THRESHOLD LOGIC APPLIED TO CHARACTER RECOGNITION

A Thesis

Presented to

the Faculty of the Department of Electrical Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Electrical Engineering

---

by

Wiley Richard Flanakin

January, 1967

THRESHOLD LOGIC APPLIED TO CHARACTER RECOGNITION

An Abstract

of

A Thesis

Presented to

the Faculty of the Department of Electrical Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Electrical Engineering

---

by

Wiley Richard Flanakin

January, 1967

ABSTRACT

The fact that many complex counting and decision functions can be realized quite simply with threshold gates suggests that they may be used to considerable advantage in problems of character recognition. A simplified recognition problem is considered involving the identification of any one of 23 alpha-numeric characters when it is superimposed on an m x n matrix. Each character is required to be identifiable under any degree of translation, stretching, and compression within the confines of the matrix. It is shown that the number of threshold gates required increases linearly as the dimensions of the matrix increase linearly. A threshold gate having the necessary output power to drive the required loads for this application is described together with its implementation in a small experimental model. The model is designed to recognize only a select group of characters, the numeric characters. The primary purpose of the model is to demonstrate some of the diverse applications of threshold logic to character recognition.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE PAGE

LIST OF TABLES

CHAPTER I

INTRODUCTION

There are today a number of outstanding problems in the field of communications and control which depend for their solution on the analysis, classification, and recognition of pictorially displayed information. The best-known example of this is the machine reading of printed or handwritten documents. Other examples are the machine recognition of human speech, the prediction of weather from past weather maps, and the classification of plants, archeological finds, and fingerprints. In each of these, the key problem is the recognition of a characteristic geometric pattern.

Of course there is nothing novel about the importance of recognizing patterns to the solution of problems in science. Pattern recognition, i.e., the recognition of order in chaos is the basis for the application of inductive inference and is, therefore, one of the key factors in the advancement of science. This thesis is concerned with patterns in two-dimensional, spatial presentations, with time as a possible third dimension. In all subsequent references to patterns, this restriction to two-dimensional geometric configurations is to be understood.

Now it will be convenient to distinguish between the problem of classifying patterns and that of recognizing them. In the classification problem, a variety of geometric configurations are given and it is desired to separate them in accord with their characteristic patterns. The final objective is a hierarchy of classes which is not specified a priori but is developed as the classification process proceeds. Familiar examples of this are the classification of plants and archeological finds. In contrast, in the recognition problem, the number of admissible classes is fixed and each class is uniquely defined, frequently by some "standard" member of the class; the problem is to determine the class with which a given pattern is associated. In either problem, however, the geometric configurations must be examined for their pattern content. Since the patterns are invariably obscured in a background of "noise", the procedure becomes one of "filtering" the geometric configuration. Stated another way, given configurations must be subjected to transformations which will highlight their pattern content.

The process of having geometric configurations undergo transformations to bring out their characteristic patterns is a highly subjective one. It is intimately a function of the source of the geometric data and the aims and objectives

for which the analysis is being carried out. Thus different objectives for pattern classification require the use of different classifying criteria. For example, it is unlikely that criteria applicable to the classification of cloud formations for meteorological purposes will be suitable for the classification of handwriting specimens. The decision scheme for pattern recognition or classification must have great flexibility as well as the ability to utilize a wide variety of classifying criteria.

Considerable study has been devoted to pattern recognition by self-organizing systems using threshold devices. There have been several important reasons for using the threshold device. First, indications are that the animal nervous system is composed of nets of neurons whose behavior resembles a threshold device. Since attemps are being made to simulate learning by the nervous system, it is only natural that the threshold device was chosen as the basic unit in these simulated systems. Secondly, the threshold device, in its own right, possesses desirable characteristics for a self-organizing system. It can (1) realize a large number of switching functions with a single unit while any function can be realized with a large enough number of units and (2) the realized functions may be easily varied.

Both threshold logic and pattern recognition have been areas of increasing interest and activity during the past few years. This thesis will show how some of the techniques of threshold logic may be exploited to considerable advantage in problems of pattern and character recognition. To accomplish this, a simplified character recognition problem involving the identification of printed alpha-numeric characters will be considered. While the character recognition problem chosen is believed to be of sufficient complexity to warrant extrapolation of results, the main emphasis will be on the techniques of threshold logic and not on character recognition per se.

In what follows, it will be convenient to divide the process of character recognition into two parts as was done by Marill and Green, and has been adopted since by other authors.[1] These two parts are a receptor, which makes initial characterizing measurements on physical samples of the patterns to be recognized, and a categorizer, which attemps to identify the pattern under consideration using the measurements provided by the receptor. In the simplified recognition problem to be considered, it will be shown that both the receptor and the categorizer may be designed and implemented using only threshold logic.

CHAPTER II

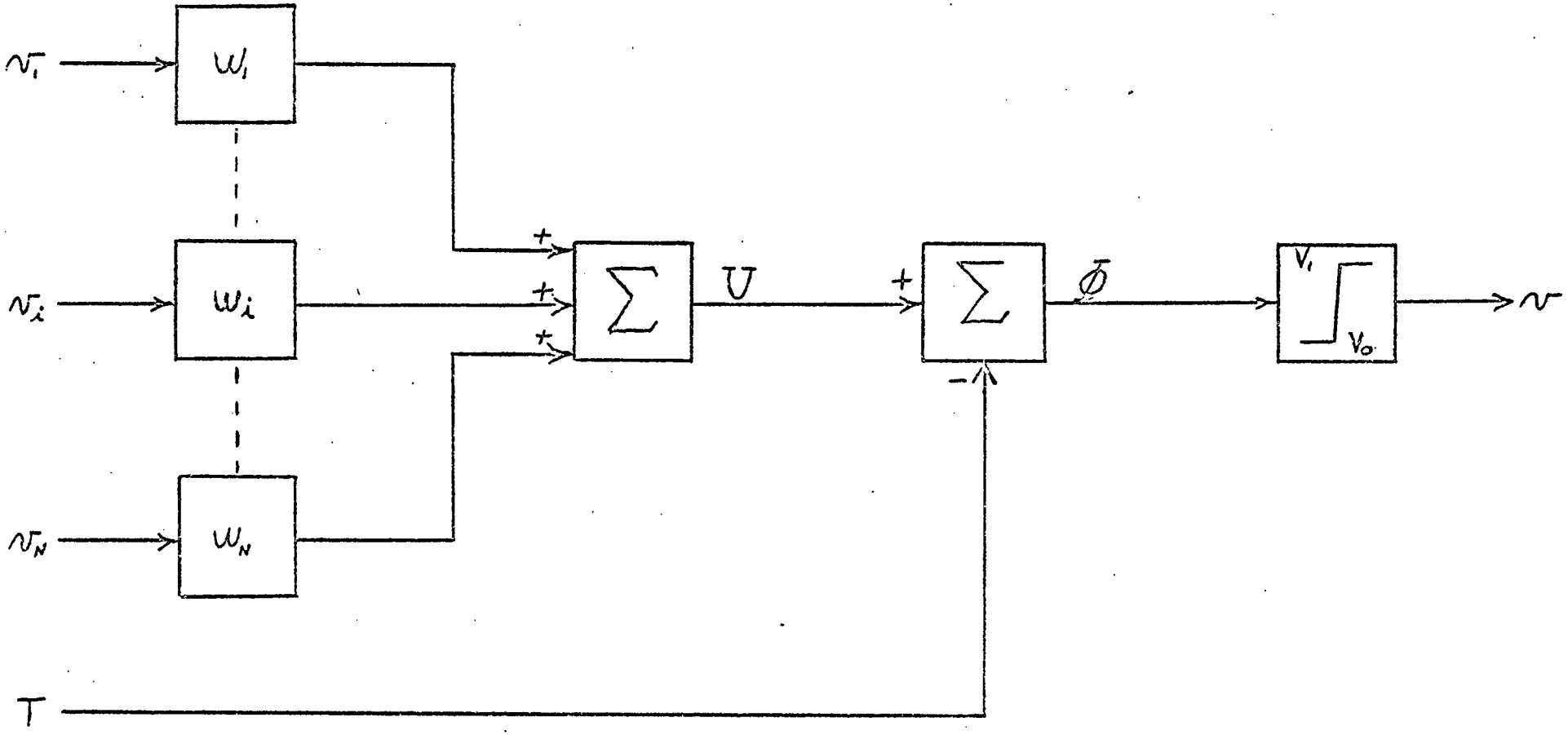
THRESHOLD LOGIC

In the introduction, several important reasons for using threshold devices were described. Now a formal description of a threshold gate will be presented.

Consider the block diagram of Fig. 1.[2] The inputs v_1 to v_N are chosen to be binary valued and the system itself produces a binary output. The unit is thus a switching circuit. Its operation is as follows. Each input is multiplied by a gain of the associated weight amplifier. The outputs of these amplifiers are fed into a linear adder which yields the sum:

$$U = \sum_{i=1}^{N} w_i v_i .$$

This sum is compared to a constant input T which is called the "threshold" of the system. The comparison, which is accomplished in the second adder, consists of forming the difference $\phi = U - T$. ϕ serves as the input to the quantizer which is defined by the input-output relationship indicated graphically in its block. If its input is equal to or greater than zero, the output is V_1. If the input is negative, the output is V_0. The over-all behavior can now be summarized as follows: (1) If $\sum_{i=1}^{N} w_i v_i - T \geq 0$ then



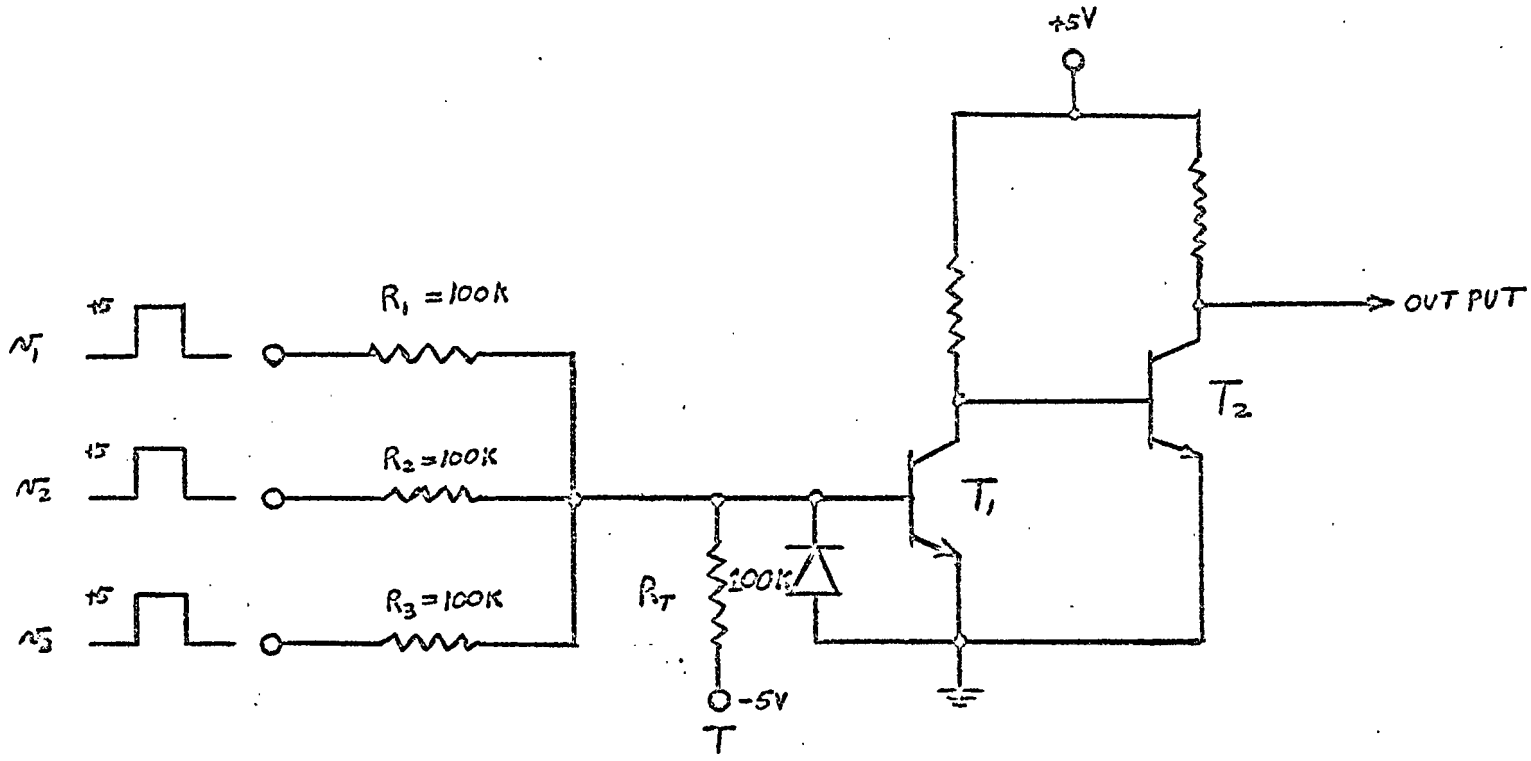
Block Diagram of a Threshold Device

Fig. 1.

the output is V_1, and (2) if $\sum_{i=1}^{N} w_i v_i - T < 0$ the output is V_0. That is the system yields a binary output whose value depends on whether or not the weighted sum of the binary inputs exceeds the threshold value. A system which performs in this manner is called a "threshold device or gate."

The gains w_1 to w_N and the threshold T are the parameters of the system. They may be set to any positive or negative value. They thus determine the particular switching function which the device implements.

As an example of a threshold gate consider the circuit of Fig. 2. The inputs v_1, v_2, and v_3 are voltage pulses whose amplitudes are either +5 volts or 0 volts. V_t is a d.c. voltage of constant magnitude equal to -5 volts. The resistors R_1, R_2, and R_3 correspond to the amplifiers of Fig. 1, and their values are analogous to the gains. For any gate R_t is selected for the proper threshold desired. The sum ϕ is formed by connecting all the resistors as shown in the figure. If ϕ is less than +.2 volts, transistor T_1 will be cutoff causing the output to be approximately zero volts. If, however, ϕ is greater than +.6 volts, transistor T_1 will be saturated causing the output of transistor T_2 to be +5 volts. For values of ϕ between +.2 and +.6 volts, the output will be equal to some value between zero and +5



A Threshold Gate of T = 2

Fig. 2.

volts. However, it will be assumed that whenever this circuit is used, the scaling is so arranged that these values do not occur. The relations for all the possible values are summarized in TABLE I.

In the above description, a threshold device was identified as a system whose performance was equivalent to that of the block diagram of Fig. 1. A more formal definition of a threshold device is that it is a system with the following properties:

1. Inputs are accepted in binary form.
2. A weighted linear sum is formed on these inputs:
$$U = \sum_{i=1}^{N} w_i v_i .$$
3. A number a_0 is added to U to give the sum
$$\phi = a_0 + \sum_{i=1}^{N} w_i v_i .$$
4. A binary output is delivered whose value is determined by whether the sum ϕ is less than some fixed value or not.

If we let $a_0 = -T$ this definition is equivalent to the input-output relationship of the block diagram of Fig. 1.

