# ASSOCIATIVE MEMORY SYSTEMS AND THEIR APPLICATIONS TO PICTURE AND ARITHMETIC PROCESSINGS

C. C. YANG

AD 641205

TECHNICAL REPORT 66-103

AUGUST 1966

Information-Processing And Control
SYSTEMS LABORATORY

NORTHWESTERN UNIVERSITY
EVANSTON, ILLINOIS
60201

Reproduction in whole or in part is permitted for any purpose of the United States Government.

The research reported in this document was jointly supported by the U. S. Air Force Office of Scientific Research under Grant No. AF-AFOSR-98-65, and the U. S. Army, Navy, and Air Force in a Joint Service Electronics Program under Office of Naval Research Contract Number N00014-66-C0020-A03 (Identification Number NR 373-502/3-14-66 Electronics Branch).

The distribution of this document is unlimited. Qualified requesters may obtain copies of this report from the Defense Documentation Center.

TECHNICAL REPORT 66-103

# ASSOCIATIVE MEMORY SYSTEMS AND THEIR APPLICATIONS TO PICTURE AND ARITHMETIC PROCESSINGS

Chao-Chih Yang

Electrical Engineering Dept.

The research reported in this document was jointly supported by the U. S. Air Force Office of Scientific Research under Grant No. AF-AFOSR-98-65, and the U. S. Army, Navy, and Air Force in a Joint Service Electronics Program under Contract No. N00014-66-C0020-A03.

August 1966

Information-Processing And Control
SYSTEMS LABORATORY

NORTHWESTERN UNIVERSITY

ACKNOWLEDGMENT

The author wishes to express his deep gratitude to Professor S. S. Yau for his valuable advise and constant encouragement during the preparation of this work and throughout the author's years as a graduate student at Northwestern University.

The author would also like to express his appreciation to Northwestern University for granting him the Walter P. Murphy Fellowships and the U.S. Air Force Office of Scientific Research for providing additional financial support.

To his wife the author owes special thanks for her constant and continuous patience during the author's long stay in the U. S. A.

Finally, the author wants to thank Mrs. Billie Bram for her patience and efficiency in typing this thesis.

TABLE OF CONTENTS


| | | Page |
|---|---|---|
| 1. | Introduction | 1 |
| 2. | A Cryogenic Associative Memory System | 4 |
| 2.1. | Introduction | 4 |
| 2.2. | The Logical Structure | 6 |
| 2.2a. | Bit-Memories | 6 |
| 2.2b. | Word Selection and Control Circuits | 9 |
| 2.2c. | Word-Write-Read-Sense Circuit | 11 |
| 2.2d. | Comparison Circuit | 15 |
| 2.2e. | Match-Indication and Memory-Clear Circuit | 15 |
| 2.3. | Operational Procedures | 16 |
| 2.3a. | Comparison Process | 16 |
| 2.3b. | Writing Process | 17 |
| 2.3c. | Non-Ordered Retrieval | 19 |
| 2.3d. | Ordered Retrieval | 21 |
| 3. | A Cutpoint Cellular Associative Memory System | 26 |
| 3.1. | Introduction | 26 |
| 3.2. | Cutpoint Cells | 26 |
| 3.3. | Th Logical Structure | 29 |
| 3.3a. | Bit-Memories | 29 |
| 3.3b. | Match-Tag Networks | 31 |
| 3.3c. | Output Networks | 31 |
| 3.3d. | Controlled Shift Register | 34 |
| 3.4. | Operational Procedures | 36 |
| 3.4a. | Comparison and Tag Processes | 36 |
| 3.4b. | Writing Process | 36 |
| 3.4c. | Reading Process | 39 |
| 4. | Pattern Recognition by Using an Associative Memory | 42 |
| 4.1. | Introduction | 42 |
| 4.2. | Representation of Input Patterns | 43 |
| 4.3. | Determination of Templates of Pattern Classes | 43 |
| 4.4. | Recognition Scheme | 44 |
| 4.5. | Further Consideration in the Determination of Templates of Pattern Classes | 48 |

4.6. Implementation . . . . . 50
4.7. Further Consideration of Recognition Scheme . . . . . 53
5. Two-Summand Addition by an Associative Processing System . . . . 56
5.1. Introduction . . . . . 56
5.2. State Representation . . . . . 56
5.2a. State Representation of Summand Digit-Pair . . . . . 56
5.2b. State Representation of Summand-Pair . . . . . 57
5.2c. Classes of Summand . . . . . 57
5.2d. Allowable and Unallowable Addition Processes . . . . . 57
5.2e. State Representations of Allowable Addition Processes . . . . . 57
5.2f. State Representations of Unallowable Addition Processes . . . . 58
5.2g. State Representations of Merged Addition Processes . . . . . 60
5.3. Organization of Associative Processor for Arithmetic Operation . 60
5.3a. Associative Memory . . . . . 60
5.3b. Shift Registers . . . . . 60
5.3c. Sequential Machine . . . . . 63
5.4. Synthesis of Sequential Machine . . . . . 63
5.4a. Machine A Synthesized From State Representations . . . . . 63
5.4b. Disjoined Sequential Machines. . . . . 67
6. Discussion . . . . . 70
Appendix I . . . . . 71
Appendix II . . . . . 72
Bibliography . . . . . 75

ABSTRACT

Two different associative memory systems implemented by cryotrons and cutpoint cells are presented. The cutpoint cellular associative memory designed on the basis of stage delay analysis does not use any ladder structure, but employs a high-speed controlled shift register for the sequential tagging of all matched memory words. Both systems are so mechanized that their structures are simple and especially compatible with the batch fabrication of integrated circuits. Moreover, both systems can perform ordered information retrieval following Lewis's sorting scheme, template-matching pattern recognition based on the union-intersection concept and nonbulk two-summand additions of signed binary numbers in the signed-2's-complement number system. The basic operations of each system include comparison, reading, writing, sensing and output-summing processes being all performed on the basis of parallel-by-bit. During simultaneous comparison, a bit-position corresponding to all bit-memories of equal significance can be masked out. In pattern recognition, each pattern class is allowed to have deviations in size, style, orientation, etc. within certain limits. Since each prestored template for a pattern class can accommodate to a number of deviated input patterns belonging to the same class, the storage capacity can be greatly saved. Because most input patterns are recognized within first processing cycle and no patterns require more than two comparison processes for their recognition, the pattern recognition scheme is simple. The speed is very high because of this simple scheme as well as the parallel processing capability of an associative memory. The processing system for arithmetic operations consists of an associative memory, right shift registers and Mealy type sequential machines. This system may have the advantage of reducing addition time by paying a limited price in the additional hardware relative to the existing software approaches for arithmetic algorithms.

## 2. A CRYOGENIC ASSOCIATIVE MEMORY SYSTEM

### 2.1. INTRODUCTION

Among the existing associative memories, more than half of them has been implemented by cryotrons because a cryotron circuit is able to integrate the storage and logic functions together so that the structure of circuitry is simple and the internal power transfers can be held to a minimum degree. Furthermore, cryotrons are very reliable and extremely small, consume little power, and can easily be used with other circuit elements such as tunnel diodes [33].

The principal limitation on the use of cryogenic associative memory systems is their operating speed, which is usually slower than those implemented by other circuit elements, such as magnetic cores and cutpoint cells. However, this can be improved by simplifying the system organization, reducing the number of stage delays in every process, and developing better batch-fabrication techniques in manufacturing the cryogenic circuits [34]. Another disadvantage of a cryogenic system arises from the fact that the whole cryogenic circuit can operate only under low temperature (close to $3.5^{o}k$) and hence it must be stored in a liquid helium regrigerator. Since the cost of cryotron circuits is relatively low compared with other circuit elements, the expense of maintaining a liquid helium refrigerator may be absorbed when the capacity of a cryogenic associative memory system is large. In current thinking, it is reasonably expected that a future large-capacity high-speed associative memory system will be made of both cryotrons and semiconductor devices.

Since most of the existing cryogenic associative memory systems possesses the weakness of low operating speed and limited applications, the purpose of this chapter is to introduce the logical structure of a new cryogenic associative memory system which has the following advantages:

1). In ordered information retrieval, Lewin's sorting scheme [11] which requires a smallest number of interrogating cycles and is independent of the number of bits per word is employed.

2). The sequential writing process and the nonordered information retrieval can be performed automatically because a circuit for automatic sequential dematching operation is employed. Manual parallel dematching operation can be used for dematching all excessive empty words after sequential writing.

## 2. A CRYOGENIC ASSOCIATIVE MEMORY SYSTEM

### 2.1. INTRODUCTION

Among the existing associative memories, more than half of them has been implemented by cryotrons because a cryotron circuit is able to integrate the storage and logic functions together so that the structure of circuitry is simple and the internal power transfers can be held to a minimum degree. Furthermore, cryotrons are very reliable and extremely small, consume little power, and can easily be used with other circuit elements such as tunnel diodes [33].

The principal limitation on the use of cryogenic associative memory systems is their operating speed, which is usually slower than those implemented by other circuit elements, such as magnetic cores and cutpoint cells. However, this can be improved by simplifying the system organization, reducing the number of stage delays in every process, and developing better batch-fabrication techniques in manufacturing the cryogenic circuits [34]. Another disadvantage of a cryogenic system arises from the fact that the whole cryogenic circuit can operate only under low temperature (close to $3.5^{o}k$) and hence it must be stored in a liquid helium regrigerator. Since the cost of cryotron circuits is relatively low compared with other circuit elements, the expense of maintaining a liquid helium refrigerator may be absorbed when the capacity of a cryogenic associative memory system is large. In current thinking, it is reasonably expected that a future large-capacity high-speed associative memory system will be made of both cryotrons and semiconductor devices.

Since most of the existing cryogenic associative memory systems possesses the weakness of low operating speed and limited applications, the purpose of this chapter is to introduce the logical structure of a new cryogenic associative memory system which has the following advantages:

1). In ordered information retrieval, Lewin's sorting scheme [11] which requires a smallest number of interrogating cycles and is independent of the number of bits per word is employed.

2). The sequential writing process and the nonordered information retrieval can be performed automatically because a circuit for automatic sequential dematching operation is employed. Manual parallel dematching operation can be used for dematching all excessive empty words after sequential writing.

## 1. INTRODUCTION

The associative memory*, since its inception in 1956 [1], has manifested considerable interest and attention in the field of computer and information sciences and has been developed in a variety of aspects [1-30]. In contrast to the location or coordinate address as used in a random access memory, this type of storage stores and retrieves information on the basis of content address. In other words, a word is written into an associative memory without assigning an address and selecting a location, and is retrieved from the storage by simply searching a part or all of the content of that word. Because the housekeeping task inherent in a conventional memory system is reduced to a considerable extent, the operating speed of an associative memory system is increased and programming such a system is simplified. Since words in an associative memory are identified and addressed by their contents, an address decoder, an address register, and a selection matrix which are required in a conventional memory system are no longer used. Thus, the cost of the required distributed logic circuits, interrogating digit drivers, and some other necessary networks may be compensated by these eliminated devices. Another important attribute of an associative memory system arises from parallel processing capability because most operations can be performed on the basis of parallel manner. Therefore, a processor consisting of an associative memory offers attractive and efficient solutions to some difficult programming problems in a conventional digital computer, such as ordered information retrieval or sorting [3,6,7,11,19], dynamic storage allocation [10,14,28,31], and possibly table loop-up [10], list processing [10], etc. Moreover, pattern recognition [20,23,25,30], bulk processing including arithmetic algorithms [14,16,26,27,30], logical operations [12,14,17,25,26,29], and matrix inversion [27], and many other functions [20-23] can be easily and efficiently performed by the use of content addresses and parallel operations.

The associative memories and processors appeared in literature have been organized in different ways and implemented by using various

---

*The associative memory is also called the catalog, data-addressed, content-addressed or parallel-search memory.

types of circuit elements, such as cryotrons [1-3,7-10, 19, 24], magnetic components [4,5,13,25,29], semiconductor devices [15,17], and possibly plated wires [26]. Although various associative memory systems have been proposed, many schemes of the early work seem to be either too primitive or too complex to satisfy all the requirements of an associative memory system and to accomplish many applications by means of the same system.

In this thesis, we shall use cryotrons and cutpoint cells [32] for designing two associative memory systems in distinct logical structures. Both systems are so organized that their structures are simple and compatible with the batch fabrication of integrated circuits. Furthermore, both systems can perform ordered information retrieval following Lewin's sorting method [11], pattern recognition and nonbulk two-summand additions of signed binary numbers in the signed-2's-complement number system. For each of our systems, the basic operations include comparison, reading, writing, sensing and output-summing processes. The comparison process is performed by using an appropriate interrogating word for parallel searching all the memory words. During comparison, a bit-position corresponding to all bit-memories of equal significance can be masked out. All the other processes are also performed on the basis of parallel-by-bit. The sensing and the output-summing processes are respectively used for ordered retrieval and pattern recognition. In pattern recognition, each pattern class is allowed to have deviations in size, style, orientation, etc. within certain limits. Because each input pattern is processed with the prestored templates for all the pattern classes simultaneously, the speed is very high. It is found that most input patterns are recognized within first comparison process and no patterns require more than two processing cycles for their recognition. Since each template can accomodate to a number of deviated input patterns belonging to the same class, the storage capacity can be greatly saved. The proposed hardware approach of the nonbulk two-summand additions may have the advantage of reducing the addition time relative to the existing software approaches for arithmetic algorithms, because of possible parallel operations.

In this thesis, chapters 2 and 3 contain respectively the cryogenic and the cutpoint cellular associative memory systems. Chapter 4 presents

the pattern recognition scheme by using an associative processor. Chapter 5 describes the arithmetic operation. A selected bibliography showing the progress of associative memories and processors in the past decade is included.

3). The logical structure of the whole system is much simpler because multiple-controlled cryotrons are used and the operations of some processes are automatic. Every bit-memory and every cryogenic selection and control circuit for each word contain fewer cryotrons compared with the existing systems on the basis of performing equivalent functions.

4). The variation of the circuit inductances for different tagged words during reading and writing and the number of stage delays for each process are both small. Hence, the time interval between the pulses for automatic reading and writing can be reduced (because the time interval has to be designed according to the word with the largest circuit inductance), and the operating speed is consequently increased.

5). In contrast to the ordinary memory systems in which the information bits are restricted to be "0" and "1", this memory may contain an additional empty "e" besides "0" & "1". Since we may have "e" in an interrogating word, the destructive reading procedure is simplified*. It is noted that in ordered retrieval, "e" is allowed to appear only in the equal significant bit-position of all the occupied words because "e" is undefined as far as its magnitude is concerned; however, there is no such restriction in non-ordered retrieval.

6). The reading process may be either destructive or nondestructive.

In this chapter, we shall first present the logical structure of the cryogenic associative memory, and then describe the operations of each process separately. The technical terms employed in the following sections can be found in Appendix I.

---

*In ordinary cryogenic associative memory, the interrogating word can have only the "0" and "1" bits. In order to search the empty words, the interrogating word used is a word consisting of "0" only. There are two different ways to differentiate the "0" and "1"; One is to use a nonzero persistent current representing "1" and a zero persistent current representing "0"; and the other is to use the two different directions of the persistent current to differentiate "0" and "1". In the former case, there is no way to distinguish an empty bit-memory and a memory occupied by a "0" information bit, and hence ambiguity may arise. In the latter case, during the destructive reading process, the information words which have been read out are first erased, and then we have to write the "0" bits in their bit-memories. If we can have "e" in the interrogating word, an empty word can merely be a memory register in which each bit-memory has zero persistent current. This eliminates the writing step in the destructive reading process.

## 2.2. THE LOGICAL STRUCTURE

The arrangement of the n bit-memories and a selection and control circuit for each of the m memory words of the proposed cryogenic associative memory system is shown in Fig. 1. The memory words are numbered from the top to the bottom. The logical structure of a typical memory word including a selection and control circuit and n bit-memories is shown in Fig. 2. Since there are some differences in the connection of the selection and control circuit for an odd-numbered word and an even-numbered word, the typical word i shown in Fig. 2 is assumed to be odd-numbered. If the word i becomes even-numbered, the $C_M$ and $M_C$ lines should be the dotted lines marked as $C_M^*$ and $M_C^*$ shown in Fig. 2. Now we describe each part of the memory system separately.

### 2.2a. Bit-Memories

As shown in Fig. 2, each bit-memory consists of 5 cryotrons, 16 connecting terminals and one memory loop. The comparison, enable-mask, read-sense, auxiliary sense, and write cryotrons in the bit-memory ij are denoted by $C_{ij}$,$E_{ij}$,$R_{ij}$,$S_{ij}$, and $W_{ij}$ respectively. We also use these symbols to represent the binary variables, each of which may be either "0" or "1" indicating the corresponding cryotron-gate being in the resistive or superconductive state respectively. From now on, if a symbol is used as the label of a cryotron or a line containing cryotron-gates, this symbol, when it appears in an equation, is also used as a binary variable being either "0" or "1" indicating the corresponding cryotron-gate or the corresponding line being resistive or superconductive. If a symbol is used as the label of a pulse type or d-c current, this symbol, when it appears in an equation, is also used as a binary variable being either"0" or "1" indicating the corresponding current being respectively smaller or larger than the critical current which keeps all the cryotron-gates controlled by that current in the resistive state. It is noted that the nonzero persistent current $I_p$ circulating in every memory loop should be larger than the critical current of $C_{ij}$ but smaller than the critical currents of $S_{ij}$ and $R_{ij}$. Furthermore, because of the arrangement of the lines $W_{Si}$ and $W_{Ri}$, the mutual induced currents in every memory loop due to the multiple-controlled cryotrons $S_{ij}$ and $R_{ij}$ are cancelled, and hence there is no unsatisfactory result caused by these induced currents.

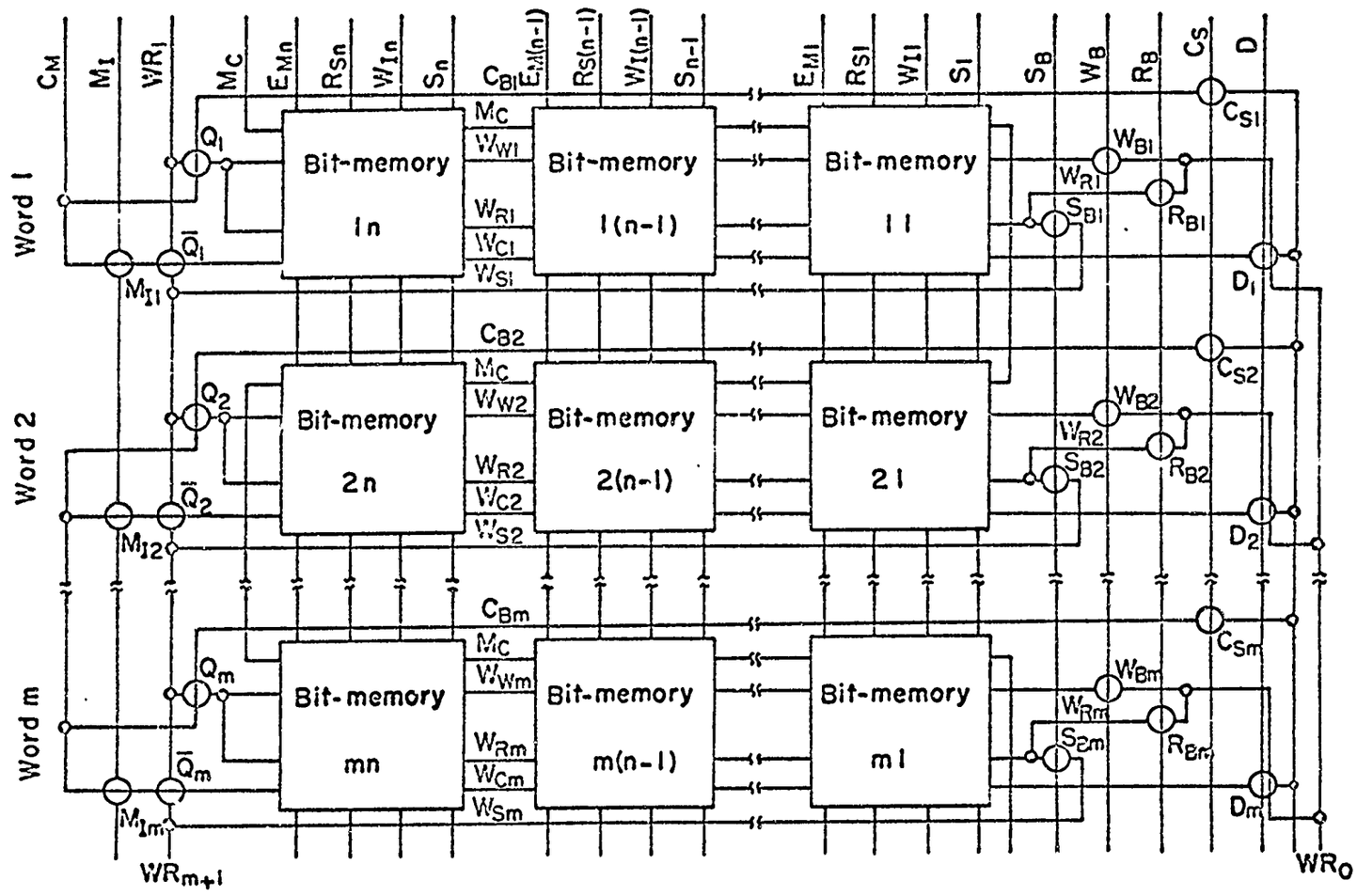



Fig. 1. The logical structure of the proposed cryogenic associative memory system.

