AN ADAPTIVE CHARACTER RECOGNITION MACHINE

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

William Perry Simonds

January, 1970

ACKNOWLEDGEMENT

I would like to gratefully acknowledge the encouragement and assistance of my advisor, Dr. J. D. Bargainer.

My deepest appreciation goes to my wife, Lou, for her encouragement and understanding.

I would also like to thank Mrs. Linda Vasquez and Mr. Bill Price for their unselfish assistance in this project.

The research reported here was supported by the National Science Foundation Graduate Traineeship Program and by NSF Grant GP-1539.

AN ADAPTIVE

CHARACTER RECOGNITION MACHINE

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

William Perry Simonds

January, 1970

ABSTRACT

High speed digital computers perform arithmetic and logical functions at extremely fast rates and line printers, plotters, and cathode ray tube displays have been developed which can output results at comparable rates. The link in the man-machine interface which is now receiving much attention is the computer input. The speed of this link could be greatly increased by a character recognition machine which could convert hand lettered programs and data directly into computer input eliminating manual card punching.

One machine which appears to be applicable is a learning machine called the perceptron. This machine can be trained to distinguish between different optical patterns in a manner similar to the learning process of humans. The original perceptron built by F. Rosenblatt at Cornell Aeronautical Laboratory used a large array of photo-cells onto which the patterns to be recognized were projected. The outputs of these receptors were connected at random to the inputs of a set of fixed threshold gates whose outputs were connected through adjustable gain amplifiers to adaptive threshold gates. Training was achieved by adjusting the gain of the amplifiers and the thresholds according to a training algorithm.

In order to investigate the feasibility of the perceptron as a practical hand lettered alpha-numeric character recognition machine a digital version has been designed and constructed using binary numbers stored in a core memory and an accumulator in place of the adjustable gain amplifiers and threshold gates of the original perceptron.

Although this machine has severe limitations due to its size, tests run on the machine indicate that the concept is feasible and for any given capability the digital machine would be smaller, and more easily trained than the original analog version.

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

Digital computers have relieved man of many tasks such as arithmetic calculations and the handling of large amounts of data or rather provided man with the capability to do more of this work in less time. The time consuming task of reading and understanding is still primarily a manual task whether it be predicting weather from satellite photos of the atmosphere or sorting mail according to destination. Much research time and money is now being spent in this broad area of pattern recognition. Much of this research has been directed toward programming large general purpose digital computers to handle these tasks. It seems more practical, however, in some cases to build special purpose machines to handle the pattern recognition directly. It is the purpose of the research reported here to investigate the feasibility of one such special purpose machine.

The field of pattern recognition can be broken down into several smaller areas each with its own unique problems. Speech recognition and translation from one language to another require that not only words but their different meanings in different contexts must be recognizable. The mail sorting problem is extremely complicated due to the wide range of size and style of characters and their location on the piece of mail. It is undesirable, if not impossible, to require everyone to use one standard style, size, and location when addressing mail.

There is one area of pattern recognition which does not have these problems to the extent of those cases mentioned above. In many cases, the procedure used by people needing to get information, whether programs or data, into a computer is to print the information on preprinted forms for this purpose with blocks corresponding to punched card columns marked on the forms. These forms are then delivered to key punch operators who transfer the information to standard punched cards. These users are already accustomed to using block lettering in specific fields thus simplifying the problem of designing a machine to transfer the information character by character from the printed forms to computer input format such as punched cards, magnetic tape or even directly into the computer. The purpose of the machine which is the subject of this thesis is to recognize hand printed alpha-numeric characters located in a well defined field.

The next problem encountered by the designer of such a machine is whether the machine should be of fixed or adaptive design. If the characters to be recognized were type written or machine printed, always using the same size and style of print, a fixed design machine would probably suffice. In the problem considered here however, even with the restrictions already mentioned, there is still enough variation in style, size, and location to indicate the necessity of an adaptive machine which could be trained to recognize the characters.

The purpose then of the research reported here is to design, construct and test an adaptive machine for the classification of hand printed alpha-numeric characters in a confining field for the purpose of determining how much variation in size, orientation, and style the machine can tolerate and still classify correctly.

CHAPTER II

THE PERCEPTRON

The principle of the perceptron was originally developed by F. Rosenblatt[5] as a model to demonstrate one theory of the mechanism used in the brain for learning to recognize visual patterns. This theory being that at birth the human nervous system contains on the order of 10^{10} randomly connected neurons and that learning and retention is implemented by the formation of connections or pathways between centers of activity.[4] The block diagram of Figure 2.1 shows the organization of a perceptron. The sensors (S-units) represent the rods and cones in the retina of the eye. These are randomly connected to a layer of association cells (A-units) which are models of neurons each of which fires on an all or nothing basis if the number of excited S-units connected to it exceeds some fixed number. The output of each A-unit is connected to an input of each response (R-unit) which is a more elaborate model of a neuron. Each input has an adjustable weight associated with it and the operation is such that if the summation of the weighted inputs is greater than some adjustable threshold level the unit produces a response. Training of the perceptron can follow any of several algorithms. The one of interest here is as follows. A pattern

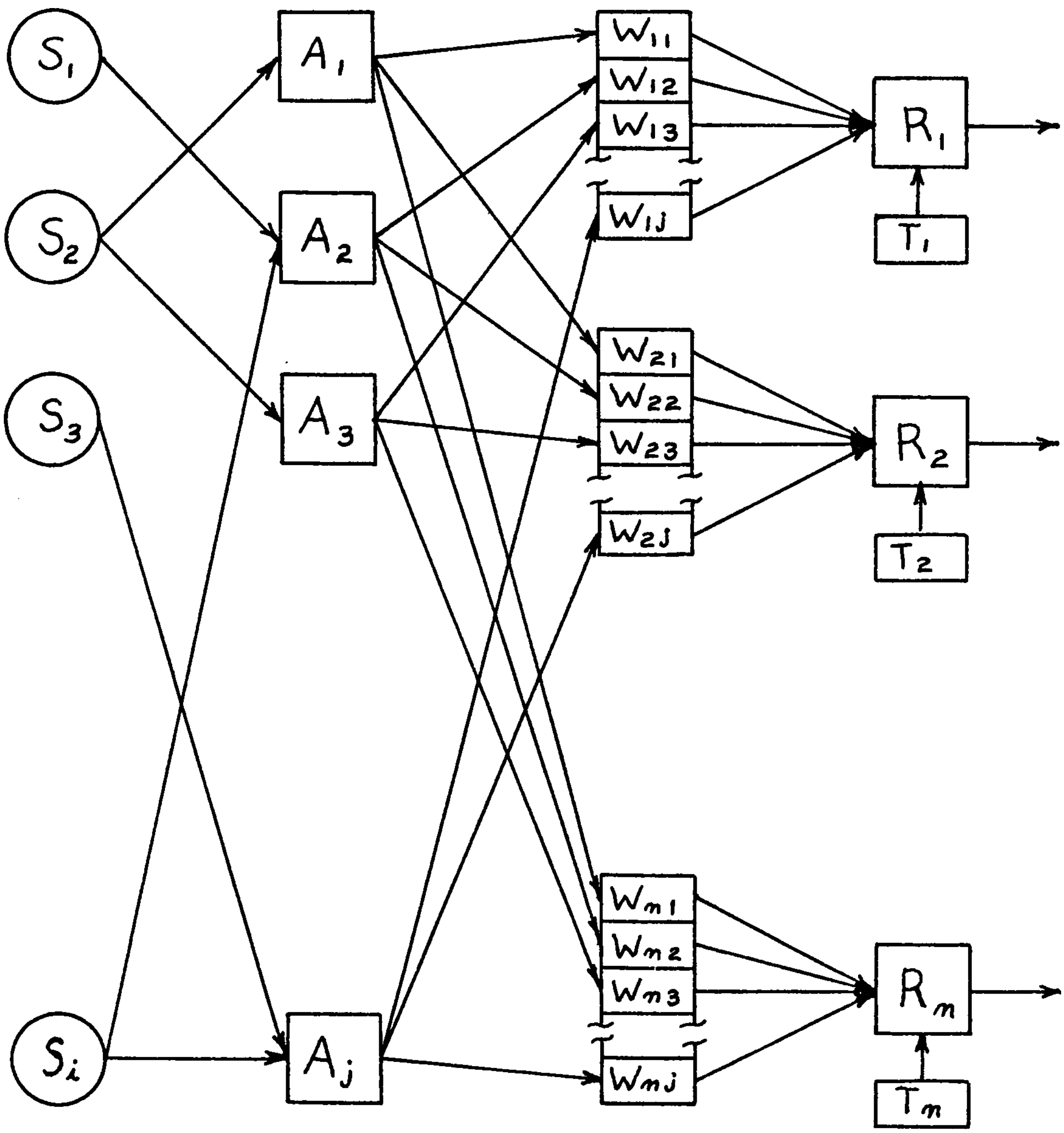


Figure 2.1

Organization of a Perceptron

is projected onto the receptor area and may or may not produce a response. If the pattern belongs to the set which is to produce a response and the correct response is produced then no action is required. If there is no response then each weight for which the corresponding A-unit is excited is increased in value by one unit and the threshold is reduced by one unit. If, however, the pattern does not belong to the set and a response is produced each of the appropriate weights is decreased and the threshold is increased by one unit. If no response was produced no action is required. This procedure is repeated for each character in a training set containing both characters which are to be recognized, i.e. produce a response, and characters which are not to produce a response. After a finite number of repetitions of this process for the entire training set the perceptron should be able to separate the set of characters into 2^n subsets where n is the number of R-units.

Rosenblatt investigated the operation of the perceptron from a probabilistic standpoint and determined the probability of correct classification in terms of such variables as the number of connections per A-unit, the threshold of each A-unit, the proportions of R-units to which an A-unit is connected, the number of A-units in the system and the number of R-units in the system[4]. The MARK-1 Perceptron was constructed using

400 receptors and 512 A-units each having 20 inputs. The amount of redundancy in this machine can be seen from the fact that after being trained to distinguish between E and X one half the A-units were switched off and the machine still performed with 100% accuracy. After 450 of the A-units were switched off the reliability did not fall below 80%(1).