"Threshold logic" is defined as the switching logic which can be realized by threshold devices. The switching functions which are so realized have been called "threshold functions" and "linearly separated truth functions." Muroga

TABLE I

V_t	v_1	v_2	v_3	ϕ	v
-5	0	0	0	-1.25	0
-5	5	0	0	0	0
-5	0	5	0	0	0
-5	0	0	5	0	0
-5	5	0	5	1.25	5
-5	0	5	5	1.25	5
-5	5	5	0	1.25	5
-5	5	5	5	2.50	5

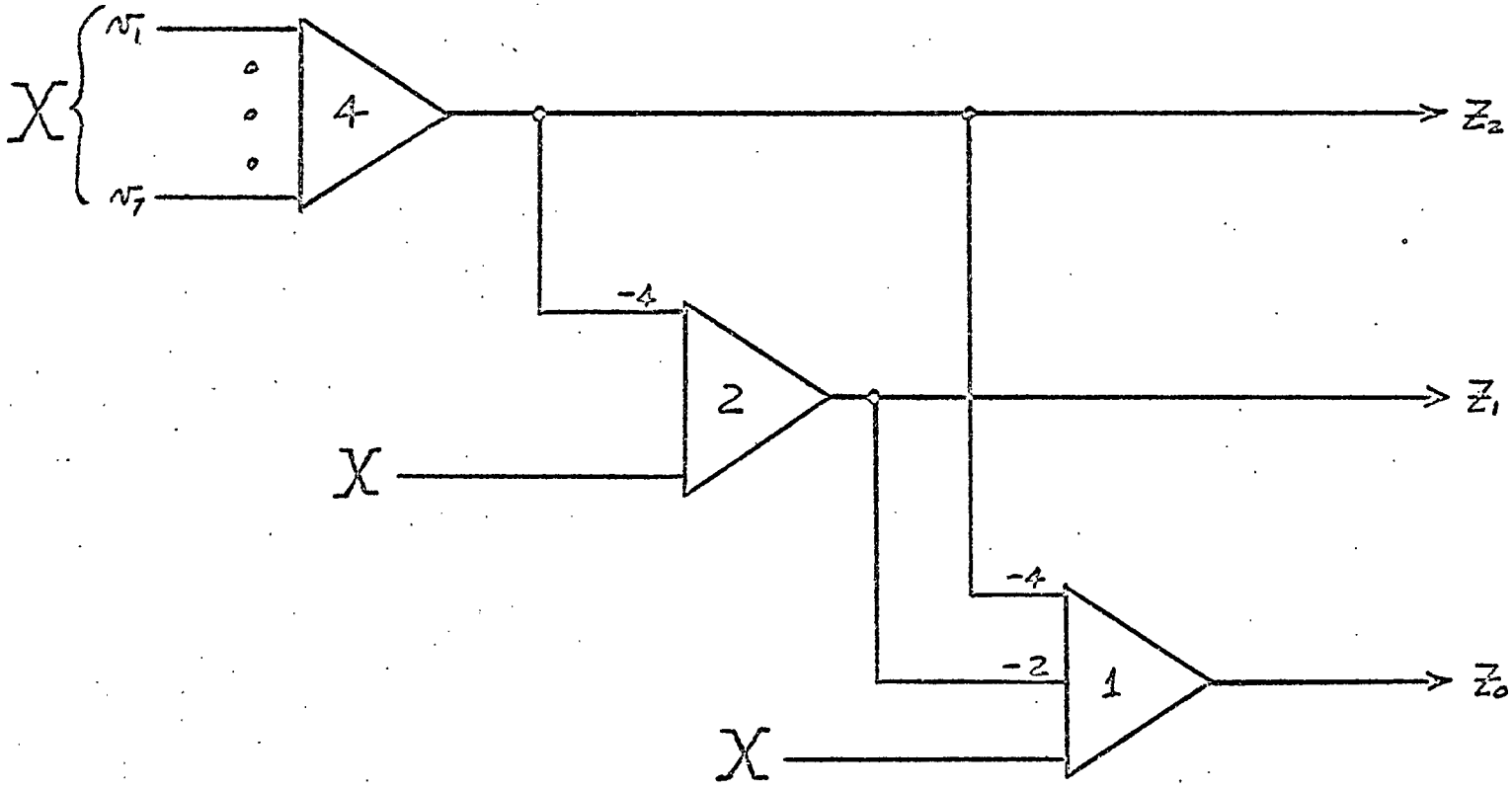
$$\Phi = \frac{\sum_{i=1}^{N} v_i + \frac{R}{R_t} V_t}{\square + \frac{R}{R_t}}$$

Input-output Relations For The Circuit of Fig. 2 With a Threshold Value of 2.

refers to them as "majority decision functions" although this name is more commonly used for a switching circuit whose output is determined by the presence or absence of a majority of the inputs.[3]

A circuit which illustrates the logical power of threshold gates and is also quite useful for pattern recognition is shown in Fig.3. This circuit has seven inputs, $\nu_1, \cdots, \nu_7$ denoted by χ. Its output, Z_0, Z_1, Z_2, is a three-bit binary number which gives the number of units on the inputs. The numbers shown within the gates represent the thresholds and the numbers on the inputs; the weights. When no weight is given, it is understood to be one.

This circuit was originally given by Kautz.[4] The extension to a greater number of inputs should be obvious from the figure. In general, $2^n - 1$ inputs require n gates. Note that Z_0 must be the odd-parity function. Note also that if the inputs are tied together the device becomes essentially an analog-to-digital converter. This circuit is extremely useful in implementing a variety of arithmetic functions. Also a "scaling" of the range over which it counts is easily accomplished. The receptor portion of the pattern recognition device to be described will use variations of this basic circuit extensively.



A Basic Counting Circuit

Fig. 3.

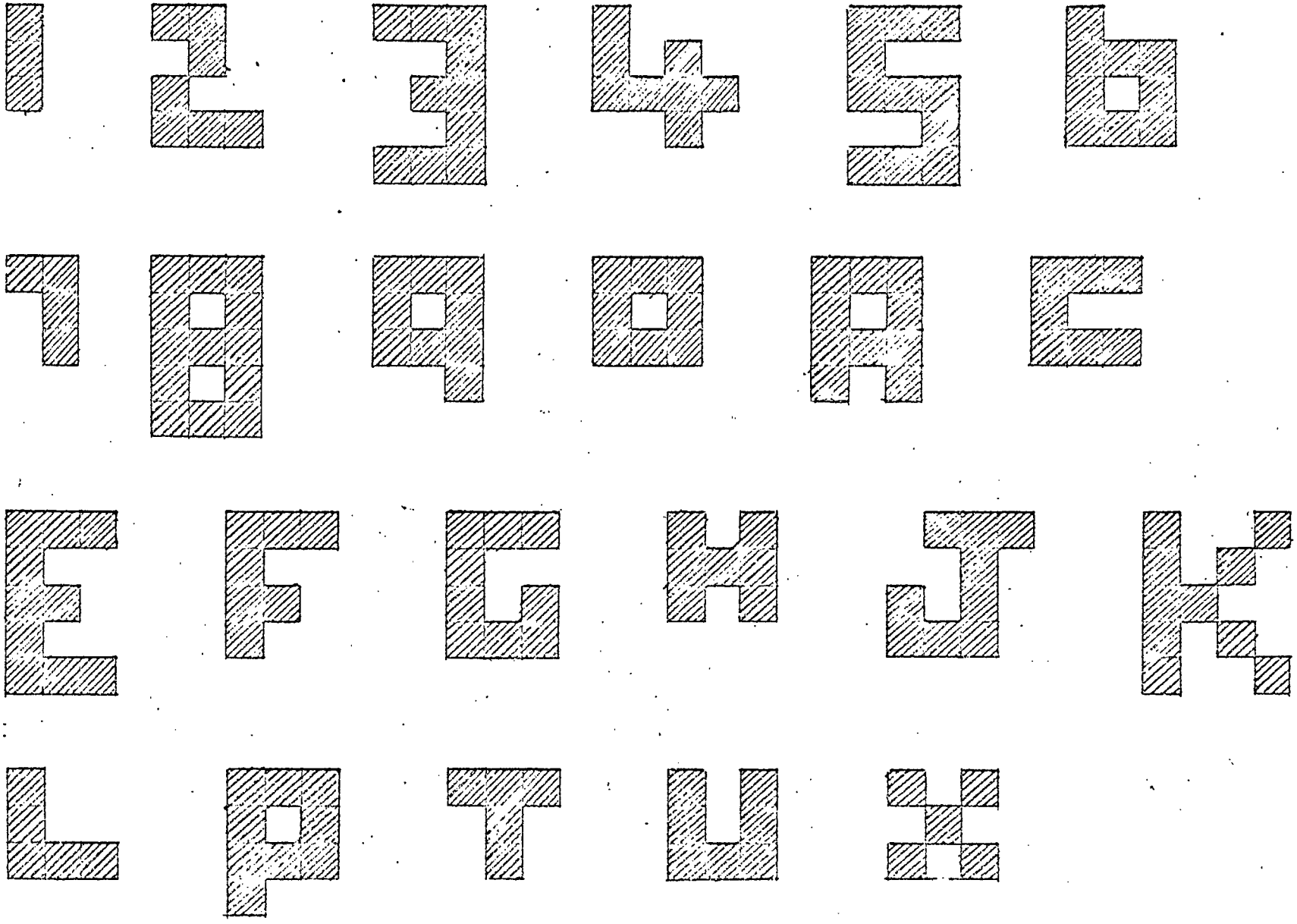
CHAPTER III

A CHARACTER RECOGNITION PROBLEM

The recognition problem to be considered involves the identification of any one of the twenty three "square" alpha-numeric characters in the forms shown in Fig. 4. The other ten alpha characters were not included for it would require a more complex categorizing scheme to distinguish all the alpha-numeric characters implemented discretely with "squares." It will be assumed that the character has been superimposed on an m x n matrix and that each element of the matrix has been quantized so that it is black or white. It will be required that each alpha-numeric character be identifiable under any degree of translation, stretching, or compression with in the confines of the matrix. Stretching and compression will be defined as a uniform lengthening or shortening of the lines of the character either vertically or horizontally without changing the width of such lines. For example, the numeric character "0" would have to be identifiable in any of the positions or configurations such as are shown in Fig. 5. Compression will be allowed down to the sizes shown in Fig. 4.

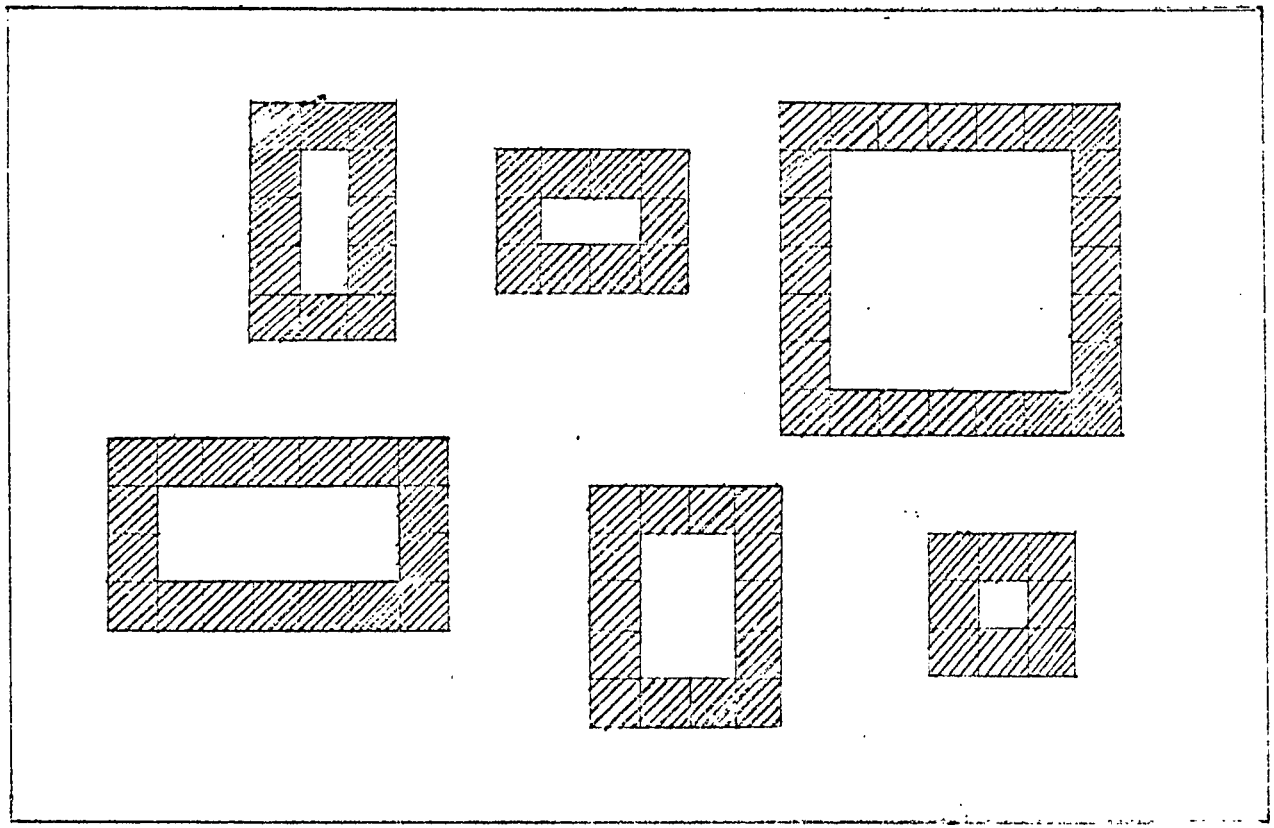
Rotation and magnification will not be included, but recognition in these cases will be discussed in a later

chapter. A block diagram of a pattern recognizer to be considered is shown in Fig. 6.



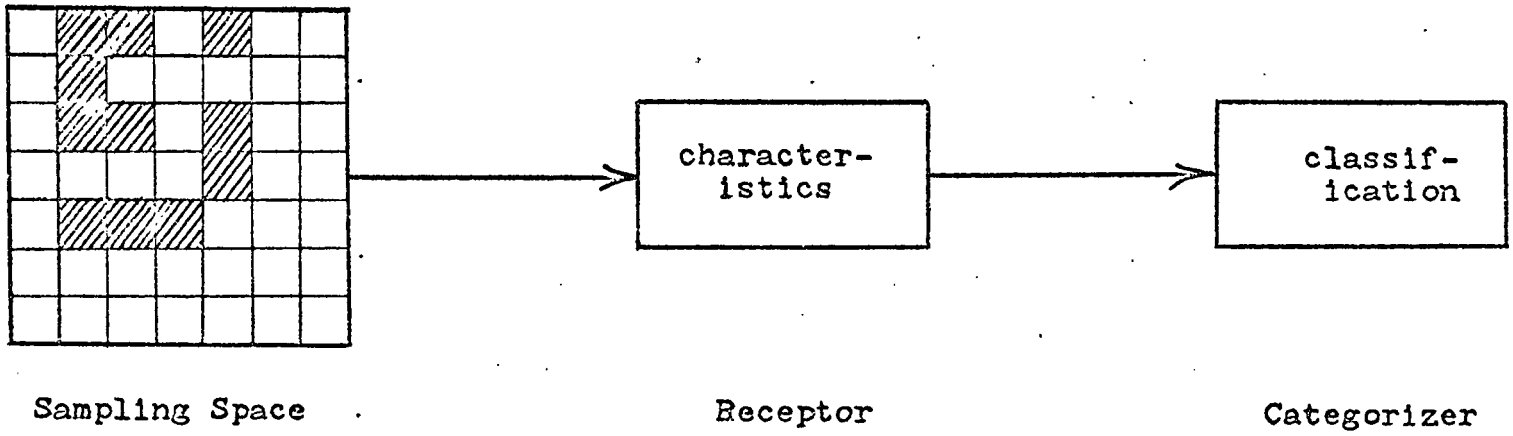
Alpha - Numeric Characters to be Identified

Fig. 4.



Some Possible Positions and Configurations of the Numeric Character "0".

Fig. 5.



Block Diagram of a Pattern Recognizer

Fig. 6.

CHAPTER IV

DESIGN OF A RECEPTOR

The receptor will make initial characterizing measurements on physical samples of the patterns displayed on the sampling space as shown in Fig. 6. The first step is to attempt to discover certain invariant characteristics of the individual alpha-numeric characters. Since threshold logic is to be employed, those characteristics which involve counting will be especially useful and easy to implement. With respect to "0", for example, it is seen that under any degree of translation, stretching, or compression, the following characteristics still hold:

1. Exactly two horizontal lines of the matrix will have three or more units.
2. At least three horizontal lines of the matrix will have two or more units.
3. No horizontal lines of the matrix will have exactly one unit.
4. At least one horizontal line of the matrix will have exactly two units.

Similar characteristics will remain invariant with respect to the vertical lines. One surprisingly useful characteristic for identification purposes is the number of

units in the outside edges of an alpha-numeric character. In particular, the following additional characteristics can be defined:

1. At least three units appear in the bottom edge.
2. At least three units appear in the right edge.
3. At least three units appear in the top edge.
4. At least three units appear in the left edge.

When these are tabulated and combined with other characteristics that apply to the twenty three alpha-numeric characters, Table II results. The twelve invariant characteristics used in Table II are identified on page 21. A "1" indicates that the letter always has the characteristic and "0" that it never has it. Blanks indicate that the characteristics may or may not be present. Inspection of this table shows that these twelve characteristics are sufficient to distinguish any of the twenty-three alpha-numeric characters from any other; i.e., every pair of rows differs in at least one column. Actually, eight columns C_3, C_5, C_6, C_8, C_9, C_{10}, C_7, and C_{12} are sufficient, but by including all twelve, a significant gain in tolerance to "noise" will be achieved.

Next, the implementation of the receptor can be considered. An examination of the characteristics reveals that for each horizontal row of an m x n matrix, it is only necessary to know whether it has zero, one, two, or at least three units.

