Scientific Journal Impact Factor (SJIF): 1.711

International Journal of Modern Trends in Engineering and Research

IJMTER

ISSN (ONLINE) : 2349 - 9745

ISSN (PRINT) : 2393 - 8161

New Technology for Memory Tests Design

Vladimir Ryabtsev[1], Mudar Almadi[2]

[1] *Department of Electrical Engineering, Volgograd Agricultural State University, Volgograd, Russia*

[2] *Semeon Analytics, Montreal, Canada,*

Abstract— Multi-valued transformations of the incidence matrix of addresses transitions graphs that reduce the complexity of the design of micro U-Y-schemes of memory's algorithms diagnostic tests are proposed. The algorithm of test MATS ++, suitable for further synthesis programming language Prover is presented.

Keywords- algorithm; memory; test; incidence matrix; transitions graphs.

I. INTRODUCTION

To perform the diagnostic test of high-speed memory chips it is appropriate to use the testers, with multi-processor structure, comprising several groups of address code generators, data and digital comparators. However, the development of algorithms and programs of these testers is difficult; therefore, the task of test program technology development design is actual, which contains methods for parallelization and displaying the operations performed by a given algorithm. The desire to improve the quality of storage devices has led to march tests family: march A, march B, march C, march CR, march G, march LA, march LR, march RAW, MATS, MATS++, and others [1-3].

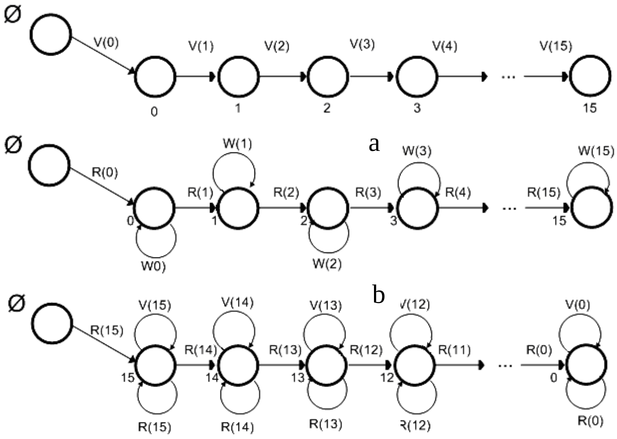
To reduce the complexity of test programs development it is necessary to create a technology of algorithms synthesis and software for diagnostic tests of high-speed memory chip that contains the methods and means for a given algorithmic description is a compact program for a test that ensures the formation of test without gaps bars refer to the object of diagnosis [7].

II. REPRESENTATION OF TEST'S ALGORITHM AS GRAPHS AND PSEUDOCODE

The most visible form of algorithms displaying is a graph of address transitions G = (A, B), where the vertices A is a set of memory cells addresses, and empty set of cells, which corresponds to the initial phase of testing when there is no selected cell: $A=\{a_0, a_1, \ldots, a_{n-1}, \varnothing\}$, where n – number of memory cells. Set of arcs B = {W, V, R} corresponds to the write operation unit (W), record zero (V) or data reading (R).

As an example, w consider the sequence of operations that performed in three cycles of test MATS ++ and in the form of graphs of address transitions are shown in Figure. 1.

However, the graph of address is an object-oriented form of the display of the operation, but does not take into account the peculiarities of the structure and format commands diagnosed device.



a

b

c

Fig. 1. Graphs address transitions of test MATS + +:
a –First cycle of operations;
b –Second cycle of operations;
c –Third cycle of operations.

To describe the sequence of performed tests operations, it is advisable to use the recording of the algorithm as pseudocode. Then, the recording of test MATS++ algorithm as pseudocode takes the form [4]:

$$\pi_{MATS\ ++} = \underset{a=0}{\overset{n-1}{P}} V(a) \underset{a=0}{\overset{n-1}{P}} (R(a)W(a)) \quad \underset{a=n-1}{\overset{0}{P}} \quad (R(a)V(a)R\,(a)).$$

Zero Write operation in the cell with an address a displays the operator V (a), recording unit – W (a), reading – R (a). Symbol P defines a sequence of performed operations; the subscript operator defines the address of the initial cell of the memory, and the superscript - end address.

