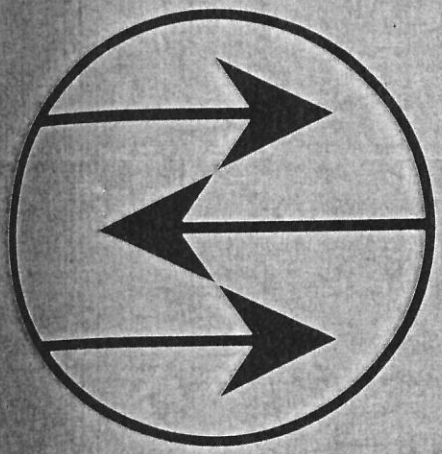


Row-by-Row Scanning Systems for IBM Punched Cards as Applied to Information Retrieval Problems

by H. P. Luhn

International Business Machines Corporation
Research Center
Yorktown Heights, New York

May 8, 1959

ROW-BY-ROW SCANNING SYSTEMS FOR IBM PUNCHED CARDS AS APPLIED TO INFORMATION RETRIEVAL PROBLEMS

H. P. Luhn
International Business Machines Corporation
Research Center
Yorktown Heights, New York

Research Report
RC 100
May 8, 1959

TABLE OF CONTENTS

TABLE OF CONTENTS (cont'd)

INTRODUCTION

In the art of mechanical information retrieval, one of the basic methods for identifying documents for selection consists of scanning through a set of document record cards. These record cards contain sets of data which characterize the contents of the individual documents. By comparing this data with a set of data characterizing an inquiry, documents may be identified which, by virtue of similarity of the respective characteristics, are liable to be relevant to the inquiry.

In using IBM punched cards as a medium for recording data for retrieval by machine, the system designer has to adjust his methods to the capabilities inherent to these devices. This is particularly true if such systems are to utilize comparatively simple equipment currently available. Such equipment lacks certain functions which are most desirable in processing information for retrieval. One of these is the ability of scanning information in serial fashion, column-by-column from left to right on the card, and of performing a reasonable degree of logic operations for the purpose of selection.

This requirement is based on the necessity of enumerating a varying number of characteristics and of having access to them for purposes of comparison no matter where they are located on a record card. However, the mode of operation of the machines in question demands that the columnar location of a desired entry be known beforehand and that, therefore, the entry be assignable to a fixed field.

In order not to forego the many advantages which punched card systems offer in processing information, a number of compromise methods have been introduced as a substitute for truly serial scanning of alphabetic or numeric punched card records. The one most widely advocated is that of superimposed coding. By this method information is no longer spelled out letter by letter or number by number. Instead, word or number notations are translated into a code of varying hole locations within a suitably large table or matrix. This table is assigned a fixed location on the card. The codes for the various characteristics to be recorded are then all punched into this same field. The result is a pattern of holes representative not only of the originally intended codes but also of spurious new code combinations, many of which may stand for characteristics which do not apply. As a consequence, when comparing for the presence of code combinations representative of the inquiry, the selection may contain false answers. This necessitates manual analysis of the answers to delete the wrong ones. It is true that by appropriate design these occurrences may be minimized, and that in certain applications some degree of such "noise" may be tolerated. But the system has other limitations. Once the entries are made into the field, there is no way of identifying thereafter what combination the code marks were intended to represent. Also there is no way of relating two or several codes to

express interdependence or combination of certain characteristics.

The row-by-row scanning system to be described here is another substitute for truly serial scanning. However, this system not only avoids the conditions just mentioned, but also offers certain advantages over column-by-column scanning.

PRINCIPLES OF ROW-BY-ROW SCANNING AND OPERATIONAL FEATURES OF THE EQUIPMENT INVOLVED

Row-by-row scanning is a method by which a plurality of items of information, recorded on cards, may be presented for analysis in serial order. As many as 12 rows of such information may be recorded on a card. Unlike the normal processing of punched cards the process of scanning does not depend on the recognition of individual numerals or letters for subsequent storage in a machine. It is necessary to ascertain only whether a given hole pattern is present or not in any of the rows.

Under these circumstances the recording of information in a single row across the card offers several advantages over conventional methods of recording. Instead of requiring 12 successive cycle points to analyze a pattern, i.e., holes representative of the recorded information, a single cycle point suffices to complete the analysis of a row. This represents not only a gain over a procedure by which a punched card might be read serially in column-by-column fashion, but also over the serial analysis of punched tapes, where each recorded character needs to be matched individually. Since as many as 9 letters or 14 numerals may be recorded in a single card row, this gain in terms of scanning time is significant. If the basis is taken that a non-optimum coding scheme will produce an average of 5 characters per row or 60 characters per card, then a card feeding rate of 450 cards per minute gives a scanning rate of 450 characters per second.

The row-by-row method of recording obviates the need of superimposed coding and overcomes the disadvantages previously enumerated. Alphabetic or numeric information may be spelled out character by character and may therefore be uniquely matched during the scanning process, thereby eliminating incidences of false selection. It is furthermore possible to express relationships amongst recorded items in many ways, to indicate ranges of values, alternative conditions and many other features. Also, the recorded information may always be recovered, which is not possible in superimposed coding schemes.

ROW-BY-ROW CODING SCHEMES

The utilization of the IBM punched card for row-by-row recording and processing demands the use of codes which are quite different in concept and function from those used in the past. Because of this departure, certain features of the presently available equipment are disadvantageous and impose limitations which have to be coped with. These limitations are, however, felt only in connection with the physical preparation of the record cards and their processing for purposes other than scanning. Otherwise, a considerable degree of freedom exists and to take optimum advantage of it calls for careful planning. The selection and design of coding schemes most appropriate for a given application is the most critical of these problems.

In order to assist the designer of row-by-row scanning systems, a review is here given of a variety of row-by-row coding schemes. Some typical examples will also be given, which favor economy in the assignment of logical decision elements that are available in the scanning equipment and of the manual effort required to program the equipment for a searching operation. Referring particularly to the IBM 101 Electronic Statistical Machine for row-by-row scanning, this economy is directed at the utilization of "recode selectors" and at the preparation of the control panel.

The coding schemes to be discussed involve machine codes, i.e., the hole combinations which are to be punched into the card to accommodate the internal control functions typical of a given piece of equipment.

Scanning Codes Having a Fixed Fraction of Elements (Holes)

This class of codes is applicable to those cases where a combination is to represent one definite meaning, such as a number, a letter, a whole word or other singular concept. When scanning, it is analyzed as a unit and the decision is either that it matches or does not. A partial match is of no significance.

There is a class of coding schemes in which the codes are permutations of a fixed number of elements, that is to say, the codes consist of a constant fraction of a given number of possible positions. For example, consider a set of codes of two holes in a field of five positions. In using this scheme, it suffices to test for the presence of the two holes only. A variable number element code, on the other hand, requires not only the testing for the presence of a given number of holes but also for the absence of holes in all other positions. It is readily seen that such variable element codes are wasteful in the utilization of decision equipment of the machine. Codes of the fixed number of elements type permit the use of rather simple matching techniques, such as optical black-out, no-pulse matching, and other complementary matching methods. Another important

feature of this type of code is its self-checking properties. These various advantages are fully exploited in the following schemes.

Number and Letter Coding

In number and letter coding the characters comprising a notation are spelled out individually and as many sub fields have to be provided as there are numbers or letters in a notation. The principle of serial scanning, row by row, demands that given numbers or letters of a notation be located at predetermined positions within each row. Offhand, this would indicate that each next word must be recorded in a new row. Since brevity is one of the desirable attributes of notations in information retrieval systems, the assignment of an individual row to each of the notations would result in very poor utilization of available card space. Therefore, a method is here proposed which will permit the recording of several notations in a single row. This method consists of dividing the row into fixed fields and to divide the vocabulary of the notations into as many groups as there are fields in a row. This grouping of the vocabulary may be accomplished in various ways as will be described in connection with the various examples. It is left to the randomness of occurrence in the various groups of the notation and to the randomness of usage of such words, when preparing a record, to obtain a reasonably even distribution of such notations amongst the fields of the card.

Coding of Number Combinations

In order to carry out the distribution of entries over various fields within a row, a number of schemes may be used. The objective here is to divide the whole set of numbers in such a way that each sub-set will contain a reasonably even number of assignments with respect to each other. If such assignment can be controlled at the time a dictionary is assembled, there is no particular difficulty of accomplishing this. A scheme which may be used here is that of assigning a numeric significance to the position of each field within a row. This scheme then creates as many sub-sets as the number of fields which may be used within a row. In assigning a numerical value to each of these fields, these values may be taken to represent the decimal digit or digits preceding those to be recorded in the respective fields. If, for instance, the values of 0, 1, 2, 3, etc. are assigned to fields across a row, the number 2463 would be punched as "463" in the third field, or the number 649 as "649" in the first field.

In order to insure randomness of distribution among the fields in a record, the assignment of numbers of this kind to dictionary entries may be made in rotation in accordance with these field numbers. Thus, numbers may be assigned to new terms in the order of accession to the dictionary as follows: 0001, 1001, 2001, 3001, 0002, 1002, 2002, 3002, ... Of course, the drawing of numbers from the whole set at random would have the same effect but would lack the convenient means of checking for completeness of a dictionary file.

There are cases where the assignment of numeric notations is outside the control of the system. There is the possibility that numbers have been assigned consecutively in ascending order, which property would bias the distribution of entries, if carried out according to the field assignments just described. This situation may be remedied to some degree by using the low-order digits of such numbers as a key for distribution. Thus, numbers may be distributed into two fields depending on whether they are odd or even. The distribution of numbers into four fields might be accomplished in accordance with the two last digits being odd or even. In this case the distribution would be as follows: even-even, even-odd, odd-even, odd-odd. These schemes have a disadvantage in not utilizing the code fields to their capacity; they reduce the capacity by 1/2 or 1/4 respectively.