TABLE II

Alpha-Numeric Characteristic Matrix

Characters ↓	Characteristics C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
1	0	0	1	0	0	0	0	0	0	1	0	1
2	1	-	1	-	-	1	1	0	1	-	-	0
3	1	1	1	1	0	1	-	1	1	1	1	0
4	1	0	1	1	-	1	1	0	0	0	0	1
5	1	1	1	0	1	1	0	-	1	1	1	1
6	1	1	1	1	-	1	0	1	1	1	0	1
7	0	0	1	-	0	0	1	0	0	1	-	0
8	1	1	0	1	1	1	0	0	1	1	1	1
9	1	1	1	1	-	1	0	1	0	1	1	1
0	1	1	0	1	-	1	0	1	1	1	1	1
A	1	1	0	1	1	1	0	1	0	1	1	1
C	1	1	1	0	0	0	0	1	1	0	1	1
E	1	1	1	-	0	1	-	1	1	0	1	1
F	1	-	1	-	0	0	-	1	0	0	1	1
G	1	1	1	1	-	1	-	1	1	1	1	1
H	1	0	0	1	-	1	1	0	0	1	0	1
J	1	1	1	1	-	-	1	1	1	0	1	0
K	1	0	-	1	0	0	1	1	0	-	0	1
L	0	0	1	0	0	0	1	0	1	0	0	1
P	1	1	1	0	-	1	0	1	0	1	1	1
T	0	0	1	0	0	0	1	0	0	0	1	0
U	1	0	0	1	-	1	1	0	1	1	0	1
X	1	0	1	1	0	-	1	1	0	0	0	0

List of Twelve Invariant Characteristics

C_1 - At least two horizontal lines have two or more units.

C_2 - At least two horizontal lines have three or more units.

C_3 - At least one horizontal line has exactly one unit.

C_4 - At least one horizontal line has exactly two units.

C_5 - At least two vertical lines have four or more units.

C_6 - At least two vertical lines have three or more units.

C_7 - At least one vertical line has exactly one unit.

C_8 - At least one vertical line has exactly two units.

C_9 - At least three units appear in the bottom edge.

C_{10}- At least three units appear in the right edge.

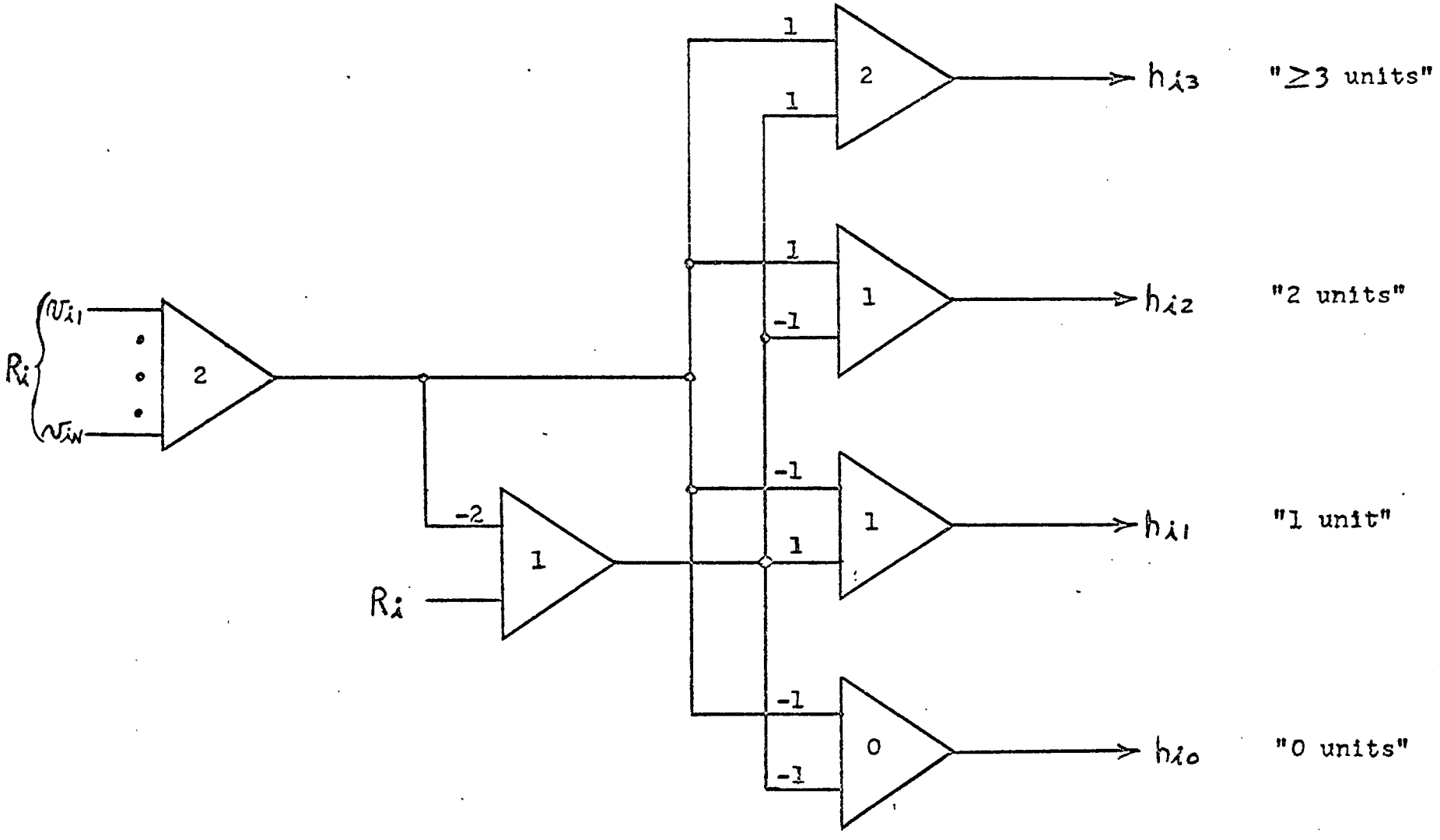
C_{11}- At least three units appear in the top edge.

C_{12}- At least three units appear in the left edge.

Fig. 7 shows the six gates necessary to accomplish this for a single row. Now the six horizontal characteristics can be obtained. For example, C_2 is the output of a threshold gate having the "h" inputs, h_{13} through h_{13}, and a threshold of two. Similar gates results for C_1, C_3, and C_4 where the inputs are $h_{12} + h_{13}$, h_{11}, and h_{12}, respectively, and the corresponding thresholds are 2, 1, and 1. The edge characteristics, C_9 and C_{11}, seem at first glance to involve prohibitively complicated logical functions. To obtain C_9, for instance, the matrix must some how be logically scanned row by row starting at the bottom until a nonzero row is reached. Then it must be determined whether or not this row has at least three units. Actually, each of these characteristics can also be derived from a single threshold gate. Appendix I describes this implementation in detail.[5]

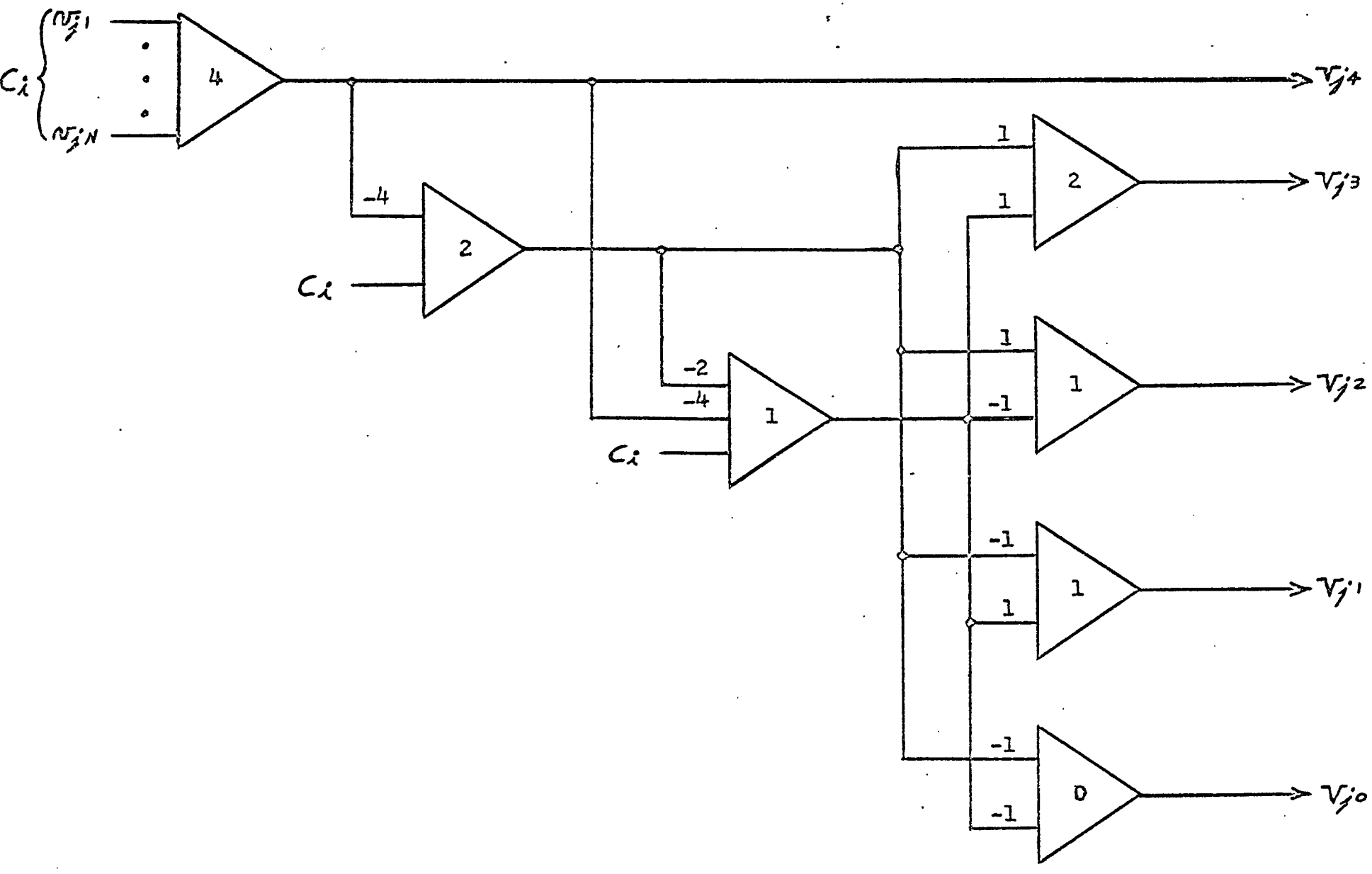
For the five vertical characteristics, seven gates are required for each column in order to specify whether the column has 0, 1, 2, 3, or ≥ 4 units. The circuit is essentially the same as Fig. 7 except that a three-gate counter is required (see Fig. 8). Again, the implementation of the gates to generate the five vertical characteristics is quite straightforward.

Given an m x n matrix, the total number of threshold gates to realize the receptor may be calculated as follows:



Counting Circuit For Row i.

Fig. 7.



Counting Circuit For Column j.

Fig.8.

To determine the number of units in each row: $6m$

To determine the number of units in each column: $7n$

To generate C_1 through C_8: 8

To generate C_9 and C_{11} (see Appendix I): $2\left(\frac{2m + 1}{5}\right)$

To generate C_{10} and C_{12} (see Appendix I): $2\left(\frac{2n + 1}{5}\right)$

Approximate total number of gates: $6.8m + 7.4n + 8.6$

This means about 150 gates for a 10 x 10 matrix. The importance of the above formula appears to be not in the exact number of gates indicated, but rather in the fact that it is linear. This means that the number of gates required increases linearly with the dimensions of the matrix, not exponentially as is so often the case with switching functions.

CHAPTER V

DESIGN OF A CATEGORIZER

Once the implementation of the receptor has been accomplished and the twelve binary characteristics, C_1 through C_{12}, are available, the design of the categorizer may be considered. Here again only threshold gates will be employed.

The concept of using threshold gates for a categorizer is certainly not new. In fact, a number of investigations have been conducted in this area by Highleyman, Mattson, Widrow, and many others. Most of these investigations have treated the threshold gate as an adaptive element. The attempt has been to train the gate by successively presenting various inputs to the gate and then adjusting the weights of the gate according to the correctness of the observed output. In this regard, the receptor as such is often bypassed completely in order to better assess the decision-making capabilities of the gates themselves. These investigations have met with varying degrees of success, but certainly the basic feasibility of the approach has been established. Particularly with problems where little a priori knowledge of the recognition process itself is available, this adaptive approach seems quite promising.

Here the emphasis is somewhat different. Because of the

relatively complex nature of the receptor, the problem of distinguishing one alpha-numeric character from another has, in a sense, already been solved. For example, one way to implement the categorizer would be to use an "and" gate for each character and to require that it have precisely the characteristics given in Table II. Thus, for "0" the gate would be

$$\text{"0"} = C_1C_2\bar{C}_3C_4C_6\bar{C}_7C_8C_9C_{10}C_{11}C_{12}.$$

Since each row of Table II differs from every other in at least one column, it follows that the output of each "and" gate would be one if and only if the complete set of binary characteristics of the corresponding character were present. Such a scheme, however, is far from efficient. If an "0" for example, is altered or distorted slightly so that even one out of the twelve characteristics changes value, then the "0" would no longer be identified. More desirable would be a method whereby "0" could be uniquely identified on the basis of a much smaller subset of characteristics. In this regard, the vote-taking ability of the threshold gate can be exploited. Since each character is known to be identifiable by a single gate, the problem becomes that of choosing the weights of the gate to optimize its performance. Appendix II defines this problem formally and shows that it may be reduced to a classic problem in game theory.[6]

Using these methods, the weights and thresholds of the twenty-three gates of the categorizer are found to be those shown in Table III. Note that fewer characteristics are vital for identification of each character. With "0", for example, the only requirement is that two of the three characteristics, C_3, C_8, and C_9, be one.

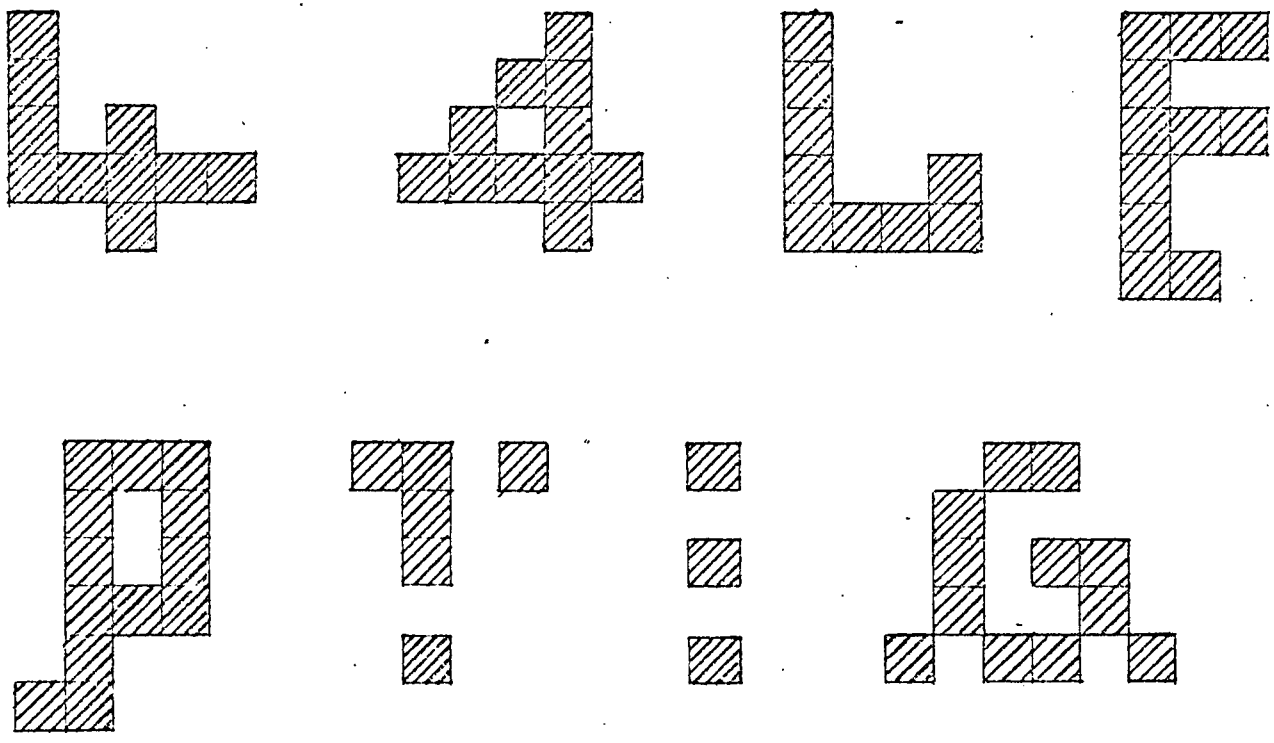
An immediate result of this optimizing process is that each letter is now insensitive to varying amounts of input "noise". Fig. 9 shows some typical configurations which would be correctly identified.

