Study of Binary Relations of Boolean Algebra Functions

Ion COJOCARU, Octavian IONEL
The University "Politehnica" Bucharest,
Spl. Independenţei 313, RO 077206, Faculty of Engineering in Foreign Languages, Bucharest
i_coj@yahoo.fr

Abstract — **In the process of digital circuits (DC) design, the logical functions (LF) are initialy represented in the boolean simple base (BSB) of the Boolean algebra (BA), which allows checking the correctitude of representing the minimal shape of the LF and the absence of the logical redundance. The usage of only 3 boolean operators - AND, OR, NOT, is imposing some restrictions in the process of synthesis, because any LF must be represented by the operator AND or OR: without negations, with negation only at the input or only at the output, with negations on the arbitrary connexions. Also, the main resources of minimization (for example, Karnaugh diagrams) and the synthesis procedures are most developped for the LF represented in BSB. The originality of this study stands in making, for the first time, a complete analysis and a mathematical description of the logical binary relations (LBR) of the LF in the BA. The purpose of the paper is to detect the LBR properties which are important for formalizying the synthesizing algorithms, the concepts development, digital structures and know-how needed to obtain adequate solutions of the design problem and to increase the technological efficiency of making the actual integrated circuits.**

Index Terms — **boolean algebra, technological efficiency, equivalence, complement, duality.**

I. I.NOTIONS AND BASIC DEFINITIONS

A. Boolean algebra

The algebra, formed by the set B={0,1} together with all possible operations in this set is called algebra of logic (AL). The algebra of logic function (or logical function) is called the function $f(x_1,..., x_i,..., x_n)$, which, like any of it's arguments $x_i, i \in \overline{1,n}$, takes values from the set B={0,1}. [1].

The algebra (P_2; $\wedge$, $\vee$, $^{-}$) - the basic set which is constituated from the complete set of logical functions P_2, the binary operations (conjonction $\wedge$, disjunction $\vee$ and the negation $^{-}$) – is called Boolean algebra [2]. The elements of Boolean algebra are: the boolean constants, boolean arguments and the boolean operations. The boolean arguments are replaced with the boolean constants. The operations of Boolean algebra – $\wedge$, $\vee$, $^{-}$ – are called boolean functions (BF). In BA, any logical function can be represented by many equivalent boolean formulas, which utilize only the operations of boolean algebra $\wedge$, $\vee$, $^{-}$, the signs of arguments and the parantheses.

Boolean algebra is the algebric system, made from the set B={a, b, c, ...} and boolean binary operators of 2 types: - $\wedge$, $\vee$ and the unar operator $^{-}$, in which there are satisfied the following axiomes [1]:

1) $a \vee b = b \vee a$, $a \wedge b = b \wedge a$ for any a, b $\in$ B (comutativity);

2) $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$, $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$ for any a, b, c $\in$ B (distributivite);

3) it exists $1 \in B$ and $0 \in B$, so $a \vee 0 = a$, $a \wedge 1 = a$ for any $a \in B$ (unar elements);

4) it exists $\overline{a} \in B$ so $a \vee \overline{a} = 1$, $a \wedge \overline{a} = 0$ for any $a \in B$ (the third is excluded)

With this axioms there can be demonstrated formally the basis theorems of BA.

The axioms and theorems are given in couples. Any expression from the left side of the couple means a true boolean statement which contains, generally, the disjunction (the basis operation), the conjonction (the simetric operation), the logic constant 1 and also paranthesis, and the respective expression from the right side represents also a true boolean statement, which contains the base conjonction (the basis operation), the disjonction (the simetrical operation), the logical constant 0 and paranthesis. Between these 2 expressions of the couple of each axiom there is a relation of symmetry: if in the boolean expression of an axiom from left (right) side there will be made the replacements ($\vee \leftrightarrow \wedge$), ($1 \leftrightarrow 0$), then there will be obtained the true boolean expression of the axiom from the right (left) side. This *phenomen* is known in BA as *principle of duality*[1]. The advantage of this *principle of duality*, allows, when we have to make the demonstration of some dual theorems, to be enough to prove only one of them: the principle of duality guarantees the demonstration of dual theorem automatically. Still, it is another type of *relation of duality*[2] of the LF, that fits better with this denomination and which [3] has a bigger importance in projecting digital circuits (DC).

