exceeding E. If any desired field length F may be used at an increasing cost $C_1(F)$ and any desired N may be used at an increasing cost $C_2(N)$, then the optimum values of F and N will be those which minimize $C_1(F) + C_2(N)$, subject to the restriction $D\ (L,\ \mathfrak{M},\ F,\ N) \leq E$.

In a similar manner, other optimization problems may be formulated, as appropriate to the particular design conditions encountered.

VII IMPROVED MODELS

Models I and II fail to take into consideration the fact that, in the usual document coding system, only a small portion of the potential vocabulary $V(F,N)$ of descriptors is actually used in constructing file entries, and consequently in constructing quizzes of the file. As a step toward a more realistic mathematical model, one might assume that a restricted vocabulary of a specified size is selected at random from the potential vocabulary, sampling without replacement. Then a random file would be constructed by selecting groups of descriptors at random, with equal probability, from this restricted vocabulary, sampling either with or without replacement. A quiz would be constructed from the restricted vocabulary in the same manner. It is conjectured that an analysis of this model would indicate higher random selection rates than obtained with Models I and II.

An additional refinement of the mathematical model would be to select file entry descriptors and quiz descriptors from the restricted vocabulary according to a probability distribution approximating the frequency of usage of descriptors in an actual file. It is conjectured that this refinement would further increase the calculated random selection rates.

REFERENCES

1. Calvin N. Mooers, "Zatocoding for Punched Cards," Zator Technical Bulletin No. 30, Zator Company, Boston (1950).

2. Carl S. Wise, "Mathematical Analysis of Coding Systems," Chapt. 21 of *Punched Cards, Their Applications to Science and Industry*, 2nd edition, edited by Robert S. Casey, James W. Perry, Madeline M. Berry, and Allen Kent (Reinhold Publishing Corporation, New York City, 1958).

3. Calvin N. Mooers, "The Exact Distribution of the Number of Positions Marked in a Zatocoding Field," Zator Technical Bulletin No. 73, Zator Company, Boston (1952).

4. G. Orosz and L. Takács, "Some Probability Problems Concerning the Marking of Codes into the Superimposition Field," *Journal of Documentation*, vol. 12, No. 4, pp. 231-34, (December 1956).

5. William Feller, *An Introduction to Probability Theory and Its Applications*, vol. 1, 2nd edition, (John Wiley and Sons, New York City, 1957).

STANFORD
RESEARCH
INSTITUTE

MENLO PARK, CALIFORNIA

REGIONAL OFFICES AND LABORATORIES

SOUTHERN CALIFORNIA LABORATORIES
820 Mission Street
South Pasadena, California

WASHINGTON OFFICE
711 14th Street, N. W.
Washington 5, D. C.

NEW YORK OFFICE
270 Park Avenue
New York 17, N. Y.

EUROPEAN OFFICE
Pelikanstrasse 37
Zurich 1, Switzerland

RESEARCH REPRESENTATIVES

HONOLULU, HAWAII
Finance Factors Building
195 South King Street
Honolulu, Hawaii

PORTLAND, OREGON
Suite 914, Equitable Building
421 Southwest 6th Avenue
Portland, Oregon

PHOENIX, ARIZONA
Suite 216, Central Plaza
3424 North Central Avenue
Phoenix, Arizona

10/67
From Bill Monroe - Access Corp.

RECAP SHEET

CHARLES BOURNE FORMULA FOR DETERMINING CHARACTERISTICS OF SUPERIMPOSED CODING IN INFO. RETRIEVAL

MAXIMUM NO. OF FALSE DROPS (EMAX)	NO. OF BITS PER TERM (m)	MIN. NO. OF BITS PER FIELD (F)	
1	4	69	10,000 DOCUMENTS 3 DESCRIPTORS PER DOCUMENT MIN. (L) 12 DESCRIPTORS PER DOCUMENT MAX. (M)
2	4	69	
3	4	69	
4	4	69	
5	4	69	
10	3	52	
20	3	52	
1	4	116	10,000 DOCUMENTS 3 DESCRIPTORS /DOC. MIN. (L) 20 DESCRIPTORS /DOC. MAX. (M)
2	4	116	
3	4	116	
4	4	116	
5	4	116	
10	3	87	
20	3	87	
1	4	173	10,000 DOCUMENTS 3 DESCRIPTORS /DOC. MIN. (L) 30 DESCRIPTORS /DOC. MAX (M)
2	4	173	
3	4	173	
4	4	173	
5	4	173	
10	3	130	
20	3	130	

FROM OCT. 4, '67 COMPUTER RUN.

NEW PARAMETERS - TO BE COMPUTED.

A. 10,000 DOCUMENTS.

SOLVE for (m) -

VALUES FOR (F) : 10, 20, 30, 45

VALUES FOR NO. OF DESCRIPTORS / DOCUMENT.

a - 3 (L) and 12 (M)
b. 3 (L) and 20 (M)
c. 3 (L) and 30 (M)
d. 12 (L) and 12 (M)
e. 20 (L) and 20 (M)
f. 30 (L) and 30 (M)

VALUES FOR EMAX

1, 2, 3, 4, 5, 10, 20

B. 10,000 DOCUMENTS.

VALUES FOR (m) : 3, 4, 5, 6, 7, 8

VALUES FOR (F) : 10, 20, 30, 45

VALUES FOR NO. OF DESCRIPTORS /DOCUMENT.

(SAME AS "A" ABOVE)

SOLVE FOR (EMAX).

ON THE RECAP SHEET

```
M=      12
C=     10000
E=        1L=           4F=          69
E=        2L=           4F=          69
E=        3L=           4F=          69
E=        4L=           4F=          69
E=        5L=           4F=          69
E=       10L=           3F=          52
E=       20L=           3F=          52
C=     30000
E=        1L=           5F=          87
E=        2L=           5F=          87
E=        3L=           4F=          69
E=        4L=           4F=          69
E=        5L=           4F=          69
E=       10L=           4F=          69
E=       20L=           4F=          69
C=     40000
E=        1L=           5F=          87
E=        2L=           5F=          87
E=        3L=           5F=          87
E=        4L=           4F=          69
E=        5L=           4F=          69
E=       10L=           4F=          69
E=       20L=           4F=          69
C=     50000
E=        1L=           5F=          87
E=        2L=           5F=          87
E=        3L=           5F=          87
E=        4L=           5F=          87
E=        5L=           4F=          69
E=       10L=           4F=          69
E=       20L=           4F=          69
C=     60000
E=        1L=           5F=          87
E=        2L=           5F=          87
E=        3L=           5F=          87
E=        4L=           5F=          87
E=        5L=           5F=          87
E=       10L=           4F=          69
E=       20L=           4F=          69
```

M = 12 C = 10,000
E = 1 L = 4 F = 69

E = NO. OF FALSE DROPS

L = NO. OF BITS PER TERM

F = NO. OF BITS PER FIELD

C = NO. OF DOCUMENTS IN FILE

M = MAXIMUM NO. OF DESCRIPTORS FOR SEARCHING.

3 = MINIMUM NO. OF DESCRIPTORS FOR SEARCHING.

What relationship exists to the total number of words in the vocabulary?

ON THE RECAP SHEET

```
M=        20
C=     10000
E=          1L=              4F=            116
E=          2L=              4F=            116
E=          3L=              4F=            116
E=          4L=              4F=            116
E=          5L=              4F=            116
E=         10L=              3F=             87
E=         20L=              3F=             87
C=     30000
E=          1L=              5F=            144
E=          2L=              5F=            144
E=          3L=              4F=            116
E=          4L=              4F=            116
E=          5L=              4F=            116
E=         10L=              4F=            116
E=         20L=              4F=            116
C=     40000
E=          1L=              5F=            144
E=          2L=              5F=            144
E=          3L=              5F=            144
E=          4L=              4F=            116
E=          5L=              4F=            116
E=         10L=              4F=            116
E=         20L=              4F=            116
C=     50000
E=          1L=              5F=            144
E=          2L=              5F=            144
E=          3L=              5F=            144
E=          4L=              5F=            144
E=          5L=              4F=            116
E=         10L=              4F=            116
E=         20L=              4F=            116
C=     60000
E=          1L=              5F=            144
E=          2L=              5F=            144
E=          3L=              5F=            144
E=          4L=              5F=            144
E=          5L=              5F=            144
E=         10L=              4F=            116
E=         20L=              4F=            116
```

ON THE RECAP SHEET.

```
M=      30
C=    10000
E=        1L=               4F=          173
E=        2L=               4F=          173
E=        3L=               4F=          173
E=        4L=               4F=          173
E=        5L=               4F=          173
E=       10L=               3F=          130
E=       20L=               3F=          130
C=    30000
E=        1L=               5F=          217
E=        2L=               5F=          217
E=        3L=               4F=          173
E=        4L=               4F=          173
E=        5L=               4F=          173
E=       10L=               4F=          173
E=       20L=               4F=          173
C=    40000
E=        1L=               5F=          217
E=        2L=               5F=          217
E=        3L=               5F=          217
E=        4L=               4F=          173
E=        5L=               4F=          173
E=       10L=               4F=          173
E=       20L=               4F=          173
C=    50000
E=        1L=               5F=          217
E=        2L=               5F=          217
E=        3L=               5F=          217
E=        4L=               5F=          217
E=        5L=               4F=          173
E=       10L=               4F=          173
E=       20L=               4F=          173
C=    60000
E=        1L=               5F=          217
E=        2L=               5F=          217
E=        3L=               5F=          217
E=        4L=               5F=          217
E=        5L=               5F=          217
E=       10L=               4F=          173
E=       20L=               4F=          173
```

How to Design the Superimposed Code

For relatively small retrieval systems, the user can generally adapt the systems and code parameters found to be successful by other users. However, for newer retrieval systems that require high performance of the superimposed coding system, a special study and code design may be in order. The design procedure is relatively simple, and considers the following parameters:[46]

- C the number of items in the total collection
- L the anticipated lower bound of the number of descriptors normally used for searching
- M the anticipated upper bound of the number of descriptors normally used for indexing
- R the tolerable noise ratio $= E_{\max}/C$
- $E_{\max}$ the maximum number of false drops with L search descriptors
- F the length of the single fixed field for the superimposed code

In terms of these parameters, each descriptor code pattern should contain m marks (or binary ones), where

$$m = \left\langle \left(\frac{1}{L}\right)(-\log_2 R) \right\rangle$$

$$= \left\langle \left(\frac{1}{L}\right)(3.31)(-\log_{10} R) \right\rangle$$

where the symbols $\langle\ \rangle$ mean that the nearest integral value is to be taken. The least number of sites (F) that must be used to contain M descriptors is

$$F = \langle 1.445mM \rangle$$

For a sample calculation, assume the following parameters:

File size (C) = one million items

Minimum number of descriptors used for searching (L) = 3

Maximum number of descriptors used to index each item (M) = 12

Maximum number of false drops tolerable with L search descriptors ($E_{\max}$) = 100

Tolerable noise ratio (R) = $E_{\max}/C$

$$R = \frac{E_{\max}}{C} = \frac{100}{10^6} = 10^{-4}$$

$$m = \left\langle \left(\frac{1}{L}\right)(3.31)(-\log_{10} 10^{-4}) \right\rangle \qquad L = 3$$

$$= \left\langle \frac{3.31}{3}(-1)(\log_{10} 10^{-4}) \right\rangle$$

$$= \left\langle \frac{3.31}{3}(-1)(-4) \right\rangle$$

$$= \langle 4.41 \rangle$$

$$= 4$$

and

$$F = \langle 1.445\ (4)\ (12) \rangle$$

$$= \langle 69.36 \rangle$$

$$= 69$$

Repeating this computation procedure for several different values of M, while keeping the same values of C, L, and $E_{\max}$ for this example, gives the following results:

Max. No. of Descriptors Used to Index Each Item (M)	Required No. of Marks per Descriptor (m)	No. of Code Positions Required (F)
3	4	14
6	4	29
12	4	69
20	4	96
40	4	191

The size of the coding field required, F, also varies with the size of the file. This slight variation is shown in Fig. 3-16, which illustrates the degree to which the specification for $E_{\max}$ influences the size of coding field required.

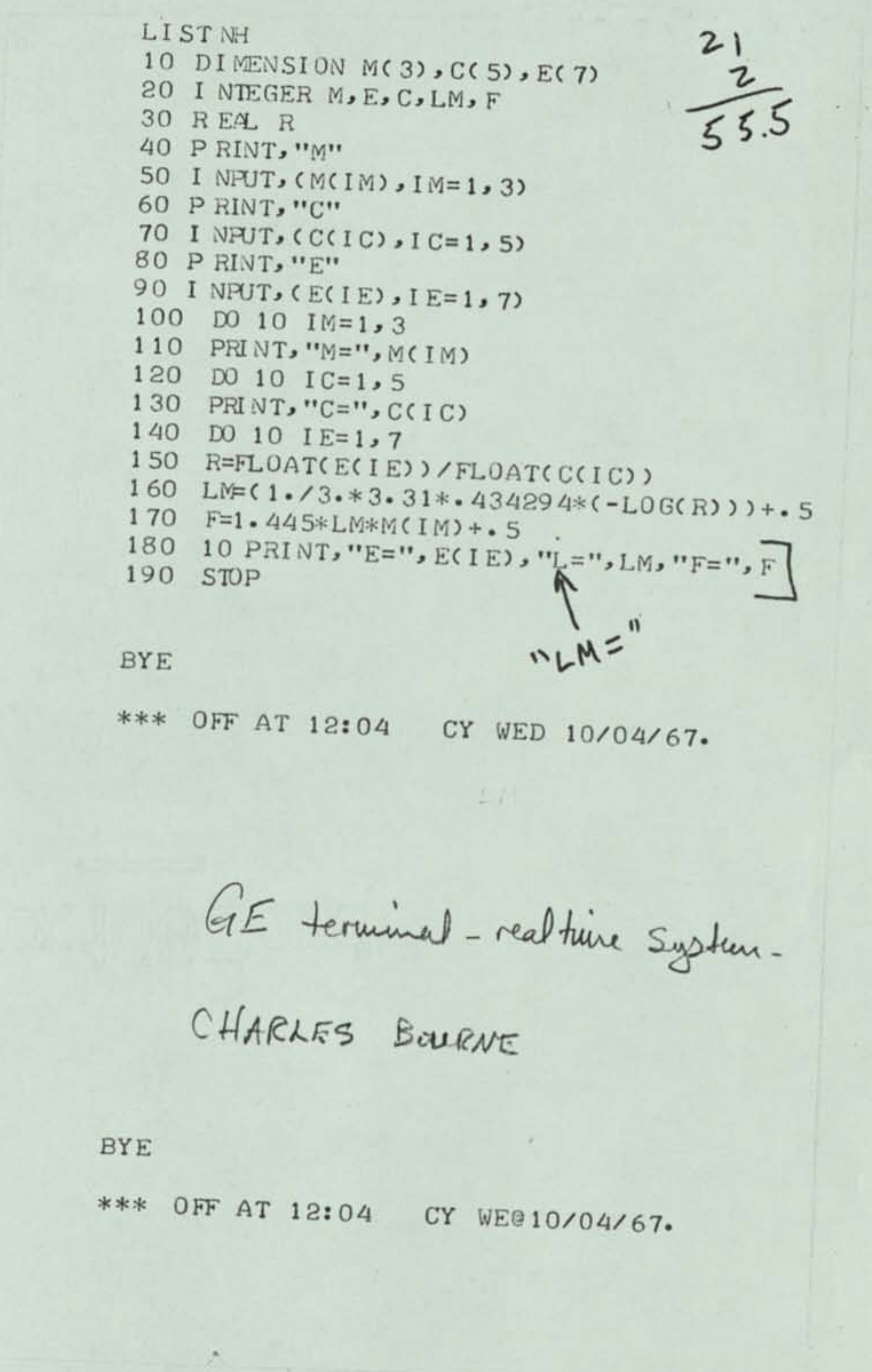
Additional References

Barnard, G. A., "Statistical Calculation of Word Entropies for Four Western Languages," *Institute of Radio Engineers Transactions of Professional Group on Information Theory,* Vol. 1, No. 1, pp. 49–53 (March 1955).

Barrett, J. A., and M. Grems, "Abbreviating Words Systematically," *Communications of the Association for Computing Machinery,* Vol. 3, No. 5, pp. 323–324 (May 1960).

Bemer, R. W., "Do It By the Numbers—Digital Shorthand," *Communications of the Association for Computing Machinery,* Vol. 3, No. 10, pp. 530–536 (October 1960).

[46] Mooers, C. N., *The Application of Simple Pattern Inclusion Selection to Large-Scale Information Retrieval Systems,* Technical Bulletin No. 131, Zator Co., Cambridge, Mass. (April 1959), AD-215 434.



```
LISTNH
10 DIMENSION M(3),C(5),E(7)
20 INTEGER M,E,C,LM,F
30 REAL R
40 PRINT,"M"
50 INPUT,(M(IM),IM=1,3)
60 PRINT,"C"
70 INPUT,(C(IC),IC=1,5)
80 PRINT,"E"
90 INPUT,(E(IE),IE=1,7)
100  DO 10 IM=1,3
110  PRINT,"M=",M(IM)
120  DO 10 IC=1,5
130  PRINT,"C=",C(IC)
140  DO 10 IE=1,7
150  R=FLOAT(E(IE))/FLOAT(C(IC))
160  LM=(1./3.*3.31*.434294*(-LOG(R)))+.5
170  F=1.445*LM*M(IM)+.5
180  10 PRINT,"E=",E(IE),"L=",LM,"F=",F
190  STOP

BYE

*** OFF AT 12:04    CY WED 10/04/67.
```

21
2
55.5

"LM="

GE terminal - real time system -

CHARLES BOURNE

```
BYE

*** OFF AT 12:04    CY WE@10/04/67.
```

(427) SUPERIMPOSABLE PUNCHED CARDS AS A MEANS OF REFERENCE TO PERIODICALS

In periodicals libraries, the problem of informing readers of the titles of available reviews is dealt with in a variety of ways, influenced by the number of titles and the characteristics of the collection. The simplest solution is to supply the reader with a list or card-index in which he will find the titles corresponding to what he wants. But when the number of reviews is considerable, searching becomes a difficult and lengthy business if the classification is based on a single characteristic (uni-dimensional classification). Let us suppose, for instance, that the reader is seeking information about reviews in German dealing with popular biology. Arrangement by languages will give him a complete list of reviews published in German, from which he will have to pick out those concerned with biology; having selected these, he will have to go through them again, to discover those which, besides being in German and dealing with biology, are also 'popular science' reviews. In brief, what the reader wants is to find several characteristics combined in a single review; and the problem is to find a document which combines the various conditions required.

Irrespective of the number of documents to be filed, the solution lies in the use of mechanical methods of selection. These methods are based on a very simple principle: the recording on a punched card (IBM, for instance), or on film,[1] of the distinctive features of each document, and the selection of all cards in the index which present all the desired features. But the method in general use in certain documentation services has one drawback—it requires expensive and bulky sorting machines, and a specially trained staff.

It is possible, however, to adopt another system, for which no machines are required: this is the system of superimposable punched cards, to which we at the Scientific and Technical Documentation Division of the National Research Centre of Egypt have had recourse in other instances.[2]

SUPERIMPOSABLE PUNCHED CARDS

This method is based on the following principle. Each document (review) is given a serial number, which may be simply its entry number; this number may also correspond to the document's position on the shelves, which will make it easier to find. We use IBM cards, but each card corresponds to a particular feature and not, as usual, to a particular document. We shall, for instance, have one card for English language reviews, another for those dealing with philosophy, etc.—in short, one card for each characteristic which may facilitate the search for a document. A single perforation denotes the serial number of each document. That number is expressed by a system of co-ordinates. The column indicates the hundreds and the tens of the number, and the position of the perforation in the column indicates the units (Fig. 1).

The card for the English language reviews will be perforated in the squares corresponding to the numbers of those reviews. An advantage of this card is that the numbers of reviews with any particular characteristic can be singled out immediately. There is one drawback, however—a card cannot take more than 800 reviews, so that a fresh card must be started for each batch of 800 reviews.

[1] J. Samain. *Onde électrique* (1956), XXXVI, p. 671-5.
[2] J. Garrido, *Bull. Soc. franç. Miner. Crist.* (1954), LXXVII, p. 989-95.

Unesco bull. libr., vol. XII, no. 10, October 1958.

SELECTION

Let us now suppose that we wish to select reviews combining two different features; we take the two cards corresponding to these two features, and by placing one on top of the other we obtain the necessary information, since the squares corresponding to reviews possessing both features will have been perforated on both cards. This method is, in fact, based on the same principle as the Cordonnier system, but it has the further advantage of using standard IBM cards, which are easy to reproduce, though it has the drawback of limited capacity.

This method is practical for collections containing comparatively few documents, characterized by features which are not capable of expression in a linear series. That is why we have adopted it for reference to reviews.

The distinctive features which we have selected for our collection of reviews come under the following headings: scientific speciality (82); country of publication (59); language (23); type of review (7); year of publication (88); frequency (13).

We have selected a total of 275 different features. Each card represents one feature and consists of two sections—one showing the numbers corresponding to the reviews, and the other, at the top of the card, containing certain extra perforations, the number of the card and the number of the series.

The production of the cards is an easy matter with the use of IBM machines. The first step is to make out a set of cards, with one card for each review; on each card we record, in code, all information relating to the corresponding review—i.e., all the distinctive features it displays. By the use of IBM sorting apparatus, we then pick out all the cards which have a common feature, and thus discover which numbers should appear on the card corresponding to that feature.

We have adopted this method for our collection of reviews, which contains about 1,500 titles, and it has been found useful both by the staff of the library and by readers, who soon learn how to employ it.