Coding of Letter Combinations

The assignment of letter significance to the position of fields within a row may not be carried out too readily with the letters of the alphabet. However, there exists some alphabetic coding schemes which use only certain portions of the alphabet at given letter positions within a word. An example is a three-letter code word set, consisting of consonant-vowel-consonant combinations. By recording the two consonants of each such code word only, a field assignment may be made for the five possible intervening vowels, A, E, I, O, U, to fields 1, 2, 3, 4, 5, respectively. In accordance with this scheme the code word "BIT" for instance would be spelled as "BT" in the third field. In those cases where alphabetic notations are assigned rather than derived from the original spelling of words, they may be constructed in such a manner that the scheme of assigning letter significance to positions of fields within a row may be properly carried out.

In all other cases the distribution may be made in accordance with characteristics of the given notations themselves. The most practical method of doing this is to make the starting letter the key for distribution. For instance, the distribution of words into two fields may be determined by dividing the alphabet into two portions, as for instance, A-M and N-Z. For three fields the division may be A-G, H-O, P-Z. The appropriate division of the alphabet for these purposes does, of course, presuppose a fair knowledge of the use frequency of the words comprising the vocabulary.

Word Codes

The use of word codes has significant advantages with respect to the economy of recording as well as of the setting up of searching patterns. Their effectiveness is, however, more dependent on the characteristics of the record card than any of the previously discussed schemes.

Word codes consist of the assignment of an arbitrary designation of a minimum number of letters or numbers to a given primary expression such as a word or words, a formula, a pictorial, etc. Rather than translating these letters or numbers individually into a sequence of machine codes for recording, the word code scheme features translation directly into a machine code of appropriate dimensions.

The difference between these two ways of machine coding is best illustrated by way of example. The numbers 0-99 may be coded by means of a 2/5 code for each of the two digits, requiring a field of 5 + 5 = 10 positions. If these ten positions were made the common basis for a code using four marks, i.e., the 4/10 code, 210 combinations may be obtained instead of the 100 of the decimal notation. These 210 combinations may be made the basis of a dictionary and a word be assigned to each of these combinations.

In this connection it becomes necessary to identify these code combinations by some convention. Because of the binary character of such codes, they may be written as a sequence of ones and zeros, the ones standing for holes and the zeros for no holes. For the purpose of reference these sequences may be ordered by the value they represent if read as binary numbers. Such notations, when extended to large code fields would, however, become quite awkward.

An alternative means of identification, which overcomes the above-mentioned drawbacks and also introduces other attractive features, is the use of octal numbers applied to the binary form of the code. In this case, every three binary digits, reading from right to left, represent an octal digit which is in turn represented by a decimal numeral, from 0 to 7.

Still another method of expressing a row pattern of holes or marks consists of assigning consecutive numbers to the consecutive positions of the field. The pattern is then specified by stating numerically the position of each hole punched, or of each mark made. Thus the pattern whose binary notation is 0010001011 is identified in this present scheme as: 3, 7, 9, 10 or 03070910. These notations may be ordered in accordance with their decimal value for purposes of reference.

An additional valuable feature of this method becomes apparent in considering the problem of randomly spreading entries over several fields in a row, in the case of coding schemes in which a particular hole pattern may be punched in one of several fields, and is assigned a different significance in each field. When the above method of row pattern identification is used, this field assignment may be notated in a very simple and convenient manner. By consecutively numbering all positions within a card row, as is virtually done on the IBM card through column numbers, a given hole combination and its placement may conveniently be identified directly by the column numbers of the field in which the combination is to be located. Thus, the combination 03070910 is notated as 23272930 when

it is to be recorded in the third ten-position field (columns 21-30).

When entries are assigned to a dictionary based on this system, they should be rotated amongst the various field locations in order to obtain a reasonable degree of random distribution, since the capacity of the card to record all entries pertinent to an item is maximized if each field is used equally.

It is apparent that not only is the size of the possible dictionary increased by a significant factor but also the plugging of the combinations on the control panel is substantially simplified because of the direct identification of the hubs involved.

The use of word codes reduces the number of active code elements required for recording. Therefore, the number of recode selectors required for setting up a question on the control panel is substantially reduced over those required in letter coding. This saving reduces the time required for setting up a question on the control panel.

Coding of Various Classes of Notations and of Degrees of Relationships

Depending on the objectives of a retrieval system, the mere enumeration of code words may not suffice. It might be necessary to distinguish between different classes of words and to indicate certain degrees of relationship among them.

(a) Grouping of Code Words

A first degree of discrimination may be indicated by a division mark, comparable to a period signifying the end of a sentence. A special mark may be punched alongside the row in which the enumeration of words of a group starts and terminates. This control code may be sensed when scanning the record and utilized to register the findings up to the terminating row.

This grouping device permits, for instance, selection of a record if two given words appear in the same group but to reject it if they appear each in a different group.

On the IBM 101 Electronic Statistical Machine, cards are fed so that the nine row is the first to be read. This does not necessarily imply that the recording on the record card needs to be done in this order, that is, from the bottom upwards. However, in the case where rows are combined into groups, attention does need to be paid to the proper interpretation of the embracing group control holes.

(b) Class Designations

A next degree of discrimination is the introduction of different classes of words. This may be accomplished either:

(1) by assigning different class identification codes to different rows

(2) by assigning different classes to different fields in a row

(3) by attaching appropriate prefixes or suffixes to the code words themselves.

In the first case, a statement would consist of so many rows for each of the various classes of code words, terminated by a division mark alongside the last one of the rows. In this way, for instance, certain materials may be associated with their properties and their applications, substantially by the types of codes previously enumerated.

The second case demands more specialized coding schemes than the ones discussed so far, in that statements made up of different classes of code words are to be represented in the same row. This means that field location may no longer be utilized as a means of extending the capacity of the record card or the size of the dictionary. The effectiveness of this arrangement depends entirely on the feasibility of reducing statements to a standard format of a fixed number of terms. There is, of course, the possibility of using a variety of formats, each applicable to a given type of statement and each identified, as a row, by a different pattern code.

The third case preserves the freedom of the basic coding schemes but burdens the code words with their individual class index. While the addition of this information reduces the size of the dictionary that may be accommodated, under comparable conditions it offers many advantages in that searches may be made including or excluding the class index. In addition, it will be possible to indicate multiple class assignments by writing the class index in the form of a "range code", i.e., by individual marks for each class within the associated special code field. Thus, for instance, if a given code word represents a chemical compound, which under given conditions might either be an intermediate or an end product of a process, a mark is recorded for each of these cases. The compound would then be selected in any search which calls for it, regardless of whether the search specifies one case or the other or both cases or none.

Tabular Scanning Codes

The principle of this type code differs from the ones previously described in that a separate meaning is assigned to each individual position in a coding field. A mark in a given position denotes that the corresponding meaning applies. This original method of recording, employed by the inventor of punched cards for the tabulation of statistical information, has never lost its usefulness. Its application to searching systems is of great utility because of the economy in utilization of recording space. Through the use of row-by-row recording this utility may be significantly extended.

While the recognition of the previously discussed codes is based on a perfect match of all of the code elements given by the question, a search addressed to tabular codes requires only that the given elements are included in the patterns being searched, all other elements being disregarded.

Simple Tabular Codes

In applying tabular coding to single rows of a record card, the information that is to be represented must be organized so as to conform to the physical limits imposed by a row. On the other hand, there is considerable freedom with respect to the number of different styles of rows that may be constructed and the manner in which they may be used. The various styles would be recorded in a manual, furnishing the key to the meaning of the various row positions for each style. Each of these different styles would be identified by its own code designation, punched alongside the row. When searching,this code designation would become part of the question.

One style of row may serve for identifying any one or several of a finite set of characteristics pertaining to an object identified in a preceding row. In this case the two rows would be tied together by appropriate division marks as previously discussed. If space permits, the identification of the object and the enumeration of its properties may be combined in a single row. There is no reason why the two types of scanning codes may not be used in the same row.

Scalar and Range Codes

This type of coding may be used where it is desired to represent values or magnitudes. In this case, a row or a part thereof would become the equivalent of a scale with the scale points represented by the possible hole positions. When recording a given value, a hole is punched at the appropriate point of this scale. In order to bring this point into range when searching, additional holes may be punched to the left and right of this hole, the number of additional holes depending on the dimensions of the scale and the desired range.

Several points may be recorded on such scales as for instance maxima, minima, or other critical values of a continuous function, or the high-low range of a variable.

A similar principle may be applied where it is desired to determine whether a given quantity is equal to or less than quantities being scanned. In this case the actual quantity is punched as well as all the quantities below it. The same principle may be employed to express the condition of equal or larger than a given quantity.

The above schemes may also be used for the charting of values which are a function of two or three variables. This is accomplished by assigning a row to each of the coordinates x and y or x , y and z and by punching the respective scale points in each. The several rows would be tied together as a group as previously explained.

There are many other schemes which may be evolved with the aid of tabular scanning codes and it is a matter of ingenuity to design such schemes so that they will be optimum for each particular situation.

Superimposed Codes

While it has been argued that the row-by-row method of recording resolves the problem which brought superimposed coding into being, there might nevertheless be situations where superimposed coding offers advantages. Whenever that is the case, certain features of the row-by-row recording method might be combined with those of the superimposed recording method.

Under certain conditions, the recording capacity of a row might be adequate to serve as a superimposed coding field, equivalent to an 8 x 8, 7 x 10, 6 x 12 or similar size matrix. A card would accommodate 12 such fields. While it is true that the use of a smaller field affects the efficiency of the method this may be offset by the possibility of using more than one such field. By recording the optimum minimum number of entries in a first row and by doing the same in subsequent rows until all entries have been accommodated, any desired degree of reliability may be achieved. The affected rows would be tied together as a group and therefore it does not matter in which particular one of the rows an entry has been recorded or found.

Inasmuch as the IBM 101 Electronic Statistical Machine with a row-by-row scanning attachment is capable of operating in the conventional mode as well as the row-by-row mode there is the possibility of processing cards containing mixtures of codes for these two modes. Therefore a superimposed code field of the conventional 2-dimensional form may occupy one portion of the card while row-by-row codes may occupy another portion.

