LATTICE THEORETIC FOUNDATION OF THE ANALYSIS AND SYNTHESIS OF DYNAMIC LOGICAL NETWORKS

By

J. SÁROSSY

Department for Theoretical Electricity, Polytechnical University, Budapest

(Received January 18, 1967)

Presented by Prof. Dr. K. SIMONYI

I. Introduction

The literature of logical design has tremendously increased since the foundational publication of SHANNON [1], and as a result of this nowadays there are many effective algebraic methods available for the logical designer. Especially well elaborated is the theory of static logical networks and synchronous automata (discrete-time networks); their usefulness were proved by design practice.

Apparently, it is quite another case for dynamic logical networks: hardly any articles published results referring to this and, according to our knowledge, a precise algebraic foundation has not been found yet. The basic circuits of dynamic logical networks are mostly very simple while their more general form affords wider design facilities. The reason of this greater generality is that beside the logic levels their changes (the so-called edges) and their quick alternations (the so-called pulses) can also be used for establishing logical relationships. In compliance with this fact, the "level-pulse variable" expression is used in the Russian literature. The greater generality and the simple means of realization promise a good prospect for the dynamic networks only the appropriate general designing methods should be found which aid the designer in the field of dynamic logic too. This paper tries to establish the algebraic foundation of these designing methods.

In order to avoid lengthiness, the proofs are in the most part only briefly summarized and in simpler cases completely omitted.

II. Definition of logical values

Our first most important step is the disavowal of the two valued Boolean algebra based exclusively on signal levels, and the introduction of new logical values. The aim is to find such a logical structure which defines operations, as best for the requirements of logical design, for the expanded set of logical values and for the functions defined on this set as well.

First of all, we define the set of symbols for the logical values and the corresponding distinctive physical states. In the usual way the symbols *0* and *1* will belong to the two logic levels while the level changes, considered as those of a finite interval, will be represented by the symbols *2* and *3*. *Thus our set of symbol-types is* $N = \{0, 1, 2, 3\}$. The connections with physical states are as follows: a logical value always symbolizes one of the possible four states of a physical quantity. If the variation domain of a physical quantity, varying continuosly according to our assumption, are divided to three subdomains of which none has zero measure, and only one (open) domain adjoins the two others (closed domains) at the same time, then the two outside domains can be represented by the *0* and *1* symbols. Let us assume that the variation of the physical quantity in the centre domain is always strictly monotonous. In this case *2* is the symbol for the state of the physical quantity when, in the centre domain, it changes from the domain already marked *0* to domain *1*. Symbol *3* represents the state when the physical quantity, also in the centre range, changes into the opposite direction. Further on, we assume that the physical quantity symbolized logically belongs to but one of the above four states, consequently the corresponding logical value can always be construed in the course of examination.

Three valued logics were already used e.g. in [2] and [3] for the synthesis of logical networks built up of elements having three states and for eliminating hazards respectively. As for the successful application of logics with more than three values, there is to be found no reference in the literature.

III. Operations in N, the $\mathscr{N}_1$ Boolean algebra

After determining the elements of the set N, disregard the concrete physical quantity and make set N and the algebraic structure formed from it the basis of our examination.

Introduce the operations of two variables the *disjunction* ($\vee$) and *conjunction* ($\wedge$) and the operation of one variable the *inversion* ($\neg$) by means of the following truth tables:

$\vee$	0	1	2	3
0	0	1	2	3
1	1	1	1	1
2	2	1	2	1
3	3	1	1	3

$\wedge$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	2	0
3	0	3	0	3

$\neg$	
0	[illegible]
1	[illegible]
2	[illegible]
3	[illegible]

(3.1)

(It would be incorrect to call the inversion negation because with more than two logical values, the negated value of an arbitrary logical value a, inasmuch as the negation means "non a", is not a determined logical value, and in this way the negation is not an operation).

Introducing the above operations the set N apparently became a structure which will be designated as $\mathcal{N}_1$.

For the sake of a more convenient way of writing let us introduce the following conventions: the metalogical operational and quantification symbols of the statements to be found in the theorems are (to make a distinction from the sign of similar operations which are just going to be discussed) the "*or*", "*and*", $\rightharpoondown$ (it is not true, that), $\Rightarrow$ (if ... then), $\Leftrightarrow$ (if, and only if), $\forall$ (for all ...), $\exists$ (there exists such ... that).

We use letters which represent an arbitrary element of the considered set orsub set. The expressions, formed by means of such letters and the introduced operational signs in a predetermined manner, are called formulae. Thus for instance if $A\,(x)$ is a formula containing letter x, and $\exists\,!$ is the sign of the singular quantifier, then the definition determining the singular quantifier is:

$$\exists!\,x\,A(x) \Leftrightarrow \exists\,x\,A(x) \quad and \quad \forall x\,\forall y\big(A(x)\ and\ A(y)\ \Rightarrow\ x = y\big).$$

Returning to the examination of $\mathcal{N}_1$, it is apparent that the disjunction defines an ordering relation as well:

$$\forall a\,\forall b(a \vee b = b\ \Leftrightarrow\ a \leq b), \tag{3.2}$$

i.e. the $a \vee b = b$ relation is reflexive, nonsymmetrical and transitive. On the basis of (3.2)

$$0 \leq 2 \leq 1 \quad and \quad 0 \leq 3 \leq 1. \tag{3.3}$$

Since

$$\forall a(0 \leq a \quad and \quad a \leq 1)$$

we may regard 0 as a minimum and 1 as a maximum element.

It is apparent that the minimum and maximum element fulfils the

$$\forall a(a \wedge 0 = 0) \tag{3.4}$$

$$\forall a(a \vee 1 = 1) \tag{3.5}$$

statements.

We introduce the notation for structures applied by [4]: if $\mathcal{S}$ is a structure, than $\mathcal{S} = \ <S;R;M;K>$, where S is the set of the elements of the structure, R is the defined relations, M is the defined operations and K is the aggregate of the special elements. For example, the Boolean algebra of the 0, 1 logical values is an ordinary Boolean algebra:

$$\mathcal{B} = \ <\{0, 1\}: \leq;\ \vee,\ \wedge, ^{-}; 0, 1>.$$

With this notation $\mathcal{N}_1 = \ <\{0,\ 1,\ 2,\ 3\};\ \leq;\ \vee,\ \wedge,\ -;\ 0, 1>$.

It can be shown with simple means that $\mathcal{N}_1$ is a *lattice* with respect to the operations $\vee$ and $\wedge$. The six axioms of lattice theory [5] require that

$$\forall a\, \forall b\, (a \vee b = b \vee a) \tag{3.6a}$$

$$\forall a\, \forall b\, \forall c\, ((a \vee b) \vee c = a \vee (b \vee c)) \tag{3.6b}$$

$$\forall a\, \forall b\, (a \vee (a \wedge b) = a) \tag{3.6c}$$

$$\forall a\, \forall b\, (a \wedge b = b \wedge a) \tag{3.6d}$$

$$\forall a\, \forall b\, \forall c\, ((a \wedge b) \wedge c = a \wedge (b \wedge c)) \tag{3.6e}$$

$$\forall a\, \forall b\, (a \wedge (a \vee b) = a). \tag{3.6f}$$

On the basis of the definition (3.1), the fulfilment of (3.6 a—f) can be simply verified.