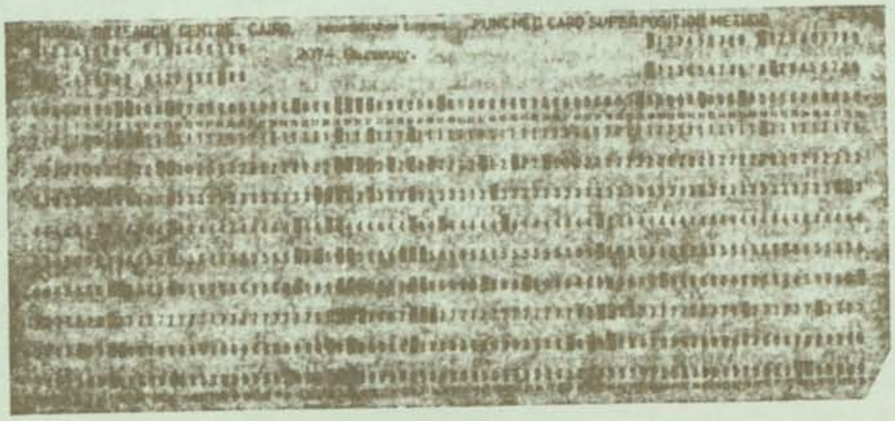


FIG. 1

STANFORD RESEARCH INSTITUTE

To:

Date: October 19, 1965

From: Charles P. Bourne

Location: 314 A

Subject: Note on Coding Method Suggested at October 1965 FID Conference by Dr. Pratt, National Cancer Institute, Washington, D.C.

Answering:

---

1. Assign a unique binary number to each term in the dictionary, e.g.,

DICTIONARY TERM	CODE NOTATION
1. APPLES	2^o
2. BEARS	2^1
⋮ ⋮	⋮
N. ZEBRAS	2^N

2. Code a document by summing all the weights for the relevant terms. E.g.,

APPLES	1	
BEARS	10	
	11	COMPOSITE INDEX TERM

3. A large dictionary leads to large numbers. (E.g., 2000th term = 2^{2000}). The 315th term requires a 95-digit decimal number. Some of this can be avoided by assigning the low value codes to the most frequently used terms.

4. This work is more suited to binary computers with chained work, than to decimal machines.

5. This notation seems to be the same as a non-ambiguous superimposed code--suggested earlier by others but rejected because practical problems of implementation.

CPB:sd

SUPPLEMENT A TO QUARTERLY REPORT 4

CONTRACT AF 30(602)-2142

Chas Bourne

3101

RANDOM SELECTION RATES FOR SINGLE-FIELD SUPERIMPOSED CODING

By: Richard C. Singleton

Prepared for:

ROME AIR DEVELOPMENT CENTER
GRIFFISS AIR FORCE BASE

AIR RESEARCH AND DEVELOPMENT COMMAND
ROME, NEW YORK

STANFORD RESEARCH INSTITUTE

MENLO PARK, CALIFORNIA

SRI

STANFORD RESEARCH INSTITUTE

SRI

MENLO PARK, CALIFORNIA

November 1960

Supplement A to Quarterly Report 4

RANDOM SELECTION RATES FOR SINGLE-FIELD SUPERIMPOSED CODING

By: Richard C. Singleton

SRI Project 3101

Prepared for:

ROME AIR DEVELOPMENT CENTER
GRIFFISS AIR FORCE BASE

AIR RESEARCH AND DEVELOPMENT COMMAND
ROME, NEW YORK

Contract AF 30(602)-2142

Approved:

J. Reid Anderson

J. Reid Anderson
Manager, Computer Techniques Laboratory

J Noe

Jerre D. Noe
Assistant Director of Engineering Research

Copy No. 6

ABSTRACT

In the design of single-field superimposed coding systems for information retrieval, it is necessary to obtain estimates of the average number of unwanted entries that will be selected from a document file during a search. It has been customary to base these estimates on approximate solutions of a mathematical model of the system. In this report, a computational procedure for obtaining an exact solution of this mathematical model is described; this procedure is based on an application of the theory of Markov processes.

CONTENTS

TABLES

RANDOM SELECTION RATES FOR SINGLE-FIELD SUPERIMPOSED CODING

I INTRODUCTION

Several procedures have been proposed for coding the contents of documents in a file so that those pertaining to a selected combination of categories can be identified by a subsequent search. In one method in use, the "Zatocoding" system,[1*] each subject category is represented by a unique pattern of N *ones* (*i.e.*, marked positions) in a field of fixed length F, called a descriptor; the balance of the field is filled *zeros*. These descriptors are originally chosen at random from the collection of all $V(F,N) = \binom{F}{N} = F!/N!(F-N)!$ possible descriptor patterns, called here the vocabulary. The individual subject descriptors for each document are combined, by taking their logical sum, into a composite descriptor for the document.

To perform a quiz of the file to identify those documents pertaining to a specified combination of subjects, the descriptors for these subjects are combined by forming their logical sum, and the search process locates those file items having descriptors containing a *one* in every position in which the quiz descriptor has a *one*.

For example, a particular document might be coded as pertaining to subjects A, B, C, and D, as follows:

1 0 0 1 0 0 0 0 0 0	Subject A
0 1 0 0 0 0 0 0 1 0	Subject B
0 0 1 0 1 0 0 0 0 0	Subject C
0 0 0 1 0 0 0 0 1 0	Subject D
1 1 1 1 1 0 0 0 1 0	Composite Descriptor.

[*] References are listed at the end of the text.

Then if the file is searched for all documents pertaining to both subjects B and D, the composite descriptor

$$\begin{array}{cccccccccc} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{array} \; ,$$

formed by taking the logical sum of the descriptors for subjects B and D, will select the above document.

In addition to the desired file entries, the search process will ordinarily result in the selection of some entries which do not correspond to the quiz. The average proportion of the file appearing as unwanted selections can be controlled in the original design of the system. Increasing the field length, F, will reduce the proportion of unwanted selections, at the expense of handling descriptors of increased length. Increasing the number of descriptors combined to form the quiz will also reduce the proportion of unwanted selections, but at the same time it will increase the likelihood that a desired file entry would be overlooked in the search. Similarly, a change in any of the other design parameters will result in a change in the proportion of unwanted selections. However, in order to determine the set of design parameters which is optimum for a given set of cost functions, it is first necessary to be able to estimate the average proportion of the file which will be selected as unwanted entries during a search.

One approach to estimating the proportion of unwanted entries selected is to describe the system by an idealized mathematical model, and to calculate the proportion on the basis of this model. This approach is taken here. Another possible approach would be to collect experience data from systems in actual use.

In the mathematical model treated here, it is assumed that the file being searched is composed of a small number of desired entries, corresponding to the quiz performed, and with the balance of the file made up by combining descriptors selected at random, independent of the search descriptors. Ordinarily the probability of constructing an additional desired entry by this random process is very small compared with the probability of constructing an entry that will be selected as unwanted by the search process;[*] thus, the former probability is neglected, and the probability of selecting an unwanted entry is estimated by the probability of selecting an entry from a randomly constructed file.

In one model, referred to here as Model I, it is assumed that the sampling process used to construct the random file is carried out with replacement. Under this assumption, Wise[2] has derived an approximation, and Mooers[1] an upper bound, to the probability of selecting an unwanted file entry; these calculations are both based on the average number of *ones* in a file entry descriptor. However, in order to calculate the exact probability of selecting an unwanted file entry, one must determine first the actual probability distribution of the number of *ones* in a file entry descriptor. A method is given here for obtaining this probability distribution, and the use of this distribution in the calculation of random selection probabilities is demonstrated. The approach used is basically that indicated in an earlier paper by Mooers.[3] The mathematical model implied in Mooers' paper is here formulated explicitly, and the theory of Markov processes is used to formulate practical computation procedures.

The alternative model in which the sampling process used to construct the random file is carried out without replacement, referred to here as Model II, has been studied by Orosz and Takács.[4] They derive the probability distribution of the number of *ones* in a composite descriptor for that model.

For the range of parameter values of usual interest, Models I and II lead to essentially identical probability distributions. Model I is adopted here, since it appears easier to use in computing actual numerical results.

The method of calculating random selection rates, using the probability distribution determined according to either Model I or II, is shown in Sec. II. In Sec. III, the method of computing the probability distribution of the number of *ones* in a composite descriptor under Model I is derived. In Sec. IV, the results for Model II are stated without proof. A simple example is carried out in Sec. V to illustrate the calculation of probability distributions and random selection rates under Model I. In Sec. VI, the possible use of the results of this analysis in the design of an optimum system is discussed briefly. Finally in Sec. VII, possible modifications to improve the mathematical model are suggested.

[*] For Model II, the probability of constructing an additional desired entry, is $\binom{V-L}{M-L}\Big/\binom{V}{M}$. For Model I, this probability is even slightly smaller.

II CALCULATION OF SELECTION RATE

First, a method will be shown for calculating the probability of selecting a random file entry with a given number of *ones* in its composite descriptor as a result of a search composed of a given number of *ones*. Then this calculation is extended to the case in which the number of *ones* in the file entry and in the search are given as random variables with known probability distributions, rather than as fixed numbers.

Suppose that a search of the file is being made, with exactly j *ones* in the composite search descriptor. Then if a file entry with exactly i *ones* in its composite descriptor is chosen at random, with each of the $\binom{F}{i}$ possible patterns of the i *ones* equally likely, the probability that the file entry will be selected by the search is

$$R_{i,j}(F) = \begin{cases} \dfrac{\binom{i}{j}}{\binom{F}{j}} & \text{for } 0 \leqslant j \leqslant i \text{ and } N \leqslant i \leqslant F \\ 0 & \text{otherwise} \quad . \end{cases} \tag{1}$$

These values can be arrayed in an $F + 1$ by $F + 1$ selection probability matrix $R(F)$; as indicated, this matrix is a function only of the field size F.

Now if one descriptor is selected at random from the total vocabulary, it will have N *ones*, with probability one. If a second descriptor is selected at random from the total vocabulary and combined with the first to form a composite descriptor, the number of *ones* in the composite descriptor is not known with certainty. However, the probability distribution for the number of *ones* can be computed. Methods for computing this distribution under two different assumptions are shown in Secs. III and IV. This distribution can be arranged as a $F + 1$ by one column matrix $Q(2, F, N)$, with elements

$$Q_{i,2}(F, N) = Pr(i \textit{ ones} \text{ in composite of 2 descriptors, each with } N \textit{ ones} \text{ in a field of length } F). \tag{2}$$

Similarly, the probability distribution of the number of *ones* in the composite descriptor after K descriptors have been combined can be represented as an $F + 1$ by one column matrix $Q(K, F, N)$; as indicated, this distribution is a function only of K, F, and N.

If it is assumed that each file entry descriptor is formed by combining exactly M vocabulary descriptors, the number of *ones* in the composite descriptor for a file entry will have a probability distribution which can be represented as above by the column matrix $Q(M, F, N)$. If this matrix is pre-multiplied by the transpose $R^t(F)$ of $R(F)$, then the resulting $F + 1$ by one column matrix

$$S(M, F, N) = R^t(F)Q(M, F, N) \qquad (3)$$

will have as its elements

$$\begin{aligned} S_i(M, F, N) &= Pr(\text{selection of a randomly chosen file entry, given} \\ &\qquad \text{that the search descriptor contains exactly } i \text{ } \textit{ones}) \qquad (4) \\ &= \sum_{j=0}^{F} R_{i,j}(F)Q_{j,M}(F, N) \quad . \end{aligned}$$

In the design of information retrieval systems, these values are useful in estimating the expected rate of selecting unwanted file entries during a search on a quiz descriptor containing i *ones*. Methods of calculating an approximate value of $S_i(M, F, N)$ are given by Mooers[1] and Wise;[2] these approximations are based on the mean number of *ones* in a file entry rather than on the probability distribution of the number of *ones*. (In a later paper,[3] however, Mooers suggests calculating the exact selection rate by essentially the method followed here.)

A problem closely related to the above is that of estimating the expected rate of selecting unwanted file entries during a search on an arbitrarily chosen descriptor formed by combining L vocabulary descriptors. The theoretical analysis given here leads to a useful answer to this problem. If a quiz descriptor is formed by combining L descriptors, chosen at random from the vocabulary, the probability distribution of the number of ones in the quiz descriptor can be represented by the column matrix $Q(L, F, N)$. Then the probability of selecting an arbitrarily chosen file entry using this arbitrarily selected quiz is given by the single number

$$
\begin{aligned}
D(L,\ M,\ F,\ N) &= Q^{t}(L,\ F,\ N)S(M,\ F,\ N) \\
&= Q^{t}(L,\ F,\ N)R^{t}(F)Q(M,\ F,\ N) \\
&= \sum_{i=0}^{F}\ \sum_{j=0}^{F} Q_{i,L}(F,\ N)R_{i,j}(F)Q_{j,M}(F,\ N) \quad . \qquad (5)
\end{aligned}
$$

To understand the meaning of this number, it may help the reader to consider the following conceptual experiment. Suppose that a sample of size M is selected at random from the vocabulary, where all possible samples of size M are equally likely to be drawn. Then suppose that a second sample of size L is selected at random from the vocabulary, where all possible samples of size L are equally likely. If the composite descriptor for each sample is constructed by forming the logical sum of the individual descriptors for that sample, $D(L,\ M,\ F,\ N)$ is the probability that the composite descriptor for the sample of size M has a *one* in every position for which there is a *one* in the composite descriptor for the sample of size L. It is not specified at this point whether or not a sample may contain duplications of descriptors, *i.e.*, in the usual terminology, whether the sampling is done with replacement or without; this difference in concept distinguishes the approaches used in Secs. III and IV, respectively, to calculate the probability distributions $Q(K,\ F,\ N)$.

The above analysis can be extended to the case in which the file entries are not all coded with the same number of descriptors. If the maximum number of descriptors used is H, and if the probability distribution of the proportion of entries with each number of descriptors is given by the H by one column matrix $\mathfrak{m}$, where

$$
\mathfrak{m}_{k} = Pr(M = k) \quad \text{for} \quad k = 1,\ 2,\ \ldots\ H \quad , \qquad (6)
$$

then the probability of selecting an arbitrary file entry with a quiz composed of L descriptors is given by

$$
\begin{aligned}
D(L,\ \mathfrak{m},\ F,\ N) &= Q^{t}(L,\ F,\ N)R^{t}(F)Q(F,\ N)\mathfrak{m} \\
&= \sum_{i=0}^{F}\ \sum_{j=0}^{F}\ \sum_{k=1}^{H} Q_{i,L}(F,\ N)R_{i,j}(F)Q_{j,k}(F,\ N)\mathfrak{m}_{k} \quad . \qquad (7)
\end{aligned}
$$

In this expression, $Q(F, N)$ is the $F + 1$ by H matrix with columns $Q(1, F, N)$, $Q(2, F, N)$, ... $Q(H, F, N)$. Similarly, if a variable number of descriptors are combined in forming searches of the file, and if the probability distribution of the proportion of quizzes with each number of descriptors is given by the H by one column matrix $\mathfrak{L}$, then the probability of selecting an arbitrary file entry with an arbitrary quiz is

$$\begin{aligned} D(\mathfrak{L}, \mathfrak{m}, F, N) &= \mathfrak{L}^{t} Q^{t}(F, N) R^{t}(F) Q(F, N) \mathfrak{m} \\ &= \sum_{i=0}^{F} \sum_{j=0}^{F} \sum_{k=1}^{H} \sum_{l=1}^{H} \mathfrak{L}_{l} Q_{i,l}(F, N) R_{i,j}(F) Q_{j,k} \mathfrak{m}_{k} \quad . \qquad (8) \end{aligned}$$

III PROBABILITY DISTRIBUTION OF NUMBER OF ONES IN COMPOSITE DESCRIPTOR—MODEL I

In this section, the probability distribution $Q(K, F, N)$ of the number of *ones* in a composite descriptor formed by combining K descriptors, each with N *ones*, selected at random with replacement from the total vocabulary, is obtained. Each of the $V^K/K!$ possible samples are considered to be equally likely.

If the sequence $\{Y(K)\}$, $(K = 1, 2, \ldots)$, of random variables is considered, where $Y(K)$ represents the number of *ones* in the composite descriptor after K descriptors have been combined, then it is observed that the sequence $\{Y(K)\}$ forms a Markov chain with stationary transition probabilities (see Chapt. XV of Feller[5]). In other words, the random variable $Y(K + 1)$ given $Y(K)$ depends only on the value of $Y(K)$, and not on K or on the values of $Y(1), Y(2), \ldots, Y(K - 1)$. The one-step Markov transition probabilities for this process are given by the $F + 1$ by $F + 1$ matrix $P(F, N)$ with elements

$$P_{i,j}(F, N) = \frac{\binom{i}{N - j + i}\binom{F - i}{j - i}}{\binom{F}{N}} \qquad \begin{matrix} \text{for} & j = 0, 1, \ldots F \\ \text{and} & i = 0, 1, \ldots F \end{matrix} \tag{9}$$

$$= Pr(\text{addition of one descriptor increases the number of } \textit{ones} \text{ in composite from } i \text{ to } j),$$

where the usual extended factorial function is used to evaluate the binomial coefficients. It should be noted that the matrix $P(F, N)$ depends on F and N, but not on K. If the $F + 1$ by one matrix $Q(K, F, N)$ is identified with the probability distribution of $Y(K)$, then these probability distributions are obtained by successively forming the matrix products

$$\begin{aligned} P^t(F, N)\, Q(1, F, N) &= Q(2, F, N) \\ P^t(F, N)\, Q(2, F, N) &= Q(3, F, N) \\ &\vdots \\ P^t(F, N)\, Q(K, F, N) &= Q(K + 1, F, N) \quad , \\ &\vdots \end{aligned} \tag{10}$$

where the initial distribution $Q(1, F, N)$ is

$$Q_{i,1}(F, N) = \begin{cases} 1 & \text{for } i = N \\ 0 & \text{otherwise.} \end{cases} \qquad (11)$$

The mean and variance of $Y(K)$ can be calculated from the distribution $Q(K, F, N)$, once it is obtained. However, it is also possible to obtain them by a different line of reasoning, without computing the explicit distribution of $Y(K)$. (see Chapt. IX of Feller[5]). If

$$X_i(K) = \begin{cases} 1 & \text{if the } i\text{th position of composite descriptor has a } \textit{one} \\ 0 & \text{otherwise} \end{cases} \qquad (12)$$

for

$$i = 1, 2, \ldots F,$$

then

$$Y(K) = \sum_{i=1}^{F} X_i(K) \quad .$$

Also,

$$\begin{aligned} Pr[X_i(K) = 1] &= 1 - Pr[X_i(K) = 0] \\ &= 1 - Pr(\textit{zero} \text{ in } i\text{th position for all } K \text{ descriptors}) \\ &= 1 - \left(\frac{F - N}{F}\right)^K \quad \text{for } i = 1, 2, \ldots F \quad . \qquad (13) \end{aligned}$$

Thus the mean of $Y(K)$ is

$$\begin{aligned} E[Y(K)] &= \sum_{i=1}^{F} E[X_i(K)] \\ &= F\left[1 - \left(\frac{F - N}{F}\right)^K\right] \quad . \qquad (14) \end{aligned}$$

Wise[2] substitutes this mean into Eq. (1) (as i) to obtain an approximation for Eq. (3). He appears to first round the value of the mean to the nearest integer. However, it should be noted that there is no need to round, since the function on the right side of Eq. (1) can be extended in the usual way to non-integer values of i, using the extended factorial or gamma function. It is likely that use of the unrounded mean would reduce the approximation error in most cases.

Continuing as above, the variance of $Y(K)$ is

$$\begin{aligned} Var[Y(K)] &= E\{Y(K) - E[Y(K)]\}^2 \\ &= E\left(\left\{\sum_{i=1}^{F} X_i(K) - E[Y(K)]\right\}\left\{\sum_{j=1}^{F} X_j(K) - E[Y(K)]\right\}\right) \\ &= \sum_{i=1}^{F} E[X_i^2(K)] + \sum_{i \neq j} E[X_i(K)X_j(K)] - \{E[Y(K)]\}^2 \ . \qquad (15) \end{aligned}$$

The first term on the right side of this equation is evaluated as

$$\sum_{i=1}^{F} E[X_i^2(K)] = \sum_{i=1}^{F} P_r[X_i(K) = 1] = E[Y(K)] \ . \qquad (16)$$

The second term is

$$\begin{aligned} \sum_{i=j} E[X_i(K)X_j(K)] &= \sum_{i \neq j} Pr(X_i = 1, X_j = 1) \\ &= \sum_{i \neq j} Pr(X_i = 1) Pr(X_j = 1 | X_i = 1) \\ &= \sum_{i \neq j} \sum_{k=1}^{K} \binom{K}{k} \left(\frac{N}{F}\right)^k \left(\frac{F-N}{F}\right)^{K-k} \left[1 - \left(\frac{F-N}{F-1}\right)^k \left(\frac{F-1-N}{F-1}\right)^{K-k}\right] \\ &= F(F-1)\left\{1 - \left(\frac{F-N}{F}\right)^K - \left(\frac{F-N}{F}\right)^K \left[1 - \left(\frac{F-1-N}{F-1}\right)^K\right]\right\} \ . \end{aligned}$$

(17)

Thus, substituting Eqs. (14), (16), and (17) into Eq. (15), and simplifying, the result

$$
\begin{aligned}
Var[Y(K)] &= F\left(\frac{F - N}{F}\right)^{K}\left[1 - F\left(\frac{F - N}{F}\right)^{K} + (F - 1)\left(\frac{F - 1 - N}{F - 1}\right)^{K}\right] \\
&= F\left(\frac{F - N}{F}\right)^{K}\sum_{k=2}^{K}\binom{K}{k}N^{k}\left[\frac{1}{(F - 1)^{k-1}} - \frac{1}{F^{k-1}}\right] \qquad (18)
\end{aligned}
$$

is obtained. The second line form of Eq. (18) is sometimes easier to evaluate than the first.

IV PROBABILITY DISTRIBUTION OF THE NUMBER OF *ONES* IN COMPOSITE DESCRIPTOR—MODEL II

In the previous section, it was assumed that composite descriptors were formed by selecting descriptors at random from the total vocabulary, sampling with replacement. In other words, the possibility that not all descriptors selected to form a composite were different was admitted. Here, the assumption is made that all of the descriptors selected are different.