B. The functions of Boolean algebra

The boolean function $f(x_0, \ldots x_{i-1}, x_i, x_{i+1}, \ldots, x_{n-1})$ is the image of elements (points) of boolean space n $\{0, 1\}^n$ in the set of elements of binary arguments {0, 1}. The elements of binary space n $\{0, 1\}^n$ represents the elements of the direct product (cartesian)

$$\{0, 1\} \times \{0, 1\} \times \ldots \times \{0, 1\} \quad (1)$$

in which {0, 1} enters n times. For example, the boolean function of 2 arguments can be represented in the two dimensional (2D) boolean space like $\{0, 1\}^2$, which means that the tuples are elements of the cartesian multiplication:

$$\{0, 1\} \times \{0, 1\} = \{(0, 0), (0, 1), (1, 0), (1, 1)\} \quad (2)$$

[1] The duality concept was introduced in the logical algebra by E. Schröder (1841-1902)

[2] Jules Henri Poincaré (1854-1912) - "Mathematics is the art of giving the same name to different things"

A *tuple* is an *ordered characteristic binary vector (OCBV)*, each element having a component (coordinate) enumerated from left to right. The coordinates of a tuple take values from the set

$\sigma \in \{0, 1\}$. The number of coordinates of the tuple represents the length or the dimension of the tuple. For example, the tuple:

$X_k = (x_0^\sigma, x_1^\sigma, ..., x_{n-1}^\sigma)$, $\sigma \in (0, 1)$ (3)

contains n coordinates, each taking values from the set $\sigma \in \{0, 1\}$, and k means the number of order of the given tuple X_k, to which it corresponds the respective LF

f_k, $k = \overline{0 \div 2^n - 1}$. The cardinal of tuples number with n coordinates is

$|K| = 2n$, (4)

The tuples are arranged in ascending order (lexicographic).

Two complements $(a_0, ..., a_{n-1})$ and $(b_0, ..., b_{n-1})$ are equal on the sets $A_1, ..., A_n$, if $a_i = b_i$, $i = \overline{0, n-1}$

The boolean variable x_i corresponds to coordinate i of the boolean space $\{0, 1\}^n \rightarrow \{0, 1\}$

It results that the BF and also its arguments can be equal to 0 or to 1:

$y_k = f_k(x_0,...,x_i,...,x_{n-1})$, $k = \overline{0 \div 2^n - 1}$, $x_i \in \{0, 1\}$ and $y_k \in \{0, 1\}$, (5)

To define a boolean function means putting in correspondance the values of the tuples of arguments

$X_k = (x_0, x_1, \ldots, x_{n-1})$, $k = \overline{0 \div 2^n - 1}$ (6)

with the values of the respective function

$F_j = F_0, F_{1,...,} F_{k,...,} F_{2^n-1}$, (7)

1.3. Some properties of the logical signals, logical gates and logical circuits.

In the process of design of the digital circuits in BA it appears the need of making some interdisciplinar studies for the establishment of some properties and also of some possible interactions of the logical signals, of logical gates (LG) AND, OR, NOT and of logical structures utilized in the purpose of making the BA functions. This specific, related to the utilization of some boolean operators AND, OR, NOT for the synthese of the logical functions (LF), is imposing the establishment of some rules or restrictions, usage that contribute to the growth of efficiency of the design process of DC. Also, it is important the study and the establishment of some types of logic structures, usage that can ease the design and diagnostic of errors already in the adjustment of technological process of fabrication of the integrated circuits (IC).