If we regard the structures in which lattice operations are defined as a *logical structure* (shortly: logic), it follows that $\mathcal{N}_1$ is a logical structure.

In addition to the characteristics (3.6 a—f) it can also be easily verified that

$$\forall a\, \forall b\, \forall c\, (a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)). \tag{3.7}$$

According to a well known lattice-theoretical theorem [5] in case of the fulfilment of (3.7) the

$$\forall a\, \forall b\, \forall c\, (a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)) \tag{3.8}$$

statement is true as well, i.e. the distributivity exists in the opposite order too, that is the *lattice is distributive.*

The *complement* of element a is element u for which the following conditions are valid

$$a \wedge u = 0 \quad and \quad a \vee u\, 1. \tag{3.9}$$

If

$$\forall a\, \exists u\, (a \wedge u = 0 \quad and \quad a \vee u = 1) \tag{3.10a}$$

holds then the lattice is called *complemented;* if furthermore

$$\forall a\, \exists!\, u\, (a \wedge u = 0 \quad and \quad a \vee u = 1) \tag{3.10b}$$

then the lattice is called *uniquely complemented.* $\mathcal{N}_1$ is uniquely complemented because for distributive lattices the theorem is valid [5] that if they have maximum and minimum elements, then not more than one u belongs to any a which fulfils (3.9), however $u = \neg\, a$ fulfils (3.9).

The complemented distributive lattices are referred to as generalized Boolean algebra or simply Boolean algebra in the lattice theoretical literature. Thus $\mathcal{N}_1$ is a *Boolean algebra.*

We should make a remark here that instead of the $\mathcal{N}_1$ four valued logical structure the $\mathcal{P} = \langle \{0, 1, 2, 3\}; \leq; \cup, \cap, -; \sim, 0, 3 \rangle$, so called *Post-logic* could have been introduced, where the definitions of operations are:

$$a \cup b \overset{\text{def}}{=} \max(a, b) \tag{3.11a}$$

$$a \cap b \overset{\text{def}}{=} \min(a, b) \tag{3.11b}$$

$$\sim a \overset{\text{def}}{=} a + 1 \pmod 4 \tag{3.11c}$$

$$- a \overset{\text{def}}{=} 3 - a. \tag{3.11d}$$

It can be briefly proved that though (3.11 a—b) define lattice operations, the complement cannot be construed and this way $\mathcal{P}$ is not a Boolean algebra. Therefore, it is more practical to work with $\mathcal{N}_1$ (or with $\mathcal{N}$ which is to be introduced later on), because far more well applicable theorems are known for Boolean algebras and in this way their theory is far more elaborated too.

It is valid for Boolean algebras and so for $\mathcal{N}_1$ too [5] that

$$\forall a \forall b (\neg (a \vee b) = \neg a \wedge \neg b) \tag{3.12a}$$

and

$$\forall a \forall b (-(a \wedge b) = \neg a \vee \neg b) \tag{3.12b}$$

respectively. These two theorems are the generalization of DeMorgan's theorems.

IV. The $\mathcal{F}_1$ Boolean algebra of functions which can be construed on $\mathcal{N}_1$

Consider A and B as two arbitrary sets. In this case the *direct product* $A \times B$ is the set of the element pairs (a, b), $a \in A$, $b \in B$. If $\mathcal{S}$ is an algebraic structure on set S, then the

$$f \colon S^{\times k} \to S \tag{4.1}$$

type homomorphisms are called *k-valued functions* on $\mathcal{S}$ (see [4]). ($S^{\times k}$ means the k-fold direct product of set S with itself, i.e. its k-th "direct power".) At the same time (4.1) also expresses that

$$f \colon (x_1, x_2, \ldots, x_k) \to f, \tag{4.2}$$

i.e. with an arbitrary choice of x_i-s the homomorphic image of the element $(x_1, x_2 \ldots, x_k) \in S^{\times k}$ is an appropriate $f \in S$ element. (In the forthcomings it will not be disturbing if both the function and its value will be denoted by the same letter.) Therefore, the homomorphism of (4.1) can be written in the form $f(x_1, x_2, \dots x_k)$ where the x_i-s are arbitrary, undetermined elements in S.

The identity of the two functions is defined:

$$f(x_1, \dots, x_k) = g(x_1, \dots, x_k) \Leftrightarrow \forall x_1 \dots \forall x_k (f = g).$$

Let as now deal with not more than k-valued functions in the form $f: N^{xk} \to N$, and within this with their set F_1^k determined hereafter:

$$0, 1, 2, 3 \in F_1^k,$$

$$x_1, x_2, \dots, x_k \in F_1^k,$$

$$f(x_1, \dots, x_k) \in F_1^k \quad and \quad h(x_1, \dots, x_k) = \neg f(x_1, \dots, x_k) \Rightarrow$$
$$\Rightarrow h(x_1, \dots, x_k) \in F_1^k,$$

$$f(x_1, \dots, x_k) \in F_1^k \quad and \quad g(x_1, \dots, x_k) \in F_1^k \quad and \quad h(x_1, \dots, x_k) =$$
$$= f(x_1, \dots, x_k) \vee g(x_1, \dots, x_k) \quad \Rightarrow \quad h(x_1, \dots, x_k) \in F_1^k,$$

$$f(x_1, \dots, x_k) \in F_1^k \quad and \quad g(x_1, \dots, x_k) \in F_1^k \quad and \quad h(x_1, \dots, x_k) =$$
$$= f(x_1, \dots, x_k) \wedge g(x_1, \dots, x_k) \quad \Rightarrow \quad h(x_1, \dots, x_k).$$

With the above definition F_1^k was directly construed as a Boolean algebra, i.e. a logical structure:

$$\mathcal{F}_1^l = < F_1^k; \leq; \vee, \wedge, \neg; 0, 1>.$$

Since the $\mathcal{F}_1^k$ structure was generated with the operations of $\mathcal{N}_1$, $\mathcal{F}_1^l$ may be regarded as the structure of functions construed on the $\mathcal{N}_1$ Boolean algebra. It is obvious that all theorems, shown so far for $\mathcal{N}_1$, are also valid for $\mathcal{F}_1^k$.

In the forthcomings we shall deal with the problem that on what conditions can be all functions in F_1^k expressed by means of favoured functions.

That r lattice element is called *irreducible for the disjunction*, for which

$$\neg \exists a \exists b (r = a \vee b \quad and \quad a \neq r \quad and \quad b \neq r). \tag{4.3}$$

If for an element a

$$a = \bigvee_{i=1}^{j} r_i \tag{4.4}$$

and $r_1, r_2, \dots, r_j$ are irreducible elements, the (4.4) is the *irreducible disjunctive expansion* of a.