Orosz and Takács[4] consider this model for the more general case of an arbitrary number of subfields. For the present case of a single field, they obtain the probabilities

$$Q^*_{i,k}(F,N) = \binom{F}{i} \sum_{j=F-i}^{F} (-1)^{j-F+i} \binom{i}{F-j} \frac{\binom{V(F-j,N)}{K}}{\binom{V(F,N)}{k}} \qquad (19)$$

corresponding to the $Q_{i,k}(F,N)$ of Model I above, where

$$V(F-j,N) = \binom{F-j}{N} \qquad \text{for } i = 0,\ 1,\ \ldots\ F-N \qquad (20)$$

is the vocabulary size if i specified positions are *zero* in each descriptor. They show that the mean of this distribution is

$$E[Y^*(K)] = F\left\{1 - \frac{\binom{V(F-1,N)}{K}}{\binom{V(F,N)}{K}}\right\}, \qquad (21)$$

and the variance is

$$\text{Var}\ [Y^*(k)] = F(F-1)\frac{\binom{V(F-2,N)}{K}}{\binom{V(F,N)}{K}} + F^2 \frac{\binom{V(F-1,N)}{K}}{\binom{V(F-1,N)}{K}} \left\{1 - \frac{\binom{V(F-1,N)}{K}}{\binom{V(F,N)}{K}}\right\}. \qquad (22)$$

From the point of view of describing the underlying information coding process, Model II is probably preferable to Model I. Duplications in descriptor assignment in Model I result in a small downward shift in the probability distribution of the number of *ones* in the composite descriptor, as compared with Model II, thus increasing the values in the selection matrix $S(M, F, N)$ given by Eq. (3). However, this change will be very slight in the parameter range of usual interest. The probability that a file entry composed of M descriptors, selected at random with replacement, contains one or more duplications of descriptors is

$$P_r \text{ (duplication)} = 1 - \frac{\binom{V(F,N)}{M}}{\frac{V^M(F,N)}{M!}} \quad . \tag{23}$$

For example, for $F = 40$, $N = 4$, and $M = 6$,

$$V(40, 4) = \binom{40}{4} = 91,390$$

and

$$P_r \text{ (duplication)} \approx \frac{M(M-1)}{2V(F,N)} = 0.000164 \quad .$$

This probability is negligible.

From the point of view of ease of computation, Model I appears to be at an advantage with respect to Model II. Thus Model I is used here to obtain the probability distributions for use in computations of random selection probabilities.

V EXAMPLE OF COMPUTATION OF RANDOM SELECTION PROBABILITIES

Here, the calculation of random selection probabilities is shown in detail, under the assumptions of Model I. The case $F = 10$, $N = 2$, and $M = 4$ will be considered; small numbers are chosen so that the various arrays can be shown in full detail.

For this case, the one-step Markov transition probability matrix $P(10, 2)$ is given by the entries $P_{i,j}(10, 2)$ in Table I. Then the probability distributions $Q(K, 10, 2)$ for the number of *ones* in the composite of K descriptors, calculated according to Eq. (10), are given by the entries $Q_{i,k}(10, 2)$ in Table II. The matrix $R(10)$ of selection probabilities, calculated from Eq. (1) is given by the entries $R_{i,j}(F)$ in Table III, where i represents the number of *ones* in the composite file descriptor and j represents the number of *ones* in the composite quiz descriptor. The symmetry of this matrix about the 45 degree angle should be noted.

Forming the random selection probabilities $S(4, 10, 2)$ according to Eq. (3), one obtains the values $S_i(4, 10, 2)$ shown in Table IV. The entries $S_i(4, 10, 2)$ give the probability of selecting a file entry by chance with a quiz containing i *ones*. Values computed by two approximation methods are also listed in Table IV for comparison. Then calculating $D(L, 4, 10, 2)$ according to Eq. (7), one obtains the probability of selecting a randomly chosen file entry with a quiz composed of L descriptors; these probabilities are listed in Table V.

These calculations have been programmed in ALGOL, and tables run on the Burroughs 220 computer for a number of cases of interest. The results indicate in general that the Wise approximation underestimates and the Mooers upper bound overestimates the random selection probabilities. As an example, for a field of length 40, 2 *ones* per descriptor, file entries each composed of 10 descriptors, and a quiz with 12 *ones*, the actual selection probability is 1.15×10^{-4}, the Wise approximation is 3.48×10^{-5}, and the Mooers upper bound is 1.74×10^{-3}. If one wishes to use the present model as a basis for system design, it would appear desirable to calculate exact probabilities.

TABLE I

ONE-STEP MARKOV TRANSITION PROBABILITIES $P_{i,j}(10, 2)$

j =	0	1	2	3	4	5	6	7	8	9	10
i = 0			1.000								
1			0.200	0.800							
2			0.022	0.356	0.622						
3				0.067	0.467	0.467					
4					0.133	0.533	0.333				
5						0.222	0.556	0.222			
6							0.333	0.533	0.133		
7								0.467	0.467	0.067	
8									0.622	0.356	0.022
9										0.800	0.200
10											1.000

TABLE II

PROBABILITY DISTRIBUTIONS $Q(K, 10, 2)$ OF NUMBER OF *ONES*

K =	1	2	3	4
i = 0				
1				
2	1.000	0.022	0.000	0.000
3		0.356	0.032	0.002
4		0.622	0.263	0.050
5			0.498	0.265
6			0.207	0.433
7				0.221
8				0.028
9				
10				
Expected Value	2.000	3.600	4.880	5.904
Variance	0.00	0.28	0.59	0.81

TABLE III

SELECTION PROBABILITIES $R_{i,j}(10)$

j = 0	0	1	2	3	4	5	6	7	8	9	10
i = 0	1.000										
1	1.000	0.100									
2	1.000	0.200	0.022								
3	1.000	0.300	0.067	0.008							
4	1.000	0.400	0.133	0.033	0.005						
5	1.000	0.500	0.222	0.083	0.024	0.004					
6	1.000	0.600	0.333	0.167	0.071	0.024	0.005				
7	1.000	0.700	0.467	0.292	0.167	0.083	0.033	0.008			
8	1.000	0.800	0.622	0.467	0.333	0.222	0.133	0.067	0.022		
9	1.000	0.900	0.800	0. 700	0.600	0.500	0.400	0.300	0.200	0.100	
10	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

TABLE IV

RANDOM SELECTION PROBABILITIES *vs* NUMBER OF *ONES* IN QUIZ

	$S_i(4, 10, 2)$	WISE APPROX.*	MOOERS APPROX.†
$i = 0$			
1			
2	0.331	0.322	0.349
3	0.173	0.157	0.206
4	0.084	0.065	0.122
5	0.036	0.021	0.072
6	0.013	0.004	0.042
7	0.004	- 0.001	0.025
8	0.001	0.000	0.015
9			
10			

* Using expected value of 5.904 *ones* per file descriptor,

$$\text{Approx. } S_i(4,\ 10,\ 2) = \frac{\binom{5.904}{i}}{\binom{10}{i}} .$$

† Using expected value of 5.904 *ones* per file descriptor,

$$\text{Approx. } S_i(4,\ 10,\ 2) = \left(\frac{5.904}{10}\right)^i ;$$

Mooers gives this formula as an upper bound for $S_i(M, F, N)$.

TABLE V

RANDOM SELECTION PROBABILITIES
vs NUMBER OF DESCRIPTORS IN QUIZ

	$D(L, 4, 10, 2)$
$L = 1$	0.331
2	0.121
3	0.048
4	0.021

VI OPTIMUM SYSTEM DESIGN

If Model I is adopted as a description of the physical system, the random selection probabilities calculated as above may be used as a basis for optimum design of a document coding system. The approach taken will depend on which parameters are assumed fixed and which variable, and on the costs associated with varying parameter values.

To take a single example, suppose that the field length F is fixed, and the probability distribution $\mathfrak{M}$ of the numbers of descriptors used to code file entries is known. Then if the cost of varying the number N of *ones* in a descriptor is neglected, the optimum value of N for a quiz of a given number of descriptors is found by determining that N which minimizes the random selection probability $D\ (L,\ \mathfrak{M},\ F,\ N)$ (subject, of course, to the practical restriction that the resulting available vocabulary size $V(F,N)$ be large enough to meet the requirements of the system). If the minimum random selection probability found for a given L is too large, then one must conclude, if no other parameters are to be changed, that a larger number of descriptors must be combined to perform a quiz. If, on the other hand, a probability distribution $\mathfrak{L}$ of the number of descriptors combined to perform quizzes of the file is given, then the optimum value of N is that which minimizes $D\ (\mathfrak{L},\ \mathfrak{M},\ F,\ N)$.

To consider another example, suppose that the distribution $\mathfrak{M}$ of the number descriptors combined to form file entries is given, and that the system is required to perform searches on a minimum number L of descriptors in a quiz, with a random selection probability not exceeding E. If any desired field length F may be used at an increasing cost $C_1(F)$ and any desired N may be used at an increasing cost $C_2(N)$, then the optimum values of F and N will be those which minimize $C_1(F) + C_2(N)$, subject to the restriction $D\ (L,\ \mathfrak{M},\ F,\ N) \leq E$.

In a similar manner, other optimization problems may be formulated, as appropriate to the particular design conditions encountered.

VII IMPROVED MODELS

Models I and II fail to take into consideration the fact that, in the usual document coding system, only a small portion of the potential vocabulary $V(F,N)$ of descriptors is actually used in constructing file entries, and consequently in constructing quizzes of the file. As a step toward a more realistic mathematical model, one might assume that a restricted vocabulary of a specified size is selected at random from the potential vocabulary, sampling without replacement. Then a random file would be constructed by selecting groups of descriptors at random, with equal probability, from this restricted vocabulary, sampling either with or without replacement. A quiz would be constructed from the restricted vocabulary in the same manner. It is conjectured that an analysis of this model would indicate higher random selection rates than obtained with Models I and II.

An additional refinement of the mathematical model would be to select file entry descriptors and quiz descriptors from the restricted vocabulary according to a probability distribution approximating the frequency of usage of descriptors in an actual file. It is conjectured that this refinement would further increase the calculated random selection rates.

REFERENCES

1. Calvin N. Mooers, "Zatocoding for Punched Cards," Zator Technical Bulletin No. 30, Zator Company, Boston (1950).

2. Carl S. Wise, "Mathematical Analysis of Coding Systems," Chapt. 21 of *Punched Cards, Their Applications to Science and Industry*, 2nd edition, edited by Robert S. Casey, James W. Perry, Madeline M. Berry, and Allen Kent (Reinhold Publishing Corporation, New York City, 1958).

3. Calvin N. Mooers, "The Exact Distribution of the Number of Positions Marked in a Zatocoding Field," Zator Technical Bulletin No. 73, Zator Company, Boston (1952).

4. G. Orosz and L. Takács, "Some Probability Problems Concerning the Marking of Codes into the Superimposition Field," *Journal of Documentation*, vol. 12, No. 4, pp. 231-34, (December 1956).

5. William Feller, *An Introduction to Probability Theory and Its Applications*, vol. 1, 2nd edition, (John Wiley and Sons, New York City, 1957).

STANFORD
RESEARCH
INSTITUTE

MENLO PARK, CALIFORNIA

REGIONAL OFFICES AND LABORATORIES

SOUTHERN CALIFORNIA LABORATORIES
820 Mission Street
South Pasadena, California

WASHINGTON OFFICE
711 14th Street, N. W.
Washington 5, D. C.

NEW YORK OFFICE
270 Park Avenue
New York 17, N. Y.

EUROPEAN OFFICE
Pelikanstrasse 37
Zurich 1, Switzerland

RESEARCH REPRESENTATIVES

HONOLULU, HAWAII
Finance Factors Building
195 South King Street
Honolulu, Hawaii

PORTLAND, OREGON
Suite 914, Equitable Building
421 Southwest 6th Avenue
Portland, Oregon

PHOENIX, ARIZONA
Suite 216, Central Plaza
3424 North Central Avenue
Phoenix, Arizona

SRI - MIRF

STANFORD RESEARCH INSTITUTE

MENLO PARK, CALIFORNIA

SRI

June 1960

Supplement A to Quarterly Report 2

SRI Project EU-3101

THE ORGANIZATION OF A MEMORY SYSTEM FOR INFORMATION RETRIEVAL APPLICATIONS

By: Charles P. Bourne

Prepared for:

ROME AIR DEVELOPMENT CENTER
Air Research and Development Command
United States Air Force
Griffiss Air Force Base
New York

Contract AF 30(602)-2142

Approved:

J. Reid Anderson

J. Reid Anderson
Manager, Computer Techniques Laboratory

Jerre D. Noe
Assistant Director of Engineering Research

Copy No. [illegible]

SRI – MIRF

copy to Trudi

chpt 3

DESIGN OF AN EXPERIMENTAL MULTIPLE INSTANTANEOUS RESPONSE FILE* AD

E. L. Younker, C. H. Heckler, Jr., D. P. Masher, and J. M. Yarborough
Stanford Research Institute
Menlo Park, California

SUMMARY

An experimental model of an electronic reference retrieval file in which all file entries are interrogated simultaneously has been designed and constructed. The experimental model is designed to store and search on a file of indexes to 5,000 documents. A document index consists of a decimal accession number and up to eight English word descriptors that are closely related to the contents of the document. The vocabulary required to describe the documents is held in a machine dictionary that has a design capacity of 3,000 words. In the model delivered to the sponsor, Rome Air Development Center, the storage capacity is only partially used. The specification for the delivered model calls for the storage of approximately 1,100 documents that were selected from the ASTIA (now DDC) Technical Abstract Bulletin and of the vocabulary needed to describe them (about 1,000 words). The document indexes and the dictionary words are stored in wiring patterns associated with arrays of linear ferrite magnetic cores.

A search question, consisting of one to eight descriptors in their natural English form, is entered by means of an electric typewriter. During entry of the search question, the dictionary magnetic store is interrogated by the alphabetic code of each search word. If a word is not contained in the dictionary, it is automatically rejected. After all words of the search question have been entered, the document magnetic store is interrogated by the search question in superimposed code form. The comparison between the search word and the document indexes is made for all documents simultaneously and the machine instantaneously determines if any documents in the file include the search question. If there are none, the machine indicates visually that there is no response. If there is at least one, the machine counts the number of responding documents and displays this number. Then it types out the indexes of all responding documents on the same typewriter that was used to ask the question.

INTRODUCTION

Memories that can be searched in parallel and from which stored information is retrieved on the basis of content have received considerable attention for application to retrieval file problems.[1, 2, 3, 4] This paper describes the design of an experimental retrieval file based on the work reported by Goldberg and Green.[3] Since the contents of the semipermanent magnetic memory used in the experimental file can be searched in parallel and multiple responses

* The work described in this paper was supported by Rome Air Development Center under Contract AF 30(602)-2772.

to the search question are permitted, the system is called MIRF—Multiple Instantaneous Response File.[5]

LOGICAL ORGANIZATION OF THE MIRF SYSTEM

The logical organization of the experimental MIRF system is illustrated by Fig. 1. Information pertaining to the document indexes and to the descriptors used in the document indexes is contained in two major units called MIRF units. A MIRF unit is basically a magnetic memory in which information is permanently stored in the wiring associated with the magnetic cores. The Document MIRF is the principal element of the system. It contains for each stored document index the document accession number and the descriptors (in coded form) that describe that document, as well as a superimposed search code that is used in the searching process. The Dictionary MIRF has two functions. During the input phase of operation it translates the alphabetic code of the English word descriptor that is entered from the typewriter into the binary serial number assigned to that English word for use inside the machine. During the output phase, the Dictionary MIRF translates the binary serial number of a word that is obtained during a search into the alphabetically coded form of that word.

After the binary serial number of an input English word has been generated, this binary number is translated by a logical process in the Search Code Generator into a search code that is assigned to the particular English word. The search codes of successive words of a search question are superimposed by adding them together, bit by bit by an inclusive-OR operation. When the search question is complete, the superimposed search code of the question is compared with the superimposed code section of the Document MIRF. Each document index whose search field includes the superimposed code of the search question is said to *respond* to the question. Frequently more than one document will respond. By a logical process for resolving multiple responses,[6] the accession number of a particular responding document is generated. Then the binary serial numbers of the English words contained in this document index are generated one at a time. By means of the Descriptor Selector, each serial number is transmitted to the Dictionary MIRF, where it is translated to the alphabetic code of the English word. This process is repeated for each responding document.

SYSTEM DESIGN

1. Magnetic Implementation of the MIRF Unit

The MIRF units of the experimental model use an interesting modification of the Dimond Ring[7] translator in which the drive and sensing functions are interchanged. Information is stored in unique wiring patterns associated with an array of linear ferrite cores as il-

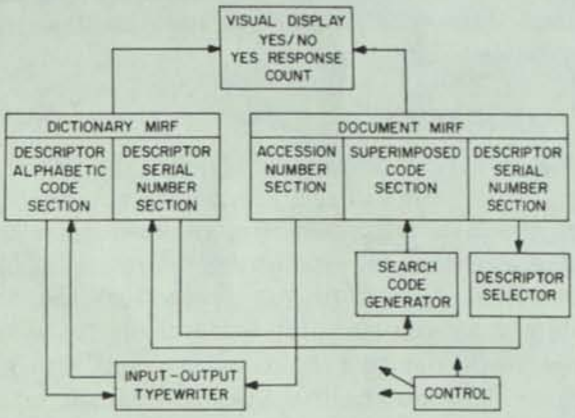


Figure 1. Simplified Block Diagram of MIRF Experimental Model.

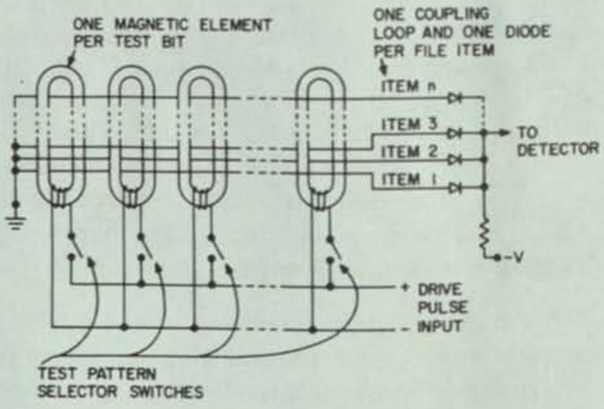

ONE MAGNETIC ELEMENT PER TEST BIT
ONE COUPLING LOOP AND ONE DIODE PER FILE ITEM
ITEM n
ITEM 3
ITEM 2
ITEM 1
TO DETECTOR
-V
+ DRIVE PULSE - INPUT
TEST PATTERN SELECTOR SWITCHES

Figure 2. Core-Wiring Arrangement for MIRF Memory.

lustrated by Fig. 2. Each item of stored information (a document index in the Document MIRF or a descriptor in the Dictionary MIRF) is represented by a conductor that passes through or around each associated core in a unique pattern determined by the information it contains. In series with each conductor is a diode. The cathodes of many diodes are connected together to form the input to a detector amplifier. Notice that one core is required for each bit of information, but that each core can be associated with a particular bit of many item conductors.

Each core has an input winding that can be selected by means of a switch. All cores whose selector switch is closed will be energized when a drive pulse is applied. A voltage will be induced in each item conductor that threads an energized core, but no voltage will be induced in conductors that do not thread the core. A test can be made on the information stored in many cores by selecting a particular set of cores and energizing them. In order for an item to match the test information, its conductor must pass outside of every energized core. Then no voltage will be generated in the item wire and the input to the detector amplifier will be held near ground through the item diode. Voltages will be induced in the conductors of items that do not match the test; the polarity of these voltages is chosen to back-bias the associated diodes. If no item matches the test information, a voltage will be induced in every item conductor and every diode will be back-biased. The input to the detector will then assume a significantly negative voltage. Thus, the presence or absence of desired stored information can be determined by applying the drive currents to a particular set of cores. This is a function of an associative or content-addressed memory: to indicate the presence or absence of certain information based on the detailed contents of a search question without regard to the actual location (or address) of that information.

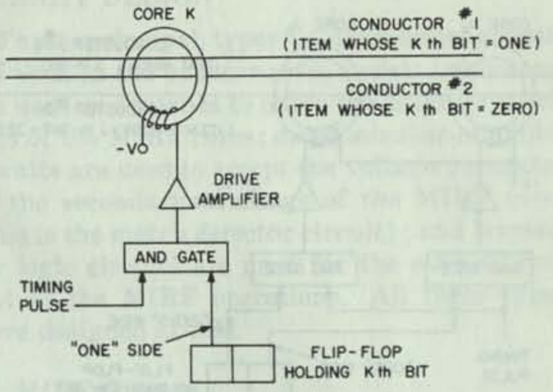


Figure 3. Circuit for Testing Inclusion.

Now consider in more detail how a bit of information of a search question is compared with information in a MIRF unit. Figure 3 illustrates how a test is made to determine whether or not the test bit is logically "included" in the stored information. This circuit is typical of those used in the superimposed section of the Document MIRF. One core is used to store the *k*th bit of many items. The *k*th bit of the search question is stored in a flip-flop whose *one* side is connected by way of an AND gate to a drive amplifier, which in turn is connected to the primary winding of the *k*th core. The conductor of an item whose *k*th bit is equal to *one* (Conductor 1) passes outside the *k*th core. On the other hand, the conductor of an item whose *k*th bit is equal to *zero* (Conductor 2) threads the core. If the flip-flop stores a *one*, the primary winding of the core will be energized when the timing pulse is applied to the AND gate. A voltage will be induced in Conductor 2 (indicating a mismatch) but none will be induced in Conductor 1 (indicating a match). If the flip-flop stores a *zero*, the primary winding will not be energized because the timing pulse will be blocked at the AND gate. No voltage will be induced in either conductor, and a match will be indicated on both lines. Therefore, it can be seen that a stored *one* bit includes both a test *one* and a test *zero*, while a stored *zero* bit includes only a test *zero*.