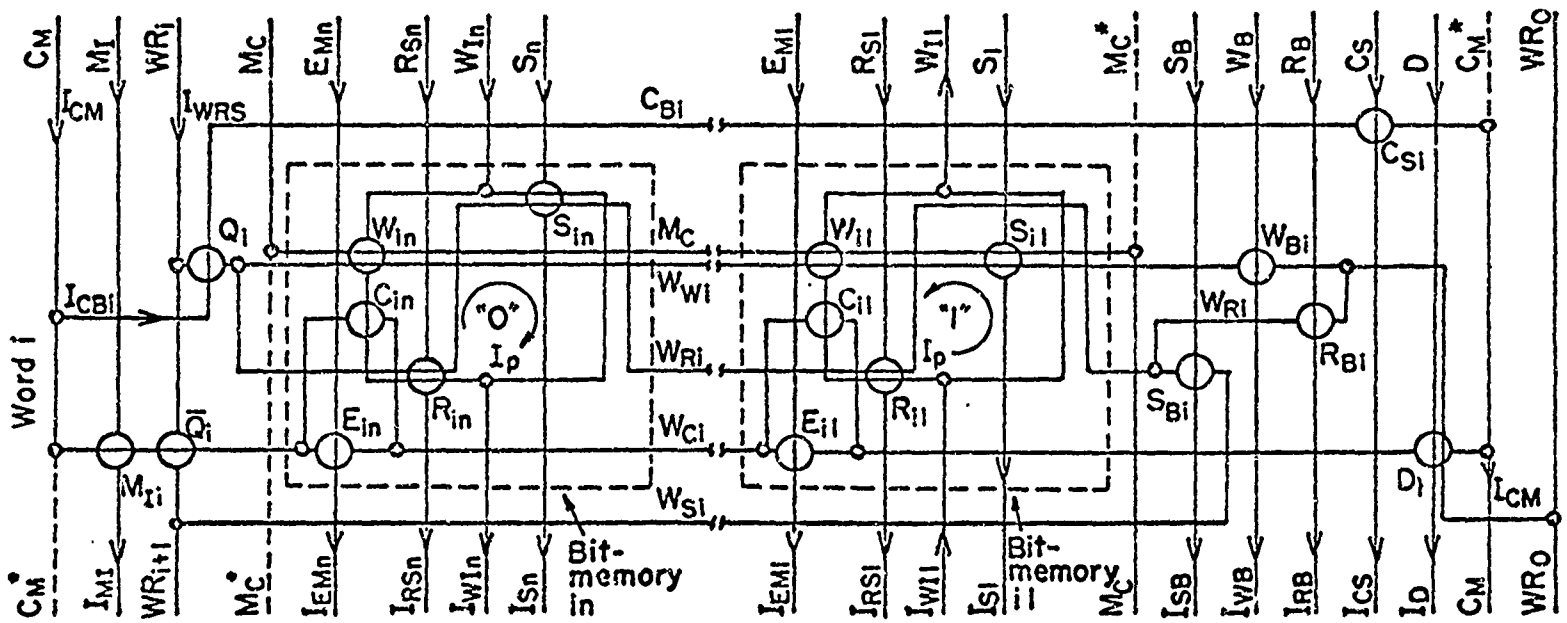



Fig. 2. The logical structure of each memory word.

In Figs.1 and 2, we see that the controls of the enable-mask cryotrons $E_{1j}$ through $E_{mj}$ are connected in series to form the enable-mask line $E_{Mj}$ for the j-th bits of all the words. The read-sense line $R_{Sj}$ and the auxiliary sense line $S_j$ are formed in the similar manner by connecting the gates of the read-sense cryotrons $R_{1j}$ through $R_{mj}$ in series and $S_{1j}$ through $S_{mj}$ in series respectively. Hence, we have the following equations:

$$R_{Sj} = \prod_{i=1}^{m} R_{ij}, \tag{1}$$

$$S_j = \prod_{i=1}^{m} S_{ij}, \tag{2}$$

where j = 1,2,...,n, and the operation "multiplication" is the same as Boolean product. In fact, in all the equations, the operations addition and multiplication and the bar over a binary variable are the same as the addition, multiplication and complementation of a binary variable in Boolean algebra respectively. Let $I_{RSj}$ be the current in the read-sense line $R_{Sj}$ and $I_{Sj}$ in the auxiliary sense line $S_j$, and $V_{RSj}$ be the voltage across $R_{Sj}$ and $V_{Sj}$ across $S_j$. Then, $V_{RSj}$ and $V_{Sj}$ can be expressed by the following equations:

$$V_{RSj} = \bar{R}_{Sj} I_{RSj}, \tag{3}$$

$$V_{Sj} = \bar{S}_j I_{Sj}, \tag{4}$$

for j = 1,2,...,n. The line which connects all the memory loops in the j-th bit-column forms the write-interrogate line $W_{Ij}$ which can be excited by the current pulse $I_{WIj}$ during the interrogating or writing period. The magnitude of $I_{WIj}$ should be twice of the persistent current $I_p$ for the bit of "0" or "1", and zero for "e". The positive (upward) or negative (downward) pulse of $I_{WIj}$ corresponds to the bit "1" or "0" respectively. This convention is shown in Table 1.

2.2b. Word Selection and Control Circuits

For each word, there is a selection and control circuit in addition to the n identical bit-memories. This circuit for each word consists of 8 cryotrons and their associated lines as shown in Fig. 2. The complementary pair of the sequencing cryotrons $Q_i$ and $\bar{Q}_i$ is used to tag the i-th word for performing the subsequent sensing, reading or writing process if word i is matched by a comparison process. Their gate-states

Table 1.

Designations of the Writing, the Interrogating and the Stored Information Bits

| Bit | Writing bit | Interrogating bit | | Stored information bit |
|---|---|---|---|---|
| | $I_{WIj}$ | $I_{WIj}$ | $I_{EMj}$ (of either polarity) | |
| 0 | $-2I_p$ (downward) | $-2I_p$ (downward) | 1 | Clockwise $I_p$ |
| 1 | $+2I_p$ (upward) | $+2I_p$ (upward) | 1 | Counterclockwise $I_p$ |
| e | Zero pulse | Zero pulse | 1 | Zero $I_p$ |
| Ø | Does not exist | Zero pulse | 0 | Does not exist |

and that of the match-indication cryotron $M_{Ii}$ are controlled by the comparison current $I_{CM}$ in the comparison line $C_M$. The cryotrons $S_{Bi}$ , $W_{Bi}$ and $R_{Bi}$, which are called the sense-inhibit, write-inhibit, and read-inhibit cryotrons are used respectively for the inhibition of the sensing, writing, and reading circuits of the i-th word. The controls of the sense-inhibit cryotrons $S_{B1}$ through $S_{Bm}$ are connected in series to form the sense-inhibit line $S_B$, which is excited by the current $I_{SB}$ in $S_B$ of either polarity during the sense-inhibition. The write-inhibit line $W_B$ and the read-inhibit line $R_B$ are formed and function in the similar manner. Furthermore, the cryotron-gates satisfy the following conditions:

$$S_{Bi}R_{Bi} = 0$$
$$S_{Bi}W_{Bi} = 0 \qquad (5)$$
$$R_{Bi}W_{Bi} = 0$$

and

$$\bar{S}_{Bi}\bar{R}_{Bi}\bar{W}_{Bi} = 0$$

for i = 1,2,...,m. The cryotron $C_{Si}$ in the bypass branch $C_{Bi}$ of the word-comparison line $W_{Ci}$ of word i is called the comparison-set cryotron. The controls of $C_{S1}$ through $C_{Sm}$ are connected in series to form the comparison-set line $C_S$. The cryotron $D_i$, called the dematching cryotron of word i has a double-control and is used to dematch the selected i-th word. One of the controls of $D_i$ connected between the terminal $WR_o$ and the common terminal of the gates of $W_{Bi}$ and $R_{Bi}$ is used for performing the automatic dematching operation. The other controls of $D_1$ through $D_m$ are connected in series to form the dematching line D, which is used for performing the manual dematching operation.

2.2c. Word-Write-Read-Sense Circuit

The word-sense circuit of word i is the circuit between the terminals $WR_i$ and $WR_{i+1}$, which consists of two parallel branches: One is the word-sense line $W_{Si}$ containing the gates of the cryotrons $Q_i$ and $S_{Bi}$ and the controls of the cryotrons $R_{i1}$ through $R_{in}$ and $S_{i1}$ through $S_{in}$, and the other is the gate circuit of the cryotron $\bar{Q}_i$. The state of the i-th-word-sense circuit $W_{SCi}$ is given by the equation

$$W_{SCi} = \bar{Q}_i + S_{Bi} \qquad (6)$$

for i = 1,2,...m, provided that (5) is satisfied.

The word-write-read circuit of word i starts from the terminal $WR_i$ to the terminal $WR_o$. It includes the gate of the cryotron $Q_i$, the parallel connection of the word-write line $W_{Wi}$ and the word-read line $W_{Ri}$, and one of the controls of $D_i$. The word-write line $W_{Wi}$ is the series connection of the controls of $W_{i1}$ through $W_{in}$ and the cryotron-gates $W_{Bi}$ and $Q_i$. The word-read line $W_{Ri}$ consists of the cryotron-gates $R_{Bi}$ and $Q_i$ and the part of the word-sense line $W_{Si}$ which is the series connection of the controls of $R_{i1}$ through $R_{in}$ and $S_{i1}$ through $S_{in}$ shown in Fig. 2. The function $W_{WRi}$ of the word-write-read circuit between the terminals $WR_o$ and $WR_i$ can be expressed logically as follows:

$$W_{WRi} = (Q_i + S_{Bi})\ (W_{Bi} + R_{Bi}), \tag{7}$$

for i = 1,2,...,m.

Figs.3 and 4, in which the shaded or unshaded circle represents respectively the resistive or superconductive state of a cryotron-gate, show how the write-read-sense current $I_{WRS}$ will flow in the superconductive word-write-read-sense circuit for the first matched word and a mismatched word respectively. In each case, all the three possible operations are illustrated. It is seen that there is one and only one superconductive path for $I_{WRS}$ passing through each word under all the circumstances. It follows from (5) and (7) that

$$W_{WRi} = Q_i \overline{S}_{Bi} \tag{8}$$

for i = 1,2,...,m. Because (6) and (8) are exactly complementary to each other and all the word-write-read-sense circuits are connected as shown in Fig. 1, the transmission function $f_o$ between the terminals $WR_1$ and $WR_o$ and $f_{m+1}$ between $WR_1$ and $WR_{m+1}$ are given by the following equations:

$$f_o = \sum_{i=1}^{m} \overline{W}_{SCi} = \sum_{i=1}^{m} Q_i \overline{S}_{Bi} \tag{9}$$

$$f_{m+1} = \prod_{i=1}^{m} W_{SCi} = \prod_{i=1}^{m} (\overline{Q}_i + S_{Bi}) \tag{10}$$

The above two equations which are complementary to each other can be expressed in terms of the currents $I_{SB}$ and $I_{CBi}$ passing through the sense-inhibit line $S_B$ and the bypass branch $C_{Bi}$ of the i-th word comparison line $W_{Ci}$ as follows:

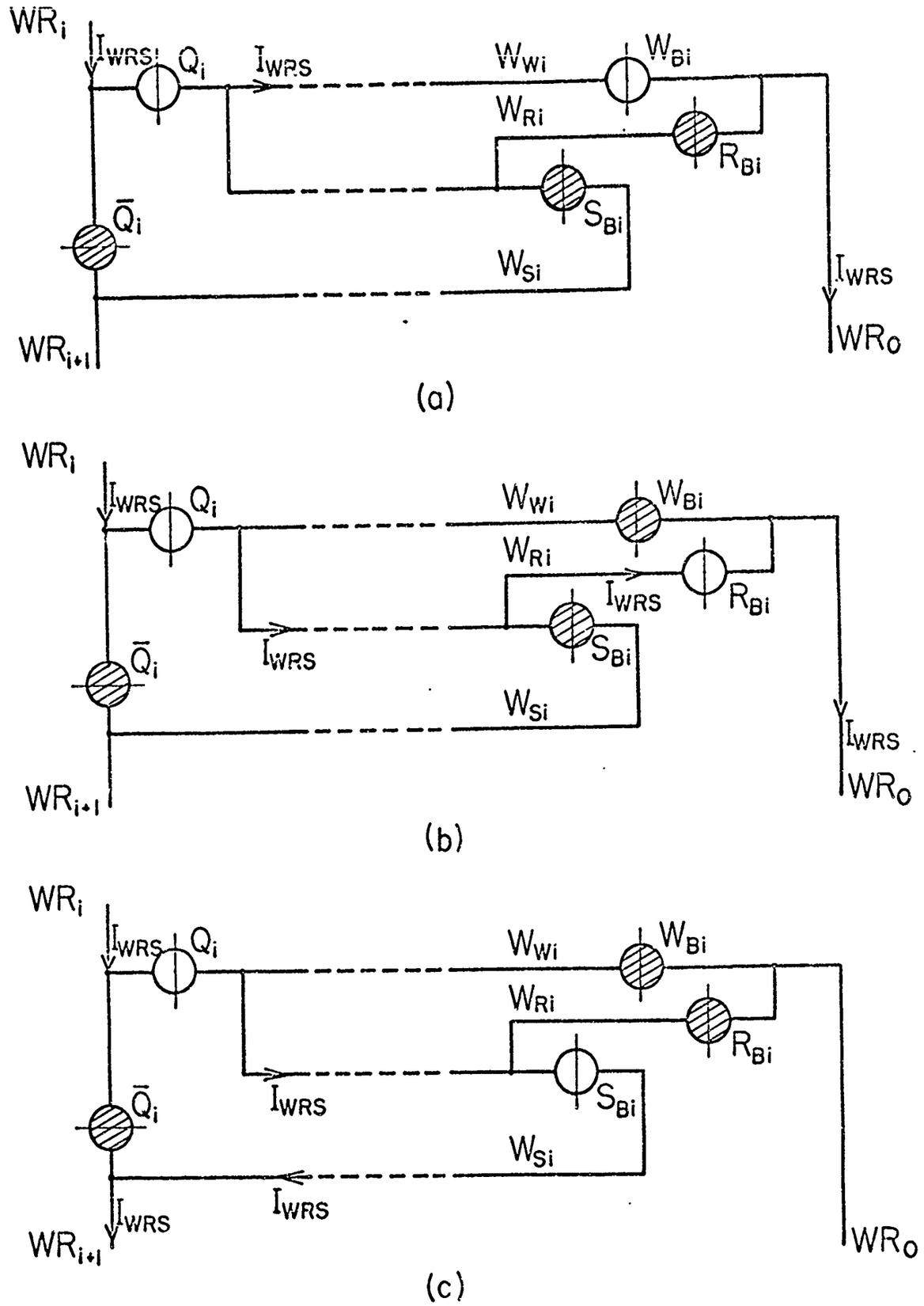



Fig. 3. The states of the cryotron-gates and the flow of the current pulse $I_{WRS}$ in the superconductive path of the word-write-read-sense circuit of the first matched word i, when the circuit (a) $W_{Wi}$, (b) $W_{Ri}$ or (c) $W_{Si}$ is selected.

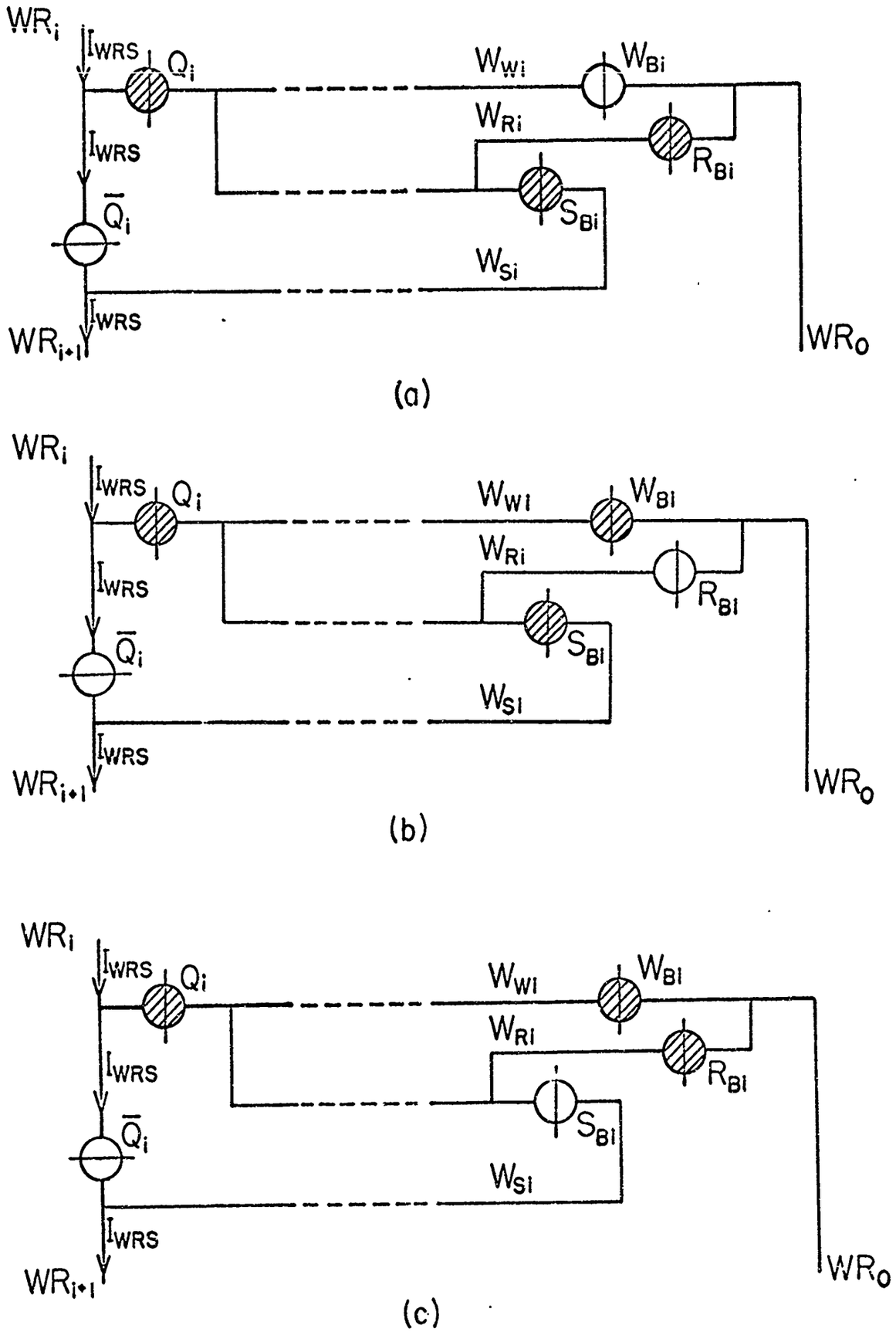



Fig. 4. The states of the cryotron-gates and the flow of the current pulse $I_{WRS}$ in the superconductive path of the word-write-read-sense circuit of a mismatched word labelled with a smaller number than that of the first matched word when the circuit (a) $W_{Wi}$, (b) $W_{Ri}$ or (c) $W_{Si}$ is selected.

$$f_o = \sum_{i=1}^{m} \bar{I}_{CBi} I_{SB} \tag{11}$$

$$f_{m+1} = \prod_{i=1}^{m} (I_{CBi} + \bar{I}_{SB}). \tag{12}$$

2.2d. Comparison Circuit

The comparison circuit consists of m word-comparison circuits. The word-comparison circuit of word i is the parallel connection of the word-comparison line $W_{Ci}$ and its bypass $C_{Bi}$ as shown in Fig. 2, and its logic equation is given by

$$W_{CMi} = D_i \prod_{j=1}^{n} (C_{ij} + E_{ij}) + C_{Si}, \tag{13}$$

for i = 1,2,...,m. This equation can be expressed in terms of the currents controlling the cryotrons in the word-comparison circuit as follows:

$$W_{CMi} = \bar{I}_D \bar{I}_{WRS} \prod_{j=1}^{n} (\bar{I}_{Cij} + \bar{I}_{CMj}) + \bar{I}_{CS}, \tag{14}$$

where $I_{Cij}$ is the net current passing through the control of $C_{ij}$, which is the algebraic sum of $I_{WIj}/2$ due to the current pulse $I_{WIj}$ applied to $W_{Ij}$ and the persistent current $I_p$ in the bit-memory ij. It is noted that due to the convention of $I_p$ and $I_{WIj}$ for the information bit we have defined in Table 1, the cryotron-gate $C_{ij}$ is in the superconductive state if the stored information $I_p$ in the bit-memory ij and the interrogating information $I_{WIj}$ are matched and resistive otherwise. We know the current in the control of $C_{ij}$ due to the interrogating current source in the j-th column is $I_{WIj}/2$ because the two sides of the memory loop between the connections of the write-interrogate line $W_{Ij}$ have the same inductance when the cryotron-gate $W_{ij}$ is superconductive.