Conventional IBM Alphanumeric Code

The utility of the conventional IBM card code in row-by-row scanning systems is limited to those functions which are to be performed by other card processing equipment. There is usually the need for identifying the record cards by a document number. This number should normally be punched in the right-most portion of the card, a location not dictated by the scanning system itself but desirable from the point of certain processing equipment that might be employed in the automatic preparation of the scanning code portion of the card. The document number coding may play a part in the searching process in that it may be read and printed out to identify the printed tally of matches discovered on the associated card.

There is, of course, no limit as to other additional identificatory information that may be recorded on the card in the conventional manner, except that each additional column appropriated for this purpose reduces the space available for recording information in scanning code.

TYPICAL SCANNING CODES AND THEIR ASSEMBLY INTO ROWS

The introduction of a card row as a scannable unit of recorded information introduces the problem of how to make most effective use of its fixed capacity. The standard type of card handling and processing equipment is not designed to operate on horizontally disposed code patterns. Since it is nevertheless desired to use such equipment in the processes of card preparation, file maintenance, card identification, and listing, it is necessary to assign an appropriate portion of the total card capacity to this function. Also, as will be seen later, certain control functions may have to be associated with each row, where such functions will have to be represented by some special code. It is assumed that both of these functions will require the card space equivalent of at least 8 card columns. This requirement fixes the maximum length of row available for row-wise coding to 72 positions. Furthermore, it might be desirable, if not necessary, to differentiate between different types of styles of row coding, calling for additional space for purposes of identification. Under these circumstances the practical length of a row may be assumed to be 64 positions. Therefore, 64 and 72 positions respectively have been given special consideration in the discussions which follow.

Measured in terms of permutations of binary elements, a field of 64 or 72 positions will produce a staggeringly large number of code combinations. Even considering the requirement that a set of scanning codes must have a fixed number of active elements (more specifically, a fixed number of holes), the number of potential combinations is still way beyond anything that would be needed in practice. But what these applications demand is abundance, not of how many things can each be represented singly in different rows, but of how many different things can concurrently be represented in a single row. The problem therefore is to develop codes which optimize the relationship between a given number of things and the highest number of different things that may be recorded simultaneously in a row. When considering the problem of processing such representations, this optimization might also include the objective of minimizing the number of logical decision elements required in the scanning operations.

The number of positions within a row, even though limited, permit a great deal of freedom, within adequate capacity, for the average type of retrieval problem. This freedom of tailoring code arrangements to suit various particular requirements is considered to be a feature of the system. The fact that different styles of coding arrangements may follow each other eliminates the rather substantial limitations imposed by the rigidity of standard coding schemes.

Typical Codes

Rather than giving examples as to how the coding of a row may be organized for a particular application, a comprehensive table has been compiled, listing a variety of fixed number element scanning codes and their characteristics. Depending on their particular properties, these codes may be assigned to letters, numerals or whole words. The codes have been selected to favor rows of 64 and 72 positions respectively. (See Table I)

Because of the inverse relationship between the size of a given dictionary and the number of items from this dictionary which may be enumerated in a single row, this dependency has been represented in tabular form for some of the codes. This arrangement should assist the systems designer in choosing a suitable combination of these two variables. The table also shows a variety of assignments of alphabets, numerals and special characters. The proper choice from amongst these depends on given conditions as well as design objectives. It may be advantageous in certain cases to reflect the properties of certain compact codes on the selection of code words so as to take advantage of the brevity of the code involved, for optimum utilization of a row.

The table also gives for each code the number of digits of octal or decimal numbers required for systematic identification. The system of identification by octal numbers consists of dividing the binary representation of a combination into octal digits, starting from the right. An octal digit comprises 3 binary digits of the values 1, 2, 4, reading from right to left. By writing each octal digit by means of a decimal numeral from 0 to 7 a convenient notation is derived that can serve for identifying each of the various code combinations.

An alternative identification consists of numbering each binary digit from left to right consecutively by decimal numbers 1, 2, 3, etc. Since the scale may go beyond 9, a two digit number is used. The number of octal digits grows with the number of positions of the code while the number of decimal digits grows with the number of holes contained in these positions.

The assignment of alphabetic characters and of numerals to several of the codes is illustrated in a number of tables. Where feasible, the assignments have been so chosen that sequence amongst the numerals and alphabetic characters is expressed by the ascending values of the codes if read as binary or octal numbers. This feature has useful applications in connection with the automatic preparation of the record cards. The feature has also applications in the scanning process in that it permits selection in accordance with certain sets of numbers or letters. (See Table II)

TABLE I: CHARACTERISTICS OF FIXED NUMBER ELEMENT CODES

Fraction: Number of Elements (Marks, Holes) per Number of Positions (Field Size)		Number of Combinations Assignable to: Letters	Numerals	Words etc.	Notation for Positional Identification by: Octal Numbers	Decimal Numbers	Typical Assignments
1	2	2	2	2	1	2	Binary Digit 0-1
1	5	5	5	5	2	2	Quinary Digits 0-4 or 5 Vowels
1	10	10	10	10	4	2	10 Numerals 0-9 (IBM Card Code)
2	4	6	6	6	2	4	5 Vowels and 1 Special Character
2	5	10	10	10	2	4	10 Numerals 0-9
2	6	15	-	15	2	4	15 Consonants
2	7	21	-	21	3	4	21 Consonants or 16 Consonants and 5 Vowels
2	8	26	-	28	3	4	Full Alphabet and 2 Special Characters
2	9	26 + 10		36	3	4	Alphabet and Numerals
2	10	26 + 10		45	4	4	Alphabet, Numerals and 9 Special Characters
2	11	26 + 10		55	4	4	" " " 19 " "
2	12	26 + 10		66	4	4	" " " 30 " " or
2	12	26					Alphabet in IBM Card Code
2	14			91	5	4	
2	16			120	6	4	
2	18			153	6	4	
2	20			190	7	4	
2	21			210	7	4	
2	24			276	8	4	
2	28			378	9	4	
2	32			496	11	4	Vocabularies, Dictionaries or Indexes
2	36			630	12	4	
2	42			861		4	
2	48			1128		4	
2	54			1431		4	
2	60			1770		4	
2	64			2016		4	
2	72			2556		4	
3	6	20	-	20	2	6	20 Consonants
3	7	25 + 10		35	3	6	20 Consonants, 5 Vowels and Numerals
3	8	26 + 10		56	3	6	Alphabet, Numerals and 20 Special Characters
3	9			84	3	6	
3	10			120	4	6	Vocabularies, Dictionaries or Indexes
3	11			165	4	6	
3	12			220	4	6	
4	8	26 + 10		70	3	8	Alphabet, Numerals and 34 Special Characters
4	9	26 + 10		126	3	8	" " " 90 " "

TABLE I: CHARACTERISTICS OF FIXED NUMBER ELEMENT CODES (cont'd)

Fraction: Number of Elements (Marks, Holes)	Fraction: Number of Positions (Field Size)	Number of Combinations Assignable to Words, Etc. Single Field	Number of Decimal Digits Required For Identification Of Code Combinations: Octal Numbers or Decimal Numbers		Total Number of Assignable Combinations — Multiple Fields Per Row: Total Positions	Number of Fields	Total Positions	Number of Fields	Total Positions	Number of Fields	Total Positions	Number of Fields
		1			2		3		4		5	
4	10	210	4	8	20	420	30	630	40	840	50	1050
4	11	330	4	8	22	660	33	990	44	1320	55	1650
4	12	495	4	8	24	990	36	1485	48	1980	60	2475
4	14	1001	5	8	28	2002	42	3003	56	4004	70	5005
4	16	1820	6	8	32	3640	48	5460	64	7280		
4	18	3060	6	8	36	6120	54	9180	72	12240		
4	20	4845	7	8	40	9690	60	14535				
4	21	5985	7	8	42	11970	63	17955				
4	24	10626	8	8	48	21252	72	31878				
4	28	20475	10	8	56	40950						
4	32	35960	11	8	64	71920						
4	36	58905	12	8	72	117810						
4	42	111930		8								
4	48	194580		8								
4	54	316251		8								
4	60	487635		8								
4	64	635376		8								
4	72	1028790		8								

Fraction		6		7	
4	10	60	1260	70	1470
4	11	66	1980		
4	12	72	2970		

		1			2		3		4		5	
5	10	252	4	10	20	504	30	756	40	1008	50	1260
5	11	462	4	10	22	924	33	1386	44	1848	55	2310
6	12	924	4	12	24	1848	36	2772	48	3696	60	4620
6	14	3003	5	12	28	6006	42	9009	56	12012	70	15015
6	16	8008	6	12	32	16016	48	24024	64	32032		
6	18	18564	6	12	36	37128	54	55692	72	74256		
6	20	38760	7	12	40	77520	60	116280				
6	21	54264	7	12	42	108528	63	162792				
6	24	134596	8	12	48	269192	72	403788				
6	28	376740	10	12	56	753480						
6	32	906192	11	12	64	1812384						
6	36	1947792	12	12	72	3895584						

Fraction		6		7	
5	10	60	1512	70	1764
5	11	66	2772		
6	12	72	5544		

		1			2		3		4		5	
7	14	3432	5		28	6864	42	10296	56	13728	70	17160
8	16	12870	6		32	25740	48	38610	64	51480		
9	18	48620	6		36	97240	54	145860	72	194480		
10	21	352716	7		42	705432	63	1058148				
12	24	2704156	8		48	5408312	72	8112468				