The circuit for testing for identity between the test bit and the information stored in the MIRF is shown in Fig. 4. This circuit is typical of those used in the alphabetic descriptor portion of the Dictionary MIRF. The *j*th bit of many items is stored in a pair of cores j_A and j_B. The *j*th bit of the test question is stored in a flip-flop. In this case, both the *one* and *zero* sides of the flip-flop are connected to AND gates

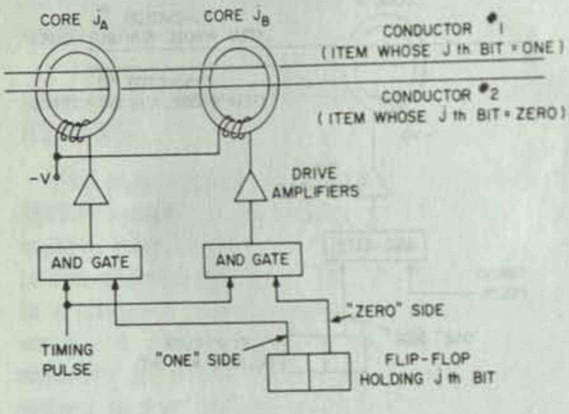


Figure 4. Circuit for Testing Identity.

whose outputs control drive amplifiers that are connected to the primary windings of the cores j_A and j_B. The conductor of an item whose *j*th bit is *one* (Conductor 1) bypasses core j_A while the conductor of an item whose *j*th bit is a *zero* threads core j_A. The threading of core j_B by the two conductors is the reverse of the wiring of core j_A. If the flip-flop stores a *one*, the primary winding of core j_A will be energized when the timing pulse occurs. No voltage will be induced in Conductor 1 (a match indication) but a voltage will be induced in Conductor 2 (a mismatch indication). If the flip-flop stores a *zero*, the primary winding of core J_B will be energized. In this case, a voltage will be induced in Conductor 1 but not in Conductor 2. Thus it can be seen that the bit stored in the MIRF must match the test bit identically for a match indication to be obtained.

2. Basic Operations Using the MIRF Units

Two types of operations involving the MIRF units are basic to the operation of this experimental model. One operation tests to see if certain information is contained in the MIRF. The other uses information that is contained in the MIRF to generate a number in a flip-flop register external to the MIRF unit. Examples of these basic operations are given in the following paragraphs.

a. *Testing of Information Contained in the MIRF Unit*

Dictionary MIRF—During the input of the English words to form a search question, the Dictionary MIRF is tested to see if the input word is contained in the vocabulary (that is, if it is a valid descriptor). This is done by gating the alphabetic descriptor register to the drive amplifiers associated with the alphabetic portion of the MIRF (50 bits long, two cores per bit). As a result, 50 drive amplifiers are energized and 50 primary windings in the MIRF carry current. If one of the stored words has a bit pattern in the alphabetic portion that matches identically the energized set of primaries, the match detector will indicate a match condition. If not, the match detector will indicate a mismatch condition. The output of the match detector is used to determine the next step in the logical sequence. It is important to note that the test is applied to the entire Dictionary MIRF simultaneously and that a match or mismatch signal for the entire MIRF is obtained in about 5 microseconds.

Document MIRF—After all words of the search question have been typed, the *superposition* of their search codes is held in the search code accumulator. At the beginning of the actual search operation, the flip-flops of the search code accumulator are gated to their associated drive amplifiers. A particular set of drive amplifiers is energized and current flows in a corresponding set of primary windings in the 80 bit superimposed code field of the Document MIRF. If the detailed bit pattern represented by the energized primaries is included in any of the superimposed fields of the stored document indexes, a match condition is indicated by the match detector. If not, a mismatch indication is given. The test is made on the entire contents of the document MIRF simultaneously and a YES/NO response is obtained in about 5 microseconds.

It should be pointed out that the criterion for a match is inclusion, not identity. A document index includes the search question if the following conditions of the superimposed search code portion of the index are satisfied. First, for every bit of the index search field that is a *one*, the corresponding bit of the search question is either a *zero* or *one*. Second, for every bit of the index search field that is a *zero* the corresponding bit of the search question is a *zero* (in other words a binary *one* includes both a *one* and a *zero,* but a binary *zero* does not include a binary *one*).

b. *Generating Numbers by the MIRF Process*—The generation of the serial number of an input descriptor illustrates this operation. Assume that an English word has been typed in and that the test for valid descriptor is true. Because a match is obtained when the alphabetic descriptor register is gated to the Dictionary MIRF, one item wire in the MIRF is effectively isolated: namely, the wire that is uniquely related to the input descriptor. The detailed wiring pattern of this wire in a group of cores outside the alphabetic code field contains the binary serial number of the input descriptor. By gating the alphabetic descriptor register to the MIRF and at the same time causing current to flow in the primary winding of a core that is in the serial number portion of the MIRF, the binary value associated with that core for the selected line can be determined. The presence of current in the additional winding tests for a binary *one* in that position. If the match detector indicates a match, the value is indeed *one.* However, if a mismatch is obtained, the value must be *zero.*

The sequence for generating the serial number is as follows: First the flip-flop register that will eventually hold the serial number is cleared to all *ones.* Then the alphabetic descriptor register is gated to its drive amplifiers and a drive amplifier associated with the parity bit of the serial number is energized. The output of the match deteetor is observed. If a match condition is observed, it is known that the parity bit is actually a *one* and the parity bit flip-flop in the serial number register is not changed. If a mismatch is observed, it is known that the parity bit is *zero* and the parity bit flip-flop in the serial number register is not to *zero.* The next step is to energize the drivers associated with the alphabetic descriptor register and a driver associated with the least significant bit of the serial number. Again the output of the match detector is observed and the flip-flop assigned to the least significant bit is either allowed to stay at *one* or is changed to a *zero.* This procedure continues for thirteen steps. At the end of this time, the 12-bit serial number and its parity bit will have been generated and stored in the serial number register.

CIRCUIT DESIGN

Three principal types of transistor circuits are used in the experimental model: transistors are used as switches to drive the primary windings of the MIRF cores; discriminator-amplifier circuits are used to accept the voltage generated on the secondary windings of the MIRF cores (this is the match detector circuit); and transistor logic circuits are used for the over-all control of the MIRF operations. All three types were designed at SRI.

1. MIRF Driver

The drive currents that are required by the ferrite cores in the Document and Dictionary MIRFs are furnished by circuits such as the one shown schematically in Fig. 5. Four MIRF driver circuits are mounted on one printed circuit plug-in board, as shown in Fig. 6. Each circuit is capable of supplying the required 2 amperes at low impedance. The power transistor that delivers the drive current (Type 2N1905) is driven by a push-pull emitter follower that provides 60 milliamperes of base drive current into 2N1905. The output power transistor has rise-and-fall time capabilities of less than 0.3 microsecond. The actual current in the load is nearly linear because of the inductive nature of the load and builds up to the 2 ampere amplitude at the end of approximately 10 microseconds. The overshoot voltage induced when the transistor is turned off is clamped by a silicon diode to —36 volts. The clamp prevents excessive voltage spikes from appearing across the output transistor while still allowing the load inductance to recover within 10 microseconds.

Two protective features of the MIRF driver circuit should be noted. One is a fuse, which is inserted in series with the load to protect against excessive load currents. Before the winding of the magnetic circuits internal to the MIRF assembly can be damaged by too much current from, say, an accidental short circuit, the fuse wire will open up. The second protective circuit includes a square-loop memory core that is threaded by the lead going to the transistor load. This core is normally biased off, but if the drive current exceeds a safe value the square-loop core will switch and induce a voltage in a sense lead. The voltage in

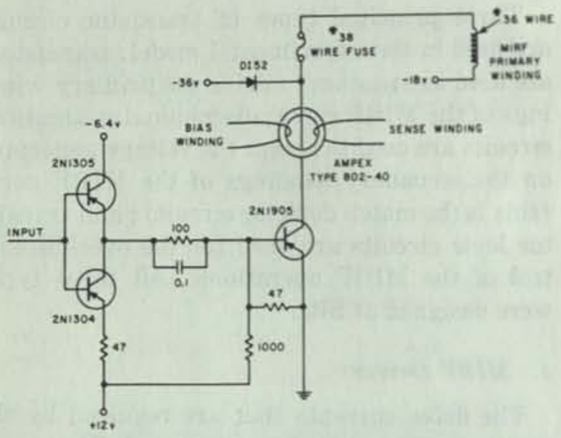


Figure 5. Schematic of MIRF Driver.

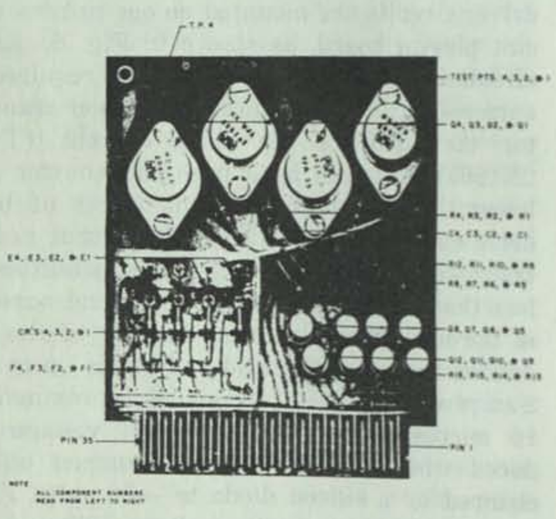


Figure 6. Component Assembly of MIRF Driver Board.

the sense lead is amplified and used to turn off the system clock. The purpose of this circuit is to protect the 2N1905 transistor against excessive heat dissipation from currents that are excessive but not large enough to burn out the fuse wire.

2. MIRF Discriminating Amplifier

The electrical output of the MIRF magnetic modules is generated by a very large diode gate including almost 300 diodes. Under the worst conditions a match signal from this array can reach a level as high as 0.4 volt. On the other hand, a mismatch signal from the same array may only generate a potential of 0.6 volt. It is necessary for the MIRF discriminating amplifier to differentiate between these two signals and generate a standard logic level output of —6 volts for a mismatch and 0 volts for a match. The circuit for the amplifier is shown in Fig. 7. In order to distinguish between very closely spaced match and mismatch signals, two thresholds are employed in the amplifier. The first threshold is provided by a 1N3605 silicon diode at the input to the amplifier. This diode does not pass signals unless they exceed approximately 0.5 volt. After passing the first threshold, the signal is amplified in a feedback amplifier with a gain of about 50. If the amplified signal then exceeds the second threshold of 3 volts, a mismatch signal is delivered at the output of the amplifier.

3. Logic Circuits

In the flip-flop register and over-all control circuits, resistor-transistor logic is used. Highly reliable circuits that operate in the 100-kc frequency range have been developed. The basic gate circuit is shown in Fig. 8. This circuit in typical use performs a simple majority operation. If one or more of its three inputs are at a negative potential, the output is held at ground potential. Since ground is defined as the *one* state in this system, and a —6 volt potential is defined as a *zero* state, the basic gate performs the "not and" or NAND operation.

All the passive components shown in Fig. 8, plus one resistor and two capacitors, are contained in one physical element supplied by Centralab, Inc. These components are screened on a passive substrate to a tolerance of 3% for the resistors (5% design tolerance) and

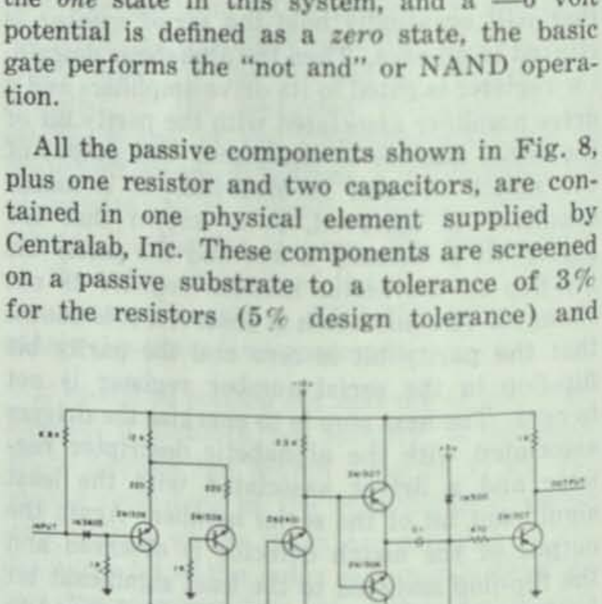


Figure 7. Schematic of Discriminating Amplifier.

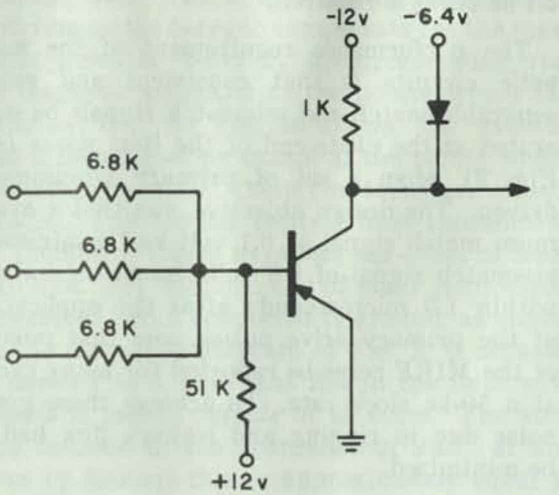


Figure 8. RTL Circuit Designed for the MIRF System.

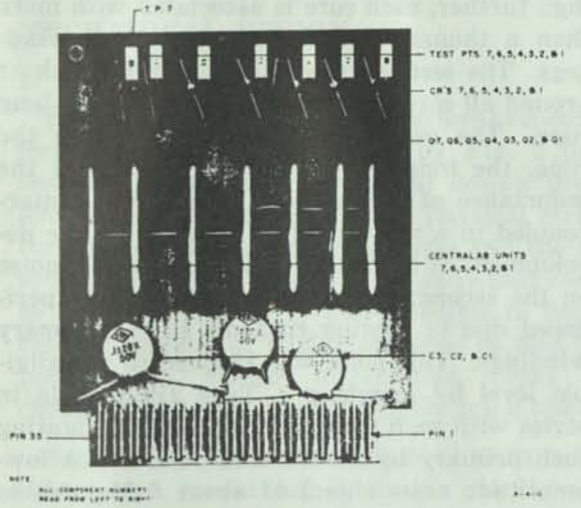


Figure 9. Component Assembly of Gate-Logic Board.

10% for the capacitors. The substrates are encapsulated with a Durez coating, and are ready for mounting to a printed circuit card via their projecting leads.

The gate circuit is a basic part of every logic circuit employed in the machine. By itself it performs the combinatorial function of logical conditions. Two gate circuits properly interconnected form a bistable, or flip-flop, circuit. Two gate circuits interconnected in a slightly different way form a monostable, or one-shot, circuit. The gate circuit is also used as a preamplifier for an emitter-follower circuit. The basic logic circuits, e.g., gates, one shots, flip-flops, etc., are mounted on plug-in logic boards. A typical logic board, with seven gate circuits mounted on a printed circuit board, is shown in Fig. 9.

MAGNETIC DESIGN

1. General Considerations

The magnetic design of a MIRF unit is centered in the individual magnetic core, which acts as a transformer with a multiturn primary winding and many single-turn secondary windings. When current flows in the primary winding, the magnetic core must be capable of producing a flux change of sufficient time duration and amplitude to generate the desired signal in secondary windings. The amplitude of the induced voltage is determined primarily by the characteristics of the diode associated with the secondary winding. The duration of the induced voltage is determined primarily by noise on the secondary winding and the consequent delay required before sampling of the output can be accomplished.

The cross-sectional area of the magnetic core is proportional.to the product of the amplitude and duration of the voltage induced in the secondary windings (this is usually referred to as the *volt-second area* of the induced voltage pulse). This was kept reasonably small by using a high-quality germanium diode (the 1N500) which requires a back-biasing voltage of only 0.6 volt in order to perform properly in the diode circuit associated with the input to the discriminating amplifier. The circumferential length of the magnetic core is determined primarily by the number of secondary windings associated with the core and the mechanical design of the supports for these windings. In the MIRF units of the experimental equipment, the core has the capacity for 2,000 secondary windings. The core's mean circumferential length is 7 inches; its cross section is a square, ¼ inch on a side.

Two other considerations influenced the selection of the magnetic cores used in the MIRF units. One is the requirement that the core be made in two pieces so that the array of cores can be separated into two portions to facilitate initial wiring and changes in wiring. The other is the necessity of using commercially available parts. The number of cores needed in this experimental equipment is too small to justify the design and production of a core of special size or shape.

2. Details of the Dictionary and Document MIRF Units

The individual cores used are the same for both the Dictionary and Document MIRF. Each core is composed of two U-shaped ferrite structures (Allen Bradley part no. UC 892–141C), which have been specially modified at the factory to permit a maximum of 0.0005 inch air gap in each leg when two such structures are joined together to produce a MIRF core. To drive each core, a twenty-turn primary winding is provided. This consists of two ten-turn windings distributed in such a manner as to minimize the leakage flux and the resulting noise signal (see Fig. 10). The primary winding drives the core from an 18-volt voltage source through a transistor switch driver. The output voltage induced upon each secondary winding is an essentially rectangular voltage pulse having a droop of 0.1 volt in 10 microseconds, from 0.8 volt at the leading edge to 0.7 volt just prior to the trailing edge. The maximum primary current, 0.7 ampere, occurs at 10 microseconds after the beginning of the pulse. To accommodate the expanded capacity of the MIRF document file (5,000 documents) three primary windings will be driven in parallel, so that a maximum driver current of 2.1 amperes is required.

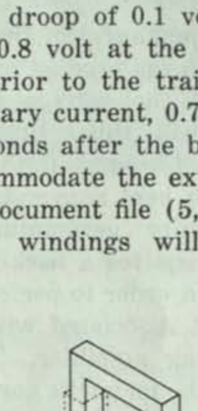


Figure 10. Details of Primary Windings.

The performance requirement of the magnetic circuits is that consistent and easily separable match and mismatch signals be generated at the diode end of the item wires (see Fig. 2) when a set of primary windings is driven. The design objective was that a maximum match signal of 0.1 volt and a minimum mismatch signal of 0.6 volt should be realized within 1.5 microseconds after the application of the primary drive pulses, and that pulsing of the MIRF cores be repeated for many cycles at a 50-kc clock rate. To achieve these goals, noise due to ringing and leakage flux had to be minimized.

A MIRF unit contains many cores (the Document MIRF has 234 and the Dictionary MIRF has 140), each with a separate primary winding; further, each core is associated with more than a thousand single-turn secondary windings. The secondary windings pass through or around all cores in the unit and so form a long rope. The capacitance between wires in the rope, the inductance of these wires, and the inductance of the primary windings are intercoupled in a very complex manner. In the development of the MIRF units, substantial noise on the secondary (item) windings was experienced due to ringing currents in the primary windings. This noise was reduced to a negligible level by inserting a Type D152 diode in series with each primary winding and shunting each primary by a 1000 ohm resistor. A low-amplitude noise signal of about 5 Mc, due to inductance and inter-item capacitance of the secondary windings, was also observed. Such noise could be reduced to a very low level by filtering at the input to the discriminating amplifier, but in the experimental system this was not necessary.

Noise due to leakage flux must be kept small in order to hold the maximum match signal at 0.1 volt. A secondary wire that represents a match item must pass outside all energized cores. Since in the worst case, 57 cores may be energized, the maximum permitted noise due to leakage flux at each core is less than 2 millivolts (this corresponds to a leakage flux of ¼ of one per cent at each core). In the experimental model two methods are used to reduce leakage flux. One is distributing the primary winding on the cores to compensate for the magnetic potential drop by a corresponding rise in magnetic potential at the points where the drop occurs. As Fig. 10 shows, the winding has a linear spacing except at the points where the air gaps occur; there two turns are closely spaced. The second method uses cancellation of induced voltages to reduce the effect of leakage flux. The common end of many item wires, instead of being connected to ground, as shown in the simplified diagram of Fig. 2, is actually connected to a wire that lies in the item wire rope and passes *outside* of all cores. The voltage induced in the "cancellation lead" at any core by leakage flux is approximately equal to that induced in item wires and is opposite in polarity (relative to the input terminals of the discriminating amplifier).

MECHANICAL DESIGN

1. *The MIRF Module*

Implementing the wiring-patterns-on-cores method of storage illustrated by Fig. 2 presented a challenging mechanical design problem. It was necessary that the physical structure containing the magnetic cores and the associated wiring be made in two parts that could be easily separated. It was desirable to fabricate submodules of wiring patterns, so that the permanently stored information could be changed mechanically in relatively small blocks.

Separate MIRF modules are used to store the information concerning document indexes and dictionary words. In each, the cores are arranged in a rectangular pattern and are supported by long bobbins. These bobbins are firmly attached to a base structure and carry the primary windings for the cores. A MIRF module is a complete assembly of magnetic cores, primary windings for the cores, and submodules of secondary windings with their associated diodes. The construction of a module is illustrated by the exploded view of Fig. 11. The principal parts of the assembly are the base, or coil bobbin, assembly and the item wiring trays.

The coil bobbin assembly consists of a field of paper bobbins (two per magnetic core) that are cemented to a ⅛-inch-thick phenolic board. Each bobbin carries a ten-turn winding. The windings on pairs of bobbins are connected in series to form the primary winding for one of the magnetic cores. An item tray is a $\frac{1}{16}$-inch thick phenolic board with a field of shallow bobbins that matches the field of coil bobbins. The bobbins on the item tray are slightly larger than the coil bobbins, permitting item trays to be stacked up on the coil bobbin assembly. One item tray can accommodate 286 item wires. The diodes that are connected in series with the secondary windings and form the input circuit to the discriminating amplifier are mounted on the edge of the item tray. A MIRF module is assembled by sliding up to seven item trays into position on the coil bobbin assembly. One set of U cores is then inserted into the set of coil bobbins and held in place by a plate with a silicone-rubber pad. The other set of U cores is then dropped into position on the opposite side of the bobbin coils. Finally, the top plate (also with a spongy pad) is dropped into position to hold the entire assembly intact. The two sets of U cores are held together under slight pressure from the silicone pads.

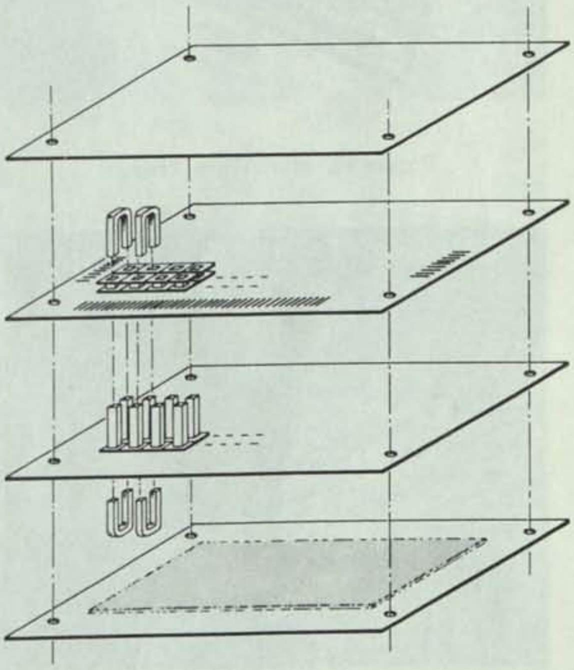


Figure 11. Exploded View of MIRF Module.

