ANALYSIS OF COMBINATIONAL NETWORKS BY TOPOLOGICAL METHOD

By

I. VÁGÓ and I. PÁVÓ

Department of Theoretical Electricity, Technical University, Budapest

Received: June 16, 1981

Introduction

In 1958 at the Namur International Congress of Cybernetics, Professor Kalmár submitted a new construction for logical networks [1]. In his system the logical values are represented by special conductivity states between three poles while the two-argument Boolean operations can be realized by pure wire-boxes. The logical values and the general wire-box are illustrated in Fig. 1.

Remark that in the construction by Kalmár the negation is represented by reversing the three-poles corresponding to the logical variable and the general wire-box has two isolated three-terminal plugs for the right-argument operation. Generally, one of the two-terminal plugs can be omitted, except in cases of equivalence and antivalence. Fig. 2 shows ten various wire-boxes associated with the two-argument logical operations.

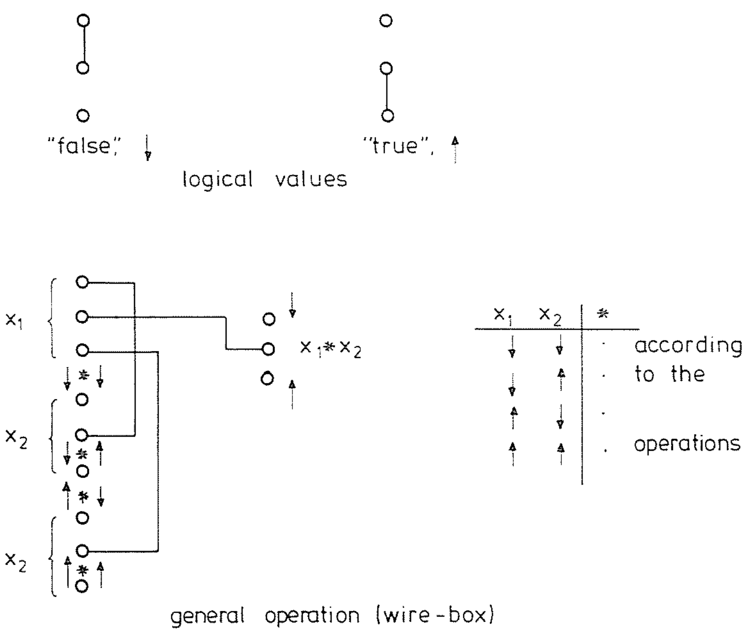

Fig. 1

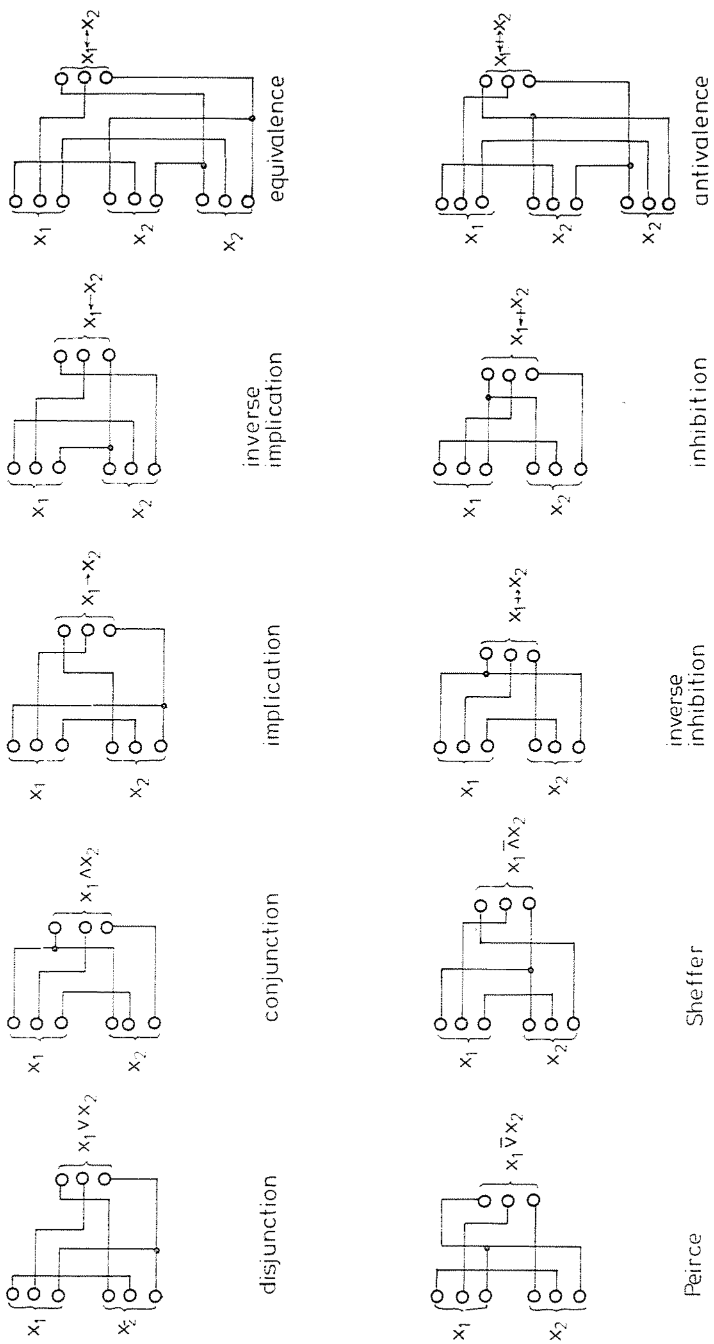
disjunction
conjunction
implication
inverse implication
equivalence
Peirce
Sheffer
inverse inhibition
inhibition
antivalence

Fig. 2

Consider a logical formula of n variables constructed by negation and any of the two-argument Boolean operations indicated in Fig. 2, and realize this formula by Kalmár boxes. Let this circuit be called Kalmár's network. Among others, the end value of a logical formula realized by Kalmár's network can be displayed on the Szeged Logical Machine [2] after choosing the input logical values of the variables. So this machine suits to establish the truth table of a logical formula.

In this paper a topological method is presented for the analysis of the Kalmár's network giving a disjunctive normal form of a logical formula. This method can be used for digital analysis as well. Since an arbitrary Boolean function can be realized by Kalmár's network, the present method can be regarded as a topological procedure for the analysis of combinational networks.

