Technical Report CS76009-R

FAULT LOCATION IN A SEMICONDUCTOR RANDOM-ACCESS MEMORY UNIT

Vason P. Srini

July 1976

Department of Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061

Keywords: RAM unit, RAM chip, Cables, fault model, fault location, diagnostic graph, test sequence, controlled register

CR categories: 6.34, 6.1

ABSTRACT

A semiconductor random-accessor memory unit (RAM unit) is a connection of RAM chips, Data Cable, Chip Select Cable, and Address Cable so that each storage element can be selected for writing or reading independent of previous write or read. The faulty RAM unit is represented by a model consisting of four types of faults: RAM chip faults, D-fault, CS-fault, and A-fault. The testing of a RAM unit and locating faults to the RAM chips or wires in the various cables is considered. A set of six tests has been designed to diagnose the faults in the model. The tests are used in defining a relation, "test tj is invalid", on the fault model. The diagnostic graph of a RAM unit is drawn by using this relation and a sequence in which tests have to be performed obtained from this graph. By using this sequence faulty components in a RAM unit are located when at most one type of fault in the model is present.

The symmetric array organization of storage elements in RAM chips is used in developing the tests with minimal length. Test generation is using the operations increment, decrement, compare and rotate, and quite easy to program.

I. INTRODUCTION

Large-Scale integrated random-access memories (RAM chips) have been increasingly used in Computer and Control systems. Functional testing of these RAM chips has been investigated by several authors [1-5]. The presence of a large number of storage elements in a RAM chip makes extensive functional testing very time consuming.

It has been shown experimentally that certain faults in memories can be detected by performing pattern-sensitivity tests [6]. These tests treat the memory as a "black box" and this approach follows the sequential machine identification philosophy. Hayes [7] has formalized the pattern sensitive testing of a RAM chip by defining an incompletely specified sequential machine with 2^N states and 3 N inputs, where N is the number of storage elements in a RAM chip. He has also shown that if unrestricted pattern sensitive faults (PSF) are present in the machine then an algorithm generated a checking sequence of length $(3\ N^2 + 2\ N)\ 2^N$, which has been shown to be computationally infeasible. However, if the fault is restricted to single PSF (SPSF) and if each storage element is a member of α distinct neighborhoods, then a checking sequence of length $(3\ \alpha^2 + 2\ \alpha)2^\alpha N$ is sufficient. With $\alpha = 5$ and $N > 4000$, the length of checking sequence for SPSF is 2720 N, which is quite large and thus making the checking sequence approach economically unattractive.

In most computer main frames the amount of memory needed is far beyond the number of storage elements in a RAM chip. So, memory units (RAM units) are built using a number of RAM chips organized in rows and columns, support circuitry for the RAM chips, and wires (lines) for selecting a storage element in a RAM chip,

writing and reading data. The problem of testing RAM units is considered. Since the RAM unit is an expensive part to be replaced on detecting a fault, it is also necessary to locate the fault to RAM chips or lines in the RAM unit. The fault detection and location in a RAM unit is performed by developing a fault model and devising efficient tests by using some of the structural properties of RAM chips.

Most of the faults in the logic circuits and lines in a RAM chip can be represented by stuck-at-0 (1) fault on the lines providing input to, and carrying output from these circuits [5,8]. Many of the faults in the storage elements can be represented by the state of a storage element stuck-at-0 (1). The high density of circuits and the characteristics of storage element circuits in MOS and Bipolar technologies present an interaction between the states of neighboring storage elements [1, 7, 9, 10]. This interaction can be represented by adjacent pattern interference fault, which is defined to be the change in the content of a storage element when only the contents of its neighbors are altered. The faults in the various lines and support circuitry of a RAM unit can be represented by stuck-at-0 (1) fault on the lines [5, 10, 11]. With this knowledge, the fault model for a RAM unit is developed in Section II. The model consists of four types of faults: D-fault, CS-fault, A-fault and RAM chip faults.

One of the desired characteristics for the mass production of RAM chips is a simple layout for the circuits of storage elements, support circuits, and connections for accessing the storage elements, with conductors running straight and fewer number of bends [9, 10, 12, 13]. By organizing the storage elements in an array, the connections for accessing it could be made by using straight conductors. RAM chips, like other large-scale integrated circuits (e.g. LSI

processor) have pin limitation. The number of pins can be reduced by using a symmetric array. For example, suppose there are 2^{2p} storage elements in a RAM chip and if it is organized in a two dimensional array with 2^p columns and rows, then it is possible to select any storage element for writing or reading independent of previous operations, by using p pins and multiplexing the address. Without the symmetric array the RAM chip would have required 2p pins for addressing. This symmetric array organization of RAM chips is utilized in designing tests for a RAM unit.

It is easy to design tests so that a test for a type of fault always fails when that type of fault alone is present and always passes for all types of faults absent. But a test for a given type of fault might fail for some other type of fault in the fault model. So the tests by themselves cannot be used for fault location. The relation "test tj is invalid" is defined on the types of faults in the fault model and it is used to draw the diagnostic graph [14] for a RAM unit. If the diagnostic graph has no cycles, then a sequence in which the tests have to be performed, can be obtained. By using this sequence faulty components in the RAM unit can be located when at most one type of fault in the model is present. The tests for the fault model are developed in sections III and IV and the diagnostic graph in section V.

A RAM unit is tested for D-fault by EXPT 1. The result of testing is available in the matrices M_0 and M_1, where the columns correspond to data lines in the RAM unit. If the entries of the i-th column of M_0 and M_1 are 0 (1), then fault is located to the corresponding data line. The presence of CS-fault is tested by EXPT 2. The result of testing is available in the matrix S. Each row of S correspond to a chip select line in the RAM unit . If the binary value

of the U-th row of S is not equal to (U-1), then the fault is located to the corresponding chip select line. Each row of RAM chips in a RAM unit is tested by EXPT 3 to locate A-fault. The result of testing the ℓ-th row of RAM chips is available in the matrix R_ℓ. The rows of R_ℓ except the last correspond to adress lines for accessing storage elements. If the binary value of the i-th row of R_ℓ is not equal to i then the fault is located to the i-th address line in the ℓ-th row of the RAM unit. EXPT 4 thru 6 test for RAM chip faults and output faulty RAM chips.

Test generation is quite simple using the operations increment, decrement, compare and rotate. The length of tests are minimal with respect to symmetric array organization. If the RAM unit has q rows of RAM chips with N storage elements in an array of 2^p rows and columns for each RAM chip, then the total test length $\cong 224\,N$, which is a twelfth of the length of checking sequence for SPSF [7].

II. FAULT MODEL OF RAM UNIT

A random access memory M is defined to be a set $C_0, C_1, \ldots, C_K, \ldots, C_{N-1}$ of binary stoarge elements, e.g. flip-flops [7]. The subscript K of C_K is its address. Each storage element can be considered to be an incompletely specified sequential machine (Mealy type) with 2 states, 3 inputs and 2 outputs. The state table of the machine is shown in Figure 1. Any storage element in M may be selected for reading or writing independent of the previous READ or WRITE operations.

The state of a storage element C_K is stuck-at-0 (1) if the output is 0(1) for the input sequence $W_K R_K (\overline{W}_K R_K)$.

An output Cable Z is a wire (line) carrying the binary output signal from each storage element in M.

A row (column) decoder DC is a 1 - 1 and onto map from the set of p-tuples to the set of 2^p outputs.

row (column) decoder is faulty if the mapping is not 1 - 1 and onto.

A row (column) cable R is a set $r_0, r_1, \ldots, r_{(2^p-1)} (c_0, c_1, \ldots, c_{(2^p-1)})$ of wires (lines) carrying binary signals from row (column) decoder output (source) to M.

A row (column) line $r_i [c_i]$ is stuck-at-0 (1), if the source is at 1(0) and the sink is at 0(1).

neighborhood of a storage element C_K is defined to be the set N_K of storage elements (including C_K), which can be considered near to C_K in some sense.

Adjacent neighborhood of C_K is defined to be the set of all physically adjacent neighbors. If the storage elements are arranged to be in a two

dimensional array, then the adjacent neighborhood of $C_{i,j}$ is the set $N_{i,j}$ containing $C_{i,j}$ and the four neighbors of $C_{i,j}$, one in each sense in the dimensional directions ($C_{(i-1),j}$, $C_{(i+1),j}$, $C_{i,(j-1)}$ and $C_{i,(j+1)}$).

The entire i-th row and j-th column of $C_{i,j}$ is defined to be the extended neighborhood of $C_{i,j}$. If the extended neighbors of $C_{i,j}$ are in state 0(1) and the output of $C_{i,j}$ is 0(1) for the input sequence $W_{i,j}\ R_{i,j}$ ($\overline{W}_{i,j}\ R_{i,j}$) then $C_{i,j}$ has an extended 0(1) fault.

If for any of the possible $\{0,1\}^{(N_K-1)*}$ combination of states of the adjacent neighbors of C_K, the output of C_K is 0(1) when the input sequence is $W_K R_K (\overline{W}_K R_K)$, then C_K has an adjacent pattern interference fault. In other words, the change in the content of a storage element when only the contents of its neighbors are altered is defined to be the adjacent pattern interference fault. Note that if $N_K = 5$, there is a total of 32 distinct combinations of states for the adjacent neighborhood of C_K.

A RAM chip is a connection of M, DC, R and Z so that every storage element may be selected for reading or writing independent of previous READ or WRITE. A typical RAM chip is shown in Figure 2 and the function of each unit is described in [10,12]. A RAM chip is faulty if any of the above mentioned faults is present.

A Data Cable D is a set $d_1, d_2, \ldots, d_W$ of lines carrying binary signals from a source to sink. Chip select cable CS is a set $s_0, s_1, \ldots, s_{(q-1)}$ of lines carrying binary signals to the RAM chips. Address cable A is a set $a_1, a_2, \ldots, a_{2p}$ of lines carrying binary signals representing an address to the RAM chips. We use 2p lines instead of p lines for the sake of clarity in understanding the tests

* $\{0,1\}^{(N_K-1)}$ is the notation for the Cartesian product $\underbrace{\{0,1\} \times \cdots \times \{0,1\}}_{(N_K-1) \text{ times}}$.

to be discussed. Stuck-at-0 (1) fault on the lines of D, CS and A are defined in a manner similar to that of row (column) lines.

A RAM unit is a connection of RAM chips, D, CS and A so that every storage element in the RAM chips may be selected for reading or writing independent of previous READ or WRITE. A typical RAM unit is shown in Figure 3 and the function of each unit is described in [9,10,15]. The RAM unit is faulty if any of the above mentioned faults is present in any of the RAM chips, D, CS or A.

DIAGNOSTIC GRAPH OF RAM UNIT

Let $F = \{f_1, f_2, \ldots, f_n\}$ be a set of faults in the RAM unit and $T = \{t_1, t_2, \ldots, t_p\}$ be a set of complete tests for F. A test $t_j \in T$ is a complete test for fault $f_i \in F$ if t_j always fails when f_i alone is present in the RAM unit and always passes for all faults in F absent [14]. In addition, if t_j fails for f_i it locates the fault to a collection of RAM chips or the lines in D, CS or A. The set of tests that are complete for fault f_i is denoted by $t(f_i)$.

A "test t_j is invalid" in the presence of fault f_i in case $t_j \notin t(f_i)$ and t_j might fail. The diagnostic graph of the RAM unit is a directed graph with a node for each fault in F. A directed edge with label t_j is drawn from node f_i to f_K in case $t_j \in t(f_K)$ and t_j is invalid in the presence of fault f_i.

If the diagonstic graph has no cycles, then it can be used by the following algorithm to locate faulty components in the RAM unit when at most one fault from the set F is present.