The comparison circuit is the series connection of the m word-comparison circuits $W_{CM1}$ through $W_{CMm}$. Its logic equation $C_M$ is given by

$$C_M = \prod_{i=1}^{m} \left[ \bar{I}_D \bar{I}_{WRS} \prod_{j=1}^{n} (\bar{I}_{Cij} + \bar{I}_{CMj}) + \bar{I}_{CS} \right]. \tag{15}$$

2.2e. Match-Indication and Memory-Clear Circuit

The match-indication line $M_I$ is the series connection of the cryotron-gates $M_{I1}$ through $M_{Im}$ as shown in Fig. 1. The match-indication voltage $V_{MI}$ across the line $M_I$ is equal to $\omega R_{MI} I_{MI}$, where $\omega$ is the number of matched words, $I_{MI}$ is the constant current applied to the $M_I$ line, and $R_{MI}$ is the resistance of a match-indication cryotron-gate $M_{Ii}$ when it is in the resistive state.

The memory-clear line $M_C$ is the series connection of one of the controls of all the write cryotron $W_{ij}$ in the memory as shown in Figs. 1 and 2.

## 2.3. OPERATIONAL PROCEDURES

### 2.3a. Comparison Process

The comparison process is required to select memory words for reading, writing or sensing process. We shall describe various comparison techniques due to different interrogating words, and the procedure for comparison in this section.

When the interrogating word consists of only the don't-care bits "Ø", the comparison process is called the masked comparison. It may be employed during the initial interrogating cycle of the ordered retrieval. We shall see that by this process all the descriptors "0", "1" or "e" of the matched words can be identified respectively by the "sense 0", "sense 1" or "sense Y" at the double-sense line $R_{Sj}$ and $S_j$. When the interrogating word consists of "0", "1" and "e" but no "Ø", the process is called the enabled comparison. This method is used when a particular record is to be retrieved from a large file stored in the memory system. The comparison result is indicated by either "matching" or "mismatching". When the interrogating word consists of "0", "1", "e" and "Ø", the process is called the enabled-masked comparison. In fact, a bit column containing all equal significant bits of the memory words is said to be masked out during a comparison process if the corresponding interrogating digit for the comparison process is "Ø". This method is usually employed in most comparison processes. It is noted that the "e" in the interrogating word for ordered retrieval is not used because "e" is restricted to be a descriptor and it is interrogated by a "Ø ".

The comparison process starts with applying a current pulse $I_{CS}$ to the comparison-set line $C_S$. Then, all the cryotron-gates $C_{S1}$ through $C_{Sm}$ become temporarily resistive, and the d-c comparison current $I_{CM}$ is compelled to flow or remains flowing through the word-comparison lines $W_{C1}$ through $W_{Cm}$ because at this time there are no currents applied to the enable-mask lines $E_{M1}$ through $E_{Mn}$. The memory words are then interrogated by the current pulses $I_{WI1}$ through $I_{WIn}$ with magnitudes and directions being determined by the interrogating word as defined in Table 1. It is seen that the cryotron-gate $C_{ij}$ will change to the resistive state whenever the net control current $I_{Cij}$ of $C_{ij}$ reaches $I_{WIj}/2$. After all the necessary state-transitions are completed, the enable-mask current $I_{EMj}$ is

applied to each enable-mask line $E_{Mj}$ according to the corresponding interrogating digit. It follows from the current conventions we have chosen that there is at least one pair of the cryotron-gates $C_{ij}$ and $E_{ij}$ being both resistive if and only if the i-th word is not matched. In this case, $I_{CM}$ is switched to the bypass branch $C_{Bi}$ of $W_{Ci}$ and the gates of $\bar{Q}_i$ and $M_{Ii}$ become superconductive. On the other hand, if the i-th word is matched, $W_{Ci}$ remains superconductive and $I_{CM}$ is kept flowing through $W_{Ci}$, and consequently the gates of $\bar{Q}_i$ and $M_{Ii}$ remain resistive. Therefore, the voltage across the gate of $M_{Ii}$ being nonzero or zero indicates whether the i-th word is matched or mismatched. After all the pulses $I_{WIj}$ and $I_{EMj}$ vanish, the comparison process terminates and the d-c comparison current $I_{CM}$ has established its distribution to control the gate-states of $Q_1$ through $Q_m$ and $\bar{Q}_1$ through $\bar{Q}_m$, which selects all the matched words for subsequent writing, reading or sensing operation.

The time sequences shown in the left part of Fig. 5 illustrate the comparison process for selecting the empty words (assume the i-th word is empty). In Fig. 5, the switching time of a cryotron is denoted by T with the label of the cryotron as its subscript.

2.3b. Writing Process

The process of storing information into an associative memory is called writing. To do writing, we first have to find all the empty words, each of which consists of only the empty "e". This can be accomplished by an enabled comparison with a string of "e" as the interrogating word. If there are no empty words, we can clear the occupied words by exciting the memory clear line $M_C$.

After locating the empty words, the writing circuit is selected by exciting the sense-inhibit line $S_B$ and the read-inhibit line $R_B$ and leaving only the write-inhibit line $W_B$ unexcited. The situation for the first matched word and a mismatched word are shown in Figs. 3(a) and 4(a) respectively. The information to be stored is written in by applying the current pulses $I_{WI1}$ through $I_{WIn}$ to the write-interrogate lines $W_{I1}$ through $W_{In}$ according to the convention shown in Table 1. At the same time a current pulse $I_{WRS}$, whose source is turned on by the match-indication voltage $V_{MI}$, is fed to the terminal $WR_1$. The cryotron-gates $W_{i1}$ through $W_{in}$ in the first matched empty word i become temporarily resis-

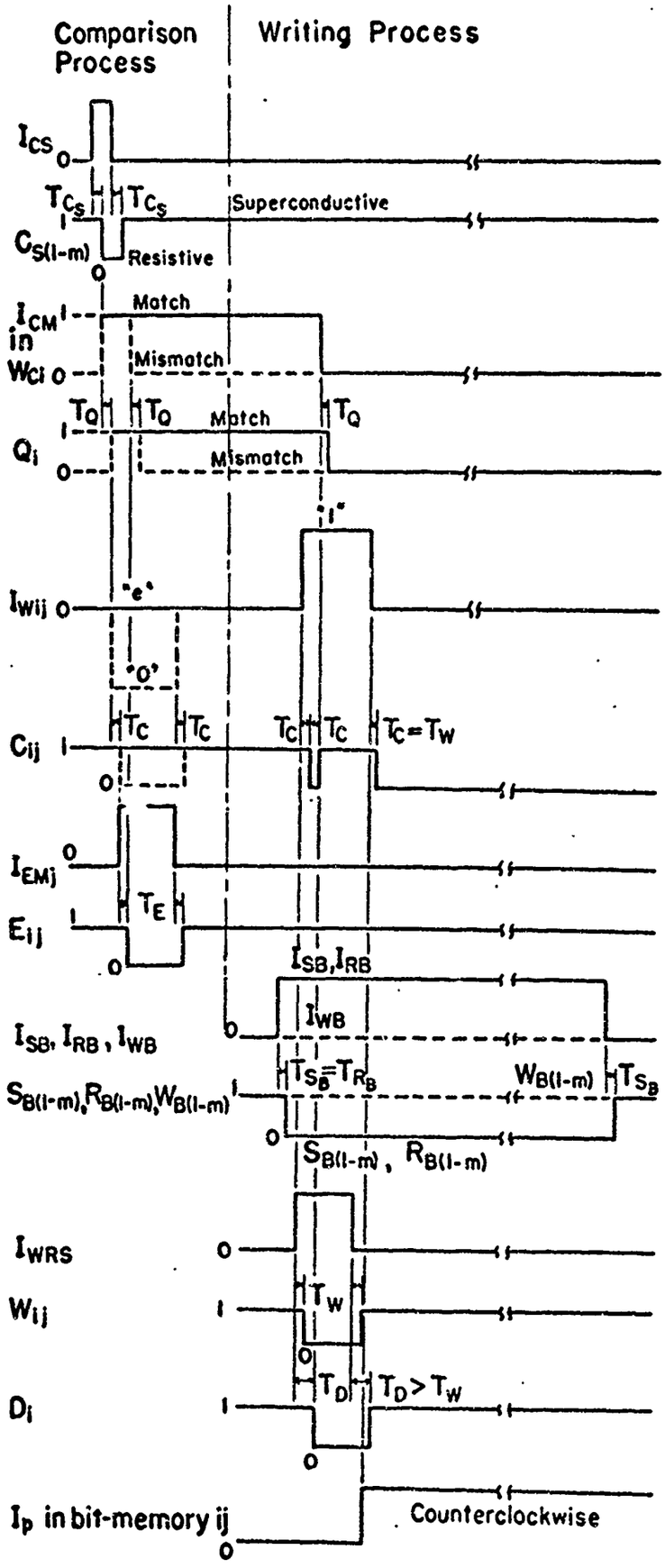



Fig. 5. Time sequences for comparison and writing processes.

tive and hence each of the current pulses $I_{WI1}$ through $I_{WIn}$ passes through the right branch of each memory loop. When $W_{i1}$ through $W_{in}$ become superconductive again $I_{WI1}$ through $I_{WIn}$ vanish and the appropriate persistent currents in word i are established. The newly occupied word i is automatically dematched after the dematching cryotron $D_i$ is changed from superconductive state to resistive state by $I_{WRS}$. The writing procedure is then repeated for the next matched empty word. The process terminates after all the new information has been written in or after all the matched empty words are occupied. In the former case, the dematching line D may be excited at the instant of completing the writing of new information; and in the latter case, the match-indication voltage $V_{MI}$ is reduced to zero, and the $I_{WRS}$ source is turned off. It is noted that the switching time $T_D$ of $D_i$ must be greater than $T_W$ of $W_{ij}$ in order to complete the writing procedure before the word i is dematched. In fact, it will be seen later that $T_D$ also should be greater than $T_R$ and $T_S$. The time sequences shown in the right-hand part of Fig. 5 illustrate the writing procedure in which a bit "1" is written in the matched empty bit-memory ij.

2.3c. Non-Ordered Retrieval

To do non-ordered retrieval, an enabled-masked comparison is first performed to match all the desired words to be retrieved from the memory system. This is done by using the common identifier of all the desired words to be retrieved as the enabled part of the interrogating word. Then, the word-read circuit is selected by exciting the write-inhibit line $W_B$ and the sense-inhibit line $S_B$ and leaving only the read-inhibit line $R_B$ unexcited. The situations for various words are shown in Figs. 3(b) and 4(b). Then applying the current pulse $I_{WRS}$ to the terminal $WR_1$ and d-c currents to all the double-sense lines $R_{S1}$ through $R_{Sn}$ and $S_1$ through $S_n$, the first matched word i is read out by measuring the voltages across all the double-sense lines. It follows from the convention defined in Table 2 that the values (0,1), (1,0) and (0,0) of the voltage pair ($V_{RSj}$, $V_{Sj}$) appearing in the double-sense line $R_{Sj}$ and $S_j$ during the non-ordered reading process correspond to "0", "1" and "e" respectively. We can see that only $V_{RSj}$ needs to be measured if each bit-memory in the tagged word contains either "0" or "1". The word i is automatically dematched after the dematching cryotron $D_i$ is changed from superconductive state to

Table 2.

Sensing Results of Cryogenic Associative Memory

| Type of sense | Sensing voltages | | Lewin's assignment | Our assignment |
|---|---|---|---|---|
| | $V_{RSj}$ | $V_{Sj}$ | Contents of j-th bit-memories in tagged words | Contents of j-th bit-memories in tagged words |
| 0 | 0 | 1 | 0 | 0; or 0 and e |
| 1 | 1 | 0 | 1 | 1; or 1 and e |
| X | 1 | 1 | 0 and 1 | 0 and 1; or 0, 1 and e |
| Y | 0 | 0 | No words are matched | e |

resistive state by the first current pulse $I_{WRS}$, and the next current pulse $I_{WRS}$ will go to the next matched word. Thus, all the matched words are read out sequentially according to the ascending order of the matched-word-position numbers. After all the matched words are read out, the match-indication voltage $V_{MI}$ is reduced to zero which turns off $I_{WRS}$ and the d-c currents to $R_{S1}$ through $R_{Sn}$ and $S_1$ through $S_n$, and the reading process ends.

It is noted that the above reading process is non-destructive. For destructive reading, the procedure is to excite the sense-inhibit line $S_B$ and the read-inhibit line $R_B$ and leaving the write-inhibit line $W_B$ un-excited. When the current pulse $I_{WRS}$ is applied to the terminal $WR_1$, the voltage appearing on $W_{Ij}$ is zero or a non-zero pulse of either polarity according to whether the information in the j-th bit-memory of the first matched word i is "e", "0" or "1" respectively. The polarity of a non-zero voltage pulse on $W_{Ij}$ representing "0" or "1" can easily be determined by the convention of the persistent current we have used. It is noted that when $I_{WRS}$ passes through $W_{Wi}$, the persistent current in every bit-memory of word i is reduced to zero because every cryotron-gate $W_{ij}$ in word i becomes resistive. Same as the non-destructive reading process, the i-th word is dematched as soon as the dematching cryotron $D_i$ is changed from superconductive state to resistive state by $I_{WRS}$, and the next pulse $I_{WRS}$ will go to the next matched word. The method of using the word-write line $W_{Wi}$ for destructive reading is first employed by Ahrons [19].

2.3d. Ordered Retrieval

For ordered retrieval, a sorting scheme is required to read out all the desired words sequentially in an ascending (or descending) order of the magnitudes of their binary equivalent values after all the desired words are matched by a comparison process. The sorting scheme used in this system is described in the flow chart shown in Fig. 6, which is the same as Lewin's method [11], except that the sensing results are interpreted differently in order to include the possibility of the appearance of "e" in the desired word contents.

Before we discuss the details of ordered retrieval in our system, let us consider another process, namely the sensing process, which is required in ordered retrieval. This process is used to generate the

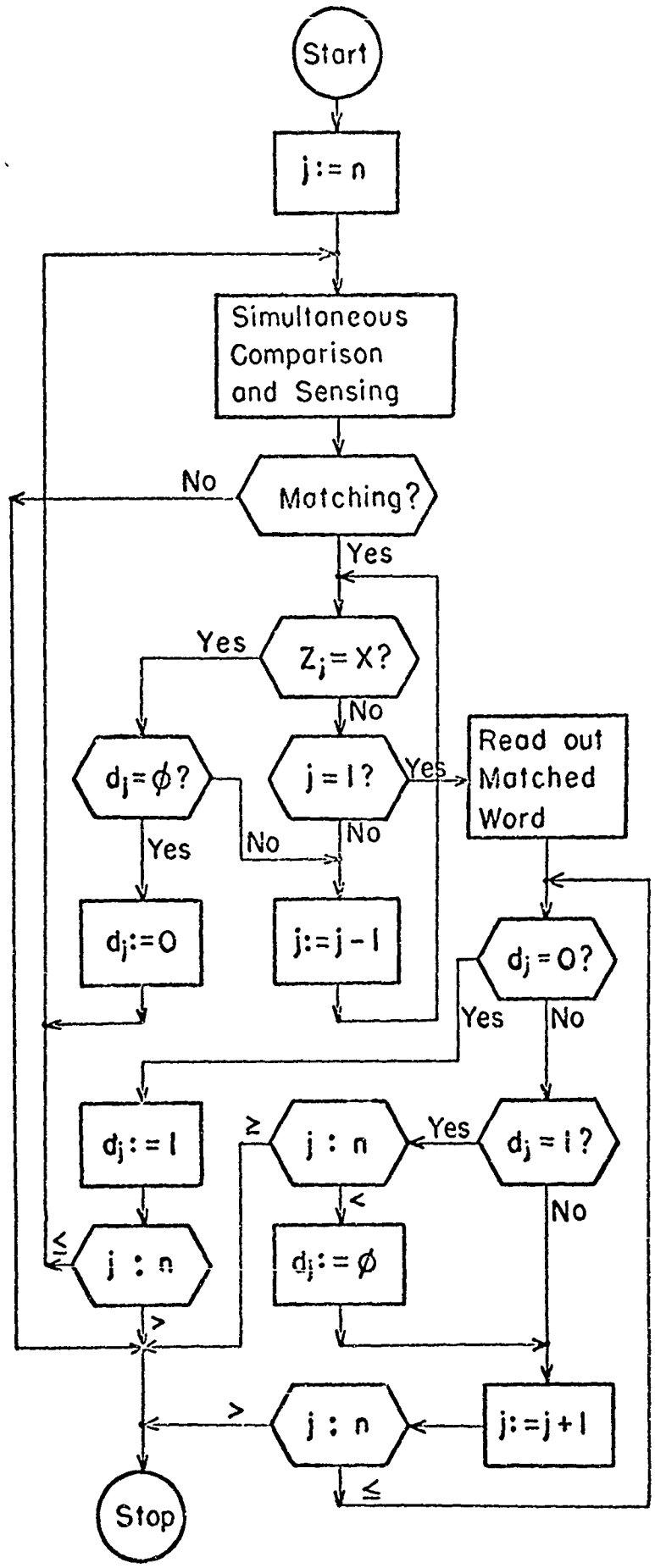



Fig. 6. A flow chart for ordered retrieval.

sensing result which indicates the contents of all the matched words of a given comparison process. This process can be described as follows: After a given comparison process, the word-sense circuit is selected by exciting the read-inhibit line $R_B$ and the write-inhibit line $W_B$ and leaving only the sense-inhibit line $S_B$ unexcited. The situations for the 1st matched word and a mismatched word are shown in Figs. 3(c) and 4(c) respectively. Then, the current pulse $I_{WRS}$ is applied to the terminal $WR_1$ and d-currents to all the double-sense lines $R_{S1}$ through $R_{Sn}$ and $S_1$ through $S_n$. The voltage pair $(V_{RSj}, V_{Sj})$ across the double-sense line $R_{Sj}$ and $S_j$ may have four different values, whose corresponding bit contents are shown in Table 2. Lewin's assignment for these values is also listed in order to compare the one used in our system. It is seen that the possibility of "e" appearing in the contents of matched occupied words is included in our assignment. The sensing process terminates when the current pulses $I_{WRS}$ and those in all the double-sense lines vanish.

The sorting scheme for ordered retrieval shown in Fig. 6 consists of $2\omega$-1 interrogating cycles for $\omega$ matched occupied words. Each interrogating cycle consists of a comparison process and a sensing process. The first interrogating cycle starts with the comparison process to match all the desired words to be retrieved in the memory system. In this comparison process, the interrogating word must contain some "Ø" in its bits and all others are either "0" or "1". Each initial interrogating digit $d_j$ which is not "Ø" must be a common identifier digit of all matched words, and hence it has no effect on the ordered retrieval. Therefore, we only consider those bit-positions which are "Ø" in the interrogating word. After each comparison process,we apply a sensing process to obtain the sensing results $Z_1$ through $Z_n$ for all bit-columns which are interpreted according to Table 2. Because "e" must be a descriptor in ordered retrieval, ambiguity arises only when "X" appears in the sensing result. If we want to retrieve all the desired words in the ascending order of the magnitudes of the binary values of their word contents, "Ø" is changed to "0" starting from the left to the right until no "X" appear in the sensing result. At this stage, the word with the minimum magnitude of word content is matched. After the matched word is read out, the same digits are changed from "0"→"1",→"Ø",→"0",→"1" in the interrogating

words for the subsequent interrogating cycles, starting from the right to the left. This procedure is expressed in the flow chart shown in Fig. 6, and an example is given in Table 3 which retrieves out all the stored words. If we want to retrieve all desired words in the descending order of the magnitudes of the binary values of their word contents, the procedure is the same except that "Ø" is first changed to "1", and then "1"→"0",→"Ø",→"1",→"0".

It is noted that the reading process used in the ordered retrieval may be non-destructive or destructive. Their operations are the same as in the non-ordered retrieval case.

Table 3.

An Illustrating Example for Ordered-Retrieval

| Interrogating cycle | Interrogating Word $d_5d_4d_3d_2d_1$ | Sensing result $Z_5Z_4Z_3Z_2Z_1$ | Matched occupied words | Word content being retrieved |
|---|---|---|---|---|
| 1 | Ø Ø Ø Ø Ø | X 0 X Y 1 | 1,2,3,4 | |
| 2 | 0 Ø Ø Ø Ø | 0 0 X Y 1 | 1,3 | |
| 3 | 0 Ø 0 Ø Ø | 0 0 0 Y 1 | 1 | 0 0 0 e 1 |
| 4 | 0 Ø 1 Ø Ø | 0 0 1 Y 1 | 3 | 0 0 1 e 1 |
| 5 | 1 Ø Ø Ø Ø | 1 0 X Y 1 | 2,4 | |
| 6 | 1 Ø 0 Ø Ø | 1 0 0 Y 1 | 2 | 1 0 0 e 1 |
| 7 | 1 Ø 1 Ø Ø | 1 0 1 Y 1 | 4 | 1 0 1 e 1 |

## 3. A CUTPOINT CELLULAR ASSOCIATIVE MEMORY SYSTEM

### 3.1. INTRODUCTION

In this chapter, we propose an associative memory implemented by the cutpoint cellular logic. The cutpoint cellular logic which was first considered by Minnick[32] consists of the cutpoint cells with various output functions. These cells are especially compatible with the batch fabrication of integr ed circuits, and possess the advantages of high speed, low cost, small size and good reliability.