Let us now define the *atoms* of the lattice. The element q is the atom of the lattice if

$$\forall x\big((q \wedge x = 0 \text{ or } q \wedge x = q) \text{ and } q \neq 0\big). \tag{4.5}$$

It is apparent that q is at the same time an irreducible element for the disjunction, otherwise

$$\exists a \, \exists b \, (q = a \vee b \text{ and } a \neq q \text{ and } b \neq q \text{ and } a \neq 0$$

because from (4.3)

$$\rightarrow (r = n \vee b \text{ and } a \neq r \text{ and } b \neq r \text{ and } a = 0);$$

thus

$$\exists \, a \, (q \wedge a = a \text{ and } a \neq q \text{ and } a \neq 0)$$

as a consequence of (3.6 f). Comparing with (4.5) it may be seen, that q cannot be an atom.

Similar considerations can be taken as regards the irreducible elements for conjunction and dual atoms resp. (in each relationship only the $\vee \rightarrow \wedge$, $\wedge \rightarrow \vee$, $0 \rightarrow 1$ replacements should be made); because of the complete analogy, following from the duality theorem of lattice theory, this case is not dealt with separately.

A lattice is called *atomic* if Q is the set of its atoms and

$$\forall x \, \exists \, q \, (x \neq 0 \ \Rightarrow \ x \wedge q \text{ and } q \in Q). \tag{4.6}$$

It can be stated as a special case of a general theorem that all the lattices, having a finite number of elements, are atomic [5].

In distributive lattices the theorem is valid [5]: each element of the lattice, differing from the minimum element, disregarding the sequence of the components, has at the maximum one not abbreviatable irreducible disjunctive expansion. Such expansion of an irreducible element is the element itself in compliance with the definition.

In Boolean algebras, all the elements, irreducible for disjunction, are at the same time atoms too. This theorem can be proved e. g. on the basis of a well known theorem that states the unique expandability of each lattice—element with the aid of complete conjunctive terms, that is with the aid of atoms.

Applying the above considerations for $\mathscr{F}_1^k$, it can be seen that $\mathscr{F}_1^k$ is atomic (the number of the not more than k valued functions is obviously finite), and all of its elements (apart from the 0 valued constant function) can be expanded uniquely by the disjunctive expression of atoms. This expansion is called the *canonical form* in $\mathscr{F}_1^k$.

Omitting the simple proof, we mention that the atoms of $\mathcal{F}_1^k$ can be expressed in the form $2 \wedge Q_k^i$, $3 \wedge Q_k^i$ from where

$$Q_k^i = \bigwedge_{j=1}^{k} (m_j^i \wedge x_j \vee \neg\, m_j^i \wedge \neg\, x_j). \tag{4.7}$$

Here $m_j^i = 0,\ 1$ and $i = \sum_{j=1}^{k} m_j^i\ 2^{j-1},\ 0 \leq i \leq 2^k - 1.$

From the foregoings it is evident that the basic characteristics of $\mathcal{F}_1^k$ greatly correspond with the characteristics of the Boolean algebra $\mathcal{G}^k$ of not more than k-valued functions construed on the $\mathcal{B}$ ordinary Boolean algebra. The question arises, whether $\mathcal{F}_1^k$ can be decomposed, in some way, into two twovalued $\mathcal{G}^k$ Boolean algebras?

In general, a lattice $\mathcal{L}$ can be decomposed into the *direct composition* of lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ (in designation $\mathcal{L} = \mathcal{L}_1 \otimes \mathcal{L}_2$), if an $a_1 \in \mathcal{L}_1$, $a_2 \in \mathcal{L}_2$ elementpair can be assigned uniquely to an element $a \in \mathcal{L}$ (in designation $a \longleftrightarrow (a_1, a_2)$) so that

$$a \cup b \longleftrightarrow (a_1 \cup_1 b_1, a_2 \cup_2 b_2) \tag{4.8a}$$

$$a \cap b \longleftrightarrow (a_1 \cap_1 b_1, a_2 \cap_2 b_2) \tag{4.8b}$$

$$\neg\, a \longleftrightarrow (\neg_1 a_1, \neg_2 a_2) \tag{4.8c}$$

where $\cup, \cap, \neg$ in $\mathcal{L}$, $\cup_1, \cap_1, \neg_1$ in $\mathcal{L}_1$, and $\cup_2, \cap_2, \neg_2$ in $\mathcal{L}_2$ are defined as lattice operations and complement formations [5]; $b \in \mathcal{L}$, $b_1 \in \mathcal{L}_1$, $b_2 \in \mathcal{L}_2$.

On the basis of the foregoing, $\mathcal{F}_1^k$ is *directly decomposable* in the $\mathcal{G}^k \otimes \mathcal{G}^k$ form. The decomposition can be given e.g. in the following manner: $a \longleftrightarrow (a_1, a_2)$, where in a "truth" table:

a	a_1	a_2
0	0	0
1	1	1
2	0	1
3	1	0

(4.9)

replacing $\cup, \cap, \neg$ with the operations defined in $\mathcal{F}_1^k$ and the operations of the ordinary Boolean algebra putting instead of $\cup_1, \cup_2, \cap_1, \cap_2, \neg_1, \neg_2$, the above decomposition can be verified.

The four-valued Boolean algebra, discussed in the foregoing, is a means applicable for the description of static logic networks because the OR-gates, AND-gates and inverters playing a role in such networks, realize in effect the operations defined in (3.1). Though, by means of the four-valued Boolean algebra beside the levels the edges can be considered as well, the procedures in the above structure do not differ in effect from those used in the ordinary Boolean algebra, moreover, $\mathcal{F}_1^k$ decomposes into two ordinary Boolean algebras.

We mention it here, that apart from the redesignation of the four logical values (3.1), the definition of operations is unique if we want that $\mathcal{N}_1$ and $\mathcal{F}_1^k$ resp. should be Boolean algebra.

V. The $\mathcal{N}$ and $\mathcal{F}^k$ complete Boolean algebras

There is one essential difference between the Boolean algebras $\mathcal{C}_j^k$ and $\mathcal{F}_1^k$: while $\mathcal{C}_j^k$ comprises all $\varphi : B^{\times k} \to B$ functions, $\mathcal{F}_1^k$ does not include all mappings of the form $f: N^{\times k} \to N$, but it is the Boolean sub-algebra of these transformations (functions).

To verify this statement, a function of a single variable is shown, which, despite of being an $N \to N$ mapping is not the element of $F_1^k : f: 0 \to 0, 1 \to 0, 2 \to 1, 3 \to 0$. Designate this function $\triangleq x$ (read delta-dash); in the forthcoming it will play an important role.