Since this character recognizer was designed solely on the basis of distinguishing any one of the twenty-three letters from the other twenty-two, it follows that its operation in the presence of an alien pattern would be largely unpredictable. For example, if the matrix is filled entirely with ones, a "5" would be identified. Other odd configurations may result in several characters being identified. Despite the fact that this feature seems to provide an unending source of amusement to persons experimenting with the model which has been built, it is clearly not desirable in a practical system. A simple way to largely eliminate this effect is to derive a set of properties common to all characters and prohibit identification of any pattern not possessing these properties. A typical set of

TABLE III

Alphá-numeric Weight and Threshold Matrix

Characters ↓	Characteristic Weights W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	W_9	W_{10}	W_{11}	W_{12}	T
1	0	0	1	0	-1	0	0	-1	0	1	0	1	3
2	1	0	1	0	0	0	1	-1	1	0	0	0	4
3	0	0	0	0	0	0	0	1	0	1	0	-1	2
4	1	-1	1	0	0	0	0	-1	0	0	0	1	3
5	0	0	0	-1	1	0	-1	0	1	0	0	0	2
6	1	0	1	0	0	0	0	0	1	0	-1	1	4
7	0	0	1	0	0	0	1	-1	-1	1	0	0	3
8	0	1	0	1	0	0	0	-1	0	0	0	1	3
9	1	0	1	1	0	0	0	0	-1	1	1	0	5
0	0	0	-1	0	0	0	0	1	1	0	0	0	2
A	0	0	-1	0	1	0	0	1	-1	0	0	0	2
C	0	0	0	-1	-1	-1	0	0	0	0	1	0	1
E	0	1	0	0	0	1	0	0	0	-1	0	1	3
F	1	0	0	0	0	-1	0	0	-1	0	1	0	2
G	0	0	1	1	0	0	0	0	1	1	1	1	6
H	0	-1	0	1	0	0	0	0	-1	1	0	1	3
J	0	1	0	0	0	0	0	1	0	-1	0	-1	2
K	1	-1	0	0	0	0	0	0	0	0	-1	1	2
L	0	0	1	0	-1	0	0	-1	1	0	0	1	3
P	0	0	0	-1	0	-1	-1	0	-1	1	1	0	2
T	0	-1	0	0	0	-1	0	0	0	-1	1	-1	1
U	2	0	0	0	0	0	1	-1	2	0	-1	2	7
X	0	-1	0	1	0	0	0	0	0	-1	0	-1	1



Some Recognizable Characters

Fig. 9.

properties for this case would be

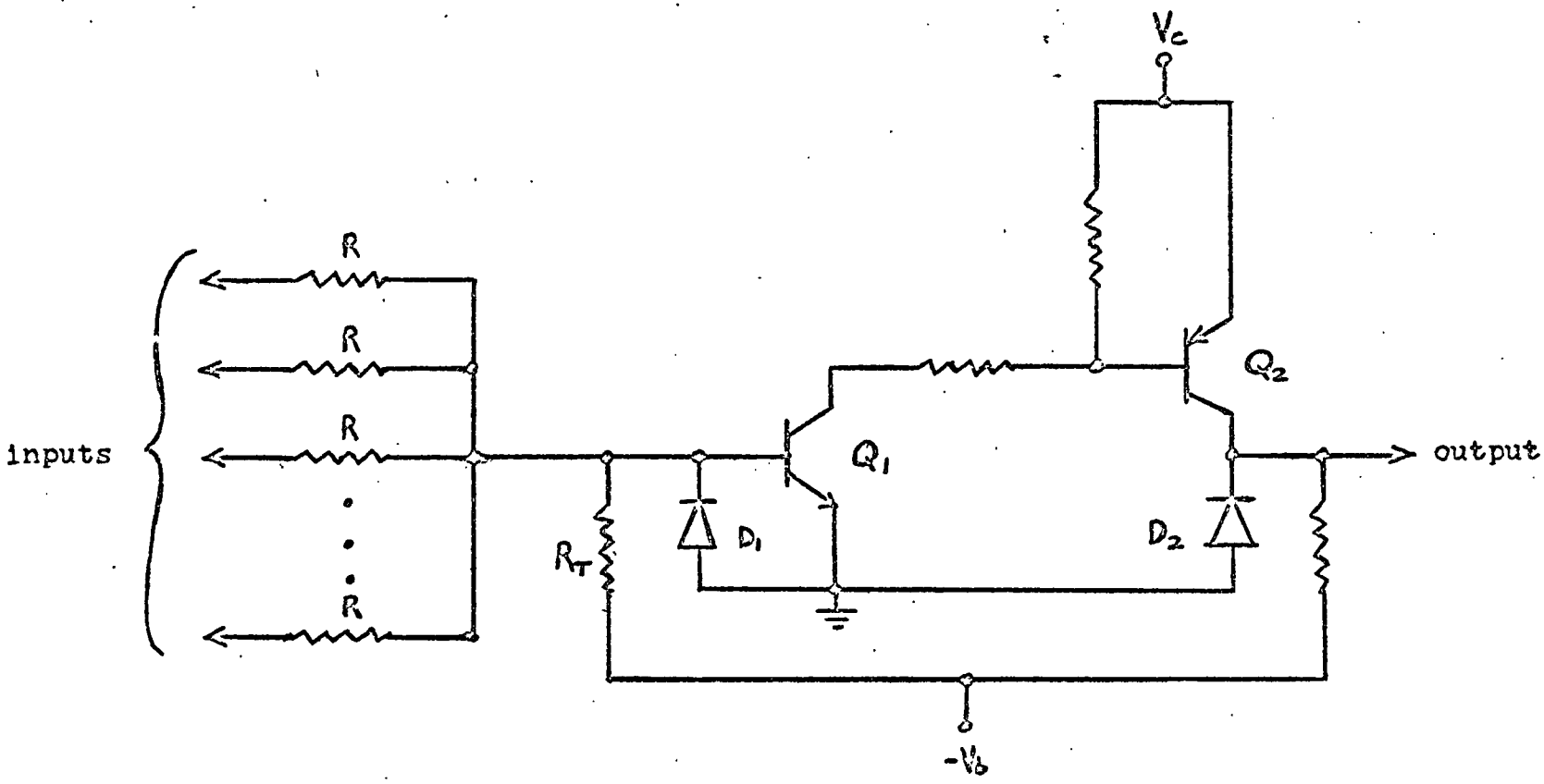
1. At least one vertical line must have three or more units.
2. At most two vertical lines can have four or more units.
3. At most three horizontal lines can have three or more units.
4. At least one edge must have three or more units.
5. At least one horizontal line must have one or two units.

Implementation of this additional logic would require more gates. For more sophisticated problems, considerably more restrictions may have to be imposed; such as a backwards "F" may be considered an alien pattern.

CHAPTER VI

CIRCUIT REALIZATION

Any circuit realization of a threshold logic gate must perform three basic functions: 1) a weighted summation of the inputs, 2) comparison of this sum with the threshold, and 3) generation of an output suitable for driving other gates. One such circuit that performs these three operations is shown in Fig. 10. The input resistors R permits the use of a simple Kirchhoff current adder for the weighted sum. The negative voltage $-V_b$ is the constant threshold voltage which is scaled by resistor R_t. Each input is either at $+V_c$, $-V_b$, or the clamp voltage, V_{D2}, of the circuit providing this input. The base of Q_1 may be characterized by two voltages: V_f, below which the transistor will be off, and V_o, above which it is saturated. These two voltages are the results of the comparison of the inputs to the gate threshold. Q_1 should be a high-quality silicon transistor with excellent repeatability of emitter-base characteristics and high gain at low current. For the case of a simple majority gate with n odd inputs and a threshold of n/2 the circuit values are chosen so that the voltage at the base of Q_1 will be less than V_f for (n-1)/2 "on" inputs and greater than V_o for (n+1)/2. Q_2 is chosed to be a germanium transistor with very



A Threshold Gate

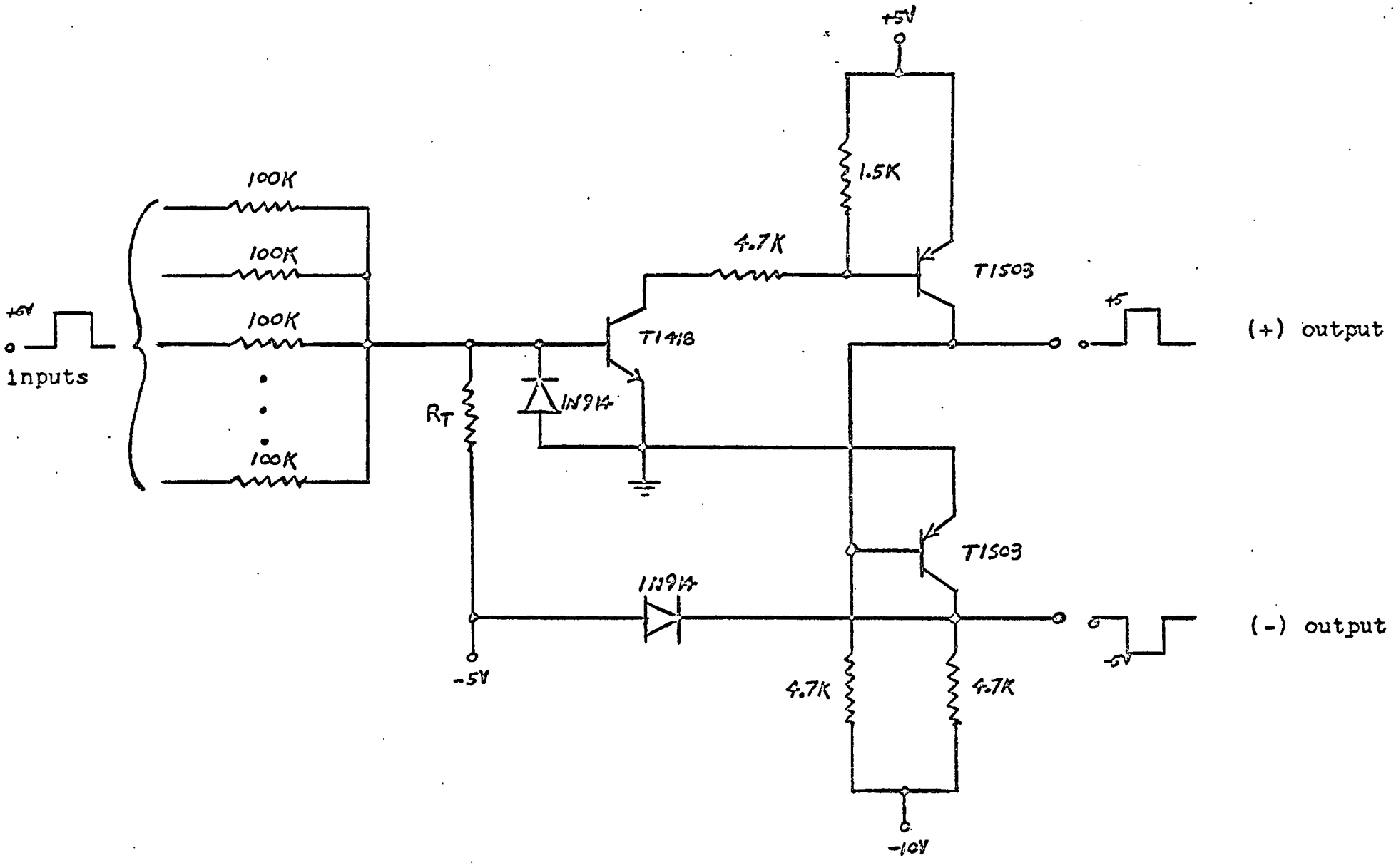
Fig. 10

low saturation resistance, therefore in combination with the clamp diode D_2 results in a very low impedance voltage source. Thus, the output voltage is independent of the load and there is no interaction between various stages fed from the same output. If a input to any gate calls for a negative weight, the source must be inverted from $+V_c$ to $-V_b$. This is accomplished by inverting the output of Q_2 as shown in Fig. 11. Typical performance characteristics of the circuit shown in Fig. 11 are given in Table IV. Switching time voltage waveforms are shown in Fig. 12.

To determine the correct value of R_t to achieve a desired threshold for a gate, the following steps were taken:

1. supplied +5V to the minimum number of inputs to produce the desired output.
2. ground the remaining inputs.
3. increase R_t until the desired output is produced.
4. then decrease the input voltage and vary R_t to such a value that one less input will not produce an output at an input voltage equal to +5V.

In Table V the experimental results of the above steps are shown. This table become vary useful in determining the required R_t to produce a desired threshold value of a gate.



A Threshold Gate Plus Output Inverter

Fig. 11.

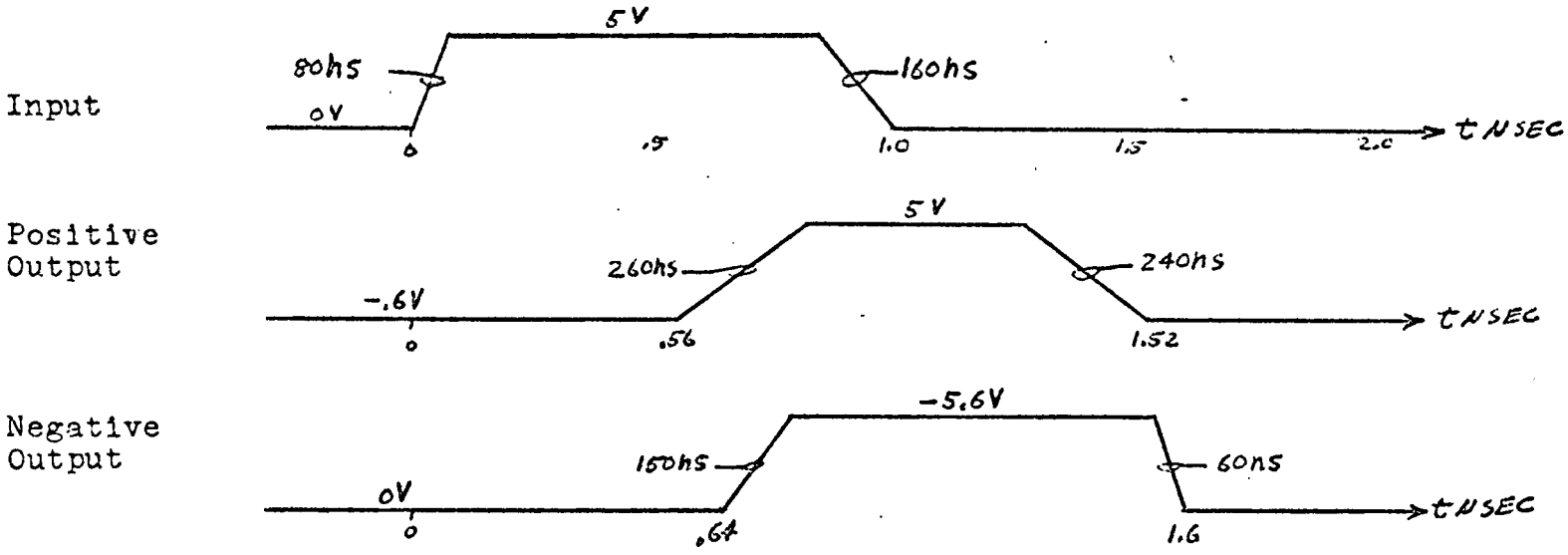
TABLE IV

Input	Positive Output	Negative Output
0V	-.6V	-.2V
+5V	+5V	-5.6V
cutoff frequency	2MHz	5MHz
output impedance	24 ohms	23 ohms
maximum loading	240 ohms	4K ohms

input impedance = $\frac{100K \text{ ohms}}{\#\text{ of inputs}}$

Typical Performance Characteristics of

a Threshold Gate of Threshold T=1



Input Signal: f = 1MHz, pw = 1usec., t_f = 160nsec., t_r = 80nsec.

		Positive Pulse	Negative Pulse	
			vs. input	vs. positive input
Delay Time	t_d	560nsec.	640nsec.	80nsec.
Rise Time	t_r	260nsec.	-	60nsec.
Storage Time	t_s	440nsec.	-	260nsec.
Fall Time	t_f	240nsec.	-	150nsec.
Propagation Delay	t_p		335nsec.	

Switching Time Voltage Waveforms

Fig. 12.