TABLE II: ASSIGNMENTS OF FIXED NUMBER ELEMENT CODES

#	3/7 Code	Octal #	Assignments
1	X X X	7	0 — also letter O
2	X X X	13	9
3	X X X	15	8
4	X X X	16	7 These are the
5	X X X	23	6 complements of
6	X X X	25	5 the conventional
7	X X X	26	4 2/5 code
8	X X X	31	3
9	X X X	32	2
10	X X X	34	1 — also letter I
11	X X X	43	A
12	X X X	45	B
13	X X X	46	C
14	X X X	51	D
15	X X X	52	E
16	X X X	54	F
17	X X X	61	G
18	X X X	62	H
19	X X X	64	J
20	X X X	70	K
21	X X X	103	L
22	X X X	105	M
23	X X X	106	N
24	X X X	111	P
25	X X X	112	Q
26	X X X	114	R
27	X X X	121	S
28	X X X	122	T
29	X X X	124	U
30	X X X	130	V
31	X X X	141	W
32	X X X	142	X
33	X X X	144	Y
34	X X X	150	Z
35	X X X	160	Space/Special Character

#	2/8 Code	Octal #	Assignments	
1	X X	3	A	1
2	X X	5	B	2
3	X X	6	C	3 Codes for
4	X X	11	D	4 numerals
5	X X	12	E	5 identical
6	X X	14	F	6 with
7	X X	21	G	7 conventional
8	X X	22	H	8 2/5 code
9	X X	24	I	9
10	X X	30	J	0
11	X X	41	K	
12	X X	42	L	
13	X X	44	M	
14	X X	50	N	
15	X X	60	O	
16	X X	101	P	
17	X X	102	Q	
18	X X	104	R	Special
19	X X	110	S	Characters
20	X X	120	T	
21	X X	140	U	
22	X X	201	V	
23	X X	202	W	
24	X X	204	X	
25	X X	210	Y	
26	X X	220	Z	
27	X X	240	Space	Space
28	X X	300	Special	Special

Numbers or special characters to be preceded by "special" code combination #28; letters following numbers or special characters to be preceded by space code #27.

#	2/7 Code	Assignments: All Letters	Assignments: Consonants Only
1	X X	A	B
2	X X	B	C
3	X X	C X	D
4	X X	D	F
5	X X	E	G
6	X X	F	H
7	X X	G	J
8	X X	H	K
9	X X	I	L
10	X X	K Q	M
11	X X	L	N
12	X X	M	P
13	X X	N	Q
14	X X	O	R
15	X X	P	S
16	X X	R	T
17	X X	S Z	V
18	X X	T	W
19	X X	U	X
20	X X	V W	Y
21	X X	Y J	Z

#	3/6 Code	Assignment Consonants Only
1	X X X	B
2	X X X	C
3	X X X	D
4	X X X	F
5	X X X	G
6	X X X	H
7	X X X	J
8	X X X	K Q
9	X X X	L
10	X X X	M
11	X X X	N
12	X X X	P
13	X X X	R
14	X X X	S
15	X X X	T
16	X X X	V
17	X X X	W
18	X X X	X
19	X X X	Y
20	X X X	Z

#	2/5 Code	Assignment
1	X X	1
2	X X	2
3	X X	3
4	X X	4
5	X X	5
6	X X	6
7	X X	7
8	X X	8
9	X X	9
10	X X	0

#	2/4 Code	Assignments	
1	X X	A	1
2	X X	E	2
3	X X	I	3
4	X X	O	4
5	X X	U	5
6	X X	Q	6 (0)

Typical Row Assemblies

The technique of assembling appropriate styles of codes to form a row of the record card is shown by a number of examples assembled into a table. These examples deal with the problem of accommodating the maximum number of entries from a given size vocabulary. The numbers or words are spelled out digit by digit or letter by letter with the use of various of the codes shown in the table of fixed number element scanning codes. The samples are also illustrative of the many ways in which one may utilize the capacity available within a row. (See Table III)

Identification of Different Row Patterns

As was pointed out earlier, an important feature of row-by-row scanning is that rows of different patterns may be recorded in any desired sequence. It is also not necessary to assign special locations on the card to rows of specific types of information, unless such positioning is to become a means of differentiation by itself.

Generally the different types of rows are identified by a special set of code marks appended to the row. The extent of such a special code field depends on the total number of patterns to be used in a system. Fixed number element codes should be used for this purpose. For instance a 2/4 code would allow for six different types of patterns. Care must be taken that these row identification codes are in a fixed location for all patterns. This means that the longest of all the patterns determines the location. The row identification code must be included in the search pattern when scanning.

Group Identification

It may be desired to organize the information to be recorded on a card into groups to express varying degrees of relatedness amongst statements. Such grouping may be equivalent to the relationships between words, phrases, sentences, paragraphs, etc. in normal writing. Scanning devices like the Row-by-Row 101 Electronic Statistical Machine have the functional ability to respond to special operational signals for treating information in accordance with such grouping. These signals may be caused by two special holes to be punched alongside the several rows which are to be treated as a group. The first hole would mark the start of a group and the second hole the end of the group. Again these two holes must appear in a fixed location within all the rows.

This method may be extended to handle sub-groups within groups. In these cases, additional control positions must be provided.

TABLE III: TYPICAL ROW LAYOUTS FOR CHARACTER BY CHARACTER RECORDING OF NUMBERS OR WORDS

#	Size of Dictionary	Digits or Letters per Word Field	Word Fields per Row	Max. Total of Words per Card	Marks or Holes per Word Field	Total Positions within Row	Coding Schemes Within a Row
							Numeric Entries
1	300	2	3	36	2	60	0-99 | 1/10 | 1/10 ; 100-199 | 1/10 | 1/10 ; 200-299 | 1/10 | 1/10 ; 60
2	1800	3	3	36	4	72	0-599 | 2/4 | 1/10 | 1/10 ; 600-1199 | 2/4 | 1/10 | 1/10 ; 1200-1799 | 2/4 | 1/10 | 1/10 ; 72
3	4000	4	2	24	4	64	0-1999 | 1/2 | 1/10 | 1/10 | 1/10 ; 2000-3999 | 1/2 | 1/10 | 1/10 | 1/10 ; 64
4	12000	4	2	24	4	72	0-5999 | 1/6 | 1/10 | 1/10 | 1/10 ; 6000-11999 | 1/6 | 1/10 | 1/10 | 1/10 ; 72
5	700	2	7	84	4	70	0-99 | 2/5 | 2/5 ; 100-199 | 2/5 | 2/5 ; 200-299 | 2/5 | 2/5 ; 300-399 | 2/5 | 2/5 ; 400-499 | 2/5 | 2/5 ; 500-599 | 2/5 | 2/5 ; 600-699 | 2/5 | 2/5 ; 70
6	1200	3	6	72	5	72	0-199 | 1/2 | 2/5 | 2/5 ; 200-399 | 1/2 | 2/5 | 2/5 ; 400-599 | 1/2 | 2/5 | 2/5 ; 600-799 | 1/2 | 2/5 | 2/5 ; 800-999 | 1/2 | 2/5 | 2/5 ; 1000-1199 | 1/2 | 2/5 | 2/5 ; 72
7	3000	3	5	60	6	70	0-599 | 2/4 | 2/5 | 2/5 ; 600-1199 | 2/4 | 2/5 | 2/5 ; 1200-1799 | 2/4 | 2/5 | 2/5 ; 1800-2399 | 2/4 | 2/5 | 2/5 ; 2400-2999 | 2/4 | 2/5 | 2/5 ; 70
8	4000	3	4	48	6	60	0-999 | 2/5 | 2/5 | 2/5 ; 1000-1999 | 2/5 | 2/5 | 2/5 ; 2000-2999 | 2/5 | 2/5 | 2/5 ; 3000-3999 | 2/5 | 2/5 | 2/5 ; 60
9	12000	4	4	48	7	72	0-2999 | 1/3 | 2/5 | 2/5 | 2/5 ; 3000-5999 | 1/3 | 2/5 | 2/5 | 2/5 ; 6000-8999 | 1/3 | 2/5 | 2/5 | 2/5 ; 9000-11999 | 1/3 | 2/5 | 2/5 | 2/5 ; 72
10	30000	4	3	36	8	60	0-9999 | 2/5 | 2/5 | 2/5 | 2/5 ; 10,000-19,999 | 2/5 | 2/5 | 2/5 | 2/5 ; 20,000-29,999 | 2/5 | 2/5 | 2/5 | 2/5 ; 60
11	180000	5	3	36	10	72	0-59,999 | 2/4 | 2/5 | 2/5 | 2/5 | 2/5 ; 60,000-119,999 | 2/4 | 2/5 | 2/5 | 2/5 | 2/5 ; 120,000-179,999 | 2/4 | 2/5 | 2/5 | 2/5 | 2/5 ; 72
							Alphabetic Entries
12	2000	3	5	60	6	60	B-Z A B-Z | 3/6 | 3/6 ; B-Z E B-Z | 3/6 | 3/6 ; B-Z I B-Z | 3/6 | 3/6 ; B-Z O B-Z | 3/6 | 3/6 ; B-Z U B-Z | 3/6 | 3/6 ; 60
	CVC Words						
13	17576	3	3	36	6	72	A-G | 2/8 ; A-Z | 2/8 ; A-Z | 2/8 ; H-O | 2/8 ; A-Z | 2/8 ; A-Z | 2/8 ; P-Z | 2/8 ; A-Z | 2/8 ; A-Z | 2/8 ; 72
	3 Letter Words						
14	11025	4	3	36	6	72	B-H | 2/7 | A-U | 1/5 ; B-Z | 2/7 | A-U | 1/5 ; J-R | 2/7 | A-U | 1/5 ; B-Z | 2/7 | A-U | 1/5 ; S-Z | 2/7 | A-U | 1/5 ; B-Z | 2/7 | A-U | 1/5 ; 72
	CVCV Words						
15	456946	4	2	24	8	64	A-L | 2/8 ; A-Z | 2/8 ; A-Z | 2/8 ; A-Z | 2/8 ; M-Z | 2/8 ; A-Z | 2/8 ; A-Z | 2/8 ; A-Z | 2/8 ; 64
	4 Letter Words						

C = Consonants Only

V = Vowels Only

Grouping of Words to Express Relationships

Word Pairs

Individual Pairs

The grouping of two words into a pair may be recorded in a pair of rows as follows. The word having the lower code number is recorded in the appropriate field of the first of the chosen pair of rows. The second word is recorded in the same field of the second row. The true field number of the second word is expressed as a prefix, the code for which is split in half; the first half is recorded as part of the first word, the second half as part of the second word. Example:

First word, octal notation, 6/12 code, field 4 . . . . . 4/4313
Second word, octal notation, 6/12 code, field 3 . . . . . 3/5270
Prefix for field 3, in 2/4 code . . . . . |-|x‖x|-|

	Prefix		Field number four											Group control		
First row		x	x				x	x			x		x	x	x	
Second row	x		x		x		x		x	x	x				x	

Typical schemes are as follows:

Size of Dictionary	Codes		Fields per row	Positions		Maximum number of pairs per card
	Word	Prefix		per field	per row	
1512	5/10	2/4	6	12	72	36
4620	6/12	2/4	5	14	70	30
17472	5/16	1/4	4	18	72	24
45760	7/16	1/4	4	18	72	24

Pairs by Node and Branches

Presentation in single row by mixed systems:

In the mixed system, the node is recorded in a discrete code of the type discussed above, in a specified field; the branches are recorded by superimposed coding.