Construe the $\triangleq$-function as an operation defined in N:

$$\begin{array}{c|c} \triangleq & \\ \hline 0 & 0 \\ 1 & 0 \\ 2 & 1 \\ 3 & 0 \end{array} \tag{5.1}$$

Hereupon, we shall form a new structure: $\mathcal{N} = \langle N; \leq; \vee, \wedge, \neg, \triangleq; 0, 1\rangle$. We can construe the $\mathcal{F}^k$ function-structure on this structure, only in part IV in the definition of F_1^k, F^k should be written instead of F_1^k and it should be completed with the definition line

$$f(x_1, \ldots, x_k) \in F^k \quad and \quad h(x_1, \ldots, x_k) = \triangleq f(x_1, \ldots, x_k) \Rightarrow$$
$$\Rightarrow h(x_1, \ldots, x_k) \in F^k .$$

Now let us define the following basic functions:

$$\Box x \stackrel{\text{def}}{=} \triangleq x \vee \triangleq \neg x \tag{5.2}$$

$$p_0(x) \stackrel{\text{def}}{=} \neg x \wedge \neg \Box x \tag{5.3a}$$

$$p_1(x) \stackrel{\text{def}}{=} x \wedge \neg \Box x \tag{5.3b}$$

$$p_2(x) \stackrel{\text{def}}{=} \triangle x \tag{5.3c}$$

$$p_3(x) \stackrel{\text{def}}{=} \triangle \neg x \tag{5.3d}$$

(On the left side of (5.2) read: square.) By means of these functions, the atoms of $\mathcal{F}^k$ can be written like this:

$$q_k^j = 2 \wedge \bigwedge_{m=1}^{k} p_{i_m}(x_m), \quad 0 \leq j < 4^k, \quad j = \sum_{m=1}^{k} i_m 4^{m-1}, \tag{5.4a}$$

$$q_k^j = 3 \wedge \bigwedge_{m=1}^{k} p_{i_m}(x_m), \quad 4^k \leq j < 2.4^k, \quad j = 4^k + \sum_{m=1}^{k} i_m 4^{m-1} \tag{5.4b}$$

where in both cases $0 \leq i_m \leq 3$. If we use the exponent designation, usual in ordinary Boolean algebras, and definition like

$$x^i \stackrel{\text{def}}{=} p_i(x), \tag{5.5}$$

then

$$q_k^j = 2 \wedge \bigwedge_{m=1}^{k} x_m^{i_m}, \quad 0 \leq j < 4^k, \quad j = \sum_{m=1}^{k} i_m 4^{m-1}, \tag{5.6a}$$

$$q_k^j = 3 \wedge \bigwedge_{m=1}^{k} x_m^{i_m}, \quad 4^k \leq j < 2.4^k, \quad j = 4^k + \sum_{m=1}^{k} i_m 4^{m-1}. \tag{5.6b}$$

Now the statement can be made that $\mathcal{F}^k$ *is the Boolean algebra of all* $f: N^{\times k} \to N$ *type, four-valued functions of not more than k-variables.* Only that should be shown that (5.6 a—b) really produce the atoms of all four valued functions, since hereafter all the other functions can be given as irreducible disjunctive forms.

First of all, the proof of an lemma is outlined. Let $\mathcal{L} = \langle L; \leq; \cup, \cap, \neg; 0, n-1\rangle$ be an atomic lattice of n elements where $L = \{0, 1, \ldots, n-1\}$, $\cup$ and $\cap$ are lattice operations, while $\neg$ means the forming of a complement. Let f be the mapping of the elements of set $L^{\times k}$, into the elements of set L. Let $\mathcal{H}^k$ be the lattice of functions definable in lattice $\mathcal{L}$, of not more than k variables. Herewith, f is an atom in $\mathcal{H}^k$ if and only if

$$f: L^{\times k} \to \{0, q\} \text{ and } \left(f^{-1}: q \to L_q^{\times k} \Rightarrow card\,(L_q^{\times k}) = 1\right) \tag{5.7a}$$

Here q means any atom of L, f^{-1} is the inverse transformation of f, $card\,(L_q^{\times k})$ is the cardinality of subset $L_q^{\times k}$. This condition can be written in another form:

$$\begin{gathered} \forall x_1 \ldots \forall x_k \left(f(x_1, \ldots, x_k) = 0 \quad \text{or} \quad f(x_1, \ldots, x_k) = q\right) \\ \textit{and} \quad \exists! x_1 \ldots \exists! x_k \left(f(x_1, \ldots, x_k) = q\right) \end{gathered} \tag{5.7b}$$

where q, just like above is an arbitrary atom of L.

If φ is an arbitrary element of $\mathscr{H}^k$, on the basis of definition (4.5) only that should be verified that

$$\forall\varphi(x_1,\dots,x_k)\big((f(x_1,\dots,x_k)\wedge\varphi(x_1,\dots,x_k)=0 \quad or \\ f(x_1,\dots,x_k)\wedge\varphi(x_1,\dots,x_k)=f)\quad and\quad f(x_1,\dots,x_k)\neq 0\big). \tag{5.8}$$

First of all the formula (5.7 b) containing singular quantifiers, directly ensures that $f(x_1,\dots,x_k)\neq 0$. On the other hand, it is evident, that

$$\forall x_1\dots\forall x_k\big((x_1,\dots,x_k)\notin L_q^{\times k}\Rightarrow f(x_1,\dots,x_k)\wedge\varphi(x_1,\dots,x_k)=0\big)$$

and

$$(x_1,\dots,x_k)\in L_q^{\times k}\Rightarrow\big(f(x_1,\dots,x_k)\wedge\varphi(x_1,\dots,x_k)=0\quad or \\ f(x_1,\dots,x_k)\wedge\varphi(x_1,\dots,x_k)=f(x_1,\dots x_k)\big)$$

since, in case of the fulfilment of the implicant foremember, the value of f can only be an atom of L, in compliance with (5.7 a). Comparing these results we obtain (5.8). It can also be proved that if a $g(x_1\dots x_k)$ does not fulfil (5.7 a), it can always be produced as the disjunction of at least two functions, complying (5.7 a), provided that $g(x_1\dots x_k)\neq 0$.

Now, it is easy to comprehend the functions q_k^j, defined by (5.6 a) and (5.6 b), are really the atoms of the lattice of functions type $f: N^{\times k}\to N$, since on the basis of (5.3 a—d) and (5.4 a—b)

$$\forall m\,(i_m=x_m)\Leftrightarrow\bigwedge_{j=1}^{k}x_j^{i_j}\neq 0$$

with such values of i_m (5.6 a) and (5.6 b) give 2 and 3, resp. i.e. the atoms of N.

To prove that the atoms according to (5.6 a) and (5.6 b) give the totality of functions f: $N^{\times k}\to N$, we leave for the reader.

Having these finally obtained results, it follows directly the corollary: $\mathscr{F}^k$ is the Boolean algebra of all possible, not more than k-valued functions, as was stated before.

There remains the objective, to give the canonical expansion of an arbitrary $f\in F^k$ function. Let $N^{\times k}(i)\subseteq N^{\times k}$ $0\leq i\leq 3$ be in such a way that

$$f\colon (x_1,\dots,x_k)\to i\ \Rightarrow\ (x_1,\dots,x_k)\in N^{\times k}(i).$$

With this notation for arbitrary $f(x_1,x_2,\dots x_k)$

$$f(x_1,\dots,x_k)=\bigvee_{\substack{(i_1,\dots,i_k)\in\\ N^{\times k}(2)\cup N^{\times k}(1)}}q_k^j\vee\bigvee_{\substack{(i_1,\dots,i_k)\in\\ N^{\times k}(3)\cup N^{\times k}(1)}}q_k^{4^k+j}, \tag{5.9}$$

where $j=\sum_{m=1}^{k}i_m\cdot 4^{m-1}$.

