Functional aspects in the design for testability

"You will never solve a problem if you will think the same way as those who created it."
Albert Einstein

Ion COJOCARU
Universitatea Politehnica Bucure□ti
i_coj@yahoo.fr

Mariana EŞANU
S.C. CRISINVEST S.R.L, Bucure□ti
mariana_esanu@yahoo.com

Abstract -
Addressing Design for testability of digital structures requires performing complex studies of phenomena at the frontiers of scientific directions as mathematical logic, physics, chemistry, biology. Half a century passed from the moment of issue of Integrated Circuits Design for testability has not yielded the expected results, which confirms the need for a new paradigm based on the principles of the very nature of existence. Creating the design for testability paradigm involves research, analysis, and use of certain functional aspects between logical functions of different structures. This can lead to the creation of new types of elementary structures and properties intrinsic to the development of modern theories and efficient design for testability. Are highlighted and considered various functional forms of logical functions: equivalent, reverse duality complementary. These functional aspects can be considered as a small step towards establishing innovative concepts of the future transition to the new state of knowledge. The paper is based on analysis of reference works in research logic algebra functions and design for testability.

***Index Terms* — design for testability, echivalence, inversion, duality, complementarity, intrinsic properties**

I. INTRODUCTION

Rapid development of integrated circuits, their use in the manufacture of microscopes and telescopes electronics, computers, control systems lead to permanent requirements to increase the functionality of digital structures (DS). The resulting complexity integratelor greatly increase. Gordon Moore foresaw this trend properly: him in 1965 predicted that the number of transistors that can be placed on an integrated circuit (IC) would double every two years. This complexity hamper the timely performance of the verification process integratelor: necessary tests could not be generated in a timely manner or, for some defects, tests generally could not be generated.
Ongoing requirements of functionality IC growth lead to increased structural complexity and functional-logic, making it impossible to generate tests. (See: [1] Bennetts: ″In fact the only real measure of testability is the cost of generating the corresponding set of tests for the circuit″ - pag. 53; [2] G. Russell, I.L. Sayers: ″ Results have shown that test generation times increase as the square of circuit complexity, assuming that a path sensitization algorithm is used and that the amount of backtracking to resolve inconsistencies is negligible″ -pag. 15-16).
Such diagnostic technique was a new scientific direction - Design for Testing (DFT). This problem made half a century ago, has not been solved properly so far, the current state of knowledge is insufficient. This paper appears as a natural necessity of a new attempt to create a solution to the DFT, innovative Knowledge borders current state of knowledge.

I.1.Basic concepts and definitions

In the paper we use the Notions, basic definitions and notations from [3; 4]]. Some mathematical aspects are interpreted as in [5; 6; 7]. Aria logic functions (LF) used l.ogicii algebra (LA) is more extensive, this may give rise to different LF couples than those formed by Boolean operators - AND, OR, NOT. However, the Fact That the area of logical functions (LF) used in logic algebra is larger, leads to the apparition of new LF couples, Which Can not Be Realized with the Boolean functions (BF) AND, OR, NOT. This feature gives the Logic Algebra (LA) new properties of relations between LF (F_i, F_i) and (F_i, F_j) .Functional aspects describe properties that can be used to compile a database of basic digital structures as essential intrinsic properties of a new foundation of DFT. Analysis and comparison of logical and functional properties of LF and establish a DFT methods development are a priority. Obtaining encouraging results were important not only for solving the DFT. Analysis of primary structures of biological cells under these elaborations would, at least in empirical intuition of concepts to create primitive structures as a result of interaction between primary entities under the action of environmental factors.

I.2. Logic algebra

The algebra formed form the set B = {0, 1} together with all the possible operations in this set is called logic algebra (LA). A function $f(x_1,..., x_i,..., x_n)$ is said to be contained in LA (or logic function) if it, together with its arguments $x_i, i \in \overline{1,n}$, takes values from the set B={0,1} [3; 4].