The proposed associative memory consisting of m n-bit words is also word-organized, and each bit-memory provides a pair of complementary outputs which satisfy the basic requirement for the ordered retrieval based on Lewin's scheme [11]. The selection of the memory words for reading or writing is performed by a simultaneous comparison and a tag operation. The tag of the matched words can be performed on the basis of either serial-by-word or parrallel-by-word. The serial-by-word tagging is accomplished by using a controlled shift register. The dematching of the tagged word is completely automatic. The memory sequential control system does not use any ladder structure which is commonly employed in previous associative memory systems, and hence the operating speed is largely increased. The readout of stored information is nondestructive. The stage delays for various operations are given, and may be utilized to estimate the speed of the associative memory. In the following sections, the logical structure and each operational procedure will be described separately. The derivations of several stage delays and the time interval for the shift or writing pulses can be found in Appendix II.

### 3.2. CUTPOINT CELLS

Before presenting our proposed cutpoint cellular associative memory, we first describe the cutpoint cells [32], which are the basic elements used in our memory. Each cutpoint cell is designated by an index k and has two inputs x and y and two outputs x and z as shown in Fig. 7. The relations between the output z and the index k are listed in Table 4. It is noted that when k=13, the cutpoint cell performs as a set-reset flip-flop. Thus, we may take another lead from the cell, which has the output $\bar{z}$, where $\bar{z}$ is the complement of z. It will be seen that this additional output terminal for a cutpoint cell of index 13 will help

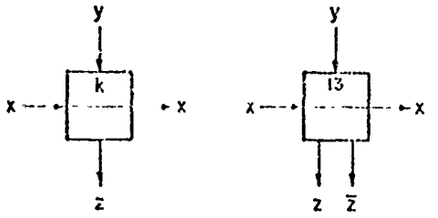



Fig. 7. Symbolic diagram of Minnick's cutpoint cell.

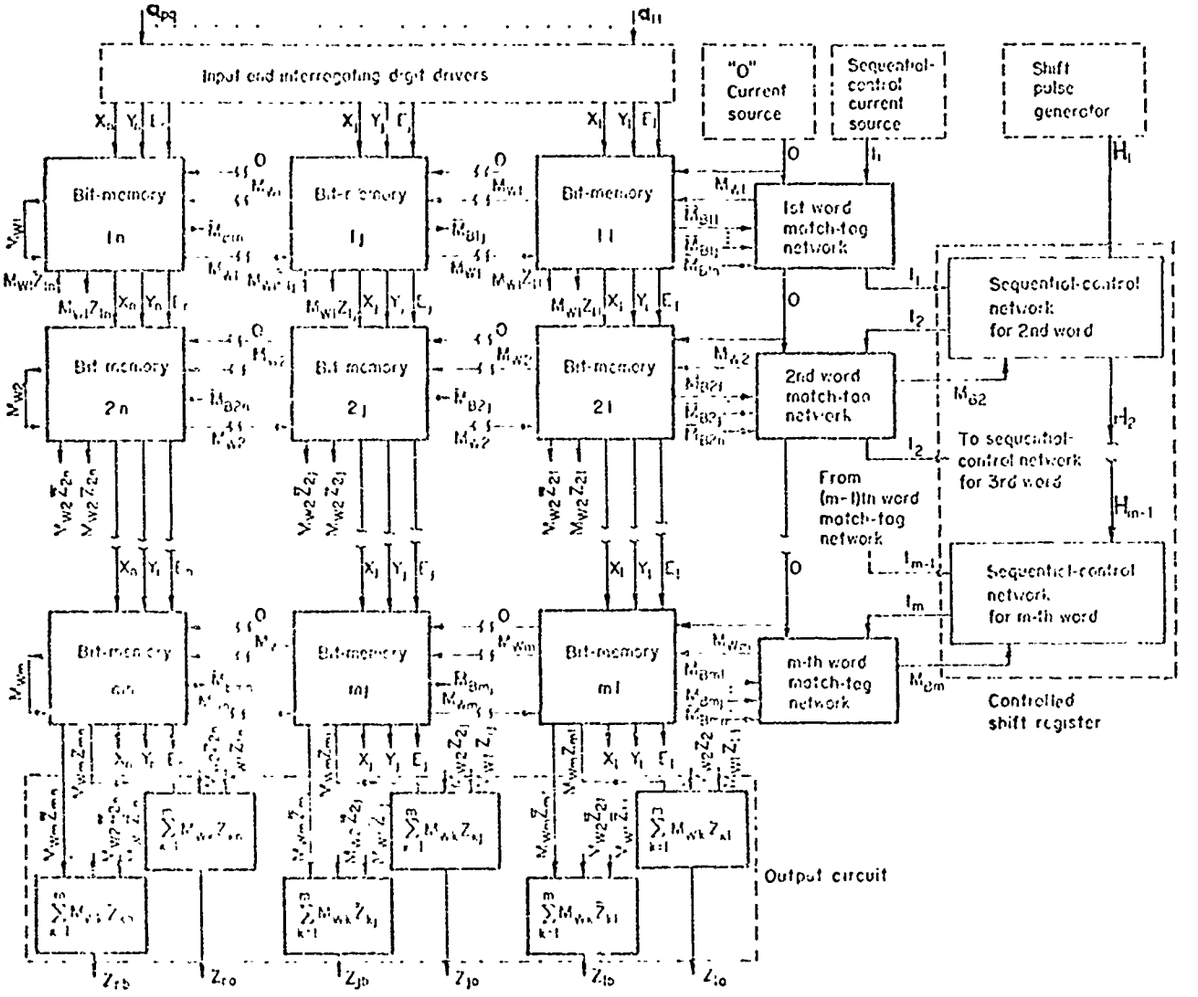



Fig. 8. Block diagram of a cutpoint cellular associative memory

Table 4.

Indices and Output Functions of Minnick's Cutpoint Cells [32]

| Index k | Output z | Output $\bar{z}$ | Operation | Stage delay $T_k$ |
|---|---|---|---|---|
| 0 | 1 | None | | $T_0$ |
| 1 | $\bar{y}$ | " | Complementer | $T_1$ |
| 2 | $\bar{x}+\bar{y}$ | " | NAND | $T_2$ |
| 3 | $\bar{x}\,\bar{y}$ | " | NOR | $T_3$ |
| 4 | $x+y$ | " | OR | $T_4$ |
| 5 | $x\,\bar{y}$ | " | Inhibitor | $T_5$ |
| 6 | $x \oplus y$ | " | Exclusive-or | $T_6$ |
| 7 | 0 | " | | $T_7$ |
| 13 | 1 (0) | 0 (1) | Set (x)-reset (y) flip-flop | $T_{13}$ |

simplify the logical organization of our memory. In Table 4, we also list the stage delays $T_k$ for the outputs of cutpoint cells with various indices. They will be used to estimate the operating speed of our associative memory.

### 3.3. THE LOGICAL STRUCTURE

The block diagram of the proposed cutpoint cellular associative memory is shown in Fig. 8. The bit-memories are organized in the usual manner. Each word, except the first one, of the memory contains n identical bit-memories, one match-tag network, and one sequential control network. The structure of the first word is the same as others except that it has no sequential control network. The m bit-memories of equal significance have a common pair of output networks.

#### 3.3a. Bit-Memories

As shown in Fig. 9, the bit-memory ij belonging to the j-th significant digit of the i-th word consists of nine cutpoint cells. The input, interrogating, and enabling-masking lines for all m bit-memories 1j through mj are the vertical lines $X_j$, $Y_j$ and $E_j$ respectively. The double-sense lines are labeled by their output functions $M_{Wi}Z_{ij}$ and $M_{Wi}\overline{Z}_{ij}$, where $M_{Wi}$ is the word-match variable for the i-th word and $Z_{ij}$ and $\overline{Z}_{ij}$ are the output variables for the writing flip-flop which is a cutpoint cell of index 13.

The circuit for writing contains mainly a set-reset flip-flop. The logic functions for setting $W_s$ and clearing $W_c$ the writing flip-flop are obviously the following:

$$W_s = M_{Wi}X_j, \tag{16}$$

$$W_c = M_{Wi}\overline{X}_j. \tag{17}$$

The circuit for comparison has the output function $\overline{M}_{Bij}$, which is the complement of the bit-match variable $M_{Bij}$. It follows from Fig. 9 and Table 4, that $\overline{M}_{Bij}$ is given by the equation

$$\overline{M}_{Bij} = \overline{E}_j(Y_j \oplus Z_{ij}), \tag{18}$$

where $\oplus$ is the exclusive-or operation. The m j-th bit-memories being enabled or masked depend upon whether the binary variable $E_j=0$ or $E_j=1$ respectively. The binary variable $Y_j$ is called the interrogating variable, which combined with $E_j$ is employed to determine the three possibilities "0",

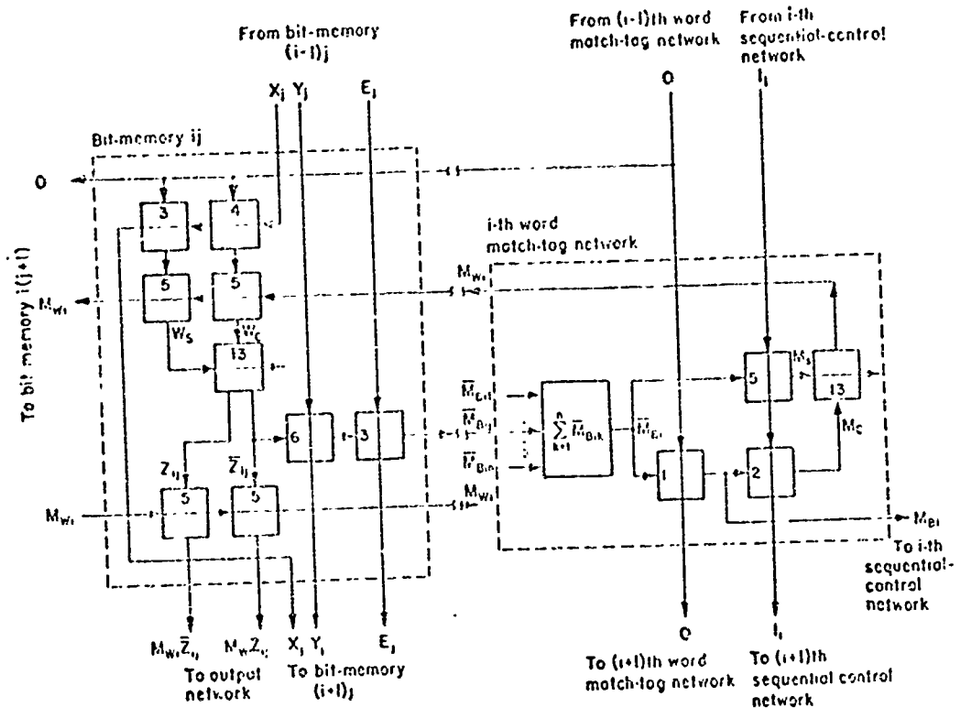



Fig. 9. Logical structure of a bit-memory and a match-tag network for the ith word.

"1" and "Ø" for an interrogating digit. This situation is shown in Table 5.

The circuit for reading contains two identical cutpoint cells of index 5. One of these cells has the output $M_{Wi}\overline{Z}_{ij}$, and the other has the output $M_{Wi}Z_{ij}$. When the i-th word is tagged, $M_{Wi}=1$ and consequently, these two outputs are complementary to each other. When the i-th word is not tagged, $M_{Wi}=0$ and then both outputs are zero.

3.3b. Match-Tag Networks

The match-tag network of the i-th word is shown in Fig. 9. It performs the following functions:

1) It is used as a matched-word-indicator, for word i is matched when I .

t is employed to set the flip-flop in the sequential control network of the first matched word i so that $I_i=1$. This is accomplished when $M_{Bi}=1$ under the conditions that $M_{i-1}=1$ and $I_{i-1}=1$.

3) The first matched word i is tagged when $M_{Bi}=1$ and $I_i=1$.

4) The tagged word i is automatically dematched after $M_{Bi}$ changes from 1 to 0.

It follows from Fig. 9 that $M_{Bi}$ is given by the equation

$$M_{Bi} = \prod_{j=1}^{n} [E_j+(Y_j \odot \overline{Z}_{ij})]. \tag{19}$$

The logic functions for setting $M_s$ and clearing $M_c$ the tag flip-flop are obviously the following:

$$M_s = I_i M_{Bi} \tag{20}$$

$$M_c = \overline{I}_i + \overline{M}_{Bi}. \tag{21}$$

The bit-match circuit of the i-th word is shown in Fig. 10. If the maximum allowable fan-in of each cutpoint cell is two, this network can be synthesized in a tree-type structure consisting of only the cutpoint cells of index 4 as shown in Fig. 10(b). The stage delay of this tree-type circuit implemented by n such cells is proportional to $\lceil \log_2 n \rceil$ which denotes the smallest integer being not smaller than $\log_2 n$.

3.3c. Output Networks

The m bit-memories of equal significance have a pair of common output networks possessing the following two summed outputs:

Table 5.

Interrogating Digits for Bit-Memories of Equal Significance

| $E_j$ | $Y_j$ | Interrogating digit | Remark |
|---|---|---|---|
| 0 | 0 | 0 | Enabled comparison |
| 0 | 1 | 1 | |
| 1 | 0 | Ø | Masked comparison |
| 1 | 1 | Ø | |

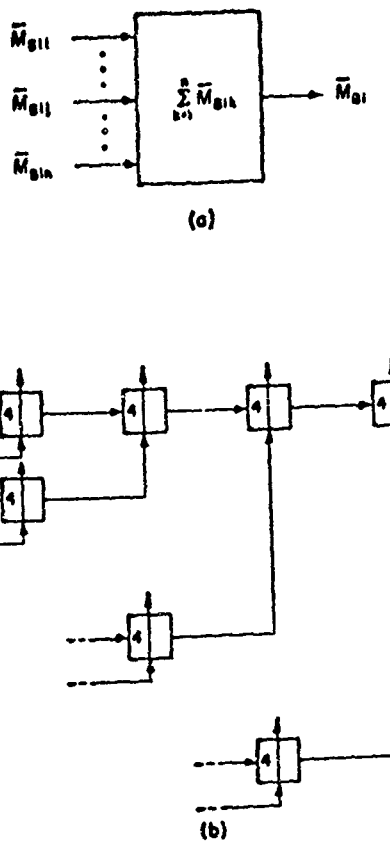



Fig. 10. (a) Block diagram and (b) tree-type structure of the bit-match circuit for the ith word.

$$Z_{ja} = \sum_{k=1}^{m} M_{Wk} Z_{kj}, \tag{22}$$

$$Z_{jb} = \sum_{k=1}^{m} M_{Wk} \bar{Z}_{kj}. \tag{23}$$

3.3d. Controlled Shift Register

The controlled shift register consists of m-1 sequential control networks for words 2 through m as shown in Fig. 8. The logical structure of each sequential control network is shown in Fig. 11. It is noted that there is a switch with two positions "W" and "R". The switch is normally turned to position "W", and only during the sensing or output-summing process, it is turned to position "R".

After the control current $I_1$ for the first word is on, the controlled shift register is used for turning on the control currents $I_2$ through $I_m$ at the appropriate moment in order to satisfy the requirements of the serial-by-word tagging of all matched words. Each matched word i is tagged when $I_i=1$. The controlled shift register is also used for turning off the control currents in order to dematch all the tagged words.

The controlled shift register is essentially a high speed shift register associated with other control circuits. As shown in Fig. 11, each sequential control network has three inputs $I_{i-1}, H_{i-1}, M_{Bi}$ and two outputs $I_i$ and $H_i$. The logic function for the output current $I_i$ is given by the following equations:

$$I_i = \bar{M}_{Bi} I_{i-1} + M_{Bi} I_i'. \tag{24}$$

The symbol $\bar{I}_i'$ corresponding to the output terminal $\bar{z}$ of a set-reset flip-flop is the output of the flip-flop (the cutpoint cell of index 13). This flip-flop is set or cleared by $I_s'$ or $I_c'$ which are given below:

$$I_s' = H_{i-1} I_{i-1} M_{Bi}, \tag{25}$$

$$I_c' = H_{i-1} \bar{I}_{i-1}. \tag{26}$$

Furthermore, the shift pulse $H_i$ has the following recursive property:

$$\begin{aligned} H_i &= H_{i-1} \bar{M}_{Bi} \\ &= H_{i-2} \bar{M}_{B(i-1)} \bar{M}_{Bi} \\ &= \cdots\cdots \\ &= H_1 \bar{M}_{B2} \bar{M}_{B3} \cdots \bar{M}_{B(i-1)} \bar{M}_{Bi}. \end{aligned} \tag{27}$$

This recursive property is essential for serial-by-word tag process.

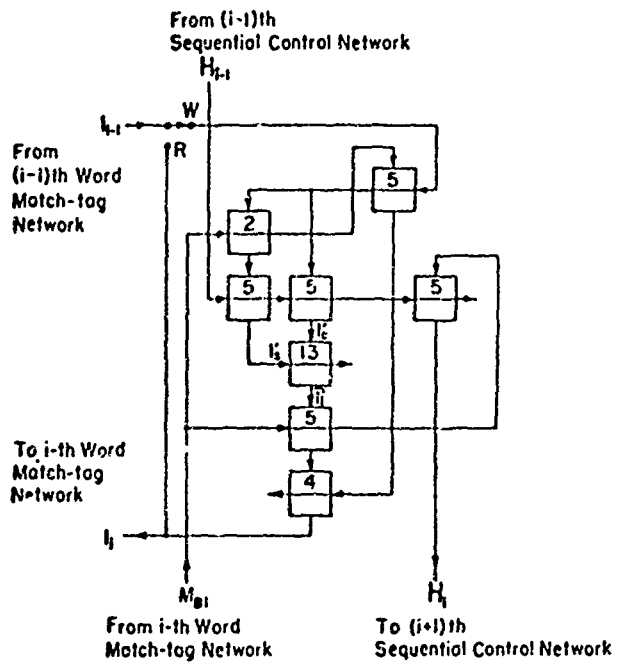



**Fig. 11.** The *i*th sequential control network of controlled shift register.

## 3.4. OPERATIONAL PROCEDURES

### 3.4a. Comparison and Tag Processes

The comparison is a simultaneous or parallel-by-word and parallel-by-bit process. A memory word which is matched or mismatched to an interrogating word is indicated by $M_{Bi}=1$ or $M_{Bi}=0$ respectively.

The comparison process starts with applying the interrogating word to the associative memory. The tag operation begins when the sequential control current $I_1$ is turned on. It follows from Figs. 9 and 10(b) and Table 4 that the stage delay $T_{yb}$ between the application of the interrogating word and the indication of the matched word ($M_{Bi}=1$) is given by the equation

$$T_{yb}=T_1+T_3+T_6+\left[\log_2 n\right]T_4. \tag{28}$$

During the writing process, we need the serial-by-word tagging, which can be accomplished due to the recursive property (27) of $H_i$. The stage delay between the shift pulses $H_1$ and $H_{i-1}$ is obviously $(i-2)T_5$. During the sensing process of ordered retrieval or the output-summing process for pattern recognition, we require the parallel-by-word tagging. This is accomplished by applying a common control current so that the switch is placed on position R in each sequential control network. It is obvious that there is no stage delay for this type of tagging. A memory word which is tagged or not tagged is indicated by $M_{Wi}=1$ or $M_{Wi}=0$ respectively. To terminate the comparison and tag processes, the interrogating digits and the current $I_1$ are first removed. Due to (24), each current $I_i$ is automatically turned off by $I_{i-1}$. The stage delay $T_{it}$ between the trailing edges of the current waves $I_1$ and $I_i$ is given by the equation

$$T_{it}=(i-1)(T_4+T_5). \tag{29}$$

### 3.4b. Writing Process

We first have to locate all the empty words before starting the writing process. An empty word in our memory is a word consisting of a string of 0's, and hence all empty words can be found by an enabled comparison with a string of n bits "0" as the interrogating word. If there are no empty words, we can select the occupied words, which will no longer be used, by an enabled-masked or a masked comparison.