ALGORITHM 1:

1. Pick a node in the graph with no incoming edge.
2. Output this node.

3. Delete this node and all its outgoing edges from the graph.

4. Repeat steps 1 thru 3 if the graph has at least one node.

5. Halt.

If the output sequence is $(f_{K_1}, f_{K_2}, \ldots, f_{K_n})$, $K_i \neq K_j$, $i \neq j$, then the single fault in the RAM unit is located by performing the sequence of tests $(t(f_{K_1}), t(f_{K_2})$ --- $f(f_{K_n}))$.

The description of the faults in F is shown below. The faults f_4 thru f_8 are treated as a single class and represented by a single node in the diagnostic graph.

CABLE FAULTS:

f_1: D-fault: One or more of the lines in D, but not all, stuck-at-0 or 1.

f_2: CS-fault: One or more of the lines in CS, but not all, stuck-at-1.

f_3: A-fault: Stuck-at fault on one or more, but not all, of the lines of A.

RAM CHIP FAULTS:

f_4: M-fault: States of storage elements of M stuck-at-0 or 1 in a RAM chip.

f_5: DC-fault: The row (column) decoder is faulty in a RAM chip.

f_6: R-fault: The lines of R stuck-at-0 (1) in a RAM chip.

f_7: E-fault: Extended 0(1) fault of storage elements of M in a RAM chip.

f_8: API-fault: Adjacent pattern interference fault in the storage elements of M in a RAM chip.

Many of the faults in current logic circuits realizing a RAM chip can be represented by M, DC, R or E faults [3,8,11,16,17]. The interaction between storage elements in M can be represented by API fault [7,18-20]. Most of the

faults in the logic circuitry for realizing a RAM unit manifest themselves as logical stuck-at-0 or 1 fault on the lines of D, CS or A and this can be represented by cable faults [8,10,17]. So we use cable faults and RAM chip faults in the testing of a RAM unit. Before getting into the details of testing, we describe the organization of a RAM unit shown in Figure 3.

The RAM chips in a RAM unit are arranged in a two dimensional array with q rows and W columns, where q, W >1 and W is a power of 2, so that at any time we may select the W RAM chips in a row for reading or writing. The q rows permit us to expand in capacity as a multiple of N, where N>1 is the number of storage elements in a RAM chip. Since at any time at most one row is selected, the Z cables of RAM chips on a column can be connected to one line in D cable and thus we have W lines in D cable. In addition, the array organization of RAM chips facilitates regularity in connecting CS and A cables to RAM chips. We use this array organization of RAM unit in devising tests for cable and RAM chip faults. We also use the term Word to represent a string of 0's and 1's with length W. We assume that at most (q-1) rows in the array of a RAM unit can have RAM chip faults with a maximum of t RAM chips faulty in a row, where t is given by the following expression:

Let $W = 2^{m_1}$. Let m and b be smallest integers such that $q \leq 2^m$ and

$$m < 1 + \binom{m}{1}1 + \binom{m}{2}1 + \ldots + \binom{m}{b}1 = k_1. \qquad (1)$$

Then t is the greatest integer such that $t \leq \frac{(2^{(m_1-b)}-1)}{2}$. (i.e. t is the error correcting capability of a Reed-Muller (W,k_1) code)[21].

We use the terms location, storage word, contents of location, store, retrieve and address with the usual meaning.

III. DIAGNOSING CABLE FAULTS

We now develop complete tests for D-fault, CS-fault and A-fault in a RAM unit. The primary concern is to devise tests so that it is not invalidated by RAM chip faults and thus eliminating cycles in the diagnostic graph with at least one node corresponding to D-fault, CS-fault or A-fault.

DIAGNOSING D-FAULT:

One easy way to test for D-fault is to write a word 000...0(111...1) in any one location and read the result. If line d_i in D is stuck-at-1, then we will observe 000...1...0 after writing 000...0...0. If line d_i is stuck-at-0, then we will observe 111...0...1 after writing 111...1...1. But there is a flaw in this test. We note from the array organization of the RAM unit that the Z cables of all RAM chips in the i-th column is connected to line d_i in D. If any one of the Z cables on the i-th column is stuck-at-0 (1), then by the above experiment, we might conclude incorrectly that line d_i in D is stuck-at-0 (1).

However, this problem can be overcome by writing a distinct word (complement of the word) in any one location of each of the q rows of RAM chips in a RAM unit and reading the results. If line d_i in D is stuck-at-1, then we will observe ---...1...--- after writing each of the q words and the q complements of the words. If line d_i in D is stuck-at-0, then we will observe ---...0...--- after writing each of the q words and its q complements. Since i is arbitrary, the above scheme detects and locates all single line and multiple lines stuck-at-0 or 1 and thus diagnosing D-fault. The test length is 2 q. Test generation is possible by using increment and decrement operations and it is described in the following experiment:

Let q be the number of rows of RAM chips, $q \leq 2^m$ where m is the smallest positive integer, and (2p+m) be the total number of lines in A and CS cables of a RAM unit. Let W be the length of each word.

EXPT 1:

1. Fix a 2p-tuple and form an address by combining an m-tuple of all zeroes followed by the 2p-tuple.
2. Store a word W0 with all bits zero in the addressed location.
3. Increment the m tuple by 1 to form a new address. Increment W0 by 1.
4. Write W0 in the addressed location.
5. Repeat steps 3 and 4 until the value of the m tuple is (q-1).
6. Retrieve the contents of storage words in the sequence in which they are stored and form a matrix M_0 of q rows and W columns.
7. Repeat steps 1 thru 5 after making two changes

   (i) W0 has all bits 1 in step 2.

   (ii) Decrement W0 by 1 in step 3.

   Retrieve the contents of storage words in the sequence in which they are stored and form a matrix M_1, of q rows and W columns.

Theorem 1:

If the i-th line in D is stuck-at-0 (1) then the i-th columns of matrices M_0 and M_1, formed in EXPT 1, have all elements 0(1), where $1 \leq i \leq W$.

Proof:

In step 4 of EXPT 1 a word W0 is stored in the row of RAM chips with label (j-1), retrieved and entered in the j-th row of matrix M_0. In step 7 of EXPT 1 the complement of W0 is stored in the row of RAM chips with label (j-1),

retrieved and entered in the j-th row of matrix M_1.

If the i-th line in D is stuck-at-0 (1) then the i-th element in the j-th row of M_0 and M_1 are 0(1). Each row in M_0 and M_1 corresponds to a row of RAM chips and there is one line for each column of RAM chips. Then the i-th line in D stuck-at-0 (1) implies all elements in the i-th column of M_0 and M_1 are 0(1).

Q.E.D.

The test for D-fault described in EXPT 1 is said to have failed if any one of the columns of M_0 and M_1 is all 0(1) and it passes otherwise.

EXPT 1 is illustrated by the following example.

EXAMPLE 1: Let us consider a RAM unit with q = 8, W = 8, and 2p = 10. Let the 2p tuple be 00000 00000. The matrices M_0 and M_1 are shown in Table 1.

TABLE 1

ADDRESS VALUE m-tuple	ADDRESS VALUE 2p-tuple	DATA STORED $d_1d_2d_3d_4d_5d_6d_7d_8$	DATA RETRIEVED (M_0) $d_1d_2d_3d_4d_5d_6d_7d_8$	DATA STORED $d_1d_2d_3d_4d_5d_6d_7d_8$	DATA RETREIVED (M_1) $d_1d_2d_3d_4d_5d_6d_7d_8$
0 0 0	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 1 0 1 0	1 1 1 1 1 1 1 1	0 1 1 1 1 1 1 1
0 0 1	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1	0 0 0 0 1 0 1 1	1 1 1 1 1 1 1 0	0 1 1 1 1 1 1 0
0 1 0	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0	0 0 0 0 1 0 1 0	1 1 1 1 1 1 0 1	0 1 1 1 1 1 1 1
0 1 1	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 1	0 0 0 0 1 0 1 1	1 1 1 1 1 1 0 0	0 1 1 1 1 1 1 0
1 0 0	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 1 0 0	0 0 0 0 1 1 1 0	1 1 1 1 1 0 1 1	0 1 1 1 1 0 1 1
1 0 1	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 1 0 1	0 0 0 0 1 1 1 1	1 1 1 1 1 0 1 0	0 1 1 1 1 0 1 0
1 1 0	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 1 1 0	0 0 0 0 1 1 1 0	1 1 1 1 1 0 0 1	0 1 1 1 1 **0** 1 1
1 1 1	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 1 1 1	0 0 0 0 1 1 1 1	1 1 1 1 1 0 0 0	0 1 1 1 1 0 1 0

Lines d_5 and d_7 stuck-at-1 and line d_1 stuck-at-0.

DIAGNOSING CS-FAULT:

The presence of one or more of the lines in CS but not all stuck-at-1 has been defined to be CS-fault. The physical significance of this logical fault is that if line s_i in CS is stuck-at-1 then we cannot select any of the locations in the row of RAM chips corresponding to line s_i for reading or writing. That is, we always read, say 111...1...1 from every location in the row of RAM chips corresponding to line s_i. An easy way to test for CS-fault is to write a distinct word in any one location in each of the q rows of RAM chips and read the result. If line s_i in CS is stuck-at-1, then we will observe 111...1...1 after writing a distinct word. Since i is arbitrary and we use q distinct words, the above test detects and locates CS-fault. There is one problem in this test when t or less RAM chips in a row are faulty. This is solved by utilizing a code with t error correcting capability to generate code words for the q distinct words and then using these code words in the tests.

The Reed-Muller Code is selected for two reasons: it is easy to encode and decode and code length is a power of 2. With word length $W(=2^m1)$ usually ≥ 8 and the number of rows of RAM chips in a RAM unit limited by physical dimension, in most cases we have $t \geq 1$. It is possible to use an extended Hamming Code when W=8 and p=5 or 6.

The above test with length q can be generated, except for the encoding and decoding part, by using increment operation alone. This is shown in the following experiment. A Reed-Muller (W,k_1) Code is used in this experiement.

EXPT 2:

1. Fix a 2p-tuple. Form an address by combining an m-tuple of all zeroes followed by the 2p-tuple. Let K be a binary string of length k_1

(defined by (1) in Section II) with initially all bits zero.

2. Form a code word with the bits of K as information bits and store the code word in the addressed location.
3. Increment the value of the m-tuple in the address and the value of K by one.
4. Repeat 2) and 3) when the value of the m-tuple is less than (q-1).
5. Retrieve the contents of storage words in the sequence in which they are stored, decode them and form a matrix S of q rows and k_1 columns.

Theorem 2:

If line $s_{(u-1)}$ in CS is stuck-at-1 then the binary value of the u-th row of matrix S, formed in EXPT 2, is not (u-1) when at most t RAM chips are faulty in a row of a RAM unit, where $1 \leq u \leq q$.

Proof:

During the u-th iteration of EXPT 2 the value of K is (u-1), $1 \leq u \leq q$. In step 2 of EXPT 2 the code word corresponding to K is stored in a location in the (u-1)-st row of RAM chips by selecting $s_{(u-1)}$ line in CS. In step 5 of EXPT 2, the retrieved word from the (u-1)-st row of RAM chips in decoded and entered in the u-th row of the matrix S.

If line $s_{(u-1)}$ is stuck-at-1 then the retrieved code word is 111...1...1, and the decoded word is 100...0...0 with length k_1 and it is the u-th row of S. By definition $m < k_1$ and $q \leq 2^m$ and so the binary value of 1000...0...0 is not equal to (u-1), $1 \leq u \leq q$

Q.E.D.

We say that the test for CS-fault described in EXPT 2 has failed if the

binary value of row u in S is not equal to (u-1), $1 \leq u \leq q$ and it passes otherwise.

EXPT 2 is illustrated by the following example.

EXAMPLE 2:

Let us consider a RAM unit with q = 4, W = 8 and 2p = 12. Let the 2p tuple be 000000000000.

A Reed-Muller (8,4) Code is used. This Code has single error correcting capability. The matrix S of EXPT 2 is shown in Table 2.

