Approach to the In-Circuit Testers Training for Electronic Devices Identification

by

Baida N.P., P.O.Box 6516, 286030, Vinnitsa-30, Ukraine,
Mesyura V.I.,Kosmonavtov Street 61/2, 286027, Vinnitsa, Ukraine,
Roik A.M.,Khmelnitske Shose, 95, Vinnitsa, 286021, Ukraine.

Abstract

Algorithms of synthesis of test actions must provide given trustworthiness and completeness of test relatively to considered class of unit under test (UUT) faults. Efficiency of such algorithms in the decisive degree is determined by the presence of adequate mathematical model of object. Basis for creating mathematical model is, as a rule, UUT functional diagram. However, in case of technical documentation lack, information about demanded UUT characteristics we can receive only in the UUT identification mode by building mathematical model of UUT on the results of input and output signals measurements [1].

The goal of the realization in-circuit identification mode is expending of in-circuit test systems (ICT) functional possibilities:

- restoring UUT functional diagram in case of technical documentation lack or with existing changes of diagram which is not reflected in this documentation;
- automated test program generation is realized in the ICT self learning mode (in this case studying UUT may be defective).

Present paper is devoted to consideration of training system of electronic devices in-circuit test and identification.

Keywords: digital component, identification, in-circuit tester, training.

Formal description of identification object.

The offered structural-functional approach for digital UUT identification is based on complex analysis of structural and functional characteristics of components D_j and UUT in the whole. Accordingly, mathematical model of component D_j is presented in the following form:

$$M_j = (F_j, V_j),$$

where F_j -- function realizing by component, reflecting relationship between signals levels presented at its pin set V_j,

$$V_j = X_j \cup Y_j,$$

where X_j - set of component inputs and Y_j -- set of component outputs.

Component structural properties can be represented by structural properties vector $Z(D_j)$. Its length coincides with number of component pins, and its coordinates meaning is defined as following:

$$z_{i,j} = \begin{cases} 1, & \text{if } v_{i,j} \in X_j; \\ 0, & \text{if } v_{i,j} \in Y_j. \end{cases}$$

For reduction of task dimension we use decomposition presentation of digital component function :

$$F_j = \left\{ f_{i,j} \,\middle|\, i = \overline{1,|Y_j|} \right\},$$

where $f_{i,j}$ -- set of elementary functions implemented by component.

Each elementary function $f_{i,j}$ reflect dependence of the component D_j output $y_{i,j} \in Y_j$ state on states of component inputs certain subset $X_{i,j} \subset X_j$ in algebraic form [2].

For example, state of component SN74LS74 output Q_1 will be described as following:

$$Q_1 = (1 - S_1) + S_1 \times R_1 \times \left[Q^{\alpha}{}_1 + C' \times (D_1 - Q^{\alpha}{}_1) \right]$$

where Q^{α} -- the output Q state at the previous test tact.

C' -- the dynamic input.

Such component D_j functional properties represantation will be named as its functional-algebraic model (FAM) [2].

Let's take the meaning of signal's levels on each component pins at the given time t as $v^t_{i,j}$. So far as $v^t_{i,j}$ can acquire meanings from the set $\{0,1\}$ only, component D_j state at each time can be described by digital characteristic n^t_j:

$$n_j^t = \sum_{i=1}^{|V_j|} 2^{i-1} \cdot v_{i,j}^t$$

We will name set $N_j = \{n_j^k\}$, obtained from FAM by substitution of all possible input signals combinations, as representation of component D_j states space in digital form that is designated as N_j or $N(D_j)$. State space $N(D_j)$ represents component D_j structural and functional properties.

Knowledge representation.

ICT uses heuristic knowledge model. Knowledge about electronic devices represented by productions that describe, for example: correspondence between component elementary functions and component type, heuristic rules of electronic device functional diagram designing, heuristic rules of electronic devices arrangement, etc. For example:

IF component has counter input THEN it's counter with probability p=1,

IF component pins are linked together and with supply bus THEN there are component outputs with probability p=0.8,

IF two component pins are linked together AND aren't linked with any pin of the same component AND aren't linked with any pin of any component THEN one of them is an output and another is an input with probability p=1, etc.

Software is built as set of pattern-control modules. Current situation is analyzed at the each step of process and, after pattern analysis, program determines module that can handle this situation.

Knowledge about components being recognized and functions implementing by them realized in object types hierarchy . Set of FAM elementary functions is used as a basis of hierarchy. Population of such functions defines component functionality. Abstract object type TFuncEl (basic functional element) includes description of element function, provided by basic element, and description of basic element inputs and outputs numbers and type. The basic methods of TFuncEl are:

- element states space building by FAM,
- determination of element output state by its inputs set,
- calculation of inputs set corresponded to the element output given state,
- checking permission of certain state for given component type.

For components, fragments and printed circuits description following object types are used: TChip, TCluster and TUnit, respectively. These object types are connected with TFuncEl and between themselves by aggregation relation.

Principles of ICT training.