C. 1.3.1. The properties of logic signals

The values of the logic signals utilized in digital circuits are dependent of the choosen convention of logic – pozitive or negative. A digital circuits is designed to function onlu in one of these logics, provided already at the design level. A digital signal is represented by 2 logic levels. In the convention of positive logic (PL) the high voltage (High) corresponds to the logical signal 1, and the low voltage (Low) corresponds to the logical signal 0. Also, the active level in positive logic corresponds to the logical signal 1. The pozitive logic is mostly used in Europe and USA and also in Russia. As what regards to the negative logic (NL), the low voltage (Low) corresponds to the logic 1, and high voltage (High) to logic signal 0. Also, the active level in NL corresponds to the logical signal 0. The negative logic is mostly used in Japan, in the countries from Asia and also in Russia. The main characteristics of the logical signals are: amplitude, rise time and fall time, period of the rise front, the period of the fall front, the time for propagation through the gate.

D. 1.3.2. The properties of logic gates

The LF are realized in digital circuits using appropriate logic gates (LG). Between the logical values (LV) of the input signals of a gate and the output signal there are some relations. For example, the logical values 0 applied to only one input of the gate AND (NAND) determines uniquely the apparition of logic values 0 (1) at the output of the gate. Similarly, the LV 1 applied to a single input of the gate OR (NOR) determines uniquely the apparition of logical values 1 (0) at the output of the gate.

The LV of the input signal, which determines by itself uniquely the LV at the output of the gate, no matter of the values of the signals from the other inputs of the gate, is called *dominant logical value (of blocking).* We will designate the dominant logic value through the connexion i by: $^i d$. Is very significativ that a single dominant signal blocks the logic gate, no matter of the LV of signals from the other inputs. Also, the change of respective input signal with a homogene logic signal will unblock the gate, so LV at the output will change to oppsosite and so being possible the sensibilization of all paths through this gate.

The unique logic value of an input signal of a gate, which differs from the all other logic values at the output of this gate is called *homogene logical value (particular)*.

For the gate AND (NAND) the *homogene logical value* of the input signal is 1 (0), that being made by applyment to an order*ed characteristic binary vector - homogene tuple* 11..1. Similary, for gate OR (NOR) the *homogene logical value* of the output signal is 0 (1), that being made after applying to a *ordered characteristic binary vector – homogene tuple* 00...0. So, the LV that the signal have in the homogene tuple, is called the *homogene value of the input signal* for the respective LG. We will designate the logical homogene value on the connexion i by: $^i e$. The particularity of the homogene tuple is that it allows the activation of all paths from the inputs of LG through its output.

A homogene tuple has the following properties*:*

1) each component of the *homogene tuple ee...e*

represents a potential logical value of manifesting the errors of type *i≡d,* while the other components *ee...e* of the homogene complement ensuring unique propagation of the correct / erronated signal through the output of the gate;

2) the *homogene tuple ee...e* has maximum capacity of

detection: can detect the singular errors of type *i≡d* of the signal of any input connexion – so only one characteristic vector can detect the errors of type *i≡d* of all inputs and output of the logical gate;

3) the *homogene tuple* detects the presence of one or

many erors at the inputs of the gate, but it does not find a certain defected input. This is a property of the logical gates and there are no modalities, excepting those related to the modification of structures (auxiliar control points or auxiliar logic) to overpass this situation. The errors detected by the homogene tuple of the signals are called equivalent errors.

As for the exclusive gates OR (XOR), exclusive OR-NOT (XNOR), imparity, parity, these represents the special functions and can have as value, as a homogene logical value, any from the logical values 0 or 1.

One of the properties of the LF f_1, f_6, f_7, f_8, f_9 şi f_{14} (tab. 1) is the comutativity [10], that means the possibilty of changing the places of input variables without affecting the result. Also, it is necessary the study of properties of the LF for respective LG from the point of view of detecting some useful properties within Design For Testability (DFT).