TABLE 2

Lines in CS	Address		Information Bits	Stored Code Words	Retrieved Words	Matrix S
	m-tuple	2p-tuple				
s_0	00000000000000		0000	00000000	11111111	1000
s_1	01000000000000		0001	01010101	00010101	0001
s_2	10000000000000		0010	00110011	00110111	0010
s_3	11000000000000		0011	01100110	11111111	1000

Since the values of rows in S corresponding to s_3 and s_0 are not 3 and 0 respectively the two lines are stuck-at-1.

DIAGNOSING A -FAULT:

A stuck-at-fault on a line in A cable maps at least two addresses e.g. 000...1...0 and 000...0...0 to one storage location. We notice from the array organization of the RAM unit that the lines of the A cable are connected to every RAM chip in the array and at any one time at most one row of RAM chips is

selected for reading or writing. So we need to test for A-fault in each row of RAM chips and in each row it is necessary to test the 2p lines in the A cable. An easy way to test for A-fault is to select a row of RAM chips, then write a distinct word in each of the locations $\underset{123...i...2p}{000...1...0}$, $1 \leq i \leq 2p$ and finally the word 000...0 in the location 000...0...0, and then read the result. If line i is stuck-at either 0 or 1, then we will observe 000...0 after stroing a distinct word in the location $\underset{12\quad i\quad 2p}{000...1...0}$ and the word 000...0 in the location 000...0...0. As in the case of CS-faults, we use a t error correcting code and generate code words for the (2p+1) distinct words and use these code words in the test to take care of the situation when t or less RAM chips in a row are faulty.

A test length of (2p+1)q is sufficient to locate A-fault in a RAM unit. Except for the encoding and decoding, the test for A-fault can be generated by using left shift and increment operations. This is shown in the following experiment. A Reed-Muller (W,k) code is used, where $m \leq 1 + \binom{m}{1} + \ldots + \binom{m}{b} = k$ in equation (1).

EXPT 3:

1. Let T be a register with initial content $\underset{123\quad 2p}{(000...01)}$. Fix a m-tuple and let its binary value be ℓ. Form an address by combining the m-tuple and the content of T. Let U be a binary string of length k with initial value 1. Let u = 1.
2. Form the code word corresponding to U.
3. Store the code word in the storage word corresponding to the address.
4. Increment U and u by one, left shift T by one position and form a new address. Repeat steps 2 and 3 when the value of u is less than 2p.

5. Store the null code word (all elements in the code word zero) in the address (------00000---0---0).
m-tuple123 2p

6. Retrieve the contents of the (2p+1) storage words following the same sequence in which they are stored.

7. Decode the retrieved words and form a matrix R_ℓ of (2p+1) rows and k columns.

Theorem 3:

If the i-th line in A cable, corresponding to the ℓ-th row of RAM chips is stuck at either 0 or 1 then the binary value of the i-th row of the matrix R_ℓ, formed by EXPT 3, is not i when at most t RAM chips are faulty in the row where $1 \leq i \leq 2p$.

Proof:

The two addresses (000---010---0) and (000---000---0) map to one storage
123 i 2p 123 i 2p
word if the i-th address line is stuck at either 0 or 1, say 0.

EXPT 3 assigns the value i for U. The code word is formed and stored in the location corresponding to the address (------ 000---00---0). In step 5
m-tuple 123 2p
of EXPT 3 the null word is stored in the location corresponding to the address (--------000---00---0). Thus the i-th row of matrix R_ℓ is zero and so the
m-tuple 123 i 2p
binary value of i-th row is not equal to i.

Q.E.D.

EXPT 3 is performed for each row of RAM chips in the RAM unit. If for some i, $1 \leq i \leq 2p$, the binary value of the i-th row of the matrix R_ℓ is not equal to i for every ℓ, $0 \leq \ell \leq (q-1)$, then we say that the test for A-fault fails. It passes otherwise.

EXPT 3 is illustrated by the following example.

EXAMPLE 3:

Let us consider a RAM unit with q = 8, W = 8, 2p = 12. Let the m-tuple be (000).

An extended hamming (8,4) code with single error correcting capability is utilized in this example. The matrix R0 is shown in Table 3.

TABLE 3

ADDRESS	INFORMATION BITS	STORED CODE WORD	RETRIEVED WORD	MATRIX R_0
000000000000001	0001	0 0 0 1 1 0 1 1	0 0 0 0 1 0 1 1	0 0 0 1
000000000000010	0010	0 0 1 0 0 1 1 1	0 0 0 0 0 0 0 0	0 0 0 0
000000000000100	0011	0 0 1 1 1 1 0 0	0 0 1 1 1 1 0 0	0 0 1 1
000000000001000	0100	0 1 0 0 1 1 0 1	0 0 0 0 0 0 0 0	0 0 0 0
000000000010000	0101	0 1 0 1 0 1 1 0	0 0 0 1 0 1 1 0	0 1 0 1
000000000100000	0110	0 1 1 0 1 0 1 0	0 1 1 0 1 0 1 0	0 1 1 0
000000001000000	0111	0 1 1 1 0 0 0 1	0 1 1 0 0 0 0 1	0 1 1 1
000000010000000	1000	1 0 0 0 1 1 1 0	0 0 0 0 0 0 0 0	0 0 0 0
000000100000000	1001	1 0 0 1 0 1 0 1	1 0 0 1 0 1 0 1	1 0 0 1
000001000000000	1010	1 0 1 0 1 0 0 1	1 0 1 0 1 0 0 1	1 0 1 0
000010000000000	1011	1 0 1 1 0 0 1 0	0 0 1 1 0 0 1 0	1 0 1 1
000100000000000	1100	1 1 0 0 0 0 1 1	0 0 0 0 0 0 0 0	0 0 0 0
000000000000000	0000	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0

Lines a_2, a_4, a_8 and a_{12}, corresponding to the row of RAM chips selected by s_o, are stuck at either 0 or 1, since the matrix R_0 has the binary value 0 for rows 2,4,8 and 12.

IV. DETECTING RAM CHIP FAULTS

The objective in this section is to devise complete tests for M-fault, DC-fault, R-fault, E-fault and API-fault, so that length of tests is minimal and test generation is not too complicated. The concern is to detect the above mentioned faults in RAM chips and locating the RAM chips containing it.

The pin limitation of RAM chips results in the use of symmetric array (SA) organization for the storage elements in M since the lines in A can then be multiplexed, their number cut down to p and consequently the number of pins in RAM chip reduced. This SA organization of M is fully utilized in designing tests. By the definition of DC-fault, any one of the 2^p inputs to the row (column) decoder can be mapped to a collection of storage elements, possibly empty, in M. So the minimum number of write and read for testing DC-fault with SA organization of M is 2^p N, where $N = 2^{2p}$. Extended 0 and 1 faults each require 2^pN write and read with SA organization of M and so the minimum number of write and read for testing E-fault is $2(2^pN)$. The detection of adjacent pattern interference fault in a storage element of M requires a test with 32 ($=2^5$) writes and reads into the storage element. Then the minimum number of write and read for testing API-fault is 32 N. Three tests with a total length of $(3(2^p) + 32)$ N, which is minimal with respect to SA organization, are designed to detect RAM chip faults.

The array organization of RAM unit and SA organization of M in RAM chips is used in the following tests. The term WRITE A PATTERN is used to describe writing a given sequence of words in all the locations of a row of RAM chips. VERIFY A PATTERN reads the contents of locations following the sequence used in WRITE A PATTERN, performs an exclusive-or of each word with

the stored word, and if the resulting word has f-th bit 1 then the RAM chip corresponding to f-th bit is reported as faulty, where $1 \leq f \leq W$. The address of a location is denoted by (i,j) where i (j) is the binary value of the p-tuple to row (column) decoder and $0 \leq i,j \leq (2^p-1)$.

We say that a test has failed in case the VERIFY A PATTERN step in the test reports a fault, and it passes otherwise.

TEST FOR DC-FAULT:

An easy way to test for DC-fault is to write a word, say 111...1 in the location (i,i), then write the word 000...0 at all other locations in a row of RAM chips, read the result and repeat with a new value for i, until all values of i are covered. If the row (column) decoder is faulty then we will observe a word not equal to 111...1 in location (i,i) after storing 111...1 or a word not equal to 000...0 in at least one of the remaining locations after storing 000...0. Thus, the test consists of WRITE i-th PATTERN, VERIFY i-th PATTERN, and repeat until all values of i are covered.

WRITE i-th PATTERN is performed by using a register of length 2p containing address, and a counter for tracking the total number of locations that have been selected for writing. This is shown in the flow chart of Figure 4, where W1 is the word with all bits 1, W0 is the word with all bits 0 and the overflow of rigister ADDR is ignored. The flow chart of Figure 4 can also be used for VERIFY i-th PATTERN after changing "WRITE W1(W0) in ADDR" to "READ contents of ADDR, and EXCLUSIVE-OR with W1(W0)". If the result of EXCLUSIVE OR with W1(W0) is not zero then a fault has been detected and the RAM chips corresponding to the 1 bits are faulty, e.g. 00...1.00 indicates that the
12 f W
RAM chip in the f-th column of the row of RAM chips selected for testing is

faulty.

This test is performed on each row of RAM chips in the RAM unit to detect and locate all RAM chips with DC-fault.

The total number of write and read required in this test for a row of RAM chips is 2^p N, which is also the minimum number. The steps involved in testing for DC-fault is listed in the following experiment.

EXPT 4:

Let the x-th row of RAM chips in the RAM unit be selected for testing.

1. Let i = 0.
2. WRITE i-th PATTERN (Figure 4).
3. VERIFY i-th PATTERN.
4. Increment i by one. Repeat steps 2 and 3 when the value of i is less than (2^p-1).
5. Halt.

We now show that the test for DC-fault is also a test for R-fault. The physical significance of a stuck-at-0 fault on row (column) line is that the row (column) of storage elements in M corresponding to that line is always selected while writing or reading in any storage element of M. The significance of stuck-at-1 fault is similar to that described for lines in CS cable stuck-at-1.

By the definition of R cable, a row (column) line exists for each distinct p-tuple i(j) in the address (i,j) where $0 \leq i,j \leq (2^p-1)$.

Theorem 4:

In the R cable of a RAM chip, the row and column lines stuck-at-0 and stuck-at-1 are detected by EXPT 4.

Proof:

If the ℓ_1-th row line is stuck at 0, then the storage word accessed by every address (ℓ_3,ℓ_1), $\ell_3 = 0,1,2,---(2^P-1)$ also accesses the storage word corresponding to the address (ℓ_1,ℓ_1) where $0 \leq \ell_1 \leq (2^P-1)$. The word W1 with all bits one is stored in the storage word corresponding to address (ℓ_1,ℓ_1) in step 2 of EXPT 4. The storing of the word W0 with all bits zero corresponding to all other address values (k,s) also stores W0 in the storage word corresponding to the address (ℓ_1,ℓ_1). Stuck at 0 faults on the row lines are detected in step 3 of EXPT 4.

If the ℓ_2-th row line is stuck at 1 then every word retrieved from the memory corresponding to the addresses (ℓ_2,ℓ_3), $\ell_3 = 0,1,2,3,---(2^P-1)$ is the same at all times where $0 \leq \ell_2 \leq (2^P-1)$. A word W1 is stored corresponding to the address (ℓ_2,ℓ_2) and W0 is stored corresponding the addresses (ℓ_2,ℓ_3), $\ell_3 \neq \ell_2$, $\ell_3 = 0,1,2---(2^P-1)$ by the experiment. Stuck at 1 faults in the row lines are detected by step 3 in EXPT 4. Using a similar argument, column lines stuck at 0 and stuck at 1 are detected.

Q.E.D.

EXPT 4 is illustrated by the following example

EXAMPLE 4:

Let us consider RAM chips with 4 x 4 storage array for M, Let W = 4.

EXPT 4:

Iteration - I.