At the stage of ICT training, initial set of components $\omega_1 \ldots \omega_k$ is showed to it. These components are represented by the set of their properties $R_i = \{r_j^n\}$, $n = \overline{1,N}$. However, component's belonging to some class isn't indicated. Instead, we offer rules set. According to this set, at the stage of training, ICT on its own produces some classification, which, in common case, may differ from habitual for specialists in the given area. The aim of ICT training is to split components set showed at such classes, that we can apply certain optimized algorithms of identification to each of them.

Nevertheless, such approach frees operator from labor-consuming work of component classification and allows ICT to create most suitable classes for its functioning. From this point of view, according to approach proposed, in ICT vocabulary of properties has being included:

- set of FAM elementary function realized by digital component,
- vector of component D_j structural properties represented as outputs vector V_j^O and inputs vector V_j^I,
- inputs and outputs vectors $V_{j,k}^I$, $V_{j,k}^O$ for each basic element,
- component inputs types.

All mentioned above properties can be easily obtained from component FAM, and, on the contrary, all these properties are sufficient for FAM synthesis. It should be noted, that component supply pins described by us as input pins are not accounted in the inputs vector. For example, component properties vector can take such form:

$$Z(SN74LS74) = \{(f_l, f_m), (1E3F, 01B0), (000F, 0030; 1E00, 0180), (R1, D2, C3, R4, S10, C11, D12, R13)\},$$

where $$f_l = (1-S) + S \times R \times \left(Q^\alpha + C' \times \left(D - Q^\alpha\right)\right)$$

$$f_m = (1-R) + S \times R \times \left(\overline{Q}^\alpha + C' \times \left(D - \overline{Q}^\alpha\right)\right)$$

During training certain set of classifications is realized. This set unites components with the following properties:

- functional (function realized by component),
- structural-functional (presence of same type basic elements within various components),

• structural (coincidence, may be partial, of basic components inputs vector).

Recognition algorithm based on estimation calculation is used for simple solving about component's belonging to defined class. Corresponding subsets of properties play part of base sets of each classification algorithm.

For example, abstract object of type TFuncEl is used as a basis for classification by structural-functional properties. It disposes in vertex of corresponding functional object hierarchy tree. So, components having clocking will be attributed to one object's class, because even one from such component basic elements have a common ancestor that disposes at the top of corresponding hierarchy tree.

Fragment of such hierarchy was shown in Fig.1.

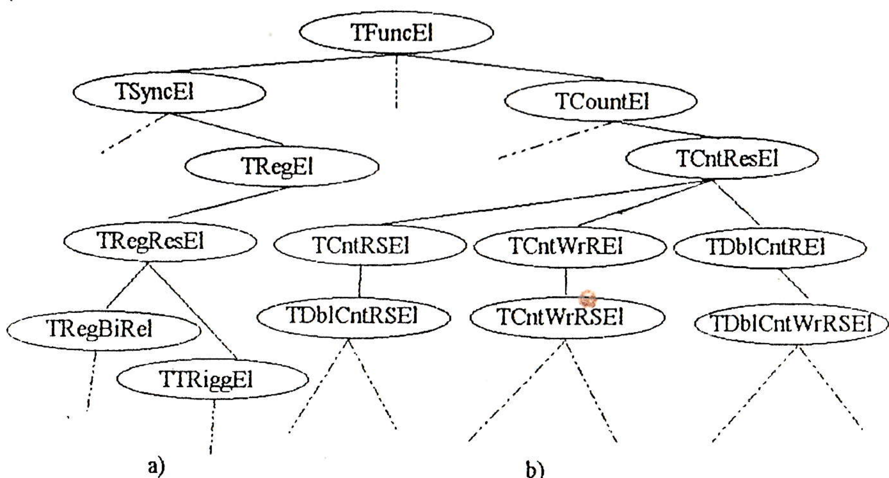

Fig.1. Fragment of base functional elements' hierarchy.

TSyncEl -- includes clock input (C),

TRegEl -- includes D and C inputs (D -- data input),

...

TTrigEl -- includes D,C,R and S inputs (R -- reset input, S -- set input),

...

TCountEl -- includes +1 input (up-counter input),

TDblCntWrRSEl -- includes +1, -1, Wr, R and S inputs (-1 -- down-counter input, Wr -- parallel write input).

Split of a class at some number of new classes can take place when component's library extends. With it, elements-ancestors will be disposed at the lower levels of hierarchy tree in comparison with initial ancestor class vertex.

For classification, by structural properties, fulfillment handy to use representation of base functional element inputs set in form of lattice and namely, unit n-dimensional cube. We will consider the principle of classification on the own sublattice of unit 4-dimensional cube showed in Fig.2.