TABLE V

Values of R_t to Produce Desired Threshold

No. of "on" inputs	Minimum input to produce output	threshold equivalent	selected R_t	minimum input to produce output with one less input
1	3.6V	1	220K	-
2	3.8V	2	100K	6.9V
3	4.3V	3	56K	6.1V
4	4.6V	4	39K	5.8V
5	4.6V	5	30K	5.7V
6	4.6V	6	25K	5.4V
7	4.5V	7	22K	5.3V
8	4.8V	8	18K	5.4V

CHAPTER VII

CONSTRUCTION OF A SMALL VERSION OF THE PATTERN RECOGNITION CIRCUIT

Using the threshold gate discussed in Chapter VI, a small version of the pattern recognition circuit was constructed. The device will recognize only a selected group of characters, the numeric characters. The input matrix was reduced to 5 x 5, resulting in a simplified logical design requiring 84 threshold gates with maximum fan in of 20.

In the design only ten of the twelve invariant characteristics were used; they are C_1, C_2, C_3, C_4, C_7, C_8, C_9, C_{10}, C_{11}, and C_{12}. By investigating Table II, it is seen that columns C_1 and C_6 are identical for the numeric characters. Thus, it is not necessary to generate both characteristics to logically distinguish one numeric character from another. Therefore either one may be removed. In this design column C_6 was chosen to be removed. Also invariant characteristic C_5 was removed because it produced very little information about the numeric characters and would require a gate for each row and column employed to generate this characteristic. When these two characteristics C_5 and C_6 are removed, Table VI results. Inspection of this table now shows that these ten

TABLE VI

Numeric Characteristic Matrix

Characters ↓	Characteristics C_1	C_2	C_3	C_4	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
1	0	0	1	0	0	0	0	1	0	1
2	1	-	1	-	1	0	1	-	-	0
3	1	1	1	1	-	1	1	1	1	0
4	1	0	1	1	1	0	0	0	0	1
5	1	1	1	0	0	-	1	1	1	1
6	1	1	1	1	0	1	1	1	0	1
7	0	0	1	-	1	0	0	1	-	0
8	1	1	0	1	0	0	1	1	1	1
9	1	1	1	1	0	1	0	1	1	1
0	1	1	0	1	0	1	1	1	1	1

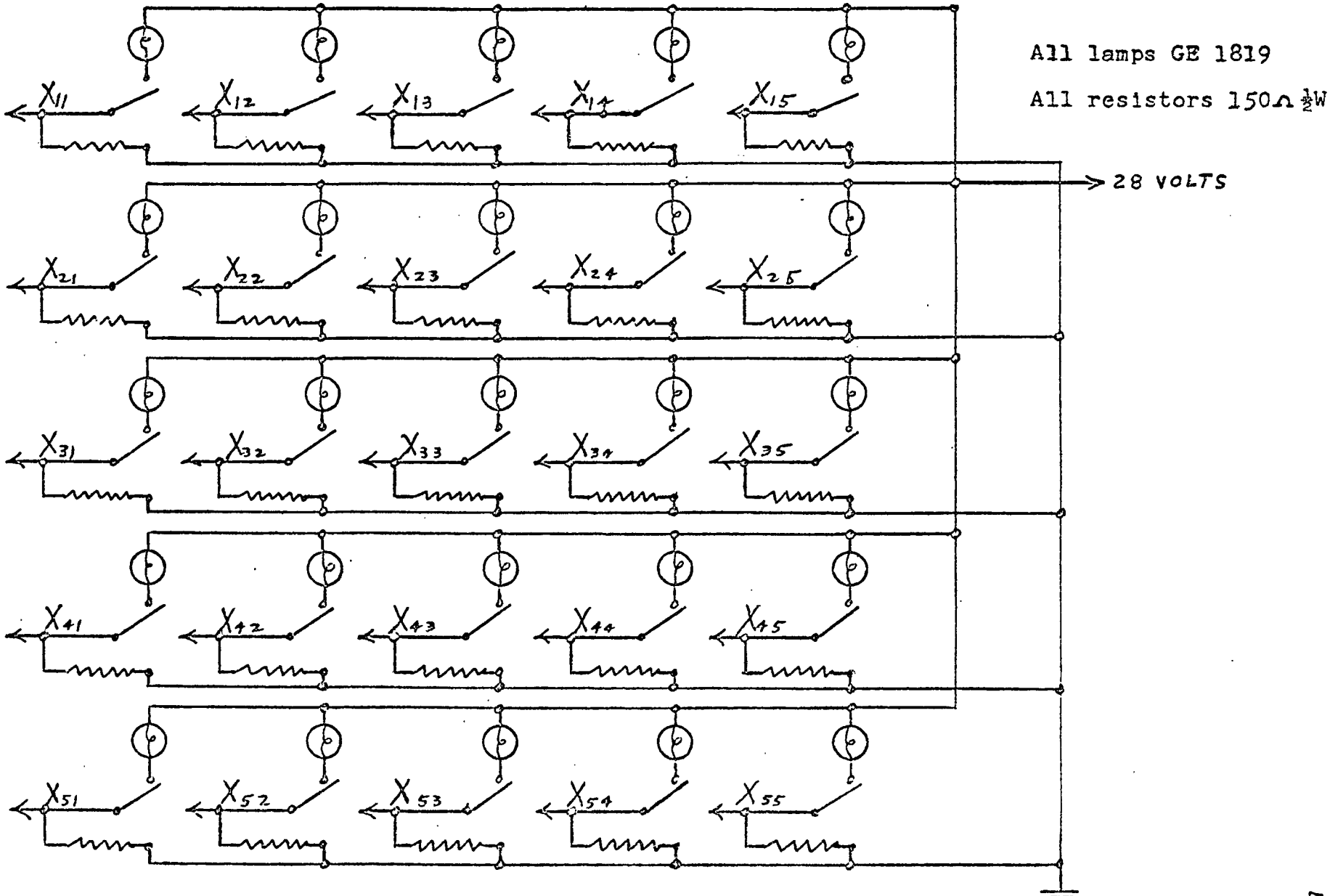
characteristics are sufficient to distinguish any of the ten numeric characters from any other. After optimization (see Appendix II), the weights and thresholds of the ten gates of the categorizer are found to be those shown in Table VII.

Now there is enough information to design and construct the receptor and categorizer but nothing has been said about the implementation of the sampling space, i.e., the matrix. The sampling space has two important functions: 1) it must provide binary data to the receptor, and 2) display the pattern to be recognized. The pattern display is accomplished by five rows and five columns of lamps. Each lamp is connected in series with a toggle switch. One side of the switch is tied to the lamp, the other to ground. As seen in Fig. 13, the common of the switch is connected through a resistor to ground. Thus, common of the switch will provide the binary data to the receptor. So the switch controls the pattern display and the binary inputs to the receptor. The binary input levels are adjusted by the resistor in series with the lamp. The resistors are adjusted to provide +5 volts to the receptor when the lamps are conducting. Fig. 14 shows the proper connections between the sampling space and the horizontal and vertical counters of the receptor. The outputs of these counters are fed into other threshold gates, as

TABLE VII

Weights of Gates of The Numeric Categorizer

Characters ↓	Characteristic Weights										Threshold
	W_1	W_2	W_3	W_4	W_7	W_8	W_9	W_{10}	W_{11}	W_{12}	T
1	0	-1	1	0	0	-1	0	1	0	1	3
2	1	0	1	0	1	-1	1	0	0	0	4
3	0	0	0	0	0	1	0	1	0	-1	2
4	1	-1	1	0	0	-1	0	0	0	1	3
5	0	0	0	-1	-1	0	1	1	0	0	2
6	1	0	1	0	0	0	1	0	-1	1	4
7	0	0	1	0	1	-1	-1	1	0	0	3
8	0	1	0	1	0	-1	0	0	0	1	3
9	1	0	1	1	0	0	-1	1	1	0	5
0	0	0	-1	0	0	1	1	0	0	0	2

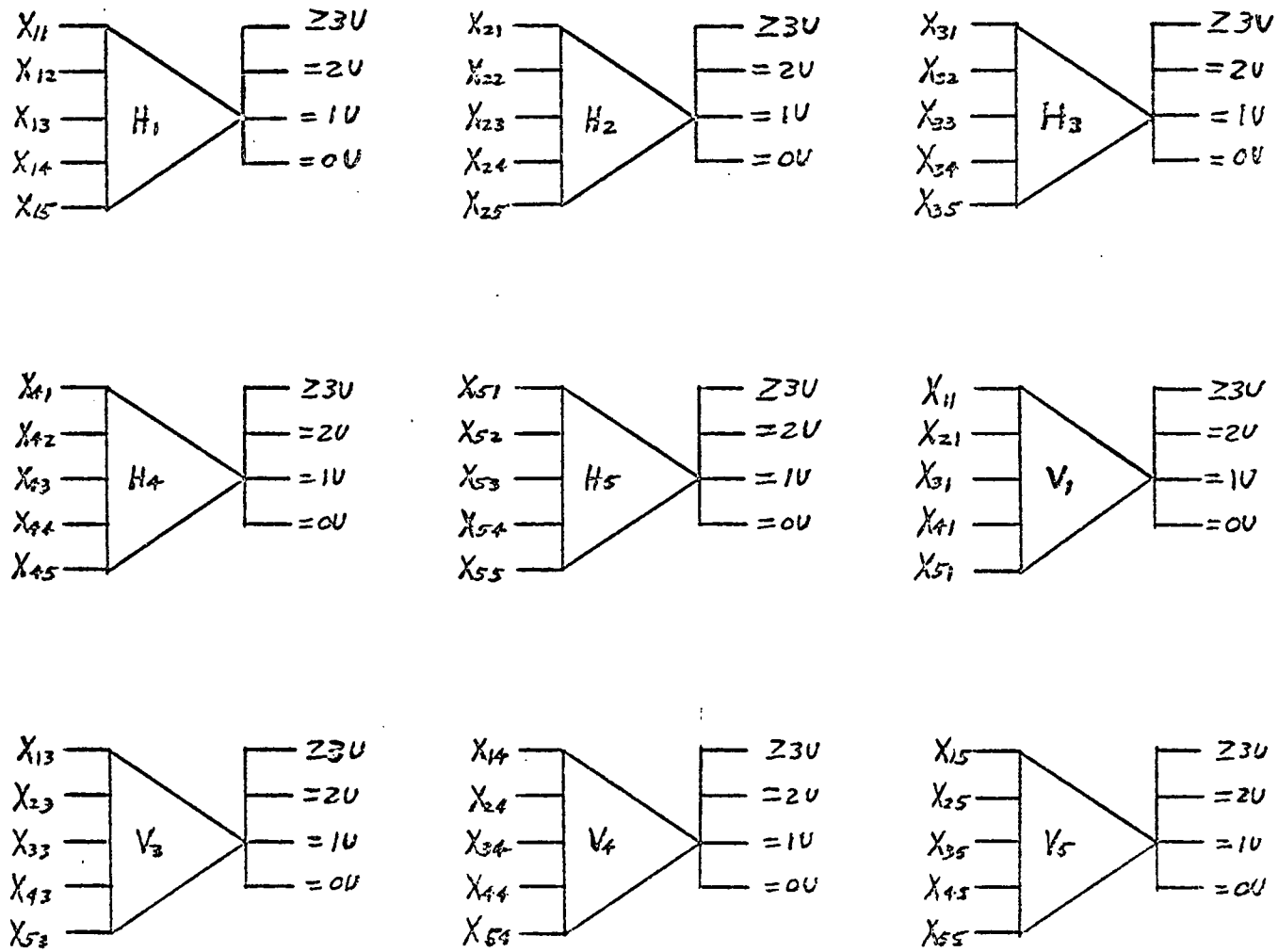


Toggle Switch Control Of Display And Binary Input To Receptor

Fig. 13.

	V_1	V_2	V_3	V_4	V_5
H_1	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}
H_2	X_{21}	X_{22}	X_{23}	X_{24}	X_{25}
H_3	X_{31}	X_{32}	X_{33}	X_{34}	X_{35}
H_4	X_{41}	X_{42}	X_{43}	X_{44}	X_{45}
H_5	X_{51}	X_{52}	X_{53}	X_{54}	X_{55}

Sampling Space



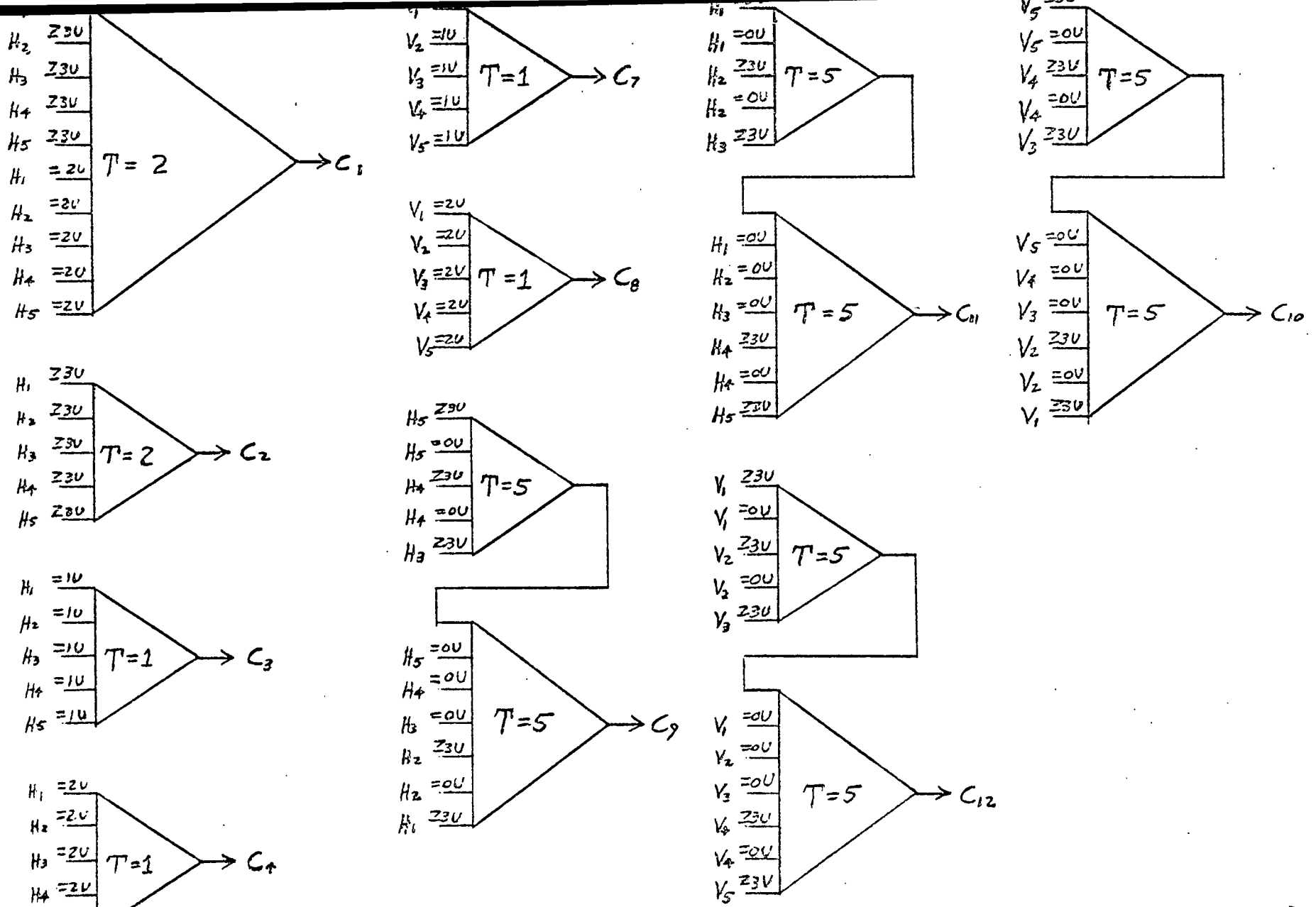
Vertical And Horizontal Counters

Fig. 14.

shown in Fig. 15, to generate the ten invariant characteristics. Finally, with reference to Table VII, the ten categorizing characters are generated, see Fig. 16. To display the recognition of the pattern of the sampling space, the output of each categorizing gate drives a transistor switch that controls the on and off conditions of a lamp. The on condition of a lamp indicates the pattern displayed on the sampling space has been recognized to be the character assigned to the lamp.