E. 1.3.3. The properties of the logical functions and of the digital structure

A LF corresponds to a law of transformation of the input LV into output LV. Digital circuits are designed in such a way to guarantee a bilateral correspondance between, - the primary variables of the LF, boolean operators, intermediare variables (functions) and output, - as elements of the formal process of synthese, one one side, and the connexions of primary inputs, logical gates, digital circuits and intermediar connexions and of output, as component parts of the DC, on the other side. A *digital structure*, as a synthese object, represents a *graphical representation* of the input and output connexions and of diverse types of logical gates, interconnected in a specific way to obtain at the output a certain LF, which is needed for a comprehensive documentation also for the designers and also for engineers.

A LG makes the respective logical function. In the general case, the functional properties of a logical structure represents the result of the interaction of properties of different types of LG. That's why, a logical structure represents the result of the synthese, consisting in certain interactions of the LG and interconnexions of those made in such a way to guarantee the establishment of the appropiate logical function.

A homogene DC (for example, made of a singel type of gates OR or AND) or an alternative circuit (of gates NAND/NOR or gates NOR/NAND with an even number of logical levels and even parities) degenerates in the equivalent LG, being no function interactions between the gates which could change the logical function. The logical structures that are maximal degenerated are made from a single type of LG and thus, the homogenity of the resulting LG assures the same logical properties to the structure maximal degenerated, equivalent to the LG of same type and with the same numer of inputs. The homogenity leads to the possibility of diagnosis of constant defects of the equivalent gates by applying only (n+1) standard tests.

The usage of operations AND, OR, NOT in the BA for making any LF leads to the appearance of some particularities of the LF relations in the boolean algebra. For example, each of the LF NAND or NOR in the actual technologies are made from a single LF, while their establishment in the BA needs the utilization of 2 logical operations: AND-NOT or, respectively OR-NOT.

II. II. STUDY OF BINARY RELATIONS OF BOOLEAN ALGEBRA FUNCTIONS

A. II.1 Signification of binary relations of logical functions

The process of design of digital structures presupposes the succesive modification of some logical functions and the replacement of some logical formulas with other equivalent formulas. From table 2 it can be observed that the disjunctive and conjunctive forms of the same LF are the same. Also, excepting the relation of equivalence, a big importance has the relations of complementarity, inversation (negation) and duality of the LF (tabel 2). In this context is necessary the establishment of semnification of this notions.

Equivalent – which has the same value, same effect, same semnification with other thing (explicative dictionary of romanian language)

Dual – the rapport of 2 properties, principles, elements, objects, humans of opposite sex, which usually form a couple, a pair (explicative dictionary of romanian language)

Complementary – property of a couple, object or phenomen to have 2 reciproc types of values, dimensions or oposite shapes, which match each other, are well fitted together (explicative dictionary of romanian language).

Also, it appears the necessity of obtaining and utilization the relations of equivalence, inversation (negation), complementarity, partial complementarity. These relations not always are the same for the representations of LF in the boolean algebra and in the algebra of logics, which is explained by the fact, that in BA there are utilized only logic operators AND, OR, NO for representing each of the LF. By the signs $\stackrel{C}{=}$, $\stackrel{D}{=}$, $\stackrel{E}{=}$, $\stackrel{CP}{=}$, there are indicated, respectively, the binary relations for complement, duality, equivalence and partial complement (table 3).

A great importance has the relations between the couples of LF represented in the positive logic and respectively, in the negative logic, which lead to the term of duality of LF, even if an equipment or digital system is made, regularly, only in one of these logics. The LF can be compared only in the case of same tuples of binary arguments, i.e. of the same *definition domain*. The analysis of most used LF is given in table 2, in which each LF is described by a Karnaugh diagram and inverse. It can be observed the coincidence of the final logical expressions of a disjunctive form and of conjunctive form, also of final logical expressions of the inverse disjunctive form and of the inverse conjunctive form. Also, the inverse forms of any of the BF AND, OR have each 2 equivalent modes of representation: complementary – with the inversion of the LF – and dual - with the inversion of arguments, which are made by inserting invertors on the connexions of the primary inputs. From these we take the following conclusions.