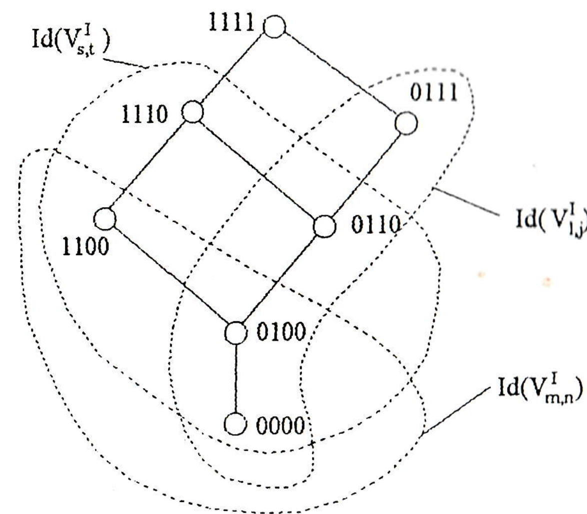

Fig.2. Lattice used for component's identification.

In such case, the component D_k with one of its base element inputs vector $V^I_{i,k}$ will be attributed to one and the same class when the following conditions are implemented:

$$V^I_{i,k} \subset Id(V^I_{l,j}) \qquad (1)$$

$$V^O_{i,k} \cap V^I_{l,j} = \varnothing \qquad (2)$$

$$V^I_{i,k} \neq \varnothing \qquad (3)$$

where $V^I_{l,j}$ -- inputs vector of ancestor basic element.

Condition (1) determines principle of polymorphism by analogy with object-oriented programming. Application points of test actions, which provide possibility of ancestor base element identification, will provide possibility of all child base element's identification.

Condition (2) guarantees backdriving absence at the child basic element output when test of element-ancestor is fed.

Condition (3) excludes the possibility of basic element attributing to the class when any of component inputs doesn't coincide with inputs set of ancestor-element, which disposes at the top of corresponding hierarchy tree. If it is possibly to include base element to more when one class then for classification implementation following basic rules is used:

- recognizing base element is included to class, which is generated by element-ancestor that disposes at the higher hierarchy level,
- if two candidates to ancestor-class, which dispose at the same hierarchy level, exist then recognizing base element will be included in more potentially powerful class,
- if the potential powers of classes, which generated by ancestor base element at the same hierarchy level, equal then recognizing base element includes in class, which includes more components at the present moment,
- if the number of components within classes equals then recognizing base element includes in any from candidates' classes.

The example given in Fig.2 illustrates the attributing of component D_k with $V^I_{i,k} = (0100)$ to one of three classes. These classes are represented by ancestors basic elements D_j, D_n and D_t with inputs vector $V_{l,j} = (0111)$, $V_{m,n} = (1100)$ and $V_{s,t} = (1110)$, respectively.

Component D_j will be included, in different situation, to the following classes:

- with the presence of ancestor basic elements $V^I_{l,j}$ and $V^I_{m,n}$ -- to the class represented by base element $V^I_{l,j}$,
- with the presence of ancestor base element $V^I_{l,j}$ and $V^I_{s,t}$ -- to the class represented by base element $V^I_{s,t}$.

Algorithm of ICT training.

We consider situation when *a priori* information about belonging of recognizing component pins to inputs and outputs sets are absent. In such situation, algorithm of ICT training includes the following basic steps:

1°. Search of test actions, which cause switching of any recognizing component pin. Then switched pins are excluded from pins set, at which test actions are fed.

2°. Identification of inputs, which belong to the same base element that have switched pin.

3°. Identification of elementary function, which is realized by base element.

4°. Try to determine component's belonging to the any class.

5°. Replenishment of base functional element's hierarchy, if it's necessary.

Simultaneously actions feeding at the *n* component pins may be needed to provide condition of pin switching. Obviously, exhaustive search of all groups in *n* component pins is practically unacceptable. Therefore, ICT training algorithm is based on idea of adaptive genetic algorithms and includes the following basic operations:

1. By turn test actions feeding at the each of component being recognized pin.
2. Feeding of sequence from *m* such test actions, that any of them is fed at *(m-1)* pins of component being recognized.
3. Feeding test sets of the same type differ by points of test action's application only. Statistical data concerning these components, which appear more often, are accounted when initial test sets are built. If feeding of all test sets are not successful, then crossing over, inversion and mutation operators are used [3]. In this case, only pin numbers of test action's application are modified, but test actions aren't changed. As a result of each test set usage, utility measure determining the probability of it further choice is calculated

Summary.

Suggested algorithms were realized for training of module in-circuit tester MTS-3 at the stage of electronic devices manufacturing. Training and identification were implemented by means of clips as well as using fixtures "bed of nails" with rearranging contact field. System training was made at the UUT including middle scale integration components. Use of suggested training algorithms provide cost decrease for library replenishment and test programs preparation and debugging on the 10-15 percent's average. The further research has been directed to improve the suggested training and identification procedures for more complex objects including analog IC and VLSI.

References.

[1] Baida N.P., Mesyura V.I., Roik A.M. Self-Learning Manufacturing Defects' Analyzers for Radio-Electronic Equipment.-Moscow: Radio and Communications, 1991. (In Russian).

[2] Baida N.P., Mesyura V.I. Method of Component Identification in Printed Circuits Units with Unknown Structure. Proceedings of the XIII IMEKO World Congress, Torino, 1994, pp. 2246-2251.