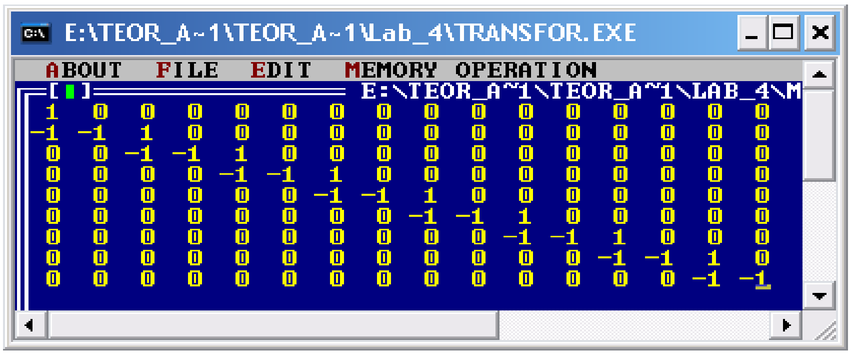
Directed graph G = (A, B), containing n vertices and m edges, can be presented as the incidence matrix M, whose elements are defined as follows: $a_{i,j}$ = 1 if edge j leaves vertex i; $a_{i,j}$ = - 1 if edge j is coming to a vertex i or vertex i has a loop, $a_{i,j}$ = 0 if edge j has no common point with the vertex i. For example, suppose the incidence matrix, which has the form and contain - rows - columns. After performing the transformations outlined in [5], we obtain the incidence matrix, which reflect the operating performance of individual processors:

$$\left(a_{i\,j}\right)_{m\,n} \to \begin{cases} \left(a_{i\,p}\right)_{m\,h} \\ \left(a_{i\,q}\right)_{m\,h} \\ \ldots \\ \left(a_{i\,w}\right)_{m\,h} \end{cases}$$

Where $a_{i\,p}, a_{i\,q}, \ldots a_{i\,w}$ – the elements of the newly formed matrix; $h = n/k$ – the number of columns in the matrices data

To parallel the Micro operations we use a method, which based on a matrix method of representing and multi-valued transformations matrices of the incidence graph of address transactions, which is implemented in the program Transfor.exe [5]. After downloading the program Transfor.exe a menu appears from which you can select the desired operation. To activate the menu key it used F10. Input data is the incidence matrix of the transformed graph, which must be presented in a separate file, or it can be typed using the option "New" of command "File". To perform the transformation of the incidence matrix it needs to activate the command "Memory Operation".

At the request of the program you must specify the dimension of the original matrix and the number of parts that are needed to parallelize the algorithm. For the second cycle of operations of the test MATS ++ algorithm for memory capacity 8 bit incidence matrix, formed on the working field of program Transfor.exe, shown in Fig. 2.


E:\TEOR_A~1\TEOR_A~1\Lab_4\TRANSFOR.EXE
ABOUT FILE EDIT MEMORY OPERATION

```
═[■]══════════════════ E:\TEOR_A~1\TEOR_A~1\LAB_4\M
  1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 -1  -1   1   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0  -1  -1   1   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0  -1  -1   1   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0  -1  -1   1   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0  -1  -1   1   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0  -1  -1   1   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0  -1  -1   1   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0  -1  -1
```

Fig. 2. Incidence matrix of addresses transitions of test MATS + +

After performing the transformations of the original matrix, we obtain the incidence matrix showing the addresses transitions that must be performed by individual shapers codes address. As is evident from the analysis of the contents of the matrix, the first and second shaper scan odd cell, while the third and fourth - pair. In each cycle of addressing to memory chip all shapers change the address code in to 2.

$$M_1 = \begin{bmatrix} 1\ 0\ 0\ 0 \\ -1\ 1\ 0\ 0 \\ 0\ 0\ 0\ 0 \\ 0\ -1\ 1\ 0 \\ 0\ 0\ 0\ 0 \\ 0\ 0\ -1\ 1 \\ 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ -1 \\ 0\ 0\ 0\ 0 \end{bmatrix}; \quad M_2 = \begin{bmatrix} 1\ 0\ 0\ 0 \\ -1\ 1\ 0\ 0 \\ 0\ 0\ 0\ 0 \\ 0\ -1\ 1\ 0 \\ 0\ 0\ 0\ 0 \\ 0\ 0\ -1\ 1 \\ 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ -1 \\ 0\ 0\ 0\ 0 \end{bmatrix};$$

$$M_3 = \begin{bmatrix} 1\ 0\ 0\ 0 \\ 0\ 0\ 0\ 0 \\ -1\ 1\ 0\ 0 \\ 0\ 0\ 0\ 0 \\ 0\ -1\ 1\ 0 \\ 0\ 0\ 0\ 0 \\ 0\ 0\ -1\ 1 \\ 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ -1 \end{bmatrix}; \quad M_4 = \begin{bmatrix} 1\ 0\ 0\ 0 \\ 0\ 0\ 0\ 0 \\ -1\ 1\ 0\ 0 \\ 0\ 0\ 0\ 0 \\ 0\ -1\ 1\ 0 \\ 0\ 0\ 0\ 0 \\ 0\ 0\ -1\ 1 \\ 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ -1 \end{bmatrix}.$$

The obtained results allow us to greatly facilitate the process of test program's synthesis.