The perceptron is examined from another viewpoint by Arkadev and Braverman(1). Each character projected onto an array of i receptors can be represented by a point in i-dimensional space. When the receptors are randomly connected to m A-units the receptor space is then divided by m randomly placed planes. This concept can be visualized by the use of Figure 2.2. This shows a cube divided by a plane separating the set of points 011, 101, 110, and 111 from the set 100, 010, 001, and 000. This is analagous to a three input threshold gate with a fixed threshold of -2. If two or more inputs are "1" the output is "1" indicating the positive side of the plane. If less than 2 inputs are "1" the output is zero indicating the negative side of the plane. In this way the m A-units cut the receptor space with m randomly placed planes and thus divide the i-dimensional space into many subvolumes. Arkadev and Braverman introduce the "hypothesis of the compactness of images" which in effect states that characters which belong to the same set are all represented by points which fall into

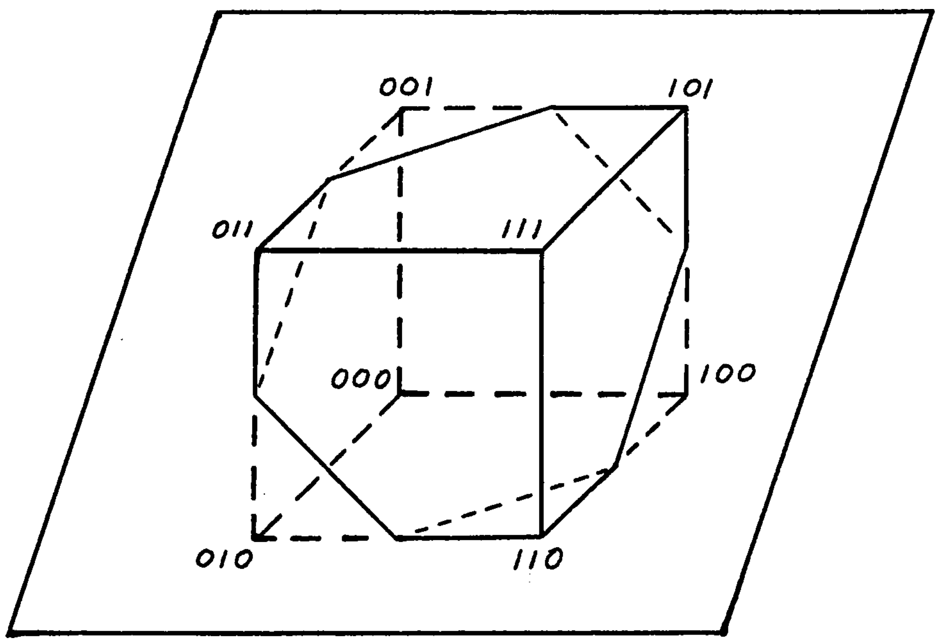


Figure 2.2

A Cube Divided by a Plane

a compact volume of receptor space which is separate from any points which do not belong to the set. Although no rigorous proof can be given for the hypothesis its intuitive truth can be illustrated by the observation that any recognizable "3" looks more like a "3" than any other character. If it were not so it would not be recognizable as a "3".

Due to the compactness of images, the pattern recognition problem reduces to the task of separating receptor space into subspaces each of which contains only points belonging to the same set and then producing the proper response for any character whose point falls in that subspace. In the perceptron the receptor space division is accomplished by the random planes created by the A-units and the adjustable weights are used during the training process to teach the machine to associate desired responses with the proper subspaces. After the training is complete and an unfamiliar character is presented, the machine recognizes it by producing the response it has been trained to give for the points near it in the training set.

Obviously the more random planes which are used to divide the receptor space the higher the probability will be that any two points belonging to different sets will have at least one plane separating them. In terms of hardware each plane requires one A-unit and n adjustable weights where n is the number of R-units. The redundancy in Rosenblatt's perceptron indicates that there were many more A-units than the minimum number required for reliable pattern recognition. This redundancy was desirable in a machine which was designed to be a model of the central nervous system which has been shown to be very redundant. However, for a machine designed specifically for character recognition it is more desirable to minimize the hardware.

CHAPTER III

MACHINE DESIGN

This section describes the operation and design of the machine from a functional block standpoint. Detailed schematics are included in the appendix.

One of the first problems which must be resolved in the design of an adaptive machine is the method of implimenting the adjustable weights. The requirements are that each weight must have a wide dynamic range of possible values but must be stable at any value once it is set. To facilitate the training of the machine the weights must be easily and quickly changed by a small percentage of their total range. There must also be provisions for accurately summing different combinations of weights and subtracting the threshold value. All these requirements are met by storing the weights and thresholds as binary numbers in a core memory and using a digital accumulator to accomplish the arithmetic. This solution to the adjustable weight problem plus the fact that training is achieved by a simple, fixed algorithm indicates that as much of the machine as possible should be implemented with digital hardware. Therefore, the machine which was designed is basically a small special purpose digital computer which simulates a four R-unit perceptron. A block diagram of the system is shown in Figure 3.1.

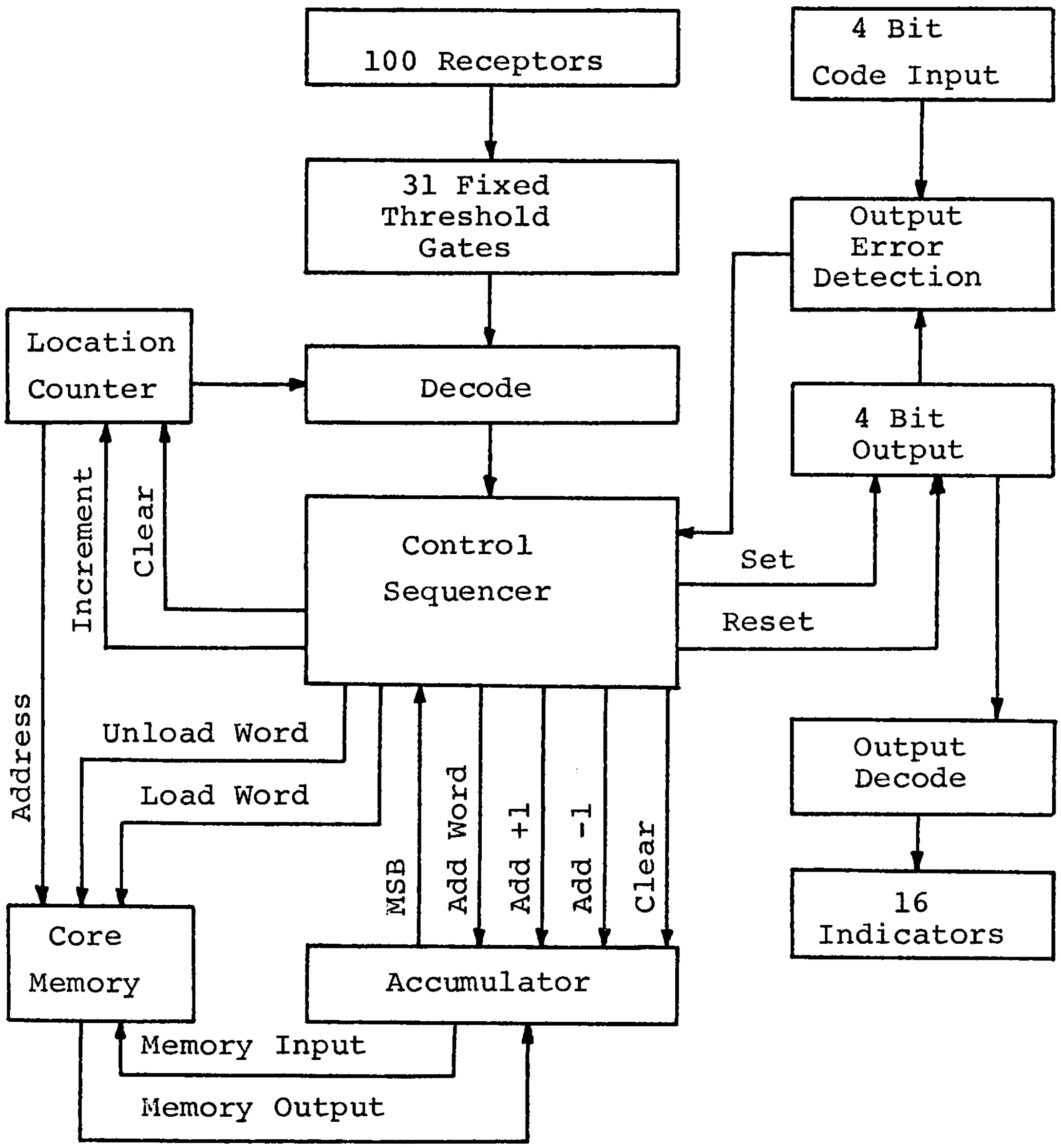


Figure 3.1

System Block Diagram

The receptor field consists of 100 cadmium-selenide photoconductive cells in a closely spaced 10 x 10 array. Each cell drives a saturating amplifier so that the output is high if part of a character falls on the photo cell and low otherwise. The outputs of the amplifiers are wired at random to the inputs of 31 threshold gates each having six inputs and a threshold of 2 so that if two or more of the inputs are high the threshold gate saturates. Therefore, when a character is projected onto the receptor screen it is represented by a 31 bit binary code at the output of the threshold gates.

A seven bit location counter is used to sequentially step through the 128 locations in memory containing 4 sets of 31 weights and 1 threshold corresponding to four 31 input adaptive threshold gates. The five least significant bits are decoded and the first 31 states each anded with one of the 31 bit binary code input lines so that if an input is up its corresponding weight in memory can be added into the accumulator. If the input bit is not up its weight is not added into the accumulator. The thirty-second state of the five least significant bits of the location counter indicates that the location of a threshold has been addressed and its value is subtracted from the weighted sum in the accumulator. The number in core memory is the two's compliment of the threshold value so that the subtraction can be performed by an add operation.

The two most significant bits are decoded to determine which of the four sets of weights and thresholds is being operated on at any step in the operation.

In order to facilitate the training of the machine, five additional light sensors are placed above the receptor array. One of these is used to initiate a read or adapt cycle when a character is projected onto the receptors and the other four are used to sense the binary code for the correct output for that character. A control sequencer circuit was designed to generate the proper signals to cause the machine to step through its sequence of operations.

Four output flip-flops corresponding to four response units are each set or reset depending on the sign of the result in the accumulator after a set of weights and a threshold has been summed. The outputs of these flip-flops are decoded to drive sixteen output indicator lights.

Each character the machine is to be trained to read is printed within a 1.85 cm square field centered on a clear plastic film in a standard 35 mm slide mount. The slides are placed in a projector which is placed such that the 1.85 cm field on the slide projects onto the 7.94 cm square receptor field. Above the character, dots are placed in the proper locations to project onto the code input light sensor the code for the correct output.