Table 1. Logical function

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. Karnaugh diagrams of the LF and equivalent formula for their representation

a\b: 0 1 0: 0 0 1: 0 1	Disjunctive form: Inverse disjunctive form: Conjunctive form: Inverse conjunctive form:	$F_1^D = a\cdot b$, $\overline{F}_1^D = \overline{a}\cdot\overline{b} \vee \overline{a}\cdot b \vee a\cdot\overline{b} = \overline{a}\vee\overline{b} = \overline{a\cdot b}$, $F_1^C = (a\vee b)(a\vee\overline{b})(\overline{a}\vee b) = a\cdot b$, $\overline{F}_1^C = \overline{a}\vee\overline{b} = \overline{a\cdot b}$,	(1.1) (1.2) (1.3) (1.4)
a\b: 0 1 0: 0 1 1: 1 1	Disjunctive form: Inverse disjunctive form: Conjunctive form: Inverse conjunctive form:	$F_2^D = \overline{a}\cdot b \vee a\cdot\overline{b} \vee a\cdot b = a\vee b$, $\overline{F}_2^D = \overline{a}\cdot\overline{b} = \overline{a\vee b}$, $F_2^C = a\vee b$, $\overline{F}_2^C = (a\vee\overline{b})\cdot(\overline{a}\vee b)\cdot(\overline{a}\vee\overline{b}) = \overline{a}\cdot\overline{b} = \overline{a\vee b}$,	(2.1) (2.2) (2.3) (2.4)
a\b: 0 1 0: 1 1 1: 1 0	Disjunctive form: Inverse disjunctive form: Conjunctive form: Inverse conjunctive form:	$F_3^D = \overline{a}\cdot\overline{b} \vee \overline{a}\cdot b \vee a\cdot\overline{b} = \overline{a}\vee\overline{b} = \overline{a\cdot b}$, $\overline{F}_3^D = a\cdot b$, $F_3^C = \overline{a}\vee\overline{b} = \overline{a\cdot b}$, $\overline{F}_3^C = (a\vee b)(a\vee\overline{b})(\overline{a}\vee b) = a\cdot b$,	(3.1) (3.2) (3.3) (3.4)
a\b: 0 1 0: 1 0 1: 0 0	Disjunctive form: Inverse disjunctive form: Conjunctive form: Inverse conjunctive form:	$F_4^D = \overline{a}\cdot\overline{b} = \overline{a\vee b}$, $\overline{F}_4^D = \overline{a}\cdot b \vee a\cdot\overline{b} \vee a\cdot b = a\vee b$, $F_4^C = (a\vee\overline{b})\cdot(\overline{a}\vee b)\cdot(\overline{a}\vee\overline{b}) = \overline{a}\cdot\overline{b} = \overline{a\vee b}$, $\overline{F}_4^C = a\vee b$,	(4.1) (4.2) (4.3) (4.4)
a\b: 0 1 0: 0 1 1: 1 0	Disjunctive form: Inverse disjunctive form: Forma conjunctivă: Inverse conjunctive form:	$F_5^D = a\cdot\overline{b} \vee \overline{a}\cdot b$, $\overline{F}_5^D = \overline{a}\cdot\overline{b} \vee a\cdot b$, $F_5^C = (a\vee b)\cdot(\overline{a}\vee\overline{b}) = a\cdot\overline{b} \vee \overline{a}\cdot b$, $\overline{F}_5^C = (a\vee\overline{b})\cdot(\overline{a}\vee b) = a\cdot b \vee \overline{a}\cdot\overline{b}$,	(5.1) (5.2) (5.3) (5.4)
a\b: 0 1 0: 1 0 1: 0 1	Disjunctive form: Inverse disjunctive form: Forma conjunctivă: Inverse conjunctive form:	$F_6^D = \overline{a}\cdot\overline{b} \vee a\cdot b$, $\overline{F}_6^D = a\cdot\overline{b} \vee \overline{a}\cdot b$, $F_6^C = (a\vee\overline{b})\cdot(\overline{a}\vee b) = a\cdot b \vee \overline{a}\cdot\overline{b}$, $\overline{F}_6^C = (a\vee b)\cdot(\overline{a}\vee\overline{b}) = a\cdot\overline{b} \vee \overline{a}\cdot b$,	(6.1) (6.2) (6.3) (6.4)