A property of the Kalmár network

Let us consider a Boolean function $F(x_1, \ldots, x_n)$ of n variables, negation and ten two-argument Boolean operations (Fig. 2), with a suitable Kalmár's network assigned to it. Terminate the input three-poles of the network by short-circuits in accordance with the actual truth values of the variables. Such termination of the network is called an allowed termination.

The following statement is valid:

A Kalmár's network completed by an allowed termination is always a 2-tree. The middle and exactly one of the outer points of the output three-poles belong to the same component of the 2-tree.

This property can be proved by inducation on the number of operations in formula F. If the formula contains only one operation the statement is trivially valid. Let us suppose that the number of the operations in the formula is $n \geqq 2$, and the statement is valid to formulas containing less than n operations. Without violating generality it can be supposed that the first symbol of the formula is not a negation and F contains at least one two-argument operation. Hence:

$$f(x_1, \ldots, x_n) = f_1(x_1, \ldots, x_n) * f_2(x_1, \ldots, x_n) \tag{1}$$

where Kalmár's networks concerning formulas f_1 and f_2 fulfil the induction hypothesis and $*$ denotes a two-argument operation.

Let us consider the Kalmár's network associated with the operation denoted by $*$. Completing the original Kalmár's network of F by an allowed termination the network associated with the operation $*$ is completed by an allowed termination as well. This termination can be simulated by suitable short-circuits on the inputs of the network $*$, the graph of which is in fact a 2-tree. After this, exchange these short-circuits with the suitable parts of the terminated Kalmár's network in that order corresponding to the simulation. So partly the original terminated Kalmár's network is obtained, and partly, the former graph contains neither circuits nor new components, which completes the proof.

This property is a trivial proof that Kalmár's networks are always a k-trees.