To initiate a read operation the slide containing the character to be read is projected onto the receptor array. The "initiate" light sensor triggers a mono-stable multivibrator which provides a delay long enough to allow the slower CdSe cells to reach their stable resistance. At the end of this delay, the accumulator and location counter are cleared and a read cycle begins. The zero state of the location counter addresses the first location in core memory which contains the first weight of the first of four sets of weights and thresholds. If the first input is up, i.e. the first threshold gate is on, as determined by the location decode, then "unload word" and "add word" signals are produced by the control sequencer to read the weight from memory and add it into the accumulator. If the first input is not up these signals are not generated. Next the control sequencer generates a signal to step the location counter to its next state which addresses the second location in memory. This address contains the second weight of the first set and it is added into the accumulator if the second input is high. This sequence of operations continues through all thirty one of the inputs. When the thirty second state of the location counter is reached, the core memory location containing the two's compliment of the first threshold value is being addressed. The control sequencer produces unload word and add word signals to subtract the threshold from the sum of the weights in the accumulator. At this point, if the result in the accumulator is positive (indicating that the sum of the weighted inputs is greater than or equal to the threshold), the first output flip-flop is set. If the result is negative (indicating that the sum of the weighted inputs is less than the threshold), the first output flip-flop is reset. The control sequencer next clears the accumulator and advances the location counter to its next state which addresses the first weight in the second set of weights and thresholds but the thirty third state of the counter has zeroes in the five least significant bits so that the first input is again interogated to determine whether or not this weight is to be added into the accumulator. The machine continues to step through the second set of weight locations adding the proper weights into the accumulator until it reaches the sixty fourth location and adds in the two's compliment of the second threshold and sets or resets the second output flip-flop. This sequence of operations continues until each of the 128 memory locations has been addressed and all four output flip-flops have been set in their proper state and one of the sixteen output indicator lamps is lit indicating the response for the character being read.

In the adapt mode, the machine first goes through a read cycle as described above and gives a response. Then the control sequencer steps the location counter through the memory again but this time performs the adapt algorithm. As the location counter proceeds through each of the four sections of memory the error detection circuit continuously compares the state of the corresponding output flip-flop with the correct output code being projected onto the four code input light sensors above the receptor field to determine whether or not each output is correct. If an output is correct no changes are made in the weights or the threshold for that section. If the output is incorrect, each weight for which the corresponding input is up is read from memory, transfered to the accumulator, and either one or minus one added to it depending on whether the desired output is one or zero respectively. After each weight correction, the new weight value is stored back in the same memory location and the accumulator is cleared before proceeding to the next address. When the threshold location is reached in a section which produced an incorrect response it must also be modified. The fixed increment training algorithm requires that if a zero response was given when a one response was required, each of the appropriate weights should be increased by one increment and the threshold decreased by one increment. If a one response was produced when a zero was required, each of the weights should be decreased and the threshold increased. Since the two's compliment of the threshold is stored in memory, when an add one operation is required on the weights an add one operation is also performed for the threshold because adding one to the two's compliment is equivalent to subtracting one from the threshold. The result in the accumulator is the two's compliment of the new threshold value and is stored back into memory.

To operate the machine, a training set is made up by preparing slides each containing one of the characters with the desired output code. The slides are loaded into the slide tray of the projector which is placed with the center of the projection lens aligned with the center of the receptor field and adjusted so that the character field on the slide matches the receptor field. After the machine power has been turned on the CLEAR MEMORY switch is depressed momentarily to zero all locations in memory. The READ-ADAPT mode switch is placed in the ADAPT position and each slide in turn is projected onto the receptor screen. As each new slide is flashed onto the machine a read cycle is automatically initiated during which a response is produced. This is followed by an adapt cycle during which, if the response was incorrect, the weights and thresholds are adjusted and the ERROR indicator is turned on. The entire training set is repeatedly presented to the machine until one complete pass is made without any errors, indicating that the machine has learned all the characters in the set. The READ-ADAPT switch is then placed in the READ position for any further reading or testing to be performed. With the mode switch in this position each time a slide is projected onto the receptors the machine goes through the read cycle only. At any time while a character is being projected a read-adapt cycle or a read only cycle, depending on the position of the mode switch, can be initiated by depressing the RESET switch on the front panel.

The machine is completely self contained requiring only 110 volt 60 cycle power. There are no internal adjustments required for the normal operation of the machine as described above. The machine was set up to operate in normal room light with the characters printed on clear plastic slides with a fine point, oil base, black marking pen and projected from a Kodak Carousel 850 projector using a 500 watt projection lamp. For detailed circuit and logic diagrams the reader is referred to the appendix.

CHAPTER IV

RESULTS AND CONCLUSIONS

This section describes the tests which were performed to determine the capabilities and limitations of the machine.

The first test was designed to examine the convergence characteristics of the machine. The set of slides shown in Figure 4.1 was made using the numbers 0 through 9 and the letters A through F. The binary code assigned to each character was the standard hexadecimal code. The machine was trained to distinguish between pairs of characters and the number of passes through the training set of two characters was noted. Fifteen such training runs were performed and the average number of passes calculated. The perceptron was then trained to recognize three characters at once. These tests were repeated, each time increasing the number of different characters in the training set until it was found that for some combinations of eleven characters the machine would not converge in one hundred passes. The results of these tests are summarized in Table 4.1.

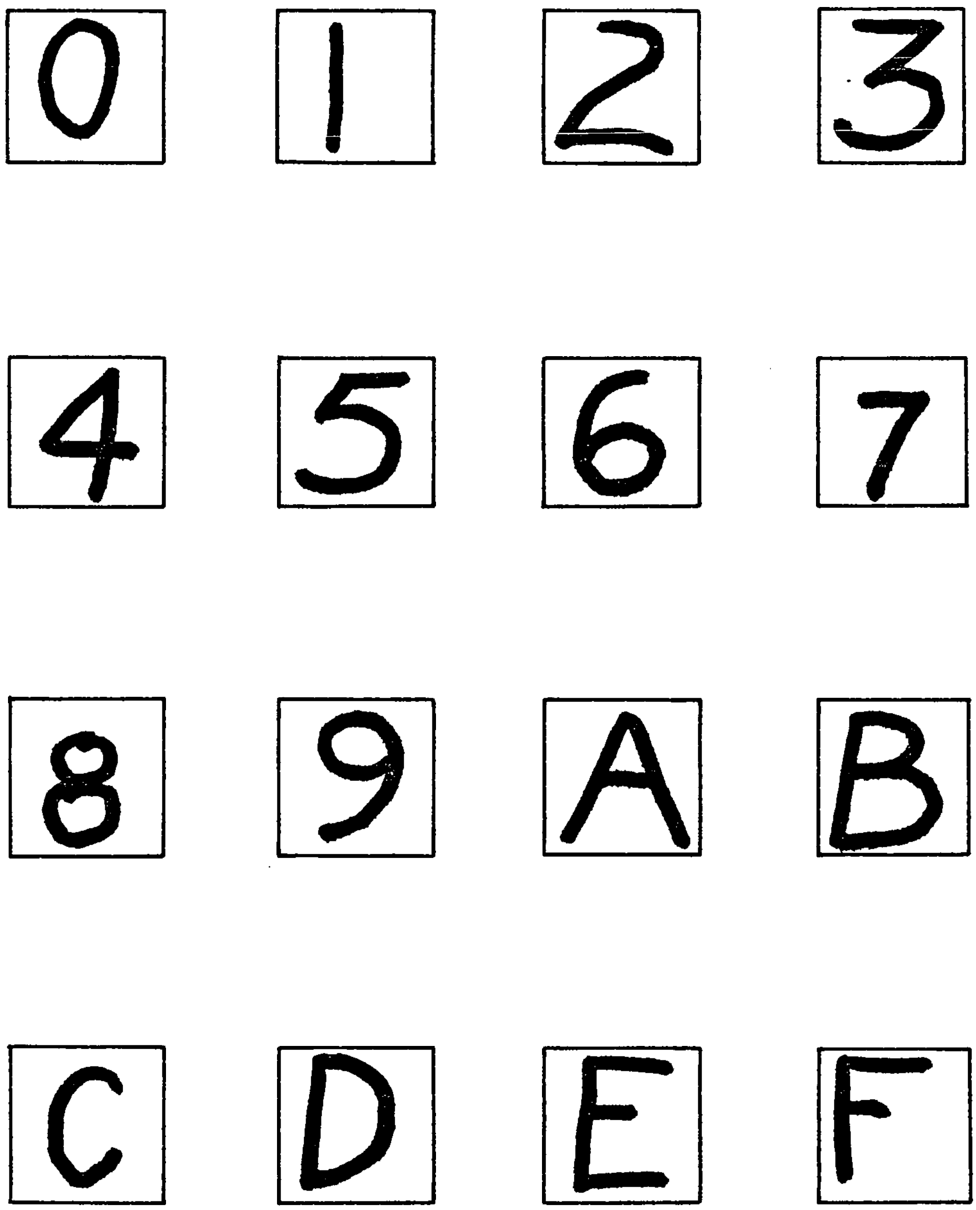


Figure 4.1

The set of 16 characters used for the first test.

Number of Characters In Training Set	Number of Passes to Converge Minimum	Average	Maximum
2	2	2.5	4
3	2	3.6	6
4	3	5.5	9
5	4	6.1	8
6	5	9.0	14
7	9	15.0	21
8	9	17.8	31
9	12	32.2	85
10	13	23.2	29
11	30	*	*

TABLE 4.1

Results of first test.

In addition to counting the number of passes through the training set required for convergence, other characteristics were noted such as the number of errors on each pass. Although the general trend was toward fewer errors as the number of passes increased the behavior was very erratic. For example,

* Some sets did not converge before the test was terminated at 100 passes.

when the machine was being trained to read the set of nine characters "4" through "C" it classified them all correctly on the seventieth pass except for the "B" which it classified as a "7". Some of the weights which were changed during the adapt cycle to correct for this error had been used to correctly classify the "4", the "6", and the "8" so that on the seventy-first pass these three characters were read incorrectly. A total of eight-five passes was required before the machine had learned to classify all nine characters correctly. Because of this behavior there is no external means of determining after any number of unsuccessful passes whether or not the machine will ever converge for a particular set of characters.

Although these tests gave some insight into the characteristics of the perceptron, the machine must be able to do more than just recognize characters in a training set to perform hand lettered character recognition. In order to be able to generalize and read correctly characters which were not in the training set, the machine must be trained using many different examples of each character it is to classify. This was demonstrated by the next test which also pointed out the severe limitations of this machine. The machine was trained using one each of the four characters "0" through "3" then tested on a set of five different examples of each of the characters and classified only 25% of them correctly. As the number of examples of each character in the training set was increased the percentage of correct responses in the test set increased. When the machine was trained on a set containing six each of the characters "0" through "3" then tested on the same set as before, 60% of the characters were classified correctly. Figure 4.2 shows some of the characters in the training set for this test. Figure 4.3 and 4.4 contain examples of characters in the test set which were classified correctly and incorrectly respectively. The machine's limitations appeared when an attempt was made to extend the above process. The perceptron would not converge on a training set consisting of seven examples each of the characters "0" through "3".

