ON FAULT CHECKING IN DISCRETE DEVICES WITH MANY-VALUED STRUCTURAL ALPHABET

V. V. Brovarnik

UDC 519.47

The development of mathematical testing methods for discrete devices with many-valued structural alphabet is motivated by their potential uses in the rapidly growing technology of large-scale integrated circuits. In this article we propose a method of constructing a minimum-length input sequence capable of detecting a given single stable fault in a discrete device with many-valued structural alphabet. We consider devices with memory without external feedback loops. The method is based on the calculus of $\hat{d}$-cubes as extended to structures with memory elements. The terminology and notation follow the usage of [1].

We focus on discrete devices consisting of logic elements and memory elements [2] with k-valued structural alphabet. The memory elements of the device form groups $D_1, \ldots, D_T$. Each memory element enters only one of these groups. The setting inputs of the memory elements of each group are connected to a single bus. A setting signal passed through the corresponding bus thus sets all the memory elements of one group to the same state in a normally functioning device. The device is initialized by applying an appropriate setting signal combination (SSC) to the buses connecting the setting inputs of the memory elements.

Integer markers are assigned to the device elements and to the external input and output poles [1]. The integer markers of the input and output gates of the elements represent their connections with other elements and with input and output poles. A mathematical model of the device is provided by degenerate coverings of its logic elements and reduced tables of cubes of its memory elements. A stable physical fault in any of the elements changes the function realized by the element. Some physical faults cause a memory element to get stuck in a wrong initial state in response to a setting signal applied to the setting input. In the mathematical model of a faulty device, this possibility is incorporated in the mathematical model of the faulty element G_*.

It is required to find an input sequence R of minimal length capable of detecting a single stable fault in the device. Since the initial states of the memory elements before the beginning of the testing experiment are unknown, R starts with SSC. R is terminated by an input signal combination (ISC) $\mathfrak{X}$ to which the normally functioning device and the device with the given fault produce different responses.

We denote by $A(G_*)$ the part of the device consisting of element G_* and those elements located along the paths from G_* to the external output poles. The collection of markers on the external input gates (output poles) of $A(G_*)$ will be designated $A_{\text{in}}(G_*)$ [$A_{\text{out}}(G_*)$].

The state vector of the device is a vector whose coordinates are the markers of the memory elements. The coordinates correspond to the states of the memory elements produced by a setting sequence starting with SSC and followed by ISC. The state vector $\mathfrak{b}$ of a normally functioning device and the state vector $\mathfrak{b}'$ of a faulty device obtained by applying the setting sequence $R_{\mathfrak{b},\mathfrak{b}'}$ will be called paired state vectors. Paired state vectors may only differ in the values of the coordinates corresponding to markers of the memory elements in $A(G_*)$.

We denote by $\mathfrak{B}(t)$ the combination of all the possible paired state vectors that can be obtained by applying a setting sequence of length t. Since setting sequences of length 1 are different SSCs, paired state vectors from $\mathfrak{B}(1)$ may only differ in the coordinate $*$. This is observed only if the physical fault in the memory element G_* affects its initialization by the setting signal.

The coordinate $*$ and the markers of the memory elements in $A(G_*)$ which have different values in the paired state vectors $\mathfrak{b}$ and $\mathfrak{b}'$ form the set of markers $I(\mathfrak{b}, \mathfrak{b}')$

For elements whose markers form $I(\mathfrak{b}, \mathfrak{b}')$, we now determine the initial test combinations. The coordinates of the cubes in the initial test combination of an element are the markers assigned to its input and output gates.

Translated from Kibernetika, No. 2, pp. 65-67, March-April, 1981. Original article submitted January 22, 1979.

If the faulty element G_* in the device is a logic element, the initial test combination H_* consists of those input signals which produce different output signals in the normally functioning logic element and in the logic element with the given fault.

Let q_i and q_i^* be the values of the coordinate i in the paired state vectors $\mathfrak{b}$ and $\mathfrak{b}'$.

If the faulty element G_* is a memory element, the initial test combination H_* consists of those input signals which produce different output signals in the normally functioning memory element in state q_* and in the memory element with the given fault in state q_*'. Possibly, $q_* = q_*'$.

