

A Class of Logic Design Problems Solved Based on Parallel Computations of „Butterfly” Configurations

V. Shmerko, S. Yanushkevich, K. Malecki

Institute of Computer Science, Technical University, Al. Piastow 41, 71-065, Szczecin, Poland, tel(+4891)336214, fax(+4891)340946, e-mail:shmerko@beta.ii.tuniv.szczecin.pl

Abstract

The class of logic problems whose decision is based on parallel algorithms described by signal flow graphs of a „butterfly” type, has been extended. This class consists of the problem of modern Computer-Aided Design (CAD) of discrete devices: polynomial representations of logic functions, Logic Differential Calculus, solving Logic and Differential Logic equations and others. It means that architectural principles, hardware and software to realise the fast algorithms well-known in digital signal processing, can be used to solve the problems.

Introduction

Modern Logic design is of inter-disciplinary form. Parallel computations is one of main it's principles, since the size of the decided problems are large enough. There is a class of CAD tasks that can be characterised by the fact that its computational processes are like a signal flow graph configuration, such as well-known „butterfly” of the fast Fourier transformation. It means that known and proven architectural principles of hardware and software for fast transformation (Fourier, Walsh, Chrestenson and its later modifications) can be applied to solve this class of logic tasks. This paper systematized authors' well-known results from the point of adequate signal flow graph configuration, i.e. the paralleling of computational process.

1. General characteristics of this class

The class, named below as the „butterfly” class, is formed by four groups of tasks. Two are concerned with CAD with multivalued devices realising multivalued logic functions. The next two groups are concerned with CAD for traditional binary gates (Fig. 1).

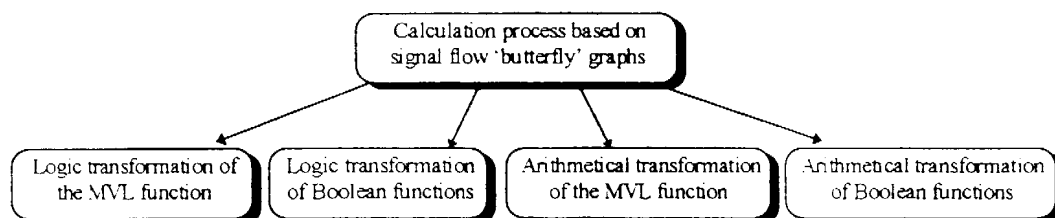


Fig. 1. The four classes of calculation processes realised on the base of signal 'butterfly' graphs

Most of all, an operation based on „butterfly” graphs is applied in Logic Differential Calculus. The Logic Differential operators are applied in analysis, finding test for binary and Multivalued switching circuits [1-3]. Computing the values of a Logic Derivative of a k -valued function with respect to a variable defined by a matrix-vector procedure, corresponding to an iteration of k -point fast Fourier transform [4]. In particular, computing the values of m -ordered Derivative likes m -iteration fast transform. The same is true for so-called Arithmetical Derivative of an MVL-function with respect to variables, i.e. analogues of the Logic Derivatives [5]. The synthesis of various classes of Polynomial forms of MVL-functions is also represented by graphs of „butterfly”

configuration. These are such forms as generalised Reed-Muller canonical forms with fixed and mixed polarity [6], Arithmetical polynomial forms with different types of polarity [7], and extension of the class for GF(k) field, including non-polynomial and polynomial-similar forms. So, all computational procedures over logic data have their analogies in Arithmetical logic.

2. Parallel computations of polynomial forms for logic functions

The term „Polynomial Forms of logic functions” describes a large class both Boolean and Multivalued functions which can be represented by the Reed-Muller expansion, its generalisations [6], and its arithmetical analogues [5]. Parallel computation of these forms denotes the problem of their synthesis and the inverse problem - recognition of a logic function from the form. The general scheme of the computations is shown in Fig.2.

Direct and inverse discrete orthogonal transformation can be executed based on fast transformation which are reduced to the well-known „butterfly” configurations. In other words, any computing facilities to fulfil the classic fast transform, in principal, can be used to synthesise polynomial forms of logic functions [6,8,9].

3. Parallel computations of Logic Derivatives

The central question of the applied Logic Differential Calculus theory is computing the differential operators. It was shown [9] that practically all Boolean differential operators are reduced to computational processes based on a „butterfly” configuration. So for example, truth vectors of Boolean Differences of a Boolean function $f(X)$ of n variables with respect to variable x_1 (x_2), $\partial f(X)/\partial x_i$, is computed based on the simplest „butterfly” graph (Fig.3). Computation of m -order Boolean Differences corresponds to an m -iteration fast transform consisting of „butterfly” - based operations. The natural mapping of these operators is linear systolic array of m Computing Cells (CC) [11].