Boolean algebra (BA) is an important subset of logic algebra: more specific BA is most frequently used for representing LF and conducting minimization with logic redundancy exclusion and initial form synthesis of LF, based on the operators of simple Boolean base. (SBB). As required, the LF, at the next synthesis steps, will be modified maintaining logical equivalence to be represented in mono-functional universal base AND-NOT or OR-NOT.

I.3. The functions of logic algebra

In table 1 we present the 2 variable LF $B(n) = 2^{2^n} = 2^{2^2} = 16$, whose definition domain is the ordered set of tuples $X_k = (x_0^{\sigma}, x_1^{\sigma},..., x_{n-1}^{\sigma})$, σ ∈ (0, 1), $k = \overline{0 \div 2^n - 1}$, and $F_j = F_0, F_{1,...,} F_{k,...,} F_{2^n-1}$ is the LF value domain. Here *k* represents the order number of the given tuple X_k, to which corresponds the respective LF $y_k = f(x_k)$, $k = \overline{0 \div 2^n - 1}$. Therefore there exists a two-way relation between the values of a tuple and the respective values of the LF:

$$y_k = f_k(x_0,...,x_i,...,x_{n-1}), \quad k = \overline{0 \div 2^n - 1}, \quad x_i \in \{0, 1\}$$

and $y_k \in \{0, 1\}$, (1)

Following, we will study the two-way relations between Boolean functions (BF) couples (F_i, F_j), which have the same domain definition [3; 4; 6; 7].

II. ASPECTE FUNCȚIONALE ÎN PROIECTAREA PENTRU TESTABILITATE

III.1. Algebraic system and Structures

Discovering LF relations and, especially, LF interactions, permits the scientific based determination of the elaboration of the elemental digital structure base, new logic-algebraic synthesis concepts and, finally, adequately solving the DFT problem. A plurality algebra consists of the data operations. Algebras are particular cases of algebraic systems. Algebraic systems are sets defining the operations and relationships. Algebras are particular cases of algebraic systems with empty set of relations [4]). A model is an important notion in the PPT as considering a FL is inseparable from consideration of the respective LF and logical struture is representing two facets of one and the same entity. A model is a particular case of algebraic systems, containing only crowd including relationships defined, the empty set of logical operations [4]. Lots *M* is partially ordered if there is a relationship between its elements .such ≤ . Algebraic structure is a partially ordered set *M* in which for any two elements *a* and *b* is defined intersection $a \cap b$ and union $a \cup b$. Therefore, the structure is a binary algebraic relationship system $\{M; \leq; \cap, \cup\}$ with a binary relation and two binary operations.

II.2. Binary relations in logical Algebra

Next we study binary relations between couples LF (F_i, F_j) , with the same field definition on the set *M*.

Crowds out the operations, data and relationships are called algebraic systems. Setting theoretical intuition along with empirical relationships and interdependencies between LF are essential in determining scientifically basic digital structures, new algebraic concepts logical synthesis and proper settlement of the issue DFT.

Binary relations can be established between two logical functions whose direct forms:

a) coincide (LF (F_i, F_i) have the same origin) ;.

b) do not coincide (LF (F_i, F_j) have the same origin).

To determine the type of logical relations between the two forms of representation of the same logic function, they must be defined in the same area have the same arity n (and cardinality). Next we use some definitions and approaches of [3], which is currently in the most appropriate manner, basic definitions, relations between different forms of representation of LF and interpretation comparison as required DFT practice.

Be given LF $f_1(x_1, x_2,..., x_n)$. LF $f_2(x_1, x_2,..., x_n)$, receives a value of 0, when f_1 is equal to 1, and the value 1 when f_1 is equal to 0 is called the inverse function tool f_1 and is denoted by $\overline{f_1}$. According to the definition f_1 $(\overline{\overline{f_1}} \equiv f_1)$. Reverse relationship between LF is mean by

$$f_1 \overset{I}{=} \overline{f_1} \quad , \qquad (2).$$

LF $f_1(x_1, x_2,..., x_n)$ and $f_2(x_1, x_2,..., x_n)$ are called dual if

$$f_1(x_1, x_2,..., x_n) \overset{D}{=} f_2(\overline{x}_1, \overline{x}_2,..., \overline{x}_n), \qquad (3)$$

and also logical symbols &, □ of LF f_1 are replaced each f_2 with the logical symbols □, &.

