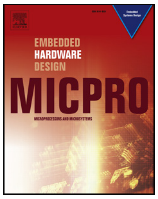
Microprocessors and Microsystems 60 (2018) 207–218

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro



Composition of switching lattices for regular and for decomposed functions

Anna Bernasconi[a], Valentina Ciriani[*,b], Luca Frontini[b], Gabriella Trucco[b]

[a] Dipartimento di Informatica, Università di Pisa, Italy
[b] Dipartimento di Informatica, Università degli Studi di Milano, Italy

ARTICLE INFO

Keywords:
Logic synthesis for emerging technologies
Switching lattices

ABSTRACT

Multi-terminal switching lattices are typically exploited for modeling switching nano-crossbar arrays that lead to the design and construction of emerging nanocomputers. Typically, the circuit is represented on a single lattice composed by four-terminal switches. In this paper, we propose a two-layer model in order to further minimize the area of regular functions, such as *autosymmetric* and *D-reducible* functions, and of decomposed functions. In particular, we propose a switching lattice optimization method for a special class of "regular" Boolean functions, called autosymmetric functions. Autosymmetry is a property that is frequent enough within Boolean functions to be interesting in the synthesis process. Each autosymmetric function can be synthesized through a new function (called *restriction*), depending on less variables and with a smaller on-set, which can be computed in polynomial time. In this paper we describe how to exploit the autosymmetry property of a Boolean function in order to obtain a smaller lattice representation in a reduced minimization time. The original Boolean function can be constructed through a composition of the restriction with some EXORs of subsets of the input variables. Similarly, the lattice implementation of the function can be constructed using some external lattices for the EXORs, whose outputs will be inputs to the lattice implementing the restriction. Finally, the output of the restriction lattice corresponds to the output of the original function. Experimental results show that the total area of the obtained lattices is often significantly reduced. Moreover, in many cases, the computational time necessary to minimize the restriction is smaller than the time necessary to perform the lattice synthesis of the entire function. Finally, we propose the application of this particular lattice composition technique, based on connected multiple lattices, to the synthesis on switching lattices of D-reducible Boolean functions, and to the more general framework of lattice synthesis based on logic function decomposition.

1. Introduction

Going beyond standard CMOS networks representing Boolean functions, the interest on new circuit models has grown in the recent literature. In this framework, the logic optimization of switching lattices for emerging nanoscale technologies have been proposed and discussed in [6,41].

A switching lattice is a two-dimensional network of four-terminal switches. Each switch is linked to the four neighbors of a lattice cell, so that these are either all connected or disconnected. A Boolean function can be represented using a switching lattice associating each four-terminal switch to a Boolean literal: if the literal has value 1 the corresponding switch is connected to its four neighbors, otherwise it is not connected. In this model, the Boolean function evaluates to 1 if and only if there exists a connected path between two opposing edges of the lattice, e.g., the top and the bottom edges (see Fig. 2 for an example). The synthesis problem on a lattice thus consists in finding an assignment of literals to switches in order to implement a given target function with a lattice of minimal size. The idea of using regular two-dimensional arrays of switches, to implement Boolean functions, was first proposed in a paper by Akers in 1972 [1]. Recently, with the advent of a variety of emerging nanoscale technologies, synthesis methods targeting lattices of multi-terminal switches have found a renewed interest [2–6,37,41].

