TABLE OF CONTENTS

[1 AUTOMATED SYNTHESIS OF TEST PROGRAMS FOR DIAGNOSIS OF MICROCIRCUITS AND MEMORY MODULES 2](#_Toc163726552)

[1.1 Application of directed graphs for parallelizing micro-operations in test algorithms 2](#_Toc163726553)

[1.2 Method for designing UY-circuits of diagnostic algorithms for high-speed storage devices 2](#_Toc163726554)

[1.3 Methods and program for selecting an effective set of tests for diagnosing storage devices 3](#_Toc163726555)

[2 METHODOLOGY FOR DESIGNING FAIL-TOlerant STORAGE DEVICES 5](#_Toc163726556)

*n*

[2.1 Structure of built-in self-testing tools for memory chips 5](#_Toc163726557)

[2.2 The use of digital machines for building built-in self-testing tools 5](#_Toc163726558)

[2.3 Design of firmware built-in self-test tools 5](#_Toc163726559)

[Conclusion 7](#_Toc163726560)

# Automated synthesis of test programs for diagnosis of microcircuits and memory modules

## Application of directed graphs for parallelizing micro-operations in test algorithms

A directed graph G=(A,B), containing n vertices and m arcs, can be represented as an incidence matrix M, the elements of which are defined as follows:

ai, j = 1 if arc j leaves vertex i; ai, j = -1, if arc j approaches vertex i or there is a loop at vertex i; ai, j = 0 if arc j does not have a common point with vertex i.

Let an incidence matrix M of size n\*m ​​elements be given:

M =

It is required to split it into k matrices of size (n\*m/k) each. To solve this problem, we perform the following transformations:

p { 0,1,...,k - 1} : = 1 ,

where - matrix element;

k – number.

It is possible to parallelize or combine the operation algorithms of individual generators into a common test algorithm.

## Method for designing UY-circuits of diagnostic algorithms for high-speed storage devices

When using the method of parallelizing micro-operations in test algorithms, performance increases and requirements are reduced.

## Methods and program for selecting an effective set of tests for diagnosing storage devices

The high time required to perform test diagnostics of large-capacity RAM determines the problem of selecting effective tests that cover the most likely faults [30]. In a formalized form, the problem of selecting tests can be represented as the following functional [3, 7, 9]:

F = <T, P, M>,

where T is the duration of testing;

P – probability of fault coverage;

M – number of tests.

This functionality allows you to create seven tasks that are of practical importance and are given in Table 1 and allow you to select a strategy for adaptive diagnostics of microcircuits and memory modules.

Table 1 - Memory module diagnostic tasks

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | T |  | P | M | Functions |
| 1 | var |  | var | var | P = f(T, M) |
| 2 | var |  | const | var | M = f(T, P = const) |
| 3 | var |  | const | var | T = f(P = const, M) |
| 4 | const |  | var | var | P = f(T = const, M) |
| 5 | const |  | var | var | M = f(T = const, P) |
| 6 | var |  | var | const | T = f(P, M = const) |
| 7 | var |  | var | const | P = f(T, M = const) |

Let us consider in more detail the description of these problems [37,75].

1. P = f(T, M) – is the task of determining the probability of fault coverage without limiting the time and number of tests. It does not have a formalized solution, since there are a large number of combinations of different sequences of tests used.
2. M = f(T, P = const) – defines an optimized set of tests that ensure the achievement of a given probability of fault coverage. The testing time is not limited.
3. T = f(P = const, M) – determines the minimum time during which the most favorable sequence of tests ensures the specified probability of fault coverage.
4. P = f(T = const, M) – determines the maximum possible probability of fault coverage when limiting the testing duration by enumerating all possible sequences of test execution.
5. M = f(T = const, P) – defines the minimum set of tests that provide the highest possible probability of fault coverage while limiting the testing duration.
6. T = f(P, M=const) – determines the number of test repetitions M, which will ensure the highest possible probability of fault coverage. Typically, this testing strategy is wiped out when testing the reliability of memory modules.

When performing test diagnostics of memory modules at early stages of the life cycle, it is advisable to use two testing strategies:

* Selection of an optimized test sequence based on the knowledge of diagnostic experts;
* Minimizing the test set through the use of heuristic and fuzzy algorithms.