Equivalence relation is denoted by the symbol $\overset{E}{=}$.

For example, $f_1(x_1, x_2,..., x_n) \overset{E}{=} \overline{f_2(\overline{x_1}, \overline{x_2},..., \overline{x_3})}$, (4)

So, reversing a scalar dual expression leads to the original position. For example, $a \cdot b \cdot c \overset{E}{=} \overline{(\overline{a} \vee \overline{b} \vee \overline{c}}$.

Therefore, the reversal of the dual scalar expression leads to the initial position. For example, the relationship of complementarity $\overset{C}{=}$ constitute a complex relationship between two different LF. The relationship of complementarity

constitutes a complex relationship between two different LF. These symbols have the meaning of " are equivalent" "are inverse", "are dual", " are complementary" and the symbols $\overset{E}{\neq}, \overset{I}{\neq}, \overset{D}{\neq}, \overset{C}{\neq}$ means that those relationships do not occur.

Table 1. Binary functions in positive logic

x_1	x_2	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Table 2. Binary functions in negative logic

x_1	x_2	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
0	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

Table 3. Karnaugh diagrams and Normal forms for LF representation

a \ bc	00	01	11	10
0	0	0	0	0
1	0	0	1	0

Disjunctive form $F_1^D = a\cdot b\cdot c$, (1.1)

Inverse disjunctive form $\overline{F}_1^D = \overline{a}\vee\overline{b}\vee\overline{c} = \overline{a\cdot b\cdot c}$, (1.2)

Conjunctive form $F_1^C = (a)\cdot(b)\cdot(c)$, (1.3)

Inverse conjunctive form $\overline{F}_1^C = (\overline{a})\vee(\overline{b})\vee(\overline{c})$, (1.4)

a \ bc	00	01	11	10
0	0	1	1	1
1	1	1	1	1

Disjunctive form $F_2^D = a\vee b\vee c$, (2.1)

Inverse disjunctive form $\overline{F}_2^D = \overline{a}\cdot\overline{b}\cdot\overline{c} = \overline{a\vee b\vee c}$, (2.2)

Conjunctive form $F_2^C = (a)\vee(b)\vee(c)$, (2.3)

Inverse conjunctive form $\overline{F}_2^C = \overline{a}\cdot\overline{b}\cdot\overline{c} = \overline{a\vee b\vee c}$, (2.4)

a \ bc	00	01	11	10
0	1	1	1	1
1	1	1	0	1

Disjunctive form $F_3^D = \overline{a}\vee\overline{b}\vee\overline{c} = \overline{a\cdot b\cdot c}$, (3.1)

Inverse disjunctive form $\overline{F}_3^D = a\cdot b\cdot c$, (3.2)

Conjunctive form $F_3^C = \overline{a}\vee\overline{b}\vee\overline{c} = \overline{a\cdot b\cdot c}$, (3.3)

Inverse conjunctive form $\overline{F}_3^C = a\cdot b\cdot c$, (3.4)

a \ bc	00	01	11	10
0	1	0	0	0
1	0	0	0	0

Disjunctive form $F_4^D = \overline{a}\cdot\overline{b}\cdot\overline{c} = \overline{a\vee b\vee c}$, (4.1)

Inverse disjunctive form $\overline{F}_4^D = a\vee b\vee c$, (4.2)

Conjunctive form $F_4^C = (\overline{a})\cdot(\overline{b})\cdot(\overline{c}) = \overline{a\vee b\vee c}$, (4.3)