We refer here that the system of functions in the above sense, can be expressed not only in the $\mathcal{F}^k = \langle F^k; \leq; \vee, \wedge, \neg, \triangleq \rangle$ form but by means of other operations, differring from these. In the foregoing, reference was already made to the Post-system [6]: $\mathcal{F}_p^k = \langle F^k; \leq; \cup, \cap, \sim, - \rangle$, this also forms a complete system. But introducing, for example, the generalized Scheffer operation:

$$a|b \overset{\text{def}}{=} \neg a \vee \neg b \tag{5.10}$$

$\mathcal{F}_s^k = \langle F^k; \leq; |, \triangleq \rangle$ is also a complete system. For the sake of interest we remark that a complete system can be produced even by means of a single operation: $F_r^k = \langle \mathcal{F}^k; \leq; 0 \rangle$ where $a \bigcirc b$ is the "Vebb operation":

$$a \bigcirc b \overset{\text{det}}{=} \sim (a \cup b) \tag{5.11}$$

VI. The application of the $\mathcal{N}$ and $\mathcal{F}^k$ Boolean algebras for the analysis and synthesis of logical networks

There were already initiatives to use lattice theoretical means for the description of logical networks [7], but with knowledge of discussions, more or less independent of lattice theory, of ordinary Boolean algebras, the use of lattice theory is really not necessary. In many-valued logics, on the contrary, one can get along, practically only with knowledge of lattice theory.

As for the description of dynamic logical networks or the introduction of many-valued logics respectively, in the Soviet Union several research workers made initiative steps ([8], [9], [10]), and achieved interesting results, but they did not lay much stress on the general algebraic treatment. Though [9] introduces six logical values and defines operations, but further on gives only the form of a half-canonical expansion. Later on, we shall show that the two logical values, introduced in [9] for the pulses, are not necessary because by means of the four-valued Boolean algebra shown in the foregoing, everything can be described for which the structure, introduced in [9], was intended.

[8], [10], and [11] deal almost exclusively with the special features of the corresponding operation, denoted as $\triangleq$ by us. They give (especially [11]) useful algorithms for the logical designer. Their common restriction is that they use two-valued Boolean algebra, in this way they cannot elaborate a uniform, consistent formalism.

Let us now pay attention to the appearence in time, of logical signals. If we disregard the very short duration pulses, it can be said that the logical values might come one after the other in a strict sequential order: *0* may be followed only by *2*, *2* only by *1*, *1* only by *3*, and *3* only by *0*. If we permit *3* to follow *2*, and *2* might come after *3*, then we already have appropriate means for the handling of pulses too. It is easy to see however, that by means of a combinational network in a wider sense (namely for which $x_{\text{out}}(t) = f(x_{\text{in}}(t))$, x_{out}, $x_{\text{in}} \in N$, $f \in F^k$) the above type pulse cannot be produced of one edge (see for example fig. 1.).

This fact would make the analysis much more difficult. Let us make some further abstractions, permit the succession of *2* and *0*, and *3* and *1* respectively

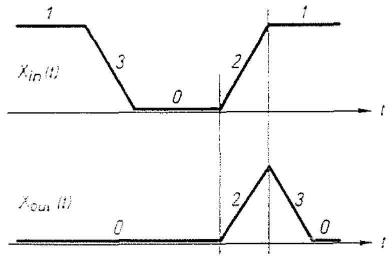

Fig. 1. Non-combinational differentiation

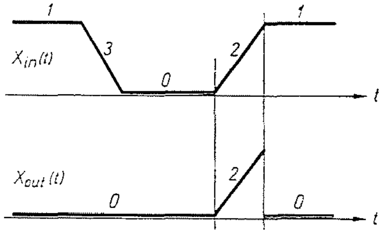

Fig. 2. "Edge preserving" differentiation

too. In this case, the formation of a pulse from an edge (see fig. 2.) can be more simply represented.

This type of "differentation" can be described with the function

$$x_{\text{out}}(t) = x_{\text{in}}(t) \wedge \triangleq x_{\text{in}}(t)$$

thus it can be handled as a combinational logic. Finally, of course, there is the possibility to connect the pulse, at the formation from an edge (in case of "differentation") to the appearance of a suitable short duration logic level (*0* or *1*) (see fig. 3.). At this point, naturally, we should assume

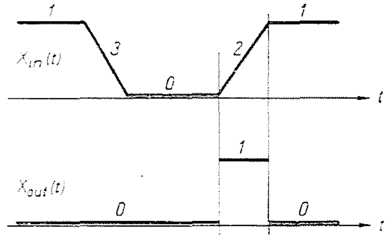

Fig. 3. "Perfect" differentiation

that the levels (*0* and *1*) might also follow each other, i.e. the obtained pulse has no edges (see [8], (3)). The relationship between the signals of fig. 3. can now be expressed especially simply:

$$x_{\text{out}}(t) = \triangleq x_{\text{in}}(t).$$

With this, at the same time, a clear interpretation of operation $\triangleq$ is given.

At this point, it is important to realize that the formalism, introduced in chapters III., IV. and V., does not mean any restrictions for the timing sequence of the individual states, and thus any representation may be applied freely, suitably to the purpose.

As for the realization techniques, in most cases, the maximum interval, while the states *2* and *3* are maintained, is commensurable with the disturbing signal shifts caused by the delay, storage, etc. of the circuits. In such realizations it is not practical to build up, for example, the direct realization of the following expression:

$$x_{\text{out}} = \underline{\triangle}\, x_{\text{in}\,1} \wedge \underline{\triangle}\, x_{\text{in}\,2} \wedge x_{\text{in}\,3}\,.$$

In other realizations it occurs that, though, the defined operations can be realized by simple physical means, the realization is not perfect (especially difficult is the realization of the relationship $2 \wedge 3 = 0$, $2 \vee 3 = 1$). These restrictive aspects should be taken into consideration just like the timing hazards in two-valued logic.

On the basis of the above said, there is no objection to the precise definition of *dynamic logical networks: to describe algebraically the operation of such networks, beside the operations defined in (3.1), it is necessary to introduce the operation of (5.1) as well.* On the other hand, *the logical networks which can be properly described by the operations of (3.1), are called static-logical networks.*

Now let us turn our attention to the extension, important from a practical aspect, of the set of operations and functions introduced so far. Let us introduce the following operations (functions):

$$\overline{\triangledown}\, x \overset{\text{def}}{=} \underline{\triangle}\, \neg\, x \tag{6.1a}$$

$$\triangle\, x \overset{\text{def}}{=} x \wedge \underline{\triangle}\, x \tag{6.1b}$$

$$\triangledown\, x \overset{\text{def}}{=} x \vee \neg\, \overline{\triangledown}\, x \tag{6.1c}$$

$$x \mathbin{\text{◺}} y \overset{\text{def}}{=} (x \wedge \neg \Box\, x \vee \neg\, \overline{\triangledown}\, x) \wedge y\, \underline{\triangle}\, y \tag{6.1d}$$

$$x \mathbin{\text{◸}} y \overset{\text{def}}{=} \neg((\neg\, x \wedge \neg \Box\, x \vee \underline{\triangle}\, x) \wedge \neg\, y \wedge \overline{\triangledown}\, y) \tag{6.1e}$$

(reading in the above order: nabla-dash, delta, nabla, pre-delta, pre-nabla). By means of these operations, the so called differentiating or priming-differentiating gates can be described. Thus, it might be said that the dynamic logical networks are characterised by the occurence of means, realizing the operations (5.1), (5.2), (6.1a—e).

