

А. Черныш
12.8.82

ЧИТАЛЬНИК
ЗАМ

ISSN 0005-1179

Vol. 42, No. 11, Part 2, November, 1981

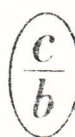
April 20, 1982

AURCAT 42(11) 1509-1582 (1981)

AUTOMATION AND REMOTE CONTROL

АВТОМАТИКА И ТЕЛЕМЕХАНИКА
(AVTOMATIKA I TELEMEXHANIKA)

TRANSLATED FROM RUSSIAN



CONSULTANTS BUREAU, NEW YORK

AUTOMATION and REMOTE CONTROL

АВТОМАТИКА
и
ТЕЛЕМЕХАНИКА

A translation of *Avtomatika i Telemekhanika*

April 20, 1982

Volume 42, Number 11, Part 2

November, 1981

CONTENTS

Engl./Russ.

ADAPTIVE SYSTEMS

- Convergence of Algorithms of Direct Adaptive Control under Disturbances—B. V. Pavlov
and I. G. Solov'ev 1509 96

EVOLVING SYSTEMS

- A Primal Decomposition Method for Linear Programs. I. Description of the Algorithm
—V. E. Krivtsov 1516 104
- Investigation of Nonlinear Optimization Models of Growth in a Multisectoral Economy.
II—V. F. Krotov 1524 114

AUTOMATA

- Bounds on the Realization Complexity of Boolean Formulas by Tree Circuits of
Tunable Modules—V. L. Artyukhov, G. A. Kopeikin, and A. A. Shalyto 1532 124
- Construction by Linear Cellular Automata—S. L. Blyumin 1538 131

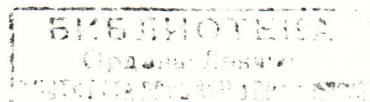
TECHNICAL DIAGNOSTICS

- Properties of Multiple Faults in Logic Circuits with Contacts—T. G. Melkadze,
V. V. Sapozhnikov, and V. V. Sapozhnikov 1545 139
- Self-Checking Computer Circuits and Systems (Review)—E. V. Slabakov
and E. S. Sogomonyan 1551 147

COMPUTING TECHNIQUES IN AUTOMATIC CONTROL

- Structural Approach to Simulation of Complex Distributed Systems. I—I. Begimov,
A. G. Butkovskii, and V. L. Rozhanskii 1567 168
- Accuracy of Simulation of Hybrid Control Systems with a Digital Controller—A. B. Danilin,
and A. G. Spiro 1577 182

The Russian press date (podpisano k pechati) of this issue was 9/29/1981.
Publication therefore did not occur prior to this date, but must be assumed
to have taken place reasonably soon thereafter.



~~April 20, 1982~~
~~11/1/1981~~

~~1538-1537~~

BOUNDS ON THE REALIZATION COMPLEXITY OF
BOOLEAN FORMULAS BY TREE CIRCUITS OF
TUNABLE MODULES

V. L. Artyukhov, G. A. Kopeikin,
and A. A. Shalyto

AuT

UDC 62-507

Upper and lower bounds are established on the circuit realization complexity of Boolean formulas in the basis of universal tunable modules.

Extensive literature is now available on design and application of tunable logic modules (TLM) [1-4]. There are, however, virtually no published results on the complexity of circuit realizations in the TLM basis.

The present article partly fills this gap for the case when the functioning algorithm of the device is represented by an h -letter Boolean formula defined in a basis with associative two-place operations and realized by a tree circuit consisting of universal TLMs in the same class of formulas with k or fewer letters [4].

1. STATEMENT OF THE PROBLEM

Consider an h -letter Boolean formula $f(z_1, z_2, \dots, z_n)$ defined in a basis with associative two-place operations (e.g., $\{\&, \vee, -\}$, $\{\&, \vee, -, \oplus\}$).

Also given is a collection of modules M realizing all the subformulas φ in this basis of length not exceeding k letters. A universal TLM in this class of formulas with k or fewer letters may be used to represent these modules. In what follows, we refer to it as the k -universal module.

From the set of circuit realizations of the formula $f(z_1, z_2, \dots, z_n)$ in the basis M , we isolate the subset of tree circuits.

Definition. A tree circuit is a single-output loopless structure in which every input variable and the output of every element are connected directly with at most one input of a single element in the structure.

Among the tree realizations, there is at least one with a minimal number of modules.

Let us estimate the number of modules $L(h; k)$ from the set M required to construct a given realization.