For a memory element G_i ($G_i \neq G_*$) the initial test combination H_i consists of those input signals which produce different output signals in states q_i and q_i'. For example, consider a memory element G_i with marker i_{in} on its input gate; if the input signal l in this element produces different output signals m and n, respectively, in states q_i and q_i', the cube

$$\begin{array}{c|c} i_{\text{in}} & i \\ \hline l & d_n^m \end{array}$$

belong to the initial test combination H_i.

All the paired state vectors from $\mathfrak{B}(t)$ which do not belong to $\mathfrak{B}(1), \ldots, \mathfrak{B}(t-1)$ and for which $\bigcup_{i \in I} H_i \neq \varnothing$ form the combination $\mathfrak{B}^*(t)$.

As in [3], we extend the basic notions and definitions of the calculus of $\hat{d}$-cubes [1] to structures with memory elements.

A mathematical model of a memory element is a reduced table of cubes, which is actually its transition table and output table. The reduced table of cubes of a memory element consists of k blocks. Let D_{in} and p_{out} be the markers of the input and the output gates of the memory element. In each cube of j-th block in the reduced table of cubes, the coordinates p_{in} and p_{out} correspond to input signals and to that state into which the memory element is switched from state j by these input signals.

The cube with coordinates p_{in} and p_{out} equal, respectively, to $\hat{d}$ and f_{tr} will be called a compact list of $\hat{d}$-cubes of the memory element.

The values of the indexes r and s ($f_{tr} = d_s^r$) in a $\hat{d}$-cube obtained by applying the choice operation are determined using its transition function λ. Thus, $r = \lambda(q, m)$, $s = \lambda(q', n)$, where q and q' are the values of the coordinate corresponding to the marker of the memory element in paired state vectors $\mathfrak{b}$ and $\mathfrak{b}'$. If r = s, this $\hat{d}$-cube will be called persistent. For a memory element with a complete system of transitions, the choice operation clearly chooses only nonpersistent $\hat{d}$-cubes.

Since the ISC $\mathfrak{T}$ is only constructed for paired state vectors, the construction of the input sequence R involves successively choosing paired state vectors from $\mathfrak{B}^*(1), \mathfrak{B}^*(2), \ldots$.

Suppose that the ISC $\mathfrak{T}$ cannot be constructed for paired state vectors from $\mathfrak{B}^*(1), \ldots, \mathfrak{B}^*(t-1)$. From $\mathfrak{B}^*(t)$ choose paired state vectors $\mathfrak{b}$ and $\mathfrak{b}'$, from $I(\mathfrak{b}, \mathfrak{b}')$ choose a marker h, from H_h choose a cube which is the first test cube c_1. For this cube determine the activity vector $\vec{a}$ and the $\hat{d}$-branching. From the $\hat{d}$-branching choose the least marker i. Using the choice operation, choose from the compact list of $\hat{d}$-cubes of the element G_i a $\hat{d}$-cube whose $\hat{d}$-intersection with the test cube c_1 gives a new test cube. Continue this process, which we call a $\hat{d}$-walkthrough, until a test cube c_* has been constructed, whose activity vector $\vec{a}$ only includes the markers of those elements which are connected directly with the external output poles from $A_{out}(G_*)$.

The process of construction of an ISC activating a path from the element G_h to external output poles from $A_{out}(G_*)$, using the test cube c_*, is called the augmentation operation. The augmentation operation is performed in two stages. The first stage yields a test cube c^* fixing the values of the signals on all the input gates from $A_{in}(G_*)$. The second stage produces a test cube in which the ISC $\mathfrak{T}$ is entered in the coordinates corresponding to the markers of the external input poles.