In the further part of this chapter, some useful identities, in connection with the above operations, are shown. The mostly simple verifications are left to the reader. To elude the unnecessary parenthesises we shall keep the following precedence order among operations: $\neg$, $\underline{\triangle}$, $\overline{\triangledown}$, $\triangle$, $\triangledown$, $\square$; $\wedge$; ◺; $\vee$; ◸. Among unary (of one variable) operations the order of writing is decisive.

First, here are the relations and their duals, related to nablas, deltas and quadrangles:

$$\triangledown a = \neg \triangle \neg a; \qquad \triangle a = \neg \triangledown \neg a \tag{6.2a}$$

$$a \wedge \triangle a = a; \qquad a \vee \triangledown a = a \tag{6.2b}$$

$$\triangle \triangledown a = 0; \qquad \triangledown \triangle a = 1 \tag{6.2c}$$

$$\triangle a \wedge \triangle \neg a = 0; \qquad \triangledown a \vee \triangledown \neg a = 1 \tag{6.2d}$$

$$\overline{\triangledown}\, a \wedge \overline{\triangledown}\, a = 0 \tag{6.2e}$$

$$\underline{\triangle}\,\underline{\triangle}\, a = 0; \qquad \overline{\triangledown}\,\overline{\triangledown}\, a = 0 \tag{6.2f}$$

$$\underline{\triangle}\,\overline{\triangledown}\, a = 0; \qquad \overline{\triangledown}\,\underline{\triangle}\, a = 0 \tag{6.2g}$$

$$\underline{\triangle}\, a \wedge \triangle b = \triangle a \wedge \triangle b; \qquad \neg \overline{\triangledown}\, a \vee \triangledown b = \triangledown a \vee \triangledown b \tag{6.2h}$$

$$\square a = \square \neg a \tag{6.2i}$$

$$\neg \square a \wedge \underline{\triangle}\, a = 0; \qquad \neg \square a \wedge \overline{\triangledown}\, a = 0. \tag{6.2j}$$

Employing the (6.2) relationships

$$a \,\text{◺}\, b = (a \wedge \neg \square a \vee \triangle \neg a) \wedge \triangle b \tag{6.3a}$$

$$\begin{aligned} a \,\text{◸}\, b &= \neg (\neg a \,\text{◺}\, \neg b) = \\ &= \neg(\neg a \wedge \neg \square a \vee \triangle a) \vee \triangledown b. \end{aligned} \tag{6.3b}$$

Also it can simply be derived that

$$\triangle a \,\text{◺}\, b = 0; \qquad \triangledown a \,\text{◸}\, b = 1 \tag{6.4a}$$

$$\triangledown a \,\text{◺}\, b = \triangle b; \qquad \triangle a \,\text{◸}\, b = \triangledown b \tag{6.4b}$$

$$a \,\text{◺}\, a = 0; \qquad a \,\text{◸}\, a = 1 \tag{6.4c}$$

$$\triangle (a \,\text{◺}\, b) = a \,\text{◺}\, b; \qquad \triangledown (a \,\text{◸}\, b) = a \,\text{◸}\, b \tag{6.4d}$$

$$\triangle (a \,\text{◸}\, b) = 0; \qquad \triangledown (a \,\text{◺}\, b) = 1. \tag{6.4e}$$

Introduce the following notation:

$$\boxbslash\, a = a \wedge \neg \square a \vee \triangle \neg a. \tag{6.5}$$

It can be verified with the appropriate "truth" tables that

$$\boxbslash\,(a \wedge b) = \boxbslash\, a \wedge \boxbslash\, b.$$

On the other hand

$$\boxbslash\,(a \vee b) \neq \boxbslash\, a \vee \boxbslash\, b$$

but

$$2 \wedge \boxbslash\,(a \vee b) = 2 \wedge (\boxbslash\, a \vee \boxbslash\, b).$$

With these and with (6.3a)

$$a \wedge b \text{◺} c = (a \text{◺} c) \wedge (b \text{◺} c)$$
$$(a \vee b) \text{◺} c = a \text{◺} c \vee b \text{◺} c.$$

From these two identities, with full induction and dualization resp., can be obtained

$$\bigwedge_i^n a_i \text{◺} b = \bigwedge_i^n (a_i \text{◺} b) \tag{6.6a}$$

$$(\bigvee_i^n a_i) \text{◺} b = \bigvee_i^n a_i \text{◺} b \tag{6.6b}$$

$$\bigwedge_i^n a_i \text{◸} b = \bigwedge_i^n (a_i \text{◸} b) \tag{6.6c}$$

$$\bigvee_i^n a_i \text{◸} b = \bigvee_i^n (a_i \text{◸} b) \tag{6.6d}$$

If we write down the "truth" tables of $\triangle\,(a \wedge b)$, and $\triangle(a \vee b)$, it can be verified merely at a glance that

$$\triangle\,(a \wedge b) = a \wedge b \wedge (\triangle\, a \vee \triangle\, b)$$
$$\triangle\,(a \vee b) = \neg\, a \text{◺} b \vee b \text{◺} a$$

or with dualization

$$\nabla\,(a \vee b) = a \vee b \vee (\nabla\, a \wedge \nabla\, b)$$
$$\nabla\,(a \wedge b) = (\neg\, a \text{◸} b) \wedge (\neg\, b \text{◸} a).$$

With induction:

$$\triangle\,(\bigwedge_i^n a_i) = \bigwedge_i^n a_i \wedge \bigvee_i^n \triangle\, a_i \tag{6.7a}$$

$$\triangle(\bigvee_i^n a_i) = \bigvee_i^n \bigwedge_{j \neq i}^n (\neg\, a_j \text{◺} a_i) \tag{6.7b}$$

$$\nabla\left(\bigwedge_{i}^{n} a_i\right) = \bigwedge_{i}^{n} \bigvee_{j\neq i}^{n} (\neg\, a_j ⊿ a_i) \tag{6.7c}$$

$$\nabla\left(\bigvee_{i}^{n} a_i\right) = \bigvee_{i}^{n} a_i \vee \bigwedge_{i}^{n} \nabla\, a_i. \tag{6.7d}$$

Using (6.7a—b) in (6.3a)

$$\begin{aligned} a ⊿ \bigwedge_{i}^{n} b_i &= \boxslash\, a \wedge \triangle\left(\bigwedge_{i}^{n} b_i\right) = \\ &= \boxslash\, a \wedge \bigwedge_{i}^{n} b_i \wedge \left(\bigvee_{i}^{n} \triangle\, b_i\right) = \\ &= \bigwedge_{i}^{n} b_i \wedge \bigvee_{i}^{n} a ⊿ b_i \end{aligned} \tag{6.8a}$$

$$\begin{aligned} a ⊿ \left(\bigvee_{i}^{n} b_i\right) &= \boxslash\, a \wedge \triangle\left(\bigvee_{i}^{n} b_i\right) = \\ &= \boxslash\, a \wedge \bigvee_{i}^{n} \bigwedge_{j\neq i}^{n} (\neg\, b_j ⊿ b_i) = \\ &= \bigvee_{i}^{n} \bigwedge_{j\neq i}^{n} (\boxslash\, a \wedge \boxslash\, \neg\, b_j \wedge \triangle\, b_i) = \\ &= \bigvee_{i}^{n} \bigwedge_{j\neq 1}^{n} (a \wedge \neg\, b_j ⊿ b_i). \end{aligned} \tag{6.8b}$$