Derivation of the output function

This property of Kalmár's network offers a direct possibility to derive the output function of the combinational network.

Let us consider all possible allowed terminations of the Kalmár's network, and fix a termination. If the output function has a logical value "true" then there is a short-circuit between the middle and the lower poles of the output three-pole, otherwise there is an open-circuit. Since in the case of an output logical value "true", the upper point of the output three-pole belongs to one of the tree components, this point can be omitted from the network together with its tree component. Now there is a single path connecting the remaining output points. A conjunction can be constructed from the input logical variables (or their negations) corresponding to the series of short-circuits in the termination. Taking all the allowed terminations into consideration the disjunction the conjunctions formed along all the possible paths is really a disjunctive normal form of the output function.

Repeating these considerations for the middle and top points of the output, to the sense, the negation of the output function is constructed.

In order to take all the allowed terminations into account, terminate the input three-pole of the jth input variable by admittances $\bar{x}_j$ above and x_j below so that the possible values of the admittances are 0 or 1, and $\bar{x}_j * x_j$. Consider the graph of the network terminated in this manner. Omit from this graph the upper point of the output three-pole together with the connected input points and the admittances fitting them. Let us define a graph, called reduced graph, the edges of which are merely admittances, and the vertices are points of the above network short-circuited by wires. Topological formula for the output function:

$$F(x_1, \ldots, x_n) = \sum_i p_i(x_1, \ldots, x_n), \tag{2}$$

$$(\bar{x}_j * x_j = 0, \qquad \bar{x}_j \cdot x_j = \bar{x}_j, \qquad x_j \cdot x_j = x_j)$$

where $p_i(\bar{x}_1, \ldots, x_n)$ is a Boolean product of admittances formed along the ith path connecting vertices associated to the output points (output vertices) in the reduced graph, and the Boolean summation (2) affects all possible paths.

Examples

1. Let us construct the Kalmár's network shown in Fig. 3 from antivalence and conjunction boxes.

For producing a disjunctive normal form of the output function let us omit output point 13 together with points 4, 7 and 12 which are connected to it (See Fig. 4). Terminate the remaining network inputs by the proper admittances. Fig. 5 shows the reduced graph.

Now there are two paths between the vertices associated with the output points 14 and 15. By the earlier convention for the admittances and this expression has but one disjunction term $x\bar{y}$ logical the formula obtained is a conjunction, namely