A complete item tray is shown in Fig. 12. The item wires start in the upper left corner of the trays, where they are connected to a common bus bar. They pass from left to right in the first row of cores, then back and forth until they emerge in the lower left center part of the tray. The wires then run to assemblies of diodes, where each wire is connected to its own individual diode. The output side of the diodes (the cathodes) are connected together and wired to a small connector, which is seen in the lower left hand portion of the tray. Even though each tray contains detailed wiring for 286 items, only two wires run from the tray to the external discriminating amplifier. Figure 12 also shows a pair of primary coil bobbins with the two U cores inserted. A closeup of a MIRF module with the top plate removed is shown in Fig. 13. The tops of one set of U cores can be seen as well as four item trays. The connectors for the output of the item trays can be seen in the lower center part of the photograph. The discriminating amplifier circuits (one for each of the seven item trays that can be included in a module) are located on the circuit board that is mounted in front of the magnetic module.

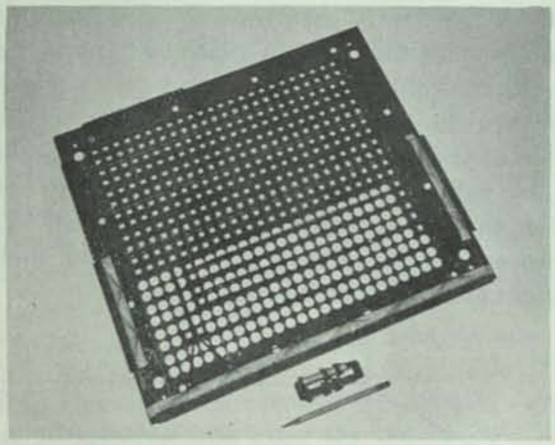


Figure 12. MIRF Item Tray.

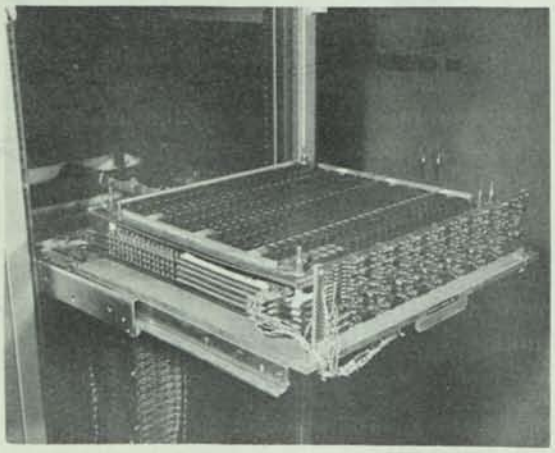


Figure 13. Close-up of Document MIRF Module (Top Plate Removed).

2. *Wiring of the Item Trays*

The item trays in the Document and Dictionary MIRF units store more than one-third of a million bits of information. To ensure the greatest possible accuracy of the wired-in information, two steps were taken. First, the raw data for the documents were computer-processed to give a set of punched cards that contain the detailed wiring information. Second, a wiring scheme was devised, which presented the detailed wiring information to a wireman in a very simple form, and which included a means of checking the accuracy of the wiring as the wiring was actually done. In this scheme, the path that a wire was to take was delineated by a set of lights in an array of incandescent lamps.

An over-all view of the item-tray wiring equipment (wiring aid) is shown in Fig. 14. The empty wiring tray is placed on the wiring jig in front of the operator. A card is then

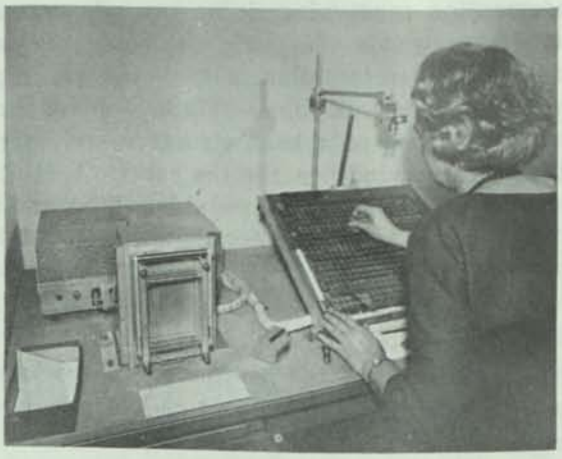


Figure 14. Over-all View of Item Tray Wiring Equipment.

placed in the punched-card reader and a pattern of lights is set up in the wiring jig. Number 36 Nyleze wire is taken from a spool through a tensioning device to the top of a special wiring tool (shown in the hand of the operator). The wire from the bottom of the wiring tool is first soldered to the common bus shown in the upper left part of the wiring tray. The tool is then moved along the path specified by the pattern of lights, leaving the wire wound in the desired pattern around the item tray bobbins. Correct wiring at a bobbin is indicated by a light turned on to yellow brilliance. If a light is off, or is on at white brilliance after the wiring tool passes a bobbin position, a wiring error is indicated.

3. *Alternative Method of Fabricating Item Trays*

Alternative methods of preparing wired-in information that may be more easily automated than stringing of small wire have been investigated. One alternative is illustrated by Fig. 15, which shows an item conductor in the form of a metallic path etched on a thin, copper-coated Mylar sheet (half-ounce copper on 2-mil Mylar). It will be noted that the item conductor is connected to a bus at the top of the sheet and to another bus at the bottom. These copper areas are used for connecting the item conductor to the common bus at one and to a diode at the other. This sheet contains one item, but two item conductors could easily be placed on one sheet, one being associated with one leg of the magnetic core and the other with the other leg. The experimental model contains a submodule of 75 items on Mylar sheets.

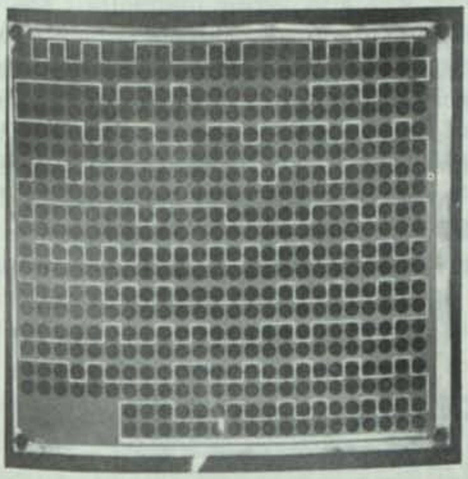


Figure 15. MIRF !tem Conductor Formed by Metallic Path on Mylar Sheet.

DELIVERED EXPERIMENTAL EQUIPMENT

The experimental Multiple Instantaneous Response File System is an all-solid state equipment. Transistor drive circuits capable of supplying two amperes of current to magnetic circuits, special discriminating amplifiers capable of operating reliably with a poor signal-to-noise ratio input signal, and transistor logic circuits were designed for high reliability, low cost, and moderate speed. About 300 current drive transistors, 2500 logic transistors, 2500 printed gate circuits (a group of 6 resistors, 2 capacitors and their interconnecting wiring on a passive substrate) and 5,000 diodes are used in the system. Except for sequences involving the input-output typewriter, the system operates synchronously under the control of clock pulses derived from a 50-kc transistor multivibrator.

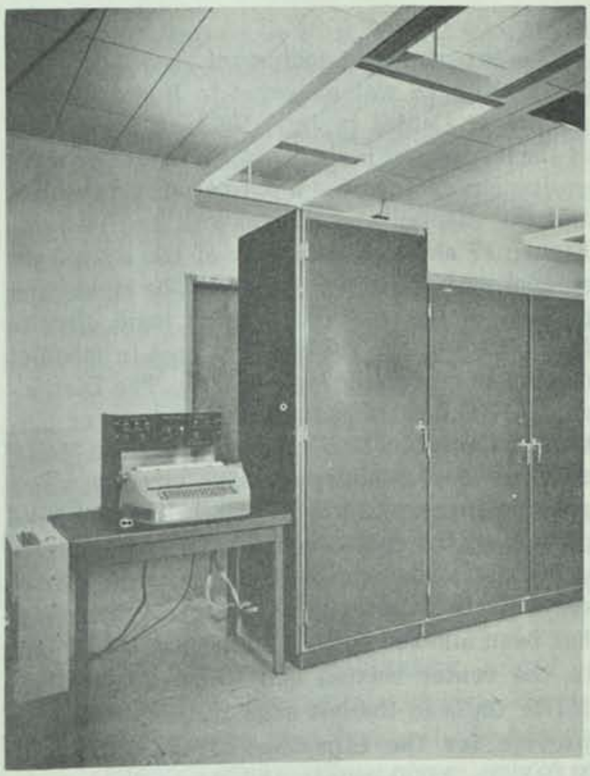


Figure 16. Front View of Experimental MIRF Equipment.

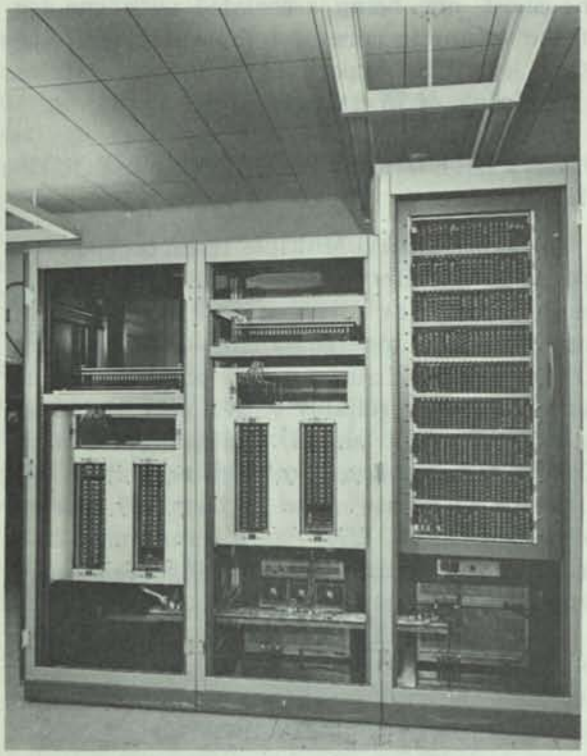


Figure 17. Rear View of Experimental MIRF Equipment (Doors Removed).

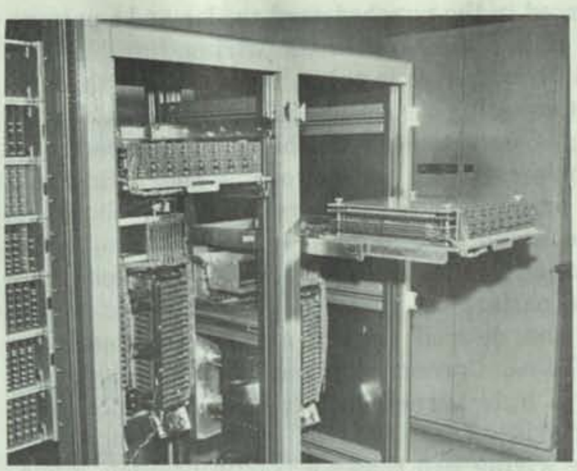
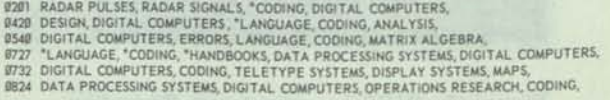


Figure 18. Front View of Equipment with Document MIRF Module in Extended Position.

The experimental equipment shown in Figs. 16 through 18 was delivered to Rome Air Development Center in July, 1963. A front view of the equipment is shown in Fig. 16. The main equipment cabinet, the input-output typewriter, and the display and control unit can be seen. Figure 17 shows a rear view of the equipment cabinet with the doors removed. The right hand portion of the cabinet contains logic circuits for control of the system, arranged in modules of plug-in transistor logic boards. The Dictionary MIRF unit is contained in the center portion of the cabinet. Directly beneath the MIRF unit are two modules of drive circuits which provide current to the MIRF. In the left hand portion of the cabinet are the Document MIRF and the transistor circuits for providing drive currents to it. It will be observed that space has been allowed for one additional MIRF unit in the center section and for two additional MIRF units in the left hand section. This is to provide for the expansion of the Dictionary MIRF to 3,000 words and expansion of the Document MIRF to 5,000 document indexes. A front view of the cabinets that house the MIRF units and their drivers is shown in Fig. 18. Here the Document MIRF unit has been pulled out to show it in its extended position. Below the MIRF units the wiring side of the transistor drive modules can be seen.

The format of the typewritten record of a search in the experimental model is shown in Fig. 19. The first two lines, "Stanford Research Institute Project 4110," etc., are a manually typed heading for the subsequent search. The heading was typed while the typewriter was effectively disconnected from the rest of the equipment. The search question consists of three words: "coding," "computers," "digital." This line was also typed manually. The rest of the printout is the machine's response to the search question. Seven documents responded. For each one, a four-digit accession number and the English words that describe the document are printed on a single line. The asterisk prefix on some words have been copied from the ASTIA abstract. It will be observed that the three search words appear in every respond-



```
                 STANFORD RESEARCH INSTITUTE PROJECT 411Ø

                   MULTIPLE INSTANTANEOUS RESPONSE FILE

CODING, COMPUTERS, DIGITAL.
Ø156  *CODING, DIGITAL COMPUTERS, DATA PROCESSING SYSTEMS, LANGUAGE,
Ø2Ø1  RADAR PULSES, RADAR SIGNALS, *CODING, DIGITAL COMPUTERS,
Ø42Ø  DESIGN, DIGITAL COMPUTERS, *LANGUAGE, CODING, ANALYSIS,
Ø54Ø  DIGITAL COMPUTERS, ERRORS, LANGUAGE, CODING, MATRIX ALGEBRA,
Ø727  *LANGUAGE, *CODING, *HANDBOOKS, DATA PROCESSING SYSTEMS, DIGITAL COMPUTERS,
Ø732  DIGITAL COMPUTERS, CODING, TELETYPE SYSTEMS, DISPLAY SYSTEMS, MAPS,
Ø824  DATA PROCESSING SYSTEMS, DIGITAL COMPUTERS, OPERATIONS RESEARCH, CODING,
```

Figure 19. Format of Typewritten Record of a Search.

ing set of indexes. It should be especially noted that the search words appear in different positions and different order in the different responding documents. This independence of order of the search words and the position of the corresponding descriptors in the document indexes is an important result of the superimposed coding of the search field.

CONCLUSIONS

From experience with the Experimental MIRF it is concluded that interrogation of the magnetic storage units and the over-all control of the system can be accomplished with reliable circuits of modest complexity. Storage of the document index information in wiring associated with arrays of cores that are physically separable appears feasible; arrays of cores can be separated, submodules of wired information can be changed, and the core arrays reassembled in a reasonably short time. More work on the mechanical design of the magnetic modules is needed, however, to permit easier and faster changing of the stored information. Based on the performance of the experimental model, which contained a file of more than 1,000 document indexes, it is concluded that with the present design a system building block should contain about 5,000 document indexes. It appears that as many as ten such building blocks could be combined in a system whose over-all control is little more complex than that for a single building block. Therefore it is concluded that files of the order of 50,000 indexes could be built with no major changes in the basic concepts or circuits used in the experimental model.

Easy communication between a human operator and the Experimental MIRF System has been demonstrated. The machine's response to a search question is essentially instantaneous in terms of human reaction time and the information content of the response is sufficient to allow the operator to start the document search with a general question and to use the information received to define a more specific question. In this way it is possible to home-in quickly on the documents of special interest. Several automatic features of the equipment have proved to be useful. One of these is the capability of accepting a synonym in the search question and automatically translating it into the synonymous descriptor contained in the machine's vocabulary. Another feature is the capability of automatically modifying the search question inserted by the human operator and initiating a new search. For example, if any of the input words have attached to them a "see-also" reference, that see-also reference will be substituted for the original word to form a new search question.

ACKNOWLEDGEMENTS

The development of the Experimental Multiple Instantaneous Response File was sponsored by the U. S. Air Force, Rome Air Development Center, under Contract No. AF 30 (602)–2772. The authors wish to acknowledge the contributions of their colleagues at Stanford Research Institute, especially C. B. Clark, D. C. Condon and V. Sanford.

REFERENCES

1. A. E. SLADE and C. R. SMALLMAN, "Thin Film Cryotron Catalog Memory," *Proc. of the Symposium on Superconductive Techniques for Computing Systems,* Washington, D. C., May 1960, published in *Solid State Electronics,* vol. 1, pp. 357–362 (September 1960).

2. J. R. KISEDA, H. E. PETERSEN, W. C. SEELBACH, and M. TEIG, "A Magnetic Associative Memory," *IBM J. of Res. and Dev.,* vol. 5, pp. 106–121 (April 1961).

3. J. GOLDBERG and M. W. GREEN, "Large Files for Information Retrieval Based on Simultaneous Interrogation of All Items," *Large Capacity Memory Techniques for Computing Systems,* M. C. Yovits, Ed., pp. 63–77, MacMillan Co., New York, 1962.

4. M. H. LEWIN, H. R. BEELITZ, and J. A. RAJCHMAN, "Fixed Associative Memory Using Evaporated Organic Diode Arrays," *AFIPS Conference Proceedings,* vol. 24, pp. 101–106 (November 1963).

5. E. L. YOUNKER, D. C. CONDON, C. H. HECKLER, JR., D. P. MASHER, and J. M. YARBOROUGH, "Development of a Multiple Instantaneous Response File—the AN/GSQ–81 Document Data Indexing Set," to be published as a Rome Air Development Center Technical Documentary Report.

6. E. H. FREI and J. GOLDBERG, "A Method of Resolving Multiple Responses in a Parallel Search File," *IRE Trans.*, EC–10, pp. 718–722 (December 1961).

7. T. L. DIMOND, "No. 5 Crossbar AMA Translator," *Bell Labs Record*, vol. 29, pp. 62–68 (February 1951).

SRI - MIRF

copy to Trudi

dupl. 2

DESIGN OF AN EXPERIMENTAL MULTIPLE INSTANTANEOUS RESPONSE FILE*

E. L. Younker, C. H. Heckler, Jr., D. P. Masher, and J. M. Yarborough
Stanford Research Institute
Menlo Park, California

SUMMARY

An experimental model of an electronic reference retrieval file in which all file entries are interrogated simultaneously has been designed and constructed. The experimental model is designed to store and search on a file of indexes to 5,000 documents. A document index consists of a decimal accession number and up to eight English word descriptors that are closely related to the contents of the document. The vocabulary required to describe the documents is held in a machine dictionary that has a design capacity of 3,000 words. In the model delivered to the sponsor, Rome Air Development Center, the storage capacity is only partially used. The specification for the delivered model calls for the storage of approximately 1,100 documents that were selected from the ASTIA (now DDC) Technical Abstract Bulletin and of the vocabulary needed to describe them (about 1,000 words). The document indexes and the dictionary words are stored in wiring patterns associated with arrays of linear ferrite magnetic cores.

A search question, consisting of one to eight descriptors in their natural English form, is entered by means of an electric typewriter. During entry of the search question, the dictionary magnetic store is interrogated by the alphabetic code of each search word. If a word is not contained in the dictionary, it is automatically rejected. After all words of the search question have been entered, the document magnetic store is interrogated by the search question in superimposed code form. The comparison between the search word and the document indexes is made for all documents simultaneously and the machine instantaneously determines if any documents in the file include the search question. If there are none, the machine indicates visually that there is no response. If there is at least one, the machine counts the number of responding documents and displays this number. Then it types out the indexes of all responding documents on the same typewriter that was used to ask the question.

INTRODUCTION

Memories that can be searched in parallel and from which stored information is retrieved on the basis of content have received considerable attention for application to retrieval file problems.[1, 2, 3, 4] This paper describes the design of an experimental retrieval file based on the work reported by Goldberg and Green.[3] Since the contents of the semipermanent magnetic memory used in the experimental file can be searched in parallel and multiple responses

* The work described in this paper was supported by Rome Air Development Center under Contract AF 30(602)-2772.

to the search question are permitted, the system is called MIRF—Multiple Instantaneous Response File.[5]

LOGICAL ORGANIZATION OF THE MIRF SYSTEM

The logical organization of the experimental MIRF system is illustrated by Fig. 1. Information pertaining to the document indexes and to the descriptors used in the document indexes is contained in two major units called MIRF units. A MIRF unit is basically a magnetic memory in which information is permanently stored in the wiring associated with the magnetic cores. The Document MIRF is the principal element of the system. It contains for each stored document index the document accession number and the descriptors (in coded form) that describe that document, as well as a superimposed search code that is used in the searching process. The Dictionary MIRF has two functions. During the input phase of operation it translates the alphabetic code of the English word descriptor that is entered from the typewriter into the binary serial number assigned to that English word for use inside the machine. During the output phase, the Dictionary MIRF translates the binary serial number of a word that is obtained during a search into the alphabetically coded form of that word.

After the binary serial number of an input English word has been generated, this binary number is translated by a logical process in the Search Code Generator into a search code that is assigned to the particular English word. The search codes of successive words of a search question are superimposed by adding them together, bit by bit by an inclusive-OR operation. When the search question is complete, the superimposed search code of the question is compared with the superimposed code section of the Document MIRF. Each document index whose search field includes the superimposed code of the search question is said to *respond* to the question. Frequently more than one document will respond. By a logical process for resolving multiple responses,[6] the accession number of a particular responding document is generated. Then the binary serial numbers of the English words contained in this document index are generated one at a time. By means of the Descriptor Selector, each serial number is transmitted to the Dictionary MIRF, where it is translated to the alphabetic code of the English word. This process is repeated for each responding document.