Previous studies on this subject [14–16] have shown how the cost of implementing a four-terminal switching lattice could be mitigated by exploiting Boolean function decomposition techniques. The basic idea of this approach is to first decompose a function into some subfunctions, according to a given functional decomposition scheme, and then to implement the decomposed blocks with physically separated regions in a single lattice. Since the decomposed blocks usually correspond to functions depending on fewer variables and/or with a smaller on-set, their synthesis should be more feasible and should produce lattice implementations of smaller size. In the framework of switching lattice

[*] Corresponding author.
E-mail addresses: anna.bernasconi@unipi.it (A. Bernasconi), valentina.ciriani@unimi.it (V. Ciriani), luca.frontini@unimi.it (L. Frontini), gabriella.trucco@unimi.it (G. Trucco).

https://doi.org/10.1016/j.micpro.2018.05.004
Received 31 January 2018; Accepted 7 May 2018
Available online 14 May 2018

synthesis, where the available minimization tools are not yet as developed and mature as those available for CMOS technology, we are interested in reducing the size of the function to be minimized with a preprocessing phase. A smaller input function to a minimization algorithm can imply a smaller area circuit and a reduced synthesis time.

In this paper we study the lattice synthesis of a special class of *regular* Boolean functions, called *Autosymmetric* functions. Thus, the regularity of a function f of n variables is expressed by an *autosymmetry degree* k (with $0 \leq k \leq n$), computed in polynomial time. While the extreme value $k = 0$ means no regularity, for $k \geq 1$ the function f is said to be *autosymmetric*, and a new function f_k, called the *restriction* of f, is identified in polynomial time

In a sense, the restriction f_k is "equivalent" to, but smaller than f, depends on $n - k$ variables $(y_1, ..., y_{n-k})$ only, and the number of points of f_k is equal to the one of f divided by 2^k. Therefore, the minimization of f_k is naturally easier than that of f. The new variables $y_1, ..., y_{n-k}$ are built as EXOR combinations of the original variables, that is $y_i = EXOR(X_i)$, with $X_i \subseteq \{x_1, ..., x_n\}$. These EXOR equations are called *reduction equations*.

Although autosymmetric functions form a subset of all possible Boolean functions, a great amount of standard functions of practical interest fall in this class. For instance, the 24% of the functions from a classical collection of benchmarks [43] have at least one non-degenerated autosymmetric output, and their minimization time is critically reduced in the frameworks of SOP and SPP synthesis [22–24]. While the total number of Boolean functions of n variables is 2^{2^n}, the number of autosymmetric functions is $(2^n - 1)2^{2^{n-1}}$, which is much larger than the number of the classical symmetric functions, i.e., the ones invariant under any permutation of their variables, that is 2^{n+1} [22]. Note that an autosymmetric function f depends in general on all the n input variables, however we are be able to study f in a $n - k$ dimensional space; i.e., f is in general non-degenerated, whereas all degenerated functions are autosymmetric.

In [16] it is described a different lattice decomposition, based on the concept of "D-reducibility". D-reducible functions, similarly to autosymmetric functions, exhibit a regular structure that can be described using the EXOR operation. However, D-reducibility and autosymmetry are different regularities: autosymmetric functions can be studied in a new space whose variables are EXOR combinations of the original ones, while D-reducible functions are studied in a projection space producing an expression where the EXOR gates are in AND with a SOP form. There are examples of autosymmetric functions that are not D-reducible, and of D-reducible functions that are not autosymmetric.

The autosymmetry of a function f can be exploited in the minimization process, according to the strategy shown in Fig. 1. First detect the autosymmetry degree k of f. If $k > 0$, derive the restriction f_k of f and the corresponding reduction equations. Second, minimize f_k with any standard method: two level logic as SOP [34], Reed Muller [42]; three-level logic as SPP [13,31,38] (OR of ANDs of EXORs), EXSOP [35,36] (EXOR of ORs of ANDs), or switching lattices, as proposed in this paper. We note that, in the worst case, the lattice minimization requires time exponential in the number of points of the function, however, this number is strongly reduced for f_k if compared to f. We can finally construct a lattice for f from the one of f_k and from the reduction equations, whose computation requires some EXORs. This approach is convenient because: (i) the number of points of f_k is $|f|/2^k$ and f_k depends only on $n - k$ variables; (ii) the lattice minimization of f_k is naturally easier than that of f; and (iii) the number of EXORs that we add to the logic is at most $2(n - k)$. On the other hand, we require a second lattice containing the EXORs whose outputs are the input variables (i.e., $y_1, ..., y_{n-k}$) of the lattice for f_k. However, the reduction equations are in general EXORs of a very reduced number of variables and their lattice implementations have limited size.

