Minimization of Permuted Reed-Muller Trees for Cellular Logic Programmable Gate Arrays

Li-Fei Wu, and Marek A. Perkowski

Department of Electrical Engineering, Portland State University
P.O. Box 751, Portland, Oregon 97207, tel. (503) 725-5411

ABSTRACT

The new family of Field Programmable Gate Arrays, CLI6000 from Concurrent Logic Inc realizes the truly Cellular Logic. It has been mainly designed for the realization of data path architectures. However, introduced by it new universal logic cell calls also for new logic synthesis methods based on regularity of connections. In this paper we present two programs, exact and approximate, for the minimization of Permuted Reed-Muller Trees that are obtained by repetitive application of Davio expansions (Shannon expansions for EXOR gates) in all possible orders of variables in subtrees. Such trees are particularly well matched to both the realization of logic cell and connection structure of the CLI6000 device. It is shown on several standard benchmarks that the heuristic algorithm gives good quality results in much less time than the exact algorithm.

1. Introduction

There is recently an increased interest in logic synthesis using EXOR gates [1,3-14]. New technologies, PLD and FPGA, either include EXOR gates, or allow to realize them in "universal logic modules". Exor circuits have smaller cost than inclusive (AND/OR) circuits. They are always very easily testable and have universal tests [5,13]. One way to realize EXOR-based circuits is through GRMs. The Generalized Reed-Muller (GRM) forms, which also include Reed-Muller (RM) form [9], are canonical. All literals in the RM form are positive. In GRM forms all variables are in a fixed form, i.e. each occurrence of a variable in products of the form is either consistently positive or consistently negative. The problem of finding the minimal

\[ \text{This research was partially supported by the NSF grant MIP-9110772}. \]
GRM form of optimal polarity [13] (called also fixed-polarity Reed-Muller), as well as the problem of finding the minimal Exclusive-OR Sum of Products (ESOP) of a Boolean function [1,7,10], are the classical ones in logic synthesis theory. Recently efficient solutions have been proposed: to ESOPs in [1], and to GRMs in [13]. However, to our knowledge, with an exception of [14] and [12] there are no any programs to minimize multi-level (more than three levels) circuits that use EXOR gates.

Recently several companies, including Concurrent Logic [2] and Algotronix brought to market a new generation of FPGAs that can be called 'cellular', since the number of global connections for a cell is very limited and most of the connections are only to the abutting cells. CLI chip is a rectangular array of cells [2]. The basic cell of CLI6000 can be programmed to the two-input multiplexer \((A \cdot B \oplus C \cdot B)\) and the AND/EXOR cell \((A \cdot B \oplus C)\) as two of its most efficient combinational logic uses. This suggests using these cells for tree-like expansions. The method outlined in this paper is particularly suitable for these new FPGAs for which no special design methods have been yet proposed. These "cellular logic" devices require regular connection patterns in the netlists resulting from logic synthesis.

The AND/OR factorization (MIS II) creates irregular circuits that are hard to map to CLI6000 cells. ESOP factorization approach from [12] creates multi-level circuits that better utilize AND/EXOR cells but are still irregular. This makes their routing to cellular logic devices (specifically CLI6000) extremely difficult and many cells are wasted while programmed as connections only. Several concepts of logic synthesis for cellular logic using EXORs have appeared in past in the literature which may be used to minimal implementations of binary and multiple-valued logic functions [3,8,10]. This paper presents a new tree searching program which generates AND/EXOR tree circuits. The Permutated-Reed-Muller-Tree (PRMT) searching program presented here, REMIT (REed-Muller Ideally permuted Trees), accepts a completely specified Boolean function in the form of an array of ON disjoint cubes as the input [4]. After decomposing the function with respect to all input variables according to Davio Expansion the program returns the exact PRM tree (the tree structure that minimizes the cost function) -- in version REMIT-EXACT, or the quasi-minimum PRM tree -- in version REMIT.

2. Davio Expansions

_Decomposition_ is commonly used in logic minimization. Using Shannon and Davio Expansions, the decomposition is achieved step-by-step with respect to all input variables. This kind of decomposition can be applied always, in contrast to other types of Boolean decompositions that are applicable to only some functions, and for which checking the decomposition possibility is very time consuming. The well-known _Shannon Expansion_ can be applied to decomposition with an EXOR gate [3]. It is called _Davio expansion_, and is defined as follows:

\[
f = x_i \cdot f_{x_i} \oplus \overline{x_i} \cdot f_{\overline{x_i}} \quad (1)
\]