III. APPLICATION OF U-Y SCHEMES TO SYNTHESIS OF TESTS ALGORITHM OF STORAGE DEVICES

For the synthesis of algorithms and the subsequent development of test programs necessary to construct a display $O:(A,B) \longrightarrow (U,Q)$ that will allow the algorithm to implement the test, given by graph G = (A, B), in the form of the algorithm defined in terms of basic conditions of U and a matrix of micro operation Q. To describe the basic conditions, and operators that are used in diagnostic testing of high-speed memory chips, it is advisable to use U-Y schemes of algorithms [6], which are determined by six:

$$(A, U, Y, T, a^{(0)}, a^{(1)}).$$

Here, A – set of circuits states, T – set of transitions, U – set of basic conditions, Y – set of basic operations, $a^{(0)}$ – initial state of the circuit, $a^{(1)}$ – final state of the circuit. Each transition in the set T is a quadruple (a, u, q,), here u – elementary condition; q – elementary operator. This quartet will be written as $a \xrightarrow{u\,/\,q} a'$. This statement reads as follows: in the true value of the condition u scheme A moves from a state to state. If the transition condition is always performed (u = 1), it can be omitted in the description of the algorithm: $a \xrightarrow{q} a'$, empty operator q = e can also be omitted: $a \xrightarrow{u} a'$. Schemes of algorithms can be conveniently represented by graphs with loaded arcs. Vertices of such graphs are the states of the circuit: $\{a^{(0)}, a_1, a_2, \ldots, a_{m-1}, a^{(1)}\}$· by changing the state $a_i \xrightarrow{u/q} a_{i+1}$, then the vertices ai and ai+1 is connected by an arc and this arc is loaded with a pair u/q. To describe the parallel micro-operations performed in the diagnostic tests of memory chips, it is proposed to use vector form of states schemes recording of algorithms $a \xrightarrow{\mathbf{U/Q}} a'$. For the chosen coefficient the parallel of k operations in a diagnostic device, these expressions take the form:

$$\mathbf{U} = \begin{bmatrix} x \\ x \\ \vdots \\ u_{k-1} \end{bmatrix}$$ –Vector-predicate of basic conditions of transitions.

Algorithm's scheme state used in the current time; x – indifferent state transition conditions; u_{k-1} – used condition for the transition formed by (k-1)-nd shaper;

$$\mathbf{Q} = \begin{bmatrix} c_0, m_{x_0}, m_{y_0}, m_{t_0} \\ c_1, m_{x_1}, m_{y_1}, m_{t_1} \\ \cdots \\ c_{k-1}, m_{x_{k-1}}, m_{y_{k-1}}, m_{t_{k-1}} \end{bmatrix}$$ – Matrix of basic micro operations, which are carried out 0, 1, …, (k-1) - operational processor.

Here c_i – the code of working recording micro-operations w, reading r and data comparison a; $m_{x_i}, m_{y_i}, m_{t_i}$ – codes of changing micro-operations in the coordinates x, y, and data code respectively, which form the i operational processor. The variable k defines the coefficient of operations paralleling.

If we choose the coefficient of parallelization of operations k = 4, then a situation arises where the third cycle test MATS++ number of operators to access the memory is equal to 3, which is not divisible by four, so you need these operators perform cyclically in a chain of consecutive commands. Number of cycles of repetition of these operators is determined by the formula: m = 3n/s, where n – capacity of tested memory; s – the number of operators to access memory in the chain of command. In turn, the number of operators of the chain s = k * z should be briefly coefficient parallel operations and the number of operators in a fragment of the algorithm test.

U-Y- chart of algorithm test MATS ++ with a coefficient parallel operations equal to 4 is shown in Fig. 3. For the cyclic performance of the operators of the third cycle test MATS ++ uses the variable j, which is decremented at the end of each cycle. By changing the initial value of variable j, we can establish the volume of products being diagnosed.

For the synthesis of tests algorithms it is applying conditions of performance of transitions U = {K, L}, which contains two subsets K and P. The subset is formed to address comparators, comparing the current address code with the code address of the initial cell or a code address last cell. The device diagnosis codes address comparison is done for each coordinate addresses of X and Y separately:

$$K = \{(x \neq nx), (x \neq gx), (y \neq ny), (y \neq gy)\},$$

where nx, ny – the destination address of the memory cells by X and Y coordinates, respectively;
gx, gy – starting memory addresses of the corresponding coordinates.