The table shown below gives the characteristics of two examples of this mixed type of coding.

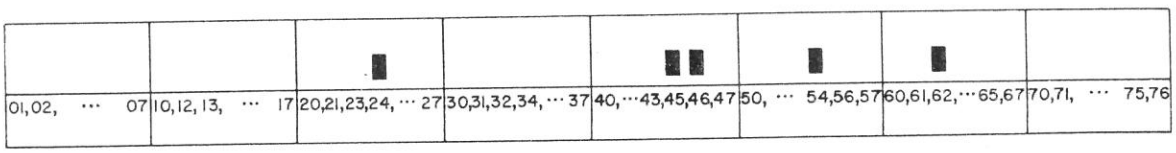
TABLE: NODE AND BRANCH CODING

Size of dictionary	8008	12,870
Codes:		
Node	6/16	8/16
Branch	five-digit octal numbers, no adjoining identical numerals	
Number of row positions:		
Discrete coding	16	
Superimposed coding	8 x 8 - 8 = 56	
Total	72	
Maximum number of nodes per card	12	
Number of holes per branch	5	
Maximum number of branches per field per row*	2 - 7	
Maximum number of pairs per card	24 - 72	

* See probability table below.

Superimposed code notations are derived by the randomizing square method. (For details, see H. P. Luhn: Superimposed Coding with the Aid of Randomizing Squares for Use in Mechanical Information Searching Systems. IBM Product Development Laboratory, Poughkeepsie, New York.)

The 56 superimposed code field positions are numbered as follows:



The recording is done by means of chain and ring spelling. Example, as shown above: 54624 = 54, 46, 62, 24, 45.

The probability of correct selections may be improved by spreading the entries over several fields in successive rows and by tying the rows together by the grouping device. Since there would be fewer entries in each of the individual fields, the probability of false selections is decreased.

Presentation by Discrete Codes and Index Field

Another method of expressing the relationship between a node and its branches consists of recording, by the type of discrete coding discussed above, the word codes of the branches as well as of the node without distinction, in the appropriate fields of a row or rows as a group, as previously described for indexing group relationships. In order to establish which of the words of a group is the node, a special index field is appended to each row as part of the row coding scheme. This index field serves to identify any one of the word fields within a row, if so desired. For instance a 2/4 code could serve as a label for one of six particular field locations in the associated row. In order to identify the node among the words of a group, it is only necessary to punch the field location number of the node word into the index field of the row in which this word happens to be recorded. When searching, this index code would have to be included in the question in order to detect the node-branch relationship of a pair of words.

Word Groups

The recording of word groupings may be performed as follows:

(a) Discrete Coding:

Recording of the words of a group in as many rows as required by use of group signal.

(b) Superimposed Coding:

Use of 8 by 8 (64) field for octal notations
Use of 8 by 8 (64) field for number or letter spellings
Use of 6 by 12 (72) field for number or letter spellings

A table of probabilities of correct selections from a superimposed code field is shown below.

PROBABILITIES OF CORRECT MATCHES FOR SUPERIMPOSED CODING IN A 64 POSITION FIELD *

Number of Words Punched In Field	Number of Holes Per Word Optimum for the System	Number of Potential Matches Announced by Machine: 1 — Probability That The Match Is Correct	2 — Probability of Correct Matches: at least 1	2 — Probability of Correct Matches: at least 2	3 — Probability of Correct Matches: at least 1	3 — Probability of Correct Matches: at least 2	3 — Probability of Correct Matches: 3	4 — Probability of Correct Matches: at least 1	4 — Probability of Correct Matches: at least 2	4 — Probability of Correct Matches: at least 3	4 — Probability of Correct Matches: 4
Dictionary = 1000 Words											
4	11	.89	.99	.79	.999	.96	.71	1.000	.995	.94	.63
6	7	.59	.83	.35	.93	.63	.21	.97	.81	.46	.12
8	6	.27	.46	.07	.60	.17	.02	.71	.29	.06	.005
Dictionary = 2000 Words											
4	11	.80	.96	.64	.992	.90	.51	.998	.97	.82	.41
6	7	.32	.54	.10	.68	.24	.03	.78	.38	.10	.01
8	6	.15	.27	.02	.38	.06	.003	.48	.11	.12	4×10^{-4}
Dictionary 5000 Words											
4	11	.62	.85	.38	.94	.68	.24	.98	.84	.51	.15
6	7	.16	.29	.03	.41	.07	.004	.50	.12	.01	6×10^{-4}
8	6	.07	.13	.004	.19	.01	3×10^{-4}	.23	.02	.001	2×10^{-5}

* This information is based on formulas given in:
"Mathematical Analysis of Various Superimposed Coding Methods" by S. Stiassny, IBM Research Center, Yorktown Heights, New York, April 1959.

Multiple Functional Relationships

A more complex problem is that of indicating multiple functional relationships among the terms to be recorded. One method, referred to as "interfixing" has been developed by the research and development group of the U. S. Patent Office. A special machine, known as the ILAS, is required for making searches by this method.

A new method, which may be used with the row type IBM 101 Electronic Statistical Machine, not only produces similar results, but also removes the limitations of the U. S. Patent Office method with respect to card space utilization. This new method is carried out in two stages. First, each of the keyterms is recorded, together with a number identifying its location on the card. Then, in subsequent rows, the relationships among the keyterms are recorded, in terms of single holes standing for the locations of the previously recorded keyterms in order. Keyterms may then be related in many ways in successive rows, one row for each relationship. Furthermore, a section of these rows is used to record certain types of functions characterizing the keyterm relationships. By punching, in a single row, a hole each for the affected keyterm locations and a hole each for the functions that apply, complex relationships may be expressed. As many rows may thus be recorded as are necessary to present the various desired relationships.

When searching, the machine first stores the location numbers of the keyterms on the card that matches the search request, and then tests for the wanted relationships and functions of those locations, which stand for the required terms.

A more detailed description of this method is given in a separate paper.

PREPARATION OF CARDS

Because of the row-by-row arrangement of information on record cards the preparation of such cards calls for special procedures and operations. Until special equipment becomes available, row-by-row card preparation is not particularly efficient. This temporary situation may be tolerated in information retrieval systems where the rate of accession is moderate.

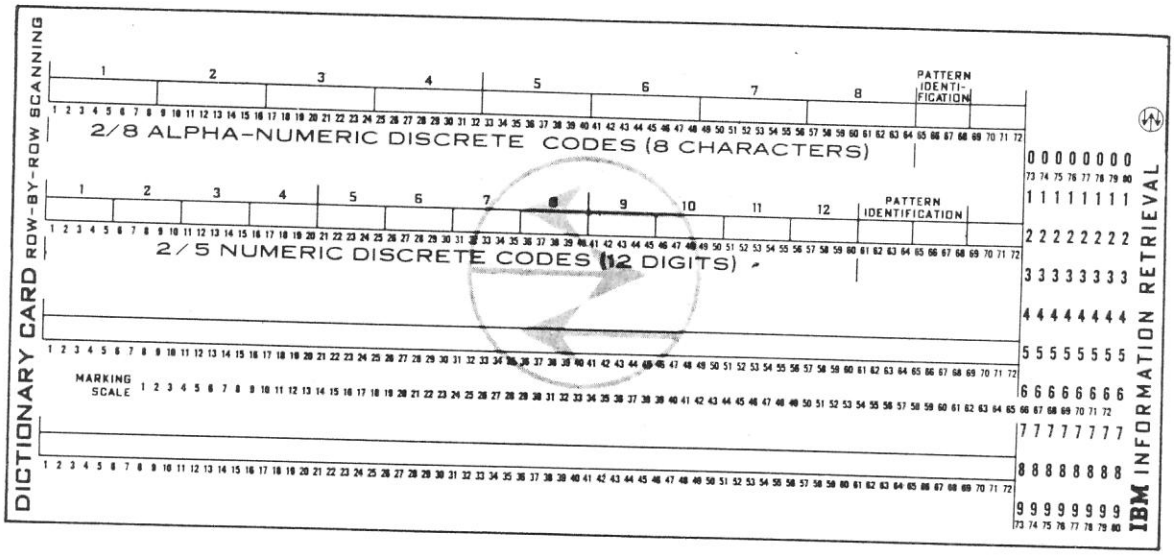
The task of card preparation may best be performed in two steps. The first is that of preparing a set of dictionary cards and the second is that of preparing the record cards with the aid of these dictionary cards.

Preparation of Dictionary Cards

The methods used here depend on whether the code representations are to be established by assignment or derivation. In the case of assignment it is best to generate a set of dictionary cards representing all permutations of the code notations to be assigned. The code is recorded in one of the rows of such cards and in the position within this row in which it is to appear on the record card. The generation of such card sets is best performed by electronic computers. Whenever a new dictionary entry is to be made, a next card is pulled from this master set and the newly assigned word is recorded on the card by means of conventional punching. This process creates a dictionary file.