Suppose that some of the coordinates corresponding to the input gates from $A_{in}(G_*)$ are not defined in the test cube c_*. Construct the cubes $J(c_*)$ and $J'(c_*)$ whose coordinates are the coordinates of the test cube c_* with one of the following values: $0, 1, \ldots, k-1, a_2, a_3, \ldots, a_g$. The coordinates of the cube $J(c_*)$ are assigned the values of these coordinates in the test cube c_*; in this cube find the largest coordinate with one of the following values: $0, 1, \ldots, k-1, a_2, a_3, \ldots, a_g$. Suppose that this is the coordinate L, whose value is l. If the marker L in the device is assigned to a logic element, select a cube from its degenerate covering whose coordinate L is also equal to l and its $\hat{d}$-intersection with the test cube c_* yields a new test cube. If the marker L in the device is assigned to a memory element whose coordinate in the state vector $\mathfrak{b}$ is j, then from the j-th block of its reduced table of cubes choose a cube in which the coordinate L also takes the value l and its $\hat{d}$-intersection with the test cube c_* gives a new test cube. Continue this procedure until we have constructed a test cube c in which the values of all the coordinates corresponding to the input gates from $A_{in}(G_*)$ are equal to those values of the input signals of $A(G_*)$ in the device with the given fault which, when applied to the input gates of the elements along the activated path from the element G_h to external output poles from $A_{out}(G_*)$, produce the signals corresponding to the cube $J(c_*)$. This is the new test cube c^*.

The process of checking the above condition for the test cube c is called the comparison operation. Construct a cube K(c) whose coordinates are the coordinates of the test cube c with one of the following values: $0, 1, \ldots, k-1, a_2, a_3, \ldots, a_g$. The coordinates of the cube K(c) corresponding to the markers of the input gates from $A_{in}(G_*)$ are assigned the values of these coordinates in the test cube c. In the cube K(c) find the least coordinate with undefined value. Let this coordinate be M. If the marker M in the device is assigned to a logic element, then in its degenerate covering find a cube in which the values of the coordinates corresponding to its input gates are equal to the values of these coordinates in the test cube c. If the marker M in the device is assigned to a memory element whose coordinate in the state vector $\mathfrak{b}'$ is j, then in the j-th block of its reduced table of cubes find a cube in which the value of the coordinate corresponding to its input gate is equal to the value of this coordinate in the test cube c. The coordinate M of the cube K(c) is assigned the value of the coordinate M in the resulting cube. Continue this procedure until all the coordinates of the cube K(c) have been assigned values. The coordinates of the cube $J'(c_*)$ are assigned the values of these coordinates in the cube K(c). If $J(c_*) = J'(c_*)$, the test cube c is the new test cube c^*.

Clearly, if $h = *$, then without performing the comparison operation we can claim that the test cube c is the new test cube c^*.

Suppose that some of the coordinates in the test cube c^* are undefined. The values of the coordinates corresponding to the external input poles of the device are determined in the second stage of the augmentation operation. The procedures for constructing new test cubes are the same for both stages of the augmentation operation.

If in one of the steps of the proposed algorithm it becomes impossible to continue the construction of the new test cube, the algorithm backtracks to the previous arbitrary choice (this choice and all the subsequent choices are eliminated) and makes a new choice.

If no test cube c^* can be constructed for the chosen paired state vectors $\mathfrak{b}$ and $\mathfrak{b}'$, no such cube obviously can be constructed for paired state vectors from $\mathfrak{B}^*(t)$. $\mathfrak{B}^*(t+1), \ldots$ in which the coordinates corresponding to the memory elements from $A(G_*)$ are equal to these coordinates in the paired state vectors $\mathfrak{b}$ and $\mathfrak{b}'$. These paired state vectors should be eliminated from $\mathfrak{B}^*(t), \mathfrak{B}^*(t+1), \ldots$.

The input sequence R of length t + 1 consists of $R_{\mathfrak{b},\mathfrak{b}'}$, $\mathfrak{T}$.

LITERATURE CITED

1. V. V. Brovarnik, "Checking combinational networks synthesized from many-valued elements," Kibernetika, No. 4, 47-55 (1978).
2. Yu. L. Ivas'kiv, "Elementary automata with memory with k-valued structural alphabet," Kibernetika, No. 1, 36-46 (1977).
3. V. V. Karibskii, "Construction of an input sequence capable of detecting a given fault in a discrete device," Avtomat. Telemekh., No. 5, 153-162 (1972).