SYSTEM DESIGN

1. Magnetic Implementation of the MIRF Unit

The MIRF units of the experimental model use an interesting modification of the Dimond Ring[7] translator in which the drive and sensing functions are interchanged. Information is stored in unique wiring patterns associated with an array of linear ferrite cores as il-

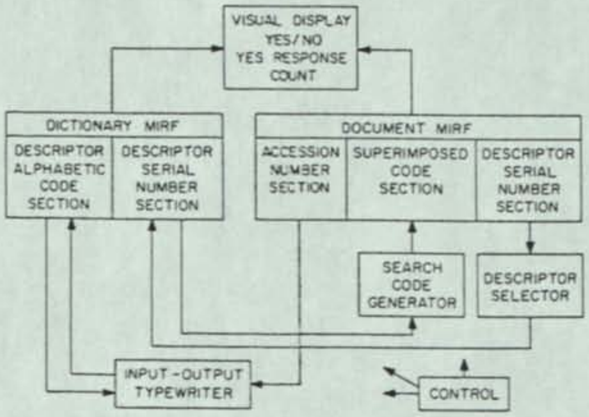


Figure 1. Simplified Block Diagram of MIRF Experimental Model.

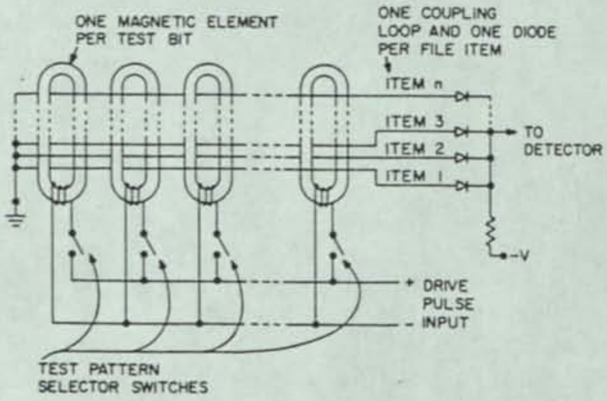

ONE MAGNETIC ELEMENT PER TEST BIT
ONE COUPLING LOOP AND ONE DIODE PER FILE ITEM
ITEM n
ITEM 3
ITEM 2
ITEM 1
TO DETECTOR
-V
+ DRIVE
PULSE
- INPUT
TEST PATTERN SELECTOR SWITCHES

Figure 2. Core-Wiring Arrangement for MIRF Memory.

lustrated by Fig. 2. Each item of stored information (a document index in the Document MIRF or a descriptor in the Dictionary MIRF) is represented by a conductor that passes through or around each associated core in a unique pattern determined by the information it contains. In series with each conductor is a diode. The cathodes of many diodes are connected together to form the input to a detector amplifier. Notice that one core is required for each bit of information, but that each core can be associated with a particular bit of many item conductors.

Each core has an input winding that can be selected by means of a switch. All cores whose selector switch is closed will be energized when a drive pulse is applied. A voltage will be induced in each item conductor that threads an energized core, but no voltage will be induced in conductors that do not thread the core. A test can be made on the information stored in many cores by selecting a particular set of cores and energizing them. In order for an item to match the test information, its conductor must pass outside of every energized core. Then no voltage will be generated in the item wire and the input to the detector amplifier will be held near ground through the item diode. Voltages will be induced in the conductors of items that do not match the test; the polarity of these voltages is chosen to back-bias the associated diodes. If no item matches the test information, a voltage will be induced in every item conductor and every diode will be back-biased. The input to the detector will then assume a significantly negative voltage. Thus, the presence or absence of desired stored information can be determined by applying the drive currents to a particular set of cores. This is a function of an associative or content-addressed memory: to indicate the presence or absence of certain information based on the detailed contents of a search question without regard to the actual location (or address) of that information.

Now consider in more detail how a bit of information of a search question is compared with information in a MIRF unit. Figure 3 illustrates how a test is made to determine whether or not the test bit is logically "included" in the stored information. This circuit is typical of those used in the superimposed section of the Document MIRF. One core is used to store the *k*th bit of many items. The *k*th bit of the search question is stored in a flip-flop whose *one* side is connected by way of an AND gate to a drive amplifier, which in turn is connected to the primary winding of the *k*th core. The conductor of an item whose *k*th bit is equal to *one* (Conductor 1) passes outside the *k*th core. On the other hand, the conductor of an item whose *k*th bit is equal to *zero* (Conductor 2) threads the core. If the flip-flop stores a *one*, the primary winding of the core will be energized when the timing pulse is applied to the AND gate. A voltage will be induced in Conductor 2 (indicating a mismatch) but none will be induced in Conductor 1 (indicating a match). If the flip-flop stores a *zero*, the primary winding will not be energized because the timing pulse will be blocked at the AND gate. No voltage will be induced in either conductor, and a match will be indicated on both lines. Therefore, it can be seen that a stored *one* bit includes both a test *one* and a test *zero*, while a stored *zero* bit includes only a test *zero*.

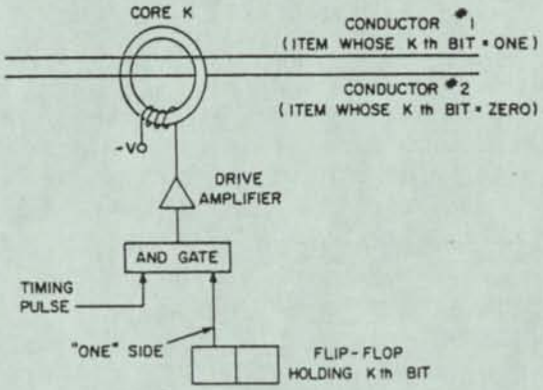


Figure 3. Circuit for Testing Inclusion.

The circuit for testing for identity between the test bit and the information stored in the MIRF is shown in Fig. 4. This circuit is typical of those used in the alphabetic descriptor portion of the Dictionary MIRF. The *j*th bit of many items is stored in a pair of cores j_A and j_B. The *j*th bit of the test question is stored in a flip-flop. In this case, both the *one* and *zero* sides of the flip-flop are connected to AND gates placed in the punched-card reader and a pattern of lights is set up in the wiring jig. Number 36 Nyleze wire is taken from a spool through a tensioning device to the top of a special wiring tool (shown in the hand of the operator). The wire from the bottom of the wiring tool is first soldered to the common bus shown in the upper left part of the wiring tray. The tool is then moved along the path specified by the pattern of lights, leaving the wire wound in the desired pattern around the item tray bobbins. Correct wiring at a bobbin is indicated by a light turned on to yellow brilliance. If a light is off, or is on at white brilliance after the wiring tool passes a bobbin position, a wiring error is indicated.

3. *Alternative Method of Fabricating Item Trays*

Alternative methods of preparing wired-in information that may be more easily automated than stringing of small wire have been investigated. One alternative is illustrated by Fig. 15, which shows an item conductor in the form of a metallic path etched on a thin, copper-coated Mylar sheet (half-ounce copper on 2-mil Mylar). It will be noted that the item conductor is connected to a bus at the top of the sheet and to another bus at the bottom. These copper areas are used for connecting the item conductor to the common bus at one and to a diode at the other. This sheet contains one item, but two item conductors could easily be placed on one sheet, one being associated with one leg of the magnetic core and the other with the other leg. The experimental model contains a submodule of 75 items on Mylar sheets.

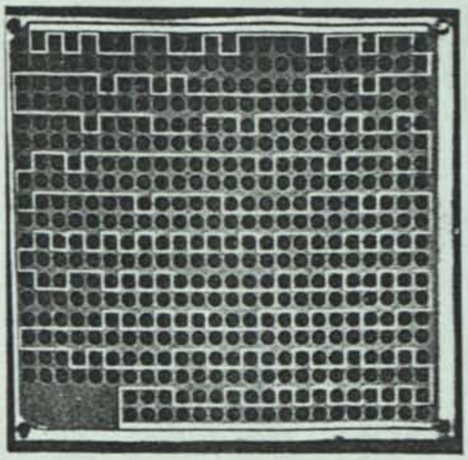


Figure 15. MIRF Item Conductor Formed by Metallic Path on Mylar Sheet.

DELIVERED EXPERIMENTAL EQUIPMENT

The experimental Multiple Instantaneous Response File System is an all-solid state equipment. Transistor drive circuits capable of supplying two amperes of current to magnetic circuits, special discriminating amplifiers capable of operating reliably with a poor signal-to-noise ratio input signal, and transistor logic circuits were designed for high reliability, low cost, and moderate speed. About 300 current drive transistors, 2500 logic transistors, 2500 printed gate circuits (a group of 6 resistors, 2 capacitors and their interconnecting wiring on a passive substrate) and 5,000 diodes are used in the system. Except for sequences involving the input-output typewriter, the system operates synchronously under the control of clock pulses derived from a 50-kc transistor multivibrator.

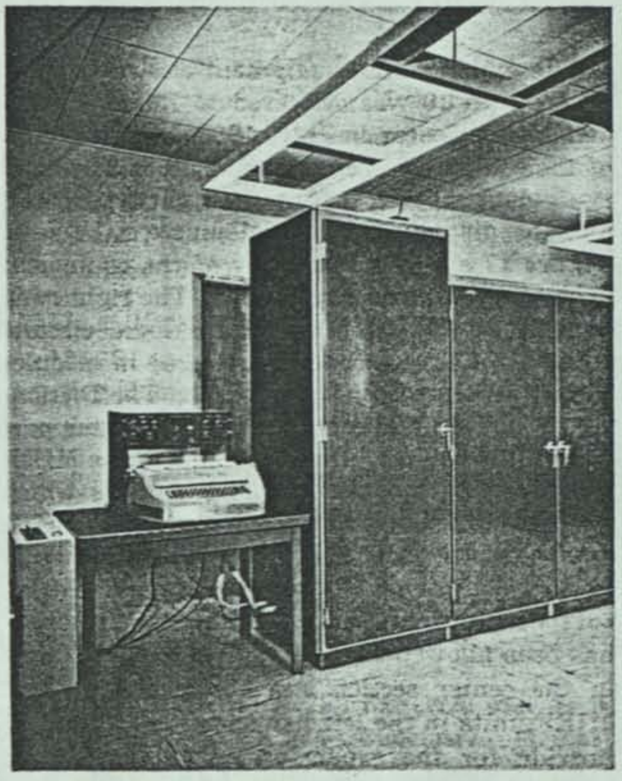


Figure 16. Front View of Experimental MIRF Equipment.

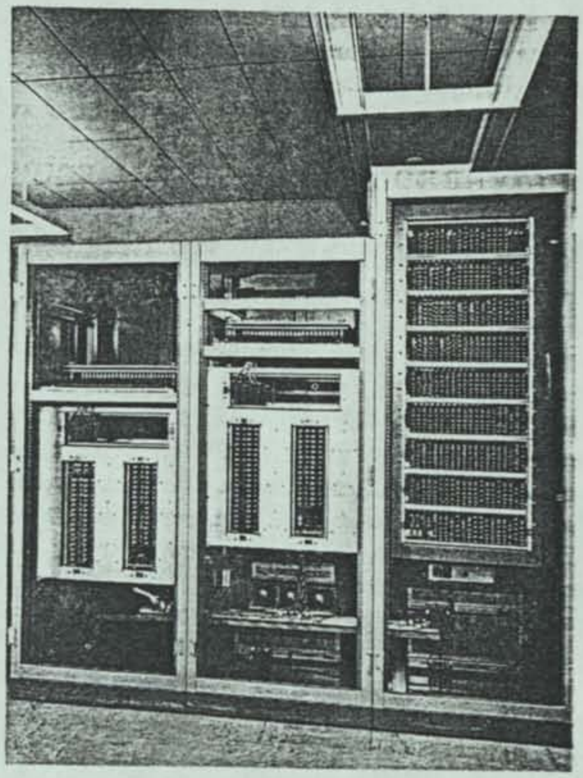


Figure 17. Rear View of Experimental MIRF Equipment (Doors Removed).

The experimental equipment shown in Figs. 16 through 18 was delivered to Rome Air Development Center in July, 1963. A front view of the equipment is shown in Fig. 16. The main equipment cabinet, the input-output typewriter, and the display and control unit can be seen. Figure 17 shows a rear view of the equipment cabinet with the doors removed. The right hand portion of the cabinet contains logic circuits for control of the system, arranged in modules of plug-in transistor logic boards. The Dictionary MIRF unit is contained in the center portion of the cabinet. Directly beneath the MIRF unit are two modules of drive circuits which provide current to the MIRF. In the left hand portion of the cabinet are the Document MIRF and the transistor circuits for providing drive currents to it. It will be observed that space has been allowed for one additional MIRF unit in the center section and for two additional MIRF units in the left hand section. This is to provide for the expansion of the Dictionary MIRF to 3,000 words and expansion of the Document MIRF to 5,000 document indexes. A

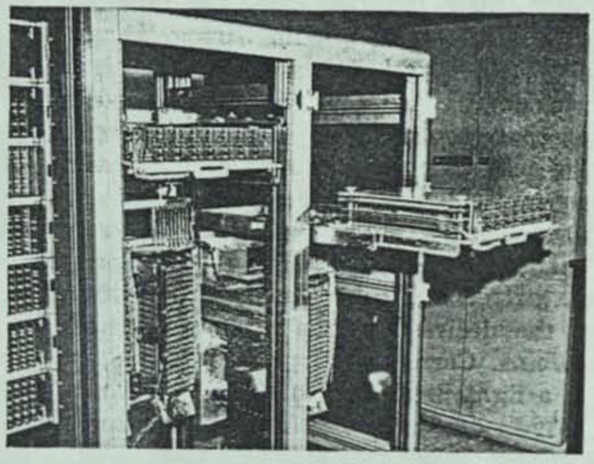
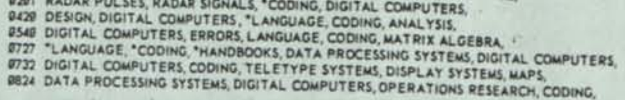


Figure 18. Front View of Equipment with Document MIRF Module in Extended Position.

front view of the cabinets that house the MIRF units and their drivers is shown in Fig. 18. Here the Document MIRF unit has been pulled out to show it in its extended position. Below the MIRF units the wiring side of the transistor drive modules can be seen.

The format of the typewritten record of a search in the experimental model is shown in Fig. 19. The first two lines, "Stanford Research Institute Project 4110," etc., are a manually typed heading for the subsequent search. The heading was typed while the typewriter was effectively disconnected from the rest of the equipment. The search question consists of three words: "coding," "computers," "digital." This line was also typed manually. The rest of the printout is the machine's response to the search question. Seven documents responded. For each one, a four-digit accession number and the English words that describe the document are printed on a single line. The asterisk prefix on some words have been copied from the ASTIA abstract. It will be observed that the three search words appear in every respond-



```
              STANFORD RESEARCH INSTITUTE PROJECT 411Ø

               MULTIPLE INSTANTANEOUS RESPONSE FILE

CODING, COMPUTERS, DIGITAL.
Ø156 *CODING, DIGITAL COMPUTERS, DATA PROCESSING SYSTEMS, LANGUAGE,
Ø2Ø1 RADAR PULSES, RADAR SIGNALS, *CODING, DIGITAL COMPUTERS,
Ø42Ø DESIGN, DIGITAL COMPUTERS, *LANGUAGE, CODING, ANALYSIS,
Ø54Ø DIGITAL COMPUTERS, ERRORS, LANGUAGE, CODING, MATRIX ALGEBRA,
Ø727 *LANGUAGE, *CODING, *HANDBOOKS, DATA PROCESSING SYSTEMS, DIGITAL COMPUTERS,
Ø732 DIGITAL COMPUTERS, CODING, TELETYPE SYSTEMS, DISPLAY SYSTEMS, MAPS,
Ø824 DATA PROCESSING SYSTEMS, DIGITAL COMPUTERS, OPERATIONS RESEARCH, CODING,
```

Figure 19. Format of Typewritten Record of a Search.

ing set of indexes. It should be especially noted that the search words appear in different positions and different order in the different responding documents. This independence of order of the search words and the position of the corresponding descriptors in the document indexes is an important result of the superimposed coding of the search field.

CONCLUSIONS

From experience with the Experimental MIRF it is concluded that interrogation of the magnetic storage units and the over-all control of the system can be accomplished with reliable circuits of modest complexity. Storage of the document index information in wiring associated with arrays of cores that are physically separable appears feasible; arrays of cores can be separated, submodules of wired information can be changed, and the core arrays reassembled in a reasonably short time. More work on the mechanical design of the magnetic modules is needed, however, to permit easier and faster changing of the stored information. Based on the performance of the experimental model, which contained a file of more than 1,000 document indexes, it is concluded that with the present design a system building block should contain about 5,000 document indexes. It appears that as many as ten such building blocks could be combined in a system whose over-all control is little more complex than that for a single building block. Therefore it is concluded that files of the order of 50,000 indexes could be built with no major changes in the basic concepts or circuits used in the experimental model.

Easy communication between a human operator and the Experimental MIRF System has been demonstrated. The machine's response to a search question is essentially instantaneous in terms of human reaction time and the information content of the response is sufficient to allow the operator to start the document search with a general question and to use the information received to define a more specific question. In this way it is possible to home-in quickly on the documents of special interest. Several automatic features of the equipment have proved to be useful. One of these is the capability of accepting a synonym in the search question and automatically translating it into the synonymous descriptor contained in the machine's vocabulary. Another feature is the capability of automatically modifying the search question inserted by the human operator and initiating a new search. For example, if any of the input words have attached to them a "see-also" reference, that see-also reference will be substituted for the original word to form a new search question.

ACKNOWLEDGEMENTS

The development of the Experimental Multiple Instantaneous Response File was sponsored by the U. S. Air Force, Rome Air Development Center, under Contract No. AF 30 (602)–2772. The authors wish to acknowledge the contributions of their colleagues at Stanford Research Institute, especially C. B. Clark, D. C. Condon and V. Sanford.

REFERENCES

1. A. E. Slade and C. R. Smallman, "Thin Film Cryotron Catalog Memory," *Proc. of the Symposium on Superconductive Techniques for Computing Systems,* Washington, D. C., May 1960, published in *Solid State Electronics,* vol. 1, pp. 357–362 (September 1960).

2. J. R. Kiseda, H. E. Petersen, W. C. Seelbach, and M. Teig, "A Magnetic Associative Memory," *IBM J. of Res. and Dev.,* vol. 5, pp. 106–121 (April 1961).

3. J. Goldberg and M. W. Green, "Large Files for Information Retrieval Based on Simultaneous Interrogation of All Items," *Large Capacity Memory Techniques for Computing Systems,* M. C. Yovits, Ed., pp. 63–77, MacMillan Co., New York, 1962.

4. M. H. Lewin, H. R. Beelitz, and J. A. Rajchman, "Fixed Associative Memory Using Evaporated Organic Diode Arrays," *AFIPS Conference Proceedings,* vol. 24, pp. 101–106 (November 1963).

5. E. L. YOUNKER, D. C. CONDON, C. H. HECKLER, JR., D. P. MASHER, and J. M. YARBOROUGH, "Development of a Multiple Instantaneous Response File—the AN/GSQ-81 Document Data Indexing Set," to be published as a Rome Air Development Center Technical Documentary Report.

6. E. H. FREI and J. GOLDBERG, "A Method of Resolving Multiple Responses in a Parallel Search File," *IRE Trans.*, EC-10, pp. 718-722 (December 1961).

7. T. L. DIMOND, "No. 5 Crossbar AMA Translator," *Bell Labs Record*, vol. 29, pp. 62-68 (February 1951).

AFIPS

CONFERENCE PROCEEDINGS

VOLUME 25

1964

SPRING JOINT COMPUTER CONFERENCE

1964
SPARTAN BOOKS, INC.
Baltimore, Md.
CLEAVER-HUME PRESS
London

References:

(1) *Études de Documentation Automatique, Rapport No. 1: Application d'un Modèle de Langage Formel à des Langages Documentaires.* Paris: SEMA, September 1961.

(2) *Éutdes de Documentation Automatique, Rapport No. 2: Application d'un Modèle Descriptif des Calculateurs Électroniques a des Équipements Existants.* Paris: SEMA, December 1961.

(3) *Études de Documentation Automatique, Rapport No. 3: Conception d'un Systeme Documentaire.* Paris: SEMA, March 1962.

(4) *Études de Documentation Automatique, Rapport No. 4: Approche Algébrique de la Pertinence.* Paris: SEMA, June 1962.

(5) Lattes, Robert, et al. "Les Langages documentaires—Modèle descriptif et problèmes fondamentaux," *Symbolic Languages in Data Processing* (proceedings of the Symposium on Symbolic Languages in Data Processing, Rome, 1962), pp. 653–674. New York/London: Gordon and Breach Science Publishers, 1962.

2.123 SOUTHWESTERN LEGAL FOUNDATION

Hillcrest at Daniels, Dallas 5, Tex.
ROBERT A. WILSON, *Project Director*

The purpose of the project is to provide an automated document retrieval system by which a researcher may retrieve from a library of legal materials stored on magnetic tape all the authorities which are pertinent to his question under search. Current efforts are directed to the storage and retrieval of court decisions although the present system should also be applicable to pertinent legislation, agency regulations, law review articles, and treatises.

The full text of the cases is copied directly from the reporter volume onto punchcards in normal language, and each case receives an identifying "document number." The text is then transferred to magnetic tape. Special machine programs written for the IBM 1401 or 1410 computer cause the machine to break the text down into its component words and to process them so that every word is listed and accounted for in a master word list, and the text location of every significant word is recorded in a machine-operated "root index".

Searching utilizes the "keywords in combination" approach. Precedents involving closely analogous fact patterns may be retrieved by including fact words, as well as legal terms, in the search request. Only one grammatical form of a keyword need be used in a search statement because the root index provides automatic access to every other form of the same word. Only words found in the decisions are used as search terms, and the researcher is furnished an alphabetical listing of these to aid in preparing his request. Synonyms and phrases may be used in search request statements.

Progress has been made in several areas of the project. Testing has been completed on the stored library of 60 arbitration cases. The text of 246 federal court decisions, dealing with the taxation of oil and gas transactions, has been keypunched, stored on magnetic tape, and machine indexed for automatic searching on an IBM 1401 computer. The stored cases contained a total of 403,350 words and used a vocabulary of 12,144 individual words. These were condensed automatically into a root index of 5,584 search terms by eliminating repetitions and nonsignificant words. The word list derived from a previously stored library of 60 decisions was used as the initial stored vocabulary for indexing the oil and gas taxation cases. A batch of 6 separate oil and gas tax questions was searched in 25 minutes, and a batch of 14 questions was searched in 44 minutes, exclusive of printout time. The last search involved machine processing of over 17,300 stored index records.