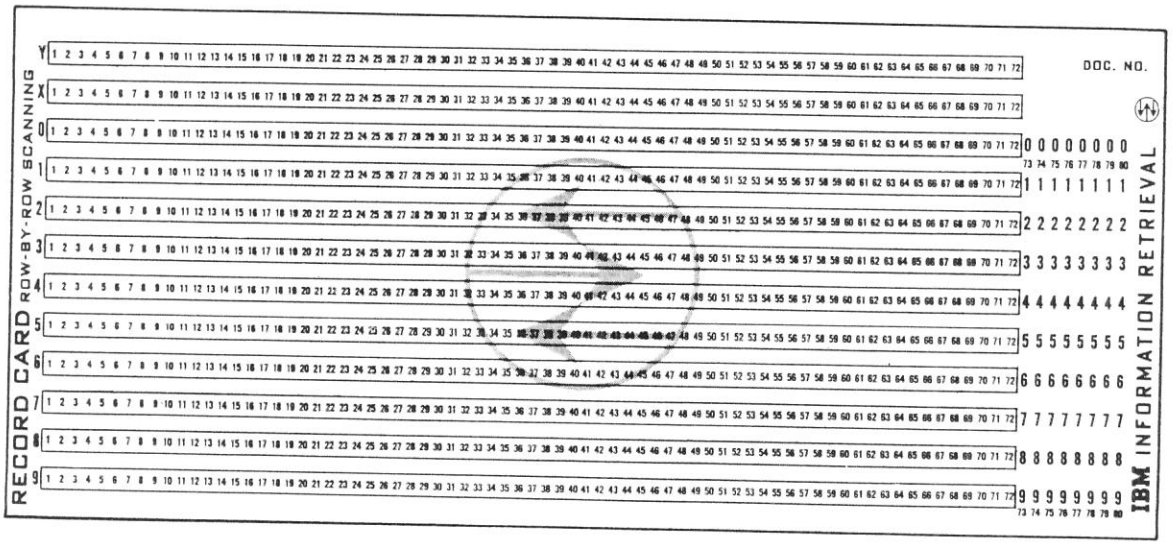
In the case where code notations are obtained by derivation from the original words, such derivation could either be performed by electronic computers, if the circumstances warrant it, or by manual means. In order to facilitate the manual process, the dictionary cards may contain a special "marking scale". This scale is printed across the card and serves to indicate by pencil marks those of the positions of the code that are to be punched. Because of the fact that on a card punch the location where a hole is to be punched is concealed by the punching device, the marking scale has been shifted to the right. The extent of the shift is such that when a mark is adjacent to the right edge of the punching station, the card is in the proper position for punching the hole which corresponds to the mark. The card punch operator needs only to depress the key at that time, and then proceed to advance the card until a next mark reaches a similar position, and repeat the operation. The marking of the scale may be simplified by using a template in conjunction with a cardholder to insure correct positioning of the marks.

The card form for dictionary file cards is shown below.



Preparation of Record Card

The creation of the record cards is carried out with the aid of the dictionary cards. Again, this process may be performed by electronic computers or manually. The manual process would be performed by means of a card punch and would consist of inserting a record card blank into the punch station and of inserting the dictionary card, pulled from the file, into the read station. By means of a special attachment to the punch, the row into which the code is to be punched on the record card may be selected and the code be duplicated into the corresponding row. This process would be repeated for each additional entry on the record card. The format of a record card is shown below.



In the case of tabular coding the individual pattern of a row may be marked on a dictionary card first and then punched in the manner just described.

In laying out the placement of the various card fields, for cards of this type, it is good practice to start assigning conventional IBM code fields from the right (80th column) end of the card, and to start assigning row-by-row code fields from the left (first column) end of the card.

I. A DESCRIPTION OF THE IBM 101 WITH ROW-BY-ROW DEVICE

The IBM 101 Electronic Statistical Machine has been popularly used in Information Retrieval applications for searching. The IBM 101 combines into one unit the functions of collecting and classifying facts on the basis of relative number or frequency of occurrence. Other functions include sorting, balancing, editing and printing summaries. Details of the operation of the IBM 101 are available in IBM Reference Manual A22-0502-0.

The row-by-row device for the 101 is an optionally available feature. The device permits discrimination between information recorded in different rows of a card. This discrimination is achieved by a special mode of recode selector operation. When in this special mode, a recode selector which is picked up by a digit impulse will remain transferred only for one row-time. A special test impulse is provided for each row during the card cycle. This impulse is used for routing through selector points.

The operation of the recode selectors on an IBM 101 with a row-by-row device attached requires special explanation. The last 10 recode selectors (51-60 on a full capacity machine) operate normally. They may be used either for standard selection or for recode hold operations. All recode selectors except the last ten are inoperative unless assigned to standard or special operation. The description which follows applies only to recode selectors which require assignment.

The row-by-row device allows for complete flexibility in the assignment of recode selectors for special operation. Any desired number of recode selectors, can be assigned for row-by-row operation by appropriate wiring in the row-by-row section.

The row-by-row section of the panel (Figure 1) replaces the sample selector section (AL-AN, 52-63). Each of its functions are briefly described.

1. Recode Assignment Hubs (AL-AN, 52-60 and AN 61)

   These hubs are used to assign recode selectors to either standard or special operation. The hubs in panel rows AL and AM represent groups of 5 recode selectors. The number shown represents the largest number in the group. For example, the common hubs labeled 15 represent recode selectors 11 through 15 inclusive.

2. STD (Standard) (AL-AN 63)

These three common hubs are used to assign recode selectors to standard operation by wiring to a desired group in the assignment section.

3. SPL (Special) (AL-AN 62)

These three common hubs are used to assign recode selectors to special row-by-row operation. Any selector picked up when in the row-by-row mode will remain transferred only for one row-time.

4. TST (Test) (AL-AM 61)

These two common hubs provide an impulse twelve times per card cycle; one for each row of the card. The impulse is used for routing through points of recode selectors which have been assigned to special row-by-row operation.

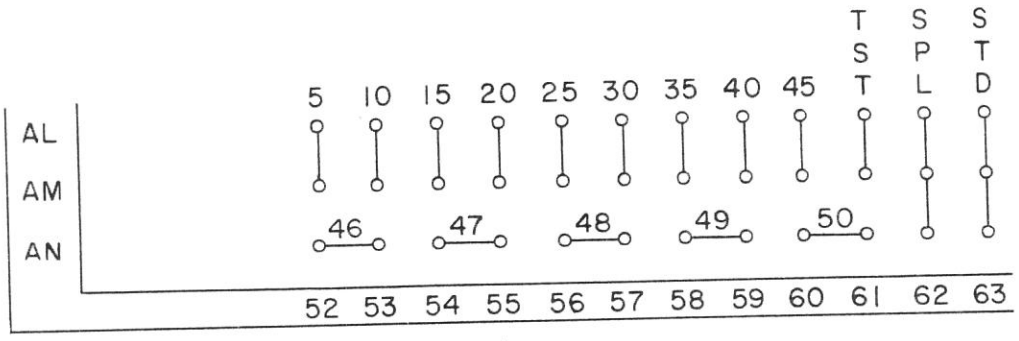


Figure 1

II. EXAMPLES OF INFORMATION RETRIEVAL APPLICATIONS

A. Figure 2 shows the wiring required to select into pocket 2 all cards with both

1. A 2-punch in cc 74.

2. A 42454852 combination of punches stored row-wise in the twenty position field 41-60. The 42454852 code is an example of a four out of twenty code, identified by column numbers, as explained on page 7.

The following describes the wiring shown in Figure 2:

1. Recode selectors 49 and 50 are assigned to special (row-by-row) operation.

2. Recode selectors 41-45 are assigned to standard operation.

3. Recode selector 44 is picked up by a 2 in cc 6 using digit emitter 14 (used as a column distributor).

4. Recode selectors 49 and 50 are picked up whenever punches occur simultaneously in a single row in columns 62, 65, 68 and 72. Recode selectors 49 and 50 will remain transferred only for one row when picked up.

5. The test (TST) impulse picks up recode selector 45 for the entire card cycle if recode selectors 49 and 50 are simultaneously picked up for one row.

6. The sort 4 impulse selects a card into sort pocket 2 if both recode selectors 44 and 46 are picked up for the card. Cards not selected fall into the reject pocket.

7. COUNTS TO to TEST is required for any selective sorting operation.

ROW-BY-ROW SCANNING SYSTEMS FOR IBM PUNCHED CARDS

AS APPLIED TO INFORMATION RETRIEVAL PROBLEMS

ERRATA

Page 30,	line 10	"a 2 in cc 6" should read: a 2 in cc 74
	line 13	"columns 62, 65, 68, and 72" should read: columns 42, 45, 48, and 52
	line 18	"44 and 46" should read: 44 and 45
Page 31		The wire from recode selector 45 pick-up, shown going to the transferred point of recode selector 50, should go to the transferred point of recode selector 49.
Page 32,	line 9	"group and a J" should read: group and an I
	line 18	"a J in cc 73" should read: an I in cc 73

