

12.9.85

ISSN 0005-1179

Vol. 45, No. 11, Part 2, November, 1984

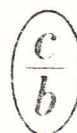
April 20, 1985

AURCAT 45(11) 1457-1536 (1984)

AUTOMATION AND REMOTE CONTROL

АВТОМАТИКА И ТЕЛЕМЕХАНИКА
(AVTOMATIKA I TELEMEXHANIKA)

TRANSLATED FROM RUSSIAN



CONSULTANTS BUREAU, NEW YORK

AUTOMATION and REMOTE CONTROL

A translation of *Avtomatika i Telemekhanika*

April 20, 1985

Volume 45, Number 11, Part 2

November, 1984

CONTENTS

Engl./Russ.

EVOLVING SYSTEMS

- Elements of the Optimal Synthesis Theory for Functioning
Mechanisms of Two-Level Active Systems. II. Synthesis
of Optimal Correct Functioning Mechanisms under Complete
Information at the Headquarters — V. N. Burkov,
A. K. Enaleev, V. V. Kondrat'ev, and A. V. Tsvetkov..... 1457 86
- A Multicommodity Network Flow Problem with Commodity Conversion
and Dependent Flows — G. E. Moiseenko..... 1464 93

AUTOMATIC CONTROL IN BIOLOGICAL SYSTEMS

- Optimization of Radiation Tumor Therapy with Photon and
Corpuscular Beams — V. K. Ivanov..... 1471 101
- Properties of a Model of an Epidemic for Certain Infectious
Diseases — A. A. Klement'ev and A. B. Stupin..... 1475 106

AUTOMATA

- Tunable Binary Procedures and Programs with Loops
— V. L. Artyukhov, B. P. Kuznetsov, and A. A. Shalyto..... 1481 112
- Realization of Finite-Automaton Algorithms by Structured
Programs. II — V. V. Devyatkov and V. N. Shinkin..... 1488 120
- A Generalized Language of Parallel Compositions — N. N. Ivanov..... 1497 130

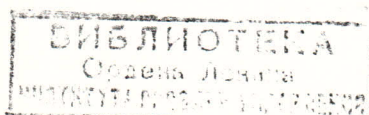
TECHNICAL DIAGNOSTICS

- A Variational Approach to Hazard Detection in Asynchronous
Sequential Circuits — A. S. Barashko..... 1506 139
- Simulation of Digital Devices with Long Input Sequences
— A. G. Birger..... 1513 147
- Unconstrained Algorithm for Multiple Fault Detection in
Combinational Networks — V. V. Kiselev and E. L. Kon..... 1520 155

COMPUTING TECHNIQUES IN AUTOMATIC CONTROL

- Effectiveness of Application Software Technology for Control
Systems — V. V. Lipaev..... 1528 164

The Russian press date (podpisano k pechati) of this issue was 10/17/1984.
Publication therefore did not occur prior to this date, but must be assumed
to have taken place reasonably soon thereafter.



(4)

AUTOMATION AND REMOTE
CONTROL

BINARY PROCEDURES AND PROGRAMS WITH LOOPS

L. Artyukhov, B. P. Kuznetsov,
A. A. Shalyto

April 20, 1985

UDC 681.3.06

English 1481-1488

RUSS 112

Method is proposed for the evaluation of a system of Boolean formulas using a program with a loop in which some parameters generated by a previously created program are successively passed to a special procedure. The program with a loop is called a cyclic binary program, and the procedure is termed a tunable binary procedure. Basic design principles and complexity bounds are considered.

INTRODUCTION

There is an ever-growing interest in program realization of systems of Boolean functions using compilers [1, 2] and interpreters [3]. Interpretation is the more promising in those cases when no special requirements are imposed on the time characteristics of the program. An interpreter is independent of the Boolean functions being realized, and conditions are specified by an array [3]. However, the generation and updating of such programs for the specification of SBF with many variables and formulas is an arduous and summing task. If we can abstract from issues of time and memory optimization, which is the primary concern of [1, 3], and focus on simplicity of the realizing procedure (as in the formula method proposed in [1]), the interpreter may be designed as a cyclic program which includes a general-purpose compiler procedure with parameters. Then the specified SBF is represented by a table (array) of parameters, and formula evaluation reduces to sequential enumeration of the table rows and substitution of the row parameters into the general-purpose procedure.

In this article we consider some principles of organization and construction of tunable procedures, parameter tables, and cyclic binary programs for SBF evaluation. Approximate complexity bounds are given.

Simple Binary Procedures

Consider a SBF of the form

$$\begin{aligned} W_1 &= Z_1 \vee Z_2 \vee Z_3; & W_2 &= Z_1 Z_2 \vee Z_3; \\ W_3 &= (Z_1 \vee Z_2) Z_3; & W_4 &= Z_1 Z_2 Z_3. \end{aligned} \quad (1)$$

Figure 1 presents graph-schemas (GS) of simple binary programs realizing these formulas. We combine the GS of the separate formulas into a generalized GS (Fig. 2a), which constitutes a simple binary program with four inputs and one output (Fig. 2b). To reduce the number of inputs, we introduce the additional variables Z_4 and Z_5 , which are called formula-parameters: $Z_4 = Z_5 = 1$ tune to W_1 , $Z_4 = 1$ and $Z_5 = 0$ tune to W_2 , $Z_4 = 0$ and $Z_5 = 1$ tune to W_3 , and $Z_4 = Z_5 = 0$ tune to W_4 . Supplementing the GS with three nodes (Fig. 2c), we obtain the GS of a binary program which constitutes a procedure evaluating any of the formulas in (1); this procedure may be utilized as a subprogram with the parameters Z_4 and Z_5 and may be designed as a function subprogram since computations with any of the formulas will give the value of W equal to 1 (0). The resulting function procedure (Fig. 2c) will be called a tunable binary procedure (TBP), which we denote by $W = W(Z_4, Z_5)$. Its tuning is accomplished by assigning the constant values 1 and 0 to the parameters Z_4 and Z_5 . The formulas (1) are evaluated using the following relationships:

$$W_1 = W(1, 1); \quad W_2 = W(1, 0); \quad W_3 = W(0, 1); \quad W_4 = W(0, 0). \quad (2)$$

The GS in Fig. 2c realizes the function

Inggrad. Translated from *Avtomatika i Telemekhanika*, No. 11, pp. 112-119, November, 1983. Original article submitted November 10, 1983.