Also possible to use a disjunction of the results of comparison on two coordinates:

$$(a \neq na) = (x \neq nx) \vee (y \neq ny); (a \neq ga) = (x \neq gx) \vee (y \neq gy).$$

Signs L of conditional transitions form a set:

$$L = \{(i \neq 0), (j \neq 0), (k \neq 0), (q \neq 0), (h \neq 0)\},$$

where i, j, k, q, h – the programmed counters for general use.

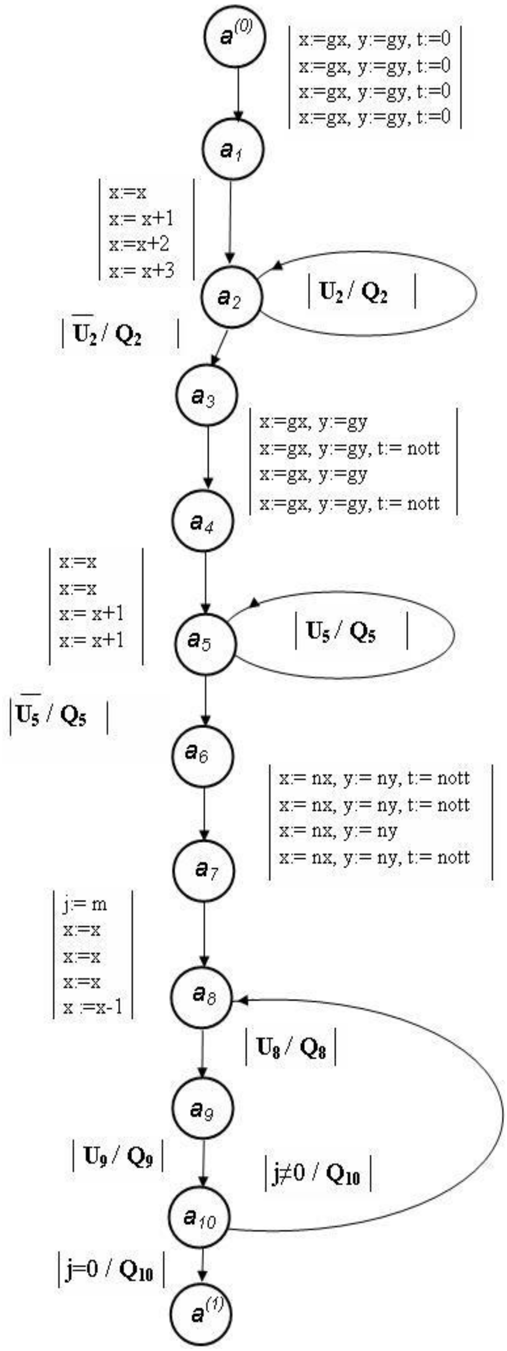


Figure 3. U-Y-chart of the test MATS + + algorithm

Set of elementary micro operation contains two subsets, the first of these Q_1 contains micro operation code change addresses, and the second – Q_2 ensures the formation of reference data.
In a subset Q_1 the following elementary micro-operations are included:

Q_1={x:=x, x:=x+1, x:=x+2, …, x:=x+p, y:=y, y:=y+1, y:=y+2, …, y:=y+p, y:=y+1*, y:= y-1*},

Where symbol * denotes the presence of the transfer or loan of coordinates X.
A subset of Q_2 includes the following micro-operation:

Q_2 = {t:=t, t := nott, t :=s, t:= t+1, t := t+2, …, , t := t+p},

Where s – code of the original data. For most of the tests it is enough the value of constant p = 4.

To select the conditions for transition in the structure of the format of microinstructions device provides a single field, providing job facilities shaper address code, the signs of transition which will be used.