When the machine will not converge on a given training set one of two possible problems exists. Two different characters in the set may be covering the same receptors and the machine cannot distinguish between them because they are represented by the same point in receptor space. The only solution to this problem would be to build a machine with more receptors to obtain a finer resolution of the optical projection. The other possible problem is that two different characters are represented by two points in receptor space which are not separated by one of the random planes created by the threshold gates. This problem could possibly be solved by

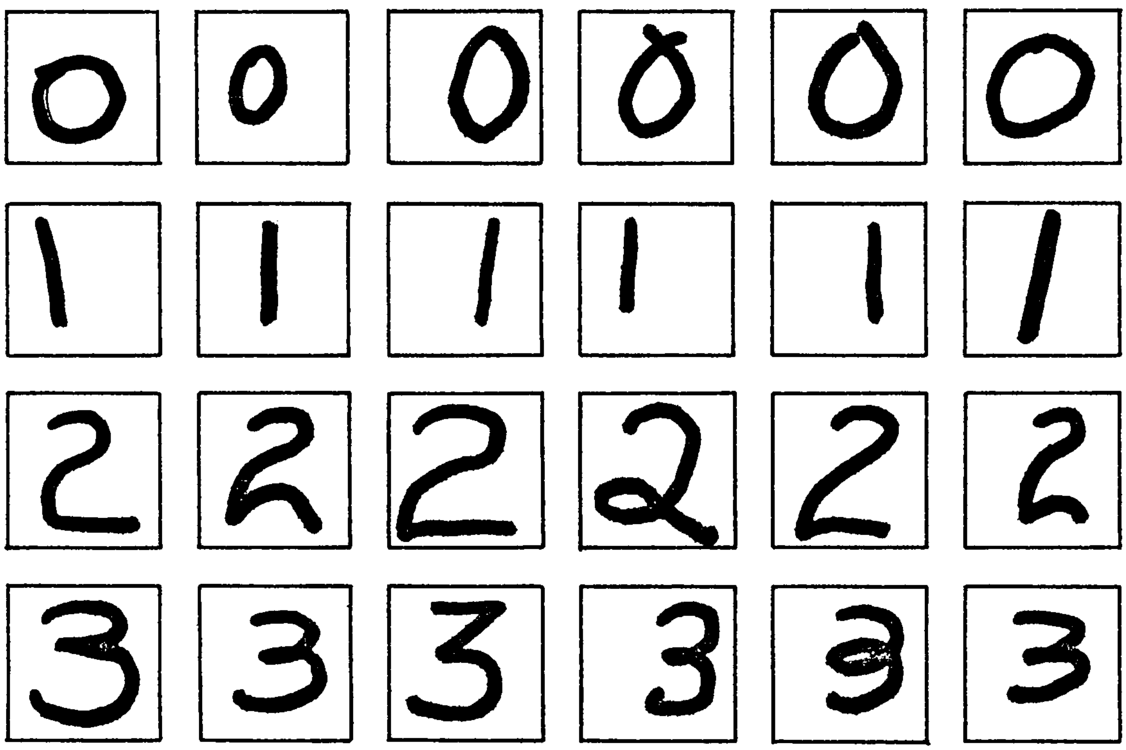


Figure 4.2

Characters in the training set.

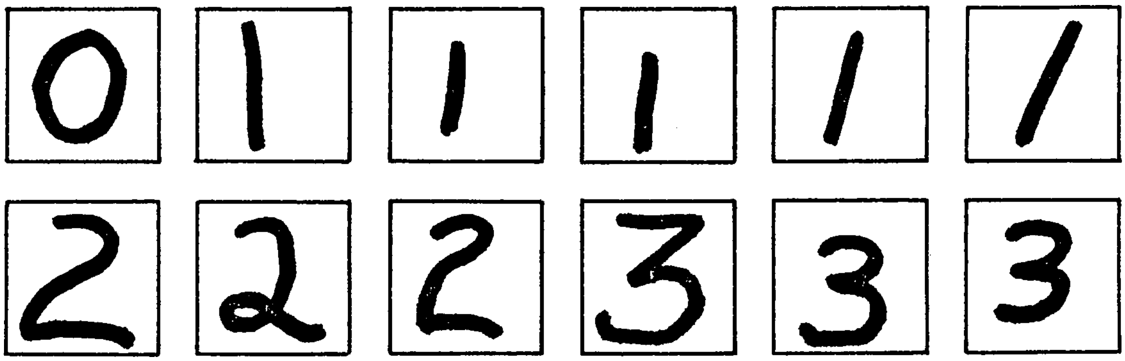


Figure 4.3

Characters classified correctly in the test set.

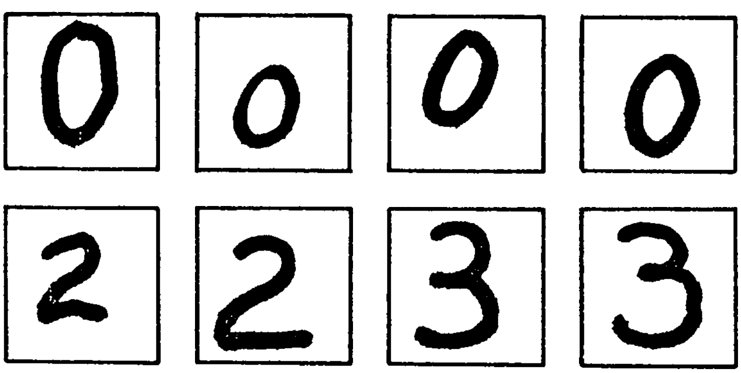


Figure 4.4

Characters classified incorrectly in the test set.

adding more threshold gates to more finely divide receptor space. However, if the two characters only differ in a few receptor positions, the probability of having a random plane fall between them is quite low and a better solution would again be to use more receptors. By having better resolution, i.e. more dimensions to receptor space, similar characters would differ in more positions and the probability of a random plane falling between them would be higher.

The testing reported above is sufficient to draw the following conclusions:

1. The hardware does operate correctly.
2. The theory of operation is sound.
3. A larger machine of this type, i.e. more receptors for better resolution and more fixed threshold gates for finer division of receptor space, would be required in a practical hand lettered character recognition machine.

The sixteen possible outputs for the machine are the sixteen states of the four output flip-flops, each of which indicates the response of an independent perceptron. For a given training set each perceptron must be trained to separate the characters having a "1" in one position of the output codes from those having a "0" in that position. For this reason the output code chosen for the characters can effect the training time and an optimum code could be found by tests run on the machine.

The fixed threshold gates were constructed so that if any two of the six inputs to a gate are up the gate is triggered. By changing the input weighting resistors this threshold could be changed and tests could be conducted to determine the effect of different threshold levels.

The training algorithm specifies that the weights and thresholds be incremented or decremented in unit steps until the perceptron converges. In any practical machine however, there must be maximum and minimum values for the weights and thresholds. If any weight or threshold reaches one of these limits before the machine converges, what should the machine do? Any action other than building a machine with a larger dynamic range will be a departure from the training algorithm and may or may not prevent the machine from ever converging. There are no provisions in the present machine to detect this condition. Therefore, when a weight reaches the value of +256 and another increment is called for, there is an undetected overflow in the accumulator and the new weight is stored as a -256. It would be interesting to add the necessary circuitry to detect this condition and inhibit the ADD +1 so that the weights or threshold would saturate rather than change sign.

Another test which could be performed would be to disable some of the threshold gates and receptors to determine the amount of deterioration in the operation of the machine. This could give an indication of how much improvement could be achieved by adding more threshold gates and receptors.

This research has shown that a working perceptron can be built using modern digital integrated circuitry and core memory. By simulating the adjustable weights with binary numbers and the threshold gates with an accumulator, the resultant system is much smaller, more accurate, and more easily trained than an analog perceptron of the same capability.

BIBLIOGRAPHY

1. Arkadev, A. G. and E. M. Braverman, Computers and Pattern Recognition. Washington, D. C.: Thompson Book Company, 1967.

2. Gschwind, H. W., Design of Digital Computers. New York: Springer-Verlog, 1967.

3. Nilsson, N. J., Learning Machines. New York: McGraw-Hill Book Company, 1965.

4. Rosenblatt, F., "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," Psychological Review, Vol. 65, pp. 386-408; 1958.

5. Rosenblatt, F., "The Perceptron: A Theory of Statistical Separability in Cognitive Systems," Cornell Aeronautical Laboratory, Inc. Report No. VG-1196-G-1. Buffalo, New York: 1958.