The writing process may be illustrated by the procedure of writing

the first two words of r matched words. The relative time sequences of various waves and pulses are shown in Fig. 12. The simultaneous comparison is first performed as indicated by $E_j$ and $Y_j$. Through the stage delay $T_{yb}$, the matched words with the positions $k_1$, $k_1+k_2$, ... are indicated by $M_{Bk_1}=1$, $M_{B(k_1+k_2)}=1$, ... At this instant, the control current $I_1$ is turned on. Then, the first shift pulse $H_1^{(1)}$ is applied at the time $T_{ys1}$ after the beginning of $Y_j$. It is seen from the Appendix II that $T_{ys1}$ is given by the equation

$$T_{ys1}=T_1+T_2+T_3+(m-2+\lceil \log_2 n \rceil)\,T_4+T_6. \tag{30}$$

The control current $I_{k_1}$ is turned on by the shift pulse $H_{k_1-1}^{(1)}$, which is initiated by the first shift pulse $H_1^{(1)}$. The stage delay $T_{si1}$ between $H_1^{(1)}$ and the leading edge of $I_{k_1}$ is

$$T_{si1}=T_4+k_1T_5+T_{13}. \tag{31}$$

The $k_1$-th word is tagged by the control current $I_{k_1}$, that is, $M_{Wk_1}$ is initiated. It follows from Fig. 9 that the stage delay $T_{iw}$ between the leading edges of $I_{k_1}$ and $M_{Wk_1}$ is found to be

$$T_{iw}=T_{2,5}+T_{13}, \tag{32}$$

where $T_{2,5}=\mathrm{Max}\,(T_2,T_5)$. Then, the first word to be written in is applied to the associative memory through the $X_j$, $j=1,2,\ldots,n$, lines. Let this word be represented by $(X_n^{(1)}\ X_{n-1}^{(1)}\ \ldots\ X_1^{(1)})$. The stage delay $T_{yx}$ from the beginning of the comparison process to the pulses $X_j^{(1)}$, $j=1,2,\ldots,n$ is given by the equation (see Appendix II)

$$\begin{aligned} T_{yx}&=T_{ys1}+T_{si1}+T_{iw}+T_w^{(1)}\\ &=T_1+T_2+T_3+(m-1+\lceil \log_2 n \rceil)\,T_4+mT_5+T_6+2T_{13}+T_{2,5}. \end{aligned} \tag{33}$$

After the first word has been written in the $k_1$-th memory word, $M_{Bk_1}$ changes from 1 to 0 in order to dematch word $k_1$. The stage delay $T_{xb}$ between the pulses $X_j^{(1)}$, $j=1,2,\ldots,n$, and the trailing edge of $M_{Bk_1}$ can be found from Fig. 9. It is given by the equation

$$T_{xb}=T_1+T_3+T_5+T_6+T_{13}+\lceil \log_2 n \rceil\,T_4+T_{3,4}, \tag{34}$$

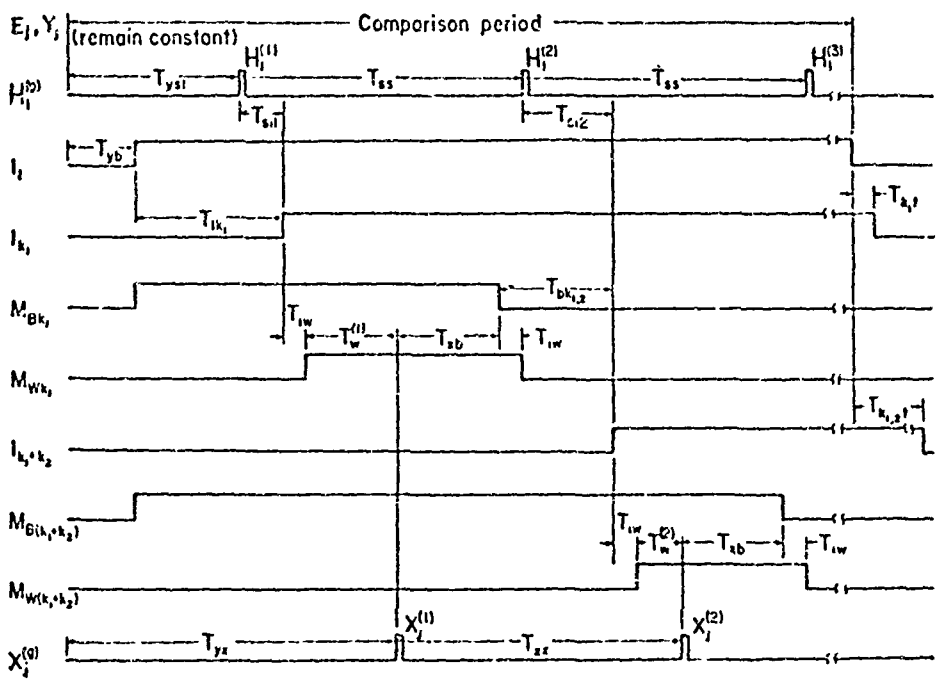



Fig. 12. Illustration of the time sequences for sequential writing process.

where $T_{3,4} = \text{Max}\ (T_3, T_4)$. Then, the tagged word is automatically dematched when $M_{Wk_1} = 0$.

Similarly, the second matched word at the position $k_1 + k_2$ is tagged by the current $I_{k_1+k_2}$, which is turned on by the shift pulse $H^{(2)}_{k_1+k_2-1}$ initiated by the second shift pulse $H_1^{(2)}$. Then, the second word $(X_n^{(2)}\ X_{n-1}^{(2)}\ \ldots\ldots\ X_1^{(2)})$ to be written in the tagged word at $k_1 + k_2$ of the memory can be applied at an appropriate time. The procedure is the same as that for writing the first word in the first tagged word.

After all the words are sequentially written in the selected memory words, the comparison process is terminated by removing the interrogating digits from the lines $Y_j$ and $E_j$, $j=1,2,\ldots,n$. At the same instant, the control current $I_1$ vanishes, and the writing process terminates. Then, the next shift pulse $H_1^{(r+1)}$ clears all previously set flip-flops in the controlled shift register. It is shown in the Appendix II that the time interval $T_{ss}$ or $T_{xx}$ between any two consecutive shift pulses $H_1^{(g)}$ and $H_1^{(g+1)}$ or any two consecutive pulses $X_j^{(g)}$ and $X_j^{(g+1)}$ should be

$$T_{ss} = T_{xx}$$
$$= T_1 + T_3 + (1 + \lceil \log_2 n \rceil)\ T_4 + (m+3)\ T_5 + T_6 + 3T_{13} + T_{2,5} + T_{3,4}. \qquad (35)$$

3.4c. Reading Process

The reading process is performed on the unique-match and non-destructive basis. An enabled-masked comparison is performed to select the memory word to be readout. Then, the unique-matched word is tagged by the method as described in the writing process. Suppose that the unique matched word is at the position $k_1$. Then after the delay $T_{iw} + T_5$ since the leading edge of $I_{k_1}$, the outputs of this tagged word are fed to the output circuit. The readout of this word is terminated after the interrogating word is removed and the control current $I_1$ is turned off. A second shift pulse $H_1^{(2)}$ is used to clear the flip-flop in the sequential control network of the $k_1$-th word.

During each interrogating cycle of ordered retrieval, we first use the comparison process to select the memory words, and then use the parallel-by-word tag process for sensing the contents of all tagged bit-memories of equal significance. The results of sensing as designated in Table 6 are employed to initiate the interrogating digit drivers so that an appro-

Table 6.

Sensing Results of Cutpoint Cellular Associative Memory

| Type of sense | Outputs at j-th bit-posision | | Contents of j-th bit memories in tagged words | Type of sense for two tagged words |
|---|---|---|---|---|
| | $Z_{ja}$ | $Z_{jb}$ | | |
| 0 | 0 | 1 | 0 | $G_0$ |
| 1 | 1 | 0 | 1 | $G_1$ |
| X | 1 | 1 | 0 and 1 | P |
| Y | 0 | 0 | No words are matched | Y |

priate interrogating word for subsequent comparison is generated. Each of the matched words is readout when a unique-tag appears. Because the sensing operation is accomplished by the parallel-by-word tagging, we need only turn the switch in each sequential control network to position "R".

# 4. PATTERN RECOGNITION BY USING AN ASSOCIATIVE MEMORY

## 4.1. INTRODUCTION

During the past three years, an associative memory has been employed in the design of high-speed pattern recognition machines [20,23,25,30]. Estrin and Fuller [20] have developed a group of commands executable in their associative memory for pattern recognition based on the extracted pattern features originally proposed by Unger [35]. Because the programming for feature extraction and pattern classification is always sequential, the speed for such a pattern recognition scheme is limited by the execution time of the programming. Simmons [23] has performed an experiment by using an associative memory as a pattern recognizer. In his experiment, the patterns, which are simulated by a random number generator and are recognized by a logical test, are simple four-element mosaic configurations. McAteer, Capobianco and Koppel [25] have considered the case, where an input pattern is compared with the stored words representing the tolerance envelopes for the desired patterns. Each input pattern is required to be normalized, sampled and quantized at set intervals, and then is classified by performing a "between limits" search in the associative memory. Fuller and Bird [30] have applied their sequential-state-transformation method for arithmetic algorithms in evaluating their threshold logic functions for their adaptive pattern recognition scheme.

The purpose of this paper is to present a pattern-recognition scheme by using any general-purpose associative memory, such as one of the systems described in chapters 2 and 3. The input patterns for recognition may have wide variations, provided that the distinct features of individual pattern classes can be extracted. These extracted features are used to determine the prestored templates for all possible pattern classes. The proposed pattern recognition scheme possesses the following advantages:

1). The operational procedure of the scheme is simple because the recognition of input patterns requires at most two processing cycles.

2). The implementation is also simple since any general-purpose associative memory can readily be utilized without increasing complexity in hardware.

3). The operating speed for recognition is high because of the simple recognition procedure and the parallel operation of associative memory.

4). Deviated patterns of the same class can be recognized by one

prestored template for that pattern class.

## 4.2. REPRESENTATION OF INPUT PATTERNS

To transform a pattern into the form of a memory or an initial interrogating word, we project this pattern on a screen which is divided in the form of a matrix of order $p \times q$. Then, the pattern can be represented by a binary $p \times q$-tuple

$$(a_{11}, a_{12}, \ldots, a_{1q}, a_{21}, \ldots, a_{ij}, \ldots, a_{pq}), \tag{36}$$

where each $a_{ij}$, $i=1,2, \ldots, p$, $j=1,2, \ldots, q$, is a binary variable being either "0" or "1" based on whether the cell at the ij entry of the matrix (i-th row and j-th column) is covered or not covered by the pattern respectively. We shall call $a_{ij}$ a <u>black</u> cell of a pattern if it is covered by the pattern, and a <u>white</u> cell of a pattern if it is not covered by the pattern. In other words, when we project a pattern on the screen, $a_{ij}$ is a black cell of the pattern if $a_{ij}=0$ and a white cell of the pattern if $a_{ij}=1$. Thus, we convert a pattern to a memory word of length $p \times q$.

When we have a pattern to be recognized, $a_{11}$ through $a_{pq}$ of the $p \times q$-tuple of the pattern are used to initiate the interrogating digit drivers of an associate memory to generate an interrogating word for the initial comparison process. When $a_{ij}=0$, the corresponding interrogating digit is also 0. When $a_{ij}=1$, the corresponding interrogating digit is $\emptyset$ which denotes the "don't-care" digit.

Before the $p \times q$-tuple of an input pattern is used for recognition, the position of the input pattern is adjusted so that at least a black cell appears in the p-th (bottom) row and at least a black cell appears in the first (left) column. This position is called the <u>standard position</u>. We can see that this condition is easily satisfied for each input pattern.

## 4.3. DETERMINATION OF TEMPLATES OF PATTERN CLASSES

The template including all possible acceptable deviated patterns in a class $h_i$, which is stored in the associative memory as a memory word can be determined as follows: We first find the intersection [36] of the set of the white cells of all possibly acceptable deviated patterns of Class $h_i$. This intersection, denoted by $F_{h_i}$, is a set of white cells representing the features of the patterns of Class $h_i$ and is invariant for all possibly acceptable patterns of Class $h_i$. In the memory-word

for Cla. $h_i$, the information stored in the corresponding bit-memories for these white cells is all "1", and the corresponding bit-memories for the remaining cells is all "0". Thus, the template of Class $h_i$ which is constituted by $F_{h_i}$ and its complement, denoted by $B_{h_i}$, is stored in a memory-word consisting of 1's and 0's.

It is noted that the set $B_{h_i}$ is actually the union [36] of the sets of the black cells of all possibly deviated patterns of Class $h_i$. When the degree of deviation of the patterns in Class $h_i$ increases, the set $F_{h_i}$ becomes smaller and $B_{h_i}$ appears larger. Therefore, the set $F_{h_i}$ not only represents the features of the patterns in Class $h_i$, but also determines the degree of allowable deviation of the patterns in Class $h_i$. In the associative memory, the bits "1" corresponding to the cells in $F_{h_i}$ represent the identifier of the stored word corresponding to the template of Class $h_i$.

The templates of other pattern classes can be determined by the same method. Because each pattern class is different from all other pattern classes, the sets $F_{h_1}$, $F_{h_2}$, ... , $F_{h_h}$ are all distinct. However, in order to allow a large degree of deviations of the acceptable patterns in each pattern class and to avoid misclassification of patterns, the sets $F_{h_1}$, $F_{h_2}$, ... , $F_{h_h}$ have to be adjusted with respect to each other. To illustrate this situation, the templates of all the capital English letters and the numerals may be proposed as shown in Fig. 13, in which each template is represented by an 8 X 8 matrix. The shaded and non-shaded regions of each template $h_i$ represent $F_{h_i}$ and $B_{h_i}$ respectively. Each of these templates is stored in a memory word, in which the extremely right bit-memory and the extremely left bit-memory store the information corresponding to the template cells $a_{11}$ and $a_{pq}$ respectively. In order to facilitate the explanation, the determination of the templates will be discussed in further detail after the recognition scheme is presented in the next section.

## 4.4. RECOGNITION SCHEME

Suppose that the template of each class has been stored in the associative memory as described before. The procedure for our recognition scheme is given below:

1). After the input pattern is placed in the standard position, a string of binary signals of the p X q-tuple of the input pattern is applied

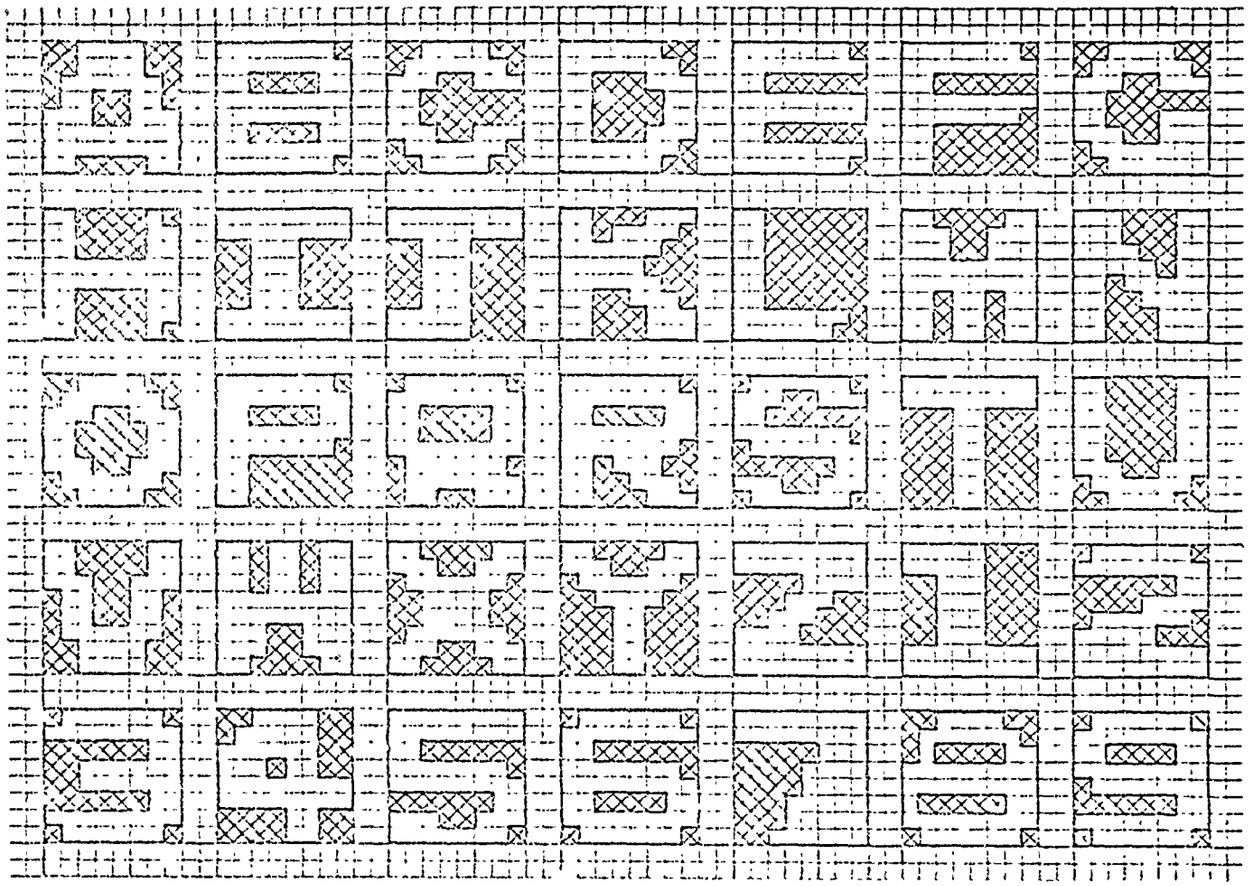


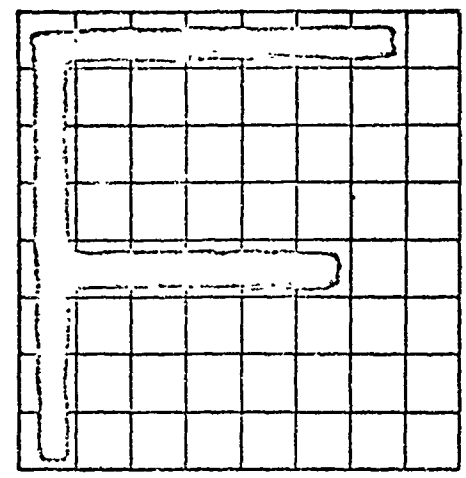
Fig. 13. Pre-stored templates for recognizing hand-written block capital English characters and numerals.

to the interrogating digit drivers. These drivers are initiated by the input signals and generate the interrogating word for the initial comparison. As described before, the output of the driver is "0" or "Ø" when its input signal is "0" or "1" respectively. Therefore, the initial cycle of recognition involves an enabled-masked comparison unless all the cells of the input pattern are black. If no matching is indicated, then the input pattern belongs to the rejection class, i.e., the input pattern does not belong to any of the pattern classes we want to classify. If a unique matching is indicated, the input pattern belongs to the class of the matched template. In both cases, the recognition process for this input pattern is completed. However, if a multiple matching occurs, follow the next step.

2). Whenever a multiple matching appears, an output-summing operation for all matched and also tagged memory words follows. The summed output signals of equal significant bit-columns of the associative memory are used to activate the interrogating digit drivers to produce another interrogating word for second comparison. The new interrogating digit is "0" or "1" when the corresponding summed output signal is "0" or "1" respectively. After the new interrogating word is generated, the second enabled comparison is performed. If a unique match occurs at this time, the matched template indicates the class of the input pattern. Otherwise, the input pattern belongs to the rejection class. The recognition process is then completed.

It is noted that multiple matching never occurs in the second cycle because the comparison is an enabled comparison process whose interrogating word contains no "don't-case" bits, and because the stored templates of the pattern classes are all distinct. Hence, the recognition process for each input pattern requires no more than two cycles.

To demonstrate this technique, let us consider the case of recognizing hand-written block capital English letters and the numerals. The templates for these 35 pattern classes are shown in Fig. 13 and the information of these 35 templates are assumed to be stored in the associative memory as described in the last section. Suppose that the pattern shown in Fig. 14(a) is the input pattern to be recognized. Then, we obtain the initial interrogating digits according to this input pattern as shown in Fig. 14(b). Since the bit-columns of the associative memory



(a)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | $\phi$ |
| 0 | $\phi$ | $\phi$ | $\phi$ | $\phi$ | $\phi$ | $\phi$ | $\phi$ |
| 0 | $\phi$ | $\phi$ | $\phi$ | $\phi$ | $\phi$ | $\phi$ | $\phi$ |
| 0 | $\phi$ | $\phi$ | $\phi$ | $\phi$ | $\phi$ | $\phi$ | $\phi$ |
| 0 | 0 | 0 | 0 | 0 | 0 | $\phi$ | $\phi$ |
| 0 | $\phi$ | $\phi$ | $\phi$ | $\phi$ | $\phi$ | $\phi$ | $\phi$ |
| 0 | $\phi$ | $\phi$ | $\phi$ | $\phi$ | $\phi$ | $\phi$ | $\phi$ |
| 0 | $\phi$ | $\phi$ | $\phi$ | $\phi$ | $\phi$ | $\phi$ | $\phi$ |

(b)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |

(c)

Fig. 14. (a) An input pattern to be recognized, (b) the initial interrogating digits for the input pattern shown in (a), and (c) the summed outputs of the tagged templates (for Classes B, E, F, P and R) for the input pattern shown in (a).

which are interrogated by the "don't-care" digits "Ø" are all masked out, the stored templates of Classes B, E, F, P and R are all matched. All these matched memory words are tagged. The contents of the tagged bit-memories of the equal significant bit-position are summed together by an output network. The summed outputs which are the Boolean sums obtained by these output networks are shown in Fig. 14 (c). According to our procedure, we need the second cycle for recognizing the input pattern. The new interrogating digits for the comparison in the second cycle are the same as shown in Fig. 14(c). Because the distribution of the 0's and 1's of the new interrogating word is identical to that of the pre-stored template for Class F shown in Fig. 13, the input pattern shown in Fig.14(a) is therefore recognized as F due to the unique match in the second comparison process.