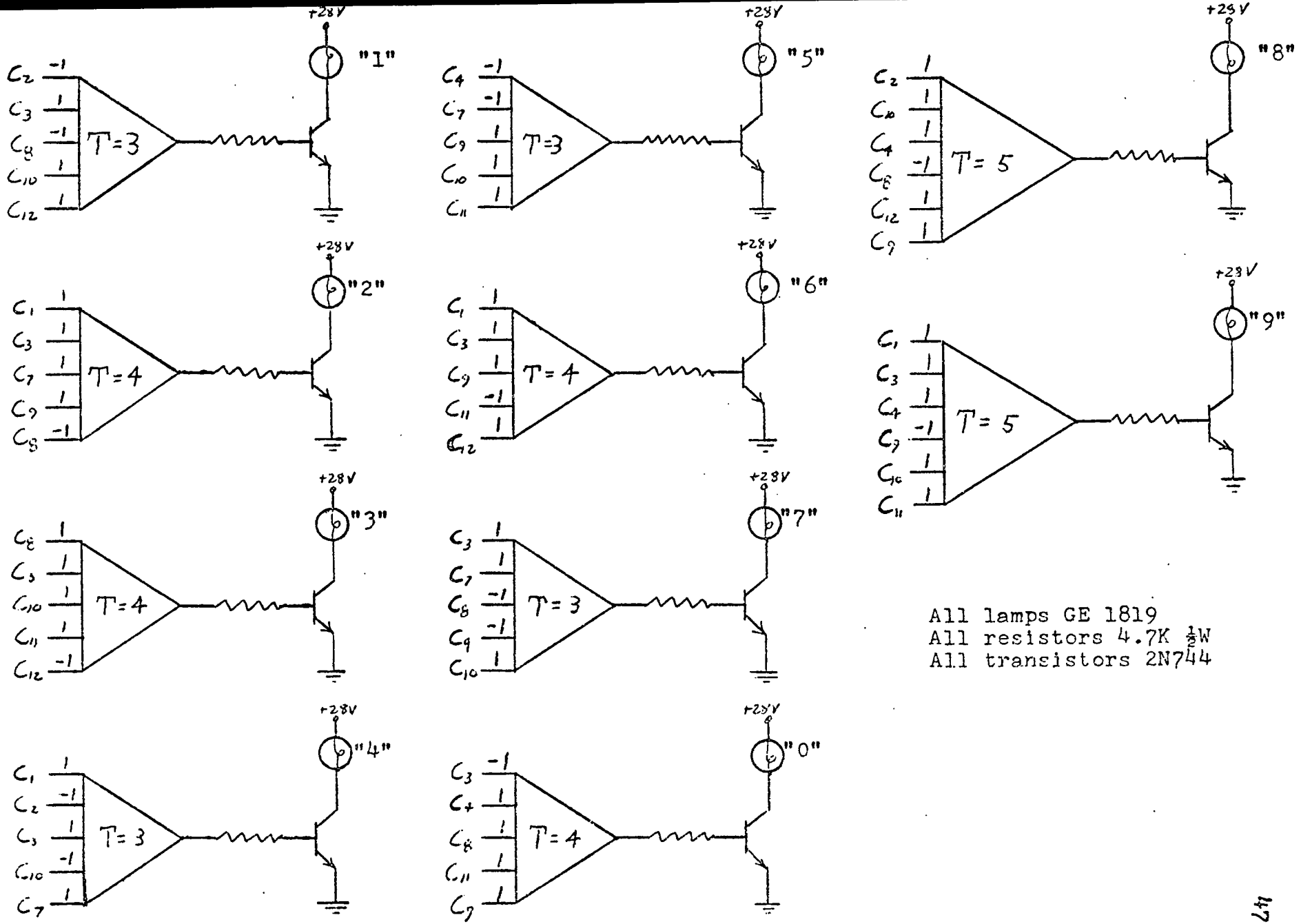
The sampling space of the device designed and built is shown in Fig. 17. In the top right of the picture is the sampling space; the left half is a 5 x 5 matrix of toggle switches that controls the on and off conditions of the lamps of the sampling space. Lamp indicators that represent the recognition result of the device are in the lower right half of the picture. The threshold logic circuits employed in the device are shown in Fig. 18.

A power supply was not built for this device; only the d.c. supplies available in the research lab were used. The d.c. power requirement for the sampling space is 28 volts d.c. @ 1 amp. Voltages required by the threshold logic are +5 volts, -5 volts, and -10 volts, non of which require more then 200 ma of current. All voltages must be 3% or better regulated.



Generation Of The Invariant Characteristics

Fig. 15.

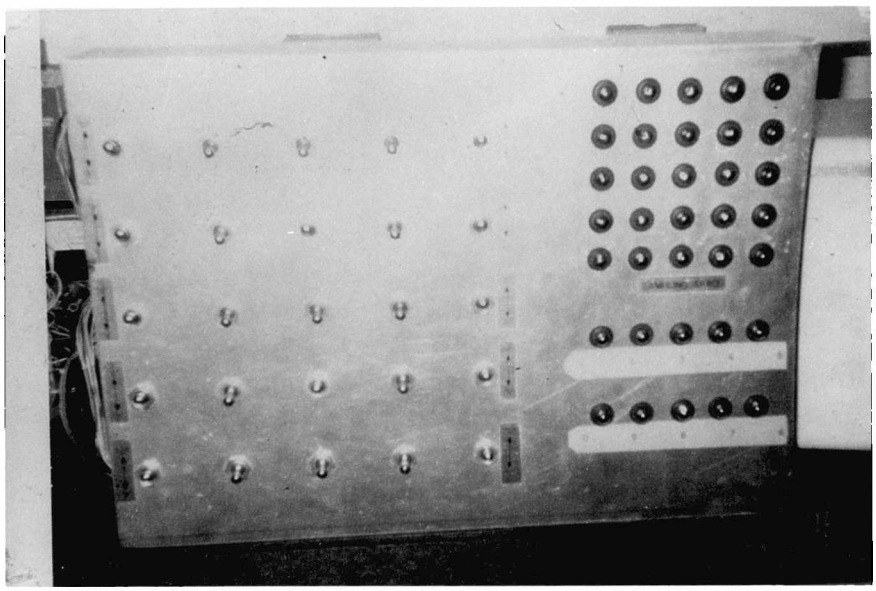


All lamps GE 1819
All resistors 4.7K $\frac{1}{2}$W
All transistors 2N744

47

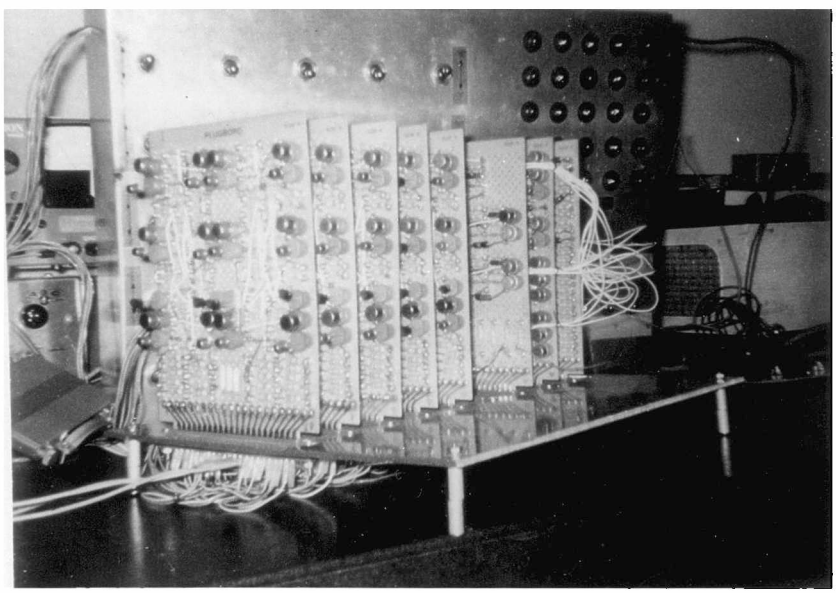
The Ten Categorizing Gates And Lamp Indicators

Fig. 16.



The 5 X 5 Sampling Space

Fig. 17.



Threshold Circuits Employed In The
Pattern Recognition Device

Fig. 18.

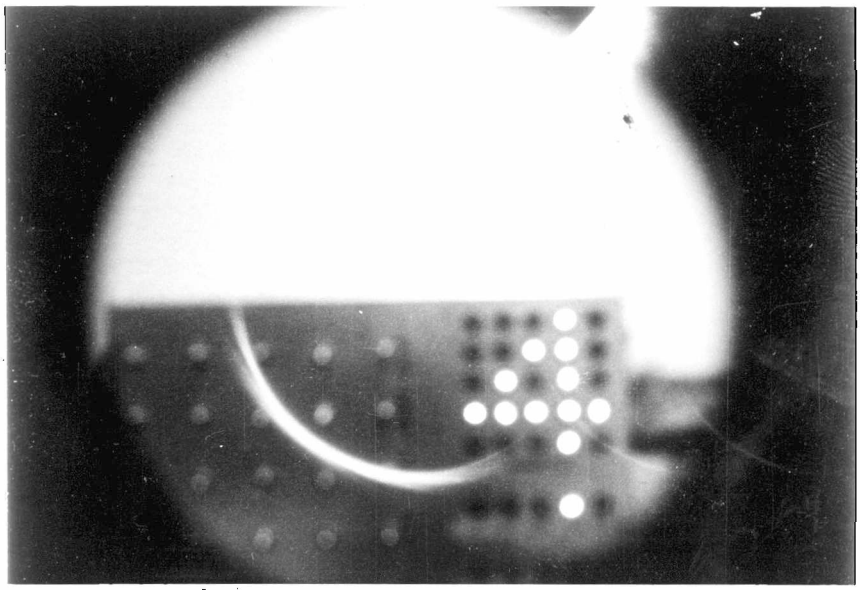
CHAPTER VIII

OPERATIONAL RESULTS OF THE PATTERN RECOGNIZER

Since the pattern recognizer was designed solely on the basis of distinguishing any one of the ten numeric characters from the other nine, it follows that its operation in the presence of an alien pattern would be largely unpredictable as discussed in Chapter V. With a 5 x 5 sampling space there are a possible 2^{25} or 33,554,432 patterns that can be displayed on the sampling space matrix. Of these patterns more than 400 of them were recognized as one of the ten numeric characters. Therefore the pattern recognizer was insensitive to varying amounts of input noise (alien patterns).

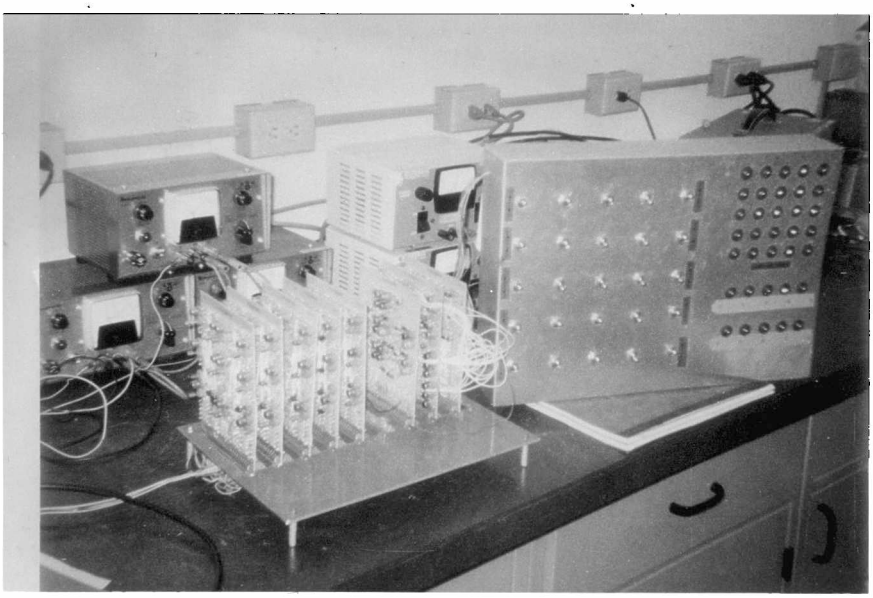
The recognition capability of the device is best illustrated by the figures 19 through 32 shown on the following pages. Fig. 19 is given mainly to show the relative position of a pattern on the sampling space and the categorizing indicators. The numeric "4" is displayed on the sampling space and the fourth lamp from the left, below the sampling space, indicates that the pattern on the sampling space has been recognized as a "4". The pattern recognizer device is shown in Fig. 20.

Because of the many variations of the numeric characters



A Closed Four Configuration And Its Categorizing Lamp Display

Fig. 19.

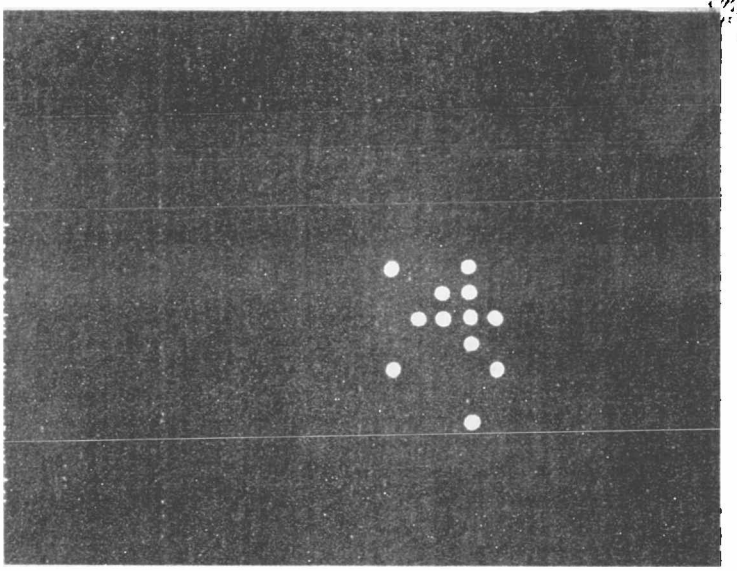


The Pattern Recognizing Device

Fig. 20.

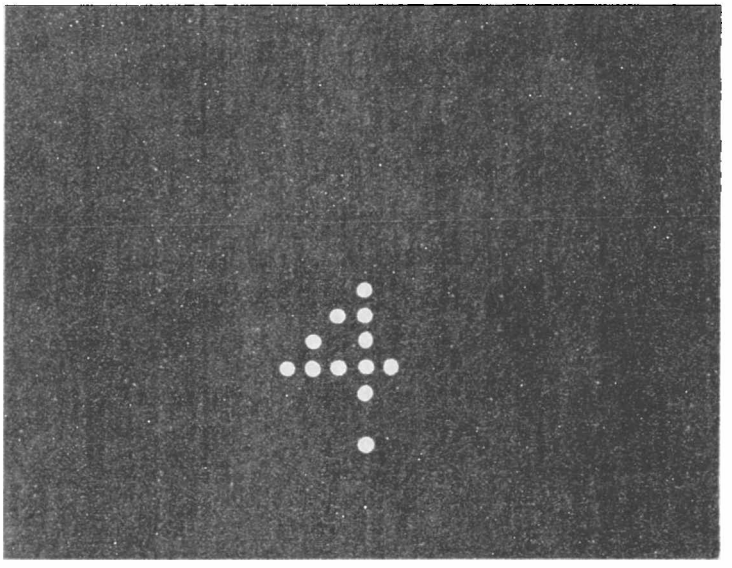
that would be recognized by the device, one particular numeric character, the "4", was chosen to illustrate the recognition capability of the device. Figures 21 through 26 are pattern variations of the closed four. Recognizable patterns of the open four are given in figures 27 through 32. These figures illustrate the capability of the recognizer to identify a character under any degree of translation, stretching, or compression within the confines of the sampling space. Also various distorted and rotated character configurations are recognized correctly. Other odd configurations may result in several characters being identified.

The fact that the device's operation in the presence of an alien pattern is largely unpredictable seems to provide an unending source of amusement to persons experimenting with this model. But it is not desirable in a practical system.



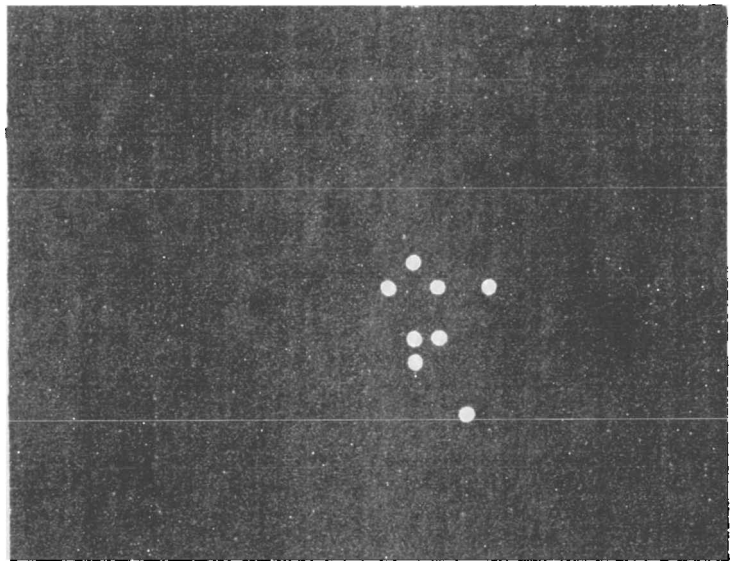
A Closed Four Configuration Plus "Noise"

Fig. 21.