Stored Word

#1	#2	#3	#4
1 0 0 0	1 0 0 0	1 0 0 0	1 0 0 0
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Retrieved Word

0 0 0 0	1 0 0 0	0 0 0 0	1 0 0 0
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 0 0 0	1 1 1 1	1 1 1 1	0 0 0 0

Iteration II

Stored Word

0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 1 0 0	0 1 0 0	0 1 0 0	0 1 0 0
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Retrieved Word

0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0	0 0 0 0	0 1 0 0
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 0 0 0	1 1 1 1	1 1 1 1	0 0 0 0

Iteration III

Stored Word

0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 0 1 0	0 0 1 0	0 0 1 0	0 0 1 0
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Retrieved Word

0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 0 1 0	0 0 1 0	0 0 1 0	0 0 1 0
0 0 0 0	1 1 1 1	1 1 1 1	0 0 0 0

Iteration IV

Stored Word

0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1

Retrieved Word

0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 0 0 1	1 1 1 1	1 1 1 1	0 0 0 1

RAM chips #1, #2 and #3 are faulty.

TEST FOR E-FAULT:

One simple way to test for E-fault is to write the word 111...1 (000...0) in the locations (i, [i+j]), $0 \leq i \leq (2^P-1)$, the word 000...0 (111...1) at the remaining locations in a row of RAM chips, read the result and repeat for $j = 0, 1, 2, \ldots (2^P-1)$, where [ ] means modulo 2^P. If the storage element $C_{(i, [i+j])}$ in a RAM chip has extended 0(1) fault, then we will observe a word not equal to 111...1(000...0) in location (i,[i+j]). Thus by using $2(2^P N)$ write and read we can detect E-fault in a row of RAM chips. It is evident that the above test also detects M-fault.

The test generation for E-fault is an extension of the scheme described for generating the test for DC-fault. The required extension is an extra counter. WRITE j-th PATTERN for E-fault is performed by the flow chart of Figure 5. This flow chart is obtained by adding the extra counter to the flow chart of WRITE i-th PATTERN for DC-fault and making some minor changes. The VERIFY j-th PATTERN for E-fault is performed by using the flow chart of WRITE j-th PATTERN

for E-fault after changing "WRITE W1(W0) in ADDR" to "READ contents of ADDR, and EXCLUSIVE-OR with W1(W0)".

The steps involved in testing for E-fault is listed in the following experiment.

EXPT 5:

Let the x-th row of RAM chips in the RAM unit be selected for testing.

1. Let j = 0.
2. WRITE j-th PATTERN (Figure 5)
3. VERIFY j-th PATTERN
4. Increment j by one. Repeat steps 2 and 3 when the value of j is less than (2^p-1).
5. Change W0 to W1 and W1 to W0. Repeat steps 1 thru 4.
6. Halt.

This test is performed on each row of RAM chips in the RAM unit to detect and locate all RAM chips with E-fault.

The WRITE j-th PATTERN is illustrated by the following example.

EXAMPLE 5:

Let us consider RAM chips with 4 x 4 storage array for M and let W = 2.

The WRITE j-th PATTERN generated by EXPT 6 is shown in Table 4.

TABLE 4

			Complements	
j = 0	1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1	1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1	0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0	0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0
j = 1	0 1 0 0 0 0 1 0 0 0 0 1 1 0 0 0	0 1 0 0 0 0 1 0 0 0 0 1 1 0 0 0	1 0 1 1 1 1 0 1 1 1 1 0 0 1 1 1	1 0 1 1 1 1 0 1 1 1 1 0 0 1 1 1
j = 2	0 0 1 0 0 0 0 1 1 0 0 0 0 1 0 0	0 0 1 0 0 0 0 1 1 0 0 0 0 1 0 0	1 1 0 1 1 1 1 0 0 1 1 1 1 0 1 1	1 1 0 1 1 1 1 0 0 1 1 1 1 0 1 1
j = 3	0 0 0 1 1 0 0 0 0 1 0 0 0 0 1 0	0 0 0 1 1 0 0 0 0 1 0 0 0 0 1 0	1 1 1 0 0 1 1 1 1 0 1 1 1 1 0 1	1 1 1 0 0 1 1 1 1 0 1 1 1 1 0 1

TEST FOR API-FAULT:

The problem of designing a test with 32 N write and read to detect API-fault is reduced to the problem of finding 32 x 9 write and read to detect adjacent pattern interference fault in each of the storage elements of a 3 x 3 block, shown in Figure 6. Let the term assignment be used to represent writing into the 9 storage elements of a block, e.g. $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$ is the assignment corresponding to $\begin{bmatrix} W_A \bar{W}_B \bar{W}_C \\ \bar{W}_D \bar{W}_E \bar{W}_F \\ W_G \bar{W}_H \bar{W}_I \end{bmatrix}$. Thus there are 32 assignments to a block. If we can find the 32 assignments to a block, then an easy way to test for API-fault is to write the ℓ-th assignment in each block of M of a row of RAM chips, read the result and repeat for $\ell = 1,2,\ldots 32$. If a storage element C_K of M in a

RAM chip has adjacent pattern interference fault then we will observe a word not equal to 111...1 (000...0) in the location corresponding to C_K.

The set of 32 assignments to a block is shown below. It can be verified that the set of 32 assignments provide every element of $\{0,1\}^5$ to each row of Table 5 and so in 32 x 9 write and read to the storage elements of a block the adjacent pattern interference fault in each of the storage elements in the block can be detected.

(1)	100 000 100	(2)	100 000 011	(3)	011 000 100	(4)	011 000 011												
(5)	101 001 101	(6)	000 100 000	(7)	001 101 001	(8)	101 001 010	(9)	010 001 101	(10)	010 001 010								
(11)	111 100 000	(12)	110 101 001	(13)	000 100 111	(14)	111 100 111	(15)	001 101 110	(16)	110 101 110								
(17)	001 010 001	(18)	001 010 110	(19)	110 010 001	(20)	110 010 110												
(21)	000 011 000	(22)	101 110 101	(23)	100 111 100	(24)	000 011 111	(25)	111 011 000	(26)	111 011 111								
(27)	010 110 101	(28)	011 111 100	(29)	101 110 010	(30)	010 110 010	(31)	100 111 011	(32)	011 111 011								

The test generation for API-fault is by using a Controlled Register to write the ℓ-th assignment in each block of M of a row of RAM chips. The controlled register consists of a counter with (2p+1) bits, and a rigister with 9 bits as shown in Figure 7. The most significant bit of the counter is called the overflow bit (OVF) and the least significant bit of the register is called

the output bit ($\hat{0}$). Initial value of the Counter is zero and that of the register is some assignment of a block. Trigger (TRG) acts as input and output is $\hat{0}$. Two operations can be performed on the register. One operation right rotates the content of the three least significant bits without disturbing the rest. The second operation right rotates the content of the entire reigster.

On receiving TRG the Controlled Register performs the following:

(i) Increments the counter by one

(ii) If OVF = 0, then the three least significant bits of the register are right rotated once

If OVF = 1, then OVF is reset to 0 and the entire register right rotated 3 times.

(iii) Outputs $\hat{0}$, where $\hat{0}$ is either 0 or 1.

The flow chart of Figure 8 describes WRITE ℓ-th PATTERN for API-fault. If we change "WRITE $W\hat{0}$ in ADDR" to "READ contents of ADDR and EXCLUSIVE-OR with $W\hat{0}$" in the above mentioned flow chart then it could be used for VERIFY ℓ-th PATTERN for API-fault.

The test for API-fault is performed on each row of RAM chips in the RAM unit to detect and locate all RAM chips with API-fault. The steps involved in the test are listed in the following experiment:

EXPT 6:

Let the x-th row of RAM chips in the RAM unit be selected for testing.

1. Let $\ell = 1$.
2. WRITE ℓ-th PATTERN (Figure 8).
3. VERIFY ℓ-th PATTERN.
4. Increment ℓ by one. Repeat steps 2 and 3 when ℓ is less than 32.
5. Halt.

V. LOCATING FAULTY COMPONENTS

The lines with CABLE FAULTS and RAM chips with RAM CHIP FAULTS can be located if we can demonstrate that the diagnostic graph has no cycles. We note that the tests described all perform select location, write, and read. During read, the lines of D cable are used. Consequently D-fault invalidates all tests. The presence of CS-fault invalidates the tests for RAM CHIP FAULTS but it does not invalidate the test for D-fault since, by the definition of CS-fault not all lines in CS cable are faulty and so the fail condition of the test for D-fault can never be satisfied. Using the same argument it is seen that CS-fault does not invalidate the test for A-fault. It is clear that A-fault invalidates the tests for RAM CHIP FAULTS. Since the test for D-fault and CS-fault writes in any one location of a row of RAM chips, the presence of A-fault does not invalidate the test for D-fault or CS-fault.

At most (q-1) rows of RAM chips in a RAM unit can be faulty and so RAM CHIP FAULTS cannot invalidate the test for D-fault or A-fault. Since a maximum of t RAM chips in a row of a RAM unit can be faulty, RAM CHIP FAULTS cannot invalidate the test for CS-fault. The diagnostic graph is drawn in Figure 9 and there are no cycles in the graph. By using Algorithm 1, there are two fault location sequences (t(D), t(CS), t(A), t(RAM)), and (t(D), t(A), t(CS), t(RAM)).

The faulty components in a RAM unit are located by using the flow chart of Figure 10, where the experiments corresponding to t(D), t(CS), t(A), t(RAM) are performed in that order. The experiments corresponding to t(RAM) can be performed in any order.

VI. CONCLUSION

Faults in a RAM unit can be modeled using CABLE FAULTS and RAM CHIP FAULTS. It is shown that with array organization for the RAM unit and SA organization for RAM chips these faults can be detected and located to lines in the various cables or RAM chips in the RAM unit by performing a sequence of tests with a total length of $\cong (32 + 3(2^p))$ Nq. The required sequence of tests is obtained from the diagnostic graph drawn by using the "test tj is invalid" relation on the set of CABLE FAULTS and RAM CHIP FAULTS.

The test for the faults are of minimal length. Test generation for CABLE FAULTS require very little computation except for the encoding and decoding. The test generation for RAM CHIP FAULTS are based on the scheme "WRITE i-th PATTERN", which uses just two operations: increment and compare. The efficiency of test generation for the API-fault in the set of RAM CHIP FAULTS is boosted by using the "Controlled Register".

The tests in the fault location scheme described in here have been programmed on a commercially available 8-bit LSI processor (microprocessor) with 2 μ sec instruction cycle time to diagnose RAM units. The estimated time for diagnosing a RAM unit with W = 8, q = 4 and N = 4K (i.e. a RAM unit with a storage capacity of 16 K bytes) is under 4 minutes. The amount of storage needed for the programs is around 1.5K bytes. The low execution time and storage requirements makes the tests furthermore attractive in diagnosing the memory of microprocessor systems [22].

REFERENCES

1. Webb, C. and Richardson, W., "Pattern sensitivity in a 4096 Bit RAM", Digest of Papers, 1974 Semiconductor test symposium, pp. 33-52.

2. Fischer, James, "Test problems and solutions for 4k RAMS", Digest of Papers, 1974 Semiconductor test symposium, pp. 53-71.

3. Huston, R. E., "Testing Semiconductor memories", Digest of Papers, 1973 symposium on Semiconductor Memory Testing, Oct., 1973 Cherry Hill, New Jersey. Sponsored by IEEE, pp. 27-62.

4. Webb, C., and Richardson, B., "Pattern Sensitivity in a 4096 bit RAM", Digest of Papers, 1974, Semiconductor test symposium, Nov. 1974, Cherry Hill, New Jersey. Sponsored by IEEE, pp. 33-52.

5. Colbourne, D. E., Coverley, G. P. and Behera, S. J., "Reliability of MOS LSI Circuits", Proceedings of IEEE, Vol. 62, No. 2, Feb. 1974, pp. 244-259.