From this example, it is easily seen that an input pattern is recognized as a pattern of Class $h_i$ due to a unique match during the second comparison process if and only if the set $F_{h_i}$ of the uniquely matched template of Class $h_i$ contains the set of white cells of each multiple matched template, as a proper subset during the initial comparison process. From the templates of Classes B,E, F, P and R shown in Fig. 13, we can see that

$$F_F \supset F_E \supset F_B \quad , \tag{37}$$

and

$$F_F \supset F_P \supset F_R \quad . \tag{38}$$

If we change the cell $a_{5,8}$ of the template of Class F in Fig. 13 from white to black as shown in Fig. 15, then the relation (38) no longer holds because the set $F_F$ in the alternate template does not contain $F_P$ in Fig. 13 as a proper subset. Should this alternate template be used for Class F, there would be no matching at all during the second comparison.

Suppose that we have the input pattern shown in Fig. 16(a). Then, the set of black cells of this input pattern is contained in the intersection $B_B \cap B_E \cap B_F \cap B_H \cap B_M \cap B_P \cap B_R \cap B_W$. The reader may verify that this input pattern will be classified in the rejection class according to our technique unless an additional template shown in Fig. 16(b) is stored in the memory.

## 4.5. FURTHER CONSIDERATION IN THE DETERMINATION OF TEMPLATES OF PATTERN CLASSES

After the template of each pattern class is taken according to the intersection of the sets of white cells of all possibly acceptable deviated

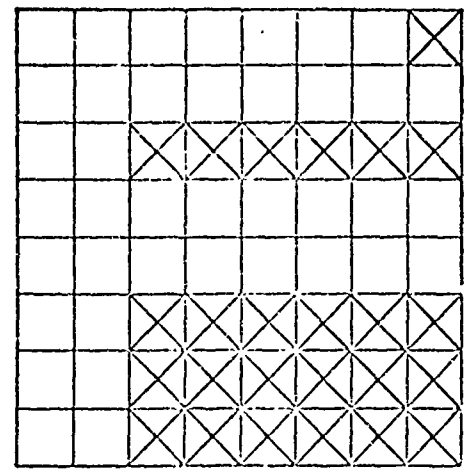


Fig. 15. An alternate template of Class F.

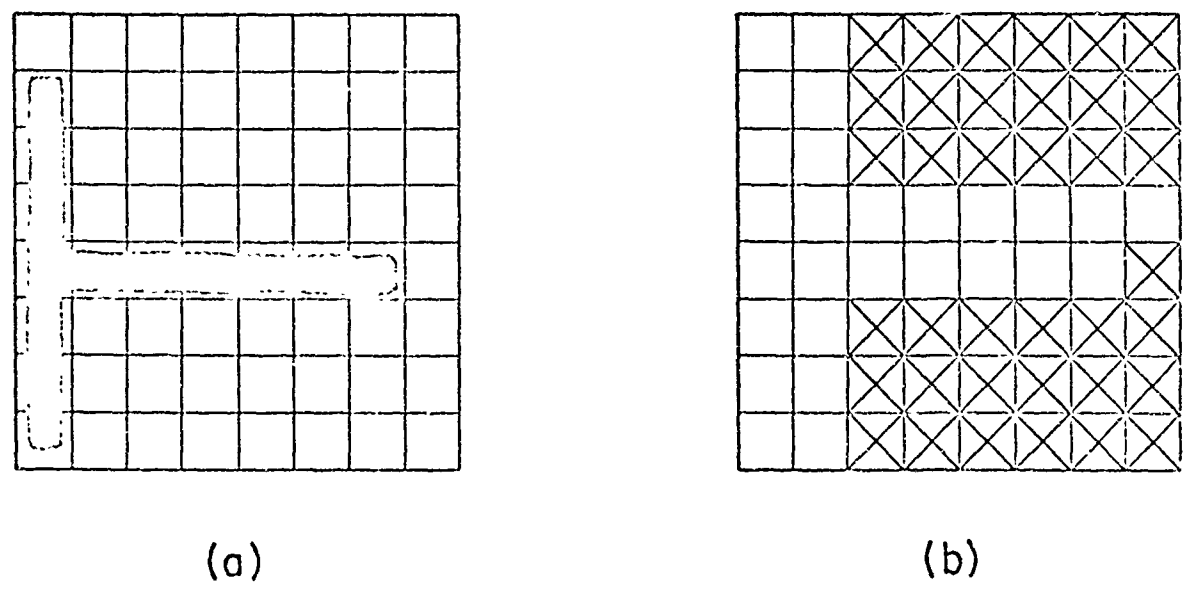



Fig. 16. The template (b) for recognizing the input pattern (a).

patterns of the class, an adjustment is required among the templates of several pattern classes if the intersection of the sets of black cells of these templates contains the set of black cells of an acceptable pattern in one of these classes. For instance, the intersection of $B_B$, $B_E$, $B_F$, $B_P$ and $B_R$ of the unadjusted templates for Classes B, E, F, P and R may contain an acceptable pattern of Class F. In order to identify this acceptable pattern of Class F, we have to reduce the set $B_F$ until $B_F$ is equal to the intersection. That is,

$$B_F = B_B \cap B_E \cap B_F \cap B_P \cap B_R. \tag{39}$$

Of course, this reduction may reject some other acceptable patterns of Class F, and hence we have to compromise these considerations in the design of the templates. For instance, if we want to classify the pattern shown in Fig. 17(a) in Class F, the $B_F$ in the template of Class F has to be reduced until

$$B_F = B_B \cap B_E \cap B_F \cap B_P \cap B_R \cap B_6 \tag{40}$$

because the set of black cells of the pattern shown in Fig. 17(a) is contained in that of the template for Class 6. The reduced template for Class F is shown in Fig. 17(b). However, this reduced template should not be used for Class F because many patterns of Class F, such as the one shown in Fig.14(a) will be rejected. Thus, we have to decide whether we want to accept the pattern in Fig. 17(a) in Class F at the expense of rejecting many other acceptable Class F patterns. It is noted that if Class 6 were not one of the pattern classes we want to classify, then the patterns shown in both Fig. 14(a) and 17(a) would be accepted as Class F patterns by using the templates shown in Fig. 13.

The relations among the sets of black cells in the templates of various pattern classes can also be expressed in terms of their sets of white cells. In order to classify all the 26 capital English letters and the 10 numerals with large allowable deviations, some relations among the sets of white cells in the templates of these 35 pattern classes are required. These relations are listed in Table 7. It is seen that the templates shown in Fig. 13 satisfy all these relations.

## 4.6. IMPLEMENTATION

The requirements for an associative memory to perform our recognition scheme are listed below:

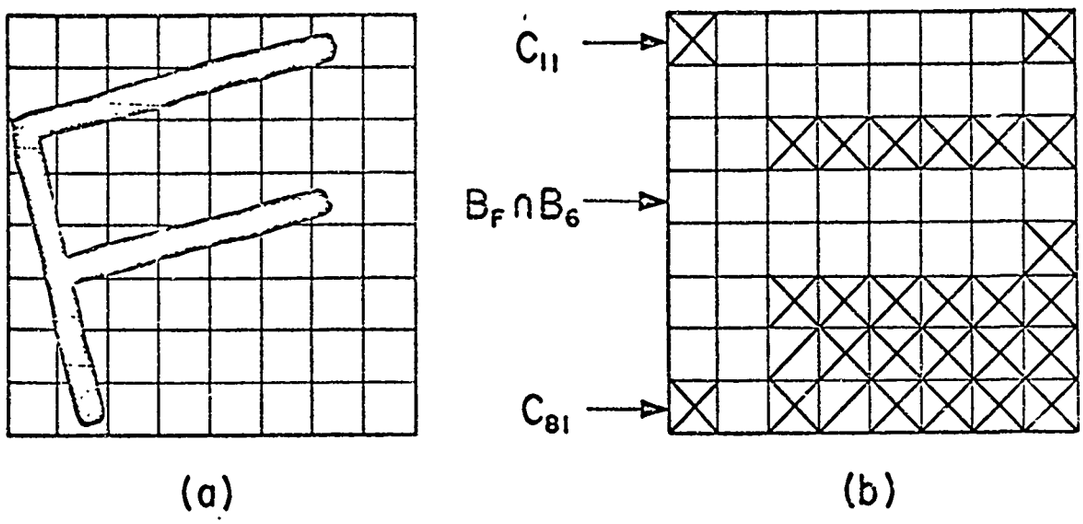



Fig. 17. (a) A deviated Class F pattern and (b) $B_F \cap B_6$.

Table 7.

Relations Among the Sets of White Cells in the Templates of Fig. 13

| $W_{k_i}$ | Subsets of $W_{k_i}$ |
|---|---|
| $W_C$ | $W_G$, $W_0$ |
| $W_E$ | $W_B$ |
| $W_F$ | $W_E$, $W_P$ |
| $W_H$ | $W_B$, $W_M$, $W_W$ |
| $W_J$ | $W_I$ |
| $W_L$ | $W_D$, $W_E$ |
| $W_P$ | $W_B$, $W_R$ |
| $W_S$ | $W_B$, $W_5$, $W_6$, $W_9$ |
| $W_T$ | $W_J$ |
| $W_1$ | $W_I$ |
| $W_3$ | $W_B$, $W_9$ |
| $W_6$ | $W_B$ |
| $W_8$ | $W_B$ |

1). It is non-destructive read-out.

2). It is word-organized and can perform the masked comparison and the output-summing operation for all tagged bit-memories.

It is obvious that these two requirements do not impose any difficulty on ordinary associative memory systems. Without any modification, the cryogenic associative memory or the cutpoint cellular associative memory is well suited for this purpose.

To see how the cutpoint cellular associative memory perform this recognition scheme, let us refer to Fig. 8. All bit-memories 1j through mj of the same j-th significant bit position can be enabled or masked out by setting the value of $E_j$ being 0 or 1 respectively. The interrogating digit for each column j is designated by the binary values of $E_j$ and $Y_j$ as shown in Table 5. Let n = pq. Then, $X_1$ through $X_n$ are the input variables for writing the components $a_{11}$ through $a_{pq}$ of each template in a tagged memory word. The summed output $Z_{ja}$ of the j-th column is derived by the right j-th output network as shown in Fig. 8, and is expressed by (22).

4.7. FURTHER CONSIDERATION OF RECOGNITION SCHEME

In each pattern class determined by the pre-stored template, many deviated patterns can be recognized. For instance, each of the input patterns as drawn by smaller scale and shown in Fig. 18 is recognized as F in two cycles. The deviation involves size, style, and orientation within certain limits, provided that the set of black cells of an input pattern is contained in $B_F$ of the pre-stored template for Class F. The degree of deviation, and possible mis-classification between B and 8 or S and 5 depend on the relative white components of the pre-determined templates stored in the memory. The following possible methods may be used for optimizing the degree of deviations, minimizing the number of mis-classification between similar characters of different classes:

1). The number of cells for each pattern can be increased to 256 or even 1024 which corresponds to p = q = 16 or 32 respectively. An associative memory possessing 1024 bits per word is possible [29]. However, when the number n of bit-memories per word is increased, the stage delay during comparison increases because it is proportional to $\lceil \log_2 n \rceil$ by using the cutpoint cellular associative memory. When p and q are increased, each curved boundary between the sets $F_{h_i}$ and $B_{h_i}$ of the template for

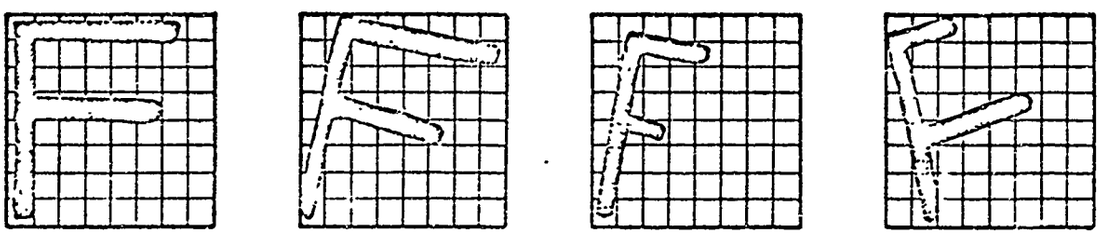


Fig. 18. Examples of deviated Class F patterns, which are recognizable by using the templates shown in Fig. 13.

Class $h_i$ can be made smoother and the determination and adjustment of the set $F_{h_i}$ become easier because of the increased flexibility due to a larger set $F_{h_i}$. Furthermore, the deviations of the pattern classes can also be increased by increasing the number of cells in the matrix.

2). The template of a certain class may be supplemented by another slightly different template, which possesses a larger set of white cells. For example, the template for Class F of Fig. 13 can be supplemented by that of Fig. 17(b) to increase the acceptable deviation of Class F.

## 5. TWO-SUMMAND ADDITION BY AN ASSOCIATIVE PROCESSING SYSTEM

### 5.1. INTRODUCTION

The early work concerning the applications of an associative memory was nearly confined to the area of information storage and retrieval. Since 1963, arithmetic algorithms processed in an associative memory have been considered by Estrin and Fuller [14], and many others [16,26,27,30]. Their algorithms are all based on software approach and most of them are processed in serial-by-bit manner. Although, Crane and Githens [27] have suggested the bulk processing technique being capable of performing arithmetic operations on the parallel-by-word and parallel-by-bit basis, their content-addressible memory derived from the distributed logic machine [18] needs to be bit-organized and their scheme may require large storage capacity for bulk processing.

The purpose of this chapter is to present a nonbulk two-summand addition method by utilizing the sensing process of a word-organized associative memory. Parallel arithmetic operations can be achieved by paying a limited price for the additional implementation besides the associative memory. However, the distributed logic circuit of a word-organized associative memory may be simpler than that of a bit-organized content-addressible memory, for example, each bit-memory of the former system does not require communicating with its two nearest neighbors.

In this chapter, we shall define a state symbol for each possible summand digit-pair so that the state form of each particular summand-pair can be developed. This concept will then be extended to establish sets of algebraic equations which characterize exactly all possible allowable addition processes (without overflow) and all possible unallowable addition processes (with overflow) for n-bit signed numbers. These numbers are represented by the 2's-complement notation. Thus a Mealy type sequential machine can be synthesized for transforming the state representation of each summand-pair to its corresponding sum. This state representation which constitutes the input of the sequential machine is readily obtained by a sensing process of an associative memory.

### 5.2. STATE REPRESENTATION

#### 5.2a. State Representation of Summand Digit-Pair

When two n-bit signed numbers are added, each summand digit-pair $(A_j, B_j)$ may possess one of the following four possible values:

(0, 0), (1, 1), (1, 0) and (0, 1),

which can be represented respectively by the symbols $G_0$, $G_1$, $P_0$ and $P_1$ where the subscripts 0 and 1 denote the binary values of the addend digit $B_j$. This type of representation is called the <u>state representation of the summand digit-pair</u>.

5.2b. <u>State Representation of Summand-Pair</u>

By using the state symbols $G_0$, $G_1$, $P_0$ and $P_1$, the two n-bit signed numbers being added can be expressed uniquely by n such symbols. The extremely left position of the state representation of two summands is determined by the sign digit-pair $(A_n, B_n)$ in which the ordinary convention 0 or 1 is used for denoting the positive or negative sign respectively. The state representation of the remaining n-1 magnitude bit positions of the given summand-pair is based on the 2's-complement number system.

5.2c. <u>Classes of Summand</u>

To simplify description, the signed summands are divided into three classes which are distinguished by the <u>class number</u> u, u=1, 2 and 3. We define Class 1 or 2 summands as both positive or both negative binary numbers respectively. Class 3 summands denote the two binary numbers with opposite signs.

5.2d. <u>Allowable and Unallowable Addition Processes</u>

When we consider the addition of two n-bit signed numbers, there are $4^n$ different cases because each digit position may have four possible state forms. For Class 1 or 2 summands, some of the $4^{n-1}$ possible addition processes should be excluded because of overflow. Overflow occurs whenever the absolute value of the sum exceeds $2^{n-1}$ for n-bit summands. The <u>allowable or unallowable addition processes</u> are defined as those additions without or with overflow respectively.

5.2e. <u>State Representations of Allowable Addition Processes</u>

The state form for the sign digit-pair $(A_n, B_n)$ of each addition process must be one of the four symbols $G_0$, $G_1$, $P_0$ and $P_1$. The state representation of the n-1 digit-pairs $(A_1, B_1)$ through $(A_{n-1}, B_{n-1})$ corresponding to the magnitudes of the two summands must be one of the permutation of n-1 objects selected from four different sets of elements, each of which consists of n-1 alike state symbols being $G_0$, $G_1$, $P_0$ and $P_1$. In order to show the state representation of all possible allowable addition processes, we define the following two operations among the state symbols $G_0$, $G_1$, $P_0$

and $P_1$. "+" is the same as <u>addition</u>, which is associative and commutative. "·" means <u>concatenation</u> which is associative but not commutative. The notation "·" for concatenation is usually omitted in the state representation of addition processes. For example, $G_0 \cdot G_1$ is usually written as $G_0G_1$. Furthermore, the concatenation of i identical state forms is usually writen as the i-th power of that state form. These two operations satisfy the following distributive law:

$$G_0(G_1+P_0) = G_0G_1+G_0P_0 \tag{41}$$

Suppose that the number 10.....0 with n-1 consecutive zeros is also interpreted as the negative number $-2^{n-1}$. The state representation $S_1(n)$ of all allowable addition processes for Class 1 n-bit summands may be written as

$$S_1(n) = G_0(P_0+P_1)^{n-2}\ (P_0+P_1+G_0)+ \sum_{i=0}^{n-3} G_0(P_0+P_1)^i\ G_0(P_0+P_1+G_0+G_1)^{n-i-2} \tag{42}$$

where the zero-th power of an expression is always 1 and the negative power of an expression is always 0. The symbols "1" and "0" act as identity and nullity elements, i.e., $G_0 \cdot 1 = 1 \cdot G_0 = G_0$, $G_0 \cdot 0 = 0 \cdot G_0 = 0$, and $G_0+0=G_0$. Thus, it is easily seen that (42) is a collection of $2^{n-2}+2^{2n-3}$ concatenations each having exactly n state symbols.

When n=2, (42) reduces to the following form:

$$S_1(2) = G_0(P_0+P_1+G_0). \tag{43}$$

This equation denotes that the three allowable addition processes for 2-bit positive summands are characterized by the three state representations $G_0P_0$, $G_0P_1$ and $G_0^2$.

For the 2's-complement number system, the state representations $S_1(n)$ through $S_3(n)$ of all possible allowable addition processes are listed in Table 8. Since the 2's-complement notation possesses the same representation for positive and negative zeros and also the end-around carries can be simply ignored, we shall consider only this number system in this chapter.

5.2f. <u>State Representation of Unallowable Addition Processes</u>

For Classes 1 and 2 summands, an addition process characterized by one of the following state representations

Table 8.

State Representations of Allowable Addition Processes

| Classes of summands $u$ | State representations $S_u(n)$ |
|---|---|
| 1 | $S_1(n)= G_0(P_0+P_1)^{n-2}(P_0+P_1+G_0) + \sum_{i=0}^{n-3} G_0(P_0+P_1)^i G_0(P_0+P_1+G_0+G_1)^{n-i-2}$ |
| 2 | $S_2(n)= G_0^n+G_1(P_0+P_1)^{n-2}G_1 + P_0(P_0+G_0)^{n-1}+P_1(P_1+G_0)^{n-1}+ \sum_{i=0}^{n-3} G_1(P_0+P_1)^i G_1(P_0+P_1+G_0+G_1)^{n-i-2}$ |
| 3 | $S_3(n)= G_0(P_0+G_0)^{n-1}+G_0(P_1+G_0)^{n-1}+P_1(P_0+P_1+G_0+G_1)^{n-1}+ P_0(P_0+P_1+G_0+G_1)^{n-1}$ |

1). $G_0(P_0+P_1)^i G_1 \ldots, \quad i \geqq 0$

2). $G_i(P_0+P_1)^{n-1}$,

3). $G_1(P_0+P_1)^i G_0 \ldots, \quad i \geqq 0$

is unallowable because of overflow. The state representations $U_1(n)$ and $U_2(n)$ of all possible unallowable addition processes are shown in Table 9.

5.2g. State Representations of Merged Addition Processes

If the state symbol of the summand digit-pair $(A_j, B_j)$ is $G_0$ or $G_1$, then a zero or non-zero carry is generated respectively; and a zero sum digit is produced in both cases. Thus the state symbols $G_0$ and $G_1$ must be differentiated because of the different generated carries. If the state symbol of $(A_j, B_j)$ is $P_0$ or $P_1$, then a zero carry and a sum digit "1" are generated in both cases. Therefore, the state symbols $P_0$ and $P_1$ shall not be differentiated and can be merged by the symbol P. The state representations S(n) and U(n) of the merged allowable and unallowable addition processes for all classes of summands are shown in Table 10.

5.3. ORGANIZATION OF ASSOCIATIVE PROCESSOR FOR ARITHMETIC OPERATION

The proposed system for our addition technique is shown in Fig. 19. It consists of an associative memory, a pair of shift registers and a six-state sequential machine which are introduced in the following sections separately.

5.3a. Associative Memory

Any associative memory possessing a double-sense line with outputs $Z_{ja}$ and $Z_{jb}$ at each bit position j can be employed in the system for arithmetic processing. In Fig. 19, the outputs $Z_{ja}$ and $Z_{jb}$ of the memory are used for obtaining the sensing results as designated in Table 6. When the number of tagged words during a sensing process is exactly two, the "sense 0", "sense 1" or "sense X"have identical bit-content with the state symbols $G_0$, $G_1$ or P respectively. Thus, we can utilize a sensing process in obtaining the state representations of the n digit-pairs if the number of tagged words is exactly two. In this case, the state representation of each summand digit-pair generated by a sensing process is actually denoted by a coded form as designated in Table 6.