APPENDIX

Photographs and Schematics

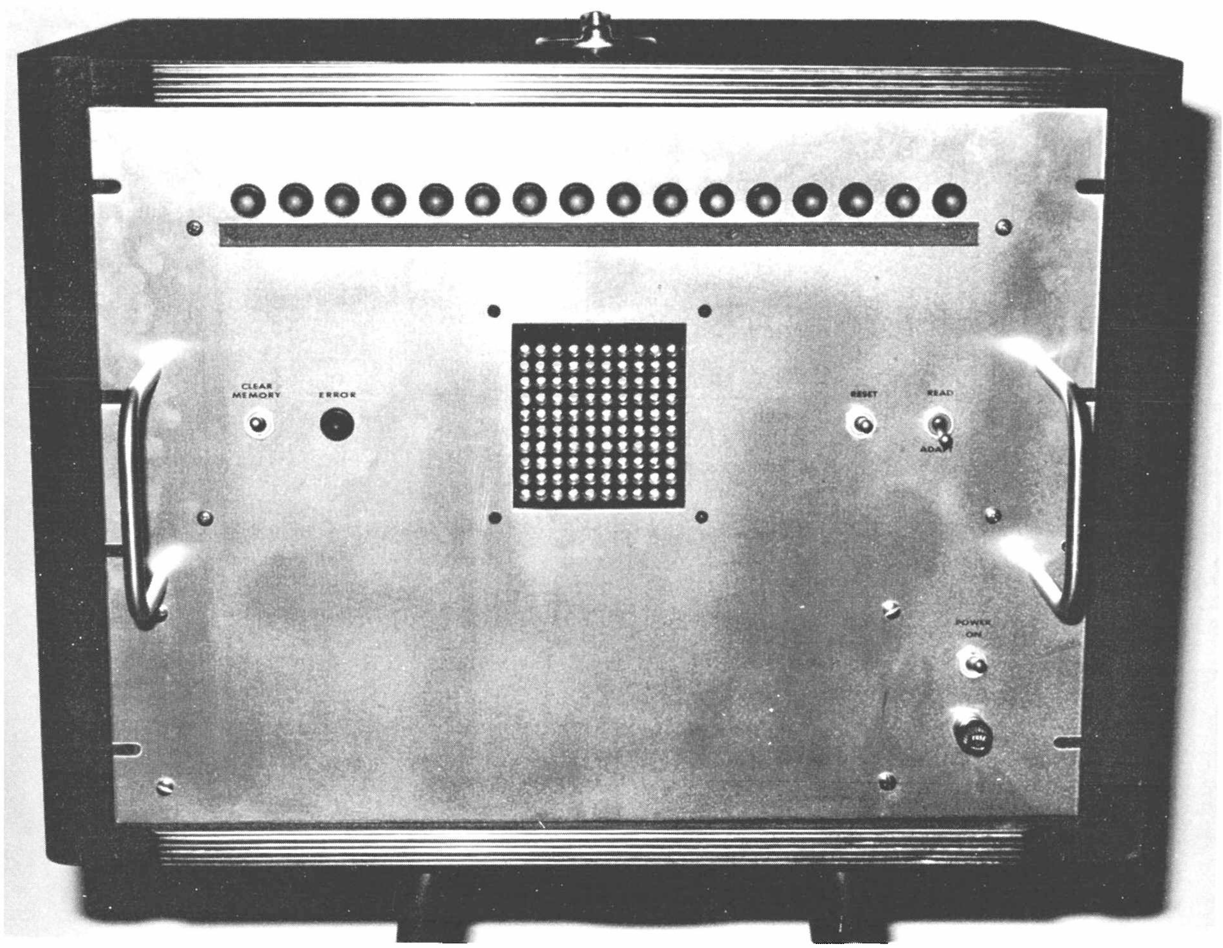


Figure A.1

The Adaptive Pattern Recognition Machine

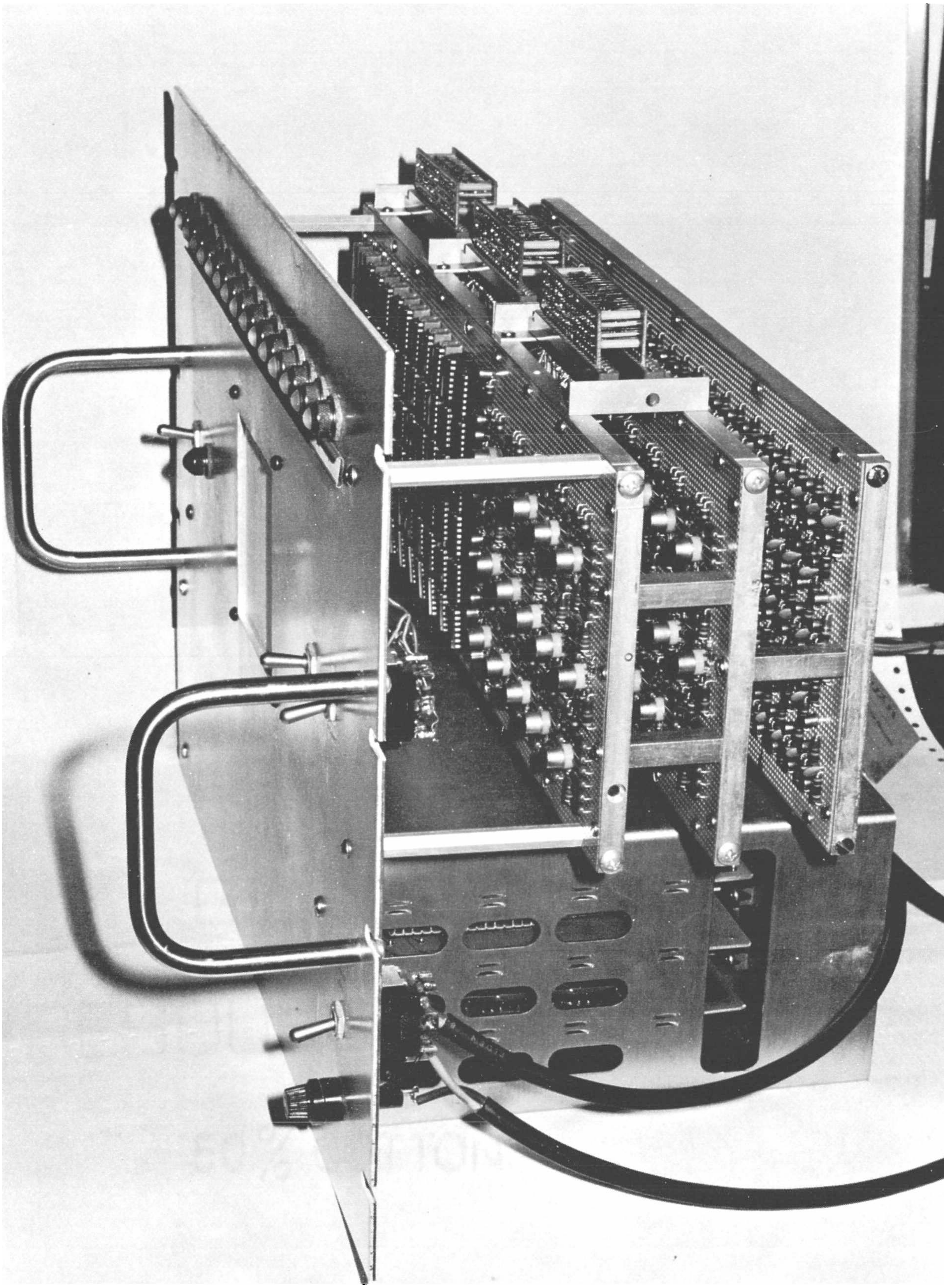


Figure A.2

Assembly Removed From Cabinet

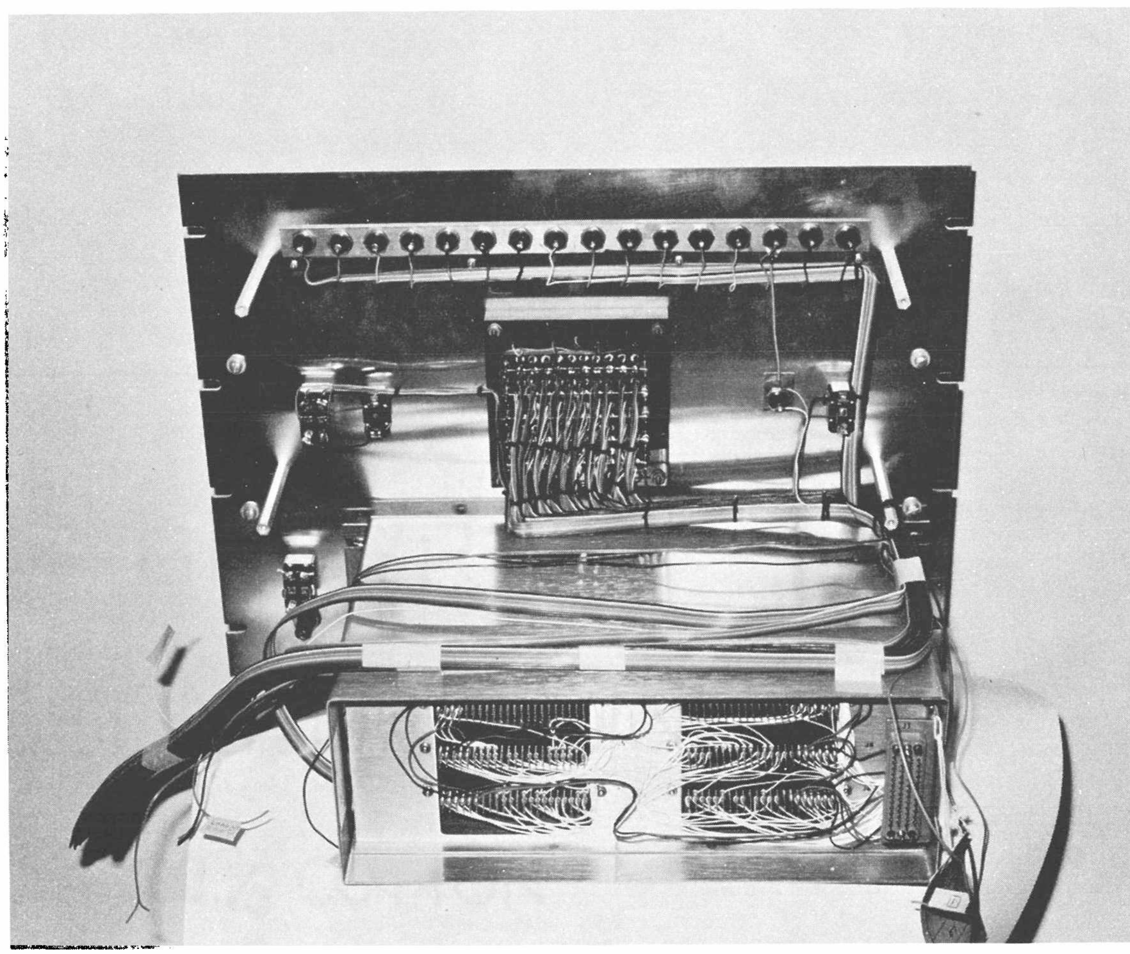


Figure A.3

Rear View of Front Panel With Photo-cell
Matrix and Feroxcube Core Memory

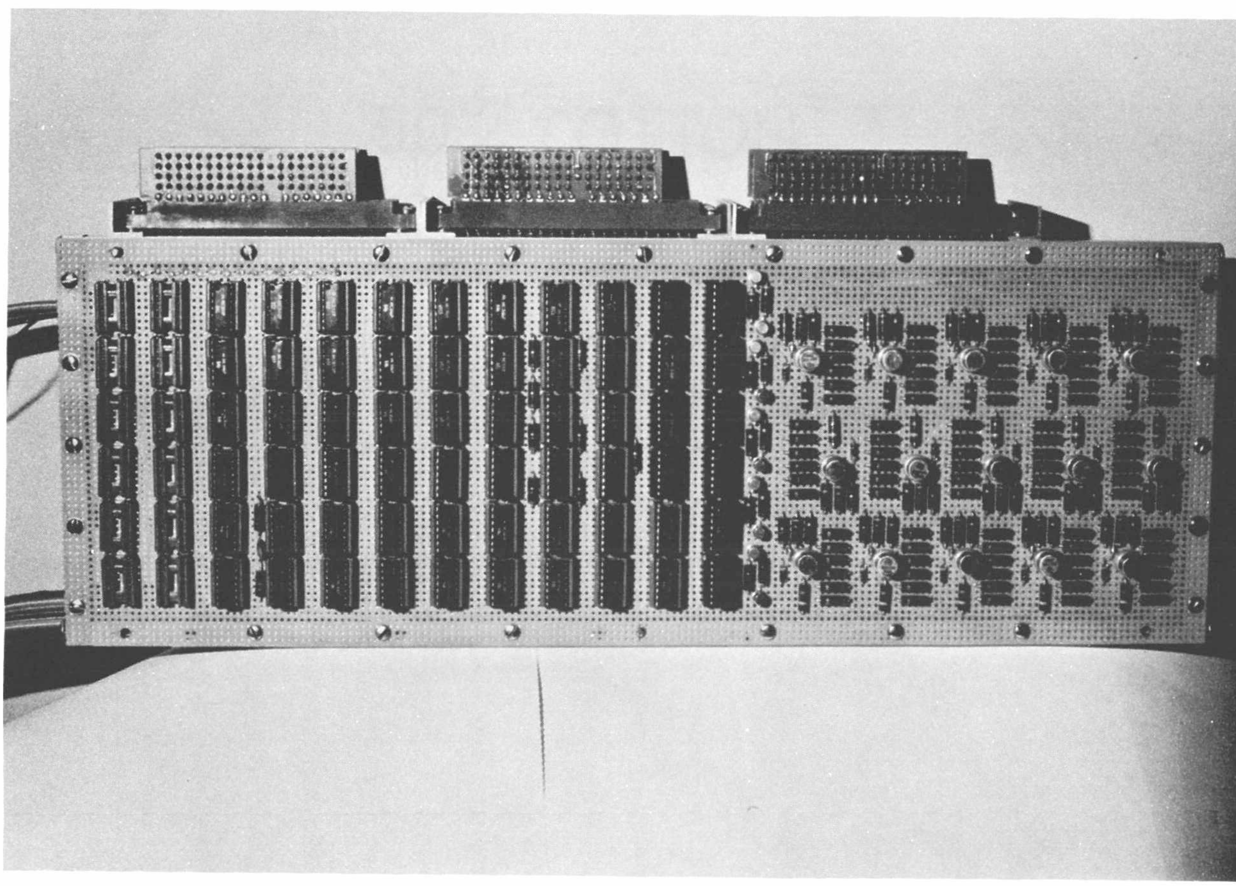