As a rule, first of all tests performed in the background recording the entire storage cells of diagnosed device. To prepare for the conveyer of addresses in the device, which provides the coefficient of parallelization of operations k = 4, you must perform the following sequence of micro-operations:

$$a^{(0)} \xrightarrow{\begin{bmatrix} x:=g\,x\,,y:=g\,y\,,t:=0 \\ x:=g\,x\,,y:=g\,y\,,t:=0 \\ x:=g\,x\,,y:=g\,y\,,t:=0 \\ x:=g\,x\,,y:=g\,y\,,t:=0 \end{bmatrix}} a_1,\ a_1 \xrightarrow{\begin{bmatrix} x:=x \\ x:=x+1 \\ x:=x+2 \\ x:=x+3 \end{bmatrix}} a_2.$$

To record a background of zeros in all cells, it is necessary to perform the following micro operations:

$$\mathbf{Q}_2 = \begin{bmatrix} w,\ x:=x+4\,,y:=y+1* \\ w,\ x:=x+4\,,y:=y+1* \\ w,\ x:=x+4\,,y:=y+1* \\ w,\ x:=x+4\,,y:=y+1* \end{bmatrix}$$

As the transition flag is used the following conditions $u_2 = (x \neq nx) \vee (y \neq ny)$ for accessing to the memory cells during the second cycle of test MATS ++ uses the following transition's condition $u_5 = u_2 = (x \neq nx) \vee (y \neq ny)$ in the third cycle of this test for the transition of program in the state of a_8 the condition $j \neq 0$ is used. In the state $a_{(1)}$ the implementation of the test is performed.

The general methodology of designing of storage devices diagnosing programs is in the following tasks:

a) The coefficient of parallel operations of test sequences forming is selected in accordance with a prescribed speed diagnosis of memory chips, which ultimately determines the number of operation processors;

b) The reconfiguration algorithm is executed so that the number of operations to access memory in a cyclically repeating fragment of the algorithm has been operating a multiple number of processors;

c) Determines the number of cycles of repetition of separate fragments of the algorithm and the micro-operations for each cycle;

d) The original task is decomposed to the task of smaller dimension, i.e. the algorithm will be designed and compiled by interpreting system Prover for memory chips with basic capacity of 256 bits;

d) The functional dependence of the program variables on the capacity of the memory test are installed;

e) For a new type of storage device that has a specified capacity, the algorithm is formed by replacing those variables which have a functional dependence on memory capacity.

IV. CONCLUSIONS

Application of the method of transformation incidence's matrices of address transitions graphs can parallelize the micro operation, individual operating units performed diagnostic device having a microprocessor structure. This method of parallelization reduces the complexity of micro-programming of all the most common tests for testers to ensure diagnosis of high-speed circuits and memory modules. Using U-Y schemes of algorithms for the synthesis of diagnostic tests reduces the complexity of design procedures due to the convenient display of running micro operations loaded arcs of the graph of the synthesized program. The advantage of using graphs with loaded arcs for circuit synthesis algorithms is to reduce the amount of graphic material, which reduces the complexity of the design work.

At changing the structure of the tester, the appearance of the test's graph algorithm virtually unchanged, and is adjusted only a set of basic conditions transitions and operators. Thus, we can use shared graphics schemes of algorithms for designing the whole set of different programs that meet the chosen structure of the diagnostic device. The developed test's algorithm of MATS++ is suitable for further synthesis of test programs in a programming language Prover [6].

REFERENCES

[1] G.A. Harutunyan, V.A. Vardanian, "March Tests for Full Diagnosis of All Simple Static Faults in Random Access Memories", Proc. of East-West & Test International Workshop (EWDTW'06), 15-19 Sep. 2006, Sochi, Russia. – Kharkiv: KHNURE, 2006, pp. 68-71.

[2] G. Harutyunyan, H. Avetisyan, V. Vardanian, Y. Zorian, "An Efficient March Test for Detection of All Two-Operation Dynamic Faults from Subclass Sav", Proc. of East-West & Test International Workshop (EWDTW'09), 18-21 Sep. 2009, Moscow, Russia. – Kharkiv: KHNURE, 2009, pp. 175-178.

[3] S. Hamdioui , A.J. Van de Goor, M. Rodgers, "March SS: A Test for All Static Simple RAM Faults", Proc. of IEEE International Workshop on Memory Technology, Design and Test, 2002, pp. 95-100.

[4] A.V. Melnikov, V.G. Ryabtsev, Control of the memory modules of computers, Korniychuk, Kyiv, 2001.

[5] M.K. Al Madi, D.N. Moamar, V.G. Ryabtsev, Algorithms of the test diagnosing of storages of data of semiconductors, Korniychuk, Kyiv, 2008.

[6] M. Almadi, D. Moamar, Ryabtsev V., "Methodology of Algorithms Synthesis of Storage Devices Test Diagnosing", Proc. of East-West & Test International Workshop (EWDTW'10), 17-20 Sep. 2010, St. Petersburg, Russia.

[7] Vladimir Ryabtsev, Mudar Almadi,Strategy of Adaptive Diagnosing of Memory Modules,
Pages1-6, Pub. Date: Mar. 9, 2015.