6. Brown, J. R., "Pattern sensitivity in MOS memories", Digest of Papers, 1972 Symposium on Semiconductor Memory Testing, Cherry Hill, New Jersey, Oct. 1972, pp. 33-46.

7. Hayes, J. P., "Detection of Pattern-Sensitive Faults in Random-Access Memories", IEEE Transactions on computers, vol. C-24, No. 2, Feb. 1975, pp. 150-157.

8. Friedman, A. D., Menon, P. R., "Fault Detection in Digital Circuits," Prentice Hall, New Jersey, 1971.

9. The INTEL Memory Design Handbook, Aug. 1973.

10. Luecke, G., Mize, J. P., and Carr, W. C., "Semiconductor memory design and application", McGraw Hill, New York, 1973.

11. Kuo, C., Kitagawn, N., Ward, E., and Drayer, P., "Sense Amplifier design is key to 1-transistor cell in 4k bit RAM", Electronics, Sept. 13, 1973, pp. 116-121.

12. Abbott, R. A., Regitz, W. M., Karp, J. A., "A 4k MOS Dynamic Random-Access Memory", IEEE Journal of solid state circuits, Vol. SC-8, No. 5, Oct. 1973, pp. 292-298.

13. Hoffman, W. K. and Kalter, H. L., "An 8k b Random Access Memory Chip using the one device FET cell", IEEE Journal of SSC, Vol. SC-8, No. 5, Oct. 1973, pp. 298-304.

14. Russell, J. D., and Kime, C. R., "System Fault Diagnosis: Closure and Diagnosability with Repair", IEEE Transactions on Computers, Vol. C-24, No-11, Nov. 1975, pp. 1078-1089.

15. Vadasz, L.L., Chua, H.T., and Grove, A.S., "Semiconductor Random Access memories", IEEE Spectrum, Vol. 8, No. 5, May 1971, pp. 40-48.

16. SEMICONDUCTOR DATA LIBRARY/MOS MEMORIES, Motorola Semiconductor products Inc., Vol. 7, Series A, 1975.

17. Foss, R.C., Harlands R., "Peripheral Circuits for one-transistor cell MOS RAM'S", IEEE Journal of Solid-State Circuits, Vol. SC-10, No. 5, Oct. 1975, pp. 255-261.

18. Hnatek, E.R., "4-Kilobit Memories Present a Challenge to Testing", Computer Design, May 1975, Vol. 14, No. 5, pp. 117-125.

19. Muehldorf, E.I., "A Quality Measure for LSI Components", IEEE Journal of Solid-State circuits, Vol. SC-9, Oct. 1974, pp. 291-297.

20. Srini, V.P., "Iterative Network Model for the Storage Array in RAM chips", Submitted to IEEE Transaction on Computers.

21. Peterson, W.W. and Weldon, Jr., E.J., Error Correcting Codes, Second Edition, MIT Press, Cambridge, Massachusetts, 1972.

22. Srini, V.P., "Fault Diagnosis of Microprocessor Systems", Computer Magazine Repository Paper, IEEE Computer Society, Dec. 1975.