$$F(x, y) = (x \leftrightarrow y) \wedge x = x\bar{y} .$$

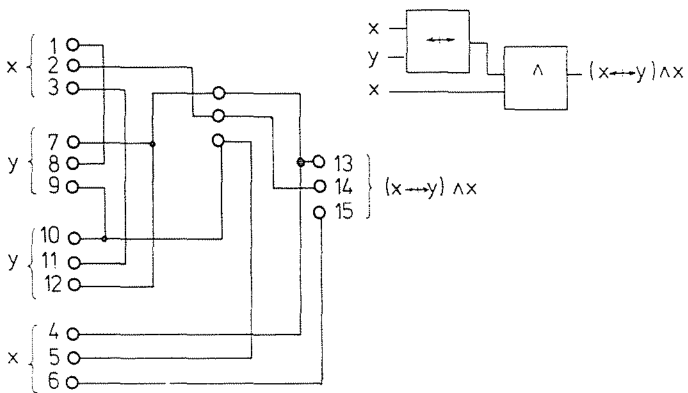

Fig. 3

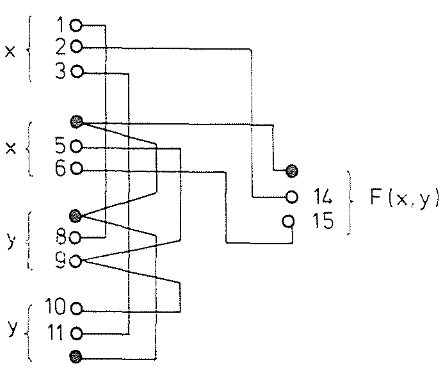

Fig. 4

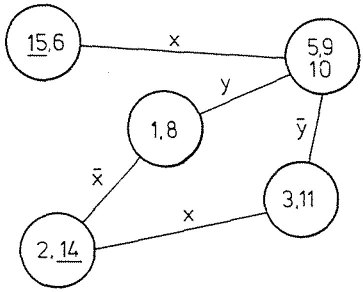

Fig. 5

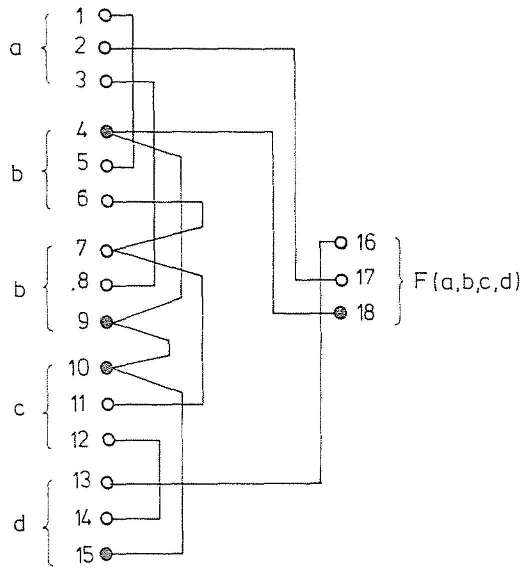

Fig. 6

2. Consider the Kalmár's network in Fig. 6. Now it is more practical to write the negated output function, for omitting point 18 yields a simpler reduced graph than omitting point 16. Fig. 7 shows the reduced graph for establishing the formula $\bar{F}(a, b, c, d)$. There are two paths between the vertices associated with 16 and 17, but both provide disjunctions, so we have:

$$\bar{F}(a, b, c, d)=\bar{d}c\bar{b}a+\bar{d}cb\bar{a} .$$

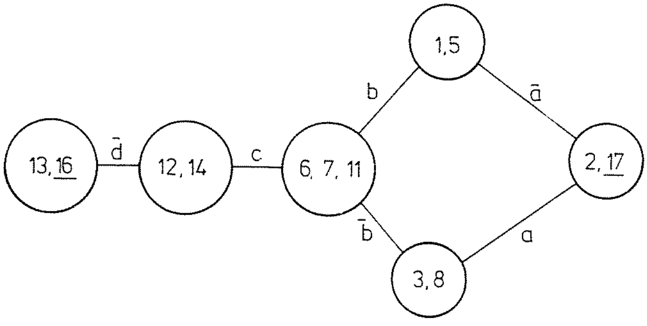

Fig. 7

Computer analysis

A simple flow chart of the procedure necessary for the analysis is illustrated in Fig. 8. For computer implementation it is most essential to generate the edge admittance product $p_i(\bar{x}_1, \ldots, x_n)$ formed along the path connecting the output vertices.

The generation is possible in several ways. The results of applied graph theory can be used to search the paths. The problem can be reduced to tree