Inverse conjunctive form $\overline{F}_4^C = a\vee b\vee c$, (4.4)

a \ bc	00	01	11	10
0	1	0	0	0
1	0	0	0	0

Disjunctive form $F_5^D = \overline{a}\cdot\overline{b}\cdot\overline{c}$, (5.1)

Inverse disjunctive form $\overline{F}_5^D = a\vee b\vee c$, (5.2)

Conjunctive form $F_5^C = (\overline{a})\cdot(\overline{b})\cdot(\overline{c})$, (5.3)

Inverse conjunctive form $\overline{F}_5^C = a\vee b\vee c$, (5.4)

a \ bc	00	01	11	10
0	1	1	1	1
1	1	1	0	1

Disjunctive form $F_6^D = \overline{a}\vee\overline{b}\vee\overline{c}$, (6.1)

Inverse disjunctive form $\overline{F}_6^D = a\cdot b\cdot c$, (6.2)

Conjunctive form $F_6^C = \overline{a}\vee\overline{b}\vee\overline{c}$, (6.3)

Inverse conjunctive form $\overline{F}_6^C = a\cdot b\cdot c$, (6.4)

a \ bc	00	01	11	10
0	0	1	1	1
1	1	1	1	1

Disjunctive form $F_7^D = a\vee b\vee c$, (7.1)

Inverse disjunctive form $\overline{F}_7^D = \overline{a}\cdot\overline{b}\cdot\overline{c} = \overline{a\vee b\vee c}$, (7.2)

Conjunctive form $F_7^C = (a)\vee(b)\vee(c)$, (7.3)

Inverse conjunctive form $\overline{F}_7^C = \overline{a}\cdot\overline{b}\cdot\overline{c} = \overline{a\vee b\vee c}$, (7.4)

a \ bc	00	01	11	10
0	0	0	0	0
1	0	0	1	0

Disjunctive form $F_8^D = a\cdot b\cdot c$, (8.1)

Inverse disjunctive form $\overline{F}_8^D = \overline{a}\vee\overline{b}\vee\overline{c} = \overline{a\cdot b\cdot c}$, (8.2)

Conjunctive form $F_8^C = (a)\cdot(b)\cdot(c)$, (8.3)

Inverse conjunctive form $\overline{F}_8^C = (\overline{a})\vee(\overline{b})\vee(\overline{c})$, (8.4)

Investigate the relationships between various couples of FL given in table 3 can be highlighted FL couples the named properties. Table 3 the indices D and C have the meaning of disjunctive, respectively conjunctive.

Table 4 allows to emphasize the following types of binary relations between different forms of representation of given logic function couples:

1)equivalence: $F_{1norm} \;/\; F_2^{\,C}$, $F_{1norm} \;/\; \overline{F_1^{\,D}}$;

$F_{2\,norm} \;/\; F_1^{\,C}$; $F_{2\,norm} \;/\; \overline{F_2^{\,D}}$;

2) inversion: $F_{1\,norm} \;/\; F_1^{\,I}$; $F_{2\,norm} \;/\; F_2^{\,I}$;

3) duality: $F_{1\,norm} \;/\; F_2^{\,D}$; $F_{2\,norm} \;/\; F_1^{\,D}$;

4)complementarity:

F_{1norm} / F_{2norm} ; F_{1norm} / F_1^C ; F_{2norm} / F_2^C ; $F_{2norm} / \overline{F}_1^D$;

.

Studying LF couples relationships is paramount in terms of logical analysis of primary structures.Investigate the relationships between various couples of FL given in Table 3 can be highlighted FL couples the named properties. Table 3 indices D and C have the meaning of disjunctive, respectively disorders. Table 4 allows to emphasize the following types of binary relations between different forms of representation of FL torque data:

1) equivalence;
2) reversal;
3) duality;
4) complementarity.