An additional project involves experimentation to determine the best methods of using optical page reading machines for rapid storage (and perhaps preliminary indexing) of full text materials for computer retrieval.

References:

(1) Wilson, Robert A. "Computer Retrieval of Case Law," *Southwestern Law Journal,* vol. 16, no. 3, September 1962, pp. 409–437. Also published in *Proceedings, Seminar on Use of Electronic Computers for Legal Research* (San Jose State College, San Jose, Calif., May 24, 1963).

(2) Wilson, Robert A. "Optical Page Reading Machines: Their Impact on Document Retrieval Systems." (In press.)

(3) Wilson, Robert A. *Videotape of a Live Demonstration of Case Law Retrieval by Computer.* Presented at Stanford University Computation Center, May 1963.

STANFORD RESEARCH INSTITUTE 2.124

Menlo Park, Calif.
E. LeROY YOUNKER, *Project Leader* 11/64

SRI-MIRF

An experimental model of an electronic reference retrieval file in which all file entries are interrogated simultaneously has been designed and constructed. The purpose of this work is to demonstrate the usefulness of a rapid-feedback, man-machine relationship in a data retrieval system.

The experimental model is designed to store the index to 5,000 documents. Each document is given an accession number and is described by up to eight English words (descriptors) selected from a 3,000-word dictionary. The delivered model will contain a 1,000-word dictionary and the index to 1,100 documents. A search question, consisting of one to eight descriptors in their natural English form, is entered by means of an electric typewriter. The machine indicates immediately whether or not any file item satisfies the search question, and if so, how many file items respond. The machine then resolves multiple responses and types out the accession number and full set of descriptors of each responding document.

The document index and the words of the dictionary are stored in arrays of linear ferrite magnetic cores. During entry of the search question, the dictionary magnetic store is interrogated by the alphabetic code of each search word. If any word is not contained in the dictionary, it is automatically rejected. After all words of the search question have been entered, the document magnetic store is interrogated by the search question in superimposed code form. Response to a word validity test or to the file search is obtained in less than 5 microseconds.

Design and construction of the Multiple Instantaneous Response File (MIRF) experimental model have been completed and checkout of the equipment is underway. Delivery to the project sponsor, the Rome Air Development Center, U.S. Air Force, will be made during the summer of 1963.

Reference:

(1) Goldberg, J., et al. *Multiple Instantaneous Response File,* Final Report, SRI Project 3101, RADC Technical Report TR 61–233, prepared under Contract AF 30(602)–2142. Menlo Park, Calif.: Stanford Research Institute, August 1961. (AD–266 169)

2.125 STICHTING STUDIECENTRUM VOOR ADMINISTRATIEVE AUTOMATISERING [THE NETHERLANDS AUTOMATIC DATA PROCESSING RESEARCH CENTER]

6 Stadhouderskade, Amsterdam, Netherlands
L. M. C. J. SICKING, *Head, the Library and Documentation Department*

The Netherlands Automatic Information Processing Research Center and the Research Center on Documentation of the Netherlands Institute of Documentation and Filing are cooperating in a project concerning the automatic analysis and handling of literature.

The purpose of the project is to develop a number of rules which are applicable to the automatic analysis and handling of professional literature in the field of the microsocial consequences of automation.

The initial subject material to be reviewed will consist of a hundred English and American publications dealing with the above-mentioned subject. After preliminary rules are developed, a larger literature collection, containing publications in languages other than English, will be reviewed.

Microcard equipment with selection and scanning devices will be utilized in the project.

During the last 3 months a list of keywords and a classification have been prepared.

2.126 SYSTEM DEVELOPMENT CORPORATION

Special Development Department, 2500 Colorado Avenue, Santa Monica, Calif.
ELDRIDGE ADAMS

The purpose of this project is to investigate the utility of machine-prepared indexes of appellate decisions (using the word "indexes" in a broad sense). It is felt that such indexes, when published, will provide at least some of the benefits of computerized legal search to those who cannot afford to use a computer.

The project was begun in the Spring of 1962, and has been conducted on a part-time basis. An experimental data-base of 37 California Supreme Court labor decisions has been keypunched and verified. The IBM 1401 computer has been used because it is widely available for demonstration, its use is compatible with other legal data-processing projects, and because there is advantage in its variable word length. Experimental routines for indexing, abstracting, editing and preparing concordances have been debugged.

The next phase will involve enlargement of the variety of indexes prepared, and circulation of them among potential users for evaluation.

Reference:

(1) Adams, Eldridge and Carabillo, Virginia. "Data Processing and the Law," *System Development Corporation Magazine,* vol. 5, no. 8, Summer 1962, pp. 1–5. Available from OTS, PB 164 251, Xerox $1.10.

SYSTEM DEVELOPMENT CORPORATION 2.127

Center for Research in System Development,
2500 Colorado Avenue, Santa Monica, Calif.
HAROLD BORKO, *Project Leader*

The activities of the Information Retrieval and Linguistics Project may be divided into three areas of work. Documentation and indexing studies concentrate on deriving automatic and semiautomatic procedures for indexing, classifying, and abstracting documents. The studies in linguistics and communication have as their objectives the explication of certain linguistic information from text to assist in machine processing of text and the identification of those psychological factors that facilitate man-machine communication using a natural-language vocabularly. Automated content analysis represents a new area and together with the fact retrieval study completes the current scope of the project.

I. Indexing and Abstracting (Harold Borko, Lauren B. Doyle, and Ronald E. Wyllys)

Work has continued on a mathematically derived classification system. In earlier studies the technique was applied to *Psychological Abstracts* and to *IEEE Transactions on Electronic Computers.* The initial results indicated that mathematically derived classification systems can be applied to abstracts of documents in the computer field (1). The programs for this technique were written for the IBM 7090 computer.

Two new sets of documents obtained from *Psychological Abstracts* are being analyzed in order to determine the reliability and consistency of factors previously derived and reported (1). In addition, a comparison will be made between machine (i.e., automatic) classification and human classification of the documents into the derived categories.

NSF-64-17

RENT RESEARCH
D DEVELOPMENT

NTIFIC
UMENTATION

13

onal Science Foundation
ce of Science Information Service

Nov
ember 1964

copy to Trudi

DE LUCIA

SRI - MIRF

58 Chestnut Hills
New Hartford, N.Y.
13413
28 August 1997

Dear Charles,

I will write this letter so that I can get it off to you today. Please return this material to me after you are done with it.

1. PACER -

In writing a book on the pre-197 history of online search services & technology you must include PACER. See the attached abstract. PACER represented the first true implementation of a work station in terms of todays objectives. For example:

- Integrated Data Base, i.e., alpha-numeric - documents; tabular, aerial photos, electronic intercept, graphics
- 48 intelligent terminals - BR-90s and RCA
- Variable function keys
- etc

Contact Bernie DiTano (formerly RADC now TRC at 315-33776- for a true appreciation of what it could do in 1971, 10-15 years ahead of the field in general.

2. MIRF - Multiple Instataneous Response File (microfiche attached)

3. Minicard - E.K. write-up and photographs

4. Minicard Viewer-Processor (copies attached)

5. Automatic Disseminator - Built by Magnavox and also installed in AFCIN (1959-60) to disseminate copies of incoming documents to customers with pre-defined requirements. I have no documentation on this. The mentor of AFCIN who got RADC to buy this equipment was Captain Robert Laidlaw.

6. Report by Tom Bagg & E. Stevens

7. Brochure on the BR-90

8. A proposal on the Magnacard. I'm not sure if this contains and information of value to you.

Please return all of this information to me the reason I could not send the unclassified PACER report to you is because of the distribution limitation on the report.

Good luck,

Al De Lucia

SRI-MIRF

copy to Trudi
from file on a poor file printer

AD-609 126

AD 609126

dup 3

RADC-TDR-63-414
Final Report

DEVELOPMENT OF A MULTIPLE INSTANTANEOUS RESPONSE FILE
THE AN/GSQ-81 DOCUMENT DATA INDEXING SET

TECHNICAL DOCUMENTARY REPORT NO. RADC-TDR-63-414

October 1964

129 p

COPY ✓ OF 3 [illegible]
HARD COPY $ 4.00
MICROFICHE $. 1.00

Information Processing Branch
Rome Air Development Center
Research and Technology Division
Air Force Systems Command
Griffiss Air Force Base, New York

Project No. 4594, Task No. 459402

(Prepared under Contract No. AF 30(602)-2772 by Stanford Research Institute, Menlo Park, California.)

ARCHIVE COPY

FOREWORD

1. Personnel

A. Major Participants

C. B. Clark, Senior Research Engineer--One third time
T. J. Drevek, Engineering Associate--One half time
C. H. Heckler, Jr., Research Engineer--Two thirds time
D. P. Masher, Senior Research Engineer--One half time
V. Sanford, Engineering Associate--One half time
J. M. Yarborough, Research Engineer--Full time
E. L. Younker, Senior Research Engineer, Project Leader--Full time

B. Other Contributors

D. C. Condon, Research Engineer
D. F. Ford, Programmer
J. Goldberg, Senior Research Engineer
M. W. Green, Senior Research Engineer
W. H. Kautz, Staff Scientist
W. K. MacCurdy, Senior Research Engineer
R. C. Singleton, Senior Research Mathematical Statistician

C. Draftsman Designer

D. B. Bell

D. Technicians

R. Bennett
N. A. Dickey
N. P. Krea

2. Acknowledgments

The authors would like to acknowledge the support and encouragement of Mr. M. B. Adams and Mr. J. R. Anderson of the Computer Techniques Laboratory and of Mr. R. Ferris and Mr. S. Stromick of the Rome Air Development Center. The many contributions of other members of the laboratory are also gratefully acknowledged.

3. Authors of the Final Technical Report

E. L. Younker
D. C. Condon
C. H. Heckler, Jr.
D. P. Masher
J. M. Yarborough

Key Words: Computer - Storage and Retrieval

ABSTRACT

An experimental model of an electronic reference retrieval file in which all file entries are interrogated simultaneously has been designed and constructed. The principal purpose of this work is to demonstrate the usefulness of a rapid-feedback, man-machine relationship in a data retrieval system.

The experimental model (designated the AN/GSQ-81 Document Data Indexing Set) is designed to store the indexes to 5,000 documents. Each document is given an accession number and is described by up to eight English words (descriptors) selected from a 3,000 word dictionary. The delivered model contains a 1,000 word dictionary and the indexes to 1,100 documents. A search question, consisting of one to eight descriptors in their natural English form, is entered by means of an electric typewriter. The machine indicates immediately whether or not any file item satisfies the search question, and if so, how many file items respond. The machine then resolves multiple responses and types out the accession number and full set of descriptors of each responding document.

The document indexes and the words of the dictionary are stored in wiring patterns associated with arrays of linear ferrite magnetic cores. During entry of the search question, the dictionary magnetic store is interrogated by the alphabetic code of each search word. If the word is not contained in the dictionary, it is automatically rejected. After all words of the search question have been entered, the document magnetic store is interrogated by the search question in superimposed code form. Response to a word validity test or to the file search is obtained in less than six microseconds.

This equipment can handle synonymous input descriptors and has the capability for automatically modifying the manually inserted search question according to certain logical rules. New searches based on the modified search question (for example, substitution of a see-also reference for one of the original descriptors) are initiated automatically.

PUBLICATION REVIEW

This report has been reviewed and is approved. For further technical information on this project, contact Mr. Stephen Stromick, EMIIH, Ext. 71105.

Approved: Stephen Stromick
STEPHEN STROMICK
Project Engineer
Directorate of Intelligence & Electronic Warfare

Approved:
FRANK J. TOMAINI
Chief, Information Processing Laboratory
Directorate of Intelligence & Electronic Warfare

FOR THE COMMANDER:
IRVING J. GABELMAN
Director of Advanced Studies

3. The Experimental Model

A. General Description

(1) Functions of the Document Data Indexing Set

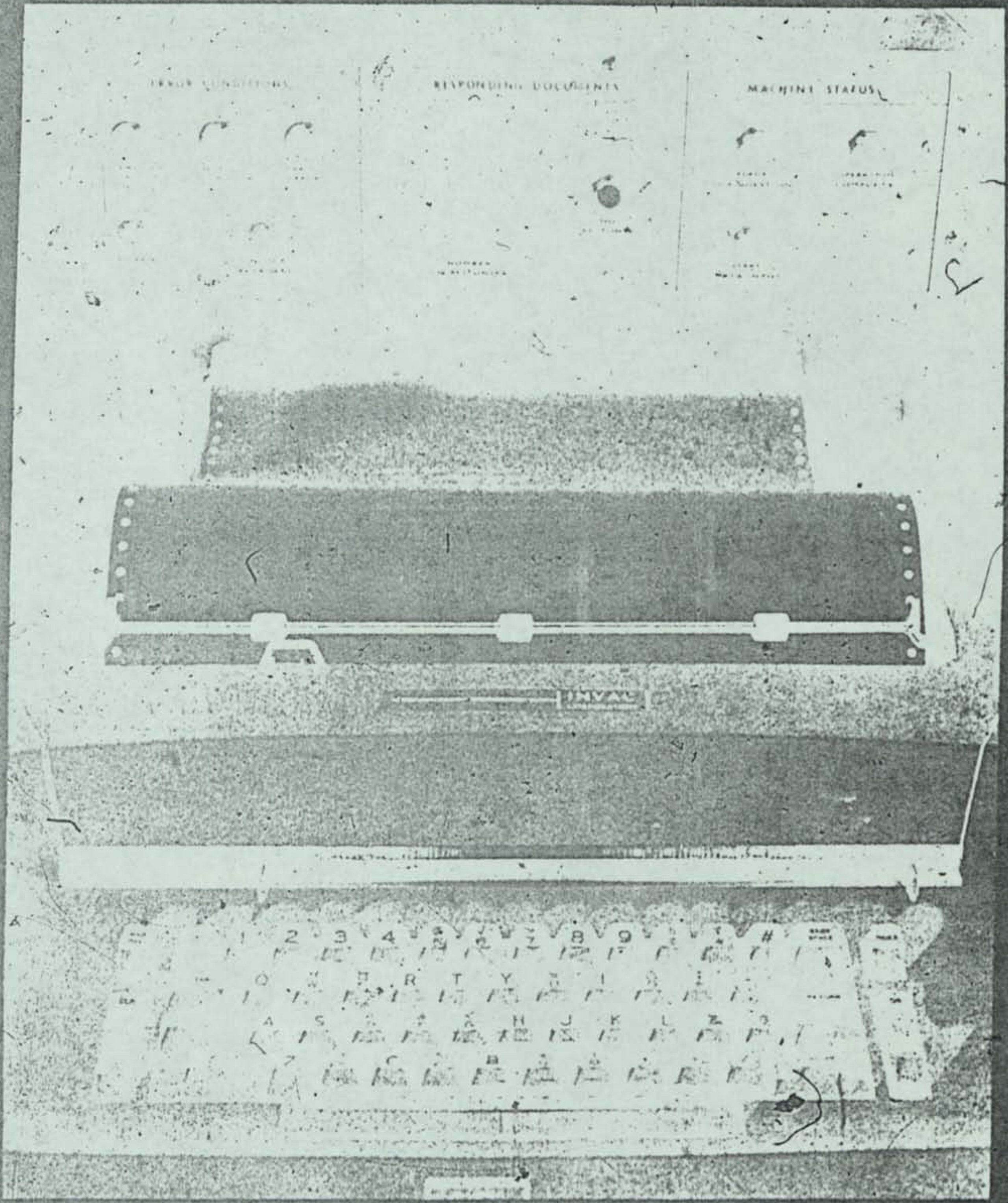
The AN/GSQ-81 Document Data Indexing Set is basically a file of document indexes and a means of retrieving particular document indexes of interest. This model contains the indexes of approximately 1000 documents that have been selected from the ASTIA Technical Abstract Bulletin. Each document is indexed by an accession number and a group of key words that describe the contents of the document. The vocabulary required to describe the 1000 stored documents contains about 1000 words. For the purpose of translating between the English formof these words and the coded form used inside the machine, a dictionary of 1000 words is contained in the equipment. The basic design of the data indexing set provides for expansion of 5000 document indexes and a 3000 word vocabulary.

The basic function of the document data indexing set is to permit the retrieval of document indexes that are related to particular subjects of interest. It allows an operator to ask a question about what is contained in the file in the form of a group of English words by typing these words on an ordinary typewriter. The machine compares the word used in the search question with the words that are used to describe the documents that are contained in the file. The comparison between the search words and the documents is made for all documents simultaneously. The machine instantaneously determines if any documents in the file include the search question. If there are none, the machine indicates visually that there is no response. If there is at least one, the machine cou ts the number of responding documents and indicates visually this number. Then it types out the indexes of all responding documents on the same typewriter that was used to ask the question. There is essentially no delay between the signal that starts the search and the beginning of typing out the responding documents. Because the results are immediately available, and because they have enough useful information to give a good idea of what each document is about, this document data indexing set makes it feasible to start the document search with a general question and to use the information received to define a more specific question. In this way it is possible to "home in" quickly on the documents of special interest.

This equipment has the capability for automatically modifying the search question inserted by the human operator and initiating a new search. In one type of machine initiated search, the original search question is modified by information associated with the input question. If any of the input words have attached to them a "see also" reference, that "see also" reference will be substituted for the original word to form the new search question. A second kind of machine initiated search uses information obtained from responding documents to modify the original search question. In this case, words appearing in responding documents that are marked by an asterisk are stored in the machine memory and later are used to replace a word in the original search question.

(2) Logical Organization of the Document Data Indexing Set

The logical organization of the Document Data Indexing Set is illustrated by Fig. 1. Information pertaining to the document indexes and to the key words used in the document indexes is contained in major units called MIRF. A MIRF is basically a magnetic memory in which information is permanently stored in the wiring associated with the magnetic cores. The Document MIRF is the principal element of the system. It contains for each stored document index the document accession number and the key words (in coded form) that describe that document as well as a search code field that is used in the searching process. The Dictionary MIRF translates during the input phase of operation from the alphabetic code of the English word descriptor that is entered from the typewriter to the binary serial number assigned to that English word for use inside the machine. During the output phase of operation, the Dictionary MIRF translates from the binary serial number of a word that is obtained during a search to the