5.3b. Shift Registers

Although the sensing results of the n bit-positions are generated

Table 9.

State Representations of Unallowable Addition Processes

| Classes of summands $u$ | State representations $U_u(n)$ |
|---|---|
| 1 | $U_1(n)= G_0(P_0+P_1)^{n-2}G_1+\sum_{i=0}^{n-3} G_0(P_0+P_1)^{i}G_1(P_0+P_1+G_0+G_1)^{n-i-2}$ |
| 2 | $U_2(n)= G_1(P_0+P_1)^{n-2}(P_0+P_1+G_0)+\sum_{i=0}^{n-3} G_1(P_0+P_1)^{i}G_0(P_0+P_1+G_0+G_1)^{n-i-2}$ |
| 3 | Does not exist. |

Table 10.

State Representations of Merged Addition Processes

| | |
|---|---|
| Allowable cases | $S(n)=\sum_{i=1}^{3} S_i(n)= 2^{n-2}G_0P^{n-2}(2P+G_0)+G_0^n+2^{n-2}G_1P^{n-2}G_1+$ $P(P+G_0)^{n-1}+2^nG_0(P+G_0)^{n-1}+2P(2P+G_0+G_1)^{n-1}+$ $\sum_{i=0}^{n-3} 2^i[(G_0P^iG_0+G_1P^iG_1)(2P+G_0+G_1)^{n-i-2}]$ |
| Unallowable cases | $U(n)=\sum_{1}^{2} U(n)= 2^{n-2}G_0P^{n-2}G_1+2^{n-2}G_1P^{n-2}(2P+G_0)+$ $\sum_{i=0}^{n-3} 2(G_0P^iG_1+G_1P^iG_0)(2P+G_0+G_1)^{n-i-2}$ |

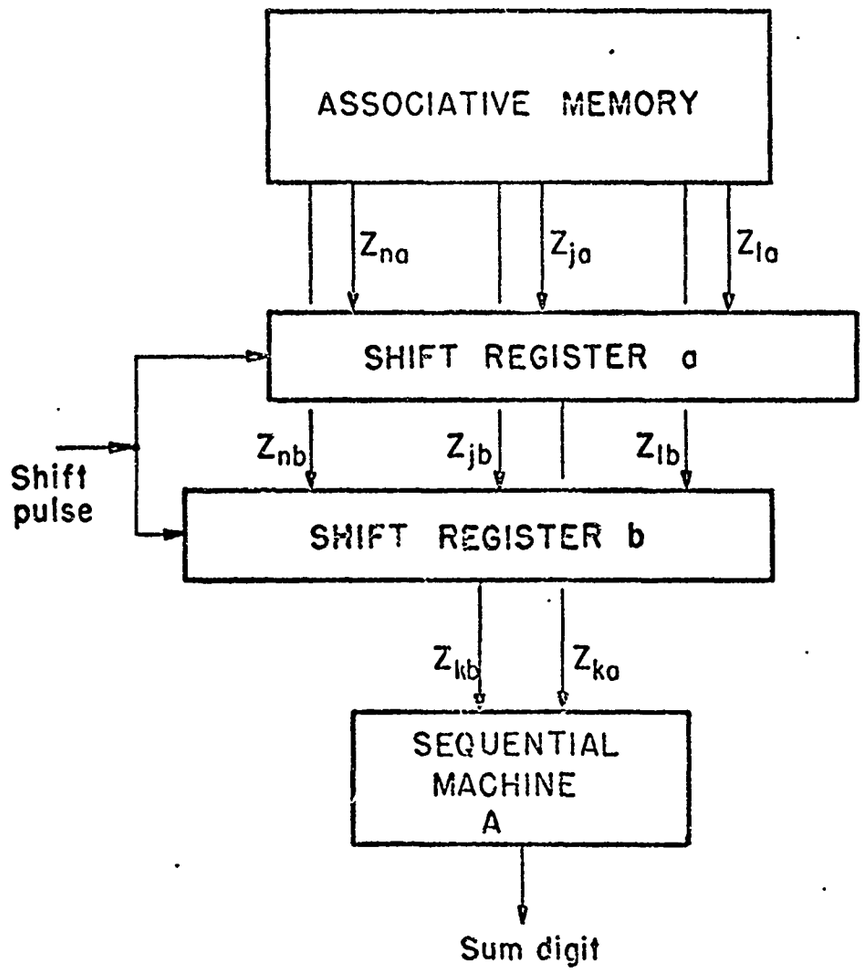



Fig. 19. An associative processor for arithmetic processing.

simultaneously, a pair of right shift registers is required for storing the sensing results, which then become the inputs to the sequential machine. As shown in Fig. 19, the shift registers a and b are employed to store the memory outputs $Z_{ja}$ and $Z_{jb}$ for j=1,2,...,n respectively, and the outputs $Z_{ka}$ and $Z_{kb}$ of the shift registers a and b feed the sequential machine as its inputs. The shift registers and the sequential machine should be operated synchronously.

5.3c. Sequential Machine

The sequential machine which is a Mealy type machine is used for transforming the coded state representation of each summand-pair to the final sum of that summand-pair. This machine possesses the input alphabet $\alpha_1=\{G_0,G_1,P,\#\}$, the output alphabet $\alpha_o=\{0,1,e,0_f\}$, and the state set $\alpha_s=\{s_0,s_1,\ldots,s_5\}$, where "#" is a marker symbol [37] indicating the beginning and end of an input string, "e" indicates a null output, and "$0_f$" denotes an overflow addition process. Since the direction of carry propagation is from the right to the left, both right and left end markers "#" denote the starting and terminating positions during transformation. In order to accommodate with the end markers and the n coded state symbols, there should be n+2 flip-flops in each shift register.

5.4. SYNTHESIS OF SEQUENTIAL MACHINE

5.4a. Machine A Synthesized From State Representations

Since different summand-pairs may have identical final sum when these summand-pairs have the same state representation, the redundancies contained in S(n) and U(n) as shown in Table 10 can be deleted by using the following relationships:

$$G_0^n \subset \sum_{i=0}^{n-3} G_0P^iG_0(P+G_0+G_1)^{n-i-2}$$

$$P(P+G_0)^{n-1} \subset P(P+G_0+G_1)^{n-1}$$

$$G_0(P+G_0)^{n-1} \subset \left[G_0P^{n-2}(P+G_0) + \sum_{i=0}^{n-3} G_0P^iG_0(P+G_0+G_1)^{n-i-2}\right]$$

Then, the distinct state representations $S^*(n)$ and $U^*(n)$ of merged addition processes for both allowable and unallowable cases are shown in Table 11.

A set of rules for transforming the input coded state symbols to the output sum digits can be depicted by the diagram shown in Fig. 20 in which

Table 11.

Distinct State Representations of Merged Addition Processes

| | |
|---|---|
| Allowable case | $S^*(n) = G_0P^{n-2}(P+G_0)+G_1P^{n-2}G_1 + P(P+G_0+G_1)^{n-1} + \sum_{i=0}^{n-3} [(G_0P^iG_0+G_1P^iG_1)(P+G_0+G_1)^{n-i-2}]$ |
| Unallowable case | $U^*(n) = G_0P^{n-2}G_1+G_1P^{n-2}(P+G_0) + \sum_{i=0}^{n-3} [(G_0P^iG_1+G_1P^iG_0)(P+G_0+G_1)^{n-i-2}]$ |

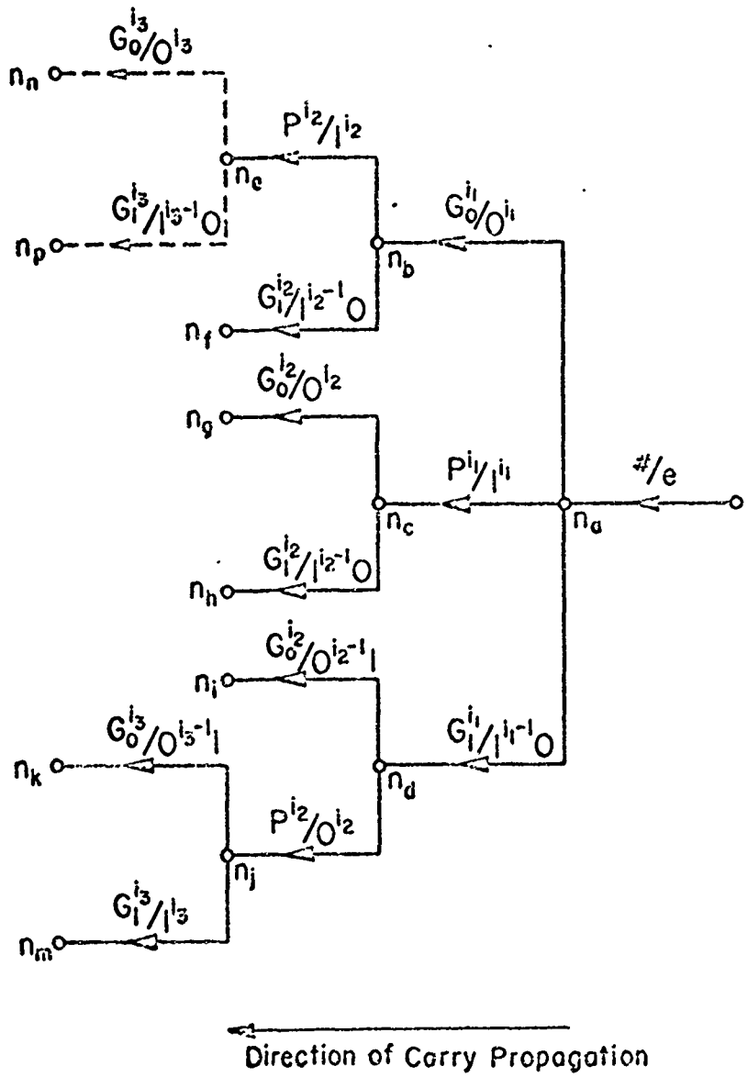



Fig. 20. Flow diagram showing the transformation rules.

the weight represents the input-output string-pair. When $i_1=n$, the possible input-output string-pair may be

$$G_0^n / 0^n,$$

$$P^n / 1^n,$$

or $$G_1^n / 1^{n-1}0$$

which is the weight of the directed branch* $(n_a,n_b)$, $(n_a,n_c)$ or $(n_a,n_d)$ respectively. Similarly, when $i_1+i_2=n$, the possible input-output string-pair consisting of two distinct types of state symbols, may be

$$P^{i_2} G_0^{i_1} / 1^{i_2} 0^{i_1},$$

$$G_1^{i_2} G_0^{i_1} / 1^{i_2-1} 0^{i_1+1},$$

$$G_0^{i_2} P^{i_1} / 0^{i_2} 1^{i_1}$$

$$G_1^{i_2} P^{i_1} / 1^{i_2-1} 0 1^{i_1},$$

$$G_0^{i_2} G_1^{i_1} / 0^{i_2-1} 1^{i_1} 0,$$

or $$P^{i_2} G_1^{i_1} / 0^{i_2} 1^{i_1-1} 0$$

which is the weight of the directed branch $(n_a,n_e)$, $(n_a,n_f)$, $(n_a,n_g)$, $(n_a,n_h)$, $(n_a,n_i)$ or $(n_a,n_j)$ respectively. Since the input-output string-pairs $G_0^{i_3} / 0^{i_3}$ and $G_1^{i_3} / 1^{i_3-1} 0$ which appear as the weights of the dotted directed branches $(n_e,n_n)$ and $(n_a,n_p)$ are identical with those string-pairs which are the weights of the directed branches $(n_c,n_g)$ and $N_c,n_h)$ respectively when $i_3=i_2$, the node $n_e$ is equivalent to $n_c$ and those dotted directed branches can be omitted in the diagram. By the same token, the nodes of $n_f$, $n_h$, and $n_m$ are equivalent to $n_d$ and the nodes $n_g$, $n_i$ and $n_k$ to $n_b$. Hence, the diagram may terminate at the nodes $n_e$, $n_f$, $n_g$, $n_h$, $n_i$, $n_k$ and $n_m$. The direction of each branch shows the direction of carry propagation.

The state graph of sequential machine A can be easily realized based on the transformation rules and the state representations $S^*(n)$ and $U^*(n)$ by setting the value of n up to 5. When $n \geqq 6$ we have found that the state graph of the sequential machine is identical with the one for n=5.

---

*A directed branch of the state graph is denoted by the ordered-pair $(n_i,n_j)$ where $n_i$ and $n_j$ represent the starting node and the termination node respectively.

The minimal state graph of machine A is shown in Fig. 21, where $s_0$ is the beginning and end state. When machine A transits to the state $s_3$ or $s_4$, a zero carry is produced. When machine A transits to the state $s_2$ or $s_5$, a nonzero carry is generated.

5.4b. Disjoined Sequential Machine

Since the state-transition of machine A is sequential, the operating speed is limited by the serial-by-bit operation. This can be improved by grouping the n summand digit-pairs into two or more sections with approximately the same number of digits. If two sections are used, the section on the left-hand side consists of two machines A and B whose inputs are parallelly connected and the section on the right-hand side has only one machine A'. The state graph of machine B shown in Fig. 22 is obtained in the same way as for machine A by adding a nonzero carry to the first bit-position of the section on the left-hand side. The state graph of machine A' as shown in Fig. 23 is found by modifying that of machine A as follows: When machine A' reaches the state $s_2'$ or $s_5'$ during the last digit transformation for the section on the right-hand side, the next input-output pair becomes $\#/(e,\theta_A)$ where $\theta_A$ is a control signal [37], which activates a logic circuit to select machine A in the section on the left-hand side. On the other hand, when machine A' reaches the state $s_3'$ or $s_4'$, the next input-output pair becomes $\#/(e,\theta_B)$, where $\theta_B$ is the control signal to select machine B. Thus, the addition speed is almost doubled when the n summand digit-pairs are grouped into two sections. It is noted that the speed can be increased further if more sections are used. However, no matter how many sections the n summand digit-pairs are grouped, every section, except the extremely right one and the extremely left one, consists of machines A' and B', where machine B' is obtained by adding the output control signals $\theta_A$ and $\theta_B$ to the appropriate state transitions of machine B. The extreme right section has machine A' only, and the extreme left section consists of machines A and B. Although this method of state transitions of increasing the addition speed is similar to that of the conditional sum adder proposed by Sklansky [38], yet the rate of increase of hardware in our system is much smaller.

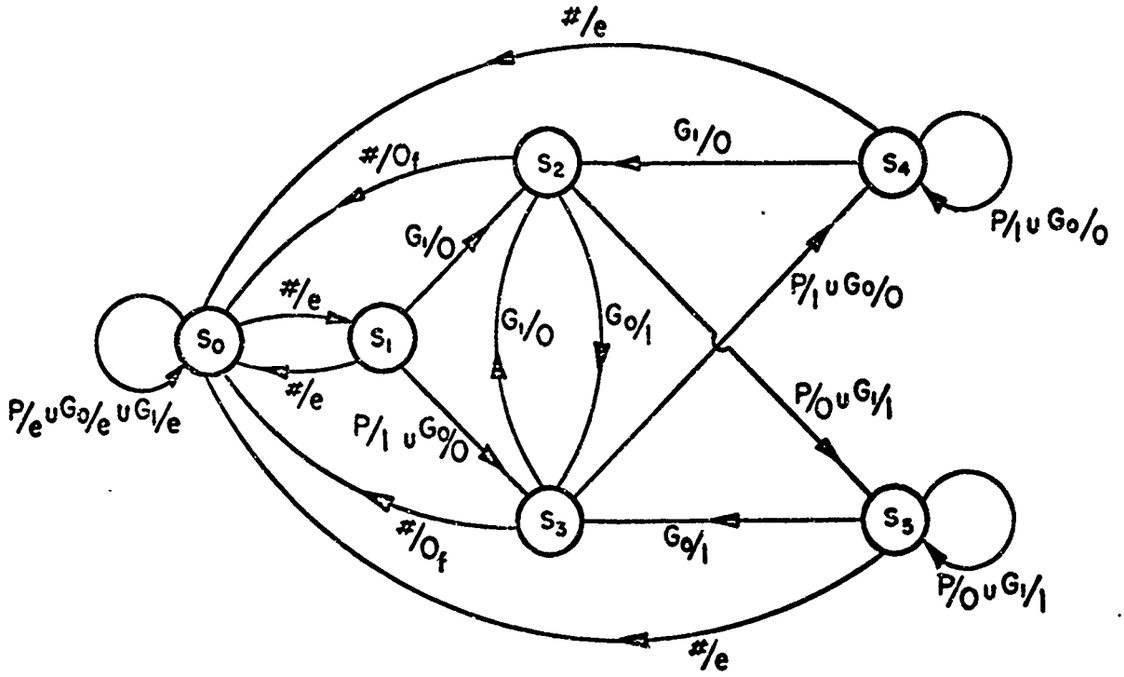



Fig. 21. Machine A.

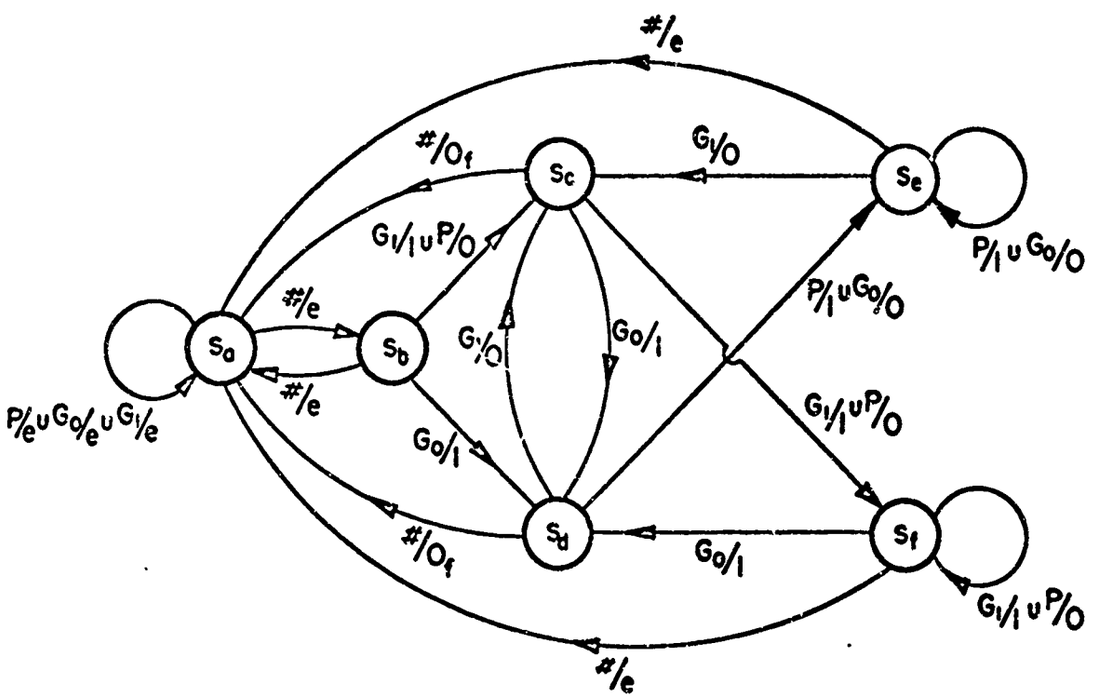



Fig. 22. Machine B.

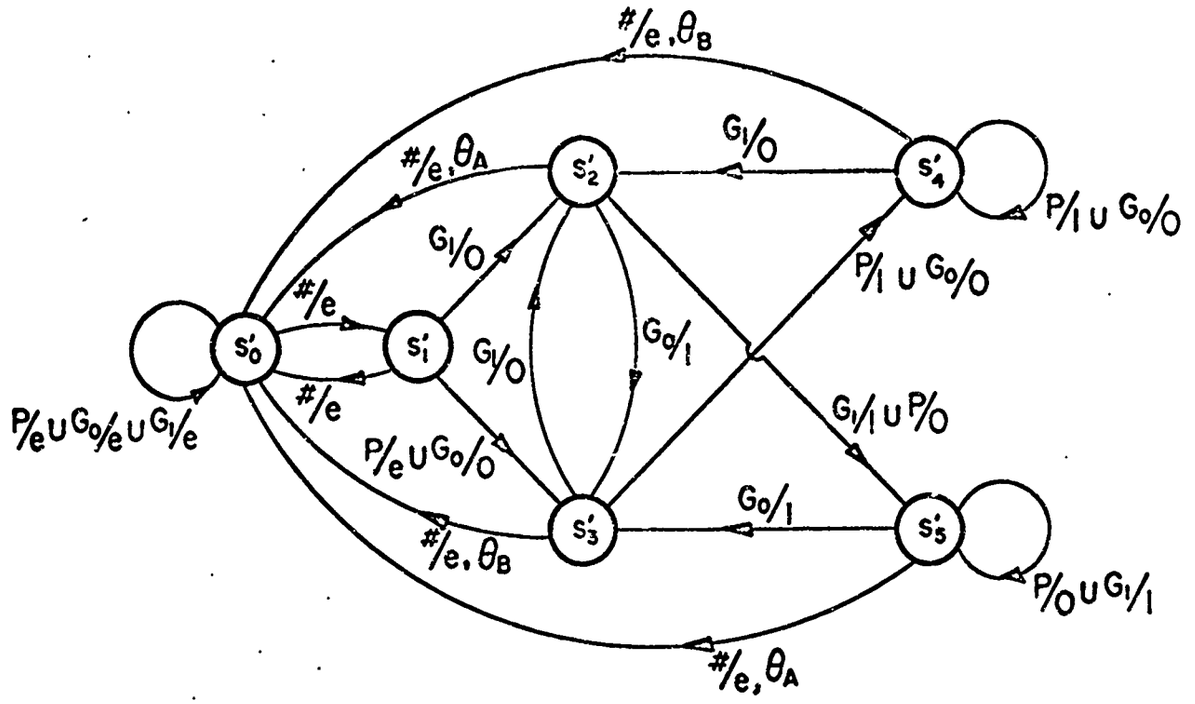



Fig. 23. Machine A'.