We denote the inputs of the tree structure by x_i , where $i = 1, \dots, h$. Then the original formula $f(z_1, z_2, \dots, z_n)$ is transformed into a repetition-free formula of the form $f(x_1, x_2, \dots, x_h)$.

On the other hand, we know [4] that a module is universal in some class of k -letter formulas only if its generating function is a combination of k -letter repetition-free formulas in the same basis.

Our problem thus reduces to decomposition of a repetition-free formula into repetition-free subformulas.

2. THE FINDINGS

Proceeding to discuss the findings, we should note that the TLM inputs may act both as information inputs and as tuning inputs. The TLM is linked with information sources and preceding modules in the logic structure by means of the information inputs. A TLM logic structures may be represented omitting the tuning inputs if each module is marked with the formula that is realized.

In what follows, we will only focus on information inputs, which we call module inputs. The module inputs to which input variables are applied are called activated; the remaining inputs are called free.

Proposition 1. The number of modules in a tree structure is given by

Leningrad. Translated from *Avtomatika i Telemekhanika*, No. 11, pp. 124-130, November, 1981. Original article submitted September 12, 1980.

$$L = \frac{h-1}{k_{av}-1}$$

where k_{av} is the average number of activated inputs in one module.

Proposition 2. The minimal number of modules in a tree structure is given by

$$L_{\min} = \left\lceil \frac{h-1}{k-1} \right\rceil,$$

where $\lceil A \rceil$ denotes rounding to the nearest integer not less than A .

Proposition 3. The minimal average number of activated inputs in a single module in a tree structure is given by

$$K_{av \min} = \begin{cases} \frac{k}{2} + 1, & \text{if at least one of the numbers } L \text{ or } K \text{ is even;} \\ \frac{k}{2} + 1 - \frac{1}{2L}, & \text{if both } L \text{ and } K \text{ are odd,} \end{cases}$$

and the maximal number of modules in an optimal tree structure is given by

$$L_{\max} = \left\lceil \frac{2(h-1)}{k} \right\rceil.$$

Propositions 1-3 are proved in the Appendix.

Thus the number of modules in a minimal tree structure is bounded by

$$\left\lceil \frac{h-1}{k-1} \right\rceil \leq L \leq \left\lceil \frac{2(h-1)}{k} \right\rceil.$$

A method for realization of formulas in this TLM basis which attains the above bounds is presented in [5].

In conclusion note that our bounds can be extended to a larger class of modules, namely universal modules in the class of all Boolean functions. Since these modules, in particular, are universal in the class of formulas in the basis $\{\&, \vee, -, \oplus\}$, we have the following bounds on the realization of an arbitrary n -variable function defined by an h -letter formula in the same basis:

$$\left\lceil \frac{n-1}{k-1} \right\rceil \leq L \leq \left\lceil \frac{2(h-1)}{k} \right\rceil.$$

APPENDIX

Proof of Proposition 1

Without loss of generality, we can consider the case when the formulas f and φ are positive monotone.

Suppose that some method (e.g., enumeration) applied to a given formula has produced a minimal tree circuit consisting of $L(h; k)$ k -universal modules. Let us analyze the resulting circuit in order to estimate the number of constituent modules.

In a tree structure there is at least one element with all the activated inputs connected only to sources of input variables. We assign the number 1 to this element and denote the number of its activated inputs by k_1 . Clearly, $k_1 \leq k$.

Element 1 realizes some repetition-free formula φ_1 consisting of k_1 letters. Substituting φ_1 for the corresponding group of letters in the original formula $f(x_1, \dots, x_h)$, we obtain a new repetition-free formula $f_1(\varphi_1, x^{(2)})$ consisting of $h - k_1 + 1$ letters.

In the remaining part of the structure there is an element (to which we assign the number 2) whose input variables belong to the set formed by the $h - k_1$ free inputs left after the extraction of the first element plus the first element output. Suppose that element 2 realizes a repetition-free formula φ_2 consisting of k_2 letters. Substituting φ_2 for the corresponding group of letters in the formula $f_1(\varphi_1, x^{(1)})$, we obtain either a formula $f_2(\varphi_1, x^{(2)})$, or a formula $f_2(\varphi_2, x^{(3)})$, both consisting of $h - k_1 - k_2 + 1 + 1 = h - k_1 - (k_2 - 1) + 1 = h - (k_1 + k_2) + 2$ letters.

Continuing the structural analysis, we finally "traverse" the entire L -element structure and arrive at the single output. At this point we have