Table 3. Couples of LF: complementary, dual, equivalent or partial complementary

Tyoe of relation between the LF of the couples	The logical functions and the analytical representation of relation between the LF			
	1	2	3	4
Complement	AND/AND-NOT $a\cdot b \overset{C}{=} \overline{a\cdot b}$	OR/OR-NOT $a\vee b \overset{C}{=} \overline{a\vee b}$	AND-NOT/AND $\overline{a\cdot b} \overset{C}{=} a\cdot b$	NOT-OR-NOT/NOT-AND-NOT $\overline{a\vee b} \overset{C}{=} \overline{\overline{a}\cdot\overline{b}}$
Duality	AND/NOT-OR $a\cdot b \overset{D}{=} \overline{a}\vee\overline{b}$	OR/NOT-AND $a\vee b \overset{D}{=} \overline{a}\cdot\overline{b}$	AND-NOT/NOT-OR-NOT $\overline{a\cdot b} \overset{D}{=} \overline{\overline{a}\vee\overline{b}}$	OR-NOT/NOT-AND-NOT $\overline{a\vee b} \overset{D}{=} \overline{\overline{a}\cdot\overline{b}}$
Equivalence	AND/NOT-OR-NOT $a\cdot b \overset{E}{=} \overline{\overline{a}\vee\overline{b}}$	OR/NOT-AND-NOT $a\vee b \overset{E}{=} \overline{\overline{a}\cdot\overline{b}}$	AND-NOT/NOT-OR $\overline{a\cdot b} \overset{E}{=} \overline{a}\vee\overline{b}$	OR-NOT/NOT-AND $\overline{a\vee b} \overset{E}{=} \overline{a}\cdot\overline{b}$
Partial complement	AND/OR $a\cdot b \overset{CP}{=} a\vee b$	OR-NOT/AND-NOT $\overline{a\vee b} \overset{CP}{=} \overline{a\cdot b}$	NOT-AND/NOT-OR $\overline{a}\cdot\overline{b} \overset{CP}{=} \overline{a}\vee\overline{b}$	NOT-OR-NOT/NOT-AND-NOT $\overline{\overline{a}\vee\overline{b}} \overset{CP}{=} \overline{\overline{a}\cdot\overline{b}}$

1. The equality of the dual expression and of the respective complementary expression confirms the laws of De Morgan;

2. The dual form and the complementary form constitues 2 equivalent logical formulas of the same inverse LF (and not diverse logical functions, as it is shown in [8, 9, 10, 11]). Also, the obtained result confirms the results from [1, 7, 12, 13]. For example, in [1], the definition of the dual expression is the following:

„*The expression obtained by reciprocal replacing in a certain boolean expression of the variable* $a \leftrightarrow \overline{a}$, *0↔1,* $\wedge \leftrightarrow \vee$ *(the logical multiplication with the logical sum) is called dual expression to the initial dual boolean expression. It is necessary the indication with the help of the paranthesis of the order of making the operations. For example, the dual expression for* $a \wedge b \vee c$ *will not be* $\overline{a} \vee \overline{b} \wedge \overline{c}$, *but* $(\overline{a} \vee \overline{b}) \wedge \overline{c}$". From here it results that the notion of duality of the BF is reffering to a certain stage of representation of the LF in the process of design of the digital structure and is based on utilizing the same definition domain of the variables of LF of the considered couple, meanwhile the duality notion of the expressions of the axioms of boolean logic is based on different definition domains – initial for an axiom of the couple and dual for the other axiom[2].