## 6. DISCUSSION

For all cryogenic associative memory systems, the speed appears to be the main problem because the operations of cryotron circuits are relatively slower than that of magnetic cores, cutpoint cells and some other devices. In our cryogenic associative memory system, we have attempted to increase the operating speed in various aspects. For example, each matched word $i$ is selected by only a pair of cryotrons $Q_i$ and $\bar{Q}_i$, and consequently there is only one stage of delay caused by the switching time $T_Q$ of $Q_i$ and $\bar{Q}_i$ between the end of comparison process and the subsequent process such as reading, writing, etc. The match-indication voltage $V_{MI}$ is used as the signal for automatic termination of sequential writing process or non-ordered information retrieval. Excessive empty words can be dematched by the current pulse $I_D$ as soon as all data words are stored in the memory. Thus unnecessary delays in these operations are eliminated.

In the cutpoint cellular associative memory system, the delay of the tree-type circuit can be reduced if we use OR-gates with more inputs instead of using cutpoint cells. It is noted that the stage delay $T_k$ of a cutpoint cell with index k is assumed to be different for various values of k. Because these cells are identical in structures with only different cutpoint, their stage delays may be compensated so that they are all the same. Then, the hardware design based on stage delay analysis is much simplified.

In pattern recognition, the capability of our scheme largely depends on the determination of the templates for pattern classes, further research is required to establish a method in finding a set of templates, which allows large deviations of each pattern class and reduces the number of mis-classification. Another direction of research in this area is to extend this recognition scheme from classifying noisy patterns based on the concept that $a_{ij}$= 0(black) if the lightest figure element in $a_{ij}$ of the input pattern is darker than the darkest part of the background and $a_{ij}$= 1(white) if otherwise. In this case, the darkest part of the background in the input pattern constitutes a threshold value and consequently a test is required when the input pattern is transformed to an initial interrogating word for the first comparison process.

## APPENDIX I

Some of the technical terms used in chapter 2 are defined as follows: An empty memory word or simply empty word is a word with all of its bit-memories containing zero persistent current. An occupied memory word or simply occupied word means a word with at least one of its bit-memories containing the information bit "0" or "1" represented respectively by the nonzero clockwise or counterclockwise persistent current circulating in the memory loop.

An interrogating word is a sequence of n bits consisting of the elements in the set { 0,1,e,Ø}, where n is the total number of bits per word, and e and Ø represent the "empty" and "don't-care" respectively. It is used for simultaneous comparison with all the memory words, which are called the interrogated words. The interrogating bits are defined in Table 1.

A matched word is an interrogated word which is matched with the interrogating word. A mismatched word is the interrogated word which is not matched with the interrogating word. The first matched word means a matched word which occupies the word-position numbered with the smallest number among all the matched words.

Ordered retrieval of information is a retrieving scheme which reads out the desired words sequentially in an ascending (or descending) order of the magnitudes of their binary equivalent values; whereas non-ordered retrieval of information is a retrieving scheme which reads out the desired words sequentially according to the order of their word locations in the memory system rather than the magnitudes of their binary equivalent values.

The identifier of a word is the portion of the word-content which is used for the comparison process during the retrieval of that word from a memory system, and the remaining portion of the word is called the don't-care part, which is to be masked out during the comparison process. The descriptor of a set A of words is a bit-position, which has the same content for all the words in A.

# APPENDIX II

## Derivation of the Stage Delay $T_{ys1}$

Referring to Fig. 12, the stage delay $T_{ys1}$ between the leading edge of $Y_j$ and the pulse $H_1^{(1)}$ is found to be

$$T_{ys1} = T_{yb} + T_{1k_1} - T_{si1}, \tag{II.1}$$

where $T_{yb}$ and $T_{si1}$ are given by (28) and (31) respectively. The stage delay $T_{1k_1}$ between the leading edges of the control current $I_1$ and $I_{k_1}$ is the sum of the following three terms: The fir t term is $(k_1-2)(T_4+T_5)$ which is the stage delay between the leading edges of $I_1$ and $I_{k_1-1}$. The second term is $(T_2+T_s^{(1)})$ which is the stage delay between the leading edge of $I_{k_1-1}$ and the shift pulse $H_{k_1-1}^{(1)}$, and the third term is $(T_4+2T_5+T_{13})$ between the pulse $H_{k_1-1}^{(1)}$ and the leading edge of $I_{k_1}$. Therefore, $T_{1k_1}$ is given by the equation

$$T_{1k_1} = T_2 + (k_1-1)T_4 + k_1T_5 + T_{13} + T_s^{(1)}. \tag{II.2}$$

Substituting (28), (31) and (II.2) into (II.1), we obtain

$$T_{ys1} = T_1+T_2+T_3+(k_1-2+\left[\log_2 n\right])\,T_4+T_6+T_s^{(1)}. \tag{II.3}$$

In order to include the possibility that the first matched word is the last memory word m, we have to set

$$T_s^{(1)} = (m-k_1)\,T_4. \tag{II.4}$$

The stage delay $T_{ys1}$ which is given by (30) yields when (II.4) is substituted in (II.3).

## Derivation of the Stage Delay $T_{yx}$

Similar to the above method, the stage delay $T_{yx}$ between the leading edges of $Y_j$ and $X_j^{(1)}$ is found to be

$$T_{yx} = T_{ys1} + T_{si1} + T_{iw} + T_w^{(1)}. \tag{II.5}$$

Substituting (30), (31) and (32) into (II.5), we have

$$T_{yx} = T_1+T_2+T_3+(m-1+\left[\log_2 n\right])\,T_4 + k_1T_5+T_6+2T_{13}+T_{2,5}+T_w^{(1)}. \tag{II.6}$$

We have to set

$$T_w^{(1)} = (m-k_1)\,T_5 \tag{II.7}$$

in order to include the possibility of the first matched word being at the position m. The stage delay $T_{yx}$ which is given by (33) yields when (II.7) is substituted into (II.6).

Derivation of the Time Interval $T_{ss}$ or $T_{xx}$

It follows from Fig. 12 that the time interval $T_{ss}^{(1)}$ between the shift pulses $H_1^{(1)}$ and $H_1^{(2)}$ is given by the equation

$$T_{ss}^{(1)} = T_{si1} + T_{iw} + T_w^{(1)} + T_{xb} + T_{b,k_{1,2}} - T_{si2}, \tag{II.8}$$

where $T_{si1}$, $T_{iw}$ and $T_{xb}$ are given by (31), (32) and (34) respectively. The stage delay $T_{b,k_{1,2}}$ between the trailing edge of $M_{Bk_1}$ and the leading edge of $I_{k_1+k_2}$ is the sum of the following three terms: The first term is $T_s^{(2)}$ which is the possible delay between the trailing edge of $M_{Bk_1}$ and the shift pulse $H_{k_1-1}^{(2)}$ initiated by $H_1^{(2)}$, the second term is $k_2T_5$ which is the delay between the pulses $H_{k_1-1}^{(2)}$ and $H_{k_1+k_2-1}^{(2)}$, and the third term is $(T_4+2T_5+T_{13})$ which is the delay between the shift pulse $H_{k_1+k_2-1}^{(2)}$ and the leading edge of $I_{k_1+k_2}$. Thus, $T_{b,k_{1,2}}$ is given by the equation

$$T_{b,k_{1,2}} = T_4+(k_2+2)\,T_5+T_{13}+T_s^{(2)}. \tag{II.9}$$

It follows from (31) that the stage delay $T_{si2}$ between the shift pulse $H_1^{(2)}$ and the leading edge of $I_{k_1+k_2}$ is found to be

$$T_{si2} = T_4+(k_1+k_2)\,T_5+T_{13}. \tag{II.10}$$

Substituting (31), (32), (34), (II.9) and (II.10) into (II.8) we have

$$T_{ss}^{(1)} = T_1+T_3+(1+\lceil\log_2 n\rceil)\,T_4+3T_5+T_6+3T_{13}+ T_{3,4}+T_{2,5}+T_s^{(2)}+T_w^{(1)}. \tag{II.11}$$

The time interval $T_{ss}^{(g)}$ between the shift pulses $H_1^{(g)}$ and $H_1^{(g+1)}$ can be obtained by modifying (II.11). It is given by the equation

$$T_{ss}^{(g)} = T_1+T_3+(1+\lceil\log_2 n\rceil)\,T_4+3T_5+T_6+3T_{13}+ T_{3,4}+T_{2,5}+T_s^{(g+1)}+T_w^{(g)}. \tag{II.12}$$

To allow the possibility that the last tagged word $k_1+k_2\ldots+k_r$ occurs at the last memory word m, we set

$$k_1+k_2+\ldots+k_r = m. \tag{II.13}$$

Moreover, the stage delay $T_{sx}^{(g)}$ between the shift pulse $H_1^{(g)}$ and the writing pulse $X_j^{(g)}$ is given by the equation

$$T_{sx}^{(g)} = T_4 + (\sum_{j=1}^{g} k_j)\ T_5 + 2T_{13} + T_{2,5} + T_w^{(g)}. \tag{II.14}$$

which is obtained from Fig. 12, (31) and (32). Because we intend to keep the time intervals $T_{sx}^{(g)}$, to be the same for g=1,2,...,r, we have

$$T_w^{(g)} = (m - \sum_{j=1}^{g} k_j)\ T_5, \tag{II.15}$$

which is obtained from (II.7), (II.13) and (II.14). If we set

$$T_s^{(g+1)} = (\sum_{j=1}^{g} k_j)\ T_5,\ g \geqq 1 \tag{II.16}$$

and substitute (II.15) and (II.16) into (II.12), we obtain

$$T_{ss} = T_1 + T_3 + (1 + \left[\log_2 r\right])\ T_4 + (m + 3)\ T_5 + T_6 + 3T_{13} + T_{3,4} + T_{2,5}, \tag{II.17}$$

in which we have dropped the superscript (g) because (II.17) is independent of the position of the tagged word. It is noted that (II.17) is exactly the same as (35), which we want to derive.

## BIBLIOGRAPHY


[1] A. E. Slade and H. O. McMahon, "A Cryotron Catalog Memory System", Proc. of Eastern Joint Computer Conf., pp. 115-120; December, 1956.

[2] A. E. Slade and C. R. Smallman, "Thin Film Cryotron Catalog Memory", Automatic Control, Vol. 13, pp. 48-50; August, 1960. Also: Solid-State Electronics, pp. 357-362; Sept., 1960.

[3] R. R. Seeber, Jr., "Associative Self-Sorting Memory", Proc. of Eastern Joint Computer Conf., pp 179-187; December, 1960.

[4] W. L. McDermid and H. E. Peterson, "A Magnetic Associative Memory System", IBM J. Res. and Dev., Vol. 5, pp. 59-62; January, 1961.

[5] J. R. Kiseda, H. E. Peterson, W. C. Seelback and M. Teig, "A Magnetic Associative Memory", IBM J. Res. and Dev., Vol.5, pp. 106-121; April, 1961.

[6] E. H. Frei and J. Goldberg, "A Method for Resolving Multiple Responses in a Parallel Search File", IRE Trans. on Electronic Computers, EC-10, pp. 718-729; Dec., 1961.

[7] R.R. Seeber Jr., and A. B. Lindquist, "Associative Memory with Ordered Retrieval", IBM J. Res. and Dev., Vol. 6, pp. 126-136; January, 1962.

[8] P. M. Davies, "A Superconductive Associative Memory", Proc. of Spring Joint Computer Conf. Vol. 21, pp. 79-88; May, 1962.

[9] V. L. Newhouse and R. E. Fruin, "A Cryogenic Data Addressed Memory", ibid, pp. 89-99.

[10] R. F. Rosin, "An Organization of an Associative Cryogenic Computer", ibid, pp. 203-212.

[11] M. H. Lewin, "Retrieval of Ordered Lists from a Content-Addressed Memory", RCA Review, Vol. 23, pp. 215-229; June, 1962.

[12] A. D. Falkoff, "Algorithms for Parallel-Search Memories", J. of Association for Computing Machinery, Vol. 9, pp 488-511; Oct., 1962.

[13] R. R. Lussier and R. P. Schneider, "All-Magnetic Content Addressed Memory", Electronic Industries, pp. 92-98; March, 1963.

[14] G. Estrin and R. H. Fuller, "Algorithms for Content-Addressable Memories", Proc. of Pacific Computer Conf., pp. 118-130; March, 1963.

[15] E.S. Lee, "Semiconductor Circuits in Associative Memories", ibid, pp. 96-108.

[16] P. M. Davies, "Design of an Associative Computer", ibid, pp.109-117.

[17] E. S. Lee, "Associative Techniques with Complementing Flip-Flops", Proc. of Spring Joint Computer Conf., Vol. 23, pp. 381-394; 1963.

[18] C. Y. Lee and M. C. Paull, "A Content Addressable Distributed Logic Memory with Applications to Information Retrieval", Proc. of IEEE, Vol. 51, pp. 924-932; June,1963.

[19] R. W. Ahrons, "Superconductive Associative Memories", RCA Rev., Vo. 24, pp. 325-354; September, 1963.

[20] G.Estrin and R. H. Fuller, "Some Applications for Content-Addressable Memories", Proc. of Fall Joint Computer Conf.,Vol.24, pp. 495-508; 1963.

[21] R. R.Seeber and A. B. Lindquist, "Associative Logic for Highly Parallel Systems", ibid, pp. 489-493.

[22] A. Kaplan, "A Search Memory Subsystem for a General-Purpose Computer", ibid, pp. 193-200.

[23] G. J. Simmons, "Application of an Associatively Addressed, Distributed Memory", Proc. of Spring Joint Computer Conf.,Vol. 25, pp. 493-513; 1964.

[24] J. D. Barnard, F. A. Behnke A. B. Lindquist, and R. R. Seeber, "Structure of a Cryogenic Associative Processor", Proc. IEEE, Vol. 52, pp. 1182-1190; October, 1964.

[25] J. E. McAteer, J. A. Capobianco and R. L. Koppel, "Associative Memory System Implementation and Characteristics", Proc. of Fall Joint Computer Conf., Vol. 26, Part 1, pp. 81-92; 1964.

[26] R. G. Ewing and P. M. Davies, "An Associative Processor", ibid pp. 147-158.

[27] B. A. Crane and J. A. Githens, "Bulk Processing in Distributed Logic Memory", IEEE Trans. on Electronic Computers, EC-14, pp. 186-196; April, 1965.

[28] Y. H. Chu, "Application of Content-Addressed Memory for Dynamic Storage Allocation", RCA Review, Vol. 26, pp. 140-152; March,1965.

[29] Y. H. Chu, "A Destructive-Readout Associative Memory", IEEE Trans. on Elec. Computers, EC-14, pp. 600-605; Aug., 1965

[30] R. H. Fuller and R. M. Bird, "An Associative Parallel Processor with Application to Picture Processing", Proc. of Fall Joint Computer Conf., Vol. 28, pp. 105-116; 1965.

[31] C. T. Gibson, "Time-Sharing in the IBM System /360: Model 67, "Proc. of Spring Joint Computer Conf., Vol. 29, pp. 61-78; 1966.

[32] R.C. Minnick, "Cutpoint Cellular Logic", IEEE Trans. on Elec. Computers, EC-13, pp. 685-698; December, 1964.

[33] E.S. Schlig, "Tunnel Diode-Cryotron Circuit for Integration of Cryotonic Systems", Proc. IEEE, Vol. 51, pp. 1359-1360; Oct.,1963.

[34] J. P. Pritchard, Jr. "Superconducting Thin-Film Technology and Applications", IEEE Spectrum, pp. 46-54; May, 1966.

[35] S. H. Unger, "Pattern Detection and Recognition", Proc. of IRE, vol. 47, pp. 1737-1752; October, 1959.

[36] Salomon Klaczko-Ryndzinn, "A Threshold-Conditioned Adaptive Template Procedure for Recognition of Normalized Connected Patterns", Computer and Information Sciences, edited by J. T. Tou and R. H. Wilcox, Spartan Books, Inc., Washington, D. C., 1964; pp. 262-287.

[37] G. K. Krulee and D. J. Kuck, "Finite State Models for Perception", J. of Mathematical Psychology, Vol. 1, pp. 316-335; July, 1964.

[38] J. Sklansky, "Conditional-Sum Addition Logic", IRE Trans. on Electronic Computers, EC-9, pp. 226-231; June, 1960.

ABSTRACT (continued)

shift registers and Mealy type sequential machines. This system may have the advantage of reducing addition time by paying a limited price in the additional hardware relative to the existing software approaches for arithmetic algorithms.

Unclassified
Security Classification

## DOCUMENT CONTROL DATA - R&D

*(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)*

| | |
|---|---|
| 1 ORIGINATING ACTIVITY *(Corporate author)*<br>Information-Processing And Control SYSTEMS LABORATORY<br>Northwestern University<br>Evanston, Illinois 60201 | 2a REPORT SECURITY CLASSIFICATION<br>Unclassified<br>2b GROUP<br>Not applicable |

3 REPORT TITLE
ASSOCIATIVE MEMORY SYSTEMS AND THEIR APPLICATIONS TO PICTURE AND ARITHMETIC PROCESSINGS

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Not applicable

5 AUTHOR(S) *(Last name, first name, initial)*
Yang, Chao C.

| | | |
|---|---|---|
| 6 REPORT DATE<br>August 1966 | 7a TOTAL NO OF PAGES<br>85 | 7b NO OF REFS<br>38 |
| 8a CONTRACT OR GRANT NO<br>AF-AFOSR-98-65 and<br>Contract No. N00014-66-C0020-A03<br>b PROJECT NO<br>NR 373-502/3-14-66 Electronics Branch<br>c. Not applicable<br>d Not applicable | 9a ORIGINATOR'S REPORT NUMBER(S)<br>Technical Report 66-103<br>9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*<br>Not applicable | |

10 AVAILABILITY/LIMITATION NOTICES
Qualified requesters may obtain copies of this report from the Defense Documentation Center

| | |
|---|---|
| 11 SUPPLEMENTARY NOTES<br>Not applicable | 12. SPONSORING MILITARY ACTIVITY<br>Air Force Office of Scientific Research, Washington, D. C.; and Office of Naval Research, Washington, D. C. |

13 ABSTRACT

Two different associative memory systems implemented by cryotrons and cutpoint cells are presented. The cutpoint cellular associative memory designed on the basis of stage delay analysis does not use any ladder structure, but employs a high-speed controlled shift register for the sequential tagging of all matched memory words. Both systems are so mechanized that their structures are simple and especially compatible with the batch fabrication of integrating circuits. Moreover, both systems can perform ordered information retrieval following Lewis's sorting scheme, template-matching pattern recognition based on the union-intersection concept and nonbulk two-summand additions of signed binary numbers in the signed-2's-complement number system. The basic operations of each system include comparison, reading, writing, sensing and output-summing processes being all performed on the basis of parallel-by-bit. During simultaneous comparison, a bit-position corresponding to all bit-memories of equal significance can be masked out. In pattern recognition, each pattern class is allowed to have deviations in size, style, orientation, etc. within certain limits. Since each prestored template for a pattern class can accommodate to a number of deviated input patterns belonging to the same class, the storage capacity can be greatly saved. Because most input patterns are recognized within first processing cycle and no patterns require more than two comparison processes for their recognition, the pattern recognition scheme is simple. The speed is very high because of this simple scheme as well as the parallel processing capability of an associative memory. The processing system for arithmetic operations consists of an associative memory, right

DD FORM 1 JAN 64 1473 D23558

continued on next sheet ----

Unclassified
Security Classification

Unclassified
Security Classification

| 14. KEY WORDS | LINK A ROLE | LINK A WT | LINK B ROLE | LINK B WT | LINK C ROLE | LINK C WT |
|---|---|---|---|---|---|---|
| 1. Cryogenic associative memory | | | | | | |
| 2. Cutpoint cellular associative memory | | | | | | |
| 3. Ordered and non-ordered information retrieval | | | | | | |
| 4. Template-matching pattern recognition scheme | | | | | | |
| 5. Nonbulk addition technique | | | | | | |
| 6. Comparison, reading, writing and sensing processes | | | | | | |
| 7. Automatic dematch | | | | | | |
| 8. Output-summing processes | | | | | | |
| 9. Parallel-by-bit operation | | | | | | |
| 10. Stage-delay analysis | | | | | | |
| 11. State representation | | | | | | |

## INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through _____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through _____."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through _____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, roles, and weights is optional.