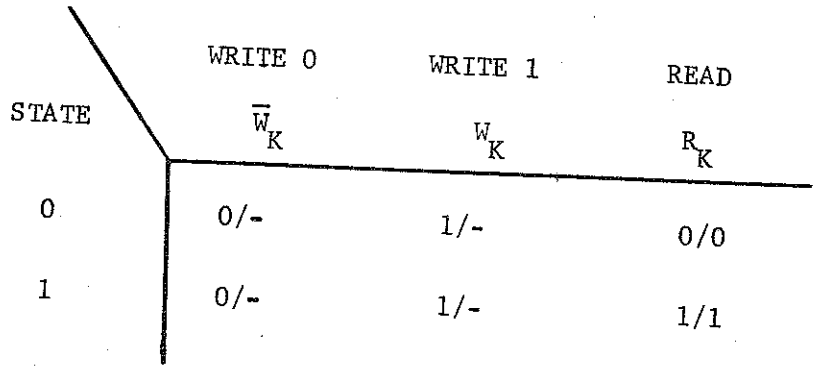


Figure 1. State table

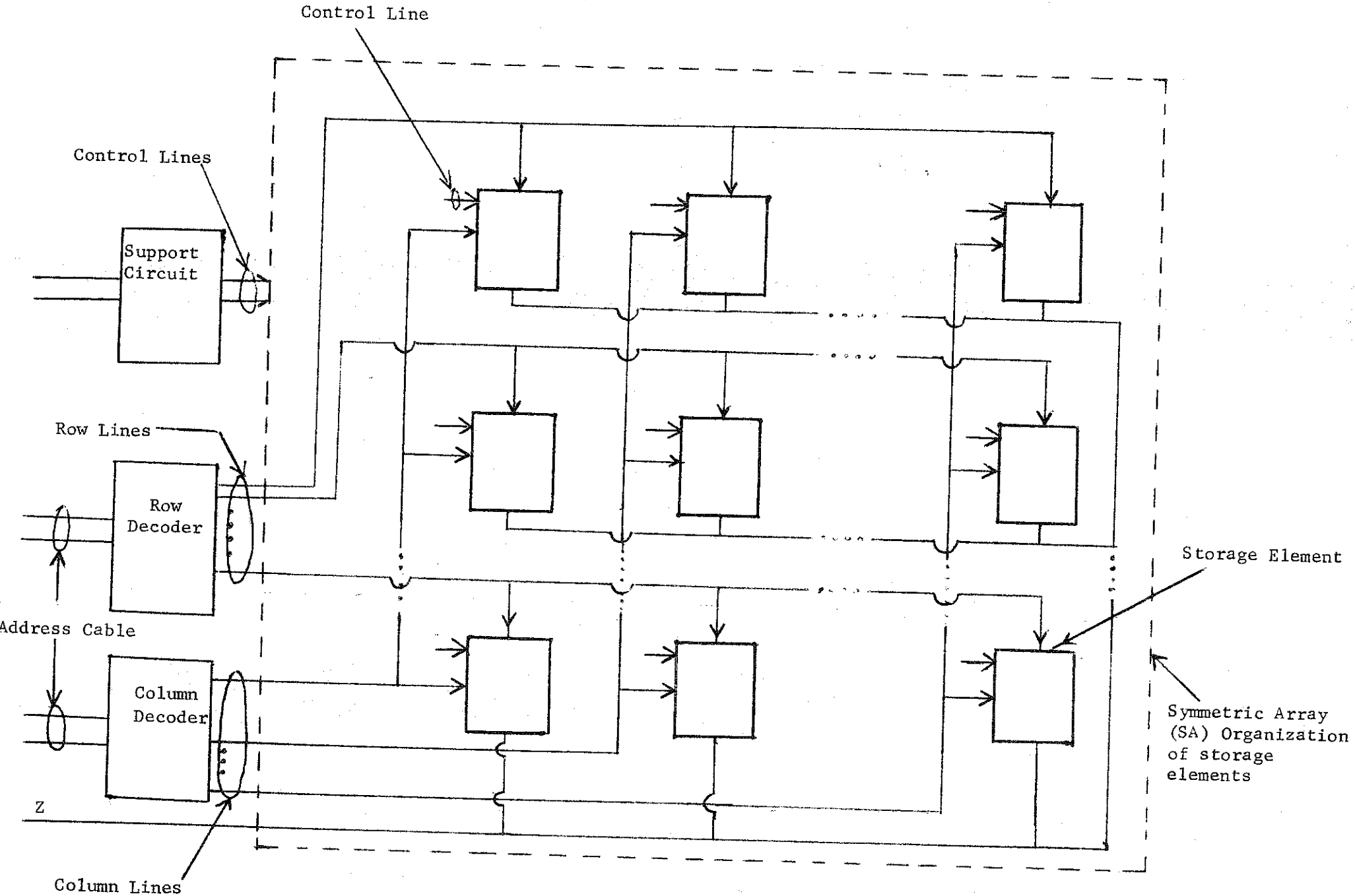


Figure 2. RAM Chip

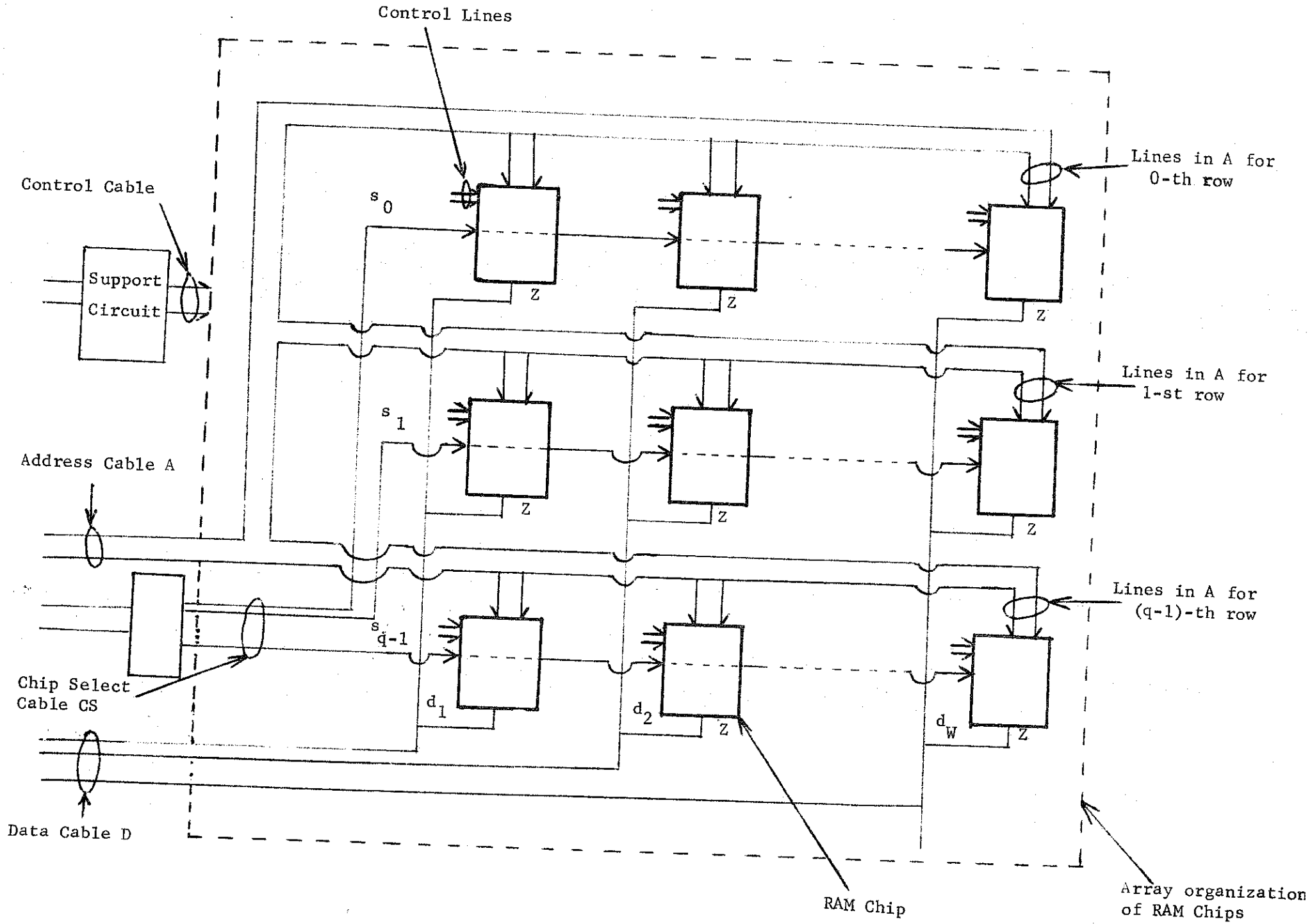


Figure 3. RAM Unit

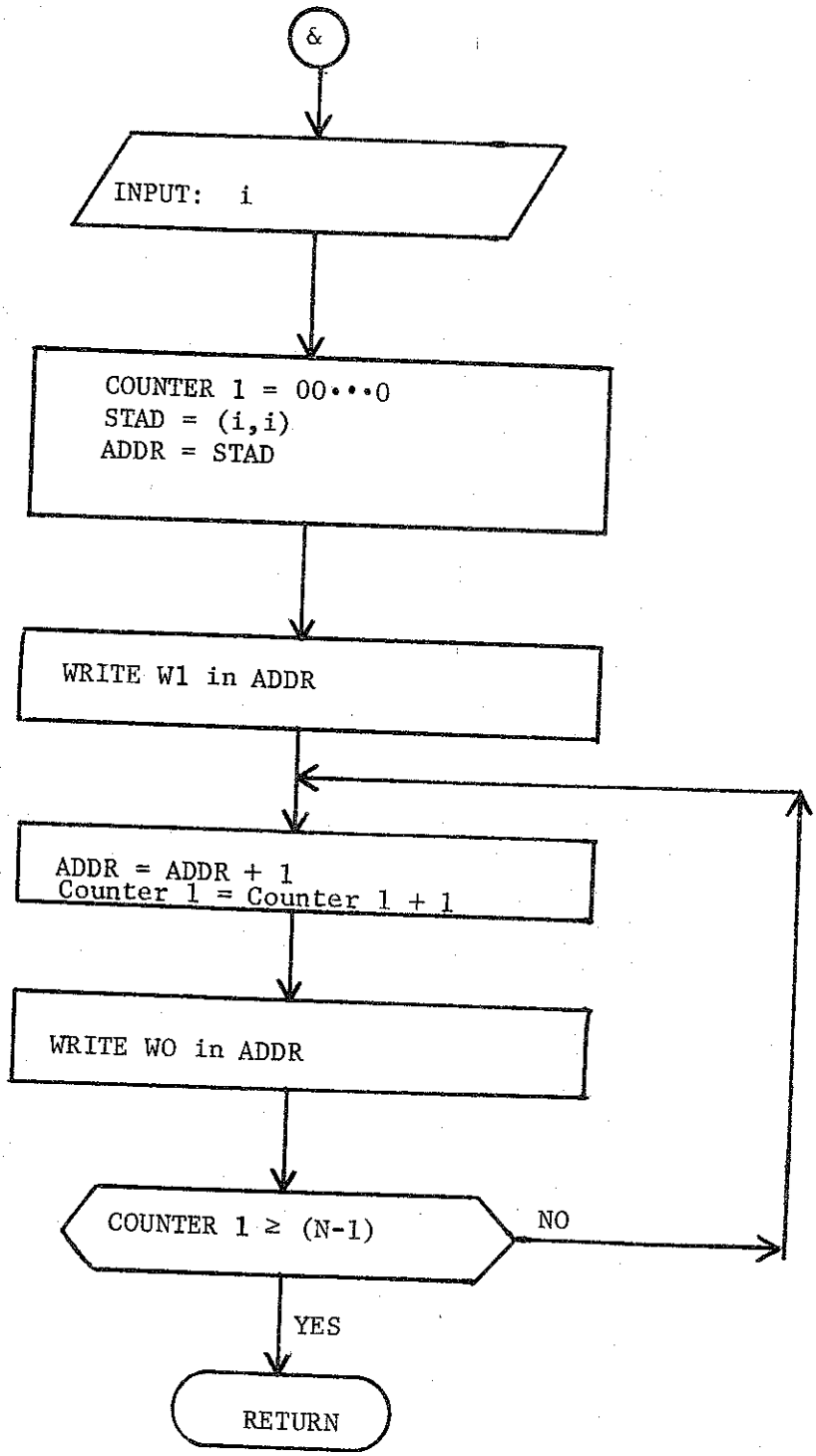


Figure 4. Flow chart for WRITE i-th PATTERN

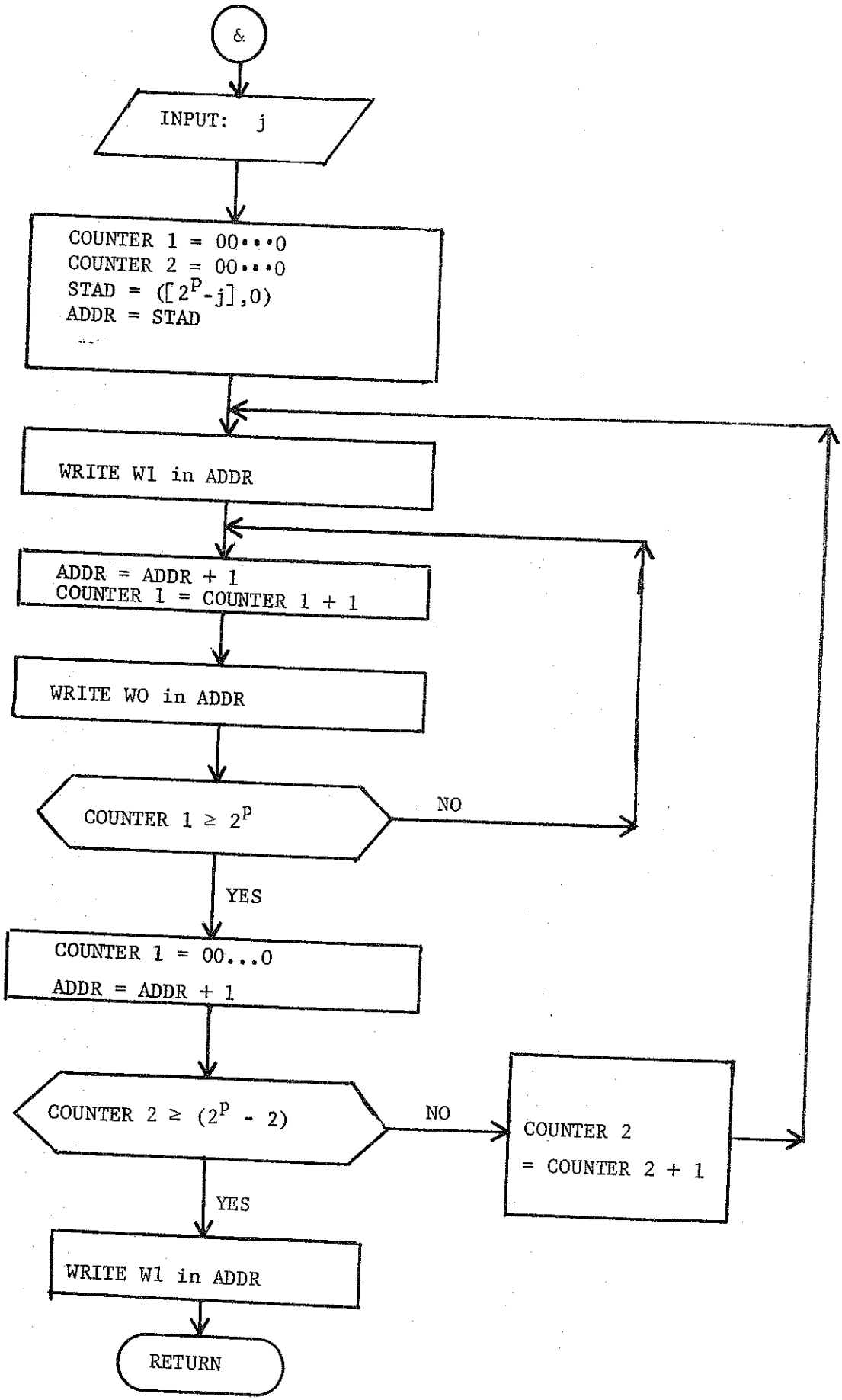
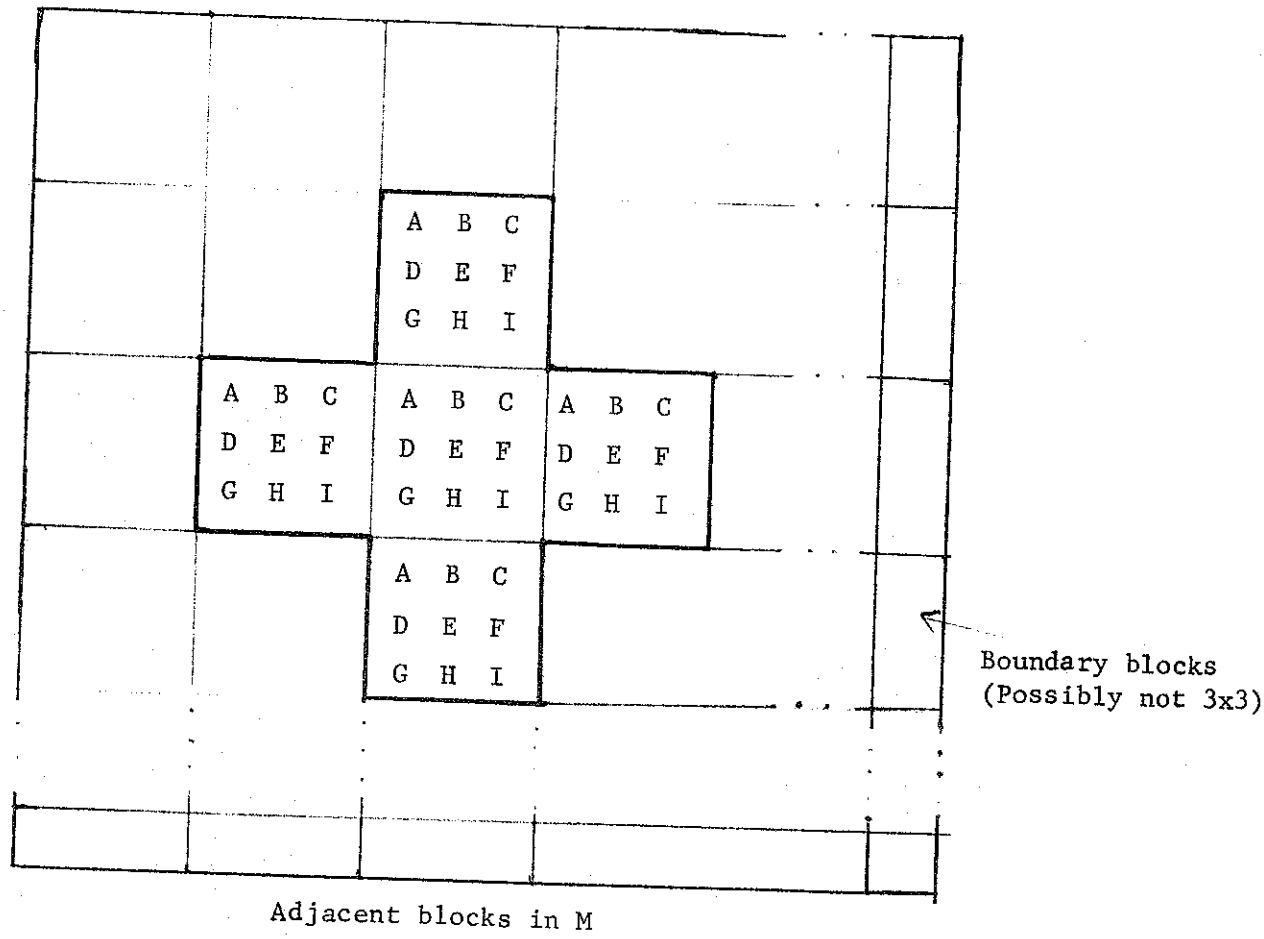


Figure 5. WRITE j-th PATTERN



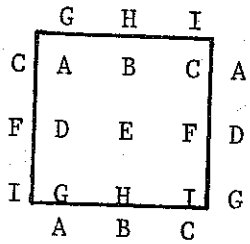


TABLE 5:

Top Neighbor	Left Neighbor	Storage Element	Right Neighbor	Bottom Neighbor
G	C	A	B	D
H	A	B	C	E
I	B	C	A	F
A	F	D	E	G
B	D	E	F	H
C	E	F	D	I
D	I	G	H	A
E	G	H	I	B
F	H	I	G	C

Figure 6. Storage elements and their neighbors in a block.