3. It exists complementary relations [3, 4, 5, 6] between the representations of BF AND/AND-NOT, BF OR/OR-NOT and BF XOR/XNOR (the last 2 function are equivalent to the Imparity($\oplus$)Parity ($\overline{\oplus}$) functions only in the case of 2 logical variables.

In the purpose of aprofundating the analysis of binary relations of the LF, we will consider the main commutative LF in the case of tuples with length of n=2 (table 4).

The analysis of LF $F_1 \div F_{10}$ allows for finding the couples of functions of equivalent logic, complementary, dual or partial complementary. So, the following LF couples are:

1) equivalent: AND/ NOT-OR-NOT; OR/ NOT-AND-NOT; AND-NOT/ NOT-OR; OR-NOT/ NOT-AND;

2) complementary: AND/ AND-NOT; OR/ OR-NOT; AND-NOT/ NOT-OR-NOT; OR-NOT/ NOT-AND-NOT; IMPARITY/ PARITY;

3) dual: AND/ NOT-OR; OR/ NOT-AND; NOT-OR/ NOT-OR-NOT; NOT-AND/ NOT-AND-NOT;

4) partial complementary: AND/ OR; OR/ AND; AND-NOT/ OR-NOT; OR-NOT/ AND-NOT; NOT-AND/ NOT-OR; NOT-OR/ NOT-AND; NOT-AND-NOT/ NOT-OR-NOT; NOT-OR-NOT/ NOT-AND-NOT.

The partial complementary BF couples have a special importance, and also, the boolean functions of these couples are not commutative. The commutativity of LF has a big role in design of DC, but is also important the BF non-commutativity of couples partialy complementar in designing for testability.

Table 4. Boolean functions with possible binary relations

cortegiu		F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}
$\langle X_i \rangle = \langle \tilde{x}_0 \tilde{x}_1 \rangle$		AND	OR	AND-NOT	OR-NOT	NOT-AND	NOT-OR	NOT-AND-NOT	NOT-OR-NOT	IMPARITY $\oplus$	PARITY $\overline{\oplus}$
$\langle X_0 \rangle$	0 0	0	0	1	1	1	1	0	0	0	1
$\langle X_1 \rangle$	0 1	0	1	1	0	0	1	1	0	1	0
$\langle X_2 \rangle$	1 0	0	1	1	0	0	1	1	0	1	0
$\langle X_3 \rangle$	1 1	1	1	0	0	0	0	1	1	0	1

Still, to illustrate the particularities of relations of equivalence, complementary, duality and partial complementary, but also the properties that unify or separate these binary relations of the BF is useful the analysis of these relations on the basis of an example of combinational circuit (CC). The interraction of the BF of a CC contributes to a appropriate comprehension regarding the aspects of the studied relations.

B. II.2. The binary relations of the logical functions realized by the combinational circuits without fan-out

Being given the initial function from the figure 1,a. By negation this function will be obtained the complementary

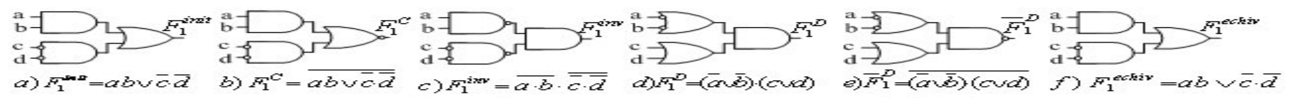

Figure 1. Binary relations of boolean function in rapport with the initial function: a) initial; b) complementary; c) inverse (negated); d) dual; e) dual complementary; f) equivalent

function (fig. 1,b). We make the equivalent logic modifications for the complementary function, using the laws of De Morgan: in this way, the negations appear at the inputs of the respective gates. Taking in considerations that a CC is not modified, if we transfer the invertors from the end of connexions to the beginning of them, we obtain CC from figure 1,c with the respective inverse function.