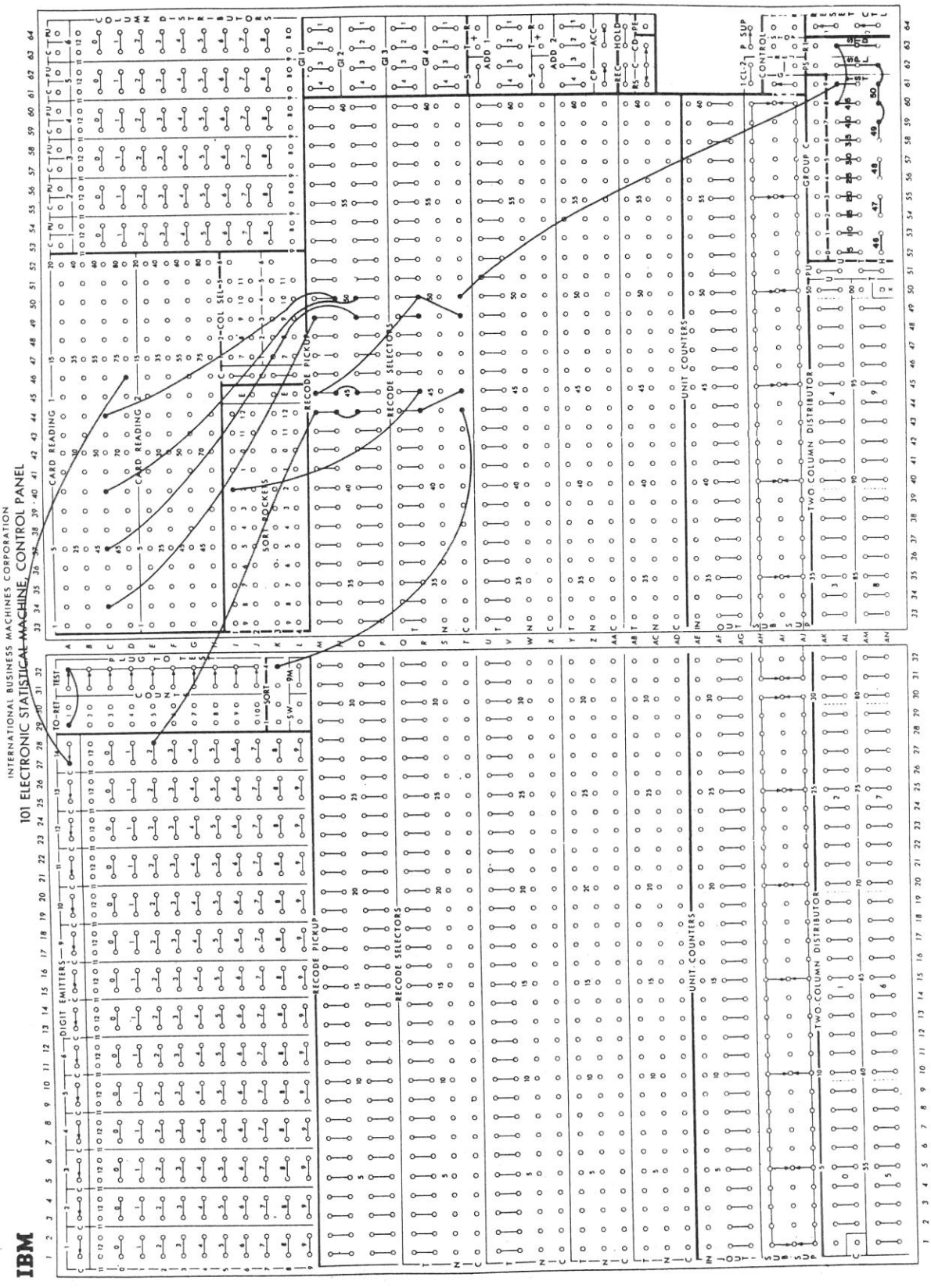

IBM
INTERNATIONAL BUSINESS MACHINES CORPORATION
101 ELECTRONIC STATISTICAL MACHINE CONTROL PANEL
CARD READING 1
CARD READING 2
SORT POCKETS
DIGIT EMITTERS
RECODE PICKUP
RECODE SELECTORS
UNIT COUNTERS
TWO COLUMN DISTRIBUTOR
ADD 1
ADD 2
ACC
HOLD
GROUP C
CONTROL

Figure 2

B. In most instances a single card will handle the keyterms associated with a single document. Occasionally the number of codes used in a single field exceeds 12. In these cases a second card is used. Actually any number of cards can be used for a single document. The method which is described in this example requires only that a distinguishing 9-punch be placed in the last card of each group including single card groups. A 12-punch is also placed in the same column for the last card of a multiple card group for visual scanning of selected cards. Thus, a 9-punch signifies a single card group and a J (9 and 12) denotes the last card of a multiple card group.

Actually two files are maintained in such an application; one for single card groups and one for multiple card groups. In this case it is not necessary to maintain order in the single card deck. Also the refiling of cards in the ordered multiple card deck is facilitated due to its smaller size. The same panel wiring is used for both decks.

The panel shown in Figure 3 is wired to select the last card of any group which satisfies the search criteria into Sort Pocket 2. A three-field, 4 out of 20 coding scheme is used. A 9 in cc 73 designates a single card group. A J in cc 73 signifies the last card of a multiple card group. Fields 01-20, 21-40 and 41-60 are used. A search is conducted for the following three codes: A (02051516), B (44475059), C (41474955).

The following describes the wiring of Figure 3:

1. Recode selectors 41 to 46 are assigned to special (row-by-row) operation.

2. Recode selector 18 (16-20) is assigned to standard operation.

3. Recode selectors 51-53 are unassigned. Being in the last ten, they are used for recode hold operation.

4. When condition A is met, selectors 41 and 42 are picked up for one row time. The test impulse (TST) then picks up selector 51 which remains transferred for the remainder of the card cycle. Similarly keyterm B picks up selector 52 and C selector 53.

5. Recode selector 18 is picked up by a 9 (early shot) in cc 73 using digit emitter 14 (used as a column distributor).

6. For a single card group in which keyterms A, B and C appear, the sort 4 impulse will pass through the transferred selectors 51, 52 and 53 and select the card into pocket 2 through the transferred selector 18.

7. For a multiple card group where one of the conditions is met before the last card (say B) one of the selectors 51-53 (52 for B) will be picked up. Since no 9 appears in cc 73, selector 18 will remain normal. The 11 impulse (a late shot) from digit emitter 13 will impulse recode hold. The picked up selector will remain transferred for the next card cycle and for every card cycle including the last card of the group. Similarly any selector (51-53) picked up before the last card will remain transferred through the last card of the group. Selectors 51-53 will be normal following the last card (9 in cc 73) of each group.

8. RS is jack-plugged to C so that recode selectors 51-60 are under control of recode hold.

9. COUNTS TO to TEST is required for any selective sorting operation.

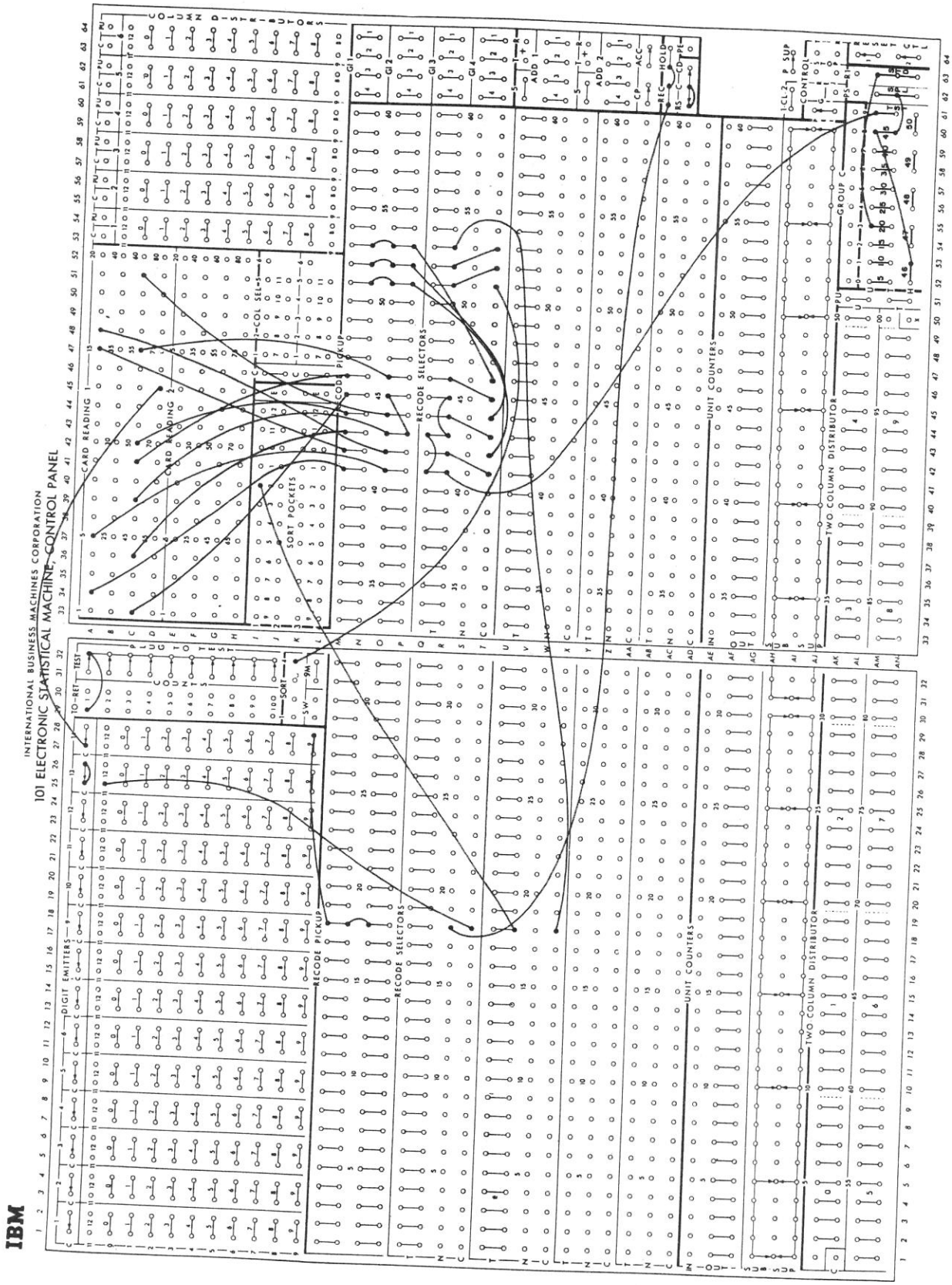

IBM
INTERNATIONAL BUSINESS MACHINES CORPORATION
101 ELECTRONIC STATISTICAL MACHINE, CONTROL PANEL
DIGIT EMITTERS
CARD READING 1
CARD READING 2
RECODE PICKUP
RECODE SELECTORS
UNIT COUNTERS
TWO COLUMN DISTRIBUTOR
SORT POCKETS
GROUP C
CONTROL

Figure 3

C. In conducting a search for documents which contain specific codes, it is possible to obtain additional useful information by extra panel wiring without loss of searching speed. For example, in searching for documents which contain codes A, B, C and D it is possible also to select the documents which contain three of the four keyterms, two of the four, etc. In fact, any combination of these specific keyterms can be selected.