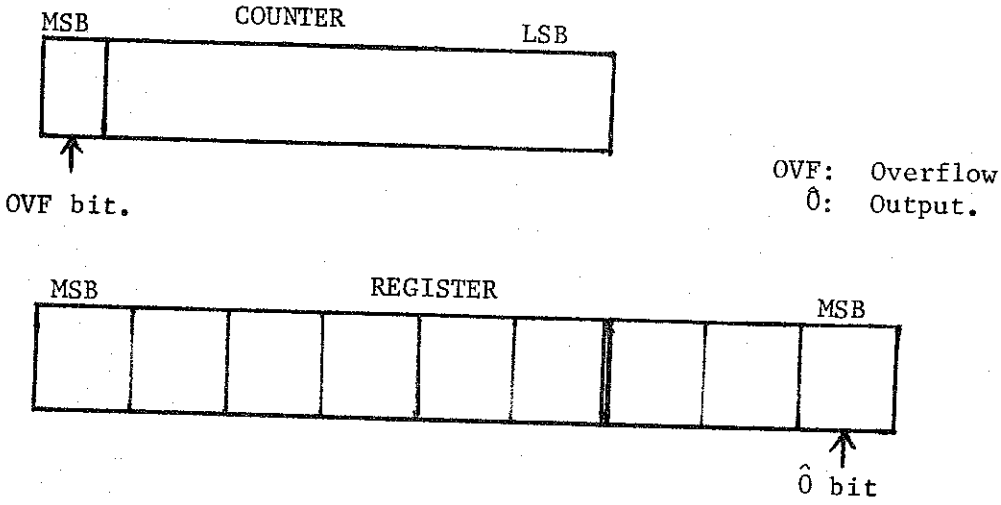


Figure 7. Controller Register

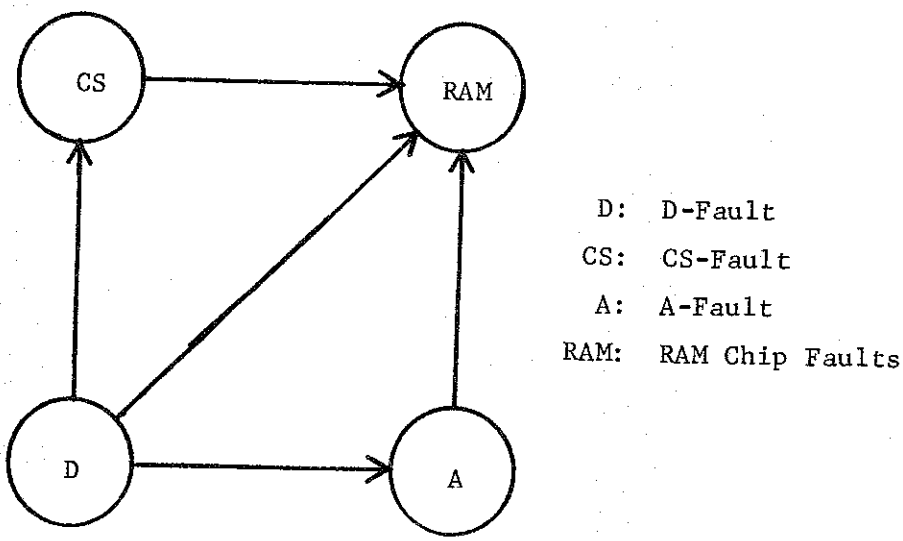


Figure 9. Diagnostic graph of a RAM unit

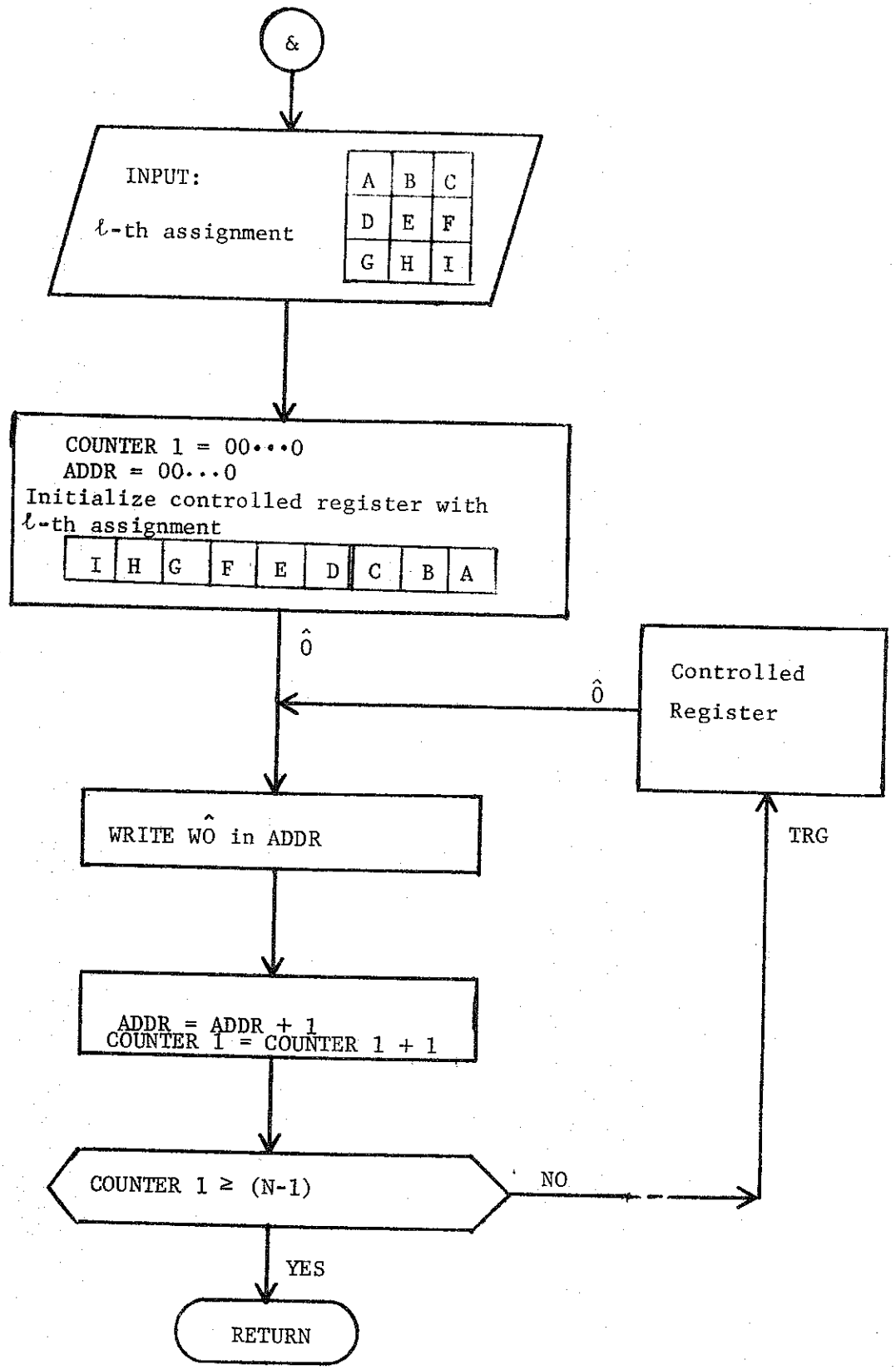


Figure 8, WRITE ℓ-th pattern.

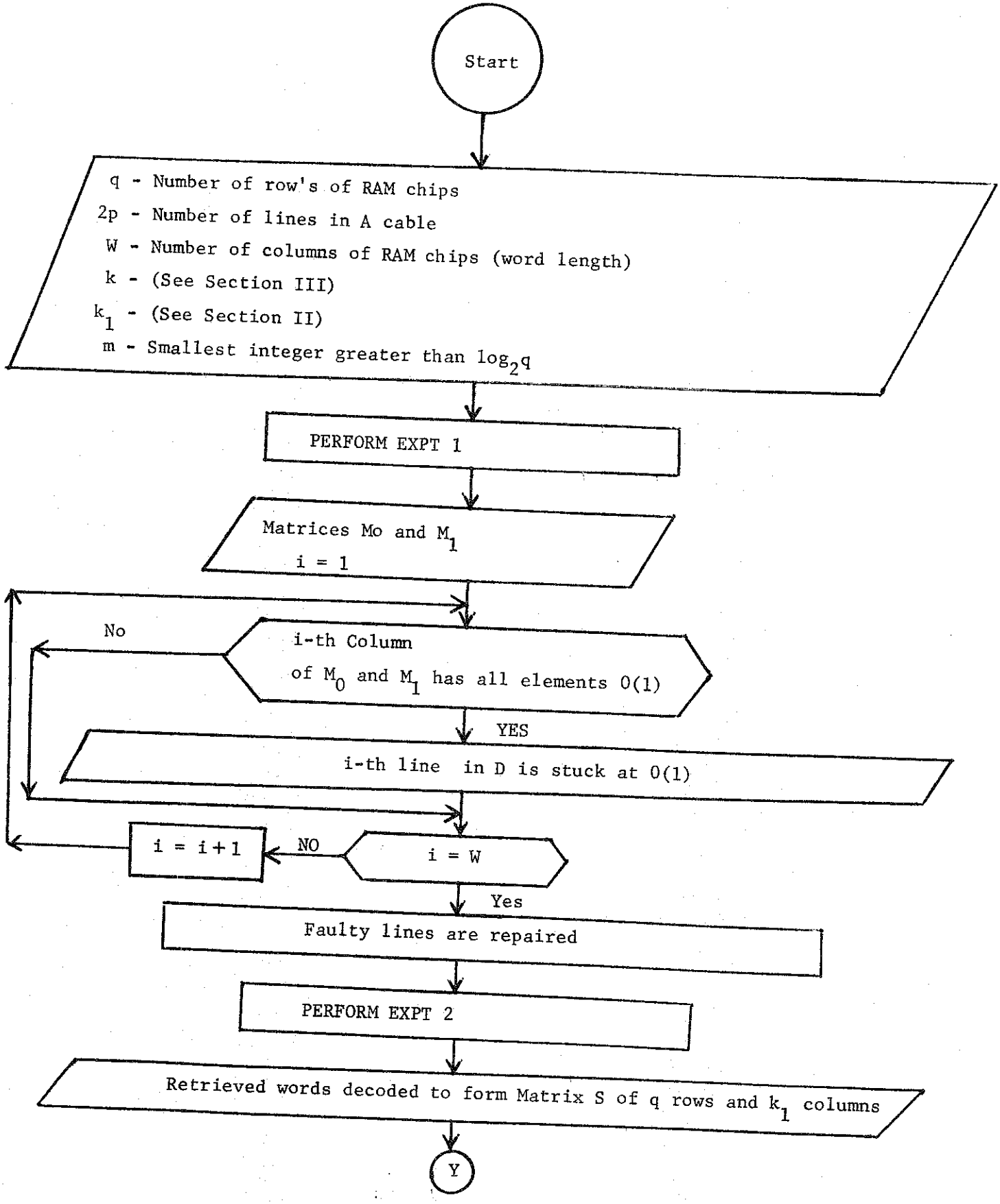


Figure 10. Flow chart for locating faulty components in a RAM unit.