Computation of Boolean Differentials with respect to variables and Full Differentials applied to find tests for switching circuits [1], has quite different features. So, the truth vector of the Full Boolean Differential of a function of n variables includes (2^n-1) truth vectors of Boolean Differences with respect to all possible variables and vectors of variables. A tree-network systolic array to compute Boolean Differentials was proposed [12]. Each cell in such a structure realises an iteration represented by the „butterfly” (it corresponds to computing a difference with respect to one variable). Such structure when $n=3$ is given in Fig.4. Note, that this array is used to find tests for switching circuits. All considered above is true for arithmetical analogs of Boolean differential operators.

The pattern of the signal flow graph for Logic Derivatives of Multivalued (k -valued) logic functions [5,6] is like the graph of fast Fourier transformation over GF(k). In particular, the signal flow graph corresponded to computing the Logic Derivative of a 3-valued logic function with respect to variables x_1, x_2 with 1- and 2-order cyclic inversion is shown in Fig.5. The computation of Partial Direct and Inverse Logic Derivatives for a k -valued function [4] is simple enough to reach the „butterfly” (Fig.6).

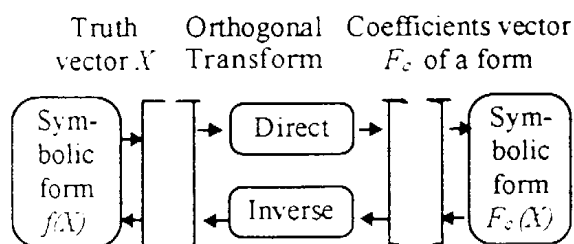


Fig.2. Scheme of applying orthogonal transformation for logic polynomial forms

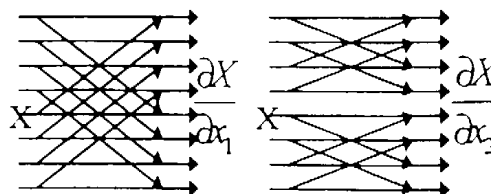


Fig.3. Signal flow graphs of computing Boolean Differences with respect to variables $x_i, i=1, 2, 3, n=3$

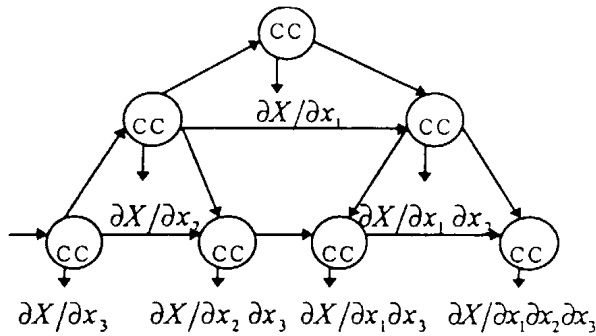


Fig.4. Tree-network systolic array to compute Full Boolean Differential, $n=3$

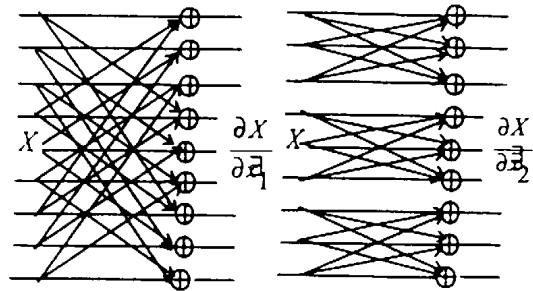


Fig.5. Signal flow graphs of computing Logic Derivatives

4. Computation of prime implicants when minimising Boolean functions

As it was shown in [13], the computed prime implicants can be executed based on the signal flow graphs of a „butterfly” type.

A linear systolic array to realise the computational process is represent in Fig. 7.

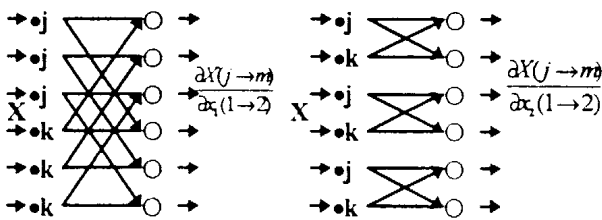


Fig. 6. Signal flow graphs of computing Partial Logic Derivatives, $k=3, n=2, i=1,2$

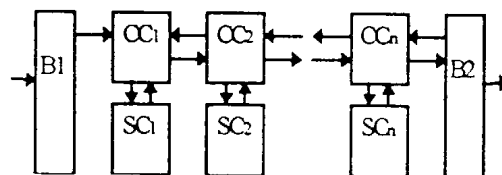


Fig. 7. Linear systolic array to compute prime implicants

5. Solving Logic and Differential Logic Equations

In Logic Design, testing and the diagnostics of digital devices it is necessary to solve Logic and Differential Logic equations [1]. The problem is in the fact that the size of the problem is quite large. Therefore it needs to decompose the initial systems of the equations. In this case, methods of paralleling are efficient enough.