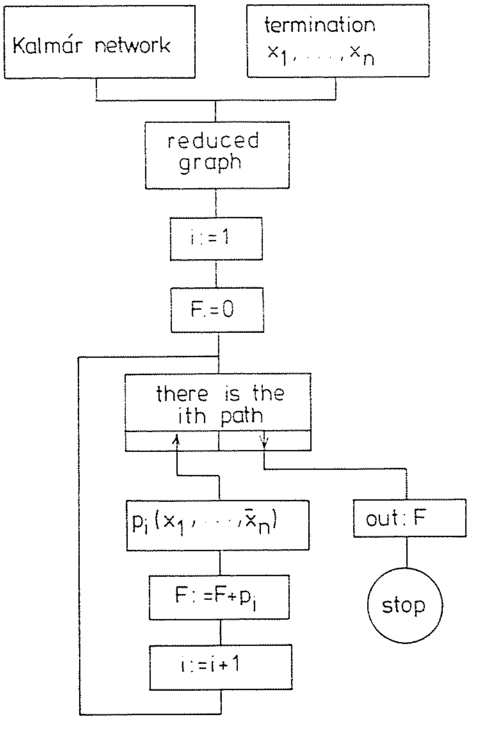

Fig. 8

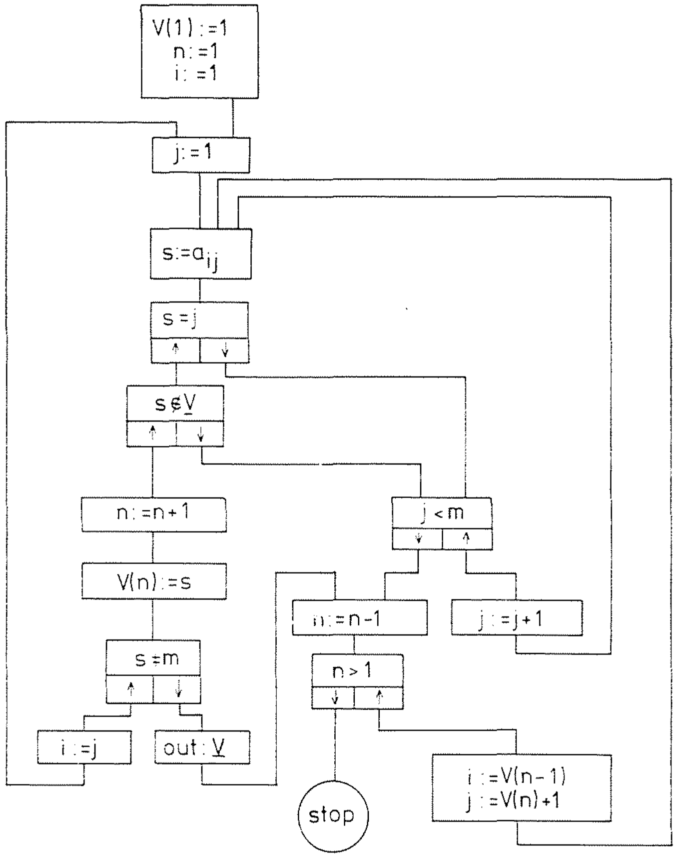

Fig. 9

generation of a graph since any path between the output vertices can be included into a tree of the reduced graph. For such a generation the method presented in [3] may be applied. In this case it is very practical to take the vertex associated with the middle point of output as the negative root. So the cycle check starting from the outside vertex, part of the complete cycle check, gives exactly the searched path. This method is very elegant but from the point of view of computer technique it has the inconvenience that a path can occur more than once, so that finally the equivalent terms of the disjunctive normal form have to be contracted.

There is a possibility to produce a single copy of each path wanted from the modified adjacency matrix of the reduced graph [3]. For this purpose let the outside output vertex be marked by 1 and the middle one by m in the reduced graph, where m is the number of vertices. Starting from the first row of the adjacency matrix, all paths leading from 1 to m can be obtained one by one.

A flow chart is seen in Fig. 9 for the construction of a computer program. The input data of the program are elements a_{ij} of the modified adjacency matrix of size $m \times m$ while the output is the actual vector V of size n, the components of which are the vertices of the path (n is the number of the vertices in the path, wanted $n \leqq m$). For the conjunction of the admittances the pairs of the vertices may be produced from the elements of V in suitable order.

Summary

In this paper combinational circuits are realized by Kalmár's network. The Kalmár's network completed by an allowed termination is shown to be always a 2-tree, so a topological formula can be constructed for the output Boolean function of a combinational network. Simple examples are given on writing the output function by this method, and finally a flow chart for computer technique is presented.

References

1. Kalmár, L.: A new principle for construction of logical machines; Proc. of The 2nd International Congress of Cybernetics, Namur, 1958, pp 458—463.
2. Nemes, T.: Cybernetic Machines (in Hungarian); Publ. House of the HAS, Budapest, 1962, pp 93—95.
3. Pávó, I.: Generation of the k-trees of a graph; Acta Cybernetica, Szeged, Tom. 1, Fasc. 2, 1971, pp. 57—68.

Prof. Dr. István Vágó, Budapest H-1521
Dr. Imre Pávó, Szeged, 6720, Somogyi út 7.