Studying FL couples relationships is paramount in terms of logical analysis of primary structures

Table 4. Binary relations between different logic functions

Tuples	F_{1norm}	F_{2norm}	F_1^I	F_2^I	F_1^D	F_2^D	F_1^C	F_2^C	$\overline{F}_1^D$	$\overline{F}_2^D$
	$a\cdot b\cdot c$	$a\vee b\vee c$	$\overline{a\cdot b\cdot c}$	$\overline{a\vee b\vee c}$	$\bar{a}\cdot\bar{b}\cdot\bar{c}$	$\bar{a}\vee\bar{b}\vee\bar{c}$	$a\vee b\vee c$	$a\cdot b\cdot c$	$\overline{\bar{a}\vee\bar{b}\vee\bar{c}} = a\cdot b\cdot c$	$\overline{\bar{a}\cdot\bar{b}\cdot\bar{c}} = a\vee b\vee c$
‹000›	0	0	1	1	1	1	0	0	0	0
‹001›	0	1	1	0	0	1	1	0	0	1
‹010›	0	1	1	0	0	1	1	0	0	1
‹011›	0	1	1	0	0	1	1	0	0	1
‹100›	0	1	1	0	0	1	1	0	0	1
‹101›	0	1	1	0	0	1	1	0	0	1
‹110›	0	1	1	0	0	1	1	0	0	1
‹111›	1	1	0	0	0	0	1	1	1	1

III.1. Conclusions

Detailed analysis of the issues of equivalence, inversion, duality and complementarity of SD is needed to highlight the quintessence energy fizică- chemical-logic and evolution of the phenomenon occurring structures on these properties. From this point of view is important not only mutual influence of the whole structure / function as two sides of one and the same entity. A major problem is the influence of the physical-logical environment. Addressing DFT constitutes a stumbling block, and also a challenge to address these primary issues of current science.

Problema DFT dezvăluie □i necesitatea schimbării modului de abordare a cecetării □tiin□ifice actuale: complexitatea problemelor nesolu□ionate actualmente, dar □i a viitoarelor provocări □tiin□ifice poate fi depă□ită mai degrabă de echipe □i mai pu□in de cercetători individuali.

DFT problem reveals the need to change the approach to current scientific research: complexity of currently unsolved problems, and future scientific challenges can be overcome rather than teams and individual researchers.

IV. REFERENCES

[1] R. G. Bennetts. Design of Testable Logic Circuits, (Brunel/Southhampton Universities), ISBN 0-201-14403, (1995)

[2]. G. Russell, I.L. Sayers. Advanced Simulation and Test Methodologies for VLSI Design, 1989. ISBN 0 7476 0001 5.

[3]. K. Kinosita, K. Asada, O. Karatsu. Logical design of VLSI. – Loghiceskoe proiectirovanie SBIS – Iwanami Shoten Publishers, Tokyo, 1985, 304 pag. Traducere în limba rusă Translation into Russian , 1988.

[4]. Кузнецов О. П., Адельсон-Вельский Г. М. Дискретная математика для инженера Discrete mathematics for engineers. – М.: Энергия, 1980 г., с.344.

[5]. .J. Borowski and J. M. Borwein. Collins dictionary of Mathematics ,2002

[6].James A. Anderson. Discrete Mathematics with Combinatorics. – Prentice Hall. University of South Carolina, Spartanburg, 2004. -960 p.

[7]. Rod Haggarty. Discrete Mathematics for computing. – Edition published by Pearson Education Limited, Harlow, UK, 2002. – 317 p.

[8]. M. Rafiquzzaman. Fundamentals of Digital Logic and Microcomputer Design.- Fifth Edition, California, WILEY INTERSCIENCE – John Wiley&Sons inc., 2005, 840 pag

[9]. Wakerly J. F. Circuite digitale. Principiile şi practicile folosite în proiectare, Bucureşti: Teora Digital circuits. Principles and practices used in the design, Bucharest: Teora,, – 2002. – 928 p. (Traducere în limba română Romanian translation)

[10]. Wilkinson B. Digital System Design. – 4nd edition, Prentice Hall, London, 2004, 320 p.

[11]. Holdsworth B. Digital Logic Design. – Butterworth & Co (Publishers), LTD, London, 1982, 285 p.