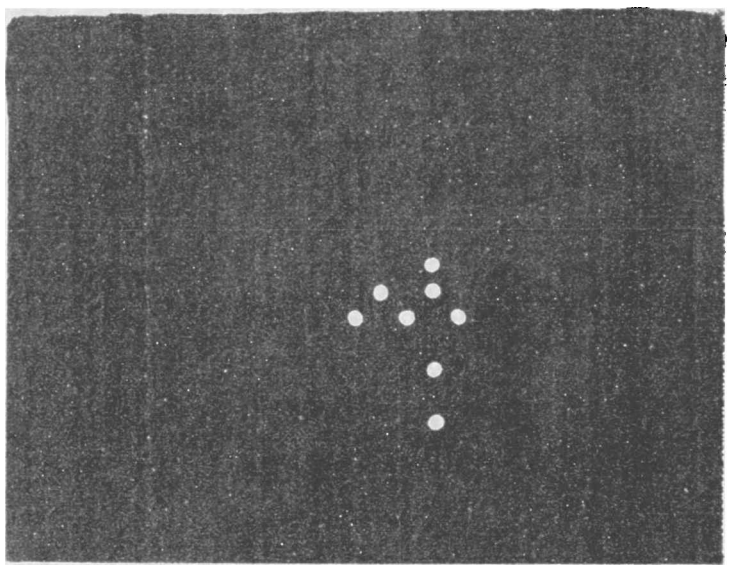
An Expanded Closed Four Configuration

Fig. 22.



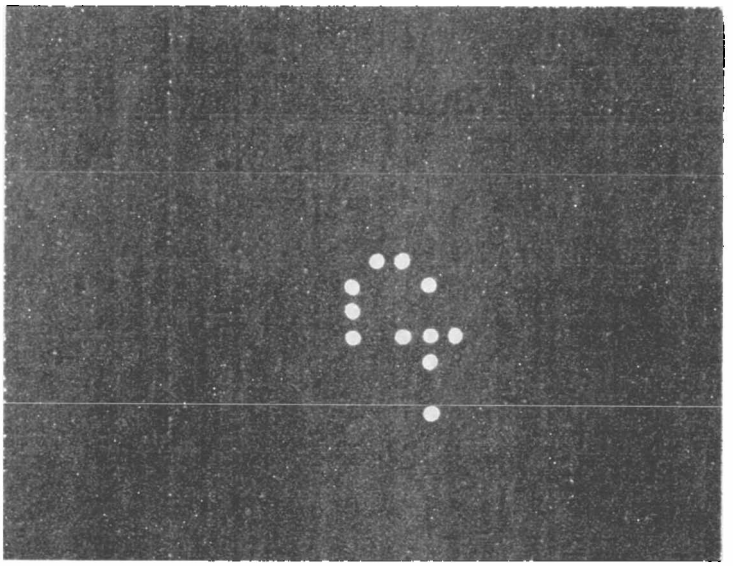
A Closed Four Configuration Rotated 180 Degrees

Fig. 23.



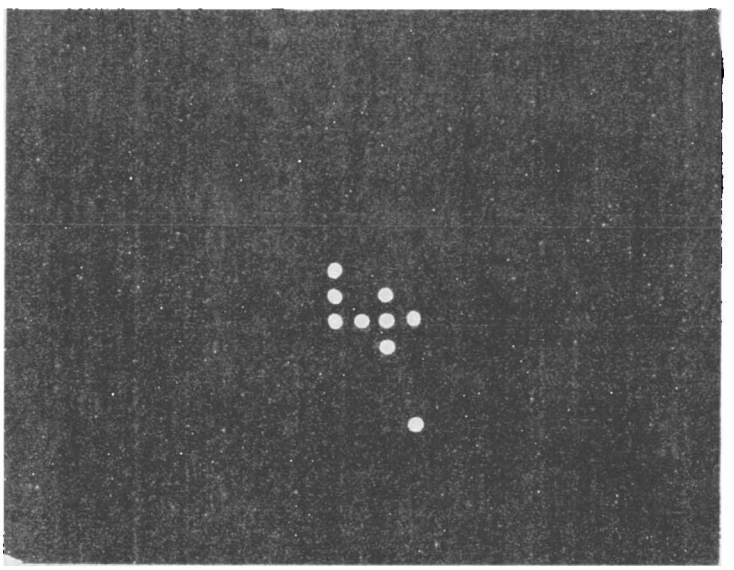
A Distorted Closed Four Configuration

Fig. 24.



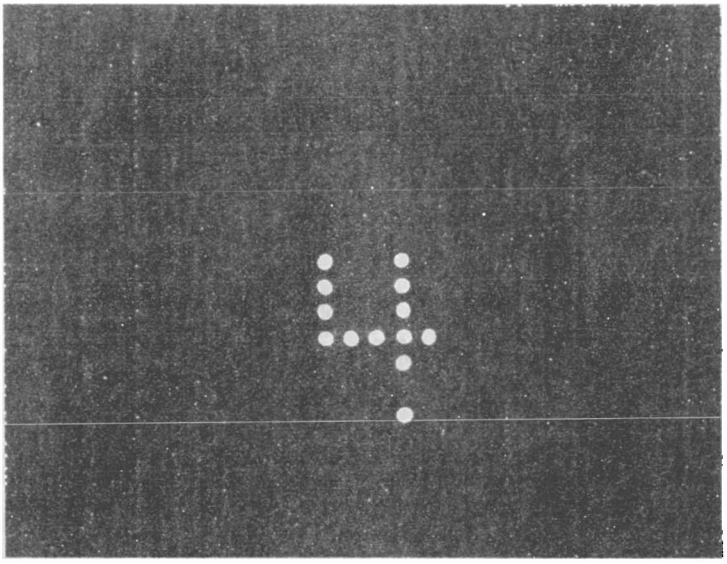
A Distorted Closed Four Configuration

Fig. 25.



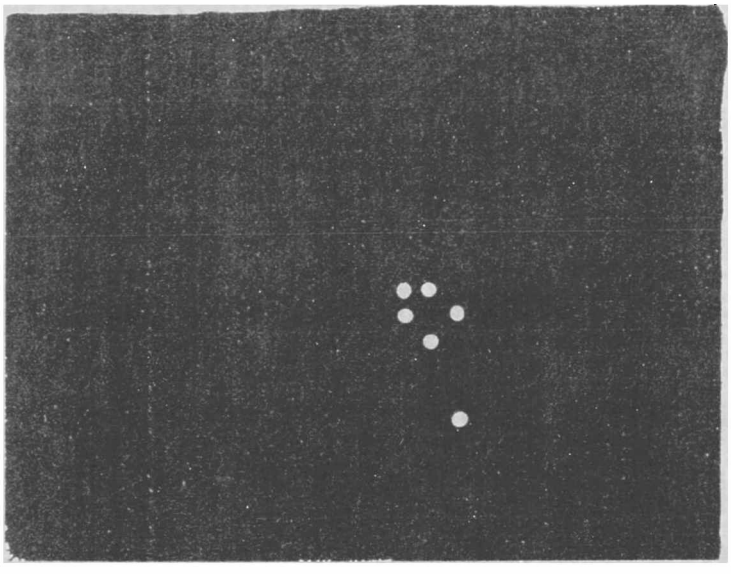
An Open Four Configuration Translated

Fig. 26.



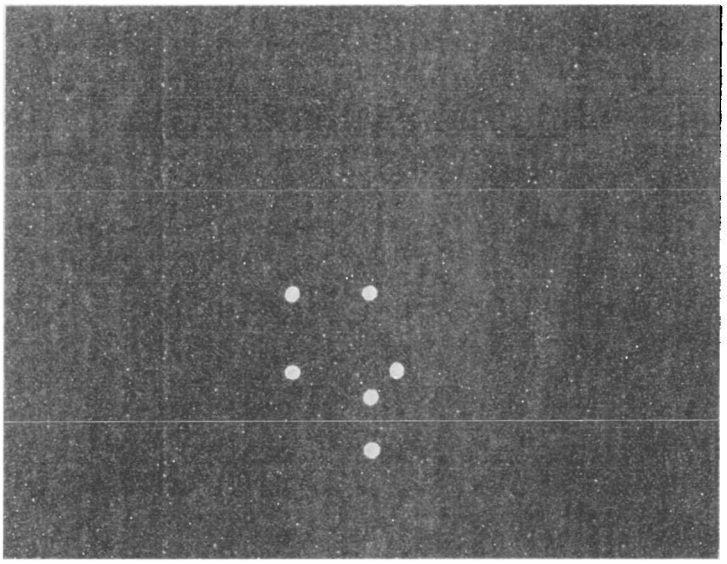
An Expanded Open Four Configuration

Fig. 27.



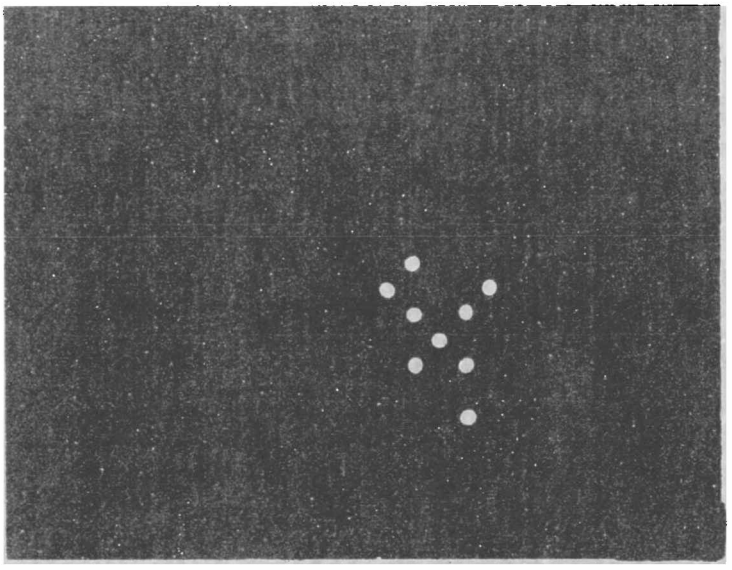
A Compressed Four Configuration Translated

Fig. 28.



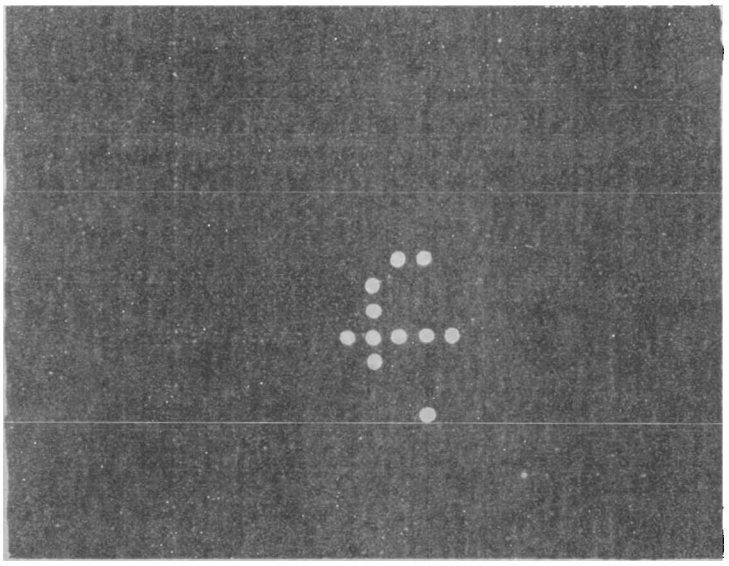
The Expanded Four Configuration Of Fig. 28

Fig. 29.



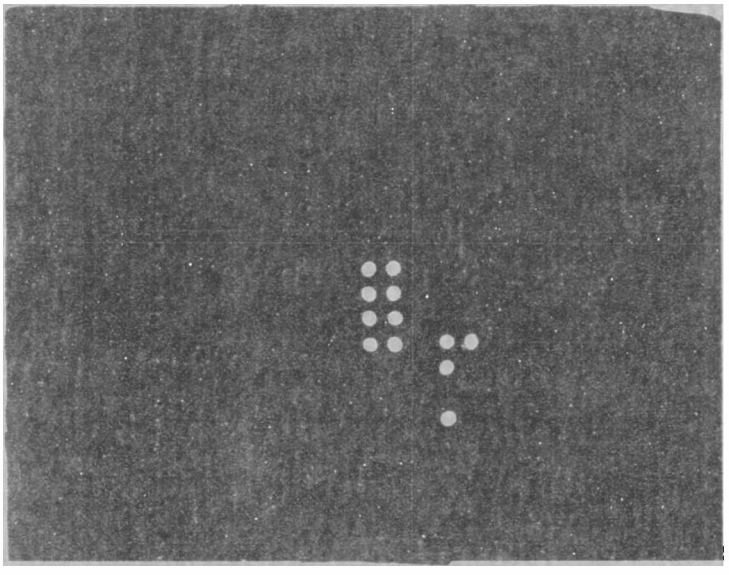
An Open Four Configuration Rotated 45 Degrees Clockwise

Fig. 30.



An Open Four Configuration Rotated 90 Degrees Clockwise

Fig. 31.



A Distorted Open Four Configuration

Fig. 32.

CHAPTER IX

ROTATION AND MAGINFICATION

In all of the foregoing discussions and in the design of the pattern recognizer, recognition has been limited to characters resulting from translation, stretching, or compression. The identification of rotated numeric "4" configurations, as shown in Chapter VIII, was purely incidental and not the result of specific design. Hence, a natural question arises concerning the extent to which translation, stretching, and compression techniques will extend when transformations involving rotation and magnification are permitted. Certainly more complex circuitry will be required, but in both cases promising approaches are available.

With regard to rotation, the problem becomes that of somehow evaluation the "horizontal" and "vertical" characteristics of a letter even though the corresponding lines of the character are not truly horizontal or vertical. One way around this dilemma would be to count the units, not in a single row but in a "block" or rows. Thus, rather than counting the number of units in row 1, the number in row 2, etc., a count could be made of the total number of units in, say, rows 1 through 4, the number in rows 2 through 5, the number in rows 3 through 6, etc. This would permit a

horizontal line of a given letter to be tilted at various angles without significantly affecting the number of units in the block in which it appears. A similar approach could be used for the vertical lines.

For cases where extreme degrees of rotation are encountered, other methods would be needed. One possibility would be to not only count the units in the individual horizontal and vertical lines of the matrix as before, but also to include counts of the units in the 30° lines, the 60° lines, the 120° lines, etc. Again, the degree of complexity would be dictated by the problem under consideration.

For recognition under various degrees of magnification, an important "scaling" property of the basic counting circuit (Fig. 3) can be employed. As given, the 7-input circuit of Fig.3 will have an output of 000 when no units appear on the inputs, 001 when one unit appears, 010 when 2 units appear, etc. Now assume that the number of inputs is increased to 21 and the other weights and thresholds are multiplied by 3. It will be seen that the circuit will have an output of 000 when 0, 1, or 2 units appear on the input, 001 when 3, 4, or 5 units appear on the input, 010 when 6, 7, or 8 units appear, etc. In general, if the number of inputs is increased or decreased to $K \cdot 2^{n} - 1$ and the other weights and thresholds are multiplied by K, then a corresponding linear scaling of the

range of the counter results.

Implementation of this feature in an actual recognition device would depend on the skill and requirements of the designer. Note that both the horizontal and vertical dimensions can be scaled separately since each gate of the receptor is concerned only with measurements along a single dimension. Such scaling could be under manual control or could conceivably be controlled automatically by measurements of the actual dimensions of the character.

CHAPTER X

CONCLUSION

The primary purpose of this thesis has been to illustrate some of the diverse applications of threshold logic to character recognition. The recognition problem considered was chosen for precisely this purpose. Actual character recognition problems are certainly much more complex and their solutions correspondingly more difficult. Likewise, other more standard character recognition procedures have deliberately been ignored. Such established techniques as curve following, waveform matching, positioning, vector crossings, moments, etc., certainly make vital contributions in their respective areas. Thus, the role of threshold logic would appear to be that of supplementing rather than replacing these existing techniques.

Certainly other more sophisticated applications of threshold logic can be made beyond those shown here. Particularly when more letters are to be recognized, additional characteristics would be needed for complete recognition. In this regard threshold logic should prove useful for measurements involving dimensions of a letter, its area, the comparative length of the lines of the letter, the existence of "holes", etc. In fact, many of these

characteristics may be of more value than some of those which were rather arbitrarily selected in this thesis.

Thus, threshold logic would appear to be a potentially powerful adjunct to existing techniques for character recognition. Its applications extend to both the receptor and categorizer portions of a recognition system, and the number of gates necessary to exploit its advantages does not appear prohibitive.