To solve the Logic equation or a system of the equations represented by the Reed-Muller form or other polynomial forms, it is necessary to fulfil inverse discrete orthogonal transformations in a corresponding basis [9,14]. It means that algorithms to realise such decision are described as fast transformations and „butterfly” graphs and are mapped into parallel and pipelining structures, in particular, linear systolic arrays. So, to solve Boolean Differential Equations, homogeneous parallel array was proposed [9]. The focus was the fact that solving these Equations was come to deciding a system of linear Logic Equations of Reed-Muller canonical form. $2^n \times 2^n$ array was needed to solve the Logic Differential Equation of n variables. However, in [14] a reduced algorithm generalised for k -valued functions, was proposed where the problem was divided into sub-tasks, each of them requiring $n \times n$ sub array or even linear array of n elements to be realized

Conclusions

The authors have obtained the following main results:

- 1) a class of logic problems described by the algorithms based on signal flow graphs of the „butterfly” type, were extended; besides polynomial transformations, the Logic Differential Calculus problems and some stages of Boolean functions minimisation, are included;

- 2) a technique was worked out to meet the class of logic problems to the computational procedures of the „butterfly” type; this technique is based on matrix mathematical tools;
- 3) a number of new generalisations for Arithmetical Logic (as analogues of logic methods and algorithms) was proposed; this means practically, that some logic problems can be solved on the basis of the same flow graphs, with logic or arithmetical operations in the graph nodes;
- 4) The results of this paper change the established views on the character of computational processes in Logic Differential Calculus for Multivalued functions; these processes have also come to „butterfly” configurations.

References

- [1] D. Bochmann, Ch. Posthoff, *Binäre dynamische Systeme*, Springer-Verlag, Berlin, 1981.
- [2] Shmerko V., Yanushkevich S. Fault Detection in Multivalued Logic Networks by New Type of Derivatives of Multivalued Functions. Proc. of the 11th European Conf. on Circuit Theory and Design. Part 1. Switzerland, 1993, p.643-646
- [3] Shmerko V., Yanushkevich S., Levashenko V., Bondar I. Technique of Computing Logic Derivatives for MVL-Functions, Proc. of the 26th Int. Symposium on Multiple-Valued Logic (to be published)
- [4] Levashenko V., Zaitseva E., Yanushkevich S., Wojciechowski D., Synthesis of Parallel Algorithms to Compute Derivatives of MVL-functions, 1995 Workshop on Sampling Theory & Applications, Latvia, Riga, 1995, pp.319-322
- [5] Yanushkevich S. Arithmetical Analogues of Reed-Muller Canonical Expansions with fixed Polarity for MVL-functions, Proc. of the Workshop on Applications of Reed-Muller Expansions in Circuit Design, Tokyo, August, 1995, p.300-307
- [6] Yanushkevich S. Spectral and Differential Methods to Synthesize Polynomial Forms of MVL-functions on Systolic Arrays. Proc. of the Fifth Int. Workshop on Spectral Techniques, Editors: C. Moraga, Z. Gongli, etc., Beijing, China, 1994, p. 78-93
- [7] Antonenko V., Ivanov A., Shmerko V., Linear Arithmetical Forms of k-valued logic functions and their realisation on Systolic Arrays, *Automatics and Telemechanics*, Russian Academy of Sciences, No.3, 1995, p. 139-155 (In Russian). Translat. Remote Control (USA)
- [8] Bondar I., Kaczmarek A., Yanushkevich S., Shmerko V. State of the art to use discrete orthogonal transforms for Canonical Expansions of Boolean functions. Proc. of the Int. Conf. on Computer-Aided Design of Discrete Devices -CAD DD'95, Minsk, Republic of Belarus, 1995, p.43-52
- [9] Kukharev G., Shmerko V., Yanushkevich S. Technique of Binary Data Parallel Processing on VLSI, *Vysheyshaya Shkola*, Minsk (1991) 220p. (In Russian)
- [10] Shmerko V., Yanushkevich S., "Binary image processing by Boolean Differential Calculus operators and its realization in linear systolic procesors", *Pattern Recognition and Image Analysis*, 4, pp.406-422, 1991.
- [11] Shmerko V., Yanushkevich S., Kochergov E. Systolic Arrays for Binary Image Processing by Using Boolean Differential Operators. Proc. of the int. Conf. on High Definition Video, Proc. SPIE, Vol 1976, 1993, p.324-335
- [12] Kaczmarek A., Majka E., Yanushkevich S. Tree-network Systolic Architecture for Computing Test Vectors of Combinational Circuits. Proc. of the First Int. Conf. on Applications of Computer Systems- ACS'94, Poland, Szczecin, 1994, p.55-58
- [13] Bondar I. and W Riesselmann. Calculation of Prime Implicants of Boolean Functions on Systolic Arrays. Proc. of the Int. Conf. on Computer-Aided Design of Discrete Devices - CAD DD'95, Minsk, Republic of Belarus, 1995, p.53-58
- [14] Yanushkevich S., Shmerko V., Wojciechowski D., Pottosina S. Logic Differential Equations in Multivalued Logic Systems, Proc. of the Second Int. Conf. on Applications of Computer Systems- ACS'95, Poland, Szczecin, 1995, p.123-130