Figure A.4

First Circuit Board Containing Input and Output Decode Matrices, Threshold Gates 1-15, and Digital Integrated Circuits

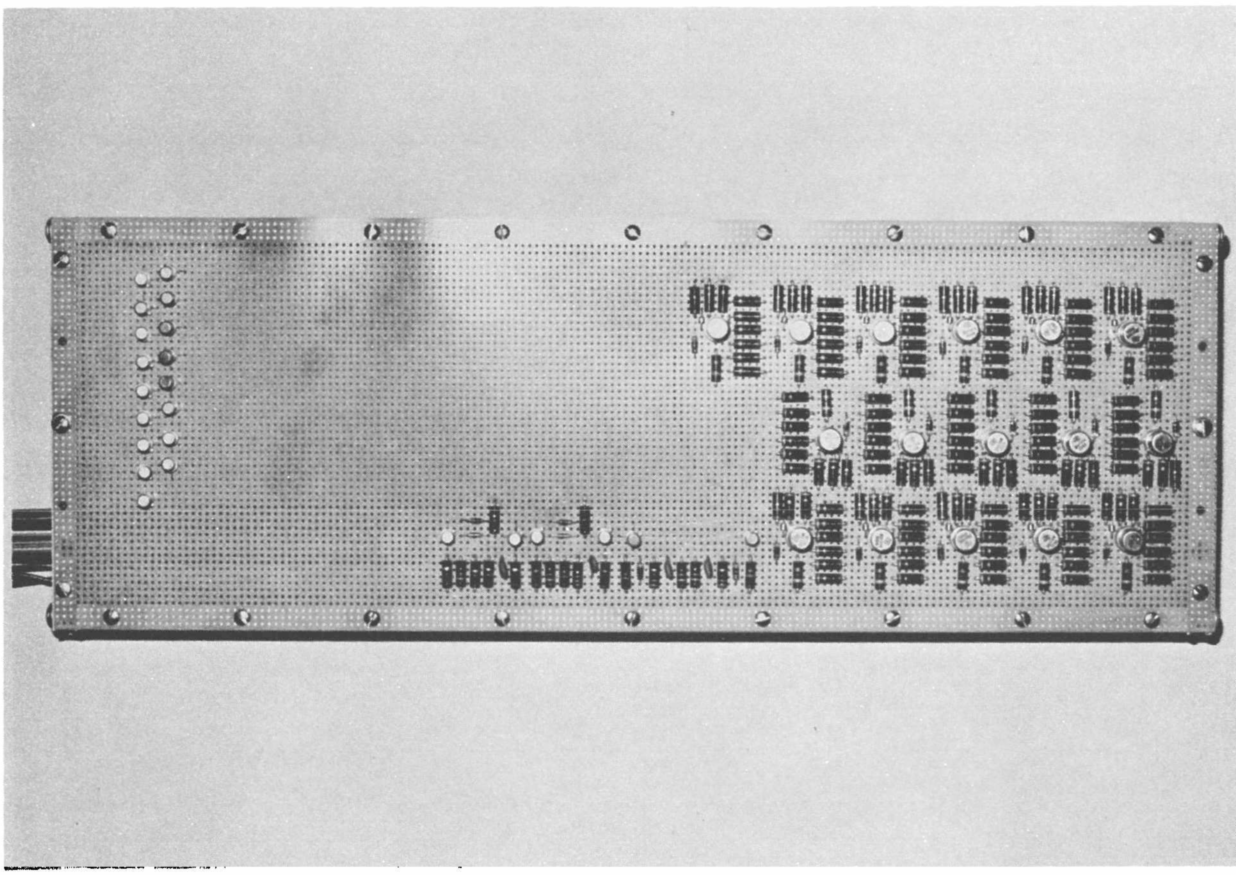


Figure A.5

Second Circuit Board Containing Threshold Gates 16-31, Clock, Load and Unload One-shots, and Indicator Lamp Drivers

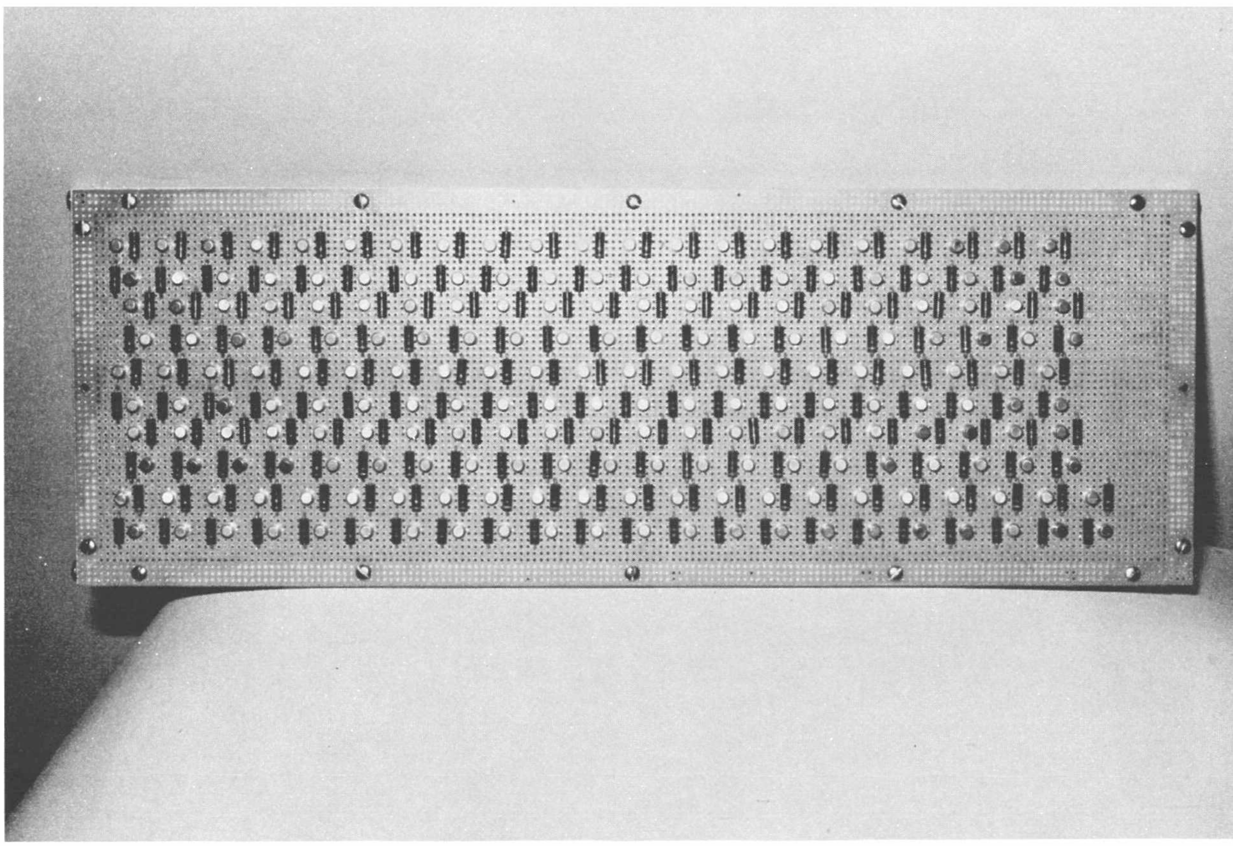
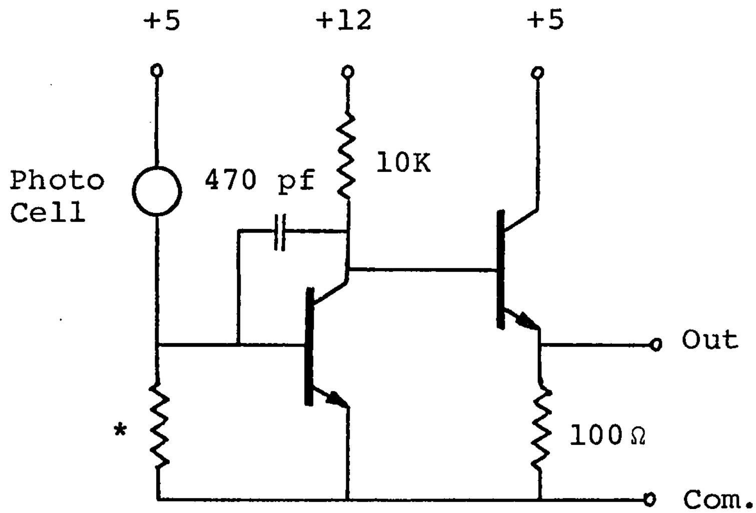