CITED REFERENCED

1. T. Marill and D.M. Green, "Statistical Recognition Functions and the Design of Pattern Recognizers," IRE Trans. On Electronic Computers, vol. EC-9, pp. 472-474; December, 1960.

2. P. Kaszerman, "An Introduction to Threshold Devices and their use in Pattern Recognition," Air Force Office of Scientific Research, Tech. Rept. 400-44, p 52; August 1961.

3. S. Muroga, I. Toda, and S. Takasu, "Theory of Majority Decision Elements," Journal of Franklin Inst., vol. 271 pp. 377-381; May, 1961.

4. W. H. Kautz, "The Realization of Symmetric Switching Functions With Linear-input Logical Elements," IRE Trans. On Electronic Computers, vol. EC-10 pp. 371-378; September, 1961.

5. S. B. Akers and B. H. Rutter, "The Application and Realization of Threshold Logic," Proc. 1963 International Solid-State Circuits Conf.; February, 1963.

6. S. B. Akers, "Threshold Logic and Two-Person, Zero-Sum Games," Proc. AIEE Symp. On Switching Circuit Theory and Logic Design; pp. 27-33; September, 1961.

7. Akers and Rutter, op. cit. pp. 277-279.

BIBLIOGRAPHY

Akers, S. B. "Threshold Logic and Two-Person, Zero-Sum Games," Proc. AIEE Symp. on Switching Circuit Theory and Logic Design; pp. 27-33; September, 1961.

Akers, S. B. and B. H. Rutter. "The Application and Realization of Threshold Logic," Proc. 1963 International Solid-State Circuits Conf.; February, 1963.

Highleyman,W. H. "Linear Decision Functions With Application to Pattern Recognition," Proc. IRE, vol 50, pp. 1501-1514; June, 1962.

Joyce, M. B. and K. K. Clarke. Transistor Circuit Analysis; Massachusetts: Addison-Wesley Publishing Co., Inc., 1964.

Kaszerman, P. "An Introduction to Threshold Devices and their use in Pattern Recognition," Tech. Rept. 400-44, New York University; Prepared for Air Force Office of Scientific Research; August 1961.

Kautz, W. H. "The realization of Symmetric Switching Functions With Linear-Input Logical Elements," IRE Trans. on Electronic Computers, vol. EC-10 pp. 371-378; September, 1961.

Luce, R. D. and H. Raiffa. Games and Decisions; New York: John Wiley and Sons, Inc., 1957.

Marill, T. and D. M. Green. "Statistical Recognition Functions and the Design of Pattern Recognizers," IRE Trans. on Electronic Computers, vol. EC-9, pp. 472-477; December, 1960.

Mattson, R. L. "An Approach to Pattern Recognition Using Linear Threshold Devices," Lockheed Aircraft Corp., Tech. Rept. LMSD-702680, ARTIA No. 246244; 1960.

Muroga, S., I. Toda, and S. Takasu. "Theory of Majority Decision Elements," J. Franklin Inst., vol. 271, pp. 376-418; May, 1961.

Paull, M. C. and E. J. McCluskey. "Boolean Functions Realizable With Single Threshold Devices," Proc. IRE, vol. 48, pp. 1335-1337; July, 1960.

Widrow, B. and M. E. Hoff. "Adaptive Switching Circuits," Stanford Electronics Lab., Calif., Tech. Rept. No. 1533-1; June, 1960.

APPENDIX I

IMPLEMENTATION OF EDGE CHARACTERISTICS

Consider the binary characteristic, C_9. It is to be one if and only there are at least three units in the bottom edge of the letter. Since the variables, h_{10} and h_{13}, are available from every row, it follows that C_9 may be expressed as

$$C_9 = h_{13} + h_{10}(h_{23} + h_{20}(h_{33} + h_{30}(h_{43} + \cdots .$$

This appears to be a rather complex function, but note that it is actually built up by alternately "and-ing" and "or-ing" each variable with all terms on its right.

A useful property of threshold logic is that if a function, $F(x_1, \cdots, x_n)$, can be realized with a single threshold gate, then for any variable, y, the functions $G = y \cdot F$ and $H = y + F$ may also be realized with a single threshold gate. For example, if $F(x_1, \cdots, x_n)$ is realized with a set of positive integral weights, $(w_1, \cdots, w_n)$, and a threshold, T, then $G(x_1, \cdots, x_n; y)$ may be realized with the weights $(w_1, \cdots, w_n; w - T + 1)$ and a threshold of $W + 1$ where $W = \sum w_i$. Likewise, $H(x_1, \cdots, x_n; y)$ may be realized with the weights $(w_1, \cdots, w_n; T)$ and the same threshold T. From this it follows that any function having the form of C_9 may be realized with a single threshold gate.

Table VIII shows how the weights and thresholds of a gate increase as additional variables are added. It will be seen that the weights increase according to the Fibonacci series, i.e., the series defined by the relations $f_1 = 1$, $f_2 = 1$, $f_i = f_{i-1} + f_{i-2}$. In general, with K variables, $w_i = f_i$, $T = f_K$ (K-odd) or f_{K+1} (K-even), and $W = f_{K+2} - 1$.

Note that when fan-in problems are encountered during implementation, the functions may be easily decomposed by simply building as large a gate as permissible and then replacing its output in the equation by a single variable and repeating the process. In particular, if the fan-in constraints on a gate are such that W + T must be less than some integer, $\emptyset$, it follows that one of the functions involving K variables can be implemented with about K + 1/ X - 4 gates where $f_X \leq \emptyset < f_{X+1}$. The maximum permissible fan-in is dependent upon several factors including component and supply tolerances, threshold uncertainty ($V_n - V_o$), transistor gain and breakdown voltage, and the maximum desired power dissipating per gate. Akers and Rutter make a detailed analysis which indicates that a majority gate with fan-in and fan-out of 51 should be feasible with presently available components, and such a gate has been constructed and operated successfully. With $\emptyset = 51$, $\emptyset$ is between $f_9 = 44$ and $f_{10} = 65$ so that X = 9. Since the generation of C_9

TABLE VIII

Function	T	w_1	w_2	w_3	w_4	w_5	w_6	W
X_1	1	1						1
$X_2 \cdot X_1$	2	1	1					2
$X_3+X_2 \cdot X_1$	2	1	1	2				4
$X_4(X_3+X_2 \cdot X_1$	5	1	1	2	3			7
$X_5+X_4(X_3+X_2 \cdot X_1$	5	1	1	2	3	5		12
$X_6(X_5+X_4(X_3+X_2 \cdot X_1$	13	1	1	2	3	5	8	20

Weights and Thresholds to Implement Edge Characteristics

requires two variables per row or a total of 2m variables, it follows that approximately 2m + 1/5 gates would be required for C_9 and likewise for C_{11}. Similarly, for C_{10} about 2n + 1/5 gates would be needed.

APPENDIX II

OPTIMIZING THE WEIGHTS OF CATEGORIZING GATES

Clearly, the weights of a categorizing gate should be choden to distinguish the character it represents from the other characters; i.e., with reference to Table II a set of weights is desired which will make the weighted sum of a particular row as large as possible while the weighted sum of any other row is as small as possible.

In order that only positive weights need be considered, the first step is to complement the columns of Table II which are 0 in the row of the character of interest so that this row becomes all 1's. Had any column been blank in this row, it would be removed since it will be of no value in distinguishing the character. But all other blanks are made 1 since it must be assumed that they will agree with the character of interest. Call this new table matrix $\left[C_{ij}\right]$.

Now the problem may be stated formally:

Find a set of weights, $(w_1,\cdots,w_{12})$, with $w_j = 0$ and $\sum w_j = 1$ such that

$$\sum w_j C_{kj} - \max_{\substack{i=1 \\ i\neq k}}^{23} \left[\sum w_j C_{ij}\right]$$

i - row index

j - column index

k - row of interest

is a maximum.

Now since

$$C_{kj} = 1, \qquad \sum_j w_j C_{kj} = 1,$$

the problem reduces to that of finding w's to minimize

$$\operatorname*{Max}_{\substack{i=1 \\ i \neq k}}^{23} \left[\sum w_j C_{ij}\right].$$

Finally, if $\left[C_{ij}\right]$ is complemented and the "k" row removed yielding the matrix $\left[C_{ij}^*\right]$, then the problem becomes: Find a set of weights, $(w_1, \cdots, w_{12})$, with $w_j \geq 0$ and $\sum w_j = 1$ such that

$$v = \operatorname*{Min}_i \left[\sum w_j C_{ij}^*\right]$$

is a maximum.

This is precisely the problem involved in game theory in finding the value v, of a two-person, zero-sum game. Hence, the known techniques for the solution of such games may be applied here. In particular, if one row of the matrix $\left[C_{ij}^*\right]$ dominates a second row, i.e., has ones every where the second does, it may be removed. Likewise, if a column is dominated it may be removed. When two rows or columns are identical, either one may be removed. Using this method an optimum set of weights may be generated for the categorizing gates.

In order to illustrate this procedure, the design of the "8 gate" of the categorizer will be followed in detail. Table II is duplicated here as Table A.

TABLE A

Alpha-Numeric Characteristic Matrix

Characters ↓	Characteristics C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
1	0	0	1	0	0	0	0	0	0	1	0	1
2	1	-	1	-	-	1	1	0	1	-	-	0
3	1	1	1	1	0	1	-	1	1	1	1	0
4	1	0	1	1	-	1	1	0	0	0	0	1
5	1	1	1	0	1	1	0	-	1	1	1	1
6	1	1	1	1	-	1	0	1	1	1	0	1
7	0	0	1	-	0	0	1	0	0	1	-	0
8	1	1	0	1	1	1	0	0	1	1	1	1
9	1	1	1	1	-	1	0	1	0	1	1	1
0	1	1	0	1	-	1	0	1	1	1	1	1
A	1	1	0	1	1	1	0	1	0	1	1	1
C	1	1	1	0	0	0	0	1	1	0	1	1
E	1	1	1	-	0	1	-	1	1	0	1	1
F	1	-	1	-	0	0	-	1	0	0	1	1
G	1	1	1	1	-	1	-	1	1	1	1	1
H	1	0	0	1	-	1	1	0	0	1	0	1
J	1	1	1	1	-	-	1	1	1	0	1	0
K	1	0	-	1	0	0	1	1	0	-	0	1
L	0	0	1	0	0	0	1	0	1	0	0	1
P	1	1	1	0	-	1	0	1	0	1	1	1
T	0	0	1	0	0	0	1	0	0	0	1	0
U	1	0	0	1	-	1	1	0	1	1	0	1
X	1	0	1	1	0	-	1	1	0	0	0	0

First complement the columns of this table which has seros in the "8" row. All blanks becomes ones. Call this matrix $[C_{ij}]$. The "8" row becomes all one's. Next complement $[C_{ij}]$ and remove the "8" row yielding the matrix $[C_{ij}^*]$. Now if one row of this matrix dominates a second row, i.e., has ones everywhere the second does, it may be removed. Likewise, if a column is dominated it may be removed.

Using just these rules, it is seen that all of the rows of the matrix except 2 dominate 0, 5, or row U. Hence, all but these four rows may be removed leaving the matrix shown in Table D. This means that if "8" can be logically distinguished from "2," "5," "0," and "U" it will automatically be distinguished from the other characters.

Finally, columns are eliminated, leaving the matrix of Table E. From this matrix the following equations are obtained.

$$w_{12}C_{12}^* = v$$

$$w_4\ C_4\ ^* = v$$

$$w_8\ C_8\ ^* = v$$

$$w_2\ C_2\ ^* = v$$

In addition we have

$$w_2 + w_4 + w_8 + w_{12} = 1.$$

TABLE B

Matrix $[c_{ij}]$

Characters ↓	Characteristics c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}
1	0	0	0	0	0	1	1	1	0	1	0	1
2	1	1	0	1	1	1	0	1	1	1	1	0
3	1	1	0	1	0	1	1	0	1	1	1	0
4	1	0	0	1	1	1	0	1	0	0	0	1
5	1	1	0	0	1	1	1	1	1	1	1	1
6	1	1	0	1	1	1	1	0	1	1	0	1
7	0	0	0	1	0	0	0	1	0	1	1	0
8	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	0	1	1	1	1	0	0	1	1	1
0	1	1	1	1	1	1	1	0	1	1	1	1
A	1	1	1	1	1	1	1	0	0	1	1	1
C	1	1	0	0	0	0	1	0	1	0	1	1
E	1	1	0	1	0	1	1	0	1	0	1	1
F	1	1	0	1	0	0	1	0	0	0	1	1
G	1	1	0	1	1	1	1	0	1	1	1	1
H	1	0	1	1	1	1	0	1	0	1	0	1
J	1	1	0	1	1	1	0	0	1	0	1	0
K	1	0	1	1	0	0	0	0	0	1	0	1
L	0	0	0	0	0	0	0	1	1	0	0	1
P	1	1	0	0	1	1	1	0	0	1	1	1
T	0	0	0	0	0	0	0	1	0	0	1	0
U	1	0	1	1	1	1	0	1	1	1	0	1
X	1	0	0	1	0	1	0	0	0	0	0	0

TABLE C

Matrix $[c_{ij}^*]$

Characters ↓	Characteristics c_1^*	c_2^*	c_3^*	c_4^*	c_5^*	c_6^*	c_7^*	c_8^*	c_9^*	c_{10}^*	c_{11}^*	c_{12}^*
1	1	1	1	1	1	0	0	0	1	0	1	0
2	0	0	1	0	0	0	1	0	0	0	0	1
3	0	0	1	0	1	0	0	1	0	0	0	1
4	0	1	1	0	0	0	1	0	1	1	1	0
5	0	0	1	1	0	0	0	0	0	0	0	0
6	0	0	1	0	0	0	0	1	0	0	1	0
7	1	1	1	0	1	1	1	0	1	0	0	1
9	0	0	1	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0
A	0	0	0	0	0	0	0	1	1	0	0	0
C	0	0	1	1	1	1	0	1	0	1	0	0
E	0	0	1	0	1	0	0	1	0	1	0	0
F	0	0	1	0	1	1	0	1	1	1	0	0
G	0	0	1	0	0	0	0	1	0	0	0	0
H	0	1	0	0	0	0	1	0	1	0	1	0
J	0	0	1	0	0	0	1	1	0	1	0	1
K	0	1	0	0	1	1	1	1	1	0	1	0
L	1	1	1	1	1	1	1	0	0	1	1	0
P	0	0	1	1	0	0	0	1	1	0	0	0
T	1	1	1	1	1	1	1	0	1	1	0	1
U	0	1	0	0	0	0	1	0	0	0	1	0
X	0	1	1	0	1	0	1	1	1	1	1	1

TABLE D

Dominant Row Matrix

	C_1^*	C_2^*	C_3^*	C_4^*	C_5^*	C_6^*	C_7^*	C_8^*	C_9^*	C_{10}^*	C_{11}^*	C_{12}^*
2	0	0	1	0	0	0	1	0	0	0	0	1
5	0	0	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0
U	0	1	0	0	0	0	1	0	0	0	1	0

TABLE E

Dominant Row-Column Matrix

	C_2^*	C_4^*	C_8^*	C_{12}^*
2	0	0	0	1
5	0	1	0	0
0	0	0	1	0
U	1	0	0	0

From these equations one finds the desired weights are

$$w_2 = w_4 = w_8 = w_{12} = \tfrac{1}{4}$$

with the rest zero, and the value of v is also $\frac{1}{4}$.

Applying these weights to the original matrix $\left[C_{ij}\right]$ it is seen that the weighted sum of the "8" row is now one and no other row has a value greater than 1 - v or 3/4. If the weights are now multiplied by a constant, k, so that they become integers, then the threshold may be chosen equal to k(1-v) + 1. In this case, the weights become all one with a threshold of 4.

Finally, each nonzero weight, w_i, which corresponds to a column which was complemented to obtain $\left[C_{ij}^*\right]$ matrix must be replaced by $-w_i$ and the threshold T decreased by w_i.

Thus, the final results is that the "8" gate of the categorizer will have inputs C_2, C_4, C_8, and C_{12} with weights of 1, 1, -1, and 1, respectively, and a threshold of 3. The weights of the other 22 categorizing gates are to be derived in this same manner.

APPENDIX III

LIST OF COMPONENTS USED IN PATTERN RECOGNIZER

ITEM DESCRIPTION	TYPE	QUANITY
Transistor	TI 418	72
"	TI S03	106
"	2N 744	10
Diode	1N 914	156
Capacitor	22uf @ 15V	12
"	15uf @ 20V	6
Resistors (all ½W/5%)	150	25
	1.5K	78
	4.7K	194
	22K	9
	33K	6
	39K	4
	43K	12
	47K	5
	56K	7
	68K	12
	100K	358
	180K	18
	220K	12
	330K	10
Switch	DPDT (2 position)	25
Lamp	GE 1819	35