In a similar way, you can obtain normalized values ​​for all assessments of test properties according to given criteria and present them in the form of a Table 1. For each criterion, you can obtain its average value using the formula:

# Methodology for designing fail-tolerant storage devices

## Structure of built-in self-testing tools for memory chips

Built-in Self Test, or BIST (built-in Self Test), is a method of designing additional hardware and software that can perform self-testing, that is, testing their own device operation (functionally, parametrically, or a combination of both), using proprietary circuitry that reduces dependence on external automated test equipment (ATE) [11, 40].

## The use of digital machines for building built-in self-testing tools

First, counters k and l are set to zero, then the starting address of the array of memory cells is selected, the code of two-dimensional Walsh functions is entered into the control data register, and information is written to the selected cell. The value of the current address register is increased by one and codes are written to all memory cells in the same way. Figure 1 show this.

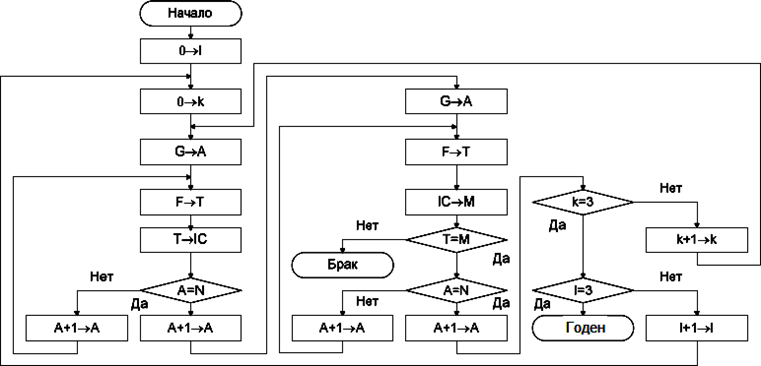


Figure 1 – Memory chip self-testing algorithm

## Design of firmware built-in self-test tools

A wide variety of self-tests can be generated using the microprogram principle of controlling the hardware built into the memory chip [75].

In Figure 2 shows the structure of a semiconductor memory chip with built-in self-testing tools.

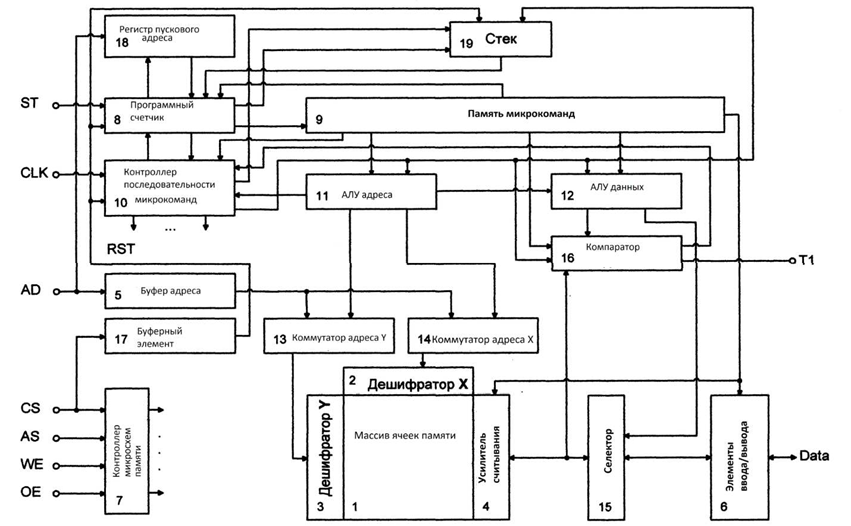


Figure 2 – Structure of a semiconductor memory chip with built-in self-testing tools

Conclusion

The complexity of developing algorithms and test programs for verifying memory chip models is significantly reduced if they are developed in the machine-oriented programming language Prover, which takes into account the structural features of the diagnosed semiconductor memory chips

The proposed method for verifying VHDL models of memory chips has the following advantages over similar developments:

1. the process of recording and comparing read data with reference values ​​is automated;
2. an algorithmic method of generating tests is used,
3. test programs previously developed in the Prover language are used, which reduces the complexity of their design and provides an easier way to localize errors in model programs and tests;