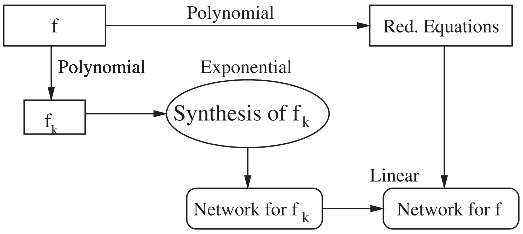


Fig. 1. Synthesis of an autosymmetric function f through the synthesis of its restriction f_k.

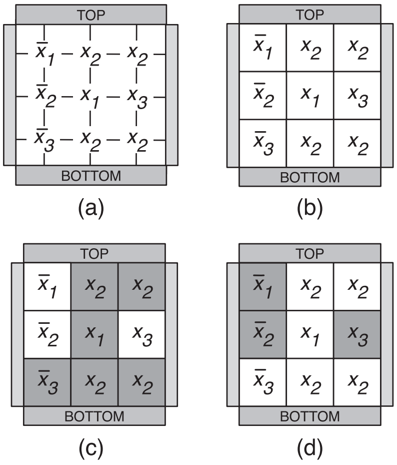

TOP
$\overline{x}_1 - x_2 - x_2$
$\overline{x}_2 - x_1 - x_3$
$\overline{x}_3 - x_2 - x_2$
BOTTOM
(a)
TOP
$\overline{x}_1$ x_2 x_2
$\overline{x}_2$ x_1 x_3
$\overline{x}_3$ x_2 x_2
BOTTOM
(b)
TOP
$\overline{x}_1$ x_2 x_2
$\overline{x}_2$ x_1 x_3
$\overline{x}_3$ x_2 x_2
BOTTOM
(c)
TOP
$\overline{x}_1$ x_2 x_2
$\overline{x}_2$ x_1 x_3
$\overline{x}_3$ x_2 x_2
BOTTOM
(d)

Fig. 2. A four terminal switching network implementing the function $f = \overline{x}_1\overline{x}_2\overline{x}_3 + x_1x_2 + x_2x_3$ (a); its corresponding lattice form (b); the lattice evaluated on the assignments 1,1,0 (c) and 0, 0, 1 (d), with grey and white squares representing ON and OFF switches, respectively.

Finally, we consider the application of this lattice composition technique, based on multiple lattices connected to each other, to the synthesis on switching lattices of D-reducible functions [8–11], and to the more general framework of lattice synthesis based on logic function decomposition, focusing in particular on the P-circuit decomposition model [19,26,27,29].

Experimental results show that by applying the proposed method to autosymmetric functions, we obtain more compact lattices and, in many cases, the computational time necessary to minimize the restriction is smaller than the time necessary to perform the lattice synthesis of the entire function. Moreover, our experimentation has confirmed that the application of this method turns out to be convenient also in other contexts of lattice synthesis based on functional decomposition.

This paper is an extended version of the conference paper in [17], where only the composition method for autosymmetric functions is described. The paper is organized as follows. Preliminaries on switching lattices and autosymmetric Boolean functions are described in Section 2. Section 3 discusses the problem of lattice composition, while Section 4 discusses the lattice implementation of autosymmetric functions. In Section 5, we show how to apply the proposed multiple-lattice composition approach both to the class of D-reducible functions, and in the general framework of lattice synthesis based on functional decomposition. Section 6 provides the experimental results and Section 7 concludes the paper.

2. Preliminaries

In this section we briefly review some basic notions and results on switching lattices [1,6,37] and autosymmetric Boolean functions [20–25,38].

2.1. Switching lattices

A switching lattice is a two-dimensional array of four-terminal