Figure A.6

Third Circuit Board Containing 105
Photo-cell Amplifiers



*Selected

Figure A.7

Photo-cell Amplifier

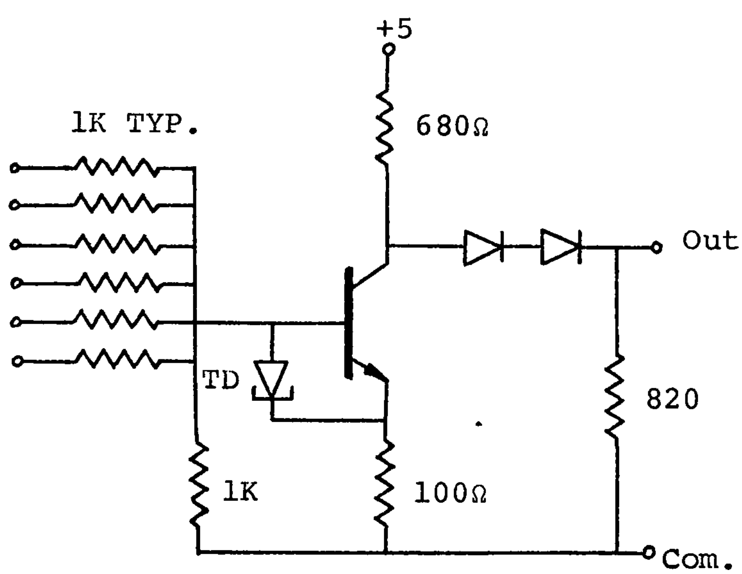


Figure A.8

Threshold Gate

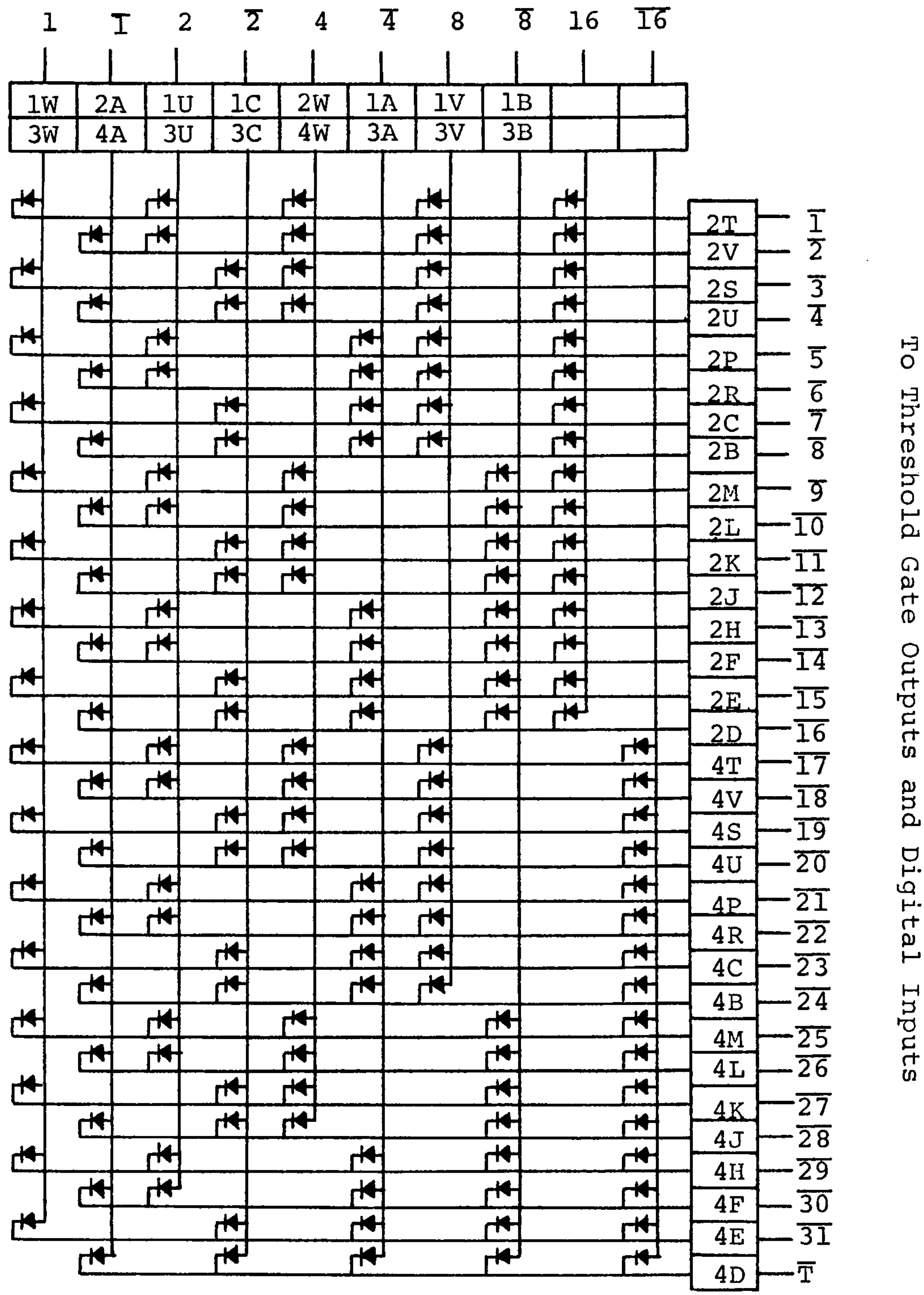


Figure A.9

Input Decode Matrix

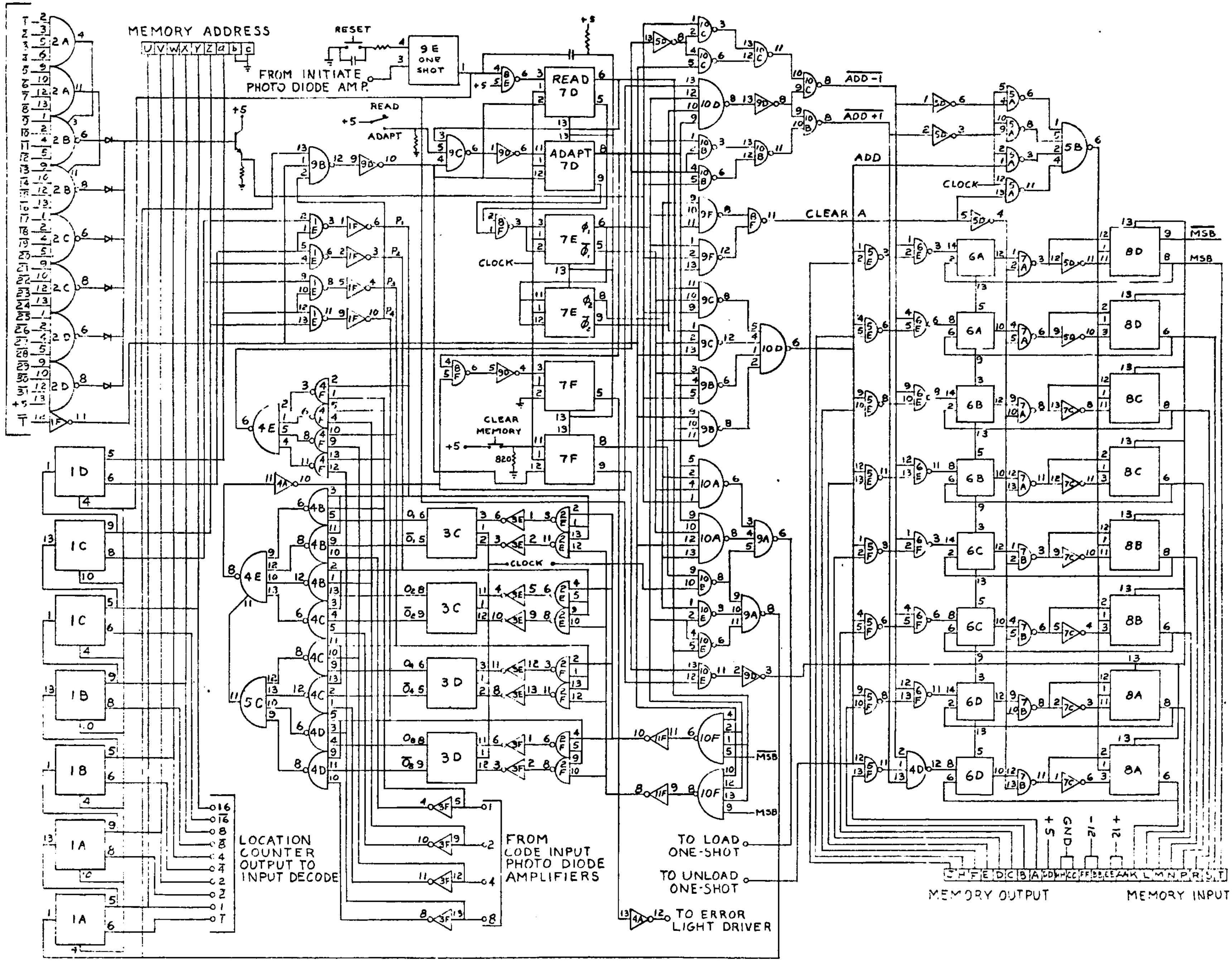


Figure A.10
Digital Logic Diagram

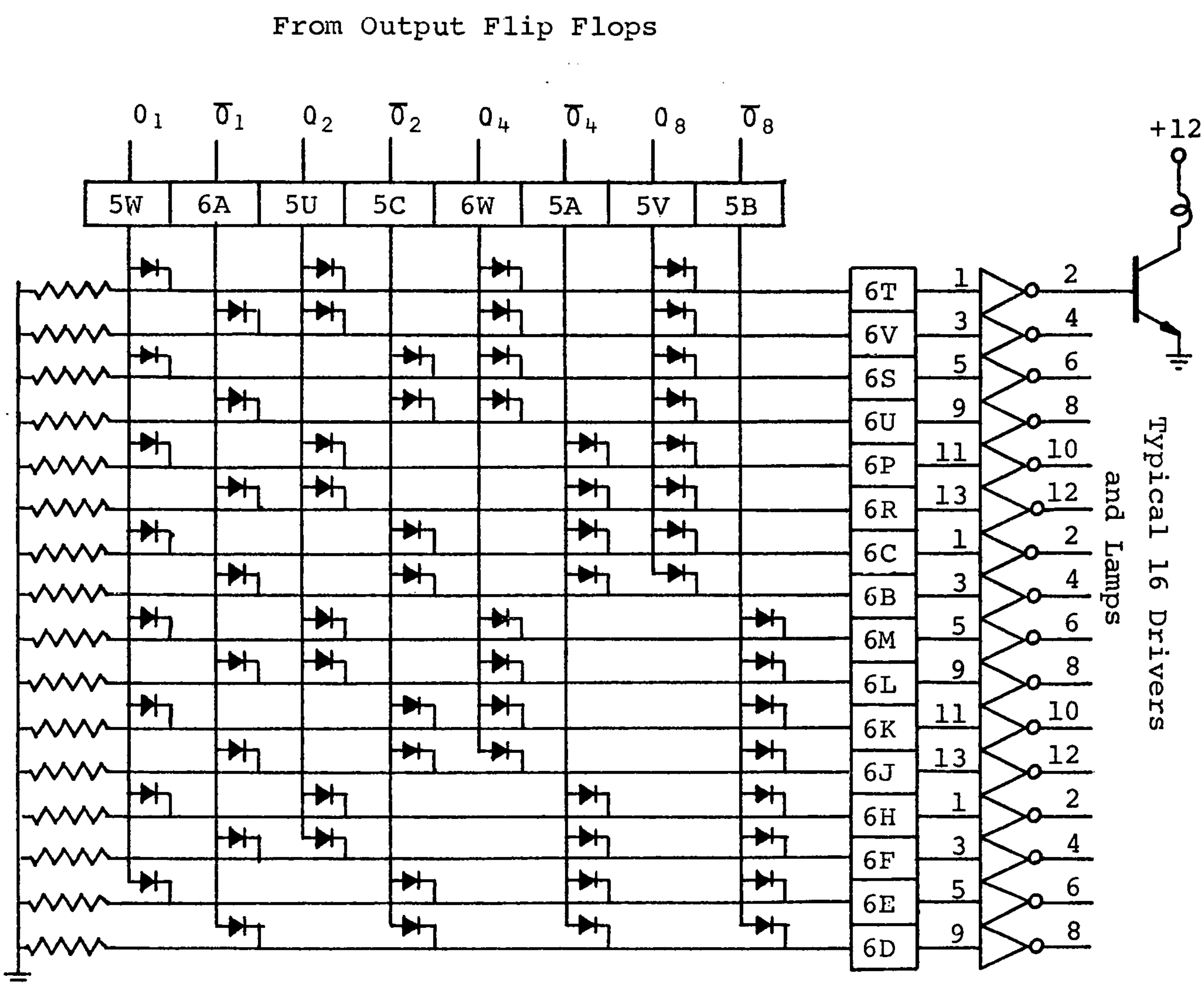


Figure A.11
Output Decode and Indicators

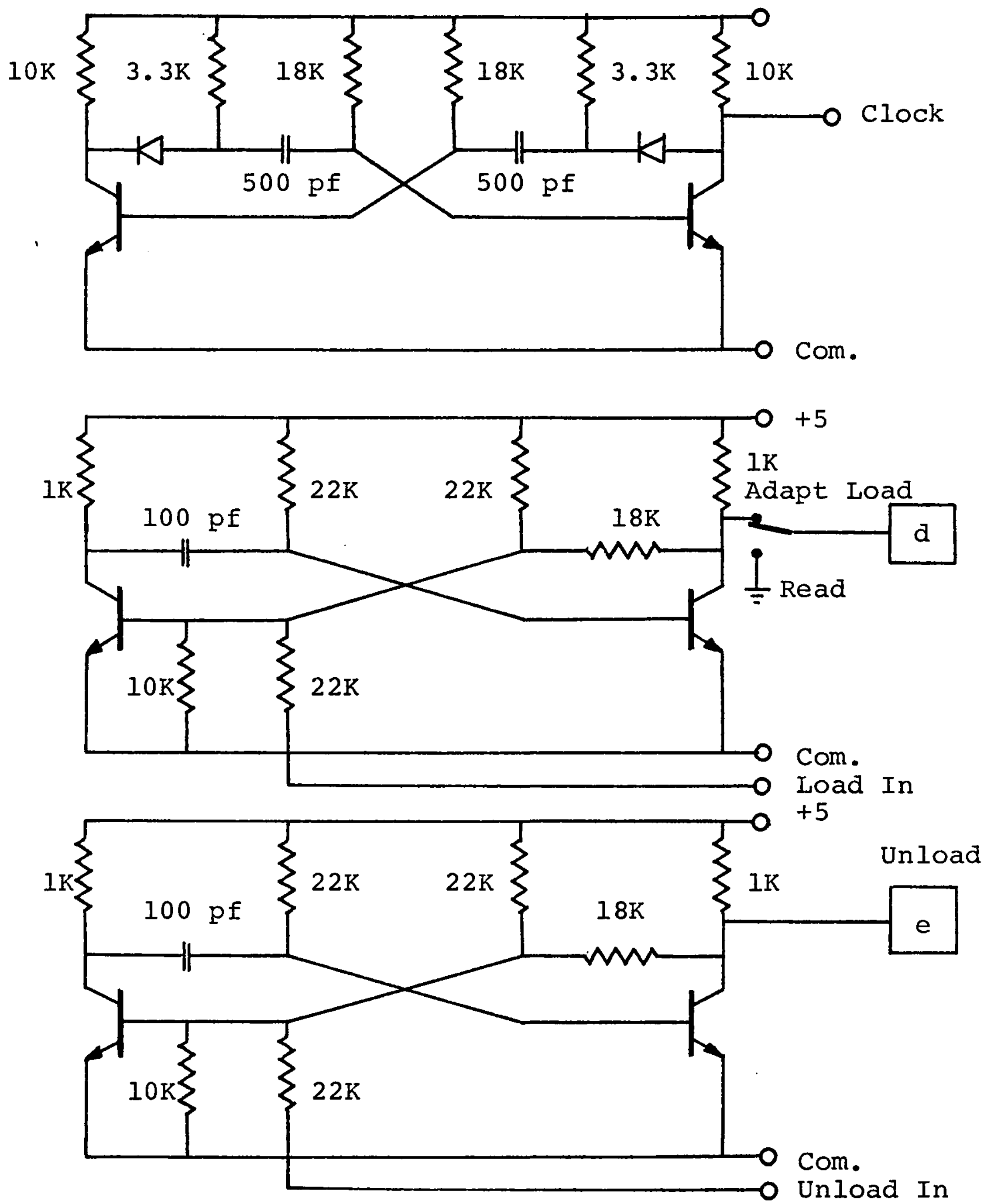


Figure A.12

Clock, Load One-shot, Unload One-shot

Threshold Gate	Photo-cells	Threshold Gate	Photo-cells
1	57,75,86,88,96,98	16	36,40,43,46,47,87
2	4,22,29,39,52,56	17	12,14,31,40,71,83
3	37,38,53,63,86,90	18	14,22,23,37,64,76
4	35,57,67,71,74,76	19	33,60,66,72,77,89
5	17,35,44,54,55,78	20	1,18,51,70,89,93
6	3,23,28,30,34,55	21	33,43,59,66,87,88
7	7,25,32,79,92,93	22	27,30,48,49,68,91
8	31,47,61,79,84,100	23	6,36,54,83,85,97
9	9,13,16,17,61,72	24	11,13,19,73,82,99
10	6, 8,20,64,81,95	25	8,21,38,67,80,85
11	24,29,34,42,63,84	26	12,16,39,48,58,80
12	5,10,42,50,69,81	27	15,21,41,62,77,96
13	2,32,53,56,59,65	28	24,26,27,45,49,58
14	28,45,46,51,74,75	29	3,30,70,73,94,98
15	15,18,19,41,52,90	30	25,44,50,60,62,78
		31	11,26,65,68,69,82

Table A.1

Photo-cell to Threshold Gate Wiring List

Location		Description
1A	MC853P	Dual J-K Flip Flop
1B	MC853P	Dual J-K Flip Flop
1C	MC853P	Dual J-K Flip Flop
1D	MC853P	Dual J-K Flip Flop
1E	MC846P	Quad 2 Input Nand
1F	MC834P	Hex Inverter
2A	MC833P	Dual 4 Input Expander
2B	MC830P	Dual 4 Input Nand
2C	MC830P	Dual 4 Input Nand