Frontispiece

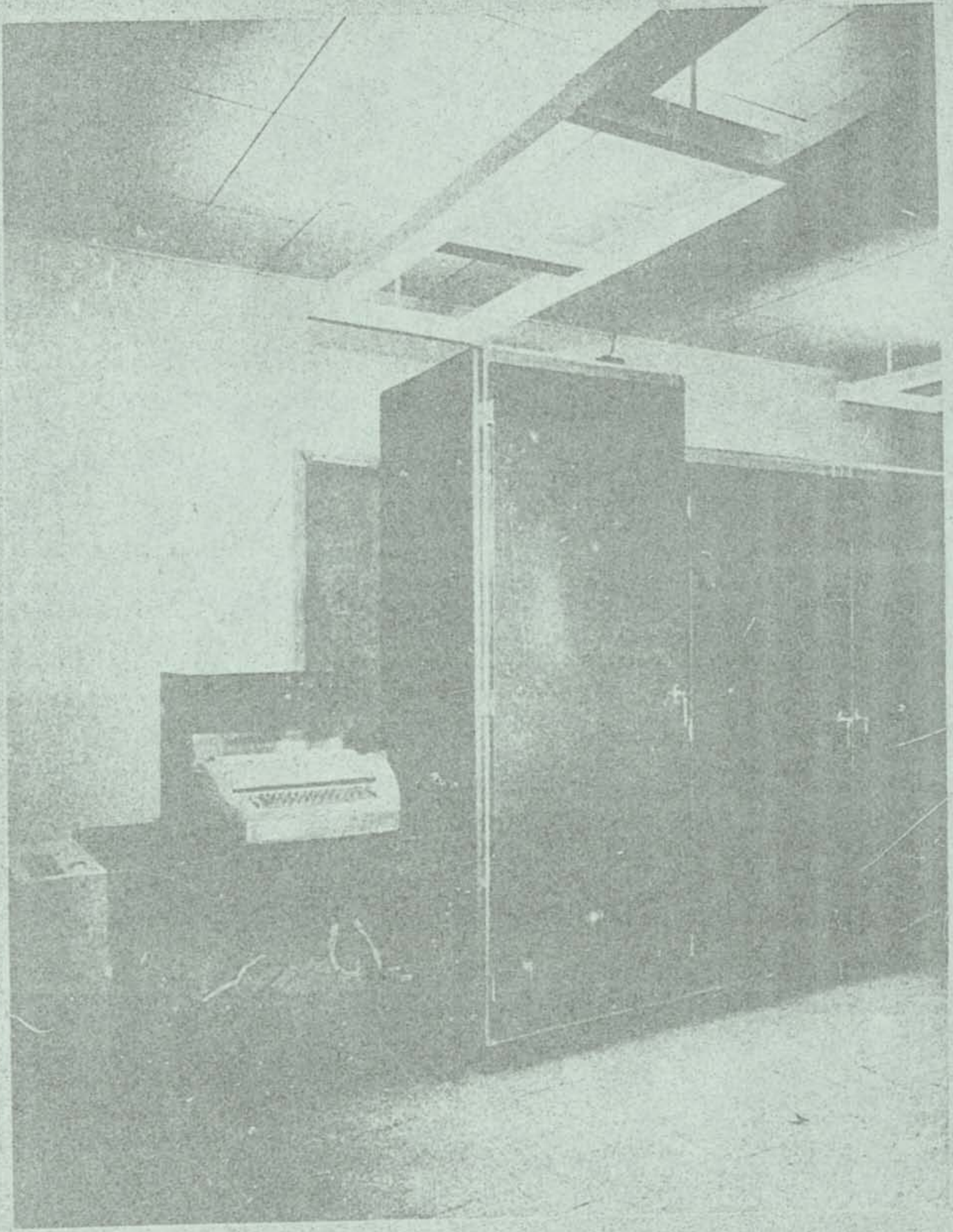


Figure 2. Front View of Document Data Indexing Set

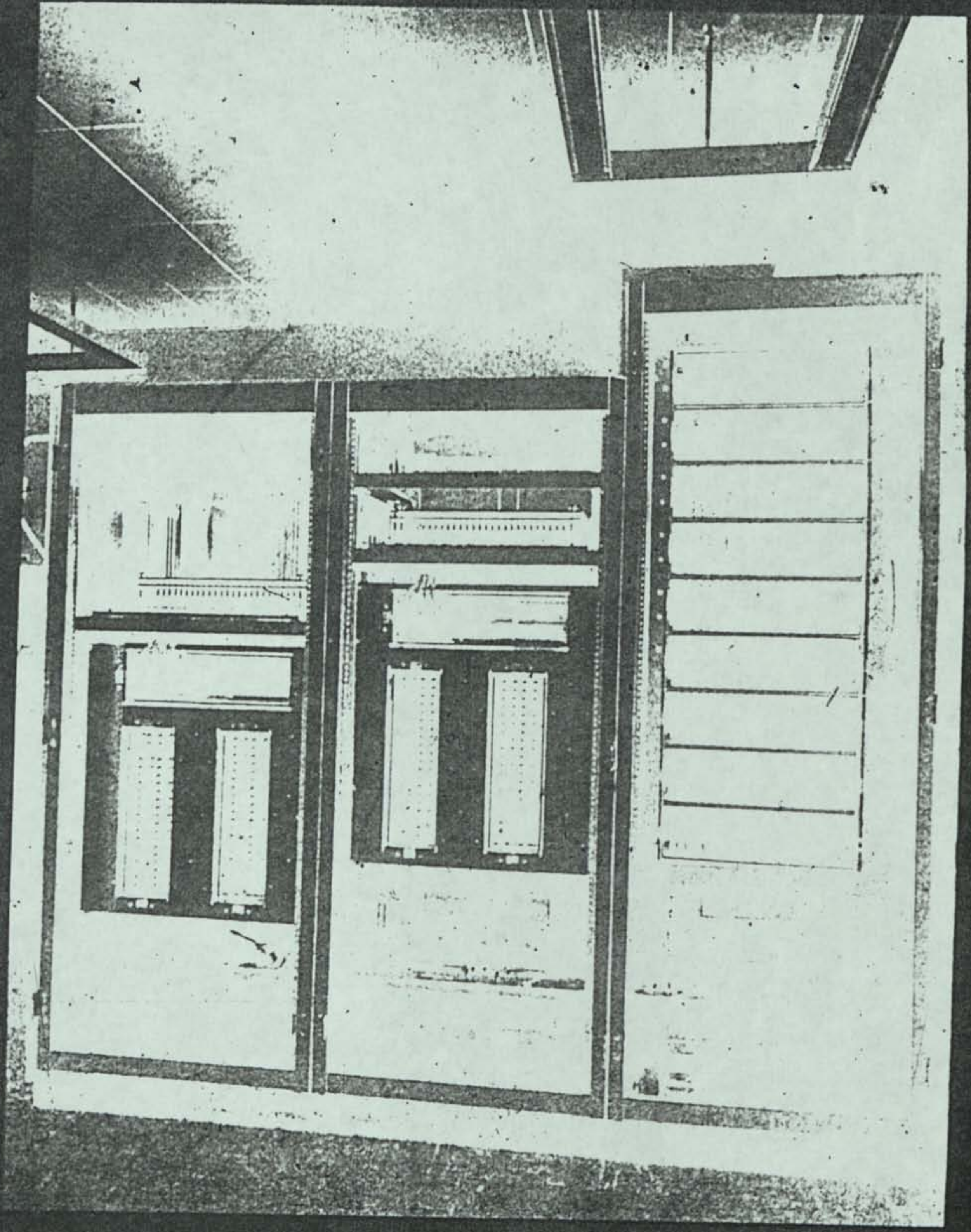


Figure 4. Rear View of Equipment Cabinet (doors removed)

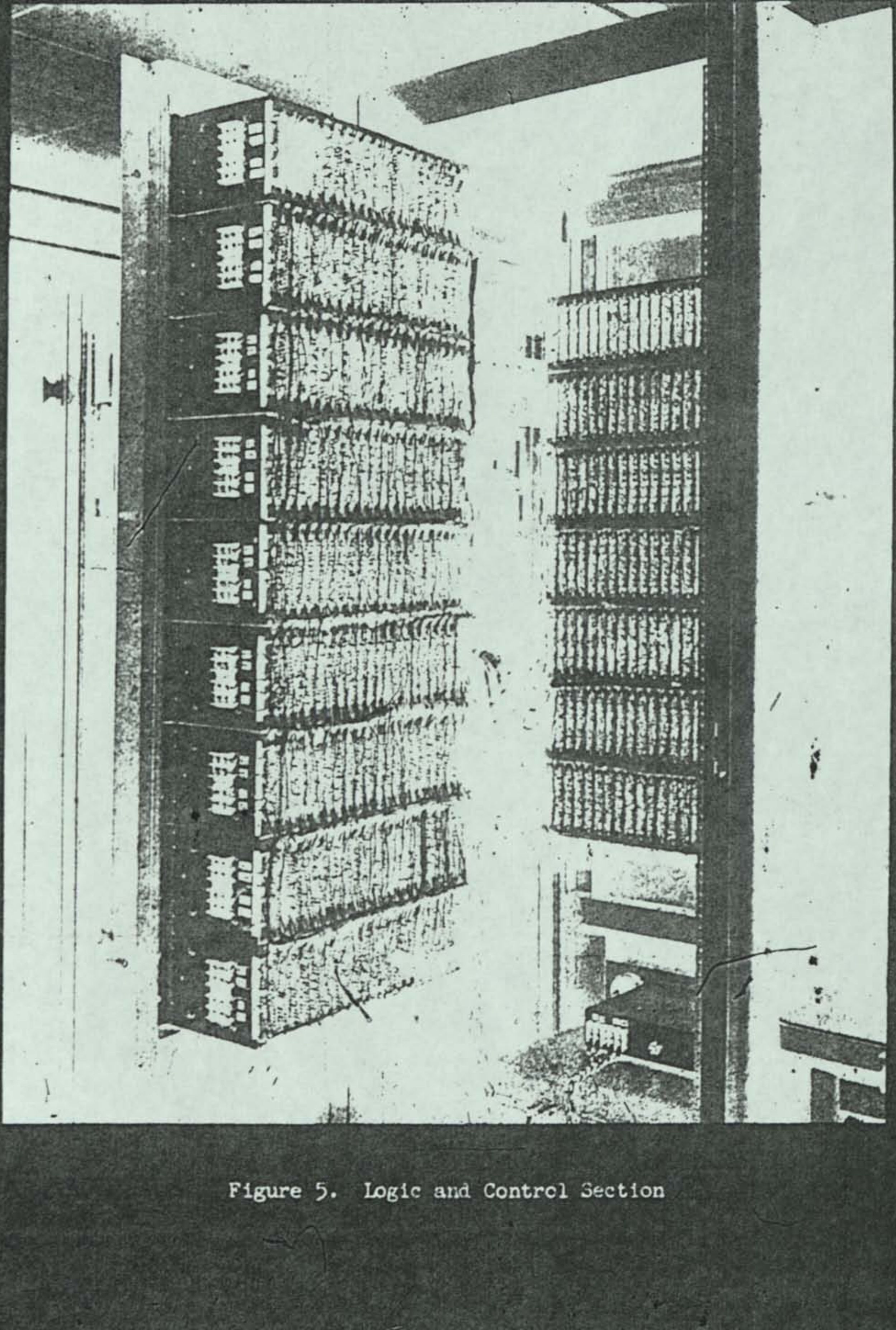


Figure 5. Logic and Control Section

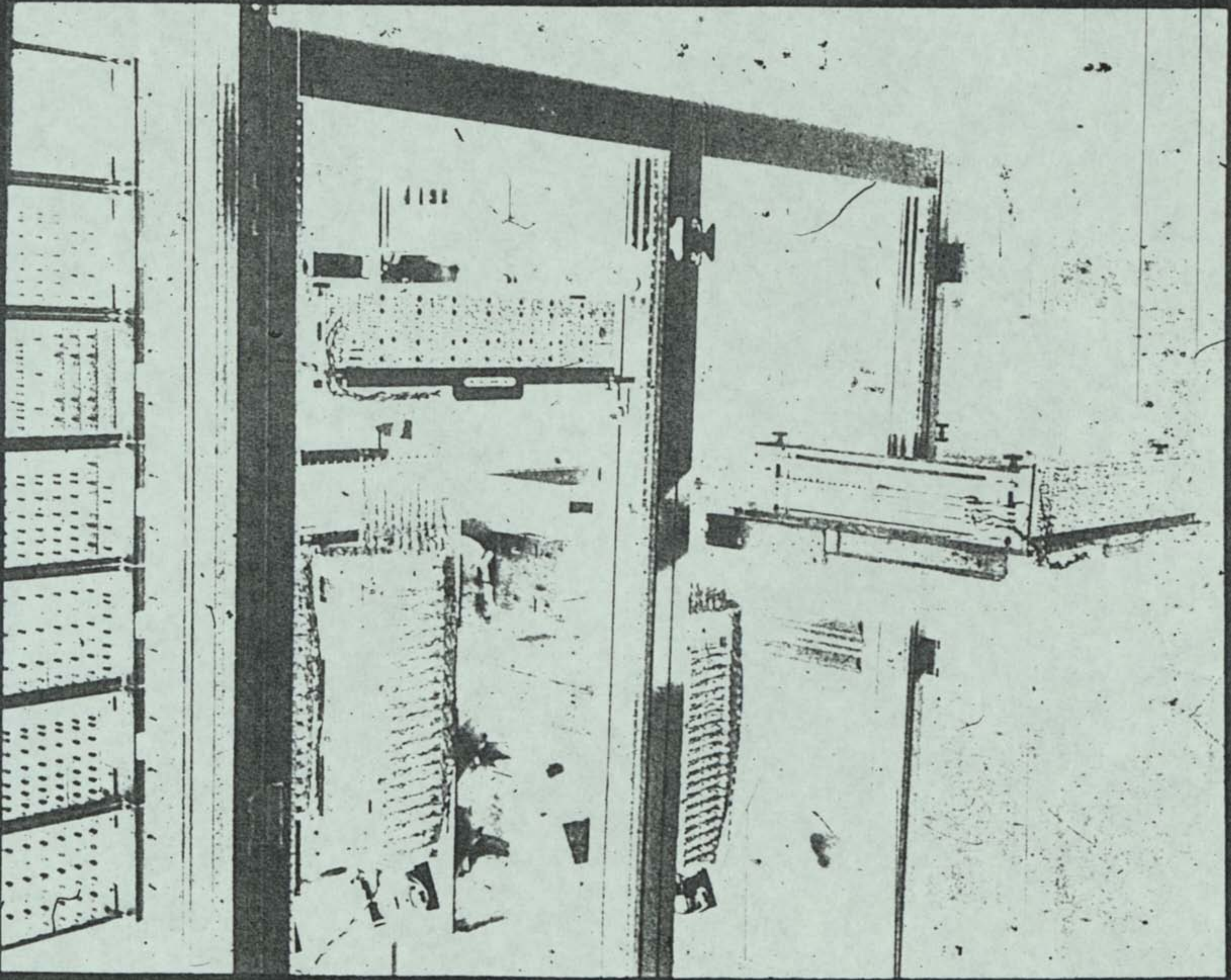


Figure 6. Front View of MIRE Section

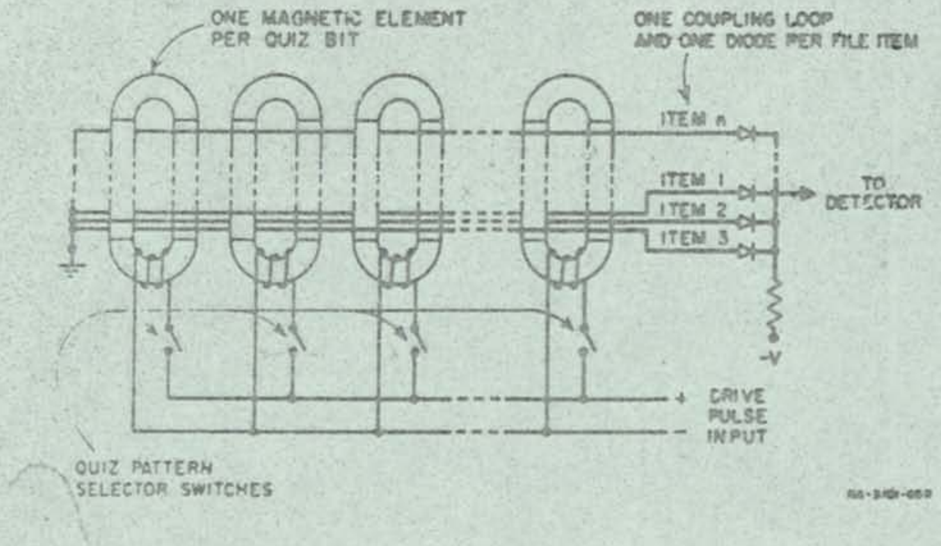


Figure 7. Core-Wiring Arrangement for MIRF Memory

4. Conclusions and Recommendations

A. Conclusions

In general it can be concluded that all the specifications set down for the experimental Multiple Instantaneous Response File are technically feasible. Interrogation of the magnetic storage units (MIRF's) and over-all control of the system can be accomplished with reliable circuits of modest complexity. Easy communication between a human operator and the machine has been demonstrated. The machine's response to a search question is essentially instantaneous in terms of human reaction time and the information content of the response is sufficient to allow the operator to modify his question and go directly to the document indexes of special interest. It can be concluded that certain automatic features such as the handling of synonomous input descriptors and the machine modification of the original search question can be achieved without undue complexity.

Several preliminary conclusions can be drawn from the machine performance observed during the check-out of the experimental model. It appears that the automatic substitution of the see-also reference of an input descriptor is a useful feature; but that the modification of the original search question by taking weighted descriptors from responding documents appears to be less useful. In many cases the combination of the asterisked descriptor with the original search terms gives a question with no responses. An examination of how to select descriptors from responding documents is needed in order to make this feature more useful. The capability of the experimental MIRF equipment to accept synonomous input descriptors also appears to be an important feature. Another interesting feature, which is not included in the present model, could be designed into future equipment. This would enable the machine to retrieve document indexes in which the form of the descriptor was somewhat different from that of the input (search) descriptor. The descriptors would be required to have the same root and meaning, but different word endings would be allowed.

Positive conclusions can be reached in regard to three of the most important properties of a file searching system of this type. First, it is concluded that searching by means of a superimposed code is feasible both from the standpoint of the circuitry required and the false response performance that is obtained. Second, it is concluded that storage of the document index information in wiring associated with arrays of cores that are physically separable is feasible. Experience with the experimental model shows that the arrays of cores can be separated, submodules of wired information can be changed, and the core arrays reassembled in a reasonably short time. More work on the mechanical design of the magnetic modules is needed, however, to permit easier and faster changing of the stored information. Third, it is concluded that a basic system building block has been established. The experimental model as delivered to the sponsor demonstrates that good performance can be obtained with a file of more than 1,000 document indexes. Experience with the experimental model indicates that expansion to five or six thousand document indexes can be achieved. It appears that with the present design the system building block should contain about 5,000 document indexes. It also appears that as many as ten such building blocks could be combined in a system whose over-all control is little more complex than that for a single building block. Therefore it is concluded that files of the order of 50,000 indexes could be built with no major changes in the basic concepts or circuits used in the experimental model.

B. Recommendations

A program should be initiated for applying the principles demonstrated in the experimental model to a substantially larger system. Special attention should be given to the method of realizing information storage in the form of wires associated with the magnetic cores. The method selected should lend itself to automation so that the complete process of preparing stored information can be machine controlled.

A study of the further application of the Multiple Instantaneous Response File concept should be initiated. In the field of document retrieval, the effect of removing limitations on word length and the number and nature of the descriptors used in document indexes could be investigated. The use of phrases of two or more words as search entities might also be examined. The application of the MIRF principles to code and language translation and other search type operations that require rapid feedback should be studied. More generalized search problems, such as pattern recognition, should be included. Applications that make use of the inherent speed of the search equipment should be investigated. (In the experimental model, the typewriter is by far the slowest part of the system). For example, the document information storage and the search facilities of the equipment could be shared by multiple user consoles. By time multiplexing techniques, many users could be given effectively private use of the machine. Applications in which the human operator is not a key figure should also be examined. The internal speed of the equipment makes feasible the use of digital computers or other computer-like machines as input-output devices.

New developments in superimposed coding should be investigated as a means of improving the efficiency of the searching operation. Recent work has shown that it is possible to design superimposed codes which can be decomposed to give the unique set of components of the superposition. Besides being uniquely decipherable, such superimposed codes also offer the possibility of retrievals with no false responses. With the new codes it may be possible to retain the advantages of superimposed coding (for example, freedom from the field indeterminacy problem) without suffering from the ordinary disadvantages of superimposed coding (for example, a finite false response probability). A simpler over-all design of the system may also be possible using the newer superimposed codes.

I. Introduction

A. Dates of Development Program

All work related to the design, construction, and checkout of the experimental model of the Multiple Instantaneous Response File described in this report was carried out during the period from 23 May 1962 to 23 July 1963.

B. Background

During the period from 1 January 1960 to 31 July 1961 a study* was made of the feasibility of constructing a data retrieval file of very large capacity in which all the data are interrogated simultaneously. One characteristic of the file was that it should be very large, containing the order of a million items of information. An item of information should consist of a single record, including an identification number, an abstract and appropriate logical specifications. Another important characteristic was that during a search, the entire file should be tested instantaneously (this requirement precluded the use of a serial search). The response time for all items responding to the search question should approach zero. The response should consist of the item identification number and index data in the form of an abstract.

During the study, general concepts for solving the search problem were developed. Codes and searching techniques suitable for such a file were examined and a simple and efficient testing algorithm for distinguishing between simultaneously responding items (multiple responses) was originated. Also several physical realizations suitable for such an index file were investigated. It was concluded from the study that the development of a data retrieval file having the stated specifications was feasible and that a magnetic implementation of the file with permanent storage of file information was attractive.

*This study was sponsored by Rome Air Development Center under Contract AF30(602)-2142. Refer to report RADC-TR-61-233, "Multiple Instantaneous Response File," by J. Goldberg et al, August 1961. ASTIA Report # AD 266 169.

During the final quarter of the study contract the preliminary design of an experimental model to demonstrate the essential features of a Multiple Instantaneous Response File was worked out. It was concluded that a model containing the order of 20,000 file items would be large enough to provide significant results and could be developed for a reasonable cost. These conclusions formed the basis for the specifications of the experimental model developed under the present contract.

C. Original Specifications of the Experimental Model

The experimental model described in the original proposal for research and the resulting contract has the following specifications:

(1) The size of the MIRF file shall be 1,000 items with design provisions for expansion to 5,000 items.

(2) Each item in the MIRF file shall be indexed by not more than 8 descriptors. A descriptor is an English word having 10 or fewer letters.

(3) Encoding of the items shall be accomplished by utilizing superimposed coding.

(4) The design of the superimposed code shall be adequate to represent a maximum of 3,000 descriptors.

(5) At least two descriptors shall be used in an interrogation.

(6) A dictionary file capable of holding 3,000 descriptors shall be provided as a part of the experimental model. Entries in the dictionary shall be one of the following types:

(a) Primary descriptor. A primary descriptor is one that can be used in indexing an item in the file. It can be used in composing the search quiz and can appear in the printed output of items that respond to a search.

(b) Synonym. A synonym is a descriptor whose meaning is synonymous with one of the primary descriptors. It can be used in composing the search quiz, but is not used to

2. Summary

The development of an experimental model of a Multiple Instantaneous Response File has been completed successfully. The experimental equipment contains more than 1,000 document indexes in its document file. The descriptors used in the document indexes are chosen from a dictionary of 1,000 words that is also part of the equipment. The logical, circuit, and mechanical designs of the model provide for a simple expansion to 5,000 document indexes and to a 3,000 word dictionary. Experience with the equipment during the checkout phase indicates that expansion to the design figures could be accomplished with little difficulty.

The specifications on a manually initiated search (listed in detail above) are satisfied by the experimental equipment. Excellent communication between a human operator and the machine has been experienced. The operator is required only to enter a search question as a group of English words by typing them on a conventional typewriter and to observe the results of the search typed out in simple format on the same typewriter. Translation of the English words into machine coding during the input phase and from coded machine responses into English words during the output phase are performed automatically. The requirements for modification of the manually inserted search question and the automatic initiation of new searches have also been satisfied. Experience with the machine shows that the see-also substitution is an important feature. Usually the additional see-also special search obtains pertinent additional responses relative to the original search and in some cases pertinent responses are obtained from the see-also search when there are no responses to the original search.

The experimental Multiple Instantaneous Response File is an all-solid state equipment. Transistor drive circuits capable of supplying two amperes of current to magnetic circuits, special discriminating amplifiers capable of operating reliably with a poor signal-to-noise ratio input signal, and transistor logic circuits tailored to the requirements of the system (high reliability, low cost, and moderate speed) were designed and constructed. About 300 current drive transistors, 2500 logic transistors, 2500 printed gate circuits (a group of 6 resistors, 2 capacitors and their interconnecting wiring on a passive substrate) and 5000 diodes are used in the system.

The documents that are stored in the experimental model were selected by the sponsor from ASTIA Technical Abstract Bulletins (TABs). The information concerning the selected documents was supplied by the sponsor in the form of a marked TAB abstract. Considerable effort was expended in reducing the raw data to a form that could be stored in the memory of the equipment. First, the relevant data from the TAB abstracts were reproduced in punched card form. Then computer programs were written for taking this raw data and preparing the data for each document to be stored in the machine (coded information for the accession number, the descriptors, and the search logic). The computing was carried out on the Control Data Corporation model 160-A. The resulting set of punched cards was used in special wiring arrangement that prepared the information that was stored in the machine. One punched card containing all the detailed information for one document was inserted into a punched card reader whose outputs were connected to a wiring jig. For each document a unique set of lights in the wiring jig was turned on and a wiring path was established. Be means of the computer handling of the raw data and the special attention given to the wiring arrangement, the errors in wiring the document information were reduced to the order of 1 error in more than 7,000 operations.