With dualization it can be directly obtained again that

$$a ◸ \bigwedge_{i}^{n} b_i = \bigwedge_{i}^{n} \bigvee_{j\neq i}^{n} (a \vee \neg\, b_j ◸ b_i) \tag{6.8c}$$

$$a ◸ \bigvee_{i}^{n} b_i = \bigvee_{i}^{n} b_i \vee \bigwedge_{i}^{n} (a ◸ b_i). \tag{6.8d}$$

Surveying all these relations, it is evident that, according to what was discussed in chapter IV., they apply to the elements of N and F^k as well.

VII. Application examples

In this chapter we shall try to demonstrate the practical application of those, shown so far, especially in the preceding chapter, on some simple, more exactly simplified, examples.

1. *Equation of dynamic flip-flop*

It is well known that the operation of storage elements cannot be described with an equations of the following type:

$$x_{\text{out}}(t) = f\big(x_{\text{in}\,1}(t), x_{\text{in}\,2}(t), \ldots\big). \tag{7.1}$$

The flip-flop is the simplest of storage elements since no time parameter should be given to describe it. That is why we choose it now as an example. Let us decide upon the following notation:

$$x'(t) = \lim_{\tau \to t} x(\tau), \quad \tau < t \tag{7.2}$$

(Introducing this notation, we must remind the logical value — physical parameter assignment, given in chapter I., especially as regards the domains of the physical parameter to be open or closed.) With this notation the general equation of the flip-flop is:

$$\begin{aligned} Q = \neg\, r \wedge (s \vee \uparrow Q') \wedge \neg(\underline{\triangle}\, s \wedge \underline{\underline{\triangle}}\, Q') \vee \\ \triangle\, s \wedge \triangle\, Q' \vee \neg\, r \wedge \underline{\triangle}\, r \wedge \overline{\triangledown}\, Q' \end{aligned} \tag{7.3}$$

with the condition that

$$\forall t\,\big(s(t) \wedge r(t)\big) = 0 \tag{7.4}$$

where

$$\uparrow Q' \overset{\text{def}}{=} \underline{\triangle}\, Q' \vee \neg \Box\, Q' \wedge Q',$$

while s and r symbolize the set and reset signals. Though (7.3) is absolutely precise, its use would lead beyond the scope of this paper, thus let us be satisfied with the treatment of the simplified equation below:

$$Q = \neg\, r \wedge (s \vee Q') \tag{7.5}$$

while (7.4) is valid further on. This formula requires two restrictions: on the one hand

$$\forall t\,\big(s(t) \wedge r'(t) = 0\big) \quad \textit{and} \quad \forall t\,\big(s'(t) \wedge r(t) = 0\big) \tag{7.6a}$$

on the other hand, for these t_0 in which

$$s(t_0) = 0 \quad \textit{and} \quad r(t_0) = 0 \quad \textit{and} \quad \big(Q'(t_0) = 2 \quad \textit{or} \quad Q'(t_0) = 3\big) \tag{7.6b}$$

the above does not give a quite correct result.

The discussion and comparison of (7.3) and (7.5) would give many interesting results but this would exceed the aims of this paper. We remark only that following from (7.6) and (7.5)

$$\forall t \,(Q(t) \neq Q'(t) \Rightarrow Q(t) = 1 \quad and \quad Q'(t) \neq 3 \\ or \quad Q(t) = 0 \quad and \quad Q'(t) \neq 2) \tag{7.7}$$

In case of dynamic set and reset

$$s = x_s \triangle c_s \tag{7.8a}$$

$$r = x_r \triangle c_r \tag{7.8b}$$

where x symbolizes the priming signal and c the firing signal. With this the equation of the dynamic flip-flop:

$$Q = \neg (x_r \triangle c_r) \wedge (x_s \triangle c_s \vee Q') \tag{7.9a}$$

$$(x_s \triangle c_s) \wedge (x_r \triangle c_r) = 0 . \tag{7.9b}$$

The equation of the complementing (T type) flip-flop derives directly

$$Q = \neg (Q \triangle c) \wedge (\neg Q \triangle c \vee Q') \tag{7.10}$$

while (7.9b) is fulfilled automatically. As a consequence of (7.8 a—b), here going beyond (7.7)

$$\forall t \,(Q(t) \neq Q'(t) \Rightarrow Q(t) = 1 \quad and \quad Q'(t) = 2 \; or \; Q(t) = 0 \\ and \quad Q'(t) = 3) \tag{7.11}$$

With a view to practice, the following two relations have great significance. Starting from (7.5)

$$\begin{aligned} \triangle Q &= \triangle (\neg r \wedge (s \vee Q')) = \\ &= \neg r \wedge (s \vee Q') \wedge (\triangle \neg r \vee \triangle (s \vee Q')) = \\ &= \neg r \wedge (s \vee Q') \wedge \triangle \neg r \vee \neg r \wedge (s \vee Q') \wedge (\neg s \triangle Q') \\ &\quad \vee \neg r \wedge (s \vee Q') \wedge (\neg Q' \triangle s) \end{aligned}$$

where we relied mainly on (6.7b). Continuing:

$$\triangle Q = s \wedge \triangle \neg r \vee Q' \wedge \triangle \neg r \wedge s \wedge (\neg s \triangle Q') \vee \\ \neg r \wedge (\neg s \triangle Q') \vee \neg r \wedge (\neg Q' \triangle s) \vee \neg r \wedge Q' \wedge (\neg Q' \triangle s).$$

Now here we utilized (6.7b) and (6.3a). If we consider (7.8 a—b), (6.3 b), (6.4e) and (3.6c), then

$$\triangle Q = \neg r \wedge (\triangle Q' \vee \neg Q' \triangle s)$$

since obviously $⧅ \neg s = 1$. If now we exclude the t_0 times, already determined above, then $\triangle Q'$ also might be left out from beside $\neg Q' ◺ s$, then considering (7.4) again, we obtain:

$$\begin{aligned} \triangle Q &= \neg Q' ◺ s = \\ &= \neg Q' \wedge x_s ◺ c_s . \end{aligned} \tag{7.12}$$

After similar deduction it might be obtained that

$$\begin{aligned} \triangle \neg Q &= Q' ◺ r = \\ &= Q' \wedge x_r ◺ c_r . \end{aligned} \tag{7.13}$$

2. *Equation of shift-register*

If a shift register consists of the Q_i, $0 \leq i \leq n$ storage elemenst, and c is the shifting pulse, then

$$x_{si} = Q_{i-1}; \quad x_{ri} = \neg Q_{i-1}, \qquad 1 \leq i \leq n$$
$$c_{si} = c_{ri} = c, \qquad 0 \leq i \leq n.$$