2D	MC830P	Dual 4 Input Nand
2E	MC846P	Quad 2 Input Nand
2F	MC846P	Quad 2 Input Nand
3A	MC837P	Hex Inverter
3B	MC837P	Hex Inverter
3C	MC852P	Dual J-K Flip Flop
3D	MC852P	Dual J-K Flip Flop
3E	MC834P	Hex Inverter
3F	MC834P	Hex Inverter
4A	MC837P	Hex Inverter
4B	MC862P	Triple 3 Input Nand

Table A.2

Integrated Circuit Location Table

Location	Description	
4C	MC862P	Triple 3 Input Nand
4D	MC862P	Triple 3 Input Nand
4E	MC830P	Dual 4 Input Nand
4F	MC846P	Quad 2 Input Nand
5A	MC846P	Quad 2 Input Nand
5B	MC830P	Dual 4 Input Nand
5C	MC833P	Dual 4 Input Expander
5D	MC834P	Hex Inverter
5E	MC846P	Quad 2 Input Nand
5F	MC846P	Quad 2 Input Nand
6A	MC796P	Dual Full Adder
6B	MC796P	Dual Full Adder
6C	MC796P	Dual Full Adder
6D	MC796P	Dual Full Adder
6E	MC846P	Quad 2 Input Nand
6F	MC846P	Quad 2 Input Nand
7A	MC846P	Quad 2 Input Nand
7B	MC846P	Quad 2 Input Nand
7C	MC834P	Hex Inverter
7D	MC852P	Dual J-K Flip Flop

Table A.2 Continued

Location		Description
7E	MC852P	Dual J-K Flip Flop
7F	MC852P	Dual J-K Flip Flop
8A	MC852P	Dual J-K Flip Flop
8B	MC852P	Dual J-K Flip Flop
8C	MC852P	Dual J-K Flip Flop
8D	MC852P	Dual J-K Flip Flop
8E	MC846P	Quad 2 Input Nand
8F	MC846P	Quad 2 Input Nand
9A	MC862P	Triple 3 Input Nand
9B	MC862P	Triple 3 Input Nand
9C	MC862P	Triple 3 Input Nand
9D	MC834P	Hex Inverter
9E	MC851P	One-shot
9F	MC862P	Triple 3 Input Nand
10A	MC830P	Dual 4 Input Nand
10B	MC846P	Quad 2 Input Nand
10C	MC846P	Quad 2 Input Nand
10D	MC830P	Dual 4 Input Nand
10E	MC846P	Quad 2 Input Nand
10F	MC830P	Dual 4 Input Nand

Table A.2 Continued