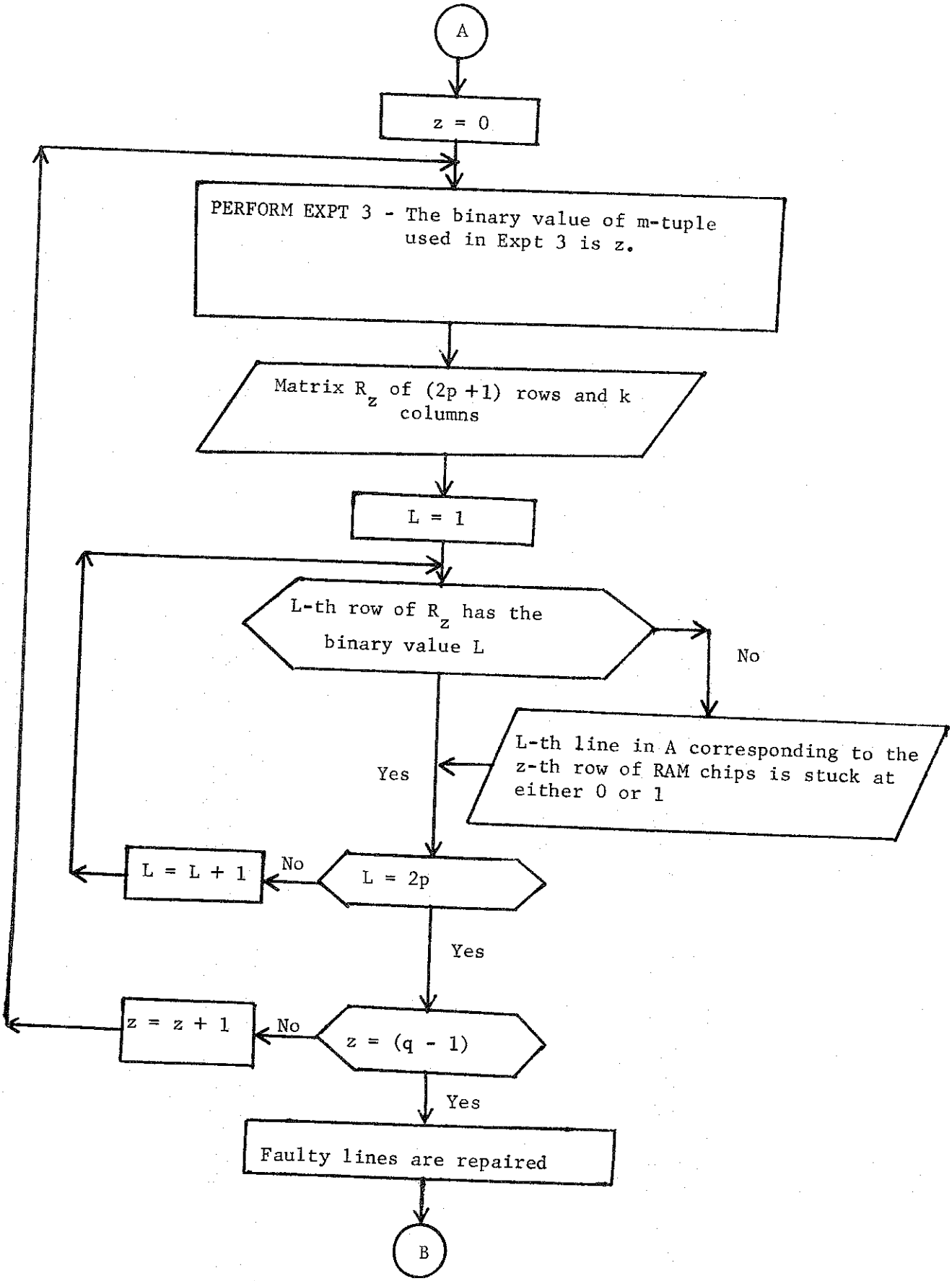

A
z = 0
PERFORM EXPT 3 - The binary value of m-tuple used in Expt 3 is z.
Matrix R_z of (2p +1) rows and k columns
L = 1
L-th row of R_z has the binary value L
No
L-th line in A corresponding to the z-th row of RAM chips is stuck at either 0 or 1
Yes
L = L + 1
No
L = 2p
Yes
z = z + 1
No
z = (q - 1)
Yes
Faulty lines are repaired
B

Figure 10.

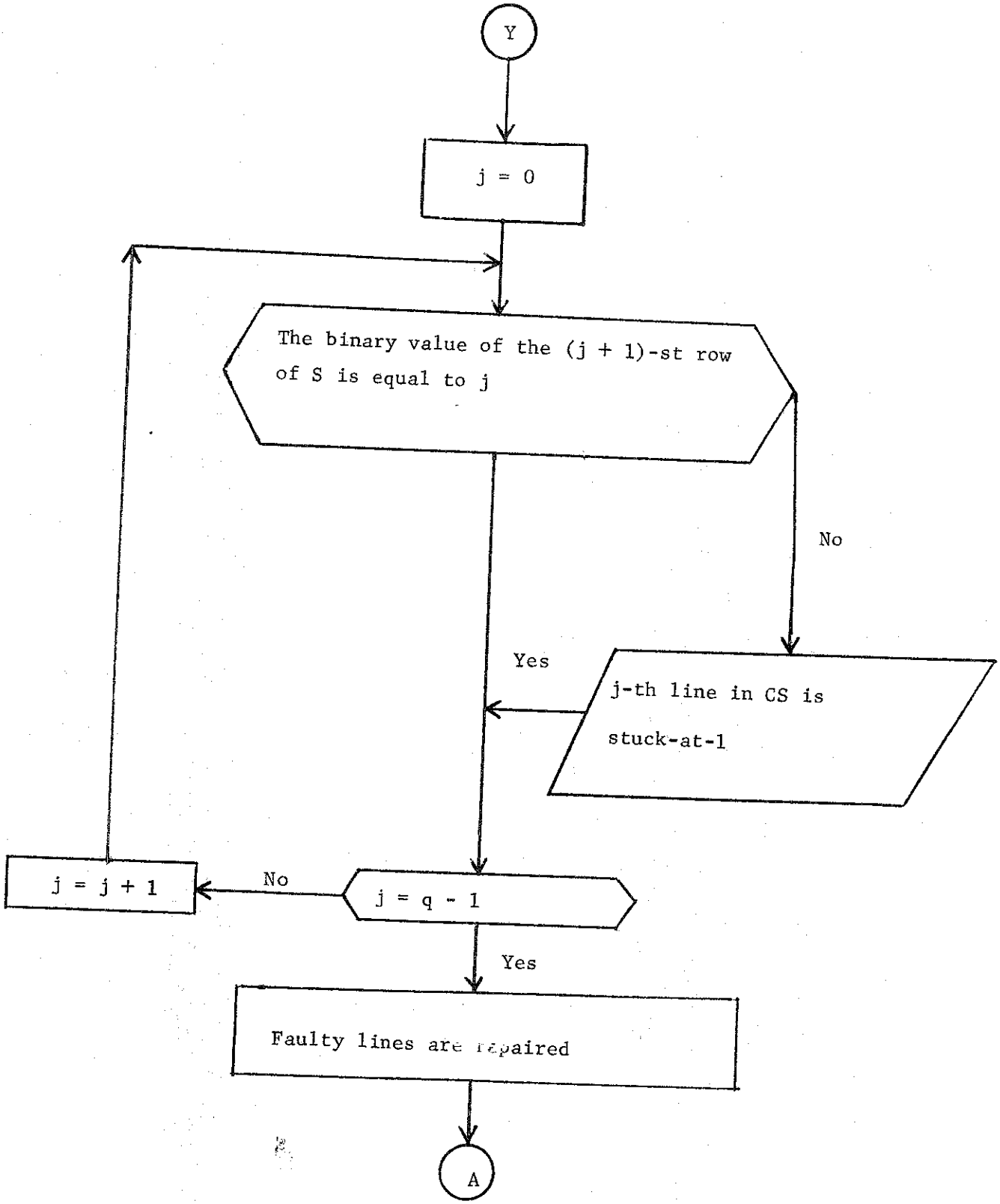

Y
j = 0
The binary value of the (j + 1)-st row of S is equal to j
No
Yes
j-th line in CS is stuck-at-1
j = j + 1
No
j = q - 1
Yes
Faulty lines are repaired
A

Figure 10.

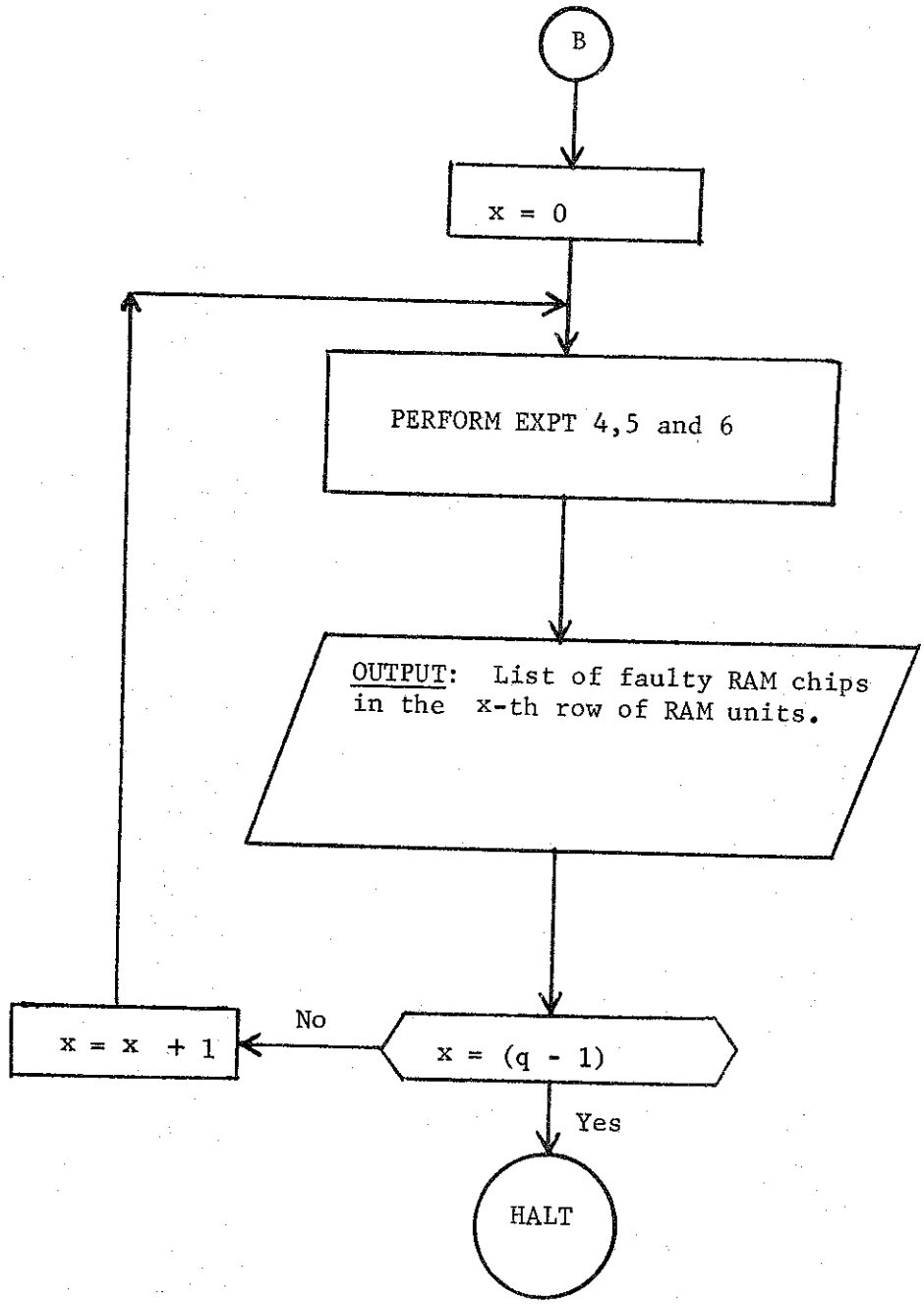

B
x = 0
PERFORM EXPT 4,5 and 6
OUTPUT: List of faulty RAM chips in the x-th row of RAM units.
x = (q - 1)
No
x = x + 1
Yes
HALT

Figure 10.