With this

$$\begin{aligned} Q_i &= \neg(\neg Q_{i-1} ◺ c) \wedge (Q_{i-1} ◺ c \vee Q_i') = \\ &= (Q_{i-1} ◸ \neg c) \wedge (Q_{i-1} ◺ c \vee Q_i'), \qquad 1 \leq i \leq n \end{aligned} \tag{7.14a}$$

$$Q_0 = (x_{in} ◸ \neg c) \wedge (x_{in} ◺ c \vee Q'_0). \tag{7.14b}$$

3. *Simple binary counter*

The rule is known, according to which, if Q_i is the i-th storage element of a binary counter $(0 \leq i \leq n)$, then

$$s_i = \neg Q_i \wedge \bigwedge_{j=0}^{i-1} Q_j ◺ c, \qquad 1 \leq i \leq n \tag{7.15a}$$

$$r_i = Q_i \wedge \bigwedge_{j=0}^{i-1} Q_j ◺ c, \qquad 1 \leq i \leq n \tag{7.15b}$$

$$s_0 = \neg Q_0 ◺ c; \quad r_0 = Q_0 ◺ c \tag{7.15c}$$

where c is the symbol of the signal to be counted and

$$Q_i = \neg r_i \wedge (s_i \vee Q_i'), \qquad 0 \leq i \leq n.$$

If we introduce the notation

$$\bigwedge_{j=0}^{i-1} Q_j ◺ c = \triangle c_i, \qquad 1 \leq i \leq n, \qquad c_0 = c \tag{7.16}$$

then

$$s_i = \neg\, Q_i \llcorner c_i\,; \quad r_i = Q_i \llcorner c_i\,, \qquad\qquad 0 \leq i \leq n.$$

On the other hand

$$\begin{aligned} \triangle\, c_i &= \bigwedge_{j=0}^{i-1} Q_j \llcorner c = Q_{i-1} \wedge \bigwedge_{j=-1}^{i-2} Q_j \llcorner c = \\ &= Q_{i-1} \llcorner c_{i-1}\,; \quad Q_{-1} = 1\,, \qquad\qquad 1 \leq i \leq n. \end{aligned} \tag{7.17}$$

If (7.13) is applied to a complementing flip-flop, then

$$\triangle \neg\, Q = Q \wedge Q' \llcorner c = Q \llcorner c$$

as a consequence of (7.11). Comparing with the expression of $\triangle c_i$:

$$\triangle\, c_i = \triangle \neg\, Q_{i-1} \tag{7.18}$$

if complementing flip-flops are chosen for storage elements. With this

$$s_i = \neg\, Q_i \llcorner \neg\, Q_{i-1}\,; \quad r_i = Q_i \llcorner \neg\, Q_{i-1}\,, \qquad 1 \leq i \leq n \tag{7.19a}$$

$$s_0 = \neg\, Q_0 \llcorner c\,; \quad r_0 = Q_0 \llcorner c. \tag{7.19b}$$

4. *Elimination of triggering hazards*

In accordance with the remarks in the preceding chapter, hazards are mostly encountered if a signal, expressible in $a \vee b$, or $a \wedge b$ form, arrived to the input of a storage element or to a differentiating input, in which case $\square\, a \wedge \square\, b \neq 0$. (It is more suitable to attribute the other type hazards to the inappropriate frequency choice or the improper adjustment of delays.) In purely dynamic networks (that is, in which differentiating gates are connected to all storage element inputs) the above circumstance should be taken into consideration only at the differentiating inputs.

Take for example the realization of the expression

$$s = x \ulcorner c_1 \wedge c_2 \tag{7.20}$$

where s e.g. means the set condition, in negative logic, of a positive going edge triggered flip-flop. The functioning of the circuit in the usual semiconductor realization is obviously unreliable if for example $c_1 = 2$ and $c_2 = 3$ might occur at the same time.

Let us convert the (7.20) expression in such a way that the realization corresponding to the new expression, could work reliably even in case of slight timing errors!

On the basis of (6.3b)

$$s = \neg(\neg x \wedge \neg \square x \vee \triangle x) \vee \nabla (c_1 \wedge c_2)$$

then taking (6.7d)

$$s = \neg(\neg x \wedge \neg \square x \vee \triangle x) \vee (\neg c_1 \not\bigtriangledown c_2) \wedge (\neg c_2 \not\bigtriangledown c_1).$$

If we again apply (6.3b) and (6.6d), we obtain the expression

$$s = ((x \vee \neg c_1) \not\bigtriangledown c_2) \wedge ((x \vee \neg c_2) \not\bigtriangledown c_1) \tag{7.21}$$

as a final result. Though, this expression can be realized only by means of considerably more elements than the (7.20) expression, but its operation is free of any timing uncertainty.

*

In the paper, discarding the ordinary Boolean algebra, we introduced a new, four-valued Boolean algebra. By means of this, the analysis and synthesis methods, elaborated on the basis of ordinary Boolean algebra, can be generalized for dynamic networks too, opening the way to the algebraic design of dynamic networks.

Acknowledgements

I should like to express here my thanks to P. Köves and A. Münnich, who contributed with many good ideas to the results of this paper.

Summary

The paper discusses a formalism which, introducing four logical values, besides the logic levels, considers the edges too. The formalism is based on lattice theoretical methods and it is pointed out that the introduced structure is in effect a generalized Boolean algebra. After denominating the atoms and the canonical decompositions many relationships, important from a practical aspect, are discussed on the basis of the introduced operations. Finally some simple design examples are given.

References

1. SHANNON, C. E.: A symbolic analysis of relay and switching circuits. AIEE Trans., 57, 713 (1938).
2. YOELI, M.: Application of ternary algebra to the study of static hazards. J. ACM, 11, 84—97 (1964.)
3. Поспелов, Д. А.: Логические методы анализа и синтеза схем. Изд. Энергия, Москва, 1964.
4. HARRISON, M. A.: Introduction to switching and automata theory. McGraw-Hill Series in Systems Science, 1965.

5. Szász, G.: Bevezetés a hálóelméletbe. Akadémiai Kiadó, Budapest, 1959.
6. Post, E. L.: Introduction to a general theory of elementary propositions. Amer. J. Math. **43**, 163—185 (1921).
7. Gilbert, E. N.: Lattice theoretic properties of frontal switching functions. J. Math. Phys. **33**, 57—67 (1954).
8. Таланцев, А. Д.: Об алгебре потенциально-импульсных объектов. Докл. АН СССР, **139**, **6**. 1332—1335 (1961).
9. Таланцев, А. Д.: Об анализе и синтезе некоторых электрических схем при помощи специальных логических операторов. Авт. и телемех. **7**, 898—907 (1959).
10. Лазарев, В. Г.—Пийьл, Е. И.: Об интегрировании потенциально-импульсных форм. Докл. АН СССР, **139**, 556—559 (1961).
11. Лазарев, В. Г.—Пийьл, Е. И.: Упрощение потенциально-импульсных форм. Авт. и телемех., **2**, 271—276 (1963).

József Sárossy, Budapest, XI., Egry József u. 18—20. Hungary.