Figure 4 illustrates the wiring required to select document cards which contain four of the keyterms A, B, C and D into pocket 4, three of the four keyterms into pocket 3, two of the four keyterms into pocket 2, one of the four into pocket 1 and none of the four into pocket 0. The diagram assumes that recode selector 1 is picked up by condition A, 2 by condition B, 3 by condition C and 4 by condition D.

Similar techniques can be used to select any desired combination of keyterms.

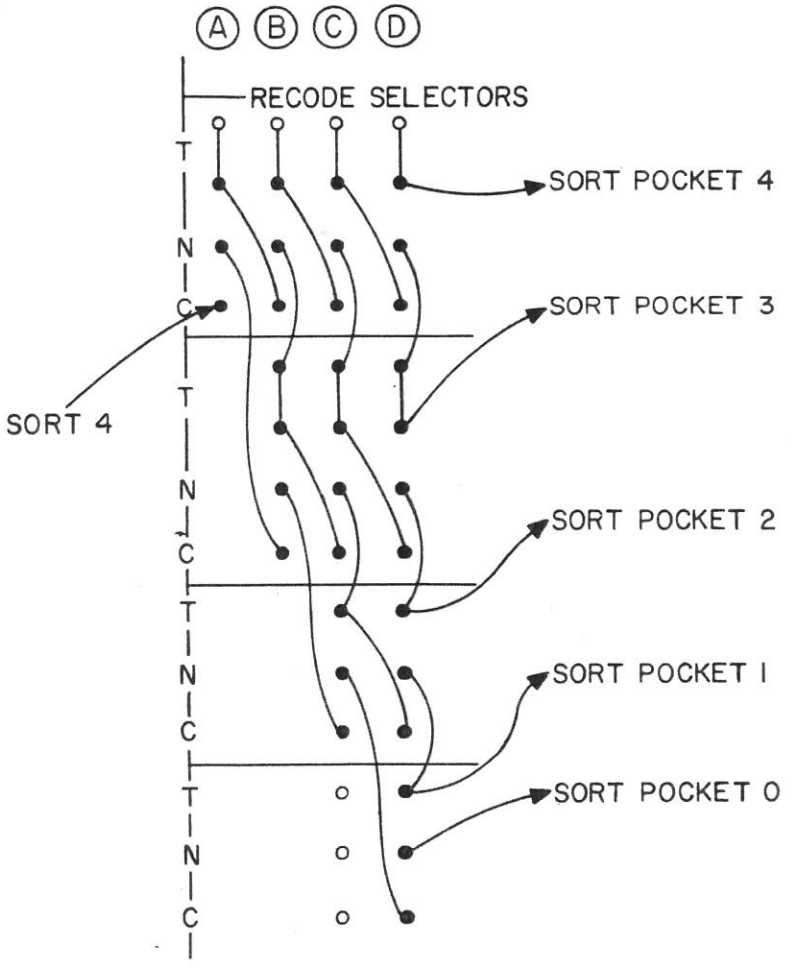


Figure 4

D. The wiring in Figure 5 illustrates a method of printing out indicative information, using the printing feature of the IBM 101. In this example the card format is similar to that used in Example B. Fields 01-20, 21-40 and 41-60 are used. Each card group is followed by an extra card with a 9-punch in cc 73. This card (known as a 9 M card) will also have punched in columns 78-80 the document label. The operational characteristics of the 101 require that the 9 M card be followed by a blank card. This procedure permits the printing out of indicative information and allows the deck to be kept intact.

In the example two codes will be used as search arguments, namely A(02051516) and B(44475059). The panel is so wired that whenever either an A or a B occurs in any card group the label of that group will be printed as well as a specific indication of the presence of an A or a B. The processing is continuous with automatic interruptions for printing following a card group which contains A or B.

The following describes the wiring in Figure 5:

1. Recode selectors 41 and 42 are picked up by condition A; 43 and 44 by condition B (wiring not shown).

2. Recode selector 21 is picked up by condition A and remains transferred for the card cycle. Similarly selector 22 is picked up by condition B.

3. Recode selector 25 is picked up by a 1 impulse (late shot). The continuous pulse from C in the REC HOLD section passes through the transferred points of selector 25 and is used to pick up selector 51 if either A or B have been encountered in the card.

4. The counts to 9 and return circuit is used to add a one into unit counter 1 if an A is present in the card. A one is added to unit counter 2 if a B is present. Following a print cycle both counters are returned to zero.

5. Recode selector 52 is picked up early in the card cycle following one in which an A or B occurred. The counts to 10 and return circuit which adds the indicative information in cc 78-80 into ADD 1 is disabled as selector 52 is picked up. Thus, the label is added into ADD 1 only once for a card group which contains an A or B. The label is read from second reading.

6. When selector 51 transfers, a 9 M card initiates a print cycle. The 9 M card is looked for at the second reading station. The printed line consists of the label and the number of cards in the group which contain A and the number which contain B.

7. Cards are sorted into pocket 0.

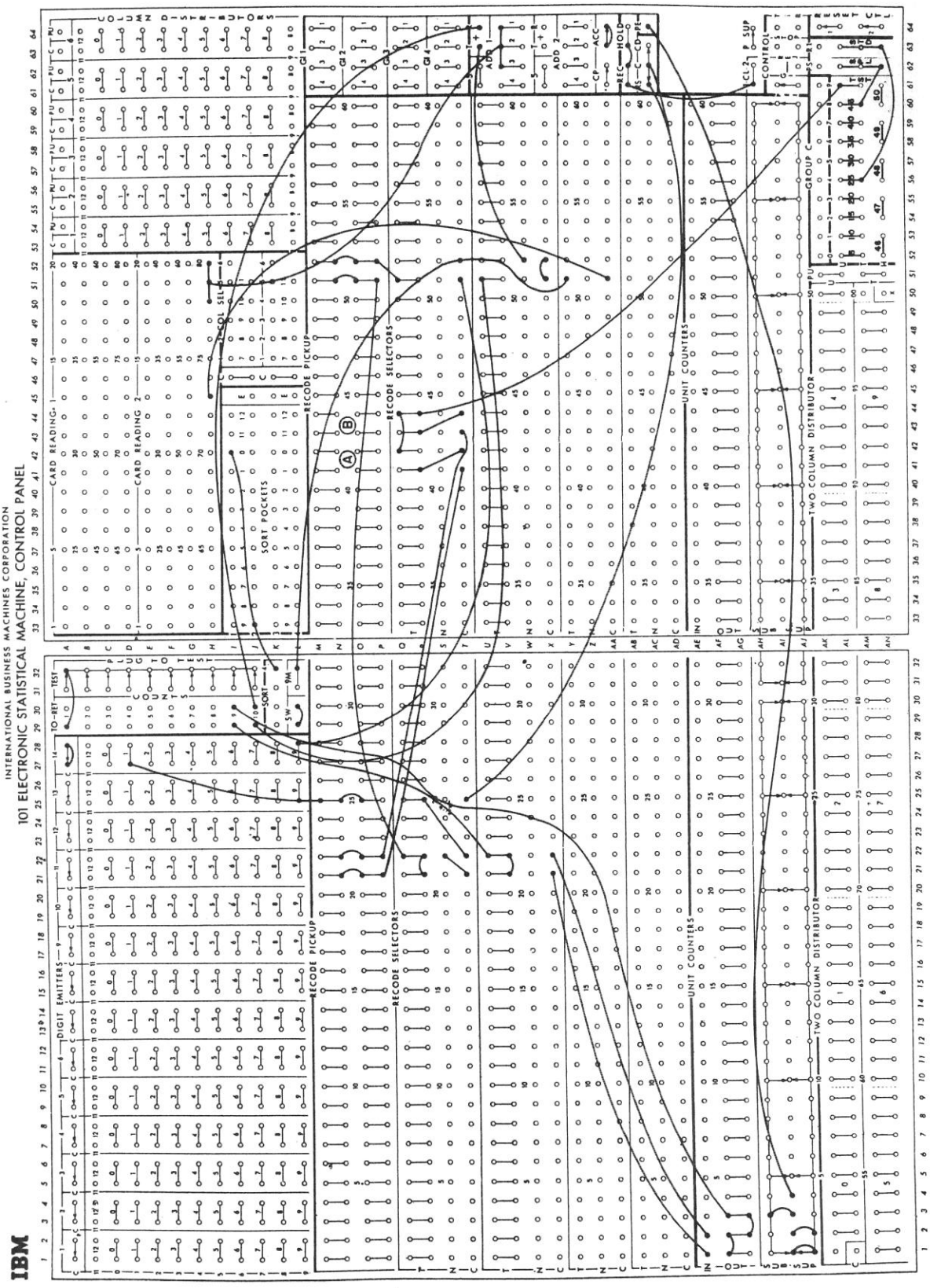

IBM
INTERNATIONAL BUSINESS MACHINES CORPORATION
101 ELECTRONIC STATISTICAL MACHINE, CONTROL PANEL
DIGIT EMITTERS
CARD READING 1
CARD READING 2
SORT POCKETS
RECODE PICKUP
RECODE SELECTORS
UNIT COUNTERS
TWO COLUMN DISTRIBUTOR
GROUP C
CONTROL
ADD 1
ADD 2
ACC
HOLD
REC
P. SUP

Figure 5