We continue the process of equivalent modification of the LF from the figure 1,c. using the laws of De Morgan. As a consequence, we obtain the circuit from figure 1,d named dual circuit, which contains invertors only on the connexions of the primary inputs. The respective LF is called dual function of the initial LF from the figure 1, a. It can be observed that the LF from figure 1, b; 1, c and 1, d are inverse LF of the figure 1, a LF. Meanwhile, the LF from figure 1, b; 1, c and 1, d are logical equivalent: the negation of each of these functions leads to obtain the initial function. On the other side, complementing the dual LF from figure 1,d we obtain LF from figure 1,e, (whose equivalent modifications are similary logic to those from figure 1,b, 1,c and 1,d). Modifying the complementary LF from figure 1, e in the dual LF we will obtain LF (from figure 1,f) equivalent to the initial LF (figure 1,a).

In the BA, to each law (axiom) based on the use of disjunction it corresponds a dual symetrical law (axiom) based on the use of disjunction. This situation is in conformity with the representation of some LF in the convention of PL and in the convention of NL. Still, DC are designed and are functioning in a one-way logic – the pozitive or the negative one. The notion of duality, but, was introduced initially for the LF realized in the convention of PL and NL. The LF can be compared only in the case of same tuples of binary arguments, that means of same definition domain. In the case of NL, the definition domain is oposed to the definition domain of same LF in PL: to any tuple of arguments values from the PL it corresponds a tuple with oposed values of the respective arguments from the NL, the respective LF having complementary values.

From here it comes and the myserious character, sometimes even enigmatic, of the duality notions of the LF represented in pozitive logic and respectively, in the negative one.

But both and the computers are designed and are functioning or in PL or in NL.

So, in many references the notion of duality is not appropiate reflected in the LA nor in the BA. Still there is changing the signification of duality notion.

The duality notion occups a special place in the Design for Testability (DFT).

The relation of duality of the LF in BA (or LA) is based on use of same definition domain of the compared LF and is made by using the inversion of logical values of arguments of some of the formulas of the logical functions.

III. CONCLUSIONS

The present work represents a study of relational properties of the LF of BA with the purpose of establishing new relational properties, which constitute the basic elements of the elaboration concepts of the digital structures, of methodes of formalizing the process of DFT and of afferent know-how.

This paper presents an analysis of the relational binary properties of the LF of BA and the obtaintion, for the first time, of original results with primordial importance for solving the DFT problem.

As a consequence of analysing the binary relations of the logical functions from boolean algebra, from figure 1 we take the following conclusions:

1. The binary relations between the LF exists from objective point of view and appears as a consequence of modification of LF in the process of design in the purpose of finding an optimal solution;
2. The boolean functions of which binary relations are studied must have same definition domain;
3. The boolean functions: dual, inverse or complementary are equivalent logic and represents 3 different hypostasis of one and same BF;
4. A complementary BF contains the inversion of function itself;
5. A dual BF contains inversions only of the primary variables;
6. A inverse BF can contain inversions of both the primary variables, and also of those internal and also of the output variable.

IV. REFERENCES

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

[2] Кузнецов О. П., Адельсон-Вельский Г. М. Дискретная математика для инженера. – М.: Энергия, 1980 г.–344 с.

[3] Николай Вавилов. Не совсем наивная теория множеств.- М., Изд-во Академии, 2004.- 325 с

[4] Белоусов А.И., Ткачев С.Б. Дискретная математика.- Изд-во МГТУ им. Н.Э. Баумана, 2004 г., -744 с

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

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

[7] С. И. Гуров. Упорядоченные множества и универсальная алгебра. Электронная версия. – Изд. МГУ, Москва, 2005 г. – 83 стр.

[8] Wakerly J. F. Circuite digitale. Principiile şi practicile folosite în proiectare, Bucureşti: Teora, – 2002. – 928 p.

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

[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.

[12] Е. Н. Вавилов, Г.П. Портной. Синтез схем электронных цифровых машин. - Изд. Советское Радио. – М.: 1963 г. – 440 с.

[13] А.А. Степанян. Цифровые вычислительные машины.- Куйбышевское книжное издательство.- 1966 г. – 191 с.